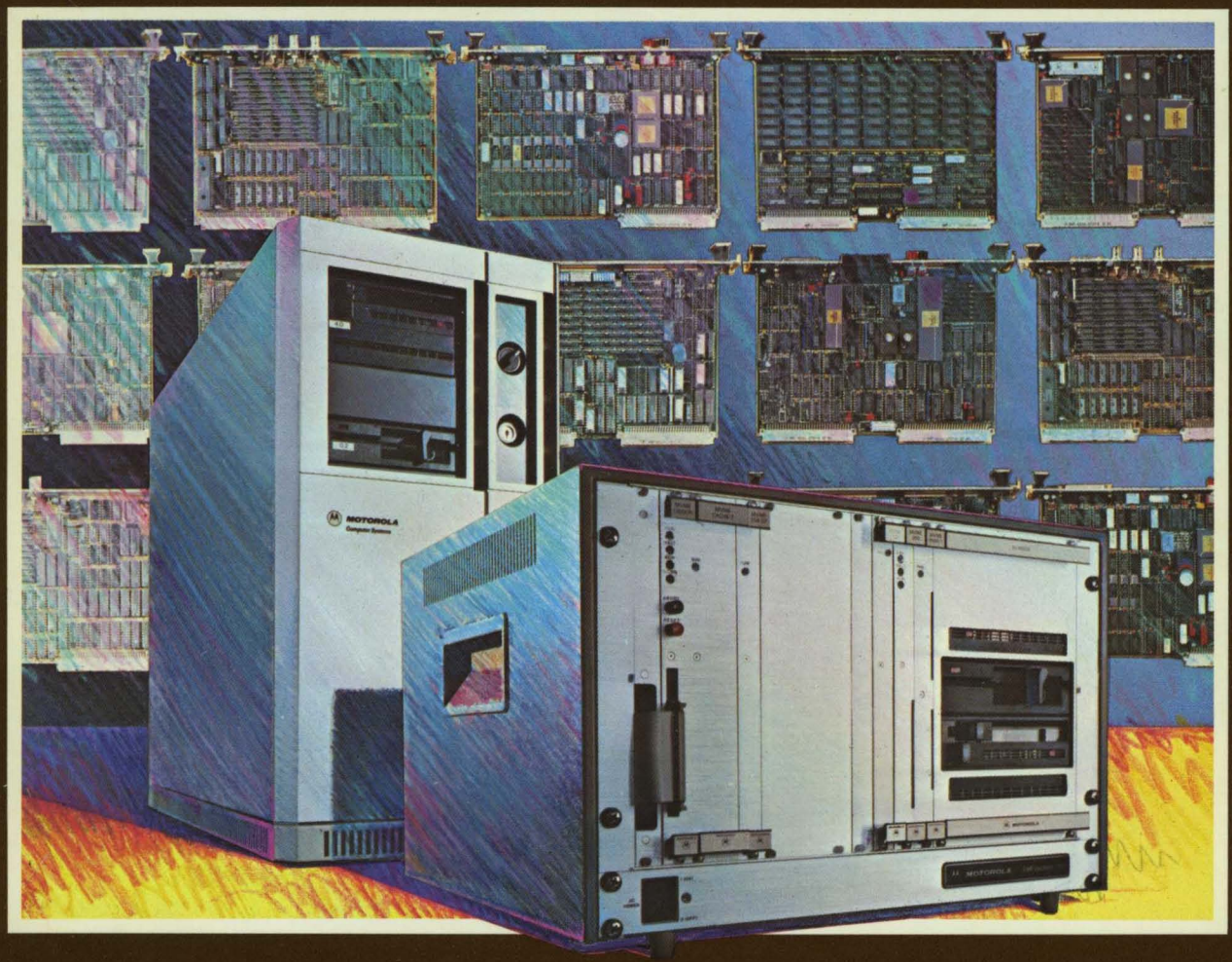
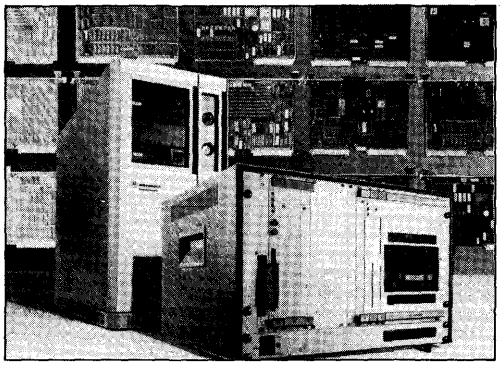
DL127
REV 2

MOTOROLA



MICROCOMPUTER SYSTEMS AND COMPONENTS



Standard Warranty

Seller warrants that its products sold hereunder will at the time of shipment be free and clear of all liens and encumbrances and will be free from defects in material and workmanship and will conform to Seller's applicable specifications or, if appropriate, to Buyer's specifications accepted by Seller in writing. If products sold hereunder are not as warranted, Seller shall, at its option, refund the purchase price, repair, or replace the product, provided proof of purchase and written notice of nonconformance is received by Seller within ninety (90) days from date of initial shipment, and provided said nonconforming products are, with Seller's written authorization,* returned FOB Seller's plant or authorized repair center within thirty (30) days from expiration of said ninety (90) day period. Upon verification by Seller that the product does not conform to this warranty, Seller will pay the cost of transporting such replacement of repaired goods to Buyer's plant within the contiguous 48 United States and Canada. This warranty shall not apply to any products Seller determines have been, by Buyer or otherwise, subjected to testing for other than specified electrical characteristics or to operating and/or environmental conditions in excess of the maximum values established in applicable specifications, or have been the subject of mishandling, misuse, neglect, improper testing, repair, alteration, damage, assembly or processing that alters physical or electrical properties. This warranty excludes all costs of shipping, customs clearance and related charges outside the contiguous 48 United States and Canada.

IN NO EVENT WILL SELLER BE LIABLE FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. THIS WARRANTY EXTENDS TO BUYER ONLY AND NOT TO BUYER'S CUSTOMERS OR USERS OF BUYER'S PRODUCTS AND IS IN LIEU OF ALL OTHER WARRANTIES WHETHER EXPRESS, IMPLIED OR STATUTORY INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS.

MOTOROLA

MICROCOMPUTER SYSTEMS AND COMPONENTS

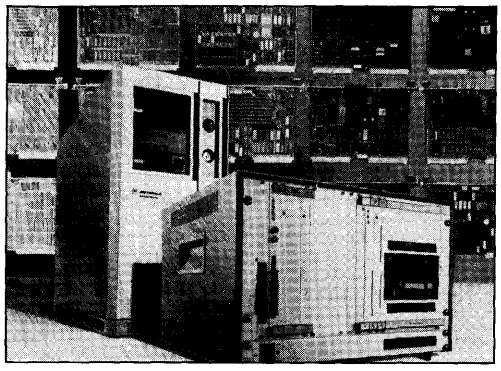
Prepared by
Technical Information Center

Preface

Following Motorola's tradition of serving current and future needs of the fast evolving electronic market, the Motorola Microcomputer Division offers systems and components ranging from VMEbus-based multiuser microcomputers to larger form-factor VERSAbus microcomputers — plus a broad line of compatible modular components, peripherals and software. System integrators and value added resellers are provided support for integration and development of today's VMEbus microcomputer and SYSTEM V/68-based products with languages and development tools, data management and communications facilities and office applications. OEMs are provided support for development of today's MC68000-, MC68010- and MC68020-based products with instrumentation, hardware/software development tools and the SYSTEM V/68 and VERSAdos Operating Systems. The consistent Motorola MC68000-family migration path and future development tools plus a multiprocessor operating system will assure support for the development of tomorrow's MC68030-based leading edge systems.

Third Edition
MOTOROLA INC., 1987

Printed in U.S.A.



A rainbow arches over the application world of a general purpose microcomputer product line, particularly the VME Delta Series which utilizes an open VMEbus architecture, an AT&T-validated UNIX V Operating System — SYSTEM V/68 — and the state-of-the-art MC68020 Microprocessor. At one end of the rainbow is a land of applications served by system integrators, value added resellers and complementary marketing organizations selling limited-user systems into business-oriented markets. The other end touches down in a nearly unbounded field of custom OEM applications in such technological areas as graphics processing for CAD simulation and medical scanning, analysis of oil exploration instrumentation data, visions systems for positioning and attribute detection in robotics and other factory automation, artificial intelligence research, in cellular communications and many other leading edge areas.

The current array of VME Delta Series products will continue to grow with new markets, offering hardware for system expansion including a cache accelerator, color and monochrome terminals and a remote service modem for remote diagnostics. Support for industry standard communication protocols is available. This includes intelligent controllers and supporting software for the IEEE 802.3 (CSMA/CD) LAN standard and the SNA/BSC/X.25 protocols. Support for business applications will grow too through intelligent controllers with supporting software for SNA and BSC and other commonly used protocols, through information retrieval, processing and development packages, through an integrated spreadsheet and color graphics support and a project management system. In time, the VME Delta Series line will span an applications world of its own.

VME Delta Series Data Sheets

MOTOROLA

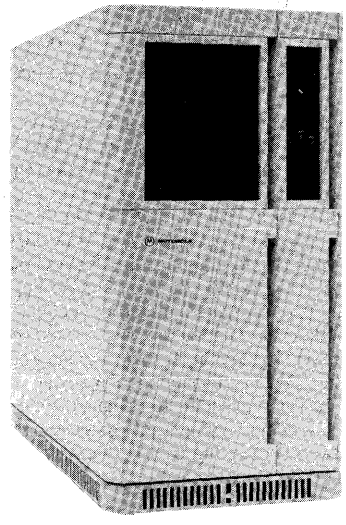
PBC101

VME Delta Series

MODEL 2016 COMPUTER SYSTEM

HIGHLIGHTS

- ☐ **Motorola MC68020 processing power**
- ☐ **12-slot VME compatible enclosure**
- ☐ **2 to 16 Mb of main memory**
- ☐ **2 to 34 RS-232 asynchronous ports**
- ☐ **85 to 340 Mb of integral Winchester disk storage**
- ☐ **Up to 1.3 Gb of external SMD capacity**
- ☐ **60 Mb streaming tape**
- ☐ **Software compliant with System V Interface Definition (SVID) Release 3**
- ☐ **Comprehensive local and remote diagnostics**



GENERAL DESCRIPTION

The VME Delta Series family of 32-bit computer systems is based on the Motorola MC68020 microprocessor, the VMEbus standard, and UNIX® System V, Release 3. The Model 2016 uses the 16.7 MHz MC68020 and supports up to 34 serial devices. The power of the MC68020 plus the expandability and flexibility of the VMEbus architecture make the Model 2016 an ideal computer solution for commercial, technical, and networking applications.

The Model 2016's 12-slot VME card cage adapts to most application interfaces and expands to support large configurations. In addition, the system enclosure accommodates up to four disk drives plus a streaming tape. External SMD and tape subsystems are also supported. Both local and wide area communications protocols link the Model 2016 to industry standard networks as well as host computers.

HARDWARE OVERVIEW

The Model 2016 hardware includes:

- ☐ The system enclosure
- ☐ The VMEbus
- ☐ The applications processor and memory
- ☐ I/O and communications controllers
- ☐ Mass storage devices
- ☐ Local and remote diagnostic capabilities

System Enclosure

The VME Delta Series enclosure offers a 12-slot VME compatible card cage, a 12-slot VME cable connection area, and an SMD and tape subsystem cable connection area. This provides connectivity to a diverse and large number of peripheral devices.

The unit houses a streaming tape drive and up to four 5-1/4" disk drives. One of the disk drive slots is used for half-height diskette drives when required by the configuration. All drives are held in shock-mounted drive assemblies.

The enclosure contains a 450 watt switching power supply and is cooled by filtered air.

VMEbus

The VME Delta Series uses today's leading 32-bit bus standard, the VMEbus. The VMEbus, in conjunction with the double-high Eurocard form factor, offers systems integrators and OEMs high performance as well as access to an expanding set of readily available controllers and system components. The Model 2016's 12-slot VME compatible card cage also allows the system to grow in a modular fashion to meet future as well as current requirements. In addition, upgrading the Model 2016 to a more powerful model in the VME Delta Series family is as

simple as replacing the processor and memory boards.

Main Processor

The Model 2016's main processor includes a 32-bit 16.7 MHz MC68020 microprocessor, an optional cache accelerator, a memory management unit, and a floating point coprocessor. Two RS-232 ports are included. These are normally used for a console port and a remote service modem. The MC68020 has an on-chip 256 byte instruction cache. The processor uses the high-speed VME Subsystem Bus (VSB) for parity memory access and the VMEbus to interface I/O controllers. The MC68881 floating point coprocessor implements the IEEE floating point standard plus trigonometric and transcendental functions.

Memory

Up to 16 Mb of main memory are offered on the Model 2016. Both ECC and parity memory options are available.

ECC memory comes in 4 Mb increments and provides single-bit error correction and double-bit error detection. A 64-bit cache register gives fast access using the VMEbus interface.

Parity memory comes in 2 Mb increments and provides byte parity generation and error checking. These memories are dual-ported featuring concurrent processor and I/O access using the VSB and VMEbus respectively.

Disk and Tape Controllers

The Model 2016 supports controllers for 5-1/4" Winchester and floppy disk drives, 8" SMD disk drives, 1/4" streaming tape, and 1/2" 9-track tape drives. All are intelligent controllers with DMA VME system interfaces. This selection of VME compatible disk and tape alternatives allows the Model 2016 to satisfy a wide spectrum of performance and application requirements.

Intelligent Communications Controllers

VMEbus compatible controllers are offered for:

- ☐ Asynchronous devices (terminals, printers, modems, etc.)
- ☐ SNA, BSC, and X.25 communications
- ☐ Ethernet local area networking

Each intelligent controller uses a 10 MHz. MC68000 family microprocessor.

The 8-Port Serial Controller can support eight asynchronous terminal, printer, or modem ports. Because each port can be independently configured, this controller can be adapted by the system integrator or OEM to support a wide range of serial devices. The Model 2016 supports up to four serial controllers.

The Intelligent Communications Con-

Controllers

Controller	Interface	Number of Channels/Drives	Transfer Rate (per second)	Controllers Per System
Winchester &	ST-506	2	Up to 5 Mbits	2
Floppy	SA-400	2	Up to 250 Kbits	
SMD	ESMD	2	Up to 25 Mbits	2
Streaming Tape	QIC-02	1	Up to 90 Kbits	1
1/2" 9-track Tape	Pertec	1	Up to 1.25 Mbits	1
8-Port Serial	RS-232-C	8	Up to 19.2 Kbits	4
Communications	RS-232-C	6	Up to 19.2 Kbits	1
Ethernet	CSMA/CD	1	10 Mbits	1

Mass Storage Devices

Form-Fit	Interface	Unformatted Capacity	Formatted Capacity	Average Access Time	Drives Per System
Disk Drives					
5-1/4"	SA-400	1 Mb	655 Kb	181 ms	2
5-1/4"	ST-506	85 Mb	67 Mb	28 ms	4
8"	ESMD	337 Mb	269 Mb	20 ms	4
Tape Drives					
5-1/4"	QIC-02	—	60 Mb	—	1
1/2" 9-track	Pertec				1
	1600 bpi	45 Mb			
	3200 bpi	90 Mb			

troller supports bit and byte oriented protocols like SDLC, HDLC, and BSC. Each port is independently programmable. A four-channel DMA controller is offered for high performance applications.

The Intelligent Ethernet LAN Controller is compatible with Ethernet, Version 2.0 and conforms to the IEEE 802.3 (CSMA/CD) LAN standard. The controller interfaces with the network at 10 Mbits per second.

Mass Storage Devices

The base Model 2016 contains a 5-1/4" 85 Mb Winchester drive and a 5-1/4" 60 Mb streaming tape for system backup and software distribution. Some configurations also include a floppy diskette which uses a disk slot. The system can be expanded to 340 Mb of Winchester disk storage and 1,348 Mb of SMD disk storage using two SMD disk subsystems.

Local and Remote Diagnostics

The VME Delta Series system diagnostics are oriented to centralized service networks and user participatory fault identification. They enhance the system's overall reliability, serviceability, and maintainability with the following diagnostic features:

- ☐ Hardware integrity verification at system power-up and reset
- ☐ System test and diagnostics in firmware, on stand-alone disk, and under the operating system
- ☐ Menu-driven user interface with on-line help facility
- ☐ Custom test suite generation, evaluation and debug tools
- ☐ Remote test and diagnostics with integral service modem

The Model 2016's comprehensive system diagnostic facilities are designed to support very cost effective customer and product service approaches.

SOFTWARE OVERVIEW

In addition to the following software products, a large number of software applications are available from independent software vendors which will run on the VME Delta Series.

UNIX System V Operating System

SYSTEM V/68™ Release 3 is Motorola's version of AT&T's UNIX® System V Release 3. Release 3 includes the following new features and enhancements:

- ☐ Remote File Sharing which allows users to share files, data, and peripheral devices across computer systems attached to a local area network.
- ☐ Streams which supports application development independent of the underlying network protocols.
- ☐ Virtual memory and demand paging support for efficient memory utilization and simpler application development.
- ☐ Menu Interface for system administration functions.
- ☐ Support for the MC68881 floating point coprocessor.
- ☐ Complete backup and restore facilities.

SYSTEM V/68 Release 3 is compliant with AT&T's System V Interface Definition (SVID).

Languages and Development Tools

- ☐ C Compiler and FORTRAN 77
- ☐ RM/COBOL®
- ☐ Micro Focus Level II COBOL™/ET, ANIMATOR™, and FORMS-2™
- ☐ Absoft FORTRAN
- ☐ Q-MENU® Menu Development Tool

Data Management Facilities

- ☐ UNIFY™ Relational Data Base Management System
- ☐ Oracle® Relational Data Base Management System
- ☐ Informix® Relational Data Base Management System

Data Communications Facilities

- ☐ 3274 SNA and BSC
- ☐ 3776 SNA
- ☐ RJE/HASP, 2780, and 3780
- ☐ X.25 with TCP/IP
- ☐ Ethernet with XNS
- ☐ Ethernet with TCP/IP

Office Applications

- ☐ Business Assistant™ User Services
- ☐ Q-ONE® Word Processor
- ☐ 20/20™ Spreadsheet
- ☐ Office Services (Electronic Mail, Calendar, etc.)
- ☐ Q-CHART™ Business Graphics
- ☐ MicroTrak™ Project Management System
- ☐ Q-FILE™ Information Storage and Retrieval System

PBC101

SPECIFICATIONS

System
- VMEbus (IEEE P1014)
- Eurocard form factor
  - Depth: 6.30" (160 mm)
  - Height: 9.17" (233 mm)
- 96 pin DIN 41612(P1) connectors
- 12 double-high VME card slots

Processor
- MC68020 at 16.7 MHz.
- Gate array MMU (1 Kb page) compatible with MC68851 PMMU
- MC68881 Floating point coprocessor (IEEE P754)
- 16 Kb cache accelerator option
- 2 RS-232-C asynchronous ports
  - 110 bps to 9,600 bps
- VSB Memory Bus

Memory
- ECC Memory
  - 4 to 16 Mb
  - 4 Mb increments
  - 64-bit cache register
  - 560 ns 32-bit average access
  - Single-bit error correction
  - Double-bit error detection
- Parity Memory
  - 2 to 8 Mb
  - 2 Mb increments
  - Dual-ported (VMEbus and VSB)
  - 660 ns 32-bit average access
  - Byte parity generation
  - Byte parity checking

Winchester and Floppy Disk Controller
- 1 or 2 per system
- Up to 2 ST-506 Winchester drives per controller
- Up to 2 floppy disk drives per system
- 5 Mbits per second serial data rates
- Multiple-sector read/write
- Automatic head switching and cylinder positioning
- 2:1 interleaving
- 32-bit ECC (Winchester)

SMD Disk Controller
- Up to 2 per system
- 1 or 2 SMD disks per controller
- Up to 25 Mb per second data rates
- Zero-latency reads and writes
- 1:1 interleave over multiple cylinders
- 32-bit ECC

5¼" Streaming Tape Controller
- QIC-02 Interface
- Transfer rate:
  - 90 Kb continuous
  - 200 Kb burst
- Streams at 5 Mb per minute

½" 9-Track Controller
- Pertec Interface
- Up to 200 ips GCR drive support
- Start/stop or streaming support
- GCR, PE, or NRZI recording

8-Port Serial Controller
- Up to 4 controllers per system
- 8 asynchronous RS-232-C ports
- 10 MHz. MC 68010 microprocessor
- 128 Kb RAM memory
- 32 Kb ROM
- Port selectable:
  - 50 – 19,200 bps
  - 1, 1.5, or 2 stop bits
  - 5 to 8 data bits
  - odd, even, or no parity
  - DTR, DSR, RTS, CTS, and DCD control functions
- Buffered pipe communication protocol

Intelligent Communications Controller
- 6 synchronous/asynchronous communications ports
- 10 MHz. MC68010 microprocessor
- 512 Kb RAM memory
- 128 Kb ROM
- SDLC, HDLC, and BSC protocol support
- Port selectable:
  - 50 – 19,200 bps
  - 1, 1.5, 2 stop bits
  - 5 to 8 data bits
  - odd, even, or no parity
  - NRZ, NRZI, or FM encoding
- Buffered pipe communications protocol

Intelligent Ethernet LAN Controller
- Ethernet, Version 2.0 compatible
- Conforms to IEEE 802.3 (CSMA/CD) LAN standard
- 10 Mbits per second
- 10 MHz. MC68000 microprocessor
- 512 Kb RAM memory
- 32 Kb EPROM
- Buffered pipe communications protocol

PHYSICAL SPECIFICATIONS

Height: 25.6 in. (65.0 cm)
Width: 12.5 in. (31.8 cm)
Depth: 28.0 in. (71.1 cm)
Weight: 135 lbs. (61 kg) Fully loaded Model 2016

ENVIRONMENTAL AND SAFETY SPECIFICATIONS

Safety
Meets UL 478 (EDP) and 114 (Office Equipment)
Meets CSA 154 (EDP) and 143 (Office Equipment)
Meets VDE 0806/8.81 (Office Equipment)
Meets IEC 380 (Office Equipment)

Emissions
Meets VDE 0871/6.78, Class A
Meets FCC Part 15, sub-part J, Class A

ESD
5,000 Volts: No observable effect
12,000 Volts: No operator perceived errors
24,000 Volts: No permanent equipment damage

Altitude
Operating: 10,000 feet ASL
Non-Operating: 30,000 feet ASL

Acoustic Noise Level
55 dBA maximum

Temperature
Operating: 5°C to 40°C
Non-Operating: – 40°C to 60°C

Relative Humidity
Operating: 10% to 80% RH, non-condensing
Non-Operating: 10% to 90% RH, non-condensing

Shock
Operating: .5 g
Non-Operating: 15 g

Transportation
Packaging and shipping containers and procedures comply with the current NSTA preship test procedures.

Shipping Container
Height: 34.5" (87.6 cm)
Width: 22.5" (57.2 cm)
Length: 41.0" (104.1 cm)

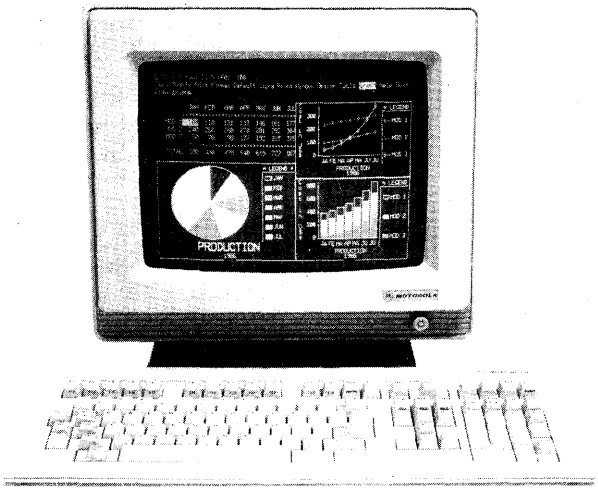
MOTOROLA

VME Delta Series

TM3000 TERMINALS

HIGHLIGHTS

- ☐ **Alphanumeric, monochrome, color, and graphics models**
- ☐ **DEC VT220 compatibility**
- ☐ **Five display modes up to 132 columns with 28 lines**
- ☐ **640 x 300 graphic display option**
- ☐ **Tektronix 4105 graphic commands**
- ☐ **Eight colors from a palette of 64**
- ☐ **Concurrent transparent print protocol**
- ☐ **RS-232 and RS-422 interfaces**



DESCRIPTION

Motorola TM3000 Terminals are a compatible family of high resolution ANSI X3.64 compliant terminals. They support tilt-and-swivel 14″ anti-glare displays with small footprints, ergonomic appearance and low profile keyboards.

TM3000 Terminals uniquely combine features of DEC, IBM and Tektronix terminals into a single multifunctional terminal for general office applications. Supported DEC features include complete DEC VT220 terminal emulation plus many enhancements. Supported IBM features allow comprehensive software emulation of IBM 3179 and 3180 Display Stations. These IBM features include four display modes up to 132 columns with 28 lines including a user programmable status line, IBM-styled keyboards and enhanced full color displays. Supported Tektronix features include compatible Tektronix 4105 eight color graphic and eight color character commands.

TM3000 Terminals provide the following family members:

- ☐ Color Displays.
- ☐ Amber Displays.
- ☐ Green Displays.
- ☐ Color and Monochrome 4105 Graphics.
- ☐ IBM 3179 Style Keyboards.
- ☐ Office Touch Type Keyboards.

FEATURES

Enhanced Display Features

- ☐ 14″ anti-glare display for larger viewing area.
- ☐ Displays up to 3696 characters.
- ☐ Displays 347 unique character symbols.
- ☐ Five display modes (column x line): 132 x 28, 132 x 25, 80 x 25, 80 x 33* and 80 x 44*.
- ☐ Character attributes of underline, intensity, blinking, reverse video and eraseable.
- ☐ Dynamic focus for clear readable characters.
- ☐ Special host programmable user line.
- ☐ Special symbols including IBM 3179 status line symbols.
- ☐ Three speed smooth scroll rate for convenient reading of reports.
- ☐ 10 x 12 character cell in 80 column mode.
- ☐ Character matrix supports 2 dot lowercase descenders.

(*) 33 and 44 line display use automatic vertical scrolling.

Enhanced Color Features

TM3000 Color Terminals support these additional features:

- ☐ Seven colors: Sky blue, red, magenta, green, cyan, yellow, white plus black.
- ☐ BW3 phosphors for brighter blues.

- ☐ Non-embedded attributes per character: background: 1 of 8 colors, foreground: 1 of 8 colors plus bold, blink, underlining and eraseable.
- ☐ Optional graphics supports 8 colors from a palette of 64 colors.
- ☐ Enhanced contrast glass reduces light reflections.
- ☐ Dot pitch of .31mm.

Enhanced Operational Features

- ☐ Concurrent transparent printing protocol allows convenient locally attached printer on the RS-232 printer port to print host data while the display displays and the user types.
- ☐ Standard keylock.
- ☐ Tilt-and-swivel base.
- ☐ Low-profile 30mm DIN keyboard adjustable to three levels.
- ☐ Other terminal emulations: DEC VT220, VT100 and VT52.
- ☐ RS-232 (up to 300') and RS-422 (up to 2000') host port interfaces.
- ☐ Selectable port rates up to 19,200 bps.

Graphics Features

- ☐ Field upgradable option.
- ☐ Supports Tektronix 4105 compatible graphic commands.
- ☐ Color display supports 8 colors per pixel selected from a palette of 64 colors.
- ☐ Monochrome display supports 8 gray levels per pixel.
- ☐ 640 x 300 graphic display.
- ☐ Selectable character and graphic displays.
- ☐ Character display may overlap graphic display.

DEC VT220 Compatible Features

- ☐ Emulates: DEC VT220, DEC VT100/102 and DEC VT52.
- ☐ Programmable function keys.
- ☐ Downloadable character sets.
- ☐ 80 and 132 column displays.
- ☐ Compose character key.
- ☐ Split screen smooth scrolling.
- ☐ 13 resident national character sets: North American, United Kingdom, Canada (French), Denmark, Finland, France (Belgium), Germany/Austria, Holland, Italy, Norway, Spain, Sweden and Switzerland.
- ☐ Line graphics and other special symbols.

Keyboard Features

- ☐ Office Keyboard: 107 key typewriter styled keyboard layout for efficient touch typing and general office application usage.
- ☐ IBM Styled Keyboard: 122 key keyboard styled after IBM 3180 and 3179 Display Station's keyboard for efficient data entry operations and for compatible operations.
- ☐ Word processing and data processing key legends.
- ☐ Extra Escape and Line Feed keys.
- ☐ Low-profile 30mm DIN keyboard.
- ☐ Adjustable to three levels (5, 10 and 15 degrees).
- ☐ Detached with coiled cord.
- ☐ Step sculptured keys.
- ☐ Selectable on/off audible keyclick.
- ☐ Separate numeric keypad.
- ☐ Separate editing keypad.
- ☐ Compose character key.

Print Features

- ☐ Concurrent transparent print.
- ☐ Transparent print.
- ☐ Print screen.
- ☐ Auto print.
- ☐ Print cursor line.

Operation Modes

- ☐ Monitor.
- ☐ Full duplex.
- ☐ Selectable local echo.
- ☐ Conversational local.
- ☐ Set-up (Menu-driven, non-volatile).

SPECIFICATIONS

Display

Viewable Display Area: Horizontal: 240mm, Vertical: 180mm ± 5mm.

Display Mode (Column x Line)	Display Type	Character Cell
80 x 25	Monochrome	10 x 12
132 x 25	Monochrome	8 x 12
132 x 28	Monochrome	8 x 11
80 x 25	Color	10 x 12
132 x 25	Color	7 x 12
132 x 28	Color	7 x 11

Physical

Monitors:	Monochrome	Color
Height:	14.1" (358mm)	15.6" (395mm)
Depth:	13.7" (348mm)	16.2" (411mm)
Width:	13.3" (338mm)	15.3" (388mm)
Weight:	25 lbs (11.3 kg)	39.8 lbs (18.1 kg)
Articulation:	Tilt: +5 degrees, −20 degrees. Rotation: ± 45 degrees.	

Keyboards:	
Height:	1.2" (30mm)
Depth:	7.4" (188mm)
Width:	20.4" (519mm)
Weight:	3.3 lbs. (1.5 kg)

Power

	Monochrome	Color
RMS Power:	30 watts	64 watts
Heat Output:	102 Btu/hour (26 Kcal/hour)	242 Btu/hour (61 Kcal/hour)
Rated current:		
60 Hz, 110VAC:	1 amp.	2 amp.
50 Hz, 220VAC:	.5 amp.	1 amp.

Safety
Meets UL 478 (EDP) and 114 (Office Equipment)
Meets CSA 154 (EDP) and 143 (Office Equipment)
Meets VDE 0806/8.81 (Office Equipment)
Meets IEC 380 (Office Equipment)

Emissions
Meets VDE 0871/6.78, Class A
Meets FCC Part 15, sub-part J, Class A

ESD
12,000 Volts: No operator perceived errors.
24,000 Volts: No permanent equipment damage.

Altitude
Operating: 0 to 10,000 feet.
Non-Operating: 0 to 20,000 feet.

Temperature
Operating: 5°C to 40°C.
Non-Operating: −40°C to 60°C (Storage).

Relative Humidity
Operating: 20% to 80% RH non-condensing.
Non-Operating: 10% to 90% RH non-condensing.

Transportation
Packaging and shipping containers and procedures comply with the current NSTA preship test procedures.

MOTOROLA

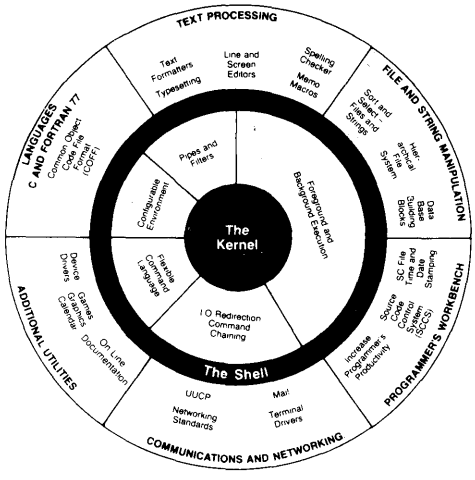
VME Delta Series

SYSTEM V/68 OPERATING SYSTEM SOFTWARE FEATURES

Advance Information

HIGHLIGHTS

- ☐ **Compliant with AT&T's UNIX® System V interface definition**
- ☐ **Object code compatible with SYSTEM V/68 Release 2**
- ☐ **Transparent sharing of data and peripheral devices across a network**
- ☐ **Applications can be run independent of the underlying network**
- ☐ **Virtual memory provides optimum utilization of physical memory**
- ☐ **Fast file access and reliable file system**
- ☐ **Efficient use of disk space and memory through shared libraries**
- ☐ **Supports MC68881 floating point coprocessor**



GENERAL DESCRIPTION

The SYSTEM V/68 Release 3 Operating System is the standard operating system for the M68000 family of microprocessors. It affords users a powerful computing environment for developing, executing and supporting technical, commercial and network applications.

Compliant with AT&T's UNIX System V Interface Definition (SVID), SYSTEM V/68 Release 3 provides users with the ability to share data, applications and resources among different computers transparently. Release 3 is object code compatible with SYSTEM V/68 Release 2. It offers an efficient kernel which provides process scheduling and I/O facilities to all programs. In addition, a powerful command processor, the Shell, provides for interactive control. An extensive set of tools and utilities provide support for program development, text processing, electronic mail and system to system communication.

OVERVIEW

The Kernel

The kernel supports a multitasking multiuser software development and OEM application execution environment. As the heart of the system, the kernel schedules tasks and manages system resources. It acts as a dispatcher of tasks competing for the system resources and as a traffic coordinator for the data flowing through the system while managing and allocating memory as required.

Virtual Memory

The demand paged, virtual memory feature of SYSTEM V/68 Release 3 has several advantages over swapping:

- ☐ Programs larger than physical memory can be executed.
- ☐ Only the portions of a program which are actually being used reside in memory at any particular time.
- ☐ External fragmentation is completely eliminated.
- ☐ 32 Megabyte process space.
- ☐ Two level page table.

Berkeley Signals

A new set of system calls is provided in Release 3 of the SYSTEM V/68 Operating System which resemble those of the Berkeley 4.2 BSD signal interface. The new implementation provides enhancements to the existing SYSTEM V/68 interprocess signalling mechanism while preserving the SYSTEM V/68 signal semantics.

Process Fork Mechanism

This feature allows processes to "fork" without duplicating the data area of the parent. It is commonly referred to as "copy-on-write" and essentially allows the child process to share the parent's data area. Modified pages are duplicated for each process before the access takes place. This feature provides the performance improvement of the Berkeley 4.2 BSD mechanism without affecting compatibility with the SYSTEM V/68 fork system call interface.

The Shell

The shell interprets user commands and calls system tasks to perform the requested operations. It accepts commands one at a time or in a series called a pipe, where output from one job becomes input to another job.

Features of the shell include: control-flow primitives, parameter passing, variables and string substitution. In addition, constructs such as *while, if-then-else, case* and *for* are available. Two-way communication is also possible between the shell and the commands. String-valued parameters, typically file names or flags may be passed to a command. A return code is set by commands that may be used to determine control-flow, and the standard output from a command may be used as shell input.

The shell can modify the environment in which the user command runs. Input and output can be redirected to files and processes that communicate through pipes can be invoked. Commands are found by searching directories in the file system in a sequence that can be defined by the user. Commands can be read either from the workstation or from a file, which allows command procedures to be stored for later use.

The SYSTEM V/68 File System

The SYSTEM V/68 file system comprises a uniform set of files and directories organized as a hierarchical, tree-like file structure of arbitrary size. Branching from the beginning of the file system, also known as the root directory, are several system directories, including the user directory. A directory can contain any number of subdirectories and files, which allows for almost any kind of branching structure that is required for the filing system. Major features of the file system are:

- ☐ 1024 byte file block size.
- ☐ Simple and coherent naming conventions.
- ☐ File linking across directory boundaries.
- ☐ Automatic file space allocation and de-allocation.
- ☐ Mountable file systems.
- ☐ Hashed 1-node file and directory look up.
- ☐ Device-independence: each physical I/O device is treated like a file, thus allowing for uniform file and device I/O.
- ☐ Flexible directory and file protection modes: allows for the setting of "read," "write" and "executive" authority separately for the owner, for a group of users and for all other users. These protection modes are altered dynamically.
- ☐ Facilities are provided for creating, moving, accessing and processing groups of or single files and directories.

Optimal block allocation and de-allocation provides for fast access of files where data resides in the disk cache. Additionally, a configurable feature is included in Release 3 which causes all writes to the file system to be synchronized with the disk. This configurable feature reduces the likelihood of lost user data in the event of a system crash and improves the robustness and reliability of the file system.

Remote File Sharing

The Remote File Sharing (RFS) feature allows users to share files, data and peripheral devices, such as tape drivers and printers, transparently across different computers on a network.

Existing applications can be installed and executed in an RFS environment without any modifications and these programs can access remote files. The full file system semantics, including Named Pipes and File and Record Locking, are preserved when accessing remote resources. File and Record Locking preserves data integrity by protecting files from simultaneous access by two or more users.

Extensive administrative tools allow each computer on the network to control the specific resources it makes available for sharing as well as the specific remote resources it makes available to local users. Security features are provided to enable a local system to accept or deny access to local resources on a per remote system and per remote user basis.

Streams

The Streams mechanism provides network protocol and media independence by allowing applications software to be independent of the underlying network. A change in the transmission medium or protocol can be accommodated by substituting Streams modules without modifying the applications software. Streams modules can be combined to perform sophisticated network services and the same modules can be used over different media and in different network architectures.

In addition to modularity, Streams provides the following facilities:

- ☐ Simple User Interface
- ☐ Buffer Management
- ☐ Flow Control
- ☐ Scheduling of Streams Protocol Modules
- ☐ Efficient Message Passing
- ☐ Multiplexing
- ☐ Error and Trace Loggers

The Streams mechanism is implemented as an independent subsystem which allows protocols implemented using Streams primitives to be easily ported or migrated. This allows protocols to be implemented in a way that is independent of the underlying machine architecture.

Transport Level Interface Library

The Transport Level Interface (TLI) along with the Transport Provider Interface (TPI) provide a specific mechanism that allows a user to run applications independent of the underlying network. An application written using the TLI Library will work without modification over any network implemented according to the TPI specification.

The traditional *uucp* commands have been modified to use the TLI Library. The changes make the uucp family independent of the underlying network and usable across all Streams-based transport providers. This change eliminates the need for different file transfer commands for different networks.

Communications and Network Support

The SYSTEM V/68 Operating System provides support for electronic mail, communications and networking. Electronic mail allows users to communicate with one another using the system as a mail box or as a bulletin board. The communications utilities allow a SYSTEM V/68 user to communicate to mainframe computers. The tools and utilities supported under SYSTEM V/68 Release 3 include:

- ☐ uucp, UNIX-to-UNIX copy program, is a utility used for communication between UNIX systems.
- ☐ cu is a utility used to call up another UNIX system.
- ☐ mail sends mail and reads mail sent by users to other users within a system.
- ☐ interprocess functionality, including shared memory, messages and semaphores.

The data communications products supported under SYSTEM V/68 Release 3 include:

- ☐ 3274 SNA and BSC
- ☐ 3776 SNA
- ☐ RJE/HASP, 2780 and 3780
- ☐ X.25 with TCP/IP

Networking support allows several computers to be linked together, either through dedicated links or by dial-up connections. Ethernet Local Area Networking (LAN) support with XNS or TCP/IP is provided by SYSTEM V/68. So that the software can be easily upgraded to other protocols, each module corresponds to a specific layer of the OSI interconnection model.

SOFTWARE DEVELOPMENT TOOLS

Languages

Provided as an integral part of SYSTEM V/68 are the C language compiler, "cc" and the High Level C Code Optimizer. The C language has established itself as one of the most popular and widely used commercially supported programming languages, and produces portable application software. The optimizer is an added phase for the compilers derived from the MC68000 Software Generation System (SGS). It aids the code generator to produce code which results in 10% to 20% performance improvements.

The SYSTEM V/68 Operating System also provides support for the FORTRAN language by including a FORTRAN 77 compiler "f77," a rational (structured) FORTRAN preprocessor "ratfor," and an extended FORTRAN language preprocessor "ef1."

Floating point support is also provided for both the C and FORTRAN 77 compilers. The MC68881 Coprocessor provides the following features:

- ☐ Eight general purpose floating data registers, each supporting a full 80-bit extended-precision real data format.
- ☐ A 67-bit Arithmetic Logical Unit to allow very fast calculations.
- ☐ A 67-bit barrel shifter for high-speed shifting operations.
- ☐ Fully conforms to IEEE P754 Standard (draft 10.0) plus a full set of trigonometric and logarithmic functions.
- ☐ Overlapped instruction execution enhances throughput while maintaining the programmer's model of sequential instruction execution.

An MC68000 assembler, "as," and a linker/loader, "ld," complements the C and FORTRAN language compilers.

Specifically designed to make software development easier, the SYSTEM V/68 programming environment also provides a set of software tools to aid in efficient and professional tuning and maintenance of applications software. These tools include the following:

- ☐ *cflow* is a tool that generates a C flow graph by analyzing a collection of C, yacc, lex, assembler and object files and builds a graph charting the external references.

- *sdb* is a symbolic debugger.
- *prof* is a code profiler that provides, for external symbols, the percentage of time spent between a given symbol and the next, together with the number of times that functions are called.
- *size* is a command that provides section size information for the text, data and bss (uninitialized data) sections of the common object files. The total size for the module is also provided.
- *nm* is a command that prints the symbol table of each common object file filename.
- *ar* is the archive and library maintainer for portable archives.
- *make* is a tool to maintain, update and regenerate programs; make only updates a module if it depends on modules that are newer than it.
- *dump* is a command that provides a dump of selected parts of an object file.
- *lint* is a checker for C programs. It attempts to detect features of C programs that are likely to be bugs, nonportable or wasteful.
- *cb* is a C program beautifier that produces code with spacing and indentation that displays the structure of the code. cb "canonicalizes" the code to the style of Kernighan and Richie.
- *cxref* is a tool that analyzes a collection of C files and builds a cross reference table.

Programmer's Workbench

The Programmer's Workbench is a collection of tools that support the development of large systems of software in a professional and automatic manner. In particular, this includes Source Code Control System (SCCS) which provides the facility to store, update and retrieve all versions, past and present of source code modules.

Text Processing

Support for text processing under the SYSTEM V/68 Operating System is provided by the inclusion of four editors: vi, ed, ex and sed. Ex and vi are full screen editors, ed is a line editor and sed is a non-interactive stream oriented editor.

The vi and ex editors support a large number of existing terminal types through the use of the terminal database "terminfo." Additional entries to terminfo can be added by the user to support other terminals.

System Maintenance

Ram-Based Logical Disk allows Release 3 to be booted from tape and run with a memory-based file system in order to repair a disk or install software on an empty disk. This enhancement allows use of selected system utilities without the use of a hard disk while preserving the standard SYSTEM V/68 user interface. This capability is intended to facilitate system maintenance and installation.

Backup and Restore Facility

The Backup and Restore Facility is enhanced with a menu interface. This facility provides attended full and incremental backups, scheduling of unattended incremental backups, selective backup/restore of directories, files and sub-trees as well as selective backup/restore of files owned by user.

Menu Interface

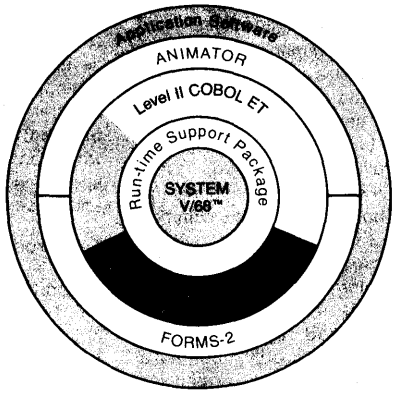
A menu-based interface package is provided in SYSTEM V/68 Release 3. This interface, designed to simplify system administration functions, also serves as a helpful front end for interfacing to SYSTEM V/68 utility programs. This facility allows command lines to be built interactively and teaches the user command options. As such, this feature is useful for the novice user and the experienced user can utilize this facility to create and modify menus as well.

MOTOROLA

PBC104

VME Delta Series

MICRO FOCUS LEVEL II COBOL ENHANCED TECHNOLOGY™



HIGHLIGHTS

- ☐ **GSA certified to the Federal High Level COBOL Standard**
- ☐ **Super-high performance execution**
- ☐ **Powerful file handling**
- ☐ **Program performance analysis**
- ☐ **Optimized memory utilization**
- ☐ **Powerful development tools**

GENERAL DESCRIPTION

The Micro Focus LEVEL II COBOL Enhanced Technology™ compiler allows large existing ANSI '74 COBOL applications written for minicomputers and mainframes to be transferred to VME Series 20™ computer systems with minimal code modification. It uses advanced code generation techniques which lead to very high execution speeds, permitting the VME Series 20 computer system to equal the performance of a VAX® computer running COBOL. Complementing the LEVEL II ET compiler is a set of development tools, namely, the ANIMATOR™ debugging facility, the FORMS-2™ forms generator and UPGRADE-3™, a RM/COBOL® to LEVEL II ET source code conversion utility.

The Micro Focus LEVEL II COBOL ET compiler meets the Federal High Level ANSI '74 COBOL Standard (with zero error established) by the U.S. Government General Services Administration (GSA) and also conforms to the COBOL standard defined by the X/OPEN Group of European Computer Manufacturers.

For application developers, Micro Focus LEVEL II COBOL ET permits COBOL application programs written for most mainframe and minicomputers to be ported onto the VME Series 20 computer systems operating under the SYSTEM V/68™ Operating System. (The SYSTEM V/68 Operating System is derived from AT&T's UNIX® System V.) The available application software consists not only of the large base of LEVEL II COBOL applications, but also other COBOLs such as RM/COBOL. Additionally, the LEVEL II COBOL ET application development environment increases programming productivity by allowing programmers to create, modify and maintain more applications faster and with greater quality than otherwise possible. The fast compile speed of the LEVEL II COBOL ET package reduces the time spent during the debugging cycle because the programmer typically needs to compile several times in order to stabilize the program. This allows VME Series 20 computer systems to be used as cost-effective, highly productive application development and maintenance stations at data processing mainframe installations.

For end-users, Micro Focus LEVEL II COBOL ET provides access to a wide range of existing application packages written in LEVEL II COBOL. This permits the VME Series 20 computer systems to be employed as departmental systems using the existing suite of mainframe COBOL application programs.

DESCRIPTION

The Micro Focus LEVEL II COBOL ET Compiler supports all the features included in the Federal High Level Standard and contains extensions to complement and take advantage of the interactive benefits and capabilities of the VME Series 20 computer systems operating under the SYSTEM V/68 Operating System. Significant attributes include:

GSA High Level Conformity

With zero error, the LEVEL II COBOL ET compiler meets ANSI X3.23 1974 COBOL to the Federal High Level, the highest level of COBOL implementation defined by the U.S. Government.

The GSA test consists of 225,000 lines of COBOL program source code that comprises 5,500 different tests designed to ensure that the standard is followed absolutely. This compiler is the only UNIX based system COBOL compiler to have passed these tests with zero error.

Enhanced Interactive Screen Handling

Extensions beyond ANSI '74 COBOL to the ACCEPT and DISPLAY verbs facilitate more flexible user-configurable screen handling. Extensions include cursor addressing, protected fields, automatic zero suppression, calculator style input with function key handling and highlighting, data verification and cursor position detection. The extensions enhance the ANSI standard to allow COBOL programmers to handle screen Input/Output.

File Support

Four file formats are supported by the compiler: Relative, Sequential, Line Sequential and Indexed Sequential.

Relative files allow random and sequential access to records in a mass storage file. Each record in a relative file is identified by an integer value greater than zero which specifies the record's position in the file.

Sequential and Line Sequential files allow records to be accessed sequentially in the order they were written. The Line Sequential structure is an extension to ANSI '74 COBOL which allows for reading files created by a text editor.

Line Advancing is a feature that allows the user to create files for printing. These files have the necessary control characters to print on a standard printer.

Indexed Sequential file handling is implemented through a fast B-tree structure that is self-balancing. It allows use of alternate keys with duplicates. Records can be accessed randomly or sequentially. Each record in an indexed file is uniquely identified by the value of one key within the record, and can have up to 63 alternate keys. A data record and its key can be deleted from a file. Its space in the index and data files is then made available for another record to be added. The maximum file size can be up to 1 Gbyte, limited only by the disk size. Index key length extends to 120 bytes, and the maximum record length is 8191 bytes. The file handling facility allows data files to expand as required by the application. This ensures that disk space is used efficiently.

Cross Reference Listing

The Micro Focus LEVEL II COBOL ET compiler's cross reference facility provides users and programmers with an additional compiler listing to aid debugging.

Large Program Size Allowed

A LEVEL II COBOL ET program can include multiple segmented procedural code of 16 Mbytes per segment (up to 800 Mbytes of code). In addition, the programmer can arrange to dynamically load as many as 100 programs (segmented and/or non-segmented). There are virtually no limitations on the table size, an addressable data item can be as large as 256 Mbytes.

Enhanced Multi-User Support

Multi-user support is available with the C-ISAM® package which allows file sharing and record locking for indexed files. The LEVEL II COBOL ET communication module is fully implemented to the GSA standard. The LEVEL II ET compiler also supports file locking for sequential, relative and line sequential files and record locking for sequential and relative files.

Runtime Specification of File Names

The LEVEL II COBOL ET package allows programmers and users to specify external file names, program names and UNIX pipes dynamically at run time. For example, the printer spooler can be specified directly.

Fast Compilation

The LEVEL II COBOL ET compiler produces intermediate code from the programmer's source in a single pass. This allows very fast compilation, typically more than 1000 lines per minute. Testing and debugging is carried out using this intermediate code, thus keeping the development cycle as short as possible. Finally, when the code is fully debugged the intermediate code is converted to machine code just once using the appropriate native code generator.

Optimized Memory Utilization

Optimized memory utilization eliminates the need for each user to have a copy of the development software in memory. This capability frees valuable memory space, thereby increasing system performance.

Performance Analysis Tool

A programmer may use a compiler directive that allows the creation of a file dynamically at runtime which will contain statistical data reflecting the overall usage of a paragraph and/or section. This data can be accessed by an analysis utility to provide specific area(s) that may be enhanced to increase the speed and performance of the application.

Full UNIX Support

LEVEL II COBOL ET supports all SYSTEM V/68 Operating System features including terminal handling, directory handling, CALLs to C subroutines, UNIX utilities, programs in other programming languages, file handling, performance analysis and dynamic memory capability.

Source Maintenance Flexibility

Source code can be kept in other variations of COBOL, such as RM/COBOL and passed through a very fast pre-processor during every program compilation, or the program can be converted a single time to Micro Focus LEVEL II COBOL ET source code by using the pre-processor as a converter. This gives the application programmer the option of keeping and maintaining the source code in either dialect of COBOL.

Powerful Development Tools

Complementing the LEVEL II COBOL ET compiler is a powerful set of (optional) development tools: the ANIMATOR source code level program analysis and debugging tool to help debug and maintain programs and the FORMS-2 source code generator which automates the writing of interactive screen display programs.

ANIMATOR allows direct observation and control of a program's execution, statement by statement at the source code level. In addition, it allows the programmer, at any point during the run, to "break in" and halt the execution of the program. Access is then provided to a main menu of commands that allow interrogation and modification of data values, alteration of the control flow, replay of any part of an execution, "what-if" analysis and many other program analysis and debugging capabilities. The ANIMATOR software speeds up the debugging process of new programs, but it also makes it easier to understand and maintain existing programs.

The FORMS-2 generator is a powerful visual programming tool designed to speed the creation of interactive screen handling programs. The FORMS-2 product is extremely user-friendly. It allows the programmer to use the keyboard to "paint" a form directly on the screen, exactly as it will appear to the user at runtime. The FORMS-2 software then automatically generates the COBOL source code to handle the screen.

PBC104

SPECIFICATIONS

Verbs

ACCEPT	—AT
	—FROM CRT
	—FROM CONSOLE
ADD	—TO
	—GIVING
	—CORRESPONDING
ALTER	
CALL	—USING
CANCEL	
CLOSE	
COMPUTER	
COPY	—
	—REPLACING
DELETE	
DISABLE	
DISPLAY	—AT
	—UPON CRT
	—UPON CONSOLE
DIVIDE	—INTO
	—ROUNDED
	—BY
	—REMAINDER
ENABLE	
ENTER	
EXIT	—
	—PROGRAM
GO	—TO
	—DEPENDING
IF	—AND/OR
	—ELSE
	—NEXT SEQUENCE
INSPECT	—TALLYING
	—REPLACING
	—TALLYING/REPLACING
MERGE	—ASCENDING
	—DESCENDING
MOVE	—
	—CORRESPONDING
MULTIPLY	
OPEN	—INPUT
	—OUTPUT
	—I-O
	—EXTEND
PERFORM	—THROUGH
	—THROUGH TIMES
	—THROUGH UNTIL
	—THROUGH VARYING
READ	—INTO
	—NEXT
RECEIVE	
RELEASE	
RETURN	
REWRITE	
SEARCH	—VARYING
	—ALL
SEND	
SET	—TO
	—UP BY
	—DOWN BY
SORT	—ASCENDING
	—DESCENDING
STOP	
STRING	
SUBTRACT	—FROM ROUNDED
	—FROM GIVING
UNSTRING	
USE	—AFTER
	—DEBUGGING PROCEDURE
	—DEBUGGING DATA
WRITE	—FROM
	—BEFORE
	—AFTER

File Types
Relative
Sequential
Line Sequential
Indexed Sequential

Access Mode
Dynamic
Random
Sequential

Conditions
ALPHABETIC
AND/OR
EQUAL
GREATER/LESS
NOT
NUMERIC
POSITIVE/NEGATIVE
ZERO

DOCUMENTATION

The following manuals form the documentation set for the Micro Focus LEVEL II COBOL Enhanced Technology product:

Language Reference Manual
LEVEL II COBOL ET Operating Guide
Getting Started Manual
ANIMATOR Operating Guide (Optional)
FORMS-2 Operating Guide (Optional)
The Error Message Manual
The UPGRADE-3 Operating Guide

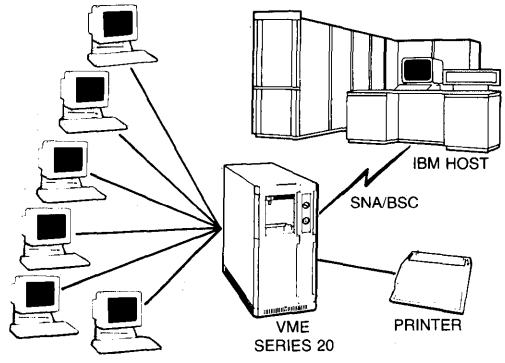
MOTOROLA

VME Delta Series

SNA AND BSC COMMUNICATION PRODUCTS

HIGHLIGHTS

- ☐ **3274 SNA Interactive**
- ☐ **3770 SNA RJE**
- ☐ **3274 BSC Interactive**
- ☐ **2780 RJE**
- ☐ **3780 RJE**
- ☐ **HASP RJE**



GENERAL DESCRIPTION

The VME Series 20 supports a broad offering of System Network Architecture (SNA) and Binary Synchronous (BSC) communication facilities which run with the SYSTEM V/68™ Operation System. Both interactive and batch protocol products are supported on the Series 20. All protocol products are supported by an Intelligent Communications Controller which uses a Motorola 68010 microprocessor and features 512K bytes of dynamic RAM to offload the main system application processor from supporting the communication protocol.

3274 SNA Description

The 3274 SNA product provides for 3274 type commands, orders, field and character attributes, attentions, control characters and address sequences. The following IBM 3274 devices are emulated:

- ☐ 3274 Model 51C Control Unit
- ☐ 3278 Model 2 and 5 Display Stations
- ☐ 3287 Model 2 Printer (DCS and SCS modes)
- ☐ 3279 Models 2A and 2B Display Stations

Attributes of the IBM 3278 Model 2 Display Station are supported. These include protect, non-display, high intensity and numeric. Attributes for extended highlighting and extended color are also supported. The workstations display 24 lines with 80 characters per line or 27 lines with 132 characters per line.

3274 SNA Features

- ☐ EBCDIC character set
- ☐ Leased and dial line support
- ☐ 1200–9600 bps over half-duplex facilities
- ☐ 24 Programmable Function (PF) codes
- ☐ User Menu
- ☐ Multiple sessions on each terminal
- ☐ System status line

3770 SNA Description

The 3770 SNA RJE product enables the user to transfer batch data to and from an IBM SNA host. The system supports unattended operation for the transfer of files. The user is provided with system status capability. For example, record counts may be displayed on the terminal. Card reader files and console keyboard input data are accepted. Output device files include printer and card punch files.

3770 Features

- ☐ Compatibility with IBM 3776 or 3777 Model 3 or 4, SNA MLU (Multiple Logic Unit devices)
- ☐ Unattended operation
- ☐ Leased or switched line support
- ☐ Up to 9600 bps, half-duplex
- ☐ Console support
- ☐ Space compression and expansion
- ☐ Transparency
- ☐ Decompaction of host to terminal data streams
- ☐ Spooled and direct print capabilities

3274 BSC Description

The 3274 Binary Synchronous Com-

munications program emulates the 3274 BSC system. It provides support for 3274 type commands, orders, field and character attributes, attentions, control characters and address sequences. The following IBM 3274 devices are emulated:

- ☐ 3274 Model 51C Control unit
- ☐ 3278 Model 2 and 5 Display Stations
- ☐ 3287 Model 2 Printer
- ☐ 3279 Model 2A and Model 2B Display Stations

Attributes of the IBM 3278/3279 Display Stations are supported. These include protect, non-display, high intensity and numeric. Attributes for extended highlighting and extended color are also supported. The workstations display 24 lines with 80 characters per line or 27 lines with 132 characters per line.

3274 BSC Features

- ☐ Leased and dial-up line support
- ☐ 1200–9600 bps over half-duplex facilities
- ☐ EBCDIC character sets
- ☐ 24 Programmable Function (PF) codes
- ☐ System status line
- ☐ User Menu
- ☐ Multiple sessions on each terminal

2780/3780/HASP RJE Description

These batch transmission products enable the user to communicate to another system or to a host computer using standard IBM 2780/3780/HASP RJE protocols. A range of printers is available to support hard copy output. Unattended operation allows the system to send and receive files without operator intervention. Leased or dial-up capability supports from 1200 bps to 9600 bps over half-duplex, leased or private (non-switched) lines and from 1200 bps to 4800 bps over half-duplex dial (switched) lines.

The 3780/HASP space compression and expansion features are supported. Consecutive spaces are compressed from the data prior to transmission and consecutive spaces are inserted as data is received.

Transparency allows any eight-bit data configuration to be sent or received. This enables object code and binary files to be transmitted to another system.

2780/3780/HASP RJE Features

- ☐ I/O devices include card reader, card punch and printer files
- ☐ Unattended operation (auto answer and auto disconnect)
- ☐ Leased and switched line support (1200–9600 bps)
- ☐ Space compression and expansion (3780 and HASP)
- ☐ Transparency
- ☐ Point-to-point configuration
- ☐ Console support

PBC106

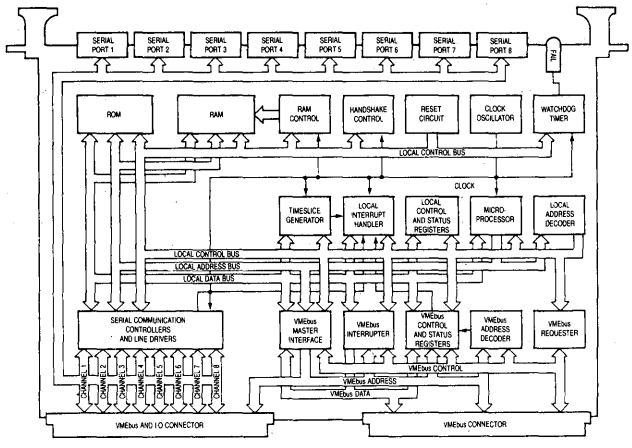
MOTOROLA

VME Delta Series

8-PORT SERIAL CONTROLLER

HIGHLIGHTS

- ☐ **Eight Asynchronous RS-232-C Ports**
- ☐ **Up to 4 Controllers per System**
- ☐ **Up to 19,200 bps on all Ports**
- ☐ **10 MHz MC68010 Microprocessor**
- ☐ **VMEbus Compatible with Double-high Eurocard Form Factor**
- ☐ **Buffered Pipe Communications Protocol**
- ☐ **Driver for Motorola SYSTEM V/68 Operating System Available**



GENERAL DESCRIPTION

The 8-Port Serial Controller provides the VME Series 20 with an eight channel asynchronous interface between its VMEbus and serial peripheral devices. Built around a MC68010 microprocessor, the controller improves the speed of data transfer and off-loads the system's central processor from serial communications controlling tasks. It handles I/O related functions without having to interact with the central processor, thereby increasing overall system performance.

Each serial port can operate individually at speeds up to 19,200 bps.

Typical applications for the 8-Port Serial Controller include connections to asynchronous devices such as terminals, printers and modems.

HARDWARE OVERVIEW

Controller Microprocessor and Memory

A powerful MC68010 running at 10 MHz is the microprocessor on the controller. This processor controls all connected devices through local data, address and control buses.

Local memory includes 32Kb of EPROM and 128Kb of dynamic RAM. The EPROMs contain the firmware resident ICC program that is copied to and executed from RAM after power-up reset. The RAM also holds the intermediate I/O data to be processed and transferred.

VMEbus Interface

The MC68010 microprocessor accesses VME system memory through a VMEbus master interface with software selectable address modifier codes. VMEbus is the leading 32-bit industry standard bus. This important fact gives systems integrators and OEMs the flexibility to build high performance systems for many different types of applications.

Peripheral Interface

Four serial communication controllers are utilized, each providing two independent channels. These controllers perform the peripheral data transfer and data link control functions at the hardware level. Port speeds, ranging from 50 to 19,200 bps, as well as data characteristics and control functions are all software selectable.

Timers

The Serial Controller has a time-slice

generator, which is a programmable counter that generates periodic interrupts to the local microprocessor. Controller firmware requires these interrupts for its time-slice based executive concept.

The watchdog timer is a free running counter which is periodically reset by the executive. Should the executive lose control, the watchdog counter will time out and turn on the FAIL LED on the board.

SOFTWARE OVERVIEW

The 8-Port Serial Controller has dedicated firmware in local ROM for controlling eight asynchronous communications channels, each with individual character processing and control functions.

The controller has a standard I/O independent interface. It receives macro commands through a pipeline structure in system memory, which is shared by the VME Series main processor and the Serial Controller. The Serial Controller resident firmware program interprets and executes macro commands, and returns status messages through the second pipeline after completion. The transfer of commands and status messages follow Motorola's Buffered Pipe Protocol for interprocessor communications.

Character processing during input is controlled via three user tables, the Receive Translation Table (RTT), the Immediate Receive Action Table (IRAT) and the Receive Action Table (RAT).

The equivalent tables for output are the Write Action Table (WAT), Transmit Action Table (TAT) and the Transmit Translation Table (TTT).

RTT and TTT can be used to translate from one standard to another, and from lowercase to uppercase. The translation tables can also be used to accommodate keyboards encoded in different languages.

Serial Controller firmware also contains self-test routines, which are executed after system reset, as well as diagnostics.

Driver support for Motorola's System V/68 operating system (an enhanced version of AT&T's Unix System V) is available.

SPECIFICATIONS

8-PORT SERIAL CONTROLLER
- Up to 4 Controllers per system
- 8 asynchronous RS-232-C ports
- 10 MHz MC68010 microprocessor
- 128Kb RAM
- 32Kb ROM
- Port Selectable:
  - 50 to 19,200 bps
  - 1, 1.5 or 2 stop bits
  - 5 to 8 data bits
  - Odd, even or no parity
  - DTR, DSR, RTS, CTS and DCD control functions
- Buffered Pipe Protocol (BPP) for communications

ENVIRONMENTAL
Power Requirements: +5 Vdc (±5%); 2.6 A (typ), 3.2 A (max)
Operating Temperature: 0° to 55°C
Non-Operating Temperature: −40° to 100°C
Relative Humidity: 0% to 90% (non-condensing)

PHYSICAL
Height: 9.2 in. (233 mm)
Depth: 6.3 in. (160 mm)

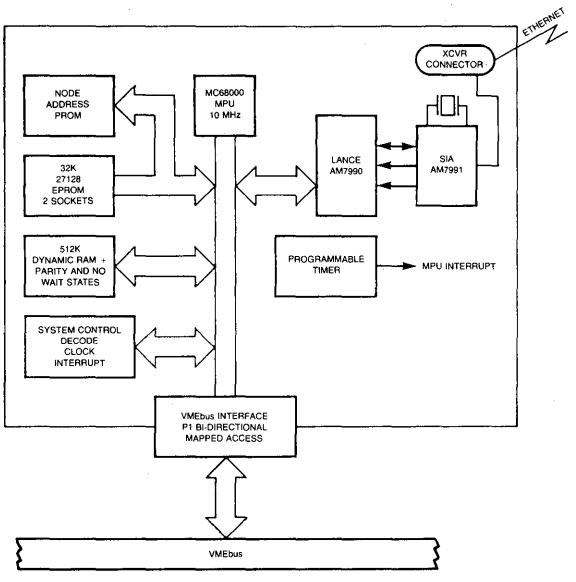
MOTOROLA

VME Delta Series

ETHERNET LAN CONTROLLER

HIGHLIGHTS

- ☐ **Ethernet, Version 2.0 Compatible**
- ☐ **Conforms to IEEE 802.3 LAN Standard**
- ☐ **Performance Up to 10 Mbits per Second**
- ☐ **10 MHz MC68000 Microprocessor**
- ☐ **512Kb RAM memory**
- ☐ **32Kb EPROM**
- ☐ **Buffered Pipe Communications Protocol**
- ☐ **VMEbus Compatible with Double-High Eurocard Form Factor**



GENERAL DESCRIPTION

The Ethernet LAN controller is an advanced communications processor, which gives the VME Series 20 family of computer systems interface capabilities to a high performance Ethernet local area network. It conforms to the IEEE 802.3 local area network specification CSMA/CD and VMEbus industry standards. The controller is also fully supported by Motorola's SYSTEM V/68 operating system (an enhanced version of AT&T's Unix System V).

The LAN controller helps to free the VME Series 20 CPU from a significant protocol burden by means of a double-high Eurocard design, incorporating an MC68000 microprocessor and a VLSI Lance Controller. Data transfer across the physical VMEbus interface between host and LAN module is accomplished using a shared memory interface.

HARDWARE OVERVIEW

Controller Microprocessor and Memory

The 10 MHz MC68000 microprocessor is responsible for moving commands and data to and from system memory, and responding to and generating VMEbus interrupts. Other functions include executing the network and transport XNS layers, providing timer functions and controlling the LANCE in its execution of the data link protocol.

The Ethernet LAN controller has 512Kb of dynamic RAM with parity. The memory is accessible from the VMEbus, LANCE and the MC68000 MPU.

For use by protocol processing software, the Ethernet LAN controller has a timer, which causes the on-board MPU to be interrupted every 2 ms.

Local Area Network Controller For Ethernet (LANCE)

The LANCE is a VLSI device used for interfacing the microprocessor system to the Ethernet local area network. It is designed to operate in an environment comprising local memory and the microprocessor. Memory serves as

the communication link between the microprocessor and the LANCE, and as a buffer for the Ethernet packets.

LANCE operates at a 10 Mbit per second data rate and is MC68000 compatible. Some of its specific features include a 16-bit data bus, a multiplexed address/data bus, a DMA controller with 24-bit addressing, and a 48-byte data buffer. The 48-byte data buffer reduces the initial response time so that once DMA is initiated between LANCE and local memory, an average of one word is transferred each 1.6 microseconds.

It also features diagnostic aids, three modes of destination address comparison, executes a CSMA/CD network access algorithm and provides extensive error reporting.

Individual packet errors reported by LANCE include CRC error, framing error, and overflow and underflow.

VMEbus Interface

The Ethernet LAN controller's interface is today's leading 32-bit bus standard, the VMEbus. Along with the double-high Eurocard form factor, the VMEbus interface allows the LAN Controller to be an important part of the VME Series 20's modularity. This modularity offers both systems integrators and OEMs a high performance Ethernet LAN Controller that will meet both present and future needs.

SOFTWARE OVERVIEW

The Ethernet LAN Controller is supplied with VME Series 20 specific software and firmware. The software supports the Motorola SYSTEM V/68 operating system, and provides a communications executive kernel which controls the LANCE hardware.

The software is highly modularized, and supports OSI's 7-layer interconnection model. Transfer of commands and status messages follow the Motorola standard Buffered Pipe Protocol (BPP).

SYSTEM V/68 Drivers

Device drivers running under the Motorola SYSTEM V/68 operating system are also provided. For further flexibility and ease of use, the bus interface between the host system and the LAN controller utilizes a shared memory protocol.

Communications Executive Kernel

The LAN controller kernel performs all functions required for controlling and monitoring the LAN controller hardware. Kernel functions include managing LANCE status registers and LANCE controlled message descriptor rings, performing timer functions, and managing interrupts and retrieving LANCE generated statistics.

The communications executive executes on the LAN controller microprocessor to supervise the LANCE chip in its processing of XNS. A VLSI Serial Input/Output Adapter provides Manchester encoding/decoding for the Ethernet interface.

Protocols for Higher Layer Software

XNS implementation is featured as higher layer software support for the LAN controller. The following XNS protocols are included:

- ☐ Internetwork Datagram Protocol — formats internet packets and performs internet addressing and routing.
- ☐ Transport Layer Protocol — resident protocol that handles echo, error, sequenced packet, packet exchange and routing information.

Self Tests and Diagnostics

Upon power up or system reset, the Ethernet LAN Controller executes a series of ROM-based self tests to determine that the controller is functioning properly. Tests include a LANCE register and loopback test, a microprocessor test, a memory test for the dynamic memory, an EPROM checksum test and a status and control register test.

SPECIFICATIONS

LAN Communications and Standards
Ethernet, Version 2.0 Compatible
Conforms to IEEE 802.3 (CSMA/CD) LAN Standard
Buffered Pipe Protocol (BPP) for Communications

Performance
Up to 10 Mbits per second

Microprocessor
10 MHz MC68000

Connectors
VMEbus: DN#41612C96
Ethernet Transceiver Port Cable: AMP745094-1

Power Requirements
3.8 A@ +5Vdc ±5%
0.6 A@ +12Vc ±5%
0.1 A@ −12Vdc ±5%

Environmental Tolerance
Operating Temperature: 5°C to 50°C
Storage Temperature: − 40°C to 85°C
Humidity Range: 0% to 95% (Non-Condensing)

Mechanical Specifications
Height x Depth (board): 9.2" (234mm) x 6.3" (160mm)
Height x Width (front panel): 10.3" (262mm) x 0.79" (20mm)

PBC108

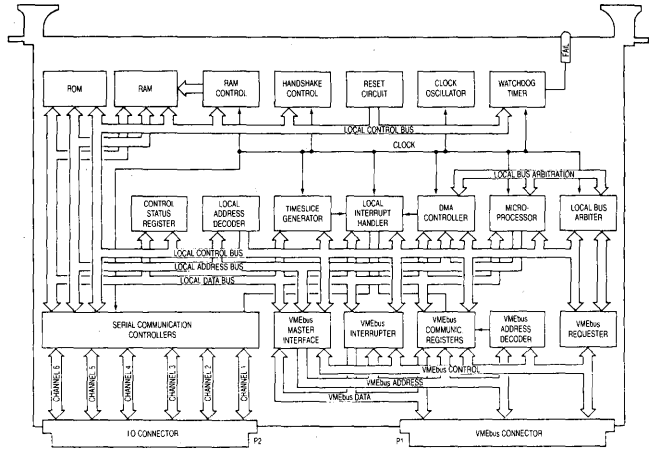
MOTOROLA

VME Delta Series

INTELLIGENT COMMUNICATIONS CONTROLLER

HIGHLIGHTS

- ☐ **Six Synchronous/ Asynchronous Communications Ports**
- ☐ **SDLC, HDLC, and BSC Protocol Support**
- ☐ **Up to 2 Controllers per System**
- ☐ **10 MHz MC68010 Microprocessor**
- ☐ **512Kb Dynamic RAM, 128Kb ROM**
- ☐ **Buffered Pipe Communications Protocol**
- ☐ **VMEbus Compatible with Double-High Eurocard Form Factor**



GENERAL DESCRIPTION

The Intelligent Communications Controller supports SDLC, HDLC and BSC and is designed for use in the high performance VME Series 20 family of computer systems. As an intelligent controller, it can handle I/O related functions locally, thus relieving the applications processor of the serial communications control functions often required in networking applications.

The Intelligent Communications Controller features control and interfaces for six high performance, asynchronous or synchronous serial ports, with a four channel DMA controller. Each serial port is port selectable and can operate from 50 to 19,200 bps.

Based on a high performance MC68010 microprocessor with 512Kb of RAM memory, the Intelligent Communications Processor has a double-high Eurocard form factor, making it a highly flexible controller for many types of applications.

HARDWARE OVERVIEW

Communication Transition Modules

Any of the six serial channels can be individually configured for RS-232-C operation using Communication Transition Modules. Each serial port can be configured independently as data circuit terminating equipment (DCE) or as data terminal equipment (DTE) by means of jumpers on the transition module.

Communications Processor and Memory

The main processing unit on the Intelligent Communications Controller is a 10 MHz, MC68010 microprocessor. It has 8 32-bit registers with a 24-bit address bus and a 16-bit data bus.

The communications processor has 512Kb of RAM memory and 128Kb of ROM memory. Data is written and read from RAM without MPU or DMAC wait states. The generation and checking of byte parity is performed automatically during RAM accesses and a detected parity error causes interrupts to the communications processor. Refresh of the dynamic RAM is automatic with local arbitration logic controlling the access/refresh cycles.

VMEbus Interface

The Intelligent Communications Controller supports the leading industry standard VMEbus. By doing so, it represents a high performance controller for integrators and OEMs who offer the VME Series 20 family of systems.

Transfer of commands and status messages follow the Motorola standard Buffered Pipe Protocol for interprocessor communications.

Two access modes are provided. The window addressing mode, in which a 4Mb local address map is used as a window to the VMEbus; and the alternate addressing mode, in which an address extension register is used to obtain mapping of a contiguous 16Mb address space.

Peripheral Interface

Four ports can be configured for DMA data transfer offering four half duplex or two full duplex DMA channels to the VMEbus. Under local DMA control, data can be transferred between system and local memory; between system memory and devices on the serial channel; and between local memory and devices on the serial channel.

Transfer of both bit and byte data is supported, and each port can be programmed for a number of different synchronization modes. Both monosynchronous and bisynchronous modes, with or without CRC generation and checking, can be used for byte data transfer.

The SDLC and HDLC protocol with automatic CRC generation and checking can be used for the transfer of bit data. An underrun causes generation of an interrupt to the communications processor.

For the synchronous transfer of byte or bit data, odd, even or no parity may be programmed and NRZ, NRZI or FM coding may be selected.

When asynchronous data transfer is desired, the following parameters can be programmed: 50 to 19,200 bps baud rates; odd, even or no parity; 1, 1.5, or 2 stop bits; 5 to 8 bit character lengths; and NRZ, NRZI or FM data encoding.

SOFTWARE OVERVIEW

The Intelligent Communications Controller supports 2780/3780 BSC, 3274 SNA, 3770 SNA, 3274 BSC, and Hasp BSC communication protocols.

The controller also has power up self-test and internal diagnostic capabilities.

SPECIFICATIONS

INTELLIGENT COMMUNICATIONS CONTROLLER
- Up to 2 controllers per system
- 6 synchronous/asynchronous communications ports
- 10 MHz MC68010 microprocessor
- 512Kb RAM
- 128Kb ROM
- SDLC, HDLC and BSC protocol support
- Port selectable:
  - 50 to 19,200 bps
  - 1, 1.5, 2 stop bits
  - 5 to 8 data bits
  - Odd, even or no parity
  - NRZ, NRZI or FM encoding
- Buffered Pipe Protocol (BPP) for communications

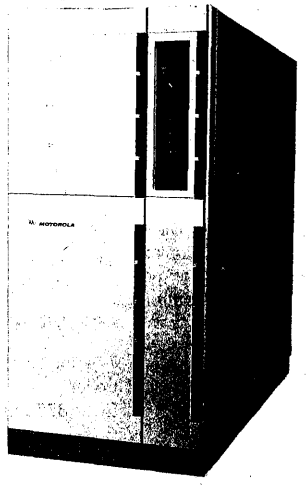
ENVIRONMENTAL
Power Requirements: +5 Vdc (±5%); 3.8 A (typ), 4.4 A (max)
Operating Temperature: 0° to 55°C inlet air, forced air cooling
Storage Temperature: −40° to 100°C
Relative Humidity: 0% to 90% (non-condensing)

PHYSICAL
Form Factor: Double-High Eurocard VME board with front panel
Board Dimensions
Height: 9.17 in. (233 mm)
Depth: 6.3 in. (160 mm)
Front Panel Dimensions
Height: 10.3 in. (262 mm)
Width: 0.79 in. (20 mm)

MOTOROLA

VME Delta Series

SMD DISK SUBSYSTEM



HIGHLIGHTS

- ☐ **337 MB to 1,348 MB of unformatted storage capacity**
- ☐ **20 milliseconds average positioning time**
- ☐ **Low acoustical noise level**
- ☐ **Drive MTBF of over 20,000 hours**
- ☐ **MC68000 based intelligent controller**
- ☐ **Zero-latency reads and writes**
- ☐ **1:1 interleave over multiple cylinder**
- ☐ **Data rates up to 24 Mbits per second**

GENERAL DESCRIPTION

The SMD Disk Subsystem includes:

- ☐ SMD disk drive
- ☐ SMD controller
- ☐ SMD Subsystem Enclosure

SMD Disk Drive

The SMD 8″ disk drive provides 337 MB of unformatted storage capacity in a compact module. The average positioning time of 20 milliseconds and the data transfer rate of 2.458 megabytes per second combine to maximize data throughput and system efficiency. A high speed version of the industry standard SMD interface is provided as part of the on-board electronics.

The SMD disk drive is highly reliable. This high reliability is assured by providing a contamination-free environment. Advanced Winchester technology simplifies the disk mechanism, and the head IC prevents read errors caused by external electrical noise. With these features, the SMD disk drive assures a mean-time-between-failures (MTBF) exceeding 20,000 power-on hours.

One or two disk drives can be configured in one enclosure for a total capacity of 674 MB. The VME Series 20 supports two SMD Disk Subsystems with four drives for a capacity of 1,348 MB using two controllers.

SMD Controller

The SMD controller supports the following important features:

- ☐ Virtual Buffer Architecture with on-board MC68000 processor
- ☐ Supports up to two SMD disk drives
- ☐ Ultra high-speed DMA with bus throttling
- ☐ On-board 32-bit ECC
- ☐ Scatter/Gather commands

The SMD controller is a powerful and intelligent controller with data rates up to 24 Mbits/sec. It utilizes the power of the Motorola MC68000 microprocessor.

The Virtual Buffer Architecture is the key to zero-latency operation. This important feature maximizes data transfer and reduces disk latencies. Memory overruns and underruns often found in less sophisticated FIFO-designs are eliminated. Using a large pool of dynamically allocated buffers, data is read as soon as the head lands on the track and then immediately transfers all sectors of interest regardless of their order on disk. This process never takes more than a single disk revolution to transfer an entire track of data.

The number of interrupts the host must process and the amount of I/O it must perform to accomplish each transaction is minimized. This frees the host

processor and the system bus for other activity.

By operating the DMA at a high speed once it is on the bus, the SMD controller preserves bus bandwidth. Thus, it minimizes the time required for each transaction thereby preserving the bus bandwidth for other system activity.

Another advantage of the virtual buffer design is that after requested sectors are read and transferred, the SMD controller will continue to read and cache sectors. When subsequent requests from the host are made for that data, it is transferred from cache without disk access.

SMD Subsystem Enclosure

The enclosure is identical in appearance as the VME Series 20 system enclosure except for the filler and operator panels. The drives are cooled by fans that provide low operating temperatures which extend the life of the drives.

Each drive is housed in a subassembly. The drive subassembly slides in and out of the enclosure. The drive can be functionally tested and inspected when slid out the front of the enclosure while still protected in its subassembly.

The operator panel provides indicators for power, ready, and fault for each drive.

DRIVE SPECIFICATIONS

Storage Capacity (unformatted): 337.10 megabytes
Storage Capacity (formatted): 269 megabytes
Disks (8-inch): 6
Heads: Read/write: 10
Servo: 1
Track capacity (unformatted): 40,960 bytes
Tracks per cylinder: 10
Cylinders: 823
Sectors (fixed): 2 or more (Max. 128)
Positioning Time: Track-to-track: 5 milliseconds
Average: 20 milliseconds
Maximum: 40 milliseconds
Average latency time: 8.3 milliseconds
Rotational speed: 3,600 rotations/minute
Recording density: 19,734 bits/inch
Track density: 683 tracks/inch
Data transfer rate: 2,458 megabytes/second
Recording code: RLL (2/7)
Interface code: NRZ
Interface: HSMD
Head positioning method: Servo-controlled track-following
Start time: 50 seconds (nominal)
Stop time: Less than 40 seconds
Mean-time-between failures (MTBF): More than 20,000 power-on hours
Error rates: Recoverable Errors: 10 per 10^{11} bits read
Unrecoverable Errors: 10 per 10^{14} bits read
Seek Errors: 10 per 10^{8} seeks

PHYSICAL SPECIFICATIONS

Height: 25.6 in. (65.0 cm)
Width: 12.5 in. (31.8 cm)
Depth: 28.0 in. (71.1 cm)
Weight: 135 lbs. (61 kg) Fully loaded

ENVIRONMENTAL AND SAFETY SPECIFICATIONS

Safety
Meets UL 478 (EDP) and 114 (Office Equipment)
Meets CSA 154 (EDP) and 143 (Office Equipment)
Meets VDE 0806/8.81 (Office Equipment)
Meets IEC 380 (Office Equipment)

Emissions
Meets VDE 0871/6.78, Class A
Meets FCC Part 15, sub-part J, Class A

ESD
5,000 Volts: No observable effect
12,000 Volts: No operator perceived errors
24,000 Volts: No permanent equipment damage

Altitude
Operating: 10,000 feet ASL
Non-Operating: 30,000 feet ASL

Acoustic Noise Level
55 dBA maximum

Temperature
Operating: 5°C to 40°C
Non-Operating: −40°C to 60°C

Relative Humidity
Operating: 10% to 80% RH, non-condensing
Non-Operating: 10% to 90% RH, non-condensing

Shock
Operating: .5 g
Non-Operating: 15 g

Transportation
Packaging and shipping containers and procedures comply with the current NSTA preship test procedures.

Shipping Container
Height: 34.5" (87.6 cm)
Width: 22.5" (57.2 cm)
Length: 41.0" (104.1 cm)

MOTOROLA

VME Delta Series

MODEL 2616 COMPUTER SYSTEM

HIGHLIGHTS

- ☐ **Motorola MC68020 processing power**
- ☐ **12-slot VME compatible enclosure**
- ☐ **2 to 12 Mb of main memory**
- ☐ **2 to 34 RS-232-C asynchronous ports**
- ☐ **85 to 728 Mb of integral Winchester disk storage**
- ☐ **Up to 1.3 Gb of external SMD capacity**
- ☐ **60 Mb streaming tape**
- ☐ **Software compliant with System V Interface Definition (SVID)**
- ☐ **Comprehensive local and remote diagnostics**

GENERAL DESCRIPTION

The VME Delta Series of 32-bit computer systems is based on the Motorola MC68020 microprocessor, the VMEbus standard, and UNIX System V Release 3. The Model 2616 uses the 16.7 MHz MC68020 and supports up to 34 serial devices. The power of the MC68020 plus the expandability and flexibility of the VMEbus architecture make the Model 2616 an ideal computer solution for commercial, technical, and networking applications.

The Model 2616's 12-slot VME card cage adapts to most application interfaces and supports large configurations. In addition, the system enclosure accommodates up to four disk drives plus a streaming tape. External SMD and tape subsystems are also supported. Both local and wide area communications protocols link the Model 2616 to industry standard networks as well as host computers.

HARDWARE OVERVIEW

The Model 2616 hardware includes:

- ☐ The system enclosure
- ☐ The VMEbus
- ☐ The applications processor and memory
- ☐ I/O and communications controllers
- ☐ Mass storage devices
- ☐ Local and remote diagnostic capabilities

System Enclosure

The VME Delta Series' floor enclosure offers a 12-slot VME compatible card cage, a 12-slot VME cable connection area, and an SMD and tape subsystem cable connection area. This provides connectivity to a diverse and large number of peripheral devices.

The unit houses a streaming tape drive and up to four 5-1/4" disk drives. One of the disk drive slots is used for half-height diskette drives when required by the configuration. All drives are held in shock-mounted drive assemblies.

The enclosure contains a 450 watt switching power supply and is cooled by filtered air.

VMEbus

The VME Delta Series uses today's leading 32-bit bus standard, the VMEbus. The VMEbus, in conjunction with the double-high Eurocard form factor, offers systems integrators and OEMs high performance as well as access to an expanding set of readily available controllers and system components. The Model 2616's 12-slot VME compatible card cage also allows the system to grow in a modular fashion to meet future as well as current requirements. In addition, upgrading the Model 2616 to a more powerful VME Delta Series model is as simple as replacing the processor and memory boards.

Main Processor

The Model 2616's main processor includes a 32-bit 16.7 MHz MC68020 microprocessor, a zero wait state cache accelerator, an MC68851 Paged Memory Management Unit (PMMU), and a floating-point coprocessor. Two RS-232-C ports are included. These are normally used for a console port and a remote service modem. The MC68020 has an on-chip 256 byte instruction cache. The processor uses the high-speed VSB for parity memory access and the VMEbus to interface I/O controllers. The MC68881 floating-point coprocessor implements the IEEE floating-point standard plus trigonometric and transcendental functions.

Memory

Up to 12 Mb of main memory are offered on the Model 2616. Both ECC and parity memory options are available.

ECC memory comes in 4 Mb increments and provides single-bit error correction and double-bit error detection. A 64-bit cache register gives fast access using the VMEbus interface.

Parity memory comes in 2 Mb increments and provides byte parity generation and error checking. These memories are dual-ported featuring concurrent processor and I/O access using the VSB and VMEbus respectively.

Disk and Tape Controllers

The Model 2616 supports controllers for 5-1/4" Winchester and floppy disk drives, 5-1/4" ESDI Winchester disk drives, 8" SMD disk drives, 1/4" streaming tape, and 1/2" 9-track tape drives. All are intelligent controllers with DMA VME system interfaces. This selection of VME compatible disk and tape alternatives allows the Model 2616 to satisfy a wide spectrum of performance and application requirements.

Intelligent Communications Controllers

VMEbus compatible controllers are offered for:

- ☐ Asynchronous devices (terminals, printers, modems, etc.)
- ☐ SNA and BSC communications
- ☐ Ethernet local area networking

Each intelligent controller uses a 10 MHz MC68000 family microprocessor.

The 8-Port Serial Controller can support eight asynchronous terminal, printer, or modem ports. Because each port can be independently configured, this controller can be adapted by the system integrator or OEM to support a wide range of serial devices. The Model 2616 supports up to four serial controllers.

Controllers

Controller	Interface	Number of Channels/Drives	Transfer Rate (per second)	Controllers Per System
Winchester & Floppy	ST-506	2	Up to 5 Mbits	2
	SA-400	2	Up to 250 Kbits	
Winchester	ESDI	4	Up to 10 Mbits	1
SMD	ESMD	2	Up to 24 Mbits	2
Streaming Tape	QIC-02	1	Up to 90 Kbytes	1
1/2" 9-track Tape	Pertec	1	Up to 1.25 Mbits	1
8-Port Serial	RS-232-C	8	Up to 19.2 Kbits	4
Communications	RS-232-C	6	Up to 19.2 Kbits	1
Ethernet	CSMA/CD	2	10 Mbits	1

Mass Storage Devices

Form-Fit	Interface	Unformatted Capacity	Formatted Capacity	Average Access Time	Drives Per System
Disk Drives					
5-1/4"	SA-400	1 Mb	655 Kb	181 ms	2
5-1/4"	ST-506	85 Mb	67 Mb	28 ms	4
5-1/4"	ESDI	182 Mb	161 Mb	16.5 ms	4
8"	ESMD	337 Mb	269 Mb	20 ms	4
Tape Drives					
5-1/4"	QIC-02	—	60 Mb	—	1
1/2" 9-track	Pertec				1
	1600 bpi	45 Mb			
	3200 bpi	90 Mb			

The Intelligent Communications Controller supports bit and byte oriented protocols like SDLC, HDLC, and BSC. Each port is independently programmable. A four-channel DMA controller is included for high performance applicatons.

The Intelligent Ethernet LAN Controller is compatible with Ethernet, Version 2.0 and conforms to the IEEE 802.3 (CSMA/CD) LAN standard. The controller interfaces with the network at 10 Mbps.

Mass Storage Devices

The base Model 2616 contains an 85 Mb or a 182 Mb 5-1/4" Winchester drive and a 5-1/4" 60 Mb streaming tape for system backup and software distribution. Some configurations also include a floppy diskette which uses a disk slot. The system can be expanded to 728 Mb of Winchester disk storage and 1348 Mb of SMD disk storage using two SMD disk subsystems.

Local and Remote Diagnostics

The VME Delta Series' system diagnostics are oriented to centralized service networks and user participatory fault identification. They enhance the system's overall reliability, serviceability, and maintainability with the following diagnostic features:

- Hardware integrity verification at system power-up and reset
- Firmware-based system tests and operating system-based confidence tests
- Menu-driven user interface with on-line help facility
- Custom test suite generation, evaluation and debug tools
- Remote test and diagnostics with integral service modem

The Model 2616's comprehensive system diagnostic facilities are designed to support very cost effective customer and product service approaches.

SOFTWARE OVERVIEW

In addition to the following software products, a large number of software applications are available from independent software vendors which will run on the VME Delta Series.

UNIX System V Operating System

SYSTEM V/68 Release 3 is Motorola's version of AT&T's UNIX System V Release 3. Release 3 includes the following new features and enhancements:

- Remote File Sharing which allows users to share files, data, and peripheral devices across computer systems attached to a local area network.
- Streams which supports application development independent of the underlying network protocols.
- Virtual memory and demand paging support for efficient memory utilization and simpler application development.
- Menu Interface for system administration functions.
- Support for the MC68881 floating-point coprocessor.
- Complete backup and restore facilities.

SYSTEM V/68 Release 3 is compliant with AT&T's System V Interface Definition (SVID).

Languages and Development Tools

- C Compiler and FORTRAN 77
- RM/COBOL
- Micro Focus Level II COBOL/ET, ANIMATOR, FORMS-2, and RunTime System
- Absoft FORTRAN
- Q-MENU Menu Development Tool

Data Management Facilities

- UNIFY Relational Data Base Management System
- Oracle Relational Data Base Management System
- Informix Relational Data Base Management System

Data Communications Facilities

- 3274 SNA and BSC
- 3776 SNA
- RJE/HASP, 2780, and 3780
- Ethernet with XNS
- Ethernet with TCP/IP

Office Applications

- Business Assistant User Services
- Q-ONE Word Processor
- 20/20 Spreadsheet
- Office Services (Electronic Mail, Calendar, etc.)
- Q-CHART Business Graphics
- MicroTrak Project Management System
- Q-FILE Information Storage and Retrieval System

SPECIFICATIONS

System
- VMEbus (IEEE P1014)
- Eurocard form factor
  - Depth: 6.30" (160 mm)
  - Height: 9.17" (233 mm)
- 96 pin DIN 41612(P1) connectors
- 12 double-high VME card slots

Processor
- MC68020 at 16.7 MHz
- MC68851 PMMU
- MC68881 floating-point coprocessor (IEEE P754)
- 16 Kb zero wait state cache accelerator option
- 2 RS-232-C asynchronous ports
  - 110 bps to 9600 bps
- VSB Memory Bus

Memory
- ECC Memory
  - 4 to 12 Mb
  - 4 Mb increments
  - 64-bit cache register
  - 560 ns 32-bit average access
  - Single-bit error correction
  - Double-bit error detection
- Parity Memory
  - 2 to 8 Mb
  - 2 Mb increments
  - Dual-ported (VMEbus and VSB)
  - 235 ns 32-bit write (VSB, cache enable mode)
  - 280 ns 32-bit read (VSB, cache enable mode)
  - Byte parity generation
  - Byte parity checking

Winchester and Floppy Disk Controller
- Up to 2 per system
- Up to 2 ST-506 Winchester drives per controller
- Up to 2 floppy disk drives per system
- 5 Mbps serial data rates
- Multiple-sector read/write
- Automatic head switching and cylinder positioning
- 2:1 interleaving
- 32-bit ECC (Winchester)

ESDI Disk Controller
- Up to 4 ESDI disks per controller
- Up to 10 Mbps data rates
- Zero-latency reads and writes
- 1:1 interleaving
- Scatter/Gather
- 32-bit ECC

SMD Disk Controller
- Up to 2 per system
- Up to 2 SMD disks per controller
- Up to 24 Mbps data rates
- Zero-latency reads and writes
- 1:1 interleave over multiple cylinders
- 32-bit ECC

5-1/4" Streaming Tape Controller
- QIC-02 Interface
- Transfer rate:
  - 90 Kb continuous
  - 200 Kb burst
- Streams at 5 Mb per minute

1/2" 9-Track Controller
- Pertec Interface
- Up to 200 ips GCR drive support
- Start/stop or streaming support
- GCR, PE, or NRZI recording

8-Port Serial Controller
- Up to 4 controllers per system
- 8 asynchronous RS-232-C ports per controller
- 10 MHz MC68010 microprocessor
- 128 Kb RAM memory
- 32 Kb ROM
- Port selectable:
  - 50 - 19200 bps
  - 1, 1.5, or 2 stop bits
  - 5 to 8 data bits
  - odd, even, or no parity
  - DTR, DSR, RTS, CTS, and DCD control functions
- Buffered pipe communications protocol

Intelligent Communications Controller
- 6 synchronous/asynchronous communications ports
- 10 MHz MC68010 microprocessor
- 512 Kb RAM memory
- 128 Kb ROM
- SDLC, HDLC, and BSC protocol support
- Port selectable:
  - 50 - 19200 bps
  - 1, 1.5, or 2 stop bits
  - 5 to 8 data bits
  - odd, even, or no parity
  - NRZ, NRZI, or FM encoding
- Buffered pipe communications protocol

Intelligent Ethernet LAN Controller
- Ethernet, Version 2.0 compatible
- Conforms to IEEE 802.3 (CSMA/CD) LAN standard
- 10 Mbps
- 10 MHz MC68000 microprocessor
- 512 Kb RAM memory
- 32 Kb EPROM
- Buffered pipe communications protocol

PHYSICAL SPECIFICATIONS

Height: 25.6 in. (65.0 cm)
Width: 12.5 in. (31.8 cm)
Depth: 28.0 in. (71.1 cm)
Weight: 135 lbs. (61 kg) Fully loaded Model 2616

ENVIRONMENTAL AND SAFETY SPECIFICATIONS

Safety
Meets UL 478 (EDP) and 114 (Office Equipment) (110 Vac only)
Meets CSA 154 (EDP) and 143 (Office Equipment) (110 Vac only)
Meets VDE 0806/8.81 (Office Equipment) (220 Vac only)
Meets IEC 380 (Office Equipment)

Emissions
Meets VDE 0871/6.78, Class A (220 Vac only)
Meets FCC Part 15, sub-part J, Class A

ESD
5000 Volts: No observable effect
12000 Volts: No operator perceived errors
24000 Volts: No permanent equipment damage

Altitude:
Operating: 10,000 feet ASL
Non-Operating: 30,000 feet ASL

Acoustic Noise Level
55 dBA maximum

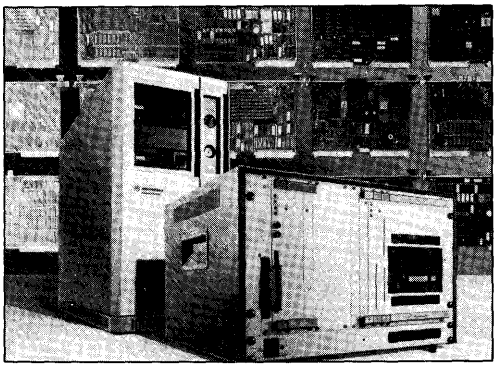
Temperature
Operating: 5°C to 40°C
Non-Operating: -40°C to 60°C

Relative Humidity
Operating: 10% to 80% RH, non-condensing
Non-Operating: 10% to 90% RH, non-condensing

Shock
Operating: .5 g
Non-Operating: 15 g

Transportation
Packaging and shipping containers and procedures comply with the current NSTA preship test procedures.

Shipping Container
Height: 34.5" (87.6 cm)
Width: 22.5" (57.2 cm)
Length: 41.0" (104.1 cm)



Microprocessor applications range in complexity from those using board level products in systems of 8/16-bit data/address paths and having modest data transfer rates to those for which solutions will be found using the 32-bit MC68020 which accesses 4 gigabytes of linear memory space and executes 2.5 million instructions per second.

For those applications requiring a system providing rugged mechanical structure, non-multiplexed and asynchronous data transfers, a powerful interrupt structure, flexible data and address paths, support for system failure detection and data transfer rates up to 40 megabytes per second, Motorola's VMEbus-compatible board level products — VMEmodules — can offer a quick solution.

Since its introduction in October, 1981, the VMEbus Interconnect System has gained such acceptance that it has become a de facto world standard. The bus is presently being standardized by both the IEEE and the IEC. Both a VMEbus Manufacturers Group and a VMEbus Users Group have been formed; the former to assure a high degree of technical compatibility between products and the latter to promote understanding of the bus, related hardware, operating systems and applications software.

VMEmodule compatibility with VMEbus results in many benefits for board level product users. One benefit is that since VMEbus is based on the EUROCARD standard which specifies not only the board format but also the enclosures, card racks and connectors, there is a high degree of mechanical and electromechanical interchangeability between the products of all vendors.

Presently, there are more than 500 products available from the more than 100 vendors worldwide giving the user an ever-widening range of choices for his designs. In addition, Motorola, Signetics and others are committed to the design and production of LSI-implemented bus-support functions which make interfacing the VMEbus easier and will result in the availability of an even greater variety of powerful VMEbus-compatible products.

VMEmodule Data Sheets

(continued)

VMEmodule Data Sheets (continued)

MicroMAP Manufacturing Automation Protocol Software

The seven layer Open Systems Interconnection (OSI) reference model is the result of work begun by an International Standards Organization committee in 1977 to provide a starting point from which beneficial compatibility among future network architectures could be realized. To provide networked data communications for their factory environments, General Motors Corporation developed the Manufacturing Automation Protocol which currently observes six of the OSI model layers. MicroMAP object code is an implementation of the MAP 2.1 specification for the Motorola MVME372 Advanced MAP Network Interface VMEmodule. For porting to another board or host system, source code is available in a variety of packages.

- Supports all MAP Layers — per MAP 2.1 and ISO/IEEE Specifications
- MAP Layers Implemented as Separate, Independent Modules
- Consistent Interfaces Between Layers
- Source Code Available:
  - Entire MicroMAP
    Host-resident, SYSTEM V/68 Dependent Code
  - Operating System Independent, MVME372-resident Code
    Layers 1–7 option
    Layers 1–4 option
    Layers 5–7 option
- Implemented in C for High Portability
- Certified by ITI
- SYSTEM V/68 Support:
  - SYSTEM V/68 Driver — Common Environment Interface, Application Task Interface Utility Libraries
- MVME372 Advanced MAP Interface Support:
  - MC68824 Token Bus Controller and MC68020 32-Bit Microprocessor-based Implementation of MAP 2.1
  - VRTX Real-Time Kernel
  - Common Environment Interface
- No Software Limit on Number of Concurrent, Full Duplex Connections
- Supports MC68824 Token Bus Controller Chip

MicroMAP LAYER DESCRIPTIONS

In the MicroMAP implementation of MAP, the coaxial cable and the node modem, for example, the MVME371FS MAP Network Interface Broadband Modem, comprise Layer 1. The functions of Layer 2 are provided partly by the MC68824 Token Bus Controller and its driver and partly by Logical Link Control and Media Access Control software. Figure 1 diagrams the correspondence of the Open System Interconnect model layers to the layers of the MicroMAP implementation of MAP 2.1. Figure 2 shows the locations of the various elements of the MicroMAP system.

LAYER 1 — PHYSICAL

The physical layer provides the electrical/mechanical connection for transmission of data between nodes. It performs the electrical encoding of the data for transmission over and regulates access to the network cable thus freeing the other MicroMAP layers from dependence on the physical medium.

LAYER 2 — LOGICAL LINK CONTROL/MEDIA ACCESS CONTROL

The Logical Link Control software design provides a connectionless service which conforms to the IEEE 802.2 standard for type 1 service (ISO 8802/2). The Media Access Control (MAC) is realized using the MC68824 Token Bus Controller chip and conforms to the IEEE 802.4 standard (ISO 8802/4). Together, these sublayers form the Data Link layer of the OSI model.

LAYER 3 — INTERNET

The MicroMAP Internet layer corresponds to the ISO 8802/2 Network layer standard (DIS 8473). It provides a connectionless service for message routing and relaying between cooperating nodes on the same network, or on interconnecting networks. Services provided by the Internet layer are independent of the distances separating the source and destination nodes.

The key characteristics of Internet services are:

- Transparency of transferred information
- Composition, decomposition, segmentation and reassembly of the data units
- Inter-network routing

LAYER 4 — TRANSPORT

The ISO 8073 Transport layer specifies a guaranteed transfer of data between cooperating session tasks in different network nodes. The layer also performs control of data flow between itself and a remote node Transport layer.

The MicroMAP Transport provides expedited data service, extended sequence number formats, user data on Connect Request/Confirm and Disconnect Request, subsequencing of Acknowledge messages and can checksum all message data.

LAYER 5 — SESSION

The intent of the Session layer (ISO 8327) is to provide cooperating tasks on different network nodes with the means to organize, synchronize and regulate the orderly exchange of data. The MicroMAP Session layer provides fundamental Session kernel capability which is to connect, transfer data and disconnect. In addition, it supports full duplex capability.

LAYER 6 — PRESENTATION

Since the MAP 2.1 specification does not define a Presentation layer, many of the applications and utilities of Layer 7 directly use the services of layer 5, the Session layer.

LAYER 7 — APPLICATION

Layer 7 provides user tasks access to network services offered by lower layers. The primary means for using the network is a collection of network application programs which reside in the application layer. These include:

- CASE — Common Application Service Elements (ISO (DP) 8649) (Kernel — MAP 2.1)
- FTAM — File Transfer, Access and Management (ISO 8571/1-4)
- AGENT — Network Management Agent (MAP 2.1)
- DIR SRVS — Directory Services (MAP 2.1)

In addition to these application programs, MicroMAP also provides Manufacturing Message Format Standard (MMFS) support. The MMFS application resides in the host computer and is accessed through the MMFS user interface library.

Common Application Service Elements (CASE)

The Common Application Service Elements (CASE) program is an interface utility residing in the Application Layer that is designed to relieve user application programs of much of the effort required when MicroMAP communication services are used. CASE includes a facility for making and breaking associations with remote user programs and for transferring data to/from remote applications. A framework is provided for authentication and authorization of users.

File Transfer, Access and Management (FTAM)

The File Transfer, Access, and Management program (FTAM) resides in the Application Layer and uses the services of the Session layer. FTAM provides the following services:

- Establish and terminate connections to a remote FTAM
- Select existing files
- Create new files
- Delete files
- Read and write files
- Read a file's attributes on a remote system

The MicroMAP FTAM implementation includes an address resolution capability for transforming a logical address into the corresponding Service Access Point (SAP) to facilitate the writing of hardware-independent software.

NETWORK MANAGEMENT AGENT (AGENT)

The MicroMAP Network Management Agent (AGENT) is an application program residing in Layer 7 that interacts with a remote node Network Manager. Typically, the remote manager requests information from specific layers on the local node via AGENT. However, while a connection with the Network Manager exists, AGENT may on its own, report on local system performance parameters that have exceeded threshold values. MicroMAP layers that interact with AGENT are Internet, Transport, Session and Case. Using System Management Protocol Units (SMPDUs), the Network Management Agent uses the services of Case to communicate with the remote Network Manager.

VRTX — MicroMAP OPERATING SYSTEM

All MicroMAP layers are implemented as tasks that execute under the control of the VRTX Real-Time, Multitasking Operating System. Post and pend message services are used for communicating with each MicroMAP task. The operating system co-resides with the Common Environment programs in a PROM set on the MVME372 Advanced MAP Network Interface Module.

The VRTX Operating System is supplied by Motorola under license to Hunter & Ready Inc.

THE COMMON ENVIRONMENT

To communicate with a remote host, an application task executing on the local host requires a path over the VMEbus and through the MicroMAP software layers to the network. On the MVME372 module, firmware comprising two executive modules and the VRTX kernel, collectively called the Common Environment, facilitate this vital function by providing a standard interface between the MVME372 and the host. The host driver and the Common Environment communicate over the VMEbus using a standard Motorola Buffered Pipe Protocol that provides the non-busy interface required for optimum data rates.

Typical outgoing communication sequence:

- The user task passes its message to the Common Environment Driver on the host
- The driver passes the message to the Common Environment on the MVME372 MAP Network Interface Module.
- The Common Environment passes the message to MicroMAP.
- MicroMAP transmits the message out onto the network.

Typical incoming communication sequence:

- MicroMAP receives an incoming message from a remote node.
- MicroMAP passes the message to the Common Environment.

- The Common Environment passes the message to the Common Environment Driver on the host processor module.
- The Driver passes the message to the recipient task.

USER INTERFACE LIBRARIES

To make individual MicroMAP functions readily available to application tasks, User Interface Libraries written in C are provided that offer the most often used MAP services. Appropriate libraries are linked with the host-resident application program to obtain the desired functions from the MicroMAP software residing on the MAP Network Interface Module. Essentially, the libraries veil the complexities of the MAP 2.1 protocols by reducing the need for the user to acquire in-depth knowledge of the MAP layers software interfaces or of the Common Environment facilities.

The following user interface libraries are provided with MicroMAP:

- CASE — Common Application Service Elements
- MMFS — Manufacturing Message Format Standard
- FTAM — File Transfer, Access and Management

The other Layer 7 application programs do not require user interface libraries since they are normally called only by other MicroMAP applications. If the services offered by DIR SRVS and AGENT are required, however, they may be obtained through direct interface by a user task.

CASE USER INTERFACE (CUI)

The CASE User Interface (CUI) is a library of function calls that allows Case services to be used without complying with the details of the Layer 7 interface to the Case application. A user program may simultaneously communicate with multiple remote user applications, have simultaneous multiple communications with one remote application, or both. All CUI services are performed synchronously, i.e., a user task having made a request must wait until the request has been satisfied (successfully or unsuccessfully) before proceeding.

FTAM USER INTERFACE (FUI)

The FTAM User Interface (FUI) is a library of function calls that allows the use of FTAM services (on the MVME372) without complying with the details of the Layer 7 to FTAM interface. User programs may simultaneously communicate with the FTAM on multiple remote nodes or have multiple simultaneous communications with any remote node or both. All FTAM services are performed synchronously, i.e., a user task having made a request must wait until the request has been satisfied (successfully or unsuccessfully) before proceeding.

MMFS USER INTERFACE (MUI)

The MMFS User Interface (MUI) library provides function calls that eliminate the encoding and decoding of MMFS messages. Each function call uses CUI to establish, maintain and release connections.

DOWNLOADING MicroMAP

The MicroMAP software and applications reside in Random Access Memory (RAM) on the MVME372 Advanced MAP Network Interface Module. When power is applied at a node or when a node is initialized, the MicroMAP software must be loaded into the MVME372 RAM. Download software is supplied with MicroMAP to perform this function.

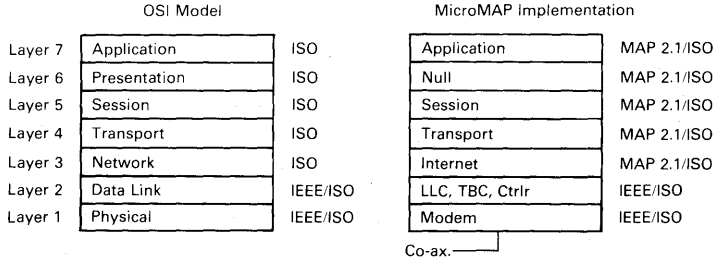


Figure 1. Correspondence of OSI Model to MicroMAP

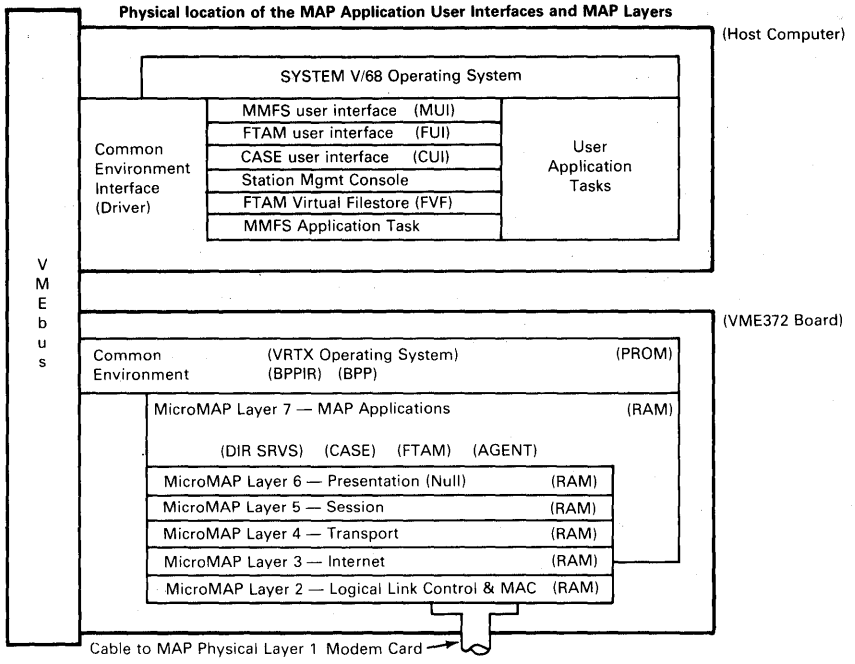


Figure 2. Physical Location of MicroMAP Elements

ORDERING INFORMATION

Part Number	Description
MicroMAP1-7	MAP 2.1 object code including Layers 1–7 for use with MVME372 Advanced MAP Network Interface in SYSTEM V/68 host system. Supplied on 5-1/4" floppy disk readable by SYSTEM V/68 host. Includes both host-resident and MVME372-resident MicroMAP and Common Environment object code.
MicroMAP1-4	MAP 2.1 object code including Layers 1–4 for use with MVME372 Advanced MAP Network Interface in SYSTEM V/68 host system. Supplied on 5-1/4" floppy disk readable by SYSTEM V/68 host. Includes both MVME372-resident MicroMAP and Common Environment object code.
MicroMAP1-7S	MAP 2.1 Source Code for Layers 1–7. Includes MICROMAPHRS. Does not include COMMONENVS. Supplied on 5-1/4" floppy disk readable by SYSTEM V/68 host.
MicroMAP1-4S	MAP 2.1 Source Code for Layers 1–4. Does not include COMMONENVS. Supplied on 5-1/4" floppy disk readable by SYSTEM V/68 host.
MicroMAP5-7S	MAP 2.1 Source Code for Layers 5–7. Does not include COMMONENVS. Supplied on 5-1/4" floppy disk readable by SYSTEM V/68 host.
MicroMAPHRS	MAP 2.1 Host Resident Source Code. Includes user interfaces for FTAM, CASE and MMFS, MMFS code, FVF (Virtual File Store and Forward), Network Management Console Source Code for Parameter Download, Table Download and Monitor Download Features. Supplied on 5-1/4" floppy disk readable by SYSTEM V/68 host. Includes host resident Common Environment source.
COMMONENVS	MAP 2.1 Common Environment Source Code. Includes driver information using MVME372 Advanced MAP Network Interface VMEmodule as a porting example.

VMEbug Debugging Packages for the VMEmodule Monoboard Microcomputer

VMEbug Resident Package

- EPROM Resident System DEbug Monitor
- More Than 30 Commands for DEbug, Up/Downline Load
- Single-line Assembler/Disassembler for Assembly Language Program Development
- Full Speed Execution of System and User-Developed Programs Operating in the VMEmodule Monoboard Microcomputer System
- Virtual Terminal Capability for Up/Downline Load from an EXORmacs Development System or from a Cross-Computer
- Powerful Software and System DEbug Command Set Allows Access to all VMEmodule I/O, Control and Memory Facilities Plus the Full 16M Byte Direct Address Range of the VMEbus System Bus
- Includes all Required Installation and Operation Documentation
- Includes Bootstrap Loader for both MVME420, SASI Peripheral Adapter and the M68RWIN1 Disk Controller

VMEbug Source and Relocatable Object Module Package

- Source and Relocatable Object Modules for VMEbug on Diskette or Cartridge Disk
- Relocatable Object Modules Allow Users to Include Only the VMEbug Items Needed in Their End System; to Link in Their Own Up/Downline Loader; and/or to Link in Their Own Bootstrap Loader
- Source Modules Allow User Modification of VMEbug as Desired

The VMEmodule Monoboard Microcomputer debug package, VMEbug, is available as two separate product offerings. VMEbug is an EPROM-based resident package ready for installation and immediate use with the VMEmodule Monoboard Microcomputer installed in a VMEbus based backplane. Such a backplane is provided within Motorola's VMEmodule Chassis (MVME940). VMEbug Source and Relocatable Object Modules are a separate product available on either VERSAdos compatible diskette, or cartridge.

VMEbug provides a powerful evaluation and system debugging tool for VMEmodule Systems. The EPROM Resident Package will operate in 32K bytes of ROM space. VMEbug uses the first 1152 words of RAM storage for Interrupt Vectors and temporary storage. The EPROM resident package is delivered in four 8K byte EPROMs. Table 1 lists the commands available to the user.

The package permits execution of system and user-developed programs operated in a VMEmodule Monoboard Microcomputer system environment under complete operator control. VMEbug may be utilized with a VMEmodule Monoboard Microcomputer in a standalone environment with only a user provided standard RS-232C asynchronous ASCII terminal. Alternately, it may be used with a second serial I/O port connected to a host computer for up/downline loading of programs in Motorola "S" Record format. When connected to a host computer in this manner, the VMEmodule/VMEbug/Operator Terminal combination appears as a normal asynchronous ASCII terminal (a virtual terminal) to the host operating system. The second serial I/O port, the host computer interface, would be implemented through an I/Omodule board or other similar functional element. VMEbug may also be used with the MVME420 or the M68RWIN1 controllers (or both) and appropriate disks to load programs from disk to memory.

In a typical debug session, the user will download his program to a VMEmodule Monoboard from the host computer used for software development. After loading, VMEbug commands may be used to examine and modify memory, set breakpoints to run particular program segments, and track program progress. The user may set up and examine a variety of conditions using any of the powerful commands listed in Table 1, such as the Register Display/Set series and the

memory block manipulation commands. The Data Conversion command serves as an aid in examining and modifying data by converting hexadecimal to decimal, and decimal to hexadecimal. If corrections or program patches are required, these may be performed and checked in the VMEmodule. The user may have a corrected copy to the host computer files by using the Memory Dump command for upline load. Creating program patches may be aided by use of the Display Offsets command to assist with relocatable and position-independent code. The user may also copy all traffic to the serial port debug terminal on a printer attached to an auxiliary parallel port by use of the Attach Printer command. This may be useful for desk debugging following a debug session. The parallel printer interface port may be implemented through the appropriate I/Omodule board.

The user may communicate with the host computer as a terminal for purposes other than up/downline load by executing the Transparent Mode command or the Send Message to Port 2 command. By using the Port Format command, the serial port may be reconfigured for such attributes as baud rate, stop bits and number of data bits. In the Transparent Mode, the two serial ports must operate at the same baud rate.

Bootstrap load and dump commands permit the user to bootstrap from several controller/device combinations. The controller boards currently supported are the M68RWIN1 Winchester Disk Controller and the MVME420 SASI Peripheral Adapter. The drives currently supported are the 5¼" Winchester and the 5¼" Floppy or 8" Floppy on the M68RWIN1, and the 8" Winchester and 8" Floppy on the MVME420. The Boot Dump command permits the user to write his operating system to a diskette/disk in bootstrap load format for subsequent use in boot loading. The IOP command permits the user to create the diskette/disk format required.

VMEbug may be used for debug in total systems environments which include the VMEmodule Monoboard Microcomputer and with other Motorola VMEmodules as well as user-developed VMEbus compatible modules.

The Source and Relocatable Object Module Package provides users with the information to link VMEbug into their specific systems in either modified or unmodified form. The Relocatable Object Modules are designed to permit creation of a load module with or without the "S" Record up/downline loader. Users may also apply the Relocatable Object Modules substituting their own device bootstrap loader for the VMEbug disk bootstrap loader.

Source Modules permit the user to modify or customize any of the VMEbug functions as desired.

TABLE 1 — VMEbug Commands

Command	Description
MD <addr1>[<count>]	Memory Display/Disassembly
MM <address>[;<opts>]	Memory Modify/Disassembly/Assembly
MS <address><data...>	Memory Set
.A0 - .A7 [<expression>]	Display/Set Address Register
.D0 - .D7 [<expression>]	Display/Set Data Register
.PC [<expression>]	Display/Set Program Counter
.SR [<expression>]	Display/Set Status Register
.SS [<expression>]	Display/Set Supervisor Stack Pointer
.US [<expression>]	Display/Set User Stack Pointer
DF	Display Formatted Registers (All)
BF <address1><address2><word>	Block Fill (with 16-bit data word) Memory
BM <address1><address2><address3>	Block Move
BS <address 1><address 2><data>	Block of Memory Search
BI <address1><address2>	Block Initialize
BT <address1><address2>	Block Test of Memory
DC <expression>	Data Conversion
OF	Display Offsets
.R0 - .R6 [<expression>]	Display/Set Relative Offset Register

TABLE 1 — VMEbug Commands (continued)

Command	Description
BR [<address>[;<count>]]	Breakpoint Set (up to 8)
NOBR [<address> <address>...]	Breakpoint Remove (any or all)
GO [<address>]	Execute Program
GT <breakpoint address>	Go Until Breakpoint (sets temporary breakpoint)
GD [<address>]	Go Direct (No Breakpoint or Track Set, and no Exception Vector Changes)
TR [<count>]	Trace (set for number of instructions)
TT <breakpoint address>	Trace to Temporary Breakpoint
PA	Printer Attach (Print as well as display)
NOPA	Reset Printer Attach
PF [<port number>]	Port Format (set Serial Port Attributes)
TM [<exit character>]	Transparent Mode (Two serial ports transparently connected)
*text...	Send Message to Port 2
HE	HELP (Display VMEbug commands)
DU <address1> <address2><text...>	Dump ("S" Record Upline load)
LO [;<opts>] [=text]	Load ("S" Record Downline load)
VE [=text]	Verify ("S" Record Downline load verify)
BD [<device>] [<controller>]	Boot Dump
BH [<device>] [<controller>]	Bootstrap Halt
BO [<device>] [<controller>] [<string>]	Bootstrap Operating System
IOP	I/O Physical to Disk
IOT	I/O "Teach" to Disk
Command Line Edit and Control Functions:	
(BREAK)	Abort Command
(DEL)	Delete Character
(CTRL-D)	Redisplay Line
(CTRL-H)	Delete Character
(CTRL-W)	Suspend Output*
(CTRL-X)	Cancel Command Line
(cr)	Send Line to Memory

*When (CTRL-W) is used, the user can cause the output display to continue by entering any character.

Ordering Information

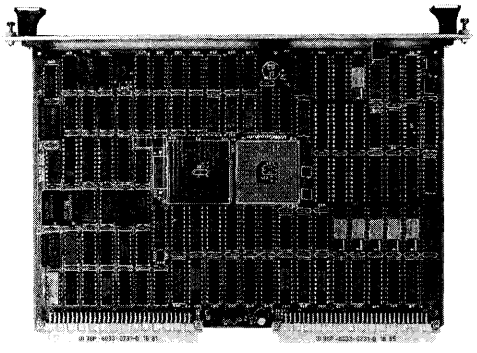
Part Number	Description
MVMEBUG	VMEbug, the VMEmodule Monoboard Microcomputer System Debug Package, includes EPROM set* and User's Manual.
MVMEBUG1	Source and Relocatable Object Modules for the VMEbug system on VERSAdos Diskette for the EXORmacs Development System.* Includes User's Manual.
MVMEBUG/D	MVMEbug Debugging Packages User's Manual

MVMEXTCAC-1
MVMEXTCAC-2

VMEmodule™ Cache Accelerator Module

For VME-Based Modular 32-Bit Microcomputer System

- Provides 25% to 100% System Performance Increase[1]
- Requires no Program Modification
- 16Kb Cache, Organized as 4K, 32-Bit Entries
- Supports Both Program and Data Transactions
- Zero Wait-State Operation[2]
- Parallel MMU Operation with Page Detection Circuitry
- Supports both VMEbus and VSB*
- Data-Fill on VSB Byte and Word Reads
- Supports all MC68020 Bus Transfer and Sizing Modes
- Diagnostic Mode



Cache memory systems are often employed to speed up the performance of microcomputers by compensating for the typical speed mismatch between a fast CPU and the relatively slow associated dynamic RAM memory system. The Cache Accelerator performs this function for microcomputers built around the MVME130 Family of M68020-based microprocessor modules.

The Cache Accelerator is a small, fast memory system that provides concurrent storage of data most recently stored in main memory. Consequently, on subsequent accesses to these "cached" locations, the much slower main memory access can be circumvented, thus speeding up the communications cycle.

Since the Cache Accelerator has a much smaller memory capacity than the system memory, little used data stored in the cache memory is routinely replaced with more active data, based on the bus traffic between the microprocessor module and the VMEbus and VSB memory interfaces.

GENERAL DESCRIPTION

The MVMEXTCAC External Cache Accelerator is a plug-in cache module, supporting the MVME130 family of Monoboard Microcomputers in VMEbus-based microcomputer systems. It is designed to enhance the operation of these MC68020-based systems by providing a typical 25% to 100% increase in performance without modification of existing programs.[1] Software transparency is achieved through the utilization of a physical address memory structure, coupled with a VMEbus Monitor. The Monitor automatically cleanses the cache of stale data, and enables the Cache Accelerator to operate in both single and multiprocessor environments without software intervention.

The Accelerator is a single-set, direct-mapped cache comprising 16Kb of high-speed static memory, organized as 4K, 32-bit data entries. Each entry replicates a 32-bit word located in VMEbus or VSB memory space and is updated concurrently with a processor write-access or initial read-access. Any combination of bytes from an access can be cached at a time, and a data-fill function implemented for VSB enables the caching of an entire 32-bit entry for byte- and word-read operations. The Accelerator caches both program and data transactions throughout, and it can be configured, by function code, to cache data from any combination of supervisor program, supervisor data, user program, and user data mode accesses.

The Cache Accelerator Module supports any speed MC68020 microprocessor, and it is jumper selectable to support either the MC68851 Memory Management Unit or M68KVMMB851 Memory Management Board, or an MMU bypass board if the MMU function is not required. With an MMU installed, the page detection circuitry on the cache enables its RAMs to operate in parallel with

1. Actual performance improvement varies with program applications.
2. With MMU bypass card installed, and dependent on MC68020 clock to address strobe timing.

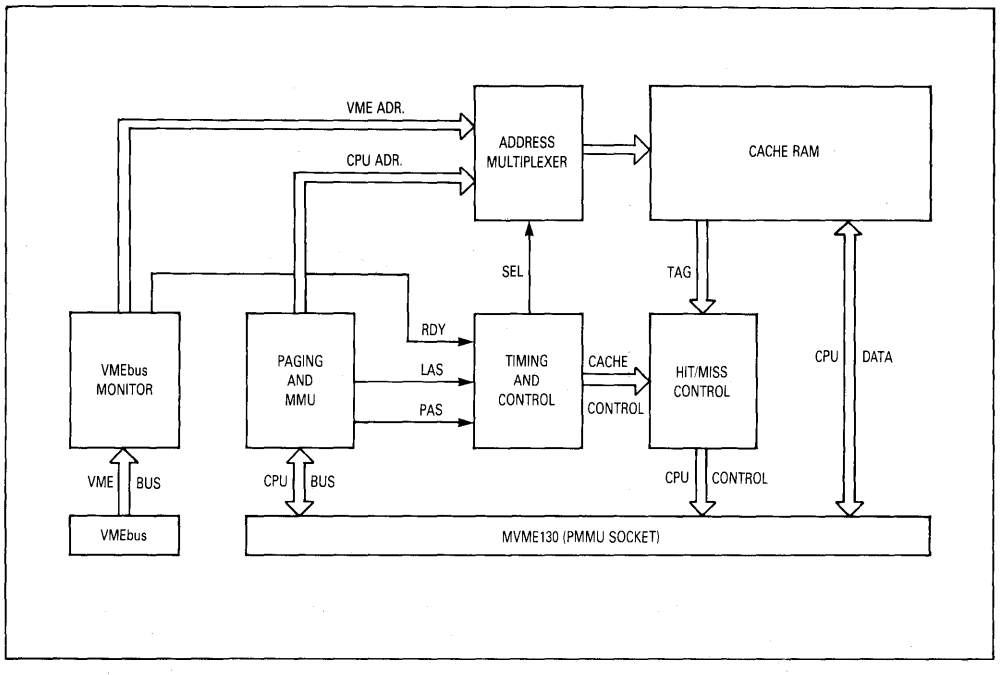


Figure 1. MVMEXTCAC Functional Block Diagram

the MMU address translation RAMs to maximize system performance. A cache page is defined by the physical address bits PA08–PA13, and if they remain the same on a following cycle, then the address access time of the cache RAMs is buried in the MMU address translation period. With a bypass board installed, the cache receives the address signals directly from the MC68020 microprocessor, and zero wait-state operation is obtainable. Zero wait-state operation is achieved at a processor frequency of 12.5 MHz[2], and it may be achieved at 16.67 MHz when faster static RAMs become available.

The Cache Accelerator provides direct access to the cache RAMs for diagnostic and inquiry purposes, and provides a test mode of operation to establish general system confidence.

BASIC CACHE OPERATION

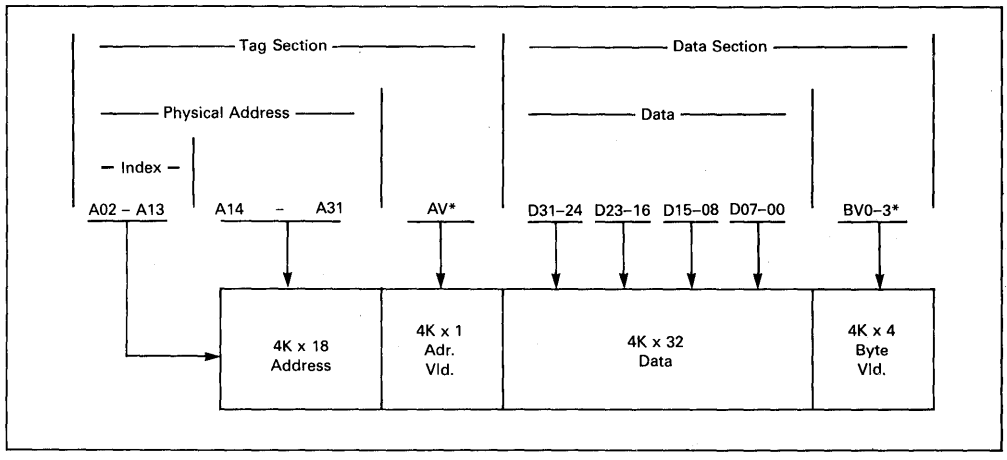
When the processor drives an address for a read operation, the index section PA02–13 addresses the tag and data RAMs to select one entry from the cache. The remaining address bits from the processor are then compared to the corresponding address bits read from the tag section of that entry. If they match, and the Address-Valid bit is set, and if the Byte-Valid bits for the bytes the processor is requesting also match, then the cycle "hits" and the cache supplies the requested data to the processor and terminates the cycle. If any event does not match, then the cycle "misses" and a VME or VSB access is initiated, the cache entry is updated with the new address and data information, and the Valid bits are set appropriately. Note that a write cycle will always cause a miss since the processor, and not the cache, is supplying the data. Several other factors determine the hit qualification and cacheability of a processor cycle. These include the control bits, the type of processor cycle, the MMU cache enable signal and the onboard address decoder.

FORM FACTOR

The form factor of the module is a double-high Eurocard that plugs into the MMU socket resident on the MVME130/131 Monoboard Microcomputer Module. The MMU is relocated from the monoboard to the cache module, and the two-Eurocard stack plugs into two consecutive card slots of the VMEbus chassis. A double-wide front panel with card-ejector handles is provided for mechanical stability and ease of handling.

2. With MMU bypass card installed, and dependent on MC68020 clock to address strobe timing.

Table 1. MVMEXTCAC — Memory Organization



MEMORY ORGANIZATION

The complete memory section of the Cache Accelerator is organized as 4K entries of 55 bits per entry (see Table 1). It consists of a "Tag Section" and a "Data Section," where the tag section stores the physical addresses of the off-board memory locations, and the data section stores the data residing at those locations. The lower address bits (PA02–PA13) form an "index" that directly addresses the cache RAMs to select the particular entry to be used, and the remaining address bits (PA14–PA31) are stored and read from the tag RAMs as data. An Address Valid bit (AV) is generated internally for each entry to indicate that the address stored in the Tag is valid information. It may not be valid either because the tag entry has not been written to since power-up or since the cache was last cleared, or because the VMEbus Monitor has set the entry invalid. The data RAMs contain the corresponding bytes that were cached from the bus, and the four Byte-Valid bits (BV0–BV3) are generated internally to qualify those bytes. BV0 is associated with byte address 0 of the long-word entry, and operates on the most significant byte of the data bus, D31–D24. The other Byte-Valid bits operate in sequence.

VMEbus MONITOR

The VMEbus Monitor is a separate machine on the Cache Accelerator that operates asynchronously with the processor to prevent "stale" data from remaining in the cache. It operates automatically so that the host user benefits from the performance advantages offered by the cache without even being aware that the Cache Accelerator has been installed in the system.

Stale data could occur in the cache when other VMEbus masters modify data in main memory which was previously cached. To prevent this, the Bus Monitor has a two-level FIFO register stack which captures the addresses of write operations initiated by bus masters other than the host, and presents the addresses to the cache for processing. The cache services the monitor by comparing the captured address with the corresponding entry in its tag memory. If the tag is Valid and compares, a "monitor hit" occurs and the cache invalidates the entry by resetting its address Valid bit. If the processor then accesses that physical address, the cache will miss because of the reset Address Valid bit, and the cycle will be routed out to main memory for the new data to update the invalid entry.

USER INTERFACE

The user interface in Table 2 shows the devices on the Cache Accelerator used to control and diagnose its operation. These consist of Cache Control and Mask Registers and two hard-decoded address spaces used to directly access the cache RAMs for diagnostic and inquiry purposes. The cache powers-up in a cleared and disabled state, and can be enabled for software transparent operation by storing a "F1" to the Cache Control Register. In addition, the cache can be tested or configured for a specific user application through the various control bits shown. Since the entire user interface address space can safely be accessed without the Cache Accelerator installed, a cache utility can be implemented without prior knowledge of its presence in the system.

Table 2. MVMEXTCAC — User Interface

Cache Control Register — Address FFFC0000

D31 (Reset State = FF)							D24
VBCEN	MONEN	MLBK	MBLC	CCLR	CWEN	CREN	CTST

Mask Register — Address FFFC0001

D23 (Reset State = xF)							D16
X	X	X	X	MSD	MSP	MUD	MUP

Data Access Space — Address FFFD000–FFFD3FFC
Data Entries, 000–FFF

D31			D00
BYTE0 D31–D24	BYTE1 D23–D16	BYTE2 D15–D08	BYTE3 D07–D00
.	.	.	.
.	.	.	.
.	.	.	.

Tag Access Space (read-only)
Addresses — FFFE0000–FFFE3FFC
Tag Entries, 000–FFF

D31 D14	D07	D03–D00
PAD31–PAD14	AV	BV0–BV3
.	.	.
.	.	.
.	.	.

DEFINITIONS

VBCEN — VMEbus Cache Enable	MSP — Mask Supervisor Program
MONEN — VMEbus Monitor Enable	MUD — Mask User Data
MLBK — Monitor Loopback (for diagnostic use)	MUP — Mask User Program
MBLC — Monitor Block (for diagnostic use)	AV — Address Valid Status
CCLR — Cache Clear	BV0 — Byte Valid Status for D31–D24 (byte address 0)
CWEN — Cache Write Enable	BV1 — Byte Valid State for D23–D16 (byte address 1)
CREN — Cache Read Enable	BV2 — Byte Valid Status for D15–D08 (byte address 2)
CTST — Cache Test (for diagnostic use)	BV3 — Byte Valid Status for D07–D00 (byte address 3)
MSD — Mask Supervisor Data	

BENCHMARK TEST RESULTS

Tests run on a 12.5 MHz MVME130 Microcomputer Module used in conjunction with the Cache Accelerator have demonstrated impressive improvement in performance. The results of two sets of benchmark programs are tabulated in Table 3. The first set utilized EDN benchmarks** which perform commonly used character and data manipulations; the second utilized user supplied routines performing data intensive array operations. The tests, run for four combinations of cache memory utilization, indicate the degree of improvement attributable to the use of the Cache Accelerator.

Table 3. Performance Test Results

Test	No Cache	MC68020 Cache	MVMEXTCAC Cache	MC68020 and MVMEXTCAC Cache
EDN-E (Char. String Search)	285	141	131	99 μs
EDN-F (Set/Reset/Test Bit)	60	36	33	30 μs
EDN-H (Link List Insertion)	121	194	70	70 μs
EDN-I (Quicksort)	22,320	11,792	11,072	9001 μs
EDN-K (Bit Matrix Translation)	243	173	159	136 μs
CUSTOMER BENCHMARK	125	104	73	68 sec.

**EDN — September 16, 1981

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristic	Specification
Cache Implementation	Single-set, direct mapped, physical
Memory Organization	4K, 32-bit data entries
Replacement Algorithm	Single entry, write-through
Hit Timing, LAS to DSACKx	No MMU — 60 nanoseconds With MMU — 70 nanoseconds (Page Hit) — 130 nanoseconds (Page Miss)
Power Requirements	+5 Vdc, 6.4 A max. (4.3 A typical)
Operating Temperature	0° to 50°C
Storage Temperature	−40° to 85°C
Relative Humidity	5% to 95% (non-condensing)
Physical Characteristics Height Width Thickness	 6.30 in (16.00 cm) 9.19 in (23.34 cm) 0.062 in (0.157 cm)
Component Projections Component Side Solder Side	 0.50 in (1.27 cm) max 0.067 in (0.17 cm) max (except for interconnect header)

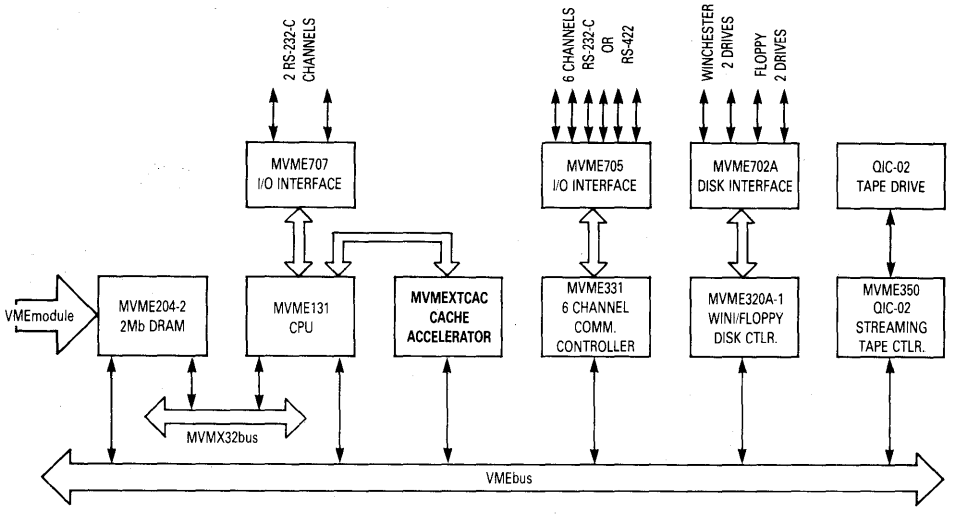
ORDERING INFORMATION

Part Number	Description
MVMEXTCAC-1	Cache expansion kit for the MVME130/131 processor module. Does not include hardware memory management. Includes User's Manual.
MVMEXTCAC-2	Same as MVMEXTCAC-1 with hardware memory management. Includes User's Manual.
MVME130XT	VMEmodule 32-bit Monoboard Microcomputer with MC68020 CPU and 16Kb cache. Includes User's Manual.
MVME131XT	Same as MVME130XT with hardware memory management. Includes User's Manual.

RELATED DOCUMENTATION

Part Number	Description
MVME130/D	MVME130 User's Manual
MVME130bug/D	130bug User's Manual
M68KVMMB851/D	M68KVMMB851 User's Manual
MVME204-1/-2/D	MVME204-1/-2 User's Manual
MVMESB/D1	VSB Specification Manual
HB212/D	VMEbus Specifications Manual

The VMEsystem



A Typical High-Speed MVME130/131XT Microcomputer VMEsystem

The MVMEXTCAC Cache Accelerator Module conforms to the VMEbus interconnect system which has emerged as a world-wide Standard for 8-/6-/32-bit microcomputers. It is part of the VME-module product family whose M68000-based processing power, Eurocard mechanical format, powerful VERSAdos real-time software support and international multiple sourcing have made it one of the most popular product lines in the industry.

The VMEsystem architecture, and a board-level component complement that relate to the Cache Accelerator, are illustrated in the above block diagram. Of particular significance are:

- The well-established functional versatility of the VMEbus whose non-proprietary technical specification is widely used for system implementation
- An auxiliary MVMXbus for high-speed memory expansion
- An MC68020-based, 32-bit CPU board (MVME131) utilizing the latest advances in microprocessor design
- A large repertoire of peripheral (controller) modules and memory boards

The unrivalled acceptance of the VMEsystem philosophy has yielded a selection of compatible board-level products from well over a hundred vendors, to offer systems manufacturers the basic tools with which to implement their designs quickly, efficiently, reliably and cost effectively.

Advance Information

117bug Debug Monitor

FIRMWARE MONITOR

- EPROM Resident Debug Monitor for MVME117
- 41 Powerful Commands
- One Line Assembler/Disassembler for Convenient Program Monitoring/Modification
- Full Speed Execution of System and User Programs Operating in a MVME117 Microcomputer System
- Virtual Terminal Capability for Up/down Load to any Host Computer Using Motorola S-Records
- Provides Access to I/O, Control and Memory Facilities of a MVME117-based System Plus a 16Mb Direct Address Range of the A:24 VMEbus
- Permits Display and Modification of the MVME117 Clock/Calendar
- Permits Configuration of Bits/Char, Baud Rate and Parity for Both MVME117 Terminal Ports
- Includes Boot Facilities for Loading the VERSAdos Operating System

SCSI/SASI FIRMWARE SUPPORT

- Includes Disk Controller Initialization and Disk I/O Commands for SCSI and SASI Controllers (compatible with ANSI X3T9.2/82-2 revision 14 "Small Computer System Interface Draft")
- Includes Low Level Functions to Support SCSI/SASI
- Supports Disk Controllers in the Initiator Role
- Supports Arbitration and Reselection
- Interface Protocol allows User-defined Command Sequences

The debug monitor for the MVME117 Monoboard Microcomputer — 117bug — is supplied as two separate products: for resident use, the object code is supplied in two 32Kb EPROMs which plug into sockets provided on the MVME117 Monoboard Microcomputer and for customization, the source code is supplied on 5-1/4" diskettes readable on systems running the Motorola VERSAdos Operating System.

The monitor, 117bug for short, is a versatile, effective tool permitting full speed execution of system and application programs under complete operator control. It offers 41 powerful commands for application program development and modification and for MVME117 system hardware diagnosis and debugging. For standalone mode operation, only a power supply and an RS-232, asynchronous ASCII terminal connected to one of the MVME117 serial ports is required. This mode allows full use of the 117bug software and hardware debugging capabilities and is commonly used for the final debugging of an application program.

So that its powerful hardware/software development capabilities and storage capacity may be employed, a host computer can be connected to the second serial port on the MVME117 for up/down loading of programs in Motorola S-Record format. For this purpose, 117bug includes commands used to place the microcomputer in the transparent mode and to configure its serial ports for the desired communication protocol.

BOOT LOAD SUPPORT

Loading of an operating system and programs from floppy or hard disk into the MVME117 system RAM is supported by 117bug. Controllers supported include those for the SCSI/SASI bus. In addition, the Motorola M68RWIN1 Winchester Disk Controller I/O Module/ MVME316 I/O Channel Interface Module combination is fully supported. Connection of an SCSI controller to the MVME117 Microcomputer is eased using an MVME708 Transition Module which provides the conversion from DIN to industry standard connectors for serial, parallel and SCSI bus interfaces.

SELF TEST CAPABILITIES

Two tests are incorporated in 117bug. One executes during the power-up sequence and tests the MC68010 Microprocessor, ROM and RAM resources required to bring up the monitor. This test can be enabled or disabled using the PT or NOPT command.

The second test, Self Test, performs more extensive testing and displays a greater number of diagnostic status and failure messages. In addition to the resources tested by the power-up test, it tests the parity logic, parity RAM, CMOS RAM, serial port, real-time clock, programmable timer module, MC68881 peripheral (if installed), NCR5380 SCSI interface chip (if installed), fail LED, abort button, front panel switch and, if desired, a disk controller and disk drive. The option of including the testing of a system disk controller and associated disk drive is provided when the ST command is invoked to start the tests.

SUPPORT OF THE MVME117 SCSI INTERFACE

SCSI INTERFACE

The SCSI interface on MVME117 is based on the NCR5380 device which is a VLSI implementation of the Small Computer Systems Interface (SCSI) defined by the ANSI X3T9.2 Committee (Revision 14). The interface takes advantage of the chip to provide full support of SCSI bus communication features including bus request/arbitration, selection/reselection, initiator/target roles and the six SCSI information transfer phases listed in Table 1.

Table 1. SCSI Information Transfer Phases

Phase Name	Description
Data Out	Transfer data from Initiator to Target
Data In	Transfer data from Target to Initiator
Command	Transfer command from Initiator to Target
Status	Transfer status from Target to Initiator
Message Out	Transfer SCSI level message from Initiator to Target
Message In	Transfer SCSI level message from Target to Initiator

Communication between the MVME117 Microcomputer and peripherals on the SCSI bus is effected using several levels within 117bug. Data is interchanged between the high level and the lower levels using a standard protocol and a complement of nine callable functions. These directly manipulate the NCR5380 to obtain read, write and other activities from disk drives connected to controllers on the SCSI bus. The nine functions, which can be called using the Trap #15 instruction of the MC68010 Microprocessor, are listed in Table 2.

Table 2. Trap #15 Callable SCSI Interface Functions

Function Code (Hexadecimal)	Description
10	Read
11	Write
12	Attach Device
13	Detach Device
14	Format Disk with/without Defect List
15	Assign Alternate Sector (SCSI)
16	Assign Alternate Track (SASI)
17	Custom SCSI Sequence
18	SCSI Bus Reset
19	SCSI Controller Reset

Because of the complexity of the task of writing a driver to interface the NCR5380 chip, use of the nine callable functions within 117bug to obtain disk I/O could save a significant portion of the software engineering resource expenditure. One of the functions — Custom SCSI Sequence — provides a means for communicating with a SCSI controller which is not supported by the callable functions within 117bug.

MVME117 SYSTEM POWER UP/RESET SEQUENCES

On power up or reset, the 117bug monitor checks the "Initial Event" flag in memory to differentiate between power-up and reset. If a reset is required and certain essential vectors are present in the RAM from 0 to 3FF used by the monitor, a warm start is performed.

If it is an initial event, (power up) the power up sequence is entered and the "power up test" switch in battery-backed RAM is checked. The power up tests are not performed if not selected. A command is included in the 117bug set to allow this switch to be toggled. If selected, the tests are run and monoboard registers, vectors and onboard RAM initialized.

Following RAM initialization, the first four bytes in every 8Kb block in memory from $180000 to $F40000 are checked for indication that "ROM Boot" is wanted. If a boot is wanted, control is given to the indicated code. A Checksum Command is included in the 117bug command set to aid the user in preparing and installing a boot program module in ROM.

If the "ROM Boot" indication is not found, system RAM is sized and discontinuities noted following which the "Autoboot" software switch in battery backed RAM is checked to determine if an autoboot is wanted or not. In the latter case, a coldstart message is given and a coldstart performed.

On finding that an autoboot is wanted, 117bug causes the desired operating system or application program to be loaded into system RAM from the device and controller specified in the "Autoboot" data area. Should the "Autoboot" switch and associated device and controller identification have been lost from battery backed RAM, 117bug informs the user and re-initializes the "Autoboot" data to default values. An Autoboot Command is included in the 117bug set as a mechanism for toggling the "Autoboot" switch and specifying the device and controller to use.

CALLABLE 117bug FUNCTIONS

A TRAP #15 handler is included in 117bug which allows calls from a user program to input, output and other useful functions. Such a system call is made by including in the source program a TRAP #15 instruction line followed by a DC.W instruction containing the number of the desired function. Use of these functions provides a means of quickly preparing a small test program to aid in system development. The callable 117bug functions used for debugging are listed in Table 3.

Table 3. Trap #15 Callable Debug Functions

Function Code (Hexadecimal)	Description
0	Return to 117bug
1	Input data from console
2	Output data, CR/LF to console
3	Input data from specified port, no echo
4	Output data, CR/LF to specified port
5	Output data to specified port, no CR/LF
6	Output data to printer port, no CR/LF
7	Convert ASCII string to 8-digit hex number
8	Convert 1 hex digit to ASCII string
9	Convert 2 hex digits to ASCII string
A	Convert 4 hex digits to ASCII string
B	Convert 6 hex digits to ASCII string
C	Convert 8 hex digits to ASCII string

DEBUG FACILITIES

The 117bug monitor is a command driven program which has facilities for accepting and executing commands entered at the system console. The debug command set includes commands for examining and modifying registers and memory, commands which allow blocks of memory to be filled, moved or searched for the occurrence of specified data, breakpoint commands which allow program segments to be run, trace commands for examining the execution of instructions or small program portions, commands for beginning execution at a specified address, and commands for displaying the offset registers.

Other commands in the set facilitate host/MVME117 communications, the formatting of the monoboard's serial ports and attaching/detaching the system printer. Another utility converts the number base of data.

PROGRAM DEBUGGING

In a typical debug session after loading the program under development, appropriate commands from the debug set are invoked so that operation of a particular program portion can be checked and, if necessary, the code modified. Often, using the powerful Register Display/Modify, Memory Display/Modify and Block Memory Move Commands, trial conditions for a program portion are established and the code executed. Results can be determined by examining the processor register or memory values.

An alternative method is to set a breakpoint in place of an appropriate instruction to halt execution and obtain automatic display of processor register values. A more detailed examination of program validity is offered by the Trace Commands which allow execution of one instruction at a time, with accompanying processor register display following each execution. If required, program code is easily modified with the aid of the one line assembly/disassembly function provided by the Memory Modify Command. The Data Conversion Command which converts values from decimal to hexadecimal or hexadecimal to decimal is also a useful modification tool.

Debugging of modular code is greatly eased by the support of the offset registers provided by the 117bug command set.

When the program is fully debugged, it is saved or the Dump S-Records Command used to format and upload the code to the host for storage.

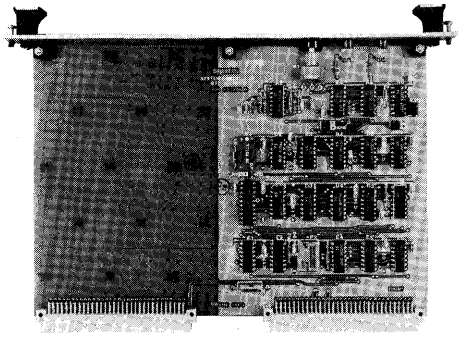
ORDERING INFORMATION

Part Number	Description
MVME117BUG	117bug, the Debug Monitor for the MVME117 Monoboard Microcomputer. Includes set of two 32Kb EPROMs and User's Manual.
M68V1XSBG117	Source modules for 117bug Monitor and Relocatable Object modules (.R0) for SCSI firmware. On 5-1/4" diskettes readable on any system running the VERSAdos Operating System. Includes User's Manual.
M68V1XSSCSI	Source modules for SCSI firmware, providing bus arbitration, handshaking and data transfers to and from SCSI bus. On 5-1/4" diskettes readable on any system running the VERSAdos Operating System.

MVME117 Firmware Monitor Commands		
[NO]AB	[<drive> <controller>]	Auto Boot Enable/Disable
BD	[<device>] [,<controller>]	Boot Dump
BF	<address1> <address2> <pattern>	Block Fill
BH	[<device>] [,<controller>]	Boot Halt
BI	<address1> <address2>	Block Initialize
BM	<address1> <address2> <address3>	Block Move
BO	[<device>] [,<controller>]	Boot Load
[NO]BR	[<address>[;<count>]]	Breakpoint Set/Clear
BS	<address1> <address2> 'literal string'	Block Search
BS	<address1> <address2> <data> [<mask>] [;<options>] (;W ;L ;-)	Block Search
BT	<address1> <address2>	Block Test
CS	[<address1> [<address2>]	Calculate Checksum
DC	<expression>	Data Convert
DF		Display Formatted Registers
DU	[<port_#>] <address1> <address2> [text]	Dump S-Records
G[O]	[<address>]	Go
GD	<address>	Go Direct
GT	<address>	Go Till
HE		Help
IOC	I/O Command	
IOP	I/O (Physical)	
IOT	[<drive>] [,<controller>] I/O Teach	
LO	[<port_#>] [;<options>] = <text> Load S-Records	
MD	[<port_#>] <address> [<count>] [;<options>] (;DI ;S) Memory Display	
M[M]	<address>[;<options> Memory Modify	
MS	<address> <data>	Memory Set
[NO]PA	[<port_#>] Printer Attach	
OF		Display Offset Registers
PF	[<port_#>] Port Format	
PT		Power Up Test
SET	Set Date/Time	
ST		Self Test
TIME	Display Date/Time	
T[R]	[<count>] Trace	
TT	<address>	Trace Till
TA	[<port_#>] Terminal Attach	
TM	[<port_#>] [<exit character> <trailing character>]	Transparent Mode
VE	[<port_#>] [;<options>] = <text> Verify S-Records	
VI		Vector Initialize (coldstart)
.A0-.A7	[<expression>]	Display/Modify Address register
.D0-.D7	[<expression>]	Display/Modify Data register
.R0-.R7	[<expression>]	Display/Modify Offset register
.PC	[<expression>]	Display/Modify Program Counter
.SSP	[<expression>]	Display/Modify Supervisor Stack Pointer
.USP	[<expression>]	Display/Modify User Stack Pointer
.SR	[<expression>]	Display/Modify Status register
.DFC	[<expression>](68010)	Display/Modify Data Function Code
.SFC	[<expression>](68010)	Display/Modify Supervisor Function Code
.VBR	[<expression>](68010)	Vector Base Register
(Break)	Abort command or process	
(CTRL D)	Redisplay line	
(CTRL H)	Delete Character	
(CTRL S)	Suspend output, any character to continue (User Selectable)	
(CTRL X)	Cancel Command line	

MVME025

VMEmodule System Controller



The MVME025 System Controller offers the arbitration, monitor and utility functions usually required in a VMEbus system. In one module it combines clock generation, AC fail and system reset control, bus timeout and error control and bus arbitration eliminating the need for these functions on other modules. The module is especially useful in multiprocessor and intelligent DMA device applications. Figure 1 is a functional block diagram of the System Controller module.

- Provides VMEbus System Clock, AC Fail, System Reset and System Test Utility Functions
- Arbiter with Selectable Priority or Round Robin Select Modes
- Supports Four Levels of Bus Request/Allocation
- Supports VMEbus System Power Fail Timing for Orderly Power Down
- Selectable Power Fail Input Polarity
- Supports VMEbus System Power Restart Timing for Power Up and Self Test
- 16 MHz System Clock
- Bus Watchdog Timer — Period Selectable from 4 to 8192 microseconds
- Longword Access Error Detector
- Front Panel BERR* LED Indicator
- Front Panel SYSFAIL* LED Indicator
- Double High Eurocard Form Factor
- VMEbus Compatible

BUS ARBITER

Bus arbitration is used to prevent simultaneous access of the VMEbus by two masters and for scheduling bus allocation to optimize resource use in a multiprocessor configuration. The System Controller module has a header for jumper selection of one of two arbitration modes: priority or round robin select. In the priority mode when more than one bus request is pending, the arbiter will grant the bus to the requester of highest priority level. This mode is typically used in systems where bus loading by fixed rate masters such as DMA devices is nearly 100%. Where it is desirable to uniformly distribute the bus loading available to non-fixed data rate masters, the round robin select mode is used. This mode assigns the bus on a rotating priority basis.

AC POWER FAIL AND RESET PROVISIONS

A means for executing an orderly power down in the event of a power failure is a requisite for any computer system. Such a means is provided by the arbitration structure of the VMEbus which also provides a system reset line and an ac power fail line for use in conjunction with an external power monitor module whose function is to detect power failure and also reset the system on power up, initiating a power-up self test.

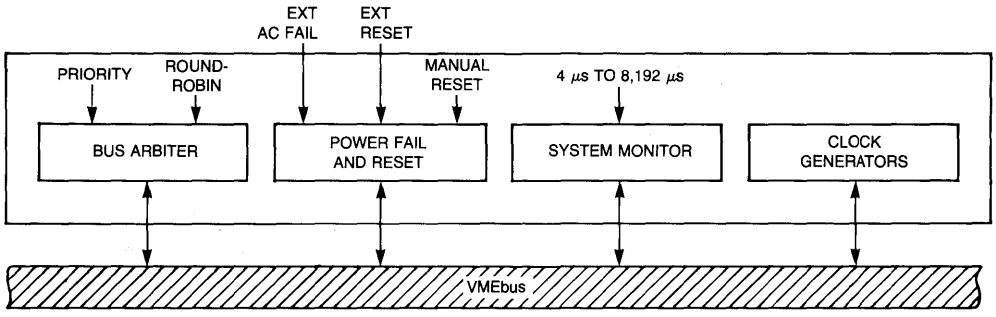
Both the system reset and ac fail lines are available at connector P2 on the system controller module which has logic for detecting the ac fail signal and for asserting the VMEbus ACFAIL* and SYSRESET* signals using proper timing. The module also has a header by means of which positive true or negative true ac power fail input signal polarity can be selected or the signal disabled.

Logic on the module, in addition to detection of the signal from the power monitor, also causes an approximately 500 millisecond assertion of the SYSFAIL* signal on 1) detection of the ACFAIL* assertion, 2) operation of the front panel push-button RESET switch and 3) execution of power up.

BUS WATCHDOG TIMER

To provide a mechanism for recovering from errors, the module has bus watchdog timer circuitry which terminates a data transfer cycle if a response is not received from a slave within a selected period of time. The circuitry monitors the AS*, DS1* and DS2* lines and starts the timer when any of these is asserted. Lines DTACK* and BERR* are also monitored and when either is asserted in a normal data transfer sequence, the timer is stopped. In an instance where timeout occurs before either is asserted, BERR* is driven low and remains low until AS*, DS1* and DS0* are driven high regardless of the state of DTACK*. The module has a jumper header for selection of a timer period in microseconds. A value equivalent to any of the twelve powers of two between 4 and 8192 can be selected.

FIGURE 1 — System Controller Module Block Diagram



LONGWORD ERROR DETECTOR

Circuitry which checks the accessed location against the size of the data being transferred can be a valuable tool for preventing one common software error. The module monitors the LWORD* and A01 lines and when LWORD* is asserted at the start of a longword transfer asserts BERR* if A01 is high (indicating an incorrect double word transfer). BERR* remains high until AS*, DS1* and DS0* are driven high regardless of the state of DTACK*.

FRONT PANEL INDICATORS

On its front panel, the module has LED indicators for the BERR* and SYSFAIL* lines each of which lights when the corresponding signal is asserted.

SYSTEM CLOCK

The System Controller provides a clock output to VMEbus specifications: SYSCLK. SYSCLK is a 16 MHz symmetrical system clock which can be used by other modules on the VMEbus for general clocking functions.

Electrical, Mechanical and Environmental Specifications

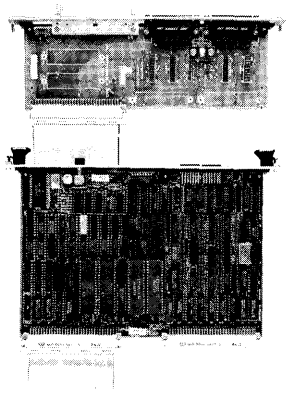
Characteristics	Specifications
Bus Allocation Speed	100 ns (typ) 130 ns (max)
Form Factor	Double High Eurocard
Power Requirements	+5 Vdc at 0.5 A (typ) +5 Vdc at 0.8 A (max)
Temperature Operating Storage	 0°C to +55°C −55°C to +85°C
Humidity	8% to 80% (non condensing)
Physical Characteristics PC Board Height PC Board Depth	 9.2 in. (234 mm) 6.3 in. (160 mm)

Ordering Information

Part Number	Description
MVME025	VMEmodule System Controller with Two Mode Bus Arbiter, AC Power Fail Circuitry, Watchdog Bus Timer and System Clock. Includes User's Manual.
MVME025/D1	VMEmodule System Controller User's Manual

MVME050
MVME701

System Controller and I/O Transition Modules



- System Controller
  - — 4-level Priority Bus Arbiter
  - — Power-up System Reset
  - — Front Panel System Reset
  - — System Clock Generator
  - — Serial Bus Clock Generator
  - — Bus Access Monitor/Time-out Bus Error Generator
- Time-of-day Clock (with optional battery back-up from +5 standby or MVME701)
- Printer Port (Centronics parallel port)
- Two Serial Ports (RS-232C Multi-protocol)
- Global Interrupter, for Processor to Processor Interrupts
- 2-Digit Front Panel Diagnostic Display (User Programmable)
- Green and Red LED's Showing VMEbus SYSFAIL* Status
- Front Panel 8-Segment Software Readable Switch
- Eight 28-Pin Sockets for EPROM/RAM
- Full 32-bit VMEbus Slave Interface for MVME050 Memory (EPROM/RAM sockets)
- A32/A24:D32/D16/D8 VMEbus Slave Interface for EPROM/RAM Sockets
- A16:D8 for I/O

The MVME050 System Controller Module is a combination system controller, system utility, and debug/diagnostic board. It is designed to implement functions that are only required once in a multi-processing system, such as bus arbitration, bus time-out, system clock generation, and time-of-day clock. This capability frees up space on the processor modules, avoiding the waste of having functions duplicated on every processor in the system (where only one can be enabled). Figure 1 is a Functional Block Diagram of the MVME050.

The MVME701 I/O Transition Module is an optional, Euro-format, double-high companion board to the MVME050. It may be attached by a ribbon cable to the MVME050 P2 connector, providing standard connectors for the System Controller's serial and parallel ports, and optionally, batteries for back-up of the MVME050's time-of-day clock. There are no active components on the MVME701.

BUS ARBITER

Bus arbitration is straight 4-level priority arbitration. All four VMEbus request lines are continuously monitored. As each request is put on the bus, its priority is compared with all other requests, with bus grant being given to the highest priority requester as soon as the current bus master releases the bus. Several requesters may be daisy-chained on a given priority line, producing positional priority at that level. The bus arbiter also issues a Bus Clear signal whenever there is a requester with higher priority than the current master.

SYSTEM RESET

On power-up or when the front panel reset switch is pressed, the MVME050 activates the SYSRESET* line of the VMEbus to reset all boards on the bus together.

As a further option, especially for OEMs configuring a VMEbus system inside of a larger system, a 4-pin connector on the companion MVME701 board allows attachment of a remote system reset switch, which may be used to activate SYSRESET*

SYSTEM CLOCKS

Both the 16 MHz VMEbus system clock and the VMSbus serial bus clock are driven by the MVME050. Both are derived from an onboard 32 MHz oscillator.

BUS TIME-OUT

To avoid bus hang-up and to tell a bus master that it has attempted to access an invalid location, the MVME050 can generate a bus error if no board has responded to a bus access within a pre-set time. The time-out time is user selectable between 2.5 and 160 microseconds (Jumpers).

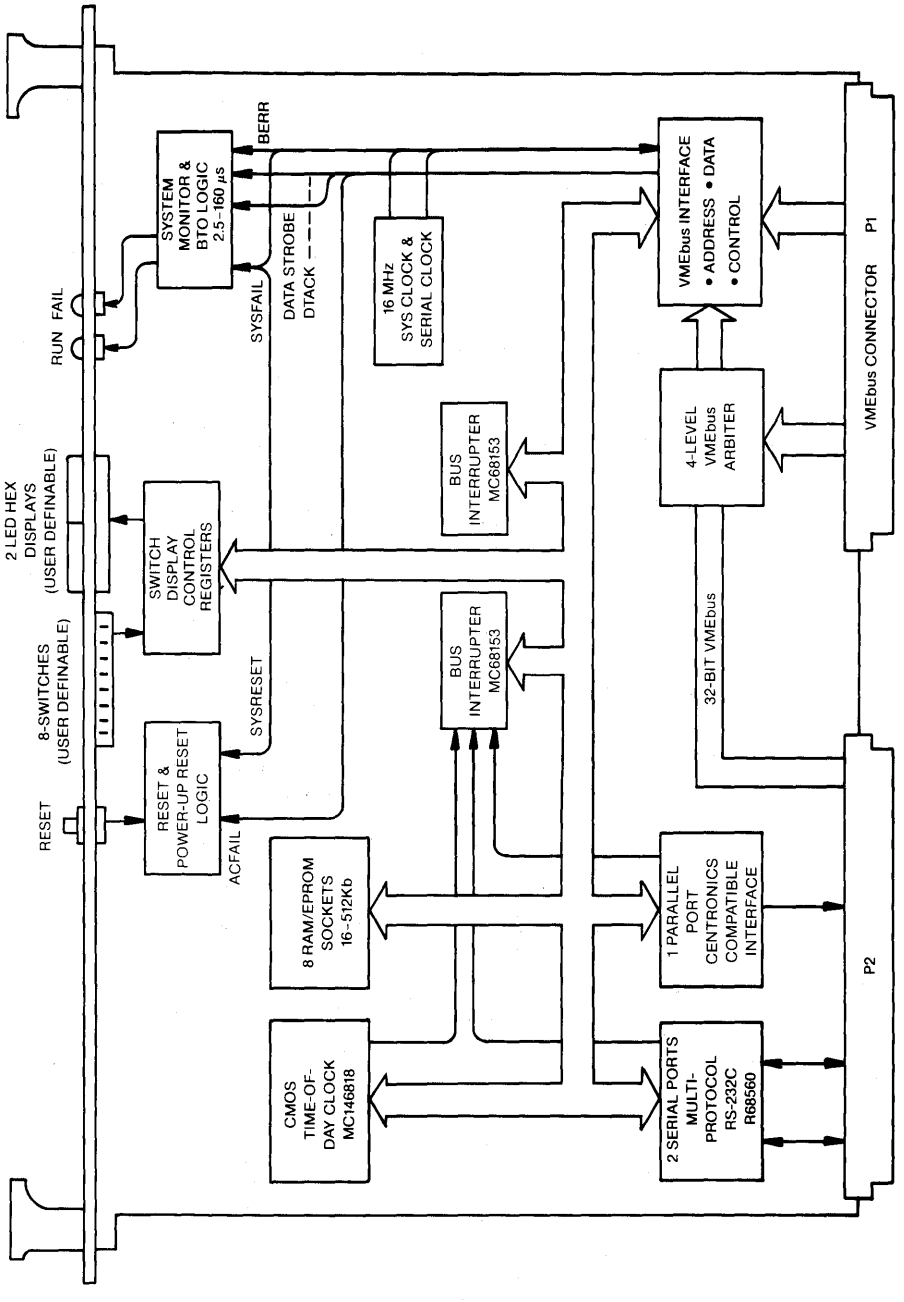


FIGURE 1 — MVME050 Functional Block Diagram

TIME-OF-DAY CLOCK

A Motorola MC146818 clock chip keeps track of date (year, month, day) and time (hours, minutes, seconds) with an accuracy based on a 0.001% quartz crystal. The clock can generate VMEbus interrupts at a software selected time of day, or daily at a selected time, or at periodic intervals ranging from 100 microseconds to 500 milliseconds. Optional power-fail backup operation is available either from batteries on the companion MVME701 I/O transition module or from the VMEbus +5 standby, as selected by jumpers.

PRINTER PORT

A buffered Centronics-type parallel port at the MVME050's P2 connector is provided for a system printer. When the MVME050's P2 connector is cabled to the companion MVME701 I/O transition module, a Motorola standard 50-pin Centronics parallel port connector is provided.

TWO SERIAL PORTS

Using the Rockwell R68560 Multi-protocol Communications Controller (MPCC), each serial port can operate in full duplex, asynchronous or synchronous protocols, at up to 19,200 baud (Async.) or 1 MHz (Sync.). Baud rate limitation will be determined by the restrictions of the software being used. Synchronous operations supported include several bit and character oriented protocols, such as SDLC, HDLC, X.25, DDCMP, X3.28, or ISO IS1745. With the MVME701 connected to the MVME050's P2 connector, the RS-232C signals are routed to two standard 25-pin DB-25 connectors on the front edge of the MVME701.

GLOBAL INTERRUPTER

A global four-channel interrupter allows one processor module to interrupt another in a multi-processor system. This could be used to signal process completion in a task sharing system or to signal semaphores which control access to shared resources in a multi-user system. The global interrupter function is implemented by the Motorola MC68153, a four-channel interrupter chip, with programmable levels and vectors for each channel. One channel may optionally be connected to the VMEbus SYSFAIL* line.

FRONT PANEL DISPLAYS

There are two front panel displays on the MVME050, two SYSFAIL* LED's, and a diagnostic display.

Two LED's on the front panel display the status of the VMEbus SYSFAIL* signal: a red LED for SYSFAIL* active, and a green LED for SYSFAIL* inactive.

A 2-digit hexadecimal display (capable of displaying the decimal digits zero through nine and the letters A through F, representing the values in the base 16 number system) may be used to display diagnostic information in case of system error, or any other user defined information. The display can show 100 decimal numbers from 00 to 99, or 256 hexadecimal numbers from hex 00 to hex FF.

SOFTWARE READABLE SWITCH

An 8-segment software readable switch on the MVME050 front panel allows an operator to choose one of 256 different binary codes to select different system modes or other functions a system designer may assign to the switches. The switches might, for example, be used to select system diagnostic modes or to select between different primary boot load disks.

RAM/EPROM

Eight 28-pin memory sockets on the MVME050 may be used to provide system RAM for common data structures such as semaphores or job queues. The sockets may also provide system EPROM for common test routines or boot loaders or a combination of both. Headers allow for configuration of each socket quad (group of four sockets addressed together by a 32-bit VMEbus access) as necessary to accommodate a wide variety of memory devices. Maximum capacity is 64Kb of RAM (eight 8K × 8 static RAM), or 512Kb of EPROM (eight 64K × 8 EPROMs), or a RAM/EPROM combination with 32Kb of RAM and 64Kb of EPROM.

The VMEbus interface to the memory on the MVME050 is designed in anticipation of the 32-bit VMEbus systems of the near future, with a full 32-bit interface (A32:D32). It remains compatible with today's 16-bit systems, supporting 24-bit addressing and 16- or 8-bit data transfers as well (A24:D16/D8). Switching between modes is automatic, depending on the mode in which it is accessed. Even in a multiprocessing environment with mixed 16- and 32-bit processors, this capability could provide common system memory for all.

NOTE: All devices in a quad must be of the same type.

ADDRESS MAP OF MVME050

RAM/EPROM Memory

Address is jumper-definable for EPROM and RAM independently (on boundaries equal to eight times the device size).

EXAMPLE: A quad of 2K × 8 RAMS (e.g. the Motorola MCM65116) could be set at VMEbus address: Hex 000000, or Hex 004000, or Hex 008000, etc.

I/O Memory

The short I/O address space occupied by the MVME050 is a 512-byte block (odd bytes only), which may be located on 512-byte boundaries starting at FF0000 (e.g. FF0000, FF0200, ... FF1000, etc.).

NOTE: Factory jumper position is FF1000, as shown below:

FF1000	RS-232C Serial Port 1	64 Bytes
FF1040	RS-232C Serial Port 2	64 Bytes
FF1080	Parallel Printer Port	32 Bytes
FF10A0	Read: Switch/Write: Display	32 Bytes
FF10C0	Interrupter 1 (MVME050 interrupts)	32 Bytes
FF10E0	Interrupter 2 (global interrupts)	32 Bytes
FF1100	Time-of-day Clock	128 Bytes
FF1180	Unused	
FF1200		

Mechanical and Environmental Specifications — MVME050

Characteristics	Specifications	
Configuration	EPROM/RAM	I/O
	DTB SLAVE: A24/A32 D8/D16/D32	A16 D8
	Arbiter: Priority, 4 level Interrupter: Any one of I (1)..I (7), Dynamic	
Form Factor	Double High Eurocard — Card dimensions: Height 9.2 in. (234 mm), width 6.3 in. (160 mm) — Front panel dimensions: Height 10.2 in. (261 mm), width 0.83 in. (21 mm)	
Power Requirements	+5 Vdc at 3.7 A (typ), 4.4 A (max) +12 Vdc at 140 mA (typ), 170 mA (max) -12 Vdc at 35 mA (typ), 42 mA (max)	
Environmental Limits Operating Temperature Storage Temperature Humidity	 0° C to 50° C inlet air temperature for 200 LFM air flow (forced air cooling) -40° C to 85° C 5% to 90% Relative Humidity (non-condensing)	

Mechanical and Environmental Specifications — MVME701

Characteristics	Specifications
Configuration	MVME050 transition card, no VMEbus connection
Form Factor	Double High Eurocard — Card dimensions: Height 9.2 in. (234 mm), width 3.1 in. (80 mm) — Front panel dimensions: Height 10.2 in. (261 mm), width 0.83 in. (21 mm)
Power Requirements	None (No active components on MVME701)
Environmental Limits Operating Temperature Storage Temperature Humidity	 0° C to 50° C Ambient -40° C to 85° C 5% to 90% Relative Humidity (non-condensing)

Ordering Information

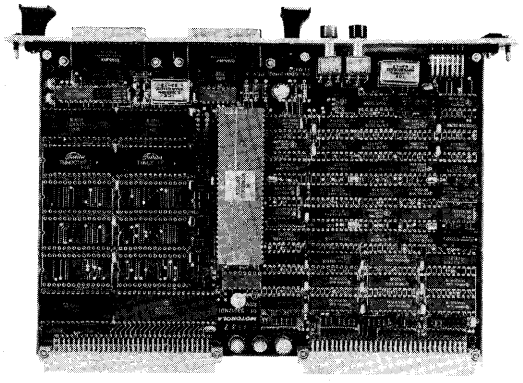
Part Number	Description
MVME050	System Controller Module & User's Manual
MVME701	I/O Transition Module for MVME050. Includes User's Manual
MVME050/D	MVME050 System Controller Module and MVME701 I/O Transition Module User's Manual

Related Documentation

Part Number	Description
MVMESYSAM/D	VME System Architecture Guide
68650N06	Rockwell, R68560 Product Description
MC68153/D	Motorola MC68153 VBIM Data Sheet
MC146818/D	Motorola MC146818 Time-of-day Clock Chip Data Sheet

MVME101

VMEmodule Monoboard Microcomputer



- High Performance 16-bit Monoboard Microcomputer
- MC68000 16-Bit MPU
  - — Sixteen 32-bit data, address and stack registers
  - — 14 addressing modes
  - — 16 megabyte direct addressing range
  - — Memory mapped I/O
  - — 56 powerful instruction types
  - — Operations on five data types including bit, byte, word, long word and BCD
  - — Provides interlock instruction for multiprocessor systems
  - — 256 multilevel vectored interrupts including internal exceptions, traps and external interrupts
  - — Architecturally optimized for efficient support of high-level languages
- VMEbus Compatible
- Double Eurocard Form Factor
- Up to 128K Bytes of On-Board ROM
- Up to 64K Bytes of On-Board RAM
- Full Operation Isolated from VMEbus (Multiprocessor Mode)
- Two RS-232C Serial I/O Ports
- 20 Programmable I/O Lines
- Triple Programmable 16-Bit Counter/Timer (accessible from off board)
- Hexadecimal LED Status Display
- Seven Jumper-Selectable Interrupt Priority Levels
- VMEbus Arbiter
- 0°–70°C Operating Temperature Range

The MVME101 Monoboard Microcomputer is a high performance processing module designed to function as a standalone microcomputer, as a single CPU/controller in a VMEbus system or as a single CPU element in a multiprocessor VMEbus configuration. This module features Motorola's MC68000 16-bit microprocessor with an address range of 16 megabytes.

Sockets are provided for up to 256K bytes of user-supplied memory. Synchronous and asynchronous serial communication at up to 19.2K baud is supported through two front panel ports and two independent 8-bit parallel communication channels are available at a rear panel connector. Access to a triple 16-bit programmable timer is also provided.

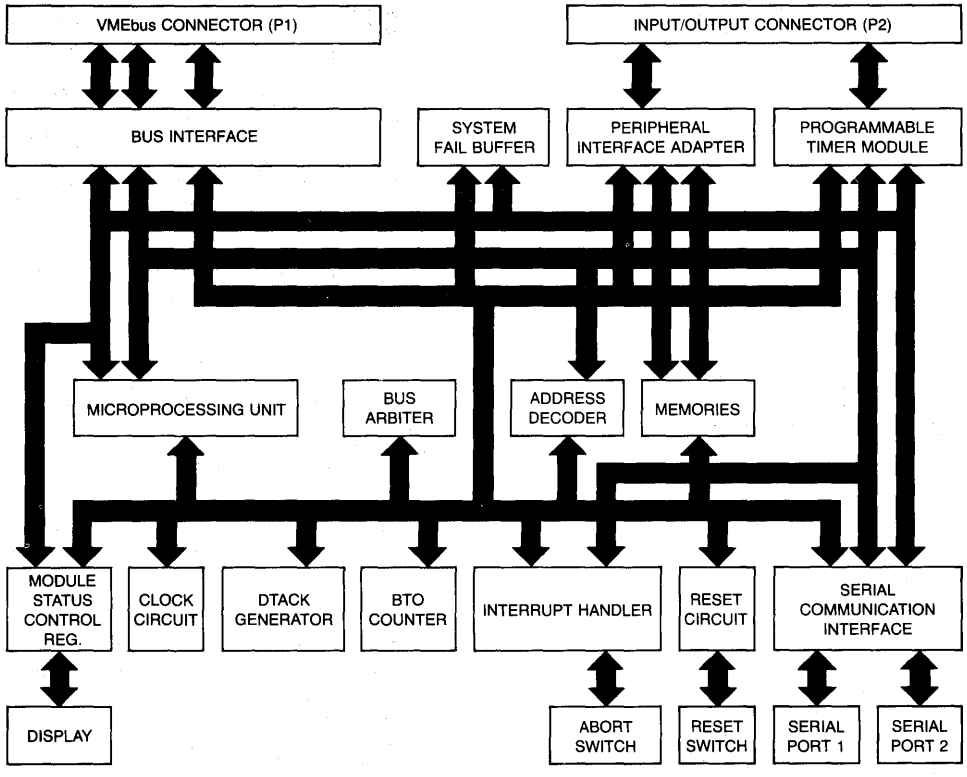
The MVME101 Monoboard Microcomputer in combination with a VMEmodule chassis, other VMEmodules and the VERSAdos Real-Time Operating System can provide a complete design environment which frees the system designer to develop the unique software/firmware required for an application. Figure 1 diagrams the major functional components of the MVME101 Monoboard Microcomputer.

DATA ORGANIZATION IN MEMORY

To provide asynchronous byte addressing over the eight megaword range accessible using address lines A01 through A23, the MC68000 separates its 16-bit data word into a lower data byte (D00–D07) and an upper data byte (D08–D15). Rather than address line A00, the external signals lower data strobe (corresponding to the lower data byte) and upper data strobe (corresponding to the upper data byte) are used to access a byte within a data word.

Accordingly, any memory block for the MC68000 must be made up of two identical blocks, one connected to the lower data lines and activated by the lower data strobe LDS*, the other connected to the upper data byte lines and activated by the upper data strobe UDS*.

FIGURE 1 — MVME101 Functional Block Diagram



MVME101 Memory Array

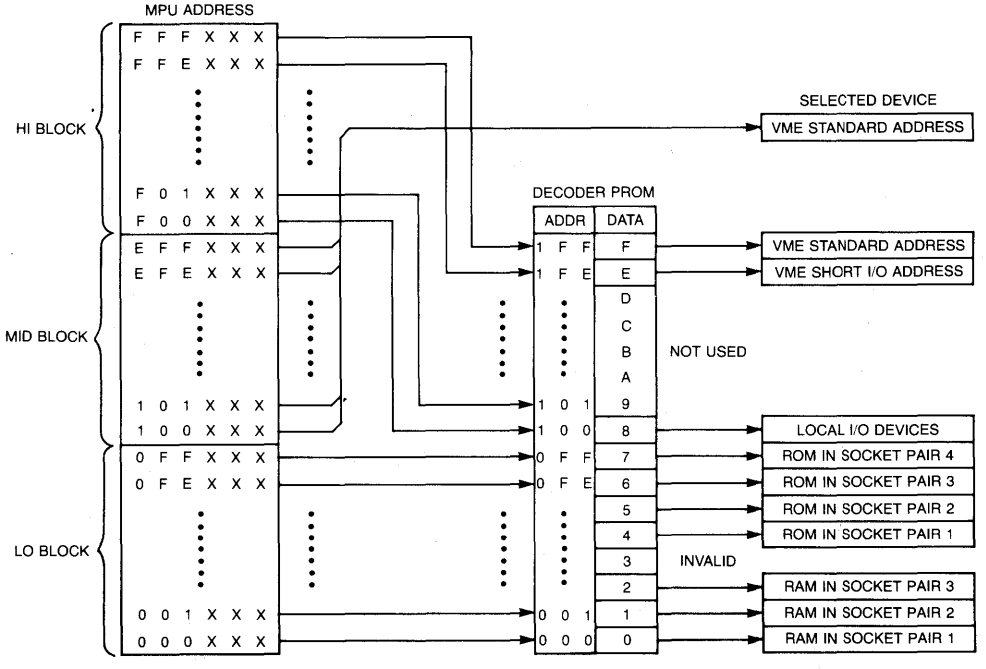
An array of eight 28-pin sockets connected as four pairs is provided on the module for user-supplied RAM or ROM. These can be any JEDEC standard byte-wide static memory in a 24- or 28-pin dual in-line package and of 2K, 4K, 8K, 16K or 32K size, of single +5.0 V operation, having high impedance (MOS) inputs and three-state outputs and meeting the timing requirements described in the MC68000 Monoboard Microcomputer User's Manual, MVME101/D1.

Address Map Configuration

The module has address decoding logic which divides the 16 megabyte address range of the MPU into high, middle and low blocks, as shown in Figure 2. All accesses by the MPU of the middle block, from 100000 to F00000, are directed off-board to the VMEbus. Accesses of the 000000 to 0FFFFF low range addresses and of the F00000 to FFFFFF high range addresses are decoded by a decoder PROM and on-board device selection logic. According to the internal ROM program, high and low block accesses are directed to global memory and memory-mapped devices, local I/O devices and the four local memory device socket pairs.

The decoder PROM, organized as 512 x 4 bits, divides the two megabytes of high and low block addresses into 512 4K

FIGURE 2 — Address Map Configuration



byte segments. Use of this programmable device provides a means of mapping local and global resources into separate memory areas as required by an application. The monoboard is shipped with the decoder PROM programmed to provide the map organization shown in Table 1. Local I/O devices and the module Control and Status Registers are mapped by the PROM as shown in Table 2.

The PROM program and the MVME101 Debug Package accommodate the requirement of the monoboard for ROM in memory socket pair #4. ROM is required since, following power up or reset, the MPU on the first four cycles fetches the supervisor stack and program counter values from the first eight locations of the memory socket pair #4 address area. Note also that the RAM, in socket pair #1, required by the debugger is reflected in the decoder PROM program map.

Local Memory Device Configuration

The module has four jumper headers which are used to configure the module for use with the specific type and size of memories installed in socket pairs one through four. Another header allows jumper selection of the proper timing for the type of ROM devices used.

VMEbus ARBITER/REQUESTER

A system which permits multiple master-type modules to share the data transfer bus must use a means of dealing in an orderly manner with concurrent requests for use of the bus. The VMEbus means is to use modules having bus request and/or bus arbitration capabilities and to designate one master type module as the system controller. The MVME101

TABLE 1 — MVME101 Address Map (As Shipped)

Address	Contents	Selected Devices
FFFFFF : FF0000	VMEbus Short I/O Addresses	Global I/O-devices
FEFFFF : FE1000	VMEbus Standard Addresses	Global Memory or Memory-mapped Devices
FE0FFF : FE0000	On-board I/O Registers (Only odd addresses used)	Local I/O-devices
FDFFFF : F04000	VMEbus Standard Addresses	Global Memory or Memory-mapped Devices
F03FFF : F00000	MVME101bug Debug Package or User-provided Program	2 x 8K bytes Local ROM in Memory Socket Pair 4
EFFFFF : : : : : : : : : : : 003000	VMEbus Standard Addresses	Global Memory or Memory-mapped Devices
002FFF : 002000	User Program/Data	2 x 2K bytes Local RAM in Memory Socket Pair 3
001FFF : 001000	User Program/Data	2 x 2K bytes Local RAM in Memory Socket Pair 2
000FFF : 000800	User Program/Data	2 x 2K bytes Local RAM in Memory Socket Pair 1
0007FF : 000400	MVME101bug Data/Stack	
0003FF : 000000	MPU Exception Vectors	

TABLE 2 — I/O Register Address Map (As Shipped)

Device	Address	Mode	Register
MCR	FE00F1	r/w	Module Control Register
MSR	FE00E1	r/w	Module Status Register
PTM	FE00DF	read	LSB buffer register
	FE00DF	write	Timer #3 latches
	FE00DD	read	Timer #3 counter
	FE00DD	write	MSB buffer register
	FE00DB	read	LSB buffer register
	FE00DB	write	Timer #2 latches
	FE00D9	read	Timer #2 counter
	FE00D9	write	MSB buffer register
	FE00D7	read	LSB buffer register
	FE00D7	write	Timer #1 latches
	FE00D5	read	Timer #1 counter
	FE00D5	write	MSB buffer register
	FE00D3	read	status register
	FE00D3	write	control register #2
	FE00D1	read	no operation
	FE00D1	write	CR20 = 1: control register #1
	FE00D1	write	CR20 = 0: control register #3
PIA	FE00C7	r/w	Section B control register
	FE00C5	r/w	CRB-2 = 1: Section B peripheral register
	FE00C5	r/w	CRB-2 = 0: Section B data direction register
	FE00C3	r/w	Section A control register
	FE00C1	r/w	CRA-2 = 1: Section A peripheral register
	FE00C1	r/w	CRA-2 = 0: Section A data direction register
PCI2	FE00B7	r/w	command register
	FE00B5	r/w	mode register #1 / mode register #2
	FE00B3	read	status register
	FE00B3	write	SYN1 register / SYN2 register / DLE register
	FE00B1	read	receive holding register
	FE00B1	write	transmit holding register
PCI1	FE00A7	r/w	command register
	FE00A5	r/w	mode register #1 / mode register #2
	FE00A3	read	status register
	FE00A3	write	SYN1 register / SYN2 register / DLE register
	FE00A1	read	receive holding register
	FE00A1	write	transmit holding register

module uses a programmable logic array to implement bus requester and arbiter functions which comply with the bus arbitration protocols of the VMEbus specification. Additional circuitry is used to meet the VMEbus timing and driving requirements.

BUS ARBITER

So that it can be used as the system controller in a VMEbus system, the MVME101 module has an option ONE single level arbiter which arbitrates requests on level 3. System controller operation requires that the module be placed in slot #1 of the VMEbus backplane to insure that the module is first in the daisy chain arbitration structure and has, therefore, the highest priority. The module also has a header from which a jumper is removed to disable the arbiter when the module is used at a lower priority in a multi-processor system.

2

BUS REQUESTER

The module has a type ROR (release on request) bus requester so that the module can be used in systems where maximum data transfer rate is essential. The requester monitors all four bus request lines and releases the VMEbus signal BBSY* only when another bus request is pending. This operation reduces the number of arbitrations required of a bus master.

Two means of requesting the VMEbus are provided by the module:

- the ROR mode in which the bus is automatically requested when the MPU starts either a VMEbus data transfer cycle or interrupt vector fetch and
- under program control by setting the Bus Block Transfer Request (BBTR) control register bit.

The latter method protects routines against interruption by other bus requests. With BBTR set, VMEbus is never released except when the module is used in a multilevel arbitration system. Then a higher level request causes the module arbiter to assert BCLR* resulting in a maskable auto vector interrupt request at the MPU providing the option of clearing BBTR under control of an interrupt service routine. To control idle state time in the software transparent ROR mode, the 128 microsecond Bus Request Time Out counter can be activated by setting an appropriate control bit in the module control register.

The module has two headers for jumper determination of the priority level at which the requester will operate. One allows connection of the module's bus request out signal to the VMEbus signal line of appropriate priority level. The other allows:

- connection of the VMEbus bus grant in line of the appropriate level to the bus requester,
- connection of the requester bus grant out signal to the appropriate VMEbus line
- connection of unused VMEbus bus grant input to bus grant output lines for propagation of these signals to the system modules having bus requesters of those levels.

VMEbus INTERFACE

VMEbus is characterized by the asynchronous bidirectional operation required for complex, high performance systems. The VMEbus interface on MVME101 supports operation in a multiprocessor system and the full 16 megabyte address range of the MC68000 MPU. Access to the backplane address, data and control lines is provided by the triple row, 96-pin VMEbus connector at the upper rear of the module. Pin assignments, connector physical characteristics and VMEbus signal and timing requirements are fully described in the VMEbus specification manual — MVMEBS/D1.

On the module, logic independent of the MPU generates the signal handshaking and timing required by the VMEbus data transfer protocol. Of the 14 address modifier codes defined by the VMEbus specification, a subset of six is supported by the module. These are listed in Table 3. Note that address modifier lines 3 and 5 are not driven by the interface logic but are kept in the high state by terminating resistors on the backplane.

Bus Supervision Counters

The module has two counters for supervising VMEbus accesses: the Bus Request Timeout Counter (BRTO) and the Data Transfer Timeout Counter (DTTO). Each can independently be enabled and disabled under software control by setting in the control register the bit corresponding to that counter.

If bit 6 in the control register is set at the time the MPU accesses an off-board location, the BRTO counter starts. After 128 microseconds, if the bus is not yet available bit 6 in the status register is set and the signal Bus Error is asserted.

If bit 7 in the control register is set at the time the MPU asserts a data strobe at the beginning of an off-board data transfer cycle, the DTTO counter starts. After 8 microseconds, if the data transfer is not yet acknowledged bit 7 in the status register is set and the signal Bus Error is asserted.

Operating Mode Control

The module has a header for jumper connection to the arbiter of the VMEbus system control signals required for

TABLE 3 — Address Modifier Codes

AM Code	Address Modifier 5	4	3	2	1	0	Function
3E	1	1	1	1	1	0	standard supervisory program access
3D	1	1	1	1	0	1	standard supervisory data access
3A	1	1	1	0	1	0	standard non-privileged program access
39	1	1	1	0	0	1	standard non-privileged data access
2D	1	0	1	1	0	1	short supervisory I/O data access
29	1	0	1	0	0	1	short non-privileged I/O data access

operating in the system controller configuration or in the standard non-controller configuration. Included are the output signal SYSCLK, the bidirectional signal SYSFAIL*, and the RESET* input and output signals. When the module is operated in the isolated mode, none of these signals is connected.

Reset and Halt Functions

System and local reset can be performed by either the power up reset circuitry or the reset switch. Local reset and system halt can be executed by the MPU. Connection or not to VMEbus of the signals SYSFAIL* and SYSRESET* required for operating in a particular mode is accomplished by jumper in the mode configuration header.

INTERRUPT HANDLER

The module interrupt handler circuitry manages all interrupt requests of local and system origin. It determines the pending interrupt request of highest priority and asserts a corresponding code on the three MPU interrupt priority lines. The module has two headers for jumper determination of which of the

TABLE 4 — Connector P2 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Description
Row A Pins		
1 through 32	GND	System Ground
Row C Pins		
1	+5.0 V	Supply
2	CB2	PIA Peripheral Control, Side B
3	CB1	PIA Interrupt Control, Side B
4	PB7	PIA Peripheral Data, Side B
5	PB6	PIA Peripheral Data, Side B
6	PB5	PIA Peripheral Data, Side B
7	PB4	PIA Peripheral Data, Side B
8	PB3	PIA Peripheral Data, Side B
9	PB2	PIA Peripheral Data, Side B
10	PB1	PIA Peripheral Data, Side B
11	PB0	PIA Peripheral Data, Side B
12	PA7	PIA Peripheral Data, Side A
13	PA6	PIA Peripheral Data, Side A
14	PA5	PIA Peripheral Data, Side A
15	PA4	PIA Peripheral Data, Side A
16	PA3	PIA Peripheral Data, Side A
17	PA2	PIA Peripheral Data, Side A
18	PA1	PIA Peripheral Data, Side A
19	PA0	PIA Peripheral Data, Side A
20	CA2	PIA Peripheral Control, Side A
21	CA1	PIA Peripheral Interrupt, Side A
22	+5.0 V	Supply
23	C3*	PTM Clock Input 3
24	O3	PTM Output 3
25	G3*	PTM Gate Input 3
26	C2*	PTM Clock Input 2
27	O2	PTM Output 2
28	G2*	PTM Gate Input 2
29	C1*	PTM Clock Input 1
30	O1	PTM Output 1
31	G1*	PTM Gate Input 1
32	+5.0 V	Supply

seven VMEbus interrupt lines, which of the four local I/O devices or which of the two VMEbus signals BCLR* and SYSFAIL* are used as user vector and auto vector interrupt request sources by the MPU. The non-maskable, auto vectored interrupt level 7 is not available but is reserved for software abort and ac power failure.

CONTROL REGISTER

The module control register is an 8-bit read/write register which resides in the local I/O segment of the memory map. Bits in the register can be set/cleared to control the module hexadecimal display, the SYSFAIL* signal output, the block transfer request, the bus request timeout counter and data transfer timeout counter.

STATUS REGISTER

The module status register is an 8-bit read/write register which resides in the local I/O segment of the memory map. Bits in the register indicate the current level of the VMEbus signals ACFAIL*, SYSFAIL*, ABORT*, BCLR*, BAV*, the local signal PCI1RXD* and whether or not a bus request timeout or a data transfer timeout has occurred.

VMEbus CONNECTOR P1

The electrical and mechanical characteristics of VMEbus connector P1 are fully described in the VMEbus Specification Manual MVMEBS/D1.

PERIPHERAL INPUT/OUTPUT CONNECTOR P2

Peripheral Connector P2 is a DIN 41612 connector with 64 pins (rows a & c) installed. Pin assignments and signal descriptions for Connector P2 are given in Table 4.

COMMUNICATIONS INTERFACES

Interfaces for both serial and parallel communications are provided on the module. An externally accessible triple programmable timer module is also provided. The two serial ports are accessible via two 25-pin connectors on the front panel. The programmable timer module and the parallel port are accessible via a rear panel connector, P2.

Serial Ports

The serial communication capability of MVME101 is derived from two MC68661C Enhanced Programmable Communication Interface devices. These devices support several synchronous or asynchronous protocols in full or half duplex mode, provide software selectable baud rates ranging from 50 to 19200 baud. On MVME101, both ports are RS-232C compatible and may, by jumper, be configured as data set or data terminal. Prior to beginning serial data communications, the MC68661C registers must be loaded with a set of mode and command bytes as described in the device data sheet.

Parallel Port

A universal means of interfacing peripheral equipment to the module is provided by a MC6821 Peripheral Interface Adapter (PIA). This device has two 8-bit bidirectional peripheral data buses and four control lines providing a general parallel communications capability for the control of various peripherals. Each of the peripheral data lines can be programmed as an input or output and each of the four control/interrupt lines can be programmed for operation in one of the several modes. The module has a header in which, to establish interrupt priorities, PIA interrupt output lines can be jumper connected to the desired autovectored interrupt request lines.

Programmable Timer Module

A generally useful timing function is provided by a MC6840 Programmable Timer Module (PTM) which contains three cascadable, 16-bit binary counters, three corresponding control registers and a status register. The module has a header for jumper configuration of the PTM inputs and outputs to obtain various modes of operation. The PTM can be programmed to generate module interrupts and/or output signals such as square waves, gated delay signals and signal pulses of controlled or modulated duration for use in event counting and interval or frequency measurement.

Software/Firmware Support

Motorola provides standard software packages to support VMEmodule Monoboard Microcomputers within the categories of Real-Time Executives and operating systems, and Debuggers/Loaders. The principal features of these software products are as follows:

RMS68K– Real-Time Multitasking System Software

- Memory Resident (ROMable)
- Physical (Channel) I/O
- Multitask Dynamic Scheduling
- Software and Hardware Interrupt Processing
- High Speed Interrupt Response
- Intertask Communication and Task Synchronization
- Dynamic Allocation and Management of RAM
- User Trap Handling
- Exception Processing
- Time Delay, Periodic Task Activation, Time-Of-Day
- Easy Addition Of User-Written Device Drivers
- Upward Compatible To Real-Time Disk Operating System
- Compatible with EXORmacs System Software
- Customization via SYSGEN

VERSAdos — Real-Time Disk Operating System

- Provides All Real-Time Multitasking Software Features of RMS68K
- Device Independent I/O and Logical I/O
- Wait and Proceed I/O
- Standard Device Drivers
- Multi-Level File Directories
- Shared File Access
- Dynamic or Contiguous File Space Allocation
- Random, Sequential, and Indexed Sequential File Access

VME101bug — Debug/Monitor/Loader Firmware

- Initialization
- Display/Change Memory
- Display/Change Registers
- Set and Clear Breakpoints
- Block Initialize
- Block Move
- Search for a (masked) Value
- Trace with Optional Instruction Count
- Downline/Upline Load
- Single-line Assembler/Disassembler

Hardware/Software Development Support

The recommended vehicles for developing 16/32-bit microcomputer systems based on a VMEmodule Monoboard Microcomputer are the EXORmacs MC68000 Development System and the VME/10 Microcomputer System in either the VERSAdos Real Time Operating System or SYSTEM V/68 Operating System environment. Both operating systems offer a complete set of high performance software development tools. Both support Motorola's 16/32-bit hardware emulators and bus state analyzer used for hardware/software integration and debugging in the target system.

Mechanical and Environmental Specifications

Characteristics	Specifications
Power Requirements	+5.0 Vdc at 180 mAdc (typ)
Temperature	
Operating	0°C to +70°C
Storage	−55°C to +85°C
Relative Humidity	0 to 95% (non-condensing)
Physical Characteristics	
PC Board only	
Height	9.2 in. (234 mm)
Depth	6.3 in. (160 mm)
Thickness	0.63 in. (16 mm)
PC Board & Front Panel	
Height	10.3 in. (262 mm)
Depth	7.4 in. (188 mm)
Thickness	0.8 in. (20.3 mm)
PC Board Form Factor	Double High Eurocard

Ordering Information

MVME101	VMEmodule Monoboard Microcomputer with the MC68000L8 MPU, two serial ports and two parallel I/O ports. Includes eight sockets for 2K to 32K-byte RAM/ROM devices, 16 bi-directional parallel I/O lines, three off-board-accessible programmable 16-bit counter/timers, hexadecimal LED status display and seven interrupt levels with bus arbitration.
MVME101/D	VMEmodule Monoboard Microcomputer User's Manual

MVME101BUG
MVME101BUGLF
MVME101BUGLC

VME101bug Debugging Packages for the MVME101 Monoboard Microcomputer

101bug Resident Package

- EPROM Resident System Debug Monitor
- 30 Powerful Commands
- Single-line Assembler/Disassembler for convenient Program Monitoring
- Full Speed Execution of System and User-Developed Programs Operating in the VMEmodule Monoboard Microcomputer System
- Virtual Terminal Capability for Up/Downline Load from an EXORmacs Development System or from any Host Computer
- Command Set Allows Access to all VMEmodule I/O, Control and Memory Facilities Plus the Full 16 Mbyte Direct Address Range of the VMEbus
- Includes Disk Controller Initialization and Disk I/O commands for the MVME315 Intelligent Floppy Controller/SASI Interface
- Includes Boot Facilities for Loading the VERSAdos O.S. and for Dumping VMEbus System RAM Contents to Disk.
- Includes all Required Installation and Operation Documentation

101bug Source and Relocatable Object Module Package

- Source and Relocatable Object Modules for 101bug on Diskette or Cartridge Disk
- Relocatable Object Modules Allow Users to Include Only the 101bug Items Needed in Their End System
- Source Modules Allow User Modification of 101bug as Desired

The MVME101 Monoboard Microcomputer debug package, 101bug, is available as two separate product offerings. 101bug is an EPROM-based resident package ready for installation and immediate use with the MVME101 Monoboard Microcomputer installed in a VMEbus backplane. Such a backplane is provided within Motorola's MVME900 Series Chassis. 101bug Source and Relocatable Object Modules are a separate product available on either VERSAdos compatible floppy disk or cartridge.

101bug provides a powerful evaluation and system debugging tool for VMEmodule Systems. The EPROM Resident Package will operate in 32K bytes of ROM space. 101bug uses the first 4K bytes of RAM storage for Interrupt vectors and temporary storage. The EPROM resident package is delivered in two 16K byte EPROMs. Table 1 lists the commands available to the user.

The package permits execution of system and user-developed programs operated in a MVME101 Monoboard Microcomputer system environment under complete operator control. 101bug may be utilized with a Monoboard Microcomputer in a standalone environment with only a user provided standard RS-232C asynchronous ASCII terminal. Alternately, it may be used with the second serial I/O port of the MVME101 connected to a host computer for up/downline loading of programs in Motorola "S" Record format. When connected to a host computer in this manner, the MVME101/101bug combination appears as a virtual terminal to the host operating system.

MVME101bug also provides program and operating system downloading facilities from floppy or hard disk into the VMEbus system RAM through the MVME315 Intelligent Floppy Controller/SASI Interface.

After loading, 101bug commands may be used to examine and modify memory, set breakpoints to run particular program segments, and track program progress. The user may set up and examine a variety of conditions using any of the powerful commands listed in Table 1, such as the Register Display/Set series and the memory block manipulation commands.

The Data Conversion command serves as an aid in examining and modifying data by converting hexadecimal to decimal, and decimal to hexadecimal. If corrections or program patches are required, these may be performed and checked in the VMEmodule. The user may send a corrected copy to the host computer files by using the Memory Dump command for upline load. Alternatively, memory contents can be saved on floppy or hard disk via the MVME315 Intelligent Floppy Controller/SASI Interface. Creating program patches may be aided by use of the Display Offsets command to assist with relocatable and position-independent code. The user may also copy all traffic to the serial port debug terminal on a printer attached to the MVME101 parallel port by use of the Attach Printer command. This may be useful for disk debugging following a debug session.

The user may communicate with the host computer as a terminal for purposes other than up/downline load by executing the Transparent Mode command. By using the Port Format command, the serial port may be reconfigured for such attributes as baud rate, stop bits and number of data bits.

Bootstrap load and dump commands permit the user to bootstrap from several device combinations through the MVME315 Intelligent Floppy Controller/SASI Interface. The drives currently supported are the 5¼" Winchester and the 5¼" Floppy and/or 8" Floppy. The Boot Dump command permits the user to write complete memory contents to a diskette/disk in bootstrap load format for subsequent use in boot loading. The IOT command permits the user to create the floppy/hard disk format required.

101bug may be used for debug in system environments which include the MVME101 Monoboard Microcomputer, other Motorola VMEmodules and user-developed VMEbus compatible modules.

The Source and Relocatable Object Module Packages provide users with the information to link 101bug into their specific systems in either modified or unmodified form. The Relocatable Object Modules are designed to permit creation of a load module with or without the "S" Record up/downline loader.

Source Modules permit the user to modify or customize any of the 101bug functions as desired.

TABLE 1 — 101bug Commands

Command	Description
MD <addr>[<count>]	Memory Display/Disassembly
MM <address>[;<opts>]	Memory Modify/Disassembly/Assembly
MS <address><data. . .>	Memory Set
.A	Display All Address Registers
.A0 – .A7 [<expression>]	Display/Set Address Register
.D	Display All Data Registers
.D0 – .D7 [<expression>]	Display/Set Data Register
.PC [<expression>]	Display/Set Program Counter
.SR [<expression>]	Display/Set Status Register
.SS [<expression>]	Display/Set Supervisor Stack Pointer
.US [<expression>]	Display/Set User Stack Pointer
DF	Display Formatted Registers (All)
BF <address1><address2><word>	Block Fill (with 16-bit data word) Memory
BM <address1><address2><address3>	Block Move
BS <address1><address2><data>[<mask>][;<opts>]	Block of Memory Search
BI <address1><address2>	Block Initialize
BT <address1><address2>	Block Test of Memory
DC <expression>	Data Conversion
OF	Display Offsets
.R0 – .R6 [<expression>]	Display/Set Relative Offset Register

TABLE 1 — 101bug Commands (continued)

Command	Description
BR [<address>[;<count>]]	Breakpoint Set (up to 8)
NOBR [<address><address>. . .]	Breakpoint Remove (any or all)
GO <address>]	Execute Program
GT <breakpoint address>	Go Until Breakpoint (sets temporary breakpoint)
GD [<address>]	Go Direct (No Breakpoint or Track Set, and no Exception Vector Changes)
TR [<count>]	Trace (set for number of instructions)
TT <breakpoint address>	Trace to Temporary Breakpoint
PA	Printer Attach (Print as well as display)
NOPA	Reset Printer Attach
PF [<port number>]	Port Format (set Serial Port Attributes)
TM [<exit character>]	Transparent Mode (Two serial ports transparently connected)
HE	HELP (Display VMEbug commands)
DU <address1><address2><text. . .>	Dump ("S" Record Upline load)
LO [;<opts>] [=text]	Load ("S" Record Downline load)
VE [=text]	Verify ("S" Record Downline load verify)
BD [<device>] [,<controller>]	Boot Dump
BH [<device>] [,<controller>]	Boot Halt
BO [<device>] [,<controller>] [,<string>]	Boot Operating System
IOP	Disk I/O Physical
IOT	Disk I/O "Teach"
Command Line Edit and Control Functions:	
(BREAK)	Abort Command
(DEL)	Delete Character
(CTRL-D)	Redisplay Line
(CTRL-H)	Delete Character
(CTRL-W)	Suspend Output*
(CTRL-X)	Cancel Command Line
(cr)	Send Line to Memory

*When (CTRL-W) is used, the user can cause the output display to continue by entering any character.

Ordering Information

Part Number	Description
MVME101BUG	101bug, the MVME101 Monoboard Microcomputer System Debug Package, includes EPROM set* and User's Manual.
MVME101BUGLF	Source and Relocatable Object Modules for the 101bug system on VERSAdos Diskette for the EXORmacs Development System.* Includes User's Manual.
MVME101BUGLC	Source and Relocatable Object Modules for the 101bug system on VERSAdos Cartridge Disk for the EXORmacs Development System.* Includes User's Manual.
MVME101BUG/D	MVME101bug Debugging Packages User's Manual

*The MVME101BUG EPROM set is copyrighted by Motorola and may be copied only under prior written agreement from Motorola. MVME101BUGLF and MVME101BUGLC Sources are copyrighted and licensed by Motorola. They may be obtained only under the required license agreement with Motorola.

MVME105
MVME104
MVME106
MVME107

Advance Information

VMEmodule™ VMEbus Processor Module

- Low Cost Single Board Computer Offered in Four Versions

Standard Features, on all Four Versions:

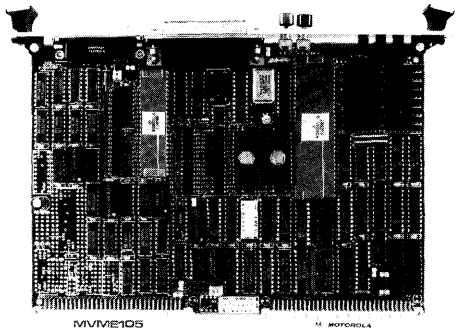
- A24/A16:D16/D08(E0) VMEbus Interface
- MC68010 CPU at 10 MHz
- 512Kb Dual Access DRAM, with Zero-Wait Cycle Read, One Wait Cycle Write
- Two EPROM Sockets, for up to 128Kb of Zero-Wait Cycle EPROM
- Two Additional Sockets, for EPROM, EEPROM, or SRAM
  - — Can also be used for CMOS RAM or clock/calendar with on-chip battery!
- Async/Sync RS-232-C Serial Port on Front Panel
- Async/Sync RS-485/422 Serial Port on P2 Connector
- 8-Bit Parallel I/O Port with Four Handshake and Five Control Lines (for Centronics printer or parallel I/O)
- Four Timers: 24-Bit Timer with 125 ns Resolution, Local and VMEbus Timeout, and Watchdog Timers
- Global Control/Status Register (CSR)
- Software Readable 8-Bit Switch
- Single Level Arbiter and System Controller
- Optional Debug Monitor available for each VMEbus Processor Module

Special Features, Unique to Each Version:

- Each of the four versions has a different I/O interface on its P2 connector:
  - — MVME104: Motorola I/O Channel Bus
  - — MVME105: — No Connect —
  - — MVME106: 5-1/4" Floppy Disk
  - — MVME107: SCSI Bus Interface
- Optional Debug Monitor Available for each Processor Module

The MVME105 family of VMEbus single board computers are designed for embedded controller and small real-time system applications. They have a common architecture and offer standard features sufficient to satisfy the requirements of most applications. Special I/O on the P2 connector optimizes each of the four versions for particular markets and applications.

The **MVME104** with Motorola's I/O channel bus interface may be used as the controller for a complete system



based on the industrial I/O peripherals available for this bus (MVME400 and MVME600 series modules).

For EPROM based systems or those using VMEbus for mass storage, the base board **(MVME105)** has no special I/O interface. As such, it is also the least expensive board in the family.

Small systems requiring only a small amount of mass storage can use the **MVME106** with a 5-1/4" floppy disk interface. Even larger systems with other mass storage might use the MVME106 to add floppy disk inexpensively.

For those with more elaborate I/O and mass storage requirements, the **MVME107** with its standard small computer systems interface (SCSI) bus provides access to high speed disk, tape, and other peripherals for a small additional cost. This will become increasingly attractive as more peripheral vendors offer their products with an embedded SCSI interface.

The common architecture means easy portability of user application software from one CPU to any of the others as system requirements change. One might, for example, develop programs on the MVME107 (with its SCSI mass storage), and then use an MVME105 or MVME106 in the production system. Or, if a working MVME105-based system develops problems at the end customer's site, the service technician can simply replace it with a MVME106 or MVME107 to run diagnostics on the rest of the system and the software.

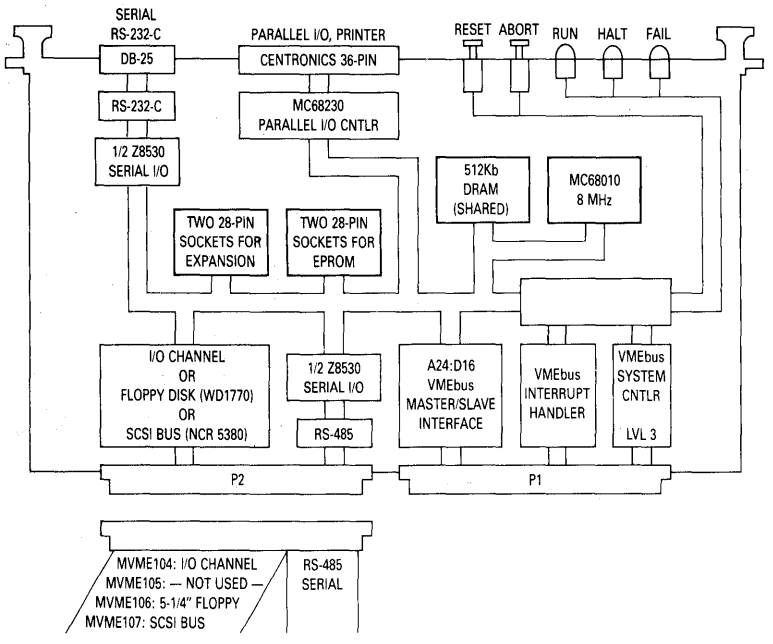


Figure 1. MVME105 Series Block Diagram

MICROPROCESSOR

All four members of the MVME105 family are based on the MC68010, a microprocessor of proven architecture, optimized for efficient support of high-level languages and secure operating systems.

MC68010 features include:

- 32-Bit Internal Data, Address, and Stack Registers
- 14 Addressing Modes
- 16Mb Direct Addressing Range
- 57 Instruction Types
- Operation on Five Data Types: Bit, Byte, Word, Longword, & BCD
- 256 Multi-level Vectored Interrupts
- Fast "Loop-mode" Execution for Frequently Needed Sequences

ON-BOARD MEMORY

512Kb of dynamic RAM (DRAM) and sockets for up to 256Kb of EPROM provide sufficient memory for large single-board-computer (SBC) applications. The DRAM is dual-access, meaning that it can be accessed by the on-board CPU or by other CPU's and peripherals from the VMEbus. There is no parity checking on the DRAM.

Read accesses of DRAM by the on-board CPU take four CPU clock cycles. This is the maximum possible speed, also known as ZERO WAIT CYCLE operation. Write cycles are slower, taking five CPU clock cycles (one wait cycle).

Access from the VMEbus is slightly slower, since the on-board CPU must complete whatever cycle is in process and release control of the DRAM to the VMEbus. Access times from the VMEbus will range between approximately 650 ns to 1200 ns, depending on what operation the local CPU was performing when the request was made.

From the point of view of the on-board CPU, the DRAM normally occupies the bottom of the memory map, 000 000 thru 07F FFF (hexadecimal). See Table 1. Optionally, the DRAM can be mapped to address E80 000 thru EFF FFF if the user wishes to access the VMEbus at address zero.

From the point of view of a device on the VMEbus, the DRAM may be set to occupy any address range in the 16Mb address space of the A24 VMEbus (on a 512Kb boundary). Switches are provided for this selection. Only the DRAM and a global control/status register are accessible from the VMEbus. Other on-board resources (EPROM, serial ports, etc.), are all 'private,' and are accessible only from the on-board CPU.

There are four 28-pin memory sockets on the board. Two of these are dedicated for EPROM, and accept either 32K x 8 or 64K x 8 devices. Accessed as a single 16-bit wide memory, they provide up to 128Kb of EPROM memory. With 300 ns or faster EPROMs, these memories also provide ZERO WAIT CYCLE operation. The board is shipped with these sockets empty, but in a typical application, as with the VERSAdos operating system, they would be used for the debug/monitor firmware.

The other two sockets, also accessed as a single 16-bit wide memory, can be used for additional EPROM (for a total of 256Kb), but also support read/write access for use with static RAM, nonvolatile memory like EEPROM, CMOS RAM with built-in battery, or special-purpose devices which combine RAM with a clock/calendar function. A logical configuration might be a RAM with on-chip battery in one socket and a RAM/clock/calendar with on-chip battery in the other.

SERIAL I/O

The Zilog Z8530A Serial Communications Controller provides two multiprotocol serial ports for the MVME105 family of microcomputers. On the front panel is an RS-232-C compatible serial port with a standard DB-25 connector. Jumpers can be used to configure the port for DCE or DTE operation, and for synchronous or asynchronous protocol. The second serial I/O port is brought out through the P2 connector on the module's back edge. It is wired as an RS-485 differential interface, and jumpers are used to configure the interface as master or slave, and to select 8-, 4-, or 2-wire operation.

The RS-485 interface is an extension of the RS-422 differential interface, with greater current drive and tri-state capability. It can be used for point-to-point communications, or in a multi-drop configuration with several RS-485 interfaced devices sharing a common serial line. Differential drive is much less sensitive to noise than RS-232-C and is particularly well suited for use in an electrically noisy environment and/or for very long cable lengths at high data transfer rates.

A typical system configuration might be a series of small control or data collection nodes, each with an MVME105 single board computer as the local intelligence, and all linked together and to a central host computer by means of an inexpensive two wire RS-485 multi-drop line.

The Z8530 serial port controller chip offers the following features:

- Software Programmable Baud Rates from 50 Baud to more than 64K Baud
- Software Selectable NRZ, NRZI, or FM Data Encoding
- Asynchronous Mode, with Software Programmable:
  - — 5–8 bits per character
  - — 1, 1.5 or 2 stop bits per character
  - — BREAK detection and generation
  - — Parity, overrun, and framing error detection
- Synchronous Mode, with:
  - — Internal character synchronization on one or two characters
  - — CRC generation and checking with CRC-16 or CRC-CCITT
  - — Digital phase-locked loop for clock recovery
- Local Loopback and Auto Echo Features

PARALLEL I/O

A full Centronics-compatible, 8-bit parallel I/O port is provided on each of the MVME105 family of CPU's. Implemented as a standard 36-pin shielded Centronics connector on the front panel, it can directly support parallel printers or can be used for general parallel I/O. There are eight I/O data lines, four handshake lines and five control lines, all controlled by means of an on-board MC68230 Parallel Interface/Timer.

TIMERS

Four timers are implemented on the module. Three are dedicated to error recovery functions: local bus time-out, VMEbus time-out, and watchdog timing. They are designed to aid the CPU in regaining control if an addressed device on the local bus or VMEbus fails to respond, or if an executing program "gets lost" (watchdog timer). The fourth timer, a 24-bit timer running at 8 MHz (125 ns resolution) is available for use by the operating system or application software.

A special clock/calendar device (such as the THOMPSON/MOSTEK MK48T02/25) may be installed in one of the auxiliary memory sockets, if desired.

SYSTEM CONTROLLER / BUS REQUESTER

The MVME105 series microcomputers can work as the VMEbus system controller in the slot 1 position in a VMEbus chassis. A system controller is responsible for VMEbus arbitration (in this case, single level arbitration on level 3), SYSCLK generation (the 16 MHz reference clock on VMEbus), system reset (SYSRESET*) generation on power-up or from the front panel, and VMEbus time-out. A switch disables these functions when another board is acting as system controller.

Bus request logic on the MVME105 boards allows the modules to request access to the VMEbus on any of the four VMEbus request levels (jumper selectable). Only level 3 may be used if the system controller functions are enabled. The bus release protocol is release-on-request (ROR), and early release of bus busy is supported to allow concurrent arbitration for optimum bus utilization.

VMEbus INTERFACE

The MVME105 series modules all have a 24-bit address / 16-bit data VMEbus interface (A24:D16), for both master and slave operations. They may be used in a standard A24:D16 VMEbus system with a P1-only backplane, or in an extended A32:D32 system with a combined P1 and P2 backplane. If used in a 32-bit system, the designer must recognize that they will still respond only to transfers with "A24" or "A16" address modifier codes.

There is no connection to the extended VMEbus address and data signals (on row b of the P2 connector), but P2 rows a and c are used for I/O, so a P2 backplane (if used) must have pass-thru pins for this purpose.

There are, in a sense, three interfaces between the MVME105 series CPU's and the VMEbus.

1. What the local CPU sees of VMEbus when it is VMEbus master.
2. The dual-access DRAM as it appears on VMEbus to other VME masters.
3. The 105's CSR space as it appears on VMEbus to other VME masters.

As VMEbus master, the MVME105 family module can access most of the lower 16Mb of VMEbus address space. As shown in the memory map in Table 1, "VMEbus" occupies all of the memory space except where on-board DRAM and other local resources are decoded. In addition, the VMEbus address modifier lines can be manipulated to access any of the alternate VMEbus memory spaces reserved for user-defined applications.

CONTROL/STATUS REGISTER

The control/status register (CSR) is a special-purpose, dual-ported, 16-bit memory that exists both in the local CPU's memory space and in the short I/O space on VMEbus. It is used to control certain functions on the MVME105 family module, to allow the status of those functions to be read both by the local CPU and other devices on VMEbus, and to provide a quick means for a VMEbus device to signal the local CPU. As seen from the VMEbus, the CSR register resides at one of two switch selectable short I/O space addresses relative to the selected address of the 512Kb DRAM (module base address). Seen from the local CPU, the CSR resides at a fixed address in local memory. Figure 2 shows the CSR bit functions.

Control functions in the lower 8 bits include the watchdog timer (enable/disable), interrupt enable (enable all/disable all), the system fail signal to VMEbus (drive active/not driven), and two of the address modifier control lines mentioned above. These can all be read or written by the local CPU, but are read-only for a device on VMEbus.

The upper 8 bits (bits 8..F) are read/write for both VMEbus and the local CPU. Primarily intended to allow a VMEbus device a means other than VMEbus interrupts to signal the local CPU, it works by writing values with a '0' in any of the bits 8..B — which generates a local interrupt on level 2 or level 7. The other bits can be used as desired, possibly as a source identifier or command code. One use of this mechanism might be to implement the buffered pipe protocol, where the 'signal' tells the local CPU to look for data in global memory.

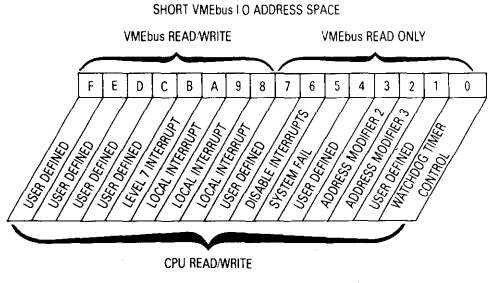


Figure 2. Module CSR Register

I/O CHANNEL (MVME104 only)

The I/O channel is an 8-bit data, 12-bit address bus for specific application system expansion. Modules from Motorola's MVME400 series and MVME600 series can be added to provide a wide range of functions including serial and parallel port expansion, Winchester and floppy disk (with the M68RWIN1), 9-track tape, analog I/O, digital I/O, and even ac power I/O. Except for the M68RWIN1, all of these conform to the single high Eurocard form factor, and are supported with cables and backplanes for use in VMEbus chassis.

Appearing in the MVME104's local memory space as a 4Kb block, the I/O channel is easy to use and is fully supported by the VERSAdos Real-Time operating system.

FLOPPY DISK INTERFACE (MVME106 only)

The MVME106 has a direct interface to Shugart SA400 compatible 5-1/4" floppy disks (up to 4 daisy-chained disks). Implemented with the Western Digital WD1770 disk controller, it supports 125Kb/s FM or 250Kb/s MFM formats for single or double density disks, single or double sided, 48 or 96 tracks per inch.

SCSI bus (MVME107 only)

The **S**mall **C**omputer **S**ytems **I**nterface (SCSI) bus is quickly being accepted as the standard bus interface for peripheral I/O. Designed as an extension of Shugart's SASI bus, SCSI is a medium performance bus supporting data transfer rates up to 1.5Mb/s (asynchronous) and up to seven peripherals. Supported peripherals under the VERSAdos operating system include floppy disk, and Winchester disk. Other peripherals available with a SCSI bus interface that could be used with the MVME107 (but not currently supported with VERSAdos drivers) include SMD disks, optical disks, streaming tape, 9-track tape, and even system-to-system communications.

Using the NCR 5380 SCSI bus controller chip, the MVME107 implements a single ended, arbitrating, asynchronous SCSI bus interface as defined in revision 16 of the ANSI X3T9.2 committee specification. It can operate in either the initiator or target roles, and supports disconnect/reconnect and the common command set.

Data transfer between SCSI bus and the on-board memory (or VMEbus memory) is handled by the local CPU using what NCR calls the 'psuedo-DMA' mode of the NCR 5380. Running at 10 MHz, the MC68010 CPU can transfer data over SCSI bus at approximately 700Kb/s.

Table 1. Physical Address Map
(View from on-board CPU, typical configuration)

000 000 thru 000 007	Reset vector (from EPROM) for the first four memory cycles after RESET, RAM thereafter.
000 000 thru 07F FFF	Dynamic RAM
080 000 thru EFF FFF	VMEbus
F00 000 thru F1F FFF	EPROM (2 sockets, possibly repeated)
F20 000 thru F3F FFF	Auxiliary RAM/ROM/EEPROM/clock sockets
F40 000 thru FBF FFF	VMEbus
FC0 000 thru FCF FFF	Local resources (Serial & Parallel ports)
FD0 000 thru FDF FFF	Control/status register (CSR), I/O channel, etc.
FF0 000 thru FFF FFF	VMEbus short I/O space

SOFTWARE SUPPORT

A debug monitor for the MVME105 series of VMEbus Processors is available in two EPROMs that plug into sockets on a module. To allow customization, it is also available as source code on 5-1/4" diskettes readable on any system running the Motorola VERSAdos Real-Time Operating System.

A powerful, sophisticated tool, MVME105bug, or 105bug for short, allows full speed execution of system and user programs under complete operator control. It offers 35 powerful commands for application program development and modification and for hardware diagnosis and debugging of systems based on any of the processor modules.

Facilities for loading an operating system and programs from floppy or hard disk into a target system are included in 105bug. Supported disk and tape controllers for each module are shown in Table 2.

Test and diagnostic capabilities in 105bug include a power up sequence self test that tests the MC68010 Microprocessor, ROM and RAM resources required to bring up the monitor and a test invoked by user command that performs more intensive testing.

A TRAP #15 handler is included in 105bug that facilitates calls from a user program to input, output and other useful 105bug functions and also facilitates, in the MVME107, communication with the MCR5380 device used for the SCSI interface implementation. A complement of functions is provided that can be used to obtain read, write and other activities from disk drives connected to controllers on the SCSI bus.

Table 2. Controllers Supported by MVME105bug

	MVME104	MVME105	MVME106	MVME107
MVME319	X	X	X	X
MVME320A	X	X	X	X
MVME360	X	X	X	X
Local Floppy			X	
SCSI				X
MVME350	X	X	X	X

X = Supported

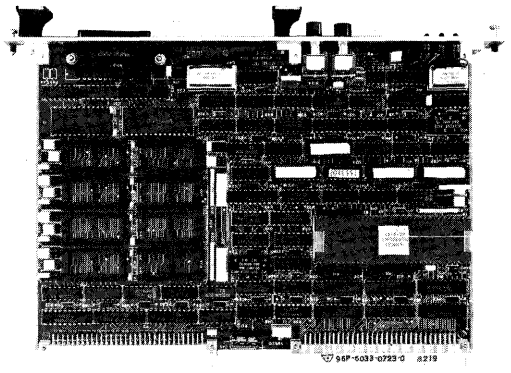
MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
VMEbus Configuration	DTB Master A24/A16:D16/D08(EO) DTB Slave A24/A16:D16/D08(EO) Single level arbiter (level 3) Requester: Any one of R(0)..R(3) Static ROR, Early Release Interrupt handler: Any of IH(0)..IH(7), Static
Form Factor	Double High Eurocard
Physical Dimensions Card dimensions Front panel	 Height 234mm (9.2"), Depth 160mm (6.3") Height 261mm (10.2"), Width 21mm (0.83")
Power Requirements MVME105 MVME104/106/107	 +5 V, 5 A typ ±12 V Preliminary Values +5 V, 5 A typ ±12 V
Environmental Limits Operating temperature Storage temperature Humidity	 0°C to 55°C inlet air temperature with forced air cooling 0°C to 40°C ambient, convection cooling −40°C to 85°C 5% to 90% relative humidity, non-condensing

ORDERING INFORMATION

Part Number	Description
MVME105	VMEbus single board computer module featuring 10 MHz MC68010 CPU, 512Kb shared DRAM, sockets for up to 256Kb EPROM, RS-232-C serial port, RS-485 serial port, 8-bit parallel port, 24-bit timer, and system controller.
MVME104	Same as MVME105, but with I/O channel interface.
MVME106	Same as MVME105, but with 5-1/4" floppy interface.
MVME107	Same as MVME105, but with SCSI bus interface.
MVME105BUG	105bug, the Debug Monitor for the MVME104, MVME105, MVME106 and MVME107 VMEbus Processor Modules. Includes set of two 32Kb EPROMs and User's Manual.
M68VIXSBG105	105bug Source Modules on 5-1/4" Diskette readable on any system running the Motorola VERSAdos Operating System. Includes User's Manual.

