

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 *Curtiss-Wright / Synergy Microsystems ETVI* at our website: ***Click HERE***

V452 SERIES

VMEbus Single or Dual 68040/68060 Single Board Computer User Guide

Revision 1.0 **December 30, 1999**

SYNERGY microsystems

9605 Scranton Road
Suite 700
San Diego, CA 92121-1773

3895 N. Business Center Drive
Suite 100
Tucson, AZ 85705-6909

(858) 452-0020 · (858) 452-0060 (FAX)
Web: www.synergymicro.com

96-0391/UG-V452-01

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

V452 SERIES USER GUIDE

Copyright © 1997-1999 Synergy Microsystems, Inc.

This manual is copyrighted under Title 17 US Code of the United States Copyright Law. All rights are reserved by Synergy Microsystems, Inc. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, scanned, or reduced to any electronic medium or readable form without the express written consent of Synergy Microsystems, Inc.

This document contains material of a proprietary nature to Synergy Microsystems, Inc. All manufacturing, use, and sales rights pertaining to this product are expressly reserved. Distribution of this material does not convey any license or title under any patent or copyright. It is submitted in strict confidence to provide technical information for purchasers of this product or for those considering the purchase of the product. Each recipient, by accepting this document, agrees that its contents will not be disclosed in any manner or any person except to serve this purpose.

Synergy Microsystems, Inc. reserves the right to make changes to the specifications and contents in this document without prior notification. If in doubt, users are urged to consult Synergy to determine whether any such changes have been made.

Synergy products are not intended for use in life support systems or other applications where a failure of the product could result in injury or loss of life. Customers using or selling this product in systems or applications serving such a function do so at their own risk and agree to fully indemnify Synergy Microsystems, Inc. for any and all damages arising from improper use.

This product and associated manuals are sold “AS IS” without implied warranty as to their merchantability or fitness for any particular use. In no event shall Synergy Microsystems, Inc. or anyone involved in the creation, production, or delivery of this product be liable for any direct, incidental, or consequential damages, such as, but not limited to, loss of anticipated profits, benefits, use, or to data resulting from the use of this product or associated manuals or arising out of any breach of warranty. In states that do not allow the exclusion or limitation of direct, incidental, or consequential damages, this limitation may not apply.

Synergy™, V452 Series™, V4xx Series™, V30 Series™, V20 Series™, R452™, and EZ-bus™ are trademarks of Synergy Microsystems, Inc..

Synergy wishes to acknowledge that the names of products and companies mentioned in this manual are trademarks of their respective manufacturers.

PRINTED IN THE USA

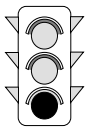
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Manual overview

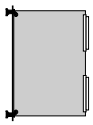
This manual is divided into the following sections that can be identified by an icon appearing on the upper outside corner of each page.



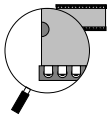
1 – Overview describes the models, options, features architecture, specifications and revisions made to V452 Series boards.



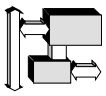
2 – Getting Started contains procedures for installing V452 Series hardware and hardware options.



3 – Board Facilities describes board-level functions and options including the addresses, interrupts, jumpers, control registers and default conditions.



4 – Local Components describe the operation of local bus components including the microprocessor, CPU mailbox, watchdog, RAM, and timer/counters.



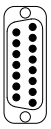
5 – Interface Options describe the on-board circuitry that communicates with external devices including the EZ-bus, the serial I/O interface, the VMEbus interface and the System Controller.



6 – Code Examples contains programming examples for selected V452 Series components or processes.



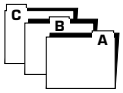
7 – Warranties & Service provides complete information about Synergy product warranties and service.



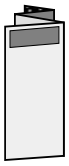
Appendix A — Cables & Connectors describes the V452 cables and connectors.

Appendix B — Specifications lists the V452 specifications.

Appendix C — Board Revision Summary summarizes the V452 change history.



Glossary & Index defines terms used in the manual and provides an alphabetically arranged index.



Each manual includes the ***V452 Series Quick Reference Card*** which lists the in-formation you're likely to need for day-to-day use in a convenient shirt-pocket size.

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

Table of contents

Preface vii

Manual revision summary x

Section 1, Overview

V452 Series features 1-3
- Physical configuration 1-5
- Functional block diagram 1-6
- Feature summary 1-7

Comparison: V440/V460 and V452 SBCs 1-11

Section 2, Getting Started

Minimum system requirements 2-3

Installing a monitor PROM 2-7

Installing the R452/R453 memory module 2-13
- Installing/upgrading the R452/R453 memory module 2-15

Setting up the V452 Series hardware 2-17
- Default configuration 2-18
- Configuration tasks 2-18
  - Enable/Disable Flash write 2-20
  - Set the VMEbus request level 2-21
  - Enable VME Slave remote reset 2-22
  - Select Round Robin or Priority request handling 2-23
  - System controller 2-24
  - Boot select, EPROM or onboard Flash 2-25

Installing V452 Series CPU boards 2-27
- Power-on banners 2-30

Setting up the V452 Series software 2-35
- Default (pre-initialization) conditions 2-36
- Initialization tasks 2-38

Section 3, Board Facilities

Address map 3-3

Interrupts 3-7
- Interrupt vectors 3-7
- Enabling/disabling interrupts 3-9
- Interrupts on single vs. dual processor boards 3-11
- On-board interrupt sources 3-12
- Local bus timeout 3-19
- VMEbus interrupt handler 3-19
- Configuring the VMEbus interrupter/resetter 3-22

Jumpers, switches, LEDs & Fuses 3-25
- Jumpers 3-26
- Front panel 3-29
- 8-bit ID switch (front panel) 3-30
- Toggle switches 3-31

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

LEDs 3-32
Fuses 3-35

V452 Series internal registers 3-37
Status register 3-38
ID status register 3-39
Mode registers 3-40
Using Mode register functionsPrimary & Extended Mode registers 3-41
Controlling the User LEDs 3-42
2692 mode registers 3-45
VME interrupt vector register 3-49
Control registers 3-50
Primary & extended control registers 3-51
Interrupt control registers 3-52
Slave interface control register 3-54
Ethernet/VMEbus control registers 3-55
Board information registers 3-55
Slot ID, FE39 0001 3-56
CPU & board type, FE39 0003 3-57
Memory size and installed options, FE39 4003 3-57
Modifications and ECO level, FE39 8003 3-58
PCB revision, FE39 C003 3-58

Default & reset conditions 3-59
Default conditions 3-59
Reset sources 3-61
Reset sequence 3-63
Reset via software 3-65
VME Level 0 interrupt reset 3-65
Watchdog timer enable 3-66
Remote reset register 3-67

Section 4, Local Components

68040 CPU 4-3
Introduction 4-3
Additional 68040 documentation 4-4
Programming model 4-8
Instruction set overview 4-14
Exception processing 4-16
Operand transfer types 4-23
Bus snooping 4-24
Memory management unit 4-27

68060 CPU 4-31
Introduction 4-31
Additional 68060 documentation 4-33
Programming model 4-33
Programmer's model differences: 68060 vs. 68040 4-35
Data type differences: 68060 vs. 68040 4-36
Addressing mode differences: 68060 vs. 68040 4-37
Instruction set overview 4-37
Integer unit 4-39
Exception processing 4-40
Exception differences: 68060 vs. 68040 4-41

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

Instruction and data caches 4-43
Cache differences: 68060 vs. 68040 4-43
Cache organization 4-44
Cache coherency 4-45
Setting up the 68060 caches 4-45
Paged memory management unit (PMMU) 4-48
PMMU differences: 68060 vs. 68040 4-48
PMMU architecture summary 4-49

CPU Mailbox 4-51
Mailbox memory areas 4-51
Mailbox interrupts 4-53

CPU Watchdog 4-57

Dynamic RAM 4-63
DRAM access optimization 4-63
DRAM tuning strategies 4-64
Memory parity 4-65
Clearing the parity error bit 4-68
Detecting & isolating bad parity 4-68
Multi-port memory contention 4-69
DRAM address decoding 4-70
R452/R453 memory modules 4-72

EPROM 4-73
Selecting the monitor EPROM type 4-74
Flash EPROM configuration 4-75
EPROM use 4-76
Flash considerations 4-77

Flash memory module 4-79
Installing the Flash module option 4-80
Accessing the Flash module 4-82
Block organization 4-82
Flash considerations 4-83

Onboard flash memory 4-85
Introduction 4-85
Writing and erasing 4-86
Block organization 4-86
Booting from onboard Flash 4-87

Timers & counters 4-89
16-bit timers (2692) 4-91

Clock calendar 4-101

Non-volatile 8K x 8 SRAM 4-105
Replaceable battery 4-106

Section 5, Interface Options

Asynchronous serial interface 5-3
P2 access to serial interface 5-6

Ethernet 10Base-T interface option 5-7
Introduction 5-7
Ethernet network connections 5-9

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

Ethernet ID 5-11
Avoiding bus contention — CSMA/CD 5-11
Interchange signals 5-12
Address map 5-13
Interrupts and vectors 5-13
EZ-bus interface 5-15
EZ-bus modules 5-16
EZ-bus connectors 5-18
VME Slave interface 5-21
Setting up the VME Slave interface 5-22
Configuring Slave write access memory protection 5-24
Enabling/disabling the Slave interface 5-31
Multi-port memory access contention 5-32
VME Slave block transfer (BLT) 5-33
Data broadcasting 5-35
Hardware setup for data broadcasting 5-36
Setting up data broadcasting groups 5-37
Enabling/disabling data broadcasting 5-40
Configuring a sample data broadcast group 5-41
VME Master interface 5-47
Setting up the Master interface 5-47
VME Master data transfer bandwidth 5-51
VME Master BLT 5-53
System Controller 5-65
Forcing system controller 5-66
Configuring the bus arbiter 5-66
VMEbus arbitration timeout 5-67
Configuring the bus error timeout interval 5-67
Enabling VME SysFail interrupts 5-68

Section 6, Code Examples

Programming differences 6-3
Block transfer (BLT) with DMA example 6-5
Flash EPROM programming tools 6-15
Timer code examples 6-25
2692 DUART code example 6-33

Section 7, Warranties & Service

Warranty terms & options 7-3
Customer service 7-5

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

Appendices

Appendix A, Cables & Connectors A-1
VMEbus connectors (P1 & P2) A-3
EZ-bus connectors (P3 & P4) A-5
Memory module connectors (P9, P10 & P11) A-7
Ethernet 10Base-T connector (P8) A-11
Asynchronous serial connectors (P5–P7) A-13
Serial I/O cabling options A-15
P2 serial interface option A-29

Appendix B, Specifications B-1
VMEbus compliance B-1
Physical dimensions B-1
Weight B-1
Power requirements B-2
Operating environment B-2
Number of slots B-2
Board layout B-3

Appendix C, Board Revision Summary C-1

Glossary & Index

Glossary Glos-1

Index Indx-1

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

Preface

This manual provides detailed information on how to install, configure, and operate the V452 Series of single-board computers (SBCs). This chapter will help you use the contents of this document to your best advantage. It is designed to help you:

- **evaluate** the V452 Series for purchase:

  To aid your buying decision, the ***Overview*** contains pertinent information to help you determine if the V452 Series meets the needs of your system design. These chapters include technical descriptions, a list of features, and general specifications.

- **start up** your V452 Series board:

  Getting Started contains the information you need to install your system for the first time. It outlines minimum system requirements, configuration checklists and installation procedures.

  For previous customers of Synergy CPU boards, the ***Getting Started*** section also includes a chapter that highlights new product features and describes the primary operational differences between V452 Series boards and Synergy's V30 Series and V20 Series boards.

- **answer** technical questions about the V452 Series:

  The ***Board Facilities, Local Components,*** and ***Interface Options*** sections form an in-depth reference for using the V452 Series and peripherals.

  Often-used information is included in a pull-out ***V452 Series Quick Reference Card***.

Typographical conventions

This manual observes the following conventions:

1. The term **V452 Series** is used in conjunction with information that applies to **ALL** models of the board series. When differences among models exist, specific model numbers are used to described any special features.
2. In diagrams and descriptions in this manual, signal names followed by a backslash (\) are **active low**.

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

Related publications

Special attention has been paid to making this manual as complete as possible to minimize the number of times you may need to reference other manuals. However, should you need additional information, the following books expand on several related topics:

Synergy publications

- *V4xx Series 68040 VME Single Board Computer User guide* — describes the features, installation, and operation of Synergy's V4xx Series of CPU boards.
- *V30 Series 68030 VME Single Board Computer User guide* — describes the features, installation, and operation of Synergy's V30 Series of CPU boards.
- *V20 Series 68020 VME Single Board Computer User guide* — describes the features, installation, and operation of Synergy's V20 Series of CPU boards.
- *Datasheet Supplement, VME CPU Version* — contains reprints of selected portions of the manufacturer's datasheets for major components used on Synergy VME SBCs. Each included datasheet appears in the Datasheet Supplement courtesy of their respective manufacturers.
- *EZ-bus™ Designer's Guide* — describes the functions and specifications of the Synergy EZ-bus. All Synergy VMEbus SBCs include two EZ-bus connectors allowing the connection of up to two EZ-bus interface or I/O modules. This manual describes all of the available daughter modules and the necessary mechanical and interface information required to create new modules that are compatible with the EZ-bus.
- *SMon User Guide — describes* Synergy's monitor PROM and application development/debugger software. SMon provides a powerful cross-plaform development environment with full support for host/target communication over a TCP/IP Ethernet network.

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

IC manufacturer's publications

- *M68060™ Microprocessor User's Manual*; Rev. 1; Motorola® — (520) 994-6561.
- *MC68040™ Enhanced Microprocessor User's Manual*; 2nd Edition; Prentice Hall® or Motorola® — (520) 994-6561.
- *MC68000PM/AD, MC68000 Programmer's Reference Manual*; Prentice Hall® or Motorola® — (520) 994-6561.
- *MC68881/MC68882™ Floating Point Co-processor User's Manual*; 2nd Edition; Prentice Hall or Motorola.

VMEbus specification & publications

- *The VMEbus™ Specification*; Rev D.1; VMEbus International Trade Association (VITA) — (602) 951-8866.
- *The VMEbus™ Handbook*; 2nd Edition; VMEbus International Trade Association (VITA) — (602) 951-8866.

Software manufacturer's publications

- *pSOS^{+}/68k™ Evaluation Package User's Guide*; Software Components Group; — (408) 437-0700.
- *Using Professional OS9*, Microware Systems Corporation
- *VxWorks™ Reference Manual*; Version 5.1; Wind River Systems®; — (800) 545-WIND.

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

Manual revision summary

Revision level	Revision date	Section	Affected chapter/description
1.0a	12/21/96		1st preliminary release
1.0b	3/26/97		2nd preliminary release
1.0c	1/8/99		3rd preliminary release **Section 3**/Corrected configuration jumper information for Flash Boot.
1.0d	5/24/99		4th preliminary release **Section 0, Section 9**/Revised for new telephone area code.
1.0	12/30/99		Initial GA release.

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



Overview

The following chapters introduce the **V452 Series** of single-board computers.

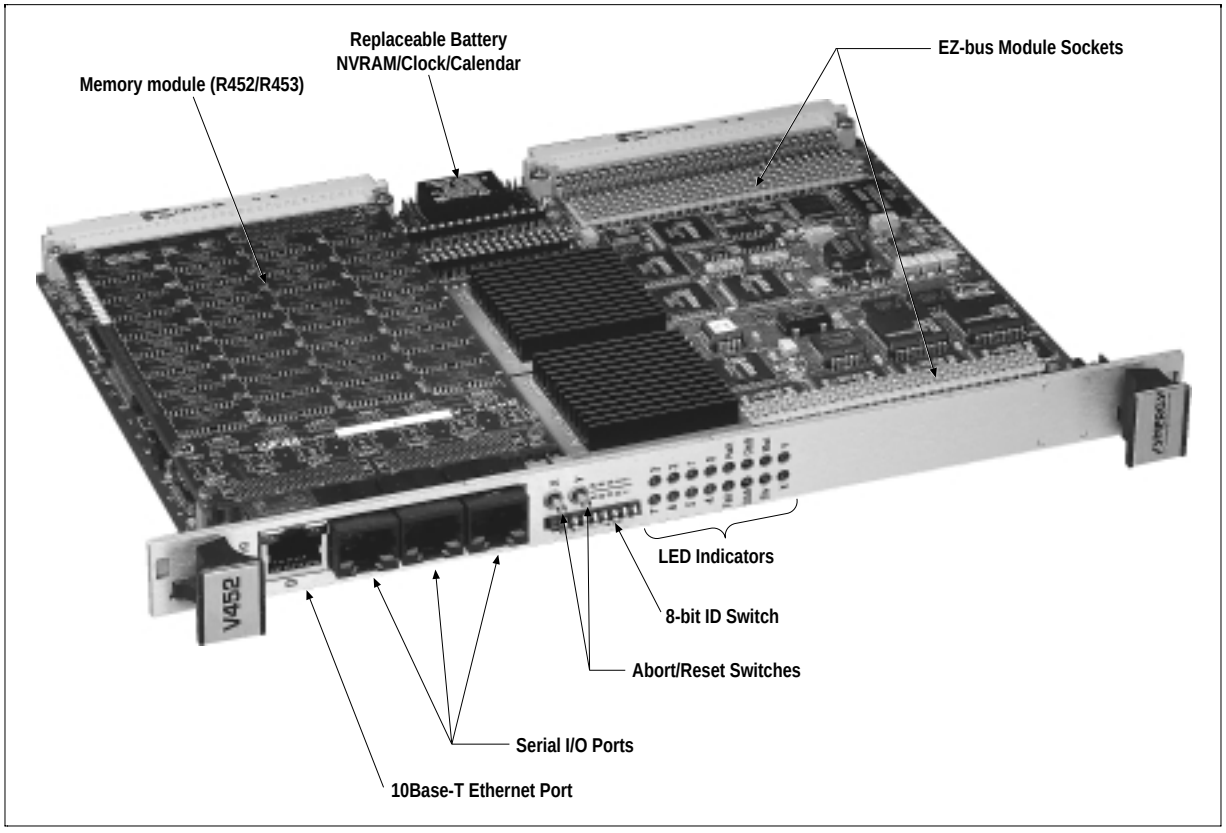
- V452 Series features
- Comparison: V440/V460 and V452

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

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

V452 Series features

The V452 Series are high-performance, VMEbus-compatible single board computers (SBCs) using single or dual Motorola **MC68040 or MC68060 microprocessors**. The general component side layout of the V452 SBC is shown below.



V452 Series single board computer

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

The V452 Series offers flexibility and performance to satisfy a wide range of applications. Standard features include low power operation, Bellcore/Nynex compliant design, auto system controller, and VME64 extensions support.

The V452's **EZ-bus** interface is a high-performance daughter module bus that supports DMA burst accesses to motherboard memory. The EZ-bus interface affords valuable configuration flexibility for your system designs. All necessary signals and a 32-bit data bus are provided to implement many optional features including:

- VSB interfaces
- Ethernet, SCSI and/or serial interfaces
- custom P2 I/O
- multiple serial ports and other uses

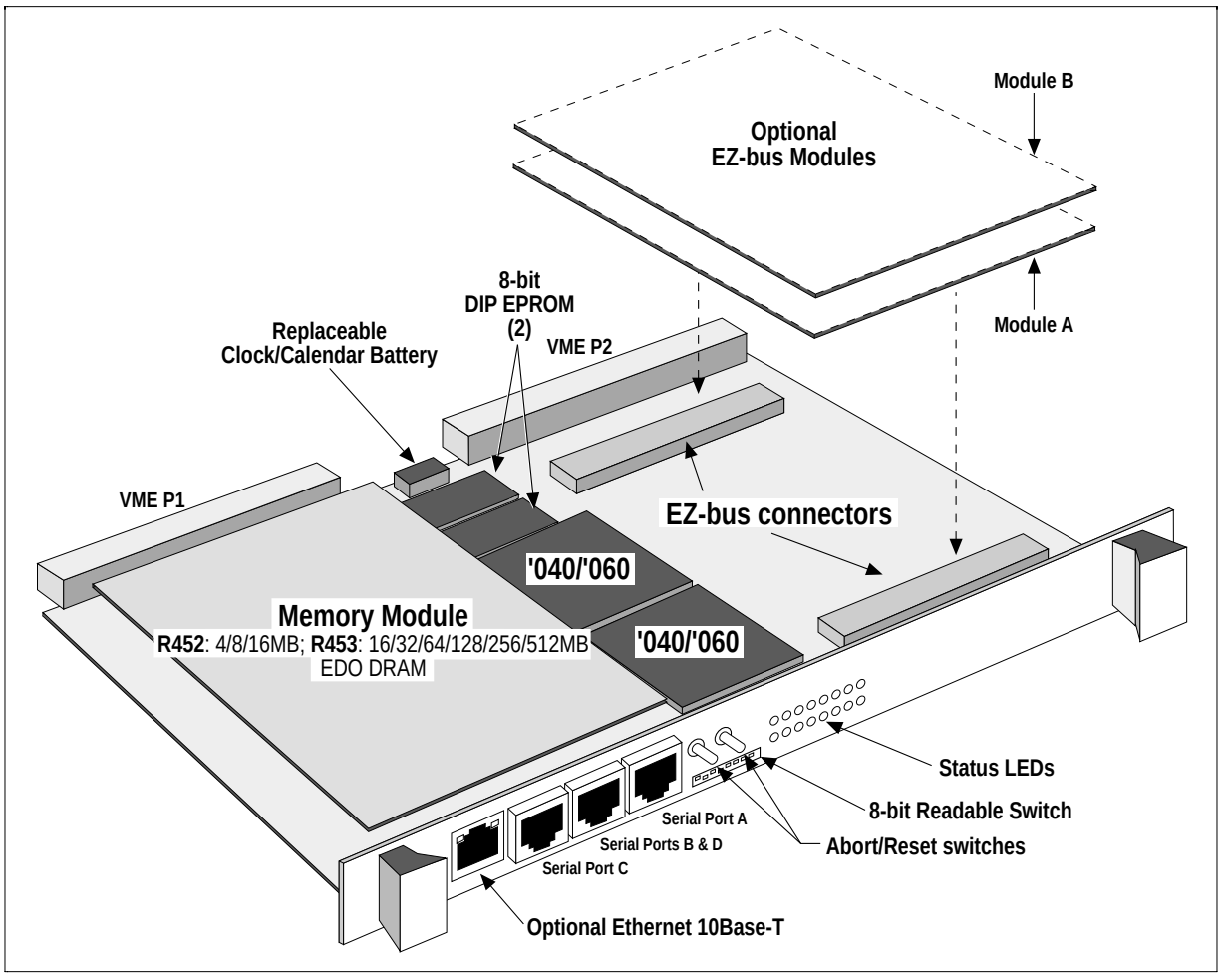
Standardized and custom-designed daughter modules are also available providing true single-board solutions for a wide range of processing and interface needs. The V452 is provided with an enhanced EZ-bus that accommodates PCI bus-based devices on the EZ-bus module. This broadens the range of EZ-bus solutions to meet the requirement of most any application.

A full line of system **monitor**, **kernal**, and **operating system** software/ firmware is also available from Synergy and leading developers.

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

Physical configuration

The V452 SBC uses a 6U VME form factor. Onboard connectors are used to attach modular memory and I/O options (EZ-bus modules). Four RJ-45 jacks are mounted on the front panel; one is for the optional Ethernet 10Base-T port with the remainder providing 4 serial ports (serial ports B and D share a jack). Also on the front panel are abort & reset switches, 16 status LEDs, and a readable 8-bit DIP switch.

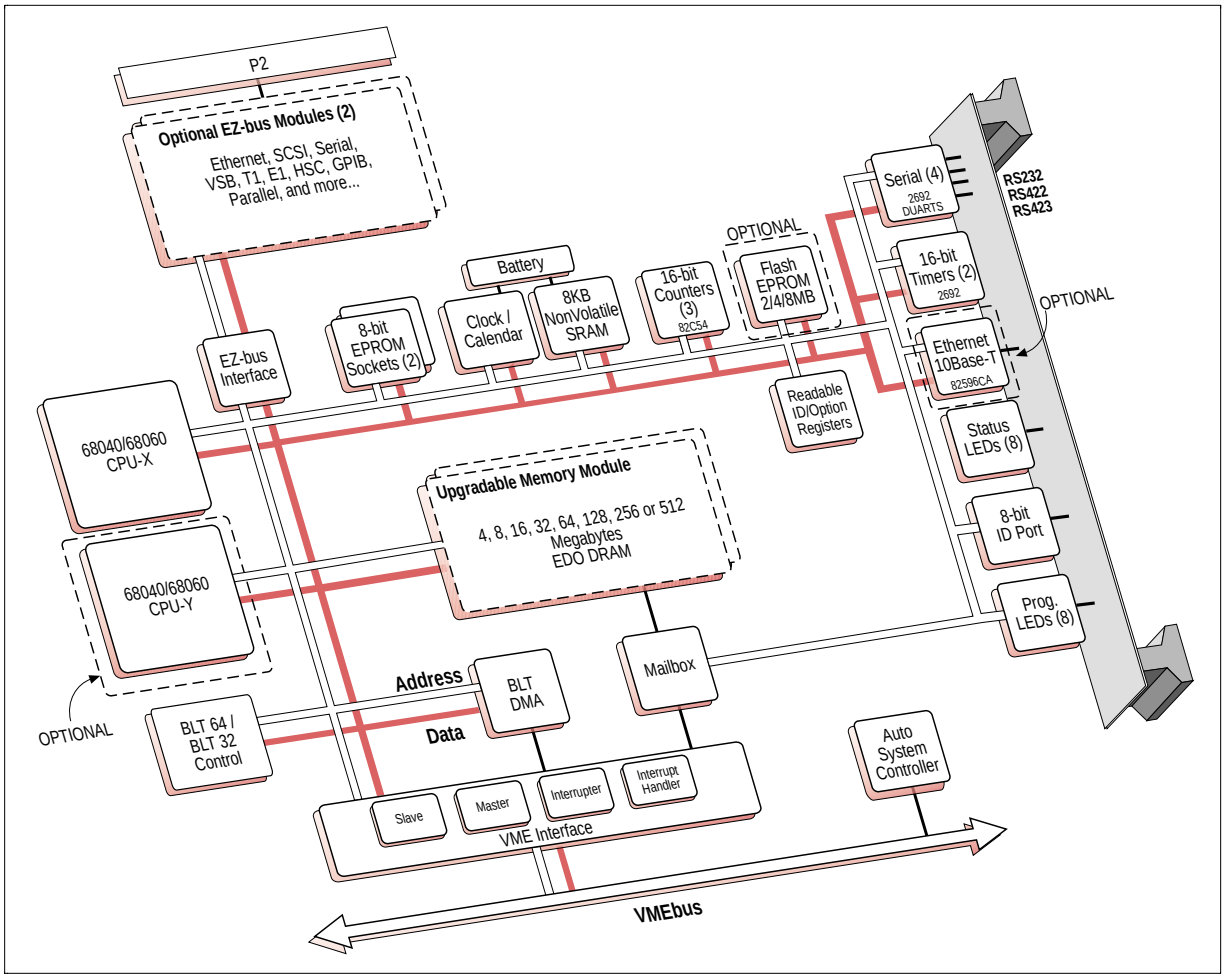


V452 configuration

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

Functional block diagram

The figure below shows the functional block diagram for the V452 SBC.



V452 functional block diagram

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



Feature summary

The V452 Series provides the following list of powerful features and functions that are common to all of the boards in the V452 Series product line:

CPU

- Motorola® **68040** 32-bit CPU(s) featuring **16** 32-bit registers, independent internal 4KB **instruction** & **data caches**, a **19/25 MIPS** processor core, a built-in **3.6/4.7 MFLOPS** floating-point and integer coprocessor, and a **25/33** MHz CPU clock speed.
- Motorola® **68060** 32-bit CPU(s) featuring a RISC-like architecture that includes branch prediction logic and superscalar pipelines. The '060 is user object code compatible with the 68040 and all previous 68K processors. Compatibility is assured through the use of dual '040-compatible CPU integer cores, an '040-compatible floating point core, 8KB instruction and data caches, an '040-compatible paged memory management unit, and bus controller. The processor runs internally at **50/66** MHz with external bus operation at **25/33** MHz.

Memory

- 4, 8, 16, 32, 64, 128, 256 or 512 MB of **EDO** dynamic RAM with a 32-bit data path on modular daughter modules for configuration flexibility and for easy memory upgrades in the field.
- Support for the 68040 optimized burst mode.
- EDO DRAM (Extended Data Out DRAM) reduces DRAM cycle time yet provides valid data for much longer time for increased read/write access performance over conventional DRAM designs.
- Two independent 32-pin **EPROM** sockets provide up to 2 MB of PROM Monitor space consisting of EPROM, FLASH EPROM or a combination of these devices. These sockets also accept special modules that provide up to 16 MB of Intel FlashFile™ memory in addition to the normal 1 MB space allocated for the boot EPROM.
- Optional 2/4/8 MB of onboard Intel FlashFile™ memory.

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

Construction

- Bellcore/Nynex-compliant construction.
- Ruggedized option (single CPU only) available.
- Optional conformal coat.

VMEbus

- **BLT64** Master/Slave Block Transfers (BLT) DMA to **72 MB/sec** supporting VME Specification Revision D.1.
- **BLT32** Master/Slave Block Transfers (BLT) DMA support for transfers up to **33 MB/sec**.
- Supports the **bus snooping** for write accesses to local memory via the VMEbus Slave interface.
- VME interface supports A32/24/16 and D32/16/8 in **Master** mode; A32/24 and D32/16/8 in **Slave** mode.
- VMEbus **requester** supports Release-When-Done (RWD) or Release-On-Request (ROR) for bus release and FAIR bus request arbitration.
- VMEbus **priority request level** is software or jumper-selectable at 4 different priority levels.
- **Automatic System Controller** lets the SBC configure itself automatically as the System Controller when installed in slot 1. The System Controller provides prioritized and/or round-robin arbitration, bus arbiter timeout, and a programmable VMEbus grant timeout period. The System Controller is isolated for continued operation in the unlikely event of a failure elsewhere on the board. The System Controller circuit is able to assert the VMEbus Reset signal under software control without causing a local reset of itself. A jumper is provided to disable the automatic System Controller feature to ensure compatibility with other boards in the system.
- **Software programmable** VME Slave interface featuring a Slave access window size equal to installed memory that can be assigned to **any location** within the entire 4 GByte VME address space.
- Supports unaligned VMEbus transfers.
- Software-controllable **data broadcast mode** for simultaneous data transfers to multiple VME Slaves.

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



- Supports **remote reset** under software control by a remote VME Master.
- Standard **6U** VME card format with VMEbus P1 and P2 connectors.

VME64 extensions

- Supports Auto System Controller.
- Supports readable System ID and configuration.
- Optional **wide** 160-pin P1 and P2 connectors for VME64 style I/O and 3V Vcc backplane.

Interrupts

- Software-controllable **interrupt reception** for on-board interrupt sources via programmable register.
- Programmable **interrupt level assignments** for on-board interrupt sources via a socketed Programmed Array Logic (PAL) device.
- VMEbus **interrupter** can assert interrupts at any level under software control.
- VME **interrupt handler** for any group of up to 7 VMEbus interrupt priority levels.
- Interrupt-driven CPU **Mailbox** for inter-CPU and inter-process communications.
- **Remote** reset via VME writeable register.
- **Watchdog** timer can reset board if software hangs or halts.
- **Local bus time-out** circuit asserts a local bus error to the CPU when individual data transfers across the local bus take longer than 30 microseconds.

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

Peripherals

- Four **RS-232/RS-423** single-ended, or 2 **RS-422A** differential and 2 **RS-232/RS-423** single-ended serial channels for asynchronous serial communication up to 38.4K baud terminating in three RJ-45 10-pin connectors on the front panel.
- Optional onboard 10Base-T Ethernet port.
- EZ-bus daughterboard interface is provided for **custom P2 I/O**, including VSB, memory expansion, SCSI, Ethernet, or high-speed floating point coprocessor. Synergy offers a standard set of off-the-shelf boards or can custom design for specialized needs. Customers may design their own daughter modules by using the EPRO/VPRO prototype board and the *EZ-bus Designer's Guide*, both available from Synergy. The V452 EZ-bus interface supports PCI-like initiator-throttled DMA accesses.

System timer/counter facilities

- Five **16-bit counter/timers** with interrupt support. The two timers from the two **2692** chips feature prescalers. The three counters from the **82C54** chip allow *on-the-fly* counter readings.
- **Time-of-day** clock/calendar with 8KB of non-volatile RAM for general use and with user-replaceable battery.

User interface

- Eight front panel **user-programmable** LEDs indicating application events and conditions.
- Eight front panel **status** LEDS indicating the **HALT** and/or **SysFail** status of the on-board CPUs, whether the V452 Series board is currently operating as the **VMEbus Master**, or whether an on-board CPU(s), EZ-bus module or an external VMEbus Master is in control of the **local bus**.
- **RESET** toggle switches can assert a CPU, board, and/or VMEbus level RESET signal; nmi **ABORT** switch for each on-board CPU.
- Front panel **Link** and **Transmit** status LED indicators for optional onboard 10Base-T Ethernet port.
- Eight position software-readable **ID switch** for processor identification or option selection.

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



Comparison: V440/V460 and V452 SBCs

For information reference, this chapter summarizes key differences between the Synergy Microsystems V440/V460 and V452 SBCs.

Memory Architecture

- V440/V460 — maximum of 64MB w/double-stacked memory module using EDRAM chips
- V452 — maximum of 512MB w/quad-stacked memory module using EDO DRAM chips

Low Power

- V440/V460 — not originally designed for low power
- V452 — designed from ground up for low power

EZ-bus interface

- V440/V460 — conventional (non-enhanced) EZ-bus
- V452 — enhanced EZ-bus; accommodates PCI-bus based chips on EZ-bus module for wider selection of I/O options

Flash

- V440/V460 — optional Flash EPROM option (512KB max.) or Flash memory module (DELF/DEFL) up to 16MB
- V452 — optional onboard 2/4/8MB Flash or Flash memory module (DELF/DEFL) up to 16MB

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

RTC NVRAM and Battery

- V440/V460 — 2KB NVRAM; battery not user replaceable
- V452 — NVRAM, 8KB (user-selectable) or 2KB (default) for compatibility with pre-V45x SBCs; battery user replaceable

Bellcore/Nynex compliance

- V440/V460 — not compliant
- V452 — compliant (no parts on rear, no parts under parts, no dissimilar metal connectors, minimal use of sockets and jumpers)

Ethernet interface

- V440/V460 — needs optional EZ-bus module support for AUI, 10Base-2, 10Base-T
- V452 — optional onboard 10Base-T port accessible from front panel and P2 user I/O

CPU Support

- V440/V460 — single or dual '040 or '060 (provided as V450)
- V452 — single or dual '040 or '060

Ruggedized

- V440/V460 — not available
- V452 — option available for single CPU boards only

Auto System Controller

- V440/V460 — not available
- V452 — comes standard

VME64 extensions support

- V440/V460 — not available
- V452 — comes standard with auto system controller and readable system ID; wide (160-pin) P1/P2 available as option

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



Conformal coat

- V440/V460 — not available
- V452 — available as option

Front panel with EMI shielding

- V440/V460 — not available
- V452 — available as option

Readable board information register

- V440/V460 — not available
- V452 — comes standard (indicates slot ID; CPU and board type; memory size and installed options; Mods/ECO level; and PCB revision)

VMEbus request level

- V440/V460 — jumper selectable
- V452 — jumper and software selectable

Serial I/O via P2 connector

- V440/V460 — supported (TxD, RxD, and handshaking signals)
- V452 — supported (TxD and RxD signals only)

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

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



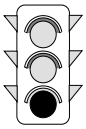
Getting Started

2

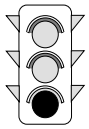
This section provides configuration, setup and general information for the V452 SBC.

- Minimum system requirements
- Installing a monitor EPROM
- Installing the R452/R453 memory module
- Setting up the V452 Series hardware
- Installing V452 Series CPU boards
- Setting up the V452 Series software

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



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



Minimum system requirements

The following system components are required to install and test V452 Series boards:

- **6U VMEbus-compatible card cage with P1 backplanes installed** — A card cage with forced air cooling is required. In addition, if 64 or 32-bit addresses or 64 or 32-bit (BLT or D32) data transfers are to be used, a P2 backplane must also be installed.

  V452 Series boards feature state-of-the-art, high-speed, transfers across the VMEbus that in some cases may approach the maximum VME specifications for transfer speeds. As a result, to support these transfers the underlying connectors, circuitry, and PCB boards used in the VME card cage must be constructed of high-quality materials that are fully-compliant with VME specifications.

  For example, VME card cages containing 10-layer, PCB boards are normally required to support high-speed VME transfers. Older style, card cages containing 6-layer boards may have some difficulty conducting these signals without generating excessive noise.

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

Pin row **B** of the P2 backplane is defined by VMEbus specifications and is bussed across the entire backplane. Pin rows **A** and **C** are user configured and, if connected at all, are normally connected to adjacent slots via wirewrap or special cables.

Because the P2 pinout may vary between backplanes or even slots in the same backplane, **DO NOT INSTALL the V452 Series into a system slot whose P2 backplane is not compatible with the V452 Series' P2 pinout. Failure to observe this warning can cause the complete destruction of many on-board components and also voids the product warranty.**

The V452 Series pinout meets standard VME specifications for row ***B***, but rows ***A*** and ***C*** will vary according to the EZ-bus daughterboard installed. Daughterboard pinouts are shown in the associated daughterboard manual. If no daughterboard is present, P2 backplane rows ***A*** and ***C*** are defined as ***no-connects***.

For a complete list of the V452 Series P2 assignments, see the ***VMEbus connectors (P1-P2)*** chapter in Appendix A.

- **Power supply** – Standard V452 Series boards require the following power supply voltage levels:

  V452 single processor, typical power consumption:

  '040/33 MHz with 16MB DRAM:

  +5.0v ±5% = 3.4 Amps

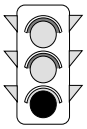
  -12.0v ±5% = 30 mA

  '060/50 MHz with 16MB DRAM:

  +5.0v ±5% = 3.0 Amps

  -12.0v ±5% = 30 mA

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



V452 dual processor, typical power consumption:

'040/33 MHz with 16MB DRAM:

+5.0v ±5% = 4.1 Amps

-12.0v ±5% = 30 mA

'060/50 MHz with 16MB DRAM:

+5.0v ±5% = 3.8 Amps

-12.0v ±5% = 30 mA



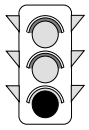
Ensure that the power supply is capable of meeting the above requirements plus the requirements of any additional boards in the system. An extra 20% margin of current capacity should be factored in for safety.

- **One modular serial I/O cable** (and Modular-to-D adapter if necessary) **—** The serial ports on V452 Series boards use 10-pin modular telephone-type (AT&T) connectors. A Modular-to-D adapter is required to connect the modular jacks to serial devices (e.g, terminals) that have D-type serial connectors. Additionally, the use of both Serial Ports B and D require the use of a special adapter. Refer to the ***Serial cabling options*** chapter in Appendix A.

- **RS-232 compatible video display terminal**

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

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



Installing a monitor PROM

V452 Series boards provide two, 32-pin, 8-bit, DIP EPROM sockets that can accept any combination of the following devices:

- 27C010 1 Mbit EPROM (128 KB)
- 27C020 2 Mbit EPROM (256 KB)
- 27C040 4 Mbit EPROM (512 KB)
- 27C080 8 Mbit EPROM (1 MB)
- 28F010 1 Mbit Flash EPROM (128K x 8, 256 KB)
- 28F020 2 Mbit Flash EPROM (256K x 8, 512 KB))

Normally, all boards ship from the factory with the appropriate monitor PROM already installed, however new or updated PROMs can also easily be added or changed in the field.

The paragraphs below describe a field installation of a new EPROM and all of the potential configuration changes you may need to make to the V452 Series CPU board as a result:



If the desired monitor PROM(s) is(are) already present on the V452 Series board, proceed to the next chapter in this section.

If the optional Flash memory module is to be installed instead of EPROM(s), or for information about the Flash memory module itself, see the ***Flash memory module*** chapter in Section 4.

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

Materials

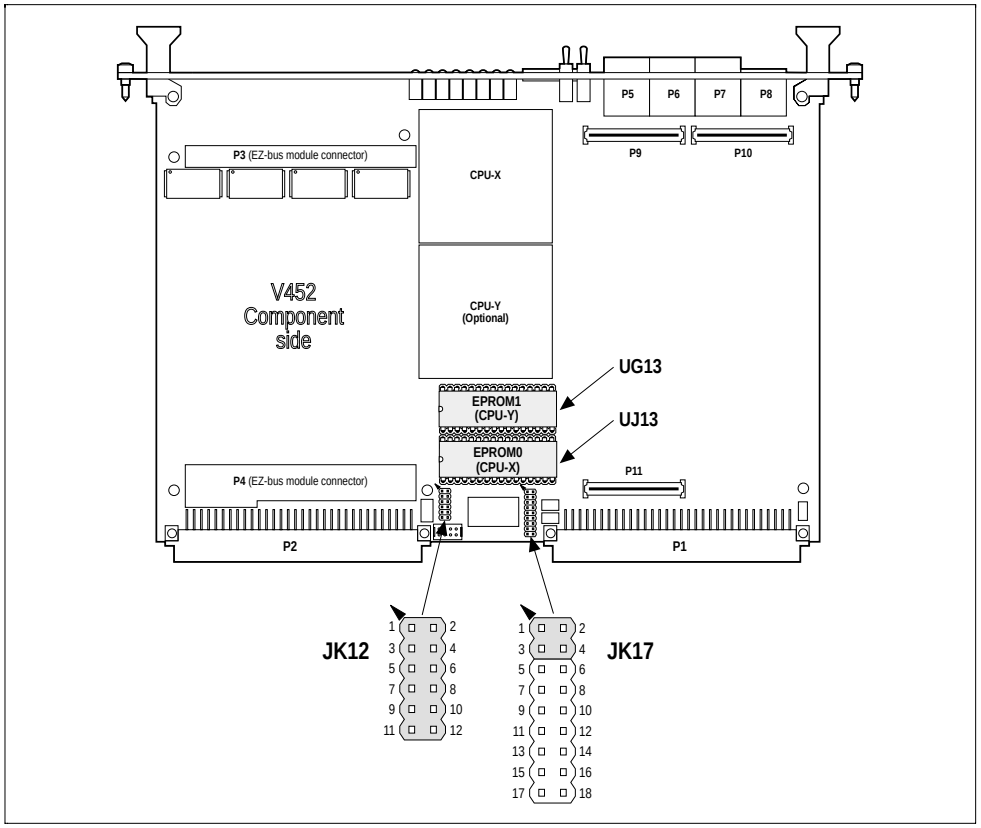
To complete this procedure, you will need the following materials:

- The desired monitor firmware PROM(s),
- A chip extractor tool to remove the current PROM (if necessary),
- The manual for the software product on the new EPROM.

Procedural steps

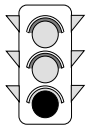
To install a monitor EPROM, complete the following procedure:

❶ **Verify proper operation of the motherboard** (if replacing an existing monitor PROM) — Before attempting to install a new EPROM on an existing board, consider checking that the motherboard (and any attached daughterboards) are operating properly by performing the installation/checkout procedure for the V452 Series board as described later in this section.



Monitor EPROM sockets and jumpers on V452 Series boards

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





The black triangle near the jumper block and connectors points to pin 1.

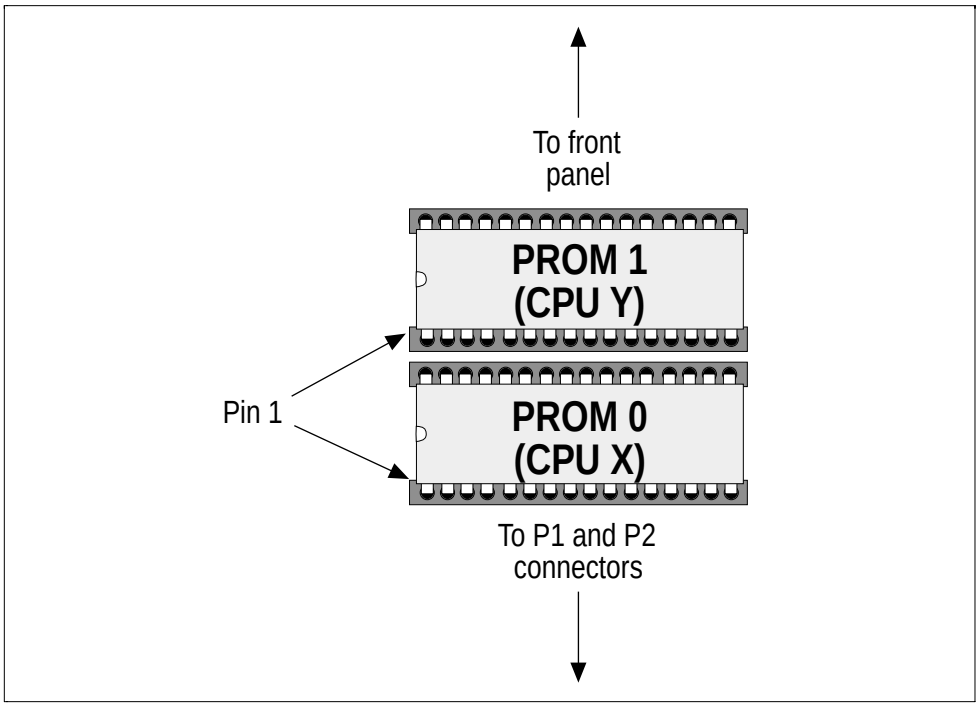
❷ **Power-down and remove the CPU motherboard from the card cage** (if necessary) — Power-down the system and remove the V452 Series CPU board from the card cage.



Synergy CPU motherboards contain static-sensitive devices. Make sure you are properly grounded (by putting on a ground-strap, touching the system power supply, etc.) before removing and handling the board.

❸ **Locate the current monitor PROM(s) or sockets on the CPU board and remove the current monitor PROM** (if necessary) — The figure on the previous page shows the location of the monitor PROM socket(s) and the monitor PROM configuration jumpers on V452 Series boards

❹ **Install the EPROM** — Install the PROM in the appropriate socket. The PROM socket(s) on Synergy boards accept 32-pin PROM devices. The figure below shows the orientation of each PROM after proper installation.

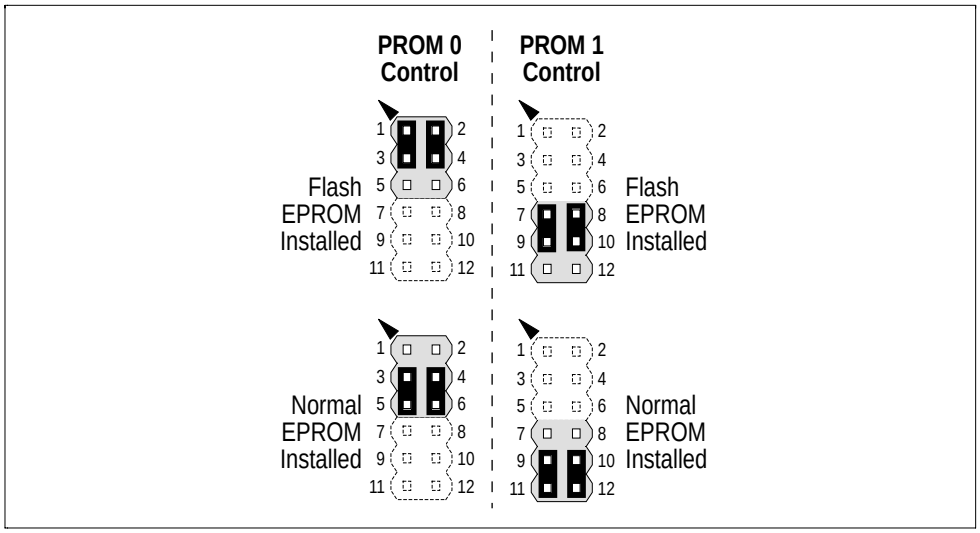


EPROM devices properly installed in sockets

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

❺ **Select the Monitor PROM type** (if required) — V452 Series boards provide two independent 32-pin, 8-bit sockets typically used for system firmware. These two sockets can accept 1 Mbit - 8 Mbit EPROM chips, 512 KB of Flash EPROM or a combination of these devices.

To indicate what type of PROM device is installed in each socket, configure the **JK12** jumper as shown in the figure below:



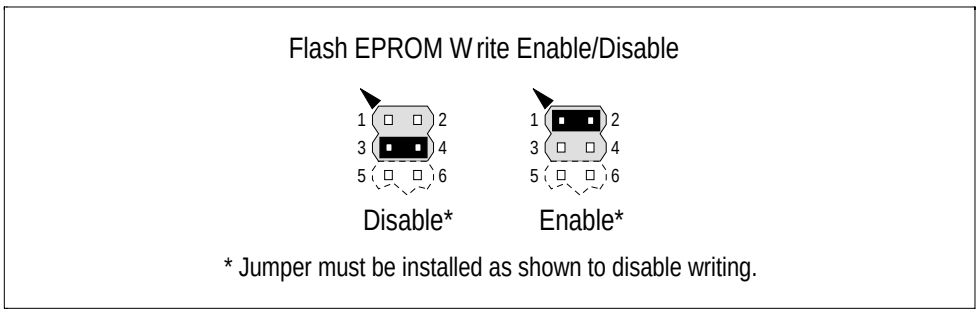
EPROM type jumper settings (JK12)

⚠ One of the shunt configurations shown above must be in place for proper operation of each PROM installed on the board.

Check the configuration of JK12 whenever a new EPROM or Flash PROM is plugged into the board.

❻ **Enable/disable Flash EPROM writes** (if required) — Flash EPROM can be programmed using circuitry on the motherboard, allowing firmware code maintenance and updates to be performed across a network. If Flash EPROM has been installed on the board as described in Step ❺ above, configure the **JK17** jumper as shown in the figure below:

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



Flash EPROM Write enable/disable jumper settings (JK17)

Note

Enabling Write enable function allows Flash EPROM writing to occur but does not actually start the process. For this activity to occur, a series of other software instructions must be executed which apply various power and data signals to the Flash EPROM chip.

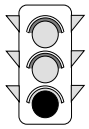
For more information about on-board programming of Flash EPROM devices, see the datasheet for the Flash EPROM devices in the datasheet for the 28F010 Mbit Flash EPROM datasheet in the ***Datasheet Supplement*** supplied with this manual. The programming sequence described in this datasheet can be applied to both the **28F010** and the **28F020** device types. Also refer to the ***EPROM*** chapter in Section 4 and the ***Flash EPROM programming tools*** chapter in Section 6.

If Flash EPROM is installed in either or both of the PROM sockets, one of the shunt configurations shown above must be in place for proper operation. If Flash EPROM is not installed, no jumper is necessary.

To verify the proper operation of the newly installed EPROM, perform the procedure as described in the ***Installing V452 Series CPU boards*** chapter in this section.

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

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



Installing the R452/R453 memory module

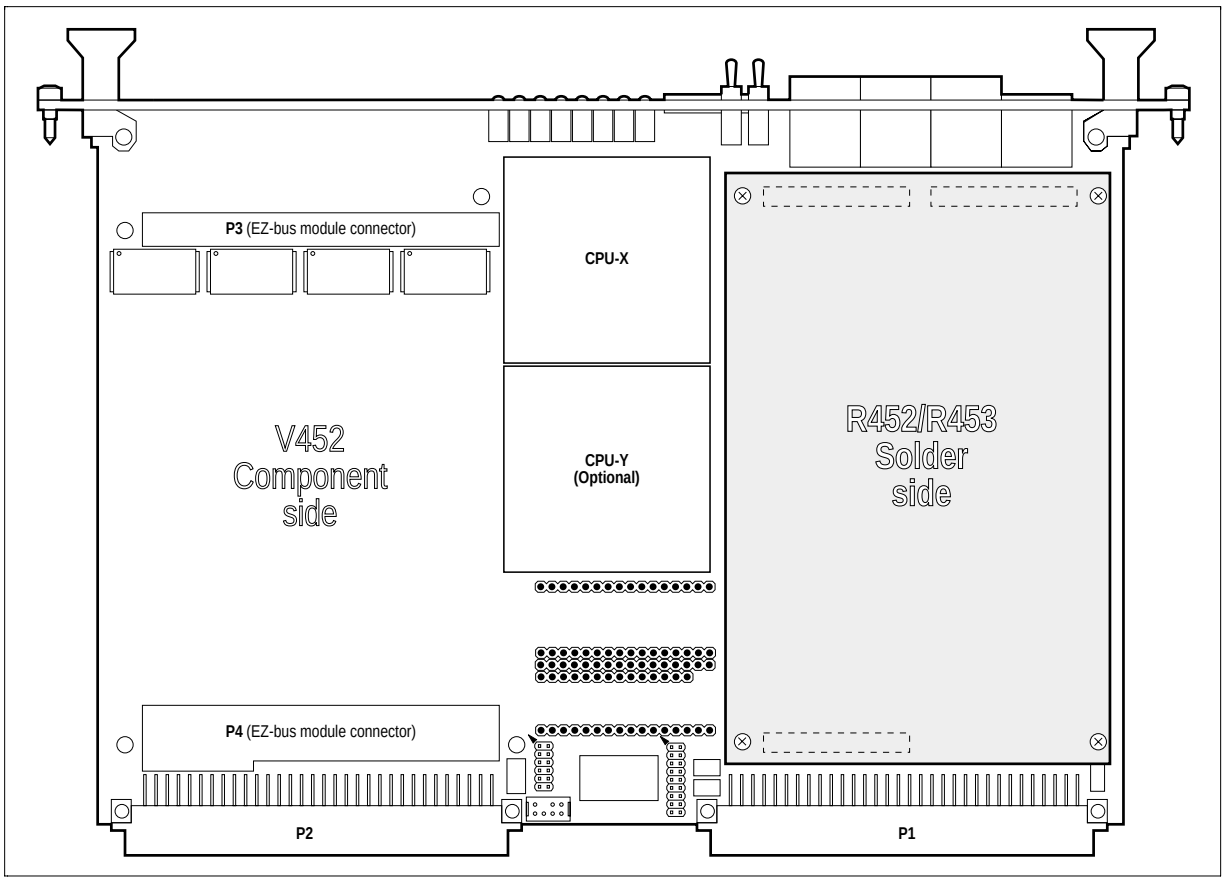
V452 Series boards provide all on-board DRAM on upgradable memory modules. Modules are available with the following amounts of DRAM:

- 4 MB
- 8 MB
- 16 MB
- 32 MB
- 64 MB
- 128 MB
- 256 MB (special factory order)
- 512 MB (special factory order)

Normally, all V452 Series boards ship from the factory with a memory module installed. The modular design of the V452 Series DRAM interface, however, allows for easy DRAM upgrades in the field.

The drawing below shows the location of the R452/R453 memory module on the motherboard.

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

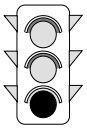


R452/R453 memory module location (top view)

This chapter describes field installation of an R452/R453 memory module.

> **Note** If the desired R452/R453 module is already present on the V452 Series board, proceed to the next chapter in this section.

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



Installing/upgrading the R452/R453 memory module

Perform the following steps to install or upgrade a R452/R453 memory module.

❶ **Verify proper operation of motherboard** (if replacing an existing R452/R453 memory module) — Before attempting to install a new R452/R453 memory module on a working CPU motherboard, consider checking that the motherboard (and any attached EZ-bus modules) are operating properly.

Perform the installation/checkout procedure for the V452 Series board as described later in this section.

❷ **Power-down and remove CPU motherboard from card cage** (if necessary) — Power-down the system and remove the V452 Series CPU board from the card cage.

> ⚠ Synergy CPU motherboards contain static-sensitive devices. Make sure you are properly grounded (by putting on a ground-strap, touching the system power supply, etc.) before removing and handling the board. Use an ESD-protected workstation for module removal and installation work.

❸ **Remove existing R452/R453 memory module from CPU motherboard** (if you are replacing — refer to drawing below for assembly details):

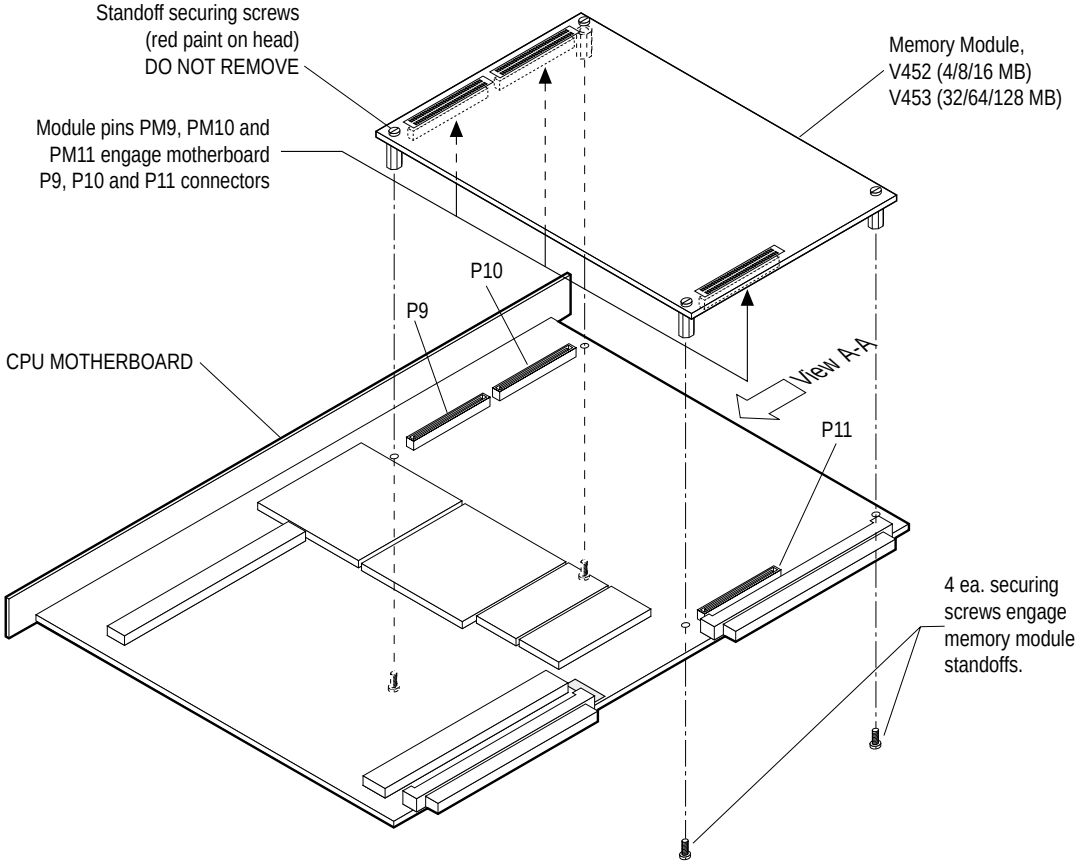
a. Place V452/R45x assembly face-down, that is with large circuit board (motherboard) on top, on a flat surface of an ESD-protected workstation.

b. Remove four M2.5 slot-head screws from rear (solder) side of V452 motherboard (area of P9, P10 and P11).

c. Turn V452/R45x assembly over.

d. Grasp R45x at sides and gently pull up until the connectors come loose.

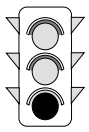
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



R452/R453 module installation

❹ **Install R452/R453 module on motherboard** — Installation of R452/R453 memory module is reverse of removal. Before installing, make sure that connectors are engaged before fully seating module. Refer to drawing on preceding pages for assembly details.

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



Setting up the V452 Series hardware

This chapter lists the hardware configuration decisions and steps that need to be made before installing V452 Series boards.

Normally, boards are ordered and built for a particular application that establishes a configuration that encompasses many if not all of the configuration tasks described in this chapter.

As a result, because many of these set-up decisions are application dependent and/or already set by your ordering specification, this chapter seeks to **describe** the general hardware configurations rather than **prescribe** any particular system arrangement or suggest that the configuration of the board as sent from the factory **needs to be changed**.

Rather, it is intended as a summary/index of the required configuration steps, a *quick start* or *installer's checklist* for those who already have a working knowledge of the issues involved, and an introduction to the special features of the components or architecture on V452 Series boards.



This chapter refers to several operational subjects that in most cases are covered in more detail elsewhere in this manual. If more information on a particular subject is available, the descriptions in this chapter tell where it can be found.

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

Default configuration

The table shown below lists the default hardware configuration for the V452 Series boards before any jumpers, shunts, or custom modifications are made.

Default hardware conditions

Jumpers (presumes no jumper installed)		Description
JK17	— Flash write enable/disable	Disabled*
	— Bus Request Level	Set by software
	— VME Remote Reset	Disabled
	— Round Robin/Priority VME request handling	Priority
	— Auto System Controller	Enabled **
	— EPROM/Flash Boot select	Flash boot

Notes: * Default state is 'disabled' with no jumper installed, but input is floating. To guarantee write protection of flash memory, install jumper in disabled position.

** Board automatically set up as System Controller when installed in Slot 1 of card cage.

Configuration tasks

It may be necessary to change the default conditions listed in the table above by installing shunts on the appropriate jumper block. The required hardware configuration tasks are outlined below and described in more detail in the following pages:

❶ Verify that the board has a monitor EPROM and memory module installed.

❷ Verify/install shunt at JK17 to enable or disable writing to Flash.

❸ Install a shunt at JK17 to set the bus request level (remove shunt for software control of bus request level).

❹ Install a shunt at JK17 to enable VME remote reset (if desired).

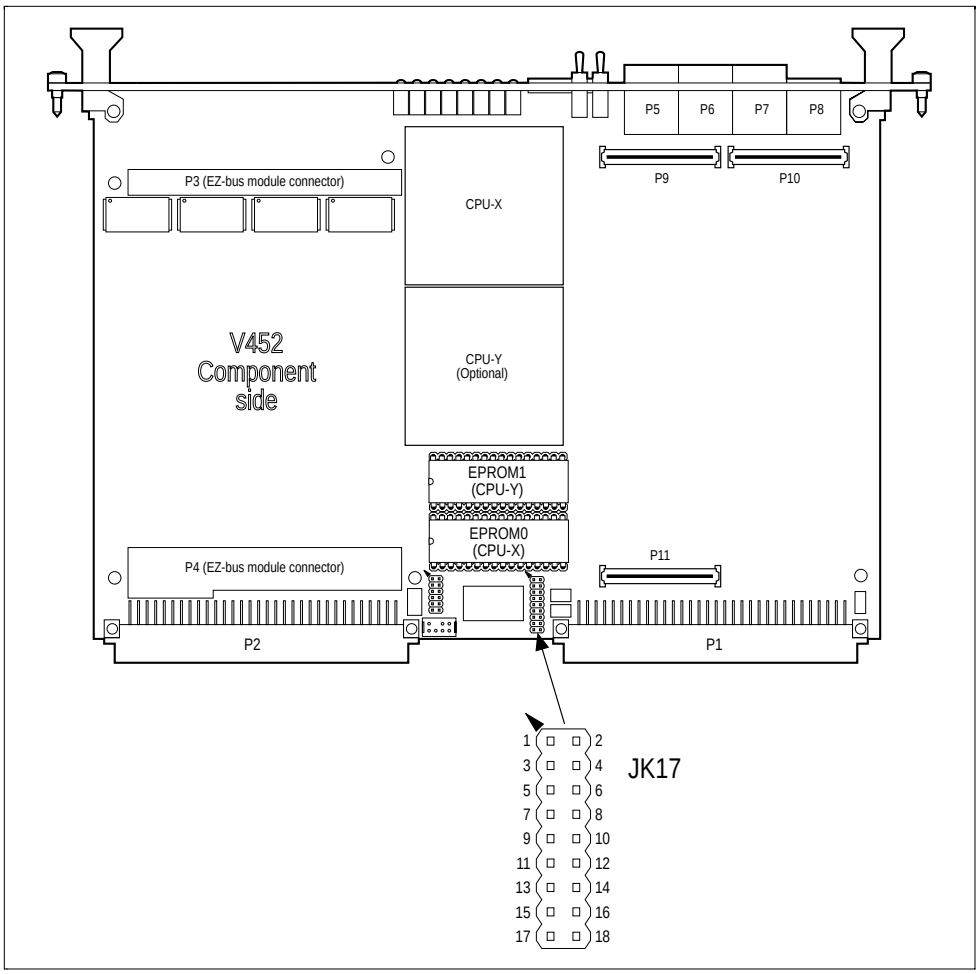
❺ Install shunt at JK17 to prevent the board from being set up as the VME System Controller as required.

❻ Install a shunt at JK17 to select Round Robin interrupt handling (only if the board is to serve as System Controller).

❼ Install a shunt at JK17 to select Flash for booting (remove shunt to boot from EPROM).

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

Several of these configuration tasks involve the **JK17** jumper block. The drawing below shows the location of this jumper block.

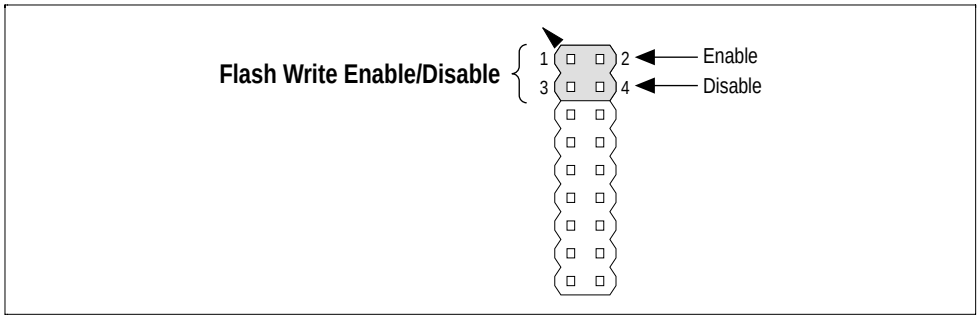


Jumper JK17 location

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

Enable/Disable Flash write

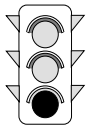
Write protection of Flash memory installed in the V452 Series EPROM sockets is provided by two pair of jumper pins on JK17. To enable Flash write, install a shunt across JK17 pins **1** and **2**. To disable Flash write, install a shunt across JK17 pins **3** and **4**. The figure below shows the jumper pins controlling Flash write.



Jumper settings (JK17) — Flash write enable/disable

Note If no jumper is installed at all in either of the two positions, Flash writing is disabled by default, though input pins (Vpp) are left floating. To guarantee write protection, install a shunt across JK17 pins 3 and 4.

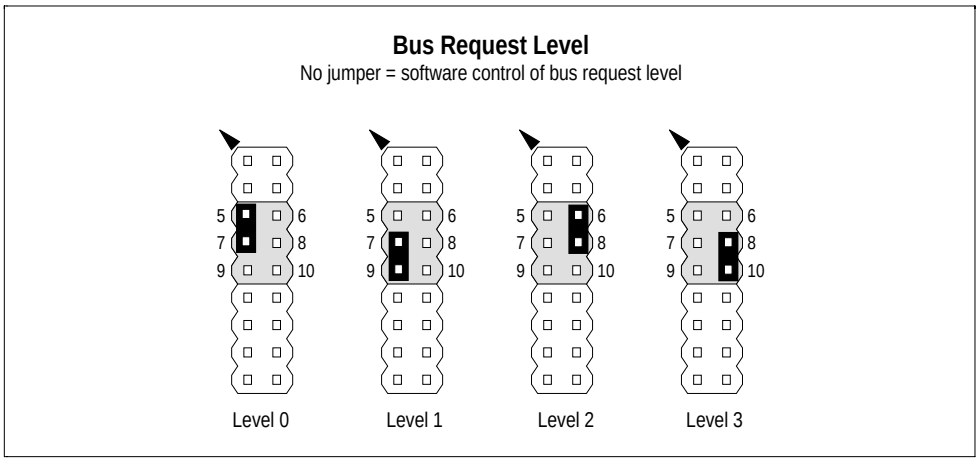
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Set the VMEbus request level

The VME interface on V452 Series boards can request access to the VMEbus at four different priority levels from lowest (level 0) to highest (level 3) priority. The V452 bus request level can be set in either hardware (JK17 jumpers) or software. If no bus request level jumpers are installed, the Ethernet/VMEbus control registers (FE3A 4000-F) take over to set the bus request level.

To set the bus request level in hardware, configure JK17 pins **5-10** for one of the bus request levels as shown in the figure below:



Bus request level jumper settings (JK17)

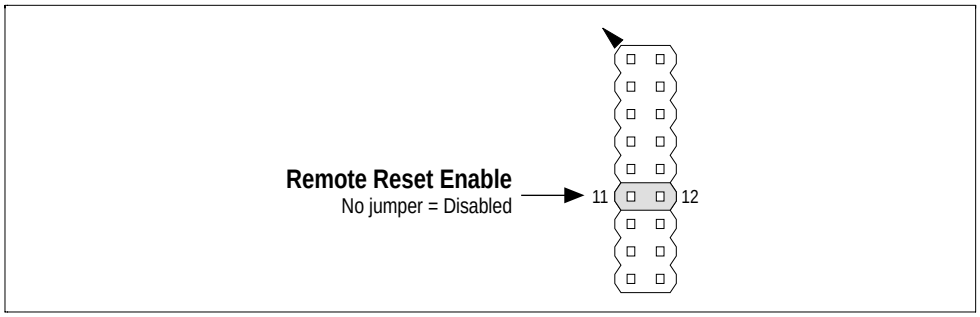


For more information about setting the bus request level in software, see the ***V452 Series internal registers*** chapter in Section 3. For more information about selecting a VMEbus request level, see the ***VME Master interface*** chapter in Section 5.

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

Enable VME Slave remote reset

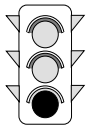
V452 Series boards support remote reset by external VME Masters. To enable VME Slave remote reset, install a shunt across JK17 pins **11** and **12** as shown in the figure below. Remove the shunt to disable this feature (if necessary).



Jumper settings (JK17) — Remote Reset enable

Note For more information about VME remote reset, see the ***Default and reset conditions*** chapter in Section 3 and the ***VME Slave interface*** chapter in Section 5

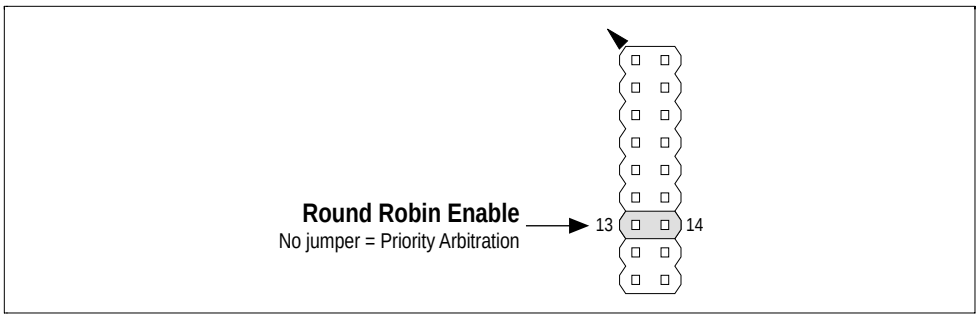
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Select Round Robin or Priority request handling

If acting as the System Controller, V452 Series boards can support either Round Robin or Priority handling of VMEbus requests.

To select **Round Robin** request handling, install a shunt across JK17 pins **13** and **14** as shown in the figure below. To select **Priority** handling remove this shunt (if it is installed).



Jumper settings (JK17) — Round Robin vs. Priority requests



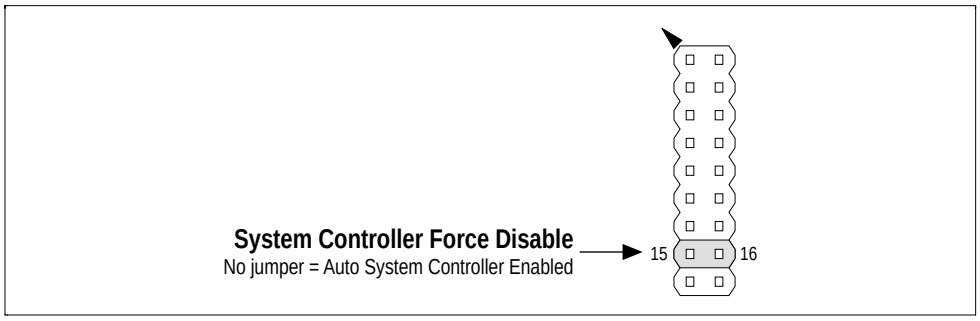
For more information about these VMEbus request methods, see the ***System Controller*** chapter in Section 5.

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

System controller force disable

All V452 Series boards contain circuitry allowing them to automatically serve as the System Controller on the VMEbus when installed in slot 1.

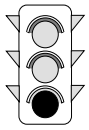
A jumper is provided to force disable of the system controller. Normally, this jumper is not installed. However, in the event of unwanted interactions with the auto-detection circuitry, force the system controller to the disabled state by installing a shunt across JK17 pins **15** and **16** as shown in the figure below.



Jumper settings (JK17) — System Controller force disable

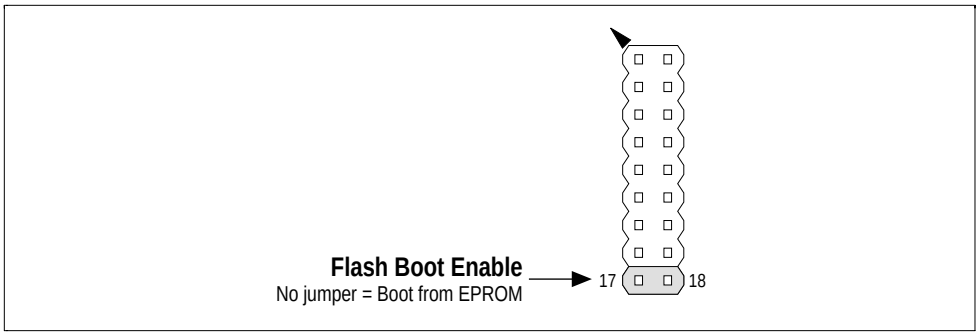
Note For more information about the capabilities and features of the V452 Series System Controller circuit, see the ***System Controller*** chapter in Section 5.

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



Boot select, EPROM or onboard Flash

A jumper is provided to select between booting from EPROM socket devices (i.e., regular EPROM or Flash EPROM) or the onboard Flash memory. As shown in the figure below, install a shunt across JK17 pins **17** and **18** to boot from Flash. To boot from EPROM, remove the shunt from JK17 pins **17** and **18**.

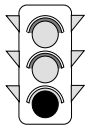


Jumper settings (JK17) — EPROM/Flash boot select

With no jumper installed at JK17 pins 17 and 18, the CPU reads its reset vector from EPROM0 at FE00 0000. When this jumper is installed, the CPU reads its reset vector from the onboard Flash memory at FC00 0000.

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

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



Installing V452 Series CPU boards

This chapter explains how to install an V452 Series board in a VME card cage.

Materials

To complete this procedure, you will need the following materials:

- V452 Series board to be installed,
- VME card cage meeting the minimum system requirements (See the chapter on *Minimum system requirements* earlier in the ***Getting Started*** section.)

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

Procedural steps

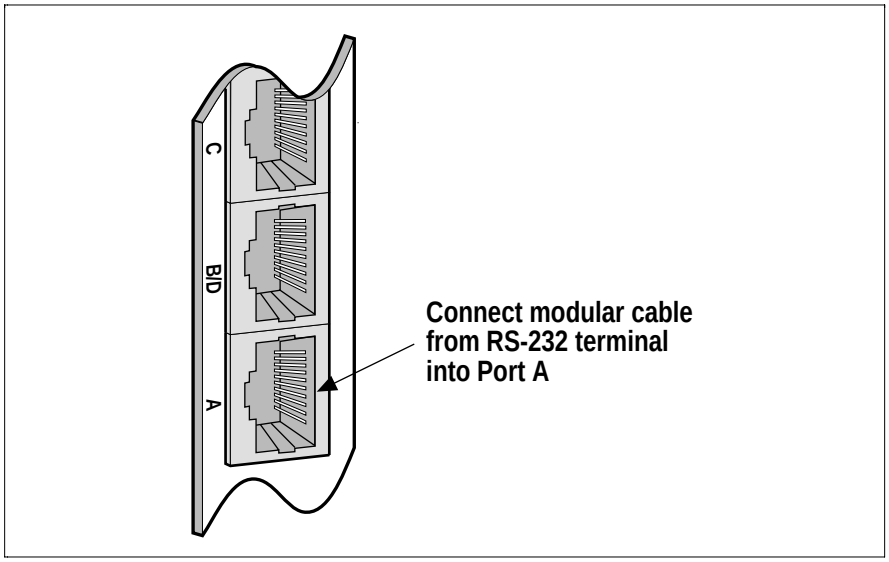
To install the V452 Series boards in a VMEbus card cage, perform the following steps:

❶ Install the board in a VMEbus slot — if the board is to be the System Controller, install it in Slot 1.

> **Note** If you intend to install the board in a slot that has been empty up to this point, be sure to remove the daisy-chain jumpers for that slot from the VMEbus. For more information, refer to the documentation accompanying the system or VMEbus card cage assembly you are using.

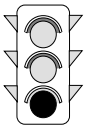
❷ **Attach a modular serial interface cable to Serial I/O connector A on the front panel** — Once the board is installed, plug a modular serial I/O cable into the 10-pin, modular Serial I/O connector A (P5) on the front panel as shown in the figure below.

For more information, see the ***Serial cabling options*** chapter in Appendix A.



Serial I/O port A (front panel)

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



❸ **Connect terminal to I/O cable** — Connect an RS-232/423 or RS-422 compatible terminal to the serial I/O cable. Set the terminal communication configuration to:

- 9600 baud
- receive, no handshake
- transmit, Xon/Xoff
- no parity
- one stop bit

To allow the modular cable to connect to a D-type connector on the terminal, use a Modular/EIA adapter described in the ***Serial cabling options*** chapter in Appendix A.



Note that serial ports A and C can support either RS-232/RS-423 single ended or RS-422A signals to either DTE or DCE devices by using different interface cable and adapter configurations. Serial ports B & D support only RS-232/RS-423. To ensure that the cable/adapter you are using provides the required signals for your terminal or serial device, see the ***Serial cabling options*** chapter in Appendix A.

❹ **Power-on all units** — Power-up the card cage containing the V452 Series board and the video terminal that is connected.

❺ **Verify boot sequence and appearance of start-up banner** — Immediately upon power-up, all of the yellow LEDs on the board's front panel flash momentarily. Then, the green X LED should illuminate as the CPU accesses the boot PROM.

The start-up banner for the monitor PROM/EPROM you have installed on the board should then appear. If the banner appears, the V452 Series is in working condition and ready for user-configuration.

The power-up banners you may see are described below:

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

Power-on banners

The paragraphs below show example power-on banners for the various operating system PROM/EPROMs that are available for V452 Series boards. Note that firmware revisions may result in actual displays that differ from the given examples.

OS-9

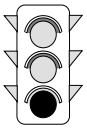
Shown below is an example of the OS-9 PROM's power-up banner. For more information, see the *Using Professional OS-9*, product documentation from Microware Systems Corporation or contact:

OS-9/68000 Operating System
Microware Systems Corporation
1900 N.W. 114th Street.
Des Moines, Iowa 50322
(515) 224-1929



In the below example of the OS-9 startup banner, the user must enter a **g <Enter>** in response to the **RomBug:** prompt in line 11 to see the remainder of the banner. The last six lines of the banner also includes a login routine in which user input is required.

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



```
OS-9/68K System Bootstrap

<Called>
Searching special memory list for symbol modules...

rombug40X_8M
dn: 00000000 00002000  00000000 00000000   00000000 00000001  FFFFE000 0000B1F0
an: FE000AAC FE000500  FE040000 FE040000   0000C200 00004000  0000B200 0000B1F0
pc: FE0009AE  sr:2708 (--SI-7-N---)t:OFF   msp:00039172   usp:00000000   ^isp^
Boot               >43FAFBAC         lea.l Reset(pc),a1
RomBug: g

BOOTING PROCEDURES AVAILABLE ---- <INPUT>

Boot from Viper tape drive ------ <vs>
Boot from SCSI(SCCS) hard drive - <hs>
Boot from ROM ------------------- <ro>
Load Bootfile from ROM ---------- <lr>
Synergy Ram Boot ---------------- <ra>
Boot from BOOTP I82596 ---------- <ie>
Reconfigure the boot system ----- <rc>
Restart the system -------------- <q>

Select a boot method from the above menu: hs

A valid OS-9 bootfile was found.
-t -np
*
*   OS-9
*   Copyright 1984 by Microware Systems Corporation
*
*   The commands in this file are highly system dependent and should
*   be modified by the user.
*
setime -s                     ;* start system clock
December 11, 1996 Wednesday 10:55:27 am
iniz /h0
*chd /h0/SYS
*startisp
*chd /h0
* iniz r0 h0 d0 t1 p1         ;* initialize devices
* load utils                  ;* make some utilities stay in memory
* load bootobjs/dd.r0                ;* get the default device descriptor
* init.ramdisk >/nil >>/nil&  ;* initialize it if its the ram disk
* tsmon /t1 &                 ;* start other terminals
* list sys/motd
ex tsmon /term
1 devices online

OS-9/68K V3.0.2   Synergy Microsystems SV40 - 68040   96/12/11 10:55:32

User name?: super
Password:
Process #06 logged on    96/12/11 10:55:35
Welcome!

Super>
```


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

pSOS/pROBE (pSOS+/68k)

Shown below is an example of the pSOS/pROBE PROM's power-up banner. For more information see the *pSOS+/68k Evaluation Package User's Guide*; Manual Version 1.1; Software Components Group or contact:

Synergy Microsystems, Inc.
(858) 452-0020

OR

Software Components Group, Inc
1731 Technology Drive, Suite #300
San Jose, CA 95110
(408) 437-0700

```
PROBE V3.14 (68040)
COPYRIGHT 1986, SOFTWARE COMPONENTS GROUP INC.
ALL RIGHTS RESERVED

pROBE>
```

pROBE.jr (pROBE/68k)

Shown below is an example of the pROBE.jr PROM's power-up banner. For more information, see *pROBE.jr User Guide*; Synergy Microsystems, Inc., or contact:

Synergy Microsystems, Inc.
(858) 452-0020

OR

Software Components Group, Inc
1731 Technology Drive, Suite #300
San Jose, CA 95110
(408) 437-0700

```
Synergy Microsystems Inc.
 pROBE.jr v3.1.0 DebugMon
 Software Components, Inc
  <board/mod> - mm/dd/yy

pROBE.jr:
```

VxWorks

Shown below is an example of the VxWorks PROM's power-up banner. For more information see the *VxWorks Reference Manual*; Version 5.2; Wind River Systems or contact:

Synergy Microsystems, Inc.
(858) 452-0020

OR

Wind River Systems, Inc.
1351 Ocean Ave.
Emeryville, CA 94608
(800) 545-WIND

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

```
boot device          : ei
processor number     : 0
host name            : host
file name            : /vw/trg/sv462x
inet on ethernet (e) : 197.42.42.225:ffffff00
inet on backplane (b): 198.0.0.1
host inet (h)        : 197.42.42.10
user (u)             : vw
flags (f)            : 0x0

Attaching network interface ei0... done.
Subnet Mask: 0xffffff00
Attaching network interface lo0... done.
Loading... 490048 + 30480 + 24382
Starting at 0x1000...

Target Name: vxTarget
Attaching network interface ei0... done.
Initializing backplane net with anchor at 0x600... done.
Backplane anchor at 0x600... Attaching network interface sm0... done.
Attaching network interface lo0... done.
Loading symbol table from host:/vw/trg/sv462x.sym ...done


 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
      ]]]]]]]]]]]  ]]]]     ]]]]]]]]]]       ]]              ]]]]         (R)
 ]     ]]]]]]]]]  ]]]]]]     ]]]]]]]]       ]]               ]]]]
 ]]     ]]]]]]]  ]]]]]]]]     ]]]]]] ]     ]]                ]]]]
 ]]]     ]]]]] ]    ]]]  ]     ]]]] ]]]   ]]]]]]]]]  ]]]] ]] ]]]]  ]]   ]]]]]
 ]]]]     ]]]  ]]    ]  ]]]     ]] ]]]]] ]]]]]]   ]] ]]]]]]] ]]]] ]]   ]]]]
 ]]]]]     ]  ]]]]     ]]]]]      ]]]]]]]] ]]]]   ]] ]]]]    ]]]]]]]    ]]]]
 ]]]]]]      ]]]]]     ]]]]]]    ]  ]]]]]  ]]]]   ]] ]]]]    ]]]]]]]]    ]]]]
 ]]]]]]]    ]]]]]  ]    ]]]]]]  ]    ]]]   ]]]]   ]] ]]]]    ]]]] ]]]]    ]]]]
 ]]]]]]]]  ]]]]]  ]]]    ]]]]]]]      ]     ]]]]]]]  ]]]]    ]]]]  ]]]] ]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]       Development System
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks version 5.2
 ]]]]]]]]]]]]]]]]]]]]]]]]]]       KERNEL: WIND version 2.4
 ]]]]]]]]]]]]]]]]]]]]]]]]]       Copyright Wind River Systems, Inc., 1984-1995

                               CPU: Synergy SV-462x.  Processor #0
                              0x20000000[0x200000]  BSP version 1.0/4
                             Friday Nov 17, 1995  <15:44:17> GMT

 ->
```


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

SMon

Shown below is an example of the SMon PROM's power-up banner. For more information see the *SMon User Guide* or contact:

Synergy Microsystems, Inc.
9605 Scranton Road, Suite 700
San Diego, CA 92121-1773

```
SMon BOOT ROM V2.65  (Synergy Microsystems - SV400/440/460 X CPU)

Copyright (c) 1992,1993,1994, Synergy Microsystems, Inc.

---------------------------------------------------------------
HARDWARE PARAMETERS:
    This Board's address on the VME bus is 0x02000000
    The serial channel will use a baud rate of 9600
POWERUP TEST(S):
    Powerup test(s) are disabled
    After board is reset, startup code will wait 2 seconds
---------------------------------------------------------------

To change any of this, hit any key within 2 seconds



_________________    _/_/_/    _/       _/
  _____________  _/        _/  _/_/   _/_/
    _________  _/          _/ _/  _/_/     _/_/_/     _/  _/_/
      _______    _/_/_/    _/   _/  _/    _/      _/   _/_/    _/
        _____          _/  _/       _/   _/       _/   _/      _/
          __  _/       _/  _/       _/   _/       _/   _/      _/
              _/_/_/    _/         _/     _/_/_/     _/      _/

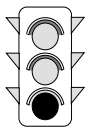

        Synergy V4X0 Monitor & C Compiler    Rev 2.65, 1994/06/28

                     Type "help" for help.


hello world

Aug 12 12:02:26 1994
SMon>
```


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



Setting up the V452 Series software

Upon power-up, V452 Series boards require a set of software instructions to set up several on-board functions. The monitor EPROM that may be provided with the V452 Series board establishes a particular initialization or environment encompassing many if not all of these initialization tasks.