VMEmodule Monoboard Microcomputer



- High-Performance 16-Bit Monoboard Microcomputer
- MC68000 16-bit MPU
  - — 16 32-Bit Data, Address and Stack Registers
  - — 14 Addressing Modes
  - — 16 Megabyte Direct Addressing Range
  - — Memory Mapped I/O
  - — 56 Powerful Instruction Types
  - — Operations on Five Data Types Including Bit, Byte, Word, Long Word and BCD
  - — Provides Interlock Instruction for Multiprocessor Systems
  - — 256 Multilevel Vectored Interrupts Including Internal Exceptions, Traps and External Interrupts
  - — Architecturally Optimized for Efficient Support of High-Level Languages
- VMEbus Compatibility with Bus Arbitration Logic
- Double Eurocard Form Factor
  - — Incorporating High-reliability Pin/Plug Type Connectors
- 8 MHz Version Available, Customer Upgradeable to 10 MHz
- RS-232C Serial Port Configured as DCE, may be Connected to DTE for Debugging
- I/O Channel Support for Off-board Serial, Parallel I/O and A/D, D/A, AC and DC Switching and Mass Storage Functions
- Eight 28-Pin Sockets for User Provided 2, 4, 8, 16, 32K x 8 ROM/PROM/EPROM or 2, 4, 8K x 8 RAM Devices
- Zero Wait State Operation at 8 MHz With 150 ns or Faster Static on Board RAMs
- Up to 7 Levels of Interrupt Priority May Be Jumper Selected
- Three 16-Bit, Cascadable Programmable Timer/Counters, Jumper Selectable MPU E Clock or Baud Rate Clock Input
- VMEbus Requester and Interrupt Support
- Pushbutton RESET and ABORT Controls
- FAIL, HALT, and RUN LED Status Displays
- 0°C–70°C Operating Temperature Range

FUNCTIONAL DESCRIPTION

The VMEmodule Monoboard Microcomputer is a high performance processing module, designed to function as a standalone microcomputer, as a single CPU/controller in a VMEbus system, or as a single CPU element in a multiprocessor VMEbus configuration. This module features Motorola's MC68000 16-bit microprocessor with a total address range of 16 megabytes. Sockets are provided to accommodate up to 256K bytes of user-supplied memory. Full support is provided for the Motorola I/O Channel which provides access to a large variety of peripheral and industrial I/O functions. An on-board serial communications port is also included.

The MVME110-1 Monoboard Microcomputer in combination with the VMEmodule chassis, VMEmodule accessory cards, I/O Channel accessory cards and VERSAdos Real Time Multitasking Disk Operating System provides a complete design environment that frees the system designer to develop the unique software/firmware required for the application. Figure 1 diagrams the major functional components of the MVME110-1 Monoboard Microcomputer.

LOCAL MEMORY

Sockets are provided for use of 28-pin 8K, 16K, and 32K byte ROM/PROM/EPROM and RAM devices. A jumper header is provided so that compatible 24-pin 2K, 4K, and 8K byte devices may optionally be used. Another header facilitates jumper selection of operation with memory devices of various access times.

Local on-board RAM is not accessible from the VMEbus interface and, under program control, local RAM can be write protected against a program executing in the MC68000 user state. Any number of 2K byte blocks in the ranges 000000

through 03FFFF and F00000 through F3FFFF can be configured for software-controlled write protection by reprogramming the VME110-1 map decoder PROM.

LOCAL BUS

The VME110 employs a Motorola MC68000 16-bit microprocessor operating at 8 MHz. To allow full speed processing while another VMEbus master is operating, a local bus is used to interconnect the MPU with ROM, RAM, the serial I/O port, the I/O channel, the Programmer Timer Module (PTM) and the VME110 status and control registers. A header is also provided to allow jumper enabling of a local bus time-out counter that generates a bus error signal for any cycle not completed within 200 μs.

SERIAL PORT

An RS-232C serial port is provided to facilitate downloading of programs and use of a terminal. This front-accessible port is implemented as Data Circuit-Terminating Equipment (DCE) and may be connected to a Data Terminal Equipment (DTE) device for use in debugging.

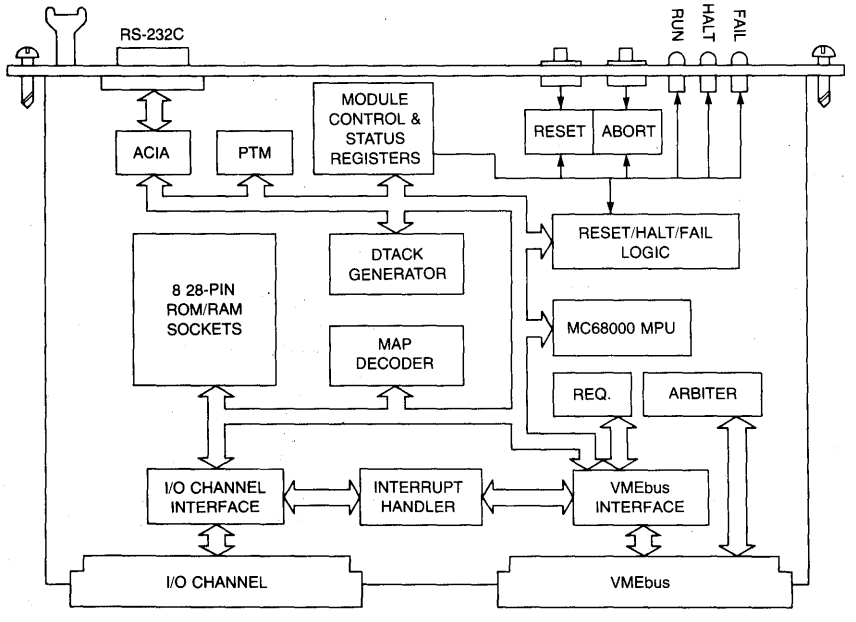
PROGRAMMABLE TIMER/COUNTER

For implementing various interrupts and interval timers, an MC6840 programmable timer module provides three independent cascadable 16-bit counters. Input to one counter may be jumper connected to the MPU E clock or baud rate clock. Two counters and an enable/disable bit in the Module Control Register (MCR) may be used to implement a watchdog timer for resetting the VME110 or external system if the processor fails to service an interrupt within a specified interval. This feature can be used to protect hardware and software and facilitate recovery from fault conditions.

STATUS MONITORING AND CONTROL

Push buttons are provided for the functions of RESET and ABORT. These buttons may be disabled by the user, after system development, if the board is to be used in a critical application. LED status lights are provided to indicate RUN, HALT, and FAIL status of the board. FAIL indication may be due to system failure, or to an error detected in a user-supplied power-on self-test routine.

FIGURE 1 — Block Diagram
VMEmodule Monoboard Microcomputer



VMEbus INTERFACE

VMEbus is characterized by the asynchronous bidirectional operation required for complex, high-performance systems. The VMEbus interface provided on the VME110 module supports direct memory access (DMA), multiprocessor operation and the full 16 megabyte address range of the MC68000 MPU. Designs requiring an expanded microcomputer function can utilize the VMEbus interface to add other resources such as RAM and intelligent I/O controllers. Pins for all VMEbus address, data and control lines are provided in the triple row, 96-pin VMEbus connector P1.

DATA TRANSFER BUS ARBITRATION

Each VME110 module contains the requester logic required to request and acknowledge mastership of the data transfer bus (DTB) on any one of four priority levels (bus request lines). In a multi-master system, one VME110 module is configured as system controller and performs single level arbitration for all DTB masters on bus request level three only. On any level when a request is received and the bus is not busy, the arbiter issues a bus grant via the bus grant daisy chain and waits for the grantee to activate bus busy. The cycle is then completed by the arbiter deactivating bus grant in.

DTB REQUEST/RELEASE

Programmed access by a VME110 module to an off-board VMEbus resource is obtained by a request being placed on bus request line corresponding to the level selected by strap on the requesting module. When the DTB is no longer busy, the bus arbiter grants mastership to the requester via the bus grant daisy chain. After using the bus, a VME110 requester releases the bus according to the mode determined by the bus release bits in its own module control register.

INTERRUPT HANDLER

The VME110 can respond to seven levels of prioritized interrupts. This capability is used to accommodate two distinct groups of seven interrupts each: interrupts incoming over the VMEbus interrupt lines IRQ1* through IRQ7* and local interrupt requests. For the latter group, the MPU's auto vector feature is used to obtain service for interrupts from local sources such as the Asynchronous Communications Interface Adapter (ACIA) and the Programmable Timer Module (PTM), for the SYSFAIL signal and for the I/O Channel interface. Four interrupt levels are assigned to the I/O Channel.

Vectors for IRQn* signals are read from the DTB during an interrupt acknowledge. In this cycle after gaining bus mastership, the VME110 places on the lower three address lines the interrupt level to be acknowledged and activates IACK* and the appropriate strobe signals. The interrupting device then places the interrupt vector on the lower data byte lines and acknowledges the data transfer. The vector is then used as a pointer to the MPU exception vector table.

I/O CHANNEL

The Motorola I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, and an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For those modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable up to 12 feet in length.

Available I/O Channel modules include:

- MVME400 Dual Channel RS-232C Serial Port (Synchronous/Asynchronous)
- MVME410 Dual Channel 16-bit Parallel Port (Centronics compatible)
- MVME420 SASI Peripheral Adapter
- MVME600,605 Analog Input and Output
- MVME610, 615, 616 Opto Isolated 120V/240V Input and Output
- MVME620, 625 Opto Isolated 30 Vdc Input and Output
- M68RWIN1 Winchester Disk Controller Module
- M68RAD1 Remote Intelligent Analog-to-Digital Conversion Module
- M68RIO1 Remote Input/Output Module

MEMORY MAPPED I/O

The memory map for the MVME110-1 Monoboard Microcomputer is shown in Figure 2.

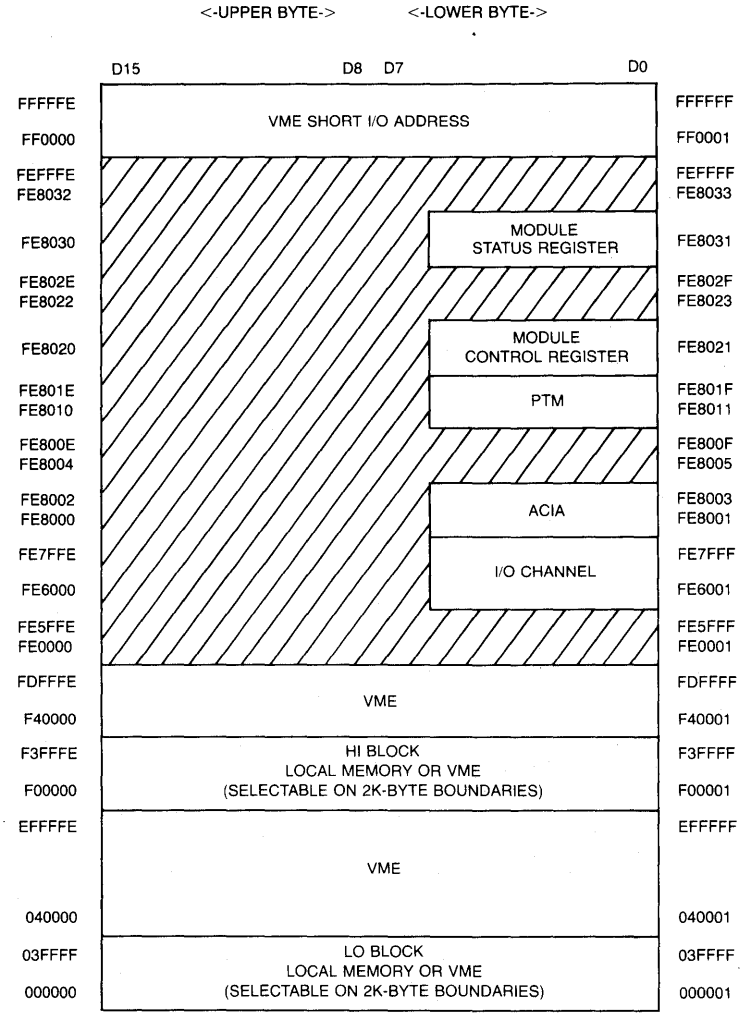
SOFTWARE/FIRMWARE SUPPORT

Motorola provides software packages to support the VMEmodule Monoboard Microcomputer, within the categories of Real-Time Executives and Operating Systems, and Debuggers/Loaders. The principal features of these software products are as follows:

RMS68K — Real-Time Multitasking System Software

- Memory-Resident (ROMable)
- Physical (Channel) I/O
- Multitask Dynamic Scheduling
- Software and Hardware Interrupt Processing
- High-Speed Interrupt Response
- Intertask Communication and Task Synchronization

FIGURE 2 — MVME110-1 Memory Map



NOTES:
Shaded portions indicate redundant I/O addresses and should not be accessed. The initial addresses for the SSP and the PC are obtained from the first four word locations of the ROM installed in socket pair 1.

- Dynamic Allocation and Management of RAM
- User Trap Handling
- Exception Processing
- Time Delay, Periodic Task Activation, Time-Of-Day
- Easy Addition of User-Written Device Drivers
- Upward Compatible to Real-Time Disk Operating System
- Compatible with EXORmacs System Software
- Customization via SYSGEN

VERSAdos — Real-Time Disk Operating System

- Provides all Real-Time Multitasking Software Features of RMS68K
- Device Independent I/O and Logical I/O
- Wait and Proceed Mode I/O
- Standard Device Drivers
- Multi-Level File Directories
- Shared File Access
- Dynamic or Contiguous File Space Allocation
- Random, Sequential, and Indexed Sequential File Access

VMEbug — Debug/Monitor/Loader Firmware

- Initialization
- Display/Change Memory
- Display/Change Registers
- Set and Clear Breakpoints
- Block Initialize
- Block Move
- Search for a (Masked) Value
- TRACE with optional instruction count
- Downline/Upline Load
- Single-line Assembler/Disassembler
- Boot routine for M68RWIN1 (Winchester Controller) and MVME420 (SASI Peripheral Adapter)

HARDWARE/SOFTWARE DEVELOPMENT SUPPORT

For development of VMEbus applications utilizing any Motorola 16-bit or 8-bit MPU/CPU chips, the recommended development system is the VME/10 Microcomputer System. VME/10 is a 5-1/4" floppy disk and 5-1/4" Winchester-based system designed around the MC68010 16/32-bit Microprocessor Unit and the MC68451 Memory Management Unit. VME/10 provides an I/O Channel interface, offers I/O Channel and VMEbus card cages, and can host all Motorola family hardware development tools. These include: the HDS-400 for M68000 Family 16/32-bit emulation, the HDS-200 for M6800 family emulation, and the Bus State Analyzer for logic analysis.

The VME/10 incorporates the real-time, multitasking operating system VERSAdos, an MC68000 family macro assembler, a symbolic debugger and a diagnostic/bootstrap monitor and offers advanced software development tools. These include a Pascal compiler with a fast floating point option, a Fortran compiler and CRT and linkage editors. The UNIDOS Operating System package is also available for VME/10.

For multiuser development of VMEbus applications based on the MC68000, the EXORmacs Development System is recommended. Since EXORmacs also supports VERSAdos, it can utilize the same hardware and software development tools as VME/10. Application programs designed to operate under VERSAdos may easily be developed and checked out in the EXORmacs environment, then downloaded into the VMEbus target system for final debug.

SYSTEM EXPANSION

VMEmodules designed for use with MVME110-1 include:

- 64K and 256K byte DRAM Modules
- 16-socket RAM/ROM/EPROM Module
- IEEE-488 Listener/Talker/Controller Modules

I/O Channel modules in single high EUROCARD format for use with MVME110-1 module include:

- Dual Port, Synchronous/Asynchronous Serial Module
- Dual Port Parallel (Centronics compatible) Module
- SASI Peripheral Adapter (interface for 8" or 5-1/4" disks)
- Buffered 9-Track Magnetic Tape Adapter
- Multichannel, 12-bit A/D Module
- Multichannel, 12-bit D/A Module
- Opto Isolated ac I/O Module
- Opto Isolated dc I/O Module

I/O Channel modules in non-EUROCARD format include:

- Remote, Intelligent A/D Module
- Remote, 16-socket, Solid State Relay (Opto-22 type) Module
- Winchester Disk Controller Module — Hard and Floppy 8" and 5¼" Disks

Packaging and Accessories

- 5-, 9-, and 20-Slot Backplanes
- 20-Slot Card Cage
- 40 Amps at 5 Vdc Power Supply with Optional Power Monitor
- Single (30) and Double (60) Wirewrap and Extender Cards
- Adapter to front mount I/O Channel to 50-pin ribbon connector

SPECIFICATIONS

General specifications for the VME110 are as listed:

TABLE 1 — VME110 Specifications

Characteristic	Specification
Power requirements (with all eight sockets unpopulated)	+5 Vdc (±5%), 2.1 A (typical), 2.4 A (max.) +12 Vdc (±5%), 25 mA (typical), 50 mA (max.) −12 Vdc (±5%), 25 mA (typical), 50 mA (max.) (see NOTE)
Power requirements (with all eight sockets populated)	+5 Vdc (±5%), 2.6 A (typical), 3.0 A (max.) +12 Vdc (±5%), 25 mA (typical), 50 mA (max.) −12 Vdc (±5%), 25 mA (typical), 50 mA (max.) (see NOTE)
Temperature Operating Storage	 0° – 70°C −55° to +85°C
Relative Humidity	0% to 90% (non-condensing)
Physical Characteristics Height Depth Thickness	Double-high VME board 9.2 in. (234 mm) 6.3 in. (160 mm) .662 in. (16.77 mm)

NOTE: The currents at +12 Vdc and −12 Vdc are specified for the MVME110 module with the serial port connectors open. The actual required values depend on the load of the RS-232C port. All serial port outputs are current-limited to sink or source 12 mA (max.) each.

TABLE 2 — RS-232C Serial Port Connector J15 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Signal Name and Description
1, 4, 9–19, 21–25	(Reserved)	Not connected.
2	TXD	TRANSMIT DATA — Transmit data from terminal. This signal is connected to the ACIA receive data input.
3	RXD	RECEIVE DATA — Receive data to terminal. This signal is connected to the ACIA transmit data output.
5	CTS	CLEAR TO SEND — Indicates terminal may send data. This signal is controlled by the ACIA $\overline{RTS}$ output.
6	DSR	DATA SET READY — Indicates to terminal that port is ready. When power is applied, this signal is true.
7	GND	SIGNAL GROUND
8	DCD	DTA CARRIER DETECT — Indicates to terminal that data carrier is present. When power is applied, this signal is true.
20	DTR	DATA TERMINAL READY — Indicates to port that terminal is ready. This signal is connected to the ACIA $\overline{CTS}$ input and must be true for the ACIA to transmit data.

Ordering Information

Part Number	Description
MVME110-1	VMEmodule Monoboard Microcomputer. This module contains the MC68000L8 MPU, sockets for up to 256K bytes of RAM/ROM/EPROM, serial port, a triple programmable timer/counter, VMEbus interface, I/O Channel interface, and System Controller features. Operates at 8 MHz clock. Includes User's Manual
MVME110/D	VMEmodule Monoboard Microcomputer User's Manual

Related Documentation

MC68000UM	MC68000 16-Bit Microprocessor User's Manual
MC6840UM	MC6840 Programmable Timer Fundamentals and Applications
MVMEBUG/D	VMEbug Debugging Packages User's Manual
HB212/D	VMEbus Specification Manual
M68RIOCS	Input/Output Channel Specification Manual

Accessory Modules Include:

Part Number	Description
MVME200	64K Dynamic RAM VMEmodule
MVME201	256K Dynamic RAM VMEmodule
MVME210	Static RAM/ROM 1K, 2K, 4K, 8K, 16K x 8 VMEmodule
MVME211	Static RAM/PROM 1K, 2K, 4K, 8K x 8 + 5 V Standby for CMOS RAM VMEmodule
MVME300	GPIB Listener, Talker, Controller VMEmodule
MVME400	Dual Channel RS-232C Serial Port I/O Channel Module
MVME410	Dual Channel 16-Bit Parallel Port I/O Channel Module
MVME420	SASI Peripheral Adapter I/O Channel Module
MVME600	12-Bit, 16-Channel Single Ended, 8-Channel Differential A/D I/O Channel Module
MVME601	16-Channel expansion board for MVME600
MVME605	12-Bit, 4-Channel D/A I/O Channel Module
MVME610	Opto Isolated, 8-Channel ac Input I/O Channel Module
MVME615	Opto Isolated, 8-Channel, Zero Crossing ac Output I/O Channel Module
MVME616	Opto Isolated, 8-Channel, Non-Zero Crossing ac Output I/O Channel Module
MVME620	Opto Isolated, 8-Channel, dc Input I/O Channel Module
MVME625	Opto Isolated, 8-Channel dc Output I/O Channel Module
M68RAD1	Remote, Intelligent A/D Conversion Module
M68RI01-1	Remote Input/Output Module
M68RWIN1	Winchester Disk Controller Module

NOTE: All VMEmodule 100 Series and 200 Series modular products are of double Eurocard form factor, VMEbus compatible.

Applicable Software/Firmware

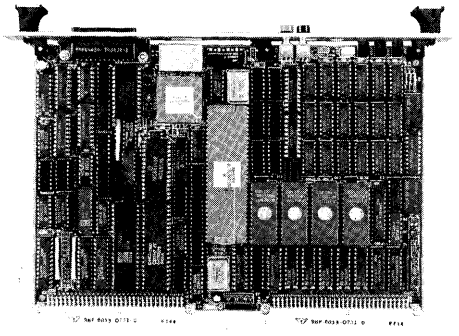
MVMEbug	VMEbus Debugging Packages: 32K x 8 EPROM, Diskette or Cartridge
M68KVDOS 4.5	VERSAdos Operating System

MVME117-3
MVME117-3FP
MVME117-4

Advance Information

VMEmodule™ Monoboard Microcomputer

- MC68010 Microprocessor Running at 10 MHz
- Provision for MC68881 Floating Point Peripheral (customer-supplied option)
- Zero Wait-state Memory (512Kb); One Wait Cycle Memory with Parity Enabled
- One Multi-protocol RS-232C Serial Port (P2 interface) and one Asynchronous Serial Debug Port (to terminal interface only)
- Two Independent 8-bit Parallel Ports with I/O Handshake Lines
- SCSI Bus Interface with Pseudo-DMA Channel
- Time-of-day Clock/Calendar with Battery Backup
- Battery-backed Socket Accepts 2K x 8 or 8K x 8 CMOS RAM
- MC68B40 Programmable Timer Module Provides Three Independent Timer/Counters
- Interrupt Handler with Programmable Vectors for On-board Interrupts (PROM)
- VMEbus System Controller
  - — Provides system clock (SYSCLK)
  - — Single-level arbiter (Level 3 only)
  - — VMEbus time-out
  - — System reset generator
- VMEbus Interface (D:8/D:16, A:24 master)
- Stand-alone Operation Requires a Terminal, Power Supply, and Optionally a SCSI Disk
- Four EPROM Sockets Accept 8K x 8, 16K x 8, 32K x 8, or 64K x 8 JEDEC Standard Parts. Devices with 250 ns Access Times (or less) run with One Wait Cycle. Devices with 450 ns Access Times (greater than 250 ns and less than 450 ns) Run with Three Wait Cycles
- VMEbus Requester
- Status LEDs for HALT, RUN, and FAIL
- RESET and ABORT Switches
- Remote RESET (P2 interface)
- 8-Bit Software-Readable Front Panel Switch
- 8-Bits of Software-readable Board Configuration (PROM)
- MVME708 Interconnect Transition Module Available to Direct P2 Interface to Remote Reset in Addition to Standard SCSI, RS-232C, and Printer Connectors



- Software Support
  - — 117bug Debug Monitor with SCSI support available in source or object code
  - — RMS68K real-time multitasking kernel available in source or object code
  - — VERSAdos real-time operating system available in source or object code
  - — ROM-based Basic Interpreter
  - — An application note describing the software macros required to implement the MC68881 floating point co-processor as a peripheral in an MC68000/008/010/012 system
  - — High-level language support (VERSAdos). Pascal and FORTRAN available in source or object code

The MVME117 VMEbus monoboard microcomputer is a high-functionality, high-performance VMEbus-compatible microprocessor module that offers most functions needed for a complete computer system on a single Eurocard. This microprocessor module is versatile enough to run stand-alone as a packaged single board computer or as a part of a modular, user-configurable VMEbus computer system. There are five versions of the MVME117 designed to meet the processor needs of a wide array of users. Refer to the ordering information for descriptions of the five versions. Figure 1 is a functional block diagram of the MVME117 module.

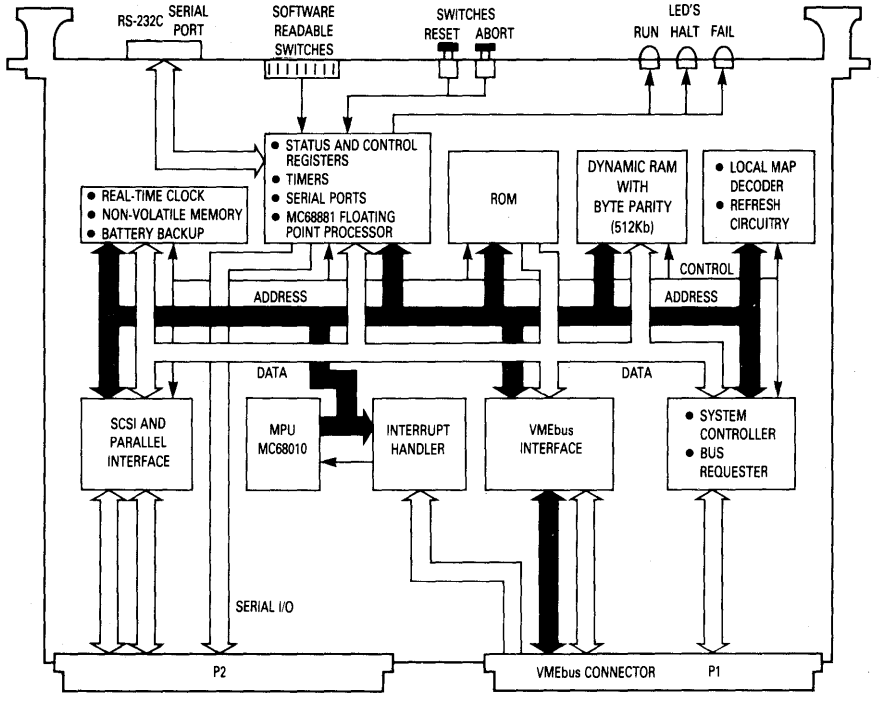


Figure 1. MVME117 Functional Block Diagram

MC68010 MPU

Processing on the MVME117 is performed by the MC68010 16-bit microprocessor operating at a fixed 10 MHz. This microprocessor has a proven architecture optimized for efficient support of high-level languages and secure operating systems. The MC68010 offers the following popular features:

- 32-Bit Data, Address, and Stack Registers
- 14 Addressing Modes
- 16Mb Direct Addressing Range
- 57 Instruction Types
- Operation on Five Data Types, Including Bit, Byte, Word, Longword, and BCD
- Interlock Instruction for Multiprocessor System Operation
- Fast Loop Mode Operation
- 256 Multilevel Vectored Interrupts

MC68881 FLOATING POINT PERIPHERAL

The MVME117 module is equipped with a socket to accept the MC68881 floating point peripheral, a customer-supplied option. The MC68881 is a high-performance, single chip HCMOS VLSI floating-point co-processor. The MC68881 is primarily intended to operate as a closely coupled co-processor with the full 32-bit MC68020 MPU, but it can also operate as a memory mapped peripheral as it is on the MVME117. When installed, the MC68881 functions as a 16-bit peripheral to the MC68010 that performs floating point arithmetic functions to full IEEE compatibility. The MVME117 can supply frequencies of 10 MHz, 16 MHz, or 20 MHz to the MC68881 socket, thereby allowing the maximum performance possible with the available MC68881.

ONBOARD MEMORY

The MVME117 is equipped with 512Kb of onboard local RAM with byte parity. When parity is disabled, programs operating from onboard RAM run without wait cycles using a 10 MHz MC68010 microprocessor. When parity is enabled, these programs encounter one wait cycle.

Four 28-pin sockets are available for EPROMs. For example, these sockets may contain the MVME117bug debug monitor with the intelligence for the SCSI bus interface, self-test and bootload. The user may install up to 256Kb of ROM in these sockets.

One socket is provided for battery-backed CMOS RAM. This 28-pin socket accepts a 24-pin 2K x 8 CMOS RAM or a 28-pin 8K x 8 CMOS RAM. System constants and other parameters that can change over time but must not be lost in a power-down or power-fail condition may be stored and retrieved in this battery-backed memory.

SERIAL I/O

The Zilog Z8530A Serial Communications Controller provides two serial ports for the MVME117. A RS-232C compatible serial port with a DCE (to terminal) asynchronous interface is accessible on the front panel of the MVME117. The other RS-232C compatible multiprotocol serial port is accessible at the P2 connector of the MVME117, or optionally, on the front panel of the MVME708 transition module. The multiprotocol port has the following hardware features:

- Software Programmable Baud Rates
- Digital Phase Locked Loop for Clock Recovery
- Multiprotocol Operation under Software Control; Programmable for NRZ, NRZI, or FM Data Encoding
- Asynchronous Mode with 5- to 8-Bits Per Character and 1, 1-1/2, or 2 Stop Bits Per Character; Programmable Clock Factor; Break Detection and Generation, Parity, Overrun, and Framing Error Detection
- Synchronous Mode with Internal Character Synchronization on One or Two Characters and CRC Generation and Checking with CRC-16 or CRC-CCITT Reset to either 1's or 0's
- SDLC/HDLC Mode
- Local Loopback and Auto Echo Modes
- DCE or DTE Interface (jumper-selected on MVME708 module)

PARALLEL I/O PORTS

Two independent 8-bit parallel ports are provided on the MVME117. Each port has one output handshake line and one input handshake line. The input handshake signal may be programmed to interrupt on a high level or a low level. Each 8-bit port may be programmed as buffered or latched input or as latched output. The two ports may be combined and used as a Centronics printer port. These ports may be interfaced via the MVME117 P2 connector or, optionally, they may be interfaced through the MVME708 transition module.

SCSI BUS INTERFACE

The MVME117 provides a Small Computer Systems Interface (SCSI) Bus Interface on the P2 connector. Alternatively, the user may employ a standard connector SCSI Bus Interface on the optional MVME708 transition module. Implemented with the NCR5380, this interface supports the SCSI bus as defined by the ANSI X3T9.2 committee. With the NCR5380, the MVME117 can operate in either the initiator or target roles and therefore can be used as a host or as a control unit. The MC68010 interface to the NCR5380 incorporates a pseudo-DMA channel that (with the loop mode of the MC68010) allows a burst data transfer rate of approximately 500Kb/sec between the on-board RAM and the SCSI bus in either direction.

TIME-OF-DAY CLOCK/CALENDAR

A battery-backed real-time clock is provided on the MVME117 to provide time-keeping and calendar functions. This clock provides time-keeping from tenths of seconds to tens of years. This clock also may interrupt the processor periodically every 0.1 second, 0.5 second, 1 second, 5 seconds, 10 seconds, 30 seconds, or 60 seconds. The device used is a CMOS MM58274.

PROGRAMMABLE TIMER MODULE

The MC68B40 provides three independent 16-bit timer counters. One of these is used as a watchdog and the other as a tick-timer. The third timer is available for application-specific users.

BATTERY BACKUP SYSTEM

A battery is provided along with the necessary circuitry to backup the CMOS RAM and the real-time clock whenever power is turned off. The battery is capable of 750mAH with an estimated data retention time of approximately five years.

INTERRUPT HANDLER

The MVME117 interrupt handler manages interrupts from the serial ports, timer, SCSI, parallel ports, abort switch, real-time clock, SYSFAIL, ACFAIL, and the seven interrupts from the VMEbus. Interrupt vectors for the local interrupts may be assigned in the vector PROM. The serial ports have software-programmable vectors.

VMEbus SYSTEM CONTROLLER

The MVME117 may work as a VMEbus system controller for other VMEbus masters if it is physically located in Slot 1 of a VMEbus chassis and if the VMEbus system controller function is enabled with a jumper.

VMEbus REQUESTER

The MVME117 VMEbus requester operates in either the Release on Request (ROR) or the Release When Done (RWD) mode with early release of Bus Busy (BBSY). The mode is software-selected through the module control register. In addition, the VMEbus requester can operate on any of the four (0 to 3) VMEbus levels.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Configuration	DTB Master: A24, D16 Requester: Any one of R(0), R(1), R(2), R(3), Static; ROR or RWD Arbiter: R(3) only Interrupt Handler: Any of IH(0 . . . 7)
Form Factor	Double High Eurocard
Physical Dimensions Card size Height Width Front Panel Height Width	 9.2 in. (23.37 cm) 6.3 in. (16.00 cm) 10.2 in. (26.10 cm) 0.83 in. (2.10 cm)
Power Requirements	+5 Vdc 3.9 A +12 Vdc 50 mA −12 Vdc 50 mA
Operating Temperature	0°C to 55°C inlet air temperature — forced air cooling required
Storage Temperature	−40°C to 85°C
Humidity	5% to 90% relative humidity (non-condensing)

ORDERING INFORMATION

Part	Description
MVME117-3	VMEbus Single Board Computer with 10 MHz MC68010 MPU, 512Kb zero-wait-state DRAM, SCSI Bus Interface, Battery-backed time-of-day clock, four EPROM sockets, MC68881 floating point processor socket, two serial ports, and one parallel port. Includes user's manual.
MVME117-3FP	Same as MVME117-3, but with MC68881 processor onboard.
MVME117-4	Same as MVME117-3, but without SCSI.
MVME117bug	MVME117 Debug Monitor with self-test, debug utilities, and SCSI bus support. Includes user's manual.
MVME708-1	MVME708-1 Interconnect Transition Module for the MVME117 family of microprocessor modules. Includes user's manual.
MVME117/D	MVME117 Monoboard Microcomputer user's manual.
MVME117BUG/D	MVME117 Debug Monitor user's manual.
MVME708/D	MVME708 Interconnect Transition Module user's manual.

RELATED DOCUMENTATION

MVMESYSAM/D	VMEsystem Architecture Guide
MC68010/D	Microprocessor Data Manual
MC68881/D	Floating Point Peripheral Data Manual

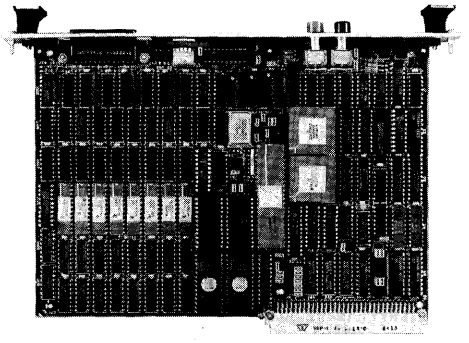
MVME121
MVME123

VMEmodule™ VMEbus Microprocessor Modules

- MC68010 Microprocessor Running at 10 or 12.5 MHz
  - — Hardware virtual memory support, with instruction continuation after a fault
- 512Kb Dual-Ported Dynamic RAM with Byte Parity
- 4Kb of Zero Wait-state Instruction Cache, Providing Significant Execution Speed-up (Optional)
- MC68451 Memory Management Unit (MMU) (MVME121)
- Two 28-pin Sockets for User-supplied ROM/EPROM Devices
- A24, D16 VMEbus Interface
- Status/Control Registers
- Interrupt Handler
  - — Any one of the seven interrupt request levels
- Four Onboard Timers
  - — System 'TICK', Watchdog, Baud, and Delay
- RS-232-C Serial Port, for Debug Terminal
- Local Reset Switch
- Status Display LED's: Run, Halt, Fail
- Optimized for Multiprocessor Applications
  - — No common system controller functions onboard (removed to system controller)
  - — Large onboard RAM and private ROM; can work at full speed in parallel with other processors without causing unnecessary loading of the bus
  - — Dual port memory between the MC68010 MPU and the VMEbus allows efficient communication between co-operating processors without the time-consuming delays required to copy from global memory to onboard memory

Note: A module with system controller functions (arbitration), such as the MVME050 or the MVME025, is required for operation.

The MVME121 Family of VMEbus Microprocessor Modules are versatile, VMEbus-compatible microprocessor modules that offer the performance required for high-speed data processing, data management, and industrial control. Applications range from high-speed single processor systems through systems with complex multiprocessor architectures.



The two versions of the microprocessor module are the MVME121 and MVME123. The following options are available:

- 10 MHz or 12.5 MHz MC68010 Microprocessor
- MC68451 Memory Management Unit (MMU)

Table 1 lists the specific features of each module.

The MVME121 derives its processing power from the superior resources of the MC68010 16-bit Virtual Memory Microprocessor which supplies hardware support for virtual memory environments. The combination of the MC68010, the MC68451 Memory Management Unit, and a 4Kb no wait-state instruction cache create a MVME121 design highly suited for multiprocessor systems. As a result, such mainframe computer features as virtual machine, virtual I/O, and virtual memory (a key requirement for the UNIX operating system) are able to be implemented on an internationally standard microprocessor bus.

Figure 1 is a functional block diagram of the MVME121.

Table 1. MVME121 Family Options

	MVME121	MVME123
MC68010 MPU	10 MHz	12.5 MHz
MC68451 MMU	YES	NO
DYNAMIC RAM	512Kb	512Kb
CACHE MEMORY	YES	YES

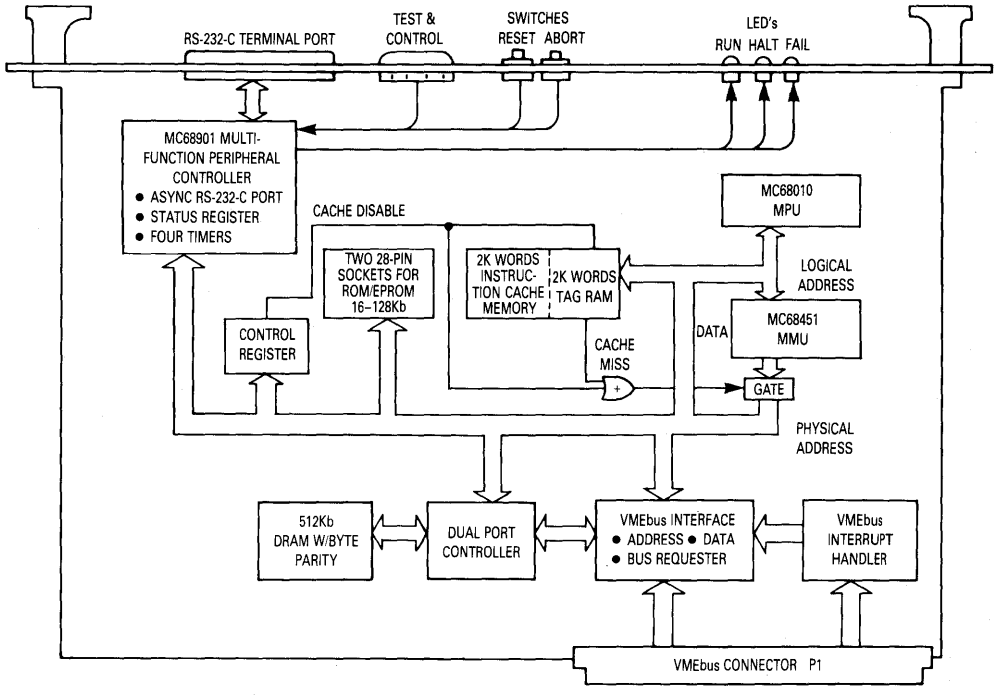


Figure 1. MVME121 Functional Block Diagram

MC68010 MPU

At 10 MHz, the MC68010 executes 25% faster than the 8 MHz MC68000 and has all of its popular features:

- 32-Bit Data, Address, and Stack Registers
- 14 Addressing Modes
- 16Mb Direct Addressing Range
- 57 Instruction Types
- Operation on Five Data Types, Including Bit, Byte, Word, Longword, and BCD
- Interlock instruction for Multiprocessor System Operation
- 256 Multi-level Vectored Interrupts
- Proven Architecture, Optimized for Efficient Support of High-level Languages and Secure Operating Systems

In addition, the MC68010 provides hardware support for the implementation of virtual memory through instruction continuation on a page fault. Also included is a program control instruction that speeds loop mode execution by suppressing operation code prefetches until an exit condition is met.

MC68451 MEMORY MANAGEMENT UNIT (MMU)

The MC68451 MMU maps the MPU's 16Mb logical memory space into the system physical memory, and provides for the isolation and protection of the supervisor and user segments of memory.

The MC68010 MPU and MC68451 MMU are designed to work together to simplify implementation of mainframe-style virtual memory, a requirement for the UNIX multiuser operating system.

MC68901 MULTI-FUNCTION PERIPHERAL (MFP)

Several MVME121 functions are implemented in the MC68901 peripheral chip.

- Four timers: System 'TICK' Function (interrupt at programmable intervals for multiprocessing task switching, etc.), Watchdog, Baud, Delay
- Asynchronous RS-232-C Functions for Debug Port
- Status Register

CACHE

Zero wait-state instruction fetch is a mainframe hardware feature on a VME format processor!

The cache is a small, very fast memory situated between the MPU and its "regular" memory. As each instruction is fetched from anywhere in "regular" memory (with attendant delays for memory and/or bus access), it is stored in the cache on its way to the MPU. If this same instruction is then needed again, as when executing tight loops for example, the next fetch will be able to get the

instruction from cache (called a cache "hit") with minimal delay (zero wait-state access).

The cache memory is single-set associative with a block size of two bytes and a total capacity of 4096 bytes. The cache is used for instruction and program counter relative data storage, and holds up to 2048 instructions.

Performance improvement calculations give the following: Executing a fast floating point multiply, where the instruction to data mix on memory fetches is 7.75:1, there would be an 86% hit rate — 86% of the memory accesses would be from the cache, with execution approximately 40% faster than the same program without cache.

ONBOARD MEMORY

Two 28-pin sockets are available for private memory. For example, they may contain selftest or bootload. They may be configured for several standard EPROM/ROM devices from 4K x 8 to 64K x 8.

The onboard memory includes 512Kb of 150 ns access shared dynamic RAMs that are accessible both as local memory and from the VMEbus. VMEbus access is used for fast DMA loading from disk or for communication with other VMEbus processors. Data is then accessible for program execution and data manipulation with no bus load or bus delay.

BUS REQUESTER

The bus requester requests VMEbus mastership on any of four request levels (user-configurable via jumpers), and fully supports the bus grant daisy chain. The bus requester operates in the Release on Request (ROR) mode and requests bus mastership only if the MPU attempts a VMEbus transfer when the module does not already possess VMEbus mastership.

INTERRUPT HANDLER

The interrupt handler enables the MC68010 microprocessor to sense and respond to all onboard interrupts and all seven VMEbus interrupts. The VMEbus interrupts are independently routed through onboard headers to provide a means whereby the onboard MPU can selectively ignore individual VMEbus interrupts.

PHYSICAL ADDRESS MAP (as seen from the MC68010)

000000 thru 000007	Reset vector for first four memory cycles, RAM thereafter. Vector may be from onboard ROM or VMEbus (jumper option).
000008 thru 01FFFF	Onboard RAM. (Dual port)
020000 thru EFFFFF	VMEbus
F00000 thru F0FFFF	Onboard ROM/EPROM
F20000 thru F2002F	MFP (MC68901)
F40000	Module Control Register
F60000 thru F6003F	MMU Registers
F80000 thru F80006	Cache Control Signals
FA0000 thru FEFFFF	VMEbus
FF0000 thru FFFFFF	VMEbus (global short I/O page)

ADDRESS MAP OF MVME121/123 (as seen from VMEbus)

000000 thru 01FFFF	Dual port RAM (with standard decoder).

MEMORY TIMING (wait cycles)

MVME121	READ		WRITE	
	wait-cycles	access-time	wait-cycles	access-time
Cache	0		n/a	n/a
Onboard RAM*	2	150 ns	2	150 ns
Onboard EPROM/ROM*	5	250 ns	n/a	n/a
VMEbus**	3	70 ns	—	—
	4	170 ns	4	113 ns
	5	270 ns	5	213 ns
	6	370 ns	6	313 ns
	7	470 ns	7	413 ns

MEMORY TIMING (wait cycles)

MVME123	READ		WRITE	
	wait-cycles	access-time	wait-cycles	access-time
Cache	0		n/a	n/a
Onboard RAM*	2	150 ns	2	150 ns
Onboard EPROM/ROM*	5	250 ns	n/a	n/a
VMEbus**	4	150 ns	4	93 ns
	5	230 ns	5	173 ns
	6	310 ns	6	253 ns
	7	390 ns	7	333 ns

*Access times are device access times
**VMEbus access times are from Data Strobe* to Dtack*

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Configuration	DTB Master: A24, D16 DTB Slave: A24, D16 — No arbiter on-board (use MVME050 or other System Controller) — Requester: Any one of R(0), R(1), R(2), R(3) Static, ROR — Interrupt Handler: Any of IH(0...7), Static
Form Factor Card dimensions Front panel dimensions	Double High Eurocard Height 9.2 in. (234 mm) x Width 6.3 in. (160 mm) Height 10.2 in. (261 mm) x Width 0.83 in. (21 mm)
Power Requirements MVME121 MVME123	At +5 Vdc: 4 A typ, 4.45 A max 4.3 A typ, 4.65 A max 17 mA typ, 20 mA max at +12 Vdc 12 mA typ, 14 mA max at −12 Vdc
Environmental Limits Operating Temperature Storage Temperature Humidity	 0°C to 55°C inlet air temperature (forced air cooling required) −40°C to 85°C 5% to 90% relative humidity (non-condensing)

ORDERING INFORMATION

Part Number	Description
MVME121	VMEbus Microprocessor Module featuring 10 MHz MC68010 MPU, MC68451 MMU, 512Kb RAM, and 4Kb no-wait-state instruction cache. Includes user's manual.
MVME123	VMEbus Microprocessor Module featuring 12.5 MHz MC68010 MPU, 512Kb RAM, and 4Kb no-wait-state instruction cache. Includes user's manual.
MVME120/D	MVME121, MVME123 Microprocessor Module User's Manual.
MVME120BUG	Debug monitor firmware for the MVME121 family of Microprocessor Modules. Includes user's manual.
MVME120BUG/D	MVME120 Debug Monitor User's Manual.

COMPATIBLE SYSTEM CONTROLLER VMEmodules

The MVME121 Processing Module requires another module with bus arbitration capabilities to act as system controller. VMEmodules compatible with the MVME121 include:

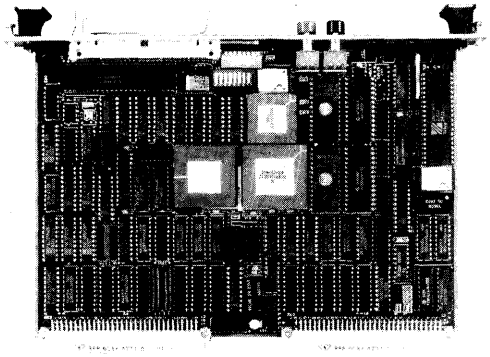
Part Number	Description
MVME025	VMEbus System Controller Module
MVME050	VMEbus System Controller Module

RELATED DOCUMENTATION

Part Number	Description
MVMESYSAM/D	VME System Architecture Guide
MC68010/D	Microprocessor Data Manual
MC68451/D	Memory Management Unit Data Manual
MC68901/D	Multi-function Peripheral Data Manual

MVME130
MVME131

VMEmodule 32-Bit Monoboard Microcomputer



The VMEmodule 32-bit Monoboard Microcomputer (MVME130) is designed to function in those applications requiring maximum performance while maintaining the versatility inherent in VMEmodule systems. Highest performance is attained when the MVME130 is used in conjunction with one or more MVME204 Dual Ported RAM Cards operating as a main memory. The MVME130 is the first VMEmodule product to offer the following:

- MC68020 Microprocessor with 32-bit Address and Data VMEbus Interface
- Provision for MC68881 Floating Point Coprocessor (customer-supplied option)
- Provision for Demand Paged Virtual Memory Management Board (MMB) Implemented with Gate Array Technology
- Two 28-pin JEDEC Sockets for up to 16Kb Onboard Static RAM
- VSB* Interface Provides High-speed Data Path to/from Memory
- VMEbus Interface Allows User Configuration of Microcomputer System to Fit the Application
- VMEbus Interrupter, Interrupt Handler, and Arbiter Onboard
- Programmable Timer Module
- Dual Multiprotocol Serial I/O Ports
- Accepts MVME130bug Debug Monitor Firmware (optionally available)
- Two ROM Sockets Configured for Industry Standard 28-Pin ROM/EPROM devices. Four ROM Sockets Available when no Sockets are Devoted to RAM
- Broadcast Interrupt Capability for Multi-processor Synchronizing Applications

CPU

The MVME130 uses an MC68020 Microprocessor operating at a fixed speed of 12.5 MHz. This CPU will eventually operate at 16.67 MHz on the MVME130. The MC68020 is the first product within the popular MC68000 family to offer external 32-bit address and data paths. With its higher clock rate, advanced architecture, enhanced addressing modes, and on-chip instruction cache, this product offers state-of-the-art performance while maintaining software compatibility with its widely accepted predecessors.

DEBUG MONITOR FIRMWARE

The MVME130bug Debug Monitor firmware package is optionally available for use with the MVME130 module. This firmware offers 32 debug, up/downline load, one line assembler/disassembler and disk bootstrap load commands.

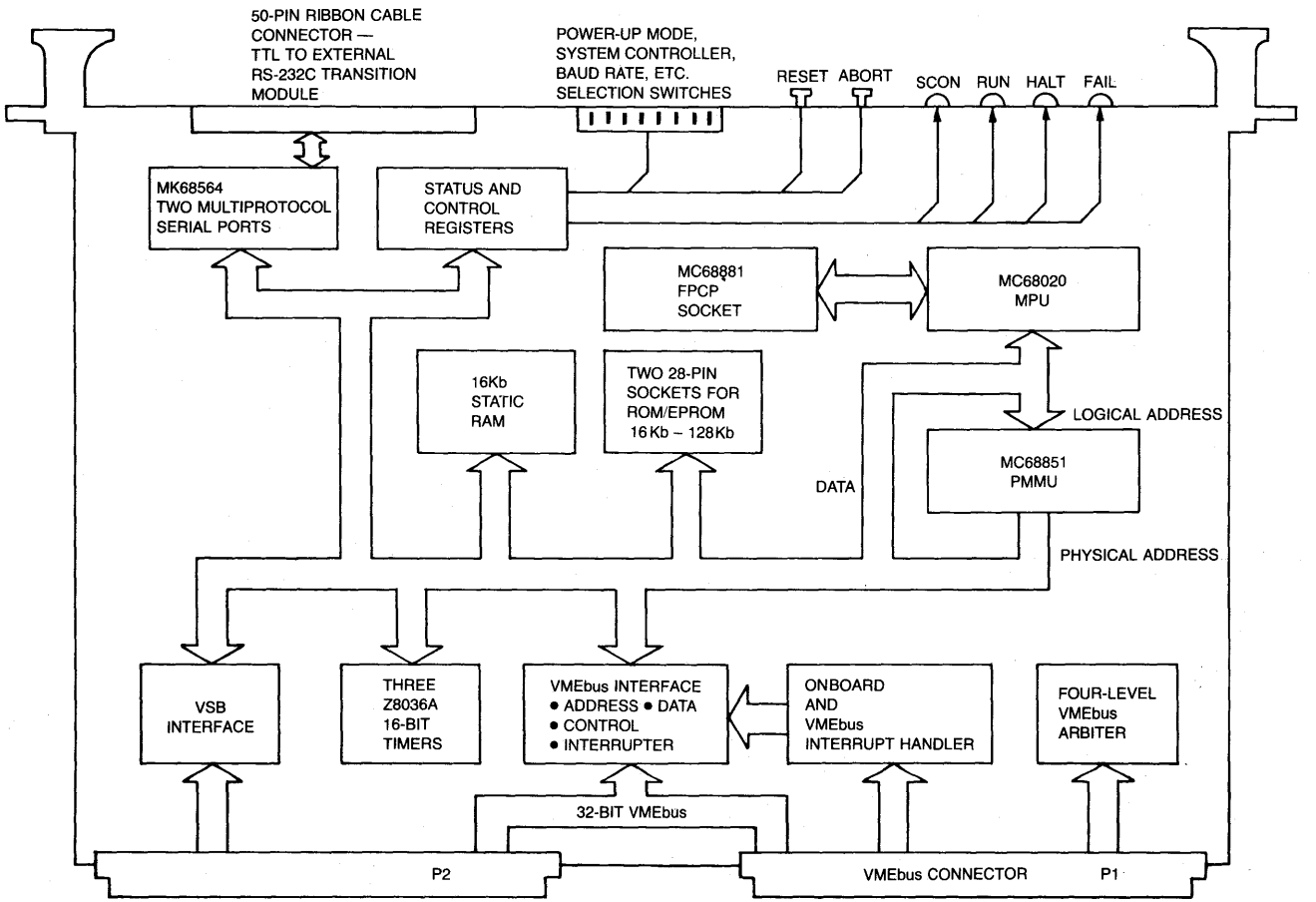
COPROCESSOR

The MVME130 module is equipped with a socket to accept the MC68881 Floating Point Coprocessor, a customer supplied option. When available and installed, the MC68881 Floating Point Coprocessor will improve computing speed of the MVME130 when the system is utilized in applications requiring arithmetic operations.

The MVME130 utilizes the MC68020 Coprocessor Interface to provide a logical extension to the MC68020 architecture and instruction set.

Figure 1 is a block diagram of the MVME130 module.

FIGURE 1 — MVME130 32-Bit Microcomputer Module



MEMORY MANAGEMENT

The MVME130 is designed to eventually utilize the MC68851 Demand Paged Virtual Memory Management Unit. However, one initial version of the MVME130 (MVME131) utilizes a Memory Management Board (MMB) which conforms to a subset of the MC68851. This subset is implemented via gate array technology together with other semiconductor devices mounted on a secondary PC board. The socket assigned for future use with the MC68851 is utilized to mount the secondary PC board in a mezzanine fashion on the introductory version of the MVME130. The MMB offers the following features:

- Logical address consisting of:
  - — 32-bit Address
  - — 3-bit Function Code
- Physical address output of 32 bits.
- Logical to physical address translation;
  - — Table walking algorithm
  - — Single logical bus master
  - — Single page size (1Kb)
- Used, modified, and write protect bits.
- Supports MC68020 and external data cache.
- 512 entry set associative cache.

The eventual use of the MC68851 will add multiple Protection/Privilege capabilities.

The principal operating function of the MMB on the MVME130 is to provide the hardware logical to physical address translation and access protection necessary for operating system support. The MMB logical to physical translation mapping function utilizes a table-walking algorithmic search through memory-resident translation tables. The MMB utilizes a cache architecture to eliminate table-walking for most CPU bus cycles.

The MMB utilizes the MC68020 clock circuitry, but does not use the coprocessor interface. This requires the internal registers to be in the CPU address space, whereas the MC68851 internal registers will be accessed via the coprocessor interface. This will require a change in driver routines when the MC68851 is used with the MVME130.

ONBOARD STATIC RAM

The MVME130 Static Memory is implemented with two 28-pin industry-standard JEDEC sockets. The MVME130 will provide 16Kb of single-ported memory for the MC68020 local processor. However, if the user does not require the onboard memory, the two static RAM sockets can be reconfigured to accept EPROMS up to 512K in size.

VSB INTERFACE

The VSB Interface occupies 64 P2 I/O pins and is designed to serve as the system's high-speed memory access channel. The signals utilize multiplexing of address and data in order to accommodate full 32-bit functionality, along with appropriate control signals, into the 64-pin allotment.

The VSB Interface on the MVME130 is limited to one master. The interface can be turned off via a control bit to force all accesses to go to the VMEbus, and speed up VMEbus accesses. The MVME130, with one or more MVME204's, creates a memory system with performance similar to that of a VMEmodule with onboard memory, but with virtually unlimited memory size constraints.

MEMORY MAP

The address map, (refer to Table 1) indicates the 32-bit address map for different devices addressed by the MVME130. The 24-bit address map is determined by eliminating the two most significant hex digits from the 32-bit addresses. A control bit on the control register allows the user to switch between 16- and 32-bit data paths, and between 32-/24-bit address modifier codes.

TABLE 1 — Address Map

Device	Physical Address	Comments
VMEbus/VSB	00000000-FFEFFFFF	32/16-bit port size; R/W.
ROM	FFF00000-FFF8FFFF	16-bit port size; read only.
Local Static RAM	FFF20000-FFF2FFFF	16-bit port size; R/W.
Spare	FFF40000-FFF6FFFF	32-bit port size; R/W provided for cache expansion ability.
Timer	FFFBxx00-FFFBxx2F	8-bit port size; R/W.
Status & Control	FFFBxx30-FFFBxx3F	32-bit port size; Status is read only; Control is R/W.
Serial I/O Port	FFFBxx40-FFFBxx5F	8-bit port size; most are R/W.
Reserved	FFFBxx60-FFFBxx6F	Unimplemented.
Spare	FFFBxx70-FFFEFFFF	Unimplemented; Access will cause LBTO (BERR source) cycle termination.
VMEbus Short I/O	FFFF0000-FFFFFFFF	16-bit; R/W; VMEbus Short I/O Address Modifier will be generated.

PROGRAMMABLE TIMER MODULE

The MVME130 Programmer Timer Module provides the following:

- Three Cascadable, Independent 16-bit Counters
- Interrupt Capability

SERIAL I/O

The MVME130 is equipped with two multiprotocol serial I/O ports with connection via a 50-pin flat ribbon cable connector at the top of the board. Signal levels at this connector correspond to TTL specifications. A transition board/cable/connector (MVME707) is made available to transform the TTL levels to RS-232C and provide a standard DB-25 interface. The ports have the following features:

- Programmable Baud Rates
- Full and Half Duplex Compatibility
- 5- to 8-bits Per Character Plus Parity
- Synchronous or Asynchronous Operation, Including
  - — Byte-oriented protocols (BISYNCH)
  - — Bit-oriented protocols (SDLC)
- Data Rate Capability to 800 Kilobits/Second

STATUS/CONTROL REGISTERS

The MVME130 Status and Control Registers provide local and system wide status information to the on-board MPU. By reading the Status and Control register bits, the MPU can determine VMEbus information, interrupt status, the source of a particular BERR, and which self-test to perform. The MVME130 Status/Control Register bits control the module's hardware by providing configuration and control information to the VMEbus Interrupter, the Interrupt Handler, the VMEbus Arbiter and VSB. For example, the Multi-Processor Interrupt Request (MPIRQ*) is an input from the VMEbus IRQ1* signal. MPIRQ* enables the user to interrupt several MVME130 modules in a multi-processor system with a single VMEbus interrupt.

The Control Register bits are software alterable. The Status/Control registers are privileged resources and in special cases can be made accessible by the user. The Status Register format is shown in Table 2 and the Control Registers are described in Table 3.

TABLE 2 — Status Register Format

<STAT0>

31			Front Panel Switch Selectable				24
MODE	AMSEL	SC*	GPC	GPB	GPA	GP9	GP8

<STAT1>

23			Front Panel Switch Selectable				16
GP7	GP6	GP5	GP4	GP3	GP2	GP1	GP0

Provides VME130 Networking Base Address
and useful as general purpose status

<STAT2>

15			Power Up Reset Condition: "1"				08
**	**	ABORT	SYSFAIL*	VBERR*	VXBERR*	MMUBERR*	LBTO*

BERR Source Status Register

(**denotes: unimplemented — always read as "1").

MODE, AMSEL	<Power Up VMEbus Mode>
SC*	<VMEbus System Controller>
GPC-GP0	<General Purpose User Status Bits>
ABORT	<Abort Pushbutton Status>
SYSFAIL*	<VMEbus System Failure Flag>
VBERR*	<VMEbus BERR Flag>
VXBERR*	<VSB BERR Flag>
MMUBERR*	<MMU BERR Flag>
LBTO*	<Local Bus Timeout BERR Flag>

TABLE 3 — Control Register Format

<CNT0>

31	Power Up Reset Condition: "1"						24
VBV7	VBV6	VBV5	VBV4	VBV3	VBV2	VBV1	VBV0
← VMEbus IRQ Vector →							
← Port A on the Z8036 Timer →							

<CNT1>

31	Power Up Reset Condition: "1"						24
MPIRQ*	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
← Port B on the Z8036 Timer →							

<CNT2>

31	Power Up Reset Condition: "1"						24
VBIMSK7	VBIMSK6	VBIMSK5	VBIMSK4	VBIMSK3	VBIMSK2	VBIMSK1	VBISMSK
← which VMEbus IRQ's this Interrupt Handler responds to →							

<CNT3>

23	Power Up Reset Condition: "0"						16
Reserved	ALLIEN	SYSFIEN	Reserved	IL2	IL1	IL0	Reserved
← Interrupt Enables →				← VME IRQ Request Level, Status →			

<CNT4>

15	Power Up Reset Condition; "1"						08
Reserved	Reserved	Reserved	32/24*	Reserved	Reserved	Reserved	Reserved

<CNT5>

07	Power Up Reset Condition: "1"						00
PMEN	VXDIS	VXRR*	VXDEN	UXIRQ	32/24*	32/16*	BDFAIL

VBV7-VBV0	<VMEbus IRQ Vector Number Register>	VXDIS	<VSB Disable>
MPIRQ*	<Multi-Processor Interrupt Request>	VXRR*	<VSB Read Only Mode>
VBIMSK7-1	<VMEbus Interrupt Masks>	VXDEN	<VSB Decode Enable>
VBISMSK	<VMEbus Interrupt Status IRQ Mask>	VXIRQ	<VSB Interrupt Request>
ALLIEN	<All IRQ Enable>	32/24*	<VMEbus Address Size>
SYSFIEN	<SYSFAIL IRQ Enable>	32/16*	<VMEbus Data Bus Width>
IL2-IL0	<VMEbus IRQ Request Level>	BDFAIL	<Board of System Failure>

VMEbus INTERFACE

The VMEbus interface provides for VMEbus arbitration; for buffering of data, address, and control signals; for word data manipulation to accommodate MC68020 and VMEbus data handling differences and for Interrupt handling.

The VMEbus arbiter arbitrates up to four levels of bus mastership on a priority basis, utilizes a Pal chip, and incorporates the Release on Request (ROR) option described in the VMEbus specification (HB212).

INTERRUPT HANDLER

The MVME130 allows interrupts to the onboard CPU from up to 20 sources. The interrupt handler preprocesses interrupt sources into three groups of seven interrupts corresponding to the seven possible MC68020 interrupt levels. The groups are labeled Group 1, Group 2, and Group 3. The interrupt service priority is determined by the interrupt level and the group number. Interrupts with different interrupt levels are processed according to the standard interrupt processing discipline. Interrupts within an interrupt level are processed according to the group number.

Group 1 is reserved for VMEbus interrupts. The interrupt handler processes Group 2 and 3 interrupts differently from Group 1 interrupts. If the interrupt being acknowledged is a Group 2 or 3 interrupt, the interrupt handler fetches the appropriate exception vector number from PROM and sends it to the CPU via the local bus. If the interrupt being acknowledged is a Group 1 interrupt, the exception vector number is fetched from the VMEbus where it was placed by the interrupting device.

The Interrupt assignments are described in Table 4.

VMEbus INTERRUPTER

The VMEbus Interrupter can generate any level of VMEbus interrupt under software control. When the VMEbus interrupt is acknowledged, the Interrupter places the appropriate vector from a hardware register on the VMEbus.

TABLE 4 — Interrupt Handler Priority Assignments

IRQ Level	Priority Within a Particular IRQ Level Determines the Service Order (Highest) Group 3	(Middle) Group 2	(Lowest) Group 1
7	ABORT*	ACFAIL Interrupt ACFIRQ*	IRQ7* (VMEbus)
6	TMRIRQ*	SYSFIRQ*	IRQ6* (VMEbus)
5	SIOIRQ*	VBISIRQ*	IRQ5* (VMEbus)
4	Unassigned	VXIRQ*	IRQ4* (VMEbus)
3	Unassigned	Unassigned	IRQ3* (VMEbus)
2	Unassigned	Unassigned	IRQ2* (VMEbus)
1	N/A	Unassigned	IRQ1* (VMEbus)

Within a Group x, interrupt priority decreases from top to bottom in the table.
Within a particular IRQ level, interrupt priority decreases from left to right.

Group 3 Interrupts

ABORT*	<Abort Pushbutton IRQ>
TMRIRQ*	<Timer IRQ>
SIOIRQ*	<Dual Channel Serial Port IRQ>

Group 2 Interrupts

ACFIRQ*	<VMEbus AC Power Fail Interrupt>
SYSFIRQ*	<VMEbus System Fail Interrupt>
VBISIRQ*	<VMEbus IRQ Request Acknowledge Interrupt>
VXIRQ*	<VSB Interrupt>

Group 1 Interrupts

IRQ7-1*	<VMEbus Interrupts>

Mechanical and Environmental Specifications

Characteristics	Specifications
Power Requirements MVME130 MVME131	 +5 Vdc, 5 A (typ), 6 A (max) +5 Vdc, 6 A (typ), 7 A (max) +12 Vdc, 250 mA (max) −12 Vdc, 250 mA (max)
Clock Signal	12.5 MHz clock frequency
Addressing Total system size ROM/EPROM/RAM	 4 Gigabytes (32-bit addressing) Four 28-pin sockets for 16/32/64Kb devices using +5 Vdc only.
Operating Temperature	0° to 50°C
Storage Temperature	−40° to 85°C
Relative Humidity	5% to 95% (non-condensing)
Physical Dimensions Height Width	 6.3 in (16.0 cm) 9.19 in (23.34 cm)
Thickness (MVME130) (MVME131)	.71 in (1.8 cm) 1.51 in (3.84 cm)

Ordering Information

Part Number	Description
MVME130	VMEmodule 32-bit Monoboard Microcomputer with MC68020 CPU. Includes User's Manual.
MVME131	VMEmodule 32-bit Monoboard Microcomputer with MC68020 CPU and Hardware Memory Management. Includes User's Manual.
MVME130/D	VMEmodule 32-bit Monoboard Microcomputer User's Manual.

Related Product

Part Number	Description
MVME130bug	MVME130bug Debugging Package for VMEmodule 32-bit Monoboard Microcomputer. Includes User's Manual.
MVME204	VMEmodule 1024Kb Dynamic RAM Module with VSB. Includes User's Manual.
MVME214	VMEmodule Static RAM/ROM Module with VSB. Includes User's Manual.
MVME707	MVME130 RS-232C Distribution Board with Dual RS-232C Serial I/O Cable Assembly. Includes User's Manual.

Advance Information

130bug Debug/ Diagnostic Monitor

130bug RESIDENT PACKAGE

- Debug and Diagnostic Monitor in EPROM for the MC68020-based MVME130 and MVME131 Monoboard Microcomputers
- Full Speed Execution of System and User Programs in a MVME130/MVME131 Monoboard Microcomputer System
- More than 20 Debug and Program Development Commands
- 18 Diagnostic and Hardware Debug Commands
- Provides Access to all I/O, Control and Memory Facilities of a MVME130/MVME131-based System Plus the Full 4 Gigabyte Direct Address Range of the VMEbus
- Support for the MC68881 Floating Point Coprocessor (FPC)
- Supports the MVME320 Disk Controller
- Virtual Terminal Capability for Up/Down Load to/from any Host via Motorola S-Records
- One Line Assembler/Disassembler for Convenient Program Monitoring/Modification
- 21 I/O, Data Conversion and Timing Routines Callable from User-written Programs
- Complete MVME130/MVME131 Hardware Diagnostics with Loop Continuous and Loop-on-error Modes
- Self Test verifies System Integrity on Power Up

130bug SOURCE MODULE PACKAGE

- 130bug Source Modules Supplied on VME/10 Microcomputer System or EXORmacs Development System Diskette
- Source Modules Allow Modification of 130bug as Required

130bug DEBUG/DIAGNOSTIC MONITOR

The Debug/Diagnostic Monitor, 130bug, is supplied as two separate products:

- For Resident Use
  - — The object code is supplied in two 32Kb EPROMs which plug into sockets provided on the MVME130 and MVME131 Monoboard Microcomputers
- For Customization
  - — The source code is supplied on VME/10 Microcomputer System-compatible 5-1/4" diskettes or on EXORmacs Development System-compatible 8" diskettes

Any MVME130 or MVME131 system having a VMEbus backplane such as that provided by the chassis for one of Motorola's new "Open System" Microcomputer Families, the 9-slot MVME943 VMEmodule Chassis and the 20-slot MVME944 VMEmodule Chassis can use 130bug.

The 130bug firmware monitor is a versatile, effective tool. It offers many powerful commands for application program development and modification and for MVME130 or MVME131 system hardware diagnosis and debugging in a VMEbus or standalone environment. It permits, under complete operator control, full-speed execution of system and application programs including code utilizing the MC68881 Floating Point Coprocessor.

Memory space required by 130bug includes 8Kb of RAM storage for exception vectors and stack/work space and 8Kb for user space. The monitor normally reserves the 16Kb of static RAM on board MVME130 and MVME131 but can optionally search system memory space for off-board RAM. To operate the 130bug/monoboard combination in the standalone mode requires that the onboard memory be used.

With a user-supplied module translating the TTL levels of the two MVME130 or MVME131 serial ports to RS-232-C to create standard DB-25 interfaces (one such board is the Motorola MVME707 RS-232 Serial Port Distribution Module), a standard RS-232-C, asynchronous, ASCII terminal can be connected to one serial port so that the 130bug/monoboard system can be operated in a standalone mode. This mode allows full use of the 130bug program development and hardware diagnostic capabilities and is also commonly used for the final debugging of the application system. Figure 1 depicts operating in the standalone mode.

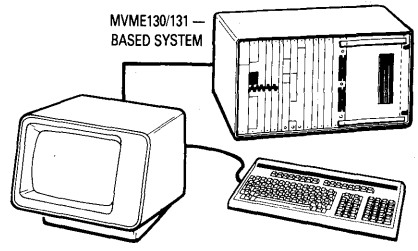


Figure 1. Standalone Operation

The translation board also allows a direct or dial-up RS-232-C link from a host computer to be interfaced to the second serial port so that programs in Motorola S-Record format may be up/down loaded. For this purpose, 130bug provides commands used to place the monoboard in the transparent mode and to configure the

ports for the desired communication protocol. For example, both host and target must be set to the same baud rate. Viewed from the host, while in the transparent mode, the monoboard appears to be an asynchronous ASCII terminal (virtual terminal). Figures 2 and 3 show operation in the transparent mode for communication with a host computer.

Support provided within 130bug for the MVME320 Disk Controller includes a command which can be used to load into a system any VERSAdos-bootable operating system.

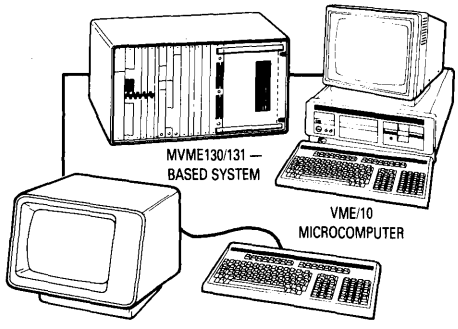


Figure 2. Transparent Mode — Direct Serial Connection With Host Computer

DEBUG/DIAGNOSTIC FACILITIES

The 130bug monitor is a command driven program which has facilities for accepting a command entered at the system console and for operating in the debug or in the diagnostic mode. To indicate the current operating mode, 130bug displays an individual prompt: 130Bug> or 130Diag>. Debugging activities are supported by a large set of versatile commands many of which can be used for diagnosis or self test of the MVME130 and MVME131 monoboards.

On power-up and reset, 130bug follows certain default condition procedures according to the value of a switch-settable flag on the monoboard. These include the allocation of onboard or of system memory.

PROGRAM DEBUGGING

Debugging support includes commands for examining and modifying registers and memory, commands which allow blocks of memory to be filled, moved or searched for the occurrence of specified data, breakpoint commands which allow program segments to be run, trace commands for examining the execution of instructions or small program portions, commands for beginning execution at a specified address, commands which facilitate host/target communication, printer attach/detach commands and utility commands which provide the capabilities of switching to the other operating mode, formatting the monoboard's serial ports, displaying or modifying the offset registers, converting the number base of data and displaying the name and a brief description of the entire command set. Table 1 lists all debugging commands.

Several commands are included for performing disk I/O using the MVME320 Disk Controller Module. The parameters of several commonly-used disk configurations are already known to the monitor and a command, I/O Teach, is included which allows a user to "teach" the monitor the specifics of another configuration. Particular sectors can be read from or written to using the I/O Physical To Disk command. Another command, Boot Operating System And Halt, is useful for debugging bootable code.

In a typical debug session after loading the program under development, appropriate commands from the debug set are invoked so that operation of a particular program portion can be checked and, if necessary, the code modified. Often, using the powerful Register Display/Modify, Memory Display/Modify and Block Memory Move commands, trial conditions for a program portion are established and the code executed. Results can be determined by examining the processor registers or memory values. A user may also examine the registers of the MC68881 FPC and display or change the contents of memory in any one of the various floating point formats.

An alternative method is to set a breakpoint in place of an appropriate instruction to halt execution and obtain automatic display of processor register values. A more detailed examination of program validity is offered by the Trace commands which allow execution of one instruction at a time, with accompanying processor register display following each execution. If required, program code is easily modified with the aid of the one line assembly/disassembly function provided by the Memory Modify command. The Data Conversion command which converts values from decimal to hexadecimal or hexadecimal to decimal is also a useful modification tool.

Debugging of modular code is done using the commands for displaying and modifying the offset registers which 130bug provides to make this task easier.

When the program is fully debugged, it is saved to disk or the Dump S-Records command is used to format and upload the code to the host for storage.

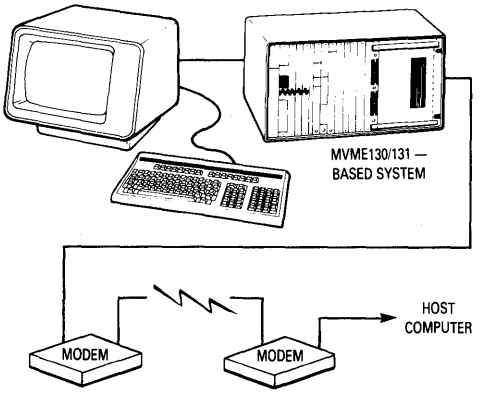


Figure 3. Transparent Mode — Dial-Up Serial Link With Host Computer

Table 1. 130bug Debugging Commands

MEMORY COMMANDS	SYNTAX
Memory Display	MD[S] <ADDR>[:<COUNT> | <ADDR>][;[B | W | L | D]]
Memory Modify/Disassembly/Assembly	MM <ADDR> [;[[N] [B | W | L | A]|DI]
Memory Set	MS <ADDR> {Hexadecimal number} {'string'}
REGISTER COMMANDS*	**SYNTAX**
Register Display	RD
Register Modify	RM <REG>
BLOCK OF MEMORY COMMANDS	**SYNTAX**
Block of Memory Fill	BF <RANGE><DEL><DATA>[;B | W | L]
Block of Memory Move	BM <RANGE><DEL><ADDR>[;B | W | L]
Block of Memory Search	BS <RANGE><DEL><TEXT> | <DATA>[<MASK>] [;B | W | L /N]
DISK SUPPORT COMMANDS	**SYNTAX**
Boot Operating System	BO
Boot Operating System and Halt	BH
I/O Physical To Disk	IOP
I/O "Teach"	IOT
HOST COMMUNICATION COMMANDS	**SYNTAX**
Dump S-Records	DU[n] <RANGE><DEL>[<TEXT>] [<ADDR>] [;B | W | L]
Load S-Records	LO [<ADDR>] [;[-C/X]=<TEXT>]
Transparent Mode	TM <ESCAPE>
Verify S-Records	VE [<ADDR>] [;[-C/X]=<TEXT>]
GO COMMANDS	**SYNTAX**
Go Direct (Ignore Breakpoints)	GD [<ADDR>]
Go Execute User Program	GO [<ADDR>]
Go To Next Instruction	GN
Go To Temporary Breakpoint	GT <ADDR>
TRACE COMMANDS	**SYNTAX**
Trace	T [<COUNT>]
Trace on Change of Control Flow	TC [<COUNT>]
Trace To Temporary Breakpoint	TT <ADDR>
BREAKPOINT COMMANDS	**SYNTAX**
Breakpoint Insert	BR {<ADDR>[:<COUNT>]}
Breakpoint Delete	NOBR {<ADDR>}
PRINTER COMMANDS	**SYNTAX**
Printer Attach	PA
Printer Detach	NOPA
MISCELLANEOUS UTILITY COMMANDS	**SYNTAX**
Data Conversion	DC <expression>
Help	HE [<COMMAND>]
Offset Registers Display/Modify	OF [Rn[;A]]
Port Format	PF[n]
Switch Directories	SD

HARDWARE DIAGNOSTICS

Test and diagnostic capabilities offered by 130bug are provided by commands which include:

- Set-up Utilities for Selecting an Operating Mode such as Stop-on-error
- Tests of MVME130/131 Hardware Including Tests of the Onboard Memory, MC68020 Cache, Counter/Timer, Serial I/O, VMEbus/VSB, Tests of the Memory Management Board and Final Assembly Tests
- Commands for Hardware Debugging which Provide a Self-test Sequence and Continuous Write and Read Loops

When the 130bug diagnostic mode is entered, the monitor displays the prompt: "130Diag>" indicating that any of the commands in the diagnostic directory may be used. The 130bug Diagnostic Commands are listed in Table 2.

CALLABLE 130bug ROUTINES

A TRAP #15 handler is included in 130bug which allows calls from user-written programs to input, output and other useful routines within 130bug. Such a system call is made by including in the source program a TRAP #15 instruction line followed by a DC.W instruction line containing the code of the desired function. The callable 130bug functions and their codes are listed in Table 3.

Table 2. 130bug Diagnostic Commands

Type	Command Mnemonic	Description
Diagnostic Set-Up Utilities	LE	Loop-on-Error Mode
	SE	Stop-on-Error Mode
	LC	Loop Continuous Mode
	NV	Non-Verbose Mode
	DP	Display Pass Count
	ZP	Clear Pass Count
	DE	Display Errors
	ZE	Clear Error Counters
Hardware Debugging Tools	ST	Run Self-Test Sequence
	WL	Continuous Write Loop
	RL	Continuous Read Loop

Type	Command Mnemonic	Description
Hardware Diagnostics	FAT	Final Assembly Tests
	MT	Memory Tests
	CA20	MC68020 On-Chip Cache Tests
	CIO	Counter/Timer Tests
	SIO	Serial I/O Tests
	BUS	VMEbus and VSB Tests
	MMB	Memory Management Board Tests

Table 3. Callable 130bug Functions

Code	Function	Description
$0000	.INCHR	Input character from default input port
$0001	.INSTAT	Input serial port status of default input port
$0002	.INLN	Input line from default input port (format 1)
$0003	.READSTR	Input string from default input port (format 2)
$0004	.READLN	Input line from default input port (format 2)
$0020	.OUTCHR	Output character to default output port
$0021	.OUTSTR	Output string to default output port (format 1)
$0022	.OUTLN	Output line to default output port (format 1)
$0023	.WRITE	Output string to default output port (format 2)
$0024	.WRITELN	Output line to default output port (format 2)
$0025	.WRITDLN	Output line w/data to default output port (format 2)
$0026	.PCRLF	Output carriage return and line feed to default port
$0027	.ERASLN	Erase line being input from default input port
$0040	.TM_INI	Initialize MVME130's onboard timer
$0041	.TM_STR0	Start timer at T = 0
$0042	.TM_RD	Read timer
$0060	.REDIR	Redirect I/O of a TRAP 15 function to use other than the default input or output port
$0061	.REDIR_I	Change the default input port number
$0062	.REDIR_O	Change the default output port number
$0063	.RETURN	Return to 130bug
$0064	.BINDEC	Convert binary to decimal

Format 1 — pointer/pointer format Format 2 — pointer/count format

ORDERING INFORMATION

Part Number	Description
MVME130BUG	130bug, the Debug/Diagnostic Monitor for the MVME130 and MVME131 Monoboard Microcomputers. Includes EPROM set and User's Manual.
M68V2FSBG130	130bug Source Modules on EXORmacs 8" Diskettes. Includes User's Manual.
M68V2XSBG130	130bug Source Modules on VME/10 5-1/4" Diskettes. Includes User's Manual.
MVME707	TTL To RS-232-C Level Translator Module and Dual RS-232-C Serial I/O Cable Assembly. Includes User's Manual.
MVME130BUG/D	130bug Debug/Diagnostic Monitor User's Manual.

MVME130XT
MVME131XT

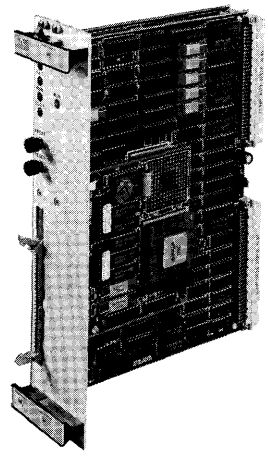
VMEmodule™ 32-Bit Microcomputer with Cache Accelerator

- MC68020 Microprocessor with 32-bit Address and Data Bus
- MC68881 Floating Point Coprocessor
- Demand Paged Virtual Memory Management Capability (MVME131XT only)
- VSB* Interface for High-Speed Data Path to and from Memory
- 16Kb Cache Memory for Concurrent Data Storage with Accelerated Data Access, and Zero Wait-state Operation
- Onboard Interrupt, Interrupt Handler and Arbiter
- Programmable Timer
- Dual Multiprotocol Serial I/O Ports

The MVME130XT is a two-board VME-based microcomputer assembly utilizing state-of-the-art components to achieve the highest level of performance currently available. It consists of an MVME130 Monoboard Microcomputer Module, factory coupled to an MVMEXTCAC Cache Accelerator Module . . . a combination which provides up to 100% improvement in system speed[1] compared with the MVME130 Monoboard Microcomputer alone.

The two-board system plugs into a standard VMEbus backplane and is fully compatible with the wide selection of VME peripheral modules.

Utilizing the MC68020 Microprocessor, the MVME130XT has onboard provisions for plugging in an MC68881 Floating Point Coprocessor and an MC68851 Paged Memory Management Unit when the latter becomes available. In the meantime, a second version of the system, the MVMX131XT, employs an optional auxiliary Memory Management Board which conforms to a subset of the MC68851 and plugs into an onboard socket of the assembly to provide interim memory management functions.



GENERAL DESCRIPTION

The MVME130XT Microcomputer Assembly combines the processing power of the 32-bit MC68020 Microprocessor with the speed-enhancement properties of a 16Kb Cache Memory Accelerator. The Cache Accelerator, a small, fast memory system, compensates for the typical speed mismatch between a very fast CPU and its relatively slow associated dynamic RAM main-memory system. It does so by concurrently storing the data most recently stored in main-memory locations. This data can subsequently be obtained by fast accesses of "cached" locations rather than by much slower accesses of main memory.

To accommodate for the fact that the cache memory normally has much less capacity than the main memory system, little-used accesses stored in the cache memory are routinely replaced with more active ones, based on the bus traffic between the MVME130 Microprocessor Module and the VMEbus and VSB memory interfaces. This assures maximum utilization of the cache.

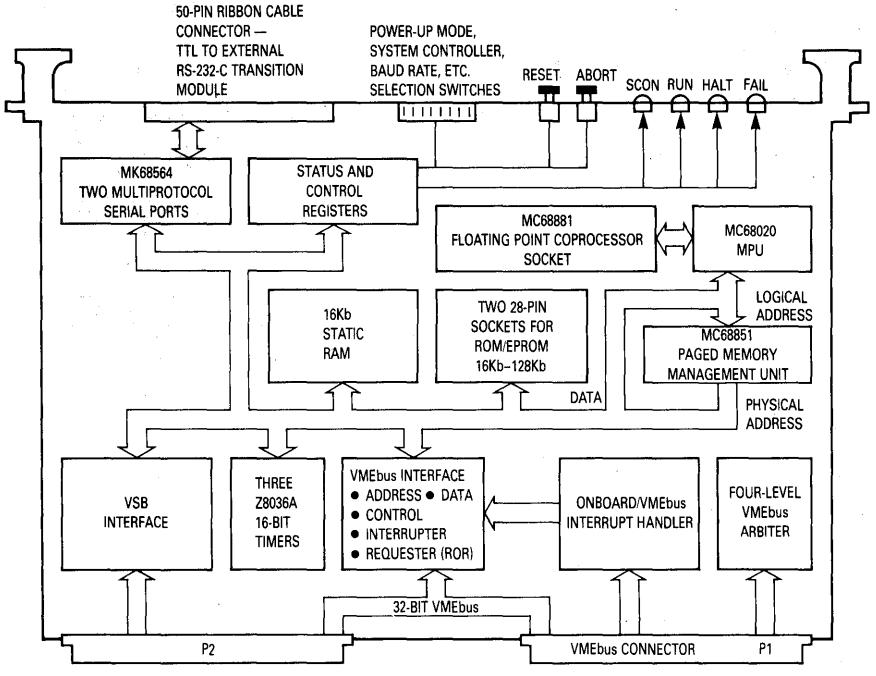


Figure 1. MVME131 Functional Block Diagram

THE MICROCOMPUTER BOARD — MVME130/MVME131

MICROPROCESSORS

A block diagram of the MVME131 Microcomputer Board is shown in Figure 1. The heart of the system is the MC68020 Microprocessor operating at a fixed frequency of 16.67 MHz. This latest member of the Motorola M68000 Family offers full 32-bit operation with 32-bit external address and data paths. With its high clock rate, enhanced addressing modes, and on-chip instruction cache, this advanced MPU provides leading-edge performance while maintaining software compatibility with its widely accepted predecessors.

On the MVME130XT Module the MC68020 Microprocessor is supported by an onboard socket that is prewired to accept the MC68881 Floating Point Coprocessor. This optional accessory chip greatly improves computing speed of the system and is recommended for applications involving a high level of arithmetic operations.

OPTIONAL MEMORY EXPANSION

In support of the microprocessor(s), the microcomputer board has two prewired 28-pin sockets for up to 16Kb of Static RAM. If the intended application does not require onboard RAM, however, these JEDEC sockets can be reconfigured to accept EPROMs up to 512K in size.

With one or more MVME204-1 or -2 DRAM with VSB Modules which provide 1Mb and 2Mb, respectively, the MVME130 can avail itself of virtually unlimited memory capacity while maintaining a performance level similar to that of a Monoboard VMEmodule Microcomputer with onboard memory. Also available for high performance memory expansion is the MVME214 Static RAM/ROM with VSB. This module has provisions for battery backup and has 16 sockets which accept many standard JEDEC 24-pin and 28-pin RAM, ROM and EPROM devices. It can provide up to 1Mb of storage capacity.

Memory modules for expanding VME system memory include the MVME202, MVME222-1 and MVME222-2 DRAM modules which have a 8-/16-bit data bus width and offer 512K, 1Mb and 2Mb of capacity, respectively, the MVME225 DRAM, a full 32-bit module offering 2Mb of capacity, the MVME215-1, -2, and -3 CMOS RAM modules with onboard power monitor and battery for power down backup and which offer 256Kb, 512Kb and 1Mb of capacity, respectively, and the MVME211 Static RAM/ROM module with 16 sockets for JEDEC 24-pin and 28-pin devices and which can provide up to 1Mb of storage capacity.

MEMORY MANAGEMENT UNIT

The MVME130XT anticipates the eventual utilization of the MC68851 Demand Paged Virtual Memory Management Unit with an onboard socket prewired for this device. In the interim, until the MC68851 MMU becomes available, Motorola furnishes the MVME131XT System which includes a Daughter Board plugged in the socket that is allocated for the MMU Chip. This auxiliary board contains circuitry that conforms to a subset of the MC68851, and provides the hardware logical-to-physical address translation and access protection necessary for operating system support.

VMEbus INTERFACE

The VMEbus interface permits:

- VMEbus arbitration
- Buffering of data, address and control signals
- Word data manipulation
- Interrupt handling

The VMEbus arbiter arbitrates up to four levels of bus mastership on a priority basis, and incorporates the Release on Request option described in the VMEbus specification.

MVMX32bus INTERFACE

This interface occupies 64 I/O pins which provide access to the system's high-speed memory channel. The signals utilize multiplexing of address and data to accommodate full 32-bit functionality, along with appropriate control signals, within the 64-pin allotment. On the MVME130 Module, this interface is limited to serving one master. It can be turned off via a control bit to force all accesses to go to the VMEbus, and speed up VMEbus accesses.

PROGRAMMABLE TIMER

The onboard programmable timer provides three independent cascadable 16-bit counters with interrupt capability.

SERIAL I/O

The Module is equipped with two multiprotocol serial I/O ports accessible via a 50-pin flat ribbon cable connector at the top of the board. Signal levels at this connector correspond to TTL specifications, but can be transformed to RS-232-C levels by means of a separately available distribution board and cable assembly (MVME707). The ports have full- and half-duplex compatibility with programmable baud rates. They are suitable for synchronous or asynchronous operation, with 5 to 8 bits per character, plus parity.

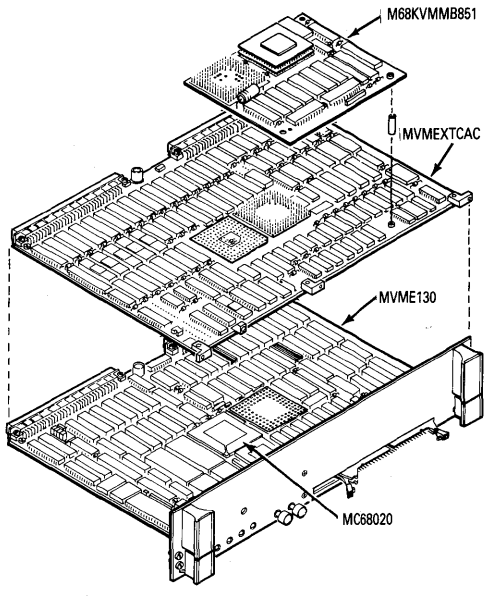


Figure 2. Exploded View Showing MVME131XT Assembly Details

MEMORY MAP

Table 1 indicates the 32-bit memory address map for various devices addressed by the MVME130 Module. A 24-bit address map is determined by eliminating the two most significant hex digits from the 32-bit addresses. A control bit on the control register permits a switch between 16- and 32-bit data paths, and between 32-/24-bit modifier codes.

Table 1. Address Map

Device	Physical Address	Comments
VMEbus/VSB	00000000-FFEFFFFF	32/16-bit port size; R/W.
ROM	FFF00000-FFF8FFFF	16-bit port size; read only.
Local Static RAM	FFF20000-FFF2FFFF	16-bit port size; R/W.
Spare	FFF40000-FFF6FFFF	32-bit port size; R/W provided for cache expansion ability.
Timer	FFFBxx00-FFFBxx2F	8-bit port size; R/W.
Status & Control	FFFBxx30-FFFBxx3F	32-bit port size; Status is read only; Control is R/W.
Serial I/O Port	FFFBxx40-FFFBxx5F	8-bit port size; most are R/W.
Reserved	FFFBxx60-FFFBxx6F	Unimplemented.
Spare	FFFBxx70-FFFEFFFF	Unimplemented; Access will cause LBTO (BERR source) cycle termination.
VMEbus Short I/O	FFFF0000-FFFFFFFF	16-bit; R/W; VMEbus Short I/O Address Modifier will be generated.

INTERRUPT HANDLER

The MVME130 allows interrupts to the onboard CPU from up to 20 sources. The interrupt handler preprocesses interrupt sources into three groups of seven interrupts corresponding to the seven possible MC68020 interrupt levels. The groups are labeled Group 1, Group 2, and Group 3. The interrupt service priority is determined by the interrupt level and the group number. Interrupts with different interrupt levels are processed according to the standard interrupt processing discipline. Interrupts within an interrupt level are processed according to the group number.

Group 1 is reserved for VMEbus interrupts. The interrupt handler processes Group 2 and 3 interrupts differently from Group 1 interrupts. If the interrupt being acknowledged is a Group 2 or 3 interrupt, the interrupt handler fetches the appropriate exception vector number from PROM and sends it to the CPU via the local bus. If the interrupt being acknowledged is a Group 1 interrupt, the exception vector number is fetched from the VMEbus where it was placed by the interrupting device.

The Interrupt assignments are described in Table 2.

VMEbus INTERRUPTER

The VMEbus Interrupter can generate any level of VMEbus interrupt under software control. When the VMEbus interrupt is acknowledged, the Interrupter places the appropriate vector from a hardware register on the VMEbus.

STATUS/CONTROL REGISTERS

The MVME130 Status and Control Registers provide local and system-wide status information from which the onboard MPU can determine VMEbus information, interrupt status, the source of a particular BERR, and which self-test to perform. The MVME130 Status/Control Register bits control the module's hardware by providing configuration and control information to the VMEbus Interrupter, the Interrupt Handler, the VMEbus Arbiter and VSB.

The Control Register bits are software alterable. The Status/Control registers are privileged resources and in special cases can be made accessible by the user. The Status and Control Register formats are described in Tables 3 and 4, respectively.

Table 2. Interrupt Handler Priority Assignments

IRQ Level	Priority Within a Particular IRQ Level Determines the Service Order (Highest)	(Middle)	(Lowest)
	Group 3	Group 2	Group 1 (VMEbus)
7	ABORT	ACFAIL Interrupt ACFIRQ	IRQ7
6	TMRIRQ	SYSFIRQ	IRQ6
5	SIOIRQ	VBISIRQ	IRQ5
4	Unassigned	VXIRQ	IRQ4
3	Unassigned	Unassigned	IRQ3
2	Unassigned	Unassigned	IRQ2
1	N/A	Unassigned	IRQ1

Within a Group x, interrupt priority decreases from top to bottom in the table.
Within a particular IRQ level, interrupt priority decreases from left to right.

Group 3 Interrupts

ABORT	<Abort Pushbutton IRQ>
TMRIRQ	<Timer IRQ>
SIOIRQ	<Dual Channel Serial Port IRQ>

Group 2 Interrupts

ACFIRQ	<VMEbus AC Power Fail Interrupt>
SYSFIRQ	<VMEbus System Fail Interrupt>
VBISIRQ	<VMEbus IRQ Request Acknowledge Interrupt>
VXIRQ	<VSB Interrupt>

Group 1 Interrupts

IRQ7-1	<VMEbus Interrupts>

Table 3. Status Register Format

<STAT0>

31	Front Panel Switch Selectable						24
MODE	AMSEL	SO	GPC	GPB	GPA	GP9	GP8

<STAT1>

23	Front Panel Switch Selectable						16
GP7	GP6	GP5	GP4	GP3	GP2	GP1	GP0

Provides VME130 Networking Base Address

and useful as general purpose status

<STAT2>

15	Power Up Reset Condition: "1"						08
**	**	ABORT	SYSFAIL	VBERR	VXBERR	MMUBERR	LBTO

BERR Source Status Register

(**denotes: unimplemented — always read as "1").

MODE, AMSEL	<Power Up VMEbus Mode>
SC	<VMEbus System Controller>
GPC-GP0	<General Purpose User Status Bits>
ABORT	<Abort Pushbutton Status>
SYSFAIL	<VMEbus System Failure Flag>
VBERR	<VMEbus BERR Flag>
VXBERR	<VSB BERR Flag>
MMUBERR	<MMU BERR Flag>
LBTO	<Local Bus Timeout BERR Flag>

Table 4. Control Register Format

<CNT0>

31	Power Up Reset Condition: "1"						24
VBV7	VBV6	VBV5	VBV4	VBV3	VBV2	VBV1	VBV0

VMEbus IRQ Vector

Port A on the Z8036 Timer

<CNT1>

31	Power Up Reset Condition: "1"						24
MPIRQ	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Port B on the Z8036 Timer

<CNT2>

31	Power Up Reset Condition: "1"						24
VBIMSK7	VBIMSK6	VBIMSK5	VBIMSK4	VBIMSK3	VBIMSK2	VBIMSK1	VBISMSK

which VMEbus IRQ's this Interrupt Handler responds to

<CNT3>

23	Power Up Reset Condition: "0"						16
Reserved	ALLIEN	SYSFIEN	Reserved	IL2	IL1	IL0	Reserved

Interrupt Enables — VME IRQ Request Level, Status

<CNT4>

15	Power Up Reset Condition: "1"						08
Reserved	Reserved	Reserved	32/24	Reserved	Reserved	Reserved	Reserved

<CNT5>

07	Power Up Reset Condition: "1"						00
PMEN	VXDIS	VXRR	VXDEN	UXIRQ	32/24	32/16	BDFAIL

VBV7-VBV0	<VMEbus IRQ Vector Number Register>
MPIRQ	<Multi-Processor Interrupt Request>
VBIMSK7-1	<VMEbus Interrupt Masks>
VBISMSK	<VMEbus Interrupt Status IRQ Mask>
ALLIEN	<All IRQ Enable>
SYSFIEN	<SYSFAIL IRQ Enable>
IL2–IL0	<VMEbus IRQ Request Level>
VXDIS	<VSB Disable>
VXRR	<VSB Read Only Mode>
VXDEN	<VSB Decode Enable>
VXIRQ	<VSB Interrupt Request>
32/24	<VMEbus Address Size>
32/16	<VMEbus Data Bus Width>
BDFAIL	<Board of System Failure>

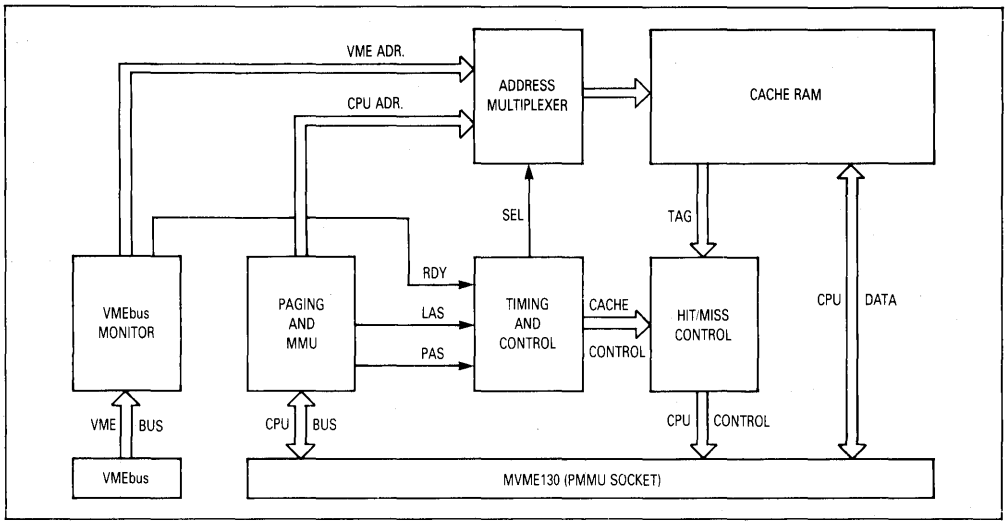


Figure 3. MVMEXTCAC Functional Block Diagram

THE CACHE ACCELERATOR MODULE

The Cache Accelerator Module (MVMEXTCAC), normally an independent product of the Motorola VME-module Family, is plugged into the MMU socket of the MVME130 Microcomputer Module and becomes an integral part of the Assembly. Its function is to accelerate the performance of MC68020-based systems without modification of existing programs. The degree to which this objective is being achieved is indicated by the results of the two benchmark programs tabulated in Table 5. The first of these utilized EDN benchmarks[2] which perform commonly used character and data manipulations; the second utilized user-supplied routines performing data intensive array operations. The tests, run for four combinations of cache memory utilization, indicate the degree of improvement attributable to the use of the Cache Accelerator.

Table 5. Performance Test Results

Test	No Cache	MC68020 Cache	MVMEXTCAC Cache	MC68020 and MVMEXTCAC Cache
EDN-E (Char. String Search)	285	141	131	99 μs
EDN-F (Set/Reset/Test Bit)	60	36	33	30 μs
EDN-H (Link List Insertion)	121	194	70	70 μs
EDN-I (Quicksort)	22,320	11,792	11,072	9001 μs
EDN-K (Bit Matrix Translation)	243	173	159	136 μs
CUSTOMER BENCHMARK	125	104	73	68 sec.

2. EDN — September 16, 1981

The Accelerator is a single-set, direct-mapped cache comprising 16Kb of high-speed static memory, organized as 4K, 32-bit data entries. Each entry replicates a 32-bit word located in memory across the VMEbus or VSB memory space, and is updated concurrently with a processor write-access or initial read-access. Any combination of bytes from an access can be cached at a time, and a data-fill function implemented for VSB enables the caching of the entire 32-bit entry for byte- and word-read operations. The Accelerator caches both program and data transactions throughout, and it can be configured, by function code, to cache any combination of supervisor program, supervisor data, user program and user data mode accesses.

An onboard Monitor automatically cleanses the cache of stale data and enables the Cache Accelerator to operate in both single and multiprocessor environments without software intervention.

The Cache Accelerator Module supports any speed MC68020 microprocessor, and is jumper selectable to support either the MC68851 Memory Management Unit or M68KVMMB851 Memory Management Board, or an MMU bypass board if the MMU function is not required. With an MMU installed, the page detection circuitry on the cache enables its RAMs to operate in parallel with the MMU address translation RAMs to maximize system performance. A cache page is defined by the physical address bits PA08–PA13, and if they remain the same on a following cycle, then the address access time of the cache RAMs is buried in the MMU address translation period. With a bypass board installed, the cache receives the address signals directly from the MC68020 microprocessor and zero wait-state operation is obtainable. Zero wait-state operation is achieved at a processor frequency of

12.5 MHz[3], and it may be achieved at 16.67 MHz when faster static RAMs become available.

The Cache Accelerator provides direct access to the cache RAMs for diagnostic and inquiry purposes, and provides a test mode of operation to establish general system confidence.

MEMORY ORGANIZATION

The complete memory section of the Cache Accelerator is organized as 4K entries of 55 bits per entry (see Table 6). It consists of a "Tag Section" and a "Data Section," where the tag section stores the physical addresses of the off-board memory locations, and the data section stores the data residing at those locations. The lower address bits (PA02–PA13) form an "index" that directly addresses the cache RAMs to select the particular entry to be used, and the remaining address bits (PA14–PA31) are stored and read from the tag RAMs as data. An Address Valid bit (AV) is generated internally for each entry to indicate that the address stored in the Tag is valid information. It may not be valid either because the tag entry has not been written to since power-up or since the cache was last cleared, or because the VMEbus Monitor has set the entry invalid. The data RAMs contain the corresponding bytes that were cached from the bus, and the four Byte-Valid bits (BV0–BV3) are generated internally to qualify those bytes. BV0 is associated with byte address 0 of the long-word entry, and operates on the most significant byte of the data bus, D31–D24. The other Byte-Valid bits operate in sequence.

BASIC CACHE OPERATION

When the processor drives an address for a read operation, the index section PA02–13 addresses the tag and data RAMs to select one entry from the cache. The remaining address bits from the processor are then compared to the corresponding address bits read from the tag section of that entry. If they match, and the Address-Valid bit is set, and if the Byte-Valid bits for the bytes the processor is requesting also match, then the cycle "hits" and the cache supplies the requested data to the processor and terminates the cycle. If any event does not match, then the cycle "misses" and a VME or VSB bus access is initiated, the cache entry is updated with the new address and data information, and the Valid bits are set appropriately. Note that a write cycle will always cause a miss since the processor, and not the cache, is supplying the data. Several other factors determine the hit qualification and cacheability of a processor cycle. These include the control bits, the type of processor cycle, the MMU cache enable signal and the onboard address decoder.

VMEbus MONITOR

The VMEbus Monitor is a separate machine on the Cache Accelerator that operates asynchronously with the processor to prevent "stale" data from remaining in the cache. It operates automatically so that the host user benefits from the performance advantages offered by the cache without even being aware that the Cache Accelerator has been installed in the system.

Stale data could occur in the cache when other VMEbus masters modify data in main memory which was previously cached. To prevent this, the Bus Monitor has a two-level FIFO register stack which captures the addresses of write operations initiated by bus masters other than the host, and presents the addresses to the cache for processing. The cache services the monitor by comparing the captured address with the corresponding entry in its tag memory. If the tag is Valid and compares, a "monitor hit" occurs and the cache invalidates the entry by resetting its Address Valid bit. If the processor then accesses that physical address, the cache will miss because of the reset Address Valid bit, and the cycle will be routed out to main memory for the new data to update the invalid entry.

USER INTERFACE

The user interface in Table 7 shows the devices on the Cache Accelerator used to control and diagnose its operation. These are Cache Control and Mask Registers, and two hard-decoded address spaces used to directly access the cache RAMs for diagnostic and inquiry purposes. The cache powers-up in a cleared and disabled state, and can be enabled for software transparent operation by storing an "F1" to the Cache Control Register. In addition, the cache can be tested or configured for a specific user application through the various control bits shown. Since the entire user interface address space can safely be accessed with or without the Cache Accelerator installed, a cache utility can be implemented without prior knowledge of its presence in the system.

MEMORY MANAGEMENT BOARD

The M68KVMMB851 Memory Management Board, Figure 4, is an interim solution for implementing demand-paged virtual memory operations prior to the availability of the MC68851 single-chip Paged Memory Management Unit. It provides the required logical to physical address translation by performing chip-controlled searching of translation tables in main memory.

The 4.5" x 3.5" mezzanine board plugs into a socket of the 131XT board which is reserved for the MC68851 chip. It contains an MC68461 Memory Management Controller and an external Address Translation Cache. The Controller contains address translation circuitry and a state machine which, together, execute the table walking algorithm of the MC68851. The Address Translation Cache, a 512-entry set associative cache, is a fast memory that translates logical addresses and function codes to corresponding physical addresses and protection bits. It speeds address translation by avoiding the overhead of searching translation tables for each bus access.

The functional differences between the MC68851 PMMU and the MMB are detailed in Table 8. The MMB requires an addition to the supervisor's bus error handler to allow the use of indivisible memory cycles (i.e. the TAS instruction). The MMB must use the bus error signal to inform the MC68020 microprocessor that the address of this indivisible cycle is not resident in the address translation cache. When the MC68020 takes the bus error signal, the MMB is allowed to request the bus and to perform

3. With MMU bypass card installed and dependent on MC68020 clock to address strobe timing.

the necessary address translation table walk. The MC68020 then proceeds to the bus error routine where software must perform an instruction retry to the indivisible cycle.

Functionally, the MMB is compatible with the MC68851 PMMU to the extent that algorithms and programs developed for a system using the MMB can be moved to one using the MC68851 after a few minor changes.

Table 6. MVMEXTCAC — Memory Organization

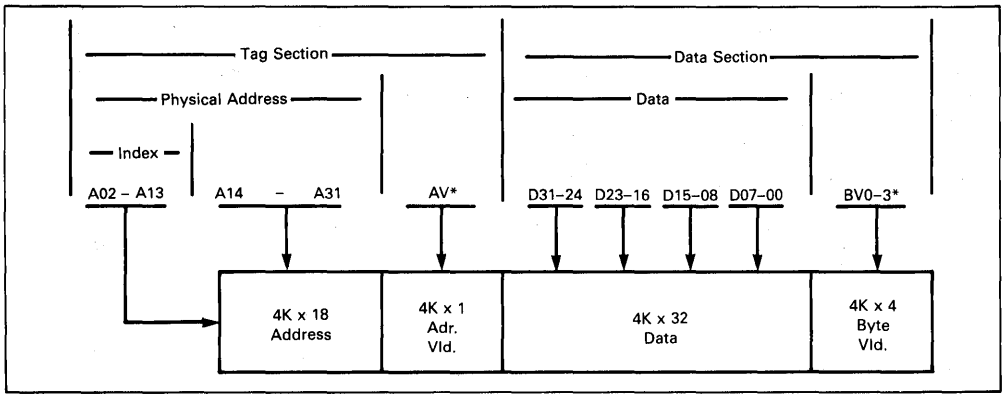


Table 7. MVMEXTCAC — User Interface

Cache Control Register — Address FFFC0000

(Reset State = FF)

D31							D24
VBCEN	MONEN	MLBK	MBLC	CCLR	CWEN	CREN	CTST

Mask Register — Address FFFC0001

(Reset State = xF)

D23							D16
X	X	X	X	MSD	MSP	MUD	MUP

Data Access Space — Address FFFD000–FFFD3FFC
Data Entries, 000–FFF

D31			D00
BYTE0 D31–D24	BYTE1 D23–D16	BYTE2 D15–D08	BYTE3 D07–D00
.	.	.	.
.	.	.	.
.	.	.	.

Tag Access Space (read-only)
Addresses — FFFE0000–FFFE3FFC
Tag Entries, 000–FFF

D31 D14	D07	D03–D00
PAD31–PAD14	NA	AV	BV0–BV3
.	.	.
.	.	.
.	.	.

DEFINITIONS

VBCEN	— VMEbus Cache Enable	MSP	— Mask Supervisor Program
MONEN	— VMEbus Monitor Enable	MUD	— Mask User Data
MLBK	— Monitor Loopback (for diagnostic use)	MUP	— Mask User Program
MBLC	— Monitor Block (for diagnostic use)	AV	— Address Valid Status
CCLR	— Cache Clear	BV0	— Byte Valid Status for D31–D24 (byte address 0)
CWEN	— Cache Write Enable	BV1	— Byte Valid State for D23–D16 (byte address 1)
CREN	— Cache Read Enable	BV2	— Byte Valid Status for D15–D08 (byte address 2)
CTST	— Cache Test (for diagnostic use)	BV3	— Byte Valid Status for D07–D00 (byte address 3)
MSD	— Mask Supervisor Data		

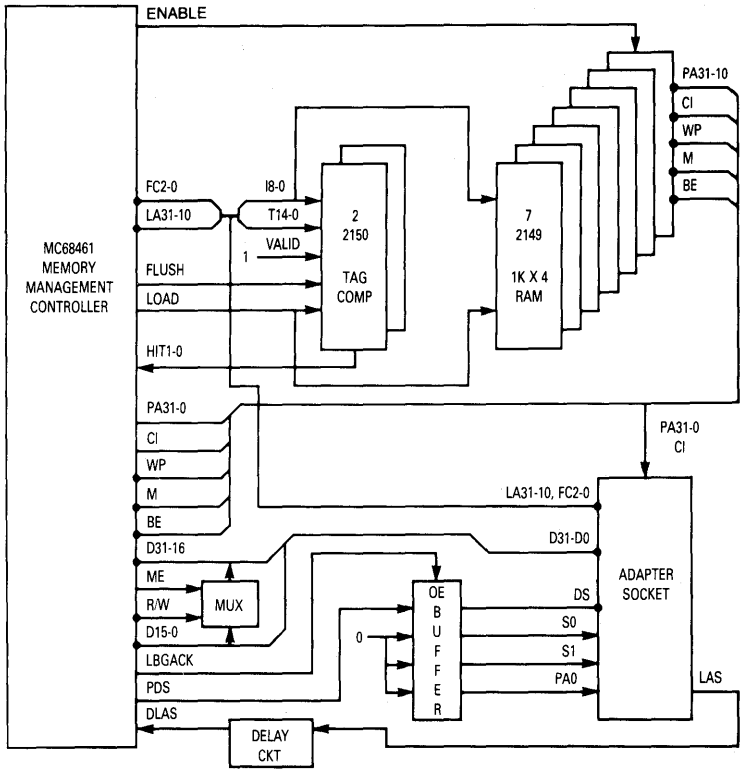


Figure 4. M68KVMMB851 Memory Management Board

Table 8. Functional Differences Between Chip and Board

Feature	MC68851 PMMU	MC68KVMMB851 Boards
Address Translation Cache	Internal, Fully Associative	External, Set Associative
Page Size	256, 512, 1K, 2K, 4K, 8K, 16K & 32Kb	1Kb
Logical Address Support	From 17 to 32 Bits	24- or 32-Bit
Supports Multiple Logical Bus Masters	Yes	No
Logical and Physical Bus Arbitration Support	Yes	Yes, But Not Identical to the MC68851
Access Level Support	Yes	Does Not Support Supervisor Root Pointer, Root Pointer Cache or DMA Root Pointer
Coprocessor Interface Support	Yes	No, Operations are Performed by Accessing Memory Mapped Locations
Translation Table Configuration	Supports Several	Expects One Given Table Walking Algorithm
Translates CPU Space Accesses	Yes	No, Acts as Peripheral Only
Pin-out	—	Compatible with MC68851

2

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristic	Specification
Cache Implementation	Single-set, direct mapped, physical
Memory Organization	4K, 32-bit data entries
Replacement Algorithm	Single entry, write-through
Hit Timing, LAS to DSACKx	No MMU — 60 nanoseconds With MMU — 70 nanoseconds (Page Hit) — 130 nanoseconds (Page Miss)
Power Requirements MVME130XT MVME131XT	 +5 Vdc, 9.3 A typ, 12.4 A max +5 Vdc, 10.3 A typ, 13.4 A max +12 Vdc, 250 mA max −12 Vdc, 250 mA max
Clock Signal	16.67 MHz clock frequency
Addressing Total System Size ROM/EPROM/RAM	 4 Gigabytes (32-bit addressing) Four 28-pin sockets for 16/32/64Kb devices using +5 Vdc only.
Operating Temperature	0° to 50°C
Storage Temperature	−40° to 85°C
Relative Humidity	5% to 95% (non-condensing)
Physical Characteristics Height Width Thickness (MVME130XT) (MVME131XT)	 6.30 in (16.00 cm) 9.19 in (23.34 cm) 1.438 in (3.652 cm) 2.238 in (5.684 cm)
Component Projections Component Side Solder Side	 0.50 in (1.27 cm) max 0.067 in (0.17 cm) max (except for interconnect header)

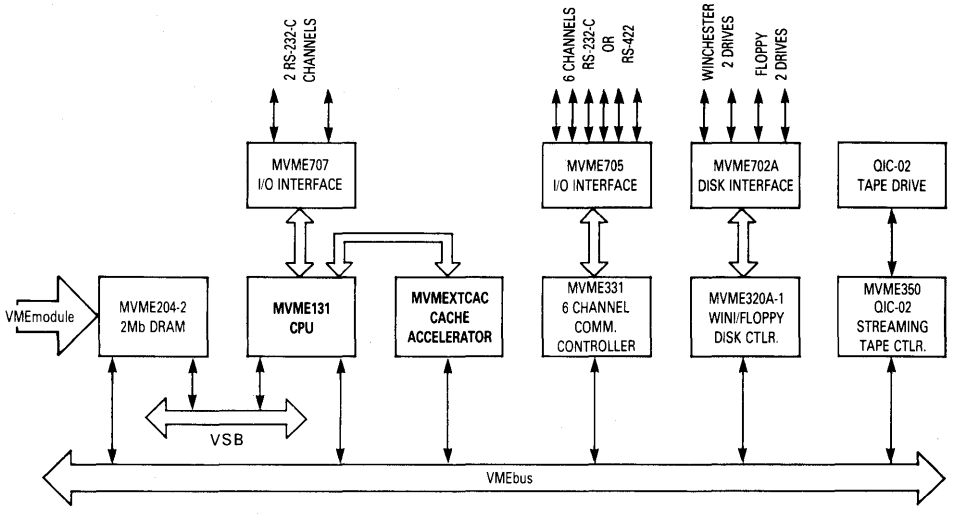
ORDERING INFORMATION

Part Number	Description
MVME130XT	VMEmodule 32-Bit Monoboard Microcomputer with 16.67 MHz MC68020 CPU and MC68881 Floating Point Coprocessor, MVMEXTCAC-1 Cache Accelerator Module and MVME130bug Debug/Diagnostic Monitor. Includes set of User's Manuals.
MVME131XT	VMEmodule 32-Bit Monoboard Microcomputer with 16.67 MHz MC68020 CPU, MC68881 Floating Point Coprocessor, MVMEXTCAC-2 Cache Accelerator Module with Memory Management Unit and MVME130bug Debug/Diagnostic Monitor. Includes set of User's Manuals.
MVME130XT/D	User's Manual Set for MVME130XT.
MVME131XT/D	User's Manual Set for MVME131XT.
MVME130BUG/D	130bug Debug/Diagnostic Monitor User's Manual.

RELATED DOCUMENTATION

Part Number	Description
MVME204-1	VMEmodule 1024Kb Dynamic RAM with Byte Parity and VSB. Includes User's Manual.
MVME204-2	VMEmodule 2048Kb Dynamic RAM with Byte Parity and VSB. Includes User's Manual.
MVME214	VMEmodule Static RAM/ROM with VSB and up to 1Mb RAM capacity. Includes User's Manual.
MVME707	Dual RS-232-C Serial I/O Cable Assembly. Includes User's Manual.

The VMEsystem



A Typical High-Speed MVME130/131XT Microcomputer VMEsystem

The MVME131XT 32-Bit Microcomputer with Cache Accelerator Module conforms to the VMEbus interconnect system which has emerged as a world-wide Standard for 8-/6-/32-bit microcomputers. It is part of the VMEmodule product family whose M68000-based processing power, Eurocard mechanical format, powerful VERSAdos real-time software support and international multiple sourcing have made it one of the most popular product lines in the industry.

The VMEsystem architecture, and a board-level component complement that relate to the Cache Accelerator, are illustrated in the above block diagram. Of particular significance are:

- The well-established functional versatility of the VMEbus whose non-proprietary technical specification is widely used for system implementation
- An auxiliary VSB for high-speed memory expansion
- An MC68020-based, 32-bit CPU board (MVME131) utilizing the latest advances in microprocessor design
- A large repertoire of peripheral (controller) modules and memory boards

The unrivalled acceptance of the VMEsystem philosophy has yielded a selection of compatible board-level products from well over a hundred vendors, to offer systems manufacturers the basic tools with which to implement their designs quickly, efficiently, reliably and cost effectively.

MVME133
MVME133-1

VMEmodule™ 32-Bit Monoboard Microcomputer

- MC68020 Microprocessor with 32-Bit Address and Data at 12.5 MHz or 16.67 MHz
- MC68881 Floating Point Coprocessor at 12.5 MHz or 16.67 MHz
- 1Mb of Shared Local DRAM Accessible from the VMEbus (32-bits wide)
- Four 28-pin JEDEC Sockets for ROM/PROM/EPROM/EEPROM (two banks/16-bits wide)
- One Front Panel Serial Debug Port
- One RS-232-C Multiprotocol Serial Port
- One RS-485/RS-422 Multiprotocol Serial Port
- Three 8-Bit Timers
- Real-Time Clock
- A24/D32 VMEbus Master Interface
- VMEbus System Controller Functions with Level-three Arbiter
- Single Level Bus Requester (level jumper selectable)
- Level Three VMEbus Interrupter
- Seven Level Interrupt Handler

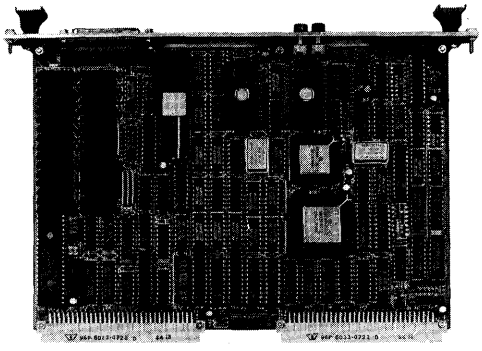
The MVME133 VMEmodule 32-bit Monoboard Microcomputer is designed to function in those applications requiring medium performance and functionality. With its 1Mb of onboard DRAM and MC68881 Floating Point Coprocessor, the MVME133 is an excellent choice for applications requiring real-time operation and arithmetic capability, such as industrial automation and robotics.

CPU

The MVME133 and MVME133-1 use an MC68020 Microprocessor operating at a fixed speed of 12.5 MHz and 16.67 MHz, respectively. The MC68020 is the first product within the popular MC68000 family to offer external 32-bit address and data paths. With its higher clock rate, advanced architecture, enhanced addressing modes and on-chip instruction cache, this product offers state-of-the-art performance while maintaining software compatibility with its widely accepted predecessors.

DEBUG MONITOR FIRMWARE

The MVME133bug Debug Monitor firmware package is optionally available for use with the MVME133 module. This firmware offers 32 debug, up/downline load and disk bootstrap load commands, as well as a full set of onboard diagnostics and a one line assembler/disassembler with full MC68881 support.



COPROCESSOR

The MVME133 module is equipped with the MC68881 Floating Point Coprocessor which substantially improves computing speed in applications requiring arithmetic computations. The MC68881 is a full implementation of the IEEE Standard (P754) for Binary Floating-Point Arithmetic, providing the following operations: add, subtract, multiply, divide, remainder, square root, integer part and compare. Each of these operations supports seven different data formats: byte, word and long word integers; single, double and extended precision real numbers; and packed binary coded decimal string real numbers. In addition, the MC68881 supports a full set of trigonometric and transcendental functions.

Figure 1 is a block diagram of the MVME133.

ONBOARD DYNAMIC RAM

The onboard dynamic ram uses 32-256K x 1 dynamic RAM ZIPS (zigzag-in-line package), making a total of 1Mb of local DRAM. It is accessible by the MC68020, the refresh circuitry, and the VMEbus each of which requests and is granted the DRAM by a multiport arbiter.

The user can select the address at which the onboard DRAM appears from the VMEbus by changing a jumper header or by reprogramming a PAL (Programmable Array Logic). As shipped from the factory, the four address spaces which can be selected by the jumper header are:

$000000–$0FFFFF
$100000–$1FFFFF
$200000–$2FFFFF
$300000–$3FFFFF

Onboard DRAM responds to the VMEbus only when address modifier lines AM0–AM5 indicate standard, priv-

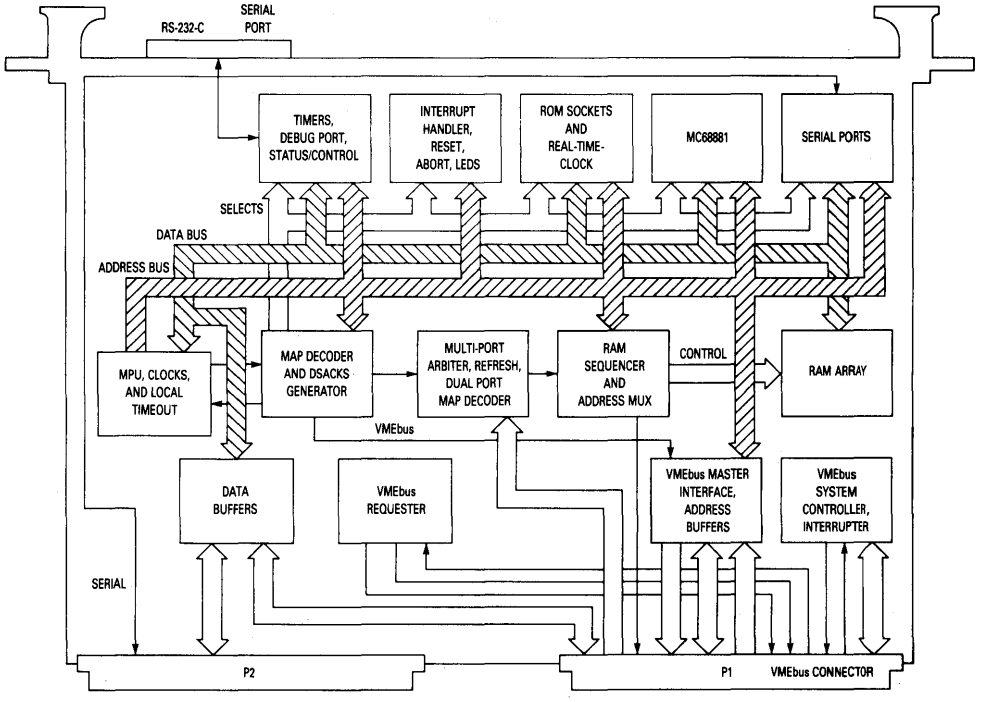


Figure 1. MVME133 Functional Block Diagram

ileged, or non-privileged data or program space, and when the MVME133 on which it resides is not the VMEbus master.

FUNCTION CODE MAP DECODER

At the beginning of each MPU cycle, the space decoder determines what kind of cycle is taking place and which device or function is selected within that cycle type. The cycle types, shown in Table 1, are determined by the function code signal lines which are driven by the MC68020.

MEMORY MAP

The memory map (refer to Table 2) indicates those devices that occupy user data, user program, supervisor data and supervisory program spaces. The MVME133 supports only 24-bit addressing so the two most significant hex digits in the address map are ignored.

Table 1. Function Code Space Decoder

FC2	FC1	FC0	Cycle Type	Responding MVME133 Devices/Functions
0	0	0	Reserved	None except Dynamic RAM
0	0	1	User Data	All except Interrupt Handler and MC68881
0	1	0	User Program	All except Interrupt Handler and MC68881
0	1	1	Reserved	None (Causes Local Timeout)
1	0	0	Reserved	None except Dynamic RAM
1	0	1	Supervisory Data	All except Interrupt Handler and MC68881
1	1	0	Supervisory Program	All except Interrupt Handler and MC68881
1	1	1	CPU (IACK)	VMEbus, Z8530, MC68901, Interrupt Handler
1	1	1	CPU Coprocessor	MC68881

Table 2. Memory Map

Address Range	D31 D24	D23 D16	D15 D08	D07 D00
xx000000 xx000007	Onboard ROM for first four memory cycles after reset. Onboard Dynamic RAM thereafter.			
xx000008 xx0FFFFF	Onboard Dynamic RAM			
xx100000 xxEFFFFF	VMEbus			
xxF00000 xxF1FFFF	Onboard ROM/PROM/EPROM/EEPROM Bank 1 **			
xxF20000 xxF3FFFF	Onboard ROM/PROM/EPROM/EEPROM Bank 2 **			
xxF40000 xxF7FFFF	Onboard ROM/PROM/EPROM/EEPROM Banks 1 and 2 Repeat in this space			
xxF80000	MSR *	MFP GPIP	MSR *	MFP AER
xxF80004	MSR *	MFP DDR	MSR *	MFP IERA
xxF80008	MSR *	MFP IERB	MSR *	MFP IPRA
xxF8000C	MSR *	MFP IPRB	MSR *	MFP ISRA
xxF80010	MSR *	MFP ISRB	MSR *	MFP IMRA
xxF80014	MSR *	MFP IMRB	MSR *	MFP VR
xxF80018	MSR *	MFP TACR	MSR *	MFP TBCR
xxF8001C	MSR *	MFP TCDCR	MSR *	MFP TADR
xxF80020	MSR *	MFP TBDR	MSR *	MFP TCDR
xxF80024	MSR *	MFP TDDR	MSR *	MFP SCR
xxF80028	MSR *	MFP UCR	MSR *	MFP RSR
xxF8002C	MSR *	MFP TSR	MSR *	MFP UDR
xxF80030 xxF9FFFF	The Status and MFP registers appear repeatedly in this space.			
xxFA0000	SIOB RRO SIOB WRO	SIOB RxData SIOB TxData	SIOA RRO SIOA WRO	SIOA RxData SIOA TxData
xxFA0004 xxFAFFFF	The above SIO registers appear repeatedly in this space.			

xxFB0000	UUUU	RTC00	UUUU	RTC01	UUUU	RTC02	UUUU	RTC03
xxFB0004	UUUU	RTC04	UUUU	RTC05	UUUU	RTC06	UUUU	RTC07
xxFB0008	UUUU	RTC08	UUUU	RTC09	UUUU	RTC10	UUUU	RTC11
xxFB000C	UUUU	RTC12	UUUU	RTC13	UUUU	RTC14	UUUU	RTC15

xxFB0010 xxFBFFFF	The above Real-Time Clock registers appear repeatedly in this space.			
xxFC0000 xxFEFFFF	VMEbus			
xxFF0000 xxFFFFFF	VMEbus Short I/O space			

NOTES:

xx Denotes don't care. However, when A24 is 0, the VMEbus is treated as a 32-bit data port for longword aligned transfers and when A24 is 1, the VMEbus is treated as a 16-bit port for all transfers. Reading the Real-Time Clock with A25 = 1 causes the VMEbus Interrupter to assert a level 3 interrupt, while reading it with A25 = 0 has no effect on the Interrupter.

* The Module Status Register (MSR) is read only. It should not be written to. Cycles that access the Status Register also access the Multi-Function Periphery MFP. The Status Register is connected to D24–D31 and the MFP is connected to D16–D23.

** Writes to the EEPROM must always be 16-bit wide.

UUUU Denotes undefined bits.

RTC Denotes Real-Time Clock.

COPROCESSOR REGISTER MAP

The MVME133 devices and functional blocks respond to only two types of CPU space accesses (FC2–FC0 = 111): coprocessor and interrupt acknowledge. The function code space decoder selects the MC68881 anytime the MPU executes a coprocessor cycle (FC2–FC0 = 111 and A19–A16 = 0010). The recommended coprocessor ID (bits 9–11 of the coprocessor instruction word) for the MC68881 is 001. However, the MVME133 selects the MC68881 regardless of the ID. The MC68881 registers are addressed by A04–A01 as shown in Table 3.

Table 3. Coprocessor Register Map

A04–A00 (In Binary)	MC68881 Register D31	D16	D15	D00
00000	Response	(Read Only)	Control	(Write Only)
00100	Save	(Read Only)	Restore	(Read/Write)
01000	Reserved		Command	(Write Only)
01100	Reserved		Condition	(Write Only)
10000	Operand		(Read/Write)	
10100	Register Select (Read Only)		Reserved	
11000	Instruction Address		(Write)	
11000	Operand Address		(Read/Write)	

INTERRUPT HANDLER AND INTERRUPT ACKNOWLEDGE MAP

The interrupt handler gives the onboard MPU the ability to sense and respond to all onboard interrupts, all seven VMEbus interrupts, VMEbus ACFAIL*, VMEbus SYSFAIL* and the ABORT switch. All of the VMEbus interrupts and the ABORT interrupt can be enabled/disabled by jumpers. All other interrupts are enabled/disabled individually under software control. Also all interrupts may be disabled with a software control bit.

Function code space decoder selects the interrupt handler anytime the MC68020 executes an Interrupt Acknowledge cycle (FC2–FC0 = 111 and A19–A16 = 1111). The interrupt handler then determines the acknowledge level by examining A01–A03. It then decides which device the interrupt acknowledge cycle is for. If the acknowledge cycle was for VMEbus ACFAIL*, ABORT switch or real-time clock, the interrupt handler causes the MC68020 to generate the interrupt vector internally. If not, the interrupt handler initiates a vector fetch cycle for the appropriate device. If both onboard and VMEbus interrupts are activated on the same interrupt level, the interrupt handler will acknowledge the onboard interrupt first.

Table 4 summarizes all the interrupt sources and their associated interrupt vectors.

Table 4. Interrupt Sources and Vectors

Interrupt Source	Path	Vector Passed	Vector Offset	Level
VMEbus IRQ1*	Direct	From Interrupting VMEbus Slave	4 x Vector	1
VMEbus IRQ2*	Direct	Same As Above	4 x Vector	2
VMEbus IRQ3*	Direct	Same As Above	4 x Vector	3
VMEbus IRQ4*	Direct	Same As Above	4 x Vector	4
MM58274A Real-Time Clock	Direct	None	$70	4
VMEbus IRQ5*	Direct	From Interrupting VMEbus Slave	4 x Vector	5
DDTR*	MC68901 GPIO0	Refer to MC68901 Data Sheet	4 x Vector	5
VMEbus BERR*	MC68901 GPIO1	Refer to MC68901 Data Sheet	4 x Vector	5
LTO Local Bus Timeout	MC68901 GPIO2	Refer to MC68901 Data Sheet	4 x Vector	5
MC68901 Timer D	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
MC68901 Timer C	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5

(continued)

Table 4. Interrupt Sources and Vectors (continued)

Interrupt Source	Path	Vector Passed	Vector Offset	Level
MC68901 Timer B	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
Transmit Error (Serial)	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
Transmit Buffer Empty (Serial)	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
Receive Error (Serial)	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
Receive Buffer Full (Serial)	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
MC68901 Timer A	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
OIRQ (Bus Interrupter)	MC68901 GPIO6	Refer to MC68901 Data Sheet	4 x Vector	5
VMEbus SYSFAIL	MC68901 GPIO7	Refer to MC68901 Data Sheet	4 x Vector	5
VMEbus IRQ6*	Direct	From Interrupting VMEbus Slave	4 x Vector	6
Z8530 Serial Ports	Direct	Refer to Z8530 Data Sheet	4 x Vector	6
VMEbus IRQ7*	Direct	From Interrupting VMEbus Slave	4 x Vector	7
VMEbus ACFAIL	Direct	None	$7C	7
ABORT*	Direct	None	$7C	7

MULTIFUNCTION PERIPHERAL

The MVME133 uses the multifunction peripheral (MC68901) for its front panel debug port, tick timers, watchdog timer and the status and control registers. The MC68901 has the ability to interrupt the MPU on level 5 when required for the timers, debug port and GPIO (status) bits.

MC68901 DEBUG PORT

The front panel debug port is a minimal implementation of an RS-232-C serial port. It uses RXD as its transmit data output and TXD as its receive data input. DSR is driven true and DCD is pulled up to +12 V. CTS is controlled by a software bit and RTS is monitored by another software bit, providing minimal flow control.

The baud rate generator for the serial port is timer C in the MC68901. The XTAL input to the MC68901 is 1.23 MHz, supporting baud rates from 110–9600 baud.

MC68901 TIMERS

There are four timers in the MC68901. They are assigned as follows:

Timer A — Software tick timer.
Timer B — Tick timer overflow/watchdog timeout.
Timer C — Baud rate generator for the front panel serial debug port.
Timer D — Delay mode only. Unassigned.

NOTE: The watchdog timeout resets the MPU module when the timer B output is high.

MC68901 GENERAL PURPOSE I/O

The MC68901 has eight General Purpose I/O pins. The MVME133 uses five of these pins as status inputs and three of them as control outputs. The assignment of GPIO0–GPIO7 is shown in Table 5.

MODULE STATUS REGISTER

In addition to the general purpose status and control bits of the MC68901, the MVME133 has eight read only status bits. Collectively, these bits are called the module status register (MSR). The MSR and the MC68901 are grouped together and appear as a 16-bit word port to the MPU. Therefore, it is important to note that even though the MSR ignores all write accesses, a write to the MSR will affect the MC68901.

The bit assignments for the module status register are shown in Table 6.

Z8530 DUAL SERIAL PORTS

The MVME133 uses the Z8530 to implement its two multiprotocol serial ports, providing multifunction support for handling the large variety of serial communications protocols available. The Z8530 can be programmed to satisfy special serial communications requirements as well as to follow standard formats such as byte-oriented synchronous, bit-oriented synchronous and asynchronous. In addition, protocol variations are supported within each operating mode by checking odd or even parity, character insertion or deletion, CRC generation and

Table 5. GPIO Register

GPIO7 INPUT	GPIO6 INPUT	GPIO5 OUTPUT	GPIO4 OUTPUT	GPIO3 OUTPUT	GPIO2 INPUT	GPIO1 INPUT	GPIO0 INPUT

GPIO7 — VMEbus SYSFAIL Flag
GPIO6 — VMEbus IRQ3 Request Flag
GPIO5 — Board or System Failure Control
GPIO4 — Enable/Disable Interrupts Control
GPIO3 — DCTS Control
GPIO2 — Local Bus Timeout Flag
GPIO1 — VMEbus BERR Flag
GPIO0 — DRTS Flag

Table 6. Module Status Register

BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
ACFAIL	SYSCON	PWRUP*	SRBIT4	SRBIT3	SRBIT2	SRBIT1	SRBIT0

ACFAIL — VMEbus AC Power Fail
SYSCON — VMEbus System Controller
PWRUP* — Power-Up Reset
SRBIT4 — Status Register Bit 4
SRBIT3 — Status Register Bit 3
SRBIT2 — Status Register Bit 2
SRBIT1 — Status Register Bit 1
SRBIT0 — Status Register Bit 0

checking, break and abort generation and detection and many other protocol dependent features.

Because of the internal structure of the Z8530, there are several means of obtaining the baud rate clocks for each of the two serial channels. Many of the more frequently used baud rates are supported by the MVME133 internally, with additional support provided for external generation of any other desired frequency. Reference should be made to the Z8530 Data Manual and the MVME133 User's Manual for additional information.

All drivers and receivers for both serial channels are provided on the MVME133, with port A implemented for RS-485/RS-422 and port B for RS-232 communications. User can configure the RS-232-C port (Port B) as DCE or as DTE using the jumpers provided by MVME133. The RS-485 port may be configured by software to be Master or Slave, full duplex or half duplex. All signals are made available on rows A and C of the VMEbus P2 connector. The RS-232 connections can be made directly from P2 to a DB-25 connector, although the RS-485 may require a crossover cable. Refer to the MVME133 User's Manual for pinout information.

REAL-TIME CLOCK

The Real-Time clock on the MVME133 is an MM58274, providing a timekeeping function from tenths of seconds to tens of years in independently accessible registers, an hours counter programmable for 12- or 24-hour operation, an independent interrupting timer and its own on-board crystal controller oscillator. The counters are arranged as 4-bit words and can be randomly accessed for reading and setting time.

ONBOARD ROM/PROM/EPROM/EEPROM

The MVME133 has four 28-pin ROM/PROM/EPROM/EEPROM sockets that are organized as two banks with two sockets per bank. Each bank appears as a 16-bit word port to the MPU and can be separately configured for 8K x 8, 16K x 8, 32K x 8 or 64K x 8 ROM/PROM/EPROMs or for 2K x 8, 8K x 8 or 32K x 8 EEPROMs. When a bank is configured for EEPROM, writes to that bank must always be 16-bits wide.

There are several different algorithms for erasing/writing to EEPROMs, depending on the manufacturer. The MVME133 supports only those devices which have a "Static RAM" compatible erase/write mechanism. Note that the MVME133 requires that the EEPROMs must allow wired-OR on the RDY/BSY* pin.

VMEbus INTERFACE

The MVME133 provides an A24/D32 VMEbus interface for buffering of data, address and control, for word data manipulation to accommodate MC68020 and VMEbus data handling differences and for interrupt handling and control of misaligned transfers. In addition, system controller functions are available including a VMEbus time-out generator, a SYSCLK driver, an IACK* daisy-chain driver, a SYSRESET* driver and a level three arbiter. All of these system controller functions are enabled/disabled by a jumper on the MVME133.

VMEbus REQUESTER

The VMEbus requester is used to obtain and relinquish mastership of the VMEbus. The MVME133 operates in the Early Release-on-Request (ROR) mode and can request VMEbus mastership on any one of the four request levels, depending upon jumper configurations on the MVME133.

VMEbus INTERRUPTER

The MVME133's single-level Interrupter generates interrupt requests on IRQ3*. It provides the value $FF as its status ID byte. It is an 8-bit interrupter and responds to all sizes of interrupt acknowledge cycles. The interrupter drives IRQ3* whenever the MPU reads the real-time clock with A25 = 1. The state of the interrupter is reflected as the OIRQ (GPIO6) bit of the MFP GPIO port.

Table 7. MVME133 Timing
(All Times Are Total Cycles)

Access Sequence	MVME133		MVME133-1		Notes
	Read	Write	Read	Write	
MPU to local DRAM	4 Cycles	4 Cycles	4 Cycles	4 Cycles	1, 2
MPU to local ROM/ PROM/EPROM/EEPROM	6 Cycles	6 Cycles	6 Cycles	6 Cycles	1, 3
VMEbus to local DRAM	9 Cycles (720 ns)	9 Cycles (720 ns)	9 Cycles (540 ns)	9 Cycles (540 ns)	4, 5
MPU to global RAM (MVME133)	13 Cycles	14 Cycles	13 Cycles	14 Cycles	5, 6
MPU to global RAM (MVME204-1)	9 Cycles	7 Cycles	11 Cycles	8 Cycles	6, 7

NOTES: 1. No arbitration overhead.
2. Except RMW cycles where MVME133 is required to obtain VMEbus mastership before RAM cycle can be started.
3. Device access time must be 300 ns or less for MVME133 and 250 ns or less for MVME133-1.
4. DS0*/DS1* asserted to DTACK* time.
5. Typical values. Actual values may be greater or less depending on the state of the slave MVME133,-1.
6. Assume the master MVME133-1 is the current VMEbus master.
7. The total number of clock cycles = 4 + (Ta/T) for a READ and 5 + (Ta/T) for a WRITE, where Ta = DS0*/DS1* to DTACK* time, and T = MPU clock period.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements MVME133 MVME133-1	 +5 Vdc, 5 A max, 4 A typ ±12 Vdc, 250 mA max +5 Vdc, 6 A max, 4.5 A typ ±12 Vdc, 250 mA max
Clock Frequency MVME133 MVME133-1	 12.5 MHz clock frequency 16.7 MHz clock frequency
Operating Temperature	0° to 55°C Inlet air temperature with forced air cooling
Storage Temperature	−40° to 85°C
Relative Humidity	5% to 90% (non-condensing)
Physical Dimensions Height Width	 10.31 in (261.8 mm) 7.40 in (188.0 mm)
Thickness	0.83 in (21.0 mm)

ORDERING INFORMATION

Part Number	Description
MVME133	VMEmodule 32-bit Monoboard Microcomputer with 12.5 MHz MC68020 CPU. Includes User's Manual.
MVME133-1	VMEmodule 32-bit Monoboard Microcomputer with 16.7 MHz MC68020 CPU. Includes User's Manual.
MVME133/D	User's Manual for MVME133/MVME133-1.

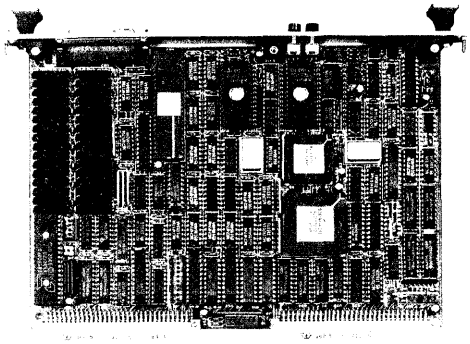
RELATED PRODUCTS

Part Number	Description
MVME133bug	MVME133bug Diagnostic/Debug Package for the MVME133/ MVME133-1. Includes User's Manual.
MVME225	VMEmodule 2Mb Dynamic RAM with Byte Parity. Includes User's Manual.
MVME215-1/-2/-3	256Kb/512Kb/1Mb CMOS RAM Module. Includes User's Manual.

VMEmodule 32-Bit Monoboard Microcomputer

- MC68020 Microprocessor with 32-Bit Address and Data at 20 MHz
- 1Mb of Shared Local DRAM Accessible from the VMEbus (32-bits wide)
- Four 28-pin JEDEC Sockets for ROM/PROM/EPROM/EEPROM (two banks/16-bits wide)
- One Front Panel Serial Debug Port
- One RS-232-C Multiprotocol Serial Port
- One RS-485/RS-422 Multiprotocol Serial Port
- Three 8-Bit Timers
- Real-Time Clock
- A32/D32 VMEbus Master Interface
- VMEbus System Controller Functions with Level-three Arbiter
- Single Level Bus Requester (level jumper selectable)
- Level Three VMEbus Interrupter
- Seven Level Interrupt Handler

The MVME133A VMEmodule 32-bit Monoboard Microcomputer is designed to function in those applications requiring medium performance and functionality. With its 1Mb of onboard DRAM, the MVME133A is an excellent choice for applications requiring real-time operation such as industrial automation and robotics.



CPU

The MVME133A uses an MC68020 Microprocessor operating at a fixed speed of 20.0 MHz. The MC68020 is the first product within the popular M68000 Family to offer external 32-bit address and data paths. With its higher clock rate, advanced architecture, enhanced addressing modes and on-chip instruction cache, this product offers state-of-the-art performance while maintaining software compatibility with its widely accepted predecessors.

DEBUG MONITOR FIRMWARE

The MVME133Abug Debug Monitor firmware package is optionally available for use with the MVME133A module. This firmware offers 32 debug, up/downline load and disk bootstrap load commands, as well as a full set of onboard diagnostics and a one line assembler/disassembler.

COPROCESSOR

The MVME133A is equipped with a 20.0 MHz MC68881 Floating-Point Coprocessor which will substantially improve computing speed in applications requiring arithmetic computations. The MC68881 is a full implementation of the IEEE Standard (P754) for Binary Floating-Point Arithmetic, providing the following operations: add, subtract, multiply, divide, remainder, square root, integer part and compare. Each of these operations supports seven different data formats: byte, word and long word integers; single, double and extended precision real numbers; and packed binary coded decimal string real numbers. In addition, the MC68881 supports a full set of trigonometric and transcendental functions.

Figure 1 is a block diagram of the MVME133A.

ONBOARD DYNAMIC RAM

The onboard dynamic ram uses 32-256Kx1 dynamic RAM ZIPs (zigzag-in-line package), making a total of 1Mb of local DRAM. It is accessible by the MC68020, the refresh circuitry, and the VMEbus each of which requests and is granted the DRAM by a multiport arbiter.

The onboard shared DRAM address is controlled by U82 and by J2. U82 selects one 16-Mbyte block within the 4 gigabytes range for the MVME133A. The default factory program for U82 puts the base address of this 16-Mbyte block at $00000000. J2, then, selects one of the 16 address spaces within this 16-Mbyte block for the 1 Mbytes of onboard shared DRAM.

Furthermore, the MVME133A allows the user to select 32-bit or 24-bit address option for VMEbus references. By properly jumpering J18, the user can configure the MVME133A to operate in a mixed 32-bit and 24-bit address system or a fully 32-bit address system. The MVME133A VMEbus memory map is directly affected by the address option selected as follows:

If MVME133A is in a mixed 32-bit and 24-bit address system:

00000000-000FFFFF	No VMEbus activity, onboard DRAM area.
00100000-00EFFFFF	VMEbus STANDARD (24-bit) addresss space.
00F00000-FFEFFFFF	VMEbus EXTENDED (32-bit) address space.
FFF00000-FFFBFFFF	No VMEbus activity, local resource area.
FFFC0000-FFFEFFFF	VMEbus EXTENDED (32-bit) address space.
FFFF0000-FFFFFFFF	VMEbus SHORT I/O (16-bit) address space.

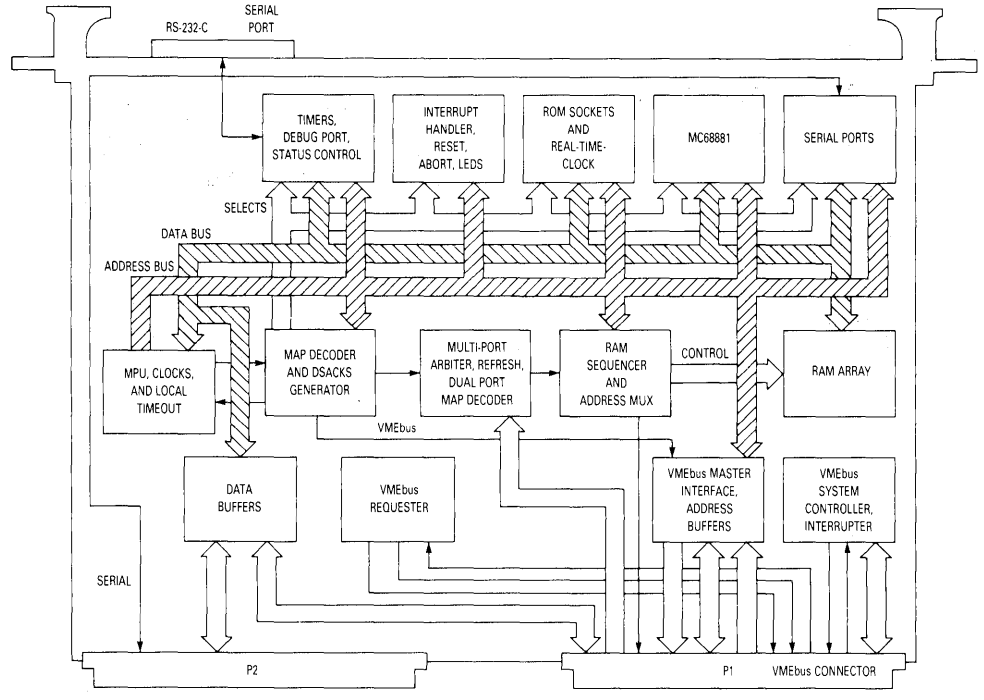


Figure 1. MVME133A Functional Block Diagram

If MVME133A is in a fully 32-bit address system:

00000000-000FFFFF	No VMEbus activity, onboard DRAM area.
00100000-FFEFFFFF	VMEbus EXTENDED (32-bit) address space.
FFF00000-FFFBFFFF	No VMEbus activity, local resouce area.
FFC00000-FFFEFFFF	VMEbus EXTENDED (24-bit) address space.
FFFF0000-FFFFFFFF	VMEbus SHORT I/O (16-bit) address space.

The MVME133A's DRAM responds to the VMEbus accesses only when the addresses match and the Address Modifiers indicate privileged or non-privileged, data or program space, and when this board is not the current VMEbus master.

FUNCTION CODE MAP DECODER

At the beginning of each MPU cycle, the space decoder determines what kind of cycle is taking place and which device or function is selected within that cycle type. The cycle types, shown in Table 1, are determined by the function code signal lines which are driven by the MC68020.

Table 1. Function Code Space Decoder

FC2	FC1	FC0	Cycle Type	Responding MVME133A Devices/Functions
0	0	0	Reserved	None (Causes Local Timeout)
0	0	1	User Data	All except Interrupt Handler and MC68881
0	1	0	User Program	All except Interrupt Handler and MC68881
0	1	1	Reserved	None (Causes Local Timeout)
1	0	0	Reserved	None (Causes Local Timeout)
1	0	1	Supervisory Data	All except Interrupt Handler and MC68881
1	1	0	Supervisory Program	All except Interrupt Handler and MC68881
1	1	1	CPU (IACK)	VMEbus, Z8530, MC68901, Interrupt Handler
1	1	1	CPU Coprocessor	MC68881

MEMORY MAP

The memory map (refer to Table 2) indicates those devices that occupy user data, user program, supervisor data and supervisory program spaces.

Table 2. Memory Map

Address Range	D31 D24	D23 D16	D15 D08	D07 D00
00000000 00000007	Onboard ROM for first four memory cycles after reset. Onboard Dynamic RAM thereafter.			
00000008 000FFFFF	Onboard Dynamic RAM			
00100000 FFEFFFFF	VMEbus			
FFF00000 FFF1FFFF	Onboard ROM/PROM/EPROM/EEPROM Bank 1 **			
FFF20000 FFF3FFFF	Onboard ROM/PROM/EPROM/EEPROM Bank 2 **			
FFF40000 FFF7FFFF	Onboard ROM/PROM/EPROM/EEPROM Banks 1 and 2 Repeat in this space			
FFF80000	STATUS *	MFP GPIP	STATUS *	MFP AER
FFF80004	STATUS *	MFP DDR	STATUS *	MFP IERA
FFF80008	STATUS *	MFP IERB	STATUS *	MFP IPRA
FFF8000C	STATUS *	MFP IPRB	STATUS *	MFP ISRA
FFF80010	STATUS *	MFP ISRB	STATUS *	MFP IMRA
FFF80014	STATUS *	MFP IMRB	STATUS *	MFP VR
FFF80018	STATUS *	MFP TACR	STATUS *	MFP TBCR
FFF8001C	STATUS *	MFP TCDCR	STATUS *	MFP TADR
FFF80020	STATUS *	MFP TBDR	STATUS *	MFP TCDR
FFF80024	STATUS *	MFP TDDR	STATUS *	MFP SCR
FFF80028	STATUS *	MFP UCR	STATUS *	MFP RSR
FFF8002C	STATUS *	MFP TSR	STATUS *	MFP UDR
FFF80030 FFF9FFFF	The status and MFP registers occur repeatedly in this space.			
FFFA0000	SIOB RRO SIOB WRO	SIOB RxData SIOB TxData	SIOA RRO SIOA WRO	SIOA RxData SIOA TxData
FFFA0004 FFFAFFFF	The above SIO registers appear repeatedly in this space.			
FFFB0000	UUUU RTC00	UUUU RTC01	UUUU RTC02	UUUU RTC03
FFFB0004	UUUU RTC04	UUUU RTC05	UUUU RTC06	UUUU RTC07
FFFB0008	UUUU RTC08	UUUU RTC09	UUUU RTC10	UUUU RTC11
FFFB000C	UUUU RTC12	UUUU RTC13	UUUU RTC14	UUUU RTC15
FFFB0010 FFFB7FFF	The above real-time clock registers appear repeatedly in this space.			
FFFB8000 FFFBFFFF	VMEbus Interrupt			
FFFC0000 FFFEFFFF	VMEbus			
FFFF0000 FFFFFFFF	VMEbus Short I/O Space			

NOTES: * The status register is read only. It should not be written to. Cycles that access the status register also access the MFP. The status register is connected to [D24] - [D31] and the MFP is connected to [D16] - [D23].

** Writes to the EEPROM must always be 16-bit wide.

UUUU Denotes undefined bits.

COPROCESSOR REGISTER MAP

The MVME133A responds to only two types of CPU space accesses (FC2 – FC0 = 111): coprocessor and interrupt acknowledge. The function code space decoder selects the MC68881 anytime the MPU executes a coprocessor cycle (FC2 – FC0 = 111 and A19 – A16 = 0010). The recommended coprocessor ID (bits 9-11 of the coprocessor instruction word) for the MC68881 is 001. However, the MVME133A selects the MC68881 regardless of the ID. The MC68881 registers are addressed by A04-A01 as shown in Table 3.

Table 3. Coprocessor Register Map

A04–A00 (In Binary)	MC68881 Register D31 – D16		MC68881 Register D15 – D00	
00000	Response	(Read Only)	Control	(Write Only)
00100	Save	(Read Only)	Restore	(Read/Write)
01000	Reserved		Command	(Write Only)
01100	Reserved		Condition	(Write Only)
10000	Operand		(Read/Write)	
10100	Register Select (Read Only)		Reserved	
11000	Instruction Address		(Write)	
11100	Operand Address		(Read/Write)	

INTERRUPT HANDLER AND INTERRUPT ACKNOWLEDGE MAP

The interrupt handler gives the onboard MPU the ability to sense and respond to all onboard interrupts, all seven VMEbus interrupts, VMEbus ACFAIL*, VMEbus SYSFAIL* and the ABORT switch. All of the VMEbus interrupts and the ABORT interrupt can be enabled/disabled by jumpers. All other interrupts are enabled/disabled individually under software control. Also all interrupts may be disabled with a software control bit.

Function code space decoder selects the interrupt handler anytime the MC68020 executes an Interrupt Acknowledge cycle (FC2 – FC0 = 111 and A19 – A16 = 1111). The interrupt handler then determines the acknowledge level by examining A01-A03. It then decides which device the interrupt acknowledge cycle is for. If the acknowledge cycle was for VMEbus ACFAIL*, ABORT switch or real-time clock, the interrupt handler causes the MC68020 to generate the interrupt vector internally. If not, the interrupt handler initiates a vector fetch cycle for the appropriate device. If both onboard and VMEbus interrupts are activated on the same interrupt level, the interrupt handler will acknowledge the onboard interrupt first.

Table 4 summarizes all the interrupt sources and their associated interrupt vectors.

Table 4. Interrupt Sources and Vectors

Interrupt Source	Path	Vector Passed	Vector Offset	Level
VMEbus IRQ1*	Direct	From Interrupting VMEbus Slave	4 x Vector	1
VMEbus IRQ2*	Direct	Same As Above	4 x Vector	2
VMEbus IRQ3*	Direct	Same As Above	4 x Vector	3
VMEbus IRQ4*	Direct	Same As Above	4 x Vector	4
MM58274A Real-Time Clock	Direct	None	$70	4
VMEbus IRQ5*	Direct	From Interrupting VMEbus Slave	4 x Vector	5
DDTR*	MC68901 GPIO0	Refer to MC68901 Data Sheet	4 x Vector	5
VMEbus BERR*	MC68901 GPIO1	Refer to MC68901 Data Sheet	4 x Vector	5
LTO Local Bus Timeout	MC68901 GPIO2	Refer to MC68901 Data Sheet	4 x Vector	5
MC68901 Timer D	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
MC68901 Timer C	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5

(continued)

Table 4. Interrupt Sources and Vectors (continued)

Interrupt Source	Path	Vector Passed	Vector Offset	Level
MC68901 Timer B	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
Transmit Error (Serial)	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
Transmit Buffer Empty (Serial)	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
Receive Error (Serial)	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
Receive Buffer Full (Serial)	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
MC68901 Timer A	MC68901	Refer to MC68901 Data Sheet	4 x Vector	5
OIRQ (Bus Interrupter)	MC68901 GPIO6	Refer to MC68901 Data Sheet	4 x Vector	5
VMEbus SYSFAIL	MC68901 GPIO7	Refer to MC68901 Data Sheet	4 x Vector	5
VMEbus IRQ6*	Direct	From Interrupting VMEbus Slave	4 x Vector	6
Z8530 Serial Ports	Direct	Refer to Z8530 Data Sheet	4 x Vector	6
VMEbus IRQ7*	Direct	From Interrupting VMEbus Slave	4 x Vector	7
VMEbus ACFAIL	Direct	None	$7C	7
ABORT*	Direct	None	$7C	7

MULTIFUNCTION PERIPHERAL

The MVME133A uses the multifunction peripheral (MC68901) for its front panel debug port, tick timers, watchdog timer and the status and control registers. The MC68901 has the ability to interrupt the MPU on level 5 when required for the timers, debug port and GPIO (status) bits.

MC68901 DEBUG PORT

The front panel debug port is a minimal implementation of an RS-232-C serial port. It uses RXD as its transmit data output and TXD as its receive data input. DSR is driven true and DCD is pulled up to +12 V. CTS is controlled by a software bit and RTS is monitored by another software bit, providing minimal flow control.

The baud rate generator for the serial port is timer C in the MC68901. The XTAL input to the MC68901 is 1.23 MHz, supporting baud rates from 110-9600 baud.

MC68901 TIMERS

There are four timers in the MC68901. They are assigned as follows:

Timer A — Software tick timer.
Timer B — Tick timer overflow/watchdog timeout.
Timer C — Baud rate generator for the front panel serial debug port.
Timer D — Delay mode only. Unassigned.

NOTE: The watchdog timeout resets the MPU module when the timer B output is high.

MC68901 GENERAL PURPOSE I/O

The MC68901 has eight General Purpose I/O pins. The MVME133A uses five of these pins as status inputs and three of them as control outputs. The assignment of GPIO0-GPIO7 is shown in Table 5.

MODULE STATUS REGISTER

In addition to the general purpose status and control bits of the MC68901, the MVME133A has eight read only status bits. Collectively, these bits are called the module status register (MSR). The MSR and the MC68901 are grouped together and appear as a 16-bit word port to the MPU. Therefore, it is important to note that even though the MSR ignores all write accesses, a write to the MSR will affect the MC68901.

The bit assignments for the module status register are shown in Table 6.

Z8530 DUAL SERIAL PORTS

The MVME133A uses the Z8530 to implement its two multiprotocol serial ports, providing multifunction support for handling the large variety of serial communications protocols available. The Z8530 can be programmed to satisfy special serial communications requirements as well as to follow standard formats such as byte-oriented synchronous, bit-oriented synchronous and asynchronous. In addition, protocol variations are supported within each operating mode by checking odd or even parity, character insertion or deletion, CRC generation and checking, break and abort generation and detection and many other protocol dependent features.

Table 5. GPIO Register

GPIO7 INPUT	GPIO6 INPUT	GPIO5 OUTPUT	GPIO4 OUTPUT	GPIO3 OUTPUT	GPIO2 INPUT	GPIO1 INPUT	GPIO0 INPUT

GPIO7 — VMEbus SYSFAIL Flag
GPIO6 — VMEbus IRQ3 Request Flag
GPIO5 — Board or System Failure Control
GPIO4 — Enable/Disable Interrupts Control
GPIO3 — DCTS Control
GPIO2 — Local Bus Timeout Flag
GPIO1 — VMEbus BERR Flag
GPIO0 — DRTS Flag

Table 6. Module Status Register

BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
ACFAIL	SYSCON	PWRUP*	SRBIT4	SRBIT3	SRBIT2	SRBIT1	SRBIT0

ACFAIL — VMEbus AC Power Fail
SYSCON — VMEbus System Controller
PWRUP* — Power-Up Reset
SRBIT4 — Status Register Bit 4
SRBIT3 — Status Register Bit 3
SRBIT2 — Status Register Bit 2
SRBIT1 — Status Register Bit 1
SRBIT0 — Status Register Bit 0

Because of the internal structure of the Z8530, there are several means of obtaining the baud rate clocks for each of the two serial channels. Many of the more frequently used baud rates are supported by the MVME133A internally, with additional support provided for external generation of any other desired frequency. Reference should be made to the Z8530 Data Manual and the MVME133A User's Manual for additional information.

All drivers and receivers for both serial channels are provided on the MVME133A, with port A implemented for RS-485/RS-422 and port B for RS-232 communications. User can configure the RS-232-C port (Port B) as DCE or as DTE using the jumpers provided by MVME133A. The RS-485 port may be configured by software to be Master or Slave, full duplex or half duplex. All signals are made available on rows A and C of the VMEbus P2 connector. The RS-232 connections can be made directly from P2 to a DB-25 connector, although the RS-485 may require a crossover cable. Refer to the MVME133A User's Manual for pinout information.

REAL-TIME CLOCK

The Real-Time clock on the MVME133A is an MM58274, providing a timekeeping function from tenths of seconds to tens of years in independently accessible registers, an hours counter programmable for 12- or 24-hour operation, an independent interrupting timer and its own onboard crystal controller oscillator. The counters are arranged as 4-bit words and can be randomly accessed for reading and setting time.

ONBOARD ROM/PROM/EPROM/EEPROM

The MVME133A has four 28-pin ROM/PROM/EPROM/EEPROM sockets that are organized as two banks with two sockets per bank. Each bank appears as a 16-bit word port to the MPU and can be separately configured for 8Kx8, 16Kx8, 32Kx8 or 64Kx8 ROM/PROM/EPROMs or for 2Kx8, 8Kx8 or 32Kx8 EEPROMs. When a bank is configured for EEPROM, writes to that bank must always be 16-bits wide.

There are several different algorithms for erasing/writing to EEPROMs, depending on the manufacturer. The MVME133A supports only those devices which have a "Static RAM" compatible erase/write mechanism. Note that the MVME133A requires that the EEPROMs allow wired-OR on the RDY/BSY* pin.

VMEbus INTERFACE

The MVME133A provides an A32/D32 VMEbus interface for buffering of data, address and control, for word data manipulation to accommodate MC68020 and VMEbus data handling differences and for interrupt handling and control of misaligned transfers. In addition, system controller functions are available including a VMEbus time-out generator, a SYSCLK driver, an IACK* daisy-chain driver, a SYSRESET* driver and a level three arbiter. All of these system controller functions are enabled/disabled by a jumper on the MVME133A.

VMEbus REQUESTER

The VMEbus requester is used to obtain and relinquish mastership of the VMEbus. The MVME133A operates in the Early Release-on-Request (ROR) mode and can request VMEbus mastership on any one of the four request levels, depending upon jumper configurations on the MVME133A.

VMEbus INTERRUPTER

The MVME133A's single-level Interrupter generates interrupt requests on IRQ3*. It provides the value $FF as its status ID byte. It is an 8-bit interrupter and responds to all sizes of interrupt acknowledge cycles. The interrupter drives IRQ3* whenever the MPU performs a read access to a location within FFFB8000 to FFFBFFFF. The state of the interrupter is reflected as the OIRQ (GPIO6) bit of the MFP GPIO port.

Table 7. MVME133A Timing
(All Times Are Total Cycles)

Access Sequence	MVME133A		Notes
	Read	Write	
MPU to local DRAM	4 Cycles	4 Cycles	1, 2
MPU to local ROM/ PROM/EPROM/EEPROM	6 Cycles	6 Cycles	1, 3
VMEbus to local DRAM	9 Cycles (450 ns)	9 Cycles (450 ns)	4, 5
MPU to global RAM (MVME133A)	13 Cycles	14 Cycles	5, 6
MPU to global RAM (MVME204-1)	9 Cycles	7 Cycles	6, 7

NOTES: 1. No arbitration overhead.
2. Except RMW cycles where MVME133A is required to obtain VMEbus mastership before RAM cycle can be started.
3. Device access time must be 200 ns or less.
4. DS0*/DS1* asserted to DTACK* time.
5. Typical values. Actual values may be greater or less depending on the state of the slave MVME133A.
6. Assume the master MVME133A is the current VMEbus master.
7. The total number of clock cycles = 4 + (Ta/T) for a READ and 5 + (Ta/T) for a WRITE, where Ta = DS0*/DS1* to DTACK* time, and T = MPU clock period.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements MVME133A-20	+5 Vdc, 7 A max, 5 A typ ±12 Vdc, 250 mA max
Clock Frequency MVME133A-20	20.0 MHz clock frequency
Operating Temperature	0° to 55°C Inlet air temperature with forced air cooling
Storage Temperature	−40° to 85°C
Relative Humidity	5% to 90% (non-condensing)
Physical Dimensions Height Width	10.31 in (261.8 mm) 7.40 in (188.0 mm)
Thickness	0.83 in (21.0 mm)

ORDERING INFORMATION

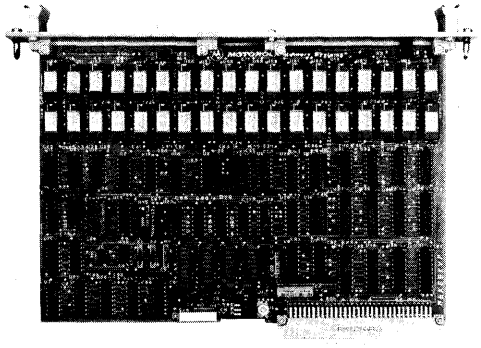
Part Number	Description
MVME133A-20	VMEmodule 32-bit Monoboard Microcomputer with 20.0 MHz MC68020 CPU. Includes User's Manual.
MVME133A/D	VMEmodule 32-Bit Monoboard Microcomputer User's Manual.

RELATED PRODUCTS

Part Number	Description
MVME133Abug	MVME133Abug Diagnostic/Debug Package for the MVME133A. Includes User's Manual.
MVME225	VMEmodule 2Mb Dynamic RAM with Byte Parity. Includes User's Manual.
MVME215-1/-2/-3	256Kb/512Kb/1Mb CMOS RAM Module. Includes User's Manual.
M68V2XSBG133	MVME133Abug Source and Object Modules on 5¼'' Diskette. Includes User's Manual.

VMEmodule 64K/256K Byte Dynamic RAM

- 64K Byte and 256K Byte Versions
- On-Board Refresh Circuitry
- Byte Parity (odd) Generation and Detection
- Byte or Word Addressable
- Jumper Selectable Memory Map Assignment in Four Independent Blocks
- Read/Write Cycle Time 565 ns (max)
- LED Error Display
- Double Eurocard Form Factor
- VMEbus Compatible
- 0° C–70° C Operating Temperature Range



The MVME200 and MVME201 are VMEmodule Dynamic RAM boards used in VMEbus-based systems to increase global memory. Both modules provide four independent blocks of memory which can be located on appropriate boundaries throughout the 16 megabyte MC68000 address space. The modules use an 18-bit row organization to implement both byte and double byte (word) accessing and to implement byte parity generation and detection. Figure 1 is a functional block diagram of the modules.

Both MVME200 and MVME201 are comprised of two 18-device rows of 200 ns dynamic RAM's. The 64K byte (32K words) MVME200 is organized into four 16K-byte blocks. A capacity of 256K bytes (128K words) in 64K-byte blocks is provided by MVME201. The modules have refresh circuitry and perform refresh every 16 μs. While refresh is in progress, any memory access is delayed until refresh completion. The modules have a header for jumper disabling of refresh generation, as a diagnostic aid.

Whether a write or read access is to the low or high order byte of a word is determined, respectively, by use of the VMEbus signals DS0* or DS1*, the lower and upper data strobes. Both data strobes are used to access a full word. Write and read cycle timing parameters are shown in Figures 2 and 3, respectively.

BASE ADDRESS SELECTION

The MVME200 and MVME201 modules each have four headers, one for jumper selection of a base address for each of the four blocks in which the total memory offered by the module is organized. The base address of any MVME200 block can be set on any 16K byte boundary. The base address of any MVME201 block can be set on any 64K byte boundary. Address boundaries throughout the full 16 megabyte MC68000 address space can be chosen for both MVME200 and MVME201.

Unused blocks of memory must be jumper disabled. An additional pin is provided for this in the base address selection header for each block on both modules.

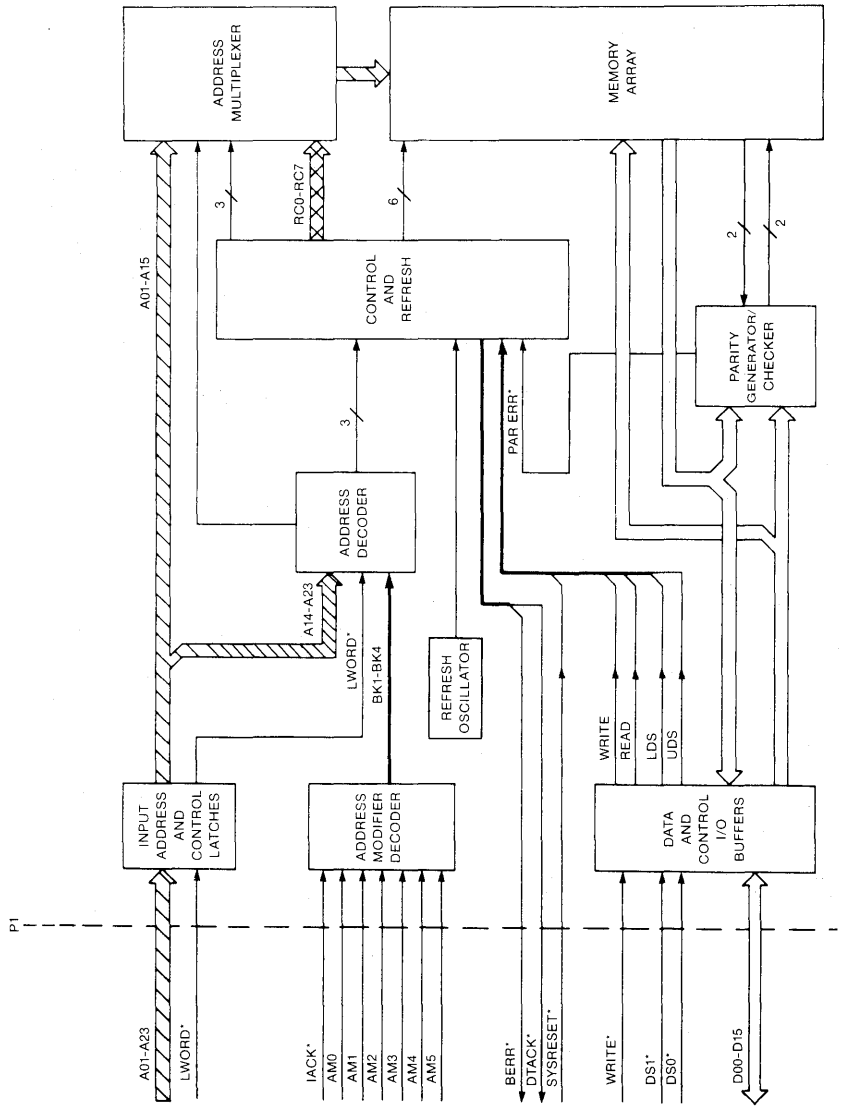
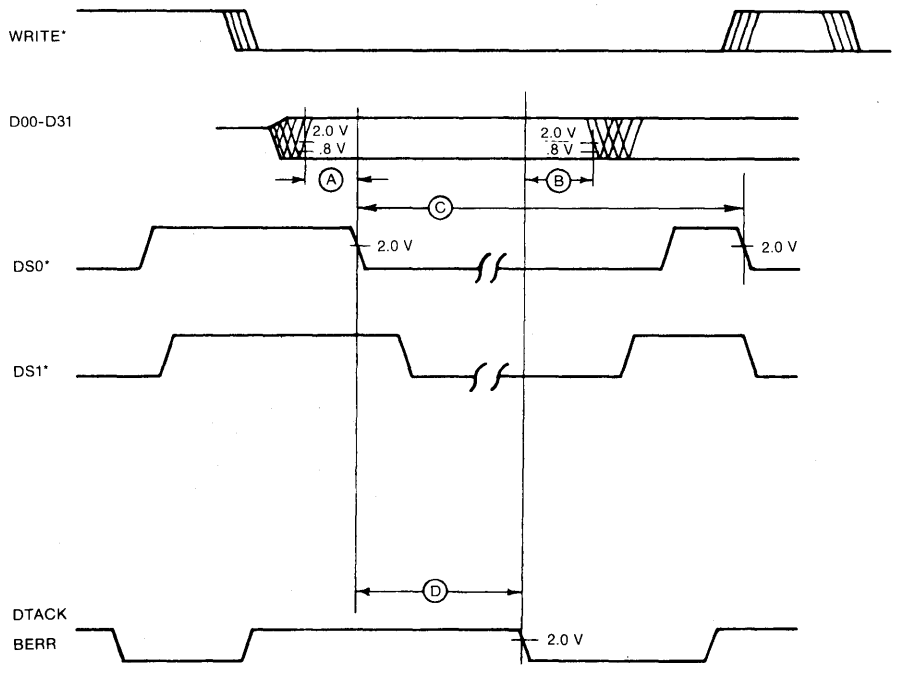


FIGURE 1 — MVME200/201 Functional Block Diagram

WRITE CYCLE TIMING

The timing parameters for the cycle performed to write data in memory are shown in Figure 2. A write cycle is initiated when WRITE*, followed by DS0* and DS1*, become active on the VMEbus.

FIGURE 2 — Write Cycle Timing Parameters

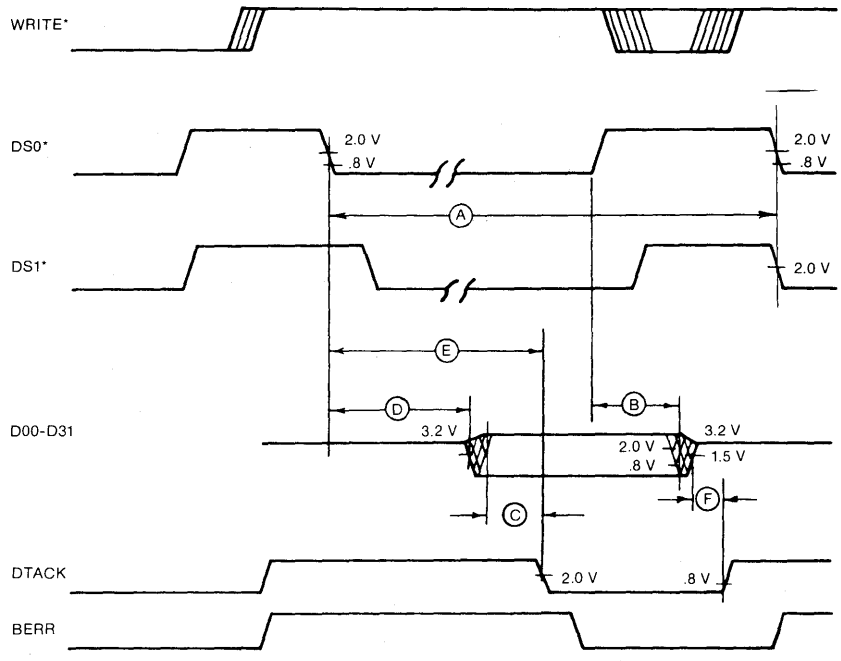


Parameter	Description	Min	Max	Unit
A	Valid data to DS0*	0		ns
B	DTACK* to invalid data	0		ns
C	Cycle time	565	595	ns
D	DS0* to DTACK*	315	370	ns

READ CYCLE TIMING

The timing parameters for the cycle performed to read data from memory are shown in Figure 3. A read cycle is initiated when WRITE* is inactive and DS0* and DS1* become active on the VMEbus.

FIGURE 3 — Read Cycle Timing Parameters



Parameter	Description	Min	Max	Unit
A	Cycle time	565	595	ns
B	Inactive DS0* to invalid data	10		ns
C	Valid data to DTACK*	28		ns
D	DS0* to active data	9		ns
E	DS0* to DTACK*	330	385	ns
F	Active data to inactive DTACK*	28		ns

BYTE PARITY GENERATION CHECKING

To insure the greatest integrity of stored data, MVME200 and MVME201 have circuitry for generating and checking byte parity (odd). On a write access, parity is calculated and the appropriate bit value is stored with the data byte. On a read access, parity is re-calculated for the data byte and compared with the stored parity bit value.

When an error is detected, the local signal PAR ERR* is generated, causing an active open collector signal to be placed on the MVMEbus BERR* line. Simultaneously, the front panel PARITY ERROR light comes on. This red LED remains on until any byte is read without parity error.

As a diagnostic aid in determining a source of parity errors, both MVME200 and MVME201 have a header for disabling by jumper the generation of the VMEbus BERR* signal on parity error detection. Disabling BERR* also disables the parity error indicator. An additional header equipped with four jumpers which allow parity to be forced in high and low memory bytes is only used for parity testing.

ALTERNATE ADDRESSING MODES

To provide a flexible means of using the VMEbus address modifier lines to implement system features such as separation of user access from privileged access in an operating system, MVME200 and MVME201 have a programmable, bipolar, address modifier PROM (256 × 4). The PROM decodes the states of the VMEbus IACK* and AM0 through AM5 lines to produce four block select signals which are applied to the modules' address decoder. Response to a custom input code may be obtained by reprogramming the address modifier PROM.

VMEbus INTERFACE

All VMEbus address and data signals and some control signals entering and leaving a module pass through buffers and/or latches. The LWORD* and all address lines are connected to active latches. Information on these lines is latched by a locally generated strobe. The IACK* and AM0 through AM5 lines are connected to logic circuitry which generates a block select signal that is gated to the main address decoding circuitry by a locally generated strobe. All data lines and the WRITE*, DS0* and DS1* signals are interfaced to buffered logic which generates local write, read, upper data strobe and lower data strobe signals and which transmits data when gated by a locally generated strobe. The VMEbus control signals BERR*, DTACK* and SYSRESET* are interfaced directly to the module control circuitry.

MVME200/201 SPECIFICATIONS

The specifications for MVME200 and MVME201 are listed in Table 1:

TABLE 1 — MVME200/201 Specifications

Characteristic	Specification	
Storage Capacity	64K Bytes (MVME200) 256K Bytes (MVME201)	
Word Length	8 Bits or 16 Bits	
Memory Organization	16K Byte Blocks (MVME200) 64K Byte Blocks (MVME201)	
Write Cycle Time	595 ns max.	
Read Cycle Time	595 ns max.	
Error Detection	Byte Parity, Odd	
Input Loading	One Schottky TTL Load Per Line	
Output Loading	Open-collector Output (I_{sink} max. = 48 mA) Three-state Output (I_{sink} max. = 64 mA)	
Temperature Operating Storage	 0° to 70° C -55° to 85° C	
Relative Humidity	0% to 90% (non-condensing)	
Power Requirements MVME200 MVME201	 +5 Vdc @ 2.7 A (max.) +12 Vdc @ 0.5 A (max.) −12 Vdc @ 6.5 mA (max.) +5 Vdc @ 3.3 A (max.)	
Dimensions Height Depth Thickness	*(Board Only)* 6.31 in. (160.3 mm) 9.19 in. (233.4 mm) 0.062 in. (1.57 mm)	*(With Front Panel)* 7.40 in. (188 mm) 10.31 in. (261.9 mm) 0.80 in. (20.32 mm)

MVME200/201 USAGE

MVME110-1	VMEmodule Monoboard Microcomputer
VME/10	Microcomputer System
MVME101	VMEmodule Monoboard Microcomputer

Ordering Information

Part Number	Description
MVME200	VMEmodule 64K Byte Dynamic RAM with Byte Parity, Includes User's Manual
MVME201	VMEmodule 256K Byte Dynamic RAM with Byte Parity, Includes User's Manual
MVME200/D	MVME200/201 64K/256K Byte Dynamic Memory Module User's Manual

Other VMEmodules Include:

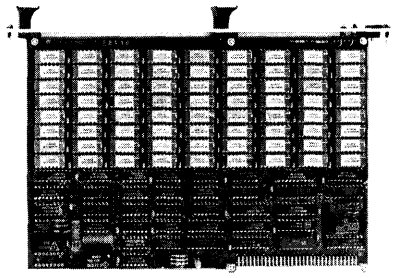
Part Number	Description
MVME101	VMEmodule Monoboard Microcomputer (8 MHz MC68000 MPU, Serial & Parallel Ports)
MVME110-1	VMEmodule Monoboard Microcomputer (8 MHz MC68000 MPU, Serial Port, I/O Channel)
MVME211	VMEmodule Static RAM/ROM (16 Sockets for up to 128K Bytes of RAM/ROM/PROM/EPROM)
MVME300	VMEmodule GPIB Controller with DMA (Provides IEEE-488 Listener, Talker, Controller Functions)

Related Documentation

HB212/D	VMEbus Specification Manual

MVME202
MVME222-1
MVME222-2

VMEmodule 512K/1M/2M Byte Dynamic RAM



- 512K Byte Version Using 64K-Bit Devices
- 1M and 2M Byte Versions Using 256K-Bit Devices
- Byte or Word Data Format
- Selectable Two-Way Interleaving
- Selectable Parity Generation/Checking High or Low Byte
- Base Address Selectable Over 16 Megabyte Map on Boundaries Corresponding to RAM Capacity
- Read Access Time, Max (Module Only)
  - — 280 ns (without parity)
  - — 310 ns (with parity)
- Write Access Time, Max (Module Only)
  - — 60 ns
- On-Board Refresh Circuitry
- PROM Address Modifier Decoder
- Two Front Panel Status Indicator LEDs
  - — Parity Error
  - — Module Busy
- Operating Temperature Range 0° C to 70° C
- Double High Eurocard Form Factor
- VMEbus Compatible

The MVME202, MVME222-1 and MVME222-2 are VME-module Dynamic RAM boards used in VMEbus-based systems to increase global memory. They use an 18-bit row organization to implement both byte and double byte (word) accessing and upper or lower byte parity generation and checking, as selected.

The four 18-device rows on MVME202 hold 64K-bit memories providing a capacity of 512K bytes. The four 18-device rows on MVME222-2 hold 256K-bit memories providing a capacity of two megabytes. On MVME222-1, two 18-device rows are filled with 256K-bit memories to provide a capacity of one megabyte. Figure 1 is a Functional Block Diagram of the Dynamic RAM modules.

BASE ADDRESS SELECTION

Each module has a header for jumper selection of a base address. For each module, the boundaries on which an address can be set are spaced by an amount equivalent to the module's size. For MVME202 for example, one of 32 base addresses on 512K-byte boundaries over a 16 megabyte space can be selected.

ADDRESS DECODING

Each module has control logic which activates latches to retain information on the VMEbus address lines. In addition, the code on the address modifier lines and the IACK* and LWORD* signals are routed to an address modifier decoder which generates a local select signal.

The information on address lines A01 and A19-A23 is sent to the module address decoder and if it matches the module address, an address select signal is asserted. A row and column address multiplexer tests address lines A02-A17 (64K) and A19 (256K) to determine the location of the memory being accessed in the row.

The memory array (row) selection logic uses address lines A18 and A19 (64K) or A20 and A21 (256K) to select a particular row.

Both the local select signal and the address select signal are combined to generate a board select signal and illuminate the module busy LED on the front panel.

INTERLEAVED OPERATION

Interleaving is a means of improving throughput during sequential write accessing such as DMA. Words are stored alternately on two individual modules using one for the even locations, the other for the odd locations. Latches are required for data and address lines so the module can complete a write access on its own allowing an MPU to proceed with the next write access without waiting for an acknowledge signal to be placed on the bus by the module. For interleaved operation, each module decodes address line A01 so that even and odd locations can be recognized, latches information on data and address lines and has a header allowing configuration for even or odd address response.

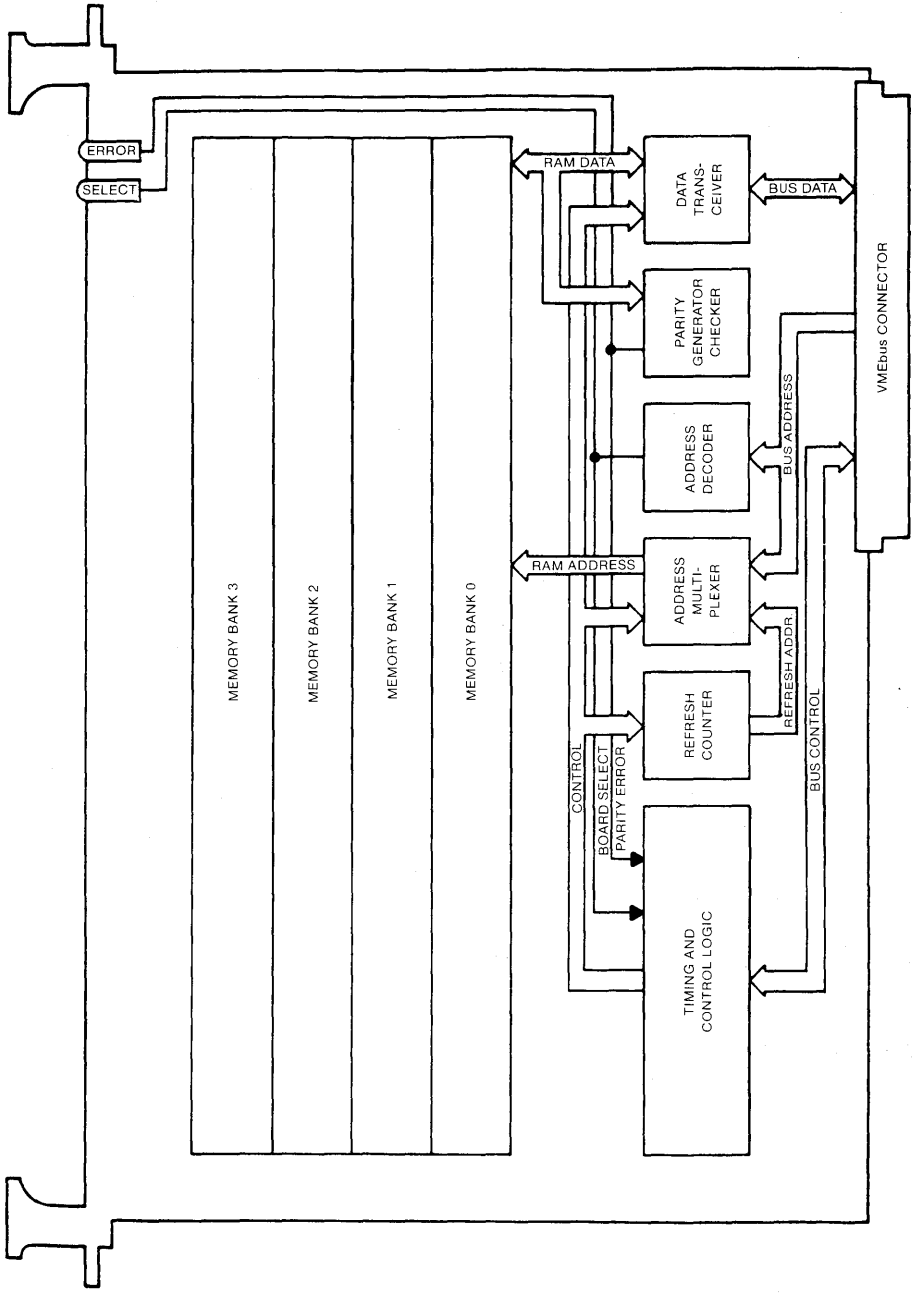


FIGURE 1 — VMEmodule Dynamic RAM Functional Block Diagram

PARITY GENERATION AND CHECKING

The three modules have parity generation and checking circuitry and a header for enabling or disabling the function or for forcing incorrect parity for testing. During a write operation, the parity generator for both the high and low order bytes calculates odd parity and sets the parity bit in each byte accordingly. During a read operation, the stored data and the parity bit for the high and low order bytes are processed by the parity checker which recalculates the parity values and compares them with the stored values. A detected error causes illumination of the front panel parity error LED and assertion of the VMEbus signal BERR*. Otherwise DTACK* is generated and asserted.

TIMING AND REFRESH CONTROL

The modules have refresh circuitry including a 64 kHz oscillator which continuously and asynchronously initiates refresh cycles so that memory devices are refreshed every 15 microseconds. Contention between a request for initiation of a memory cycle caused by a read or a write access and for initiation of a refresh cycle is accommodated using an arbiter and timer.

Mechanical and Environmental Specifications

Characteristics	Specifications
Memory Device Type	2 or 4 Rows of 64K-Bit or 256K-Bit Devices
Memory Addressing	1 Block of 512K, 1M or 2M Bytes in a 16 Megabyte Map
Data Format	Byte or Word
Parity Generation/Checking	Selectable Odd Parity, Upper and Lower Byte
Interleaving Provision	Selectable Odd or Even Access Response
Status Indicators	Parity Error Front Panel LED Module Busy Front Panel LED
Read Access Time (max)	280 ns (with parity) 250 ns (without parity)
Write Access Time (max)	60 ns
Form Factor	Double High Eurocard
Power Requirements	2.0 mA (max) at 5.0 Vdc
Temperature Operating Storage	 0° C to 70° C -25° C to 85° C
Humidity	0% to 95% (non-condensing)
PC Board Dimensions Height (without panel) Height (with panel) Depth Thickness	 9.2 in. (234 mm) 10.3 in. (262 mm) 6.3 in. (160 mm) 0.79 in. (20 mm)

Ordering Information

Part Number	Description
MVME202	VMEmodule 512K Byte Dynamic RAM with Byte Parity. Includes User's Manual.
MVME222-1	VMEmodule 1M Byte Dynamic RAM with Byte Parity. Includes User's Manual.
MVME222-2	VMEmodule 2M Byte Dynamic RAM with Byte Parity. Includes User's Manual.
MVME202/D	MVME202/222 512K/1M/2M Dynamic RAM User's Manual.

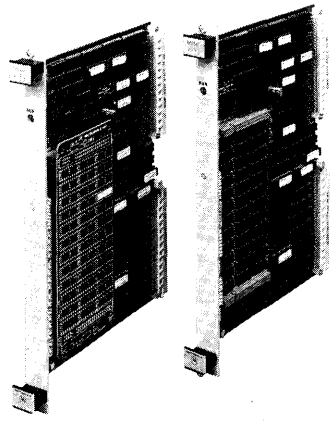
Related Documentation

HB212/D	VMEbus Specification Manual.

MVME204-1
MVME204-2

VMEmodule™ 1Mb/2Mb Dynamic RAM w/VSB*

- Supports VMEbus/VSB
- Dual-ported — 32-Bit Address/data VMEbus and Multiplexed 32-Bit Address/data VSB Interface
- Interleaving — Two-way Interleaving on VMEbus or VSB
- Configurable Array — Dynamically Alterable to be Partially or Fully Private to the VSB Port. Selectable in 1/4 Population Increments
- Byte Parity Generation and Error Checking Circuitry
- Longword (32-bit), Word (16-bit), or Byte (8-bit) Data Transfers
- VMEbus Addressing — Automatic Selection of 24- or 32-Bit Addressing on VMEbus Interface, 32-Bit Addressing on VSB
- Memory Base Address Settable on Board-Size Boundaries throughout VMEbus and VSB Address Space
- Transparent Refresh Support



FUNCTIONAL DESCRIPTION

The MVME204-1 and MVME204-2 are VMEmodules offering, respectively, one and two megabytes of dual-ported dynamic RAM with parity for use with VMEbus and the dedicated, high-speed memory bus tailored for the MC68020 32-Bit Microprocessor — VSB. Interfaces to this bus enhance the performance of the MVME130 VMEmodule 32-bit Monoboard Microcomputer by allowing the transfer of data between the MVME130 and MVME204 modules concurrent with DMA transfers over VMEbus.

The one megabyte capacity of MVME204-1 is attained using 256K by 1-bit dynamic RAM devices. A mezzanine board containing an additional megabyte is added to achieve the two megabyte capacity of the MVME204-2. Both modules have parity generation and detection circuitry which, together with a VMEbus accessible control and status register, can be used for error detection and memory diagnostics.

Having a VMEbus interface, the modules can be applied in systems using this interconnect structure. Care must be taken with existing VMEmodule-based systems, since VSB and the extended 32-bit data and address use P2 of the VMEbus connector. The pins on connector P2 of the existing VMEbus system are commonly used as I/O pins.

VMEbus ADDRESS MODIFIER CODE RESPONSE

VMEbus address modifier line decoding on the MVME204 provides response to both standard and extended, supervisory and non-privileged program and data accesses.

A programmable array logic device (PAL) is used for address modifier line decoding. This PAL is socketed to facilitate application of the modules in systems having special requirements.

Figure 1 is a functional block diagram of the modules.

VMEbus ADDRESS MAPPING

Both modules provide two switches (12 bits) used to select for a module, a base address in VMEbus address space. All 12 switch positions are used to set MVME204-1 VMEbus memory to begin on any 1024Kb boundary. Eleven switch positions are used to set MVME204-2 VMEbus memory to begin on any 2048Kb boundary. The ending address in VMEbus space for a module is the sum of the starting address and the module population.

VSB Address Mapping

Both modules provide two switches (12 bits) used to select for a module a base address in VSB address space. All 12 switch positions are used to set MVME204-1 memory to begin on any 1024Kb boundary. Eleven switch positions are used to set MVME204-2 memory to begin on any 2048Kb boundary. The ending address in VSB space for a module is the sum of the starting address and the module population. Note that an additional requirement is placed on the system programmer when the option of using non-identical starting addresses for a module in the VMEbus and VSB address spaces is chosen.

*MVMX32bus is a Subset of the VME Subsystem Bus.
VMEmodule is a trademark of Motorola Inc.

STATUS/CONTROL REGISTER

An 8-bit, writable and readable Status/Control Register is used for controlling module functions and for reporting a parity error incurred during a VMEbus or VSB read access. The register is accessed by placing on the VMEbus address modifer lines the short supervisory I/O access code (hexadecimal 15). Location in the VMEbus I/O space of the register is switch selectable.

The modules facilitate dynamic global/private memory allocation using three bits in the Status/Control Register. Bit 6, when set, disables any VMEbus access in effect allocating all four module memory segments to access from VSB only. Various combinations of the states of bits 4 and 5 are used to remove one to four memory segments from VSB-only allocation and allow access also from VMEbus. Segment size is 256Kb and 512Kb for the MVME204-1 and MVME204-2, respectively.

Bit 3 of the Control/Status Register may be set to enable reporting via bus error of any VMEbus device attempting to access module memory allocated to private VSB access.

In applications where it is not desirable for the MVME130 system to cache the VSB port, this function may be disabled by setting bit 2 in the Control/Status Register.

For use in diagnostics, bit 1 in the Control/Status Register may be set to cause the wrong parity value to be written to an addressed location.

Two bits in the Control/Status Register are involved in parity control and reporting. The module indicates that a parity error has occurred on a VMEbus or VSB read access by setting bit 7 which is normally cleared from the VMEbus but can also be set for diagnostic use. Bit 0 is set by the user to enable indication to a selecting device via the BERR* or MERR* signals of a parity error. Module reporting of read access parity status via bit 7 is not affected by the state of bit 0.

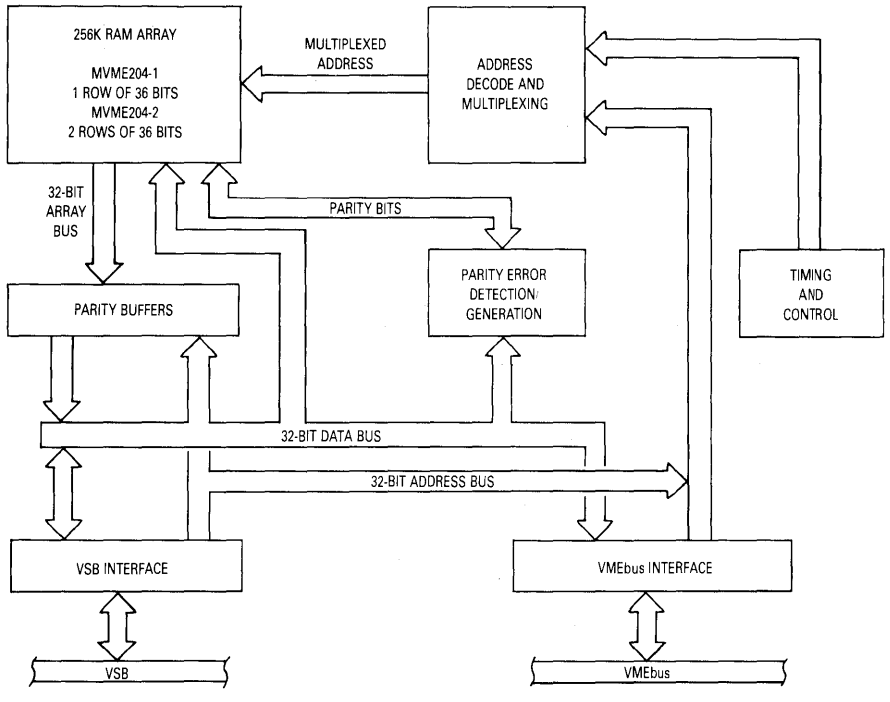


Figure 1. MVME204 Functional Block Diagram

MVME204 STATUS CONTROL REGISTER FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
PE	RP	PR1	PR0	PRBEREN	MCACHE*	WWP	EPER

PE: <Parity Error> The MVME204 status bit, when set indicates that a parity error has occurred from a VMEbus or VSB read access. It can be cleared by writing a zero from the VMEbus or can be set for diagnostic purposes.

RP: <VSB Private> When set, RP disables any VMEbus access to the MVME204 DRAM array.

PR1,PR0: <Private RAM> PR1 and PR0 allocates the MVME204 DRAM array area to the VSB shown by the following table. These bits are "don't cares" when the RP bit is set.

PR1, PR0	Memory Segment VSB Accessible
0 0	3, 2, 1, 0
0 1	3, 2, 1
1 0	3, 2
1 1	3

PRBEREN: <Private Bus Error Enable> When set, PRBEREN causes a Bus Error to be issued to the VMEbus device attempting to access memory that has been allocated private to VSB by the PR1 or PR0 bits, or the RP bit of the MVME204 Control Register.

MCACHE*: <Memory Cacheable> When cleared allows the VSB port to be cached by the MVME130 system. MCACHE enables assertion of MCACHE* on the VSB with the same timing as MASACKO, 1*. Indicating that the selected memory is cacheable.

WWP: <Write Wrong Parity> When set, causes the wrong parity to be written to the addressed location for diagnostic purposes.

EPER: <Enable Parity Error Report> When set, allows the MVME204 error detecting circuitry to report errors to the selecting device. Errors are indicated by the BERR* or MERR*. The PE bit in the MVME204 status/control register will not be affected by EPER. The PE bit will always indicate an error if one occurs. The EPER and WWP bits should not be set simultaneously.

*Active Low

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements	MVME204 — +4.75 to 5.25 Vdc @ 5 A (max)
Operating Temperature	0° to 55°C
Storage Temperature	−40° to 85°C
Relative Humidity	5% to 95% (non-condensing)
Physical Dimensions Height Width Thickness	 9.25 in. (23.50 cm) 14.50 in. (36.83 cm) 0.60 in. (1.52 cm) MVME204-1 0.75 in. (1.91 cm) MVME204-2
Storage Capacity	1Mb (MVME204-1) 2Mb (MVME204-2)
Data Transfer Size	8-, 16-, 32-bits
Error Detection	Odd Byte Parity
Data Input/Output	32-bit VMEbus/VSB data
Input Address	32-bit VMEbus/VSB addressing

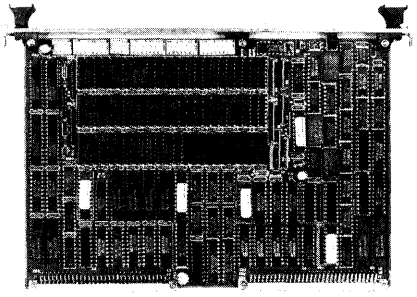
ORDERING INFORMATION

Part Number	Description
MVME204-1	VMEmodule 1024Kb Dynamic RAM with Byte Parity. Includes User's Manual.
MVME204-2	VMEmodule 2048Kb Dynamic RAM with Byte Parity. Includes User's Manual.
MVME204-1, -2/D	VMEmodule Dynamic RAM User's Manual.

RELATED PRODUCTS

Part Number	Description
MVME130	VMEmodule 32-bit Monoboard Microcomputer with MC68020 CPU and VSB. Includes User's Manual.
MVME131	Same as MVME130 but also includes Memory Management Board.
MVME214	Static RAM/ROM VMEmodule with VSB. Includes User's Manual.

VMEmodule™ 2Mb Dynamic RAM Module with VSB



- Enhanced Performance Over the MVME204-2
- Supports VMEbus/VSB (VME Subsystem Bus)
- Dual-ported 32-Bit Address/Data VMEbus and Multiplexed 32-Bit Address/Data VSB Interface
- Interleaving — Two-way Interleaving on VMEbus or VSB
- Cache and Non-Cache Operation
- Configurable Array — Dynamically Alterable to be Partially or Fully Private to the VSB Port. Selectable in 1/4 Population Increments
- Byte Parity Generation and Error Checking Circuitry
- Longword (32-bit), Word (16-bit) or Byte (8-bit) Data Transfers
- VMEbus Addressing — Automatic Selection of 24- or 32-Bit Address on VMEbus Interface, 32-Bit Addressing on VSB
- Memory Base Address Settable on Board-Size Boundaries throughout VMEbus and VSB Address Space
- Transparent Refresh Support

FUNCTIONAL DESCRIPTION

The MVME204-2F is a VMEmodule offering two megabytes of dual-ported dynamic RAM with parity for use with VMEbus and the dedicated, high-speed secondary bus — VSB. Interfaces to VSB enhance the performance of processor modules such as the MVME130 by allowing the transfer of data between the MVME130 and the MVME204-2F modules concurrent with transfers over VMEbus.

The 2Mb capacity of the MVME204-2F is attained using 256K by 1-bit dynamic RAM ZIP devices. The module has parity generation and detection circuitry which, together with a VMEbus accessible control and status register, can be used for error detection and memory diagnostics.

Having a VMEbus interface, the module can be applied in systems using this interconnect structure. Care must be taken with existing VMEmodule-based systems, since VSB and the extended 32-bit address and data use P2 of the VMEbus connector. The pins in rows A and C of P2 are commonly used on existing VMEmodules as I/O pins.

Figure 1 is a functional block diagram of the module.

VMEbus ADDRESS MODIFIER CODE RESPONSE

The address modifier line decoding on the MVME204-2F provides response to both standard and extended, supervisory and non-privileged program and data accesses.

VMEmodule is a trademark of Motorola Inc.

A programmable array logic device (PAL) is used for address modifier line decoding. This PAL is socketed to facilitate application of the modules in systems having special requirements.

VMEbus ADDRESS MAPPING

The MVME204-2F provides two switches (11 bits) used to select a base address in VMEbus address space. These switches set the VMEbus base address to begin on any 2048Kb boundary. The ending address in VMEbus space is the sum of the starting address and the module population (2Mb).

VSB ADDRESS MAPPING

The MVME204-2F provides two switches (11 bits) used to select a base address in VSB address space. These switches set the VSB base address to begin on any 2048Kb boundary. The ending address in VSB space is the sum of the starting address and the module population (2Mb). Note that an additional burden is placed on the system programmer when the option of using non-identical starting addresses for a module in the VMEbus and VSB address spaces are chosen.

STATUS AND CONTROL REGISTER

An 8-bit wide, writable and readable Status/Control Register is used for controlling module functions and for reporting a parity error incurred during a VMEbus or VSB read access. The register is accessed by placing on the VMEbus address modifier lines the short supervisory or short non-privileged I/O access code (hexadecimal 2D or 29). Location on word boundaries of the register in the VMEbus I/O space is switch selectable.

The module facilitates dynamic global/private memory allocation using bits 4, 5 and 6 in the Status/Control Register. Bit 6, when set, disables any VMEbus access, in effect allocating all four module memory segments to accesses from VSB only. Various combinations of the

states of bits 4 and 5 are used to remove one to four memory segments from VSB-only allocation and allow access also from VMEbus. Segment size is 512Kb.

Bit 3 of the Control/Status Register may be set to enable reporting via bus error of any VMEbus device attempting to access memory allocated to private VSB accesses.

In applications where it is not desirable for the memory of the MVME204-2F to be cached through the VSB port, this function may be disabled by setting bit 2 in the Control/Status Register.

For use in diagnostics, bit 1 in the Control/Status Register may be set to cause the wrong parity value to be written to an addressed location.

Two bits in the Control/Status Register are involved in parity control and reporting. The module indicates that a parity error has occurred on a VMEbus or VSB read access by setting bit 7, which is normally cleared from the VMEbus, but can also be set for diagnostic use. Bit 0 is set by the user to enable the indication to a selecting device via the BERR* or MERR* signals of a parity error. Module reporting of read access parity status via bit 7 is not affected by the state of bit 0.

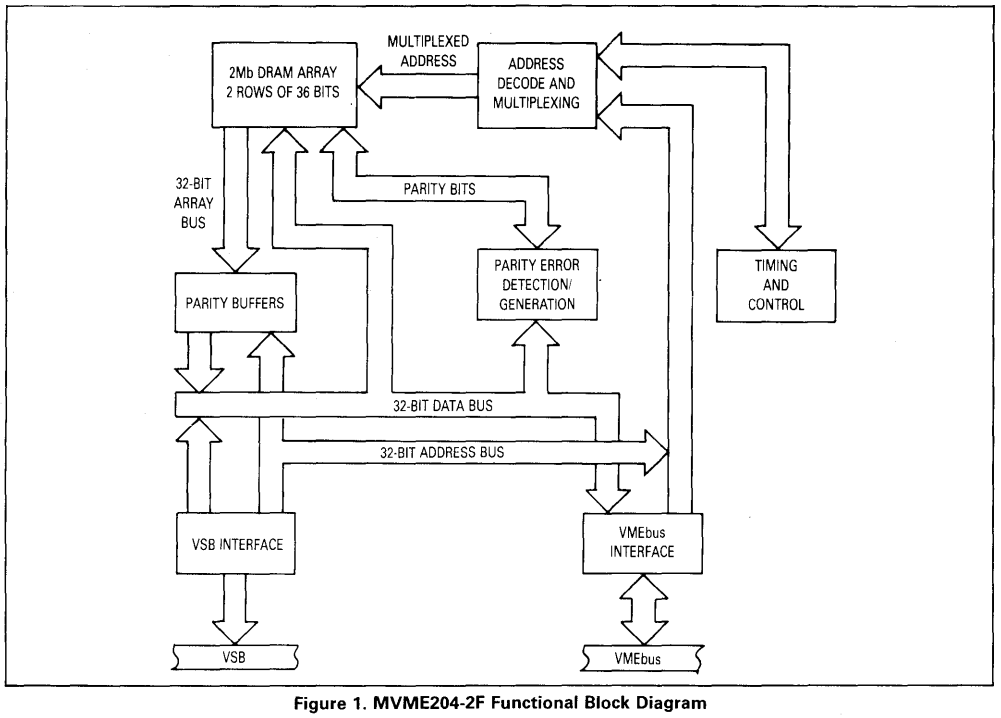


Figure 1. MVME204-2F Functional Block Diagram

MVME204-2F INTERLEAVING

To provide an increase in speed by decreasing access time, MVME204-2F VMEmodules support two way interleaving both on the VMEbus and on the VSB. Interleaving is done by permitting a first module to handle even addresses and a second module to handle odd addresses so that the two appear as a single 4Mb module and are treated as such for data transfers. There is no limit to the number of pairs of modules that can be used in the interleave mode.

Interleaving provides the greatest speed increase for VMEbus write cycles because of the fast write mode built into the MVME204-2F VMEbus interface that allows the second module to be accessed while the first is completing the access portion of its cycle.

VSB CACHE MODE OPERATION

In a system with a VMEbus processor having data cache capabilities, the MVME204-2F can operate in the cache mode. In this mode, significant access time is saved since the processor can begin a read access on the VSB before, as it normally would, determining a hit and receiving the data from cache memory then terminating the data transfer cycle. The MVME204-2F, in the cache mode, waits for the DS* signal (Data Strobe) before starting its internal memory read cycle to preclude being in this condition when the VSB cycle is terminated on a cache hit. On a cache miss when the data is not in cache, the read cycle initiated by the processor continues to full completion.

MVME204-2F Status/Control Register Format

D7	D6	D5	D4	D3	D2	D1	D0
PE	VSBP	PR1	PR0	PRBEREN	MCACHE*	WWP	EPER

PE: <Parity Error> The MVME204-2F status bit, when set, indicates that a parity error has occurred from a VMEbus or VSB read access. It can be cleared by writing a zero from the VMEbus or can be set for diagnostic purposes.

VSBP: <VSB Private> When set, VSBP disables any VMEbus access to the MVME204-2F DRAM array.

PR1,PR0: <Private Ram> PR1 and PR0 allocate the MVME204-2F DRAM array area to the VSB shown by the following table. These bits are "don't cares" when the VSBP bit is set.

PR1, PR0	Memory Segment VSB Accessible
0 0	3, 2, 1, 0
0 1	3, 2, 1
1 0	3, 2
1 1	3

PRBEREN: <Private Bus Error Enable> When set, PRBEREN causes a Bus Error to be issued to the VMEbus device attempting to access memory that has been allocated private to VSB by the PR1 or PR0, and the VSBP bit of the MVME204-2F Control Register.

MCACHE*: <Memory Cacheable> When cleared, allows the VSB port to be cached. MCACHE* enables assertion of MCACHE* on VSB with the same timing as MASACK0,1*, indicating that the selected memory is cacheable.

WWP: <Write Wrong Parity> When set, causes the wrong parity to be written to the addressed location for diagnostic purposes.

EPER: <Enable Parity Error Report> When set, allows the MVME204-2F error detecting circuitry to report errors to the selecting device. Errors are indicated by the BERR* or MERR*. The PE bit in the MVME204-2F Control/Status Register is not affected by EPER and will always indicate an error if one occurs. The EPER and WWP bits should not be set simultaneously.

*Active Low

MVME204-2F Timing
(All times are typical values)

Access Sequence	Write	Read	Notes
DS* to ACK* (VSB)	135 ns (7)	110 ns (7)	1, 5
AS* to ACK* (VSB)	235 ns (7)	210 ns (7)	1, 5
DS* to ACK* (VSB)	135 ns (8)	180 ns (7)	2, 5
AS* to ACK* (VSB)	235 ns (8)	280 ns (7)	2, 5
AS* to DTACK* (VMEbus)	150 ns (8)	260 ns (10)	3, 5
AS* to DTACK* (VMEbus)	150 ns (8)	240 ns (9)	4, 5

NOTES: 1. VSB is in the non-cacheable mode, which causes the MVME204-2F to start internal DRAM access prior to receiving DS* from the initiating device.
2. VSB is in the cacheable mode, which causes the MVME204-2F to hold the actual DRAM access sequence until after DS* has been received from the initiating master.
3. VMEbus is not enabled for early release of DTACK*.
4. VMEbus is enabled for early release of DTACK*.
5. Numbers in parenthesis are the total number of cycles required to complete the specified transfer (read or write) when connected to the MVME130DON Processor Module, an MC68020-based monoboard microcomputer operating at 16.67 MHz.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements	+4.75 to 5.25 Vdc @ 5 A (max)
Operating Temperature	0°C to +55°C inlet air temperature with forced air cooling 0°C to +40°C ambient, convection cooling
Storage Temperature	−40°C to +85°C
Relative Humidity	5% to 90% relative humidity (non-condensing)
Physical Dimensions Height Width Thickness	 9.25 in. (23.50 cm) 14.50 in. (36.83 cm) 0.60 in. (1.52 cm)
Storage Capacity	2Mb
Data Transfer Size	8-, 16-, or 32-bits
Error Detection	Odd Byte Parity
Data Input/Output	32-bit VMEbus/VSB Data
Input Address	32-bit VMEbus/VSB Address

ORDERING INFORMATION

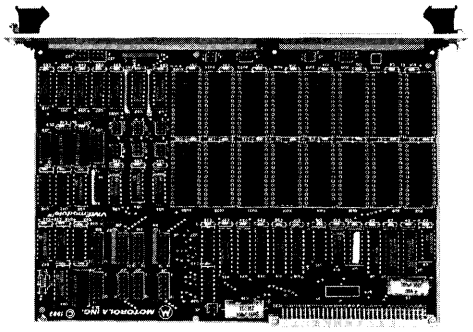
Part Number	Description
MVME204-2F	2Mb, Dynamic RAM VMEmodule with odd byte parity and full 32-bit VMEbus and VSB interfaces. Includes User's Manual.
MVME204-2F/D	MVME204-2F User's Manual.

RELATED PRODUCTS

Part Number	Description
MVME130	VMEmodule 32-bit Monoboard Microcomputer with MC68020 MPU and VSB Interface. Includes User's Manual.
MVME131	VMEmodule 32-bit Monoboard Microcomputer with MC68020 MPU, Memory Management and VSB Interface. Includes User's Manual.
MVME130XT	VMEmodule 32-bit Monoboard Microcomputer with MC68020 MPU, MC68881 FPCP, VSB Interface, a 16Kb Instruction/Data Cache and MVME130bug. Operates at 16.67 MHz. Includes User's Manual.
MVME131XT	VMEmodule 32-bit Monoboard Microcomputer with MC68020 MPU, MC68881 FPCP, Memory Management, VSB Interface, a 16Kb Instruction/Data Cache and MVME130bug. Operates at 16.67 MHz. Includes User's Manual.

MVME211

VMEmodule STATIC RAM/ROM



- Double High Eurocard Form Factor
- VMEbus Compatible
- Up to 1 Megabyte of Storage Capacity (using 64K x 8 devices)
- MVME211 Cycle Time 145 ns, Nominal
- Access Times from 35 to 510 ns Typical
- Accepts JEDEC Standard 24- and 28-Pin Devices
- Battery Backup Provision
- 0°C to +70°C Operating Temperature Range

The MVME211 Static RAM/ROM VMEmodule is used in VMEbus-based microcomputer systems to provide additional data and program storage. It offers up to 1 megabyte of capacity using 64K X 8 devices. Sixteen sockets are provided for JEDEC standard 24- and 28-pin memories. Devices having access times ranging from 35 ns to 510 ns can be used. Table 1 lists some currently popular devices which can be used with the static RAM/ROM module.

To provide mapping and system configuration flexibility, memory is organized in two independent blocks of 8K, 16K, 32K, 64K, 128K or 256K 16-bit words in which 2K, 4K, 8K, 16K, 32K or 64K byte devices can be used. Provision for implementing battery backup operation is provided. Figure 1 is a functional block diagram of the static RAM/ROM module.

BASE ADDRESS SELECTION

The module has headers for jumper selection of a base address for each of the two memory blocks. The base address for a block can be set on any 4K byte boundary throughout the VMEbus address space. The memory space for a block can incorporate that of unused words (socket pairs), if desired. A block can be made up of mixed pairs of equivalent devices provided the pairs are pin compatible and of equal capacity. An unused block may be disabled.

DEVICE SIZE SELECTION

The static RAM/ROM module is designed to support many of the JEDEC standard 24- and 28-pin memory devices, some of which are listed in Table 1. Two headers allow jumper selection of devices ranging in size from 2K x 8 through 64K x 8.

ACCESS TIME SELECTION

The module has two headers for jumper selection of a read/write cycle access time which corresponds to installed memory devices. Access time choices range in 25 ns or 50 ns increments from 35 to 510 ns. In instances where devices are mixed, the access time of the slowest device is used.

DEVICE TYPE SELECTION

To accommodate the several types of supported devices, ten headers allow jumper connection of the required voltage and time varying signals to the appropriate pins of the device types used. The static RAM/ROM module supports JEDEC standard 24- and 28-pin RAM, ROM and EPROM devices.

BATTERY BACKUP

The module has a header by means of which, in conjunction with device type headers, battery backup operation can be jumper selected for one or both memory blocks. Backup operation requires that the system SYSRESET* signal timing complies with the VMEbus specification.

TABLE 1 — Typical MVME211 Supported Devices

Type	Size	No. of Pins	Device Number
EPROM	64K x 8	28	Intel 27512
EPROM	32K x 8	28	Intel 27256
EPROM	16K x 8	28	Intel 27128
EPROM	8K x 8	28	Intel 2764
EPROM	4K x 8	24	Intel 2732
EPROM	2K x 8	24	Intel 2716
ROM	32K x 8	28	MCM63256
ROM	16K x 8	28	MCM27128
ROM	8K x 8	28	MCM68369
RAM	8K x 8	28	Toshiba TC5564
RAM	2K x 8	24	Toshiba TC5517

MVME211 Specifications

The environmental, mechanical and electrical specifications for the MVME211 Static RAM/ROM VMEmodule are given in Table 2.

TABLE 2 — MVME211 Specifications

Characteristic	Specification
Memory Organization	Two 8 device blocks
Storage Capacity Bytes/Block	16K, 32K, 64K, 128K, 256K 512K (64K x 8 devices)
Data Organization	Word (16 bits)
Operating Modes	Write (Word or Byte) Read (Word or Byte) Read/Modify/Write (Byte)
Power Requirements Normal Operation Battery backup (excludes memory devices)	 +5.0 Vdc at 7.5 W (Max) +3.0 to +5.0 Vdc at 50 μW (Max)
Environmental Limits Operating Temperature Storage Temperature Humidity Range	 0°C to 70°C −20°C to 85°C 5.0 to 80% (non-condensing)
Mechanical Specifications Height x Depth (board) Height x Width (front panel)	 6.3″ (160 mm) x 9.2″ (234 mm) 10.3″ (262 mm) x 0.79″ (20 mm)

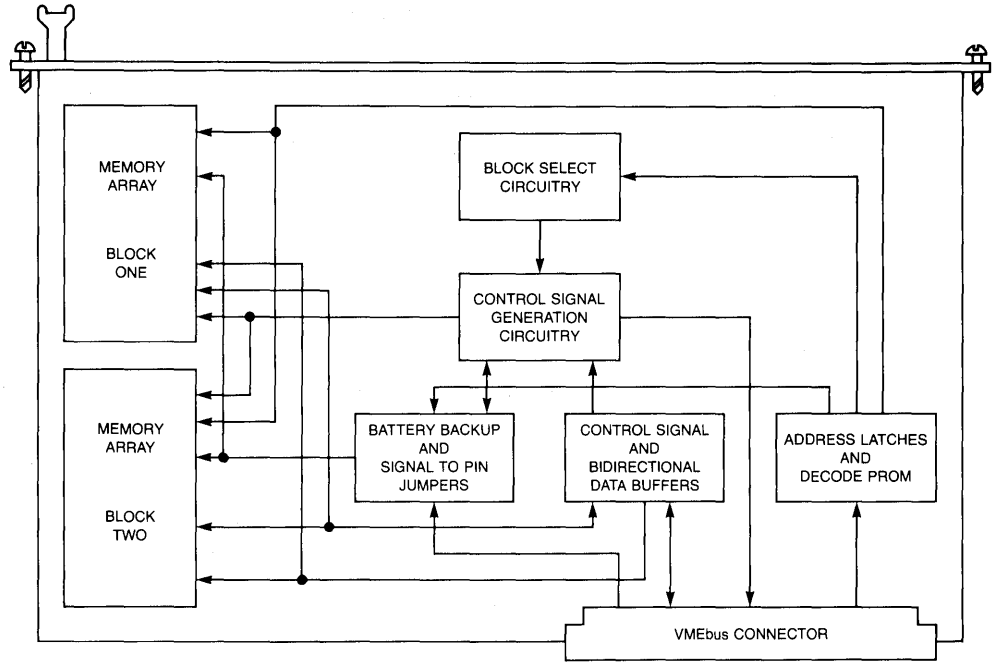


FIGURE 1 — MVME211 Functional Block Diagram

Ordering Information

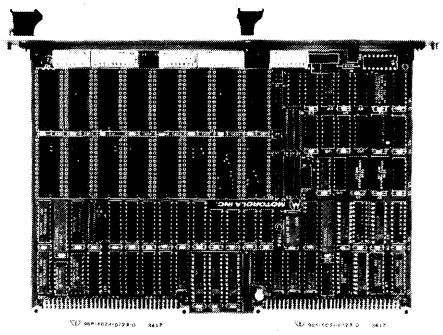
Part Number	Description
MVME211	VMEmodule Static RAM/ROM/EPROM Memory. Includes User's Manual
MVME211/D	VMEmodule RAM/ROM/EPROM Memory Module User's Manual

Related Documentation

HB212/D	VMEbus Specification Manual

VMEmodule™ Static RAM/ROM with VSB*

- Dual-Ported VMEbus/VSB
- Storage Capacity of 32Kb, 64Kb, 128Kb, 256Kb, 512Kb or 1Mb
- Full or Half-Populated Memory (switch selected)
- Supports 24- or 32-Bit addressing on VMEbus, 32-Bit addressing on VSB
- VSB Private Mode (jumper selected)
- Accepts JEDEC Standard 24- and 28-pin Devices
- Cycle Time 130 ns (nominal)
- Access Times from 100 ns to 400 ns (typical)
- Battery Backup Provision



The MVME214 Static RAM/ROM VMEmodule is used in VMEbus-based microcomputer systems to provide additional data and program storage. This VMEmodule offers up to 1Mb of storage capacity using 64Kb x 8 devices. To provide system configuration flexibility, memory is organized in two independent blocks in which 2Kb, 4Kb, 8Kb, 16Kb, 32Kb, or 64Kb devices can be used. Figure 1 is a functional block diagram of the MVME214.

VSB

The MVME214 VMEmodule supports the VSB, a dedicated memory channel tailored to the MC68020 microprocessor and designed to enhance the performance of the MVME130 VMEmodule 32-bit Monoboard Microcomputer. The MVME214 can be configured to be a private resource to the VSB. In this private mode, a VMEbus access to the module will not be acknowledged.

DEVICE SELECTION

The MVME214 Static RAM/ROM VMEmodule is provided with 16 sockets to support many of the JEDEC standard 24- and 28-pin RAM, ROM, and EPROM devices. To accommodate the several types of supported devices, jumper headers allow connection of the required voltage and time varying signals to the appropriate pins of the device type used. Devices selected may range in size from 2Kb × 8 through 64Kb × 8.

Different size devices cannot be mixed. However, different types of devices can be installed so long as they are separated by blocks. When ROM/EPROM devices are selected, the MVME214 will assert a bus error for write access attempts to these devices. Half population of the MVME214 is permitted. However, half population is restricted to memory block 1. Table 1 is a partial list of popular devices which can be used with the module.

BASE ADDRESS SELECTION

The MVME214 has switches which are used to select a base address for each of the two memory ports. The base address for a block may be set on population boundaries throughout the 4 gigabyte address space. The ending address is determined by setting size-select jumpers to values of 32Kb, 64Kb, 128Kb, 256Kb, 512Kb, or 1024Kb.

ACCESS TIME SELECTION

The module has two jumper-selected access time options related to a read/write access time of a device programmed using programmable array logic (PAL). The access time is set for 100 ns or 300 ns devices but can be programmed to produce access times varying from 100 ns through 400 ns. In instances where devices are mixed, the access time of the slowest device is used.

BATTERY BACKUP

The module has a header that, when used in conjunction with device-type headers, allows battery backup operation to be jumper-selected.

*The VME Subsystem Bus (VSB) is a subset of the VMEbus.

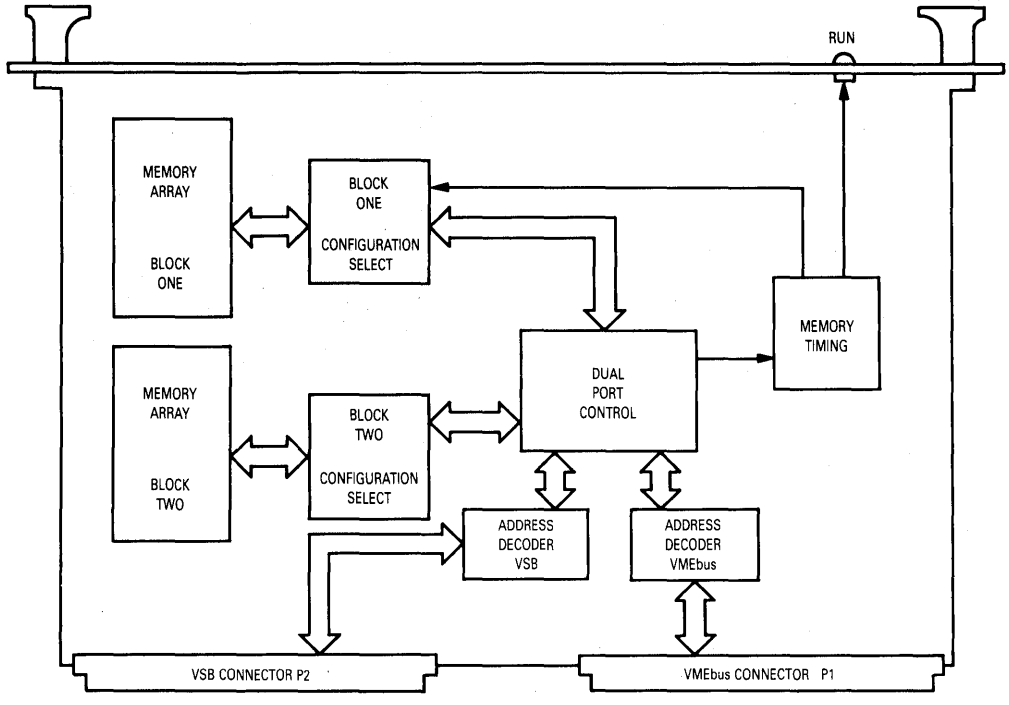


Figure 1. MVME214 Functional Block Diagram

Table 1. Typical MVME214 Supported Devices

Type	Size	No. of Pins	Device Number
EPROM	64K x 8	28	Intel 27512
EPROM	32K x 8	28	Intel 27256
EPROM	16K x 8	28	Intel 27128
EPROM	8K x 8	28	Intel 2764
EPROM	4K x 8	24	Intel 2732
EPROM	2K x 8	24	Intel 2716
ROM	32K x 8	28	MCM63256
ROM	16K x 8	28	MCM27128
ROM	8K x 8	28	MCM68369
RAM	8K x 8	28	Hitachi HM6264

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristic	Specification
Memory Organization	Two 8 device blocks
Storage Capacity Bytes/Block	16K, 32K, 64K, 128K, 256K, 512K (64K × 8 devices)
Data Organization	Longword (32 bits)
Operating Modes	Write (Byte, Word, or Longword) Read (Byte, Word, or Longword) Read/Modify/Write
Power Requirements Normal Operation Battery Backup (excludes memory devices)	 +5 Vdc at 3 A (typ) 4.5 A (max) +3 to +5 Vdc at 5 μW (max)
Operating Temperature	0°C to 50°C inlet air temperature
Storage Temperature	−20°C to 85°C
Humidity Range	5% to 80% (non-condensing)
Physical Dimensions Height Width Thickness	 6.30 in (16.00 cm) 9.20 in (23.37 cm) 0.062 in (0.157 cm)
Component Projections Component Side Solder Side	 0.50 in (1.27 cm) max 0.062 in (0.157 cm) max

ORDERING INFORMATION

Part Number	Description
MVME214	Static RAM/ROM VMEmodule with VSB. Includes User's Manual.
MVME214/D	Static RAM/ROM VMEmodule User's Manual.

RELATED PRODUCTS

Part Number	Description
MVME130	VMEmodule 32-bit Monoboard Microcomputer with MC68020 CPU and VSB. Includes User's Manual.
MVME131	Same as MVME130 but equipped with Memory Management Board.
MVME204	VMEmodule 1024Kb Dynamic RAM with VSB. Includes User's Manual.

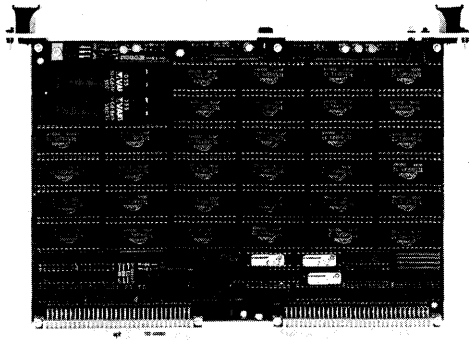
MVME215-1
MVME215-2
MVME215-3

Product Preview

256K/512K/1Mbyte CMOS RAM VMEmodules

- 256K/512K/1Mb RAM Capacity
- 8-, 16-, 32-Bit Data Bus Width
- 24-, 32-Bit Address Bus Width
- Address Modifier Codes Programmable
- Onboard Power Monitor
- Onboard Battery for Power Down Backup
- VMEbus Rev. C Compatible

The MVME215 is a high speed, high performance static CMOS random access memory module with various capacities (up to 1Mb). It can be accessed via the 8-/16-/32-bit data bus. Circuitry is provided which maintains the information stored in the RAM during power down. This module is ideally suited to applications requiring large-capacity non-volatile storage, for example in diskless systems.



VMEbus INTERFACE

The MVME215 is designed for use as a slave in a VME environment which complies with VMEbus Specification, Rev. C. It supports the standard addressing range of 16Mb and the extended addressing range of 4Gb. The appropriate mode is selected by means of address modifier codes.

The data path can be 8-,16- or 32-bit wide. Logic is provided which routes the local 8-bit data paths to their correct counterparts on the VMEbus.

The following slave module types are supported by the MVME215:

SLAVE MODULE OPTIONS:

	BLT Block Transfer Capability	RMW Read-Modify-Write Capability	UAT Unaligned Transfer Capability	ADO Address-Only Capability
A24 D08 (EO)	N	Y	N	N
A24 D16	N	Y	N	N
A24 D32	N	Y	N	N
A32 D08 (EO)	N	Y	N	N
A32 D16	N	Y	N	N
A32 D32	N	Y	N	N

EO → Even, Odd Byte Access

The VMEbus BERR* (Bus ERRor) signal will be asserted if a quad byte access is initiated but address line A01 is set high.

The VMEbus lines used by the MVME215 are as follows:

Address Lines	A01–A31
Data Lines	D00–D31
Data Direction Line	WRITE*
Address Modifier Lines	AM0–AM5
Strobe Lines	DS0*, DS1*, AS*, LWORD*
Transfer Handshake Line	DTACK*
Interrupt Control Lines	BERR*
Additional Control Lines	SYSRESET*
Power Supply Lines	+5 V, +12 V, GND

All unused daisy-chained lines like the interrupt and bus grant lines are passed through the module. There is no interrupter on the module.

CMOS RAM DEVICES

The MVME215 is populated with 8K x 8 bit static CMOS RAM devices. (256Kb version) or 32K x 8-bit static CMOS RAM devices (512Kb and 1Mb version).

ADDRESS BUS INTERFACE

The MVME215 can be accessed in the standard addressing range (24-bit address) or the extended addressing range (32-bit address). The appropriate addressing range is determined by decoding the VMEbus address modifier lines.

Selection of the module's base address is provided in 1Mb, 512Kb, 256K, 128K, or 64K steps depending on the assembly option.

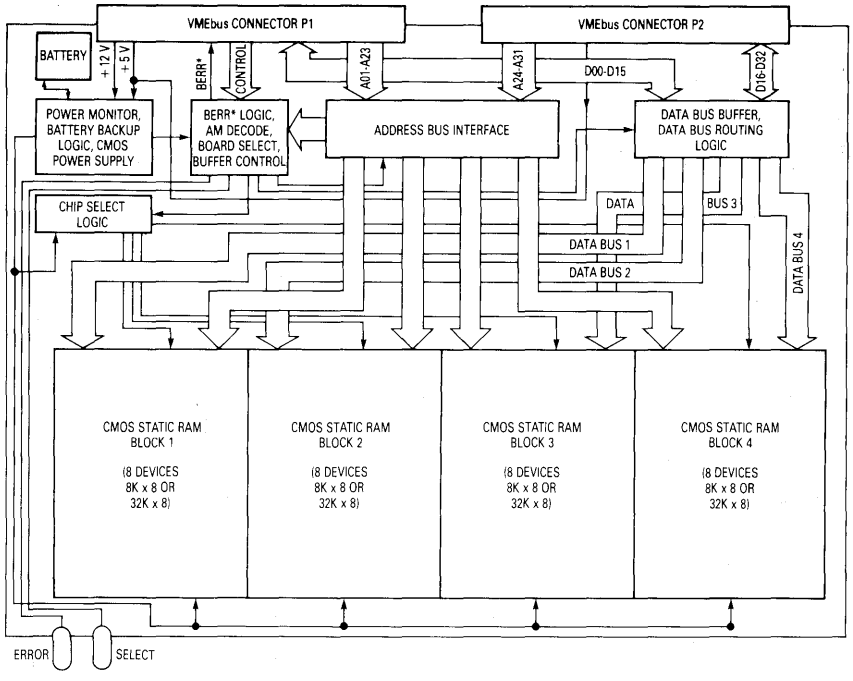


Figure 1. MVME215 Block Diagram

DATA BUS INTERFACE

Data transfer to or from the MVME215 can be byte-wide (8-bit), double byte-wide (16-bit), or quad byte-wide (32-bit). The appropriate data path width is decoded by means of the VMEbus signals DS0*, DS1*, A01*, and LWORD*.

ADDRESS MODIFIERS

The MVME215 is delivered with a programmed PROM installed in a socket to decode the address modifier codes. The module will respond if the following address modifier codes are asserted:

A24 slave — $39, $3A, $3D, and $3E A32 slave — $09, $0A, $0D, and $0E

As A24 slave, address lines A16 (A17, A18) through A23 are decoded to generate the module select signal. As A32 slave, address lines A16 (A17, A18) through A31 are decoded to generate the module select signal.

POWER MONITOR

The module has circuitry that monitors both the +5 V and +12 V supplies. When the 5 V supply decreases to 4.75 V or the 12 V supply decreases to 11.4 V the local power fail signal is asserted. The local power fail signal is also asserted by the VMEbus SYSRESET* signal.

Asserting the local power fail signal disables all RAM select lines.

BATTERY BACKUP CIRCUITRY

When power is lost or switched off, the information stored in the modules RAM devices is maintained. This is accomplished by means of a NiCad battery mounted on the module.

When the module is powered from the VMEbus, the battery is constantly charged to maintain its maximum capacity. When power is switched off, or the module is removed from the VMEbus power lines, information stored in the RAM will be maintained for about 2500 hours. Lithium batteries may be installed by the user instead of a NiCad battery to store the information in RAM for longer periods.

INDICATORS

There are two LEDs provided on the front panel of the MVME215. LED1 is switched on when the board is selected. LED2 is switched on if a bus error condition is detected by the MVME215. Any access to the module after the bus error condition has occurred will switch off the bus error LED.

2

CONNECTOR P1

Rows A, B, and C of connector P1 are used for address, data, and control signal lines.

CONNECTOR P2

Row B of connector P2 is used for power, address and data lines. Rows A and C are not used.

POWER LINES

The +5 V and +12 V power are supplied from the +5 V and +12 V pins of connector P1 and P2. All GND pins of connectors P1 and P2 are connected to the MVME215.

Power for the CMOS circuits and some pull-up resistors is provided by the battery back-up circuitry which consists of the battery charger, a separate power supply and logic which selects between stand-by and active mode. In the active mode, the CMOS circuits are supplied by a separate onboard power supply. To suppress CMOS latch-up, the power is derived from the +12 V power line. In the stand-by mode, the CMOS circuits are supplied by the onboard battery.

The NiCad battery is continuously charged while power is applied.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements	+5 Vdc +0.25/−0.125 V 1.3 A (typ) +12 Vdc +0.6/−0.36 V 30 mA (typ)
Operating Temperature	0° to +55°C
Storage Temperature	−20° to +45°C
Relative Humidity	0% to 90% (non-condensing)
Physical Dimensions Height Depth	 Double high Eurocard format 9.2 in. (233.6 mm) 6.3 in. (160 mm)
Connectors P1, P2	 96 Pin, DIN41612
Front Panel	4 TE, 6 HE
Battery	High-Temperature NiCad (as supplied) Lithium batteries (user option)
Capacity	256Kb, 512Kb, 1Mb options
Access time	230 ns (typ)
AS* → DTACK*	285 ns (max)
Input	VMEbus Rev. C, compatible
Output	VMEbus Rev. C, compatible Open-collector outputs (Isink max = 48 mA) Three-state outputs (Isink max = 48/64 mA)

ORDERING INFORMATION

Part Number	Description
MVME215-1	256Kb CMOS RAM Module. Includes User's Manual.
MVME215-2	512Kb CMOS RAM Module. Includes User's Manual.
MVME215-3	1Mb CMOS RAM Module. Includes User's Manual.

Notes: All modules are supplied with onboard battery.
For other memory size configurations contact your sales representative.

RELATED DOCUMENTATION

Part Number	Description
HB212/D	VMEbus Specifications Manual

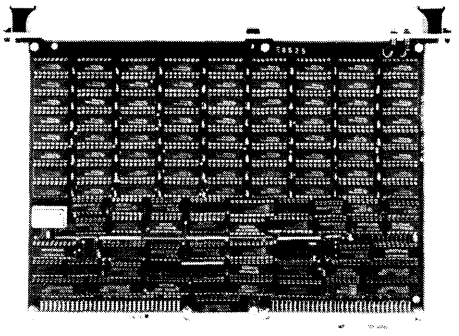
MVME225-1
MVME225-2

2

Product Preview

VMEmodule™ 1Mb/2Mb Dynamic RAM

- 8-/16-/32-Bit Data Bus Width
- A24/A32 Address Bus Width
- Byte Parity Generation and Checking
- Base Address Selectable in 64Kb Steps
- Interleave mode
- PROM Address Modifier Decoder
- 280 ns Read Access Time (with parity)
- 90 ns Write Access Time
- VMEbus Compatible



The MVME225-1 and MVME225-2 are high speed, high performance VMEmodules in double Eurocard format. They provide in a single module 1Mb or 2Mb of dynamic random access memory for 8-, 16-, or 32-bit data transfers, respectively.

Self-contained parity generation and check on all stored data bytes are features of both modules.

A two-way interleaving mode may be selected to decrease the effective memory write access time.

Figure 1 shows the block diagram of the MVME225 modules. The modules consist of the VMEbus interface timing and refresh control circuitry, board select logic, logic to generate and check parity for each storage block and four RAM storage blocks with a maximum capacity of 512Kb each.

The start address of the module is jumper selectable in 64Kb steps over a selected 16 megabyte partition of the four gigabyte address range. The module responds to 24- or 32-bit addresses depending on the Address Modifier Code. A LED on the front panel informs the user when a read or write access to the memory board takes place. For fast memory access the modules have a two-way interleave mode.

For each byte written to memory, parity is generated during the access and stored in the associated parity bit. For each byte read from memory, parity is generated and compared with the stored parity bit.

If a parity error is detected, this error is latched and displayed on the front panel ERROR-LED and the access cycle is terminated by assertion of the bus error signal (BERR*).

The VMEbus BERR* signal is also asserted (and the ERROR-LED will be switched on) if a quad byte access is initiated but address line A01 is set high.

The onboard refresh logic refreshes the data in a transparent mode.

VMEbus INTERFACE

The modules are designed for use as slaves in a VME environment which complies with VMEbus Specification, Revision C.

SLAVE MODULE OPTIONS:

	BLT Block Transfer Capability	RMW Read-Modify-Write Capability	UAT Unaligned Transfer Capability	ADO Address-Only Capability
A24 D08 (EO)	N	Y	N	N
A24 D16	N	Y	N	N
A24 D32	N	Y	N	N
A32 D08 (EO)	N	Y	N	N
A32 D16	N	Y	N	N
A32 D32	N	Y	N	N

EO → Even, Odd Byte Access

Signals used:

Address Lines	A01–A23, A24–A31
Data Lines	D00–D15, D16–D31
Address Modifier Lines	AM0–AM5
Control Lines	DS0*, DS1*, AS*, WRITE*, BERR*, DTACK*, LWORD*, SYSRESET*

All unused daisy-chain signals are routed from the input pin to the output pin.

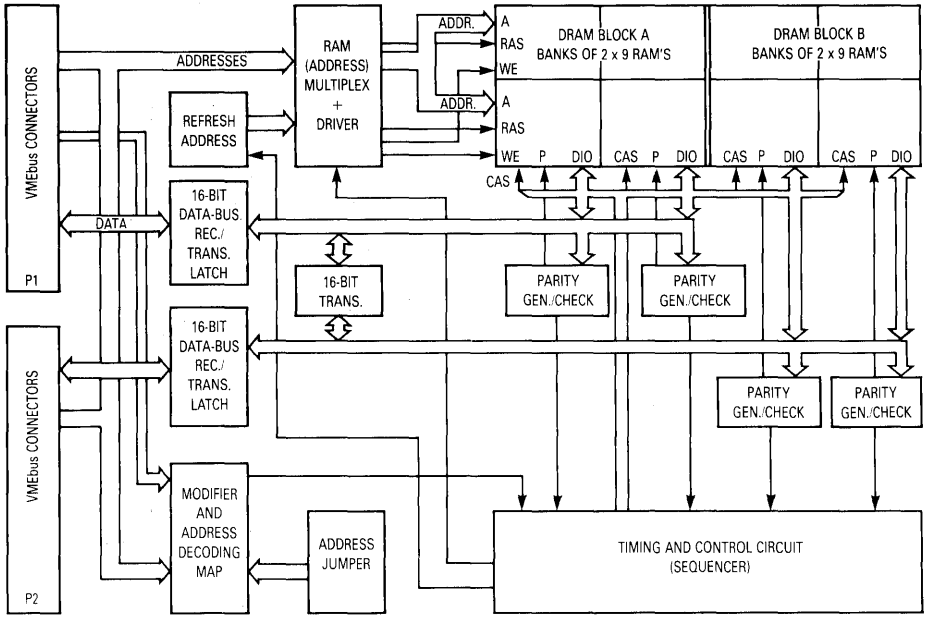


Figure 1. MVME225 Block Diagram

24-/32-BIT ADDRESSING

The 24- or 32-bit address bus width is determined by address modifier codes according to the VMEbus specification. The 32-bit address bus width is selected by assertion of AM-codes $09, $0A, $0D, and $0E. 24-bit address bus width is selected by assertion of AM-codes $39, $3A, $3D, and $3E. When a 24-bit address bus width is selected, address lines A24 to A31 are irrelevant. The AM-codes are stored in a PROM.

START ADDRESS SELECTION

A module start address can be set on any 64Kb boundary over a user-selected 16Mb range.

Using unique start addresses 32 (8) memory modules can be supported in a 16Mb VMEbus system. In an A32 VMEbus system (4Gb) this number can be multiplied by 256.

With use of the address modifier codes this number can be increased or several modules can have the same start address.

INTERLEAVE MODE

The interleave is accomplished by steering memory access to pairs of memory modules on an alternating basis requiring memory modules using this mode to exist in pairs. This allows a master access to one of them, while the other module is completing the present cycle. Interleaving can enhance the memory write throughput.

Depending on the system environment (D16 or D32) a double byte or quad byte two-way interleave can be selected by the user.

ACCESS AND PARITY ERROR HANDLING

If a parity error occurs during a read access cycle or an access error occurs, the VMEbus BERR* signal is asserted. This error condition is also indicated by switching the front panel ERROR-LED on. The next error-free access switches the ERROR-LED off.

MEMORY ARRAY

The memory array consists of two 36 device blocks, each of them physically arranged on the module in two banks of 18 RAMs. The MVME225-2 uses four banks of 256K-bit devices for a capacity of 2Mb. The MVME225-1 uses two banks of 256K-bit devices for a capacity of 1Mb. Each bank consists of two rows of nine RAMs comprising one byte plus a parity bit.

DATA PARITY GENERATION/CHECKING

Whenever data is written to the memory module, one parity bit for the low order byte and one for the high order byte for each double byte are generated by the parity generator/checker and stored in memory.

When a read cycle is performed, new parity bits are generated and compared with the old ones.

In the case of a parity error the BERR* signal is asserted on the VMEbus, otherwise DTACK* is generated after the data set up time.

This parity error signal switches the front panel ERROR-LED on. The next error-free board access switches the ERROR-LED off.

By jumper, parity checking can be disabled and, for test purposes, the parity generator can be forced to write a wrong parity bit in all selected memory banks.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements	+5 Vdc
Operating Temperature	0° to 55°C
Storage Temperature	−40° to +85°C
Relative Humidity	0% to 90% (non condensing)
Physical Dimensions Height Depth	 Double High Eurocard format 9.2 in. (233.5 mm) 6.3 in. (160 mm)
Connectors P1, P2	96 PIN, DIN41612
Front Panel	4TE, 6HE
Capacity	612Kb or 2Mb
Read Access Time	280 ns (with parity)

ORDERING INFORMATION

Part Number	Description
MVME225-1	VMEmodule 1Mb Dynamic RAM with Byte Parity. Includes User's Manual.
MVME225-2	VMEmodule 2Mb Dynamic RAM with Byte Parity. Includes User's Manual.

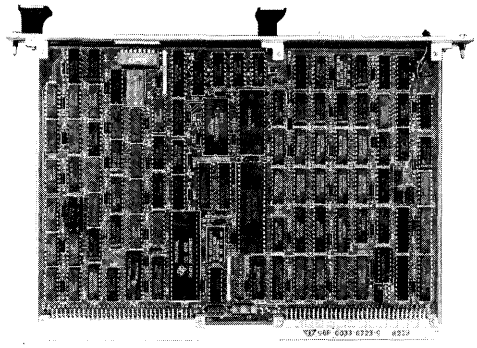
RELATED DOCUMENTATION

Part Number	Description
HB212/D	VMEbus Specification Manual

MVME300

2

VMEmodule GPIB Controller With DMA



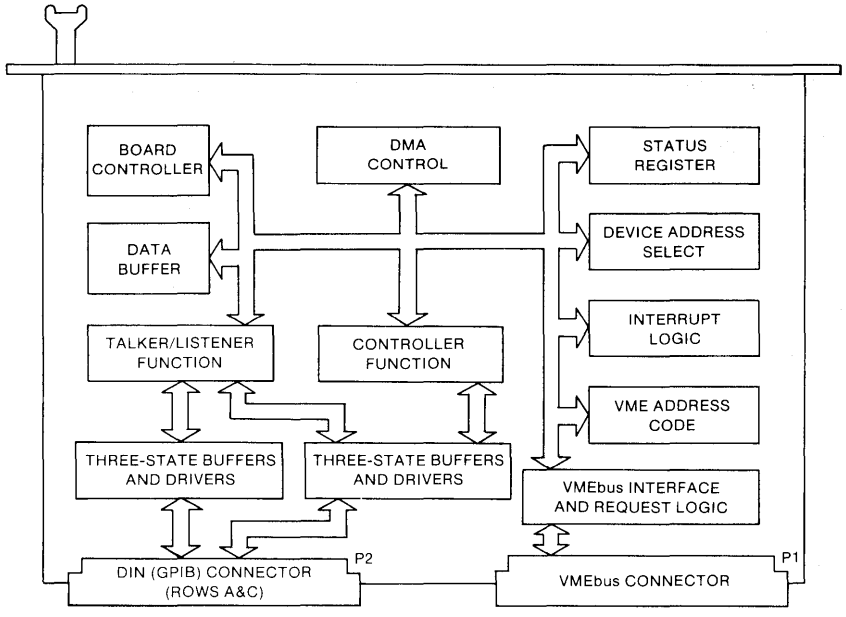
- Meets Complete IEEE 488-1978 Standard and the IEEE 488-1980 Supplement
- VMEbus Compatible
- Functions Under Master or Slave Configurations on VMEbus
- Complete Controller, Talker, and Listener Capability
- DMA Interface for High Speed Transfers
- Transfer Data Under DMA or Non-DMA Mode
- Supports up to 0.5 Megabyte/Sec Data Transfer Rate on the GPIB Interface
- 256/1K Byte FIFO Buffer
- Uses State-of-the-Art GPIA Device - TMS9914A GPIB Adapter
- Programmable Interrupt Levels/Vectors - MC68153 Bus Interrupter Module
- Programmable End-of-String Character
- Jumper Selectable VMEbus Base Address
- Accessible Via a Set of 32 On-Board Registers
- 6-Bit Module Status Register
- Writable System Fail LED
- Intelligent Bus Requester for Optimum Throughput (Discrete Logic)
- Double High Eurocard Form Factor
- 0°C-70°C Operating Temperature Range

The MVME300 GPIB Controller VMEmodule facilitates interface of a VME system to any equipment on the IEEE standard 488-1978 General Purpose Instrumentation Bus. The complete Controller, Talker, and Listener functions of the standard and a DMA interface are provided. The functional blocks of MVME are shown in Fiqure 1.

To a host on the VMEbus, the MVME300 GPIB Controller VMEmodule (GPIBC) appears as 32 adjacent 16-bit read/write registers with which data transactions are performed using the odd bytes only.

A front panel accessible DIP switch is provided for selection of the GPIB base address. The DIP switch is also used in conjunction with the software driver to enable/disable GPIB talker/listener functions and for configuring the module as the GPIB system controller.

FIGURE 1 — MVME300 Functional Block Diagram



ADDRESS MODIFIER REGISTER

A six bit read /write register is provided on the GPIBC for use in DMA operations. Prior to the operation, the register is loaded to identify the VMEbus address modifier lines that will be asserted while memory is accessed.

RESET REGISTER

This write only register may be loaded with any value to obtain a hardware reset of all registers except the bus release count register (VBRCR), the memory address register (MAR) and certain registers in the TMS9914A GPIB Adapter chip which must be reset via software command. The signal generated as a result of writing a value to the reset register is ORed with the VMEbus signal SYSRESET*. The user should refer to the data sheet for the GPIB Adapter chip to identify the non-hardware resettable registers therein.

GPIBC STATUS REGISTER

A six-bit, read only controller status register (CSREG) is used to convey the status of GPIBC activities. Register contents are not cleared by a read operation. The significance of each of the six bits is as follows:

Bit	Description
S0	FIFO empty
S1	GPIBC in listen mode (EODNE): EOS received or EOI detected and FIFO is empty. Bit is cleared by a GPIBC reset or SYSRESET*
S2	FIFO full
S3	Byte count has been reached during DMA (BCRDN). This bit is cleared by writing hex 20 to the Uniline Command Register (UCR)
S4	GPIBC System Fail LED is set (SYSFAIL). The bit is set during power on or by writing hex 10 to the UCR. The bit is cleared by a GPIBC reset command.
S5	Set during DMA by a bus error (BERR) signal from a memory board due to an invalid memory access. The bit is cleared by writing hex 20 to the UCR or by SYSRESET*

2

THE IEEE-488 GENERAL PURPOSE INSTRUMENTATION BUS

In general terms, the purpose of the IEEE-488 General Purpose Instrumentation Bus (GIPB) is to facilitate bi-directional asynchronous intercommunications among digital data utilizing and producing equipment. More specifically, the GPIB was devised to provide a standard interface to ease the task of achieving intercommumications in an instrumentation system configured from available programmable devices having different ports. The GPIB attains this goal by defining the interface structure through and by means of which data is transferred. To communicate with other apparatus connected to the instrumentation bus, a device must operate through an interface that complies in part or fully with this standard.

The standard describes a byte serial, bit parallel medium speed bus intended for a maximum of 15 devices communicating over a limited distance. An 8-bit parallel bidirectional data bus with eight additional control lines is described. Three lines control transfer of the data byte; five are used for general interface management.

Table 1 lists the GPIB functions provided by MVME300.

DMA OPERATIONS

The GPIBC design provides for DMA operation in either direction under local (VMEbus host) or remote (GPIB active controller) control. A DMA read operation (talker function) can be initiated locally by the host processor or remotely by the active controller. A DMA write operation (listener function) can also be initiated locally or remotely.To facilitate VMEbus request and release, DMA data is transferred by the GPIBC through a 256/1k-byte FIFO type buffer.

DMA logic utilizes three registers dedicated to DMA and three multiple purpose registers. The three dedicated registers include:

- Three 8-bit registers comprising a 24-bit read/write memory address register(MAR). Prior to a DMA operation, the starting memory address is loaded by user software into this register. The starting address is incremented by the GPIBC after transfer of each data byte.

TABLE 1 — MVME300 IEEE 488-1978 Functions

Capability Code	Name	Description
SH1	Source Handshake	Initiate, control and terminate asynchronous transfer of data byte to device having acceptor handshake capability.
AH1	Acceptor Handshake	Allow initiation, continuation or termination of data byte transfer requested by device having source handshake capability.
T5, TE5	Talker (T. Extended)	Transfer device dependent data and serial poll status data.
L3, LE3	Listener (L. Extended)	Receive device dependent data and status data.
SR1	Service Request	Asynchronously request service from the controller-in-charge and synchronize transfer of message.
RL1	Remote Local	Select interface (remote) or front panel (local) control information source.
PP1, PP2	Parallel Poll	Transfer to the requesting controller a parallel poll response over the assigned signal line.
DC1	Device Clear	Receive request to initialize self.
DT1	Device Trigger	Receive request to initiate self function. Device group receive request to initiate self functions.
C1-C4, C9	Controller	Be system controller. Receive/pass control. Take control synchronously. Respond to a service request from a controller. Conduct a parallel poll.

- Two 8-bit registers (BCR1, BCR2) forming a 16-bit byte count register that is loaded by the user prior to DMA with the number of bytes to be transferred (1's complement) and is automatically incremented on transfer of each byte. The byte count register's 16-bit capacity allows single operation transfer of up to 65,536 bytes.
- Four write only bits forming a VMEbus release count register (VBRCR) whose function, on GPIBC receipt of a bus clear from a device of higher priority than the current DMA device, is to hold the bus until the specified number of bytes have been transferred. Prior to DMA, a jumper selected count number (0 through 15) is loaded into the register.

The three multipurpose registers used in DMA operations include:

- An 8-bit read/write FIFO data I/O register (FDIOR) through which data is transferred during DMA between VMEbus and the GPIB when the GPIB is functioning as bus master. The register may also be used by the host processor for diagnostic purposes.
- An 8-bit write only end-of-string-character compare register (ESCCR) into which the user program may load a selected character for use in terminating data transfer. When enabled during listen mode, end-of string (EOS) logic on detecting a character match provides a signal for use in generating a DMA done interrupt when the FIFO becomes empty.
- A 6-bit write only uniline command register (UCR). As the last step in DMA initiation, the user program sets the DMA write or DMA read bit in this register, starting the transfer. For applications using an EOS character detection scheme to terminate data transfer in listen mode, the UCR enable EOS bit may be set to enable the EOS character comparison logic. For applications using a non-EOS approach to terminate data transfer in the talk mode, the EOI line is set true by programming the TMS9914A.

VMEbus INTERRUPTER

The GPIBC interrupt requesting logic utilizes an MC68153 Bus Interrupter Module (BIM). The BIM is an LSI device implemented using the TTL-compatible MCA1300ALS Macrocell Gate Array and is housed in a 40-pin DIP package. It is capable of handling interrupt requests from four sources, of generating an interrupt on any of the seven VMEbus IRQ lines, or interrupt acknowledge cycle response and proper release of an IRQ line.

The BIM contains eight 8-bit read/write registers. Four registers can be programmed with status/ID byte or with a vector number pointing to an interrupt service routine. Each of the four control registers can be programmed, for an interrupt source, with the desired interrupt request level, to enable or disable the interrupt source and to select transfer of an internal vector previously written in the corresponding vector/status-ID register or transfer of a vector from the interrupting device. A bit is provided in each control register for semaphore-like synchronization of communications between processors in a multi-processor system. A bit is also provided for selection of automatic de-allocation of that register following interrupt acknowledge.

On the GPIBC, the BIM processes interrupts from just three sources. One interrupt input is used to accommodate those interrupts generated by the TMS9914A GPIB Adapter in its regulation of the talker, listener and controller device functions. Another BIM interrupt input is connected to the DMA control logic to process interrupts generated on completion of DMA operations. The third BIM interrupt imput is dedicated to processing interrupts generated by the GPIBC on detection of a bus error signal from a slave on the VMEbus. These can occur in the memory read/write cycle during DMA.

GPIBC SOFTWARE DRIVER

A software driver is available for use in applications utilizing a VERSAdos 4.3 or higher Operating System. The driver utilizes the VERSAdos Input/Output handler (IOS).

To accommodate the widest variety of applications in a multiuser multitasking environment, the GPIBC driver provides two major modes of operation which treat the total GPIB system as a shared resource. To gain an understanding of these modes, it is important to understand the rules for access to a GPIB and to devices on the bus. Consider a system having a single MVME300 and several devices on the GPIB.

When the system is booted, no assignments to devices on the bus or to the bus exist. The bus is in the unassigned mode.

On the first assignment, one of two driver modes is entered; session-exclusive mode if the assignment is to the bus itself, or shared-bus mode if the assignment is to a device on the bus.

To enter the session-exclusive mode, a task must make an exclusive assignment to the bus itself. A public assignment to the bus is not allowed. Once in the mode, the task and other tasks in the same session can make exclusive or public assignments to devices on the bus. All assignments to all devices on the bus must be closed before the assignment to the bus itself can be closed causing a return to unassigned mode.

If the initial assignment (when neither the bus nor any device on the bus is currently assigned) is to a device on the bus rather than the bus itself, the shared bus mode is entered. Any task in any session can make exclusive or public assignment to any device on the bus. But no task is allowed to assign the bus itself. Closing all assignments to devices on the bus causes a return to unassigned mode.

The GPIBC driver modes are depicted in Figure 2. Figure 3 shows the relationship of the GPIBC driver to the VERSAdos Operating System and a GPIB system comprised of a number of subsystems each having one MVME300 Controller and up to 14 additional devices which could also be GPIBC or other intelligent devices.

In a GPIB subsystem having two or more controller devices, one must be given exclusive control of the REN and IPC lines in that subsystem so that it can act as system controller. A dual inline switch is provided on the MVME300 for configuring the module as system controller and for setting its primary GPIB address. Devices in the overall system are made known to the driver via the system generation utility and system initialization.

Nine of the data transfer and command functions provided by the IOS module which performs all of the general physical I/O under the VERSAdos Operating System are supported by the GPIBC driver. The nine are listed in Table 2 opposite codes 00, 01 and 80. Also shown are the additional GPIB capabilities the driver provides. These are shown opposite codes 40 in five groups of similar functions: command I/O, secondary address I/O, device status commands, bus control commands, and device control commands. Under the VALID IN heading, the two rightmost columns identify the operating mode in which the subsystem controller must be for a user task to obtain the corresponding function.

FIGURE 2 — GPIBC Driver Modes

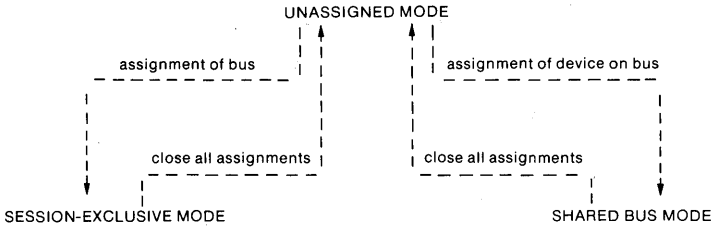


FIGURE 3 — GPIB System Under VERSAdos

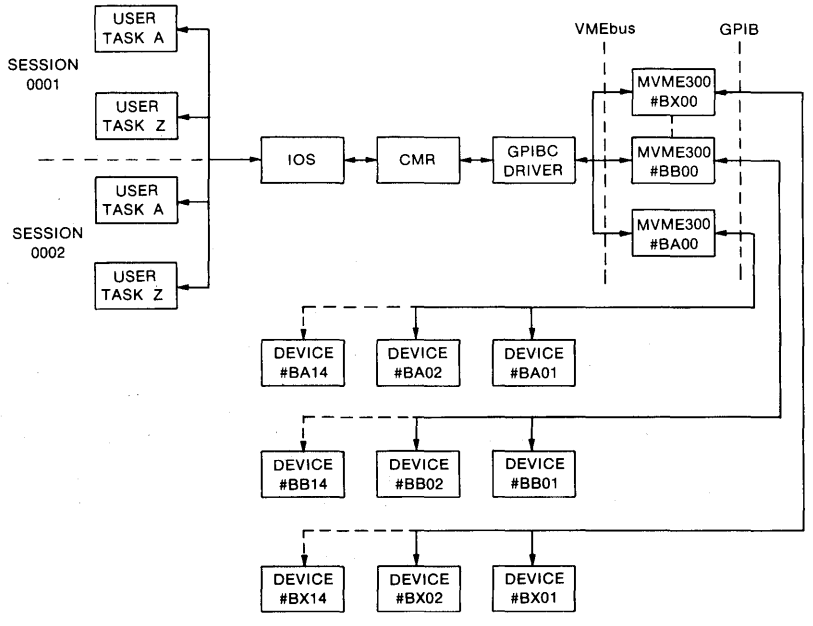


TABLE 2 — GPIBC – Supported IOS Commands

Function Name	Code (HEX)	Function (HEX)	Valid In	
			Controller	Talker/Listener
READ	00	01	YES	YES
WRITE	00	02	YES	YES
OUTPUT W/ INPUT	00	04	YES	YES
TEST I/O	01	04	YES	YES
WAIT	01	08	YES	YES
HALT I/O	01	10	YES	YES
REQUEST DEVICE STATUS	01	40	YES	YES
CONFIGURE DEVICE	01	80	YES	YES
CHANGE DEFAULT CONFIG	80	02	YES	YES
COMMAND OUT/DATA OUT	40	10	YES (1)	NO
COMMAND OUT/DATA IN	40	11	YES (1)	NO
COMMAND OUT	40	12	YES (1)	NO
READ SECONDARY ADDR	40	20	YES	YES
WRITE SECONDARY ADDR	40	21	YES	YES
UNCONFIG PARALLEL POLL	40	30	YES (1)	NO
CONFIG PARALLEL POLL	40	31	YES	NO
CONDUCT PARALLEL POLL	40	32	YES	NO
SET PARALLEL POLL STAT	40	33	NO	YES
SET SERIAL POLL STAT	40	34	NO	YES
CONDUCT SERIAL POLL	40	35	YES	NO
DISABLE PARALLEL POLL	40	36	YES (1)	NO
INTERFACE CLEAR	40	40	YES (1, 2)	NO
REMOTE ENABLE	40	41	YES (1, 2)	NO
REMOTE DISABLE	40	42	YES (1, 2)	NO
CLEAR ALL DEVICES	40	44	YES (1)	NO
CLEAR ALL LISTENERS	40	45	YES (1)	NO
LOCKOUT ALL DEVICES	40	46	YES (1)	NO
PASS CONTROL	40	47	YES (1)	NO
LOCAL ALL LISTENERS	40	48	YES (1)	NO
TRIGGER ALL LISTENERS	40	49	YES (1)	NO
ABORT BUS & ALL DEVICES	40	4A	YES (1, 2)	NO
ABORT ALL BUS MESSAGES	40	4B	YES (1)	NO
TAKE CONTROL	40	4C	YES (1)	NO
CLEAR A DEVICE	40	50	YES	NO
REMOTE A DEVICE	40	51	YES	NO
TRIGGER A DEVICE	40	52	YES	NO
LOCAL A DEVICE	40	53	YES	YES

NOTES: (1) Must be active controller (2) Must be the system controller

VMEbus INTERFACE

VMEbus is characterized by the asynchronous bidirectional operation required for complex, high-performance systems. The VMEbus interface provided on the MVME300 module supports DMA operation, operation in a multiprocessor system and the full 16-megabyte address range of the MC68000 MPU. Pins for the VMEbus address, data and control lines are provided in the triple row, 96-pin VMEbus connector P1. VMEbus signal and connector physical requirements are fully described in MVMEBS-VMEbus Specification Manual.

GPIB INTERFACE

A standard DIN double-row, 64-pin male connector, P2, accommodates physical connection to the GPIB. A flat ribbon cable of the required length one end of which provides the corresponding female half of connector P2 and the other end of which provides a 24-pin female GPIB connector is used to make the actual physical connection to the instrumentation bus.This 24-pin female connector complies with the signal and physical requirements described in the IEEE-488 standard. Table 3 lists the corresponding pin numbers and signal descriptions for both female connector P2 and the female GPIB connector on the other end of the ribbon cable.

TABLE 3 — Pin Assignments and Signal Descriptions – GPIB Connector P2

GPIB Connector Pin Number	P2 Connector Pin Number	Signal Mnemonic	Signal Name/Description
1	C1	DIO1	Data Line 1
13	A1	DIO5	Data Line 5
2	C2	DIO2	Data Line 2
14	A2	DIO6	Data Line 6
3	C3	DIO3	Data Line 3
15	A3	DIO7	Data Line 7
4	C4	DIO4	Data Line 4
16	A4	DIO8	Data Line 8
5	C5	EOI	End or Identify
17	A5	REN	Remote Enable
24	A12	GND	Logic Ground, Return for EOI and REN
6	C6	DAV	Data Valid
18	A6	GND	Return for DAV
7	C7	NRFD	Not Ready for Data
19	A7	GND	Return for NRFD
8	C8	NDAC	Not Data Accepted
20	A8	GND	Return for NDAC
9	C9	IFC	Interface Clear
21	A9	GND	Return for IFC
10	C10	SRQ	Service Request
22	A10	GND	Return for SRQ
11	C11	ATN	Attention
23	A11	GND	Return for ATN
12	C12	GND	Connector Shield

Mechanical and Environmental Specifications

Characteristics	Specifications
Power Requirements	+5 Vdc at 2.5 A, Typical
Temperature • Operating • Storage	 0° C to 70° C -55° C to +85° C
Relative Humidity	0-90% (non-condensing)
Physical Characteristics PC Board Only • Height • Depth • Thickness	 9.2 in. (233 mm) 6.3 in. (160 mm) 0.63 in. (16.0 mm)
PC Board with Front Panel • Height • Depth • Thickness	 10.3 in. (262 mm) 7.4 in. (188 mm) 0.8 in. (20.3 mm)
PC Board Form Factor	Double High Eurocard

Ordering Information

Part Number	Description
MVME300	VMEmodule GPIB Controller with DMA. Provides IEEE-488 Listener, Talker and Controller functions. Cable with standard IEEE-488 (mechanical) connector provided for connection to GPIB. Includes User's Manual.
MVME300/D	VMEmodule GPIB Controller with DMA User's Manual.

Related Documentation — Hardware

Part Number	Title
MC68000UM	MC68000 Microprocessor User's Manual
MP033A	TMS9914A GPIB Controller Data Manual
NP-151	MC68153 Bus Interrupter Module Data Sheet
HB212/D	VMEbus Specification Manual
IEEE STD 488-1978	Digital Interface for Programmable Instrumentation
IEEE STD 488A-1980	Supplement to Above Standard

Related Documentation — Software

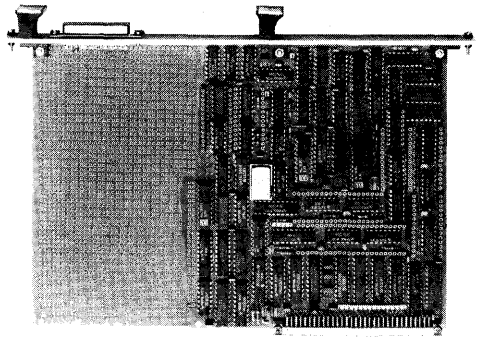
Part Number	Title
M68KVOVER/D	VERSAdos Overview
M68KRMS68K/D	M68000 Realtime Multitasking Software User's Manual
RMS68KIO/D	VERSAdos Data Managment Services and Program Loader User's Manual
M68KSYSGEN/D	System Generation Facility User's Manual
MVME3SW	MVME300 GPIB Controller with DMA Software Manual

Other Double Eurocard Form Factor VMEmodules

Part Number	Description
MVME101	VMEmodule Monoboard Microcomputer
MVME110-1	VMEmodule Monoboard Microcomputer
MVME200/201	VMEmodule 64K/256K Byte Dynamic RAM
MVME211	VMEmodule Static RAM/ROM/EPROM/CMOS RAM
MVME310	VMEmodule Universal Intelligent Peripheral Controller
MVME315	VMEmodule Intelligent Floppy Controller/SASI Interface

MVME310

VMEmodule Intelligent Controller for Custom Interfacing



- Based on MC68121 Intelligent Peripheral Controller
- Standalone debugger supporting:
  - — Four Channel DMA Controller
  - — Interrupt Controller
  - — Serial download from host
  - — Checksum self test
- Large Wirewrap Area (3.5″ x 6.3″) for Prototype Construction
- Local Bus Interface Signals Available at Header Locations
- Serial Port Connector
- Two Sockets Expandable to Four for up to 32K Bytes Dual Ported RAM Buffer, 4K Bytes Static RAM Supplied
- Two sockets for up to 16K Bytes Program ROM/PROM. One Socket Static RAM Configurable
- Double Eurocard Form Factor
- VMEbus Compatible
- 0°C–70°C Operating Temperature Range

The MVME310 is a double high Eurocard module which is used for developing the prototype for a custom interface to a VMEbus peripheral. It provides a large wirewrap area, has interrupt and DMA controller devices and, to facilitate development of executive firmware, is based on the MC68121 Intelligent Peripheral Controller.

MVME310 provides two sockets for 16K of byte wide, dual ported static RAM buffer (4K bytes are supplied and two additional sockets in the user area can be used for buffer expansion) and two sockets for up to 32K bytes of user program in ROM/PROM. One of these sockets is configurable for 8K bytes of static RAM. One PROM socket contains the 4K byte standalone debugger — IPCbug. A serial port connector is also provided. Figure 1 is a block diagram of MVME310.

One example of the kinds of interfaces that can be implemented using as a basis MVME310 with IPCbug is found in the MVME315 Intelligent Floppy Controller/SASI Interface. This module expanded the basic MC68121 functions into a universal intelligent peripheral controller (UIPC) function which provided a means of implementing an interface using a standard protocol for host/MVME315 communications and for development of control firmware. The control programs for the UIPC, the floppy disk controller, and the SASI interface are contained in two 8K PROMS on MVME315.

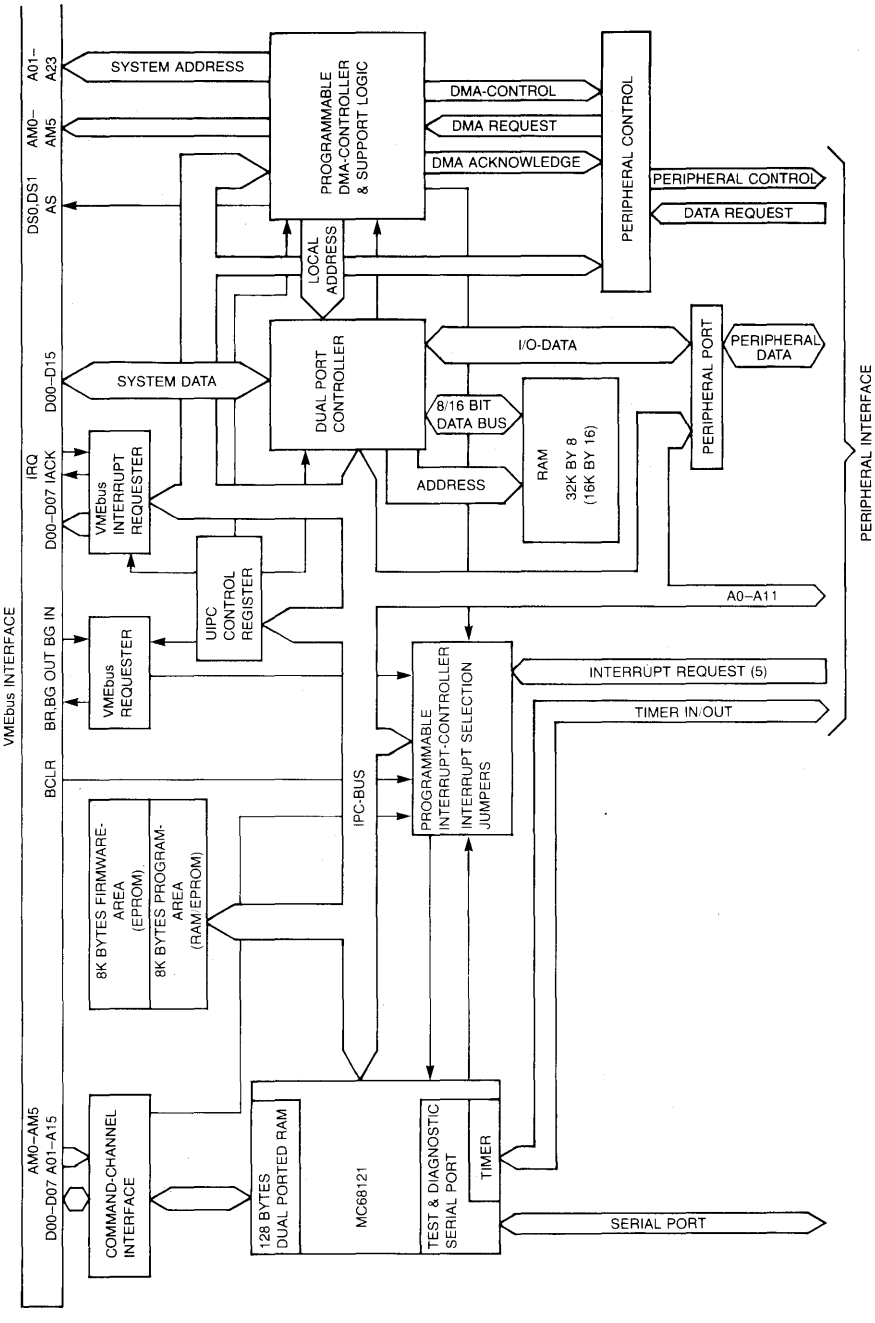


FIGURE 1 — MVME310 Functional Block Diagram

PERIPHERAL INTERFACE SUPPORT

The MVME310 offers the user the basic hardware facilities for implementing one or more peripheral interfaces and for developing the required control firmware. Access to the local bus is provided by a dual 28-pin row of plated through holes for mounting two headers. Bus signals available at the header locations are listed in Table 1A and Table 1B.

To accomodate the timing requirements of the various available interface chips, the MVME310 has a programmable logic array which, with additional user-supplied hardware, can be used to adjust the timing of the accesses from the on-board IPC.

TABLE 1A — Header Position K4 Hole Assignments and Signal Names

Pin Number	Signal Name	Pin Number	Signal Name
1	LATCH*	15	DREQ3*
2	NOT USED	16	DACK1
3	(2XE)	17	DACK3
4	NOT USED	18	IA1
5	TOUT	19	IA3
6	IPCRW	20	IA5
7	NOT USED	21	IA7
8	PRD*	22	IA9
9	AIOEN*	23	IA11
10	IIRQ4*	24	I/O1
11	IIRQ6*	25	I/O3
12	(TXC)	26	I/O5
13	DREQ0*	27	I/O7
14	DREQ2*	28	+5.0 Vdc

TABLE 1B — Header Position K5 Hole Assignments and Signal Names

Pin Number	Signal Name	Pin Number	Signal Name
1	DACK0*	15	−12 Vdc
2	DACK2*	16	I/ODMA
3	IA0	17	NOT USED
4	IA2	18	NOT USED
5	IA4	19	TIN
6	IA6	20	E
7	IA8	21	SRESET*
8	IA10	22	PWR*
9	I/O0	23	IDEN*
10	I/O2	24	IIRQ1*
11	I/O4	25	IIRQ5*
12	I/O6	26	IIRQ7*
13	GND	27	PREADY
14	+12 Vdc	28	DREQ1*

DMA SUPPORT

For DMA capability, MVME310 has an AMD Multimode DMA Controller-AM9517-5. This device has four independent DMA channels each with separate registers. To facilitate the development of control firmware, IPCbug offers three commands: DMA Controller Status Display (DS); DMA Control Register Modify (DM); and Initiate and Execute DMA transfer (DX). Provided RAM and available sockets are sufficient for a buffer to accomodate differences in VMEbus and peripheral data transfer rates.

BUS REQUEST LEVEL SELECTION

The MVME310 has three headers for jumper selection of one of the four VMEbus request levels.

INTERRUPTER LEVEL SELECTION

The MVME310 has two headers for jumper selection of one of the seven VMEbus interrupt priority levels.

ENABLING THE SYSTEM FAIL FUNCTION

The MVME310 has a header for jumper connection of the board fail timer circuit to the VMEbus SYSFAIL* line.

DATA AND PROGRAM MEMORY DEVICE SIZE AND TYPE SELECTION

The MVME310 has five headers by means of which four data and program memory sockets are jumper configured for the supported memory device sizes and types. Two headers provide for jumper configuration of the two dual ported local RAM sockets for 2K x 8 or 8K x 8 devices. This also configures the two data memory socket locations in the user wirewrap area. Three headers allow jumper configuration of the program memory sockets. Two of these are used for jumper configuration of the low program memory block for 2K, 4K or 8K ROM devices. The other header is used for configuring the high program memory block for a 2K, 4K or 8K RAM/ROM device.

Interrupt Source Selection

Five signals are brought directly to the on-board interrupt controller from the peripheral interface. The MVME310 also has a header for jumper connection to the on-board interrupt controller of two additional interrupt sources. These can be selected from the VMEbus signals BCLR*, IRQ6* and IRQ7* and the local signal IPCWR*.

Base Address Selection

For command channel accesses, the MVME310 has a header for jumper selection of a base address. This can be set on any 512 byte boundary throughout the 65,536 byte VMEbus short I/O address space.

Serial Communications

The MVME310 has a 25-pin sub-D female connector for implementing a serial data function in order to connect a terminal. Firmware support for a serial function is provided by the LO command of IPCbug which provides download from a host of a user program in Motorola S-Record format.

Users can provide their own serial communications firmware using the support offered by the on-board MC68121 microcomputer for full duplex, asynchronous communications. These are:

- The rate and mode control register
- The transmit/receive control and status register
- The transmit data register
- The receive data register

The basic data rate can be obtained from an external source or the 16-bit internal timer. Subsequently, operation at one of four baud rates is program selectable. Data format can be NRZ or bi-phase.

Usage

The MVME310 will operate as a slave in VMEbus systems having any of the listed bus masters:

MVME101 VMEmodule Monoboard Microcomputer
MCME110-1 VMEmodule Monoboard Microcomputer
MVME115 VMEmodule Monoboard Microcomputer

MVME310 SPECIFICATIONS

The specifications for the MVME310 are listed in Table 2.

TABLE 2 — MVME310 Specifications

Characteristic	Specification
Power Requirements	+5.0 Vdc ± 5% at 2.5 A (typical) +12 Vdc ± 5% −12 Vdc ± 5% at 2.5 A (typical)
Environmental Requirements Operating Temperature Storage Temperature Humidity Range	 0° to 70°C −55°C to +85°C 0% to 95% (non-condensing)
Mechanical Specifications Height x Depth (board) Height x Width (front panel)	 9.2″ (234 mm) x 6.3″ (160 mm) 10.3″ (262 mm) x 0.79″ (20 mm)
Connectors VMEbus Serial Connector	 D1N41612 C96 25-pin D-Subminiature

Software Driver

For users desiring to write their own device driver, a manual, Guide to Writing Device Drivers for VERSAdos, M68DRVGD/D1, detailing how to write a driver that runs under the M68000 Real-time Multitasking kernel or under VERSAdos is available.

VMEbus Interface

The MVME310 has a VMEbus interface which decodes all address and address modifier lines and provides bus requester and bus interrupter functions.

VMEbus Connector P1

The electrical and mechanical characteristics of VMEbus connector P1 are fully described in the VMEbus Specification Manual MVMEBS/D1.

Serial Connector

The serial connector is a Sub-D female, TMC-5-3-25, or equivalent.

Ordering Information

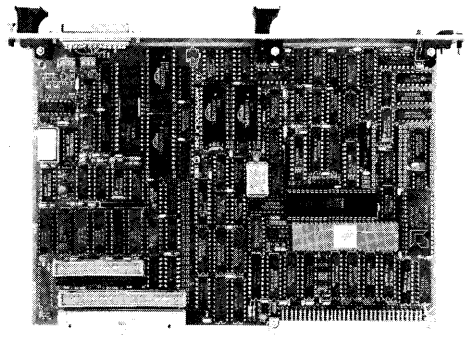
Part Number	Description
MVME310	VMEmodule Intelligent Controller For Custom Interfacing. Includes User's Manual.
MVME310/D	VMEmodule Intelligent Controller for Custom Interfacing User's Manual

Related Documentation

Part Number	Description
HB212/D1	VMEbus Specification Manual
M68DRVGD/D	Guide to Writing Device Drivers for VERSAdos
M68IPCS/D	M68000/IPC Command Channel Software Interface Reference Manual

MVME315

VMEmodule Intelligent Floppy Controller/ SASI Interface



- Controls up to Four 8" or 5-1/4" Floppy Disk Drives
- Provides Intelligent Interface to SASI Bus
- Supports 4-Channel DMA
- Provides RS-232-C Serial Port
- Supports Single and Double Density Formats for Single and Double-Sided Drives
- Based on MC68121 Intelligent Peripheral Controller
- VERSAdos 4.3 Support
- Double Eurocard Form Factor
- VMEbus Compatible
- 0°C–70°C Operating Temperature Range

The MVME315 combines in one module a multiple floppy disk controller, a SASI bus interface and a universal intelligent peripheral controller (UIPC). It is used in applications having intensive real-time disk I/O or multiprocessing structures to reduce VMEbus traffic and increase system throughput and to add mass storage capacity.

To the functions offered by an MC68121 Intelligent Peripheral Controller device, the UIPC adds a DMA controller, 8K bytes of dual ported RAM with controller, a peripheral interface including a peripheral interrupt controller and a full VMEbus interface including a VMEbus requester and interrupt controller. Two 8K byte PROM's contain the control program for the UIPC functions as well as those for the floppy disk controller and SASI interface sections. The serial port function and service are provided by the MC68121 device. Figure 1 is a Functional Block Diagram of MVME315.

HOST/MVME315 COMMUNICATIONS

A driver running under a host communicates with the MVME315 via a command channel implemented in 128 bytes of dual ported RAM within the MC68121 IPC. Using a standard protocol, the driver transmits to the command channel packets of high-level application program requests for specific peripheral actions. The MVME315 control firmware retrieves the packets and executes the device dependent, low-level instructions required to provide the requested activities. In a similar manner, the control firmware places status packets in the command channel for retrieval by the host driver.

High level floppy disk functions supported by the MVME315 control firmware include:

- Read specified sectors
- Write specified sectors
- Read sector ID/head position
- Verify sector header and data
- Seek (specified track) and Restore (head to track 0)
- Format disk or track

The command channel is used for transmitting information, such as start and stop addresses, to the DMA controller. Diagnostic and self-test commands are also sent to MVME315 via the command channel.

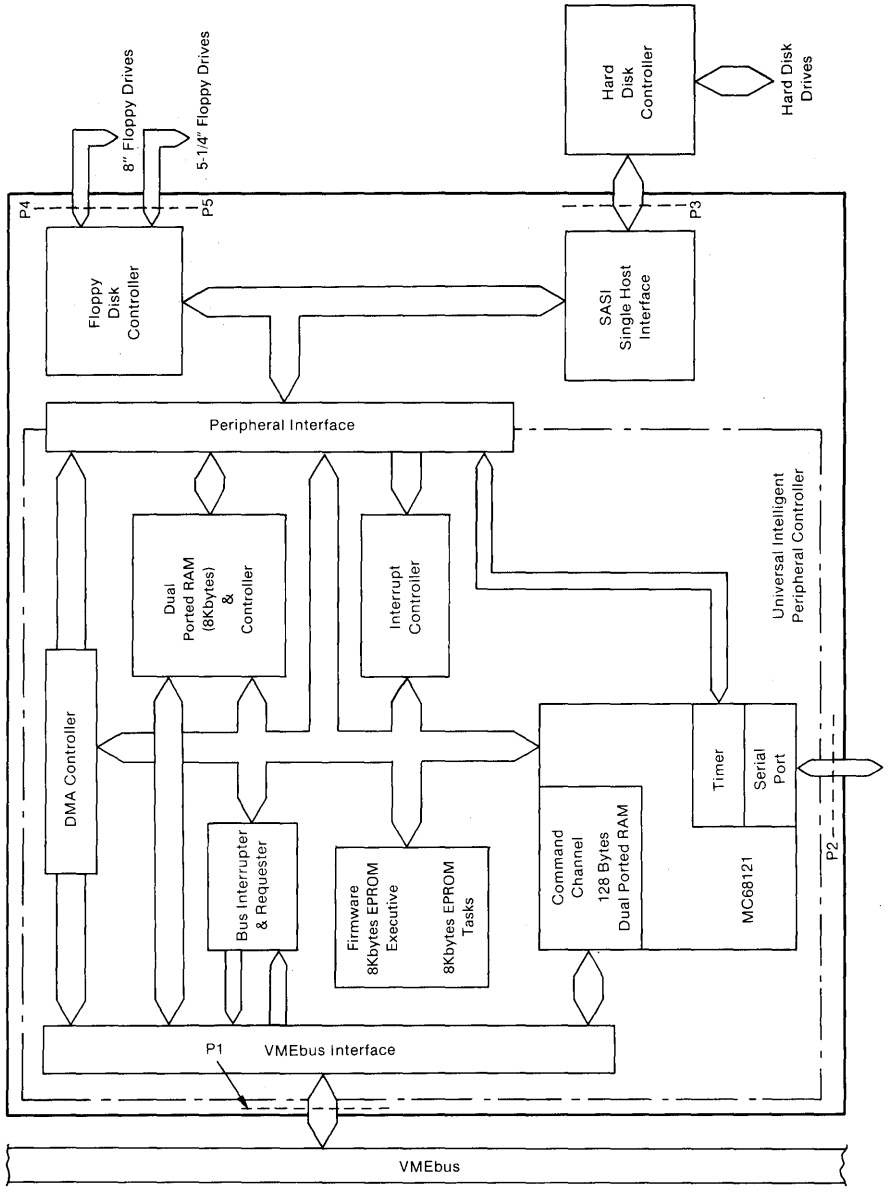


FIGURE 1 — MVME315 Functional Block Diagram

DMA Data Transfers

The MVME315 has a 4-channel controller which, under UIPC firmware supervision, manages the transfer of data between system memory and disk storage. An 8K byte dual ported RAM buffer with its own controler accommodates differences in VMEbus and disk data transfer rates. A host requests a DMA transaction by transmitting via the command channel source, destination and block size information to the DMA controller which then monitors the transfer of 16-bit data from system memory to disk or 8-bit data from disk to system memory.

Base Address Selection

The MVME315 has a header for jumper selection of a base address. A base address can be set on any 512 byte boundary throughout a 65,536 byte memory space.

Interrupt Priority Selection

The MVME315 has two headers for jumper selection of one of the seven VMEbus interrupt priority levels.

Bus Request Level Selection

The MVME315 has three headers for jumper selection of one of the four VMEbus bus request levels.

Local Memory Size Selection

The MVME315 has two headers for jumper selection of one of the two supported memory types: 2K x 8 static RAM (2716) or 8K x 8 static RAM (2764).

READY* Signal Disable

So that it may be used with 5-1/4" drives which do not provide a READY* signal for use by the floppy disk controller, MVME315 automatically generates a READY* signal from the INDEX* signal. A header is provided so that the generation of READY* can be jumper disabled so that MVME315 can safely be used with drives which produce this signal.

Serial Port

The MVME315 offers a 2-line serial port on the front panel supported by the MC68121 IPC. This port can be used for a debug terminal or download of programs from a host or development system.

Usage

The MVME315 will operate as a slave in a VMEbus system having any of the listed bus masters:

MVME101	VMEmodule Monoboard Microcomputer
MVME110-1	VMEmodule Monoboard Microcomputer
MVME115	VMEmodule Monoboard Microcomputer

Software Driver

A driver for the MVME315 is incorporated in the 4.3 release of the VERSAdos Real-Time Multitasking Operating System. For users desiring to write their own driver, a manual, Guide to Writing Device Drivers for VERSAdos, M68DRVGD/D1, detailing how to write a driver that runs under the M68000 Real-Time Multitasking kernel or under VERSAdos is available.

VMEbus Interface

The MVME315 has a full VMEbus interface which decodes all address and address modifier lines and provides bus requester and bus interrupter functions.

MVME315 SPECIFICATIONS

The specifications for the MVME315 are listed in Table 1.

TABLE 1 — MVME315 Specifications

Characteristic	Specification
Power Requirements	+5 Vdc ±5% +12 Vdc ±5% -12 Vdc ±5%
Environmental Requirements Operating Temperature Storage Temperature Humidity Range	 0° to 70° C -55° C to +85° C 0% to 95% (non-condensing)
Mechanical Specifications Height x Depth (board) Height x Width (front panel)	 9.2" (234 mm) x 6.3" (160 mm) 10.3" (262 mm) x 0.79" (20 mm)
Connectors VMEbus Serial Port SASI Interface Floppy Disk Interface	 DIN41612 C96 25-pin D-Subminiature 50-pin Right Angle 50-pin (8") 34-pin (5-1/4")

VMEbus Connector P1

The electrical and mechanical characteristics of VMEbus connector P1 are fully described in the VMEbus Specification Manual, MVMEBS/D1.

Serial Port Connector P2

Serial port connector P2 is a Sub D female, ERNI TMC-S-3-25-L or equivalent. The assigned pins and signals are pin 1 - TXD, pin 2 - RXD and pin 3 - GND.

SASI Interface Connector P3

The pin number and signal descriptions for SASI interface connector P3 are given in Table 2.

TABLE 2 — SASI Interface Connector P3 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Description
1 through 49 (odd only)	GND	System Ground
2 through 16 (even only)	DB0*-DB7*	Bidirectional data bus (7 = msb)
18	DPB*	Not used
20 through 30 (even only)	----	Not used
31	ATN*	Not used
34	Spare	Not used
36	BSY*	BUSY
38	ACK*	ACKNOWLEDGE
40	TRI*	RESET (RST*)
42	MSG*	MESSAGE
44	SEL*	SELECT
46	C*/D	CONTROL/DATA
48	REQ*	REQUEST
50	I*/O	INPUT/OUTPUT

8" Floppy Disk Interface Connector P4

The pin number and signal descriptions for 8" floppy disk interface connector P4 are given in Table 3.

TABLE 3 — 8" Floppy Disk Interface Connector P4 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Description
1 through 49 (odd only)	GND	System Ground
2, 4, 6	----	Not used
8	TG43*	TRACKS >43
10	DUAL*	Dual-Sided Indication
12	----	Not used
14	SSI*	SIDE SELECT
16	----	Not used
18	HLD*	HEAD LOAD
20	INDEX*	INDEX
22	READY*	READY
24	----	Not used
26	DRV0*	DRIVE 0 SELECT
28	DRV1*	DRIVE 1 SELECT
30	DRV2*	DRIVE 2 SELECT
32	DRV3*	DRIVE 3 SELECT
34	DIR*	DIRECTION
36	STEP*	STEP
38	WD*	WRITE DATA
40	WG*	WRITE GATE
42	TR00*	TRACK 0
44	WPRT*	WRITE PROTECT
46	RDD*	READ DATA
48, 50	----	Not used

5-1/4" Floppy Disk Interface Connector P5

The pin numbers and signal descriptions for 5-1/4" floppy disk interface connector P5 are given in Table 4.

TABLE 4 — 5-1/4" Floppy Disk Interface Connector P5 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Description
1 through 33 (odd only)	GND	System Ground
2	HLD*	HEAD LOAD
4	----	
6	READY*/ DRV3*	READY/DRIVE 3 SELECT
8	INDEX*	INDEX
10	DRV0*	DRIVE 0 SELECT
12	DRV1*	DRIVE 1 SELECT
14	DRV2*	DRIVE 2 SELECT
16	MOT*	MOTOR ON
18	DIR*	DIRECTION
20	STEP*	STEP
22	WD*	WRITE DATA
24	WG*	WRITE GATE
26	TR00*	TRACK 0
28	WPRT*	WRITE PROTECT
30	RDD*	READ DATA
32	SS1*	SIDE SELECT
34	----	Not used

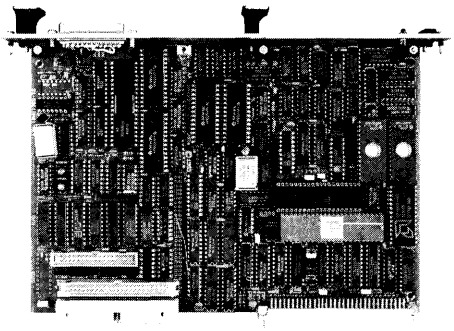
Ordering Information

Part Number	Description
MVME315	VMEmodule Intelligent Floppy Controller/SASI Interface with DMA. Includes User's Manual
MVME315/D	MVME315 Intelligent Floppy Controller/SASI Interface User's Manual

Related Documentation

HB212/D	VMEbus Specification Manual
M68DRVGD/D	Guide to Writing Device Drivers for VERSAdos

VMEmodule™ Intelligent SASI/SCSI Interface, Floppy Disk and Floppy-Tape Controller



- Supports up to two Winchester Controllers via a SASI/SCSI Interface
- Controls a 1/4" Cartridge Tape Drive Plus Two 5-1/4" Floppy Disk Drives or four 8" or 5-1/4" Drives in any Combination
- Supports all Standard Floppy Disk Formats
- Based on MC68121 Intelligent Peripheral Controller
- SYSTEM V/68 Support (UNIX)

The MVME319 combines in one double Eurocard module a floppy disk controller and a SASI/SCSI bus interface. Both are controlled by a universal intelligent peripheral controller (UIPC). Firmware supplied with the MVME319 supports either the XEBEC (SASI/ST506) or the Adaptec (SCSI/ST506) Winchester controllers. Also supported through the 8" floppy disk interface is a Cipher Floppy Tape using1/4" cartridges with up to 25Mb storage capacity. The tape interface may be used for Winchester disk backup as well as for selective file copies. Because the tape format is similar to that of a floppy disk, single sectors can be randomly accessed and modified.

In addition to the functions offered by the MC68121 Intelligent Peripheral Controller device, the MVME319 has a Direct Memory Access (DMA) controller, 32Kb of dual-ported RAM with controller, a peripheral interface including a peripheral interrupt controller, and a full VMEbus interface including a VMEbus requester and interrupt controller. Two 8Kb PROMs contain the control program for the UIPC functions as well as those for the floppy disk/tape controller and SASI/SCSI interface sections. Figure 1 is a functional block diagram of MVME319.

HOST/MVME319 COMMUNICATIONS

A driver program running on a VME system communicates with the MVME319 via a command channel implemented in 128 bytes of dual-ported RAM within the MC68121 IPC. Using a block-oriented protocol, the driver uses command channel packets to transmit a series of high-level application program requests for specific peripheral actions. The MVME319 control firmware retrieves the packets and executes the device-dependent, low-level instructions required to provide the requested activities. In a similar manner, the control firmware places status packets in the command channel for retrieval by the system CPU.

Due to the sector-oriented tape format, tape commands are the same as floppy disk commands.

High-level hard disk/floppy tape commands supported by the MVME319 control firmware include:

- Read Specified Sectors
- Write Specified Sectors
- Read Sector ID/head Position
- Verify Sector Header and Data
- Seek (specified track) and Restore (head to track 0)
- Format Hard Disk/Floppy Tape

Configuration and status request commands are also sent to the MVME319 via the command channel.

DMA DATA TRANSFERS

The MVME319 has a 4-channel controller which, under UPIC firmware supervision, manages the transfer of data between system memory and disk storage. A 32Kb dual-ported RAM buffer with its own controller accommodates differences in VMEbus and disk/tape data transfer rates. A host requests a DMA transaction by transmitting (via the command channel) source destination and block size information to the DMA controller. The DMA controller performs the transfer of 16-bit data between system memory and the RAM buffer, and the 8-bit data transfer between RAM buffer and SASI/SCSI interface or floppy tape.

BASE ADDRESS SELECTION

The MVME319 has a header for jumper selection of the command channel base address. A base address can be set on any 512-byte boundary throughout a 65,536-byte memory space.

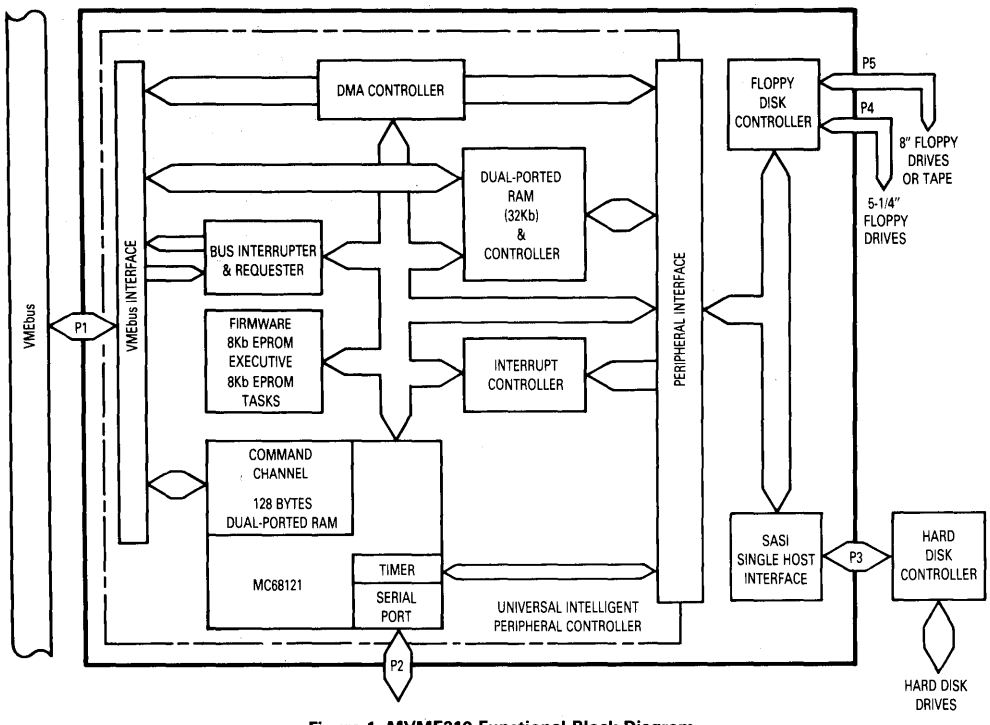


Figure 1. MVME319 Functional Block Diagram

INTERRUPT PRIORITY SELECTION

The MVME319 has two headers for jumper selection of one of the seven VMEbus interrupt priority levels.

BUS REQUEST LEVEL SELECTION

The MVME319 has three headers for jumper selection of one of the four VMEbus bus request levels.

READY* SIGNAL DISABLE

So that it may be used with 5-1/4" drives which do not provide a READY* signal for use by the floppy disk controller, the MVME319 automatically generates a READY* signal from the INDEX* signal. A header is provided so that the generation of READY* can be jumper disabled so that the MVME319 can safely be used with drives which produce this signal.

SOFTWARE DRIVER

A device driver for the MVME319 is available under SYSTEM V/68 which supports the disks and the tape.

VMEbus INTERFACE

The MVME319 has a full VMEbus interface which decodes all address and address modifier lines and provides bus requester and bus interrupter functions.

VMEbus CONNECTOR P1

The electrical and mechanical characteristics of VMEbus connector P1 are fully described in the VMEbus Specification Manual (MVMEBS/D).

SERIAL PORT CONNECTOR P2

Serial port connector P2 is a Sub D female, ERNI TMC-S-3-25-L or equivalent. The assigned pins and signals are pin 1 — TXD, pin 2 — RXD and pin 3 — GND.

The serial port is for hardware debug only, and is not supplied in the firmware.

SASI/SCSI INTERFACE CONNECTOR P3

The pin numbers and signal descriptions for SASI/SCSI interface connector P3 are given in Table 1.

8" FLOPPY DISK INTERFACE CONNECTOR P4

The pin numbers and signal descriptions for 8" floppy disk/CT525 Cipher Tape interface connector P4 are given in Table 2.

5-1/4" FLOPPY DISK INTERFACE CONNECTOR P5

The pin numbers and signal descriptions for the 5-1/4" floppy disk interface connector P5 are given in Table 3.

Table 1. SASI/SCSI Interface Connector P3 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Description
1 through 49 (odd only)	GND	System Ground
2 through 16 (even only)	DB0*–DB7*	Bidirectional data bus (7 = msb)
18	DPB*	Not used
20 thru 30 (even only)	—	Not used
31	ATN*	Not used
34	Spare	Not used
36	BSY*	BUSY
38	ACK*	ACKNOWLEDGE
40	TRI*	RESET (RST*)
42	MSG*	MESSAGE
44	SEL*	SELECT
46	C*/D	CONTROL/DATA
48	REQ*	REQUEST
50	I*/O	INPUT/OUTPUT

Table 2. 8″ Floppy Disk/Tape Interface Connector P4 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Description
1 through 49	GND	System Ground
2, 4, 6	—	Not used
8	TG43*	TRACKS >43
10	DUAL*	Dual-Sided Indication
12	—	Not used
14	SSI*	DISK SIDE SELCT/TAPE STREAM SELECT
16	—	Not used
18	HLD*	HEAD LOAD/TAPE DRIVE MOTOR ON/OFF
20	INDEX*	INDEX
22	READY*	READY
24	—	Not used
26	DRV0	DISK DRIVE 0 SELECT/TAPE STREAM SELECT
28	DRV1*	DISK DRIVE 1 SELECT/TAPE STREAM SELECT
30	DRV2*	DISK DRIVE 2 SELECT/TAPE STREAM SELECT
32	DRV3*	DISK DRIVE 3 SELECT/TAPE STREAM SELECT
34	DIR*	DIRECTION
36	STEP*	STEP
38	WD*	WRITE DATA
40	WG*	WRITE GATE
42	TR00*	TRACK 0
44	WPRT*	WRITE PROTECT
46	RDD*	READ DATA
48, 50	—	Not used

Table 3. 5-1/4″ Floppy Disk Interface Connector P5 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Description
1 through 33 (odd only)	GND	System Ground
2	HLD*	HEAD LOAD
4	—	
6	READY*/ DRV3*	READY/DRIVE 3 SELECT
8	INDEX*	INDEX
10	DRV0*	DRIVE 0 SELECT
12	DRV1*	DRIVE 1 SELECT

Cont.

Table 3. 5-1/4" Floppy Disk Interface Connector P5 Pin Assignments and Signal Descriptions (Cont.)

Pin Number	Signal Mnemonic	Description
14	DRV2*	DRIVE 2 SELECT
16	MOT*	MOTOR ON
18	DIR*	DIRECTION
20	STEP*	STEP
22	WD*	WRITE DATA
24	WG*	WRITE GATE
26	TR00*	TRACK 0
28	WPRT*	WRITE PROTECT
30	RDD*	READ DATA
32	SS1*	SIDE SELECT
34	—	Not used

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristic	Specification
Power Requirements	+5 Vdc ±5% 3.7A(typ) 4.6 A (max) +12 Vdc ±5% 30 mA 50 mA −12 Vdc ±5% 20 mA 33 mA
Environmental Requirements Operating Temperature Storage Temperature Humidity Range	 0° to 50°C −55°C to 85°C 5% to 95% (non-condensing)
Mechanical Specifications Height x Depth (board) Height x Width (front panel)	 9.2" (234mm) x 6.3" (160mm) 10.3" (262mm) x 0.79" (20mm)
Connectors VMEbus Serial Port SASI/SCSI Interface 8" Floppy Disk/Tape Interface 5-1/4" Floppy Disk Interface	 DIN41612C96 25-pin D-Subminiature 50-pin Right Angle 50-pin 34-pin

ORDERING INFORMATION

Part Number	Description
MVME319	VMEmodule Intelligent Floppy/Tape Controllers and SASI/SCSI Interface with DMA. Includes User's Manual
MVME319/D	MVME319 Intelligent Floppy Disk/Tape Controller and SASI/SCSI Interface User's Manual

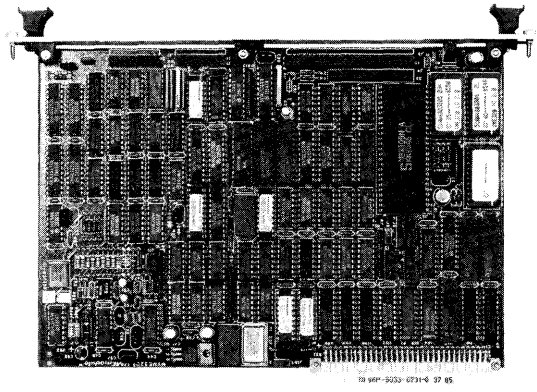
RELATED DOCUMENTATION

HB212/D	VMEbus Specification Manual

MVME320A
MVME320A-1

VMEmodule™ VMEbus Disk Controller Module

- Controls Mixed 5-1/4" and 8" Drives: Up to Two Winchester Hard and Two Floppy Disk Drives or up to Four Floppy Disk Drives
- Provides DMA Data Channels
- Supports Serial Data Rates up to 5 Megabits Per Second
- Uses Standard IBM Formats and FM and MFM Recording
- User-Programmed Hard Disk Format
- Supports Soft Sectored Drives
- Multiple-sector Read/write
- Implied Seek
- Bit Serial 16-Bit CRC/32-Bit ECC Generator
- Single-phase Write and Read Clocks
- Data Transfer — 16-Bit DMA/8-Bit Register
- Self Diagnostics
- VMEbus Requester and Interrupter



The MVME320A VMEbus Disk Controller Module is a double-high Eurocard module for adding mass storage capacity to a VMEbus system. It provides direct memory access (DMA) data channels between system memory and Winchester hard disk drives and/or floppy disk drives. The MVME320A is used in applications having intensive real-time disk I/O or multiprocessing structures designed to reduce VMEbus traffic and increase system throughput. Figure 1 is a functional block diagram of the disk controller module.

The "A" and "A-1" versions are each an upgrade/cleanup of the original MVME320 and can be used as a direct replacement. The major change is the addition of a connector for direct interface to floppy disk drives (which greatly simplifies cabling). In the "A-1" version, connections can be made without passing cables through the front panel, as is required with the "A" version.

HOST/MVME320A COMMUNICATIONS

An intelligent module, the MVME320A offers the user a high level, easy to program interface. Microinstructions in a 4K x 24 ROM are executed on a Signetics 8 x 305 Microcontroller at a 200 ns per instruction rate to provide the macro I/O activities requested by the user program.

To a driver running under the host operating system, the controller appears as seven 8-bit control registers at the base address of the module, one block of global memory for each disk drive and in global memory a table of pointers to the starting address of each block. General control of module operations is obtained using the seven registers to pass initialization, vector number, first pointer address and command initiation information to the controller. The registers are also used to obtain interrupt source and drive status information from the module.

The other host/module communications avenue is the event control areas (ECA's). The ECA's are memory blocks used to convey drive control parameters and command execution parameters from host to module and the status of executed commands from module to host. An ECA comprised of static and dynamic fields is created by the host when requesting the execution of a command. Static fields remain unchanged. Dynamic fields are used during execution and are updated at completion by the controller.

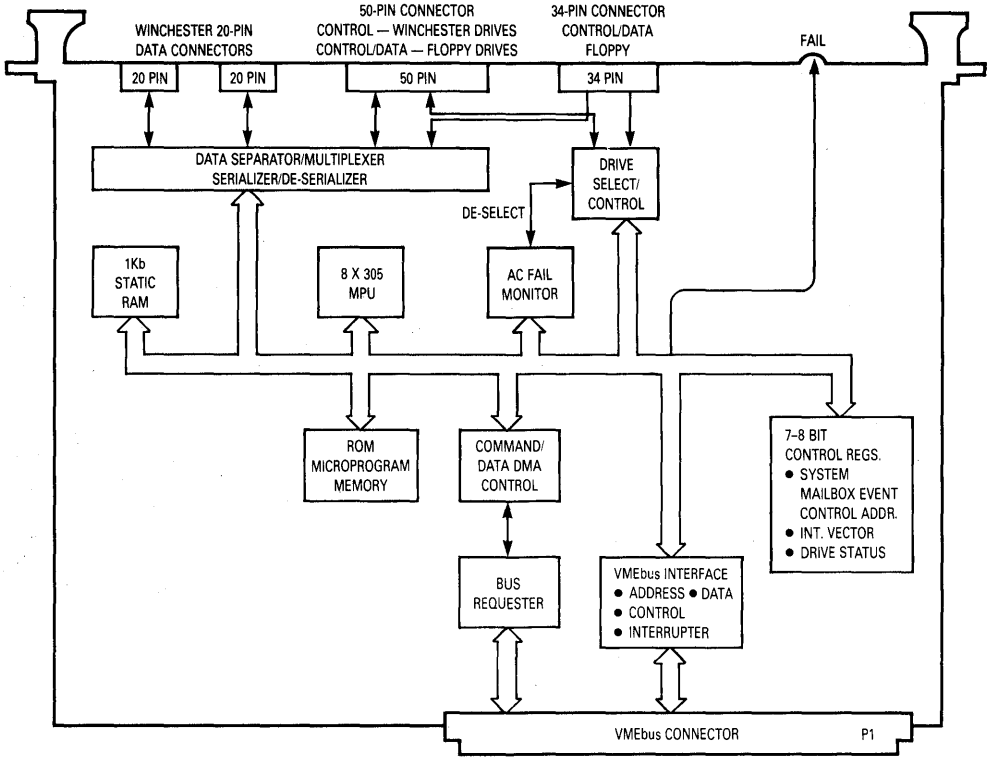


Figure 1. MVME320A Functional Block Diagram

DISK DRIVE INTERFACE

The disk interface consists of two sections:

- Input and output ports that sense and generate slow changing or static control signals
- Serial data I/O and high speed disk control circuitry.

Both sections are serviced by the fast 8 x 305 microcontroller allowing the module to control a variety of disk types and media formats.

Up to four disk drives can be interfaced to the MVME320A. Of these, two can be Winchester 5-1/4" or 8" hard disk and two can be 5-1/4" and/or 8" floppy disk. If all are floppy, four can be used. Mixed single and double sided floppy disk drives of single or double density are supported.

All head positioning (up to 64K cylinders/drive) and head selection (up to 8 heads/drive) is automatically provided by the controller for host requested I/O.

Table 1 is a partial list of the more popular disk drives which can be used with the controller module. Other compatible disk drives may be substituted for those listed in the table.

DATA INTERFACE

For a requested read or write operation, data is automatically transferred beween any of the four disk drives and system memory which can be located anywhere in the VMEbus 16Mb address space.

Floppy disk data transfer is done using two byte buffering. For hard disks, a 1024 byte local buffer is used. Automatic head switching and cylinder positioning is performed for operations requiring multiple sector read or write.

The MVME320A design uses a multiplexer to transfer disk data to a phase locked loop data separator before sending the data to serializer/deserializer (SERDES) circuitry. This section converts serial disk data to parallel bus data for bus transfer and parallel bus data to serial for transfer to disk. The SERDES section also generates and checks CRC code and has a watchdog circuit that monitors the various disk operations for timely completion. The number of retries of an operation that the module will attempt is programmable.

Table 1. MVME320A — Compatible Disk Drives

Hard Disk Drives	
5-1/4"	
Seagate ST506, ST512, ST406, ST412 Shugart SA600 RMS500, RMS506, RMS512	
Floppy Disk Drives	
8"	5-1/4"
Shugart SA800, Sa801, SA810, SA850	Amlyn 5850 Shugart SA400, SA410, SA450 1.2 Mb Floppy

Table 2. Disk Controller Command Set

Command	Description
CALB	Recalibrate (reposition head) to track zero.
REMS	Using implied seek, read multiple sectors.
WRMS	Using implied seek, write multiple sectors.
FORM	Using implied seek, format tracks.
WRDD	Write using deleted data address mark/flag.
VER	Using implied seek, verify data (no transfer).
TSR	Transparent sector read (ignore CRC/ECC errors).
CORR	Using ECC remainder, correct data in memory.

Table 3. Boot/Load Support

Microcomputer	Debug/Monitor	Disk Controllers with Boot Support in the Monitor
MVME101	MVME101bug	MVME315
MVME110-1	MVMEbug	MVME315, MVME320A, M68RWIN1
MVME120, 121, 122, 123	MVME120bug	MVME315, MVME320A
MVME130, 131	MVME130bug	MVME319, MVME320A, MVME360

ERROR DETECTION AND CORRECTION

To insure data integrity, the MVME320A provides two software selectable modes of data validation: 16-bit cyclic redundancy check (CRC) or 32-bit error checking and correction (ECC). The ECC algorithm used can correct errors separated in data by as many as 11 bits.

COMMAND SET

The controller module offers the set of eight high level commands listed in Table 2.

DISK CONTROLLER USAGE

The Disk Controller can be used as a slave in a VMEbus system in which any of the monoboard microcomputers listed in Table 3 operates as bus master. Note that not all support boot load from an MVME320A, however, Table 3 lists the disk controllers for which bootstrap, load and dump capabilities are provided by the debugger supplied with the corresponding microcomputer.

BASE ADDRESS SELECTION

The module has a jumper selection header for a base address which can be set on any 1Kb boundary throughout the short I/O VMEbus address space.

ADDRESS MODIFIER CODE SELECTION

The MVME320A has a header for selection of response to address modifier codes for user (hex 29) and supervisor (hex 2D) or for response to the supervisor code only. As shipped, the module responds to both codes.

VMEbus INTERRUPTER

The module has circuitry for generating an interrupt on any seven VMEbus interrupt priority and interrupt acknowledge levels: 1 through 7. Selection of an interrupt and an acknowledge level is accomplished by jumper in the corresponding headers. As shipped, both levels are set at 3.

VMEbus REQUESTER

The MVME320A has the logic required to request mastership of the data transfer bus (DTB) on any one of the four bus request and four bus release levels and for operation in the release-when-done mode. The module has two headers for jumper selection of one of the four bus request levels and one of the four bus grant levels. As shipped, both are set for level 3.

SOFTWARE DRIVER

A driver for the disk controller module is incorporated in the 4.5 release of the VERSAdos Operating System and is also available as a separate product. The driver supports interface with the drives listed in Table 1. For users desiring to write their own driver, a manual, Guide To Writing Device Drivers for VERSAdos (M68KDRVGD/D1), details how to write a driver that runs under the M68000 Real-Time Multitasking kernel or under VERSAdos.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Configuration	DTB Master: A24, D16 DTB Slave: A24, D8
Form Factor	Double High Eurocard
Power Requirements	+5 Vdc at 2.6 A typ ±12 Vdc at 20 mA typ
Environment Limits Operating Temperature Storage Temperature Humidity	 0°C to +50°C inlet air, forced air cooling −55°C to +85°C 8% to 80% (non-condensing)
Physical Characteristics PC Board Height PC Board Depth	 9.2 in. (234 mm) 6.3 in. (160 mm)

ORDERING INFORMATION

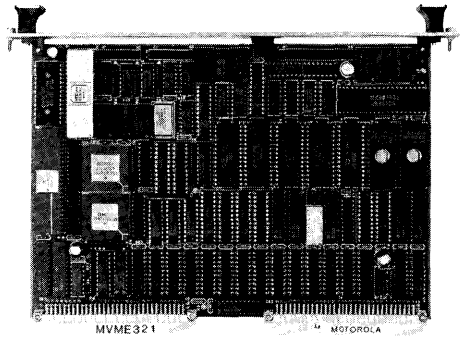
Part Number	Description
MVME320A	VMEmodule Winchester/Floppy Disk Controller. Includes User's Manual.
MVME320A-1	Same as MVME320A, but with "side-mount" connectors. Include's User's Manual.
MVME320A/D	VMEmodule Winchester/Floppy Disk Controller User's Manual.
MVME702A	Connector Transition Board for MVME320A, for simplified cabling or expansion disk connection. Includes User's Manual.
MVME702A/D	Transition Module User's Manual.

RELATED DOCUMENTATION

Part Number	Description
HB212/D	VMEbus Specification Manual.
M68KDRVGD/D	Guide to Writing Device Drivers for VERSAdos

Product Preview

VMEmodule™ VMEbus Disk Controller Winchester and Floppy



- Controls up to Two 5-1/4" Winchester Hard Disks, ST506 Compatible
- Controls up to Four 5-1/4" Floppy Disks, SA400 Compatible
- Double-High VMEmodule, Occupies a Single VMEbus Slot
- Supports 24- or 32-bit DMA Addressing
  - — A32/A24:D32/D16 VMEbus Master Interface
  - — A16:D16/D08(E0) VMEbus Slave Interface
- Software Programmable Interrupt Levels and Vectors
- 32Kb High-Speed Static RAM for Data Buffering
  - — Zero-Wait Cycle Access for Local CPU and DMA Controller
  - — Multi-track Data Caching for Optimal Performance
- 8Mb/s DMA Transfer over VMEbus (32-bit transfers)
- Optimum 1:1 Sector Interleave on Winchester Disk
- Intelligent Disk Management and Standard Buffered Pipe Protocol Implemented by Onboard MC68010 CPU (10 MHz)
- Multitasking Control Kernel Supports Concurrent Execution of Peripheral Device Operations, Multi-sector Transfers with Implied Seek, and Command Chaining for Efficient Paged Memory Management Transfers
- 32-Bit Error Correction (ECC) Allows Transparent Correction of up to 11-Bit Burst Errors on Winchester Disk
- 16-Bit Cyclic Redundancy Check (CRC) on Floppy Disk
- Transparent Bad Block Handling with no Added Seeks
- Fully VMEbus Specification Rev C.1 Compatible

The MVME321 disk controller provides a high performance DMA interface between VMEbus and rotating mass storage; 5-1/4" Winchester and floppy disks. It is intended for use in applications having extensive operating system or real-time disk I/O requirements, where the onboard processor and high-speed DMA can improve over-all system throughput.

Figure 1 is a functional block diagram of the MVME321.

VMEbus INTERFACE

The interface between the MVME321 and VMEbus can be considered from two points of view: from a purely hardware perspective, or, from a higher-level host-operating-system or system-integration perspective.

At the hardware level, the MVME321 has an A16:D16/D08(E0) VMEbus slave interface and an A32/A24:D32/D16 VMEbus master interface. Translated, this means that the MVME321 appears as a slave device in the VMEbus address space called "short I/O space" (A16) where it will accept 8- or 16-bit transfers (D16/D08), and can become a VMEbus master with the ability to generate 24- or 32-bit addresses and execute 16- or 32-bit data transfers.

To request control of the VMEbus as a bus master, the MVME321 has bus request logic which can be jumper set for any of the four VMEbus request levels BR0..BR3, and can also be programmed to release the bus either on request from another device (ROR) or when done (RWD). Both release mechanisms execute "early release of bus busy," which encourages optimal VMEbus utilization by allowing the system controller to perform arbitration for the next VMEbus master concurrently with the MVME321's last cycle.

To signal completion of a requested operation (reset, disk write, etc), the MVME321 has a VMEbus interrupt generator implemented by the Motorola MC68153 interrupter chip. With the capability of generating four simultaneous, independent interrupts, each with its own programmable vector and level, this device provides the onboard firmware with the tools to implement very sophisticated concurrent disk management processes each having its own interrupt.

CONTROL/STATUS REGISTER

Addressed with only 16 address lines (A16), the "short I/O space" on VMEbus is used primarily for slave modules' control and status registers (CSR). For the MVME321, a Motorola standard CSR for intelligent peripheral controllers is used, as shown in Table 1. The

2

MVME321's CSR space is the only part of the MVME321 that is "visible" to a host on VMEbus. Through the CSR, the host can check the status of the MVME321 (reset, busy, etc), give the MVME321 an address in VMEbus global memory where a command block may be found, or check the status of the most recent command (command complete, error codes, etc). Provision is also made for the host to request the attention of the local MPU with a level 2 interrupt, and for other control functions such as directly forcing a hard MVME321 reset.

The MVME321 CSR space is 16 bytes long, and may be mapped onto any 256 byte boundary in the short I/O space, by setting switches.

The MVME321 may be used in a 'standard' A24:D16 VMEbus system with a P1-only backplane, or in an extended A32:D32 system with a combined P1 and P2 backplane. Internal control bits allow the local firmware to select either 16- or 32-bit DMA transfers on VMEbus. The extended VMEbus address and data signals on the center row, b, of the P2 backplane connector are fully connected, but can be left 'open' in a P1-only A24:D16 system. Power and ground on the P2-b connector MUST, however, be connected, even in a P1-only system. The other two rows of the P2 connector, a & c, are used for the floppy disk interface. I/O connections to the Winchester disks are made through connectors near the front edge of the board.

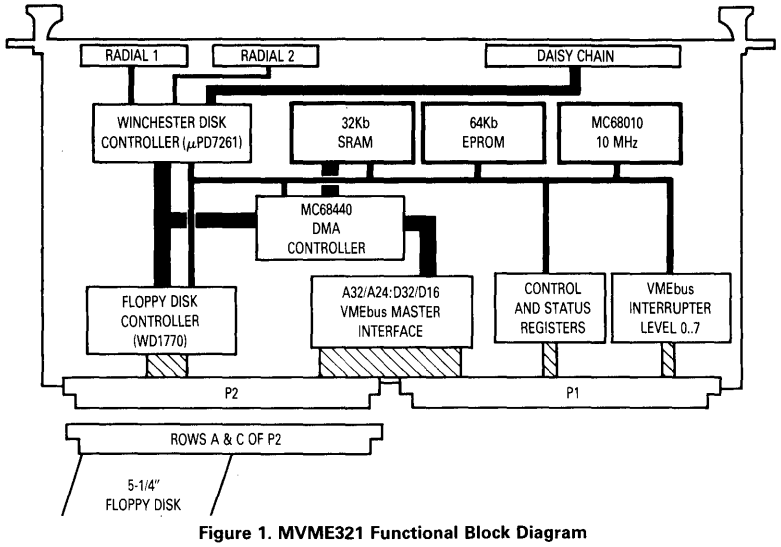


Figure 1. MVME321 Functional Block Diagram

Table 1. Control and Status Register

Offset from base address			
0	A31 ..	IPC ADDRESS REGISTER (msw)	.. A16
2	A15 ..	IPC ADDRESS REGISTER (lsw)	.. A0
4	IPC ADDR MODIFIER REGISTER	— reserved —	
6	IPC CONTROL REGISTER	— reserved —	
8	IPC STATUS REGISTER	— reserved —	
A	IPC ID REGISTER	— reserved —	
C	IPC ABORT VECTOR REGISTER	— reserved —	
E	CHANNEL CONTROL REGISTER		
+	D15		D0

MVME321/HOST INTERFACE

From the other perspective, the point of view of a host operating system, the MVME321 is an "intelligent" peripheral controller, with an onboard microprocessor. This allows implementation of a standard, high-level interface to the VMEbus, which is independent of the type of peripheral being controlled.

Motorola's advanced Buffered Pipe command Protocol (BPP), used on the MVME321, is a general purpose message passing mechanism between the host operating system driver software and the MVME321's onboard firmware. To execute a disk I/O operation, for example, the driver builds a command packet in global VMEbus memory consisting of high-level commands, status bytes, and pointers to data blocks in global memory. The driver then passes a pointer to this packet to a generic BPP routine that queues it into the MVME321's input pipe (linked list), and possibly generates an interrupt to alert the MVME321's firmware that a new packet is ready.

Another generic BPP routine (part of the firmware executing on the MVME321's local MPU) reads the packet into local memory and passes control to the board-specific command interpretation and execution logic which understands the details of the local hardware and can carry out the command. On completion, the process reverses to return the command status to the calling routine. The BPP protocol allows one or more hosts to buffer commands to the MVME321 without requiring a special interlock mechanism. Multiple commands can be queued without the host CPU having to wait for completion of earlier requests, and command chaining is supported for efficient operation in a system with paged memory management.

The local firmware is a multitasking kernel comprising the BPP module, command interpretation logic, and control logic for the direct memory access (DMA) data handler & the disk controller chips. It supervises concurrent execution of I/O managers and manages a pool of data buffers that is shared by all devices. The hard disk manager performs implied seeks, overlapped operations beween drives, and schedules transfers to optimize head movement. Also included is a power-up diagnostic self-test.

Using BPP, the peripheral controller's VMEbus hardware interface can be made standard, regardless of the function that the module is to perform, and a substantial part of the local firmware and operating system driver becomes standard (BPP generic routines) making it much easier to write both firmware and drivers. Also, since the hardware interface is independent of the module function and the software interface is at a high level, significant hardware upgrades and modifications can be made without affecting the host software. It is further possible to move functionality from the operating system down into onboard firmware simply by changing the driver and firmware — requiring no hardware modifications. An example might be a custom version of the MVME321 with a full file manager on board.

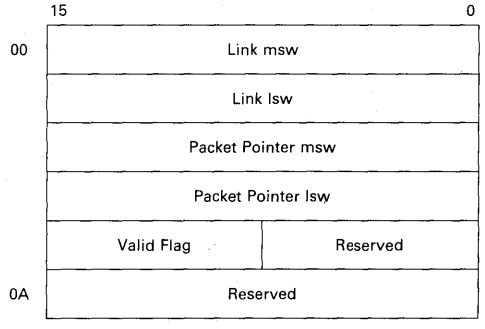


Figure 2. Channel Pipe Envelope

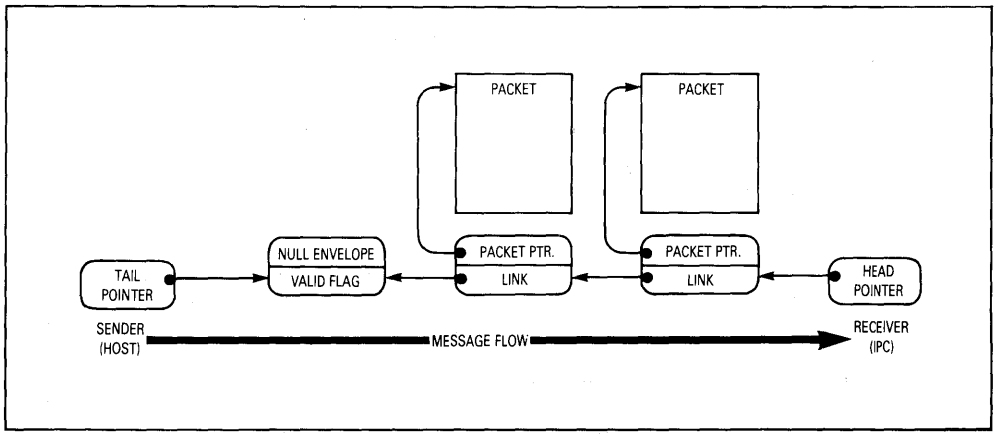


Figure 3. Message Flow — Channel Command Pipe

MICROPROCESSOR AND MEMORY

A local MC68010 MPU running at 10 MHz executes the MVME321's multitasking control kernel. This kernel implements Motorola's "buffered pipe command protocol" VMEbus interface, and handles all local buffer and DMA control for concurrent operation.

The MC68010, is a microprocessor of proven architecture, optimized for efficient support of high-level languages and secure operating systems.

MC68010 features include:

- 32-Bit Internal Data, Address, and Stack Registers
- 14 Addressing Modes
- 16Mb Direct Addressing Range
- 57 Instruction Types
- Operation on Five Data Types: Bit, Byte, Word, Longword, & BCD
- 256 Multi-level Vectored Interrupts
- Fast 'Loop-mode' Execution for Frequently Needed Sequences

Two 28-pin JEDEC sockets are used for firmware EPROMs. Organized as a 16-bit wide memory, they accept either 32K x 8 or 64K x 8 EPROMs, for up to 128Kb of firmware code. EPROMs with 300 ns or faster access time operate with one wait cycle for the MPU at 10 MHz.

Local buffer memory is implemented in four 28-pin JEDEC sockets. Using 8K x 8 static RAMs, this is 32Kb of zero wait cycle memory for the MPU and DMA buffering. Optionally, use of 32K x 8 RAMs could expand the buffer to 128Kb.

Local control I/O and a 24-bit timer are provided by a Motorola MC68230 parallel interface/timer. Clocked at 10 MHz, the timer has a 100 ns resolution.

DIRECT MEMORY ACCESS (DMA)

High-speed data paths are provided between the VMEbus & local buffer memory, and between local buffer & the disk controllers. Implemented using a Motorola MC68440 direct memory access (DMA) controller, data transfer speeds of 8Mb/s are supported to and from VMEbus (to a sufficiently fast VMEbus memory — 400 ns or faster cycle time), as are full disk data transfer rates to and from disk.

Particularly for the Winchester disk, supporting maximum disk transfer rates means that the disk can be formatted for optimum performance with an interleaving factor of 1:1, or no sector interleaving. Combining this with a large buffer sufficient to hold several complete tracks and well designed control firmware for buffer management, the MVME321 obtains the best performance possible from a Winchester disk or disks. Local firmware controlled DMA is used for all accesses to VMEbus including reading command blocks and writing back the completion codes and status. Optimized buffer management during data transfers, also under firmware control, ensures minimum command response time. For a disk read, this is the time from the command being actually copied into the MVME321 to the beginning of data transfer to VMEbus, and is heavily influenced by any buffer allocation or switching overhead in the firmware.

DMA transfers to and from VMEbus can be "throttled" under software control to minimize VMEbus loading and promote fairness of VMEbus arbitration. A typical throttle setting would be for eight VMEbus cycles — meaning that the bus would be released and re-requested every eight cycles to allow other VMEbus masters plenty of opportunity to access the bus.

For most systems, the MVME321 can be set for "release-on-request" (ROR) mode of operation, with NO throttling. In this mode, it uses the full VMEbus bandwidth whenever it needs it, but releases automatically when another bus request is made.

DISK CONTROLLERS

An μPD7261 hard disk controller from NEC interfaces the Winchester disks. Two connections from the controller are required for each attached disk. A "daisy chain" cable goes to each disk in sequence, for addressing and control, and a second cable for data transfer goes directly from the controller to each disk individually. The MVME321 has connectors for one hard disk daisy chain cable and two data cables, called radial cables, and can therefore support one or two 5-1/4" hard disks. The interface is compatible with the Seagate Technology ST506 interface, and supports serial data transfer at 5Mbit/s (which is approximately equivalent to 600Kbytes/s of parallel data from the μPD7261).

Winchester drives with both fixed and removable media are supported, as well as the very high density drives which require that the controller be capable of handling up to 16 heads.

Control and status information is accessed directly in the μPD7261 by the local MC68010 MPU. Once both the μPD7261 and the DMA controller have been set up, data will be transferred automatically as required, and the MPU can proceed to other functions such as acquiring the next command.

NOTE: The error correction code (ECC) used by the μPD7261 is not compatible with that used by the MVME320A Winchester disk controller. This means that an MVME320A cannot read a disk written by an MVME321, and vice-versa. This is particularly of concern for removable media.

The Western Digital WD1770 floppy disk controller is used to interface 5-1/4" floppy disks. It supports both 125Kbit/s 'single density' and 250Kbit/s 'double density' serial transfer rates, according to the Shugart Associates SA400 interface standard, and track densities of 48 or 96 tracks/inch, single or double sided.

A single daisy chain cable connection, which comes out the P2 connector on the back edge of the board, is required for floppy disks. This connection is terminated with resistor pull-ups on the MVME321, and requires only a 34-pin flat ribbon cable to connect up to four floppy drives.

NOTE: 8" floppy drives are NOT supported, nor are the 'triple density' 5-1/4" drives that operate at the 8" 500Kbit/s data rate (double density 8" floppy data rate).

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
VMEbus Configuration	DTB Master A32/A24:D32/D16 DTB Slave A16:D16/D08(E0) — No VMEbus arbiter — Requester: Any one of R(0)..R(3) Static ROR or RWD Early release of bus busy (BBSY*) — No Interrupt handler for VMEbus interrupts — Interrupter: Any of I(0)..I(7), Programmable
Form Factor Card Dimensions Front Panel	Double High Eurocard Height 234mm (9.2"), Depth 160mm (6.3") Height 261mm (10.2"), Width 21mm (0.83")
Power Requirements MVME321	 +5 V (tbd) typical, (tbd) maximum
Environmental Limits Operating Temperature Storage Temperature Humidity	 0°C to 55°C inlet air temperature with forced air cooling −40°C to 85°C 5% to 90% relative humidity (non condensing)

ORDERING INFORMATION

MVME321	VMEbus disk controller for two 5-1/4" Winchester and four 5-1/4" floppy disk drives. Full A32:D32 and four 5-1/4" floppy disk drives. Full A32:D32 VMEbus interface, 8Mbyte/s DMA data transfer, multi-track cache, 1:1 interleave, multitasking control kernel running on 10 MHz MC68010 MPU, supports BPP interface. Includes User's Manual. Supported by SYSTEM V/68 and VERSAdos operating system drivers.
MVME321/D1	MVME321 VMEbus Disk Controller User's Manual.

RELATED DOCUMENTATION

HB212/D	VMEbus Specification Manual
MC68010UM	MC68010 Microprocessor User's Manual

MVME330
MVME330-1
MVME330-2

VMEmodule™ LAN Controller

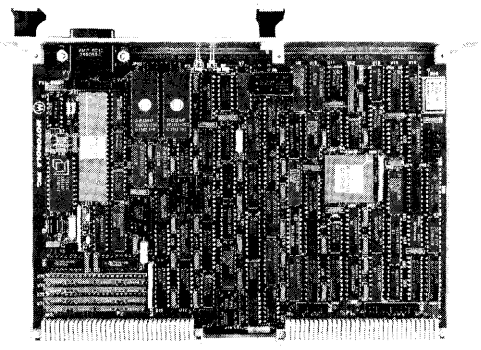
- Frees VMEbus Hosts From Protocol Processing Burden
- Interfaces with VMEbus and Hosts in the Ethernet System
- Interfaces with Hosts Running Under VERSAdos or SYSTEM V/68 in the Same Ethernet System
- Based on AM7990 LANCE Ethernet Controller, AM7992 Serial Input/Output Adapter and MC68000 16-/32-Bit Microprocessing Unit or MC68010 16-Bit Virtual Memory Microprocessor

Hardware Features

- 10 MHz MC68000 or MC68010 MPU
- VMEbus Compatible
- Ethernet, Version 2.0 Compatible
- AM 7990/7992 (LANCE/SIA) VLSI
- 128K or 512K Dynamic RAM with Parity
- Interrupt Capabilities
  VMEbus to MPU
  MPU to VMEbus with Programmable Vector
- 2 ms Timer
- Bus Requester/Master Capability
- 8K or 32K EPROM
- Power Up Self-Test
- One Wait State RAM Write Access
- No Wait State RAM Read Access
- 0°–50°C Operating Temperature Range

Software Features

- Host-Specific Application Packages: File Transfer, Network Utilities, Virtual Terminal, Runtime Library
- SYSTEM V/68 or VERSAdos Drivers
- XNS Protocol Package Includes: Echo, Error, Sequenced Packet, Datagram
- Firmware and Communications Executive



The MVME330, MVME330-1 and MVME330-2 LAN Controller VMEmodules are a family of advanced communications processor boards that provide for VME-based systems the interface and the performance for 10 Mbps Ethernet 2.0 implementation. Each matches Ethernet conformance to the IEEE 802.3 local area network specification (CSMA/CD), conforms to VMEbus industry standards and is fully supported by Motorola's SYSTEM V/68 and VERSAdos operating systems.

The MVME330 family frees any VMEbus host from significant protocol burden by means of a double high Eurocard node processor design incorporating an MC68000 16-Bit Microprocessing Unit or MC68010 16-Bit Virtual Memory Microprocessor and the VLSI Lance Ethernet Controller — AM7990. A communications executive executes on the LAN controller MPU to supervise the Lance chip in its processing of the Xerox Network System (XNS) protocol package. A VLSI Serial Input/Output Adapter — AM7992 — provides Manchester encoding/decoding for the Ethernet interface. When the Ethernet interface is operating at 10 Mbps, the data exchange rate at the VMEbus interface is about 130 one kilobyte packets per second. A block diagram of the modules is shown in Figure 1.

The module's on-board MPU facilitates the downloading of custom protocol implementations. Host-specific drivers can be written for interfacing an MVME330 family board to another host.

By interfacing directly with the LANCE chip to perform all Ethernet I/O, the controller kernel acts as a standard interface between the hardware and the XNS software which in turn communicates out over VMEbus to the host application software. The LANCE interfaces via Ethernet to another node and then through that LANCE to the peer XNS software at another node. A controller's communications executive comprises initialize, timer, transmit, receive, status and network statistic functions, as shown in Figure 2.

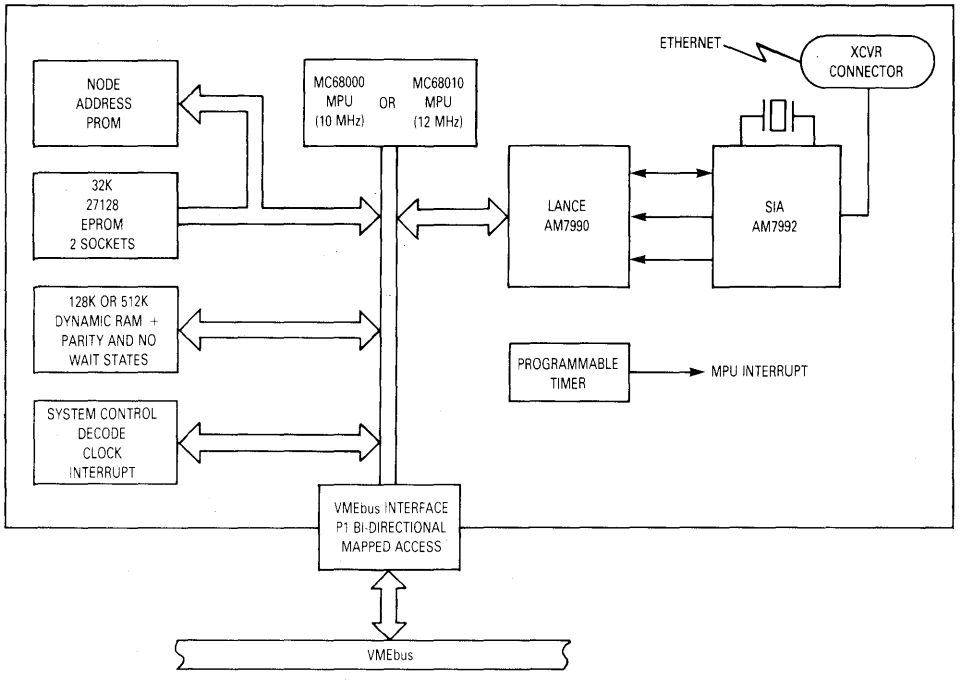


Figure 1. MVME330 Family LAN Controllers

The MVME330-resident layers include XNS inter-network Datagram protocol in the network layer and XNS Echo, Error and Sequenced Packet modules in the transport layer. The host interfaces the operating system with File Transfer, Network Utilities, Run Time Library, Virtual Terminal and user written application packages. Data transfer across the physical VMEbus interface between host and LAN module is accomplished using a shared memory interface.

Figure 3 depicts a two node Ethernet network.

FUNCTIONAL DESCRIPTION — HARDWARE

MVME330 offers SYSTEM V/68 or VERSAdos hosts on VMEbus a single board connection to an Ethernet system. Compatible with Ethernet Version 2.0, it can support a data transfer rate of 130 1Kb length packets per second. Requiring no additional hardware but a cable, any MVME330 family module can be connected to standard Ethernet transceivers.

The MVME330 is designed so that it can be used in File, Print and Terminal Servers and other network applications. It facilitates network management by monitoring collisions and retrys, frame checking, timeouts and error free transmissions and receptions. Such configuration control functions as initializing, suspending, and resuming data link operations, setting the physical address and the addressing mode (normal, broadcast, multicast) are also supported by the hardware.

MPU

The processor on board MVME330 and MVME330-1 is a 10 MHz MC68000. The processor on board MVME330-2 is a 12 MHz MC68010. Processor duties include moving commands and data to and from system memory; generating VMEbus interrupts, executing the network and transport XNS protocol layers, providing timer functions and, under direction of the communications executive, controlling the LANCE in its execution of the data link protocol.

DYNAMIC MEMORY

As shipped, MVME330 has 128K and MVME330-1,-2 have 512K bytes of dynamic RAM with parity which is accessible from VMEbus, the LANCE and the processor, in that order of priority.

ROM/EPROM

Each LAN controller has two 28 pin sockets in which various 8K x 8 and 4K x 8 EPROM devices can be used.

LOCAL AREA NETWORK CONTROLLER FOR ETHERNET (LANCE)

The LANCE is a VLSI device used for interfacing a microprocessor system to a local area network; namely the base-band, carrier sense multiple access with collision detection (CSMA/CD) shared medium defined by

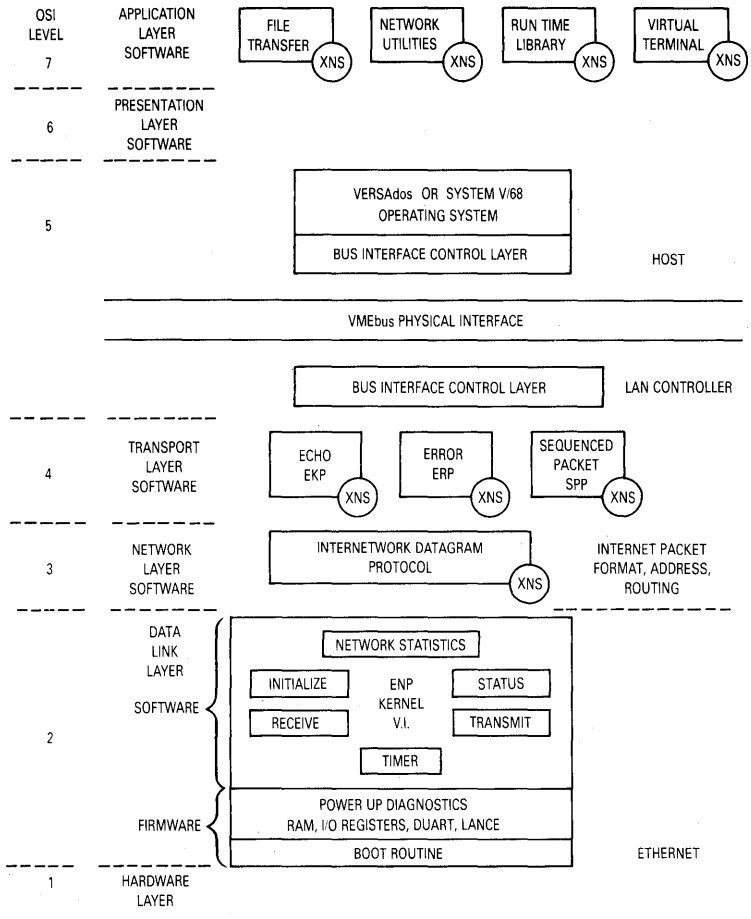


Figure 2. Host/MVME330 Family Ethernet Functions

Ethernet 2.0. It is designed to operate in an environment comprising local memory and a microprocessor with the memory serving as the communication link between the microprocessor and the LANCE and as a buffer for Ethernet packets.

LANCE operates at a 10 megabit data rate, is MC68000 and MC68010 compatible, has: a 16-bit data bus, a multiplexed address/data bus, a DMA controller with 24-bit addressing and a 48-byte data buffer with buffer management. It also offers diagnostic aids, three modes of destination address comparison, executes a CSMA/CD network access algorithm and provides extensive error reporting.

LANCE diagnostics include:

- Pseudo full duplex capability for use in testing via incoming and outgoing loopback packets
- 32-bit CRC function usable in the transmission, reception and two loopback operating modes
- 10-bit wide time domain reflectometry counter used for determining the location of a cable fault.

Three modes of checking the received network destination address against initialization values are provided. These include:

- Physical mode: a comparison of address bits with corresponding bits in the physical address register
- Logical address filter using as an index the CRC value determined over the destination address
- Promiscuous mode: in which all packets are received regardless of address.

The LANCE executes the full carrier sense multiple access with collision detection (CSMA/CD) algorithm in

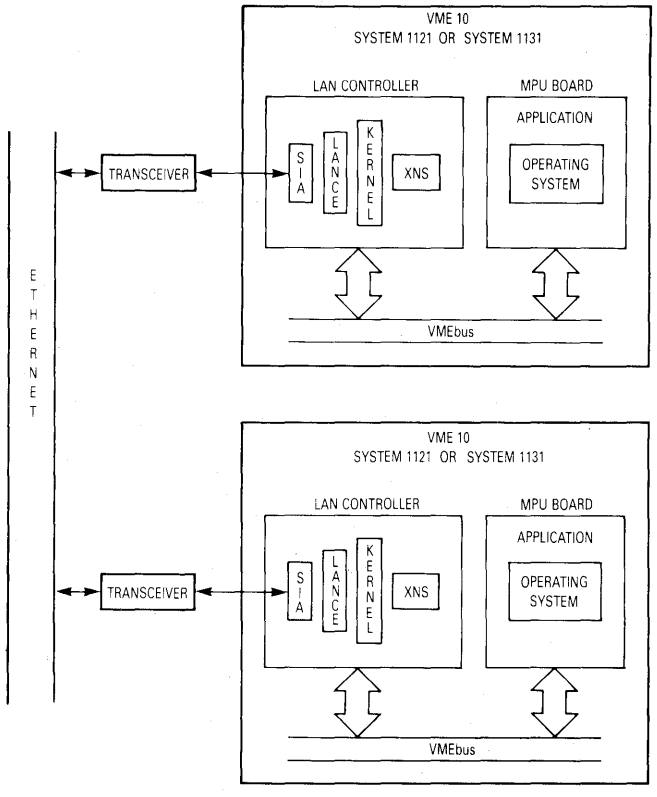


Figure 3. Two Node Network

which on detecting collision it transmits Collision Jam data and performs a backoff algorithm before again attempting to transmit. Only after sixteen consecutive collision detections does the LANCE report an error. Individual packet errors reported by LANCE include: CRC error, framing error and SILO under- or overflow.

LANCE reported errors which result in a controller generating an interrupt on the VMEbus are: babbling transmitter (transmission of more than 1518 bytes), nonfunctional collision detection circuitry, missed packet due to insufficient buffer space and memory timeout.

Host/LANCE communications are accomplished via transmit and receive ring structures in memory. Each ring is a circular queue comprising up to 128 message descriptors four words long. Each descriptor defines a LANCE — controlled buffer or chain of buffers holding a packet awaiting further LANCE processing. Orderly management of the message descriptors is contingent on use of a bit in each descriptor whose state indicates ownership by LANCE or host of that descriptor and on strict observance of a protocol which allows LANCE and Host to relinquish but never take ownership and forbids changes to data in non-owned descriptors.

SERIAL INTERFACE ADAPTER (SIA)

The SIA performs the Manchester encoding/decoding necessary for interfacing LANCE to Ethernet. It is compatible with standard Ethernet bus transceivers operating at 10 megabits/sec. The SIA decoder acquires the clocks and data within six bit times (600 ns). It features guaranteed carrier detection and collision detection threshold limits and transient noise rejection. The receiver decodes Manchester data in the presence of up to plus or minus 20 ns clock jitter, which represents ⅛ of a bit time.

VMEbus REQUESTER/BUS MASTERSHIP

Since each controller has bus request circuitry which complies with the VMEbus specification, it can act as bus master in a system having an arbiter elsewhere on the bus. It supports all four bus request levels, the bus request in/bus grant out daisy chain and provides jumper selection of the module bus request level.

VMEbus INTERRUPTER

The interrupter circuitry on each LAN Controller complies with the VMEbus specification, supporting the interrupt acknowledge daisy chain and allowing jumper selection for the module of one of the seven interrupt priority levels.

TIMER

For use by protocol processing software timers, each LAN Controller has a timer which causes the on-board MPU to be interrupted every 2 ms.

MEMORY MAP

Decoding of the upper seven address lines and a header permit jumper selection of a base address for the LAN Controller on 128K byte boundaries throughout the VMEbus memory space. As bus master, MVME330 can access most of the bus address space.

POWER UP TEST

Upon power up, or system reset, the LAN Controller executes a series of ROM-based self tests to determine that the board is functioning properly. Upon successful completion of these, the fail LED is turned off. Tests include a LANCE register and loopback test, an MPU test, a memory test for the dynamic memory, EPROM checksum test, and a status and control register test (I/O registers).

MVME330 FAMILY SOFTWARE

A controller module is supplied with host-specific software/firmware offering the functionality and performance for immediate use and for implementing custom applications. The software supports the Motorola SYSTEM V/68 or the VERSAdos operating systems and provides a communications executive kernel which controls the LANCE hardware in its performance of basic Ethernet data transfer, status reporting, statistical and diagnostic functions.

The software is designed to interconnect VME/10-SYSTEM 1121-, and SYSTEM 1131-based systems via Ethernet. Device drivers running under the Motorola SYSTEM V/68 or the VERSAdos operating systems are also provided. The software supports only MMU-based systems.

KERNEL FUNCTIONS

The LAN Controller kernel performs all functions required for controlling and monitoring MVME330 family hardware. Included are: managing LANCE status registers and LANCE-controlled message descriptor rings, performing timer functions, managing interrupts and retrieving LANCE generated statistics.

APPLICATION PACKAGES

The MVME330 LAN Controller family comes with a powerful set of application packages. These include:

- File Transfer: Files can be moved between two VME/10, SYSTEM 1121 and SYSTEM 1131 hosts on the network running under VERSAdos or SYSTEM V/68.
- Virtual Terminal: From the terminal of one host, a user via Ethernet can log on a remote host and execute commands and programs on that computer as though directly connected. At present; only Motorola SYSTEM V/68 to Motorola SYSTEM V/68 and VERSAdos to VERSAdos communications are supported.

FILE TRANSFER CAPABILITIES

Files are transferred between nodes on Ethernet using the File Transfer application package. At present the following services are provided:

- File Management
- File Access
- File Protection, on both the SYSTEM V/68 or the VERSAdos operating systems.

PROTOCOLS FOR HIGHER LAYER SOFTWARE

As high layer software support for the MVME330 LAN Controller family, Motorola offers a XEROX Network Service (XNS) implementation consisting of the following protocols:

- Internetwork Datagram Protocol — corresponding to layer 3 of the OSI model.
- Transport Layer Protocol — corresponding to layer four of the OSI model, this LAN-resident protocol handles echo, error and sequenced packet.

OPERATING SYSTEMS INTERFACES

A MVME330 family LAN Controller provides a bus interface for VMEbus in the form of shared memory protocol. This interface provides access to Ethernet from Motorola SYSTEM V/68 or the VERSAdos Operating Systems via the corresponding driver supplied with the software.

SYSTEMS INTEGRATION

A basic download capability useful for integrating system software with system hardware is provided with the MVME330 family controllers.

In conjunction with other kernel software; the download function can assist with board diagnostics and program development by allowing the required software to be downloaded.

Statistics are supported by the kernel which can be utilized as applications require. Supported statistics are listed in Table 1. Supported statistics are tallied from the last reset of the statistics or of the LAN Controller.

Table 1. Network Statistics

Ethernet messages transmitted	Collision errors
Multiple Retries reported	Memory errors on transmit
Single Retries reported	Ethernet messages received
Deferrals reported	Missed packets reported
Transmit buffer error	CRC errors reported
Silo underruns	Framing error
Late collisions	Silo Overruns
Carrier loss	Memory errors on receive
Babbling Transmitter errors	

NETWORK PERFORMANCE

The system throughput, including upper layer XNS software, is approximately 130 packets/second, for packets of 1024 bytes, average. Each controller can handle multiple back-to-back packets of any length.

Sustained system throughput of the MVME330 has been measured at the following speeds using "NTST" and "NSTAT" (supplied) in packets per second. "NSTAT" measures the Network Layer while "NTST" measures the Application Layer.

Example: (« ntst net xdg <packet size> 10000 nstat 10

	Network Layer	Application Layer
64 byte packets	185	165
1458 byte packets	145	120

Configuration: System 1121, MVME330-2, VERSAdos 4.5, idle network.

MVME330 FAMILY SPECIFICATIONS

The specifications for the MVME330 family are listed in Table 2.

NETWORK DISTANCES

The maximum distance between any two receivers is 2500 meters. The maximum length of the transceiver cable (from the LAN Controller to the Ethernet cable) is 50 meters.

SYSTEMS NETWORK CONCEPT

Figure 4 is an overall concept of an Ethernet system involving the various Motorola Microsystems hardware and software elements described in this data sheet.

STARTER KITS

For SYSTEM V/68 and VERSAdos system users wishing to begin networking quickly or who desire to learn local area network concepts, the components for a two node Ethernet network are offered. All required software, hardware and documentation can be obtained.

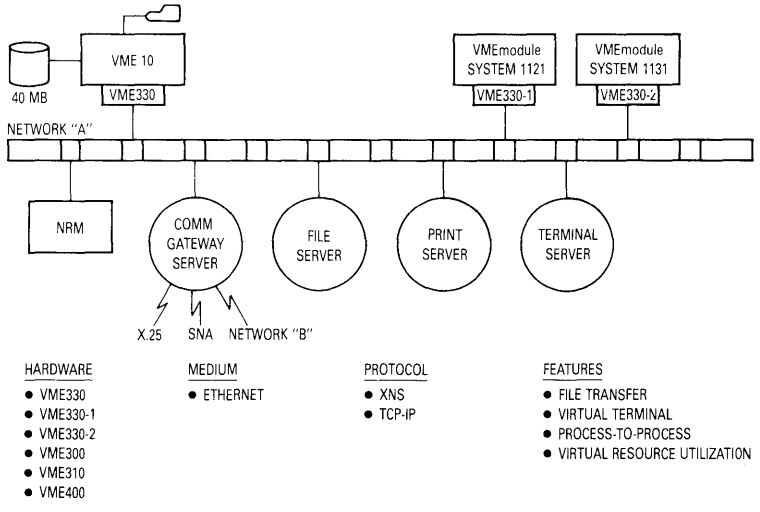


Figure 4. Systems Network Concept

2

MVME330 FAMILY SPECIFICATIONS

Characteristic	Specification
Power Requirements	3.8 A @ +5 Vdc ±5% 0.6 A @ +12 Vdc ±5% 0.1 A @ −12 Vdc ±5%
Environmental Tolerance	
Operating Temperature	5°C to 50°C
Storage Temperature	−40°C to 85°C
Humidity Range	0% to 95% (non-condensing)
Mechanical Specifications	
Height x Depth (board)	9.2" (234 mm) x 6.3" (160 mm)
Height x Width (front panel)	10.3" (262 mm) x 0.79" (20 mm)
Connectors	
VMEbus	DIN#41612C96
Ethernet transceiver port cable	AMP745094-1

VMEbus CONNECTOR P1

The electrical and mechanical characteristics of VMEbus connector P1 are fully described in the VMEbus Specification Manual, HB212.

ETHERNET TRANSCEIVER PORT CONNECTOR

The Ethernet transceiver port connector is a 15-pin subminiature for Ethernet 2.0.

ORDERING INFORMATION

A MVME330 family LAN Controller is ordered as a standalone board which includes the executive kernel software; or as a package including the board, executive kernel, a complete set of XNS protocol and network application software, and appropriate host-specific software including device driver and the BIV bus interface control layer. The entire software package (less kernel) may be purchased separately.

Part Number	Description
MVME330	VMEmodule Ethernet LAN Controller. This module provides high performance, intelligent single board connection of VMEbus Systems to Ethernet, a Local Area Network. Includes 128Kb RAM, LANCE (7990), SIA (7992), MC68000 MPU, Kernel Firmware and Power up self-test.
MVME330-1	VMEmodule Ethernet LAN Controller. This module provides high performance, intelligent single board connection of VMEbus Systems to Ethernet, a Local Area Network. Includes 512Kb RAM, LANCE (7990), SIA (7992), MC68000 MPU, Kernel Firmware and Power up self-test.
MVME330-2	VMEmodule Ethernet LAN Controller. This module provides high performance, intelligent single board connection of VMEbus Systems to Ethernet, a Local Area Network. Includes 512Kb RAM, LANCE (7990), SIA (7992), MC68010 MPU, Kernel Firmware and Power up self-test.
MVME330-UX*	MVME330 plus appropriate XNS software on SYSTEM V/68 file system 5-1/4" floppy disk.
MVME330-VX*	MVME330 plus appropriate XNS software on VERSAdos file system 5-1/4" floppy disk.
MVME330-1UX*	MVME330-1 plus appropriate XNS software on SYSTEM V/68 file system 5-1/4" floppy disk.
MVME330-1VX*	MVME330-1 plus appropriate XNS software on VERSAdos file system 5-1/4" floppy disk.
MVME330-2UX*	MVME330-2 plus appropriate XNS software on SYSTEM V/68 file system 5-1/4" floppy disk.
MVME330-2VX*	MVME330-2 plus appropriate XNS software on VERSAdos file system 5-1/4" floppy disk.
MVMELANSKIT	Ethernet LAN starter kit includes: (2) Ethernet transceivers (2) Transceiver cables (5 meters) (2) Coaxial terminators (1) Ethernet coax cable (24.6 meters) (1) Installation kit (1) Tap block Associated documentation
MVMESKIT-VX*	MVMELANSKIT, plus (2) MVME330's VERSAdos XNS on 5-1/4" floppy MVME330 User's Manual XNS Manual
MVMESKIT-UX*	MVMELANSKIT, plus (2) MVME330's SYSTEM V/68 XNS on 5-1/4" floppy MVME330 User's Manual XNS Manual

*NOTE: For any of the above part numbers that specify XNS software, the object code modules supplied include:

— XNS protocol package including Echo, Error, Sequenced Packet and Datagram.
— Network Application Software including File Transfer, Network Utilities, Runtime Library and Virtual Terminal.
— Host Specific Network Software including a Device Driver for the MVME330.
— Software documentation, and User's Manual.

So that it can be reconfigured without need for source code, object code is supplied as bootable load modules, and unlinked modules.

MVME331
MVME705

VMEmodule™ Six-Channel Synchronous/ Asynchronous Communications Controller

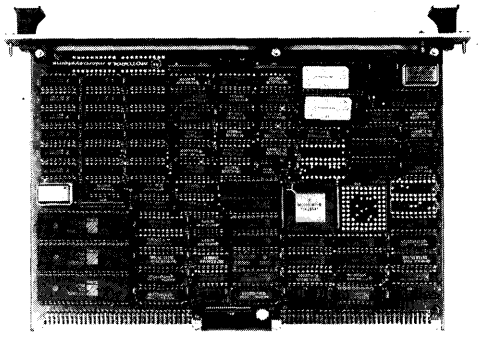
- Intelligent Communications Controller (ICC)
- Six RS-232-C or RS-422 Ports (Selection on separate I/O Transition Module)
- Data Transfer Modes:
  - Synchronous 1 Mbit/s, Single Channel
  - Asynchronous 78.6K Baud, Full Duplex, Single Channel
  - 9600 Baud, Full Duplex, 6-Channels, Concurrent
- 128Kb Dynamic RAM (Zero CPU Wait Cycles)
- 10 MHz, MC68010 16-/32-Bit Microprocessor
- Three Z8530 Serial Communications Controllers
- Double High Eurocard Form Factor
- VMEbus Compatible
- Drivers for VERSAdos and SYSTEM V/68 (UNIX) available

The MVME331 Intelligent Communications Controller offers the control and interfaces for six high performance, asynchronous or synchronous serial channels. A microcomputer comprised of an MC68010 Microprocessor, 128Kb of dynamic RAM and a control program in 32Kb of EPROM is used to free a system host of most of the serial communications overhead and achieve high data transfer rates. Each serial port can operate individually at a synchronous data rate of up to one megabit per second or asynchronously at up to 76.8 kilobaud, full duplex. A block diagram of the controller is shown in Figure 1.

The controller is designed for use in VMEbus-based systems requiring high performance. Because it handles all I/O related functions locally without interacting with the host processor, system performance is increased and applications can be more modular and flexible. The controller checks, translates and individually processes received and transmitted characters. It also performs, autonomously, typical peripheral control functions such as echo, software or hardware handshake, insertion and deletion of control characters and line editing. All communication protocol parameters for the six ports are dynamically reconfigurable under software control, providing the maximum in application versatility.

The MVME331 has VMEbus control and status registers which allow a host to send interrupts and commands to the controller and the controller to provide status information to a host.

The ICC firmware contains self-test routines which are executed after system reset. Controller malfunctions are indicated on the front panel by a FAIL LED.



Typical applications for the MVME331 include terminal connection in multiuser systems, connection of asynchronous devices (e.g., printers, PROM programmers), links between computer systems and all types of data acquisition in control environments.

USER INTERFACE

A system host communicates with the MVME331 via an easy-to-use interface that is virtually operating system and serial device independent. Macro commands are sent through a pipeline structure in system memory which is shared by the host and the controller. The commands are retrieved and executed by the controller and a report of execution results is sent to the host through another, similar structure. This software interface complies with the standard Motorola Buffered Pipe Protocol for interprocessor communications.

VMEbus/MVME331/SERIAL DEVICE INTERCONNECTION

VMEbus/MVME331 interconnection is through rows A & C of the rear P2 connector. Signals for all six channels are brought to P2 at TTL levels and require off-board translation of these levels to standard communication values (RS-232-C, RS-422-B or other). Changing to a different protocol thus requires only a new external level shifter — with no change to the MVME331 module or to software. The user may provide this via a bulkhead connector panel or by means of the available transition modules.

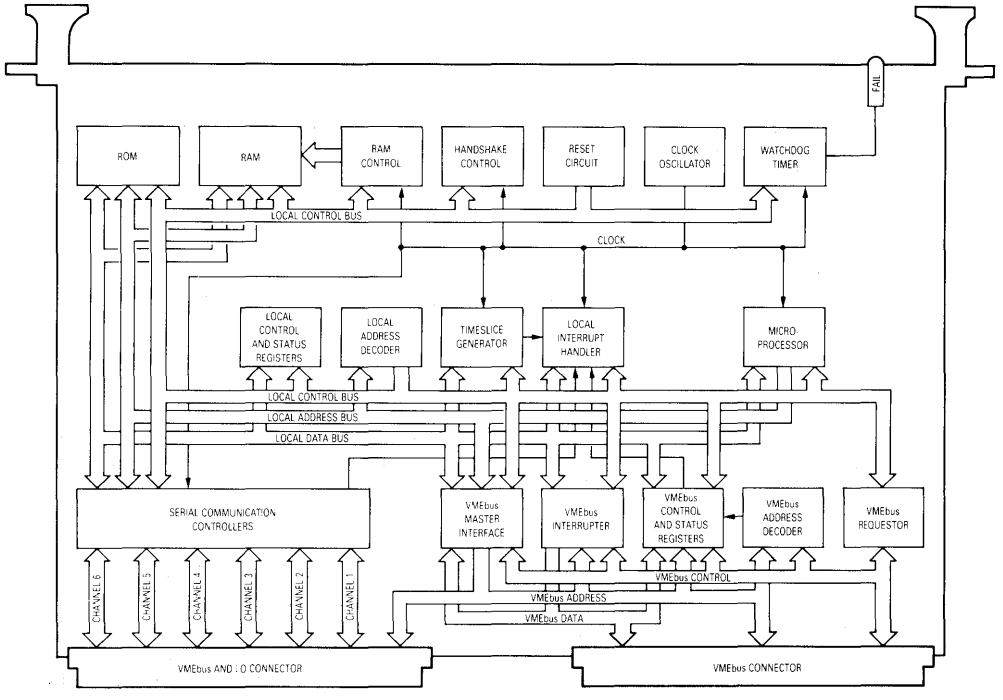
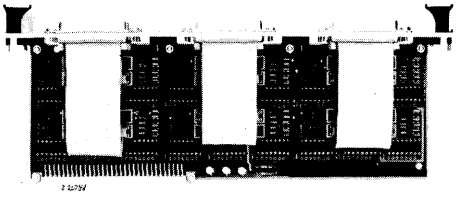


Figure 1. MVME331 Block Diagram

MVME705 COMMUNICATIONS TRANSITION MODULE

By means of a separate MVME705 module, any of the six serial channels can be individually configured for RS-232-C or RS-422-B operation. The MVME705 module provides the receiver and trasmitter circuits for converting the input/output signals of the controller to conform to the requirements of the desired standard. Figure 2 is a block diagram of the MVME705 module.

A transition module is connected to the MVME331's P2 connector via a 64-conductor flat ribbon cable having DIN41612 C64 connectors at each end. Six standard 25-pin subminiature "D" connectors are provided on a transition module's front panel for attaching serial peripheral devices. Each serial port can be configured independently as data circuit terminating equipment (DCE) or as data terminal equipment (DTE) by means of jumpers on the transition module. Tables 1 through 3 provide, for each configuration, the RS-232-C and RS-422-B pin/signal correspondences at the P2 and front panel connectors.



Configuring a channel on the MVME705 for RS-232-C or RS-422-B operation is done by installing in the IC sockets on the module the appropriate transmitter/receiver devices (supplied with the MVME705). The MVME705 package contains all devices required for either operating standard. In addition, the module includes locations in which the user can install capacitors for slew-rate limitation and noise filtering (for RS-232-C) or resistors for balanced line termination (for RS-422-B).

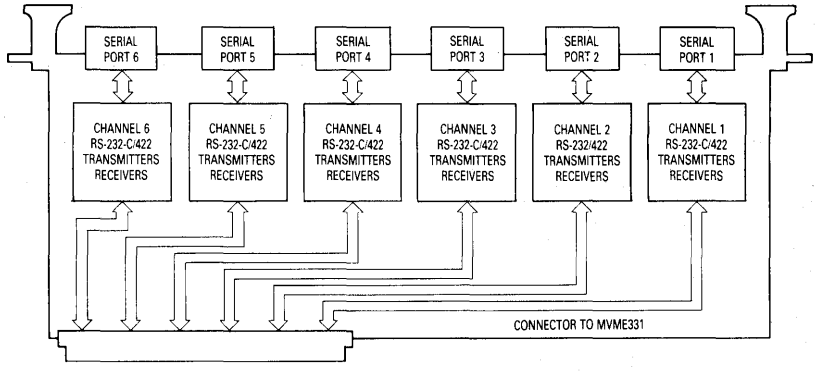


Figure 2. MVME705 Functional Block Diagram

Table 1. Serial Port in RS-232-C DTE Configuration

Pin	Signal	Direction	Connected With
2	TXD	output	P2 TXD
3	RXD	input	P2 RXD
4	RTS	output	P2 RTS*
5	CTS	input	P2 CTS*
6	DSR	input	P2 DSR*
7	GND	—	P2 GND
8	DCD	input	P2 DCD* or open
15	TXCI	input	P2 TRXC* or open
17	RXCI	input	P2 RTXC*
20	DTR	output	P2 DTR*
24	TXCO	output	PT TRXC* or open

Table 2. Serial Port in RS-232-C DCE Configuration

Pin	Signal	Direction	Connected With
2	TXD	input	P2 RXD
3	RXD	output	P2 TXD
4	RTS	input	P2 CTS*
5	CTS	output	P2 RTS*
6	DSR	output	P2 DTR*
7	GND	—	P2 GND
8	DCD	output	+ 12 Volts
15	TXCI	output	P2 TRXC* or open
17	RXCI	output	P2 TRXC* or open
20	DTR	input	P2 DSR*
24	TXCO	input	P2 RTXC*

Table 3. Serial Port in RS-422-B Configuration

Pin	Signal	Pin	Signal	Direction	Direction With
2	−TXD	15	+TXD	output	P2 TXD
3	−RXD	16	+RXD	input	P2 RXD
4	−RTS	17	+RTS	output	P2 RTS*
5	−CTS	18	+CTS	input	P2 CTS*
6	−DSR	19	+DSR	input	P2 DSR*
7	GND			—	P2 GND
8	−DCD	21	+DCD	input	P2 DCD* or open
9	−DTR	22	+DTR	output	P2 DTR*
10	−TXCI	23	+TXCI	input	P2 TRXC* or open
11	−TXCO	24	+TXCO	output	P2 TRXC* or open
12	−RXCI	25	+RXCI	input	P2 RTXC*

COMMUNICATIONS CONTROLLER MEMORY

Local memory consists of 32Kb of EPROM and 18 64Kb RAM devices providing a capacity of 128Kb with parity. The EPROMs contain the firmware resident ICC program that is copied to and executed from RAM following power-up reset. The RAM also holds the intermediate I/O data to be processed and transferred. Data is written into and read from RAM without MPU wait states. The generation and checking of byte parity is performed automatically during RAM accesses and a detected parity error causes generation of an autovectored level 7 interrupt to the MPU. Wrong parity can be forced for test purposes by setting the WWP* bit in the module control register. Refresh of the dynamic RAM is automatic with local arbitration logic controlling the access/refresh cycles.

The two provided ROM sockets, one for odd and one for even addresses, can accept a maximum of 128 kilobytes of firmware. Means are included for jumper configuration to accommodate 2764, 27128, 27256 or 27152 type EPROM devices, having access times of 250, 350, or 450 nanoseconds.

VMEbus INTERFACE

For data interchange with a host on the MVMEbus, the controller utilizes an A32:D16 or A24:D16 master interface supported by address decoding logic, address modifier registers and an option Release When Done (RWD) bus requester. System memory accesses are controlled by the option RWD bus requester which can be set by jumper to any one of the four priority levels. Data transfers are supervised by a bus timer with a jumper selectable period. Two access modes are provided: the window addressing mode in which a 4Mb local address map is used as a window to the VMEbus and the alternate space addressing mode in which an address extension register is used to obtain mapping of a contiguous 16Mb address space anywhere within the full 4 gigabyte VMEbus range.

Interrupt requests from the controller to a system host are generated by a VMEbus interrupter having provision for a programmable status/ID byte. When the bus is acquired for MPU accesses, the requester releases on the

VMEbus INTERFACE (Cont.)

next local data transfer. The controller status register contains a bit reflecting the state of the interrupt request output which can be polled by the controller MPU.

An option A16:D16 slave interface accommodates accesses from the VMEbus of the ICC control and status registers. These occupy, in the short I/O address space, a single word which appears redundantly throughout a 128 word segment. By jumper, the segment can be placed on any 256 byte boundary relative to the I/O space base address. A write operation accesses the VME control register. A read operation accesses the VME status register.

Short supervisory and non-privileged access of the short I/O space is provided by a bipolar PROM which decodes address modifier codes $29 and $2D. The PROM may be reprogrammed if use of other codes is desired.

Tables 4 and 5 identify, respectively, the connectors P1 and P2 pin/VMEbus signal assigments.

Table 4. Connector P1 VMEbus Signal Locations

Pin No.	Row A Signals	Row B Signals	Row C Signals	Pin No.
1	D00	BBSY*	D08	1
2	D01	(BCLR*)	D09	2
3	D02	(ACFAIL*)	D10	3
4	D03	BG0IN*	D11	4
5	D04	BG0OUT*	D12	5
6	D05	BG1IN*	D13	6
7	D06	BG1OUT*	D14	7
8	D07	BG2IN*	D15	8
9	GND	BG2OUT*	GND	9
10	(SYSCLK)	BG3IN*	SYSFAIL*	10
11	GND	BG3OUT*	BERR*	11
12	DS1*	BR0*	SYSRESET*	12
13	DS0*	BR*	LWORD*	13
14	WRITE*	BR2*	AM5	14
15	GND	BR3*	A23	15
16	DTACK*	AM0	A22	16
17	GND	AM1	A21	17
18	AS*	AM2	A20	18
19	GND	AM3	A19	19
20	IACK*	GND	A18	20
21	IACKIN*	(SERCLK)	A17	21
22	IACKOUT*	(SERDAT)	A16	22
23	AM4	GND	A15	23
24	A07	IRQ7*	A4	24
25	A06	IRQ6*	A13	25
26	A05	IRQ5*	A12	26
27	A04	IRQ4*	A11	27
28	A03	IRQ3*	A10	28
29	A02	IRQ2*	A09	29
30	A01	IRQ1*	A08	30
31	−12 V	(5 V STBY)	+12 V	31
32	+5 V	+5 V	+5 V	32

Note: Signals in parentheses are not used by the MVME331.

Table 5. Connector P2 Signal Locations

Pin No.	Row A Signals	Row B Signals	Row C Signals	Pin No.
1	+5 V	+5 V	+5 V	1
2	−12 V	GND	+12 V	2
3	TRXC6*	(RESERVED)	GND	3
4	DTR6*	A24	DCD6*	4
5	CTS6*	A25	DSR6*	5
6	RXD6	A26	RTS6*	6
7	TXD6	A27	RTXC6*	7
8	TRXC5*	A28	GND	8
9	DTR5*	A29	DCD5*	9
10	CTS5*	A30	DSR5*	10
11	TXD5	A31	RTS5*	11
12	TXD5	GND	RTXC5*	12
13	TRXC4*	+5 V	GND	13
14	DTR4*	(D16)	DCD4*	14
15	CTS4*	(D17)	DSR4*	15
16	RXD4	(D18)	RTS4*	16
17	TXD4	(D19)	RTXC4*	17
18	TRXC3*	(D20)	GND	18
19	DTR3*	(D21)	DCD3*	19
20	CTS3*	(D22)	DSR3*	20
21	RXD3	(D23)	RTS3*	21
22	TXD3	GND	RTXC3*	22
23	TRXC2*	(D24)	GND	23
24	DTR2*	(D25)	DCD2*	24
25	CTS2*	(D26)	DSR2*	25
26	RXD2	(D27)	RTS2*	26
27	TXD2	(D28)	RTXC2*	27
28	TRXC1*	(D29)	GND	28
29	DTR1*	(D30)	DCD1*	29
30	CTS1*	(D31)	DSR1*	30
31	RXD1	GND	RTS1*	31
32	TXD1	+5 V	RTXC1*	32

Note: Signals in parentheses are not used by the MVME331. I/O signals are at TTL levels.

PERIPHERAL INTERFACE

Asynchronous and synchronous transfer of data through the serial channels is effected using three Z8530 Serial Communication Controller (SCC) devices. Each offers two independent, full duplex serial channels, is capable of transferring data at rates up to 1 Mbit per second over a single channel and provides the hardware support for the software control of communication modes and protocols. Control provided by the SCC devices allows the independent selection of an encoding mode and the programming of parameters for synchronous or asynchronous operation of any serial channel.

Signals at the TTL level are provided at the VMEbus P2 connector and, by means of a separate communications transition module which also provides the serial port connectors, are translated to the levels and connections required by the RS-232-C or RS-422-B communications standards. The communications signal assignments to pins in connector P2 are identified in Table 5.

The SCC devices are clocked at 4.9152 MHz. By division of this frequency, all baud rates for serial data transfer are generated.

ASYNCHRONOUS OPERATION

When asynchronous data transfer is required, the following parameters can be programmed:

— baud rate from 50 to 76800
— odd, even or no parity
— 1, 1.5 or 2 stop bits
— character length of from 5 to 8 bits
— NRZ, NRZI or FM data encoding

SYNCHRONOUS OPERATION

Transfer of both bit and byte data is supported and the SCC devices can be programmed for a number of different synchronization modes.

Both monosync and bisync modes with or without CRC checking can be used for byte data transfer.

The SDLC or the HDLC protocol with automatic CRC generation and checking can be used for the transfer of bit data. An underrun causes generation of an interrupt to the MPU.

For the synchronous transfer of byte or bit data, odd, even or no parity may be programmed and NRZ, NRZI or FM coding may be selected.

COMMUNICATIONS CONTROLLER TIMER/COUNTERS

The MVME331 Communications Controller contains three timer/counters used for periodic interrupt generation, for the monitoring of module malfunctions and for supervising accesses of the module's data transfer bus and accesses of the VMEbus.

TIMER-SLICE COUNTER

To facilitate the generation of firmware which takes advantage of the powerful data transfer capabilities offered by the serial communications controllers, a programmable timer is included which is used for the generation of periodic interrupts to the module MPU as a basis for the time-slice driven resident executive.

A value is loaded into an 8-bit register, counted down to zero and the resulting output used to generate an autovectored level 2 interrupt request. The interrupt request is latched and reset when acknowledged by the MPU.

Clocked at 78.125 kHz, the counter offers time periods ranging between 3.25 and 12.8 microseconds. The counter can also be disabled and restarted by program instructions.

WATCHDOG TIMER

To provide user software a means of indicating fatal MVME331 malfunctions, a timer with a period of 100 milliseconds is provided on the controller module. On timing out, this timer sets the fail bit in the status register, lights the fail LED on the module front panel and may be jumpered to cause assertion of the VMEbus SYSFAIL* line. User software prevents timeout of the counter by simply accessing the watchdog timer reset location.

DATA TRANSFER BUS TIMER

For supervision of all MPU data transfer cycles on both the local bus and the VMEbus, the MVME331 module includes a timer which generates a local bus error signal if an accessed device fails to respond with a data transfer acknowledge signal before the counter times out. The timeout period is fixed at 25 microseconds for all accesses to local devices. For VMEbus accesses, a timeout period of 25, 50 or 100 microseconds is jumper selectable. The selected period should exceed the greatest time expected to elapse between the assertion of a bus request and the reception of a bus grant in the actual system configuration.

HOST/CONTROLLER COMMUNICATIONS

In order for two intelligent devices to communicate with each other, a channel must exist. In a bus-based system, two kinds of channels are commonly used: physical and logical.

Operation of a physical channel depends on memory which can be shared by the two devices. Often this is dual-ported RAM located on a peripheral device controller in which the host CPU writes commands for execution and in which the controller reports the results of executing the commands. Coherent use of the dual-ported memory as a communications channel generally requires a buffer with its own semaphore register for passing commands and a separate buffer/semaphore register for reporting status.

The other type of communications channel — logical — depends on the sharing of a region of system RAM for the exchange of messages. This concept offers the significant advantage of not being constrained by the size of dual-ported RAM on board an intelligent module. Motorola's Buffered Pipe Protocol facilitates the implementation of logical communications channels.

THE BUFFERED PIPE PROTOCOL

Motorola's Buffered Pipe Protocol defines a communications channel which uses two queues of the FIFO type in system RAM for the exchange of commands and status reports between a system host and an intelligent peripheral device. The buffered pipe implementation exhibits the following features:

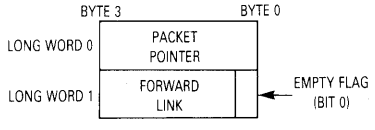
- Broadly Usable — Virtually Operating System and Board Independent
- Supports both Interrupt-driven and Polled Modes of Operation
- Minimizes Hardware — Imposes only Ability to Access System RAM and Interrupt VMEbus Devices
- Non-busy Interface — Always Accessible
- Infinite Queue — Virtually no Message Quantity Restriction
- No Worst Case Handling — Send/Receive Using One Small Set of Instructions
- Simple — Uses a Few Simple Procedure Calls and One, Simple Data Structure
- Symmetrical — Uses Same Mechanism for Virtually all Communication Between Two Processors
- Tolerant — Does Not Restrict Message Content in any Way
- Flexible — Processors Receive Messages in Sent Order but are not Required to Process them in Sent Order

PROTOCOL ELEMENTS

There are four principal elements utilized in the buffered pipe protocol, as shown in the diagrams: a put pointer associated with a sender, a get pointer associated with a receiver, the Buffered Pipe Envelope (BPE) and the command/status/data packet. An envelope is simply two pointers: one to the next envelope and one to the packet associated with the current envelope. The packet is a fixed length data structure residing anywhere in system memory and containing commands, status reports or other parameters. To ensure that a new envelope can be queued by a processor at any time, the protocol prescribes that the last envelope have no associated packet, that its forward link act also as the empty envelope flag by pointing to an odd address (rather than the even address at which a full envelope is required to reside) and that no receiver dequeue this last envelope from the pipe.

CHANNEL INITIALIZATION

Two buffered pipes are used for a MVME331 communication channel: one in which a system driver sends commands and configuration information to a device controller and one in which the controller sends status reports to the driver. Since the communication channel is interrupt driven, each interrupts the other following the queuing or dequeuing of an envelope. It is incument on the driver to initialize the channel and controller by: creating from system memory a pool of envelopes and a pool of packets, initializing a command pipe and a status pipe by placing an empty envelope in each, initializing the command get and status get pointers and passing these to the device controller using the Get Pointers command. This is a command supported by the controller firmware which uses interrupt handshaking for passing a sequence of bytes via the controller's control register. The number of packets required and processing overhead are both minimized by requiring the controller to return status in the same packet in which a command is received.



Buffered Pipe Envelope (BPE)

MVME331/DRIVER COMMUNICATIONS

Communications between a host driver and devices connected to MVME331 serial ports are supported by firmware in 32Kb of ROM onboard the controller module. This firmware observes the Motorola standard buffered pipe protocol for host/intelligent device communications and facilitates initialization and full runtime control of serial device communication modes, codes and protocols. The versatile control and character processing provided by the MVME331 firmware and hardware relieves a system host of much of the overhead imposed when six asynchronous communication channels are operated concurrently.

The MVME331 firmware architecture utilizes a time slice driven, multitasking executive to control read and write channels each of which has a buffer, a dynamic queue and three actions tables. The buffers reduce loading of the VMEbus by allowing transmission of 16 byte blocks for a single bus arbitration cycle and the 256 byte queues aid Xon/Xoff control of the rate of data flow.

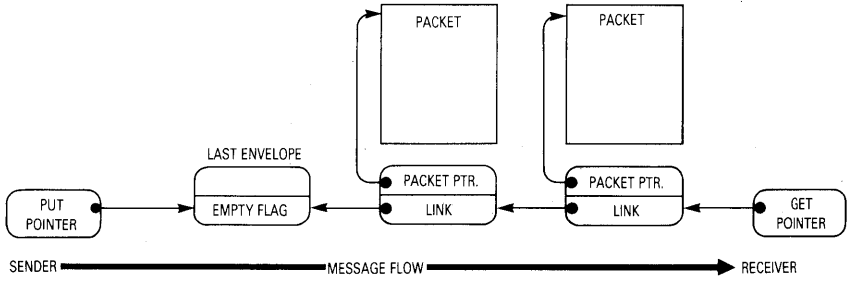
Overall control of the data flow is implemented by using tables to perform character translation so that different codes can be accommodated, to provide control character services, to handle escape key sequences, and to provide editing functions like echo/delete character or cancel line. The tables provide great versatility since to the standard functions provided by the firmware, a user can add control routines to each table to comply with the specific requirements of nearly any application.

READ CHARACTER FLOW

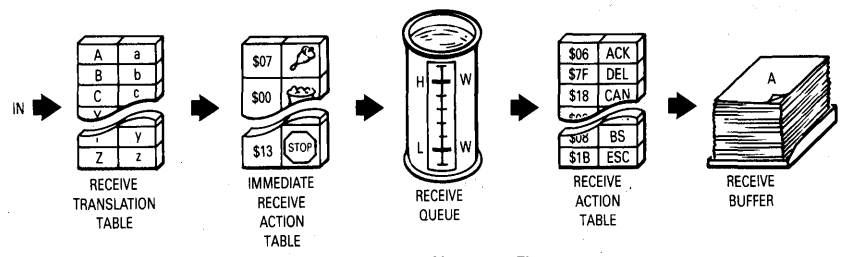
As the diagram shows, the read character flow is: serial port --> Receive Translation Table --> Immediate Receive Action Table --> Receive Queue --> Receive Action Table --> Receive Buffer --> VMEbus.

From a serial device and following any required code translation called for in the Receive Translation Table, characters are sent to the Immediate Receive Action Table where they're screened against values in the Immediate Receive Action Table. Ordinary data characters are passed to the Receive Queue but codes requiring an immediate action such as the use of Xoff to stop transmission on that channel are detected and acted on. The table is also used for other functions such as stripping fill characters so that they do not occupy queue space.

On leaving the Receive Queue, a character is screened against values in the Receive Action Table where other



Buffered Pipe Message Flow



MVME331 Read Character Flow

requirements are detected and acted on. Primarily, these include the support of editing functions such as deleting characters and canceling lines as well as the accommodation of any variance in escape sequences between different kinds of terminals. Before being placed on the VMEbus, characters are sent to the Receive Buffer which facilitates transmission of small bursts rather than single characters.

As the diagram shows, the write character flow is: VMEbus --> Transmit Buffer --> Write Action Table --> Transmit Action Table --> Transmit Queue --> Transmit Translation Table --> Serial Port.

WRITE CHARACTER FLOW

From the VMEbus, bursts of characters are stored in the Transmit Buffer, from which they march single file into the Write Action Table where they are screened. Ordinary data characters are sent directly to the Transmit Queue but a few non-data characters initiate an immediate action such as the sending of a line termination string after a carriage return character. All other non-data characters are sent to the Transmit Action Table where their corresponding editing functions and transmission requirements are identified and performed. From the Transmit Queue, data characters are moved to the Transmit Translation Table for code translation, if required, before they are transmitted through a serial port.

The two communication paths do not operate in total independence particularly when the mode is full duplex. The Receive Action Table, for instance, communicates directly with the Transmit Action Table on receiving a line termination code.

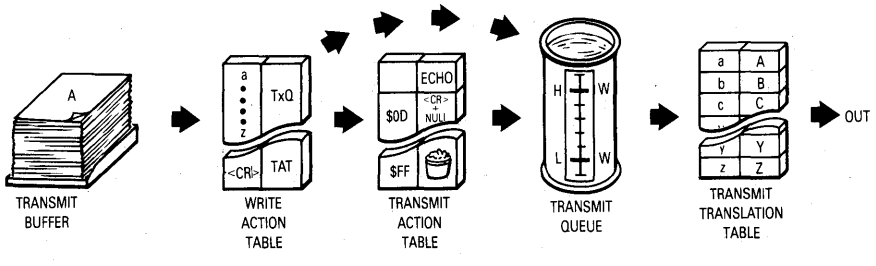
Codes for the standard commands supported by the MVME331 Intelligent Communications Controller (ICC) firmware are shown in Table 6. Codes for standard status messages used for reporting execution status for the commands of Table 6 are shown in Table 7. For a particular application, the user can add commands and status messages, as required.

Table 6. MVME331 Firmware Command List

Code	Command	Code	Command
$01	Read	$0C	Request Port Status
$02	Write	$0D	Request Table Status
$03	Abort	$0E	Reset Port
$04	Solicit Event	$0F	Send Break
$05	Configure ICC	$10	Flush Receive Queue
$06	Configure Port	$11	Flush Transmit Queue
$07	Configure Table	$12	Boot
$0B	Request ICC Status	$13	Go

Table 7. MVME331 Status Messages

Code	Terminating Condition	Code	Terminating Condition
$00	Normal command termination	$0A	Invalid type of parity
$01	End of system memory block	$0B	Invalid handshake selection
$02	Read timeout	$0C	Rx queue overrun
$03	Write timeout	$0D	Invalid delete character string length
$04	Command aborted	$10	Invalid start address
$05	Insufficient local memory	$14	Least number of characters received
$06	Invalid item number	$81	Parity error
$07	Invalid baud rate	$82	Overrun error
$08	Invalid number of bits per character	$83	Overrun and parity error
$09	Invalid number of stop bits	$84	Framing error
		$85	Overrun and framing error



MVME331 Write Character Flow

MVME331 SPECIFICATIONS

Characteristics	Specifications
General	The MVME331 Intelligent Communications Controller is a VMEmodule for interfacing six asynchronous serial communication channels to a VME system.
Performance	The MVME331 can simultaneously control six channels at 9600 baud full duplex with individual character translation and control code processing for each channel.
Local Microcomputer	The local microcomputer consists of a 10 MHz MC68010 microprocessor, 32Kb ROM, 128Kb RAM, interrupt handling logic, and various timers.
VMEbus Interface	The control and status registers on the ICC are accessed from the VME-bus through an option A16:D16 slave interface. The ICC accesses system memory through an option A32:D16 or A24:D16 master interface and supervises data transfers with a bus timer of variable length. The option RWD bus requester can be selected to operate on any one of the priority levels. The priority level and the status/ID byte of the interrupter are software programmable.
Peripheral Interface	Serial link control and data transfer are performed by three Z8530 Serial Communications Controller devices. Baud rates (50 to 76800), data characteristics (number of data bits and stop bits, even or odd parity) and control signal functions (DTR, DSR, RTS, CTS, DCD) are independently selectable for each of the six channels.
Operating System Interface	The transfer of commands, status messages and data packets is performed through a pipeline in VME system memory. This transfer follows the Motorola standard Buffered Pipe Protocol for processor to processor communication. Drivers are available for VERSAdos and SYSTEM V/68.
ICC Commands	The ICC commands include: — ICC configuration — Port configuration — Data read and write operations — Break transmission — Software download and start — Reset and abort functions
ICC Status Messages	The ICC status messages include: — Normal command completion — ICC and port status — Parity and framing errors — Break detection — Data overrun and buffer overflow — Read and write time-out — Invalid commands and parameters
Indicator	FAIL LED on the front panel
Power Requirements	+5 Vdc (±5%), 3.8 A typ, 4.4 A max
Operating Temperature	0° to 55°C Inlet air, forced air cooling
Storage Temperature	−40° to 100°C
Relative Humidity	0% to 90% (non-condensing)
Physical Dimensions Height Depth Front Panel	Double High VME board with front panel 9.2 in. (233 mm) 6.3 in. (160 mm) 10.3 in. (262 mm) 0.8 in. (20 mm)

MVME705 SPECIFICATIONS

Charateristics	Specifications
General	The MVME705 module provides transmitter and receiver circuits for the MVME331 Intelligent Communications Controller module. Both modules are interconnected with a 64-conductor flat-ribbon cable. The serial port signals are available at 25-pin connectors on the front panel.
Serial Port Characteristics	Each channel can be configured for the RS-232-C or RS-422-B EIA standard. The front panel connectors can be configured as data circuit terminating equipment or as data terminal equipment.
Supported Signals	The MVME705 supports the following signals: TXD — Transmitted Data RXD — Received Data RTS — Request To Send CTS — Clear To Send DSR — Data Set Ready DTR — Data Terminal Ready DCD — Data Carrier Detect TRXC — Transmitter-Receiver Clock RTXC — Receiver-Transmitter Clock
Additional User Options	The user can install capacitors for slew-rate limitation and noise filtering (for RS-232-C) and resistors for balanced line termination (for RS-422) on free locations on the board.
Physical Dimensions Height Depth Front Panel	Double-high/reduced depth VMEboard 9.2 in. (233 mm) 3.5 in. (80 mm) 10.3 in. (262 mm) x 1.57 in. (40 mm)
Connectors	One 64-pin DIN41612 connector to the MVME331. Six 25-pin Sub-D serial port connectors.
Power Requirements	+5 Vdc, 1.5 A (max) ±12 Vdc, 150 mA (max)
Operating Temperature	0°C to 55°C Inlet air, forced air cooling
Storage Temperature	−40° to 100°C
Relative Humidity	0% to 90% (non-condensing)

ORDERING INFORMATION

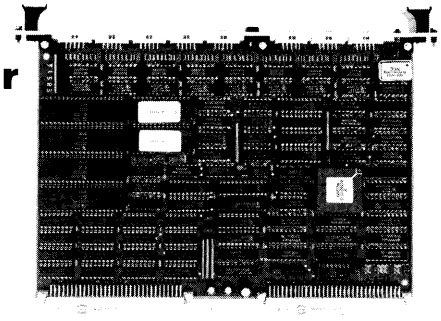
Part Number	Description
MVME331	Six-Channel Intelligent Communication Controller. Includes User's Manual.
MVME705	I/O Transition Communications Module with RS-232 or RS-422 interface. Includes User's Manual.
M68V1XS331DRV	VERSAdos Driver on 5-1/4" diskette (source)
M68N1XB331DRV	SYSTEM V/68 Driver on 5-1/4" diskette (binary)
M68N1XS331DRV	SYSTEM V/68 driver on 5-1/4" diskette (source)

RELATED DOCUMENTATION

HB212/D	VMEbus Specification Manual.

Product Preview

Eight-Channel Intelligent Communications Controller



- MC68010 16-/32-bit Microprocessor
- Double-High Eurocard Form Factor
- VMEbus Compatible
- Eight RS-232-C Ports on 4TE Front Panel (occupying one slot)
- I/O Signals Duplicated on Rear Connector
- 9600 Baud Simultaneously on all Eight Channels
- 76.8K Baud on Single Channel
- Drivers for VERSAdos and SYSTEM V/68 (UNIX) Available

The MVME332 Intelligent Communication Controller (ICC) provides an eight channel asynchronous interface between the VMEbus and serial peripheral devices. High performance is achieved using a complete microcomputer built around an MC68010 microprocessor and 128Kb RAM area, which speeds up the data transfer and relieves the system host processor from serial communication controlling tasks. Each port can operate individually at speeds up to 76.8K baud. When all eight asynchronous serial communication channels are simultaneously active they can operate at up to 9600 Baud half duplex.

Typical applications for the MVME332 include terminal connection in multiuser systems, connection of asynchronous devices (e.g. printers, PROM programmers), link between computer systems and all types of data acquisition in control applications.

The MVME332 is an intelligent VMEmodule which handles all I/O related functions onboard without interacting with a host processor thereby increasing overall system performance. The ICC isolates the I/O part of the system, allowing applications to be modular and flexible.

A VMEbus host communicates with the MVME332 using a standardized, I/O independent interface. The MVME332 receives macro commands through a pipeline structure in system memory, which is shared by the host and the ICC. The ICC firmware resident program interprets and executes the macro commands, and returns status messages through a second pipeline after completion. The transfer of commands and status messages follows the Motorola standard Buffered Pipe Protocol for inter-processor communication.

The ICC presents an easily manageable peripheral-independent interface to an operating system by individually checking, translating and pre-processing received and transmitted characters. It also executes, autonomously, typical peripheral control functions, such as software or hardware handshake, insertion and deletion of control characters, and echo and line editing functions.

The ICC firmware contains self-test routines which are executed after system reset. Controller malfunctions are indicated on the front panel by a FAIL LED.

All peripheral signals are available on the front panel (see Figure 1). At P2, all signals are duplicated with the exception of TRXC and RTXC which are omitted on four of the eight channels on the rear connector (TRXC and RTXC are not required for asynchronous operation).

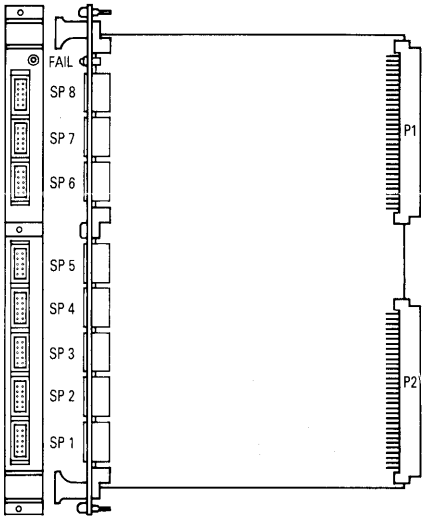


Figure 1. MVME332 Connector Locations

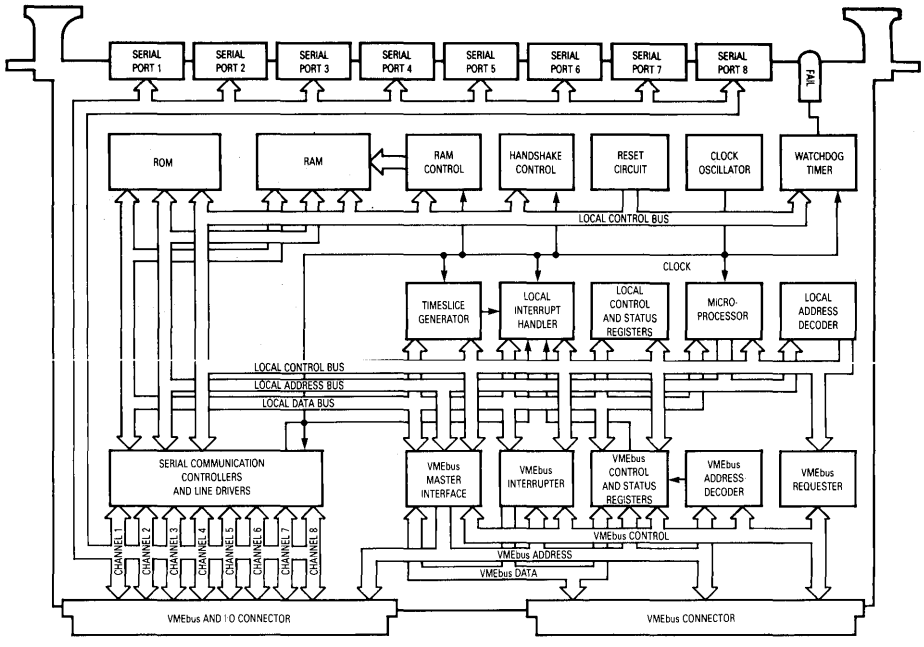


Figure 2. MVME332 Block Diagram

The user can connect cables to the MVME332 from the front or the rear of the VME rack. The standard method, using the connectors on the front panel, is the most compact solution, as board and connectors occupy only one slot. If the connection is made at the rear a flat cable is attached at P2. Various kinds of connectors may be attached to this cable. The signals at both the front and rear connectors are at RS-232-C levels.

HARDWARE DESCRIPTION

The microprocessing unit on the ICC is an MC68010, running at a clock speed of 10 MHz (see Figure 2). The microprocessor controls all devices on board the module through local data, address and control busses.

Local memory consists of 32Kb of EPROM and 128Kb of dynamic RAM. The EPROMs contain the firmware resident ICC program, that is copied to, and executed from RAM after power-up reset. The RAM also holds the intermediate I/O data to be processed and transferred.

The local microprocessor accesses VME system memory through an option A32:D16 or A24:D16 VMEbus master interface with software selectable address modifier codes. Bus arbitration is performed by an option RWD (release when done) bus requester on a jumper selectable priority level The VMEbus is never kept busy for more than eight data transfer cycles. The handshake control logic contains a bus timer which terminates an unsuccessful attempt to transfer data over the VMEbus after a jumper selectable time period.

Interrupts to a system host are generated by a VMEbus interrupter with priority level and status/ID byte being software selectable.

A control register is used for resetting and interrupting the ICC and for transmitting initialization information to the ICC. Another register holds status information. The VME system host processor accesses both registers through an option A16:D16 slave interface with the short I/O access address modifier code.

Four Z8530 Serial Communication Controller devices are used each providing two independent channels. These controllers perform the peripheral data transfer and data link control functions at the hardware level. Baud rates, data characteristics and control functions are software selectable.

The time-slice generator is a programmable counter which generates periodic interrupts to the local microprocessor. The ICC firmware requires these interrupts for its time-slice based executive concept.

The watchdog timer is a free running counter which is periodically reset by the ICC executive. Should the executive lose control, the watchdog counter will time out and turn on the FAIL LED on the front panel.

SOFTWARE DESCRIPTION

The Intelligent Communications Controller has dedicated firmware in local ROM for controlling eight asynchronous communication channels, with individual character processing and versatile control functions for each channel.

Character processing during input is controlled via three user tables, the Receive Translation Table (RTT), the Immediate Receive Action Table (IRAT) and the receive Action Table (RAT).

The equivalent tables for output are the Write Action Table (WAT), Transmit Action Table (TAT) and the Transmit Translation Table (TTT).

RTT and TTT can be used to translate from one standard to another (e.g. ASCII to EBCDIC) or from lowercase to uppercase. The translation tables can also be used to accommodate keyboards encoded for different languages.

IRAT and TAT can convert device-specific control characters for different operating systems. RAT acts upon defined characters in real-time and handles XON/XOFF handshaking, solicited event handling, or discards null pads. RAT and WAT perform editing functions such as cancel line, redisplay current line or previous line, delete character, read control character etc.

Use of the translation and action tables simplifies interfacing host computers from different manufacturers.

SOFTWARE-CONTROLLED PORT CONFIGURATION

Baud rates, number of bits per character, number of stop bits, parity type or other characteristics of the eight ports can be dynamically reconfigured using the RESET PORT and CONFIGURE PORT commands incorporated in the firmware.

Table 1. Connector P1 VMEbus Signal Locations

Pin No.	Row A Signals	Row B Signals	Row C Signals	Pin No.
1	D00	BBSY*	D08	1
2	D01	(BCLR*)	D09	2
3	D02	(ACFAIL*)	D10	3
4	D03	BG0IN*	D11	4
5	D04	BG0OUT*	D12	5
6	D05	BG1IN*	D13	6
7	D06	BG1OUT*	D14	7
8	D07	BG2IN*	D15	8
9	GND	BG2OUT*	GND	9
10	(SYSCLK)	BG3IN*	SYSFAIL*	10
11	GND	BG3OUT*	BERR*	11
12	DS1*	BR0*	SYSRESET*	12
13	DS0*	BR1*	LWORD*	13
14	WRITE*	BR2*	AM5	14
15	GND	BR3*	A23	15
16	DTACK*	AM0	A22	16
17	GND	AM1	A21	17
18	AS*	AM2	A20	18
19	GND	AM3	A19	19
20	IACK*	GND	A18	20
21	IACKIN*	(SERCLK)	A17	21
22	IACKOUT*	(SERDAT)	A16	22
23	AM4	GND	A15	23
24	A07	IRQ7*	A14	24
25	A06	IRQ6*	A13	25
26	A05	IRQ5*	A12	26
27	A04	IRQ4*	A11	27
28	A03	IRQ3*	A10	28
29	A02	IRQ2*	A09	29
30	A01	IRQ1*	A08	30
31	−12 V	(5 V STBY)	+12 V	31
32	+5 V	+5 V	+5 V	32

Note: Signals in parentheses are not used by the MVME332

Table 2. Front Panel Connectors (SP1–SP6)

Pin No.	Signal	Description	Pin No.	Signal	Description
1	RXD	Received Data	2	TXD	Transmitted Data
3	CTS	Clear To Send	4	RTS	Request To Send
5	DTR	Data Terminal Ready	6	GND	—
7	DCD	Data Carrier Detect	8	TRXC	Transmit Clock
9	RTXC	Receive Clock	10	GND	—

VMEbus INTERFACE

The VMEbus interface provides the signal path between the ICC and the VMEbus backplane. The interface complies with all requirements for the signal driver/receiver characteristics and bus operation protocols, as specified in the VMEbus Specification Rev. B.

The locations of the VMEbus signals at connectors P1 and P2 are shown in Table 1 and Table 3 respectively.

SERIAL INTERFACE

The eight serial ports conform to a subset of the RS-232-C standard. Signals are available on 10-pin connectors on the front panel and at the P2 connector at the rear of the module (see Tables 2 and 3). RTXC and TRXC are omitted on four of the ports at P2. The front panel con-

Table 3. Connector P2 Signal Locations

Pin No.	Row A Signals	Row B Signals	Row C Signals	Pin No.
1	GND	+5 V	DCD8	1
2	RTS8	GND	DTR8	2
3	TXD8	(RESERVED)	CTS8	3
4	DCD7	A24	RXD8	4
5	DTR7	A25	GND	5
6	CTS7	A26	RTS7	6
7	RXD7	A27	TXD7	7
8	GND	A28	DCD6	8
9	RTS6	A29	DTR6	9
10	TXD6	A30	CTS6	10
11	DCD5	A31	RXD6	11
12	DTR5	GND	GND	12
13	CTS5	+5 V	TRS5	13
14	RXD5	(D16)	TXD5	14
15	TRXC4	(D17)	RTXC4	15
16	GND	(D18)	DCD4	16
17	RTS4	(D19)	DTR4	17
18	TXD4	(D20)	CTS4	18
19	RTXC3	(D21)	RXD4	19
20	DCD3	(D22)	TRXC3	20
21	DTR3	(D23)	GND	21
22	CTS3	GND	RTS3	22
23	RXD3	(D24)	TXD3	23
24	TRXC2	(D25)	RTXC2	24
25	GND	(D26)	DCD2	25
26	RTS2	(D27)	DTR2	26
27	TXD2	(D28)	CTS2	27
28	RTXC1	(D29)	RXD2	28
29	DCD1	(D30)	TRXC1	29
30	DTR1	(D31)	GND	30
31	CTS1	GND	RTS1	31
32	RXD1	+5 V	TXD1	32

Note: Signals in parentheses are not used by the MVME332

nectors are 3M type 3654-5002JC. The mating connector type is 3473-6010. Retaining clips type 3591-8110 are available.

The I/O signals at P2 are provided to accommodate applications where the cable must be connected at the rear of the VME rack. If the application demands a type other than the 10-pin connectors on the MVME332 front panel, P2 can be used to connect via flat cable to a user panel board designed to the required specifications (e.g. 25-pin subminiature D).

ORDERING INFORMATION

Part Number	Description
MVME332	Eight-Channel Intelligent Communications Controller. Includes User's Manual.

MVME332 SPECIFICATIONS

Characteristics	Specifications
General	The MVME332 Intelligent Communication Controller is a VME-module for interfacing eight asynchronous serial communication channels to a VME system.
Performance	The MVME332 can simultaneously control eight channels at 9600 Baud half duplex with individual character translation and control code processing for each channel.
Local Microcomputer	The local microcomputer consists of a 10 MHz MC68010 microprocessor, 32Kb ROM, 128Kb RAM, interrupt handling logic, and various timers.
VMEbus Interface	The control and status registers on the ICC are accessed from the VMEbus through an option A16:D16 slave interface. The ICC accesses system memory through an option A32:D16 or A24:D16 master interface and supervises data transfers with a bus timer of variable length. The option RWD bus requester can be selected to operate on any one of the priority levels. The priority level and the status/ID byte of the interrupter are software programmable.
Peripheral Interface	Serial link control and data transfer are performed by four Z8530 Serial Communication Controller devices. Baud rates (50 to 76800), data characteristics (number of data bits and stop bits, even or odd parity) and control signal functions (DTR, DSR, RTS, CTS, DCD) are selectable independently for each of the eight channels.
Operating System Interface	The transfer of commands, status messages and data packets is performed through a pipeline in VME system memory. This transfer follows the Motorola standard Buffered Pipe Protocol for processor to processor communication. Drivers are available for VERSAdos and SYSTEM V/68.
ICC Commands	The ICC commands include: — ICC configuration and diagnostics — port configuration and diagnostics — data read and write operations — break transmission — software download and start — reset and abort functions
ICC Status Messages	The ICC status messages include: — normal command completion — ICC and port status — parity and framing errors — break detection — data overrun and buffer overflow — read and write time-out — invalid commands and parameters
Indicators	FAIL LED on the front panel
Front Panel Connectors	Eight 3M 3654-5002JC ten-pin connectors

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements	+5 Vdc (±5%): 2.6 A (typ) 3.2 A (max)
Operating Temperature	0° to 55°C
Storage Temperature	−40° to 100°C
Relative Humidity	0% to 90% (non-condensing)
Physical Dimensions — Height Depth	 9.2 in (233 mm) 6.3 in. (160 mm) Double high VME board with front panel

MVME333-2
MVME705

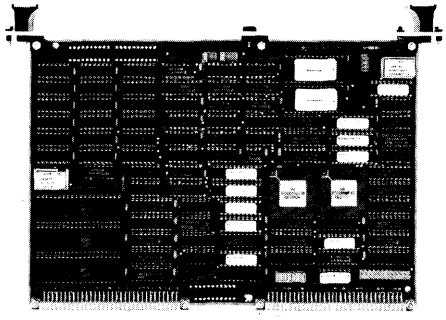
2

VMEmodule™ Six-Channel Synchronous/ Asynchronous Communications Controller with DMA

- Intelligent Communications Controller (ICC)
- Six RS-232-C or RS-422 Ports (Selection on separate I/O Transition Module)
- Data Transfer Modes:

Synchronous	1 Mbit/s, Single Channel
Asynchronous	78.6K Baud, Full Duplex, Single Channel
	9600 Baud, Full Duplex, 6-Channels, Concurrent
DMA	Half Duplex, Four Channels
	Full Duplex, Two Channels

- 512Kb Dynamic RAM (Zero CPU Wait Cycles)
- 10 MHz, MC68010 16-/32-Bit Microprocessor
- MC68450 DMA Controller
- Three Z8530 Serial Communications Controllers
- Double High Eurocard Form Factor
- VMEbus Compatible
- Debug Monitor Available



The MVME333-2 Intelligent Communications Controller offers the control and interfaces for six high performance, asynchronous or synchronous serial channels and DMA data transfer support for up to four of them. An MC68010 Microprocessor, 512Kb of dynamic RAM, an MC68450 DMA Controller and three Z8530 Serial Communications Controllers provide the hardware support which, in conjunction with user-provided firmware, comprises a powerful and versatile microcomputer for controlling serial communications in industrial and commercial applications. A block diagram of the controller is shown in Figure 1.

The controller is designed for use in VMEbus-based systems requiring high performance. With appropriate user-supplied firmware, it can handle all I/O related functions locally. Relieving a host of the overhead of performing peripheral control functions and of individually processing received and transmitted characters can improve system throughput and ease the design of flexible, modular applications.

Each serial port can operate individually at a synchronous data rate of up to one megabit per second or asynchronously at up to 78.6K baud, full duplex. All six ports can operate concurrently at a data rate of 9600 baud, full duplex.

Four ports can be configured for DMA data transfer offering four half duplex or two full duplex communications channels, or one full duplex channel and DMA to the VMEbus. Under DMA control, data can be transferred between system and local memory, between system memory and devices on the serial channels and between local memory and devices on the serial channels.

The MVME333-2 accesses the VMEbus through an A32:D16 or A24:D16 master interface, supported by interrupter logic and a release-when-done (RWD) bus requester. Control and status registers allow a host to send interrupts and commands to the controller and the controller to provide status information to a host. The memory map of Table 1 shows register and module device access locations.

VMEbus/MVME333-2 interconnection is through rows A & C of the rear P2 connector. Signals for all six channels are brought to P2 at TTL levels and require off-board translation of these levels to standard communication values (RS-232-C, RS-422-B or other). Changing to a different protocol thus requires only a new external level shifter — with no change to the MVME333-2 module or to software. The user may provide this via a bulkhead connector panel or by means of the available transition modules.

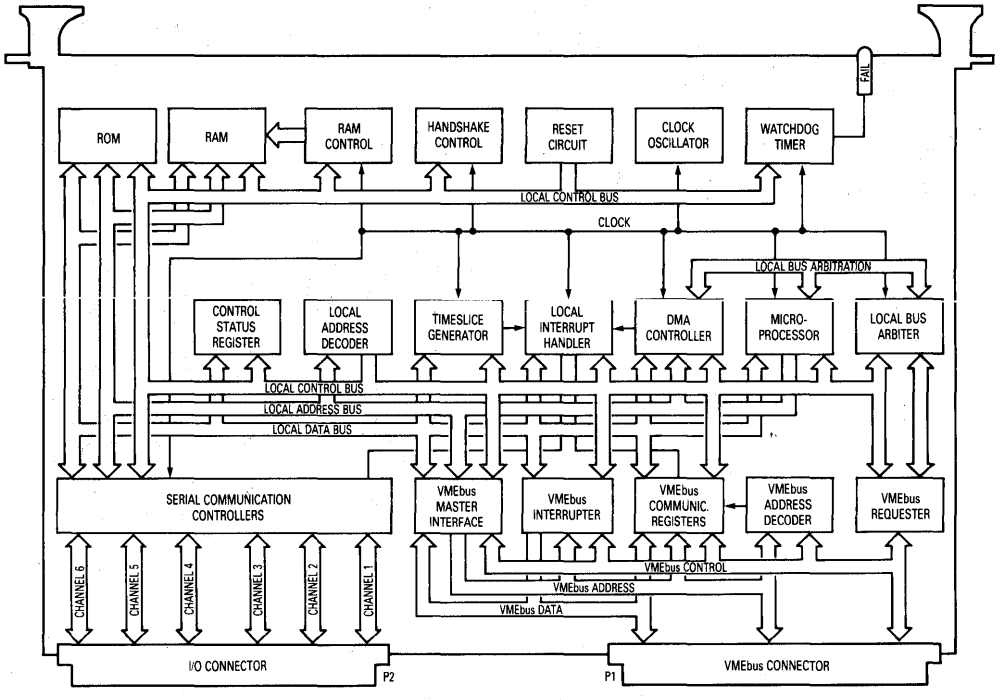
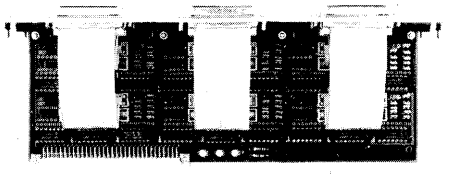


Figure 1. MVME333-2 Functional Block Diagram

MVME705 COMMUNICATIONS TRANSITION MODULE

By means of a separate MVME705 module, any of the six serial channels can be individually configured for RS-232-C or RS-422-B operation. The MVME705 module provides the receiver and transmitter circuits for converting the input/output signals of the controller to conform to the requirements of the desired standard. Figure 2 is a block diagram of the MVME705 module.

A transition module is connected to the MVME333-2's P2 connector via a 64-conductor flat ribbon cable having DIN41612 C64 connectors at each end. Six standard 25-pin subminiature "D" connectors are provided on a transition module's front panel for attaching serial peripheral devices. Each serial port can be configured independently as data circuit terminating equipment (DCE) or as data terminal equipment (DTE) by means of jumpers on the transition module. Tables 2 through 4 provide, for each configuration, the RS-232-C and RS-422-B pin/signal correspondences at the P2 and front panel connectors.



Configuring a channel on the MVME705 for RS-232-C or RS-422-B operation is done by installing in the IC sockets on the module the appropriate transmitter/receiver devices (supplied with the MVME705). The MVME705 package contains all devices required for either operating standard. In addition, the module includes locations in which the user can install capacitors for slew-rate limitation and noise filtering (for RS-232-C) or resistors for balanced line termination (for RS-422-B).

Table 1. MVME333-2 Local Address Map

ADDRESS	EVEN BYTES	ODD BYTES	ADDRESS
$FF8F00	Address Extension Register		$FF8F01
$FF8E00	Local Status Register and Local Control Register		$FF8E01
$FF8DFE		Address Modifier Register	$FF8D01
:	MPU	Communication Channel 5 (SCC3 Channel A)	$FF8C27 $FF8C25
:	MPU	Communication Channel 6 (SCC3 Channel B)	$FF8C23 $FF8C21
:	Not used	Communication Channel 3 (SCC2 Channel A)	$FF8C17 $FF8C15
:	DMAC	Communication Channel 4 (SCC2 Channel B)	$FF8C13 $FF8C11
:	DMAC	Communication Channel 1 (SCC1 Channel A)	$FF8C07 $FF8C05
:	DMAC	Communication Channel 2 (SCC1 Channel B)	$FF8C03 $FF8C01
:		Time Slice Counter	$FF8B01
:		VME Command Byte Register	$FF8A01
$FF8902		VME Interrupt Location	$FF8903
$FF8900	VME Interrupter Register		$FF8901

ADDRESS	EVEN BYTES	ODD BYTES	ADDRESS
$FF88FE : $FF88C0	DMAC Channel 3		$FF88FF : $FF88C1
$FF88BE : $FF8880	DMAC Channel 2		$FF88BF : $FF8881
$FF887E : $FF8840	DMAC Channel 1		$FF887F : $FF8841
$FF883E : $FF8800	DMAC Channel 0		$FF883F : $FF8801
$F00000	Watchdog Timer Reset Location		$F00001
$8FFFFE : $800000	VMEbus Window Border (BERR generated)		$8FFFFF : $800001
$7FFFFE : : : : : $400000	VMEbus Window		$7FFFFF : : : : : $400001
$3FFFFE : $300000	VMEbus Window Border (BERR generated)		$3FFFFF : $300001
$17FFFE : : : $100000	Local RAM (512Kb)		$17FFFF : : : $100001
$01FFFE : : : $000000	Local ROM (16, 32, 64 or 128Kb)		$01FFFF : : : $000001

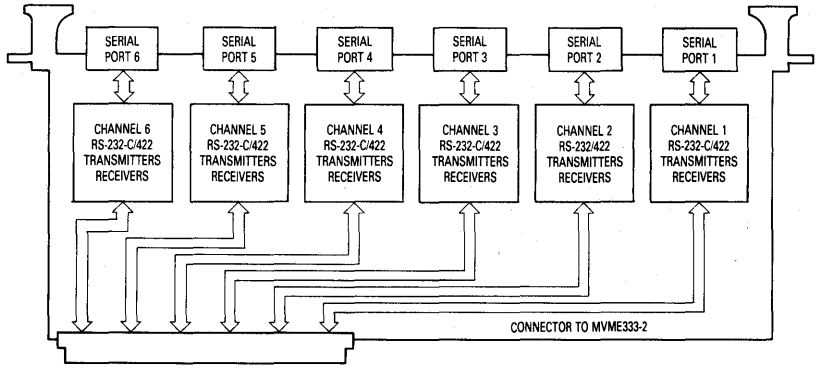


Figure 2. MVME705 Functional Block Diagram

Table 2. Serial Port in RS-232-C DTE Configuration

Pin	Signal	Direction	Connected With
2	TXD	output	P2 TXD
3	RXD	input	P2 RXD
4	RTS	output	P2 RTS*
5	CTS	input	P2 CTS*
6	DSR	input	P2 DSR*
7	GND	—	P2 GND
8	DCD	input	P2 DCD* or open
15	TXCI	input	P2 TRXC* or open
17	RXCI	input	P2 RTXC*
20	DTR	output	P2 DTR*
24	TXCO	output	P2 TRXC* or open

Table 3. Serial Port in RS-232-C DCE Configuration

Pin	Signal	Direction	Connected With
2	TXD	input	P2 RXD
3	RXD	output	P2 TXD
4	RTS	input	P2 CTS*
5	CTS	output	P2 RTS*
6	DSR	output	P2 DTR*
7	GND	—	P2 GND
8	DCD	output	+ 12 Volts
15	TXCI	output	P2 TRXC* or open
17	RXCI	output	P2 TRXC* or open
20	DTR	input	P2 DSR*
24	TXCO	input	P2 RTXC*

COMMUNICATIONS CONTROLLER MEMORY

The module contains one bank of 18 RAM devices providing a capacity of 512Kb by the use of 256Kb devices. Data is written into and read from RAM without MPU or DMAC wait states. The generation and checking of byte parity is performed automatically during RAM accesses and a detected parity error causes generation of an autovectored level 7 interrupt to the MPU. Wrong parity can be forced for test purposes by setting the WWP* bit in the module control register. Refresh of the dynamic RAM is automatic with local arbitration logic controlling the access/refresh cycles.

The two provided ROM sockets, one for odd and one for even addresses, can accept a maximum of 128 kilobytes of firmware. Means are included for jumper configuration to accommodate 2764, 27128, 27256 or 27152 type EPROM devices, having access times of 250, 350, or 450 nanoseconds.

USER INTERFACE

To aid the development of software/firmware for an application using the MVME333-2 module, an optional debug monitor, 333bug, is available which can be used in conjunction with a hardware/software or hosted development system or for the standalone programming and operating environment it can provide.

Table 4. Serial Port in RS-422-B Configuration

Pin	Signal	Pin	Signal	Direction	Connected With
2	−TXD	15	+TXD	output	P2 TXD
3	−RXD	16	+RXD	input	P2 RXD
4	−RTS	17	+RTS	output	P2 RTS*
5	−CTS	18	+CTS	input	P2 CTS*
6	−DSR	19	+DSR	input	P2 DSR*
7	GND			—	P2 GND
8	−DCD	21	+DCD	input	P2 DCD* or open
9	−DTR	22	+DTR	output	P2 DTR*
10	−TXCI	23	+TXCI	input	P2 TRXC* or open
11	−TXCO	24	+TXCO	output	P2 TRXC* or open
12	−RXCI	25	+RXCI	input	P2 RTXC*

333bug FIRMWARE MONITOR FACILITIES

The 333bug monitor is a command driven program with facilities for accepting and executing commands entered at the system console. The command set, listed in Table 5, includes commands for examining and modifying memory and MPU registers, commands which allow blocks of memory to be filled, moved or searched for the occurrence of specified data, breakpoint commands which allow program segments to be run, trace commands for examining the execution of instructions or small program portions, commands for beginning execution at a specified address, and commands for displaying all relative offsets. The memory display command includes, also, a disassemble capability while the memory modify command has both a disassemble and an assemble capability.

Other commands in the set facilitate host/MVME333-2 communications, the conversion of the number base of data and the dumping/loading of memory data in the standard Motorola S-Record format.

CALLABLE MVME333 DEBUG MONITOR FUNCTIONS

A TRAP #15 handler is included in 333bug which allows calls from a user program to input, output and other useful functions. Such a system call is made by including in the source program a TRAP #15 instruction line followed by a DC.W instruction containing the code number of the desired function. Use of these functions provides a means of quickly preparing a small test program to aid in system development. The callable 333bug functions are listed in Table 6.

Table 5. MVME333bug Commands

Command	Syntax
Display all address registers	.A
Display/set address register	.A0–.A7 [<addr>/<data>]
Display all data registers	.D
Display/set destination function code	.DFC
Display/set data register	.D0–.D7 [<addr>/<data>]
Display/set program counter	.PC [<addr>/<data>]
Display/set source function code	.SFC
Display/set status register	.SR [<addr>/<data>]
Display/set superv. stack pointer	.SS [<addr>/<data>]
Display/set user stack pointer	.US [<addr>/<data>]
Display/set relative offset	.R0–.R6 [<addr>/<data>]
Display/set vector base register	.VBR
Display formatted MPU registers	DF
Display all relative offsets	OF
Block of memory fill	BF <addr1> <addr2> <data>
Block of memory initialize	BI <addr1> <addr2>
Block of memory move	BM <addr1> <addr2> <addr3>
Block of memory search	BS <addr1> <addr2> <data>/<text>
Block of memory test	BT <addr1> <addr2>
Memory display/disassembly	MD[<port>] <addr>[<count>] [;<opt>]
Memory modify/disassembly/assembly	MM <addr>[<;opts>]
Memory set	MS <addr> <data>/<text> . . .
Breakpoint set	BR [<addr1>[<;count1>]] . . .
Breakpoint remove	NOBR [<addr1>] . . .
Go execute program	GO [<addr>]
Go execute program to breakpoint	GT <addr>
Go direct execute program	GD [<addr>]
Trace one instruction	TR [<count>]
Trace to temporary breakpoint	TT <addr>
Dump memory (S-records)	DU[<port>] <addr1> <addr2> [<text>]
Load memory (S-records)	LO[;[<opts>] [=<text>]]
Verify memory (S-records)	VE[;=<text>]
Data conversion	DC <addr>/<data>
Help DMAC addresses	HD
Help	HE
Help SCC addresses	HS
Transparent mode	TM [<char>]

Table 6. Trap #15 Callable Routines

Code	Description
$00	Convert binary data to ASCII string
$01	Convert ASCII string to binary data
$03	Change BREAK handler entry address
$10	Display MPU registers and return to 333bug
$11	Receive ASCII character from Serial Port 1
$12	Transmit ASCII character through Serial Port 1
$13	Receive ASCII string from Serial Port 1
$14	Transmit ASCII string through Serial Port 1
$15	Check Serial Port 1 for BREAK condition
$21	Receive ASCII character from Serial Port 2
$22	Transmit ASCII character through Serial Port 2
$23	Receive ASCII string from Serial Port 2
$24	Transmit ASCII string through Serial Port 2
$34	Transmit ASCII string through Parallel Port

VMEbus INTERFACE

For data interchange with a host on the MVMEbus, the controller utilizes an A32:D16 or A24:D16 master interface supported by address decoding logic, address modifier registers and an option Release When Done (RWD) bus requester. System memory accesses are controlled by the option RWD bus requester which can be set by jumper to any one of the four priority levels. Data transfers are supervised by a bus timer with a jumper selectable period. Two access modes are provided: the window addressing mode in which a 4Mb local address map is used as a window to the VMEbus and the alternate space addressing mode in which an address extension register is used to obtain mapping of a contiguous 16Mb address space anywhere within the full 4 gigabyte VMEbus range.

VMEbus INTERFACE (Cont.)

Interrupt requests from the controller to a system host are generated by a VMEbus interrupter having provision for a programmable status/ID byte. When the bus is acquired for MPU accesses, the requester releases on the next local data transfer. When the bus is acquired for the DMA controller, it is held by the requester until the DMA controller causes release of the bus. The controller status register contains a bit reflecting the state of the interrupt request output which can be polled by the controller MPU.

An option A16:D16 slave interface accommodates accesses from the VMEbus of the ICC control and status registers. These occupy, in the short I/O address space, a single word which appears redundantly throughout a 128 word segment. By jumper, the segment can be placed on any 256 byte boundary relative to the I/O space base address. A write operation accesses the VME control register. A read operation accesses the VME status register.

Short supervisory and non-privileged access of the short I/O space is provided by a bipolar PROM which decodes address modifier codes $29 and $2D. The PROM may be reprogrammed if use of other codes is desired.

Tables 7 and 8 identify, respectively, the connectors P1 and P2 pin/VMEbus signal assignments.

Table 7. Connector P1 VMEbus Signal Locations

Pin No.	Row A Signals	Row B Signals	Row C Signals	Pin No.
1	D00	BBSY*	D08	1
2	D01	(BCLR*)	D09	2
3	D02	(ACFAIL*)	D10	3
4	D03	BG0IN*	D11	4
5	D04	BG0OUT*	D12	5
6	D05	BG1IN*	D13	6
7	D06	BG1OUT*	D14	7
8	D07	BG2IN*	D15	8
9	GND	BG2OUT*	GND	9
10	(SYSCLK)	BG3IN*	SYSFAIL*	10
11	GND	BG3OUT*	BERR*	11
12	DS1*	BR0*	SYSRESET*	12
13	DS0*	BR1*	LWORD*	13
14	WRITE*	BR2*	AM5	14
15	GND	BR3*	A23	15
16	DTACK*	AM0	A22	16
17	GND	AM1	A21	17
18	AS*	AM2	A20	18
19	GND	AM3	A19	19
20	IACK*	GND	A18	20
21	IACKIN*	(SERCLK)	A17	21
22	IACKOUT*	(SERDAT)	A16	22
23	AM4	GND	A15	23
24	A07	IRQ7*	A14	24
25	A06	IRQ6*	A13	25
26	A05	IRQ5*	A12	26
27	A04	IRQ4*	A11	27
28	A03	IRQ3*	A10	28
29	A02	IRQ2*	A09	29
30	A01	IRQ1*	A08	30
31	−12 V	(5 V STBY)	+12 V	31
32	+5 V	+5 V	+5 V	32

Note: Signals in parentheses are not used by the MVME333-2.

Table 8. Connector P2 Signal Locations

Pin No.	Row A Signals	Row B Signals	Row C Signals	Pin No.
1	+5 V	+5 V	+5 V	1
2	−12 V	GND	+12 V	2
3	TRXC6*	(RESERVED)	GND	3
4	DTR6*	A24	DCD6*	4
5	CTS6*	A25	DSR6*	5
6	RXD6	A26	RTS6*	6
7	TXD6	A27	RTXC6*	7
8	TRXC5*	A28	GND	8
9	DTR5*	A29	DCD5*	9
10	CTS5*	A30	DSR5*	10
11	RXD5	A31	RTS5*	11
12	TXD5	GND	RTXC5*	12
13	TRXC4*	+5 V	GND	13
14	DTR4*	(D16)	DCD4*	14
15	CTS4*	(D17)	DSR4*	15
16	RXD4	(D18)	RTS4*	16
17	TXD4	(D19)	RTXC4*	17
18	TRXC3*	(D20)	GND	18
19	DTR3*	(D21)	DCD3*	19
20	CTS3*	(D22)	DSR3*	20
21	RXD3	(D23)	RTS3*	21
22	TXD3	GND	RTXC3*	22
23	TRXC2*	(D24)	GND	23
24	DTR2*	(D25)	DCD2*	24
25	CTS2*	(D26)	DSR2*	25
26	RXD2	(D27)	RTS2*	26
27	TXD2	(D28)	RTXC2*	27
28	TRXC1*	(D29)	GND	28
29	DTR1*	(D30)	DCD1*	29
30	CTS1*	(D31)	DSR1*	30
31	RXD1	GND	RTS1*	31
32	TXD1	+5 V	RTXC1*	32

Note: Signals in parentheses are not used by the MVME333-2. I/O signals are at TTL levels.

PERIPHERAL INTERFACE

Asynchronous and synchronous transfer of data through the serial channels is effected using three Z8530 Serial Communication Controller (SCC) devices. Each offers two, independent, full duplex serial channels, is capable of transferring data at rates up to 1Mbit per second over a single channel and provides the hardware support for the software control of communication modes and protocols. Signals at the TTL level are provided at the VMEbus P2 connector and, by means of a separate communications transition module which also provides the serial port connectors, are translated to the levels and connections required by the RS-232-C or RS-422-B communications standards. The communications signal assignments to pins in connector P2 are identified in Table 8.

The SCC devices are clocked at 4.9152 MHz. By division of this frequency, all baud rates for serial data transfer are generated.

Control provided by the SCC devices allows the independent selection of an encoding mode and the programming of parameters for synchronous or asynchronous operation of any serial channel. User-provided firmware can obtain, by virtue of the MC68450 Direct Memory Access Controller, four half duplex or

two full duplex, DMA-supported serial channels. For half or full duplex operation under control of the DMA device, the only restrictions on the programming of SCC parameters by user-provided firmware/software is that W/REQA*, W/REQB* and DTR/REQA* outputs used be programmed as DMA REQ outputs.

The other serial channels on the board are under direct control of the module MPU. The user can supply firmware/software to operate these channels as an application requires.

ASYNCHRONOUS OPERATION

When asynchronous data transfer is required, the following parameters can be programmed:

- — baud rate from 50 to 76800
- — odd, even or no parity
- — 1, 1.5 or 2 stop bits
- — character length of from 5 to 8 bits
- — NRZ, NRZI or FM data encoding

SYNCHRONOUS OPERATION

Transfer of both bit and byte data is supported and the SCC devices can be programmed for a number of different synchronization modes.

Both monosync and bisync modes with or without CRC checking can be used for byte data transfer.

The SDLC or the HDLC protocol with automatic CRC generation and checking can be used for the transfer of bit data. An underrun causes generation of an interrupt to the MPU.

For the synchronous transfer of byte or bit data, odd, even or no parity may be programmed and NRZ, NRZI or FM coding may be selected.

DATA TRANSFER UNDER MC68450 CONTROL

For transferring data beween VMEsystem memory, communications controller memory and SCC devices, the module uses an MC68450 Direct Memory Access Controller (DMAC). This high speed device is clocked at 10 MHz and offers four independent channels. Each channel contains a complete set of registers which can be independently configured for the desired type of operation.

With appropriate user-supplied firmware installed on the MVME333-2, data can be transferred between the following system and module resources:

- — between VMEsystem memory and module memory
- — between VMEsystem memory and the two SCC devices interfacing the DMAC
- — between module memory and the two SCC devices interfacing the DMAC
- — or various combinations of the above

Data transfer modes supported by module hardware which can be implemented by user-supplied firmware include:

- — implicit, single address, for transfer between a SCC device and VMEsystem memory or module memory
- — explicit, dual address, for transfer between VMEsystem and module memory
- — external request, for transfer from a SCC device to VMEsystem memory or module memory
- — limited rate auto request, for transfer between VMEsystem memory and module memory

COMMUNICATIONS CONTROLLER TIMER/COUNTERS

The MVME333-2 Communications Controller contains three timer/counters used for periodic interrupt generation, for the monitoring of module malfunctions and for supervising accesses of the module's data transfer bus and accesses of the VMEbus.

TIME-SLICE COUNTER

To facilitate the generation of firmware which can take advantage of the powerful serial data transfer capabilities offered by the controller, a programmable timer is included which can be used for the generation of periodic interrupts to the module MPU as a basis for a time-slice driven resident executive.

A value is loaded into an 8-bit register, counted down to zero and the resulting output used to generate an autovectored level 2 interrupt request. The interrupt request is latched then reset when acknowledged by the MPU.

Clocked at 78.125 kHz, the counter offers time periods ranging between 3.25 and 12.8 microseconds. The counter can also be disabled and restarted by program instructions.

WATCHDOG TIMER

To provide user software a means of indicating fatal MVME333-2 malfunctions, a timer with a period of 100 milliseconds is provided on the controller module. On timing out, this timer sets the fail bit in the status register, lights the fail LED on the module front panel and may be jumpered to cause assertion of the VMEbus SYSFAIL* line. User software prevents timeout of the counter by simply accessing the watchdog timer reset location at $F00000.

DATA TRANSFER BUS TIMER

For supervision of all MPU and DMAC data transfer cycles on both the local bus and the VMEbus, the MVME333-2 module includes a timer which generates a local bus error signal if an accessed device fails to respond with a data transfer acknowledge signal before the counter times out. The timeout period is fixed at 25 microseconds for all accesses to local devices. For VMEbus accesses, a timeout period of 25, 50 or 100 microseconds is jumper selectable. The selected period should exceed the greatest time expected to elapse between the assertion of a bus request and the reception of a bus grant in the actual system configuration.

MVME333-2 SPECIFICATIONS

Characteristics	Specifications
General	The MVME333-2 Asynchronous/Synchronous Communications Controller is an intelligent VMEmodule for interfacing six serial communication channels with a VMEsystem.
Performance	The MVME333-2 hardware can simultaneously control six full duplex channels. The maximum data transfer rate is 76800 baud in asynchronous and 1Mb/s in synchronous modes.
Local Microcomputer	The local microcomputer consists of a 10 MHz MC68010 MPU, a 10 MHz MC68450 DMAC, sockets for up to 128Kb ROM, 512Kb RAM with parity, interrupt handling, address decoding, local bus arbitration and reset logic and various timers.
DMA Control	Hardware is provided for four channels of DMA control of data transfers between local and system memory via the VMEbus, or between memory and the serial communication devices. Either four half duplex or two full duplex communication channels can be configured for DMA control.
VMEbus Interface	A VMEsystem host accesses control and status registers on the controller through an option A16:D16 slave interface. The controller accesses VMEbus global system memory through an option A32:D16 or A24:D16 master interface and supervises data transfers with a bus timer of variable length. The option RWD bus requester can be strapped to operate on any one of the four priority levels. The priority level and the status/ID byte of the interrupter are software programmable.
Peripheral Interface	Serial link control and data transfer are performed by three Z8530 Serial Communication Controller devices. Asynchronous data transfer is supported, with baud rates (50 to 76800), data characteristics (number of data bits and stop bits, even or odd parity) and control signal functions (DTR, DSR, RTS, CTS, DCD) being independently selectable for each of the six channels. Byte or bit orientated synchronous communication with monosync, bisync or SDLC/HDLC formats are also supported. Coding may be NRZ, NRZI or FM. The peripheral signals appear as TTL levels at the lower rear connector P2 and can be configured for either the RS-232-C or the RS-422-B standard on a separate communications transition module (MVME705).
Indicators	FAIL LED on the front panel.

MVME333-2 MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements	+5 Vdc (±5%), 3.8 A typ, 4.4 A max
Operating Temperature	0° to 55°C Inlet air, forced air cooling
Storage Temperature	−40° to 100°C
Relative Humidity	0% to 90% (non-condensing)
Physical Characteristics	
Form Factor	Double High VME board with front panel
Board Dimensions	
Height	9.17 in. (233 mm)
Depth	6.3 in. (160 mm)
Front Panel Dimensions	
Height	10.3 in. (262 mm)
Width	0.79 in. (20 mm)

MVME705 SPECIFICATIONS

Characteristics	Specifications
General	The MVME705 module provides transmitter and receiver circuits for the MVME333-2 Intelligent Communications Controller module. Both modules are interconnected with a 64-conductor flat-ribbon cable. The serial port signals are available at 25-pin connectors on the front panel.
Serial Port Characteristics	Each channel can be configured for the RS-232-C or RS-422-B EIA standard. The front panel connectors can be configured as data circuit terminating equipment or as data terminal equipment.
Supported Signals	The MVME705 supports the following signals: TXD — Transmitted Data RXD — Received Data RTS — Request To Send CTS — Clear To Send DSR — Data Set Ready DTR — Data Terminal Ready DCD — Data Carrier Detect TRXC — Transmitter-Receiver Clock RTXC — Receiver-Transmitter Clock
Additional User Options	The user can install capacitors for slew-rate limitation and noise filtering (for RS-232-C) and resistors for balanced line termination (for RS-422) on free locations on the board.

MVME705 MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements	+5 Vdc, 1.5 A (max) ±12 Vdc, 150 mA (max)
Operating Temperature	0°C to 55°C Inlet air, forced air cooling
Storage Temperature	−40° to 100°C
Relative Humidity	0% to 90% (non-condensing)
Physical Characteristics Form Factor Board Dimensions Height Depth Front Panel Dimensions Height Width	 Double-high/reduced depth VMEboard 9.2 in. (233 mm) 3.5 in. (80 mm) 10.3 in. (262 mm) 1.57 in. (40 mm)
Connectors	One 64-pin DIN41612 connector to the MVME333-2. Six 25-pin Sub-D serial port connectors.

ORDERING INFORMATION

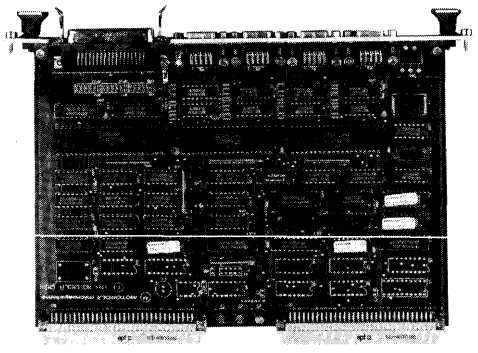
Part Number	Description
MVME333-2	Six Channel Intelligent Serial I/O Controller with DMA. 512Kb RAM. Includes User's Manual.
MVME705	I/O Transistion board with RS-232 or RS-422 interface. Includes User's Manual.
M68V1XS333BUG	Debug Monitor Package. Set of EPROMs. Source and object code on 5-1/4" VERSAdos diskette. Includes User's Manual.

RELATED DOCUMENTATION

Part Number	Description
HB212/D	VMEbus Specification Manual
MC6845	4-Channel DMA Controller Advance Information
Zilog	Z8030/Z8530 Serial Communication Controller Technical Manual.
MVME333E/D	MVME333 Intelligent Communication Controller User's Manual.
MVME333bug/D	MVME333bug Debug Package User's Manual.

Product Preview

VMEmodule Serial and Parallel I/O Module



The MVME335 is a low cost non-intelligent serial/parallel input/output on a double EUROcard form factor printed circuit board. It contains four RS-232-C compatible asynchronous serial I/O ports and one Centronics compatible parallel printer port. There is also a 24-bit timer which can generate periodic interrupts or a single interrupt after an elapsed time period. The bus interface is compatible to the VMEbus Specification Revision C.

Features

- Four serial asynchronous RS-232-C ports using two MC68681 DUART devices
- One Centronics compatible parallel printer port using the MC68230 P1/T
- Three timers contained in the MC68230 P1/T and the MC68681 DUART devices
- Separate baud rate generator for each serial I/O port
- I/O lines routed to front panel connectors and to connector P2
- Three VMEbus interrupters; request level jumperable
- Separate interrupt vector for each I/O port and timer
- VMEbus slave interface

MC68681 DUAL ASYNCHRONOUS SERIAL I/O CONTROLLER

The MC68681 is a dual universal asynchronous receiver transmitter (DUART) which interfaces directly to the MC68000 processor via an asynchronous bus. All serial I/O signals are buffered with RS-232-C drivers and receivers. The following features of the MC68681 are supported on the MVME335:

- Two independent full-duplex asynchronous receiver/transmitter channels
- Quadruple-buffered receiver data register
- Double-buffered transmitter data register
- Programmable baud rate for each transmitter and receiver
- Programmable data format
- Programmable channel modes
- Automatic wake-up mode for multidrop applications
- Multifunction 16-bit programmable timer/counter
- Versatile interrupt capability
- Parity, framing, and overrun error detection
- False start bit detection
- Line break detection and generation
- Start/end break interrupt/status

PARALLEL I/O CIRCUIT MC68230

The MC68230 is a parallel interface circuit including an internal 24-bit timer. It provides all the necessary signals required for the parallel printer port. Additional hardware generates the strobe and acknowledge handshake. All printer I/O signals are buffered with TTL drivers and receivers.

VMEbus SLAVE INTERFACE

The VMEbus slave interface contains the address decoder and data buffers to select the board and to pass data from and to the I/O controller. The board is used in the VMEbus short I/O address space with address modifier codes $29 and $2D. The base address can be selected in 256 byte steps. According to VMEbus specification, Rev. C, the following slave module type is supported:

A16, D08(O), NBLT, RMW, NUAT

VMEbus INTERRUPTER

There are three interrupters on the board. Each of them can assert interrupt requests on the VMEbus independently. The interrupt request source of the first interrupter is the 24-bit timer contained in the MC68230. The second interrupter has two interrupt request sources. These are serial I/O circuit 1 and serial I/O circuit 2. The parallel printer interface is the interrupt request source of the third interrupter.

Each interrupter operates on an independently selectable priority level and each interrupt source has its own interrupt vector. According to the VMEbus specification, Rev. C, the following interrupter type is supported:

D08(O) I(1-7) RORA

The interrupt sources are prioritized in a local daisy-chain which gives the timer highest priority, then the serial ports, followed by the printer port.

RS-232-C SERIAL I/O PORTS

For the four RS-232-C ports there are connectors provided on the front panel. The connector type is a 9-pin, female, PCB right angle, subminiature-D connector. The RS-232-C I/O lines are also routed to rows a and c of connector P2.

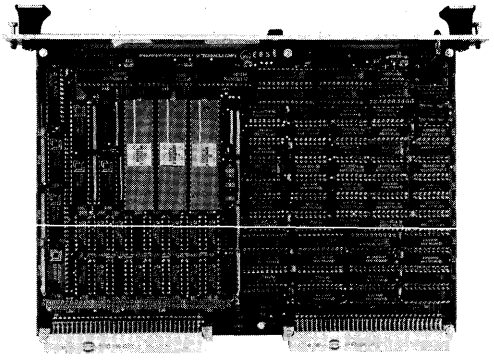
CENTRONICS PARALLEL PRINTER PORT

On the front panel there is also a connector for a parallel printer port provided. The connector type is a 36-pin, female, Centronics compatible connector. The parallel printer port lines are also routed to rows a and c of connector P2. The serial I/O lines routed to connector P2 have RS-232-C voltage levels. The ports are configured as modem to allow easy connection of a terminal.

SPECIFICATIONS

Characteristic	Specification
Temperature Operating Storage	 0°C to +55°C −40°C to +85°C
Relative Humidity	0% to 90% (non-condensing)
Board Dimensions	Double Eurocard format (160 mm × 233.6 mm)
Connector P1 Connector P2	96 Pin, DIN 41612 64 Pin, DIN 41612
Front Panel	4 TE, 6 HE
Front Panel Connectors	Four 9 pin, female, subminiature-D connectors One 36 pin, female, PCB right angle, CENTRONICS connector.

VMEmodule™ Parallel Interface/Timer Module



- 64 Parallel I/O Lines
- Three 24-Bit Timers
- Three Interrupters; Interrupt-level and Interrupt Vector for each Interrupter are Software-Programmable
- One 32-Bit and one 16-Bit Data Channel or any Combination of 16-Bit and 8-Bit Data Channels
- Optional 4 Gigabyte, 16Mb or 64Kb Address Range
- Programmable Status LED on Front Panel
- VMEbus Compatible

The MVME340 Parallel Interface/Timer Module permits the user to control peripheral hardware devices such as valves, displays, etc. The interface module supports up to 64 parallel I/O lines which can be programmed for input, output, handshake and timer functions. All these functions for input are performed by the parallel I/O circuit MC68230. Each line can be individually programmed for input or output by means of data direction registers. Bidirectional operation controlled by handshake lines is also available.

There are three 24-bit timers provided which can be programmed independently. The clock for the timer can either be the system clock and/or an external clock applied to the counters via connector P2. Figure 1 is a functional block diagram of the MVME340.

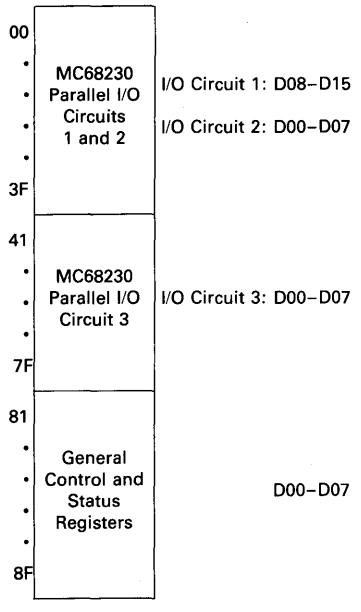
PARALLEL I/O CIRCUIT MC68230

The MC68230 is a parallel interface circuit including an internal 24-bit timer. It provides all the necessary signals asserted to connector P2. The MC68230 parallel interface can operate in unidirectional or bidirectional modes. In the unidirectional mode, the data direction register determines the direction of the input/output lines. In the bidirectional mode the transfer direction is controlled by handshake lines. It contains basically 16 input/output lines. A third I/O port is provided. The meaning of specific I/O lines of this port depends on the programmed mode of the MC68230. The lines can be used as simple I/O lines, as handshake and timer control lines, or as Direct Memory Access (DMA) support lines.

The timer consists of a 24-bit counter and a 5-bit prescaler. The clock source can be either the system clock or an external clock. It can generate periodic interrupts, a square wave, or a single interrupt after an elapsed time period.

REGISTER ADDRESS MAP

The MVME340 module occupies a 128-byte address space in the VME standard or short addressing range. The following figure shows the register map.