Because many of these set-up decisions are application-dependent and/or already set by your system firmware, this chapter seeks to **describe** the general requirements for system initialization rather than **prescribe** any particular system arrangement or to suggest that any changes need to be made to the software configuration provided by the system software package that may have been supplied with the board.

This chapter describes the general default or pre-initialized state of V452 Series boards following a power cycle or board reset and then goes on to describe the required and/or optional software initialization tasks needed to bring the board up from scratch. It is intended to serve as a summary/index of the required configuration steps, a *quick start* or *programmers checklist* for those who already have a working knowledge of the issues involved, and an introduction to the special features of the components or architecture of V452 Series boards.



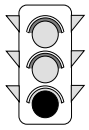
This chapter refers to several operational subjects that in most cases are covered in more detail elsewhere in this manual. If more information on a particular subject is available, the descriptions in this chapter tell where it can be found.

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

Default (pre-initialization) conditions

V452 Series boards present a series of default conditions that exist before software initialization occurs. To provide maximum flexibility, V452 Series boards provide several software initialization registers to set up the board for a specific use. Some of these conditions are unique to the V452 Series architecture/circuitry while others pertain to individual chips. The following table lists these pre-initialization conditions.

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



Default conditions on power-up

Front panel LEDs	
Application LEDs (front panel)	On
Fail LED	On
On-board interrupts	
ABORT switch (Level 7 interrupt)	Enabled (non-maskable)
Status for all other Interrupt sources (Levels 7 - 2)	Disabled
68040/060 functions	
Internal cache contents	Indeterminate
Internal instruction & data caches	Disabled
Memory areas not subject to caching (transparent translation)	None
On-board DRAM	
Content of on-board RAM and RAM parity	Indeterminate
Memory parity checking	Disabled
CPU Watchdog	
CPU Watchdog Halt monitor	Disabled
CPU Watchdog Run monitor	Disabled
Serial ports B & D	Disabled
EPROM1 access by CPU-Y	Enabled
VME Slave interface	
Slave interface status	Disabled
Slave memory write access	Disabled
Slave memory write access protection level	Supervisor-only
Address width	A32
Window size	16 MB
Base Address	0 MB (x000 000)
Lower vs. Upper VME A32/D32 address range	Lower (Top of DRAM to 0x3FFF FFFF)
VME Master interface	
VMEbus release configuration	Release on Request (ROR)
VMEbus request configuration	Not FAIR requests
VMEbus request level configuration (if not set by JK17 jumper)	Level 3
VME interrupts	
VME Interrupter status	Reset
VME interrupt level asserted by the VME interrupter	Level 7
VME interrupt vector asserted by the VME interrupter	Indeterminate
VME Interrupt handler level	None
VME SysRes signal	
VME SysRes (reset) reception respond/ignore	Respond
VME SysFail signal	
VME SysFail interrupt reception (via 2692 DUART)	Disabled
VME SysFail\ signal	Asserted
2692 control registers	
Serial ports B and D	Disabled
VME interrupter level	Level 7
SysFail reception	Off

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

Initialization tasks

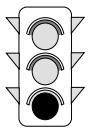
The default conditions listed in the table above can be changed, if required, by executing an initialization program sequence during the system boot process. The same instructions from this sequence can also be used to alter settings *on-the-fly* to produce special environments or temporary effects.

The primary software initialization tasks are:

❶ Set up the 68040/060 caches & data fetching behavior,
❷ Enable the desired on-board and VME interrupts and configure the VME interrupter,
❸ Initialize on-board devices,
❹ Enable serial channels B and D (if desired),
❺ Turn off the application LEDs,
❻ Initialize RAM, clear parity, and enable memory parity protection (if desired),
❼ Set up the VME Master interface (if needed),
❽ Set up and enable the VME Slave interface (if needed),
❾ Perform self test (if desired), release the VME SysFail signal, and turn off the Fail LED,
❿ Set up and enable Watchdog timer (if desired).

The remaining paragraphs in this chapter describe these software initialization tasks in greater detail.

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



Set up the CPU caches & data fetching behavior

The 68040/060 implements data and instruction caching to improve performance.

Flushing the cache contents — A power cycle or processor reset does not invalidate the cache lines. The caches, therefore, must be invalidated early in the initialization sequence to flush out old instructions and data. This is done with the 68040 CINVA assembler instruction which explicitly clears the cache:

```
cinva
nop
```

If you do not have assembler that specifically supports the 68040, the following "hand-assembled" command can also be used:

```
.short 0xF4DB
nop
```



The **nop** instruction is recommended by Motorola to resynchronize the CPU's internal pipelines.

Using the Transparent Translation register — Certain types of data are not compatible with the CPU's caching and processing techniques. The Transparent Translation register in the CPU identifies areas in memory to be "translated" directly or given special handling by the CPU's MMU. The address space assigned to I/O functions is a particularly important area for such special treatment for the following two reasons:

- I/O areas are not compatible with data caching because the data they contain can be changed by an external source without the knowledge of the CPU's cache circuitry.
- Normally, I/O activity must be processed in a strict sequence of steps. Unless otherwise instructed by the MMU, the 68060 will choose the order of read and write instructions to optimize its own operations without regard for the needs of external devices.

As a result, the data (dtt0) in I/O regions (above **0xFE00 0000** on V452 Series boards) must be identified for "serialized" processing and must be inhibited from data caching using the following set of 68040 assembler commands:

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

```
movel       #0xFE01C040, d0
movec       d0, dtt0
```

The hexadecimal expression in the **movel** instruction given above directs the 68060 to perform the following individual actions:

❶ The **FE** byte designates the 16 MB area from 0xFE00 0000 to 0xFEFF FFFF as the base area for transparent translation.

❷ The **01** byte designates an additional 16 MB area above the base area (0xFF00 0000 to 0xFFFF FFFF) for transparent translation.

❸ The **C0** byte enables the transparent translation register to contain the memory range expressed in the previous two bytes. It also enables transparent translation for both user and supervisor modes.

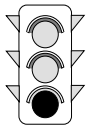
❹ The **40** byte directs the 68060 to inhibit caching, and to serialize the designated memory region.

In lieu of a 68040-compatible assembler, the following sequence can be used which includes a *hand-assembled* command for the **movec** command listed above:

```
movel       #0xFE01C040, d0
.long       0x4E7B0006
```

Configure the 68060 caches (68060 only) — The 68060's cache control register (CACR) is a 32-bit register that controls operation of the instruction and data caches. A MOVEC instruction sets all of the bits in the CACR. A reset clears the CACR which disables both caches. After a reset, therefore, enable caches and other options as required for your application. The table below summarizes the CACR.

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



CACR Bit Assignments

Bit(s)	Function	When 0	When 1
31	Enable Data Cache	Data cache is disabled.	Data cache is enabled.
30	No Allocate Mode (data cache)	Read and write misses will allocate in data cache.	Read and write misses will not allocate in data cache.
29	Enable Store Buffer	All writes to writethrough or cache-inhibited imprecise pages will bypass the store buffer and generate bus cycles directly.	The four-entry FIFO store buffer to the '060 is enabled.
28	Disable CPUSH Invalidation	Each cache line is invalidated as it is pushed. Affects only data cache.	CPUSHed lines remain valid in the cache.
27	1/2 Cache Operation Mode Enable (data cache)	The data cache operates in normal, full-cache mode.	The data cache operates in 1/2 cache mode.
26–24	*reserved*	—	—
23	Enable Branch Cache	The branch cache is disabled and branch cache information is not used in the branch prediction strategy.	The on-chip branch cache is enabled. Branches are cached.
22	Clear All Entries in Branch Cache	No operation is done on the branch cache.	The entire content of the '060 branch cache is invalidated.
21	Clear All User Entries in Branch Cache	No operation is done on the branch cache.	All user-mode entries in the '060 branch cache are invalidated; supervisor mode branch cache entries remain valid.
20–16	*reserved*	—	—
15	Enable Instruction Cache	Instruction cache is disabled.	Instruction cache is enabled.
14	No Allocate Mode (instruction cache)	Accesses that miss in the instruction cache will allocate.	The instruction cache will continue to supply instructions to the processor, but an access that misses will not allocate.
13	1/2 Cache Operation Mode Enable (instruction cache)	The instruction cache operates in normal, full-cache mode.	The instruction cache operates in 1/2 cache mode.
12–0	*reserved*	—	—

Enabling and disabling the caches — To enable the instruction and data caches, use the following **68030**-compatible assembler command:

```
movel  #0x80008000,  d0     |Write to intermediary scratch register
movec  d0,           cacr   |Enable data & instruction caches
```

To disable the instruction and data caches, execute the following assembler command:

```
movel  #0x00000000,  d0     |Write to intermediary scratch register
movec  d0,           cacr   |Disable data & instruction caches
```


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



Refer to the applicable Motorola CPU User's Manual for more information on additional cache configuration options provided by the **CACR** register.

Enable on-board and VME interrupts & configure VME interrupter

In the default condition on V452 Series boards, the CPU is masked from receiving any on-board or VME interrupts (except the non-maskable Level 7 interrupt from the on-board ABORT toggle switch). Muzzling interrupts in this fashion allows the system to initialize on-board devices for reliable operation before attempting to honor interrupt requests that could potentially be spurious.

Interrupts are enabled by writing to specific address locations in the Interrupt Control registers at **0xFE39 0000** to **0xFE39 C00F**. These registers provide a means to enable or disable interrupts from devices on V452 Series boards and all seven VMEbus interrupts.



The V452 Series interrupt architecture provides independent interrupt control for both CPUs (CPU-X and CPU-Y) on dual-CPU models.

For single-CPU models, set interrupts only for **CPU-X**. Setting interrupts for CPU-Y has no effect.

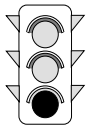
Because there are several on-board and VME interrupt sources, this section of the initialization program normally requires several lines of code. However, the structure of the commands needed to enable each interrupt is the same.

For example, to enable interrupts for the asynchronous serial channels A & B (2692) to CPU-X, execute the following 680x0 assembler instruction:

```
moveb          #0x00, 0xFE390001
```

Enabling Parity and VME SysFail interrupts — V452 Series product series boards provide two layers of control for setting the Level 7 interrupts for SysFail and parity errors providing broad flexibility in interrupt handling.

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



- The Interrupt Control register enables and/or disables all of the maskable level 7 interrupt sources as a group (i.e., Parity error, SysFail, and ACFail as a group). In the default condition, this interrupt is disabled.

  To enable Level 7 interrupts to CPU-X, execute the following 680x0 assembler instruction:

  ```
  moveb       #0x00, 0xFE394001
  ```

  or to enable Level 7 interrupts to CPU-Y, execute the following 680x0 assembler instruction:

  ```
  moveb       #0x00, 0xFE39C001
  ```

- To enable SysFail interrupts, it is also necessary to write a **04** to a register location **0xFE28003B** on 2692 DUART A using the instruction below:

  ```
  moveb       #0x04, 0xFE28003B
  ```

  For more information about setting the SysFail interrupt, see the ***System Controller*** chapter in Section 5.

- To enable parity checking (so that an interrupt is possible), it is also necessary to write **0E** to the **Primary Mode register** at **0xFE38 0003** using the instruction below:

  ```
  moveb       #0x0E, 0xFE380003
  ```

  For more information about using parity checking, see the ***Dynamic RAM*** chapter in Section 4.

  Disabling the maskable Level 7 interrupts using the Interrupt Control register disables SysFail and parity interrupts regardless of the settings for the other registers described above.

For a complete list of all the interrupt sources and Interrupt Control register locations, see the ***Interrupts*** chapter in Section 3 of this manual or the ***V452 Series Quick Reference Card***.

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

Initialize on-board devices

Many of the V452 Series on-board I/O and timer/counter devices require special initialization before they can function reliably as interrupt sources.



Generally, the monitor software provided with your V452 Series board completes device initialization tasks (other than clearing RAM) automatically.

However, if you intend to produce a boot routine from scratch, ensure that each device to be used has been properly initialized prior to enabling its associated interrupts.

Enable serial ports B and D

In the default condition after a reset or power cycle of the V452 Series board, serial ports B and D are disabled by the DUART's OP4 output driving the corresponding line driver's TxD and RTS lines to the high impedance mode (tri-state).

To enable serial port B, execute the following 680x0 assembler instruction:

```
moveb          #0x10, 0xFE28003B     ;enable TxD, RTS
```

To enable serial port D, execute the following 680x0 assembler instruction:

```
moveb          #0x10, 0xFE20003B     ;enable TxD, RTS
```

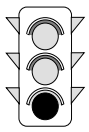
Serial ports A and C on V452 Series boards are hardwired to the enabled state. No tri-state control of line drivers is done on these ports.

Turn off Application LEDs (0-7)

V452 Series boards provide 8 LEDs that can be turned on and off by user applications.

In the default condition, all eight of these LEDs are turned ON. They may be turned OFF by writing the correct value to the Primary Mode register (LEDs 0-3) or Extended Mode register (LEDs 4-7). For example,

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



to turn off LED 0, write **08** to the Primary Mode register at **0xFE38 0003** using the following 680x0 assembler command:

```
moveb          #0x08, 0xFE380003
```



For the proper Mode register values for controlling the other user-programmable LEDs, see the ***V452 Series internal registers*** chapter in Section 3.

Initialize RAM, clear parity, and enable parity checking

Upon power-up or reset, the contents of RAM must be initialized to clear bad parity **before** memory parity checking can be enabled. If it is important to maintain the contents of RAM, (e.g., for debugging) clear bad parity by reading every memory location and then writing it back.

V452 Series boards can provide memory parity checking for all on-board RAM. If bad parity is encountered, a local Level 7 interrupt is asserted.

In the default condition, parity checking is disabled. To enable it, write **0E** to the Primary Mode register at **0xFE38 0003** using the following 680x0 assembler command:

```
moveb          #0x0E, 0xFE380003
```



For more information about parity checking, see the ***Dynamic RAM*** chapter in Section 4.

Set up the VME Master interface

V452 Series boards support FAIR and non-FAIR bus request and either Release on Request (ROR) or Release When Done (RWD) bus release schemes. These options are selected by writing to the Bus Request and Control (0xFE38 C000).



For more information about these bus request and release schemes, see the ***VME Master interface*** chapter in Section 5.

In the default condition, Release on Request (ROR) is selected. To select Release When Done (RWD), execute the following 680x0 assembler command:

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

```
moveb          #0x00, 0xFE38C001
```

In the default condition, the FAIR bus request scheme is **not active**. To direct the V452 Series board to use FAIR bus requests, execute the following 680x0 assembler command:

```
moveb          #0x00, 0xFE38C003
```

In order for the FAIR bus request scheme to be used, **ALL devices on the VMEbus must be set to use it**.

Set up and enable the VME Slave interface

In the default condition, V452 Series boards disable the VME Slave interface, disable Slave access write protection, and provide a default Slave base address, window size, and operating mode.

The following Slave interface characteristics can be set or changed via software:

- A32 (default) or A24 addressing,
- Slave write access protection level as either no write access allowed, Supervisor-only accesses (default) or write accesses allowed by all VME Masters,
- Slave access window size and base address and either the upper or lower VME address range (if applicable).

A summary of the software commands required to set these characteristics is described below:

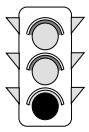
Select A32 vs A24 addressing — V452 Series Slave interface can decode either 32-bit (A32) Extended Addressing or 24-bit (A24) Standard Addressing from a VMEbus Master. The Slave Interface Control register determines the current address mode setting.



V452 Series boards do not support **A16** VME address mode accesses as a VME Slave but are able to generate 16-bit (A16) addresses as VMEbus Masters. For more information see the ***VME Master interface*** chapter in Section 5.

The default setting for V452 Series boards is A32 addressing. To set up a V452 Series board for A24 addressing, execute the following 680x0

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



assembler instruction (or include it in the board's boot/initialization code):

```
moveb        #0x00, 0xFE388001     | A24 addressing
```

To return to A32 addressing, either reset the board or execute the following instruction.

```
moveb        #0x00, 0xFE388000     | A32 addressing
```

Configure the Slave write access protection level — When the Slave interface is enabled, all VMEbus Masters are given complete read access to on-board RAM memory. However, Slave write access is limited to only those Masters whose processors are operating at that moment in Supervisor mode. This default *Supervisor-only* write access condition is one of three possible write access protection levels available for V452 Series boards.

It is also possible to set up the Slave interface to inhibit all write accesses by writing **0D** to the Primary Mode register at **0xFE38 0003**. Execute the following 680x0 assembler command:

```
moveb      #0x0D, 0xFE380003     | Prohibit all Slave write accesses
```

If on the other hand, you wish to grant write access privileges to all Masters, execute the following 680x0 assembler command:

```
moveb      #0x00, 0xFE388003     | Allow all Masters to write
```



For more information about setting the Slave write access memory protection level, see the ***VME Slave interface*** chapter in Section 5.

Select the Lower or upper VME A32/D32 address range — For A32/D32 Slave accesses, V452 Series boards reserve the following two locations where the Slave access window for the board can appear on the VMEbus:

- **Lower A32/D32 range** (default) — up to **3FFF FFFF**. (The lower boundary of this range depends on the amount of on-board memory. For more information see the ***Address map*** chapter in Section 3.
- **Upper A32/D32 range — 8000 0000 - BFFF FFFF**.

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

The default setting for V452 Series boards is to use the lower range for A32/D32 accesses. To set up a V452 Series board to use the upper A32/D32 range, execute the following 680x0 assembler instruction (or include it in the board's boot/initialization code):

```
moveb       #0x00, 0xFE38800F     | Upper VME A32/D32 range
```

To return to using the lower A32/D32 range, either reset the board or execute the following instruction.

```
moveb       #0x00, 0xFE38800E     | Lower VME A32/D32 range
```

Indicate the A32/D32 window size and base address — Once the desired memory range has been selected for VME accesses (as described in the previous paragraphs) locations 0xFE38 8004 — 0xFE38 800D of the Slave Interface Control register are used to select the window size and precise base address for VME Slave accesses.

The desired window size and base address is indicated by performing write accesses to a specific group of locations between 0xFE38 8004 — 0xFE38 800D range. For example, selecting the following setup:

- Local memory size (default)
- 64 MB base address

would require the lower (or default) A32/D32 address range (as described above) and the execution of the following sequence of 680x0 assembler commands:

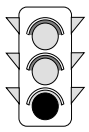
```
moveb    #0x00, 0xFE3A8004    | 64 MB base address (0x04000000)
moveb    #0x00, 0xFE388004    | "
moveb    #0x00, 0xFE388006    | "
moveb    #0x00, 0xFE388008    | "
moveb    #0x00, 0xFE38800A    | "
moveb    #0x00, 0xFE38800D    | "
```



For a complete list of all the meaningful combinations and the window size/base address selected by each, see the ***VME Slave interface*** chapter in Section 5 or the ***V452 Series Quick Reference Card.***

Enable the Slave interface — Once it has been set to the desired arrangement, the Slave interface must be enabled before it will actually be able to operate. To enable it, write **0F** to the Extended Mode register

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



at **0xFE38 4003** by executing the following 680x0 assembler command:

```
moveb          #0x0F, 0xFE384003
```



To disable the Slave interface either reset the board or execute the following instruction:

```
moveb    #0x07, 0xFE384003
```

When slave access is enabled, all of RAM is VME-accessible by default. For all of local DRAM to be VME-accessible, a particular address alignment must be used depending on the amount of onboard DRAM:

- Boards with 16MB or less must be 16MB aligned
- Boards with 32MB must be 32MB aligned
- Boards with 64MB must be 64MB aligned
- Boards with 128MB must be 128MB aligned
- Boards with 256MB must be 256MB aligned
- Boards with 512MB must be 512MB aligned

Perform self-test and suppress VME SysFail

After power cycling or reset, V452 Series boards assert the VME SysFail signal across the VMEbus and continue to assert it until it is disabled. As a final step to software initialization it may be desirable to perform any desired self-test routine(s) to make sure critical systems on the board are ready to go on-line and to indicate whether boot-up has been successful.

To indicate that the board is ready to join the VMEbus, SysFail is suppressed by writing **0C** to the Primary Mode register at **0xFE38 0003** using the following 680x0 assembler command:

```
moveb          #0x0C, 0xFE380003
```

Completing this task also turns off the Fail LED providing a convenient visual check of the V452 Series board's boot completion status.



For more information about the Mode register, see the ***V452 Series internal registers*** chapter in Section 3.

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

Enable bus snooping

The '040/'060 processor features internal data and instruction caches to improve processing speed. To help maintain coherency between data stored in RAM and data undergoing processing in these internal caches, the CPU is able to "snoop" on local RAM write accesses from an external bus.

To maintain cache coherency, the '060 automatically invalidates the cache line whenever snooping is performed and a write cycle hits in the '060s internal cache.

In the default condition, all bus snooping is disabled. To enable bus snooping for the EZ-bus, write **0D** to the Extended Mode register at 0xFE38 4003**.** To enable bus snooping for the VMEbus, write **0E** to the Extended Mode register at 0xFE38 4003. Use the following assembler commands:

```
moveb     #0x0D, 0xFE384003   |Enable EZ-bus snooping
moveb     #0x0E, 0xFE384003   |Enable VMEbus snooping
```



For more information about bus snooping, see the applicable *680x0 CPU* chapter in Section 4.

Enable CPU Watchdog

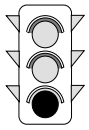
V452 Series boards include a CPU Watchdog circuit (Dallas 1232 or equivalent) that can be configured to monitor the operation of the CPU(s) in either one or both of the following two ways.

- **Halt monitor** — automatically resets the V452 Series board if the CPU(s) halt.
- **Run monitor** — automatically resets the board if not addressed by the CPU(s) every 600 ms.

In the default condition, the Watchdog's Halt monitor is disabled. To enable the Halt monitor, write to location 0xFE38 C005 in the Bus Request and Control register using the following 680x0 assembler command:

```
moveb     #0x00, 0xFE38C005
```


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





For dual-CPU V452 Series models it may be desirable to disable the Halt monitor so that the running CPU can debug and reset the halted CPU.

In the default condition, the Watchdog's Run monitor is disabled. To enable it, write **0C** to the Extended Mode register at 0xFE38 4003 using the following 680x0 assembler command:

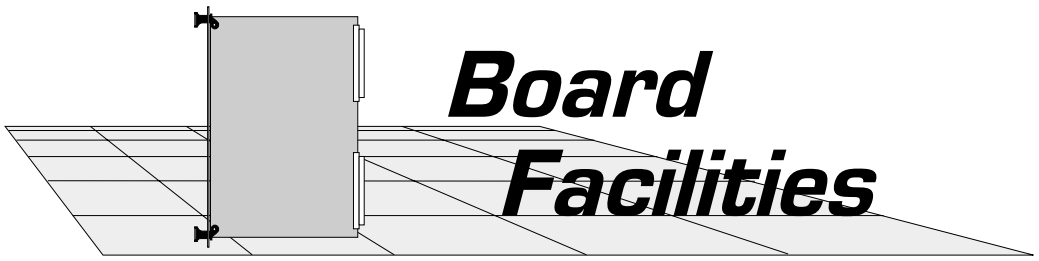
```
moveb          #0x0C, 0xFE384003
```



For more information about the Watchdog, see the ***CPU Watchdog*** chapter in Section 4.

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

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



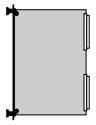
Board Facilities

3

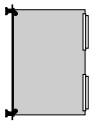
This section contains in-depth information about common, board-level facilities and architecture for **V452 Series** boards.

- Address map
- Interrupts
- Jumpers, switches, LEDs & fuses
- V452 Series internal registers
- Default and reset conditions

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



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



Address map

This chapter lists the memory locations assigned to components, peripherals, and registers for the V452 Series board. V452 Series boards provide 4, 8, 16, 32, 64, or 128 MB of zero or one wait-state EDO DRAM on a plug-in memory module. Higher capacity memory modules, 256MB and 512MB, are available as a special factory order option.

The locations assigned to specific components, peripherals and registers are the same for all boards regardless of their on-board memory size. However, the total memory present on the board determines the location of boundary between on-board memory and off-board accesses.

Access behavior

Not all models of the V452 Series contain all of the components or implement all of the functions listed on the map. Reading or writing to a vacant on-board location yields meaningless data or does nothing. Performing either of these actions **does not** cause any sort of bus error or exception.

A portion of the address map is assigned to the Synergy EZ-bus which provides an interface to one or two optional daughter modules. Each daughter module governs the behavior of all accesses to it across the EZ-bus. Attempting to access a non-existent daughter module causes a bus error.

A large portion of the address map is assigned to the VMEbus. This region is further divided as described in this chapter. As with the EZ-bus, the specific device at each location governs access behavior and access to non-existent locations causes a bus error.

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

Address map

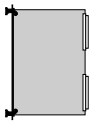
The table below lists the address map as seen by the on-board CPU or a 32-bit daughterboard Master that has been installed on the board. A 24-bit daughterboard Master can access the on-board DRAM directly. The remaining locations are not accessible. A VME Master can access only on-board DRAM through an access window as described in the *VME Slave interface* chapter in Section 5.

Note The following table represents the **standard** V452 Series address map. Other maps can be special-ordered or created for your application if necessary. Contact Synergy Customer Service for more information.

V452 Series address map

Address range	Address/data width Off-board	Address/data width On-board	Device / address space description
0000 0000 - 003F FFFF		D32	**4MB** On-board DRAM
003F FE00 - 003F FEFF		D32	Remote reset register (via VME Slave Interface)
003F FF00 - 003F FF7F		D32	Mailbox write area to CPU-X (all models) *
003F FF80 - 003F FFFF		D32	Mailbox write area to CPU-Y (dual-processor) *
0040 0000 - 7BFF FFFF	A32/D32		Lower VME A32/D32 address range
0000 0000 - 007F FFFF		D32	**8MB** On-board DRAM
007F FE00 - 007F FEFF		D32	Remote reset register (via VME Slave Interface)
007F FF00 - 007F FF7F		D32	Mailbox write area to CPU-X (all models) *
007F FF80 - 007F FFFF		D32	Mailbox write area to CPU-Y (dual-processor) *
0080 0000 - 7BFF FFFF	A32/D32		Lower VME A32/D32 address range
0000 0000 - 00FF FFFF		D32	**16MB** On-board DRAM
00FF FE00 - 00FF FEFF		D32	Remote reset register (via VME Slave Interface)
00FF FF00 - 00FF FF7F		D32	Mailbox write area to CPU-X (all models) *
00FF FF80 - 00FF FFFF		D32	Mailbox write area to CPU-Y (dual-processor) *
0100 0000 - 7BFF FFFF	A32/D32		Lower VME A32/D32 address range
0000 0000 - 01FF FFFF		D32	**32MB** On-board DRAM
01FF FE00 - 01FF FEFF		D32	Remote reset register (via VME Slave Interface)
01FF FF00 - 01FF FF7F		D32	Mailbox write area to CPU-X (all models) *
01FF FF80 - 01FF FFFF		D32	Mailbox write area to CPU-Y (dual-processor) *
0200 0000 - 7BFF FFFF	A32/D32		Lower VME A32/D32 address range
0000 0000 - 03FF FFFF		D32	**64MB** On-board DRAM
03FF FE00 - 03FF FEFF		D32	Remote reset register (via VME Slave Interface)
03FF FF00 - 03FF FF7F		D32	Mailbox write area to CPU-X (all models) *
03FF FF80 - 03FF FFFF		D32	Mailbox write area to CPU-Y (dual-processor) *
0400 0000 - 7BFF FFFF	A32/D32		Lower VME A32/D32 address range

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



V452 Series address map (cont.)

Address range	Address/data width		Device / address space description
	Off-board	On-board	
0000 0000 - 07FF FFFF		D32	**128MB** On-board DRAM
07FF FE00 - 07FF FEFF		D32	Remote reset register (via VME Slave Interface)
07FF FF00 - 07FF FF7F		D32	Mailbox write area to CPU-X (all models) *
07FF FF80 - 07FF FFFF		D32	Mailbox write area to CPU-Y (dual-processor) *
0800 0000 - 7BFF FFFF	A32/D32		Lower VME A32/D32 address range
0000 0000 - 0FFF FFFF		D32	**256MB** On-board DRAM
0FFF FE00 - 0FFF FEFF		D32	Remote reset register (via VME Slave Interface)
0FFF FF00 - 0FFF FF7F		D32	Mailbox write area to CPU-X (all models) *
0FFF FF80 - 0FFF FFFF		D32	Mailbox write area to CPU-Y (dual-processor) *
1000 0000 - 7BFF FFFF	A32/D32		Lower VME A32/D32 address range
0000 0000 - 1FFF FFFF		D32	**512MB** On-board DRAM
1FFF FE00 - 1FFF FEFF		D32	Remote reset register (via VME Slave Interface)
1FFF FF00 - 1FFF FF7F		D32	Mailbox write area to CPU-X (all models) *
1FFF FF80 - 1FFF FFFF		D32	Mailbox write area to CPU-Y (dual-processor) *
2000 0000 - 7BFF FFFF	A32/D32		Lower VME A32/D32 address range
7C00 0000 - 7CFF FFFF	A24/D32		VME A24 / D32 — Standard address space
7D00 0000 - 7DFF FFFF	A24/D16		VME A24 / D16 — Standard address space
7E00 0000 - BFFF FFFF	A32/D32		Upper VME A32/D32 address range
C000 0000 - DFFF FFFF			EZ-bus module A (See user guide for module)
E000 0000 - FBFF FFFF			EZ-bus module B (See user guide for module)
FC00 0000 - FC7F FFFF		D32	Onboard Flash space
FD00 0000 - FDFF FFFF		D8	EPROM1/Flash Memory Module†
FE00 0000 - FE0F FFFF		D8	EPROM0
FE10 0000 - FE10 07F7		D8	Non-volatile Static RAM (2KB default)
FE1F E000 - FE1F FFF7		D8	Non-volatile Static RAM (8KB extended)
FE10 07F8 - FE10 07FF		D8	Clock/calendar (default)
FE1F FFF8 - FE1F FFFF		D8	Clock/calendar (extended)
FE20 0003 - FE20 003F		D8	Serial interface (channels C & D) — Timer 1
FE28 0003 - FE28 003F		D8	Serial interface (channels A & B) — Timer 0
FE2A 0003 - FE2A 003F		D8	16-bit Counters (82C54)
FE30 0000		D8	VME Interrupt vector register (write access)
FE30 0000		D8	CPU mailbox read area — 64x4 FIFO (read access)**
FE38 0002		D8	Status register (read access)
FE38 0003		D8	ID register — 8-bit front-panel switch (read access)
FE38 0003		D8	Primary Mode register (write access)
FE38 4003		D8	Extended Mode register (write access)
FE38 8000 - FE38 800F		D8	Slave Interface Control register (write access)
FE38 C000 - FE38 C00F		D8	Primary Control register
FE39 0001		D32/D8	Slot ID register (read access)
FE39 0003		D32/D8	CPU/Board type register (read access)
FE39 4003		D32/D8	Option register (read access)
FE39 8003		D32/D8	Board ECO register (read access)
FE39 C003		D32/D8	Board revision register (read access)

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

V452 Series address map (cont.)

FE39 0000 - FE39 000F		D8	Interrupt Control register #1 (write access)
FE39 4000 - FE39 400F		D8	Interrupt Control register #2 (write access)
FE39 8000 - FE39 800F		D8	Interrupt Control register #3 (write access)
FE39 C000 - FE39 C00F		D8	Interrupt Control register #4 (write access)
FE3A 0000 - FE3A 000F		D8	Extended Control register (write access)
FE3A 4000 - FE3A 400F		D8	Ethernet/VMEbus Control register (write access)
FE3B 8000 - FE3B 8003		D32	Ethernet Command Port (write access)
FE3B C000 - FE3B C003		D32	Ethernet Channel Attention (write access)
FE40 0000 - FE4F FFFF		D8	EPROM1 (old space)†
FE60 0000 -		D32	32-bit Master BLT register (BLT32)
FE60 0020 -		D32	64-bit Master BLT register (BLT64)
FE80 0000 - FEBF FFFF			EZ-bus module A (See user guide for module)
FEC0 0000 - FEFF FFFF			EZ-bus module B (See user guide for module)
FF00 0000 - FF00 FFFF	A16/D32		VME A16/D32 short address space
FF80 0000 - FF80 FFFF	A16/D16		VME A16/D16 short address space

Notes: * The two Mailbox write areas occupy the top 256 bytes of on-board DRAM.

** For dual-processor V452 Series models, each CPU accesses its own non-steerable Mailbox FIFO at this same address.

† EPROM1 normally resides at FD000000–FDFFFFFF. For backward compatibility with early revisions of V440 and V460 SBCs, however, the old EPROM1 space at FE400000–FE4FFFFF is available for EPROM1 if necessary. Refer to the ***Flash memory module*** chapter in Section 4 for more information.

Address and data size

The standard address map shows the VMEbus address/data as:

Amm/Dnn

where

mm is the number of address bits valid on the bus and

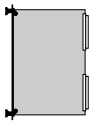
nn is the maximum data transfer size in bits.

The address size controls the address modifier bits placed on the bus, while the data size controls the data strobes and long-word signals onto the bus. Refer to the *VMEbus Specification* for more detailed information.

VME Slave address map

The V452 Series board's VME Slave address map can be found in the ***VME Slave interface*** chapter in Section 5 and also on the ***V452 Series Quick Reference Card***.

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



Interrupts

V452 Series boards feature a VMEbus interrupter, a VME interrupt handler, on-board interrupt handler and a software-driven Interrupt Control register for enabling and disabling the reception of all on-board and VME interrupts. This chapter describes all V452 Series interrupt operations.



For complete lists and descriptions of the exception vectors for the V452 Series board and the processor, see the applicable (68040 or 68060) *CPU* chapter in Section 4 or the ***V452 Series Quick Reference Card***.

Interrupt vectors and levels

The table on the following page lists the standard interrupts and vectors for V452 Series boards and in the respective order of priority level from highest to lowest.



It is possible to assign the same interrupt priority to both an **on-board** and a VMEbus interrupt. In this case the **on-board** interrupt has priority over the VMEbus interrupt.

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

V452 Series interrupt levels and vectors

Interrupt Source		Level	Vector	Description
On-board	*VMEbus*			(see *Chapter* in manual)
ABORT pushbutton (non-maskable)		7A	0x7C	*Jumpers, switches, LEDs & fuses; Dynamic RAM*
Parity error, SysFail, ACFail (maskable)				
	VMEbus Int. 7	7B	[1]	
Timers A & B [2692]		6A	0x78[2]	*Timers & counters*
	VMEbus Int. 6	6B	[1]	
Serial interface — Ports A -D [2692]		5A	0x74[2]	*Asynchronous serial interface*
	VMEbus Int. 5	5B	[1]	
EZ-bus interface — Modules A & B, Ethernet 10Base-T		4A	0x70[3]	*EZ-bus interface*
	VMEbus Int. 4	4B	[1]	
CPU Mailbox (0-CPU-X; 1-CPU -Y)		3A	0x6C[2]	*CPU mailbox*
	VMEbus Int. 3	3B	[1]	
Counters (0-CPU-X; 1-CPU -Y) [82C54]		2A	0x68[2]	*Timers & counters*
	VMEbus Int. 2	2B	[1]	
	VMEbus Int. 1	1	[1]	

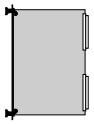
Notes: [1] Interrupt vector is provided by the VMEbus board asserting the interrupt.
[2] Uses 68040 Autovector in standard configuration
[3] Daughter board dependent - may request 68040 Autovector or may provide its own vector. Vector number shown is the level four Autovector.

The interrupt levels and vectors listed above are provided as standard for the V452. The logic establishing these assignments is contained in one of the in-system-programmable (ISP) high density programmable logic devices on the V452 board. It is possible to special-order boards with different programming for a customized set of interrupt levels. It can also be reprogrammed later by the factory or even by a customer in the field with a Synergy-supplied programming kit.



For more information about the programming kit and programming the custom interrupt vector and/or level configurations, contact Synergy customer service.

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



Enabling/disabling interrupts

V452 Series boards provide control over interrupt handling by enabling and/or disabling reception of interrupts by the on-board CPU(s) via a set of 4 write-only registers located within the following address ranges:

Interrupt Control Reg. #1: **0xFE39 0000 – 0xFE39 000F**

Interrupt Control Reg. #2: **0xFE39 4000 – 0xFE39 400F**

Interrupt Control Reg. #3: **0xFE39 8000 – 0xFE39 800F**

Interrupt Control Reg. #4: **0xFE39 C000 – 0xFE39 C00F**

When combined, these registers provide a separate address location to enable and also to disable reception of interrupts from each on-board source and all seven VMEbus interrupt levels. The table below lists the function of all the address locations in the Interrupt Control registers.

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

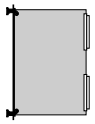
Address locations — Interrupt Control register

Interrupt	Disable (default)	Enable
Serial Port A&B to CPU-X Interrupt (2692)	FE39 0000	FE39 0001
Serial Port C&D to CPU-X Interrupt (2692)	FE39 0002	FE39 0003
Timer A to CPU-X Interrupt (2692)	FE39 0004	FE39 0005
Timer C to CPU-X Interrupt (2692)	FE39 0006	FE39 0007
Daughter module A to CPU-X	FE39 0008	FE39 0009
Daughter module B to CPU-X	FE39 000A	FE39 000B
Mailbox to CPU-X	FE39 000C	FE39 000D
Counter to CPU-X (82C54)	FE39 000E	FE39 000F
Level 7 interrupts (parity, SysFail, ACFail) to CPU-X	FE39 4000	FE39 4001
VME Level 1 to CPU-X	FE39 4002	FE39 4003
VME Level 2 to CPU-X	FE39 4004	FE39 4005
VME Level 3 to CPU-X	FE39 4006	FE39 4007
VME Level 4 to CPU-X	FE39 4008	FE39 4009
VME Level 5 to CPU-X	FE39 400A	FE39 400B
VME Level 6 to CPU-X	FE39 400C	FE39 400D
VME Level 7 to CPU-X	FE39 400E	FE39 400F
Serial Port A&B to CPU-Y Interrupt (2692)	FE39 8000	FE39 8001
Serial Port C&D to CPU-Y Interrupt (2692)	FE39 8002	FE39 8003
Timer A to CPU-Y Interrupt (2692)	FE39 8004	FE39 8005
Timer C to CPU-Y Interrupt (2692)	FE39 8006	FE39 8007
Daughter module A to CPU-Y	FE39 8008	FE39 8009
Daughter module B to CPU-Y	FE39 800A	FE39 800B
Mailbox to CPU-Y	FE39 800C	FE39 800D
Counter to CPU-Y (82C54)	FE39 800E	FE39 800F
Level 7 interrupts (parity, SysFail, ACFail) to CPU-Y	FE39 C000	FE39 C001
VME Level 1 to CPU-Y	FE39 C002	FE39 C003
VME Level 2 to CPU-Y	FE39 C004	FE39 C005
VME Level 3 to CPU-Y	FE39 C006	FE39 C007
VME Level 4 to CPU-Y	FE39 C008	FE39 C009
VME Level 5 to CPU-Y	FE39 C00A	FE39 C00B
VME Level 6 to CPU-Y	FE39 C00C	FE39 C00D
VME Level 7 to CPU-Y	FE39 C00E	FE39 C00F

After power cycling or a board reset, all interrupts revert to the default disabled condition (with the exception of the ABORT pushbutton Level 7 interrupt which is non-maskable) and remain in this condition until initialized by the system boot software/firmware.

Enabling an interrupt involves performing a bytewise write access to the appropriate location in the Interrupt Control register.

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



For example, to enable interrupts for the asynchronous serial channels A & B (2692) to CPU-X, execute the following 68000 instruction:

```
moveb          #0x00, 0xFE390001
```

To disable the same asynchronous serial channels A & B (2692) as an interrupt source to CPU-X, reset the board or execute the following 68000 instruction:

```
moveb          #0x00, 0xFE390000
```



The interrupt control circuitry is designed to decode only the address portion of instructions for the Interrupt Control register. The data value expressed in the instruction does not matter.

Interrupts on single vs. dual processor boards

V452 Series models featuring single verses dual processors require special consideration when using interrupts.

Single processor models

When using a single processor V452 Series model, inadvertently setting interrupts to CPU-Y has no effect on operations in any way.

Dual processor models

The V452 Series interrupt architecture provides independent interrupt control for both processors (CPU-X and CPU-Y) on dual-CPU V452 Series models. When implementing an interrupt structure for dual processors, it is important to observe the difference between the V452 Series steerable and non-steerable interrupt sources:

- **Steerable interrupt sources** — Because the following interrupt sources are not associated with a particular CPU, each of them can be used by either CPU. However, if active for one CPU, the same source should not be used for the other CPU at the same time:
  - Maskable Level 7 interrupts
  - Serial ports A&B
  - Serial ports C&D

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

- Timer A (2692)
- Timer B (2692)
- EZ-bus module A
- EZ-bus module B
- VME interrupts 1 - 7

- **Non-steerable interrupt sources** — The following interrupt sources have separate interrupt circuitry that is dedicated to a specific CPU. As a result, these interrupt sources can be used by **either** or **both** CPUs at the same time:
  - CPU Mailbox
  - Counter0 CPU-X (82C54)
  - Counter1 CPU-Y (82C54)

On-board interrupt sources

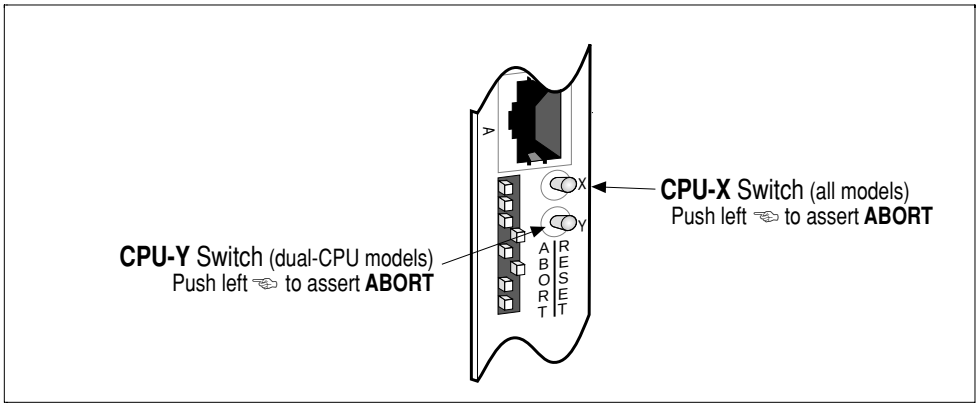
The paragraphs below describe the on-board devices and processes that can generate interrupts:

> **Note** The interrupt levels and vectors described in the paragraphs below refer to the standard V452 Series interrupt configuration. For information on custom interrupt assignments, contact Synergy.

Non-maskable Level 7 interrupt source (ABORT)

The X ABORT switch asserts a non-maskable Level 7 interrupt to CPU-X (or the only CPU on a single-processor board) and the Y ABORT witch asserts a non-maskable Level 7 interrupt to CPU-Y on dual-processor boards as shown in the figure below. Pushing both switches to the left at once asserts a non-maskable Level 7 interrupt to both CPUs.

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



ABORT switch operation

The ABORT toggle switch is a non-maskable interrupt source which means that it is always enabled and cannot be disabled.

> Note
> All of the remaining on-board interrupt sources described in the paragraphs to follow default to the **disabled** state after a power cycle or reset of the board and must be enabled by system software before use.

Maskable Level 7 interrupt sources

Enabling Maskable Level 7 interrupt sources makes it possible for parity errors and reception of the VME SysFail or ACFail signals to cause a Level 7 interrupt. The V452 Series maskable Level 7 interrupt sources are **steerable** interrupt sources. On dual-processor V452 Series boards this means that both maskable Level 7 interrupts can be configured to interrupt either CPU.

Enabling the maskable Level 7 interrupt enables all three of these sources as a group. However, both the parity and SysFail sources also have individual enable control bits allowing them to be turned off as an interrupt source individually while the other active sources in the group remain in force.

All three of these sources in this group have readable status bits in the V452 Status register (0xFE38 0002) as listed in the table below. These bits allow the Level 7 interrupt routine to determine the cause of the interrupt and take the appropriate action.

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

Status register bits for Interrupt Level 7 sources

Bit	When 0	When 1
0	No parity error	**Parity error**
1	**ABORT/Level 7 int. requested**	ABORT/int. not requested
2	**SysFail\ present on VMEbus**	SysFail\ not present
3	**ACFail\ present on VMEbus**	VME ACFail\ not present

The paragraphs below describe each of the Level 7 interrupt sources in greater detail.

- **Parity error** — V452 Series boards provide parity checking for all on-board DRAM which can be either enabled or disabled by writing an appropriate value to the Primary Mode register. The V452 Series interrupt circuitry asserts a Level 7 interrupt request whenever the following three conditions are met:
  - parity checking is enabled,
  - the miscellaneous Level 7 interrupt source is enabled (to CPU-X and/or CPU-Y),
  - a parity error is encountered.

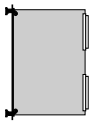
  Note: If desired, it is possible to disable parity checking by writing to the Primary Mode register (at 0xFE38 0003). For more information, see the ***Dynamic RAM*** chapter in Section 4.

- **Remote VME SysFail** — If interrupts from the maskable Level 7 sources have been enabled and the SysFail interrupt has been enabled (by writing to a register on the 2692 DUART), V452 Series boards request a Level 7 interrupt whenever the VME SysFail signal is asserted by another device on the VMEbus. This condition persists until the device in question stops asserting SysFail.

  Note: For more information about enabling the VME SysFail interrupt, see the *System Controller* chapter in Section 5.

- **ACFail signal** — If interrupts from the maskable Level 7 sources have been enabled, V452 Series boards request a Level 7 interrupt whenever the ACFail signal is detected. The ACFail signal warns of the loss of AC power to the system power supply before individual boards on the VMEbus actually lose their DC power. The length of the interval between the loss of the system's AC power and the loss of DC power on an individual

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



board is dependent on many factors including the capacitive characteristics of the system power supply and the total power load of the boards in the system.

This signal can be used to trigger an interrupt routine which can in turn direct internal circuitry to complete an orderly shut-down before the loss of DC power.

Timers

V452 Series boards provide two independent timer interrupt sources (one from each of the two 2692 UARTs). Timers A and C are **steerable** interrupt sources. On dual-CPU boards, this means that each timer interrupt source can be configured for use with either CPU.

These interrupt sources are provided to allow systems programmers to use different interrupt vectors for the 2692 timers (system tick clock) verses the 2692's console input/output functions.

Although a timer interrupt is available via the 2692's internal interrupt mechanism, use of this mechanism is often impractical because it must deal with both the clock and serial I/O interrupts. By providing a separate timer interrupt, the V452 board's interrupt circuitry makes it much easier for system programmers to use timer and serial port interrupts together.



On single-processor boards only one timer should be used as an interrupt source at the same interrupt level at the same time. The other timer can be used for timing functions, but should not be used as interrupt source.

The timer interrupt from each chip is asserted by the **OP3** programmable output pin which can be programmed to generate a square wave whose period is the same as the timer. The 2692 asserts the timer interrupt whenever OP3 goes low.

Timers interrupts remain in force until cleared via software. To clear a timer interrupt, disable the timer as an interrupt source and then re-enable it. For example an interrupt from Timer A to CPU-X could be cleared by executing the following 68000 assembler commands:

```
moveb    #0x00, 0xFE390004   |Disable timer int. source
moveb    #0x00, 0xFE390005   |Re-enable timer int. source
```


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

For more information, see the ***Timers & counters*** chapter in Section 4 or refer to the manufacturer's data sheet that is included for the 2692 DUART. For a code example, see the ***Timer code example*** and ***2692 DUART code example*** chapters in Section 6.

Asynchronous serial interface (2692)

V452 Series boards provide two asynchronous serial interface sources for the four asynchronous serial ports on the front panel as shown in the figure below (one source each from the two on-board 2692 UARTs):

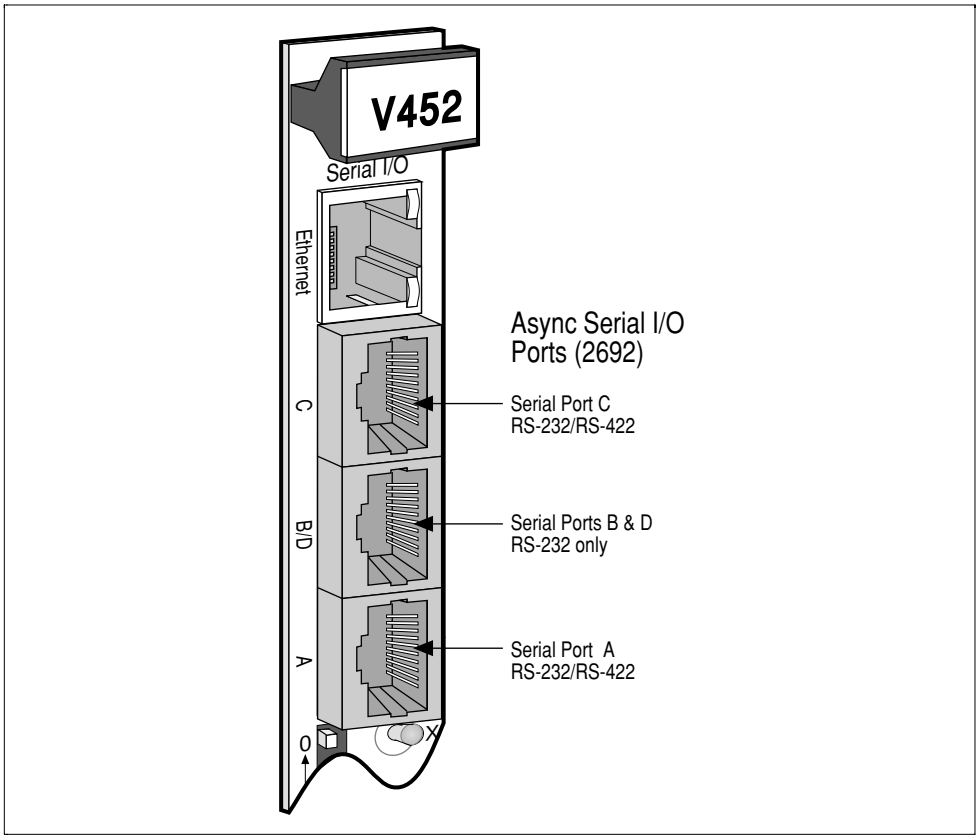
- **Serial interrupt source A** carries interrupt signals corresponding to the asynchronous serial ports A (P5) and B(P6).
- **Serial interrupt source B** carries interrupt signals corresponding to the asynchronous serial ports C (P7) and D (P8).

Serial interrupt sources A and B and **steerable** interrupt sources. On dual-processor V452 Series boards this means that each asynchronous serial interrupt source can be configured for use with either CPU.

An interrupt from the V452 Series asynchronous serial interface means that one of the 2692 DUARTs has sensed some event for which it has been programmed to generate an interrupt, such as timer time-out, received character, transmitter empty, etc.

The "interrupt status" byte from the 2692s indicates which event(s) generated the interrupt. In general, the level 5 (standard) autovector interrupt routine would check this status byte and then jump to the appropriate service routine.

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



Asynchronous serial I/O ports

EZ-bus modules

The EZ-bus daughter modules (if present) can be configured either to generate a Level 4 autovector, or provide their own interrupt vector number depending on the daughter module in use. The meaning of the daughter module interrupt also depends on the nature of the daughter module being used.

EZ-bus modules A and B are **steerable** interrupt sources. On dual-processor V452 Series boards this means each module can be configured for use with either CPU.

For more information, see the ***EZ-bus interface*** chapter in Section 5 and the manual accompanying the EZ-bus module in use.

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

CPU Mailbox

V452 Series boards provide CPU mailbox circuitry that can receive messages to the on-board multi-ported memory space from either the local CPU or an external VME or EZ-bus Master. The CPU "reads" these 4-bit messages from a 64x4 Mailbox FIFO in local I/O space.

CPU Mailboxes 0 and 1 are **non-steerable** autovector Level 3 interrupt sources. On dual-processor V452 Series boards this means that the CPUs can receive interrupts only from their own mailbox circuitry (i.e., Mailbox 0 to CPU-X and Mailbox 1 to CPU-Y). Both CPU mailboxes can be active interrupt sources at the same time (on dual-processor boards).

If the mailbox interrupt is enabled, the CPU mailbox circuitry interrupts the corresponding CPU as an autovector Level 3 source whenever it receives a message. The mailbox is able to hold up to 64 separate messages and sends an interrupt to the CPU for each message received.

For more information, see the *CPU Mailbox* chapter in Section 4.

82C54 counter

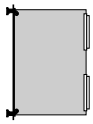
The V452 Series boards provide two autovector Level 2 interrupt sources from the on-board 82C54 CHMOS programmable interval timer. The 82C54 can be used as an event timer, elapsed time indicator, programmable one-shot, or other counter application.

The 82C54 provides three 16-bit counters of which only two (Counter 0 and Counter 1) can assert an interrupt. These two counters are **non-steerable** interrupt sources. On dual-processor V452 Series boards this means that the CPUs can receive interrupts only from their own counter (i.e., Counter 0 to CPU-X and Counter 1 to CPU-Y). Both Counters can be active interrupt sources at the same time (on dual-processor boards).

Counter interrupts remain in force until cleared via software. To clear a counter interrupt, disable the counter as an interrupt source and then re-enable it. For example, an interrupt from Counter0 to CPU-X could be cleared by executing the following two 68000 assembler commands:

```
moveb    #0x00, 0xFE39000E   |Disable counter int. source
moveb    #0x00, 0xFE39000F   |Re-enable counter int. source
```


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



Local bus timeout

The V452 Series boards include a local bus timeout circuit that generates a bus error if an individual bus transfer (or individual BLT) exceeds 30 microseconds. The local bus timeout generator provides a convenient method for checking for the presence or absence of an installed EZ-bus module(s). Attempting a transfer via an absent daughter module causes a local bus error, thereby providing a graceful exit from the faulty condition.



This local bus timer is automatically disabled whenever the CPU (or an EZ-bus module acting as the VME Master) is performing transfers across the VMEbus. In this case, the System Controller monitors the progress of the transfer and generates a VMEbus error if necessary.

VMEbus interrupt handler

The VMEbus specification establishes a systematic method by which one board can interrupt another board on the same bus. VME interrupts are requested by a board called the **interrupter** that asserts an interrupt at a particular interrupt level (Level 7 – 1). VME interrupts are serviced by a board on the bus called the **interrupt handler** that responds to interrupts at a single or a contiguous range of interrupt levels.



If using the VMEbus Interrupt Handler, do not use interrupt Level 4 if the V452 Series board is installed with a daughter board. This will avoid conflict with EZ-bus interrupts.

The paragraphs below describe the sequence of events that occur between the interrupter board and the corresponding interrupt handler board during a typical interrupt sequence on the VMEbus:

❶ The interrupter generates an interrupt request at a given level via the VMEbus interrupt request (IRQn*) lines.

❷ The interrupt request is received by the interrupt handler that has been assigned to that level. The interrupt handler acknowledges the receipt of the interrupt by generating an interrupt acknowledge cycle. The interrupt acknowledge is passed via daisy-chain from board to board in the bus starting at the board in Slot 1 and proceeding to Slot 2, Slot 3 and so on though all of the boards in

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

the card cage. The daisy chain automatically skips over empty slots for which the IACK daisy-chain jumper has been installed. Each slot (except Slot 1) provides an IACK daisy-chain jumper near the P1 connector on the back side of the VMEbus backplane.

❸ The interrupter places an interrupt vector on the bus in response to the interrupt acknowledge cycle.

❹ The interrupt handler board uses this interrupt vector to direct its processor to the proper interrupt vector routine.

Each of the seven VME interrupt levels acts as a **steerable** interrupt source. On dual-processor V452 Series boards this means that all VME interrupts can be configured for use with either CPU.



If the same interrupt priority is assigned to both an **on-board** and a VMEbus interrupt, the **on-board** interrupt has priority over the VMEbus interrupt.

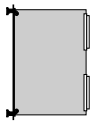
Once configured, the operation of the interrupt handler on the V452 Series board is fairly automatic. The CPU on the V452 Series board is programmed to respond to one or more VMEbus interrupt levels by writing to the address corresponding to the desired VMEbus interrupt level as listed in the table below.

Address locations — Interrupt Control register

Interrupt	Disable (default)	Enable
VME Level 1 to CPU-X	FE39 4002	FE39 4003
VME Level 2 to CPU-X	FE39 4004	FE39 4005
VME Level 3 to CPU-X	FE39 4006	FE39 4007
VME Level 4 to CPU-X *	FE39 4008	FE39 4009
VME Level 5 to CPU-X	FE39 400A	FE39 400B
VME Level 6 to CPU-X	FE39 400C	FE39 400D
VME Level 7 to CPU-X	FE39 400E	FE39 400F
VME Level 1 to CPU-Y	FE39 C002	FE39 C003
VME Level 2 to CPU-Y	FE39 C004	FE39 C005
VME Level 3 to CPU-Y	FE39 C006	FE39 C007
VME Level 4 to CPU-Y *	FE39 C008	FE39 C009
VME Level 5 to CPU-Y	FE39 C00A	FE39 C00B
VME Level 6 to CPU-Y	FE39 C00C	FE39 C00D
VME Level 7 to CPU-Y	FE39 C00E	FE39 C00F

* Note: Do not use VME Interrupt Level 4 if daughter board installed on the V452 Series board.

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





After power cycling or local reset, the V452 Series interrupt handler does **NOT** respond to any VMEbus interrupt levels. Typically, the bootup software/ firmware configures the interrupt handler to respond to the desired VMEbus levels.

For example, to enable VMEbus interrupt level 1 to be handled by CPU-X, the following instruction would be executed:

```
moveb #0x00,0xFE394003      | Enable level 1 VME ints to CPU-X
```

To disable CPU-X from handling level 1 VMEbus interrupts, the following instruction would be executed:

```
moveb #0x00,0xFE394002      | Disable level 1 VME ints to CPU-X
```

Once the V452 Series interrupt handler circuitry is configured to respond to the desired VMEbus interrupt(s), the CPU must also be programmed as follows.

- An interrupt vector table must be created in memory that contains the address of the interrupt service routine associated with each vector.
- Each of the interrupt service routines listed in the vector table must be loaded into memory at the address listed in the table.
- The CPU must be told where in memory to find the start of the vector table mentioned above. This is done by writing the starting memory address of the vector table in the CPU's internal Vector Base register.



Creating a vector table and service routine in memory and programming the Vector Base register in the 68040/68060 are identical to the methods used on preceding members of the 68000 processor family. For more information, consult the Motorola programmers guide for the applicable CPU.

Once the V452 Series board's interrupt handler and the CPU have been programmed to respond to VMEbus interrupts as described above, the CPU will respond to the configured VMEbus interrupt level and have its execution temporarily directed to the interrupt service routine according to the vector addresses stored in its vector table.

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

Configuring the VMEbus interrupter/resetter

The V452 Series VMEbus interrupter can assert interrupts to the VMEbus at any one of seven VMEbus interrupt priority levels and corresponding to any one of the 256 possible 68000 interrupt/exception vectors. This same circuitry can be used to assert the VMEbus Reset signal to cause a local reset of each board on the VMEbus.

> ⚠ To ensure that it is able to assert the VMEbus System Reset signal for a long enough period to ensure that all boards on the bus are reset, the V452 Series board cannot respond to this signal itself. Thus before V452 Series boards can be used to assert the VMEbus System Reset signal, the board must also be configured to ignore the VMEbus System Reset signal itself. For information about setting up the board to ignore the VME System Reset signal, refer to the ***Reset via software*** discussion in the ***Default and reset conditions*** chapter in Section 3.

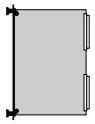
Setting the VME interrupt level

The level of the VME interrupt asserted by the interrupter is set by writing to a pair of registers that reside on one of the board's 2692 dual-UART. The general-purpose output bits 5, 6, & 7 for the 2692 select the interrupt level that is used as listed in the table below:

VME interrupt level / VME reset configuration values

Interrupt level	Programmable bits 7 6 5	Write to FE28003F	Write to FE28003B
Assert VMEbus Reset	0 0 0	0x00	0x0E
1	0 0 1	0x20	0xC0
2	0 1 0	0x40	0xA0
3	0 1 1	0x60	0x80
4	1 0 0	0x80	0x60
5	1 0 1	0xA0	0x40
6	1 1 0	0xC0	0x20
7 (default)	1 1 1	0xE0	0x00

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





In the default condition after a reset or power cycle, these 2692 registers select a VME Level 7 interrupt.

For example, to set up the board to assert a Level 5 interrupt, execute the following two 680x0 assembler commands:

```
moveb     #0xA0, 0xFE28003F   |Select Level 5 VME interrupt
moveb     #0x40, 0xFE28003B   |Select Level 5 VME interrupt
```



The two register locations on the 2692 DUART that are used to set the VME interrupt level asserted by the VME interrupter (0xFE28 003F, 0xFE28 003B) are also used to configure several other critical functions on V452 Series boards including VME SysFail reception, Serial Port B enable/disable and hardware handshaking (RTS) enable/disable. As a result, exercise due caution when writing to these locations. For information about the other functions controlled by these 2692 registers, see the ***V452 Series internal registers*** chapter in this section.

Setting the vector for VME interrupts

The 8-bit VME interrupt vector register (0xFE30 0000) supplies the 680x0 interrupt/exception vector for an asserted VME interrupt. Writing a byte to the VME Interrupt Vector register configures the vector number driven to the VMEbus during the interrupt acknowledge cycle. For example, the VME interrupter vector could be set to 0x50 by executing the following 680x0 assembler instruction:

```
moveb     #0x50, 0xFE300000   | set VME int. vector to 0x50
```

The interrupter generates an interrupt by first resetting the interrupter circuit then invoking the interrupter once. The code sequence in 680x0 assembly language is:

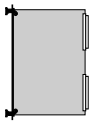
```
moveb     #0x07,0xFE380003    | reset interrupter
moveb     #0x0F,0xFE380003    | invoke interrupter
```


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



For a complete list of all the interrupts/ exceptions and their associated vectors for the 68040/68060 and V452 Series boards, see the appropriate ***680x0 CPU*** chapter in Section 4 or the ***V452 Series Quick Reference Card.***

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



Jumpers, switches, LEDs & Fuses

This chapter identifies the user-selectable jumpers, switches, LED indicators and fuses on V452 Series boards.

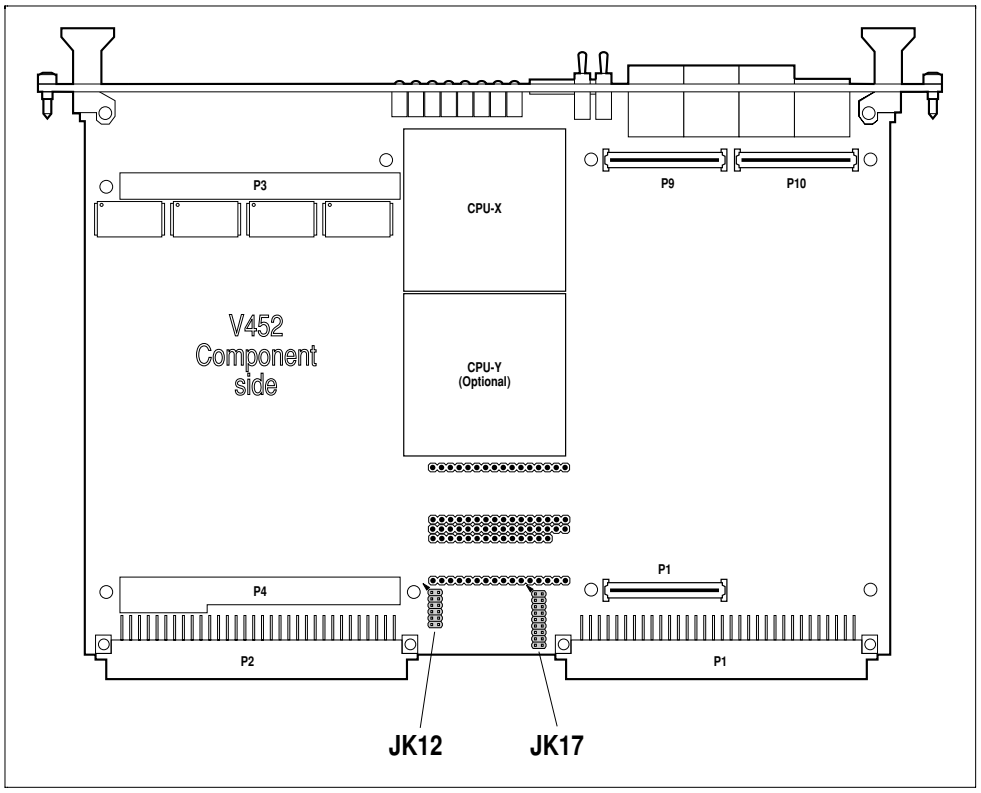
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Jumpers

The table below lists the V452's user-selectable jumper blocks. The location for these jumper blocks is shown in the drawing below.

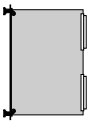
V452 Series user-selectable jumper blocks

Jumper block	Function	For more info, see (chapter)
JK12	Monitor EPROM type selection	*EPROM*
JK17	Flash EPROM Write Enable/Disable	*EPROM/Flash Memory Module*
	VME Master request level select	*VME Master interface*
	Remote Reset Enable	*VME Slave interface*
	Round Robin Enable	*VME Master interface*
	System Controller Force Disable	*System Controller*
	Flash Boot Enable	*EPROM/Flash Memory Module*

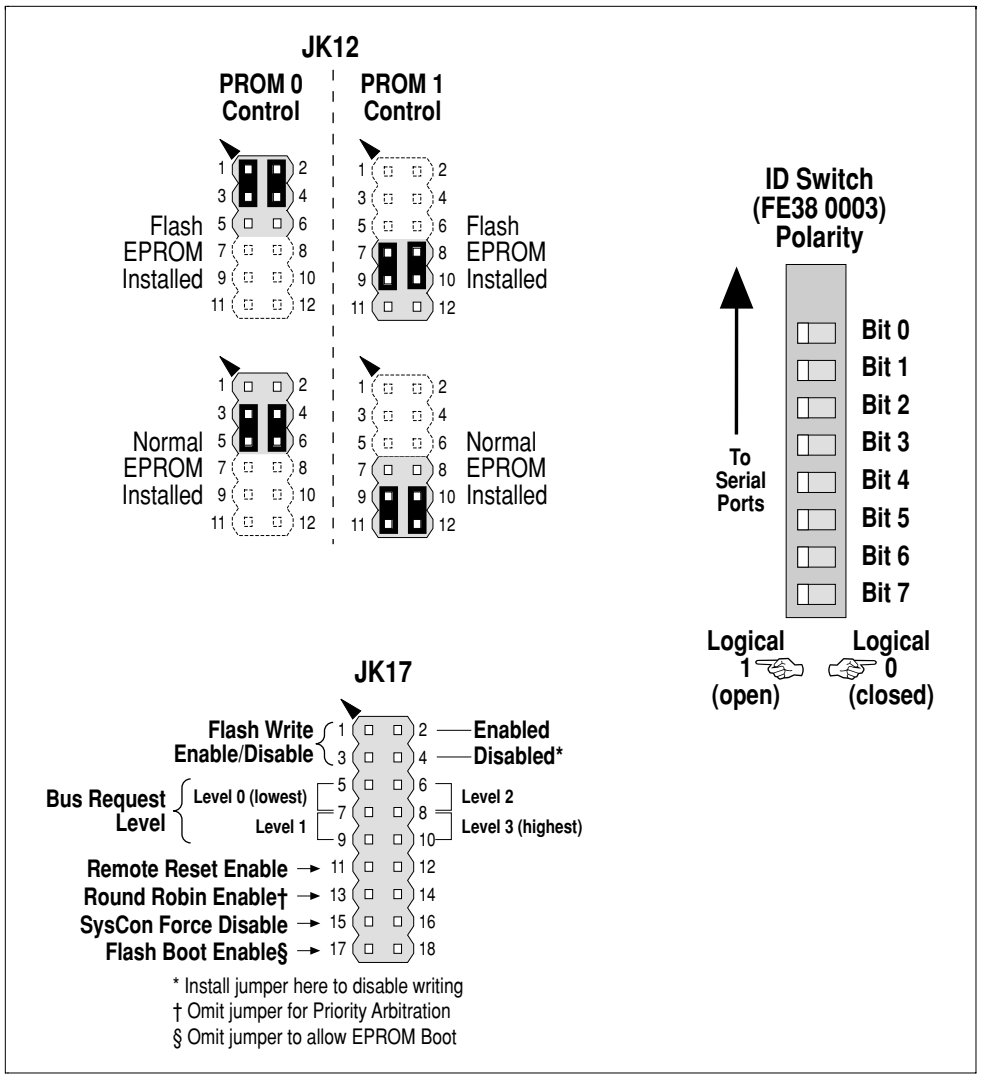


V452 Series jumper locations

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



The figure below shows the jumper settings to configure V452 Series boards. The jumper assignments are summarized in the following tables.



V452 Series jumper settings

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

JK12 jumper assignments

	Flash EPROM installed	Normal EPROM installed
PROM0, JK12 jumpers on pins:	1 & 3, 2 & 4	3 & 5, 4 & 6
PROM1, JK12 jumpers on pins:	7 & 9, 8 & 10	9 & 11, 10 & 12

JK17 jumper assignments

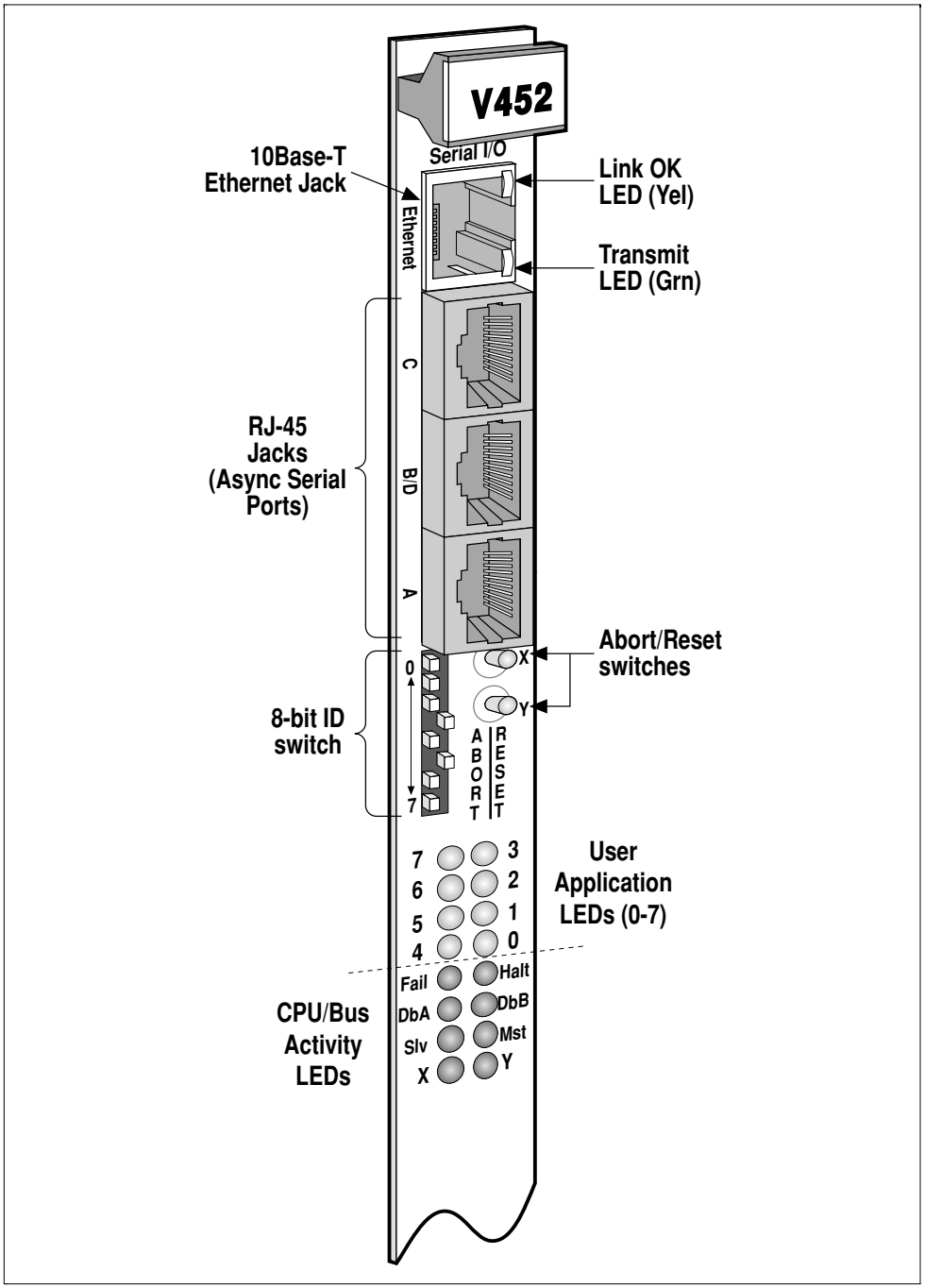
	Enable	Disable
Flash Write	1 & 2	3 & 4
Bus Request Level 0[1]	5 & 7	—
Bus Request Level 1[1]	7 & 9	—
Bus Request Level 2[1]	6 & 8	—
Bus Request Level 3[1]	8 & 10	—
Remote Reset Enable	Jumper installed on 11 & 12	Jumper removed from 11 & 12
Round Robin Enable[2]	Jumper installed on 13 & 14	Jumper removed from 13 & 14[2]
System Controller Force Disable	Jumper installed on 15 & 16	Jumper removed from 15 & 16
Flash Boot Enable[3]	Jumper installed on 17 & 18	Jumper removed from 17 & 18[3]

Notes:
1. Bus request level set by software if no jumpers installed for bus request level.
2. Board set to Priority Arbitration with jumper removed from 13 & 14.
3. Board boots from EPROM with jumper removed from 17 and 18.

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

Front panel

The drawing below shows the layout of the connectors, controls, and indicators on the V452 front panel.



V452 Series front panel

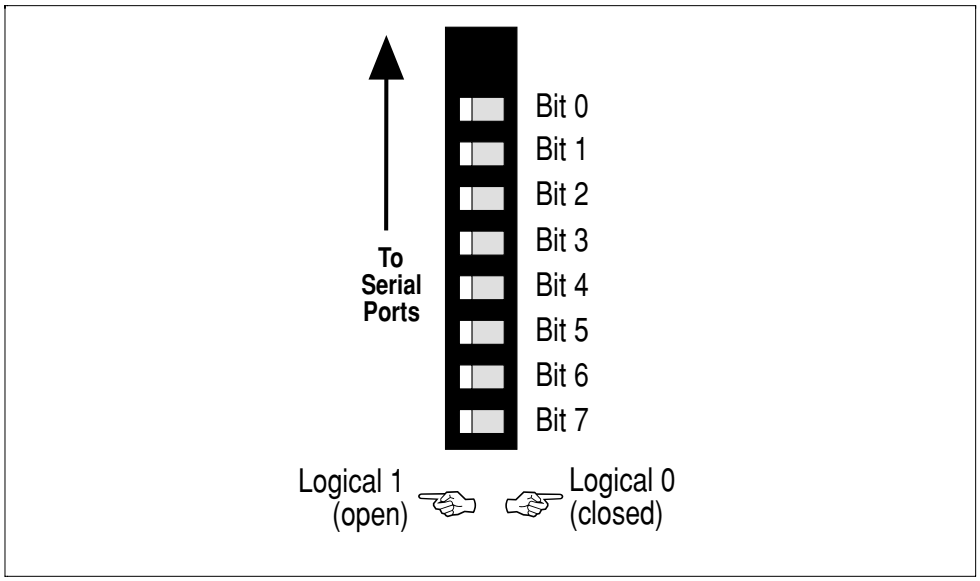
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8-bit ID switch (front panel)

The eight position ID switch on the front panel of the V452 Series provides an 8-bit software-readable ID switch.

Readable switches can be very useful in target applications where applications programs can read the switch to discover what their function should be, the nature of their peripherals, etc.

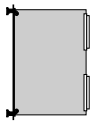
The CPU reads the switch setting by performing a byte-wide read to the ID Status register at memory location **0xFE38 0003.** The figure below shows the register bits corresponding to each of the eight switch positions.



ID switch polarity

Numbering may appear on the switch component itself that conflicts with the numbering shown above. Ignore all numbering schemes except what is shown above and on the ***Quick Reference Card***.

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



Toggle switches

V452 Series boards provide RESET and ABORT switches that allow you to apply these functions to the on-board CPUs:

RESET

Asserts either a **CPU** or **board-level** RESET as described in the figure and text below:

Pushing **one** of the switches to the right asserts a CPU-level RESET to the corresponding CPU. The **CPU-X** (top) switch asserts a reset to the CPU on single CPU models and to **CPU-X** on the dual CPU models. The **CPU-Y** switch (bottom) asserts a reset to CPU-Y which has an effect only on dual CPU models.

Pushing **both** switches to the right at the same time asserts a board-level reset on all V452 Series models:

- Resets the CPU(s).
- Resets all on-board components that have such a function and clears all on-board control registers.
- Asserts a VME RESET if the board is serving as the System Controller.

For more information, see the ***Default & reset conditions*** chapter in Section 3 and the ***System controller*** chapter in Section 5.

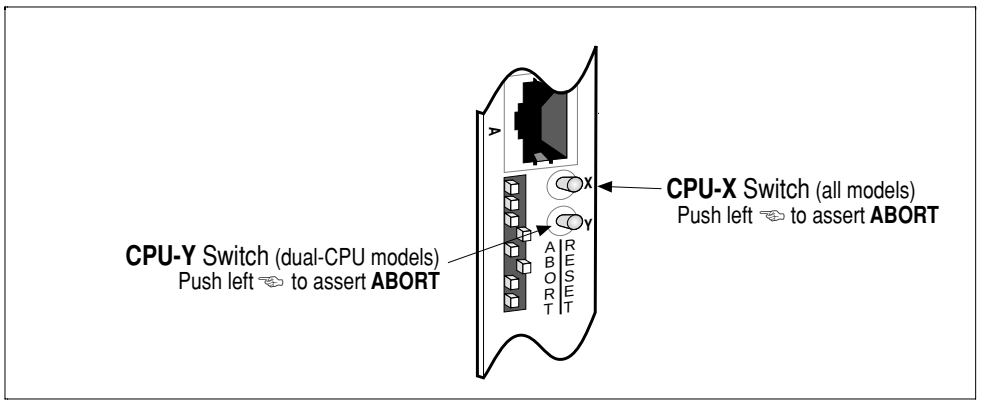
ABORT

Pushing either switch to the **left** as shown in the figure above asserts an ABORT to the respective CPU. Assert an ABORT **stops the current process** by issuing a **non-maskable level 7 interrupt** (nmi).

Pushing the bottom switch to the left has **no effect** on single processor boards.

The ABORT switch is always assigned to interrupt level 7 and cannot be set to another level. For more information, see the ***Interrupts*** chapter in this section.

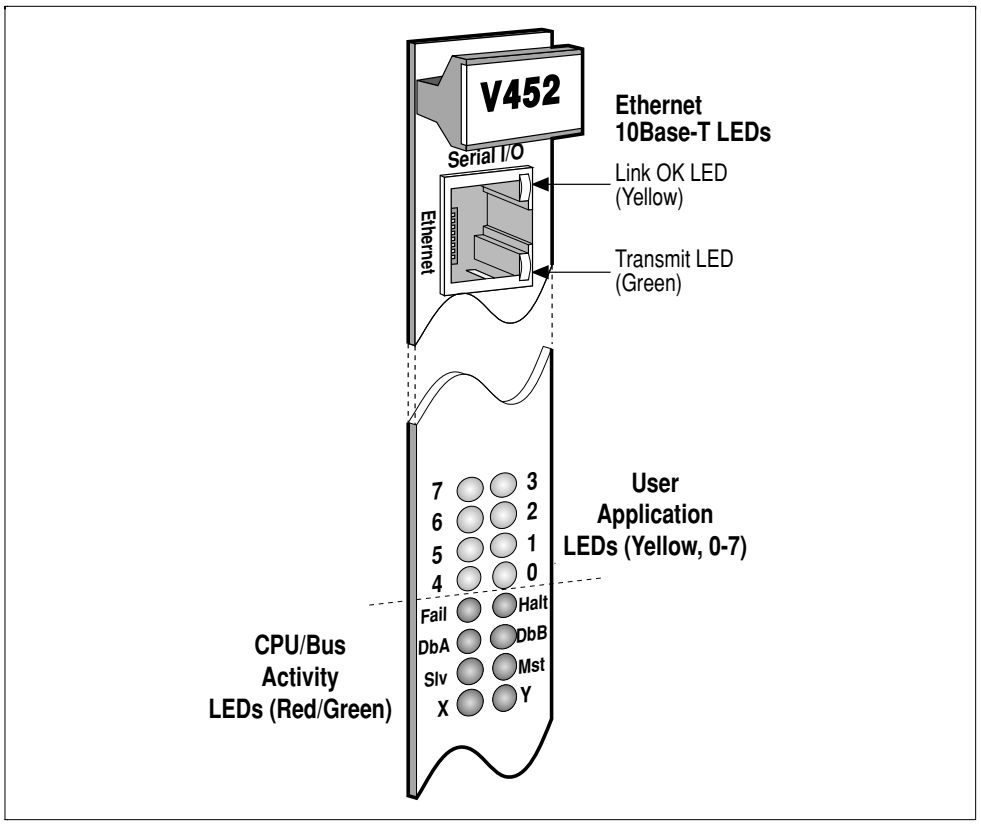
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Close-up of RESET and ABORT switches

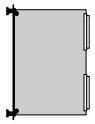
LEDs

Shown below are the V452 front panel LEDs which provide a quick visual indication of board activity. The following discussion describes the function of these LEDs.