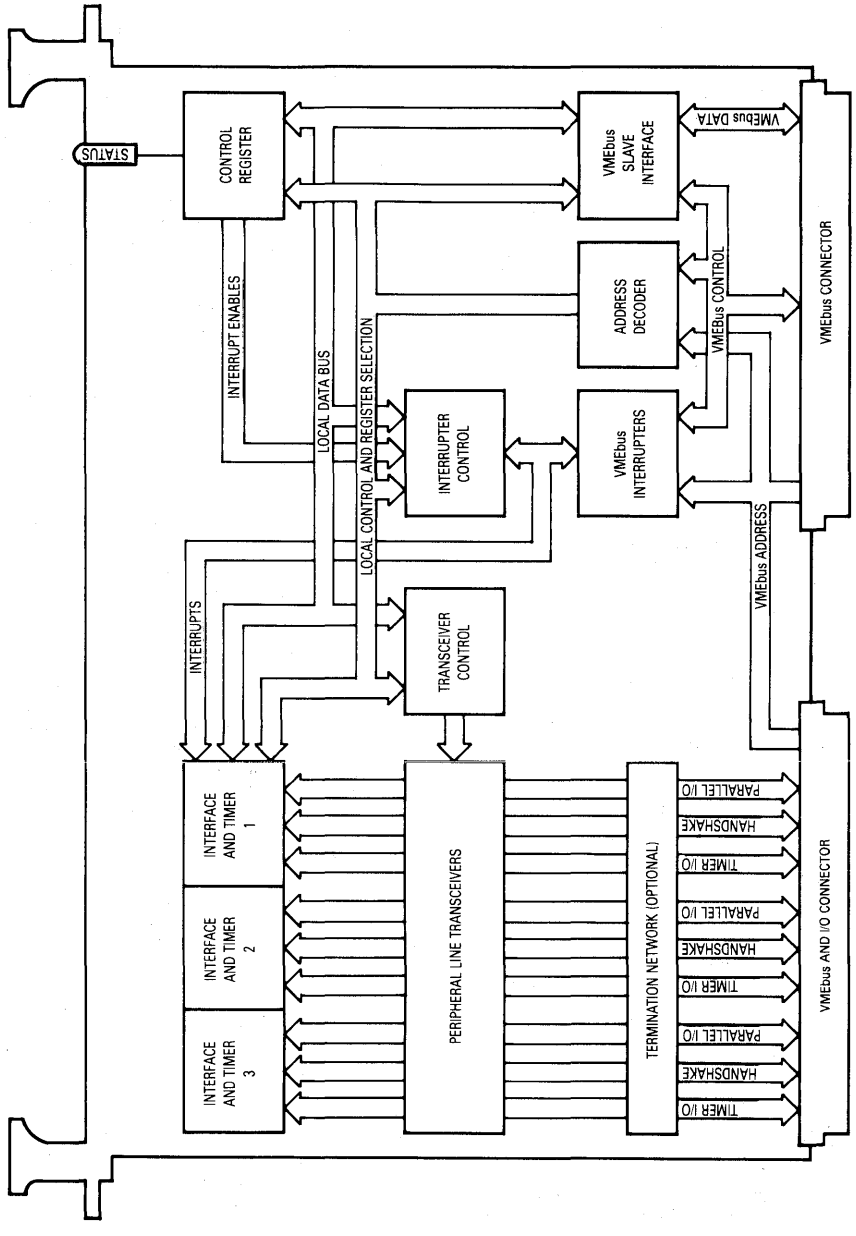


Figure 1. MVME340 Functional Block Diagram

CONTROL AND STATUS REGISTERS

In addition to the MC68230 internal control and status registers, external registers are provided which control data line terminations, enabling of the I/O buffers, indicator functions, interrupt select, etc.

ADDRESS DECODING LOGIC

The address decoding logic supports the 64Kb short addressing, the 16Mb standard addressing, and the 4 gigabyte extended addressing range. The addressing mode is determined by adjusting the jumper configuration. The module's base address is jumper-selectable in 256-byte steps.

ADDRESS MODIFIER

The address modifiers (AMs) are decoded by a PROM which allows them to be easily customized to a specific requirement. The module will respond in the standard configuration to address modifier codes $29, $2D, $39, $3D, $09 and $0D. These AM codes are defined as follows:

$29	Short Non-Privileged I/O Access
$2D	Short Supervisory I/O Access
$39	Standard Non-Privileged Data Access
$3D	Standard Supervisory Data Access
$09	Extended Non-Privileged Data Access
$0D	Extended Supervisory Data Access

When AM codes $29 or $2D are asserted, only address lines A01 to A15 are used to decode the module's base address. Otherwise, address lines A01 to A23 or A01 to A31 are decoded. To prevent address conflicts, one of these modes (short, standard, or extended addressing) can be selected by altering jumpers.

INTERRUPT LOGIC

Three interrupters are provided on the MVME340. The interrupt request levels are selected by programming the interrupt control registers. For each of the three interrupters two interrupt sources are available. This can be an interrupt generated by the appropriate handshake line or one which is generated when the timer count reaches zero. For each source of every level, a programmable 8-bit interrupt vector is applied to data lines D00 to D07 if the specific interrupt level is granted by the master.

PARALLEL I/O INTERFACE

The parallel I/O interface consists of 64 data lines routed via connector P2 on rows A and C. The I/O lines are divided into the following functional groups:

48 Input/Output lines
8 Handshake lines
8 Dual function lines.

The dual function lines can be used either as standard I/O lines or as timer I/O lines. The specific function can be selected by software programming and by jumper setting.

Each input/output data line can be programmed individually for input or output. This is achieved by programming data direction registers.

The drivers for the input/output lines meet the electrical specifications defined in the IEEE-488 1975 document. These specifications include voltage levels with corresponding logic states, driver and receiver requirements, a description of the resistor termination network, and negative voltage clamping circuitry.

Each I/O line can be terminated by a resistor network to V_{CC} and Ground. This ensures that the signal line voltage level rises to a high level whenever it is not driven low. Termination networks should be provided on each end of the cable between the parallel I/O module and the level adapter boards. There are spare locations provided which allow the user to install the termination networks on the module. The user can use either external termination resistors or the internal termination resistors built into the driver circuits. There are pull-up control bits provided in a register which allow the selection of open-collector or three-state driver configuration.

The interface to external components can be done by means of level adapter boards (customer-designed). These level adapter boards convert the TTL voltage levels to appropriate voltage or current levels depending on the application.

PROGRAMMING CONSIDERATIONS

The MVME340 Parallel Interface/Timer Module can read or write the register contents using 16-bit data transfer or 8-bit data transfer commands. The 16-bit data transfer operation is possible when accessing I/O circuits 1 and 2. I/O circuit 1 can be accessed via data lines D08 to D15, I/O circuit 2 via D00 to D07, and I/O circuit 3 via D00 to D07. All control and status registers can be accessed via D00 to D07.

This accessing scheme is a very flexible way to read or write data to/from an I/O interface. Nearly any speed requirements for accessing data can be met with this scheme (e.g., 32-bit I/O is carried out by means of the M68000 'Move Long Word' instruction, accessing circuit 1 and circuit 2).

Two modes are enabled to react to external or internal events generated by the MVME340. These are the interrupt and the polling modes. In the interrupt mode, events are signalled to the VME system by means of the interrupt asserting protocol defined in the VMEbus Specification (MVMEBS/D). In the polling mode appropriate status bits have to be monitored to detect the occurrence of an event. External events are generated by the handshake lines while internal events are issued from the timers. Selecting the appropriate mode is achieved by programming the interrupt control register.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristic	Specification
Bus Interface	VMEbus (P1)
Address Lines	A01–A31
Data Lines	D00–D15
Address Modifier Lines	AM0–AM5
Control Lines	DS0*, DS1*, AS*, DTACK*, IRQ1*, IRQ7*, IACK*, IACKIN*, IACKOUT*, SYSRESET*, LWORD*, SYSCLK, WRITE*
Peripheral Interface	3 x MC68230
Data Format	Byte or 16-bit Word
Status Indicator	One LED on front panel
Connector Type	IEC603-2 (DIN41612), P1 96-pin, P2 96-pin
Dimensions — PC Board — Front Panel	 160mm x 234mm (6.3" x 9.2") 262mm x 20mm (10.3" x 0.79") 4TE
Operating Temperature	0°C to 50°C
Storage Temperature	−25°C to 85°C
Relative Humidity	0 to 95% (non-condensing)
Shock	80g for 11 ms (3 levels)
Vibration	2g 10 – 100 – 10 Hz (3 levels)
Power Consumption	+5 Vdc, typ 3.2 A

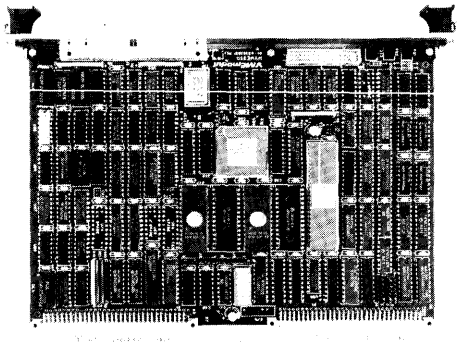
ORDERING INFORMATION

Part Number	Description
MVME340	VMEmodule Parallel Interface/Timer Module
MVME340/D	VMEmodule Parallel Interface/Timer Module User's Manual

Advance Information

VMEmodule™ Streaming Tape Controller

- Double High VMEmodule
- QIC-02 Streaming Tape Interface
- Supports One QIC-02 compatible 1/4-inch Streaming Tape Drive
- Standard VMEbus Interface
- Supports 24- or 32-bit DMA Addressing/16-bit Data
- Generates Seven Levels of VMEbus Interrupts with Programmable Interrupt Vector
- 10 MHz MC68010 Microprocessor
- 90Kb/s Continuous Transfer Rate for QIC-02 Interface, 200Kb/s Burst rate
- Controls Tape Cartridges Offering 20Mb, 45Mb and 60Mb of Formatted Data Storage
- MC68230 PI/T-based Timer
- 16Kb of Static RAM Provides Buffer Storage and CPU Workspace
- Multitasking Kernel-based Firmware Package
- Buffered Pipe Communication Protocol Allows Multiple Hosts to Queue Commands Without Interlock
- High Level Command/Status Packets offer efficient Operating System Support
- Permits Chaining of Host Commands



VMEbus applications that generate large quantities of data which must be archived and systems of limited data space can profitably use the economical storage capacity offered by streaming tape. Drives are presently available which utilize tape cartridges of up to 60 megabytes capacity.

GENERAL DESCRIPTION

A VMEmodule Intelligent Peripheral Controller (IPC), MVME350 provides control of one QIC-02 compatible, 1/4 inch streaming tape drive. Figure 1 is a functional block diagram of the module.

The 16Kb of static RAM onboard MVME350 provides working storage for the MC68010 vector table and stack frames. It also serves as buffer storage between the QIC-02 interface and the VMEbus. Optimum performance for typical applications is offered since the MC68010, at 10 MHz, performs zero wait state accesses of the SRAM space.

Two 28-pin EPROM sockets are provided which support devices up to 64K x 8 with access times of 200 nanoseconds. The MVME350 is shipped with firmware installed in these sockets.

The VMEbus interface is designed to provide uniformity and compatibility among MVMEmodule drivers running under the SYSTEM V/68 and VERSAdos Real-Time Operating Systems as well as ease the development, by a system integrator, of software to communicate with the controller module. Controller firmware based on a multitasking kernel handles all input/output interrupts and data transfers in order to minimize impact on system throughput. Macro commands are retrieved from a pipeline structure in system memory which is shared by a system host, such as an MVME121 or MVME131 microcomputer, and the tape controller. Status information is returned to a host through another pipeline. Exchange of command and status packets follows the Motorola buffered pipe protocol for interprocessor communication. Figures 3 and 4 illustrate the packet and packet envelope formats used.

OPERATING SYSTEM SUPPORT

Support for the MVME350 will be provided for both the VERSAdos and SYSTEM V/68 operating systems. A driver and utilities for the MVME350 will be included in Release 2.0, Version 3.x of SYSTEM V/68 for use in systems running this UNIX-derived operating system. General support will be provided for both file system and disk image operations.

The following SYSTEM V/68 backup and restore utilities will be supported: finc (fast incremental backup), frec (recover file systems from a backup tape), volcopy (copy file

system), cpio (copy file archives in and out) and tar (tape file archiver). In addition, the system call IOCTL will be supported to provide an application program with the means of obtaining complete control over a streaming tape device. Available functions will include:

- Rewind Tape
- Erase Tape
- Retension Tape
- Write Filemark
- Read Filemark
- Set DMA Buffer Size
- Get DMA Buffer Size

QIC-02 INTERFACE

The ANSI QIC-02 Interface Standard describes an intelligent interface for streaming tape drives which relieves a host of the overhead functions of tape formatting, error processing and tape positioning. On the MVME350 controller module, this interface is implemented using an MC68230 Programmable Interface/Timer device and firmware executing on the module's own MC68010 microprocessor.

A QIC-02 interface uses an eight-bit wide bidirectional bus for moving both data and commands/status. Eight unidirectional lines are used for control: four from host to tape drive and four from tape drive to host. These lines are used to report exceptions from the drive, reset the drive and perform handshaking. Two pairs of handshaking lines are used: one pair for the movement of data

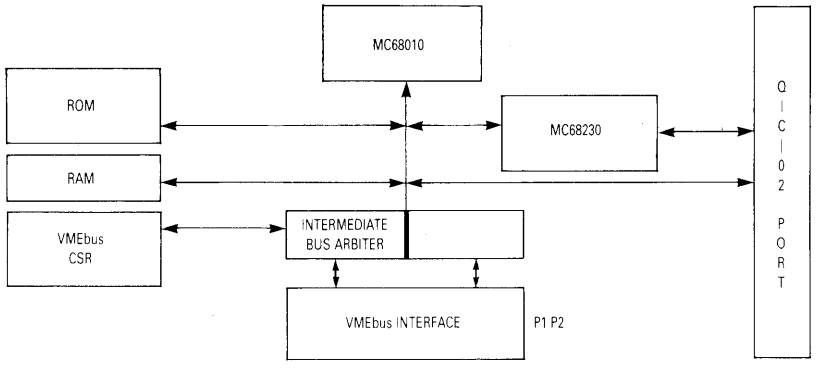


Figure 1. MVME350 Block Diagram

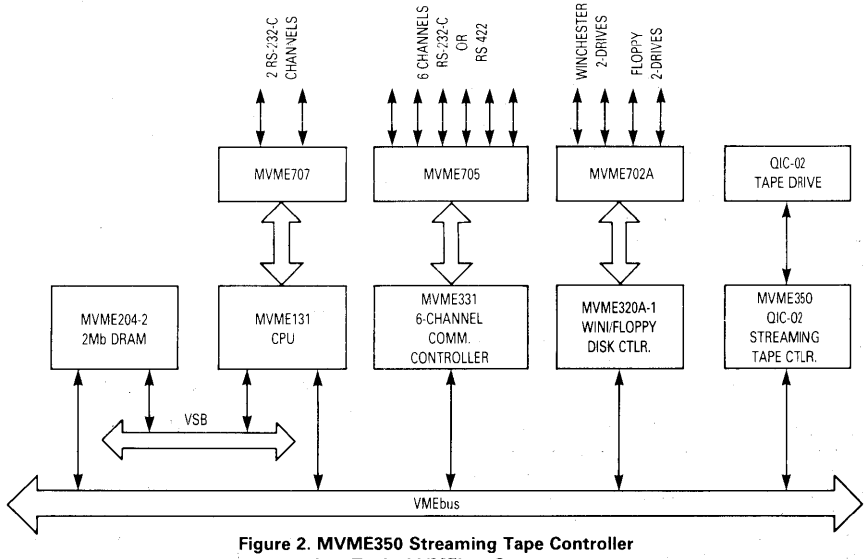


Figure 2. MVME350 Streaming Tape Controller in a Typical VMEbus System

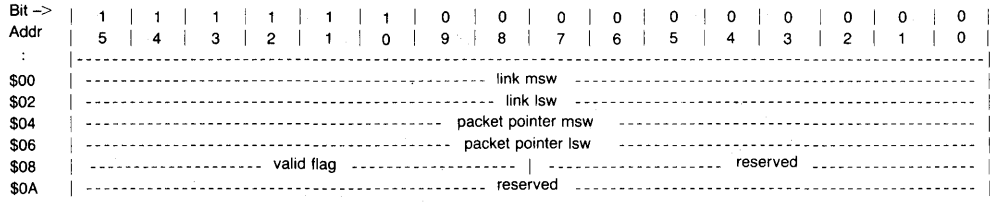


Figure 3. Channel Packet Envelope Format

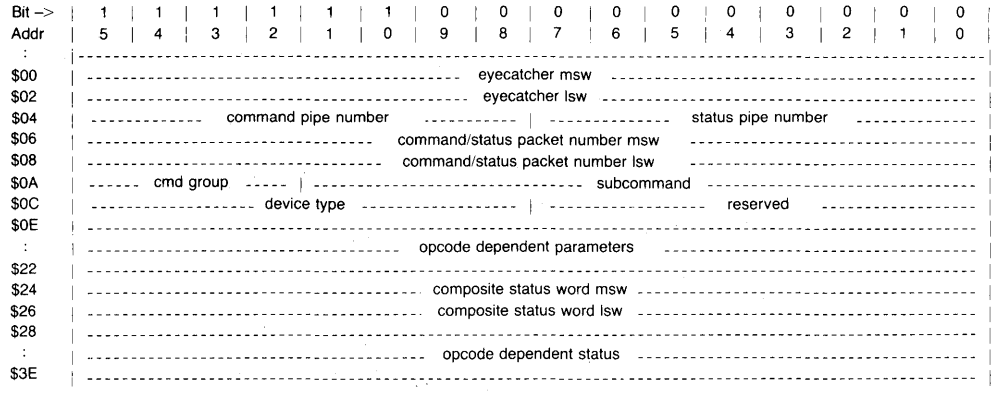


Figure 4. Channel Packet Format

and the other pair for commands and status. To eliminate tight timing constraints, the handshakes for both pairs are asynchronous.

Three sets of command types are described by the QIC-02 specification: a set of standard commands, commands which may optionally be supported and vendor unique commands. The standard commands supported by the MVME350 are listed below. A complete description of QIC-02 commands can be found in the ANSI QIC-02 Interface Standard.

Beginning of Tape	Read Data
Erase Tape	Read Filemark
Re-tension Tape	Read Status
Write Data	Write Filemark

VMEbus INTERFACE

The VME350 may be a 24-bit (standard) or 32-bit (extended) VMEbus master to support Motorola's extended address line of modules. All accesses to the VMEbus are made through an 8Mb window. This is accomplished using a 9-bit extended address register which allows access to the entire 4 gigabytes available in 32-bit systems. Address modifiers are asserted under software control which allows the VME350 to utilize any addressing mode such as Short I/O, Privileged, Non-privileged, Program, and Data. Typical access times are listed in Table 1.

The VME350 implements a Release-On-Request (ROR) VMEbus requester. All bus request lines are monitored and the bus is relinquished anytime another bus master requests the bus. Early bus busy (BBSY*) release is also used to hide the VMEbus arbitration time when the bus is requested during a VME350 VMEbus cycle.

Table 1. Typical Master Access Times VME350 To Memory Modules

		Access	Cycle
MVME202 (150 ns DRAMs)	R	330 ns	800 ns
	W	140 ns	600 ns
MVME204-1 (150 ns DRAMs)	R	370 ns	800 ns
	W	260 ns	700 ns
MVME214 (120 ns SRAMs)	R	260 ns	660 ns
	W	360 ns	710 ns

CONTROL/STATUS REGISTERS

An IPC Control and Status Register (IPC-CSR) is provided for another VMEbus master to control the IPC initialization and operation. This shared area is 32 bytes deep and may be mapped on any 256-byte boundary in the 64K VMEbus Short I/O space via an 8-position switch. Functions that are provided include: registers for passing addresses and address modifiers, a Test and Set (TAS) bit for CSR control, a reset (RST) bit for holding the IPC in reset, and an (ATN) bit for interrupting the IPC microprocessor (see Figure 4).

DMA DATA TRANSFER

Using the onboard 16Kb static RAM as a data buffer, MVME350 Streaming Tape Controller firmware facilitates DMA-like transfer in either direction through the QIC-02 interface of data blocks of arbitrary size. This process requires use of the CSR register for host/controller signaling, draws on the buffered pipe communication channel capabilities implemented in the firmware and exercises the bus master capabilities supported by the hardware.

Typically, an application program or operating system driver is responsible for communicating its requirements to the VME350 via the command channel and for directing the other mass storage controller involved in the transaction to transfer the block of data to/from system memory. Because the tape controller is informed of the quantity and location in system memory of/for the data and can accommodate the QIC-02 streaming tape drives' 512Kb data block size requirement, it can automatically execute a transfer through the interface of data blocks of any size.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Compatibility Computer Streaming Tape	 VMEbus compatible microcomputer system QIC-02
Streaming Tape Data Rate	90Kb/s 200Kb/s Burst to Buffer
Data buffer	16Kb static RAM
VMEbus DMA transfer rate	System memory dependent
Power requirements	+5 Vdc ±5% @ 3.5 A
VMEbus Slave Mode Address Data Interrupter	 A16, VMEbus Short I/O D16 Level and vector software programmable
VMEbus Master Mode Address Data Requester	 A24 or A32 D16 Release-On-Request
Temperature Operating Storage	 0°C to 55°C −40° to 85°C
Relative humidity	5% to 95% (non-condensing)
Dimensions Height Width Thickness	 9.25 inch (23.49 cm) 6 inch (36.83 cm) 0.6 inch (1.52 cm)

CSR Mapping on the Bus is shown in Figure 5.

Base Address Offset	Even D15	Odd D0
0	A31 IPC Addr	Register
2	IPC Addr	Register A0
4	IPC AM Register	Unused
6	IPC Control Register	Reserved
8	IPC Status Register	Reserved
A	IPC MDB/ID Register	Reserved
C	IPC Abort Vector Register	Unused
E	TAS	Register

The above addresses are relative.

Figure 5. Control Status Registers

ORDERING INFORMATION

Part Number	Description
MVME350	VMEmodule Streaming Tape Controller. Includes User's Manual and IPC Firmware User's Manual.
MVME350/D	VMEmodule Streaming Tape Controller User's Manual
MVME350FW/D	MVME350 IPC Firmware User's Manual

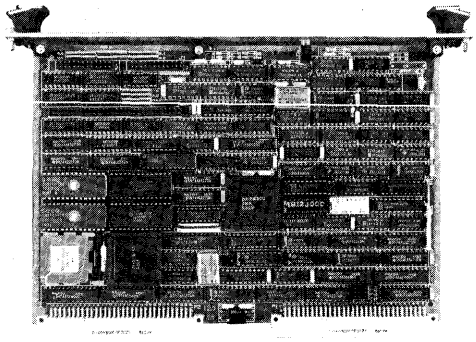
RELATED DOCUMENTATION

HB212/D	VMEbus Specification Manual
MVME1121V	VMEsystem 1121 Manual for VERSAdos
MVME1131V	VMEsystem 1131 Manual for VERSAdos
MVME1121U	VMEsystem 1121 Manual for SYSTEM V/68
MVME1131U	VMEsystem 1131 Manual for SYSTEM V/68

MVME360
MVME360UX
MVME360VX

VMEmodule™ Disk Controller /Formatter

For SMD-Compatible Disk Drives In VMEbus-Based Systems



- Supports Two SMD Disk Drives
- Multitasking Architecture
- Zero-Latency Reads and Writes
- 8-, 16- or 32-Bit Data Transfers
- 16-, 24-, or 32-Bit Addressing
- Disk Data Rates To 20 Mb/s
- Ultra High Speed DMA
- 12Kb of Onboard Memory
- Onboard Error Correction — 32-Bit ECC
- Software Programmable Interrupt Levels (1–7)
- Selectable Bus Priority (0–3)
- Double High Board VMEmodule

The MVME360 is a microprocessor-based intelligent Controller/Formatter used to interface VMEbus-based systems to SMD-compatible disk drives. It uses a 68000 microprocessor, plus two high-speed bipolar state machines in a multitasking configuration providing very high average throughput rates. Its unique virtual buffer architecture reduces or eliminates data-transfer delays caused by disk rotational latency, and data overrun/underrun problems associated with FIFO-based controllers. It offers a 1:1 disk interleave and a caching scheme designed for UNIX™, VERSAdos and similar operating systems.

GENERAL DESCRIPTION

The MVME360 Controller/Formatter features programmable sector and gap sizes, interleave factor and ultra high interface speed that ensures optimum performance with all SMD-compatible disk drives. Zero-latency operation guarantees that it will never take more than one revolution to transfer an entire data track, and spiral formatting allows multisector/multitrack operations without missing a disk revolution. The disk can be addressed either physically or by logical sectors, and up to two drives may be supported, even if they are not of the same type.

A double high VMEbus controller, the MVME360 acts as a bus master supporting extremely high DMA rates. Interrupt levels (1–7) are software programmable on a transaction-by-transaction basis and bus priority (0–3) is strap selectable. The controller supports disk data rates to 20 Mb/s, provides onboard error correction with 32-bit ECC, and an interrupt-on-drive-status-change capability for true overlapped seeks.

Figure 1 is a functional block diagram of the MVME360.

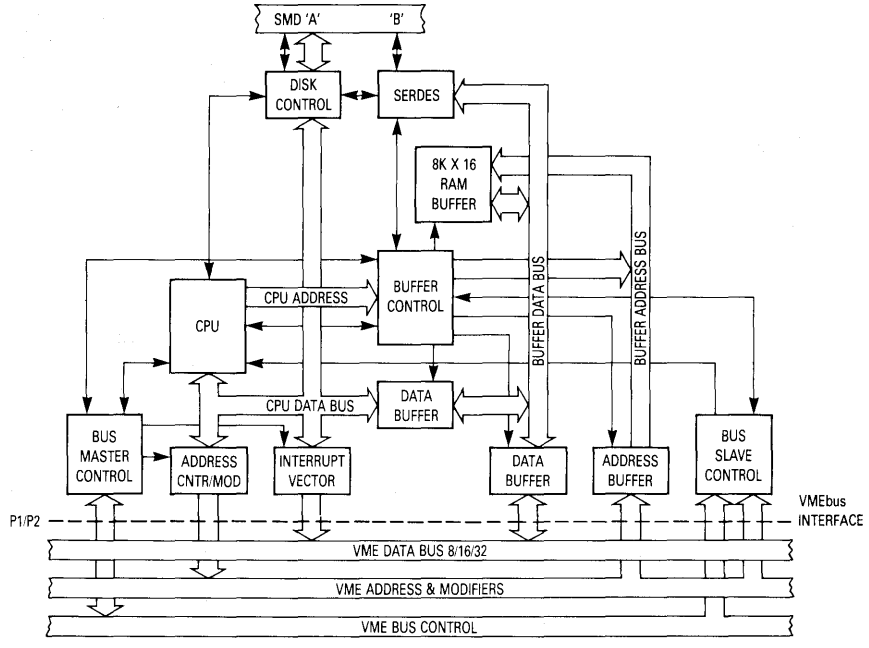


Figure 1. MVME360 Functional Block Diagram

SYSTEM CONFIGURATION

The key to the MVME360 is its multitasking, virtual buffer architecture. Two bipolar state machines are used to manage the high speed data streams. One manages data movement between the MVME360 and the VMEbus; the other manages data movement between the MVME360 and its disk drives. These state machines operate independently allowing simultaneous data movement between the MVME360 and the bus and between the MVME360 and its peripherals.

The 68000 microprocessor is used to regulate activity on the controller. It breaks up commands, sets up data transfer operations, and most importantly, it manages the 12Kb of onboard memory which can be treated as a series of sector buffers. At any given time, individual buffers may be allocated to either the disk or VMEbus processes. The 68000 will dynamically allocate and deallocate buffers as they are requested or released by the various processes. Since each device can draw on the pool of buffers, overrun and underrun problems are eliminated.

With a large pool of buffers available, the MVME360 is able to apply techniques which can reduce or eliminate disk rotational latency. A traditional controller, upon receiving a multi-sector request from the operating system, will wait until it encounters the first requested sector before beginning to read and transfer data. Thus, it will undergo, on the average, a rotational latency equivalent to half a track, or eight milliseconds. If the request is for a full track of data, the traditional controller will take, on the average, a revolution and a half to accomplish the transfer. The MVME360, on the other hand, will begin reading as soon as the head lands on the track and will begin transferring data as soon as it encounters *any* of the sectors of interest, without waiting to rotate around to the beginning of the requested string. Thus, the MVME360 will never take more than a single revolution to transfer an entire track of data.

CACHING

While the MVME360 can operate continuously at a 1:1 interleave over multiple cylinders of data, in many cases its intelligent caching scheme will provide performance better than the traditional 1:1 interleave. When the MVME360 has completed a read operation and transferred the data requested, it will continue to read and cache sectors until it has filled all its available buffers or until it receives another command from the host. Thus, if subsequent requests from the operating system are for sectors logically contiguous with those previously requested, as is frequently the case in file-oriented transactions, these requests can be satisfied directly from the cache without having to access the disk. In disk intensive applications, this can greatly improve overall system throughput.

This form of caching is particularly useful for UNIX, UNIX-like, and VERSAdos operating systems. Tests have

shown improvements in disk operation averaging greater than 40 percent, compared to a traditional 1:1 interleave without caching.

Caching shows the greatest improvement when the disk activity is characterized by a large number of short transactions. It becomes less important as the transaction size increases. However, as the transaction size becomes larger, the ability to reduce rotational latency comes into play. For transactions of a track or larger, rotational latency will go to zero. This is a very powerful technique which can show greatly improved disk response for the full range of transaction sizes.

The MVME360 will transfer data from its buffer to system memory as soon as the disk read operation begins; it does not have to wait for the entire sector to be read from the disk. This provides maximum performance and operation similar to that of a FIFO without the data overruns and underruns inherent in FIFO-based systems.

FUNCTIONAL ALTERNATIVES

The MVME360 provides a great many options, most of which are under software control. The controller supports 8-, 16-, and 32-bit data transfer bus options and 16-, 24-, and 32-bit address line options. Commands are received in slave mode and data transfers are done with the MVME360 as the bus master. The disk may be addressed by physical cylinder, head or sector or by logical sector number. If logical sector mode is chosen, the logical sector address is converted to a physical address by the onboard 68000, thus relieving the system CPU of that task.

Sector sizes are programmable on any even word count from 128 to 2048 bytes. Interrupt levels are software programmable on a per command basis to facilitate multiprocessor applications. Automatic error correction with 32-bit ECC for 11-bit correction is selectable on a drive-by-drive basis. Overlapped and implied seeks are supported on all drives. Defective media replacement is accomplished either through alternate track mapping or sector slipping.

CONTROLLER OPERATION

I/O PARAMETER BLOCK (IOPB)

An IOPB is a list of parameters necessary to define a macro function or string of functions. Those parameters are built in the IOPB either at program assembly time or dynamically by the disk I/O drivers before the transaction is started.

The MVME360 uses the simple concept of an IOPB in combination with switch selectable I/O registers to implement the mass memory interface. It transfers data in and out of system memory by becoming the system master and doing a DMA.

In order to start the disk/memory transaction, simply output the IOPB to the controller, then output a GO to the command register. The transaction will be automatically completed and the user will be interrupted when done.

Table 1. IOPB Format

Word #	Description
0	Command Code and Options
1	Status and Error Codes
2	Cylinder Select
3	Head/Sector Number
4	Sector Count
5	Buffer Address ..... MSW
6	Buffer Address ..... LSW
7	Memory Type/Address Modifier
8	Interrupt Level/Normal Complete Vector
9	DMA Burst Count/Error Vector
10	IOPB Pointer ..... MSW
11	IOPB Pointer ..... LSW
12	IOPB Memory Type/Address Modifier
13	Skew Offset

Table 2. Command I/O Register

15 0

UNIT 1 STATUS								UNIT 0 STATUS							
SLED	BOK	SFEN	BDCLR	0	0	0	BERR	GO/ BUSY BIT	OPER DONE (INT 0)	STATUS CHANGE (INT 1)	ERROR LAST CMD	0	STATUS CHANGE SOURCE	0	0

COMMAND/STATUS REGISTERS

The command I/O register serves double duty as a status register and always contains the status of the current transaction and drive. The user may optionally choose to operate using interrupts (software selectable to one of seven levels) or using status only via the status byte of the IOPB.

Multisector transfers are allowed in one IOPB, assuming contiguous sectors. Discontiguous sectors may be linked in one macro instruction by chaining IOPBs, each block pointing to the next IOPB in sequence. In fact, a string of dissimilar commands (READ, then WRITE, etc.) can be linked using this facility. Functions other than READ and WRITE are also provided. Data may be verified (checked for errors) without any data transfer, a new disk can be formatted, and the head can be explicitly moved (SEEK) to any cylinder for such things as speed optimization and overlapped seeks. The bad track and sector mapping features allow disk surface imperfections to be automatically avoided. The RESTORE command forces the head to move to cylinder 0 and clears all drive logic. The RESET command clears all controller logic.

ERROR CORRECTION

Over 50 error codes indicate the source of almost all error conditions. In case of data errors the MVME360 will use hardware ECC for high speed error correction. A 32-bit ECC code is used for error detection. Up to an 11-bit data error can be corrected. Firmware error recovery handles other error types. In case unrecoverable errors occur, a status code of 82 will indicate an error, and the type of error will be indicated in the error code byte of the IOPB. The MVME360 does comprehensive error checking on both headers and data fields on the disk, and reports the nature of any problem that may exist.

Table 3. Command Codes

70 Diagnostic	87 Initialize
71 Read Long	89 Restore (return to zero)
72 Write Long	8A Seek
74 Read Header	8B Re-format
77 Report Configuration	8C Format with sector data
78 Write Buffer	8F Reset
79 Read Buffer	90 Map Sector
81 Read Sector(s)	94 Read Non-cached
82 Write Sector(s)	97 Clear Fault
83 Verify Sector(s)	98 Clear Status Change
84 Format Track	99 Verify Track
85 Map Track	9A Track ID
86 Handshake	9B Fetch and Execute IOPB

Table 4. Error Codes

10	Disk Not Ready	2D	Seek Timeout Error
11	Invalid Disk Address	2E	Busy Timeout
12	Seek Error	2F	Not on Cylinder
13	ECC Code Error-Data Field	30	RTZ Timeout
14	Invalid Command Code	40	Unit Not Initialized
16	Invalid Sector in Command	42	Gap Specification Error
18	Bus Time Out	4B	Seek Error
1A	Disk Write Protected	4C	Mapped Header Error
1B	Unit Not Selected	50	Sectors Per Track Specification Error
1E	Drive Faulted	51	Bytes/Sector Specification Error
23	Uncorrectable Error	52	Interleave Specification Error
27	Data Overrun	53	Invalid Head Address
28	No Index Pulse on Write Format	5D	Invalid DMA Burst Count
29	Sector Not Found	60	IOPB Failed
2A	ID Field Error — Wrong Head	61	DMA Failed
2B	Invalid Sync in Data Field		

UNIT INITIALIZATION BLOCK

SMD drive characteristics and system configurations vary widely in terms of capacity, number of heads, number of sectors/track, number of bytes/sector, format gap sizes and optimum sector interleaving factor. In addition, a system can include more than one type of drive. To cope with this, MVME360 uses a simple memory resident Unit Initialization Block (UIB), in conjunction with the INITIALIZE command, to set the operating parameters for each attached unit. This unique feature allows a system to be optimized to get the maximum data capacity, systemwide throughput, and reliability out of the drive(s) being used. It also allows optimizing for software operating system characteristics and system bus and multiple-master loading considerations. The UIB also allows the user to enable/disable ECC correction, set the maximum number of retrys, enable/disable caching, enable/disable transferring uncorrectable data into memory, enable/disable Re-seeking, automatic increment by head or cylinder on multitrack commands, enable/disable interrupt on drive status change and other operation options.

Table 5. UIB Format

Byte #	Description	
0	Volume Zero Starting Head No.	
1	Volume Zero No. of Heads	
2	Volume One Starting Head No.	
3	Volume One No. of Heads	
4	Sectors Per Track	
5	Spiral Skew Factor	
6, 7	Bytes Per Sector	MSB, LSB
8	GAP1	
9	GAP2	
10	Sector Interleave	
11	Retry Count	
12, 13	Number of Cylinders	MSB, LSB
14	Attribute Flags	
15	Reserved	
16	Status Change Interrupt Level	
17	Status Change Interrupt Vector	

Table 6. Unit Status Register Detail

MSB							LSB
UNIT READY	UNIT PRESENT (SELECTABLE)	SEEK ERROR	ON CYLINDER	FAULT	BUSY	WRITE PROTECTED	DRIVE READY

Table 7. Command Options Detail

7	6	5	4	3	2	1	0
UNIT #	VOLUME #	LINK IOPB	TRANSLATE LOGICAL SECTORS	RESERVE DUAL PORT	DISABLE ERROR DETECTION	ENABLE INTRPT	ENABLE ECC

COMMAND OPTIONS

The command options byte identifies the addressing mode to be used, data bus width (BYTE, WORD, or LONG WORD mode), and whether the IOPB is to be linked to another IOPB. The DMA Count is a throttle to set transfers per bus acquisition which controls the bus latency, due to disk activity, seen by other bus masters.

The parameters in the UIB, which generally are set only once after power up, plus the command options and DMA count bytes, which could vary from operation to operation, make the MVME360 the most versatile SMD controller available.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristic	Specification
Configuration	DTB Master: A32, D32 for DMA Transfer DTB Slave: A16, D16 for Commands and Status — Requester: Any of R (0–3) (Stat) — Interrupt Handler: Any of I (1–7) (Dyn)
Environmental Limits Operating Temperature Maximum Humidity	 0° to 55°C 10% to 90% (non-condensing)
Power Requirements	3 A max @ +5 Vdc 0.5 A max @ −12 Vdc
Physical Dimensions	Double High Eurocard format 9.2 in. (233 mm) 6.3 in. (160 mm)

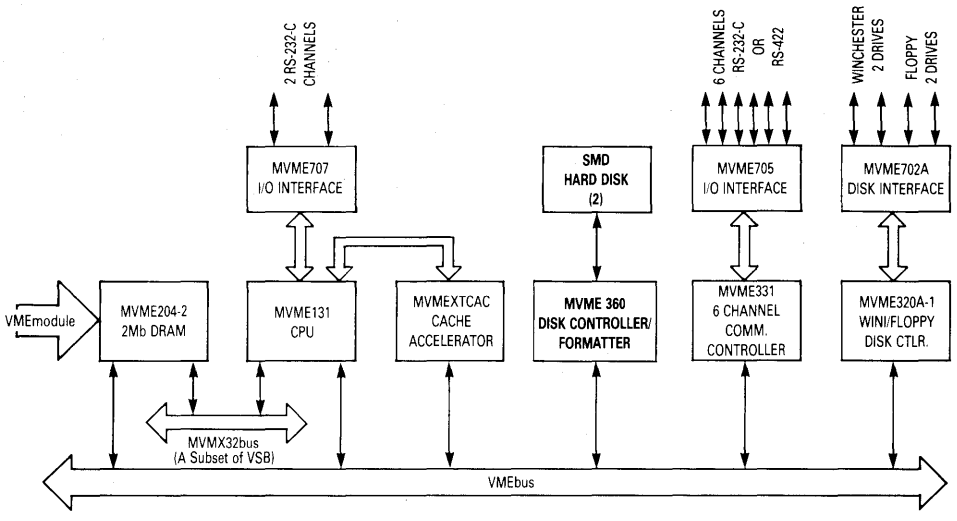
ORDERING INFORMATION

Part Number	Description
MVME360	Intelligent SMD Disk Controller. Includes User's Manual.
MVME360UX	Intelligent SMD Controller with SYSTEM V/68 UNIX Driver. This driver features standard SYSTEM V/68 driver interface and functionality, and includes normal and alternate track formatting support, disk initialization and format utilities, and error and system activity logging support. It supports one or two SMD or XSMD compatible drives on each of up to two MVME360 controllers. It can be used in Motorola MVME121-based systems running SYSTEM V/68 Release 2 Version 1.1 or later, and in MVME131-based system using SYSTEM V/68 Release 2 Version 2.1 or later. Includes User's Manual.
MVME360VX	Intelligent SMD Controller with VERSAdos driver. This driver features a general interface and may be used in a VMEsystem based on MVME121, MVME123, MVME130, or MVME131 Microprocessor Modules running VERSAdos version 4.4 or later. Includes User's Manual.

RELATED DOCUMENTATION

Part Number	Description
MVME360/D1	MVME360 User's Manual.
HB212/D	VMEbus Specification Manual.

The VMEsystem



A Typical High-Speed Microcomputer VMEsystem

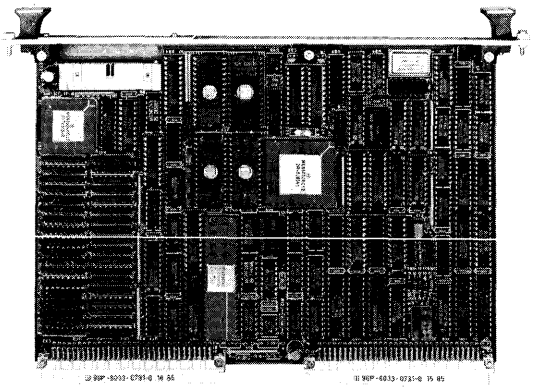
The MVME360 Disk Controller/Formatter Module conforms to the VMEbus interconnect system which has emerged as a world-wide Standard for 8-/6-/32-bit microcomputers. It is part of the VMEmodule product family whose M68000-based processing power, Eurocard mechanical format, powerful VERSAdos real-time software support and international multiple sourcing have made it one of the most popular product lines in the industry.

The VMEsystem architecture, and a board-level component complement of a typical VME-based system are illustrated in the above block diagram. Of particular significance are:

- The well-established functional versatility of the VMEbus whose non-proprietary technical specification is widely used for system implementation
- An auxiliary MVMXbus for high-speed memory expansion. MVMXbus is a subset of the VME Subsystem Bus
- An MC68020-based, 32-bit CPU board (MVME131) utilizing the latest advances in microprocessor design
- A large repertoire of peripheral (controller) modules and memory boards

The growing acceptance of the VMEsystem philosophy now yields a selection of compatible board-level products from well over a hundred vendors, offering systems manufacturers the basic tools with which to implement their designs quickly, efficiently, reliability and cost effectively.

VMEmodule™ Advanced MAP Network Interface



A new VMEmodule, the MVME372 Advanced Manufacturing Automation Protocol Network Interface provides connection between an IEEE 802.4 Broadband Token Bus Modem and the VMEbus. To implement all seven layers of the MAP 2.1 protocol standard (except the physical layer), the MVME372 utilizes the latest Motorola high performance VLSI devices including the MC68824 Token Bus Controller and the MC68020 32-Bit Microprocessor.

The MC68824 is an 84-Pin Grid Array implementation of the IEEE 802.4 Token Bus Media Access Control (MAC) sublayer protocol. It uses a powerful shared memory structure to provide the full set of MAC services as well as network monitoring and diagnostics aids. The device also incorporates an intelligent DMA function and a 40-byte FIFO used to support the MAC protocol at data rates up to 10 Mbits/sec.

The MVME372 has a generic serial interface to the off-board Modem which allows connection to either a broadband or carrierband physical layer. A broadband Modem board, MVME371FS, is presently offered for this purpose. Rapid data throughput for all seven MAP layers and fast real-time response is achieved via the power of the MC68020 32-Bit Microprocessor.

MVME372 Features:

- Supports All Seven MAP 2.1 Layers
- MC68020 32-Bit Microprocessor (12.5 MHz)
- MC68824 Implementation of MAC Sublayer Includes:
  - — Intelligent DMA Controller
  - — 40-Byte FIFO
- Serial Interface Supports Connection to Broadband, Carrierband or other Physical Layers
- 640Kb of Dynamic RAM — 128Kb shared by VMEbus/MC68824/MC68020
- Full A32:D16 or A24:D16 VMEbus-master Interface
- VMEbus Interrupter with Programmable Level/Vector
- Full A32:D16 or A24:D16 VMEbus Slave Interface
- VMEbus Requester with Release On Local Cycle Mode
- 256 x 4 of EEPROM-Based NVRAM for Station Address and other Variables
- Debug Serial Port (RS-232-C compatible) for System Configuration and Debugger Firmware
- Watchdog and Tick Timers
- Memory-Mapped Slave Interrupt
- Partial VMEbus System Controller for Standalone Operation
  - — Single Level Arbiter
  - — Global Bus Timeout
  - — RESET Driver
  - — SYSFAIL Monitoring
- Level three (only) VMEbus Interrupt Handler for Standalone Operation
- Four 28-Pin JEDEC Standard Sockets for EPROMs — sizes 8K x 8 to 64K x 8 or EEPROMs sizes 8K x 8 to 32K x 8
- Programmable VMEbus Base Address
- Module-Status LED
- Power-on/Reset PROM-based Diagnostics
- 40-Pin Standard Modem Interface from Front Panel or P2 (back connector)
- FCC Compliant (Class A, Subpart J of Part 15)

GENERAL DESCRIPTION

A major architectural feature of the MVME372 is the division of the 640Kb of dynamic memory into 256Kb of 32-bit wide memory and 384Kb of 16-bit wide memory. This allows the MC68020 to execute at top speed out of the 32-bit wide memory and also simplifies the token bus chip interface to the 16-bit wide memory. Although the TBC interfaces only the 384Kb of wordwide memory, the dynamic bus sizing feature of the MC68020 permits it to access all 640Kb of memory. Execution time for programs in both 32-bit and 16-bit wide memory is minimized by the instruction cache of the MC68020.

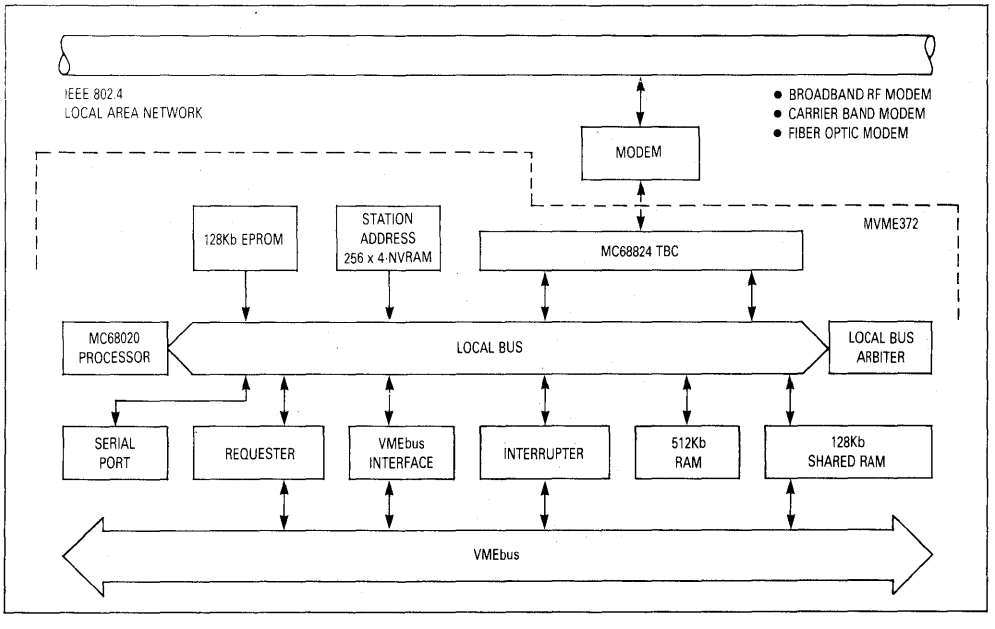
In addition to the dynamic RAM, up to 256Kb of EPROM can be used (or 512Kb of masked ROM). For fast MC68020 execution, this memory is 32-bits wide. If EEPROM rather than EPROM is desired, the module can accomodate up to 128Kb.

For standalone operation such as might be needed in a bridge or gateway, the MVME372 can be used as a partial VMEbus system controller. An Ethernet-to-token bus bridge, for example, could be implemented using just three modules (MVME330, MVME372 and an MVME371 Broadband Modem) avoiding the need for a separate MPU or system controller board. Another implementation would be multiple token bus-to-X.25 or token bus-to-RS-232-C connections using MVME331 or MVME333-2 Six-Channel Serial Communication Modules. The VMEbus system controller features of MVME372 include a single level arbiter, global bus timeout circuitry, a RESET line driver, SYSFAIL line monitoring and a level three VMEbus interrupt handler.

Careful attention is given in the MVME372 design to ensure that at the full 10 megabit data rate the TBC is protected from underrunning or overflowing. In the local bus arbiter, highest priority is given to the TBC. If the TBC needs the local bus during an MC68020 VMEbus access, a retry is given to the MC68020. The retry access then causes control of the local bus to be immediately given to the TBC. Circuitry is included that ensures conformance to the VMEbus specification during internal retries.

Motorola has developed software, called the Common Environment, that provides a standard means of communicating with other host processor and communication modules on the VMEbus. The MVME372 is shipped with installed Common Environment and debugger firmware.

Capabilities provided to the user by the Common Environment firmware include the downloading of software, inter-task communication, local and global resource control and access to a real-time operating system. The inter-task communication feature allows MVME372 tasks to communicate with other MVME372 tasks as well as communicate with host or other off board tasks. Local and global resource control provides buffer pool management and message passing across the VMEbus. In addition, the Common Environment does not use a proprietary operating system. Rather, it uses a fully debugged, application-proven system in common use — VRTX. VRTX is supplied under license from Hunter and Ready Incorporated, and is one of the industry's standard real-time kernels.



MVME372 Advanced MAP Network Interface

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristic	Specification
Power Requirements	+5 Vdc, 4.2 A (typ), 4.5 A (max) ±12 Vdc, 0.025 A (typ), 0.050 A (max)
Operating Temperature	0° to 65°C
Storage Temperature	−40° to 85°C
Relative Humidity	5° to 95° (non-condensing)
Operating Altitude (max)	10,000 ft. (3,000 m)
Physical Dimensions — Height Depth	 9.2 in. (233.4 mm) 6.3 in. (160 mm)
Connectors — To VMEbus To Modem To Debug Cable	 Two VME standard 96-Pin male DIN 40-Pin 3M 3595-5002 Use above 96-Pin male DIN of P2

ORDERING INFORMATION

Part Number	Description
MVME372	VMEmodule Advanced MAP Network Interface, including an MC68020 32-Bit Microprocessor, MC68824 Token Bus Controller, 640Kb DRAM, serial debug port, Modem interface and VMEbus interface. Includes Common Environment and debug firmware. Includes complete set of User's Manuals.
MVME372 D	VMEmodule MAP Interface User's Manual.

OTHER COMMUNICATION VMEmodules

Part Number	Description
MVME330-1	VMEmodule Ethernet LAN Controller. Provides high performance intelligent single board connection of VMEbus System to Ethernet, a Local Area Network. Includes 512K DRAM, LANCE (7990), SIA (7991), 68000 MPU, Kernel Firmware and Power Up Self Test.
MVME331	Six-Channel Synchronous/Asynchronous Communications Controller. Includes User's Manual.
MVME333-2	Six-Channel Synchronous/Asynchronous Communications Controller with DMA. Includes User's Manual.
MVME371FS-1 MVME371FS-2 MVME371FS-3	VMEmodule MAP Network Interface Modem board, providing all logic, RF circuitry and standard coax connectors required for interfacing MAP 2.1-compatible broadband network cable. Includes User's Manual.
SYS1121UY221	Hardware host package for Advanced Map Network Interface, comprising VMEbus chassis with MC68010-based microcomputer, 1Mb RAM, disk controller, 40Mb Winchester disk and 655Kb floppy disk. Includes SYSTEM V/68 Operating System derived from UNIX System V plus comprehensive User's Manual.
SYS1131UY331	VMEBUS Modular System with SYSTEM V/68, includes MVME131 VMEmodule Monoboard Microcomputer MVME204-2 2Mb Dual-Ported DRAM MVME320A-1 VMEbus Disk Drive Controller MVME050 System Controller MVME701A Transition Module MVME707 Transition Module MVME833 Plug-In Mass Storage Module with — 70Mb Winchester Drive — 655Kb Floppy Disk Drive MVME945 VMEbus Chassis Assembly with — 32-Bit VMEbus Backplane — 400 W Power Supply User's Manuals for SYSTEM V/68, chassis and all modules included

MVME372BBKIT
MVME372BBKIT2

MAP Broadband Network Developer's Kit

In one convenient product, the MAP Broadband Developer's Kit provides all software and hardware needed for system integrators to begin developing a complete three node, 10Mbits/sec, broadband, token bus local area network. The kit can be an especially useful design tool when used with a VMEbus host system.

To aid developers in verifying that an assembled LAN is operating properly, demonstration software and comprehensive user documentation is provided in the kit. Complete MicroMAP interface documentation is also provided to accelerate the development of application programs.

All hardware components in the MAP Broadband Network Developer's Kit are constructed of materials rated by the Underwriters Laboratories (UL) for application in a test environment or in a permanent installation.

- Three Node VMEmodule MAP Network Interface Board Set
  - — Adheres to IEEE 802.4 and MAP 2.1 Standards
  - — Executes MicroMAP Layers 1–7 of MAP 2.1 including: IEEE 802.4 Physical and Media Access Control (MAC) and IEEE 802.2 Type 1 Logical Link Control (LLC) which combine to create Layer 2, Internet Network Layer, Class IV Transport Layer, Session and Application Layer (includes CASE, FTAM, MMFS and User Interface Routines)
  - — Supports Downloading of Network Protocols and Services from a VMEbus host running SYSTEM V/68
  - — Contains the MVME372 Advanced MAP Network Interface Module based on a 12.5 MHz, 32-Bit MC68020 Microprocessor with 640Kb DRAM, MC68824 Token Bus Controller Chip, A32:D16 or A24:D16 VMEbus Interface and IEEE Compliant MAC-to-Physical Layer Interface for Modem connection
  - — 10Mbit/sec Broadband Token Bus Modem Module for use with the 3'/4''P/Q Channel Group
- 19" Rack Mountable Head-End Remodulator serving as Central Transmission Facility for a 12 MHz, IEEE 802.4 MAP 2.1 Broadband Local Area Network
  - — Remodulates signals from Transmit to Receive Frequency as specified by MAP 2.1
  - — Permits Selection of any of Five Separate Channel Groups via Secure Internal Switch
  - — Includes Lockable, Key-Operated ON, OFF and AUTO-TEST Switch
  - — Includes Adjustable Output Level
  - — Includes Power and Fault Detect LEDs on Front Panel
- Broadband Cable Kit for Assembling Three Node Network includes: Frequency Splitter, Four-Way Tap, Attenuator, Terminators and various Lengths of Cable.
- Software Package (on 5-1/4" diskettes under SYSTEM V/68) Package includes:
  - — Certified MAP 2.1 Object Code for Layers 1–7
  - — Routine for Download of MAP Layers from Host to MVME372 Module for execution
  - — MAP Interface Boardset Bootload and Driver Software
  - — Demo Program to Test Board Sets in a Three Node Assembled Network
- Complete Documentation Package includes:
  - — User's Manuals for the MVME372 MAP Network Interface Module and MVME371FS-1 Broadband Modem.
  - — Network Assembly Instructions
  - — MicroMAP User's Manual
  - — Demo Program User's Instructions

2

BROADBAND NETWORK KIT COMPONENTS

HEAD-END REMODULATOR

The head-end remodulator supplied with a Broadband Developer's Kit is a 19" rack-mountable system used as the central re-transmission facility for an IEEE 802.4, 10Mbits/sec Manufacturing Automation Protocol (MAP) broadband local area network. It provides translation between received and transmitted signals and decodes and re-encodes data. It is housed in a chassis 19" wide x 3.5" high x 16" deep (including rear panel connectors). Handles and switches protrude an additional 1.7" beyond the front panel. The unit operates at 110 Vac — +10%, −13% (also available in 220 Vac — +15%, −10%). Operating temperature range is 0°C to +50°C.

Two type F connectors on the rear panel provide for input and output signals to and from the network. On the front panel, two LEDs indicate power and fault conditions. Controls are provided for adjusting transmit levels. A switch is also provided for selecting one of the three MAP 2.1 channel groups: 3'/4'/P/Q, 4A'/5'/R/S and 6'/FM1/T/U.

BROADBAND COAX CABLES

The flexible three, ten, twenty-five and fifty foot cables in the kit are 1/4" in diameter and are fabricated of RG-59 coaxial cable that has two foil and two braided shields for maximum noise immunity. Type F connectors are used. The cables comply with the requirements of U.L. Style 1478 for Interconnection of Electrical Equipment. Structural return loss and attenuation tests are run on each cable.

The four lengths of cables are interchangeable to accommodate different lab configurations.

FREQUENCY SPLITTER

Separation of the lower frequency signals being transmitted by a Modem board from the higher frequency signals being received by Modem board is performed by a frequency splitter. Over the transmitted or received bandwidths, the insertion loss is less than 1 decibel.

ATTENUATOR

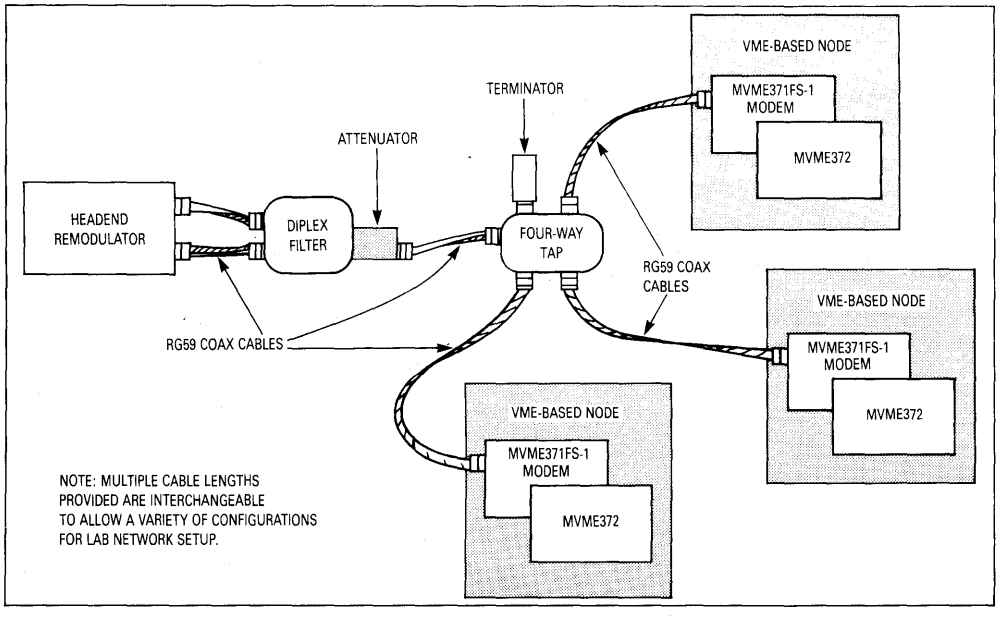
Connected between the frequency splitter and four-way tap, the attenuator introduces an additional 10 dB of attenuation for best results on the network when using a four-way passive tap.

FOUR-WAY TAP

A tap provides the means for a node to be connected to the network. The four-way tap provided with the kit introduces 30 dB of attenuation between the input from the head-end remodulator and an output to a node Modem resulting in a total of 40 dB for the path from remodulator to Modem board.

TERMINATORS

The 75 ohm terminators supplied with the kit are installed on the four-way tap in the absence of a node connection. Since they have the characteristic impedance of the coaxial cable, they absorb unused transmitted energy with minimum reflection.



Broadband Developer's Kit

RELATED PRODUCTS

Part Number	Description
SYS1121UY221	VMEbus Modular System with SYSTEM V/68 suitable for use as host for MAP Network Interface. Hardware/Software package includes: SYSTEM V/68 Operating System MVME202 512Kb DRAM Module MVME320A-1 Disk Controller MVME050 System Controller 40Mb Winchester Disk Drive 655Kb Floppy Disk Drive Three RS-232-C Serial Ports (total) One Centronics Printer Interface Five VMEbus Expansion Slots MVME943 19" Rack Mount Chassis Includes: — Table Top Enclosure — Transverse 400 W Mid-Chassis Power Supply — Nine-Slot, A24:D16 VMEbus-Compatible Backplane MVME1121 Microcomputer with: — MC68010 16-Bit Virtual Memory Microprocessor — MC68451 Memory Management Unit — 512Kb Dual-Ported DRAM — 4Kb Instruction Cache — A24:D16 VMEbus Interface — One RS-232-C Serial Port
SYS1131UY331	VMEbus Modular System with SYSTEM V/68 suitable for use as host for MAP Network Interface. Hardware/Software package includes: SYSTEM V/68 Operating System MVME050 System Controller MVME204-2 2Mb Dual-ported DRAM with VME Subsystem Bus (VSB) Interface MVME320A-1 Disk Controller MVME701A Transition Module MVME707 Transition Module 70Mb Winchester Disk Drive 655Kb Floppy Disk Drive Four RS-232-C Ports (total) MVME945 19" Rack Mount Chassis including: — Table Top Enclosure — Transverse 400 W Mid-Chassis Power Supply — 12-Slot, VMEbus-Compatible Card Cage with A32:D32 Backplane — Accommodates up to 20 Rear-mounted 80 mm I/O Transition Modules MVME131 Microcomputer with: — MC68020 32-Bit Microprocessor (16.67 MHz) — MMB851 Memory Management Unit — 16Kb Instruction/Data Cache RAM — Two Sockets for JEDEC 28-Pin ROM/EPROM Devices — Two Sockets for JEDEC 28-Pin RAM/ROM Devices — Two Multi-protocol Serial Ports — A32:D32 VMEbus Interface — VME Subsystem Bus (VSB) Interface — System Level Diagnostics

ORDERING INFORMATION

Part Number	Qty	Description
MVME372BBKIT		Three Node MAP Broadband Network Developer's Kit. Includes the following items:
	3	MVME372/MVME371FS-1 Board Sets for 3'/4'/P/Q Channel Group.
	3	MicroMAP1–7O Software Packages (on 5-1/4" diskettes under SYSTEM V/68) including: — MAP 2.1 Certified Layers 1–7 Object — Host Resident Interfaces and Drivers — LAN Software Download Program — Common Environment Object
	3	Kit Demo Software Packages
	1	Head-End Remodulator, 19" Rack Mountable (110 V).
	1	Cable Kit consisting of: 2 50-foot RG-59 Coax Cables 2 25-foot RG-59 Coax Cables 2 10-foot RG-59 Coax Cables 2 3-foot RG-59 Coax Cables 1 Frequency Splitter 1 Attenuator 1 Four-Way Tap 4 Terminators
	1	Set of User Documentation
MVME372BBKIT2		Same as MVME372BBKIT except has 220 V Head-End Remodulator

MVME372SET-1
MVME372SET-2
MVME372SET-3

2

VMEmodule™ MAP Network Interface

The Manufacturing Automation Protocol (MAP) Network Interface for VMEbus systems provides the connection between an IEEE-802.4 standard broadband token-bus local area network and a host system based on the VMEbus. The MAP Network Interface, which includes the 10Mbits/sec Modem, is implemented on two double-height VME boards, allowing the complete network interface to be enclosed in a VME card cage. The MAP Network Interface handles token-bus protocols with MAP 2.1 compatible hardware and software.

- Two-board VMEbus Interface — Plugs Directly into VME Backplane (occupies two slots)
- Adherence to International Networking Standards, Including IEEE-802.4 and MAP 2.1, for Compatibility with Present and Future Token-bus LAN Products
- Execution of MicroMAP Layers 1–7 of MAP 2.1 Including: IEEE 802.4 Physical and Media Access Control (MAC) Layers, IEEE-802.2 Type Logical Link Control (LLC), ISO Internet Network Layer, and ISO Class IV Transport Layer, Session Layer and Application Layer Including:
  - — CASE
  - — FTAM
  - — MMFS
  - — Network Management Agent
  - — User Interfaces to Application Layer
- 10 Mbps Broadband Token-bus Modem Board
- Controller Board, Based on a 12.5 MHz, 32-bit Microprocessor, with 640Kb of RAM Memory, Token-bus Logic, Modem Control, and VMEbus Interface (A32/D16 or A24/D16)
- Controller uses MC68824 Token Bus Controller Chip
- Industrial Design Specifications Including Operation at 0°C to 65°C
- Downloading of Network Protocols and Services from Host Processor
- 256 x 4 at EEPROM-Based NVRAM for Station Address and other Variables
- PROM-based Diagnostics following Power-on or Reset
- VRTX Real-Time Operating System

FUNCTIONAL OVERVIEW:

The MAP Network Interface is based on the IEEE-802.4 token-bus specification, and provides layers 1 and 2, the Physical and Data Link layers of the ISO Open System Interconnect model. It also implements the OSI layers 3 through 7 communications architecture specified by MAP 2.1. These layers reside in RAM after being downloaded to the Interface via either the MAP network or the local bus.

Interfaces are provided into the MAP Network Interface architecture at the application layer for CASE, FTAM and MMFS.

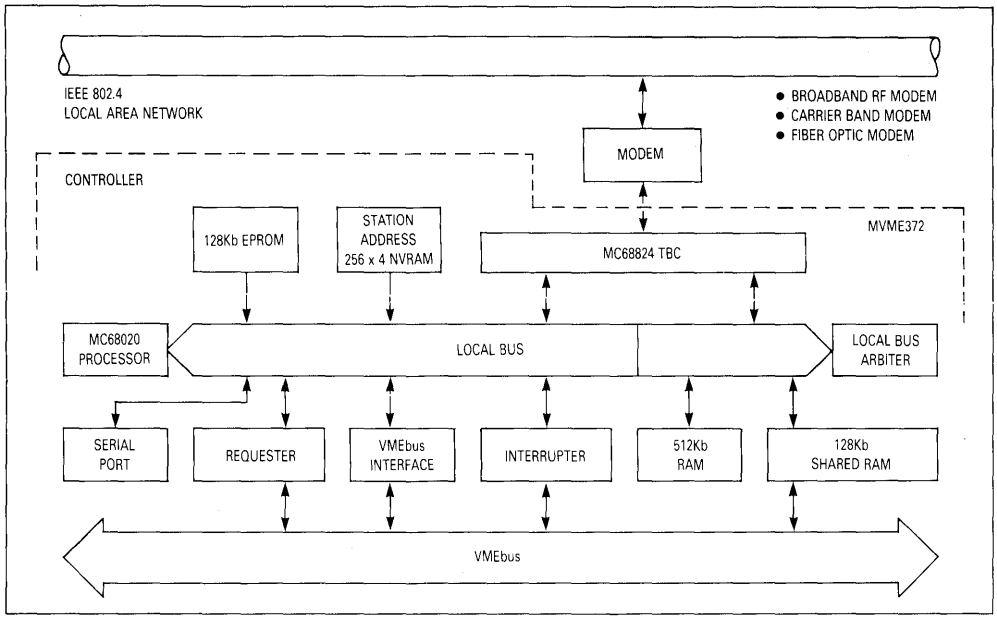
The interfaces at these layers are provided, along with documentation, to enable development of application software.

OPERATING CHANNELS:

A version of the MAP Network Interface is available for each of the broadband channel groups specified by MAP:

1. 3'/4'/P/Q
2. 4A'/5'/R/S
3. 6'/FM1'/T/U

2



VMEmodule MAP Network Interface Block Diagram

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	MVME372 Specifications	MVME371FS Specifications
Power Requirements	+5 Vdc, 4.2 A typ, 4.5 A max ±12 Vdc, 0.025 A typ, 0.050 A max	+5 Vdc, 1.5 A max +12 Vdc, 0.4 A max −12 Vdc, 0.2 A max
Operating Temperature	0° to 65°C	0°C to 70°C
Storage Temperature	−40° to 55°C	−40°C to 85°C
Relative Humidity	5% to 90% (non-condensing)	5% to 95% (non-condensing)
Physical Dimensions* Height Depth	 9.2 in. (233.4 mm) 6.3 in. (160 mm)	 9.2 in. (233.4 mm) 6.3 in. (160 mm)
Connectors — To VMEbus To LAN Cable To MVME372	Two VME standard 96-pin Male DIN	Type F Female connector on Modem board 40-Pin 3M 3595-5002

*Each board of the pair is implemented in the standard double-height VMEbus form factor. The boards occupy two adjacent slots in a VMEbus backplane, and are interconnected by a ribbon cable at the bottom edge (front panel).

MVME372SET-1, MVME372SET-2, MVME372SET-3

ORDERING INFORMATION

Part Number	Description
MVME372SET-1	VMEmodule MAP Network Interface, a two-board set comprising a Controller board and Modem board, for operation on MAP 2.1 channel group 3'/4'/P/Q. The Controller board includes a 32-bit processor, 640Kb of RAM, token bus logic, Modem control and VMEbus interface. The Modem board provides all logic, RF circuitry, and connector necessary to interface into MAP 2.1 compatible broadband network media. Also includes MicroMAP object software provided on 5-1/4" floppy disk, UNIX™ System V format, implementing MAP 2.1 layers 1–7, downloadable into the MAP Network Interface from customer-provided host. Includes comprehensive set of User's Manuals with both hardware reference and software reference documentation.
MVME372SET-2	Same as MVME372SET-1, but designed for operation on MAP 2.1 channel group 4A'/5'/R/S.
MVME372SET-3	Same as MVME372SET-1, but designed for operation on MAP 2.1 channel group 6'/FM1'/T/U.
MVMEMAPUM/D1	MicroMAP User's Manual
MVMEMAPIF/D1	Host-Resident Library Interface Manual
MVMECEUM/D1	Common Environment User's Manual
MVMECEDRV/D1	Common Environment Driver User's Manual
MVME372/D1	MVME372 MAP Interface User's Manual
MVME371FS/D1	MVME371FS Broadband Modem User's Manual

RELATED PRODUCTS

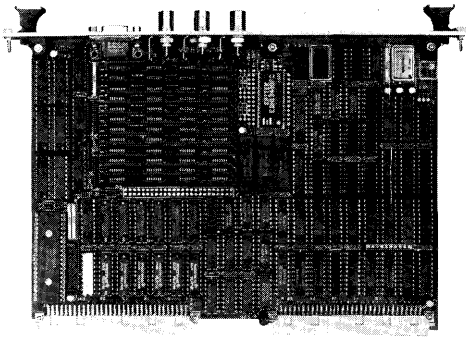
Part Number	Description
SYS1121UY221	VMEbus Modular System with System V/68 suitable for use as host for MAP Network Interface. Hardware/software package includes: MVME202 512Kb DRAM Module MVME320A-1 VMEbus Disk Controller MVME050 System Controller 40Mb Winchester Disk Drive 655Kb Floppy Disk Drive Three RS-232-C Serial Ports (total) One Centronics Printer Interface Five VMEbus Expansion Slots MVME943 19" Rack Mount Chassis Includes: — Table Top Enclosure — Transverse 400 W Mid-Chassis Power Supply — Nine Slot, A24:D16 VMEbus-Compatible Backplane MVME1121 Microcomputer with: — MC68010 16-Bit Virtual Memory Microprocessor — MC68451 Memory Management Unit — 512Kb Dual-Ported DRAM — 4Kb Instruction Cache — A24:D16 VMEbus Interface — One RS-232-C Serial Port

MVME372SET-1, MVME372SET-2, MVME372SET-3

Part Number	Description
SYS1131UY331	VMEbus Modular System with System V/68 suitable for use as host for MAP Network Interface. Hardware/Software package includes: SYSTEM V/68 Operating System MVME204-2 2Mb Dual-ported DRAM with VME Subsystem Bus (VSB) Interface MVME320A-1 VMEbus Disk Controller MVME050 System Controller MVME701A Transition Module MVME707 Transition Module 70Mb Winchester Disk Drive 655Kb Floppy Disk Drive Four RS-232-C Ports (total) MVME945 19" Rack Mount Chassis includes: — Table Top Enclosure — Transverse 400 W Mid-Chassis Power Supply — 12-Slot, VMEbus-Compatible Card Cage with A32:D32 Backplane — Accommodates up to 20 Rear-Mounted 80 mm I/O Transition Modules MVME131 Microcomputer with: — MC68020 32-Bit Microprocessor (16.67 MHz) — MMB851 Memory Management Unit — 16Kb Instruction/Data Cache RAM — Two Sockets for JEDEC 28-Pin ROM/EPROM Devices — Two Sockets for JEDEC 28-Pin RAM/ROM Devices — Two Multi-protocol Serial Ports — A32:D32 VMEbus Interface — VME Subsystem Bus (VSB) Interface — System Level Diagnostics

Product Preview

VMEmodule™ Graphics Interface Module



- Optional GKS Support for FORTAN 77 and C Applications in VERSAdos and SYSTEM V/68 Operating System Environments
- High Resolution for Bit-Mapped Graphics
  - — 1024 x 1024 Pixels Ranging to 512 x 380 Pixels
  - — Square or Rectangular Format
- Three Bit Planes for Multiple Color Graphics
- Interlaced or Non-Interlaced Operation
- TTL RGB/Monochrome Video Outputs
- RS-343-A Compatible Output
- VMEbus Slave Interface
- Accessed in Standard (24-bit) or Extended (32-bit) Address Space
- 16-Bit Data Interface
- Vertical Sync Interrupt Available
  - — Level and Vector Programmable
- Fast Data Strobe to DTACK Memory Access Time
  - — Read — 320 ns
  - — Write — 280 ns

The MVME390A Graphics Interface Module is a VMEbus-based, non-intelligent, medium-to-high resolution, raster scan interface for displaying CRT graphics. It utilizes a bit-mapped memory architecture providing three bit planes (color-planes) plus a pixel (bit) addressing capability. Figure 1 is a functional block diagram and Table 1 is a map of graphic memory relative to the modules base address.

The MVME390A Graphics Interface Module is designed for use in high resolution bit-mapped graphics displays such as those required in industrial process control and computer-aided manufacturing and design. The module is well suited, for example, for displaying medical or cartographic images or for use in document handling and simulation applications. It can be used in any application requiring fast, localized updates in frame buffers.

The MVME390A Graphics Interface Module can be used with RGB color monitors having an analog video input (RS-343-A), binary TTL color and monochrome monitors requiring pixel resolutions of up to 1024 x 1024.

DISPLAY MEMORY

Display memory on the module comprises 384Kb of video RAM which is dual ported between the VMEbus and the display refresh logic. The memory is organized as three 128Kb bit planes of 1024 x 1024 picture elements (pixels) that simultaneously produce eight colors for TTL-level color monitors, eight gray scale levels for TTL-level monochrome monitors or eight colors from a palette of 512 colors for an analog input color monitor.

Video RAM is available from the VMEbus 95% of the time. Because the video RAM design permits the refresh logic to transfer a complete line (1024 bits) into the high-speed video output shift register in just one read cycle, a memory read is required only once each 16 μs even when operating in the highest performance mode (rewriting 1024 x 1024 pixels each frame and outputting one pixel every 12 ns, approximately). The nearly unencumbered video RAM is thus available for accesses from the VMEbus while the current line is being sent to the display monitor.

The speed of the vertical scan (refresh) logic brings the advantage of image updates that can be more quickly written into display memory and the advantage of a display that does not break up when a new image is written. Also, access to the display memory need not be restricted to the short horizontal or vertical retrace/blanking intervals.

ADDRESSING MODES

The MVME390A Graphics Interface Module utilizes a flexible design that accommodates both 32-bit and 24-bit addressing. It allows the display memory to be accessed from the VMEbus on a bit-per-pixel (TTL) basis or on a word-per-pixel basis (TTL) for the purpose of transferring image data to memory for automatic display.

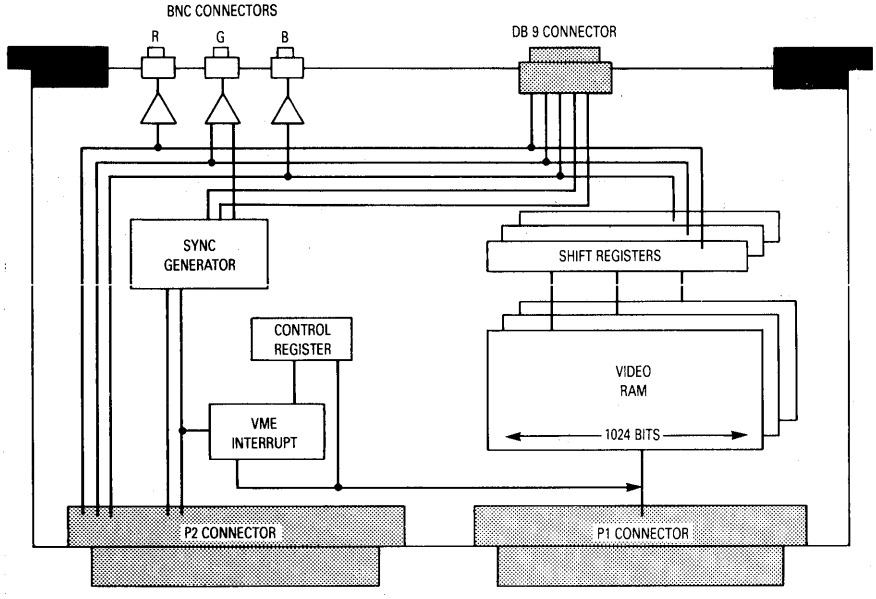


Figure 1. MVME390A Functional Block Diagram

Table 1. Memory Map

Offset Address	Description	R/W	Size
000000 – 1FFFFF	Pixel Address Block	R/W	2Mb
200000 – 21FFFF	Plane 0 Memory (green)	R/W	128Kb
220000 – 23FFFF	Plane 1 Memory (red)	R/W	128Kb
240000 – 25FFFF	Plane 2 Memory (blue)	R/W	128Kb
260000 – 27FFFF	Reserved		128Kb
280000	Not used		1 Byte
280001	Control Register 1	R/W	1 Byte
280002	Not used		1 Byte
280003	Control Register 2	R/W	1 Byte
280004	Not used		1 Byte
280005	Interrupt Vector Register	R/W	1 Byte
280006	Not used		1 Byte
280007	Reserved		1 Byte
280008	Not used		1 Byte
280009	Status Register	R	1 Byte
28000A	Not used		1 Byte
28000B	Reserved		1 Byte
28000C	Not used		1 Byte
28000D	68A45S Address Register	W	1 Byte
28000E	Not used		1 Byte
28000F	68A45S Control Register	W	1 Byte
280010 – 2FFFFF	Reserved		512Kb

COLOR PLANE ADDRESSING MODE

MVME390A design provides a fast, color plane (bit-plane) access mode that facilitates display of complete screen images and full screen updates at maximum speed. In this mode, three 128Kb memory blocks having identical bit-to-pixel correspondences reside contiguously in VMEbus space. Each block corresponds to a color plane and is accessed individually to supply the data required for that color. For a color plane image, the word-to-pixel correspondences and the bit order within a word are shown in Figures 2 and 3, respectively.

PIXEL ADDRESSING MODE

Display memory can also be addressed as a contiguous 2Mb block in VMEbus space. In this mode, a pixel corresponds to one 16-bit word. The three least significant bits of a word correspond to the red, green and blue bit planes and are automatically accessed in parallel when image data is written in the pixel word. This addressing mode conserves VMEbus bandwidth and can be more convenient for programming the depiction of line segments and vectors than the color plane mode. It can also be used for the concurrent drawing of objects on a color plane image display. The word-to-pixel correspondence and the functions of the bits within a word for the pixel addressing mode are shown in Figure 4 and Table 5, respectively.

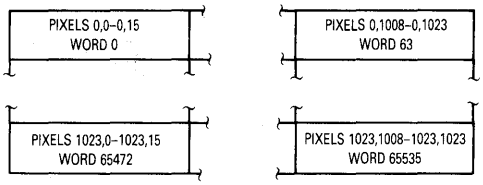


Figure 2. Word/Pixel Correspondence, Color Plane Addressing

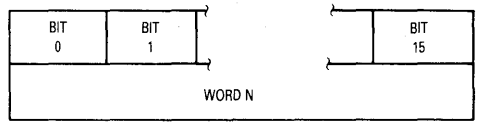


Figure 3. Bit Order, Color Plane Word

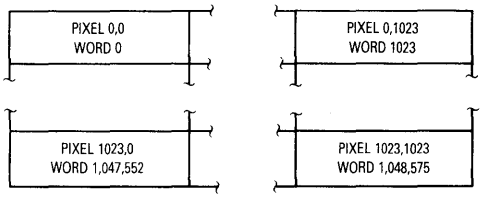


Figure 4. Word/Pixel Correspondence, Pixel Addressing

Table 2. Data Access with Pixel Addressing

Data Bits	Pixel Write Format	Pixel Read Format
D15	Not used	Undefined
D14	Not used	Undefined
D13	Not used	Undefined
D12	Not used	Undefined
D11	Reserved	Undefined
D10	Plane 2 Write Enable	Undefined
D09	Plane 1 Write Enable	Undefined
D08	Plane 0 Write Enable	Undefined
D07	Not used	Undefined
D06	Not used	Undefined
D05	Not used	Undefined
D04	Not used	Undefined
D03	Reserved	Undefined
D02	Plane 2 Write Data	Plane 2 Read Data
D01	Plane 1 Write Data	Plane 1 Read Data
D00	Plane 0 Write Data	Plane 0 Read Data

DISPLAY INTERFACE

The MVME390A supports analog-input color and monochrome monitors. The RS-343-A compatible analog video outputs are available at three front panel BNC connectors. Sync-on-green is used to provide horizontal and vertical synchronization.

Color and monochrome monitors having TTL-level inputs are also supported by the MVME390A. Three color plane outputs plus individual horizontal and vertical synchronization outputs are available both at the VMEbus connector P2 and at a DB-9 connector on the front panel for CRT's requiring signals at TTL levels. Any of the TTL-level outputs available at the VMEbus P2 connector and at the front panel DB-9 connector may be used for single level monochrome display.

GRAPHICS INTERFACE RESOLUTION

By programming the control register in the MC6845 CRT Controller device and by selecting the appropriate dot clock frequency, a large range of display resolutions, from the highest resolution of 1024 x 1024 pixels to the lowest resolution of 512 x 512 pixels may be obtained, as shown in Table 3. For example, a 1024 x 1024 pixel color display can be obtained using a dot clock frequency of 80 MHz which results in a refresh (vertical scan) rate of 56 cycles/second. The module's display interface design permits selection of almost any combination of the three parameters which, together with the scan mode, define the performance envelope for any video display application.

Table 3

Resolution	CRT Horizontal Frequency	Dot Clock Frequency (Bandwidth)	
1024 x 1024*	59.52 kHz	80 MHz	(as shipped)
1024 x 768	48.10 kHz	60.989 MHz	
800 x 600	18.89 kHz	18.70 MHz	
768 x 768	48.10 kHz	45.742 MHz	
768 x 512	32.06 kHz	28.251 MHz	
512 x 512	32.06 kHz	18.34 MHz	

* Vertical scan (refresh) rate of 56 cycles/sec.
1. MVME390A display resolutions selected by dot clock frequency and MC6845 CRTC controller program
2. CRT horizontal frequencies may vary with monitor selection

CONTROL/STATUS REGISTERS

The module has two control registers that may be programmed to select various operating parameters and modes and a read-only status register that can be used to provide current configuration or other information to an application program. Control registers 1 and 2 are diagrammed in Figures 6 and 7. The status register is diagrammed in Figure 8.

CONTROL REGISTER 1

The states of control register 1 bits and groups of bits may be programmed to select operating conditions and modes. Bit 7 is set to select non-interlaced mode operation. Setting and clearing bit 6 causes the lower half of video memory to be swapped with the upper half and vice versa. This capability can be useful for applications

utilizing low vertical resolution. When bit 5 is cleared, the TTL level vertical synchronization output signal is low true. When bit 4 is cleared, the TTL level horizontal synchronization signal is low true. Bit 3 is used to enable the generation of a VMEbus interrupt by the module. When this bit is cleared, an interrupt is generated on each vertical sync signal. The level of the generated interrupt is set using bits 0, 1, and 2. Control register 1 is accessed at hexadecimal address 280001, relative to the base address of the graphics module.

CONTROL REGISTER 2

The states of control register 2 bits may be programmed to select different operating conditions. The state of bit 7 is controlled to obtain a desired background color. Bits 6, 5, 4 and 3 are reserved for future expansion. Bits 0, 1 and 2 are set to enable the TTL level outputs for plane 0, 1 and 2 bits, respectively, and cleared to hold the TTL outputs for plane 0, 1 and 2 bits to a logic 0 level. Control register 2 is accessed at hexadecimal address 280003 relative to the base address of the graphics module.

STATUS REGISTER

The status register is a one byte register provided for the discretionary use of the system designer. Its four least significant bits may be read (only) from the VMEbus. A header is provided on the module in which jumpers can be placed to establish the states of the four bits as a means of indicating to a program the current configuration of the module or other information, as required. The four most significant bits in the status register are undefined. The status register is accessed at hexadecimal address 2800009 relative to the base address of the module.

VMEbus INTERFACE

The MVME390A Graphics Interface Module has interface circuitry suitable for its role of a slave module on the VMEbus. This includes a D:16 data bus interface and an A:24/A:32 address bus interface. The module includes an interrupter that supports all seven VMEbus interrupt levels. A particular interrupt level is enabled under program control by setting appropriate bits in Control Register 1.

SOFTWARE SUPPORT

Aid for the application programmer is available in the form of an optional package that includes the industry-standard Larson Software Technology CONCEPT/GKS implementation of the ANSI and ISO standard for the Graphical Kernel System (GKS). The package also includes an MVME390A driver under the Motorola SYSTEM V/68 Operating System (Release 2, Version 2.2) or the VERSAdos 4.5 Real-Time Multitasking Operating System plus a library providing binding for FORTRAN (compatible with Absoft FORTRAN 77 for SYSTEM V/68 or VERSAdos) or a library providing binding for C (compatible with Alcyon C for VERSAdos or SYSTEM V/68 compilers).

A hardware-independent interface for 2D graphics output and interactive input, CONCEPT/GKS provides a common syntax for programmer use by standardizing the way functions are accessed thus offering a consistent interface to high-level languages and permitting programs to be transported between different systems.

CONCEPT/GKS contains a library of subroutines used within an application program to produce and manipulate pictures. These may be line graphs, bar charts, medical images, architectural drawings or any other picture that can be described to GKS. The program also contains a supervisory interface between GKS and graphics devices, the Virtual Device Interface (VDI). VDI causes all devices to appear as "identical" virtual devices by requiring use of a standard protocol to isolate the unique physical characteristics of each physical graphics device in the corresponding device driver software module.

CONCEPT/GKS Features include:

- Application Portability
- Selective Viewing using Window and Viewporting
- Supports Raster Functions
- Virtual Device Interface (VDI) Supervisor Eases Development of Device Drivers
- Graphical Output Primitives
  - — Polyline — Set of Connected Lines (defined point sequences)
  - — Polymarker — Centered Symbol at given position
  - — Text — Character String at given position
  - — Polygonal Fill Area: hatch, pattern, or color fill or empty Cell Array — Array of pixels or individual colors (raster images)
  - — GDP — special geometric workstation capabilities such as drawing spline curves, circles or arcs
- Settable Graphical Output Attributes
  - — Polyline line type (solid, dashed), line width, and color
  - — Polymarker type and size
  - — Text size, font, line width, color path and alignment
  - — Fill area pattern, pattern size and color
- Graphical Input Primitives
  - — Locator — return a point in world coordinates
  - — Stroke — provide a sequence of points in world coordinates
  - — Valuator — return a real number
  - — Choice — select from a number of choices
  - — Pick — return a segment name
  - — String — return a character string (i.e. keyboard)
- Inquiry Functions
  - — Return current settings of any output attributes
  - — Return current transform settings
- Metafiles for Storing Graphical Information in Device Independent Form

CONCEPT/GKS SPECIFICATIONS

Level: Oa, conforms to ANSI definitions
Languages supported: Fortran 77 and C
Language: The Fortran GKS library (binding) is written in Fortran 77.
The C GKS library (binding) is written in the C language.
The device drivers are written in the C language.
Memory Usage: The GKS Library requires 40Kb to 80Kb depending on the number of functions used.
Each device driver requires 20Kb to 30Kb.

Note: Exact requirements will depend on the CPU and compiler used. The above figures are based on an MC68010 CPU and compiler.

7	6	5	4	3	2	1	0
INTRLAC*	ALTPAGE*	VSI*	HSI*	ISNBL*	IS3	IS2	IS1

INTRLAC* — Logic 0 indicates interlace mode. Logic 1 indicates non-interlace mode or interlace and low resolution.

ALTPAGE* — Logic 0 swaps the top and the bottom halves of the display memory. This bit is useful for low vertical resolution (512 or fewer lines).

VSI* — Vertical Sync Invert. This bit controls the polarity of the TTL vertical sync signal. Vertical sync output is low true when this bit is 0.

HSI* — Horizontal Sync Invert. This bit controls the polarity of the TTL horizontal sync signal. Horizontal sync output is low true when this bit is 0.

ISNBL* — Interrupt Enable. A logic 0 enables the board to generate a VMEbus interrupt every time vertical sync occurs. A logic 1 disables the interrupter.

IS3–IS1 — Interrupt level select. Interrupt level may be programmed as follows:

IS3	IS2	IS1	Interrupt Level
0	0	0	No interrupt
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Figure 6. Control Register 1

7	6	5	4	3	2	1	0
ALTMAP	[*]	[*]	[*]	[*]	TTLEN2	TTLEN1	TTLEN0

ALTMAP — This bit may be jumpered to control 0,1 or 2 MSB bits of the 4-bit value of each of the three colors.

TTLEN2 — When set, enables TTL outputs for plane 2. When cleared, holds the TTL outputs for plane 2 to a logic 0.

TTLEN1 — When set, enables TTL outputs for plane 1. When cleared, holds the TTL outputs for plane 1 to a logic 0.

TTLEN0 — When set, enables TTL outputs for plane 0. When cleared, holds the TTL outputs for plane 0 to a logic 0.

[*] — Reserved bits.

Figure 7. Control Register 2

7	6	5	4	3	2	1	0
*	*	*	*	ST3	ST2	ST1	ST0

— Status Register is a 4-bit wide, read-only register.

— Only four least significant bits are valid when read. The most significant bits are undefined.

— Status bits are at logic 1 when jumper is removed and at logic 0 when jumper is installed.

— None of the status bits are assigned by hardware, thus, user may define them as necessary.

Figure 8. Status Register

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements	+5 Vdc +5%, 6 A (typ)
Environmental Limits	
Operating Temperature	0°C to +55°C
Storage Temperature	−40°C to +85°C
Relative Humidity	5% to 95%, (non-condensing)
Form Factor	Double High Eurocard with front panel
Board Dimensions	
Height	9.25 in. (235 mm)
Depth	6.3 in. (160 mm)
Front Panel Dimensions	
Height	10.3 in. (262 mm)
Width	0.79 in. (20 mm)

ORDERING INFORMATION

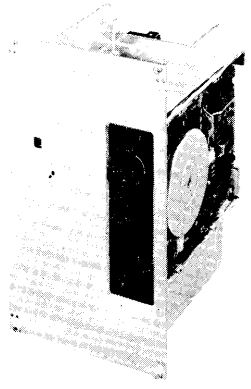
Part Number	Description
MVME390A	Graphics Display Interface VMEmodule with 384Kb Video RAM and D:16 A:24/A:32 VMEbus Slave Interface. Includes User's Manual.
MVME390A/D	MVME390A Display Interface VMEmodule User's Manual
M68NNXBGKS390C	CONCEPT/GKS Level Oa Object Code Package including: GKS Library and VDI Supervisor Alcyon C Compiler Binding Driver under the SYSTEM V/68 Release 2, Version 2.2 Operating System Includes 90 Day Hot Line Support from: A.T. Barrett & Associates 11501 Chimney Rock Houston, TX 77035 (713) 728-9688 Includes User's Manual

NOTE: Source code and redistribution license available from A.T. Barrett & Associates

Part Number	Description
M68NNXBGKS390F	Same as M68NNXBGKS390C except with Absoft FORTRAN 77 Language Binding
M68VKXBGKS390C	Same as M68NNXBGKS390C except with Driver under the VERSAdos 4.5 Real-Time Operating System
M68VKXBGKS390F	Same as M68NNXBGKS390F except with Driver under the VERSAdos 4.5 Real-Time Operating System

MVME820
MVME821
MVME822

VMEmodule™ VMEbus Plug-In Mass Storage Modules



STANDARD FEATURES

- All modules consist of standard 5-1/4" mass storage units (Winchester drives, floppy drives, etc.) mounted in an enclosure which plugs into a VME943 chassis
- All units include the MVME702 Disk Interface Module and necessary cabling for simple interface to VMEbus via the MVME320 Disk Controller

MVME820

- 15Mb (formatted) 5-1/4" Winchester Disk Drive (ST506 Interface)
- 650Kb (formatted) 5-1/4" double-sided, double density, 96 TPI, floppy disk drive

MVME821

- Two 650Kb (formatted) 5-1/4" double-sided, double-density, 96 TPI, floppy disk drives

MVME822

- 40Mb (formatted) 5-1/4" Winchester disk drive (ST506 Interface)
- 650Kb (formatted) 5-1/4" double-sided, double density, 96 TPI, floppy disk drive

Add mass storage capacity to a MVME943-based VMEbus system using any of three new mass storage modules: MVME820, MVME821 or MVME822. Each is supplied as a complete enclosed unit ready to be mounted in the right front side of the MVME943 Chassis by means of top and bottom rails that slide into four heavy-duty card guides provided in the chassis for this purpose. Four captive screws on the VME-type double-high front panel retain the storage unit.

Disk Drive Assembly MVME820 contains a 15Mb (formatted) 5-1/4" Winchester disk drive and a 650Kb (formatted) 5-1/4" double-sided, 96 TPI, floppy disk drive.

Disk Drive Assembly MVME821 contains two 5-1/4" double-sided, double-density, 96 TPI, floppy disk drives.

Disk Drive Assembly MVME822 contains a 40Mb (formatted) 5-1/4" Winchester disk drive and a 650Kb (formatted) 5-1/4" double-sided, double-density, 96 TPI, floppy disk drive.

A mass storage unit (less interface module) comprises a physical drive assembly and a PCB assembly which provides an industry-standard interface allowing up to four 5-1/4" drive units to be connected to a single formatter/controller. Maintenance is simplified by all hard disk electronic components being located outside the head/disk assembly.

Supplied with each mass storage module is a double-high disk interface module designed for mounting anywhere in the rear of the MVME943 chassis. The interface module greatly simplifies the task of connecting the MVME320 Disk Controller to a mass storage module and also facilitates the addition of other hard and floppy disks. When additional storage is added, standard 34-pin and 50-pin cables can be used.

WINCHESTER HARD DISKS

The CM5619 hard disk supplied with the MVME820 Plug-In Storage Unit was chosen from the CM5000 Series of 5-1/4" Winchester Disk Drives manufactured by Computer Memories Inc. of Chatsworth, CA. This series utilizes a Winchester-type head/media technology similar to that of IBM-3350 type disk drives.

The CM5619 disk drive utilizes three platters having two data surfaces apiece. Each data surface has one movable head to service the 306 data tracks. Both the dc operating voltages required by the CM5619 and its outline dimensions are consistent with those of industry-standard minifloppy disk drives.

Mechanical and contamination protection for the CM5619 heads, actuator and disks are provided by an impact resistant aluminum enclosure. A self contained recirculating system supplies clean air through a 0.3 micron filter. A breather filter equalizes internal/ambient air pressures without chance of contamination.

The Model 1304 hard disk supplied with the MVME822 Plug-In Storage Unit was chosen from the 1300 Series of 5-1/4" Winchester Disk Drives manufactured by the Micropolis Corporation of Chatsworth, CA.

The 1304 disk drive utilizes four disks and six data surfaces having a total of 4980 physical tracks. When MFM recording format is used, 8192 bytes per track or 32 sectors per track (256-byte sectors) are obtained. Each of the six movable heads services 830 data tracks.

A "chassis-within-a-chassis" design utilizes a die cast drive assembly suspended within a die cast outer frame to provide maximum isolation against mechanical vibration and shock. An active air filtration system utilizes a 0.3 micron filter and another 0.3 micron filter insures contamination-free equalization of pressures between the clean area and the external environment.

HARD DISK DRIVE FEATURES

- Storage Capacities:
  — MVME820 — 19.14Mb unformatted, 15Mb formatted
  — MVME822 — 51.90Mb unformatted, 40Mb formatted
- Reliable Winchester Design
- Standard Minifloppy Outline/mounting
- Transfer Rate — 5Mbits/second
- Brushless DC Spindle Motor
- Electronic Damping

FLOPPY DISK DRIVE

The floppy disk supplied with the MVME820, MVME821, and MVME822 Plug-In Mass Storage Modules was selected from the FD-55 series of LSI mini-flexible disk drives manufactured by The TEAC Corporation of America of Montebello, CA. The model chosen: FD-55F, is a 5-1/4", soft sector, FM/MFM version of industry standard outline and half height — 1.63".

FLOPPY DISK DRIVE FEATURES

- Storage Capacity:
  — 650Kb, Formatted, MFM Mode
- Data Transfer Rate — 250Kb/s (MFM)
- Average Access Time — 94 ms
- Track Access Time — 3 ms
- Settling Time — 15 ms
- Tracks Per Disk — 160
- Track Density — 96 Tracks Per Inch

CONTROLLER — DISK DRIVE — INTERFACE MODULE CONNECTIONS

MVME820/MVME822 CABLING

Hard disk data signals are conducted between MVME320 Disk Controller jack J2 and jack J2 on the MVME820 or MVME822 Hard Disk Drive via the provided 20-line flat ribbon cable. The provided Y-cable, which has two 50-pin connectors and lines 1 through 34 brought out to a 34-pin connector, is used for conducting control and data signals between the disk controller, the 5-1/4" hard disk drive and the disk interface module for connection (by separate cable) to the storage assembly's floppy disk and (optionally) to additional floppy or hard disk drives. One 50-pin end of the Y-cable is connected to jack J2 on the MVME320 Disk Controller and the 34-pin connector to J1 on MVME820 or MVME822. The other 50-pin end is connected to J9 on the interface module. Control and data signals are sent to the 5-1/4" floppy disk drive by connecting the provided 34-conductor cable between jack J10 on the interface module and jack J3 on the disk drive.

Facilities for connecting additional mass storage to the MVME320 Disk Controller are also provided by the interface module. Additional 5-1/4" or 8" hard disks and 5-1/4" or 8" floppy disks are connected to the interface module using standard 20-, 34- and 50-conductor cables, as required. Should additional storage beyond that provided by the MVME820 or MVME822 Plug-In Modules not be required or should the user not use the interface module, a custom 50-conductor cable can be fabricated for connecting jack J3 on the disk controller to jacks J1 and J3 on the hard and floppy disk drives, respectively.

MVME821 CABLING

Connections between the disk controller and the dual 5-1/4" floppies of the MVME821 Plug-In Module are made using the supplied 50-conductor cable which is connected between J3 on the controller and J9 on the interface module. To complete transmission to the floppy disk drives of data and control signals which reach the interface module via the 50-conductor cable, the supplied 34-conductor Y-cable is connected between jacks J10 on the interface module and J3 on each floppy drive.

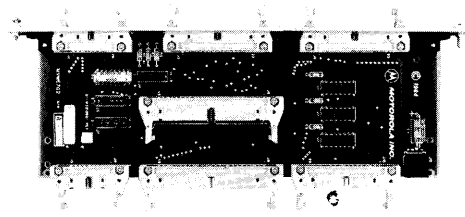
For the MVME821 as well as the MVME820 and MVME822 Plug-In Modules, facilities for connecting additional mass storage to the MVME320 Disk Controller are provided by the interface module. Additional 5-1/4" or 8" hard disks and 5-1/4" or 8" floppy disks can be connected to the interface module using standard 20-, 34- and 50-conductor cables, as required. Should additional storage beyond that provided by the MVME821 Plug-In Module not be required or should the user not use the interface module, a custom 50-conductor cable can be fabricated for connecting jack J3 on the disk controller to the J3 jacks on the two floppy disk drives.

DISK INTERFACE MODULE

Designed for mounting anywhere along the back of the MVME943 Chassis (or standalone use), the MVME702 Disk Interface Module supplied with each of the three storage modules extends the functions of the MVME320 Disk Controller by:

1. Organizing the controller's data and control output signals into the conventional groups required for interfacing hard and floppy disk drives;
2. Buffering and terminating the 5-1/4" hard disk control signals output from the controller;
3. Providing jumper headers which permit configuring for the quantity and types of hard and floppy disk drives required for an application.

All capabilities of the MVME320 to control various combinations of 5-1/4" and 8" hard and floppy disk drives are supported by the MVME702 Disk Interface Module.



PLUG-IN STORAGE MODULE SPECIFICATIONS

Characteristics	Specifications
Power Requirements	
Voltage	12 Vdc +5%
Current at 12 Vdc	
MVME820	1.93 A (typ) 4.4 A (max)
MVME821	0.85 A (typ) 1.8 A (max)
MVME822	2.30 A (typ) 3.7 A (max)
Voltage	5 Vdc +5%
Current at 5 Vdc	
MVME820	1.34 A (typ) 1.6 A (max)
MVME821	0.88 A (typ) 1.2 A (max)
MVME822	0.90 A (typ) 1.5 A (max)
MVME702	0.38 A (typ) 0.48 A (max)
Environmental Limits	
Operating Temperature	0°C to +55°C
Storage Temperature	−40°C to +85°C
Humidity Range	5 to 90%, non-condensing
Mechanical Specifications	
Height x Width	
(front panel)	10.3" x 6" (262mm x 152mm)
(drive package)	9.6" x 5.4" (244mm x 137mm)
Depth	8.2" (208mm)
Height x Depth	
MVME702	9.2" x 3.2" (234mm x 81mm)
Thickness	
MVME702	0.5" (12.7mm)
Data Rates	
Winchester	5Mbits/second
Floppy	250Kbits/second

ORDERING INFORMATION

Ordering Number	Description
MVME820	Plug-In Mass Storage Unit — 15Mb Hard Disk Drive and MVME702 Disk Drive Interface Module assembly. Installs in MVME943 Chassis. Includes User's Manuals.
MVME821	Plug-In Mass Storage Unit — Two 650Kb Double Sided, Double Density, Half Height Floppy Disk Drives and MVME702 Disk Drive Interface Module assembly. Installs in MVME943 Chassis. Includes User's Manuals.
MVME822	Plug-In Mass Storage Unit — 40Mb Hard Disk Drive and MVME702 Disk Drive Interface module assembly. Installs in MVME943 Chassis. Includes User's Manuals.
MVME820/D	Plug-In Mass Storage Unit User's Manual.
MVME702/D	Disk Interface Module User's Manual.

RELATED DOCUMENTATION

MVME320/D	VMEbus Disk Controller Module User's Manual.
M68KVSWD1/D	VME/10 Microcomputer System Computer Memories Model CM5000 Series Winchester Disk Drive User's Manual.
M68KVSFD2/D	TEAC Model FD-55 Series Floppy Disk Drive User's Manual.
MVME943/D	VMEmodule Chassis User's Manual

MVME900 Series

VMEmodule—I/Omodule Card Cages, Chassis, Power Supplies, Backplanes, and Accessories

The MVME900 Series of packaging, prototyping and power accessories facilitates the use of VMEmodules and I/Omodules in VMEbus-based microcomputer systems meeting specific performance and installation requirements. The MVME900 Series includes the following products:

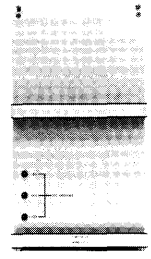
- Plug-In Power Supply
- 20-Slot VMEbus Backplane
- 9-Slot VMEbus Backplane
- 5-Slot I/O Channel Backplane
- Double High Extender Board
- Double High Wirewrap Board
- Single High Extender Board
- Single High Wirewrap Board
- Remote I/O Channel Connector Board
- Chassis with Power Supply for VMEbus and I/O Channel Modules
- Chassis for VMEmodules and I/O Channel Modules
- Chassis for VMEmodules

VMEmodule 200 Watt, Plug-In Power Supply

The MVME910-3 Power Supply provides sufficient power for a MVME940-1 Chassis filled with VMEmodules (9) and I/Omodules (10). It is a plug-in unit of switching design and occupies a space in the chassis equivalent to five double high VMEmodules. The MVME910-3 operates from 90 to 270 volt line supplies at 45 to 64 Hz and is designed to meet the requirements of UL, CSA and VDE safety standards.

MVME910-3 Features:

- +5 Vdc @ 30 A
- +12 Vdc @ 3 A
- -12 Vdc @ 1 A
- Overload Protection
- Remote Sensing for Accurate Board Voltage
- Short Circuit Protection



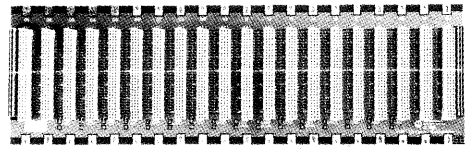
MVME910-3

VMEbus Backplanes

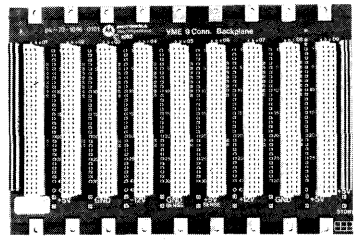
The MVME920 and MVME921 VMEbus Backplanes are used to interconnect VMEmodules. They are fully assembled, with connectors and terminators. The MVME920 has 20 connectors (a full 19" chassis width), and the MVME921 has 9 connectors.

MVME920 and MVME921 Features:

- Operation up to 20 MHz
- All Signal and Power Line Connections Provided
- DIN-41612C 96-Pin Connectors
- "Fast-On" Power and Ground Terminals
- Bus Termination Networks on Both Ends
- Test Connector for Bus Signal Access
- Jumper Areas Provided for Daisy-Chain Lines



MVME920



MVME921

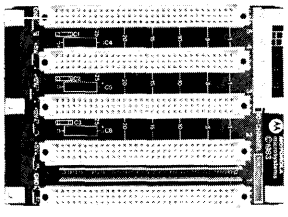
2

I/O Channel Backplane

The MVME922 I/O Channel Backplane is used to interconnect up to five I/Omodules. It is fully assembled, with connectors and terminators.

MVME922 Features:

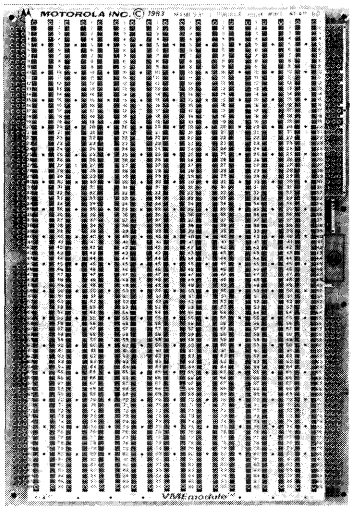
- All Signal and Power Line Connections Provided
- DIN-41612C 64-Pin Connectors
- "Fast-On" Power and Ground Terminals
- Termination for all I/O Channel Signal Lines except INT-4 and XACK*



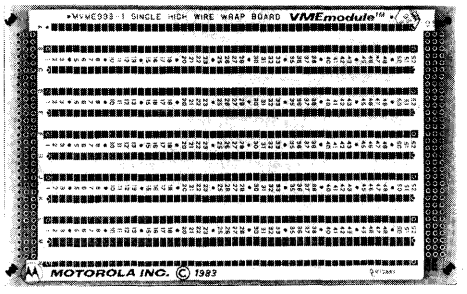
. MVME922

VMEbus and I/O Channel Wirewrap Boards

The MVME931-1 and MVME933-1 Wirewrap Boards are used for developing custom I/O and interface modules for VMEbus systems. MVME931-1 is a double high Eurocard board for interfacing the VMEbus. MVME933-1 is a single high board for interfacing the I/O Channel. DIN connectors are included.



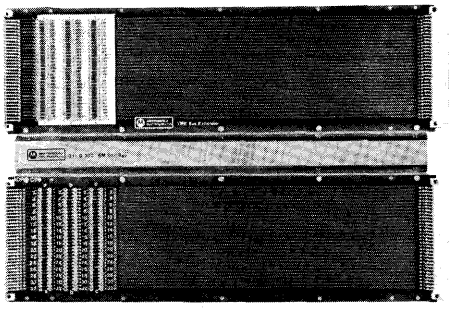
MVME931-1



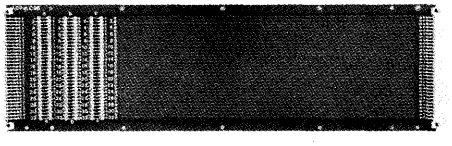
MVME933-1

VMEbus Extender Boards

The MVME930 and MVME932 Extender Boards facilitate the testing and debugging of system modules by providing front panel DIN connector access to the VMEbus. MVME930 is used with double high Eurocard modules. MVME932 is used with single high Eurocard modules. Access to all 96 VMEbus data, address, control and power lines is provided at the front panel connector. MVME932 can also be used to connect single high modules to the I/O Channel.



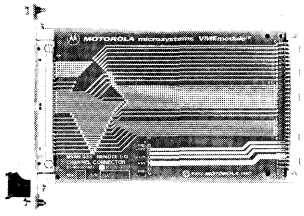
MVME930



MVME932

Remote I/O Connector Board

The MVME935 Remote I/O Connector Board brings the I/O Channel to a 50-pin connector at the front panel. It is used with remote modules which are connected to the I/O Channel via a ribbon cable, e.g., the M68RAD1 Remote Intelligent Analog-To-Digital Conversion Module. Access to all I/O Channel data, address and control lines is provided at the front panel connector. Access to power lines is not provided.



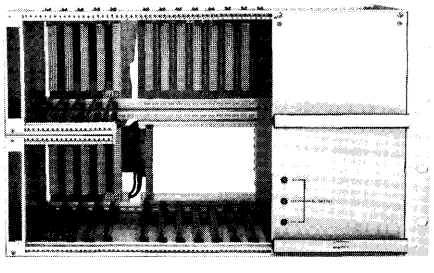
MVME935

Chassis with Power Supply for VMEbus and I/O Channel

The MVME940-1 is a fully assembled chassis complete with a power supply which supports a mixture of VMEmodules and I/Omodules. This is a fully tested system with the hardware and cables to support a maximum of seven VMEmodules and ten I/Omodules. It is suitable for application development and prototyping and for use as the basis of a complete end product.

MVME940-1 Features:

- 19" Rack Mounting Chassis
- Card Guides and Hardware to Support 7 Double-High Eurocard and 10 Single-High Eurocard Modules
- One MVME921 9-Slot VMEbus Backplane
- Two MVME922 5-Slot I/O Channel Backplanes
- An MVME910-3 200-Watt Power Supply
- DC Wiring Harness
- I/O Channel Signal Cables



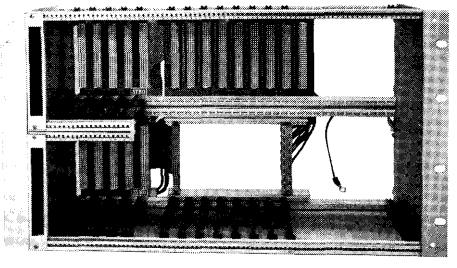
MVME940-1

CHASSIS FOR VMEbus AND I/O CHANNEL

The MVME941 is a fully-assembled rack mounting chassis which supports a mixture of VMEmodules and I/Omodules. Except for cables and a power supply which the user must supply, MVME941 has all the required hardware, including backplanes and connectors.

MVME941 Features:

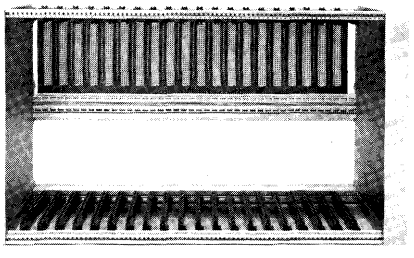
- 19" Rack Mounting Chassis
- Card Guides and Hardware to Support 9 Double-High and 10 Single-High Eurocard Modules
- One MVME921 9-Slot VMEbus Backplane
- Two MVME922 5-Slot I/O Channel Backplanes



MVME941

CHASSIS FOR VMEbus

The MVME942 is a fully assembled rack mounting unit which supports only double-high Eurocard VMEmodules. It has the hardware required, including backplanes and connectors, to support 20 VMEbus modules.



MVME942

Ordering Information

Power Supply

Part Number	Description
MVME910-3	200 Watt Plug-In Chassis Power Supply, 110 Vac

Backplanes and Accessories

MVME920	20-Slot VMEbus Backplane
MVME921	9-Slot VMEbus Backplane
MVME922	5-Slot I/O Channel Backplane
MVME930	VMEbus (Double high) Extender Board
MVME931-1	VMEbus (Double high) Wirewrap Board
MVME932	I/O Channel (Single high) Extender Board
MVME933-1	I/O Channel (Single high) Wirewrap Board
MVME935	Remote I/O Connector Board

Assembled Chassis

MVME940-1	19 in. wide card rack with handles, 9-slot VMEbus backplane, two 5-slot I/O Channel backplanes, single size module conversion hardware, plug-in power supply. Includes User's Manual
MVME941	19 in. wide card rack with a single 9-slot VMEbus backplane, two 5-slot I/O Channel backplanes, and single size modules conversion hardware. Includes User's Manual
MVME942	19 in. wide card rack with 20-slot VMEbus backplane. Includes User's Manual

Related Documentation

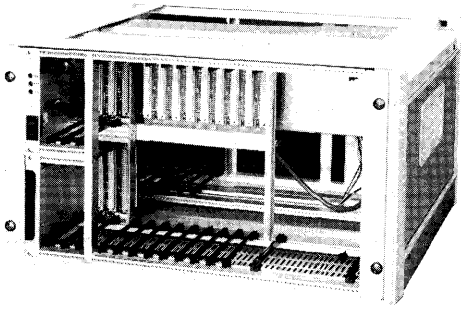
HB212/D	VMEbus Specification Manual
M68RIOCS/D	Input/Output Channel Specification Manual
MVME900/D	MVME900 Series Equipment User's Manual

VMEbus Chassis Assembly

MVME943-1
MVME943-2
MVME944-1
MVME944-2

STANDARD FEATURES

- 19-inch Rack-mountable Chassis
- AC Filter/breaker Module
- DC Voltage Indicators (LEDs)
- AC/DC Wiring Harnesses
- DC Power Terminal Strip
- Two VMEbus Backplane Terminating Modules
- 400-watt Power Supply
  - — Short circuit protection
  - — Over-voltage/Over-temperature protection
  - — Remote sensing for accurate module voltage
  - — Power fail logic for VMEbus
  - — AC fail sensing
  - — 115 Vac or 230 Vac input operation



MVME943 — SPECIFIC FEATURES

- I/O Channel Signal Cable
- Provision for User-supplied Mass Storage Unit
- MVME921A 9-slot VMEbus Backplane
  - — Operates at up to 30 MHz
  - — Provides all signal/power line connections
  - — Faston power, sense, and ground connections
  - — Jumper areas for daisy-chain lines
  - — Bus termination with active negative spike suppression
  - — DIN 41612C 96-pin sockets for VMEmodules
- Two MVME924 3-slot I/O Channel Backplanes (6 slots total)
  - — Provides all signal/power line connections
  - — Faston power and ground connections
  - — Bus termination, using 330/470-ohm SIP resistor packs
  - — DIN 41612C 96-pin sockets for I/Omodules

MVME944 — SPECIFIC FEATURES

- MVME920A 20-slot VMEbus Backplane
  - — Operates at up to 30 MHz
  - — Provides all signal/power line connections
  - — Faston power, sense, and ground connections
  - — Jumper areas for daisy-chain lines
  - — Bus termination with active negative spike suppression
  - — DIN 41612C sockets for VMEmodules

The MVME943 and MVME944 VMEbus Chassis Assemblies are designed to convert individual VMEmodules into complete, compact, self-contained microcomputer systems. These chassis allow the user to efficiently meet a variety of stringent microcomputer requirements. The MVME943 and MVME944 permit the convenient assembly of custom features including table-top or rack-mounted configurations.

MVME943 VMEbus CHASSIS ASSEMBLY

The front of the MVME943 chassis assembly houses Eurocard card cage with an MVME921A nine-slot backplane suited for VMEmodules and two MVME924 three-slot backplanes configured for I/Omodules. The MVME943 features a front panel with an opening on the right-front side for an MVME820/MVME821/MVME822 mass storage unit (user-supplied option). A power switch module located on the left-front side of chassis consists of an ac power switch, a printed wiring board, dc power connector, and three green LEDs (dc power indicators). These components are attached to a single-high VME plate which is screwed to the chassis in the same manner as a VMEmodule.

The rear of the chassis supports standard double-high Eurocard panels and 80mm modules such as Motorola's MVME7xx series modules. This area can also be used for mounting additional Eurocard rails which support single-

MVME943 ASSEMBLY (Cont.)

high I/O Channel backplanes and modules. A total of sixteen modules can be installed in the rear of the chassis. An ac circuit breaker is also located in the rear of the chassis.

An internal 400-watt power supply is mounted in a fan-cooled air duct assembly located in the upper half of the chassis, slightly toward the rear. The fan inside the air duct provides cooling for the power supply only. Additional cooling (if necessary) for VMEmodules and I/O-modules must be supplied externally by the user.

MVME921A NINE-SLOT VMEbus BACKPLANE

The MVME943 chassis contains one MVME921A nine-slot VMEbus backplane mounted near the center-front region of the chassis, adjacent to the I/O Channel backplanes and to the left of the optional mass storage area. The MVME921A uses all signal and power lines for bus connector P1 of the VMEbus. This backplane is designed for bus transfer speeds up to 30 MHz. Connector slots 01 and 09 (rear side) are designed to receive plug-in terminator modules.

MVME924 THREE-SLOT CHANNEL BACKPLANES

Two MVME924 three-slot I/O Channel backplanes are mounted, one above the other, on the left-front of the MVME943 chassis, adjacent to the MVME921A backplane. These backplanes are connected together via a 50-wire ribbon cable. A connector on one end of the cable connects to a VMEbus or an I/O Channel interface module. Terminating resistors are mounted between sockets 01 and 02 on each backplane.

MVME944 VMEbus CHASSIS ASSEMBLY

The MVME944 VMEbus chassis assembly houses a front-mounted Eurocard card cage equipped with an MVME920A 20-slot backplane. The front of the chassis also contains a power switch module located on the extreme left-side of the unit. The main purpose of this module is to provide convenient access to the ac power switch. In addition to this switch, the power switch module includes a printed wiring board, dc power connector, and three green LEDs (dc power indicators). These components are attached to a half-height VME front plate which is screwed to the chassis in the same manner as a VMEmodule. The rear of the chassis, including the power supply/air duct assembly, is identical to that of the MVME943 chassis.

MVME920A 20-SLOT VMEbus BACKPLANE

The MVME920A 20-slot VMEbus backplane extends across the entire width of the MVME944 chassis. The backplane uses all signal and power lines for bus connector P1 of the VMEbus. The MVME920A is designed for bus transfer speeds of up to 30 MHz. Connector slots 01 and 20 (rear side) are capable of receiving plug-in terminator modules.

MVME943/MVME944 — MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Dimensions	
Width (with front flange)	19.00 in. (482.6mm)
(without front flange)	17.20 in. (436.8mm)
Height	10.50 in. (266.7mm)
Depth	19.75 in. (501.7mm)
Weight (without disk drives, VMEmodules, or I/Omodules)	
MVME943	20.75 lbs. (9.4 Kg)
MVME944	19.80 lbs. (9.0 Kg)
Temperature*	
Operating	+5°C to 40°C
Storage	−30°C to +60°C
Relative humidity*	20% to 80% (non-condensing)
Cooling	Power supply requires 60 CFM forced air cooling. Single fan provided for power supply cooling only.
Input Voltage	
MVME943-1, MVME944-1	115 Vac
MVME943-2, MVME944-2	230 Vac
AC Input Frequency	47 Hz to 63 Hz

*These are generic limitations imposed by the MVME83X Mass Storage Modules. Broader limits are possible if peripherals are not mounted in the MVME943/MVME944, or if more tolerant peripherals are selected, or if more powerful fans are used.

ORDERING INFORMATION

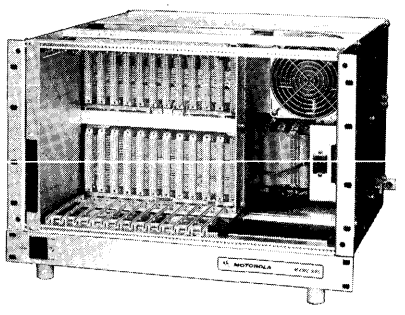
Part Number	Description
MVME943-1	VME Chassis Assembly with MVME921A VMEbus 9-slot backplane, two MVME924 3-slot I/O Channel backplanes, 400-watt power supply, and provision for mass storage unit. Operates at 115 Vac input voltage. Includes MVME943/D user's manual.
MVME943-2	Same as MVME943-1, but operates at 230 Vac.
MVME944-1	VME Chassis Assembly with MVME920A 20-slot VMEbus backplane and 400-watt power supply. Operates at 115 Vac input voltage. Includes MVME944/D user's manual.
MVME944-2	Same as MVME944-1, but operates at 230 Vac.

RELATED DOCUMENTATION

MVME920A/D	MVME920A 20-slot VMEbus Backplane User's Manual
MVME921A/D	MVME921A 9-slot VMEbus Backplane User's Manual
MVME924/D	MVME924 3-slot I/O Channel Backplane User's Manual
MVME943/D	MVME943 Chassis User's Manual
MVME944/D	MVME944 Chassis User's Manual
HB212/D	VMEbus Specification Manual

VMEbus Chassis Assembly, Mass Storage Modules and Accessories

MVME945-1
MVME945-2
MVME833
MVME834
ACCESSORIES



MVME945 CHASSIS FEATURES

- Internationally Approved (UL, CSA, VDE) 400 Watt Power Supply
  - — Jumper-selectable 115/230 Vac input
  - — Short circuit protection
  - — Over-voltage protection
  - — Remote sensing for accurate module voltage
- 12-Slot, 32-Bit, VMEbus Compatible Backplane
  - — Provides all signal/power line connections for full 32-bit operation
  - — Jumper areas for daisy-chain signals
  - — Bus terminators provided
- 19-inch Rack-mountable Chassis
- Forced Air Cooling for Power Supply, Mass Storage Modules and VMEmodules
- Rear Mounted Card Cage for Up To 21–700 Series Transition Modules
- ESD Hardening and Electromagnetic Interference (EMI) Shielding (FCC and VDE Compliant)
- Meets UL, VDE and CSA Safety Requirements

MVME833 FEATURES

- 70Mb (formatted) 5-1/4" Full-High Winchester Disk Drive, ST506 Compatible
- 655Kb (formatted) 5-1/4" Double-Sided, Double-Density, Half-High Floppy Disk Drive, SA400 Compatible

MVME834 FEATURES

- 70Mb (formatted) 5-1/4" Full-High Winchester Disk Drive, ST506 Compatible
- 655Kb (formatted) 5-1/4" Double-Sided, Double-Density, Half-High Floppy Disk Drive, SA400 Compatible
- QIC-02 Compatible Streaming Tape Drive. Accepts ANSI Standard X3.55-1982 1/4" Tape Cartridges

ACCESSORIES

- MVME945CNVKT — 115 Vac to 230 Vac Conversion Kit
- MVME945I/O — I/O Channel Card Cage Kit
- MVME945CVR — Desk Top Cover for MVME945
- MVME83XKT — Mass Storage Expansion Kit
- MVME83XSTKT — Streaming Tape Expansion Kit

MVME945 CHASSIS ASSEMBLY

The MVME945 chassis is designed to help convert individual VMEmodule board-level products into a complete, compact, self-contained microcomputer system. This packaging design helps to best meet the user's specific performance and installation requirements. Various options permit the convenient assembly of systems for custom installations/applications.

The following paragraphs present an overview of the MVME945 chassis and its major assemblies and/or components.

CHASSIS

The MVME945 chassis is 19" wide, 19-3/4" deep, and 12-1/4" high. It houses a VMEmodule card cage with 12 VMEbus slots. It has a front panel opening for the MVME83x series of mass storage modules. An ON/OFF power switch is located on the lower, left front and a fuse assembly is located on the lower, left rear. Power is supplied by an internal 400 watt power supply, which is mounted in the upper half, and toward the rear of the chassis. There are provisions for 21 rear panel mounted 80 mm interface modules (MVME7xx).

Three dc-powered fans provide forced air cooling for the power supply, mass storage enclosure and VMEmodules. Convection cooling is used for the MVME7xx series modules.

Double-high 80 mm modules such as Motorola's MVME7xx series modules can be mounted in the rear of the chassis, although no power or support is provided. Alternatively, additional Eurocard chassis rails and backplanes for single-high I/O Channel modules can be installed in this area. Nine I/O modules can be used in the MVME945 chassis when the optional I/O Channel kit (MVME945I/O) is installed.

The MVME945 specifications are shown in Table 1.

Table 1. MVME945 Chassis Specifications

Characteristics	Specifications
Dimensions Width (with front flange) (without front flange) Depth Height	 19 in. (482.6 mm) 17.2 in. (436.8 mm) 19.75 in. (501.7 mm) 12.2 in. (266.7 mm)
Weight	TBD
Input Voltage	90 Vac to 132 Vac or 180 Vac to 264 Vac
Input Frequency	47 Hz to 63 Hz
Input Power	575 VA (max)
Output Voltages	+5 Vdc @ 50 A (max) +12 Vdc @ 10 A (max) −12 Vdc @ 5 A (max)
Output Power	400 Watts (max)
Temperature* Operating Storage	 +5°C to +40°C −30°C to +60°C
Relative Humidity*	20% to 80% (non-condensing)
Altitude	10,000 ft.

*These are generic limitations imposed by the MVME83X Mass Storage Modules. Broader limits are possible if peripherals are not mounted in the MVME945, or if more tolerant peripherals are selected, or if more powerful fans are used.

POWER SUPPLY

DC power for the backplane, optional disk drives and all VMEmodules is supplied by a UL, CSA, VDE certified 400 Watt power supply. This supply is easily converted from 115 Vac to 230 Vac operation by installing the MVME945CNVKT (115 Vac to 230 Vac) conversion kit, and provides overvoltage protection for the +5 Vdc output. This power supply is mounted in the fan-cooled air duct assembly, located in the upper half of the chassis behind the backplane for ease of servicing. The air duct also provides EMI/RFI shielding for the power supply.

Several different power supplies may be qualified for use in the MVME945 chassis, each of which may have slightly different specifications. For this reason, the power supply specifications shown in Table 2 should be considered approximate, with the possibility of some variation.

VMEbus BACKPLANE

The MVME945 chassis contains one 12 slot, full 32-bit VMEbus backplane mounted to the left of the optional mass storage area. This backplane contains 12 connectors (sockets) on the top row to receive connector P1 of the VMEbus and another 12 connectors on the bottom row to receive P2 of the VMEbus. All rows of P1 (A, B, C) are interconnected across all 12 slots providing the required signals for A24/D16 operation and the center row (B) of P2 is interconnected across all 12 slots providing the required signals for extended operation (A32/D32). Rows A and C of P2 are not interconnected, but are left available to the user for I/O or expansion bus operation.

To manage signal reflections and provide noise margin, all signal lines not daisy-chained are terminated with 470 Ω (pull-down) and 330 Ω (pull-up) resistors. These resistors are arranged in networks and are permanently installed on the rear outer edges of the backplane board adjacent to the P1 and P2 connectors.

The terminators are also intended to provide "active negative spike suppression" for the SYSCLK and SERCLK signal lines.

The +5 Vdc power lines (input from power supply) are connected to the backplane through ring lugs and #10 studs at the center of the backplane. The +12 Vdc, −12 Vdc and +5 Vdc sense power lines are connected to the backplane through a 5-position "mate-n-lok" connector. These connectors are rated at 60 A for the +5 Vdc lines and 10 A for the +12 Vdc and −12 Vdc

Table 2a. Power Supply Specifications

Characteristics	Specifications
Output Ripple and Noise Ripple Noise	 1% peak-to-peak or 100 mV, whichever is greater 0.2% V_{rms}
Holdover Storage	16 ms/min at Vac = 115 and rated power
Overload Protection	All outputs are protected against damage from overloads and shorts.
Rated Output Power	400 Watts (max)
Output Ratings Minimum Load Operation	 5 A on the +5 Vdc Output

Table 2b. Power Supply Output

Output Number	DC Voltage	Current Amp(s)	Load Reg. (20% to 100% change)	Line Reg. (Low Vac to High Vac)	Cross Reg. (35% to 85% Change on Output #1)
1	+5	50	+1%	+0.2%	+1%
2	+12	10	Must stay within +5% for any combination of line, load, cross regulation and centering.		
3	−12	5			

lines. A 2-position "mate-n-lok" connector provides a +5 Vdc standby power source to the backplane. This connector is rated at 2 A.

Four 5-position "mate-n-lok" "output" connectors are provided for distributing dc power to peripheral devices, such as disk drives and fans. They provide distribution for +5 Vdc, +12 Vdc and −12 Vdc and are rated at 5 A. Also, a 4-position connector provides power for dc status indicator lamps. This connector distributes +5 Vdc, +12 Vdc and −12 Vdc at 100 mA.

The backplane specifications are shown in Table 3.

Table 3. VMEbus Backplane Specifications

Characteristics	Specifications
Dimensions Width Height	 10.10 in. (256.54 mm) 10.31 in. (261.87 mm)
Power Consumption	6 Watts (1.2 A @ +5 Vdc) (Required by terminator networks)
VMEbus Connectors	24 sockets: DIN 41612, Type C, 96-pin Socket spacing (center-to-center) = 0.8 inches
Power Connectors	INPUT Voltage / Current / Connector +5 Vdc / 60 A / #10 studs +5 Vdc Sense / — / mate-n-lok 5-position +12 Vdc / 10 A −12 Vdc / 10 A +5 Vdc Stby / 2 A / mate-n-lok 2-position OUTPUT Voltage / Current / Connector +5 Vdc / 5 A / mate-n-lok 5-position +12 Vdc / 5 A −12 Vdc / 5 A +5 Vdc / 100 mA / Amp +12 Vdc / 100 mA −12 Vdc / 100 mA / 4-position
VMEbus Headers	Headers provided on front and rear of backplane for daisy-chain of VMEbus signals.
Bus Termination	Signal lines are terminated with 330 Ω pull-up resistors and 470 Ω pull-down resistors. SYSCLK and SERCLK signal lines have clamp diodes for negative spike suppression.

FCC AND VDE COMPLIANCE

The Federal Communications Commission (FCC) has established technical standards regarding radiation and conduction of EMI and Radio Frequency Interference. The MVME945 chassis has been tested with Motorola VMEmodules and mass storage enclosures and found to comply with the emission limits for a class A computing device in accordance with the specifications of the FCC and VDE regulations. Compliance was achieved under the following conditions:

a) The chassis was connected to earth ground through the safety ground wire in the power cord. This provided the path for connecting shields to earth ground.
b) All external I/O cables were high quality shielded types with metal shell connectors. External I/O connectors on VMEmodules were metal shell types connected to chassis ground. This was accomplished through the conductive panels in contact with the chassis rails. VMEmodules must have conductive panels to ensure a good RF ground with the conductive chassis rails.
c) Blank panels were installed to cover all unused chassis slots to minimize RF leaks in the chassis.
d) Front panel screws were properly tightened for good grounding.
e) The desk top cover was installed. (If the MVME945 is rack mounted, the system integrator must ensure that the system remains compliant.)

For minimum RF emissions, it is essential that the foregoing conditions be implemented. Failure to do so could compromise the FCC or VDE compliance of the system.

In addition to meeting the FCC and VDE emission standards, the MVME945 meets all UL, VDE and CSA safety requirements.

CHASSIS COOLING CONSIDERATIONS

The MVME945 chassis provides forced air cooling for the standard VMEmodules, mass storage modules and power supply. The VMEmodules are cooled by air which is drawn into the bottom of the chassis by two low noise, 70 CFM, dc fans, forced past the VMEmodules, and blown out the top of the chassis. These fans provide air flow which is sufficient for most VMEmodules. Certain high density, high power modules, when used at elevated temperatures may require additional cooling. When possible, high power modules should be distributed throughout the chassis to reduce "hot spots." Also, the 70 CFM fans can be replaced by 100 CFM fans to improve air flow, but, at the expense of higher acoustic noise. The system integrator must ensure that adequate cooling is provided for all modules used in the system.

The mass storage enclosure and power supply are cooled by air which is drawn into the mass storage enclosure by a low noise, 70 CFM, dc fan and then forced past the power supply and out the left side of the chassis.

No forced air cooling is provided for the rear mounted MVME7xx series modules and optional I/O Channel card cage. The MVME7xx series modules are typically transition modules with few active components, and cooling is generally not required. Many I/O Channel modules are low power, low density designs that will be cooled adequately by convection. Thus, such modules normally do not require forced air cooling. However, if the MVME945 chassis is used at high temperatures (greater than 25°C), with the optional I/O Channel card cage and high power I/O Channel modules installed, forced air cooling for the I/O Channel modules might be required. The system integrator must ensure that adequate cooling is provided.

General Cooling Precautions

a) Ensure the air space above and below the chassis is not blocked, and the inlet screens are clean.
b) Air inlet temperature must not exceed 40°C.
c) All unused VMEmodule slots must be covered to prevent cooling air from escaping the chassis.

d) Chassis should be used only in a clean environment.

CAUTION: If the MVME945 is installed in a rack with other chassis, be sure each chassis in the rack is provided with its own supply of fresh outside cooling air.

MVME83x MASS STORAGE ASSEMBLIES

The MVME83x series assemblies provide mass storage for Motorola VMEsystems. They slide directly into the right front side of an MVME945 chassis and are connected by cable to the MVME320A-1 and MVME350 controller boards (Not included with the Mass Storage Assemblies). A support frame which slides into four heavy duty VME-module card guides and is retained in place by four captive screws on the front panel is used for mounting an assembly. Major components of the MVME83x series assemblies are described in the following paragraphs.

WINCHESTER HARD DISK (MVME833/MVME834)

The Winchester disk supplied with the MVME833 and MVME834 Plug-in Mass Storage Modules will be a Micropolis 1320 series, a Toshiba MK-56 series, or similar type drive. Both of these drives utilize a Winchester-type head/media technology and provide both dc operating voltages and outline dimensions consistent with the industry-standard. The specifications for the Micropolis 1320 series are shown in Table 4 and the specifications for the Toshiba MK-56 are shown in Table 5. Should a different Winchester hard disk drive be utilized in the MVME833/MVME834, the specifications may not be identical to those shown.

Table 4. Micropolis 1320 Specifications

Characteristics	Specifications		
Capacity			
Unformatted			
Per Drive (Mb)	85.30		
Per Surface (Mb)	10.70		
Per Track (Bytes)	10,416		
Formatted*			
Per Drive (Mb)	67.10		
Per Surface (Mb)	8.39		
Per Track (Bytes)	8,192		
Per Sector (Mb)	256		
Sectors/Track	32		
Cylinders	1,024		
Read/Write Heads	8		
Pre Comp Cylinder	DO NOT USE		
Seek Time (includes settling)			
Track-to-Track (ms)	6		
Average (ms)	28		
Maximum (ms)	62		
Latency Time (avg)(ms)	8.33		
Power Requirements	Idle	Seeking	Peak
+5 Vdc +5%	0.9 A	0.9 A	0.9 A
+12 Vdc +5%	2.1 A	3.3 A	3.9 A
Power Dissipation	28 W	38 W	

*Assumes format compatible with Motorola OS.

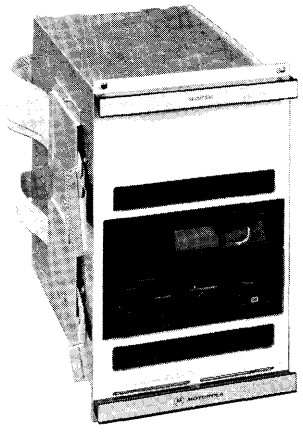


Table 5. Toshiba MK-56FB Specifications

Characteristics	Specifications
Capacity	
Unformatted	
Per Drive (Mb)	86.50
Per Surface (Mb)	8.64
Per Track (Bytes)	10,416
Formatted*	
Per Drive (Mb)	68
Per Surface (Mb)	6.80
Per Track (Bytes)	8,192
Per Sector (Mb)	256
Sectors/Track	32
Cylinders	830
Read/Write Heads	10
Pre Comp Cylinder	512
Seek Time (includes settling)	
Track-to-Track (ms)	6
Average (ms)	25
Maximum (ms)	45
Latency Time (avg)(ms)	8.33
Power Requirements	1.3 A max 0.09 A typ
+5 Vdc +5%	Seeking 4.2 A max 10 ms
+12 Vdc +5%	PK Starting 3.8 A max 120 sec
	Idle 1.5 A
Power Dissipation	30 W nom

*Assumes format compatible with Motorola OS.

FLOPPY DISK (MVME833/MVME834)

The floppy disk supplied with the MVME833/MVME834 Plug-In Mass Storage Modules is an industry standard 5-1/4", half height (1.63"), soft sector, FM/MFM mini-flexible disk drive. Several manufacturers' drives are qualified to be used in the MVME833/MVME834 and the specifications shown in Table 6 are generic to those manufacturers.

Table 6. Floppy Disk Specifications

Characteristics	Specifications
Capacity	
Unformatted	
Per Drive (Mb)	1
Per Surface (Mb)	0.5
Per Track (Bytes)	12,500
Formatted*	
Per Drive (Mb)	0.655
Per Surface (Mb)	0.3277
Per Track (Bytes)	8,192
Per Sector (Mb)	256
Sectors/Track	32
Cylinders	40
Head/Write Heads	2
Pre Comp Cylinder	NOT REQUIRED
Seek Time (includes settling)	
Track-to-Track (ms)	18
Average (ms)	109
Maximum (ms)	62
Head Settle Time (ms)	15
Latency Time (avg)(ms)	10
Power Requirements	
+5 Vdc +5%	0.3 A typ 0.38 A max
+12 Vdc +5%	0.22 A typ 0.54 A max
Power Dissipation	6 W

*Assumes format compatible with Motorola OS.

Table 7. Archive 5945L3 Specifications

Characteristics	Specifications
No. of Tracks	9
No. of Channels*	2
Capacity — DC 300XL	30Mb
Capacity — DC 600XL	60Mb
Backup Time — DC 300XL	9 min
Backup Time — DC 600XL	12 min
Recording Mode	NRZI
Recording Data Density	8000 bpi
Encoding Method	4 to 5 RLL**
Track Capacity — DC 300XL	5Mb
Track Capacity — DC 600XL	6.6Mb
Data Transfer Speed	90Kb/s
Tape Speed	90 ips
Start/Stop Time	300 ms max

STREAMING TAPE DRIVE (MVME834 ONLY)

The streaming tape drive supplied with the MVME834 Plug-In Mass Storage Module will be the Archive Corp. model 5945L3 or similar type drive, a 9-track half-height drive capable of streaming or start/stop operation with data capacities of 30Mb or 60Mb when used with the DC 300XL or DC 600XL cartridges respectively. Also included is a formatter/controller for a QIC-02 interface which, when integrated with the drive results in a full height 5-1/4" module. Specifications for the 5945L3 are shown in Tables 7 and 8. Should a different streaming tape drive be used in the MVME834, the specifications may not be identical to those shown.

MVME833/MVME834 CABLING

The MVME833 and MVME834 are provided with the cabling required to interface directly to a series of Motorola mass storage controllers. The Winchester and floppy disk drives directly interface the MVME320A or MVME320A-1, and the streaming tape drive connects directly to the MVME350. Cabling instructions are provided in the MVME833 and MVME834 manuals.

ACCESSORIES

MVME945I/O — This kit provides the hardware required to add 9-slots of I/O Channel expansion capability to the MVME945 chassis. Included is the I/O Channel card cage, backplanes and interconnection cable to attach directly to a P2 connector of the 12-slot VMEbus backplane. When assembled and installed in the rear of the MVME945 chassis, the I/O Channel expansion kit occupies 11 of the MVME7xx transition module slots, leaving 10 for transition module requirements.

For I/O Channel operation, an I/O Channel master VME-module such as the MVME110-1 or MVME316 (not included in the kit) is required.

MVME945CNVKT — This kit provides the material required to convert the MVME945 from 115 Vac operation to 230 Vac operation. Included are 230 Vac rating labels, 2 fuses and detailed instructions for the conversion process.

MVME945CVR — This kit provides the desk top cover and attachment hardware for the MVME945 chassis. The cover is designed to provide FCC and VDE compliance but cannot be used when the chassis is rack mounted.

MVME83XKT — This kit provides the metal work used in the MVME83x series mass storage modules. The frame is capable of accepting 2 full-height and 1 half-height, 5-1/4" mass storage modules, and gives the system integrator the ability to construct a custom mass storage module to meet his own application requirements. No drives or cables are supplied, but detailed instructions are provided for assembly as well as examples of how to install and cable various mass storage modules.

MVME83XSTKT — This kit provides all the necessary material to convert an MVME833 to an MVME834. Included is the required metal work, hardware, streaming tape drive and cables. The streaming tape controller board (MVME350) is not included and should be ordered separately.

Table 8. Archive 5945L3 Power Specifications

DC Voltage	+12 Vdc	+5 Vdc
Tolerance (includes 200 mv max ripple)	+10%	+10%
Operational Current	1.75 A +0.8 (cartridge dependent)	2.4 A max
Tape Start or Stop Surge Current Peak	4.15 A max up to 300 ms	0.0 A
Power On Surge Current	Thru 200 μF max cap	Thru 60 μF max cap
Voltage Rise Time	100 ms max	100 ms max
Power Sequence	None	None
Power Dissipation (in continuous streaming mode)	21 Watts	12 Watts
Power Dissipation (during start or stop power surges)	50 Watts	12 Watts

ORDERING INFORMATION

Part Number	Description
MVME945-1	VME Chassis Assembly with 12-slot Backplane, 400 W Power Supply and provision for mass storage modules. Operates at 115 Vac. Includes User's Manual.
MVME945-2	VME Chassis Assembly with 12-slot Backplane, 400 W Power Supply and provision for mass storage modules. Operates at 230 Vac. Includes User's Manual.
MVME833	Plug-In Mass Storage Unit for use with MVME945 — Includes 70Mb Winchester 655Kb Floppy and all required cables. Includes User's Manual.
MVME834	Plug-In Mass Storage Unit for use with MVME945 — Includes 70Mb Winchester 655Kb Floppy, Streaming Tape and all required cables. Includes User's Manual.
MVME945I/O	I/O Channel Expansion Kit. Includes installation instructions.
MVME945CNVKT	115 Vac to 230 Vac Conversion Kit. Includes installation instructions.
MVME945CVR	Desk Top Cover for MVME945. Includes installation instructions.
MVME83XKT	Custom Mass Storage Expansion Kit. Includes installation instructions.
MVME83XSTKT	Streaming Tape Upgrade Kit for MVME833. Includes installation instructions.

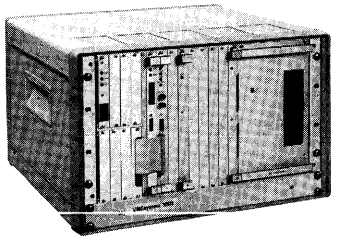
RELATED DOCUMENTATION

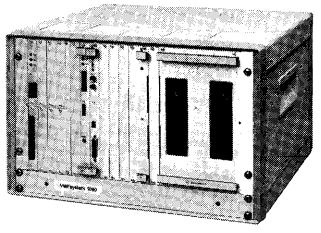
Part Number	Description
MVME945/D	MVME945 Chassis User's Manual.
MVME830/D	MVME833/MVME834 Plug-In Mass Storage User's Manual.
MVME945IO/D	MVME945 I/O Channel Expansion Kit Installation Instructions.
MVME945CP/D	MVME945 115 Vac to 230 Vac Conversion Kit Installation Instructions.
MVME945CVR/D	MVME945 Desk Top Cover Installation Instructions.
MVME83XKT/D	MVME83x Custom Mass Storage Expansion Kit Installation Instructions.
MVME834UP/D	MVME833 Streaming Tape Upgrade Kit Installation Instructions.
MVME320A/D	VMEbus Disk Controller Module User's Manual.
MVME350/D	VMEbus Streaming Tape Controller User's Manual.

SYS1121UY221
SYS1131UY231

Advance Information

VMEbus Modular Systems With SYSTEM V/68™





SYS1121 FEATURES

- SYSTEM V/68, Release 2
- MVME202 512Kb DRAM Module
- MVME320 Disk Controller
- MVME050 System Controller
- 40Mb Winchester
- 655Kb Floppy Disk
- Three RS-232-C Serial Ports (total)
- One Centronics Printer Interface
- Five VMEbus Expansion Slots
- MVME121 Microcomputer with
  - — MC68010 Microprocessor
  - — MC68451 Memory Management Unit
  - — 512Kb Dual-Ported DRAM
  - — 4Kb Instruction Cache
  - — A24:D16 VMEbus Interface
  - — One RS-232-C Serial Port

SYS1131 FEATURES

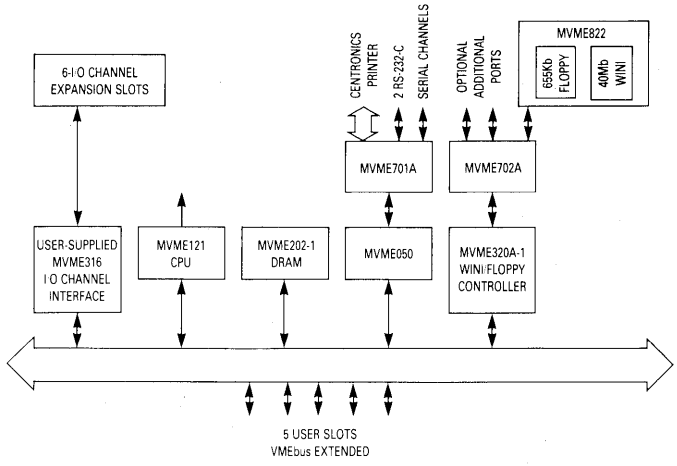
- SYSTEM V/68, Release 2
- MVME204-2 2Mb Dual-Ported DRAM with MVMX32bus Interface
- MVME320 Disk Controller
- MVME050 System Controller
- 70Mb Winchester
- 655Kb Floppy Disk
- Four RS-232-C Ports (total)
- One Centronics Printer Interface
- MVME131 Microcomputer with
  - — MC68020 Microprocessor
  - — MMB851 Memory Management Unit
  - — Socket for MC68881 Floating Point Co-Processor
  - — Two Sockets for JEDEC 28-pin ROM/EPROM Devices
  - — Two Sockets for JEDEC 28-pin RAM/ROM Devices
  - — Two Multi-Protocol Serial Ports
  - — A32:D32 VMEbus Interface
  - — MVMX32bus Interface

SYS1121 & SYS1131 MVME943 CHASSIS FEATURES

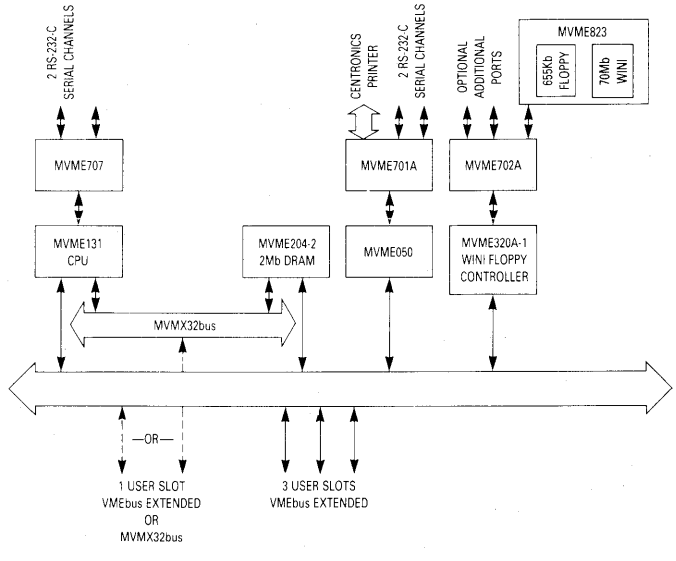
- 19" Rack Mount Chassis with Tabletop Enclosure
- Transverse 400 W Mid-Chassis Power Supply
- Nine-Slot, VMEbus-Compatible Backplanes
  - — SYS1121 — A24, D16 Backplane
  - — SYS1131 — A32, D32 Backplane
- Accommodate up to 16 Rear-mounted 80 mm I/O Transition Modules
- SYS1121 — Double, 3-Slot, Single-High, Front-Accessible I/O Channel Backplane

The greatest advantage of a VMEbus-based system is its modularity. To a core system, add only those software and hardware components required for the application. Then serve a new application by changing only a component or two.

The two VMEbus Modular Systems with SYSTEM V/68, SYS1121UY221 and SYS1131UY231, are versatile core systems configured using standard VMEbus modules. Supplied in 19" rack-mountable chassis with an enclosure for desktop use, the systems have ample room for hardware expansion. Each system offers outstanding performance by virtue of an intelligent mass storage controller, high-performance system RAM and a powerful monoboard microcomputer based on a fast, MC68000-family device: SYS1121U uses the MC68010 16-/32-Bit Microprocessor, SYS1131U uses the MC68020 32-Bit Microprocessor. The MVME050 System Controller and the MVME701A I/O Transition Modules provide bus control and accommodate serial and parallel I/O for both systems.



SYS1121U221 Hardware Configuration



SYS1131U231 Hardware Configuration

These modular systems provide a high-performance core around which, using standard VMEmodules, a microcomputer system serving a specific application can be configured. Five additional slots in the VMEbus backplane are available for SYS1121 and four for SYS1131. Positions are available for adding additional I/O transition modules, if desired.

For software development under the multiuser, multitasking system — SYSTEM V/68, SYS1121U provides a powerful 3-user environment while SYS1131U provides a 4-user environment. Additional users can be accommodated by adding an appropriate VMEmodule such as the MVME331 Intelligent 6-Channel Serial Communication Module. The MVME320A-1 Intelligent Mass-Storage Controller supplied with the core systems enhances the use of system resources by SYSTEM V/68.

SYSTEM V/68

Derived from UNIX System V/M68000, the product jointly developed by Motorola Inc. and AT&T, SYSTEM V/68 is the validated port to the M68000 family of microprocessors. It offers a small compact kernel, which provides process scheduling and I/O facilities to all programs. Its powerful command processor, the Shell, provides for interactive control. An extensive set of tools and utilities supports program development, text processing, electronic mail and system-to-system communications. As an environment for developing application software, SYSTEM V/68 is unsurpassed.

THE SYSTEM V/68 KERNEL

The SYSTEM V/68 kernel supports a multiuser, multitasking software development and/or application execution environment. The kernel size is typically 150Kb but, depending on target system requirements, may be reduced. SYSTEM V/68 supports a layered or hierarchial file system of arbitrary size. File blocks are 1024 bytes long and a hashed i-node look up feature quickens file access. The kernel is of the swapping variety. Processes may share memory segments through the optionally configurable shared memory support.

The kernel utilizes the MC68451 Memory Management Unit chip (SYS1121U) and the MMB851 gate array implementation of the MC68851 Paged Memory Management Unit chip functionality (SYS1131U) to support several SYSTEM V/68 features.

HIGH LEVEL SUPPORT

Integral to SYSTEM V/68 are two categories of high level software tools which provide the means for creating, optimizing and maintaining application software. Other tools supplied with SYSTEM V/68 support text processing and communications between systems running UNIX.

Major components of the first category are the C language compiler "cc," the FORTRAN 77 compiler "f77" and the M68000/M68010/MC68020 assembler "as." Both compilers are supported by the linker/loader "ld" which supports the common object file format: COFF. Supplied also is the symbolic debugger "sdb" which can be used on C source code, FORTRAN and assembler code modules.

Software components in the second category include five editors in support of text processing under SYSTEM V/68 and two utilities supporting text formatting. The full screen editors are "ex" and "vi," "ed" is a line editor while "sed" is stream oriented and noninteractive. The editor "bfs" is a big file scanner similar to "ed."

Under SYSTEM V/68, support is also provided for electronic mail, communications and networking. By allowing a system to be used as a mail box or bulletin board, the mail utility facilitates intrasystem communications between users. Other utilities support communications with mainframe computers and networking utilities allow several computers to be linked together through dedicated links or by dial-up connections so that SYSTEM V/68 users can communicate with users on other systems running UNIX.

SYSTEMS CENTRAL PROCESSING UNIT

To provide the computing power needed to take full advantage of the vast capabilities of UNIX V, in the form of the SYSTEM V/68 Multiuser, Multitasking Operating System, a monoboard microcomputer based on a fast microprocessor of the Motorola MC68000 family is used as the central processing unit by the two core systems.

SYS1121UY221 CPU — Serving as the MPU nerve center of this core system is an MVME121 VMEbus Microprocessor Module which features a 10 MHz MC68010 16-Bit Virtual-Memory Microprocessor, an MC68451 Memory Management Unit, 4Kb of instruction-only cache memory and 512Kb of dual ported dynamic RAM with byte parity.

Use of the large onboard RAM provided with this core system to continuously hold the kernel during run time can optimize SYSTEM V/68 performance. Enabling the cache memory can further improve overall system performance.

On its way to the MVME121 MPU after a fetch from system memory, an instruction is stored in the very fast, onboard cache. A subsequent fetch of that instruction, if needed, is made from cache rather than system memory significantly reducing execution time. Depending on the type of program, use of the cache memory can improve system performance by 30% to 40%.

SYS1131UY231 — Unsurpassed in its processing power, the intelligent master of this core system is an MVME131 VMEbus Microprocessor Module featuring the state-of-the-art, full 32-bit MC68020 Microprocessor and including also MVMX32bus, a high-speed dedicated interface to the MVME204-2 2Mb Dynamic RAM, provision for adding the MC68881 Floating Point Coprocessor and the functionality of the MC68851 Paged Memory Management Unit implemented with gate array technology.

The SYS1131UY231 system also includes an MVME707 RS-232 Serial Port Distribution Module which translates the TTL levels output by the two MVME131 ports to provide full support of RS-232-C asynchronous serial communications and which provides also two DB-25 connectors for serial I/O cables. This module is a double-high, half deep board mounted on the upper and lower rear chassis rails and connected to the MVME131 by a 50-conductor ribbon cable.

HARDWARE SUPPORT FOR MEMORY MANAGEMENT

Since SYSTEM V/68 utilizes memory management techniques in the manner of mainframe operating systems, hardware support for the translation of logical addresses into physical addresses and for managing the separation of user and supervisor memory spaces is essential. The MC68451 Memory Management Unit provides this support for SYS1121UY221. For the SYS1131UY231 core system, memory management support is provided by the MMB851 gate array implementation of the MC68851 Paged Memory Management Unit function.

CORE SYSTEM MAIN MEMORY

SYS1121UY221 — The 1Mb of DRAM can be supplemented by 1/2Mb, 1Mb, or 2Mb per VMEbus slot, using respectively, MVME202, MVME222-1 or MVME222-2. Up to 10Mb of additional DRAM can be added to a system.

All three memories support both byte and word accessing and upper or lower byte parity generation and checking, as selected. The MVME202 uses 64K-bit devices. MVME222-1 and MVME222-2 use 256K-bit devices. A memory module base address is selectable over the entire 16Mb address range of the MVME121 on boundaries equal to the module population. Parity Error and Module Busy status LEDs are provided on module front panels.

SYS1131UY231 — A 2Mb DRAM with parity dual-ported between VMEbus and MVMX32bus — MVME204-2 — is supplied with this core system. Up to four additional MVME204-2's can be added to this interface to provide a core system with 8Mb of high speed memory. A base address for a module can be set on boundaries separated by the module population.

Tailored to the MC68020 Microprocessor and specifically designed to enhance the performance of the MVME131 VMEmodule 32-Bit Monoboard Microcomputer, the MVME204-2/MVMX32bus interface allows concurrent transfers on VMEbus with MVMX32bus transfers between the MVME131 and MVME204-2. Implemented using high density 256K x 1-bit dynamic RAM devices, the MVME204-2 has parity generation and detection circuitry which, together with accessible control and status registers, can be used for error detection and memory diagnostics.

System memory capacity can be expanded by adding DRAM modules to VMEbus space. For SYS1121, choose from MVME202, MVME222-1 or MVME222-2 to add 1/2Mb, 1Mb or 2Mb, respectively. For SYS1131, use MVME204-1, or -2 to add 1Mb or 2Mb, respectively.

SYSTEM CONTROLLER — I/O INTERFACES

Certain control functions essential to a multiprocessor, bus-based system are required once and need not be supplied by each module. These include system reset, priority bus arbitration, bus timeout monitoring and the generation of system, serial and time-of-day clocks. The MVME050 System Controller generates these functions for optional use and also provides a Centronics-compatible parallel printer port and two multiprotocol RS-232-C serial ports.

Both systems also utilize an MVME701A I/O Transition Module to provide the connectors for the serial and parallel ports of the MVME050 and to accommodate the optional backup battery for the time-of-day clock. The module is connected by 64-conductor ribbon cable to MVME050 connector P2.

The TTL levels output by the dual multiprotocol serial ports of the MVME131 processor module used by SYS1131UY231 are translated to RS-232-C levels by means of an MVME707 I/O Transition Module which also provides the appropriate connectors for external equipment observing this communications protocol. A 50-conductor ribbon cable is used for the MVME131-to-MVME707 connection.

The rear-mounting MVME701A and MVME707 modules are two of several which permit I/O connections to be made at the back of a rack if required by the particular application or from the front in a development environment.

SYS1121 I/O CHANNEL

The Motorola I/O Channel provides a 12-bit address, an 8-bit bidirectional data bus, 4Kb of memory mapped I/O and a data transfer rate up to 2Mb per second. An I/O Channel Specification defines connectors, pin assignments, ribbon cable characteristics, board level loading and driver/receiver parameters.

Most of Motorola's I/Omodules are implemented using the single-high Eurocard format. Serial and parallel interface, disk controller, magnetic tape interface, A/D and D/A converter and ac and dc input and output functions are offered in this format. Other functions compatible with the I/O Channel protocol are offered in special mechanical formats. Note that support for the I/O Channel interface is required for use of I/Omodules in an expanded core system.

MASS STORAGE DEVICES CONTROLLER

To control the provided floppy disk drive and Winchester disk drive, both VMEbus Modular Core Systems utilize the MVME320A-1 VMEbus Disk Controller. Although not supported by either core system, this VME-module has the capability of controlling mixed 5-1/4" and 8" drives and can accommodate two 5-1/4" Winchester hard disk drives and two 5-1/4" floppy disk drives or up to four floppy disk drives.

So that commonly available rather than custom cables can be used for the connections between the controller and the disk drive connectors, an intermediate transition board, the MVME702A Disk Interface Module is used. This module re-orders MVME320A-1 signals to the requirements of the standard 34-pin and 50-pin connector interfaces used by most 5-1/4" and 8" hard disk and floppy disk drives. It also buffers signals for 5-1/4" drives and provides a means of terminating unused signals from the controller.

Because the controller provides high-performance direct memory access (DMA) data channels between core system memory and the floppy and hard disk storage, the core systems can be used in applications having intensive real-time disk I/O or multiprocessing structures designed to reduce VMEbus traffic and increase system throughput. Other MVME320A-1 features include:

- Supports Serial Data Rates to 5Mbps
- Supports Standard IBM Formats and MFM Recording
- 16-Bit Cyclic Redundancy Check
- 32-Bit Error Checking and Correction

MASS STORAGE DEVICES

SYS1121UY221 — Mass storage capacity for this core system is provided by a floppy disk drive and a Winchester hard disk drive.

The floppy disk is a 5-1/4" slim line drive for soft-sectored dual-sided dual-density, 96 TPI diskettes each

with a capacity of 655Kb of formatted data. Features include a track density of 96 T.P.I., a recording density of 5922 B.P.I. (double density) and a data transfer rate of 250Kb per second.

The hard disk utilizes reliable Winchester technology and provides a capacity of 40.8Mb, formatted. Features include 32 sectors per track, 256 bytes per sector and a data transfer rate of 5Mb per second.

SYS1131UY231 — Mass storage capacity for this storage system is provided by a floppy disk drive and a hard disk drive.

The floppy disk is a 5-1/4" slim line drive for soft-sectored dual-sided dual-density, 96 TPI diskettes each with a capacity of 655Kb of formatted data. Features include a track density of 96 T.P.I., a recording density of 5922 B.P.I. (double density) and a data transfer rate of 250Kb per second.

The hard disk uses reliable Winchester technology and provides a capacity of 70Mb, formatted. Features include 32 sectors per track, 256 bytes per sector and a data transfer rate of 5Mb per second.

USER INTERFACE

Starting with a chassis suitable for nourishing and protecting the inhabitants, a VMEbus-based system can be constructed from the wide variety of commercially available modules to serve almost any application. The core of this system must simultaneously offer configuration flexibility and ease of operation.

Each core system is housed in a MVME943 chassis assembly, the desktop decorative cover of which can be removed so that the system can be mounted in a standard 19-inch rack, if the application requires. The front of the chassis houses a card cage with backplane which accepts nine standard double high Eurocard VMEmodules. SYS1121 also includes two card cage/backplanes which each accept three single high I/Omodules. On the right, the floppy and hard disk drives supplied with a core system are housed. On the left, a power switch module is attached.

On the left in the chassis rear, a circuit breaker panel is attached by screws to the upper and lower rails, as are the two or three 80 mm I/O Transition Modules supplied with their respective core systems. The remaining space can be used for additional transition modules.

A 400 watt power supply is mounted in a fan-cooled air duct assembly located in the upper half of the chassis to the rear of the VMEmodule backplane. Additional cooling sufficient for an expanded system is provided in the chassis.

By virtue of the high intelligence and integration levels of their modules, both core systems are able to present to the user an uncluttered front which offers the essential basic controls and system status indications and, at the same time, the means of configuring the hardware to provide certain system level functions.

Chassis Front Panel

+5 Vdc, +12 Vdc, −12 Vdc Indicators, Green LED
ON/OFF Power Switch, Rocker

MVME050 Front Panel

RUN Indicator, Green LED
FAIL Indicator, Red LED
Dot Matrix Display, 2 Hex Digits
Reset Switch, Spring-Loaded Pushbutton
8-Position DIP Switch

MVME320A-1 Front Panel

FAIL Indicator, Red LED
Two Slots for ribbon cable entry/exit

MVME121 Front Panel — System 1121UY

FAIL Indicator, Red LED
HALT Indicator, Red LED
RUN Indicator, Red LED
4-Position DIP Switch
DB-25 Connector for terminal cable

MVME131 Front Panel — System 1131UY

FAIL Indicator, Red LED
HALT Indicator, Red LED
RUN Indicator, Green LED
SCON Indicator, Green LED
Reset Switch, spring-loaded pushbutton
Abort Switch, spring-loaded pushbutton
8-Position DIP Switch
50-Pin Connector

MVME204-2 Front Panel — System 1131UY

Power Indicator, Green LED

MVME202, MVME222-1,-2 — Optional System DRAM

Select Indicator, Green LED
Parity Error, Red LED

To accommodate the need for configuring a system to the requirements of an application or for development work, the DIP front panel switches on MVME121, MVME131 and MVME050 are software readable and the programmable, two digit display on the MVME050 System Controller can also be used for diagnostic purposes.

EXPANSION FOR ADDITIONAL USERS

The backplane supplied with the core system chassis can accommodate additional VMEmodules. Access to the unused slots is gained by removing as many dummy module front panels as required by removing the screws fastening them to the upper and lower rails of the chassis front.

The three or four serial interfaces provided with the core systems for terminals or other I/O devices can be expanded by six RS-232-C or RS-422-B ports by installing the MVME331 Intelligent Communications Controller Module in the chassis backplane. A companion I/O Transition Module, the MVME705 which supplies the DB-25 connectors and the transmitters/receivers required to support the two communication protocols is installed by screwing the module front panel to the top and bottom rear rails of the chassis. Connection is made between the two modules using a 64-conductor ribbon cable.

Memory for either system can be supplemented in increments of 0.5, 1 or 2 megabytes by adding the MVME202, MVME222-1 or MVME222-2 Dynamic RAM modules, respectively. SYS1131 memory can be expanded using the MVME204-1 or -2 to add 1Mb or 2Mb, respectively.

Extensive documentation for the hardware and SYSTEM V/68 is supplied with the core systems which details the procedure for reconfiguring and for generating a new kernel to accommodate the addition of any new modules.

Documentation for SYSTEM V/68 supplied with the core systems describes in detail the process of generating a new kernel for use with an expanded system. Hardware and module documentation supplied with the core systems and expansion modules details the process of configuring a system to comply with the requirements of various applications.

VMEbus

Rapidly becoming the leading bus standard in U.S., European and other world markets, VMEbus is a 8-/16-/32-bit interconnect that has a master/slave asynchronous, non-multiplexed data transfer structure, seven levels of priority interrupt, four levels of data bus arbitration, multiple master support, fault detection and control, and special transfer cycles.

The VMEbus functional structure consists of backplane interface logic, four groups of signal lines called buses each having its own collection of functional modules any number of which can be grouped on a board and which communicate with the other functional modules of a bus via onboard or backplane signal lines.

The VMEbus mechanical structure comprises subracks, backplanes, front panels, plug-in boards and other elements all of which conform to relevant IEC Eurocard specifications to insure that all components fit together properly.

These features combined with reliability, mechanical integrity, 20 megabytes-per-second performance and the fact that it is nonproprietary have resulted in almost complete acceptance of VMEbus by the electronics industry. Currently over 200 vendors are manufacturing over 1000 VMEsystem-compatible products, assuring for the expansion of a core system, the widest selection of functions from the greatest number of sources so that the broadest range of applications can be served.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

MVME1121UY221	
Characteristics	**Specifications**
MCU	MVME121 VMEbus Microprocessor Module (10 MHz MC68010 Microprocessor)
Instruction Cache	4Kb
Memory	1Mb Dynamic RAM (MVME121 plus MVME202)
Mass Storage Controller	MVME320A-1 VMEbus Disk Controller Module
Mass Storage	5-1/4" 40Mb (formatted) Winchester Disk 5-1/4" 655Kb (formatted) Floppy Disk Drive
I/O Interfaces	2 x RS-232-C (programmable to 19.2 Kbaud) 1 x RS-232-C (terminal mode only, programmable to 9600 baud) 1 x Centronics-compatible parallel printer interface Rear serial/parallel I/O connections via MVME70IA Module
I/O Channel	Double, 3-Slot, Single-High I/O Channel Backplane
Expansion Slots	9-slot Backplane, 5 slots free for user
Power Supply	400 W (115 Vac)
Operating Temperature	5° to 40°C
Operating Humidity	20% to 80% (non-condensing)

2

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

MVME1131UY231	
MCU	MVME131 VMEmodule 32-Bit Address/Data Monoboard Microcomputer (12.5 MHz MC68020 Microprocessor)
Memory	2Mb Dual-Ported RAM with MVMX32bus (MVME204-2)
Mass Storage Controller	MVME320-1 VMEbus Disk Controller
Mass Storage	70Mb (formatted) Winchester Disk Drive 655Kb (formatted) Floppy Disk Drive
I/O Interfaces	4 x RS-232-C programmable to 19.2K baud 1 x Centronics-Compatible Printer Interface Rear serial/parallel I/O connections via MVME701A & MVME707 Transition Modules
Expansion Slots	9-slot Backplane, 4 slots free for user
Power Supply	400 W (115 Vac)
Operating Temperature	5° to 40°
Operating Humidity	20% to 80% (non-condensing)

ORDERING INFORMATION

Part Number	Description
SYS1121UY221	VMEbus Modular System with SYSTEM V/68, includes: MVME121 VMEbus Microprocessor Module MVME320A-1 VMEbus Disk Drive Controller MVME202 512Kb Dynamic RAM MVME050 System Controller MVME701A Transition Module MVME822 Plug-In Mass Storage Unit with — 40Mb Winchester Disk Drive — 655Kb Floppy Disk Drive MVME943 VMEbus Chassis Assembly with — MVME921A 9-Slot VMEbus Backplane — Two MVME924 3-Slot I/O Channel Backplanes — 400 W Power Supply User's Manuals for SYSTEM V/68, chassis and all modules included
SYS1131UY231	VMEbus Modular System with SYSTEM V/68 includes: MVME131 VMEmodule Monoboard Microcomputer MVME204-2 2Mb Dual-Ported DRAM MVME320A-1 VMEbus Disk Drive Controller MVME050 System Controller MVME701A Transition Module MVME707 Transition Module MVME823 Plug-In Mass Storage Module with — 70Mb Winchester Disk Drive — 655Kb Floppy Disk Drive MVME943 VMEbus Chassis Assembly with — 32-Bit VMEbus Backplane — 400 W Power Supply User's Manuals for SYSTEM V/68, chassis and all modules included

OPTIONS

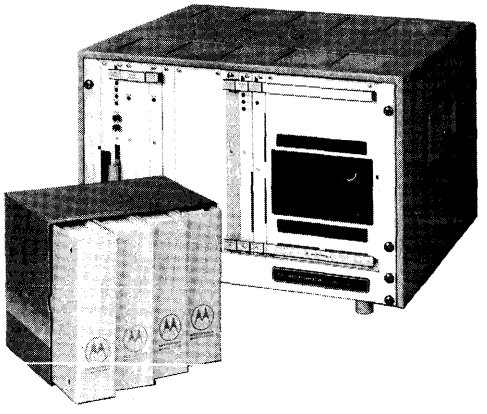
Part Number	Description
MVME202	512Kb Interleaved DRAM Module
MVME204-1,-2	1Mb/2Mb Interleaved Dual-Ported DRAM Modules
MVME214	A32/D32 Static RAM/ROM Memory Module
MVME222-1,-2	1Mb/2Mb Interleaved DRAM Modules
MVME331,332	6/8 Channel Intelligent Communications Controllers
MVME705	I/O Transition Board for MVME331
MVME350	Intelligent Streaming Tape Controller
MVME390A	Color Graphics Controller Module

RELATED DOCUMENTATION

Part Number	Description
MVMESYSAM/D	VME System Architecture Guide
MC68010/D	MC68010 Microprocessor Data Manual
MC68020/D	MC68020 Microprocessor User's Manual
MC68451/D	MC68451 Memory Management Unit Data Manual

Advance Information

Software Development System



- Complete, High-Performance MC68020-Based SYS1131UY341 Plus Additional Software Development Support:
  - MC68000/008/010 Cross Pascal Compiler
  - M68000/008/010/020/851/881 Family Assembler/Linker
  - SYSTEM V/68 to VERSAdos Code Conversion/Utility Toolkit
  - MC68000/008/010 Cross C Compilation System
  - Resident MC68020 C Source Level Debugger
  - HDS-300 Development System Interface Support For Adding Hardware Development Capabilities
- Application Information. Topics Include:
  - Creating A Run-Time Library
  - How To Prepare ROMable Code
  - File Conversion
  - Preparing A Target Simulator
  - Cross MPU Considerations

Development of software for microcomputer systems is typically done with the aid of a host computer used for its storage facilities and its independent software development power. The SYS1131DVLP System is a standalone, self-contained MC68020-based microcomputer host offering extremely high performance and versatile support to those developing software for systems based on 68000 family devices. Its target system hardware development capabilities are readily enhanced by interfacing Motorola's emulation/analysis tool — the HDS-300 Microprocessor Hardware/Software Development Station. The HDS-300 supports development of systems based on the MC68020, MC68010, MC68008, MC68000/HC000, MC6801/03, MC6809, and MC68HC11 microprocessors.

A 19" Rack Mount Chassis with a 12-slot VMEbus-compatible card cage (A32:D32 backplane) houses the five VMEmodules supplied with SYS1131DVLP allowing ample space for future expansion. Complementing the MC68020-based microcomputer engine, MVME131XT, are the MVME204-2 2Mb Dual-Ported DRAM, the MVME320A-1 Disk Controller, the MVME350 Streaming Tape Controller and the MVME331 6-Channel Serial Communications Controller. Mass memory includes a 70Mb Winchester Disk Drive, a 655Kb Floppy Disk Drive and a QIC-02 Cartridge Streaming Tape Drive.

The SYS1131DVLP chassis has a transverse, center-mounted 400 W power supply, can accommodate up to 20 rear-mounted 80 mm I/O transition modules and is supplied with an enclosure for tabletop use.

With the SYSTEM V/68 Operating System installed, SYS1131DVLP becomes a host environment of unsurpassed power for developing application software. The utilities and other facilities of SYSTEM V/68 are supplemented by an assortment of software tools to accommodate a wide variety of development environments.

The multitasking operating system services up to eight users and additional users can be accommodated by adding to the chassis an appropriate VMEmodule such as another MVME331 Intelligent 6-Channel Serial Communications Controller. Use of system resources by SYSTEM V/68 is enhanced by the MVME320A-1 Intelligent Mass Storage Controller supplied with SYS1131DVLP.

SYSTEM V/68

Derived from UNIX System V/M68000, the product jointly developed by Motorola Inc. and AT&T, SYSTEM V/68 is the validated port to the M68000 family of microprocessors. It offers a small compact kernel, which provides process scheduling and I/O facilities to all programs. Its powerful command processor, the Shell, provides for interactive control. An extensive set of tools and utilities supports program development, text processing, electronic mail and system-to-system communications.

MC68000/MC68008/MC68010 CROSS C COMPILATION SYSTEM

Because it is structured, typed and extensible, the general purpose language C has become especially useful for system programming applications. The popular language, in which about 90% of the AT&T operating system, UNIX, is written has several attractive attributes. Using C rather than another high level language, a programmer can more easily manipulate the hardware comprising a system and the resulting application program is more easily ported to another operating system.

The history of C has seen users expand its basic I/O functions to the point that a very large number of functions are readily available which are easily adaptable for use in an application program. Furthermore, in the SYSTEM V/68 environment with its I/O redirection capabilities, even more power and versatility are available to the system programmer for developing system input and output support.

Installed on the SYS1131DVLP, the Cross C Compilation System may be used to generate code which will run on a target system based on an MC68000, MC68008 or MC68010 microprocessor. In addition to the standard SYSTEM V/68 utilities, the compiler provides an extensive set of utilities for cross development.

M68000 FAMILY ASSEMBLER/LINKER

Those preferring to develop application software in assembly language may produce code executable on targets using any of the MC68000, MC68008, MC68010 and MC68020 microprocessors by means of the M68000 Assembler/Linker. This program was developed using the standard Motorola syntax for these processors and includes the following features:

- Absolute/relocatable Code Generation
- Complex Expressions
- Symbol Table Listing
- Macros
- Conditional Assembly
- Structured Syntax
- Cross Reference Listing
- IEEE Standard (MC68881 only) Floating Point
- MC68851 PMMU Support

PAL MC68000/MC68008/MC68010 CROSS PASCAL COMPILER

The PAL Compiler is a cross software package which runs under SYSTEM V/68 on SYS1131DVLP and produces object code for execution on target systems based on the MC68000, MC68008 and MC68010 microprocessors. This product is called PAL (Pascal/Assembler/Linkage Editor) to distinguish it from the common assembler (as) and linker (ld) within SYSTEM V/68. All support required for producing code modules executable on the target microprocessor is provided including: a structured macro assembler, a linkage editor and VERSAdos runtime libraries. SYSTEM V/68 libraries are also included for SYS1131DVLP target system programs.

The PAL Pascal compiler processes Motorola Pascal source code generating relocatable object code in a format defined by the PAL Linkage Editor. This code may be linked, if desired, with relocatable object code generated by the PAL Assembler. The resulting code is linked with the appropriate runtime libraries to form executable load modules for a target processor.

Assembly language programming support for the SYS1131DVLP host running SYSTEM V/68 is provided by the PAL Assembler. This is a structured assembler providing macro instruction and conditional assembly capabilities, evaluation of complex expressions, inclusion of input from other disk files, and to facilitate the writing of structured code, high level control including: for, while, repeat, and if-then, if-then-else structures. The assembler generates absolute or relocatable code, provides symbol table and cross reference listings and supports the MC68881 Floating Point Co-Processor and the MC68851 Paged Memory Management Unit.

The PAL Linkage Editor provides linkage and relocation capabilities for producing executable load modules from code generated by the PAL Pascal Compiler and PAL Assembler.

SYSTEM V/68/VERSAdos CODE CONVERSION/ UTILITY TOOLKIT

The Toolkit is a software package comprising various software development environment utilities and conversion programs which allow modules created to run under one operating system to be modified for use with another operating system. Some of the capabilities the Toolkit provides include:

- Interface Support for Connecting an HDS200 or HDS-300 Hardware/Software Development System to the SYS1131DVLP
- SYSTEM V/68 Assembler as(1) to Motorola ASM Assembler Relocatable Object File Conversion
- Conversion of Assembler Relocatable Object Files to SYSTEM V/68 C Compiler Relocatable Object Files and Vice Versa
- Upload/Download of S-Records for File Transport
- Initialization of VERSAdos Disks
- Inter-operating System File Copy
- Communication to Models 29 A, B Data I/O PROM Programmers

RESIDENT C/ASSEMBLER SOURCE LEVEL DEBUGGER

The Source Level Debugger (SLD) is a state of the art development tool that facilitates debugging of C or assembly language source code written for the MC68020 microprocessor. (Extension to other M68000 family processors is planned.) It can be used on assembler source or on C source code produced by any C compiler which generates an output file in the UNIX common output file format (COFF) and also supplies appropriate symbol information. The new tool has a multiple window design for broad program perspective and a one line assembler/disassembler that eases the modification of source lines. Major features of the SLD include:

- Breakpoint Insertion
  - — C Source Lines
  - — Assembly Language Lines
  - — Functions
  - — Physical Addresses

- Execution Control
  - — C Source or Assembly Level Single Step
  - — Re-execution
  - — Free Running Execution
- Symbolic Examine and Set
  - — Simple and Register Variables
  - — Structures
  - — Arrays
  - — Pointers
- Screen Oriented Display
  - — C Source
  - — C Intermixed with Disassembled Code

HDS-300 MICROPROCESSOR HARDWARE/ SOFTWARE DEVELOPMENT STATION SUPPORT

An emulator/analyzer is a tool that reduces system development time by simplifying and speeding-up the debugging and testing of hardware and software. The HDS-300 Microprocessor Hardware/Software Development Station is a powerful tool for developing a system based on one of the MC68000, MC68008, MC68010 or MC68020 microprocessors. It also supports development of systems based on the MC68000/HC000, MC6801/03, MC6809, MC6809E and MC68HC11 microprocessors.

An HDS-300 consists of a control station with executive software which is connected by cable to an emulator module — the enclosure for the target system microprocessor or microcomputer chip emulator.

The emulator duplicates in real time all the functions and performance provided by the chip so that it replaces the chip in the target system. This allows control, configuration and evaluations capabilities to be provided which would not otherwise be possible. For connection to the target system, an emulator module has a cable with a mating plug for the socket of the replaced processor.

Although the HDS-300 can be host independent, it would typically be used with SYS1131DVLP as host. Three configurations are possible including standalone (no host), host attached and hosted (terminal at host). Programs are loaded via standard RS-232-C links using standard utilities available on many hosts.

Any computer system running SYSTEM V/68 can host the HDS-300 Microprocessor Hardware/Software Development Station including the following Motorola hosts: SYS1131DVLP, IBM PC-XT or PC-AT with an installed MPCKN2M (PC/68000 Co-Processor Module), or the VME/10 Microcomputer System.

Important HDS-300 capabilities include:

- Real-Time In-Circuit MPU Emulation
- Bus State Monitor/Analyzer
- Source Level Debugger Compatibility
- Emulation Memory
- Local Assembly/Disassembly
- Extensive, Multi-level Help Command

LIMITED VERSAdos C RUN-TIME LIBRARY

Included with SYS1131DVLP is a limited set of C library functions which run under the VERSAdos Operating System and can be used with the M68000 family linker in the creation of load modules for a target environment. Source code is distributed to permit the user to extend/ customize this simple library to a particular target environment. As a further aid to users in the preparation of run time library functions, supplied documentation provides descriptions of the necessary procedures and includes detailed examples.

Run-Time Library functions provided include:

- Memory Allocation
- Program Load Control
- File Control
  - — Open
  - — Close
  - — Create
  - — Lseek
  - — Unlink
- I/O Control
  - — Read
  - — Write
- Terminal and Disk I/O

APPLICATION DOCUMENTATION

Supplementing the comprehensive documentation supplied with SYS1131DVLP which includes User Manuals for all products and an extensive SYSTEM V/68 manual set, an applications guide is included to aid users with their target system development. Major topics treated in the documentation include:

- Writing A Run-Time Library
- How To Prepare ROMable Code
- File Conversion
- Preparing A Target Simulator
- Cross-MPU Considerations

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications	
SYS1131DVLP Hardware		
MCU	MVME131XT VMEmodule 32-Bit Address/Data Monoboard Microcomputer (16.7 MHz MC68020 Microprocessor, 16Kb Physical Cache Memory and MC68881 FPCP)	
Memory	2Mb Dual-Ported RAM with MVMX32bus (MVME204-2)	
Mass Storage Controllers	MVME320A-1 VMEbus Disk Controller MVME350 VMEbus Streaming Tape Controller	
Serial I/O Controller	MVME331 Serial Communications Controller	
Mass Storage	70Mb (formatted) Winchester Disk Drive 655Kb (formatted) Floppy Disk Drive Streaming Tape Cartridge Drive	
I/O Interfaces	8 x RS-232-C programmable to 19.2K baud rear serial I/O connection via MVME705 & MVME707 Transition Modules	
Expansion Slots	Five backplane slots free for user	
Chassis Dimensions Height Width (w/front flange) Depth Power Supply Mounting Shielding Backplane Power Switch	MVME945 Rack Mounting with Tabletop Enclosure Without Enclosure 10.5 in. (266.7 mm) 19 in. (482.6 mm) 19.75 in. (501.7 mm) 400 W, Switching Transverse, mid-chassis, in fan cooled duct Duct mounting provides EMI/RFI shielding A:32/D:32, Accepts 12 6U Eurocards Front panel module with indicator LEDs	With Enclosure 12.20 in. (309.9 mm) 19.0 in. (482.6 mm) 19.75 in. (501.7 mm)
Operating Temperature	5°C to 40°C inlet air temperature, forced air cooling	
Operating Humidity	20% to 80% (non-condensing)	
SYS1131DVLP Software		
Operating System	SYSTEM V/68, Release 2, Version 2.2	
Cross C Compiler	MC68000/MC68008/MC68010 Cross C Compilation System	
Cross Pascal Compiler/ Assembler/Linker	MC68000/MC68008/MC68010 Cross Pascal Compiler	
Toolkit	VERSAdos/SYSTEM V/68 Code Conversion/Utility Toolkit	
Source Level Debugger	Resident C/Assembler Source Level Debugger	
Run-Time Library	Limited VERSAdos C Run-Time Library	

ORDERING INFORMATION

Part Number	Description
SYS1131DVLP	SYS1131UY341 Modular System plus Cross C Compilation System, Cross Pascal Compiler, Resident Assembler/linker, VERSAdos/SYSTEM V/68 Toolkit, Resident C/Assembler Source level Debug, VERSAdos Run-Time Library, HDS-300 Support and comprehensive documentation. Documentation includes: the 4-volume binder set of SYSTEM V/68 Technical Documentation and User's Manuals for the chassis and all included modules and software.

OPTIONAL DEVELOPMENT PRODUCTS

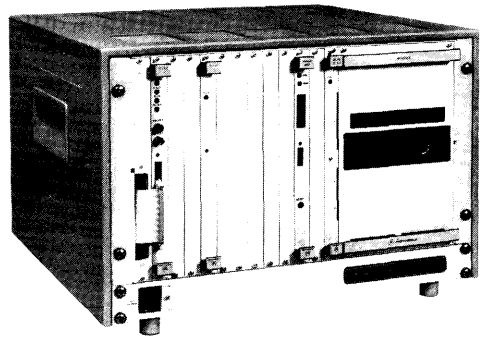
Part Number	Description
M68HDS300	HDS-300 Control Station (90 V–120 V, 50/60 Hz). Includes chassis with built-in power supply, control boards, 5-1/4" floppy disk drive and printer cable. Order appropriate Emulator Module and Software.
M68020HM3-1,-2,-3	MC68020 Emulator Module with 64K RAM (-1), 256K RAM (-2), 1Mb RAM (-3) including personality software and PGA target cable.
M68HDS300SPA	MC68020 System Performance Analyzer Module.
M68010HM3	MC68010 Emulator Module including personality software and Dual In-Line Target Cable.
M68008HM3	MC68008 Emulator Module including personality software and Dual In-Line Target Cable.
M68000HM3	MC68000 Emulator Module including personality software and Dual In-Line Target Cable.
M68N2XBSLD20	MC68020 Source Level Debug for C Compiler under SYSTEM V/68 on SYS1131DVLP/HDS-300. Requires appropriate HDS300 Emulator Module and Control Station.
M68N2XBSLD00	MC68000/MC68008/MC68010 Source Level Debug for C Compiler under SYSTEM V/68 on SYS1131DVLP/HDS-300. Requires appropriate HDS300 Emulator Module and Control Station.

RELATED DOCUMENTATION

Part Number	Description
HB212/D	VMEbus Specification Manual
MVMESYSAM/D	VME System Architecture Guide
MC68020/D	MC68020 Microprocessor User's Manual

SYS1131UY331
SYS1131VY331

VMEbus Modular Systems With SYSTEM V/68™ Or VERSAdos™ 4.5



FEATURES:

- SYSTEM V/68, Release 2 (SYS1131UY331)
- VERSAdos 4.5 (SYS1131VY331)
- MVME204-2 2Mb Dual-Ported DRAM with MVMX32bus Interface*
- MVME320A-1 Disk Controller
- MVME050 System Controller
- 70Mb Winchester
- 655Kb Floppy Disk
- Four RS-232-C Ports (total)
- One Centronics Printer Interface
- MVME131 Microcomputer with
  - — MC68020 Microprocessor
  - — MMB851 Memory Management Unit
  - — Socket for MC68881 Floating Point Co-Processor
  - — Two Sockets for JEDEC 28-pin ROM EPROM Devices
  - — Two Sockets for JEDEC 28-pin RAM ROM Devices
  - — Two Multi-Protocol Serial Ports
  - — A32:D32 VMEbus Interface
  - — MVMX32bus Interface*

SYS1131 MVME945 CHASSIS FEATURES

- 19" Rack Mount Chassis with Tabletop Enclosure
- Transverse 400 W Mid-Chassis Power Supply
- Twelve-Slot, VMEbus-Compatible Cardcages
  - — SYS1131 — A32, D32 Backplane
- Accommodate up to 20 Rear-mounted 80 mm I/O Transition Modules

The greatest advantage of a VMEbus-based system is its modularity. To a core system, add only those software and hardware components required for the application. Then serve a new application by changing only a component or two.

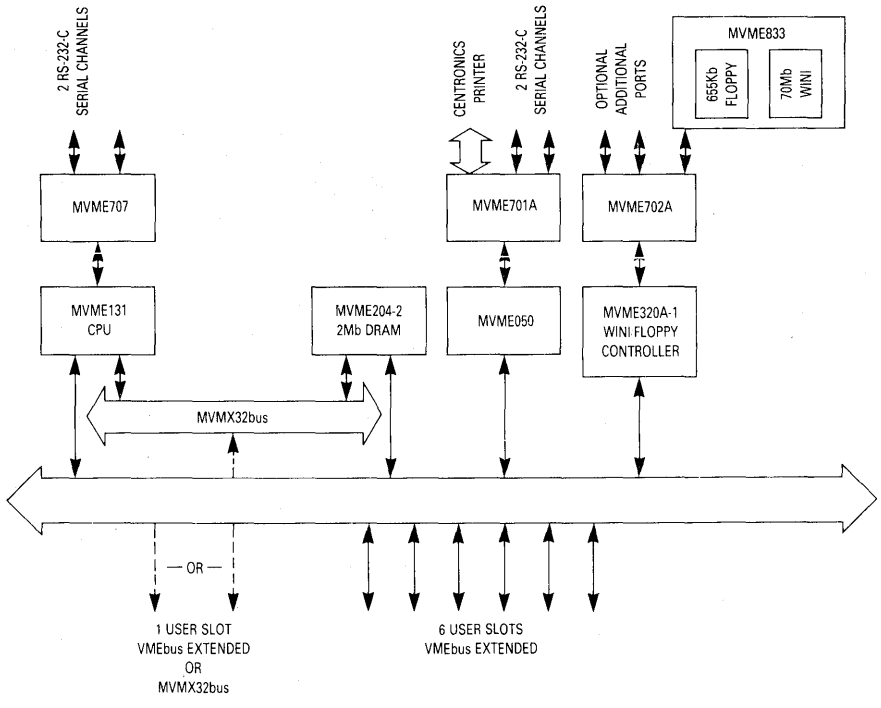
The two VMEbus Modular Systems are versatile core systems configured using standard VMEbus modules. Supplied in 19" rack-mountable chassis with an enclosure for desktop use, the systems have ample room for hardware expansion. Each system offers outstanding performance by virtue of an intelligent mass storage controller, high-performance system RAM and a powerful monoboard microcomputer based on the MC68020 32-Bit Microprocessor. The MVME050 System Controller and the MVME701A I/O Transition Modules provide bus control and accommodate serial and parallel I/O for both systems.

These modular systems provide a high-performance core around which, using standard VMEmodules, a microcomputer system serving a specific application can be configured. Seven additional slots in the VMEbus backplane are available for expansion. Positions are available for adding additional I/O transition modules, if desired.

For software development under, the SYS1131 provides a 4-user environment. Additional users can be accommodated by adding an appropriate VMEmodule such as the MVME331 Intelligent 6-Channel Serial Communication Module. The MVME320A-1 Intelligent Mass-Storage Controller supplied with the core systems enhances the use of system resources by SYSTEM V/68 and VERSAdos.

SYSTEM V/68

Derived from UNIX System V/M68000, the product jointly developed by Motorola Inc. and AT&T, SYSTEM V/68 is the validated port to the M68000 family of microprocessors. It offers a small compact kernel, which provides process scheduling and I/O facilities to all programs. Its powerful command processor, the Shell, provides for interactive control. An extensive set of tools and utilities supports program development, text processing, electronic mail and system-to-system communications. As an environment for developing application software, SYSTEM V/68 is unsurpassed.



SYS1131U331 Hardware Configuration

THE SYSTEM V/68 KERNEL

The SYSTEM V/68 kernel supports a multiuser, multitasking software development and/or application execution environment. The kernel size is typically 150Kb but, depending on target system requirements, may be reduced. SYSTEM V/68 supports a layered or hierarchial file system of arbitrary size. File blocks are 1024 bytes long and a hashed i-node look up feature quickens file access. The kernel is of the swapping variety. Processes may share memory segments through the optionally configurable shared memory support.

The kernel utilizes the MMB851 gate array implementation of the MC68851 Paged Memory Management Unit chip functionality (SYS1131U) to support several SYSTEM V/68 features.

HIGH LEVEL SUPPORT

Integral to SYSTEM V/68 are two categories of high level software tools which provide the means for creating, optimizing and maintaining application software. Other tools supplied with SYSTEM V/68 support text processing and communications between systems running UNIX.

Major components of the first category are the C language compiler "cc," the FORTRAN 77 compiler "f77" and the M68000/M68010/MC68020 assembler "as." Both compilers are supported by the linker/loader "ld" which supports the common object file format: COFF. Supplied also is the symbolic debugger "sdb" which can be used on C source code, FORTRAN and assembler code modules.

Software components in the second category include five editors in support of text processing under SYSTEM V/68 and two utilities supporting text formatting. The full screen editors are "ex" and "vi," "ed" is a line editor while "sed" is stream oriented and noninteractive. The editor "bfs" is a big file scanner similar to "ed."

Under SYSTEM V/68, support is also provided for electronic mail, communications and networking. By allowing a system to be used as a mail box or bulletin board, the mail utility facilitates intrasystem communications between users. Other utilities support communications with mainframe computers and networking utilities allow several computers to be linked together through dedicated links or by dial-up connections so that SYSTEM V/68 users can communicate with users on other systems running UNIX.

VERSAdos 4.5

The software provided with the SYS1131V System is version 4.5 of VERSAdos. This operating system is the standard M68000 operating system for real-time operations. It is also an excellent environment for development of application programs.

VERSAdos was developed by Motorola in 1978 as a real-time multitasking, multiuser Disk Operating System for the MC68000 family of 16-/32-bit microprocessors. Today more than 10,000 users are working with VERSAdos. Just as the MC68000 microprocessor family provides a consistent upgrade path from 8- to 32-bit processors VERSAdos gives a consistent software environment to the user from revision to revision.

VERSAdos is available for all members of the MC68000 family: this not only includes the different processor types, but also drivers for all peripheral chips within the MC68000 family.

VERSAdos

VERSAdos is now becoming the standard Operating System for VME. In addition to Motorola, several other vendors of VME equipment support VERSAdos. Motorola has ported VERSAdos to all its different VME-Processor boards and supplies drivers for all I/O boards.

One of the major advantages of VERSAdos is the fact that it can be used for a target environment running specific application software as well as in a development environment.

The VME system SYS1131VY331 provides a complete software development environment (including editors, assemblers, linkers etc.) but can be used at the same time as a target system, leaving out all development utilities. The software will then be optimized to run the target application specific software.

VERSAdos MEANS MODULARITY

In the same way the user selects VMEmodules to construct modular systems, VERSAdos provides a modular software environment. The ROMable kernel of VERSAdos is called RMS68K and has a minimum size of 25Kb. The size of VERSAdos itself, including file management, is about 100Kb.

The complete VERSAdos package including development utilities and drivers for all Motorola VME boards is about 7Mb in size. This is the package supplied with the VMEsystem SYS1131VY331.

The user can easily configure VERSAdos according to his specific VME board configuration. All the tools to do this are included in the standard VERSAdos package. Writing one's own driver software for third party products is also an easy taskthanks to the well-defined and documented interfaces in the VERSAdos I/O system.

VERSAdos GENERAL CONSTRUCTION

The VERSAdos Operating System permits programs to execute in dynamically-assigned, variable-length memory segments with read/write privileges. Instructions and data are located in separate memory areas, which enable sharing of program code and re-entrant coding practices. A process-to-process facility permits communication between independent programs or nodes of a distributed system.

The heart of the operating system is a real-time executive which provides task services and supports memory allocation. It also allows inter-task communication, provides exception monitor facilities, and handles system interrupts.

The input/output subsystem supports device independence, logical input/output and overlapped computation during physical input/output. New device drivers can be added without impacting the central core portion of the Operating System. Both sequential and random record access are supported by the VERSAdos Operating System.

A powerful file management system supports three types — contiguous, sequential and indexed sequential. Other features include volume and file protection, shared file access, dynamic file allocation and fixed or active protection.

EDITING

Screen-oriented file editing is supported by a terminal-independent editor and an EXORterm-compatible editor. This uses cursor movement and other function keys for fast, easy-to-use text and source file input and maintenance.

ASSEMBLY

Source code programs written in the M68000 assembly language (EXORmacs and VME/10 compatible) can be converted into machine code using the assembler included with the operating system. Macro instructions and structured control statements (while . . . do, etc.) are supported by the assembler. Code for the MC68010 and MC68020 can also be generated.

LINKAGE EDITOR

Using this utility, relocatable modules can be linked together to form a load module ready to execute in the system memory.

A software package is available that outputs code via one of the RS-232-C ports to a Data I/O PROM programmer.

This package is ordered separately (see ordering information).

OPTIONAL HIGH LEVEL LANGUAGES

MOTOROLA PASCAL

This compiler is a category 1 software product and is fully maintained and supported by Motorola. Software updating is available on an annual subscription basis. The language is standard Pascal, as defined by Wirth, with Motorola extensions:

- Non-decimal integers
- Address specifications for variables
- Alphanumeric labels for GO TO statements
- EXIT statement
- HALT statement
- OTHERWISE clause for case statements
- String data type

— Structured function values
— External procedures and functions
— Runtime file assignment

The compiler may consist of two or three phases. Phase 1 processes a source program and produces a compilation listing and error messages, as well as an intermediate code file. Optionally, Phase 1.5 may be invoked in order to optimize the intermediate code, i.e. to reduce the size of the code file. Phase 2 creates a relocatable object file and a listing.

MOTOROLA FORTRAN

This compiler is a subset of ANSI 1977 standard with Motorola extensions. It is a category 1 software product; support, maintenance, and customer updates are supplied by Motorola.

Motorola extensions are:
— Integer data types (2 and 4 byte)
— Real data types (4 and 8 byte)
— Hexadecimal and Hollerith constants
— Bit manipulation functions
— Double precision intrinsic functions
— Structured programming statements and other features.

ABSOFT FORTRAN

This third party software product can be ordered from Motorola. Support and maintenance are provided by Absoft Corporation, Royal Oak, Minnesota, U.S.A. According to Absoft, the compiler speed is in excess of 1000 lines/min, the language adheres to ANSI 77 standard and many features of the new proposed standard (block structures) generate relocatable and reentrant object code. An intrinsic function library is available for user assembler programs. Includes symbolic debugger with single step capability, dynamic runtime loader and linker, supports external procedures written in assembler and provides for incremental compilation of subprogram libraries.

USER DOCUMENTATION

The following manuals are supplied with the system:

A 2-volume binder set, containing the following documents:
— Overview Manual
— CRT Text Editor User's Manual
— Command Summary
— System Facilities Manual
— System Messages
— Macro Assembler User's Guide
— Linkage Editor User's Guide
— SYSGEN User's Manual
— RMS68K User's Manual
— SYMbug Reference Manual
— Data Management Services and Program
— Loader User's Manual
— Guide to Writing Device Drivers
— VME101 System VERSAdos Hardware and Software Configuration Manual

A set of hardware manuals is also supplied with the system.

SYSTEMS CENTRAL PROCESSING UNIT

To provide the computing power needed to take full advantage of the vast capabilities of both UNIX V and VERSAdos 4.5, a monoboard microcomputer based on a fast microprocessor of the Motorola MC68000 family is used as the central processing unit by the two core systems.

Unsurpassed in its processing power, the intelligent master of these core systems is an MVME131 VMEbus Microprocessor Module featuring the state-of-the-art, full 32-bit MC68020 Microprocessor and including also MVMX32bus, a high-speed dedicated interface to the MVME204-2 2Mb Dynamic RAM, provision for adding the MC68881 Floating Point Coprocessor and the functionality of the MC68851 Paged Memory Management Unit implemented with gate array technology.

The SYS1131 system also includes an MVME707 RS-232 Serial Port Distribution Module which translates the TTL levels output by the two MVME131 ports to provide full support of RS-232-C asynchronous serial communications and which provides also two DB-25 connectors for serial I/O cables. This module is a double-high, half deep board mounted on the upper and lower rear chassis rails and connected to the MVME131 by a 50-conductor ribbon cable.

HARDWARE SUPPORT FOR MEMORY MANAGEMENT

Since SYSTEM V/68 and VERSAdos utilize memory management techniques in the manner of mainframe operating systems, hardware support for the translation of logical addresses into physical addresses and for managing the separation of user and supervisor memory spaces is essential. For the SYS1131 core system, memory management support is provided by the MMB851 gate array implementation of the MC68851 Paged Memory Management Unit function.

CORE SYSTEM MAIN MEMORY

A 2Mb DRAM with parity dual-ported between VMEbus and MVMX32bus — MVME204-2 — is supplied with both core systems. Up to four additional MVME204-2's can be added to this interface to provide a core system with 8Mb of high speed memory. A base address for a module can be set on boundaries separated by the module population.

Tailored to the MC68020 Microprocessor and specifically designed to enhance the performance of the MVME131 VMEmodule 32-Bit Monoboard Microcomputer, the MVME204-2/MVMX32bus interface allows concurrent transfers on VMEbus with MVMX32bus transfers between the MVME131 and MVME204-2. Implemented using high density 256K x 1-bit dynamic RAM devices, the MVME204-2 has parity generation and detection circuitry which, together with accessible control and status registers, can be used for error detection and memory diagnostics.

System memory capacity can be expanded by adding DRAM modules to VMEbus space by using the MVME204-1, or -2 to add 1Mb or 2Mb, respectively.

SYSTEM CONTROLLER — I/O INTERFACES

Certain control functions essential to a multiprocessor, bus-based system are required once and need not be supplied by each module. These include system reset, priority bus arbitration, bus timeout monitoring and the generation of system, serial and time-of-day clocks. The MVME050 System Controller generates these functions for optional use and also provides a Centronics-compatible parallel printer port and two multiprotocol RS-232-C serial ports.

Both systems also utilize an MVME701A I/O Transition Module to provide the connectors for the serial and parallel ports of the MVME050 and to accommodate the optional backup battery for the time-of-day clock. The module is connected by 64-conductor ribbon cable to MVME050 connector P2.

The TTL levels output by the dual multiprotocol serial ports of the MVME131 processor module used by SYS1131 are translated to RS-232-C levels by means of an MVME707 I/O Transition Module which also provides the appropriate connectors for external equipment observing this communications protocol. A 50-conductor ribbon cable is used for the MVME131-to-MVME707 connection.

The rear-mounting MVME701A and MVME707 modules are two of several which permit I/O connections to be made at the back of a rack if required by the particular application or from the front in a development environment.

MASS STORAGE DEVICES CONTROLLER

To control the provided floppy disk drive and Winchester disk drive, both VMEbus Modular Core Systems utilize the MVME320A-1 VMEbus Disk Controller. Although not supported by either core system, this VME-module has the capability of controlling mixed 5-1/4" and 8" drives and can accommodate two 5-1/4" Winchester hard disk drives and two 5-1/4" floppy disk drives or up to four floppy disk drives.

So that commonly available rather than custom cables can be used for the connections between the controller and the disk drive connectors, an intermediate transition board, the MVME702A Disk Interface Module is used. This module re-orders MVME320A-1 signals to the requirements of the standard 34-pin and 50-pin connector interfaces used by most 5-1/4" and 8" hard disk and floppy disk drives. It also buffers signals for 5-1/4" drives and provides a means of terminating unused signals from the controller.

Because the controller provides high-performance direct memory access (DMA) data channels between core system memory and the floppy and hard disk storage, the core systems can be used in applications having intensive real-time disk I/O or multiprocessing structures designed to reduce VMEbus traffic and increase system throughput. Other MVME320A-1 features include:

- Supports Serial Data Rates to 5Mbps
- Supports Standard IBM Formats and MFM Recording
- 16-Bit Cyclic Redundancy Check
- 32-Bit Error Checking and Correction

MASS STORAGE DEVICES

Mass storage capacity for these storage systems is provided by a floppy disk drive and a hard disk drive.

The floppy disk is a 5-1/4" slim line drive for soft-sectored dual-sided dual-density, 96 TPI diskettes each with a capacity of 655Kb of formatted data. Features include a track density of 96 T.P.I., a recording density of 5922 B.P.I. (double density) and a data transfer rate of 250Kb per second.

The hard disk uses reliable Winchester technology and provides a capacity of 70Mb, formatted. Features include 32 sectors per track, 256 bytes per sector and a data transfer rate of 5Mb per second.

USER INTERFACE

Starting with a chassis suitable for nourishing and protecting the inhabitants, a VMEbus-based system can be constructed from the wide variety of commercially available modules to serve almost any application. The core of this system must simultaneously offer configuration flexibility and ease of operation.

Each core system is housed in a MVME945 chassis assembly, the desktop decorative cover of which can be removed so that the system can be mounted in a standard 19-inch rack, if the application requires. The front of the chassis houses a card cage with backplane which accepts twelve standard double high Eurocard VMEmodules. On the right, the floppy and hard disk drives supplied with a core system are housed. On the left, a power switch module is attached.

On the left in the chassis rear, the two 80 mm I/O Transition Modules supplied with their respective core systems are attached by screws to the upper and lower rails. The remaining space can be used for additional transition modules.

A 400 watt power supply is mounted in a fan-cooled air duct assembly located in the upper half of the chassis to the rear of the VMEmodule backplane. Additional cooling sufficient for an expanded system is provided in the chassis.

By virtue of the high intelligence and integration levels of their modules, both core systems are able to present to the user an uncluttered front which offers the essential basic controls and system status indications and, at the same time, the means of configuring the hardware to provide certain system level functions.

Chassis Front Panel

ON/OFF Power Switch, Rocker

MVME050 Front Panel

RUN Indicator, Green LED
FAIL Indicator, Red LED
Dot Matrix Display, 2 Hex Digits
Reset Switch, Spring-Loaded Pushbutton
8-Position DIP Switch

MVME320A-1 Front Panel

FAIL Indicator, Red LED
Two Slots for ribbon cable entry/exit

MVME131 Front Panel — System 1131UY

FAIL Indicator, Red LED
HALT Indicator, Red LED
RUN Indicator, Green LED
SCON Indicator, Green LED
Reset Switch, spring-loaded pushbutton
Abort Switch, spring-loaded pushbutton
8-Position DIP Switch
50-Pin Connector

MVME204-2 Front Panel — System 1131UY

Power Indicator, Green LED

EXPANSION FOR ADDITIONAL USERS

The backplane supplied with the core system chassis can accommodate additional VMEmodules. Access to the unused slots is gained by removing as many dummy module front panels as required by removing the screws fastening them to the upper and lower rails of the chassis front.

The four serial interfaces provided with the core systems for terminals or other I/O devices can be expanded by six RS-232-C or RS-422-B ports by installing the MVME331 Intelligent Communications Controller Module in the chassis backplane. A companion I/O Transition Module, the MVME705 which supplies the DB-25 connectors and the transmitters/receivers required to support the two communication protocols is installed by screwing the module front panel to the top and bottom rear rails of the chassis. Connection is made between the two modules using a 64-conductor ribbon cable.

Memory for either system can be supplemented in increments of 1 or 2 megabytes by adding the MVME204-1 or -2.

Extensive documentation for the hardware and software is supplied with the core systems which details the procedure for reconfiguring and for generating a new kernel to accommodate the addition of any new modules.

Documentation supplied with the core systems describes in detail the process of generating a new kernel for use with an expanded system. Hardware and module documentation supplied with the core systems and expansion modules details the process of configuring a system to comply with the requirements of various applications.

VMEbus

Rapidly becoming the leading bus standard in U.S., European and other world markets, VMEbus is a 8-/16-/32-bit interconnect that has a master/slave asynchronous, non-multiplexed data transfer structure, seven levels of priority interrupt, four levels of data bus arbitration, multiple master support, fault detection and control, and special transfer cycles.

The VMEbus functional structure consists of backplane interface logic, four groups of signal lines called buses each having its own collection of functional modules any number of which can be grouped on a board and which communicate with the other functional modules of a bus via onboard or backplane signal lines.

The VMEbus mechanical structure comprises subracks, backplanes, front panels, plug-in boards and other elements all of which conform to relevant IEC Eurocard specifications to insure that all components fit together properly.

These features combined with reliability, mechanical integrity, 20 megabytes-per-second performance and the fact that it is nonproprietary have resulted in almost complete acceptance of VMEbus by the electronics industry. Currently over 200 vendors are manufacturing over 1000 VMEsystem-compatible products, assuring for the expansion of a core system, the widest selection of functions from the greatest number of sources so that the broadest range of applications can be served.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

MVME1131UY331	
MCU	MVME131 VMEmodule 32-Bit Address/Data Monoboard Microcomputer (12.5 MHz MC68020 Microprocessor)
Memory	2Mb Dual-Ported RAM with MVMX32bus (MVME204-2)
Mass Storage Controller	MVME320A-1 VMEbus Disk Controller
Mass Storage	70Mb (formatted) Winchester Disk Drive 655Kb (formatted) Floppy Disk Drive
I/O Interfaces	4 x RS-232-C programmable to 19.2K baud 1 x Centronics-Compatible Printer Interface Rear serial/parallel I/O connections via MVME701A & MVME707 Transition Modules
Expansion Slots	12-slot Backplane, 7 slots free for user
Power Supply	400 W (115 Vac)
Operating Temperature	5° to 40°
Operating Humidity	20% to 80% (non-condensing)

ORDERING INFORMATION

Part Number	Description
SYS1131UY331	VMEbus Modular System with SYSTEM V/68, includes: MVME131 VMEmodule Monoboard Microcomputer MVME204-2 2Mb Dual-Ported DRAM MVME320A-1 VMEbus Disk Drive Controller MVME050 System Controller MVME701A Transition Module MVME 707 Transition Module MVME833 Plug-In Mass Storage Unit with — 70Mb Winchester Disk Drive — 655Kb Floppy Disk Drive MVME945 VMEbus Chassis Assembly with — 32-Bit VMEbus Backplane — 400 W Power Supply User's Manuals for SYSTEM V/68, chassis and all modules included
SYS1131VY331	VMEbus Modular System with SYSTEM V/68 includes: MVME131 VMEmodule Monoboard Microcomputer MVME204-2 2Mb Dual-Ported DRAM MVME320A-1 VMEbus Disk Drive Controller MVME050 System Controller MVME701A Transition Module MVME707 Transition Module MVME833 Plug-In Mass Storage Module with — 70Mb Winchester Disk Drive — 655Kb Floppy Disk Drive MVME945 VMEbus Chassis Assembly with — 32-Bit VMEbus Backplane — 400 W Power Supply User's Manuals for VERSAdos, chassis and all modules included

OPTIONS

Part Number	Description
MVME204-1,-2	1Mb/2Mb Interleaved Dual-Ported DRAM Modules
MVME214	A32/D32 Static RAM/ROM Memory Module
MVME331,332	6/8 Channel Intelligent Communications Controllers
MVME705	I/O Transition Board for MVME331
MVME350	Intelligent Streaming Tape Controller
MVME390A	Color Graphics Controller Module

RELATED DOCUMENTATION

Part Number	Description
HB212/D	VMEbus Specification
MC68020/D	MC68020 Microprocessor User's Manual

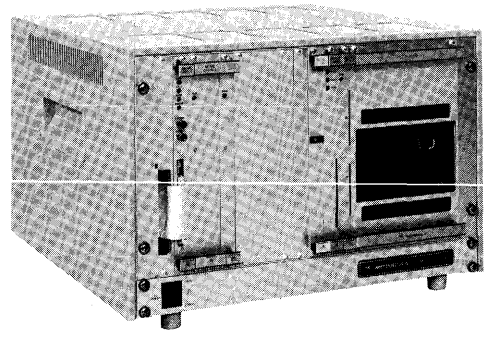
VMEbus Modular System With SYSTEM V/68™

FEATURES:

- SYSTEM V/68, Release 2
- MVME204-2 2Mb Dual-Ported DRAM with MVMX32bus Interface
- MVME320A-1 Disk Controller
- MVME350 Streaming Tape Controller
- MVME331 6-Channel Serial Communications Controiier
- 70Mb Winchester Disk Drive
- 655Kb Floppy Disk Drive
- QIC-02 Cartridge Streaming Tape Drive
- Eight RS-232-C Ports (Total)
- MVME131XT Microcomputer with
  - — MC68020 Microprocessor (16.67 MHz)
  - — MC68881 Floating Point Coprocessor (16.67 MHz)
  - — MMB851 Memory Management Unit
  - — 16Kb Instruction/Data Cache
  - — Two Sockets for JEDEC 28-Pin ROM/EPROM Devices
  - — Two Sockets for JEDEC 28-Pin RAM/ROM Devices
  - — Two Multi-protocol Serial Ports
  - — A32:D32 VMEbus Interface
  - — MVMX32bus Interface*
- System Level Diagnostics

MVME945 CHASSIS FEATURES

- 19" Rack Mount Chassis with Tabletop Enclosure
- Transverse 400 W Mid-Chassis Power Supply
- 12-Slot, VMEbus-Compatible Card Cage
  - — A32:D32 Backplane
- Accommodates up to 20 Rear-mounted 80 mm I/O Transition Modules



The greatest advantage of a VMEbus-based system is its modularity. To a core system, add only those software and hardware components required for the application. Then serve a new application by changing only a component or two.

The SYS1131UY341 with SYSTEM V/68, is a versatile core system configured using standard VMEbus modules. Supplied in 19" rack-mountable chassis with an enclosure for desktop use, the system has ample room for hardware expansion. This system offers outstanding performance by virtue of an intelligent mass storage controller, intelligent streaming tape backup, high-performance system RAM and a powerful monoboard microcomputer with cache based on the MC68020 32-Bit Microprocessor. The MVME331 Serial Communications Controller and the MVME705 I/O Transition Module provides 6-channel RS-232-C serial I/O.

This modular system provides a high-performance core around which, using standard VMEmodules, a microcomputer system serving a specific application can be configured. Five additional slots in the VMEbus backplane are available for expansion. Positions are available for adding additional I/O transition modules, if desired.

For software development under the multiuser, multitasking system — SYSTEM V/68, SYS1131UY341 provides a powerful 8-user environment. Additional users can be accommodated by adding an appropriate VMEmodule such as the MVME331 Intelligent 6-Channel Serial Communication Module. The MVME320A-1 Intelligent Mass-Storage Controller supplied with the core systems enhances the use of system resources by SYSTEM V/68.

SYSTEM V/68

Derived from UNIX System V/M68000, the product jointly developed by Motorola Inc. and AT&T, SYSTEM V/68 is the validated port to the M68000 family of microprocessors. It offers a small compact kernel, which provides process scheduling and I/O facilities to all programs. Its powerful command processor, the Shell, provides for interactive control. An extensive set of tools and utilities supports program development, text processing, electronic mail and system-to-system communications. As an environment for developing application software, SYSTEM V/68 is unsurpassed.

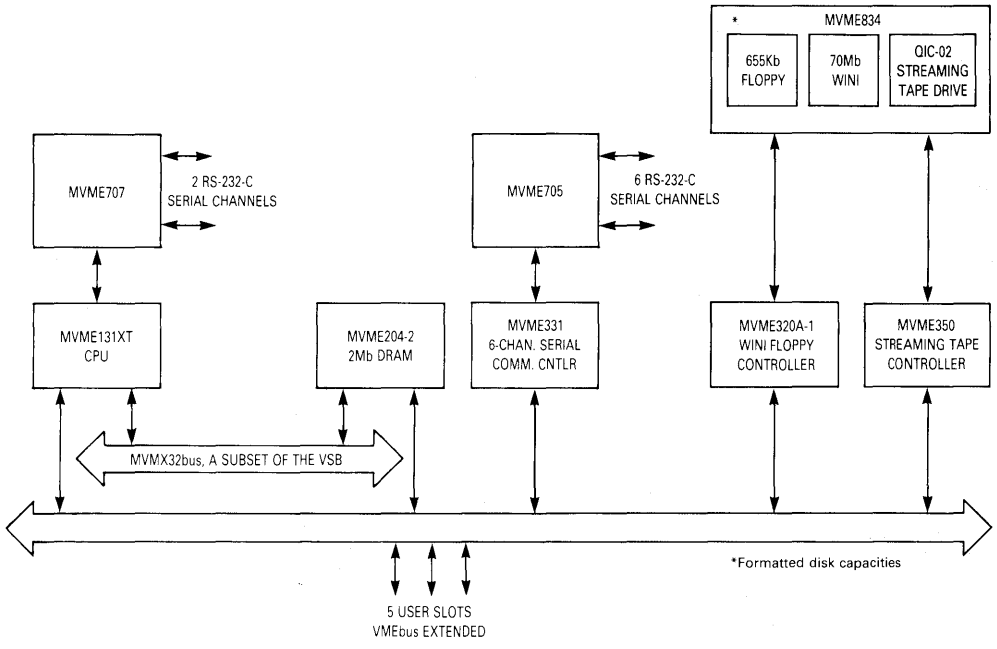


Figure 1. SYS1131UY341 Hardware Configuration

THE SYSTEM V/68 KERNEL

The SYSTEM V/68 kernel supports a multiuser, multitasking software development and/or application execution environment. The kernel size is typically 125Kb but, depending on target system requirements, may be reduced. SYSTEM V/68 supports a layered or hierarchial file system of arbitrary size. Logical file blocks are 1024 bytes long and a hashed i-node look up feature quickens file access. The kernel is of the swapping variety. Processes may share memory segments through the optionally configurable shared memory support.

The kernel utilizes the MMB851 gate array implementation of the MC68851 Paged Memory Management Unit chip functionality to support several SYSTEM V/68 features.

HIGH LEVEL SUPPORT

Integral to SYSTEM V/68 are two categories of high level software tools which provide the means for creating, optimizing and maintaining application software. Other tools supplied with SYSTEM V/68 support text processing and communications between systems running UNIX.

Major components of the first category are the C language compiler "cc," the FORTRAN 77 compiler "f77" and the M68000/M68010/MC68020 assembler "as." Both compilers are supported by the linker/loader "ld" which supports the common object file format: COFF. Supplied also is the symbolic debugger "sdb" which can be used on C source code, FORTRAN and assembler code modules.

Software components in the second category include five editors in support of text processing under SYSTEM V/68 and two utilities supporting text formatting. The full screen editors are "ex" and "vi," "ed" is a line editor while "sed" is stream oriented and noninteractive. The editor "bfs" is a big file scanner similar to "ed."

Under SYSTEM V/68, support is also provided for electronic mail, communications and networking. By allowing a system to be used as a mail box or bulletin board, the mail utility facilitates intrasystem communications between users. Other utilities support communications with mainframe computers and networking utilities allow several computers to be linked together through dedicated links or by dial-up connections so that SYSTEM V/68 users can communicate with users on other systems running UNIX.

SYSTEMS CENTRAL PROCESSING UNIT

To provide the computing power needed to take full advantage of the vast capabilities of UNIX V, in the form of the SYSTEM V/68 Multiuser, Multitasking Operating System, a monoboard microcomputer based on the MC68020 microprocessor is used as the central processing unit by the SYS1131UY341.

Unsurpassed in its processing power, the intelligent master of this core system is an MVME131XT VMEbus Microprocessor Module featuring the state-of-the-art, full 32-bit MC68020 Microprocessor and including also MVMX32bus, a high-speed dedicated interface to the MVME204-2 2Mb Dynamic RAM, the MC68881 Floating

Point Coprocessor, a functionality subset of the MC68851 Paged Memory Management Unit implemented with gate array technology, and the Cache Accelerator, a plug-in cache memory unit designed for use with the MVME131.

The Cache Accelerator is a small, fast, static RAM system that provides concurrent, local storage of the microprocessor's VMEbus and VMX32bus memory transactions. The Cache Accelerator was designed for users wanting to accelerate the performance of their MC68020-based systems without having to modify existing programs. This software transparency feature is achieved through the utilization of a physical address cache memory structure coupled with a VME Bus Monitor.

The SYS1131UY341 system also includes an MVME707 RS-232 Serial Port Distribution Module which translates the TTL levels output by the two MVME131 ports to provide full support of RS-232-C asynchronous serial communications and which also provides two DB-25 connectors for serial I/O cables. This module is a double-high, half deep board mounted on the upper and lower rear chassis rails and connected to the MVME131XT by a 50-conductor ribbon cable.

The 130bug resident debug package (refer to Motorola publication MVME130BUG) is the firmware package designed for use with the VME131 microprocessor. 130bug provides a powerful system debugging tool for VME-module systems. This firmware features a one line assembler/disassembler, provisions for up/downline load, and disk bootstrap commands. 130bug is available as an EPROM-based resident package. It is plugged into IC sockets U28 and U31 on the VME131.

HARDWARE SUPPORT FOR MEMORY MANAGEMENT

Since SYSTEM V/68 utilizes memory management techniques in the manner of mainframe operating systems, hardware support for the translation of logical addresses into physical addresses and for managing the separation of user and supervisor memory spaces is essential. For the SYS1131UY341 core system, memory management support is provided by the MMB851 gate array implementation of the MC68851 Paged Memory Management Unit function.

CORE SYSTEM MAIN MEMORY

A 2Mb DRAM with parity dual-ported between VMEbus and MVMX32bus — MVME204-2 — is supplied with this core system. Up to four additional MVME204-2's can be added to this interface to provide a core system with 8Mb of high-speed memory. The base address for a module can be set on boundaries separated by the module's population.

Tailored to the MC68020 Microprocessor and specifically designed to enhance the performance of the MVME131XT VMEmodule 32-Bit Monoboard Microcomputer, the MVME204-2/MVMX32bus interface allows concurrent transfers on VMEbus with MVMX32bus transfers between the MVME131XT and MVME204-2. Implemented using high density 256K x 1-bit dynamic RAM devices, the MVME204-2 has parity generation and detection circuitry which, together with accessible control and status registers, can be used for error detection and memory diagnostics.

System memory capacity can be expanded by adding DRAM modules to VMEbus space. Use the MVME204-1, or the -2 to add 1Mb or 2Mb, respectively.

SERIAL COMMUNICATIONS CONTROLLER

The MVME331 Intelligent Communication Controller (ICC) supports six asynchronous serial communication channels at a maximum simultaneous speed of 9600 baud full duplex. The module contains a complete 16-bit microcomputer consisting of an MC68010 MPU, 128Kb RAM and firmware in two 16Kb EPROMs. This microcomputer controls three Z8530 Serial Communication Controller devices, pre-processes and transfers data between the VME system memory and the serial ports.

The transfer of commands, status messages and data packets is performed through a pipeline structure in VME system memory, which is shared between the operating system and the MVME331 firmware. This transfer follows the Buffered Pipe Protocol (BPP), a Motorola standard for processor to processor communication.

Each of the six serial channels can be configured to conform to the RS-232-C or RS-422-B standard. This is implemented on a separate Transceiver Module (MVME705 or MVME706), a double-high, half deep board mounted on the upper and lower rear chassis rails and connected to the MVME331 by a 50-conductor ribbon cable.

MASS STORAGE DEVICE CONTROLLERS

THE MVME320A-1 VMEbus DISC CONTROLLER

To control the provided floppy disk drive and Winchester disk drive, the SYS1131UY341 utilizes the MVME320A-1 VMEbus Disk Controller. Although not supported, this VMEmodule has the capability of controlling mixed 5-1/4" and 8" drives and can accommodate two 5-1/4" Winchester hard disk drives and two 5-1/4" floppy disk drives or up to four floppy disk drives.

Because the controller provides high-performance direct memory access (DMA) data channels between core system memory and the floppy and hard disk storage, the core system can be used in applications having intensive real-time disk I/O or multiprocessing structures designed to reduce VMEbus traffic and increase system throughput. Other MVME320A-1 features include:

- Supports Serial Data Rates to 5Mbps
- Supports Standard IBM Formats and MFM Recording
- 16-Bit Cyclic Redundancy Check
- 32-Bit Error Checking and Correction

THE MVME350 STREAMING TAPE CONTROLLER

A VMEmodule Intelligent Peripheral Controller (IPC), MVME350 provides control of one QIC-02 compatible, 1/4 inch streaming tape drive.

The 16Kb of static RAM onboard MVME350 provides working storage for the MC68010 vector table and stack frames. It also serves as buffer storage between the QIC-02 interface and the VMEbus. Optimum performance for typical applications is offered since the MC68010, at

10 MHz, performs zero wait state accesses of the SRAM space.

The VMEbus interface is designed to provide uniformity and compatibility among MVMEmodule drivers running under the SYSTEM V/68 as well as ease the development, by a system integrator, of software to communicate with the controller module. Controller firmware based on a multitasking kernel handles all input/output interrupts and data transfers in order to minimize impact on system throughput. Macro commands are retrieved from a pipeline structure in system memory which is shared by the MVME131XT microcomputer, and the tape controller. Status information is returned to a host through another pipeline. Exchange of command and status packets follows the Motorola buffered pipe protocol for interprocessor communication.

The following SYSTEM V/68 backup and restore utilities will be supported: finc (fast incremental backup), frec (recover file systems from a backup tape), volcopy (copy file system), cpio (copy file archives in and out) and tar (tape file archiver). In addition, the system call M350CTL will be supported to provide an application program with the means of obtaining complete control over a streaming tape device. M350CTL will support:

- Rewind Tape
- Erase Tape
- Retension Tape
- Write Filemark
- Read Filemark
- Set DMA Buffer Size
- Get DMA Buffer Size
- Byte Swapping

MASS STORAGE DEVICES

Mass storage capacity for the SYS1131UY341 system is provided by a floppy disk drive, a hard disk drive and a streaming tape.

The floppy disk is a 5-1/4" slim line drive for soft-sectored dual-sided dual-density, 96 TPI diskettes each with a capacity of 655Kb of formatted data. Features include a track density of 96 T.P.I., a recording density of 5922 B.P.I. (double density) and a data transfer rate of 250Kb per second.

The hard disk uses reliable Winchester technology and provides a capacity of 70Mb, formatted. Features include 32 sectors per track, 256 bytes per sector and a data transfer rate of 5Mb per second.

The streaming tape drive consists of a drive unit in a standard 5-1/4" half-height enclosure and a formatter/controller module which fastens to the enclosure and forms a full-height 5-1/4" product. By using the industry standard DC600 tape cartridge, a full 60Mb of tape backup can be provided.

USER INTERFACE

Starting with a suitable chassis a VMEbus-based system can be constructed from the wide variety of commercially available modules to serve almost any application. The core of this system must simultaneously offer configuration flexibility and ease of operation.

The SYS1131UY341 is housed in a MVME945 chassis assembly, the desktop decorative cover of which can be removed so that the system can be mounted in a standard 19-inch rack, if the application requires. The front of the chassis houses a card cage with backplane which accepts twelve standard double high Eurocard VMEmodules. On the right, the floppy hard disk and streaming tape drives supplied with the core system are housed in the MVME834. On the left, a power switch module is attached.

On the left in the chassis rear, attached by screws to the upper and lower rails, are the two 80 mm I/O Transition Modules supplied with the SYS1131UY341. The remaining space can be used for additional transition modules.

A 400 watt power supply is mounted in a fan-cooled air duct assembly located in the upper half of the chassis to the rear of the VMEmodule backplane. Additional cooling sufficient for an expanded system is provided in the chassis.

By virtue of the high intelligence and integration levels of its modules, the SYS1131UY341 is able to present to the user an uncluttered front which offers the essential basic controls and system status indications and, at the same time, the means of configuring the hardware to provide certain system level functions.

MVME131XT Front Panel —

FAIL Indicator, Red LED
HALT Indicator, Red LED
RUN Indicator, Green LED
SCON Indicator, Green LED
Reset Switch, spring-loaded pushbutton
Abort Switch, spring-loaded pushbutton
8-Position DIP Switch
50-Pin Connector
RUN Indicator, cache module, Green LED

MVME204-2 Front Panel —

Power Indicator, Green LED

MVME320A-1 Front Panel

FAIL Indicator, Red LED
Two Slots for ribbon cable entry/exit

MVME350 Front Panel

FAIL Indicator, Red LED
HALT Indicator, Red LED
RUN Indicator, Green LED

EXPANSION FOR ADDITIONAL USERS

The backplane supplied with the SYS1131UY341 chassis can accommodate additional VMEmodules. Access to the unused slots is gained by removing the screws fastening dummy module front panels to the upper and lower rails of the chassis front.

The eight serial interfaces provided with this core system for terminals or other I/O devices can be expanded by six RS-232-C or RS-422-B ports by installing the MVME331 Intelligent Communications Controller Module in the chassis backplane. A companion I/O Transition Module, the MVME705 which supplies the DB-25 connectors and the transmitters/receivers required to support the two communication protocols is installed by screwing the module front panel to the top and bottom rear rails of the chassis. Connection is made between the two modules using a 64-conductor ribbon cable.

Memory can be supplemented in increments of 1 or 2 megabytes by adding the MVME204-1 or -2.

Extensive documentation for the hardware and SYSTEM V/68 is supplied with the SYS1131UY341. Documentation supplied for SYSTEM V/68 describes in detail the process of generating a new kernel for use with an expanded system. Hardware and module documentation supplied with the core system and expansion modules details the process of configuring a system to comply with the requirements of various applications.

SYS1131UY341 SYSTEM TEST

The SYS1131UY341 System Test is a comprehensive set of powerful system level tests and fault isolation diagnostics. The software executes on this VMEbus Modular System with SYSTEM V/68 or on other VMEbus hosts and can be a versatile tool for system operators, as well as engineering, manufacturing and field service personnel. It can be used for new design verification, for system product verification or for performance measurement and fault isolation in installed systems. It can also serve the system integrator as the basic test structure in which tests for added modules can be included as a means of offering a system test facility with the custom configuration.

SYS1131UY341 System Test features include:

- Simple Default Mode — Test of the as-shipped SYS1131UY341 VMEbus Modular System
- Flexible, easy-to-use Operator Interface Accommodates Varying Needs:
  - — Change complement of tests for a new system configuration
  - — Change sequence of tests to provide a new test environment
  - — Change internal execution of a test to meet specific requirements
- Elaborate Test Environment Controls for Fault Isolation
- Timer Functions to Support System Performance Measurement
- Error Logger and I/O Re-direction for Test Record Keeping
- Extensive Manual Debugger for Fault Isolation and Specific Function Checks
- Target Environment Simulated by Tasks Executing Under a Real-Time, Multitasking Kernel
- C Language Source Available for Straightforward Porting

To provide the essential synchronizing functions and task control utilities required for the simulation of a target operating system, SYS1131UY341 System Test utilizes a real-time, multitasking kernel. This allows multiple test tasks to execute concurrently creating a realistic on-line simulation in a controlled environment. Use of a real-time, multitasking kernel also facilitated the creation of utilities for dynamically tailoring test conditions to match changing hardware configurations and for pinpointing a specific fault or measurement state. The addition of a multi-faceted debug utility resulted in a system test capability that isolates faults to the replaceable component level — and beyond.

VMEbus

Rapidly becoming the leading bus standard in U.S., European and other world markets, VMEbus is a 8-/16-/32-bit interconnect that has a master/slave asynchronous, non-multiplexed data transfer structure, seven levels of priority interrupt, four levels of data bus arbitration, multiple master support, fault detection and control, and special transfer cycles.

The VMEbus functional structure consists of backplane interface logic, four groups of signal lines called buses each having its own collection of functional modules any number of which can be grouped on a board and which communicate with the other functional modules of a bus via onboard or backplane signal lines.

The VMEbus mechanical structure comprises subracks, backplanes, front panels, plug-in boards and other elements all of which conform to relevant IEC Eurocard specifications to insure that all components fit together properly.

These features combined with reliability, mechanical integrity, 20 megabytes-per-second performance and the fact that it is nonproprietary have resulted in almost complete acceptance of VMEbus by the electronics industry. Currently over 200 vendors are manufacturing over 1000 VMEsystem-compatible products, assuring for the expansion of a core system, the widest selection of functions from the greatest number of sources so that the broadest range of applications can be served.

ORDERING INFORMATION

Part Number	Description
SYS1131UY341	VMEbus Modular System with SYSTEM V/68, includes: MVME131XT VMEmodule Monoboard Microcomputer with cache MVME204-2 2Mb Dual-Ported DRAM MVME320A-1 VMEbus Disk Drive Controller MVME331 Serial Communications Controller MVME350 Streaming Tape Controller MVME705 Transition Module MVME 707 Transition Module MVME834 Plug-In Mass Storage Module with — 70Mb Winchester Disk Drive — 655Kb Floppy Disk Drive — Streaming Cartridge Tape Drive MVME945 VMEbus Chassis Assembly with — 32-Bit VMEbus Backplane — 400 W Power Supply User's Manuals for SYSTEM V/68, chassis and all modules included

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

MVME1131UY341	
MCU	MVME131XT VMEmodule 32-Bit Address/Data Monoboard Microcomputer (16.67 MHz MC68020 Microprocessor, 16Kb Physical Cache Memory and MC68881 FPCP)
Memory	2Mb Dual-Ported RAM with MVMX32bus (MVME204-2)
Mass Storage Controllers	MVME320A-1 VMEbus Disk Controller MVME350 VMEbus Streaming Tape Controller
Serial I/O Controller	MVME331 Serial Communications Controller
Mass Storage	70Mb (formatted) Winchester Disk Drive 655Kb (formatted) Floppy Disk Drive Streaming Cartridge Tape Drive
I/O Interfaces	8 x RS-232-C programmable to 19.2K baud Rear serial I/O connections via MVME705 & MVME707 Transition Modules
Expansion Slots	12-slot Backplane, 5 slots free for user
Power Supply	400 W (115 Vac)
Operating Temperature	5° to 40° inlet air temperature, forced air cooling
Operating Humidity	20% to 80% (non-condensing)

OPTIONS

Part Number	Description
MVME204-1,-2	1Mb/2Mb Interleaved Dual-Ported DRAM Modules
MVME214	A32/D32 Static RAM/ROM Memory Module
MVME331,332	6-/8-Channel Intelligent Communications Controllers
MVME705, 706	I/O Transition Boards for MVME331

RELATED DOCUMENTATION

Part Number	Description
MVMESYSAM/D	VME System Architecture Guide
MC68020/D	MC68020 Microprocessor User's Manual
HB212/D	VMEbus Specification Manual

2

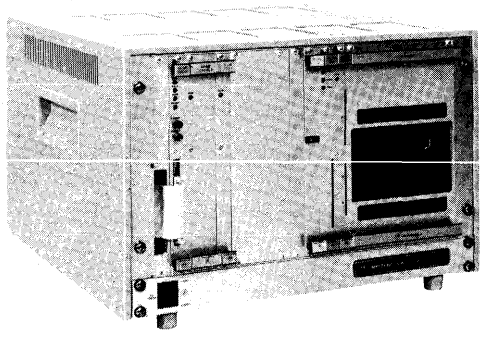
VMEbus Modular System With VERSAdos 4.5

FEATURES:

- VERSAdos 4.5
- MVME204-2 2Mb Dual-Ported DRAM with VSB
- MVME320A-1 Disk Controller
- MVME350 Streaming Tape Controller
- MVME331 6-Channel Serial Communications Controller
- 70Mb Winchester Disk Drive
- 655Kb Floppy Disk Drive
- QIC-02 Cartridge Streaming Tape Drive
- Eight RS-232-C Ports (Total)
- MVME131XT Microcomputer with
  - — MC68020 Microprocessor (16.67 MHz)
  - — MC68881 Floating Point Coprocessor (16.67 MHz)
  - — MMB851 Memory Management Unit
  - — 16Kb Instruction/Data Cache
  - — Two Sockets for JEDEC 28-Pin ROM/EPROM Devices
  - — Two Sockets for JEDEC 28-Pin RAM/ROM Devices
  - — Two Multi-protocol Serial Ports
  - — A32:D32 VMEbus Interface
  - — VSB Interface*
- System Level Diagnostics (VMEsystest)

MVME945 CHASSIS FEATURES

- 19″ Rack Mount Chassis with Tabletop Enclosure
- Transverse 400 W Mid-Chassis Power Supply
- 12-Slot, VMEbus-Compatible Card Cage
  - — A32:D32 Backplane
- Accommodates up to 20 Rear-mounted 80 mm I/O Transition Modules



The greatest advantage of a VMEbus-based system is its modularity. To a core system, add only those software and hardware components required for the application. Then serve a new application by changing only a component or two.

The SYS1131VY341 with VERSAdos 4.5, is a versatile core system configured using standard VMEbus modules. Supplied in 19″ rack-mountable chassis with an enclosure for desktop use, the system has ample room for hardware expansion. This system offers outstanding performance by virtue of an intelligent mass storage controller, intelligent streaming tape backup, high-performance system RAM and a powerful monoboard microcomputer with cache based on the MC68020 32-Bit Microprocessor. The MVME331 Serial Communications Controller and the MVME705 I/O Transition Module provides 6-channel RS-232-C serial I/O.

This modular system provides a high-performance core around which, using standard VMEmodules, a microcomputer system serving a specific application can be configured. Five additional slots in the VMEbus backplane are available for expansion. Positions are available for adding additional I/O transition modules, if desired.

For software development under VERSAdos 4.5, the SYS1131VY341 provides a powerful 8-user environment. Additional users can be accommodated by adding an appropriate VMEmodule such as the MVME331 Intelligent 6-Channel Serial Communication Module. The MVME320A-1 Intelligent Mass-Storage Controller supplied with the core systems enhances the use of system resources by VERSAdos 4.5.

VERSAdos 4.5

VERSAdos is a real-time, multiuser, multitasking Operating System for the Motorola MC68000 family of microprocessors. Its modular and multilayered design supports a variety of application environments, and is especially well suited to real-time, on-line control systems. This Operating System serves as a major software building block for real-time applications which use VMEmodule and VERSAmodule board products and the MC68000 family. VERSAdos provides the basic capability to service a wide range of requirements — including process control, data acquisition, inventory control, and commercial and scientific applications.

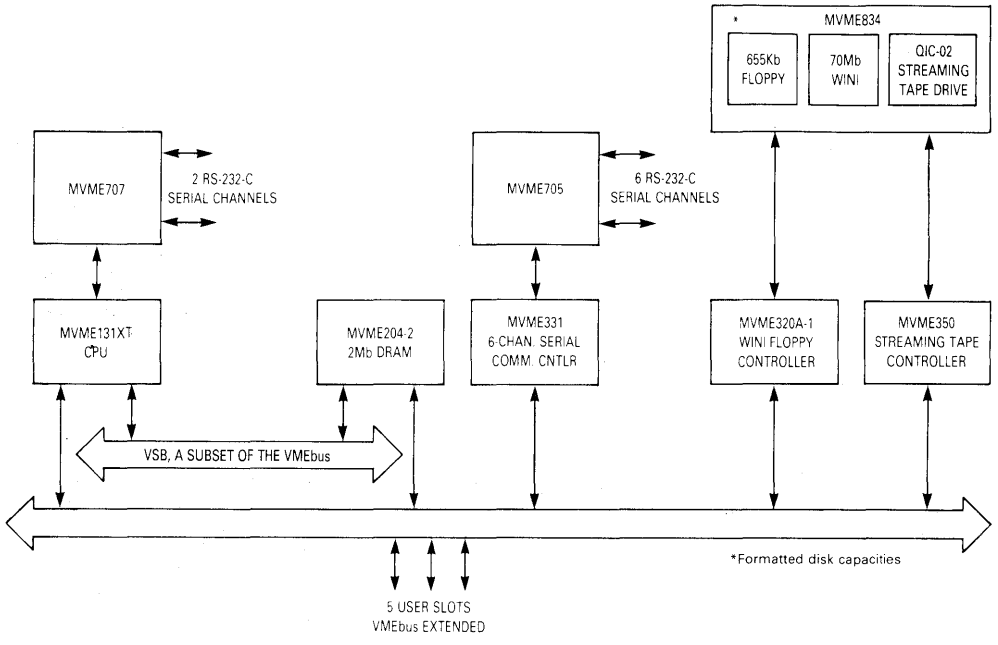


Figure 1. SYS1131VY341 Hardware Configuration

The resources of the Operating System are the MC68020, memory, I/O facilities and secondary storage. Application programs are separated and relieved of the necessity of direct interaction with system hardware by the Operating System. Programs communicate input/output requests via the Operating System in a readily understood protocol.

The Operating System is responsible for accepting, checking, interpreting, and expediting user requests. In order to fulfill its requested task, the Operating System (OS) may call upon OS support routines for assistance. System support routines normally perform functions that are not directly accessed by the application program. These routines assist in operator control, computer memory management, the loading of task segments, or overlays and input/output control for various hardware subsystems.

The degree of user interaction with the system depends on the application. This ranges from almost unattended dedicated process control operation to near full-time attention in such systems as air traffic control.

User requests may be made for:

- Program Load/Initiation
- Hardware Control and Status
- Reading or Setting Current Time of Day
- Requesting I/O
- Storage and Retrieval of Data Files
- Resource Allocation/De-allocation
- Timer Services

VERSAdos DESIGN SUMMARY

The VERSAdos Operating System permits programs to execute in dynamically-assigned, variable-length memory segments with read/write privileges. Instructions and data can be located in separate memory areas, which enable sharing of program code and re-entrant coding practices. A process-to-process facility permits communication between independent programs or nodes of a distributed system.

The heart of the operating system is a real-time executive which provides task services and supports memory allocation. It also allows inter-task communication, provides exception monitor facilities, and handles system interrupts.

The input/output subsystem supports device independence, logical input/output and overlapped computation during physical input/output. New device drivers can be added without impacting the central core portion of the Operating System. Both sequential and random record access are supported by the VERSAdos Operating System.

A powerful file management system supports three types — contiguous, sequential and indexed sequential. Other features include volume and file protection, shared file access, dynamic file allocation and fixed or active protection.

VERSAdos FUNCTIONS

VERSAdos is a coherent group of tasks which use the

2

services of RMS68K. User tasks and VERSAdos communicate via a server mechanism with RMS68K handling all synchronization and scheduling. As a result, VERSAdos is an orderly, easily extendable operating system.

REAL-TIME CONCEPTS

A real-time system responds to external events as they occur. Since the precise time of these external events is an unknown factor, a real-time system is said to execute asynchronously.

Unlike a batch system where one operation is completed before a new operation is started, a real-time system can delay the completion of one operation in order for another operation to be started, continued, or completed. This mechanism (where more than one operation is in progress at a given time) is called concurrent processing. Even though only one operation can be executed at a given time using a single central microprocessing unit, the concurrent processing mechanism of a real-time system gives the illusion of several operations executing simultaneously.

A real-time application system can be broken down into several tasks. A task is a function (or operation) which can execute concurrently with other tasks/functions. A task can be written to process a single type of event, or it can process more than one type of event.

THE EXECUTIVE

The VERSAdos Executive, known as RMS68K, is the nucleus of the Operating System. RMS68K is comprised of an inner kernel (or nucleus) that supports the real-time, priority-driven, multitasking environment. It services all hardware and software interrupts and dispatches them to the proper task for processing. The Executive also resolves conflicts resulting from competing tasks attempting to use the MPU, and has facilities which permit inter-task communication and task synchronization. The Executive provides protection of the user environment and diagnostic feedback during error conditions. Like the rest of VERSAdos, the Executive is designed in a structured fashion. RMS68K may be used as supplied, or tailored to an exact configuration. Modules may be deleted and/or added to RMS68K, based on the users' requirements. RMS68K may also be ROM based or RAM based.

SYSTEMS CENTRAL PROCESSING UNIT

To provide the computing power needed to take full advantage of the VERSAdos 4.5 Multiuser, Multitasking Operating System, a monoboard microcomputer based on the MC68020 microprocessor is used as the central processing unit by the SYS1131VY341.

Unsurpassed in its processing power, the intelligent master of this core system is an MVME131XT VMEbus Microprocessor Module featuring the state-of-the-art, full 32-bit 16.67 MHz MC68020 Microprocessor and including also VSB, a high-speed dedicated interface to the MVME204-2 2Mb Dynamic RAM, the 16.67 MHz MC68881 Floating Point Coprocessor, a functionality subset of the MC68851 Paged Memory Management Unit implemented with gate array technology, and the Cache Accelerator, a plug-in cache memory unit designed for use with the MVME131.

The Cache Accelerator is a small, fast, static RAM system that provides concurrent, local storage of the microprocessor's VMEbus and VSB memory transactions. The Cache Accelerator was designed for users wanting to accelerate the performance of their MC68020-based systems without having to modify existing programs. This software transparency feature is achieved through the utilization of a physical address cache memory structure coupled with a VME Bus Monitor.

The SYS1131VY341 system also includes an MVME707 RS-232 Serial Port Distribution Module which translates the TTL levels output by the two MVME131 ports to provide full support of RS-232-C asynchronous serial communications and which also provides two DB-25 connectors for serial I/O cables. This module is a double-high, half deep board mounted on the upper and lower rear chassis rails and connected to the MVME131XT by a 50-conductor ribbon cable.

The 130bug resident debug package (refer to Motorola publication MVME130BUG) is the firmware package designed for use with the VME131 microprocessor. 130bug provides a powerful system debugging tool for VMEmodule systems. This firmware features a one line assembler/disassembler, provisions for up/downline load, and disk bootstrap commands. 130bug is available as an EPROM-based resident package. It is plugged into IC sockets U28 and U31 on the VME131.

HARDWARE SUPPORT FOR MEMORY MANAGEMENT

Since VERSAdos 4.5 utilizes memory management techniques in the manner of mainframe operating systems, hardware support for the translation of logical addresses into physical addresses and for managing the separation of user and supervisor memory spaces is essential. For the SYS1131VY341 core system, memory management support is provided by the MMB851 gate array implementation of the MC68851 Paged Memory Management Unit function.

CORE SYSTEM MAIN MEMORY

A 2Mb DRAM with parity dual-ported between VMEbus and VSB — MVME204-2 — is supplied with this core system. Up to four additional MVME204-2's can be added to this interface to provide a core system with 8Mb of high-speed memory. The base address for a module can be set on boundaries separated by the module's population.

Tailored to the MC68020 Microprocessor and specifically designed to enhance the performance of the MVME131XT VMEmodule 32-Bit Monoboard Microcomputer, the MVME204-2/VSB interface allows concurrent transfers on VMEbus with transfers between the MVME131XT and MVME204-2. Implemented using high density 256K x 1-bit dynamic RAM devices, the MVME204-2 has parity generation and detection circuitry which, together with accessible control and status registers, can be used for error detection and memory diagnostics.

System memory capacity can be expanded by adding DRAM modules to VMEbus space. Use the MVME204-1, or the -2 to add 1Mb or 2Mb, respectively.

SERIAL COMMUNICATIONS CONTROLLER

The MVME331 Intelligent Communication Controller (ICC) supports six asynchronous serial communication channels at a maximum simultaneous speed of 9600 baud full duplex. The module contains a complete 16-bit microcomputer consisting of an MC68010 MPU, 128Kb RAM and firmware in two 16Kb EPROMs. This microcomputer controls three Z8530 Serial Communication Controller devices, pre-processes and transfers data between the VME system memory and the serial ports.

The transfer of commands, status messages and data packets is performed through a pipeline structure in VME system memory, which is shared between the operating system and the MVME331 firmware. This transfer follows the Buffered Pipe Protocol (BPP), a Motorola standard for processor to processor communication.

Each of the six serial channels can be configured to conform to the RS-232-C or RS-422-B standard. This is implemented on a separate Transceiver Module (MVME705 or MVME706), a double-high, half deep board mounted on the upper and lower rear chassis rails and connected to the MVME331 by a 50-conductor ribbon cable.

MASS STORAGE DEVICE CONTROLLERS

THE MVME320A-1 VMEbus DISK CONTROLLER

To control the provided floppy disk drive and Winchester disk drive, the SYS1131VY341 utilizes the MVME320A-1 VMEbus Disk Controller. Although not supported, this VMEmodule has the capability of controlling mixed 5-1/4" and 8" drives and can accommodate two 5-1/4" Winchester hard disk drives and two 5-1/4" floppy disk drives or up to four floppy disk drives.

Because the controller provides high-performance direct memory access (DMA) data channels between core system memory and the floppy and hard disk storage, the core system can be used in applications having intensive real-time disk I/O or multiprocessing structures designed to reduce VMEbus traffic and increase system throughput. Other MVME320A-1 features include:

- Supports Serial Data Rates to 5Mbps
- Supports Standard IBM Formats and MFM Recording
- 16-Bit Cyclic Redundancy Check
- 32-Bit Error Checking and Correction

THE MVME350 STREAMING TAPE CONTROLLER

A VMEmodule Intelligent Peripheral Controller (IPC), MVME350 provides control of one QIC-02 compatible, 1/4 inch streaming tape drive.

The 16Kb of static RAM onboard MVME350 provides working storage for the MC68010 vector table and stack frames. It also serves as buffer storage between the QIC-02 interface and the VMEbus. Optimum performance for typical applications is offered since the MC68010, at 10 MHz, performs zero wait state accesses of the SRAM space.

The VMEbus interface is designed to provide uniformity and compatibility among MVMEmodule drivers running under VERSAdos 4.5 as well as ease the development, by a system integrator, of software to communicate with the controller module. Controller firmware based on a multitasking kernel handles all input/output interrupts and data transfers in order to minimize impact on system throughput. Macro commands are retrieved from a pipeline structure in system memory which is shared by the MVME131XT microcomputer, and the tape controller. Status information is returned to a host through another pipeline. Exchange of command and status packets follows the Motorola buffered pipe protocol for interprocessor communication.

VERSAdos provides a complete set of utilities in support of the MVME350 streamer tape. Tape mount and dismount functions are included, as well as tape initialization and directory list. Disk backup and restore utilities exist to allow the user to do disk image copies from/to the hard disk. Additionally, incremental file backups and restores are supported. Another feature supported is the capability to boot VERSAdos directly from tape and rebuild the hard disk from the tape contents.

MASS STORAGE DEVICES

Mass storage capacity for the SYS1131VY341 system is provided by a floppy disk drive, a hard disk drive and a streaming tape.

The floppy disk is a 5-1/4" slim line drive for soft-sectored dual-sided dual-density, 96 TPI diskettes each with a capacity of 655Kb of formatted data. Features include a track density of 96 T.P.I., a recording density of 5922 B.P.I. (double density) and a data transfer rate of 250Kb per second.

The hard disk uses reliable Winchester technology and provides a capacity of 70Mb, formatted. Features include 32 sectors per track, 256 bytes per sector and a data transfer rate of 5Mb per second.

The streaming tape drive consists of a drive unit in a standard 5-1/4" half-height enclosure and a formatter/controller module which fastens to the enclosure and forms a full-height 5-1/4" product. By using the industry standard DC600 tape cartridge, a full 60Mb of tape backup can be provided.

USER INTERFACE

Starting with a suitable chassis a VMEbus-based system can be constructed from the wide variety of commercially available modules to serve almost any application. The core of this system must simultaneously offer configuration flexibility and ease of operation.

The SYS1131VY341 is housed in a MVME945 chassis assembly, the desktop decorative cover of which can be removed so that the system can be mounted in a standard 19-inch rack, if the application requires. The front of the chassis houses a card cage with backplane which accepts twelve standard double high Eurocard VMEmodules. On the right, the floppy hard disk and streaming tape drives supplied with the core system are housed in the MVME834. On the left, a power switch module is attached.

2

On the left in the chassis rear, attached by screws to the upper and lower rails, are the two 80 mm I/O Transition Modules supplied with the SYS1131VY341. The remaining space can be used for additional transition modules.

A 400 watt power supply is mounted in a fan-cooled air duct assembly located in the upper half of the chassis to the rear of the VMEmodule backplane. Additional cooling sufficient for an expanded system is provided in the chassis.

By virtue of the high intelligence and integration levels of its modules, the SYS1131VY341 is able to present to the user an uncluttered front which offers the essential basic controls and system status indications and, at the same time, the means of configuring the hardware to provide certain system level functions.

MVME131XT Front Panel —

FAIL Indicator, Red LED
HALT Indicator, Red LED
RUN Indicator, Green LED
SCON Indicator, Green LED
Reset Switch, spring-loaded pushbutton
Abort Switch, spring-loaded pushbutton
8-Position DIP Switch
50-Pin Connector
RUN Indicator, cache module, Green LED

MVME204-2 Front Panel —

Power Indicator, Green LED

MVME320A-1 Front Panel

FAIL Indicator, Red LED
Two Slots for ribbon cable entry/exit

MVME350 Front Panel

FAIL Indicator, Red LED
HALT Indicator, Red LED
RUN Indicator, Green LED

EXPANSION FOR ADDITIONAL USERS

The backplane supplied with the SYS1131VY341 chassis can accommodate additional VMEmodules. Access to the unused slots is gained by removing the screws fastening dummy module front panels to the upper and lower rails of the chassis front.

The eight serial interfaces provided with this core system for terminals or other I/O devices can be expanded by six RS-232-C or RS-422-B ports by installing the MVME331 Intelligent Communications Controller Module in the chassis backplane. A companion I/O Transition Module, the MVME705 which supplies the DB-25 connectors and the transmitters/receivers required to support the two communication protocols is installed by screwing the module front panel to the top and bottom rear rails of the chassis. Connection is made between the two modules using a 64-conductor ribbon cable.

Memory can be supplemented in increments of 1 or 2 megabytes by adding the MVME204-1 or -2.

Extensive documentation for the hardware and VERSAdos 4.5 is supplied with the SYS1131VY341. Documentation supplied for VERSAdos 4.5 describes in detail the process of generating a new kernel for use with an expanded system. Hardware and module documentation supplied with the core system and expansion modules details the process of configuring a system to comply with the requirements of various applications.

SYS1131VY341 SYSTEM TEST (VMEsystest)

The SYS1131VY341 System Test is a comprehensive set of powerful system level tests and fault isolation diagnostics. The software executes on this VMEbus Modular System with VERSAdos 4.5 or on other VMEbus hosts and can be a versatile tool for system operators, as well as engineering, manufacturing and field service personnel. It can be used for new design verification, for system product verification or for performance measurement and fault isolation in installed systems. It can also serve the system integrator as the basic test structure in which tests for added modules can be included as a means of offering a system test facility with the custom configuration.

SYS1131VY341 System Test features include:

- Simple Default Mode — Test of the as-shipped SYS1131VY341 VMEbus Modular System
- Flexible, easy-to-use Operator Interface Accommodates Varying Needs:
  - — Change complement of tests for a new system configuration
  - — Change sequence of tests to provide a new test environment
  - — Change internal execution of a test to meet specific requirements
- Elaborate Test Environment Controls for Fault Isolation
- Timer Functions to Support System Performance Measurement
- Error Logger and I/O Re-direction for Test Record Keeping
- Extensive Manual Debugger for Fault Isolation and Specific Function Checks
- Target Environment Simulated by Tasks Executing Under a Real-Time, Multitasking Kernel
- C Language Source Available for Straightforward Porting

To provide the essential synchronizing functions and task control utilities required for the simulation of a target operating system, SYS1131VY341 System Test utilizes a real-time, multitasking kernel. This allows multiple test tasks to execute concurrently creating a realistic on-line simulation in a controlled environment. Use of a real-time, multitasking kernel also facilitated the creation of utilities for dynamically tailoring test conditions to match changing hardware configurations and for pinpointing a specific fault or measurement state. The addition of a multi-faceted debug utility resulted in a system test capability that isolates faults to the replaceable component level — and beyond.

SYS1131VY341 System Test has been developed in a SYSTEM V/68 environment using the SYSTEM V/68 "C" compiler. Changes to VMEsystest source code should be done in the SYSTEM V/68 environment to minimize any compatibility problems.

VMEbus

Rapidly becoming the leading bus standard in U.S., European and other world markets, VMEbus is a 8-/16-/32-bit interconnect that has a master/slave asynchronous, non-multiplexed data transfer structure, seven levels of priority interrupt, four levels of data bus arbitration, multiple master support, fault detection and control, and special transfer cycles.

The VMEbus functional structure consists of backplane interface logic, four groups of signal lines called buses each having its own collection of functional modules any number of which can be grouped on a board and which communicate with the other functional modules of a bus via onboard or backplane signal lines.

The VMEbus mechanical structure comprises subracks, backplanes, front panels, plug-in boards and other elements all of which conform to relevant IEC Eurocard specifications to insure that all components fit together properly.

These features combined with reliability, mechanical integrity, 20 megabytes-per-second performance and the fact that it is nonproprietary have resulted in almost complete acceptance of VMEbus by the electronics industry. Currently over 200 vendors are manufacturing over 1000 VMEsystem-compatible products, assuring for the expansion of a core system, the widest selection of functions from the greatest number of sources so that the broadest range of applications can be served.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

MVME1131VY341	
MCU	MVME131XT VMEmodule 32-Bit Address/Data Monoboard Microcomputer (16.67 MHz MC68020 Microprocessor, 16Kb Physical Cache Memory and MC68881 FPCP)
Memory	2Mb Dual-Ported RAM with MVMX32bus (MVME204-2)
Mass Storage Controllers	MVME320A-1 VMEbus Disk Controller MVME350 VMEbus Streaming Tape Controller
Serial I/O Controller	MVME331 Serial Communications Controller
Mass Storage	70Mb (formatted) Winchester Disk Drive 655Kb (formatted) Floppy Disk Drive Streaming Cartridge Tape Drive
I/O Interfaces	8 x RS-232-C programmable to 19.2K baud Rear serial I/O connections via MVME705 & MVME707 Transition Modules
Expansion Slots	12-slot Backplane, 5 slots free for user
Power Supply	400 W (115 Vac)
Operating Temperature	5° to 40° inlet air temperature, forced air cooling
Operating Humidity	20% to 80% (non-condensing)

2

ORDERING INFORMATION

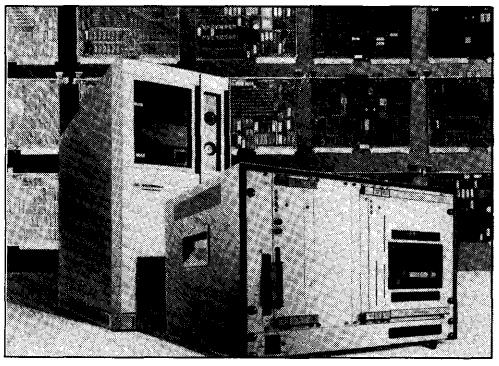
Part Number	Description
SYS1131VY341	VMEbus Modular System with VERSAdos 4.5, includes: MVME131XT VMEmodule Monoboard Microcomputer with cache MVME204-2 2Mb Dual-Ported DRAM MVME320A-1 VMEbus Disk Drive Controller MVME331 Serial Communications Controller MVME350 Streaming Tape Controller MVME705 Transition Module MVME 707 Transition Module MVME834 Plug-In Mass Storage Module with — 70Mb Winchester Disk Drive — 655Kb Floppy Disk Drive — Streaming Cartridge Tape Drive MVME945 VMEbus Chassis Assembly with — 32-Bit VMEbus Backplane — 400 W Power Supply User's Manuals for VERSAdos 4.5, chassis and all modules included

OPTIONS

Part Number	Description
MVME204-1,-2	1Mb/2Mb Interleaved Dual-Ported DRAM Modules
MVME214	A32/D32 Static RAM/ROM Memory Module
MVME331,332	6-/8-Channel Intelligent Communications Controllers
MVME705, 706	I/O Transition Boards for MVME331

RELATED DOCUMENTATION

Part Number	Description
MC68020/D	MC68020 Microprocessor User's Manual
HB212/D	VMEbus Specification Manual



Many of today's microprocessor applications demand such data handling and processing power that designers must consider use of board level products having a form factor larger than that of VMEbus-compatible modules. VERSAmodules, a broad line of 16/32-bit system components are designed to meet this challenge.

A VERSAmodule system is interconnected using Motorola's standard VERSAbus architecture which supports non-multiplexed, asynchronous operation, multiple processors and bus masters, a powerful 7-level priority interrupt structure, system failure detection and offers 8- to 32-bit data and address paths for data transfer rates up to 20 megabytes per second.

The VERSAmodule format provides a board area nearly three times that of the VMEbus-compatible double Eurocard allowing implementation of functions having significantly greater memory capacity and computing power. Available modules include MC68000-, MC68010-, and MC68020-based monoboard microcomputers, memory modules with up to four megabyte capacity, peripheral interface, controller and communication modules plus system packaging and accessories. VERSAmodule products, of course, are fully supported by both the VERSAdos Real-Time Operating System and the SYSTEM V/68 Operating System.

VERSAmodule Data Sheets

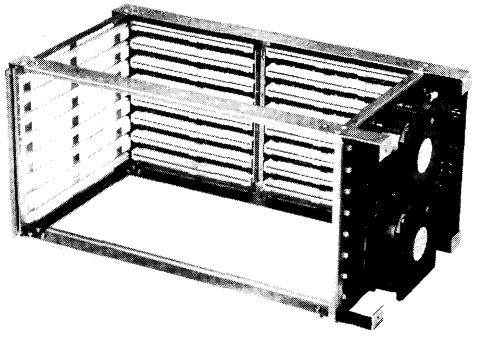
(continued)

VERSAmodule Data Sheets (continued)

MVMCH3-1, -2
MVMCC3
M68KVMPM1

VERSAmodule Accessories Card Cage, Chassis, Power Supplies, Power Monitor

Interconnects and Hardware



These accessories are designed to convert the individual MC68000-based VERSAmodules (cards) into a complete, compact, self-contained microcomputer system that best meets the user's performance and installation requirements. Sufficient options are provided to permit the convenient assembly of systems for custom installation features, including Table-Top or RETMA Rack Mounted Configurations.

VERSAmodule Card Cage

- Eight-Slot Backplane
  - — VERSAbus Compatible
  - — Provides 100 input/output connector pins per slot, each with corresponding ground pin
  - — Input/output mating connector kits available
- Bus Terminators Included
- Rugged, Open-Frame Metal Construction
- Mechanically and Electrically Expandable to Two Units (16 Slots)
  - — Expansion kit available
  - — Mounting/stacking holes provided on top and bottom
- Mating dc Power and Control Signal Connector Kit Included
- Weight 5.0 lbs. (2.0 kg)

The VERSAmodule Card Cage allows the user to implement a VERSAmodule system of from one to fifteen cards in a housing appropriate to the application, e.g., NEMA enclosure. The basic Card Cage provides an eight-slot VERSAbus backplane in a rugged, open-frame metal housing. Two units can be bolted together to form a sixteen slot system. The backplane VERSAbus signals can be electrically interconnected with the optional expansion kit.

Bus termination devices are included in the Card Cage. The VERSAbus backplane also provides 100 input/output signal pins per slot with 100 corresponding ground pins in four groups of 50 pins (25 signal, 25 ground) in addition to the microcomputer system bus address, data and control signals at each slot. Mating power connectors for the backplane are included.

VERSAmodule Chassis

- Incorporates Eight-Slot Card Cage in Front-Load Metal Housing
- All versions incorporate 6-slot I/Omodule card cage with rear-panel access
- RETMA Rack Mountable with Slide Kit Available
- Snap-on, Removable Front Panel with Molded Bezel and Removable Metal Plate for User-Added Controls and Indicators (8.71" (132.6 cm) panel height)
- Decorative Cover Available for Table-Top Configuration
- Power On-Off Rocker Switch and Power Indicator Accessible from Front Panel
- Modular Rear Panel for User-Installed Peripheral Input/Output Interface Connectors with Internal I/O Ribbon Cabling
- Three Fans for Cooling Cards and Power Supply
- 400 Watt Switching Supply with Overvoltage/Overload/Overtemperature Protection (see following features)
- Power Monitor Module Included (see following features)
- Weight 42 lbs. (19 kg) with Power Supply
- Operating Ambient Temperature — 0°C to 50°C

The VERSAmodule Chassis incorporates the eight-slot Card Cage in a front-loading, metal enclosure. The Chassis is rack mountable in a standard 19" RETMA rack using the optional slide kit or may be used as a table-top unit with an optional decorative cover. The Chassis has a snap-on, removable front panel with a molded bezel and a modular backpanel that allows the user to easily add any required peripheral connectors.

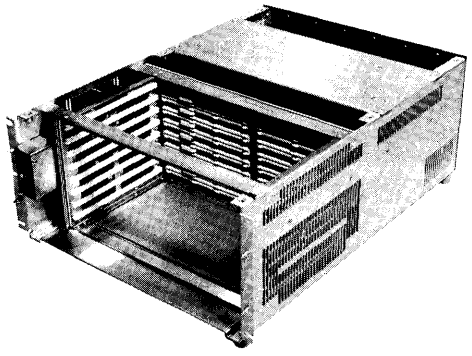
For users planning to employ single Eurocard format I/O-module cards, all versions have a 6-slot I/Omodule Card Cage installed with rear panel access.

Each Chassis has an ac power on/off rocker switch and power indicator accessible from the front panel. The ac line fuse and detachable line cord are located at the rear of the unit. Three fans are included that provide forced-air cooling for the Power Supply and the VERSAmodule cards.

A removable metal panel is provided on the rear of the enclosure to allow the addition of required system input/output interface connectors. Four 50-pin input/output ribbon connectors are provided to aid the user in the construction of ribbon cables that attach to the 100-pin signal/100-pin ground pins on the backplane at each slot.

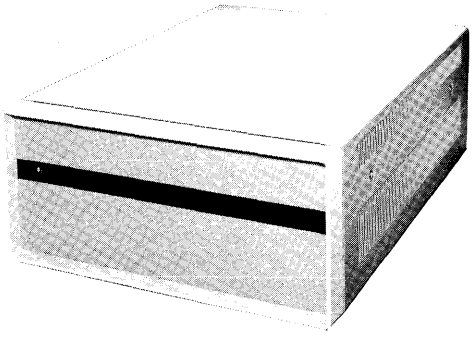
Connections For Backup

The backpanel is partitioned to handle the connect requirements of each peripheral board. The partitioning handles connector and transition boards or panels associated with each peripheral. The connector and transition boards come complete with backplane connector cables. Due to the number of combinations of peripheral boards, the backpanel cannot handle all combinations of disk, communication and monocomputer boards.



The power supply is a 400 watt switching power supply that provides +5 Vdc, and ±12 Vdc. Features include overvoltage and overload protection and over-temperature protection.

A Power Monitor Module is included that monitors the Power Supply Low Line Detect signal and ac line voltage for both cycle dropout and low-line voltage conditions. The circuit generates the proper power-up/power-down sequence of VERSAbus control signals allowing the system to take appropriate action. A VERSAbus ac line clock signal is also driven by the Power Monitor. See the following Power Monitor description for details.

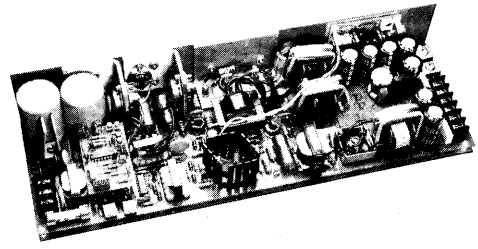


400 Watt Switching Power Supply

- AC Input Voltage Range of 90-130 Vac, Single Phase, 47-440 Hz
- Factory configurable to 180–264 Vac
- Output Voltages and Maximum Current of:

  +5.0 Vdc @ 60 A
  +12 Vdc @ 8.0 A
  −12 Vdc @ 6.0 A
  } Output Floating

- Power Fail Holdup Time — All Outputs
  Full Rated Load = 16 ms (Min.) Nominal Line Voltage
- Generates PS Low Line Detect Signal to Power Monitor Module a Minimum of 4 Milliseconds Prior to Loss of Regulation of dc Outputs
- Power Monitor Module included (see following details)
- Overload Protection Foldback Current Limiting on Each Output
- Overvoltage Protection of 5 V at 125% ±5% of Nominal. Reduced to rated output within 500 microseconds
- Overtemperature Thermal Shutdown (generates Supply Fail signal)
- EMI and RFI Filtering Included
- Dimensions: L x W x H: 15″ (203.2 mm) x 2.5″ (157.5 mm) x 5.0″ (127 mm)
- Weight: 11 lbs (5.0 kg) — exclusive of fan and power monitor module
- Operating Temperature Environment: 0°C to 50°C
  Forced air @ 80 CFM (Min.) from integral fan unit



The VERSAmodule 400 W Power Supply is a high efficiency switching supply that provides all of the required operating voltages for a VERSAmodule system. The regulated supply outputs are +5 Vdc for the logic circuits, and ±12 Vdc for the serial I/O circuits.

Power Supply output characteristics are given in Table 1. The supply has overload, overvoltage and overtemperature protection. The supply also generates a PS Low Line Detect Signal to the Power Monitor Module. This signal will occur a minimum of 4 milliseconds prior to the loss of dc regulation in the supply. The Power Monitor uses this signal to generate the sequence of power-up and power-down signals used on VERSAbus. A fan is included as part of the assembly to provide forced-air cooling.

Table 1–VERSAmodule 400W Power Supply Output Characteristics

Output Voltage (Vdc)	Max. Current (Amps)	Adjust Range (Vdc)	Load Reg (No Load to max) (%)	Line Reg (low ac to high ac line volt) (%)	Cross Reg 50%–100% load change any output (%)	Output Noise and Ripple (mVp-p)	Temp. Coeff (%/°C)
+5.0	60	4.8 to OVP	0.5	0.2	1.0	100	0.02
+12.0	8.0	Fixed	Stay within 5% for any combination of line, load on cross regulation			120	0.05
−12.0	6.0	Fixed				120	0.05

VERSAmodule Power Monitor

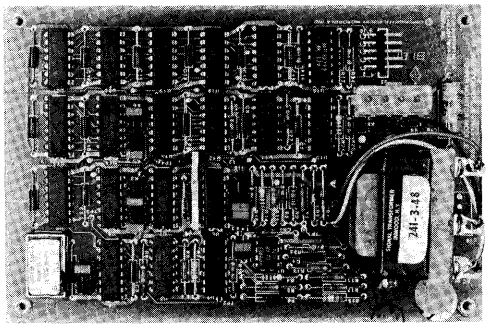
- Monitors System AC Line (on-board transformer) and Supply Fail Signal
- Operates from +12 Vdc
- Detects Cycle Dropout (user selectable for 0.5, 1.0 and 1.5 cycles) Below a 90 or 100 Vac Input Line Voltage
- Generates Proper Sequence of Power-up and Power-down Signals for VERSAbus
- Provides ACCLK Signal for VERSAbus for AC Line Clock Applications
- Size (including components, see Figure 1) Width x Height x Thickness 6.0″ (152.4 mm) x 4″ (101.6 mm) x 1.375″ (34.9 mm)

System integrity is enhanced when it can predict the loss of operating voltages and take appropriate action before the voltages drop below the system minimums. This is accomplished by monitoring the system ac prime power for both cycle dropout and low-line voltage conditions and signaling to the system to allow orderly shutdown and recovery to occur. In operating environments where periodic cycle dropout conditions occur, the ability to selectively ignore dropouts not affecting the proper operation of the system is important. These features are provided by the Power Monitor Module, which is included in the VERSAmodule Chassis, in the Switching Power Supply, and is available separately.

As previously indicated, the VERSAmodule 400 watt Switching Power Supply monitors its internal operation and will generate a PS Low Line Detect signal to the Power Monitor a minimum of 4 milliseconds before the loss of dc regulation. The Power Monitor also monitors the ac line voltage for user selectable conditions of 0.5, 1.0, or 1.5 cycles of voltage that are below the threshold of the input to the power supply.

The Power Monitor will, based on either PS Low Line Detect or cycle dropout detect, generate a VERSAbus control signal (ACFAIL*) that can initiate a system power-down sequence. Following a wait of 8 milliseconds, a VERSAbus control reset signal (SYSRESET*) will be generated which may be used to inhibit further memory writes and to reset the appropriate system elements. The Power Monitor also generates a power-up sequence in which the system is held in reset for a period of 500 milliseconds to allow the dc power to stabilize.

The Power Monitor also generates a VERSAbus ac line clock signal for applications requiring line sychronization.



The following connections are required by the Power Monitor:

AC Line Input

- 110 Vac connection to transformer via two fast-ons (3/16″)

Power Supply Inputs on 4-Pin Connector (J1)

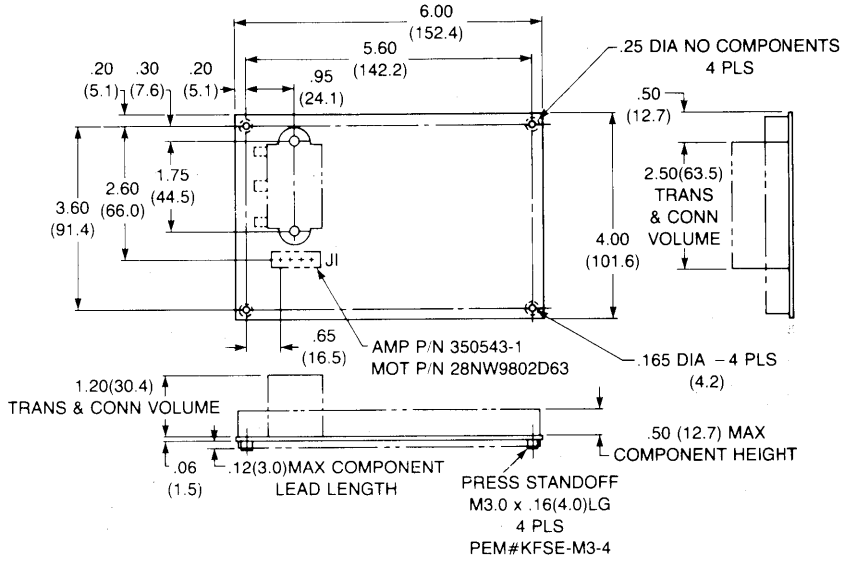
- +5 V (for Monitoring During Power Up)
- +12 Vdc circuit power @ 100 mA (Max.)
- +12 Vdc supply fail signal (active low)
  V_{IL} = 2.0 Vdc (Max.) @ 3.0 mA (Min.)
  V_{IH} = 7.0 Vdc (Min.)
- Board mating connector (P1)
  Amp (Mate-N-Lok) 1-480424-0 (4 pin)

VERSAbus Output Signals on 10-Pin Connector (J2)

- ACCLK
- SYSRESET*
- ACFAIL*
- Board mating connector (P2):
  3M 3473-7000 (10-pin)

Note that when purchased and installed as a separate assembly, the Power Monitor should be located near the VERSAbus backplane rather than at a remotely located Power Supply to minimize noise on the VERSAbus.

FIGURE 1 — Power Monitor Circuit Module



Ordering Information

Part Number	Description
MVMCC3	Eight-slot, open frame card cage. Includes User's Manual.
MVMCH3-1	Eight-slot, rack mountable chassis with 110 V, 400 watt switching power supply and power monitor module. Four 50-pin connectors to interconnect to the VERSAbus input/output pins are also included. Includes User's Manual.
MVMCH3-2	Eight-slot, rack mountable chassis with 220 V, 400 watt switching power supply and power monitor module. Also includes 6-slot card cage for single Eurocard format I/O cards, plus four 50-pin connectors to interconnect to the VERSAbus Input/Output pins. Includes User's Manual.

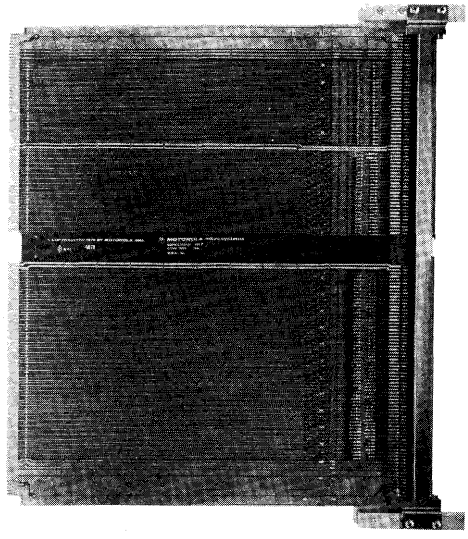
Options

Part Number	Description
M68KVMPM1	Power monitor module for use with VERSAmodule Card Cage (included in both versions of chassis and in switching power supply). Includes User's Manual.
MVMCH3CHE	Expansion kit that provides the VERSAbus interconnect cables to use between two card cages. Includes installation instructions.
M68KVMCHS	Rack mount slides and mounting hardware for VERSAmodule Chassis. Includes installation instructions.
MVMCH3CVRD	Decorative cover for eight-slot VERSAbus Chassis when used as a table-top unit. Includes installation instructions.
MVMCH3-103	Serial I/O cable assembly for VERSAmodules 02, 03, and 80. Provides 50-pin connector and ribbon cable to two EIA RS-232C connectors on a chassis backpanel mountable circuit board. Synchronous or asynchronous operation for VM03. Asynchronous only to VM02. Includes installation instructions.
MVMCH3-104	Serial I/O cable assembly for VERSAmodule 04. Provides 50-pin connector and ribbon cable to convert TTL to EIA for two RS-232C connectors on a chassis backpanel mountable circuit board. Synchronous or asynchronous operation. Includes installation instructions.
MVMCH3-121	Serial I/O Cable Assembly for VERSAmodule M68KVM21. Provides ribbon cable from VM21 to chassis backpanel mountable connectors. Includes installation instructions.
MVMCH3-122	Serial I/O Cable Assembly for VERSAmodule M68KVM22. Provides ribbon cable from VM22 to chassis backpanel mountable connectors. Includes installation instructions.
MVMCH3-123	Serial I/O Cable Assembly for VERSAmodule M68KVM23. Provides ribbon cable from VM23 to chassis backpanel mountable connectors. Includes installation instructions.
MVMCH3-132	Serial I/O Cable Assembly for VERSAmodule M68VM30 or M68VM32. Provides ribbon cable to eight EIA RS-232C connectors on a chassis backpanel mountable circuit board. May be used with two VM30's or VM322. Synchronous or asynchronous for VM30. Asynchronous operation for VM32. Includes installation instructions.
MVMCH3-133	Serial I/O Cable Assembly for VERSAmodule M68KVM33. Provides 15-pin DLC cable to chassis backpanel mountable connectors. Includes installation instructions.

Documentation

M68KVMESH3/D1	VERSAmodule System Chassis/Card Cage Enclosure User's Manual describing all features and options relevant to the MVMCH3-1, -2 and MVMCC3 Chassis and Card Cages.
M68KVMPS1/D1	VERSAmodule System Switching Power Supply User's Manual.
M68KVMPM1/D1	VERSAmodule System Power Monitor User's Manual.
M68KVBS/D4	VERSAbus Specification Manual.
M68RIOCS/D1	Input/Output Channel Specification Manual.

VERSAbus Extender Module



Electrical Features

- Electrostatic Shielding Between Adjacent Signals
- Internal Ground Plane for Excellent Front-to-Back Signal Isolation
- Individual Test Points for Signals
- Signal Isolation

Mechanical Features

- Mechanical Inserters and Ejectors for the Extender Module to VERSAbus Connection
- Mechanical Inserters and Ejectors for the Module Under Test to Extender Module Connection
- Full Length Card Guides

VERSAbus Extender Module provides a convenient system for the routine testing or trouble shooting of VERSAbus modules. The module under test is mechanically inserted into the Extender Module card guides, thus raising it to a convenient level for servicing. The Extender Module "extends" VERSAbus signals and power to the module under test.

Individual test points are provided for each extended signal. The bus signal may be isolated by cutting the track between two .10" spaced hole patterns. A header with shorting bars may be inserted to restore signal continuity. Mechanical ejectors ease the removal of test modules as well as the extender module itself.

Ordering Information

Part Number	Description
M68KEXTM	VERSAbus Extender Module

M68KVBUG, 2
M68KVBUGLF, 2LF
M68KVBUGLC, 2LC
M68KVBUG2LMC

VERSAbug Debugging Packages for the M68KVM01A and M68KVM02-3 VERSAmodule Monoboard Microcomputers

VERSAbug Resident Package

- EPROM resident system debug monitor
- Dual port RS-232C Serial I/O Cable Assembly allows connection of debug terminal and up/downline load host to the VERSAmodule Monoboard Microcomputer (VMM) through the 50-pin I/O Connector arrangement of VERSAbus
- 39 debug, up/downline load and disk bootstrap load commands.
- Full speed execution of system and user developed programs operating out of the VERSAmodule Monoboard Microcomputer
- Virtual terminal capability for up/downline load from an EXORmacs/EXORciser Development System or from a cross-computer
- Powerful software and system debug command set allows access to all VMM I/O, control and memory facilities plus the full 16M byte direct address range of the VERSAbus system bus
- Disk Bootstrap load/dump from standard Motorola Floppy and Hard Disk Systems
- Includes all required installation and operation documentation

VERSAbug Source and Relocatable Object Module Package

- Source and Relocatable Object Modules for VERSAbug on diskette or cartridge disk
- Relocatable Object Modules allow the user to include only the VERSAbug items needed in their end system; to link in their own up/downline loader; and/or to link in their own bootstrap loader
- Source Modules allow user modification of VERSAbug as desired.

The VERSAmodule Monoboard Microcomputer debug package, VERSAbug, is available as two separate product offerings. VERSAbug comes as an EPROM-based resident package with dual serial I/O cable assembly ready for installation and immediate use with the Monoboard Microcomputer in a VERSAbus based backplane. Such a backplane is provided by Motorola's Card Cage (M68KVMCC1), or Chassis. VERSAbug Source and Relocatable Object Modules are available as a separate product on either EXORmacs Development System compatible diskette, or disk cartridge.

VERSAbug provides a powerful evaluation, use and system debugging tool for VERSAmodule Systems. The EPROM Resident Package (M68KVBUG) will operate in a minimum of 8K bytes of ROM space. An Extended Functions package requires an additional 4K bytes of ROM for a total of 12K bytes. VERSAbug uses the first 1024 words of RAM storage for Interrupt Vectors and temporary storage. The EPROM resident package is delivered in six 2K byte EPROMs, though only four are required if the Extended Functions are not needed. Table 1 lists the commands available to the user in 8K ROM space, Table 2 lists the Extended Functions and Commands available in the 4K byte extended ROM space.

The package permits full speed execution of system and user-developed programs operated in a VERSAmodule Monoboard Microcomputer (VMM) system environment under complete operator control. The dual serial I/O cable assembly provided with VERSAbug allows terminal and host access to the two serial ports on the VMM. VERSAbug may be utilized with a VMM in a stand-alone environment with only a user provided standard RS-232C asynchronous ASCII terminal. Alternately, it may be used with the second serial I/O port direct connected to a host computer for up/downline loading of programs in Motorola "S" Record format. When directly connected to a host computer in this manner, the VMM/VERSAbug/Operator Terminal combination appears as a normal asynchronous ASCII terminal (a virtual terminal) to the host operating system. Figures 1 and 2 illustrate two typical configurations using the VERSAbug EPROM set installed in a VMM.

In a typical debug session, the user will download his developed program to a VMM from the host computer used for software development. This may be a Motorola EXORmacs or EXORciser Development System. Following load, VERSAbug commands may be used to examine and modify memory, set breakpoints to run particular program segments, and trace program progress. The user may set up and examine a variety of conditions using any of the powerful commands listed in Tables 1 and 2, such as the Register Display/Set series and the memory block manipulation commands. The Data Conversion command serves as an aid in examining and modifying data by providing a means of converting hexadecimal to decimal, and decimal to hexadecimal. If corrections or program patches are required, these may be performed and checked in the VMM. The user may then save a corrected copy to the host computer files utilizing the Memory Dump command for upline load. Creating program patches may be aided by use of the Display Offsets command to assist with relocatable and position independent code. The user may also copy all traffic to the serial port debug terminal on a printer attached to one of the VMM parallel ports by use of the Attach Printer command. This may be useful for desk debugging following a debug session.

The user may communicate directly with the host computer as a terminal for purposes other than up/downline load by executing the Transparent Mode command or the Send Message to Port 2 command. By using the Port Format command, the serial ports on the VMM may be reconfigured for such attributes as baud rate, stop bits and number of data bits. In the Transparent Mode, the two serial ports must be operated at the same baud rate.

VERSAbug may be used for debug in total systems environments including the VMM together with other Motorola VERSAmodules (RAM, floppy and hard disk controllers, communications controllers, A/D controllers, etc.) as well as user-developed VERSAbus compatible modules.

Bootstrap load and dump commands permit the user to bootstrap from standard Motorola floppy and hard disk systems utilizing media with the Motorola EXORmacs diskette/disk format. The Boot Dump command permits the user to write his operating system to an EXORmacs diskette/disk in bootstrap load format for subsequent use in boot loading. The IOP command permits the user to create the EXORmacs diskette/disk format required.

The Source and Relocatable Object Module Package (M68KVBUGL) provides the user with the information necessary to link VERSAbug into their specific system in either modified, or unmodified form. The Relocatable Object Modules are designed to permit creation of a load module with or without the "S" Record up/downline loader, the VERSAbug disk bootstrap loader, and/or the Extended Functions package. The user may also use the Relocatable Object Modules substituting their own device bootstrap loader for the VERSAbug disk bootstrap loader.

Source Modules permit the user to modify, or customize any of the VERSAbug functions as desired.

The dual port serial I/O cable assembly provided with the EPROM Resident Package is available as a separate product (M68KVMSIOC1).

TABLE 1 — Commands Available in 8K Byte ROM

COMMAND	DESCRIPTION
MD <addr1>[<count>]	Memory Display
MM <address> [;<opts>]	Memory Modify
MS <address> <data...>	Memory Set
.A0 - .A7 [<expression>]	Display/Set Address Register
.D0 - .D7 [<expression>]	Display/Set Data Register
.PC [<expression>]	Display/Set Program Counter
.SR [<expression>]	Display/Set Status Register
.SS [<expression>]	Display/Set Supervisor Stack Pointer
.US [<expression>]	Display/Set User Stack Pointer
DF	Display Formatted Registers (All)

TABLE 1 — Commands Available in 8K Byte ROM (continued)

COMMAND	DESCRIPTION
BF <address1> <address2> <word>	Block Fill (with 16 bit data word)
BM <address1> <address2> <address3>	Block Move
BI <address1> <address2>	Block Initialize
BT <address1> <address2>	Block of Memory Test
DC <expression>	Data Conversion
OF	Display Offsets
.R0 - .R6 [<expression>]	Display/Set Relative Offset Register
BR [<address>[;<count>]]	Breakpoint Set (up to 8)
NOBR [<address> <address> . . .]	Breakpoint Remove (any or all)
GO [<address>]	Go Until Breakpoint, Exception or Trace
GT <breakpoint address>	Same as GO (may be set Temporary Breakpoint)
GD [<address>]	Go Direct (No Breakpoint or Trace Set, and no Exception Vector Changes)
TR [<count>]	Trace Set (for number of instructions)
TT <breakpoint address>	Trace Set (Trace to Temporary Breakpoint)
PA	Printer Attach (Print as well as display)
NOPA	Reset Printer Attach
PF [<port number>]	Port Format (set Serial Port Attributes)
TM [<exit character>]	Transparent Mode (Two serial ports transparently connected)
* text....	Send Message to PORT2
HE	Help (Lists commands)
DU <address1> <address2> [<text..>]	Memory Dump ('S' Record Upline load)
LO [;<opts>] [=text]	Load ('S' Record Downline load)
VE [=text]	Verify ('S' Record Downline load verify)
BH [<device>,<controller>]	Bootstrap Halt (Boot and Halt)
BO [<device>,<controller>,<filename>]	Bootstrap Operating System (Boot and Go)
Command Line Edit and Control Functions:	
(BREAK)	Abort Command
(DEL)	Delete Character
(CTRL-D)	Redisplay Line
(CTRL-H)	Delete Character
(CTRL-W)	Suspend Output*
(CTRL-X)	Cancel Command Line
(cr)	Send Line to Memory

*When (CTRL-W) is used, the user can cause the output display to continue by entering any character.

TABLE 2 — Extended Functions and Commands Available in 4K Bytes Extended ROM

COMMAND	DESCRIPTION
BS <address1> <address2> <data> [<mask>] [;<option>]	Block Search (Search designated memory for specified data ... complete data used if no mask specified)
BD [<device>,<controller>]	Bootstrap Dump
IOP	I/O Physical dump to disk
RM	Register Modify (scroll mode of all Registers for Display/Set)
Error Message Features:	
Print System Exception Messages	

FIGURE 1 — System Configuration #1 — Monoboard with VERSAbug and Cross-Computer Serial Communications Link (EXORmacs, EXORciser or other computer with 'S' record files)

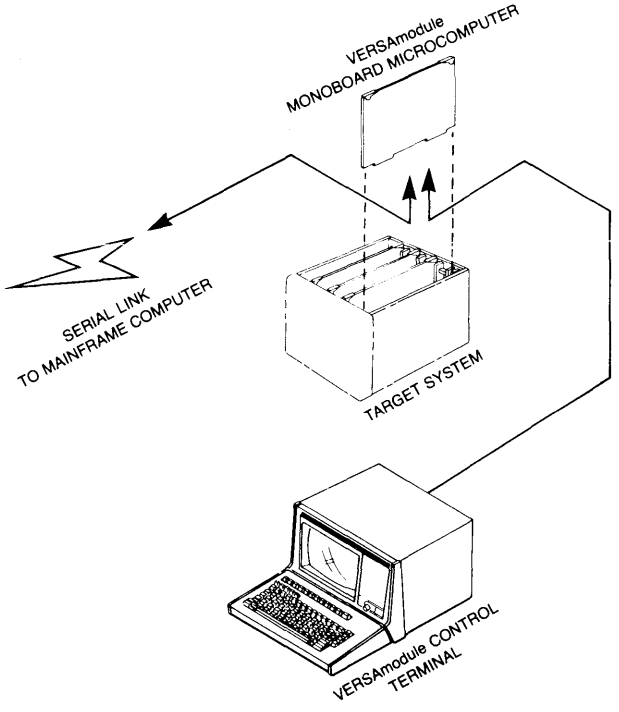
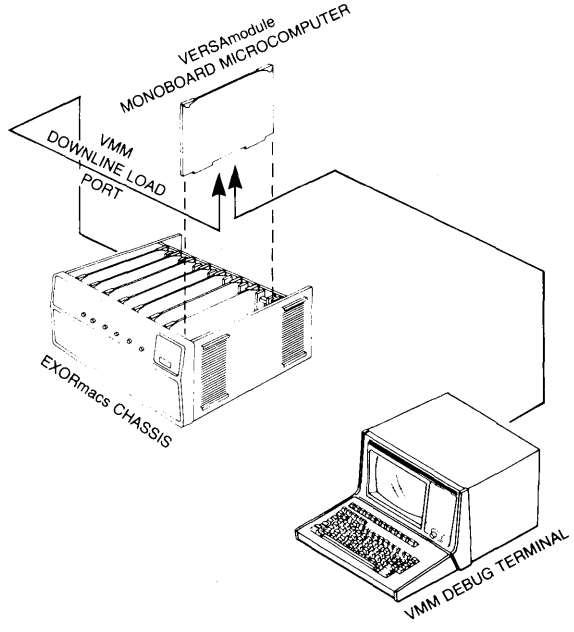


FIGURE 2 — Configuration #2 — Monoboard with VERSAbug in EXORmacs



Ordering Information

Part Number	Description
M68KVBUG (For M68KVM01A)	VERSAbug, the VERSAmodule Monoboard Microcomputer System Debug Package, includes EPROM set* and Dual RS-232C Serial I/O Cable Assembly allowing connection to user debug terminal and a up/downline load host. Includes User's Manual
M68KVBUGLF (For M68KVM01A)	Source and Relocatable Object Modules for the VERSAbug system on EXORmacs Diskette.* Includes User's Manual
M68KVBUGLC (For M68KVM01A)	Source and Relocatable Object Modules for the VERSAbug system on EXORmacs Cartridge Disk.* Includes User's Manual
M68KVMSIOC1	Dual RS-232C Serial I/O Cable Assembly for interconnecting between a VERSAbus 50 Pin I/O arrangement with pin out compatible to the VERSAmodule Monoboard Microcomputer (M68KVM01A and M68KVM02-3) and external RS-232C terminals. May be used to connect a debug terminal and an up/downline load host to a VMM
M68KVBUG2 (For M68KVM02-3)	VERSAbug, the VERSAmodule Monoboard Microcomputer System Debug Package, includes EPROM set* and Dual RS-232C Serial I/O Cable Assembly allowing connection to user debug terminal and a up/downline load host. Includes User's Manual
M68KVBUG2LF (For M68KVM02-3)	Source and Relocatable Object Modules for the VERSAbug system on EXORmacs Diskette.* Includes User's Manual
M68KVBUG2LC (For M68KVM02-3)	Source and Relocatable Object Modules for the VERSAbug system on EXORmacs Cartridge Disk.* Includes User's Manual
M68KVBUG2LMC (For M68KVM02-3)	VERSAbug Source and Relocatable Object Modules on VERSAdos LMD Cartridge for use with M68KVM02 Microcomputer. 16-Bit Languages, Utilities, etc.
M68KVBUG/D1	User's Manual

*The M68KVBUG and M68KVBUG2 EPROM sets are copyrighted by Motorola and may be copied only under prior written agreement from Motorola. M68KVBUGLF, M68KVBUGLC, M68KVBUG2LF, M68KVBUG2LC and M68KVBUG2LMC Sources are copyrighted and licensed by Motorola. They may be obtained only under the required license agreement with Motorola.

M68KVBUG3
M68KVBUG3LF
M68KVBUG3LC

VERSAbug 3.0 Debugging Package for the VERSAmodule 3 MC68010 Monoboard Microcomputer

VERSAbug 3.0 Resident Package

- EPROM resident system debug monitor
- Dual port RS-232C Serial I/O Cable Assembly allows connection of debug terminal and up/downline load host to the M68KVM03 (VM03) through the 50-pin I/O Connector arrangement of VERSAbus
- 44 debug, up/downline load and disk bootstrap load commands
- Full speed execution of system and user developed programs on the VM03
- Virtual terminal capability for up/downline load from an EXORmacs/EXORciser Development System or from a cross-computer
- Powerful software and system debug command set allows access to all VM03 I/O, control and memory facilities plus the full 16M byte direct address range of the VERSAbug system bus
- Disk Bootstrap load/dump from Motorola Floppy and Hard Disk Systems: M68KVM20, M68KVM21, and M68KVM22
- Includes all required installation and operation documentation

VERSAbug Source and Relocatable Object Module Package

- Source and Relocatable Object Modules for VERSAbug on diskette or cartridge disk
- Relocatable Object Modules allow the user to include only the VERSAbug items needed in their end system; to link in their own up/downline loader; and/or to link in their own bootstrap loader
- Source Modules allow user modification of VERSAbug as desired

The standard debug package for the VERSAmodule 3 Monoboard Microcomputer, VERSAbug 3.0, is available as two separate product offerings. VERSAbug comes as an EPROM-based resident package with dual serial I/O cable assembly ready for installation and immediate use with the VM03 in a VERSAbus based backplane. Such a backplane is provided by Motorola's Card Cage (M68KVMCC1), or Chassis. VERSAbug Source and Relocatable Object Modules are available as a separate product on either EXORmacs Development System compatible diskette, or disk cartridge.

VERSAbug provides a powerful evaluation, use and system debugging tool for VERSAmodule Systems. The EPROM Resident Package will operate in 32K bytes of ROM space. VERSAbug uses the first 1280 words of RAM storage for Interrupt Vectors and temporary storage. The EPROM resident package is delivered in two 16K byte EPROMs. Table 1 lists the commands available to the user.

The package permits full speed execution of system and user-developed programs in a VM03 system environment under complete operator control. The dual serial I/O cable assembly provided with VERSAbug allows terminal and host access to the two serial ports on the VM03. VERSAbug 3.0 may be utilized with a VM03 in a stand-alone environment with only a user provided standard RS-232C asynchronous ASCII terminal. Alternately, it may be used with the second serial I/O port direct connected to a host computer for up/downline loading of programs in Motorola "S" Record format. When directly connected to a host computer in this manner, the VM03/VERSAbug/Operator Terminal combination appears as a normal asynchronous ASCII terminal (a virtual terminal) to the host operating system. Figures 1 and 2 illustrate two typical configurations using the VERSAbug EPROM set installed in a VM03.

In a typical debug session, the user will download his developed program to a VM03 from the host computer used for software development. This may be a Motorola EXORmacs or EXORciser Development System. Following load, VERSAbug commands may be used to examine and modify memory, set breakpoints to run particular program segments, and trace program progress. The user

may set up and examine a variety of conditions using any of the powerful commands listed in Table 1, such as the Register Display/Set series and the memory block manipulation commands. The Data Conversion command serves as an aid in examining and modifying data by providing a means of converting hexadecimal to decimal, and decimal to hexadecimal. If corrections or program patches are required, these may be performed and checked in the VM03. The user may then save a corrected copy to the host computer files utilizing the Memory Dump command for upline load. Creating program patches may be aided by use of the Display Offsets command to assist with relocatable and position independent code. The user may also copy all traffic to the serial port debug terminal on a printer attached to one of the VM03 parallel ports by use of the Attach Printer command. This may be useful for desk debugging following a debug session.

The user may communicate directly with the host computer as a terminal for purposes other than up/downline load by executing the Transparent Mode command or the Send Message to Port 2 command. By using the Port Format command, the serial ports on the VM03 may be reconfigured for such attributes as baud rate, stop bits and number of data bits. In the Transparent Mode, the two serial ports must be operated at the same baud rate.

VERSAbug 3.0 may be used for debug in total systems environments including the VM03 together with other Motorola VERSAmodules (RAM, floppy and hard disk controllers, communications controllers, A/D controllers, etc.) as well as user-developed VERSAbus compatible modules.

Bootstrap load and dump commands permit the user to bootstrap from standard Motorola floppy and hard disk systems utilizing media with the Motorola EXORmacs diskette/disk format. The Boot Dump command permits the user to write his operating system to an EXORmacs diskette/disk in bootstrap load format for subsequent use in boot loading. The IOP command permits the user to create the EXORmacs diskette/disk format required.

The Source and Relocatable Object Module Packages provide the user with the information necessary to link VERSAbug 3.0 into their specific system in either modified, or unmodified form. The Relocatable Object Modules are designed to permit creation of a load module with or without the "S" Record up/downline loader, the VERSAbug disk bootstrap loader, and/or the Extended Functions package. The user may also use the Relocatable Object Modules substituting their own device bootstrap loader for the VERSAbug disk bootstrap loader.

Source Modules permit the user to modify, or customize any of the VERSAbug functions as desired.

The dual port serial I/O cable assembly provided with the EPROM Resident Package is available as a separate product (M68KVMSIOC1).

TABLE 1 — Available Commands

COMMAND	DESCRIPTION
MD <addr1>[<count>]	Memory Display
MM <address> [;<opts>]	Memory Modify
MS <address> <data...>	Memory Set
.A0 - .A7 [<expression>]	Display/Set Address Register
.D0 - .D7 [<expression>]	Display/Set Data Register
.PC [<expression>]	Display/Set Program Counter
.SR [<expression>]	Display/Set Status Register
.SS [<expression>]	Display/Set Supervisor Stack Pointer
.US [<expression>]	Display/Set User Stack Pointer
.VBR [<expression>]	Display/Set Vector Base Register
.DFC [<expression>]	Display/Set Destination Function Code Register
.SFC [<expression>]	Display/Set Source Function Code Register
DF	Display Formatted Registers (All)

TABLE 1 — Available Commands (continued)

COMMAND	DESCRIPTION
BF <address1> <address2> <word>	Block Fill (with 16 bit data word)
BM <address1> <address2> <address3>	Block Move
BI <address1> <address2>	Block Initialize
BS <address1> <address2> <data> [<mask>] [;<option>]	Block Search (Search designated memory for specified data . . . complete data used if no mask specified)
BT <address1> <address2>	Block of Memory Test
DC <expression>	Data Conversion
OF	Display Offsets
.R0 - .R6 [<expression>]	Display/Set Relative Offset Register
BR [<address>[;<count>]]	Breakpoint Set (up to 8)
NOBR [<address> <address> . . .]	Breakpoint Remove (any or all)
GO [<address>]	Go Until Breakpoint, Exception or Trace
GT <breakpoint address>	Same as GO (may be set Temporary Breakpoint)
GD [<address>]	Go Direct (No Breakpoint or Trace Set, and no Exception Vector Changes)
T	Trace One Instruction
TR [<count>]	Trace Set (for number of instructions)
TT <breakpoint address>	Trace Set (Trace to Temporary Breakpoint)
PA	Printer Attach (Print as well as display)
NOPA	Reset Printer Attach
PF [<port number>]	Port Format (set Serial Port Attributes)
TM [<exit character>]	Transparent Mode (Two serial ports transparently connected)
ST [<device>,<controller>]	Self Test Diagnostic
* text....	Send Message to PORT2
HE	Help (Lists commands)
DU <address1> <address2> [<text..>]	Memory Dump ('S' Record Upline load)
LO [;<opts>] [=text]	Load ('S' Record Downline load)
VE [=text]	Verify ('S' Record Downline load verify)
BD [<device>,<controller>]	Bootstrap Dump
BH [<device>,<controller>,<starting head no.>]	Bootstrap Halt (Boot and Halt)
BO [<device>,<controller>,<starting head no.>,<filename>]	Bootstrap Operating System (Boot and Go)
Command Line Edit and Control Functions:	
(BREAK)	Abort Command
(←)	Delete Character
(CTRL-D)	Redisplay Line
(CTRL-H)	Delete Character

TABLE 1 — Available Commands (continued)

COMMAND	DESCRIPTION
Command Line Edit and Control Functions (continued):	
(CTRL-W)	Suspend Output*
(CTRL-X)	Cancel Command Line
(cr)	Send Line to Memory
*When (CTRL-W) is used, the user can cause the output display to continue by entering any character.	
Extended Functions:	
IOP	I/O Physical read/write to disk
IOT	I/O "Teach" Disk Configuration to System
Error Message Features:	
Print System Exception Messages	

FIGURE 1 — System Configuration #1 — Monoboard with VERSAbug and Cross-Computer Serial Communications Link (EXORmacs, EXORciser or other computer with 'S' record files)

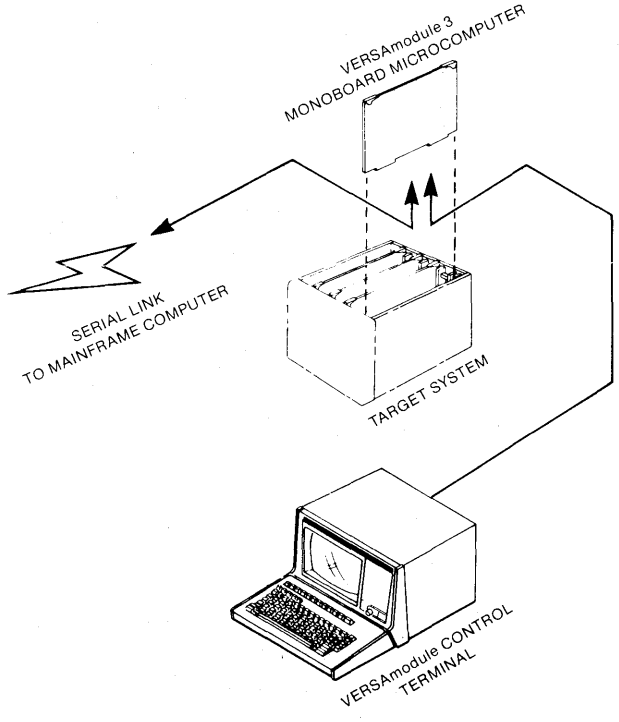
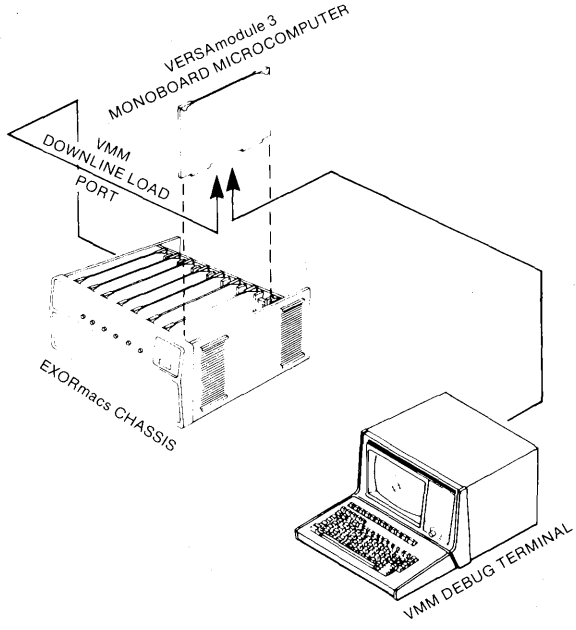


FIGURE 2 — Configuration #2 — Monoboard with VERSAbug in EXORmacs



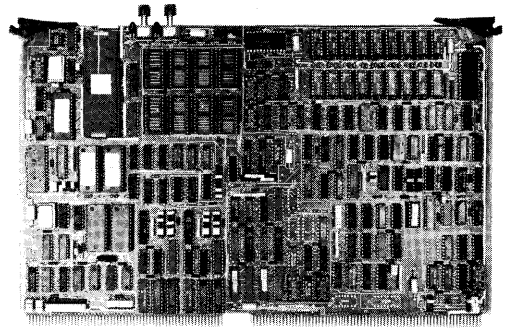
Ordering Information

Part Number	Description
M68KVBUG3	VERSAbug Debugging Package for use with VERSAmodule M68KVM03-1 Microcomputer, provided as an EPROM set* and dual port RS-232C Serial I/O Cable Assembly, provides 44 debug, up/downline load and disk bootstrap load commands for quick debug and execution of system and user developed programs. Includes User's Manual
M68KVBUG3LC	VERSAbug Source Code and Relocatable Object Modules on VERSAdos CMD Cartridge* for use with M68KVM03-1 Microcomputer. Includes User's Manual
M68KVBUG3LF	VERSAbug Source Code and Relocatable Object Modules on VERSAdos 8" Diskette* for use with M68KVM03-1 Microcomputer. Includes User's Manual
M68KVBUG3/D1	User's Manual

*The M68KVBUG3 EPROM set is copyrighted by Motorola and may be copied only under prior written agreement from Motorola. M68KVBUG3LF and M68KVBUG3LC Sources are copyrighted and licensed by Motorola. They may be obtained only under the required license agreement with Motorola.

M68KVM01A1
M68KVM01A2

VERSAmodule Monoboard Microcomputer



- MC68000 Advanced 16-Bit Microprocessor, offering the following features:
  - — 32-Bit Data and Address Registers (a total of 15)
  - — 16 Megabyte Direct Addressing Range
  - — 56 Powerful Instruction Types
  - — Operations On Five Main Data Types
  - — Memory Mapped I/O
  - — 14 Addressing Modes
  - — Architecturally Optimized for Efficient Support of High-Level Languages
- 8 MHz Microprocessor Clock Operation
- VERSAbus Interface for 16-bit Applications (including multiprocessor system architectures)
- 32K or 64K Bytes Dynamic RAM with Byte Parity
- 8 Sockets for up to 64K Bytes of User-Provided Pin-Compatible 2K, 4K, or 8K Byte ROM, PROM, or EPROM devices
- Private bus for on-board intercommunications between the MPU, ROM, RAM, Serial I/O, Parallel I/O, and Timer Resources
- User strappable to handle inputs on any or all of the VERSAbus Interrupt lines
- 2 Serial I/O Ports, each with RS-232C interface, supporting asynchronous format full duplex communications in the range of 50 to 19.2K baud. One of the two ports is programmable for synchronous operation with byte-oriented protocols, and is user-strappable for an RS-422 electrical interface. Baud rates to 1 Mbps can be used with external clock.
- 4 Parallel I/O Ports, bidirectional, each with 8 Data and 2 "Handshake" Lines
- Triple 16-Bit programmable timer/counter
- User strappable as System Controller, to provide the following system functions:
  - — VERSAbus Arbitration
  - — System Clock (16 MHz)
  - — System Reset
- System Test Switch and Board Fault Indicator LED
- RESET Switch and CPU Halt Indicator LED

The VERSAmodule Monoboard Microcomputer is a complete microcomputer system-on-a-board. At its heart is the powerful microprocessor representing a significant advance in 16-bit units — the MC68000. Its architecture is optimized for high level language support to foster rapid and economical program development.

The Monoboard Microcomputer in combination with the VERSAmodule Chassis and Real-time Multitasking Software (RMS68K) provides a complete design environment that frees the system designer to develop the software required for the unique I/O hardware of his application.

TABLE 1 — Specifications (continued)

Characteristics	Specifications
VERSAbus Interface Functions — (contd)	
	— Secondary Map Selection — Data/Program Map Selection — Supervisory/User Map Selection
Data Transfer Control	(a) Data Strobes (2) (b) Write (c) Address Strobe (d) Data Transfer Acknowledge (e) Bus Error
System Control	System RESET
Priority Interrupt Control	(a) Interrupt Request (7 Priority Lines) (b) Interrupt Acknowledge In (Daisy Chain) (c) Interrupt Acknowledge Out (Daisy Chain)
Bus Arbitration Control	(a) Bus Busy (b) Bus Clear (c) Bus Release (Emergency Bus Request) (d) Bus Request (5 Priority Lines) (e) Bus Grant In (Daisy Chain — 5 Lines) (f) Bus Grant Out (Daisy Chain — 5 Lines)
System Test	System Fail
Power Monitor	AC Failure
Misc. Functions	(a) System Clock (16 MHz — square wave) (b) AC Clock (50/60 Hz — square wave) (c) +5 Vdc (d) +12 Vdc (e) −12 Vdc
Operating Temperature	0° to 70°C
Humidity	0% to 95%, non-condensing
Physical Characteristics	
Height	9.25 in. (32.5 cm)
Width	14.5 in. (36.8 cm)
Thickness	0.6 in. (1.5 cm)
BUS MATING CONNECTOR TYPES	
VERSAbus Connector (P1)	Stanford Applied Eng'g CPH7000-140ST Micro Plastics, Inc. MP-0100-70-DW-5H
I/O Connector (P2)	Stanford Applied Eng'g CPH7000-120ST Micro Plastics, Inc. MP-0100-60-DW-5H
POWER REQUIREMENTS	**+5 Vdc** / **+12 Vdc** / **−12 Vdc**

	+5 Vdc		+12 Vdc		−12 Vdc	
	Typ	**Max**	**Typ**	**Max**	**Typ**	**Max**
Current Requirements: Without ROM/EPROM	5.7 A	8.2 A	0.5 A	0.8 A	0.1 A	0.2 A
Add for each EPROM *Pair* (Up to four pairs)	0.1 A	0.3 A	—	—	—	—
Supply Voltages	(a) +5 V ±5% (b) +12 V ±5% (c) −12 V ±5%					

TABLE 2 — Signal Characteristics, "BUS" Connector P1

Connector Pin	Signal Mnemonic	Functional Description	Signal Characteristics	
			Input	Output
1,2	+5 V	+5 Vdc Power	(See VERSAbus	(See VERSAbus
3,4	Gnd	Ground	Spec)	Spec)
5	D00*	Data Bit 0	(Note 1)	(Note 1)
6	D01*	Data Bit 1	(Note 1)	(Note 1)
7	D02*	Data Bit 2	(Note 1)	(Note 1)
8	D03*	Data Bit 3	(Note 1)	(Note 1)
9	D04*	Data Bit 4	(Note 1)	(Note 1)
10	D05*	Data Bit 5	(Note 1)	(Note 1)
11	D06*	Data Bit 6	(Note 1)	(Note 1)
12	D07*	Data Bit 7	(Note 1)	(Note 1)
13	D08*	Data Bit 8	(Note 1)	(Note 1)
14	D09*	Data Bit 9	(Note 1)	(Note 1)
15	D10*	Data Bit 10	(Note 1)	(Note 1)
16	D11*	Data Bit 11	(Note 1)	(Note 1)
17	D12*	Data Bit 12	(Note 1)	(Note 1)
18	D13*	Data Bit 13	(Note 1)	(Note 1)
19	D14*	Data Bit 14	(Note 1)	(Note 1)
20	D15*	Data Bit 15	(Note 1)	(Note 1)
21,22	—	(Reserved)	(Note 1)	(Note 1)
23,24	GND	Ground	(Note 1)	(Note 1)
25	DS0*	Data Strobe 0	(Note 1)	(Note 1)
26	DS1*	Data Strobe 1	(Note 1)	(Note 1)
27,28	GND	Ground	(Note 1)	(Note 1)
29	DTACK*	Data Transfer Acknowledge	(Note 1)	(Note 1)
30	AS*	Address Strobe	(Note 1)	(Note 1)
31,32	GND	Ground	(Note 1)	(Note 1)
33	—	(Reserved)	(Note 1)	(Note 1)
34	WRITE*	Read/Write Indicator	(Note 1)	(Note 1)
35	—	(Reserved)	(Note 1)	(Note 1)
36	A01*	Address Bit 1	(Note 1)	(Note 1)
37	A02*	Address Bit 2	(Note 1)	(Note 1)
38	A03*	Address Bit 3	(Note 1)	(Note 1)
39	A04*	Address Bit 4	(Note 1)	(Note 1)
40	A05*	Address Bit 5	(Note 1)	(Note 1)
41	A06*	Address Bit 6	(Note 1)	(Note 1)
42	A07*	Address Bit 7	(Note 1)	(Note 1)
43	A08*	Address Bit 8	(Note 1)	(Note 1)
44	A09*	Address Bit 9	(Note 1)	(Note 1)
45	A10*	Address Bit 10	(Note 1)	(Note 1)
46	A11*	Address Bit 11	(Note 1)	(Note 1)
47	A12*	Address Bit 12	(Note 1)	(Note 1)
48	A13*	Address Bit 13	(Note 1)	(Note 1)
49	A14*	Address Bit 14	(Note 1)	(Note 1)
50	A15*	Address Bit 15	(Note 1)	(Note 1)
51	A16*	Address Bit 16	(Note 1)	(Note 1)
52	A17*	Address Bit 17	(Note 1)	(Note 1)
53	A18*	Address Bit 18	(Note 1)	(Note 1)
54	A19*	Address Bit 19	(Note 1)	(Note 1)
55	A20*	Address Bit 20	(Note 1)	(Note 1)
56	A21*	Address Bit 21	(Note 1)	(Note 1)

(1) For electrical characteristics and additional information, refer to VERSAbus Specification Manual, M68KVBS.

* Signal categories are defined on page 17.

FIGURE 1 — Memory Map

FFFFFE F70032	(OFF-BOARD)	FFFFFF F70033
F70030	STATUS/CONTROL REGISTER	F70031
F7002E F70020	PARALLEL PORTS	F7002F F70021
F7001E	SERIAL PORT #2	F7001F F70019
	SERIAL PORT #1	F70017 F70011
F70000	PTM	F7000F F70001
F6FFFE F10000	(OFF-BOARD)	F6FFFF F10001
F0FFFE F00000	16K/32K/64K ON-BOARD EPROM/ROM OR 32K/64K ON-BOARD RAM	F0FFFF F00001
EFFFFE 010000	(OFF-BOARD)	EFFFFF 010001
00FFFE 000008	32K/64K ON-BOARD RAM OR 16K/32K/64K ON-BOARD EPROM/ROM	00FFFF 000009
000006 000000	ON-BOARD RESET VECTORS (ROM)	000007 000001
← MOST SIGNIFICANT BYTE →		← LEAST SIGNIFICANT BYTE →

FIGURE 2 — Block Diagram

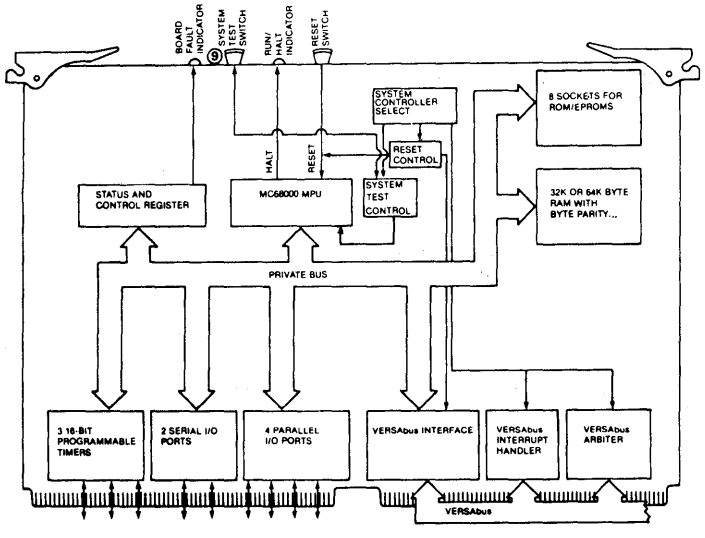


TABLE 1 — Specifications

Characteristics	Specifications
Microprocessor	MC68000L
Clock Frequency	32 MHz, Crystal Controlled, providing 8 MHz to the MPU, and 16 MHz to the VERSAbus "System Clock" line.
Data Bus Width	16 Bits
Address Bus Width	24 Bits
Instructions	56 Variable Length Instructions (Up to 10 Bytes)
Addressing Modes	Fourteen Addressing Modes: (A) Register Direct Modes — Data Register Direct — Address Register Direct (B) Memory Address Modes — Address Register Indirect (ARI) — ARI w/Post Increment — ARI w/Pre Decrement — ARI w/Displacement — ARI w/Index (C) Special Address Modes — Absolute Short Address — Absolute Long Address — Program Counter w/Displacement — Program Counter w/Index — Immediate Data, Byte — Immediate Data, Word — Immediate Data, Long Word
Registers	**No. — Type — Size** 8 — DATA REGISTERS — (32-BIT) 7 — ADDRESS REGISTERS — (32-BIT) 2 — STACK POINTERS — (32-BIT) 1 — PROGRAM COUNTER — (32-BIT) 1 — STATUS REGISTER — (16-BIT) See the MC68000 16-Bit Microprocessor User's Manual for additional details (MC568000UM)
Memory Capabilities	
Total Directly Addressable (on-board and off-board)	16,777,216 Bytes
ROM/PROM/EPROM (user-supplied)	Eight 24-pin sockets provided for 2K, 4K, or 8K byte +5 V devices. User selects part density and "access time range" by strap option. Total ROM capacity is 64K bytes when using 8K byte devices.
Dynamic RAM (on-board)	32K Bytes (M68KVM01A1), or 64K Bytes (M68KVM01A2) with byte parity and on-board refresh control circuitry. Byte parity generation and checking may be activated or deactivated by user strap option.
Memory Timing	On-board and off-board RAM, ROM and I/O timing are best stated in terms of "wait" clock periods that must be added per read or write cycle associated with the instruction execution time of each particular MC68000 instruction. For these MPU instruction execution times, see Appendix D of the MC68000 MPU User's Manual (MC68000UM(AD2)).

The VERSAmodule Monoboard Microcomputer features provide maximum utility in a wide spectrum of industrial automation, lab automation, and general information processing applications. The principal features of this microcomputer board illustrated in the Block Diagram (Figure 2) include:

Local Memory

Both ROM and RAM facilities are provided for direct access by the MC68000 processor. The RAM is offered in two sizes as a purchase option: 32K bytes or 64K bytes. It is implemented as dynamic RAM using 4116-type parts with on-board automatic refresh control, plus byte parity with automatic retry. Eight sockets are also provided for user-supplied ROM devices, which may be of the industry standard pinout 5-volt-only type of ROM, PROM, or EPROM up to 64K bytes maximum capacity. The user specifies by a strap option whether 2K, 4K or 8K byte ROM devices are being used.

Serial and Parallel I/O

Two independent and versatile serial ports are provided, each capable of several programmable modes of operation with the RS-232C terminal or MODEM interface in asynchronous operation to 19.2K baud. One of the two serial ports can be configured under program control for synchronous operation with byte-oriented protocols (such as Bi-Sync), and may be strapped to serve as an RS-422 standard interface. Use of external clocking with this second serial port permits bit transfer rates in excess of 19.2K baud. Parallel I/O is implemented as four independent bidirectional ports, each having 8 data lines and 2 "handshake" lines, for a total of 32 lines of general purpose parallel I/O capability. Implementation as 16-bit ports is a strapping option. All serial and parallel I/O functions are presented at the bottom-edge I/O connector, P2.

Programmable Timer/Counter

Several powerful timer/counter functions are provided by the MC6840 PTM (Programmable Timer Module) device. Three independent 16-bit counters are available to support applications such as frequency measuring, event counting, and interval measuring. The PTM is accessible from the MC68000, and can provide interrupts to the MPU upon occurrence of specified events. External gating and signalling lines used by the PTM are available at the P2 connector, along with a ground pin for each signal pin. Internal input timing options permit use of ac line clock, 2 MHz system clock, plus other timing modes.

Status/Control Register

This 16-bit wide register permits software to perform certain control activities, and to read status for several board functions:

- Issue interrupt on VERSAbus
- Set interrupt mask bits for on-board serial and parallel I/O ports
- Block Transfer Request
- Emergency and Normal VERSAbus Request
- Emergency and Normal VERSAbus Grant

Memory Mapped I/O

Memory addresses in the range of F70001 (Hex) to F70031 are allocated in the system memory map to data, status, and control registers in the on-board serial I/O, parallel I/O, and PTM functional modules. See the memory map in Figure 1.

VERSAbus Interface

A VERSAbus Interface is incorporated in the VERSAmodule Monoboard Microcomputer to allow its use in a high performance system requiring additional off-board resources such as RAM and intelligent I/O controllers. VERSAbus is characterized by asynchronous, bidirectional operation and supports Direct Memory Access (DMA) and multiprocessor system operation. The VERSAbus Interface supports the full 16-megabyte address range of the MC68000 MPU. All VERSAbus data, address, and control lines for 16-bit system applications are present on the 140-pin connector, P1. All I/O lines for the VERSAmodule Monoboard Microcomputer supporting the Serial I/O, Parallel I/O, and PTM functions, are present on the 120-pin connector, P2.

Private Bus

The MPU, RAM, ROM, serial I/O, parallel I/O, PTM, and Status and Control Register elements are interconnected by a private bus which is connected in turn to the VERSAbus *only* if the MC68000 is engaged in access of off-board resources. The private bus feature allows processing to continue at full speed on the monoboard microcomputer, while VERSAbus activities (such as DMA to off-board RAM from a disk) occur simultaneously.

Bus Request

The MC68000 gains access to the VERSAbus resources upon becoming bus master through the process of BUS REQUEST. The priority level of the BUS REQUEST is selectable to one of five levels by user strap option. There are two methods by which the BUS REQUEST can be made. The *indirect,* or software transparent method, and the *direct* method by specific request made through the Status and Control Register.

Indirect Request — When the program attempts to access an off-board memory location while the board is not currently the bus master, the memory decode logic automatically initiates a BUS REQUEST at the user selected priority level. When the memory access is complete, VERSAbus mastership is automatically released. The indirect method

is not only software transparent but permits multiple processor boards to cycle-steal bus resources following each bus access cycle.

Direct Request — Alternately, a BUS REQUEST can be initiated through the Status and Control Register under direct program control. When using the direct request method, bus mastership is retained by the board until released under program control via the Status and Control Register. The direct method permits maximum speed block transfers to be performed by a board. However, the block transfer can be interrupted by a higher priority BUS REQUEST.

System Controller Functions

By means of a user strap option, the Monoboard Microcomputer may be configured as a "System Controller" to provide system management and control functions:

- VERSAbus Arbitration — The system controller element accepts bus requests from various bus masters on five bus request priority levels, and issues a "bus grant" back to the highest priority requester. This function facilitates orderly management of the contention for bus mastership on the VERSAbus.
- System Clock — A 16 MHz clock signal is provided to other VERSAbus devices for various counting and synchronizing tasks.
- RESET — Upon being placed in the RESET state, the Monoboard Microcomputer will additionally drive the RESET line in the VERSAbus if it is acting as "system controller." An on-board RESET switch can be used to initiate this function.

System Controller functions are normally provided by *only one* module plugged into the VERSAbus backplane. If *more than one* VERSAmodule Monoboard Microcomputer is used in a multiprocessor system, only one of those boards is designated as System Controller.

VERSAbus Interrupter

The Monoboard Microcomputer is capable of generating VERSAbus interrupts under software control. When this interrupt is acknowledged by the SYSTEM INTERRUPT HANDLER, a software-specified vector number is provided to the MPU being interrupted.

System Interrupt Handler

The Monoboard Microcomputer can be strapped to respond to VERSAbus interrupt requests appearing on any or all of the seven priority level lines, issuing the interrupt acknowledge and receiving the interrupt vector number. When more than one monoboard microcomputer is used on VERSAbus in a multiprocessor configuration, each monoboard can be configured to handle a different subset of all VERSAbus interrupt lines.

Power-Down Monitor Features

The Monoboard Microcomputer monitors the VERSAbus "AC Fail" line that can be driven from an external power fail sense module. A user strap option allows a level change on the AC Fail line to generate a non-maskable interrupt to the MC68000 MPU on this board. This feature allows user-provided power-down and power-up firmware routines to perform whatever system-wide activities would be appropriate, such as storing away critical data in non-volatile RAM (elsewhere on the VERSAbus) in the event of a power-down condition.

Self-Test Hardware Features

The Monoboard Microcomputer provides both a self-test button and a "fail" light at the top board edge to facilitate user on-site maintenance activities.

When depressed, the TEST BUTTON performs two functions. One is to cause a level 2 auto-vector interrupt at the MC68000 MPU. The other is to light the "Fault Indicator" LED at the top board edge.

The interrupted MPU can then execute a user-provided self-test routine which turns off the LED at the top board edge if all tests complete successfully. If the failure indicator LED remains lit at the end of this sequence, it alerts maintenance personnel of a detected failure condition. This visual indicator feature significantly facilitates removal and replacement of defective modules to minimize system downtime.

TABLE 1 — Specifications (continued)

<table>
<tr><th>Characteristics</th><th>Specifications</th></tr>
<tr><td>Memory Capabilities — (cont'd)
On-Board Accesses</td><td>
DEVICE — # of "WAIT" clock periods (typical)
<table>
<tr><th></th><th>Write</th><th>Read</th><th>Read With Parity Check*</th></tr>
<tr><td>— RAM MC4116BC20</td><td>3</td><td>3</td><td>4</td></tr>
<tr><td>— ROM/EPROM (2716-25; 250 ns access)</td><td>—</td><td>1</td><td>—</td></tr>
<tr><td>— Serial I/O</td><td>1</td><td>2</td><td>—</td></tr>
<tr><td>— Parallel I/O</td><td>0</td><td>1</td><td>—</td></tr>
</table>
*Does not include automatic retry on parity error.
</td></tr>
<tr><td>Off-Board Accesses</td><td>
Slave Board Access Times and Wait State Pairs
<table>
<tr><th colspan="2">SLAVE BOARD ACCESS TIME (ns)</th><th colspan="2">MONOBOARD CLOCK WAIT STATE PAIRS</th></tr>
<tr><th>Min</th><th>Max</th><th>READ</th><th>WRITE</th></tr>
<tr><td>0</td><td>110</td><td>2</td><td>1</td></tr>
<tr><td>111</td><td>235</td><td>3</td><td>2</td></tr>
<tr><td>236</td><td>360</td><td>4</td><td>3</td></tr>
<tr><td>361</td><td>485</td><td>5</td><td>4</td></tr>
<tr><td>486</td><td>610</td><td>6</td><td>5</td></tr>
<tr><td>611</td><td>735</td><td>7</td><td>6</td></tr>
<tr><td>736</td><td>860</td><td>8</td><td>7</td></tr>
<tr><td>861</td><td>985</td><td>9</td><td>8</td></tr>
</table>
Assumptions:
1. Off-board access time is referenced at the remote board edge connector, and is the time from Data Strobe asserted to DTACK asserted. (Read or write access time is specified on the remote board data sheet.)
2. No bus arbitration is required. (The processor holds the bus between access cycles.) If the monoboard gives up the bus between access cycles, wait states will be indeterminate because the bus may never be granted to the monoboard, or the bus may be granted to it immediately.
3. VERSAbus propagation delay is assumed to be 10 ns.
</td></tr>
<tr><td>Memory Map User Options</td><td>On-board ROM and RAM may be located in the address map as per the following strap-selectable options:
Option #1:
RAM Base Address 000008 (Hex)
ROM Base Address F00000
Option #2:
RAM Base Address F00000
ROM Base Address 000000</td></tr>
<tr><td>Input/Output Capabilities
Serial I/O</td><td>Two programmable serial I/O ports are provided, implemented by 2661-type devices. All electrical lines are presented on the bottom-edge I/O connector, P2. Interrupts are provided from these ports to the MPU on</td></tr>
</table>

TABLE 1 — Specifications (continued)

Characteristics	Specifications
Input/Output Capabilities — (contd)	
	a user-strappable priority level. Baud rates for both ports are software programmable to standard rates in the range of 50 to 19.2K baud, as follows:
Baud Rates	**Programmable Baud Rates**: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200
Port Options	Port "1" — RS-232C, asynchronous only — Terminal or MODEM (user-strap selectable) Port "2" — RS-232C or RS-422 (user-strap selectable) — Terminal or MODEM (user-strap selectable, RS-232C Only) — Asynchronous or Synchronous Operation (user-programmable) — Synchronous Operation in excess of 19.2K baud is possible with externally-provided clock.
Parallel I/O	Four bidirectional ports are provided, each with 8 data lines and 2 "handshake" lines. Each port may be written or read individually by the MPU, or each of the 2 "port pairs" may be written or read 16 bits at a time. The ports may be individually strap-selected for output-only or for input-only. User strap options for each of the ports include the following: (A) input/output configuration (B) polarity of peripheral control output lines (hi-strobe or low strobe) (C) polarity of interrupt flags and peripheral control input lines (interrupt on rising or falling edge) (D) interrupt flag clearing options Interrupts from all of the parallel ports are wire-ORed together, and are provided to the MPU on a user-selectable priority level (4, 5 or 6). Each one of the four parallel ports may be individually masked through the Board Status/Control Register. The MPU may poll each Status Register associated with a parallel port, to determine which of the four ports generated an interrupt.
Programmable Timer/Counter (PTM)	Implemented by means of an MC6840 device, which provides three 16-bit programmable binary counters. Each 16-bit section may be operated independently, or multiple sections may be cascaded to provide 32-bit or 48-bit operation under control of the MPU. The following 3 lines from *each* of the 16-bit counter sections (a total of 9 lines) are provided at the VERSAbus I/O connector, P2: (a) Gate Input (b) Clock Input (c) PTM Output

TABLE 1 — Specifications (continued)

<table>
<tr><th>Characteristics</th><th>Specifications</th></tr>
<tr><td>Programmable Timer/Counter (PTM)
— (contd)</td><td>User strap options allow the following choices of clock functions to be provided to the PTM:
(a) "AC Clock" (from VERSAbus)
(b) Baud rate clocks from the on-board serial port baud rate generators.
(c) Address strobe from the on-board private bus.
(d) 2 MHz clock (derived from the on-board 8 MHz MPU clock)
(e) Externally-provided time/count source from VERSAbus connector, P2.
The PTM Interrupt output is provided to the MPU on a user-selectable priority level.
See MC6840 Fundamentals and Applications Manual (MC6840UM(AD)) for further details.</td></tr>
<tr><td>Interrupts
Auto-Vectored, "Local"</td><td>Interrupts from the following sources provide auto-vectored interrupts to the MPU:
<table><tr><th>Interrupt Level</th><th>Interrupt Source</th></tr><tr><td>1</td><td>"Bus Clear" (From VERSAbus)</td></tr><tr><td>2</td><td>Test Button</td></tr><tr><td>3</td><td>Interrupter Vector Read</td></tr><tr><td>4, 5, or 6*</td><td>PTM</td></tr><tr><td>4, 5, or 6*</td><td>Serial I/O Ports</td></tr><tr><td>4, 5, or 6*</td><td>Parallel I/O Ports</td></tr><tr><td>7</td><td>"AC Fail" (from VERSAbus) if System Controller</td></tr><tr><td>7</td><td>"Bus Release" (from VERSAbus) if not System Controller</td></tr></table>*Strap Selectable
Note: Interrupts from all serial ports are wire-OR'd together to provide one interrupt, as are interrupts from all parallel ports.
Interrupts from the on-board serial and parallel ports are individually maskable through the Board Status/Control Register.</td></tr>
<tr><td>VERSAbus Interrupts</td><td>Any or all of the seven Interrupt Request Lines from the VERSAbus can be strapped to generate the equivalent interrupt priority level at the on-board MPU. In responding to a VERSAbus interrupt, the monoboard microcomputer must (a) request and gain bus mastership, (b) acknowledge the VERSAbus interrupt request, and (c) accept the interrupt vector from the requesting device. (In addition, the MPU responds to six levels of "on-board" interrupts. When on-board and off-board interrupts at the level occur at the same time, they are both acknowledged but the on-board interrupt is acknowledged first.)</td></tr>
<tr><td>Interrupter</td><td>A VERSAbus interrupt may be generated at a jumper selectable level by writing to the Status/Control Register. Five bits of the eight-bit vector number may be set by this same control register. The remaining three bits are set by jumpers on the board.</td></tr>
</table>

TABLE 1 — Specifications (continued)

Characteristics	Specifications
System Controller Functions	These functions are activated only if the board is strap-selected as"System Controller"
VERSAbus Arbitration	— Accepts bus requests from potential bus masters on any of five VERSAbus priority line levels. — Issues a "Bus Clear" to the current bus master if a bus request is received at a higher priority level than that of the current bus master. — Issues a "Bus Grant" back to the highest priority requester when the bus is clear.
System Clock	VERSAbus "System Clock" line driven with 16 MHz signal.
Reset	VERSAbus "System Reset" line driven upon occurrence of either of the following conditions: 1. Manual initiation of the RESET button on the top board edge. 2. Power-up
Power Down Provision	If the board is configured as System Controller, it monitors the "AC Fail" line on the VERSAbus. When "AC Fail" is asserted, an auto-vectored non-maskable interrupt is generated on-board. It also generates a "Bus Release" signal onto VERSAbus.
Self-Test	
Hardware Activation Modes	Manual depression of self-test button at top board edge.
Internal Function	(a) Interrupts on-board MPU at priority level 2, and sets a bit in Board Status/Control Register. Generates auto-vectored interrupt request. (b) Lights the "Failure Indicator" LED at the top board edge. (c) Under software control, by writing a bit into the Board Status/Control Register, the "Failure Indicator" LED can be turned off.
Board Status/Control Register	
Size	16 bits
Control Outputs	(a) Interrupt mask bits for Parallel I/O Ports (4) and Serial I/O Ports (2). (b) Emergency and Normal VERSAbus Request (c) Turn-off"Failure Indicator" LED at the top board edge. (d) System Fail (drives VERSAbus line) (e) Mask all interrupts to on-board MPU (f) Interrupt another master and supply vector number (g) Perform block transfer
Status Outputs	(a) Emergency VERSAbus Grant (b) System Fail (senses VERSAbus line) (c) Board selected as System Controller (d) Manual Test Request (Test Button)
VERSAbus Interface Functions	The following VERSAbus functions implemented on the VERSAmodule Monoboard Microcomputer represent a *subset* of the complete VERSAbus function set. Note: For additional VERSAbus details, refer to the VERSAbus Specification Manual (M68KVBS(D2)).
Data	16 Lines
Address	23 Lines
Address Modifiers	8 Lines, providing the following functions: — I/O Map Selection — Extended Map Selection — Interrupt Acknowledge Map Selection

TABLE 2 — Signal Characteristics, "BUS" Connector P1 (continued)

Connector Pin	Signal Mnemonic	Functional Description	Signal Characteristics	
			Input	Output
57	A22*	Address Bit 22	(Note 1)	(Note 1)
58	A23*	Address Bit 23	(Note 1)	(Note 1)
59	AM4*	Address Modifier Bit 4	(Note 1)	(Note 1)
60	AM7*	Address Modifier Bit 7	(Note 1)	(Note 1)
61	GND	Ground	(Note 1)	(Note 1)
62	GND	Ground	(Note 1)	(Note 1)
63	AM3*	Address Modifier Bit 3	(Note 1)	(Note 1)
64,65,66	—	(Reserved)	(Note 1)	(Note 1)
67,68	GND	Signal Ground	(Note 1)	(Note 1)
69	ACCLK	Power Line Frequency (ac clock)	(Note 1)	(Note 1)
70	SYSCLK	16 MHz Clock	(Note 1)	(Note 1)
71,72	GND	Ground	(Note 1)	(Note 1)
73	—	(Reserved)	(Note 1)	(Note 1)
74	SYSRESET*	System Reset	(Note 1)	(Note 1)
75,76,77	—	(Reserved)	(Note 1)	(Note 1)
78	ACFAIL*	AC Input Power Failure	(Note 1)	(Note 1)
79	—	(Reserved)	(Note 1)	(Note 1)
80	SYSFAIL*	System Fail	(Note 1)	(Note 1)
81	BERR*	Bus Error	(Note 1)	(Note 1)
82	—	(Reserved)	(Note 1)	(Note 1)
83	AM0*	Address Modifier Bit 0	(Note 1)	(Note 1)
84	AM1*	Address Modifier Bit 1	(Note 1)	(Note 1)
85	AM2*	Address Modifier Bit 2	(Note 1)	(Note 1)
86	AM6*	Address Modifier Bit 6	(Note 1)	(Note 1)
87	IRQ1*	Interrupt Request 1	(Note 1)	(Note 1)
88	IRQ2*	Interrupt Request 2	(Note 1)	(Note 1)
89	IRQ3*	Interrupt Request 3	(Note 1)	(Note 1)
90	IRQ4*	Interrupt Request 4	(Note 1)	(Note 1)
91	IRQ5*	Interrupt Request 5	(Note 1)	(Note 1)
92	IRQ6*	Interrupt Request 6	(Note 1)	(Note 1)
93	IRQ7*	Interrupt Request 7	(Note 1)	(Note 1)
94	AM5*	Address Modifier Bit 5	(Note 1)	(Note 1)
95	ACKIN*	Acknowledge In	(Note 1)	(Note 1)
96	ACKOUT*	Acknowledge Out	(Note 1)	(Note 1)
97	BG0IN*	Bus Grant In, #0	(Note 1)	(Note 1)
98	BG0OUT*	Bus Grant Out, #0	(Note 1)	(Note 1)
99	BG1IN*	Bus Grant In, #1	(Note 1)	(Note 1)
100	BG1OUT*	Bus Grant Out, #1	(Note 1)	(Note 1)
101	BG2IN*	Bus Grant In, #2	(Note 1)	(Note 1)
102	BG2OUT*	Bus Grant Out, #2	(Note 1)	(Note 1)
103	BG3IN*	Bus Grant In, #3	(Note 1)	(Note 1)
104	BG3OUT*	Bus Grant Out, #3	(Note 1)	(Note 1)
105	BG4IN*	Bus Grant In, #4	(Note 1)	(Note 1)
106	BG4OUT*	Bus Grant Out, #4	(Note 1)	(Note 1)
107	BR0*	Bus Request, #0	(Note 1)	(Note 1)
108	BR1*	Bus Request, #1	(Note 1)	(Note 1)
109	BR2*	Bus Request, #2	(Note 1)	(Note 1)
110	BR3*	Bus Request, #3	(Note 1)	(Note 1)

(1) For electrical characteristics and additional information, refer to VERSAbus Specification Manual M68KVBS.
* Signal categories are defined on page 17.

TABLE 2 — Signal Characteristics, "BUS" Connector P1 (continued)

Connector Pin	Signal Mnemonic	Functional Description	Signal Characteristics	
			Input	Output
111	BR4*	Bus Request, #4	(Note 1)	(Note 1)
112	BBSY*	Bus Busy	(Note 1)	(Note 1)
113	BCLR*	Bus Clear	(Note 1)	(Note 1)
114	BREL*	Bus Release	(Note 1)	(Note 1)
115	—	(Reserved)	(Note 1)	(Note 1)
116	—	(Reserved)	(Note 1)	(Note 1)
117	—	(Reserved)	(Note 1)	(Note 1)
118	—	(Reserved)	(Note 1)	(Note 1)
119,120	GND	Ground	(Note 1)	(Note 1)
121,122	−12 V	−12 Vdc Power	(Note 1)	(Note 1)
123,124	GND	Ground	(Note 1)	(Note 1)
125,126	+12 V	+12 Vdc Power	(Note 1)	(Note 1)
127,128	+12 V	+12 Vdc Power	(Note 1)	(Note 1)
129,130	+5 V	+5 Vdc Power	(Note 1)	(Note 1)
131,132	+5 V	+5 Vdc Power	(Note 1)	(Note 1)
133,134	—	(Reserved)	(Note 1)	(Note 1)
135–140	GND	Ground	(Note 1)	(Note 1)

(1) For electrical characteristics and additional information, refer to VERSAbus Specification Manual, M68KVBS(D2).
* Signal categories are defined on page 17.

TABLE 3 — Signal Characteristics, I/O Connector P2

Connector Pin	Signal Mnemonic	Functional Description	Signal Characteristics	
			Input	Output
1–6	GND	Ground	—	—
7–10	+5 V	+5 Vdc Power	—	—
11,12	+12 V	+12 Vdc Power	—	—
13,14	GND	Ground	—	—
15,16	−12 V	−12 Vdc Power	—	—
17	GND	Ground	—	—
18	CLOCK 1	PTM Clock 1 Input	C*	(N/A)
19	TXD1	Transmit Data, Serial Port #1	E**	E
20	GATE 1	PTM Gate 1 Input	D	(N/A)
21	RXD1	Received Data, Serial Port #1	E**	E
22	OUTPUT 1	PTM #1 Output	(N/A)	B
23	RTS1	Request to Send, Serial Port #1	E**	E
24	CLOCK 2	PTM Clock 2 Input	D	(N/A)
25	CTS1	Clear to Send, Serial Port #1	E**	E
26	GATE 2	PTM Gate 2 Input	D	(N/A)
27	DSR1	Data Set Ready, Serial Port #1	E**	E
28	OUTPUT 2	PTM #2 Output	(N/A)	B
29	GND	Ground	—	—
30	RR+	Receiver Ready +, Serial Port #2 (RS-422)	F	(N/A)
31	DCD1	Data Carrier Detect, Serial Port #1	E**	E
32	RR−	Receiver Ready, Serial Port #2 (RS-422)	F	(N/A)

*Signal categories are defined following this table.
**Input/Output characteristics vary depending on whether this port is defined as a Terminal or Modem.

TABLE 3 — Signal Characteristics, I/O Connector P2 (continued)

Connector Pin	Signal Mnemonic	Functional Description	Signal Characteristics	
			Input	Output
33	DTR1	Data Terminal Ready, Serial Port #1	E**	E
34	TR+	Terminal Ready +, Serial Port #2 (RS-422)	(N/A)	F
35		(Unused)	—	—
36	TR−	Terminal Ready −, Serial Port #2 (RS-422)	(N/A)	F
37		(Unused)	—	—
38	DM+	Data Mode +, Serial Port #2 (RS-422)	F	(N/A)
39	OUTPUT 3	PTM #3 Output	(N/A)	B
40	DM−	Data Mode −, Serial Port #2 (RS-422)	F	(N/A)
41	GATE 3	PTM Gate 3 Input	D	(N/A)
42	CS+	Clear to Send +, Serial Port #2 (RS-422)	F	(N/A)
43	CLOCK 3	PTM Clock 3 Input	D*	(N/A)
44	CS−	Clear to Send −, Serial Port #2 (RS-422)	F	(N/A)
45	GND	Ground	—	—
46	RT+	Receive Timing +, Serial Port #2 (RS-422)	F	(N/A)
47	TXD2	Transmit Data, Serial Port #2 (RS-232C)	E**	E
48	RT−	Receive Timing −, Serial Port #2 (RS-422)	F	(N/A)
49	RXD2	Received Data, Serial Port #2 (RS-232C)	E**	E
50	RS+	Request to Send +, Serial Port #2 (RS-422)	(N/A)	F
51	RTS2	Request to Send, Serial Port #2 (RS-232C)	E**	E
52	RS−	Request to Send −, Serial Port #2 (RS-422)	(N/A)	F
53	CTS2	Clear to Send, Serial Port #2 (RS-232C)	E**	E
54	RD+	Receive Data +, Serial Port #2 (RS-422)	F	(N/A)
55	DSR2	Data Set Ready, Serial Port #2 (RS-232C)	E**	E
56	RD−	Receive Data −, Serial Port #2 (RS-422)	F	(N/A)
57	GND	Ground	—	—
58	ST+	Send Timing +, Serial Port #2 (RS-422)	F	(N/A)
59	DCD2	Data Carrier Detect, Serial Port #2 (RS-232C)	E**	E
60	ST−	Send Timing −, Serial Port #2 (RS-422)	F	(N/A)

*Signal categories are defined following this table.
**Input/Output characteristics vary depending on whether this port is defined as a Terminal or Modem.

TABLE 3 — Signal Characteristics, I/O Connector P2 (continued)

Connector Pin	Signal Mnemonic	Functional Description	Signal Characteristics	
			Input	Output
61	DTR2	Data Terminal Ready, Serial Port #2 (RS-232C)	E**	E
62	SD+	Send Data +, Serial Port #2 (RS-422)	(N/A)	F
63	RXC2	Receive Data Clock, Serial Port #2 (RS-232C)	E**	E
64	SD−	Send Data −, Serial Port #2 (RS-422)	(N/A)	F
65	TXC2	Transmit Data Clock, Serial Port #2 (RS-232C)	E**	E
66	GND	Ground	—	—
67–70		(Unused)	—	—
71	P1CB2	Parallel Port 1, CB2 Control Line	(N/A)	B
72,73	GND	Ground	—	—
74	P2CA1	Parallel Port 2, CA1 Control Line	B*	(N/A)
75	P1CB1	Parallel Port 1, CB1 Control Line	B	(N/A)
76,77	GND	Ground	—	—
78	P2CA2	Parallel Port 2, CA2 Control Line	(N/A)	B
79	P1PB7	Parallel Port 1, Upper Data Byte, Bit #7	A	A
80	GND	Ground	—	—
81	P1PB6	Parallel Port 1, Upper Data Byte, Bit #6	A	A
82	P2PA0	Parallel Port 2, Lower Data Byte, Bit #0	A	A
83	P1PB5	Parallel Port 1, Upper Data Byte, Bit #5	A	A
84	P2PA1	Parallel Port 2, Lower Data Byte, Bit #1	A	A
85	P1PB4	Parallel Port 1, Upper Data Byte, Bit #4	A	A
86	P2PA2	Parallel Port 2, Lower Data Byte, Bit #2	A	A
87	P1PB3	Parallel Port 1, Upper Data Byte, Bit #3	A	A
88	P2PA3	Parallel Port 2, Lower Data Byte, Bit #3	A	A
89	P1PB2	Parallel Port 1, Upper Data Byte, Bit #2	A	A
90	P2PA4	Parallel Port 2, Lower Data Byte, Bit #4	A	A
91	P1PB1	Parallel Port 1, Upper Data Byte, Bit #1	A	A
92	P2PA5	Parallel Port 2, Lower Data Byte, Bit #5	A	A
93	P1PB0	Parallel Port 1, Upper Data Byte, Bit #0	A	A

*Signal categories are defined following this table.
**Input/Output characteristics vary depending on whether this port is defined as a Terminal or Modem.

TABLE 3 — Signal Characteristics, I/O Connector P2 (continued)

Connector Pin	Signal Mnemonic	Functional Description	Signal Characteristics	
			Input	Output
94	P2PA6	Parallel Port 2, Lower Data Byte, Bit #6	A	A
95	P1PA7	Parallel Port 1, Lower Data Byte, Bit #7	A	A
96	P2PA7	Parallel Port 2, Lower Data Byte, Bit #7	A	A
97	P1PA6	Parallel Port 1, Lower Data Byte, Bit #6	A	A
98	P2PB0	Parallel Port 2, Upper Data Byte, Bit #0	A	A
99	P1PA5	Parallel Port 1, Lower Data Byte, Bit #5	A*	A
100	P2PB1	Parallel Port 2, Upper Data Byte, Bit #1	A*	A
101	P1PA4	Parallel Port 1, Lower Data Byte, Bit #4	A*	A
102	P2PB2	Parallel Port 2, Upper Data Byte, Bit #2	A*	A
103	P1PA3	Parallel Port 1, Lower Data Byte, Bit #3	A*	A
104	P2PB3	Parallel Port 2, Upper Data Byte, Bit #3	A*	A
105	P1PA2	Parallel Port 1, Lower Data Byte, Bit #2	A*	A
106	P2PB4	Parallel Port 2, Upper Data Byte, Bit #4	A*	A
107	P1PA1	Parallel Port 1, Lower Data Byte, Bit #1	A*	A
108	P2PB5	Parallel Port 2, Upper Data Byte, Bit #5	A*	A
109	P1PA0	Parallel Port 1, Lower Data Byte, Bit #0	A*	A
110	P2PB6	Parallel Port 2, Upper Data Byte, Bit #6	A	A
111	GND	Ground	—	—
112	P2PB7	Parallel Port 2, Upper Data Byte, Bit #7	A	A
113	P1CA2	Parallel Port 1, CA2 Control Line	(N/A)	B
114,115	GND	Ground	—	—
116	P2CB1	Parallel Port 2, CB1 Control Line	B	(N/A)
117	P1CA1	Parallel Port 1, CA1 Control Line	B	(N/A)
118,119	GND	Ground	—	—
120	P2CB2	Parallel Port 2, CB2 Control Line	(N/A)	B

*Signal categories are defined following this table.

TABLE 4 — Signal Category Definitions

Signal Type "A"					
Input	**Min**	**Max**	**Output**	**Min**	**Max**
Allowed Input Voltage	0 V	7.0 V	Guaranteed High Voltage When Sourcing 14.6 mA	2.0	—
Allowed Input For "High"	2.0 V	—	Guaranteed Low Voltage When	—	0.5
"Low"	—	0.8 V	Sinking 23.8 mA		
Current Sinked When Driven					
"High"	—	40 μA			
"Low"	—	220 μA			

Signal Type "B"					
Input	**Min**	**Max**	**Output**	**Min**	**Max**
Allowed Input Voltage	0 V	7.0 V	Guaranteed High Voltage When Sourcing 15 mA	2.0 V	—
Allowed Input For "High"	2.0 V	—	Guaranteed Low Voltage When	—	0.5 V
"Low"	—	0.8 V	Sinking 24 mA		
Current Sinked When Driven					
"High"	—	20 μA			
"Low"	—	690 μA			

Signal Type "C"			Signal Type "D"		
Input	**Min**	**Max**	**Output**	**Min**	**Max**
Allowed Input Voltage	−0.3 V	V_{CC}	Allowed Input Voltage	−0.3	V_{CC}
Allowed Input For "High"	2.0 V	—	Allowed Input For "High"	2.0 V	—
"Low"	—	0.8 V	"Low"	—	0.8 V
Current Sinked When Driven			Current Sinked When Driven		
"High"	—	2.5 μA	"High"	—	20 μA
"Low"	—	2.5 μA	"Low"	—	700 μA

Signal Type "E" (RS-232C Levels)					
Input	**Min**	**Max**	**Output**	**Min**	**Max**
Allowed Input Voltage	−30 V	30 V	Guaranteed High Voltage (Space) Across 3K Load	7.0 V	—
Allowed Input For "Space"	3.0 V	—	Low Voltage (Mark)	−7.0 V	—
"Mark"	−3.0 V	—			
Current Sinked When "Space" (V_{on} = 25 V)	—	8.3 mA			
Current Sinked When "Mark" (V_{off} = −25 V)	—	−8.3 mA			

TABLE 4 — Signal Category Definitions (continued)

Signal Type "F" (RS-422 Levels)					
Input	**Min**	**Max**	**Output**	**Min**	**Max**
Allowed Differential Voltage	—	± 25 V	Guaranteed High Voltage When	2.5 V	—
Allowed Common Mode Voltage	—	± 15 V	Sourcing 20 mA		
Differential Threshold Voltage ($-7\ V \leq VI \leq 7\ V$)	± 0.2 V	—	Guaranteed Low Voltage Voltage When Sinking 48 mA	—	0.5 V
Input Current:					
For VI = 10 V	—	3.25 mA			
−10 V	—	3.25 mA			

SOFTWARE/FIRMWARE SUPPORT

Motorola provides standard software packages to support the VERSAmodule Monoboard Microcomputer, within the categories of Real-Time Executives and Operating Systems, Debuggers/Loaders, and Self-Test Systems. The principal features of these software products are as follows.

RMS68K — Real-Time Multitasking System Software

- Memory-Resident (ROMable)
- Physical (Channel) I/O
- Multitask Dynamic Scheduling
- Software and Hardware Interrupt Processing
- High-Speed Interrupt Response
- Intertask Communication and Task Synchronization
- Dynamic Allocation and Management of RAM
- User Trap Handling
- Exception Processing
- Time Delay, Periodic Task Activation, Time-Of-Day
- Easy Addition Of User-Written Device Drivers
- Upward Compatible To Real-Time Disk Operating System
- Compatible with EXORmacs System Software
- Customization via SYSGEN

VERSAdos — Real-Time Disk Operating System

- Provides All Real-Time Multitasking Software Features Of RMS68K
- Device Independent I/O and Logical I/O
- Wait and Proceed Mode I/O
- Standard Device Drivers
- Multi-Level File Directories
- Shared File Access
- Dynamic or Contiguous File Space Allocation
- Random, Sequential, and Indexed Sequential File Access

VERSAbug — Debug/Monitor/Loader Firmware

- Initialization
- Display/Change Memory
- Display/Change Registers
- Set and Clear Breakpoints
- Block Initialize
- Block Move
- Search for a (Masked) Value
- TRACE with optional instruction count
- Downline Load

VERSAmodule — Self-Test Software

This package of self-test routines for the monoboard microcomputer will be available to assist the user in customizing a total self-test capability for his specific system. Anticipated availability of this software is mid-1981.

HARDWARE/SOFTWARE DEVELOPMENT SUPPORT

The recommended development system vehicle for developing microcomputer systems based around the VERSAmodule Monoboard Microcomputer is the EXORmacs MC68000 Development System. EXORmacs is a multiuser development system (floppy disk or hard disk based) with advanced software development tools including a Macro Assembler, Pascal Compiler, CRT Editor, and Linkage Editor. Since EXORmacs also provides the VERSAbus interconnect structure and the VERSAdos operating systems VERSAmodule applications can be easily developed and checked out in the EXORmacs chassis and transferred to the target system for final debug.

Cross software support in the form of a Macro Assembler, Pascal Compiler, and Linkage Editor is also planned for IBM host computers.

SYSTEM EXPANSION

The VERSAmodule Monoboard Microcomputer is system-compatible with a growing family of VERSAmodule products:

Dynamic RAM Module — available in 32K, 64K, and 128K byte versions. Includes byte parity with automatic retry on parity error.

Universal Disk Controller (UDC) — A 2-board set that provides industry standard SMD interface to hard disk drives (up to 2 drives of 96 megabytes each) and floppy disk drives (up to 4 drives of 0.5 megabyte each).

Floppy Disk Controller (FDC) — A single board that provides an interface to up to four double-sided single-density floppy disk drives of 0.5 megabyte each.

Multichannel Communications Module (MCCM) — Provides four asynchronous serial ports, each with RS-232C interface, plus an industry-standard parallel printer interface port.

Universal Intelligent Peripheral Controller (UIPC) — Provides IPC architecture on a single board with a DMA channel to global memory on the VERSAbus. A parallel interface is provided to which a user may interface a special device such as tape, disk, or high-speed communications controllers.

The UDC, FDC, MCCM, and UIPC modules each provide a consistent electrical and logical interface to the VERSAbus, and to VERSAdos or RMS68K system software. This allows simplified device-independent I/O for the main application.

PACKAGING AND ACCESSORIES

VERSAmodule-based applications can be conveniently packaged in and supported by a family of hardware accessories. (For additional information on packaging options and power supplies, refer to the chassis/card cage data sheet MVMCH3-1.)

Four-Slot Card Cage — Mechanically and electrically expandable up to 3 units (12 Slots)

Four-Slot Chassis — Provides card cage, power supply, fans, enclosure, and rack-mount or table-top usage capability.

Power Supplies — Low-Power (15 A @ 5 Vdc) and high-power (30 A @ 5 Vdc) versions are available. Each includes an ac power failure detection capability with VERSAbus interface.

VERSAbus Adapter Module — An interface module for adapting 8-bit EXORbus boards into the 16-bit VERSAbus systems.

VERSAbus Extender Module — Enables extension of VERSAbus modules for serving, testing, trouble-shooting, and debugging.

VERSAbus Wirewrap Module — Permits construction and incorporation of custom circuits into VERSAbus system.

Ordering Information

Part Number	Description
M68KVM01A1	VERSAmodule Monoboard Microcomputer. This module contains the MC68000L MPU, 32K bytes of Dynamic RAM, sockets for up to 64K bytes of ROM/EPROM, 2 serial ports, 4 byte-wide parallel ports, a triple programmable timer/counter, VERSAbus interface, and System Controller features. Includes User's Manual.
M68KVM01A2	Same as M68KVM01A1, but includes 64K bytes of Dynamic RAM.
M68KVM01A/D1	VERSAmodule Monoboard Microcomputer User's Manual.

Related Documentation

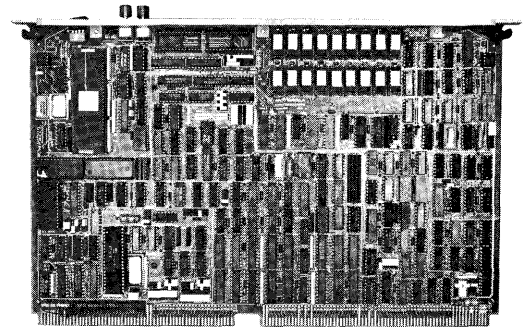
M68RIOCS	I/O Channel Specification Manual
M68KVBS	VERSAbus Specification Manual
MC68000UM(AD-3)	MC68000 Microprocessor User's Manual
M6840UM	MC6840 Fundamentals and Applications Manual

Ordering Information (continued)

Optional Accessories

Part Number	Description
M68KVM10-3	128K Byte VERSAbus Dynamic RAM Module
M68KVM21	Universal Disk Controller (UDC)
M68KVM20	Floppy Disk Controller (FDC)
M68KVM30	Multi-Channel Communications Module (MCCM)
M68KVM60	Universal Intelligent Peripheral Controller (UIPC)
M68KORMS68K	Real-Time Multitasking Software
M68K0VDOS	Real-Time Disk Operating System
M68KVBUG	Debug/Monitor/Loader Firmware
M68KVMCC1	4-Slot VERSAmodule Card Cage
M68KVMCH1-1	4-Slot VERSAmodule Chassis with 123 Watt Linear Power Supply
M68KVMCH1-2	4-Slot VERSAmodule Chassis with 228 Watt Switching Power Supply
MVMCH1-2	Same as M68KVMCH1-2 with 5-Slot Card Cage for Single Width Eurocard format I/O Cards
M68KVAM	VERSAbus Adapter Module
M68KEXT	VERSAbus Extender Module
M68KWW	VERSAbus Wirewrap Module

VERSAmodule Monoboard Microcomputer



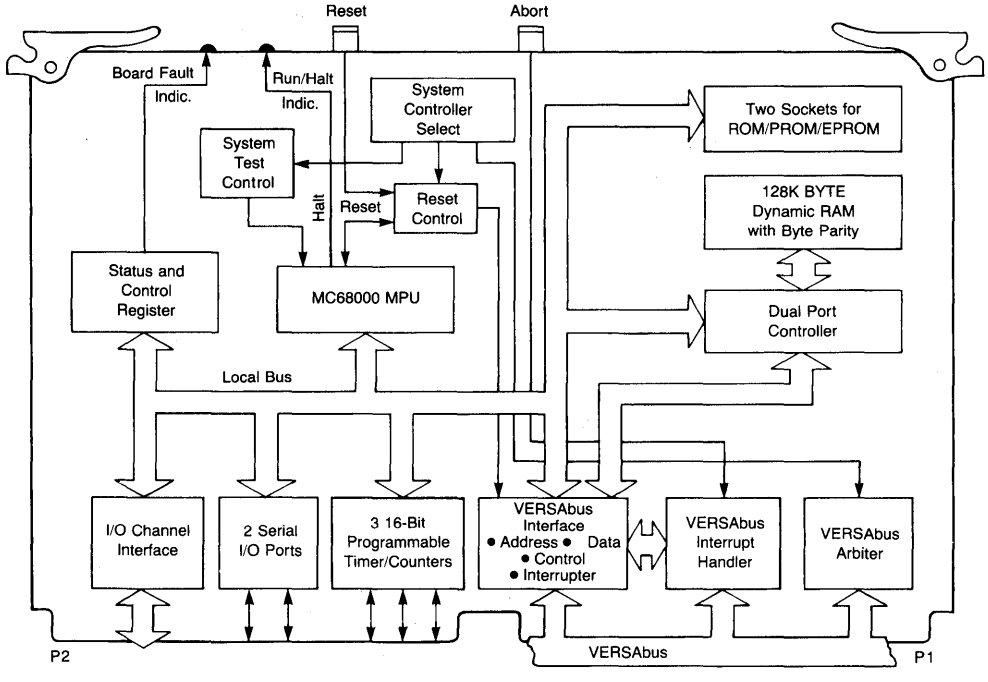
- MC68000 8 MHz 16-bit MPU
  - — 16 32-bit data, address and stack registers
  - — 14 addressing modes
  - — 16 megabyte direct addressing range
  - — Memory mapped I/O
  - — 56 powerful instruction types
  - — Operations on five data types including bit, byte, word, long word and BCD
  - — Provides interlock instruction for multiprocessor systems
  - — 256 multilevel vectored interrupts including exceptions, traps and external interrupts
  - — Architecturally optimized for efficient support of high-level languages.
- VERSAbus system bus compatibility with bus arbitration logic.
- Local on-board bus for intercommunications between the MPU, ROM, RAM, serial I/O and timer/counter resources as well as interface to VERSAbus.
- I/O Channel for interfacing off-board resources such as A/D, discrete I/O and parallel I/O to the monoboard microcomputer.
- 128K byte Dynamic RAM with shared memory access from local bus and VERSAbus via a dual port controller. Byte parity with automatic retry is a jumper option. RAM may be strapped to operate from VERSAbus +5 Vdc standby power for external battery backup. Power fail write inhibit logic is included.
- Two 28-pin sockets for up to 64K bytes of user provided 2, 4, 8, 16 or 32K byte ROM/PROM/EPROM devices.
- Two multiprotocol serial I/O ports with RS-232C interface selectable for MODEM or terminal use. Asynchronous and synchronous byte-oriented protocols (including IBM Bi-sync) as well as SDLC and HDLC bit-oriented protocols are supported. Internal clock rates strappable from 50 bps to 19.2 kbps. External clock rates to 600 kbps supported.
- Three 16-bit programmable timer/counters. All three are cascadable. When not programmed to create an interrupt the timer may be programmed to issue an output to an external device. By jumper selection, time/count inputs can be connected to:
  - — Serial port baud rate clocks
  - — 2 MHz clock
  - — VERSAbus ac line clock
  - — External input
- 0° C–70° C operating temperature range

The VM02 VERSAmodule Monoboard Microcomputer is a complete microcomputer system-on-a-board. At its heart is the powerful microprocessor representing a significant advance in 16-bit units — the MC68000. Its architecture is optimized for high-level language support to foster rapid program development.

The Monoboard Microcomputer in combination with the VERSAmodule Chassis and Real-time Multitasking Software (RMS68K) provides a complete design environment that frees the system designer to develop the software required for the unique I/O hardware of his application.

Many powerful features equip VM02 for application in a wide spectrum of industrial automation and general information systems. For example, its shared RAM permits efficient DMA operation with VERSAbus Intelligent Peripheral Controller (IPC) Modules as well as intercommunications between multiple monoboards and processors in complex systems. Figure 1 diagrams the major functional components of the Monoboard Microcomputer.

FIGURE 1 — M68KVM02 Block Diagram



I/O Channel

An I/O Channel lets small, single function, non-VERSAbus boards be easily added to enlarge the microcomputer function. One such board can be directly connected to the monoboard itself. Other boards, externally mounted, can access the I/O Channel from the VERSAbus backplane connectors. These can be connected using up to 12′ of multidrop ribbon cable.

Serial I/O Ports

The VM02 serial port 1 and serial port 2 are independent communication channels each comprising a parallel-to-serial conversion section using RS-232 serial drivers and a serial-to-parallel section using RS-232 serial receivers.

The Transmit Clock (TXC) input of each port can be connected by jumper selection to the port baud rate generator to obtain any of 16 data transmission rates. Or the input can

be synchronized with an external serial receiver via serial link. Similarly, jumper selected connection of the Receiver Clock (RXC) input to the baud rate generator can provide any of 16 data reception rates. Or the rate can be set by synchronization with an external transmitter via serial link.

For interfacing at high data rates over longer distances than those provided by RS-232 devices, the TXC and RXC inputs on port 2 are supported at TTL levels permitting use of a user-supplied TTL-to-RS-422 adapter board powered by VM02 via port 2.

Both ports are software configurable to support asynchronous and synchronous protocols. Synchronous protocols include monosync and bisync Character Oriented Protocols (COP) and SDLC and HDLC Bit Oriented Protocols (BOP). Parity checking is software selectable for all modes and CRC operations are supported for the COP and BOP procotols.

All interrupts from the serial ports are routed to the MPU over a single line the priority level of which can be strapped for level six or wire-wrapped for levels 1–5. A serial interrupt cycle is completed by the MPU causing the dual-ported RAM section to place a program supplied vector number on the data lines.

Several interrupt-causing modes can exist within each port. The condition-affects-vector mode can be enabled so that the port status register can be read to determine the cause of an interrupt. When the mode is disabled, various registers must be examined for the interrupt cause. Interrupt causing conditions are cleared via software commands sent to the serial port control register.

Programmable Timer

Each of the three separate programmable 16-bit timers within an MC6840 Programmable Timer (PTM) device can operate in any of four modes: 16-bit continuous, single shot, period measurement or puise width measurement. Each timer can be cascaded with another and programmed to use the internal or external clock. This PTM versatility equips the VM02 for straightforward application in environments requiring pulse generation, interval and period measurement, industrial timing control and programmable one-shot functions.

The timers appear in the VM02 memory map at locations F70001 to F7000F. Only the lower bytes are used.

Local Memory

Jumper selection allows the 28-pin ROM/PROM/EPROM sockets to be used for 24-pin 2716/2732 devices or 28-pin 16K byte and 32K byte devices. Jumper selection of VERSAbus Data Acknowledge (DTACK) response time permits devices of various speeds to be used. Device access times can range from 0 to 500 ns.

The on-board RAM is fully accessible to the processor via one port of the dual port controller. For access from the VERSAbus interface via the second controller port, the RAM base address is PROM configurable on 1K byte boundaries within a 256K byte jumper selectable block of VERSAbus space. Thus on-board RAM appears as a separate RAM board to other modules on the VERSAbus.

The 1K byte blocks can be individually configured as:

— Local RAM

— Shared Read/Write RAM

— Shared Read-Only RAM

— Shared Program Write Protectable RAM

RAM blocks configured as local are shielded from VERSAbus access. Blocks configured as shared program-protectable can be write protected by the on-board processor under program control via the control register. This feature can provide protected operation following an initial bootstrap load.

VERSAbus Interface

VERSAbus is characterized by asynchronous, bidirectional operation and support of Direct Memory Access (DMA), multiprocessor operation and the full 16 megabyte address range of the MC68000 MPU. Design requiring an expanded microcomputer function can utilize the VERSAbus interface to add other resources such as RAM and intelligent I/O controllers. Pins for all address, data, and control lines are provided in the 140-pin VERSAbus connector, P1. The 120-pin connector P2 provides interface to the serial I/O and programmable timer functions and to monoboard microcomputer support of the I/O Channel.

Local Bus

On board functional components are interconnected by a local bus which is connected to the VERSAbus interface, the I/O Channel interface or one of the serial ports when the on-board MPU accesses an off-board resource. This feature allows monoboard microcomputer processing to proceed simultaneously with the activities of another bus master.

Bus Request

Normal access to off-board VERSAbus resources is provided the MPU by means of a five-priority-levels bus request method. The monoboard level is strap selectable.

For normal access, VERSAbus mastership is gained in one of two ways:

Direct Request — A program can use the VM02 status and control registers to insure bus mastership prior to performing a function requiring access to a VERSAbus resource. Bus mastership is retained until released by the program via the status and control register. The direct request method permits a board to transfer blocks of data at the maximum rate.

Indirect Request — A program can also access a VERSAbus resource without first insuring bus mastership through

use of the status and control register. In this case if the monoboard is not currently bus master, a BUS REQUEST at the strap-selected priority level is automatically issued. On completion of the access, the bus is automatically released by the monoboard. This software-transparent indirect method provides a means by which, in a multiple processor-board system, each processor can access memory on a cycle-by-cycle basis.

VERSAbus Interrupter

A VERSAbus interrupter function provides a means of communication between monoboard microcomputers in a multiprocessor environment or between a monoboard and other interrupt-handling boards. Under program control, the monoboard MPU can cause the VERSAbus interrupter to generate an interrupt request signal by writing a value in the interrupt bits in the control register. The signal is placed on the one of seven interrupt lines corresponding to the specified priority level.

System Interrupt Handler

VM02 response to VERSAbus interrupts is configured by strap option. Any combination of the seven priority levels can be selected. In a multiple monoboard environment, this allows a unique set of levels to be chosen for each monoboard on the VERSAbus.

On recognizing an interrupt of valid priority, the interrupt handler requests control of the VERSAbus. When granted, the handler initiates an interrupt acknowledge cycle then passes to the MPU the vector number placed on the VERSAbus data lines by the interrupting board.

Power-Down Monitor

VM02 monitors the VERSAbus AC Fail line which can be driven from an external power fail sense module. If the nonmaskable interrupt priority level is strap selected, a low level on the AC Fail line will cause the MPU to be interrupted. This feature allows user-provided firmware routines to take emergency measures. These might include saving critical data in non-volatile VERSAbus RAM or local RAM powered from the 5.0 V standby line on the VERSAbus (a jumper option).

System Controller Functions

When configured by strap option as system controller, VM02 provides the following system management and control functions:

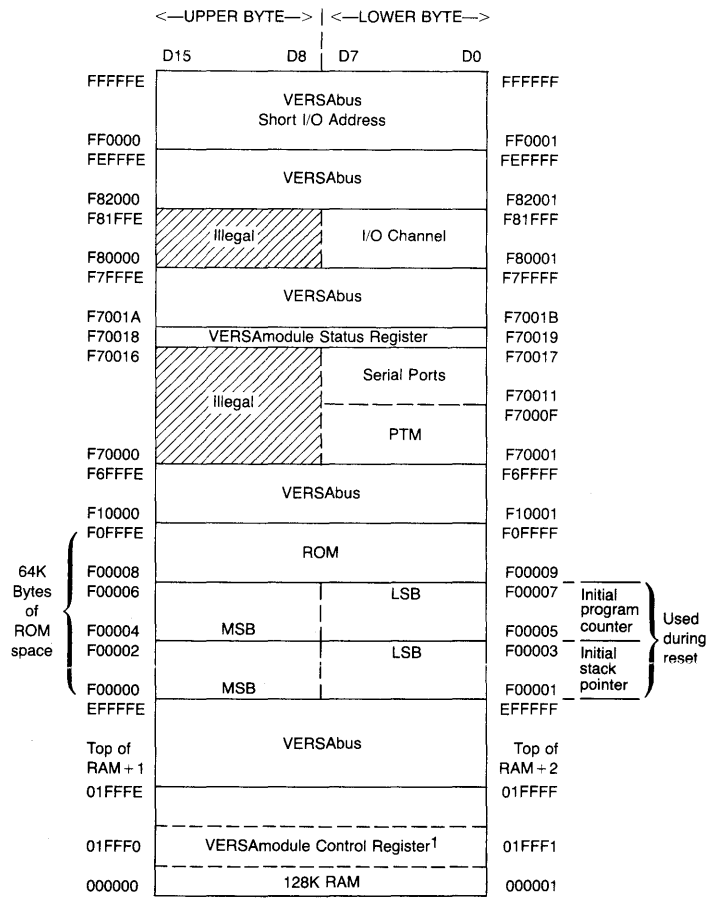
- VERSAbus Arbitration — The arbiter accepts bus requests on five priority levels from other bus masters and grants the bus to the highest priority requester. This function facilitates orderly management of the contention for bus mastership on the VERSAbus.
- System Clock — A 16 MHz clock signal is provided to other VERSAbus devices for various counting and synchronizing tasks.
- Reset — On entering the reset state, the monoboard microcomputer additionally drives the reset line in the VERSAbus low.
- Bus Timeout — Bus timeout generates a bus error (BERR) when a nonexistent bus address is placed on the VERSAbus. Timeout is selectable as 8, 16, 32, or 64 microseconds, or can be disabled.

System controller functions are normally provided by only one module plugged into the VERSAbus backplane. If more than one VERSAmodule Monoboard Microcomputer is used in a multiprocessor system, only one can be strapped as system controller.

Memory Mapped I/O

Memory addresses in the range of F70000 (Hex) through F70019 are allocated to the status register, the serial I/O ports, and to the PTM modules. The memory map of the VM02 monoboard microcomputer is shown in Figure 2.

FIGURE 2 — Memory Map



1. Control Register image only. Register not directly accessible.

TABLE 1 — VM02 Specifications

Characteristics	Description
Microprocessor	MC68000
Clock Frequency	32 MHz, crystal controlled, providing 8 MHz to the MPU, and 16 MHz to the VERSAbus "System Clock" line.
Data Bus Width	16 Bits
Address Bus Width	24 Bits
Instructions	56 Variable Length Instructions (From 2 to 10 bytes)
Addressing Modes	Fourteen Addressing Modes
Registers	19 Registers (Data, Address, Stack Pointer, Program Counter, Status)
	See the MC68000 16-Bit Microprocessor User's Manual for additional details.
Memory Capabilities	
Total Directly Addressable (on-board and off-board)	16,777, 216 Bytes
ROM/PROM/EPROM (user-supplied)	Two 28-pin sockets are provided for 2K, 4K, 8K, 16K or 32K byte devices using +5.0 Vdc only. Total ROM capacity is 64K bytes.
ROM Base Address	F00000 (Fixed)
Dynamic RAM (on-board)	128K bytes with on-board refresh control circuitry.
Error Checking	Byte parity generation and checking with automatic retry and generation of bus error on fail may be activated or deactivated by user strap option.
Battery Backup	Jumper option battery backup capability (user must provide +5.0 volt standby power and the system requires the power fail monitor M68KVMPM1).
Base Address	PROM configurable on 1K byte boundaries within one jumper selectable 256K byte block of VERSAbus space.
Access Timing (on-board)	Number of MPU Wait Cycles for accessing the on-board RAM.

8 MHz MPU Access Cycle	No. of Wait Cycles Typ.	Max	MPU Cycle Time Typ.	Max.
Write	0	1	625 ns	750 ns
Read with parity detection disabled	2	3	750 ns	875 ns
Read with parity detection enabled	3	4	875 ns	1000 ns

NOTE: Times shown assume no memory access contentions from the VERSAbus or the refresh circuitry.

TABLE 1 — VM02 Specifications (continued)

<table>
<tr><th>Characteristics</th><th>Description</th></tr>
<tr><td>Access Timing
(off-board)</td><td>Slave Board Access Times vs. Wait State Pairs
<table>
<tr><th colspan="2">Slave Board Access Times (ns)</th><th colspan="2">8 MHz MC68000 Wait State Pairs</th></tr>
<tr><th>Min.</th><th>Max.</th><th>Read</th><th>Write</th></tr>
<tr><td>0</td><td>70</td><td>1</td><td>0</td></tr>
<tr><td>71</td><td>195</td><td>2</td><td>1</td></tr>
<tr><td>196</td><td>320</td><td>3</td><td>2</td></tr>
<tr><td>321</td><td>445</td><td>4</td><td>3</td></tr>
<tr><td>446</td><td>570</td><td>5</td><td>4</td></tr>
<tr><td>571</td><td>695</td><td>6</td><td>5</td></tr>
<tr><td>696</td><td>820</td><td>7</td><td>6</td></tr>
<tr><td>821</td><td>945</td><td>8</td><td>7</td></tr>
</table>
Assumptions:
1. Slave board access time is referenced at the edge connector for that particular board. Access time is the time from a data strobe (DS0* or DS1*) low to DTACK* low.
2. No bus arbitration is required (the MPU holds the bus between cycles). If the VM02 gives up the bus between access cycles, the numbers of wait states is indeterminate.
3. VERSAbus propagation delay is assumed to be 10 ns.
4. VM02 provides the following delays from AS* to DS1*/DS0* low with an 8 MHz processor.
0 ns min. to 50 ns max. (read cycle)
0 ns min. to 85 ns max. (write cycle)</td></tr>
<tr><td>Serial I/O Ports</td><td>Two NEC 7201-Implemented serial I/O ports selectable for terminal or MODEM use.</td></tr>
<tr><td>Interface</td><td>RS-232C. Support inputs TXC and RXC of Port 2 at TTL levels.</td></tr>
<tr><td>Protocols</td><td>Asynchronous and synchronous byte-oriented (including IBM Bisync) plus SDLC and HDLC bit-oriented protocols.</td></tr>
<tr><td>Baud Rates</td><td>Support external clock rate to 600 kbps and sixteen strap-selectable rates:
<table>
<tr><th colspan="2">Strap-Selectable Rates</th></tr>
<tr><td>50</td><td>1800</td></tr>
<tr><td>75</td><td>2000</td></tr>
<tr><td>110</td><td>2400</td></tr>
<tr><td>134.5</td><td>3600</td></tr>
<tr><td>150</td><td>4800</td></tr>
<tr><td>300</td><td>7200</td></tr>
<tr><td>600</td><td>9600</td></tr>
<tr><td>1200</td><td>19200</td></tr>
</table></td></tr>
</table>

TABLE 1 — VM02 Specifications (continued)

Characteristics	Description
Programmable Timer/Counter (PTM)	Implemented using an MC6840 device which provides three 16-bit programmable binary counters. Each 16-bit section may be operated independently or sections may be cascaded to provide 32-bit or 48-bit operation under control of the MPU. Provides access to three lines from each of the 16-bit counter sections (a total of nine lines) at the I/O connector, P2: Gate Input Clock Input PTM Output Strap options allow five clock functions to be provided to the PTM: "AC Clock" (from VERSAbus) Baud rate clocks from the on-board serial port baud rate generators Address strobe from the local bus 2 MHz clock (derived from the on-board clock) Externally-provided time/count source from I/O connector, P2. Provide the PTM Interrupt output to the MPU on a user-selectable priority level. See MC6840 Fundamentals and Applications Manual (MC6840UM) for further details.
Interrupts	
VERSAbus Interrupts	Permits any or all of the seven interrupt request lines from the VERSAbus to be strapped to enable generation of an interrupt of corresponding priority level that is sent to the on-board MPU. In responding to a VERSAbus interrupt, the monoboard microcomputer must (a) request and gain bus mastership, (b) acknowledge the VERSAbus interrupt request, and (c) accept the interrupt vector from the interrupting device. In addition, the MPU responds to 12 "on-board" interrupts. When on-board and off-board interrupts of the same level occur at the same time, the on-board interrupt is serviced first.
Interrupter	Permits a VERSAbus interrupt to be generated at a software selectable level by writing to the status control register.
System Controller Functions	Activated only if the board is strap-selected as system controller.
VERSAbus Arbitration	Accepts priority level requests from potential bus masters on the five VERSAbus bus request lines. Issues a bus clear signal to the current bus master if a bus request at a higher priority level than that of current bus master is received. Issues a bus grant signal back to the highest priority requester when the bus is clear.
System Clock	Drives VERSAbus system clock line with 16 MHz signal.
Reset	Drives VERSAbus system reset line low upon occurrence of either of the following conditions: Manual activation of the RESET button on the top board edge Power-up

TABLE 1 — VM02 Specifications (continued)

Characteristics	Description
Power Down Provision	If the board is configured as system controller, it monitors the AC Fail line on the VERSAbus and, if the line is driven low by an external power failure detector, 1. generates and sends a non-maskable interrupt to the on-board MPU and, 2. generates and places a Bus Release signal on the VERSAbus.
Board Status/Control Registers	
Size	28 bits
Status Inputs 12 Bits	System Controller VERSAbus Available VERSAbus Interrupt Serviced System Failure VERSAbus Test User-Defined (6)
Control Outputs 16 Bits	VERSAbus Interrupt VERSAbus Interrupt Acknowledge Mask System Controller VERSAbus Transfer Request VERSAbus Block Transfer Request Board Fail Status Interrupt Mask VERSAbus Available Mask System Fail Interrupt Mask Write Protect I/O Channel Interrupt Mask VERSAbus Resource Management (4)
VERSAbus Interface Functions	The following subset of the complete VERSAbus function set is implemented. See the VERSAbus Specification Manual (M68KVBS).
Data	16 Lines
Address	23 Lines
Address Modifiers	8 Lines, providing the following functions: Short I/O Address Map Selection Interrupt Acknowledge Map Selection Supervisory Program Map Selection Supervisory Data Map Selection User Program Map Selection User Data Map Selection Special function user-defined maps

TABLE 1 — VM02 Specifications (continued)

Characteristics	Description
Data Transfer Control	Data Strobes (2) Write Line Address Strobe Data Transfer Acknowledge Bus Error
System Control	System Reset
Priority Interrupt Control	Interrupt Request (7 Priority Lines) Acknowledge In (Daisy Chain) Acknowledge Out (Daisy Chain)
Bus Arbitration Control	Bus Busy Bus Clear Bus Release (Emergency Bus Request) Bus Request (5 Priority Lines) Bus Grant In (Daisy Chain — 5 Lines) Bus Grant Out (Daisy Chain — 5 Lines)
System Test	System Fail, 2 Test Lines
Power Monitor	AC Failure
Misc. Functions	System Clock (16 MHz — square wave) AC CLock (50/60 Hz — square wave) receive only +5 Vdc +12 Vdc −12 Vdc
Operating Temperature	0° to 70°C
Humidity	0% to 95%, non-condensing
Physical Characteristics	
Height Width Thickness	9.25 in. (32.5 cm) 14.5 in. (36.8 cm) 0.6 in. (1.5 cm)
Bus Mating Connector Types	
VERSAbus Connector (P1)	Stanford Applied Eng'g CPH7000-140ST Micro Plastics, Inc. MP-0100-70-DW-5H
I/O Connector (P2)	Stanford Applied Eng'g CPH7000-120ST Micro Plastics, Inc. MP-0100-60-DW-5H
Power Requirements	**+5 Vdc** / **+12 Vdc** / **−12 Vdc**
Current Requirements Two MCM68764 with EPROMS	**Typ.** 5.5 A / **Max.** 6.4 A; **Typ.** .045 A / **Max.** .055 A; **Typ.** .035 A / **Max.** .045 A
Dynamic RAM Current Requirements	**Typ.** / **Max.**
Active Standby	430 mA / 500 mA 320 mA / 360 mA
Supply Voltages	+5 V ±5% +12 V ±5% −12 V ±5%

TABLE 2 — Signal Characteristics, I/O Connector P2

Connector Pins	Signal Mnemonic	Signal Name and Description	Signal Characteristic	
			Input	Output
33,35–44,46	A00–A11	I/O Channel address lines.	—	B
53	CLK	CLOCK — I/O Channel 4-MHz clock.	—	B
72	CLOCK1*	CLOCK 1 — May be used as an input clock for timer 1.	A	—
78	CLOCK2*	CLOCK 2 — May be used as an input clock for timer 2.	A	—
84	CLOCK3*	CLOCK 3 — May be used as an input clock for timer 3.	A	—
79 107	CTS1 CTS2	CLEAR TO SEND — Indicates that terminal may transmit data.	C** C**	C** C**
19,21–27	D0–D7	I/O channel data bits.		
85 113	DCD1 DCD2	DATA CARRIER DETECT — Indicates to terminal that a suitable data carrier is present.	C** C**	C** C**
81 109	DSR1 DSR2	DATA SET READY — Indicates that the data set (modem) is ready.	C** C**	
87 115	DTR1 DTR2	DATA TERMINAL READY — Indicates that data terminal is ready to transmit or receive data. The on-to-off transition will signal the modem to "hang up" the line.	C** C**	C** C**
74	GATE1*	GATE 1 — May be used to inhibit CLOCK1*.	A	—
80	GATE2*	GATE 2 — May be used to inhibit CLOCK2*.	A	—
86	GATE3*	GATE 3 — May be used to inhibit CLOCK3*.	A	—
1–6,18,20, 27,30,32, 24,45,48, 50,52,54, 56,58,60, 62,64,66, 71,83,99, 111,117	GND	GROUND	—	—
59	INT1*	INTERRUPT 1 — I/O channel interrupt 1.	B	
61	INT2*	INTERRUPT 2 — I/O channel interrupt 2.	B	
63	INT3*	INTERRUPT 3 — I/O channel interrupt 3.	B	
65	INT4*	INTERRUPT 4 — I/O channel interrupt 4.	B	

*A, B, C, and D characteristics defined in Table 3.
**Signal characteristics may vary according to whether this port is defined as terminal or modem.

TABLE 2 — Signal Characteristics, I/O Connector P2 (continued)

Connector Pins	Signal Mnemonic	Signal Name and Description	Signal Characteristic	
			Input	Output
76	OUTPUT1	TIMER 1 OUTPUT — The Timer 1 output may be selected to appear on this pin.	—	B
82	OUTPUT2	TIMER 2 OUTPUT — The Timer 2 output may be selected to appear on this pin.	—	B
88	OUTPUT3	TIMER 3 OUTPUT — The Timer 3 output may be selected to appear on this pin.	—	B
57	RESET*	RESET — I/O channel reset (output) signal.	—	D
77 105	RTS1 RTS2	REQUEST TO SEND — Indicates that terminal wishes to send data. On a half duplex channel, this signal controls direction of data transmission.		C** C**
89 117	RXC1 RXC2	RECEIVE CLOCK — May be an output from baud rate generator or an input clock for transmitter or receiver.	C** C**	C** C**
75 103	RXD1 RXD2	RECEIVE DATA — Used to receive data as an input, or transmit data as an output.	C** C**	C** C**
31	STB*	STROBE — I/O channel output signal.		
91 119	TXC1 TXC2	TRANSMIT CLOCK — May be an output from baud rate generator or an input clock for transmitter or receiver.	C** C**	C** C**
73 101	TXD1 TXD2	TRANSMIT DATA — Used to transmit data as an output, or receive data as an input.	C** C**	C** C**
29	WT*	WRITE — I/O channel output signal.	—	B
55	XACK*	TRANSFER ACKNOWLEDGE — I/O channel data transfer acknowledge — input signal.	B	—
7–10	+5 V	+5 Vdc Power — Used by VM02 logic circuits.	—	—
93	+ 5 VOUTB	+5 Vdc Power — Jumper selectable for I/O.	—	—
15,16	−12 V	−12 Vdc Power — Used by VM02 logic and interface circuits.	—	—
95	−12 VOUTB	−12 Vdc Power — Jumper selectable for I/O.	—	—
11,12	+12 V	+12 Vdc Power — Used by VM02 logic and interface circuits.	—	—
97	+12 VOUTB	+12 Vdc Power — Jumper selectable for I/O.	—	—
67,68	−15 V	−15 Vdc Power — Not Used.	—	—
17,45, 69,70	+15 V	+15 Vdc Power – Not Used.	—	—

TABLE 3 — Signal Category Definitions

Signal Type "A"					
Input	**Min.**	**Max.**	**Output**	**Min.**	**Max.**
Allowed Input Voltage	0 V	7.0 V	Guaranteed High Voltage When Sourcing 14.6 mA	2.0	—
Allowed Input for "High"	2.0 V	—	Guaranteed Low Voltage When Sinking 23.8 mA	—	0.5
"Low"	—	0.8 V			
Current Sinked When Driven	—	40 μA			
"High"	—	220 μA			
"Low"					

Signal Type "B"					
Input	**Min.**	**Max.**	**Output**	**Min.**	**Max.**
Allowed Input Voltage	0 V	7.0 V	Guaranteed High Voltage When Sourcing 15 mA	2.0	—
Allowed Input for "High"	2.0 V	—	Guaranteed Low Voltage When Sinking 24 mA	—	0.5 V
"Low"	—	0.8 V			
Current Sinked When Driven	—	20 μA			
"High"	—	690 μA			
"Low"					

Signal Type "C" (RS-232C Levels)					
Input	**Min.**	**Max.**	**Output**	**Min.**	**Max.**
Allowed Input Voltage	−30 V	30 V	Guaranteed High Voltage (Space) Across 3K Load	7.0	—
Allowed Input for "Space"	3.0 V	—	Low Voltage (Mark)	−7.0	—
"Mark"	−3.0 V	—			
Current Sinked When "Space" (V_{on} = 25 V)	—	8.3 mA			
Current Sinked When "Mark" (V_{off} = −25 V)	—	−8.3 mA			

Signal Type "D" (Open Collector Output)					
Input	**Min.**	**Max.**	**Output**	**Min.**	**Max.**
Allowed Input for "High"	2.0 V	—	Guaranteed Low Voltage When Sinking 24 mA	—	0.5
"Low"	—	0.8 V			
Current Sinked When Driven	—	20 μA			
"High"	—	690 μA			
"Low"					

VERSAdos — Real-Time Disk Operating System

- Provides all Software Features of the RMS68K Kernel
- Device Independent I/O and Logical I/O
- Wait and Proceed Mode I/O
- Standard Device Drivers
- Multi-Level File Directories
- Shared File Access
- Dynamic or Contiguous File Space Allocation
- Ramdon, Sequential, and Indexed Sequential File Access

VERSAbug — Debug/Monitor/Loader Firmware

- Initialization
- Display/Change Memory
- Display/Change Registers
- Set and Clear Breakpoints
- Block Initialize
- Block Move
- TRACE with optional instruction count
- Downline Load
- Single Line Assembler/Disassembler

HARDWARE/SOFTWARE DEVELOPMENT SUPPORT ıment Support

The recommended development system vehicle for developing microcomputer systems based on the VERSAmodule Monoboard Microcomputer is the EXORmacs MC68000 Development System. EXORmacs is a multiuser development system (floppy disk or hard disk based) with advanced software development tools including a macro assembler, pascal compiler, CRT editor, and linkage editor. Since EXORmacs also provides the VERSAbus interconnect structure and the VERSAdos operating systems, VERSAmodule applications can be easily developed and checked out in the EXORmacs chassis and transferred to the target for final debug.

Cross software support in the form of a macro assembler, pascal compiler, and linkage editor is also available for IBM host computers.

SYSTEM EXPANSION

The VERSAmodule Monoboard Microcomputer is system-compatible with a growing family of VERSAmodule products:

Dynamic RAM Modules — available 128K, 256K and 512K byte versions. Includes byte parity with automatic retry on parity error.

Universal Disk Controller (UDC) — A 2-board set that provides industry standard SMD interface to hard disk drives (up to two drives of 96 megabytes each) and floppy disk drives (up to four drives of 0.5 megabytes each).

Floppy Disk Controller (FDC) — A single board that provides an interface up to four double-sided single-density floppy disk drives of 0.5 megabyte each.

Multichannel Communications Module (MCCM) — Provides four asynchronous serial ports, each with RS-232C interface, plus an industry-standard parallel printer interface port.

Universal Intelligent Peripheral Controller (UIPC) — Provides IPC architecture on a single board with a DMA channel to global memory on the VERSAbus. A parallel interface is provided to which a user may interface a special device such as tape, disk or high-speed communications controllers.

Color Graphic Processor — In conjunction with a high-resolution color monitor, the Color Graphic Processor adds full color graphic display capabilities to any VERSAbus compatible system. Provides a powerful set of graphic instructions to ease development of graphics application software. Contains standard VERSAmodule I/O Channel interface for connection to off-board graphics peripherals.

The UDC, FDC, MCCM, UIPC and Color Graphic Processor modules each provide a consistent electrical and logical interface to the VERSAbus, and to VERSAdos or RMS68K system software. This allows simplified device-independent I/O for the main application. In addition, the I/O Channel interface provides access to a growing family of I/Omodules which are system compatible with VM02.

Remote Intelligent Analog-To-Digital Conversion Module (RAD1) — Provides 32 single-ended or 16 differential A/D channels with a choice of parallel or serial I/O operation.

Four-Slot Card Cage — Mechanically and electrically expandable to three units (12 Slots).

Four-Slot Chassis — Provides card cage, power supply, fans, enclosure, and rack-mount or tabletop usage capability.

Power Supplies — Low-Power (15 A @ 5 Vdc) and high-power (30 A @ 5 Vdc) versions are available. Each includes an ac power failure detection capability with VERSAbus interface.

VERSAbus Adapter Module — A module for interfacing 8-bit EXORbus boards to 16-bit VERSAbus systems.

VERSAbus Extender Module — Extends VERSAbus modules for serving, testing, troubleshooting, and debugging.

VERSAbus Wirewrap Module — Facilitates construction and incorporation of custom circuits into VERSAbus systems.

Packaging and Accessories

VERSAmodule-based applications can be conveniently packaged in and supported by a family of hardware accessories. (For additional information on packaging options and power supplies, refer to the chassis card cage data sheet MVMCH3-1.)

Four-Slot Card Cage — Mechanically and electrically expandable to three units (12 Slots).

Four-Slot Chassis — Provides card cage, power supply, fans, enclosure, and rack-mount or tabletop usage capability. I/O Channel card cage optional.

Power Supplies — Low-Power (15 A @ 5 Vdc) and high-power (30 A @ 5 Vdc) versions are available. Each includes an ac power failure detection capability with VERSAbus interface.

VERSAbus Adapter Module — A module for interfacing 8-bit EXORbus boards to 16-bit VERSAbus systems.

VERSAbus Extender Module — Extends VERSAbus modules for serving, testing, troubleshooting, and debugging.

VERSAbus Wirewrap Module — Facilitates construction and incorporation of custom circuits into VERSAbus systems.

I/O Channel Components

Winchester Disk Controller I/Omodules — Interface an I/O channel host to 5¼" or 8" Winchester and floppy disk drive combinations.

Floppy Disk Controller I/Omodule — Interfaces an I/O channel host to 5¼" or 8" floppy disk drives.

Analog Input and Output Modules — Provides a complete multichannel, 12-bit, data acquisition system (input module). Provides four independent, 12-bit analog signals (output module).

Opto Isolated 120 V/240 V Input and Output Modules — Provide interface with line voltage ac for control of motors, relays, contactors, and signal lamps.

Opto Isolated 30 Vdc Input and Output Modules — Provide interface with dc operated devices such as motors, relays, lamps, etc.

Remote Input/Output Module (RIO1) — Provides mounting for up to 16 solid-state-relay input or output modules.

Dual Channel RS-232C Serial Port — Provides two independent, full duplex, multiprotocol serial communication input/output ports with RS-232C interfaces.

Dual Channel 16-Bit Parallel Port — Provides four 8-bit data ports with two handshake lines per port that are controlled by a microcomputer I/O Channel interface.

SASI™ Peripheral Adapter — A single high Eurocard module providing interface between a microcomputer I/O Channel interface and a Shugart Associates SASI bus for use of an SA1400 hard disk controller.

Buffered 9-Track Magnetic Tape Adapter — Provides an I/O Channel interface and an interface for up to two daisy chained industry standard 9-Track, ½" dual density magnetic tape formatters capable of handling up to four drives each. The adapter includes a 4K byte FIFO buffer.

Ordering Information

Part Number	Description
M68KVM02-3	VERSAmodule Monoboard Microcomputer. This module contains the MC68000 MPU, 128K bytes RAM, sockets for up to 64 bytes of ROM/EPROM, two multiprotocol serial ports, a triple programmable timer/counter, VERSAbus interface, system controller features and I/O Channel interface. Includes User's Manual, and an I/O Channel Specification Manual.
M68KVM10-3	128K Byte VERSAbus Dynamic RAM Module
M68KVM11-1	256K Byte VERSAbus Dynamic RAM Module with ECC
M68KVM11-2	512K Byte VERSAbus Dynamic RAM Module with ECC
M68KVM21	Universal Disk Controller (UDC)
M68KVM20	Floppy Disk Controller (FDC)
M68KVM30	Multi-Channel Communications Module (MCCM)
M68KVM60	Universal Intelligent Peripheral Controller (UIPC)
M68K0RMS68K	Real-Time Multitasking Software
M68K0VDOS	Real-Time Disk Operating System
M68KVBUG2	Debug/Monitor/Loader Firmware
M68KVMCC1	4-Slot VERSAmodule Card Cage
M68KVMCH1-1	4-Slot VERSAmodule Chassis with 123 Watt Linear Power Supply
M68KVMCH1-2	4-Slot VERSAmodule Chassis with 228 Watt Switching Power Supply
MVMCH1-2	Same as M68KVMCH1-2 with 5-Slot Card Cage for Single Width Eurocard Format I/O Cards

Ordering Information (continued)

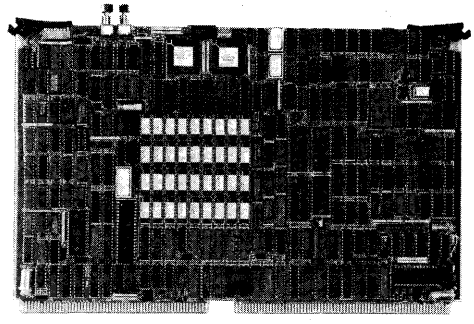
Part Number	Description
M68KVAM	VERSAbus Adapter Module
M68KEXT	VERSAbus Extender Module
M68KWW	VERSAbus Wirewrap Module
MVME400	Dual Channel RS-232C Serial Port
MVME410	Dual Channel 16-bit Parallel Port
MVME420	SASI™ Peripheral Adapter
MVME435	Buffered 9-Track Magnetic Tape Adapter
M68RWIN1-1,2	Winchester Disk Controller I/Omodule
M68RFDC1	Floppy Disk Controller I/Omodule
MVME600,605	VMEmodule Analog Input and Output
MVME610,615,616	VMEmodule Opto Isolated 120 V/240 V Input and Output
MME620,625	VMEmodule Opto Isolated 30 Vdc Input and Output

Related Documentation

M68RIOCS	I/O Channel Specification Manual
M68KVBS	VERSAbus Specification Manual
MC68000UM (AD-3)	MC68000 Microprocessor User's Manual
MC6840UM	MC6840 Fundamentals and Applications Manual

M68KVM03-1
M68KVM03-3
M68KVM03-4
M68KVM03-5

VERSAmodule Monoboard Microcomputer



The VM03 microcomputer is a high performance VERSAmodule designed for use in a wide variety of VERSAbus systems ranging from those based on a single monoboard through complex multiprocessor systems. Figure 1 is a functional block diagram of the module. The VM03 has the following features:

- MC68010 Microprocessor Running at 10 MHz.
- MC68451 Memory Management Unit (MMU).
- 256K Bytes (M68KVM03-1) or 1024K bytes (M68KVM03-4) of On-Board Dynamic RAM with Parity Check using 64K or 256K by 1 DRAM Devices.
- Dual Port Controller Provides True Shared Access to RAM without Danger of Lock-up from the Local Bus and VERSAbus.
- A Jumper Option Allows Powering the RAM from a Standby Supply During Power Fail.
- Hardware Independent Refresh Circuitry.
- VERSAbus Arbiter.
- VERSAbus Interrupt Handler.
- VERSAbus Interrupter.
- Two 28-pin Sockets for up to 64K Bytes of User Provided 8K by 8, 16K by 8, or 32K by 8 ROM/PROM/EPROM Devices Operating at 350 Nanoseconds.
- Two Multiprotocol Serial Input/Output Ports with RS-232C Interface.
- MC6840 Programmable Timer Module.
- MC146818 Real Time Clock.
- I/O Channel Interface for Adding Off-Board Resources.
- VERSAbus Compatible.
- 0°C–70°C Operating Temperature Range.
- Full Operating System Support:
  - — The SYSTEM V/68 Operating System — Derived from UNIX System V, M68000 Version.
  - — The VERSAdos Real-Time Multitasking Operating System.

CPU

The VM03 uses a MC68010 Microprocessor operating at a fixed speed of 10 MHz. Both the CPU and system clocks are derived from clock oscillators operating at twice the desired clock rate. The system clock oscillator runs at 32 MHz and is divided to produce the 16 MHz VERSAbus clock plus 8, 4, and 2 MHz general purpose clocks. The CPU clock oscillator operates at 20 MHz and is divided to produce the CPUCLK, CPUCLK/2, CPUCLK/4, and CPUCLK/8 outputs.

Resets to the VM03 may originate from power-up, VERSAbus reset, CPU reset or a pushbutton switch on the card edge. Halt and bus error logic asserts the HALT* and BERR* inputs to the CPU during reset and bus error cycles. This logic also facilitates bus retrys on memory errors when using the MC68010 MPU.

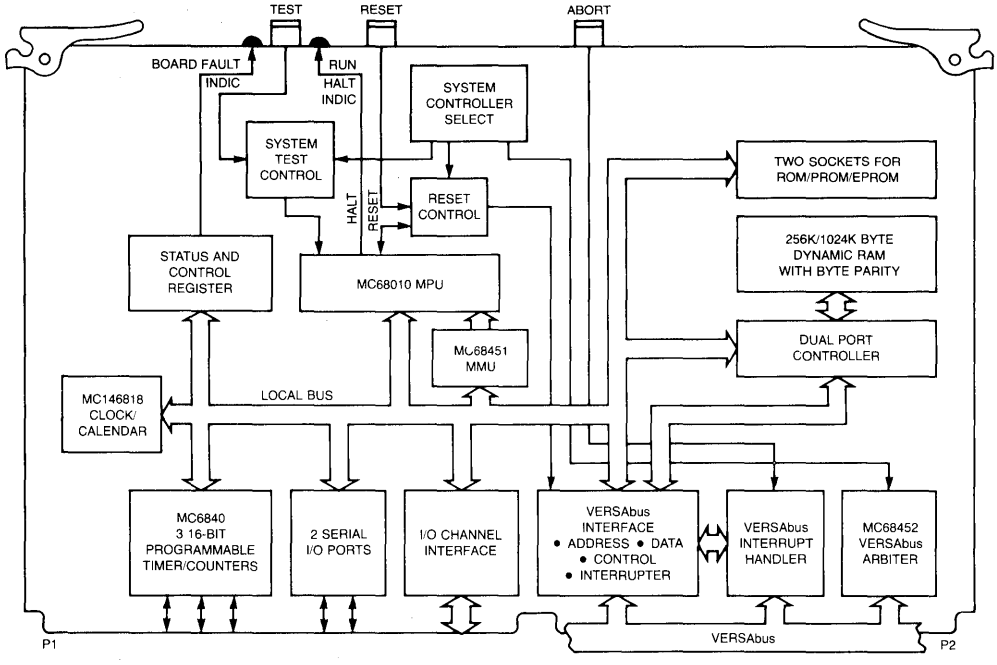
Bus timeout can reach a delay of 64 microseconds before a CPU bus error is generated. Separate delays are provided for local bus and VERSAbus protection. The user may disable the VERSAbus timeout, via a jumper, to use the VM03 in applications where it is not the System Controller.

A pushbutton abort switch generates a high level interrupt. Via a jumper, the user can select the level or disable the switch.

MEMORY MAP

The MC68010 can physically address 16M bytes of memory. Addresses are mapped directly to the physical address space by the MPU or by the MC68451, when the MMU option is enabled. Physical address definition is provided by an address decode PLA.

FIGURE 1 — M68KVM03 Functional Block Diagram



Immediately following a total system reset, the first 8 bytes of ROM/PROM/EPROM are mapped to address 000000 and should, therefore, contain the system reset program counter and stack pointer values.

The complete VM03 memory map is shown in Table 1.

BASE ADDRESS

A header on VM03 permits jumper selection of a base address for the module on any 256K byte boundary throughout the 16 megabyte VERSAbus physical address space. Decoding circuitry supports contiguous mapping of local and global RAM.

DUAL PORT RAM

The VM03-1 has 36–64K dynamic RAMs, providing 256K bytes of parity protected dynamic RAM. On-board memory is increased to 1024K bytes in the VM03-4 which has 36–256K dynamic RAMs.

The physical address of the on-board RAM starts at 000000 and extends to 3FFFFF (0FFFFF). Optionally, the user may divide the on-board memory into four segments and dynamically allocate from one to four of these segments as private RAM. If an attempt is then made to access these segments from the VERSAbus, a bus error will occur. This feature is operable whether or not the MC68451 is available, see Table 2.

A byte parity check is generated and stored on a DRAM write operation. A read operation calculates byte parity and compares it to the parity bit stored in memory. If stored and calculated parity bits differ, a bus error is generated. This signal becomes active after DTACK.

Timing characteristics for the on-board RAM and the VM03-to-VERSAbus access are provided in Table 3.

TABLE 1 — Address Map

Physical Address	Comments
000000–3FFFFF(0FFFFF)	On-board RAM 256K/1024K bytes
400000(100000)–EFFFFF	VERSAbus
F00000–FGFFFF	EPROM (for 64K byte option)
F10000–F7FFFF	VERSAbus
F80000–F8003F	MMU
F80040–F8004F	PTM
F80060–F8007F	Serial I/O Port
F80080–F800FF	RTC
F80100–F80107	Control/Status Registers
F80109	VERSAbus Interrupt Vector Register
F82000–F9FFFF	VERSAbus
FA0000–FA1FFF	I/O Channel
FA2000–FEFFFF	VERSAbus Request
FF0000–FFFFFF	VERSAbus Short I/O

TABLE 2 — VM03 Memory Control Register

07	06	05	04	03	02	01	00

00	<Parity Enable>	05–06	<Local RAM Select>
01	<Write Wrong Parity>	07	<Local RAM Enable>
02–04	<Not Used>		

TABLE 3 — VM03 Performance (Wait cycles at 10 MHz)

Condition		w/MMU	wo/MMU
Local	read	3	1
	write	4	2
Remote w/arbitration	read	9	8
	write	6	5
Remote wo/arbitration	read	8	7
	write	5	4
VERSAbus access to local RAM	read	550 ns nominal	
	write	600 ns nominal	

PROGRAMMABLE TIMER MODULE

The module has a MC6840 Programmable Timer which provides three 16-bit counters whose outputs may be connected to counter inputs to allow cascading.

REAL-TIME CLOCK

The VM03 contains a MC146818 Real-Time Clock with an option to use a standby power supply during power fail. The jumper to the standby power supply is the same as that used for RAM.

SERIAL INPUT/OUTPUT

The two independent serial ports on VM03 are user-configurable. With Port A configured as a terminal and Port B as a modem the module supports asynchronous communications. Internally generated baud rates are 300, 600, 1200, 2400, 4800, 9600, and 19,200 baud. Externally generated baud rates are also supported. The hardware also supports implementation of synchronous byte and bit oriented (SDLC, ADLC, HDLC) protocols.

The headers for implementing jumper options for the serial interfaces are provided on transition boards. The user may

configure the ports to function with terminals or modems. When shipped from the factory, both interfaces are defined as terminals.

I/O CHANNEL

The Motorola I/O Channel is specifically designed to provide efficient, low-cost distributed communications between CPU and peripheral and I/O controller boards. It provides a 12-bit address bus, an 8-bit bidirectional data bus and supports asynchronous operation at data rates up to 2 megabytes/sec. For those modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable up to 12 feet in length. Available I/O Channel modules include:

Single High Eurocard Form Factor Modules

- MVME400 Dual Channel RS-232C Serial Port (Synchronous/Asynchronous)
- MVME410 Dual Channel 16-bit Parallel Port (Centronics compatible)
- MVME420 SASI Peripheral Adapter
- MVME600, 605 Analog Input and Output
- MVME610, 615, 616 VMEmodule Opto Isolated 120V/240V Input and Output
- MVME620, 625 Opto Isolated 30 Vdc Input and Output
- M68RWIN1 Winchester Disk Controller Module

Non-Eurocard Modules

- M68RAD1 Remote Intelligent Analog-to-Digital Conversion Module
- M68RI01 Remote Input/Output Module
- M68RSC1, -2 Remote Serial Conversion Module

INTERRUPT HANDLER

The VM03 allows interrupts to the on-board MPU from up to 21 sources. Interrupt handler circuitry pre-assigns an interrupt to one of three groups labeled 0, 1, or 2. Within a group, the interrupt is further assigned to a level corresponding to one of the seven MC68010 interrupt levels. Service priority is thus determined both by group and level, as shown in Table 4. The module has a header in which any of the 21 interrupt sources can be jumper disabled.

Group 0 levels are reserved for VERSAbus interrupts. Group 1 and group 2 levels are interchangeably assignable to interrupts generated by the I/O Channel or the on-board devices.

Group 2 and 1 interrupts are processed differently from group 0 interrupts. If the interrupt being acknowledged is a group 2 or 1 interrupt, the handler fetches the appropriate exception vector number from PROM and sends it to the CPU via the local bus. If the interrupt being acknowledged is a group 0 interrupt, the exception vector number is fetched from the VERSAbus where it was placed by the interrupting device. Interrupt assignments are shown in Table 4.

VERSAbus INTERRUPTER

The VERSAbus interrupter can, under software control, generate any level of VERSAbus interrupt. When the VERSAbus interrupt is acknowledged, the interrupter fetches the appropriate exception vector number from a hardware register and places it on the VERSAbus.

STATUS/CONTROL REGISTERS

The status and control Registers provide local and global status information to the on-board MPU. By reading the status and control register bits, the MPU can determine VERSAbus information, interrupt status, the source of a particular BERR, and which self test to perform. The status register bits and the programmable control register bits control the MPU board's hardware by providing configuration and control information to the VERSAbus interrupter, the interrupt handler, and the VERSAbus arbiter. The status/control registers are privileged resources and in special cases can be made accessible by the user. The status and control Register formats are shown in Table 5.

TABLE 4 — VM03 Interrupt Handling

MPU IRQ Level	First > Last		
	Group 2	Group 1	Group 0
7	Abort Pushbutton	ACFAIL/BREL	IRQ7
6	I/O Channel INT4	PTM	IRQ6
5	I/O Channel INT3	RTC	IRQ5
4	I/O Channel INT2	Serial Ports	IRQ4
3	I/O Channel INT1	BUSAV	IRQ3
2	Bus Clear	VBIACK	IRQ2
1	MMU	SYSFAIL	IRQ1

TABLE 5 — VM03 Status/Control Register Formats

Control Register — Lower Byte

07	06	05	04	03	02	01	00

00–02 <VERSAbus Interrupt Level>
03 <VERSAbus Interrupt Acknowledge Enable>
04 <System Controller Transfer Request>
05 <Block Transfer Request>
06 <Board Fail>
07 <Interrupt Enable>

Control Register — Upper Byte

15	14	13	12	11	10	09	08

08 <VERSAbus Available Interrupt Enable>
09 <System Fail Interrupt Enable>
10 <Reserved>
11 <I/O Channel Interrupt Enable>
12–15 <Address Modifiers A through D>

Status Register — Lower Byte

07	06	05	04	03	02	01	00

00 <SCON Enable>
01 <VERSAbus Available>
02 <VERSAbus IRQ>
03 <SYSFAIL>
04 <TEST0>
05 <TEST1>
06 <Maintenance Mode>
07 <Loopback Status>

Status Register — Upper Byte

15	14	13	12	11	10	09	08

08 <Local Bus Timeout>
09 <VERSAbus BERR>
10 <MMU BERR>
11 <Memory BERR>
12–13 <MMU Configuration>
14–15 <User Configured>

VERSAbus INTERFACE

The VERSAbus interface supports VERSAbus arbitration, buffering of data, address, and control signals, and interrupt handling.

The VERSAbus arbiter handles up to 5 levels of bus requests on a priority basis. It utilizes the MC68452 BAM chip and also offers the Power Fail (PF) option described in the VERSAbus specification.

OPERATING SYSTEMS SUPPORT

The VM03 will be supported by SYSTEM V/68 on release and is fully supported by VERSAdos release 4.3.

DEBUG SUPPORT

The VERSAbug 3.0 Debugging Package provides for VERSAmodule 3 a powerful evaluation, use and system debugging tool. It permits full speed execution of system and user-developed programs in a VM03 system environment under complete operator control. Forty four debug, up/down-line load and disk bootstrap load commands are provided.

TABLE 6 — M68KVM03 Specifications

Characteristics	Specification
Microprocessor	MC68010 (refer to Motorola Publication ADI-942)
Power Requirements	+5 Vdc, 6.4 A max. (5.5 A typical) +12 Vdc, 55 mA max. (45 mA typical) −12 Vdc, 45 mA max. (35 mA typical)
On-board RAM Power Requirements Standby RAM (refresh only) Active RAM	 700 mA max. (550 mA typical) 1200 mA max. (750 mA typical)
Clock Signal	8 MHz or 10 MHz on-board input signals to the MPU. The system clock signal (16 MHz, SYSCLK) is available for either option when the VM03 is the system controller.
Addressing Total System Size (on- and off-board) ROM/PROM/EPROM Dynamic RAM	 16 megabytes Two 28-pin sockets for 4K-, 8K-, 16K-, or 32K-byte devices using +5 Vdc only (up to 128K bytes). 256K bytes (for VM03-1, -3) 1024K bytes (for VM03-4, -5)
I/O Ports Serial Parallel	 Two multiprotocol serial communications channels (uses μPD7201) with RS-232C interface. Parallel I/O Channel supports 4K byte memory mapped I/O and four IRQ lines.
Timer	One triple 16-bit programmable timer (MC68B40).
Interrupts	Any one of seven levels (IRQ1–IRQ7) can be generated by the on-board MPU or received by the board from the VERSAbus, or both. Fourteen additional interrupts are received from local sources and the I/O Channel.
Bus Arbitration	Up to 5 levels of bus request (MC68452 BAM) When the VM03 is the System Controller, it arbitrates all system requests for bus mastery.
Reset	RESET pushbutton or VERSAbus SYSRESET* line resets the MPU. HALT LED indicates when the MPU has halted as a result of a bus fault.
Test	If the VM03 is the System Controller, it can generate all four VERSAbus test modes. VERSAbus test lines are software readable.
Operating Temperature	0° to 70°C
Storage Temperature	−40° to 85°C
Relative Humidity	5% to 95% (non-condensing)
Physical Dimensions Height Width Thickness	 9.25 in (23.50 cm) 14.50 in (36.83 cm) 0.60 in (1.52 cm)
Board Mating Connectors P1 (140-pin) P2 (120-pin)	 Stanford Applied Engineering CHP7000-140ST or Micro Plastics, Inc. MP-0100-70-DW-5H. Stanford Applied Engineering CHP7000-120ST or Micro Plastics, Inc. MP-0100-60-DW-5H.

Ordering Information

Part Number	Description
M68KVM03-1	VERSAmodule Monoboard Microcomputer with MC68010 MPU and MC68451 MMU plus 256K Bytes of Dynamic RAM. Includes User's Manual.
M68KVM03-4	VERSAmodule Monoboard Microcomputer with MC68010 MPU and MC68451 MMU plus 1024K Bytes of Dynamic RAM.
M68KVM03-3	MC68010, No MMU, 256K
M68KVM03-5	MC68010, No MMU, 1024K

Related Product

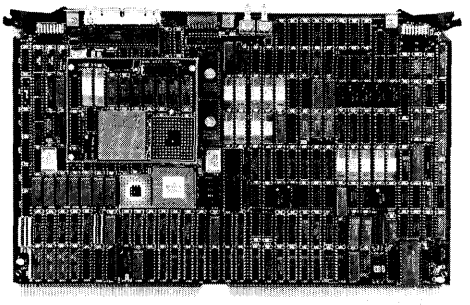
M68KVBUG3	VERSAbug 3.0 Debugging Package for the VERSAmodule 3 Monoboard Microcomputer

Related Documentation

M68KVMO3/D1	VERSAmodule Monoboard Microcomputer User's Manual.
M68KVBS/D4	VERSabus Specification Manual

M68KVM04-1

VERSAmodule 32-Bit Monoboard Microcomputer



The VERSAmodule 32-bit Monoboard Microcomputer (VM04) is designed to function in those applications requiring maximum performance while maintaining the versatility inherent with VERSAmodule systems. Highest performance is attained when the VM04 is used in conjunction with one or more M68KVM13 (VM13) Dual Ported RAM Cards operating as a main memory. The VM04 is the first VERSAmodule product to offer the following:

- MC68020 Microprocessor with 32-bit address and data.
- Provision for MC68881 Floating Point Coprocessor (customer-supplied option).
- Demand Paged Virtual Memory Management Module implemented with gate array technology (MMB).
- An on-board software transparent cache configured as 4K entries, with each entry supporting full 32-bit address/data.
- RAMbus Interface provides high-speed data path to/from memory.
- VERSAbus Interface allows user configuration of Microcomputer System to fit the application.
- VERSAbus Interrupter, Interrupt Handler, and Arbiter onboard.
- Programmable Timer Module.
- Dual Multiprotocol Serial I/O Ports.
- Two ROM sockets configured for Industry Standard 28-pin ROM/EPROM devices.
- Accepts 020bug DeBug Monitor Firmware (optionally available).
- 0°C–70°C Operating Temperature Range.

CPU

The VM04 uses an MC68020 Microprocessor operating at a fixed speed of 16.67 MHz. The MC68020 is the first product within the popular MC68000 family to offer external 32-bit address and data paths. With its higher clock rate, advanced architecture, enhanced addressing modes, and on-chip instruction cache, this product offers state-of-the-art performance while maintaining software compatibility with its widely accepted predecessors.

DEBUG MONITOR FIRMWARE

The 020bug Debug Monitor firmware package is optionally available for use with the VM04 board. This firmware offers 32 debug, up/downline load, one line assembler/disassembler and disk bootstrap load commands. Refer to its data sheet for complete specifications.

COPROCESSOR

The VM04 board is equipped with a socket to accept the MC68881 Floating Point Coprocessor, a customer supplied option. When available and installed, the MC68881 Floating Point Coprocessor will improve computing speed of the VM04 when the system is utilized in applications requiring arithmetic operations.

The VM04 utilizes the MC68020 Coprocessor Interface to provide an execution model as a transparent logical extension to the MC68020 architecture and instruction set.

Figure 1 is a block diagram of the M68KVM04 Module.

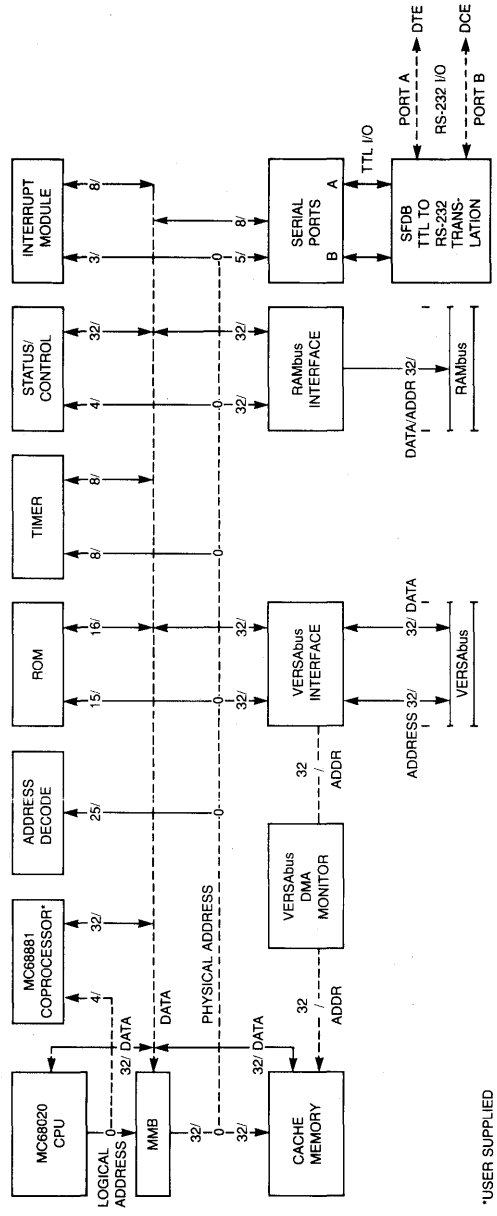


FIGURE 1 — M68KVM04 Block Diagram

MEMORY MANAGEMENT

The VM04 is designed to eventually utilize the MC68851 Demand Paged Virtual Memory Management Unit. However, the initial versions of the VM04 utilize a Memory Management Board (MMB) which conforms to a subset of the MC68851. This subset is implemented via gate array technology together with other semiconductor devices mounted on a secondary PC board. The socket assigned for future use with the MC68851 is utilized to mount the secondary PC board in a mezzanine fashion on the introductory version of the VM04. (Note: The possibility exists that the pin-signal configuration utilized on the VM04 will not match the MC68851.) The functionality of the MMB is as follows:

- Logical address consisting of:
  - — 32-bit Address
  - — 3-bit Function Code
- Physical address output of 32 bits.
- Logical to physical address translation;
  - — Table walking algorithm
  - — Single logical bus master
  - — Single page size (1K byte)
- Used, modified, and write protect bits.
- Supports MC68020 and external data cache.
- 512 entry set associative cache.

The eventual use of the MC68851 will add multiple Protection/Privilege capabilities.

The principal operating function of the MMB on the VM04 is to provide the hardware logical to physical address translation and access protection necessary for operating system support. The MMB logical to physical translation mapping function utilizes a table-walking algorithmic search through memory-resident translation tables. The MMB utilizes a cache architecture to eliminate table-walking for most CPU bus cycles.

The MMB utilizes the MC68020 clock circuitry, but does not use the coprocessor interface. This requires the internal registers to be in the CPU address space, whereas the MC68851 internal registers will be accessed via the coprocessor interface. This will require a change in driver routines when the MC68851 version of the VM04 is designed to be a true subset of the MC68851.

ON-BOARD CACHE

The cache on the VM04 is designed to complement the on-chip instruction cache of the MC68020. It is located on the physical side of the Memory Management Circuitry, and consists of 4K entries by 32 bits. The cache is capable of operating with Byte, Word, and Long-Word transfers and functions with both instruction and data oriented operations.

A write operation by the CPU to cacheable memory causes data to be written into the cache. (In normal operation, with cache enabled, cacheable memory includes all VERSAbus Standard and Extended addresses as well as all RAMbus addresses).

A write operation by a secondary master on the VERSAbus to cacheable memory will be detected by monitor circuitry on the VM04. An access to the cache Tag Field will be generated. If a match exists (indicates that an overwrite of cached data may have occurred), the appropriate Invalid Data Bit in the Tag Field is set.

These hardware provisions, together with the physical address location of the on-board cache, make the cache software transparent in most applications. A variety of control bits in the VM04 Control/Status Register set allow software intervention if desired. These include:

- Clear Cache.
- Enable/disable Cache Operation.
- Mask Cache by Data Type.
  - — Supervisor data access
  - — Supervisor program access
  - — User data access
  - — User program access
- VERSAbus Memory Access not Cached.
- Cache Test Mode.

RAMbus INTERFACE

The RAMbus Interface occupies 50 P2 I/O pins and is designed to serve as the system's high speed memory access channel. The signals utilize multiplexing of address and data in order to accommodate full 32-bit functionality, along with appropriate control signals, into the 50-pin allotment.

The RAMbus Interface is limited to no more than one primary and one secondary master. This eliminates much of the arbitration overhead required by the VERSAbus Interface. The VM04, as primary master with one or more VM13's creates a memory system with performance similar to that of the on-board memory of the VM03, but with virtually unlimited size constraints.

MEMORY MAP

The Address map, see Table 1, indicates the 32-bit address map for different devices addressed by the VM04. The 24-bit address map is determined by eliminating the two most significant hex digits from the 32-bit addresses.

TABLE 1 — Address Map

Device	Physical Address	Comments
VERSAbus/RAMbus	00000000-FFF7FFFF	32/16-bit port size; R/W.
ROM	FFF80000-FFF8FFFF	16-bit port size; read only.
Cache Data RAM	FFF90000-FFF97FFF	32-bit port size; R/W; Any write to this space will invalidate the corresponding byte valid bits.
Cache Byte Valid RAM	FFF98000-FFF9FFFF	32-bit port size; D03-D00 contain the byte valid information; read only.
Spare	FFFA0000-FFFAFFFF	Unimplemented; Access will cause LBTO (BERR source) cycle termination.
Timer	FFFBxx00-FFFBxx2F	8-bit port size; R/W.
Status & Control	FFFBxx30-FFFBxx3F	32-bit port size; Status is read only; Control is R/W.
Serial I/O Port	FFFBxx40-FFFBxx5F	8-bit port size; most are R/W.
Reserved	FFFBxx60-FFFBxx6F	Unimplemented.
Spare	FFFBxx70-FFFEFFFF	Unimplemented; references to this space will cause the cycle to terminate with a BERR trap.
VERSAbus Short I/O	FFFF0000-FFFFFFFF	16-bit; R/W; VERSAbus Short I/O Address Modifier will be generated.

PROGRAMMABLE TIMER MODULE

The VM04 Programmer Timer Module provides the following:

- Three Cascadable, Independent 16-bit Counters Internally
- Interrupt Capability

SERIAL I/O

The VM04 is equipped with two multiprotocol serial I/O ports with connection via a 34-pin flat ribbon cable connector at the top of the board. Signal levels at this connector correspond to TTL specifications. A transition board/cable/connector will be made available to transform the TTL levels to RS-232C and provide a standard DB-25 interface. The ports have the following features:

- Programmable Baud Rates
- Full and Half Duplex Compatibility
- 5- to 8-bits Per Character Plus Parity
- Synchronous or Asynchronous Operation, Including
  - — Byte oriented protocols (BISYNCH)
  - — Bit oriented protocols (SDLC)
- Data Rate Capability to 800 Kilobit/Second

STATUS/CONTROL REGISTERS

The VM04 Status and Control Registers provide local and system wide status information to the on-board MPU. By reading the Status and Control register bits, the MPU can determine VERSAbus information, interrupt status, the source of a particular BERR, and which self test to perform. The VM04 Status/Control Register bits control the MPU board's hardware by providing configuration and control information to the VERSAbus Interrupter, the Interrupt Handler, the VERSAbus Arbiter, RAMbus, and the cache memory. The Control Register bits are software alterable. The Status/Control registers are privileged resources and in special cases can be made accessible by the user. The Status and Control Register formats are shown in Table 2.

TABLE 2 — Status Register Format

<STAT0>

31			Front Panel Switch Selectable				24
TEST1	TEST0	SC	ENVIR	DB21	DB20	DB11	DB10

<STAT1>

23			Front Panel Switch Selectable				16
GP7	GP6	GP5	GP4	GP3	GP2	GP1	NOMMU

Provides VM04 Networking Base Address and useful as general purpose status

TABLE 2 — Status Register Format (continued)

<STAT2>

15			Power Up Reset Condition: "1"				08
**	**	SYSFAIL*	VBERR*	RBERR*	MMUBERR*	LRERR*	LBTO*
			← BERR Source Status Register →				

<SPARE>

07							00
**	**	**	**	**	**	**	**

(**denotes; unimplemented — always read as "1").

TEST 1, TEST0	<Power Up VERSAbus Test>	LBTO	<Local Bus Timeout BERR Flag>
SC	<VERSAbus System Controller>	LRERR	<Local Resource Access Violation Flag>
ENVIR	<Operating Environment Status Bit>	MMUBERR	<MMU BERR Flag>
DBB1, DBB0	<Serial Port B Default Baud>	RBERR	<RAMbus BERR Flag>
DBA1, DBA0	<Serial Port A Default Baud>	VBERR	<VERSAbus BERR Flag>
GP7-GP1	<General Purpose User Status Bits>	SYSFAIL	<VERSAbus System Failure Flag>
NOMMU	<No MMU Status Flag>		

Control Registers

<CNT0>

31			Power Up Reset Condition: "1"				24
VBV7	VBV6	VBV5	VBV4	VBV3	VBV2	VBV1	VBV0
← VERSAbus IRQ Vector →							
← Port A on the Z8036 Timer →							

<CNT1>

31			Power Up Reset Condition: "1"				24
Reserved	Reserved	Reserved	Reserved	Reserved	VBLOK*	32/16*	SYSRST
← Port B on the Z8036 Timer →							

<CNT2>

31			Power Up Reset Condition: "1"				24
VBIMSK7	VBIMSK6	VBIMSK5	VBIMSK4	VBIMSK3	VBIMSK2	VBIMSK1	VBISMSK
← which VERSAbus IRQ's this Interrupt Handler responds to →							

<CNT3>

23			Power Up Reset Condition: "0"				16
OVRIEN	ALLIEN	SYSFIEN	CTEST	IL2	IL1	IL0	VBIS
← Interrupt Enables, Cache Test →				← VB IRQ Request Level, Status →			

<CNT4>

15			Power Up Reset Condition; "1"				08
MONTST*	MONEN	VBCEN	32/24*	CC	CWDIS	CRDIS	BDFAIL

<CNT5>

07			Power Up Reset Condition: "1"				00
Reserved	RBDIS	RR*	RBDEN	MSD*	MSI*	MUD*	MUI*

TABLE 2 — Status Register Format (continued)

VBV7-VBV0	<VERSAbus IRQ Vector Number Register>	VBCEN	<VERSAbus Cacheable Enable>
32/16*	<VERSAbus Data Bus Width>	32/24*	<VERSAbus Address Size>
VBLOK*	<VERSAbus Read/Modify/Write Bus Lock>	CC	<Cache Clear Bit>
SYSRST	<VERSAbus System Reset>	CWDIS	<Cache Write Disable>
VBIMSK7-1	<VERSAbus Interrupt Masks>	CRDIS	<Cache Read Disable>
VBISMSK	<VERSAbus Interrupt Status IRQ Mask>	BDFAIL	<Board or System Failure>
OVRIEN	<DMA Monitor FIFO Overun IRQ Enable>	RBDIS	<RAMbus Disable>
ALLIEN	<All IRQ Enable>	RR*	<RAMbus Read Only Mode>
CTEST	<Cache Test Bit>	RBDEN	<RAMbus Decode Enable>
SYSFIEN	<SYSFAIL IRQ Enable>	MSD*	<Mask Supervisor Data>
IL2-IL0	<VERSAbus IRQ Request Level>	MSI*	<Mask Supervisor Instruction>
VBIS	<VERSAbus Interrupt Status>	MUD*	<Mask User Data>
MONTST*	<DMA Monitor Test>	MUI*	<Mask User Instruction>
MONEN	<DMA Monitor Enable>		

VERSAbus INTERFACE

The VERSAbus interface provides for VERSAbus arbitration; for buffering of data, address, and control signals; for word data manipulation to accommodate MC68020 and VERSAbus data handling differences and for Interrupt handling.

The VERSAbus arbiter arbitrates up to five levels of bus mastership on a priority basis and utilizes the MC68452 BAM chip and incorporates the Power Fail (PF) and Release on Request (ROR) options specified in the VERSAbus specification.

INTERRUPT HANDLER

The VM04 allows interrupts to the on-board CPU from up to 20 sources. The interrupt handler preprocesses interrupt sources into three groups of seven interrupts corresponding to the seven possible MC68020 interrupt levels. The groups are labeled Group 1, Group 2, and Group 3. The interrupt service priority is determined by the interrupt level and the group number. Interrupts with different interrupt levels are processed according to the standard interrupt processing discipline. Interrupts within an interrupt level are processed according to the group number.

Group 1 is reserved to VERSAbus interrupts. The interrupt handler processes Group 3 and 2 interrupts differently from Group 1 interrupts. If the interrupt being acknowledged is a Group 3 or 2 interrupt, the interrupt handler fetches the appropriate exception vector number from PROM and sends it to the CPU via the local bus. If the interrupt being acknowledged is a Group 1 interrupt, the exception vector number is fetched from the VERSAbus where it was placed by the interrupting device.

The Interrupt assignments are described in Table 3.

TABLE 3 — Interrupt Handler Priority Assignments

IRQ Level	Priority Within a Particular IRQ Level Determines the Service Order (Highest)	(Middle)	(Lowest)
	Group 3	Group 2	Group 1
7	Abort Pushbutton	ACFIRQ* if Sys Controller; else BRELIRQ*	IRQ7* (VERSAbus)
6	Serial Port SPIRQ*	SYSFIRQ*	IRQ6* (VERSAbus)
5	TMRIRQ*	VBISIRQ*	IRQ5* (VERSAbus)
4	Unassigned	RBIRQ*	IRQ4* (VERSAbus)
3	Unassigned	Unassigned	IRQ3* (VERSAbus)
2	OVRIRQ*	Unassigned	IRQ2* (VERSAbus)
1	N/A	Unassigned	IRQ1* (VERSAbus)

Within a Group x, interrupt priority decreases from top to bottom in the table.
Within a particular IRQ level, interrupt priority decreases from left to right.

TABLE 3 — Interrupt Handler Priority Assignments (continued)

Group 3 Interrupts

ABORT*	<Abort Pushbutton IRQ>
SPIRQ*	<Dual Channel Serial Port IRQ>
TMRIRQ*	<Timer IRQ>
OVRIRQ*	<DMA Monitor FIFO Overrun IRQ>

Group 2 Interrupts —

ACFIRQ*	<VERSAbus AC Power Fail Interrupt>
BRELIRQ*	<VERSAbus Bus Release Interrupt>
SYSFIRQ*	<VERSAbus System Fail Interrupt>
VBISIRQ*	<VERSAbus IRQ Request Acknowledge Interrupt>
RBIRQ*	<RAMbus Interrupt>

Group 1 Interrupts

IRQ7-1*	<VERSAbus Interrupts>

VERSAbus INTERRUPTER

The VERSAbus Interrupter can generate any level of VERSAbus interrupt under software control. When the VERSAbus interrupt is acknowledged, the Interrupter places the appropriate vector from a hardware register on the VERSAbus.

SPECIFICATIONS

The M68KVM04 specifications are shown in Table 4.

TABLE 4 — M68KVM04 Specifications

Characteristics	Specifications
Power requirements	+5 Vdc, 6.0 A (Typ) +12 Vdc, 100 mA (Typ) (Maximum current requirements TBD) −12 Vdc, 100 mA (Typ)
Clock signal	16.67 MHz clock frequency
Addressing Total system size ROM/EPROM	 4 Gigabytes (32 bit addressing) Two 28-pin sockets for 32K-byte devices using +5 Vdc only.
Operating Temperature	0° to 70°C
Storage Temperature	−40° to 85°C
Relative Humidity	5% to 95% (non-condensing)
Physical Dimensions Height Width Thickness	 9.25 in (23.50 cm) 14.50 in (36.83 cm) 0.60 in (1.52 cm)

Ordering Information

Part Number	Description
M68KVM04-1	VERSAmodule 32-bit Monoboard Microcomputer with MC68020 CPU and Hardware Memory Management plus 4K Long Words (16K bytes) of software transparent cache. Includes User's Manual.
M68KVM04-2	VERSAmodule 32-bit Monoboard Microcomputer with MC68020 CPU and 4K Long Words (16K bytes) of software transparent cache. Includes User's Manual.

Related Documentation

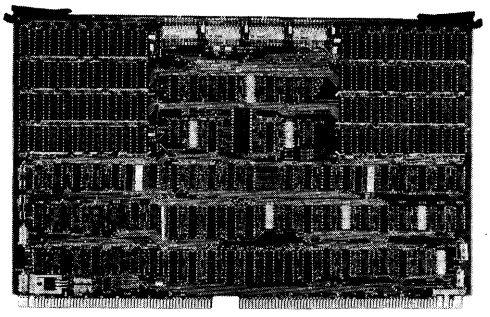
Part Number	Description
M68KVM04/D1	VERSAmodule 32-bit Monoboard Microcomputer User's Manual
TBD	MC68020 MPU Technical Summary
TBD	MC68020 MPU User's Manual

Related Products

Part Number	Description
M68K2RBBUG4	020bug Debugging Package for VERSAmodule 32-bit Monoboard Microcomputer. Includes EPROM set and Dual RS-232C Serial I/O Cable, plus User's Manual.
M68KVM13-1	VERSAmodule 1024K Byte Dynamic RAM Module with RAMbus. For use with VM04 Monoboard Microcomputer.
M68KVM13-2	VERSAmodule 4096K Byte Dynamic RAM Module with RAMbus. For use with VM04 Monoboard Microcomputer.

M68KVM10-3
M68KVM10-2

VERSAmodule 128K/64K Byte RAM



- 128K And 64K Byte Versions
- VERSAbus Compatible
- Includes Byte Parity
- Addressable in Bytes or 16-Bit Words
- Blocks of 32K Bytes Separately Addressable
- Less Than 400 ns Access Time
- 0°C-70°C Operating Temperature Range

The two VERSAbus compatible Memory Modules (128K and 64K) provide the user with a wide choice of RAM storage elements. The base address is separately selectable via a 10-bit DIP switch. This address flexibility provides for placement of memory throughout the full 16 megabyte range of the MC68000 microprocessor.

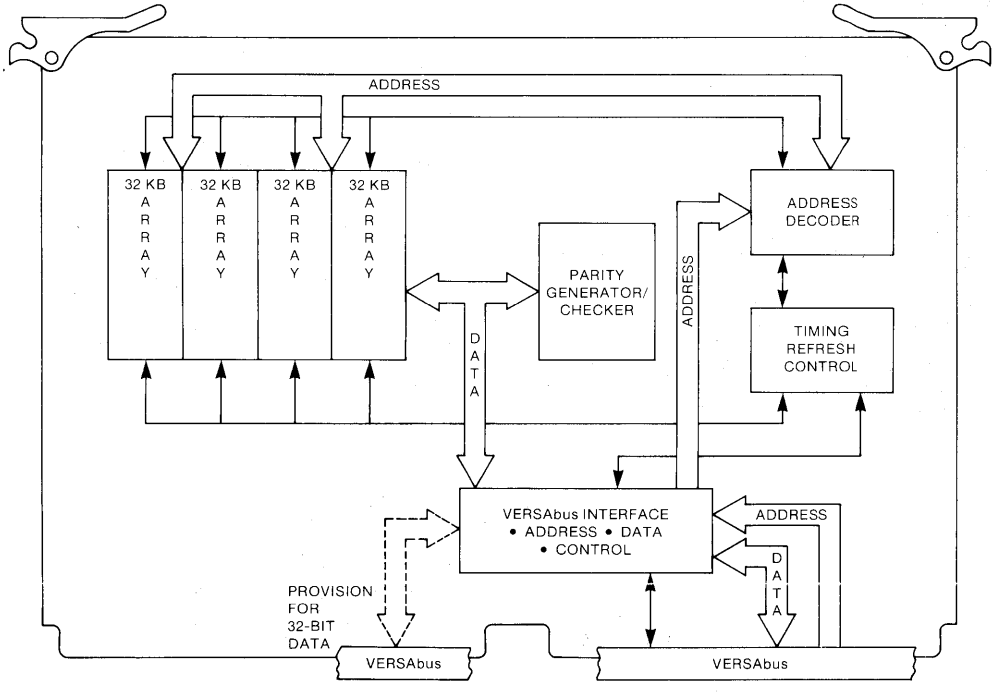
These memory modules are supplied with byte parity. If in the course of a normal read cycle, a parity error is detected, the module will generate a VERSAbus Bus Error (BERR) and transfer it to the system VERSAbus.

Because these Memory Modules are VERSAbus compatible, there are many products in which these modules can be utilized for memory expansion. VERSAmodule based systems of course fall into this category, including Motorola's VMC 68/2 Microcomputer System, which is based on VERSAmodules. EXORmacs, Motorola's development system for the M68000 family of products also uses VERSAbus as its system interconnect structure.

Memory Module Specifications

Characteristic	Specification
Memory Device Type	MOS Dynamic RAM (16,384 × 1)
Memory Organization 64K Byte 128K Byte	16K × 16-Bit Words Plus Two Parity Bits Two Memory Rows (32K Words) Four Memory Rows (64K Words)
Parity Check	Odd Parity
Read Cycle Time	425 ns (Max.)
Access Time	390 ns (Max.) From Data Strobe (DS*) To DTACK*
Error Check Time	Error Check Time Is Included In The Access Time Of 390 ns.
I/O Signals	VERSAbus Compatible
Temperature Operating Storage	 0° to 70°C -40°C to +85°C

VERSAmodule 128K/64K Byte RAM



Memory Module Specifications (continued)

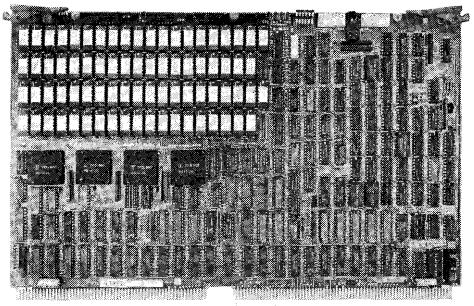
Characteristic	Specification
Relative Humidity	0 To 90% (Non-condensing)
Operating Temperature Range	0° To 70°C
Power Requirements	
64K Byte	+5.0 Vdc @ 3.5 A (Max.) Standby +5.0 Vdc @ 3.5 A (Max.) Operating +12 Vdc @ 80 mA (Max.) Standby +12 Vdc @ 700 mA (Max.) Operating -12 Vdc @ 7.3 mA (Max.) Standby -12 Vdc @ 9.1 mA (Max.) Operating
128K Byte	+5.0 Vdc @ 3.5 A (Max.) Standby +5.0 Vdc @ 3.5 A (Max.) Operating +12 Vdc @ 148 mA (Max.) Standby +12 Vdc @ 741 mA (Max.) Operating -12 Vdc @ 7.3 mA (Max.) Standby -12 Vdc @ 9.1 mA (Max.) Operating
Dimensions	
Height	9.25 Inches (23.49 cm)
Width	14.50 Inches (36.83 cm)
Thickness	0.6 Inch (1.52 cm)

Ordering Information

Part Number	Description
M68KVM10-3	128K Byte Dynamic RAM Module. Includes User's Manual.
M68KVM10-2	64K Byte Dynamic RAM Module. Includes User's Manual.
M68KDRAM/D2	Dynamic RAM Memory Module User's Manual.
M68KVBS	VERSAbus Specification.

M68KVM11-1
M68KVM11-2

VERSAmodule 256K/512K Byte RAM



- VERSAbus Compatible
- 256K Byte and 512K Byte Memory Versions Utilizing 64K × 1 HMOS RAM Technology
- Error Detection and Correction (EDAC) Circuitry
- May Be Set for 16 or 32-Bit Data Word Operation
- Memory Divisible into Two Independent Segments Each With Its Own VERSAbus Base Address Switches
- Memory Base Address Switches Permit Placement on 128/256K Byte Boundaries Throughout the 16 Mbyte VERSAbus Address Space
- Eight 16-Bit Word Control and Status Registers Addressable as I/O for Initialization, Mode Control, Error Interrupt/Bus Error Control and Error Location and Logging
- Four Board-Edge Diagnostic Indicators and a Multi-digit 7-Segment Display for RAM Chip/Row Error Identification
- Provides for Fast Sequential Access When Two M68KVM11 Boards are Set for Interleaved Operation
- 0°C–70°C operating temperature range

FUNCTIONAL DESCRIPTION

The facilities provided on this board make it a highly reliable and maintainable module suitable for use in high speed memory intensive applications.

Because these Memory Modules are VERSAbus compatible, there are many products in which these modules can be utilized for memory expansion. VERSAmodule based systems of course fall into this category, including Motorola's VMC 68/2 Microcomputer System, which is based on VERSAmodules. EXORmacs, Motorola's development system for the M68000 family of products also uses VERSAbus as its system interconnect structure.

RAM FACILITIES

The 256/512K byte RAM Module (see Block Diagram) provides high capacity global memory for the VERSAmodule System. The module is available in two capacity versions, the M68KVM11-1 providing 256K bytes of memory and the M68KVM11-2 providing 512K bytes of storage capacity. This high capacity module is attained using high density 64K × 1-bit dynamic RAM HMOS chip technology. The board utilizes Error Detection and Correction (EDAC) circuitry to produce highly reliable information storage. On-board maintenance and diagnostic facilities are provided to allow on-line system error logging as well as manual aids for efficient off-line preventative maintenance using system diagnostic programs.

The module is divided into two segments, 128K byte segments on the 256K byte module and 256K byte segments on the 512K byte module. Each segment has its own base address switches that permit placement on 128K/256K byte boundaries throughout 16 Mbyte VERSAbus address space. Where high speed sequential access is required, two M68KVM11 boards may be set for interleaved memory operation. Interleaved operation is implemented setting one board to respond to even 16-bit word addresses and the second board to respond to odd-word addresses. In interleaved operation, second and subsequent words in a sequential access (such as DMA operation) are pre-accessed and buffered on-board for fast data transfers.

EDAC

On-board EDAC circuitry utilizes a modified Hamming Code to detect all multi-bit errors and correct all single bit errors. Corrected information is rewritten into the location as well as transferred to the requesting processor.

For operational as well as VERSAbus addressable maintenance and diagnostic purposes, eight 16-bit word Status and Control Registers are implemented in conjunction with EDAC. When single or multibit errors are detected, a status bit is set identifying the type of error. Each of these status error conditions are capable of generating an interrupt to VERSAbus. These interrupts are program initiated via a corresponding interrupt enable bit in a control register. The VERSAbus interrupt priority level is selectable to any of the seven VERSAbus interrupt request levels. A VERSAbus Bus Error signal (BERR) may also be generated and is also program initiated.

Initialization and Modes of Operation

All locations throughout the RAM are initialized on power-up. All status and control bits are cleared.

There are three modes of operation:

- Normal EDAC Operation
- Operation with EDAC Disable
- Read/Write Check Bits

The Normal Mode is generally used in system operation. The EDAC Disabled Mode and the Read Check Bits Mode can be used in maintenance and diagnostics to verify EDAC operation by writing errors in any bit position or positions in any RAM location including Check Bits.

Normal Operation and Error Reporting

During normal operation, if a multi-bit error is detected the "Multi-bit Error" LED is illuminated, the corresponding bit is set in the Status Register and a VERSAbus interrupt request is generated if not masked under software control. A Bus Error Signal may also (or alternately) be generated if not masked under software control. The system address and board chip/row designation are stored in the Status Registers for system error logging and maintenance purposes.

There are five LEDs on the board:

- MULTI-BIT ERROR LED
- SOFT ERROR LED
- HARD ERROR LED
- OVERFLOW LED
- SYSTEM FAIL LED

The MULTI-BIT ERROR LED is illuminated when a non-correctable error is detected on a read cycle. The SOFT ERROR LED is illuminated when a correctable error is detected. The HARD ERROR LED is illuminated when a second single bit (correctable) error is detected in any one RAM chip.

The OVERFLOW LED is illuminated when the sixteenth correctable single bit error occurs.

The SYSFAIL LED is illuminated upon powerup.

Each of these LED's has a corresponding bit in an on-board status register and is capable of generating an interrupt if enabled. In addition, each may be "reset" by a write to an on-board control register.

Maintenance

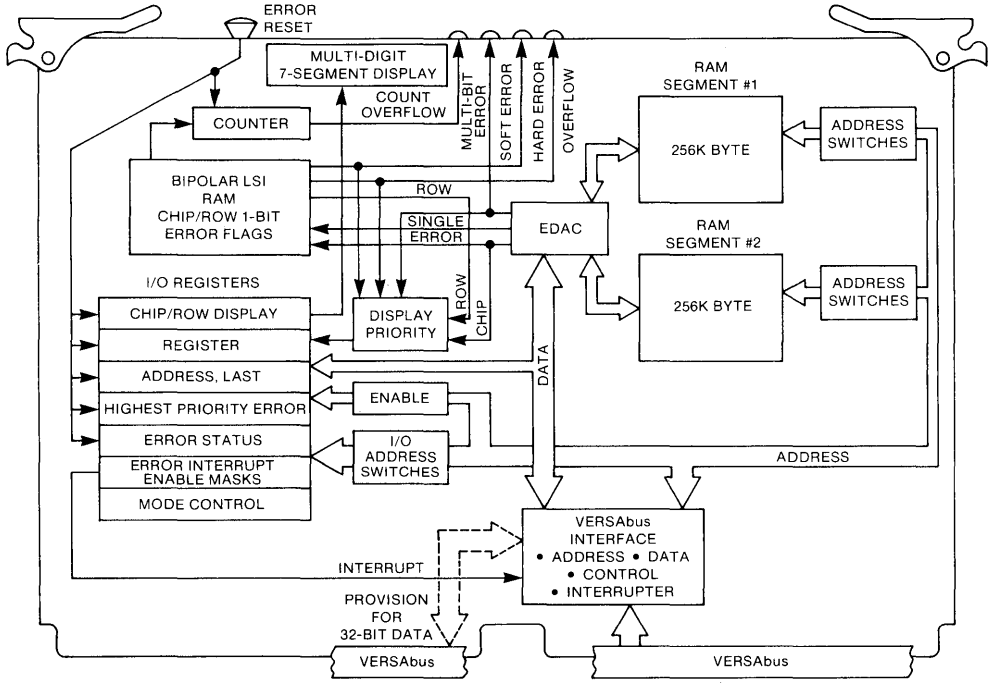
The five board-edge LED's together with a multi-digit 7-segment display and an Error Reset Switch provide manual/visual aid to maintenance personnel during off-line preventative maintenance and repair. The display indicates the board chip position or row number of the highest priority failing RAM chip or chip row for single and multi-bit errors, respectively. A multi-bit error has highest priority followed by "hard error" and then "soft error". The Error Reset Switch clears the display LED's and all current Error Status (just as occurs on power-up).