V452 Series LEDs

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



The eight **yellow** LEDs indicate **application events**:

0 - 7 Software-programmable LEDs are controlled by the V452 Series's Primary and Extended Mode registers. They indicate the current **operating mode** of the board as defined by the software currently running.

For more information about how to program these LEDs, see the chapter on the *V452 Series internal registers* which follows this chapter.

The two **red** LEDs indicate the **run status** of the board:

Fail Indicates the **SysFail** status of the board.

When the CPU is reset (either by the VMEbus SysRes\ line or the front panel RESET toggle), the Fail LED turns on and the board drives the VME SysFail\ signal.

During normal operation the system boot software clears this condition shortly after RESET. The SysFail LED and driver signal are cleared by performing a write of a **0C** to the Primary Mode register at **0XFE38 0003**. The program may also turn this LED on (i.e. assert SysFail on the VMEbus) by writing a **04** to **0xFE38 0003**.

Halt On single-68040 model boards, this LED provides a visual indication that the on-board CPU has **HALTed.**

On dual-processor boards, this LED indicates that one or both CPUs has HALTed. If only one CPU has HALTed, it is possible to identify which one by also looking at the **X** and **Y** LEDs (described below). Generally, the LED for the HALTed CPU is **NOT** lit. However, this indication is very dependent on the application being run and how closely coupled the two CPUs are operating.

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

The six **green** LEDs indicate the device currently in **control** of the V452 Series board's local bus and/or the VMEbus:

DbA / DbB	Indicate that either the first or second installed EZ-bus daughter module (A or B) is in control of the board's local bus. These LEDs are active only if one or two EZ-bus modules (that can take control of the bus) have been installed. For more information about EZ-bus modules, see the the ***EZ-bus interface*** chapter in Section 5.
Slv	Indicates that an external VMEbus Master is in control of the local bus and the board is acting as a VMEbus Slave. For more information, see the ***VME Slave interface*** chapter in Section 5.
Mst	Indicates that the V452 Series board is acting as the Master on the VMEbus. For more information, see the ***VMEbus Master interface*** chapter in Section 5.
X	Indicates that CPU-X is currently in control of the local bus. CPU-X is installed on **all** V452 Series boards.
Y	Indicates that CPU-Y is currently in control of the local bus. CPU-Y is installed only on **dual-processor** V452 Series boards.

The two LEDs on the onboard Ethernet 10Base-T jack indicate Ethernet port status as follows:

Yellow	Link OK – lit when 10Base-T cable is properly plugged into a functioning 10Base-T network.
Green	Transmit – lit whenever port passes data.

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

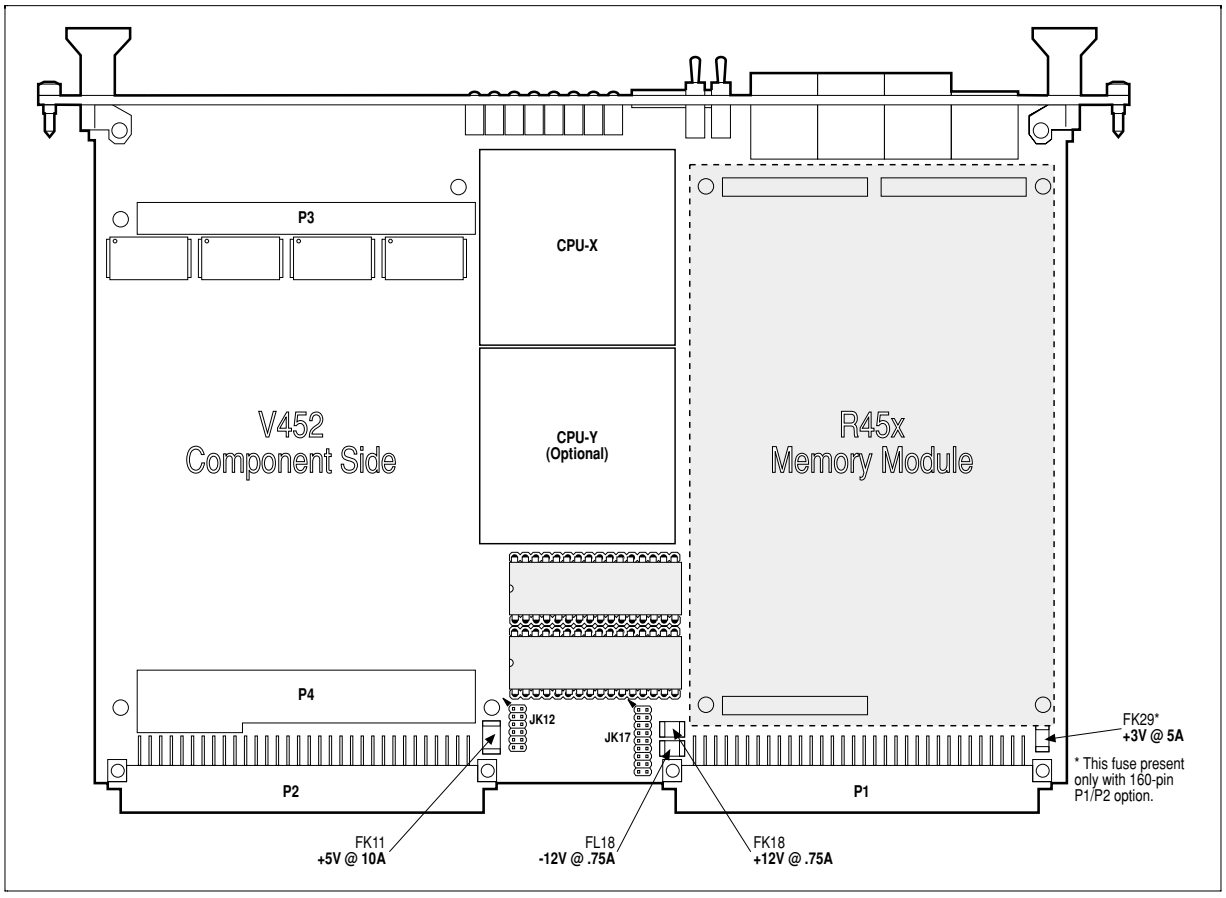
Fuses

The table below lists the fuses used on the V452 SBC. All fuses are surface-mounted devices which are factory replaceable parts only. The drawing below shows where these fuses are located on the SBC.

Fuse listing, V452

Fuse ref. designation	Rating	Resettable?* (Yes/No)
FK11	5V @ 10A	No
FK18	+12V @ .75A	Yes
FL18	–12V @ .75A	Yes
FK29	+3V @ 5A	No

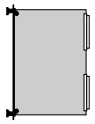
* To reset fuse, remove power for minimum of 20 seconds then reapply power.



Fuse locations, V452

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

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



V452 Series internal registers

This chapter describes the contents and use of the internal registers provided on V452 Series boards to monitor and control the operation of various on-board features and functions. These registers fall into one of the following four categories:

- **Status registers** — are *read-only* registers that indicate the *status or condition* of on-board devices or processes. Using these registers involves reading the register and interpreting the bit pattern found there.
- **Mode registers** — are *write-only* registers that *set up the board* to perform a given operation or function. Mode registers are *address and data-specific*. Using a Mode register involves writing a *particular hexadecimal value* to a *specific address* location.
- **Control registers** — are *write-only* registers that also *set up the board* to perform operations or functions. However, Control registers are only *address-specific*. Using a Control register involves writing to a specific address location; the hex value of the data that is written DOES NOT matter.

  For purposes as setting a convention for software coding, all sample instructions to perform writes to Control registers appearing in this manual will specify a 0x00 hex value even though any value could be used.

- **Board information registers** — are *read-only* registers that provide the system with ID and configuration information per the VME64 extensions specification.

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

Status register

The 8-bit Status register indicates the presence or absence of various signals or the status of certain processes on the local or VMEbus. Application programs can determine the operating status of the board by reading the Status register bits using either of the methods listed below:

- A **byte** read at **0xFE38 0002** reveals the Status register bits listed below.
- A **word** read of the Status register at **0xFE38 0002** reveals both the Status register bits and the ID switch register (described later in this chapter). In the resulting word, the Status register bits are shifted to the left by eight bits to most significant byte (8-15) and the ID switch bits appear in the least significant byte (0-7).

Status register bits

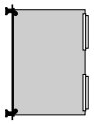
Bit	When 0	When 1	For more info, see chapter
0	No parity error	Parity error	*Dynamic RAM*
1	ABORT/Level 7 int. requested	ABORT/int. not requested	*Interrupts*
2	SysFail\ present on VMEbus	SysFail\ not present	*Interrupts*
3	ACFail\ present on VMEbus	VME ACFail\ not present	*Interrupts*
4	Block Transfer error asserted	No BLT error	*VME Master BLT*
5	Other CPU halted	Other CPU running	*68040/68060 CPU*
6	Single 68040 configuration	Dual 68040 configuration**	*68040/68060 CPU*
7	CPU-X controls the local bus	CPU-Y controls the local bus*	*68040/68060 CPU*

Note: * For single CPU boards, these bits always = 0.
** Requires a PROM to be installed in both monitor PROM sockets.



Status bit 7 indicates which CPU has control of the bus. On dual-CPU boards, this bit can be used by either CPU to identify itself. By accessing this bit, the CPU takes control of the local bus, thus Status register bit 7 serves as a “mirror” of sorts, allowing the CPU in control to identify itself.

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

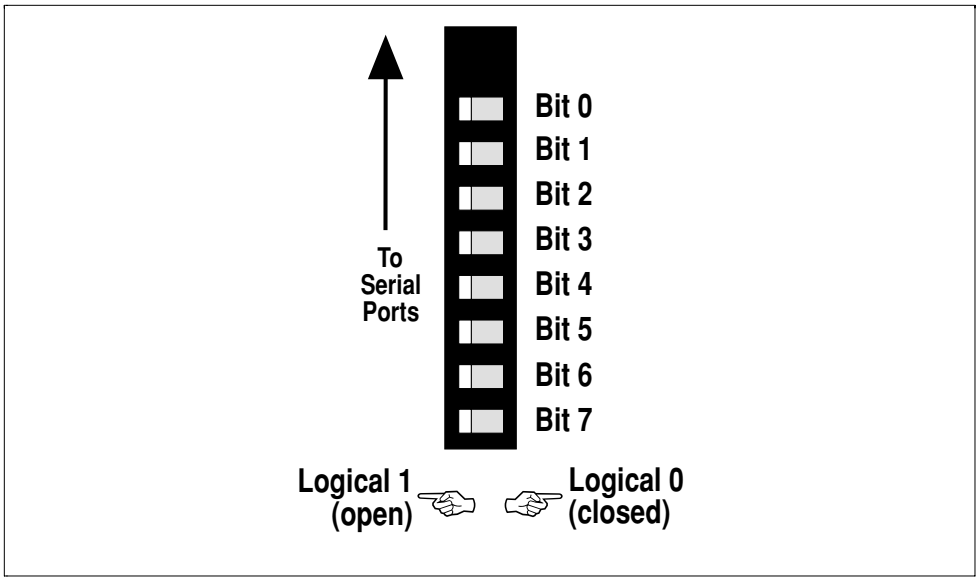


ID status register

The register containing the settings for the 8-bit ID switch on the front panel is located at **0xFE38 0003** (read access).

ID switch bits

Bit	When 0	When 1	For more info, see chapter
0 - 7	Switch closed (to right)	Switch open (to left)	*Jumpers, switches, LEDs & fuses*



ID switch polarity

This register can be accessed in two ways:

- Performing a **byte** read at **0xFE38 0003** reveals the current state of all eight positions of the ID switch in bits 0 - 7.
- Performing a **word** read of the Status register at **0xFE38 0002** is a convenient way to access both the eight Status register bits (described earlier in this chapter) and the 8 ID switch settings. In the resulting word, the Status register bits are shifted to the left by eight bits to most significant byte (8 -15) and the ID switch bits appear in the least significant byte (0-7).

For more information about using the ID switch, see the ***Jumpers, Switches, LEDs and fuses*** chapter in this section.

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

Mode registers

The V452 Series uses the following Mode registers to control various on-board functions as described in the paragraphs below:

- the ***Primary*** Mode register is located at address **0xFE38 0003** (write access only). It controls half of the front panel user LEDs and various other functions.
- the ***Extended*** Mode register is located at address **0xFE38 4003** (write access only). It controls the remaining half of the front panel user LEDs and various other functions.
- V452 Series boards use several of the general purpose mode registers in the ***2692 UART*** to control reception of the VMEbus SysFail\ signal, to operate the VMEbus interrupter, and to enable serial ports B and D**.**
- the 8-bit, ***VME interrupt vector register*** (0xFE30 0000) on V452 Series supplies the 680x0 interrupt/exception vector for the asserted VME interrupt.

Using Mode register functions

Activating a Mode register function involves writing the appropriate hexadecimal data value to the appropriate Mode register:

For example, turning off LED 0 would require writing **08** to the Primary Mode register at **0xFE38 0003** using the following 68040 instruction:

```
moveb          #0x08, 0xFE380003
```

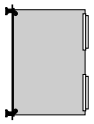
Enabling the V452 Series VME Slave interface would require writing **0F** to the Extended Mode register at **0xFE38 4003** using the following 68040 instruction:

```
moveb          #0x0F, 0xFE384003
```

Mode register conditions after Reset

After power cycling or a system reset, both mode registers return to their pre-initialized default conditions. In the tables to follow these default values are identified.

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



Primary & Extended Mode registers

The tables below list the functions controlled by the Primary and Extended Mode registers and the data value required to invoke them.

Primary Mode register (Mode1; OxFE38 0003)

Hex data	Function	For more info, see chapter
00	Turn on LED 0 (default)	*V452 Series internal reg.*
01	Turn on LED 1 (default)	*"*
02	Turn on LED 2 (default)	*"*
03	Turn on LED 3 (default)	*"*
04	Turn on SysFail\ light and bus signal (default)	*", Interrupts*
05	Turn VME Slave memory protection off (default)	*VME Slave interface*
06	Disable on-board RAM parity error reporting (default)	*"*
07	Reset VMEbus interrupter	*Interrupts*
08	Turn off LED 0	*V452 Series internal reg.*
09	Turn off LED 1	*"*
0A	Turn off LED 2	*"*
0B	Turn off LED 3	*"*
0C	Turn off SysFail\ light and bus signal	*", Interrupts*
0D	Turn memory protect on	*VME Slave interface*
0E	Enable on-board RAM parity error reporting	*"*
0F	Invoke VMEbus interrupter/resetter once	*Interrupts*

Extended Mode register (Mode2; OxFE38 4003)

Hex data	Function	For more info, see chapter
00	Turn on LED 4 (default)	*V452 Series internal reg.*
01	Turn on LED 5 (default)	*"*
02	Turn on LED 6 (default)	*"*
03	Turn on LED 7 (default)	*"*
04	Disable CPU Watchdog Run monitor (default)	*CPU Watchdog*
05	Disable EZ-bus snooping (default)	*68040/68060 CPU*
06	Disable VMEbus snooping (default)	*68040/68060 CPU*
07	Disable Slave access from VMEbus Masters (default)	*VME Slave interface*
08	Turn off LED 4	*V452 Series internal reg.*
09	Turn off LED 5	*"*
0A	Turn off LED 6	*"*
0B	Turn off LED 7	*"*
0C	Enable CPU Watchdog Run monitor	*CPU Watchdog*
0D	Enable EZ-bus snooping	*68040/68060 CPU*
0E	Enable VMEbus snooping	*68040/68060 CPU*
0F	Enable Slave access from VMEbus Masters	*VME Slave interface*

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

Controlling the User LEDs

The eight user-programmable LEDs on the front panel of V452 Series boards are among the functions controlled by the Mode register. The paragraphs below describe two methods for controlling the user LEDs via the Mode register.

Manual method — This method involves setting each LED manually by writing the appropriate hex value directly to the Mode register.

Arbitrary Pattern algorithm — This second method uses a modular algorithm that automates display of all four LEDs in a program loop. Using this algorithm (which is included) it is possible to specify an LED pattern by performing a **single** write instruction as input to a program loop, rather than by performing **four** separate write instructions as required for the manual method.

Controlling the User LEDs manually — The first method for controlling the user LEDs involves writing directly to the Mode register to control a specific LED. Each LED has a hex data value associated with its ON and OFF state as listed in the table below:

Mode register (0xFE38 0003) bit definitions for LED functions

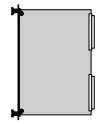
Hex data	Function
00	Turn ON LED 0 (default)
01	Turn ON LED 1 (default)
02	Turn ON LED 2 (default)
03	Turn ON LED 3 (default)
04 - 07	Other functions *
08	Turn OFF LED 0
09	Turn OFF LED 1
0A	Turn OFF LED 2
0B	Turn OFF LED 3
0C - 0F	Other functions *

* For more information about the other *Mode register* functions, see the complete *Mode register* table on the previous pages.

For example, it is possible to turn ON LED 3 using the following 68000 assembler instruction.

```
moveb        #0x03, 0xFE380003
```

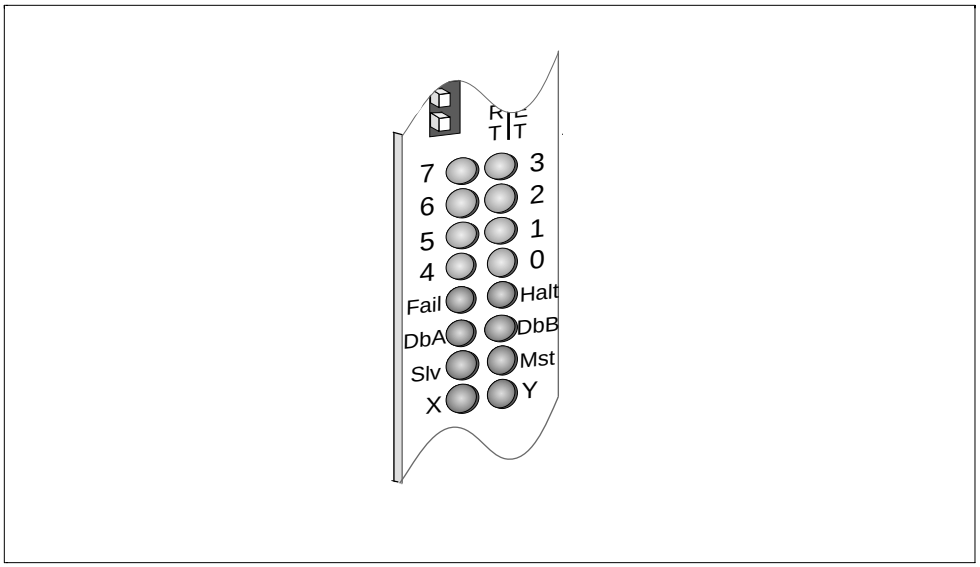

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Arbitrary LED control — The second method for controlling the user LEDs uses a function written in C that is listed on the next page. This function accepts a single hex integer (from 0x0–0xF) to specify a 4-bit LED pattern (either LEDs 0–3 or LEDs 4–7).

The LED "bank" this function controls at any given moment depends on which CPU is in control of the local bus when the function is invoked (as indicated by Bit 7 of the Status register).

When CPU-X is in control of the bus and Status register bit 7 is 0, the DisplayLED function writes a 4-bit pattern to LEDs 4–7 which are the four yellow user LEDS ABOVE the green CPU-X LED as shown in the cut away view of the V452 Series front panel below. When CPU-X is in control of the bus, Status register bit 7 is 0, the DisplayLED function writes a 4-bit pattern to LEDs 4-7 which are the four yellow user LEDS ABOVE the green CPU-Y LED.



V452 Series LEDs



The DisplayLED function on the next page is specifically written for a dual-CPU V452 Series model to provide each CPU with four separate user LEDs. Thus, using this function without modification on a single-CPU V452 Series board would allow the one CPU to use only four of the 8 available LEDS. Providing a single CPU with access to all 8 LEDs would require modification of the code.

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

Display LED function (in C)

```
#define  Primary ModeReg       0xFE38 0003
#define  ExtendedModeReg       0xFE38 4003
#define  LED_Off               0x08
```

* **The Display LEDS function** - displays the lowest 4 bits of a passed value in either the right LED bank (0-3) or left LED bank (4-7) depending on which CPU is in control of the local bus at that moment. This function automatically associates CPU-X activities with the left LED bank (4-7) and CPU-Y activities with the right LED bank (0-3).

```
DisplayLEDS( int value )
{
  u_char *LEDRegister;
  int i;

  if ( isCPU-X )                                    /* If status register bit 7 = 0; CPU-X has bus
    {
      LEDRegister = (u_char *)ExtendedModeReg;      /* Adjust LEDs 4-7 via Extended mode reg.
    }
  else if ( isCPU-Y )                               /* If status register bit 7 = 1; CPU-Y has bus
    {
      LEDRegister = (u_char *)PrimaryModeReg;       /* Adjust LEDs 0-3 via Primary mode reg.
    }

  for( i = 0; i < 4; i++ )
    if ( value & (1<<i) )
      *LEDRegister = i;                             /* Sets the "ith" LED to default state or ON
    else
      *LEDRegister = (i | LED_Off);                 /* Turns the "ith" LED off
}
```

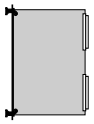
Consider the following LED display pattern:

LED 0 or 4 — ON
LED 1 or 5 — OFF
LED 2 or 6 — ON
LED 3 or 7 — OFF

If expressed as a 4-bit binary value, this LED ON/OFF pattern equals **0101** which in turn equals **0x5** in hexadecimal.

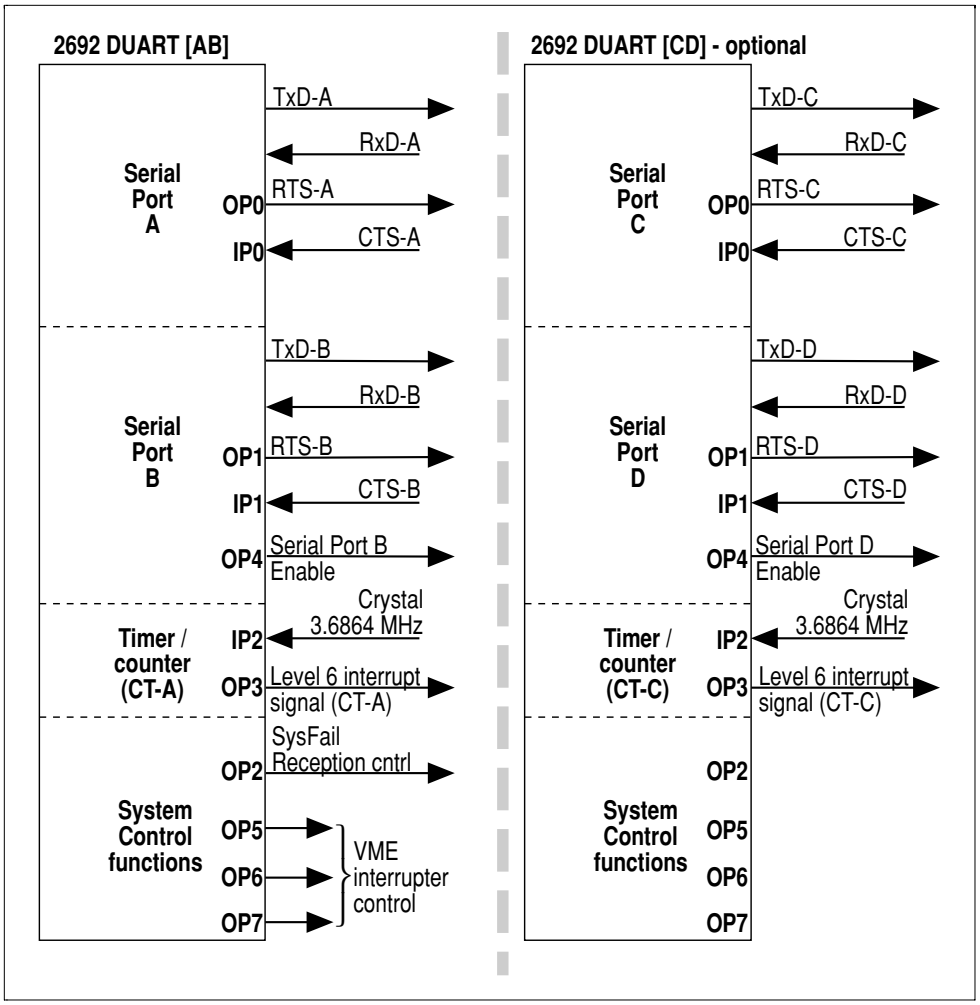
Thus, if the DisplayLED function were invoked and passed the 0x05 hex integer value while CPU-Y had the bus the LED pattern listed above would appear on LEDs 0–3.

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



2692 mode registers

V452 Series boards use the user-programmable Input Port (IP) and Output Port (OP) registers on the two 2692 dual-UARTs to control several on-board functions. The figure below is a schematic of the interface for the 2692's user-programmable Input and Output ports.



V452 Series 2692 interface (IP & OP registers)

The four tables on the facing page summarize the bits in the 2692 that are used by V452 Series boards to control the 2692 functions shown above and lists the address location for each IP and OP bit.

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

2692 [ABCD] Input Port (IP) functions (Read only)

2692 Input Port (IP) bit	FE28 0037 (Read address)	FE20 0037 (Read address)
IP0	Read CTS-A Status	Read CTS-B Status
IP1	Read CTS-C Status	Read CTS-D Status
IP2	Ext. Crystal Input to Timer CT-A (3.6864 MHz)	Ext. Crystal Input to Timer CT-C (3.6864 MHz)

2692 [ABCD] Output Port Configuration [OPCR] functions (Write only)

2692 Input Port (IP) bit	FE28 0037 (Write address)	FE28 0037 (Write address)
OPCR	Enable CT-A Level 6 interrupt (0x4)	Enable CT-C Level 6 interrupt (0x4)

2692 [AB] Output Port [OP] functions (Write only)

2692 Output Port (OP) bit	FE28 003B	FE28 003F
OP0	Assert RTS-A	Deassert RTS-A
OP1	Assert RTS-B	Deassert RTS-B
OP2	Enable SysFail Reception	Disable SysFail Reception
OP3	Timer (CT-A) Level 6 interrupt signal *	—
OP4	Enable Port B	Disable Port B
OP5	Set level for VME interrupter	Set level for VME interrupter
OP6	Set level for VME interrupter	Set level for VME interrupter
OP7	Set level for VME interrupter	Set level for VME interrupter

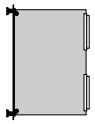
Notes * If configured to do so, the 2692 [AB] uses its OP3 pin to output the Level 6 interrupt signal from the CT-A timer to the processor. Setting this bit asserts a Level 6 interrupt.

2692 [CD] Output Port [OP] functions (Write only)

2692 Output Port (OP) bit	FE20 003B	FE20 003F
OP0	Assert RTS-C	Deassert RTS-C
OP1	Assert RTS-D	Deassert RTS-D
OP2	—	—
OP3	Timer (CT-C) Level 6 interrupt signal *	—
OP4	Enable Port D	Disable Port D
OP5	—	—
OP6	—	—
OP7	—	—

Notes * If configured to do so, the 2692 [CD] uses its OP3 pin to output the Level 6 interrupt signal from the CT-C timer to the processor. Setting this bit asserts a Level 6 interrupt.

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





These functions shown in the diagram on the facing page and the table above represent only the functions assigned to the "user-programmable" pins and registers on the 2692. The 2692 contains various other readable and writeable registers that control various serial communications and timer/counter functions. For more information about using these registers, see the ***Timers & counters*** chapter in Section 4 and the ***Asynchronous serial interface*** chapter in Section 5 and/or the *2692 DUART datasheet* in the ***VME CPU Datasheet Supplement*** which is supplied with this manual.

Enabling serial ports B and D

In the default condition after a reset or power cycle of the V452 Series board, serial ports B and D are disabled by the DUART's OP4 output driving the corresponding line driver's TxD and RTS lines to the high impedance mode (tri-state).

To enable serial port B, execute the following 680x0 assembler instruction:

```
moveb       #0x10, 0xFE28003B     ;enable TxD, RTS
```

To enable serial port D, execute the following 680x0 assembler instruction:

```
moveb       #0x10, 0xFE20003B     ;enable TxD, RTS
```

Disabling serial ports B and D

If for some reason you want to disable serial port B, either reset the board or execute the following 680x0 assembler instruction:

```
moveb       #0x10, 0xFE28003F     ;tri-state TxD, RTS
```

To disable serial port D, either reset the board or execute the following 680x0 assembler instruction:

```
moveb       #0x10, 0xFE20003F     ;tri-state TxD, RTS
```


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

Serial ports A and C on V452 Series boards are hardwired to the enabled state. No tri-state control of line drivers is done on these ports.

Configuring 2692 timer interrupt levels

The timer on each of the 2692 chips can be configured to assert its interrupt via the same pin as the serial port resulting in a Level 5 interrupt or via a separate pin (OP3) which the processor recognizes as a Level 6 interrupt.

To configure the timer for the Channel A&B 2692 (CT-A) to assert interrupts via the OP3 pin (Level 6), execute the following 680x0 assembler instruction which writes to the 2692's Output Port Configuration register (OPCR):

```
moveb       #0x4, 0xFE28037   |Config CT-A as Level 6 int. source
```

To configure the timer for the Channel C&D 2692 (CT-C) to assert interrupts via the OP3 pin (Level 6), execute the following 680x0 assembler instruction:

```
moveb       #0x4, 0xFE20037   |Config CT-C as Level 6 int. source
```

Writing any other value to these two bits causes the timer on the 2692 to assert its interrupt on the same pin as the serial ports (Level5).

Controlling VME SysFail reception

The VMEbus signal is often monitored by one board in a VMEbus system for system integrity purposes. Reception of this signal may be enabled or disabled via the general-purpose output bit 2 (OP2) on the Channel A&B 2692.

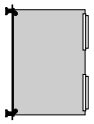
In the default condition or after a board-level reset, reception of the SysFail signal is disabled. To enable reception of the SysFail signal, execute the following 680x0 assembler instruction:

```
moveb       #0x04, 0xFE28003B
```

To disable reception of the SysFail interrupt, execute the following 680x0 assembler instruction:

```
moveb       #0x04, 0xFE28003F
```


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



VMEbus interrupter mode register

The V452 Series VMEbus interrupter can assert interrupts to the VMEbus at any one of the seven VMEbus interrupt priority levels and corresponding to any one of the 256 possible 68000 interrupt/exception vectors. Output port bits 5, 6, & 7 (OP5-7) on the Channel A&B 2692 select the level used by the V452 Series board to assert VMEbus interrupts as listed in the table below. Note that interrupt Level 0 is used to produce a VME reset (SysRes). Refer to the ***Interrupts*** chapter for more information on programming and using the interrupter.

VME interrupt level configuration values

Desired interrupt level	Programmable bits 7 6 5	Write to 0xFE28003B	Write to 0xFE28003F
0 (RESET)	0 0 0	0x0E	0x00
1	0 0 1	0xC0	0x20
2	0 1 0	0xA0	0x40
3	0 1 1	0x80	0x60
4	1 0 0	0x60	0x80
5	1 0 1	0x40	0xA0
6	1 1 0	0x20	0xC0
7 (default)	1 1 1	0x00	0xE0



In the default condition (after a reset or power cycle), these 2692 registers select a VME Level 7 interrupt.

VME interrupt vector register

The 8-bit VME interrupt vector register (0xFE30 0000) supplies the 680x0 interrupt/exception vector for an asserted VME interrupt. Writing a byte to the VME Interrupt Vector register configures the vector number driven to the VMEbus during the interrupt acknowledge cycle. For example, the VME interrupter vector could be set to 0x50 by executing the following 680x0 assembler instruction:

```
moveb        #0x50, 0xFE300000    | set VME int. vector to 0x50
```


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



For a complete list of all the interrupts/ exceptions and their associated vectors for the 68040 and V452 Series boards, see the applicable *CPU* chapter in Section 4 or the ***V452 Series Quick Reference Card.***

Control registers

V452 Series provide the following control registers that control and configure various feature and functions on V452 Series boards:

- the ***Primary*** control register is located at address **0xFE38 0003** (write access). It controls several miscellaneous functions.
- the ***Extended*** control register is located at address **0xFE38 4003** (write access). It controls several miscellaneous functions.
- the ***Interrupt*** control registers are a series of four 8-bit registers located at addresses **0xFE390000**, **0xFE394000**, **0xFE398000**, **0xFE39C000**, (write access). It enables, disables and steers the reception of on-board interrupts.
- the ***Slave Interface*** control register is located at address **0xFE388000** (write access). It sets up various aspects of the VME Slave interface.
- the ***Ethernet/VMEbus*** control registers start at address **0xFE3A4000** (write access). These registers set up the onboard Ethernet 10Base-T interface and the VMEbus request level selection.

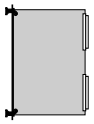
Using control register functions

Activating a control register function involves writing to the appropriate Control register location.

Control register conditions after reset

After power cycling or a system reset, both mode registers return to their pre-initialized default conditions. The following tables identify these default values.

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



Primary & extended control registers

Listed below are the Primary and Extended control register locations:

Primary control register (FE38 C000)

Address	Function	Description
FE38 C000	ROR release	Select Release on Request bus request handling (***default)***
FE38 C001	RWD release	Select Release When Done bus request handling
FE38 C002	non-FAIR requests	Selects non-FAIR (normal) bus release handling (***default)***
FE38 C003	FAIR requests	Select FAIR bus release handling
FE38 C004	CPU Watchdog	Disable CPU Watchdog Halt monitor ***(default)***
FE38 C005	CPU Watchdog	Enable CPU Watchdog Halt monitor.
FE38 C006	VME remote reset	Respond to VMEbus reset signal (***default)***
FE38 C007	VME remote reset	Ignore VMEbus reset signal
FE38 C008	Data broadcasting	Disable data broadcast mode (***default)***
FE38 C009	Data broadcasting	Enable data broadcast mode
FE38 C00A	VME data cache	Inhibit VME data cache (***default)***
FE38 C00B	VME data cache	Enable VME data cache
FE38 C00C	NVRAM/Clk/Cal	Default address space (***default, 2KB NVRAM)***
FE38 C00D	NVRAM/Clk/Cal	Extended address space (for NVRAM/Clk/Cal w/8KB NVRAM)
FE38 C00E-F	*reserved*	—

Notes: [1] Before you can use Data Broadcasting mode, you must perform a few simple hardware configuration tasks on all boards to be included in a broadcast group. For more information, see the ***Data broadcasting*** chapter in Section 5.

Extended control register (FE3A 0000)

Address	Function	Description
FE3A 0000-3	VMEBErr Timeout	Set VMEbus error timeout interval [1]
FE3A 0004	Slave window size	All installed memory VME-accessible (***default)*** [2]
FE3A 0005	"	8 MB slave window for 8 MB board
FE3A 0006	EPROM1 programming	Enable EPROM1 access by CPU-Y, normal operation (***default)***
FE3A 0007	"	Disable EPROM1 access by CPU-Y for Flash programming
FE3A 0008	Slave address [3]	Set Slave address bit A27 to 0 (binary) (***default)***
FE3A 0009	"	Set Slave address bit A27 to 1 (binary)
FE3A 000A	"	Set Slave address bit A28 to 0 (binary) (***default)***
FE3A 000B	"	Set Slave address bit A28 to 1 (binary)
FE3A 000C	"	Set Slave address bit A29 to 0 (binary) (***default)***
FE3A 000D	"	Set Slave address bit A29 to 1 (binary)
FE3A 000E	"	Set Slave address bit A30 to 0 (binary) (***default)***
FE3A 000F	"	Set Slave address bit A30 to 1

Notes: [1] For information on setting the timeout interval, see the ***System Controller*** chapter.
[2] To set address bits A22-A26, and A31, see the ***VME Slave interface*** *chapter.*
[3] On boards with 4MB of installed memory.

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

The V452 Series control register circuitry decodes only the address lines of a write access. As a result, the data that is actually written into the register **DOES NOT** matter. A write access using any data is all that is required.

> Note For more information, see the ***VME Slave interface*** and ***VME Master interface*** chapters in Section 5 and the ***CPU Watchdog*** chapter in Section 4.

Interrupt control registers

V452 Series boards provide control over interrupt handling by enabling and/or disabling reception of interrupts by the on-board CPU(s) via the four 16-bit Interrupt control registers at:

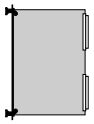
- **FE39 0000** (Interrupt control register #1)
- **FE39 4000** (Interrupt control register #2)
- **FE39 8000** (Interrupt control register #3)
- **FE39 C000** (Interrupt control register #4)

These registers provide a location to enable or disable reception of interrupts from on-board devices and all seven VMEbus interrupt levels.

> Note For directions on how to use this register, see the ***Interrupts*** chapter in this section.

The table below lists the function of all the address locations in the Interrupt control registers:

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



Interrupt control register

Interrupt source	Disable (default)	Enable
Serial Port A&B to CPU-X (2692)	FE39 0000	FE39 0001
Serial Port C&D to CPU-X (2692)	FE39 0002	FE39 0003
Timer A to CPU-X (2692)	FE39 0004	FE39 0005
Timer C to CPU-X (2692)	FE39 0006	FE39 0007
Daughter module A to CPU-X	FE39 0008	FE39 0009
Daughter module B to CPU-X	FE39 000A	FE39 000B
Mailbox to CPU-X	FE39 000C	FE39 000D
Counter to CPU-X (82C54)	FE39 000E	FE39 000F
Maskable Level 7 (parity, SysFail, or ACFail) to CPU-X	FE39 4000	FE39 4001
VME Level 1 to CPU-X	FE39 4002	FE39 4003
VME Level 2 to CPU-X	FE39 4004	FE39 4005
VME Level 3 to CPU-X	FE39 4006	FE39 4007
VME Level 4 to CPU-X	FE39 4008	FE39 4009
VME Level 5 to CPU-X	FE39 400A	FE39 400B
VME Level 6 to CPU-X	FE39 400C	FE39 400D
VME Level 7 to CPU-X	FE39 400E	FE39 400F
Serial Port A&B to CPU-Y (2692)	FE39 8000	FE39 8001
Serial Port C&D to CPU-Y (2692)	FE39 8002	FE39 8003
Timer A to CPU-Y (2692)	FE39 8004	FE39 8005
Timer C to CPU-Y (2692)	FE39 8006	FE39 8007
Daughter module A to CPU-Y	FE39 8008	FE39 8009
Daughter module B to CPU-Y	FE39 800A	FE39 800B
Mailbox to CPU-Y	FE39 800C	FE39 800D
Counter to CPU-Y (82C54)	FE39 800E	FE39 800F
Maskable Level 7 (parity, SysFail, or ACFail) to CPU-Y	FE39 C000	FE39 C001
VME Level 1 to CPU-Y	FE39 C002	FE39 C003
VME Level 2 to CPU-Y	FE39 C004	FE39 C005
VME Level 3 to CPU-Y	FE39 C006	FE39 C007
VME Level 4 to CPU-Y	FE39 C008	FE39 C009
VME Level 5 to CPU-Y	FE39 C00A	FE39 C00B
VME Level 6 to CPU-Y	FE39 C00C	FE39 C00D
VME Level 7 to CPU-Y	FE39 C00E	FE39 C00F

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

Slave interface control register

The 16-bit Slave Interface control register defines the characteristics of the VME Slave interface. Normally, the Slave interface is set up as part of the board's initialization program that runs at power-up or reset. However, it is also possible to alter the VME Slave Interface *on-the-fly*. The table below lists the locations in the Slave Interface control register and describes how each is used to configure the VME Slave Interface.

Slave Interface control register (FE38 8000)

Address	Function	Description
FE38 8000	A32	Select A32 VME extended addressing (***default)***
FE38 8001	A24	Select A24 addressing.
FE38 8002	Supervisory write accesses only	Allows write accesses only by Masters whose processors are operating in 68000 Supervisor mode (if Memory Protect is OFF) [1] ***(default)***
FE38 8003	No memory protect	Allows all VME Masters to perform write accesses (if Memory protect is OFF) [1]
FE38 8004	VME Slave address	Set Slave address bit A22 to 0 (***default)***
FE38 8005	"	Set Slave address bit A22 to 1
FE38 8006	"	Set Slave address bit A23 to 0 (***default)***
FE38 8007	"	Set Slave address bit A23 to 1
FE38 8008	"	Set Slave address bit A24 to 0 (***default)***
FE38 8009	"	Set Slave address bit A24 to 1
FE38 800A	"	Set Slave address bit A25 to 0 (***default)***
FE38 800B	"	Set Slave address bit A25 to 1
FE38 800C	"	Set Slave address bit A26 to 0 (***default)***
FE38 800D	"	Set Slave address bit A26 to 1
FE38 800E	"	Set Slave address bit A31 to 0 (***default)*** [2]
FE38 800F	"	Set Slave address bit A31 to 1 [2]

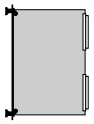
Notes: [1] Memory protect is turned ON/OFF via the *Primary Mode register* earlier in this chapter.
[2] To set address bits A27 - A30 see the *Secondary Control register* earlier in this chapter.

Configuration of the Slave interface is accomplished by performing write accesses to the Slave Interface control register. The circuitry for this register decodes only the address lines of all write accesses—the data that is actually written into the register **DOES NOT** matter.



For more information about the V452 Series VME Slave interface and how to use the Slave Interface control register, see the ***VME Slave interface*** chapter in Section 5.

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



Ethernet/VMEbus control registers

The Ethernet/VMEbus control register sets up the onboard Ethernet 10Base-T interface and the VMEbus request level in absence of JK17 configuration jumpers setting the bus request level. The table below lists functions and register addresses of the Ethernet/VMEbus control registers.

Ethernet/VMEbus control registers (FE3A 4000–F)

Function	Write to: (default)	Write to:
Ethernet Int. to CPU-X	FE3A 4000 Disable	FE3A 4001 Enable
Ethernet Int. to CPU-Y	FE3A 4002 Disable	FE3A 4003 Enable
Ethernet Jabber	FE3A 4004 Enable	FE3A 4005 Disable
Auto Polarity/Enhanced Squelch	FE3A 4006 Enh. Squelch	FE3A 4007 Auto Polarity
VMEbus Request Level*	FE3A 400A |— Level 3 FE3A 4008 |	FE3A 400A |— Level 2 FE3A 4009 |
		FE3A 400B |— Level 1 FE3A 4008 |
		FE3A 4009 |— Level 0 FE3A 400B |
– Reserved – Bits FE3A 400C–F		

* Note: Write to the indicated address pair for the bus request level shown.

Board information registers

In accordance with the VME64 extensions specification, the V452 SBC is provided with readable information registers that help in its initialization, test, and configuration when plugged into the VME backplane. The table below lists the register functions and corresponding address locations. These registers can be read using either longword or byte read accesses. The register data is described in the following paragraphs.

Board information registers, address & function

Register Address	Function
FE39 0001	Slot ID
FE39 0003	CPU & Board Type
FE39 4003	Memory size and installed options
FE39 8003	Modifications and ECO level
FE39 C003	PCB revision

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

Slot ID, FE39 0001

In this byte location, 6 bits are used to indicate the slot ID in which the board is installed (the 2 higher order bits are unused). The data is in the following format:

xxP43210

Where:

xx = unused bits

P = odd parity bit

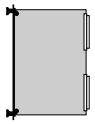
43210 = binary encoded (2^5–2^1) slot number (inverted bits)

Odd parity is used to identify when the Slot ID function is missing from the SBC. A binary value of 11 1111 or 0x3F indicates that the backplane is the old 3-row style and can't provide a geographical address. To deal with the unused higher order bits and extract valid data, the programmer must AND the register data with 0x3F. For convenience, the table below lists the slot positions and associated values (binary and hex) provided by the Slot ID register.

Slot ID register, slot numbers vs. register values

Slot No.	Binary Value	Hex Value	Slot No.	Binary Value	Hex Value
1	11 1110	3E	**12**	01 0011	13
2	11 1101	3D	**13**	11 0010	32
3	01 1100	1C	**14**	11 0001	31
4	11 1011	3B	**15**	01 0000	10
5	01 1010	1A	**16**	10 1111	2F
6	01 1001	19	**17**	00 1110	0E
7	11 1000	38	**18**	00 1101	0D
8	11 0111	37	**19**	10 1100	2C
9	01 0110	16	**20**	00 1011	0B
10	01 0101	15	**21**	10 1010	2A
11	11 0100	34	—	—	—

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



CPU & board type, FE39 0003

The byte data in this register forms a two hex digit code in the following format:

CB

Where:

C = CPU type code
- 4: 68040
- 6: 68060

B = Board type code
- 1: V451
- 2: V452

Memory size and installed options, FE39 4003

The byte data in this register is made up of one-digit hex code to indicate memory size and 4 bits to indicate installed options. The data is in the following format:

Merrf

Where:

M = Memory size code

0: none	5: 64MB	C: 2GB
1: 4MB	8: 128MB	
2: 8MB	9: 256MB	
3: 16MB	A: 512MB	
4: 32MB	B: 1GB	

e = Ethernet option bit
- 1: installed
- 0: not installed

r = Future option bits (2 ea. undefined, returns '0' value)

f = Flash option bit
- 1: installed
- 0: not installed

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

Modifications and ECO level, FE39 8003

The byte data in this register forms a two-digit hex code in the following format:

DE

Where:

D = Modification number code

0-F = 0-15

E = ECO number code

0-F = 0-15

PCB revision, FE39 C003

The byte data in this register forms a two-digit hex code in the following format:

rR

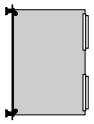
Where:

r = reserved (always '0')

R = PCB revision code

0-F = rev. A-F

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



Default & reset conditions

This chapter describes the hardware and programming considerations for V452 Series default conditions and reset functions.

Default conditions

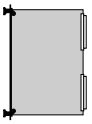
The table on the following page lists the default conditions present on several circuitry areas following a local reset of the V452 Series board.

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

Default conditions

Front panel LEDs	
Application LEDs (front panel)	On
Fail LED	On
On-board interrupts	
ABORT switch (Level 7 interrupt)	Enabled (non-maskable)
Status for all other Interrupt sources (Levels 7 - 2)	Disabled
68040 functions	
Internal cache contents	Indeterminate
Internal instruction & data caches	Disabled
Memory areas not subject to caching (transparent translation)	None
On-board DRAM	
Content of on-board RAM and RAM parity	Indeterminate
Memory parity checking	Disabled
CPU Watchdog	
CPU Watchdog Halt monitor	Disabled
CPU Watchdog Run monitor	Disabled
Serial ports B & D	Disabled
EPROM1 access by CPU-Y	Enabled
VME Slave interface	
Slave interface status	Disabled
Slave memory write access	Disabled
Slave memory write access protection level	Supervisor-only
Address width	A32
Window size	16 MB
Base Address	0 MB (x0000 0000)
Lower vs. Upper VME A32/D32 address range	Lower (Top of DRAM to 0x3FFF FFFF)
VME Master interface	
VMEbus release configuration	Release on Request (ROR)
VMEbus request configuration	Not FAIR requests
Bus request level	3 (with no jumper config.)
VME interrupts	
VME Interrupter status	Reset
VME interrupt level asserted by the VME interrupter	Level 7
VME interrupt vector asserted by the VME interrupter	Indeterminate
VME Interrupt handler level	None
VME SysRes signal	
VME SysRes (reset) reception respond/ignore	Respond
VME SysFail signal	
VME SysFail interrupt reception (via 2692 DUART)	Disabled
VME SysFail\ signal	Asserted
2692 control registers	
Serial ports B and D	Disabled
VME interrupter level	Level 7
SysFail reception	Off

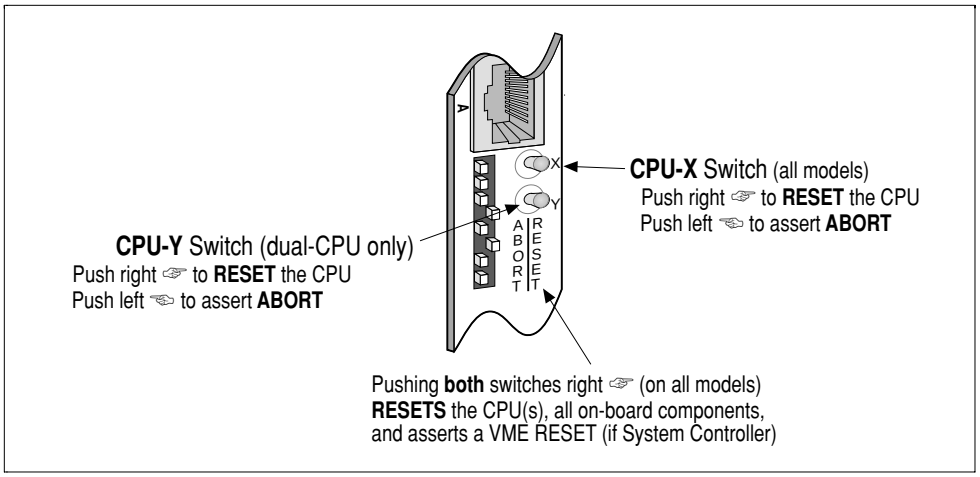
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Reset sources

The following sources can assert the Local Reset line:

- **Power cycling circuitry** — special circuitry asserts a local reset when the board is power-cycled. A power-up reset lasts at least 250 ms as required by the VMEbus.
- **Front-panel RESET toggle switch** — The V452 Series front panel RESET toggle switches can reset an individual CPU chip or assert a board-level local reset as shown below.



RESET switch operation

- **VME SysRes\ line** — when a remote device asserts the VME System Reset (SysRes\) it also asserts the Local Reset line.

  The V452 Series VMEbus interrupter circuitry can be used to assert the VMEbus Reset signal to cause a local reset of each board on the VMEbus.

  ⚠ To assert the VMEbus System Reset signal for a long enough period to ensure that all boards on the bus are reset, the V452 Series board cannot respond to this signal itself. Thus, before a V452 Series board can be used to assert the VMEbus System Reset signal, the board must also be configured to ignore the VMEbus System Reset signal itself. For information about setting up the board to ignore the VME System Reset signal, see the *VMEbus Slave interface* chapter in the ***Interface Options*** section.

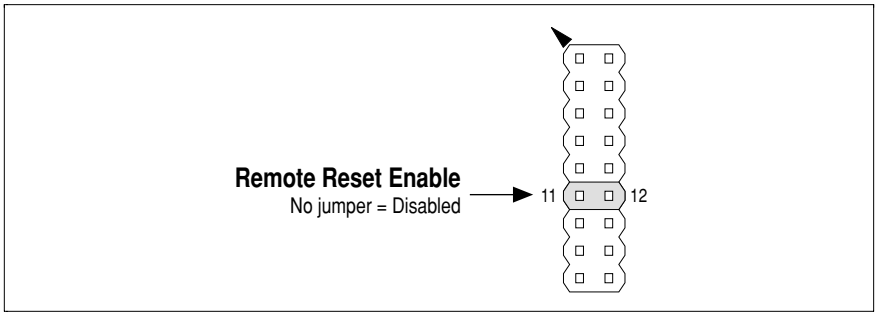
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- **CPU Watchdog** — can assert a board-level local reset whenever either one or both of the following two conditions are present:
  - If the CPU(s) fails to address the watchdog chip at least once every 600 ms.
  - If the CPU(s) halts.

  The V452 Series allows you to program the Watchdog timer to watch for both conditions or one and not the other if desired.

  For more information see the chapter on the ***CPU Watchdog*** chapter in Section 4.
- **Processor instruction reset** — executing a 68040 software RESET instruction, directs the CPU to assert the Local Reset line. However, a 68040 RESET instruction only resets external devices, it does not reset the CPU itself.
- **Slave Remote Reset** — another VMEbus Master can reset a specific V452 Series board by writing to the board's Slave reset registers. The second to last 256 byte region of on-board RAM acts as a reset register. A Supervisor mode write to byte zero of this area (Slave Base Address + 3FFEX0 on a 4 MB card) by another VMEbus Master resets the V452 Series board.

  The remote slave reset feature can be enabled or disabled via jumper **JK17** as shown in the figure below. A shunt between JK17 pins **11** and **12** enables Remote Reset, while removing the shunt disables Remote Reset.

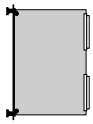


Enabling/disabling Slave Remote reset via JK17

  For more information on setting the VME Slave address for use with the Remote Reset option, see the **VME Slave interface** chapter in Section 5.
- **EZ-bus reset** — an installed daughter module can assert a reset via the V452 Series EZ-bus connector.

  For more information see the ***EZ-bus interface*** chapter in the ***Interface Options*** section.

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



- **Software reset** — with software, a reset can be performed in one of three ways: 1) VME level 0 interrupt, 2) direct watchdog timer enable, and 3) writing to remote reset register.

Reset sequence

The paragraphs below describe the V452 Series reset sequence:

Power-up reset considerations

Upon power-up, special circuitry, via the Dallas 1232, asserts the board's on-board local reset line. The power-up reset lasts for at least 250 ms as required by the CPU and VMEbus. Certain on-board devices have their hardware reset pins connected directly to the local reset line. For those devices, the assertion of the local reset line will initialize each device to a known internal state from which the CPU can easily restart the system. All on-board devices and control registers are directly connected to the local reset line.

Default conditions after a local reset

Asserting the local reset line, via one of the methods listed above, causes a board-level local reset that clears all the control registers in the CPU and on-the V452 Series board. The table on the next page lists the default conditions for V452 Series boards after a local reset and before system initialization by the system boot software/firmware.

Entering boot state after a local reset

After power-cycling or when the local reset line is asserted and then released by one of the sources listed previously, the board enters a condition called boot state. In boot state, the CPU begins a special type of exception processing. It internally generates exception vector numbers 0 and 1 and attempts to fetch its initial interrupt stack pointer and program counter vector from address locations 0x0 and 0x4.

Because this behavior occurs automatically upon reset, the reset circuitry temporarily alters the board's address map while in boot state. In boot state, either the 8-bit wide EPROM or the onboard Flash appears everywhere in memory as determined by the EPROM boot enable jumper at JK17 pins 17 and 18 (on = EPROM, off = Flash).

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

This relocation allows the CPU(s) to fetch reset vectors, the exception vector table, and its start-up code from the EPROM or onboard Flash as required.

Start-up vectors

To exit reset state, the CPU must first read the reset vector at 0x0 and 0x4 as described above. The reset vector consists of a long word for the initial stack pointer (SP) address and a long word for the initial Program Counter (PC) in that order. The CPU must then fetch code from the regular EPROM or Flash address space (FE000000 or FC000000 respectively).

If EPROM boot is **enabled**:

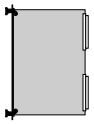
- CPU-X executes from **EPROM0** at **FE000000–FE0FFFFF**
- CPU-Y (dual-CPU only) executes from **EPROM1** at **FE400000–FE4FFFFF** (old space) or from Flash module at **FD000000–FDFFFFFF**

Except for the first 3 fetches in boot state in which CPU-X fetches from EPROM0 and CPU-Y (if present) fetches from EPROM1, either CPU can execute from either or both of the EPROMs.

For a single CPU board, both EPROM sockets can contain code for the single CPU.

If EPROM boot is **disabled**, both CPUs execute from onboard Flash at FC000000–FC7FFFFF.

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



Reset via software

There are three ways to perform a reset via software:

- VME Level 0 interrupt
- Watchdog timer enable
- Remote reset register

VME Level 0 interrupt reset

By setting the board's interrupter to produce a Level "0" interrupt, the board will produce a VME system reset (SysRes) upon generation of the interrupt. This will reset all boards in the system that are configured to respond to the VME SysRes signal. For more information on the VME interrupter, refer to the *Configuring the VMEbus interrupter/resetter* discussion in the ***Interrupts*** chapter in Section 3.

To set the interrupter to Level 0, execute the following 680x0 assembler commands:

```
moveb    #0x00, 0xFE28003F        /* set desired level bits */
moveb    #0x0E, 0xFE28003B        /* clear the other bits */
```

To generate the interrupt, write the following to the Primary Mode register:

```
moveb    #0x07, 0xFE380003        /* reset interrupter */
moveb    #0x0F, 0xFE380003        /* invoke interrupter */
```

After setting the interrupter to Level "0", executing these instructions will cause the board to reset itself and all other boards in the system that are configured to respond to the SysRes signal.



It is recommended that the board asserting the VME SysRes be set up to ***ignore*** the VME SysRes signal to ensure a long enough period for all boards to be reset. Refer to the discussion below.

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

Setting up the board to respond to/ignore VME SysRes

V4xx boards can be configured to either respond to or ignore the VME SysRes signal. This lets you chose the boards that will reset with SysRes.

The VME SysRes respond to/ignore function is set in the Primary Control register, location 0xFE38C000. Refer to the ***Control registers*** discussion in the ***V452 internal registers*** chapter in Section 3 for more information on control registers.

To configure the board's behavior to SysRes, write anything to these register locations:

- To respond, write to: **0xFE38C006**
- To ignore, write to: **0xFE38C007**

The assembly code below shows how this is done. The default after a board reset is for the board to respond to SysRes.

```
moveb    #0x00, 0xFE380006        /* respond to SysRes */
moveb    #0x00, 0xFE380007        /* ignore SysRes */
```

Watchdog timer enable

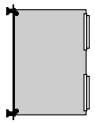
Software can also reset a board by enabling the board's watchdog timer. After a ***single*** enable, the watchdog timer times out after approximately 600 ms to 1.2 sec. The time out causes the board to reset. This method of reset is generally suited for resetting a specific, non-system controller board. If the target board is the system controller, then the entire system is reset.

The watchdog timer function is controlled in the Extended Mode register, location 0xFE384003. Writing 0x0C to this location enables the watchdog. Refer to the ***Mode registers*** discussion in the V452 internal registers chapter for more information on mode registers. Refer also to the ***CPU Watchdog*** chapter in the ***Local Components*** section for more information on programming the watchdog timer.

Shown below is the watchdog enable assembler instruction:

```
moveb    #0x0C, 0xFE384003        /* enable watchdog */
```


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



Remote reset register

V4xx boards have a remote reset register that resides at the top of memory just below the mailbox area: *xxx*FFE00–*xxx*FFEFF. Writing any value to this register resets the board. This method of reset is also generally suited for resetting a specific, non-system controller board. If the target board is the system controller, then the entire system is reset. For remote reset to work, a jumper must be installed at JK17 pins 11 and 12. Refer to the *Enabling/disabling VME Slave remote reset* discussion in the ***VME Slave interface*** chapter in Section 5.

A particular board's remote reset address is the slave address plus the remote reset register address as determined by the amount of onboard RAM:

- 0x003FFE00 (4 MB)
- 0x007FFE00 (8 MB)
- 0x00FFFE00 (16 MB)
- 0x01FFFE00 (32 MB)
- 0x03FFFE00 (64 MB)
- 0x07FFFE00 (128 MB)
- 0x0FFF FE00 (256 MB, *special factory order*)
- 0x1FFF FE00 (512 MB, *special factory order*)

For example, for a 32MB V4xx board with a slave address of 0x4000000, the remote reset address is 0x05FFFE00 (0x01FFFE00 + 0x04000000).

Shown below is the instruction for writing to our example board's remote reset register. In this example, a zero byte is written, but any value will suffice.

```
movel     #0x00, 0x05FFFE00          /* write zero to remote reset  reg */
```


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

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

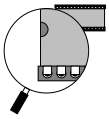


Local Components 4

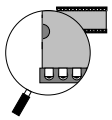
This section describes the primary components and supporting circuitry responsible for local operations on the V452 Series boards.

- 68040 CPU
- 68060 CPU
- CPU Mailbox
- CPU Watchdog
- Dynamic RAM
- EPROM
- Flash memory module
- Onboard Flash memory
- Timers & counters
- Clock calendar
- Non-volatile 8K x 8 SRAM

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



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



68040 CPU

Introduction

The Motorola® MC68040 is a third generation, 68000-compatible, high-performance, 32-bit microprocessor. The 68040 is a virtual memory microprocessor employing multiple, concurrent execution units. The 68040's highly-integrated architecture includes an 68030-compatible integer unit, an IEEE 754-compatible floating point unit (FPU), fully independent instruction and data demand-paged memory management units (MMU's), and independent 4KB instruction and data caches on a single chip. The main features of the 68040 include:

- 20 MIPS integer performance,
- 3.5 MFLOPS floating-point performance,
- IEEE 754-compatible FPU,
- Independent instruction and data MMUs,
- 4KB physical instruction cache and 4KB physical data cache accessed simultaneously,
- 32-Bit, non-multiplexed external address and data buses with synchronous interface,
- User-object code compatibility with all other M68000 microprocessors,
- Concurrent integer unit, FPU, MMU, bus controller, and bus snooper maximize throughput,
- 4 GB direct addressing range.

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

Additional 68040 documentation

This chapter introduces some of the primary features of the 68040 and describes specific considerations that apply to the use of the 68040 on V452 Series boards. It is not meant to serve as a complete guide for programming the 68040.

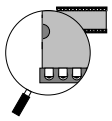
For detailed information, see the following Motorola publications (the major topics covered in each manual are listed below the title):

- *MC68040UM/AD, MC68040 User's Manual.* — describes the capabilities, operation, and programming of the 68040:
  - Programming model
  - Data organization and addressing capabilities
  - Instructions set
  - Signal description
  - Memory management
  - Instruction and data caches
  - Bus operation
  - Exception processing
  - Instruction execution processing
  - Electrical characteristics

- *MC68040PM/AD, MC68000 Programmer's Reference Manual* — describes the software instructions used by the processors in the 68000 family:

  - Integer instructions
  - Floating-point instructions
  - Supervisor (privileged) instructions
  - CPU32 instruction summary
  - Instruction format summary
  - Processor instruction summary

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



These publications are available from Motorola at:

Motorola Semiconductor Products Sector
Literature Distribution Center
P.O. Box 20924
Phoenix, Arizona 85036-0924

(800) 521-6274
(520) 994-6561

For information on technical training, contact:

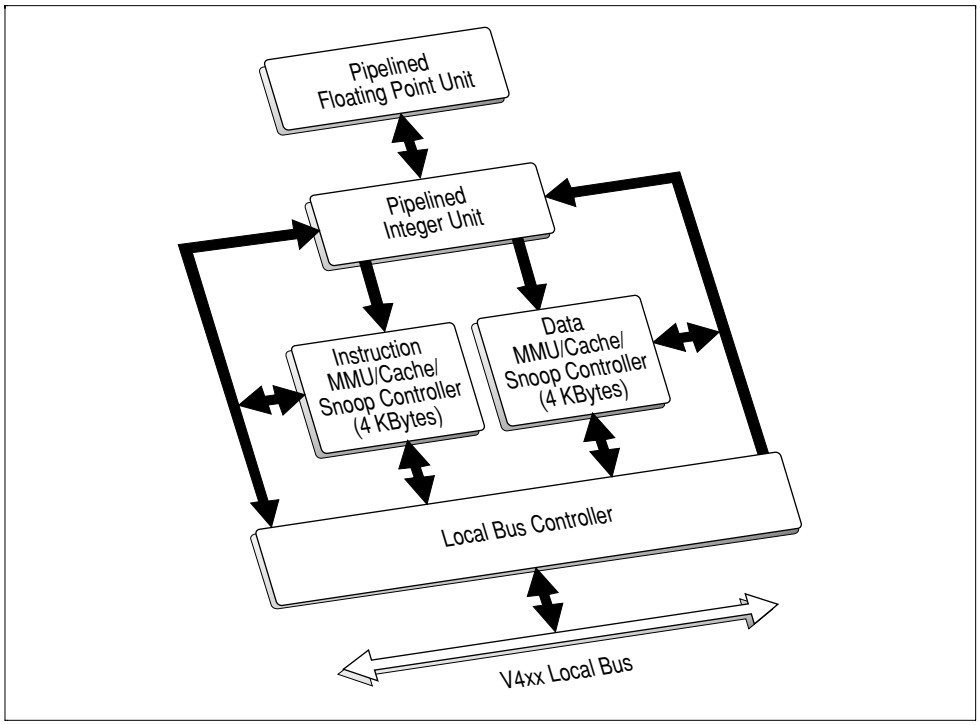
Motorola Semiconductor Products Sector
World Marketing Training Operations EL524
P.O. Box 21007
Phoenix, Arizona 85036-0924

(800) 521-6274 (ask for “Training”)
(520) 897-3665

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

Introduction

The 68040 is an enhanced, 32-bit, HCMOS microprocessor that combines the integer unit processing capabilities of the MC68030 microprocessor with independent 4KB data and instruction caches and on-chip FPU.

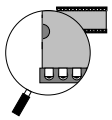


Simplified 68040 block diagram

The 68040 maintains the 32-bit registers available with the entire M68000 Family as well as the 32-bit address and data paths, a rich instruction set, and versatile addressing modes. Instruction execution proceeds in parallel with accesses to the internal caches, MMU operations, and bus controller activity. Additionally, the integer unit is optimized for high-level language environments.

The 68040 FPU is user-object-code compatible with the MC68882 floating-point coprocessor and conforms to the ANSI/IEEE Standard 754 for Binary Floating-Point Arithmetic. The FPU has been optimized to execute the most commonly used subset of the MC68882 instruction set, and includes additional instruction formats for single-and double-precision rounding of results. Floating-point instructions in the FPU execute concurrently with integer instructions in the integer unit.

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



The MMUs support multi-processing, virtual memory systems by translating logical addresses to physical addresses using translation tables stored in memory. The MMUs store recently used address mappings in two, separate, on-chip ATCs. When an ATC contains the physical address for a bus cycle requested by the processor, a translation table search is avoided, and the physical address is supplied immediately, incurring no delay for address translation. Each MMU has two transparent translation registers available that define a one-to-one mapping for address space segments ranging in size from 16 MB to 4 GB each.

Each MMU provides read-only and supervisor-only protection on a page basis. Also, processes can be given isolated address spaces by assigning each a unique table structure and updating the root pointer upon a task swap. Isolated address spaces protect the integrity of independent processes.

The instruction and data caches operate independently from the rest of the machine, storing information for fast access by the execution units. Each cache resides on its own internal address bus and data bus, allowing simultaneous access to both. The data cache provides write-through or copy-back write modes that can be configured on a page-by-page basis.

The 68040 bus controller supports a high-speed, non-multiplexed, synchronous external bus interface, which allows the following transfer sizes: byte, word (2 bytes),long word (4 bytes), and line (16 bytes). Line accesses are performed using burst transfers for both reads and writes to provide high data transfer rates.

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

Programming model

The 68040 integrates the functions of the integer unit, MMU, and FPU. The registers depicted in the programming model provide access and control for the three units. The registers are partitioned into two levels of privilege: user and supervisor. User programs, executing in the user mode, can only use the resources of the user model. System software, executing in the supervisor mode, has unrestricted access to all processor resources.

The integer portion of the user programming model, consisting of 16, general purpose, 32-bit registers and two control registers, is the same as the user programming model of the MC68030. The 68040 user programming model also incorporates the MC68882 programming model consisting of eight, floating-point, 80-bit data registers, a floating-point control register, a floating-point status register, and a floating-point instruction address register.

The supervisor programming model is used exclusively by 68040 system programmers to implement operating system functions, I/O control, and memory management subsystems. This supervisor/user distinction in the M68000 architecture was carefully planned so that all application software can be written to execute in the non-privileged user mode and migrate to the 68040 from any M68000 platform without modification. Since system software is usually modified by system designers when porting to a new design, the control features are properly placed in the supervisor programming model.

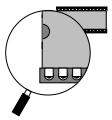
For example, the transparent translation registers of the 68040 can only be read or written by the supervisor software; programming resources of user application programs are unaffected by the existence of the transparent translation registers.

68040 registers

The paragraphs below describe the 68040's internal registers:

D0-D7 these **data registers** contain operands for bit and bit field (1 to 32 bits), byte (8 bit), word (16 bit), long-word (32 bit), and quad word (64 bit) operations.

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



A0-A7 Registers A0-A6 and the stack pointer registers (user, interrupt, and master) are address registers that may be used as software stack pointers or base address registers. Register A7 is the user stack pointer in user mode and is either the interrupt or master stack pointer (A7' or A7") in supervisor mode. In supervisor mode, the active stack pointer (interrupt or master) is selected based on a bit in the status register (SR). The address registers can be used for word and long-word operations, and all 16 general-purpose registers (D0-D7, A0-A7) can be used as index registers.

FP0-FP7 **Floating-point data registers** — these eight, 80 bit registers are analogous to the integer data registers (D0-D7) in all M68000 family processors. Floating-point data registers always contain extended-precision numbers. All external operands, regardless of the data format, are converted to extended-precision values before being used in any floating-point calculation or stored in a floating-point data register.

PC **Program counter** usually contains the address of the instruction being executed by the 68040. During instruction execution and exception processing, the processor automatically increments the contents of the PC or places a new value in the PC, as appropriate.

SR **Status register** in the supervisor programming model contains the condition codes that reflect the results of a previous operation and can be used for conditional instruction execution of a program.

CCR **Condition Code register** — The lower byte of the SR is accessible in user mode as the condition code register. Access to the upper byte of the SR is restricted to the supervisor mode.

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

VBR — **Vector base register** — as part of exception processing, the vector number of the exception provides an index into the exception vector table. The base address of the exception vector table is stored in the vector base register. The displacement of an exception vector is added to the value in the VBR when the 68040 accesses the vector table during exception processing.

SFC / DFC — **Source and destination function code registers** — contain 3-bit function codes. Function codes can be considered extensions of the 32-bit linear address. Function codes are automatically generated by the processor to select address spaces for data and program accesses at the user and supervisor modes. The alternate function code registers are used by certain instructions to explicitly specify the function codes for various operations.

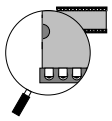
CACR — **Cache control register** — controls enabling of the on-chip instruction and data caches of the 68040.

SRP / URP — **Supervisor and user root pointer registers** — point to the root of the address translation table tree to be used for supervisor and/or user accesses. The URP is used if FC2 of the logical address is zero, and the SRP is used if FC2 is one.

TC — **Translation control register** — enables logical-to-physical address translation and selects either 4K or 8K page sizes. As shown in Figure 2, there are four transparent translation registers–ITT0 and ITT1 for instruction accesses and DTT0 and DTT1 for data accesses. These registers allow portions of the logical address space to be transparently mapped and accessed without the use of resident descriptors in an ATC.

MMUSR — **MMU status register** — contains status information from the execution of a PTEST instruction. The PTEST instruction searches the translation tables for the logical address as specified by this instruction's effective address field and the DFC.

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



FPCR **Floating -point control register** — this 32-bit register contains an exception enable byte that enables/disables traps for each class of floating-point exceptions and a mode byte that sets the user-selectable modes. The FPCR can be read or written to by the user and is cleared by a hardware reset or a restore operation of the null state. When cleared, the FPCR provides the IEEE 754 standard defaults.

FPSR **Floating-point status register** — contains a condition code byte, quotient bits, an exception status byte, and an accrued exception byte. All bits in the FPSR can be read or written by the user. Execution of most floating-point instructions modifies this register.

FPIAR **Floating-point instruction address register** — for the subset of the FPU instructions that generate exception traps, this 32-bit register is loaded with the logical address of an instruction before it is executed. This address can then be used by a floating-point exception handler to locate a floating-point instruction that has caused an exception. The move floating-point data register (FMOVE) instruction (to/from the FPCR,FPSR, or FPIAR) and the move multiple data registers (FMOVEM) instruction cannot generate floating-point exceptions; therefore, these instructions do not modify the FPIAR. Thus, the FMOVE and FMOVEM instructions can be used to read the FPIAR in the trap handler without changing the previous value.

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

Data types

The 68040 supports the basic data types listed in the table below. Some data types apply only to the integer unit, some only to the FPU, and some to both the integer unit and the FPU. In addition, the instruction set supports operations on other data types such as memory addresses.

68040 data types

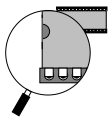
Operand data type	Size	Support	Notes
Bit	1 Bit	IU	—
Bit Field	1-32 Bits	IU	Field of Consecutive Bit
BCD	8 Bits	IU	Packed: 2 Digits/Byte - Unpacked: 1 Digit/Byte
Byte Integer	8 Bits	IU,FPU	—
Word Integer	16 Bits	IU,FPU	—
Long-Word Integer	32 Bits	IU,FPU	—
Quad-Word Integer	64 Bits	IU	Any Two Data Registers
16-Byte	128 Bits	IU	Memory-Only,Aligned to 16-Byte Boundary
Single-Precision Real	32 Bits	FPU	1-Bit Sign, 8 Bit Exponent, 23-Bit Mantissa
Double-Precision Real	64 Bits	FPU	1-Bit Sign, 11 Bit Exponent, 52-Bit Mantissa
Extended-Precision Real	80 Bits	FPU	1-Bit Sign, 15-Bit Exponent, 64-Bit Mantissa

The three integer data formats common to both the integer unit and the FPU (byte, word, and long word) are the standard twos-complement data formats defined in the M68000 Family architecture. Whenever an integer is used in a floating-point operation, the integer is automatically converted by the FPU to an extended-precision floating-point number before being used. The ability to effectively use integers in floating-point operations saves user memory because an integer representation of a number usually requires fewer bits than the equivalent floating-point representation.

Single-and double-precision floating-point data formats are implemented in the FPU as defined by the IEEE 754 standard. These data formats are the main floating-point formats and should be used for most calculations involving real numbers.

The extended-precision data format is also in conformance with the IEEE 754 standard, but the standard does not specify this format to the bit level as it does for a single and double precision. The memory format for the FPU consists of 96 bits (three long words). Only 80 bits are actually used; the other 16 bits are reserved for future use and for long-word alignment of the floating-point data structures in memory. The extended-precision format has a 15-bit exponent, a 64-bit mantissa, and a 1-bit mantissa sign. Extended-precision numbers are intended for use as temporary variables, intermediate values, or where extra precision is needed.

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



Address modes

The 68040 addressing modes are listed in the table below. The register indirect addressing modes support post-increment, pre-decrement, offset, and indexing, which are particularly useful for handling data structures common to sophisticated applications and high-level languages.

Addressing modes

Modes	Syntax
Register Direct	
Data Register Direct	Dn
Address Register Direct	An
Register Indirect	
Address Register Indirect	(An)
Address Register Indirect with Post-increment	(An)+
Address Register Indirect with Pre-decrement	-(An)
Address Register Indirect with Displacement	(d_{16},An)
Register Indirect with Index	
Address Register Indirect with Index (8-Bit Displacement)	(dg,An,Xn)
Address Register Indirect with Index (Base Displacement)	(bd,An,Xn)
Memory Indirect	
Memory Indirect Post-indexed	([bd,An],Xn,od)
Memory Indirect Pre-indexed	([bd,An,Xn],od)
Program Counter Indirect with Displacement	(d_{16},PC)
Program Counter Indirect with Index	
PC Indirect with Index (8-Bit Displacement)	(dg.PC,Xn)
PC Indirect with Index (Base Displacement)	(bd,PC,Xn)
Program Counter Memory Indirect	
PC Memory Indirect Post-indexed	([bd,PC],Xn,od)
PC Memory Indirect Pre-indexed	([bd,PC,Xn],od)
Absolute	
Absolute Short	xxx.W
Absolute Long	xxx.L
Immediate	#<data>

Notes:

Dn= Data Register,D7-D0

An= Address Register, A7-A0

dg,d_{16}= A twos-complement or sign extended displacement; added as part of the effective address calculation; size is 8 (dg) or 16 (d16) bits; when omitted, assemblers use a value of zero.

Xn= Address or data register used as an index register; form is Xn.SIZE/SCALE, where SIZE is .W or .L (indicates index register size) and SCALE is 1, 2, 4, or 8 (index register is multiplied by SCALE); use of SIZE and/or SCALE is optional.

bd= A twos complement base displacement; when present, size can be 16 or 32 bits.

od= Outer displacement, added as part of effective address calculation after any memory indirection; use is optional with size of 16 or 32 bits.

PC= Program Counter

<data>= Immediate value of 8, 16, or 32 bits.

( )= Effective Address

[ ]= Used as indirect access to long-word address.

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

The program counter indirect mode also has indexing and offset capabilities; this addressing mode is typically required to support position-independent software. In addition to these addressing modes, the 68040 provides index sizing and scaling features that enhance software performance. Data formats are supported orthogonally by all arithmetic operations and by all appropriate addressing modes.

Instruction set overview

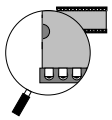
The instructions provided by the 68040 are listed in Table 3. The instruction set has been tailored to support high-level languages and is optimized for those instructions most commonly executed (however, all instructions listed are fully supported). Many instructions operate on bytes, words, and long-words, and most instructions can use any of the addressing modes of Table 2.

The 68040 floating-point instructions, a commonly used subset of the 68882 instruction set, are implemented in hardware. The remaining unimplemented instructions are less frequently used and are efficiently emulated in software, maintaining compatibility with the 68881/68882 floating-point coprocessors.

The 68040 instruction set includes MOVE16, a new user instruction that allows high-speed transfers of 16-byte blocks between external devices, such as memory to memory or coprocessor to memory.

For detailed information on the 68040 instruction set, refer to *M68000 PM/AD, M6800 Programmer's Reference Manual*.

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



68040 instruction set

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
ADDA	Add Address
ADDI	Add Immediate
ADDQ	Add Quick
ADDX	Add with Extend
AND	Logical AND
ANDI	Logical AND Immediate
ASL, ASR	Arithmetic Shift Left and Right
Bcc	Branch Conditionally
BCHG	Test Bit and Change
BCLR	Test Bit and Clear
BFCHG	Test Bit Field and Change
BFCLR	Test Bit Field and Clear
BFEXTS	Signed Bit Field Extract
BFEXTU	Unsigned Bit Field Extract
BFFFO	Bit Field Find First One
BFINS	Bit Field Insert
BFSET	Test Bit Field and Set
BFTST	Test Bit Field
BKPT	Breakpoint
BRA	Branch
BSET	Test Bit and Set
BSR	Branch to Subroutine
BTST	Test Bit
CAS	Compare and Swap Operands
CAS2	Compare and Swap Dual Operands
CHK	Check Register Against Bounds
CHK2	Check Register Against Upper and Lower Bounds
**CINV*	*Invalidate Cache Entries*
CLR	Clear
CMP	Compare
CMPA	Compare Address
CMPI	Compare Immediate
CMPM	Compare Memory to Memory
CMP2	Compare Register Against Upper and Lower Bounds
**CPUSH*	*Push then Invalidate Cache Entries*
DBcc	Test Condition, Decrement and Branch
DIVS,DIVSL	Signed Divide
DIVU, DIVUL	Unsigned Divide
EOR	Logical Exclusive OR
EORI	Logical Exclusive OR Immediate
EXG	Exchange Registers
EXT, EXTB	Sign Extend
**FABS*	*Floating-Point Absolute Value*
**FADD*	*Floating-Point Add*
FBcc	Floating-Point Branch
FCMP	Floating-Point Compare
FDBcc	Floating-Point Decrement and Branch
**FDIV*	*Floating-Point Divide*
**FMOVE*	*Move Floating-Point Register*
FMOVEM	Move Multiple Floating-Point Registers
**FMUL*	*Floating-Point Multiply*
**FNEG*	*Floating-Point Negative*
FRESTORE	Restore Floating-Point Internal State
FSAVE	Save Floating-Point Internal State
FScc	Floating-Point Set According to Condition
**FSQRT*	*Floating-Point Square Root*
**FSUB*	*Floating-Point Subtract*
FTRAPcc	Floating-Point Trap-On Condition
FTST	Floating-Point Test
ILLEGAL	Take Illegal Instruction Trap
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link and Allocate
LSL,LSR	Logical Shift Left and Right
MOVE	Move
**MOVE16*	*16-Byte Block Move*
MOVEA	Move Address
MOVE CCR	Move Condition Code Register
MOVE SR	Move Status Register
MOVE USP	Move User Stack Pointer
**MOVEC*	*Move Control Register*
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral
MOVEQ	Move Quick
**MOVES*	*Move Alternate Address Space*
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NEGX	Negate with Extend
NOP	No Operation
NOT	Logical Complement
OR	Logical Inclusive OR
ORI	Logical Inclusive OR Immediate
PACK	Pack BCD
**PEA*	*Push Effective Address*
**PFLUSH*	*Flush Entry(ies) in the ATCs*
**PTEST*	*Test a Logical Address*
RESET	Reset External Devices
ROL,ROR	Rotate Left and Right
ROXL,RORX	Rotate with Extend Left and Right
RTD	Return and Deallocate
RTE	Return from Exception
RTR	Return and Restore Codes
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Scc	Set Conditionally
STOP	Stop
SUB	Subtract
SUBA	Subtract Address
SUBI	Subtract Immediate
SUBQ	Subtract Quick
SUBX	Subtract with Extend
SWAP	Swap Register Words
TAS	Test Operand and Set
TRAP	Trap
TRAPcc	Trap Conditionally
TRAPV	Trap on Overflow
TST	Test Operand
UNLK	Unlink
UNPK	Unpack BCD

Note: * *New 68040-only instructions.*

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

Exception processing

The 68040 provides the same extensions to the exception stacking process as the MC68030. If the M bit in the SR is set, the master stack pointer is used for all task-related exceptions. When a nontask-related exception occurs (i.e., an interrupt), the M bit is cleared, and the interrupt stack pointer is used. This feature allows a task's area to be carried within a single processor control block, and new tasks can be initiated by simply reloading the master stack pointer and setting the M bit.

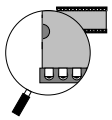
The externally generated exceptions are interrupts, bus errors, and reset conditions. The interrupts are requests from external devices for processor action; whereas, the bus error and reset signals are used for access control and processor initialization. The internally generated exceptions come from instructions, address errors, tracing, or break points.

The TRAP, TRAPcc, TRAPVcc, FTRAPcc, CHK, CHK2, and DIV instructions can all generate exceptions as part of their instruction execution. Tracing behaves like a very high-priority, internally-generated interrupt whenever it is processed. The other internally-generated exceptions are caused by unimplemented floating-point instructions, illegal instructions, instruction fetches from odd addresses, and privilege violations and for invalid descriptors encountered during table searches.

Exception processing for the 68040 occurs in the following sequence:

❶ an internal copy is made of the SR,
❷ the vector number of the exception is determined,
❸ current processor status is saved, and
❹ the exception vector offset is determined by multiplying the vector number by four. This offset is then added to the contents of the VBR to determine the memory address of the exception vector. The instruction at the address given in the exception vector is fetched, and normal instruction decoding and execution is started.