Specifications

Characteristic	Specification
Storage Capacity:	256K Bytes or 512K Bytes
Word Length:	16 Bits/32 Bits
Read Access Time — Normal: Fast Access:	450 Nanoseconds (Nom) 160 Nanoseconds (Nom)
Write Access Time:	160 Nanoseconds (Nom)
Read Cycle Time:	525 Nanoseconds (Nom)
Write Cycle Time:	525 Nanoseconds (Nom)
Refresh Cycle Time:	15 Microseconds (Max)
Mode(s) of Operation:	Normal Read Cycle Fast Access Read Cycle Write Cycle Refresh Cycle
Input:	1 Schottky TTL Load per Output Line
Output:	Open Collector Output (I_{sink} max = 48 mA) Tri-State Output (I_{sink} max = 48 mA)
Data Input/Output:	32 Lines (D00-D31)
Input/Output Controls:	14 Lines Input 11 Lines Output
Input Address:	23 Lines (A01-A23)
Temperature Range:	0°C to 70°C (Operating) -40°C to 85°C (Non-operating)
Relative Humidity:	5% to 90% without Condensation
Input Power:	+5.0 V @ 5.0 A (Typical) +5.0 V @ 6.5 A (Max)
Dimensions:	14.50 in. (368.30 mm) Wide 9.25 in. (234.95 mm) High

3

VERSAmodule 256K/512K Byte RAM

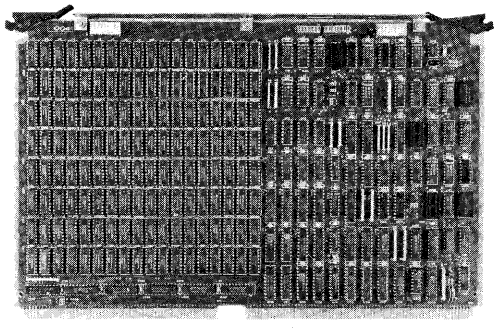


Ordering Information

Part Number	Description
M68KVM11-1	256K Byte RAM Module, VERSAbus compatible with Error Detection and Correction features. Includes User's Manual.
M68KVM11-2	Same as -1 version, but 512K byte size.
M68KVM11/D2	256K/512K Byte Dynamic RAM Memory Module User's Manual
M68KVBS	VERSAbus Specification

M68KVM12
M68KVM12-2

VERSAmodule 1024K/4096K Byte RAM



- VERSAbus Compatible
- Parity Generation & Error Checking Circuitry
- Longword (32-bit), Word (16-bit), or Byte (8-bit) Data Transfers
- Selectable 24 or 32-bit Addressing
- Memory Base Address Settable on 64K Byte Boundaries Throughout the VERSAbus Address Space
- High Speed Operation:
  - — 375 ns Read Access Time
  - — 180 ns Write Access Time
  - — 475 ns Cycle Time (Read or Write)
- 0°C–70°C Operating Temperature Range
- Provision for Adding Battery Backup

FUNCTIONAL DESCRIPTION

Used for high speed storage in VERSAbus systems, M68KVM12 (VM12) is a 4096K byte (-2) or 1024K byte dynamic RAM module with parity. It attains its capacity and performance using high density HMOS 256K x 1-bit (-2) or 64K x 1-bit dynamic RAM chips. VM12 has parity generation and detection circuitry which, together with accessible control and status registers, can be used for error detection and memory diagnostics. Figure 1 is a functional block diagram of the module.

Because of its VERSAbus interface, VM12 is compatible with other systems using this interconnect structure. These include VERSAmodule-based systems such as Motorola's VMC 68/2 Microcomputer System and the EXORmacs Development System which provides extensive and powerful support for development of systems based on the MC68000 family of products.

ADDRESS MAPPING

The VM12 has two switches used to set module memory to begin on any 256K byte (-2) or 64K byte boundary within the selected 16 Mbyte page. Sixty-four such pages exist in systems supporting the VERSAbus Extended Addressing Mode. A header allows the upper eight address select bits (along with the associated page decode circuitry) to be jumper disabled if the module is to be used in systems limited to 24 address lines.

The ending address of module memory is normally the sum of the starting address and the board population. However, if the starting address is within one megabyte of an upper page boundary, the ending address is the page boundary.

The VME12 utilizes one word within VERSAbus I/O Space as a location for the Control/Status Register. A header is provided for jumper selection of the Control/Status Register location within the FF0000 through FFFFFF address range.

ADDRESS MODIFIER CODE RESPONSE

VERSAbus address modifier line decoding on VM12 provides response to both standard and extended, supervisory and non-privileged program and data accesses. VM12 has headers for selecting the standard or extended addressing mode and for selecting the address modifier line decoding for that mode. The short supervisory I/O access code (15 hex) must be placed on the address modifier lines to access the Control/Status register.

An 82S129 bipolar PROM is used for address modifier line decoding. This PROM is socketed to facilitate application of VM12 in systems having special requirements.

CONTROL/STATUS REGISTER

The VM12 Control/Status Register is accessible as a 16-bit word in which the most significant bit provides parity error indication. The three least significant bits facilitate control of VM12 parity circuitry by offering a means of enabling/disabling under software control the parity generation, the wrong parity generation and the parity error checking circuitry. The Control/Status Register is shown in Figure 2.

FIGURE 1 — M68KVM12 Block Diagram

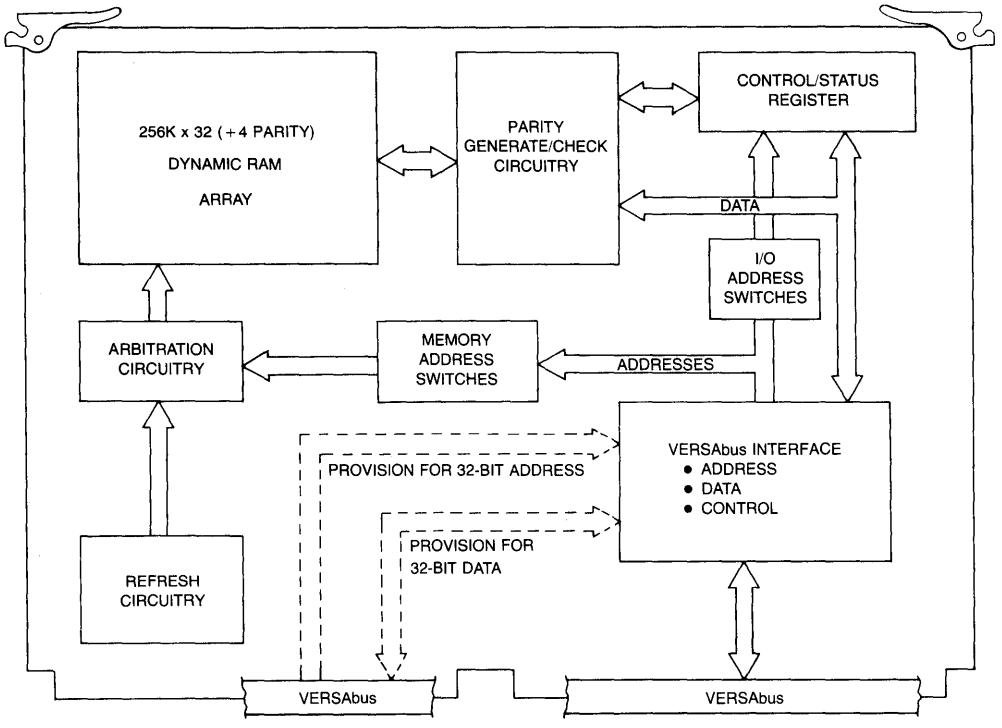
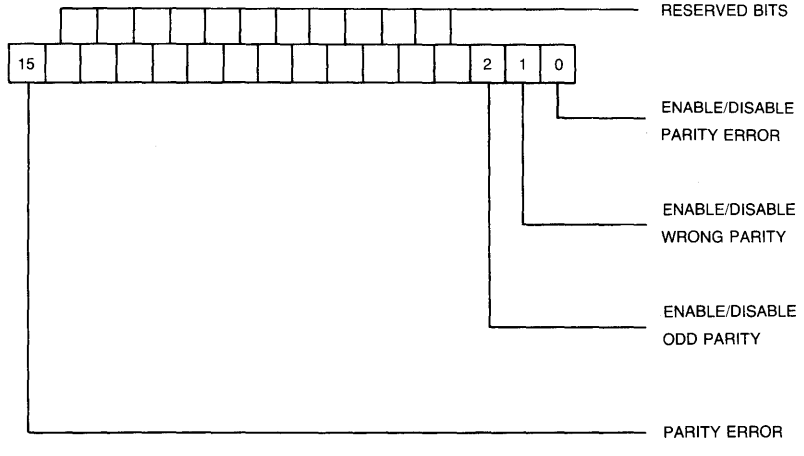


FIGURE 2 — M68KVM12 Control/Status Register



On power up, all Control/Status bits are cleared. Bits 0 through 3 are automatically cleared by an active SYSRESET* signal on the VERSAbus.

PARITY FUNCTIONS

In normal operation, VM12 generates and writes an even parity bit with each byte stored in its memory. When a byte is read, even parity is again generated and compared with the stored parity bit. On detection of an error, the VERSAbus BERR* signal line is driven low and bit 15 in the Control/Status register is set.

A header is provided for jumper disabling or enabling of bus error generation. With the bus error circuitry jumper enabled, generation of the signal may be selected under software control.

To facilitate the testing of memory, the Control/Status register has two bits for software control of (1) even parity generation and (2) wrong parity generation. Bit 2 may be set to inhibit the writing of the even parity value into the parity bit on writing a byte into memory. Bit 1 may be set to enable the writing of the wrong parity value into a parity bit.

BATTERY BACKUP

The memory array and refresh circuitry may optionally be powered by +5 V or +5 V STDBY supplies. When the latter option is employed, the VM12 is capable of retaining data indefinitely while consuming minimal current (1.75 A typ) from the +5 V STDBY supply.

TABLE 1 — M68KVM12 Specifications

Characteristics	Specifications
Storage Capacity	4096K/1024K Bytes
Data Length	Byte (8-bits), Word (16-bits) or long word (24-bits)
Memory Organization	One 4096K/1024K Byte Block
Write Access Time	160 ns (max)
Read Access Time	375 ns (max)
Write/Read Cycle Time	475 ns (max)
Refresh Cycle Time	15 ns (max)
Error Detection	Even Byte Parity, on Read Access
Power Requirements	+5 Vdc @ 4.2 A (typical, operating) +5 Vdc @ 5.0 A (maximum, operating) +5 Vdc @ 1.75 A (typical, standby) +5 Vdc @ 2.1 A (maximum, standby)
Environmental	
Operating Temperature	0°C to 70°C
Storage Temperature	−20°C to +85°C
Relative Humidity	0% to 95% (non-condensing)
Dimensions	
Height	9.25 in. (23.5 cm.)
Width	14.5 in. (36.8 cm.)
Thickness	0.6 in. (1.5 cm.)

Ordering Information

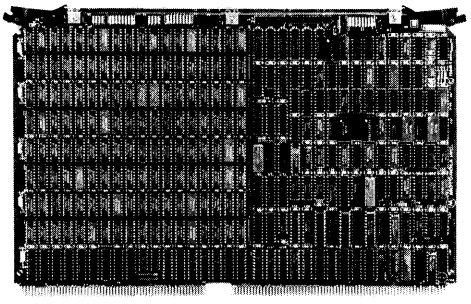
Part Number	Description
M68KVM12	VERSAmodule 1024K Byte Dynamic RAM with Byte Parity. Includes User's Manual
M68KVM12-2	VERSAmodule 4096K Byte Dynamic RAM with Byte Parity. Includes User's Manual
M68KVM12/D1	M68KVM12 1024K/4096K Byte Dynamic RAM with Byte Parity User's Manual

Related Documentation

M68KVBS/D4	VERSAbus Specification Manual

M68KVM13-1
M68KVM13-2

VERSAmodule 1024K/4096K Byte Dynamic RAM w/RAMbus



- Supports VERSAbus/RAMbus
- Dual Ported — 32-bit address and data VERSAbus and multiplexed 32-bit address/data RAMbus interface
- Interleaving — Two-way on-board read interleaving on RAMbus or VERSAbus
- Configurable Array — Dynamically alterable to be partial or full private to the RAMbus port. Selectable in 1/4 population increments
- Byte Parity Generation and Error Checking Circuitry
- Longword (32-bit), Word (16-bit), or Byte (8-bit) Data Transfers
- VERSAbus Addressing — Header Selectable 24- or 32-bit Addressing on VERSAbus interface, 32-bit addressing on RAMbus
- Memory Base Address Settable on-board size boundaries throughout VERSAbus and RAMbus Address Space
- Transparent and Synchronous Refresh Support — Utilize on-board refresh signal or RAMbus synchronous refresh signal to synchronize VM13 refresh to an external source (i.e., Video Display Generator)

Functional Description

The M68KVM13 (VM13) is a dual ported 1024K/4096K byte dynamic RAM module with parity for use with VERSAbus and RAMbus. The VM13 RAMbus interface is tailored to the MC68020 and is specifically designed to enhance M68KVM04 (VM04) performance. The RAMbus interface, since it is a high speed dedicated memory bus, allows concurrent DMA transfers on VERSAbus with RAMbus transfers between the VM04 and VM13.

The VM13 attains its capacity and performance using high density 64K/256K × 1-bit dynamic RAM devices. The VM13 has parity generation and detection circuitry which, together with accessible control and status registers, can be used for error detection and memory diagnostics.

Figure 1 is a functional block diagram of the module.

Because of its VERSAbus interface, the VM13 can be used in systems using this interconnect structure. Care must be used with existing VERSAmodule-based systems, since RAMbus and the extended 32-bit data and address use P2 of the VERSAbus connector. The pins on connector P2 of the existing VERSAbus system are commonly used as I/O pins.

VERSAbus Address Mapping

The VM13 has two switches (12 bits) used to set module memory to begin on any 1024K byte boundary. A header allows the upper eight address select bits (along with the associated page decode circuitry) to be jumper disabled if the module is to be used in systems limited to 24 address lines.

The ending address of module memory is the sum of the starting address and the board population.

The VM13 utilizes one byte within VERSAbus I/O Space as a location for the Control/Status Register. Selection of the Control/Status Register location within the VERSAbus I/O address range is switch selectable.

VERSAbus Address Modifier Code Response

VERSAbus address modifier line decoding on the VM13 provides response to both standard and extended, supervisory and non-privileged program and data accesses. The VM13 has headers for selecting the standard or extended addressing mode and for selecting the address modifier line decoding for that mode. The short supervisory I/O access code (15 hex) must be placed on the address modifier lines to access the Control/Status register.

An 82S129 bipolar PROM is used for address modifier line decoding. This PROM is socketed to facilitate application of the VM13 in systems having special requirements.

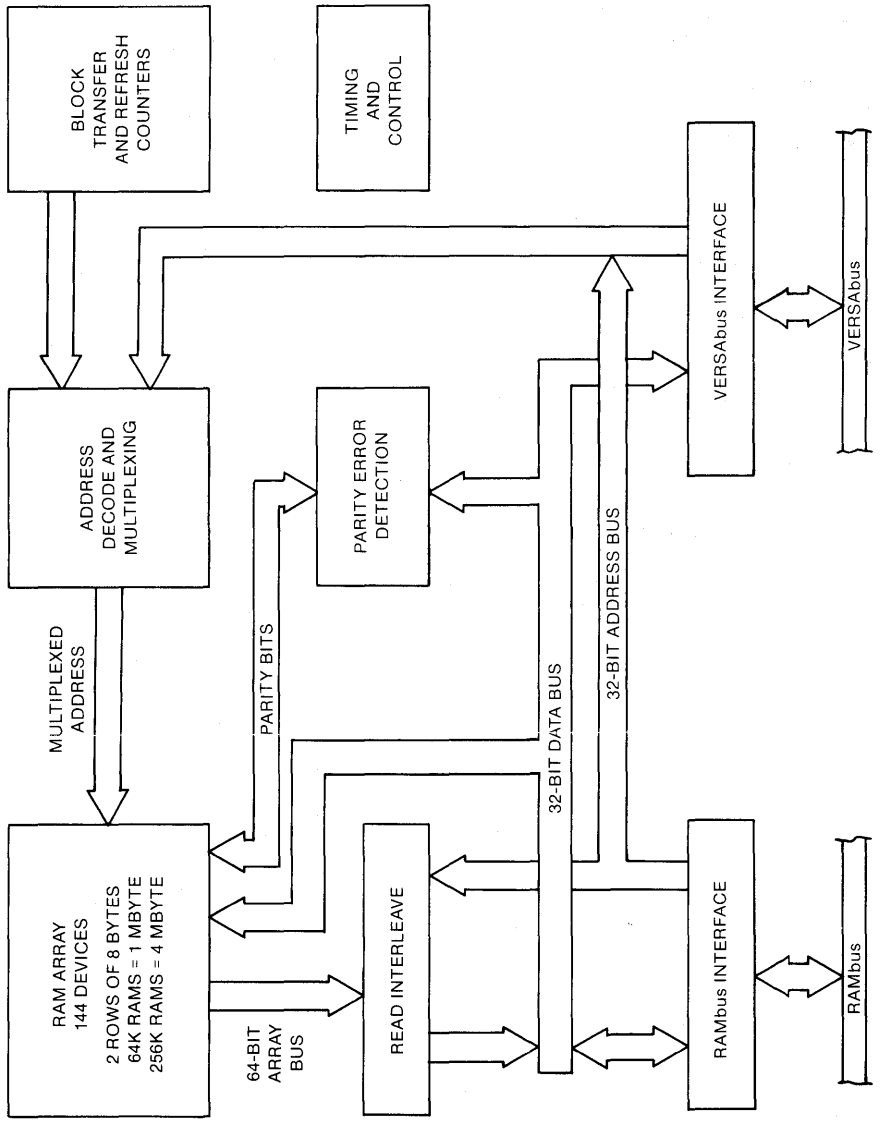


FIGURE 1 — M68KVM13 Block Diagram

RAMbus Address Mapping

The VM13 has two switches (12 bits) used to set module memory to begin on any 1024K byte boundary. The ending address of module memory is the sum of the starting address and the board population.

Global/Private memory mapping is dynamically alterable via the Status/Control register. One bit enables RAMbus private mode. Alternately, two bits determine the size of the private memory in 256K/1 Mbyte segments. The private memory is allocated from the upper to lower segments on the VM13. The system configurer, at the time private memory is allocated, must be cognizant of any holes that may be introduced into the memory map due to the allocation of private memory.

Status/Control Register

The VM13 Status/Control block provides VERSAbus accessible registers for VM13 control and status. The Control Register is writable and readable from the VERSAbus. The only status is parity error Bit 7 of the Control Register. The Control and Status Registers are reset at power up. The Parity Error Status bit is resettable by writing a zero to Bit 7 of the Control Register. The Control and Status Registers are defined below.

PRBEREN: <Private Bus Error Enable> When set, PRBEREN causes a Bus Error to be issued to the VERSAbus device attempting to access memory that has been allocated private to RAMbus by the PR1, 0 bits, or the RP bit of the VM13 Control Register.

MCACHE: <Memory Cacheable> When set allows the RAMbus port to be cached by the VM04 system. MCACHE enables assertion of MCACHE* on the RAMbus with the same timing as MASACKO, 1*. Indicating that the selected memory is cacheable.

WWP: <Write Wrong Parity> When set, causes the wrong parity to be written to the addressed location for diagnostic purposes.

EPER: <Enable Parity Error Report> When set, allows the VM13 error detecting circuitry to report errors to the selecting device. Errors are normally indicated by the BERR* or MERR* signals and the PE bit in the VM13 Status Control Register.

VM13 Status/Control Register Format

D7	D6	D5	D4	D3	D2	D1	D0
PE	RP	PR1	PR0	PRBEREN	MCACHE	WWP	EPER

PE: <Parity Error> The VM13 status bit, when set indicates that a parity error has occurred from a VERSAbus or RAMbus read access. It can be cleared by writing a zero from the VERSAbus or can be set for diagnostic purposes.

RP: <RAMbus Private> When set, RP disables any VERSAbus access to the VM13 DRAM array.

PR1,0: <Private RAM> PR1 and PR0 allocated the VM13 DRAM array area to the VERSAbus by the following table. These bits are don't cares when the RP bit is set.

PR1,0	Memory Segment VERSAbus Accessible
0 0	3, 2, 1, 0
0 1	3, 2, 1
1 0	3, 2
1 1	3

Parity Functions

In normal operation, the VM13 generates and writes an even parity bit with each byte stored in its memory. When a byte is read, even parity is again generated and compared with the stored parity bit. On detection of an error, the VERSAbus BERR*, or RAMbus MERR*, signal line is driven low and bit 7 in the Control/Status register is set.

A header is provided for jumper disabling or enabling of bus error (BERR*, MERR*) generation. With the bus error circuitry jumper enabled, generation of the signal may be selected under software control.

To facilitate the testing of memory, the Control/Status register has two bits for software control of (1) enable parity error reporting and (2) wrong parity generation. Bit 1 may be set to enable the writing of the wrong parity value into a parity bit. Bit 0 enables parity error reporting.

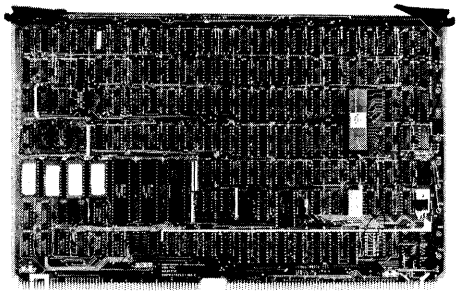
TABLE 1 — M68KVM13 Specifications

Characteristics	Specifications
Power Requirements	M68KVM13-1 (1 Mbyte) — +4.75 to 5.25 Vdc @ 6.8 A (max) M68KVM13-2 (4 Mbyte) — +4.75 to 5.25 Vdc @ 6.8 A (max)
Operating Temperature	0° to 70°C
Storage Temperature	-40° to 85°C
Relative Humidity	5% to 95% (non-condensing)
Physical Dimensions Height Width Thickness	 9.25 in. (23.50 cm) 14.50 in. (36.83 cm) 0.60 in. (1.52 cm)
Storage Capacity	1 Mbyte (for M68KVM13-1) 4 Mbyte (for M68KVM13-2)
Data Transfer Size	8, 16, 32 bits
Error Detection	Even Byte Parity
Data Input/Output	32 bit VERSAbus/RAMbus data
Input Address	32 bit VERSAbus/RAMbus addressing

Ordering Information

Part Number	Description
M68KVM13-1	VERSAmodule 1 Mbyte Dynamic RAM with Byte Parity. Includes User's Manual.
M68KVM13-2	VERSAmodule 4 Mbyte Dynamic RAM with Byte Parity. Includes User's Manual.
M68KVM13/D1	VERSAmodule 1M/4M Byte Dynamic RAM User's Manual.

VERSAmodule Floppy Disk Controller



- VERSAbus compatible
- Host Computer commands serviced by on-board standard Intelligent Peripheral Controller based on the MC6801 MCU
- Floppy Disk Controller controlled by IPC provides for control of up to four Motorola EXORdisk floppy disk drives
- Floppy Disk formatted capacity from 1/2 Mb for one EXORdisk II two-drive unit, to 2 Mb for two EXORdisk III two-drive units
- Shared RAM command channel and high-speed DMA data channel to VERSAbus
- Provides 16 × 16-bit word on-board data buffer
- Capable of addressing DMA transfers to any address in 16 Mb VERSAbus memory space
- Interrupt or ready status indicates completion of data transfer
- Supported by VERSAdos Operating System
- On-board Self-Test and Diagnostic Firmware
- 0°-70° C Operating Temperature Range

The Floppy Disk Controller (FDC) adds mass storage capability to any MC68000 VERSAbus-compatible system and is fully supported by Motorola's VERSAdos Operating System. The FDC provides the necessary functions, utilizing a combination of hardware and software, to control the operations of the EXORdisk II/III Disk Drive Unit. Specifications for the FDC are shown in Table I. The FDC is comprised of two major functional groups, the FDC interface and the Intelligent Peripheral Controller (IPC). A functional block diagram of the FDC is illustrated in Figure 1.

Floppy Disk Controller Interface

The Floppy Disk Controller Interface monitors the state of the disk drives and controls the reading and writing of data to/from the floppy disk media. The data and status information is passed to/from the floppy disk controller interface via the IPC. This provides better efficiency in the utilization of the host processor and bus.

Intelligent Peripheral Controller

The IPC utilizes an MC6801 MCU that provides an intelligent interface between the peripheral and the host system.

The IPC isolates the host system from the peripherals, which enhances the operation of the total system by performing trivial tasks that would normally be performed by the host system. The host system need be concerned only with the sending of the command packet to the FDC and receiving the status indicating that the transfer has been completed. Also, DMA data transfers into main memory are performed by the IPC, thus freeing the host processor. The IPC also performs diagnostic tests during power-up or by command of the host processor during on-line operations.

The IPC consists of three basic parts, the command/status channel, the DMA data channel and the MC6801 MCU. The command/status channel is a 128-byte shared memory which can be addressed by either the MCU or the system VERSAbus. The data channel is a DMA controller that transfers data directly between the IPC and main memory. The MCU operates in the expanded multiplexed mode, utilizing ROM for diagnostics. The MCU also provides the system timer, RAM, and I/O which are used extensively by the FDC firmware.

A typical installation utilizing the FDC is illustrated in Figure 2. The FDC is installed in a VERSAbus-compatible backplane and the EXORdisk II/III Disk Drive Unit is connected via a 40-pin ribbon cable to the backplane I/O pins.

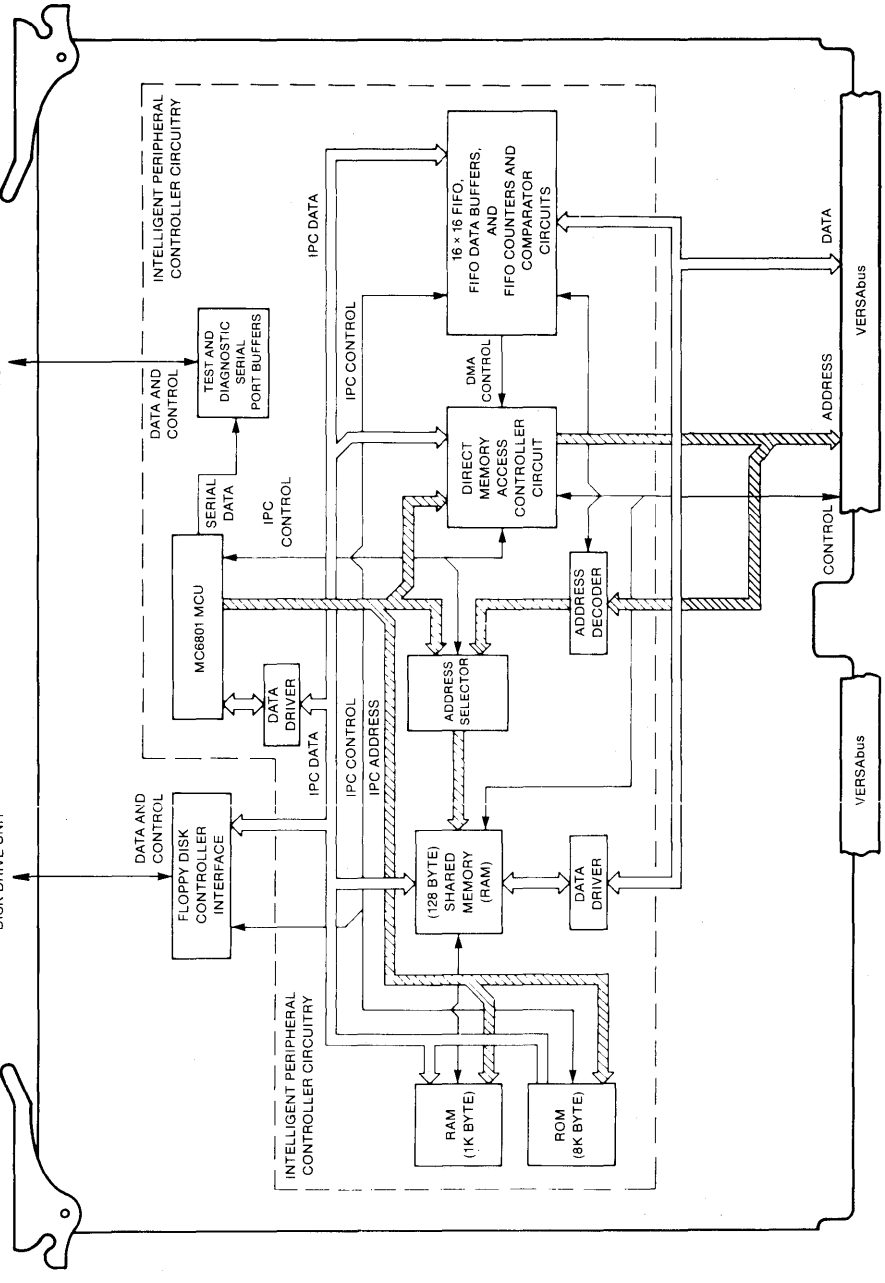


FIGURE 1 — Floppy Disk Controller Functional Block Diagram

FIGURE 2 — Typical Floppy Disk Controller Installation

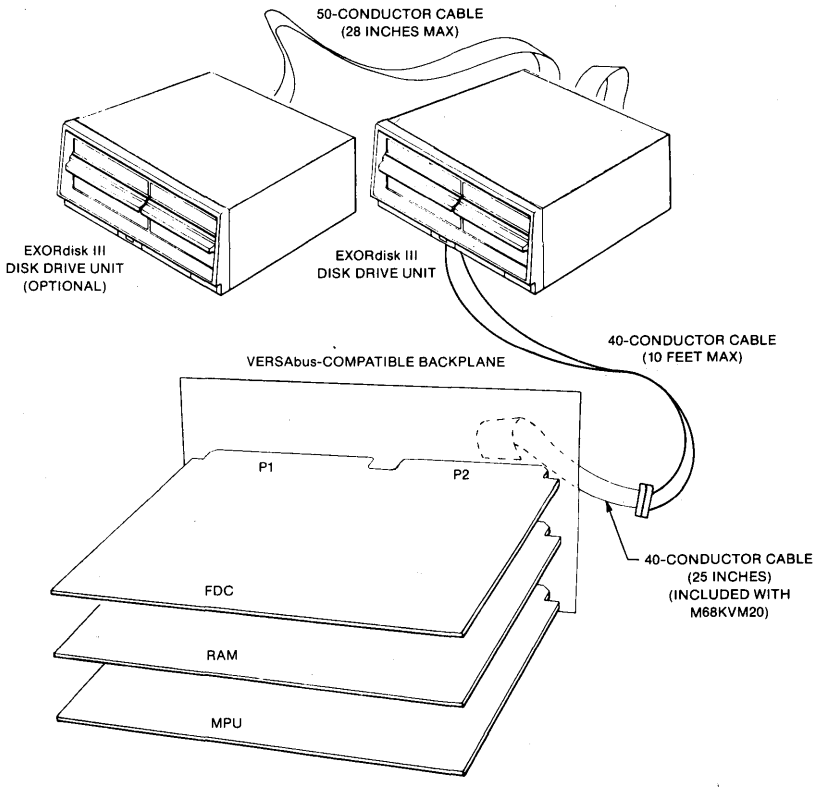


TABLE 1 — Specifications

Characteristics	Specifications
Compatibility Computer	Any VERSAmodule/VERSAbus monoboard microcomputer.
Floppy Disk Drive	EXORdisk II or III.
Media Format	EXORmacs VERSAdos/MDOS compatible.
Drive Data Rate	250K bits/sec, max.
Drive Error Detection	CRC check.
Sectors/Track	26
Sector Size	128 Bytes (VERSAdos commands allow transfers in increments of 128- or 256-byte blocks).
VERSAbus Command Channel	Permits processing of one command at a time for status, or data transfer in full block increments of up to 4,096 blocks (512 Kbytes).

TABLE 1 — Specifications (continued)

Characteristics	Specifications
VERSAbus Data Channel	VERSAbus data transfers performed via DMA at word rates of up to 4 Mwords (8 Mbytes)/sec., max., with actual rate determined by cycle time of the addressed VERSAbus memory board. With the M68KVM10 128 kB RAM module the maximum transfer rate is approximately 1.5 Mwords/sec.
FIFO Data Buffer	16-word FIFO used to time buffer synchronous disk data to asynchronous system bus.
VERSAbus Master	During DMA, bus mastership retained for a maximum of 16 bus transfer cycles to memory per data transfer burst before bus is released to other masters.
Power Requirements (Typical)	+5 Vdc ± 5% @ 3.5 A +12 Vdc ± 5% @ 110 A -12 Vdc ± 5% @ 70 A
Temperature Operating Storage	 0° to +70° C -40° to +85° C
Relative Humidity	0 to 90% (non-condensing)
Dimensions Height x Width x Thickness (including components)	 9.25 x 14.50 x 0.6 inches (32.5 x 36.8 x 1.5 cm)

Ordering Information

Part Number	Description
M68KVM20	VERSAmodule Floppy Disk Controller (FDC). This module controls up to 4 floppy disk drives. Drives compatible with this module are a single 0.5 Mb EXORdisk II unit (2 drives) or one or two EXORdisk III units (4 drives). The drive systems are connected to the module through the VERSAbus backplane via a single daisy-chain cable (M68KVMCFD1) included with this module. This cable is VERSAmodule System Chassis (M68KVMCH1) mountable and provides connection between the VERSAbus backplane i/O connector and the EXORdisk II/III drive cable. User's Guide Included.

Documentation

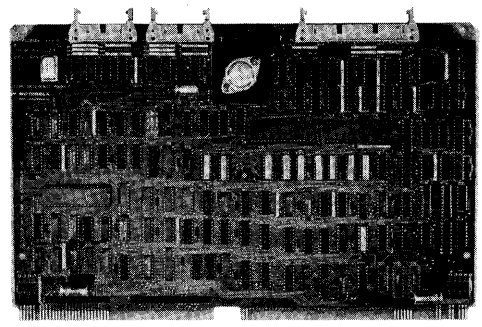
M68KVM20/D1	FDC User's Manual
M68KIPCS/D1	M68000/IPC Command Channel Software Interface Reference Manual
M68KVBS/D4	VERSAbus Specification
M68SFDU3R/D1	EXORdisk II/III Disk Drive Unit Maintenance Manual

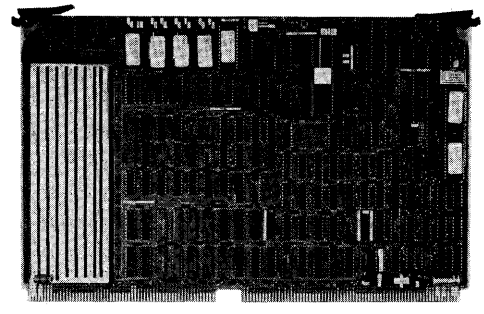
Options

Part Number	Description
M68KFD1100	EXORdisk II dual floppy drive unit, single-sided, single-density, w/10' cable
M68KFD1102	EXORdisk III dual floppy drive unit, double-sided, single-density, w/10' cable
M68SFDU1102E	EXORdisk III dual drive expansion unit, w/28' expansion cable

3

VERSAmodule Universal Disk Controller





- VERSAbus compatible
- Host Computer commands serviced by on-board standard Intelligent Peripheral Controller based on the MC68120 IPC
- Separate Hard Disk Sequencer and Floppy Disk Controller controlled by IPC
- Provides control for up to two Storage Module Drive (SMD) interface compatible hard disk drives and up to 4 Motorola EXORdisk floppy disk drives
- 13.6 Mb to 512 Mb formatted hard disk capacity per controller
- Flexibility to define type and capacity of drive unit connected to each SMD interface port including SMD, CMD, MMD, LMD and 14" Winchester drives with fixed, removable and/or combination fixed/removable media
- Floppy Disk formatted capacity of 0.5 Mb for one EXORdisk II two-drive unit, or 2.0 Mb for two EXORdisk III two-drive units
- Shared RAM command channel and high-speed DMA data channel to VERSAbus
- Provides 24 × 16-bit word on-board data buffer
- Capable of addressing DMA transfers to any address in 16 Mb VERSAbus memory space
- Interrupt or ready status indicates completion of data transfer
- Supported by VERSAdos Operating System
- On-board Self-Test and Diagnostic Firmware

The Universal Disk Controller (UDC) adds large mass storage capability to any MC68000 VERSAbus compatible system and is fully supported by Motorola's VERSAdos Operating System. The UDC provides control for SMD interface compatible hard disk drives and optional Motorola EXORdisk floppy disk drives. Separate controller and device interface ports are provided for these devices. Specifications for the UDC are shown in Table 1. Table 2 lists the compatible drives handled by the UDC together with their storage capacities. The controller consists of two boards, the Disk Interface Module (DIM) and the Universal Intelligent Peripheral Controller (UIPC), with interconnecting 50 conductor flat ribbon cable. A functional block diagram of the UDC is illustrated in Figure 1.

DISK INTERFACE MODULE (DIM)

The DIM provides the physical drive interface and control for both the hard disk and the floppy disk drives. The board contains connectors for up to two SMD interface compatible drives and for up to four daisy-chained EXORdisk floppy disk drives.

The SMD ports may be individually configured so that different types of SMD interface compatible drives may be placed on each port in the same system. Industry Standard SMD interface cabling including radial data transfer cables, daisy-chain control cable and daisy-chain expansion cable are available from Motorola or may be purchased from any commercial cable supplier. Motorola EXORdisk drives come equipped with appropriate cabling for use with the UDC. VERSAmodule chassis rearpanel mountable cable assemblies are also available.

The DIM board has no direct VERSAbus interface and draws only power when plugged into a VERSAbus backplane. The interface to VERSAbus is through the UIPC board via the 50 conductor DMA Channel interface interconnect cable. Figure 2 illustrates UDC cabling when mounted in a VERSAbus compatible backplane.

Though the interconnect cable provided with the UDC is intended for use between DIM and UIPC boards located in adjacent VERSAbus card slots, the DIM contains mounting holes and power connectors for external mounting at customer provided cable distances of up to five feet from the UIPC board. Figure 3 illustrates UDC cabling when the DIM is mounted externally from the VERSAbus backplane.

The on-board floppy disk controller and hard disk sequencer are controlled from the MC68120 Intelligent Peripheral Controller based UIPC board via commands from the VERSAbus host computer operating system.

UNIVERSAL INTELLIGENT PERIPHERAL CONTROLLER (UIPC)

The UIPC consists of a Microcomputer Unit based on the MC68120 IPC, a DMA Channel Interface, a FIFO Buffer and a VERSAbus Interface.

The Microcomputer Unit includes the MC68120 IPC chip with internal 128 byte dual port RAM, 4K bytes of static RAM, four ROM/EPROM sockets with IPC Monitor and UDC device control application firmware installed, and a DMA FIFO Controller. The UDC device control firmware processes system "macro" commands for data transfer, status and device/UDC self-test. The 128 byte dual port RAM is used as a Command Channel Software Interface for entry of these commands to the UIPC. Entry is via VERSAbus from the operating system software. Once the operating system command is entered into the dual port RAM, the VERSAbus host computer may conduct other transactions on VERSAbus while the UIPC simultaneously carries out the details of the "macro" command via its on-board bus.

The operating system software communicates with the UIPC using a standard command channel protocol regardless of the device being controlled. The Command Channel Software Interface Reference Manual defines this standard protocol and the specific commands applicable to the UDC.

The DMA Channel Interface provides a physical device command and status path between the UIPC Microcomputer Unit and the DIM. The DMA Channel Interface together with the FIFO Buffer provide a path for data transfers between VERSAbus and the DIM under control of the MC68120 and the DMA FIFO Controller. During data transfer, the DMA FIFO Controller requests VERSAbus mastership whenever the FIFO Buffer needs to be filled or emptied to maintain pace with the synchronous disk transfer. The FIFO Controller holds bus mastership for approximately 8 microseconds before releasing it for use by other bus masters. Additional bus mastership requests are generated in the above manner until the data transfer is complete. Transfer Completion is signalled by the UIPC via an Interrupt to VERSAbus and/or by Ready Status indication in the VERSAbus Command Channel. The UIPC Microcomputer Unit generates and checks CRC information for hard and floppy disk during Write and Read Command data transfer, respectively. Automatic retry is performed on CRC error detection.

The FIFO time buffers synchronous disk data transfers to and from the asynchronous system bus. In the case of the UDC, the 32 byte UIPC FIFO buffer is augmented by a 16 byte FIFO on the DIM to provide the additional time buffering required for the 1.25M byte per second SMD interface.

Since the high performance UIPC VERSAbus Interface allows very high data rate transfers, the FIFO Buffer approach permits contiguous multi-sector disk transfer without incurring the system performance degrading effects inherent in slower systems requiring full sector buffering and discontiguous sector interleaving to avoid data overrun. The full sector buffering (256 bytes) required by slower system bus architectures means sector size DMA block transfers. The consequence of sector block transfers is system bus lockout of other bus masters for very long time durations relative to the short 8 μs, DMA cycle stealing bursts applicable to UDC FIFO buffering. In addition, the discontiguous sector interleaving required by slower systems bus architectures means a very long time duration to complete a multi-sector disk transfer. The consequence of discontiguous sector interleaving is slower system performance than for the contiguous multi-sector disk transfers permitted by the UDC.

Finally, the VERSAbus Interface provides standard VERSAbus data transfer, interrupter and bus requester functions as defined in the VERSAbus Specification (M68KVBS). The base address, interrupt vector number, interrupt request priority level, and the bus request priority level are selectable by on-board strap options permitting more than one UDC per system. VERSAbus Interface speed and FIFO Buffer control readily permit the use of two UDC's per system using RAM boards of up to 450 ns cycle time.

FIGURE 1 — Universal Disk Controller (UDC)

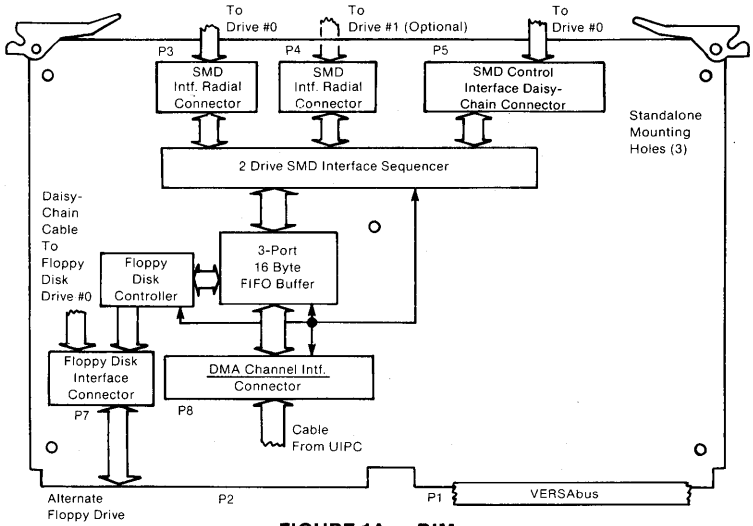


FIGURE 1A — DIM

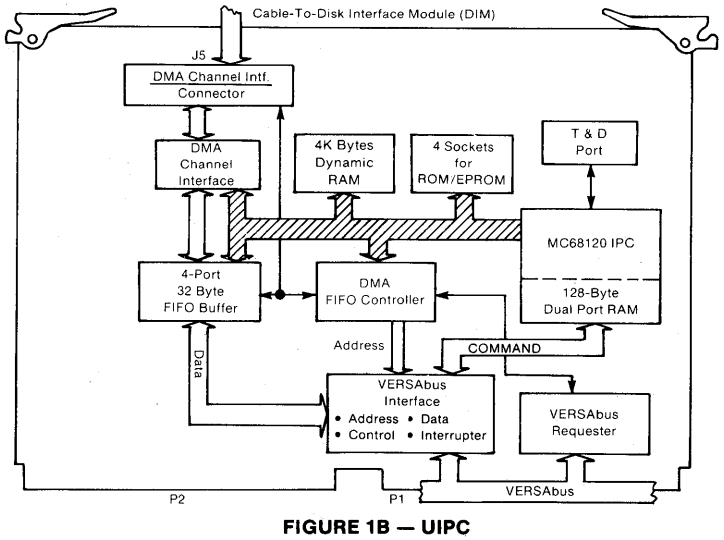


FIGURE 1B — UIPC

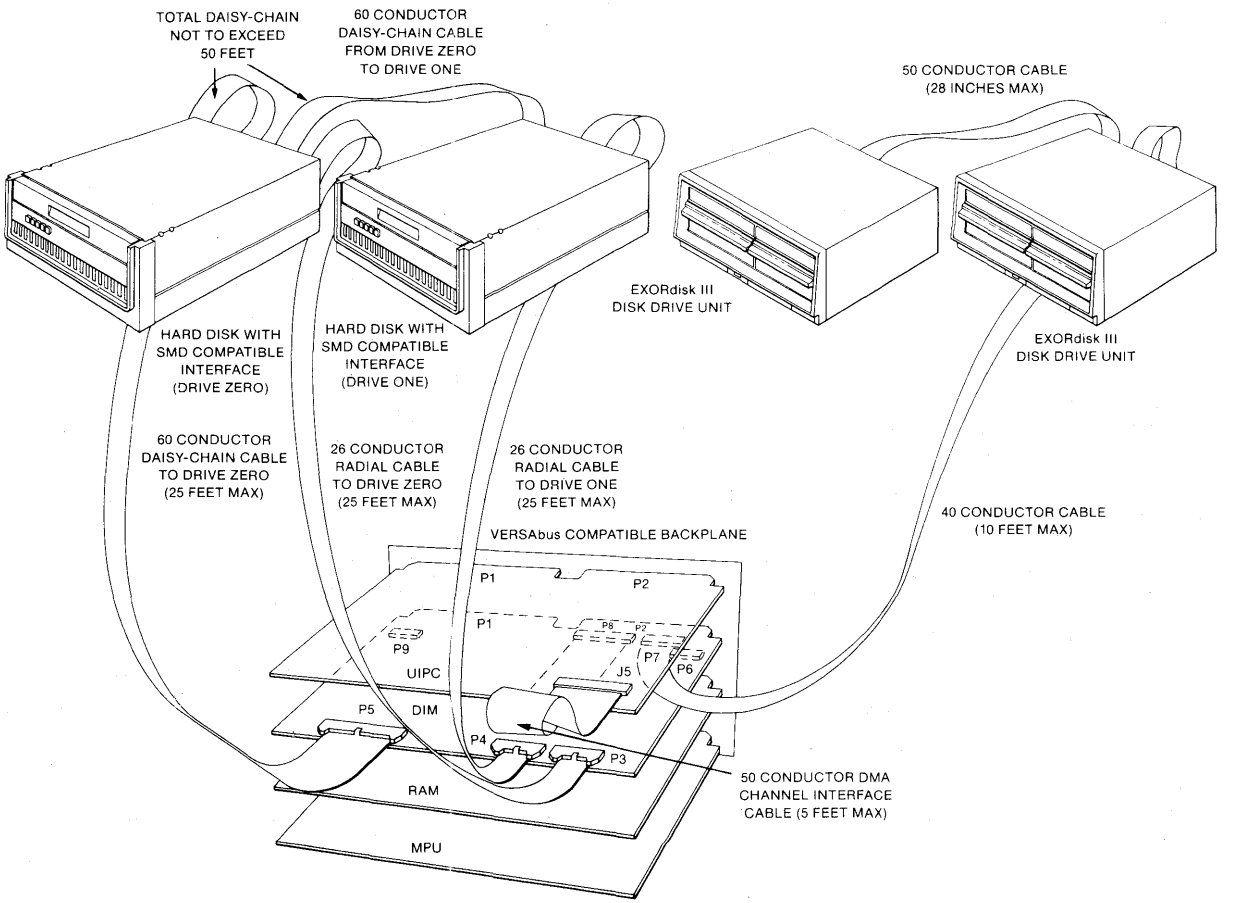


FIGURE 2 — Typical Installation of the Universal Disk Controller

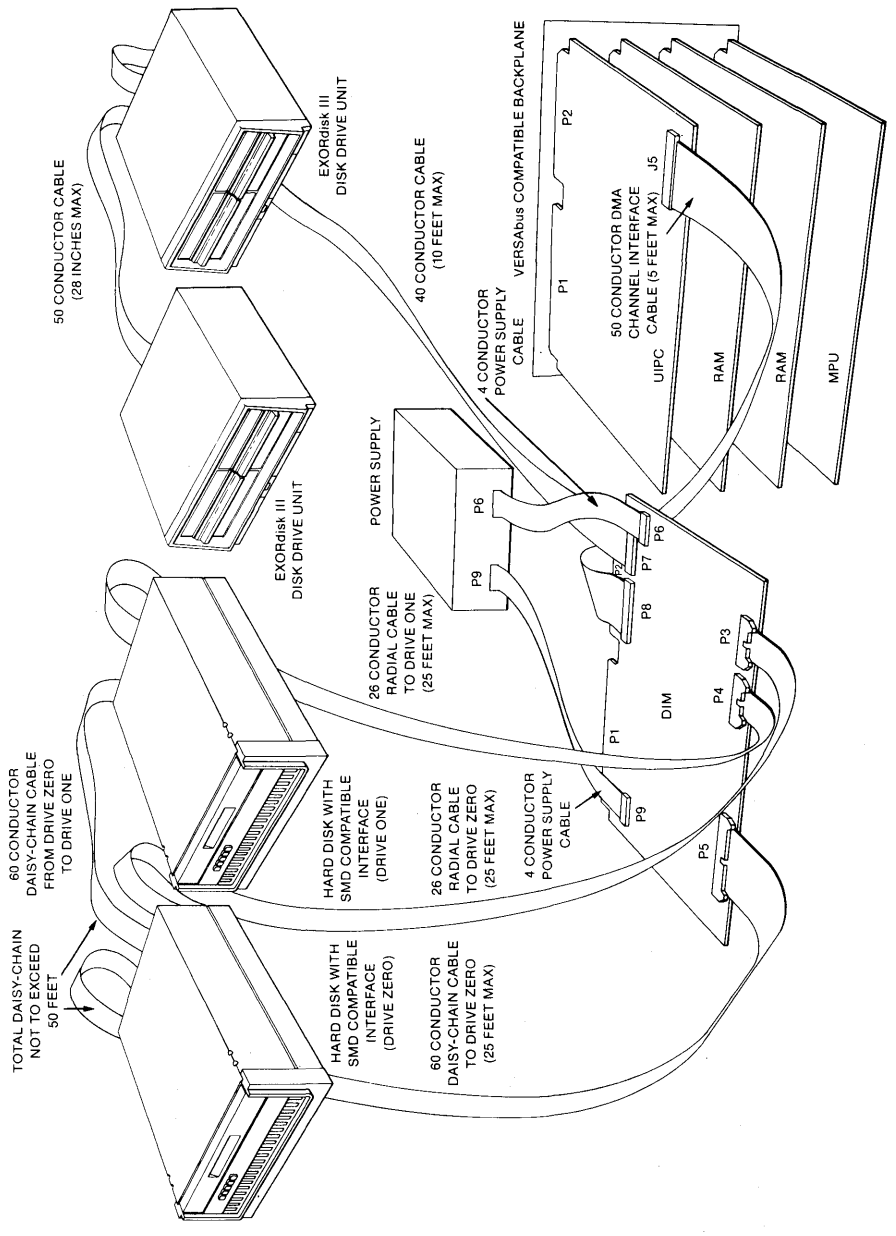


FIGURE 3 — Typical Installation of the Universal Disk Controller Utilizing Standalone DIM

TABLE 1 — Specifications

Characteristics	Specifications
Compatibility	
Computer	Any VERSAmodule/VERSAbus Monoboard Microcomputer
Hard Disk Drive	Any of the SMD interface compatible disk drives listed in Table 2
Floppy Disk Drive	Motorola EXORdisk II or III
Media Format	
Hard Disk	Motorola EXORmacs VERSAdos Compatible
Floppy Disk	Motorola EXORmacs VERSAdos/EXORciser MDOS Compatible
Drive Data Rate	
Hard Disk	10 Mbits/sec, max
Floppy Disk	250 kbits/sec, max
Drive Error Detection	CRC Check
Sectors/Track	
Hard Disk	64
Floppy Disk	26
Sector Size	
Hard Disk	256 bytes (VERSAbus commands allow transfers in increments of 256 byte blocks)
Floppy Disk	128 bytes (VERSAbus commands allow transfers in increments of 128 or 256 byte blocks)
VERSAbus Command Channel	Permits processing of one command* at a time for Status or for Data Transfers in one or more full block increments up to 65,535 blocks (16 Mb), max.
VERSAbus Data Channel	VERSAbus Data Transfers performed via DMA at word rates of up to 4 Mwords (8 Mb)/sec., max., with actual rate determined by cycle time of the addressed VERSAbus memory board. With the M68KVM10 128K RAM module the maximum transfer rate is approximately 1.5 Mwords/sec.
FIFO Data Buffer	48 byte FIFO used to time buffer synchronous disk data to asynchronous system bus.
VERSAbus Master Retention Time	During DMA, Bus Mastership retained for approximately 8 μs, max., per data transfer burst before bus released to other masters
Power Requirements (Typical)	
UIPC	+5 Vdc ±5% @ 7.0 A +12 Vdc ±5% @ 50 mA -12 Vdc ±5% @ 50 mA
DIM	+5 Vdc ±5% @ 5.0 A -12 Vdc ±5% @ 550 mA

*See MC68000/IPC Command Channel Software Interface Reference Manual M68KIPCS for command details.

TABLE 1 — (Continued)

Characteristics	Specifications
Temperature Operating Storage	 0° to +65° C -40° to +85° C
Relative Humidity	0 to 90% (non-condensing)
Dimensions Length × width × height (including components)	14.50 × 9.25 × 0.6 inches (each PCB)

TABLE 2 — Compatible Disk Drives

Model	Type*	Formatted Capacity	Removable/Fixed
CDC 9762	SMD	67.4 Mb	Removable
CDC 9764		128 Mb	
CDC 9766		256 Mb	
CDC 9448-32	CMD	13.5/13.5 Mb	Removable/Fixed
CDC 9448-64		13.5/40.5 Mb	
CDC 9448-96		13.5/67.4 Mb	
CDC 9730-80	MMD	67.4 Mb	Fixed
CDC 9730-160		135 Mb	
CDC 9455	LMD	6.8/6.8 Mb	Removable/Fixed
CDC		13.5/13.5 Mb	
PRIAM 3350	14" Winchester	27.6 Mb	Fixed
PRIAM 8650		50.3 Mb	
EXORdisk II	8" Floppy	512 kb dual drive, single-sided, single density unit.	
EXORdisk III	8" Floppy	1 Mb dual drive, double-sided, single density unit.	

* SMD = Storage Module Drive
CMD = Cartridge Module Drive
MMD = Mini Module Drive
LMD = Lark Module Drive

NOTE: Access to the configuration table in the supplied UDC firmware that corresponds to a specific disk drive type and model is provided by jumper selection on the DIM. Space is reserved in the firmware sufficient for four additional user-supplied configuration tables to accommodate other user disk drives.

Ordering Information

Part Number	Description
M68KVM21	VERSAmodule Universal Disk Controller (UDC). This controller is a two board set with interconnecting cable. The two boards are the Universal Intelligent Peripheral Controller (UIPC) and the Disk Interface Module (DIM). The UDC may control up to two industry standard SMD interface compatible hard disk drives and up to two EXORdisk II floppy disk drives (one EXORdisk II unit) or up to four EXORdisk III floppy disk drives (two EXORdisk III units). Includes User's Manual and M68000/IPC Command Channel Software Interface Reference Manual.
Related Documentation	
M68KVM21/D2	UDC User's Manual
M68KIPCS/D2	M68000/IPC Command Channel Software Interface Reference Manual
M68KVBS	VERSAbus Specification

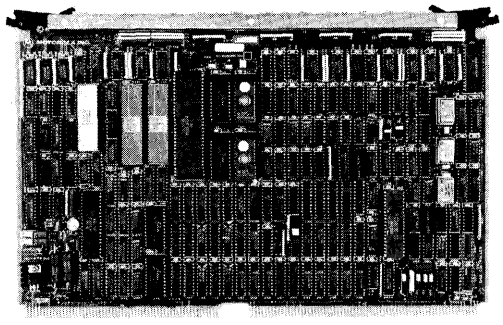
Options

Part Number	Description
M68KVMSMDR	SMD Radial Data Cable, 15'
M68KVMSMDC	SMD Daisy-Chain Control Cable, 15'
M68KVMSMDE	SMD Daisy-Chain Expansion Cable, 6'8"
M68KVMCHD1	Cable Assembly routing UDC SMD interface from DIM to VERSAmodule Chassis rearpanel for external chassis disconnect. Includes two 34" Radial Data Cables and one 42" Daisy-Chain Control Cable all of which are rearpanel mountable.
M68KVMCFD1	Cable Assembly, routing UDC floppy disk interface from the VERSAbus backplane to VERSAmodule Chassis rearpanel for external chassis disconnect. Includes one 25" rearpanel mountable floppy disk cable.
M68KFD1100	EXORdisk II dual floppy drive unit, single-sided, single-density, w/10' cable.
M68KFD1102	EXORdisk III dual floppy drive unit, double-sided, single density, w/10' cable.
M68SFDU1102E	EXORdisk III dual drive expansion unit, w/28" expansion cable.

M68KVM22

3

VERSAmodule Disk Controller



- Single VERSAmodule Board.
- 32-bit Addressing.
- MC68000-Based with On-Board Read Only Memory (ROM) and Random Access Memory (RAM).
- A High-Level Command Structure Includes Multisector/Track Transfers with Implied Seeks.
- Command Chaining Feature Facilitates Transfer of Non-contiguous Blocks using a Single Command Structure.
- Utilizes Direct Memory Access (DMA) for Data Transfers to/from Global Memory (off-board RAM).
- Interleaved Commands Provide Overlapped Seeks and High Throughput.
- Supports Four SMD-Compatible Disk Drives:
  — 256, 512, or 1024 Bytes/Sector
- Supports SA800-Compatible 8-inch Floppy Disk Drives or SA400-Compatible 5¼-inch Floppy Disk Drives in any Combination of Four:
  — 128, 256, 512, or 1024 Bytes/Sector
  — Frequency Modulation (FM)/Modified Frequency Modulation (MFM) recording.
- 32-bit Error Correction Code (ECC) Allows Transparent Correction of 11-Bit Burst Errors on SMD Disks.
- 16-bit Cyclic Redundancy Check (CRC) on Floppy Disks.
- Multisector, 4K byte First In First Out (FIFO) Data Buffer.
- Power-up/Reset Self-test.
- Commands are Passed Through Global RAM.
- Download Command Allows User to Customize Firmware.
- 0°C–70°C Operating Temperature Range.

The Disk Controller is an intelligent interface used for adding mass storage capacity to a VERSAbus system. It provides high performance DMA data transfer channels between system memory and hard disk drives and/or floppy disk drives. The module is applied in environments having intensive real-time disk I/O or multiprocessing structures to reduce VERSAbus traffic and increase system throughput. Figure 1 is a functional block diagram of VM22.

HOST/CONTROLLER COMMUNICATION

An intelligent module, VM22 offers the user a high level, easy to program interface. Instructions in 8K bytes of ROM are executed on the MC68000 Microprocessor CPU to provide the macro I/O activities requested by the user program.

General control of data transfer operations is obtained using six write and one read registers to send command information to the module and to receive execution status information from the module. The registers are used in conjunction with global memory areas in which formatted, 24-byte packets containing command and status details are passed between host and module. The VM22 registers are listed in Table 2.

The host transmits a high level command to the VM22 by writing a command packet in global memory at a specified address, identifying the starting address of the packet by loading the command address pointer registers on the VM22, and finally queuing the command by setting a bit in the control register.

Upon completion of the command, the VM22 writes the resulting status packet in global memory immediately following the command packet, then interrupts the host at the interrupt number and level defined in the command packet.

SELF TEST

A power-up/reset self-test is performed when power is applied to the VM22, and during a reset condition by a software reset from the host system.

VERSAbus INTERFACE —

The VM22 performs the function of master (DMA transfers), slave (R/W control registers) and interrupter (command completion) on the VERSAbus.

As a VERSAbus master, the VM22 supports a 32-bit address bus and a 16-bit data bus. The R/W control registers (slave) are 8-bit wide and are located in the short 16-bit I/O address space.

BASE ADDRESS

The VM22 has a header for jumper assignment of a base address. A base address can be set on any 16 byte boundary throughout the VERSAbus short I/O address space.

DISK INTERFACES —

The four SMD drives interface the VM22 via five connectors located at the top of the module, providing for star clustering (four 26-pin connectors) or daisy chain (one 60-pin connector) connection.

Floppy connections are made through P2 of the VERSAbus. There are two possible connections; a 50-pin (8-inch floppy) and a 34-pin (5¼-inch floppy).

Table 1 lists the compatible SMD drives with storage capacities that can be handled by the VM22.

Figure 1 — M68KVM22 Block Diagram

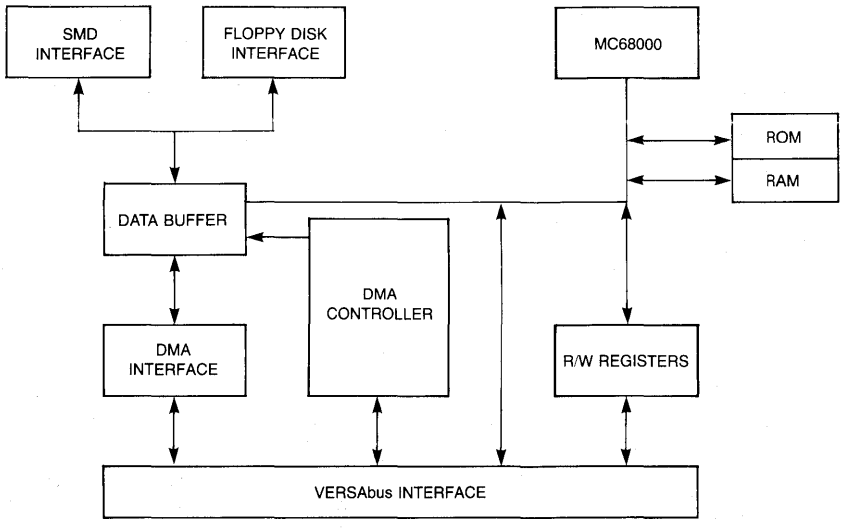


Table 1 — SMD Compatible Disk Drive Parameters

Manuf.	Model	Media	Sectors/ Track	Data Heads	Cylinders	Embedded Servo	Unformatted Capacity (Mbytes)
Amcodyne	7110	Removable Fixed	64 64	2 2	644 644	Yes Yes	26.6 26.6
Ampex	DFR-932	Removable Fixed	64 64	1 1	823 823	No No	16.6 16.6
Ampex	DFR-948	Removable Fixed	64 64	1 3	823 823	No No	16.6 49.8
Ampex	DFR-996	Removable Fixed	64 64	1 5	823 823	No No	16.6 82.9
Ampex	330	Fixed	64	16	1024	No	330.3
Ampex	165	Fixed	64	10	823	No	165.9
CDC (1)	9448-32	Removable Fixed	64 64	1 1	823 823	No No	16.6 16.6
CDC	9448-64	Removable Fixed	64 64	1 3	823 823	No No	16.6 49.8
CDC	9448-96	Removable Fixed	64 64	1 5	823 823	No No	16.6 82.9
CDC	9455	Removable Fixed	64 64	2 2	206 206	Yes Yes	8.3 8.3
CDC	9457	Removable Fixed	64 64	2 2	624 624	Yes Yes	25.7 25.7
CDC	9762	Removable	64	5	823	No	82.9
CDC	9766	Removable	64	19	823	No	315.2
CDC	9730-80	Fixed	64	5	823	No	82.9
CDC	9730-160	Fixed	64	10	823	No	165.9
CDC	9710	Removable	64	5	823	No	82.9
CDC	9715	Fixed	64	10	823	No	165.9
CDC	9775	Fixed	64	40	823	No	679.8
Priam	3350	Fixed	64	3	561	No	33.9
Priam	6650	Fixed	64	3	1024	No	61.9
Priam	15450	Fixed	64	7	1024	No	144.5
Priam	803	Fixed	64	5	850	No	85.7
Priam	804	Fixed	64	5	1024	No	105.8

Note: (1) CDC = Control Data Corporation

DISK BUFFER MANAGEMENT —

The VM22 contains a 4K byte buffer that allows the host system to accept data at rates independent of the disk data transfer rate. The data buffer is organized as a "pool" of one sector buffers. During normal operation the local processor allocates sector buffers to the VERSAbus DMA controller and an SMD or floppy disk controller chip as required to transfer data.

During a disk read operation, for example, the local processor allocates the first buffer to a disk controller. When the buffer is filled, the local processor allocates the second buffer to the disk controller and the first buffer to the DMA controller. When data in the first buffer is transferred to global RAM by the DMA controller, the buffer is returned to the "pool" of empty buffers. During a disk write operation the buffers are "filled" by the DMA controller and "emptied" by the disk controller.

SYSTEM PERFORMANCE —

To obtain maximum performance during multi-sector data transfers, VERSAbus request levels of the system must be properly set to avoid data buffer underflow or overflow conditions. (The VM22 recovers from underflow and overflow by retries, causing degradation in system performance.)

Bus retention time for the DMA controller is controlled by time-on and time-off bus timers. The VM22 is factory set for 16 microseconds time-on and 4 microseconds time-off, allowing other system masters to the access bus during disk operations. Other options are user configurable.

VM22 COMMAND STRUCTURE —

The command structure necessary for communication with the VM22 consists of the on-board, memory mapped read/write registers, commands which are reorganized by the VM22 and command formats a packet through which commands are transferred to the VM22.

The on-board read/write registers are used to initialize the controller, queue commands, request special functions (i.e., perform self-test), and report status to the host. Table 2 defines the on-board read/write registers.

Table 2 — Read/Write Registers

	Register	Byte Offset From Base Address	Function
Write	WR0	1	Control Register
	WR1	3	Reserved
	WR2	5	Reserved
	WR3	7	Command Access Address Modifier
	WR4	9	Command Address Pointer, MSB
	WR5	11	Command Address Pointer, Byte 2
	WR6	13	Command Address Pointer, Byte 1
	WR7	15	Command Address Pointer, LSB
Read	RR0	1	Controller Status
	RR1	3	Reserved
	RR2	5	Reserved
	RR3	7	Reserved
	RR4	9	Reserved
	RR5	11	Reserved
	RR6	13	Reserved
	RR7	15	Reserved

The commands recognized by the VM22 define macro disk operations. These are expanded by the local processor into a set of micro commands that execute the required function. Table 3 defines the macro commands supported by the VM22.

Table 3 — M68KVM22 Commands

Command Opcode	Packet Type	Function
$00	1	Check Unit Status
$01	1	Recalibrate
$02	2	Format Unit
$03	2	Format Track
$04	0	Read Sectors
$05	0	Write Sectors
$06	0	Check Sectors
$07	2	Seek Track
$08	3	Configure Unit
$09	4	Download
$0A	1	Return Self-test Status

Commands are transferred through global memory in the form of 24-byte command packets. Packets for the same logical disk unit can be chained (i.e., read sector, read sector, read sector). The status packet of 20 bits follows the last command packet in the chain.

Command packets are tailored to the type of command they specify. The general form of a command packet is given in Table 4.

There are two formats for status packets; one issued to report the completion of a command packet, and the other to report the results of self-test.

Table 4 — Command Packet Format

	Byte 0	Byte 1	Byte 2	Byte 3
0	Command	Command Opcode	Logical Unit No.	Addr Mod
4	INT Level	INT Number	Retry Count	Control
8 12 16 20	Command Specific Information			

Table 5 — M68KVM22 Specifications

Characteristics	Specifications
Compatibility (See Note) Computer Hard disk drive Floppy disk drive	 VERSAbus compatible microcomputer system SMD compatible disk drive SA800 compatible 8-inch floppy disk drive or SA400 compatible 5¼-inch disk drive
Media format Hard disk Floppy disk	 NEC UPD 7261 compatible Motorola or IBM compatible
Disk drive data rate Hard disk Floppy disk	 12M bits/second maximum 500K bits/second maximum
Error detection/correction Hard disk Floppy disk	 32-bit ECC allows correction of 11-bit burst errors 16-bit CRC for error detection
Sector size Hard disk Floppy disk (FM) Floppy disk (MFM)	 256, 512, or 1024 bytes/sector (hard sector) 128, 256, 512, or 1024 bytes/sector (soft sector) 256, 512, or 1024 bytes/sector (soft sector)
Data buffer	4K-byte FIFO type multiple sector buffer
VERSAbus DMA transfer rate	3M-bytes/second (with M68KVM10-3 128K-byte Dynamic RAM Module)
Power requirements	+5.0 Vdc ±5% @ 8.0 A (typical), 10 A (max) −12 Vdc ±10% @ 600 mA (typical), 750 mA (max)
Slave Mode Bus Characteristics Address Data Parity Interrupter	 A16 D8 No address bus or data bus parity Level and vector software programmable

Table 5 — M68KVM22 Specifications (continued)

Characteristics	Specifications
Master Mode Bus Characteristics	
Address	A32
Data	D16
Parity	No address bus or data bus parity
Requester	Release when done
Temperature	
Operating	0° to 70°C
Storage	−40° to 85°C
Relative humidity	5% to 95% (non-condensing)
Dimensions	
Height	9.25 in (23.49 cm)
Width	14.5 in (36.83 cm)
Thickness	0.6 in (1.52 cm)

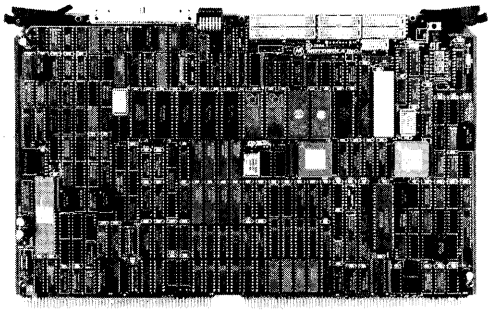
Note:
- The SMD disk format of the VM22 is not compatible with that of the M68KVM21 universal disk controller for the following reasons:
  1. The VM21 uses a 3-byte ID header; the VM22 uses five bytes.
  2. The VM21 uses a 2-byte CRC: the VM22 uses a 4-byte ECC.
- The VM22 also requires a different software driver than the VM21.
- Attempts to read or write data from a disk formatted by the VM21 will result in access errors but will not disturb the data on the disk.
- The VM22 will support the floppy disk format used by the VM21 as well as the IBM standard dual-density format.

Ordering Information

Part Number	Description
M68KVM22	VERSAmodule Disk Controller, includes User's Manual
Documentation	
M68KVM22/D1	VERSAmodule Disk Controller User's Manual
M68KIPCS/D1	M68000/IPC Command Channel Software Interface Reference Manual
M68KVBS/D4	VERSAbus Specification Manual

M68KVM23

VERSAmodule Disk Controller



- Single VERSAmodule Board
- MC68010 Based Intelligent Peripheral Controller
- Supports 24- or 32-bit DMA Addressing
- Supports Four ST506/ST412 Compatible 5¼-inch Winchester Disks
- Supports Four SA400 Compatible 5¼-inch Floppy Disks
- Supports One QIC-02 Compatible ¼-inch Streaming Tape Drive
- Supports One Centronics Compatible Printer
- Utilizes Direct Memory Access (DMA) for Device to Local Buffer and Local Buffer to Global Memory Transfers
- 32Kb Dual Ported Static RAM for Data Buffering and Local MPU
- 8 Byte FIFO for VERSAbus Transfers
- 32-bit Error Correction Code (ECC) Allows Transparent Correction of 11-bit Burst Errors on Winchester Disk
- 16-bit Cyclic Redundancy Check (CRC) on Floppy Disks and Streaming Tape
- 7 Segment LED Located at Card Edge for Diagnostic Messages
- High-level Command Packets for Efficient Operating System Support
- Buffered Pipe Command Protocol Allows Multiple Hosts to Queue Commands Without Interlock
- Intelligent Disk Manager Schedules Transfers to Optimize Head Movement on Each Drive and Interleaves Operations Between Drives
- Multitasking Kernel for Concurrent Execution of Multiple Device Managers
- Command Chaining for Simplifying Data Transfers in a Paged Environment
- Onboard Multi-track Data Caching for Efficient Task Swapping Using Onboard Multi-track Data Buffers
- High-speed Image Backup from Disk to Tape with No Host Intervention
- Transparent Bad Block Handling with No Added Seeks

The M68KVM23 (VM23) is an intelligent interface used for adding mass storage capacity to a VERSAbus system. It provides high-performance DMA data transfer channels between system memory and Winchester disks, floppy disks, a streaming tape drive, and/or a printer. The module is applied in environments having intensive realtime disk I/O or multiprocessing structures to reduce VERSAbus traffic and increase system throughput. Figure 1 is a functional block diagram of the VM23.

INTELLIGENT PERIPHERAL CONTROLLER

An Intelligent Peripheral Controller, the VM23 is based on the MC68010 MPU. Independent DMA Controllers allow concurrent device to buffer and buffer to VERSAbus transfers. A MC68440 Dual Channel DMA Controller is dedicated for disk and streaming tape transfers. An 8-byte FIFO and separate DMA controller provide efficient back-to-back transfers over the VERSAbus. The VM23's local bus and the VERSAbus are interconnected through 32Kb of dual-ported RAM.

The MC68010 interprets the high-level commands passed to it through global memory and the Control/Status Registers. Once it has disassembled these commands, the MPU supervises the activity of the onboard device and DMA controllers. The ROM-resident multitasking kernel supervises concurrent execution of I/O managers and manages a pool of data buffers that are shared by all devices.

The hard disk manager performs implied seeks, overlaps operations between drives, and schedules transfers to optimize head movement. The streaming tape manager buffers data to ensure tape transfers are made at the full speed of the drive.

The power-up sequence performs a diagnostic self-check of the VM23 displaying diagnostic messages in a 7-segment LED.

The command packet protocol allows one or more hosts to buffer commands to the VM23 without requiring an interlock mechanism. The packet format is shown in Figures 2 and 3.

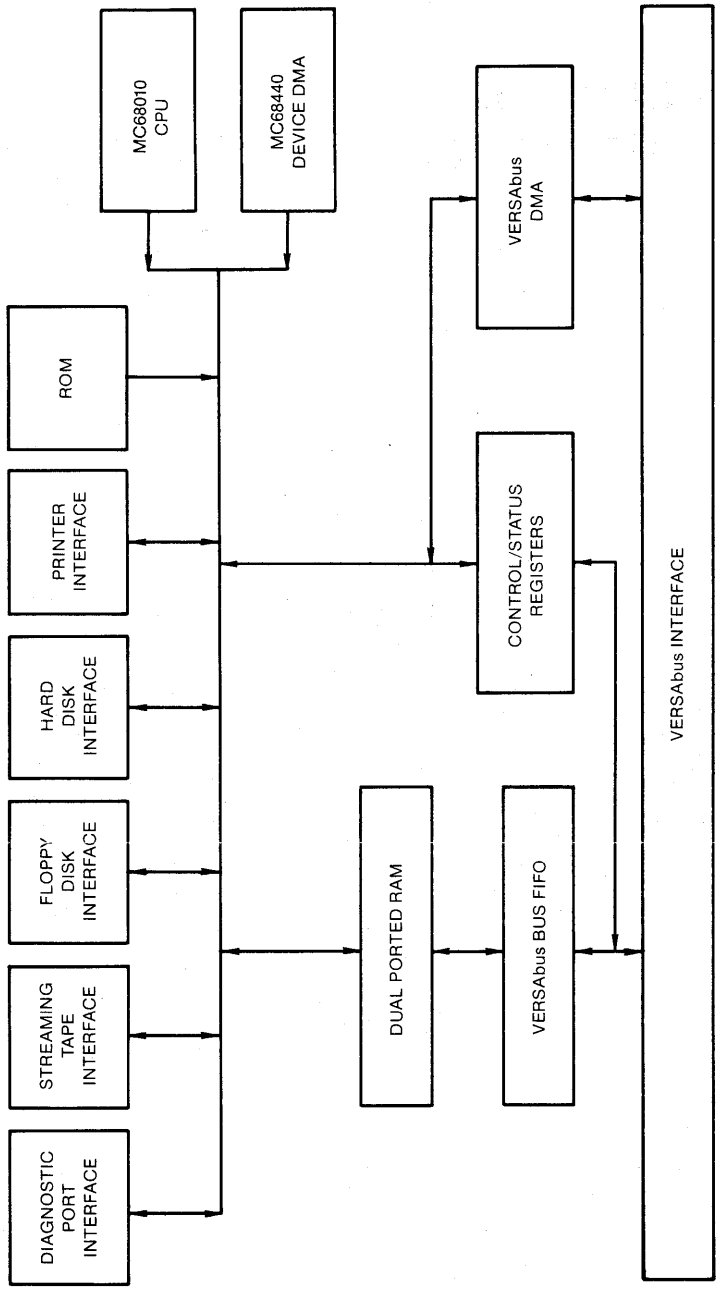


FIGURE 1 — M68KVM23 Block Diagram

FIGURE 2 — Channel Packet Envelope Format

Bit -->	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Addr																
$00	link msw															
$02	link lsw															
$04	packet pointer msw															
$06	packet pointer lsw															
$08	valid flag								reserved							
$0A	reserved															

FIGURE 3 — Channel Packet Format

Bit -->	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Addr																
$00	eyecatcher msw															
$02	eyecatcher lsw															
$04	command queue number								status queue number							
$06	command/status packet number msw															
$08	command/status packet number lsw															
$0A	cmd group					subcommand										
$0C	device type								unit number							
$0E																
:	opcode dependent parameters															
$2E																
$30	status word															
$32																
:	opcode dependent status															
$3E																

VERSAbus INTERFACE

Base Address

The VM23 has a header for base address selection of the Control/Status Registers that are used to control the module. A base address can be set on any 256 byte boundary of the VERSAbus Short I/O Space.

Direct Memory Access

The VM23 request mastership of the VERSAbus through a programmable Release-On-Request (ROR) or Release-When-Done (RWD) requester. As the VERSAbus master, the VM23 is program-selectable to provide a 24- or 32-bit address, and to provide 16 or 32 bits of data. The VM23 utilizes an 8 byte (2 × 32) FIFO for back-to-back VERSAbus cycles.

CONTROL/STATUS REGISTERS

Operating systems communicate with the VM23 through a standard set of Control and Status Registers. These registers are used to initialize the controller, queue commands, request special functions, and report status to the host. Figure 4 provides the offset address from the selected base, and the bit definitions for the individual registers. Refer to the M68KVM23 User's Guide for a detailed definition of these registers and programming information.

FIGURE 4 — Control Status Registers

Base Address Offset	Even (D15 … D0)	Odd (D15 … D0)
0	A31 … IPC Addr	Register
2	IPC Addr	Register … A0
4	IPC AM Register	Unused
6	IPC Control Register	Reserved
8	IPC Status Register	Reserved
A	IPC MDB/ID Register	Reserved
C	IPC Abort Vector Register	Unused
E	Unused	Unused

DEVICE INTERFACES

Four 5¼-inch Winchester disk drives are connected to the VM23 by four radial data cables (26-pin connectors) and one daisy-chain cable (34-pin connector) located at the top of the module.

Floppy drives are interfaced by one daisy-chain cable (34-pin connector) also located at the top of the module.

The remaining 50-pin connector is used to interface the streaming tape drive.

The printer is interfaced through a 36-pin cable to the backplane P2 card edge connector.

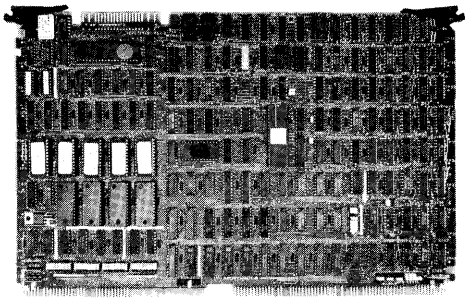
Mechanical and Environmental Specifications

Characteristics	Specifications
Compatibility	
Computer	VERSAbus compatible microcomputer system
Hard disk drive	ST506/ST412 compatible 5¼" disk drive
Floppy disk drive	SA400 compatible 5¼" disk drive
Streaming tape drive	QIC-02 compatible ¼" tape drive
Media format	
Hard disk	NEC UPD 7261 compatible
Floppy disk	Motorola or IBM compatible
Streaming tape	QIC-24 compatible
Disk transfer rates	
Hard disk	5 megabits/second maximum
Floppy disk	250 kilobits/second maximum
Streaming tape	90 kilobytes/second maximum
Error detection/correction	
Hard disk	32-bit ECC allows correction of 11-bit burst errors
Floppy disk	16-bit CRC for error detection
Streaming tape	16-bit CRC for error detection
Data buffer	16Kb multiple sector buffer
VERSAbus DMA transfer rate	3.5 megabytes/second with 16-bit slave 7 megabytes/second with 32-bit slave (actual performance depends on slave cycle times)
Power requirements	+5.0 Vdc +5% @ TBD Amps +12 Vdc ±10% @ TBD Amps
VERSAbus Slave Mode	
Address	A16, VERSAbus short I/O
Data	D16
Interrupter	Level and vector software programmable
VERSAbus Master Mode	
Address	A24 or A32
Data	D16 or D32
Requester	Release-On-Request or Release-When-Done
Temperature	
Operating	0° C to 55° C
Storage	-40° to 85° C
Relative Humidity	5% to 95% (non-condensing)
Dimensions	
Height	9.25 inch (23.49 cm)
Width	14.5 inch (36.83 cm)
Thickness	0.6 inch (1.52 cm)

Ordering Information

Part Number	Description
M68KVM23	VERSAmodule Disk Controller, includes User's Guide
MVMCH3-123	VM23 Distribution Panel with cables for use with MVMCH3, 8-slot chassis
M68KVM23/D1	VERSAmodule Disk Controller User's Guide
M68KVBS/D4	VERSAbus Specification Manual

VERSAmodule Multi-Channel Communications Module



- VERSAbus compatible
- Interfaces four asynchronous serial data terminal/modems to VERSAbus
- Interfaces one asynchronous parallel data hard copy printer to VERSAbus
- Terminal/Modem I/O — RS-232C standard
- Printer I/O — Centronics interface
- Independent software controlled baud rate selection for each RS-232C I/O port
- Internally generated 50 to 9.6K baud rate
- RS-232C data transmission — up to 50 feet at 9.6K baud over unbalanced lines
- MC6801 (MCU) based Intelligent Peripheral Controller (IPC) circuitry with firmware control and test and diagnostic programs
- MC2661 Enhanced Programmable Communications Interface (EPCI) based terminal/modem interface circuitry
- MC6821 Peripheral Interface Adapter (PIA) based printer circuitry
- 0° C-70° C Operating Temperature Range.

The Multi-Channel Communications Module (MCCM) can expand the user's system by interfacing four asynchronous serial terminal/modem devices and one asynchronous parallel printer to the VERSAbus. A simplified functional block diagram of the MCCM is illustrated in Figure 1 and the specifications are identified in Table 1.

This module is composed of the following functional circuits—Intelligent Peripheral Controller, Printer Interface and a Terminal/Modem Interface. Overall control of the MCCM, including VERSAbus communication and data transfer operations, is provided by the MC6801 MCU-based IPC circuitry. The MC6821 PIA-based printer interface circuitry provides communication and data transfer operations between the IPC and an asynchronous parallel printer. Table 2 provides pin assignments and signal descriptions for the printer interface connector, P3.

The MC2661 EPCI-based terminal/modem interface circuitry provides communication and data transfer operations between the IPC and four external RS-232C terminal/modem devices. Table 3 provides pin assignments and signal descriptions for the connector for this interface, P2.

The MCCM is installed directly into the user's system chassis card rack and connected to the chassis backplane (VERSAbus). Printer interfacing is accomplished by cable interconnections between the MCCM edge connector and associated printer equipment. Terminal/modem interfacing is accomplished by cable interconnections between the MCCM edge connector and associated RS-232C terminal/modem equipment. The rear panel mountable optional cable assembly (M68KVM232CP) may be used with the Motorola VERSAmodule Chassis (see installation diagrams, Figure 2 and 3).

The MCCM also has the following selection capabilities:

- Base Address Selection — The MCCM base address may be selected
- Interrupt Level Selection — Interrupt (priority) level numbers may be selected
- Terminal/Modem Interface Selection — Allows the user to properly interface the four RS-232C I/O ports to external terminal/modem equipment
- Interrupt Vector Selection — Up to 256 interrupt vector numbers may be selected
- Bus Arbitration Level Selection — The MCCM bus arbitration level is selected and provides a bypass operation for unused Bus Grant (BG) signals
- Reset Signal Disabling.

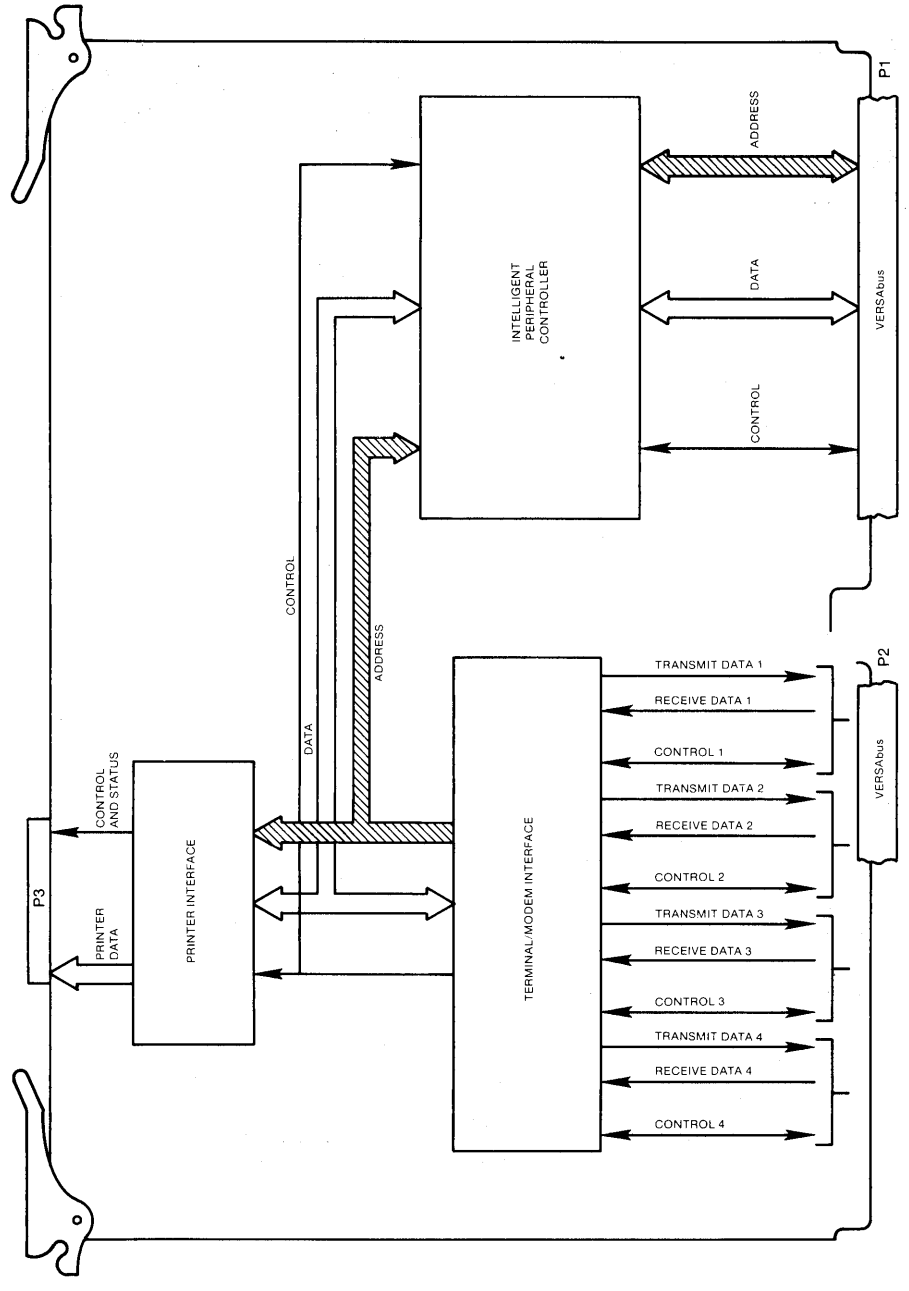


FIGURE 1 — MCCM Functional Block Diagram

TABLE 1 — MCCM Specifications

Characteristics	Specifications
Power Requirements	+5 Vdc, ± 5% @ 4 A +12 Vdc, ± 5% @ 250 mA −12 Vdc, ± 5% @ 200 mA
Frequency Internal External	 5.0688 MHz @ 0.05% 16 MHz system clock
VERSAbus I/O (P1) Signals Transfer Type	 VERSAbus compatible Parallel, asynchronous
Terminal/Modem I/O (P2) Signals Transfer Type	 RS-232C compatible Serial, asynchronous, 50-9.6K baud
Printer I/O (P3) Signals Transfer Type	 "Centronics" compatible Parallel, asynchronous
Temperature Operating Storage	 0° to +65°C −40° to 85°C
Relative Humidity	0 to 90% (non-condensing)
Dimensions Height x Length x Width	 9.25 x 14.50 x 0.6 inches (32.5 x 36.8 x 1.5 cm)

TABLE 2 — Pin Signal Assignments, Edge Connector P3

Connector Pin	Signal Mnemonic	Signal Name and Description
All even pins	GND	Ground
3, 7, 9, 11, 13, 15, 41, 45, 49	—	Not applicable to MCCM
1	IP*	INPUT PRIME—Output for clearing and initializing printer logic.
5	FAULT*	FAULT—Input indicating printer fault condition.
19	BUSY	BUSY—Input indicating printer cannot receive data.
21	PE	PRINTER EMPTY—Input indicating printer out of paper.
23	SEL	SELECT—Input indicating printer is selected.

TABLE 2 — Pin Signal Assignments, Edge Connector P3 (continued)

Connector Pin	Signal Mnemonic	Signal Name and Description
25 27 29 31 33 35 37 39	PD8 PD7 PD6 PD5 PD4 PD3 PD2 PD1	PD1-PD8 Lines over which data is transferred from MCCM to printer.
43	DS*	DATA STROBE—Output clock pulse (>1.0 μs)
47	ACK*	ACKNOWLEDGE—Input indicating receipt of data or end of functional operation.

NOTE: Signals descriptions are with respect to MCCM.

TABLE 3 — Pin Signal Assignments, Edge Connector P2

Connector Pin	Signal Mnemonic	Signal Name and Description
1-6	GND	Frame ground connection between MCCM and RS-232C devices.
7-10	+5 V	+5 Vdc POWER—Used by the MCCM logic circuitry.
11, 12	+12 V	+12 Vdc POWER—Used by the MCCM logic and interface circuitry.
15, 16	-12 V	-12 Vdc POWER—Used by the MCCM logic and interface circuitry.
54(1), 61(2), 22(3), 29(4)	SIG GND	Ground
64(1), 51(2) 32(3), 21(4)	TXD	TRANSMIT DATA—(Terminal) Input for data transfer from the terminal. (Modem) Output for data transfer to a modem.
62(1), 53(2), 30(3), 21(4)	RXD	RECEIVE DATA—(Terminal) Output for data transfer to the terminal. (Modem) Input for data transfer from a modem.
60(1), 55(2), 28(3), 23(4)	RTS	REQUEST TO SEND—(Terminal) Input through which permission to transfer data is requested by the terminal. (Modem) Output through which permission to transfer data to a modem is requested.
58(1), 57(2), 26(3), 25(4)	CTS	CLEAR TO SEND—(Terminal) Output indicating that the terminal has permission to transmit data. (Modem) Input indicating that the modem is ready to transmit data.

TABLE 3 — Pin Signal Assignments, Edge Connector P2 (continued)

Connector Pin	Signal Mnemonic	Signal Name and Description
56(1), 59(2), 24(3), 27(4)	DSR	DATA SET READY—(Terminal) Output indicating the MCCM is ready. (Modem) Input indicating the modem is ready or off hook.
52(1), 63(2), 31(4)	DCD	DATA CARRIER DETECT—(Terminal) Output which is always on when MCCM is powered. (Modem) Input indicating the modem is receiving a suitable carrier.
50(1), 65(2), 18(3), 33(4)	DTR	DATA TERMINAL READY—(Terminal) Input indicating a ready condition. (Modem) Output indicating an MCCM ready condition. On-to-Off transition signals modem to "hang up" line.
35-48 67-120	— —	Not applicable to MCCM. See the VERSAbus Specification Manual M68KVBS for information on these signals.

NOTES:
1. In Connector Pin column, numbers in parentheses indicate I/O port assignments.
2. Descriptions with respect to MCCM when connected to (Terminal) or (Modem).

FIGURE 2 — MCCM Overall Installation Using The Optional M68KVM232CP Cable Assembly

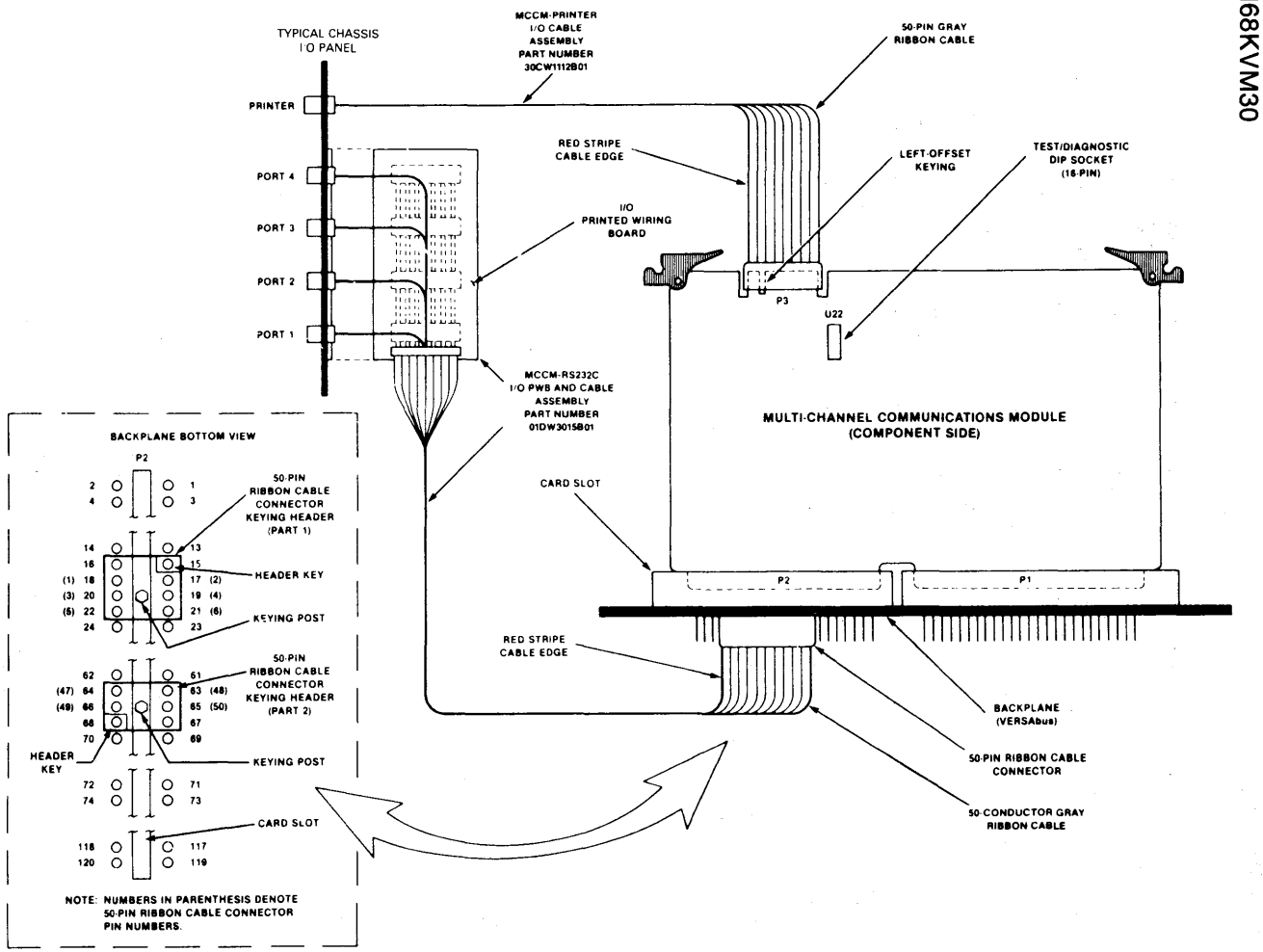
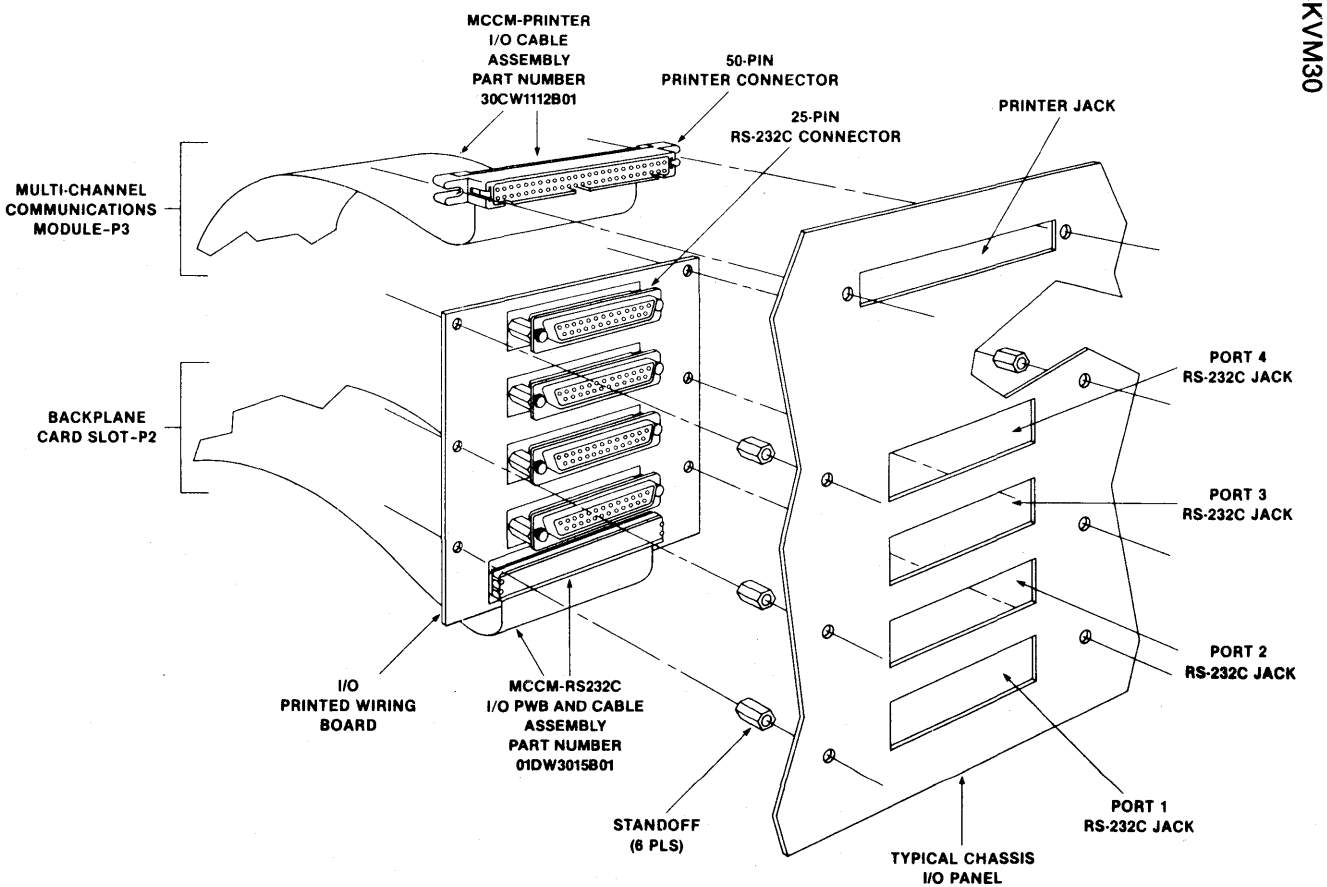


FIGURE 3 — Rear Panel Connector Installation Using The Optional M68KVM232CP Cable Assembly



Ordering Information

Part Number	Description
M68KVM30	VERSAmodule Multi-Channel Communications Module (MCCM). This communications module provides an interface for up to four asynchronous RS-232C serial I/O devices and one parallel I/O printer with a Centronics/compatible interface. The module is controlled from VERSAbus through an on-board Intelligent Peripheral Controller (IPC). Each serial I/O port may be programmed to operate at a standard baud rate ranging from 50–9600 bps. Cable assemblies are optional, and must be ordered separately. Includes User's Manual

Related Documentation

M68KVM30/D1	MCCM User's Manual
M68KVBS/D4	VERSAbus Specification Manual
M68KIPCS/D1	68000/IPC Command Channel Software Interface Specification Manual

Options

Part Number	Description
M68KVM232CP	VERSAmodule System Chassis (M68KVMCH1) rear panel mountable Cable Assembly comprises two cable assemblies for connecting MCCM to chassis I/O panel to accommodate external connections, from four RS-232C devices and a parallel I/O Centronics compatible printer
M68KVMPRTCE	10-ft. cable assembly provides connection between a printer and the VERSAmodule System Chassis mountable Cable Assembly (M68KVM232CP)
M68RS232-10	RS-232C 10-foot cable
M68RS232-25	RS-232C 25-foot cable
M68RS232-50	RS-232C 50-foot cable

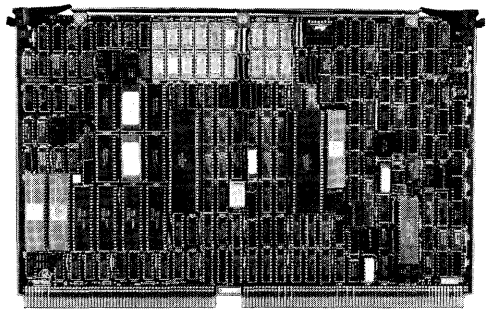
M68KVM31

VERSAmodule™
Eight-Channel Intelligent Communications Module

- Interfaces Eight Full Duplex, Synchronous/Asynchronous Serial Channels to the VERSAbus
- Asynchronous Internally Generated Baud Rates from 75 to 19.2K Baud
- Synchronous Internally Generated Baud Rates from 1200 to 153.6K Baud
- Aggregate Character Rate of up to 30K Characters Per Second
- Terminal/Modem Interface — RS-232C Standard
- Dual Motorola 10 MHz MC68000 Processor Architecture
- Multiprotocol — Asynchronous, Bisynchronous, SDLC/HDLC, X.25
- VERSAmodule Form Factor/VERSAbus Interface
- Intelligent DMA
- 128Kb Dynamic RAM, expandable to 512Kb; 32Kb Static RAM; 32Kb EPROM
- Generates Seven VERSAbus Interrupt Levels
- VERSAbus-accessible Command Channel Attention Register
- Interruptable from VERSAbus Command Channel Attention Register
- External Command Channel

The M68KVM31 Eight-Channel Intelligent Communications VERSAmodule expands the user's system environment by interfacing up to eight synchronous/asynchronous serial, terminal/modem devices to the VERSAbus. This dual-processor module provides all the functions of a high-performance communications processor and executes concurrently with other VERSAbus system processor modules. Figure1 is a system block diagram of the M68KVM31 showing the interrelation of the main functional features:

- Private RAM/ROM
- Shared RAM
- Command Channel Attention Register
- VERSAbus Interrupter
- VERSAbus Interface and Bus Requestor
- Character Processor
- Character Processor Status and Control Register
- Protocol Processor
- Protocol Processor Status and Control Register
- Eight Serial Ports
- Distribution Panel



PRIVATE RAM/EPROM

Private memory is accessed via the Character Processor bus. It consists of up to 32Kb of no wait state static RAM and up to 32Kb of EPROM mounted in six 28-pin sockets. Since two of the six sockets accommodate RAM only, two are for EPROM only, and two may be either RAM or EPROM. Additional EPROM may be added by reducing the available RAM. All RAM and all EPROM addresses are contiguous.

SHARED RAM

Shared RAM consists of 128Kb (expandable to 512Kb) of DRAM tri-ported for refresh, Character Processor access, and Protocol Processor access. Performance is as follows:

Write	— No Wait State
Read (parity disabled)	— One Wait State
Read (parity enabled)	— Two Wait State

COMMAND CHANNEL ATTENTION REGISTER

The Command Channel Attention Register consists of four read/write registers which perform the following functions:

- Pass Messages and Commands Between the VERSAbus Host and Onboard Processor
- Pass M68KVM31 Status to VERSAbus Host
- Interrupt M68KVM31 via VERSAbus Host

VERSAbus INTERFACE AND BUS REQUESTOR

The VERSAbus Interface provides complete interface-to-data transfers to/from global memory via the VERSAbus. The VERSAbus Requestor automatically requests VERSAbus control if the M68KVM31 is not the current bus master and an onboard processor attempts

a bus access. The VERSAbus Requestor provides an early Bus Busy Release on the last transfer. A programmable bus cycle count allows the developer to designate the number of VERSAbus data transfers which can occur before the M68KVM31 surrenders bus mastership.

CHARACTER PROCESSOR

The Character Processor's primary function is to control the transmission of data to and from eight serial ports. It responds to vector interrupts from each serial port and moves a character to or from shared RAM. The Character Processor has priority access to shared RAM and waits only for the completion of a Protocol Processor or refresh cycle. The Character Processor also initializes and controls the serial ports, controls and responds to the modem control lines, interrupts or accepts interrupts from the Protocol Processor, executes power-on/reset self-tests, and boots the M68KVM31 through either the VERSAbus or a serial port.

CHARACTER PROCESSOR STATUS AND CONTROL REGISTER

Two MC68230 Parallel Interface/Timers are used for this register which controls or monitors modem control lines and processor-to-processor interrupts. It also controls a number of miscellaneous functions, including Protocol Processor reset request, test lines, fail line and Command Channel Attention Register interrupts.

PROTOCOL PROCESSOR

The Protocol Processor handles communication between the M68KVM31 and the VERSAbus including data transfer between onboard shared memory and system memory, generation of VERSAbus interrupts and management of the Command Channel which is in system memory. The Protocol Processor also handles frame level protocol processing, interrupts the character processor and services interrupts from the character processor.

PROTOCOL PROCESSOR STATUS AND CONTROL REGISTER

The Protocol Processor Status and Control Register consists of one MC68230 Parallel Interface/Timer which controls:

- Onboard Serial Loop-back
- Even or Odd Parity
- VERSAbus Page Register (selects one of eight 2Mb "windows")
- Bus Cycle Count of the Bus Requestor
- Vectored Interrupts to the Protocol Processor by the Command Channel Attention Register

SERIAL PORTS

Eight serial ports support multiple protocols and provide daisy-chained vectored interrupts. The serial ports consist of four dual-channel devices capable of full duplex operation. The device supports asynchronous, bisynchronous, SDLC/HDLC and X.25 protocols. External clocks may be used with the module. The M68KVM31 board-generated clock rates are switch selectable and consist of the following baud rates:

Asynchronous	Synchronous
75	1200
150	2400
300	4800
600	9600
1200	19.2K
2400	38.4K
3600	57.6K
4800	76.8K
9600	153.6K
19.2K	

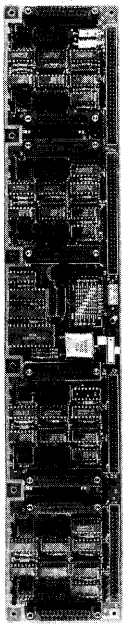
Clocks may also be supplied externally.

DISTRIBUTION PANEL

The VM31 distribution panel converts the TTL signal levels on the M68KVM31 P2 connector to RS-232C signal levels required for terminal/modem interface. The panel supplies transmit/receive clocks and loop-back clocks. It also provides terminal/modem jumper selection and baud rate selection for each port.

The RS-232C signals implemented are:

- Signal and Chassis Ground
- Transmit and Receive Data
- Transmit and Receive Clock
- Request and Clear to Send
- Data Terminal Ready
- Data Carrier Detect
- Modem Control Line Pin 18 and Pin 25



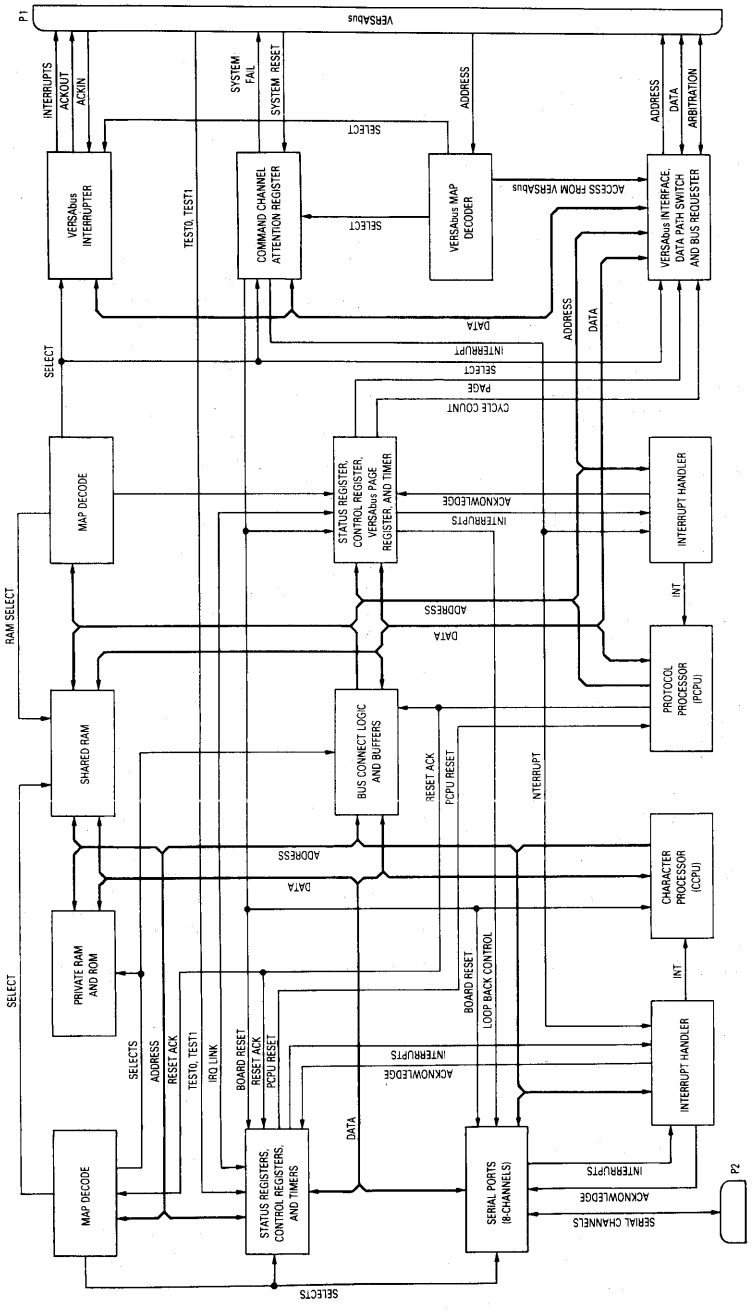


Figure 1. M68KVM31 Functional Block Diagram

REAL-TIME KERNEL

The M68KVM31 can be delivered with an onboard kernel designed to control task execution. The real-time kernel is a multitasking executive which provides the services necessary to operate in a real-time environment with minimal overhead. The executive supports the execution of multiple tasks with each task executing as a separate independent process. Functions of the executive include:

- Memory Allocation and Management
- Task Control
- Interrupt Service Management
- Supports Hardware and Software Interrupts
- Task Synchronization
- Task Dispatching
- Message (Event) Passing Between Tasks
- System Clock Functions
- I/O Initiation and Completion
- System Service Calls

M68KVM31 DEBUG/DIAGNOSTIC MONITOR

The M68KVM31 features an EPROM-resident Debug/Diagnostic Monitor package equipped with one-line assembler/disassembler capabilities. The Monitor's powerful debugging command set permits access to CPUs, I/O ports, onboard RAM, and the 16Mb addressing range of the VERSAbus. The Monitor is equipped with virtual terminal support which allows up/downline loading functions from an EXORmacs or VME/10 Development System.

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristics	Specifications
Power Requirements	+5 Vdc, ±5%, @ 7.2 A
VERSAbus I/O (P1) Signals Transfer type	 VERSAbus compatible Parallel, asynchronous
Terminal/Modem I/O (P2) Signals Transfer type	 LS TTL compatible Serial, synchronous/asynchronous, 75 to 307.2K baud
Temperature Requirements Operating Storage	 0° to +50°C −40° to +85°C
Relative Humidity	5% to 95% (non-condensing)
Dimensions Height Width Thickness	 9.25 in. (23.49 cm) 14.5 in. (36.83 cm) 0.622 in. (1.52 cm)

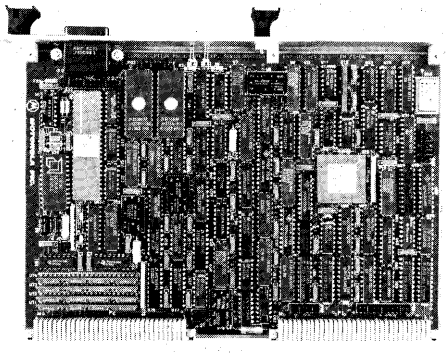
ORDERING INFORMATION

Part Number	Description
M68KVM31	Eight-Channel Intelligent Communications VERSAmodule Equipped to Interface Eight Full Duplex, Synchronous/Asynchronous Serial Channel to the VERSAbus. Includes User's Manual.
M68KVM31bug	Debug/Diagnostic Monitor for the M68KVM31 Eight-Channel Intelligent Communications VERSAmodule. Includes User's Manual.
TBD	M68KVM31 Onboard Real-Time Kernel for Multitasking Program Control.
M68KVM31/D	M68KVM31 Eight-Channel Intelligent Communications VERSAmodule User's Manual.

MOTOROLA

M68KVM33

VERSAmodule LAN Controller



- Frees VERSAbus Hosts From Protocol Processing Burden
- Interfaces with Both VMEbus (via MVME330) and VERSAbus Hosts in the Same Ethernet System
- Interfaces with Hosts Running Under VERSAdos and SYSTEM V/68 in the Same Ethernet System
- Based on AM7990 LANCE Ethernet Controller, AM7991 Serial Input/Output Adapter and MC68000 16/32-Bit Microprocessing Unit

Hardware Features

- 10 MHz MC68000 MPU
- VERSAbus Compatible
- Ethernet, Version 2.0 Compatible
- AM 7990/7991 (LANCE/SIA) VLSI
- 128K Dynamic RAM with Parity
- Interrupt Capabilities
  VERSAbus to MPU
  MPU to VERSAbus with Programmable Vector
- 2 ms Timer
- Bus Requester/Master Capability
- 8K or 32K EPROM
- Power Up Self-Test
- One Wait State RAM Write Access
- No Wait State RAM Read Access
- 32 bit VERSAbus addressing
- 0°–70°C Operating Temperature Range

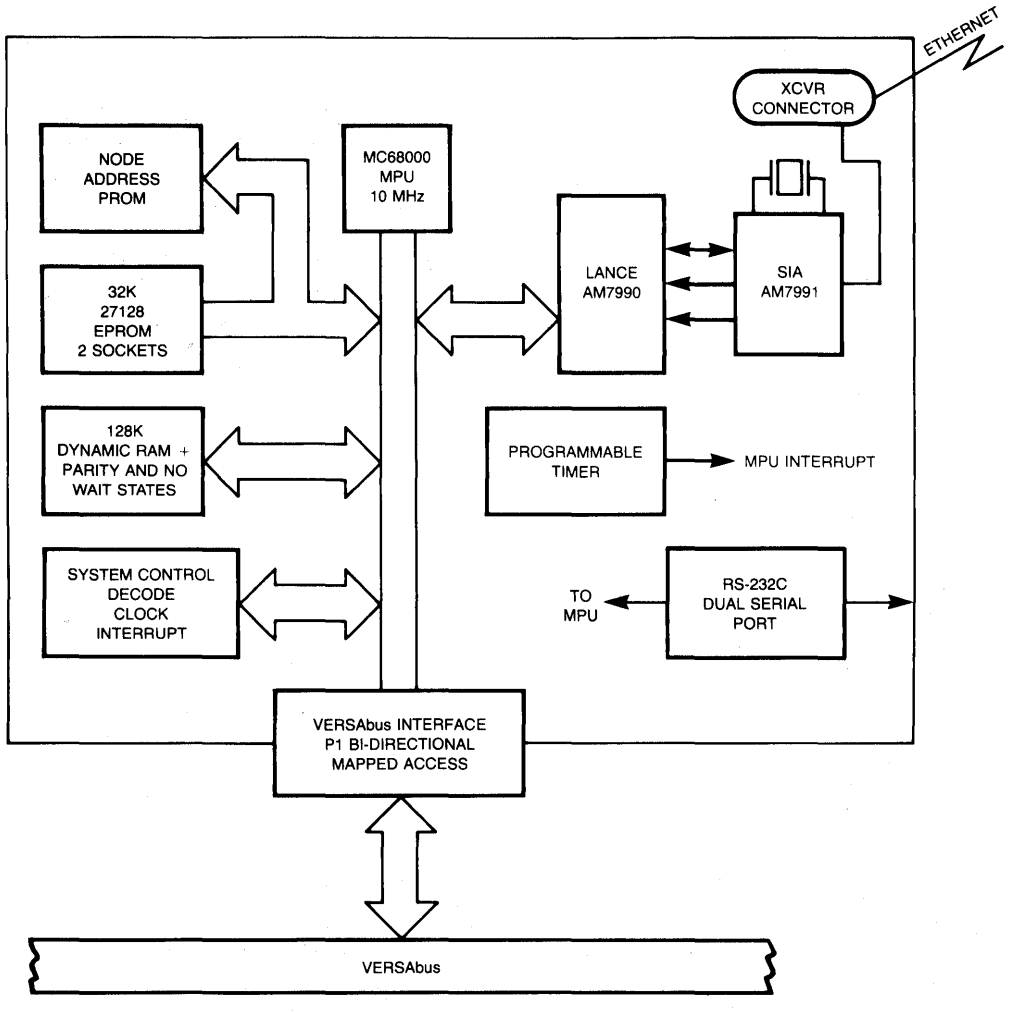
Software Features

- Host-Specific Application Packages: File Transfer, Network Utilities, Virtual Terminal, Runtime Library
- SYSTEM V/68 and VERSAdos Drivers
- XNS Protocol Package Includes: Echo, Error, Sequenced Packet, Packet Exchange, Routing Info, Datagram
- Firmware and Communications Executive
- Clean, Well-Defined Host Interface

Obtaining for a microcomputer system the data exchange benefits of Ethernet requires a means of connecting information processing devices to this local area network. The M68KVM33 LAN Controller VERSAmodule is an advanced communications processor which provides for VERSAbus-based systems the interface and the performance for 10 Mbps Ethernet 2.0 implementation. It matches Ethernet conformance to the IEEE 802.3 local area network specification (CSMA/CD), conforms to VERSAbus industry standards and is fully supported by Motorola's SYSTEM V/68 and VERSAdos operating systems.

M68KVM33 frees any VERSAbus host from significant protocol burden by means of a node processor design incorporating an MC68000 16-Bit Microprocessing Unit and the VLSI Lance Ethernet Controller — AM7990. A communications executive executes on the LAN controller MPU to supervise the Lance chip in its processing of the Xerox Network System (XNS) protocol package. A VLSI Serial Input/Output Adapter — AM7991 — provides Manchester encoding/decoding for the Ethernet interface. When the Ethernet interface is operating at 10 Mbps, the data exchange rate at the VERSAbus interface is about 200 one kilobyte packets per second. A block diagram of M68KVM33 is shown in Figure 1.

FIGURE 1 — M68KVM33 LAN Controller

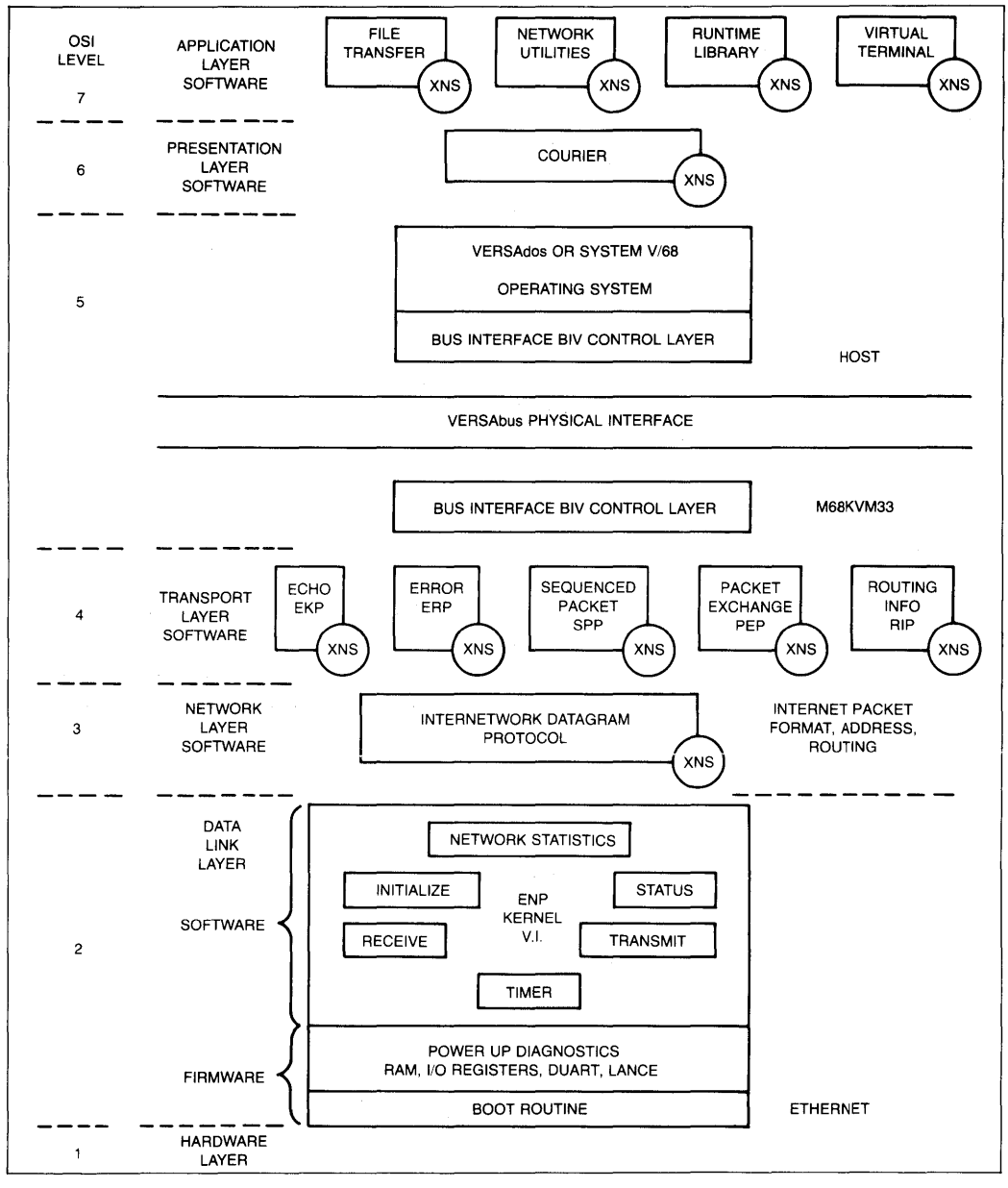


M68KVM33's on-board MC68000 MPU facilitates the downloading of custom protocol implementations. Host-specific drivers also are easily written for interfacing M68KVM33 to another host.

By interfacing directly with the LANCE chip to perform all Ethernet I/O, the M68KVM33 kernel acts as a standard interface between the hardware and the XNS software which in turn communicates, over VERSAbus to the host application software; and through the LANCE over Ethernet to another node and thence through that LANCE to the peer XNS software at that node.

The M68KVM33 communications executive comprises initialize, timer, transmit, receive, status and network statistic functions, as shown in Figure 2.

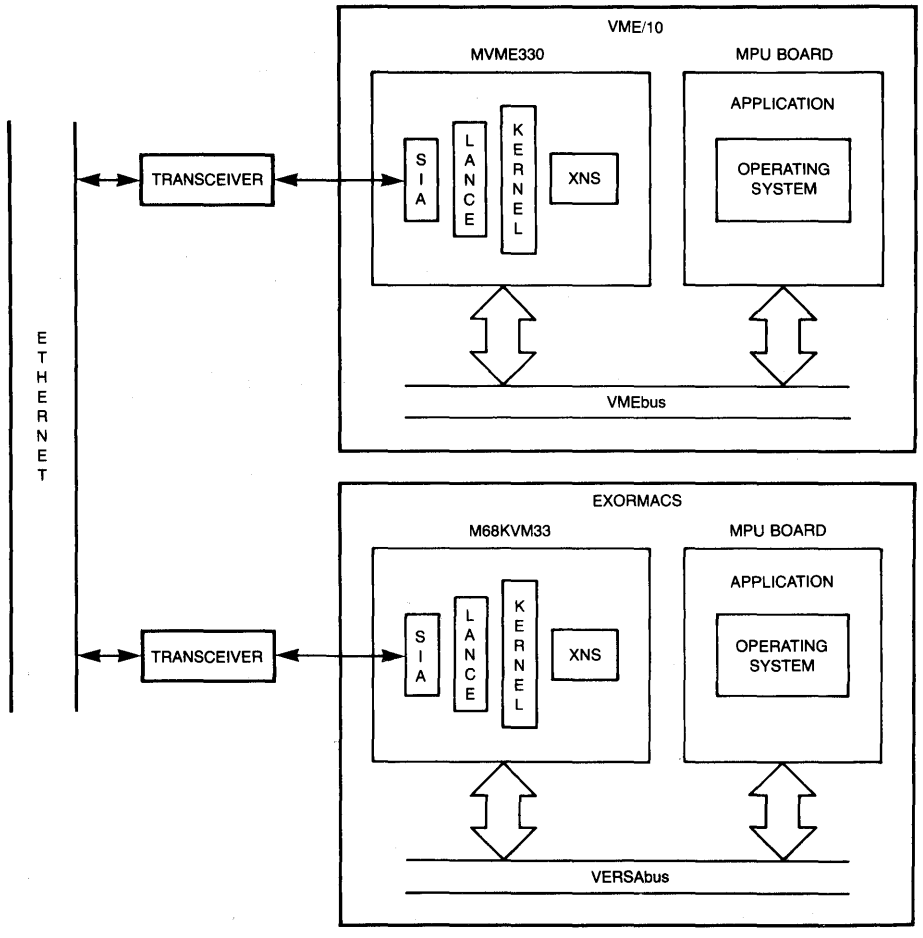
FIGURE 2 — Host/M68KVM33 Ethernet Functions



The M68KVM33-resident layers include XNS inter-network Datagram protocol in the network layer and XNS Echo, Error, Sequenced Packet, Packet Exchange and Routing Info modules in the transport layer. Within the host, an XNS Courier package interfaces the operating system with File Transfer, Network Utilities, Runtime Library, Virtual Terminal and user written application packages. Data transfer across the physical VERSAbus interface between host and LAN module is accomplished using a clean, well-defined shared memory interface.

Figure 3 depicts a two node Ethernet network.

FIGURE 3 — Two Node Network



FUNCTIONAL DESCRIPTION — HARDWARE

M68KVM33 offers SYSTEM V/68 and VERSAdos hosts on VERSAbus a single board connection to an Ethernet system. Compatible with Ethernet Version 2.0, it can support a data transfer rate greater than 200 maximum length packets per second. Requiring no additional hardware but a cable, M68KVM33 can be connected to standard Ethernet transceivers.

The M68KVM33 is designed so that it can easily be used in File, Print and Terminal Servers and other network applications. It facilitates network management by monitoring collisions and retrys, frame checking, timeouts and error free transmissions and receptions. Such configuration control functions as initializing, suspending, and resuming data link operations, setting the physical address and the addressing mode (normal, broadcast, multicast) are also supported by the hardware.

MPU

The processor on board M68KVM33 is a 10 MHz MC68000. Its duties include moving commands and data to and from system memory; responding to and generating VERSAbus interrupts, executing the network and transport XNS protocol layers, providing timer functions and, under direction of the communications executive; controlling the LANCE in its execution of the data link protocol.

Dynamic Memory

As shipped, M68KVM33 has 128K bytes of dynamic RAM with parity which is accessible from VERSAbus, the LANCE and the MC68000, in that order of priority.

ROM/EPROM

The M68KVM33 has two 28 pin sockets in which 16K x 8, 8K x 8 and 4K x 8 EPROM devices can be used.

Local Area Network Controller For Ethernet (LANCE)

The LANCE is a VLSI device used for interfacing a microprocessor system to a local area network; namely the baseband, carrier sense multiple access with collision detection (CSMA/CD) shared medium defined by Ethernet 2.0. It is designed to operate in an environment comprising local memory and a microprocessor with the memory serving as the communication link between the microprocessor and the LANCE and as a buffer for Ethernet packets.

LANCE operates at a 10 megabit data rate, is MC68000 compatible, has: a 16-bit data bus, a multiplexed address/data bus, a DMA controller with 24-bit addressing and a 48-byte data buffer with powerful buffer management. It also offers diagnostic aids, three modes of destination address comparison, executes a CSMA/CD network access algorithm and provides extensive error reporting. The 48 bytes of internal buffer reduce the initial response time so that once DMA is initiated between LANCE and local memory an average of one word is transferred each 1.6 microseconds sustaining a 10 Mbps data transfer rate.

LANCE diagnostics include:

- Pseudo full duplex capability for use in testing via incoming and outgoing loopback packets
- 32-bit CRC function usable in the transmission, reception and two loopback operating modes
- 10-bit wide time domain reflectometry counter used for determining the location of a cable fault.

Three modes of checking the received network destination address against initialization values are provided. These include:

- Physical mode: a comparison of address bits with corresponding bits in the physical address register
- Logical address filter using as an index the CRC value determined over the destination address
- Promiscuous mode: in which all packets are received regardless of address.

The LANCE executes the full carrier sense multiple access with collision detection (CSMA/CD) algorithm in which on detecting collision it transmits Collision Jam data and performs a backoff algorithm before again attempting to transmit. Only after sixteen consecutive collision detections does the LANCE report an error. Individual packet errors reported by LANCE include: CRC error, framing error and SILO under — or overflow.

LANCE reported errors which result in M68KVM33 generating an interrupt on the VERSAbus are: babbling transmitter (transmission of more than 1518 bytes), nonfunctional collision detection circuitry, missed packet due to insufficient buffer space and memory timeout.

Host/LANCE communications are accomplished via transmit and receive ring structures in memory. Each ring is a circular queue comprising up to 128 message descriptors four words long. Each descriptor defines a LANCE — controlled buffer or chain of buffers holding a packet awaiting further LANCE processing. Orderly management of the message descriptors is contingent on use of a bit in each descriptor whose state indicates ownership by LANCE or host of that descriptor and on strict observance of a protocol which allows LANCE and Host to relinquish but never take ownership and forbids changes to data in non-owned descriptors.

Serial Interface Adapter (SIA)

The SIA performs the Manchester encoding/decoding necessary for interfacing LANCE to Ethernet. It is compatible with standard Ethernet bus transceivers operating at 10

megabits/sec. The SIA decoder acquires the clocks and data within six bit times (600 ns). It features guaranteed carrier detection and collision detection threshold limits and transient noise rejection. The receiver decodes Manchester data in the presence of up to plus or minus 20 ns clock jitter, which represents 1/8 of a bit time.

VERSAbus Requester/Bus Mastership

Since M68KVM33 has bus request circuitry which complies with the VERSAbus specification, it can act as bus master in a system having an arbiter elsewhere on the bus. It supports all five bus request levels, the bus request in/bus grant out daisy chain and provides jumper selection of the module bus request level.

VERSAbus Interrupter

The M68KVM33 interrupter circuitry complies with the VERSAbus specification, supporting the interrupt acknowledge daisy chain and allowing jumper selection for the module of one of the seven interrupt priority levels.

Interrupting the M68KVM33 LAN Controller

A host access of any of the top 512 bytes of the 128K bytes of VERSAbus memory space assigned to the LAN Controller causes an automatic interrupt of the on-board MPU. This access differs from a true LAN Controller memory access in that no data is written in or fetched from the 512 bytes and the host need not wait for a memory acknowledge since a response is guaranteed and the on-board MPU is automatically interrupted.

Timer

For use by protocol processing software timers, the M68KVM33 LAN Controller has a timer which causes the on-board MPU to be interrupted every 2 ms.

Memory Map

Decoding of the upper seven address lines and a header permit jumper selection of a base address for the LAN Controller on 128K byte boundaries throughout the VERSAbus memory space. As bus master, M68KVM33 can access most of the bus address space.

Power Up Test

Upon power up, or system reset, the LAN Controller executes a series of ROM-based self tests to determine that the board is functioning properly. Upon successful completion of these, the fail LED is turned off. Tests include a LANCE register and loopback test, an MPU test, a memory test for the dynamic memory, EPROM checksum test, and a status and control register test (I/O registers).

M68KVM33 SOFTWARE

The M68KVM33 is supplied with host — specific software/firmware offering the functionality and performance for immediate use and for implementing custom applications. The software supports the Motorola SYSTEM V/68 and VERSAdos operating systems and provides a communications executive kernel which controls the LANCE hardware in its performance of basic Ethernet data transfer, status reporting, statistical and diagnostic functions.

The software is designed to interconnect VME/10 and EXORmacs based systems via Ethernet. It is highly modularized and observes the 7-layer interconnection model (open System Interconnect — OSI) defined by the International Standards Organization (ISO). So that the software can easily be upgraded to future protocols, each module corresponds to a specific layer of the OSI interconnection model. Device drivers running under the Motorola SYSTEM V/68 and VERSAdos operating systems are also provided. For further flexibility and ease of use, the bus interface between a host system and the LAN controller utilizes a clean well-defined shared memory protocol. (The initial software supports only MMU-based systems.)

Kernel Functions

The LAN Controller kernel performs all functions required for controlling and monitoring M68KVM33 hardware. Included are: managing LANCE status registers and LANCE-controlled message descriptor rings, performing timer functions, managing interrupts and retrieving LANCE generated statistics. Through use of these functions, user software can be written in PASCAL, C or other high level language and down-loaded to the LAN Controller for final development.

Debugger

To aid diagnostics and development, an optional full debugger is available. Provided functions include setting multiple breakpoints, memory examine, modify, test and move; an assembler, a disassembler, download capability and software memory refresh.

Application Packages

The M68KVM33 LAN Controller comes with a powerful set of application packages. These include:

- File Transfer: Files can be moved between two VME/10 and EXORmacs hosts on the network.
- Virtual Terminal: From the terminal of one host, a user via Ethernet can log on a remote host and execute commands and programs on that computer as though directly connected. At present; only Motorola SYSTEM V/68 to Motorola SYSTEM V/68 and VERSAdos to SYSTEM V/68 communications are supported.

- Electronic Message: Users can send and receive messages across the network. At present, there is no message queue for users not logged into network management when a message is sent.
- Datagram: A user program running on one host can send/receive information to/from a user program running on another host without first establishing a virtual connection. A datagram is a single packet with self-sufficient addressing information. Electronic message facilities are based on the datagram interface.
- Network Management: Each station is provided daily, hourly or per minute packets sent/received information. From this and LANCE reported errors, a network manager can decide to remove any port or station having excessive errors. A utility interfacing network management can request display of information within Ethernet data structures on board the LAN Controller.

File Transfer Capabilities

Files are transferred between nodes on Ethernet using the File Transfer application package. At present the following services are provided:

- File Management
- File Access
- File Protection, on both the SYSTEM V/68 and VERSAdos operating systems.

Protocols for Higher Layer Software

As high layer software support for the MVME330 LAN Controller, Motorola offers a XEROX Network Service (XNS) implementation consisting of the following protocols:

- Internetwork Datagram Protocol — corresponding to layer 3b of the OSI model, this LAN — resident protocol formats internet packets and performs internet addressing and routing.
- Transport Layer Protocol — corresponding to layer four of the OSI model, this LAN — resident protocol handles echo, error, sequenced packet, packet exchange and routing information.
- Courier (Remote Procedure) Protocol — corresponding to layer six of the OSI model, this host-resident protocol standardizes the format of request and reply messages and the format of the network representatives for a family of data types from which request and reply parameters can be constructed.

Operating Systems Interfaces

The M68KVM33 LAN Controller provides a bus interface for VMEbus/VERSAbus (BIV) in the form of shared memory protocol. This interface provides access to Ethernet from Motorola SYSTEM V/68 and VERSAdos Operating Systems via the corresponding driver supplied with the software.

To facilitate its use in the development of custom application packages or network protocols, the BIV is clean and well defined.

Systems Integration

Two basic capabilities useful for integrating system software with system hardware are provided with M68KVM33: download and network management.

In conjunction with other kernel software; the download function can assist with board diagnostics and program development by allowing the required software to be downloaded. The network management function provides a network manager with the mechanism for administering a network system, for setting configuration options and privilege levels and for generating network reports. Commands within network management have privilege levels: user level for accessing parameters and statistics via any other connection.

The network manager can request generation of performance statistics for:

- Busiest Second
- Busiest Minute
- A Specified Hour
- Average for the Preceding 24-hour Period.

Other statistics are supported by the kernel which can be utilized as applications require. Supported statistics are listed in Table 1. Supported statistics are tallied from the last reset of the statistics or of the LAN Controller.

TABLE 1 — Network Statistics

Ethernet messages transmitted	Collision errors
Multiple Retries reported	Memory errors on transmit
Single Retries reported	Ethernet messages received
Deferrals reported	Missed packets reported
Transmit buffer error	CRC errors reported
Silo underruns	Framing error
Late collisions	Silo Overruns
Carrier loss	Memory errors on receive
Babbling Transmitter errors	

Network Performance

The system throughput, including upper layer XNS software, is 200 packets/second, for packets of 1024 bytes, average. The M68KVM33 can handle multiple back-to-back packets of any length.

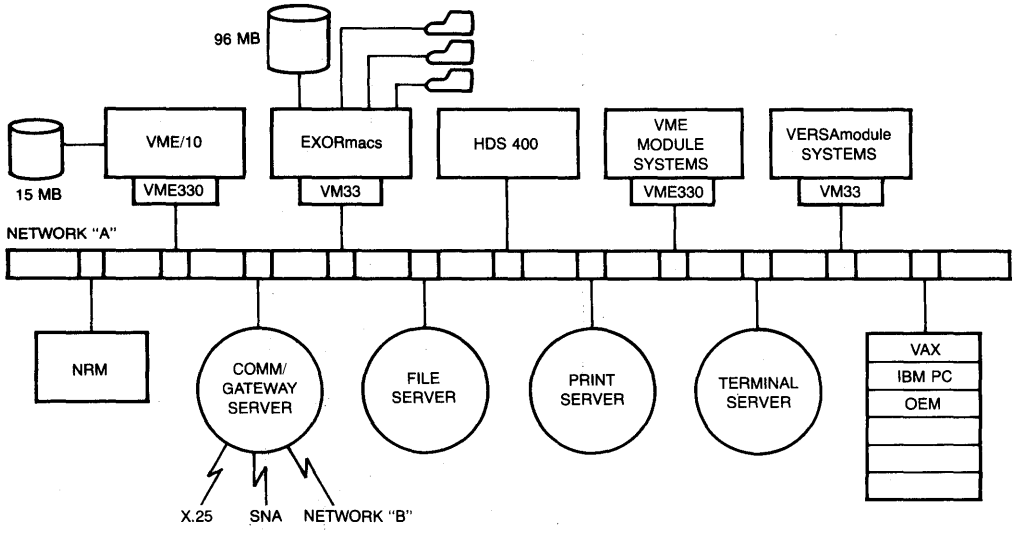
SYSTEMS NETWORK CONCEPT

Figure 4 is an overall concept of an Ethernet system involving the various Motorola Microsystems hardware and software element described in this data sheet.

NETWORK DISTANCES

The maximum distance between any two receivers is 2500 meters. The maximum length of the transceiver cable (from the LAN Controller to the Ethernet cable) is 50 meters.

FIGURE 4 — Systems Network Concept



HARDWARE
- VME330
- VM33
- VME300
- VME310
- VME400
- VM31
- VM32

MEDIUM
- ETHERNET
- TOKEN BUS
- TOKEN RING

PROTOCOL
- XNS
- TCP-IP
- ISO

FEATURES
- FILE TRANSFER
- VIRTUAL TERMINAL
- ELECTRONIC MAIL
- PROCESS-TO-PROCESS
- NETWORK RESOURCE MGMT.
- VIRTUAL RESOURCE UTILIZATION

M68KVM33 SPECIFICATIONS

The specifications for the M68KVM33 are listed in Table 2.

TABLE 2 — M68KVM33 Specifications

Characteristic	Specification
Power Requirements	4.0 A @ +5 Vdc ±5% 0.6 A @ +12 Vdc ±5% 0.1 A @ −12 Vdc ±5%
Environmental Tolerance	
Operating Temperature	5°C to 50°C
Storage Temperature	−40°C to 85°C
Humidity Range	0% to 95% (non-condensing)
Mechanical Specifications	
Height x Depth (board)	9.25″ (235 mm) x 14.5″ (368 mm)
Height x Width (front panel)	9.25″ (235 mm) x 0.6″ (15.24 mm)
Connectors	
VERSAbus	Stanford Applied Engineering CPH 7000-140 ST Micro Plastics Inc. MP-0100-60-DW-5H
Ethernet transceiver cable port	Amphenol AMP745094-1

VERSAbus Connector P1

The electrical and mechanical characteristics of VERSAbus connector P1 are fully described in the VERSAbus Specification Manual, M68KVBS/D4.

Ethernet Transceiver Port Connector

The Ethernet transceiver port connector is a 15-pin subminiature, AMP745094-1.

Ordering Information

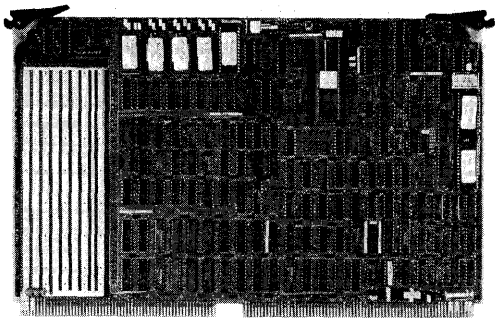
The M68KVM33 LAN Controller is ordered as a standalone board which includes the executive kernel software; or as a package including the board, executive kernel, a complete set of XNS protocol and network application software, and appropriate host-specific software including device driver and the BIV bus interface control layer. The entire software package (less kernel firmware) may be purchased separately.

Part Number	Description
M68KVM33	VERSAmodule Ethernet LAN Controller. This module provides high performance, intelligent single board connection of VERSAbus Systems to Ethernet, a Local Area Network. Includes 128K RAM, LANCE (7991), SIA (7990), 68000 MPU, Kernel Firmware and Power up self-test.
M68NNHBVMLAN for System V/68 Host M68VVHBVMLAN for VERSAdos Host	Object Software supplied on 8″ floppy. Object code modules include: — XNS protocol package including Echo, Error, Sequenced Packet, Packet Exchange, Routing Info and Datagram. — Network Application Software including File Transfer, Network Utilities, Runtime Library, Virtual Terminal. — Host Specific Network Software including Device Driver, Host BIV and M68KVM33 BIV. — Software documentation, and User's Manual. Object code is supplied as bootable load modules, and unlinked modules, so that the OEM can reconfigure without source.
M68KVM33-UX	M68KVM33 plus appropriate software for SYSTEM V/68 on 8″ floppy
M68KVM33-VX	M68KVM33 plus appropriate software for VERSAdos on 8″ floppy

NOTE: Cable with AMP745094-1 connector, or equivalent, required for serial communications.

3

VERSAmodule Universal Intelligent Peripheral Controller



The Universal Intelligent Peripheral Controller (UIPC) provides a simple means of interfacing special purpose, relatively complex, high-speed I/O devices to the VERSAbus. Offering all the standard computer interface and control functions, it frees the user to concentrate on the unique aspects of his interface design. A functional block diagram of the UIPC is shown in Figure 1.

The UIPC consists of a microcomputer unit based on the MC68120 IPC, a DMA controller, a FIFO Buffer and a VERSAbus interface.

Hardware Features

- VERSAbus Compatible
- Simple Interface via 50-pin On-Board Ribbon Cable Connector to User Physical Device Logic (Port A) or User Logic may be Implemented in On-Board Wirewrap Area (Port B)
- MC68120 Provided Serial Port (Port C) Baud Rate: 300 to 9600 Baud
- Shared RAM Command/Status Channel Between IPC and VERSAbus
- High-speed DMA Data Channel to VERSAbus with 8 μs Burst Mode Operation
- DMA Transfers can be Directed to any Address in 16 MB VERSAbus Memory Space
- Four port 32 Byte FIFO Buffer Allows Data to be Transferred Between Any Two Ports: VERSAbus, Port A, Port B, MC68120
- VERSAbus Interrupter.

Software Features

- UNIbug Debug Monitor
- UNITst Self-Test Controller
- Provision for User-Defined Self-Test Routines
- Real-Time Executive aids in Interfacing User-Provided Application Firmware
- Input and Output Processors Support Standard Command Channel Interface Protocol
- UIPC Self-Test Firmware
- Macro Calls for DMA Control
- Documentation to Support User Implementation of:
  - — Hardware Interface
  - — Device Control and Self-Test Firmware
  - — System I/O Driver Software
- Documentation to Aid Development of I/O Driver for Addition to VERSAdos Operating System

The microcomputer unit includes the MC68120 IPC chip with internal 128 byte dual port RAM and serial port, 4K bytes of static RAM and four ROM/EPROM/RAM sockets.

A four port, 32 byte FIFO buffer allows data to be transferred beween any two of the four UIPC ports: MC68120, VERSAbus, Port A or Port B.

The UIPC is controlled by memory mapped registers located in the MC68120 memory map. Table 1 lists the UIPC control registers.

DMA DATA TRANSFER

Data is transferred to the VERSAbus by a high-speed DMA controller. In the usual transfer mode and in one operation, a block of data ranging in size up to 64K bytes can be transferred to any starting address in the 16 megabyte VERSAbus memory space. Source data can also start from or end on odd or even byte boundaries and can also be transferred to alternate odd or alternate even byte addresses, if desired.

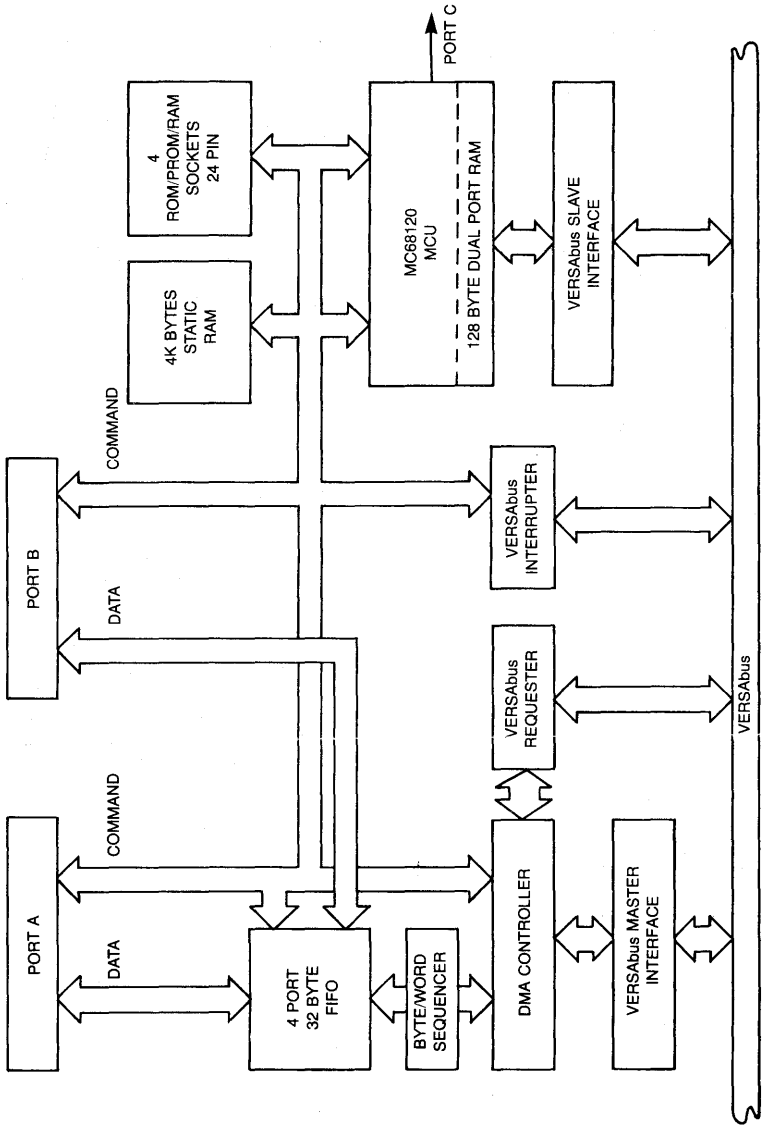


FIGURE 1 — UIPC Functional Block Diagram

TABLE 1 — UIPC Control Registers

MC68120 Address	Logic Element	Accessed By
$30	DMA address counter MSB	Write only
$31	DMA address counter LSB	Write only
$32	DMA address counter LSM	Write only
$33	DMA byte counter MSB	Write only
$34	DMA byte counter LSM	Write only
$35	Reserved/not used	
$36	Reserved/not used	
$37	Reserved/not used	
$38	FIFO input/output MSB	Read/Write
$39	FIFO input/output LSB	Read/Write
$3A	DMA/FIFO control register	Write only
$3A	DMA/FIFO status register	Read only
$3B	VERSAbus control register	Write only
$3B	Processor VERSAbus location	Read only
$3C	Processor interrupt register	Read/Write
$3D	Processor watchdog timer reset	Write only
$3E	Processor to VERSAbus interrupt	Write only
$3F	DMA interrupt/FIFO clear	Write only

To maintain pace with the user device, during DMA transfers to VERSAbus, the DMA controller requests VERSAbus mastership whenever the FIFO buffer needs to be filled or emptied. The DMA controller holds bus mastership for a maximum of 8 μs or until the FIFO is full or empty before releasing it for use by other bus masters.

Additional bus mastership requests are generated in the above manner until data transfer is complete. The controller then interrupts the MC68120.

Data Transfer Rates

When used with the M68KVM10 128K byte dynamic RAM module, the maximum data transfer rate is approximately 3 Mbytes/sec (1.5 mwords/sec) with actual data rates dependent on the cycle time of the addressed memory card. Use of the VERSAbus by other bus masters will reduce this rate. The transfer rate may be increased by using faster RAM cards.

VERSAbus Interface

The VERSAbus interface provides standard VERSAbus data transfer, interrupt and bus requester functions as defined in the VERSAbus Specification (M68KVBS). The base address, interrupt vector number, interrupt request priority level, and the bus request priority level are selectable by on-board strap options permitting use of more than one UIPC per system.

Refer to the VERSAbus Specification Manual M68KVBS, for Connector P1 pin assignments and detailed signal descriptions. Table 2 provides pin assignments and signal descriptions for Connector P2.

TABLE 2 — Connector P2 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Signal Name and Description
1–6	GND	GROUND
7–10	+5 V	+5 Vdc Power — Used by module logic circuits.
11, 12	+12 V	+12 Vdc Power — Not used.
13, 14	GND (±15 V)	ANALOG GROUND — Not used.
15, 16	−12 V	−12 Vdc Power — Not used.
17–66, 71–120	[I/O Pin]	INPUT/OUTPUT PIN — Not used.
67, 68	−15 V	−15 Vdc Power — Not used.
69, 70	+15 V	+15 Vdc Power — Not used.
71–87, 99–100	[RESERVED]	RESERVED — Not used.
88	APARITY1*	ADDRESS PARITY 1 — Not used.
89–96	A24*–A31*	ADDRESS BUS (bits 24–31) — Not used.
97, 98, 101, 102	GND	GROUND — Not used.
103	DPARITY*	DATA PARITY 2 — Not used.
104	DPARITY3*	DATA PARITY 3 — Not used.
105–120	D16*–D31*	DATA BUS (bits 16–31) — Not used.

NOTE: 1. Refer to VERSAbus Specification Manual (M68KVBS) for detailed information.

User System Interfaces

Ports A, B, and C allow user hardware to be connected to the UIPC.

Port A is a well defined, buffered interface for connecting hardware to the UIPC via a 50-pin ribbon cable (five feet max.). Port A provides a data path and a command path. The data path is a byte wide data link to the FIFO and is used for high-speed transfer of data from the hardware, generally to the VERSAbus. The command path is an extension of the MC68120 address and data bus lines that allows the UIPC to access control registers in the user interface hardware. Sixty-four bytes of the total MC68120 address space are reserved for the interface. Figures 2 and 3 show the UIPC memory map as seen from VERSAbus and the MC68120. Table 3 shows the pin assignments and signal descriptions for Port A.

FIGURE 2 — UIPC Memory Map from VERSAbus

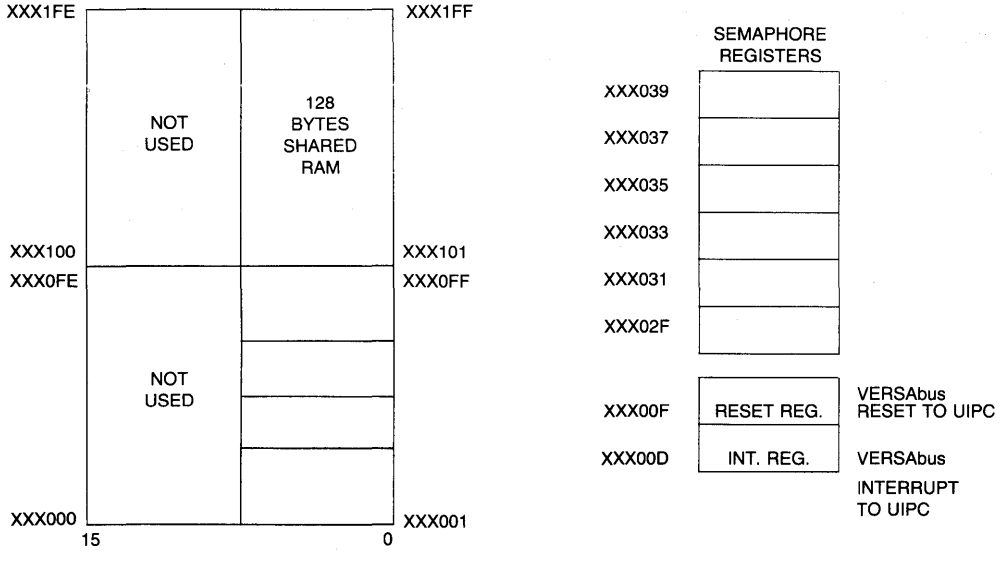


FIGURE 3 — UIPC Memory Map From the MC68120

Address	Logic Element(s)	Comments
FFFF F800	UIPC UNIbug Monitor	Enables U1
F7FF F000	Reserved	
EFFF E000	UIPC Self-Test UIPC Real-Time Executive	Enables U2
DFFF D000	User-Available for Program	Enables U3
CFFF C000	User-Available for Program	Enables U4
BFFF B000	U1 May be located here when an MC68120 with Mask ROM is used	Optionally Enables U1
AFFF 4000	User-Available	
3FFF 2000	UIPC-Reserved	
1FFF 1000	UIPC RAM	
0FFF 0100	User-Available	
00FF 0080	UIPC Command Channel (MC68120 Shared RAM)	
007F 0040	Port A Control Resistors	
003F 0030	UIPC Control Registers	
002F 0020	UIPC Reserved	
001F 0000	MC68120 Internal Registers	

TABLE 3 — Port A Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Signal Name and Description
1, 5	NC	No Connection
2, 4, 6, 7, 9, 19, 20, 29, 30, 32, 34, 35, 37, 38, 40–42	GND	GROUND
3	CIRQ*	CONTROL INTERRUPT REQUEST. From port A to processor. Active when low.
8	CE	CONTROL ENABLE. A redriven E CLOCK signal from the processor.
10	CRESET*	CONTROL RESET. From RESET switch or SYSRESET. Used to initialize port A circuitry. Active when low.
11	CSEL*	CONTROL SELECT. From processor. Used to indicate that port A circuitry is being enabled. Active when low. Address is $40–$7F using standard FPLA.
12	CWRITE*	CONTROL WRITE. When low, indicates a write function from the processor to port A. When high, indicates a read function from port A to the processor. Meaningful only when the CSEL* is low.
13–18	CA0–CA5	CONTROL ADDRESS bus (bits 0–5) lines from processor. May be decoded to allow processor to access up to 64 elements in port A. CA0 is the LSB. Meaningful only when the CSEL* signal is low.
21–28	CD0–CD7	CONTROL DATA bus (bits 0–7) lines between processor and port A. When CWRITE* is low, data flow is from processor to port A. When CWRITE* is high, data flow is from port A to processor. CD0 is the LSB. Meaningful only when CSEL* is low.
31	DWRITE*	DATA WRITE. Used between FIFO buffer and port A using the DD bus. When low, indicates a write function from FIFO buffer to port A. When high, indicates a read function from port A to FIFO buffer.
33	DSELECT*	DATA SELECT signal. Used between FIFO buffer and port A. When low, indicates FIFO control has been set up for a data transfer between port A and FIFO buffer. This signal is low for both a read and a write operation.
36	DREADY*	DATA READY. Used between FIFO buffer and port A. When low, indicates port A is ready to receive byte of data from FIFO buffer, or port A is ready to transmit a byte of data to FIFO.
39	DSTROBE*	DATA STROBE. Used between port A and FIFO buffer. Meaningful only when DSELECT* signal is low. When there is a data transfer from FIFO buffer to port A, a DSTROBE* low instructs port A to strobe data from the DD bus and force the DREADY* signal high until port A can accept the next byte of data. When there is a data transfer from port A to the FIFO buffer, a DSTROBE* low instructs port A to gate data onto the DD bus and force the DREADY* signal high until the data has been gated onto the bus.
43–50	DD0–DD7	DATA bus (bits 0–7) between port A and FIFO buffer. Meaningful only when DSELECT* signal is low. Bidirectional bus. Direction of data flow controlled by the DWRITE* signal.

Port B can be used to interface circuitry implemented in the on-board wirewrap area to the UIPC. Port B is unbuffered for access to the internal UIPC bus structure and can provide a data path and command path similar to port A. Table 4 provides pin assignments and signal descriptions for Port B.

TABLE 4 — Port B Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	Signal Name and Description
1	E	ENABLE signal from processor to port B. Port B may use this signal to cycle-share RAM.
2–5, 7, 8, 10, 12–20	LA0–LA15	LOCAL ADDRESS bus (bits 0–15). Main address bus for processor synchronous operation. Active when high. The bus may serve as input to port B, or by activating the UABE* signal, port B can drive the bus. LA0 is the LSB.
6	RWE*	RAM WRITE ENABLE. Write pulse to the 4K bytes of RAM. Active during the high state of the E signal when processor writes into RAM. Driven by an open collector device, it may be controlled by port B during the low state of the E signal, when port B writes into RAM.
9	UABE*	USER ADDRESS BUS ENABLE. Output from port B. When driven active, will isolate the processor for the LA, MD, and RD buses. This permits port B to control any or all of these buses through its access. The processor must be halted prior to activating UABE*.
11	UNLOAD	UNLOAD signal input to port B. Becomes active during a port B to FIFO buffer data transfer, indicating the FIFO buffer has accepted the byte of data on the FIBUS and port B should drop its UOR request.
21–28	MD0–MD7	MEMORY DATA bus (bits 0–7) for the processor RAM/ROM. Active when high. The bus may serve as an input to port B, or by activating the UABE* signal, port B can drive the MD bus. MD0 is the LSB.
29, 30	USI0–USI1	USER SELECT IN (bits 0 and 1). Output from port B. Permits modifying the processor address map for synchronous bus system.
31, 32, 34	US0–US2	USER SELECT OUT (bits 0–2). Input to port B. User may program the FPLA device to activate these leads for a selected group of processor addresses.
33, 35–44, 46–50	RA0–RA15	RAM ADDRESS bus (bits 0–15). Main address bus for the processor RAM. Active when high. Normally used during high state of the E signal but available to port B during low state of E signal. RA0 is the LSB.
45, 64, 66, 68, 71–73, 79, 84–87, 89–111, 113–133, 141, 143–146	NC	No Connection.
51	CS2*	CHIP SELECT (bit 2) for RAM chips U25 and U29. Driven by an open collector device, this signal is active only during the high state of the E signal. This makes CS2* controllable by port B during the low state of the E signal to gain access to RAM chips U25 and U29.
52	CS3*	CHIP SELECT (bit 3) for RAM chips U24 and U28. Similar to signal CS2*.
53	CS0*	CHIP SELECT (bit 0) for RAM chips U27 and U31. Similar to signal CS2*.
54	CS1*	CHIP SELECT (bit 1) for RAM chips U26 and U30. Similar to signal CS2*.
55	CS7*	CHIP SELECT (bit 7) signal that may be strapped to control memory element in U1.
56	CS6*	CHIP SELECT (bit 6) signal that may be strapped to control memory element in U2. Similar to signal CS7*.
57	CS5*	CHIP SELECT (bit 5) signal that may be strapped to control memory element in U3. Similar to signal CS7*.
58	CS4*	CHIP SELECT (bit 4) signal that may be strapped to control memory element in U4. Similar to signal CS7*.

TABLE 4 — Port B Pin Assignments and Signal Descriptions (continued)

Pin Number	Signal Mnemonic	Signal Name and Description
59–62, 67	USX3*–USX7*	USER SELECT EXTERNAL (bits 3–7). Input to port B. User may program the FPLA device to activate these leads for a selected group of processor addresses.
63	NMI*	NON-MASKABLE INTERRUPT. Output from port B. When active, causes the processor to perform a non-maskable interrupt sequence.
65	IRQ*	INTERRUPT REQUEST. Output from port B. When active, causes the processor to perform an interrupt sequence unless the mask inhibits it.
69, 75, 77	GND	GROUND
70	PORTBINT	PORT B INTERRUPT. Output from port B. When active, causes an interrupt request to the processor. The processor can read the state of this signal as bit 2 of the processor interrupt register ($3C). The PORT B INTERRUPT is grounded when shipped from the factory. To enable use, the ground must be removed by cutting the trace between J4–69 and J4–70.
74	URE*	USER READ ENABLE. Input to port B. When active, indicates port B is selected as the source of data for the FIFO buffer. The user must furnish logic to gate the data onto the FIBUS using this control signal.
76	UIR	USER INPUT READY. Output from port B. Used during transfer of data from the FIFO buffer to port B to request the FIFO buffer to furnish a byte of data to the FOBUS.
78	UOR	USER OUTPUT READY. Output from port B. Used during transfer of data from port B to the FIFO buffer to request the FIFO buffer to accept the byte of data placed on the FIBUS by port B.
80	UWE*	USER WRITE ENABLE. Input to port B. When active, indicates that port B is selected as destination for FIFO buffer data. The user must furnish logic to accept the FIFO buffer data.
81	RESET*	RESET. Input to port B. When active, initializes port B circuitry to the correct starting state.
82	EDEL	E DELAYED. Input to port B. May be useful in forming write pulses.
83	LWRITE*	LOCAL WRITE. Input to port B from the processor.
88	LOAD	LOAD. Input to port B. Becomes active during a FIFO buffer to port B data transfer. Indicates the data on FOBUS should be strobed into a user-provided register. Port B should immediately drop UIR request line.
112	ACLOCK*	CLOCK. Input to port B. 12 MHz squarewave. May be useful in port B circuitry.
134–140, 142	FIBUS0–FIBUS7	FIFO IN BUS (bits 0–7). Input bus used in all data transfers to the FIFO buffer. These signals are active when high. User must provide logic to drive this bus. FIBUS0 is the LSB.
147–154	RD0–RD7	REGISTER DATA bus (bits 0–7). Data read/write bus for the TTL part of the processor. Active when high. The bus may be used to extend the processor controlled registers into port B, or by activating the UABE* signal, port B can drive the RD bus. RD0 is the LSB.
155–162	FOBUS0–FOBUS7	FIFO OUT BUS (bits 0–7). Output bus used in all data transfers from the FIFO buffer. When Port B is to be the destination for the data, user must provide an interface register and control to store the FOBUS data. FOBUS0 is the LSB.

Port C is a simple RS-232C-compatible interface provided by the MC68120 MCU. The baud rate is selectable from 300 to 9600 baud. Table 5 provides the pin assignments and signal descriptions for Port C.

TABLE 5 — Port C Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonic	RS-232C Pin Number	Signal Name and Description
3	TXD	2	TRANSMIT DATA from CRT to processor.
5	RXD	3	RECEIVE DATA from processor to CRT.
9	CTS	5	CLEAR TO SEND — Always +12 Vdc.
11	DSR	6	DATA SET READY — Always +12 Vdc.
13	GND	7	GROUND.
15	DCD	8	DATA CARRIER DETECT — Always + 12 Vdc.

UIPC Control of Peripherals

Control of a peripheral is initiated by the host sending a command to the UIPC, generally through the dual port RAM channel. This command, usually a request for the peripheral to perform a "macro" function, is converted by the MC68120 to a series of executable "micro" commands under the direction of firmware, both supplied and user, on the UIPC. The firmware may also provide indication of function completion.

Both host and MC68120 can send a VERSAbus interrupt to the other. Headers are provided for setting by jumper the UIPC interrupt level and vector number.

Development Firmware

To aid in development, the following firmware is supplied with the UIPC: the UNIbug monitor, the UNItst self-test controller, a real-time executive providing UIPC and input/output control, and UIPC self-test. The UNIbug monitor and the UNItst self-test controller are supplied in a single 2K EPROM. UNIbug provides aid in hardware and firmware debugging and includes load, trace, breakpoint, memory display/set and register display/set commands.

The self-test controller, UNItst, first tests the MPU and the UNIbug ROM and RAM required by UNIbug. UNItst then tests all ROM and RAM resources required by the process modules. Finally, self-test routines within user-written modules are enabled.

UIPC self-test routines, the real-time executive and the process modules supplied with the UIPC are packaged in a 4K EPROM. The real-time executive program controls all UIPC operation and is highly modular to facilitate the addition of process modules written for the user application. The program consists of a monitor control program, interrupt handlers and provisions for accessing up to sixteen process modules. Four are included with the executive. Interprocess communication is implemented using monitor operator calls (MOCS) and circular queues.

Software Interface to the UIPC

Software interface of UIPC to VERSAbus as implemented in the supplied input process module reflects the standard command channel used in other MC68000-based Motorola systems. This command/status transfer protocol is fully described in the M68000/IPC Command Channel Software Interface Reference Manual, M68KIPCS. Examples of command packet construction and transmission are included.

The UIPC Module User's Manual describes how to develop a simple digital interface to a peripheral device, how to develop application firmware to control the device, and how to develop test firmware for device/board self-test.

Physical Interface to the UIPC

Physical interface to the UIPC is from the Port A connector via a 50-pin ribbon cable.

The VM60 UIPC is used together with a Disk Interface Module (DIM) by Motorola in its two-board M68KVM21 Universal Disk Controller (UDC). Figures 4 and 5 illustrate UIPC interconnections in two UDC installations.

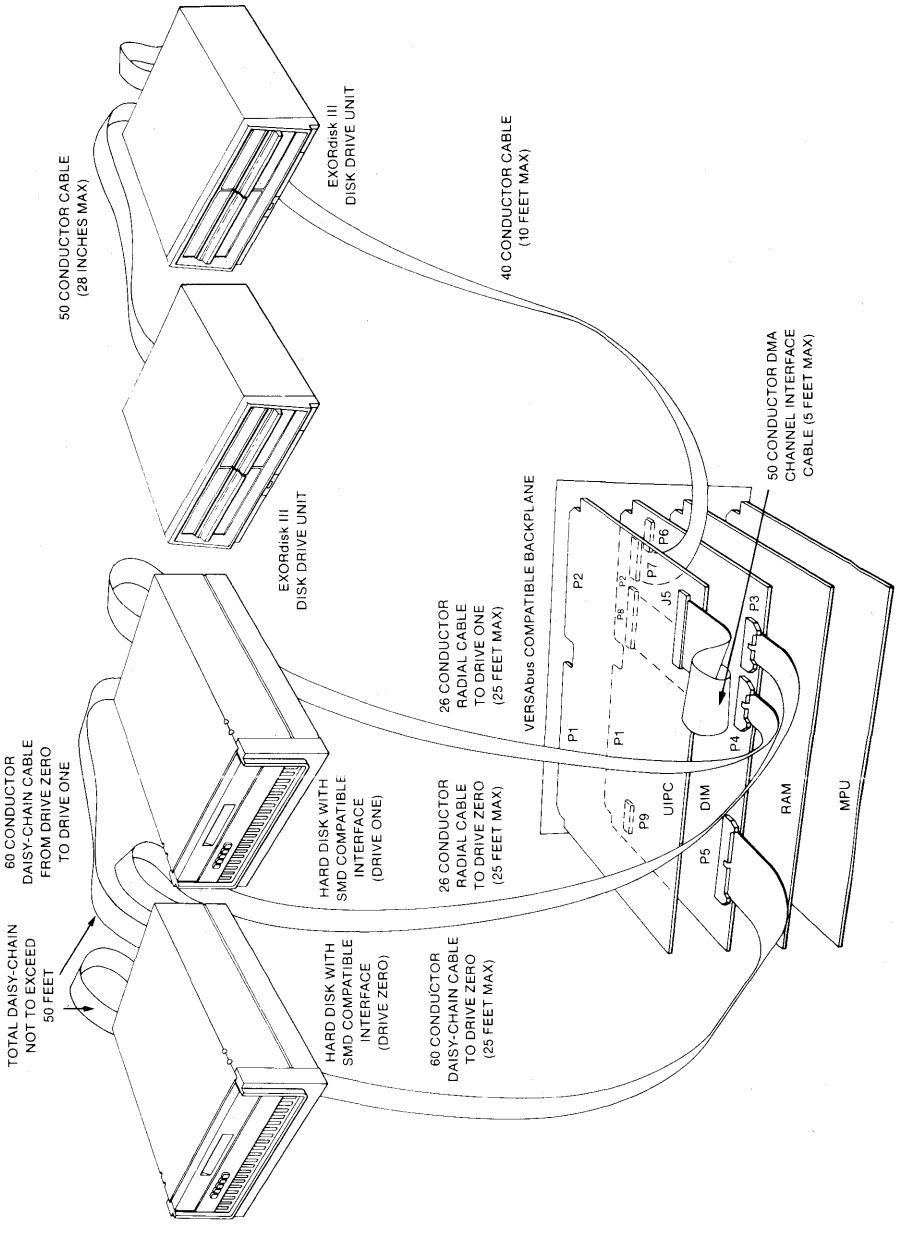


FIGURE 4 — UIPC Interconnection in a Typical Installation of the Universal Disk Controller

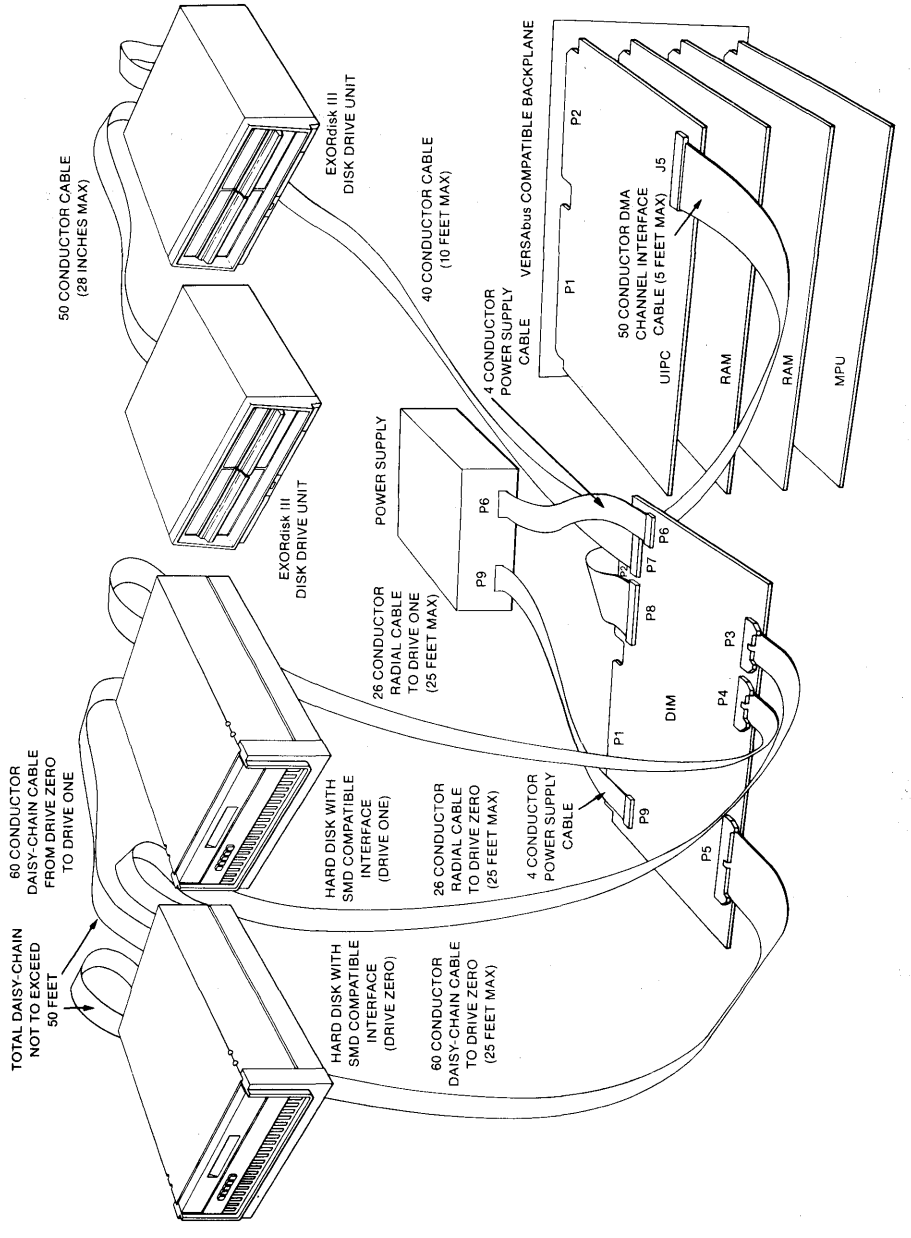


FIGURE 5 — UIPC Interconnection in a Typical Installation of the Universal Disk Controller Utilizing Standalone DIM

TABLE 6 — Specifications

Characteristics	Specifications
VERSAbus Command Channel	128 byte shared RAM allows communication between the UIPC and the HOST.
VERSAbus Data Channel	VERSAbus data transfers performed via DMA at word rates of up to 4 Mwords (8 Mb)/sec., max., with actual rate determined by cycle time of the addressed VERSAbus memory board. With the M68KVM10 128K RAM module the maximum transfer rate is approximately 1.5 Mwords/sec.
FIFO Data Buffer	32 byte FIFO used to regulate transfer of data to asynchronous system bus.
VERSAbus Master Retention Time	During DMA, bus mastership retained for approximately 8 μs, max., per data transfer burst before release to other masters.
UIPC Power Requirements (Typical)	+5 Vdc ±5% @ 7.0 A +12 Vdc ±5% @ 50 mA −12 Vdc ±5% @ 50 mA
Operating Temperature Range Storage Temperature Range	0° to +65°C −40° to +85°C
Relative Humidity	0 to 90% (non-condensing)
Dimensions length x width x height (including components)	14.50 x 9.25 x 0.60 inches

Ordering Information

Part Number	Description
M68KVM60	VERSAmodule Universal Intelligent Peripheral Controller (UIPC)

Documentation

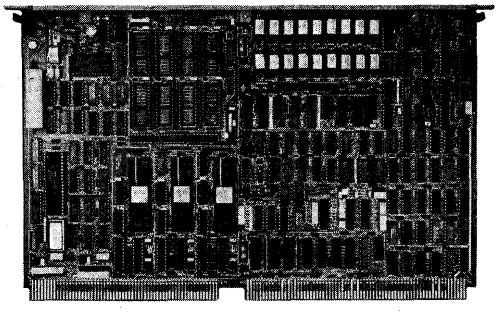
M68KVM60/D2	Universal Intelligent Peripheral Controller Module User's Manual
M68KIPCS	M68000/IPC Command Channel Software Interface Reference Manual
M68KDRVGD	Guide to Writing Device Drivers for VERSAdos

Related Documentation

Part Number	Description
M68KVBS	VERSAbus Specification Manual

M68KVM80-1
M68KVM80-4

VERSAmodule Combination Memory, I/O and Time-Of-Day Clock



- VERSAbus Compatible
- 0/128K Byte Dynamic RAM Options
- Eight 28-Pin Sockets For Up To 256K Bytes of User Provided 2K, 4K, 8K, 16K or 32K Byte ROM/PROM/EPROM Devices.
- Two Multiprotocol Serial I/O Ports with RS-232C Asynchronous and Synchronous Interface, Individually Selectable for Modem or Terminal Use
- Six Parallel I/O Ports
- Three 24-Bit Programmable Timer/Counters
- CMOS Time-of-Day Clock with Battery Backup
- Test Register and Board Fault Indicator
- 0°C-70°C Operating Temperature Range

This VERSAmodule Combination Memory-I/O-Time-of-Day Clock board provides a cost effective solution for expansion of these basic features in a VERSAbus based system. It is complementary for modular systems based on the VERSAmodule 01A and 02 monoboard microcomputers, or other VERSAbus compatible monoboards. The module can also provide convenient memory expansion, along with serial and parallel I/O, for additional functions in the VMC 68/2 Microcomputer System (which incorporates the VM02, processor module).

The serial ports can provide suitable expansion for serial oriented devices such as user terminals and serial interface printers. They alternately can provide a high speed serial link operating with modular bit oriented protocols in various networking configurations. The parallel ports provide the interface required for adding medium to high speed printers to a VERSAbus based system. Industrial control interfaces requiring high speed parallel transfers are also accommodated by the parallel ports.

The combination of features and capabilities included on the VM80 module means economical functional expansion in VERSAmodule based systems applied to a wide spectrum of uses in industrial control, laboratory automation, communications and many other applications. The module is available in two versions — the first with I/O only and the other with 128K bytes of RAM with I/O. The user may add ROM/PROM/EPROM devices to either version to provide from 4K bytes to 256K bytes of read-only memory.

Memory

The 28-pin ROM/PROM/EPROM sockets together with the proper jumper selection permits the use of 24-pin 2716/2732 compatible devices as well as 28-pin 16K and 32K byte devices. Asynchronous Data Acknowledge (DTACK) response time jumper selection is provided to permit the use of a full range of devices having access times from 30 ns to 450 ns.

RAM and ROM/PROM/EPROM may each be placed on any even byte boundary of their respective device implementation size (i.e., 16, 32, 64, 128 or 256K byte boundary) throughout the 16 megabyte VERSAbus space via strap option. On the 128K byte RAM version, byte parity with automatic retry is a jumper option. If the retry is unsuccessful the module will generate a VERSAbus Bus Error (BERR) and transfer it to the system VERSAbus. RAM may be strapped to operate from VERSAbus +5 Vdc standby power for external battery backup. Power fail write inhibit logic is included. The RAM module may be write-protected when presented with certain address modifier codes. The RAM section could appear as read-write while in the supervisor state, and be write-protected in the user state. An attempt to write into RAM that has been write-protected will result in a bus error (BERR).

Serial Ports

Two multiprotocol serial I/O ports with RS-232 interface selectable by jumper option for modem or terminal use are contained on the VM80. The serial ports are controlled by the NEC uPD7201 Multiprotocol Serial Communications Controller. Asynchronous and synchronous byte-

oriented protocols (including IBM Bisync) as well as SDLC and HDLC bit oriented protocols are supported. The serial ports can operate on internal clock rates from 50 bits-per-second (bps) to 19.2K bps, and on external clock rates up to 600K bps. All I/O device bus addresses are located in a 512 byte block which may be placed on any even 512 byte boundary in the upper 64K bytes of VERSAbus space except the top 512 byte block via strap option.

Parallel Ports

Six parallel I/O ports are provided on the VM80 by three MC68230 Parallel Interface/Timer (PI/T) M68000 family support chips. Each of the six ports have eight data and two handshake lines and each pair is programmable as 8 or 16-bit ports which gives the option of three 16-bit ports. The ports can also be programmed as either unidirectional or bidirectional.

Timer/Counters

Three 24-bit programmable timer/counters are provided with the MC68230 PI/T chips. Each timer contains a 24-bit counter which is loaded by three 8-bit counter preload registers. The timers can be programmed to create a VERSAbus interrupt, or to issue an output to an external device. The inputs of the timer/counter can be individually selected by jumpers to one of the following sources:

- Serial port baud rate clocks
- 250 kHz clock
- VERSAbus ac line clocks
- Square wave clock from time-of-day clock
- External input

Time-of-Day Clock

The time-of-day clock is implemented by an MC146818 time-of-day clock chip and provides month/day/hour/minute/second information. Battery backup for 100 hours (typical) operation during a power outage is provided by an on-board nickel-cadium battery with recharge circuitry operational during normal operation. Strap option permits operation from VERSAbus +5 Vdc standby power for external battery backup.

Test Register

The VM80 contains a test register and board fault indicator (LED) which allows user supplied test routines to indicate the status of the board. The VERSAbus has four signal lines to support testing of the system. The signals involved are SYSRESET*, TEST0*, TEST1*, and SYSFAIL*. The TEST0* and TEST1* lines are driven by some other board in the system. They tell the VM80 whether or not a test will be performed on it.

The test lines (TEST0*, TEST1*) are latched upon the release of SYSREST* (low-to-high transistion). If either test line is in the low state, VM80 will assume a test is about to be performed and will drive SYSFAIL* low and light the board FAIL LED. At this point, the user's test program must determine the functionality of the board and make a pass/fail decision. If the board passes the test, the user writes to the test register, which releases SYSFAIL*, and turns off the FAIL LED. If, however, the board does not pass the test, or cannot be tested, the SYSFAIL* signal will remain low and board FAIL LED will remain illuminated.

VERSAbus Interface

A VERSAbus interface is incorporated in the combination module. VERSAbus is characterized by asynchronous, bidirectional operation and supports the full 16-megabyte address range of the MC68000 MPU. All VERSAbus data, address, and control lines for 16-bit system applications are present on the 140-pin connector, P1. All I/O lines for the combination module supporting the serial I/O, parallel I/O and timer functions, are present on the 120-pin connector, P2 (see block diagram).

VERSAbus interrupter logic permits each of the on-board I/O devices to place an interrupt request on one of the seven VERSAbus interrupt request priority lines. All on-board I/O devices may be individually programmed to any of the seven VERSAbus interrupt request levels and provide individually programmable interrupt vector numbers to VERSAbus.

Related Documentation

Documentation related to the VM80 is listed below:

VERSAbus Specification Manual M68KVBS
VERSAmodule Chassis, Card Cage, Power Supply, and Power Monitor User's Manual, M68KVMESH
VERSAmodule 02 Monoboard Microcomputer User's Manual, M68KVM02
NEC uPD7201 Multiprotocol Serial Communication's Controller User's Guide (included with M68KVM80 User's Guide)
MC146818 Time-of-Day Clock Chip Data Sheet
MC68230L8, L10 Parallel Interface/Timer (PI/T) Data Sheet

TABLE 1 — VERSAmodule 80 Specificatons

Characteristics	Specifications
Power Requirements M68KVM80-1 M68KVM80-4	 (See Table 2 for details.) (See Table 3 for details.)
Memory Organization Dynamic RAM	 M68KVM80-1: No RAM available M68KVM80-4: 128K byte RAM Parity check jumper selectable.
ROM/PROM/EPROM (User-supplied)	Eight 28-pin sockets on-board for 2K, 4K, 8K, 16K byte devices using +5 Vdc only. Access times jumper selectable from 30 ns to 450 ns, in 20-ns increments.
Serial Ports	Two RS-232C ports selectable for connection to terminal or to modem. Can operate synchronous or asynchronous. Transmit and receive clocks can be supplied externally or internally by baud rate generator. Baud rates selectable as follows: Asynchronous mode 50 — 19.2K Synchronous mode 800 — 316.8K
Parallel Ports	Six 8-bit ports, with two handshake lines each. (Can be used as three 16-bit ports.)
Timer	Three 24-bit programmable timers. Inputs individually selectable.
Time-of-Day Clock	Year, month, day, hour, minute, second data. Periodic interrupts and alarm. On-board battery provides standby power for three days (minimum).
Device Access Timing	(See Table 4 for details.)
Interrupts	Each peripheral device may generate one or more interrupts: Serial Port Interrupter Timer 1 Interrupter Timer 2 Interrupter Timer 3 Interrupter Parallel Port 1 Interrupter Parallel Port 2 Interrupter Parallel Port 3 Interrupter Time-of-Day Clock Interrupter Each interrupter can be programmed for VERSAbus interrupt levels 1 through 7. A unique interrupt vector can be programmed for each device.
Test	A test register and board fail LED are included for indicating the results of a user created test routine.
Bus Mating Connector Types VERSAbus Connector (P1) I/O Connector (P2)	 Stanford Applied Eng'g CPH7000-140ST Micro Plastics, Inc. MP-0100-70-DW-5H Stanford Applied Eng'g CPH7000-120ST Micro Plastics, Inc. MP-0100-60-DW-5H

TABLE 1 — VERSAmodule 80 Specifications (continued)

Characteristics	Specifications
Operating Temperature Range	0° to 70° C
Storage Temperature Range	-40° C to +85° C
Relative Humidity	0% to 90% (non-condensing)
Physical Characteristics Height Width Length	 0.6 inch (1.5 cm) 14.50 inches (36.83 cm) 9.25 inches (23.5 cm)
Supply Voltages	(a) +5 V ±5% (b) +12 V ±5% (c) +12 V ±5%

TABLE 2 — VM80-1 Power Requirements

	+5 Vdc		+12 Vdc		-12 Vdc		+5 V Stdby	
Section (Module) of Board	**Typ**	**Max**	**Typ**	**Max**	**Typ**	**Max**	**Typ**	**Max**
Board — without ROM	4.25 A	5.10 A	55 mA	100 mA	40 mA	80 mA	—	—
Add for each PROM pair (up to four pairs)	100 mA	300 mA	—	—	—	—	—	—

TABLE 3 — VM80-4 Power Requirements

	+5 Vdc		+12 Vdc		-12 Vdc		+5 V Stdby	
Section (Module) of Board	**Typ**	**Max**	**Typ**	**Max**	**Typ**	**Max**	**Typ**	**Max**
RAM — powered from +5 V standby, without ROM	4.25 A	5.10 A	55 mA	100 mA	40 mA	80 mA	—	—
Active Inactive							430 mA 320 mA	500 mA 360 mA
Add for each ROM pair (up to four pairs)	100 mA	300 mA	—	—	—	—	—	—

NOTE: In Tables 2 and 3, the currents at the +12 Vdc and -12 Vdc are specified for the serial port signals not loaded. The actual required values depend upon the load of the RS-232C ports. Each fully loaded serial port signal may add up to 12 mA to these currents.

VM80 Timing Information

Access times for the VM80 are given in the following table.

TABLE 4 — VM80 Access Times

VM80 Module Element	Typical Access Time (in Nanoseconds)	
	Read Cycle	Write Cycle
RAM (200 ns RAM's)		
Parity enabled	510	210
Parity disabled	440	210
ROM (headers J2/J3)	70 (1)	(Bus error)
Time-of-day clock	760	760
Serial ports	580	580
Parallel timers 1, 2, 3	310	310
Interrupt control registers	1,000	1,000
Station address register	130	(Bus error)
Test register	110	110

NOTE: (1) Plus delay time obtained by configuring headers J2 and J3.

VM80 — Memory Map

	UPPER BYTE	LOWER BYTE		
$FF01FE		49 CONTIGUOUS BYTES OF CMOS RAM (1)	$FF01FF	
$FF019E			$FF019F	
$FF019C		TDC INTERRUPT VECTOR REG (1)	$FF019D	
$FF019A		REGISTER D (1)	$FF019B	
$FF0198		REGISTER C (1)	$FF0199	
$FF0196		REGISTER B (1)	$FF0197	
$FF0194		REGISTER A (1)	$FF0195	
$FF0192		YEAR REGISTER (1)	$FF0193	
$FF0190		MONTH REGISTER (1)	$FF0191	TIME-OF-DAY
$FF018E		DATE OF MONTH REGISTER (1)	$FF018F	CLOCK
$FF018C		DAY OF WEEK REGISTER (1)	$FF018D	
$FF018A		HOURS ALARM REGISTER (1)	$FF018B	
$FF0188		HOURS REGISTER (1)	$FF0189	
$FF0186		MINUTES ALARM REGISTER (1)	$FF0187	
$FF0184		MINUTES REGISTER (1)	$FF0185	
$FF0182		SECONDS ALARM REGISTER (1)	$FF0183	
$FF0180		SECONDS REGISTER (1)	$FF0181	
$FF017E		TEST REGISTER (1)	$FF017F	
$FF017C		TEST REGISTER (1)	$FF017D	
$FF017A	ILLEGAL (3)	TEST REGISTER (1)	$FF017B	
$FF0178		TEST REGISTER (1)	$FF0179	TEST
$FF0176		TEST REGISTER (1)	$FF0177	
$FF0174		TEST REGISTER (1)	$FF0175	
$FF0172		TEST REGISTER (1)	$FF0173	
$FF0170		TEST REGISTER	$FF0171	
$FF016E		SERIAL PORTS ICR	$FF016F	
$FF016C		TIMER 3 ICR	$FF016D	
$FF016A		PARALLEL PORT 3 ICR	$FF016B	INTERRUPT
$FF0168		TIMER 2 ICR	$FF0169	CONTROL
$FF0166		PARALLEL PORT 2 ICR	$FF0167	REGISTERS
$FF0164		TIMER 1 ICR	$FF0165	
$FF0162		PARALLEL PORT 1 ICR	$FF0163	
$FF0160		TIME OF DAY CLOCK ICR	$FF0161	
$FF015E		STATION ADDRESS REGISTER (1)	$FF015F	
$FF015C		STATION ADDRESS REGISTER (1)	$FF015D	STATION
$FF015A		STATION ADDRESS REGISTER (1)	$FF015B	ADDRESS
$FF0158		STATION ADDRESS REGISTER (1)	$FF0159	REGISTER
$FF0154		STATION ADDRESS REGISTER (1)	$FF0155	
$FF0152		STATION ADDRESS REGISTER (1)	$FF0153	
$FF0150		STATION ADDRESS REGISTER	$FF0151	
$FF014E		SERIAL CHANNEL B CONTROL (1)	$FF014F	
$FF014C		SERIAL CHANNEL A CONTROL (1)	$FF014D	
$FF014A		SERIAL CHANNEL B DATA (1)	$FF014B	
$FF0148		SERIAL CHANNEL A DATA (1)	$FF0149	SERIAL PORTS
$FF0146		SERIAL CHANNEL B CONTROL	$FF0147	1 AND 2
$FF0144		SERIAL CHANNEL A CONTROL	$FF0145	
$FF0142		SERIAL CHANNEL B DATA	$FF0143	
$FF0140		SERIAL CHANNEL A DATA	$FF0141	
$FF013E		NULL (2)	$FF013F	
$FF013C		NULL (2)	$FF013D	
$FF013A		NULL (2)	$FF013B	PARALLEL
$FF0138		NULL (2)	$FF0139	INTERFACE/
$FF0136		NULL (2)	$FF0137	TIMER 3
$FF0134		TIMER 3 STATUS REGISTER	$FF0135	

VM80 — Memory Map (continued)

	UPPER BYTE	LOWER BYTE		
$FF0132		COUNT REGISTER LOW	$FF0133	
$FF0130		COUNT REGISTER MID	$FF0131	
$FF012E		COUNT REGISTER HIGH	$FF012F	
$FF012C		NULL (2)	$FF012D	
$FF012A		COUNT PRELOAD REGISTER LOW	$FF012B	
$FF0128		COUNT PRELOAD REGISTER MID	$FF0129	
$FF0126		COUNT PRELOAD REGISTER HIGH	$FF0127	
$FF0124		NULL (2)	$FF0125	
$FF0122		TIMER 3 INTERRUPT VECTOR	$FF0123	
$FF0120		TIMER 3 CONTROL REGISTER	$FF0121	
$FF011E		NULL (2)	$FF011F	
$FF011C		NULL (2)	$FF011D	PARALLEL
$FF011A	ILLEGAL (3)	PORT 3 STATUS REGISTER	$FF011B	INTERFACE/
$FF0118		PORT 3C DATA REGISTER	$FF0119	TIMER 3
$FF0136		PORT 3B ALTERNATE REG	$FF0137	(CONT'D)
$FF0114		PORT 3A ALTERNATE REG	$FF0115	
$FF0112		PORT 3B DATA REGISTER	$FF0113	
$FF0110		PORT 3A DATA REGISTER	$FF0111	
$FF010E		PORT 3B CONTROL REGISTER	$FF010F	
$FF010C		PORT 3A CONTROL REGISTER	$FF010D	
$FF010A		PORT 3 INTERRUPT VECTOR	$FF010B	
$FF0108		PORT 3C DATA DIRECTION	$FF0109	
$FF0106		PORT 3B DATA DIRECTION	$FF0107	
$FF0104		PORT 3A DATA DIRECTION	$FF0105	
$FF0102		PORT 3 SERVICE REQUEST	$FF0103	
$FF0100		PORT 3 GENERAL CONTROL	$FF0101	
$FF00FE		NULL (2)	$FF00FF	
$FF00FC		NULL (2)	$FF00FD	
$FF00FA		NULL (2)	$FF00FB	
$FF00F8		NULL (2)	$FF00F9	
$FF00F6		NULL (2)	$FF00F7	
$FF00F4		TIMER 2 STATUS REGISTER	$FF00F5	
$FF00F2		COUNT REGISTER LOW	$FF00F3	
$FF00F0		COUNT REGISTER MID	$FF00F1	
$FF00EE		COUNT REGISTER HIGH	$FF00EF	
$FF00EC		NULL (2)	$FF00ED	
$FF00EA		COUNT PRELOAD REGISTER LOW	$FF00EB	
$FF00E8		COUNT PRELOAD REGISTER MID	$FF00E9	
$FF00E6		COUNT PRELOAD REGISTER HIGH	$FF00E7	
$FF00E4		NULL (2)	$FF00E5	PARALLEL
$FF00E2		TIMER 2 INTERRUPT VECTOR	$FF00E3	INTERFACE/
$FF00E0	ILLEGAL (3)	TIMER 2 CONTROL REGISTER	$FF00E1	TIMER 2
$FF00DE		NULL (2)	$FF00DF	
$FF00DC		NULL (2)	$FF00DD	
$FF00DA		PORT 2 STATUS REGISTER	$FF00DB	
$FF00D8		PORT 2C DATA REGISTER	$FF00D9	
$FF00D6		PORT 2B ALTERNATE REGISTER	$FF00D7	
$FF00D4		PORT 2A ALTERNATE REGISTER	$FF00D5	
$FF00D2		PORT 2B DATA REGISTER	$FF00D3	
$FF00D0		PORT 2A DATA REGISTER	$FF00D1	
$FF00CE		PORT 2B CONTROL REGISTER	$FF00CF	
$FF00CC		PORT 2A CONTROL REGISTER	$FF00CD	
$FF00CA		PORT 2 INTERRUPT VECTOR	$FF00CB	PARALLEL
$FF00C8		PORT 2C DATA DIRECTION	$FF00C9	INTERFACE/
$FF00C6		PORT 2B DATA DIRECTION	$FF00C7	TIMER 2
$FF00C4		PORT 2A DATA DIRECTION	$FF00C5	(CONT'D)
$FF00C2		PORT 2 SERVICE REQUEST	$FF00C3	
$FF00C0		PORT 2 GENERAL CONTROL	$FF00C1	

VM80 — Memory Map (continued)

	UPPER BYTE	LOWER BYTE		
$FF00BE		NULL (2)	$FF00BF	
$FF00BC		NULL (2)	$FF00BD	
$FF00BA		NULL (2)	$FF00BB	
$FF00B8		NULL (2)	$FF00B9	
$FF00B6		NULL (2)	$FF00B7	
$FF00B4		TIMER 1 STATUS REGISTER	$FF00B5	
$FF00B2		COUNT REGISTER LOW	$FF00B3	
$FF00B0		COUNT REGISTER MID	$FF00B1	
$FF00AE		COUNT REGISTER HIGH	$FF00AF	
$FF00AC		NULL (2)	$FF00AD	
$FF00AA		COUNT PRELOAD REGISTER LOW	$FF00AB	
$FF00A8		COUNT PRELOAD REGISTER MID	$FF00A9	
$FF00A6		COUNT PRELOAD REGISTER HIGH	$FF00A7	
$FF00A4		NULL (2)	$FF00A5	
$FF00A2		TIMER 1 INTERRUPT VECTOR	$FF00A3	PARALLEL
$FF00A0	ILLEGAL (3)	TIMER 1 CONTROL REGISTER	$FF00A1	INTERFACE/
$FF009E		NULL (2)	$FF009F	TIMER 1
$FF009C		NULL (2)	$FF009D	
$FF009A		PORT 1 STATUS REGISTER	$FF009B	
$FF0098		PORT 1C DATA REGISTER	$FF0099	
$FF0096		PORT 1B ALTERNATE REGISTER	$FF0097	
$FF0094		PORT 1A ALTERNATE REGISTER	$FF0095	
$FF0092		PORT 1B DATA REGISTER	$FF0093	
$FF0090		PORT 1A DATA REGISTER	$FF0091	
$FF008E		PORT 1B CONTROL REGISTER	$FF008F	
$FF008C		PORT 1A CONTROL REGISTER	$FF008D	
$FF008A		PORT 1 INTERRUPT VECTOR	$FF008B	
$FF0088		PORT 1C DATA DIRECTION	$FF0089	
$FF0086		PORT 1B DATA DIRECTION	$FF0087	
$FF0084		PORT 1A DATA DIRECTION	$FF0085	
$FF0082		PORT 1 SERVICE REQUEST	$FF0083	
$FF0080		PORT 1 GENERAL CONTROL	$FF0081	
$FF007E		49 CONTIGUOUS BYTES OF CMOS RAM	$FF007F	
$FF001E			$FF001F	
$FF001C		TDC INTERRUPT VECTOR REG	$FF001D	
$FF001A		REGISTER D	$FF001B	
$FF0018		REGISTER C	$FF0019	
$FF0016		REGISTER B	$FF0017	
$FF0014		REGISTER A	$FF0015	
$FF0012		YEAR REGISTER	$FF0013	TIME-OF-DAY
$FF0010	ILLEGAL (3)	MONTH REGISTER	$FF0011	CLOCK
$FF000E		DATE OF MONTH REGISTER	$FF000F	AND RAM
$FF000C		DAY OF WEEK REGISTER	$FF000D	
$FF000A		HOURS ALARM REGISTER	$FF000B	
$FF0008		HOURS REGISTER	$FF0009	
$FF0006		MINUTES ALARM REGISTER	$FF0007	
$FF0004		MINUTES REGISTER	$FF0005	
$FF0002		SECONDS ALARM REGISTER	$FF0003	
$FF0000		SECONDS REGISTER	$FF0001	

NOTES:
1. Redundant Location
2. Null Locations — The null register returns all zeroes for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle, but no write occurs.
3. Upper byte access is considered illegal. Specifically, any write operation will write the value of D00-D07 into the addressed location. A bus error is not issued if an attempt is made to access the upper byte.
4. The addresses shown on this map are from $FF0000-$FF01FF. The map occupies 512 bytes, 256 upper bytes and 256 lower bytes. The entire map may be moved, as a group, to reside on any 512 byte boundary between $FF0000 through $FFFE00.

VM80 VERSAbus I/O Connector P2 Signals

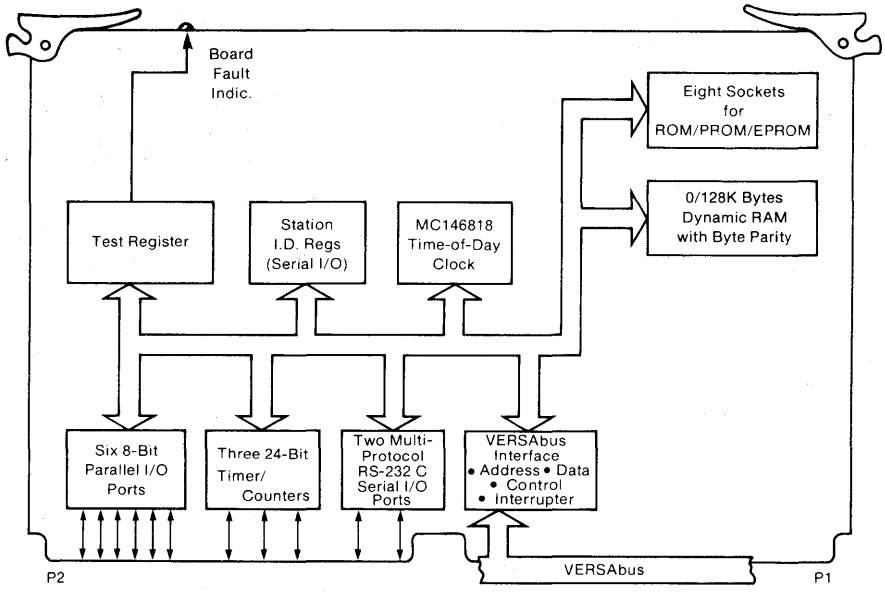
SIGNAL MNEMONIC	PIN NUMBER	TO TERMINAL	TO MODEM	SIGNAL NAME AND FUNCTION
CLOCK1*	77			CLOCK1 — May be used as an input clock for timer 1.
CLOCK2*	114			CLOCK 2 — May be used as an input clock for timer 2.
CLOCK3*	60			CLOCK 3 — May be used as an input clock for timer 3.
CTS1 CTS2	25 53	OUTPUT OUTPUT	INPUT INPUT	CLEAR TO SEND — Indicates that terminal may transmit data.
DCD1 DCD2	31 59	OUTPUT OUTPUT	INPUT INPUT	DATA CARRIER DETECT — Indicates to terminal that a suitable data carrier is present.
DSR1 DSR2	27 54	OUTPUT OUTPUT	INPUT INPUT	DATA SET READY — Indicates that the data set (modem) is ready. It is in the "off-hook" state in switched service, and not in the test, talk, or dial mode.
DTR1 DTR2	33 61	INPUT INPUT	OUTPUT OUTPUT	DATA TERMINAL READY — Indicates that data terminal is ready to transmit or receive data. The on-to-off transition will signal the modem to "hang-up" the line.
GATE1*	75			GATE 1 — May be used to inhibit CLOCK1*.
GATE2*	116			GATE 2 — May be used to inhibit CLOCK2*.
GATE3*	62			GATE 3 — May be used to inhibit CLOCK3*.
GND	1-6,29, 57,97,98, 101,102			GROUND
GND (±15 V)				Not used.
OUTPUT1	73			TIMER 1 OUTPUT — The Timer 1 output may be selected to appear on this signal.
OUTPUT2	118			TIMER 2 OUTPUT — The Timer 2 output may be selected to appear on this signal.
OUTPUT3	64			TIMER 3 OUTPUT — The Timer 3 output may be selected to appear on this signal.
P1PA0- P1PA7	117,115, 113,111, 109,107, 105,103			PARALLEL PORT 1 PERIPHERAL DATA, A SIDE (bits 0-7) — Buffered I/O data lines.
P1PB0- P1B07	101,99, 97,95,93, 91,89,87			PARALLEL PORT 1 PERIPHERAL DATA, B SIDE (bits 0-7) — Buffered I/O data lines.
P1CA1	85			PARALLEL PORT 1 CONTROL, A SIDE — Input signal connected to port 1 H1.
P1CA2	83			PARALLEL PORT 1 CONTROL, A SIDE — Output signal connected to port 1 H2.

VM80 VERSAbus I/O Connector P2 Signals (continued)

SIGNAL MNEMONIC	PIN NUMBER	TO TERMINAL	TO MODEM	SIGNAL NAME AND FUNCTION
P1CB1	81			PARALLEL PORT 1 CONTROL, B SIDE — Input signal connected to port 1 H3.
P1CB2	79			PARALLEL PORT 1 CONTROL, B SIDE — Output signal to port 1 H4.
P2PA0- P2PA7	74,76,78, 80,82,84, 86,88			PARALLEL PORT 2 PERIPHERAL DATA, A SIDE (bits 0-7) — Buffered I/O data lines.
P2PB0- P2PB7	90,92,94, 96,98,100, 102,104			PARALLEL PORT 2 PERIPHERAL DATA, B SIDE (bits 0-7) — Buffered I/O data lines.
P2CA1	106			PARALLEL PORT 2 CONTROL, A SIDE — Input signal connected to port 2 H1.
P2CA2	108			PARALLEL PORT 2 CONTROL, A SIDE — Output signal connected to port 2 H2.
P2CB1	110			PARALLEL PORT 2 CONTROL, B SIDE — Input signal connected to port 2 H3.
P2CB2	112			PARALLEL PORT 2 CONTROL, B SIDE — Output signal connected to port 2 H4.
P3PA0- P3PA7	20,22,24, 26,28,30, 32,34			PARALLEL PORT 3 PERIPHERAL DATA, A SIDE (bits 0-7) — Buffered I/O data lines.
P3PB0- P3PB7	36,38,40, 42,44,46, 48,50			PARALLEL PORT 3 PERIPHERAL DATA, B SIDE (bits 0-7) — Buffered I/O data lines.
P3CA1	52			PARALLEL PORT 3 CONTROL, A SIDE — Input signal connected to port 3 H1.
P3CA2	54			PARALLEL PORT 3 CONTROL, A SIDE — Output signal connected to port 3 H2.
P3CB1	56			PARALLEL PORT 3 CONTROL, B SIDE — Input signal connected to port 3 H3.
P3CB2	58			PARALLEL PORT 3 CONTROL, B SIDE — Output signal connected to port 3 H4.
RTS1 RTS2	23 51	INPUT INPUT	OUTPUT OUTPUT	REQUEST TO SEND — Indicates that terminal wishes to send data. On a half duplex channel, this signal controls direction of data transmission.
RXC1 RXC2	35 63	OUTPUT OUTPUT	INPUT INPUT	RECEIVE CLOCK — May be an output from baud rate generator or an input clock for transmitter or receiver.
RXD1 RXD2	21 49	OUTPUT OUTPUT	INPUT INPUT	RECEIVE DATA — Used for receive data as an input, or transmit data as an output.

VM80 VERSAbus I/O Connector P2 Signals (continued)

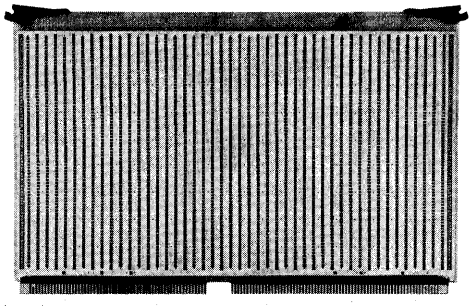
SIGNAL MNEMONIC	PIN NUMBER	TO TERMINAL	TO MODEM	SIGNAL NAME AND FUNCTION
TXC1 TXC2	37 65	OUTPUT OUTPUT	INPUT INPUT	TRANSMIT CLOCK — May be an output from baud rate generator or an input clock for transmitter or receiver.
TXD1 TXD2	19 47	INPUT INPUT	OUTPUT OUTPUT	TRANSMIT DATA — Used for transmit data as an output, or receive data as an input.
+5 V	7-10			+5 Vdc POWER — Used by VM80 logic circuits.
+5 VOUTA	39			+5 Vdc POWER — Jumper selectable for I/O.
+5 VOUTB	71,119			+5 Vdc POWER — Jumper selectable for I/O.
+5 VOUTC	18,66			+5 Vdc POWER — Jumper selectable for I/O.
+5 VOUTD	72,120			+5 Vdc POWER — Jumper selectable for I/O.
-12 V	15,16			-12 Vdc POWER — Used by VM80 logic and interface circuits.
-12 VOUTA	41			-12 Vdc POWER — Jumper selectable for I/O.
+12 V	11,12			+12 Vdc POWER — Used by VM80 logic and interface circuits.
+12 VOUTA	43			+12 Vdc POWER — Jumper selectable for I/O.
-15 V	67,68			-15 Vdc POWER — Not used.
+15 V	17,45 69,70			+15 Vdc POWER — Not used.



Ordering Information

Part Number	Description
M68KVM80-1	VERSAmodule Combination Memory, I/O and Time-of-Day Clock. This module contains no Dynamic RAM. The module contains eight sockets for ROM/PROM/EPROM, six 8-bit Parallel Ports, three Timer/Counters, two Serial I/O ports and a battery backed-up Time-of-Day Clock. Includes User's Manual.
M68KVM80-4	Same as M68KVM80-1, but includes 128K bytes of Dynamic Ram.
M68KVM80/D1	Combination Memory, I/O, and Time-of-Day Clock User's Manual. Includes NEC uPD7201 Multiprotocol Serial Communications Controller Manual.
M68KVBS	VERSAbus Specification Manual.

VERSAbus Wirewrap Module



- Standard pin spacing for 14, 16, 18, 24, 40 and 64-pin Wirewrap Sockets
- Positions for four axial-lead type bulk filter capacitors
- Provision for decoupling capacitors
- Provision for ribbon cable connectors
- Card ejector ears included for ease of insertion and removal
- Silk screen marking

The VERSAbus Wirewrap Module permits the user to construct and incorporate his custom circuits into an EXORmacs Development System, VMC 68/2 Microcomputer System, or any other VERSAbus application. Incorporated on the module are the power bus and ground bus printed wiring runs. On the component or front side of the board are metalized ground strips. On the solder side are alternating rows of ground and +5 volt strips, with ground strips surrounding the board edge as a safety feature. A silk screen marking process shows edge connector numbers and makes the matrix more visible for location of positions. The module has standard pin spacing and provisions for 14, 16, 18, 24, 40 and 64-pin wirewrap sockets; or all dual in-line packages with 3/10 centers or their multiples. The four rows of undedicated holes at the top of the board can be used as wirewrap area, wirewrap connectors, ribbon cable connectors, switch and jumper locations, or any similar function the user requires.

Ordering Information

Part Number	Description
M68KWW	VERSAbus Wirewrap Module

M68K2RBBUG4

020bug Debugger/ Diagnostic Package

020bug RESIDENT PACKAGE

- EPROM Resident Debug and Diagnostic Monitor
- Full Speed Execution of System and User-developed Programs Operating out of the Benchmark 20 System Package
- Powerful Debugger Command Set Allows Access to all (VM04) I/O, Control, and Memory Facilities Plus the Full 4 Gigabyte Direct Address Range of the VERSAbus System Bus
- Virtual Terminal Capability for Up/downline Load from an EXORmacs or VME/10 Development System or from a Cross-computer
- One-line Assembler/Disassembler
- User-callable Routines for I/O, Data Conversion and More
- Complete VM04 Hardware Diagnostics with Loop Continuous and Loop-on-error Modes
- Self-test on Power-up Feature Verifies Integrity of System
- Dual Port RS-232C Distribution Board (MVMCH3-104) Allows Connection of Debug Terminal and Up/downline Load Host to the VERSAmodule Benchmark 20 System Package through the RS-232C Ports on the VM04

020bug SOURCE AND RELOCATABLE OBJECT MODULE PACKAGE

- Source and Relocatable Object Modules for 020bug on Diskette or Cartridge Disk
- Relocatable Object Modules Allow Users to Include Only the 020bug Items Needed in Their End System; to Link in Their Own Up/downline Loader; and/or to Link in Their Own Bootstrap Loader
- Source Modules Allow User Modification of 020bug as Desired

The VERSAmodule Benchmark 20 System debug package, 020bug, is available as two separate product offerings. 020bug comes as an EPROM-based resident package consisting of two 32Kb EPROMs designed to plug into the EPROM sockets of the M68KVM04 VERSAmodule 32-bit Monoboard Microcomputer (VM04). This resident package includes a RS-232C Distribution Board (MVMCH3-104) for connecting two RS-232C devices with standard DB-25 connectors to the 50-pin serial I/O connector on the VM04. The resident package is ready for installation and immediate use with the VM04 Monoboard Microcomputer in a system built around a VERSAbus backplane. (Such a backplane is provided by Motorola's MVMCC3 VERSAmodule Card Cage or MVMCH3 VERSAmodule Chassis.) 020bug Source and Relocatable Object Modules are also available as a separate product on either EXORmacs Development System-compatible diskette or disk cartridge.

020bug requires 8Kb of RAM storage for exception vectors and stack/work space. At power-up, 020bug searches the system memory space from low to high and allocates the 8Kb of space for its own use and another 8Kb for user program space in the lowest RAM memory found. In systems with multiple VM04 boards, each 020bug will allocate its own 16Kb segment of memory. If no RAM is found, then 020bug will disable the VM04's 16Kb onboard cache and allocate the cache memory, allowing the debugger to operate on a stand-alone VM04 board.

020bug provides a powerful evaluation and system debugging tool for VERSAmodule Systems. The debugger permits full speed execution of system and user-developed programs operated in a VERSAmodule Monoboard Microcomputer system environment under complete operator control. The MVMCH3-104 provided with 020bug allows terminal and host access to the two serial ports on the VM04. 020bug may be utilized with a VM04 in a stand-alone environment with only a user-provided standard RS-232C asynchronous ASCII terminal. Alternately, 020bug may be used with the second serial I/O port direct-connected to a

host computer for up/downline loading of programs in Motorola "S" Record format. When directly connected to a host computer in this manner, the VM04/020bug/Operator Terminal combination appears as a normal asynchronous ASCII terminal (a virtual terminal) to the host operating system. Figures 1 and 2 illustrate two typical configurations using the 020bug EPROM set installed in a VM04.

In a typical debug session, the user will download a developed program to the VM04 from the host computer used for software development. This may be a Motorola EXORmacs or VME/10 Development System. Following download, 020bug debugger commands may be used to examine and modify memory, set breakpoints to run particular program segments, and trace program progress. The user may

FIGURE 1 — Benchmark 20 with 020bug Connected to a VME/10 Serving Both the Control and Downline Load Functions.

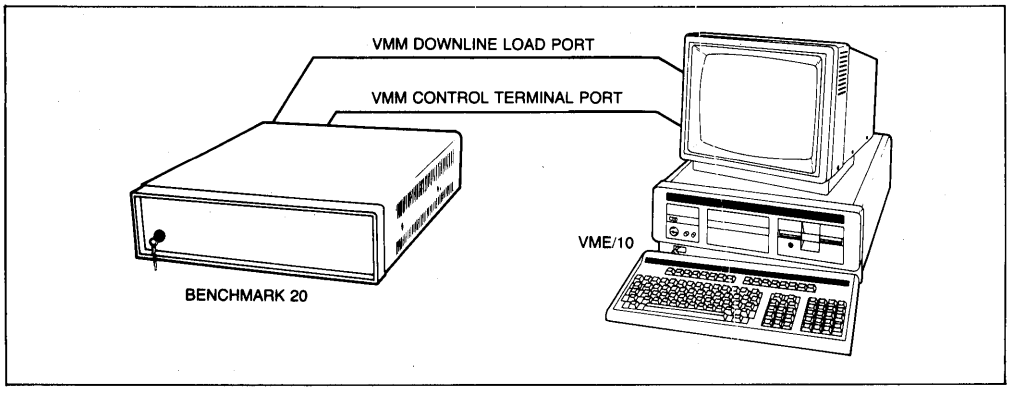
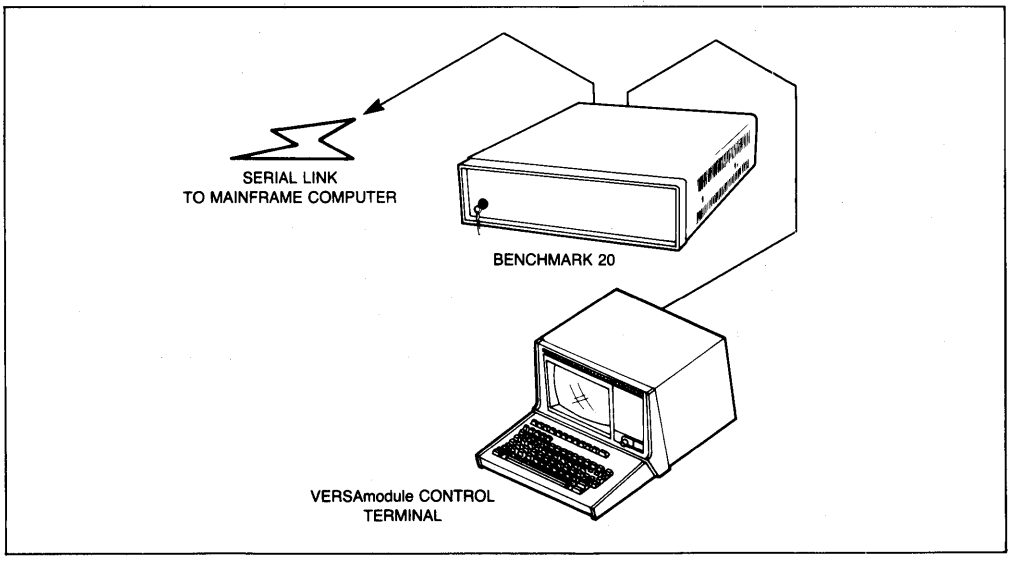


FIGURE 2 — Benchmark 20 with 020bug and Cross-Computer Serial Communications Link (EXORmacs, or Other Computer with "S" Record Files)



set up and examine a variety of conditions using any of the powerful commands listed in Table 1, such as the Register Display/Modify series and the Memory Block Manipulation commands. The Data Conversion command serves as an aid in examining and modifying data by providing a means of converting hexadecimal to decimal and decimal to hexadecimal. If corrections or program patches are required, these may be performed and checked in the down-loaded program. The user may then save a corrected copy to the host computer files utilizing the Memory Dump command for upline load. Debugging relocatable and position-independent code is easier with the built-in offset registers, which are set up using the Offset Registers Display/Modify command. The user may also copy all traffic on the system console terminal to a serial printer attached to the VM04's port 1 by using the Printer Attach command.

The user may communicate directly with the host computer as a terminal for purposes other than up/downline load by executing the Transparent Mode command. By using the Port Format command, the serial ports on the VM04 may be reconfigured for such attributes as baud rate, stop bits and number of data bits.

020bug may be used for debug in total systems environments including the VM04 together with other Motorola VERSAmodules (RAM, floppy and hard disk controllers, communications controllers, A/D controllers, etc.) as well as user-developed VERSAbus compatible modules.

Source modules permit the user to modify or customize any of the 020bug functions as desired.

TABLE 1 — 020bug Debugger Commands

MEMORY COMMANDS	SYNTAX
Memory Display	MD[S] <ADDR>[:<COUNT> | <ADDR>] [; [B | W | L | D]]
Memory Modify/Disassembly/Assembly	MM <ADDR> [;[[N] [B | W | L | A]|DI]
Memory Set	MS <ADDR> {Hexadecimal number}/{'string'}

REGISTER COMMANDS*	SYNTAX
Register Display	RD
Register Modify	RM <REG>

BLOCK OF MEMORY COMMANDS	SYNTAX
Block of Memory Fill	BF <RANGE><DEL><DATA>[;B | W | L]
Block of Memory Move	BM <RANGE><DEL><ADDR>[;B | W | L]
Block of Memory Search	BS <RANGE><DEL><TEXT> | <DATA>[<MASK<] [;B | W | L /N]

HOST COMMUNICATION COMMANDS	SYNTAX
Dump S-Records	DU[n] <RANGE><DEL>[<TEXT>] [<ADDR>] [;B | W | L]
Load S-Records	LO [<ADDR>] [;[-C/X]=<TEXT>]
Transparent Mode	TM [ESCAPE]
Verify S-Records	VE [<ADDR>] [;[-C/X]=<TEXT>]

GO COMMANDS	SYNTAX
Go Direct (Ignore Breakpoints)	GD [<ADDR>]
Go Execute User Program	GO [<ADDR>]
Go To Temporary Breakpoint	GT <ADDR>

TRACE COMMANDS	SYNTAX
Trace	T [<COUNT>]
Trace on Change of Control Flow	TC [<COUNT>]
Trace to Temporary Breakpoint	TT <ADDR>

BREAKPOINT COMMANDS		SYNTAX
Breakpoint Insert	BR	{<ADDR>[:<COUNT>]}
Breakpoint Delete	NOBR	{<ADDR>}

PRINTER COMMANDS	SYNTAX
Printer Attach	PA
Printer Detach	NOPA

MISCELLANEOUS UTILITY COMMANDS	SYNTAX
Data Conversion	DC <expression>
Help	HE [<COMMAND>]
Offset Registers Display/Modify	OF [Rn[;A]]
Port Format	PF[n]
Switch Directories	SD

*The register command options found in other Motorola Debug Monitors will be implemented on the 020bug during the first quarter of 1985.

HARDWARE DIAGNOSTICS

The 020bug packages include a separate directory that contains a complete set of hardware diagnostics for testing and troubleshooting the VM04 VMM. When this directory is accessed, a different prompt appears on the terminal screen to remind the user that the diagnostics are in use. The diagnostic command set is listed in Table 2.

Each of the hardware diagnostic commands actually refer to a family of subtests that may be invoked from the command line singly or in sequence. The subtest menu is displayed if a command is entered without specified subtests.

TABLE 2 — 020bug Diagnostic Commands

Type	Command Mnemonic	Description
DIAGNOSTIC SET-UP UTILITIES	LE	Loop-on-Error Mode
	SE	Stop-on-Error Mode
	LC	Loop Continuous Mode
	NV	Non-Verbose Mode
	DE	Display Errors
	ZE	Clear Error Counters
HARDWARE DEBUGGING TOOLS	ST	Run Self-Test Sequence
	WL	Continuous Write Loop
	RL	Continuous Read Loop
HARDWARE DIAGNOSTICS	FAT	Final Assembly Tests
	MT	Memory Tests
	CA	Cache Tests
	CIO	Counter/Timer Tests
	SIO	Serial I/O Tests
	BUS	VERSAbus and RAMbus Tests
	MMB	Memory Management Board Tests

USER CALLABLE ROUTINES

The 020bug TRAP #15 handler allows system calls from user programs. The system calls can be used to access functional routines contained within 020bug, including input and output routines.

To invoke a system call from a user program simply insert a TRAP #15 instruction into the source program. The code corresponding to the particular system routine is specified in the word following the TRAP opcode.

TABLE 3 — TRAP #15 Functions

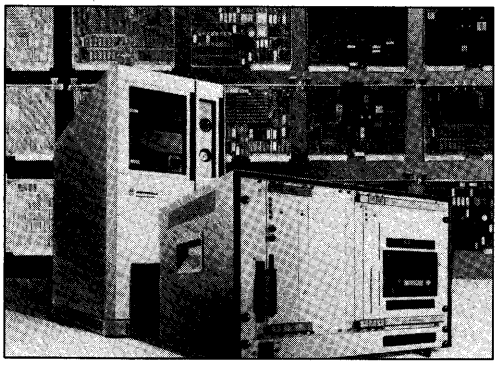
Code	Function	Description	
$0000	.INCHR	Input character from default input port	
$0001	.INSTAT	Input serial port status of default input port	
$0002	.INLN	Input line from default input port	(format 1)
$0003	.READSTR	Input string from default input port	(format 2)
$0004	.READLN	Input line from default input port	(format 2)
$0020	.OUTCHR	Output character to default output port	
$0021	.OUTSTR	Output string to default output port	(format 1)
$0022	.OUTLN	Output line to default output port	(format 1)
$0023	.WRITE	Output string to default output port	(format 2)
$0024	.WRITELN	Output line to default output port	(format 2)
$0025	.WRITDLN	Output line w/data to default output port	(format 2)
$0026	.PCRLF	Output carriage return and line feed to default port	
$0027	.ERASLN	Erase line being input from default input port	
$0040	.TM_INI	Initialize VM04's on-board timer	
$0041	.TM_STR0	Start timer at T = 0	
$0042	.TM_RD	Read timer	
$0060	.REDIR	Redirect I/O of a TRAP 15 function to use other than the default input or output port	
$0061	.REDIR_I	Change the default input port number	
$0062	.REDIR_O	Change the default output port number	
$0063	.RETURN	Return to 020bug	
$0064	.BINDEC	Convert binary to decimal	

Format 1 — pointer/pointer format Format 2 — pointer/count format

Ordering Information

Part Number	Description
M68K2RBBUG4	020bug, the VERSAmodule Benchmark 20 System Debug Package, includes EPROM set* and RS-232C Distribution Board (MVMCH3-104) allowing connection to user debug terminal and an up/downline load host. Includes User's Manual.
M68K2FSBUG4	Source and Relocatable Object Modules for the 020bug system on EXORmacs Diskette.* Includes User's Manual.
M68K2CSBUG4	Source and Relocatable Object Modules for the 020bug system on EXORmacs Cartridge Disk.* Includes User's Manual.
M68K2XSBUG4	Source and Relocatable Object Modules for the 020bug system on VME/10 Diskette.* Includes User's Manual.
MVMCH3-104	M68KVM04 RS-232C Distribution Board with Dual RS-232C Serial I/O Cable Assembly. Includes User's Manual.
M68K2RBBUG4/D1	020bug Debug Monitor User's Manual.

*The M68K2RBBUG4 EPROM set is copyrighted by Motorola and may be copied only under prior written agreement from Motorola. M68K2FSBUG4, M68K2CSBUG4 and M68K2XSBUG4 sources are copyrighted and licensed by Motorola. They may be obtained only under the required license agreement with Motorola.



The application dependent portion of many systems is comprised of special purpose I/O devices such as A/D converters, stepper motor controllers and printer and disk system interfaces. These devices transfer data at much slower rates than the core modules on the system bus and can reduce system throughput if allowed unrestricted access. Further, implementation of many I/O functions requires significantly less board area than processor or memory functions.

Systems based on either VMEbus or VERSAbus can reduce system bus traffic and obtain cost effective implementation of the specific I/O required for the application by using Motorola's line of I/O Channel-compatible modules.

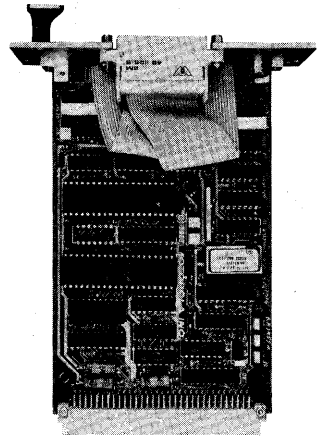
The I/O Channel is an advanced architectural feature of VMEmodule and VERSAmodule systems. It provides a 12-bit address, an 8-bit bidirectional data bus, 4K bytes of memory mapped I/O and a data transfer rate up to 2 megabytes per second. In addition to the bus signal protocol, the I/O Channel Specification defines connectors, pin assignments, ribbon cable characteristics, board level loading and driver/receiver parameters.

Most of Motorola's I/Omodules are implemented using the single-high Eurocard format. Serial and parallel interface, disk controller, magnetic tape interface, A/D and D/A converter and ac and dc input and output functions are offered in this format. Other functions compatible with the I/O Channel protocol are offered in special mechanical formats.

I/Omodule Data Sheets

MVME400

Dual Channel RS-232C Serial Port I/Omodule



- Single Eurocard Form Factor
- Two Independent, Full-Duplex Serial Input/Output Ports
- Motorola I/O Channel Interface Compatible
- Full RS-232C Interface Including Ring Indicator (RI) and Data Set Ready (DSR) Interrupt Capability. DSR Status may also be Monitored
- Jumper Configurable as Terminal or Modem Ports with Hardware Support for 103-J, 201 B&C, 202 R&S, 208, 209 and 212 Modems
- Multiprotocol Serial Controller (NEC7201) Provides Asynch, Bisynch, HDLC and SDLC Protocols
- Eight Jumper Selectable Baud Rates from 110 to 19.2K and 16 Software Programmable Baud Rates from 50 to 19.2K
- Interrupts are Jumperable to any of the Four Prioritized I/O Channel Interrupt Lines
- Jumper Selectable Base Address — any $10 Byte Block within the First $100 Bytes of I/O Channel Memory
- Self-test FAIL LED Controlled by System Self-Test Software
- 0° C-70° C Operating Temperature Range

The Dual Channel RS-232C Serial Port Module is used to expand the capabilities and number of serial communications channels available on the system I/O channel. It provides two independent full duplex serial input/output ports that interconnect to the microcomputer via the I/O channel interface. Through software and jumper selection, each port can be indepedently configured for:

1. Protocol: asynch, bisynch, HDLC or SDLC through the user's application software.
2. Baud Rates: a PIA-controlled baud rate generator provides 16 baud rates. Eight rates from 110 to 19.2K baud are jumper selectable. Eight other rates from 50 to 19.2K baud are user software selectable.
3. EIA interface: full RS-232C interface that can be configured through jumpers for either terminal or modem connection. The modem configuration is compatible with 103-J, 201 B&C, 202 R&S, 208, 209 and 212 modems. Both the Ring Indicator and the Data Set Ready signals, if used, can be jumpered to the PIA's controlled interrupt input lines. Data Set Ready is also connected to an input data bit on the PIA allowing its status to be monitored. These interrupts as well as the signal interrupt from the serial controller device can be jumpered to any of the four prioritized interrupt lines of the I/O Channel.

   Signals supported by these interfaces include Request To Send, Clear To Send, Data Carrier Detect, Data Terminal Ready, Data Set Ready, Ring Indicator, Transmitted Data, Received Data, Transmitter Clock and Receiver Clock.

SOFTWARE DRIVER

A driver for this Motorola I/O Channel interface module is incorporated in the VERSAdos M68000 Real-Time Operating System. It provides a host-resident application program with a ready means of communicating with the module and of obtaining the necessary and optional service functions required for controlling the peripheral resource to which the module is connected. Driver documentation is provided with the VERSAdos System. A manual, Guide to Writing A Device Driver, M68DRVGD/D1, detailing how to write a device driver that runs under the M68000 Real-Time Multitasking kernel (RMS68K) or under VERSAdos is also available.

Usage

Any I/Omodule in the MVME400 Series will operate with any of the following masters in control of the Motorola I/O Channel:

MVME110	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer

MVME400 Series

The 400 Series is a family of I/Omodule peripheral interface cards designed for modular, low-cost applications. These modules are mechanically compatible with a single Eurocard form factor, and will operate from the Motorola I/O Channel. Use of the I/O Channel allows the users to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

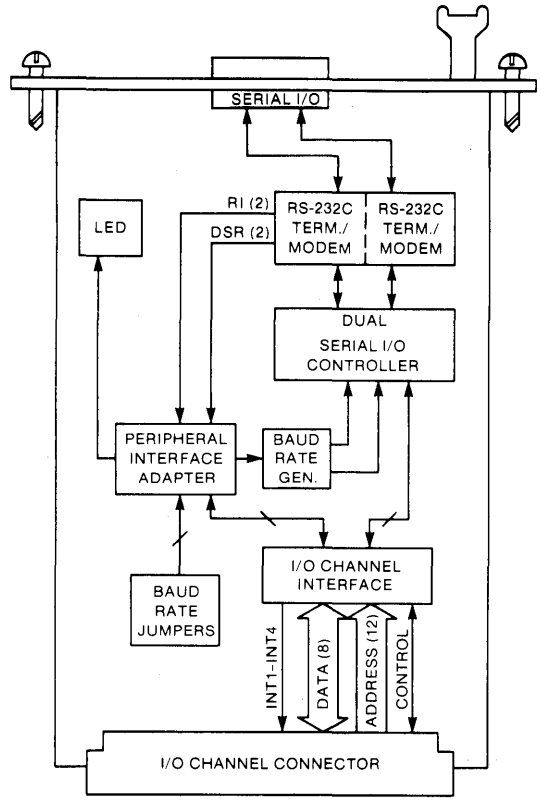
I/O Channel

The Motorola I/O Channel is specifically designed to provide efficient, low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 mbytes/sec. For those modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products, support Motorola's modular product families: VMEmodule and VERSAmodule.

TABLE 1 — Serial Port Connectors J1/J2 Pin Assignments and Signal Descriptions

Pin No.	Mnemonic	Description
2	TXD	Transmitted Data
3	RXD	Received Data
4	RTS	Request to Send
5	CTS	Clear to Send
6	DSR	Data Set Ready
7	SIG-GND	Signal Ground
8	DCD	Data Carrier Detect
15	TXC	Transmitter Clock
17	RXC	Receiver Clock
20	DTR	Data Terminal Ready
22	RI	Ring Indicator
24	TXC	Transmitter Clock

MVME400 — Dual Channel RS-232C Serial Port Block Diagram



Dual RS-232C Serial Port Specifications

Characteristic	Specification
Power Requirements	+5 Vdc @ 450 mA typical (991 mA maximum) +12 Vdc @ 50 mA typical -12 Vdc @ 40 mA typical
Temperature	
Operating	0° to 70° C
Storage	-40° to 85° C
Relative Humidity	0% to 90% (non-condensing)
Physical Characteristics	
PC Board only	
Height	6.30 in. (160 mm)
Depth	3.94 in. (100 mm)
Thickness	0.59 in. (15 mm)
PC Board with Connectors and Board Stiffener	
Height	7.40 in. (188 mm)
Depth	5.12 in. (130 mm)
Thickness	1.60 in. (41 mm)

Ordering Information

Part Number	Description
MVME400	Dual Channel RS-232C Serial Port I/Omodule. This module provides two independent, full-duplex serial input/output ports with RS-232C interfaces that connect to the microcomputer via the I/O Channel interface. Includes User's Manual.
MVME400/D1	MVME400 Dual RS-232C Serial Port Module User's Manual.

Related Documentation

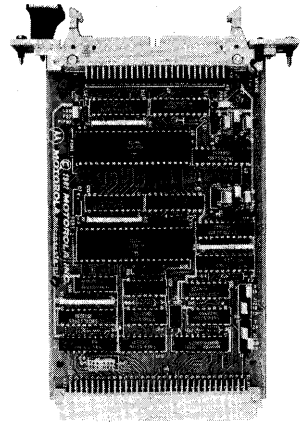
M68RIOCS/D2	Input/Output Channel Specification Manual
UPD7201	NEC Corp. data sheet for the UPD7201 Multiprotocol Serial Controller.

Other Modules in the MVME400 Series

MVME410	Dual Channel 16-Bit Parallel Port I/Omodule
MVME420	SASI' Peripheral Adapter I/Omodule
MVME435	Buffered 9-Track Magnetic Tape Adapter I/Omodule

MVME410

Dual Channel 16-Bit Parallel Port I/Omodule



- Single Eurocard Form Factor
- Four Independent 8-Bit TTL Level Parallel Input/Output Ports with Two Handshake Lines Per Port using Two Buffered MC6821 PIA's
- Motorola I/O Channel Interface Compatible
- Input/Output Provided on Two 50-Pin Shrouded Headers with Latching Ears. Pin-out is Compatible with Standard Centronics Type Printer Interface
- Each of the 8-Bit Port's Buffers are Jumper Configurable as Eight Inputs or as Eight Outputs or, using the Control Line CA2 or CB2 the Associated 8-Bit Port can be Software Configured for Input or Output
- For each Port, One Handshake Line, CA1 or CB1, is Buffered as an Input and One Handshake Line, CA2 or CB2, is Jumper Configurable as an Input, as an Output or as a Control Signal for the Data Direction Buffers
- Interrupts are Jumperable to any of the Four Prioritized I/O Channel Interrupt Priority Lines
- A FAIL LED is Jumperable to one of the PIA Data Bus Lines for Desired Software Control
- 0°C–70°C Operating Temperature Range

The Dual Channel 16-Bit Parallel Port (DPP) is a single Eurocard form factor module used to expand a microcomputer system. It provides connection to the processor via the Motorola I/O Channel for resources requiring a parallel interface. Two fully buffered MC6821 Peripheral Interface Adapters (PIAs) are employed in the DPP to provide ports offering a choice of two standard Centronics-type printer interfaces, two general purpose 16-bit parallel data I/O interfaces, or one of each.

Each of the four 8-bit data ports can be jumper configured as buffered inputs or outputs. In addition, the CA2 or CB2 control lines associated with each PIA can be jumpered for software control of the data direction buffers. If software control of the data direction is not required, CA2 and CB2 can be jumper configured as buffered inputs or outputs. The CA1 or CB1 control lines are buffered as inputs.

The 16 buffered data lines and four control lines from each PIA are connected to a 50-pin shrouded header with latching ears that is mounted on the module's front panel. The pin/signal assignments, see Table 1, are compatible with the standard Centronics type printer interface to provide interface between the microcomputer and two printers or to 32-bits of TTL level input or output data with two handshake lines for each 8-bit port.

If desired, bit PB7 of one PIA can be connected to a self-test FAIL LED for software controlled self-test status display. The four IRQ outputs of the PIA's can be jumpered to any of the four I/O Channel interrupt priority level lines.

The base address, modulo 16, of the two PIA's can be jumper selected to appear at any 16 byte boundary within the first 256 byte block of the I/O Channel memory space.

Software Driver

A driver for this Motorola I/O Channel interface module is incorporated in the VERSAdos M68000 Real-Time Operating System. It provides a host-resident application program with a ready means of communicating with the module and of obtaining the necessary and optional service functions required for controlling the peripheral resource to which the module is connected. Driver documentation is provided with the VERSAdos System. A manual, Guide to Writing A Device Driver, M68DRVGD/D1, detailing how to write a device driver that runs under the M68000 Real-Time Multitasking kernel (RMS68K) or under VERSAdos is also available.

Usage

MVME110	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer

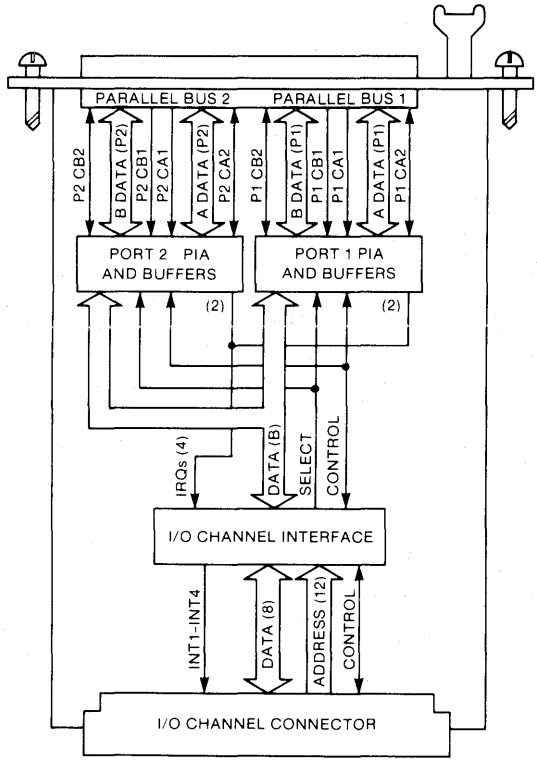
MVME400 Series

The 400 Series is a family of I/Omodule peripheral controller cards designed for modular, low-cost applications. These modules are mechanically compatible with a single Eurocard form factor, and operate from the Motorola I/O Channel. Use of the I/O Channel allows the user to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

I/O Channel

The I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For those modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support Motorola's modular product families: VMEmodule and VERSAmodule.

MVME410 Dual Channel 16-Bit Parallel Port Block Diagram



Typical Dual Parallel Port Interconnections

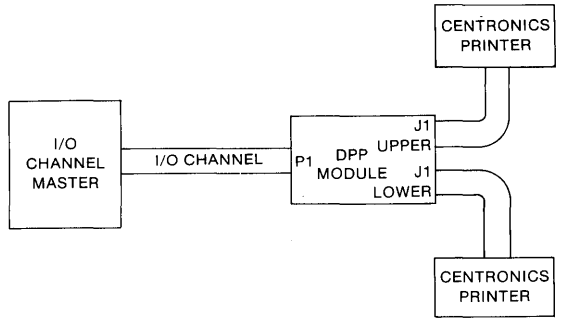


TABLE 1 — Connector J1 Pin Assignments and Signal Descriptions

J1 Pin Number	Signal Mnemonics	Signal Name And Description
1	PXCB2	INPUT PRIME — A low-level output signal which clears the printer buffer and initializes the logic. (Not used by all printers.)
3	GND	GROUND
5	PXCB1	FAULT — A low-level input signal that indicates a printer fault condition such as paper empty, light detect, or a deselect condition. (Not used by all printers.)
7,41,45,49	RESERVED	N/A
9	PXPB7	N/A
11	PXPB6	N/A
13	PXPB5	N/A
15	PXPB4	N/A
17	PXPB3	N/A
19	PXPB2	BUSY — An input signal indicating that the printer cannot receive data.
21	PXPB1	OUT OF PAPER — A high-level input indicating the printer is out of paper.
23	PXPB0	SELECT — A high-level signal indicating that the printer is selected.
25	PXPA7	PERIPHERAL DATA LINE (PD8) — Output data to printer from PA7 of PIA.
27	PXPA6	PERIPHERAL DATA LINE (PD7) — Same as pin 25 except from pin A6.
29	PXPA5	PERIPHERAL DATA LINE (PD6) — Same as pin 25 except from pin A5.
31	PXPA4	PERIPHERAL DATA LINE (PD5) — Same as pin 25 except from pin A4.
33	PXPA3	PERIPHERAL DATA LINE (PD4) — Same as pin 25 except from pin A3.
35	PXPA2	PERIPHERAL DATA LINE (PD3) — Same as pin 25 except from pin A2.
37	PXPA1	PERIPHERAL DATA LINE (PD2) — Same as pin 25 except from pin A1.
39	PXPA0	PERIPHERAL DATA LINE (PD1) — Same as pin 25 except from pin A0.
41	RESERVED	N/A
43	PXCA2	DATA STROBE — An output pulse used to clock data from the MPU to the printer logic. The pulse is active low and at least 1 μs wide.

TABLE 1 — Connector J1 Pin Assignments and Signal Descriptions (continued)

J1 Pin Number	Signal Mnemonics	Signal Name And Description
47	PXCA1	ACKNOWLEDGE — A low-level input pulse indicating the input of a character into memory or the end of a functional operation.
EVEN NUMBERS 2-50 (Except 8)	—	GROUND
8	—	No connection.

Dual 16-Bit Parallel Port Specifications

Characteristic	Specification
Power Requirements	+5 Vdc @ 762 mA typical (991 mA maximum)
Temperature	
Operating	0° to 70°C
Storage	-40° to 85°C
Relative Humidity	0% to 90% (non-condensing)
Physical Characteristics	
PC Board only	
Height	6.30 in. (160 mm)
Depth	3.94 in. (100 mm)
Thickness	0.59 in. (15 mm)
PC Board with Connectors and Board Stiffener	
Height	7.40 in. (188 mm)
Depth	5.12 in. (130 mm)
Thickness	0.83 in. (21 mm)

Ordering Information

Part Number	Description
MVME410	Dual Channel 16-Bit Parallel Port I/Omodule. This module provides four 8-bit data ports with two handshake lines per ports that are controlled by a microcomputer I/O Channel interface. The ports may be directly connected to a Centronics printer type interface. Includes User's Manual.
MVME410/D1	MVME410 VMEmodule Dual Parallel Port Module User's Manual.

Related Documentation

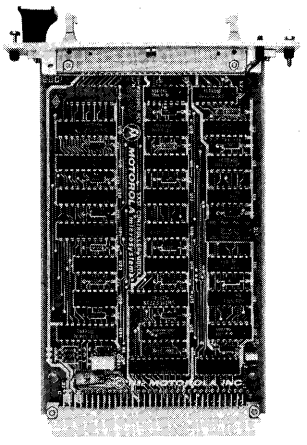
M68RIOCS/D2	Input/Output Channel Specification Manual
MC6821	Peripheral Interface Adapter Data Sheet

Other Modules in the MVME400 Series

MVME400	Dual Channel RS-232C Serial Port I/Omodule
MVME420	SASI' Peripheral Adapter I/Omodule
MVME435	Buffered 9-Track Magnetic Tape Adapter I/Omodule

MVME420

SASI™ Peripheral Adapter I/Omodule



- Single Eurocard Form Factor
- Provides an Interface to the Shugart Associates SASI Bus
- Supports a Single Host, Non-Arbitrating, No Parity Implementation of the SASI Bus
- Supports the Shugart Associates SA1403D Controller
- Motorola I/O Channel Interface Compatible
- Appears as Eight Adjacent 1-Byte Read/Write Registers in the I/O Channel Address Map
- Base Address Jumper Selectable — any of Sixteen 8-Byte Blocks in $00 to $80 I/O Channel Range
- Maskable Interrupt is Jumperable to any of Three Prioritized I/O Channel Interrupt Priority Lines
- Status Flag Register Included for Polling Mode of Operation
- Test LED Controlled by System Diagnostic Software
- 0°C-70°C Operating Temperature Range

The SASI Peripheral Adapter Module (SAM) is used to add a Shugart Associates disk drive or other SASI bus compatible peripheral to microcomputer system resources on the Motorola I/O Channel. It provides interface between the I/O Channel and the Shugart Associates SASI bus, which can then be connected to the single host, non-arbitrating SA1403D controller. Implemented using the single Eurocard form factor, the SAM connects to the I/O Channel by means of a standard triple row, DIN4162 64-pin connector and to the SASI bus using a standard double row, 50-pin connector, 3M 3425-5000 or equivalent. Complete buffering of all I/O Channel and SASI bus address data and control lines is provided. A SAM front panel red FAIL LED is provided for reset indication and, under software control, indication of exception or other status.

The SASI Adapter Module contains the following registers:

1. Select Register — a write only register used to select the controller.
2. Flag Register — contains information about the module status and the SASI bus state.
3. Command Register — a write only register used to transfer the command sequence to the SA1403D controller.
4. Status Register — contains the SA1403D status, used at the completion of the message transfer sequence.
5. Test Register — write/read register used to check communication path between host and SAM.
6. Control Register — a write only register used to control the interrupt enable, FAIL LED and reset operations.
7. Data Register — used to read data and sense from the controller and write data in the controller. This register resides at four redundant locations in the I/O Channel address map.

Software Driver

A driver for this Motorola I/O Channel interface module is incorporated in the VERSAdos M68000 Real-Time Operating System. It provides a host-resident application program with a ready means of communicating with the module and of obtaining the necessary and optional service functions required for controlling the peripheral resource to which the module is connected. Driver documentation is provided with the VERSAdos System. A manual, Guide to Writing A Device Driver, M68DRVGD/D1, detailing how to write a device driver that runs under the M68000 Real-Time Multitasking kernel (RMS68K) or under Vdos is also available.

Usage

Any I/Omodule in the MVME400 series will operate with any of the following masters in control of the Motorola I/O Channel.

MVME110	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer

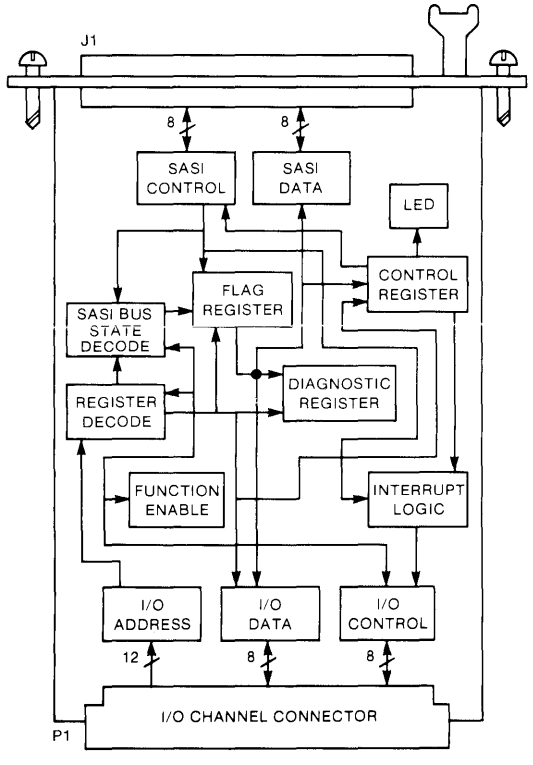
MVME400 Series

The 400 Series is a family of I/Omodule peripheral controller cards designed for modular, low-cost applications. These modules are mechanically compatible with a single Eurocard form factor, and will operate from the Motorola I/O Channel. Use of the I/O Channel allows the user to divert heavy peripheral data flow from the System bus onto a local controller to improve system throughput. Operation from the I/O Channel also allowed these controllers to be standardized across Motorola's product families: VERSAmodules and VMEmodules.

I/O Channel

The Motorola I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For those modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products, support Motorola's 16-bit modular product families: VMEmodule and VERSAmodule.

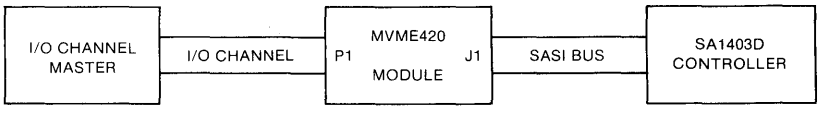
MVME420 — SASI Peripheral Adapter Block Diagram



SASI Peripheral Adapter Specifications

Characteristic	Specification
Power Requirements	+5 Vdc @ 1.0 maximum
Temperature	
Operating	0° to 70° C
Storage	-40° to 85° C
Relative Humidity	0% to 90% (non-condensing)
Physical Characteristics	
Single Eurocard board	
PC Board only	
Height	6.30 in. (160 mm)
Depth	3.94 in. (100 mm)
Thickness	0.59 in. (15 mm)
PC Board with Connectors and Board Stiffener	
Height	7.40 in. (188 mm)
Depth	5.12 in. (130 mm)
Thickness	0.83 in. (21 mm)

Typical System Interconnections



Connector J1 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonics	Signal Name And Description
1-49 (odd numbers)	GND	GROUND
2	-DB0	DATA BIT 0 — Bidirectional data lines used to transfer 8-bit parallel data to/from the host adapter. Negative logic is used and bit 7 is the MSB.
4	-DB1	DATA BIT 1 — Same as DB0 on pin 2.
6	-DB2	DATA BIT 2 — Same as DB0 on pin 2.
8	-DB3	DATA BIT 3 — Same as DB0 on pin 2.
10	-DB4	DATA BIT 4 — Same as DB0 on pin 2.
12	-DB5	DATA BIT 5 — Same as DB0 on pin 2.
14	-DB6	DATA BIT 6 — Same as DB0 on pin 2.
16	-DB7	DATA BIT 7 — Same as DB0 on pin 2.
18	(Reserved)	Not used.
20-34 (even numbers)	(Reserved)	Not used.
36	-BSY	BUSY — Made true in response to the SEL line from the host adapter to indicate that the host bus is currently in use.
38	-ACK	ACKNOWLEDGE — Made true in response to each true REQ signal from the controller.
40	-RST	RESET — When made true by the host, the controller goes to an idle condition.
42	-MSG	MESSAGE — When true, indicates that the command is completed and status has been transferred.
44	-SEL	SELECT — When made true, indicates the beginning of the command transaction.
46	-C/D	CONTROL/DATA — When true, the data transmitted will be command or status bytes; when false, the data will be disk data bytes.
48	-REQ	REQUEST — When true and I/O is true, data on the host bus is driven by the controller. When true and I/O is false, data is driven by the host adapter.
50	-I/O	INPUT/OUTPUT — When true, data on the bus is driven by the controller; when false, data is driven by the host adapter.

NOTE: A minus sign preceding a signal mnemonic denotes an active low signal.

Ordering Information

Part Number	Description
MVME420	SASI Peripheral Adapter I/Omodule. This module provides an interface between a microcomputer I/O Channel interface and a Shugart Associates SASI bus for a SA1403D hard disk controller. Includes User's Manual.
MVME420/D1	MVME420 SASI Adapter Module User's Manual.

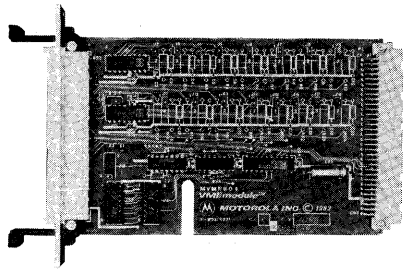
Related Documentation

M68RIOCS/D2	Input/Output Channel Specification Manual
SA1403D	Shugart Associates Technical Manual for the SA1403D Controller.

Other Modules in the MVME400 Series

MVME400	Dual Channel RS-232C Serial Port I/Omodule
MVME410	Dual Channel 16-Bit Parallel Port I/Omodule
MVME435	Buffered 9-Track Magnetic Tape Adapter I/Omodule

Analog Input And Expander I/Omodules



- 16 Single Ended/8 Differential A/D Channels With:
  - Two jumper selectable conversion ranges together with programable X1 or X10 amplifier gain provide four full scale input signal ranges:
    Single Ended 0-0.5 V, 0-1.0 V, 0-5 V, 0-10 V
    Differential ±0.5 V, ±1.0 V, ±5 V, ±10 V
  - 12 bit conversion, offset binary
  - 38 μs conversion time to 0.1% ±1 LSB (typical at 25°C)
  - Programmable conversion start: external trigger or program initiation
  - Programmable channel selection
- Expandable to 96 Channels (Using Five MVME601 Modules)
- Voltage or Current Inputs
- Handles 4-20 mA/10-50 mA Instruments (with Customer Provided Resistors)
- Bipolar Inputs
- input Protection to 100 V for 1 ms, Input to Input, Input to Ground
- Input Impedance Greater than 10 MΩ, Input to Ground
- Jumper Selectable I/O Channel Base Address
- Motorola I/O Channel Compatible
- 0°C-70°C Operating Temperature Range
- Single +5 V Operation

A microcomputer system for the process control, automatic test and other industrial and laboratory automation environments often utilizes both digital and analog equipment, imposing the need for suitable interfaces. Such interfaces typically provide conversion into digital form of sampled voltages and currents and conversion of digital information to voltage and current.

The MVME600 Analog Input I/Omodule is an analog data acquisition device capable of sampling a voltage or current, converting the sample into a 12-bit offset binary value and making the value available to a host on the Motorola I/O Channel. Inputs are provided for 16 single ended or eight differential channels. The number of channels can be increased using MVME601 Expander I/Omodules, each of which offers 16 additional inputs. A single MVME600 can accommodate five expander modules for a total of 96 channels per A/D subsystem, as shown in Figure 1.

GENERAL DESCRIPTION — MVME600/MVME601

The MVME600 Analog Input I/Omodule is a single high Eurocard form factor module containing two 8-input CMOS multiplexers, a dual input instrumentation amplifier and a high speed analog to digital converter of 12 bit resolution. The module has I/O Channel interface circuitry which includes data bus transceivers and address decoding logic. A control register and two data registers facilitate intercommunication with a host on the I/O Channel. A functional block diagram of MVME600 is shown in Figure 2.

The MVME601 Expander I/Omodule is a single high Eurocard form factor module containing two 8-input CMOS multiplexers, I/O Channel data bus transceivers, an 8-bit write only control register, and two connectors for daisy chain cable interconnection to a single MVME600 and/or other MVME601 modules. Figure 3 is a functional block diagram of an MVME601 module connected to an MVME600 module in an A/D subsystem. The diagram illustrates how additional expander modules are connected.

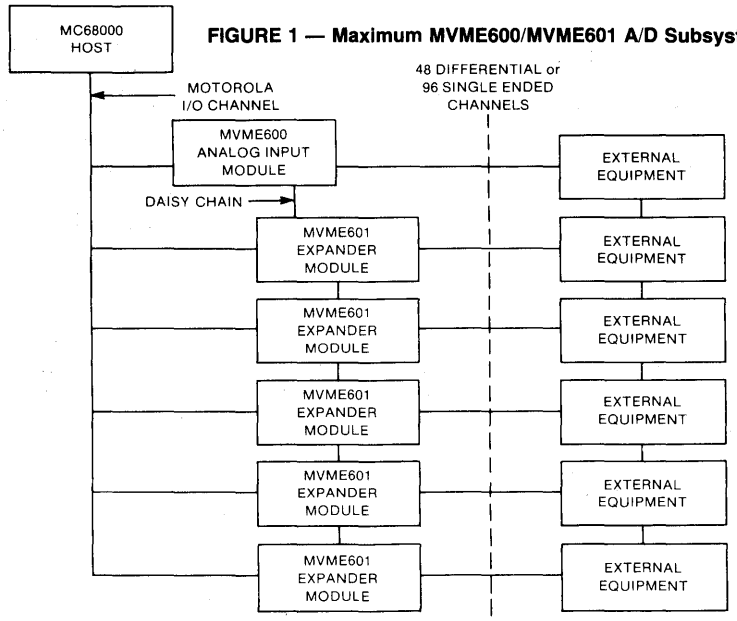


FIGURE 1 — Maximum MVME600/MVME601 A/D Subsystem

FIGURE 2 — MVME600 Functional Block Diagram

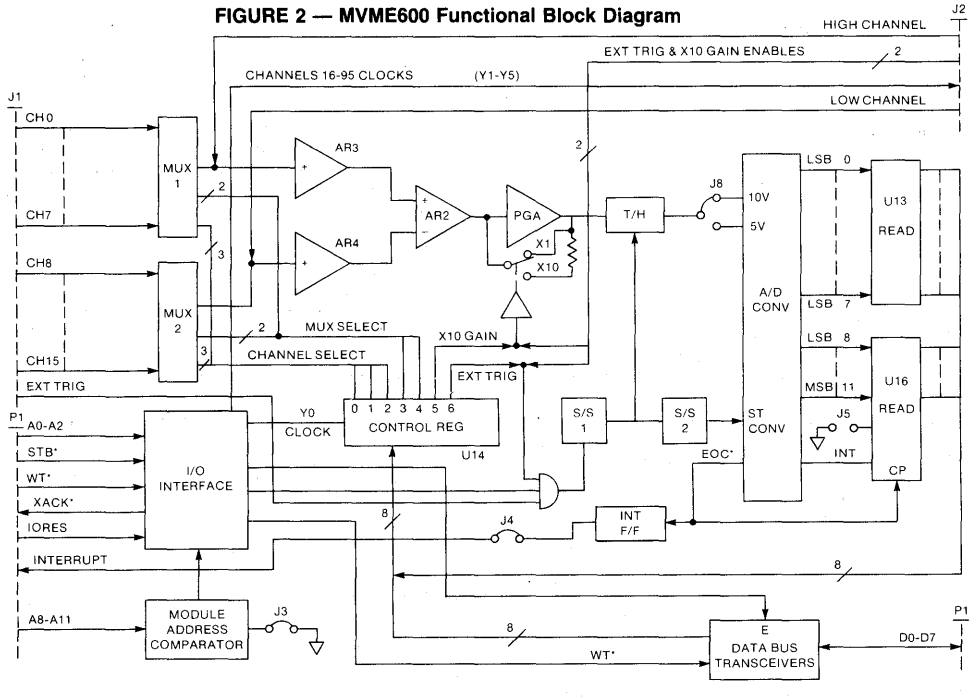
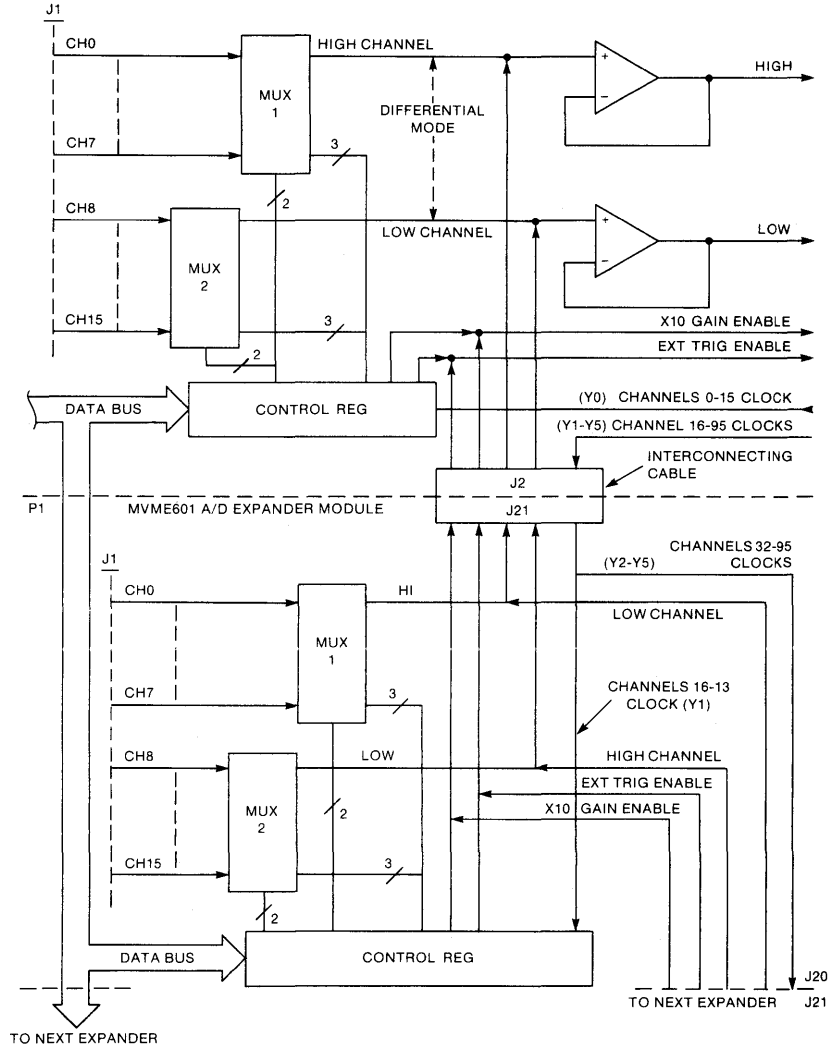


FIGURE 3 — MVME600/MVME601 Interconnection Diagram



MVME600/MVME601 Measurement Capabilities

Both the MVME600 and MVME601 can be connected for operation in the single ended or differential measurement mode. Single ended operation provides 16 input channels per module; differential provides eight. In one A/D subsystem, i.e., an MVME600 and the expander modules connected to it, all inputs must be configured for single ended or configured for differential measurements.

In the differential mode, measurements of unipolar and bipolar voltages can be made as well as measurements of unipolar and bipolar currents, (using customer supplied current sensing resistors). Measurements of unipolar and bipolar voltages can be made in the single ended mode.

Jumper selection on MVME600 of a basic voltage range of 5 V or 10 V in conjunction with a program selected programmable amplifier gain of X1 or X10 provides a choice under program control of one of two pairs of full scale voltages: ±5 V or ±0.5 V and ±10 V or ±1.0 V.

Both module types provide input-to-input and input-to-ground protection against 100 V for up to one millisecond. Input-to-input and input-to-ground impedance of >10 megohms and >80 dB of common mode rejection ratio are provided by the input instrumentation amplifier on MVME600.

A/D Subsystem Operation

To a user program accessing an MVME600, the module appears as three registers in the I/O Channel memory space: an 8-bit write only control register and two 8-bit read only data registers. An expander module, on the other hand, appears as a single control register at a unique I/O Channel location since a voltage or current input to an expander channel is sent to and converted on the MVME600 module of the A/D subsystem and the resulting offset binary value read from the MVME600 data registers. Thus an A/D subsystem fully complemented with one MVME600 and five MVME601 modules appears to the user program as six control registers and two data registers. Fiqure 4 shows the address of each register in an A/D subsystem relative to the MVME600 I/O Channel base address.

For each conversion, a binary value is written in the control register of the appropriate module to select the input channel and measurement scale and to obtain external triggering or programmed start of conversion.

The most significant byte and least significant byte are obtained by consecutive reads of the MVME600 data registers before a value is written into a control register to set up the next measurement. Figure 5 and 6 show the use of each bit in the control and data registers. Channel selection details for single ended and differential measurements are shown in Figures 7 and 8.

FIGURE 4 — MVME600/MVME601 I/O Channel Map

BASE ADDRESS		OFFSET
$XXXXXX1	MVME600 CONTROL	0
$XXXXXX2		1
$XXXXXX3	MVME600 DATA (MSB)	2
$XXXXXX4		3
$XXXXXX5	MVME600 DATA (LSB)	4
$XXXXXX6		5
$XXXXXX7	MVME601 # 1 CONTROL	6
$XXXXXX8		7
$XXXXXX9	MVME601 # 2 CONTROL	8
$XXXXXXA		9
$XXXXXXB	MVME601 # 3 CONTROL	10
$XXXXXXC		11
$XXXXXXD	MVME601 # 4 CONTROL	12
$XXXXXXE		13
$XXXXXXF	MVME601 # 5 CONTROL	14

As directed by the control register, an input signal on the selected channels is conditioned by the instrumentation amplifier and temporarily stored in the track and hold circuitry. On start of conversion, the offset binary value from the previous measurment is latched into the data registers. When conversion of the current sample is completed, an end-of-conversion (EOC) signal from the A/D converter sets the interrupt flip flop. If an interrupt

priority level is selected (by jumper), the signal is routed to the I/O Channel Interrupt line of that priority. At the same time, bit 15 of the MVME600 data register is set informing the user program that conversion is complete and that an interrupt is pending.

FIGURE 5 — MVME600/MVME601 Control Register (Write Only)

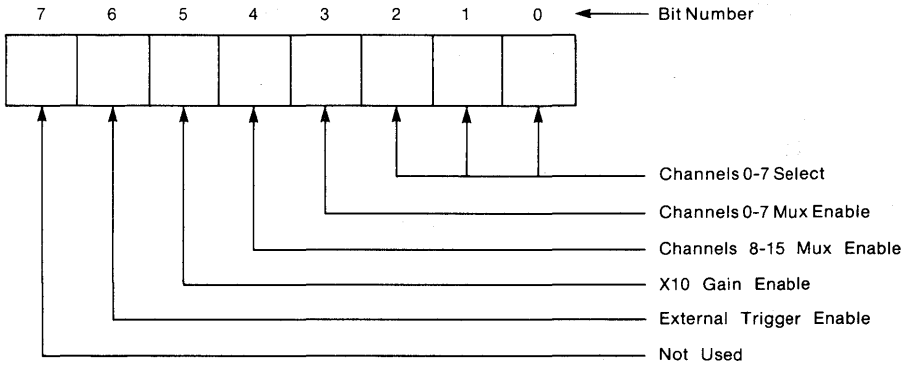


FIGURE 6 — MVME600 Data Registers (Read Only)

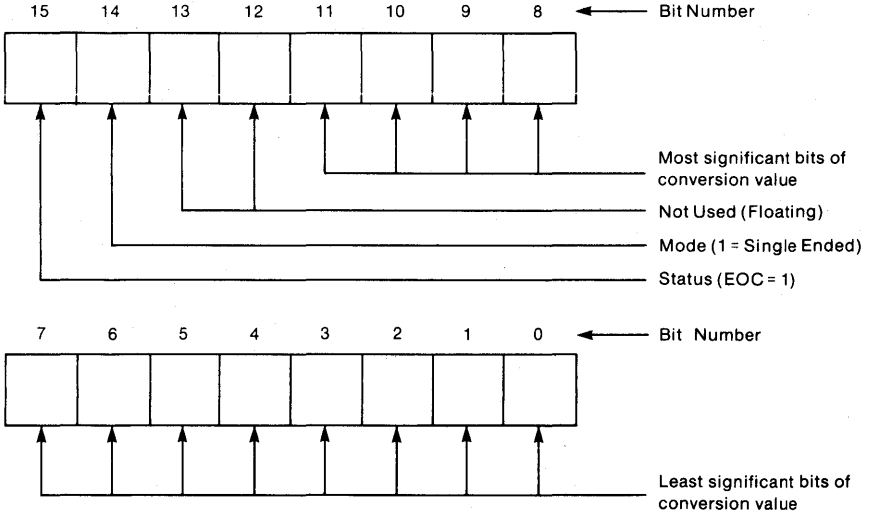


FIGURE 7 — Channel Selections for Single-Ended Measurements

Channel-Selection Bits / Multiplexer-Selection Bits					Channel Selected
4	3	2	1	0	
0	1	0	0	0	0
0	1	0	0	1	1
0	1	0	1	0	2
0	1	0	1	1	3
0	1	1	0	0	4
0	1	1	0	1	5
0	1	1	1	0	6
0	1	1	1	1	7
1	0	0	0	0	8
1	0	0	0	1	9
1	0	0	1	0	10
1	0	0	1	1	11
1	0	1	0	0	12
1	0	1	0	1	13
1	0	1	1	0	14
1	0	1	1	1	15

FIGURE 8 — Channel Selections for Differential Measurements

Channel-Selection Bits / Multiplexer-Selection Bits					Channel Selected
4	3	2	1	0	
1	1	0	0	0	0 and 8
1	1	0	0	1	1 and 9
1	1	0	1	0	2 and 10
1	1	0	1	1	3 and 11
1	1	1	0	0	4 and 12
1	1	1	0	1	5 and 13
1	1	1	1	0	6 and 14
1	1	1	1	1	7 and 15

Base Address Selection

A header is provided on MVME600 that allows jumper selection of a base address in the I/O Channel memory space. Selection is implemented using the four highest order of the 12 I/O Channel address lines allowing the base address of the module to be located on any 256 byte boundary throughout the space.

Logic on the MVME600 module compares the levels on the I/O Channel address lines with the jumper selected address. On recognizing the address of a control register on an expander module or the MVME600, the circuitry generates and sends an enabling clock signal to the corresponding register allowing a control value to be written by the user program.

Seen from a host, I/O Channel memory space (4K bytes, 12 address lines) begins at an address determined by the particular host. The current addresses are hexadecimal F80000 (M68KVM02-3), FA0000 (M68KVM03) FE6000 (MVME110-1) and F1C000 (VME/10). Because of the manner in which the VMEbus is interfaced to the I/O Channel address bus, only the odd MC68000 addresses are used.

Interrupt Priority Selection

To facilitate use in multiple interrupt priority level systems, the MVME600 has a header that permits connection, by jumper, to one of the four prioritized I/O Channel interrupt lines. This sets the interrupt level for all channels in the A/D subsystem based on that MVME600.

Software Driver

A driver is available for use in A/D systems utilizing the RMS68K Executive Kernel or the VERSAdos Operating System. For users desiring to write their own driver, a manual, "Guide to Writing A Device Driver", part number M68DRVGD, is available.

The A/D driver requires that a level be jumper selected for the EOC interrupt and that the level be identified at system generation time. However, sense loop operation is not precluded since the driver can mask the interrupt level and continuously test the EOC/interrupt bit (#15) in the MVME600 data register for end of conversion indication.

Each of the current Motorola Monoboard Microcomputers (MVME110-1, M68KVM03, M68KVM02-3) provides a way for a device on the I/O Channel to interrupt the MC68000 and for the interrupt to be captured by the Channel Management Routine (CMR) of the RMS68K executive kernel. When CMR polls the various I/O drivers having the priority level of the interrupt, the A/D driver will claim and process the interrupt, if MVME600 generated.

The MVME600/MVME601 driver is an I/O handler that runs under the RMS68K executive kernel. It provides the device dependent portion of the software interface required for an A/D system.

The A/D driver is re-entrant so that only a single copy is needed by a user task to control the maximum number of MVME600/MVME601 A/D subsystems that can be supported by the Motorola I/O Channel. This is 16 subsystems of 96 measurement channels each or a total of 1536.

The basic function of the driver is to manage the logical connections between tasks and the A/D subsystem measurement channels. Beyond this, the driver provides tasks with four capabilities to facilitate control of the measurements. These are the OPEN, CLOSE, MEASURE, and STOP requests.

In response to an OPEN request, the driver opens the specified measurement channel for the exclusive use of the requesting task until the same task issues a CLOSE request. A MEASURE request allows a task to set up on its own open channel any number of measurements of any nature consistent with the physical set up of the hardware. The measurements can be sequenced a specified number of times at specified intervals, if desired. If requested, notification of individual measurement completion or full sequence completion will be sent to the task. Additional flexibility for control of measurements is provided by the STOP command which allows a task to stop a measurement sequence at any time.

Although more than one task may not simultaneously perform measurements on the identical channel on one MVME600 or MVME601, multiple tasks can independently utilize channels on the same module. At one time on a particular channel, a single task is allowed only one outstanding MEASURE request.

A task may cause the driver to queue an event to or wake up the requesting task on completion of any measurement or sequence of measurements requested by the task on any channel or channels opened by the task. A task may also cause the driver to call a user-specified subroutine on completion of an individual measurement or sequence of measurements. The latter capability can be used to minimize measurement data buffer size since a task is required to specify a storage area in which the driver must cache the result of each measurement.

MVME600/MVME601 Usage

MVME110-1 VMEmodule Monoboard Microcomputer

M68KVM02-3 VERSAmodule Monoboard Microcomputer

M68KVM03 VERSAmodule Monoboard Microcomputer

VME/10 Microcomputer System

MVME600 Series

The MVME600 Series is a family of I/Omodule peripheral interface cards designed for modular low cost applications. These modules are mechanically compatible with a single Eurocard form factor and will operate from the Motorola I/O Channel. Use of the I/O Channel allows the users to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

I/O Channel

The I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For modules performing time critical operations four prioritized interrupt lines are also provided.The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support Motorola's modular product families: VMEmodule and VERSAmodule.

Specifications

The general, analog section and digital section specifications of the MVME600/MVME601 modules are provided in Tables 1, 2 and 3 respectively.

TABLE 1 — MVME600/MVME601 Specifications – General

Characteristic	Specification
Power	
Requirement	±5 Vdc at 0.75 A max. with dc/dc converter
Supply Tolerance	Single High Eurocard, 4" (102 mm) x 6-1/4" (159 mm)
Environmental Requirements	
Operating Temperature	0 to 70° C
Storage Temperature	-20 to +85° C
Humidity	0% to 95%, non-condensing
Mechanical Specifications	
Board Size	Single High Eurocard, 4" (102 mm) x 6-1/4" (159 mm)
Connectors	J1, 32 pin analog input, AMP 2 - 164306 or equiv. P1, 64-pin PCB

TABLE 2 — MVME600/MVME601 Specifications – Analog Section

Characteristic	Specification
Inputs	
Voltage Channels	16 single-ended or 8 differential
Ranges	±0.5 Vdc, ±1.0 Vdc, ±5.0 Vdc, ±10 Vdc
Current Channels	Eight
Ranges	4-20 mA, 10-50 mA (using customer supplied resistors)
Impedance	>10 Megohms
Maximum Input Voltage	±10 Vdc continuous ±100 Vdc (less than 1 ms)
Gain (software programmable)	X1, X10
Accuracy	
System	±0.1% (±10 Vdc range)
Drift (Gain = 1)	<100 PPM/°C
Linearity	±1/2 LSB
Gain Error	Adjustable to zero
Offset Error	Adjustable to zero
Transfer Characteristics	
Resolution	12 bits
Conversion Time (Gain = 1)	38 μs to 0.1% ±1 LSB
Instrumentation Amplifier CMRR	>80 dB (dc to 1 kHz, Gain = 1)
Output Coding	
Bipolar	Offset binary

TABLE 3 — MVME600/MVME601 Specifications – Digital Section

Characteristic	Specification
Addressing	Fixed on board decoding
Interrupts	Any one of four I/O Channel levels, jumper selected, EOC enabled.
I/O Channel Address	12 bits - 4 MSB (jumper selectable) used to define MVME600 address
Data	8 bits - bidirectional
Control	4 lines: Module Reset (IORES*), Transmit Acknowledge (XACK*), Strobe (STR*), and Write (WT*)

Analog Input and Expander Connectors

Table 4 provides the pin assignments and channel identifications for analog input connector J1. Table 5 provides the pin assignments and signal descriptions for expander connector J2 (J20 and J21, MVME601). The I/O Channel connector P1 mechanical and electrical characteristics are fully described in Motorola I/O Channel Specification M68RIOCS.

TABLE 4 — Analog Input Connector J1

Differential Channels	Pin Number	Single Ended Channels
+CH0	z14	CH0
-CH0	d20	CH8
+CH1	d14	CH1
-CH1	z20	CH9
+CH2	z12	CH2
-CH2	d22	CH10
+CH3	d12	CH3
-CH3	z22	CH11
+CH4	d18	CH4
-CH4	d26	CH12
+CH5	z18	CH5
-CH5	z26	CH13
+CH6	z16	CH6
-CH6	z24	CH14
+CH7	d16	CH7
-CH7	d24	CH15

Analog common d2
External trigger d30

NOTE: All unused inputs should be grounded

TABLE 5 — MVME600 Module Connector J2 (MVME601 J20/J21)

Pin Number	Signal Mnemonic	Description
1	HI	HIGH - output signal from number one multiplexer
2	LO	LO - output signal from number two multiplexer
3	X10 GAIN	X10 Gain - output signal from control register
4	16-31	CHANNEL 16-31 SELECT - output signal from the address decoder used to enable the control register on the expander module strapped for that channel group
5	32-47	CHANNEL 32-47 SELECT - same as signal on pin 4
6	48-63	CHANNEL 48-63 SELECT - same as signal on pin 4
7	64-79	CHANNEL 64-79 SELECT - same as signal on pin 4
8	80-95	CHANNEL 80-95 SELECT - same as signal on pin 4
9, 10		Not Used
11	INT RESET*	INTERNAL RESET - output signal used to reset the expander modules
12	-15 V	15 Vdc Power - used by the expander module multiplexers
13	EXT TRIG	EXTERNAL TRIGGER BIT - high-level output signal used to control the track and hold circuit
14	GND	ANALOG GROUND
15	GND	GROUND
16	+15V	15 Vdc Power - used by the expander module multiplexers

Ordering Information

Part Number	Description
MVME600	Analog Input I/Omodule providing 16 single-ended or eight differential channels, with 12 bit resolution, ±0.1% accuracy and ± 1/2 LSB linearity. Includes User's Manual.
MVME601	Analog Input Expander I/Omodule providing 16 single-ended or eight differential channels for use with and offering performance of MVME600. Includes interconnecting cable.
MVME600/D1	Anolog Input I/Omodule User's Manual

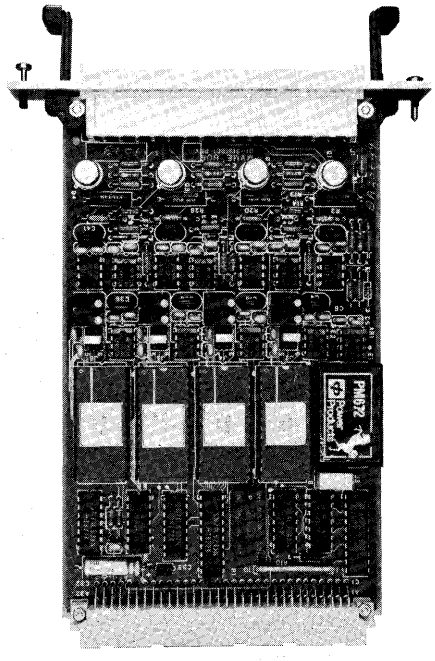
Related Documentation

M68RIOCS/D2	Input/Output Channel Specification Manual
M68DRVGD/D1	Guide to Writing a Device Driver
M68KRMS68K	M68000 Realtime Multitasking Software User's Manual
M68KSYSGEN	System Generation Facility User's Manual

Modules in the MVME600 Series

MVME605	Analog Output I/Omodule
MVME610	Opto Isolated 120/240 Vac Input I/Omodule
MVME615	Opto Isolated 120/240 Vac Output I/Omodule with Zero Crossing Switching
MVME616	Opto Isolated 120/240 Vac Output I/Omodule with Non-Zero crossing switching
MVME620	Opto Isolated 60 Vdc Input I/Omodule
MVME625	Opto Isolated 60 Vdc Output I/Omodule

Four Channel Analog Output I/Omodule



- Analog Output Module
- Four Independent Programmable Channels
- 12-bit Resolution
- 5.0 μs Conversion Time
- ±0.1% FS, ±1.0 LSB Accuracy
- Bipolar Voltage Outputs
- Three Full Scale Ranges ±10 V; ±5.0 V, ±2.5 V
- 4–20 mA Current Loop Output on each Channel Simultaneous with Voltage
- Power Requirements ±5.0 V, ±12 V
- On Board DC/DC Converter Supplies ±15 V for D/A Amplifiers
- Output Current Protection
- 0°C–70°C Operating Temperature Range
- Single Eurocard Form Factor
- Motorola I/O Channel Compatible

A microcomputer system for the process control, automatic test and other industrial and laboratory automation environments often utilizes both digital and analog equipment, imposing the need for suitable interfaces. Such interfaces typically provide conversion into digital form of sampled voltages and currents and conversion of digital information into voltage and current.

The MVME605 Four Channel Analog Output I/Omodule is a data distribution device. It can accept 12-bit offset binary data from a host on the Motorola I/O Channel, convert the data into voltage and current and place the equivalent analog values on any of four analog output channels. One of three voltage ranges can be jumper selected for each individual channel: ±2.5 V, ±5.0 V or ±10 V. If the power input pins of the analog output connector are supplied with V_{CC} of +10 V to +30 V, each output can also provide a constant 4–20 mA unipolar current. A functional block diagram of MVME605 is shown in Figure 1.

The MVME605 Four Channel Analog Output I/Omodule is a single high Eurocard form factor module containing four digital to analog converters serviced by a common I/O channel interface. An individual operational amplifier transforms the output of each D/A converter into an equivalent voltage for output. Two op amps and a final transistor stage provide proportional current from each D/A converter for output to an analog channel. The module has provisions for recalibration in the event of D/A replacement.

Analog Output Module Operation

The MVME605 and its functions are accessible to a user program via eight adjacent memory locations in the I/O Channel address space. On board decoding reserves two locations for each D/A channel relative to the jumper selected base address. The lower location of a channel pair is accessed to write the eight least significant bits of the input data into the internal register of the associated D/A converter. The higher location of a channel pair is accessed to write the four most significant bits of the input data into the internal register of the D/A converter. Figure 2 diagrams the MVME605 memory map. Table 1 shows the functions of the active bits in an MVME605 address word.

Table 2 for a few representative values written into the input register of a D/A converter lists the corresponding constant current outputs and corresponding output voltages for the three jumper selectable ranges.

FIGURE 1 — MVME605 Functional Block Diagram

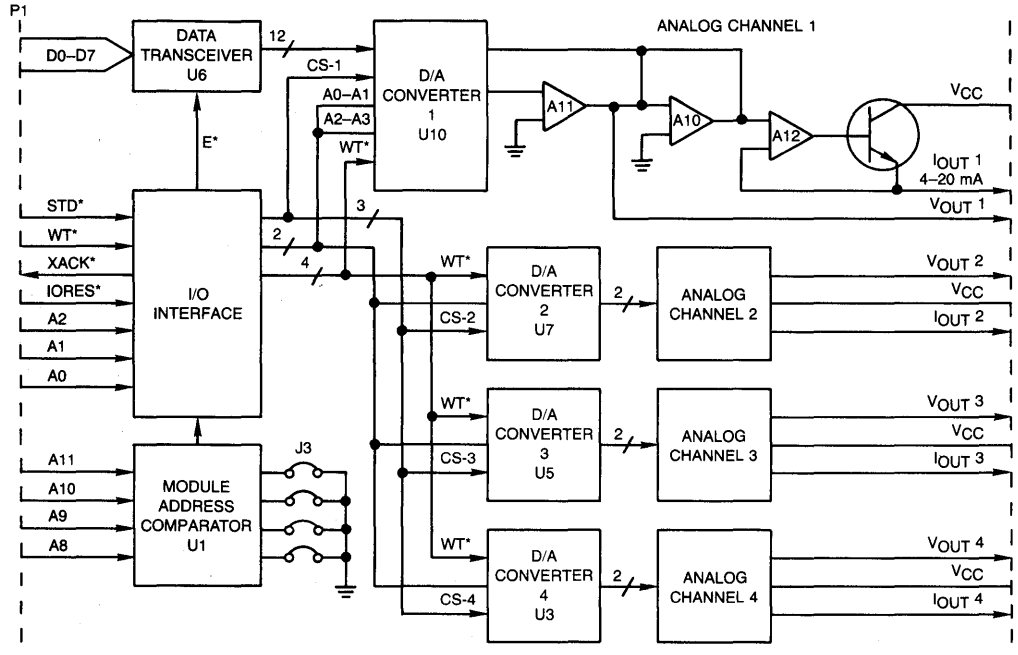
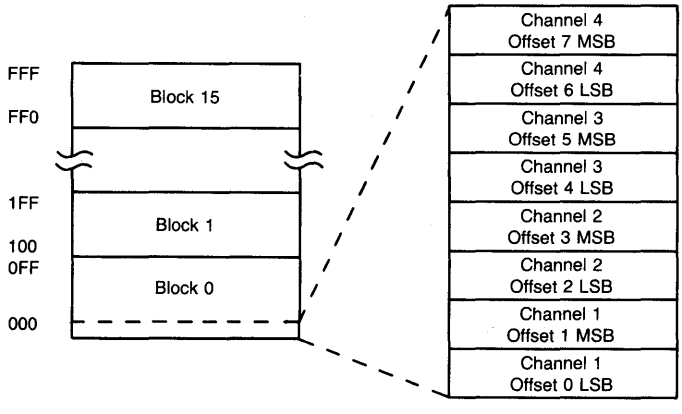


FIGURE 2 — MVME605 Memory Map



NOTE:
As shown, all header J3 jumpers in

TABLE 1 — Address Word Bit Functions

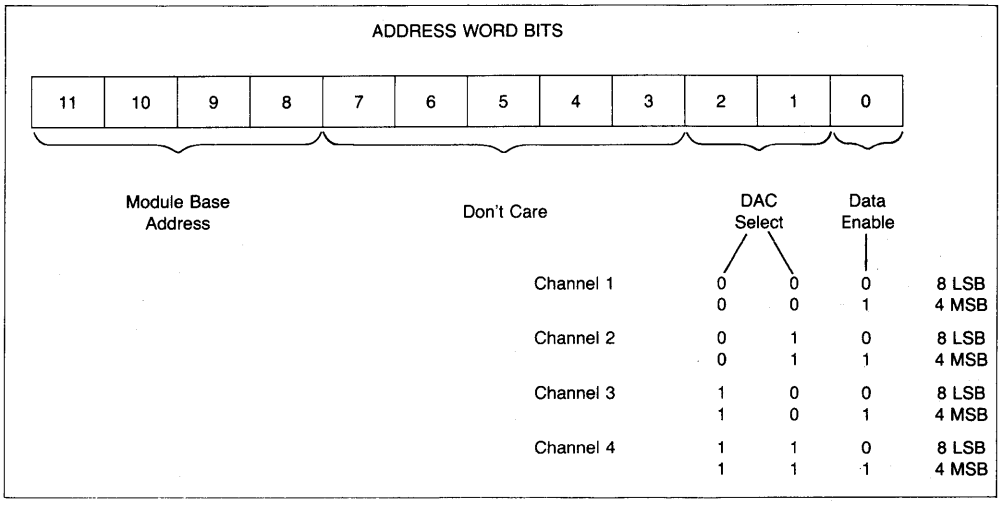


TABLE 2 — Outputs for Selected Input Values

Value Input To D/A	±10 V Output Voltage	±5.0 V Output Voltage	±2.5 V Output Voltage	4–20 mA Output Current
FFF	+10.0 V	+5.00 V	+2.500 V	20 mA
E00	+ 7.5 V	+3.75 V	+1.875 V	16 mA
C00	+ 5.0 V	+2.50 V	+1.250 V	12 mA
A00	+ 2.5 V	+1.25 V	+0.625 V	8 mA
800	0.0 V	0.00 V	0.000 V	4 mA
600	− 2.5 V	−1.25 V	−0.625 V	NA
400	− 5.0 V	−2.50 V	−1.250 V	NA
200	− 7.5 V	−3.75 V	−1.875 V	NA
000	−10.0 V	−5.00 V	−2.500 V	NA

Base Address Selection

The MVME605 has a header for jumper selection of a base address in the I/O Channel address space. Selection is implemented using the four highest order of the 12 address lines allowing the base address of the module to be located on any 256 byte boundary throughout the space.

Seen from a host, I/O Channel memory space (12 address lines, 4 K bytes) begins at an address determined by the particular host. The current addresses are F80000 (M68KVM02-3), FA0000 (M68KVM03), FE6000 (MVME110-1) and F1C000 (VME/10). Because of the manner in which the VMEbus is interfaced to the I/O Channel address bus, only the odd MC68000 addresses are used.

Software Driver

A driver is available for use in systems utilizing the RMS68K Executive Kernel. The MVME605 driver is an I/O handler that provides the device dependent portion of the software interface required for a D/A system. It has features that facilitate use of a single module by multiple tasks, execution of a prescribed sequence of timed control events and other data distribution activities. For users desiring to write their own driver, a manual "Guide to Writing a Device Driver," part number M68DRVDG is available.

MVME605 Usage

MVME110	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer
VME/10	Microcomputer System

MVME600 Series

The MVME600 Series is a family of I/Omodule peripheral interface cards designed for modular low-cost applications. These modules are mechanically compatible with a single Eurocard form factor and will operate from the Motorola I/O Channel. Use of the I/O Channel allows users to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

I/O Channel

The I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support Motorola's modular product families: VMEmodule and VERSAmodule.

Specifications

The general and the analog and digital specifications of the MVME605 are shown in Tables 3 and 4 respectively.

Analog Output Module Connectors

Table 5 provides the pin assignments and signal descriptions for analog output connector J1. The I/O channel connector P1 mechanical and electrical characteristics are fully described in Motorola I/O Channel Specification M68R1OCS.

Table 3 — MVME605 Specifications — General

Characteristics	Specifications
Power	
Requirements	+5.0 Vdc @ 1.1 A maximum +12 Vdc @ 60 mA maximum −12 Vdc @ 110 mA maximum
Temperature	
Operating	0 to +70°C
Storage	−25° to +85°C
Humidity	0% to 95%, non-condensing
Mechanical Specifications	Refer to M68RIOCS/D2, Input/Output Channel Specifications Manual
Board Size	Single High Eurocard
Height x Depth	5.12 in. (130 mm) x 7.40 in. (188 mm)
Thickness	0.83 in. (21 mm)
Connectors	P1 64-Pin PCB J1 Amp 2-164306 or equivalent

TABLE 4 — MVME605 Specifications — Analog and Digital

Characteristics	Specifications
Analog Section	
Full Scale Output Ranges	±2.5 Vdc, ±5.0 Vdc, ±10.0 Vdc
Voltage Mode Output Current	2.0 mA maximum
Current Loop Output	4–20 mA
Transfer Characteristics	
Conversion Time	0.5 μs to 0.1% ±1.0 LSB (typ. at +25°C)
Sample Time Per Channel	
Voltage Mode	1.5 μs (10 V step)
Current Mode	5.0 μs
Digital Section	
D/A Input Coding	Offset binary
Addressing	Fixed on-board decoding
I/O Channel	
Address	12 bits — 4 MSB used to define MVME605 address (jumper selectable)
Data	8 bits, bidirectional
Control	4 lines: Module Reset (IORES*), Transmit Acknowledge (XACK*), Strobe (STR*), and Write (WT*)

TABLE 5 — Analog Output Connector J1

Pin Number	Signal Mnemonic	Signal Name and Description
D2	P/S1	CHANNEL 1 POWER SUPPLY — Input connection for user-supplied V_{CC} of +10 V to +30 V.
D4	VOUT1	CHANNEL 1 VOLTAGE OUTPUT — bipolar output selectable as either ±10 V, ±5.0 V, or 2.5 V.
D6	IOUT1	CHANNEL 1 CURRENT OUTPUT — With user-supplied V_{CC}, constant 4–20 mA unipolar current output.
D8	P/S2	CHANNEL 2 POWER SUPPLY — same as P/S1 on pin D2.
D10	VOUT2	CHANNEL 2 VOLTAGE OUTPUT — same as VOUT1 on pin D4.
D12	IOUT2	CHANNEL 2 CURRENT OUTPUT — same as IOUT1 on pin D6.
D14	P/S3	CHANNEL 3 POWER SUPPLY — same as P/S1 on pin D2.
D16	VOUT3	CHANNEL 3 VOLTAGE OUTPUT — same as VOUT1 on pin D4.
D18	IOUT3	CHANNEL 3 CURRENT OUTPUT — same as IOUT1 on pin D6.
D20	P/S4	CHANNEL 4 POWER SUPPLY — same as P/S1 on pin D2.
D22	VOUT4	CHANNEL 4 VOLTAGE OUTPUT — same as VOUT1 on pin D4.
D24	IOUT4	CHANNEL 4 CURRENT OUTPUT — same as IOUT1 on pin D6.
D26–D32		Not used.
Z2–Z32	GND	GROUND

Ordering Information

Part Number	Description
MVME605	Analog Output I/Omodule providing four independent output channels with full scale output ranges of ±2.5 V, ±5 V and ±10 V having 12 bit resolution, ±0.1% F.S. ±1 LSB accuracy and 5 μs conversion time. Also providing simultaneous 4–20 mA constant unipolar current output (with user supplied V_{CC} of 10 to 30 Vdc on power supply pins of analog output connector).
MVME605/D1	Analog Output I/Omodule User's Manual.

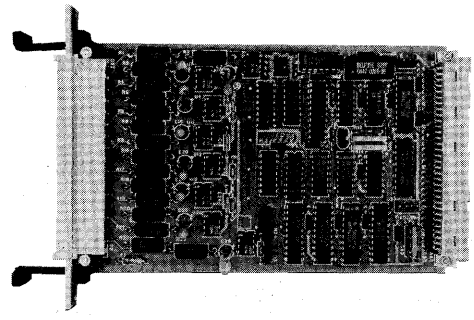
Related Documentation

M68RIOCS/D2	Input/Output Channel Specification Manual
M68DRVGD/D1	Guide to Writing a Device Driver
M68KRMS68K	M68000 Realtime Multitasking Software User's Manual
M68SYSGEN	System Generation Facility User's Manual

Modules in the MVME600 Series

MVME600/601	Analog Input and Expander I/Omodules
MVME610	Opto Isolated 120/240 Vac Input I/Omodule
MVME615	Opto Isolated 120/240 Vac Output I/Omodule with zero crossing switching
MVME616	Opto Isolated 120/240 Vac Output I/Omodule with non-zero crossing switching
MVME620	Opto Isolated 60 Vdc Input I/Omodule
MVME625	Opto Isolated 60 Vdc Output I/Omodule

AC Input I/Omodule



- Eight Independent Channels
- 90 Vac to 264 Vac Input Range
- Program Selectable Interrupt Generation, Four Channels
- Program Selectable Mode of Input Transition Sense for Interrupt Generation
- Jumper Selectable Interrupt Levels
- Jumper Selectable Base Address - Any of 16 Four Byte Blocks in the 000 to 03F I/O Channel Address Range
- Accessible to Host at two Adjacent I/O Channel Locations
- 2500 Vrms Isolation between Channels
- Motorola I/O Channel Compatible
- Single High Eurocard Form Factor
- 0° C–70° C Operating Temperature Range

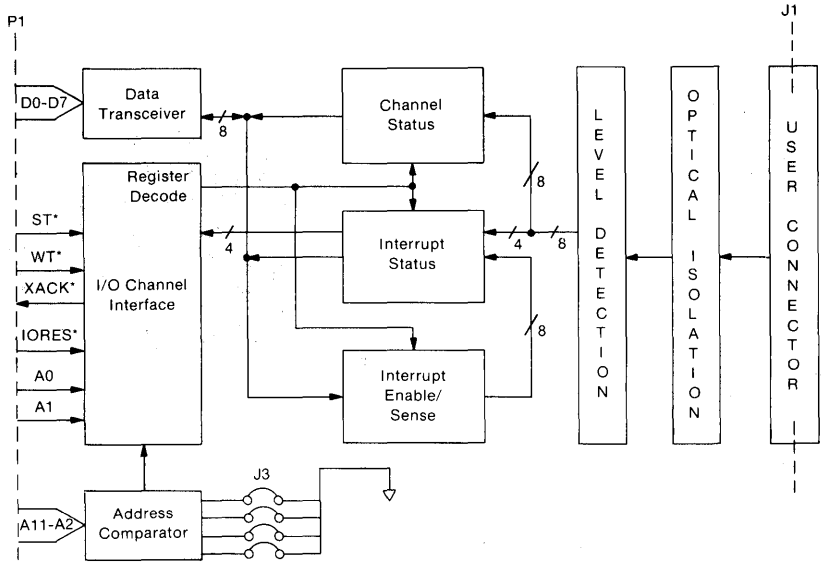
A microcomputer system for the process control, automatic test and other industrial and laboratory automation environments often utilizes both digital and analog equipment, imposing the need for suitable interfaces. Such interfaces typically provide conversion into digital form of sampled voltages and currents and conversion of digital information into voltage and current. Frequently an interface is needed for the simple determination of the operating state of an ac device, i.e., whether or not it is energized.

The MVME610 AC Input I/Omodule (ACIM) is a data acquisition device that can indicate to a host on the Motorola I/O Channel the energized/non-energized states of ac equipment connected to its eight input channels. All eight ACIM channels set a status flag on sensing an energized input and four channels can be programmed to place an interrupt on the I/O Channel on sensing a transition to the energized or de-energized state, as programmed. The ACIM has a header for selection by jumper of one of the four I/O Channel interrupt priority levels.

GENERAL DESCRIPTION

The AMIC has three general sections: (1) an analog input section which optically isolates the level detection circuitry from the input logic, (2) three registers which facilitate host/ACIM communications and, (3) an I/O Channel interface with data bus transceivers, an address bus decoder/comparator and driver/receivers for the I/O Channel control and interrupt lines. Figure 1 is a functional block diagram which shows the three general sections of the ACIM.

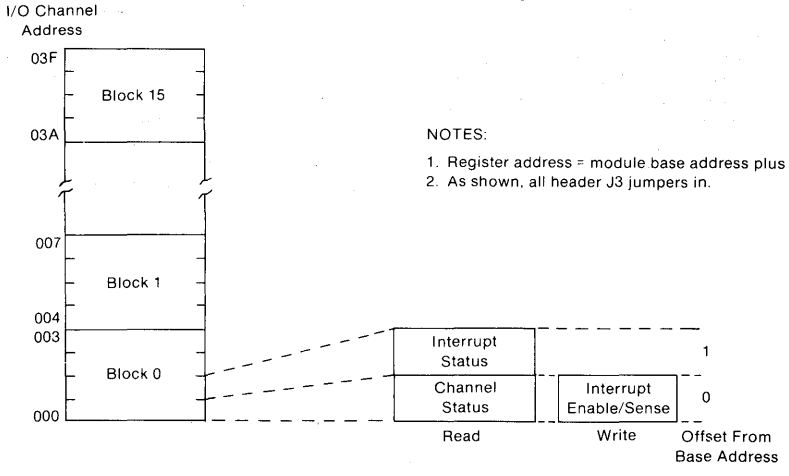
FIGURE 1 — MVME610 Functional Block Diagram



Host/ACIM Communications

An application driver obtains input channel status and interrupt status information from the ACIM by means of two read only registers and establishes interrupt channels and their initiation sense using a write only register. Seen from the host, the read only channel status register and the write only interrupt enable/sense register are both accessed at the lowest location of the ACIM address block in the I/O Channel space, i.e., the ACIM base address. The other read only register, interrupt status, is accessed at the base address plus one. This scheme is shown in the memory map of Figure 2.

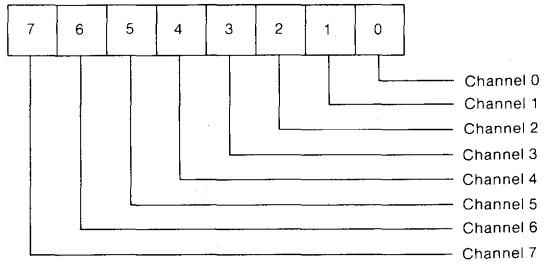
FIGURE 2 — MVME610 Memory Map



Channel Status Register

Bits 0 through 7 of the read only channel status register serve as flags which reflect the current state of ac input channels 0 through 7, respectively. Transitions between the energized and de-energized states are detected on each input channel and the corresponding register bits are updated accordingly. A logic "1" register bit indicates that the corresponding input channel is currently energized. The read only channel status register bit/channel correspondence is shown in Figure 3.

FIGURE 3 — Channel Status Register

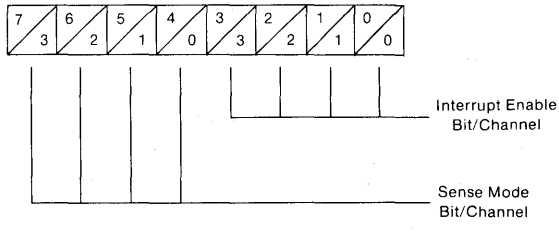


Interrupt Enable/Sense Register

The write only interrupt enable/sense register has two uses: select the mode of interrupt generation and enable/disable transmission of interrupts. On channels 0 through 3, an interrupt is generated each time the transition specified by the currently selected mode is sensed on an input. Transmission of the interrupt to the I/O channel is enabled by writing a logic "1" in the interrupt enable/sense register bit corresponding to that channel. A logic "0" disables transmissions. Register bits 0 through 3 correspond to channels 0 through 3 respectively. A host may assert the I/O Channel control signal Input/Output Reset (IORES*) to simultaneously disable interrupt transmission on all four channels. The ACIM has a header for jumper selection of one of the four prioritized I/O Channel interrupt lines.

Interrupt enable/sense register bits 4 through 7 are used to select the sense mode for channels 0 through 3, respectively. A logic "1" written into one of bits 4 through 7 obtains generation of an interrupt when the corresponding channel becomes energized. A logic "0" obtains generation of an interrupt on de-energization. Assertion of IORES* clears bits 4 through 7 thus all four channels are simultaneously placed in the sense de-energization mode. The bit functions of the write only interrupt enable/sense register are shown in Figure 4.

FIGURE 4 — Interrupt Enable/Sense Register



Interrupt Status Register

The read only interrupt status register provides two kinds of information: the state of the interrupt enable bits in the interrupt enable/sense register and indication of pending interrupts generated by enabled channels. The current states of interrupt enable/sense register bits 0 through 3 are reflected by interrupt status register bits 4 through 7 providing an image for use by an application driver.

Pending interrupts and their sources are flagged by interrupt status register bits 0 through 3, respectively. According to the programmed value of the corresponding bit in the interrupt enable/sense register, a flag bit is set to 1 when the respective channel is energized or de-energized and an interrupt is placed on the I/O Channel interrupt line of strap selected priority. All four flag bits are cleared when the register is read. Organization of the interrupt status register is shown in Figure 5.

Software Driver

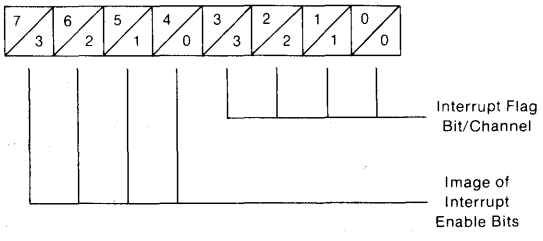
A software driver is available for use in systems utilizing the RMS68K executive kernel or the VERSAdos Operating System. For users desiring to write their own driver, a manual, "Guide to Writing a Device Driver", part number M68DRVGD, is available.

The 610/620 driver is an I/O handler which runs under the RMS68K executive kernel. It provides the device dependent portion of the software interface required for a system using up to 16 MVME610, or up to 16 MVME620 or any combination of up to 16 of these modules.

The 610/620 driver is re-entrant so that only a single copy is needed by a user task to control the 128 analog input channels that would be provided by 16 modules, the maximum number supported by the Motorola I/O Channel.

The 610/620 driver assumes that a base address has been selected, an I/O Channel interrupt priority level has been selected and a maximum interrupt service queue size defined and that these were identified during system generation.

FIGURE 5 — Interrupt Status Register



Base Address Selection

The MVME610 has a header for jumper selection of a base address in the I/O Channel address space. Selection is implemented so that a base address may be chosen on any four byte boundary in the 000 to 03F I/O Channel address range.

Seen from a host, I/O Channel address space (12 address lines, 4K bytes) begins at an address determined by the particular host. The current hexadecimal addresses are F80000 (M68KVM02-3), FA0000 (M68KVM03), FE6000 (MVME110-1) and F1C000 (VME/10). Because of the manner in which the VMEbus is interfaced to the I/O Channel address bus, only the odd MC68000 addresses are used.

Interrupt Selection

To facilitate application in systems using multiple interrupt priority levels, the MVME610 has a header which permits connection, by jumper, to one of the four prioritized I/O Channel interrupt lines. This sets the interrupt priority level for all eight ACIM channels.

Each of the current Motorola Monoboard Microcomputers (MVME110-1, VME/10, M68KVM02-3 and M68KVM03) provides a way for a device on the I/O Channel to Interrupt the MC68000 and for the interrupt to be captured by the Channel Management Routine (CMR) of the RMS68K executive kernel. When CMR polls the various I/O drivers of devices having the priority level of the interrupt, the 610/620 driver will claim and process the interrupt, if MVME610 or MVME620 generated.

The basic function of the driver is to manage the logical connections between tasks and MVME610 and MVME620 input channels. To do this it offers five user commands: open channels (INOPEN), close channels (INCLOS), read input status (INSTAT), enable interrupts (INENAB) and disable interrupts (INDSAB). The driver allows a single MVME610 or MVME620 to be shared by multiple task within any session. However, for a single channel, only one outstanding open channel command is allowed at one time.

MVME610 Usage

MVME110-1	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer
VME/10	Microcomputer System

MVME600 Series

The MVME600 Series is a family of I/Omodule peripheral interface cards designed for modular low cost applications. These modules are mechanically compatible with a single Eurocard form factor and will operate from the Motorola I/O Channel. Use of the I/O Channel allows users to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

I/O Channel

The I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support Motorola's modular product families: VMEmodule and VERSAmodule.

Specifications

The specifications of MVME610 are shown in Table 1.

TABLE 1 — MVME610 Specifications

Characteristic	Specification
Power Requirements	
I/O Channel Interface Section	800 mA (typical) @ +5 Vdc 950 mA (maximum) @ +5 Vdc
Isolated Input Section	8.2 mA (maximum) per channel @ 240 Vac
Input Protection	
Surge Voltage	1200 V (maximum) for 100 ms
Temperature	
Operating	0 to 70° C
Storage	-40 to +85° C
Humidity	0% to 90% non-condensing
Mechanical Specifications	
Height x Depth	5.12 in. (130 mm) x 7.0 in. (178 mm)
Thickness	0.83 in. (21 mm)
Connectors	
P1	64-pin PCB
J1	DIN 41612 double row (d & z) 32-pin male

AC Input Module Connectors

Table 2 provides the pin assignments and signal escriptions for ac input connector J1. The I/O Channel connector P1 mechanical and electrical characteristics are fully described in Motorola I/O Channel Specification M68RIOCS.

TABLE 2 — User Equipment Interface Connector J1 Signals

Pin Number	Signal Mnemonic	Signal Name and Description
Z2-Z32	GND	COMMON — Tied to ground through fuse F1
D2	V1	Channel 0
D4	V1	Channel 0
D6	V2	Channel 1
D8	V2	Channel 1
D10	V3	Channel 2
D12	V3	Channel 2
D14	V4	Channel 3
D16	V4	Channel 3
D18	V5	Channel 4
D20	V5	Channel 4
D22	V6	Channel 5
D24	V6	Channel 5
D26	V7	Channel 6
D28	V7	Channel 6
D30	V8	Channel 7
D32	V8	Channel 7

Ordering Information

Part Number	Description
MVME610	AC Input I/Omodule having eight independent differential channels accepting inputs ranging from 90 to 264 Vac, of which four channels are programmable for interrupt generation and mode of interrupt generation. Includes User's Manual.
MVME610/D1	AC Input Module User's Manual

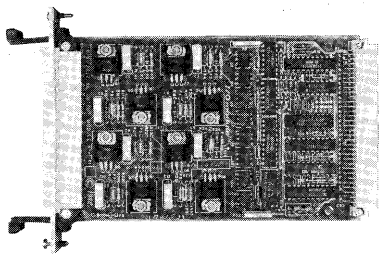
Related Documentation

M68RIOCS/D2	Input/Output Channel Specification Manual
M68DRVGD/D1	Guide to Writing a Device Driver
M68KRMS68K	M68000 Realtime Multitasking Software User's Manual
M68KSYSGEN	System Generation Facility User's Manual

Modules in the MVME600 Series

MVME600/601	Analog Input and Expander I/Omodules
MVME605	Analog Output I/Omodule
MVME615	Opto Isolated 120/240 Vac Output I/Omodule with Zero Crossing Switching
MVME616	Opto Isolated 120/240 Vac Output I/Omodule with Non-Zero Crossing Switching
MVME620	Opto Isolated 60 Vdc Input I/Omodule
MVME625	Opto Isolated 60 Vdc Output I/Omodule

AC Output Module



- Eight Independent Channels
- 2.0 A per Channel Output Current (Convection Cooling)
- 120 to 240 Vac Output Voltage
- 2500 V_{rms} Isolation Between Channels
- Zero-Crossing (MVME615) or Non-Zero Crossing (MVME616) Switching
- Fuse Overload Protection - 2.0 A Each Channel
- Jumper Selectable Base Address - Choice of Any Even Location in the 001-01F I/O Channel Address Range
- Accessible to Host at Single I/O Channel Location
- Motorola I/O Channel Compatible
- Single High Eurocard Form Factor
- 0° C-70° C Operating Temperature Range

A microcomputer system for the process control, automatic test and other industrial and laboratory automation environments often utilizes both digital and analog equipment, imposing the need for suitable interfaces. Such interfaces typically provide conversion into digital form of sampled voltages and currents and conversion of digital information into voltage and current. Frequently a device is needed for the simple energizing and de-energizing of attached ac devices.

The MVME615 and MVME616 Analog Output I/Omodules are used to switch ac devices on and off. A host on the Motorola I/O Channel can set or clear register bits in these modules to turn on or off any of eight output channels. Up to 2.0 A of channel load current can be switched at 120 or 240 Vac over an operating temperature range of 0° to 70° C. Load switching triggered by zero crossing detection is provided by MVME615, conventional switching by MVME616. For host use, both have a status register in which the states of individual bits reflect the energized or non-energized states of the corresponding channel loads. A functional block diagram of MVME615 and MVME616 is shown in Figure 1. For convenience in the following discussions, the acronym ACOM, for ac output module, is used to refer to either the MVME615 or MVME616.

GENERAL DESCRIPTION

The ACOM is a single high Eurocard form factor I/Omodule having eight output channels serviced by a common I/O channel interface. The output channels are protected against voltage transient peaks and are fused against current overloads. Channel loads are switched in and out using triacs which are optically isolated from the channel switch and channel status registers and from the I/O Channel interface.

Channel Switch and Channel Status Registers

Host/ACOM communications are facilitated by the channel switch and channel status registers in which bits 0 through 7 correspond to channels 0 through 7, respectively. As shown in the ACOM memory map of Figure 2, both the registers are accessed at the same I/O Channel location.

A channel is energized by setting the bit in the channel switch register corresponding to the desired channel. Similarly, a channel is de-energized by clearing the proper bit. The trigger point value of the 120 or 140 Vac output supply is determined on MVME616 by fixed resistors. On MVME615, triggering occurs when the module detects the zero crossing point of the output supply voltage swing.

Bits in the channel status register are updated to reflect the states of output channels as they are energized and de-energized. This allows a host to test the current state of a channel, if required.

Base Address Selection

Each ACOM has a header for jumper selection of a base address in the I/O Channel address space. Selection is implemented so that a base address may be located on any two byte boundary in the 000 to 01F I/O Channel address range.

Seen from a host, I/O Channel address space (12 address lines, 4K bytes) begins at an address determined by the particular host. The current addresses are F80000 (M68KVM02-3), FA0000 (M68KVM03), FE6000 (MVME110-1) and F1C000 (VME/10). Because of the manner in which the VMEbus is interfaced to the I/O Channel address bus, only the odd MC68000 addresses are used.

FIGURE 1 — MVME615/MVME616 Functional Block Diagram

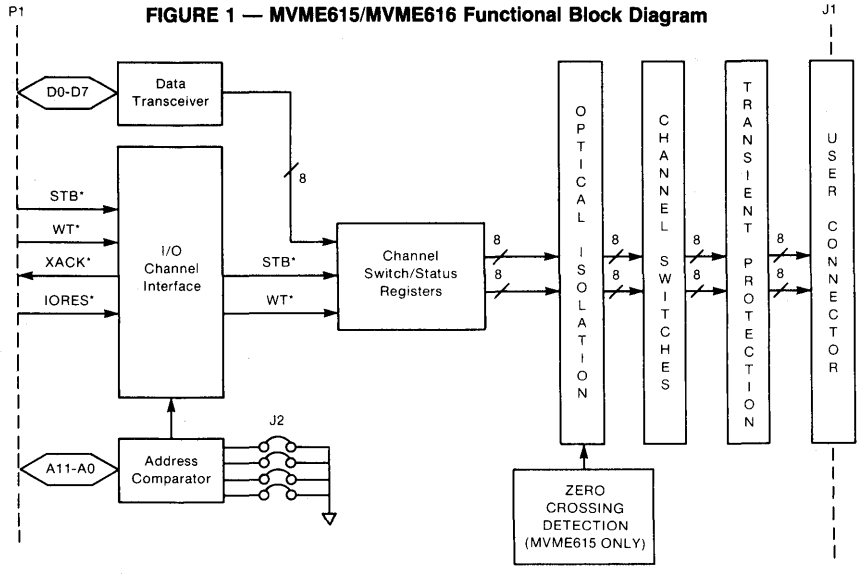
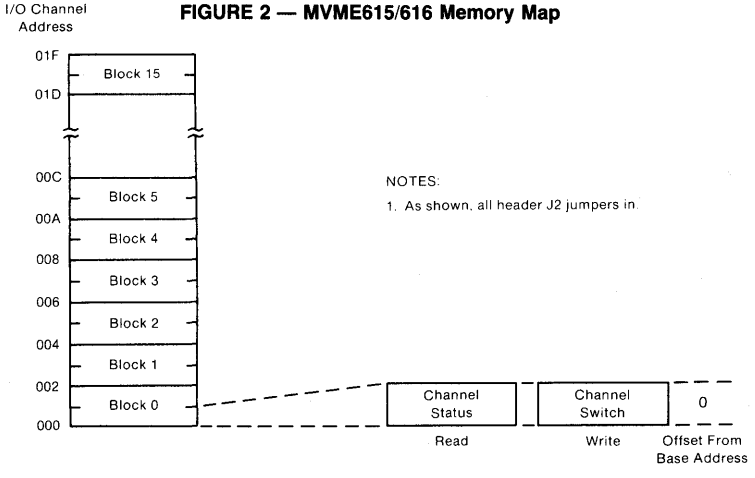


FIGURE 2 — MVME615/616 Memory Map



Software Driver

A software driver is available for use in systems utilizing the RMS68K executive kernel or the VERSAdos Operating System. For users desiring to write their own driver, a manual, "Guide to Writing a Device Driver", part number M68DRVGD, is available. The 615/616 driver is an I/O handler which runs under the RMS68K executive kernel. It provides the device dependent portion of the software interface required for a system using any combination of up to 16 MVME615 or MVME616 modules.

The 615/616 driver is re-entrant so that only a single copy is needed by a user task to control the output channels that would be provided by these modules.

The 615/616 driver assumes that a unique base address has been selected for each ACOM and that the number of ACOM's and the number of valid ACOM/user combinations were identified during system generation.

Management of the logical connections between user tasks and the system output channels is the basic function of the 615/616 driver. To achieve this function the capabilities of the Channel Management Routine (CMR) of the RMS68K executive kernel are extended to allow user tasks to open and close a channel, start and stop output on a channel, to pulse an output channel and to read the status of an output channel. The respective commands for these extended capabilities are: ACOPEN, ACOCLS, ACOOUT, ACOOFF, ACOPLS and ACOSTA. The driver allows a single MVME615 or MVME616 to be shared by multiple tasks within any session; however a single channel may be assigned to but one task at a time.

MVME615/MVME616 Usage

MVME110-1	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer
VME/10	Microcomputer System

MVME600 Series

The MVME600 Series is a family of I/Omodule peripheral interface cards designed for modular low cost applications. These modules are mechanically compatible with a single Eurocard form factor and will operate from the Motorola I/O Channel. Use of the I/O Channel allows users to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

I/O Channel

The I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support Motorola's modular product families: VMEmodule and VERSAmodule.

Specifications

The specifications of MVME615/616 are shown in Table 1.

TABLE 1 — MVME615/616 Specifications

Characteristic	Specification
Power Requirements	+5 Vdc @ 700 mA, maximum +5 Vdc @ 400 mA, typical
Output Device On-State Voltage Output Voltage Capability Output Current Capability Output Surge Current	0.5 V typical, @ 2.0 A 260 Vac maximum, per channel 2.0 A per channel 4.0 A maximum, for 5.0 seconds
Temperature Operating Storage	 0 to 70° C -40 to +85° C
Humidity	0% to 90% (non-condensing)
Mechanical Specifications Board Size Height x Depth Thickness Connector P1 Connector J1	 Single High Eurocard 7.00 in. (178 mm) x 5.12 in. (130 mm) 0.83 in. (21 mm) 64-pin PCB 32 pin DIN, type F, male

Analog Output Module Connectors

Table 2 provides the pin assignments and signal descriptions for user equipment interface connector J1.

The I/O Channel connector P1 mechanical and electrical characteristics are fully described in Motorola I/O Channel Specification M68RIOCS.

TABLE 2 — User Equipment Interface Connector J1 Signals

Pin Number	Signal Mnemonic	Signal Name and Description
Z2-Z32	GND	COMMON (entire row) - Tied to logic ground through fuse F9
D2 & D4		Channel 0
D6 & D8		Channel 1
D10 & D12		Channel 2
D14 & D16		Channel 3
D18 & D20		Channel 4
D22 & D24		Channel 5
D26 & D28		Channel 6
D30 & D32		Channel 7

Ordering Information

Part Number	Description
MVME615	Analog Output I/Omodule providing eight optically isolated, independent channels having a maximum output current capability of 2.0 A at 120 or 240 Vac and providing zero-crossover switching operation. Includes User's Manual.
MVME616	Identical to MVME615 but without zero-crossover switching feature.
MVME615/D1	MVME615/MVME616 AC Output Module User's Manual

Related Documentation

M68RIOCS/D2	Input/Output Channel Specification Manual
M68DRVGD/D1	Guide to Writing a Device Driver
M68KRMS68K	M68000 Realtime Multitasking Software User's Manual
M68KSYSGEN	System Generation Facility User's Manual

Modules in the MVME600 Series

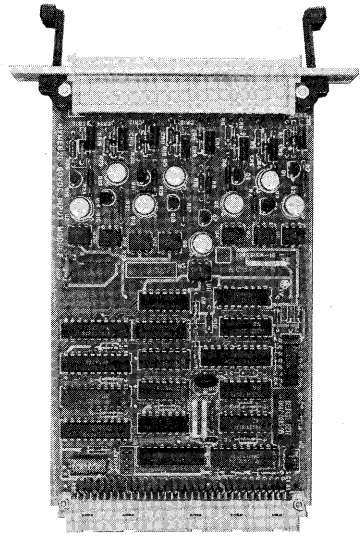
MVME600/601	Analog Output and Expander I/Omodules
MVME605	Analog Output I/Omodule
MVME610	Opto Isolated 120/240 Vac Input I/Omodule
MVME620	Opto Isolated 60 Vdc Input I/Omodule
MVME625	Opto Isolated 30 Vdc Output I/Omodule

DC Input I/Omodule

- Eight Independent Differential Channels
- 10 Vdc to 70 Vdc Input Range
- Program Selectable Interrupt Generation-Four Channels
- Program Selectable Mode of Input Transition Sense for Interrupt Generation
- Jumper Selectable Interrupt Levels
- Jumper Selectable Base Address - Any of 16 Four Byte Blocks in the 000 to 03F I/O Channel Address Range
- Accessible to Host at two Adjacent I/O Channel Locations
- 2500 Vrms Isolation between Channels
- Reverse Voltage Protection
- Input Spike Protection of 10 kW for 1μs
- Motorola I/O Channel Compatible
- Single High Eurocard Form Factor
- 0° C-70° C Operating Temperature Range

A microcomputer system for the process control, automatic test and other industrial and laboratory automation environments often utilizes both digital and analog equipment, imposing the need for suitable interfaces. Such interfaces typically provide conversion into digital form of sampled voltages and currents and conversion of digital information into voltage and current. Frequently an interface is needed for the simple determination of the operating state of a dc device, i.e., whether or not it is energized.

The MVME620 DC Input I/Omodule (DCIM) is a data acquisition device that can indicate to a host on the Motorola I/O Channel the energized/non-energized states of dc equipment connected to its eight input channels. All eight

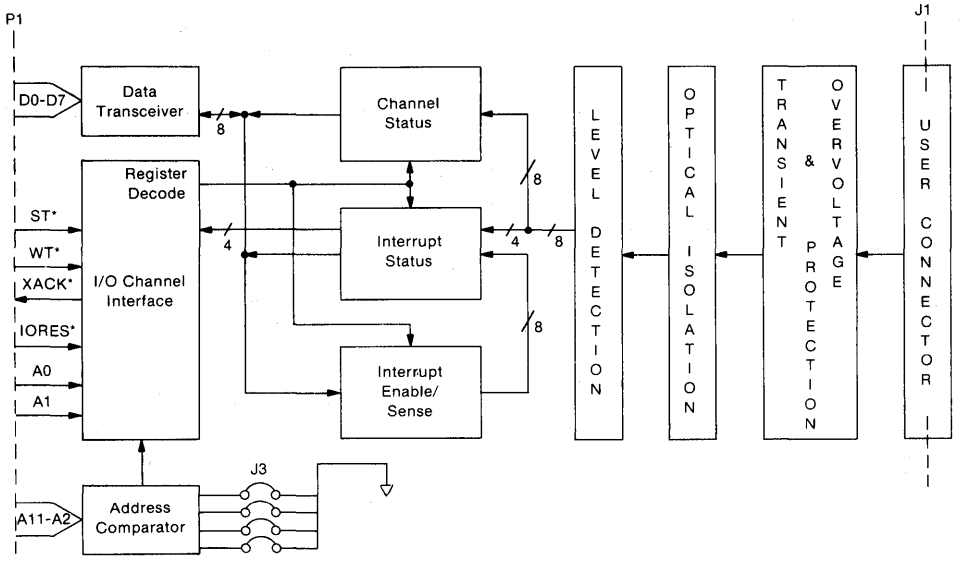


DCIM channels set a status flag on sensing an energized input and four channels can be programmed to place an interrupt on the I/O Channel on sensing a transition to the energized or de-energized state, as programmed. The DCIM has a header for selection by jumper of one of the four I/O Channel interrupt priority levels.

GENERAL DESCRIPTION

The DMIC has four general sections: (1) an input section which protects the circuitry against voltages of reverse polarity, transients and overvoltages, (2) an analog input section which optically isolates the level detection circuitry from the input logic, (3) an I/O Channel interface with data bus transceivers, an address bus decoder/comparator and driver/receivers for the I/O Channel control and interrupt lines. Figure 1 is a functional block diagram which shows the four general sections of the DCIM.

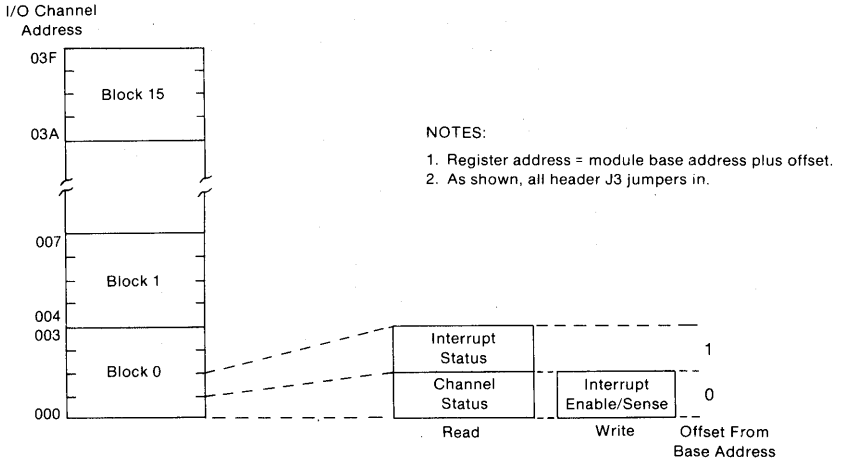
FIGURE 1 — MVME620 Functional Block Diagram



Host/DCIM Communications

An application driver obtains input channel status and interrupt status information from the DCIM by means of two read only registers and establishes interrupt channels and their initiation sense using a write only register. Seen from the host, the read only channel status register and the write only interrupt enable/sense register are both accessed at the lowest location of the DCIM address block in the I/O Channel space, i.e., the DCIM base address. The other read only register, interrupt status, is accessed at the base address plus one. This scheme is shown in the memory map of Figure 2.

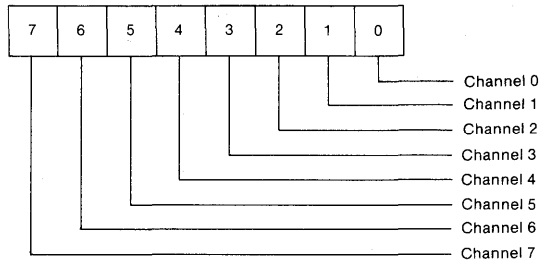
FIGURE 2 — MVME620 Memory Map



Channel Status Register

Bits 0 through 7 of the read only channel status register serve as flags which reflect the current state of dc input channels 0 through 7, respectively. Transitions between the energized and de-energized states are detected on each input channel and the corresponding register bits are updated accordingly. A logic "1" register bit indicates that the corresponding input channel is currently energized. The read only channel status register bit/channel correspondence is shown in Figure 3.

FIGURE 3 — Channel Status Register

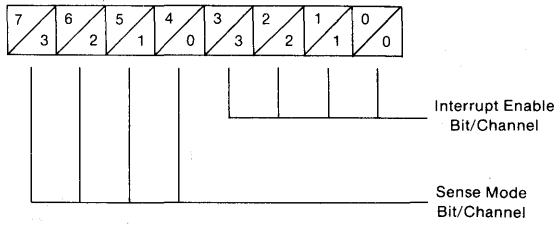


Interrupt Enable/Sense Register

The write only interrupt enable/sense register has two uses: select the mode of interrupt generation and enable/disable transmission of interrupts. On channels 0 through 3, an interrupt is generated each time the transition specified by the currently selected mode is sensed on an input. Transmission of the interrupt to the I/O channel is enabled by writing a logic "1" in the interrupt enable/sense register bit corresponding to that channel. A logic "0" disables transmissions. Register bits 0 through 3 correspond to channels 0 through 3 respectively. A host may assert the I/O Channel control signal Input/Output Reset (IORES*) to simultaneously disable interrupt transmission on all four channels. The ACIM has a header for jumper selection of one of the four prioritized I/O Channel interrupt lines.

Interrupt enable/sense register bits 4 through 7 are used to select the sense mode for channels 0 through 3, respectively. A logic "1" written into one of bits 4 through 7 obtains generation of an interrupt when corresponding channel becomes energized. A logic "0" obtains generation of an interrupt in de-energization. Assertion of IORES* clears bits 4 through 7 thus all four channels are simultaneously placed in the sense de-energization mode. The bit functions of the write only interrupt enable/sense register are shown in Figure 4.

FIGURE 4 — Interrupt Enable/Sense Register



Interrupt Status Register

The read only interrupt status register provides two kinds of information: the state of the interrupt enable bits in the interrupt enable/sense register and indication of pending interrupts generated by enabled channels. The current states of interrupt enable/sense register bits 0 through 3 are reflected by interrupt status register bits 4 through 7 providing an image for use by an application driver.

Pending interrupts and their sources are flagged by interrupt status register bits 0 through 3, respectively. According to the programmed sense of the corresponding bit in the interrupt enable/sense register, a flag bit is set to 1 when the respective channel is energized or de-energized and an interrupt is placed on the I/O Channel interrupt line of strap selected priority. All four flag bits are cleared when the register is read. Organization of the interrupt status register is shown in Figure 5.

Software Driver

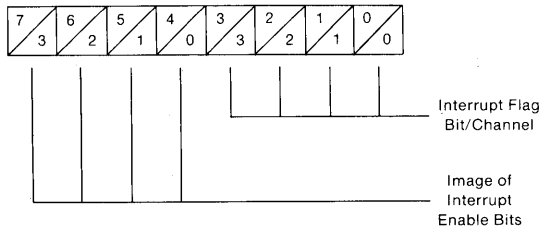
A software driver is available for use in systems utilizing the RMS68K executive kernel or the VERSAdos Operating System. For users desiring to write their own driver, a manual, "Guide to Writing a Device Driver", part number M68DRVGD, is available.

The 610/620 driver is an I/O handler which runs under the RMS68K executive kernel. It provides the device dependent portion of the software interface required for a system using up to 16 MVME610, or up to 16 MVME620 or any combination of 16 of these modules.

The 610/620 driver is re-entrant so that only a single copy is needed by a user task to control the 128 analog input channels that would be provided by 16 modules, the maximum number supported by the Motorola I/O Channel.

The 610/620 driver assumes that a base address has been selected, an I/O Channel interrupt priority level has been selected and a maximum interrupt service queue

FIGURE 5 — Interrupt Status Register



Base Address Selection

The MVME620 has a header for jumper selection of a base address in the I/O Channel address space. Selection is implemented so that a base address may be chosen on any four byte boundary in the 000 to 03F I/O Channel address range.

Seen from a host, I/O Channel address space (12 address lines, 4K bytes) begins at an address determined by the particular host. The current hexadecimal addresses are F80000 (M68KVM02-3), FA0000 (M68KVM03), FE6000 (MVME110-1) and F1C000 (VME/10). Because of the manner in which the VMEbus is interfaced to the I/O Channel address bus, only the odd MC68000 addresses are used.

Interrupt Selection

To facilitate application in systems using multiple interrupt priority levels, the MVME620 has a header which permits connection, by jumper, to one of the four prioritized I/O Channel interrupt lines. This sets the interrupt priority level for all eight DCIM channels.

size defined and that these were identified during system generation.

Each of the current Motorola Monoboard Microcomputers (MVME110-1, VME/10, M68KVM02-3 and M68KVM03) provides a way for a device on the I/O Channel to Interrupt the MC68000 and for the interrupt to be captured by the Channel Management Routine (CMR) of the RMS68K executive kernel. When CMR polls the various I/O drivers of devices having the priority level of the interrupt, the 610/620 driver will claim and process the interrupt, if MVME610 or MVME620 generated.

The basic function of the driver is to manage the logical connections between tasks and MVME610 and MVME620 input channels. To do this it offers five user commands: open channels (INOPEN), close channels (INCLOS), read input status (INSTAT), enable interrupts (INENAB) and disable interrupts (INDSAB). The driver allows a single MVME610 or MVME620 to be shared by multiple task within any session. However, a single channel may be assigned to one and only one task at a time.

MVME620 Usage

MVME110-1	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer
VME/10	Microcomputer System

MVME600 Series

The MVME600 Series is a family of I/Omodule peripheral interface cards designed for modular low cost applications. These modules are mechanically compatible with a single Eurocard form factor and will operate from the Motorola I/O Channel. Use of the I/O Channel allows users to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

I/O Channel

The I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support Motorola's modular product families: VMEmodule and VERSAmodule.

Specifications

The specifications of MVME620 are shown in Table 1.

TABLE 1 — MVME620 Specifications

Characteristic	Specification
Power Requirements	
I/O Channel Interface Section	800 mA (typical) @ +5 Vdc 950 mA (maximum) @ +5 Vdc
Isolated Input Section	6.5 mA (maximum) per channel @ 60 Vdc
Input Protection	
Surge Voltage	10 kW (maximum for 1.0 μs)
Temperature	
Operating	0 to 70° C
Storage	-40 to +85° C
Humidity	0% to 90% non-condensing
Mechanical Specifications	
Height x Depth	5.12 in. (130 mm) x 7.0 in. (178 mm)
Thickness	0.83 in. (21 mm)
Connectors	
P1	64-pin PCB
J1	DIN 41612 double row (d & z) 32-pin male

DC Input Module Connectors

Table 2 provides the pin assignments and signal descriptions for dc input connector J1. The I/O Channel connector P1 mechanical and electrical characteristics are fully described in Motorola I/O Channel Specification M68RIOCS.

TABLE 2 — User Equipment Interface Connector J1 Signals

Pin Number	Signal Mnemonic	Signal Name and Description
Z2-Z32	GND	COMMON — Tied to ground through fuse F1
D2	+V1	Positive lead of Channel 0
D4	-V1	Negative lead of Channel 0
D6	+V2	Positive lead of Channel 1
D8	-V2	Negative lead of Channel 1
D10	+V3	Positive lead of Channel 2
D12	-V3	Negative lead of Channel 2
D14	+V4	Positive lead of Channel 3
D16	-V4	Negative lead of Channel 3
D18	+V5	Positive lead of Channel 4
D20	-V5	Negative lead of Channel 4
D22	+V6	Positive lead of Channel 5
D24	-V6	Negative lead of Channel 5
D26	+V7	Positive lead of Channel 6
D28	-V7	Negative lead of Channel 6
D30	+V8	Positive lead of Channel 7
D32	-V8	Negative lead of Channel 7

Ordering Information

Part Number	Description
MVME620	DC Input I/Omodule having eight independent differential channels accepting inputs ranging from 10 to 70 Vdc, of which four channels are programmable for interrupt generation and mode of interrupt generation. Includes User's Manual.
MVME620/D1	DC Input I/O module User's Manual

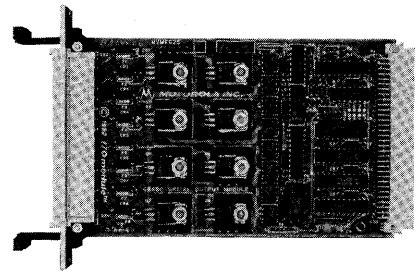
Related Documentation

M68RIOCS/D2	Input/Output Channel Specification Manual
M68DRVGD/D1	Guide to Writing a Device Driver
M68KRMS68K	M68000 Realtime Multitasking Software User's Manual
M68KSYSGEN	System Generation Facility User's Manual

Modules in the MVME600 Series

MVME600/601	Analog Input and Expander I/Omodule
MVME605	Analog Output I/Omodule
MVME610	Opto Isolated 120/240 Vac Input I/Omodule
MVME615	Opto Isolated 120/240 Vac Output I/Omodule with Zero Crossing Switching
MVME616	Opto Isolated 120/240 Vac Output I/Omodule with Non-Zero Crossing Switching
MVME625	Opto Isolated 60 Vdc Input I/Omodule
MVME625	Opto Isolated 60 Vdc Output I/Omodule

DC Output I/Omodule



- Eight Independent Channels
- 2.0 A per Channel Output Current (Convection Cooling)
- 10 to 60 Vdc Output Voltage Range
- 2500 V_{rms} Isolation Between Channels
- Fuse Overload Protection
- Zener Transient Protection
- Jumper Selectable Base Address - Choice of Any Even Location in the 001-01F I/O Channel Address Range
- Accessible to Host at Single I/O Channel Location
- Motorola I/O Channel Compatible
- Single High Eurocard Form Factor
- 0° C-70° C Operating Temperature Range

A microcomputer system for the process control, automatic test and other industrial and laboratory automation environments often utilizes both digital and analog equipment, imposing the need for suitable interfaces. Such interfaces typically provide conversion into digital form of sampled voltages and currents and conversion of digital information into voltage and current. Frequently an interface is needed for the simple energizing and de-energizing of attached dc devices.

The MVME625 Analog Output I/Omodule (DCOM) is used to switch dc devices on and off. A host on the Motorola I/O Channel can set or clear register bits to turn on or off any of eight output channels. Up to 2.0 A of channel load current can be switched over a 10 to 60 Vdc output voltage range and over an operating temperature range of 0° to 70° C. For host use, the DCOM has a status register in which the states of individual bits reflect the energized or non-energized states of the corresponding channel loads. A functional block diagram of MME625 is shown in Figure 1.

GENERAL DESCRIPTION

The DCOM is a single high Eurocard form factor I/Omodule having eight output channels serviced by a common I/O Channel interface. The output channels are protected against inductive load transients and are fused against current overloads. Channel loads are switched in and out using Darlingtons which are optically isolated from the channel switch and channel status registers and from the I/O Channel interface.

Channel Switch and Channel Status Registers

Host/DCOM communications are facilitated by the channel switch and channel status registers in which bits 0 through 7 correspond to channels 0 through 7, respectively. As shown in the DCOM memory map of Figure 2, both the registers are accessed at the same location.

A channel is energized by setting the bit in the channel switch register corresponding to the desired channel. Similarly, a channel is de-energized by clearing the proper bit.

Bits in the channel status register are updated to reflect the states of output channels as they are energized and de-energized. This allows a host to test the current state of a channel, if required.

Base Address Selection

Each DCOM has a header for jumper selection of a base address in the I/O Channel address space. Selection is implemented so that a base address may be located on any two byte boundary in the 000 to 01F I/O Channel address range.

Seen from a host, I/O Channel address space begins at an address determined by the particular host. The current addresses are F80000 (M68KVM02-3), FA0000 (M68KVM03), FE6000 (MVME110-1) and F1C000 (VME/10). Because of the manner in which the VMEbus is interfaced to the I/O Channel address bus, only the odd MC68000 addresses are used.

FIGURE 1 — MVME625 Functional Block Diagram

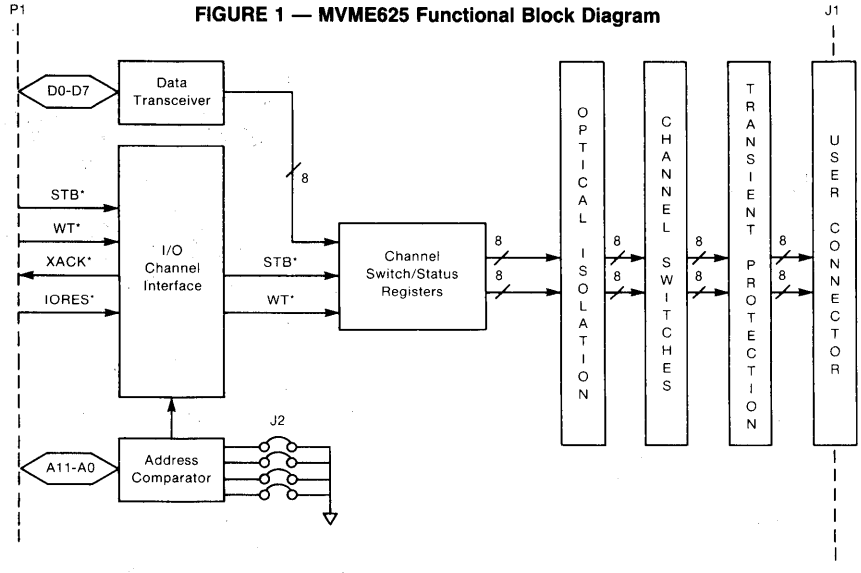
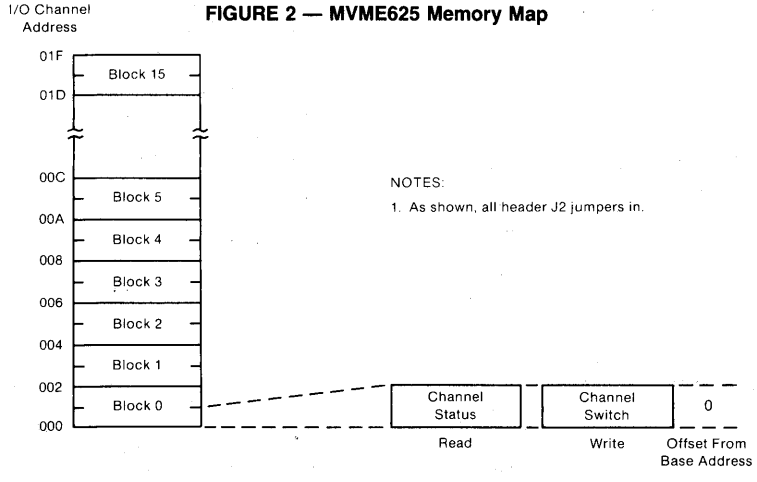


FIGURE 2 — MVME625 Memory Map



Software Driver

A software driver is available for use in systems utilizing the RMS68K executive kernel or the VERSAdos Operating System. For users desiring to write their own driver, a manual, "Guide to Writing a Device Driver", part number M68DRVGD, is available.

The 625 driver is an I/O handler which runs under the RMS68K executive kernel. It provides the device dependent portion of the software interface required for a system using MVME625 modules.

The 625 driver is re-entrant so that only a single copy is needed by a user task to control the output channels that would be provided by these modules.

DC Output Module Connectors

Table 2 provides the pin assignments and signal descriptions for user equipment and interface connector J1. The I/O Channel connector P1 mechanical and electrical characteristics are fully described in Motorola I/O Specification M68RIOCS.

TABLE 2 — User Equipment Interface Connector J1 Signals

Pin Number	Signal Mnemonic	Signal Name and Description
Z2-Z32	GND	COMMON (entire row) — Tied to logic ground through fuse F9
D2	+V1	Positive lead of Channel 0
D4	-V1	Negative lead of Channel 0
D6	+V2	Positive lead of Channel 1
D8	-V2	Negative lead of Channel 1
D10	+V3	Positive lead of Channel 2
D12	-V3	Negative lead of Channel 2
D14	+V4	Positive lead of Channel 3
D16	-V4	Negative lead of Channel 3
D18	+V5	Positive lead of Channel 4
D20	-V5	Negative lead of Channel 4
D22	+V6	Positive lead of Channel 5
D24	-V6	Negative lead of Channel 5
D26	+V7	Positive lead of Channel 6
D28	-V7	Negative lead of Channel 6
D30	+V8	Positive lead of Channel 7
D32	-V8	Negative lead of Channel 7

Ordering Information

Part Number	Description
MVME625	Analog Output I/Omodule providing eight optically isolated, independent channels having a maximum output current capability of 2.0 A at 10 to 60 Vdc. Includes User's Manual.
MVME625/D1	MVME625 DC Output Module User's Manual

Related Documentation

M68RIOCS/D2	Input/Output Channel Specification Manual
M68DRVGD/D1	Guide to Writing a Device Driver
M68KRMS68K	M68000 Realtime Multitasking Software User's Manual
M68KSYSGEN	System Generation Facility User's Manual

Modules in the MVME600 Series

MVME600/601	Analog Input and Expander I/Omodules
MVME605	Analog Output I/Omodule
MVME610	Opto Isolated 120/240 Vac Input I/Omodule
MVME615/616	Opto Isolated 120/240 Vac Output I/Omodule
MVME620	Opto Isolated 60 Vdc Input I/Omodule

The 625 driver assumes that a unique base address has been selected for each DCOM and that the number of DCOM's and the number of valid DCOM/user combinations were identified during system generation.

Management of the logical connections between user tasks and the output channels of system DCOM's is the basic function of the 625 driver. To achieve this function, the capabilities of the Channel Management Routine (CMR) of the RMS68K executive kernel are extended to allow user tasks to open and close a channel, start and stop output on a channel, to pulse an output channel and to read the status of an output channel. The respective commands for these extended capabilities are: DCOPEN, DCOCLS, DCOOUT, DCOOFF, DCOPLS and DCOSTA. The driver allows a single MVME625 to be shared by multiple tasks within any session; however, a single channel may be assigned to but one task at a time.

MVME625 Usage

MVME110-1	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer
VME/10	Microcomputer System

MVME600 Series

The MVME600 Series is a family of I/Omodule peripheral interface cards designed for modular low cost applications. These modules are mechanically compatible with a single Eurocard form factor and will operate from the Motorola I/O Channel. Use of the I/O Channel allows users to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

I/O Channel

The I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support Motorola's modular product families: VMEmodule and VERSAmodule.

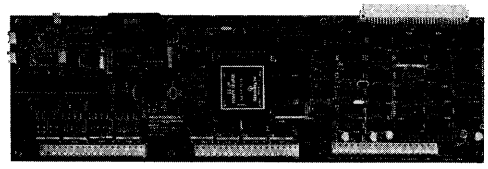
Specifications

The specifications of MVME625 are shown in Table 1.

TABLE 1 — MVME625 Specifications

Characteristic	Specification
Power Requirements	
Maximum	5 Vdc @ 700 mA
Typical	5 Vdc @ 400 mA
Output Device On-State Voltage	2.0 V (typical) @ 2.0 A
Output Current	
Continuous	2.0 A per channel
Surge (5-second)	4.0 A per channel
Temperature	
Operating	0 to 70°C
Storage	-40 to +85°C
Humidity	0% to 90% (non-condensing)
Mechanical Specifications	
Board Size	Single High Eurocard
Height x Depth	7.00 in. (178 mm) x 5.12 in. (130 mm)
Thickness	0.83 in. (21 mm)
Connector P1	64-pin PCB
Connector J1	32 pin DIN, type F, male

Remote Intelligent Analog-To-Digital Conversion Module



- 32 Single-Ended/16 Differential A/D Channels with:
  - Removable mass termination strips facilitate quick replacement of RAD1 module without disturbing field wiring
  - 0 to 10 V and ±10 V conversion ranges together with four programmable amplifier gain ranges produce the following full scale input signal ranges:
    Single-ended 0–10 V, 0–5 V, 0–2.5 V and 0–1 V
    Differential ±10 V, ±5 V, ±2.5 V and ±1 V
  - Handles 4-20 mA/10-50 mA instruments (with customer provided resistors)
  - 12 bit conversion; offset binary or 2's complement code
  - 33 microsecond conversion time including channel switching (gain = 1)
  - ±0.05% full scale accuracy (0–10 V scale)
  - Short cycle auto gain ranging for higher resolution
  - Start conversion on software command or on external trigger
  - ±100 V input voltage protection signal to ground
  - ±32 V input voltage protection, channel to channel
  - 80 dB (dc-1 kHz) common mode rejection (0–10 V scale)
  - 80 dB channel to channel cross talk rejection (0–10 V scale)
- Parallel or Serial Communications with Host for Remote Operation:
  - Operate at distance of up to 12' from I/O Channel master via 50-conductor ribbon cable
  - Operate at distance of up to 3900' from host serial port via dual twisted par wiring (full duplex)
- Serial Port Circuitry Supports RS-232C, RS-422 and RS-485 Electrical Requirements
- Provides Four Jumper Selectable Serial Communications Port Baud Rates: 300, 1200, 9600, and 19,200 Baud
- Selectable Base Address and Selectable Serial Address (Communication Port I.D.) for Application of Multiple Boards on the I/O Channel or Multidrop Serial I/O Network, Respectively
- Controlled by an On-Board Intelligent Peripheral Controller (IPC) (MC68B09 MPU)
- Self-Test Capability and Board Fault Indicator Utilizing IPC Firmware and On-Board Voltage Reference
- 5" × 16-1/4" Board Form Factor for NEMA and 19" RETMA Cabinet Mounting
- 0°C–70°C Operating Temperature Range
- Optional (User Installed) dc/dc Converter for ±15 Vdc Power from Single +5 Voltage
- 2K Bytes of On-Board Static RAM
- 256 Bytes of the Static RAM are Shared by a Host Microprocessor on the Motorola I/O Channel and the MC6809 Microprocessor on the RAD1
- A Command Channel Implemented in the Shared RAM Facilitates the Bidirectional, Parallel Exchange of Instructions and Data Between Host and RAD1 Microprocessors
- RAD1 Firmware in ROM1 Supports the Upload/Download of Programs and Data Permitting a Program or Routine to be Downloaded into On-Board RAM to Maintain Real-Time Control without Burdening the Host Processor
- Upload/Download Capability Facilitates Fine Tuning During Control System Installation or Modification
- Additional 24-Pin Socket for User-Supplied 4K × 8 EPROM/ROM Device (ROM2)
- 17 Subroutines and Six Command Functions in Motorola Supplied EPROM (ROM1) Available to:
  - Host application driver (at command channel interface)
  - Downloaded control program in RAD1 static RAM (on RAD1 bus)
  - User-written control program in ROM2 socket device (on RAD1 bus)

Remote Intelligent Analog to Digital Conversion Module

The general function of the Remote Intelligent Analog to Digital Conversion Module (RAD1) is to permit a host-resident application program in a general purpose processor to receive information derived from sample measurements of voltage or current originating in one or more external devices. Need for such measurements is often found in process control and automatic test equipment applications.

Typically a remote A/D conversion device such as the RAD1 is controlled by an application driver running under the host operating system. Communications between driver and conversion device usually take place over a serial link.

Although it too offers a serial port, the RAD1 incorporates innovations that make possible parallel communications over the Motorola I/O Channel (12 address lines, 8 data lines). These innovations include (1) 2K bytes of on-board static RAM in which 256 bytes are shared between the I/O channel master and the RAD1 MC68B09 microprocessor on the RAD1 bus and (2) a command channel implemented in the 256 bytes of shared memory and supported by firmware in an on-board 4K byte EPROM.

Moreover, the firmware supports upload via the command channel to host memory and download into the RAD1 on-board RAM. Since a program resident in RAM can access A/D conversion control routines resident in EPROM a program can be downloaded for supervisory control or to facilitate fine tuning during control system installation or modification.

Functional Description

The RAD1 is an intelligent 12-bit, 16 or 32-channel analog-to-digital conversion module. The module is independently controlled by an on-board MC68B09 microprocessor. Figure 1 provides a functional block diagram of the module.

The RAD1 provides the user with a choice of parallel or serial communication to the monoboard microcomputer for command and data transfers as shown in Figure 2. Parallel transfers are accomplished via the I/O channel extension feature of any VERSAmodule or VMEmodule Monoboard Microcomputer. Additional modules and other types of I/Omodules may be connected to the I/O channel at distances up to 12 feet maximum from the monoboard microcomputer.

Serial transfers are accomplished via the serial port, which supports RS-232C protocol for short distances (to 50 feet) or RS-485/RS-422 protocol for greater distances (to 3900 feet) at up to 9600 bps.

An on-board, removable terminal strip facilitates connection of field wiring to the 32 single-ended or 16 differential A/D channels. On-board signal conditioning provides ±100 V signal-to-ground and ±32 V channel-to-channel protection. A wide range of voltage and current input ranges can be accepted, providing fast conversions to 12-bit accuracy. Microprocessor-controlled, short cycle auto-ranging provides increased low level signal resolution. Conversion may be started immediately on internal channel command or on occurrence of an externally-supplied trigger signal.

FIGURE 1 — M68RAD1 Functional Block Diagram

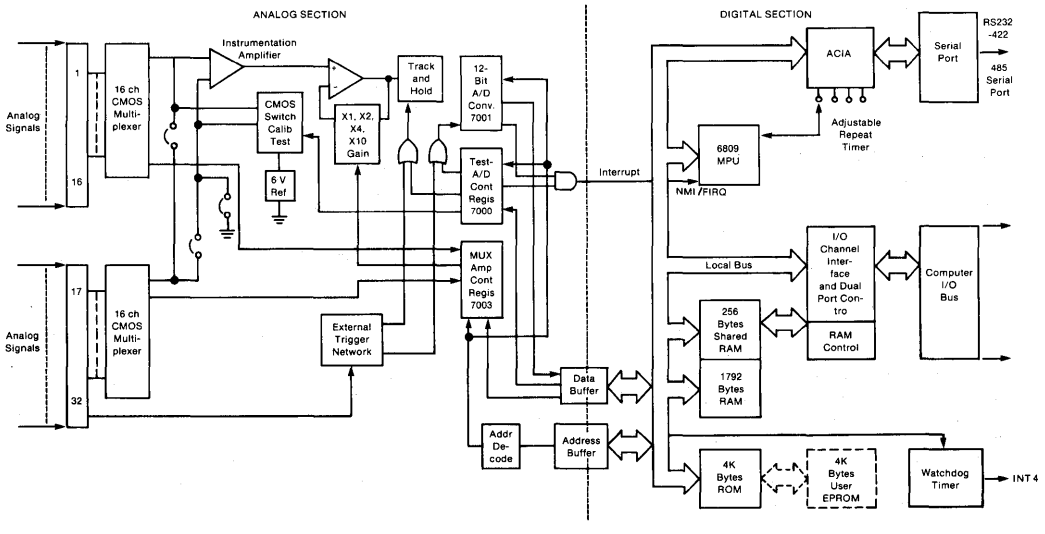
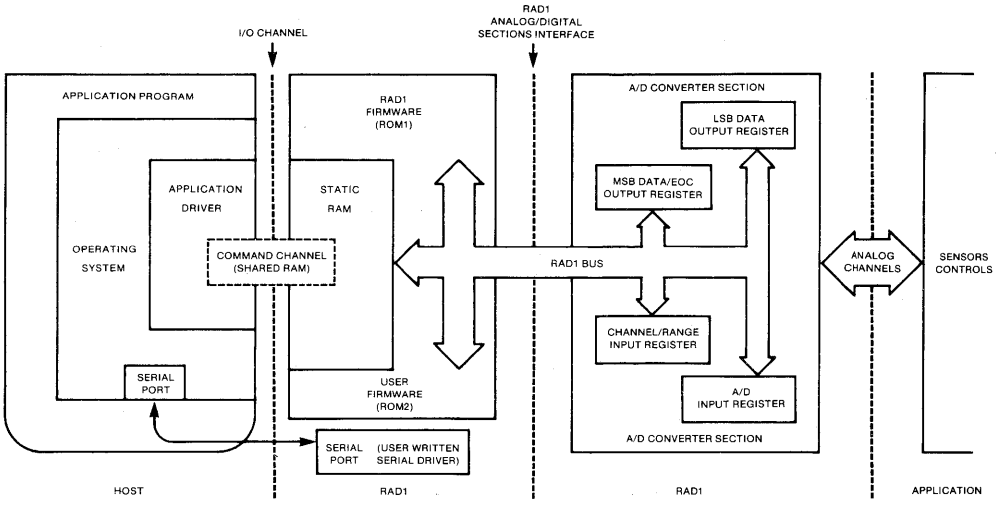


FIGURE 2 — M68RAD1 Digital and Analog Communications



The board provides self-test and board fault isolation, using MC68B09 firmware and an on-board voltage reference. Also, offset and gain error correction, on any amplifier gain setting, can be accomplished in software by using the program selectable 6.000 Vdc calibration and 0.000 Vdc offset switches.

Components are mounted on a slimline 5 x 16-1/4 inch board for NEMA or 19″ RETMA mounting.

Base Address Selection

The RAD1 module resides at one of sixteen base addresses on 4K-byte boundaries in the I/O Channel memory map. A header for jumper selection in conjunction with an on-board digital comparator provides the address decoding that allows up to 16 RAD1 modules to be used in a single system.

Interrupt Priority Selection

To accommodate use of RAD1 modules in multiple priority level applications, a header is provided that permits connection, by jumper, to the four prioritized I/O channel lines. The module watchdog fail interrupt may only be connected to the I/O channel interrupt level four line. The module set point interrupt may be connected to any one of the level one, level two or level three lines.

Serial Interface Address

For applications requiring serial communications between multiple modules as in a multidrop interconnected system, the RAD1 module provides a header permitting jumper selection of a serial address (serial port I.D.). Any of 16 serial addresses can be selected allowing use of up to 16 RAD1 modules in a single system.

Repeat Interrupt Timer

For applications requiring a repeat interrupt timer, the RAD1 module provides selectable interrupt pulses from the band rate clock with periods from 3.2 μs to 0.377. These pulses interrupt the FIRQ line to the MC68B09 processor.

Supervision of Remote Scanning and Conversion Activities

RAD1 analog-to-digital conversion is controlled using an on-board intelligent peripheral controller (IPC). The IPC consists of a 2 MHz MC6809 processor, 2K bytes of static RAM, an Asynchronous Communications Interface Adapter (ACIA) — implemented serial port and firmware installed in two 4K ROM sockets. Within the static RAM, 256 bytes operate as shared RAM between the I/O channel and MC6809 providing an interface between the host driver and the RAD1 through which commands and data

can easily be transferred. For convenience, this interface is called a command channel. See Figure 3.

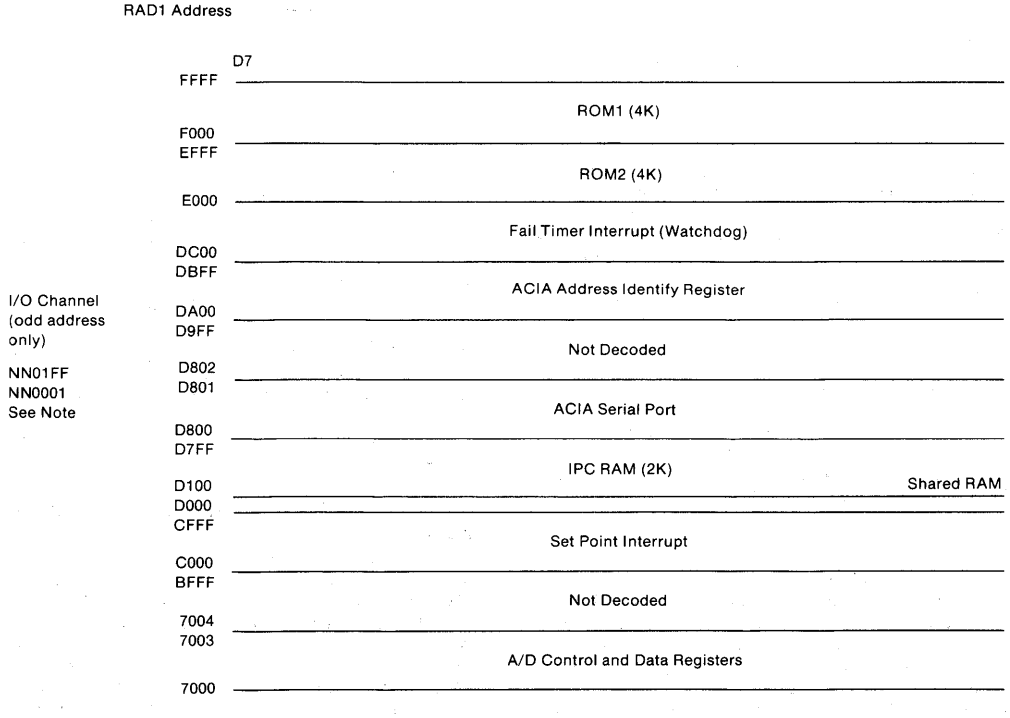
One of the 4K ROM sockets is available for a device containing user-written application routines (ROM2). In the other socket, ROM1 (supplied with the RAD1) contains a number of firmware routines which may be accessed by the driver to perform needed tasks. Additional routines or control software can be downloaded to the RAD1 2K static RAM to enlarge the remote operation of the module. These features greatly simplify the generation of the application program required to control the operation of and read data from the analog-to-digital converter.

Command Channel Used to Invoke ROM1 Service Routines

As supplied, the RAD1 can execute four ROM1-resident service routines which together enable an application program to supervise remote scanning and conversion activities. Such activities are controlled through execution, by the RAD1 MC6809 microprocessor, of instructions resident in RAD1 memory. The function offered by a service routine is obtained by transferring the appropriate value, in the command byte field of the command channel to the RAD1. Table 1 lists each field in the command channel and briefly describes the corresponding functions.

The command channel is also utilized for transfer of other support of the remote activities. An additional service routine provides remote self test of RAD1 functions (exclusive of power-on self tests). Table 2 briefly describes the function provided by remote execution of each ROM1 service routine.

FIGURE 3 — Map of M68RAD1 Local Memory Map Showing Command Channel



NOTE: Digits NN firmware dependent, as shown, all header J24 jumpers in.

TABLE 1 — Command Channel Interface Fields

Offset	Field	Function	Sender
$00	Stop	Requests termination of a continuously running function.	Application Driver
$01	Command	Requests execution of a command service routine.	Application Driver
$02	Interrupt Flag	Value identifies by type the last RAD1-generated interrupt.	RAD1 Firmware
$03	Command Status	Indicates completion of function or command execution.	RAD1 Firmware
$04	Interrupt Mask	Value controls type and timing of RAD1-generated interrupts.	Application Driver
$05	Start of FIFO	Offset from base of shared RAM to beginning of FIFO area.	Driver/RAD1
$06	End of FIFO	Offset from base of shared RAM to end of FIFO area.	Driver/RAD1
$07	Next "To Get" Byte	Offset for getting first of 3 bytes of conversion data from FIFO.	Driver
$08	Next "To Put" Byte	Offset for putting first of 3 bytes of conversion data in FIFO.	Driver
$09	Parameter Pointer	Points to first parameter in shared RAM. Used by RAD1 or host to retrieve passed parameters.	Driver/RAD1
$0A	Reserved		
$2A	User Defined	FIFO	Driver/RAD1
$FB	Self Test Status	Indicates RAD1 self test results prior (only) to execution of first command.	RAD1 Firmware
FC[1]	Data	High Speed Conversion Data (2 Bytes).	High Speed Driver/ RAD1
FC[1]	Co	Flag polled by user/RAD1 to control transfer of high speed conversion data.	High Speed Driver/ RAD1
$FF	Hardware Status	Serves as register to identify the interrupting RAD1 module to the application driver.	RAD1 Hardware

NOTE: For high speed (33 μs) scan, the FIFO is not used. An area for data transfer is defined as required.

TABLE 2 — ROM1 Service Routines

Command Field Byte Value	Command	Service Routine Function
01	LOAD	Download function/program into RAD1, RAM, under control of application driver.
02	DUMP	Upload RAM/ROM into host memory, under control of RAD1 firmware.
03	JUMP	Transfer control to function/program downloaded into RAD1 RAM.
10 nn nnnn or 11 nn nnnn	EXECUTE	Start execution of specified ROM1 or ROM2 — resident function (n = function number, see Table 3.
04	SELF-TEST	Initiate self-test of following resources and functions: RAM — Completely tests all RAD1 RAM. ROM2 — If ROM2 present, verify ID header and checksum. Analog Circuits — Test gain and range at 6 V and 0 V references. Tests stability of sample and hold circuit. ACIA — If a terminal is connected to the RAD1 serial port, output a test message. Error messages, if any, are sent to the command channel for pickup by the application driver.

EXECUTE Command-Invoked ROM1/ROM2 Functions

Six commonly needed scanning and conversion functions are supplied in ROM1 with the RAD1. An application driver can invoke any of these by placing, for transfer to the RAD1, the appropriate EXECUTE command value in the command byte of the command channel. For example, a value of 86_{16} (10000110_2) obtains high speed single channel scan. Table 3 describes the six functions supplied in ROM1.

The user can add those functions required by the application by installing them in the ROM2 device. On invocations, the value passed in the command byte of the command channel indicates both the user function number and the use of a ROM2 table. Up to 64 user functions in ROM2 are supported.

Execution of a function can be allowed to terminate independently or on driver command passed through the command channel.

TABLE 3 — EXECUTE Command-Invoked ROM1 Functions

Function No.	Description
1	Single/Multiple Channel — External trigger converts a list of channels to scan starting when the external event occurs. The list will be scanned again following the next external event.
2	Multiple Channels, with a set-point value specified, are scanned until a set-point is hit. After that, all specified channels will be converted and data is put in shared RAM.
3	This function supports three modes of operation: • Repeated single or multiple channel conversions performed on a list of channels with a specified time between successive starts of the list. • Performs a single conversion on each channel requested. • Performs repeated conversions on all requested channels with approximately 100 μs between conversions.
4	Auto Range Single/Multiple Channels adjusts the gain select for the specified channels and then places the conversion data and gain range in shared RAM.
5	This calibrate function returns the offset and gain corrections to the host.
6	High Speed Single Channel Scan provides conversions repeated at 33 μs intervals of a single specified channel.

Subroutine Functions Supplied with RAD1

The subroutines listed in Table 4 and Table 5 which are utilized by RAD1 firmware to perform control functions, are available (1) to user-written firmware residing in the ROM/EPROM device installed in the second 24-pin socket (ROM2) or (2) to a user task downloaded into the RAD1 2K × 8 static RAM. Subroutines are accessed using a jump table that resides at the top of ROM1.

TABLE 4 — ROM1 Subroutines

Subroutine Name(s)	Jump Table Address	Description
QUEDT QUEDAT	$F011 $F00E	Provides, in shared memory, a FIFO interface to a host for conversion data.
INITF	$F04D	Initializes FIFO to a specified length and starting address.
DINITF	$F04A	Initializes FIFO to the default length (max. size) and starting address.
FIRQ	—	Services timer interrupts and polls the stop byte of the command channel.
TRMCOM	$F005	Performs command/function completion of housekeeping. Updates command status byte.
GETPRM	$F002	Moves RAD1 input parameters out of shared RAM.
EXTTRG	$F047	Sets up a channel for conversion on external trigger.
SCANLT	$F008	Scans entire channel list once, performs conversions and queues data in shared RAM.
AUTO	$F00B	For a specified channel, finds the correct range and performs a conversion.
CALADR	$F014	Calculates next "to put" offset (beginning address of data from conversion) in shared RAM.
MAPCHL	$F017	Translates user representation of channel number and gain range to the hardware code for channel number gain range and multiplexer selection.
CONURT	$F038	From previously started conversion, gets data and stores it in temporary memory (rather than queueing the data in shared RAM). Starts the next conversion.
CONVOV	$F03B	From previously started conversion, gets data and stores it in the X register. Starts the next conversion.
SMXCHL	$F03E	When auto trigger mode is enabled, starts conversion of the specified channel.
GETCON	$F041	On receipt of EOC signal, gets conversion data and stores it in the X register.
WATEOC	$F044	If a conversion has begun, regains control for calling routine on conversion completion.

TABLE 5 — ROM Enable/Disable Subroutines

Subroutine Name(s)	Jump Table Address	Description
ESTM6V	$F01A	Enable 6 V self test.
ESTMOV	$F01D	Enable 0 V self test.
ESHRTC	$F020	Enable short cycle.
EAUTO	$F023	Enable auto trigger mode.
ENAEXT	$F026	Enable external trigger mode.
DSHRTC	$F029	Disable short cycle.
DAUTO	$F02C	Disable auto trigger mode.
DISEXT	$F02F	Disable external trigger mode.
DSTM6V	$F032	Disable 6 V self test.
DSTMOV	$F035	Disable 0 V self test.

Example of Two Channel Continuous Scan Under Control of a Downloaded Program

The example program for which a source listing and explanation keyed to line numbers is shown makes the following assumptions:

1. The program has been downloaded into the 2K static RAM on board the RAD1.
2. The two channels to be continuously scanned have been made known to the ROM1 firmware by program equates HCHNL1 and HCHNL2.
3. The command channel FIFO is used for communication of acquired data to the application driver.

```
*
*       Scan 2 Channels Continuously
*
*
```

1		JSR	$F04A	Initialize Command Channel FIFO
2	LOOP	LDB	#HCHNL1	Prepare for first channel
3		JSR	$F03E	Call ROM1 start conversion subroutine
4		LDB	#HCHNL2	Prepare for second channel
5		JSR	$F03B	Call ROM1 to get conversion data in X and start conversion subroutine
6		TFR	X,Y	Get ready to transfer data
7		LDA	#CHNL1	To application driver
8		JSR	$F011	Queue data in FIFO
9		JSR	$F041	Call get conversion data on EOC subroutine
10		TFR	X,Y	Get ready to transfer data
11		LDA	#CHNL2	To application driver
12		JSR	$F011	Queue data in FIFO
13		STA	Watchdog	Strobe watchdog timer
14		LDA	SHRSTP	Has application driver stopped me yet?
15		BEQ	LOOP	No! Continue
16		LDB	#GOOD	Status for application driver
17		LDA	#01	Interrupt mask
18		BRA	$F005	Call function completion subroutine

Line No.	Comment
1	Subroutine DINITF is called to initialize the FIFO to the default values of length and starting address.
2	HCHNL1 is a value the user has equated to the 8-bit code required by the hardware to identify the channel, gain range and one of the two input multiplexers. For a similar conversion function requested by an application driver via the command channel, a different hardware code is required and is calculated by the ROM1 firmware.
3	Subroutine SMXCHL is called to start conversion of the first channel.
4	HCHNL2 is the 8-bit hardware code for the second channel.
5	This routine, CONVOV, gets data in the X register from a previously started conversion and starts the next conversion.
6	Data to be queued must be in the Y register.
7	The application driver also requires the channel number.
8	Subroutine QUEDT places the channel number and conversion data in the FIFO for pickup by the application driver.
9	Subroutine GETCON stores conversion results in the X register on receipt of an EOC signal.
10	Same as 6.
11	Same as 7 but with number of second channel.
12	Same as 8.
13	Strobing the board watchdog timer with any value keeps the RAD1 FAIL LED off and prevents an interrupt from being sent to the host.
14	Loads the A register with the value from the Stop field of the command channel. A driver uses this byte to obtain termination of a continuously running function.
15	Branch to LOOP if the A register did not contain the stop value.
16	Preparing to terminate a function requires that status be sent to the application dirver. Good status = 00.
17	The application driver sets bits in this byte to inform the RAD1 when to generate an interrupt. The terminate command subroutine; TRMCOM, at $F005 checks the mask byte to determine if an interrupt should be generated.
18	Exit this function. Go back to the command mode.

Interfacing Analog Signals to the RAD1

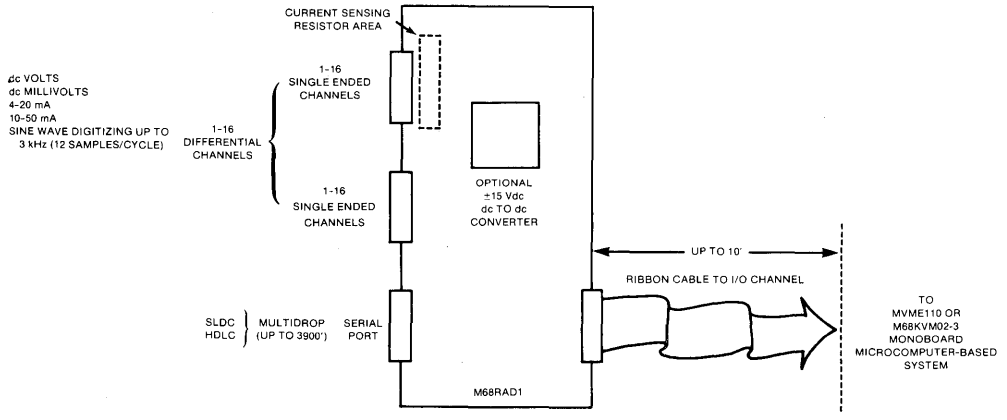
The RAD1 is designed to be used in many common applications in commerce and industry requiring analog to digital conversion for measurement and control. Typical uses include the measurement of voltages, temperature, pressure, strain and force. Almost any process in which a sensor is used to transform a change in a physical quantity into current or voltage is suitable for RAD1 application. Some idea of the wide applicability of the modules is provided in Figures 4, 5, 6 and 7.

M68RAD1-1

The RAD1 can be used to measure voltage signals such as dc volts to ±10 Vdc millivolts to 1 MV resolution, 4-20 or 10-50 mA remote sensors, and sine wave sampling up to 3 kHz.

The RAD1 can be operated through the I/O Channel cable up to 12' from the CPU board, or through the serial port using a standard EIA RS-232C serial communications link. The serial port can be jumpered to operate through RS-422 or -485 SDLC or HDLC Multidrop communications network up to 3900' from the host serial port.

FIGURE 4 — Typical Data Acquisition Configuration



The RAD1 can be configured for single ended, common ground inputs or differential, floating ground inputs. The differential input allows the user to measure inbalanced, to ground, voltage inputs and current inputs, using user supplied precision resistors. Voltage and current can simultaneously be measured using the differential input mode. Sensor measurements of 4-20 mA or 10-50 mA are accomplished by connecting a precision 500 ohm, 0.1% tolerance resistor for 4-20 mA in the current sensing resistor area. This will yeild a 1-5 Vdc signal.

FIGURE 5 — Typical Voltage and Current Sensor Differential Mode Connections

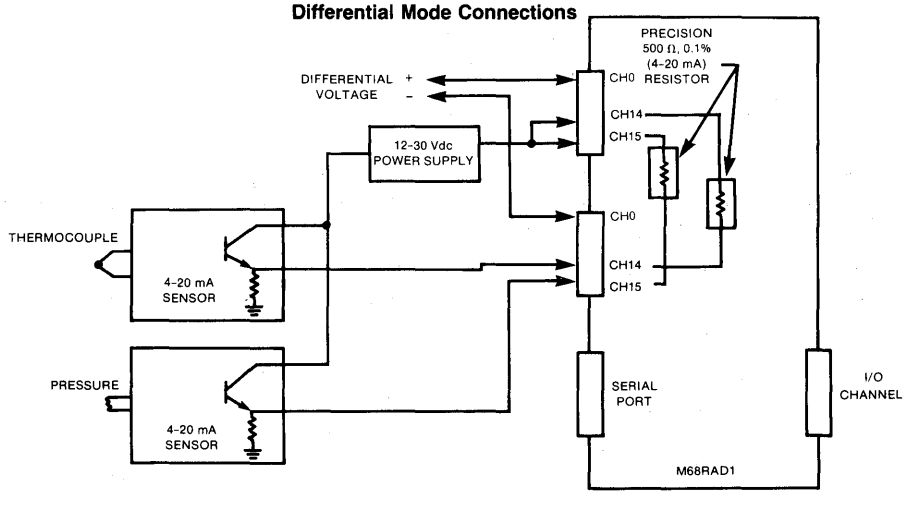
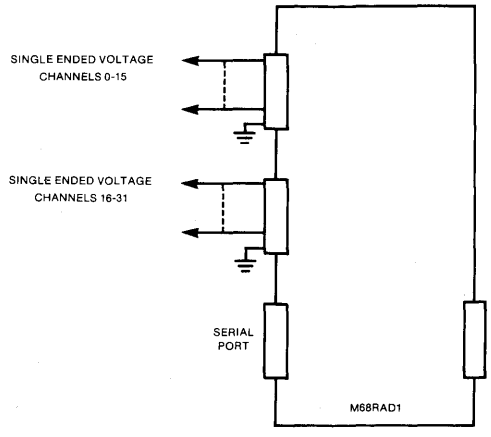
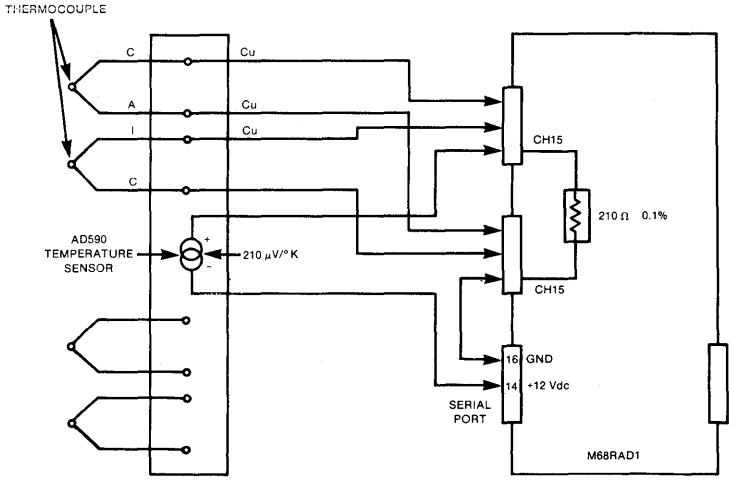


FIGURE 6 — Single Ended Mode (32 Channels)



Thermocouples can be connected to the RAD1 through a user supplied isothermal block. The isothermal blocks temperature effect is compensated by the user adding a AD590 solid state temperature sensor across a precision 210 ohm resistor in the current resistor area. Voltage for the AD590 is obtained from the serial port +12 Vdc terminal. The isothermal block temperature offset and thermocouple nonlinearity is corrected in software.

FIGURE 7 — Thermocouple Isothermal Block Temperature Compensation Using A Solid State Temperature Sensor



Software Driver

A driver for the RAD1 is incorporated in the VERSAdos M68000 Real-Time Operating System. It provides a host-resident application program with a ready means of communicating with the module and of obtaining the necessary and optional control and sensing service functions required for the external equipment to which the module is connected. Driver documentation is provided with the VERSAdos System. A manual, Guide to Writing A Device Driver, M68DRVGD/D1, detailing how to write a device driver that runs under the M68000 Real-Time Multitasking kernel (RMS68K) or under VERSAdos is also available.

Compatible I/O Channel Controllers

The RAD1 module will operate with any of the following in control of the I/O Channel.

MVME110	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer
VME/10	Microcomputer System

MVME400 Series

The 400 Series is a family of I/Omodule peripheral controller cards designed for modular, low-cost applications. These modules are mechanically compatible with the single Eurocard form factor, and operate from the Motorola I/O Channel. Use of the I/O Channel allows the user to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

I/O Channel

The I/O Channel is specifically designed to provide efficient low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 megabytes/sec. For the modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support Motorola's modular product families: VMEmodule and VERSAmodule.

Specifications

Table 6 provides the general RAD1 specifications, Table 7 provides the analog specifications and Table 8 provides the digital specifications.

TABLE 6 — RAD1 Specifications — General

Characteristics	Specifications
Power Requirements	
RAD1 (1)	+5 Vdc ±5% at 2.7 A max. with dc/dc converter (1.7 A without converter) ±12 Vdc at 25 mA max. +15 Vdc at 95 mA max. (2) -15 Vdc at 80 mA max. (2)
Environmental Requirements	
Operating Temperature	0 to 70°C
Storage Temperature	-25 to +125°C
Humidity	Operable from 5% to 95% RH non-condensing
Mechanical Specifications	
Board Size	5.0 × 16.25 inches with eight 0.75-inch long × 0.25-inch diameter standoffs for no. 6-32 screw mounting.
Connectors	
J1/J2/J3	18-pin terminal strip — Eléctrovert 25.600.1853
J4	12-pin connector with 0.156-inch pin centers — Molex No. 09-74-1121
J5	64-pin connector (2 × 32) — DIN41612C Series

NOTES:

(1) Since the I/O Channel provides power to the peripheral control modules, the user should be warned of powering more than two RAD1 boards using this scheme. The I/O Channel supports 16 various boards; however, it should not carry more than 4.5 A. The other boards may be powered via power connector J4.

(2) Optional on-board power converter may supply ±15 Vdc.

TABLE 7 — RAD1 Specifications — Analog Section

Characteristics (1)	Specifications	
Inputs		
Number of voltage channels	32 single-ended or 16 differential	
Ranges	**Bipolar**	**Unipolar**
	-10 to +10 Vdc	0 to +10 Vdc
	-5 to +5 Vdc	0 to +5 Vdc
	-2.5 to +2.5 Vdc	0 to +2.5 Vdc
	-1 to +1 Vdc	0 to +1 Vdc
Current Input Channels	4-20 mA; 10-50 mA (using customer-selected resistors)	
Number of Current Channels	16	
Impedance	>10 megohms (10 pF "off" channel or 100 pF "on" channel)	
Maximum Input Voltage	±25 Vdc maximum continuous ±100 Vdc (less than 1 ms)	
Gain (Software Programmable)	X1, X2, X4, or X10	
Bias Current	2 nA max.	
Offset Current	100 pA max.	
Accuracy		
System	±0.05% (+10 Vdc range)	
Drift (Gain = 1)	±50 ppm of full scale range/°C	
Linearity	±1/2 LSB	
Quantizing Error	±1/2 LSB	
Gain Error	Adjusted to zero	
Offset Error	Adjusted to zero	
Transfer Characteristics		
Resolution	12 bits	
Conversion Time (Gain = 1)	33 microseconds	
S/H Acquisition Time	10 microseconds	
Channel Cross-Talk	>80 dB	
Instrumentation Amplifier CMRR	>80 dB (1 kHz, Gain = 1)	
Programmable Short Cycle	4 bits	
A/D Output Coding		
Unipolar	Complementary binary	
Bipolar	Complementary two's complement	

NOTE: (1) Typical characteristics at 25°C.

TABLE 8 — RAD1 Specifications — Digital Section

Characteristics	Specifications
Local Bus	
Microprocessor	MC68B09
Crystal Frequency	8 MHz
Processor Speed	2 MHz
Memory Size	
ROM	Two 24-pin sockets for 4K × eight 5 Vdc ROM devices (250 ns access)
RAM	One 2K × 8 static RAM (2016P-1) (100 ns access)
Addressing	Fixed on-board decoding
Serial Port	One ACIA (68B50) with selectable RS-232C, RS-422, or RS-485 interface

TABLE 8 — RAD1 Specifications — Digital Section (continued)

Characteristic	Specification
Baud Rate	Selectable for 19.2K, 9600, 1200 or 300 baud
Interrupts	(1) IRQ from ACIA (2) NMI/FIRQ from A/D converter to MC68B09 (user selectable) (3) Optional timer interrupt from baud rate generator to FIRQ of MC68B09
I/O Channel	
Address	12 bits — 4 MSB (jumper selectable) to define board address
Data	8 bits — bidirectional
Control	STROBE/ACKNOWLEDGE for asynchronous data transfer to dual-ported RAM. WT = write line for read/write operations to RAM. IORESET = external RAD1 reset capability.
Interrupts	Level 4 (board fail-watchdog) interrupt from RAD1 (jumper enabled) — Level 1, 2, 3 interrupt (software enabled — jumper-selectable).

Input/Output Connectors

Table 9 provides the analog input connections (J1 and J2), and Table 10 provides the serial I/O connections (J3). These three connectors are 18-pin terminal strips that may be unplugged from the board without disturbing the field wiring. Table 11 provides the optional dc power connections (J4).

The I/O Channel connections and interface requirements are fully described in Motorola Specification M68RIOCS.

TABLE 9 — Analog Input Connections (J1, J2)

Single-Ended	Differential	Pin Number
CH0 CH16	+CH0 -CH0	J1-1 J2-1
CH1 CH17	+CH1 -CH1	J1-2 J2-2
CH2 CH18	+CH2 -CH2	J1-3 J2-3
CH3 CH19	+CH3 -CH3	J1-4 J2-4
CH4 CH20	+CH4 -CH4	J1-5 J2-5
CH5 CH21	+CH5 -CH5	J1-6 J2-6
CH6 CH22	+CH6 -CH6	J1-7 J2-7
CH7 CH23	+CH7 -CH7	J1-8 J2-8
CH8 CH24	+CH8 -CH8	J1-9 J2-9

TABLE 9 — Analog Input Connections (J1, J2) (continued)

Single-Ended	Differential	Pin Number
CH9 CH25	+CH9 -CH9	J1-10 J2-10
CH10 CH26	+CH10 -CH10	J1-11 J2-11
CH11 CH27	+CH11 -CH11	J1-12 J2-12
CH12 CH28	+CH12 -CH12	J1-13 J2-13
CH13 CH29	+CH13 -CH13	J1-14 J2-14
CH14 CH30	+CH14 -CH14	J1-15 J2-15
CH15 CH31	+CH15 -CH15	J1-16 J2-16
SIGNAL GROUND	SIGNAL GROUND	J1-17 J1-18 J2-18
EXT. TRIG INPUT		J2-17

NOTE: All unused inputs should be grounded.

TABLE 10 — Serial Connector (J3) — Pin Descriptions

Pin No.	Signal Mnemonic	Description
J3-1	RxD	RECEIVE DATA (serial data output from RAD1)
J3-2	GND	GROUND
J3-3	TxD	TRANSMIT DATA (serial data input to RAD1)
J3-4	RD+	Receive Data (+)
J3-5	RD-	Receive Data (-)
J3-6	SD+	Send Data (+)
J3-7	SD-	Send Data (-)
J3-8	RTS	REQUEST TO SEND
J3-9	CTS	CLEAR TO SEND
J3-10	DSR	DATA SET READY
J3-11	DTR	DATA TERMINAL READY
J3-12	DCD	DATA CARRIER DETECT
J3-13	+5V	+5 Vdc Power
J3-14	+12V	+12 Vdc Power
J3-15	-12V	-12 Vdc Power
J3-16	GND	GROUND
J3-17	GND	GROUND
J3-18	GND	GROUND

TABLE 11 — DC POWER CONNECTOR (J4) — Pin Description

Pin No.	Signal Mnemonic	Description
J4-1	GND	GROUND
J4-2	GND	GROUND
J4-3	+5V	+5 Vdc Power
J4-4	-5V	-5 Vdc Power
J4-5	NC	NO CONNECTION
J4-6	GND	GROUND
J4-7	+12V	+12 Vdc Power
J4-8	-12V	-12 Vdc Power
J4-9	NC	NO CONNECTION
J4-10	GND	GROUND
J4-11	+15V	+15 Vdc Power
J4-12	-15V	-15 Vdc Power

Ordering Information

Part Number	Description
M68RAD1	Remote Intelligent Analog-to-Digital Conversion Module. Includes 32 single-ended or 16 differential A/D channels, choice of parallel or serial I/O operation, self-test capability, and board fault indicator LED. Includes User's Manual.
M68RAD1/D1	RAD1 User's Manual including complete hardware and software descriptions.

Related Documentation

M68RIOCS/D2	Input/Output Channel Specification Manual
M68DRVGD/D1	Guide to Writing a Device Driver

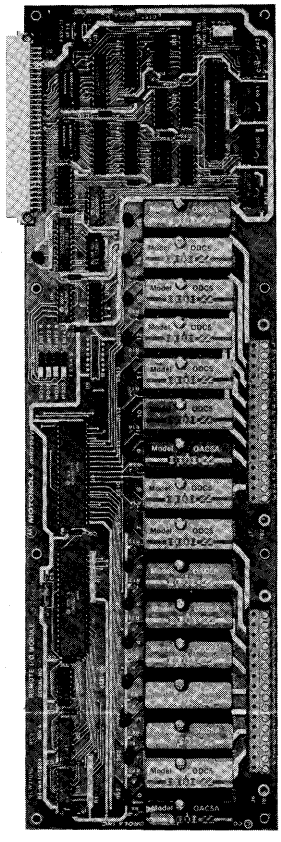
Other Modules in the MVME400 Series

MVME400	Dual Channel RS-232C Serial Port I/Omodule
MVME410	Dual Channel 16-Bit Parallel Port
MVME420	SASI Peripheral Adapter I/Omodule
MVME435	Buffered 9-Track Magnetic Tape Adapter I/Omodule

Remote Input/Output Module

- Removable Mass Termination Strip (M68RIO1-1) Fixed Strip (M68RIO1-2)
- Quick Replacement of Remote Input/Output Module (M68RIO1-1) without Disturbing Load Wiring
- Accepts up to 16 Solid State Plug-In Input/Output Modules, any Mix of Crydom Series 6 Opto 22, Gordos or other Compatible Modules
- Quick Replacement of Plug-In Module without Disturbing Load Wiring
- Individual LEDs Signal Operation of each Plug-In Input/Output Module
- LED Fault Indicator for Diagnostic Testing of Module
- 5" x 16-1/4" Board Form Factor for NEMA or 19" RETMA Mounting
- Individual 5 A Fuse for each Plug-In Input/Output Module
- Motorola I/O Channel Compatible
- Selectable Base Address — any 16 Byte Boundary in the 4K Byte I/O Channel Address Range
- Up to 16 M68RIO1 Modules can be Connected to One Port for Control of up to 256 Plug-In Modules at Distances of up to 12' from the I/O Channel Master
- An I/O Channel Interrupt can be Generated by up to Eight Plug-In Modules
- 0° C-70° C Operating Temperature Range

The general function of the M68RIOM Remote Input/Output Module (RIOM) is to permit a host-resident application program to cause actuation of remote switches to external equipment and to receive notification of remote switch actuation caused by external equipment. In the industrial environment the role of such an interface between digital logic and the control/sense function is well performed by optically isolated input/output modules. The RIOM provides a means of utilizing the many available plug-in, encapsulated input/output modules to accommodate a wide range of power control and sensing applications.



FUNCTIONAL DESCRIPTION

The RIOM facilitates the parallel communications between a host microprocessor on the Motorola I/O Channel and up to 16 user-supplied input/output modules connected to external equipment. Figure 1 is a functional block diagram showing this link. RIOM specifications are provided in Table 1.

FIGURE 1 — Remote I/O Module Block Diagram

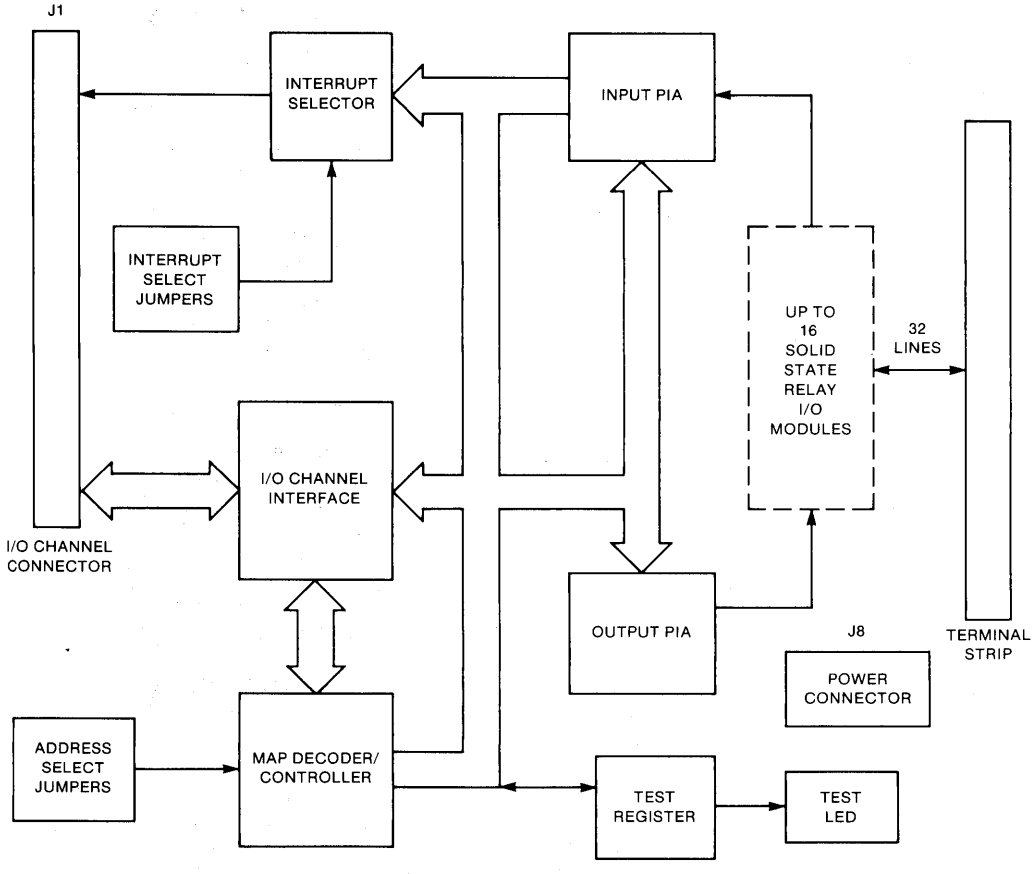


TABLE 1 — RIO1 Specifications — General

Characteristics	Specifications
Power Requirements	+5 Vdc, ±5% @ 0.750 A, typical (1.5 A max) All sockets populated with dc output modules supplied at 12-pin power header as shipped, or from 64-pin I/O Channel connector by user option.
Mating Connector for Power Header J8	Molex 09-06-0128 09-06-0127 or equivalent .156 inch (3.90 cm) insulation displacement connector (female)
Mating Connector for I/O Channel Connector J1	Winchester Electronics 96S-6053-05-31-1-2 or equivalent sockets connector (female)
Solid-State Relay Sockets	16 sockets for Crydom Series 6, Opto-22, Gordos or equivalent solid-state relays, with positive retention provided
Terminal Strips	
Two 16-Pin, Plug-In Terminal Strips (M68RIO1-1 Version)	Electrovert 25.600.1653 5 mm spacing No. 14 AWG., maximum
Two 16-Pin, Fixed Terminal Strips (M68RIO1-2 Version)	Electrovert 25.100.1653 5 mm spacing No. 14 AWG , maximum
Temperature	
Operating	0° to +70° C
Storage	-40° to 80° C
Operating Humidity	0 to 90% (non-condensing)
Dimensions	
Length	16.25 inches (40.63 cm)
Width	5.00 inches (12.50 cm)
Height	2.00 inches (5.00 cm) (includes .750-inch (1.88 cm) standoffs.) Relay height not included.

FUNCTIONAL DESCRIPTION (continued)

The RIOM is implemented on a 5″ × 16-1/4″ printed circuit board to accommodate NEMA or 19″ RETMA mounting. Mass termination strips are provided for connecting field wiring to the RIOM. Up to #14 AWG wire can be used. For applications in which quick module replacement is required, M68RIOM1-1 offers removable termination strips so that a module can be replaced without disconnecting the field wiring.

Up to 16 encapsulated, plug-in, optically coupled input/output modules can be mounted on one RIOM. Any mix of modules similar to those Opto 22, Gordos and Crydom modules listed in Table 2 can be used. On the RIOM, each module position is protected with a 5 A fuse and is provided with an indicator LED which lights when the module in that position is actuated. A LED fault indicator is provided on the RIOM to assist in diagnostic testing.

TABLE 2 — Compatible Solid-State I/Omodules

Vendor	Part Number	Type	Input Current vs Voltage	Max. Load Voltage	Load Current
Crydom	6101	dc input	7 mA @ 32 Vdc		
	6201	ac input	6 mA @ 120 Vac		
	6202	ac input	6 mA @ 240 Vac		
	6301	dc output		60 Vdc	3.5 Adc @ 60 Vdc (1)
	6401	ac output		140 Vac	3.5 A rms @ 140 Vac (1)
	6402	ac output		280 Vac	3.5 A rms @ 280 Vac (1)
	6311	dc output (buffered)		60 Vdc	3.5 Adc @ 60 Vdc (2)
	6411	ac output (buffered)		140 Vac	3.5 A rms @ 140 Vac (2)
	6412	ac output (buffered)		280 Vac	3.5 A rms @ 280 Vac (2)
Opto-22	IDC5	dc input	32 mA @ 32 Vdc		
	IAC5	ac input	10 mA @ 140 Vac		
	IAC5-A	ac input	6.5 mA @ 280 Vac		
	ODC5	dc output		60 Vdc	3 Adc @ 60 Vdc (1)
	OAC5	ac output		140 Vac	3 A rms @ 140 Vac (1)
	OAC5-A	ac output		280 Vac	3 A rms @ 280 Vac (1)
Gordos	IDC5	dc input	32 mA @ 32 Vdc		
	IAC5	ac input	10 mA @ 140 Vac		
	IAC5-A	ac input	10 mA @ 280 Vac		
	ODC5	dc output		60 Vdc	3 Adc @ 60 Vdc (3)
	OAC5	ac output		140 Vac	3 A rms @ 140 Vac (3)
	OAC5-A	ac output		280 Vac	3 A rms @ 280 Vac (3)

NOTES:
1. Derate .040 Amp per degree centigrade from 45 degrees centigrade.
2. Derate .033 Amp per degree centigrade from 45 degrees centigrade.
3. Derate .033 Amp per degree centigrade from 20 degrees centigrade.

Check manufacturer's specifications for minimum load requirements and test conditions. Use of inverting output modules is not recommended.

Base Address Selection

For systems requiring multiple modules interfaced to the Motorola I/O Channel, the RIOM provides a header for jumper selection of a base address in the 4K byte I/O Channel memory space. Any address on the 16-byte boundaries throughout the lowest 256-byte space may be selected. A maximum of 16 RIOM boards can be connected to one port at distance of up to 12 feet from the I/O Channel master. Control of up to 256 plug-in input/output modules is thus possible.

Interrupt Selection

Systems requiring interrupts initiated from plug-in input/output modules on a RIOM are accommodated by four headers that permit connection by jumper selection (or wirewrap) to any of the four I/O Channel interrupt lines for a maximum of eight modules per RIOM. For a system in which more than one plug-in input/output module can cause an interrupt to be placed on the same I/O Channel interrupt line, the user-written application driver must provide additional means of determining the specific interrupting module.

Host/RIOM Communications

The RIOM is communicated with as though it is eight read/write registers for input/output module control and sensing and one test register. The eight input/output module registers are accessed at the lower eight adjacent addresses in the 16-byte block relative to the RIOM base address and the test register at the top address in the block. Two MC68B21 Peripheral Interface Adapters (PIA) are assigned the lower eight addresses and provide the means by which input to and output from the board is effected. Four addresses associated with the two PIAs (one input, one output) are written to during system initialization to set up the 16 PIA channels for input or output according to the type of the plug-in module corresponding to each channel. Thereafter, the host merely reads from (input) or writes to (output) the particular one of the remaining four input or output PIA addresses that corresponds to the modules of interest in order to obtain the desired remote sense/control activity. Figure 2 depicts the general communications scheme and Figures 3 and 4 show the I/O Channel base address selection and PIA and test register assignments within a 16-byte block for the RIOM.

FIGURE 2 — RIOM Communications

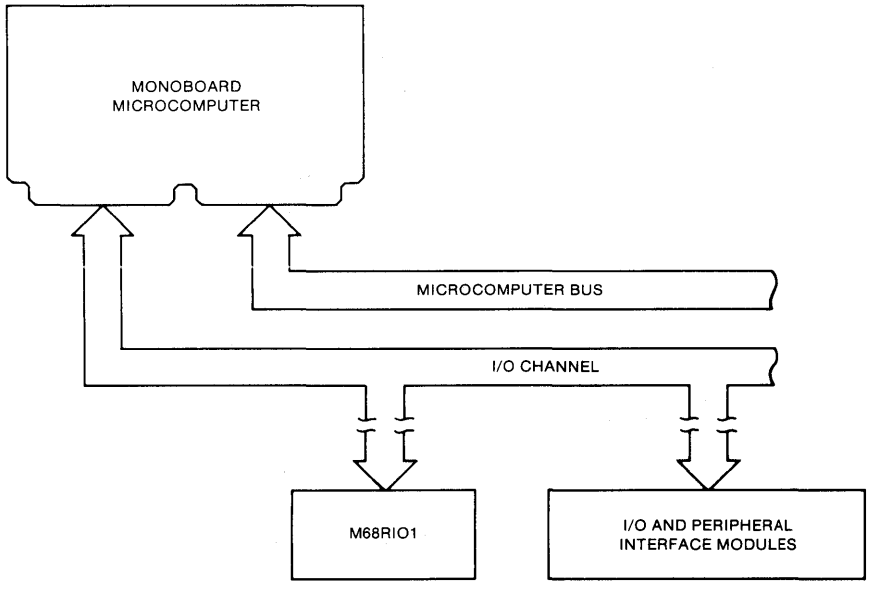


FIGURE 3 — RIOM I/O Channel Memory Base Address Selection

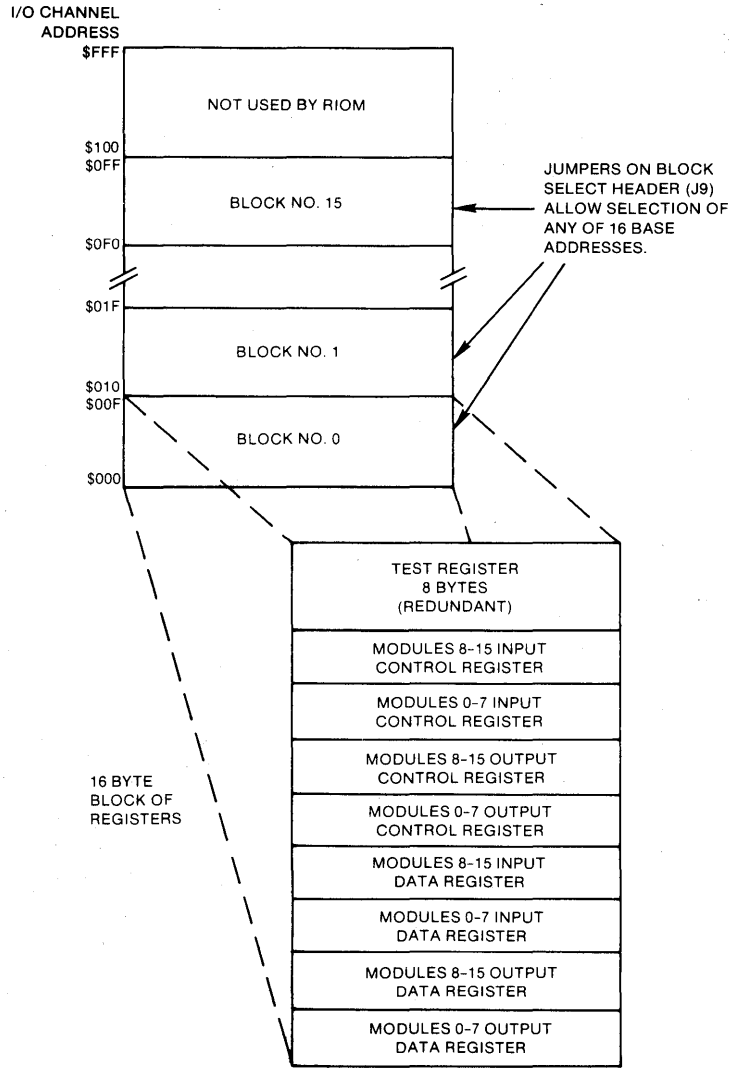


FIGURE 4 — RIOM PIA and Test Register Addresses

I/O CHANNEL ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
	TEST REGISTER (REDUNDANT)							
$0X8–$0XF	FAIL BIT 1 = FAIL	1	1	1	1	1	1	1
	MODULES 8–15 INPUT CONTROL REGISTER							
$0X7	#2 INT. FLAG	#3 INT. FLAG	INPUT PIA CB2 CONTROL			DATA DIR. ACCESS	INPUT PIA CB1 CONTROL	
	MODULES 0–7 INPUT CONTROL REGISTER							
$0X6	#0 INT. FLAG	#1 INT. FLAG	INPUT PIA CA2 CONTROL			DATA DIR. ACCESS	INPUT PIA CA1 CONTROL	
	MODULES 8–15 OUTPUT CONTROL REGISTER							
$0X5	#6 INT. FLAG	#7 INT. FLAG	OUTPUT PIA CB2 CONTROL			DATA DIR. ACCESS	OUTPUT PIA CB1 CONTROL	
	MODULES 0–7 OUTPUT CONTROL REGISTER							
$0X4	#4 INT. FLAG	#5 INT. FLAG	OUTPUT PIA CA2 CONTROL			DATA DIR. ACCESS	OUTPUT PIA CA1 CONTROL	
	MODULES 8–15 DATA INPUT/DIRECTION REGISTER							
$0X3	#15	#14	#13	#12	#11	#10	#9	#8
	MODULES 0–7 DATA INPUT/DIRECTION REGISTER							
$0X2	#7	#6	#5	#4	#3	#2	#1	#0
	MODULES 8–15 DATA OUTPUT/DIRECTION REGISTER							
$0X1	#15	#14	#13	#12	#11	#10	#9	#8
	MODULES 0–7 DATA OUTPUT/DIRECTION REGISTER							
$0X0	#7	#6	#5	#4	#3	#2	#1	#0

NOTES: 1. X depends on address select jumper configuration (J9).
2. Refer to data sheet MC68B21 for specific definitions.
3. Data register convention: 0 = module activated; 1 = module deactivated.

Supplying Power to the RIOM

The RIOM can be powered from the I/O Channel interface, connector J1, or from an external supply using the 12-pin power connector J8. Care should be taken, in a system in which the RIOM derives power at connector J1, to follow the current loading and power distribution recommendations specified in the Input/Output Channel Specification Manual, M68RIOCS. See Figure 5 for the recommended method of powering the RIOM. Table 3 provides pin assignments and signal descriptions for the Motorola standard power connector, J8. Compatible J8 mating connectors (female) are Molex part numbers 09-06-0128 (closed-end housing) and part number 09-06-0127 (daisy-chain housing).

TABLE 3 — Power Connector (J8) Pin/Signal Assignments

Pin Number	Signal Mnemonic
1	GND
2	GND
3	+5V
4	+5V
5	RESERVED
6	±12V GND
7	+12V
8	-12V
9	RESERVED
10	±15V GND
11	+15V
12	-15V

NOTE: Unused Pins 5-12 shown for reference only; not connected on the RIOM.

Software Driver

A driver for the M68RIOM1-1 and -2 Remote Input and Output Module is incorporated in the VERSAdos M68000 Real-Time Operating System. It provides a host-resident application program with a ready means of communicating with the RIOM to obtain the remote sensing and control required by the application. Driver documentation is provided with the VERSAdos System. A manual, Guide to Writing a Device Driver, M68DRVGD/D1, detailing how to write a device driver that runs under the M68000 Real-time Multitasking kernel (RMS68K) or under VERSAdos is also available.

Usage

Any I/O module in the MVME 400 and 600 Series will operate with any of the following masters in control of the Motorola I/O Channel.

MVME110	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer

MVME600 Series

The MVME600 series is a family of VMEmodule industrial I/O interfaces which communicate in a distributed VME system, by means of the Motorola I/O Channel. These modules are based upon the single width Eurocard form factor, and provide a modular, low cost, reliable I/O function intended for wiring-intensive applications.

MVME600/605	Analog Input and Output
MVME610/ 615/616	Opto Isolated 120 V/240 V Input and Output
MVME620/625	Opto Isolated 30 Vdc Input and Output

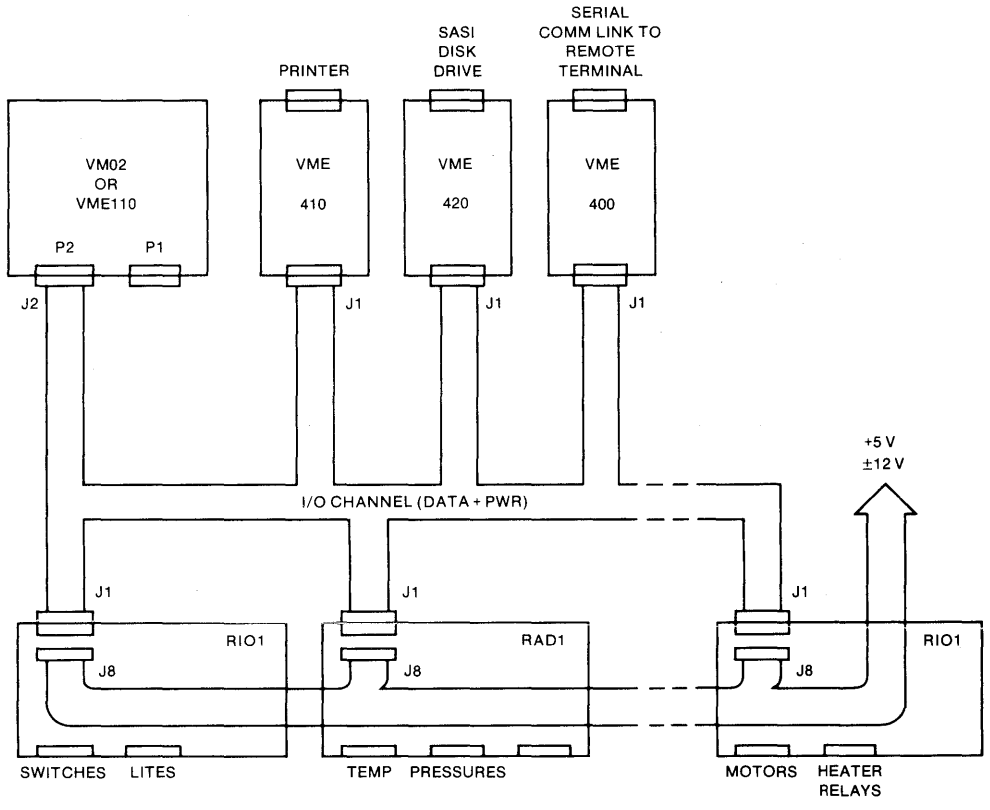
MVME400 Series

The 400 Series is a family of VMEmodule peripheral interface cards designed for modular, low-cost applications. These modules are mechanically compatible with a single Eurocard form factor, and will operate from the Motorola I/O Channel. Use of the I/O Channel allows the user to divert heavy peripheral data flow from the VMEbus onto a local controller to improve system throughput.

I/O Channel

The Motorola I/O Channel is specifically designed to provide efficient, low-cost distributed communciations to peripheral and I/O controller boards. It provides a 12-bit address bus, an 8-bit bidirectional data bus and supports asynchronous operation at data rates up to 2 megabytes/sec. For those modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support all of Motorola's modular product families, including VMEmodules and VERSAmodules.

FIGURE 5 — Recommended Method of Powering the RIOM in a Multi-I/Omodule Application



A typical configuration of a VME-I/O Channel bussed system would consist of a VM02 or VME110 host processor connected to peripheral control boards through the I/O Channel. The power to the VME boards is provided through the I/O Channel cable or the I/O Channel backplane. The Remote RIO1 and RAD1 boards have separate power connectors (J8) through which power is supplied. The I/O Channel cable power is limited to five boards and a cable of 8″ maximum length, unless the power is supplied separately to the backplane using the individual Faston power connectors.

Ordering Information	
Part Number	**Description**
M68RIO1-1	Remote Input/Output Module. Includes provision for mounting up to 16 user-supplied solid-state relay input or output modules. Also includes Fault-indicator for displaying results of diagnostics testing. The two terminal strips are plug-in (readily removable).
M68RIO1-2	(Same as above except terminal strips are fixed.)

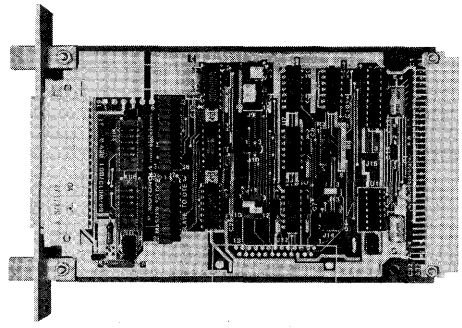
Related Documentation

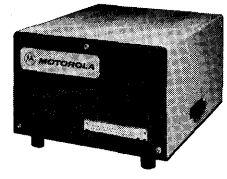
M68RIOCS/D2	Input/Output Channel Specification Manual
M68DRVGD	Guide to Writing a Device Driver

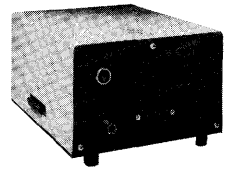
Other Modules in the MVME400 Series

MVME400	Dual Channel RS-232C Port
MVME410	Dual Channel 16-Bit Parallel Port
MVME420	SASI' Peripheral Adapter
MVME435	Buffered 9-Track Magnetic Tape Adapter

Remote Serial Conversion Module







- Actively converts serial port electrical characteristics from RS-232C to RS-485 for multidrop operation.
- Adapts an RS-232C serial port for RS-449 point-to-point operation.
- Increases the transmission rate of a communications channel to 3900 feet (1.2K meters) at 100K bits per second.
- Raises the transmission rate of a communications channel to 600K bits/second at 326 feet (200 meters).
- Offers Manchester II encoding for half and full duplex synchronous operation.
- Supports half and full duplex modes of synchronous and asynchronous operation.
- Accommodates up to 32 drops for full duplex multidrop operation.
- Provides jumper selection of master or slave multidrop operation.
- M68RSC1 module — a single Eurocard form factor board containing a standard Eurocard DIN connector and a standard 37 pin RS-449 connector.
- M68RSC2 — a self-powered standalone enclosure with a single Eurocard form factor board providing externally accessible standard 37-pin RS-449 and 25-pin RS-232C connectors.
- 0° C-70° C Operating Temperature Range

The M68RSC1 and the M68RSC2 Remote Serial Conversion Modules (RSCM) change the electrical characteristics of an RS-232C serial port to those of an RS-449 or RS-485 digital interface. Any system having an RS-232C compatible serial port can use the RSCM to obtain interface conversion. This includes also the serial ports of many Motorola Microsystem products such as VERSAmodules and Micromodules, the EXORset and EXORmacs Development Systems and the VMC 68/2 Microcomputer System.

The RSCM's RS-485 line driver circuitry provides sufficient drive for the serial port to operate in a multidrop (party line) network of up to 32 stations. Or the port can be used for point-to-point operation by virtue of the RSCM's RS-422 balanced voltage driver circuitry.

From the RS-232C maximum of 50 feet, the RSCM increases a port's transmission distance for multidrop and point-to-point operation to 3900 feet at 100K bits per second.

The RSCM supports synchronous and asychronous serial communications permitting half and full duplex operation in multidrop and point-to-point applications. Sockets are provided in the RSCM for strap selection of Manchester encoding (self-clocking data) for use in synchronous half and full duplex operations.

The Remote Serial Conversion Module is available in two versions: M68RSCM1 and M68RSCM2.

Comprising only a 4-inch by 6⅝-inch printed circuit board containing the circuitry, a standard 64-pin DIN connector and a 37-pin, RS-449 connector, the Remote Serial Conversion Module, M68RSCM1, is the basic version. This board can be used in the following ways:

— Mounted in the rear of VERSAmodule Chassis MVMCH1-2.

— Mounted in the rear of VERSAmodule Chassis M68KVMCH1-1 or -2 with added optional M68RIOCC1 Five Slot I/Omodule Card Cage Adapter Kit.

— Mounted to a NEMA cabinet wall, using standoffs.

The second version, RS-232C Terminal Adapter Enclosure, M68RSC2, is comprised of an RSCM mounted in a 4.75 inch by 6.75 inch by 8.00 inch enclosure with a power supply. M68RSC2 contains an externally accessible 37-pin, RS-449 connector. It also contains a power connector and an externally accessible 25-pin RS-232C connector (rather than the 64-pin DIN connector of M68RSC1). Its power supply allows the M68RSC2 to be used in remote locations lacking power and its RS-232C connector permits "to modem" interface such as with a terminal.

Functional Description

The two basic functions of the RSCM, increasing the transmission distance of an RS-232C serial port and translating from an RS-232C interface to a multidrop interface, are accomplished by actively connecting the RSCM from RS-232C to RS-422-compatible balanced-line driver/receiver circuits. A second module at the far end of the line is used to convert the RS-422 back to RS-232C. Thus, an RS-232C to RS-232C link, 3900 feet in length, is provided.

The multidrop (party line) capability provides for connection of up to 32 stations (master and slave stations included). Both the point-to-point and the multidrop modes provide either full-duplex or half-duplex operation.

The same drivers, TI 75174's, and the same receivers, TI 75175's, are used in both modes of operation. The RS-232C drivers and receivers are 1488's and 1489's, respectively.

The RSCM provides EIA RS-232C interface compatibility so that it can be interfaced to the vast majority of remote terminals. (This applies to remote terminals not requiring secondary, or reverse-channel, operation — as though the module were Data Communications Equipment.)

The RSCM looks like a modem to the RS-232C port. In the party line mode of operation, the signal Request To Send (RTS) is used to enable or disable (place in a high impedance state) the RS-422-compatible party line drivers. When RTS is a one, the RSCM will immediately make CTS a one. Software protocol must ensure that only one slave board at a time will have permission to turn RTS on. The TI 75175 receivers are always enabled.

The signal lines supported include: TXD (Transmit Data), RXD (Receive Data), TXC (Transmitter Signal Element Timing), RXC (Receiver Signal Element Timing), RTS (Request to Send), DSR (Data Set Ready), DTR (Data Terminal Ready), DCD (Data Carrier Detect), and SIG GND (Signal Ground).

In the RS-449 mode of operation, the RS-449 signals supported are: SD (Send Data), RD (Receive Data), RS (Request To Send), CS (Clear To Send), TR (Terminal Ready), DM (Data Mode), RR (Receiver Ready), TT (Terminal Timing), ST (Send Timing), RT (Receive Timing), and SG (Signal Ground).

In the party line mode of operation, various configurations of the output are possible through jumper options. One configuration is separate lines for SD, RD, TT, and RT which allows full-duplex, synchronous operation. Another configuration is combined data lines (connect SD and RD), and combined clock lines (connect TT and RT). This configuration allows half-duplex, synchronous operation, over just two pairs of wires. Full-duplex, Manchester-encoded synchronous operation is also jumper selectable.

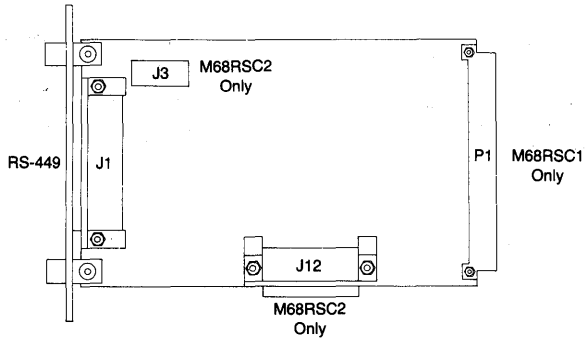


FIGURE 1 — RSCM Connector Locations

RSCM Specifications

Characteristics	Specifications
Power requirements	+5 Vdc @ 500 mA ±12 Vdc @ 70 mA
Mating connectors	P1: (used only on M68RSC1) DIN 41612 specification 96-position receptacle, with rows A and C used. *EXAMPLES:* Erni SKV-N-C64 a, c Winchester 965-6053-0531-13 J1: 37-position D subminiature pin connector *EXAMPLE:* Cannon DCC-37P J3: (used only on M68RSC2) Socket connector, 6-position, on .156-inch centers. *EXAMPLES:* Panduit CE156F18-6 Molex 09-06-0068 J12: (used only on M68RSC2) 25-position D subminiature pin connector *EXAMPLE:* Cannon DBU-25P
Temperature Operating Storage	 0°C to 70°C −40°C to 85°C
Relative humidity	0% to 90% (non-condensing)
Dimensions *M68RSC1:* Height Width Length *M68RSC2:* Height Width Length	*PC board only:* 3.94 in. (10 cm) 6.31 in. (16 cm) 0.6 in. (1.5 cm) *(Enclosure):* 4.75 in. (12 cm) 6.75 in. (17.2 cm) 8.00 in. (20.3 cm)

PC board, board stiffener, and connectors:

5.10 in. (13 cm)
7.38 in. (18 cm)
0.80 in. (2 cm)

Power Connector (J3) Pin Identification

Connector Pin	Mnemonic	Description
1	+5 V	+5 Vdc power for logic circuits.
2	+12 V	+12 Vdc power for logic circuits.
3	CHAS GND	Chassis ground.
4,5	GND	Logic ground.
6	−12 V	−12 Vdc power for logic circuits.

Power Supply Specifications — M68RSC2

Characteristics	Specifications					
Input Voltage	104 to 126 Vac 47 to 440 Hz, single phase		207 to 253 Vac 47 to 440 Hz, single phase			
	NOTE: Power supply is shipped wired for input voltage of 115 Vac.					
Output Voltage	Voltage	Maximum Output Current (Amps)	Adjustment Range (Vdc)	Load Reg. (50% Load Change)	Line Reg. (Low Vac to High Vac)	Max. Ripple (Peak-to-Peak)
	+5	2.0	±0.25	±0.05%	±0.05%	3.0 mVdc
	+12	0.4	±0.60	±0.05%	±0.05%	3.0 mVdc
	−12	0.4	±0.60	±0.05%	±0.05%	3.0 mVdc
Overload Protection	Automatic current limiting foldback; recovers to normal automatically when overload condition is removed.					
Overvoltage Protection	+5 Vdc output protected for overvoltage condition. Overvoltage protection (OVP) trip range is 6.2 Vdc, ±0.4 Vdc for the +5 Vdc output. Overvoltage is reduced to the rated (nominal) output voltage, or less, within 50 μsec. OVP can be reset by power off/on sequence.					
Load Change Transient Response	DC output voltages return to regulated limits within 30 μsec in response to 50% load increase without tripping OVP circuit.					
Temperature	Operating 0° to 70°C (continuous duty). Linear current derating to 40% of maximum rated output between 50°C to 70°C. Storage −55° to 105°C.					
Cooling	Convection. Meets all specification requirements with up to 50% of maximum rated output loading.					

TABLE 1 — RS-232C Signals and Corresponding RS-449 Signals

RS-232C Signal	RS-449 Signal
TRANSMIT DATA (TXD)	SEND DATA (SD)
RECEIVE DATA (RXD)	RECEIVE DATA (RD)
REQUEST TO SEND (RTS)	REQUEST TO SEND (RS)
CLEAR TO SEND (CTS)	CLEAR TO SEND (CS)
DATA TERMINAL READY (DTR)	TERMINAL READY (TR)
DATA SET READY (DSR)	DATA MODE (DM)
DATA CARRIER DETECT (DCD)	RECEIVER READY (RR)
TRANSMITTER SIGNAL ELEMENT TIMING (TXC)	TERMINAL TIMING (TT) or SEND TIMING (ST)
RECEIVER SIGNAL ELEMENT TIMING (RXC)	RECEIVE TIMING (RT)
SIGNAL GROUND (SIG GND)	SIGNAL GROUND (SG)

TABLE 2 — RS-449 Interface Connector (J1) Pin Assignments and Signal Descriptions

Connector Pin	Signal Mnemonic	Signal Name/Description
1	—	Shield
4	SD+	SEND DATA (positive) — These data signals are originated by the data terminal equipment (DTE) and transmitted via the data channel to one or more remote data stations to the data circuit terminating equipment (DCE).
5	ST+	SEND TIMING (positive) — These data signals are originated at the DCE and provide the DTE with transmit signal element timing information. The DTE provides a data signal on circuit Send Data (SD) in which the transitions between signal elements normally occur at the time of the transmissions from OFF to ON condition of the signal on circuit Send Timing (ST).
6	RD+	RECEIVE DATA (positive) — These data signals are generated by the DCE and are in response to data channel line signals received from a remote data station. They are transmitted on this circuit to the DTE.
7	RS+	REQUEST TO SEND (positive) — Signals on this circuit control the data channel transmit function of the local DCE.
8	RT+	RECEIVE TIMING (positive) — These data signals are originated at the DCE and provide the DTE with receive signal element timing information. The transition from ON to OFF condition nominally indicates the center of each signal element on Receive Data (RD).
9	CS+	CLEAR-TO-SEND (positive) — These signals originate at the DCE and indicate whether or not the DCE is conditioned to transmit data on the data channel. The OFF condition of CS indicates to the DTE that it should not transmit data across the interface on circuit SD, because this data will not be transmitted to the line. CS+ controls the RS-232C signal CTS.
11	DM+	DATA MODE (positive) — These signals originate from the DCE and indicate the status of the local DCE. The ON condition indicates that the DCE is in the data transfer mode. DM+ controls the RS-232C signal DSR.
12	TR+	TERMINAL READY (positive) — Signals on this circuit control switching of the DCE to and from the communication channel. TR+ is controlled by the RS-232C signal DTR.
13	RR+	RECEIVER READY (positive) — These signals indicate that the receiver in the DCE is conditioned to receive data signals from the communication channel. RR+ controls the RS-232C signal DCD.
17	TT+	TERMINAL TIMING (positive) — These signals provide the DCE with transmit signal element timing information. The ON-to-OFF transition nominally indicates the center of each signal element on circuit Send Data (SD).
19	SG	SIGNAL GROUND — This circuit connects the DTE circuit ground to the DCE circuit ground to provide a conductive path between the DTE and DCE.
22	SD−	SEND DATA (negative) — Equivalent to SD (positive).
23	ST−	SEND TIMING (negative) — Equivalent to ST (positive).
24	RD−	RECEIVE DATA (negative) — Equivalent to RD (positive).
25	RS−	REQUEST TO SEND (negative) — Equivalent to RS (positive).
26	RT−	RECEIVE TIMING (negative) — Equivalent to RT (positive).
27	CS−	CLEAR TO SEND (negative) — Equivalent to CS (positive).

TABLE 2 — RS-449 Interface Connector (J1) Pin Assignments and Signal Descriptions (continued)

Connector Pin	Signal Mnemonic	Signal Name/Description
28	IS	TERMINAL IN SERVICE — This signal can be jumpered (see header J2) to position "dummy generator," and used anywhere a positive signal is needed.
29	DM –	DATA MODE (negative) — Equivalent to DM (positive).
30	TR –	TERMINAL READY (negative) — Equivalent to TR (positive).
31	RR –	RECEIVER READY (negative) — Equivalent to RR (positive).
35	TT –	TERMINAL TIMING (negative) — Equivalent to TT (positive).
37	SC	SEND COMMON — This circuit is connected to the DTE circuit ground (circuit common), and is used at the DCE as a reference potential for Category II interchange circuit receivers.

NOTE: Pins 2, 3, 10, 14 through 16, 18, 20, 21, 32 through 34, and 36 are not connected.

TABLE 3 — RS-232C Connector (J12) Pin Assignments and Signal Descriptions (M68RSC2)

Connector Pin	Signal Mnemonic	Signal Name/Description
2	TXD	TRANSMIT DATA — Used for transmit data as an output from the connected RS-232C device.
3	RXD	RECEIVE DATA — Used for receive data as an input to the connected RS-232C device.
4	RTS	REQUEST TO SEND — Indicates that terminal wants to send data. On a half-duplex channel, this signal controls direction of data transmission. The signal can be used to enable or disable (high impedance) the transmit data and terminal timing drivers in a multidrop environment.
5	CTS	CLEAR TO SEND — Indicates that terminal can transmit data.
6	DSR	DATA SET READY — Indicates that the RSCM is ready.
7	GND	SIGNAL GROUND
8	DCD	DATA CARRIER DETECT — Indicates to terminal that a suitable data carrier is present. When using Manchester encoding, this signal indicates valid Manchester code is being received.
14	+12 V	+12 Vdc power — Used by RSCM logic circuits.
15	TXC	TRANSMITTER SIGNAL ELEMENT TIMING (DCE source) — Provides timing information for the transmitted data. The ON-to-OFF transition nominally indicates the center of each transmitted data signal element. This signal originates at the Data Communication Equipment (DCE).
16	+5 V	+5 Vdc power — Used by RSCM logic circuits.
17	RXC	RECEIVER SIGNAL ELEMENT TIMING — Provides timing information for the received data. The ON-to-OFF transition nominally indicates the center of the received data signal element. This signal originates at the Data Communication Equipment (DCE).
18	–12 V	–12 Vdc power — Used by RSCM logic circuits.
20	DTR	DATA TERMINAL READY — Indicates that data terminal is ready to transmit or receive data. DTR controls the RS-449 signal TR.
24	TXC (DTE)	TRANSMITTER SIGNAL ELEMENT TIMING (DTE source) — Provides timing information for the transmitted data. The ON-to-OFF transition nominally indicates the center of each transmitted data signal element. This signal originates at the Data Terminal Equipment (DTE).

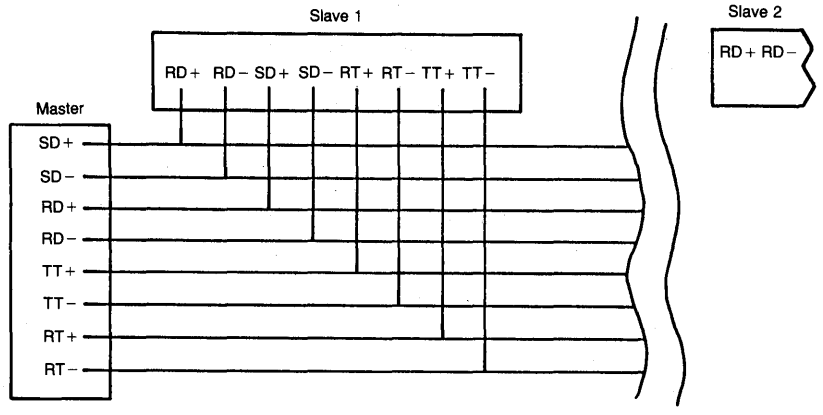
NOTE: Pins 1, 9–13, 19, 21–23, and 25 are not connected.

TABLE 4 — Connector P1 Pin Assignments and Signal Descriptions Identification (M68RSC1)

Connector Pin	Signal Mnemonic	Signal Name/Description
		Signals from MM17 or MM27:
A1	+12 V	+12 Vdc power — Used for RSCM logic circuits.
A2	GND	GROUND
A3	+5 V	+5 Vdc power — Used for RSCM logic circuits.
A5	−12 V	−12 Vdc power — Used for RSCM logic circuits.
A7	DTR	DATA TERMINAL READY — See Notes.
		Signals from VERSAmodule 1, 2, or 80:
A15	TXC	TRANSMITTER SIGNAL ELEMENT TIMING — See Notes.
A16	RXC	RECEIVER SIGNAL ELEMENT TIMING — See Notes.
A17	DTR	DATA TERMINAL READY — See Notes.
A18	DCD	DATA CARRIER DETECT — See Notes.
A19	SIG GND	SIGNAL GROUND
A20	DSR	DATA SET READY — See Notes.
A21	CTS	CLEAR TO SEND — See Notes.
A22	RTS	REQUEST TO SEND — See Notes.
A23	RXD	RECEIVE DATA — See Notes.
A24	TXD	TRANSMIT DATA — See Notes.
		Signals (Power) from VERSAmodule Chassis:
A26	−12 V	−12 Vdc power — Used for RSCM logic circuits.
A28	+12 V	+12 Vdc power — Used for RSCM logic circuits.
A29	+5 V	+5 Vdc power — Used for RSCM logic circuits.
A30	+5 V	+5 Vdc power — Used for RSCM logic circuits.
A31, A32	GND	GROUND
		Signals from MM17 or MM27:
C2	TXD	TRANSMIT DATA — See Notes.
C3	RXD	RECEIVE DATA — See Notes.
C4	RTS	REQUEST TO SEND — See Notes.
C5	CTS	CLEAR TO SEND — See Notes.
C6	DSR	DATA SET READY — See Notes.
C7	SIG GND	SIGNAL GROUND
C8	DCD	DATA CARRIER DETECT — See Notes.
C9	+5 V	+5 Vdc power — Used for RSCM logic circuits.
C10	GND	GROUND
		Signals from VERSAmodule 1, 2, or 80:
C15–C25	GND	GROUND
C26	−12 V	−12 Vdc power — Used for RSCM logic circuits.
C28	+12 V	+12 Vdc power — Used for RSCM logic circuits.
C29	+5 V	+5 Vdc power — Used for RSCM logic circuits.
C30	+5 V	+5 Vdc power — Used for RSCM logic circuits.
C31, C32	GND	GROUND

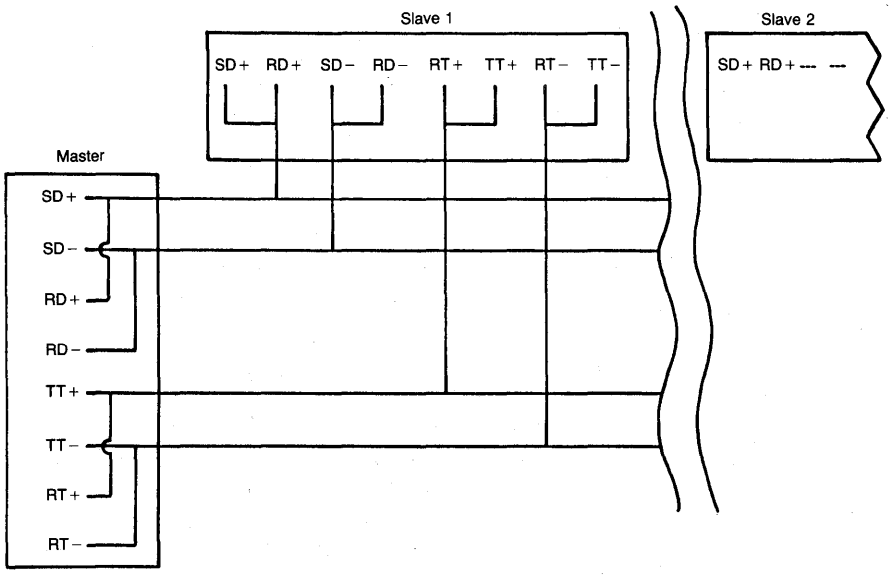
NOTES: See Table 3 for description of this signal. Pins A4, A6, A8–A14, A25, A27, C1, C11–C14, C27 are not connected.

FIGURE 2 — RSCM Interconnections for Full Duplex, Synchronous Operation



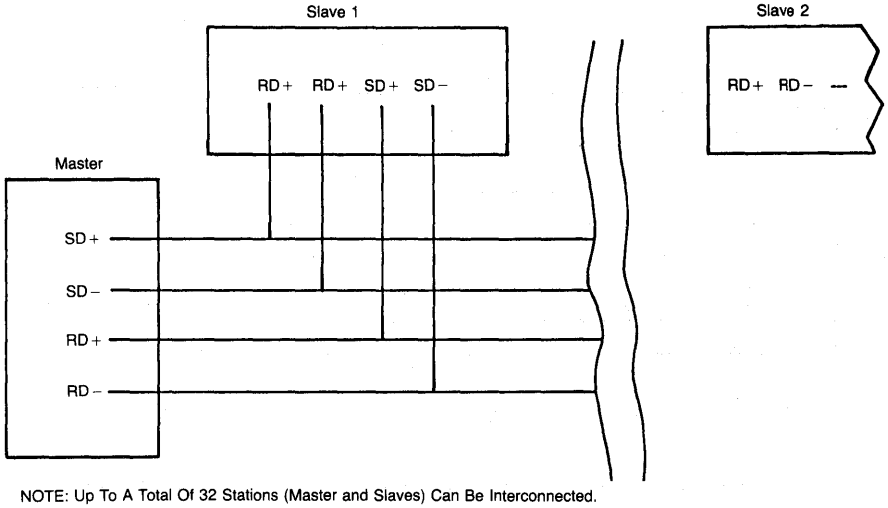
NOTE: Up To A Total Of 32 Stations (Master and Slaves) Can Be Interconnected.

FIGURE 3 — RSCM Interconnections for Half-Duplex, Synchronous Operation



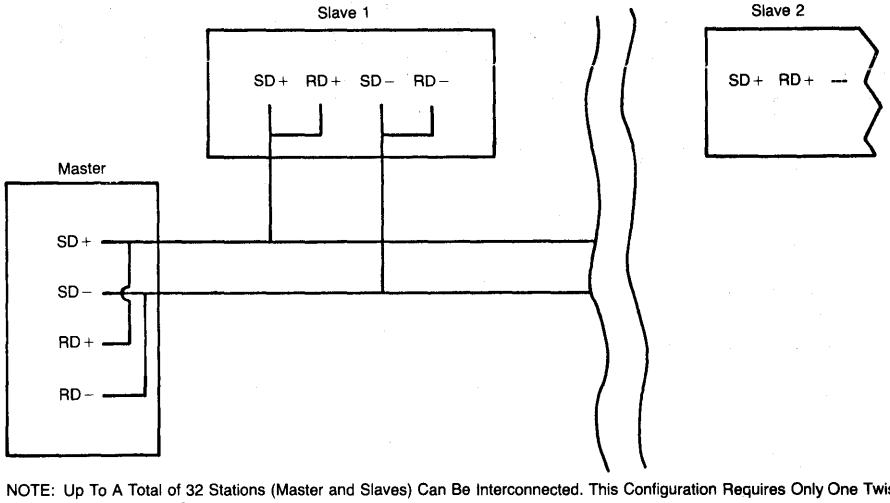
NOTE: Up To a Total Of 32 Stations (Master and Slaves) Can Be Interconnected.
(Header J5 Jumpers Are Installed.)

FIGURE 4 — RSCM Interconnections for Full-Duplex, Asynchronous or Full-Duplex, Manchester-Encoded Synchronous Operation



NOTE: Up To A Total Of 32 Stations (Master and Slaves) Can Be Interconnected.

FIGURE 5 — RSCM Interconnections for Half-Duplex, Asynchronous or Half-Duplex, Manchester-Encoded Synchronous Operation



NOTE: Up To A Total of 32 Stations (Master and Slaves) Can Be Interconnected. This Configuration Requires Only One Twisted Pair of Interconnecting Cable.
(Header J5 Jumpers Must Be Installed.)

FIGURE 6 — RSCM Application in Point-To-Point Operation

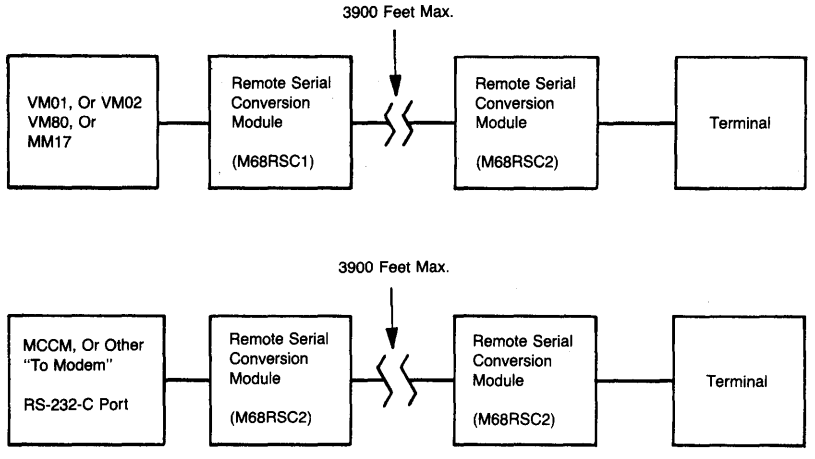


FIGURE 7 — RSCM Application in Multidrop Use

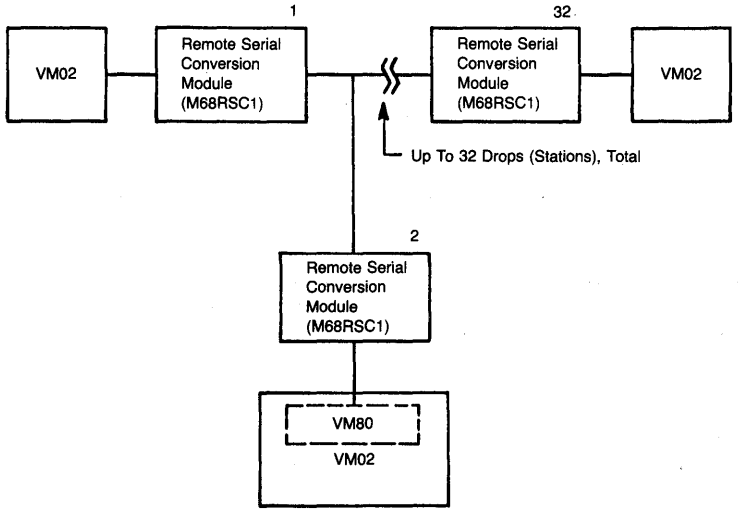
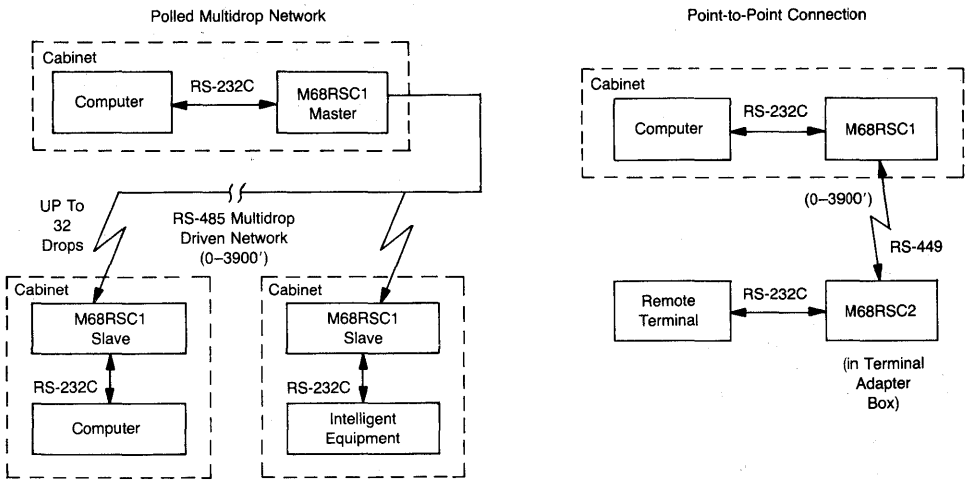


FIGURE 8 — Typical RSCM Systems Applications



Ordering Information

Part Number	Description
M68RSC1	Remote Serial Conversion Module translates an RS-232C port to RS-449 point-to-point with RS-422 balanced line drivers or to multidrop/party line network port using RS-485 drivers. Includes User's Manual.
M68RSC2	RS-232C Terminal Adapter Box with self-contained power supply and Remote Serial Conversion Module. Contains RS-449 37-pin and RS-232C 25-pin connectors. Includes User's Manual.

Documentation

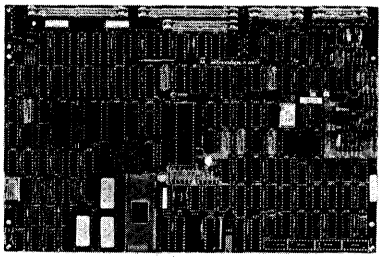
M68RSC1/D1	Remote Serial Conversion Module and Terminal Adapter Enclosure User's Manual.

Related Documentation

The following documentation is related to the RSCM, and is recommended for reference:

M68MM17/D2	Micromodule 17 Monoboard Microcomputer User's Manual
M68KVM01/D1	Monoboard Microcomputer User's Guide
M68KVM30/D1	Multichannel Communications Module User's Guide
M68KVM02/D1	VERSAmodule 02 Monoboard Microcomputer User's Guide
M68KVM80/D1	VERSAmodule Combination Memory, I/O, Time-of-Day Clock User's Manual
M68RAD1/D1	VERSAmodule Remote Analog to Digital Converter User's Manual

Winchester Disk Controller I/Omodule



- Motorola I/O Channel Interface Compatible
- Intelligent Controller Services High Level Host Macro Commands
- Supported by VERSAdos Operating System and by VMEbug
- Separate Winchester and Floppy Disk Data Separators and Generators Controlled by the Intelligent Controller
- Provides Control for up to Two Daisy-Chained 5-1/4" (M68RWIN1-1) or 8" (M68RWIN1-2) Winchester Disk Drives and up to Two Daisy-Chained Double Density Single or Double-sided Floppy Disk drives of Either 5-1/4" or 8" type.
- Designed to Handle up to Four Platter Winchester Drives
- High Performance Winchester Interface allows Data Transfers of Consecutive Sectors (No Interleaving), No Latency on Head Switches and One Revolution Latency on a Cylinder Switch
- 32-Bit Error Correction Code (ECC) for up to 7-Bit Burst Error Correction on Winchester Drives (Host Performs Correction)
- Appears to the Host as a Set of Eight Write and Eight Read Registers on the I/O Channel for Passing Commands, Data and Status Between Host and Controller
- 4K Byte FIFO Enables Data Transfer Rates Greater Than One Mbyte/sec to and from the FIFO Buffer
- 256 Byte Winchester Sector Size; Floppy Sector Size is Host Software Selectable as 128/256/512/1024 Bytes
- Host Initiated Self-Test
- 0°C-70°C Operating Temperature Range
- 12.4" × 8.25" Controller Board Mounts in Motorola M68RMSE1-2 or M68RMSE1-3 RETMA Rack Mountable Enclosures for 5-1/4" and 8" Drives, Respectively

The Winchester Disk Controller (WDC) is a non-Eurocard I/Omodule used to add mass storage capacity to any system having a Motorola I/O Channel-compatible interface. It provides an intelligent interface between a host on the Motorola I/O Channel and up to two Winchester disk drives and two floppy disk drives. The WDC executes high-level commands such as read/write data, format track and verify data and performs implied seeks, automatic track/head switching,alternate sectoring and error checking. Data is transferred using interrupt-initiated block and byte transfers or by processor polling.

Available in two models, one for 8" (SA1000-compatible) drives and one for 5-1/4" (ST500-compatible) drives, the WDC is designed for mounting with two disk drives in a user-supplied enclosure.

For 5-1/4" Winchester drives, the M68RWIN1-1 controller is used. This controller supports up to two Seagate ST500 Series Compatible 5-1/4" Winchester drives and up to two daisy-chained Shugart Series SA400 compatible 5-1/4" floppy drives (or Shugart SA800 Series compatible; 8" floppy drives.

For 8" Winchester drives, the M68RWIN1-2 controller is used. This controller supports up to two Shugart SA1000 Series compatible 8" Winchester drives and up to two daisy-chained Shugart SA800 Series compatible 8" floppy drives (or Shugart SA400 Series compatible 5-1/4" floppy drives).

Both controllers contain two 20 pin radial data connectors for Winchester drives, a single 50 pin connector for the Winchester drive and 8" Floppy disk daisy-chain and a separate 34 pin connector for the 5-1/4" Floppy disk daisy-chain.

To the driver running under a host operating system, the WDC appears as a set of registers on the Motorola I/O Channel: eight write registers for passing to the WDC commands, parameters and data, and eight read registers for passing to the host status, sense information and data. An application typically requires that the physical parameters of mounted drives be described to the host via system generation, and the drives formatted. On subsequent start ups, the driver obtains the required configuration information and transfers it to the WDC RAM for use by the controller firmware. A description of the WDC commands is provided in Table 1.

Data transfers can be made on a 128 byte block or byte basis.

The WDC performs self-test when reset under host software control or at power up. Functional elements tested include the floppy disk controller IC, the FIFO buffer, the FIFO counter and local RAM.

TABLE 1 — Winchester Disk Controller Commands

Command	Description
Configure Drive	Describes to the WDC the physical parameters of the mounted drives, i.e., the number of cylinders and number of data surfaces
Recalibrate	Positions the read heads to track 000
Check Drive Status	Returns to host "Seek Complete", "Write Protected" or "Ready" drive status
Seek	Initiates a seek to the specified sector returning immediately to host before the seek is completed, to allow overlapped operation
Read Sectors	Reads specified (1 to 256) consecutive sectors from the disk
Write Sectors	Writes specified (1 to 256) consecutive sectors to the disk
Scan Sectors	Checks the ECC of specified (1 to 256) consecutive sectors (no data transfer)
Format Drive	Initializes (rewrites) all ID headers and data sectors of the specified drive
Format Track	Initializes (rewrites) the ID headers and data sectors of the specified track
Format Alternate and Defective Tracks	Initializes (rewrites)the ID headers and data sectors of the specified tracks to identify alternate and defective sectors to facilitate transparent re-mapping of defective disk sectors
Read ECC	Reads one sector and the appended ECC bytes (diagnostic)
Write ECC	Writes one sector and appends ECC bytes. Host supplies ECC bytes (diagnostic)
Read Track	Transfers image of specified track including ID headers, data and ECC bytes to host (diagnostic)

A jumper header on the WDC provides selection of several floppy disk and Winchester parameters. These Include:

- Winchester
  - Fixed or cartridge drives.
  - Un-buffered or buffered stepping rate (3.2 ms or 20 μs).
- Floppy Disk
  - 5-1/4" or 8" drives.
  - 48 or 96 TPI density.
  - Stepping rate (3.2, 10, 20 or 35 ms).

Software Driver

A driver for this Motorola I/O Channel interface module is incorporated in the VERSAdos M68000 Real-Time Operating System. It provides a host-resident application program with a ready means of communicating with the module and of obtaining the necessary and optional service functions required for controlling the peripheral resource to which the module is connected. Driver documentation is provided with the VERSAdos System. A manual, "Guide to Writing a Device Driver", (M68DRVGD/D1), detailing how to write a device driver that runs under the M68000 Real-Time Multitasking kernel (RMS68K) or under VERSAdos is also available.

Usage

M68RWIN1 will operate with any of the following masters in control of the Motorola I/O Channel.

MVME110	VMEmodule Monoboard Microcomputer
M68KVM02-3	VERSAmodule Monoboard Microcomputer
M68KVM03	VERSAmodule Monoboard Microcomputer

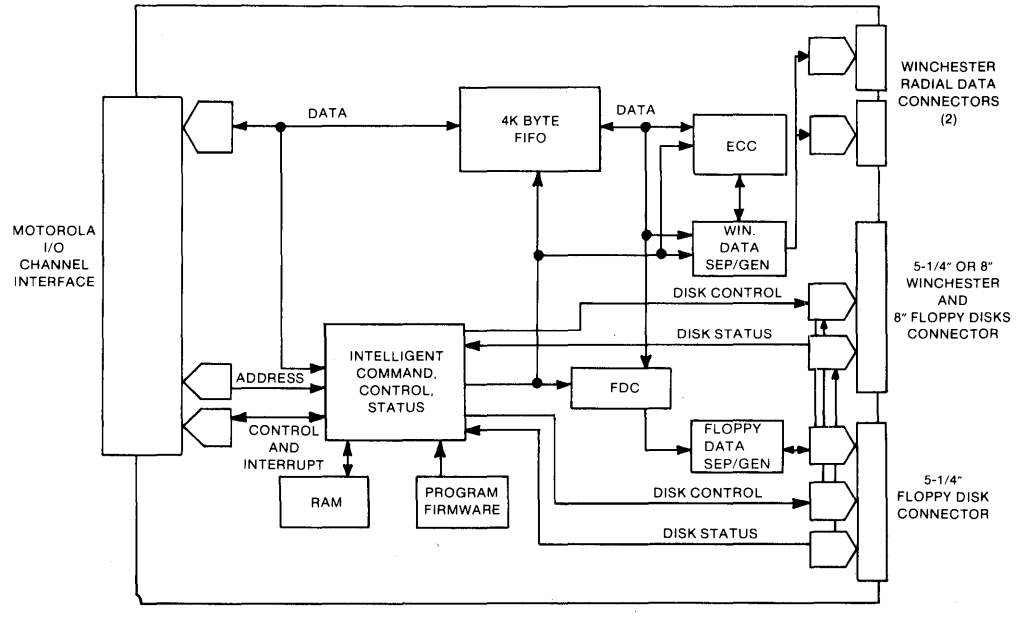
I/O Channel

The I/O Channel is specifically designed to provide efficient, low-cost distributed communciations to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus and supports asynchronous operation at data rates up to 2 megabytes/sec. For those modules performing time-critical operations, four prioritized interrupt lines are also provided. The I/O Channel is designed to operate over either a backplane, or a ribbon cable. The I/O Channel, and related I/Omodule products support all of Motorola's modular product families: VMEmodules and VERSAmodules.