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



Exception vector assignments (V452 Series boards and the 68040)

Vector number (dec)	Vector number (hex)	Vector address (hex)	Offset space	Assignment	Status
0	00	000	SP	Reset Initial Interrupt Stack Pointer	---
1	01	004	SP	Reset Initial Program Counter	---
2	02	008	SD	Bus Error	Yes
3	03	00C	SD	Address Error	Yes
4	04	010	SD	Illegal Instruction	No
5	05	014	SD	Integer divide by zero	No
6	06	018	SD	CHK, CHK2 Instruction	No
7	07	01C	SD	cpTRAPcc, TRAPcc, TRAPV Instructions	No
8	08	020	SD	Privilege Violation	No
9	09	024	SD	Trace	Yes
10	0A	028	SD	Line1010 Emulator	No
11	0B	02C	SD	Line 1111 Emulator	No
12	0C	030	SD	— Unassigned, reserved —	
13	0D	034	SD	— Unassigned, reserved — ***	No
14	0E	038	SD	Format Error	No
15	0F	03C	SD	Uninitialized Interrupt	Yes
16 - 23	10-17	40 - 5C	SD	Unassigned, **(EZ-bus modules) ****	
24	18	060	SD	Spurious Interrupt	Yes
25	19	064	SD	**On-board** Level 1 Interrupt **	Yes
26	1A	068	SD	**On-board** Level 2 Interrupt ** **(counters; 82C54)**	Yes
27	1B	06C	SD	**On-board** Level 3 Interrupt ** **(CPU Mailboxes)**	Yes
28	1C	070	SD	**On-board** Level 4 Interrupt ** **(EZ-bus modules)**	Yes
29	1D	074	SD	**On-board** Level 5 Interrupt ** **(Async. serial interface; 2692 UART)**	Yes
30	1E	078	SD	**On-board** Level 6 Interrupt ** **(timers; 2692 UART)**	Yes
31	1F	07C	SD	**On-board** Level 7 Interrupt ** **(ABORT, Bad parity, ACFail, SysFail)**	Yes
32 - 47	20-2F	80 - BC	SD	TRAP #0-15 Instruction Vectors	No
48	30	0C0	SD	FP Branch or Set on Unordered Condition	
49	31	0C4	SD	FP Inexact Result	No
50	32	0C8	SD	FP Divide by Zero	No
51	33	0CC	SD	FP Underflow	No
52	34	0D0	SD	FP Operand Error	No
53	35	0D4	SD	FP Overflow	No
54	36	0D8	SD	FP Signaling NAN	No
55	37	0DC	SD	— FP Unimplemented data type — *	No
56	38	0E0	SD	— Unassigned, reserved — ***	No
57 - 63	39 - 3D	E4 - F6	SD	— Unassigned, reserved —	
64 - 255	40-FF	100-3FC	SD	User-defined Vectors (192) **(EZ-bus modules)**	

Notes: SP = Supervisor Program Space;
SD = Supervisor Data Space
* Assignment is different than that used on the 68030.
** Assignment is particular to the V452 Series boards.
*** Exception is assigned for 68020 and 68030 processors, but not the 68040.
On-board interrupts appear in **bold face**.
The user-defined vectors (64 to 255) may be assigned to the EZ-bus (for daughter modules) if desired.

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

Instruction and data caches

Studies have shown that typical programs spend much of their execution time in a few main routines or tight loops. Earlier members of the 68000 Family took advantage of this locality-of-reference phenomenon to varying degrees. The 68040 takes further advantage of cache technology with its two, independent, on-chip, physical address space caches, one for instructions and one for data. The caches reduce the processor's external bus activity and increase central processor unit (CPU) throughput by lowering the effective memory access time. For a typical system design, the large caches of the 68040 yield a very high hit rate, providing a substantial increase in system performance. Additionally, the caches are automatically burst-filled from the external bus whenever a cache miss occurs.

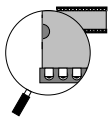
The autonomous nature of the caches allows instruction stream fetches, data stream fetches, and a third external access to occur simultaneously with instruction execution. For example, if the 68040 requires both an instruction stream access and an external peripheral access and if the instruction is resident in the on-chip cache, the peripheral access proceeds unimpeded rather than being queued behind the instruction fetch. If a data operand is also required and is resident in the data cache, it can also be accessed without hindering either the instruction access or the external peripheral access. The parallelism inherent in the 68040 also allows multiple instructions that do not require any external accesses to execute concurrently while the processor is performing an external access for a previous instruction.

Cache organization

The four-way set-associative instruction and data caches have 64 sets of four 16-byte lines for a total cache storage of 4K bytes each. Each 16-byte line contains an address tag and state information. State information for each entry consists of a valid flag for the entire line in both instruction and data caches and write status for each long word in the data cache. The write status in the data cache signifies whether or not the long-word data is dirty (meaning that the data in the cache has been modified but has not been written back to external memory) for data copy-back pages.

The caches are accessed by physical addresses from the on-chip MMUs. The translation of the upper bits of the logical address occurs concurrently with the accesses into the set array in the cache by the lower address bits. The output of the ATC is compared with the tag field in the cache to determine if one of the lines in the selected set matches

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



the translated physical address. If the tag matches and the entry is valid, then the cache has a hit.

If the cache hits and the access is a read, the appropriate long word from the cache line is multiplexed onto the appropriate internal bus. If the cache hits and the access is a write, the data, regardless of size, is written to the appropriate portion of the corresponding long-word entry in the cache.

When a data cache miss occurs and a previously valid cache line is needed to cache the new line, any dirty data in the old line will be internally buffered and copied back to memory after the new cache line has been loaded. Pushing of dirty data can be forced by the CPUSH instruction.

Cachability of data in each memory page is controlled by two bits in the page descriptor for each page. Cachable pages may be either write-through or copy-back, with no write-allocate for misses to write-through pages. Noncachable pages may also be specified as noncachable I/O, forcing accesses to these pages to occur in order of instruction execution.

Cache coherency

Although the large size of the 68040's internal caches significantly increase their ability to boost performance, it also increases the chance that the contents of cache may be different from what is in memory whenever the CPU is operating on memory locations that may also be written to or manipulated by another processor, DMA controller, or VME Master.

To deal with these potential cache-to-memory coherency problems, the 68040 provides several different ways to structure and control how data and instructions are cached:

- **Transparent translation register** — this 68040 register identifies memory areas in 16MB intervals that will not be cached at all. Although this option completely eliminates all possibility of cache-to-memory incoherency, it also seriously degrades the 68040's potential performance.
- **Write-through versus copy-back cached memory** — these cache modes effect how memory is updated in response to updates to the cache. In a write-though cache, the 68040 automatically *writes through* all updates to the cache to memory as well. In a copy-back cache the 68040 does not necessarily *copy back*

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

cache updates to memory until later or until forced by the **cpush** instruction described in the next paragraph.

- **cinv & cpush instructions** — these new 68040 instructions invalidate either a line, page, or all of the cache that has the effect of forcing the 68040 to perform subsequent memory accesses out of actual memory rather than a cache. However, unlike cinv command which only invalidates the cache, cpush forces any changed or *dirty* data to be written back to memory before invalidating the specified line, page, or the entire contents of the cache.
- **Bus snooping** — the 68040 has the ability to snoop the external bus during accesses by other bus masters to maintain coherency between the 68040 caches and external memory systems. External write cycles are snooped by both the instruction cache and data cache; whereas, external read cycles are snooped only by the data cache. In addition, external cycles can be flagged on the bus as snoopable or nonsnoopable. When an external cycle is marked as snoopable, the bus snooper checks the caches for a coherency conflict based on the state of the corresponding cache line and the type of external cycle.

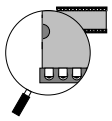
  Although the internal execution units and the bus snooper circuit all have access to the on-chip caches, the snooper has priority over the execution units to allow the snooper to immediately resolve coherency discrepancies.

  Despite its sophisticated features bus snooping may be the least desirable method for ensuring cache coherency for the following two reasons:

  - In its present implementation, the 68040 bus snooping cannot reliably monitor BLT64 transfers for potential cache coherency problems.
  - Bus snooping is a very slow process when applied to large non-BLT data transfers that seriously erodes any performance increase to be gained by caching at all. As a general rule, snooping should only be used as a cache coherency tool in cases where only small non-BLT data transfers are likely to be encountered. For large non-BLT data transfers, it is far more efficient to use one or a combination of the other cache coherency tools listed above.

For more information about using these cache management tools and techniques, see the *MC68040 User Guide* and the *MC68000 Family Programmer's Reference Manual* from Motorola.

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



Setting up the 68040 caches

The paragraphs below describe how to set up the 68040's data and instruction caches after a power-up or reset.

Invalidating cache contents — After a power cycle or reset, any instructions or data remaining in the 68040's internal caches are not reliable and need to be invalidated early in the initialization sequence. This can be accomplished by issuing the following 68040 assembler commands:

```
cinva
nop
```

If you do not have assembler that specifically supports the 68040, the following "hand-assembled" command can also be used:

```
.short 0xF4DB
nop
```



The **nop** instruction is recommended by Motorola to resynchronize the 68040's internal pipelines.

Using the Transparent Translation register — Certain types of data are not compatible with the 68040's new caching and processing techniques. The Transparent Translation register in the 68040 identifies areas in memory to be "translated" directly or given special handling by the 68040's MMU. More detailed information about this register appears later in this chapter.

The address space assigned to I/O functions is a particularly important area for such special treatment for the following two reasons:

- I/O areas are not compatible with data caching because the data they contain can be changed by an external source without the knowledge of the 68040's cache circuitry.
- Normally, I/O activity must be processed in a strict sequence of steps. However, unless otherwise instructed by the MMU, the 68040 may change the order of read and write instructions to optimize its own operations without regard for the needs of external devices.

As a result, the data (dtt0) in I/O regions (above **0xFE00 0000** on V452 Series boards) must be identified for "serialized" processing and must be inhibited from data caching using the following set of 68040 assembler commands:

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

```
movel       #0xFE01C040, d0
movec       d0, dtt0
```

The hexadecimal expression in the **movel** instruction given above directs the 68040 to perform the following individual actions:

❶ The **FE** byte designates the 16 MB area from 0xFE00 0000 to 0xFEFF FFFF as the base area for transparent translation.

❷ The **01** byte designates an additional 16 MB area above the base area (0xFF00 0000 to 0xFFFF FFFF) for transparent translation.

❸ The **C0** byte enables the transparent translation register to contain the memory range expressed in the previous two bytes. It also enables transparent translation for both the user and supervisory modes.

❹ The **40** byte directs the 68040 to inhibit caching, and to serialize the designated memory region.

In lieu of a 68040-compatible assembler, the following sequence can be used which includes a *hand-assembled* command for the **movec** command listed above:

```
movel       #0xFE01C040, d0
.long       0x4E7B0006
```

Enabling and disabling the 68040 caches — With setup complete, the instruction and data caches can be enabled using the following **68030**-compatible assembler command:

```
movel  #0x80008000,  d0      |Write to intermediary scratch register
movec  d0            cacr    |Enable data & instruction caches
```

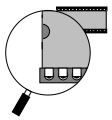
To disable the instruction and data caches execute the following assembler command:

```
movel  #0x00000000,  d0      |Write to intermediary scratch register
movel  d0            cacr    |Disable data & instruction caches
```



Only bits 31 and 15 are meaningful in the **cacr** register. **Bit 31** (0xxx xxxx) turns on and off the 68040's **data** cache. Bit 15 (xxxx 0xxx) turns on and off the 68040's **instruction** cache.

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



Operand transfer types

The 68040 external synchronous bus supports multiple masters and overlaps arbitration with data transfers. The bus is optimized to perform high-speed transfers to and from an external cache or memory. The data and address buses are each 32 bits wide.

The 68040 provides two signals (TT1,TT0) that define four types of bus transfers: normal access, MOVE16 access, alternate access, and interrupt acknowledge access. Normal accesses identify normal memory references; MOVE16 accesses are memory accesses by a MOVE16 instruction; and alternate accesses identify accesses to the undefined address spaces (function code values of 0,3,4,7). The interrupt acknowledge access is used to fetch an interrupt vector during interrupt exception processing.

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

Bus snooping

Bus snooping is a mechanism the 68040 provides to ensure cache coherency, that is, to ensure that data in main memory is consistent with the corresponding data in the 68040's on-chip caches.

Cache coherency is an important consideration while using the 68040 in a mult-ported local memory environment. This is because local memory may be updated by DMA devices, other VMEbus masters, or another 68040 chip causing the data in the 68040's cache from that same area to be invalid or *stale*.

Cache coherency can also be a problem while the 68040 is operating in a high-performance mode called *copyback* mode. In copyback mode the 68040 can update its own cache without updating local memory as well. A subsequent read of local memory by a DMA engine or other Master in this case would also return stale data.

Bus snooping is a means by which the 68040 processor(s) maintain cache coherency between their internal cache and system memory by monitoring or "snooping" external reads or writes to local memory from another CPU Master on the VMEbus or EZ-bus



At present, the 68040's bus snooping feature cannot monitor BLT32 or BLT64 transfers for potential cache-to-memory coherency problems. For more information about other methods to ensure cache coherency, see the section on *Cache coherency* that appears earlier in this chapter.

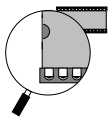
V452 Series boards have several potential masters that transfer data:

- the two 68040 processors
- the VMEbus
- the EZ-bus

Each master may either write data into or read data out of local DRAM memory. The 68040s also contain caches that may hold copies of lines of main memory data; they may be coherent with memory or may contain dirty data (modified data not yet written to main memory).

The task of the snooping logic is to ensure that the 68040's cache is informed of any data writes into main memory by any device, and that

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



any reads of main memory are given the most recent data (the data in a 68040's cache, if it's more recent than the main memory's copy).

If an external bus Master performs a read transfer on the bus while snooping is enabled (via the primary mode register) and if the snoop logic within a 68040 chip determines that the on-chip data cache has *dirty* data (i.e., data valid but not consistent with memory) that is requested for the transfer, memory is prevented from responding to the read request, and the 68040 supplies the data from its cache directly to the Master. If an external bus Master performs a write transfer on the bus while snooping is enabled and if the 68040's snoop logic determines that one of the on-chip caches has a valid line for this request, the 68040 will invalidate the affected cache line.



When there are two 68040's on the board snooped write accesses will properly invalidate both CPU's cache entries if necessary. However, if both CPUs respond to a snooped read access, data contention can occur. The programmer must take appropriate measures to avoid this situation.

Write snooping

The mechanism used for snooped writes is that of invalidating the cache line containing stale (no longer valid) data. The snooping controller inside the 68040, when directed by the Snoop Control bits (see below), holds off the memory response while it searches its caches for an entry corresponding to the address of the memory write. If such an entry exists, then it is marked invalid, and the memory write is then allowed to proceed. This has the effect of requiring the processor to read the valid data from memory the next time it needs it, thus ensuring that the processor gets the most up-to-date data.

Read snooping

The snooping of read cycles is done by allowing the 68040 to provide dirty data to the master. The snooping controller inside the 68040, when directed by the Snoop Control bits (see below), holds off the memory response while it searches its caches for an entry corresponding to the address of the memory read. If such an entry exists, then it is checked to see if it contains dirty data or not.

If the data is not dirty, meaning that the main memory agrees with the cache, the memory read is allowed to proceed. If the cache contains

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

dirty data, however, then the 68040 provides the data to the master while keeping memory inhibited. This has the effect of ignoring the (wrong) contents of memory, while giving the master the most up-to-date data. Subsequent reads from this location will continue to be snooped in this manner until the processor writes the updated data to memory.

Snoop control bits

Snoop cycle control is provided for each of the possible snooped local bus owners. These control bits are listed below.

68040

Each 68040 has two output lines called User-Programmable Attribute bits. These are connected to the Snoop Control bits of both processors so that whichever processor is the master will drive the other processor's Snoop control bits.

The UPA bits are controlled in the MMU. Refer to the 68040 Programmer's Manual for further information regarding the setup and use of these bits.

VMEbus

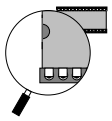
A mode bit called **VSnupEn** controls VMEbus slave cycle snooping. This bit, when clear, will prevent VMEbus slave cycles from being snooped When set, this bit allows VMEbus slave write cycles to be snooped with the SC bits set to 10, causing cache invalidation of stale data. All VMEbus read cycles, when **VSnupEn** is set, will be snooped with the SC bits set to 01, causing the slave processor to provide the dirty data in its cache.

EZ-bus

The EZ-bus snooping is similar to the VMEbus snooping. The control bit **DSnupEn**, when set, allows EZ-bus DMA cycle snooping according to the same rules as for VMEbus snooping.

An additional snooping control for the EZ-bus exists in the **DSnRq** signal. An EZ-bus module that drives this line active will request snooping for this cycle IN ADDITION TO any snooping allowed by the **DSnupEn** bit. In other words, EZ-bus snooping will occur if either **DSnupEn** is set or if **DSnRq** is driven by the EZ-bus module.

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



Snooping between 68040s

Any snoopable memory shared by the two 68040s must be cached in writethrough mode. This must be done because copyback mode doesn't cause a memory cycle when the cache is updated. If processor A and processor B have the same line of data in their caches and processor A updates its cache line in copyback mode, processor B will not know that its corresponding cache entry has become stale.

Memory management unit

The full addressing range of the 68040 is 4 GB (4,294,967,296 bytes); however, most 68040 systems implement a much smaller physical memory. Nonetheless, by using virtual memory techniques, the system can be made to appear to have a full 4 GB of physical memory available to each user program. The independent instruction and data MMUs fully support demand-paged virtual-memory operating systems with either 4K and 8K page sizes. In addition to its main function of memory management, each MMU protects supervisor areas from accesses by user programs and also provides write protection on a page-by-page basis. For maximum efficiency, each MMU operates in parallel with other processor activities.

Translation mechanism

Because logical-to-physical address translation is one of the most frequently executed operations of the 68040 MMUs, this task has been optimized. Each MMU initiates address translation by searching for a descriptor containing the address translation information in the ATC. If the descriptor does not reside in the ATC, then the MMU performs external bus cycles via the bus controller to search the translation tables in physical memory. After being located, the page descriptor is loaded into the ATC, and the address is correctly translated for the access provided no exception conditions are encountered.

Address translation cache

An integral part of the translation function previously described is the dual cache memory that stores recently used logical-to-physical address translation information (page descriptors) for instruction and data accesses. These caches are 64-entry, four way, set-associative. Each ATC compares the logical address of the incoming access against its entries. If one of the entries matches, there is a hit, and the ATC sends the physical address to the bus controller, which then starts the external bus

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

cycle (provided no hit occurred in the corresponding cache for the access).

Translation tables

The translation tables of the 68040 have a three-level tree structure and reside in main memory. Since only a portion of the complete tree needs to exist at any one time, the tree structure minimizes the amount of memory necessary to set up the tables for most programs. As shown in Figure 4, either the user root pointer or the supervisor root pointer points to the first-level table, depending on the value of the function code for an access. Table entries at the second level of the tree (pointer tables) contain pointers to the third level (page tables). Entries in the page tables contain either page descriptors or indirect pointers to page descriptors. The mechanism for performing table search operations uses portions of the logical address (as indices) at each level of the search. All addresses in the translation table entries are physical addresses.

There are two variations of table searches for both 4K and 8K page sizes: normal searches and indirect searches. An indirect search differs in that the entry in the third-level page table contains a pointer to a page descriptor rather than the page descriptor itself.

Entries in the translation tables contain control and status information in addition to the physical address information. Control bits specify write protection, limit accesses to supervisor only, and determine cachability of data in each memory page. Each page descriptor also has two user-programmable bits that appear on the UPA0 and UPA1 signals during an external access for use as address modifier bits.

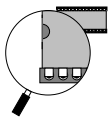
A global bit can be set in each page descriptor to prevent flushing of the ATC entry for that page by some PFLUSH instructions variants, allowing system ATC entries to remain resident during task swaps. If these special PFLUSH instructions are not used, this bit can be user defined. The MMUs automatically maintain access-history information for the pages by updating the used (U) and modified (M) status bits.

MMU instructions

The MMU instructions supported by the 68040 are as follows:

PFLUSH	Allows flushing of either selected ATC entries by function code and logical address or the entire ATCs.

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



PTEST Takes an address and function code and searches the translation tables for the corresponding entry, which is then loaded into the ATC. The results of the search are available in the MMU SR and are often useful in determining the cause of a fault.

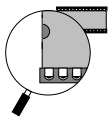
All 68040 MMU instructions are privileged and can only be executed from the supervisor mode.

Transparent translation

Four transparent translation registers, two each for instruction and data accesses, are provided on the 68040 MMU to allow portions of the logical address space to be transparently mapped and accessed without the need for corresponding entries resident in the ATC. Each register is used to define a range of logical addresses from 16 MB to 4 GB with a base address and a mask. All addresses within these ranges are not mapped and are not optionally protected against user or supervisor accesses and write accesses. Logical addresses in these areas become the physical addresses for memory access. The transparent translation feature allows rapid movement of large blocks of data in memory or I/O without disturbing the context of the on-chip ATCs or incurring delays associated with translation table searches.

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

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



68060 CPU

Introduction

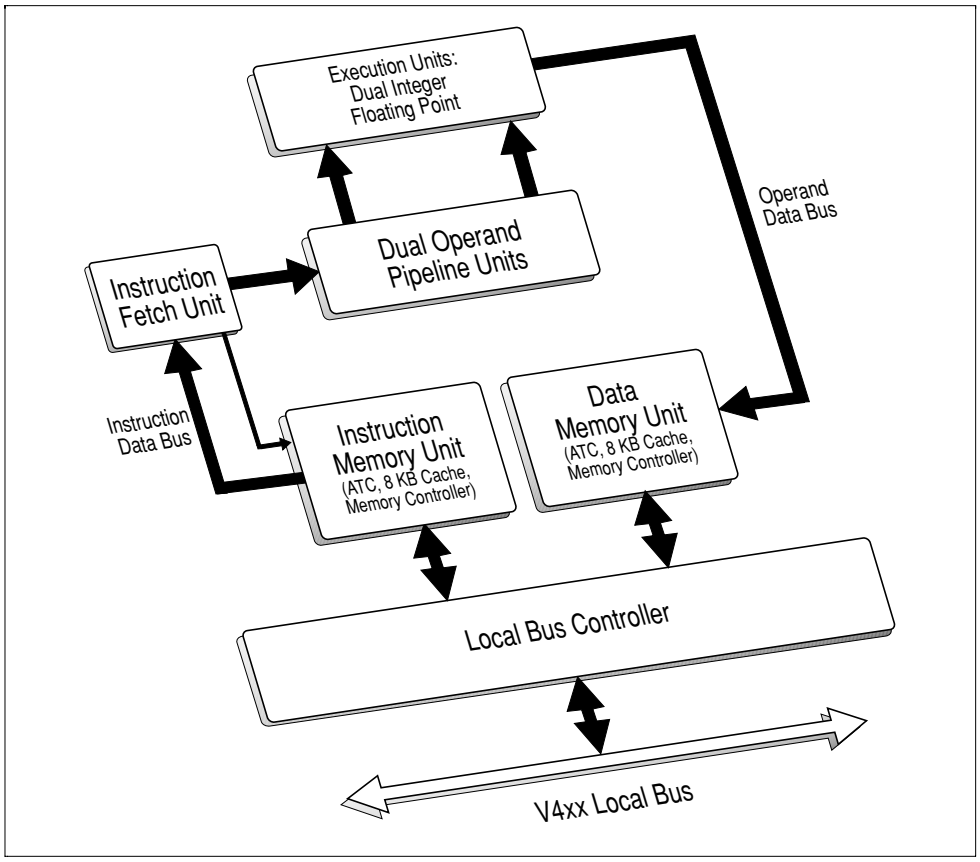
The Motorola® MC68060 is a fourth generation, 68000-compatible, high-performance, 32-bit microprocessor that incorporates some RISC-like features. The 68060 is superscalar, meaning it can perform multiple instructions in a single clock cycle. Integrated within the 68060 chip are dual 68040-compatible CPU integer cores, a 68040-compatible floating point core, independent 8 KB instruction and data caches, a 68040-compatible PMMU and a bus controller. The main features of the 68060 include:

- 1.6 — 1.7 times the 68040 performance on SPEC benchmarks at the same clock rate with existing compilers.
- Dual 8KB instruction and operand data caches (Harvard architecture) with independent, decoupled instruction and operand pipelines.
- Branch prediction logic with a 256-entry, 4-way set-associative, virtually-mapped branch cache for improved branch instruction performance.
- Superscalar pipeline and dual integer execution units achieving simultaneous, but not out-of-order, instruction execution.
- IEEE standard, 68040, 68881/2 compatible floating point execution unit.
- 68040-compatible paged memory management unit with dual 64-entry address translation caches (ATCs).
- Flexible, high-bandwidth synchronous bus interface.

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

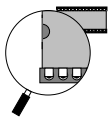
- Upward user-object code compatibility with the 68040 and all previous M68000 microprocessors

Below is a simplified block diagram showing the 68060's main functional blocks.



Simplified 68060 block diagram

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



Additional 68060 documentation

This chapter is not meant to serve as a complete guide for programming the 68060. For detailed 68060 information, refer to Motorola's 68060 publications. These publications are available at:

Motorola Semiconductor Products Sector
Literature Distribution Center
P.O. Box 20924
Phoenix, Arizona 85036-0924

(800) 521-6274
(520) 994-6561

For information on technical training, contact:

Motorola Semiconductor Products Sector
World Marketing Training Operations EL524
P.O. Box 21007
Phoenix, Arizona 85036-0924

(800) 521-6274 (ask for "Training")
(520) 897-3665

The following paragraphs introduce some of the primary features of the 68060 as set forth in the Motorola 68060 Design Specification, Revision 2.0. Differences between the 68060 and 68040 processors are noted where applicable.

Programming model

The 68060 integrates the functions of the integer unit, PMMU, and FPU. The registers contained in the programming model provide access and control for the three units. The registers are partitioned into two levels of privilege: user and supervisor. User programs, executing in the user mode, can only use the resources of the user model. System software, executing in the supervisor mode, has unrestricted access to all processor resources.

The user-level programmer model for the 68060 processor is identical to the MC68020 with the addition of the floating point registers found in the 68881/2 FPU and 68040 CPU.

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

The supervisor-level programmer model is the same as the 68040 — it is used exclusively by system programmers to implement operating system functions, I/O control, and memory management subsystems.

The user-level programming model includes the following:

Eight 32-bit Data Registers (D0-D7)
Seven 32-bit Address Registers (A0-A6)
32-bit Program Counter (PC)
32-bit User Stack Pointer (USP, A7)
16-bit Status Register (as CCR, lower byte of SR only)
Eight 80-bit Floating-point Data Registers (FP0-FP7)
32-bit Floating-point Control Register (FPCR)
32-bit Floating-point Status Register (FPSR)
32-bit Floating-point Instruction Address Register (FPIAR)

The supervisor-level programming model includes all of the above user registers plus the following registers:

One 32-bit Supervisor Stack pointer (SSP, A7)
32-bit Status Register (includes both lower and upper bytes)
One 32-bit Vector Base Register (VBR)
One 32-bit Source Function Code Register (SFC)
One 32-bit Destination Function Code Register (DFC)
One 32-bit Processor Configuration Register (PCR)

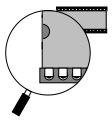
For Paged Memory Management Unit's functions:

One 32-bit Supervisor Root Pointer (SRP)
One 32-bit User Root Pointer (URP)
One 32-bit Translation Control Register (TC)
Two 32-bit Instruction Transparent Translation Registers (ITT0, ITT1)
Two 32-bit Data Transparent Translation Registers (DTT0, DTT1)

For Cache and Bus functionality:

One 32-bit Cache Control Register (CACR)
One 32-bit Bus Control Register (BUSCR)

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



Programmer's model differences: 68060 vs. 68040

• The 68060 has one Supervisor Stack Pointer; the 68040 has the Master Stack Pointer and the Interrupt Stack Pointer.

• Status Register format changes: a single T (trace) bit on the 68060, not T1/T0 of the 68040. 68060 only provides trace capability on every instruction (no trace on change of flow).

• There is no MMUSR register on the 68060.

• The following are new 68060 registers:

— Processor Configuration Register (PCR)

— Bus Control Register (BUSCR)

Data types

The 68060 supports the basic data types listed in the table below. Some data types apply only to the integer unit, some only to the FPU, and some to both the integer unit and the FPU. In addition, the instruction set supports operations on other data types such as memory addresses.

68060 data types

Operand data type	Size	Support	Notes
Bit	1 Bit	IU	--
Bit Field	1-32 Bits	IU	Field of Consecutive Bit
BCD	8 Bits	IU	Packed: 2 Digits/Byte - Unpacked: 1 Digit/Byte
Byte Integer	8 Bits	IU,FPU	--
Word Integer	16 Bits	IU,FPU	--
Long-Word Integer	32 Bits	IU,FPU	--
16-Byte	128 Bits	IU	Memory-Only,Aligned to 16-Byte Boundary
Single-Precision Real	32 Bits	FPU	1-Bit Sign, 8 Bit Exponent, 23-Bit Mantissa
Double-Precision Real	64 Bits	FPU	1-Bit Sign, 11 Bit Exponent, 52-Bit Mantissa
Extended-Precision Real	80 Bits	FPU	1-Bit Sign, 15-Bit Exponent, 64-Bit Mantissa

The three integer data formats common to both the integer unit and the FPU (byte, word, and long word) are the standard twos-complement data formats defined in the M68000 Family architecture. Whenever an integer is used in a floating-point operation, the integer is automatically converted by the FPU to an extended-precision floating-point number before being used. The ability to effectively use integers in floating-point operations saves user memory because an integer representation of a number usually requires fewer bits than the equivalent floating-point representation.

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

Single-and double-precision floating-point data formats are implemented in the FPU as defined by the IEEE 754 standard. These data formats are the main floating-point formats and should be used for most calculations involving real numbers.

The extended-precision data format is also in conformance with the IEEE 754 standard, but the standard does not specify this format to the bit level as it does for a single and double precision. The memory format for the FPU consists of 96 bits (three long words). Only 80 bits are actually used; the other 16 bits are reserved for future use and for long-word alignment of the floating-point data structures in memory. The extended-precision format has a 15-bit exponent, a 64-bit mantissa, and a 1-bit mantissa sign. Extended-precision numbers are intended for use as temporary variables, intermediate values, or where extra precision is needed.

Data type differences: 68060 vs. 68040

The 68060 does not support quadword integers that are supported on the 68040. Attempted execution of an instruction using quadword integers will result in an "Unimplemented Integer" exception.

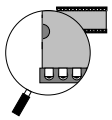
Address modes

The 68060 addressing modes are listed below. The register indirect addressing modes support post-increment, pre-decrement, offset, and indexing, which are particularly useful for handling data structures common to sophisticated applications and high-level languages.

Addressing modes

Register Direct Data Register Direct Address Register Direct
Register Indirect Address Register Indirect Address Register Indirect with Post-increment Address Register Indirect with Pre-decrement Address Register Indirect with 16-bit displacement
Register Indirect with Index Address Register Indirect with Index (8-Bit displacement) Address Register Indirect with Index and base (16-, 32-bit) displacement)
Memory Indirect Memory Indirect Post-indexed Memory Indirect Pre-indexed PC Memory Indirect Post-indexed PC Memory Indirect Pre-indexed

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



Addressing modes (continued)

Program Counter Indirect with 16-bit displacement
Program Counter Indirect with Index PC Indirect with Index (8-Bit Displacement) PC Indirect with Index and base (16-, 32-bit) displacement)
Absolute Absolute Short Absolute Long
Immediate (byte, word, longword, single-precision, double-precision real)

The program counter indirect mode also has indexing and offset capabilities; this addressing mode is typically required to support position-independent software. In addition to these addressing modes, the 68060 provides index sizing and scaling features that enhance software performance. Data formats are supported orthogonally by all arithmetic operations and by all appropriate addressing modes.

Addressing mode differences: 68060 vs. 68040

The 68060 does not support extended-precision floating-point immediate operands supported by the 68040. An attempt to use this addressing mode will generate an “unimplemented EA” exception.

The 68060 does not support usage of immediate operands for the FMOVEM load of floating-point control register (FPCR, FPSR, and FPIAR) combinations that would require 64 or 96 bits of data. An attempt to use this addressing mode will generate an “unimplemented EA” exception.

Instruction set overview

The table on the next page summarizes the 68040 instruction set. A changed instruction as it pertains to the 68060 is shown in bold italic typeface. The table following this one lists the instruction set differences between the ‘040 and ‘060 processors.

If execution of a non-supported interger instuction is attempted, the 68060 processor responds with a unique exception vector (“unimplemented instruction”). Non-supported F (floating point) opcodes generate a Line F Emulator exception.

For detailed information on the 68060 instruction set, refer to the Motorola data book for the 68060 processor.

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

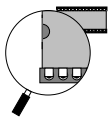
68040 instruction set with 68060 differences noted

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
ADDA	Add Address
ADDI	Add Immediate
ADDQ	Add Quick
ADDX	Add with Extend
AND	Logical AND
ANDI	Logical AND Immediate
ASL, ASR	Arithmetic Shift Left and Right
Bcc	Branch Conditionally
BCHG	Test Bit and Change
BCLR	Test Bit and Clear
BFCHG	Test Bit Field and Change
BFCLR	Test Bit Field and Clear
BFEXTS	Signed Bit Field Extract
BFEXTU	Unsigned Bit Field Extract
BFFFO	Bit Field Find First One
BFINS	Bit Field Insert
BFSET	Test Bit Field and Set
BFTST	Test Bit Field
BKPT	Breakpoint
BRA	Branch
BSET	Test Bit and Set
BSR	Branch to Subroutine
BTST	Test Bit
CAS	***Compare and Swap Operands***
CAS2	***Compare and Swap Dual Operands***
CHK	Check Register Against Bounds
CHK2	***Check Register Against Upper and Lower Bounds***
CINV	Invalidate Cache Entries
CLR	Clear
CMP	Compare
CMPA	Compare Address
CMPI	Compare Immediate
CMPM	Compare Memory to Memory
CMP2	***Compare Register Against Upper and Lower Bounds***
CPUSH	Push then Invalidate Cache Entries
DBcc	Test Condition, Decrement and Branch
DIVS,***DIVSL***	Signed Divide
DIVU, ***DIVUL***	Unsigned Divide
EOR	Logical Exclusive OR
EORI	Logical Exclusive OR Immediate
EXG	Exchange Registers
EXT, EXTB	Sign Extend
FABS	Floating-Point Absolute Value
FADD	Floating-Point Add
FBcc	Floating-Point Branch
FCMP	Floating-Point Compare
FDBcc	***Floating-Point Decrement and Branch***
FDIV	Floating-Point Divide
FMOVE	Move Floating-Point Register
FMOVEM	***Move Multiple Floating-Point Registers***
FMUL	Floating-Point Multiply
FNEG	Floating-Point Negative
FRESTORE	Restore Floating-Point Internal State
FSAVE	Save Floating-Point Internal State
FScc	***Floating-Point Set According to Condition***

Mnemonic	Description
FSQRT	Floating-Point Square Root
FSUB	Floating-Point Subtract
FTRAPcc	***Floating-Point Trap-On Condition***
FTST	Floating-Point Test
ILLEGAL	Take Illegal Instruction Trap
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link and Allocate
LSL,LSR	Logical Shift Left and Right
MOVE	Move
MOVE16	16-Byte Block Move
MOVEA	Move Address
MOVE CCR	Move Condition Code Register
MOVE SR	Move Status Register
MOVE USP	Move User Stack Pointer
MOVEC	***Move Control Register***
MOVEM	Move Multiple Registers
MOVEP	***Move Peripheral***
MOVEQ	Move Quick
MOVES	Move Alternate Address Space
MULS	***Signed Multiply***
MULU	***Unsigned Multiply***
NBCD	Negate Decimal with Extend
NEG	Negate
NEGX	Negate with Extend
NOP	No Operation
NOT	Logical Complement
OR	Logical Inclusive OR
ORI	Logical Inclusive OR Immediate
PACK	Pack BCD
PEA	Push Effective Address
PFLUSH	Flush Entry(ies) in the ATCs
PTEST	***Test a Logical Address***
RESET	Reset External Devices
ROL,ROR	Rotate Left and Right
ROXL,RORX	Rotate with Extend Left and Right
RTD	Return and Deallocate
RTE	Return from Exception
RTR	Return and Restore Codes
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Scc	Set Conditionally
STOP	Stop
SUB	Subtract
SUBA	Subtract Address
SUBI	Subtract Immediate
SUBQ	Subtract Quick
SUBX	Subtract with Extend
SWAP	Swap Register Words
TAS	Test Operand and Set
TRAP	Trap
TRAPcc	Trap Conditionally
TRAPV	Trap on Overflow
TST	Test Operand
UNLK	Unlink
UNPK	Unpack BCD

Note: ***Bold Italic items denote change for 68060.*** See table “68060/68040 hardware-supported instruction set differences”

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



68060/68040 hardware-supported instruction set differences

Mnemonic	Description	Notes
CAS	Compare and swap with operand	Emulation support on 68060 for misaligned operand
CAS2	Compare and swap dual operands	Emulation support on 68060, not implemented in hardware
CHK2	Check register against upper and lower bound	Emulation support on 68060, not implemented in hardware
CMP2	Compare register against upper and lower bound	Emulation support on 68060, not implemented in hardware
DIVS.L	Signed divide	Emulation support on 68060 for 64/32
DIVU.L	Unsigned divide	Emulation support on 68060 for 64/32
FDBcc	FP decrement and branch	Emulation support on 68060, not implemented in hardware
FINT	FP integer part	Implemented on hardware on 68060, not on 68040
FINTRZ	FP integer part, round-to-zero	Implemented on hardware on 68060, not on 68040
FMOVEM	FP move multiple data registers	Emulation support on 68060 for dynamic register list
FScc	FP set according to condition	Emulation support on 68060, not implemented in hardware
FTRAPcc	FP trap on condition	Emulation support on 68060, not implemented in hardware
LPSTOP	Low power stop	Implemented on 68060, not on 68040
MOVEC	Move control registers	Revised functionality
MOVEP	Move peripheral	Emulation support on 68060, not implemented in hardware
MULS.L	Signed long multiply	Emulation support on 68060 for 32 x 32 > 64
MULU.L	Unsigned long multiply	Emulation support on 68060 for 32 x 32 > 64
PTEST	Test a logical address	Not implemented on 68060, PLPA added on 68060 (equivalent functionality)
PLPA	Load physical address	Implemented on 68060, not on 68040

Integer unit

Refer to the discussions on *Programming mode*, *Data types*, *Addressing modes*, and *Exception processing* for lists of programming differences between the 68060 and 68040 integer units.

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

Exception processing

Exception processing is made up of actions by the processor which are outside the normal processing associated with the execution of instructions. The exception state can be externally or internally forced. The externally generated exceptions are interrupts, bus errors, and reset conditions. The interrupts are requests from external devices for processor action; whereas, the bus error and reset signals are used for access control and processor initialization. The internally generated exceptions come from instructions, address errors, tracing, or break points.

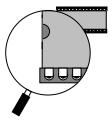
The TRAP, CHK, CHK2, and DIV instructions can all generate exceptions as part of their instruction execution. Tracing behaves like a very high-priority, internally-generated interrupt whenever it is processed. The other internally-generated exceptions are caused by unimplemented floating-point instructions, illegal instructions, instruction fetches from odd addresses, and privilege violations and for invalid descriptors encountered during table searches.

Exception processing for the 68060 occurs in the following sequence:

1. an internal copy is made of the SR,
2. the vector number of the exception is determined,
3. current processor status is saved, and
4. the exception vector offset is determined by multiplying the vector number by four. This offset is then added to the contents of the VBR to determine the memory address of the exception vector. The instruction at the address given in the exception vector is fetched, and normal instruction decoding and execution is started.

The 68060 processor features a simplified exception processing model since it implements the concept of instruction restart to resume execution, rather than instruction continuation. This restart approach causes the faulted instruction to be refetched and re-executed from the beginning.

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



Exception differences: 68060 vs. 68040

- Exception type changes:
  - The 68060 adds “Unimplemented EA” exception type
  - The 68060 adds “Unimplemented Integer” exception type
  - The 68060 does not implement the “Uninitialized Interrupt” exception type.
- Exception stack frame changes:
  - The Type 1 (Throwaway 4-word), because there is no Master Stack Pointer, and Type 7 (Access Error, 30-word) stack frames of the 68040 are not implemented on the 68060.
  - A Type 4 (8-word) stack frame for Access Error exceptions is introduced with the 68060 (with a new Fault Status Longword entry).
- PC and Address Longword Stack contents have changed on 68060 from the 68040 for selected exception types.
- The 68060 processor does not implement “trace on change of flow”.
- Access error changes:

  On the 68060 processor, an access error encountered in the Instruction Fetch Pipeline will not be acted upon until execution is attempted for the instruction associated with the access error.

  The 68060 implements both a precise and imprecise exception model related to bus errors on write operations. Exceptions occurring on writes from the push buffer behave differently on the 68060 than on the 68040

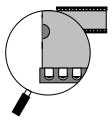
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Exception vector assignments (V452 Series boards and the 68060)

Vector number (dec)	Vector number (hex)	Vector address (hex)	Offset space	Assignment	Status Asserted
0	00	000	SP	Reset Initial Interrupt Stack Pointer	---
1	01	004	SP	Reset Initial Program Counter	---
2	02	008	SD	Bus Error	Yes
3	03	00C	SD	Address Error	Yes
4	04	010	SD	Illegal Instruction	No
5	05	014	SD	Integer divide by zero	No
6	06	018	SD	CHK, CHK2 Instruction	No
7	07	01C	SD	TRAPcc, TRAPV Instructions	No
8	08	020	SD	Privilege Violation	No
9	09	024	SD	Trace	Yes
10	0A	028	SD	Line 1010 Emulator	No
11	0B	02C	SD	Line 1111 Emulator	No
12	0C	030	SD	Emulator Interrupt	
13	0D	034	SD	— Unassigned, reserved — ***	No
14	0E	038	SD	Format Error	No
15	0F	03C	SD	Uninitialied Interrupt	Yes
16 - 23	10-17	40 - 5C	SD	Unassigned, **(EZ-bus modules) ****	
24	18	060	SD	Spurious Interrupt	Yes
25	19	064	SD	**On-board** Level 1 Interrupt **	Yes
26	1A	068	SD	**On-board** Level 2 Interrupt ** **(counters; 82C54)**	Yes
27	1B	06C	SD	**On-board** Level 3 Interrupt ** **(CPU Mailboxes)**	Yes
28	1C	070	SD	**On-board** Level 4 Interrupt ** **(EZ-bus modules)**	Yes
29	1D	074	SD	**On-board** Level 5 Interrupt ** **(Async. serial interface; 2692 UART)**	Yes
30	1E	078	SD	**On-board** Level 6 Interrupt ** **(timers; 2692 UART)**	Yes
31	1F	07C	SD	**On-board** Level 7 Interrupt ** **(ABORT, Bad parity, ACFail, SysFail)**	Yes
32 - 47	20-2F	80 - BC	SD	TRAP #0-15 Instruction Vectors	No
48	30	0C0	SD	FP Branch or Set on Unordered Condition	
49	31	0C4	SD	FP Inexact Result	No
50	32	0C8	SD	FP Divide by Zero	No
51	33	0CC	SD	FP Underflow	No
52	34	0D0	SD	FP Operand Error	No
53	35	0D4	SD	FP Overflow	No
54	36	0D8	SD	FP Signaling NAN	No
55	37	0DC	SD	— FP Unimplemented data type — *	No
56 59	38	E0 - EC	SD	— Unassigned, reserved — ***	No
60		0F0	SD	Unimplemented Effective Address	
61		0F4	SD	Unimplemented Integer Instruction	
62 - 63		F8 - FC	SD	Reserved	
64 - 255	40-FF	100-3FC	SD	User-defined Vectors (192) **(EZ-bus modules)**	

Notes: SP = Supervisor Program Space; SD = Supervisor Data Space
* Assignment is different than that used on the 68030.
** Assignment is particular to the V452 Series boards.
*** Exception is assigned for 68020 and 68030 processors, but not the 68060.
On-board interrupts appear in **bold face**.
The user-defined vectors (64 to 255) may be assigned to the EZ-bus (for daughter modules) if desired.

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



Instruction and data caches

Studies have shown that typical programs spend much of their execution time in a few main routines or tight loops. Earlier members of the 68000 family took advantage of this locality-of-reference phenomenon to varying degrees. The 68040 and 68060 takes further advantage of cache technology with its two, independent, on-chip, physical address space caches, one for instructions and one for data. The caches reduce the processor's external bus activity and increase central processor unit (CPU) throughput by lowering the effective memory access time. For a typical system design, the large caches of the 68060 yield a very high hit rate, providing a substantial increase in system performance. Additionally, the caches are automatically burst-filled from the external bus whenever a cache miss occurs.

The autonomous nature of the caches allows instruction stream fetches, data stream fetches, and a third external access to occur simultaneously with instruction execution. For example, if the CPU requires both an instruction stream access and an external peripheral access and if the instruction is resident in the on-chip cache, the peripheral access proceeds unimpeded rather than being queued behind the instruction fetch. If a data operand is also required and is resident in the data cache, it can also be accessed without hindering either the instruction access or the external peripheral access. The parallelism inherent in the CPU also allows multiple instructions that do not require any external accesses to execute concurrently while the processor is performing an external access for a previous instruction.

Cache differences: 68060 vs. 68040

- Each cache is 8K bytes, vs. 4K bytes on the 68040

- The Operand Data Cache is organized in a banked structure to allow simultaneous read/write accesses.

- Cache entry dirty (modified) designation on the 68060 is done on a line basis, as opposed to one dirty bit per longword on the 68040.

- Cache pushes, whether initiated by the CPUSH instruction or by the replacement algorithm, are always line-size. In addition to line pushes, the 68040 could also do longword sized pushes when only one longword in the line was dirty.

- Operands which require bus transactions are not re-ordered on the 68060 processor (i.e., operand write bus cycles are not deferred past read bus cycles).

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

• Only snoop invalidate is supported; the 68040 supported sourcing and sinking of modified cache data in addition to invalidate. On the 68060 processor, if a snoop hits an entry in either cache, the line will only be invalidated. The 68040 could source a dirty entry before invalidating it.

• The 68040 has a four-deep write buffer for additional performance when writing to writethrough and cache-inhibited imprecise mode pages. This provides decoupling of the processor pipeline from extenal memory for write operations for these references.

• The 68060 implements a “no allocate mode,” on a per cache basis, that provides a frozen cache capability.

Cache organization

The four-way set-associative instruction and operand data caches have 128 sets of four 16-byte lines for a total cache storage of 8K bytes each. Each 16-byte line contains an address tag and state information. State information for each entry consists of a valid flag for the entire line in both instruction and data caches and write status (dirtiness) for each line in the data cache.

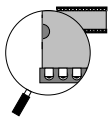
The caches are accessed by physical addresses from the on-chip PMMUs. The translation of the upper bits of the logical address occurs concurrently with the accesses into the set array in the cache by the lower address bits. The output of the ATC is compared with the tag field in the cache to determine if one of the lines in the selected set matches the translated physical address. If the tag matches and the entry is valid, then the cache has a hit.

If the cache hits and the access is a read, the appropriate long word from the cache line is multiplexed onto the appropriate internal bus. If the cache hits and the access is a write, the data, regardless of size, is written to the appropriate portion of the corresponding long-word entry in the cache.

When a data cache miss occurs and a previously valid cache line is needed to cache the new line, any dirty data in the old line will be internally buffered and copied back to memory after the new cache line has been loaded. Pushing of dirty data can be forced by the CPUSH instruction.

Cachability of data in each memory page is controlled by two bits in the page descriptor for each page. Cachable pages may be either write-through or copy-back, with no write-allocate for misses to write-through pages. Noncachable pages may also be specified as noncachable I/O,

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



forcing accesses to these pages to occur in order of instruction execution.

Cache coherency

Cache coherency is an important consideration while using the 68060 in a mult-ported local memory environment. This is because local memory may be updated by DMA devices, other VMEbus masters, or another 68060 chip causing the data in the 68060's cache from that same area to be invalid or *stale*.

Cache coherency can also be a problem while the 68060 is operating in a high-performance mode called *copyback* mode. In copyback mode the 68060 can update its own cache without updating local memory as well. A subsequent read of local memory by a DMA engine or other Master in this case would also return stale data.

One way to maintain cache coherency is through bus snooping. With bus snooping, the 68060 monitors the external bus during bus cycles generated by other masters (bus snooping). If the SNOOP\ pin is asserted, read and write cycles of other bus masters will snoop the instruction and operand data caches, and the push buffer, invalidating entries that match. The instruction cache does not monitor internal operand data accesses.

Bus snooping occurs only when the bus is granted away from the 68060 processor. The processor may continue to operate out of the caches until it needs to access the bus. The bus snooper is given priority over the processor for accesses into the cache which occur simultaneously. Therefore, bus snooping prevents the processor from accessing the caches for the duration of the tag compare.

Other memory updating techniques supported by the processor include copyback and writethrough. Refer to the 68060 data book for more information on maintaining cache coherency.

Setting up the 68060 caches

The paragraphs below describe how to set up the 68060's data and instruction caches after a power-up or reset:

Invalidating cache contents — After a power cycle or reset, any instructions or data remaining in the 68060's internal caches are not reliable and need to be invalidated early in the initialization sequence. This can be accomplished by issuing the following 68040 assembler commands:

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

```
cinva
nop
```

If you do not have assembler that specifically supports the 68040, the following "hand-assembled" command can also be used:

```
.short 0xF4DB
nop
```



The **nop** instruction is recommended by Motorola to resynchronize the 68040's internal pipelines.

Using the Transparent Translation register — Certain types of data are not compatible with the 68060's caching and processing techniques. The Transparent Translation register in the 68060 identifies areas in memory to be "translated" directly or given special handling by the 68060's MMU. Refer to the Motorola 68040 or 68060 databook for more detailed information about this register.

The address space assigned to I/O functions is a particularly important area for such special treatment for the following two reasons:

- I/O areas are not compatible with data caching because the data they contain can be changed by an external source without the knowledge of the 68060's cache circuitry.
- Normally, I/O activity must be processed in a strict sequence of steps. However, unless otherwise instructed by the MMU, the 68060 may change the order of read and write instructions to optimize its own operations without regard for the needs of external devices.

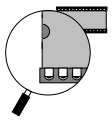
As a result, the data (dtt0) in I/O regions (above **0xFE00 0000** on V452 Series boards) must be identified for "serialized" processing and must be inhibited from data caching using the following set of 68040 assembler commands:

```
movel       #0xFE01C040, d0
movec       d0, dtt0
```

The hexadecimal expression in the **movel** instruction given above directs the 68060 to perform the following individual actions:

❶ The **FE** byte designates the 16 MB area from 0xFE00 0000 to 0xFEFF FFFF as the base area for transparent translation.

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



❷ The **01** byte designates an additional 16 MB area above the base area (0xFF00 0000 to 0xFFFF FFFF) for transparent translation.

❸ The **C0** byte enables the transparent translation register to contain the memory range expressed in the previous two bytes. It also enables transparent translation for both the user and supervisory modes.

❹ The **40** byte directs the 68060 to inhibit caching, and to serialize the designated memory region.

In lieu of a 68040-compatible assembler, the following sequence can be used which includes a *hand-assembled* command for the **movec** command listed above:

```
movel       #0xFE01C040, d0
.long       0x4E7B0006
```

Enabling and disabling the 68060 caches — With setup complete, the instruction and data caches can be enabled using the following **68030**-compatible assembler command:

```
movel  #0x80008000,  d0     |Write to intermediary scratch register
movec  d0            cacr   |Enable data & instruction caches
```

To disable the instruction and data caches execute the following assembler command:

```
movel  #0x00000000,  d0     |Write to intermediary scratch register
movel  d0            cacr   |Disable data & instruction caches
```



The **cacr** register's **Bit 31** (0xxx xxxx) turns on and off the 68060's **data** cache. Bit 15 (xxxx 0xxx) turns on and off the 68060's **instruction** cache. Refer to the Motorola '060 User's Manual for more information on additional cache configuration options provided by the **cacr** register.

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

Paged memory management unit (PMMU)

Like the 68040, the 68060 includes independent instruction and data paged memory management units (PMMUs). The primary function of the PMMU is to translate logical addresses to physical addresses using translation tables stored in physical memory. The address translation cache (ATC) retains copies of recently used address translations. The ATC provides a fast mechanism for address translation by avoiding the overhead associated with table lookup to obtain the mapping of logical to physical addresses.

PMMU differences: 68060 vs. 68040

• There is no MMUSR. The information contained in the 68040 MMUSR is part of the access error exception stack frame on the 68060 processor.

• Instruction changes (PTEST, MOVEC, and PLPA):

  — No PTEST instruction on the 68060 processor, it traps as an illegal instruction exception.
  — New PLPA instruction, load physical address, added to perform PMMU translation/tablewalk and return a 32-bit physical address.
  — MOVEC to/from MMUSR: traps as an illegal instruction exception.

• The 68060 implements a dedicated hardware tablewalker.

• Page table entries cannot be cached in the Operand Data Cache; the tablewalker interfaces directly to the Bus Controller.

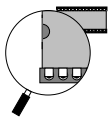
• Differences in Fault Type Information on PMMU access error exceptions:

  — Exception Stack Frame information contained in Fault Status Longword
  — No PTEST instruction necessary on 68060

• No ATC update on the 68060 on tablewalks that terminate due to invalid descriptor type.

• The 68060 implements a “no allocate mode” for the ATCs.

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

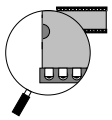


PMMU architecture summary

- Same translation table and register formats as the 68040
- 32-bit logical address translated to 32-bit physical address
- User-defined 2-bit physical address extension (UPA pins)
- Separate instruction and data ATCs
- ATCs each have 64 entries organized in 4-way sets
- Translations that hit in the ATC add no additional time to the access
- Three level page table hierarchy with optional indirection at table descriptor
- Separate supervisor and user translation trees
- History information automatically maintained in the table descriptors
- 4K or 8K page size
- Two memory blocks per PMMU may transparently bypass the ATC
- Instruction and operand data cache modes selectable by page or by transparent block
- Supervisor and write protections
- External PMMU-disable (MDIS pin) for emulator support
- Cache-inhibit output on a page or transparent block basis
- Dedicated tablewalk hardware implementation
- Table descriptors are not cached

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

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



CPU Mailbox

The V452 Series mailbox circuitry provides a flexible and convenient way to implement interrupt-driven inter-process communications control:

- By writing to its own mailbox, the Mailbox circuitry allows an individual CPU to post an interrupt to itself while processing a routine.
- Because each CPU on dual-CPU V452 Series boards has its own separate Mailbox, this circuitry can be invaluable in coupling the dual CPUs for shared processing.
- Mailboxes allow external bus Masters from either the EZ-bus or the VMEbus to effect processing by the on-board CPU(s) via an "on-board" or local interrupt.

This chapter describes the features and functions for the CPU mailbox.

Mailbox memory areas

The memory areas for writing to and reading from each Mailbox are located in different places in the V452 Series address map.

Mailbox write area

The CPU mailbox consists of a message area in memory and a 64x4 FIFO. Whenever a 32-bit write access is made anywhere in the CPU's message area in main memory the FIFO automatically copies and stores the four highest data bits written. The FIFO stores this four-bit message and generates an interrupt to the appropriate CPU, which reads the message from the FIFO.

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

The main memory area used to **write** a message to the on-board CPU lies within the multi-ported memory space accessible by the on-board CPU(s), VMEbus Masters, and/or EZ-bus Masters.

Each on-board CPU has a non-steerable 128-byte Mailbox write area in the last 256 bytes of on-board DRAM (regardless of the size of on-board memory) as follows:

0XXF FF00 — 0XXF FF7F	Mailbox 0 — CPU-X
0XXF FF80 — 0XXF FFFF	Mailbox 1 — CPU-Y **(dual-CPU models only)**



Each Mailbox write area is completely separate from the other. For example, accesses to Mailbox 1 on V452 Series models containing only 1 CPU has no effect on single CPU models or to CPU-X on dual-CPU models.

Mailbox read area

As mentioned in the previous paragraphs, the mailbox circuitry for each on-board CPU includes a 64x4 FIFO. After it copies the top four bits (D28-31) corresponding to each message, the FIFO serves as read area for the on-board CPU.

The CPU "reads" its Mailbox by performing a read access to this FIFO at memory location:

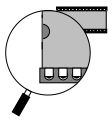
FE30 0000



On dual-CPU models the Mailbox FIFO read area is located at this same address for both CPUs. The Mailbox circuitry allows each CPU to access only to read only its own Mailbox.

Each FIFO can store four bits in this fashion corresponding to up to 64 separate messages.

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



⚠ The 64-deep FIFOs ensure the integrity of up to 64 messages without loss of data. However, the hardware does not provide any provision to prevent overruns once this limit has been met. If the board is to operate in an environment where more than 64 messages may be pending at once, consider implementing a software-driven counter or "test and set" semaphore that can signal when the FIFO is full.

Mailbox interrupts

Both Mailbox FIFOs automatically generate an on-board Level 3 interrupt to the corresponding CPU whenever they contain at least one message. To have an effect, however, the on-board CPU(s) must be programmed to receive the Mailbox interrupt by writing to the appropriate register location in the Interrupt Control register.

Enabling the mailbox(s) as interrupt source(s)

The mailbox for each CPU is a **non-steerable** interrupt source:

- On single-CPU boards there is only one mailbox,
- On dual-CPU boards there are two mailboxes but each CPU has access only to its own mailbox.

To enable Mailbox0 as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE39000D      |Enable Mailbox0 to CPU-X
```

To enable Mailbox1 as an interrupt source to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE39800D      |Enable Mailbox1 to CPU-Y
```


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

Disabling the mailbox(es) as interrupt source(s)

The mailbox(es) can be disabled as an interrupt source(s) by writing to the Interrupt Control registers as described below.

To disable Mailbox0 as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb        #00, 0xFE39000C        |Disable Mailbox0 to CPU-X
```

To disable Mailbox1 as an interrupt source to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb        #00, 0xFE39800C        |Disable Mailbox1 to CPU-Y
```

Mailbox interrupt control

The Mailbox FIFO's ability to generate an interrupt provides interrupt control over multiple message sources.

The FIFO automatically generates an interrupt whenever it contains a message and automatically clears the interrupt when empty.

For example, consider three processors: A, B, and C. Suppose processor B writes to processor A's Mailbox. While processor A is responding to the resulting interrupt, suppose processor C also writes to processor A's Mailbox. The following actions occur:

❶ **Proc B**'s access to **Proc A**'s Mailbox automatically causes a byte to be written to the Mailbox FIFO which in turn generates an interrupt to **Proc A.**

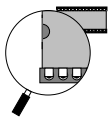
❷ **Proc A** fetches the message written by **Proc B** and processes the interrupt.

❸ After returning from interrupt processing, the FIFO, which still contains the byte from **Proc C**'s access to the mailbox, generates another interrupt.

❹ **Proc A** fetches this second byte of data from the FIFO and processes the interrupt.

❺ After returning from the second interrupt, the FIFO is now empty and **Proc A** resumes its previous processing activities.

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



Message systems

The meaning of these 4-bit "messages" in the Mailbox FIFO is established by system software. One common use of these four bits is to identify the source or Master (i.e., the VMEbus, daughter module, or 68060) of the device sending the message. If multiple message sources are present, it is possible to use the FIFO and mailbox memory area to create a **sorting bin** for incoming messages as follows:

❶ Allocate a section of mailbox memory for each source of messages (Mailbox Master) in the Mailbox write area.

❷ When a corresponding Mailbox Master wants to send a message, it writes a data word, byte, or longword into its assigned Mailbox. The top four bits of the word contain the ID code of the requesting Master.

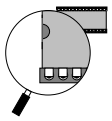
❸ When the receiving CPU encounters the resulting interrupt, it reads the Mailbox FIFO to determine the Master ID.

❹ The receiving CPU then looks for the message in the address location of Mailbox memory that is assigned to the identified Mailbox Master.

If there is only a single message source, another common use of the 4-bit message is to indicate a specific location in a pre-arranged memory range where the remainder of the message can be found.

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

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



CPU Watchdog

V452 Series boards use the Dallas Semiconductor DS1232 Micro Monitor chip (or equivalent) as the primary component in the CPU Watchdog circuitry. Additional product information for this chip is available from:

Dallas Semiconductor
4401 South Beltwood Parkway
Dallas, Texas 75244-3292
(214) 450-0448

The V452 Series CPU Watchdog circuitry can be programmed to monitor the status of the on-board CPU(s) and assert a board-level Local Reset, equivalent to manually toggling **both** front panel RESET switches if either one and/or both of the following conditions occur.

- **CPU Run monitor** — If enabled to do so, the Watchdog Run monitor can assert a reset unless it is toggled off and then back on again at least every **250 ms**. The default condition for the Run monitor is disabled.
- **CPU Halt monitor** — The Watchdog can also assert a reset if the on-board CPU(s) halts. The default condition for the Halt monitor is also disabled.

In addition, the CPU Watchdog can be deactivated completely by disabling both of the monitors listed above.

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

Control registers

V452 Series boards control the CPU Watchdog circuitry via software-programmable registers listed in the table below. Using these registers you can set the Watchdog to monitor the software's run status, the CPU's halt status or both.

Extended Mode register (OxFE38 4003)

Hex data	Function	For more info, see chapter
04	Disable CPU Watchdog Run monitor ***(default)***	*Status, ID & Mode registers*
0C	Enable CPU Watchdog Run monitor	

Extended Control register (OxFE38 C000)

Reg address	Function	For more info, see chapter
FE38 C004	Disable CPU Watchdog Halt monitor ***(default)***	—
FE38 C005	Enable CPU Watchdog Halt monitor	—

Watchdog Run monitor

The CPU Watchdog circuitry can be used to monitor the run status of the application software by responding to the execution (or the lack of execution) of special access instructions imbedded in application programs.

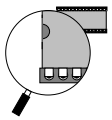
Once enabled, the Watchdog Run monitor automatically asserts a board-level Local Reset if the CPU(s) fails to execute the enable instruction sequence described below at least every 600 ms. To *hold off* a reset, this sequence must be inserted into a module of the application software that can ensure that it executed within this time frame.



The Run monitor circuitry can be held off (and/or disabled) from asserting a reset only by a supervisor write accesses from the CPU. Accesses from the VMEbus or an active VSB bus has no effect.

Using this technique the Watchdog Run monitor can effectively watch program execution to ensure particular sequences execute successfully. If the sequence fails to complete (which results in the reset *hold off* instructions not being executed), the Watchdog automatically resets the board (or the entire VMEbus if the board is also serving as the System Controller) thereby returning the system to a "safe" state.

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



The paragraphs below describe the instructions required to use the CPU Watchdog Run monitor.

Enabling the Run monitor

After power cycling or a rest, the Watchdog Run monitor is disabled. To enable it, execute the following 68030 (or above) assembler instructions to toggle off and then on the Run monitor:

```
moveb       #0x04, 0xFE384003
moveb       #0x0C, 0xFE384003
```

Note Location 0xFE38 4003 is the V452 Series **Extended Mode register**. For more information, see the ***V452 Series internal registers*** chapter in Section 3.

Holding off a reset

Once enabled, the Watchdog automatically asserts a board-level Local Reset if the CPU(s) fails to repeat the **same** enable instruction sequence shown above at least every 250ms.

Specifications for the Dallas 1232 chip list the tolerance for the access period at 250ms–1000ms with 600ms as the typical value.

Note The Run monitor circuitry can be held off (and/or disabled) from asserting a reset only by a supervisor write accesses from the CPU. Accesses from the VMEbus or an active VSB bus has no effect.

Disabling the Run monitor

The Run monitor is disabled whenever the board is reset or after execution of the following 68030 (or above) assembler instruction:

```
moveb       #0x04, 0xFE384003
```


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

Watchdog Halt monitor

The CPU Watchdog circuitry can be used to monitor the HALT status of the on-board CPU(s). Once enabled, the Watchdog Halt monitor automatically asserts a board-level Local Reset if the CPU(s) halts or hangs.



The only way to hang the CPU while the Watchdog Halt monitor is enabled is to attempt an VMEbus access while no System Controller is active.

The paragraphs below describe the instructions required to use the CPU Watchdog Halt monitor.

Disabling the Halt monitor

After power cycling or a reset, the Watchdog Halt monitor is disabled. To enable it, execute the following 680x0 assembler instruction:

```
moveb          #0x00, 0xFE38C005
```



Location 0xFE38 C004 is located within the V452 Series **Extended Control register**. For more information, see the ***V452 Series internal registers*** chapter in Section 3.

Enabling the Halt monitor

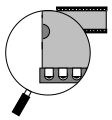
The Halt monitor is disabled whenever the board is reset or after execution of the following 68030 (or above) assembler instruction:

```
moveb          #0x00, 0xFE38C004
```



Issuing a 68000 STOP instruction stops the CPU but doesn't halt it. Thus, you can safely stop the CPU to wait for an interrupt without causing the Watchdog Halt monitor to assert a reset.

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



Dual-CPU considerations

On dual-CPU models, it may be desirable to disable the Halt monitor so that the CPU that remains running can debug and reset the halted CPU.

Disabling all Watchdog functions

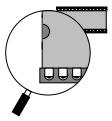
Both the Watchdog Halt and Run monitors are disabled in the default condition. If they are subsequently enabled, the single assembly instruction listed below can be used to to disable both features at once:

```
moveb          #0x00, 0xFE38C004
```

It is also possible to disable both of the Watchdog monitors by resetting the V452 Series board.

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

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



Dynamic RAM

All V452 Series boards use extended data out dynamic RAM (EDO DRAM) on a plug-in board. The EDO DRAM is multi-ported, that is, it is accessible by the following devices:

- The on-board CPU(s).
- Daughterboard Master(s) across the Synergy EZ-bus.
- The on-board Block Transfer (BLT) DMA engine.
- Other VMEbus Masters.

The CPU and EZ-bus daughter module Master can access on-board DRAM directly at the addresses listed in the *Address map* chapter. VME Masters can access on-board DRAM via an access window described later in the *VME Slave interface* chapter.

This chapter describes the architecture and operation of the DRAM.

DRAM access optimization

DRAM memory on Synergy CPU boards is optimized to allow the fastest possible accesses by the on-board CPU chip and for BLT DMA transfers. Although this approach tends to reduce certain benchmarks for single-cycle accesses from the VMEbus, the net result is a significant improvement to overall system performance because memory accesses by the on-board CPU and for BLTs represent a very large percentage of total memory activity.

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

DRAM tuning strategies

CPU board performance is strongly affected by the tuning of the RAM speed to the CPU speed. The paragraphs below describe three DRAM tuning strategies:

Maximum MIPS per MHz Maximize the performance of the SBC by choosing higher-speed **DRAM** for a **given** CPU speed thus reducing DRAM wait-states. This strategy makes sense when only one CPU speed is available or when the fastest CPU available is still below your performance expectations.

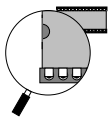
Maximum MIPS per dollar Maximize performance by upgrading to a higher-speed **CPU** rather than to faster and significantly more expensive RAM. This strategy is particularly well-suited to the 68040/060, which offers multiple clock speeds, and in cases where a large amount of DRAM would need to be replaced.

The V452 Series burst mode DRAM can provide 4-1-1-1 clock read and 4-2-2-2 clock write accesses. As an example of what these numbers represent, a 4-1-1-1 burst read cycle means the CPU fetches 4 consecutive, uninterrupted longwords from memory. The first longword is fetched in 4 clock cycles and the additional 3 longwords are each fetched in 1 clock cycle each. Therefore, a 4-1-1-1 burst cycle fetches 4 longwords in a total of 7 clock cycles (4+1+1+1=7), which is equal to an average cycle time per longword of 1.75 clock cycles (At 50 MHz a clock cycle is equal to 20 nanoseconds). At one long word/35 nanoseconds (20 x 1.75) yields a data transfer rate of 114 MB/s.

Both the 68040/060 data and instruction caches use burst accesses to fill themselves. **But to get any benefit from the burst accesses, the caches must be enabled which automatically enables burst mode.** For general information about the operation of the cache, see the Cache Control register discussion in the Motorola 68040/060 manual. For more information about turning on the caches, see *Setting up the V452 Series software* in the ***Getting Started*** section.

The following table shows the relative performance in Dryhstones of various DRAM/CPU combinations for the currently available speeds of 68040, 68030, and 68020 processors.

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



Relative speed of 68040-68020 designs (in dhrystones)

MHz	68040		68030				68020	
cycles	3-111*	4-111*	2	3	4-222*	5-222*	0-wait	1-wait
12.5	—		—	—	—	—	***4847***	4020
16.6	—		—	—	—	—	***6437***	5338
20.0	—		9460	8230	***7473***	7095	7757	***6432***
25.0	***32600***		11825	10264	—	***8868***	—	***8041***
33.3		***43030***	15750	13717	—	***11812***	—	10710
50.0	—		23500	***20500***	—	***17500***	—	—

Notes: * Access speed listed is for read accesses.

The performance figures are shown in Dhrystones. Because the Dhrystone benchmark is affected by the compiler used as well as the hardware performance, the absolute values shown are less important than the relative values for the different CPU-DRAM configurations. The configurations giving the best MIPS/Dollar are shown ***in this type face***.

Burst transfer operation

The 68040/060 employs a special type of data transfer called a burst. This is used to transfer a cache line (16 bytes) between the data or instruction caches and external memory. The V452 Series boards provide support for burst transfers to/from the main onboard DRAM memory. The advantage of the burst transfer is its speed. A burst read transfer moves 16 bytes of data in 6 clock cycles, whereas a non-burst 16 byte transfer requires 15 clock cycles, for a 2.5-to-1 speed improvement.

The burst transfer is used only when 68040/060 caching is enabled. Thus, the caches must be enabled to obtain the best performance from the DRAM. However, the MMU transparent translation registers must first be set up to disable caching of I/O locations. For more information about cache operations, see the appropriate CPU chapter that appears earlier in this section.

Memory parity checking

Although modern dynamic RAM is very reliable, it is still subject to errors. As a result, DRAM is often designed with parity checking in order to increase the likelihood that memory failures will be detected.

Each 8-bit byte in the on-board memory has an associated 9th bit which is used to encode odd parity for that byte. Odd parity is generated for

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

each byte whenever a write of any size (32 bits, 16 bits or 8 bits) is performed, that is, the 9th bit is set or cleared in order to make the number of one's in the 9-bit byte be an odd number. If parity checking is enabled, even parity results in a parity error.

All V452 Series CPU boards provide longword parity detection that can be turned on or off. It is also possible (if desired) to configure the board to assert a Level 7 interrupt following the detection of parity errors. When it encounters a parity error (with parity checking enabled), the error detection circuitry writes a 1 (or high) to bit 8 in the Status register (0xFE38 0002). The paragraphs below describe functional aspects and some operational considerations for using memory parity checking on V452 Series boards.

Enabling/disabling parity checking

After DRAM initialization, it is possible to safely enable memory parity checking by writing to the **Primary Mode register** (0xFE38 0003).

During power-up or after a reset, parity checking defaults to the disabled state. To enable parity checking, write **0x0E** to the **Primary Mode register** using the following 68000 assembler instruction:

```
moveb       #0x0E, 0xFE38 0003    |Enable parity checking
```

To disable parity checking, write **0x06** to the **Primary Mode register** using the following 68000 assembler instruction:

```
moveb       #0x06 0xFE38 0003     |Disable parity checking
```

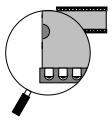
Enabling/disabling parity error interrupts

If memory parity checking is enabled, V452 Series boards can also be configured to generate a Level 7 interrupt whenever a parity error occurs. During power-up or after a reset, however, the maskable Level 7 interrupt is disabled. To enable it on a single-CPU board or for CPU-X on a dual-CPU board, perform a write access to location 0xFE39 4001 in the **Interrupt Control register** using the following 68000 assembler instruction:

```
moveb       #0x00, 0xFE39 4001    |Enable Level 7 int to CPU-X
```

To enable the Level 7 parity error interrupt for CPU-Y on the dual-CPU board, perform a write access to location 0xFE39 C001 in the **Interrupt Control register** using the following 68000 assembler instruction:

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



```
moveb          #0x00, 0xFE39 C001     |Enable Level 7 int to CPU-Y
```



Level 7 interrupts can be generated by multiple sources on V452 Series boards including the front panel ABORT switches, the VME SysFail, and ACFail signals. To determine whether a parity error was the source of a pending Level 7 interrupt, the interrupt routine can read **bit 8** in the **Status register** (0xFE38 0002). This bit is set to a 1 (or high) whenever a parity error is detected.

For more information about Level 7 interrupts and interrupt sources, see the *Interrupts* chapter in the ***Board Facilities*** section.

To disable the Level 7 parity error interrupt on a single-CPU board or for CPU-X on a dual-CPU board, either reset the board or perform a write access to location 0xFE39 4000 in the **Interrupt Control register** using the following 68000 assembler instruction:

```
moveb          #0x00, 0xFE39 4000     |Disable Level 7 int  to CPU-X
```



The circuitry for the Interrupt Control register is designed to decode only the address portion of write accesses to it. The data value expressed in the instruction does not matter.

To disable the Level 7 parity error interrupt for CPU-Y on the dual-CPU 420, either reset the board or perform a write access to location 0xFE39 C000 in the **Interrupt Control register** using the following 68000 assembler instruction:

```
moveb          #0x00, 0xFE39 C000     |Disable Level 7 int  to CPU-X
```



If Level 7 interrupts are left **disabled** while parity checking remains **enabled**, the board no longer asserts an interrupt after a parity error but can still detect parity errors and report them via **bit 8** in the **Status register** (0xFE38 0002).

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

Clearing the parity error bit

As described above, when parity checking is enabled, bit 8 in the Status register is set to 1 (high). This bit can be cleared to the default 0 (low) value by either resetting the board or by disabling and then re-enabling parity checking by executing the following 68000 instructions:

```
moveb    #0x06, 0xFE38 0003   |Disable parity checking
moveb    #0x0E, 0xFE38 0003   |Enable parity checking
```



Clearing the parity error bit in the Status register as described above causes the Status register to stop indicating a parity error, but it does **NOT** in any way clear the actual bad parity condition at the faulty memory location. Rather, clearing the parity error bit in the Status register restores the ability of the board to perform parity checking. This ability to restore parity checking without clearing the bad parity is very useful in isolating an individual offending byte responsible for a parity error after a 16 or 32-bit access to memory. This isolation technique is described in more detail in the next section.

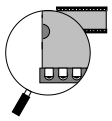
Detecting & isolating bad parity

Odd parity is generated for each byte whenever a write of any size (32 bits, 16 bits, or 8 bits) is performed, that is, the 9th or parity bit is set or cleared in order to make the number of one's in the 9-bit byte be an odd number. If the data in each byte becomes corrupted in such a way as to change the parity value to an even number, the parity protection circuitry indicates that a memory failure has occurred by writing to the Status register as described in the preceding paragraphs.

Whenever a read of any length is performed, the CPU fetches an entire 32-bit longword. The parity protection circuitry checks the entire long-word for a parity. If any of the bytes in the longword have bad parity, the parity checking circuitry reports a parity error regardless of the parity status of the individual byte or nibble that was originally accessed.

When a parity error is encountered, the offending byte can be isolated by reading and writing each byte in the longword and then rechecking

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



the longword for parity. The following procedure can be used after a parity error has been encountered:

1. **Clear the parity error bit in the Status register** — by disabling and then re-enabling the parity detection as described above.
2. **Disable the 68060's data and instructions caches** (if active).
3. **Disable the Miscellaneous Level 7 interrupts** (if active) — Isolating which byte has the error may require multiple checking passes. To allow the isolation routine to remain in control despite the occurrence of a parity error, disable the Miscellaneous Level 7 interrupt group (of which parity errors is one of the sources).

   In addition to parity errors, the maskable Level 7 interrupt group also includes the VMEbus SysFail and ACFail signals as possible interrupt sources. For more information, see the ***Interrupts*** chapter in Section 3.

4. **Read and Write back individual bytes and recheck the parity of longword until the error clears** — Repeat steps 1 and 4 as necessary to recheck parity of the longword.

Multi-port memory contention

If another VMEbus and/or EZ-bus daughter module Master attempts to access on-board memory while the CPU is also executing out of on-board memory, a priority scheme must be implemented to manage contention.

On a V452 Series board, the CPU is allowed to complete its memory cycle first, followed by either the VMEbus or EZ-bus daughter module(s) depending on which one accessed on-board memory last (i.e. The VMEbus Master and EZ-bus daughter module(s) exchange memory access priority on a round-robin basis). The CPU must then wait until after the VMEbus or daughter module Master(s) relinquishes the local bus before attempting its next cycle.

This alternation of bus Mastership can potentially reduce the performance of the CPU depending on how often such "other Masters" access the triple-ported on-board memory. The worst case would occur if the CPU and another Master were alternately executing instructions or fetching data from on-board memory at the same time, thus forcing arbitration for the Local bus on every cycle. Of course, memory contention overhead is greatly reduced when the CPU is using its instruc-

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

tion and data caches while another Master is accessing on-board memory.

If a VMEbus Master attempts to access on-board RAM while the CPU is simultaneously attempting to access the VMEbus, the CPU backs off and lets the VME Master complete its access. The CPU then retries its access to the VMEbus. This pause by the CPU is totally transparent to the software.



Whenever an external VMEbus Master, DMA Controller, or EZ-bus Master writes to a portion of the V452 Series memory that is being cached by the on-board CPU, a cache-to-memory incoherency condition is possible that could result in a loss of data. For a discussion of these cache coherency considerations and a summary of some useful cache management techniques, refer to the approriate ***CPU*** chapter in this section and the applicable Motorola databook.

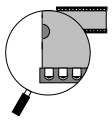
DRAM address decoding

The paragraphs below describe how the V452 Series board locates a specific memory address location.

As seen by the 680x0

The on-board DRAM occupies 200,000, 400,000, 800,000, 1,000,000, 2,000,000, 4,000,000 or 8,000,000 hex bytes of address space corresponding to 2, 4, 8, 16, 32, 64 or 128 MB of DRAM. In the standard V452 Series configuration, the DRAM is configured to begin at address 0 as seen by the CPU.

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



As seen by an EZ-bus daughter module

Any EZ-bus daughter module that is capable of serving as a bus Master can be configured to provide either 24-bit or 32-bit Master mode addressing:

- Modules using 24-bit addressing, can see the same DRAM address map as the CPU. Unlike the CPU, however, modules that use 24-bit addressing will not be able to access any address space beyond 16 MB (i.e., V452 Series peripherals, expansion RAM beyond 16 MB, or VME address space). For most EZ-bus daughter modules, 24-bit addressing is the standard board configuration.
- Modules using 32-bit addressing use the same address map as the on-board CPU. The V452 Series boards are also designed to support a 32-bit EZ-bus daughter module Master. This setting gives the daughter module the same address map as the CPU.



For more information about addressing modes on EZ-bus modules, see the ***EZ-bus Designer's Guid***e from Synergy.

As seen by the BLT DMA engine

The Block Transfer DMA engine sees the same address map as the on-board CPU. For more information, see the ***VME Master BLT*** chapter in Section 5.

As seen by the VMEbus

The VMEbus sees and accesses on-board DRAM through an access window that is described in the ***VME Slave interface*** chapter in Section 5.

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

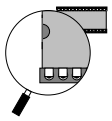
R452/R453 memory modules

All V452 Series boards supply local memory on a modular R452/R453 daughter card that plugs onto the P9, P10 and P11 connectors on the main board. Separate modules are available containing 4, 8, 16, 32, 64 and 128 MB of DRAM. This modular design provides an easy way to change the amount of on-board memory without taking the board out of service and returning it to the factory for rework.

Depending on the current use of the board as a Slave in a larger VMEbus system, some software configuration changes may be required to prepare the board for active use.

For a procedure that describes how to install or replace R452/R453 modules on V452 Series boards, see the ***Installing the R452/R453 memory module*** in Section 2.

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



EPROM

V452 Series boards provide **two 32-pin** sockets (.6 inches wide) at UG13 (EPROM1) and UJ13 (EPROM0) for one or two of the following types of JEDEC-standard byte-wide EPROM memories† :

- 27C010 1 Mbit EPROM
- 27C020 2 Mbit EPROM
- 27C040 4 Mbit EPROM
- 27C080 8 Mbit EPROM
- 28F010 1 Mbit Flash EPROM (128K x 8)
- 28F020 2 Mbit Flash EPROM (256K x 8)

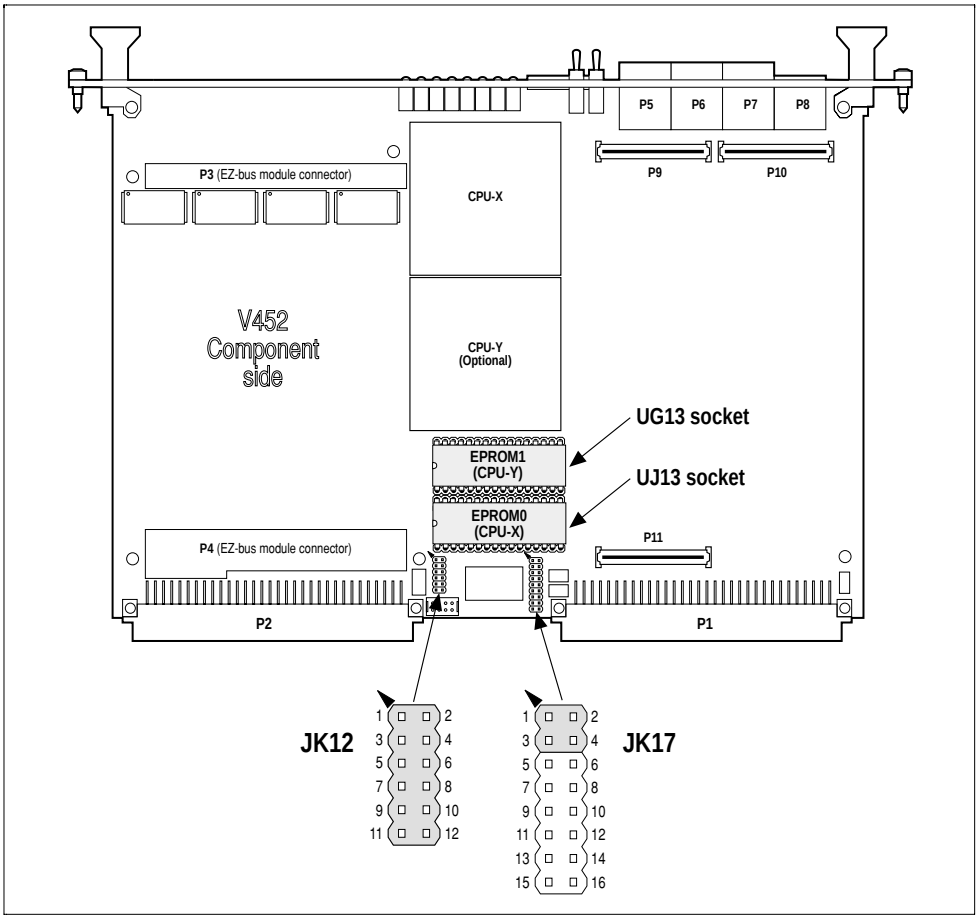
The V452 Series PROM socket circuitry provides the following features and capabilities:

- primary interface for monitor firmware.
- an 8-bit data path to the CPU that minimizes the number of EPROMs and power usage.
- up to 2 MB of 8-bit wide EPROM memory space when the largest supported EPROM type (2 ea. 27C080) is used.
- up to 512 KB of 8-bit wide Flash EPROM memory space when the largest supported Flash EPROM type (2 ea. 28F020) is used.
- allows combination of one EPROM and one Flash EPROM device on the same board.

The figure below shows the location of the two EPROM sockets and jumpers on V452 Series boards.

† TI brand EPROMs cannot be used. Their requirement for Vcc on unused pins prevents a TI PROM from being used in a general purpose socket. EPROMs from other manufacturers such as Intel, AMD, etc. work without problem.

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



V452 Series jumper and socket locations for EPROM configuration

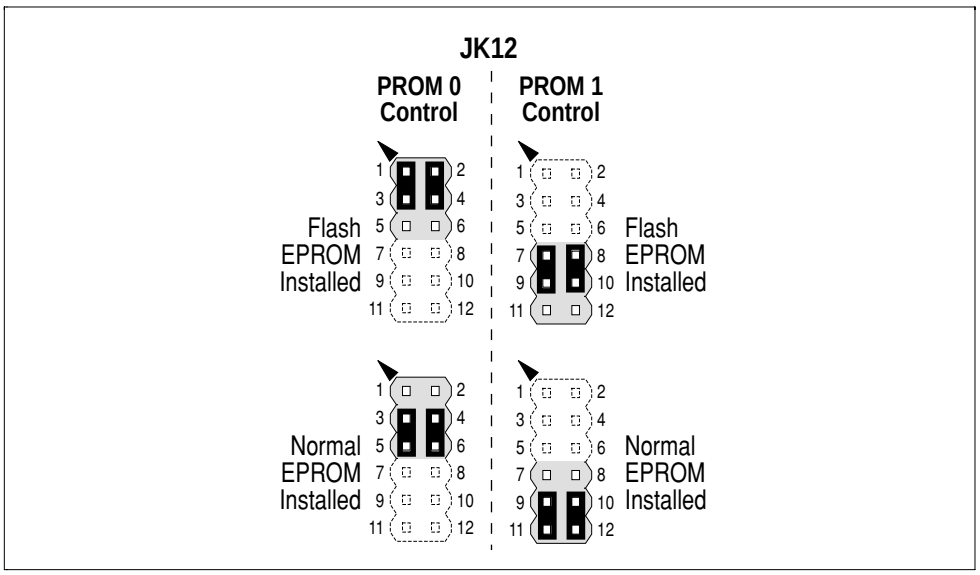
Selecting the monitor EPROM type

The type of EPROM (i.e., Flash or normal EPROM) installed on V452 Series boards must be indicated by configuring jumper JK12 as shown in the figure on the following page.

For proper operation, all V452 Series boards must include an EPROM at the PROM socket 0 position. The PROM 1 socket is required only on dual-CPU models or to contain additional firmware for single-CPU model boards.

> **Note** Unless specifically ordered otherwise, all V452 Series boards are supplied with the required EPROM and JK12 jumper settings. For more information about installing customized firmware, contact Synergy.

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



JK12 — EPROM type selection

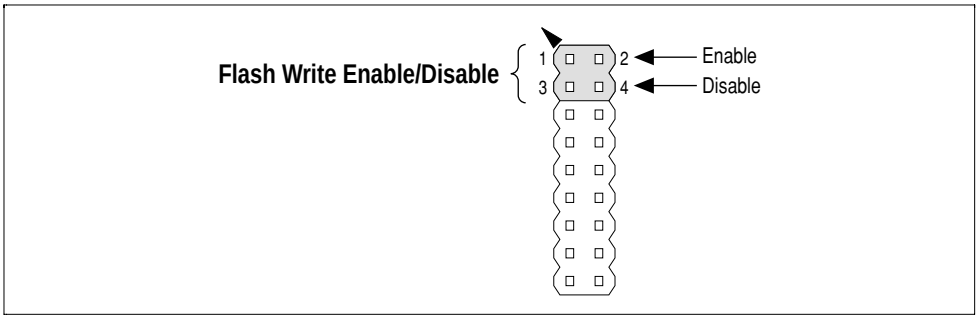
Flash EPROM configuration

The PROM circuitry on V452 Series boards include all of the required lines and signals to allow on-board reprogramming of Flash EPROM devices.

> **Note** For more information about the software aspects of writing to Flash EPROM devices, see the chapter containing *Flash EPROM programming tools* in the ***Code Examples*** section and the *28F010 1 Mbit Flash EPROM* data sheet in the ***Datasheet Supplement*** that contains reprinted portions of the manufacturer's datasheets for devices used on V452 Series and other boards from Synergy.

In addition to the software instructions and signals that must be asserted to enable writes to installed Flash EPROM devices, V452 Series boards also require a jumper setting as well. The figure below shows the jumper JK17 settings to enable or disable writing to the Flash device installed in the EPROM sockets.

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



JK17 — Flash EPROM Write enable/disable

Note The EPROM circuitry reads this jumper only if Flash EPROM has been selected using jumper **JK12**. If Flash EPROM is not selected for either PROM, the setting for jumper **JK17** does not matter.

EPROM use

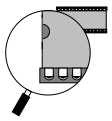
V452 boards provide two EPROM sockets for the storage of boot and other firmware for the on-board CPU(s). With the board configured to boot from EPROM (EPROM boot enable jumper installed on JK17 pins 17 & 18), a V452 power cycle or reset will cause the CPU(s) to fetch its start-up vector on:

- Single-CPU models — EPROM0 (0xFE000000). This start-up vector can point to any location on either EPROM.

- Dual-CPU models — EPROM0 (0xFE000000) for CPU-X and EPROM1 (FD000000 or FE400000) for CPU-Y.

  The start-up vector can point to any location in EPROM0 or EPROM1 (old or new space), that is, any address from FE000000–FE0FFFFF or FD000000–FDFFFFFF (new EPROM1 space) or FE400000–FE4FFFFF (old EPROM1 space). Refer to the ***Address Map*** chapter in Section 3 and this section's ***Flash memory module*** chapter for more information on the EPROM1 address space.

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



Flash considerations

When Flash Write is enabled — Your application code must contain safeguards when Flash Write is enabled to avoid inadvertent writing of Flash memory which may render the processor(s) inoperative. In particular, CPU-Y will not boot after a reset if the reset vectors have been overwritten.

> ⚠ If Flash memory is to be long-term, read-only, set the JK17 Flash Write enable/disable jumper to the **disable** position to avoid accidental writing of Flash memory.

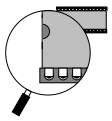
When programming Flash on a dual-CPU board — On a dual-CPU board, CPU-Y might interfere with attempts to reprogram the Flash by attempting to boot itself. In this case, your programming code must first prevent CPU-Y from making EPROM1 accesses. This is done with the ***EPROM1 programming*** bits provided in the extended control register (FE3A0006-7). The basic procedure is:

1. Write to FE3A0007. Setting this bit prevents CPU-Y from accessing EPROM1.
2. Program the Flash. At this point, CPU-X can program the Flash unhampered.
3. After programming, restart the dual-CPU board to resume normal operation. After restart, CPU-Y defaults to normal access of EPROM1 (FE3A0006 = 1).

For more information on the V452 control registers, refer to the ***V452 internal registers*** chapter in Section 3.

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

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



Flash memory module

As described in the preceding chapter, the V452 Series EPROM0 and EPROM1 sockets each support 28F010 or 28F020 Flash EPROMs which have a capacity of 128KB or 256KB respectively. If more Flash memory capacity is desired, optional Flash memory modules provide up to 4 MB or 16 MB respectively.

This option differs from the Flash EPROM option in that the Flash memory modules are based on Intel's FlashFile™ memory chips. These chips have a byte write and block erase architecture with data storage similar to that of a sectored hard disk. The Flash module, thus, lets the SBC load an operating system or execute a large program locally for operation as a diskless or standalone (non-networked) unit. Listed below are the available Flash module types.

- **DEFL**: 2 or 4 MB of Flash memory
- **DELF**: 4, 8, or 16 MB of Flash memory

A Flash module replaces EPROM1. It is plugged into the motherboard's EPROM0 and EPROM1 sockets and an auxiliary socket strip. EPROM0 is still used, but it is plugged piggyback-fashion onto the top of the Flash module.

Flash memory address location

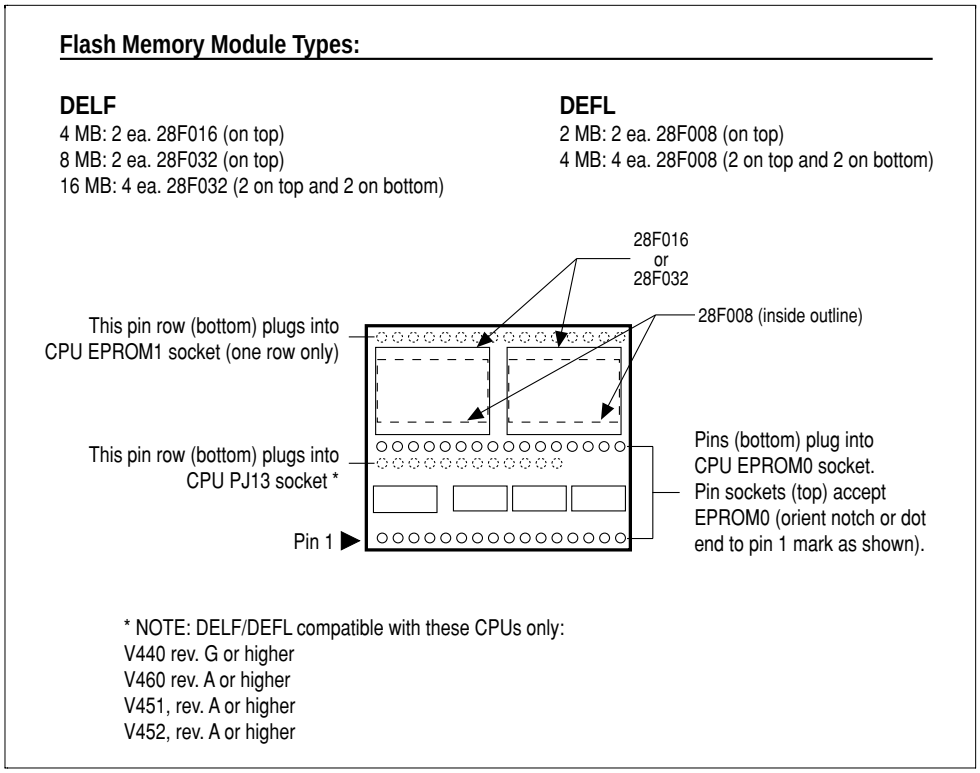
Address	Data width	Description
FD00 0000 – FDFF FFFF	D8	Flash module/EPROM1 read/write*

* NOTE: This address space is also recommended for EPROM1 for newer motherboards without the Flash module option. The old EPROM1 space FE400000–FE4FFFFF, however, is available for EPROM1 or regular Flash EPROM if necessary. See ***Accessing the Flash module*** below for more information.

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

Note Because the Flash module takes over the function of EPROM1, you must program the Flash module with the reset vector for CPU-Y (applies to dual-CPU boards only).

The drawing below summarizes the Flash memory module.



Flash memory module (Top View)

Installing the Flash module option

Follow these steps to install the Flash module:

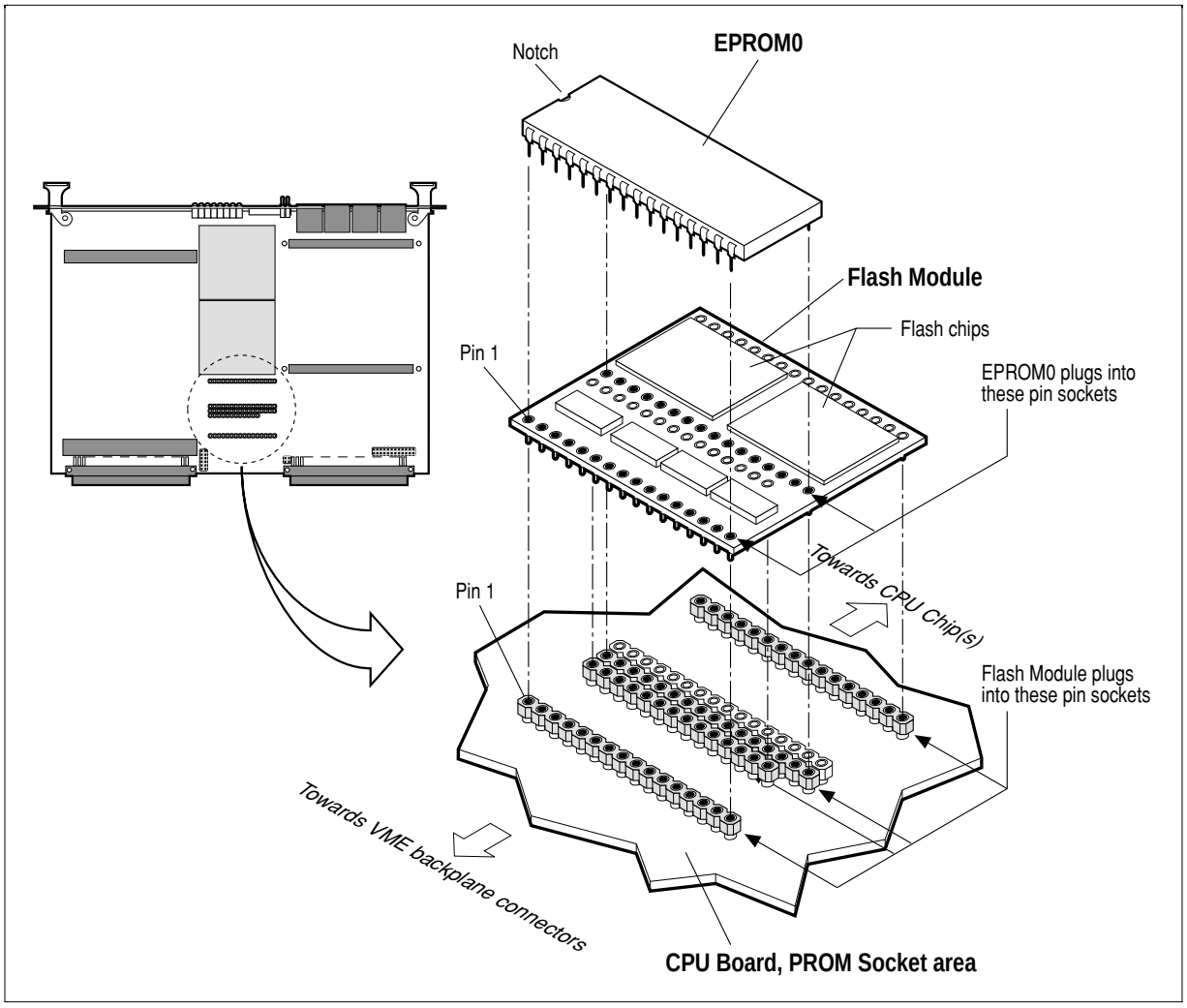
1. Remove EPROM0 and EPROM1 (if present) from motherboard. EPROM1 will no longer be used since the Flash module will contain the reset vectors for CPU-Y (if applicable).

2. Plug Flash module into motherboard sockets as shown in drawing below. Note orientation of Flash chips towards CPU chip(s). Make sure that all pins are properly engaged before fully seating module.

3. Plug EPROM0 into pin sockets on top of Flash module. Orient EPROM's notch or dot end to pin 1 socket as shown in drawing below.

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

4. Set appropriate EPROM configuration jumpers. For JK12 (PROM control), set EPROM1 jumpers to **Flash** and EPROM0 jumpers to **Normal** or **Flash** depending on the type of EPROM0 device installed on Flash module. For JK17 (Flash Write enable/disable) set the jumper to the **Write** position if it is desired to write to Flash. Refer to the ***EPROM*** chapter in Section 4 for more information on the EPROM jumpers.



Installing the Flash module

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

Accessing the Flash module

All Flash module varieties are accessed in the address range FD000000–FDFFFFFF. This provides a 16 MB space. This address range is also the recommended space for EPROM1 on newer Synergy motherboards (includes V440 rev. G or higher and V460 rev. A or higher).

For backwards compatibility, EPROM1 and regular Flash EPROM (not the Flash module) can be accessed by code using the old EPROM1 space used by earlier Synergy motherboards. This space is the address range FE400000–FE4FFFFF.

For detailed information on programming the Flash module devices, refer to Intel's "28F016SA 16-Mbit FlashFile™ Memory User's Manual," Order Number 297372-001.

⚠ Use byte accesses with data cache disabled, for programming, and program verification. Do not use word or longword accesses.

Also note that the Flash module's byte-write program should have a timeout of >3 milliseconds. The Flash device programming time is variable. For more information, contact Intel:

1-800-628-8686

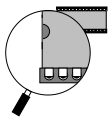
Block organization

For convenience in programming or erasing, the block information for the various Flash memory modules is listed in the table below.

Flash memory module block information

Product/Mem. Size	Total Blocks	Block Size	Block Numbers
DELF/4MB	64	64KB	0-63
DELF/8MB	128	64KB	0-127
DELF/16MB	256	64KB	0-255
DEFL/2MB	32	64KB	0-31
DEFL/4MB	64	64KB	0-63

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



Flash considerations

When Flash Write is enabled — Your application code must contain safeguards when Flash Write is enabled to avoid inadvertent writing of Flash memory which may render the processor(s) inoperative. In particular, CPU-Y will not boot after a reset if the reset vectors have been overwritten.

> ⚠ If Flash memory is to be long-term, read-only, set the JK17 Flash Write enable/disable jumper to the **disable** position to avoid accidental writing of Flash memory.

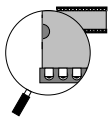
When programming Flash on a dual-CPU board — On a dual-CPU board, CPU-Y might interfere with attempts to reprogram the Flash by attempting to boot itself. In this case, your programming code must first prevent CPU-Y from making EPROM1 accesses. This is done with the ***EPROM1 programming*** bits provided in the extended control register (FE3A0006-7). The basic procedure is:

1. Write to FE3A0007. Setting this bit prevents CPU-Y from accessing EPROM1.
2. Program the Flash. At this point, CPU-X can program the Flash unhampered.
3. After programming, restart the dual-CPU board to resume normal operation. After restart, CPU-Y defaults to normal access of EPROM1 (FE3A0006 = 1).

For more information on the V452 control registers, refer to the ***V452 internal registers*** chapter in Section 3.

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

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



Onboard Flash memory

Introduction

V452 boards provide **2/4/8MB** of onboard Flash memory as an option. The memory chips have a byte write and block erase architecture with data storage similar to that of a sectored hard disk. The onboard Flash option lets the SBC load an operating system from local Flash for operation as a diskless or standalone (non-networked) unit.

Onboard Flash memory address location

Address	Data width	Description
FC00 0000 – FC7F FFFF	D32	Onboard Flash memory space

Flash memory is made up of 4 ea. 8-bit flash memory chips. These devices are connected so that they appear as one 32-bit device. The table below lists the device used for each size of onboard Flash.

Onboard Flash devices

Onboard Flash size	Manufacturer[1]	Part Number[1]	Organization
2MB	AMD	29F016	2Mb x 8
4MB	Intel	28F008	1Mb x 8
8MB	AMD	29F040	512Kb x 8

Note: 1. Equivalent parts may be used; therefore, actual manufacturer and part number may differ from the listing.

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

Writing and erasing

The onboard Flash is always write-enabled. Take special care when using the onboard Flash to avoid overwriting data.

Note The Flash write enable/disable jumper (JK17) applies to Flash installed in the EPROM sockets only.

Writing to onboard Flash may fail if JK12 is configured improperly.

The Flash memory chips have embedded byte write and block/sector erase algorithms. For more information on the chip itself and on the software aspects of writing/erasing Flash memory, refer to the applicable manufacturer's (AMD or Intel) Flash memory databook.

- Flash Memory (2-volume set), Intel
  For ordering information, contact:

  Intel Literature Sales
  P.O. Box 7641
  Mt. Prospect, IL 60056-7641

  In U.S. and Canada, call toll free: (800) 548-4725

- Flash Memory Products Data Book/Handbook, AMD
  For ordering information, contact:

  Advanced Micro Devices, Inc.
  One AMD Place
  P.O. Box 3453
  Sunnyvale, CA 94088-3453

  In U.S. and Canada, call toll free: (800) 222-9323

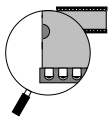
Block organization

For convenience in programming or erasing, the block information for the onboard Flash memory is listed in the table below.

Flash memory module block information

Flash Memory Size	Total Blocks	Block Size	Block Numbers
2MB	8	256KB	0-7
4MB	16	256KB	0-15
8MB	32	256KB	0-31

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



Note that Flash support and example driver code is supplied in Synergy's SMon Application Developer and Debugger package. Contact Synergy for details.

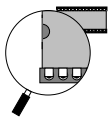
Booting from onboard Flash

The V452 can be configured to boot from onboard Flash via the Flash boot enable jumper at JK17 pins 17 & 18. Refer to the ***Jumpers, switches, LEDs & fuses*** chapter in Section 3 for more information on configuration jumpers.

To boot from onboard Flash, program the Flash memory with the reset vector and boot code, then install the Flash boot enable jumper on JK17 pins 17 & 18. With this jumper installed, a power cycle or local reset will cause the CPU to look for its reset vector at the base of onboard Flash (FC000000) instead of the EPROM. Refer to the ***Default & reset conditions*** chapter in Section 3 for more information about the boot state and start-up vectors.

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

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



Timers & counters

V452 Series boards provide a total of five timers/counters as follows:

- A 16-bit timer/counter is provided on each of the two 2692 DUART chips on V452 Series boards. Each of these timers can assert a Level 6 interrupt.
- Three additional timer/counters typically used for counter applications are provided by the 82C54 counter chip. Two of the three counters can assert Level 2 interrupts.

> Note
> For more information about these devices, see the *2692 DUART* and *82C54 counter* data sheets in the ***Datasheet Supplement*** containing reprinted portions of the manufacturer's data sheets.

The **2692** timers and **82C54** counters remain synchronized to support cascade-type applications.

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

Feature comparison

The timer/counters on the 2692 and 85C54 provide a slightly different set of features. The unique differences are explained below.

2692 timers and support circuitry:

- Both 2692 timers include a prescaler
- The 2692 counter/timer can be read on the fly, but only in the counter mode. Some means must be provided to protect against a carry from bit 7 to bit 8 during the reading of the upper and lower bytes of the 16-bit counter. The preferred method is to stop the counter, read the value and then to restart the counter. Other methods involving multiple reads can be used as protection against this carry over situation. Note that in the timer mode, the running count cannot be read under any circumstance.

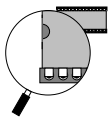
82C54 counters and support circuitry:

- All three 82C54 counters support *on-the-fly* counter reading.



The timer/counters on the 2692 and 82C54 chips can be configured to operate as either timers or counters. However, due to the characteristics of the chips themselves, as described above, and the interrupt levels assigned to them on V452 Series boards, the timer/ counters on the 2692 are typically used for timer functions; whereas those on the 82C54 are typically used for counter functions. For the sake of clarity, therefore, all references in this chapter to **timers** refer to the **2692**; whereas references to **counters** refer to the **82C54**.

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



16-bit timers (2692)

Each of the two 2692 DUARTs on the main board contains a 16-bit timer register that can be used for general-purpose timing applications.

- Timer **CT-A** resides on the DUART controlling serial channels A and B.
- Timer **CT-C** resides on the DUART controlling serial channels C and D.

The table below summarizes the address map locations for the registers on the 2692 chip that control timer/counter functions.

Timer-related registers on 2692 UARTs

Register	Address	Timer	Read function	Write function
1	FE280007	CT-A	—	Clock select register
4	FE280013	CT-A	—	Aux. Control register
6	FE28001B	CT-A	CT upper 8 bits	CT upper 8 bits
7	FE28001F	CT-A	CT lower 8 bits	CT lower 8 bits
9	FE280027	CT-A	—	Clock select register
14	FE28003B	CT-A	Start counter	Set Output
15	FE28003F	CT-A	Stop counter CT-A*	Reset Output
1	FE200007	CT-C	—	Clock select register
4	FE200013	CT-C	—	Aux. Control register
6	FE20001B	CT-C	CT upper 8 bits	CT upper 8 bits
7	FE20001F	CT-C	CT lower 8 bits	CT lower 8 bits
9	FE200027	CT-C	—	Clock select register
14	FE20003B	CT-C	Start counter	Set Output
15	FE20003F	CT-C	Stop counter CT-C*	Reset Output

Notes: * When operating in timer mode, the 2692's timer/counters cannot be stopped; only when in counter mode can they be turned on and off.



For more information about the other registers and serial functions for the 2692 chip, see the ***Asynchronous serial interface*** chapter in Section 5.

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

Selecting the timer mode and source

Bits 4-6 of the Auxiliary Control (ACR) register for each 2692 chip (see register 4 in the table on the previous page) selects either counter or timer mode and which source the 2692 is to count or time.

The table below lists the appropriate value to select each mode/source.

Timer mode and source selection

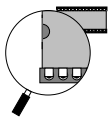
ACR register values Bits 4,5,6	Mode	Clock source
0 0 0	Counter	External (IP2)
0 0 1	Counter	TXCA - 1X clock of serial channel A
0 1 0	Counter	TXCB - 1X clock of channel B
0 1 1	Counter	Crystal or ext. clock/16
1 0 0	Timer	External (IP2)
1 0 1	Timer	External (IP2)/16
1 1 0	Timer	Crystal or ext. clock
1 1 1	Timer	Crystal or ext. clock/16.

Performing timer operations

The paragraphs below describe how to perform some basic counter/timer operations:

- **Start** counter/timer **A** by performing a byte read from register 0xFE28003B. **Stop A** (in Counter mode only) by performing a byte read from register 0xFE28003F.
- **Start** and s**top** counter/timer **C** in the same manner using register locations 0xFE20003B and 0xFE20003F respectively.
- **Access** the counter or timer value by performing two 8-bit reads or writes to register 6 (CT upper) and 7 (CT lower) for the desired DUART as listed in the table of 2692 register addresses appearing earlier in this chapter.

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



Enabling the timers as interrupt sources

The two 2692 timers are **steerable** interrupt sources which has different implications for single and dual-CPU V452 Series model boards:

- Single-CPU boards can use one timer as an independent interrupt source. (Using both timers as interrupt sources at the same interrupt level, is not recommended.)
- On Dual-CPU boards each timer can be assigned to serve as an interrupt source to either CPU but should be assigned to only one CPU at the same time.

Interrupt Control register configuration — Enabling the 2692 timers as interrupt sources is a two step process. The first step is to configure the Interrupt Control registers on the V452 Series board as described below:

To enable Timer A as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE390005      |Enable CT-A to CPU-X
```

To enable Timer C as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE390007      |Enable CT-C to CPU-X
```

To enable Timer A as an interrupt source to **CPU-Y** on a dual-CPU boards, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE398005      |Enable CT-A to CPU-Y
```

To enable Timer C as an interrupt source to **CPU-Y** on a dual-CPU boards, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE398007      |Enable CT-C to CPU-Y
```


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

Configuring timer interrupt level — The second step in enabling the 2692 timers is to configure the Output Configuration register on the 2692 chips themselves as described below:

The timer on each of the 2692 chips can be configured to assert an interrupt either at the same level as the serial interface (Level 5) or at its own level (Level 6) as follows.

To enable the 2692 timers to assert a separate Level 6 interrupt, execute the following 680x0 assembler instructions:

```
moveb     #04, 0xFE280037     |Enable CT-A as Level 6 int.
moveb     #04, 0xFE200037     |Enable CT-C as Level 6 int.
```

Writing any other value to these two bits causes the timer on the 2692 timer to assert a Level 5 interrupt.

Disabling the timers as interrupt sources

The timers can be disabled as interrupt sources by writing to the Interrupt Control registers as described below:

To disable Timer A as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb     #00, 0xFE390004     |Disable CT-A to CPU-X
```

To disable Timer C as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb     #00, 0xFE390006     |Disable CT-C to CPU-X
```

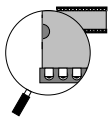
To disable Timer A as an interrupt source to **CPU-Y** on a dual-CPU boards, execute the following 68000 assembler instruction:

```
moveb     #00, 0xFE398004     |Disable CT-A to CPU-Y
```

To disable Timer C as an interrupt source to **CPU-Y** on a dual-CPU boards, execute the following 68000 assembler instruction:

```
moveb     #00, 0xFE398006     |Disable CT-C to CPU-Y
```


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



Timer interrupt service routine

To acknowledge a timer interrupt, toggle off and on the appropriate timer in the Interrupt Control register as described below.

To clear an interrupt from Timer A on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instructions:

```
moveb    #00, 0xFE390004    |Disable CT-A to CPU-X
moveb    #00, 0xFE390005    |Re-enable CT-A to CPU-X
```

To clear an interrupt from Timer C on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instructions:

```
moveb    #00, 0xFE390006    |Disable CT-C to CPU-X
moveb    #00, 0xFE390007    |Re-enable CT-C to CPU-X
```

To clear an interrupt from Timer A to **CPU-Y** on a dual-CPU boards, execute the following 68000 assembler instructions:

```
moveb    #00, 0xFE398004    |Disable CT-A to CPU-Y
moveb    #00, 0xFE398005    |Re-enable CT-A to CPU-Y
```

To clear an interrupt from Timer C to **CPU-Y** on a dual-CPU boards, execute the following 68000 assembler instructions:

```
moveb    #00, 0xFE398006    |Disable CT-C to CPU-Y
moveb    #00, 0xFE398007    |Re-enable CT-C to CPU-Y
```


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

16-bit counters (82C54)

The 82C54 counter on V452 Series boards provide three 16-bit counter registers for general-purpose timing applications.

- If enabled, **Counter 0** and **Counter 1** can be programmed to assert a Level 2 interrupt.
- **Counter 2** cannot assert an interrupt.

Each counter can operate independently in one of six operational modes to serve as event counters, elapsed time indicators, programmable one-shots, or other similar applications.

The address map location for the registers on the 82C54 are listed in the table below:

82C54 counter registers

Counter/ Register	Address
Counter 0	FE2A 0003
Counter 1	FE2A 0007
Counter 2	FE2A 000B
Control Word register	FE2A 000F

Selecting the counter mode and source

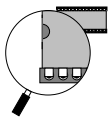
The 82C54 counters are programmed/started in two basic steps:

1. Writing a 8-bit control word to the Control Word register.
2. Writing the initial count into the counter itself.

The control word selects the counter and operational mode:

- Bits 7 and 6 select the Counter to be used (0, 1, or 2) to be configured or selects the read back command for reading the contents of a counter.
- Bits 5 and 4 select a read/write operation to be performed.

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



- Bits 3, 2, and 1 can select one of six operational modes. However, on V452 Series boards only the following three modes are useable:
  - **Mode 0** — Interrupt on terminal count; this mode can only be used as a 1 microsecond free-running counter. To use the counter in this manner without generating spurious interrupts, the counter interrupt must be enabled).
  - **Mode 2** — Rate generator
  - **Mode 4** — Software-triggered strobe

  Note Because V452 Series boards automatically generate an interrupt in response to a low-going pulse from the counters, only the modes listed above can be used. Attempting to use any of the other modes, while the counter is serving as an interrupt source can result in spurious interrupts. For more information about counter modes, see the *82C54 counter* datasheet in the ***Datasheet Supplement***.

- Bit 0 selects either binary or Binary Coded Decimal (BCD) as the output format for the counter.

  Note For more information about programming the 82C54 see the *82C54 counter* data sheet in the ***Datasheet Supplement***.

Enabling the counters as interrupt sources

The two counters on the 82C54 that can assert interrupts are **non-steerable** interrupt sources. This fact has different implications for single and dual-CPU V452 Series model boards:

- Single-CPU boards can use all three counters but only one as an interrupt source (Counter 0).
- On dual-CPU boards, each of the two interrupt driven counters can serve as an interrupt source only for the CPU to which they are assigned (i.e., CPU-X to Counter 0; CPU-Y to Counter 1).

To enable Counter0 as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

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

```
moveb       #00, 0xFE39000F     |Enable Counter0 to CPU-X
```

To enable Counter1 as an interrupt source to **CPU-Y** on a dual-CPU boards, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE39800F     |Enable Counter1 to CPU-Y
```

Disabling the counters as interrupt sources

The counters can be disabled as interrupt sources by writing to the Interrupt Control registers as described below:

To disable Counter0 as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE39000E     |Disable Counter0 to CPU-X
```

To disable Counter1 as an interrupt source to **CPU-Y** on a dual-CPU boards, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE39800E     |Disable Counter1 to CPU-Y
```

Clearing counter interrupts

To acknowledge a counter interrupt, toggle off and on the appropriate counter interrupt in the Interrupt Control registers as described below:

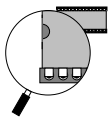
To clear an interrupt from Counter0 on single-CPU boards or to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instructions:

```
moveb       #00, 0xFE39000E     |Disable Counter0 to CPU-X
moveb       #00, 0xFE39000F     |Re-enable Counter0 - CPU-X
```

To clear an interrupt from Counter1 to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instructions:

```
moveb       #00, 0xFE39800E     |Disable Counter1 to CPU-Y
moveb       #00, 0xFE39800F     |Re-enable Counter1 - CPU-Y
```


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



Using Counter 2 as a BLT Throttle

While BLT DMA transfers are in progress, the V452 Series boards normally yield the entire local bus bandwidth to service the transfer. For transfers being controlled by an on-board CPU, this approach provides the fastest possible BLT DMA transfer rates without significantly compromising the CPU because it is *busy* managing the transfer and can effectively execute out of its own internal caches.

However, the V452 Series boards can also provide a special feature, using Counter 2 of the 82C54, to allow the on-board CPU(s) to perform local bus operations while an external processor performs BLT DMA transfers at the same time.

Such a concurrent processing arrangement can be achieved using the V452 Series board and an EZ-bus module that contains its own processor or intelligent DMA chip which can be programmed to operate the motherboard's BLT DMA engine.

Without some way to limit or throttle the bandwidth usage of the BLT transfer, however, having the intelligent EZ-bus module manage BLT DMA transfers on the V452 Series board might lock the on-board CPU(s) off its own local bus. Counter 2 of the 82C54 (whose count is available to the intelligent daughter module via pin D7 (CntC\) on the P4 EZ-bus connector) provides a local bandwidth *throttle* as described below.

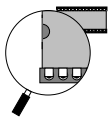
Whenever a BLT DMA is in progress, circuitry on the V452 Series board gates off or *freezes* Counter 2. As soon as an individual transfer ends, this circuitry frees Counter 2 to start counting. By programming the intelligent EZ-bus module to pause for a certain number of counts before starting the next BLT transfer, Counter 2 can open a variable-length, *bandwidth window* for local bus operations by the on-board CPU(s) while the EZ-bus module performs BLT DMA transfers.



The BLT throttling features of Counter 2 cause it to operate in a non-continuous fashion **whenever** BLT DMA transfers are in use. If you need continuous counting operations lasting more than the duration of a single BLT DMA transfer, consider using another 82C54 counter or one of the 2692 timers.

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

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



Clock calendar

The V452 Series provide clock/calendar data including the year, month, date, day, hour, minutes, and seconds data in 24-hour BCD format from a SGS-Thomson M48T18 Timekeeper RAM chip. The clock calendar is backed-up by a user-replaceable battery that should last for several years.

> For more information about this device see the *M48T18 Timekeeper RAM* data sheet in the ***Datasheet Supplement*** that contains reprinted portions of the manufacturer's datasheets for devices used on V452 Series and other boards from Synergy. This data sheet includes a series of formulas and curves for estimating the active life of the M48T18 chip's Snaphat battery.

The M48T18 also has its own Power-fail Detect circuit. The circuit deselects the device whenever Vcc is out of range, providing a high degree of data security during power-up and power-down times.

The V452 Series clock/calendar/NVRAM provides either 2K bytes or 8K bytes of storage. The 2K mode is provided for backwards compatibility to preserve the location of the clock/cal register set at FE10 07F8-FE10 07FF.

If your software requires compatibility with the V400 or V440 Series, then do not set the 8K mode bit. If backwards compatibility is not required, then it is recomended to set the 8K mode bit immediately on booting (before the NVRAM or clock/calendar are used).

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

To set the NVRAM to 8K mode, write a byte of data (value unimportant) to FE38 C00D. Doing so will also shift the data that was stored in the NVRAM from FE10 0000-FE10 07F7 up to the address range FE10 1800-FE10 1F7F.

To revert to 2K compatibility mode, write a byte of data (value unimportant) to FE38 C00C.

- FE38 C00C selects 2K mode (default)
- FE38 C00D selects 8K mode

Clock address locations

The M48T18 Clock/SRAM is an 8-bit peripheral. Each M48T18 memory location must be accessed on successive byte boundaries as illustrated in the table below.

Clock/calendar registers

Register address	Data bits 0-7								Range	
	b7	b6	b5	b4	b3	b2	b1	b0		
FE10 1FFF	--	--	--	--	--	--	--	--	Year	00-99
FE10 1FFE	0	0	0	--	--	--	--	--	Month	00-12
FE10 1FFD	0	0	--	--	--	--	--	--	Date	00-31
FE10 1FFC	0	FT	0	0	0	--	--	--	Day	00-07
FE10 1FFB	0	0	--	--	--	--	--	--	Hour	00-23
FE10 1FFA	0	--	--	--	--	--	--	--	Minutes	00-59
FE10 1FF9	ST	--	--	--	--	--	--	--	Seconds	00-59
FE10 1FF8	W	R	s	c	c	c	c	c	Control	

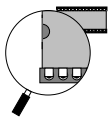
Notes: R = Read Bit
W = Write Bit
ccccc=calibration bits
ST = Stop Bit
FT = Freq. Test Bit
s = sign bit

Accessing clock data

Access to the clock is as simple as conventional byte-wide RAM access because the RAM and the clock are combined on the same die. The Timekeeper registers are located in the upper eight locations of the RAM as listed in the above table.

These registers contain, beginning from the top: year, month, date, day, hour, minutes, and seconds data in 24-hour BCD format. Corrections for leap year and the number of days in the month are made automatically. The eighth location is the Control register. These registers are not

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



the actual clock counters, but BiPort read/write static RAM memory locations. The 48T02 includes a clock control circuit that, once a second, dumps the counters into the BiPort RAM.

Clock operations

Updates to the Timekeeper registers should be temporarily suspended before clock data is read to prevent reading of data in transition. Because the BiPort Timekeeper cells in the RAM array are only data registers and not the actual counters, updating the registers can be suspended without disturbing the clock itself.

Updating the data registers is suspended when a 1 is written into the Read bit, the seventh most significant bit in the Control register. As long as a 1 remains in that position, data register updates are suspended. After the Read bit is set, the registers reflect the count, i.e., the day, date, and time that were current at the moment the Read command was issued. All of the Timekeeper registers are updated simultaneously. The Read command will not interrupt an update in progress. Registers are again updated in a normal fashion within a second after the Read bit is reset to a 0.

Setting the clock — The eighth bit of the Control register is the Write bit. Setting the Write bit to a 1, like the Read bit, suspends updates to the Timekeeper registers. The user can then load them with the correct day, date, and time data in 24-hour BCD format.

Resetting the Write bit to a 0 transfers those values into the actual Timekeeper counters and allows normal operation to resume. The FT bit, as well as the bits marked with zeros in the above table, must be written with zeros to allow normal Timekeeper and RAM operation.

Stopping and starting the oscillator — The oscillator may be stopped at any time. If the CPU board is going to spend a significant amount of time on the shelf, the oscillator can be turned off to minimize current drain from the battery. The STOP bit is the MSB of the Seconds register. Setting it to a "1" stops the oscillator.

To start the oscillator, implement the following procedure.

❶ Set the Write bit to "1".
❷ Reset the Stop bit to "0".
❸ Reset the Write bit to "0".
❹ Wait two seconds.

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

❺ Set the Write bit to "1".
❻ Set the correct time and date.
❼ Reset the Write bit to "0".

Calibrating the clock speed — The low-order 5 bits of the control register (ccccc in the table above) represent any value between 0 and 31 in binary form. The sixth bit is a **sign** bit (the **s** bit in the table above) where:

- **S=1** indicates a positive calibration and speeds up the oscillator.
- **S=0** indicates a negative calibration and slows down the oscillator.

Calibration corrections are applied within a 64 minute cycle. The first 62 minutes in each 64 minute cycle may, once per minute, have one second either shortened or lengthened by:

128/32768 seconds (3.906 ms)

If a binary 1 is loaded into the ccccc bits, only the first 2 minutes in the 64 minute cycle will be modified; if a binary 6 is loaded, the first 12 minutes of the 64 minute cycle will be affected, and so on. If the oscillator is running precisely at its nominal frequency (32768 Hz), each of the 31 increments in the calibration bits represents 5.35 seconds per (average) month, or, more precisely, 175.78 ms per day. This affords a total calibration range of about 5.4 seconds per day.

The simplest and most accurate method to calibrate the clock:

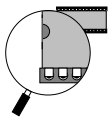
❶ **Synchronize** the clock to an accurate timing source such as a GPS receiver or WWV radio transmissions from the National Bureau of Standards in Fort Collins Colorado (available at 5,000 kHz, 10,000 kHz, and 15,000 kHz on the AM band).
❷ **Accumulate** an error for a few weeks or months if necessary.
❸ **Compare** the clock to the original source.

This procedure yields an accurate correction. Even a manual comparison, which has an error of a second or more, is sufficient to adjust the clock to within a single count of the calibration register.

Timer code example

See the ***Code examples*** in Section 6 for an example of how to set and use the M48T18 calendar chip.

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



Non-volatile 8K x 8 SRAM

Each SRAM location must be accessed on successive byte-aligned boundaries in the address range shown in the table below:

Non-volatile SRAM address location

Address	Data width	Description
FE1F E000 – FE1F FFF7	D8	8K bytes of battery-backed SRAM



Because the registers on the 48T18 are constructed using BiPort memory cells, access to the SRAM portion of the chip proceeds unhindered by updates to the clock calendar registers, even if these registers are being updated simultaneously with an SRAM access.

The V452 Series clock/calendar/NVRAM provides either 2K bytes or 8K bytes of storage. The 2K mode is provided for backwards compatibility to preserve the location of the clock/cal register set at FE10 07F8-FE10 07FF.

If your software requires compatibility with the V400 or V440 Series, then do not set the 8K mode bit. If backwards compatibility is not required, then it is recomended to set the 8K mode bit immediately on booting (before the NVRAM or clock/calendar are used).

To set the NVRAM to 8K mode, write a byte of data (value unimportant) to FE38 C00D. Doing so will also shift the data that was stored in the NVRAM from FE10 0000-FE10 07F7 up to the address range FE10 1800-FE10 1F7F.

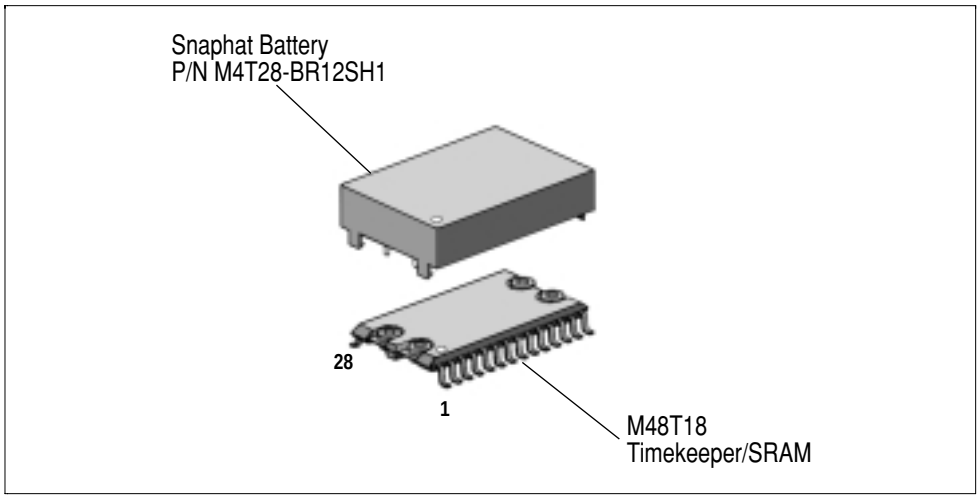
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To revert to 2K compatibility mode, write a byte of data (value unimportant) to FE38 C00C.

- FE38 C00C selects 2K mode (default)
- FE38 C00D selects 8K mode

Replaceable battery

The replaceable battery (p/n M4T28-BR12SH1) is a Snaphat type with two latches on each end that secure it onto the M48T18 chip. To remove, use a small screwdriver to pry the battery off the chip. Snap in the replacement battery matching the orientation of the chip pin 1 to the dot marking on the battery. The battery has a key tab to ensure proper mating to the chip. See drawing below.

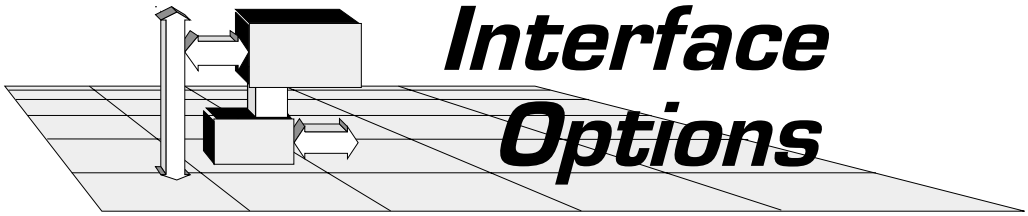


Timekeeper battery removal/installation

For your convenience, one source for the Snaphat battery is listed below:

Mouser Electronics
Phone: 1-800-34MOUSE
Stock No.: 511-M4T28BR12SH1
Unit price (as of Oct. 1998): $3.10 (USD)

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



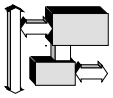
Interface Options

5

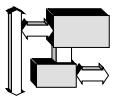
This section contains in-depth information about the architecture and function of the V452 Series I/O, peripheral, and bus interfaces.

- Asynchronous serial interface
- Ethernet 10Base-T interface option
- EZ-bus interface
- VME Slave interface
- Data broadcasting
- VME Master interface
- VME Master BLT
- System controller

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



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



Asynchronous serial interface

The V452 Series boards provide four high-speed, programmable asynchronous channels (**A**, **B**, **C**, & **D**) terminating in the three RJ-45 jacks on the front panel (channels B & D share a jack).

Two Signetics/Motorola 2692 style DUARTs, or Dual Universal Asynchronous Receiver/Transmitters control serial I/O functions. Each 2692 provides two fully-programmable asynchronous serial channels and one 16-bit timer with prescaler.



For more information about using the 2692's counter/timer, see the chapter on *Timers & counters*. For more information about programming the 2692, see the *2692 DUART* data sheet in the ***Datasheet Supplement*** that contains reprinted portions of the manufacturer's data sheets for devices used on V452 Series and other boards from Synergy.

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

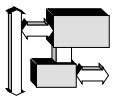
Each 2692 device contains sixteen registers. The table below lists these registers:

Asynchronous serial interface registers

Register	Address	Channel	Read function	Write function
0	FE280003	A	Mode register	Mode register
1	FE280007	A	Status register	Clock Select register
2	FE28000B	A	- reserved -	Command register
3	FE28000F	A	Receive register	Transmit register
4	FE280013	-	Input port change reg.	Aux. Ctrl. register
5	FE280017	-	Interrupt status register	Interrupt mask
6	FE28001B	-	Counter A - upper 8 bits *	Counter A - upper 8 bits *
7	FE28001F	-	Counter A - upper 8 bits *	Counter A - upper 8 bits *
8	FE280023	B	Mode registers	Mode registers
9	FE280027	B	Status Reg.	Clock Sel. register
10	FE28002B	B	- reserved -	Command register
11	FE28002F	B	Receive Reg.	Transmit register
12	FE280033	-	- reserved -	- reserved -
13	FE280037	-	Input port	Output Port Config.
14	FE28003B	-	Start Counter A *	Set Output
15	FE28003F	-	Stop Counter A *	Reset Output
0	FE200003	C	Mode register	Mode register
1	FE200007	C	Status register	Clock Sel. register
2	FE20000B	C	- reserved -	Command register
3	FE20000F	C	Receive register	Transmit register
4	FE200013	-	Input port change reg.	Aux. Ctrl. register
5	FE200017	-	Interrupt status register	Interrupt mask
6	FE20001B	-	Counter C - upper 8 bits *	Counter C - upper 8 bits *
7	FE20001F	-	Counter C - upper 8 bits *	Counter C - upper 8 bits *
8	FE200023	D	Mode register	Mode registers
9	FE200027	D	Status register	Clock Sel. register
10	FE20002B	D	- reserved -	Command register
11	FE20002F	D	Receive register	Transmit register
12	FE200033	-	- reserved -	- reserved -
13	FE200037	-	Input port	Output Port Config.
14	FE20003B	-	Start Counter C *	Set Output
15	FE20003F	-	Stop Counter C *	Reset Output

Notes: * These locations pertain only to timer functions on the 2692.
For more information, see the chapter on *Timers & counters.*

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



Back-to-back access protection

The 2692 needs to "rest" for a few hundred nanoseconds between accesses. As a result, the CPU can perform accesses much faster than the 2692 chip can handle them.

V452 Series boards contain hardware that automatically provides protection for back-to-back accesses.

Controlling serial interface interrupts

The two 2692 serial interfaces are **steerable** interrupt sources which have different implications for single and dual-CPU V452 Series model boards:

- Single-CPU boards can use both interfaces (four ports).
- On Dual-CPU boards each interface can be assigned to serve as an interrupt source to either CPU.

Enabling the serial ports as interrupt sources

To enable serial ports A and B as an interrupt sources on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb     #00, 0xFE390001     |Enable serial A&B to CPU-X
```

To enable serial ports C and D as interrupt sources on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb     #00, 0xFE390003     |Enable serial C&D to CPU-X
```

To enable serial ports A and B as an interrupt sources to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb     #00, 0xFE398001     |Enable serial A&B to CPU-Y
```

To enable serial ports C and D as interrupt sources to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb     #00, 0xFE398003     |Enable serial C&D to CPU-Y
```


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

Disabling the serial ports as interrupt sources

To disable serial ports A and B as an interrupt sources on single-CPU boards or to **CPU-X** on a dual-CPU board, reset the board or execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE390000      |Disable serial A&B to CPU-X
```

To disable serial ports C and D as interrupt sources on single-CPU boards or to **CPU-X** on a dual-68060 board, reset the board or execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE390002      |Disable serial C&D to CPU-X
```

To disable serial ports A and B as an interrupt sources to **CPU-Y** on a dual-CPU board, reset the board or execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE398000      |Disable serial A&B to CPU-Y
```

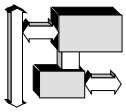
To disable serial ports C and D as interrupt sources to **CPU-Y** on a dual-CPU board, reset the board or execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE398002      |Disable serial C&D to CPU-Y
```

P2 access to serial interface

If desired, V452 Series boards can be configured (via special modification) to present signals from Ports A and B of the asynchronous serial interface to the VMEbus P2 connector. For more information about the required modification and a pinout for where the serial interface signals will appear on the P2, see the ***P2 serial interface option*** chapter in Appendix A.

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



Ethernet 10Base-T interface option

Introduction

Ethernet is a LAN (local area network) architecture that provides the means for computers and other peripherals located in a moderately sized geographical area to communicate with each other at high speed.

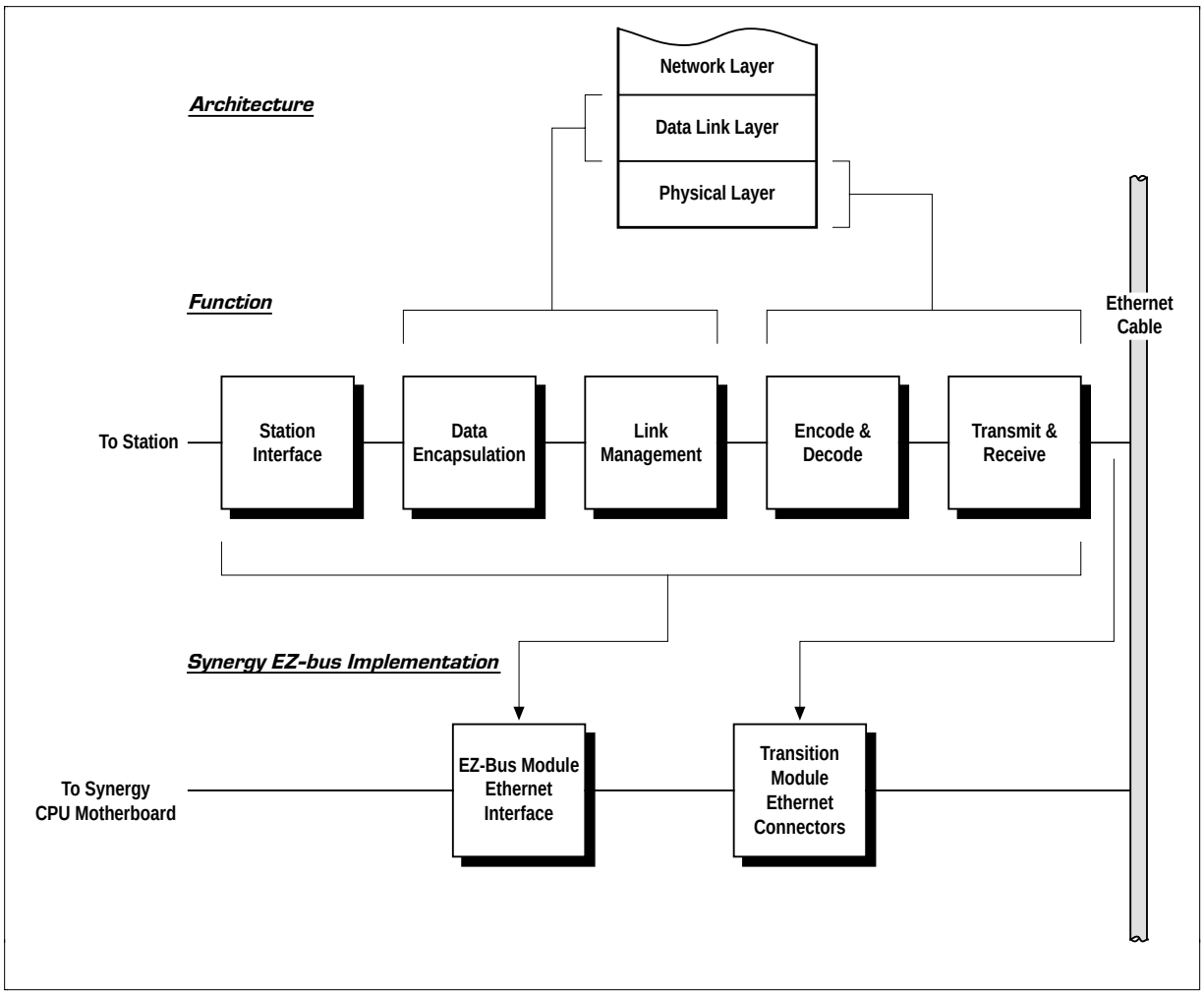
The Ethernet prototype was developed by Xerox Corporation in 1975 and grew to a standard LAN specification 10 years later (IEEE 802.3-1985) with the collaborative efforts of Digital Equipment Corporation, Intel Corporation, and Xerox Corporation.

Ethernet provides what is called a "link level" facility since it deals with the lowest two layers of network architecture as defined by the ISO Model for Open Systems Interconnection: the Physical Level and the Data Link Layer.

With Ethernet, the type of data it transmits is immaterial since it does not concern itself with data protocol and interpretation. As such, Ethernet LANs are used for various types of computing platforms such as mainframe computers, Macintoshes, IBM PC and compatibles, SPARC workstations, and UNIX systems.

The figure below diagrams the Ethernet architecture and functional blocks and their relationship to the V452 Ethernet interface.

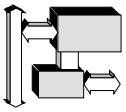
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Ethernet Architecture and Functional Blocks

The V452 Ethernet interface is based on the Intel 82C596 intelligent Ethernet coprocessor. The 82C596 can be configured to support all ANSI/IEEE 802.3-1985 specifications. The 82C596's DMA controller lets the V452 Ethernet interface access the motherboard bus and memory with little or no CPU intervention.

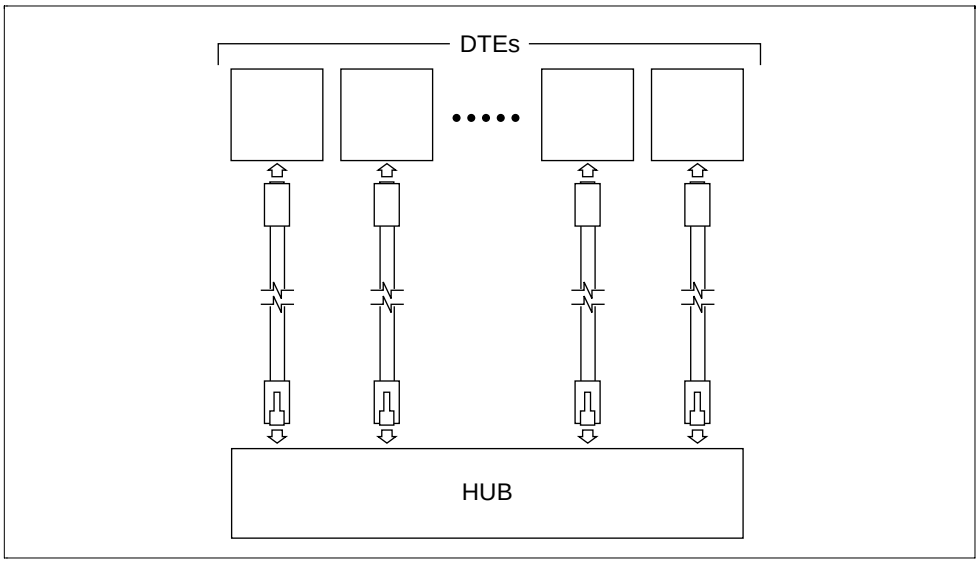
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Ethernet network connections

The V452 provides an Ethernet 10Base-T port at the front panel. A 10Base-T system is a star topology network in which each DTE (data terminal equipment) is connected to a shared hub through a single, 4-pair unshielded twisted pair (UTP) cable. The UTP cable is similar to modular telephone cable. For network use, however, a higher grade (or category) of cable is typically used. Category 3 is the minimum for 10Base-T, but Category 4 or 5 is more often recommended. Cable connections are made to an 8-pin RJ-45 modular jack. The maximum distance between DTE and hub is 100 m (328 ft.).

The figure below shows a typical 10Base-T single hub network.



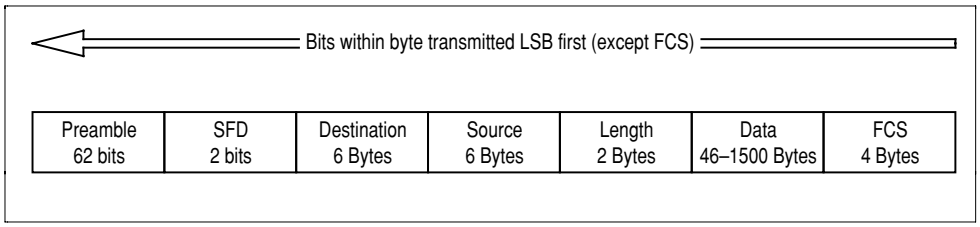
10Base-T single hub network

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

Data transmission

Both clock and NRZ data information is Manchester-encoded in bit-serial form and encapsulated in a basic unit called a frame packet.

The frame packet is made up of seven fields in which the data field is bracketed by several bytes of information. The figure below shows the format of an Ethernet frame.



Ethernet Frame Packet Format

The packet fields are summarized below.

Preamble — is a series of alternating 1's and 0's that serve to synchronize the clock and other circuitry on all the receivers and repeaters on the network.

Start of Frame Delimiter (SFD) — consists of two consecutive 1's to signal the start of a frame.

Destination — six bytes to indicate the destination of the packet on the network.

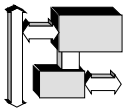
Source — six bytes to indicate the node that sent the packet.

Length — two bytes to indicate the number of bytes contained in the data field.

Data — 46–1500 data bytes. Stations that need to send less than 46 bytes of data must pad the data to reach the minimum requirement. Stations that need to send in excess of 1500 bytes of data must send multiple frame packets.

Frame Check Sequence — CRC value of packet (not including preamble and SFD fields) for error detection. Receiver rejects the frame if the calculated CRC value of the received data does not match the transmitted CRC value.

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



Ethernet ID or physical address

An Ethernet board is typically designed with a unique Ethernet ID (also called physical address) in ROM; by default any Ethernet packet sent to this ID will be received by the board and passed to the host. Packets addressed to other Ethernet IDs will be seen by the board, but ignored (by default).

The Ethernet ID is a 12-digit number. This number is made up of three bytes of manufacturer's ID followed by another three bytes of a unique identifier number. The Ethernet ID is what's contained in the Destination and Source fields of the Ethernet packet.

For Synergy boards, Synergy's 3-byte manufacturer's ID (00:80:F6) is compiled into the Ethernet driver code as a macro. The second half of the Ethernet ID is made up of the 6-digit SBC serial number which is stored as 3 bytes of BCD in these NVRAM locations:

- NVRAM address 0xFE10_0778: single processor, CPU-X
- NVRAM address 0xFE10_0774: dual processor, CPU-Y

Synergy's 3-byte manufacturer's ID is combined with the board serial number to produce the Ethernet ID of the board's Ethernet interface. For example, for a board serial number of '123456', the Ethernet ID is "00:80:F6:12:34:56".

For more information on the V452 non-volatile SRAM, refer to the ***Non-volatile 8K x 8 SRAM*** chapter in Section 4.

Avoiding bus contention — CSMA/CD

To avoid contention from two or more stations trying to talk at the same time on the network, Ethernet uses a media access method called CSMA/CD (Carrier Sense Multiple Access with Collision Detection). With CSMA/CD, a station transmits a frame only when the network is not busy. If a collision does occur after a transmission, the station resolves it by retransmitting the frame.

- To avoid contention, stations monitor a carrier signal (an encoded clock signal integrated with the data) that indicates whether or not another station is transmitting. If a station has data of its own to transmit and the network is not busy, it is sent immediately. Otherwise, if the network is busy, the station waits until it senses no activity plus an extra delay time padding for channel recovery before transmitting its own data.

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

- When a collision does occur, all stations are notified of the occurrence by a signal applied to their Collision Detect input. Any station that is currently transmitting must stop and wait a certain amount of time before retransmitting the frame. The station's location on the network is factored into the time delay to ensure that no overlap occurs with other stations that may also be retransmitting their data. A packet less than the minimum size (512 bits) is considered a collision remnant and is ignored by the receiving station.

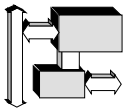
Interchange signals

Ethernet uses differential driver circuits for its interchange signals. For the onboard 10Base-T interface, the transmit data and receive data signals are transformer coupled internally on the SBC and routed to the front panel RJ-45 jack. The table below lists the interchange signals and their pin assignments on the RJ-45 jack.

P8, RJ-45 Pin	IEEE 802.3 Name	Function	Signal from:
1	DO+ (Data Out +)	Transmit Pair	DTE
2	DO– (Data Out –)		DTE
NC	DO S (DO Shield)		DTE
3	DI+ (Data In +)	Receive Pair	MAU
6	DI– (Data In –)		MAU
NC	DI S (DI Shield)		DTE
NC	CO+ (Control Out +)	Optional Pair	DTE
NC	CO– (Control Out –)		DTE
NC	CO S (CO Shield)		DTE
NC	CI+ (Control In +)	Collision Detect	MAU
NC	CI– (Control In –)		MAU
NC	CI S (Shield)		DTE
NC	VC (Voltage Common)	Power Pair	DTE
NC	VP (Voltage Plus)		DTE
NC	VS (Voltage Shield)		DTE
NC	PG (Protective Gnd)		DTE

DTE = Data Terminal Equipment
MAU = Medium Attachment Unit (Transceiver)
NC = no connection

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



Address map

The table below lists the address map and control register locations of the onboard Ethernet 10Base-T interface.

Ethernet interface address map (82C596)

Address	Width	Reg Number	Read Access	Write access
FE3B 8000	D32	EthPrt	—	Ethernet interface
FE3B C000	D32	EthCA	—	Ethernet Channel Attention

Ethernet/VMEbus control registers (FE3A 4000–F)

Function	Write to: (default)	Write to:
Ethernet Int. to CPU-X	FE3A 4000 Disable	FE3A 4001 Enable
Ethernet Int. to CPU-Y	FE3A 4002 Disable	FE3A 4003 Enable
Ethernet Jabber	FE3A 4004 Enable	FE3A 4005 Disable
Auto Polarity/Enhanced Squelch	FE3A 4006 Enh. Squelch	FE3A 4007 Auto Polarity
VMEbus Request Level	FE3A 400A |— Level 3 FE3A 4008 |	FE3A 400A |— Level 2 FE3A 4009 |
		FE3A 400B |— Level 1 FE3A 4008 |
		FE3A 4009 |— Level 0 FE3A 400B |
– Reserved – Bits FE3A 400C–F		

* Note: Write to the indicated address pair for the bus request level shown.

Interrupts and vectors

The table below lists the interrupt levels and vectors for the onboard Ethernet 10Base-T interface.

Ethernet interface interrupts

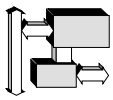
Interrupt source	Interrupt priority level	Vector number (Hex)
Ethernet interface	4	0x45

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

82C596 programming details

The most useful excerpts from the manufacturer's data sheet for the 82C596 data sheet can be found in the *82C596 datasheet* chapter in the ***Datasheet Supplement*** manual. Refer to this manual for details about the registers and programming of the 82C596.

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



EZ-bus interface

The Synergy EZ-bus is a flexible, interrupt-driven interface allowing Synergy CPU boards to accept connection of one or two independent daughter modules. Most modules are **half-size** and occupy the **same slot** as the motherboard when connected.

The 32-bit EZ-bus can support up to **60 MB/s** data transfers as a bus Master or Slave.

The EZ-bus interface provides a DMA capability so that daughter modules that are able to serve as bus Masters can transfer data directly to/from the V452 Series on-board triple-ported Dynamic RAM without CPU intervention.



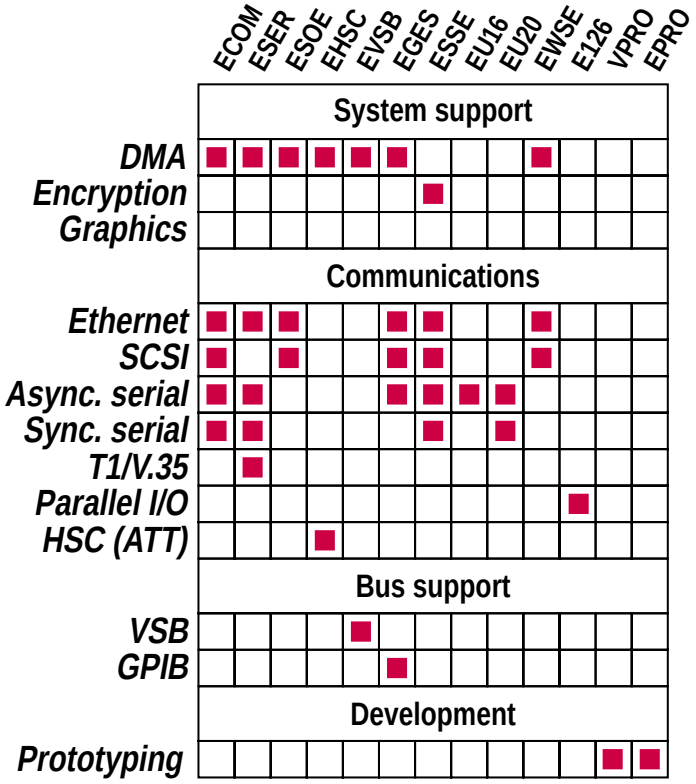
Whenever an external EZ-bus Master writes to a portion of the V452 Series memory that is being cached by the on-board CPU, a cache-to-memory incoherency condition is possible that could result in a loss of data. For a discussion of these cache coherency considerations and a summary of some useful cache management techniques, refer to the chapter on the *68060 CPU*.

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

EZ-bus modules

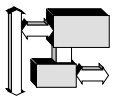
The following EZ-bus modules are interchangeable among all of Synergy's VMEbus CPU board product lines. The table below summarizes the major features provided by each module. The paragraphs that follow briefly describe each of the EZ-bus modules listed in the table. Half-size modules, which occupy the same slot as the CPU board, are identified.

EZ-bus module feature selection chart



- **E126** — provides a high speed parallel port. (half-size)
- **ECOM** — provides concurrent central I/O processing and 10 DMA channels. Includes SCSI-2/SCSI-1, Ethernet, and four serial asynchronous/synchronous channels all with DMA. (half-size)
- **EGES** — provides an interface between 680x0 and the General Purpose Interface Bus (GPIB). The board is compatible with IEEE-488, concurrent central I/O processing, 2 DMA channels and support for SCSI-2/SCSI-1 and Ethernet/Thinnet (half-size)

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



- **EHSC** — provides a 16-bit, 10 MB/sec, parallel port conforming to the AT&T High-speed Channel (HSC) specification. (half-size)
- **ESOE** — provides a high performance fast, wide SCSI-2 interface and a 10Base-T Ethernet interface supporting up to 8 channels, all with DMA support. (full-size)
- **VPRO** — is a wire-wrap board containing all required interface connectors for prototyping new EZ-bus modules. It includes the *EZ-bus Designer's Guide* describing the design features and characteristics for the EZ-bus. (full-size)
- **EPRO** — is a wire-wrap board containing all required interface connectors for prototyping new EZ-bus modules. It includes the *EZ-bus Designer's Guide* describing the design features and characteristics for the EZ-bus. EPRO is designed for development in a stacked, daughter board configuration in which an EZ-bus module is already installed in the module A position. (full-size)
- **ESER** — provides an Ethernet and four high performance universal synchronous/asynchronous serial interfaces all with DMA. The serial interfaces can be used in a wide variety of applications including T1/E1 digital phone channels. (half-size)
- **ESSE** — provides SCSI, Ethernet, two asynchronous serial ports, eight asynchronous serial ports, and DES data encryption. Alternate versions are also available containing SCSI only, Ethernet only, or serial only. (half-size)
- **EU16** — provides 16 asynchronous serial RS-232 channels featuring Clear-to-Send (CTS) and Request-to-Send (RTS) signals. (half-size)
- **EU20** — provides 2 synchronous serial channels with full modem control signals and 18 asynchronous channels with Clear-to-Send (CTS) and Request-to-Send (RTS) signals. Supports various serial protocols. (half-size)
- **EVSB** — provides an interface and full support for the Motorola VME subsystem bus (VSB). (half-size)
- **EWSE** — provides a high performance wide SCSI-2 and Ethernet interface with DMA support. (half-size)
- **E???** — Order or create custom EZ-bus modules using the *EZ-bus Designer's Guide*.

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



New daughter modules are continuously being added to this list.

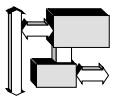
The ***EZ-bus Designer's Guide*** specifies the mechanical and electrical characteristics and requirements for EZ-bus modules. It also contains more detailed descriptions of all of the available EZ-bus modules.

This information is of interest to all EZ-bus module users and is essential for those who wish to design their own custom daughter modules. Synergy will design custom daughter modules for OEMs and quantity buyers. Please contact Synergy for more information.

EZ-bus connectors

The EZ-bus sockets (P3 & P4) are located on the P2 side of the board The EZ-bus sockets connect to rows A and C of P2 (and to rows D and Z if optional 160-pin VMEbus connectors are used), allowing connection either to I/O or to a side bus such as VSB or VMX. These sockets also connect to the CPU's local bus and power pins. For pinouts of the P3 and P4 EZ-bus connectors see Appendix A, ***Cables & Connectors***.

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



Controlling EZ-bus module interrupts

The two EZ-bus modules (if present) are **steerable** interrupt sources which have different implications for single and dual-CPU V452 Series model boards:

- Single-CPU boards can receive interrupts from both EZ-bus modules that may be installed.
- On Dual-CPU boards, each interface can be assigned to serve as an interrupt source to either CPU.

> Each EZ-bus module is assigned only one interrupt level (Level 4) on V452 Series boards. However, some EZ-bus modules contain more than one interrupt source. For modules with more than one source, interrupt arbitration is performed on the module, providing separate interrupt vectors for the different interrupt sources before being asserted to the motherboard. For more information, see the manual for the individual EZ-bus module you are using.

Enabling the EZ-bus modules as interrupt sources

To enable EZ-bus module A as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE390009      |Enable EZ-bus A to CPU-X
```

To enable EZ-bus module B as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE39000B      |Enable EZ-bus B to CPU-X
```

To enable EZ-bus module A as an interrupt source to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE398009      |Enable EZ-bus A to CPU-Y
```


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

To enable EZ-bus module B as an interrupt source to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE39800B      |Enable EZ-bus B to CPU-Y
```

Disabling the EZ-bus modules as interrupt sources

To disable EZ-bus module A as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE390008      |Disable EZ-bus A to CPU-X
```

To disable EZ-bus module B as an interrupt source on single-CPU boards or to **CPU-X** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE39000A      |Disable EZ-bus B to CPU-X
```

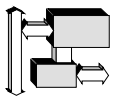
To disable EZ-bus module A as an interrupt source to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE398008      |Disable EZ-bus A to CPU-Y
```

To disable EZ-bus module B as an interrupt source to **CPU-Y** on a dual-CPU board, execute the following 68000 assembler instruction:

```
moveb       #00, 0xFE39800A      |Disable EZ-bus B to CPU-Y
```


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



VME Slave interface

The V452 Series provides on-board multi-ported RAM memory making it accessible by the on-board CPU(s), other VMEbus Masters, or by a EZ-bus daughter module Master that is mounted on the V452 Series board's expansion bus.

External VMEbus Masters access the V452 Series on-board RAM via the board's VME Slave interface. The V452 Series VME Slave interface recognizes both the A32 or A24 VME address modes and accepts D32, D16, and D8 data transfers. V452 Series boards can also accept both 64-bit (VME64) and 32-bit (VME32) VME block transfers (BLTs).

V452 Series boards can supply 4, 8, 16, 32, 64, 128, 256 or **512 MBytes** of dynamic RAM (last two sizes are special order items). They also provide software-controlled circuitry allowing the Slave access window to reside anywhere within the 4 GByte VMEbus addressing range.

V452 Series boards provide other software-programmable registers to configure and control other characteristics of the VME Slave interface. This chapter describes these controls and the available configuration options.



V452 Series boards include an interrupt-driven **CPU Mailbox** that can play a key role in inter-CPU communications. For more information, see the ***CPU Mailbox*** chapter in Section 4.

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

Setting up the VME Slave interface

The VME Slave interface for V452 Series boards can be programmed for various characteristics and operations. The Slave access window to the board can appear in VME Extended address space (**A32/D32** or **A32/D16**), VME Standard address space (**A24/D32 or A24/D16**) or may even be disabled as a Slave completely.

The following jumpers and software-accessible registers control the characteristics and operation of the VME Slave interface:

- Jumper **JK17** enables/disables VME Slave remote resets,
- The **Slave Interface Control register** (0xFE38 8000) controls several Slave interface characteristics and operational modes,
- The **Primary** and **Extended Control registers** (0xFE38 C000 & FE3A 0000) control other Slave interface characteristics and operational modes,
- The **Extended Mode register** (0xFE38 4003) enables/disables the VME Slave interface,

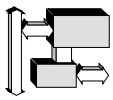
The paragraphs that follow describe the default characteristics of the Slave Interface, the configuration choices, and the way to implement the desired configuration.

Default Slave interface conditions

V452 Series boards provide the following default conditions for the VME Slave interface:

- VME Slave remote reset is **disabled** — This feature is enabled/disabled via jumper,
- VME Slave accesses are **disabled** — While in this disabled state, the remaining conditions appearing in the bullets below can be set but are not active,
- Slave write accesses (if enabled) are limited to Masters operating in 68000 Supervisor mode,
- If the Slave interface is enabled, the following default configuration is present as follows,
  - 32-bit (**A32**) Extended mode addressing
  - **0 MB** (x000 0000) Base address
  - Slave window **equal to installed memory**

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



- VME broadcasting is **disabled.**

> **Note** The list above describes the default conditions for the board. Default, in this case, refers to a board that contains no VME Slave interface jumpering and has not been initialized or booted by any software.

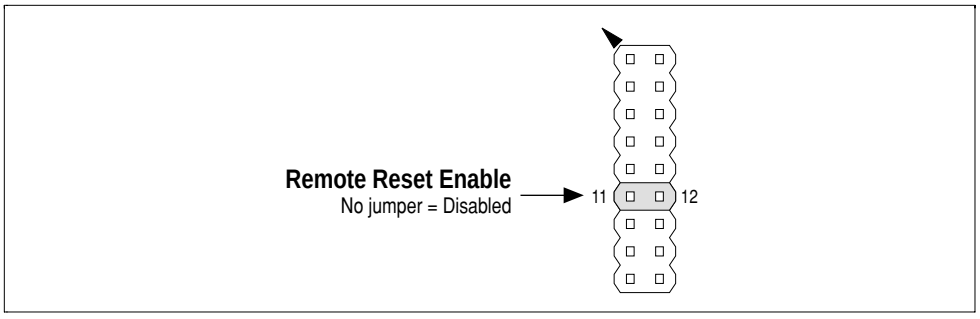
Enabling/disabling VME Slave remote reset

V452 Series board can support remote reset by Masters across the VMEbus. If this feature is enabled, another VMEbus Master can reset a specific V452 Series board by writing to the board's Slave reset registers.

The second to last 256 byte region of on-board RAM acts as a reset register. A Supervisor mode write to byte zero of this area (Slave Base Address + 3FFEX0 on a 4MB card) by another VMEbus Master resets the V452 Series board. Refer to the ***Reset via software*** discussion in the ***Default & reset conditions*** chapter for information on setting up and using remote reset.

The remote slave reset feature is enabled or disabled via jumper **JK17** as shown in the figure below.

Installing a shunt between pins **11** and **12** of jumper field **JK17** enables Remote Reset, while removing the shunt disables remote reset.



Jumper setting (JK17) — VME Slave remote reset.

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

Setting A32 or A24 VME addressing

V452 Series Slave interface can decode either 32-bit (A32) Extended Addressing or 24-bit (A24) Standard Addressing from a VMEbus Master. The Slave Interface Control register determines the current address mode setting.

> **Note** V452 Series boards do not support **A16** VME address mode accesses as a VME Slave but are able to generate 16-bit (A16) addresses as VMEbus Masters. For more information see the ***VME Master interface*** chapter in this section.

The default setting for V452 Series boards is A32 addressing. To set up a V452 Series board for A24 addressing, execute the following 680x0 assembler instruction (or include it in the board's boot/initialization code):

```
moveb          #0x00, 0xFE388001      | A24 addressing
```

To return to A32 addressing, either reset the board or execute the following instruction.

```
moveb          #0x00, 0xFE388000      | A32 addressing
```

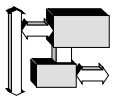
Configuring Slave write access memory protection

If enabled, the V452 Series Slave interface provides complete read access to all VMEbus Masters and one of the following three write access privilege levels:

- *Memory protect* — this level blocks all attempted write accesses to on-board RAM by external VMEbus Masters.
- *Supervisor only* — this level limits write accesses to Master processors currently operating in 68000 *Supervisor* mode. Write accesses from Master processors operating in 68000 *User* mode are blocked.
- *No memory protect* — this level grants write access privileges to all VMEbus Masters and disables all memory protection.

The table below shows the slave write access protection levels as set by the Primary Mode register and Slave Interface Control register.

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



Slave write access protection levels

Slave write access level	Memory protect control (Primary Mode register)	Supervisor-only control (Slave Interface Control reg.)
Memory protect (from all Slave write accesses)	ON	Don't care
Supervisor-only write accesses	OFF	ON
No memory protect	OFF	OFF

The paragraphs below describe these Slave write access protection levels in greater detail and the controls used to enable or disable them.



As shown in the matrix, the *memory protect* control setting in the Primary Mode register takes precedence over the *Supervisor-only* controls in the Slave Interface Control register. Thus, to set either *supervisor-only* or *no memory protect* level, *memory protect* must be turned OFF.

Memory write protect — this setting blocks all attempts by VMEbus Masters to write to on-board memory while still allowing memory read access. Slave write access is enabled/disabled by writing the appropriate hex value to the Primary Mode register. The hex values corresponding to this functions are listed below:

Primary Mode register [FE38 0003] — VME Slave functions

Hex data value	Function
05	Memory protect OFF — Allow Slave write accesses to RAM (***default)***
0D	Memory protect ON — Do NOT allow Slave write accesses to RAM

The default setting for the memory protect control in the Primary Mode register is OFF, which allows Slave write accesses to on-board memory. To turn ON Slave write access protection, execute the following 680x0 assembler instruction (or include it in the boot/initialization code):

```
moveb        #0x0D, 0xFE380003     | Turn memory protect ON
```

To turn OFF Slave write access protection (once it is ON), either reset the board or execute the following instruction.

```
moveb        #0x05, 0xFE380003     | Turn memory protect OFF
```


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



When Slave write access memory protection is ON, no Slave write accesses are allowed regardless of the settings made to the Slave Interface Control register. The Slave Interface Control register affects Slave write accesses only when memory protect is in the OFF or default state.

Supervisor-only vs. No memory protect — If Slave write accesses are allowed, (as described above) the V452 Series Slave interface allows you to control which Masters are allowed to write to Slave on-board memory as follows:

- **Supervisor-only — allows** write accesses from VME Masters whose processor is operating at that moment in 68000 **Supervisor** mode and **prohibits** write accesses from VME Masters whose processor is operating at that moment in 68000 **User** mode. Supervisor-only access is the **default** Slave write access condition for V452 Series boards.
- **No memory protect —** allows write accesses to all VME Masters.

The board's Slave interface circuitry determines the Master's operating mode by reading the Address Modifier (AM) codes that accompany each Slave address. Among other things these codes identify the impending data transfer as either a *Supervisory* access (i.e., originating from a Master in 68000 Supervisor mode) or a *Non-privileged* access (i.e., originating from a Master in 68000 User mode).

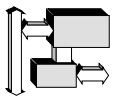
If the Slave write access memory protect is OFF (as described in the previous group of paragraphs) and Supervisor-only function is ON, the board's Slave interface allows all Supervisory accesses and block all user-mode write accesses it receives.

If the Slave write access memory protect is OFF (as described in the previous group of paragraphs) and Supervisor-only function is OFF, the board's Slave interface allows all write access from the VMEbus.

Presuming that the memory protect function is set in the default or OFF position (as described in previous paragraphs), the Supervisor-only function is enabled/disabled on V452 Series boards by writing to the Slave Interface Control register.

In the default condition, Supervisor-only accesses are enabled. To set up a V452 Series board to allow all VME Masters to access on-board

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



memory, execute the following 680x0 assembler instruction (or include it in the board's boot/initialization code):

```
moveb       #0x00, 0xFE388003     | No memory protect
```

To return to Supervisor-only Slave write accesses, either reset the board or execute the following instruction:

```
moveb       #0x00, 0xFE388002     | Supervisor-only enable
```

Setting the Slave window base address

The base address for the board's VMEbus Slave access window is set by writing to the Slave Interface Control (FE38 8000) and Extended Control registers (FE3A 0000).



V452 Series boards use the placement of the VME Slave access window to set up the board for VME data broadcasting. If you intend to use the data broadcasting features, please review the next chapter on *Data broadcasting* before setting up the Slave window location.

The table on the facing page is a worksheet to help you determine how to configure the Slave Interface Control and Extended Control locations to place the VME Slave access window in the desired location. The paragraphs below describe how to use this worksheet:

❶ Write the three most significant hexadecimal digits of the desired VMEbus Slave base address into the three empty boxes on the left side of the table. Write the most significant hex digit into the top box, the 2nd most significant digit into the middle box and the 3rd most significant digit into the bottom box.

❷ Translate each hexadecimal digit and write in its binary equivalent into smaller boxes in the second column to the left. Each of these boxes is associated with a particular bit in the VMEbus Slave base address as listed in the table.

The two columns on the right-side of the table, list the register locations that set each bit to either the "0" or "1" condition as required. To set the Slave base address, write the appropriate location for each of the binary values that you have written on the second column from the left.

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

When the board is reset, all the bits listed in the table are set to “0”. Thus, to set the Slave base address after booting the board, just write to the locations for the bit(s) that need to be set to “1”.

VMEbus Slave window base address configuration

Write in 3 most significant hex digits of desired VME Slave base address [1]	Write binary equivalent of each digit [2]	Don't care bit if mem. ≥X MB[5]	VME addr bit	To set this bit to "0" write to:[3]	To set this bit to "1" write to:
Most significant hex digit			Bit 31	FE38 800E [4]	FE38 800F
			Bit 30	FE3A 000E [4]	FE3A 000F
			Bit 29	FE3A 000C [4]	FE3A 000D
		512MB	Bit 28	FE3A 000A [4]	FE3A 000B
2nd most significant hex digit		256MB	Bit 27	FE3A 0008 [4]	FE3A 0009
		128MB	Bit 26	FE38 800C [4]	FE38 800D
		64MB	Bit 25	FE38 800A [4]	FE38 800B
		32MB	Bit 24	FE38 8008 [4]	FE38 8009
3rd most significant hex digit		16MB	Bit 23 [6]	FE38 8006 [4]	FE38 8007
		8MB	Bit 22 [7]	FE38 8004 [4]	FE38 8005
	N/A		Bit 21 [8]	N/A	N/A
	N/A		Bit 20 [8]	N/A	N/A

Notes:

1 Write the 3 most significant digits of the desired VME Slave base address in the three boxes in this column. Write the most significant hex digit in the top box, the 2nd most significant digit in the middle box, and the 3rd most significant digit in the bottom box.

2 Translate each hex digit into the equivalent 4-bit binary expression. Write this expression with the most significant bit at the top and the least significant at the bottom for each group of four bits.

3 V452 decodes only the address portion of accesses to these registers. The data value written does NOT matter.

4 After a power cycling or reset, the Slave address bits are set to all "0"s.Thus, if you are setting the VME Slave base address after boot-up, it is NOT necessary to write to any of the "0" register locations; it is only necessary to write to the "1" locations.
However, if changing the VME Slave window "on the fly" or after setting it previously, it is highly recommended that you specify the entire address by writing to both the "0" and "1" locations.

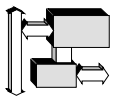
5 If local memory size is equal to or greater than value shown, the bit in next column is a "don't care" bit. If local memory size is less than this amount, write to the appropriate register location as required.

6 The slave window will be 16MB if this bit and bit 22 are both set to 0 and 8MB mode is OFF. This bit must be set to "0" if using data broadcasting. For more information see the next chapter.

7 The slave window will be 16MB if this bit and bit 23 are both set to 0 and 8MB mode is OFF. This bit must be set to "0" if using data broadcasting. For more information see the next chapter.

8 These bits are not configurable, the minimum slave window size is 4 MB.

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



For example, to set the base address for a VME Slave window to be:

7AC0 0000

would require writing in the 3 most significant hex digits in the far left column and then the binary equivalents in the second column resulting in a table that looks like the one shown below:

VMEbus Slave window base address configuration — EXAMPLE

Write in 3 most significant hex digits of desired VME Slave base address [1]	Write binary equivalent of each digit [2]	VME address bit	To set this bit to "0" write to:	To set this bit to "1" write to:
Most significant hex digit	0	Bit 31	FE38 800E [4]	FE38 800F
	1	Bit 30	FE3A 000E [4]	FE3A 000F
7	1	Bit 29	FE3A 000C [4]	FE3A 000D
	1	Bit 28	FE3A 000A [4]	FE3A 000B
2nd most significant hex digit	1	Bit 27	FE3A 0008 [4]	FE3A 0009
	0	Bit 26	FE38 800C [4]	FE38 800D
A	1	Bit 25	FE38 800A [4]	FE38 800B
	0	Bit 24 [5]	FE38 8008 [4]	FE38 8009
3rd most significant hex digit	1	Bit 23 [6]	FE38 8006 [4]	FE38 8007
	1	Bit 22 [7]	FE38 8004 [4]	FE38 8005
C	N/A	Bit 21 [8]	N/A	N/A
	N/A	Bit 20 [8]	N/A	N/A

To set up the Slave window base address at the location listed in the table, you would need to include the following instructions (shown in 680x0 assembler) in the intialization code for the board:

```
moveb   #0x00, 0xFE38800E  | Set bit A31 to "0"
moveb   #0x00, 0xFE3A000F  | Set bit A30 to "1"
moveb   #0x00, 0xFE3A000D  | Set bit A29 to "1"
moveb   #0x00, 0xFE3A000B  | Set bit A28 to "1"
moveb   #0x00, 0xFE3A0009  | Set bit A27 to "1"
moveb   #0x00, 0xFE38800C  | Set bit A26 to "0"
moveb   #0x00, 0xFE38800B  | Set bit A25 to "1"
moveb   #0x00, 0xFE388008  | Set bit A24 to "0"
moveb   #0x00, 0xFE388007  | Set bit A23 to "1"
moveb   #0x00, 0xFE388005  | Set bit A22 to "1"
```


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



The instructions shown on the previous page for address bits A31 (first line), A26 (sixth line), and A24 (eighth line) *are required* only if the state of these bits have been previously changed from the default "0" condition. For the sake of modularity, however, it may still be advisable to include an instruction for each address line as shown in this sample.

VME Slave memory-size-specific information/8 MB mode

4MB — This size uses slave address bits A22 to A31 to set the slave address on a 4MB boundary. However, when slave address is set to xx000000 in A32 address mode, then the slave window is 16MB, not 4MB.

Set the 8 MByte mode to OFF (default).

8MB — This size uses slave address bits A23 to A31 to set the slave address on an 8MB boundary. This is true for either A32 or A24 address modes.

Set the 8 MByte mode to ON. To do this, execute the following assembly-language instruction:

```
moveb    #0x00,0xFE3A0005    | sets slave to 8MB window size
```

16MB — This size uses slave address bits A24 to A31 to set the slave address on a 16MB boundary in A32 address mode. It is possible to access only one 4 MB segment of the DRAM in A24 mode. The accessible segment is that whose address matches bits A22 and A23 of the VME slave address window.

Set the 8 MByte mode to OFF (default).

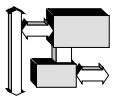
32MB — This size uses slave address bits A25 to A31 to set the slave address on a 32MB boundary in A32 address mode. A VME slave access in A24 mode produces undefined results.

Set the 8 MByte mode to OFF (default).

64MB — This size uses slave address bits A26 to A31 to set the slave address on a 64MB boundary in A32 address mode. A VME slave access in A24 mode produces undefined results.

Set the 8 MByte mode to OFF (default).

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



128MB — This size uses slave address bits A27 to A31 to set the slave address on a 128MB boundary in A32 address mode. A VME slave access in A24 mode produces undefined results.

Set the 8 MByte mode to OFF (default).

256MB — This size uses slave address bits A28 to A31 to set the slave address on a 256MB boundary in A32 address mode. A VME slave access in A24 mode produces undefined results.

Set the 8 MByte mode to OFF (default).

512MB — This size uses slave address bits A29 to A31 to set the slave address on a 512MB boundary in A32 address mode. A VME slave access in A24 mode produces undefined results.

Set the 8 MByte mode to OFF (default).

Enabling/disabling the Slave interface

V452 Series boards provide a *master switch* for the Slave interface that either turns ON or completely turns OFF the Slave interface under software control.

In the default condition the Slave interface is disabled. While disabled, the Slave interface on a V452 Series board is unable to respond to accesses from the VMEbus in any way, but can be configured using any of the methods described in the preceding paragraphs.

Once it has been set to the desired arrangement, the Slave interface must be enabled before it will actually be able to operate.

To enable it, write **0F** to the Extended Mode register at **0xFE38 4003** using the following assembler commands:

```
moveb          #0x0F, 0xFE384003
```

To disable the Slave interface, which effectively blocks all accesses to the board's on-board RAM from the VMEbus, either reset the board or write **07** to the Extended Mode register at **0xFE38 4003** using the following assembler commands:

```
moveb          #0x07, 0xFE384003
```


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


Unlike the other initialization command described in previous paragraphs, writes to the Mode registers described above require correct address **AND** data expressions.

Multi-port memory access contention

Because the board's memory is triple-ported, contention can occur when simultaneous access is made to it by more than one Master. This Local bus contention is discussed in the *Dynamic RAM* chapter in the section entitled *Resolving lock contention*.

Self references

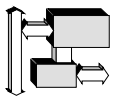
If a board attempts a VMEbus access to its own VME Slave address, the access is ignored by its Slave logic. In most configurations, there will be no other VMEbus Slave responding to that address and the VMEbus time-out counter will fire and cause a VME bus error. Should the card be operating with no VMEbus or without a VME System Controller, any VMEbus or "self references" will cause the card to hang, unless the on-board system controller has been jumpered to provide bus timeouts.

In some dual V452 Series board systems where it may be desirable to load identical software onto each board, it is useful for both V452 Series boards to have the same VMEbus Slave address. In this case, each board is able to access the other's memory by ignoring self-references and by allowing the other card to respond.

VME addresses

The standard address map used by other VMEbus Masters to access the V452 Series board's on-board DRAM is listed in the *Address map* chapter in Section 3 or on the ***V452 Series Quick Reference Card***.

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



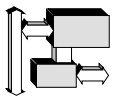
VME Slave block transfer (BLT)

Normal VME transfers move **one** 32-bit word per address strobe. Due to arbitration and contention overhead, such transfers are typically limited to about 8 MB/sec for multi-ported memory. VME block transfers (BLTs) on the other hand, transfer up to **256** 32-bit words per address strobe. Because arbitration and contention overhead are amortized over many words, BLTs can run in excess of 30 MB/sec even while executing from triple-ported memory.

The V452 Series supports BLT32 Slave transfers at 30+ MB/sec and BLT64 Slave transfers at 60+ MB/sec. The VME Master determines whether or not a data transfer uses the block transfer technique. As a VME Slave with the BLT option the V452 Series simply responds to the BLT Master. No software or jumpers are required to implement this feature. When combined with data broadcasting, the BLT feature provides amazing data rates. For example, a single VME Master writing to 8 CPU boards can transfer data at an effective rate of over **240 MB/sec.**

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

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



Data broadcasting

Data broadcasting is an optional feature of V452 Series boards that allows any board in an established broadcasting *group* to write the same data to all other boards in the same group as a single bus transfer event. This feature is offered on Synergy CPU boards and may not be available on products from other manufacturers.

Data broadcasts are performed by writing to a special *group address* that causes every board within the broadcast group to be written to. However, unlike a normal VME transfer, the data broadcasting circuitry does not respond with a VME Acknowledge signal until all boards in the group have finished receiving the write.

This effect is possible because each V452 Series board in the group is tied together via a common open-collector *group wire*. When each board completes its write after receiving the broadcast, it acknowledges completion of the write by de-asserting the group wire. When all boards in the group have received and written the broadcast transfer DTAck\ is asserted and the broadcast cycle is complete.



The data broadcast feature does not interfere with a board's ability to respond to its individual VMEbus Slave address. Rather, it allows boards to respond to two addresses; their individual VME Slave address **and** the group address as well.

Setting up V452 Series boards to support data broadcasting requires the following three configuration steps:

1. Setting up the V452 Series to support data broadcasting by installing a small *group wire* on the solder side of the board,

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

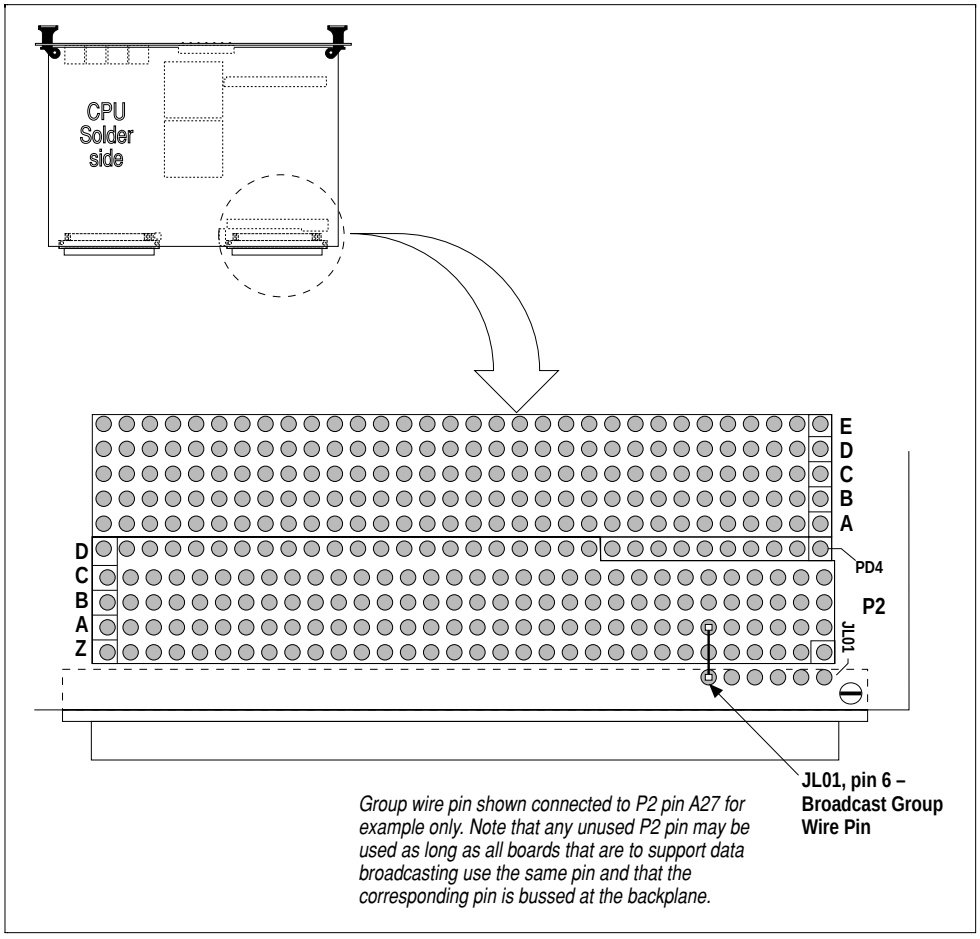
❷ Setting up broadcasting groups by configuring the VME Slave access window location for all boards in a group,

❸ Enabling the data broadcasting function on each board by writing to an on-board register.

The paragraphs below describe each of these steps in greater detail.

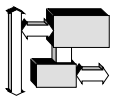
Hardware setup for data broadcasting

All V452 Series boards that are to support data broadcasting require the installation of a *group wire* on the solder side of the board. Install the group wire by connecting JL01 pin 6 to an unused P2 pin using 30 gauge insulated wire. See the diagram below. Attach the group wire to this same pin for all other boards in the group. Bus all these pins to each other on the backplane with wire-wrap.



Broadcasting group wire installation

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





Use a unique group wire location for each additional data broadcasting group in the system.

Setting up data broadcasting groups

The Data Broadcast circuit uses the board's VME Slave window location to set up data broadcasting. As described in the previous chapter, the location of the Slave access window is configured by writing to a series of on-board registers. The data broadcasting circuitry uses this same information to assign the board to a broadcasting group. The paragraphs below describe this process in greater detail.

Assigning boards to a broadcast group

Broadcast groups consist of a series of **256 MB** boundaries. within the larger 4 GB VMEbus addressing range. The most significant four bits (A31-A28) of the base address for a board's VME Slave access window determines to which broadcast group it belongs. Each board that you want to have in the same broadcast group must have its VME Slave access window set to be within the same 256 MB boundary.

In theory this scheme could allow the 4 GB addressing range of the VMEbus to support up to 16 different 256 MB broadcast groups (where the range for **Group 1** = 0000 0000–0FFF FFFF, **Group 2** = 1000 0000–1FFF FFFF and so on). In practice, however, the number of possible broadcast groups is more limited because local DRAM and other devices on each board occupy a share of the total address space that can accessed by each board's VME Master interface.

The tables below lists the possible broadcast groups that can be setup for V452 Series boards.

Available address ranges for broadcast groups

Address/ data width	Broadcast Group	Address range
A32/D32	1	1000 0000 – 1FFF FFFF
	2	2000 0000 – 2FFF FFFF
	3	3000 0000 – 3FFF FFFF
	4	8000 0000 – 8FFF FFFF
	5	9000 0000 – 9FFF FFFF
	6	A000 0000 – AFFF FFFF
	7	B000 0000 – BFFF FFFF

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

Available address ranges for broadcast groups (continued)

A32/D16	1	4000 0000	4FFF FFFF
	2	5000 0000	5FFF FFFF
	3	6000 0000	6FFF FFFF
	4	7000 0000	7FFF FFFF

As shown in the tables, if using A32/D32 addressing, up to 7 different ranges are available for broadcast groups. If using A32/D16 addressing, up to four ranges can be used for data broadcast groups.

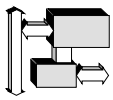
Arranging boards within a broadcast group

Each of the 256 MB group ranges contain the *group window* that is used by every board in the group as well as all of the unique VME Slave windows for the boards in the group. The proper arrangement of the group and Slave access windows is effected by the following factors:

- **Location of the group window** — The data broadcasting circuitry automatically positions the group window at the **bottom** of the addressing range for the respective broadcasting group.
- **Size of the group window** — The size of the group window is determined by the amount of local DRAM on the boards in the group. If all of the boards in the group contain 16 MB (or less) of local DRAM, the group window occupies the **bottom 16 MB** of the group's address range. If one or more of the boards in the group contains 32 MB of local DRAM, the group window is the **bottom 32 MB** of the group range. The data broadcasting circuitry automatically "detects" the maximum amount of local memory on all board in the group and sets the group window accordingly. The **minimum size** for the group window is 16 MB.
- **Location of each VME Slave window** — The acceptable location for the Slave access window for each board in the group must be consistent with the size of the group window. If the group window is 16MB (because all of the boards in the group contain 16 MB or less of local RAM), then the VME Slave access window for each board in the group must on a **16 MB boundary** within the group range (not including the bottom 16 MB where the group window resides) even if they have less than 16 MB of on-board DRAM. Thus, a "16 MB" broadcast group can have up to **15 member boards**.

  If the group window is 32 MB (because one or more boards in the group contain 32 MB of local RAM), then the VME Slave access window for each board in the group must set up on a **32**

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



MB boundary within the group range (not including the bottom 32 MB where the group window resides) even if they have less than 32 MB of on-board DRAM. Thus, a "32 MB" broadcast group can have up to **7 member boards**.

- **Groups with mixed local memory sizes** — As was stated above, the logic for the data broadcasting hardware automatically sets the size of the group window to either 16 MB or 32 MB if any of the boards in the group contain 32 MB of local memory. This feature is designed to make data broadcasting easier to use. However, this characteristic has important implications if you intend to use boards with mixed memory sizes in the same group. It is important to remember in this case that a board with 4 MB of local memory will only be able to "see" 4 MB of the group window despite its real size. Because all transfers sent via the group window are sent to all boards in the group, these transfers should be made usable by the board with the lowest amount of DRAM in the group.

Each of the 256 MB group ranges contain the *group window* that is used by every board in the group as well as all of the unique VME Slave windows for the boards in the group. The proper arrangement of the group and Slave access windows is affected by the following factors:

Address bits A27-A24) of the base address for a board's VME Slave access window assigns the location of the Slave window within a broadcast group. The tables below list the acceptable Slave window locations within 16 MB and 32 MB broadcast groups.

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

VME Slave window ranges for 16 and 32 MB broadcast groups

Group window size	Slave window	Address range
16 MB (all boards in group contain 16 MB or less of local memory)	1	x100 0000 – x1FF FFFF
	2	x200 0000 – x2FF FFFF
	3	x300 0000 – x3FF FFFF
	4	x400 0000 – x4FF FFFF
	5	x500 0000 – x5FF FFFF
	6	x600 0000 – x6FF FFFF
	7	x700 0000 – x7FF FFFF
	8	x800 0000 – x8FF FFFF
	9	x900 0000 – x9FF FFFF
	10	xA00 0000 – xAFF FFFF
	11	xB00 0000 – xBFF FFFF
	12	xC00 0000 – xCFF FFFF
	13	xD00 0000 – xDFF FFFF
	14	xE00 0000 – xEFF FFFF
	15	xF00 0000 – xFFF FFFF
32 MB (one or more boards in group contain 32 MB of local memory)	1	x200 0000 – x2FF FFFF
	2	x400 0000 – x4FF FFFF
	3	x600 0000 – x6FF FFFF
	4	x800 0000 – x8FF FFFF
	5	xA00 0000 – xAFF FFFF
	6	xC00 0000 – xCFF FFFF
	7	xE00 0000 – xEFF FFFF

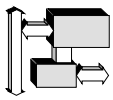
Enabling/disabling data broadcasting

The last configuration step that is required to allow a board to support data broadcasting is to enable the data broadcasting function by writing to the Primary Control register. In the default condition, data broadcasting is **disabled**.

To enable data broadcasting, execute the following 680x0 assembler command:

```
moveb          #0x00, 0xFE38C009
```


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



To disable data broadcasting, either reset the board or execute the following 680x0 assembler command:

```
moveb        #0x00, 0xFE38C008
```

Configuring a sample data broadcast group

The paragraphs in this section are a procedure that describe the required configuration to place three boards into a **sample** broadcast group. These three boards have the following memory characteristics:

Board #1 32 MB local DRAM
Board #2 16 MB local DRAM
Board #3 16 MB local DRAM

The remaining paragraphs in this chapter describe the require configuration to establishing a sample broadcasting group consisting of the three board listed above.

❶ **Choose the broadcasting group address range** — Presuming that all three boards already have a group wire installed, the first step in the configuration process is to choose the address range for the broadcast group.

Presuming further that A32/D32 VME addressing is desired, suppose that the Group 2 range is selected which is highlighted in the table on the next page.

Address range for SAMPLE broadcast group

Address/data width	Broadcast Group	Address range
A32/D32	1	1000 0000 – 1FFF FFFF
	2	2000 0000 – 2FFF FFFF
	3	3000 0000 – 3FFF FFFF
	4	8000 0000 – 8FFF FFFF
	5	9000 0000 – 9FFF FFFF
	6	A000 0000 – AFFF FFFF
	7	B000 0000 – BFFF FFFF

❷ **Choose the Slave window arrangement** — Because one of the boards in this group has 32 MB of on-board memory, the sample broadcast group must have the following characteristics:

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

- 32 MB group window (set automatically by hardware)
- VME Slave windows for all boards in the group must be set on a 32 MB boundary.

The table below shows three base address locations (highlighted) that could be used for the three VME Slave windows in the sample group.

VME Slave window ranges for the SAMPLE broadcast group

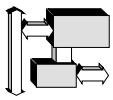
Group window size	Slave window	Address range
32 MB (one or more boards in group contain 32 MB of local memory)	**1 (Board #1)**	**x200 0000 – x2FF FFFF**
	2 (Board #2)	**x400 0000 – x4FF FFFF**
	3 (Board #3)	**x600 0000 – x6FF FFFF**
	4	**x800 0000 – x8FF FFFF**
	5	**xA00 0000 – xAFF FFFF**
	6	**xC00 0000 – xCFF FFFF**
	7	**xE00 0000 – xEFF FFFF**

❸ **Configure Board #1** — To implement the scheme selected in the first two steps, the base address of the VME Slave window for Board #1 needs to be set to **2200 0000** which would require the register configuration shown below.

Slave base address configuration — BOARD #1 (SAMPLE)

Write in 3 most significant hex digits of desired VME Slave base address [1]	Write binary equivalent of each digit [2]	VME address bit	To set this bit to "0" write to:	To set this bit to "1" write to:
Most significant hex digit	0	Bit 31	FE38 800E [4]	FE38 800F
	0	Bit 30	FE3A 000E [4]	FE3A 000F
2	1	Bit 29	FE3A 000C [4]	FE3A 000D
	0	Bit 28	FE3A 000A [4]	FE3A 000B
2nd most significant hex digit	0	Bit 27	FE3A 0008 [4]	FE3A 0009
	0	Bit 26	FE38 800C [4]	FE38 800D
2	1	Bit 25	FE38 800A [4]	FE38 800B
	0	Bit 24 [5]	FE38 8008 [4]	FE38 8009
3rd most significant hex digit	0	Bit 23 [6]	FE38 8006 [4]	FE38 8007
	0	Bit 22 [7]	FE38 8004 [4]	FE38 8005
0	N/A	Bit 21 [8]	N/A	N/A
	N/A	Bit 20 [8]	N/A	N/A

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



The following instructions would set up and enable the VME Slave interface and data broadcasting for Board #1:

```
moveb    #0x00, 0xFE38800E  | Set bit A31 to "0"
moveb    #0x00, 0xFE3A000E  | Set bit A30 to "0"
moveb    #0x00, 0xFE3A000D  | Set bit A29 to "1"
moveb    #0x00, 0xFE3A000A  | Set bit A28 to "0"
moveb    #0x00, 0xFE3A0008  | Set bit A27 to "0"
moveb    #0x00, 0xFE38800C  | Set bit A26 to "0"
moveb    #0x00, 0xFE38800B  | Set bit A25 to "1"
moveb    #0x00, 0xFE388008  | Set bit A24 to "0"
moveb    #0x00, 0xFE388006  | Set bit A23 to "0"
moveb    #0x00, 0xFE388004  | Set bit A22 to "0"
moveb    #0x0F, 0xFE384003  | Enable VME Slave interface
moveb    #0x00, 0xFE38C009  | Enable data broadcasting
```

❹ **Configure Board #2** — The base address of the VME Slave window for Board #1 needs to be set to **2400 0000** which would require the register configuration shown below:

Slave base address configuration — BOARD #2 (SAMPLE)

Write in 3 most significant hex digits of desired VME Slave base address [1]	Write binary equivalent of each digit [2]	VME address bit	To set this bit to "0" write to:	To set this bit to "1" write to:
Most significant hex digit	0	Bit 31	FE38 800E [4]	FE38 800F
	0	Bit 30	FE3A 000E [4]	FE3A 000F
2	1	Bit 29	FE3A 000C [4]	FE3A 000D
	0	Bit 28	FE3A 000A [4]	FE3A 000B
2nd most significant hex digit	0	Bit 27	FE3A 0008 [4]	FE3A 0009
	1	Bit 26	FE38 800C [4]	FE38 800D
4	0	Bit 25	FE38 800A [4]	FE38 800B
	0	Bit 24 [5]	FE38 8008 [4]	FE38 8009
3rd most significant hex digit	0	Bit 23 [6]	FE38 8006 [4]	FE38 8007
	0	Bit 22 [7]	FE38 8004 [4]	FE38 8005
0	N/A	Bit 21 [8]	N/A	N/A
	N/A	Bit 20 [8]	N/A	N/A

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

The following instructions would set up and enable the VME Slave interface and data broadcasting for Board #2:

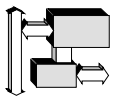
```
moveb    #0x00, 0xFE38800E  | Set bit A31 to "0"
moveb    #0x00, 0xFE3A000E  | Set bit A30 to "0"
moveb    #0x00, 0xFE3A000D  | Set bit A29 to "1"
moveb    #0x00, 0xFE3A000A  | Set bit A28 to "0"
moveb    #0x00, 0xFE3A0008  | Set bit A27 to "0"
moveb    #0x00, 0xFE38800D  | Set bit A26 to "0"
moveb    #0x00, 0xFE38800A  | Set bit A25 to "1"
moveb    #0x00, 0xFE388008  | Set bit A24 to "0"
moveb    #0x00, 0xFE388006  | Set bit A23 to "0"
moveb    #0x00, 0xFE388004  | Set bit A22 to "0"
moveb    #0x0F, 0xFE384003  | Enable VME Slave interface
moveb    #0x00, 0xFE38C009  | Enable data broadcasting
```

❺ **Configure Board #3** — Even though it has only 16 MB of on-board DRAM, Board #3 would also need to be set to a 32 MB boundary, in this case **2600 0000**, which would require the register configuration shown below:

Slave base address configuration — BOARD #3 (SAMPLE)

Write in 3 most significant hex digits of desired VME Slave base address [1]	Write binary equivalent of each digit [2]	VME address bit	To set this bit to "0" write to:	To set this bit to "1" write to:
Most significant hex digit	0	Bit 31	FE38 800E [4]	FE38 800F
	0	Bit 30	FE3A 000E [4]	FE3A 000F
2	1	Bit 29	FE3A 000C [4]	FE3A 000D
	0	Bit 28	FE3A 000A [4]	FE3A 000B
2nd most significant hex digit	0	Bit 27	FE3A 0008 [4]	FE3A 0009
	1	Bit 26	FE38 800C [4]	FE38 800D
6	1	Bit 25	FE38 800A [4]	FE38 800B
	0	Bit 24 [5]	FE38 8008 [4]	FE38 8009
3rd most significant hex digit	0	Bit 23 [6]	FE38 8006 [4]	FE38 8007
	0	Bit 22 [7]	FE38 8004 [4]	FE38 8005
0	N/A	Bit 21 [8]	N/A	N/A
	N/A	Bit 20 [8]	N/A	N/A

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



The following instructions would set up and enable the VME Slave interface and data broadcasting for Board #3:

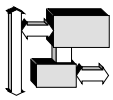
```
moveb    #0x00, 0xFE38800E   | Set bit A31 to "0"
moveb    #0x00, 0xFE3A000E   | Set bit A30 to "0"
moveb    #0x00, 0xFE3A000D   | Set bit A29 to "1"
moveb    #0x00, 0xFE3A000A   | Set bit A28 to "0"
moveb    #0x00, 0xFE3A0008   | Set bit A27 to "0"
moveb    #0x00, 0xFE38800D   | Set bit A26 to "1"
moveb    #0x00, 0xFE38800B   | Set bit A25 to "1"
moveb    #0x00, 0xFE388008   | Set bit A24 to "0"
moveb    #0x00, 0xFE388006   | Set bit A23 to "0"
moveb    #0x00, 0xFE388004   | Set bit A22 to "0"
moveb    #0x0F, 0xFE384003   | Enable VME Slave interface
moveb    #0x00, 0xFE38C009   | Enable data broadcasting
```



The instructions shown on the previous sample code listing include lines to configure register locations to hold a binary "0" value. These instructions are required ***only*** if the state of these bits have been previously changed from the default "0" condition. For the sake of modularity, however, it may still be advisable to include an instruction for each address bit as shown in these samples.

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

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



VME Master interface

The V452 Series provides a full VME Master interface that supports A16, A24, and A32 addressing, and D8, D16, or D32 data widths.



Whenever an external VMEbus Master writes to a portion of the V452 Series memory that is being cached by the on-board CPU, a cache-to-memory incoherency condition is possible that could result in a loss of data. For a discussion of these cache coherency considerations and a summary of some useful cache management techniques, refer to the applicable *CPU* chapter in Section 4.

Setting up the Master interface

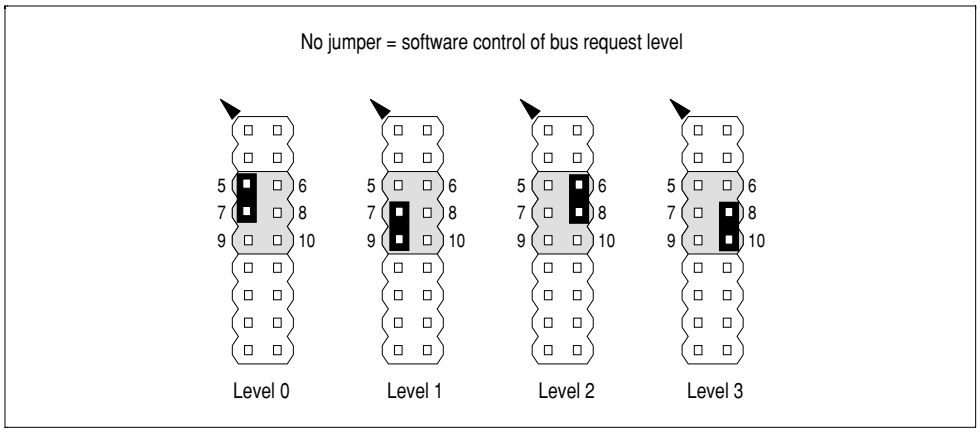
When multiple CPU boards want control of the VMEbus, a detailed set of arbitration rules and conditions dictate how bus mastership is determined. V452 Series boards provide several configurable options that allow you to organize how the V452 Series board fits into this larger VMEbus arbitration scheme.

Setting the VMEbus request level

The VME interface on V452 Series boards can request access to the VMEbus at four different priority levels from **lowest** (level 0) to **highest** (level 3) priority. This request level operates in conjunction with the request/release methods (ROR, RWD or FAIR) to pinpoint the specific bus arbitration position of a specific board.

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

To set the bus request level, configure pins **5-10** on the **JK17** jumper as shown in the figure below:



Bus request level jumper settings (JK17)

Note If none of the shunt configurations shown above is in place, the Ethernet/VMEbus control register takes over to set the bus request level (default = Level 3).

Primary Control register

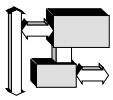
The remaining bus control functions are set via the Primary Control register. The location pertaining to the Master interface configuration are listed in the table below:

Primary Control register — Master interface functions

Address	Function	Description
FE38 C000	ROR release	Select Release on Request bus request handling. (***default)***
FE38 C001	RWD release	Select Release When Done bus request handling.
FE38 C002	non-FAIR requests	Selects non-FAIR (normal) bus release handling. (***default)***
FE38 C003	FAIR requests	Select FAIR bus release handling.

Note The V452 Series Control register circuitry decodes only the address lines of all write accesses. As a result, the data that is actually written into the register **DOES NOT** matter. A write access using any data is all that is required.

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



FAIR vs non-FAIR bus requests

V452 Series boards support FAIR and non-FAIR bus requester as described below:

- **FAIR bus request** — gives all Masters on the VMEbus an equal share of the overall bus bandwidth.

  If enabled, the FAIR requester circuitry on each participating Master does not allow its own Master to request control of the data transfer bus until all other Masters on the same VMEbus request level (as described in the paragraphs above) have had control of the bus. In other words, the FAIR requester does not issue a bus request until no other Master is requesting control of the bus.

  For FAIR bus requests to work, **ALL DEVICES on the same bus request level must use it**. The net effect of this scheme is to force Masters that frequently need/request the bus or are closer to the System Controller's arbiter in the daisy-chain to allow equal bus access by other Masters that may request the bus less often or are further away from the arbiter in the daisy-chain.

  Enabling the FAIR requester is independent of the priority or round-robin request scheme set by the System Controller. The FAIR request scheme applies only to Masters on the same VME request level. As a result, it is possible to configure a system where the Masters on one level use FAIR requests while Masters on other levels use either the priority or round-robin schemes from the System Controller to manage bus requests.

  To implement FAIR requests across the entire VMEbus system, it is best to configure all Masters to use FAIR bus requests on the same request level.

  Boards that observe the FAIR request scheme can be used with boards not supporting it, if the boards that do not support FAIR requests are programmed to use only a small percentage of the VMEbus bandwidth. Non-FAIR request boards can also be assigned a priority level lower than the level used by the FAIR request boards.

- **Non-FAIR bus request** (*default*) — with the FAIR request option OFF Masters compete for the bus on either a straight **priority** or **round-robin** basis as determined by the System Controller. For more information about the meaning of these terms, see the ***System Controller*** chapter in this section.

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

In the default condition, the FAIR bus request scheme is **not active**. To direct the V452 Series board to use FAIR bus requests, execute the following 680x0 assembler command:

```
moveb          #0x00, 0xFE38C003
```

To return of non-FAIR bus requests, either reset the board or execute the following 680x0 assembler command:

```
moveb          #0x00, 0xFE38C002
```

ROR vs. RWD bus release

When the board attempts to access VME address space, a request for control of the VMEbus is made to the VME Arbiter. The VMEbus requester on V452 Series boards can be configured to use on of the following two bus release methods:

- **Release On Request** (ROR) ***default*** — requests and holds the VMEbus until it senses another VMEbus Master making a bus request. The ROR bus requester minimizes VMEbus arbitration overhead in a system transferring bursts of data.
- **Release When Done** (RWD) — requests and then releases the VMEbus immediately after completing the VME transfer.

The desired request method is set by writing to the Extended Control register at 0xFE38 0000 or 0xFE38 0001.

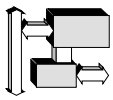
In the default condition, Release on Request (ROR) is selected. To select Release When Done (RWD), execute the following 680x0 assembler command:

```
moveb          #0x00, 0xFE38C001
```

To return to ROR requests, either reset the board or execute the following 680x0 assembler command:

```
moveb          #0x00, 0xFE38C000
```


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



VME Master data transfer bandwidth

After the V452 Series board acquires the VMEbus, a V452 Series making standard (non-BLT) VME Master accesses has a data transfer bandwidth that can be calculated from the following formula:

$$\text{Xfer in MB/sec} = \frac{\text{Bytes-per-xfer} * 1000}{55 + X + (4 * C)}$$

where:

- **Bytes-per-Xfer =** Transfer data width (e.g., 4 for a 32-bit transfer)
- **X** = Slave access time in nanoseconds
- **C** = CPU clock cycle time in ns (e.g., 40 for a 25 MHz CPU)

A 25 MHz V452 Series accessing a Slave with an access time (Address Strobe to Ack Time) of 100 ns, has a cycle time of 315 ns, for a bandwidth of approximately 12 MB/sec, assuming 32-bit transfers.



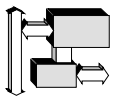
The initial arbitration time for the VMEbus is ignored in the above equation.

If the CPU (rather than a DMA device) is doing a VME transfer, the overhead of accessing the local memory must be added to the denominator of the above equation. On average, it takes 1.5 clock cycles to fetch a word from memory and 1.4 clock cycles to write a word to memory. Of course, there is also instruction fetch time to consider too, but you can normally arrange to have the instructions for a block copy routine in CPU cache. So for the 25 MHz example above we have to add 1.5 * 40 = 60 ns to the 315 ns, for a cycle time of 375 or a sustained bandwidth of about 10.4 MB/sec transferring data from on-board memory to VME.

The 10.4 MB/sec value represents the bandwidth for random access to the VMEbus. Blocks of data can be transferred at over 60 MB/sec with DMA-assisted BLT as described in the next section.

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

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



VME Master BLT

V452 Series boards support 32-bit Master BLT with DMA (BLT32) that provides the following features:

- 32-bit DMA
- up to 40 MB/sec transfer rate (30 MB/sec typical)
- Transfer lengths from 8 to 1024 bytes.

V452 Series boards also support 64-bit Master BLT with DMA (BLT64) that provides the following features:

- 64-bit DMA
- up to 80 MB/sec transfer rate (60 MB/sec typical)
- Transfer lengths from 16 to 2048 bytes.

Performing DMA block transfers increases performance over normal data transfers for data lengths greater than 20 bytes for BLT32 transfers or 40 bytes for BLT64 transfers.



A complete C-language example for both **BLT32** and **BLT64** block transfers can be found in the *Block Transfer (BLT) with DMA example* chapter in the ***Code Examples*** section.

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

Block transfer strategy

The V452 Series BLT DMA is a simple DMA finite state machine designed for speed and flexibility. The following design characteristics of the V452 Series BLT engine effects the design of the software for performing BLT transfers.

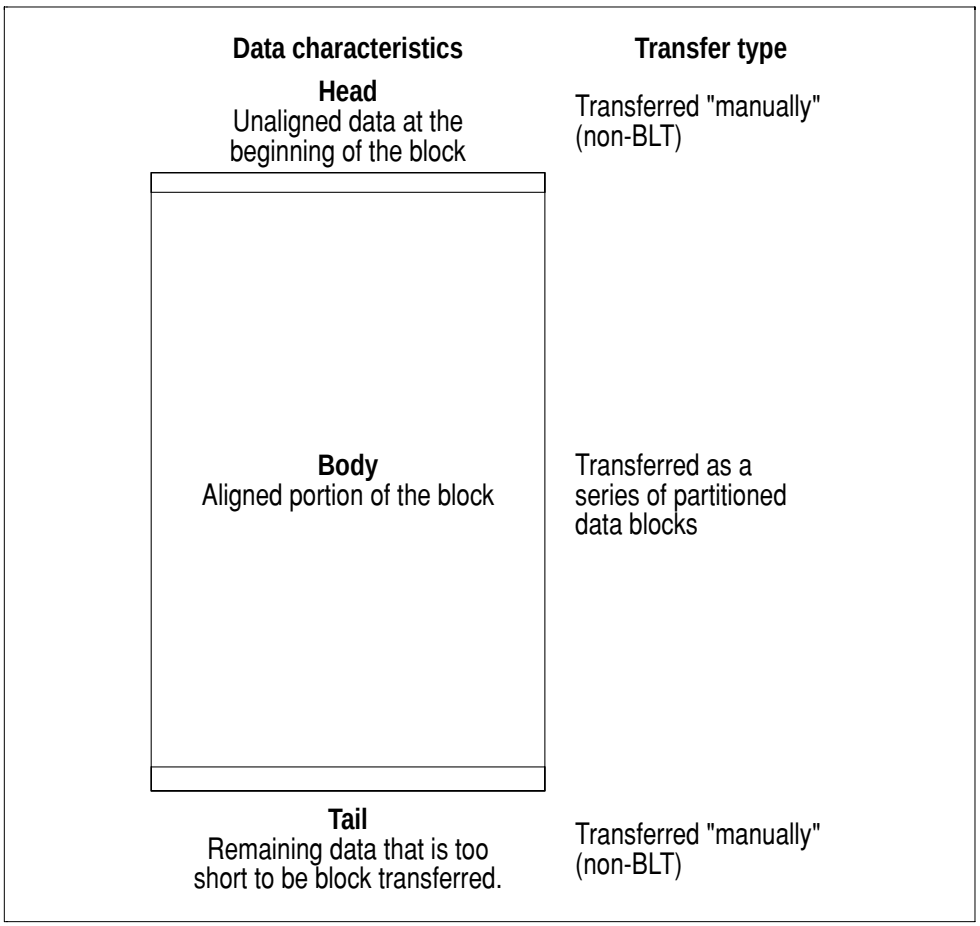
- Data blocks for BLT32 transfers must be at least **8 bytes in length** and must be aligned on **4-byte boundaries**. Data blocks for BLT64 transfers must be at least **16 bytes in length** and must be aligned on **8-byte boundaries**.
- Data blocks for both BLT32 and BLT64 transfers must be **partitioned** to observe **page boundaries** on both the target and source system.
- Because BLT64 transfers **twice as much data** as BLT32, the byte counter in the BLT software module should divide the requested transfer count **by two** before performing a BLT64 transfer. Doing so allows calling routines to use the **same unit** for the transfer count (i.e., a byte) for both BLT32 and BLT64 transfers.

These characteristics suggest the adoption of a particular strategy for implementing BLT transfers on V452 Series boards as described in the table and figure below:

Data block characteristics

Data Characteristics	Transfer type
Head	**Manual (non-BLT)** — If the first few bytes of a block to be transferred are not located on the 4 byte (BLT32) or 8-byte(BLT64) boundary the unaligned data must be transferred using a non-BLT transfer. The BLT software must look at the head portion of the data block to be transfer–red, identify the first portion of the block that is properly aligned and transfer the unaligned portion manually.
Body	**BLT Loop** — The main portion of the data block is transferred using a BLT loop. However, the BLT software must ensure that the transferred blocks do not cross any page boundaries either on the source or the target. This block partitioning can be accomplished by finding the amount of data area to the *next* page boundary on both the source and target and then partitioning the next block transfer to be equal to the size of the *lesser* of these two areas. Applying this algorithm to each cycle of the BLT transfer loop ensures that data is never read from the source or written to the target across a page boundary.
Tail	**Manual (non-BLT)** — If the BLT loop process fails to transfer all of the data because the tail portion of the data block contains less than the minimum number of bytes (i.e., 8 for BLT32 or 16 for BLT64), then the BLT software must transfer these remaining bytes of data manually.

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



BLT transfer strategy

Using the MOVES instruction

V452 Series BLT transfers use the CPU's **MOVES** instruction to activate the BLT DMA engine. As a result, the CPU must be operating in Supervisor mode to perform BLTs to/from either Supervisor or User memory space.

> ⚠ Due to problems within the '040/'060 chip itself, the MOVES instruction cannot be used to access virtual addresses that are mapped by the Paged MMU. For some reason, the MMU is unable to translate MOVES accesses appropriately. As a result, all MOVES accesses must be made to the actual physical address whenever the MMU is turned on.

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

Before MOVES can be used, the CPU's internal Source and Destination function code registers (SFC and DFC) also must be set up to indicate whether a Supervisor or User BLT is desired. The sequence shown below sets up BLTs to/from **Supervisor** space by writing a **4** to the SFC and DFC registers:

```
moveq    #0x04, d0      | 04 value sets up supervisor BLT
movc     d0, sfc        | write d0 to Source FC  register
movc     d0, sdc        | write d0 to Destination FC  register
```

To set up BLTs for transfers to/from **User** space, replace the first line of the previous instruction sequence with the following instruction which writes a **0** to the SFC and DFC registers:

```
moveq    #0x0, d0       | 0 value sets up user BLT
```



The SFC and DFC registers retain the entered value until changed or reset. The SFC and DFC set-up sequence given above is only required for the initial setup or to change the type of transfer desired (i.e., to/from Supervisor vs. User memory space). Take care not to allow an interrupt service routine to change the value of the SFC and DFC registers during a BLT transfer.

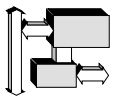
Using the **MOVES** instruction eliminates any possible interference between the BLT transfer and the operation of the CPU's internal caches. This allows the caches to remain operational during the BLT. Keeping the caches enabled during BLT transfers is desirable, because the CPU can continue to execute code from the cache during transfers (even if it's only activity is processing the loop of BLT transfers).

Because the BLT DMA engine re-enables interrupts after each individual transfer, the CPU can also service interrupts between BLT transfers, even when the loop or in-line routine is executing nothing but continuous BLT transfers.

DMA block transfer cycle

Given the transfer strategy outlined above, the required steps in a typical DMA block transfer can be depicted in the figure below and described in the paragraphs to follow:

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



❶ Check that the target and source are correspondingly aligned. (For example, if the first byte for the source of the transfer is a byte 3, the first byte for target must be either and byte 3 or 7.)

❷ Use the MOVEB instruction to transfer odd bytes at the head of the data block (if necessary) to achieve proper 4-byte (BLT32) or 8-byte (BLT64) alignment for block transfers.

❸ Determine the data area to the next page boundary on both the target and source. Partition the data for block transfer to be equal the size of the lesser of these two areas to avoid overrunning a page boundary on either the source or destination.

❹ Perform the block transfer for the next partitioned data block. Check for and report a transfer error (if desired).

> Note The specific instructions required to do this step are described in greater detail below.

❺ Adjust transfer counters and addresses to reflect the number of bytes transferred.

❻ If there are enough bytes in the data block to continue block transfers (i.e., at least 8 for BLT 32 or 16 for BLT64), return to Step ❸ otherwise go to next step.

❼ If any bytes remain in the data block that are too few to be block transferred, use the MOVEB instruction to transfer these trailing bytes.

BLT Reads from VME

The paragraphs below describe in detail the specific instruction required to perform an individual **Read** block transfer of a properly partitioned block of data.

> ⚠ This sample code presumes that the CPU's MMU is OFF during the BLT transfer. Contact Synergy customer service for application notes describing how to perform BLT transfers while the MMU is ON.

All of individual task and instructions described below are required to accomplish Step ❹ of the block transfer cycle described above.

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

❹a **Set the count and transfer width (and inhibit interrupts)** — The count in **bytes** (BLT32) or **words** (BLT64) is set by writing to the count register at:

- **0xFE60 0000** for BLT32,
- **0xFE60 0020** for BLT64.

Writing to the counter register also inhibits interrupts which is required because the three accesses that set up the DMA engine must be done without interruption.

Because the BLT is done in longwords (BLT32) or double longwords (BLT64), the count is truncated into even **multiples** of either **4** (BLT32) or **8** (BLT64). The **minimum count** is either **8** (BLT32) or **16** (BLT64).

The VME spec states that **BLT32** should not cross any **256 byte** (0x100) boundaries and that **BLT64** should not cross any **2048 byte** (0x400) boundaries.

However, the BLT logic on V452 Series boards is capable of individual BLT32 transfers up to 1024 bytes (0x400) and individual BLT64 transfers up to 2048 bytes (0xFF0) if desired. In either case, the transfer should not cross any 2048 byte boundary.

The count must appear in bits 9-0 of the register which should be accessed as a longword. Bits 0 and 1 of the count register are ignored. Bits 10-31 are reserved and should be set to zero.

Setting the count arms the DMA logic. The next MOVES instructions must be a read from VME and a write to local RAM.

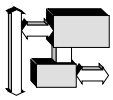
❹b **Execute a miscellaneous instruction** — Executing a "do nothing" instruction here is required to ensure that any impending interrupt is handled before the next step. The MOVE instruction shown in the instruction summaries on the next page provide the necessary pause but execute slightly faster than using the customary NOP instruction.

❹c **Set the VME beginning address and direction** — The VME beginning address and direction is set by using the **MOVES** instruction to read a longword from the first location of the VME region to be transferred.

❹d **Set the local RAM beginning address** — The local RAM beginning address is set by using the **MOVES** instruction to write the longword read in step ❹c from the VME to the first location of the local RAM region to be transferred.

Once the above setup has been done, the DMA engine performs the block transfer. The execution of the transfer is invisible to the program. After the block transfer is complete, interrupts are automatically re-enabled.

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



❹e **Check for successful completion (optional)** — If an error occurs in the individual BLT transfer, the V452 Series reports the error in Data bit 4 of Status register (0xFE380002). A **1** in this position indicates the BLT transfer was successful. A **0** indicates the BLT DMA encountered a Bus Error.



If it encounters a Bus Error during a BLT transfer, the DMA engine immediately aborts the transfer.

BLT32 Read instruction summary — the following 680x0 code instructions are required to accomplish a BLT32 read:

```
MOVE.L Count to 0xFE600000     ; set BLT Start register                    (❹a)
MOVE.L d0,d0                   ; pause for a possible interrupt to be serviced (❹b)
MOVES.L VME-Adr to register    ; select VME address                        (❹c)
MOVES.L register to RAM-Adr    ; Setting the local RAM address starts xfer (❹d)
BTST bit 4 of  0xFE380002      ; test BLT Error bit                        (❹e)
BEQ Error-Routine              ; handle error if needed                    (❹e)
```

BLT64 Read instruction summary — the following 680x0 code instructions are required to accomplish a BLT64 read:

```
MOVE.L Count to 0xFE600020     ; set BLT Start register                    (❹a)
MOVE.L d0,d0                   ; pause for a possible interrupt to be serviced (❹b)
MOVES.L VME-Adr to register    ; select VME address                        (❹c)
MOVES.L register to RAM-Adr    ; Setting the local RAM address start s xfer (❹d)
BTST bit 4 of  0xFE380002      ; test BLT Error bit                        (❹e)
BEQ Error-Routine              ; handle error if needed                    (❹e)
```



To allow a calling routine to specify the count in bytes regardless of whether a 34 or 64-bit transfer is to be used, the following instruction (which shifts the count value given in bytes to words) can be inserted at the beginning of the BLT64 read routine shown above:

lsrl #1, count

BLT Writes to VME

The paragraphs below describe in detail the specific instruction required to perform an individual **Write** block transfer of a properly partitioned block of data.

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

⚠ This sample code presumes that the CPU's MMU is OFF during the BLT transfer. Contact Synergy customer service for application notes describing how to perform BLT transfers while the MMU is ON.

The tasks and instructions described below are required to accomplish Step ❹ of the block transfer cycle described earlier in this chapter.

❹a **Set the count and transfer width (and inhibit interrupts)** — The count in **bytes** (BLT32) or **words** (BLT64) is set by writing to the count register at:

- **0xFE60 0000** for BLT32
- **0xFE60 0020** for BLT64

Writing to the counter register also inhibits interrupts which is required because the three accesses that set up the DMA engine must be done without interruption.

Because the BLT is done in longwords (BLT32) or double longwords (BLT64), the count is truncated into even **multiples** of either **4** (BLT32) or **8** (BLT64). The **minimum count** is either **8** (BLT32) or **16** (BLT64).

The VME spec states that **BLT32** should not cross any **256 byte** (0x100) boundaries and that **BLT64** should not cross any **2048 byte** (0x400) boundaries. However, the BLT logic on V452 Series boards is capable of BLT32 transfers up to 1024 bytes (0x3FC) and BLT64 transfers up to 2048 bytes (0xFF0) if desired. In either case, the transfer should not cross any 2048 byte boundary.

The count must appear in bits 9-0 of the register which should be accessed as a longword. Bits 0 and 1 of the count register are ignored. Bits 10-31 are reserved and should be set to zero.

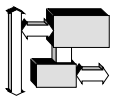
Setting the count arms the DMA logic. The next two MOVES instructions must be a read from VME and a write to local RAM.

❹b **Execute a miscellaneous instruction** — Executing a "do nothing" instruction here is required to ensure that any impending interrupt is handled before the next step. The MOVE instruction shown in the instruction summaries on the next page provide the necessary pause but execute slightly faster than using the customary NOP instruction.

❹c **Read the 1st RAM location** — Reading the first longword of the transfer into a CPU register to prepares for the first VME write.

❹d **Set the VME beginning address and direction** — The VME beginning address and direction is set by using the **MOVES**

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



instruction to write the longword read in Step ❹c to the first location of the VME region to be transferred.

❹e **Execute a NOP** — Executing a instruction here synchronizes 68060's write pipeline.

❹f **Set the Local RAM beginning address** — The local RAM beginning address is set by using the **MOVES** instruction to read a longword from the first location of the local RAM region to be transferred.

Once the above setup has been done, the DMA engine performs the block transfer. The execution of the transfer is invisible to the program. After the block transfer is complete, interrupts are automatically re-enabled.

❹g **Check for successful completion (optional)** — If an error occurs in the individual BLT transfer, the V452 Series reports the error in Data bit 4 of Status register (0xFE380002). A **1** in this position indicates the BLT transfer was successful. A **0** indicates the BLT DMA encountered a Bus Error.



If it encounters a Bus Error during a BLT transfer, the DMA engine immediately aborts the transfer.

BLT32 Write instruction summary — the following 680x0 code instructions are required to accomplish a BLT32 read:

```
MOVE.L Count to 0xFE600000     ; set BLT Start register                    (❹a)
MOVE.L d0,d0                   ; pause for a possible interrupt to be serviced (❹b)
MOVE.L RAM-Adr to register     ; get 1st longword from DRAM                (❹c)
MOVES.L register to VME-Adr ; select VME address                           (❹d)
NOP                                ; synchronize 68060 write pipeline      (❹e)
MOVES.L RAM-Adr to register ; Set the local RAM address starts transfer    (❹f)
BTST bit 4 of  0xFE380002      ; test BLT Error bit                        (❹g)
BEQ Error-Routine              ; handle error if needed                    (❹g)
```

BLT64 Write instruction summary — the following 680x0 code instructions are required to accomplish a BLT64 read:

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

```
MOVE.L Count to 0xFE600020     ; set BLT Start register                          (❹a)
MOVE.L d0,d0                   ; pause for a possible interrupt to be serviced   (❹b)
MOVE.L RAM-Adr to register     ; get 1st longword from DRAM                      (❹c)
MOVES.L register to VME-Ad     ; select VME address                              (❹d)
NOP                              ; synchronize 68060 write pipeline              (❹e)
MOVES.L RAM-Adr to register ; Setting the local RAM address starts transfer      (❹f)
BTST bit 4 of  0xFE380002      ; test BLT Error bit                              (❹g)
BEQ Error-Routine              ; handle error if needed                          (❹g)
```



To allow a calling routine to specify the count in bytes regardless of whether a 34 or 64-bit transfer is to be used, the following instruction (which shifts the count value given in bytes to words) can be inserted at the beginning of the BLT64 write routine shown above:

lsrl #1, count

Operational considerations

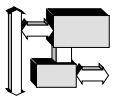
Because block transfers use all available memory bandwidth, the CPU cannot access memory directly until a transfer is done. However, the following features mitigate the effects of any delay:

- DMA BLT transfers occur very quickly. For example, assuming the largest allowed BLT transfer of 256 bytes is transferred at 40 MB per second, memory is tied up for only 6.4 µs.
- During a transfer, the CPU can continue to execute out of its internal caches.

An advantage of tying up the CPU board's data bus during the BLT transfer is that no programming is required to detect completion of the BLT transfer. Even a hundred BLTs in a row execute as fast as the transfer rate allows.

V452 Series include special features using one of the counters (82C54) allowing an EZ-bus Master to operate the V452 Series BLT DMA engine while protecting a percentage of the local bus bandwidth for the V452 Series on-board CPU(s). For more information, about this feature, refer to the ***Timers & counters*** chapter in Section 4.

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



Using Counter 2 (82C54) as a BLT Throttle

While BLT DMA transfers are in progress, the V452 Series boards normally yield the entire local bus bandwidth to service the transfer. For transfers being controlled by an on-board CPU, this approach provides the fastest possible BLT DMA transfer rates without significantly compromising the CPU because it is *busy* managing the transfer and can effectively execute out of its own internal caches.

However, the V452 Series boards can also provide a special feature, using Counter 2 of the 82C54, to allow the on-board CPU(s) to perform local bus operations while an external processor performs BLT DMA transfers at the same time.

Such a concurrent processing arrangement can be achieved using the V452 Series board and an EZ-bus module that contains its own processor or intelligent DMA chip which can be programmed to operate the motherboard's BLT DMA engine.

Without some way to limit or throttle the bandwidth usage of the BLT transfer, however, having the intelligent EZ-bus module manage BLT DMA transfers on the V452 Series board might lock the on-board CPU(s) off its own local bus. Counter 2 of the 82C54 (whose count is available to the intelligent daughter module via pin D7 (CntC\) on the P4 EZ-bus connector) provides a local bandwidth *throttle* as described below.

Whenever a BLT DMA is in progress, circuitry on the V452 Series board gates off or *freezes* Counter 2. As soon as an individual transfer ends, this circuitry frees Counter 2 to start counting. By programming the intelligent EZ-bus module to pause for a certain number of counts before starting the next BLT transfer, Counter 2 can open a variable-length, *bandwidth window* for local bus operations by the on-board CPU(s) while the EZ-bus module performs BLT DMA transfers.

The BLT throttling features of Counter 2 cause it to operate in a non-continuous fashion **whenever** BLT DMA transfers are in use. If you need continuous counting operations lasting more than the duration of a single BLT DMA transfer, consider using another 82C54 counter or one of the 2692 timers.



For more information about using the 82C54 counters, refer to the ***Timers & counters*** chapter in Section 4.

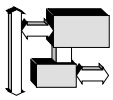
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cache coherency considerations

After completion of a BLT read transfer into the board's memory that is subsequently manipulated by another processor, a mismatch may be present between memory and the CPU's data cache. As a result, it is important to ensure that accesses to the transferred data take place in memory rather than to the CPU's data cache.

For information about other cache management techniques, see the applicable ***CPU*** chapter in Section 4.

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



System Controller

The V452 SBC configures itself automatically as the VMEbus System Controller when it is plugged into Slot 1 of the VME cardcage. The System Controller circuitry has been carefully isolated on the board so that it can continue to function in the unlikely event of a failure elsewhere on the board. The System Controller circuitry provides the following functions:

- **Four-level arbiter** — provides either Priority or Round robin bus arbitration.
- **IAck generation** — receives the wired-or VMEbus IAck signal and drives the daisy-chained IAck-In IAck-Out signals.
- **Bus error timeout** — generates a bus error time-out signal if one or both of the VMEbus data strobes remain asserted for longer than a specific interval that can be set under software control.
- **System reset generator** — generates a VMEbus reset signal upon system power-up, upon a board-level reset, and 100 microseconds after the VME ACFail signal is asserted.
- **System clock generator** — drives the VMEbus SysClk line from an on board 16-MHz oscillator with buffer drivers. (Only one card is allowed to drive the VME SysClk line.)

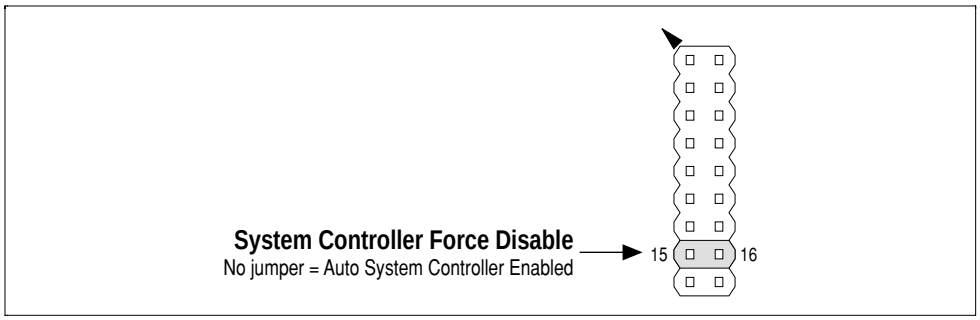


Install the board in the highest priority slot (normally Slot 1) if it is to provide any of the above functions. For more information, see the board installation chapters in ***Getting Started.***

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

Forcing system controller disabled

The system controller can be forced disabled in case other system boards cause interactions with the V452's auto-configuration circuits. To disable the system controller, install a shunt across pins **15** and **16** of jumper **JK17** as shown in the figure below. Remove this shunt to allow operation as system controller.



Jumper settings (JK17) — System Controller disable

Only one bus arbiter can be in use at any one time. When the System Controller circuitry is active, the timeout generator drives the VME BErr\ line on the bus.

Configuring the bus arbiter

One of the System Controller's main tasks is to establish and manage bus arbitration. The V452 has three arbiter configurations:

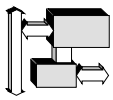
- Four-level **Priority** arbitration — grants control of the VMEbus based on the Master VMEbus request level of 0 to 3 with three being the highest priority and 0 being the lowest.

  In this configuration the arbiter grants the bus to Masters at the higher level. (For more information about VMEbus request levels, refer to the ***VME Master interface*** chapter in this section.)

- Four-way **Round Robin** arbitration — grants equal priority to all bus request levels by sampling each bus level on a rotating basis.

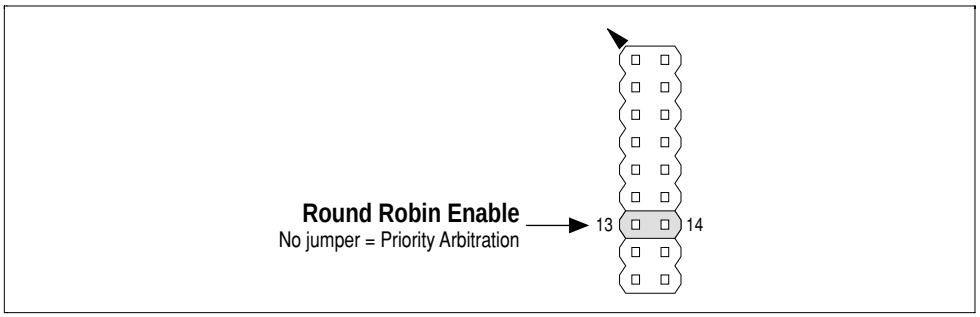
  In the Round Robin configuration the arbiter grants the bus on a rotating basis. When the current Master in control of the bus relinquishes it, the arbiter looks for a request at the next level. When a request is not pending at a certain level, the arbiter skips that level and moves on to the next.

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



- **Single-level** arbitration — can be used by selecting priority arbitration and jumpering all boards to the same bus request level. This is possible with either the Priority or Round Robin modes.

To select **Round Robin** request handling, install a shunt across jumper **JK17** pins **13** and **14** as shown in the figure below. Remove this shunt (or leave off) to select **Priority** handling.



Jumper settings (JK17) — Round Robin vs. Priority requests

VMEbus arbitration timeout

The VMEbus arbitration circuitry includes an automatic arbitration timeout period of 10 µsecs. This feature stops the VMEbus from locking up in the event of delayed bus arbitration cycle.

Configuring the bus error timeout interval

V452 Series boards generate a bus error timeout signal if one or both of the VMEbus data strobes remain asserted for longer than a specified timeout interval. The default timeout interval that is set after a power-cycling or reset is **32 µs**. The slower intervals allow the V452 Series board to accommodate slower systems. The "Never" setting, which allows a bus error to "hang" the bus, is normally only useful in system-level debugging situations.

The table below lists the timeout intervals and the corresponding *Extended Control register* write addresses for each interval.

Extended Control register (FE3A 0000)

VME Access Timeout interval	Extended Control register addresses to Write	
32 µs (default)	FE3A 0000	FE3A 0002
256 µs	FE3A 0001	FE3A 0002
1 ms	FE3A 0000	FE3A 0003
Never (do not assert timeout)	FE3A 0001	FE3A 0003

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

For example, to set the timeout interval to be *256 µs*, execute the following two assembler instructions:

```
moveb       #0x00, 0xFE3A 0001
moveb       #0x00, 0xFE3A 0003
```



The V452 Series circuitry for the *Extended Control register* decodes only the address lines of all write accesses. As a result, the data that is actually written into the register **DOES NOT** matter. A write access using any data is all that is required.

Enabling VME SysFail interrupts

While serving as the System Controller, it may be helpful to interrupt the System Controller board whenever one of the other boards on the bus asserts SysFail. This feature allows the System Controller to be used to restart and/or debug failed boards.

In the default configuration, the SysFail interrupt is **disabled**. Enabling it is a two-step process:

❶ **Enable the Level 7 interrupt group** — V452 Series boards assigns SysFail to the interrupt Level 7 interrupt group that also includes the ACFail and parity error sources.

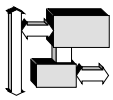
To enable these three sources as a group to interrupt CPU-X, execute the following 680x0 assembler instruction:

```
moveb       #0x00, 0xFE394001
```

To enable these three sources as a group to interrupt CPU-Y (on dual-CPU V452 Series models only), execute the following 680x0 assembler instruction:

```
moveb       #0x00, 0xFE39C001
```


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





The maskable Level 7 interrupt source group described in the paragraphs above is a *steerable* source. On dual-CPU V452 Series model boards steerable interrupt sources can be assigned to either CPU but not to serve as an interrupt source to both CPUs at the same time. For more information on steerable interrupt sources, refer to the ***Interrupts*** chapter in Section 3.

❷ **Enable the SysFail interrupt** — To enable the SysFail interrupt (once the Level 7 interrupt group has been enabled), execute the following 680x0 assembler instruction:

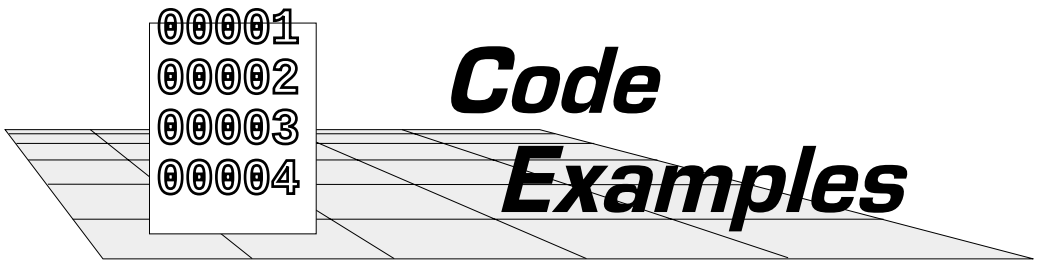
```
moveb          #0x04, 0xFE28003B
```

To disable the SysFail interrupt, reset the board or execute the following 680x0 assembler instruction:

```
moveb          #0x04, 0xFE28003F
```


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

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



Code Examples

6

This section contains coding examples for selected components on V452 Series boards and a summary description of important programming differences between the V452 and other models of Synergy CPU boards.

- Programming differences
- Block transfer (BLT) with DMA example
- Flash EPROM programming tools
- Timer code examples
- 2692 DUART code example

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

00001
00002
00003
00004

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

Programming differences

This chapter briefly lists the programming differences between the V452 SBC and other Synergy model SBCs.

If you have extensive experience with either the V20 or V30 series product, it is suggested that you read this chapter to give you a better idea as to how to program the V452 board.

1. ***8-bit vs 32-bit PROM 1 is slower*** — The V20 and V30 series boards use four EPROMs which provide significantly faster access time than a single EPROM. Expect code in this EPROM to run 3-4 times slower.

2. ***No dynamic bus sizing to EZ-bus boards*** — Any EZ-bus modules that require dynamic bus sizing will not work on the V452 Series boards. Contact the factory for further information.

Any EZ-bus modules that use longword accesses to repeatedly access a byte or word register will require code modification. The ESSE and ESPS are two examples of this type of module. Contact the factory for further information.

3. ***Added status reg bit for X/Y CPU detection*** — The addition of a second CPU requires the code to be able to detect which CPU it is running upon. The status register contains a bit with this information.

4. ***Auto System Controller (V451/V452 only)*** — The V452 SBC configures itself automatically as the system controller when installed in slot #1 of the VME cardcage. A jumper is provided to force the system controller disabled in case other boards in the system interact with the V452 auto-detection.

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

5. ***Software vs. jumper int enables*** — The V452 code must enable any necessary interrupts by writing to the mode registers provided. The V20 and V30 series boards have hardware jumpers that require no software initialization.

The interrupts may be steered to either of the two CPUs on a dual-CPU board. Care must be taken in selecting the proper mode bits to enable for the desired interrupts.

6. ***Software vs jumper VME slave addressing*** — The V452 code must enable the VMEbus slave interface and set its address and mode by writing to the mode registers provided. The V20 and V30 series boards have hardware jumpers that require no software initialization.

7. ***TTR initialization to keep I/O from caching*** — The V452 code must set up a Transparent Translation Register to cause onboard I/O to use serialized non-caching accesses. The V20 and V30 series boards do not have data cache, so this is not an issue with these products.

8. ***Cache control register bits are different*** — The 68040/68060 cache control register uses different bits in the cache control register to enable the instruction and data caches. Cache control code must be rewritten when adapting V20 or V30 code to '040 or '060 based boards.

9. ***Memory Management Unit is different***

10. ***Floating Point Unit is different (firmware transcendentals)*** — The 68881/2 FPU coprocessors used on the V20 and V30 series boards are capable of computing transcendental functions (sin, cos, log...) while the 68040 and 68060's built-in FPU is capable only of add, subtract, multiply and divide. A software package available from Synergy provides transcendental functions, and may be included in the system boot EPROM or kernel to provide application code compatibility with the 68881/2.

11. ***Snooping*** — The '040/'060 performs bus snooping to maintain cache coherency between its internal caches and the external memory. It is important to remember that strange software problems may be the result of lack of cache coherency. Be especially careful with BLT transfers, which are not snooped, and with DMA transfers.

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

Block transfer (BLT) with DMA example

This chapter provides a C-language sample of the code required to execute BLT64 and 32-bit Block Transfers with DMA support on V452 Series boards. An assembly language version of this transfer code is also available if desired. Call Synergy customer service for machine-readable copies of the C and/or assembly language versions.

> ⚠ This sample code presumes that the CPU's MMU is OFF during the BLT transfer. Contact Synergy customer service for application notes describing how to perform BLT transfers while the MMU is ON.

The portions of code in these examples that relate specifically to the BLT read and write engines should be compiler-independent.

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



To execute a block transfer with DMA, the CPU must use the MOVES instruction and therefore must be in Supervisor mode. Although it is in Supervisor mode, the CPU can use the DMA engine to move data to or from user space by appropriately setting up the SFC and DFC registers in the CPU. For more information on setting up these registers see the ***VME Master BLT*** chapter in Section 5. For some operating systems, the Supervisor requirement means that block transfers must be done with a system call. In this case, the CPU code for the BLT must be part of the device driver.

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

BLT module in C

```
/*****************************************************************************
/*
/*     v40 DMA Block Transfer Code Example -- (32bit, 64bit)
/*     Copyright 1992 Synergy Microsystems, Inc
/*
/*                                          ****
/*
/*     Routines: blt
/*
/*        Routine "blt" will move a block of RAM data to or from a correspondingly
/*     aligned (long, double long) VME data location.  The minimum transfer count
/*     is either 8(BLT32) or 16(BLT64) bytes.
/*     "blt" will partition a transfer so as not to violate local(1024) or
/*     VME(256, 1024) byte page boundaries.  Maximum performance is achieved when
/*     both source and destination addresses are on (256, 1024) byte boundaries.
/*     Otherwise, the partitions will vary according to worst case page boundary.
/*     Odd bytes at the beginning (header bytes) and end (tail bytes) are moved
/*     individually that otherwise could not be transferred through the BLT
/*     mechanism.
/*                                          ****
/*     NOTE:
/*        "blt" is structured so as to maintain all operations in CPU register
/*     space in order to facilitate parallel cached operations during DMA block
/*     transfers.
/*                                          ****
/*        Error checking is currently ENABLED. Throughput sensitive until next
/*     rev.
/*                                          ****
/*     Measured Block32 transfer performance: approx. 26.5 Mbytes/sec
/*     Measured Block64 transfer performance: approx. 59.1 Mbytes/sec
/*
/*                               -- S.I.
/*****************************************************************************/
#define MMUBLT   1

/*
/* BLT32 stuff
*/
#define VPAGE        256                          /* VME page boundary BLT 32          */
#define ALIGN          4                          /* Transfer engine address alignment */
#define MCNT             8                        /* Minimum transfer count            */

/*
/* BLT64 stuff
*/
#define VPAGE64      1024                         /* VME page boundary BLT 64          */
#define ALIGN64         8                         /* BLT64 xfer engine address alignment */
#define MCNT64         16                         /* Minimum transfer count for block64 */


/*
/* --- IO registers ---
*/
#define BL_BASE                ( *(volatile int *const)0xfe600000 )
```


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

BLT module in C (cont.)

```
/*BLT 32 control register                                                    */
#define BL_32               ( *(volatile int *const)0xfe600010 )
/*BLT 64 control register                                                    */
#define BL_64               ( *(volatile int *const)0xfe600020    )

/*Status register address                                                    */
#define BL_STAT             ( *(volatile unsigned char *const)0xfe380002 )



/*
/* General stuff
*/
#define LPAGE       1024                        /* Local page boundary      */
#define BL_ERR      0x10                      /* BLT error status mask      */
#define ERROR         -1
#define OK             0

/*
/* Macros
*/
#define  MIN(a,b)     ( ((a)<(b))?(a):(b) )     /* Return minimum value     */
#define is64(a)     ( (a & 0x80000000) )      /* msbit signifies block64    */
#define isTrue(a)   ( (a & 0x7fffffff) )      /* boolean for Write direction */


/*
/*    blt
/*  -- Block(32bit, 64bit) transfer routine.
/*
/*       unsigned char *vme;
/*       -- VME data address
/*
/*       unsigned char *local;
/*       -- Local data address (4,8)byte aligned with VME address. (see above)
/*
/*       unsigned int count;
/*       -- Count of data BYTES. Should be at least (8,16) bytes. (see above)
/*
/*       unsigned int direction;
/*       -- Direction of data, a boolean TRUE(nonzero) is transfer data TO VME.
/*       -- Bit31 signifies blt64 as opposed to blt32
/*
/*       Returns:  0 for o.k.
/*              -1 for error                                                */

int blt(vme,local,count,direction)
unsigned char *vme;                           /* VME address              */
unsigned char *local;                       /* Local address              */
unsigned int count;                           /* Number of transfer bytes */
unsigned int direction;                     /* Boolean for Write,~Read    */
{
  register unsigned int tcnt,vb,lb,lcnt,Wdir,Ocnt;
  register unsigned char *vp,*lp;
  unsigned SavCacr;
```


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

BLT module in C (cont.)

```
  /* Attempt to use registers exclusively. */
  Wdir = direction;                                   /* if true, write to VME          */
  lcnt = count;                                       /* Data byte count                */
  lp = local;                                         /* Local RAM pointer          */
  vp = vme;                                           /* VME pointer                */
  Ocnt = 0;                                           /* Odd data byte count            */

#if MMUBLT
  /*
  /* Save and turn off data cache
  */
  asm("nop");
  asm(".short 0xF478");                               /* cpusha, dc                     */
  asm("movec cacr,d0");
  asm("movel d0, %0": "r=" (SavCacr) );
  asm("andl  #0x7FFFFFFF, d0");                       /* data cache off                 */
  asm("movec d0, cacr");

#else
  /*    -- Setup CPU for special move instuctions                                       */
  asm("movel #4,d0");                          /* Undefined half-fakee           */
  asm("movec d0,sfc");                         /* Force invalid cache tags for moves instruc */
  asm("movec d0,dfc");                         /* and set fc for blt DMA mechanism       */
#endif

  /* Alignment check of Local to VME address
  /* according to block64 or block32                                                  */
  if( is64(Wdir) )
  {
  /* check alignment value for vme address with next double long word                 */
    /* If not aligned, find number of Odd data bytes                                  */
    if( (int)vp & (ALIGN64-1) )
      Ocnt = ALIGN64 - ((int)vp & (ALIGN64-1));

  /* check that local ram, when adjusted with Odd data bytes                          */
    /* is also aligned, if not, we cannot go further.                                 */
    if( ((int)lp + Ocnt) & (ALIGN64-1) )
        return ( ERROR );                          /* Cache state not restored       */
  }
  else
  {
  /* check alignment value for vme address with next long word                        */
    /* If not aligned, find number of Odd data bytes                                  */
    if( (int)vp & (ALIGN-1) )
      Ocnt = ALIGN - ((int)vp & (ALIGN-1));

  /* check that local ram, when adjusted with Odd data bytes                          */
    /* is also aligned, if not, we cannot go further.                                 */
    if( ((int)lp + Ocnt) & (ALIGN-1) )
        return ( ERROR );                          /* Cache state not restored       */
  }
 /* "Manually" move any required header bytes to achieve alignment                    */
  if( Ocnt )
```


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

BLT module in C (cont.)

```
  {
    /* Check if write to VME or read from VME                                    */
    if ( isTrue(Wdir) )
      for( ; Ocnt > 0 ; Ocnt--,lcnt-- ) *vp++ = *lp++;
    else
      for( ; Ocnt > 0 ; Ocnt--,lcnt-- ) *lp++ = *vp++;
  }


  if( is64(Wdir) )                                  /* Block64 transfers       */
  {
     while( lcnt >= MCNT64 )
    {
    /* Determine number of bytes to next vme page boundry                        */
      vb = ( (~((int)vp) & (VPAGE64 - 4)) + 4 );

    /* Determine number of bytes to next local page boundry                      */
      lb = ( (~((int)lp) & (LPAGE - 4)) + 4 );

    /* Calculate the resulting block size                                        */
      /* from the minimum of next local page,  vme page or                       */
      /* number of remaining bytes                                               */
      tcnt = MIN(MIN(lb,vb), lcnt);

      /* Adjust to word count, (64bit transfers)                                 */
      /* and guarantee proper alignment                                          */
      tcnt = (tcnt >> 1) & ~3;

      if( isTrue(Wdir) )                            /* Block64 WRITE TO VME    */
      {
#if MMUBLT
        Ocnt = *((unsigned int *)lp);
          asm("movel %1,%0" : "=g" (BL_BASE) : "g" (tcnt) ) ;
        asm("movel d0,d0");
        asm("movel %1,%0" : "=g" (BL_64) : "g" (tcnt) );
        asm("nop");
        asm("movel %1,%0" : "=g" (*vp) : "g" (Ocnt) );
        asm("nop");
        asm("movel %1,%0" : "=g" (Ocnt) : "g" (*lp) );
#else
        asm("movel %1,%0" : "=g" (BL_64) : "g" (tcnt) );
        asm("movel d0,d0");
        Ocnt = *((unsigned int *)lp);
        asm("movesl %1,%0" : "=g" (*vp) : "g" (Ocnt) );
        asm("nop");
        asm("movesl %1,%0" : "=g" (Ocnt) : "g" (*lp) );
#endif
```


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

BLT module in C (cont.)

```
        /**********************************/
        /*  Block64 DMA WRITE     */
        /**********************************/

      }
      else                                                  /* Block64 READ FROM VME              */
      {
#if MMUBLT
         asm("movel %1,%0" : "=g" (BL_BASE) : "g" (tcnt) ) ;
        asm("movel d0,d0");
        asm("movel %1,%0" : "=g" (BL_64) : "g" (tcnt) );
        asm("nop");
        asm("movel %1,%0" : "=g" (Ocnt) : "g" (*vp) ) ;
        asm("movel %1,%0" : "=g" (*lp) : "g" (Ocnt) ) ;
#else
        asm("movel %1,%0" : "=g" (BL_64) : "g" (tcnt) ) ;
        asm("movel d0,d0");
        asm("movesl %1,%0" : "=g" (Ocnt) : "g" (*vp) ) ;
        asm("movesl %1,%0" : "=g" (*lp) : "g" (Ocnt) ) ;
#endif

        /**********************************/
        /*  Block64 DMA READ      */
        /**********************************/

      }

      /* check for BLT Error during DMA transfers                                              */
      /* Optional   --                                                                         */
        if( !(BL_STAT & BL_ERR) )
          return ( ERROR );                             /* Cache state not restored          */


      tcnt <<= 1;                                       /* Readjust to byte count            */
      vp += tcnt;                                       /* Adjust address pointers           */
      lp += tcnt;                                       /*                                   */
      lcnt -= tcnt;                                     /* And adjust remaining byte count   */
    }
  }
  else                                                  /* Block32 transfer                  */
  {
     while ( lcnt >= MCNT )
    {
        /* find space available to next page boundry                                          */
        vb = ( (~((int)vp) & (VPAGE - 4)) + 4 );
        lb = ( (~((int)lp) & (LPAGE - 4)) + 4 );

      /* Calculate the resulting block size from the minimum of                               */
      /* next local page, vme page or number of remaining bytes                               */
      /* and guarantee proper alignment                                                       */
        tcnt = MIN(MIN(lb,vb), (lcnt & ~3));

        if( isTrue(Wdir) )                              /* Block32 WRITE TO VME              */
        {
```


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

BLT module in C (cont.)

```
#if MMUBLT
         Ocnt = *((unsigned int *)lp);
         asm volatile ("movel %1,%0" : "=g" (BL_BASE) : "g" (tcnt) ) ;
       asm volatile ("movel d0,d0");
         asm volatile ("movel %1,%0" : "=g" (BL_32) : "g" (tcnt) ) ;
       asm volatile ("nop");
         asm volatile ("movel %1,%0" : "=g" (*vp) : "g" (Ocnt) ) ;
         asm volatile ("nop");
         asm volatile ("movel %1,%0" : "=d" (Ocnt) : "g" (*lp) ) ;
#else
         asm("movel %1,%0" : "=g" (BL_BASE) : "g" (tcnt) ) ;
         asm("movel d0,d0");
         Ocnt = *((unsigned int *)lp);
         asm("movesl %1,%0" : "=g" (*vp) : "g" (Ocnt) ) ;
         asm("nop");
         asm("movesl %1,%0" : "=g" (Ocnt) : "g" (*lp) ) ;
#endif

       /*********************************/
       /*  Block32 DMA WRITE     */
       /*********************************/


       }
     else                                                    /* Block32 READ FROM VME              */
     {
#if MMUBLT
         asm("movel %1,%0" : "=g" (BL_BASE) : "g" (tcnt) ) ;
       asm("movel d0,d0");
         asm("movel %1,%0" : "=g" (BL_32) : "g" (tcnt) ) ;
       asm("nop");
       asm("movel %1,%0" : "=g" (Ocnt) : "g" (*vp) ) ;
       asm("movel %1,%0" : "=g" (*lp) : "g" (Ocnt) ) ;
#else
       asm("movel %1,%0" : "=g" (BL_BASE) : "g" (tcnt) ) ;
       asm("movel d0,d0");
       asm("movesl %1,%0" : "=g" (Ocnt) : "g" (*vp) ) ;
       asm("movesl %1,%0" : "=g" (*lp) : "g" (Ocnt) ) ;
#endif

       /*********************************/
       /*  Block32 DMA READ     */
       /*********************************/


     }

     /* check for BLT Error during DMA transfers                                                */
     /* Optional   --                                                                           */
       if( !(BL_STAT & BL_ERR) )
         return ( ERROR );                                   /* Cache state not restored       */


     vp += tcnt;                                             /* Adjust address pointers  */
     lp += tcnt;                                             /*                          */
     lcnt -= tcnt;                                           /* And adjust remaining byte count  */
   }
  }
```


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

```
 /* "Manually" move odd tail bytes by hand                                    */
  if( lcnt )
  {
    if( isTrue(Wdir) )
      for( ; lcnt > 0 ; lcnt-- ) *vp++ = *lp++;
    else
      for( ; lcnt > 0 ; lcnt-- ) *lp++ = *vp++;
  }

#if MMUBLT
  /*
  /* Restore data cache
  */
  asm("movel %0, d0":: "r" (SavCacr) );
  asm("movec d0, cacr");
#endif

  return(OK);
}
```


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

Flash EPROM programming tools

This chapter provides a C-language code modules that can be used to erase, write, and verify data on Flash EPROM devices that are installed in PROM sockets 0 and/or 1 on V452 Series boards.



For more information, see the *EPROM* chapter in Section 4.

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

```
/*****************************************************************************
/*
/*  flashTools for Synergy boards
/*
/*  Contains routines for programming and erasing FLASH ROMs using Intel Quick-Pulse Programming Algorithm
/*  -- Modified to accomodate 28f020
/*  -- Invoke doFlash() for full test
/*  -- supports Intel and AMD
/*
/*     -- S.I.
/*     Synergy Microsystems  (c) Copyright 1993
/*****************************************************************************                                    */
#include "../CSTUFF/cstuff.h"

typedef int status;

#define OK 0
#define ERROR -1

#ifdef V400
/* Define for access to single byte wide ROM                                                                      */
typedef unsigned char accessType;
#define NUM_ROMS  1
#define JMPR_STRG   "<JK17 1-3>"
#define BRD_STRG    "V400"
#else
/* Define for access to four byte wide ROMs                                                                       */
typedef unsigned accessType;
#define NUM_ROMS  4
#define JMPR_STRG   "<JG19 7-8>"
#define BRD_STRG    "V30"
#endif

/* Timeout constants to be used with Fdelay() w/ data cache OFF                                                   */
#define uSEC6    1                                          /* 6 microsecond constant              */
#define uSEC10   2                                            /* 10 microsecond constant             */
#define mSEC10   95                                           /* 10 millisecond constant             */

#define INTEL_MAN_CODE     ( (accessType)0x89898989 )
#define MEG1_FLASH_CODE    ( (accessType)0xB4B4B4B4 )
#define MEG2_FLASH_CODE    ( (accessType)0xBDBDBDBD )

#define AMD_MAN_CODE       ( (accessType)0x01010101 )
#define AMD_MEG1_FLASH_CODE ( (accessType)0xA7A7A7A7 )
#define AMD_MEG2_FLASH_CODE ( (accessType)0x2A2A2A2A )

#define F256KBIT        (0x40000/8)
#define F512KBIT        (0x80000/8)
#define F1MBIT          (0x100000/8)
#define F2MBIT          (0x200000/8)
#define F4MBIT          (0x400000/8)

/* Flash ROM command codes                                                                                        */
#define PROGcmd     ( (accessType)0x40404040 )
#define ST_PROGcmd  ( (accessType)0xC0C0C0C0 )
#define READcmd     ( (accessType)0x00000000 )
#define ERASEcmd    ( (accessType)0x20202020 )
#define ST_ERASEcmd ( (accessType)0xA0A0A0A0 )
#define IDcmd       ( (accessType)0x90909090 )
#define RESETcmd    ( (accessType)0xFFFFFFFF )
```


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

```
/* Access addresses */
#define CHARGEIT    ( (accessType *)0 )
#define FlashBase   0xfe400000
#define FCtrlReg    ( *((volatile accessType *)FlashBase) )
#define FManReg     FCtrlReg
#define FDevReg     ( *((volatile accessType *)(FlashBase + NUM_ROMS)) )

static unsigned RomByteSize;



/*
/* flashRomReset
/* Abort any programming of chip                                          */
static void flashRomReset()
{
  FCtrlReg = RESETcmd;
  FCtrlReg = RESETcmd;
  Fdelay(uSEC10);
  FCtrlReg = READcmd;
}



/*
/* flashDevCode
/* -- Returns Device code from specific FROM                              */
static accessType flashDevCode(void)
{
  register accessType devCode;

  FCtrlReg = IDcmd;
  devCode = FDevReg;
  Fdelay(uSEC10);
  FCtrlReg = READcmd;

  return(devCode);
}



/*
/* flashManCode
/* -- Returns Manufacturers code from specific FROM                       */
static accessType flashManCode(void)
{
  register accessType devCode;

  FCtrlReg = IDcmd;
  devCode = FManReg;
  Fdelay(uSEC10);
  FCtrlReg = READcmd;

  return(devCode);
}
```


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

```
/*
/*    flashCodeCheck
/*    -- Polls and displays I.D. codes from each byte lane
/*    -- Verification will return status                                    */

status flashCodeCheck(void)
{
  accessType val;

  /* default ROM size                                                      */
  RomByteSize = F256KBIT;

  val = flashManCode();
  printf("Flash ROM Manufacture code:0x%x ... ",val);

  switch( val )
  {
    case INTEL_MAN_CODE:
      printf("INTEL\n");
      break;

    case AMD_MAN_CODE:
      printf("AMD\n");
      break;

    default:
      printf("Unexpected manufacturer code!\n");
      return(ERROR);
  }

  val = flashDevCode();
  printf("Flash ROM Device code:     0x%x ... ",val);

  switch(val)
  {
    case AMD_MEG1_FLASH_CODE:
    case MEG1_FLASH_CODE:
      RomByteSize = F1MBIT;
      break;

    case AMD_MEG2_FLASH_CODE:
    case MEG2_FLASH_CODE:
      RomByteSize = F2MBIT;
      break;

    default:
      printf("Unexpected Device code!\n");
      return(ERROR);
  }

  printf("%d bit ROM\n\n... %d total bytes\n\n", RomByteSize*8, RomByteSize*NUM_ROMS);
  return(OK);
}
```


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

```
/*    takeJmpOff
/*    -- Prompts user to remove Vpp Jmpr that  supplies programming voltage to PROMs
/*    -- Returns only after removal of jumper                                    */
void takeJmpOff(boolean isWait)
{
  accessType val;

  printf("\n\n--->Take out Vpp Jumper %s for FLASH ROMs<---\n", JMPR_STRG);
  Fdelay(mSEC10);


  if( isWait )
  {
    val = flashManCode();

    while( flashManCode() == val )
      Fdelay(mSEC10);
    printf("What took so long?\n");
  }

#ifdef V400
  printf("\n--->Replace the Vcc Jumper at JK17 2-4<--- \n");
#endif
}




/*
/*    doProgCmd
/*    -- Called by QuickPulse to program Val to specific FROM address.  Expects four byte lanes for 32bit access    */
static status doProgCmd(accessType *pAddress, accessType Val)
{
  if( (unsigned)pAddress < FlashBase )
    return(ERROR);

  if( (unsigned)pAddress > (FlashBase+RomByteSize*NUM_ROMS) )
    return(ERROR);

  FCtrlReg = PROGcmd;
  *pAddress = Val;
  Fdelay(uSEC10);

  FCtrlReg = ST_PROGcmd;
  Fdelay(uSEC6);

  if( *pAddress != Val )
    return(ERROR);

  return(OK);
}
```


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

```
/*
/* getFlashData
/* -- initialize RAM with Flash Data to program                                    /*

status getFlashData(accessType *daddr, int numbytes)
{
  unsigned char *dAddr = (unsigned char *)daddr;
  int i;

  printf("Obtain Flash Data ");
  /* write simple pattern to RAM */
  for( i=0; i<numbytes; i++ )
  {
    daddr[i] = (unsigned char)i;
    if( (i & 0x1FFF) == 0)
      printf(".");
  }

  printf(" Done @ 0x%x\n", (unsigned)daddr );

  return(OK);
}



/*
/*    fGetchar
/*    -- Get data to program to FROM                                              */

static accessType fGetchar(accessType *caddr, int index)
{
  return(caddr == CHARGEIT ? 0:caddr[index]);
}


/*
/*    QuickPulse
/*    -- Programming routine for Flash ROMs as derived from QuickPulse flowchart in Intel's Memory Manual
/*    -- caddr == CHARGEIT will determine if the FROM s/b completely charged to accomadate QuickErase. Otherwise,
/*                  burn in the data at caddr.
/*    -- bSize determines size of data.  bSize == 0 defaults to size of FROM
*/
status QuickPulse(accessType *caddr, int bSize)
{
  accessType *fAddress = (accessType *)FlashBase;
  unsigned i=0,PlsCnt=0;
  status Stat = OK;

  if(caddr == CHARGEIT)
    printf("Charge Flash ROM  ");
  else
    printf("Programming Data  ");

  /* default to ROM size if 0 */
  if( bSize == 0 )
    bSize = RomByteSize;

  while( (i < bSize) && (PlsCnt < 25) )
```


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

```
  {
    if( doProgCmd(&fAddress[i],fGetchar(caddr,i) ) == OK )
    {
      /* Give active indication */
      if( (i & 0x1FFF) == 0 )printf(".");
      i++;
      PlsCnt = 0;
    }
    else
      PlsCnt++;
  }

  if( PlsCnt >= 25 )
  {
    printf("\nFlash ROM programming error detected @ 0x%x!\n", &fAddress[i]);
    flashRomReset();
    Stat = ERROR;
  }
  else
    printf(" Done\n");

  FCtrlReg = READcmd;
  return(Stat);
}




/*
/*    QuickErase
/*    -- Erase routine for Flash ROMs as derived from QuickErase flowchart in Intel's Memory Manual                */

status QuickErase(void)
{
  accessType *fAddress = (accessType *)FlashBase;
  unsigned i=0,PlsCnt=0;
  status Stat = OK;

  if( (Stat = QuickPulse(CHARGEIT,0)) != OK )
    return(Stat);

  printf("Erase Flash ROM   ");
  while( (i < (RomByteSize)) && (PlsCnt < 3000) )
  {
    FCtrlReg = ERASEcmd;
    FCtrlReg = ERASEcmd;
    Fdelay(mSEC10);

    while( i < RomByteSize )
    {
      fAddress[i] = ST_ERASEcmd;
      Fdelay(uSEC6);

      if( fAddress[i] == (accessType)0xFFFFFFFF )
      {
        /* Give active indication */
        if( (i & 0x1FFF) == 0 )printf(".");
        i++;
      }
```


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

```
      else
      {
        PlsCnt++;
        break;
      }
    }
  }

  if( PlsCnt >= 3000 )
  {
    printf("\nFlash ROM ERASE error detected @ 0x%x!\n", &fAddress[i]);
    flashRomReset();
    Stat = ERROR;
  }
  else
    printf(" Done\n");

  FCtrlReg = READcmd;
  return(Stat);
}



/*
/*    Fdelay
/*    -- Simple delay routine to provide programming timeout routines
/*    -- It should be obvious that this routine is sensitive to speed of board and processor!   */

Fdelay(unsigned T)
{
  unsigned i,j;

  for( i=0; i<T; i++ )
    for( j=0; j<T; j++);
}



/*
/*    flashDataVerify
/*    -- Verify address pattern that should reside in FROMs after data programming             */

static status flashDataVerify(accessType *vaddr, int bSize)
{
  accessType *fAddress = (accessType *)FlashBase;
  unsigned i,Verr = 0;

  if( bSize == 0 )
    bSize = RomByteSize;

  printf("Data verification ");
  for( i = 0; i<bSize; i++ )
  {
    if( fAddress[i] != vaddr[i] )
    {
      if( Verr++ < 10 )
        printf("\n0x%x(0x%x) <> 0x%x(0x%x)", \
        (unsigned)&fAddress[i], fAddress[i], (unsigned)&vaddr[i], vaddr[i] );
    }
```


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

```
    /* Give active indication  */
    if( (i & 0x1FFF) == 0 )
      printf(".");
  }

  if( Verr != 0 )
  {
    printf("\n");
    return(ERROR);
  }

  printf(" o.k.\n");
  return(OK);
}



/*
/* Complete test for Flash ROMs
/*
/* -- Check Flash I.D. code(s)
/* -- Size Flash Rom(s)
/* -- Erase, then program incrementing pattern
/* -- Verify pattern
/* -- Erase and report status                                                    */

status doFlash(void)
{
  accessType *cAddr;
  int mptr;
  status Stat;

  printf("\n\n\n  << FLASH ROM TEST for %s CPU boards >>\n", BRD_STRG);

  printf("\n*** Check for proper codes ***\n");
  if( (Stat = flashCodeCheck()) != OK )
  {
    printf("\nCheck the Vpp (+12v) jumper %s\n", JMPR_STRG );
    printf("to enable program mode for Flash Roms!\n\n");
    return(Stat);
  }

  printf("*** Program, Verify and Erase Flash ROM ***\n");

  /* Write incrementing pattern in RAM                                          */
  mptr = malloc( RomByteSize*NUM_ROMS + 0x10);
  cAddr = (accessType *)( (mptr + 0x10) & 0xFFFFFFF0 );

  if( (Stat = getFlashData(cAddr, RomByteSize*NUM_ROMS)) == OK )
    if( (Stat = QuickErase()) == OK )
      if( (Stat = QuickPulse(cAddr,0)) == OK )
        if( (Stat = flashDataVerify(cAddr,0)) == OK )
          if( (Stat = QuickErase()) == OK )

  takeJmpOff(FALSE);

  printf("\n**********************\n");
  if( Stat == OK )
```


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

```
    printf("*** FLASH ROM PASS ***\n");
  else
    printf("*** FLASH ROM FAIL ***\n");
  printf("**********************\n");

  free(mptr);
  return(Stat);
}


/*
/* ramJet
/* -- Flash utility to burn data from pAddr into Flash ROM.
/* -- If bSize is 0, default to ROM size.
/* Return: status      */

status ramJet( accessType *pAddr, int bSize )
{
  status Stat;

  if( (Stat = flashCodeCheck()) == OK )
    if( (Stat = QuickErase()) == OK )
      if( (Stat = QuickPulse(pAddr,bSize)) == OK )
        if( (Stat = flashDataVerify(pAddr, bSize)) == OK )

  return(Stat);
}
```


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

Timer code examples

This assembly language code example shows the use of the 2692 counter/timer as a 10 ms interrupt timer, and the use of the 48T18 Calendar chip as a date and time reference.

Like most operating system drivers, this code maintains a time of day value as a variable in RAM. Every 10 ms the timer interrupt service routine increments the time of day variable. At power up or reset, the time of day variable is initialized from the time and date found in the 48T18 Calendar chip. The Calendar chip can itself be initialized by a special call to the driver.

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

```
00001                       nam      clock
00002                       ttl      Sig   2692
00003                       psect    clock,$C01,$a000,0,0,ClkEnt
00004   *  This uses the 16 bit counter/timer in a Signetics 2692 to generate
00005   *  the time slice interrupts. For initial date and time of day it uses a 48T18
00006   *  Time Keeper calendar/clock chip with battery back-up.
00007
00008   fe280003 DUARTAdr   equ      $FE280003          ;DUART base address
00009   00000000 Bundle     equ      0                  ;=0 means serial & ctr/timer interrupts are bundled.
00010   0000001d ClkVect    equ      29                 ;Vector num for 2692 (bundled) interupt
00011   0000001b TimerVect  equ      27                 ;Vector number for separate Timer/ctr interrupt
00012   fe280003 ClkPort    equ      $FE280003          ;ctr/timer base address.
00013
00014 * Clock type: Sig2692                                ;  initialize for 10 ms intervals
00015 * some 2692 port offsets.
00016   00000003 MRA        equ      $03                   ;mode register A
00017   00000007 SRA        equ      $07                   ;Status reg./clk sel. reg. A
00018   0000000b CRA        equ      $0B                   ;command register A
00019   0000000f TRHA       equ      $0F                ;xmit/rcv holding register A.
00020   00000013 ACR        equ      $13                   ;Aux. control register/IPCR.
00021   00000017 IMR        equ      $17
00022   0000001b CTUR       equ      $1B                ;MSB counter register.
00023   0000001f CTLR       equ      $1F                ;LSB counter register.
00024   00000037 IOPC       equ      $37                ;INPUT/Output port config. register.
00025   0000003b SET        equ      $3B                ;output bit set/ start ctr. command.
00026   0000003f RSET       equ      $3F                ;output bit reset/ stop ctr. command.
00027   00000017 ISR        equ      IMR
00028   *
00029   00000008 IBIT       equ      8                  ;byte mask for clock bit in ISR and IMR.
00030   00384000 XTAL       equ      3686400            ;The clock freq which is driving ctr/timer
00031   00000064 TicksSec   equ      100                ;number of ticks per second
00032   00000480 Tck1       equ      XTAL/16/2/TicksSec    ;Value for cntr to produce 10 ms intervals.
00033
00034   *---------------------------------------
00035   *  These symbols describe the 48T18 calendar/clock chip. It records date and
00036   *  time in BCD 24 hour format. It also has 2k of nonvolatile RAM. See the
00037   *  utility called StopOsc which can be used to stop the oscillator if one
00038   *  anticipates not using the computer for a long time. Stopping the oscillator
00039   *  prolongs the battery life. This program automatically restarts oscillator
00040   *  if it discovers it has been stopped.
00041   *
00042   fe100000 CC_SRAM    equ      $FE100000          ;FWA of nonvolatile RAM in clock/calendar chip
00043   000007f8 CC_CTRL    equ      $7F8               ;Offset from CC_SRAM for control register
00044   000007f9 CC_SEC     equ      $7F9               ;The seconds byte.
00045   000007fa CC_MIN     equ      $7FA               ;The minutes byte.
00046   000007fb CC_HR      equ      $7FB                  ;The hours byte.
00047   000007fc CC_DAY     equ      $7FC               ;The day of week byte.
00048   000007fd CC_DATE    equ      $7FD               ;The day of month byte.
00049   000007fe CC_MO      equ      $7FE               ;The month byte.
00050   000007ff CC_YR      equ      $7FF                  ;The low 2 digits of year byte. (Hi 2 digits
                                                              assumed =19.
00051
```


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

```
00001
00002
00003
00004
```

```
00052   00000040 RBit       equ       $40                 ;The READ bit in the control register.
00053   00000080 WBit       equ       $80                 ;The WRITE bit in the control register.
00054   00000080 StopBit    equ       $80                 ;STOP bit in the seconds byte (stops oscillator).
00055
00057
00058   ****************************************
00059    * Entry point to clock initialization routine
00060   ****************************************
00061   * Subroutine ClkEntry
00062   *   Clock initialization entry point. If the month specified
00063   * in the caller's input is zero, the system time and date
00064   * is set from the clock/calendar chip. Otherwise, the chip is
00065   * updated from caller's data.
00066   *
00067   * Passed: (a5)=points to caller's input parameters
00068   *   R$d0.l(a5)=Time (00hhmmss)
00069   *   R$d1.l(a5)=Date (yyyymmdd)
00070   *   R$d2.l(a5)=Day of week (0=Sun, 1=Mon. etc.)
00071   *        (a6)=points to variables in vsect (variable space) at end of listing.
00072   *
00073   * Returns: cc=carry set if there is an error. d1.w = error code.
00074   * The 2 32-bit packed "time" and "date" values referenced above are packed in binary
00075   * form, NOT BCD; "yyyy" represents a 16 bit binary integer. "dd" is an 8 bit
00076   * binary integer, etc.
00077
00078   0000=4a6e  ClkEnt     tst.w     D_Tick(a6)          ;is counter/timer already running?
00079   0004  664c            bne.s     ClkOn                  ;skip timer initialization if so
00080
00081   0006  1d7c            move.b    #TicksSec,D_Tick(a6);set tick = ticks/sec
00082   000c  4a7a            tst.w     Common(PC)             ;Which clock int routine to use?
00083   0010  6608            bne.s     Clk2
00084
00085   0012  41ee            lea       ClkSrv1(a6),a0      ;get address of bundled service routine
00086   0016  701d            moveq.l   #ClkVect,d0         ;get vector number
00087   0018=60ff             bra.s     clk3
00088
00089   001a  41ee  Clk2      lea       ClkSrv2(a6),a0      ;get address of separate timer int. routine
00090   001e  701b            moveq.l   #TimerVect,D0
00091   0020  e580  Clk3      asl.l     #2,D0                  ; multiply vector number by 4
00092   0022  2240            movea.l   D0,a1                  ;=interrupt vector address.
00093   0024  2288            move.l    a0,(a1)             ;point vector at clock service routine.

00094   0026  267c            movea.l   #ClkPort,a3         ;get the ctr/timer address
00095   002c  4a2b            tst.b     RSET(A3)     ;stop the ctr/timer by reading the RSET port just to be safe.
00096   0030  177c            move.b    #(Tck1>>8),CTUR(A3)     ;set upper byte of count.
00097   0036  177c            move.b    #($FF&Tck1),CTLR(A3)    ;set lower byte of count.
00098   003c  177c            move.b    #$F0,ACR(A3)            ;use set#2, timer mode with ext. clk/16
00099   0042  4a2b            tst.b     SET(A3)                       ;start the counter.
00100   0046  002e            ori.b     #IBIT,D_DuImr(a6)       ;set IMR mask bit record in memory.
00101   004c  177a            move.b    D_DuImr(PC),IMR(A3)     ;enable interrupt.
00102   *
00103   * When control gets here the counter/timer is running, generating 10ms
00104   * time slice interrupts. Now deal with date and time.
00105   0052  207c  ClkOn     movea.l   #CC_SRAM,A0           ;Point A0 at beginning of nonvolatile RAM.
```


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

```
00106  0058=202d            move.l   R$d1(A5),d0        ;=user yr/mo/day
00107  005c  e088            lsr.l    #8,d0                  ;lo byte = months number.
00108  005e  4a00            tst.b    d0
00109  0060 6700             beq.w    RdClk                  ;is the user spec. month =0? If so, read
                                                             time from clock.
00110
00111  * if control comes here, calendar/clock must be set from user-specified values.
00112 0064 117c              move.b   #RBit,CC_CTRL(A0) ;Read seconds byte to see if oscill is on.
00113 006a 1028              move.b   CC_SEC(A0),D0
00114 006e 6a0a              bpl.s    Osc_On             ;Jump if osc. isn't stopped.
00115
00116 0070 117c              move.b   #0,CC_CTRL(A0)     ;It's stopped. Clear the W bit.
00117 0076 6100              bsr.w    StartOsc              ;This starts the oscillator.
00118 007a 117c  Osc_On      move.b   #0,CC_CTRL(A0)   ;Make sure the R & W bits are 0.
00119 0080=4ced              movem.l  R$d0(A5),D0/D1        ;Retrieve the user's specified time/date.
00120 0086 6100              bsr.w    SetVar            ;Set system time & date variables.
00121 008a 117c              move.b   #WBit,CC_CTRL(A0) ;Set the write bit to 1.
00122 0090 1400              move.b   d0,d2             ;get seconds byte.
00123 0092 6100              bsr.w    ToBCD                 convert byte in d2 to BCD.
00124 0096 1142              move.b   d2,CC_SEC(A0)    ;write to seconds byte.
00125 009a e088              lsr.l    #8,d0
00126 009c 1400              move.b   d0,d2             ;get minutes byte in d2.
00127 009e 6100              bsr.w    ToBCD                 ;convert from binary 8 bit to 2 BCD digits
00128 00a2 1142              move.b   d2,CC_MIN(A0)    ;set minutes byte.
00129 00a6 e088              lsr.l    #8,d0
00130 00a8 1400              move.b   d0,d2             ;get hours byte
00131 00aa 6100              bsr.w    ToBCD
00132 00ae 1142              move.b   d2,CC_HR(A0)      ;write hours byte.
00133 00b2 1401              move.b   d1,d2             ;get day of month.
00134 00b4 6100              bsr.w    ToBCD
00135 00b8 1142              move.b   d2,CC_DATE(A0)    ;set date.
00136 00bc e089              lsr.l    #8,d1
00137 00be 1401              move.b   d1,d2             ;get month number.
00138 00c0 6100              bsr.w    ToBCD
00139 00c4 1142              move.b   d2,CC_MO(A0)      ;set month byte.
00140 00c8=2e2d              move.l   R$d2(a5),D7       ;get user specified day of week.
00141 00cc 1147              move.b   d7,CC_DAY(A0)     ;set day of week computed earlier.
00142 00d0 e089              lsr.l    #8,d1
00143 00d2 927c              sub.w    #1900,d1          ;d1.w now = year from 0 to 99.
00144 00d6 1401              move.b   d1,d2
00145 00d8 6100              bsr.w    ToBCD
00146 00dc 1142              move.b   d2,CC_YR(A0)      ;Set year number.
00147 00e0 117c              move.b   #0,CC_CTRL(A0)    ;clear Read and Write bits.
00148 *Time and date in calendar clock chip is now valid.
00149 00e6 7200              moveq.l    #0,D1              ;make sure carry isn't set.
00150 00e8 4e75 ClocExit     rts
00151
00152  * If control comes here, the clock must be read. First verify that the
00153  * oscillator is running.
00154 00ea 7e00 RdClk        moveq.l  #0,D7                ;clear D7. Used to flag if time valid.
00155 00ec 117c              move.b   #RBit,CC_CTRL(A0) ;Set the read bit.
00156 00f2 1028              move.b   CC_SEC(A0),d0     ;Read seconds. MSB is osc. stop bit.
00157 00f6 6a32              bpl.s    OscOn1
00158
00159 00f8 117c              move.b   #0,CC_CTRL(A0)    ;osc. is stopped!  So start it.
```


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

```
00160 00fe 6100              bsr.w    StartOsc          ;This starts Osc., but time & date are almost
                                                            certainly wrong!
00161 0102 47ee              ea.l     Bad(a6),A3
00162 0106 43e8              lea.l    CC_SEC(A0),A1     ;Point A1 to beginning of time/date registers.
00163 010a 117c              move.b   #WBit,CC_CTRL(A0)
00164 0110 7006              moveq.l  #6,D0
00165 0112 13b3 BadDate      move.b   0(A3,D0),0(A1,D0) ;Copy Jan. 1, 1901, 01:01 into clock.
00166 0118 51c8              dbf      d0,BadDate
00167
00168 011c 7eff              moveq.l  #-1,d7            ;flag invalid time & date in d7.
00169 011e 117c              move.b   #0,CC_CTRL(A0)
00170 0124 117c              move.b   #RBit,CC_CTRL(A0) ;Put it back in the read mode.
00171 012a 7000 OscOn1       moveq.l  #0,D0
00172 012c 2200              move.l   d0,d1
00173 012e 2400              move.l   d0,d2
00174 0130 1428              move.b   CC_HR(A0),d2      ;Read the hours. This is 2 BCD digits.
00175 0134 6168              bsr.s    FromBCD              ;convert from BCD to 8 bit binary.
00176 0136 1002              move.b   d2,d0
00177 0138 1428              move.b   CC_MIN(A0),d2     ;Read minutes byte.
00178 013c 6160              bsr.s    FromBCD
00179 013e e188              lsl.l    #8,D0
00180 0140 1002              move.b   d2,d0             ;Pack minutes byte into D0.
00181 0142 1428              move.b   CC_SEC(A0),D2     ;read seconds digit.
00182 0146 6156              bsr.s    FromBCD
00183 0148 e188              lsl.l    #8,D0
00184 014a 1002              move.b   d2,d0             ;Pack seconds into time. D0 =00hhmmss
00185 014c 1428              move.b   CC_YR(A0),d2      ;Read years
00186 0150 614c              bsr.s    FromBCD
00187 0152 1202              move.b   d2,d1
00188 0154 d27c              add.w    #1900,d1

00189 0158 1428              move.b   CC_MO(A0),d2
00190 015c 6140              bsr.s    FromBCD
00191 015e e189              lsl.l    #8,d1
00192 0160 1202              move.b   d2,d1             ;Pack months byte into d1.
00193 0162 1428              move.b   CC_DATE(A0),d2    ;read the day of month
00194 0166 6136              bsr.s    FromBCD
00195 0168 e189              lsl.l    #8,D1
00196 016a 1202              move.b   d2,d1             ;Pack day into D1. D1 = yyyymmdd
00197 016c 117c              move.b   #0,CC_CTRL(A0)    ;Turn read mode off.
00198 0172 617e              bsr.s    SetVar            ;Set system variables for mo, yr. day, etc.
00199 0174 4a87              tst.l    d7                   ;Is date invalid?
00200 0176 6b04              bmi.s    CCBad
00201 0178 7200              moveq.l  #0,d1                ;make sure carry isn't set
00202 017a 4e75              rts
00203
00204 017c=323c CCBad        move.w   #E$NotRdy,d1      ;Flag device not ready with hi byte =1.
00205 0180 0041              ori.w    #$100,d1
00206 0184=003c              ori.w    #Carry,CCR           ;User must specify time! Clock isn't set.
00207 0188 4e75              rts
```


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

```
00209 ************************************
00210 * Given a data byte in D2, convert it to 2 BCD digits in D2. Clobbers D3
00211 * in the process.
00212 *
00213 018a 0282 ToBCD         andi.l    #$FF,D2               ;Make sure hi bytes =0.
00214 0190 84fc               divu.w    #10,d2              ;Low word = quotient = tens digit.
00215 0194 2602               move.l    d2,d3
00216 0196 e98a               lsl.l     #4,d2                   ;shift quotient 4 bits left.
00217 0198 4843               swap      d3                      ;get remainder in low byte of d3.
00218 019a d403               add.b     d3,d2
00219 019c 4e75               rts
00220
00221 ************************************
00222 * Given a 2 digit BCD value in D2.B, convert it to a 1-byte binary value in
00223 * D2.B. Clobbers D3.
00224 *
00225 019e 0282 FromBCD       andi.l    #$FF,d2
00226 01a4 2602               move.l    d2,d3
00227 01a6 e88b               lsr.l     #4,d3                   ;d3=tens digit.
00228 01a8 0242               andi.w    #$0f,d2               ;d2=ones digit.
00229 01ac c6fc               mulu.w    #10,d3
00230 01b0 d443               add.w     d3,d2
00231 01b2 4e75               rts
00232
00233 ************************************
00234 * This is called when it is discovered that the calendar/clock's oscillator
00235 * is not running. It starts it.
00236 *
00237 01b4 117c StartOsc      move.b    #WBit,CC_CTRL(A0) ;Enable write mode.
00238 01ba 117c               move.b    #0,CC_SEC(A0)       ;Set stop bit to 0.
00239


00240 01c6 117c               move.b    #0,CC_CTRL(A0)        ;Set write bit to 0.
00241 01cc 203c               move.l    #TicksSec*2,d0        ;=enough ticks for 2 seconds.
00242 01d2=4e40               bsr.w     Pause                   ;Do system call to delay for 2 secs.
00243
00244
00245
00246 01e8 4e75               rts
00247 * Clock is now running, but it needs to be set.
00248
00249 * ---------------------------------------------------------------
00250 * This table is used to set calendar to a default date.
00251 01ea 0001 Bad             dc.b         0,1,1,0,1,1,1
00252
00253 ****************************************
00254 * This takes the date & time packed in format described above and sets
00255 * the system variables D_Year, D_Month, D_Day, D_Hour, D_Min, D_Sec
00256 * This assumes that D0 = 00hhmmss and D1 = yyyymmss. This means that the year
00257 * in D1 is a 16 bit binary number, month is 1 binary byte, etc. This is not BCD.
00258 * Leaves D0 & D1 unchanged.
00259 *
00260 01f2 1d41 SetVar         move.b    D1,D_Day(a6)
00261 01f6 e099               ror.l     #8,D1
```


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

```
00262 01f8 1d41              move.b   D1,D_Month(a6)
00263 01fc e099              ror.l    #8,D1
00264 01fe 3d41              move.w   D1,D_Year(a6)
00265 0202 4841              swap     D1
00266 0204 1d40              move.b   D0,D_Sec(a6)
00267 0208 e098              ror.l    #8,D0
00268 020a 1d40              move.b   D0,D_Min(a6)
00269 020e e098              ror.l    #8,D0
00270 0210 1d40              move.b   D0,D_Hour(a6)
00271 0214 e098              ror.l    #8,D0
00272 0216 e098              ror.l    #8,D0
00273 0218 4e75              rts
00276 ******************************
00277 * Clock interrupt service routine for configuration with timer interrupting
00278 * in common with the other 2692 interrupts.
00279 *
00280  021a  48e7 ClkSrv1      movem.l  D0/A3,-(A7)     ;save registers
00281  021e  267c              movea.l  #DUARTAdr,A3      ;point at DUART
00282  0224  102b              move.b   ISR(a3),d0      ;ctr/timer causing Interrupt?
00283  0228  0200             andi.b   #IBIT,D0
00284  022c  670e             beq.s    NotClk
00285
00286  022e 102b               move.b   RSET(a3),d0  ;"stop ctr" command clears the IRQ & doesn't stop the
                                                          timer.
00287 0232=6100              bsr      ClkUpdate       ;do clock functions such as increment time, etc.
00288 0236 4cdf              movem.l  (A7)+,D0/A3  ;recover saved registers
00289 023a 4e73              rte
00290
00291 023c 4cd7 NotClk       movem.l  (a7),D0/A3      ;recover the saved registers
00292 0240=4ef9              jmp      S2692           ;jump to routine that handles the rest of the 2692
                                                          interrupts
00293
00294
00295 ******************************
00296 * Clock interrupt service routine for configuration with timer interrupting
00297 * separately from the 2692. It is not necessary for this routine to clear the
00298 * interrupt since the CPU hardware does that automatically.
00299 *
00300 0246 48e7 ClkSrv2       movem.l  D0/A3,-(a7)  ;save registers
00301 024a 267c              movea.l  #DUARTAdr,A3    ;point at DUART
00302 0250=6100              bsr      ClkUpdate       ;do clock functions such as increment time, etc.
00303 0254 4cdf              movem.l  (a7)+,D0/A3  ;recover saved registers
00304 0258 4e73              rte                      ;return from interrupt.
00305
00306 *
00307 ************************************************************************
00308 *
00309 *        This constant flags which interrupt routine is to be used.
00310 *
00311
00312 025a 0000 Common      dc.w     Bundle          ;=0 means cntr timer & serial on same interrupt.
00313 *                                                     not 0 means timer is on a separate interrupt.
00314
```


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

```
00315 *********************************************
00316 * This is the variable section. These variables are referenced relative to
00317 * the A6 register.
00318 *
00319                         vsect
00320   0000  0000  D_Year        dc.w    0               ;These hold the year, month, day, etc. as specified
                                                              by user, or as read from clock
00321   0002    00  D_Month       dc.b    0
00322   0003    00  D_Day         dc.b    0
00323   0004    00  D_Hour        dc.b    0
00324   0005    00  D_Min         dc.b    0
00325   0006    00  D_Sec         dc.b    0
00326   0007    00  D_Tick        dc.b    0               ;=number of ticks/second
00327   0008    00  D_DuImr       dc.b    0                  ;Saved value of 2692 IMR (interrupt mask
                                                              register) byte.
00328   0000000a    ends
```


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

2692 DUART code example

The C language code example given below provides sample coding for implementing the two on-board serial ports via the 2692 DUART. For more information, see the ***Timers & counters*** chapter in Section 4 and the ***Asynchronous serial interface*** chapter in Section 5.

```
/*
DESCRIPTION
This is a sample driver for the Synergy V452 Series on-board serial interface.
It emphasizes the 2692-dependent aspects of the code and does not
include most of the OS dependent code.

There are four channels on the V452 Series board, port 0..3.
There are 2 UART Chips 0,2
There are 2 channels in each Chip 0,1

Port    Chip    Channel
  0       0         0
  1       0         1
  2       2         0
  3       2         1

IOCTL
This driver responds to the normal ioctl codes .
Most baud rates between 50 and 38.4K baud are available.
*/
```


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

```
#define DEFAULT_BAUD    9600

#define N_SIO_CHANNELS       4                          /* Number of serial I/O channels          */

/* Local I/O address map */
#define V400_RES_BL             ((char *) 0xfe28003f)   /* first uart RESET register              */
#define  V400_S2692_A           ((char *) 0xfe280003)   /* first S2692 dual uart                  */
#define  V400_S2692_B           ((char *) 0xfe200003)   /* second S2692 dual uart          */

#define V400_DAUGHTER_LVL     4                          /* level 4 interrupts              */

/* Interrupt vector locations */
#define V400_TIMER_INT       30                          /* autovector level 3 for system timer in uart A
                                                                                                */
#define INT_VEC_S2692        29                          /* autovector level 5 for both serial ports  */

#define V400_S2692_ACR_VAL   0xe0                       /* value to enable timer mode and baud table 1
                                                                                                */
#define V400_TYC               1843200                  /* 1/2 the crystal freq. on uart chips     */

/* Signetics 2692 family definitions */
#define PORT_0               0
#define PORT_1               1

/* 2692 registers  */
#define   TY_MRA             0x0                         /* 0                                  */
#define   TY_SRA             0x4                         /* 4                                  */
#define   TY_CSRA      TY_SRA
#define   TY_CRA             0x8                         /* 8                                  */
#define   TY_THRA            0xc                           /* c                                */
#define   TY_RHRA      TY_THRA
#define   TY_IPCR            0x10                          /* 10                               */
#define   TY_ACR       TY_IPCR
#define   TY_ISR             0x14                        /* 14                                 */
#define   TY_IMR       TY_ISR
#define   TY_CTU             0x18                        /* 18                                 */
#define   TY_CTL             0x1c                        /* 1c                                 */
#define   TY_MRB             0x20                        /* 20                                 */
#define   TY_SRB       0x24                              /* 24                                 */
#define   TY_CSRB      TY_SRB
#define   TY_CRB             0x28                        /* 28                                 */
#define   TY_THRB            0x2c                          /* 2c                               */
#define   TY_RHRB      TY_THRB

/* reserved addresses in chip  */
#define   TY_INP             0x34                        /* 34                                 */
#define   TY_OPCR      TY_INP
#define   TY_STRT            0x38                          /* 38                               */
#define   TY_SET       TY_STRT
#define   TY_STOP            0x3c                          /* 3c                               */
#define   TY_RSET      TY_STOP

/* 2692 commands  */
#define   TYC_ET             0x04                        /* enable transmitter                 */
#define   TYC_DT             0x08                        /* disable transmitter                */
#define   TYC_MR             0x10                        /* reset mode register pointer        */
```


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

```
#define  TYC_RC              0x20                    /* reset receiver                */
#define  TYC_TX              0x30                    /* reset transmitter             */
#define  TYC_SE              0x40                    /* reset error status            */
#define  TYC_BC              0x50                    /* reset break change            */
#define  TYE_RT              0x5                     /* enable receiver and transmitter */

/* locals */
LOCAL int auxClkTicksPerSecond   = 60;
LOCAL BOOL sysAuxClkRunning      = FALSE;
LOCAL FUNCPTR sysAuxClkRoutine   = NULL;
LOCAL int sysAuxClkArg;
LOCAL BOOL auxClkIsConnected     = FALSE;

typedef struct                                       /* TY_CO_DEV                     */
    {
    TY_DEV tyDev;
    BOOL created;                                    /* true if device has really been created */
  char *tyUart;                                      /* pointer to the uart structure */
    int tyChan;                                      /* 0 for chan 0, non zero for chan 1 */
    } TY_CO_DEV;

LOCAL TY_CO_DEV tyCoDv [] =                          /* device descriptors            */
    {
    {{{{NULL}}}, FALSE, V400_S2692_A, CHAN_0},
    {{{{NULL}}}, FALSE, V400_S2692_A, CHAN_1},
    {{{{NULL}}}, FALSE, V400_S2692_B, CHAN_0},
    {{{{NULL}}}, FALSE, V400_S2692_B, CHAN_1},
    };


typedef struct                                       /* data for ioctl baud rate setting code to set
                                                        up 2692 rates                 */
    {
    int tyBaud;                                      /* resulting baud rate           */
    char tyBTable;                                   /* 0=table 0, 1=table 1, 2=either table */
    char tyBValue;                                   /* value to write to CSR[AB]     */
    } TY_CO_BAUDS;

/* only support table 1 at this time
   the speeds commented out cannot be set up on one channel without some interaction
   on the other channel */

LOCAL TY_CO_BAUDS tyBaudList [] =
    {
    /*  {50, 0, 0},  */
        {75, 1, 0},
        {110, 2, 0x11},
        {134, 2, 0x22},
        {150, 1, 0x33},
    /*  {200, 0, 0x33}, */
        {300, 2, 0x44},
        {600, 2, 0x55},
```


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

```
    /*  {1050, 0, 0x77},*/
        {1200, 2, 0x66},
        {1800, 1, 0xaa},
        {2000, 1, 0x77},
        {2400, 2, 0x88},
        {4800, 2, 0x99},
    /*  {7200, 0, 0xaa},*/
        {9600, 2, 0xbb},
        {19200, 1, 0xcc},
    /*  {38400, 0, 0xcc},  */
        {0, 0, 0},                                  /* ends with record of zero's   */
    };

/************** tyCoHrdInit - initialize the usarts.*********************************
* This routine initializes the on-board usarts. This routine must be called in supervisor mode.  */

LOCAL VOID tyCoHrdInit ()
    {
    FAST int   oldlevel;                            /* current interrupt level mask */

    oldlevel = intLock ();                          /* disable interrupts during init */
    tyCoResetChip (0);                              /* reset both ports of Chip     */
    tyCoResetChip (2);                              /* reset both ports of Chip     */

    intUnlock (oldlevel);
    }

/***********tyCoResetChip - reset a single uart chip, both channels  **************/

LOCAL VOID tyCoResetChip (port)
    int port;
    {
    FAST char *cr = tyCoDv [port].tyUart;           /* uart port base adr           */
    int delay;

    cr[TY_IMR]  = 0;                                /* clear mask register              */
    cr[TY_CRA]  = TYC_MR;                           /* reset mode reg pointer       */
    cr[TY_CRA]  = TYC_RC;                           /* reset receiver               */
    cr[TY_CRA]  = TYC_TX;                           /* reset transmitter            */
    cr[TY_CRA]  = TYC_SE;                           /* reset error status           */
    cr[TY_CRA]  = TYC_BC;                           /* reset break change           */
    cr[TY_CRB]  = TYC_MR;                           /* reset mode reg pointer       */
    cr[TY_CRB]  = TYC_RC;                           /* reset receiver               */
    cr[TY_CRB]  = TYC_TX;                           /* reset transmitter            */
    cr[TY_CRB]  = TYC_SE;                           /* reset error status           */
    cr[TY_CRB]  = TYC_BC;                           /* reset break change           */
    cr[TY_SET]  = 0xf7;                             /* clear output ,except timer output */
    cr[TY_IMR]  = 0x33;                             /* enable tx and rx interrupts      */
    cr[TY_ACR]  = V400_S2692_ACR_VAL;               /* use table 1 for baud rates       */
    for (delay = 0; delay < 1000; delay++);         /* pause to let chip recover from reset */
    }

/****************tyCoInitPort - initialize a single port**************************/

LOCAL VOID tyCoInitPort (port)
    int port;
```


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

```
    {
    FAST TY_CO_DEV *pTyCoDv = &tyCoDv [port];
    FAST char      *cr = pTyCoDv->tyUart;          /* uart port base adr          */
    FAST int        oldlevel;                      /* current interrupt level mask */
    oldlevel = intLock ();                         /* disable interrupts during init */

    if ( pTyCoDv->tyChan==0 )
    {
        cr[TY_CRA]  = TYC_MR;                      /* reset mode reg pointer      */
        cr[TY_CRA]  = TYC_RC;                      /* reset receiver              */
        cr[TY_CRA]  = TYC_TX;                      /* reset transmitter           */
        cr[TY_CRA]  = TYC_SE;                      /* reset error status          */
        cr[TY_CRA]  = TYC_BC;                      /* reset break change          */
        cr[TY_MRA]  = 0x12;                        /* 7 bits, no parity           */
        cr[TY_MRA]  = 0x7;                         /* 1 stop, no rts, no cts      */
        cr[TY_CSRA] = 0xbb;                        /* rx at 9600, tx at 9600      */
        cr[TY_CRA]  = TYE_RT;                      /* enable both rec and tx      */
    }
    else
    {
        cr[TY_CRB]  = TYC_MR;                      /* reset mode reg pointer      */
        cr[TY_CRB]  = TYC_RC;                      /* reset receiver              */
        cr[TY_CRB]  = TYC_TX;                      /* reset transmitter           */
        cr[TY_CRB]  = TYC_SE;                      /* reset error status          */
        cr[TY_CRB]  = TYC_BC;                      /* reset break change          */
        cr[TY_MRB]  = 0x12;                        /* 7 bits, no parity           */
        cr[TY_MRB]  = 0x7;                         /* 1 stop, no rts, no cts      */
        cr[TY_CSRB] = 0xbb;                        /* rx at 9600, tx at 9600      */
        cr[TY_CRB]  = TYE_RT;                      /* enable both rec and tx      */
    }
    intUnlock (oldlevel);
    }

/*****************tyCoIoctl - special device control*******************************
* This routine handles baud rate requests, and passes all other requests to tyIoctl. */

LOCAL STATUS tyCoIoctl (pTyCoDv, request, arg)
    TY_CO_DEV *pTyCoDv;                            /* device to control           */
    int        request;                            /* request code                */
    int        arg;                                /* some argument               */

    {
    FAST int     oldlevel;                         /* current interrupt level mask */
    FAST int     baudConstant,i;
    FAST STATUS  status;
    FAST char   *cr;                               /* Uart register base adr      */
    FAST TY_CO_BAUDS  *baudData;                   /* pointer to baud data        */

    switch (request)
        {
        case FIOBAUDRATE:
```


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

```
                                                    /* look thru baud rate data list for entry == arg
                                                                                                */
            for(i=0;(tyBaudList[i].tyBaud != 0)&&(tyBaudList[i].tyBaud != arg);i++) ;
            baudData = &tyBaudList[i];
                                                    /* arg not found in table                   */
            if (baudData->tyBaud == 0)
                {
                status = ERROR;                     /* baud rate out of range                   */
                break;
                }

            cr = pTyCoDv->tyUart;
                                                    /* disable interrupts during chip access    */

            oldlevel = intLock ();

            if(pTyCoDv->tyChan)
            { cr[TY_CSRB] = baudData->tyBValue; }
            else
            { cr[TY_CSRA] = baudData->tyBValue; }

            intUnlock (oldlevel);

            status = OK;
            break;

        default:
            status = tyIoctl ((TY_DEV_ID) pTyCoDv, request, arg);
            break;
        }
    return (status);
    }

/************************tyCoInt - interrupt level processing*****************************
* This routine handles interrupts from both of the Uarts
* read each chips status register to see if it need service, and do it. */

LOCAL VOID tyCoInt ()

    {
    FAST char       *cr;
    FAST char        intStatus;
    FAST TY_CO_DEV  *pTyCoDv;
    FAST int        i,chann;
    char             outChar;

                                                    /* for each channel, check for
                                                    *either receive or transmit and service
                                                    * - if xmit and nothing to send then
                                                    * disable  xmitter - if an error such as
                                                    * break, overrun, framing or parity, reset it   */
```


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

```
00001
00002
00003
00004
```

```
for (i=0;; i++)
{
    if(i>=NELEMENTS(tyCoDv))
        {
         /* done with uarts, now check timer in uart B */
        intStatus = V400_S2692_B[TY_IMR];
        if (intStatus & 0x8 )
            {
            i = (char )V400_S2692_B[TY_STOP];
            if (sysAuxClkRunning)
                {
                if (sysAuxClkRoutine != NULL)
                    (* sysAuxClkRoutine) (sysAuxClkArg);
                }
            }
        break;                                          /* done                    */
 }

    pTyCoDv = &tyCoDv[i];                               /* for channel i           */
    cr = pTyCoDv->tyUart;                               /* get uart addr           */
    chann = pTyCoDv->tyChan;                            /* get uart chann              */

    if(chann)
        { intStatus = cr[TY_SRB]; }                     /* get status              */
    else
        { intStatus = cr[TY_SRA]; }                     /* get status              */

    if(intStatus & 0x1)                                 /* RxRDY                   */
        {
        if(pTyCoDv->created)
            if(chann)
            { tyIRd(&pTyCoDv->tyDev, cr[TY_RHRB]); }
            else
            { tyIRd(&pTyCoDv->tyDev, cr[TY_RHRA]); }
        }
    if(intStatus & 0x4)                                 /* TxRDY                   */
        {
        if(pTyCoDv->created && (tyITx(&pTyCoDv->tyDev, &outChar) == OK))
        {
            if(chann)
            { cr[TY_THRB] = outChar; }
            else
            { cr[TY_THRA] = outChar; }
        }
        else
        {
            if(chann)
            { cr[TY_CRB] = TYC_DT;  }               /* disable transmitter     */
            else
            { cr[TY_CRA] = TYC_DT;  }               /* disable transmitter     */
        }
    }
    if(intStatus & 0xf0)                                /* ERRORS                  */
```


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

```
            {
            if(chann)
            { cr[TY_CRB] = TYC_SE; }                    /* reset error status          */
            else
            { cr[TY_CRA] = TYC_SE; }                    /* reset error status          */
        }
    }
}
/***************tyCoStartup - transmitter start-up routine************************
* Call interrupt level character output routine. */

LOCAL VOID tyCoStartup (pTyCoDv)
    TY_CO_DEV *pTyCoDv;                                 /* ty device to start up       */
    {
    FAST char   *cr;
    char outChar;

    cr = pTyCoDv->tyUart;                               /* get uart addr                  */

    if (tyITx (&pTyCoDv->tyDev, &outChar) == OK)
        {
        if(pTyCoDv->tyChan)                             /* chann 1                        */
        {
            cr[TY_CRB ] = TYC_ET;                       /* enable transmitter          */
            cr[TY_THRB] = outChar;
        }
        else                                                /* chann 0                    */
        {
            cr[TY_CRA]  = TYC_ET;                       /* enable transmitter          */
            cr[TY_THRA] = outChar;
        }
        }
    }
/*****************sysAuxClkDisable - turn off auxiliary clock interrupts**************/

VOID sysAuxClkDisable ()

    {
    V400_S2692_B[TY_IMR] = 0x33;                        /* turn off timer mask, leave both uarts */
    sysAuxClkRunning = FALSE;
    return(OK);
    }
/******************sysAuxClkEnable - turn auxiliary clock interrupts on**************/

VOID sysAuxClkEnable ()
    {
    FAST unsigned int temp,sctps;
    sctps = auxClkTicksPerSecond;

    if (!sysAuxClkRunning)
        {
        V400_S2692_B[TY_ACR] = V400_S2692_ACR_VAL;    /* set up counter mode for timer  */

    /* preload the preload register */
        temp = ((V400_TYC + (V400_TYC%sctps) + (sctps/2))/sctps);
        temp &= 0xffff;
        V400_S2692_B[TY_CTL] = temp & 0xff;
```


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

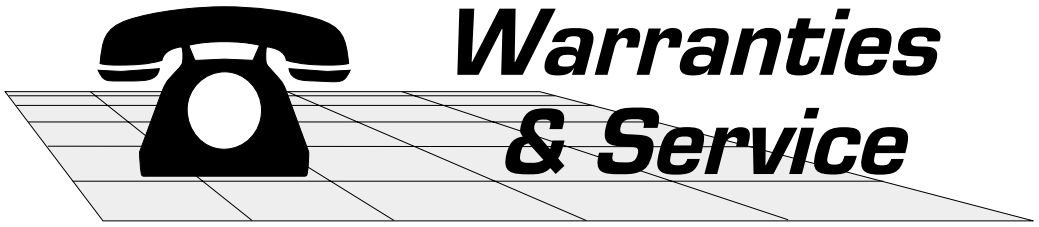
```
        V400_S2692_B[TY_CTU] = ((temp >> 8) & 0xff);

        V400_S2692_B[TY_IMR] = 0x3b;                    /* enable interrupt mask        */

        sysAuxClkRunning = (char )V400_S2692_B[TY_STRT];/* start up counter             */
        sysAuxClkRunning = TRUE;
        }
    }
```


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

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



Warranties & Service

7

This section contains information about product warranties and Synergy customer service:

- Warranty options
- Customer service

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



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



Warranty terms & options

This chapter describes the warranty terms and options provided for the V452 Series of boards.

Warranty terms

Synergy Microsystems, Inc. warrants all standard (off-the-shelf) and non-standard (custom) products to be free of defects in materials and workmanship under normal use for the applicable warranty period (as described below). This limited warranty is void if the failure has resulted from accident, abuse, alteration, or misapplication by the customer.

Product returns

The following guidelines describe warranty terms for product returns.

- **Initial product acceptance** — Synergy presumes that customers will inspect products within 14 days of receipt for conformance to the specifications stated in this manual (for standard, off-the-shelf units) and/or purchasing documentation (for custom units). Products not rejected within this period are considered by Synergy to be accepted by the customer.
- **Delivery rejection** — Products that do not conform to the specifications and standards in this manual or purchase documents can be returned to Synergy for replacement/repair. Before returning products, notify Synergy of the problem and get a Return Material Authorization (RMA) number. Board rejection will not be valid unless boards are returned in the original shipping cartons within 10 days of the receipt of the RMA number.

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

For more information about returning products, see the next chapter on *Customer service*.

If the customer adheres to these requirements, Synergy agrees to pay shipping charges, otherwise shipment costs must be paid by the customer.

- **Delivery turnaround after rejection** — Synergy's service goal is to return new or refurbished products within 14 days of the receipt of properly rejected boards that were returned in accordance with the requirements stated above and in next chapter.
- **Product returns under warranty** — Once products have been either accepted or the initial product accept/reject period has passed, products are warranted for the applicable warranty period as described below:

  For information about returning products under warranty, see the next chapter on *Customer service*.

Warranty periods

Synergy Microsystems, Inc. offers the following warranty periods:

- **90-day guarantee and limited warranty** — All standard (off-the-shelf) and non-standard (custom) products are automatically guaranteed for 90 days from the day of delivery.
- **1-year standard limited warranty** — Customers who complete payment for the product to Synergy within **30 days** of delivery receive a free warranty extension for a full year on all products covered by the payment.
- **3-year extended limited warranty** — If desired, Synergy offers an extended 3-year warranty for an additional charge. The terms for the extended warranty are identical to those listed above.

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



Customer service

Please contact Synergy Microsystems, Inc. if you have any questions, comments, or suggestions. You can contact our **customer service** department by writing or calling:

Synergy Microsystems, Inc.
9605 Scranton Rd., Suite 700
San Diego, CA 92121-1773
(858) 452-0020
(858) 452-0060 (FAX)

Reporting problems

If you encounter any difficulty with your V452 Series board, call Synergy customer service. If possible, please have the following information available to assist our staff in assessing your problem:

- V452 Series model number (silk-screened on solder side of board)
- Serial number marked on solder side of board
- V452 Series revision level (silk-screened on the solder side of board)
- ECO level (marked on solder side of board)
- Revision level of the Monitor PROM.

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

Return policies and procedures

Should it become necessary to return a board to Synergy for repair, please take the following steps.

1. Call Synergy Microsystems, Inc. customer service for a **Return Merchandise Authorization** (RMA) number. Use this number in all communications regarding the problem boards.
2. Provide the following information with all returned items:
   - V452 Series model number (solder side of the board)
   - Serial number (solder side of board)
   - V452 Series revision level (solder side of board)
   - ECO level (solder side of board)
   - Operating system and Revision level of the Monitor PROM (or other PROM/EPROM used on your board) The revision level is printed on the EPROM at location UJ13.
   - Purchase order number and billing address if the board is out of warranty.
   - Customer contact name, address, and telephone number
   - Complete description of the problem.
3. Carefully package the board to protect it during shipment; be sure that it is enclosed in an anti-static bag.
4. Mark the RMA number on the shipping container.
5. Send the board and the requested information prepaid to Synergy at the following address:

   Synergy Microsystems, Inc.
   9605 Scranton Rd., Suite 700
   San Diego, CA 92121-1773

An inspection and test charge will be applicable to all units returned for repair, unless the unit is found to be defective and under warranty. If the repair charge exceeds the inspection and test charge, we will notify you of the repair charge. The test and inspection charge will be applied to your repair charge. No repair (other than test and inspection) will be performed on products that are out of warranty until we have received your approval for the charges.

We appreciate your cooperation with these procedures. They help us give you the best possible service.

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

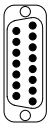
Appendix A, Connectors & Cables

This appendix contains descriptions and diagrams of the V452 connectors and specialized cabling.

- VMEbus connectors (P1 & P2)
- EZ-bus connectors (P3 & P4)
- Memory module connectors (P9, P10 & P11)
- Ethernet 10Base-T connector (P8)
- Asynchronous serial connectors (P5–P7)
- Serial I/O cabling options
- P2 serial interface option

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

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



VMEbus connectors (P1 & P2)

The P1 through P2 connectors on V452 Series boards provide the standard I/O interface to the VMEbus as listed in the table below:



The P2 connector shows the signals Synergy has assigned to the user-defined pins for rows A and C on the standard VMEbus. Both of these rows are connected to the P4 EZ-bus connector listed later in this chapter.

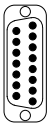
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VMEbus connector pinouts

VMEbus P1						VMEbus P2					
Pin	**Row Z[2]**	**Row A**	**Row B**	**Row C**	**Row D[2]**	**Pin**	**Row Z[1,2]**	**Row A[1]**	**Row B**	**Row C[1]**	**Row D[1,2]**
1	—	D00	BBsy\	D08	—	**1**	(D30)	(B32)	+5V	(A32)	(E32)
2	Gnd	D01	BClr\	D09	Gnd	**2**	Gnd	(B31)	Gnd	(A31)	(E31)
3	—	D02	ACFail\	D10	—	**3**	(D29)	(B30)	reserved	(A30)	(E30)
4	Gnd	D03	BG0In\	D11	—	**4**	Gnd	(B29)	A24	(A29)	(E29)
5	—	D04	BG0Out\	D12	—	**5**	(D26)	(B28)	A25	(A28)	(E28)
6	Gnd	D05	BG1In\	D13	—	**6**	Gnd	(B27)	A26	(A27)	(E27)
7	—	D06	BG1Out\	D14	—	**7**	(D25)	(B26)	A27	(A26)	(E26)
8	Gnd	D07	BG2In\	D15	—	**8**	Gnd	(B25)	A28	(A25)	(E25)
9	—	Gnd	BG2Out\	Gnd	SIIdp\	**9**	(D22)	(B24)	A29	(A24)	(E24)
10	Gnd	SysClk	BG3In\	Sysfail\	SIId0\	**10**	Gnd	(B23)	A30	(A23)	(E23)
11	—	Gnd	BG3Out\	BErr\	SIId1\	**11**	(D21)	(B22)	A31	(A22)	(E22)
12	Gnd	DS1\	BR0\	SysRes\	+3.3V	**12**	Gnd	(B21)	Gnd	(A21)	(E21)
13	—	DS0\	BR1\	LWord\	SIId2\	**13**	(D18)	(B20)	+5V	(A20)	(E20)
14	Gnd	Write\	BR2\	AM5	+3.3V	**14**	Gnd	(B19)	D16	(A19)	(E19)
15	—	Gnd	BR3\	A23	SIId3\	**15**	(D17)	(B18)	D17	(A18)	(E18)
16	Gnd	DTAck\	AM0	A22	+3.3V	**16**	Gnd	(B17)	D18	(A17)	(E17)
17	—	Gnd	AM1	A21	SIId4\	**17**	(D14)	(B16)	D19	(A16)	(E16)
18	Gnd	AS\	AM2	A20	+3.3V	**18**	Gnd	(B15)	D20	(A15)	(E15)
19	—	Gnd	AM3	A19	—	**19**	(D13)	(B14)	D21	(A14)	(E14)
20	Gnd	IAck\	Gnd	A18	+3.3V	**20**	Gnd	(B13)	D22	(A13)	(E13)
21	—	IAckIn\	SerClk(1)	A17	—	**21**	(D10)	(B12)	D23	(A12)	(E12)
22	Gnd	IAckOut\	SerDat\(1)	A16	+3.3V	**22**	Gnd	(B11)	Gnd	(A11)	(E11)
23	—	AM4	Gnd	A15	—	**23**	(D9)	(B10)	D24	(A10)	(E10)
24	Gnd	A07	IRQ7\	A14	+3.3V	**24**	Gnd	(B9)	D25	(A9)	(E9)
25	—	A06	IRQ6\	A13	—	**25**	(D6)	(B8)	D26	(A8)	(E8)
26	Gnd	A05	IRQ5\	A12	+3.3V	**26**	Gnd	(B7)	D27	(A7)	(E7)
27	—	A04	IRQ4\	A11	—	**27**	(D5)	(B6)	D28	(A6)	(E6)
28	Gnd	A03	IRQ3\	A10	+3.3V	**28**	Gnd	(B5)	D29	(A5)	(E5)
29	—	A02	IRQ2\	A09	—	**29**	(D2)	(B4)	D30	(A4)	(E4)
30	Gnd	A01	IRQ1\	A08	+3.3V	**30**	Gnd	(B3)	D31	(A3)	(E3)
31	—	-12V	+5vStdby	+12V	Gnd	**31**	(D1)	(B2)	Gnd	(A2)	Gnd
32	Gnd	+5V	+5V	+5V	—	**32**	Gnd	(B1)	+5V	(A1)	—

Notes:
1. Pins in this row connect to the P4 EZ-bus connector pin indicated in parentheses. Space is provided in these columns to write in the assigned signals, if desired. Refer to the applicable EZ-bus module User Guide for P2 pin assignments. Note that Module B's (top board in stack) P2 pin assignments apply to the P2 connector on the EZP2 adapter.
2. This row present only with optional wide (160-pin) VMEbus P1 & P2 connectors.

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



EZ-bus connectors (P3 & P4)

The P3 and P4 connectors provide an I/O interface between the V452 Series local bus and the Synergy EZ-bus for daughter modules.



Synergy offers a series of off-the-shelf daughter modules or can custom-design a board for quantity customers. For more information on the available daughter modules, see the ***EZ-bus interface*** chapter in Section 5.

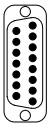
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EZ-bus connector pinouts

EZ-bus P3				EZ-bus P4						Row PD4[3]
Pin	Row A	Row B	Row C	Pin	Row A [1]	Row B[1]	Row C	Row D[1,2]	Row E[1,2]	
1	D31	Write\	Gnd	1	(C32)	(A32)	Vcc	(Z31)	—	—
2	D30	Clk	BErr\	2	(C31)	(A31)	Vcc	(Z29)	—	Gnd
3	D29	Gnd	Gnd	3	(C30)	(A30)	Gnd	Line\	(D30)	—
4	D28	Clk\	AVec\	4	(C29)	(A29)	Gnd	DBstIn	(D29)	DBstIn
5	D27	Reset\	RMC\	5	(C28)	(A28)	Gnd	(Z27)	(D28)	—
6	D26	Vcc	Vcc	6	(C27)	(A27)	Gnd	(Z25)	(D27)	—
7	D25	DS\	IAck\	7	(C26)	(A26)	+12VF	DBSnRq\\	(D26)	—
8	D24	Gnd	Gnd	8	(C25)	(A25)	-12VF	Gnd	(D25)	Line\
9	D23	DSAck0\	BRq\	9	(C24)	(A24)	A10	(Z23)	(D24)	DBSnRq\
10	D22	Vcc	Vcc	10	(C23)	(A23)	A11	(Z21)	(D23)	—
11	D21	Siz1	DBdEn\	11	(C22)	(A22)	A12	AColl-	(D22)	
12	D20	Gnd	Gnd	12	(C21)	(A21)	A13	AColl+	(D21)	
13	D19	AS\	MWDBd\	13	(C20)	(A20)	A14	(Z19)	(D20)	
14	D18	Gnd	DBCEn\	14	(C19)	(A19)	A15	(Z17)	(D19)	
15	D17	Siz0	MWVME\	15	(C18)	(A18)	A16	—	(D18)	
16	D16	Vcc	Clash\	16	(C17)	(A17)	A17	Gnd	(D17)	
17	D15	DSAck1\	B-DBdSel\	17	(C16)	(A16)	A18	(Z15)	(D16)	
18	D14	Gnd	B-Int\	18	(C15)	(A15)	A19	(Z13)	(D15)	
19	D13	DBdSel\	B-MWDBd\	19	(C14)	(A14)	A20	ERxD-	(D14)	
20	D12	Gnd	B-IAck\	20	(C13)	(A13)	A21	ERxD+	(D13)	
21	D11	Int\	Gnd	21	(C12)	(A12)	A22	(Z11)	(D12)	
22	D10	Gnd	B-DBdEn\	22	(C11)	(A11)	A23	(Z9)	(D11)	
23	D9	A0	B-BRq\	23	(C10)	(A10)	A24	—	(D10)	
24	D8	A1	B-IDEn\	24	(C9)	(A9)	A25	Gnd	(D9)	
25	D7	A2	IDEn\	25	(C8)	(A8)	A26	(Z7)	(D8)	
26	D6	A3	Freeze\	26	(C7)	(A7)	A27	(Z5)	(D7)	
27	D5	A4	NoBak\	27	(C6)	(A6)	A28	ETxD-	(D6)	
28	D4	A5	A32\ - BstAck\	28	(C5)	(A5)	A29	ETxD+	(D5)	
29	D3	A6	FC0	29	(C4)	(A4)	A30	(Z3)	(D4)	
30	D2	A7	FC1	30	(C3)	(A3)	A31	(Z1)	(D3)	
31	D1	A8	FC2	31	(C2)	(A2)	Gnd	—	(D2)	
32	D0	A9	TAck\	32	(C1)	(A1)	Gnd	Gnd	(D1)	

Notes:
1. Pins in this row connect to the VMEbus P2 connector pin indicated in parentheses. Space is provided in these columns to write in the assigned signals, if desired. Refer to the applicable EZ-bus module User Guide for P4 pin assignments. Note that Module B's (top board in stack) P4 pin assignments apply to the P4 connector on the EZP2 adapter.
2. This row present only with optional wide (160-pin) VMEbus P1 & P2 connectors.
3. PD4 row present only with standard 96-pin VMEbus P1 & P2 connectors.

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



Memory module connectors (P9, P10 & P11)

The P9, P10, and P11 connectors provide an interface between the main V452 Series board and the detachable memory module boards.



Synergy offers memory module boards containing 4, 8, 16, 32, 64, 128, 256* or 512* MB of DRAM. For more information about memory modules, see the ***Dynamic RAM*** chapter in Section 4.

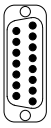
* Special order item.

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

Memory module connector, P9 pinouts

Pin	Function	Pin	Function
1	—	**33**	D10
2	—	**34**	D11
3	—	**35**	Vcc
4	—	**36**	Vcc
5	Gnd	**37**	D12
6	Gnd	**38**	D13
7	Byte0\	**39**	D14
8	Byte1\	**40**	D15
9	Byte2\	**41**	Gnd
10	Byte3\	**42**	Gnd
11	Vcc	**43**	D16
12	Vcc	**44**	D17
13	DPar0	**45**	D18
14	DPar1	**46**	D19
15	DPar2	**47**	Vdd
16	DPar3	**48**	Vdd
17	Gnd	**49**	D20
18	Gnd	**50**	D21
19	D0	**51**	D22
20	D1	**52**	D23
21	D2	**53**	Gnd
22	D3	**54**	Gnd
23	Vdd	**55**	D24
24	Vdd	**56**	D25
25	D4	**57**	D26
26	D5	**58**	D27
27	D6	**59**	Vcc
28	D7	**60**	Vcc
29	Gnd	**61**	D28
30	Gnd	**62**	D29
31	D8	**63**	D30
32	D9	**64**	D31

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



Memory module connector, P10 pinouts

Pin	Function	Pin	Function
1	—	**33**	A23
2	—	**34**	A22
3	—	**35**	Vcc
4	—	**36**	Vcc
5	Gnd	**37**	MClkD
6	Gnd	**38**	MClk
7	MA0	**39**	WShort\
8	MA1	**40**	RdDon\
9	MA2	**41**	Gnd
10	MA3	**42**	Gnd
11	Vcc	**43**	CasEn\
12	Vcc	**44**	CasEnD\
13	MA4	**45**	BLTCas\
14	MA5	**46**	BLTCasD\
15	MA6	**47**	Vdd
16	MA7	**48**	Vdd
17	Gnd	**49**	IncHld\
18	Gnd	**50**	Mux\
19	MA8	**51**	RasSS\
20	MA9	**52**	RasSX\
21	A31	**53**	Gnd
22	A30	**54**	Gnd
23	Vdd	**55**	BstHd\
24	Vdd	**56**	Rfsh
25	A29	**57**	RasEn\
26	A28	**58**	Ras\
27	A27	**59**	Vcc
28	A26	**60**	Vcc
29	Gnd	**61**	AS\
30	Gnd	**62**	WriteS\
31	A25	**63**	PreChg\
32	A24	**64**	—

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

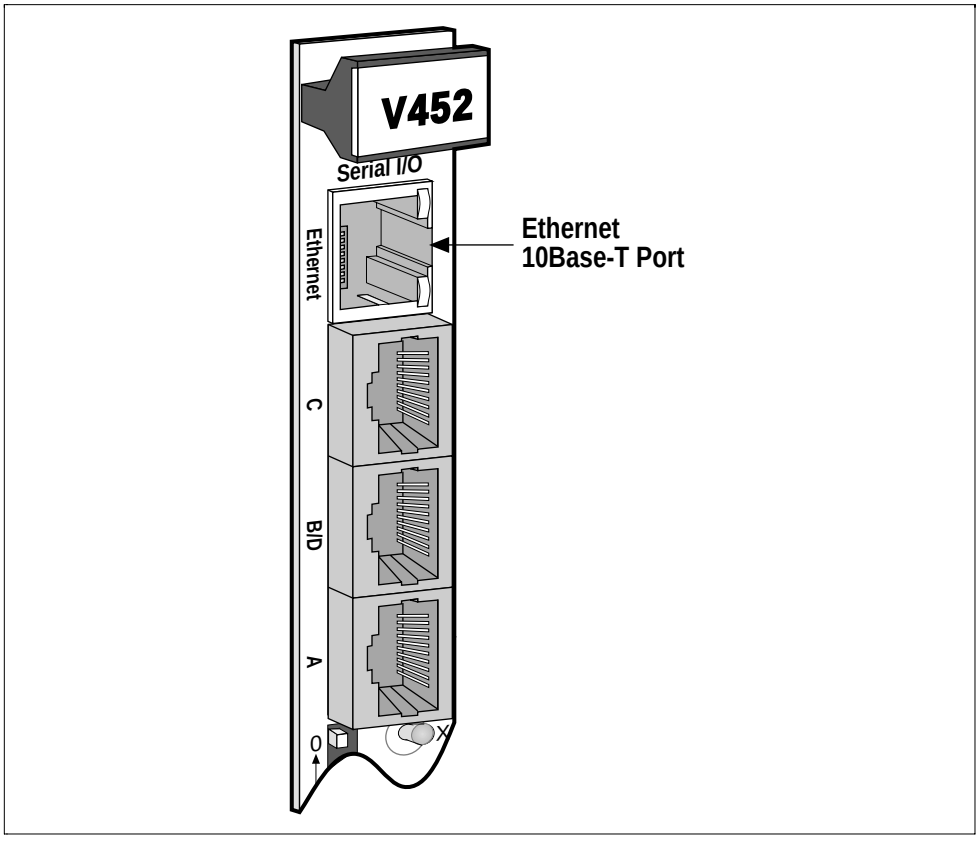
Memory module connector, P11 pinouts

Pin	Function	Pin	Function
1	Gnd	**33**	Gnd
2	Gnd	**34**	Gnd
3	Gnd	**35**	Gnd
4	Gnd	**36**	Gnd
5	Gnd	**37**	MSiz0
6	Gnd	**38**	MSiz1
7	Gnd	**39**	MSiz2
8	Gnd	**40**	MSiz3
9	Gnd	**41**	Vdd
10	Gnd	**42**	Vdd
11	Gnd	**43**	Vdd
12	Gnd	**44**	Vdd
13	—	**45**	Vdd
14	—	**46**	Vdd
15	—	**47**	Vdd
16	—	**48**	Vdd
17	Vcc	**49**	DSiz0
18	Vcc	**50**	DSiz1
19	Vcc	**51**	DSiz2
20	Vcc	**52**	DSiz3
21	Vcc	**53**	Gnd
22	Vcc	**54**	Gnd
23	Vcc	**55**	Gnd
24	Vcc	**56**	Gnd
25	Pres1	**57**	Gnd
26	Pres2	**58**	Gnd
27	Pres3	**59**	Gnd
28	Pres4	**60**	Gnd
29	Gnd	**61**	DSiz4
30	Gnd	**62**	DSiz5
31	Gnd	**63**	—
32	Gnd	**64**	—

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

Ethernet 10Base-T connector (P8)

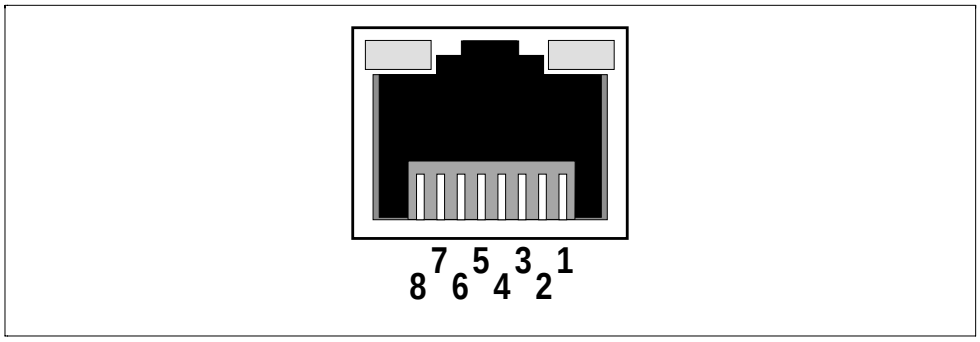
The V452 Series board's optional Ethernet 10Base-T port connects to the Ethernet 10Base-T network via the front panel jack as shown in the figure below. This chapter lists the pinouts for this connector.



Ethernet 10Base-T port

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

The figure and table below identify the pinout numbers and signals for the Ethernet 10Base-T connector on the V452 front panel.



Ethernet 10Base-T connector pin numbering

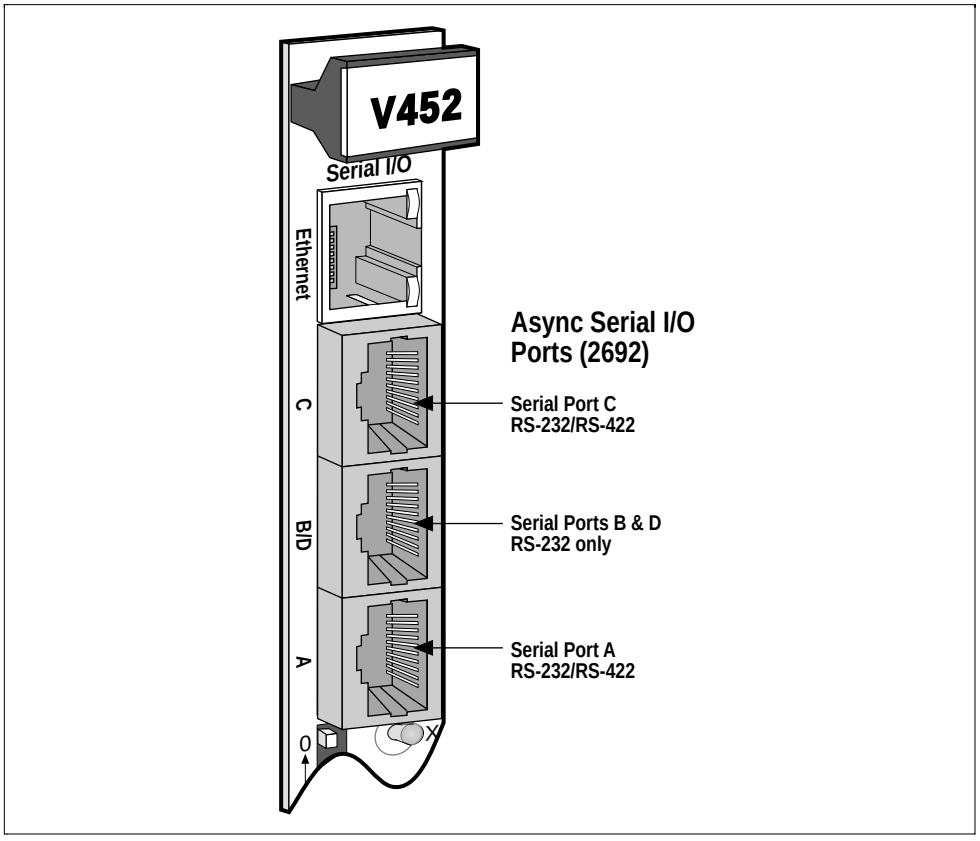
Ethernet 10Base-T port (P8) pin assignments

Pin	Function
1	**Transmit Data+**
2	**Transmit Data–**
3	**Receive Data+**
4	no connection
5	no connection
6	**Receive Data–**
7	no connection
8	no connection

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

Asynchronous serial connectors (P5–P7)

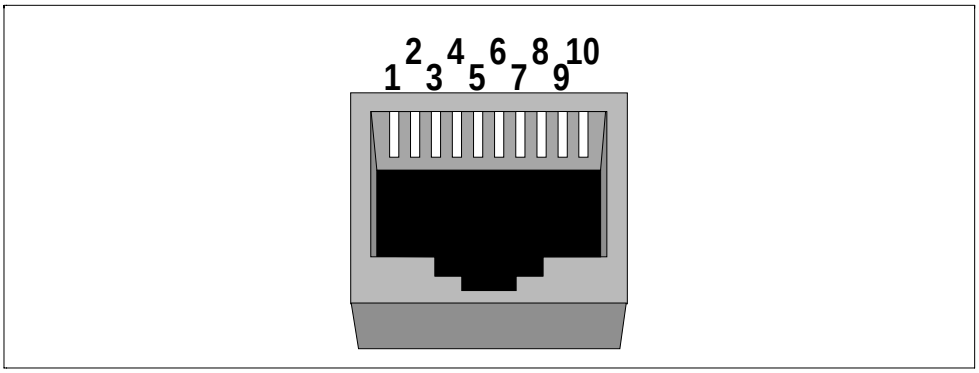
The V452 Series board's four asynchronous serial ports connect to external devices via 10-pin modular connectors as shown in the figure below. This chapter lists the pinouts for these connectors.



Asynchronous serial I/O ports

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

The figure and table identify the pinout numbers and signals for the Serial Port RJ-45 connectors on the V452 front panel.



Asynchronous serial connector pin numbering

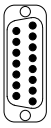
Serial Ports A & C (P5, P7) pin assignments

Pin	Function
1	**Request to Send** – (RTS+) differential RTS output signal for RS-422A communications.
2	**Request to Send** – (RTS) output indicating the modem is permitted to transmit data.
3	**Transmit Data** – (Tx+) differential transmit output signal for RS-422A communications.
4	**Transmit Data** – (Tx) output containing the data stream from the 2692.
5	**Ground** – (Gnd) internally connected to ground; connect to Signal Ground.
6	**Ground** – (Gnd) internally connected to ground; connect to Signal Ground.
7	**Receive Data** – (Rx) input containing the data stream from the modem.
8	**Receive Data** – (Rx+) differential Receive input signal for RS-422A communications.
9	**Clear to Send** – (CTS) input indicating the modem is ready to send data.
10	**Clear to Send** – (CTS+) differential CTS input signal for RS-422A communications.

Serial Ports B & D (P6) pin assignments

Pin	Function
1	**Port D, Request to Send** – (RTS) output indicating the modem is permitted to transmit data.
2	**Port B, Request to Send** – (RTS) output indicating the modem is permitted to transmit data.
3	**Port D, Transmit Data** – (Tx) output containing the data stream from the 2692.
4	**Port B, Transmit Data** – (Tx) output containing the data stream from the 2692.
5	**Ground** – (Gnd) internally connected to ground; connect to Signal Ground.
6	**Ground** – (Gnd) internally connected to ground; connect to Signal Ground.
7	**Port B, Receive Data** – (Rx) input containing the data stream from the modem.
8	**Port D, Receive Data** – (Rx) input containing the data stream from the modem.
9	**Port B, Clear to Send** – (CTS) input indicating the modem is ready to send data.
10	**Port D, Clear to Send** – (CTS) input indicating the modem is ready to send data.

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



Serial I/O cabling options

As described in the last chapter, the V452 provides a front panel RJ-50/RJ-69 modular jack for connection of the four serial channels.

A single modular cable in conjunction with a readily-available Modular-to-D connector adapter connects either Serial Port A or Serial Port C, to RS-232 or RS-422/RS-449 terminals and communications devices with a DB-9, DB-25 or DB-37 serial connector. In this scheme a standard modular cable is used to connect the V452 board's modular connector to the adapter and then the adapter routes the signal to the appropriate pin on the D-type connector. Serial Ports B and D provide only RS-232. A single cable/adapter can be connected to the B/D port to use only Serial Port B.

The Modular-to-D connector adapters typically consists of an empty D connector shell with an integral modular jack that has pigtails with D pins crimped to them. The adapter is constructed by plugging the appropriate pigtail pins into the correct D shell holes and then assembling the adapter shell. The proper adapter wiring needed for various D-type connectors and communications protocols are listed in a table appearing later in this chapter.

To use both Serial Port B and Serial Port D, a special cable adapter is required. Two types of serial cable adapters are available from Synergy:

- RJ-50/RJ-69 to male DB-25/RJ-45 jack adapter
- RJ-50/RJ-69 to dual RJ-45 jack adapter

A Modular-to-D connector adapter described above can be used to connect the RJ-45 jack(s) to the RS-232 device(s) as required.

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

Using only Serial Port A, B, and/or C

Modular cable size

The front panel serial I/O modular jack accepts modular plugs of 4, 6 (Ports A and C only), 8 or 10 pins. The smaller plugs will center themselves in the connector automatically as they are plugged in. (Beware not to use an offset 8-pin plug made for Digital Equipment Corp. computers.)

Various modular cables with the appropriate modular-to-D adapter will provide connection to Serial Port A, B or C as shown in the table below.

Modular cable types vs. serial port configuration

Cable size	RJ Type	Serial Port Configuration
4-pin cable	RJ-11	Serial ports A,, B, C: RS-232 Tx/Rx data, no handshake
6-pin cable	RJ-14	Serial ports A, C: RS-422 Tx/Rx data, no handshake
8-pin cable	RJ-45	Serial ports A, B, C: RS-232 Tx/Rx data, with handshake (RTS, CTS)
10-pin cable	RJ-50/RJ69	Serial ports A, B, C: RS-232 Tx/Rx data, with handshake (RTS, CTS) Serial ports A, C: RS-422 Tx/Rx data, with handshake (RTS/RTS-, CTS/CTS-)

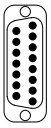


For maximum flexibility, it is possible, and perhaps even advisable, to construct all cables as the 10-pin variety. On a 10-pin cable the V452 Series boards drives all available signals.

When the other end of the cable is connected to another V452 Series board, RS-422, rather than RS-232, is automatically used.

When the other end of the cable needs to be connected to a device with a DB-9 or DB-25 connector, proper connections for the desired protocol can be wired into the Modular-to-D adapter described below.

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



Building modular cables

This interface system is designed to work with commonly available standard AT&T telephone cords. Four conductor modular cables, commonly used for internal telephone wiring, are available at most hardware and electronics stores. Cables can also be assembled to any length using readily available materials.

When making modular cable, locate the raised rib on one side of the flat cable wire. Insert the cable wire into the modular plug on both ends of the cable with the rib facing the same way. This construction causes the cable to have the necessary half-twist as specified in the AT&T standard.



It is possible to use a plug with more pins than the cable. Just make sure that the connectors are inserted in the center holes in the plug, rather than offset to one side.

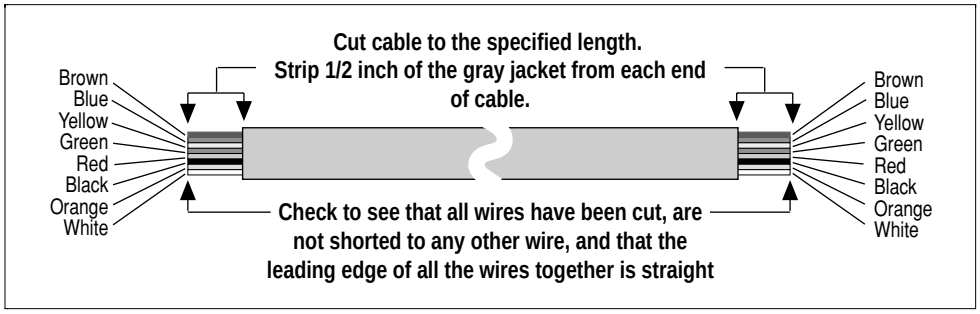


Unfortunately, several wiring color schemes are currently in wide use for modular cable and connector wiring. If you have a cable or connector that uses a different color scheme than the one depicted in this procedure, use the diagrams as locations guides only and make the necessary adjustments in the color assignments.

The paragraphs and diagram below describe how to construct an 8-conductor modular cable using cable stock that adheres to the USOC color scheme.

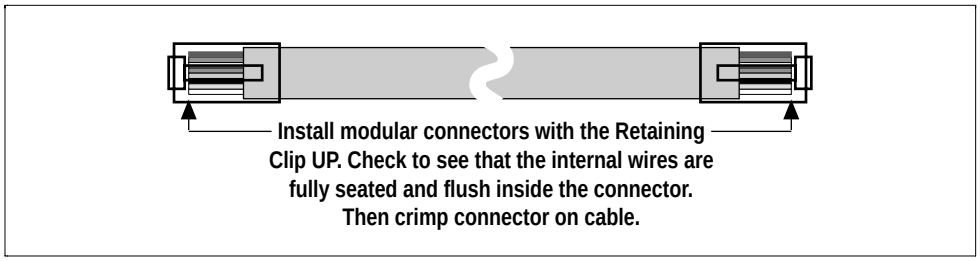
❶ **Cut modular cable to required length and strip ends** — Cut cable stock to the required length. Strip off 1/2 inch of the gray outer jacket from both ends of each cable using an 8-position Modular Cable production tool (AMP Part# 1-231652-1 or equivalent). Lay the cable on the work surface so that the rib on gray jacket is face down and the wire color scheme (USOC) on each end of the cable corresponds with that shown in the diagram below:

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



Strip modular cable

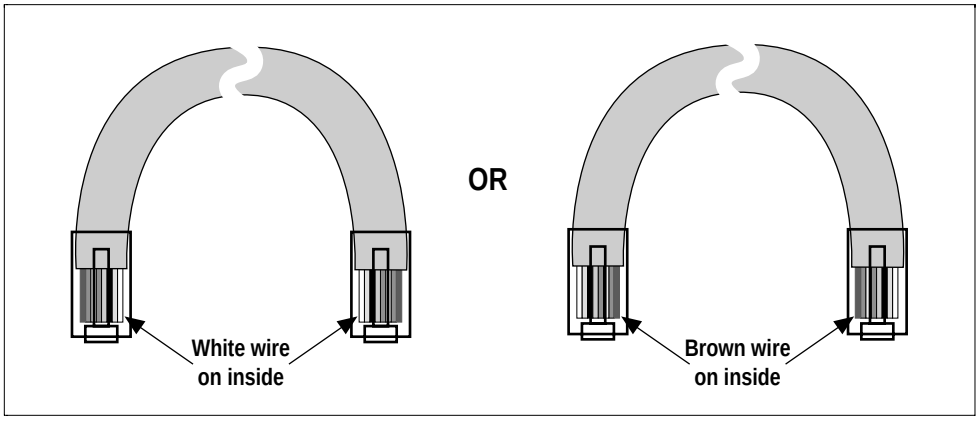
❷ **Install Modular connector** — Install an 8-pin modular cable on the end of cable so that the release tab on the connector is UP (i.e., on the opposite side from the ribbed side of the cable). Check to see that all eight wires are fully seated in the modular connector. Crimp on the 8-pin modular connector to both sides of the cable using the modular cable production tool (AMP Part# 1-231652-1 or equivalent).



Install modular connector

❸ **Verify connections** — To do a quick visual check of the finished cable, hold the two ends of the cable together with the connectors pointing in the same direction with the retaining clip on the modular connector facing up. When viewed in this arrangement, either the White or Brown wire should next to each other on each connector as shown in the diagrams below:

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



Verify connections

Wiring Modular-to-D adapters

Modular-to-D adapters, which route signals from a modular cable to EIA standard DB-9 or DB-25 connector, are available from various electronic supply stores. The two tables on the facing page list the appropriate pinouts for constructing adapters for various communication protocols to act as both DCE and DTE devices:

- The first four columns in each table refer to the number of pins in the modular jack in the adapter to which a wire from a cable is attached.

  Note: When plugging a 4-, or 8-pin plug into the V452 board's 10-pin modular connector, there are two sizes of 4-pin plugs; the telephone handset style will not work. Use the telephone line cord style of 4-pin plug. It is the same width as the 6-pin plug.

- The center column in each table, labeled **Signal name,** is the signal names and abbreviations for the signals on the V452 serial connector pinout.

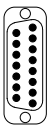
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- The remaining columns in each table list the D-connector pins for four different popular types of standard serial interfaces:
  - The **RS-423/RS-449 DB-9** is a seldom-used standard for 9-pin connectors.
  - The **IBM PC DB-9** lists the pinout for serial ports found on typical IBM PC-compatible computers.
  - The **RS-232 DB-25** column in each table lists the pinout needed to connect to the standard type of connector found on video terminals, modems, etc.
  - The **RS-422/RS-449 DB-37** is a differential RS-422 interface, using a DB-37 connector. It is commonly used on high-speed modems and other high-data-rate devices.

10-pin Modular-to-D null-modem adapter pinout (for connection to a DTE device)

Modular connector (female)				Signal name	D-type connector (male or fem.)			
4-pin con.	6-pin con.	8-pin con.	10-pin con.		DB-9 RS-423/ RS-449	DB-9 IBM PC RS-232	DB-25 RS-232	DB-37 RS-422/ RS-449
—	—	—	1	Request-to-Send + (RTS+)	—	—	—	25
—	—	1	2	Request-to-Send (RTS)	7	7	4	7
—	1	2	3	Transmit Data + (Tx+)	—	—	—	22
1	2	3	4	Transmit Data (Tx)	3	3	2	4
2	3	4	5	Ground (Gnd)	5	—	—	19
3	4	5	6	Ground (Gnd)	9	5	7	20
4	5	6	7	Receive Data (Rx)	4	2	3	6
—	6	7	8	Receive Data + (Rx+)	—	—	—	24
—	—	8	9	Clear-to-Send (CTS)	8	8	5	9
—	—	—	10	Clear-to-Send + (CTS+)	—	—	—	27

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



10-pin Modular-to-D modem adapter pinout (for connection to a DCE device)

Modular connector (female)				Signal name	D-type connector (male or fem.)			
4-pin con.	6-pin con.	8-pin con.	10-pin con.		DB-9 RS-423/ RS-449	DB-9 IBM PC RS-232	DB-25 RS-232	DB-37 RS-422/ RS-449
—	—	—	1	Request-to-Send + (RTS+)	—	—	—	27
—	—	1	2	Request-to-Send (RTS)	8	8	5	9
—	1	2	3	Transmit Data + (Tx+)	—	—	—	24
1	2	3	4	Transmit Data (Tx)	4	2	3	6
2	3	4	5	Ground (Gnd)	5	—	—	19
3	4	5	6	Ground (Gnd)	6	5	7	37
4	5	6	7	Receive Data (Rx)	3	3	2	4
—	6	7	8	Receive Data + (Rx+)	—	—	—	22
—	—	8	9	Clear-to-Send (CTS)	7	7	4	7
—	—	—	10	Clear-to-Send + (CTS+)	—	—	—	25



Many serial DB-25 devices use non-standard pin assignments. If the wiring for the RS-232/DB-25 adapter listed above does not work, try shorting pin 6 (DTR) to pin 20 (DCD) on the DB-25. If still not successful, consult the device manual.

To use these tables, first decide whether you need a DTE or DCE adapter and then locate the necessary interface type in the appropriate table.



Remember that the DCE/DTE designation applies only to the adapter; the device plugged into the adapter must be of the **opposite** type.

Next, locate the size of the adapter in the first four columns of the table. For example, for a 4-pin adapter, use the column headed by **4-pin con.** The pin numbers in this column are pin numbers on the adapter's modular jack, attached to the pigtails.

This modular jack's pin 1 is the leftmost pin with the pins at the bottom of the mating hole and the mating side away from the view; i.e. looking at the pigtail end of the connector with the pigtails near the bottom.

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

Ignore the staggering of the pigtails; pin numbers increase from left to right.

Plug each pigtail listed in the proper “adapter pin” column into the numbered hole in the DB-9/DB-25 connector shell indicated in the adapter type. For example, to make a 4-pin RS-232, DB-25, DCE adapter:

1. Plug pigtail 4 into shell hole 3 (Tx Data).
2. Plug pigtail 1 into shell hole 2 (Rx Data).
3. Plug pigtail 3 into shell hole 7 (Gnd).
4. Insulate and leave unplugged pigtail 2 (unused).
5. Assemble connector shell.

Any pigtails from the modular jack that are not used should be insulated with a 1" long piece of heat-shrink tubing or “spaghetti,” or may be cut off flush with the adapter jack after the adapter is tested.

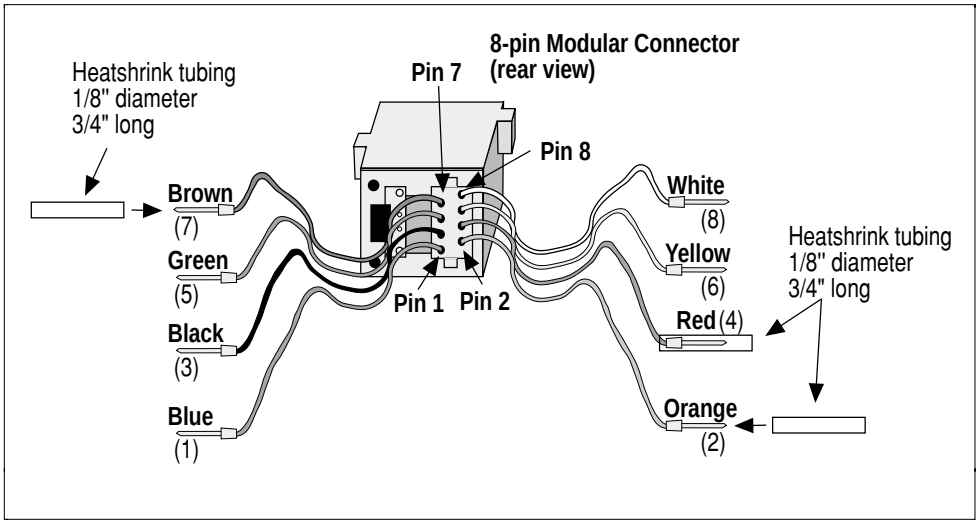
This procedure uses an EIA/Modular adapter kit containing a trapezoidal-shaped connector shell with a molded-in, pre-wired modular connector. The wires are typically color coded using the Nevada Western wiring color scheme. A DB-25 connector completes the kit. The flange of this connector snaps into the wider, open end of the housing after wiring. This kit is available from various electronic/computer supply houses.



Unfortunately, several wiring color schemes are currently in wide use for modular cable and connector wiring. If you have a cable or connector that uses a different color scheme than the one depicted in this procedure, use the diagrams as locations guides only and make the necessary adjustments in the color assignments.

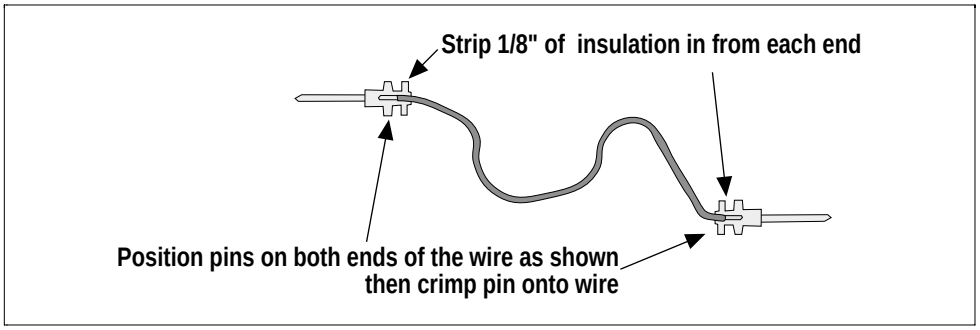
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

❶ **Install shrink-wrap insulation on unused pins** — Cut three lengths of 1/8" diameter heat shrink tubing approximately 3/4" long. Pull out the modular connector/wiring harness from the adapter shell. Slide the tubing over the metal end of the Brown (pin 7), Red (pin 4), and Orange (pin 2) leads so that all metal surfaces are covered as shown in the diagram below. Apply heat to shrink the tubing so that it fits snugly on the lead.



Insulating unused connector pins

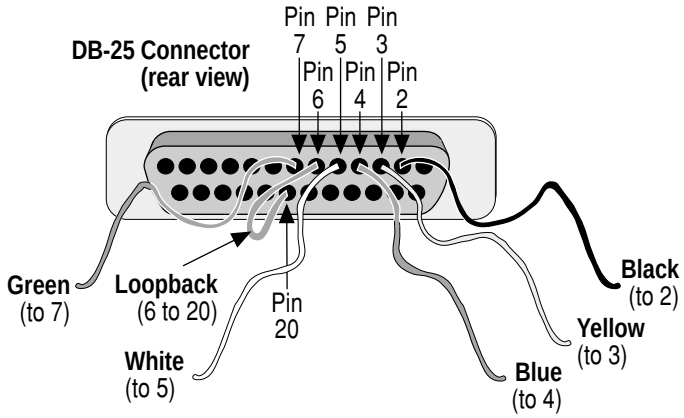
❷ **Construct a DTR-to-DCD loop back lead** — Cut a 1 and 3/4 inch section of the 8-conductor modular cable and remove the gray outer jacket. Strip approximately 1/8 inch of insulation from each side of one of the inner eight wires. Set aside the remaining wires for use in construction of another adapter. Crimp on a pin (AMP Part# 1-66506-0) onto each side of the wire using wire crimping tool (AMP Part# 90312-1 or equivalent):



DTR-DCD loop back lead

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

❸ **Install pins in DB-25** — Find the DB-25 connector in the Adapter Kit. Insert the leads from the modular connector and the loop back wire constructed in Step 2 as shown in the diagram and listed in the table below. Use a pin insertion tool (AMP Part# 91067-2 or equivalent) to fully seat the pins in the DB-25 connector.



Interconnection list

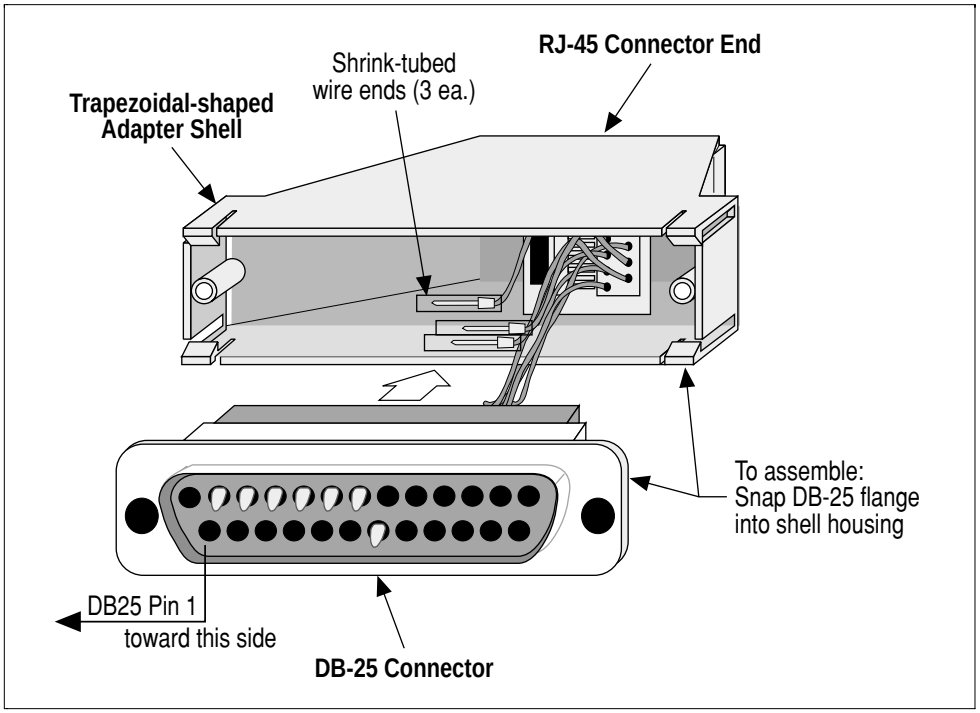
RJ45 pin	Wire Color*	DB-25 pin
1	Blue	4
2	Orange	NC - Heatshrink on wire
3	Black	2
4	Red	NC - Heatshrink on wire
5	Green	7
6	Yellow	3
7	Brown	NC - Heatshrink on wire
8	White	5
—	—	6 - Looped to DB25 pin 20
—	—	20 - Looped to DB25 pin 6

* Nevada Western Color code shown in table. Refer to RJ45 pin number if wiring on supplied modular jack uses a different color scheme.

DB-25 pin installation

❹ **Assemble adapter shell** — As shown in the drawing below, snap the DB-25 connector into the adapter shell housing. Press the DB-25 connector firmly into the housing so that all 4 housing snaps engage the connector flange.

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



Adapter shell assembly

Using dual serial port connector via Synergy serial cable adapters

The use of both V452 serial ports B and D requires special cable adapters that bring out the two serial ports from the single front panel RJ-50/RJ-69 serial I/O jack. Two types of adapters are available from Synergy (contact Customer Service for details on ordering these adapters):

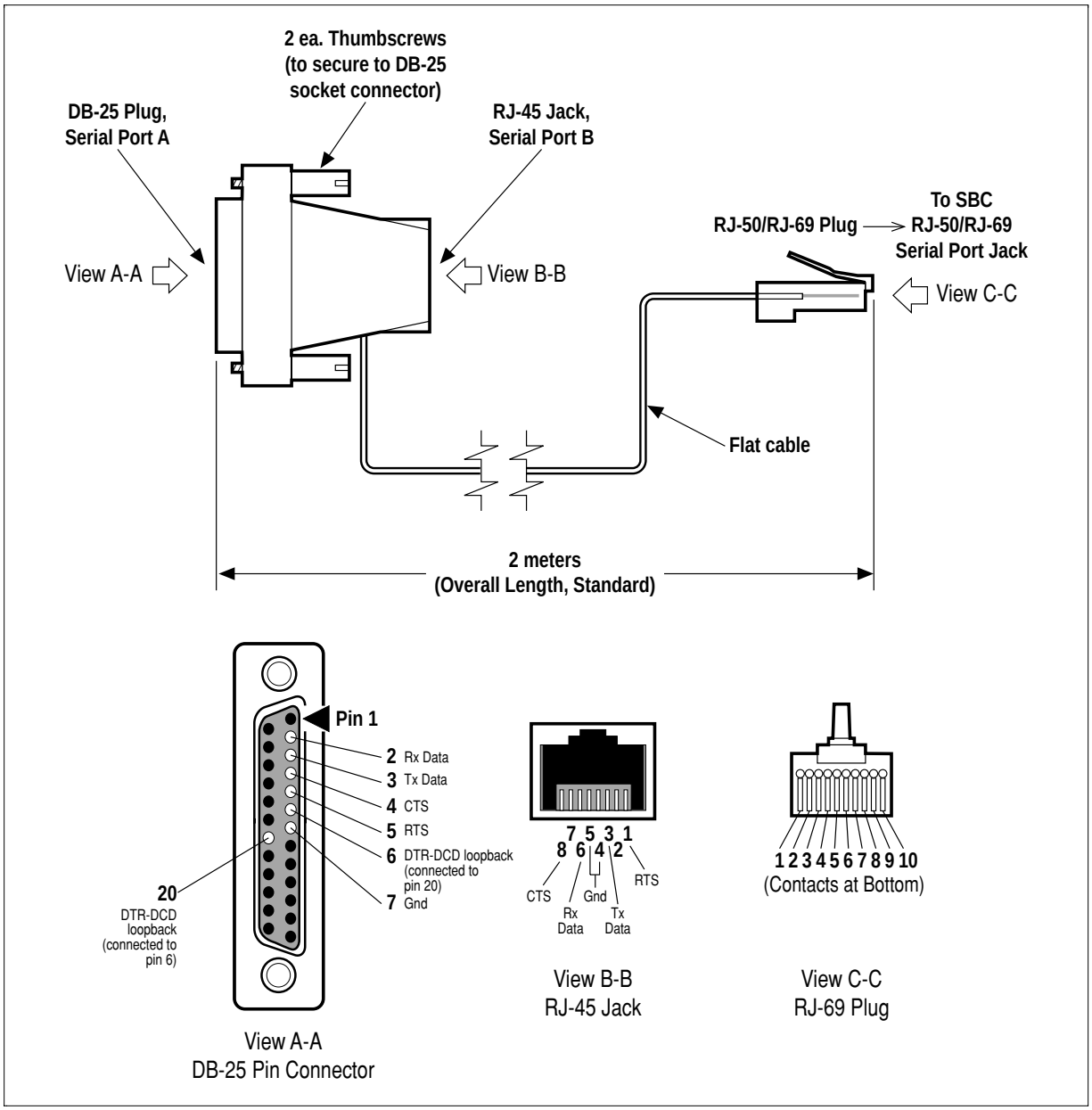
Part no. **Cbl/J10J8D25M2m** — RJ-50/RJ-69 plug to adapter head containing one DB-25 male connector for Serial Port A and RJ-45 connector jack for Serial Port B. This adapter uses a 2-meter (standard length) flat cable. A custom length can be ordered in which case the last two characters in the cable part number will reflect the custom length (e.g., "4m" for a 4-meter cable, Cbl/J10J8D25M4m).

Part no. **Cbl/J10J8J81m** — RJ-50/RJ-69 plug to adapter head containing two RJ-45 connectors, one for each serial port. This adapter uses a 1 meter (standard length) round cable. A custom length can be ordered in which case the last two characters in the cable part number will reflect the custom length (e.g., "3m" for a 3-meter cable, Cbl/J10J8J83m).

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

Modular-to-D cable adapters of the 4- or 8-pin variety as described in the ***Using only Serial Port A, B and/or C*** section can be used to connect a serial device to the serial cable adapter's RJ-45 jack(s).

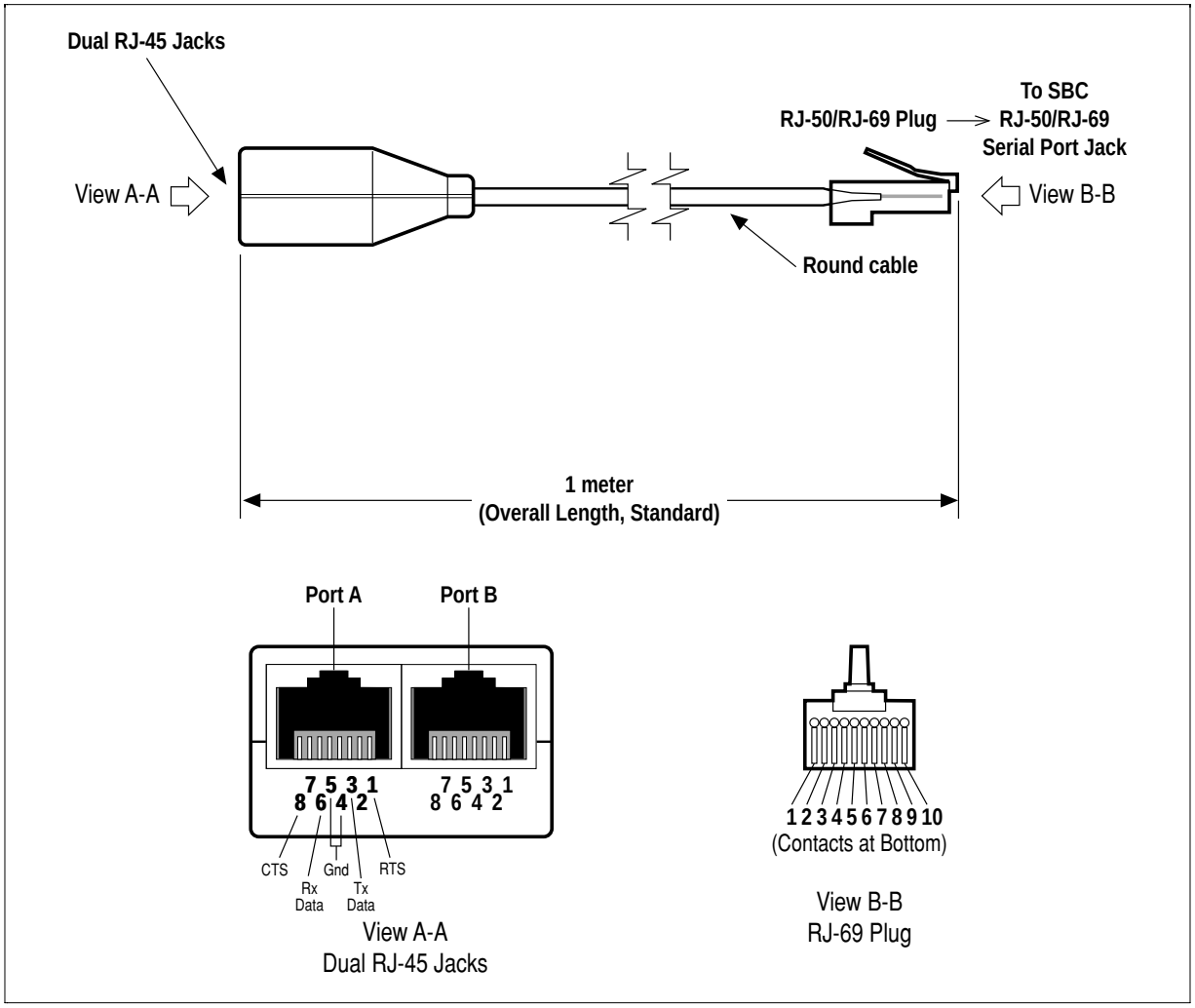
The drawing below shows the **Cbl/J10J8D25M2m** cable and pin assignments of the connectors on the peripheral end.



Cbl/J10J8D25M2m serial cable adapter

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

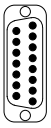
The drawing below shows the **Cbl/J10J8J81m** cable and pin assignments of the connectors on the peripheral end.



Cbl/J10J8J81m serial cable adapter

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

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



P2 serial interface option

V452 Series boards can be optionally configured to present TxD and RxD signals from asynchronous serial ports A and B to pins on the board's VMEbus P2 connector. This option is accomplished by soldering small wires on the solder side of the board between the **JL01** connector holes and the appropriate row holes for the **P2** connector.

> ⚠ The user-assignable pins in Row A of the VMEbus P2 connector on the V452 Series board are used by many of the optional EZ-bus daughter modules that are available for use with V452 Series boards. As a result, implementation of this modification may seriously limit which daughter boards can subsequently be used with the modified board. For more information, see the P2 pinout information for the prospective module in the user guide for that module or contact Synergy customer service.

Applying the modifications described in this chapter **do not disable** the modular connectors for serial ports A and B on the front panel of the board. Rather, this modification allows the port A and B serial I/O signals to be accessible from **EITHER the front panel OR the P2**. Attempting to use both of these access points at once could lead to conflicts and other related problems and is therefore **not recommended** for most applications.

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

The description of the modifications required to use this option are included in this manual for convenience only.

Customers wanting this optional configuration are **strongly urged** to have the necessary modifications **performed by Synergy** during the original manufacture of the board or later as a factory rework. For more information about having this work done, contact Synergy sales or customer service.

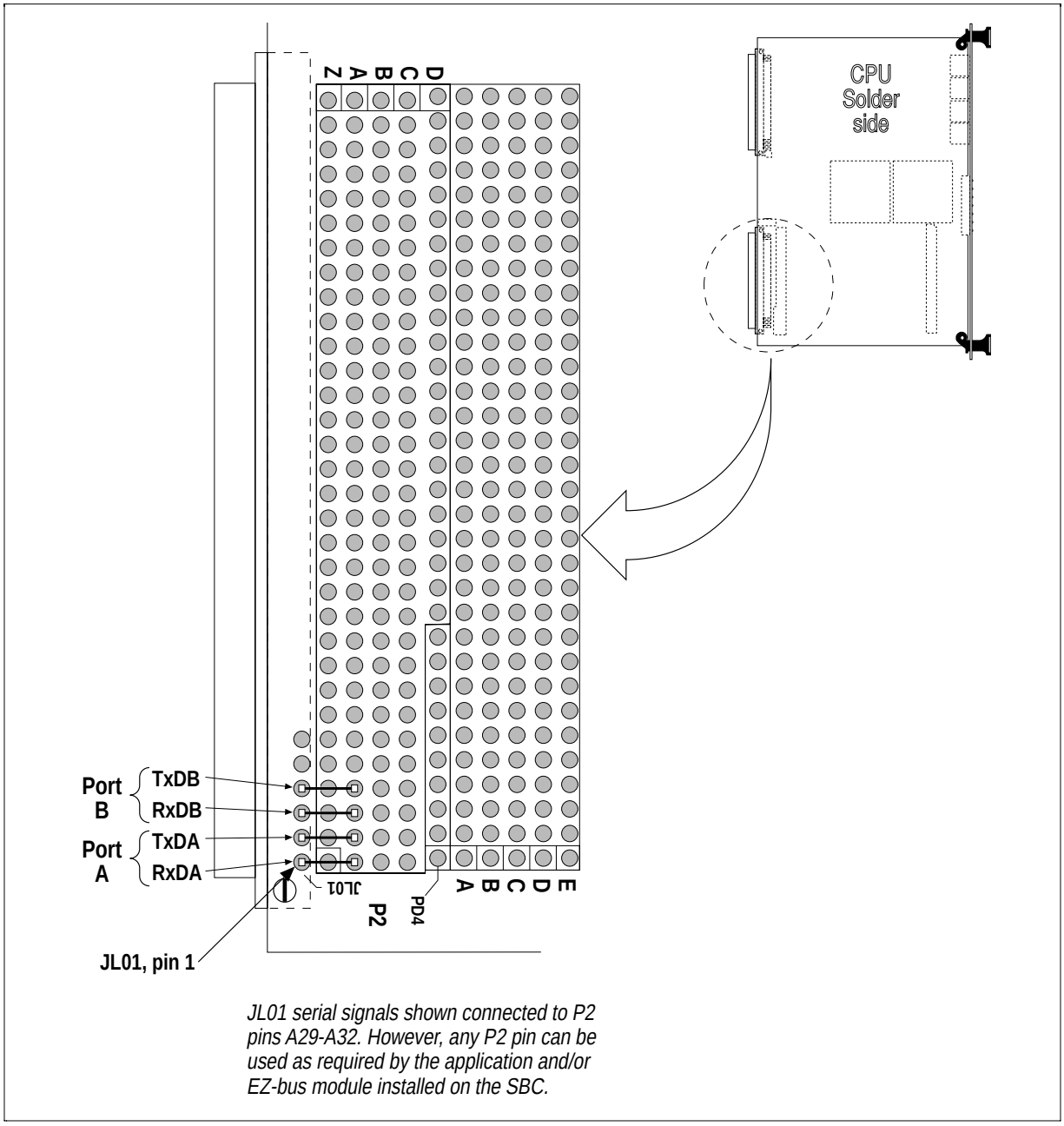
All modifications performed by customers themselves are not covered by Synergy's warranty for the board. In addition, if in the process of completing this or any other modification, other parts or areas of the board are damaged, Synergy reserves the right to refuse to perform repairs on these damaged components or board areas under the terms of the warranty as described in the next section. Required repairs in this case will be charged to the customer.

The steps below describe the procedure for performing the modification (refer to the diagram below for locations of pins and signals):

❶ Locate the **JL01** and **P2** connector holes on the solder side of the V452 Series board.

❷ The TxD and RxD signals from serial ports A and B are provided at JL01 pins 1–4. These signals can be wired to any P2 pin as required by the application. Alternatively, JL01 1–4 can be wired straight across to P2 pins A32–A29 respectively if there is no conflict with an EZ-bus module using these pins (P4 pins B1–B4). Connect the serial I/O signals to P2 using insulated 30 gauge wire. Be careful not to inadvertently desolder the P2 connector pins.

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



JL01 serial signals shown connected to P2 pins A29-A32. However, any P2 pin can be used as required by the application and/or EZ-bus module installed on the SBC.

P2 serial interface modification and pinout

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

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

Appendix B, Specifications

The V452 SBC conforms to the following set of specifications and standards.

VMEbus compliance

IEEE 1014 VMEbus Specification; Rev C.1 & D.1

Master:	A32,A24,A16/D32,D16,D08(EO):RMW RWD,ROR,FAIR:UAT, BLT32, BLT64.
Slave:	A32,A24:D32,D16,D08(EO):RMW:UAT, BLT32, BLT64.
Interrupter:	I (1-7):D08(O):ROAK.
Interrupt handler:	IH(1-7):D08(O).

Physical dimensions

The V452 printed circuit board conforms to **VME 6U** requirements for form factor, board spacing, and board thickness:

Board Size:	6U: 6.4"x 9.19"x 0.8" minus front panel
Board Thickness:	0.062 +/- 0.005 inches or 15.24 +/- 0.51 mm

Weight

V452: 16 ounces (454g)

Weight (approx.) for dual CPU version with 16 MB memory module and no EZ-bus daughter boards.

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

Power requirements

V452 typical power consumption (w/16MB RAM, no EZ-bus module installed):

V452 model (33 MHz, single '040 processor)
+5.0V ±5%, 3.38A typical @ 5.00V
-12.0V ±5% , 30 mA typical @ -12.00V

V452 model (33 MHz, dual '040 processor)
+5.0V ±5%, 4.10A typical @ 5.00V
-12.0V ±5% , 30 mA typical @ -12.00V

V452 model (50 MHz, single '060 processor)
+5.0V ±5%, 3.0A typical @ 5.00V
-12.0V ±5% , 30 mA typical @ -12.00V

V452 model (50 MHz, dual '060 processor,
+5.0V ±5%, 3.8A typical @ 5.00V
-12.0V ±5% , 30 mA typical @ -12.00V

Voltages must be kept within these tolerances to ensure proper operation.

Operating environment

Temperature: Operating: 0° to 50° C ambient with forced air cooling (minimum 200 LFM; recommended 400 LFM.)

Non-operating/Storage: -20° to +70° C

Board configurations that provide a wider operating temperature range are available. Contact Customer Service for a listing of Thermal Capability options.

Humidity: 10% to 90%, non-condensing

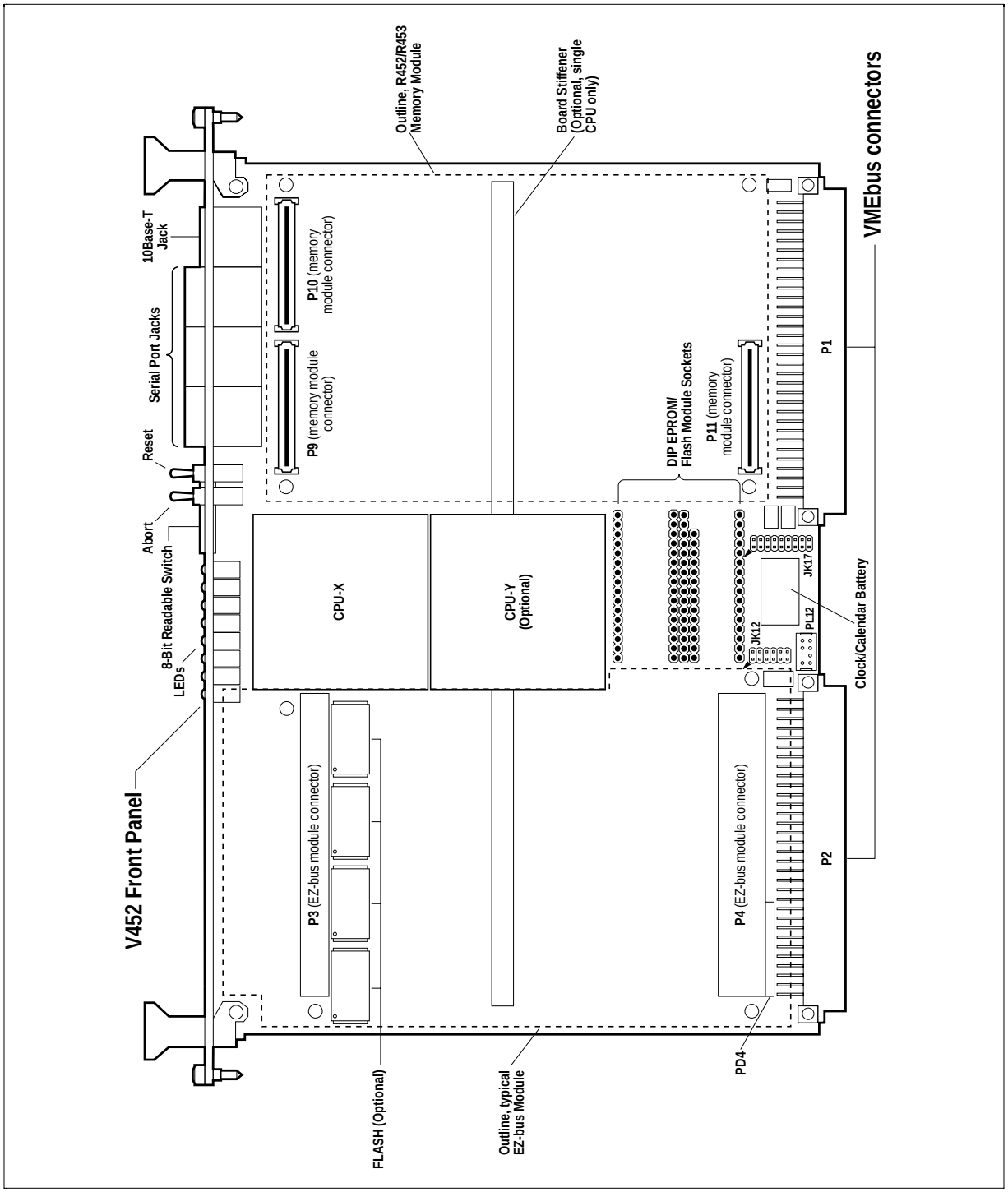
Number of slots

1

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

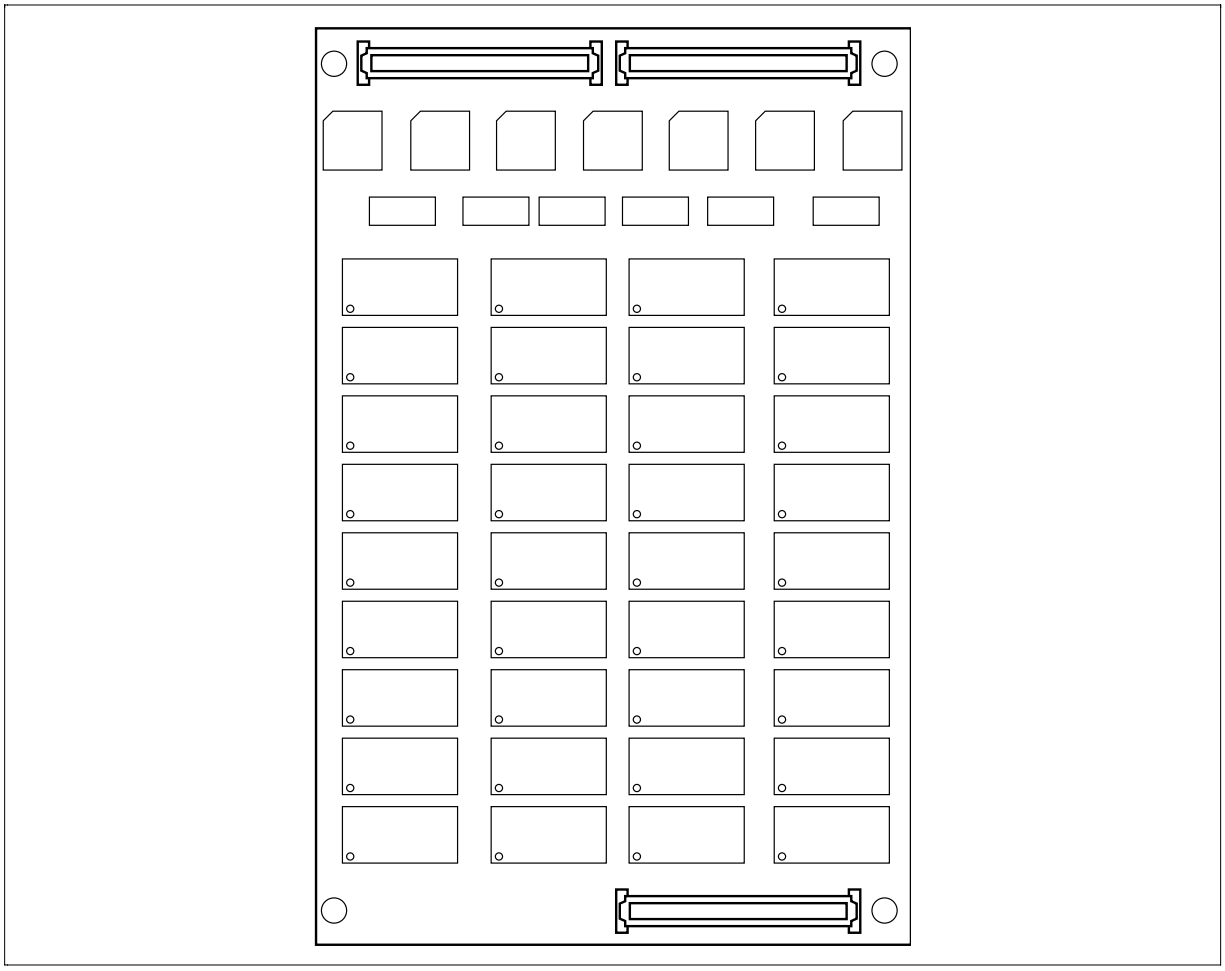
Board layout

See drawings below for V452 and memory module layout.



V452 board layout

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



R452 board layout

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

Appendix C, Board revision summary

This appendix summarizes major changes made to the V452 SBC and memory modules affecting form, fit, and/or function. The paragraphs below list the changes pertaining to the revision shown.



The revision levels for each feature represents the revision level when the listed feature was added to the standard design. Some boards with older revision levels may have had some of these features added during previous rework/upgrades.

Contact Synergy customer service for upgrade information.

V452

These paragraphs describe changes made to the V452 main board.

Revision B	Added slot ID register, improved clock signal integrity, various ECO cleanup.
Revision A	Initial board release.

R452 memory module

These paragraphs describe changes made to the R452 memory module.

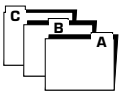
Revision B	Various ECO cleanup.
Revision A	Initial board release.

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

Appendix C

Board revision summary

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



Glossary

The paragraphs below define and describe some of the terms used in this manual. The definition entries observe the following conventions:

- Terms in definitions that appear (*in italics and in parentheses*) are related and/or alternative terms or acronym translations for the term being defined.
- Terms in definitions that appear in **boldface** in definitions are defined elsewhere in the glossary.

10Base-T — a type of **Ethernet** that uses unshielded twisted-pair (UTP) cable and modular RJ-45 connectors for **LAN** connections in a star configuration (i.e., each network node connects to a common hub). Data rate is the same as standard Ethernet: 10 Mbps.

2692 — a CMOS **DUART** chip manufactured by Signetics, similar to the 2681 containing two asynchronous serial communications ports and two counter/timers.

48T18 — a **clock-calendar** chip manufactured by SGS-Thomson. It includes 8KB of non-volatile RAM and a user replaceable battery.

A16/D16 — specifies the address and data bus sizes for a microprocessor bus such as VMEbus. This value specifies a 16-bit wide address bus and 16-bit wide data bus.

A16/D32 — specifies a 16-bit wide address bus and 32-bit wide data bus.

A24/D16 — specifies a 24-bit wide address bus and 16-bit wide data bus.

A24/D32 — specifies a 24-bit wide address bus and 32-bit wide data bus.

A32/D16 — specifies a 32-bit wide address bus and 16-bit wide data bus.

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

A32/D32 specifies a 32-bit wide address bus and 32-bit wide data bus.

ABORT an **nmi** interrupt (Level-7, autovectored) invoked via a front panel push button. Used to terminate run-away processes.

AM code bits *(Address Modifier)* code bits, AM0–AM5, used by the VMEbus to identify the size of address being expressed (A16, A24, A32, A40 or A64) and the type of transfer (Address-only, program, data, BLT32, BLT64 or IAck) being performed.

ATC (*Address Translation Cache*) used by the MMU to speed up the translation process.

banner a message displayed on a CRT when a debug monitor or operating system is starting.

base address the lowest address in range of addresses. Usually the lowest address of a memory window, or of a set of peripheral registers.

BCD (*Binary Coded Decimal)* a coding system in which four binary (1s and 0s) digits represent each digit in a decimal (0 through 9) value.

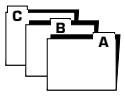
BLT (*Block Transfer*) a data transfer method for moving large amounts (blocks) of data. A BLT cycle is faster and more efficient than a regular R/W cycle because the address to start the transfer of multiple bytes is presented only once.

category 3 unshielded twisted-pair cable specification that functions at 10 Megabits per second on each pair.

category 5 unshielded twisted-pair cable specification that functions at 100 Megabits per second on each pair.

clock/calendar a device that records the progress of the time and date and makes this information available to programs running on the computer system.

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



CMOS (Complementary Metallic Oxide Semiconductor) a method of integrated circuit manufacturing that produces chips which require far less power and in some cases work faster than MOS or other devices produced using other manufacturing technologies.

copyback A caching method that leaves dirty (updated) data in the processor's data cache without writing it to main memory until the cache line containing the data is needed for new data.

CPU **(***Central Processing Unit)* central controlling device in a computer system.

CRT *(Cathode Ray Tube)* normally refers to a viewing screen; also used as a synonym for **terminal**.

cycle stealing a bus management technique that allows a peripheral device to temporarily disable CPU control of the bus to allow the device to directly access the CPU's local memory.

data broadcast (*data broadcasting*) a bus communications technique in which a single CPU board can send data to multiple CPU boards at the same time.

DB-9 a D-shaped serial interface connector for **I/O** cabling that provides access to up to 9 separate lines or pins on a matching connector.

DB-25 a D-shaped serial interface connector for **I/O** cabling that provides access to up to 25 separate lines or pins on a matching connector.

DB-37 a D-shaped serial interface connector for **I/O** cabling that provides access to up to 37 separate lines or pins on a matching connector.

DCE (*Data Communications Equipment*) the end of a serial communications link that is, or mimics, a modem (Opposite of DTE).

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

differential a method of signaling in which two wires are used, each carrying opposite versions of the signal information. This is done to increase maximum cable drive and to increase noise immunity. For example, a pair of signals are called SD0+ and SD0-. A 1 data bit may be represented by +5V on SD0+ and 0V on SD0-, and a 0 bit by 0V on SD0+ and +5V on SD0-. See **single-ended**.

dirty data Data in a cache that is newer than the data in main memory.

DMA (*Direct Memory Access*) a data transfer method in which data can pass between peripheral devices without intervention by the **CPU**.

DRAM *(Dynamic Random Access Memory)* high density fast access memory storage media that must be refreshed at continuous intervals. Also simply referred to as RAM

DTE (*Data Terminals Equipment*) the end of a serial communications link that is, or mimics, a CRT terminal or printer (Opposite of DCE).

DUART (Dual Universal Asynchronous Receiver/ Transmitter) see **UART.**

dual-ported a memory architecture which allows more than one access path to memory.

dynamic RAM see **DRAM**.

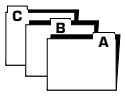
EZ-bus a Synergy semi-proprietary interface for connection of optional daughter modules to the main V452 Series motherboard.

ECO (*Engineering Change Order*) an engineering document that describes and orders a change to a released product.

EPROM (Erasable Programmable Read Only Memory) a special type of **PROM** whose programming can be erased by exposure to ultraviolet light and then reprogrammed.

Ethernet a high-speed (10 MB/sec) communications protocol and cable standard for computer networks.

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



FIFO (*First-In-First-Out*) a data storage technique in which the first item stored in memory is also the first item on the stack of items for retrieval. Also a piece of hardware that stores data in such a manner.

Flash memory a nonvolatile, random access, and rewritable solid-state storage technology that is ideal for field-upgradable code storage. Flash memory is electrically erased and programmed in-circuit.

floating-point method to represent numbers using the significant digits (mantissa) multiplied by the base of the number raised to the appropriate power (exponent). Values expressed in floating point form are similar in structure to number expressed in "scientific notation".

FPU *(Floating Point Unit)* a floating point co-processor.

GPS receiver a radio receiver that locks onto the GPS (Global Positioning System) satellites in orbit around the earth. Using a GPS receiver, you can pinpoint your exact location anywhere on earth and use the GPS satellite's onboard atomic clock as a time reference.

GPIB (*General Purpose Interface Bus*) a parallel cable bus usually used for controlling instruments.

IEEE P754 industry standard for floating point arithmetic.

IAck (*Interrupt Acknowledge*) a VMEbus signal used by a Master to indicate the act of acknowledging an interrupt.

IBM manufacturer of the Selectric typewriter and inventor of the 80-column punched card.

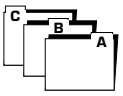
IMR (*Interrupt Mask register*) A register used to enable only certain interrupts.

interrupter a circuit that sources interrupts, usually at the behest of peripherals. An interrupter must drive an interrupt line and provide a vector number during an interrupt acknowledge cycle. It may cease driving the interrupt line upon the interrupt being acknowledged (ROAK) or wait until a register access to the peripheral explicitly removes the request (RORA).

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

interrupt handler	a circuit (usually in conjunction with a CPU) that acknowledges and handles interrupts.
I/O	(*Input/Output*)
ISP	*(In-System Programmable logic)* a high density programmable logic device that can be programmed while the device is in the circuit. ISP logic can be upgraded easily in the field using a standard PC and a simple adapter cable.
JEDEC	(*Joint Electronic Device Engineering Council*) a body that sets standards for chip packages and pinouts.
LED	*(Light Emitting Diode)* a diode that emits light when forward biased, commonly used for displays and indicators.
Level 1010 emulator	a group of opcodes currently unused in the 680X0 family, that may become real instructions in future versions to the CPU. Until that time the opcodes will produce a Trap to allow software to emulate the instruction. The 1010 refers to the first 4 bits of the opcode.
Level 1111 emulator	See Level 1010 emulator above.
mailbox	mechanism to allow any CPU or other Master to interrupt any other CPU of its choice.
Master	a device that initiates and controls the transfer of addresses and data across a bus. The opposite of Slave.
mem protect	*(Memory protect)* a bit that can be set or cleared in the Mode register. It usually is set to disable write accesses to the board and cleared to enable them, but its meaning can be changed via PALs or **ISP**s.
MMU	(*Memory Management Unit*) a circuit that provides address translation and access control services for a CPU.
MOS	**(***Metallic Oxide Semiconductor*) a method of integrated circuit manufacturing.
µs	(*microsecond*) one millionth (10^{-6}) of a second.

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



ms	(*millisecond*) one thousandth (10^{-3}) of a second.
multi-ported	a memory architecture in which the RAM can be accessed from several busses.
MSB	(*Most Significant Bit*)
NAN	(*Not A Number*) possible result of a floating point operation whose result is not defined as a number (eg., arcsine 3).
nmi	(*Non-Maskable Interrupt*) a level 7 interrupt. The 680x0 family of microprocessors treats level 7 interrupts as non-maskable.
ns	(*nanosecond*) one billionth (10^{-9})of a second.
NVRAM	*(Non-Volatile RAM)* RAM that retains its data even without external power.
object code	output from a compiler or assembler that is in machine language but still must be linked to other object code to form an executable program.
P1	the mandatory 96 pin VMEbus connector. It carries all the signals to allow transfers up to A24 and D16. On a 3U board it is the only connector.
P2	the secondary 96 pin VMEbus connector on 6U of 9U boards. 32 of is pins carry the signals necessary to allow A32 and D32 transfers. The other 64 pins are user definable.
page	the smallest unit of memory which is mapped by the **MMU**.
PAL	(*Programmable Array Logic*) integrated circuit containing an array of binary logic gates whose Boolean formula can be programmed or customized. Often used for configuration applications.
prescaler	a circuit that divides pulse train by a fixed number. The lower frequency output can then be input to a programmable divider that couldn't have handled the original input.

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

PROM (*Programmable Read-Only Memory*) a memory storage media that can be programmed using electrical pulses. Once programmed, the PROM is read-only but does not need power or refresh in order to maintain the stored data.

RAM (*Random Access Memory*) memory that is high-speed, randomly accessible (unlike a tape drive, which is sequential) and can be read and written to easily.

requester a circuit that requests Mastership of a bus.

read-modify-write see RMW.

RMA (*Return Merchandise Authorization*) a number assigned by Synergy for returning defective products.

RMW (*Read-Modify-Write*) a read memory access followed by a write access performed in such a way that no other access is allowed to the location between the read and write.

ROAK *(Release On AcKnowledge)* a type of VMEbus Interrupter module that deasserts its Interrupt Request to the VMEbus during reception of a valid IACK cycle for its interrupt level.

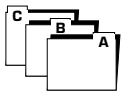
ROR (*Release On Request*) a requester strategy that once granted the bus asserts continued Mastership of the bus even if not currently needed, until another requestor requests the bus. Opposite of RWD.

RORA *(Release On Register Access)* a type of VMEbus Interrupter module that deasserts its Interrupt Request to the VMEbus during reception of a VME slave access cycle to one of its (vendor-specific) control registers.

round robin a bus sharing method that engages each device or process in a group at its turn in a fixed cycle.

RS-232 an industry standard for serial communications using ± 12V signals at up to 19.2 KB/sec for distances up to 50 ft.

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



RS-422 an industry standard serial communications using differential signals at up to 100 KB/sec for distances up to 1000 ft.

RWD (*Release When Done*) a requester strategy that once granted the bus asserts Mastership only as long as actually needed. Opposite of ROR.

SBC **(***Single Board Computer*) a printed circuit board containing microprocessor and support devices that provide **CPU**, ROM, RAM and peripheral interfaces.

SCSI *(Small Computer Systems Interface)* an industry standard parallel interface bus that provides host computers with device independence of add-on peripherals such as disk drives, tape drives, CD-ROM drives, etc. The standard began as an 8-bit parallel data interface with a max. transfer rate of 5 MB/S (SCSI-1). The next SCSI standard, SCSI-2, introduced fast/wide SCSI. SCSI-2 is commonly implemented with a 16-bit data bus with a max. transfer rate of 10 MB/S (async) and 20 MB/S (sync). The SCSI Trade Organization (STA) has categorized several other SCSI types such as Wide Ultra SCSI and Ultra2 SCSI. The T10 standards committee is working on the SCSI-3 standard which provides additional connector and cabling options, protocol extensions, and transmission schemes (high performance serial and fiber data channel).

SDRAM *(Synchronous Dynamic Random Access Memory)* a type of DRAM that operates in step with the CPU clock which allows the processor to perform more instructions over a given time.

single-ended a method of signaling in which one wire is used per signal, referenced to a common Ground signal. This is the most cost efficient signaling method for short cable runs. See **differential**.

Slave a device connected to a bus that responds to commands from a Master.

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

spurious interrupt an interrupt whose acknowledge cycle received no response. Usually caused by late acknowledgment of periodic interrupters such as timers. But may be caused by interrupt request that was aborted before being acknowledged.

SRAM *(Static Random Access Memory)* a memory storage media that needs no refresh cycle. SRAM is faster and of lower density than DRAM.

stale data a copy of data that is no longer valid because some processor or DMA activity has updated the data elsewhere.

supervisor *(supervisor mode)* a Motorola processor execution mode in which the CPU enjoys all its privileges.

***SysClk *** (*System Clock*) a signal driven by the system controller to all boards of a VMEbus system.

***SysFail *** (*System Failure*) a signal that can be driven by any board of a VMEbus system. Traditionally used to indicate a failure to one or more boards or devices on a bus.

***SysRes *** (*System Reset*) a signal driven by the system controller to reset all the cards on the VMEbus.

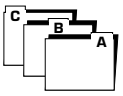
system controller on VMEbus, a group of circuits on the #1 slot VMEbus board that prioritize the bus-requests, provide a system clock, and provide system timeouts.

TrapV (*Trap on oVerflow*) a special TRAP instruction that directs the CPU to begin exception processing if the overflow bit of the CPU's status register is set. Usually used to see if the previous arithmetic operation overflowed the available bits of the CPU's registers.

triple-access (*triple-access **DRAM***) see multi-port.

UART (*Universal Asynchronous Receiver/Transmitter*) a device able to translate between parallel and an asynchronous serial communications signals for transmission and reception between a parallel processor bus and a serial communications port.

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



VMX (*VMEbus Memory eXpansion bus*) an older expansion memory bus for the P2 connector. Has been generally superseded by the VSB.

VSB (*VMEbus Subsystem Bus*) a fairly standard medium performance method of accessing expansion memory or peripherals via the P2 connector.

VMEbus *(Versa Module Eurocard bus)* a microcomputer architecture whose physical and electrical characteristics are defined in the IEC 821 and IEEE 1014-1987 specifications. The VMEbus supports separate address and data lines of up to 32 bits each. This bus uses a backplane in which VMEbus modules are interconnected using DIN-41612 connectors.

watchdog an on-board timer that can automatically reset the board if not accessed on a regular basis. Used to reset the board in response to a software loop and/or malfunction or a CPU halt.

window size the range of contiguous addresses that the board responds to is called the window. The number of addresses in the window is called the window size. The board will respond to addresses from base to base+window size.

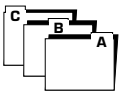
word typically, a unit of data 16 bits in length.

writethrough A caching method that writes dirty (updated) cache data to the main memory as soon as it is updated by the processor.

WWV call letters for the National Bureau of Standards radio station in Ft. Collins, Colorado. WWV broadcasts technical services including timing signals, audio frequencies, and radio-propagation disturbance warnings at the 2.5, 5, 10, 15, and 25 MHz carrier bands. Canada provides similar services on CHU.

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

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



Index

V452 Quick Index

R452
- installation **2-13**

V452
- address map **3-4**
- block diagram **1-6**
- dimensions **B-1**
- features
  - bus **1-8**, **1-9**
  - CPU **1-7**
  - interrupts **1-9**
  - memory **1-7**
  - peripherals **1-10**
- hardware configuration **2-17**-**2-23**
- installation **2-7**-**2-11**, **2-13**
- minimum system requirements **2-3**
- operating environment **B-2**
- power requirements **B-2**
- repair **7-6**
- revisions
  - levels **C-1**
- rework/upgrades **C-1**
- software and operating systems **2-30**-**2-34**
- software configuration **2-35**-**2-51**
- VMEbus compliance **B-1**
- voltages **2-4**
- warranty **7-3**, **7-6**
- weight **B-1**

Main Index

10BASE-T
- defined **Glos-1**

2692 **3-15**, **3-16**
- back-to-back protection **5-5**
- code example **6-33**
- defined **Glos-1**
- serial port registers **5-4**

2692 DUART **4-89**, **4-91**

27C010 **2-7**, **4-73**

27C020 **2-7**, **4-73**

27C040 **2-7**, **4-73**

27C080 **2-7**, **4-73**

28F010 **2-7**, **4-73**

28F020 **2-7**, **4-73**

48T02
- defined **Glos-1**

68040 **4-3**-**4-29**
- 68040 RESET instruction **3-62**
- additional documentation **4-3**, **4-4**
- addressing modes **4-14**
- bus snooping **4-24**
- caches **4-18**-**4-22**
  - cache coherency **4-19**
  - cache instructions **4-20**
  - cache organization **4-18**
  - setting up the instruction and data caches **4-21**
  - using the transparent translation register **4-29**
- data types **4-12**
- dual-68040 version
  - interrupt operations **3-11**
- exception processing **4-16**-**4-17**
  - exception vectors **4-17**
- halt **3-33**
- instruction set **4-14**, **4-15**
- MMU **4-27**-**4-29**
  - MMU instructions supported **4-28**
- programming model **4-8**-**4-11**
- registers **4-8**-**4-11**

68040/060
- caches
  - configuration **2-39**
  - using the Transparent Translation register **2-39**

68060 **4-31**

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

additional documentation **4-33**
addressing modes **4-36**, **4-37**
bus snooping **4-45**
caches **4-43**
cache coherency **4-45**
cache organization **4-44**
invalidating the caches after BLT transfers **5-64**
reducing memory contention using **4-70**
setting up the instruction and data caches **4-45**
using the transparent translation register **4-46**
data types **4-35**, **4-36**
dual-68060 version
counter operations **4-97**
timer operations **4-93**
watchdog (CPU) operations **4-61**
exception processing **4-42**
exception vectors **4-42**
instruction set **4-37**-**4-38**
PMMU **4-48**
programming model **4-33**

8 MB mode, setup **5-30**

82C54 counter **4-89**, **4-96**

82C596 **5-8**

A32/A24 VME addressing, setup **5-24**

ABORT **1-10**, **3-14**, **3-31**
as an interrupt source **3-12**
defined **Glos-2**
toggle (front panel) **3-31**

access contention **5-32**

ACFail **3-14**

address map **3-3**-**3-6**
access behavior **3-3**
decoding **4-70**

AM code bits
defined **Glos-2**

application LEDS, setup **2-44**

arbiter *(see-system controller)*

arbitration
DRAM contention **4-69**

ATC
defined **Glos-2**

back-to-back access protection (2692) **5-5**

battery, replacing (clock calendar/SRAM) **4-106**

BCD
defined **Glos-2**

BErr\ **5-66**

block diagram
V452 **1-6**
68040 **4-6**
68060 **4-32**

BLT **4-63**,**5-62**
block transfer strategy **5-54**
master **5-53**
invalidating the 68060 caches after a BLT transfer **5-64**
operational considerations **62**
setting up a BLT read from VME **5-57**-**5-59**
setting up a BLT write from VME **5-59**-**5-62**
typical BLT cycle **5-56**
using Counter2 (82C54) as a BLT throttle **4-63**, **5-63**
code example **6-5**
slave **5-33**
use with MMU (see note) **5-57**
using Counter2 (82C54) as a BLT throttle **99**
using the MOVES instruction **5-55**

board information registers **3-55**

boot select jumper **2-25**, **3-28**, **3-63**, **4-76**

boot state **3-63**
architecture **3-64**

bus address
defined **Glos-2**

bus arbiter **5-66**

bus error **5-32**, **5-65**
setting the bus error timeout interval **5-67**

Bus Request and Control register *(see—registers)*

bus snooping **2-50**

cables *(see also—serial interface)*
serial interface **A-15**

cache & data fetch, setup **2-39**

category 3

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

defined **Glos-2**

category 5
defined **Glos-2**

Cbl/J10J8D25M2m, cable adapter **A-25**

Cbl/J10J8J81m, cable adapter **A-25**

clock (DUART)
source **4-92**

clock/cal/NVRAM compatibility mode, 2KB vs. 8KB (default vs. extended) **4-101**, **4-105**

clock/calendar **4-101**
calibrating **4-104**
code example **6-25**
defined **Glos-2**

CMOS
defined **Glos-3**

code examples
2692 counter/timer code **6-25**
2692 DUART **6-33**
BLT with DMA **6-5**
clock/calendar **6-25**
Flash EPROM programming **6-15**

configuration
68040
setting up the instruction and data caches **4-21**
68040/060
setting up the instruction and data caches **2-39**
68060
setting up the instruction and data caches **4-45**
counters (82C54)
clearing counter interrupts **4-98**
disabling the counter as an interrupt source **4-98**
enabling the counter as an interrupt source **4-97**
selecting the counter mode and source **4-96**
using Counter2 as a BLT throttle **4-99**, **5-63**
default hardware configuration **2-18**
default software conditions **2-36**
enabling serial ports B and D **2-44**, **3-47**
Flash memory
write enable/disable **2-20**
hardware configuration **2-17**-**2-23**
interrupts
enabling/disabling **2-42**-**2-43**
LED **2-44**
mailbox (CPU)
disabling the mailbox as an interrupt source **4-54**
enabling the mailbox as an interrupt source **4-53**
monitor PROM (EPROM)
Flash EPROM write enable/disable **4-75**
start-up vectors **3-64**
type selection **4-74**
parity checking **2-45**
serial interface
building modular cables **A-17**
wiring Modular-to-D adapters **A-19**
software configuration **2-35**-**2-51**
system controller
configuring the bus arbiter **5-66**
enabling the system controller **2-24**, **5-66**
selecting round robin or priority request handling **2-23**, **5-66**
setting the bus error timeout interval **5-67**
system terminal or console settings **2-29**
timers (2692)
clearing timer interrupts **4-95**
disabling the timer as an interrupt source **4-94**
enabling the timer as an interrupt source **4-93**
selecting the timer mode and source **4-92**
setting the timer interrupt level **4-93**
VME master interface
selecting FAIR or non-FAIR bus requests **2-45**, **5-49**
selecting ROR or RWD bus releases **5-50**
selecting ROR or RWD bus requests **2-45**
setting the VMEbus request level **2-21**, **5-47**
VME slave interface
default conditions **5-22**
enabling the VME Slave interface **2-46**, **5-31**
enabling VME slave remote reset **2-22**, **5-23**
selecting A32 vs. A24 VME addressing **5-24**
selecting the upper or lower VME A32/D32 address range **2-48**
selecting the window size and base address **2-48**, **5-27**
selecting VME Slave characteristics **2-46**, **5-22**
setting the Slave memory protection level **2-47**, **5-24**

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

VMEbus
enabling SysFail as an interrupt source 2-43
suppressing the board's VME SysFail **2-49**
watchdog (CPU)
disabling all watchdog functions **4-61**
enabling/disabling the watchdog **2-50**
halt monitor **4-60**
run monitor **4-59**

connectors
P1 **2-3**
defined **Glos-7**
pinout **A-3**
P10
pinout **A-7–A-10**
P2 **1-10**, **2-4**
defined **Glos-7**
pinout **A-3**
serial interface option **A-29**-**A-31**
P3 **5-18**
pinout **A-5**
P4 **5-18**
pinout **A-5**
P5-P8
pinout **A-11**, **A-13**
P9
pinout **A-7**-**A-10**

construction **1-8**

contention **5-32**

copyback (cache) *(see also—68060:caches)*

counter/timer *(see counters (82C54) -or- timers (2692))*

counters (82C54) **4-96**-**4-99**, **5-63**
clearing counter interrupts **3-18**, **4-98**
disabling the counter as an interrupt source **4-98**
enabling the counter as an interrupt source **4-97**
interrupts **3-18**
selecting the counter mode and source **4-96**
using Counter2 as a BLT throttle **4-99**, **5-63**

CPU *(see also—68040)*, *(see also—68060)*
defined **Glos-3**

CPU mailbox *(see—mailbox (CPU))*

CPU watchdog *(see—watchdog (CPU)*

CPU watchdog, setup **2-50**

CRT
defined **Glos-3**

customer service **7-6**, **C-1**

cycle stealing
defined **Glos-3**

data broadcasting, setup **5-36**, **5-37**, **5-40**

data broadcasts **5-35**-**5-45**
defined **Glos-3**

daughter modules *(see—EZ-bus)*

DB-9
defined **Glos-3**

DbA / DbB **3-34**

DCE
defined **Glos-3**

DELF/DEFL **4-76**

differential
defined **Glos-4**

dimensions (board) **B-1**

DMA
defined **Glos-4**

DRAM **4-63**-**4-72**
address decoding **4-70**
BLT *(see BLT)*
contention **5-32**
defined **Glos-4**
DRAM speed in dryhstones **4-65**
memory access contention **4-69**
parity checking **4-65**-**4-69**
clearing the parity error bit **4-68**
detecting & isolating bad parity **4-68**
enabling/disabling **2-43**, **4-66**
enabling/disabling parity error interrupts **4-66**
tuning to CPU speed **4-64**

DRAM initialization **2-45**

DRAM, setup **2-45**

DTE
defined **Glos-4**

dual-CPU version
start-up vectors for CPU-X and CPU-Y **3-64**

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

dual-CPU, dual-68040 *(see—68040)*

dual-port
defined **Glos-4**

DUART *(see 2692)*
defined **Glos-10**

ECO
defined **Glos-4**

EPROM *(see also—monitor PROM (EPROM))*
defined **Glos-4**

EPROM boot enable jumper **2-25**, **3-28**, **3-63**, **4-76**

EPROM/Flash boot select, setup **2-25**

Ethernet
defined **Glos-4**

Ethernet interface **5-13**
Address map **5-13**
CSMA/CD **5-11**
Data transmission **5-9**
Ethernet ID **5-11**
Ethernet network connections **5-9**
Interchange signals **5-12**
Introduction **5-7**

exceptions (CPU) *(see—68040: exceptions)*

Extended Mode register *(see—registers)*

EZ-bus **4-69**, **4-71**, **5-15**, **5-20**
as reset source **3-62**, **3-63**
connectors **5-18**
pinout **A-5**
custom modules **5-18**
DbA & DbB LEDs **3-34**
defined **Glos-4**
interrupts **3-17**
pinout warning **2-4**

Fail *(see—LED: Fail)*

FIFO
defined **Glos-5**

Flash
block organization (Flash Module option) **4-82**
block organization (onboard Flash) **4-86**
booting from onboard Flash **4-87**
considerations when using **4-77**, **4-83**
EPROM **4-75**
Flash module option (DEFL/DELF) **4-79**
Flash module, accessing the **4-82**
Flash module, installing the **4-80**
onboard Flash memory **4-85**

Flash memory
defined **Glos-5**

Flash write enable/disable, setup **2-20**

floating-point
defined **Glos-5**

FPU
defined **Glos-5**

front panel layout **3-29**

fuses, ratings & locations **3-35**

GPIB
defined **Glos-5**

GPS receiver
defined **Glos-5**

group address **5-35**

halt (CPU) *(see also—68040: halt -or- watchdog (CPU): halt monitor)*
halt monitor **4-60**-**4-61**

hardware configuration *(see configuration)*

humidity specification **B-2**

I/O
defined **Glos-6**

IAck **5-65**
defined **Glos-5**

IBM
defined **Glos-5**

ID switch (software readable) **3-30**

ID switch register *(see—registers)*

IEEE P754
defined **Glos-5**

IMR
defined **Glos-5**

initialization
monitor PROM start-up vectors **3-64**

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

installation
V452 Series board installation
system terminal or console settings **2-29**
installing a monitor PROM **2-7**-**2-11**
installing the R452 memory module **2-13**
minimum system requirements **2-3**
operational warning when using a card cage with 6-layer PCBs **2-3**

Interrupt control register *(see—registers)*

interrupt handler *(see also—VMEbus: interrupt handler)*
defined **Glos-6**

interrupter *(see also—VMEbus: interrupter)*
defined **Glos-5**

interrupts **1-9**, **3-7**
ABORT (nmi Level 7) **3-12**
ACFail **3-14**
configuration
enabling/disabling interrupts **3-9**
counters (82C54) **3-18**
clearing counter interrupts **3-18**, **4-98**
disabling the counter as an interrupt source **4-98**
enabling the counter as an interrupt source **4-97**
CPU exceptions **4-17**, **4-42**
enabling/disabling interrupts **2-42**-**2-43**
EZ-bus **3-17**
Interrupt Control register
address locations (listed) **3-10**
enabling VMEbus interrupts **3-20**
Level 7 (maskable) **3-13**
local bus timeout **3-19**
mailbox (CPU) **4-52**, **4-53**-**4-54**
disabling the mailbox as an interrupt source **4-54**
enabling the mailbox as an interrupt source **4-53**
maskable Level 7
enabling the SysFail interrupt **2-43**
non-steerable interrupt sources **3-12**
on-board interrupts **3-12**-**3-18**
parity **3-14**
serial interface **3-16**
single vs. dual-68040 **3-11**
steerable interrupt sources **3-11**
timer
configuring for Level 6 **3-48**
timer (2692) **3-15**
timers (2692)
clearing timer interrupts **3-15**, **4-95**
disabling the timer as an interrupt source **4-94**
enabling the timer as an interrupt source **4-93**
setting the timer interrupt level **4-93**
vectors **3-7**
VME SysFail **3-14**
enabling the SysFail interrupt **2-43**

ISP
defined **Glos-6**

ispPAL
interrupt steering function **3-8**

JEDEC
defined **Glos-6**

jumpers **3-25**-**3-27**
boot select **4-76**
EPROM configuration (JK12) **4-74**
Flash write protect **4-75**
JK11 (Flash EPROM write enable), **2-10**
JK12 (PROM type selection), **2-10**, **3-26**
JK17 **3-26**
enabling Flash write **2-20**
enabling VME slave remote reset **2-22**
EPROM/Flash boot select **2-25**
selecting round robin or priority request handling **2-23**
setting the VMEbus request level **2-21**
system controller force disable **2-24**
JK17 jumper location diagram **2-19**
JK27 **2-18**
enabling the System Controller **5-66**
enabling the VME slave remote reset **5-23**
setting the VMEbus request level **5-48**
jumper functional summary **3-26**
jumper location diagram **2-8**, **3-26**, **4-74**
jumper setting summary **3-27**

LED **3-32**
0 - 7 (application) **3-33**
controlling the User LEDs **3-42**
CPU status (Halt/Fail) **3-33**
CPU-X **3-34**
CPU-Y **3-34**
DbA **3-34**
DbB **3-34**
defined **Glos-6**
Ethernet **3-34**
Mst **3-34**
Slv **3-34**
turning on and off **2-44**

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

Level 1010
defined **Glos-6**
Level 1111
defined **Glos-6**
local bus timeout **3-19**
M48T18 **4-101**, **4-102**
mailbox (CPU) **3-18**, **4-51**-**4-55**
defined **Glos-6**
interrupts **4-53**-**4-54**
disabling the mailbox as an interrupt source **4-54**
enabling the mailbox as an interrupt source **4-53**
read area **4-52**
write area **4-51**
master *(see VMEbus — master interface)*
defined **Glos-6**
memory access contention **4-69**
memory module **4-72**
connector pinouts **A-7**
installation **2-13**
memory protection
configuration **2-47**
defined **Glos-6**
MMU *(see—68060: MMU)*
defined **Glos-6**
use during BLT transfers (see note) **5-57**
mode registers **3-40**, *(see—registers)*
monitor PROM (EPROM) **4-73**, **4-76**
as boot PROM after reset **3-64**
Flash EPROM programming code sample **6-15**
Flash EPROM write enable/disable **2-10**, **4-75**
installation **2-7**-**2-11**
location of PROM sockets on board **2-8**
OS-9 **2-30**
pSOS/Probe **2-32**
start-up banners **2-30**-**2-34**
start-up vectors **3-64**
SMon **2-34**
type selection **2-10**, **4-74**
VxWorks **2-32**
MOS
defined **Glos-6**
MOVES
use during BLT transfers **5-55**
µs
defined **Glos-6**
ms
defined **Glos-7**
MSB
defined **Glos-7**
Mst **3-34**
multi-port
defined **Glos-7**
multi-ported memory
access contention **5-32**
self-references to on-board memory **5-32**
NAN
defined **Glos-7**
nmi
defined **Glos-7**
non-volatile RAM (NVRAM) *(see—SRAM (non-volatile))*
ns
defined **Glos-7**
NVRAM **4-105**
defined **Glos-7**
object code
defined **Glos-7**
operating environment **B-2**
operating systems **2-30**-**2-34**
OS-9 **2-30**
pROBE.jr **2-32**
SMon **2-34**
VxWorks **2-32**
P1, P2, etc., *(see connectors)*
PAL
defined **Glos-7**
parity **4-65**
errors **3-14**
peripherals **1-10**
physical address (Ethernet) **5-11**
physical configuration **1-5**

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

power
consumption **2-4**
power supply **2-4**
requirements **B-2**
power-cycle
as reset source **3-61**
power-on banners **2-30**
prescaler
defined **Glos-7**
Primary Mode register *(see—registers)*
priority (VMEbus request arbitration) *(see—system controller)*
priority request/round robin, setup **2-23**
pROBE.jr **2-32**
programming differences (from earlier model SBCs) **6-3**
PROM *(see also—monitor PROM(EPROM))*
defined **Glos-8**
R452/R453 **4-72**
registers (V452 Series) **3-37**
board information **3-55**
Bus Request and Control register
disabling the CPU watchdog halt monitor **2-50**
selecting FAIR or non-FAIR bus releases **2-45**
selecting ROR or RWD bus releases **2-45**
Control registers **3-50**
extended **3-51**
primary **3-51**
default conditions **3-59**
Ethernet/VMEbus control **3-55**
Extended Control register
disabling the halt monitor **4-60**
enabling the halt monitor **4-60**
selecting FAIR or non-FAIR bus requests **5-49**
selecting ROR or RWD bus releases **5-50**
VME master interface functions **5-48**
Extended Mode register
enabling the Slave interface **2-49**, **5-31**
enabling/disabling the CPU watchdog run monitor **2-51**
turning LEDs on and off **2-45**
ID switch register **3-39**
Interrupt control **3-52**
Interrupt Control register
address locations (listed) **3-10**
clearing counter (82C54) interrupts **4-98**
clearing timer (2692) interrupts **4-95**
disabling a counter (82C54) as an interrupt source **4-98**
disabling a timer (2692) as an interrupt source **4-94**
enabling a counter (82C54) as an interrupt source **4-97**
enabling a timer (2692) as an interrupt source **4-93**
enabling VMEbus interrupts **3-20**
enabling/disabling interrupt sources **2-42**, **2-43**, **3-9**
enabling/disabling parity error interrupts **4-66**-**4-67**
Mode registers **3-40**
extended **3-41**
primary **3-41**
Primary Mode register
configuring Slave memory protection **5-24**
enabling/disabling parity checking **2-43**, **4-66**
enabling/disabling parity error interrupts **2-45**
setting the Slave memory protection level **2-47**
suppressing the VME SysFail signal **2-49**
turning LEDs on and off **2-45**
Slave Interface control **3-54**
Slave Interface Control register
configuring Slave memory protection **5-24**
selecting A32 vs A24 addressing **2-46**, **5-24**
selecting the upper or lower VME A32/D32 address range **2-48**
selecting the window size and base address **2-48**, **5-27**
Status register **3-38**
reporting parity errors **4-67**
remote reset **5-23**
remote reset register **3-67**
repair **7-6**
requester *(see—VMEbus : requester)*
defined **Glos-8**
RESET **3-31**, **3-33**, **3-61**
68040 RESET instruction **3-62**
boot state **3-63**
EZ-bus **3-62**
hardware **3-63**

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

remote **3-62**
sequence **3-63**
software **3-62**, **3-65**
sources **3-61**, **3-62**
toggle (front panel) **3-31**

revisions
main board (V452) **C-1**
memory module **C-1**

rework/upgrades **C-1**

RMA **7-6**
defined **Glos-8**

RMW
defined **Glos-8**

ROAK
defined **Glos-8**

ROR **5-50**
defined **Glos-8**

RORA
defined **Glos-8**

round robin (VMEbus request arbitration) *(see—system controller)*
defined **Glos-8**

round robin/priority request, setup **2-23**

RS-232
defined **Glos-8**

RS-422
defined **Glos-9**

RWD **5-50**
defined **Glos-9**

SBC
defined **Glos-9**

SCSI
defined **Glos-9**

SDRAM
defined **Glos-9**

serial cable adapters **A-25**

serial interface **5-3**-**5-6**
2692 registers **5-4**
building modular cables **A-17**
cables and adapters **A-15**
code example **6-33**
connector pinouts (P5-P8) **A-11**, **A-13**
enabling serial ports B and D **2-44**, **3-47**
interrupts **3-16**
P2 interface option **A-29**-**A-31**
wiring Modular-to-D adapters **A-19**

serial ports B and D, setup **2-44**

single-ended
defined **Glos-9**

slave *(see VMEbus — slave interface)*
defined **Glos-9**

Slave Interface Control register *(see—registers)*

Slave interface, setup **5-31**

Slv **3-34**

SMon **2-34**

sockets
UG13/UJ13 (EPROM) **4-73**

software configuration *(see—configuration)*

software reset **3-65**

software-readable switch *(see—switches: ID switch)*

specifications
dimensions **B-1**
environmental **B-2**
power **B-2**
power requirements **B-2**
weight **B-1**

spurious interrupt
defined **Glos-10**

SRAM
defined **Glos-10**

stale data (cache) *(see also—68060:caches)*
defined **Glos-10**

start-up banners **2-29**

start-up vectors **3-64**, **4-76**

Status register *(see—registers)*

supervisor
defined **Glos-10**

switches
ABORT toggle **3-31**
ID switch (software-readable) **3-30**

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

RESET toggle **3-31**
as reset source **3-61**
SysClk\
defined **10**
SysFail *(see also—VMEbus : SysFail)*
defined **Glos-10**
SysRes\
defined **Glos-10**
system controller **5-65**
configuring the bus arbiter **5-66**
defined **Glos-10**
enabling the system controller **2-24**, **5-66**
selecting round robin or priority request handling **2-23**, **5-66**
setting the bus error timeout interval **5-67**
system controller, setup **2-23**, **2-24**, **5-66**
temperature requirements **B-2**
timeout
local bus **3-19**
timers (2692) **4-91**
clearing timer interrupts **4-95**
code example **6-25**
disabling the timer as an interrupt source **4-94**
enabling the timer as an interrupt source **4-93**
interrupts **3-15**
selecting the timer mode and source **4-92**
setting the timer interrupt level **4-93**
timer operations **4-92**
TRAPV **4-17**, **4-42**
defined **Glos-10**
UART
defined **Glos-10**
V440/V460 and V452, comparing the differences **1-11**
vectors **3-5**
video display terminal **2-5**
VME 6U **B-1**
VME interrupt level, setup **3-22**
VME interrupt vector, setup **3-23**
VME interrupts, setup **2-42**
VME Level 0 interrupt reset **3-65**
VME master interface *(see—VMEbus : master interface)*
VME master interface, setup **2-45**, **5-47**
VME slave interface *(see—VMEbus : slave interface)*
VME slave remote reset, setup **2-22**, **5-23**
VME SysFail interrupt, setup **5-68**
VME SysRes, setting up respond to/ignore **3-66**
VME64 extensions **1-9**
VMEbus **1-8**, **2-21**, **2-22**, **4-69**
ACFail **3-14**
arbiter *(see-system controller)*
bus error timeout *(see—system controller)*
compliance **B-1**
interrupt handler **B-1**
interrupter **B-1**
master interface **B-1**
operational warning when using a card cage with 6-layer PCBs **2-3**
slave interface **B-1**
connectors (on-board)
pinout **A-3**
defined **Glos-11**
interrupter **3-22**, **3-49**
interrupts **1-9**, **3-19**, **3-49**
interrupt handler **3-19**
VMEbus interrupt sequence **3-19**
master interface **5-47**-**5-51**
bandwidth **5-51**
Mst LED **3-34**
selecting FAIR or non-FAIR bus requests **2-45**, **5-49**
selecting ROR or RWD bus releases **5-50**
selecting ROR or RWD bus requests **2-45**
setting the VMEbus request level **2-21**, **5-47**
requester **5-50**
slave interface **5-21**-**5-33**
configuring Slave memory protection **2-47**, **5-24**
data broadcasting option **5-35**-**5-45**
default conditions **5-22**
enabling remote reset **2-22**, **5-23**

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

enabling the VME Slave interface **2-46**, **5-31**
selecting A32 or A24 addressing **5-24**
selecting A32 vs A24 addressing **2-46**
selecting the upper or lower VME A32/D32 address range **2-48**
selecting the VME Slave characteristics **2-46**, **5-22**
selecting the window size and base address **2-48**, **5-27**
self-references to on-board memory **5-32**
Slv LED **3-34**
SysFail **3-14**, **3-33**
enabling and disabling reception of **3-48**
enabling the SysFail interrupt **2-43**
suppressing the VME SysFail signal **49**
SysRes **3-33**
as reset source **3-61**
asserting a VME system reset **3-22**, **3-61**

VMEbus request level, setup **2-21**, **5-47**

VMX
defined **Glos-11**

voltages **2-4**, **B-2**

VSB interface **4-59**
defined **Glos-11**

VxWorks **2-32**

warranty **7-3**, **7-6**

watchdog (CPU) **4-57**, **4-61**
as reset source **3-62**
configuration **2-50**
defined **Glos-11**
disabling all watchdog functions **4-61**
halt monitor
disabling the halt monitor **4-60**
enabling the halt monitor **4-60**
run monitor **4-58**-**4-59**
disabling the run monitor **4-59**
enabling the run monitor **4-59**
holding off a reset **4-58**, **4-59**

watchdog timer enable reset **3-66**

weight **B-1**

window size *(see also—VMEbus : slave interface)*
defined **Glos-11**

word
defined **Glos-11**

writethrough (cache) *(see also—68060:caches)*

WWV
defined **Glos-11**

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

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