Related Products

The 400 Series is a family of I/Omodule peripheral controller cards designed for modular, low-cost applications. These modules are mechanically compatible with a single Eurocard form factor, and operate from the Motorola I/O Channel. Use of the I/O Channel allows the user to divert heavy peripheral data flow from the VME bus onto a local controller to improve system throughput.

Functional Block Diagram
Winchester Disk Controller

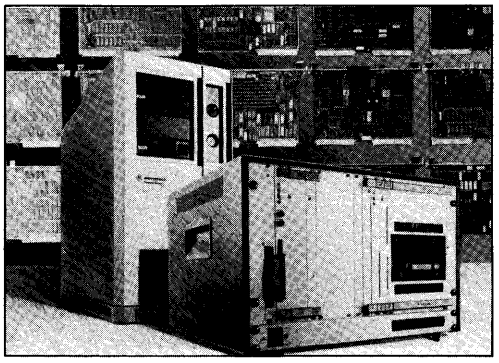


Mechanical and Environmental Specifications

Characteristic	Specification
Power Requirements	+5 Vdc ±5% @ 6.0 A max. -5 Vdc ±5% (-8 to -18 Vdc optional) @ 100 mA max
Temperature	
Operating	0° C to +70° C
Storage	-55° C to +85° C
Relative Humidity	0-95% (non-condensing)
Dimensions	
Length × Width	315 mm × 210 mm (12.4 in × 8.25 in)

Ordering Information

Part Number	Description
M68RWIN1-1	5-1/4″ Winchester Disk and 5-1/4″ or 8″ Floppy Disk Controller I/Omodule. Includes User's Manual.
M68RWIN1-2	8″ Winchester Disk and 5-1/4″ or 8″ Floppy Disk controller I/Omodule. Includes User's Manual.
M68RWIN1/D1	Winchester Disk Controller I/Omodule User's Manual.



Creation and maintenance of software for any microcomputer application can be a substantial burden; for some it is the major expense. This is particularly true where, for efficient operation, a high performance application requires a real-time, multitasking environment. Thus, the availability of operating system support, the use of which can significantly reduce the software development effort, should weigh heavily in a designer's choice of a line of board level system components for his application.

Motorola supports its 16/32-bit microcomputer board families with two full operating systems: the SYSTEM V/68 Operating System and the VERSAdos Real-Time Multitasking Operating System plus the RMS68K Real-Time Executive, the multitasking kernel of VERSAdos around which a real-time application system can be built. These are described in this book. To extend the development power and utility of both operating systems, Motorola also offers a broad range of supporting software, including high level languages and debugging packages. Use of these software tools can save much of the design, coding and testing effort necessary for system development.

Operating System Data Sheets

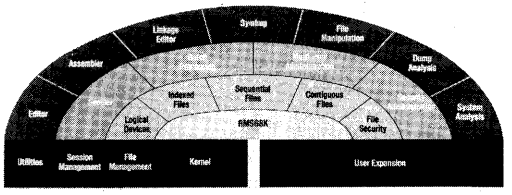
5

VERSAdos™ 4.5 Operating System

- Real-time, Multitasking, Multiuser Operating System
- Real-time, Multitasking Kernel — RMS68K
- System Initialization
- Complete File Management
- Complete I/O Management
- Complete User Interface
- Program Loader
- System Generation Facility
- Supports Entire MC68000 Microprocessor Family
- Floating Point Coprocessor Support for the MC68020
- Fully Supports VMEbus
- Fully Supports VERSAbus
- Factory Configured for VMEmodules and VERSAmodules
- Device Drivers for all VMEmodules, VERSAmodules and I/Omodules — both object and source code.
- Powerful System Utilities: File Creation and Manipulation, Analysis and Debugging, Disk Management and Target Customization

VERSAdos is a real-time, multiuser, multitasking Operating System for the Motorola MC68000 family of microprocessors. Its modular and multilayered design supports a variety of application environments, and is especially well suited to real-time, on-line control systems. This Operating System serves as a major software building block for real-time applications which use VMEmodule and VERSAmodule board products and the MC68000 family. VERSAdos provides the basic capability to service a wide range of requirements — including process control, data acquisition, inventory control, and commercial and scientific applications.

Software has become increasingly more important in the application of the microprocessor to solve problems. Control software extends the capabilities of the MPU and makes maximum use of available system resources. VERSAdos is an Operating System oriented to solving real-time application problems as well as providing support for general purpose solutions. The prime function of VERSAdos is to provide a convenient interface between the user and system hardware. It can be used in systems built around Motorola VMEmodule and VERSAmodule Monoboard Microcomputers, or in customer-designed hardware incorporating the Motorola MC68000 MPU family. All tasks performed by the system are initiated as operator requests, program requests, or hardware interrupt requests.



The basic functions of the VERSAdos Operating System include the following:

- Real-time I/O Processing
- Servicing of Directly Connected Interrupts
- Provision for Processing Interrupts
- Multiprogramming of Real-time Tasks
- Intertask Communication
- Semaphores
- Device Independent I/O
- I/O and File Handling Services
- System Utilities
- Memory Allocation and Management
- Task Management and Services
- Task Scheduling Based on Task Priority Levels (0 to 255)
- Support for Server Tasks (user customized expansion of the Operating System)
- Generic Device Libraries
- File Management (program and data catalogued into disk directories and referenced by file name)
- System Generation
- System Security

The resources of the Operating System are the MC68000, memory, I/O facilities and secondary storage. Application programs are separated and relieved of the necessity of direct interaction with system hardware by the Operating System. Programs communicate input/output requests via the Operating System in a readily understood protocol.

The Operating System is responsible for accepting, checking, interpreting, and expediting user requests. In order to fulfill its requested task, the Operating System (OS) may call upon OS support routines for assistance. System support routines normally perform functions that are not directly accessed by the application program. These routines assist in operator control, computer memory management, the loading of task segments, or overlays and input/output control for various hardware subsystems.

The degree of user interaction with the system depends on the application. This ranges from almost unattended dedicated process control operation to near full-time attention in such systems as air traffic control.

User requests may be made for:

- Program Load/Initiation
- Hardware Control and Status
- Reading or Setting Current Time of Day
- Requesting I/O
- Storage and Retrieval of Data Files
- Resource Allocation/De-allocation
- Timer Services

VERSAdos DESIGN SUMMARY

The VERSAdos Operating System permits programs to execute in dynamically-assigned, variable-length memory segments with read/write privileges. Instructions and data can be located in separate memory areas, which enable sharing of program code and re-entrant coding practices. A process-to-process facility permits communication between independent programs or nodes of a distributed system.

The heart of the operating system is a real-time executive which provides task services and supports memory allocation. It also allows inter-task communication, provides exception monitor facilities, and handles system interrupts.

The input/output subsystem supports device independence, logical input/output and overlapped computation during physical input/output. New device drivers can be added without impacting the central core portion of the Operating System. Both sequential and random record access are supported by the VERSAdos Operating System.

A powerful file management system supports three types — contiguous, sequential and indexed sequential. Other features include volume and file protection, shared file access, dynamic file allocation and fixed or active protection.

VERSAdos FUNCTIONS

VERSAdos is a coherent group of tasks which use the services of RMS68K. User tasks and VERSAdos communicate via a server mechanism with RMS68K handling all synchronization and scheduling. As a result, VERSAdos is an orderly, easily extendable operating system.

REAL-TIME CONCEPTS

A real-time system responds to external events as they occur. Since the precise time of these external events is an unknown factor, a real-time system is said to execute asynchronously.

Unlike a batch system where one operation is completed before a new operation is started, a real-time system can delay the completion of one operation in order for another operation to be started, continued, or completed. This mechanism (where more than one operation is in progress at a given time) is called concurrent processing. Even though only one operation can be executed at a given time using a single central microprocessing unit, the concurrent processing mechanism of a real-time system gives the illusion of several operations executing simultaneously.

A real-time application system can be broken down into several tasks. A task is a function (or operation) which can execute concurrently with other tasks/functions. A task can be written to process a single type of event, or it can process more than one type of event.

THE EXECUTIVE

The VERSAdos Executive, known as RMS68K, is the nucleus of the Operating System. RMS68K is comprised of an inner kernel (or nucleus) that supports the real-time, priority-driven, multitasking environment. It services all hardware and software interrupts and dispatches them to the proper task for processing. The Executive also resolves conflicts resulting from competing tasks attempting to use the MPU, and has facilities which permit inter-task communication and task synchronization. The Executive provides protection of the user environment and diagnostic feedback during error conditions. Like the rest of VERSAdos, the Executive is designed in a structured fashion. RMS68K may be used as supplied, or tailored to an exact configuration. Modules may be deleted and/or added to RMS68K, based on the users' requirements. RMS68K may also be ROM based or RAM based. Refer to the RMS68K 4.4 data sheet (ADI1165) for detailed information on the Executive.

Nucleus

The nucleus performs the following functions:

- Receive all Hardware and Software Interrupts and Dispatch them to the Proper Task for Processing
- Dispatch Tasks Competing for use of the Microprocessing Unit
- Run and Schedule Background Jobs for Device Drivers
- System Initialization
- Diagnostic Feedback During Error Conditions

Resource Managers

A resource manager consists of data structures and several directives, each providing one of the specific services required for managing that resource. The eight resource managers perform the following:

- Provide Inter-task Communication and Synchronization
- Manage and Allocate Memory
- Provide Task Initialization, Termination, and Query
- Notify Tasks of the Expiration of Pre-selected Time Intervals
- Provide Task Synchronization using Semaphores
- Allow One Task to Observe and Control the Behavior of One or More Tasks
- Simulate and Claim Hardware Interrupts, and Configure Interrupt Service Routines
- Support the Special Requirements of Operating System Tasks

Real-Time Tasks

A real-time task is one that executes specifically in the high-performance domain known as real-time. One attribute of this domain is that the entire address space accessible by the task must be mapped, with the task's logical address identical to the system's physical address.

Real-time tasks are exempt from any parameter checking. Thus, real-time tasks execute much faster than non-real-time tasks, but in a less constrained environment. Tasks may be debugged in the non-real-time mode for parameter checking and changed to real-time for actual target execution.

All tasks may use the GTTASKID; however, real-time tasks must use this directive to allow the fast access mode to work properly. All RMS68K directives are available to the real-time tasks.

Coprocessor Support

RMS68K fully supports the 68020 coprocessor interface for both the MC68881 floating point coprocessor and user defined coprocessors.

When a task wishes to use the floating point coprocessor, RMS68K must be notified. This is accomplished when the requesting task issues the CRXTCB directive (task manager). Tasks linked with the VERSAdos Linkage Editor version 1.9 may use the C = 68881 command line option to specify usage of the floating point coprocessor. These tasks are not required to issue the CRXTCB directive.

RMS68K Structure

RMS68K is structured in logical layers with each layer performing a particular range of functions.

The layers may be viewed as internal, external, and channel management layers.

The functions provided by the internal layers are used by RMS68K to manage the processor, tasks, and physical devices such as timers. In addition, the internal layer performs work on behalf of requests from user tasks.

The functions provided by the external layer are directly available to user tasks through the use of directives. A directive (or request) contains all the information needed by RMS68K to perform the desired function.

The optional channel management routines (CMRs) provide the channel-oriented, physical I/O functions. All device drivers supplied with RMS68K and VERSAdos use the CMR mechanism for I/O.

Device drives are included with RMS68K to facilitate the use of RMS68K with CPU boards and peripheral controllers. Custom device drivers are easily added to the system using the tools available on both VERSAdos and Motorola's SYSTEM V/68 operating systems.

Directives

An RMS68K directive consists of a name for identification by the user, a number for machine recognition, and a body of code to provide the service. In addition, most RMS68K directives require that the task provide additional information about the requested service. This information is collected into a block of data known as a parameter block.

Memory Requirements

RMS68K is made of several relocatable object modules, each module performing a unique function. The user may configure RMS68K for exact system requirements by excluding particular functions in the final executable system, thus decreasing memory requirements. The maximum memory requirements are given in Table 1.

Table 1. RMS68K Memory Requirements

Layer	RAM (Parameter Area)	ROM/RAM (Program Code)
Internal and External	3Kb	23Kb
Channel Management	0.25Kb per channel (minimum)	2Kb

Application Hardware Requirements

The hardware requirements for an RMS68K based application system are:

- MC68000, MC68010, or MC68020 Microprocessor
- Appropriate Memory
- Tick Timer
- Optional Memory Management Unit
- Optional Real-Time Clock

The amount of memory required will vary from one system to another, depending upon the system environment, user-supplied code and data, and the RMS68K functions configured in the user system. The maximum memory required can be decreased by including only the necessary RMS68K functions in the system.

If the system is to use the delay task and periodic task activation functions of RMS68K, a clock must be provided. This can be accomplished by means of a software clock mechanism that the user configures as part of RMS68K, or by means of a hardware device such as the Motorola MC6840 Programmable Timer Module.

The requirements listed may be satisfied through use of the Motorola VERSAmodule Monoboard Microcomputer, VMEmodules or a user-designed MC68000 microprocessor-based board.

Supported Hardware

In addition to the M68000 family of microprocessors, RMS68K also supports a variety of memory management units, timers, CPU boards, and peripheral boards. Table 2 lists some of the more popular Motorola devices supported by RMS68K.

Benchmarks

Table 3 is a listing of benchmarks that reflect the timing characteristics of some often-used RMS68K system calls. The data includes the actual timings of RMS68K.

The hardware used to determine the benchmarks included a MVME123 CPU module which features a 12.5 MHz MC68010 microprocessor, 512Kb of local RAM, and is not equipped with a Memory Management Unit. The RAM was configured with parity disabled to provide one wait-state access. The onboard cache was enabled.

Table 2. RMS68K Supported Devices

	Microprocessor	Memory Mgmt. Unit	Tick Timer	Real-Time Clock
MPU Modules				
M68KVM03	MC68010	MC68451	M146818	M146818
M68KVM04	MC68020	MMB851	Z8036	none
MVME101	MC68000	none	M6840	none
MVME110-1	MC68000	none	M6840	none
MVME117	MC68010	none	M6840	MK58274
MVME120/121	MC68010	MC68451	MK68901	Optional M146818 on MVME050
MVME122/123	MC68010	none	MK68901	Optional M146818 on MVME050
MVME130	MC68020	none	Z8036	none
MVME131	MC68020	MMB851	Z8036	none
MVME133	MC68020	none	MK68901	MK58274
Development Systems				
EXORmacs	MC68000	EXORmacs	M6840	none
VME/10	MC68010	MC68451	M146818	M146818
SYS1121VY	MC68010	MC68451	MK68901	Optional M146818 on MVME050
SYS1131VY	MC68020	MMB851	Z8036	MK58274

This hardware configuration was chosen for the benchmarks in order to reflect the actual time involved in the code with a minimum amount of hardware overhead.

Table 3. Benchmarks

System Call	Execution Time
Context Switch	91 μs
Interrupt Latency	15 μs
Interrupt Latency (post processing)	15 μs
Timer Interrupt	36 μs
Send Message	119 μs
Receive Message	97 μs
Wakeup with no wait pending	67 μs
Wait with wakeup pending	70 μs
Wait for Semaphore	70 μs
Signal Semaphore	88 μs
Relinquish Task Control	124 μs
Initiate I/O	172 μs

Task Control

Associated with each task existing in the system is a task control block (TCB). the TCB contains information about the task which allows RMS68K to maintain control of the task's execution, account for resources allocated to the task, and ensure task protection. The TCB remains associated with one task throughout that task's existence.

Task control is accomplished by RMS68K moving the tasks through various task states.

A task can make a transition from one state to another when any of the following actions take place:

- A task control directive is issued by the task while it is in the running state
- A task control directive is issued by another task while it is in the running state
- An event is placed in the task's asynchronous service queue
- An exception occurs in the task while in the running state
- Special function handling is initiated by RMS68K (such as timeouts, semaphore signaling, interrupt handling)

EVENT MANAGEMENT

The Event Manager is the facility within RMS68K that enables tasks to communicate declarative knowledge to other tasks. The event manager may also be known as: inter-task communication, inter-process communication, or message passing. Declarative knowledge has two facets: factual knowledge (the sky is blue), and temporal or event knowledge (the sun just went down). The RMS68K event manager supports the communication of both types of declarative knowledge.

Other features of the event manager:

- Operates in either a synchronous or an asynchronous mode
- Supports two modes to communicate declarative knowledge:
  - — Move the data
  - — Pass a pointer to the data
- Supports variable length messages
- Automatically queues messages if the receiving task is not ready to receive
- Supports task to task communication or driver to task communication
- High performance

The directives contained within the event manager are:

GTASQ	Allocate an ASQ for the target task.
SETASQ	A task enables or disables its ASQ, ASR, and/or default receive buffer.

QEVNT	An event is sent to the target task.
GTEVNT	A task moves itself into the GETTING EVENT state until an event arrives at which time the task is dispatched to the instruction following the GTEVNT directive call.
RDEVNT	The next event in the requesting task's ASQ is moved into the task's receive buffer.
WTEVNT	A task moves itself into the WAIT FOR EVENT state until an event arrives at which time the task is dispatched to its ASR.
RTEVNT	Return to the point of task interruption on completion of that task's ASR processing.
DEASQ	Deallocate the requesting task's ASQ.

MEMORY MANAGEMENT

The Memory Manager within RMS68K consists of the directives that support the concept of memory as a resource that can be allocated to a task, shared between two or more tasks, transferred from one task to another, and deallocated or returned to the system. The user can select beween memory allocation algorithms. The best-fit method (original) is best suited when memory resources are scarce. The buddy buffer method is best suited when a memory management unit is used, and it requires more system memory.

Segment Directives

GTSEG	Allocate a new code or data segment to the target task by placing it within the task's address space.
TRSEG	Remove a segment from the requesting task's address space and place it within the address space of another task.
RCVSA	Return a description of the specified segment to the requesting task.
DESEG	Delete a code or data segment from the target task's address space.

Shareable Segment Directives

DCLSHR	Make a segment available for shared access.
ATTSEG	Place an existing shareable segment within the requesting task's address space.
SHRSEG	Place an existing shareable segment within another task's address space.

Utility Memory Directives

MOVELL	A task requests that data be copied from the logical address space of one task to the logical address space of another task.
MOVEPL	A system task requests that data be copied from any physical address to a logical address within a target task's address space.
FLUSHC	Flushes all user mode entries from all caches known to the Executive.

TASK MANAGEMENT

The Task Manager consists of the directives that support the concept of tasks as resources that can be created, terminated, started, stopped, temporarily halted, reawakened, and inquired about. The task manager maintains a list of all tasks currently known to the system and the TCBs that contain information describing the current state and resources allocated to each task. The task manager imposes no limit on the number of tasks within a system; the limiting factors are typically the amount of available memory and the application's performance requirements.

The task manager functions are divided into three categories: initialization and termination, synchronization, and query.

The directives supporting task initialization and termination are:

CRTCB	Create a TCB for a new task, initialize the name, session number, monitor, priority, attributes and entry point fields, and place the task in the DORMANT state.
CRXTCB	Create a EXTCB for an existing task in order to use the coprocessor (68020 only).
START	Move the target task from the DORMANT to the READY state. This directive can also initialize the task's registers and specify its monitor.
SETPRI	A task changes its own priority or the priority of another task to the specified value according to the restrictions described in task priority.
STOP	Move the target task to the DORMANT state.
TERMT	Terminate a target task by releasing resources and deleting the TCB of the target task so that the task no longer exists in the system.
TERM	Terminate the requesting task by releasing all resources and deleting its TCB so that the task no longer exists in the system.
ABORT	Initiate an abnormal termination of the requesting task; its TCB is deleted and the task no longer exists in the system.
ABORTC	Initiate an abnormal termination of the requesting critical system task; its TCB is deleted and the task no longer exists in the system.

The directives supporting "primitive" task synchronization are:

WAIT	A task moves itself into the WAIT state.
WAKEUP	Move the target task from the WAIT state to the READY state.
SUSPND	A task moves itself into the SUSPEND state.
RESUME	Move the target task from the SUSPEND state to the READY state.
RELINQ	A task moves itself to the READY state forcing RMS68K to execute a dispatch cycle.

TIME MANAGEMENT

RMS68K's Time Manager supports two concepts of time: elapsed and calendar. The directives supporting elapsed time involve notifying a task when a quantum of time has expired. The directives supporting calendar time allow a task to inform the time manager of the current date and time (e.g., March 21, 1985; 12:04), or to ask the time manager for the current date and time.

The time manager can also determine when a task's timeslice has expired (if timeslicing was enabled at initialization). When a timeslice expires, the time manager places the running task back on the READY list after all tasks of the same priority and forces a dispatch cycle.

The concept of elapsed time is supported by three RMS68K directives:

- DELAY — A task moves itself into the DELAY state for a specified period of time.
- DELAYW — A task moves itself into the DELAY state for a specified period of time and returns if the time expires, a WAKEUP occurs, or an event is queued.
- RQSTPA — A task requests the timed periodic activation of itself or another task.

Calendar time is supported by two directives:

- STDTIM — A system task sets the system date and time.
- GTDTIM — A task obtains the current system date and time.

SEMAPHORE MANAGEMENT

The Semaphore Manager provides sophisticated synchronization primitives that are used to coordinate the activities of multiple tasks or to arbitrate access to shared resources.

The semaphore manager provides synchronization services via the directives:

CRSEM	A task creates a new semaphore or resets the initial count of an existing semaphore.
ATSEM	A task attaches to a semaphore and acquires use of the semaphore. If the semaphore does not exist, it is created and given an initial signal count.
WTSEM	A task requests and, if necessary, waits for semaphore-controlled access.
SGSEM	A task signals release of a semaphore-controlled resource.
DESEM	A task detaches from a semaphore.
DESEMA	A task detaches from all semaphores to which it is attached.

TRAP SERVER MANAGEMENT

Most operating systems require a class of tasks that provide services and control resources on a system-wide basis. These server tasks should be able to respond to a request from any task in the system, execute the requested service, and provide feedback to the client task about the success or failure of the request. A typical use for a server task would be to implement an input-output system, a file management system, or a database manager.

The RMS68K Trap Server Manager supports this class of server tasks by providing them with the following privileges and capabilities:

- A server task can respond to a request from any task in any session.
- The event that results from the request contains the client task's task_id, the directive number of the requested service, and a copy of the parameter block describing the request, if required.
- The server can prohibit the client from running until the service is completed.
- The server can provide feedback to the client by altering its condition codes and two of its registers.
- The server can specify what state the client task should be in when released from the server's control.
- The server can request notification of task termination to implement any necessary termination processing.

The VERSAdos real-time operating system is implemented as a collection of server tasks running under the control of RMS68K. Therefore, server tasks can be used to implement a special purpose operating system or to add domain specific extensions to VERSAdos.

The directives used for server task control are:

SERVER	A task establishes itself as a server task.
AKRQST	A server task acknowledges receipt of completion of a request by placing the requesting task into an appropriate state.
DERQST	A server task controls the receipt of requests for service.
DSERVE	A server task initiates orderly shutdown of service.

EXCEPTION MONITOR MANAGEMENT

The Exception Monitor Manager provides services to a class of tasks called exception monitors. An exception monitor is a task that can indirectly observe and control the execution of a target task. The events that an exception monitor is capable of observing are those events (called exceptions) that cause the processor to switch from executing user mode code to supervisor mode code. These exceptions include all the TRAP instructions and error conditions such as bus error, illegal instruction, and divide by zero.

The exception monitor can control the target task by reading or writing to its memory or registers, setting breakpoints, or tracing through the program in one of three trace modes.

A typical use for exception monitors is to implement symbolic debuggers or timing analysis programs. Another use would be to increase system security by observing and reporting an erratic task behavior.

Listed below is a summary of the directives contained within the exception monitor manager.

EXMON	A target task becomes associated with an exception monitor task.
EXMMSK	Events of interest to an exception monitor task are specified.
REXMON	A target task executes as directed by an exception monitor task.
RSTATE	An exception monitor task receives the current state of one of its target tasks.
PSTATE	An exception monitor task sets the current state of one of its target tasks.
DEXMON	A target task detaches from an exception monitor task.
CEXMSK	Coprocessor events of interest to an exception monitor task are specified (68020 only).
CRSTAT	An exception monitor task receives the current state of a specific coprocessor being used by one of its target tasks (68020 only).
CPSTAT	An exception monitor task sets the current state of a specific coprocessor being used by one of its target tasks (68020 only).

EXCEPTION MANAGEMENT

Exceptions are those conditions that cause the M68000 Family processor to switch from executing in the user mode to the supervisor mode. When an exception occurs the current status register, program counter, and the optional vector offset register are pushed on the supervisor stack and an exception vector number is generated, either internally by the processor or externally by the interrupting device. This vector number is within the range $0 to $FF and is multiplied by four to index into the longword vector table to access the address of the routine that handles that exception.

These exceptions may be broken down into two classes:

- Interrupts that are caused by a request for service from some external device.
- Program exceptions resulting from the program causing the processor to enter a state that requires supervisor mode processing.

Program exceptions are further broken down into two sub-classes:

- Error exceptions such as bus errors or divide by zero faults that indicate error conditions within the program.
- Trap instructions indicating that the user program is requesting service from some supervisor mode routine.

The Exception Manager provides services to support all these exception conditions via the directives:

CISR	A task configures a portion of its code to act as an Interrupt Service Routine (ISR) in response to a particular exception.
SINT	A task simulates an exception (interrupt).
RTE	A task returns from an exception. (Do not confuse the RMS68K RTE directive with the M68000 RTE instruction.)
EXPVCT	A task announces the handling of its own exceptions.
TRPVCT	A task announces the handling of its own trap instructions.
RESVCT	A task announces the handling of its own reserved vectors (68020 only).

RMS68K in ROM

A ROMed system can be built which includes the RMS68K Kernel and user applications. This system can optionally include an I/O sub-system and device drivers which allow device assignment and access. The shareable PASCAL run-time library may also be included in the system. Additionally, a table-driven task initiator (TDTI) can be included in the configured system. TDTI allows the user to modify his code without necessitating that it be sysgened and burned into ROM for each test cycle. User code can be downloaded into RAM and run until it is fully tested.

DEVICE I/O HANDLERS

Channel Management Request (CMR) routines reside as part of the Executive, RMS68K. CMR logically manages channels and provides the link between memory mapped I/O space, interrupt vectors, and interrupt service routines. CMR also provides the link between requestor commands and command service routines.

Channel Definition

A channel is defined to be a single contiguous portion of memory mapped I/O space, associated with a condition of interrupt.

A channel has a corresponding hardware interrupt vector number, a hardware priority level, and a software priority number. These three items are used to link devices into polling chains, which are used by a polling routine to determine which channel is associated with an incoming interrupt. When a channel is allocated, the CMR handler creates a channel control block (CCB) for that channel, which contains all of the information needed by the CMR handler to manage that channel. The CCB is then placed into the appropriate polling chain. There is a polling chain for every external interrupt vector. The CCB's are chained according to the software priority number — those with higher software priority numbers will be nearer to the head of the chain, and thus serviced more rapidly when an interrupt occurs.

When an interrupt occurs, control is passed through the first CCB (via a JSR) to the CMR interrupt handler routine. The CMR handler will do a minimum state save, resolve the CCB address, and call the appropriate I/O handler. If the I/O handler returns without claiming the interrupt, CMR will call the handler of the next CCB chained for that vector. This continues until the chain is exhausted.

CMR Handler

User task interactions with channels are performed through requests made to the channel management request (CMR) handler. The following CMR functions are available:

ALLOCATE	Define a channel to the system.
DELETE	Remove a channel definition from the system.
ATTACH	Establish a logical connection between a task and a channel.
DETACH	Dissolve the logical connection between a task and a channel.
PUT ON LINE	Remove a channel from off line status, making it available for subsequent CMR ATTACH requests.
TAKE OFF LINE	Put a channel into off line status, making it unavailable for CMR ATTACH requests until a PUT ON LINE request is made.
INITIATE	Invoke the appropriate I/O handler which will perform the actual I/O.
HALT	Terminate the I/O in progress on a channel.
RESET	Reset interrupts on an IPC channel after an error.

TAILORING RMS68K

The complete RMS68K package provided to the user is extensive, offering a wide selection of functions the system designer can incorporate into the application system. For systems not requiring the full set of RMS68K capabilities, the package can be tailored to fit the needs of the application.

A tailored RMS68K load module is built in four steps:

- Select the set of directives required to provide the desired functionality.
- Modify source modules, as required.
- Assemble source modules.
- Perform an RMSGEN to link the modules and produce the final load module (chain files are provided in the RMS68K package to simplify this step).

VERSAdos provides all of the tools required to create a tailored RMS68K load module.

INPUT/OUTPUT AND FILE MANAGEMENT SERVICES

INPUT/OUTPUT

The VERSAdos input/output system is an abstract I/O scheme where all files and devices are treated as files. It is essentially a device independent system. Most I/O operations refer only to logical properties (e.g., the next record) rather than to particular device characteristics or file formats. Physical I/O is performed by routines that typically are not directly called by a user.

To facilitate control of the sources and targets for I/O, the system makes use of a software feature called a logical unit assignment. An assignment is like a numbered channel in that it controls the flow of data between program accessible storage and devices/files. An assignment must be made to a file before any I/O operations can occur. The assignment specifies a logical unit number (LUN) and target (file) and also the type of connection that is required (e.g., exclusive read and write).

The type of assignment that is made to a particular file will govern which I/O calls are allowed. For instance, to RENAME a file would require an exclusive read-write (EREW) assignment.

All I/O is separated into two categories, Input/Output Services (IOS) and File Handling Services (FHS). The IOS calls are executed via a Trap 2 instruction; the FHS calls are executed via a Trap 3 instruction. This is the server capability which provides user customized expansion of the Operating System, and which the I/O module utilizes.

FILE HANDLING SERVICES

*ALLOCATE — Create a file on a random access device
*ASSIGN — Establish a logical connection
*CHANGE AP — Change access permission
*RENAME — Change a file ID
*PROTECT — Change access privileges
*DELETE — Delete a file
*CHECKPOINT — Update a file on disk
*CLOSE — Dissolve an assignment
*RETRIEVE — Get device/file attributes
*FETCH DIRECTORY — Get directory entries
*FETCH DEVICE MNEMONIC — Get device/volume names
*CHANGE — Switch LU assignment
*FETCH DEFAULT VOLUME — Retrieve system or user default volume

INPUT/OUTPUT SERVICES (IOS)

*INPUT — Read from a file/device
*OUTPUT — Write to a file/device
*OUTPUT/INPUT — Write to, then read from a device
*UPDATE — Update a record
*DELETE — Delete a record
*FORMAT — Format a disk/sector
*POSITION — Position to a particular record of a file
*REWIND — Position at beginning of file
*HALT I/O — Terminate outstanding I/O
*TEST I/O — Test I/O completion
*WAIT — Wait only
*REQUEST BREAK SERVICE — Request an attention event for a break condition
*NEGATE BREAK SERVICE — Cancel break service
*CONFIGURE DEVICE — Dynamically configure I/O device
*CHANGE DEFAULT CONFIGURATION — Change current I/O device configuration parameters
*REQUEST DEVICE STATUS — Obtain current status of I/O device.

As pointed out above, the file handling services module would provide the logical link to a file via a logical unit number. The FHS module is also called whenever a logical link is changed or dissolved. Also, if the target is a true disk file, the FHS module is called whenever the target attributes need to be changed (i.e., change access permission). In addition, the FHS module is called in order to create a true disk file.

The input/output services module is called to handle all data transfers (i.e., input, output). The IOS module needs only to know a task ID and logical unit number to resolve the target.

- WAIT ONLY places the task into I/O wait until the completion of a previous I/O Proceed request to the specified Logical Unit (LU). If a task does not have any outstanding I/O to the specified LU, control is immediately returned. Invalid LU is the only error status this call returns.
- HALT I/O cancels an "I/O and Proceed" request previously issued, which is useful on an interactive terminal device. If Halt I/O is not used, an outstanding read request must be satisfied before any other I/O can be started on a device. The printer and CRT devices support the Halt I/O request.
- CONNECTION WAIT I/O is used when a task does a wait for the requested operation. The system coordinates I/O requests so that only one can access any device or file at a time.
- PROCEED/WAIT I/O causes control to be returned to the task after initiating an I/O request, so that the task may concurrently execute with the data transfer. The status is not set until completion of the I/O except for invalid function and invalid LU, which are requested before transfer initiation.
- FORMATTED/IMAGE SUPPORT FOR TERMINAL ATTRIBUTES
  - — Read Formatted: The data read is masked to 7-bit ASCII. Data is read until a carriage return is found or until the input buffer is full. Upon termination, a carriage return-line feed sequence is sent to the screen. An option to suppress character echoing can be selected.
  - — Write Formatted: Data is output until the buffer is exhausted or until a carriage return is found in the data stream. A line feed is automatically appended to the detected carriage return; if no carriage return was detected, an LF/CT sequence is output. If a CTL-W is typed during a write operation, the display is paused. The BREAK key can be pressed at any time during a Read or Write ASCII to abort the current processing.
  - — Read or Write Image: No formatting actions occur. The user must include all carriage return and line feed operations explicitly.

FILE ACCESS

There are three file types: Contiguous, sequential and indexed sequential. Since contiguous files are not dynamically expandable, they should be allocated with the maximum amount of space they will require. Records are 256 bytes in length and the user determines their content. Access may be either sequential or random. Typically they are used to store memory loadable modules.

Sequential and indexed sequential files are conceptually a logically contiguous block of data, but are not necessarily physically contiguous. Space may be allocated on the disk to one or more groups of contiguous sectors called a segment, thus allowing dynamic allocation and de-allocation of space without any movement of the data contained in the file or in other files. Except for contiguous files, each file has a table called the File Access Block describing the segments allocated to that file. Thus, although a file may be segmented, it is treated as a logically continuous collection of sectors.

Sequential files can contain fixed or variable length records (up to 65,535 bytes) and may be accessed sequentially or randomly. In addition, block and record access is allowed. For sequential access, three modes are available: Next, current and prior. Random and sequential access may be intermixed without closing and reassigning a file. For random access, the user supplies the position within the file of the record to be accessed. The user may access a record on a random basis and then perform a series of sequential accesses. With a sequential file, the random update call is limited to replacing a record of the same length and the random write call to appending one record to the end of the file. Indexed sequential files contain variable length records and support all the functions described for sequential files. The user may access indexed sequential files by logical record number or by key value.

FILE AND DEVICE PROTECTION

Files and devices have fixed and active protection. With fixed, each file or device has associated with it two access protection codes — one for read access and one for write. The file or device cannot be assigned for read or write access unless the operator or requesting task supplies the code(s) matching the value of the protect codes. The owner of a file need not supply the read protect code. The protection codes of a file are defined when the file is allocated and may be changed only by a task having exclusive access. The access protection codes of a device are defined at INITIALIZATION time, and only the system administrator may change them.

Active protection is only in effect while the file remains assigned. During this period, a task may prevent other tasks from accessing that file during use. For this reason, the user may ask for exclusive access permission (for either read or write) at assignment time. A file cannot be assigned with an access permission that is incompatible with an existing assignment for that file. When a volume is write protected, only read assignments are accepted.

PROGRAM LOADER

The Program Loader is a TRAP #4 function and is used to:

(1) Create a new task
(2) Allocate memory segments for the task based on segment information found in the Loader Information Block (LIB) of a load file created by the linkage Editor
(3) Read the contents of each segment from the load file into the segments allocated

In a system that includes a Memory Management Unit (MMU), each memory segment loaded is assigned the logical addresses defined in the LIB. The MMU will convert logical addresses to physical addresses.

In a system lacking an MMU, the Program Loader will first determine whether or not the task being loaded is relocatable. A task that includes a segment with a logical address of zero is assumed to be a relocatable task.

A relocatable task is loaded into any available memory and its segment addresses and entry point are changed to match the actual physical address at which it is loaded.

To load a non-relocatable task, the Program Loader will use the segment addresses defined in the LIB as physical addresses. The task will be loaded only if memory is available at the physical addresses defined.

COMMAND PROCESSOR

The interface between user and VERSAdos is the command processor which accepts and interprets user commands. Several of this processor's system functions are available to all users, including: time-of-day, date, batch capability, file execution chaining and program loading.

Execution of the Command Processor is initiated after the system boot by the Executive. The Command Processor declares its program segment shareable, requests all unclaimed breaks, and allocates an asynchronous service queue (ASQ), then waits for an event to occur. Commands may be received from direct keyboard input or from a command file.

Upon receipt of an event, the Command Processor will then create a task which shares the Command Processor program segment but maintains its own data segment. When a command is received, the Command processor searches through its command list table for a match. If found, execution of the command will continue at the entry point provided by the command table. If no match is found, it is assumed that the input is the file name of an executable task on the default system volume. The Command Processor will direct the system loader to load into memory the target task. If the load is successful, a START is issued to begin execution of the task. The Command Processor will be notified upon completion of the target task whether a normal or abnormal completion occurred.

DEVICE DRIVERS

VERSAdos device drivers execute in supervisor mode as an extension of RMS68K and are called by the input/output services (IOS) and file handling services (FHS) to accomplish the physical I/O requested by a user. A driver is generally transparent to the user. Each driver contains the entire description of its corresponding device's physical attributes thus freeing IOS and FHS of physical I/O burden.

Since a device driver is easily incorporated during system generation, custom drivers may be added with no undesirable effect on the system. VERSAdos contains device drivers for all current VMEmodules and VERSAmodules. All drivers include both object and source code. A manual is also supplied detailing how to write a custom driver to run under VERSAdos.

UTILITIES

Although the primary purpose of VERSAdos is to provide real-time, multitasking support for VMEmodule, VERSAmodule and other MC68000-based systems, VERSAdos also contains a large complement of utilities to provide the user with maximum use of the system. Utility categories include: file manipulation, debug, analysis, disk management and target system utilization.

The capability of creating and managing files is provided the user through six utilities. These include a full screen text editor plus the Delete, Dump, Copy, List, Rename and Patch utilities.

Several utilities are available that make it possible to analyze the system while a program is running as well as analyze task and system level crashes. The utilities are: Dumpanal, Prtdmp, and Sysanal.

The capabilities of formatting, initializing, backing up and repairing data on disks are provided by the utilities: Init, Backup and Repair.

As an aid to users creating modules of executable code, VERSAdos contains a symbolic debugger, a linkage editor, structured macro assemblers for the MC68000 family, a spooler and other tools which facilitate connecting and downloading to a system based on any of the microprocessors in this family. VERSAdos also supports a prom programmer, a cross assembler for the MC6800 family and the high level languages: Pascal, C and Fortran.

CONFIGURATIONS

VERSAdos contains bootable, executable load modules for the following VMEmodules and VERSAmodules: MVME101, MVME110-1, MVME117, MVME115M, MVME120/121, MVME122/123, MVME130, MVME131, MVME133, VERSAmodules 03 and 04 and all VMEsystem 1000 Microcomputers. VERSAdos also contains bootable, executable load modules for the EXORmacs Development System and the VME/10 Microcomputer System. Any of these pre-configured load modules may be customized by the user to match his unique system requirements through use of the default system generation command file included in VERSAdos for this purpose. A manual is supplied which, when configuration is required, describes the procedure to be used. Most systems will not require a reconfiguration.

M68K0VDOS

ORDERING INFORMATION

VERSAdos, the real-time, multitasking, multiuser disk operating system for MC68000 applications designed around VMEmodules or VERSAmodules and other host and peripheral resources including memory, I/O, mass storage and timer functions. VERSAdos includes utilities, a system generation facility and has peripheral device drivers for the following:

VME/10	MVME117 Series	MVME315	MVME435A	MVME625	M68KVM21
M68KMACS	MVME120 Series	MVME320A	MVME600/601	M68RWIN1	M68KVM22
MVME050	MVME130 Series	MVME400	MVME605	M68KVM03	M68KVM30-1
MVME101	MVME133	MVME410	MVME610	M68KVM04	M68RAD1
MVME110-1	MVME300	MVME420	MVME615/616	M68KVM20	M68RI01

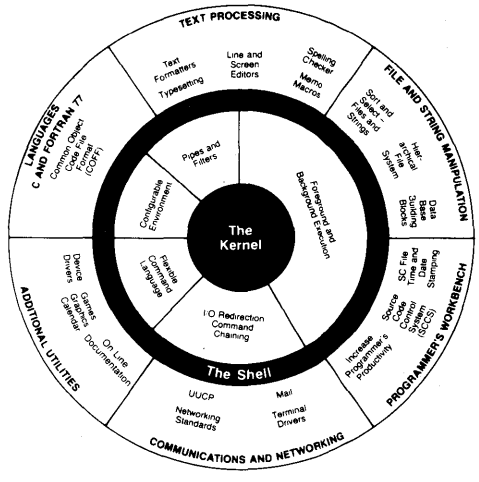
Part Number	Description
M68K0VDOST	VERSAdos source on 14" CMD cartridge, VM21 format. Requires 96Mb CMD drive.
M68K0VDOSH	VERSAdos object on 14" CMD cartridge, VM21 format. Bootable on EXORmacs.
M68K0VDOS	VERSAdos object on 8" Motorola format diskettes. Bootable on EXORmacs, VM03, VM04, and VME110-1. (Not compatible with MVME320A and MVME420 Disk Controllers.)
M68VKXBVERDOS	VERSAdos object on 5-1/4" diskettes. Bootable on VM02, VM03, VM04, VME101, VME110-1, VME115M, VME120 series, MVME130 Series, MVME133, VME/10 and the VMEsystem 1000 V Series.
M68VKXSVRDOS	VERSAdos source on 5-1/4" diskettes. Requires 40Mb Winchester drive.

RELATED DOCUMENTATION

Part Number	Description
M68KVOVER	VERSAdos Overview.
M68KVSF	M68000 Family VERSAdos Systems Facilities Reference Manual.
M68KVMSG	VERSAdos Messages Reference Manual.
M68KMASM	M68000 Family Resident Structured Assembler Reference Manual.
M68LINK	M68000 Family Linkage Editor User's Manual.
M68KEDIT	M68000 CRT Text Editor User's Manual.
M68KSYSGEN	System Generation Facility User's Manual.
M68KRMS68K	MC68000 Real-Time Multitasking Software User's Guide.
MVMEVDOS	VERSAdos to VME Hardware and Software Configuration User's Manual.
RMS68KI0	VERSAdos I/O and File Management Services Manual.
M68KDRVGD	Guide to Writing Device Drivers for VERSAdos.
M68KSYMBG	SYMbug and DEbug Monitor User's Manual.

SYSTEM V/68 Operating System Software Features

- Compliant with AT&T's UNIX® System V Interface Definition Release 3
- Object Code Compatible with SYSTEM V/68 Release 2
- Transparent Sharing of Data and Peripheral Devices Across a Network
- Applications Can Be Run Independent of the Underlying Network
- Virtual Memory Provides Optimum Utilization of Physical Memory
- Fast File Access and Reliable File System
- Efficient Use of Disk Space and Memory Through Shared Libraries
- Supports MC68881 Floating Point Coprocessor



GENERAL DESCRIPTION

The SYSTEM V/68 Release 3 Operating System is the standard operating system for the M68000 family of microprocessors. It affords users a powerful computing environment for developing, executing and supporting technical, commercial and network applications.

Compliant with AT&T's UNIX System V Interface Definition (SVID) Release 3, SYSTEM V/68 Release 3 provides users with the ability to share data, applications and resources among different computers transparently. Release 3 is object code compatible with SYSTEM V/68 Release 2. It offers an efficient kernel which provides process scheduling and I/O facilities to all programs. In addition, a powerful command processor, the Shell, provides for interactive control. An extensive set of tools and utilities provide support for program development, text processing, electronic mail and system to system communication.

OVERVIEW

THE KERNEL

The kernel supports a multitasking multiuser software development and OEM application execution environment. As the heart of the system, the kernel schedules tasks and manages system resources. It acts as a dispatcher of tasks competing for the system resources and as a traffic coordinator for the data flowing through the system while managing and allocating memory as required.

VIRTUAL MEMORY

The demand paged, virtual memory feature of SYSTEM V/68 Release 3 has several advantages over swapping:

- Programs larger than physical memory can be executed.
- Only the portions of a program which are actually being used reside in memory at any particular time.
- External fragmentation is completely eliminated.
- 32 Megabyte process space.
- Two level page table.

BERKELEY SIGNALS

A new set of system calls is provided in Release 3 of the SYSTEM V/68 Operating System which resemble those of the Berkeley 4.2 BSD signal interface. The new implementation provides enhancements to the existing SYSTEM V/68 interprocess signalling mechanism while preserving the SYSTEM V/68 signal semantics.

PROCESS FORK MECHANISM

This feature allows processes to "fork" without duplicating the data area of the parent. It is commonly referred to as "copy-on-write" and essentially allows the child process to share the parent's data area. Modified pages are duplicated for each process before the access takes place. This feature provides the performance improvement of the Berkeley 4.2 BSD vfork mechanism without affecting compatibility with the SYSTEM V/68 fork system call interface.

THE SHELL

The shell interprets user commands and calls system tasks to perform the requested operations. It accepts commands one at a time or in a series called a pipe, where output from one job becomes input to another job.

Features of the shell include: control-flow primitives, parameter passing, variables and string substitution. In addition, constructs such as *while, if-then-else, case* and *for* are available. Two-way communication is also possible between the shell and the commands. String-valued parameters, typically file names or flags may be passed to a command. A return code is set by commands that may be used to determine control-flow, and the standard output from a command may be used as shell input.

The shell can modify the environment in which the user command runs. Input and output can be redirected to files and processes that communicate through pipes can be invoked. Commands are found by searching directories in the file system in a sequence that can be defined by the user. Commands can be read either from the workstation or from a file, which allows command procedures to be stored for later use.

THE SYSTEM V/68 FILE SYSTEM

The SYSTEM V/68 file system comprises a uniform set of files and directories organized as a hierarchical, tree-like file structure of arbitrary size. Branching from the beginning of the file system, also known as the root directory, are several system directories, including the user directory. A directory can contain any number of subdirectories and files, which allows for almost any kind of branching structure that is required for the filing system. Major features of the file system are:

- 1024 byte file block size.
- Simple and coherent naming conventions.
- File linking across directory boundaries.
- Automatic file space allocation and de-allocation.
- Mountable file systems.
- Hashed 1-node file and directory look up.
- Device-independence: each physical I/O device is treated like a file, thus allowing for uniform file and device I/O.
- Flexible directory and file protection modes: allows for the setting of "read," "write" and "execute" authority separately for the owner, for a group of users and for all other users. These protection modes are altered dynamically.
- Facilities are provided for creating, moving, accessing and processing groups of or single files and directories.

Optimal block allocation and de-allocation provides for fast access of files where data resides in the disk cache. Additionally, a configurable feature is included in Release 3 which causes all writes to the file system to be synchronized with the disk. This configurable feature reduces the likelihood of lost user data in the event of a system crash and improves the robustness and reliability of the file system.

REMOTE FILE SHARING

The Remote File Sharing (RFS) feature allows users to share files, data and peripheral devices, such as tape drivers and printers, transparently across different computers on a network.

Existing applications can be installed and executed in an RFS environment without any modifications and these programs can access remote files. The full file system semantics, including Named Pipes and File and Record Locking, are preserved when accessing remote resources. File and Record Locking preserves data integrity by protecting files from simultaneous access by two or more users.

Extensive administrative tools allow each computer on the network to control the specific resources it makes available for sharing as well as the specific remote resources it makes available to local users. Security features are provided to enable a local system to accept or deny access to local resources on a per remote system and per remote user basis.

STREAMS

The Streams mechanism provides network protocol and media independence by allowing applications software to be independent of the underlying network. A change in the transmission medium or protocol can be accommodated by substituting Streams modules without modifying the applications software. Streams modules can be combined to perform sophisticated network services and the same modules can be used over different media and in different network architectures.

In addition to modularity, Streams provides the following facilities:

- Simple User Interface
- Buffer Management
- Flow Control
- Scheduling of Streams Protocol Modules
- Efficient Message Passing
- Multiplexing
- Error and Trace Loggers

The Streams mechanism is implemented as an independent subsystem which allows protocols implemented using Streams primitives to be easily ported or migrated. This allows protocols to be implemented in a way that is independent of the underlying machine architecture.

TRANSPORT LEVEL INTERFACE LIBRARY

The Transport Level Interface (TLI) along with the Transport Provider Interface (TPI) provide a specific mechanism that allows a user to run applications independent of the underlying network. An application written using the TLI Library will work without modification over any network implemented according to the TPI specification.

The traditional *uucp* commands have been modified to use the TLI Library. The changes make the uucp family independent of the underlying network and usable across all Streams-based transport providers. This change eliminates the need for different file transfer commands for different networks.

COMMUNICATIONS AND NETWORK SUPPORT

The SYSTEM V/68 Operating System provides support for electronic mail, communications and networking.

Electronic mail allows users to communicate with one another using the system as a mail box or as a bulletin board. The communications utilities allow a SYSTEM V/68 user to communicate to mainframe computers. The tools and utilities supported under SYSTEM V/68 Release 3 include:

- uucp, UNIX-to-UNIX copy program, is a utility used for communication between UNIX systems.
- cu is a utility used to call up another UNIX system.
- mail sends mail and reads mail sent by users to other users within a system.
- interprocess functionality, including shared memory, messages and semaphores.

The data communications products supported under SYSTEM V/68 Release 3 include:

- 3274 SNA and BSC
- 3776 SNA
- RJE/HASP, 2780 and 3780
- X.25 with TCP/IP

Networking support allows several computers to be linked together, either through dedicated links or by dial-up connections. Ethernet Local Area Networking (LAN) support with XNS or TCP/IP is provided by SYSTEM V/68. So that the software can be easily upgraded to other protocols, each module corresponds to a specific layer of the OSI interconnection model.

SOFTWARE DEVELOPMENT TOOLS

Languages

Provided as an integral part of SYSTEM V/68 are the C language compiler, "cc" and the High Level C Code Optimizer. The C language has established itself as one of the most popular and widely used commercially supported programming languages, and produces portable application software. The optimizer is an added phase for the compilers derived from the MC68000 Software Generation System (SGS). It aids the code generator to produce code which results in 10% to 20% performance improvements.

The SYSTEM V/68 Operating System also provides support for the FORTRAN language by including a FORTRAN 77 compiler "f77," a rational (structured) FORTRAN preprocessor "ratfor," and an extended FORTRAN language preprocessor "ef1."

Floating point support is also provided for both the C and FORTRAN 77 compilers. The MC68881 Coprocessor provides the following features:

- Eight general purpose floating data registers, each supporting a full 80-bit extended-precision real data format.
- A 67-bit Arithmetic Logical Unit to allow very fast calculations.
- A 67-bit barrel shifter for high-speed shifting operations.
- Fully conforms to IEEE P754 Standard (draft 10.0) plus a full set of trigonometric and logarithmic functions.
- Overlapped instruction execution enhances throughput while maintaining the programmer's model of sequential instruction execution.

An MC68000 assembler, "as," and a linker/loader, "ld," complements the C and FORTRAN language compilers.

Specifically designed to make software development easier, the SYSTEM V/68 programming environment also provides a set of software tools to aid in efficient and professional tuning and maintenance of applications software. These tools include the following:

- *cflow* is a tool that generates a C flow graph by analyzing a collection of C, yacc, lex, assembler and object files and builds a graph charting the external references.
- *sdb* is a symbolic debugger.
- *prof* is a code profiler that provides, for external symbols, the percentage of time spent between a given symbol and the next, together with the number of times that functions are called.
- *size* is a command that provides section size information for the text, data and bss (uninitialized data) sections of the common object files. The total size for the module is also provided.
- *nm* is a command that prints the symbol table of each common object file filename.
- *ar* is the archive and library maintainer for portable archives.
- *make* is a tool to maintain, update and regenerate programs; make only updates a module if it depends on modules that are newer than it.
- *dump* is a command that provides a dump of selected parts of an object file.
- *lint* is a checker for C programs. It attempts to detect features of C programs that are likely to be bugs, nonportable or wasteful.
- *cb* is a C program beautifier that produces code with spacing and indentation that displays the structure of the code. cb "canonicalizes" the code to the style of Kernighan and Richie.
- *cxref* is a tool that analyzes a collection of C files and builds a cross reference table.

Programmer's Workbench

The Programmer's Workbench is a collection of tools that support the development of large systems of software in a professional and automatic manner. In particular, this includes Source Code Control System (SCCS) which provides the facility to store, update and retrieve all versions, past and present of source code modules.

SHARED LIBRARIES

Shared Libraries is a feature that saves significant disk space and memory. Functions in a shared library are stored only once on disk and once in memory, and are shared by all executable files (on disk) and all processes (in memory) using them. This feature saves space and eases incorporating corrections or enhancements in the shared library functions. Updated functions in a shared library are automatically applied to binary files accessing the library.

TEXT PROCESSING

Support for text processing under the SYSTEM V/68 Operating System is provided by the inclusion of four editors: vi, ed, ex and sed. Ex and vi are full screen editors, ed is a line editor and sed is a non-interactive stream oriented editor.

There are two text formatting utilities available: "nroff" for printing on typewriter-like devices and "troff" for phototypesetting. Tools also exist for table formatting, mathematical equation setting and spelling checking.

The vi and ex editors support a large number of existing terminal types through the use of the terminal database "terminfo." Additional entries to terminfo can be added by the user to support other terminals.

SYSTEM MAINTENANCE

Ram-Based Logical Disk allows Release 3 to be booted from tape and run with a memory-based file system in order to repair a disk or install software on an empty disk. This enhancement allows use of selected system utilities without the use of a hard disk while preserving the standard SYSTEM V/68 user interface. This capability is intended to facilitate system maintenance and installation.

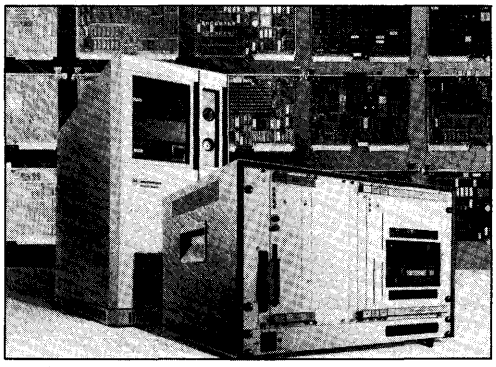
The SYSTEM V/68 Release 3 Operating System also provides the capability to detect power fail conditions when attached to an Uninterruptable Power Supply (UPS). This enhancement causes the system to shut down gracefully when an interrupt is detected.

BACKUP AND RESTORE FACILITY

The Backup and Restore Facility is enhanced with a menu interface. This facility provides attended full and incremental backups, scheduling of unattended incremental backups, selective backup/restore of directories, files and sub-trees as well as selective backup/restore of files owned by user.

MENU INTERFACE

A menu-based interface package is provided in SYSTEM V/68 Release 3. This interface, designed to simplify system administration functions, also serves as a helpful front end for interfacing to SYSTEM V/68 utility programs. This facility allows command lines to be built interactively and teaches the user command options. As such, this feature is useful for the novice user and the experienced user can utilize this facility to create and modify menus as well.



Bus Technical Summaries

NOTICE

These bus summaries are provided only as a general aid to designers. Since VMEbus, VMSbus and VSB are still evolving, a copy of the latest version should be obtained to insure use of accurate figures for an actual design.

VMEbus Specification Summary

Originally introduced in 1981, VMEbus has become the standard high performance bus structure through several years of intensive design activity. It is the bus of choice for the present and future of microcomputer architecture, supporting data transfer rates as high as 24 Mbytes in the expanded 32-bit configuration.

MEbus has a master/slave asynchronous non-multiplexed data transfer structure, seven levels of priority interrupt, four levels of data bus arbitration and rapid fault detection and control for bus, system and AC failures.

One of the many features of the 32-bit configured VMEbus system is that the bus dynamically senses whether 8-, 16-, or 32-bit data paths are needed and adjusts automatically.

VMEbus specifications were originally developed jointly by Motorola, Mostek, and Signetics/Philips and have been accepted by close to 100 manufacturers worldwide.

In integrating 8-, 16-, and 32-bit system components VMEbus is innovative, publicly documented, and eminently adaptable to new technologies. LSI/VLSI are providing interface and peripheral chip functions that vastly increase the functionality/cost ratio of VMEbus modules.

Based on the most popular Eurocard formats with DIN pin and socket connectors, VMEbus is presently being formally standardized by both the IEEE (P1014) and the IEC standards organizations.

A detailed VMEbus Specification Manual may be ordered from the Motorola Literature Distribution Center (Part Number MVMEBS/D1*).

BASIC VMEbus STRUCTURE

The VMEbus interface system consists of four groups of signal lines called "buses," and a collection of "functional modules" which can be configured as required to interface devices to the buses. The functional modules communicate with one another by means of bus signal lines provided by a backplane.

The interface functions of the VMEbus have been divided into four categories. Each functional category consists of a bus and associated functional modules which work together to perform specific duties within the system interface. Figure 1 illustrates the individual functional modules and buses contained within the VMEbus definition, and each category is briefly summarized below.

Data Transfer — Devices transfer data over the Data Transfer Bus (DTB) which contains the data and address pathways and associated control signals. Functional modules called "DTB MASTERS" and "DTB SLAVES" use the DTB to transfer data between each other.

DTB Arbitration — Since the VMEbus system may be configured with more than one DTB MASTER, a means must be provided to transfer control of the DTB between MASTERS in an orderly manner and to guarantee that only one MASTER controls the DTB at a given time. Bus arbitration is the area of the VMEbus specification which defines the signals (Arbitration Bus) and modules (DTB REQUESTERS and DTB ARBITER) to perform the control transfer.

Priority Interrupt — The priority interrupt capability of the VMEbus provides a means by which devices can request interruption of normal bus activity and can be serviced by an interrupt handler. These interrupt requests can be prioritized into a maximum of seven levels. The associated functional modules are called INTERRUPTERS and INTERRUPT HANDLERS, which use signal lines called the Interrupt Bus.

Utilities — The system clock, initialization, and failure detection have been grouped into the category of utilities. The utility bus includes a clock line, a system reset line, a system fail line, and an AC fail line.

*Revision D1 was current at the time of printing of this document.

NOTICE

These bus summaries are provided only as a general aid to designers. Since VMEbus, VMSbus and VMXbus are still evolving, a copy of the latest version should be obtained to insure use of accurate figures for an actual design.

SPECIFICATION TERMINOLOGY

In some bus specifications, the protocol is treated on an abstract level. For example, it might be said that Device A "sends a message" to Device B. While this does allow a protocol to be defined in an application independent manner, the VMEbus specification is more closely related to the physical implementation. It describes the protocol in terms of levels and transitions on bus lines.

Signal Line States

A signal line is always assumed to be in one of two ***levels*** or in ***transition*** between these levels. Whenever the term ***"high"*** is used, it refers to a high TTL voltage level ($\geq$ + 2.0 V). The term ***"low"*** refers to a low TTL voltage level ($\leq$ + 0.8 V). A signal line is "in transition" when its voltage is moving between + 0.8 V and + 2.0 V.

There are two possible transitions which can appear on a signal line, and these will be referred to as ***"edges."*** A ***rising edge*** is defined as the time period during which a signal line makes its transition from a low level to a high level. The ***falling edge*** is defined as the time period during which a signal line makes its transition from a high level to a low level.

Use of Asterisk (*)

To help define usage, signal mnemonics have an asterisk suffix where required:

- An asterisk (*) following the signal name for signals which are ***level*** significant denotes that the signal is true or valid when the signal is low.
- An asterisk (*) following the signal name for signals which are ***edge*** significant denotes that the actions initiated by that signal occur on a high to low transition.

Note: The asterisk is inappropriate for the asynchronously running clock line SYSCLK. There is no fixed phase relationship between this clock line and other VMEbus timing.

PROTOCOL SPECIFICATION

Each VMEbus functional area — such as data transfer, bus arbitration, and priority interrupt — is defined via protocol specifications. Functional modules are defined for each area (Figure 1), and a protocol is defined for each module. The protocol is a set of rules governing the interaction of the module with the VMEbus. A functional module communicates with another module by driving/receiving bus signals. The protocol governs these communications by determining:

— when a module may drive and change the level of bus signals, and

— when and how a module must respond to a bus signal.

Bus signals can be generally discussed in two classifications:

- Interlocked Bus Signals
- Broadcast Bus Signals

Interlocked Bus Signals

An interlocked bus signal is sent from a specific module to another specific module. The signal must be acknowledged by the receiving module. An interlocked relationship exists between the two modules until the signal is acknowledged. For example, an interrupt REQUESTER can send a signal asking for an interrupt. That signal must be answered at some time with an interrupt acknowledge signal (no time limit is prescribed by the VMEbus specification).

Interlocked bus signals are dedicated to coordinating internal functions of the VMEbus system, as opposed to interacting with external stimuli. Each interlocked signal has an internal source module and destination module. Also, these signals have timing specifications associated with them to assure proper bus operation.

Of significant importance are the interlocked bus signals used to coordinate transfer of addresses and data. Addresses and data cannot be considered "signals" in the strictest sense because they are not "sent" from one device to another. Instead, they are "placed" on a bus, while a separate bus signal (called a strobe) is sent to indicate their presence on the bus. The actual addresses or data have no effect on the protocol — that is, the specific address or data on the bus does not affect the strobe; however, the timing sequence (i.e., set-up time) between the specific address or data being placed on the bus and the sending of the accompanying strobe signal ***is*** important. Whenever this relationship is important, it is emphasized in the protocol definition.

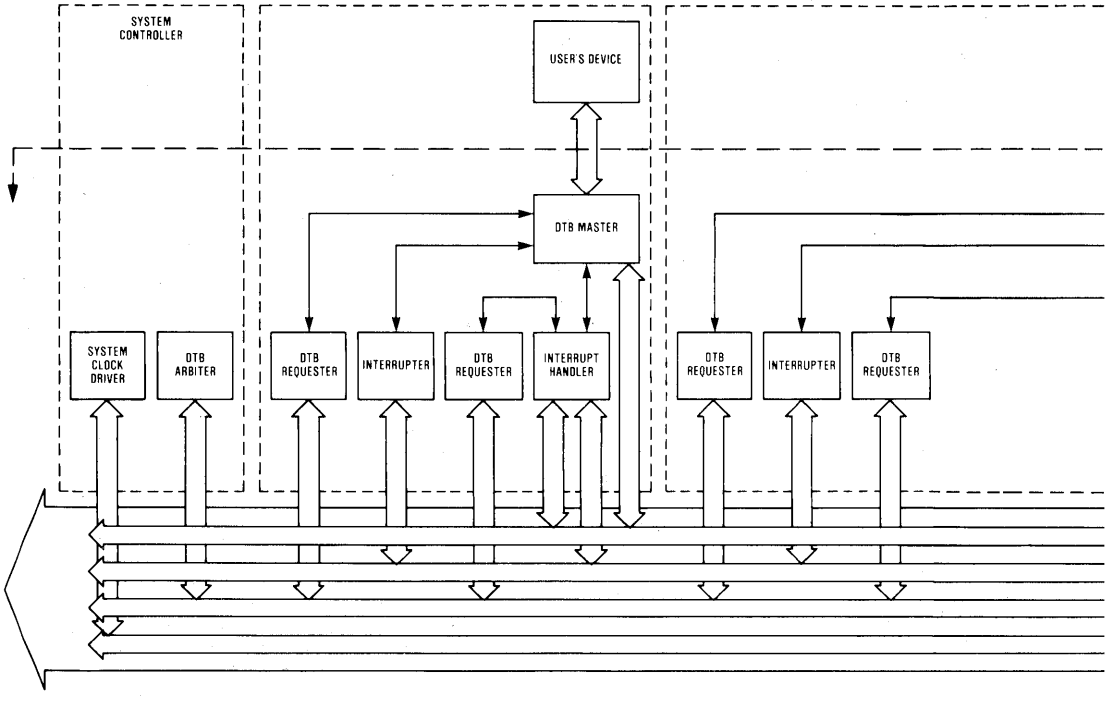
An example of a pair of interlocked signals is DS0* (or DS1*) and DTACK*, which provide interlocking between an addressed SLAVE and the active MASTER.

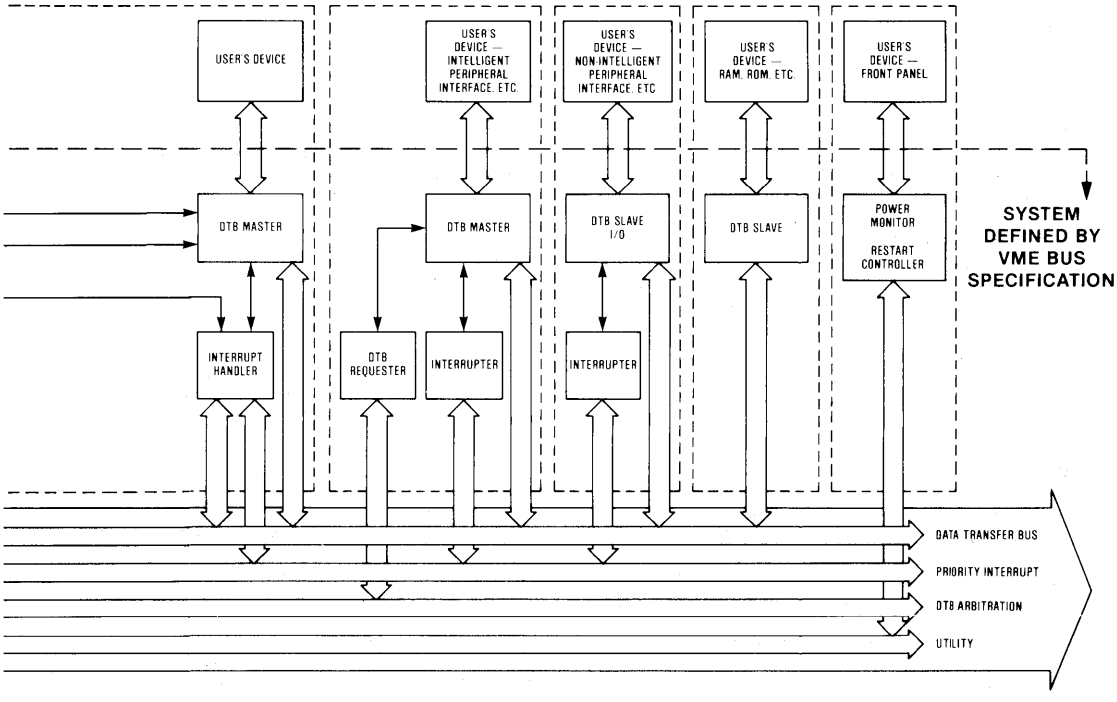
Broadcast Bus Signal

A broadcast bus signal can be placed on the bus by a module in the system in response to an external event. There is no prescribed protocol for acknowledging a broadcast signal. Instead, the broadcast is maintained for a minimum specified time period long enough to assure that all appropriate modules will detect the signal. Broadcast signals may also be monitored from outside the system to gain information about system status. The broadcast signal can be given at any time, irrespective of any other activity taking place on the bus.

Since the broadcast signal has no interlocked relationship with other bus signals, a dedicated line must be provided for each broadcast signal type. These lines are used for functions

FIGURE 1 — Functional Modules and Buses Contained Within the VMEbus Definition




USER'S DEVICE
USER'S DEVICE — INTELLIGENT PERIPHERAL INTERFACE, ETC.
USER'S DEVICE — NON-INTELLIGENT PERIPHERAL INTERFACE, ETC
USER'S DEVICE — RAM, ROM, ETC
USER'S DEVICE — FRONT PANEL
DTB MASTER
DTB MASTER
DTB SLAVE I/O
DTB SLAVE
POWER MONITOR
RESTART CONTROLLER
SYSTEM DEFINED BY VME BUS SPECIFICATION
INTERRUPT HANDLER
DTB REQUESTER
INTERRUPTER
INTERRUPTER
DATA TRANSFER BUS
PRIORITY INTERRUPT
DTB ARBITRATION
UTILITY

such as system reset and power failure sequencing. These activities also differ from interlocked signals because the modules that generate broadcast signals do not address another specific module, but announce special conditions to all modules.

VMEbus DATA TRANSFER

Introduction

The VMEbus contains a high speed asynchronous parallel Data Transfer Bus (DTB). The DTB is used by a processor or Direct Memory Access (DMA) device to select the desired peripheral or memory location and to transfer data to or from that location. The DTB can be logically subdivided into address, data, and control line groups. The number of lines in each of these groups varies with the particular VMEbus options selected by the user. There are four sets of DTB related options:

D8, D16, or D32	Data path width
A16, A24, or A32	Address path width
BTO(n)	Bus Time Out
SEQ	Sequential Access

DTB Options — Basic Description

An option D8 SLAVE can read or write only eight bits at a time on D00–D07. An option D8 MASTER must be capable of reading or writing on either the D00–D07 or D08–D15 data lines, but only eight in any one transfer.

Option D16 specifies that all data transfer activities supported by the module will be restricted to eight or sixteen bits. Option D32 specifies that the module (MASTER or SLAVE) will be capable of doing LONGWORD (32-bit) data transfers. Option D32 also requires an expanded bus system.

An option A16 MASTER can place only short address modifier (AM) codes and 15-bit addresses on the bus. An option A16 SLAVE will decode only address lines A01–A15, and it will respond only when a short address AM code is presented.

Option A24 specifies that all addresses generated by the MASTER or decoded by the SLAVE will be restricted to either 15-, or 23-bits depending upon the AM code presented.

Option A32 allows the use of 31-bit addressing also. In this case the address modifier lines indicate to SLAVES whether the address is 15-, 23-, or 31-bits. (Option A32 also requires an expanded bus system.)

For a MASTER which generates its own bus time-out, or for a bus time-out module, option BTO (n) specifies that the module will abort a data transfer cycle after "n" microseconds if no response is received from a SLAVE. This protects against bus lockups caused by an invalid address or a malfunctioning SLAVE.

An option SEQ MASTER can request a sequential access transfer. An option SEQ SLAVE will respond to a sequential access transfer.

DTB Operation

Each data transfer on the DTB occurs between a functional DTB MASTER and a functional DTB SLAVE. The data transfer is initiated by the DTB MASTER. The addressed SLAVE must then acknowledge the transfer. The asynchronous nature of the DTB allows the SLAVE to control the amount of time taken for the transfer. After receiving the transfer acknowledge, the DTB MASTER terminates the data transfer cycle.

Data Transfer Bus Lines

Depending on the options chosen, the MASTER must drive the following lines:

15, 23, or 31	address lines (A01 through A15 if option A16) (A01 through A23 if option A24) (A01 through A31 if option A32)
6	address modifier lines (no change with any option selected)

Address Lines

The smallest addressable unit of storage is capable of storing eight bits of binary data. Each 8-bit group is called a "byte," and the location in which the byte is stored is called a "byte location." Two consecutive byte locations (even byte address and the next higher sequential odd byte address) are called a "word location." A MASTER accesses a byte or word location by placing its binary "word address" on the address bus. The address lines on the bus are numbered starting with A01 instead of A00 to emphasize the fact that byte location addressing is done with data strobe lines instead of an "A00" line.

Address Modifier Lines

The address modifier lines allow the MASTER to pass additional information to the SLAVE during data transfer. This information may be used in several ways.

System Partitioning

SLAVES in the system may be configured (either dynamically or statically) to respond to a single address modifer code. If there are several MASTERS on the VMEbus, each may be assigned a code to be used when accessing the SLAVES. This allows the system to be partitioned and reduces the likelihood that a single malfunctioning MASTER will take the whole system down.

Memory Map Selection

SLAVES may be designed to respond at different addresses, depending upon the address modifier received. This allows the MASTER using the bus to place the system resources in selected map locations (or eliminate them from the map) by providing different address modifier codes.

Privileged Access

Because SLAVES could be designed to respond to some address modifiers and not to others, it is possible to establish a large number of privilege levels. Each MASTER would provide an address modifier indicating its privilege level when accessing a SLAVE. If the SLAVE did not receive an appropriate AM code, it would not respond.

Cycle Type

The AM codes can be used to specify a special type of transfer cycle. The VMEbus specifies one special cycle type. The user could use additional codes to specify others. This special cycle type is a sequential access cycle. There are four sequential access (ascending access) AM codes, and when one of these is placed on the bus, the memory boards in the system latch the address into a counter and increment the counter after each odd-byte, word, or LONGWORD transfer. (Increment = 1 for odd-byte, 2 for word, and 4 for LONGWORD.)

Distributed Memory Management

Memory management logic is often used in systems to allocate and translate memory segments dynamically. A collection of these segments is assigned to each active task. Each time the real-time executive switches from one task to the next, it must either change the contents of the segment registers or select another set of segment registers (the latter approach being much faster).

Address modifier codes may be used as segment register selectors. In this case, the MASTER places AM codes on the bus which indicate to memory management logic on the slave boards which set of segment registers should be used.

Addressing Range

The VMEbus provides 31 address lines to allow direct addressing to over four billion bytes. For most SLAVES, however, the extra logic required to decode all 31 address lines is a needless expense. For this reason, the VMEbus defines three address ranges:

Short addressing	64K bytes
Standard addressing	16M bytes
Extended addressing	4G bytes

A group of address modifier codes is set aside for each type of addressing. SLAVES receiving a short address AM code ignore the upper 16 address lines (A16–A31). SLAVES receiving a standard address AM code ignore the upper eight lines (A24–A31). When receiving an extended address AM code, the SLAVE decodes all 31 address lines.

Slave boards which do not decode address lines A24–A31 should not respond to extended address AM codes. Slave boards which do not decode address lines A16–A31 should not respond to either extended or standard AM codes.

Data Transfer Lines

Depending on the options chosen and the type of data transfer, the source of the data (MASTER or SLAVE) must drive the following data related lines:

8, 16, or 32 data lines	(D00 through D07/D08 through D15 if option D8) (D00 through D15 if option D16) (D00 through D31 if option D32)

A word transfer requires the driving of data lines D00 through D15. On a LONGWORD transfer, data lines D00 through D31 are driven. Note that word transfers always use the lower 16 lines (i.e., D00 through D15), while byte transfers use different lines for odd and even bytes (i.e., D00 through D01 or D08 through D15).

Only LONGWORD transfers use lines D16 through D31. There are two word addresses for each LONGWORD. When LONGWORD transfers take place, the address presented is the even word address of the LONGWORD location (A01 is low). When accessing the data stored in a LONGWORD location on a word basis, the even word address (A01 low) corresponds to the LONGWORD data bits D16 through D31. Likewise, the structure of the bytes within a word have the even byte (selected by DS1*) as the most significant eight bits of the word.

Data Transfer Control Lines

The MASTER will drive the following lines:

AS*	Address strobe	(On all transfers)
DS0*	Odd data byte strobe	(Each is operation
DS1*	Even data byte strobe	dependent, but at least one must always be driven)
LWORD*	LONGWORD select	(Operation dependent)
WRITE*	Read/Write select)	(Operation dependent)

The SLAVE will always drive the following lines:

BERR*	Bus error	(If data transfer is not possible)
DTACK*	Data acknowledge to MASTER	(If data transfer was successful)

AS* is the address strobe. It informs all SLAVE modules that the address is now stable and may be clocked into holding registers.

DS0* and DS1* select the data to be transferred and, on a write transfer, strobe the transferred data. DS0* low means that the byte which would be addressed with A00 (if it existed) set high (the odd byte) is to be found on data lines D00 through D07. Likewise, DS1* low means that the byte which would be addressed with A00 set low (the even byte) is to be found on data lines D08 through D15. These two lines replace the function that A00 would perform but they also allow two bytes to be transferred simultaneously which would not be possible with just an A00 line. The sender (MASTER for a write cycle; SLAVE for a read cycle) is not prohibited from driving the data lines which are not being strobed. The receiver should ignore any and all levels and/or transitions which occur on non-strobed data lines.

LWORD* specifies that 32 bits will be transferred on data lines D00 through D31.

DTACK* is driven by the SLAVE to indicate that the data was successfully received on a write cycle. On a read cycle the SLAVE uses DTACK* to indicate that the data has been read from memory and has been placed on the data bus.

BERR* is driven by the SLAVE to indicate that data was not written on a write cycle or that it could not be retrieved on the read cycle.

IACK* is the interrupt acknowledge line. During data transfers, it may be driven high by MASTERS or will be pulled high by the bus terminators, appearing as a high level to all SLAVES. If IACK* is low, the cycle is not a data transfer cycle and SLAVES should not respond.

Functional Modules

The modules involved in a data transfer are always classified as a MASTER and a SLAVE. The MASTER is the module controlling the transfer, and the SLAVE is the responding or addressed module. Some boards may be designed with both MASTER and SLAVE modules. For example, a board containing a processor which requires VMEbus access would contain a MASTER module. If the same board also contained memory accessible from the VMEbus, it would also contain a SLAVE module.

Bus Arbitration Philosophy

As microprocessor costs decrease, it is becoming more cost effective to design systems with multiple processors sharing global resources.

The most fundamental of these global resources is the data transfer bus through which all other global resources are accessed. Therefore, any system supporting multiprocessing must provide an allocation method for the data transfer bus. Because speed of allocation of the data transfer bus is vital, a hardware allocation scheme must be provided. The VMEbus meets this need with its Bus Arbitration subsystem.

The VMEbus arbitration subsystem is designed to:

- Prevent simultaneous access of the bus by two MASTERS.
- Schedule requests from multiple MASTERS for optimum resource use.

The logic used to implement the bus allocation algorithm is called the ARBITER. It is the ARBITER's responsibility to respond to requests for the bus and to optimize usage by proper control of the allocation process.

ARBITER Options

The ARBITER used to control the arbitration system may be one of three options. An option PRI (Priority) ARBITER always assigns the bus on a fixed priority basis wherein each of four bus request lines are assigned fixed priorities from highest (BR3*) to lowest (BR0*). An option RRS (Round Robin Select) ARBITER assigns the bus on a rotating priority basis. If the current bus MASTER is level "n," the highest priority will be given to level "n-1" and proceed sequentially from there. An option ONE (Single level) ARBITER only honors requests on BR3* and relies on the daisy-chain structure for priority determination.

ARBITER Operation

Except for OPTION ONE ARBITERS, the bus arbiter accepts requests for the bus on four request lines, which are driven by open-collector drivers so that several MASTERS can share a common request line. Each request line has a corresponding grant daisy-chain line (BG3IN*/OUT* through BG0IN*/OUT*). If the bus is idle when a request is received, the ARBITER will immediately respond on the grant line corresponding to the level of request pending. When that MASTER then relinquishes the bus, the ARBITER will respond with a grant to the highest pending level of request. If no requests are pending at the time the currently active MASTER relinquishes control, the ARBITER will wait in the idle state until a bus request is received.

In addition to the ARBITER allocating the bus on an assigned basis, a secondary level of prioritization is built into the bus itself. The bus grant signals are daisy-chained in such a way that REQUESTERS sharing a common request line are prioritized by slot position. The REQUESTER closest to slot one has the highest priority.

OPTION ONE ARBITERS respond only to bus request 3 (BR3*) and depend on the daisy-chain of BG3IN*/BG3OUT* to do the prioritizing.

Arbitration Bus Line Structures

The arbitration bus consists of six bused VMEbus lines and four broken or daisy-chained lines. These daisy-chained lines

require special signal names. The signals entering REQUESTERS are identified as "Bus Grant IN" lines (BGxIN*), while the signals leaving the REQUESTER are identified as the "Bus Grant OUT" lines (BGxOUT*). Therefore, the lines which leave slot N as BGxOUT* enter slot N + 1 as BGxIN*.

Note: In all descriptions, the terms BRx*, BGxIN*, and BGxOUT* are used to describe the bus request and bus grant lines, where x may have any value from zero to three.

In the VMEbus arbitration system, a REQUESTER will drive the following lines:

1 bus request line	(one of BR0* through BR3*)
1 bus grant out line	(one of BG0OUT* through BG3OUT*)
1 bus busy line	(BBSY*)

If a VME board does not use some bus request levels, it must jumper the respective bus grant in lines to their respective bus grant out lines.

The ARBITER will drive the following:

1 bus clear line	(BCLR*) (OPTION PRI ARBITER only)
4 bus grant in lines	(BG0IN* through BG3IN* at slot A1)

An OPTION ONE ARBITER drives only BG3IN* at slot A1.

Two additional lines are intimately connected with the power-up and power-down sequencing of the arbitration system. These are:

1 system reset line	(SYSRESET*)
1 AC power fail	(ACFAIL*)

PRIORITY INTERRUPT

Interrupt Philosophy

Interrupt subsystems may be divided into two groups:

- Single handler systems — have a supervisory processor which receives and services all bus interrupts.
- Distributed systems — have two or more processors which receive and service bus interrupts.

The single handler system architecture is, perhaps, easier to understand because of its similarity to single processor systems. Any system which has interrupt capability must have a set of interrupt servicing routines in its executive software. Each of the routines may be thought of as a task which is activated by an interrupt. If the system has a real-time executive, these interrupt routines would then operate as tasks under this executive.

In a single processor or single handler system, the executive software and all of the interrupt routines are executed by one processor.

Single Handler Systems

This type of architecture is well suited to machine or process control applications. The dedicated processors are the ones typically interfaced to the machine or process being controlled, so it is important that their processing is interrupted as little as possible by bus activity.

The dedicated processors in the system are typically controlling some external machine or process. The task of controlling this machine or process may consist of several subtasks, some of which are noninterruptable (i.e., loss of control may result if the task once started is not finished within a specific time). Therefore, the dedicated processor may mask some or all of its interrupts while executing these noninterruptable subtasks.

To summarize, in a dedicated system, the supervisory processor is the destination for all bus interrupts. This allows it to service all interrupts in a prioritized manner. The dedicated processors are not required to service interrupts from the bus, but give primary attention to the interrupts received from the machine or process which they control.

Distributed Systems

This type of architecture is well suited to applications where incoming tasks may be assigned to the next available processor. Each of the co-equal processors executes part of the system executive software, and services only those interrupts directed to it by other processors within the system. Since the servicing of some of these interrupts may require access to system resources, the parts of the executive software must communicate through globally accessed memory in order to allocate resources and resolve lock-ups.

Signal Lines Used In Handling Interrupts

The data transfer bus, the arbitration bus, and the interrupt bus are all used in the process of generating and handling bus interrupts.

Interrupt Bus Signal Lines

The interrupt bus consists of seven interrupt request signal lines, one daisy-chain signal line, and one interrupt acknowledge line:

IRQ1*
IRQ2*
IRQ3*
IRQ4*
IRQ5*
IRQ6*
IRQ7*
IACK*
IACKIN*/IACKOUT*

Each interrupt request line may be driven low by an INTERRUPTER to request an interrupt. In a single handler system, these interrupt request lines are prioritized, with IRQ7* having the highest priority.

The IACK* line runs the full length of the bus and is connected to the IACKIN* pin of slot A1. When it is driven low, it initiates a low-going transition down the INTERRUPT ACKNOWLEDGE DAISY-CHAIN. This may not occur immediately, since additional constraints are placed on the propagation of IACKIN*/IACKOUT*.

INTERRUPT ACKNOWLEDGE DAISY-CHAIN — IACKIN*/IACKOUT*

Each of the seven interrupt request lines may be shared by two or more INTERRUPTER modules. Because of this, some method must be provided to assure that only one of the modules is acknowledged. This is done by means of the INTERRUPT ACKNOWLEDGE DAISY-CHAIN. This daisy-chain line passes through each board on the VMEbus. When an interrupt is acknowledged, IACKIN* is driven low at slot A1. Each module which is driving an interrupt request line low must wait for the low level to arrive at its board slot before accepting the acknowledge. The module accepting the acknowledge does not pass the low level down the daisy-chain, thereby guaranteeing that only one module will be acknowledged.

VMEbus UTILITIES

Introduction

Utility lines supply periodic timing signals and provide initialization and diagnostic capability for the VMEbus. Utility lines include:

System Clock	(SYSCLK)
AC Fail	(ACFAIL*)
System Reset	(SYSRESET*)
System Test	(SYSFAIL*)

ACFAIL* and SYSRESET* are typically driven by a POWER MONITOR module. Its purpose is to detect power failures and reset the system upon power up, initiating a power-up self-test.

Utility Signal Lines

System Clock (SYSCLK) Specification

The system clock is an independent, non-gated, fixed frequency, 16 megahertz, 50 percent (nominal) duty cycle signal. it can be used to generate on-board delays or timing functions where a fixed duration delay will be generated by counting off a known time base. ***SYSCLK has no fixed phase relationships with other VMEbus timing.***

System Initialization and Diagnostics

System Reset (SYSRESET*) is an open collector line driven by a POWER MONITOR module and/or by a manual switch (such as from an operator's panel). Failure of a system test is shown via the system fail line (SYSFAIL*). It is ***recommended*** but ***not required*** that all boards within the VMEbus system drive this system fail line low upon power up, and maintain it low until they have passed their respective self tests. Non-intelligent boards which are incapable of self test could maintain this line low until a MASTER in the system completes a test on them and writes to their on-board test register. Whenever SYSRESET* is driven to low, it must be held there for a minimum period of 200 milliseconds.

After SYSRESET* is released, the system software executes a test sequence. Upon successful completion of testing, all boards in the system release SYSFAIL* and go into the normal operating mode. The SYSFAIL* line is not allowed to go high if any board detects a failure.

SYSFAIL* can also be driven low at any time during normal operation as the result of a detected system failure. As an example, an intelligent board may periodically perform an on-board self-test and drive SYSFAIL* low if a failure occurs.

VMEbus ELECTRICAL CONSIDERATIONS

Introduction

The transmission of data inside the basic rack between VME boards such as processors, memories and I/O units, takes place via a single backplane in 24-bit address/16-bit data systems and via two backplanes in 32-bit address and/or 32-bit data systems. These backplanes are characterized by the following features:

- Maximum signal line length = 19″ (50 cm) including connection to termination networks
- Maximum number of slots (loads) = 20 (excluding terminations)
- Provision for the distribution of power (+5 STDBY, +12 V, −12 V)
- Termination networks at each end of the bus

All VME boards are interfaced to the VMEbus via circuits which must follow the rules to ensure proper timing and minimize noise and crosstalk problems. VMEbus signal lines are normally driven by saturated logic (TTL) drivers, although any technology which complies with this specification may be used.

Power Distribution

Power in a VMEbus system is distributed on the backplane as regulated direct current (dc) voltages. The available voltages are described as follows:

Mnemonic	Description	Variation (see note)	Ripple Noise Below 10 MHz (PK-PK)	Connector Pin Numbers	Maximum Current Draw Per Slot
+5 V	+5 Vdc standby	+0.25/−0.125 V	50 mV	a32,b32,c32	3A
+12 V	+12 Vdc power	+0.60/−0.36 V	50 mV	c31	1A
−12 V	−12 Vdc power	−0.60/−/+.36 V	50 mV	a31	1A
+5 V STDBY	+5 Vdc standby	+0.25/−0.125 V	50 mV	b31	1A
GND	ground	Ref		a9,11,15,17,19 b20,23 c9	

Note: The non-symmetric variation spec is given to ensure that the dc power will remain within the tolerance required by most IC's despite any drops resulting from power distribution on individual VME boards.

Electrical Signal Characteristics

Other than power supply lines, all VMEbus signals are limited to positive levels between 0 and 5.0 volts.

— 0.0 V ≤ Driver low output level ≤ 0.6 V

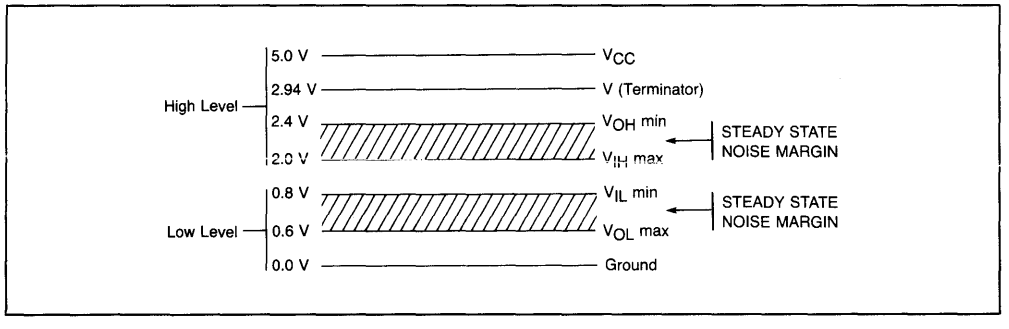
— 0.0 V ≤ Receiver low input level ≤ 0.8 V

— 2.4 V ≤ Driver high output level ≤ 5.0 V

— 2.0 V ≤ Receiver high input level ≤ 5.0 V

Figure 2 gives a simple graphic representation of these levels.

FIGURE 2 — VMEbus Signal Levels



Depending on the function required, the VMEbus uses three-state, open collector, and totem-pole drivers.

Bus Driving Requirements

The bus signals fall into five classes:

- The special (high-current) three-state class:
  - — The handshaking signals AS*, DS0*, DS1*
- The standard three-state classes:
  - — A01..A31, D00..D31
  - — The transfer status signals AM0..AM5, IACK*, LWORD*, WRITE*
- The special (high current) totem-pole class:
  - — System utilities SYSCLK, BCLR*
- The standard totem-pole class which includes:
  - — The daisy-chain signals BG0OUT*–BG3OUT*
  - — The interrupt daisy-chain line, IACKOUT*
- The open collector class which includes:
  - — The bus allocation signals BR0*–BR3*, BBSY*
  - — The interrupt signals IRQ0*...IRQ7*
  - — The response signals DTACK* and BERR*
  - — SYSFAIL*, SYSRESET*, and ACFAIL*

Backplane Signal Line Interconnections

The VMEbus is an asynchronous, high speed bus intended for high performance systems. The backplane signals must be treated as transmission lines. The address and data set-up times specified take into account the fact that drivers available on the market today may not drive the signal line above the high level threshold until a reflection is received from the end of the bus. It is important, however, to control the signal reflections to minimize ringing. The following paragraphs specify the backplane characteristics which do this.

Termination Networks

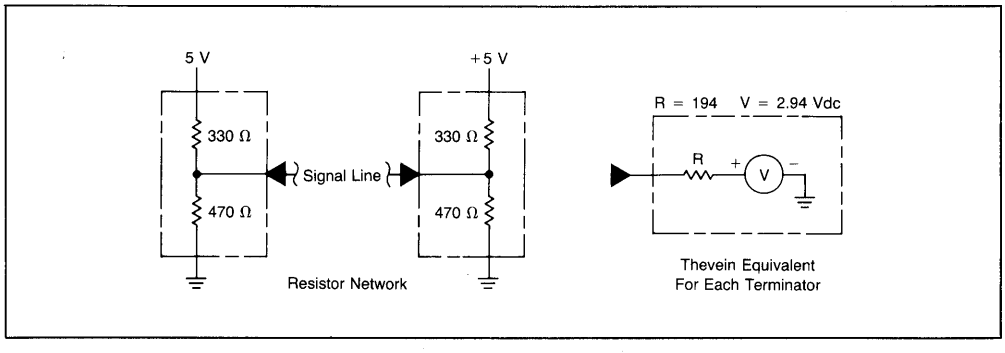
A standard termination is used on each end of all VMEbus signal lines except daisy-chain lines. The termination serves four purposes:

- Reduces reflections from the ends of the backplanes.
- Provides a high state pull-up for open-collector drivers.
- Discharges the line when three-state devices are disabled.
- Provides a standing current for the driver sink transistor to switch, helping to drive the transmission lines on positive transitions.

The Thevenin equivalent of the standard termination is shown in Figure 3. One possible resistor network configuration which achieves this is also shown. Resistor values and source voltage of the Thevenin equivalent must be 5% tolerance maximum.

Note: The example termination presents its Thevenin equivalent impedance only when the +5 V source is adequately decoupled to ground; capacitors between 0.01 and 0.1 μF are recommended as close as possible to the V_{CC} pin of the resistor termination packages.

FIGURE 3 — Standard Bus Termination



MECHANICAL SPECIFICATIONS

Introduction

Figure 4 is a simplified drawing which shows the physical relationships between the components of a typical VMEbus system. VME boards are inserted into the card rack in a vertical position from the front of the system. The board components are to the right and the "P1" Connector is on the top.

VME Board Sizes and Dimensions

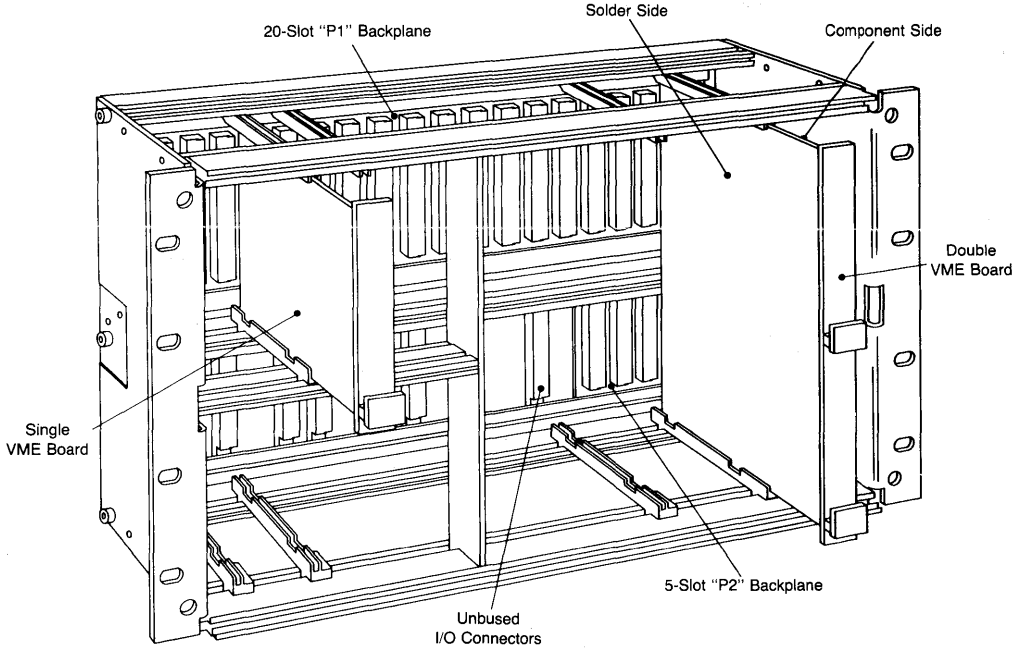
There are two sizes of VME boards: single high and double high. (See Figure 5.) The single high board has only one 96-pin DIN connector (P1) on its "back edge." When I/O signals must be connected to single high boards, they are connected to the front of the board. No standard connectors are prescribed for these connections. A double height board typically has two connectors on its back edge. The top connector (P1) is a 96-pin 603-2-IEC-C096-M connector while the bottom (P2) may be a similar 96-pin connector or may be some other DIN standard connector. The 96-pin connector must be used when the board is designed to use P2 for address and data bus expansion.

Front Panels

There are mechanical specifications for the single and double height VME board front panels. Where screws are used to secure the top and bottom of the panels to the board enclosure, the thread must be M2.5. Quarter-turn fasteners are acceptable alternatives but they must be mountable in the holes specified. All front panels are 2.5 mm thick.

Front panels may be designed for widths other than 20.32 mm (0.8 inch) but these must be integer multiples (N) of 5.08 mm.

FIGURE 4 — VME Chassis, Typical Configuration



Single High Front Panels

The grid format on the rear face of the front panel is standardized to provide an attractive frontal appearance and achieve consistency with the board grid. Wherever possible, front panel components such as LED's and switches should be centered on a grid point.

Double High Front Panels

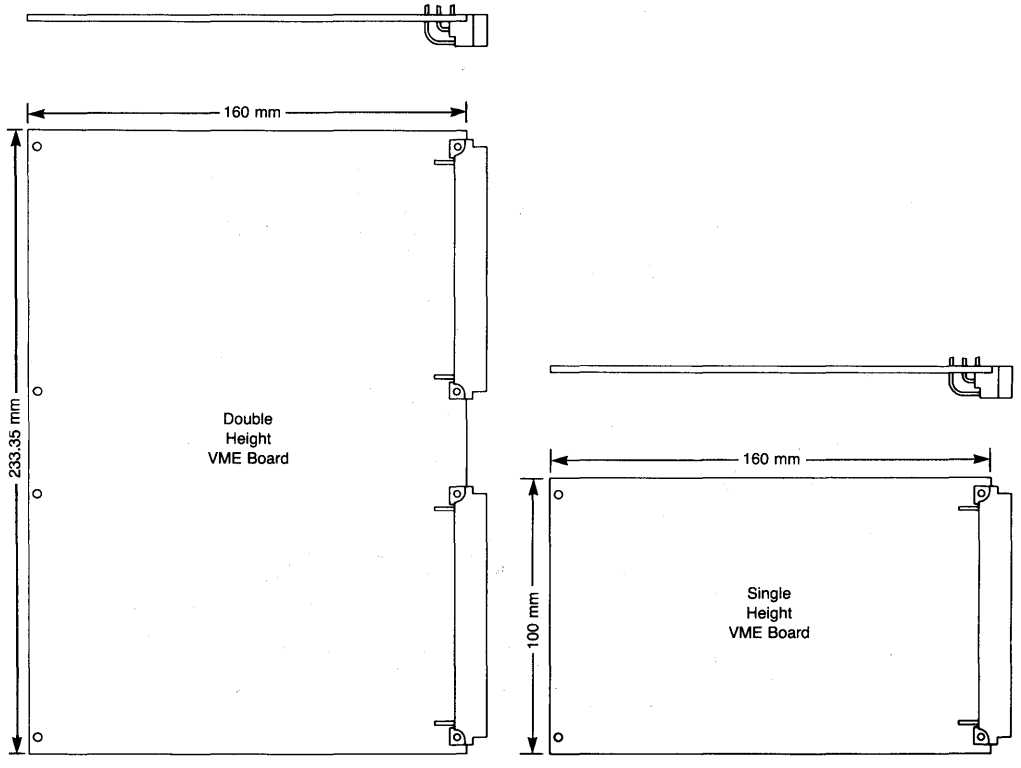
The grid format on the rear is consistent with the board grid and is used to standardize the location of the front panel components for appearance sake.

A handle is provided in the center of the double height front panel so that when double and single VME boards share the same subrack, the top row of handles will form an unbroken line.

Filler Panels

Filler panels are sometimes used where the backplane positioning requirements cause a gap in the front panel or where boards with different width panels are used. These filler panels require no mounting brackets because they are not attached to the PC boards. They are secured to the enclosure by screws at their top and bottom ends.

FIGURE 5 — VME Double and Single Height Boards



Backplane

The primary backplane is designated as the J1 backplane. In many cases this is the only backplane in the VME system. When a double high sub-rack is used, this backplane is mounted in the upper portion. When the expanded VMEbus is used, a second backplane, designated the J2 backplane, must be installed below the J1 backplane in the lower portion of the rack.

Note: The J2 backplane typically does not bus all slots across the bottom of the rack. Some of the slots are usually left unbussed to provide I/O support.

Card slots are designated A01, A02 ..... An, slot numbering starting from the left when viewed from the front of the backplane. The daisy-chain propagation must start with A01 and go to An.

Bussing on the J2 level backplane is required only on the center row (row B of contacts, pins 1 through 32).

The use of 603-2-IEC-C096-F . . . connectors on VME backplanes is mandatory.

If the J1 backplane connectors have wirewrap pins, these may be used to jumper the daisy-chain signals for slots which are currently empty. Otherwise, pins must be provided on the backplane for this purpose, and should be designated JB21 for pin B21 jumper, etc. Thus, to bypass a slot on the interrupt daisy-chain requires a jumper from JB21 to JB22. It is recommended that all of the daisy-chain jumpers for a slot be positioned next to the connector for that slot. See Figure 6.

Backplane Construction Techniques

Many backplane construction techniques are available to the designer. Backplane construction can be of a multilayer laminated or unlaminated (sandwiched) design. The unlaminated design is composed of multiple discrete, two-sided, glass epoxy boards separated by mylar insulators. The boards and mylar insulators are held together, using high force press fit connectors. This process provides a gas-tight connection between the contacts and the plated-through hole in each board. One layer provides the voltage plane, and still another layer is the ground plane. Connectors are press fitted into the boards and the mylar insulating sheets form the multilayer backplane.

A second possible backplane construction technique is to laminate the boards. This lamination process bonds the boards and insulating layers together. Connectors are then inserted into the backplane and soldered.

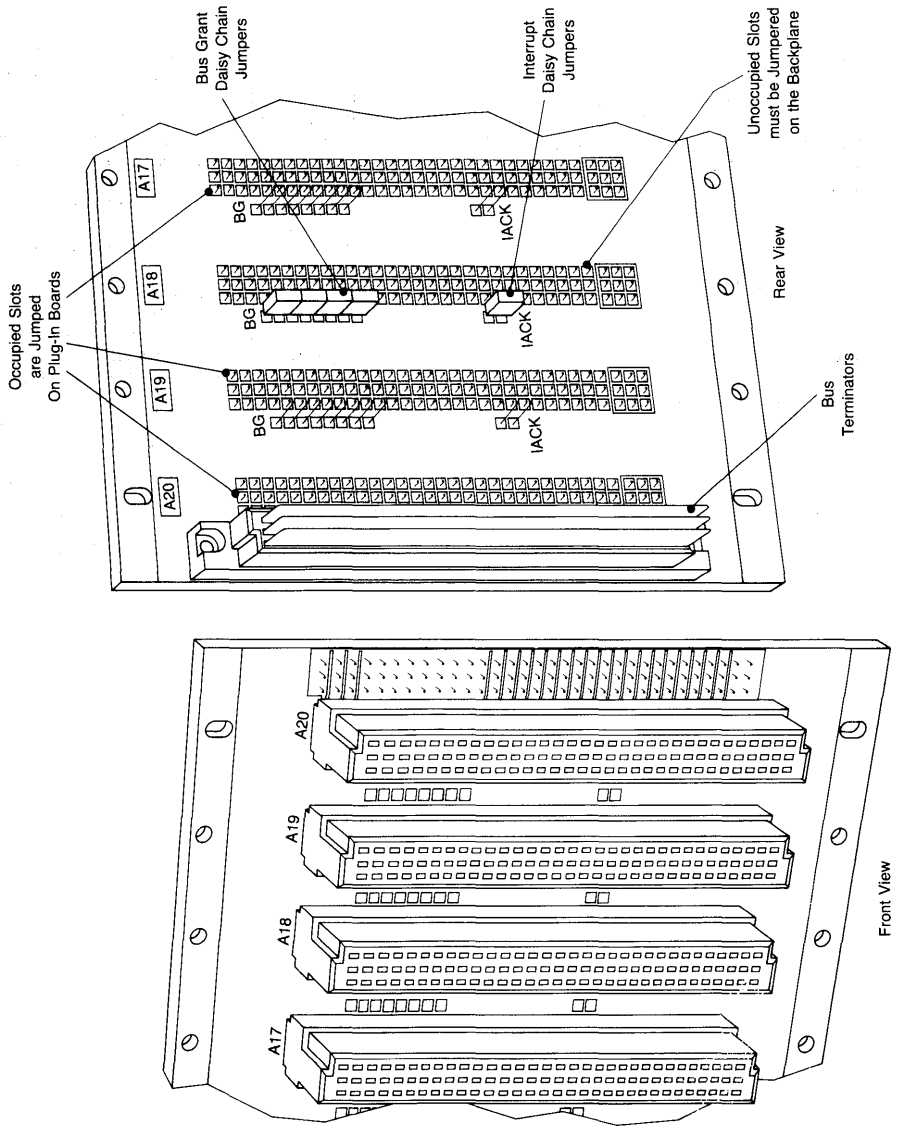


FIGURE 6 — VMEbus Backplane

VMSbus Multiprocessor Serial Communications Bus

- Provides Separate Path for Communicating the Occurrence of an Event Between Multiple Processor Modules
- Allows for Fault Tolerant System Design
- Provides for Intelligent Semaphore Control for Shared Resource Allocation
- Uses Token Passing (up to 1024 Tokens) with Low Software Overhead
- Provides Low-Cost Alternative Bus
- 3.2 Mbit/Sec. Transfer Rate

SERIAL BUS LINES

The serial bus includes three lines (included in the VMEbus backplane): Serial Clock (SERCLK), Serial Data (SERDAT*), and System Reset (SYSRESET*).

SYSRESET* used to initialize all modules on the serial bus.

SERCLK is driven by a high-current totem-pole driver from the SERIAL CLOCK module, of which there is one per system.

SERDAT* is a wired-OR line which can be driven with an open-collector driver by any module. A "one" on SERDAT* results when one or more modules drive it Low. A "zero" results when no module is driving SERDAT*, so that the backplane terminating resistors pull the signal High.

VMSbus Overview**

The serial bus is intended to meet the needs of multiprocessor systems. The following paragraphs describe some of these needs and how the serial bus provides for them.

THE SERIAL BUS ALLOWS THE COMMUNICATION OF EVENTS

An ***event*** could be defined as any significant occurrence which is not a direct, immediate, and normal result of what the processor is currently doing. For example, if a program commands a disk controller to start a transfer, we would not consider the fact that the disk controller starts to be an event. The completion of the disk transfer ***is*** an event.

We will discuss events as they impact multitasking systems. In such systems there is normally a task for each event that the system recognizes. When an event occurs, the processor may be ***reassigned*** from executing one task to executing another.

A ***task*** can also cause the reassignment of the processor. For example, the task which is currently being executed by the processor may need to wait for an event to occur before it can proceed. In this case it requests system software to reassign the processor to other tasks until the event occurs. As another example, a task may request that another task be executed. Communication between tasks is often done through such requests.

There are two kinds of events: those which occur outside the processor (external events), and those which occur inside the processor (internal events). External events are commonly communicated to the processor by means of ***interrupts.*** Internal events don't need to be communicated to the processor, but they may still cause it to be reassigned to a new task. (An example of an internal event is an attempt to divide by zero.)

Microprocessors have a small number of "interrupt request" lines that can be used to communicate events to them. Since there are usually many possible events, they must all share these interrupt request lines. When the processor detects an interrupt, it must gather more information in order to figure out which event has occurred and which task is to be executed. There are several methods for doing this: polling of I/O status registers, reading a "vector" from the interrupting device, etc. In each case the processor does one or more "read cycles" to collect this information.

The newer backplane buses have a provision for directing interrupts to several different processors. For example, VMEbus has seven interrupt request lines which can each be monitored by a different processor. If a device in the system needs to communicate an event to a processor, it can request an interrupt on a line that is monitored by that processor.

But even this approach has its limitations. When a processor acknowledges (answers) an interrupt request, the processor typically must read a vector before assigning itself to the new task. If this is done as a transfer on the system bus, the processor may have to ***request*** the bus and ***wait*** until it becomes available. If other bus masters are using the bus, this delay may be too long or too unpredictable to satisfy the needs of time-critical, high performance applications. Sometimes the problem can be solved by rearranging the priorities of the various bus masters, but the other masters' needs for the bus may be equally or more important to the performance of the system.

**This summary was prepared from Revision A, 10/83 of the VMSbus specification.

NOTICE

These bus summaries are provided only as a general aid to designers. Since VMEbus, VMSbus and VSB are still evolving, a copy of the latest version should be obtained to insure use of accurate figures for an actual design.

We conclude from all this that what is needed is a path specifically designed for communicating events. To be useful in a multiprocessing system, this path must allow any processor to communicate events to any other, as well as permitting other devices in the system to communicate events to any of the processors. Such communication is one of the primary purposes of the serial bus.

THE SERIAL BUS AIDS IN "FAULT TOLERANT" SYSTEM DESIGN

Another trend in microprocessor systems is toward applications which demand "fault tolerance." This simply means that a system must be able to continue operation despite one or more hardware failures. One way to satisfy this need is by using several processors in a system, and providing ways for each processor to observe the operation of all the other boards, to detect when one is malfunctioning. Such systems can also include hardware registers on each board which control the board's backplane bus drivers. If one of the processors detects a malfunctioning board, it can write into a control register on that board and turn off the board's bus drivers, effectively "disconnecting" it from the backplane.

The serial bus provides another pathway between boards, which can be used to prevent a failed module from interfering with the operation of the system bus. The serial bus can be used to disable the drivers of such a board, or direct a board-specific Reset to it. Conversely, if a failed board prevents the proper operation of the serial bus, the system bus can be used to disable the board's access to the serial bus.

THE SERIAL BUS PROVIDES FOR "INTELLIGENT SEMAPHORES"

The third primary use of the serial bus is again related to multiprocessor systems. When a system includes multiple processors which can simultaneously try to access and use a variety of shared resources (parts of the system), some means must be found to control this access and use. A simple example of such a problem is when two processors simultaneously set out to use a shared hardware device such as a printer or disk. Other cases where interprocessor control is needed include access to: a data file, a data or control table in memory, or a section of program code which must be used by only one processor at a time.

The most widely used solution to these problems is the "semaphore." A semaphore is simply a location in memory which multiple system processors can access, but not simultaneously. A processor first reads the memory location to test whether another processor already has control of the shared resource, and then (if not) it writes to the location to show that it now has control. The system hardware must ensure that these two steps happen without allowing any other processor to access the semaphore location. The newer backplane buses include provisions for semaphore operations.

But if the semaphore will typically stay set for a relatively long time, system software and the processor should go on to other tasks. In this case, on-board hardware must generate an interrupt when the semaphore is cleared so that the processor can retry the RMW on the system bus. Better still, the on-board hardware can itself retry setting the semaphore and interrupt the processor only when it has gained control. An example of such a "long term" semaphore would be one controlling access to a physical device like a printer or a disk.

For "long term" semaphores, the serial bus offers a great improvement. A processor can turn the entire operation of setting the semaphore over to on-board serial bus hardware, and be interrupted only when the semaphore has been set and the associated resource is actually available.

THE SERIAL BUS ALLOWS TOKEN PASSING SCHEMES

Sometimes a system has a group of several interchangeable resources. If semaphores are used to allocate these resources, then whenever a resource is needed, the semaphores must be polled until one is found to be available. In some cases it may make more sense to have a "token" in the system for each of these interchangeable resources, which is ***passed around*** among its "users" while the resource is available, but is ***retained*** by a user while it uses the resource. (Such "token passing" operations can be likened to a "daisy chain" in a backplane bus, which has been looped back on itself to make a ring.)

The serial bus allows a large number of such tokens (up to 1024) to be created and passed from board to board with very little software overhead.

THE SERIAL BUS PROVIDES A LOW COST ALTERNATIVE BUS

The serial bus is a unique new concept in microprocessor system interfacing. Its capabilities make it very attractive for use in complex multiprocessing systems. But in addition to its power, a serial bus interface can be implemented at lower cost than a system bus interface. With the LSI support which will soon be available, serial bus hardware can be implemented in a fraction of the board space required for a parallel backplane bus.

Thus the serial bus offers a very attractive ***alternative*** to use of a parallel system bus for boards which do not require a high data rate. It can even be used as the primary system bus in some applications.

SERIAL BUS OPERATION

The serial bus interface system consists of two signal lines named SERCLK and SERDAT* and six module types called HEADER SENDERS, HEADER RECEIVERS, DATA SENDERS, DATA RECEIVERS, FRAME MONITORS and a SERIAL CLOCK. The SERCLK line is driven by a totem pole driver in the SERIAL CLOCK module. The SERDAT* line can be driven Low by all six module types using open collector drivers. When no module drives SERDAT* Low, the bus terminating resistors pull it to a High level. The SERDAT* line is a Low-True signal (i.e. a "one" is represented by a Low level). This fact, plus the open collector characteristics of the drivers, results in "logical OR'ing" when data is placed on SERDAT* by more than one module.

In some cases on-board signals connect modules on the same board, but most communication between modules is done by sending "frames" on the SERDAT* line. These frames are composed of "subframes" which are sent by various modules. A frame is initiated when a HEADER SENDER module sends a "Header subframe." The other modules then respond by sending subframes according to a prescribed protocol until the end of the frame is reached.

During the Header subframe transmission, HEADER SENDERS are required to sample each bit on the SERDAT* line while they are sending. If a HEADER SENDER detects SERDAT* Low when it is sending a "zero" (i.e. when it isn't driving SERDAT* Low), it stops sending. This allows other HEADER SENDER(S) to finish sending the Header subframe without interference. This method of arbitration allows several HEADER SENDERS to start sending frames simultaneously, without affecting each other's transmissions. (One of the transmissions will be successful while the others are tried again later.)

Serial bus modules are found on board in ***groups.*** The following are the most common groups:

TYPE 1) A HEADER SENDER and a FRAME MONITOR
TYPE 2) A HEADER RECEIVER and a flip-flop
TYPE 3) A HEADER RECEIVER and a DATA SENDER
TYPE 4) A HEADER RECEIVER and a DATA RECEIVER
TYPE 5) A HEADER RECEIVER, a DATA SENDER, and a DATA RECEIVER

A TYPE 1 group is used to initiate frames by sending a "Header subframe." The Header subframe specifies what other modules will participate in the frame transmission by providing two ten bit "selection codes." Each HEADER RECEIVER on the serial bus has a ten bit code which it compares with the two codes in the subframe. If its code matches either of them, depending on the type of group the HEADER RECEIVER is in, it responds by changing the state of its flip-flop, by telling its DATA SENDER to send, or by telling its DATA RECEIVER to receive.

USING THE SERIAL BUS TO TRANSFER DATA

The HEADER SENDER can determine whether there is a frame in progress from the FRAME IN PROGRESS signal generated by its FRAME MONITOR. If there is no frame in progress it can initiate one by sending a Header subframe. This subframe has a ten bit "S field" and a ten bit "R field". To transfer data, the HEADER SENDER puts a selection code in the S field that corresponds to some TYPE 3 or TYPE 5 group on the bus, and a selection code in the R field that corresponds to some TYPE 4 or TYPE 5 group. (The actual codes used to select these groups depend on how the system software or firmware has configured the system. There are no selection codes used exclusively to select TYPE 4 groups, etc.)

Each HEADER RECEIVER on the serial bus compares these codes against its code. One or more of the TYPE 3 or TYPE 5 HEADER RECEIVERS finds a match with the S field. It tells its DATA SENDER to send data. In a similar way, one or more of the TYPE 4 or TYPE 5 HEADER RECEIVERS finds a match with the R field and tell its DATA RECEIVER to receive data.

The actual number of bytes transferred is left up to the DATA SENDER. After the HEADER SENDER has sent the Header subframe, the DATA SENDER sends a three bit subframe indicating the number of bytes it intends to send to the DATA RECEIVER, followed by the data bytes. The DATA RECEIVER then responds with an indication that it has received the data bytes.

USING THE SERIAL BUS TO SET AND RESET FLIP-FLOPS

When a HEADER SENDER is used to set or reset a flip-flop, it sends a Header subframe as in the case described above. Instead of sending codes for TYPE 3, 4 or 5 groups in the S and R fields, however, it sends the code for a TYPE 2 group in one of the fields, and a "dummy" code (all ones) in the other field. If the frame is intended to ***set*** a flip-flop, it sends the TYPE 2 code in the S field, and the dummy code in the R field. If the frame is intended to ***reset*** a flip-flop, it sends the dummy code in the S field and the code for the TYPE 2 group in the R field.

When the Header subframe is sent, each HEADER RECEIVER in the serial bus compares the codes in the S and R fields to its own code. One or more HEADER RECEIVER(S) in a TYPE 2 group matches the S (or R) field and sets (or resets) its on-board flip-flop.

OTHER USES FOR THE SERIAL BUS

Groups of serial bus modules like those described above can be used as building blocks for very powerful system

configurations. For example, the combination of a TYPE 1 and a TYPE 5 group, on each of two boards, can be used by one board to pass an address and read the contents of a memory location on the other. TYPE 2 groups can be used to reinitialize one or more of the boards in a system, or to selectively disconnect a failed board from the system bus. TYPE 2 groups can also be used to provide "semaphores" which are functionally superior to semaphores in a common memory, or to provide "token passing".

SERIAL BUS FRAMES

A frame is actually composed of subframes which are sent by several modules in sequence whenever a HEADER SENDER sends a Header subframe. Depending on the types of module groupings that a frame selects, different modules drive and receive the various subframes, as follows:

6

Frames that select TYPE 2 module groups . . .

Subframe	Sent By	Received By
Header	HEADER SENDER	HEADER RECEIVER
Frame Type	Nobody (000)	HEADER RECEIVER, FRAME MONITOR
Frame Status	HEADER RECEIVER	FRAME MONITOR

Frames that select TYPE 3, 4 and 5 module groups . . .

Subframe	Sent By	Received By
Header	HEADER SENDER	HEADER RECEIVER
Frame Type	DATA SENDER	DATA RECEIVER, FRAME MONITOR
Data	DATA SENDER	DATA RECEIVER
Frame Status	DATA SENDER, DATA RECEIVER	FRAME MONITOR, DATA SENDER, DATA RECEIVER

Frames that get cancelled . . .

Subframe	Sent By	Received By
Header	HEADER SENDER	HEADER RECEIVER
Frame Type	HEADER RECEIVER	FRAME MONITOR, DATA SENDER, DATA RECEIVER

When a Header subframe is sent, modules on the serial bus are selected to interact during the remainder of the frame. This interaction can be seen on the serial bus as a sequence of subframe transmissions. Depending on what module groups are selected, we may see eleven possible kinds of frames.

1. A Flip-flop Set frame
2. A Flip-flop Reset frame
3. A Semaphore Set frame
4. A Token Passing frame
5. A 1 byte Data Transfer frame
6. A 2 byte Data Transfer frame
7. A 0.4 byte Data Transfer frame
8. An 8 byte Data Transfer frame
9. A 16 byte Data Transfer frame
10. A 32 byte Data Transfer frame
11. A Cancelled frame

A FLIP-FLOP SET FRAME

A Flip-flop Set frame contains 4 subframes:

Header	Frame Type	Frame Status	Jam Bit
S field = TYPE 2 R field = all ones	000	010	0

The Header subframe is sent by a HEADER SENDER. This subframe has two selection code fields in it: the S field and the R field. The Flip-flop Set frame has the selection code of a TYPE 2 group in its S field and all ones in the R field. One (or more) HEADER RECEIVER(S) on the serial bus finds a match between the S field code and its own code, sets its flip-flop, and responds with 010 in the Frame Status subframe.

A FLIP-FLOP RESET FRAME

A Flip-flop Reset frame contains four subframes:

Header	Frame Type	Frame Status	Jam Bit
S field = all ones R field = TYPE 2	000	001	0

The Header subframe is sent by a HEADER SENDER. As in the Flip-flop Set frame, this subframe has two selection

codes in its S field and R field. The Flip-flop Reset frame, however, has the selection code of a TYPE 2 group in its R field and all ones in the S field. One (or more) HEADER RECEIVER(S) on the serial bus finds a match between the R field code and its own code, resets its flip-flop, and responds with 001 in the Frame Status subframe.

A SEMAPHORE SET FRAME

A Semaphore Set frame which is "successful" contains four subframes:

Header	Frame Type	Frame Status	Jam Bit
S field = TYPE 2 R field = Req. Code	000	010	0

The Header subframe is sent by a HEADER SENDER. As in all frames, this subframe has two selection codes in its S field and R field. The Semaphore Set frame has the selection code of a TYPE 2 group in its S field.

The R field of a Semaphore Set frame contains a ten bit code which represents the unique identity of the Requester that caused the frame to be sent. System software should ensure that two HEADER SENDERS are never allowed to send Semaphore Set frames with the same R field code. (In a multitasking system, this could be assured by assigning a unique Requester number to each task in the system.) The uniqueness of these Requester numbers guarantees that if two or more HEADER SENDERS try to set the same semaphore at the same time, only one of them will survive the serial bus arbitration and finish the frame.

A TOKEN PASSING FRAME

A Token Passing frame which is "successful" contains four subframes:

Header	Frame Type	Frame Status	Jam Bit
S field = TYPE 2 R field = TYPE 2	000	011	0

The Header subframe is sent by a HEADER SENDER. The Token Passing frame has the selection code of one TYPE 2 group in its S field and the code of another TYPE 2 group in its R field.

As with a TYPE 2 group used for semaphore operations, the HEADER RECEIVER in a TYPE 2 group used for token passing operations will not accept a Set frame if its flip-flop is already set. In addition, it will not accept a Reset frame while its flip-flop is reset. Whenever such a frame is sent, the HEADER RECEIVER "cancels" the frame.

A Token Passing frame is sent to simultaneously clear one flip-flop and set another. (In effect, it is passing a "token bit" from the flip-flop it resets to the one it sets.) If the flip-flop it is trying to reset is already reset (i.e. it doesn't have a token to pass) the frame will be cancelled. Likewise, if the flip-flop it is trying to set is already set (i.e. it is already holding a token) the frame will also be cancelled.

These frame cancellation features ensure that a token bit is never lost or created in the process of moving it from one board to another.

A DATA TRANSFER FRAME

A Data Transfer frame which is "successful" contains five subframes:

Header	Frame Type	Data	Frame Status	Jam Bit
S field = TYPE 3 or 5 R field = TYPE 4 or 5	001–110	(Varies)	011	0

The Header subframe is sent by a HEADER SENDER. As in all frames, this subframe has two selection codes in its S field and R field. The S field has the selection code of a TYPE 3 or TYPE 5 group, which includes a DATA SENDER. Similarly, the R-field has the selection code of a TYPE 4 or 5 group, which includes a DATA RECEIVER. At the conclusion of the Header subframe, the selected DATA SENDER(S) drive a code on SERDAT* during the Frame Type subframe, to indicate how many bytes will be sent. While it is sending this code, a DATA SENDER also samples SERDAT*. If it samples the value 111 in the Frame Type, the frame is "cancelled", and the DATA SENDER terminates its transmission.

Except in a Cancelled frame, the DATA SENDER sends a Data subframe after the Frame Type subframe. The Data subframe may be 1, 2, 4, 8, 16 or 32 bytes long.

A CANCELLED FRAME

A Cancelled frame contained only three subframes:

Header	Frame Type	Jam Bit
S field = TYPE 2, 3 or 5 R field = TYPE 4 or 5	111	0

The Header subframe is sent by a HEADER SENDER. It may be intended to set a semaphore or transfer data. In the former case, the cancelling of the frame indicates that the semaphore is already set. In the latter case, the cancelling of the frame indicates that the data transfer cannot be performed because one or more of the DATA SENDERS or DATA RECEIVERS selected by the S and R code is not ready for the transfer. A selected DATA RECEIVER may not yet have "disposed of" data it received in a previous transmission. Or, a DATA SENDER may not have been loaded with data to send.

In any of these cases, the HEADER RECEIVER with the flip-flop, DATA SENDER or DATA RECEIVER recognizes the problem and "cancels" the frame by driving all three bits of the Frame Type subframe to "one" (Low). In the case of a Data Transfer frame, the selected DATA SENDER(S) samples all three Frame Type bits as ones and doesn't send the data bytes. The 111 in the size field tells all selected DATA RECEIVERS and all of the FRAME MONITORS on the serial bus that there will be no data bytes nor any status. All FRAME MONITORS use this fact to signal their HEADER SENDERS that the serial bus is available for another frame. The FRAME MONITOR with the HEADER SENDER which initiated the frame also signals the problem to its on-board logic.

A "JAMMED" FRAME

A "jammed" frame may look like any of the frames described above. It differs from them in that (at least) the final single-bit "Jam Bit subframe" is one rather than zero as shown in the above frames. A Jammed frame occurs when one or more FRAME MONITORS in the system detects that the serial bus is "out of frame synchronization" due to an error induced by system noise, and sends a long series of ones on SERDAT*. A Jammed frame is ignored by all modules on the serial bus: no HEADER RECEIVER sets or clears an associated flip-flop, no DATA SENDER considers itself to have sent data, nor does any DATA RECEIVER consider itself to have received data. The frame which was jammed will be resent after the serial bus is "resynchronized".

VME Subsystem Bus

INTRODUCTION

The introduction of high performance 32-bit microprocessors, as well as the demands placed on microcomputers by the user community have created a need for multiprocessor systems built from board level products. The increase in the number of functions that such systems provided necessitated the introduction of a sophisticated subsystem bus. The VME Subsystem Bus (VSB) was designed to respond to these requirements.

It is a truly international bus, developed within a Special Working group which was formed by the International Electrotechnical Commission (IEC). The committee members represented East Germany, England, France, Japan, The Netherlands, The U.S.A., The U.S.S.R., and West Germany. The specification is currently studied by the IEC member nations, and is expected to become an approved international standard by the end of 1986.

The VSB allows a processor board to access additional memory and I/O over a local bus, removing traffic from the global bus and improving the total throughput of the system. The system has been designed to improve the performance of multiprocessor systems by allowing the design of functionally partitioned local subsystems.

Figure 1 shows a block diagram of the functional modules and sub-buses defined by the VSB specification. The VSB defines two sub-buses, the data transfer bus and the arbitration bus.

THE DATA TRANSFER BUS

The VSB includes a high speed asynchronous data transfer bus which allows masters to direct the transfer of binary data to and from slaves. The master initiates bus cycles in order to transfer data between itself and slaves. The slave detects bus cycles that are initiated by the active master and, when those cycles select it, transfers data between itself and the master. The VSB specification defines a mechanism that allows the master to broadcast the data to any number of slaves.

The interaction between the master and the selected slaves occurs during the execution of bus cycles. A bus cycle is a sequence of level transitions on the signal lines of the data transfer bus that results in the transfer of an address and (in most cases) data between the master and the selected slaves. The protocols of the VSB are fully asynchronous to provide a migration path from current to new technologies. The cycle is generally divided into three phases: an address broadcast, zero or more data transfers, and then cycle termination.

An interlocked relationship exists between the master and the slaves during the course of a cycle. As shown in Figure 2, the cycle begins after the master has been granted exclusive use of the bus. The active master initiates the cycle by broadcasting addressing information to the slaves, which acknowledge and respond to the address broadcast. The active master maintains the address broadcast until it detects this acknowledgment and then terminates the address broadcast.

The active master might then initiate one or more data transfers, or it might terminate the cycle without performing any data transfers. When slaves detect a data transfer they acknowledge it. The master maintains the data transfer until it detects this acknowledgment. After the master terminates the data transfer it might execute another data transfer, or it might terminate the cycle. When slaves detect a cycle termination they acknowledge it. This concludes the cycle.

The address broadcast phase is used to select the slaves that will participate in the cycle. During the address broadcast the active master broadcasts the addressing information and then asserts an address strobe. After the slaves acknowledge the address broadcast, the master terminates it.

FIGURE 1 — The Functional Modules And Sub-Buses Of The VSB

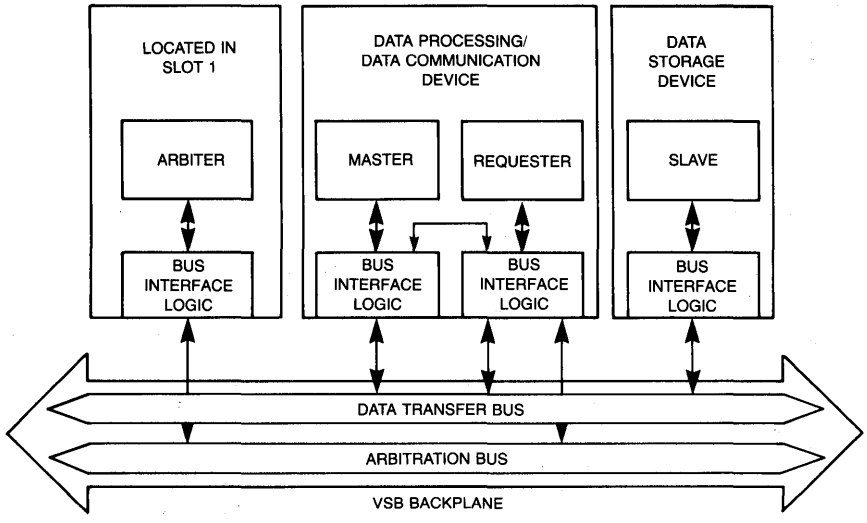
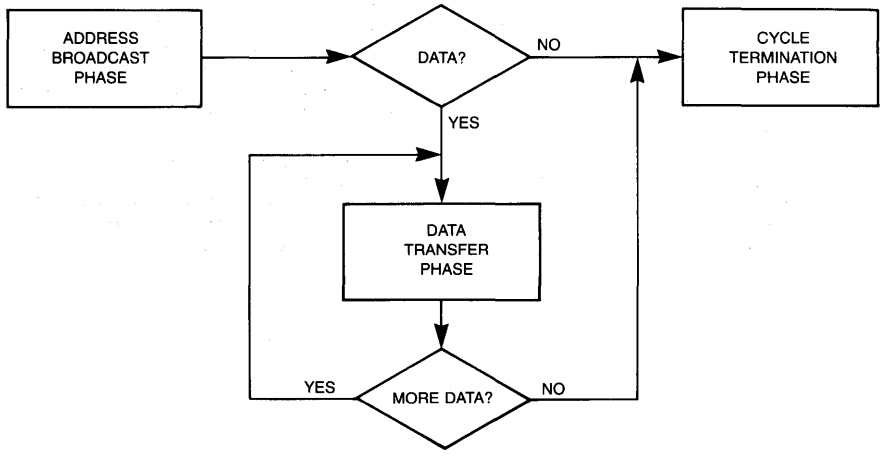


FIGURE 2 — General Flow Of A VSB Cycle



The smallest addressable unit of storage that can be provided by slaves is the byte location. A group of 4 byte locations which can be accessed simultaneously in a single data transfer is called a 4-byte group. During the address broadcast phase of the cycle, the master informs the system's slaves what is the size of the data transfer. During quad-byte cycles, the master requests access to four consecutive byte locations. During triple-byte cycles, the active master requests access to three consecutive byte locations. During double-byte cycles, the active master requests access to two consecutive byte locations. During single-byte cycles, the active master accesses one byte location.

The VSB defines three address spaces of 4 Gbytes each. The address that the master broadcasts might select byte locations in the system address space, in the alternate address space or in the I/O address space. Although not specifying the use of the three address spaces, it is expected that the system address space will primarily be used for memory, and the I/O address space for peripheral control devices. Use of the alternate address space will most likely vary with the system application. It might be used to provide a dedicated address space for co-processors, inter processor communications or any other use that might be necessary to implement the user's application.

During the address broadcast phase, the master can request indivisible access to the byte locations. Many modern processors can perform indivisible sequences, that is, sequences of memory accesses that have to proceed one after the other, without allowing any other processor to access any of the same byte locations before the sequence is complete. Such processors signal indivisible sequences differently from normal bus cycles, so that external hardware can guarantee the needed indivisibility. Some VSB slaves allow access to the byte locations they provide from more than one port; for example, their memory might be accessible over the system bus (e.g., the VMEbus), as well as over the VSB. Such slaves will typically lock out access from other ports to the byte locations that are selected during an indivisible access cycle.

When acknowledging the address broadcast, the responding slave informs the master what is the maximum number of byte locations that it can access in a single transfer. The VSB specification defines three slave sizes: single-byte, double-byte and quad-byte slaves. A single-byte slave responds to data transfer requests by accessing only one byte location during each data transfer. A double-byte slave responds to data transfer requests by accessing either one or two byte locations during each data transfer. A quad-byte slave responds to data transfer requests by accessing either one, two, three or four byte locations during each data transfer.

The protocols of the address broadcast phase provide slaves with a mechanism to prolong the address broadcast phase to fit their address decoding requirements. The way slaves acknowledge the address broadcast phase informs the master of how the system is configured. The master can detect whether a slave has been configured to respond to the cycle, whether the data transferred in the course of the cycle is broadcasted to additional participating slaves. The master can detect the fact that no slaves are installed in the system. The master's ability to positively determine whether its requested cycle will be responded to, makes it unnecessary to define a bus timer module.

Depending on the master's design, on the type of address acknowledge it receives, and on the type of cycle it executes, the master might then either terminate the cycle, or proceed to execute one or more data transfers.

The data transfer phase is used to transfer data between the master and the selected slaves. After a master initiates a data transfer cycle, it waits for the responding slave to acknowledge the transfer before finishing the cycle. The asynchronous transfer protocols of the VSB allow a slave to take as long as it needs to respond. An interlocked relationship exists between masters and slaves during the data transfer phase of the cycle. This relationship starts when the master asserts the data strobe and ends after the responding slave acknowledges the transfer and all participating slaves indicate that they are ready to participate in a new cycle.

Participating slaves release an open collector line to high to indicate that they are done with the current data transfer and are ready to participate in a new one. This allows the active master to perform broadcast operations. During a broadcast operation, the data that the active master or the responding slave drives on the data lines is captured by participating slaves.

For example, after caching a data item, the system contains two copies of it, one in global memory and one in private cache memory. To ensure that these two copies are identical, the cache controller can monitor all VSB cycles, and either invalidate the cached copy or update it when it detects a write cycle to the global copy of the data. However, this operation should be completed before the transfer that caused it is terminated, so that the cache controller is ready to monitor a new one. This requirement is handled by the WAIT* line, which can be maintained low by such a cache controller, to prevent the active master from terminating the current transfer until the cache has been updated.

In addition to acknowledging the data transfer, the responding slave informs the master whether the data transferred is cacheable. For example, a memory board might signal the master that the data transferred is cacheable, allowing it, as well as participating slaves to indeed do so. On the other hand, an I/O controller might indicate that the data is not cacheable, preventing the master from caching data that is read from a status register. This simplifies the design of cache based architectures that do not require a memory management unit.

Central to the data transfer capabilities of the VSB is the support of dynamic bus sizing. Dynamic bus sizing describes the ability of onboard logic to automatically adjust the size of the data transfer and the number of cycles that are required to transfer a data item, to the basic data transfer capabilities

of the responding board. The ability to perform dynamic bus sizing allows programmers to store information without regard for the memory organization of the system on which the software is run. The system hardware itself is then given the task of adjusting the size of the transfer to the physical constraints of the system.

The VSB protocols support dynamic bus sizing as follows:

The responding slave acknowledges the address broadcast phase by identifying itself as a single-byte, double-byte, or quad-byte slave. This allows the active master to determine which data lines are driven with valid data during the data transfer phase of read cycles. In addition, the SLAVE size information is made available to onboard logic, allowing it to perform additional cycles until all the required byte locations have been accessed.

During some write cycles, the active master drives multiple data lines with the same data bytes to accommodate the three allowed slave sizes. This operation is sometimes referred to as data replication. When the active master initiates a write cycle it has no way of knowing whether the SLAVE that will respond to the cycle is a single-, double-, or quad-byte slave. When appropriate, the active master places the same data bytes on multiple data lines to accommodate the various types of slaves. This ensures that some data will be transferred during the first data transfer even if multiple transfers are required to complete the cycle.

When a master terminates the cycle, slaves acknowledge this termination by establishing the inter cycle state of bus signals. This inter cycle state informs an active master of the existence (or lack) of slaves in the system.

Data Transfer Bus Cycles

The protocols of the VSB define two basic types of cycles that are used to transfer data: the single-transfer cycle and the block-transfer cycle. Another type of cycle, the address-only cycle, is used to extend the capabilities of masters during the address broadcast phase. Each of these types of cycles can be used to access byte locations in any of the three address spaces: system address space, alternate address space and I/O address space. In addition, each of these cycles can be made indivisible. An additional type of cycle is the interrupt-acknowledge cycle which is used by masters and slaves to coordinate the service of interrupts.

The address-only cycle is the only cycle on the data transfer bus that is not used to transfer data. The active master starts the address-only cycle by executing the address broadcast phase. After its address broadcast is acknowledged in the proper manner, the master terminates the cycle without executing any data transfers. Address-only cycles can be used to enhance performance by allowing a CPU board to broadcast an address before it has determined whether that address might select a slave on the VSB or not. Broadcasting the address in this fashion allows slaves to decode the address concurrently with the CPU board.

The single-transfer cycle consists of an address broadcast followed by one data transfer. During a single-transfer write cycle the master stores data in byte locations provided by slaves. During single-transfer read cycles the master retrieves data from byte locations provided by slaves.

The block-transfer cycle allows the master to provide a single address, and then transfer a block of data without providing additional addresses. It is similar to a string of single-transfer cycles. The difference is that the master only broadcasts the initial address once, at the beginning of the cycle, and then proceeds to execute as many data transfers as are required. When a master initiates a block-transfer cycle, the selected slaves latch some part of the address into an onboard address counter and acknowledge the address broadcast. The master then starts the data transfer. After the slaves acknowledge the data transfer the master terminates it and starts another data transfer without releasing the bus. The selected slaves use their onboard address counter to generate the address for each data transfer.

The limit is set on the number of bytes that a master is allowed to transfer during a block-transfer cycle. However, to simplify the design of slaves and to protect against loss of data, a block-transfer cycle cannot cross a board boundary. Therefore, slaves are required to inform the active master when they can no longer perform data transfers without receiving another address. For example, a slave that only latches the lower eight address lines (requiring only an 8-bit counter) will signal the end of its self addressing range when a 256-byte boundary is crossed. On the other hand, a 32-bit address counter on a slave imposes no limit on the size of the transfer. However, such a slave is still required to signal the end of its self addressing range after it has accessed the last byte location it provides onboard.

The ability of the slave to dynamically, and on a transfer by transfer basis, inform the master whether it can process a data access without a new address promotes system configurability. A CPU board can always attempt to perform the faster block transfer cycles. In such a case, a responding slave that does not have block transfer capability informs the master during the first data transfer that it needs a new address. The master then is required to issue a new address, effectively switching to a single transfer cycle.

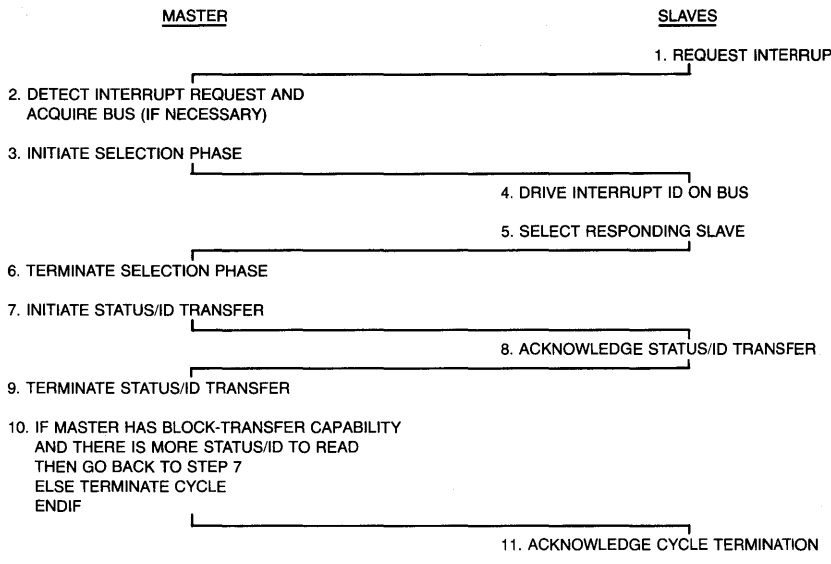
Interrupts In The VSB

The VSB defines protocols that allow slaves to request interrupt service from a master which, in turn, responds to these requests. Any system that has interrupt capability includes software routines that are called interrupt service routines. The interrupt protocol of the VSB defines how the slave transfers status or identification information to the master. This information is used by onboard logic to invoke an interrupt service routine.

The VSB protocols do not limit the amount of the status or identification information that can be transferred in the course of a single cycle. The VSB defines two basic interrupt capabilities for masters and slaves:

a. In response to an interrupt request, the master initiates read cycles to predefined addresses to determine which slave has requested an interrupt. These slaves include a STATUS/ID register, and release their contribution to the interrupt request line during the read cycle that accesses their STATUS/ID register. The master might then execute additional cycles to read addi-

FIGURE 3 — General Flow Of An Interrupt-Acknowledge Cycle



tional status or identification information from the slave. When responding to an interrupt request by polling, a master executes either a single-transfer or a block-transfer read cycle, depending on its capabilities and on the capabilities of the responding slave.

b. In response to an interrupt request, the master initiates, and slaves participate in an interrupt-acknowledge cycle. Each slave is assigned a unique interrupt ID which is used to prioritize multiple interrupts. This ID is a 7-bit code which is a combination of a 3-bit geographical slot address and a 4-bit user defined priority code.

The interrupt-acknowledge cycle is initiated by a master in response to an interrupt request from a slave. The cycle involves two types of slaves. Contending slaves are those who have an interrupt request pending and participate in the cycle. The responding slave is the one that transfers its status or identification information to the master. During the interrupt-acknowledge cycle, all contending slaves drive an interrupt priority code on the bus. This code is a combination of the geographical address of the board that is supplied by the backplane, and a priority code that is supplied by user defined onboard logic. The priority code is used to determine which one of the contending slaves will respond to the cycle.

When a master responds to an interrupt request it reads status or identification information from the slave. As described above, the master reads this information in the course of an interrupt-acknowledge cycle or by polling the slave. The general flow of the interrupt-acknowledge cycle is shown in Figure 3. The cycle starts when a slave requests an interrupt service. The master that services interrupts acquires the bus and initiates an interrupt-acknowledge cycle which is comprised of three distinct phases.

The first phase is the selection phase. It starts when the master initiates an interrupt-acknowledge cycle and ends after the responding slave is determined (steps 3–6 in Figure 3). During this phase the master can determine whether the interrupt was requested by a slave that can indeed participate in the interrupt-acknowledge cycle. If it was not, then the master terminates the cycle immediately following the selection phase, and initiates read cycles to poll the various devices to identify the one that requested an interrupt.

In case the master determines that the interrupting device can participate in an interrupt-acknowledge cycle, it proceeds to the second phase, the STATUS/ID transfer phase, reading the status or identification information from the slave that has been selected to respond to the cycle. The STATUS/ID transfer phase starts after the responding slave is selected, and ends after all of the STATUS/ID information has been transferred (steps 7–9). The third phase is the termination phase of the interrupt-acknowledge cycle (steps 9–11).

There are three types of interrupt-acknowledge cycles. During each status or identification transfer phase of quad-byte interrupt-acknowledge cycles the master retrieves four, two, or one bytes of the available status or identification information.

THE VSB ARBITRATION BUS

The arbitration bus is the second of the two sub-buses defined in the VSB specificaiton. It allows an arbiter modules and/or requester modules to coordinate the use of the data transfer bus. The VSB defines two arbitration methods — a serial/daisy-chained arbitration method and a parallel/distributed arbitration method. These arbitration methods provide protocols to implement an array of sub-system architectures. Using the serial arbitration method, a designer can implement a single master sub-system that include a single processor board requiring access to large amounts of memory. This method could be used to integrate a system that gives priority in accessing the bus to primary master who, when it can, assigns the bus to other secondary masters. At the end of the spectrum, a multiprocessing sub-system can be implemented using the parallel arbitration method.

SERIAL ARBITRATION

There is one and only one active arbiter in the serial arbitration scheme, and it is always located in slot 1. (An arbiter is not required in the parallel arbitration method). Serial requesters interact with the arbiter during the serial arbitration sequence to coordinate the use of the bus. Two types of serial requesters are defined: a release-when-done requester and a release-on-request requester. The release-when-done requester releases control of the bus when its associated master no longer needs the bus.

The release-on-request requester does not release the bus after its associated master no longer needs it unless some other requester requests to use it. It monitors the bus request line, releasing the bus only if another bus request is pending. This method reduces the number of arbitrations initiated by a processor board which is generating a large percentage of the bus traffic.

Due to the characteristics of the daisy-chain, serial requesters that are installed in the lower numbered slots of the backplane are granted the bus before those that are installed in the higher numbered slots. If a board that is installed in a low numbered slot generates a substantial amount of bus traffic, it might, in some cases, prevent boards that are installed in higher numbered slots from ever being granted the bus.

For example, consider a system that includes three processor boards: CPU A in slot 1, CPU B in slot 2, and CPU C in slot 3. In some cases all three request the bus at the same time. When the arbiter grants use of the bus, CPU A detects a low level on its bus grant input signal and starts using the bus. After CPU A releases the bus and the arbiter issues a new bus grant, CPU A passes it down the daisy chain. CPU B detects this grant and starts using the bus. While CPU B uses the bus, CPU A drives its bus request low again. After CPU B releases the bus and the arbiter grants the bus again, CPU A, being first to see the grant, assumes control of the bus. In some extreme cases CPU C will never be granted the bus.

To protect against this and to promote fairness in accessing the bus, the VSB specification recommends to design serial requesters that have just released the bus, to refrain from requesting the bus, until after they detect a high level on the bus request line. This assures that boards that reside further down the daisy-chain have equal access to the bus.

PARALLEL ARBITRATION

In the parallel arbitration method, the requester that is associated with the active master initiates an arbitration cycle when a request for the bus is pending, but only after its associated master no longer needs the bus. This cycle is used to select the master that will be granted use of the bus. At the end of the cycle the requester whose arbitration priority is the highest becomes the active.

Each of the six slots of the VSB backplane is assigned a unique slot address. This address can be read by the board that is installed in the slot. The VSB specification defines the use of the slot address for two purposes: (1) it forms part of an interrupt priority code that is used during in the course of the interrupt acknowledge cycle and, (2) it forms part of the arbitration priority code that is used during a parallel arbitration cycle. The slot ID can also be used to set global board variables such as the base address of a memory board.

The arbitration method that is used is automatically selected during the power-up sequence of the VSB subsystem, depending on the capabilities of the system's requesters. A system might be configured with a mixture of serial and parallel requesters. The decision whether the system will operate in the serial or in the parallel arbitration mode is determined during the power-up sequence.

In addition, when the parallel arbitration mode is selected, the initial active requester is selected in the course of the power-up sequence. This is necessary since the parallel arbitration mechanism requires that there always be an active requester. This is because bus mastership can only be transferred in the course of an arbitration cycle, which can only be initiated by an active requester.

VERSAbus Specification Summary

VERSAmodule boards are interconnected in a system using the VERSAbus interconnect standard. The high-speed VERSAbus interconnect is characterized by asynchronous operation supporting direct memory addressing and true multiprocessor operation. Unlike other popular bus structures, VERSAbus architecture does not limit the number or types of processors that can be used in multiprocessing applications. The number of "bus masters" or main processor boards is limited only by the number of card slots in the particular VERSAbus backplane being used. Furthermore, several lines within the VERSAbus structure enhance system reliability and integrity by providing for efficient self-diagnosis . . . resulting in minimum system downtime.

The VERSAbus, as developed by Motorola, addresses the limitations of existing bus structures and meets the needs of state-of-the-art microprocessor systems. It has been designed with special attention to the following objectives:

- To provide a comprehensive basis for microprocessor systems capable of supporting a wide range of architectures from 8- to 32-bit data paths with up to 5-MHz data transfer rates
- To provide adequate addressing range and control for large-scale systems
- To provide for system architectures involving multiple processors
- To provide sufficient flexibility to exploit the latest technologies without sacrificing ease of use to the designers of future microprocessor-based systems

AN INDUSTRY STANDARD BUS

The VERSAbus specification is an industry standard through the activities of IEEE Standards Committee P970.

A detailed VERSAbus Specification Manual may be ordered from the Motorola Literature Distribution Center (Part #M68KVBS/D4*)

VERSAbus Signal Groupings

The VERSAbus consists of four groups of signal lines called "buses" (see Figure 1).

- Data Transfer Bus (DTB)
  - — Masters and slaves transfer data over the Data Transfer Bus (DTB) using the data pathways and associated control signals.
- DTB Arbitration Bus
  - — The arbitration bus is used to assure the smooth transfer of control between DTB masters, and guarantees that only one master controls the DTB at any given time.
- Priority Interrupt Bus
  - — Through this bus interrupters can request interruption of normal bus activity and can be serviced by an interrupt handler. Interrupt requests can be prioritized into a maximum of several levels.
- Utility Bus
  - — The categories of system timing, control, and diagnostics are grouped into the area of utilities. These functions include clock lines, initialization, and system test, among others.

Each VERSAbus based system has a designated "System Controller." The controller function can contain the following functional modules:

- DTB Arbiter
- Emergency DTB Requester
- Power Up/Power Down Master
- Power Monitor (for ac clock and ac fail driver)
- System Clock Driver
- System Reset Driver
- System Test Controller

In any VERSAbus configuration, only one each of the above functional modules exist. The number one board slot is designated the System Controller position because the user will typically provide four modules (DTB arbiter, emergency requester, power up/down master, and system clock driver) on this board. The system reset and system test controller are commonly connected to an operator control panel, and may be located elsewhere. The power monitor is interfaced to the incoming AC source and may also be located remotely.

*Revision D4 was current at the time of printing of this document

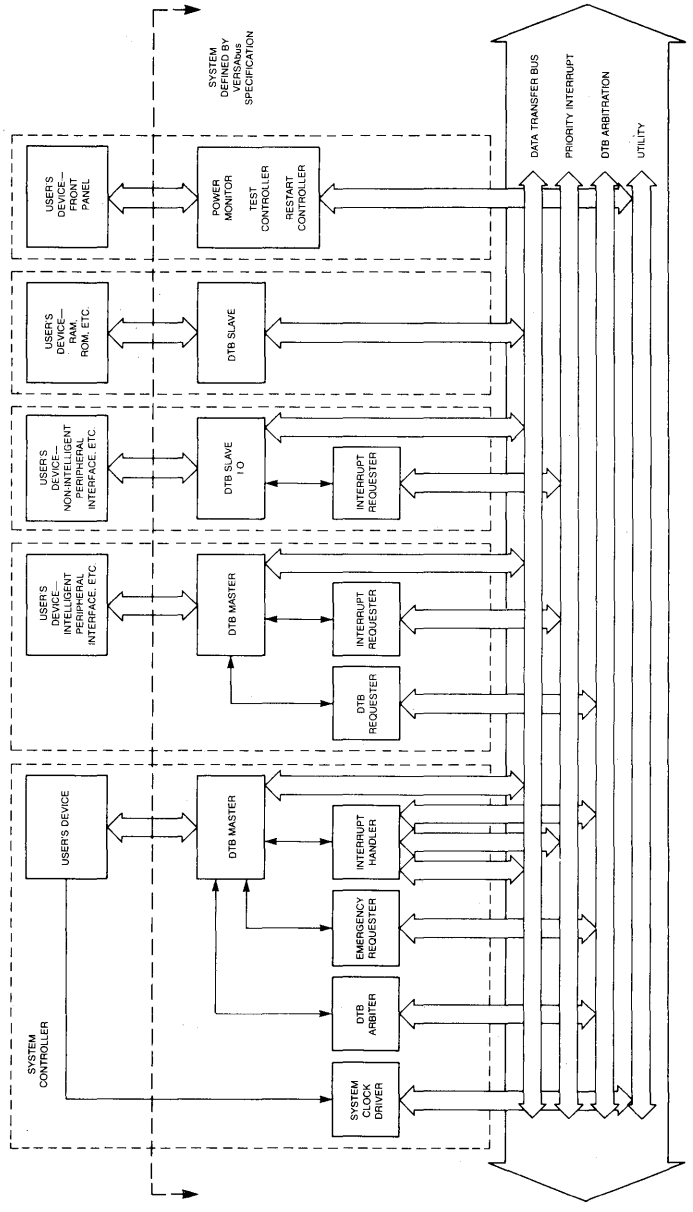


FIGURE 1 — VERSAbus Signal Groupings

VERSAbus Signal Line Terminology

Signal Line Category	Signal Line Mnemonics (1)	Terminology		
		Output	Input	Asterisk
Bus Lines (Three-state)	A01*–A31* D00*–D31* APARITY0*–APARITY1* DPARITY0*–DPARITY3* AM0*–AM7* TEST0*–TEST1* WRITE*, LWORD*	Drive X Drive X high Drive X low Place valid X Remove X Release X	Receive X driven high Receive X driven low Receive X	Indicates a low level Equals a logic 1
Drive X defines the point at which the three-state drivers are enabled. Place valid X defines the point at which the levels on the bus are valid. Remove X defines the point at which the levels on the bus are invalid. Release X defines the point at which the three-state drivers are no longer enabled.				
Strobe Lines (Three-state)	AS* DS0* DS1*	Drive X to low Drive X to high	Receive X driven to low Receive X driven to high	Indicates the information on the strobed bus is valid on the falling edge of the strobe line.
Strobe Response Lines (Open Collector)	DTACK* BERR*	Drive X to low Release X to high	Receive X driven to low Receive X high	Indicates the strobe response is valid on the falling edge of the signal line.
Shared Lines (Open Collector)	IRQ1*–IRQ7* BR0*–BR4* SYSFAIL*	Hold X low Release X	Detect X low Detect X high (only if no drivers holding line low)	Indicates this line is activated in the low state.

Bus Arbitration

VERSAbus provides simple and effective means to allocate the bus mastership among several processors/controllers. Due to the diverse nature of the potential bus masters (multiple processors, intelligent peripheral controllers, DMA controllers, emergency handlers, etc.), true multilevel bus arbitration is provided.

The VERSAbus subsystem for the Arbitration Bus consists of:

- One Data Transfer Bus (DTB) Arbiter
- One or More Data Transfer Bus (DTB) Requester(s)
- One Emergency Requester (optional)

Priority Interrupt

Multiple processor systems require a much more sophisticated interrupt handling structure than single processor systems. A bus designed to support such systems must have a flexible interrupt subsystem protocol.

Multiple processor interrupt subsystems can be divided into two categories:

- Single handler systems with a supervisory processor which receives and services all bus interrupts
- Distributed systems with two or more processors which receive and service bus interrupts

The single handler system architecture may be easier to understand because of its similarity to single processor systems. Any system which has interrupt capability must have a set of interrupt servicing routines in its executive software. Each of the routines may be thought of as a task which is activated by an interrupt. If the system has a real-time executive, these interrupt routines would then operate as tasks under this executive.

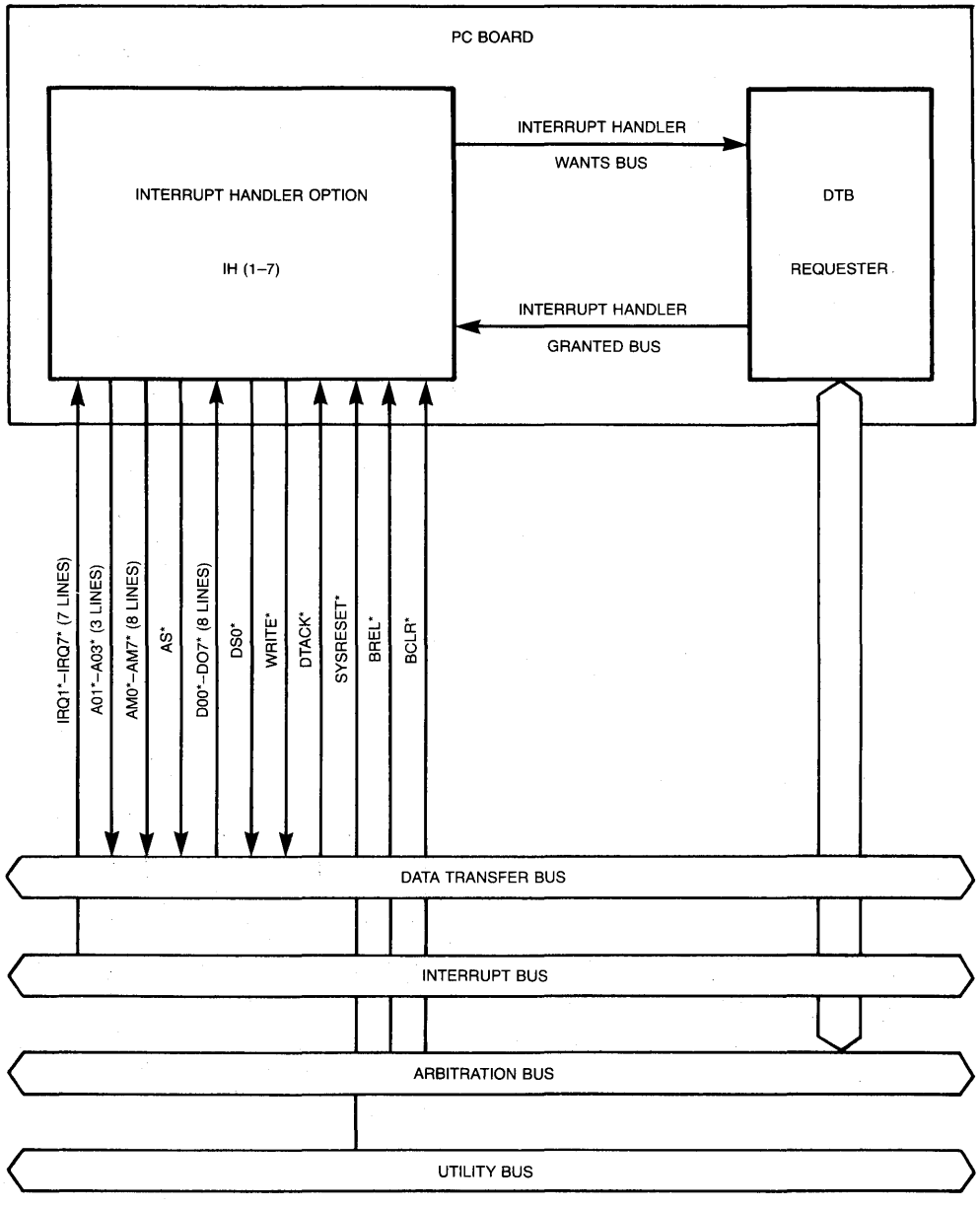
In a single processor or single handler system, the executive software and all the interrupt routines are executed by one processor.

The interrupt bus consists of seven interrupt request signal lines and one daisy-chain line:

IRQ1*	IRQ5*
IRQ2*	IRQ6*
IRQ3*	IRQ7*
IRQ4*	ACKIN*/ACKOUT*

Each interrupt request line may be driven low by an interrupter to request an interrupt. In a single handler system, these interrupt request lines are prioritized, with IRQ7* having the highest priority.

IH(1–7) Interrupt Handler Signal Lines



VERSAbus Utilities

Utility lines provide periodic timing signals, support time-of-day function, allow AC zero crossing indication, and provide start-up and testing capability for the VERSAbus. Utility lines include:

System Clock	(SYSCLK)
AC Clock	(ACCLK)
AC Fail	(ACFAIL*)
System Reset	(SYSRESET*)
System Test	(TEST0*, TEST1*, SYSFAIL*)

Two clock sources are available on the VERSAbus backplane: A system clock and a line frequency AC clock. These clocks are utility signals and have no fixed phase relationship with other VERSAbus timing. The system clock is an independent, non-gated, fixed frequency, 16 MHz signal which can be used to generate on-board delays or timing functions. The AC clock is a 50 or 60 Hz signal derived from the power supply line frequency. It can be used as a clock source for time-of-day generation, or to detect line frequency zero-point crossings.

VERSAbus allows several options for system initialization and diagnostics through a system reset, two test lines, and a system fail signal. On system power-on, the two test lines allow the system elements to enter four different test modes.

- Enter EXEC Immediate (no test required)
- Enter Debug Mode
- Long Test (no time limit) Then Enter EXEC
- Short Test ($\leq$ 2 seconds) Then Enter EXEC

Upon completion of the specified testing (if any), the system can go into normal operating mode if the functional tests show no failure. If a failure is encountered, the system fail line remains activated.

The system fail line can also be activated at any time during normal operation as the result of a detected system failure.

To allow safe system shutdown, an ac fail signal is also provided. This signal gives an early indication of an impending dc power failure and allows a graceful shutdown to be initiated.

Electrical Considerations

Power in a VERSAbus system is distributed on the backplane as regulated direct current (dc) voltages. The available voltages are described as follows.

+5 Vdc is the main logic level and normally has the largest associated current requirement. The bulk of the system circuitry — including TTL logic, MOS microprocessors, and memories — requires this voltage.

±12Vdc represent the auxiliary digital logic supplies. They supply the needs for MOS memories and I/O circuitry requiring multiple voltages. They may also be used for analog purposes. A −5 Vdc bias voltage and a −5.2 Vdc ECL voltage may also be derived from −12 Vdc, as needed. These supplies normally have lower current requirements than the +5 Vdc.

+5 Vdc standby is used for distributing battery backup power. The standby voltage is maintained during system power loss to sustain memory and time-of-day clocks. If the user is not concerned with power fail protection, this supply line should be supported by the normal +5 Vdc supply.

±15 Vdc voltages are intended for analog specifications. These voltages may be used directly by VERSAboards, or further regulated on-board where required. These voltage sources should be very carefully regulated with respect to ac ripple and noise filtering. Care should be taken in their use to avoid superimposing voltage variations on these lines due to varying loads or noise.

Specifications

Power Distribution	Regulated dc voltages available on VERSAbus:
Main logic power supply	+5 Vdc
Primary auxiliary supply	± 12 Vdc
Standby power	± 5 V STDBY
Analog application supply	± 15 V
Ground Distribution	Combined +5/+12/−12 Vdc ground return Separate ± 15 Vdc ground return
Signal Distance on Backplane	18" maximum
Backplane/VERSAboard Dimensional Requirements	
Board Spacing	PCB: 0.900 minimum WWB: 1.4 minimum
Board Thickness	0.062 ± 0.005 inch
Component Lead Length	Component leads protruding through back of VERSAboards not to exceed 0.100 inch
Component Height	Component height on front of each VERSAboard not to exceed 0.50 inch
Board Warpage	Maximum allowable VERSAboard warpage is 0.125 inch

VERSAbus Driver Specifications

Driver Type	Parameters	Min.	Max.	Unit	Test Condition
Totem-pole	Low state (V_{OL})		0.55	V	Sink 64 mA for terminated line
(High current)	High state (V_{OH})	2.0		V	Source 15 mA terminated
		2.4		V	Source 3 mA line
Totem-pole	Low state (V_{OL})		0.5	V	Sink 8 mA for unterminated line
(Low current)	High state (V_{OH})	2.7		V	Source 400 μA for unterminated line
Three-state	Low state (V_{OL})		0.55	V	Sink 64 mA for terminated line
	High state (V_{OH})	2.0		V	Source 15 mA terminated
		2.4		V	Source 3 mA line
	Off-state output current (I_{OZ})		± 50	μA	2.4 V or 0.5 V applied
Open Collector	Low state (V_{OL})		0.7	V	Sink 40 mA
	High state output current (I_{OH})		50	μA	5.0 V applied

VERSAbus Receiver Specifications

Parameter	Min.	Max.	Unit	Test Condition
Low state input voltage (V_{IL})		0.8	V	
High state input voltage (V_{IH})	2.0		V	
Low state input current (I_{IL})		−400	μA	Input voltage = 0.5 V
High state input current (I_{IH})		*50	μA	Input voltage = 2.7 V

*High state input current I_{IH} should be limited to 20 μA for low current totem-pole drive lines. (20 μA represents a standard LS TTL input.)

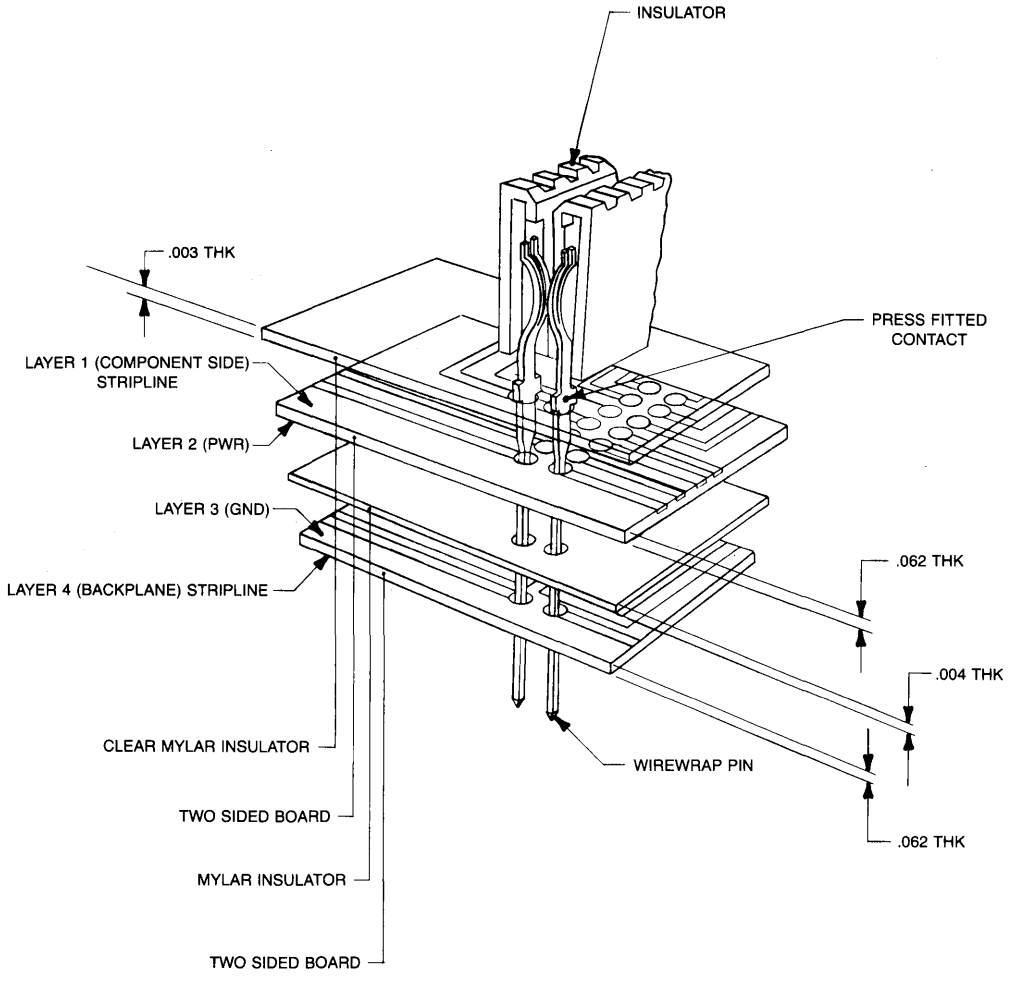
Mechanical Considerations

Backplane Construction Techniques

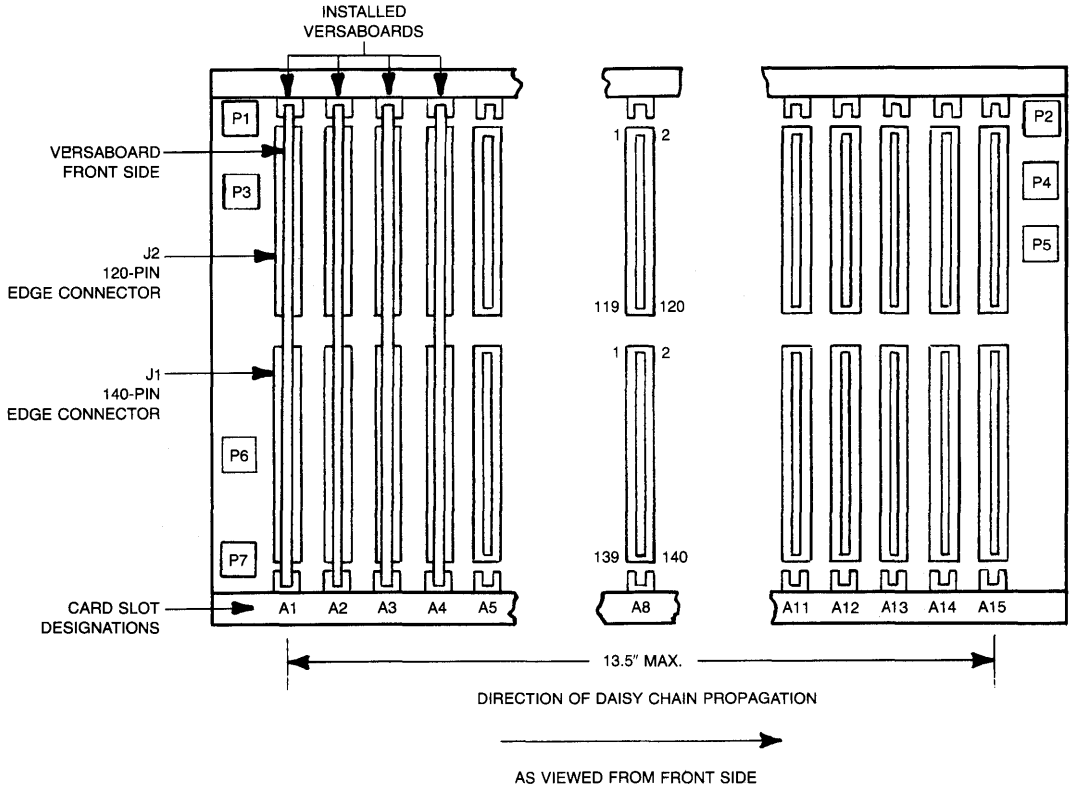
Many backplane construction techniques are available to the designer. Backplane construction can be of a multilayer laminated or unlaminated design. Following is a description of a multiple layer design composed of two discrete, two-sided, glass epoxy boards separated by a mylar insulator with a mylar clear top layer. The two boards and mylar insulators are held together, utilizing high force press fit socket contacts. This process provides a gas-tight connection between the contact and the plated-through hole in each board. Each side or layer of a board provides signal conductors or a voltage/ground plane. The first and fourth layers of the backplane contain the signal conductors. The second layer provides the voltage plane, and the third layer is the ground plane. Contact pins are press fitted into the two boards and the mylar insulators, forming the four-layer backplane. This process relies upon the backplane to provide the structural rigidity of the edge connector. Plastic insulators are then inserted over the contacts to form edge connectors.

A second possible backplane construction technique would be to laminate the two boards. This lamination process would bond the two boards and insulators together. Then edge connectors are inserted into the backplane and soldered.

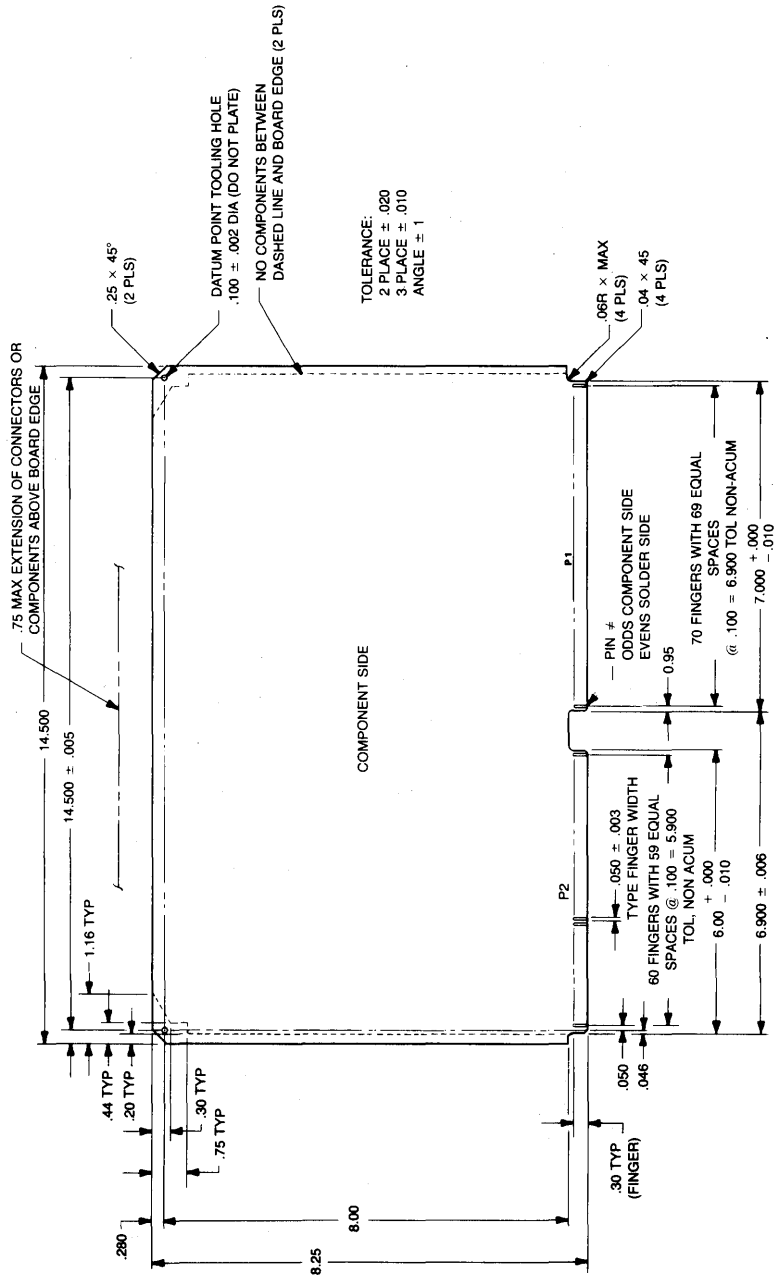
Typical Multilayer Backplane/PCB Construction Technique



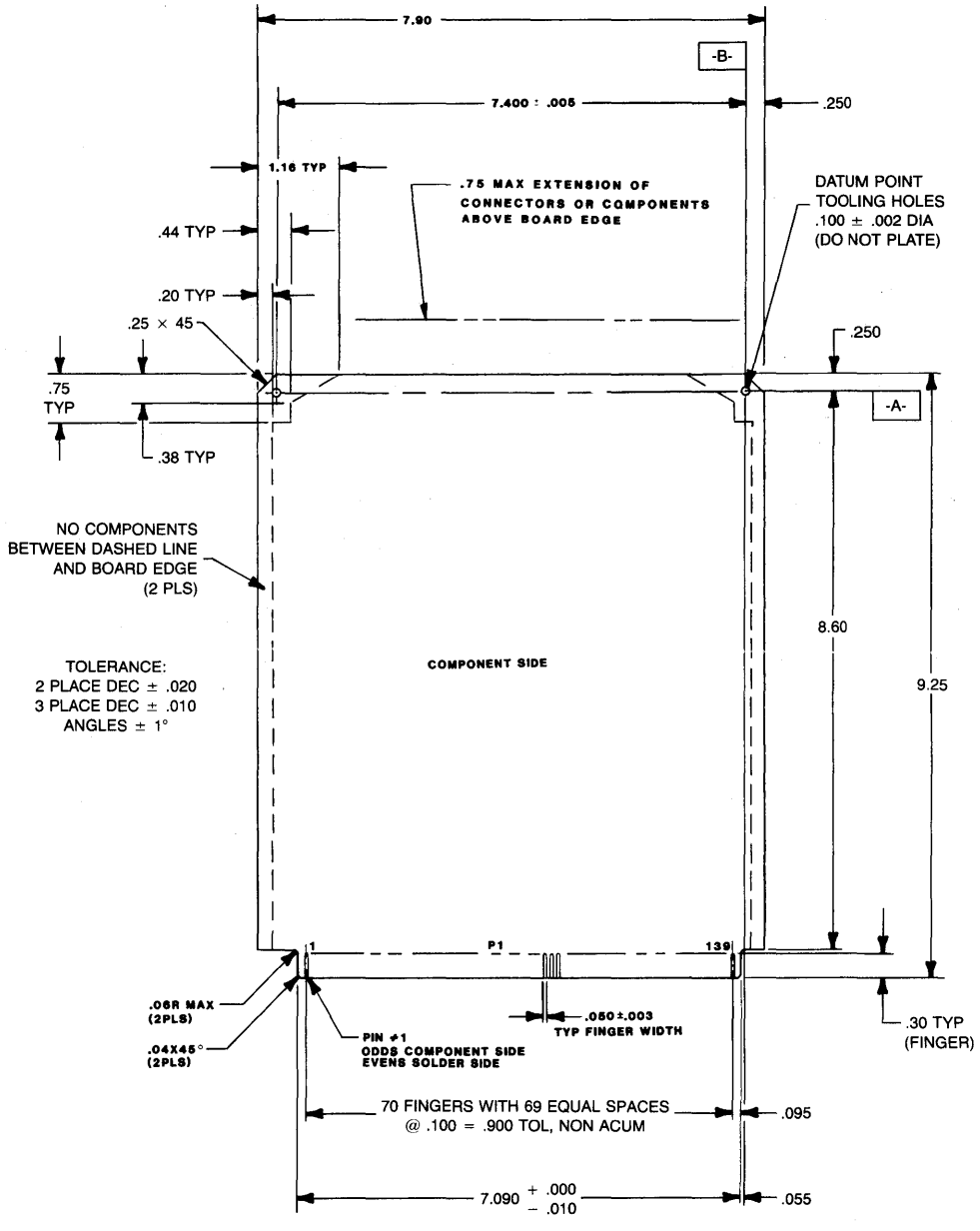
Backplane Reference Designations and Pin Numbering Standard



Standard Size VERSAboard



Half Size VERSAboard



I/O Channel Specification Summary

PHILOSOPHY

Hardware of most computer systems can be partitioned into two major parts: the application-independent part and the application-dependent part.

The application-independent portion typically consists of the CPU, RAM, ROM, timers, and certain diagnostic-related I/O ports. This portion forms a "core" system around which additional hardware is added to meet the specific applications requirements.

The application-dependent portion typically consists of various special purpose I/O devices. In an industrial control application, these may be A/D converters, stepper motor controllers, etc. In a data processing application, these may be printer interfaces, disk interfaces, etc.

The challenge which must be met to produce the most cost-effective system is to produce a low cost "core" system along with modular "add-on" I/O devices which provide only the specific I/O required.

The purpose of the I/O Channel is to provide a communication path though which the "core" system can communicate with its "add-on" I/O devices.

The I/O Channel provides the following features:

- 12-bit Address Bus
- 8-bit Bidirectional Data Bus
- Asynchronous Operation
- Up to 2-megabyte Transfer Rate
- Four Interrupt Lines
- Reset Line
- 4-MHz Free Running Clock Line

OBJECTIVES

The I/O Channel is an interfacing system whose primary purpose is to provide a high speed data path between I/O slave devices (slaves) and a core system (master). The system has been conceived with the following objectives:

- To allow the master to perform read and write operations to a slave device without disturbing internal activities of other slave devices
- Specify the electrical and mechanical constraints upon the design of the master and slave devices
- Specify protocols that define interactions between the master and slave devices
- Provide terminology and definitions that describe I/O Channel operation

TYPICAL SYSTEM CONFIGURATION

Figure 1 illustrates how a system might be configured using a ribbon cable bus I/O Channel. The bus master is typically a computer, but may also include a DMA controller for transferring blocks of data to or from a slave device at high speed.

As shown in Figure 1, there are two basic types of slaves on the I/O Channel. Slave Printed Circuit (PC) boards have 64-pin ribbon cable connectors and plug into a 64-pin connector which is crimped onto the ribbon cable. The preferred form factor for these boards is the DIN standard. These boards are powered through their 64-pin connectors and may draw +5 V and ±12 V from the bus.

When the slave PC boards are located at too great a distance from the master, power may be provided by a source close to their location. Figure 2 illustrates how this might be done.

The second type of slave is called a subsystem slave. It can be identified by the fact that it includes its own power supply. (It is usually housed in some sort of enclosure.)

A subsystem slave has two 50-pin ribbon cable connectors on its interface panel. In addition, it provides access to an internally-mounted terminator board to allow it to be configured as the last slave on the cable (terminators enabled).

FIGURE 1–Typical I/O Channel Configuration

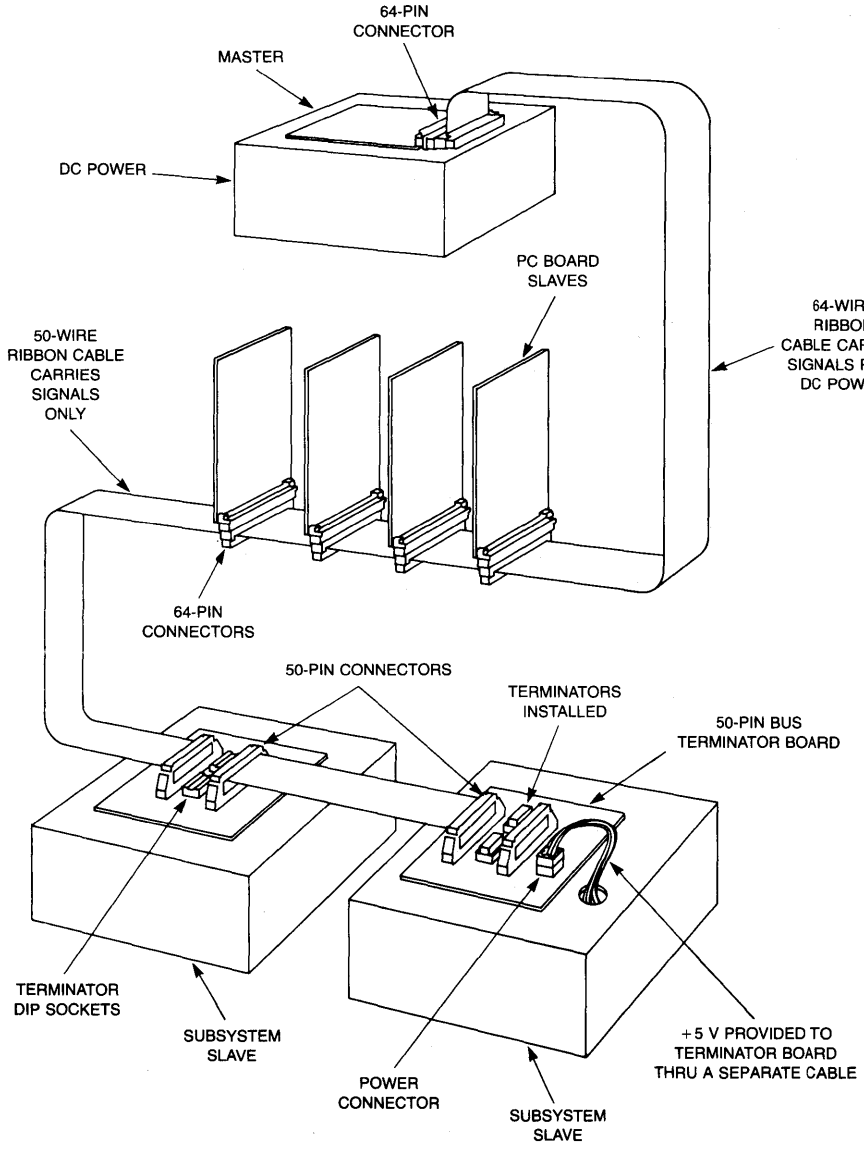
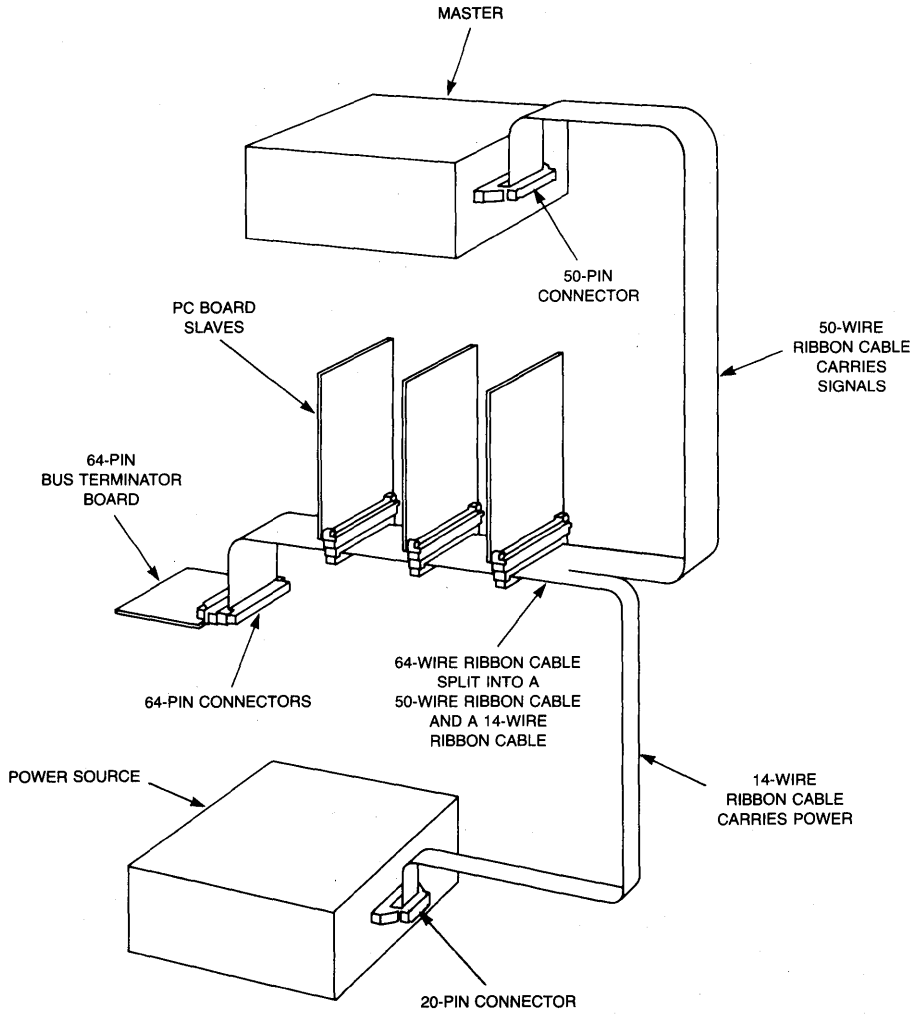


FIGURE 2–Alternate Power Wiring Configuration



I/O CHANNEL SIGNAL LINES

The following identify the I/O Channel signal lines:

A0–A11	Address channel (bits 0–11)
D0–D7	Data channel (bits 0–7)
WT*	Write
STB*	Strobe
XACK*	Transfer Acknowledge
CLK	Clock
IORES*	Input Output Reset
INT1*–INT4*	Interrupt (lines 1–4)

DATA TRANSFER PROTOCOL

All data transfers on the I/O Channel are between the master and a slave, and are initiated by the master. All data transfers are asynchronous and rely on two interlocked signal lines, STB* and XACK*. STB* is generated by the master and initiates a data transfer. XACK* is generated by the addressed slave to indicate that the data transfer has been acknowledged.

Interrupts

Slaves may interrupt the master by driving low one of the four prioritized interrupt lines (INT1*–INT4*). INT1* is the lowest priority interrupt. INT4* is the highest priority interrupt.

Any number of slaves can generate an interrupt at any given time. However, when a slave drives an interrupt line low, it must continue to drive the interrupt line low until the master acknowledges the interrupt. Any slave that is capable of generating an interrupt must also contain at least one status byte of one to eight bits in length that is readable by the master at a pre-determined address. The state of the status byte indicates whether or not that slave has a pending interrupt, and can also contain additional information about the cause of the interrupt or the action required to service or clear the interrupt.

There are three ways in which interrupts may be cleared on the I/O Channel:

a. Reset clears interrupt
b. Read clears interrupt
c. Write clears interrupt

Reset Clears Interrupt — In this mode, all interrupts that are to be removed by a reset must be cleared within ten microseconds after the IORES* line on the I/O Channel goes low. This is an initialization function. No record of past interrupt status is provided after initialization.

Read Clears Interrupt — In this mode, when the master reads data from a specific address of the interrupting slave (the addressed location may be the status byte), the slave must stop driving the particular interrupt line low (INT1*–INT4*) within three microseconds after receiving STB* low. This byte is read just like any other byte of a read operation, and must conform to all timing and handshake sequencing of a read cycle.

Write Clears Interrupt — In this mode, when the master writes to a specified location on the interrupting slave, the slave must stop driving the particular interrupt line (INT1*–INT4*) within three microseconds after receiving STB* low. This byte is written just like any other byte of a write operation, and must conform to all timing and handshake sequencing of a write cycle.

Utilities

Two additional signals — clock and reset — are included on the I/O Channel.

Clock — The CLK line is a free-running, 4-MHz (nominal) signal that may be used for miscellaneous timing by slaves. There is no relationship between the CLK line and any other timing on the I/O Channel.

Input/Output Reset — The IORES* line, when driven low by the master on a power-up reset circuit, is used to place slaves into a pre-determined state.

TERMINATION NETWORKS

The master is always located at one end of the cable and provides terminations for INT1*–INT4* and XACK*. Provision for terminating all other lines of the I/O Channel should be provided at the slave end of the cable for ribbon lengths greater than 18 inches to provide terminations for the address, data, clock, and reset lines. The IORES* line should also be pulled high by the master (via a 4.7K Ω ±5% resistor to +5 Vdc). This keeps the IORES* line from floating in non-terminated systems.

I/O CHANNEL RIBBON CABLE CHARACTERISTICS

Parameter	Ribbon Cable Specification
Center spacing	0.050 inch
Conductor	Stranded copper
Conductor size	28 AWG
Stranding	7 × 36
Conductor quantity	50 or 64
Impedance	≥ 100 ohm
Capacitance (pF/ft)	≤ 20
Inductance (μH/ft)	≤ 0.25
(ohms/1000 ft @ 20°C)	≤ 70
Propagation delay (ns/ft)	≤ 1.5

The I/O Channel is designed primarily for ribbon cable interconnection, but is not restricted to ribbon cable. For example, a PC board might be used for interconnection if its characteristics matched those of the ribbon cable as shown below. The specifications which follow, along with the terminators, timing, and board-level loading restrictions, allow 16 slaves to communicate with a master over a distance of 12 feet on a ribbon cable.

Related Documentation

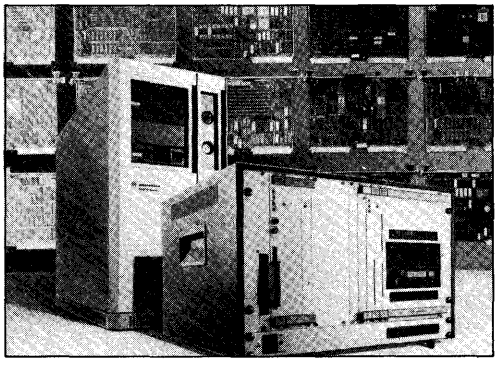
For additional information pertaining to the I/O Channel, refer to the I/O Channel Specification Manual, Motorola publication number M68RIOCS.

PIN ASSIGNMENTS FOR I/O CHANNEL

The pin assignments for the I/O Channel are as follows:

Din Connector Pin	3M Connector Pin	Mnemonic	Din Connector Pin	3M Connector Pin	Mnemonic
C1	1	INT4*	A1	2	GROUND
C2	3	INT3*	A2	4	GROUND
C3	5	INT2*	A3	6	GROUND
C4	7	INT1*	A4	8	GROUND
C5	9	IORES*	A5	10	GROUND
C6	11	XACK*	A6	12	GROUND
C7	13	CLK	A7	14	GROUND
C8	15	(Reserved)	A8	16	GROUND
C9	17	(Reserved)	A9	18	GROUND
C10	19	(Reserved)	A10	20	GROUND
C11	21	GROUND	A11	22	A11
C12	23	A9	A12	24	A10
C13	25	A7	A13	26	A8
C14	27	A5	A14	28	A6
C15	29	A3	A15	30	A4
C16	31	A1	A16	32	A2
C17	33	A0	A17	34	GROUND
C18	35	STB*	A18	36	GROUND
C19	37	WT*	A19	38	GROUND
C20	39	GROUND	A20	40	D7
C21	41	D5	A21	42	D6
C22	43	D3	A22	44	D4
C23	45	D1	A23	46	D2
C24	47	D0	A24	48	GROUND
C25	49	GROUND	A25	50	GROUND
C26		−12 Volts	A26		−12 Volts
C27		(Reserved)	A27		(Reserved)
C28		+12 Volts	A28		+12 Volts
C29		+5 Volts	A29		+5 Volts
C30		+5 Volts	A30		+5 Volts
C31		GROUND	A31		GROUND
C32		GROUND	A32		GROUND

Note: Where 50-pin ribbon cable is used, the 14 power lines shown at the bottom of this table are not connected by the cable.



Customer Support

7

USER DOCUMENTATION
Microcomputer System Components

VME Delta Series MICROCOMPUTERS SOFTWARE

MC41903/D1	OFFICELAN User's Guide
MC41904/D1	OFFICELAN Administrator's Guide
MC41905/D1	OFFICELAN Programmers Guide
MC41998/D1	3270 Interactive Emulator Operator's Guide
MC41999/D1	3270 Interactive Emulator Administrator's Guide
MC43802/D1	3770 SNA Operator's Guide
MC43803/D1	3770 SNA Administrator's Guide
MC43805/D1	BSC/RJE Operator's Guide
MC43806/D1	BSC/RJE Administrator's Guide
MD41957/D1	Terminal and Printer Diagnostics User's Guide
MH44408/D1	TM220 Video Terminal Installation and User's Guide
MV43830/D1	Q-Chart User's Guide
MV44174/D1	Office Services User's Guide
MV44175/D1	Q-One Reference Manual
MV44176-1/D1	Q-One Word Processing Tutorial — Volume 1
MV44176-2/D1	Q-One Word Processing Tutorial — Volume 2
MV44177/D1	Business Assistant Technical Installation Manual
MV44178/D1	Business Assistant User's Guide
MV44182/D1	Q-File User's Guide
MV44192/D1	Q-Menu User's Guide and Reference Manual

VMEbus COMPATIBLE MODULES AND SOFTWARE

MAPHER/D1	Token Bus LAN Head End Remodulator User's Manual
MVMEBUG/D4	VMEbug Debugging Packages User's Manual
MVMEDOS/D3	VERSAdos to VME Hardware & Software Configuration User's Manual
MVMEMAPIF/D1	MicroMAP Host Resident User's Manual
MVMEMAPUM/D1	MicroMAP Software User's Manual
MVMESB/D2	The Parallel Sub System Bus of the IEC 821 Bus
MVMESCSIFW/D1	Small Computer Systems Interface (SCSI) Firmware User's Manual
MVMEXNSV68/D1	CMC XNS User's Manual (SYSTEM V/68)
MVME025/D1	MVME025 System Controller User's Manual
MVME050/D2	MVME050 System Controller Module and MVME701 I/O Transition Module User's Manual
MVME12XBUG/D1	MVME12x Debug Monitor User's Manual for the VMEsystem 1000 Family of Microcomputer Systems

VMEbus (cont.)

MVME101/D2	MVME101 MC68000 Monoboard Computer User's Manual
MVME101BUG/D2	MVME101bug Debug Package User's Manual
MVME110/D2	VMEmodule Monoboard Microcomputer User's Manual
MVME115BUGE/D2	MVME115 Debug Monitor User's Manual
MVME115M/D1	MVME115M VMEbus Compatible Module User's Manual
MVME117/D1	VMEmodule Monoboard Microcomputer User's Manual
MVME117BUG/D2	MVME117 Debug Monitor User's Manual
MVME120/D2	MVME120, MVME121, MVME122, MVME123 VMEbus Microprocessor Module User's Manual
MVME120BUG/D1	MVME120 Debug Monitor User's Manual
MVME121SYS/D1	MVME121-Core System and MVME319/320 Controllers for SYSTEM V/68
MVME130/D3	MVME130, MVME131 VMEmodule 32-Bit Monoboard Microcomputer User's Manual
MVME130BUG/D2	MVME130 Debug Monitor User's Manual
MVME130XT/D1	MVME130XT and MVME130XT-1 Microcomputer User's Manual
MVME131XT/D1	MVME131XT and MVME131XT-1 Microcomputer User's Manual
MVME133/D1	MVME133 VMEmodule 32-Bit Monoboard Microcomputer User's Manual
MVME133BUG/D1	MVME133BUG 133Bug Debugging Package User's Manual
MVME200/D2	MVME200/201 64K/256K Byte Dynamic Memory Module User's Manual
MVME202/D1	MVME202/222 512K/1M/2M Dynamic RAM User's Manual
MVME211/D1	VMEmodule RAM/ROM/EPROM Memory Module User's Manual
MVME215E/D1	MVME215-1, -2, -3 256Kb, 512Kb, 1Mb Static RAM Module User's Manual
MVME222/D1	MVME222-1, MVME222-2 1 & 2 Megabyte Dynamic RAM Module User's Manual
MVME225E/D1	MVME225-1, -2 1Mb/2Mb Dynamic RAM Memory Module User's Manual
MVME300/D2	MVME300 GPIB Controller With DMA User's Manual
MVME310/D1	VMEmodule Intelligent Controller For Custom Interfacing User's Manual

VMEbus (cont.)

MVME315/D1	MVME315 Intelligent Disk Controller User's Manual
MVME316/D1	VMEbus to Input/Output Channel Interface Module User's Manual
MVME319E/D1	MVME319 Intelligent Disk/Tape Controller User's Manual
MVME320/D2	VMEbus Disk Controller Module User's Manual
MVME330/D1	VMEmodule LAN Controller User's Manual
MVME331/D1	MVME331 6-Channel Serial Communications Module User's Manual
MVME332E/D1	MVME332 Intelligent Communications Controller User's Manual
MVME333/D1	MVME333 6-Channel Serial Communications Module User's Manual
MVME340/D1	MVME340 Interface/Timer Module User's Manual
MVME350/D1	MVME350 Streaming Tape Controller VMEmodule User's Manual
MVME390/D1	MVME390 Graphics Display Controller Module User's Manual
MVME702/D2	MVME702 Disk Interface User's Manual
MVME705E/D1	MVME705 6-Channel Serial Transceiver User's Manual
MVME707/D1	MVME707 RS-232 Serial Port Distribution Module User's Manual
MVME820/D1	MVME820/MVME821/MVME822 Winchester/Floppy Mass Storage Assemblies User's Manual
MVME83XKT/D1	MVME830-Series Mass Storage Expansion Kit User's Manual
MVME830/D1	MVME830-Series Mass Storage Assembly User's Manual
MVME834UP/D1	Streaming Tape Drive Installation Guide for the MVME830-Series Mass Storage Assembly
MVME920A/D1	MVME920A 20-Slot VMEbus Backplane User's Manual
MVME921A/D1	MVME921A 9-Slot VMEbus Backplane User's Manual
MVME924/D1	MVME924 3-Slot I/O Channel Backplane User's Manual
MVME935/D1	Remote I/O Channel Connector Module User's Manual
MVME940/D2	MVME940-1 Chassis/Card Cage, MVME910-3 Power Supply, MVME921 Backplane and MVME922 Backplane User's Manual
MVME941/D2	MVME941 Chassis/Card Cage, MVME921 Backplane and MVME922 Backplane User's Manual
MVME942/D2	MVME942 Chassis/Card Cage and MVME920 Backplane User's Manual
MVME943/D1	MVME943 Chassis User's Manual
MVME944/D1	MVME944 Chassis User's Manual
MVME945DP/D1	MVME945 115 V to 230 V Conversion Procedure

VMEbus (cont.)

MVME945CVR/D1	MVME945 Desktop Cover Kit Installation Instructions
MVME945IO/D1	MVME945 Chassis I/O Channel Expansion Kit Installations Instructions
MVME1121V/D2	VMEsystem 1121 Manual for VERSAdos
MVME1131U/D1	VMEsystem 1131 Manual for SYSTEM V/68
SYS1131RDOM/D1	SYS1131DVLP Resident Debug Operation Manual
SYS1131U33/D1	SYS1131U33 System Manual for SYSTEM V/68

VERSAbus COMPATIBLE MODULES AND SOFTWARE

M68KVBUG/D2	VERSAbus Debugging Package User's Manual
M68KVBUG2/D1	VERSAbug 2.n Debugging Package
M68KVBUG3/D1	VERSAbug 3 Debugging Package User's Manual
M68KVMCCC/D1	Installation Instructions for VERSAmodule Card Cage I/O Connector Kit
M68KVMCHD/D1	Installation Instructions for VERSAmodule Chassis Decorative Cover Kits
M68KVMCHDSS/D1	Installation Instructions for VERSAmodule Chassis Decorative Side-Skins Kit
M68KVMCHE/D1	Installation Instructions for VERSAmodule System Chassis Expansion Kit
M68KVMCHS/D1	Installation Instructions for VERSAmodule Chassis Slide Mount Kit
M68KVMCHU/D1	Installation Instructions for VERSAmodule System Unified Power Control
M68KVMMB851/D1	Memory Management Board User's Manual
M68KVMPM1/D1	VERSAmodule System Power Monitor Module User's Manual
M68KVMPS1/D1	VERSAmodule System Switching Power Supply User's Manual
M68KVMSIOC1/D1	Installation Instructions for Dual Port RS-232C Serial I/O Cable Assembly
M68KVMSIOC2/D1	M68KVMSIOC2 Dual Sync/Async RS-232C Cable Assembly User's Manual
M68KVM01/D1	VERSAmodule Monoboard Microcomputer User's Guide
M68KVM02/D1	VERSAmodule Monoboard Microcomputer User's Manual
M68KVM03/D1	VERSAmodule Monoboard Microcomputer User's Manual
M68KVM04/D1	VERSAmodule 32-Bit Monoboard Microcomputer User's Manual
M68KVM11/D3	M68KVM11-1/-2 256K/512K Byte Dynamic RAM Memory Module User's Manual
M68KVM12/D2	VERSAmodule 1024K/4096K Byte RAM User's Manual

VERSAbus (cont.)

M68KVM13/D1	VERSAmodule 1024K/4096K Byte Dynamic RAM W/RAMbus User's Manual
M68KVM20/D2	Floppy Disk Controller Module User's Manual
M68KVM22/D1	Storage Module Drive Disk Controller User's Manual
M68KVM30/D3	Multichannel Communications Module User's Manual
M68KVM31/D1	Eight-Channel Intelligent Communications Module User's Manual
M68KVM33/D1	VERSAmodule LAN Controller User's Manual
M68KWWM/D1	M68K Wirewrap Module Information Guide

I/O CHANNEL COMPATIBLE MODULES

MVEME400/D2	Dual RS-232C Serial Port Module User's Manual
MVME410/D2	MVME410 Dual Parallel Port Module User's Manual
MVME420/D1	SASI Adapter Module User's Manual
MVME600/D3	MVME600 Analog Input Module — MVME601 A/D Expander Module User's Manual
MVME605/D2	MVME605 Analog Output Module User's Manual
MVME610/D1	MVME610 AC Input Module User's Manual
MVME615/D1	MVME615/MVME616 AC Output Module User's Manual
MVME620/D1	MVME620 DC Input Module User's Manual
MVME625/D1	MVME625 DC Output Module User's Manual
MVME900/D1	MVME900 Series Equipment User's Manual
MVME935/D1	Remote I/O Channel Connector Module User's Manual
M68KRADDRV/D1	RAD1 Device Driver Software User's Manual
M68RAD1/D2	M68RAD1 Remote Intelligent Analog-to-Digital Conversion Module User's Manual
M68RIO1/D1	Remote Input/Output Module User's Manual
M68RIOCABL/D1	Installation Instructions for I/Omodule Card Cage Cable Assembly Kit
M68RIOCC1/D1	Installation Instructions for 5-Slot I/Omodule Card Cage Adapter Kit
M68RIOCCK/D1	Installation Instructions for I/O Card Cage Connector Kit
M68RSC1/D1	Remote Serial Conversion Module User's Manual
M68RWIN1/D1	Winchester Disk Controller User's Manual

SYSTEM V/68
(for licensed customers only)

M68KUNAG/D1	SYSTEM V/68 Administrator's Guide
M68KUNAM/D1	SYSTEM V/68 Administrator's Manual

SYSTEM V/68 (cont.)

M68KUNDP/D1	SYSTEM V/68 Document Processing Guide
M68KUNMSG/D1	SYSTEM V/68 Error Message Manual
M68KUNPG/D1	SYSTEM V/68 Programming Guide
M68KUNRD/D1	SYSTEM V/68 Release Description
M68KUNSTG/D1	SYSTEM V/68 Support Tools Guide
M68KUNTA/D2	SYSTEM V/68 Transition Aids
M68KUNUG/D1	SYSTEM V/68 User's Guide
M68KUNUM/D1	SYSTEM V/68 User's Manual

VERSAdos

M68KEDIT/D9	M68000 CRT Text Editor User's Manual
M68KIPCS/D3	M68000/IPC Command Channel Software Interface Reference Manual
M68KRIODRV/D1	RIO1 Device Driver Software User's Manual
M68KRMS68K/D11	M68000 Family Real-Time Multitasking Software User's Manual
M68KSYMBG/D4	SYMbug/A and DEbug Monitor Reference Manual
M68KSYSGEN/D8	System Generation Facility User's Manual
M68KVMSG/D4	VERSAdos Messages Reference Manual
M68KVOVER/D8	VERSAdos Overview
M68KVSF/D7	M68000 Family VERSAdos System Facilities Reference Manual
MVMEDOS/D1	VERSAdos to VME Hardware and Software Configuration User's Manual
RMS68KIO/D6	VERSAdos Data Management Services & Program Loader User's Manual

16-BIT LANGUAGES

M68KFFP/D1	68343 Fast Floating Point Reference Manual
M68KFORTRN/D3	M68000 Family Resident FORTRAN Compiler User's Manual
M68KLINK/D5	M68000 Family Linkage Editor User's Manual
M68KMASM/D7	M68000 Family Resident Structured Assembler Reference Manual
M68KPASC/D7	M68000 Family Resident Pascal User's Manual

CROSS PRODUCTS

M68KPTOM/D1	PDP-11 MACRO-11 M68000 Structured Assembly Language Translator User's Manual
M68KOSIM/D2	M68000 Simulator Reference Manual
M68KXASM/D3	M6800 Cross Macro Assembler Reference Manual
M68KXASM2/D1	M68000 Cross Macro Assembler Reference Manual (IBM 370)
M68KXLNKR2/D1	M68000 Cross Linkage Editor User's Manual
M68KXPASCL/D4	M68000 Cross Pascal Compiler User's Manual (Preliminary)
M68KXPASC2/D1	M68000 Cross Pascal Compiler User's Guide

FIELD AND FACTORY SERVICE

Consistent with its responsibility as a leading producer of 16/32-bit Microcomputer Boards and Systems, Motorola MCD offers its customers Field Service of the highest quality. This service is provided in North America by CSO (Customer Support Operation), a division of General Systems Group. CSO utilizes a support staff of more than 1500 to provide service at over 175 locations nationwide. CSO offers technical support by phone, on-site installation and maintenance, scheduled maintenance, exchange/repair services and spare parts sales. A quick response nationwide can be obtained by using the following phone numbers:

S/W and H/W Support	(602) 438-3100 or (800) 528-1908
Field Engineer Dispatching	(602) 438-3100 or (800) 528-1908
Exchanges/Spare Parts	(214) 241-1300 or (800) 222-5640 and at the tone dial 8285 or 0.
Repairs	(214) 241-1300 or (800) 222-5640 and at the tone dial 8313 or 0.
S/W or H/W Maintenance Contracts	(408) 864-4865
CSO Technical Training	(214) 241-7700

Applications engineering assistance is available through a Motorola salesman at any of the Motorola Microcomputer Division sales offices throughout North America. Answers to questions on the use of Motorola Microcomputer Division-supported hardware and software products may be obtained through use of the Motorola Customer Support Operation toll-free hotline: (800) 528-1908.

TECHNICAL TRAINING COURSES

Standard Courses

Motorola Technical Training courses are scheduled throughout the world, with courses in the United States, Canada, Mexico, Europe and Asia. The schedule is advertised periodically, and information is always available from the training headquarters in Phoenix. Currently offered courses are tabulated below.

MTT1	Basic MC6800 Family
MTT2	MC6801 Basic Course
MTT3	MC6809 Update
MTT5	MC6801 Update
MTT7	Understanding Microprocessor Basics MC68000 Minicourse on Cassette
MTT8	MC68000 16/32-bit Microprocessor Family
MTT15	RMS68K Kernel
MTT17	MCA Basic CAD Course
MTT18	SYSTEM V/68 (UNIX*)
MTT19	VERSAdos/VME System Design
MTT20	MC68020 Course, Advanced
MTT23	VERSAdos Software Development
MTT24	MC68HC11, HDS300, 'C' Compiler
MTT25	Fundamentals of MAP
MTT26	Implementing a MAP system

COURSES AT YOUR OWN LOCATION:

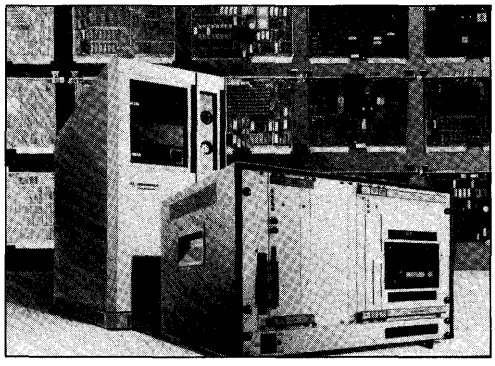
A special session of any Motorola Technical Training course may be held at your facility. This can be one of our standard courses or a course designed to fulfill your particular needs. For detailed information and scheduling, contact training headquarters.

MOTOROLA TECHNICAL TRAINING HEADQUARTERS

Telephone (602) 244-7126 or (602) 521-6274
Bill Johnson, Motorola Technical Training
Mail Drop: HW68, P.O. Box 2953, Phoenix, AZ 85062

Table of M68000 Family Devices for 16/32-Bit Microcomputer System Design

Device	Description
MC68120 Intelligent Peripheral Controller (IPC)	Serial communications interface with 21 parallel I/O lines, 128 bytes dual port RAM, programmable timer, 2K bytes ROM
MC68121	Same as MC68120 without ROM
MC68153 BUS Interrupter Module (BIM)	Fully programmable module interfaces four sources of interrupt requests to system bus
MC68172 VMEbus Controller (E-BUSCON)	The E-BUSCON performs VMEbus/local bus arbitration, VMEbus requests, bus transceiver control and guarantees VMEbus timing.
MC68173 VMSbus Controller (S-BUSCON)	The S-BUSCON interfaces the high-speed serial peripheral VMSbus to VMEbus.
MC68230 Parallel Interface/ Timer (PI/T)	Provides programmable parallel I/O and common timer requirements for microprocessor-based systems
MC68440 Dual Channel DMA Controller (DDMA)	Provides DMA support for MC68000 systems with two independent DMA channels
MC68450 DMA Controller (DMAC)	Provides DMA support for M68000 systems with four independent DMA channels
MC68451 Memory Management Unit (MMU)	Provides dynamic allocation, dynamic relocation, and resource protection for multiuser/multitask M68000 systems
MC68452 Bus Arbitration Module (BAM)	Provides control of microprocessor bus for multiple master systems
MC68454 Intelligent Multiple Disk Controller (IMDC)	Provides support for either floppy or hard disk drives
MC68486 Raster Memory Interface (RMI) MC68487 Raster Memory Controller (RMC)	The RMI and RMC comprise the Motorola Raster Memory System (RMS) which performs most of the functions required by a bit-mapped or object-oriented color graphics or alphanumeric display system.
MC68562 Dual Universal Serial Communication Controller (DUSCC)	The MC68562 controls two completely independent full duplex receiver/transmitter channels using any major communication protocol.
MC68590 Local Area Network Controller-Ethernet (LANCE)	Provides IEEE 802.3 Specification support and full M68000 interface
MC68653 Polynomial Generator/checker (PG/C)	Generates and checks error correcting codes
M68661 Enhanced Peripheral Communications Interface (EPCI)	Single channel serial I/O device which handles asynchronous and byte control protocols
MC68681 Dual Universal Asynchronous Receiver/ Transmitter (DUART)	Provides dual channel asynchronous protocol serial I/O control
MC68851 Paged Memory Management Unit (PMMU)	The MC68651 PMMU is a 32-bit memory manager which provides full support for a demand-paged virtual environment.
MC68881 Floating Point CoProcessor (FPCP)	The MC68881 FPCP provides full support for IEEE-specified (Revision 10.0) floating point high level math functions.
MC68901 Multi-function Peripheral (MFP)	Provides single channel USART, programmable interrupt controller,four timers and eight parallel I/O lines



Appendices

8

Table of Module Current Requirements

Part Number	Description	Current (Typical/Maximum) Requirements +5 Vdc	+12 Vdc	−12 Vdc	Comments
VERSAmodules					
M68KVM01A1	VERSAmodule Monoboard Microcomputer (32K RAM)	5.0 A/8.4 A	200 mA/500 mA	100 mA/160 mA	Fully populated
M68KVM01A2	VERSAmodule Monoboard Microcomputer (64K RAM)	4.0 A/8.4 A	200 mA/800 mA	100 mA/160 mA	Fully populated
M68KVM02-3	VERSAmodule Monoboard Microcomputer	5.5 A/6.4 A	45 mA/55 mA	35 mA/45 mA	Fully populated
M68KVM03	VERSAmodule Monoboard Microcomputer	6.8 A/8.2 A	45 mA/55 mA	35 mA/45 mA	Fully populated
M68KVM10-3	VERSAmodule 128K Byte RAM	2.9 A/3.5 A	148 mA/741 mA	7.3 mA/9.1 mA	±12 V currents are standby/ operating (maximum)
M68KVM11-1	VERSAmodule 256K Byte RAM	5.6 A/6.8 A	—	—	
M68KVM11-2	VERSAmodule 512K Byte RAM	5.6 A/6.8 A	—	—	
M68KVM20	VERSAmodule Floppy Disk Controller	3.5 A/4.2 A	110 mA/135 mA	70 mA/85 mA	
M68KVM21	VERSAmodule Universal Disk Controller	12 A/14.5 A	50 mA/75 mA	600 mA/750 mA	Includes two board set
M68KVM30	VERSAmodule Multi-Channel Communication	4.0 A/4.8 A	250 mA/300 mA	200 mA/250 mA	±12 Vdc does not include RS-232 port loads
M68VM33	VERSAmodule Ethernet Controller	4.0 A/4.8 A	600 mA/1.0 A	100 mA/200 mA	
M68KVM60	VERSAmodule Universal Intelligent Peripheral Controller	7.0 A/8.4 A	50 mA/75 mA	50 mA/75 mA	
M68KVM80-1	VERSAmodule Combination Memory, I/O, Time of Day Clock	4.25 A/5.1 A	55 mA/100 mA	40 mA/80 mA	Does not include ROM's
M68KVM80-4		4.68 A/5.6 A	55 mA/100 mA	40 mA/80 mA	Does not include ROM's, RAM active
VMEmodules					
MVME025	VMEmodule System Controller	0.4 A/0.5 A	—	—	
MVME050	VMEmodule System Controller	—/4.4 A	—/170 mA	—/42 mA	
MVME101	VMEmodule Monoboard Microcomputer	2.0 A/3.0 A	25 mA/50 mA	25 mA/50 mA	All eight sockets unpopulated. ±12 Vdc does not include RS-232 port loads.
MVME104	VMEmodule System Controller	—/3.2 A	—/8.0 mA	—/8.0 mA	
MVME105	VMEmodule System Controller	—/3.2 A	—/8.0 mA	—/8.0 mA	
MVME106	VMEmodule System Controller	—/3.2 A	—/8.0 mA	—/8.0 mA	
MVME107	VMEmodule Monoboard Microcomputer	3.2 A/—	—/8.0 A	—/8.0 A	
MVME110-1	VMEmodule Monoboard Microcomputer	2.1 A/2.4 A	25 mA/50 mA	25 mA/50 mA	All eight sockets unpopulated. ±12 Vdc does not include RS-232 port loads.
MVME115M	VMEmodule Monoboard Microcomputer	2.5 A/3.0 A	40 mA/50 mA	20 mA/30 mA	—
MVME117-3,-3FF,-4	VMEmodule Monoboard Microcomputer	3.9 A/4.4 A	50 mA/—	50 mA/—	
MVME121	VMEmodule Monoboard Microcomputer	4 A/4.45 A	17 mA/20 mA	12 mA/14 mA	
MVME123	VMEmodule Monoboard Microcomputer	4.3 A/4.65 A	17 mA/20 mA	12 mA/14 mA	
MVMEXTCAC-1,-2	VMEmodule Cache Accelerator Module	4.3 A/6.4 A	—	—	
MVME130,131	VMEmodule 32-Bit Monoboard Microcomputer	5 A/6 A	—/250 mA	—/250 mA	
MVME133A	VMEmodule 32-Bit Monoboard Microcomputer	5 A/7 A	—/250 mA	—/250 mA	

Table of Module Current Requirements (continued)

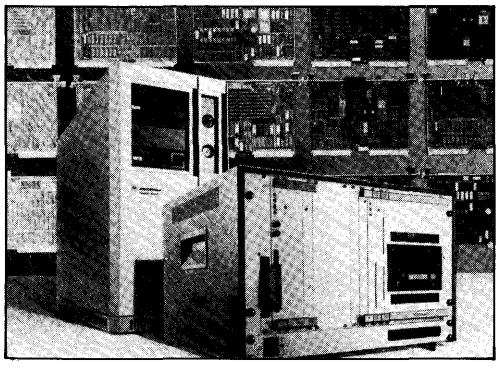
Part Number	Description	Current (Typical/Maximum) Requirements +5 Vdc	+12 Vdc	−12 Vdc	Comments
VMEmodules					
MVME200	VMEmodule 64K Byte Dynamic RAM	2.0 A/2.7 A	84 mA/500 mA	5.0 mA/6.5 mA	
MVME201	VMEmodule 256K Byte Dynamic RAM	2.5 A/3.3 A	—	—	
MVME202	VMEmodule 512K/1M/2M Byte Dynamic RAM	2.6 mA/3.0 mA	—	—	
MVME204-1,-2	VMEmodule 1Mb/2Mb Dynamic RAM	—/5 A	—	—	
MVME204-2F	VMEmodule 2Mb Dynamic RAM with VSB	—/5 A	—	—	
MVME210	VMEmodule Static RAM/ROM	0.9 A/1.4 A	30 mA/50 mA	3.9 mA/6.5 mA	All sixteen sockets unpopulated.
MVME211	VMEmodule Static RAM/ROM	1.0 A/1.7 A	—	—	Unpopulated, 2.5 mW for battery back up.
MVME214	VMEmodule Static RAM/ROM with VSB	—/4.5 A	—	—	
MVME215-3	VMEmodule 1Mb CMOS RAM	0.25 A/—	600 mA/—	—	
MVME225-2	VMEmodule 2Mb Dynamic RAM	2.0 A/3.5 A	—	—	
MVME300	VMEmodule GPIB Controller w/DMA	2.5 A/3.3 A	—	—	
MVME310	VMEmodule Univ. Intelligent Peripheral Controller w/WW	2.6 A/3.4 A	—	25 mA/50 mA	
MVME315	VMEmodule Univ. Intelligent Peripheral Controller w/SASI & FD	3.6 A/4.5 A	25 mA/50 mA	25 mA/50 mA	
MVME320	VMEmodule Winchester/ Floppy Disk Controller	2.6 A/3.0 A	20 mA/30 mA	20 mA/30 mA	
MVME320A	VMEmodule VMEbus Disk Controller	2.6 A/3.0 A	20 mA/30 mA	20 mA/30 mA	
MVME321	VMEmodule Intelligent Disk Controller	5.8 A/7.0 A	—/48 mA	—/48 mA	
MVME330	VMEmodule Ethernet Controller	3.8 A/4.6 A	600 mA/1.0 A	100 mA/200 mA	
MVME331	VMEmodule Six Channel Comm. Controller	—/1.5 A	—/150 mA	—/150 mA	
MVME332	VMEmodule Eight Channel Comm. Controller	2.6 A/3.2 A	—	—	
MVME333	VMEmodule Six Channel Comm. Controller w/DMA	—/1.5 A	—/150 mA	—/150 mA	
MVME335	VMEmodule Serial/Parallel I/O Module	1.3 A/2.1 A	15 mA/75 mA	55 mA/75 mA	
MVME340	VMEmodule Parallel Interface/Timer	3.2 A/—	—	—	
MVME350	VMEmodule Streaming Tape Controller	3.0 A/3.5 A	—	—	
MVME355	VMEmodule VMEbus ½ Inch Tape Controller	—/6.0 A	—	—	
MVME360	VMEmodule Disk Controller/ Formatter	—/3.0 A	—	—/500 mA	
MVME372	VMEmodule MAP Network Interface	4.2 A/4.5 A	25 mA/50 mA	25 mA/50 mA	
MVME372SET-1,-2,-3	VMEmodule MAP Network Interface	4.2 A/4.5 A	25 mA/50 mA	25 mA/50 mA	
MVME390A	VMEmodule Graphics Interface Module	6.0 A/—	—	—	

Table of Module Current Requirements (continued)

Part Number	Description	Current (Typical/Maximum) Requirements +5 Vdc	+12 Vdc	−12 Vdc	Comments
I/Omodules					
MVME400	Dual Serial Port I/Omodule	0.45 A/1.0 A	50 mA/100 mA	40 mA/80 mA	±12 Vdc does not include RS-232 port loads.
MVME410	Dual Parallel Port I/Omodule	0.76 A/1.0 A	—	—	
MVME420	SASI Peripheral Adapter I/Omodule	0.75 A/1.0 A	—	—	
MVME435	Magnetic Tape Adapter I/Omodule	1.8 A/2.4 A	—	—	
MVME600	Analog Input I/Omodule	0.65 A/0.75 A	4.5 mA/5.0 mA	26 mA/30 mA	
MVME601	Analog Input Extender I/Omodule	0.65 A/0.75 A	—	—	
MVME605	Analog Output I/Omodule	1.0 A/1.1 A	60 mA/60 mA	110 mA/110 mA	
MVME610	Opto Isolated 120 V/240 Vac I/Omodule	0.8 A/0.9 A	—	—	
MVME615/616	Opto Isolated 120 V/240 Vac Output I/Omodule	0.4 A/0.7 A	—	—	
MVME620	Opto Isolated 60 Vdc I/Omodule	0.8 A/0.95 A	—	—	
MVME625	Opto Isolated 60 Vdc Output I/Omodule	0.4 A/0.7 A	—	—	
M68RAD1	Remote Intelligent A/D Conversion I/Omodule	1.7 A/1.9 A	12 mA/15 mA	6.0 mA/10 mA	Unit requires +15 Vdc at 105 mA & −15 Vdc at 90 mA. Opt. dc/dc conv. req. 5 Vdc at 1 A.
M68RI01,-2	Remote Input/Output I/Omodule	0.76 A/1.5 A	—	—	Fully populated
M68RWIN1-1,-2	Winchester Disk Controller I/Omodule	4.0 A/6.0 A	—	*	*Also requires −5 Vdc or −8 to −20 Vdc at 90 mA max.

Table of Supplied Currents

VERSAmodule System Packaging		Current Provided +5 Vdc	+12 Vdc	−12 Vdc
MVMCH3-1	8-Slot Chassis with 400 Watt Power Supply	60 A	8 A	6 A
Eurocard System Packaging				
MVME910-3	200 W Plug-in Chassis Power Supply 90 V to 270 V	30 A	3 A	1 A
MVME940-1	19" Wide Chassis w/200 W Switching P/S, 9-Slot VMEbus Backplane, 2 each 5-slot I/O Channel Backplanes	30 A	3 A	1 A
MVME943-1	VMEbus Chassis Assy.	50 A	10 A	5 A
MVME944-1	VMEbus Chassis Assy.	50 A	10 A	5 A
MVME945-1	VMEbus Chassis Assy.	50 A	10 A	5 A
SYS1131_Y-331	VMEbus Modular System	28.9 A	5.1 A	4.6 A
SYS1131_Y-344	VMEbus Modular System	15.1 A	5.1 A	4.3 A



Index 9

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES