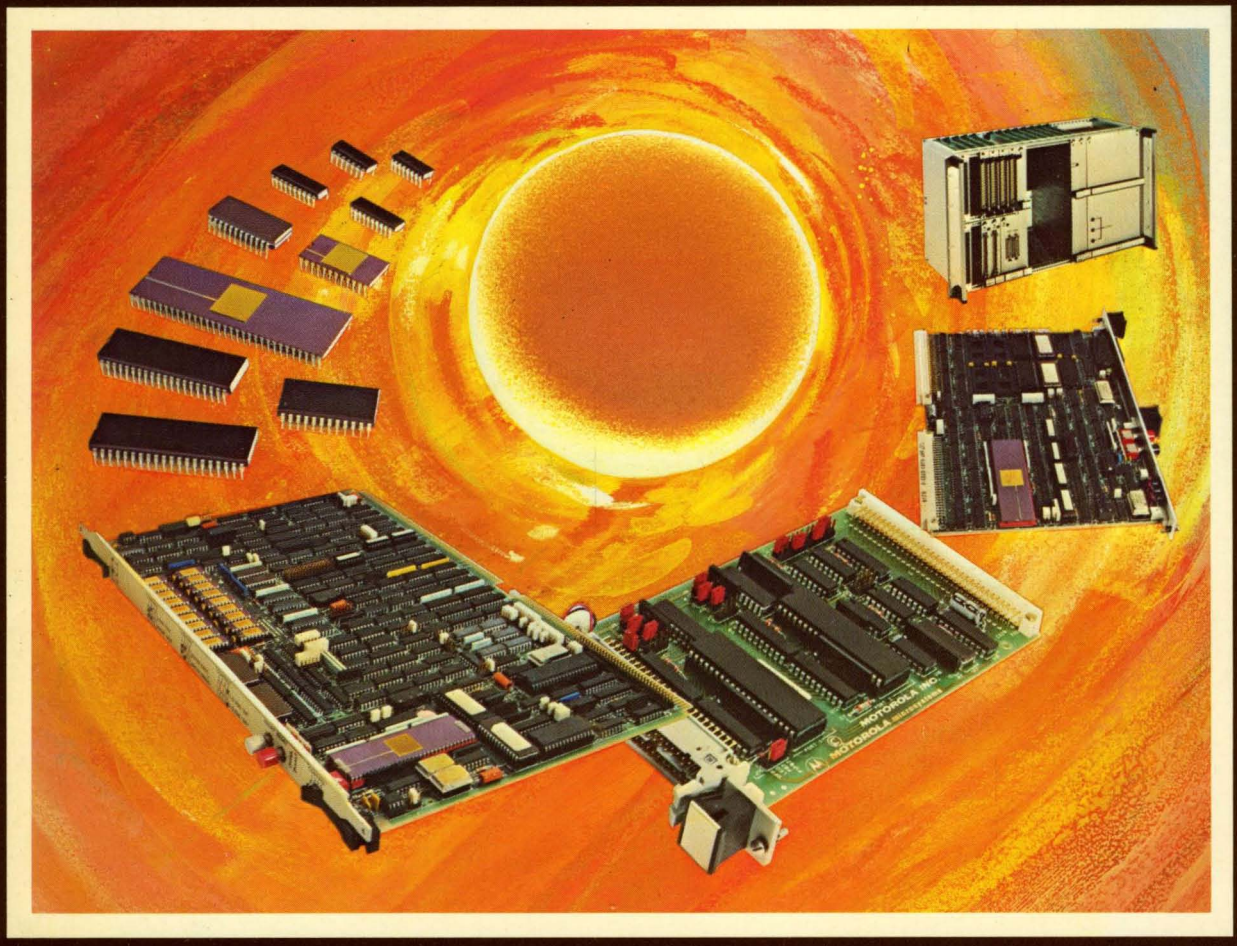
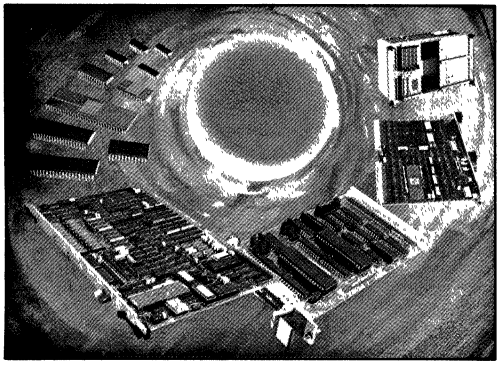
DL127

MOTOROLA INC.



16/32-BIT MICROCOMPUTER SYSTEM 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.

*Customer Return Authorization Number is available from Four-Phase Field Engineering. Call toll-free to your regional office: Eastern (800) 538-9832, Central (800) 538-9841, Western (800) 538-9800, and California Only (800) 662-9260.

MOTOROLA

16/32-BIT MICROCOMPUTER SYSTEM COMPONENTS

Prepared by
Technical Information Center

Preface

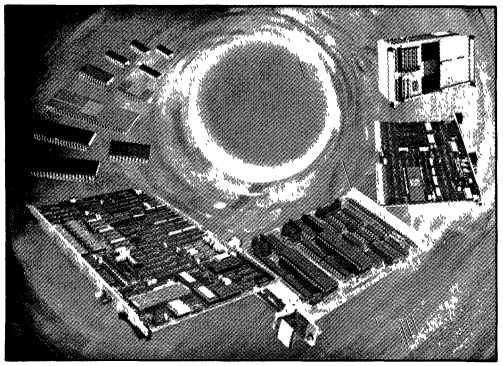
Following Motorola's tradition of serving current and future needs of the fast evolving electronics market, Motorola Microsystems offers a range of high-quality microcomputer systems, modular components, peripherals and associated hardware and software. Support for design and development or configuration and integration of OEM products for today's MC68000-, MC68008- and MC68010-based products is provided by the SYSTEM V/68 and VERSAdos Operating Systems and by development hardware/software tools, development instrumentation and board level system components. Support for tomorrow's MC68020-based OEM products will be assured by MC68020 hardware/software tools, MC68020 development instrumentation, board-level MC68020 system components and a multiprocessor operating system.

Today's OEM is offered a wide selection of microcomputer system functions implemented for Motorola's standard VERSAbus as VERSAmodules and for the international standard VMEbus as VMEmodules. A number of communications, peripheral control and industrial control functions implemented for Motorola's standard I/O Channel adjunct to VERSAbus and VMEbus are also available to the OEM for his products.

Complementing this broad range of board level, 16/32-bit microcomputer system components, Motorola Microsystems offers the OEM three categories of products which aid the design of microcomputer systems and shorten their development time. These are: 1. powerful host/development systems including the advanced VME/10 OEM Microcomputer System and the EXORmacs Development System running under the SYSTEM V/68 operating system or the VERSAdos Real-Time Multitasking Operating System, 2. development instrumentation including the HDS-400 Microprocessor Hardware/Software Development Station which is used with any of four host/operating system configurations to obtain real-time, 10 MHz, no-wait-state emulation of any member of the family of Motorola 16/32-bit microprocessors and including, also, the Bus State Analyzer which samples the system bus allowing operations and events to be recorded for analysis, and 3. software support in the form of operating systems, high-level languages support for development systems and debugging packages for VMEmodule and VERSAmodule monoboard microcomputers.

First Edition

Printed in U.S.A.



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

MVMEBUG
MVMEBUG1
MVMEBUG2

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

MVMEBUG
MVMEBUG1
MVMEBUG2

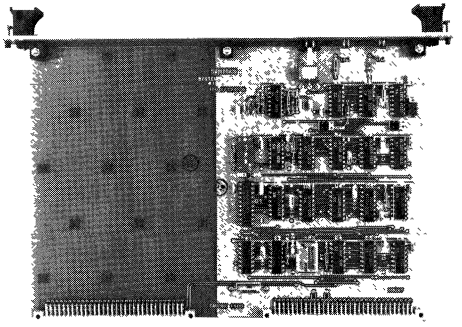
Ordering Information

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.
MVMEBUG2	Source and Relocatable Object Modules for the VMEbug system on VERSAdos Cartridge Disk for the EXORmacs Development System.* Includes User's Manual.
MVMEBUG/D2	MVMEbug Debugging Packages User's Manual

*The MVMEBUG EPROM set is copyrighted by Motorola and may be copied only under prior written agreement from Motorola MVMEBUG1 and MVMEBUG2 Sources are copyrighted and licensed by Motorola They may be obtained only under the required license agreement with Motorola

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, and is recommended in applications employing Motorola's MC68010 Monoboard Microcomputer, part number MVME115M. 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.

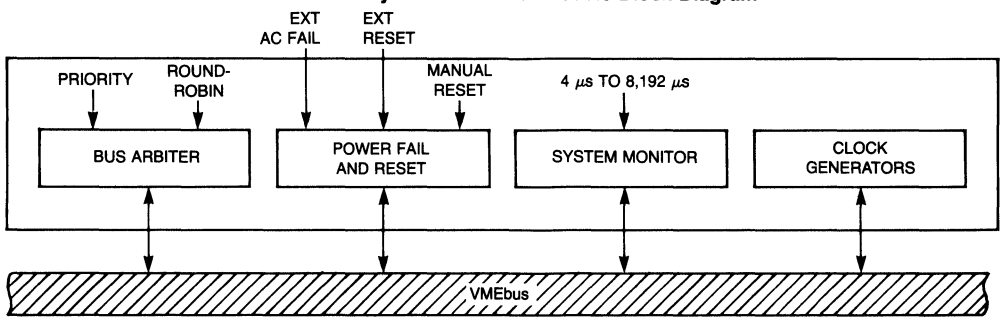
Both the system reset and ac fail lines are available at connector P2 on the system controller module which has logic for detecting the ac fail signal and for asserting the VMEbus ACFAIL* and SYSRESET* signals using proper timing. The module also has a header by means of which positive true or negative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

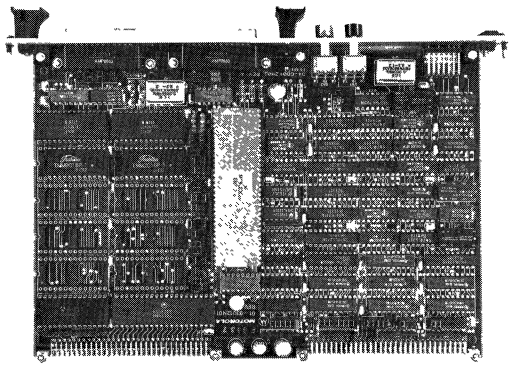
Ordering Information

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

1

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.

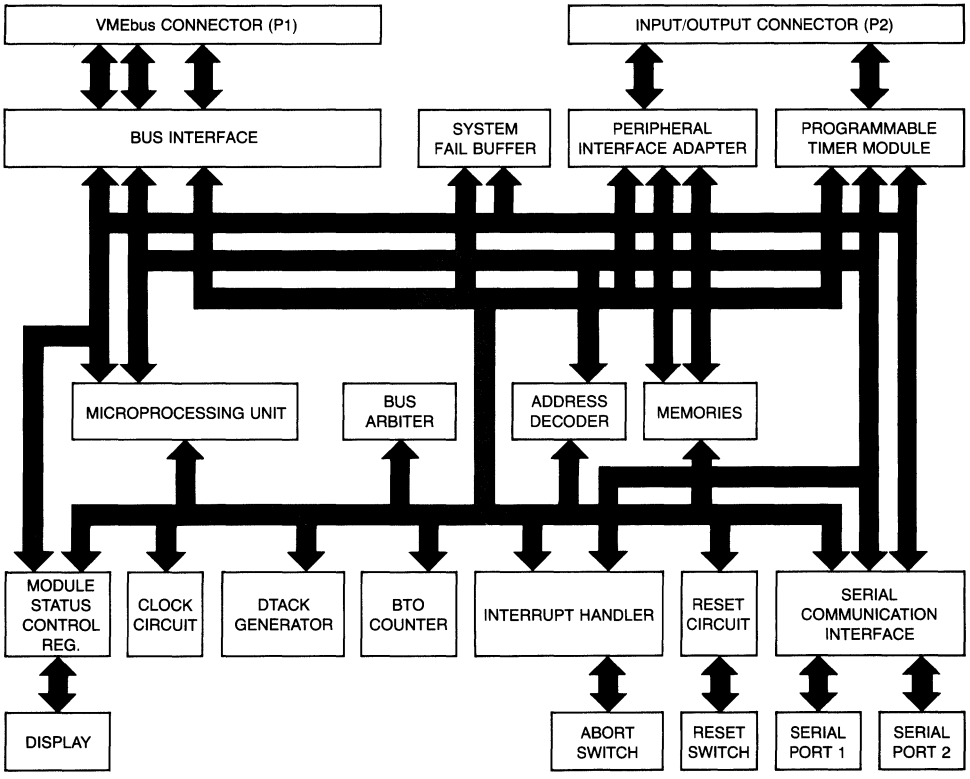
The MVME101 Monoboard Microcomputer in combination with a VMEmodule chassis, other VMEmodules and the VERSAdos Real-Time Operating System can provide a complete design environment which frees the system designer to develop the unique software/firmware required for an application. Figure 1 diagrams the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

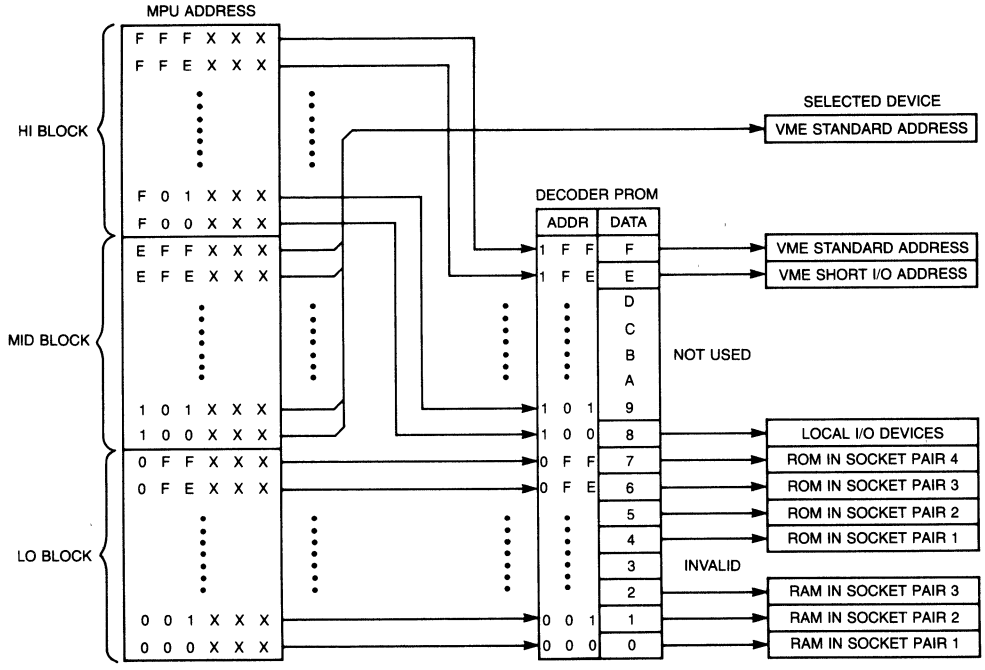
An array of eight 28-pin sockets connected as four pairs is provided on the module for user-supplied RAM or ROM. These can be any JEDEC standard byte-wide static memory in a 24- or 28-pin dual in-line package and of 2K, 4K, 8K, 16K or 32K size, of single +5.0 V operation, having high impedance (MOS) 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.

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/D1	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

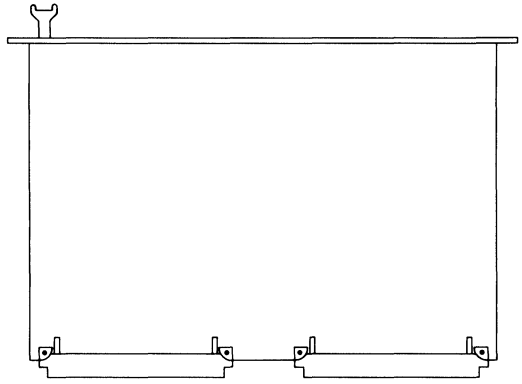
*When (CTRL-W) is used, the user can cause the output display to continue by entering any character.

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/D2	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

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 timeout 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.

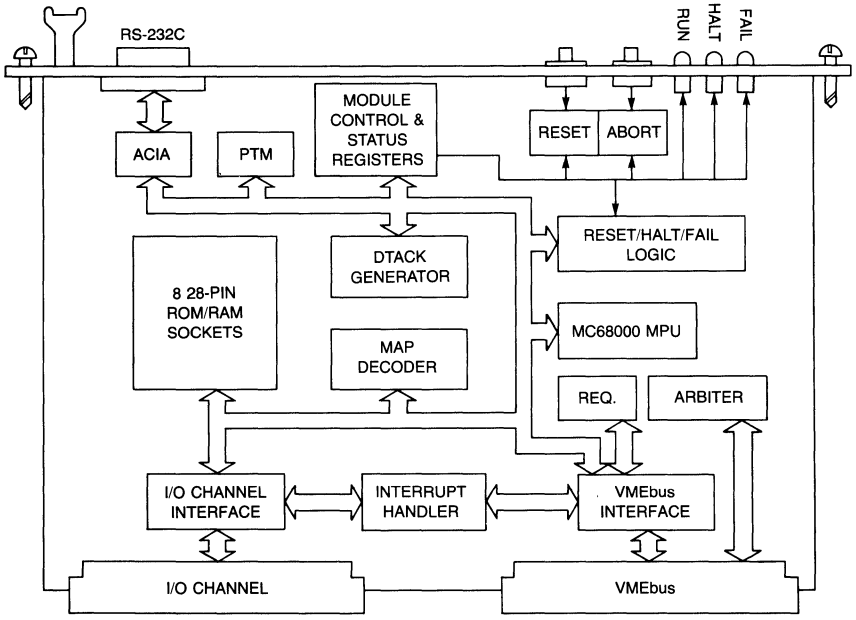
PROGRAMMABLE TIMER/COUNTER

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.

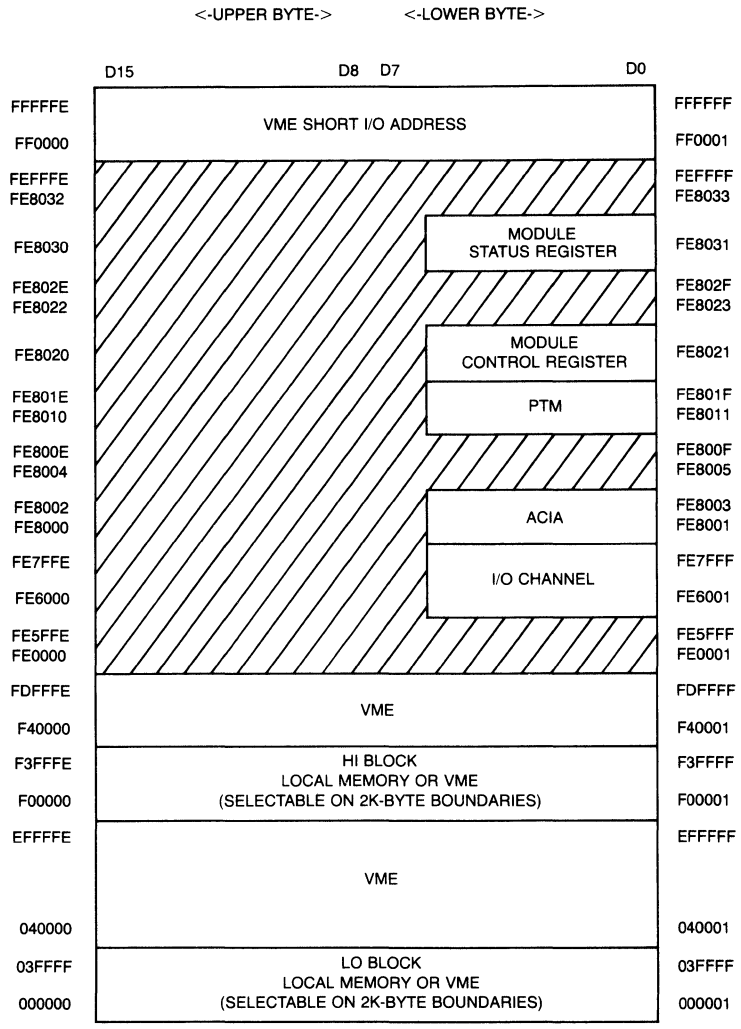
SOFTWARE/FIRMWARE SUPPORT

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	Double-high VME board
Height Depth Thickness	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 $\overline{\text{RTS}}$ 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 $\overline{\text{CTS}}$ 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/D1	VMEmodule Monoboard Microcomputer User's Manual

Related Documentation

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

Accessory Modules Include:

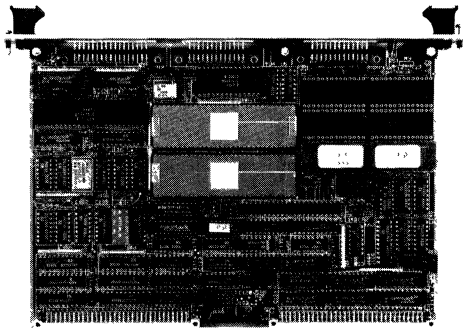
Part Number	Description
MVME200	64K Dynamic RAM VMEmodule
MVME201	256K Dynamic RAM VMEmodule
MVME210	Static RAM/ROM 1K, 2K, 4K, 8K, 16K x 8 VMEmodule
MVME211	Static RAM/PROM 1K, 2K, 4K, 8K x 8 + 5 V Standby for CMOS RAM VMEmodule
MVME300	GPIB Listener, Talker, Controller VMEmodule
MVME400	Dual Channel 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
MVME435	9-Track Magnetic Tape 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
M68RMS68K	Real-Time, Multitasking Kernel
M68KVDOS	VERSAdos Operating System

VMEmodule Monoboard Microcomputer



- MC68010 8 MHz MPU
- MC68451 Memory Management Unit
- Accepts up to 64K Bytes of On-Board ROM/PROM and eight Bytes of RAM Devices
- Provides for Battery Backup of CMOS RAM
- Two RS-232C Serial Ports
- Programmable 16-Bit Parallel Port
- Programmable 24-Bit Timer
- Four Level, Two Mode Bus Requester
- Handles Seven Interrupt Levels
- Full Operation Isolated from VMEbus (Multiprocessor Mode)
- Double Eurocard Form Factor
- VMEbus Compatible

The MVME115M Monoboard Microcomputer is a high performance processing module designed to function as a standalone microcomputer, as a single CPU in a VMEbus system or as an element in a multiprocessor VMEbus configuration having a system controller. This module features Motorola's MC68010 16-Bit Microcontroller and the MC68541 Memory Management Unit which provides address translation and access protection over the entire 16 megabyte range of the MPU.

The module can be used in applications ranging from standalone to dedicated control tasks. It can also be used in general purpose, disk-based multiprogramming/multitasking environments including high speed data processing.

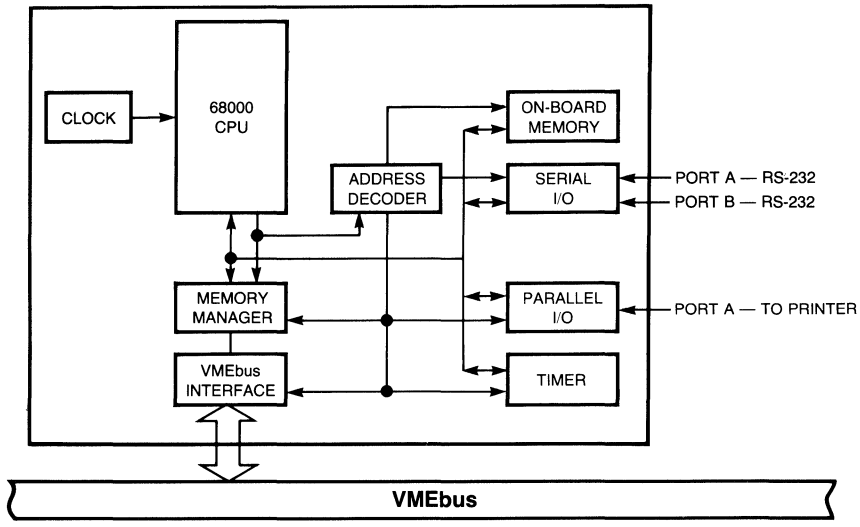
The module has provisions for configuring the data transfer bus requester, the interrupt requester and handler and memory type and speed. A multilayer printed circuit board design with internal power and ground planes is used providing a reliable industrial grade product. Figure 1 is a functional block diagram of the module.

MVME115M MPU

The module is shipped with an 8 MHz MC68010 MPU having the following features:

- Seven 32-bit data, address and stack registers
- 14 addressing modes
- 16 megabyte direct addressing range
- Virtual memory/machine support
- Memory mapped I/O
- 57 powerful instruction types
- High performance looping instructions
- 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

FIGURE 1 — MVME115M Block Diagram



MC68451 MEMORY MANAGEMENT UNIT

Some type of memory management mechanism is required for an operating system to insure the proper execution of user tasks in the system environment. The MC68451 Memory Management Unit (MMU) offers an operating system with the basic capabilities required for implementing memory management in an MC68010-based system. It offers the means for distinguishing between User and Supervisor memory spaces and for separating program and data spaces, for providing write protection for designated spaces and for translating and mapping logical addresses into the available physical space.

ON-BOARD MEMORY

The module has three pairs of sockets for 28-pin JEDEC standard byte-wide devices. These reside on a local bus for protection from VMEbus accesses. Headers provide for jumper accommodation of memory type, chip type/size and access time. Another header allows jumper selection of battery backup for operation with static RAM chips. Power is supplied from the +5 Vdc standby line on the VMEbus backplane.

MEMORY MAP

The module is shipped with an 82S123 decoder PROM which is programmed to decode the VMEbus address lines providing the selection of on-board resources shown in Table 1. This device is socketed to allow repositioning the module and its memory, serial and parallel I/O and timer devices in the address space.

SERIAL PORTS

Two RS-232C asynchronous serial ports are implemented on the module using an SC2681 DUART. This device provides two independent, full duplex, asynchronous transmitter/receiver channels which are independently programmable for operating mode, data format and a baud rate ranging from 50 to 38.4 kilobaud. The module has headers for independently configuring the serial ports as DCE or DTE and the status line of each port to support the configured mode, for generating DTACK when memory or a serial port is accessed and for configuring the SC2681 to generate interrupts.

PARALLEL PORT AND TIMER

The module utilizes an MC68230 Parallel Interface/Timer (PI/T) to provide a 16-bit wide parallel port (or dual 8-bit ports), a 24-bit timer and four handshake pins. Port modes include: bit I/O, unidirectional 8-bit or 16-bit I/O, and bidirectional 8-bit or 16-bit I/O. The handshake lines normally used in the bidirectional mode can also be used for interrupt generation. The module has headers for enabling each of the two 8-bit ports and for configuring them for input, output or for bidi-

TABLE 1 — MVME115M Address Map

000000 thru 000FFF	Onboard RAM (socket M0) 2 chips x (2K x 8 bits)
010000 thru F6FFFF	VMEbus memory address space
F70000 thru. F73FFF or F77FFF or F7FFFF	Onboard ROM or PROM (sockets M1 and M2) 4 chips x (4K x 8 bits) — (003A) 4 chips x (8K x 8 bits) — (003B)* 4 chips x (16K x 8 bits) — (003C)
F8000 thru F8003F	MMU registers (requires supervisor mode)
F81001 thru F8103F	68230 parallel interface/timer (odd addresses)
F82001 thru F8201F	2681 dual asynchronous receiver/ transmitter (odd addresses)
F90000 thru F9FFFF	VMEbus short I/O space (64K bytes)
FA0000 thru FFFFFF	Six 64K-byte segments that redundantly map to the VMEbus short I/O space

*As shipped

rectional operation. A Centronics printer-compatible interface is the default configuration

The PI/T timer contains a 5-bit prescaler and may be clocked by the system clock or, using an external prescaler, by an external clock. The timer can be used to generate periodic interrupts, a square wave, or a single interrupt after a programmed time period.

INTERRUPT HANDLER

The module can respond to local and system interrupts on any or all of the seven priority levels A header is provided for selecting the level(s)

VMEbus REQUESTER

The module has the logic required to request mastership of the data transfer bus (DTB) on any one of the four bus request levels and to operate in either of two bus release modes release-when-done (RWD) or release-on-request (ROR). The module has headers for jumper selection of a bus request level and the desired bus release mode.

RESET OPTIONS

To provide flexibility for system usage, the module has headers for jumper selection of one of three reset modes. These are:

- module reset by VMEbus RESET* signal
- module reset by local powerup circuit or
- module and VMEbus reset by local powerup circuit

Mechanical and Environmental Specifications

Characteristics	Specifications
Configuration	DTB Master: A24; D16
Form Factor	Double High Eurocard
Power Requirements	+5 Vdc @ 2 5 A (typ) +12 Vdc @ 40 mA (typ) −12 Vdc @ 20 mA (typ)
Environmental Limits Operating Temperature Storage Temperature	 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)

MVME115M

USAGE

The MVME115M Monoboard Microcomputer can operate as master in any VMEbus system in which another module with bus arbitration capability acts as system controller. Modules having bus arbiters include:

MVME025	System Controller
MVME050	System Controller
MVME101	Monoboard Microcomputer
MVME110	Monoboard Microcomputer

Other VMEmodules

MVME200	64K Dynamic RAM
MVME201	256K Dynamic RAM
MVME202	512K Dynamic RAM
MVME211	Static RAM/ROM
MVME222-1	1M Dynamic RAM
MVME222-2	2M Dynamic RAM
MVME300	GPIB Listener, Talker, Controller
MVME320	Disk Controller
MVME330	LAN Controller

Ordering Information

Part Number	Description
MVME115M	VMEmodule Monoboard Microcomputer with 8 MHz MC68010 MPU, MC68541 MMU, one parallel and two serial I/O ports. Includes three socket pairs for up to 64K bytes of ROM/RAM memory.
MVME115M/D1	VMEmodule Monoboard Microcomputer User's Manual

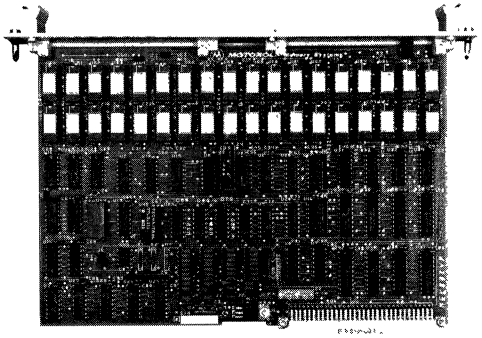
Related Documentation

Part Number	Description
MVMEBS	VMEbus Specification Manual
MC68010UM	16-Bit Virtual Memory Microprocessor
MC68451UM	Memory Management Unit

MVME200
MVME201

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

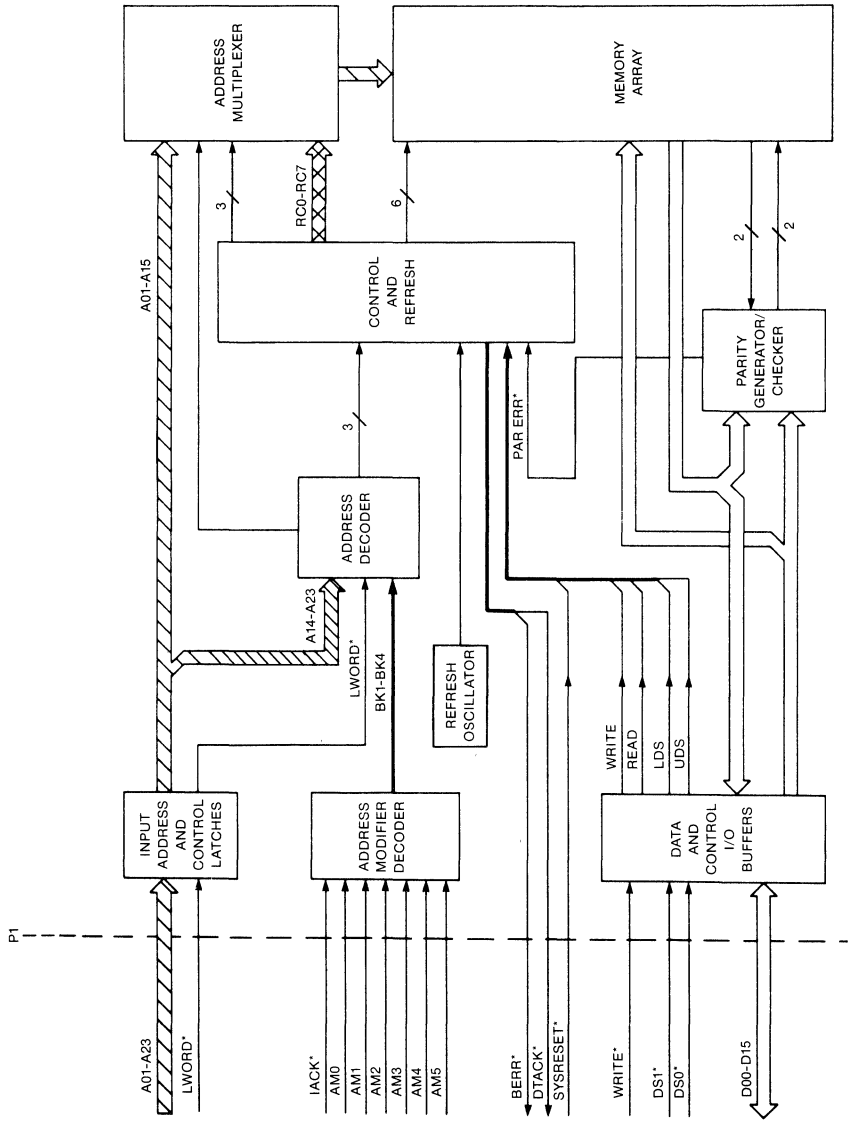
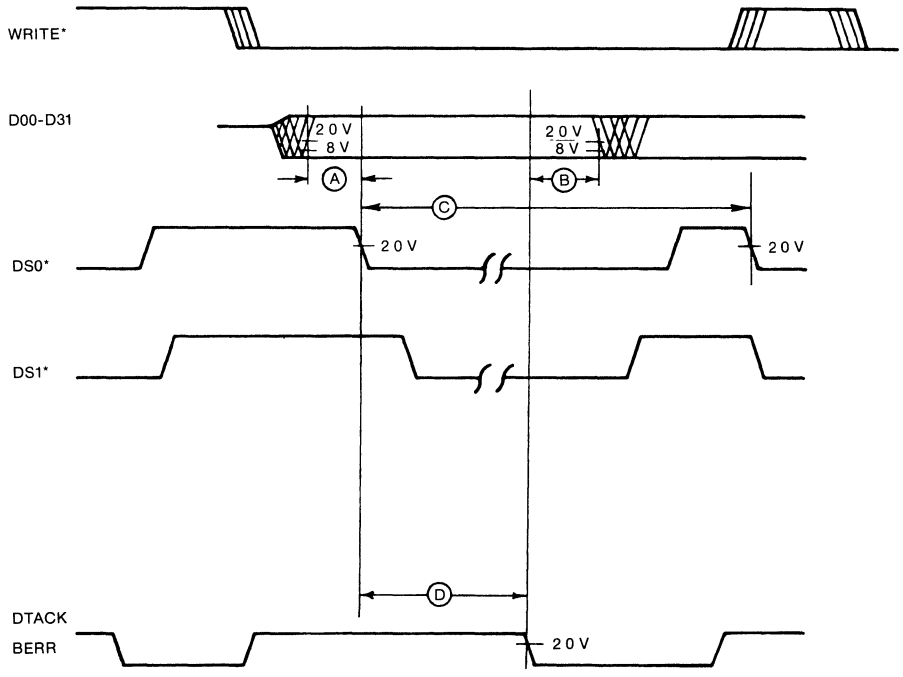


FIGURE 1 — MVME200/201 Functional Block Diagram

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

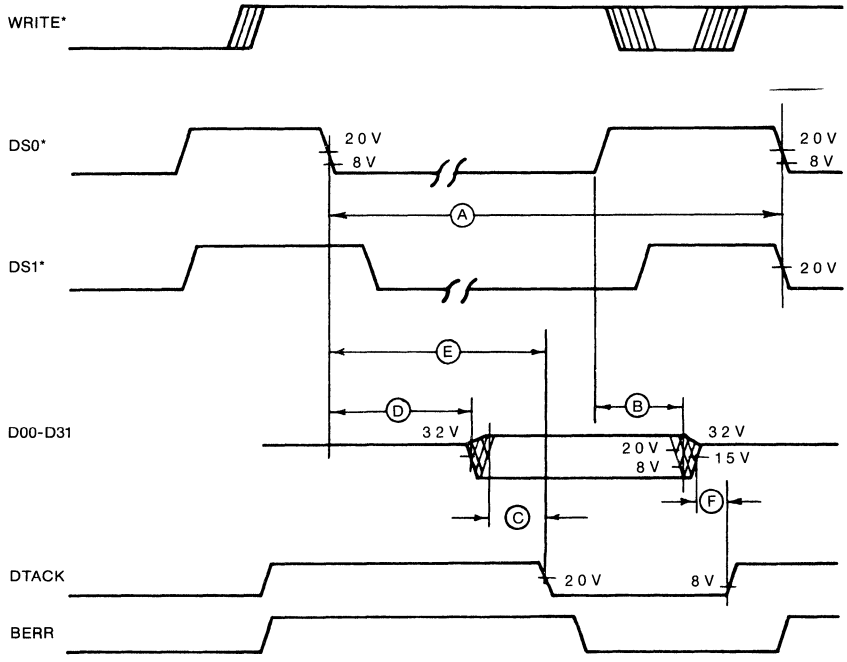


Parameter	Description	Min	Max	Unit
A	Valid data to DS0*	0		ns
B	DTACK* to invalid data	0		ns
C	Cycle time	565	595	ns
D	DS0* to DTACK*	315	370	ns

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	0° to 70° C	
Storage	-55° to 85° C	
Relative Humidity	0% to 90% (non-condensing)	
Power Requirements		
MVME200	+5 Vdc @ 2.7 A (max)	
	+12 Vdc @ 0 5 A (max)	
MVME201	-12 Vdc @ 6 5 A (max)	
	+5 Vdc @ 3 3 A (max)	
Dimensions	*(Board Only)*	*(With Front Panel)*
Height	6.31 in (160 3 mm)	7 40 in (188 mm)
Depth	9 19 in (233 4 mm)	10 31 in. (261.9 mm)
Thickness	0 062 in. (1 57 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/D2	MVME200/201 64K/256K Byte Dynamic Memory Module User's Manual

Other VMEmodules Include:

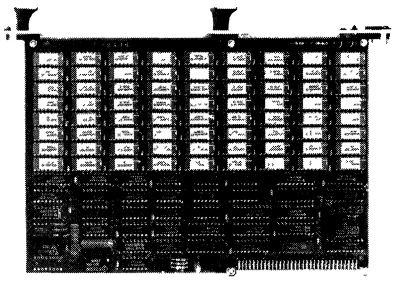
Part Number	Description
MVME101	VMEmodule Monoboard Microcomputer (8 MHz MC68000 MPU, Serial & Parallel Ports)
MVME110-1	VMEmodule Monoboard Microcomputer (8 MHz MC68000 MPU, Serial Port, I/O Channel)
MVME211	VMEmodule Static RAM/ROM (16 Sockets for up to 128K Bytes of RAM/ROM/PROM/EPROM)
MVME300	VMEmodule GPIB Controller with DMA (Provides IEEE-488 Listener, Talker, Controller Functions)

Related Documentation

MVMEBS	VMEbus Specification Manual

VMEmodule 512K/1M/2M Byte Dynamic RAM

MVME202
MVME222-1
MVME222-2



- 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)
  - 250 ns (without parity)
  - 280 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 VMEmodule 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

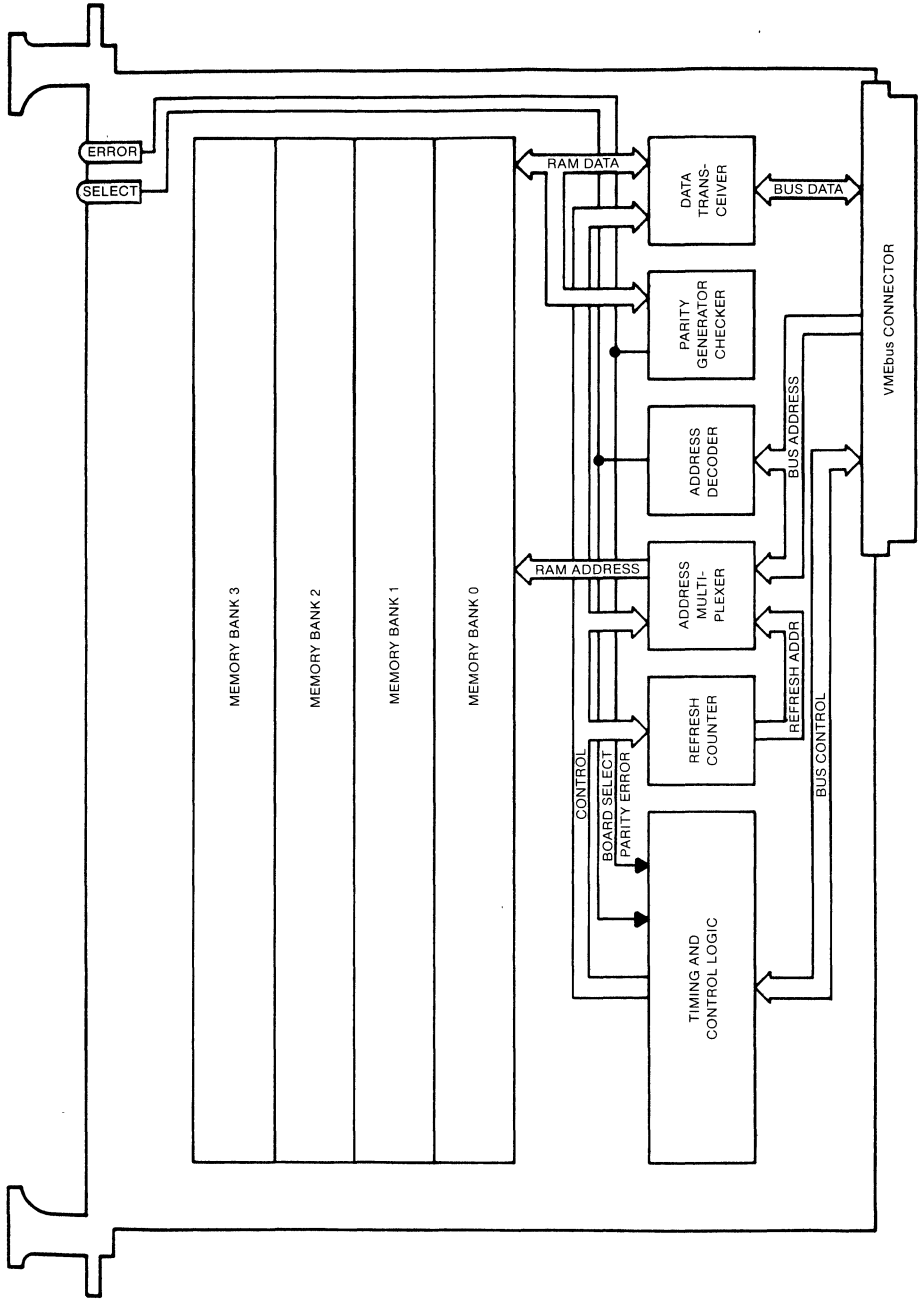


FIGURE 1 — VMEmodule Dynamic RAM Functional Block Diagram

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

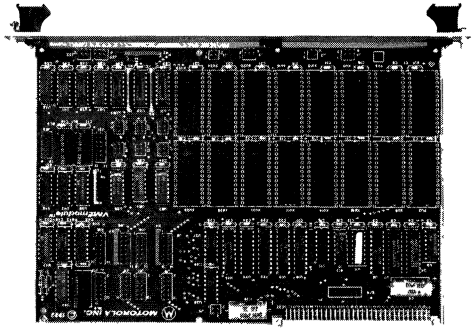
Part Number	Description
MVME202	VMEmodule 512K Byte Dynamic RAM with Byte Parity Includes User's Manual
MVME222-1	VMEmodule 1M Byte Dynamic RAM with Byte Parity Includes User's Manual
MVME222-2	VMEmodule 2M Byte Dynamic RAM with Byte Parity Includes User's Manual.
MVME202/D1	MVME202/222 512K/1M/2M Dynamic RAM User's Manual

Related Documentation

MVMEBS/D1	VMEbus Specification 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)

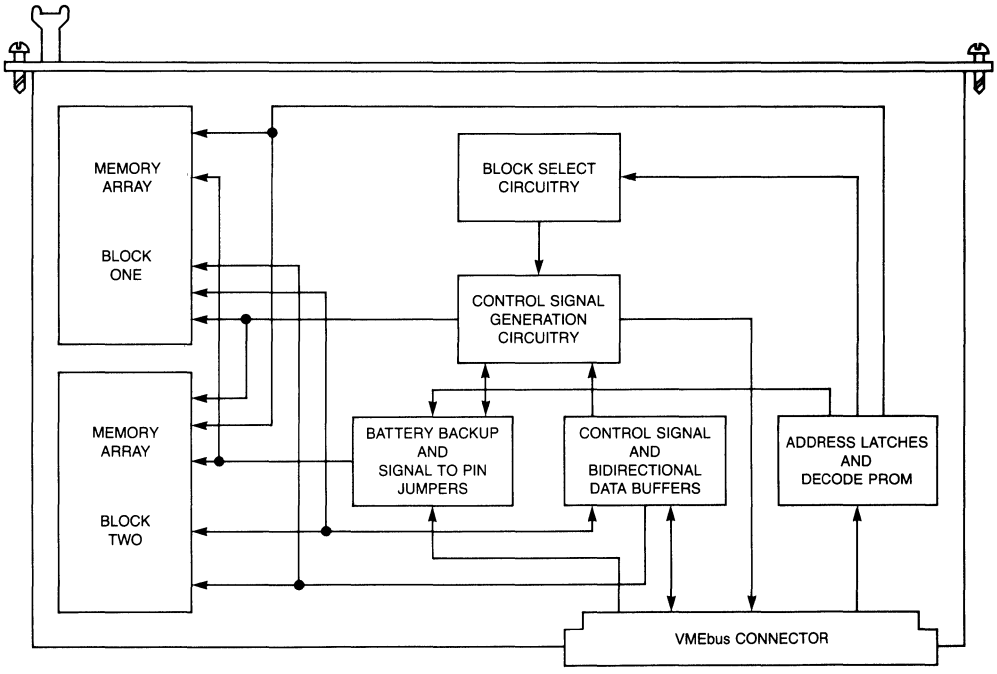


FIGURE 1 — MVME211 Functional Block Diagram

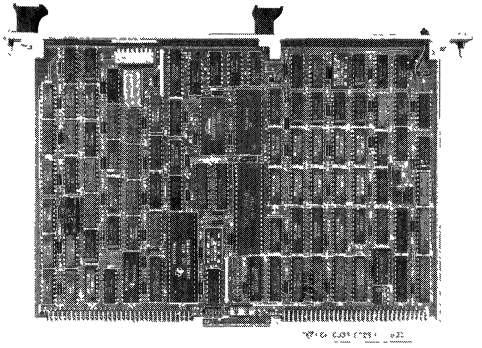
Ordering Information

Part Number	Description
MVME211	VMEmodule Static RAM/ROM/EPROM Memory. Includes User's Manual
MVME211/D1	VMEmodule RAM/ROM/EPROM Memory Module User's Manual

Related Documentation

MVMEBS/D1	VMEbus Specification Manual

VMEmodule GPIB Controller With DMA



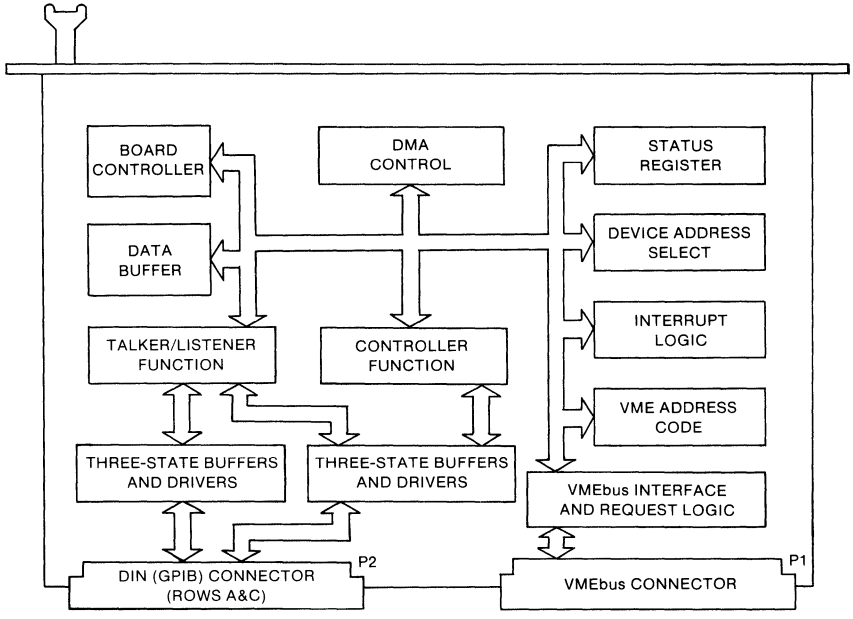
- Meets Complete IEEE 488-1978 Standard and the IEEE 488-1980 Supplement
- VMEbus Compatible
- Functions Under Master or Slave Configurations on VMEbus
- Complete Controller, Talker, and Listener Capability
- DMA Interface for High Speed Transfers
- Transfer Data Under DMA or Non-DMA Mode
- Supports 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*

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

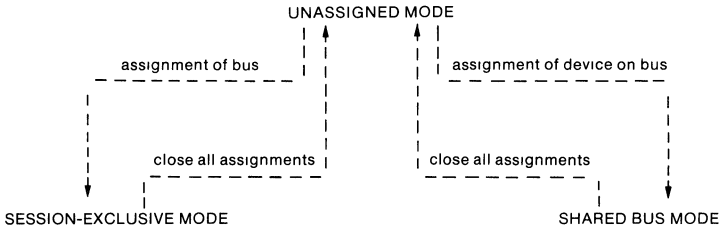


FIGURE 3 — GPIB System Under VERSAdos

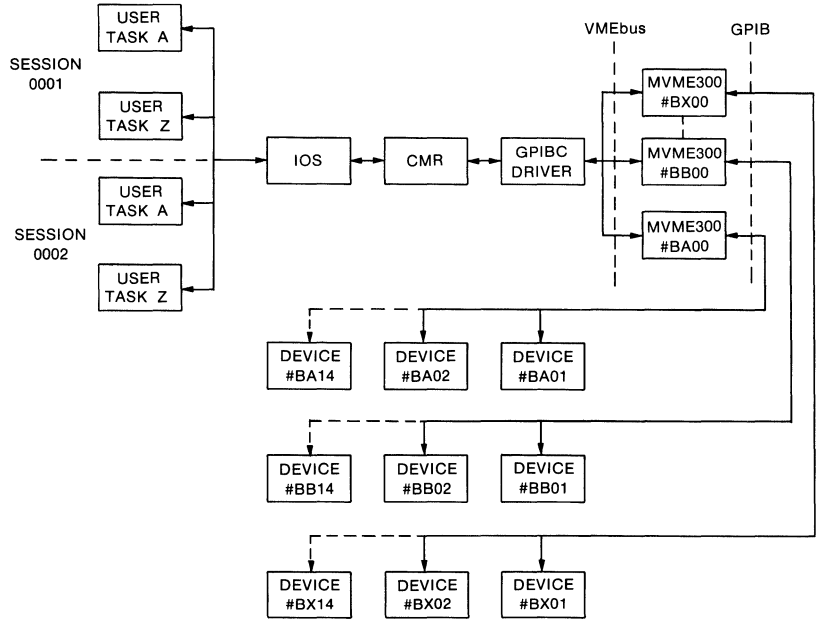


TABLE 2 — GPIBC – Supported IOS Commands

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/D1	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
MVMEBS	VMEbus Specification Manual
IEEE STD 488-1978	Digital Interface for Programmable Instrumentation
IEEE STD 488A-1980	Supplement to Above Standard

Related Documentation — Software

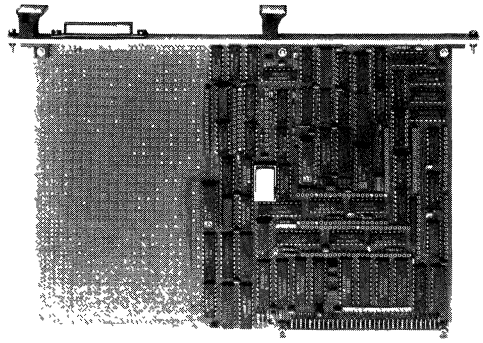
Part Number	Title
M68KVOVER/D3	VERSAdos Overview
M68KRMS68K/D3	M68000 Realtime Multitasking Software User's Manual
RMS68KIO/D2	VERSAdos Data Managment Services and Program Loader User's Manual
M68KSYSGEN/D3	System Generation Facility User's Manual
MVME3SW	MVME300 GPIB Controller with DMA Software Manual

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.

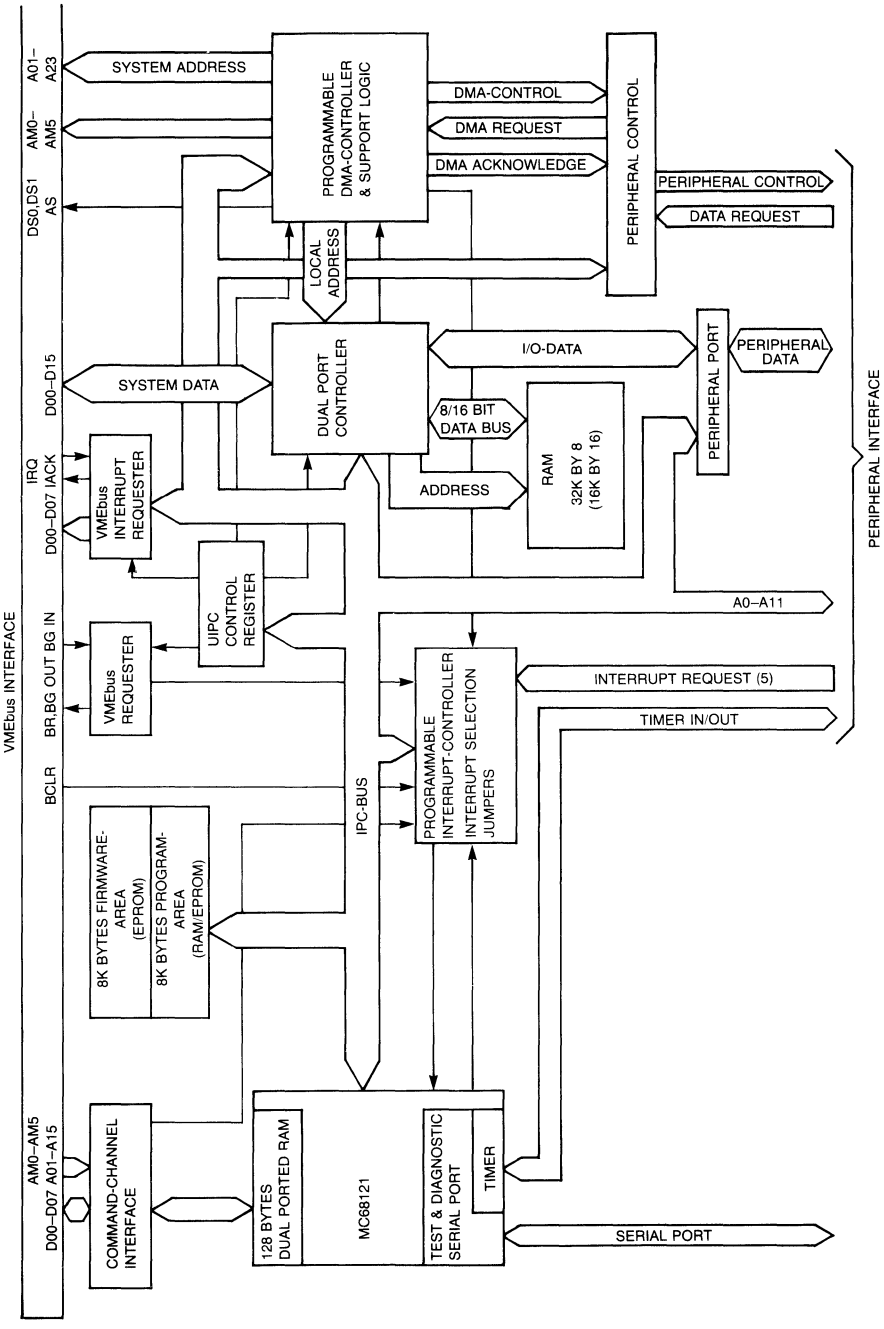


FIGURE 1 — MVME310 Functional Block Diagram

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.

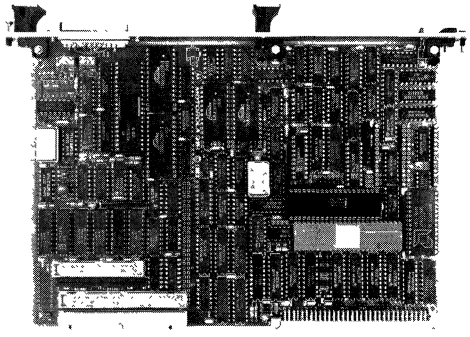
Ordering Information

Part Number	Description
MVME310	VMEmodule Intelligent Controller For Custom Interfacing. Includes User's Manual.
MVME310/D1	VMEmodule Intelligent Controller for Custom Interfacing User's Manual

Related Documentation

Part Number	Description
MVMEBS/D1	VMEbus Specification Manual
M68DRVGD/D1	Guide to Writing Device Drivers for VERSAdos
M68IPCS/D2	M68000/IPC Command Channel Software Interface Reference Manual

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-232C 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

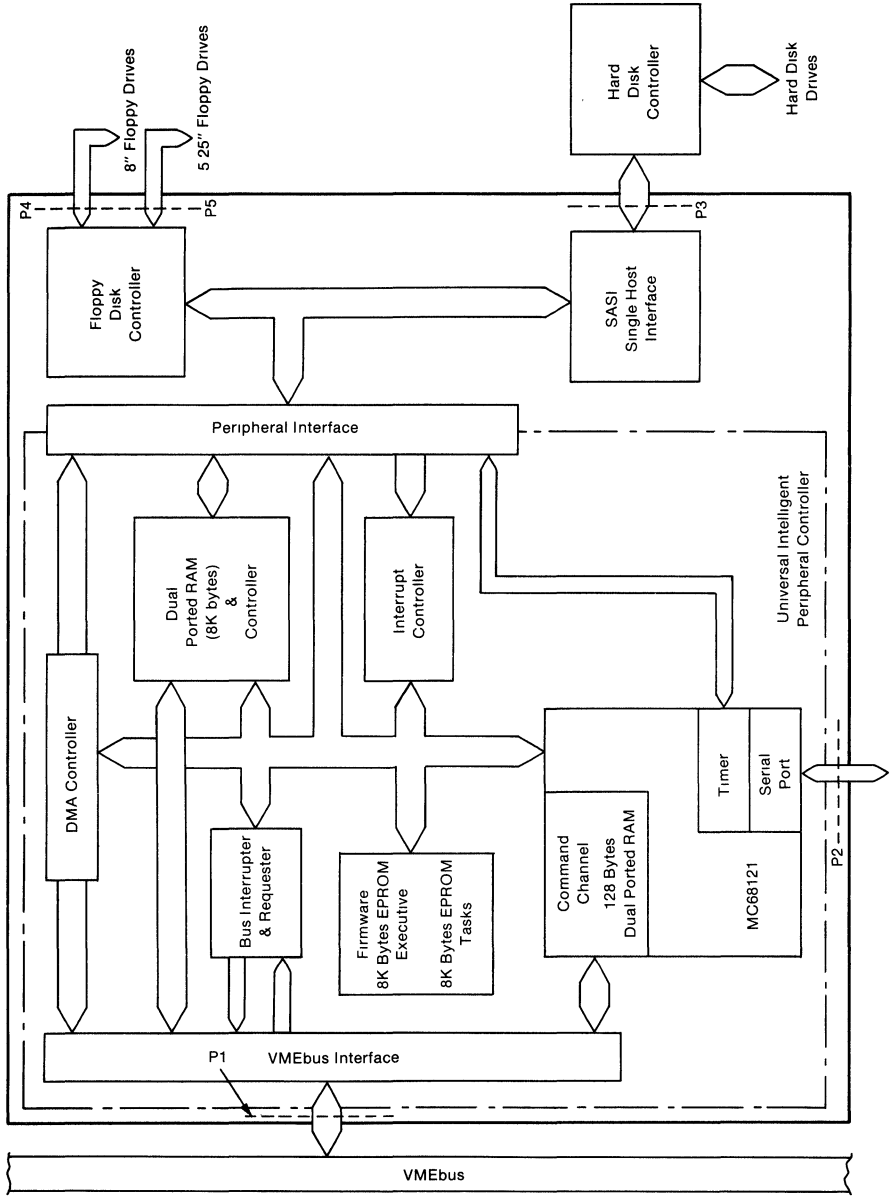


FIGURE 1 — MVME315 Functional Block Diagram

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

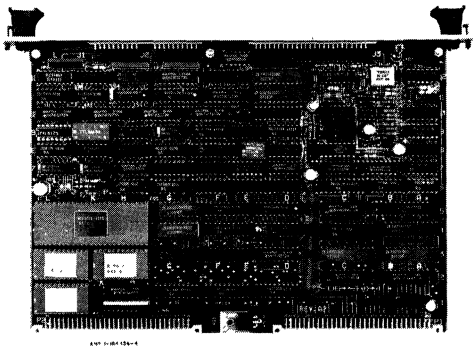
Ordering Information

Part Number	Description
MVME315	VMEmodule Intelligent Floppy Controller/SASI Interface with DMA Includes User's Manual
MVME315/D1	MVME315 Intelligent Floppy Controller/SASI Interface User's Manual

Related Documentation

MVMEBS/D1	VMEbus Specification Manual
M68DRVGD/D1	Guide to Writing Device Drivers for VERSAdos

VMEmodule Winchester/Floppy Disk Controller



- Controls Mixed 5¼" and 8" Drives: Up to Two Winchester Hard and Two Floppy Disk Drives or up to Four Floppy Disk Drives
- Provides High Performance DMA Data Channels
- Programmable Data Transfer Mode: Two Byte or 200 Microsecond Block
- 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
- Automatic Bad Sector Handling
- Single Phase Write and Read Clocks
- Data Transfer — 16-Bit DMA/8-Bit Register
- Self Diagnostics
- VMEbus Requester and Interrupter
- 0°C to +50°C Operating Temperature Range
- VMEbus Compatible

The Disk Controller is a double high Eurocard module for adding mass storage capacity to a VMEbus system It provides high performance DMA data channels between system memory and Winchester hard disk drives and/or floppy disk drives. The module is used in applications having intensive real-time disk I/O or multiprocessing structures to reduce VMEbus traffic and increase system throughput. Figure 1 is a functional block diagram of the controller module.

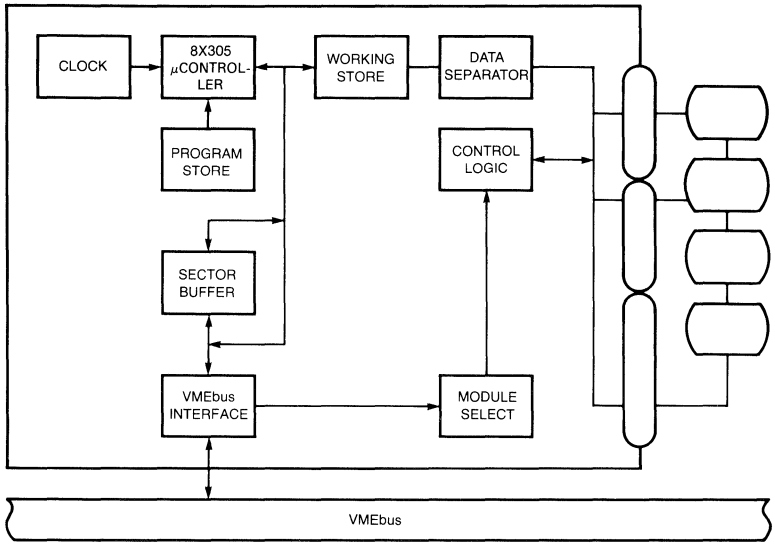
Host/MVME320 Communications

An intelligent module, the disk controller offers the user a high level, easy to program interface. Microinstructions in a 4K × 24 ROM are executed on a Signetics 8X305 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 a table in global memory 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 and to obtain interrupt source and drive status information from the module.

The other host/module communications avenue is the event control areas (ECA) which 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 comprising static and dynamic fields is created by the host when requesting execution of a command. Static fields remain unchanged. Dynamic fields are used during execution and updated at completion by the controller.

FIGURE 1 — Disk Controller 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
- Because both sections are serviced by the fast 8X305 microcontroller, the module is able to control a variety of disk types and media formats

Up to four disk drives can be interfaced to the module. Of these two can be Winchester 5¼" or 8" hard disk and two can be 5¼" 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 1024 cylinders/drive) and head selection (up to 16 heads/drive) is automatically provided by the controller for host requested I/O.

Table 1 is a partial list of disk drives which can be used with the controller module.

Data Transfer Modes

On command request, the host specifies the desired transfer mode using the DMA Type field in the ECA Two DMA data transfer modes are available: two byte and 200 μs block. In the two byte mode, the VMEbus is requested, two consecutive bytes are transferred and the bus released. In the 200 μs mode, the bus is requested, data is transferred for 200 μs and the bus released.

Data Interface

For a requested read or write operation, data is automatically transferred between any of the four disk drives and system memory which can be located anywhere in the VMEbus 16 megabyte 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

Module design uses a multiplexer to transfer disk data to a phase locked loop data separator before sending it to serializer/deserializer (SERDES) circuitry This section converts serial disk data to parallel 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.

Error Detection And Correction

To insure data integrity, the module 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.

TABLE 1 — MVME320—Compatible Disk Drives

Hard Disk Drives	
8″	5¼″
Shugart SA1000 Series Quantum Q2000 Data Peripherals	Seagate ST506, ST512, ST406, ST412 Shugart SA600 RMS500, RMS506, RMS512 Tandon International Memories Inc.
Floppy Disk Drives	
8″	5¼″
Shugart SA800, SA801, SA810, SA850	Amlyn 5850 Shugart SA400, SA410, SA450

TABLE 2 — Disk Controller Command Set

Command	Description
CALB	Recalibrate (reposition head) to track zero
RETD	From first sector encountered, return flag, ID, CRC and read-generated remainder.
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.

Mechanical And Environmental Specifications

Characteristics	Specifications
Configuration	DTB Master: A24, D16 DTB Slave: A24, D8
Form Factor	Double High Eurocard
Power Requirements	+5.0 Vdc at 2.6 A (typ) +12 Vdc at 20 mA (typ) −12 Vdc at 20 mA (typ)
Environment Limits Operating Temperature Storage Temperature Humidity	 0°C to +50°C −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)

Command Set

The controller module offers the set of nine high level commands listed in Table 2.

Control Program PROM Size Selection

The module has a header to allow selection of a 2K × 8 or 4K × 8 size for the three PROMs in which the control program is stored. This header is wired for the 4K × 8 PROMs shipped with the module.

Base Address Selection

The module has a header for jumper selection of a base address which can be set on any 1K byte boundary throughout the 16 megabyte VMEbus address space.

Address Modifier Code Selection

The module 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 six of the seven VMEbus interrupt priority and interrupt acknowledge levels: 1 through 6. 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 module 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 4.4 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, M68DRVGD/D1, detailing how to write a driver that runs under the M68000 Real-Time Multitasking kernel or under VERSAdos is available.

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. Table 3 lists the disk controllers for which bootstrap, load and dump capabilities are provided by the debugger supplied with the corresponding microcomputer.

TABLE 3 — Boot/Load/Dump Support

Microcomputer	Debugger Supported Disk Controllers
MVME101	MVME315
MVME110-1	MVME315, MVME320, M68RWIN1, MVME420/SA1403D
MVME115M	MVME320, M68RWIN1*

*Using the MVME316 VMEbus-To-I/O Channel Interface

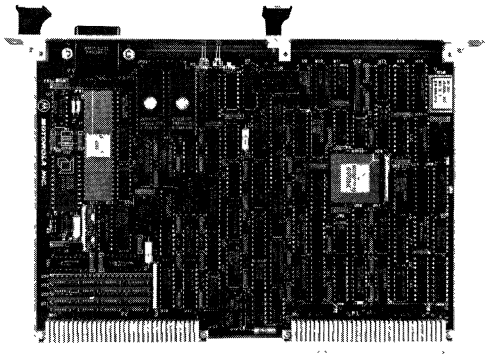
Ordering Information

Part Number	Description
MVME320	VMEmodule Winchester/Floppy Disk Controller
MVME320/D1	VMEmodule Winchester/Floppy Disk Controller User's Manual

Related Documentation

Part Number	Description
MVMEBS/D1	VMEbus Specification Manual
M68DRVGD/D1	Guide to Writing Device Drivers For VERSAdos

VMEmodule LAN Controller



- Frees VMEbus Hosts From Protocol Processing Burden
- Interfaces with Both VMEbus and VERSAbus Hosts (via M68KVM33) 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
- VMEbus Compatible
- Ethernet, Version 2.0 Compatible
- AM 7990/7991 (LANCE/SIA) VLSI
- 128K 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°–70°C Operating Temperature Range

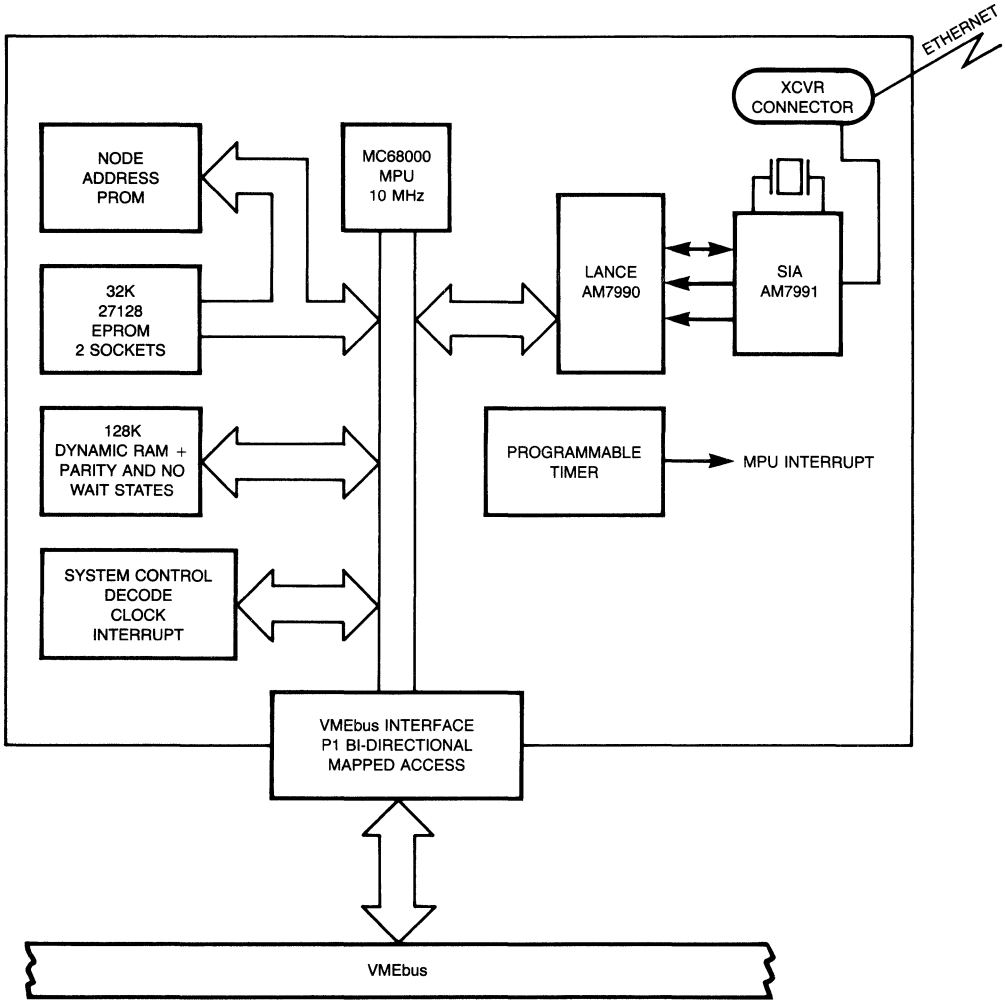
Software Features

- 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 MVME330 LAN Controller VMEmodule is an advanced communications processor which provides for VME/10-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 VMEbus industry standards and is fully supported by Motorola's SYSTEM V/68 and VERSAdos operating systems.

MVME330 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 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 VMEbus interface is about 200 one kilobyte packets per second. A block diagram of MVME330 is shown in Figure 1.

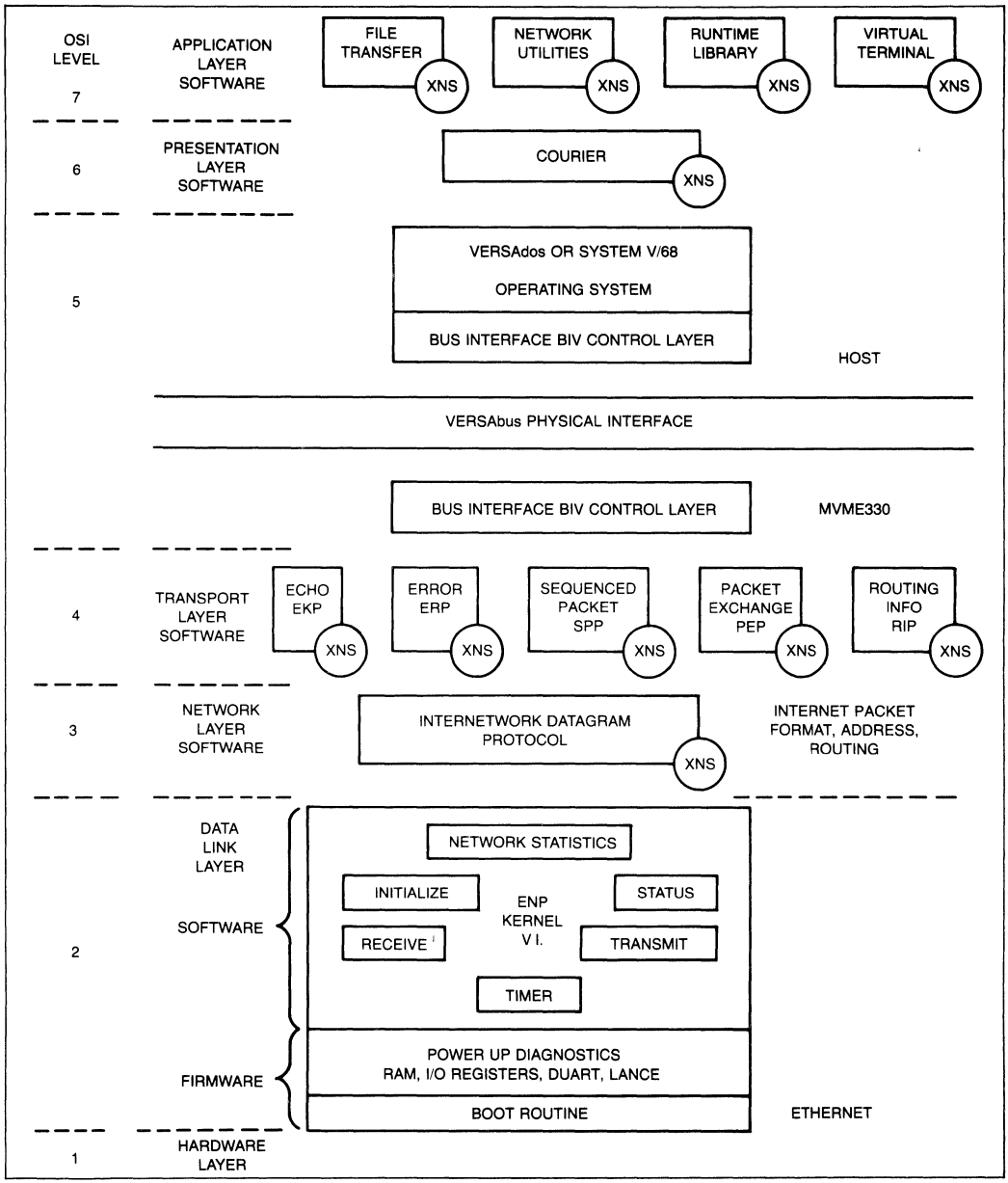
FIGURE 1 — MVME330 LAN Controller



MVME330's on-board MC68000 MPU facilitates the downloading of custom protocol implementations. Host-specific drivers also are easily written for interfacing MVME330 to another host.

By interfacing directly with the LANCE chip to perform all Ethernet I/O, the MVME330 kernel acts as a standard interface between the hardware and the XNS software which in turn communicates, over VMEbus 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 MVME330 communications executive comprises initialize, timer, transmit, receive, status and network statistic functions, as shown in Figure 2.

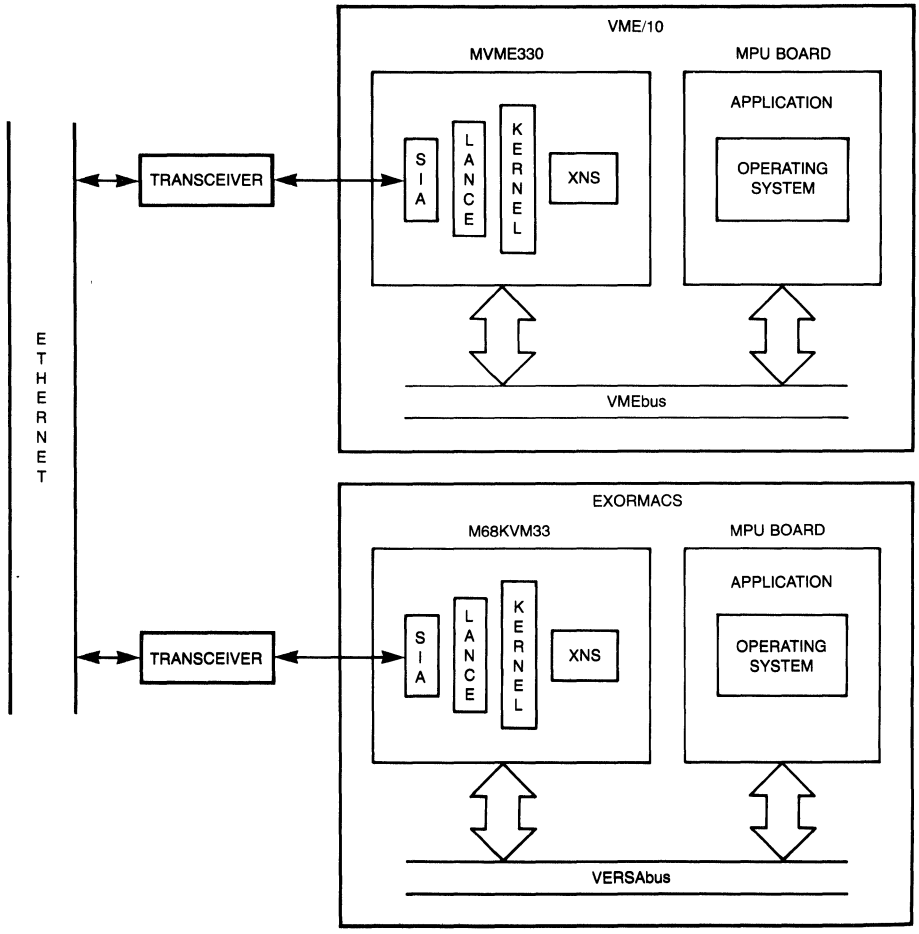
FIGURE 2 — Host/MVME330 Ethernet Functions



The MVME330-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, 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 clean, well-defined shared memory interface.

Figure 3 depicts a two node Ethernet network.

FIGURE 3 — Two Node Network



FUNCTIONAL DESCRIPTION — HARDWARE

MVME330 offers System V/68 and 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 greater than 200 maximum length packets per second. Requiring no additional hardware but a cable, MVME330 can be connected to standard Ethernet transceivers.

The MVME330 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 MVME330 is a 10 MHz MC68000. Its duties include moving commands and data to and from system memory; responding to and 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 bytes of dynamic RAM with parity which is accessible from VMEbus, the LANCE and the MC68000, in that order of priority.

ROM/EPROM

The MVME330 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 MVME330 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 1/8 of a bit time.

VMEbus Requester/Bus Mastership

Since MVME330 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 MVME330 interrupter circuitry 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.

Interrupting the MVME330 LAN Controller

A host access of any of the top 512 bytes of the 128K bytes of VMEbus 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 MVME330 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 SOFTWARE

The MVME330 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 MVME330 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, the kernel incorporates a full debugger. 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 MVME330 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 MVME330 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 MVME330: 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 MVME330 can handle multiple back-to-back packets of any length.

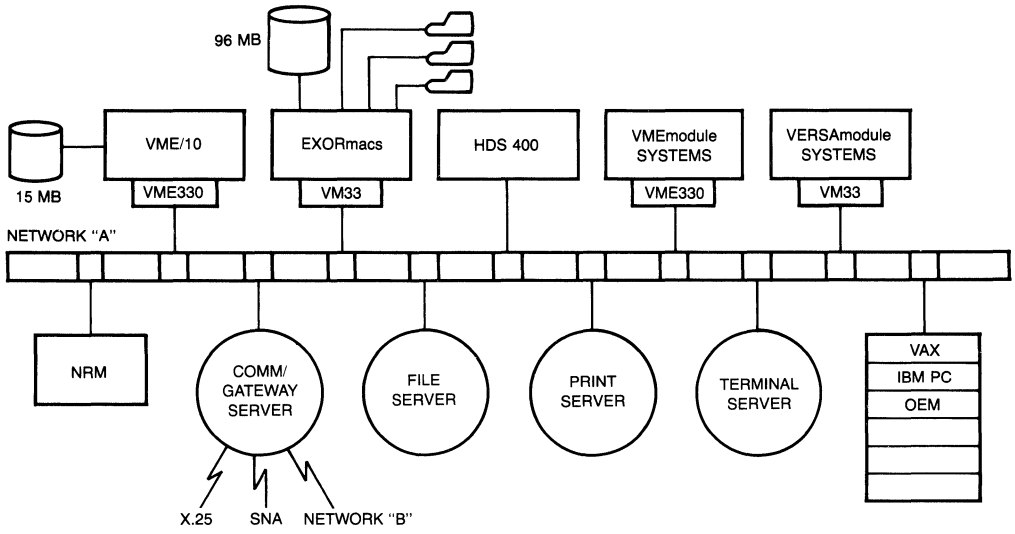
Network Distances

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.

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

MVME330 SPECIFICATIONS

The specifications for the MVME330 are listed in Table 2.

TABLE 2 — MVME330 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, MVMEBS/D1.

ETHERNET TRANSCEIVER PORT CONNECTOR

The Ethernet transceiver port connector is a 15-pin subminiature.

Ordering Information

The MVME330 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 as can the executive kernel and the kernel debugger only.

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 128K RAM, LANCE (7990), SIA (7991), 68000 MPU, Kernel Firmware and Power up self-test
M68NNXBVMELAN for SYSTEM V/68 Host M68VVXBVMELAN for VERSAdos Host	Object Software supplied on 5 25″ 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 MVME330 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
MVME330-UX	MVME330 plus appropriate software for SYSTEM V/68, on 5.25″ floppy
MVME330-VX	MVME330 plus appropriate software for VERSAdos, on 5.25″ floppy
M6811RBVMEBUG	LAN Controller Debugger software, kernel, self-test in 32K EPROM
M6811RBVMEKRN	LAN Controller Kernel, and self-test in 16K EPROM
M68NNXSVMELAN for SYSTEM V/68 Host M68VVXSVMELAN for VERSAdos Host	Source Software supplied on 5 25″ floppy includes — Ethernet 2.0 compatible Network Software device driver, — Host BIV and MVME330 BIV — Kernel, self-test, debugger and documentation (XNS Host resident utilities and protocol from third party vendor)

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

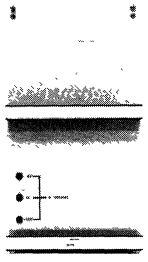
- Plug-In Power Supply
- 20-Slot VMEbus Backplane
- 9-Slot VMEbus Backplane
- 5-Slot I/O Channel Backplane
- Double High Extender Board
- Double High Wirewrap Board
- Single High Extender Board
- Single High Wirewrap Board
- Remote I/O Channel Connector Board
- Chassis with Power Supply for VMEbus and 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



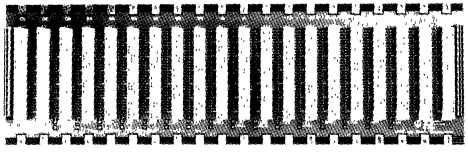
MVME910-3

VMEbus Backplanes

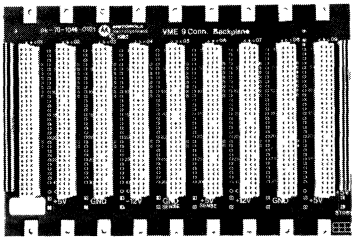
The MVME920 and MVME921 VMEbus Backplanes are used to interconnect VMEmodules They are fully assembled, with connectors and terminators The MVME920 has 20 connectors (a full 19" chassis width), and the MVME921 has 9 connectors

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

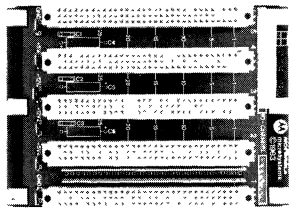
MVME900 Series

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

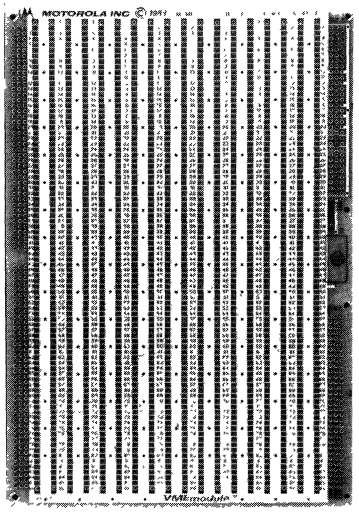
- All Signal and Power Line Connections Provided
- DIN-41612C 64-Pin Connectors
- "Fast-On" Power and Ground Terminals
- Termination for all I/O Channel Signal Lines except INT-4 and XACK*



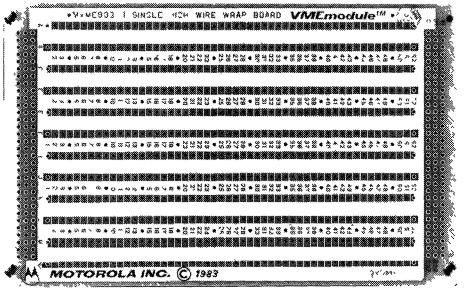
MVME922

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



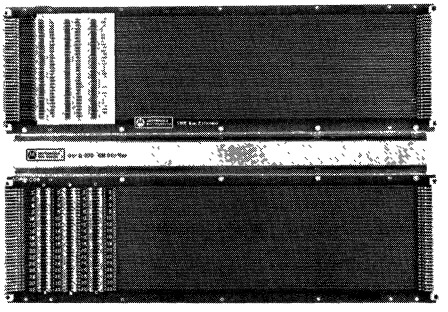
MVME931-1



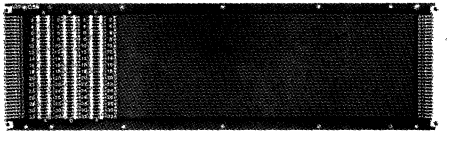
MVME933-1

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



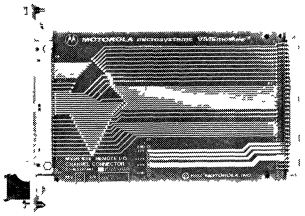
MVME930



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



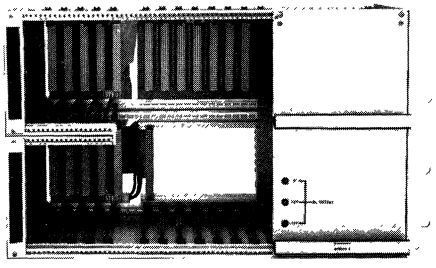
MVME935

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



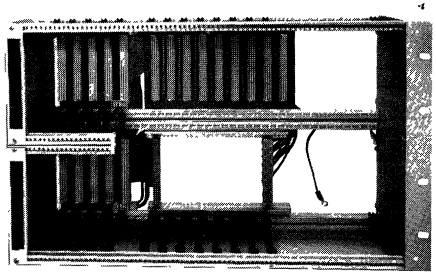
MVME940-1

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

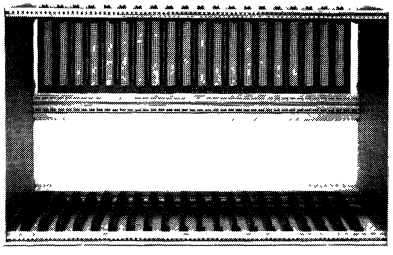
- 19" Rack Mounting Chassis
- Card Guides and Hardware to Support 9 Double-High and 10 Single-High Eurocard Modules
- One MVME921 9-Slot VMEbus Backplane
- Two MVME922 5-Slot I/O Channel Backplanes



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

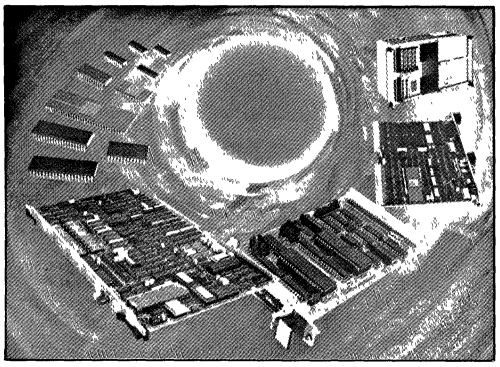
MVME920	20-Slot VMEbus Backplane
MVME921	9-Slot VMEbus Backplane
MVME922	5-Slot I/O Channel Backplane
MVME930	VMEbus (Double high) Extender Board
MVME931-1	VMEbus (Double high) Wirewrap Board
MVME932	I/O Channel (Single high) Extender Board
MVME933-1	I/O Channel (Single high) Wirewrap Board
MVME935	Remote 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

MVMEBS/D1	VMEbus Specification Manual
M68RIOCS/D2	Input/Output Channel Specification Manual
MVME900/D1	MVME900 Series Equipment User's 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 UNIX-derived SYSTEM V/68 Operating System.

VERSAmodule Data Sheets

(continued)

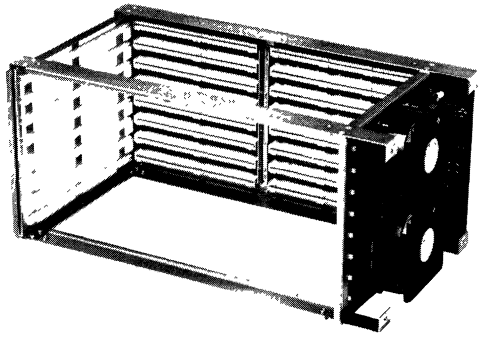
2

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.

MVMCH3-1, -2, MVMCC3, M68KVMPM1

VERSAmodule Chassis

- Incorporates Eight-Slot Card Cage in Front-Load Metal Housing
- Versions optionally available incorporating 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.

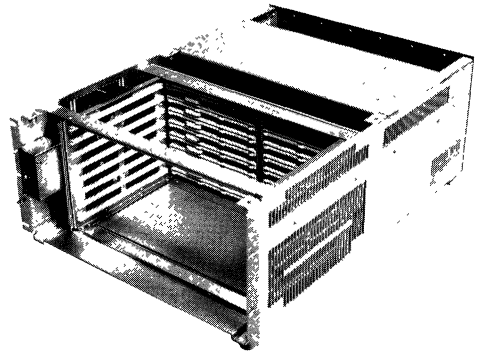
For users planning to employ single Eurocard format I/Omodule cards, a version of the chassis is available that has a 6-slot I/Omodule Card Cage installed with rear panel access.

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.

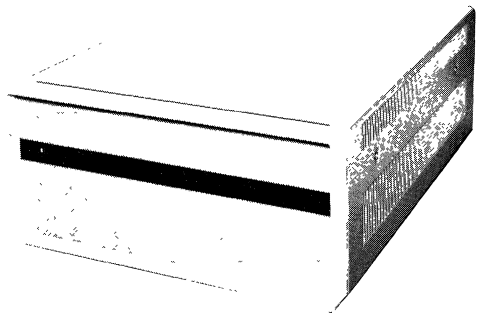
Connections For Backup

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.

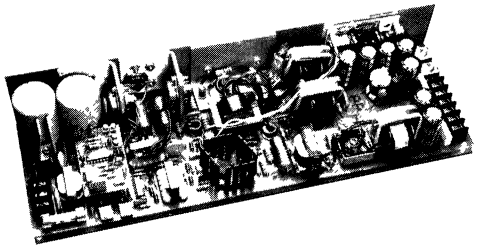


400 Watt Switching Power Supply

- 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

2

VERSAmodule Power Monitor

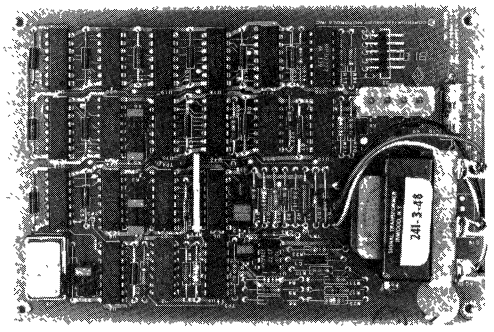
- Monitors System AC Line (on-board transformer) and Supply Fail Signal
- Operates from +12 Vdc
- Detects Cycle Dropout (user selectable for 0.5, 1.0 and 1.5 cycles) Below a 90 or 100 Vac Input Line Voltage
- Generates Proper Sequence of Power-up and Power-down Signals for VERSAbus
- Provides ACCLK Signal 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)

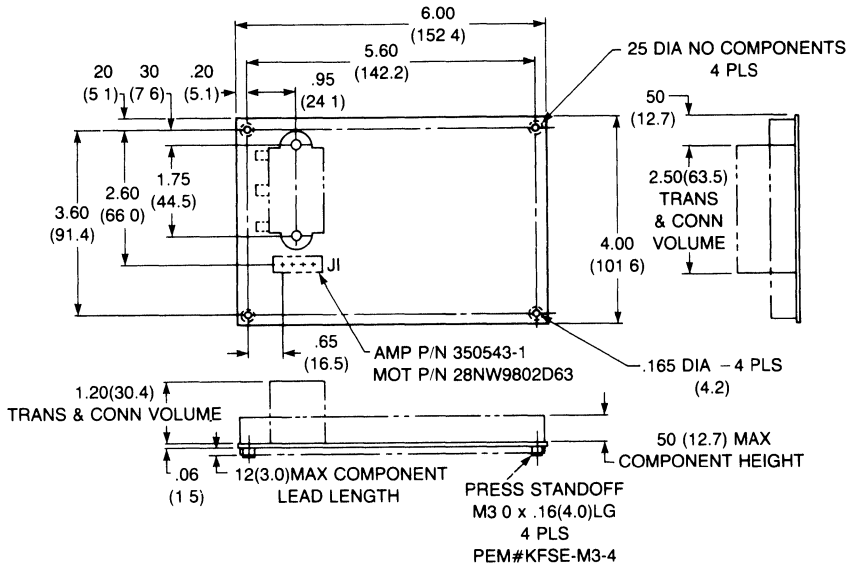
- +5 V (for Monitoring During Power Up)
- +12 Vdc circuit power @ 100 mA (Max.)
- +12 Vdc supply fail signal (active low)
  V_{IL} = 2.0 Vdc (Max.) @ 3.0 mA (Min.)
  V_{IH} = 7.0 Vdc (Min.)
- Board mating connector (P1)
  Amp (Mate-N-Lok) 1-480424-0 (4 pin)

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.

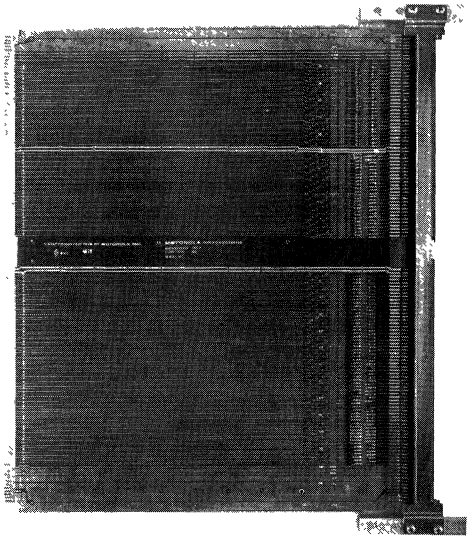
Options

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.

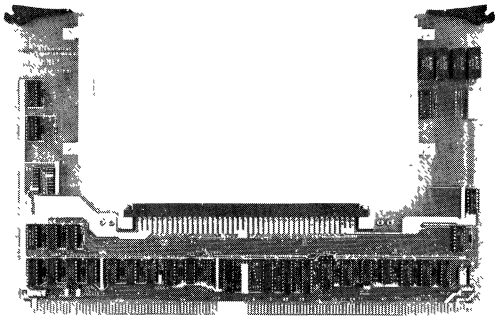
Individual test points are provided for each extended signal. The bus signal may be isolated by cutting the track between two 10" spaced hole patterns A header with shorting bars may be inserted to restore signal continuity Mechanical ejectors ease the removal of test modules as well as the extender module itself

Ordering Information

Part Number	Description
M68KEXTM	VERSAbus Extender Module

2

VERSAbus Adapter Module



- Provides VERSAbus Systems with the ability to utilize EXORbus family modules
- Micromodule compatible
- User selectable access to upper or lower data bytes
- 16-bit communication ability with use of two modules at one address
- User selectable interrupt levels
- VERSAbus interrupt control logic
- Synchronous control lines provided
- User selectable 1 MHz or 2 MHz operation for EXORbus family modules

The VERSAbus Adapter Module (VAM) is an interface between the 8-bit EXORbus modules and the 16-bit VERSAbus. The VAM is provided as an option for VERSAbus systems, enabling use of various I/Omodules, memory and Micromodules designed for the EXORbus. The following compatibility list (Table 1) provides a listing of EXORbus compatible modules which may be used in conjunction with the VAM

Synchronous timing and control are made available to the EXORbus module by use of the VAM The VAM makes adjustments for proper addressing and also provides MC68000 MPU interrupt levels and controls required by the VERSAbus interrupt scheme

A large degree of application flexibility is achieved by providing a number of user options. These are summarized in Table 2

TABLE 1 — EXORbus Module Compatibility List

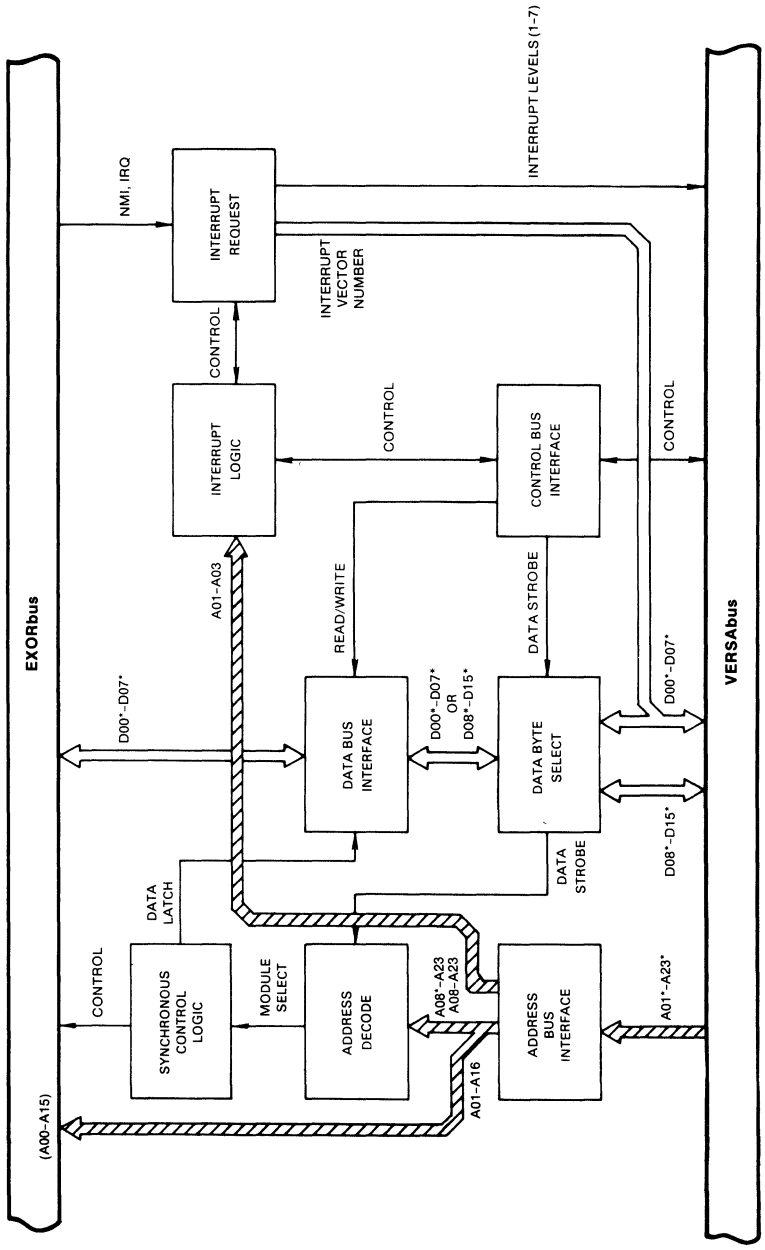
Development System I/O Modules	
MEX6821-2	PIA Module
MEX6850	ACIA Module
MEX6850-2	ACIA/SSDA Support
MEX6854	ADLC Support
MEX68488	GPIA Support
MEX68USM	Universal Support Module
M68120EVM	MC68120 Evaluation Module
Micromodule I/O Memory Modules	
M68MM03	32/32 Input/Output
M68MM05A,B,C	Analog Input/Output
M68MM07	Quad Communications Module
M68MM09	4K CMOS RAM (battery back-up)
M68MM13A,B,C	Digital Output (contact closure)
M68MM13C,D	Digital Input (optically isolated)
M68MM15A,A1,B	High-Level A/D
M68MM15CV,CI	Analog Output
M68MM16-1	Combo Memory — I/O (for MM01 Series)
M68MM16-2	Combo Memory — I/O (for MM19 Series)
M68MM18	8-Channel Serial Communications
M68MM21	8K × 8 CMOS RAM
M68MM21-1	16K × 8 CMOS RAM
M68MM22	Quad 8-Bit Parallel Interface Adapter
Development System Memory Modules	
MEX68RR	EPROM/RAM
MEX68XX-1HR	16K-64K × 8 Dynamic RAM
MEX6808-22	8K × 8 Static RAM
MEX6816-22S	16K × 8 Static RAM
Miscellaneous Modules	
M68MM10C	Battery Backed-Up Clock/Calendar
M68MM14,14A	Hardware Arithmetic Processing Unit
MEX68PP3	PROM Programmer
MGD6800DSM	Data Security Module

TABLE 2 — User Options

The VAM is designed to allow a great degree of application flexibility through various options that may be selected by inserting jumpers or cutting copper runs on the board The available options, along with the factory configuration, are summarized in the following table.

Option Type	Description
NMI (Non-Maskable Interrupt)	An NMI signal generated by the EXORbus module can be jumpered to select any one of seven VERSAbus interrupt request levels (Factory-configured for VERSAbus Level 7.)
IRQ (Interrupt Request)	The IRQ signal into the EXORbus board can be activated from any one of the seven VERSAbus interrupt request levels (Factory-configured for VERSAbus Level 7)
Interrupt Acknowledge	Selects whether or not the VERSAbus Interrupt Acknowledge signal is passed through the VAM board (Factory-configured for IACKIN pass-through.)
1 MHz/2 MHz Clock	Either clock can be selected to supply to the EXORbus module (Factory-configured for 1 MHz.)
Set Interrupt Vector	Selects the interrupt vector provided by the VAM board when acknowledging an interrupt. (Factory-configured for IRQ = $000064 NMI = $00007C)
Data Byte Select	Selects either the "lower" or "upper" data byte from the lower 16 data lines of the VERSAbus, to select the eight data lines of the EXORbus module. (Factory-configured to select upper data byte accesses.)
Address Decode	Consists of four hex address select switches to decode VERSAbus address lines A08* through A23* (Factory-configured for $8000xx.)
Address Don't Care Switch	Allows EXORbus address lines A08-A15 to be eliminated from the decode individually, with the result that the VAM can address different size blocks of available memory. (Factory-configured so that the VAM appears in the VERSAbus map at 128 byte locations in the range of $800000 to $8000FF.)
16-Bit Data Transfer	Allows use of two EXORbus modules at same address locations with a common E-clock, so that two 8-bit bytes may be transferred simultaneously between the EXORbus modules and the VERSAbus data bus.

VERSAbus Adapter Module Block Diagram



Environmental and Power Specifications

Characteristic	Specification
Power Requirements	+5 Vdc, ±5% @ 1.0 Amp maximum
Operating Frequency	1 MHz or 2 MHz
Operating Environment: Temperature Relative Humidity	 0°C to 70°C 0%–90% (non-condensing)
Shipping and Storage Environment: Temperature Relative Humidity	 -40°C to +85°C 0%–90% (non-condensing)

Mechanical Specifications

Height (Thickness)	0 75 in. (1 8 cm)
Width	14 50 in (36.8 cm)
Length	9.25 in. (23.5 cm)

Ordering Information

Part Number	Description
M68KVAM	VERSAbus Adapter Module. Includes User's Manual
M68KVAM/D1	VERSAbus Adapter Module User's Manual

Related Documentation

M68KVBS	VERSAbus Specification Manual
M68EBS/D1	EXORbus Specification Manual

VERSAbug Debugging Packages for the M68KVM01A and M68KVM02-3 VERSAmodule Monoboard Microcomputers

M68KVBUG, 2
M68KVBUGLF, 2LF
M68KVBUGLC, 2LC
M68KVBUG2LMC

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)

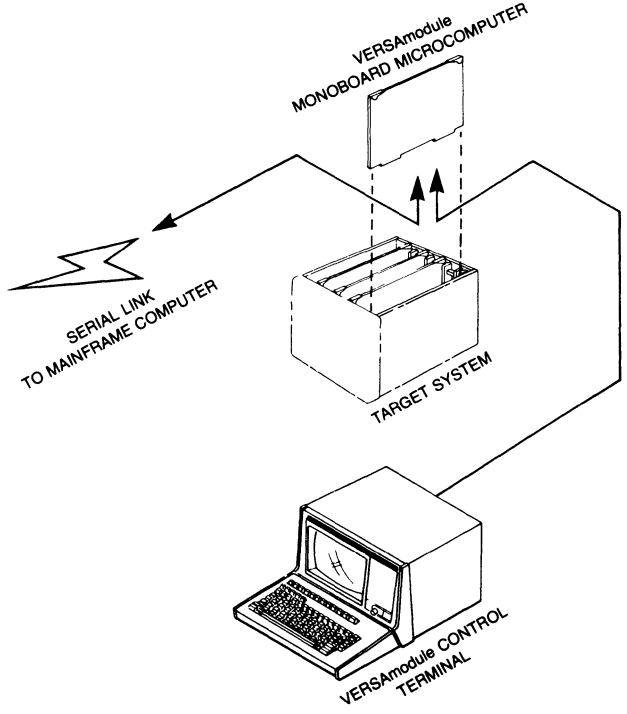
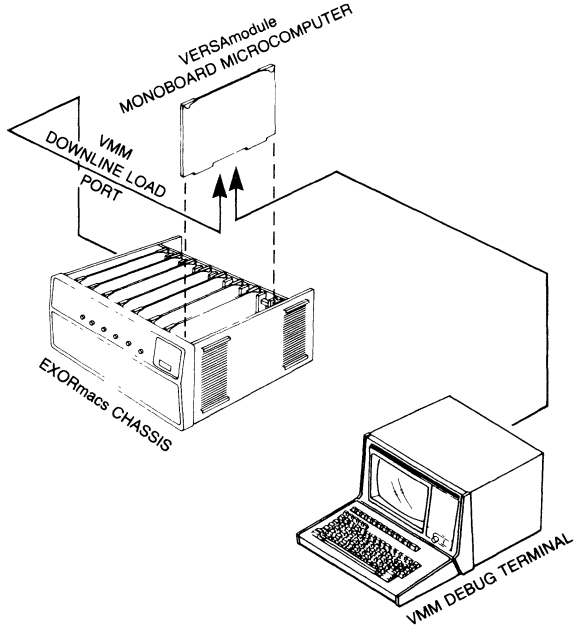


FIGURE 2 — Configuration #2 — Monoboard with VERSAbug in EXORmacs



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)

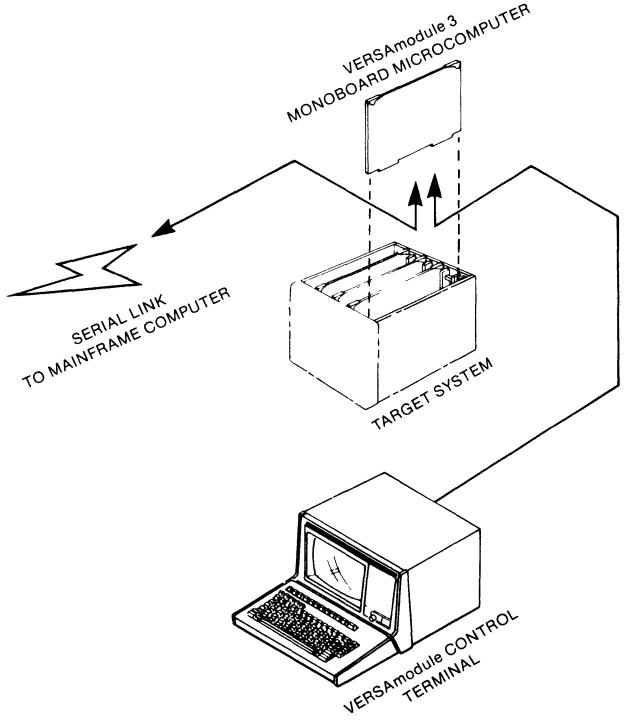
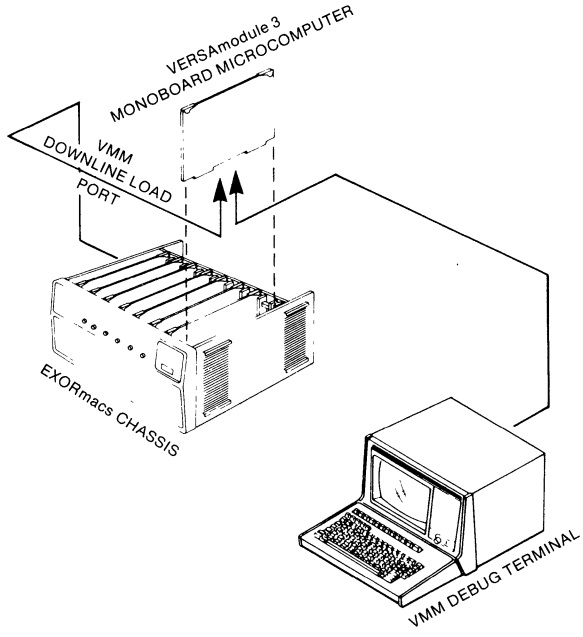


FIGURE 2 — Configuration #2 — Monoboard with VERSAbug in EXORmacs



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

2

Memory Management Board

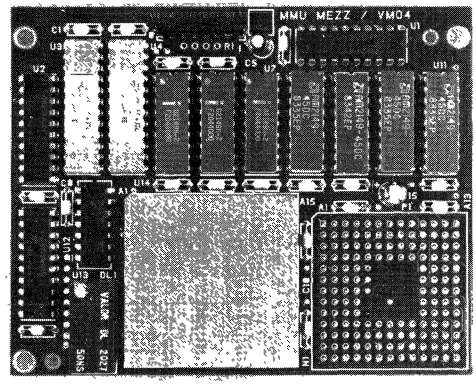
The M68KVMMB Memory Management Board (MMB) is designed to provide hardware support to MC68020/68010 users wanting to implement demand paged virtual memory systems prior to the availability of the MC68851. The MMB contains a gate array implementation of a table walking algorithm of the MC68851 and a 512 word set associative address translation cache. It is a subset of the MC68851 and is compatible to the MC68851 so that algorithms and programs developed using the MMB in a MC68020 system can be ported to the MC68851 with minor changes.

The MMB includes features that provide power and versatility for the user:

- Logical to Physical Address Translation by Chip Controlled Searching of Translation Tables in Main Memory.
- Dynamically Alterable Tree Structured Translation Tables.
- Address Translation Cache.
- 1K Byte Page Size.
- Used and Modified Bits for Pages Maintained in Translation Tables.
- Flat Address Space.
- Support of Sharing Pages Between Processes.
- Logical Address Consisting of 3-bit Function Code and 32-bit Address.
- Support for Optional 24-bit Address and for the MC68010.
- 32-bit Physical Address Output.

The MMB consists of:

- A printed circuit board containing the Memory Management Controller MC68461 and external address translation cache. The printed circuit board is 4.5″ x 3.5″ and contains a pin grid array type connector that allows it to be mounted as a mezzanine board that can be plugged into a socket reserved for the MC68851.



- A gate array Memory Management Controller MC68461 containing the address translation circuitry and state machine that implements the table walking algorithm of the MC68851.
- An external Address Translation Cache (ATC) consisting of a 512 entry set associative cache. The cache is a fast memory that translates logical addresses and function codes to corresponding physical addresses and protection bits. The address translation cache provides a fast mechanism for address translation by avoiding the overhead of searching translation tables on each bus access.

FIGURE 1 — MMB Block Diagram

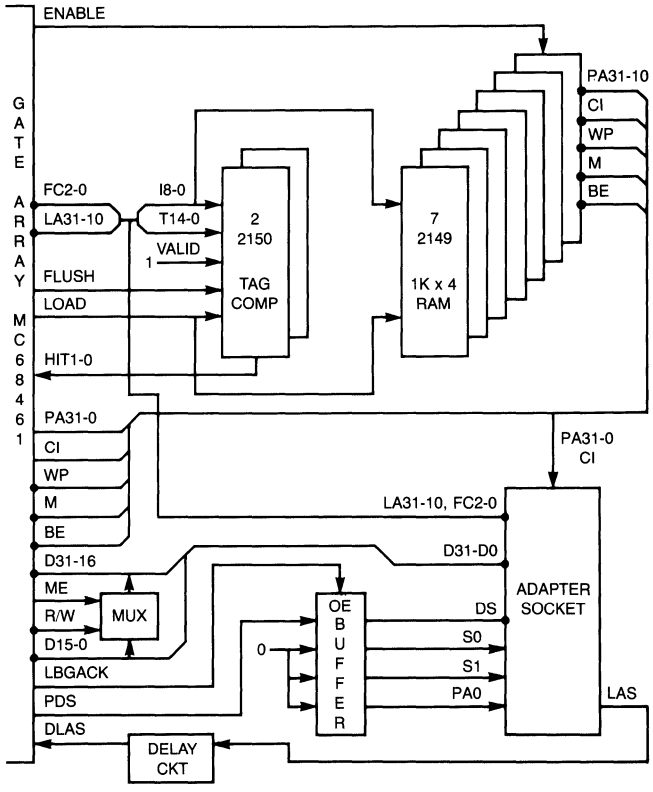
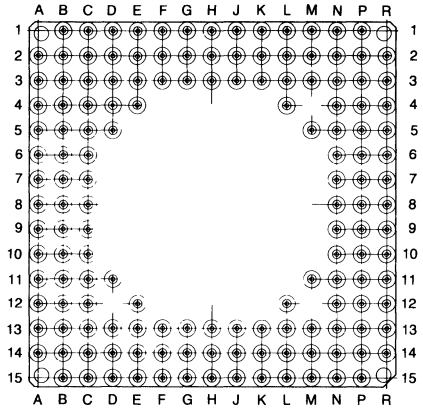
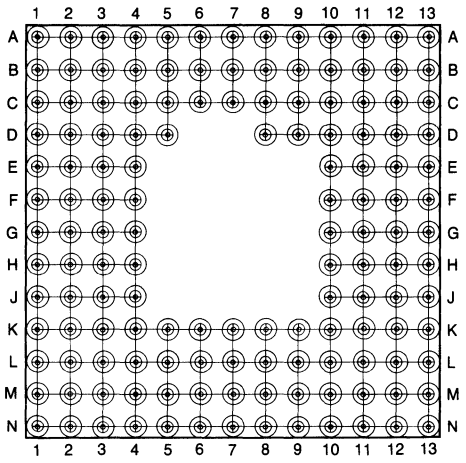


TABLE 1 — 149 – Terminal Pin Grid Array



PIN #	FUNCTION	PIN #	FUNCTION	PIN #	FUNCTION	PIN #	FUNCTION
A1		C9	V_{SS}	H2	LA24	N6	V_{SS}
A2	PLDS L	C10	V_{SS}	H3	V_{CC}	N7	V_{SS}
A3	RESET L	C11	R/W	H13	V_{CC}	N8	V_{CC}
A4	DSACK1 L	C12	ENABLE L	H14	FC2	N9	V_{SS}
A5	DSACK0 L	C13	HALT L	H15	PA3	N10	V_{SS}
A6	10/20 L	C14	PA 16	J1	D19	N11	PA13
A7	BERR L	C15	PBR L	J2	D23	N12	PA15
A8	HIT0	D1	LA27	J3	V_{CC}	N13	LA14
A9	BE	D2	CLI L	J13	V_{CC}	N14	PA18
A10	WP	D3	PA27	J14	PA10	N15	PA11
A11	CLK	D4	INDEX	J15	FC0	P1	D22
A12	LAS L	D5	V_{CC}	K1	LA29	P2	PA26
A13	M	D11	V_{CC}	K2	PA29	P3	LA22
A14	LOAD L	D13	N C	K3	V_{SS}	P4	D27
A15		D14	CS L	K13	V_{SS}	P5	D21
B1	D7	D15	D3	K14	D13	P6	LA30
B2	PDSUDS L	E1	D8	K15	LA10	P7	LA21
B3	PA1	E2	D29	L1	D28	P8	D18
B4	FLUSH L	E3	D31	L2	D25	P9	D17
B5	PA17	E4	N C	L3	D26	P10	LA13
B6	HIT1	E5	V_{SS}	L4	V_{SS}	P11	LA12
B7	DS L	E12	V_{SS}	L12	V_{SS}	P12	PA12
B8	D1	E13	D2	L13	LA19	P13	D16
B9	PA2	E14	FC1	L14	D14	P14	D10
B10	LBGI L	E15	PBG L	L15	LA18	P15	LA11
B11	PBGACK L	F1	D0	M1	LA28	R2	D24
B12	TIME L	F2	PAB	M2	PA24	R3	D20
B13	PA5	F3	V_{SS}	M3	PA22	R4	LA23
B14	PA4	F13	V_{SS}	M5	V_{CC}	R5	PA21
B15	LBRO L	F14	LBGACK L	M11	V_{CC}	R6	LA26
C1	PA7	F15	PA9	M13	D9	R7	PA31
C2	PA6	G1	LA16	M14	D31	R8	D12
C3	D6	G2	LA01	M15	D15	R9	PA19
C4	PAS L	G3	V_{CC}	N1	PA28	R10	LA20
C5	ME L	G13	V_{CC}	N2	D30	R11	LA25
C6	V_{SS}	G14	D5	N3	PA23	R12	PA20
C7	V_{SS}	G15	D4	N4	PA30	R13	LA15
C8	V_{CC}	H1	LA17	N5	PA25	R14	PA14

TABLE 2 — MMB Adapter Socket Table



PIN #	FUNCTION	PIN #	FUNCTION	PIN #	FUNCTION	PIN #	FUNCTION
A1	PBR L	C11	PA04	G10	V_{SS}	L4	D23
A2	LBRI L	C12	PA05	G11	PA23	L5	D20
A3	PAS L	C13	PA06	G12	PA22	L6	D17
A4	V_{SS}	D1	LA29	G13	PA21	L7	D14
A5	LAS L	D2	LA30	H1	LA17	L8	D11
A6	ECS L	D3	LA31	H2	LA18	L9	D08
A7	R/W	D4	V_{SS}	H3	LA19	L10	D05
A8	SIZE0	D5	V_{SS}	H4	V_{CC}	L11	D02
A9	FC1	D6	N C	H10	V_{CC}	L12	PA11
A10	DSACK0 L	D7	N.C.	H11	PA20	L13	PA10
A11	DSACK1 L	D8	V_{CC}	H12	PA19	M1	LA09
A12	CLK	D9	V_{SS}	H13	PA18	M2	D30
A13	PA00	D10	V_{SS}	J1	LA14	M3	D27
B1	WIN	D11	PA07	J2	LA15	M4	D24
B2	PBGACK L	D12	PA31	J3	LA16	M5	D21
B3	LBRO L	D13	PA30	J4	V_{SS}	M6	D18
B4	LBGACK L	E1	LA26	J10	V_{SS}	M7	D15
B5	CLI L	E2	LA27	J11	PA17	M8	D12
B6	RESET L	E3	LA28	J12	PA16	M9	D09
B7	HALT L	E4	V_{SS}	J13	PA15	M10	D06
B8	DS L	E10	V_{SS}	K1	LA11	M11	D03
B9	FC0	E11	PA29	K2	LA12	M12	D00
B10	BERR L	E12	PA28	K3	LA13	M13	PA09
B11	PA01	E13	PA27	K4	V_{SS}	N1	LA08
B12	PA02	F1	LA23	K5	V_{SS}	N2	D31
B13	PA03	F2	LA24	K6	V_{CC}	N3	D28
C1	ASYNC L	F3	LA25	K7	V_{CC}	N4	D25
C2	10/20 L	F4	V_{CC}	K8	V_{CC}	N5	D22
C3	LBGO L	F10	V_{CC}	K9	V_{SS}	N6	D19
C4	DBDIS	F11	PA26	K10	V_{SS}	N7	D16
C5	PBG L	F12	PA25	K11	PA14	N8	D13
C6	CS L	F13	PA24	K12	PA13	N9	D10
C7	LBGI L	G1	LA20	K13	PA12	N10	D07
C8	RMC L	G2	LA21	L1	LA10	N11	D04
C9	SIZE1	G3	LA22	L2	D29	N12	D01
C10	FC2	G4	V_{SS}	L3	D26	N13	PA08

TABLE 3 — Summary of Differences Between the MC68851 and M68KVMMB

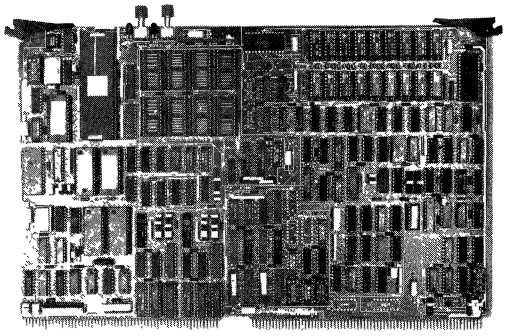
Feature	MMB	MC68851
Packaging	Gate Array Plus Support Logic	Single Pin Grid Array Device
Address Translation Cache	External, Set Associative	Internal, Fully Associative
Page Size	1K Byte	256, 512, 1K, 2K, 4K, 8K, 16K & 32K Byte
Logical Address Support	24 or 32 Bit	From 17 to 32 Bits
Supports Multiple Logical Bus Masters	No	Yes
Logical and Physical Bus Arbitration Support	Yes, But Not Identical to the MC68851	Yes
Access Level Support	Does Not Support Supervisor Root Pointer, Root Pointer Cache or DMA Root Pointer	Yes
Coprocessor Interface Support	No, Operations are Performed by Accessing Memory Mapped Locations	Yes
Translation Table Configuration	Expects One Given Table Walking Algorithm	Supports Several
Descriptor Types and Descriptor Status Bits	Some	All
Translates CPU Space Accesses	No, Acts as Peripheral Only	Yes

ORDERING INFORMATION

Part Number	Description
M68KVMMB	Memory Management Board. Includes User's Manual.
M68KVMMB/D1	Memory Management Board User's Manual.

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**
Current Requirements: Without ROM/EPROM	Typ 5.7 A, Max 8.2 A; Typ 0.5 A, Max 0.8 A; Typ 0.1 A, Max 0.2 A
Add for each EPROM *Pair* (Up to four pairs)	Typ 0.1 A, Max 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	(OFF-BOARD)	FFFFFF
F70032		F70033
F70030	STATUS/CONTROL REGISTER	F70031
F7002E	PARALLEL PORTS	F7002F
F70020		F70021
F7001E		F7001F
	SERIAL PORT #2	F70019
		F70017
	SERIAL PORT #1	F70011
		F7000F
F70000	PTM	F70001
F6FFFE	(OFF-BOARD)	F6FFFF
F10000		F10001
F0FFFE	16K/32K/64K ON-BOARD EPROM/ROM OR	F0FFFF
F00000	32K/64K ON-BOARD RAM	F00001
EFFFFE	(OFF-BOARD)	EFFFFF
010000		010001
00FFFE	32K/64K ON-BOARD RAM OR	00FFFF
000008	16K/32K/64K ON-BOARD EPROM/ROM	000009
000006	ON-BOARD RESET VECTORS (ROM)	000007
000000		000001
← MOST SIGNIFICANT BYTE →		← LEAST SIGNIFICANT BYTE →

FIGURE 2 — Block Diagram

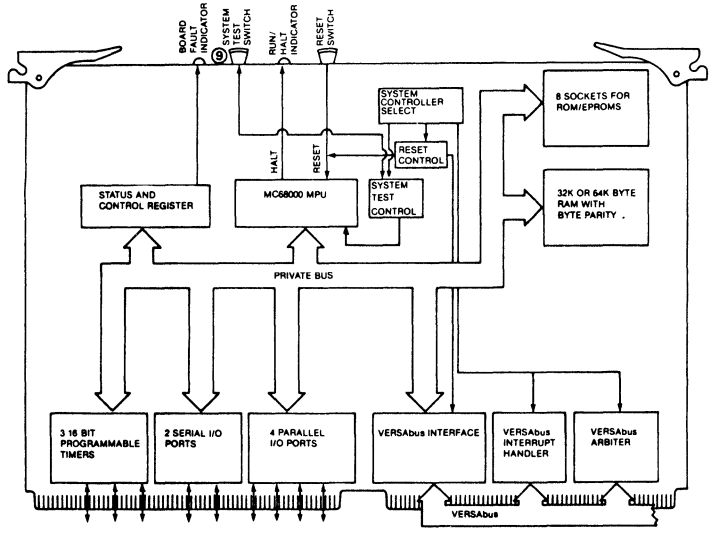


TABLE 1 — Specifications

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)	
Baud Rates	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:

Programmable Baud Rates	
50	1800
75	2000
110	2400
134 5	3600
150	4800
300	7200
600	9600
1200	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)

Characteristics	Specifications
Programmable Timer/Counter (PTM) — (contd)	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
Interrupts	
Auto-Vectored, "Local"	Interrupts from the following sources provide auto-vectored interrupts to the MPU.

Interrupt Level	Interrupt Source
1	"Bus Clear" (From VERSAbus)
2	Test Button
3	Interrupter Vector Read
4, 5, or 6*	PTM
4, 5, or 6*	Serial I/O Ports
4, 5, or 6*	Parallel I/O Ports
7	"AC Fail" (from VERSAbus) if System Controller
7	"Bus Release" (from VERSAbus) if not System Controller

	*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
VERSAbus Interrupts	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.)
Interrupter	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.

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	± 0 2 V	—	Guaranteed Low Voltage	—	0 5 V
(−7 V ≤ VI ≤ 7 V)			Voltage When Sinking 48 mA		
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

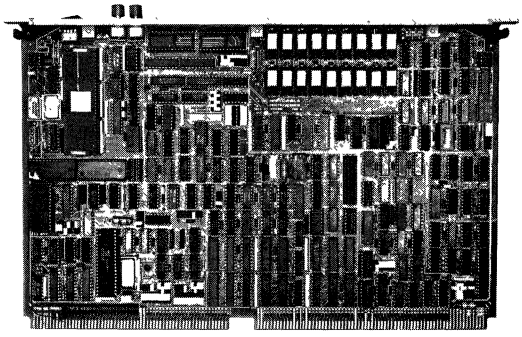
M68RIOCS	I/O Channel Specification Manual
M68KVBS	VERSAbus Specification Manual
MC68000UM(AD-3)	MC68000 Microprocessor User's Manual
M6840UM	MC6840 Fundamentals and Applications Manual

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



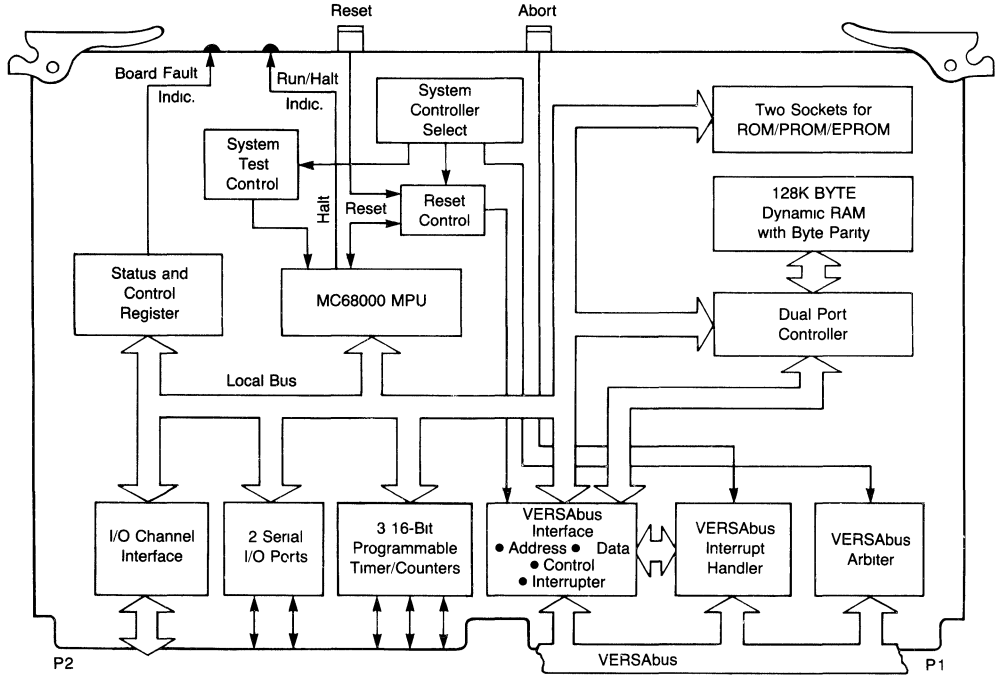
- MC68000 8 MHz 16-bit MPU
  - — 16 32-bit data, address and stack registers
  - — 14 addressing modes
  - — 16 megabyte direct addressing range
  - — Memory mapped I/O
  - — 56 powerful instruction types
  - — Operations on five data types including bit, byte, word, long word and BCD
  - — Provides 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 pulse 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:

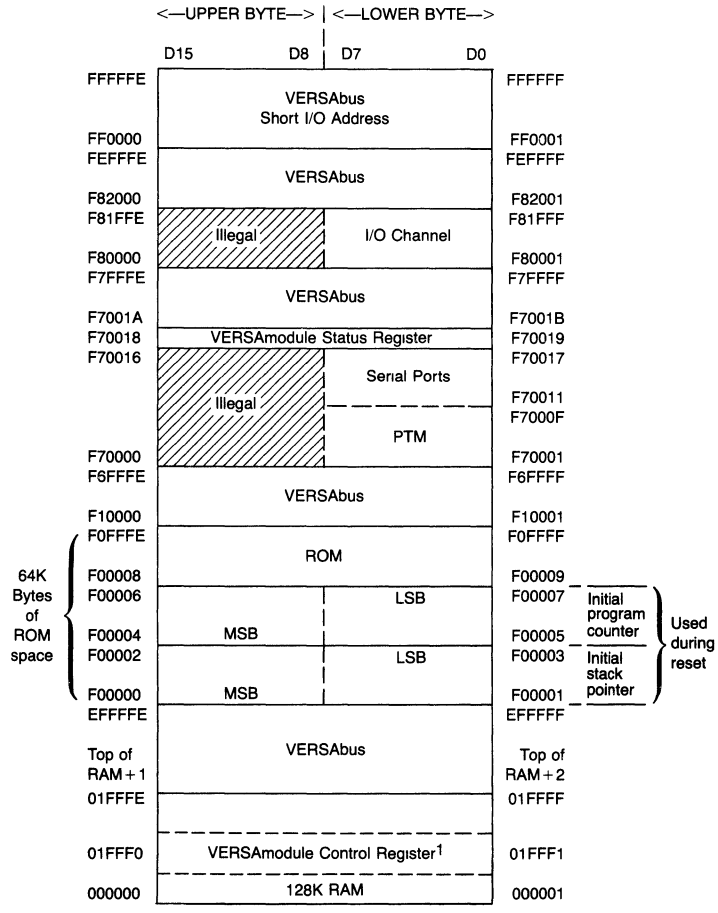
- VERSAbus Arbitration — The arbiter accepts bus requests on five priority levels from other bus masters and grants the bus to the highest priority requester. This function facilitates orderly management of the contention for bus mastership on the VERSAbus.
- System Clock — A 16 MHz clock signal is provided 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

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)

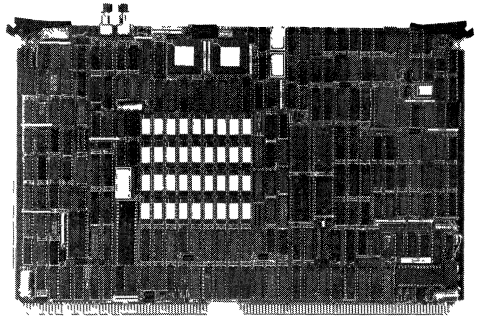
Part Number	Description
M68KVAM	VERSAbus Adapter Module
M68KEXT	VERSAbus Extender Module
M68KWW	VERSAbus Wirewrap Module
MVME400	Dual Channel RS-232C Serial Port
MVME410	Dual Channel 16-bit Parallel Port
MVME420	SASI™ Peripheral Adapter
MVME435	Buffered 9-Track Magnetic Tape Adapter
M68RWIN1-1,2	Winchester 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

VERSAmodule Monoboard Microcomputer

M68KVM03-1
M68KVM03-3
M68KVM03-4
M68KVM03-5



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.

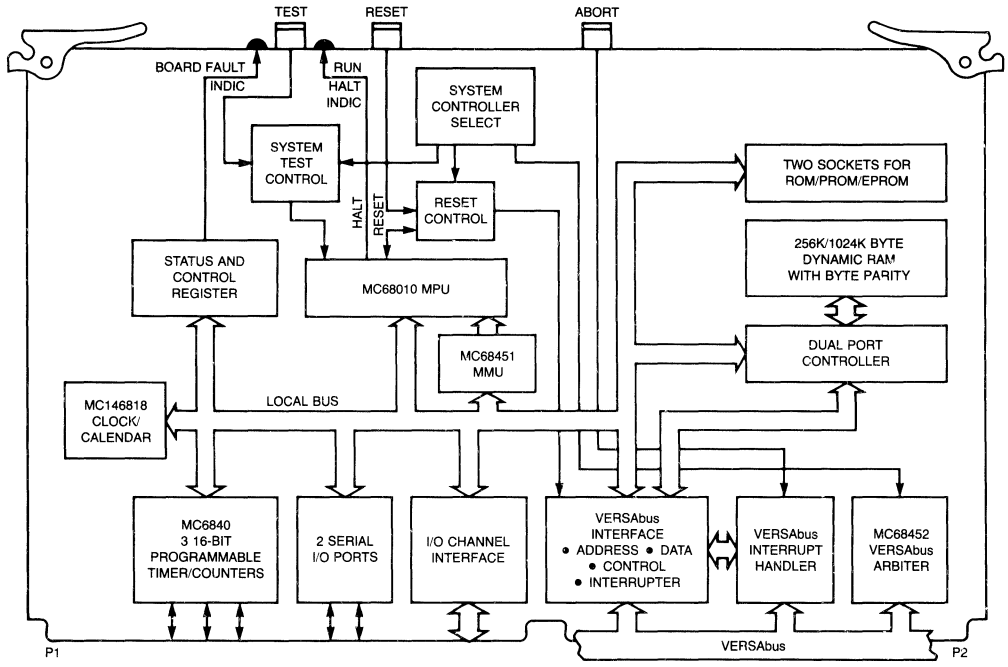
Bus timeout can reach a delay of 64 microseconds before a CPU bus error is generated. Separate delays are provided for local bus and VERSAbus protection. The user may disable the VERSAbus timeout, via a jumper, to use the VM03 in applications where it is not the System Controller.

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 impiementing 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/downline 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

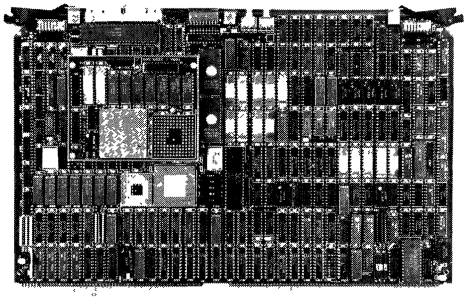
Related Documentation

M68KVMO3/D1	VERSAmodule Monoboard Microcomputer User's Manual.
M68KVBS/D4	VERSabus Specification Manual

M68KVM04-1

2

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.

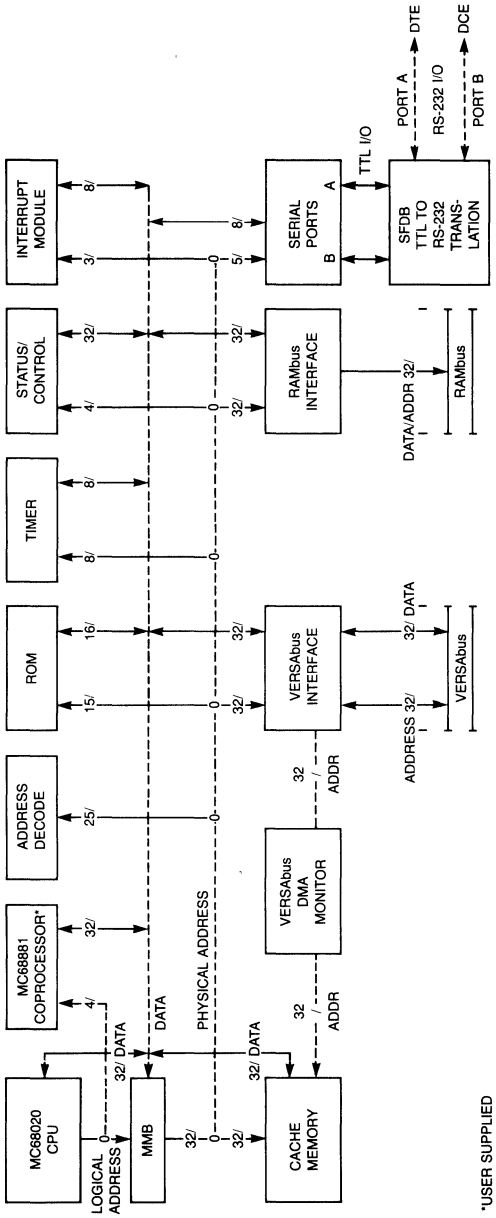


FIGURE 1 — M68KVM04 Block Diagram

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		Group 2 Interrupts —	
ABORT*	<Abort Pushbutton IRQ>	ACFIRQ*	<VERSAbus AC Power Fail Interrupt>
SPIRQ*	<Dual Channel Serial Port IRQ>	BRELIRQ*	<VERSAbus Bus Release Interrupt>
TMRIRQ*	<Timer IRQ>	SYSFIRQ*	<VERSAbus System Fail Interrupt>
OVRIRQ*	<DMA Monitor FIFO Overrun IRQ>	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

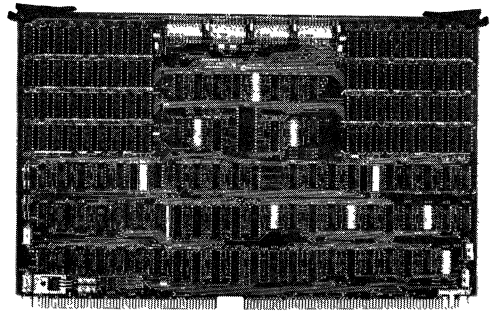
Part Number	Description
M68KVM04/D1	VERSAmodule 32-bit Monoboard Microcomputer User's Manual
TBD	MC68020 MPU Technical Summary
TBD	MC68020 MPU User's Manual

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

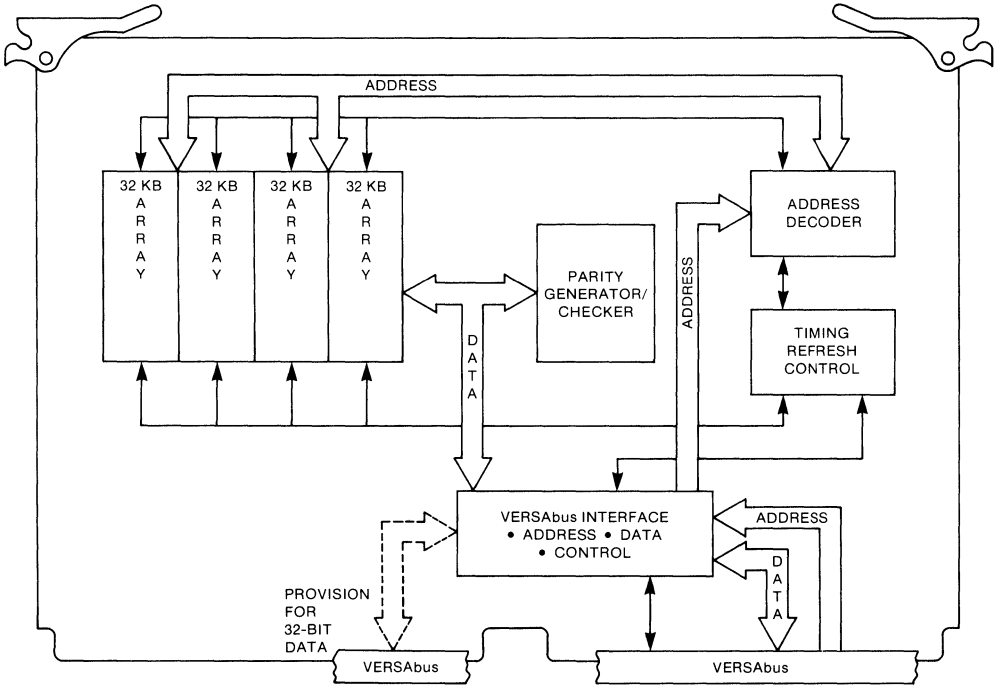
These memory modules are supplied with byte parity. If in the course of a normal read cycle, a parity error is detected, the module will generate a VERSAbus Bus Error (BERR) and transfer it to the system VERSAbus

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)

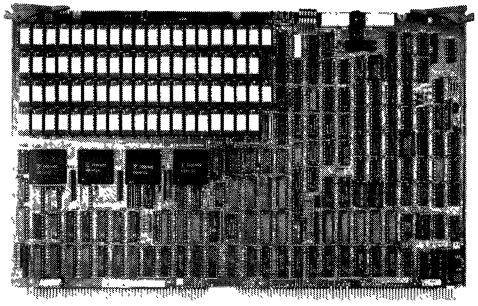
Characteristic	Specification
Relative Humidity	0 To 90% (Non-condensing)
Operating Temperature Range	0° To 70°C
Power Requirements	
64K Byte	+5 0 Vdc @ 3.5 A (Max.) Standby +5 0 Vdc @ 3 5 A (Max) Operating +12 Vdc @ 80 mA (Max.) Standby +12 Vdc @ 700 mA (Max) Operating -12 Vdc @ 7.3 mA (Max) Standby -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 Multidigit 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

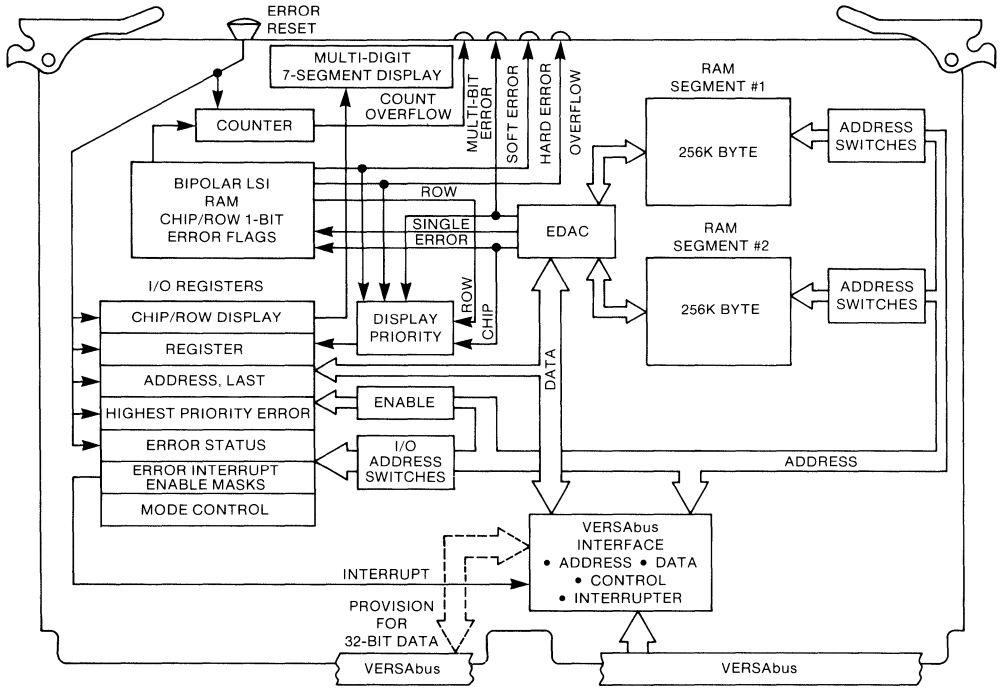
The five board-edge LED's together with a multi-digit 7-segment display and an Error Reset Switch provide manual/visual aid to maintenance personnel during off-line preventative maintenance and repair. The display indicates the board chip position or row number of the highest priority failing RAM chip 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

2

VERSAmodule 256K/512K Byte RAM



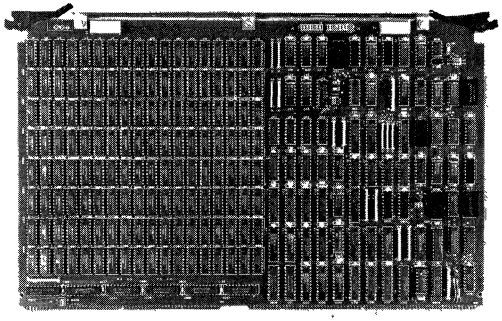
Ordering Information

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

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.

2

FIGURE 1 — M68KVM12 Block Diagram

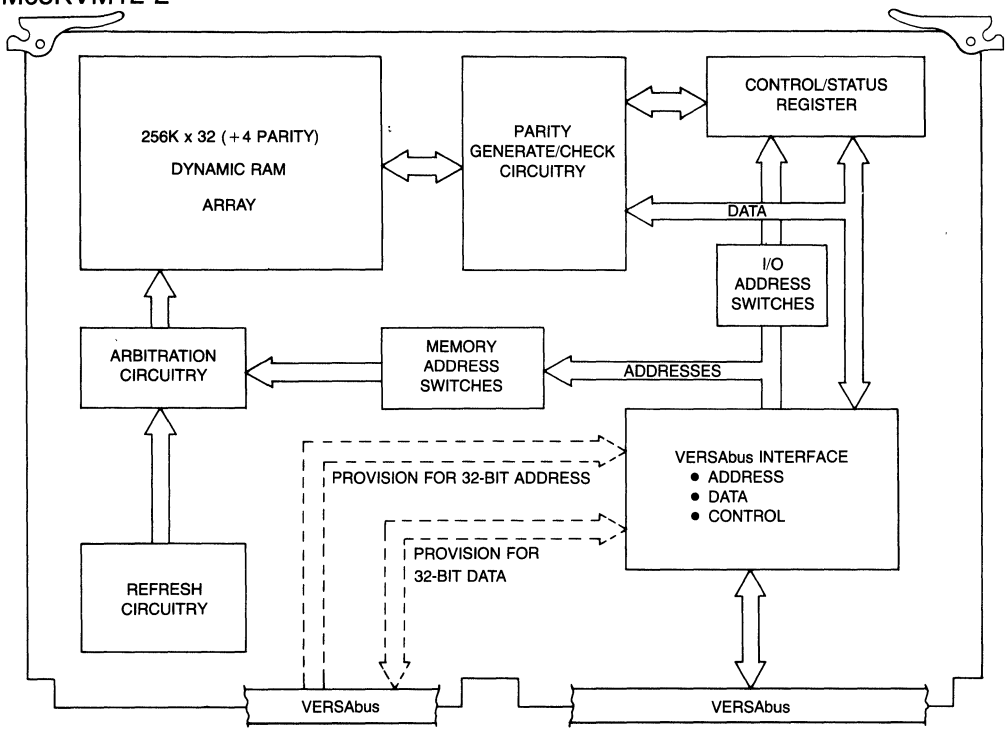
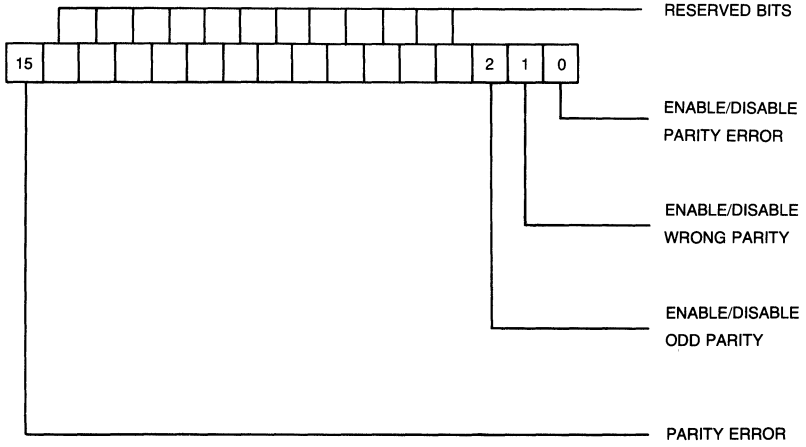


FIGURE 2 — M68KVM12 Control/Status Register



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

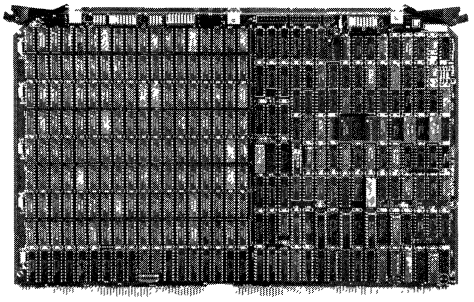
Related Documentation

M68KVBS/D4	VERSAbus Specification Manual

M68KVM13-1
M68KVM13-2

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

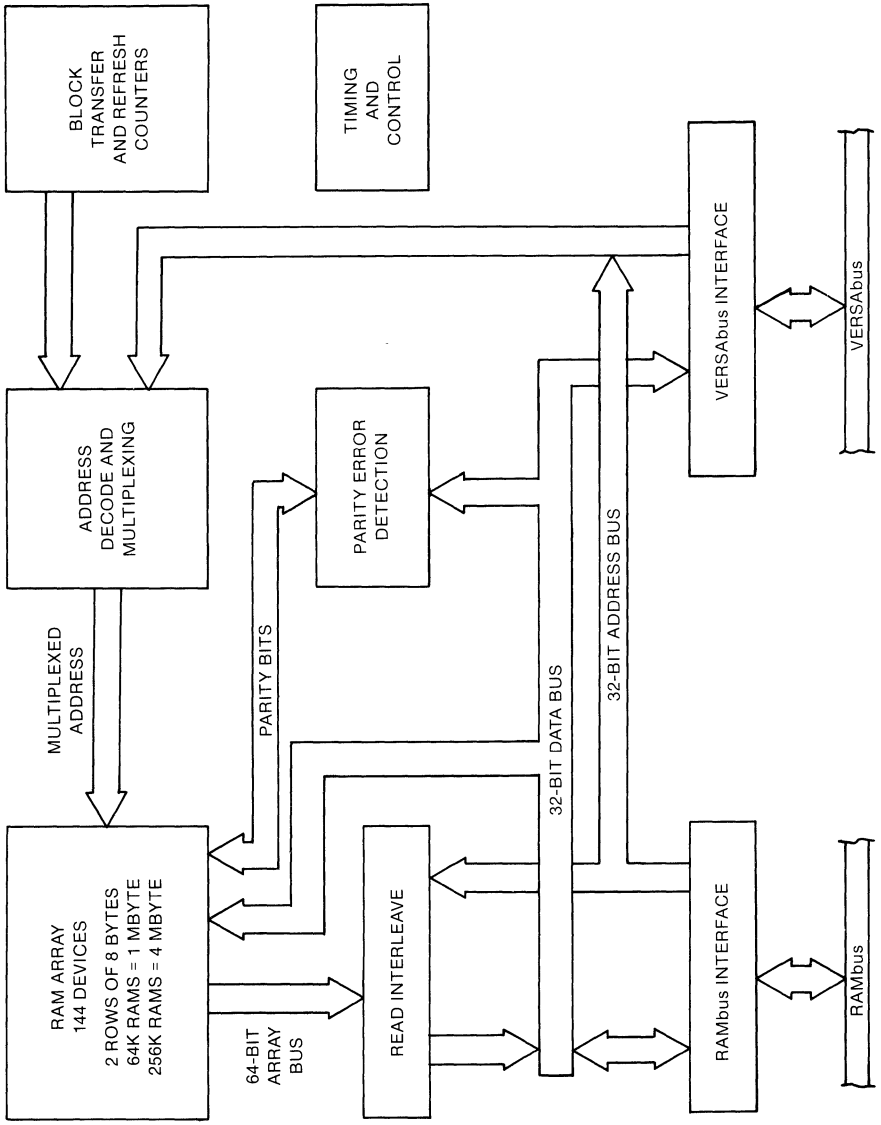


FIGURE 1 — M68KVM13 Block Diagram

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

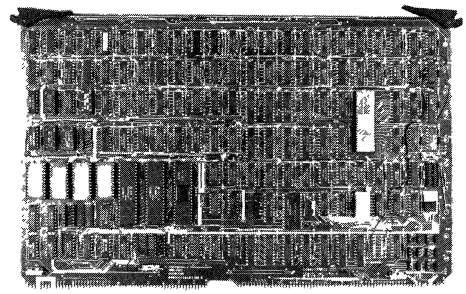
TABLE 1 — M68KVM13 Specifications

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.

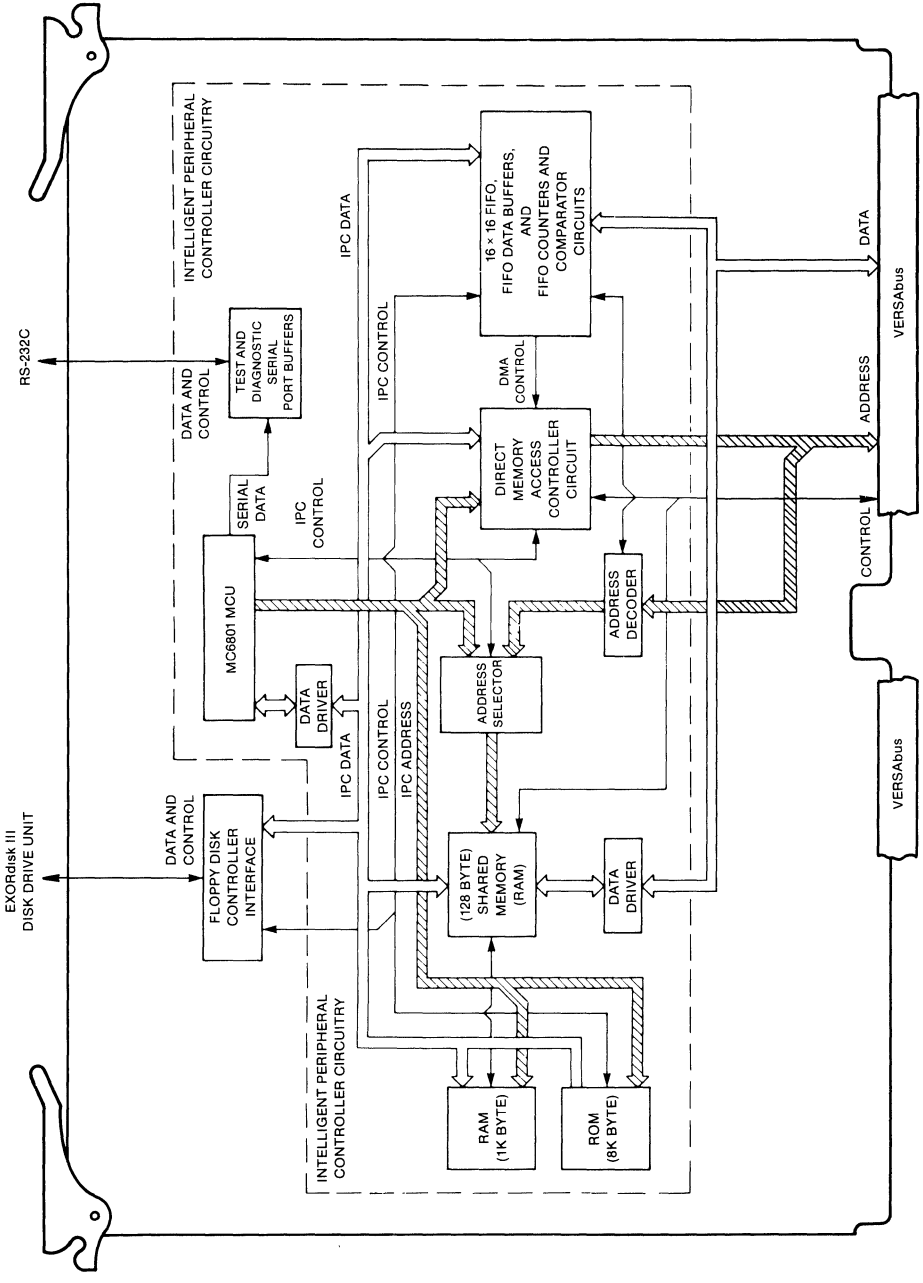


FIGURE 1 — Floppy Disk Controller Functional Block Diagram

FIGURE 2 — Typical Floppy Disk Controller Installation

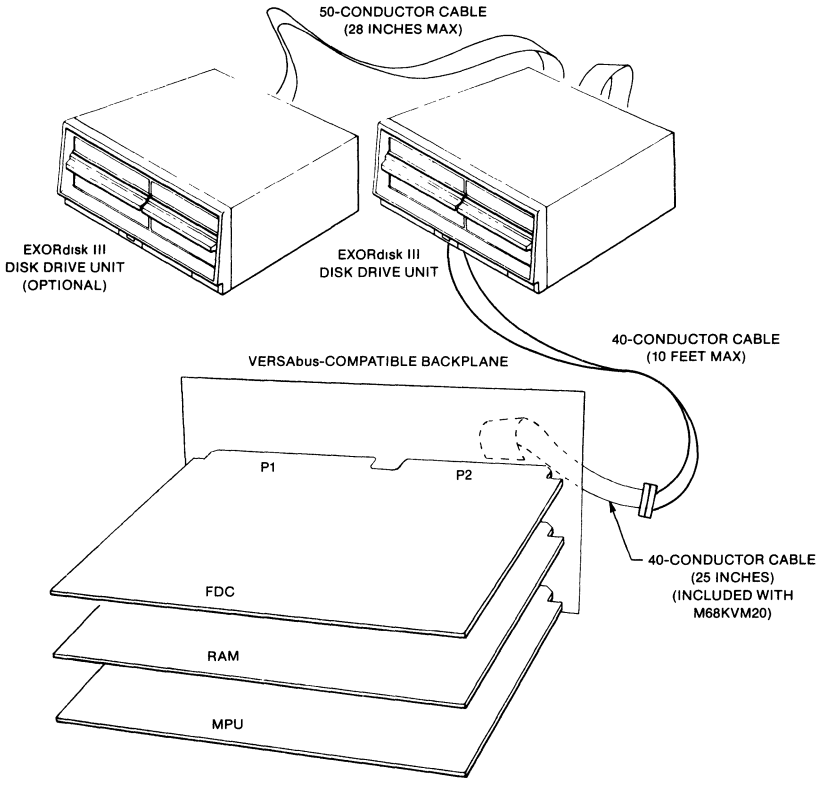


TABLE 1 — Specifications

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

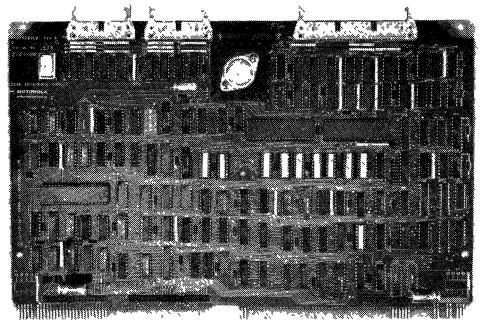
M68KVM20/D1	FDC User's Manual
M68KIPCS/D1	M68000/IPC Command Channel Software Interface Reference Manual
M68KVBS/D4	VERSAbus Specification
M68SFDU3R/D1	EXORdisk II/III Disk Drive Unit Maintenance Manual

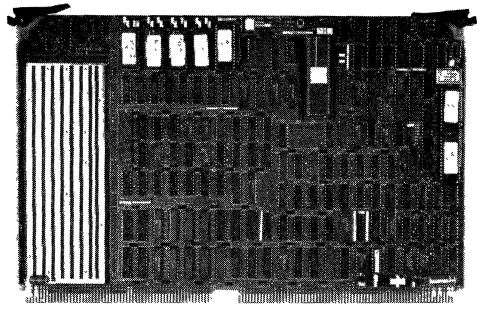
Options

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

2

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

2

FIGURE 1 — Universal Disk Controller (UDC)

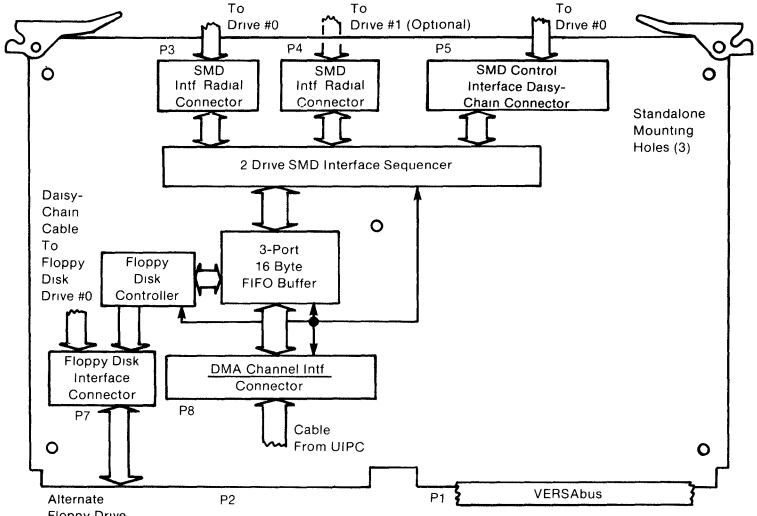


FIGURE 1A — DIM

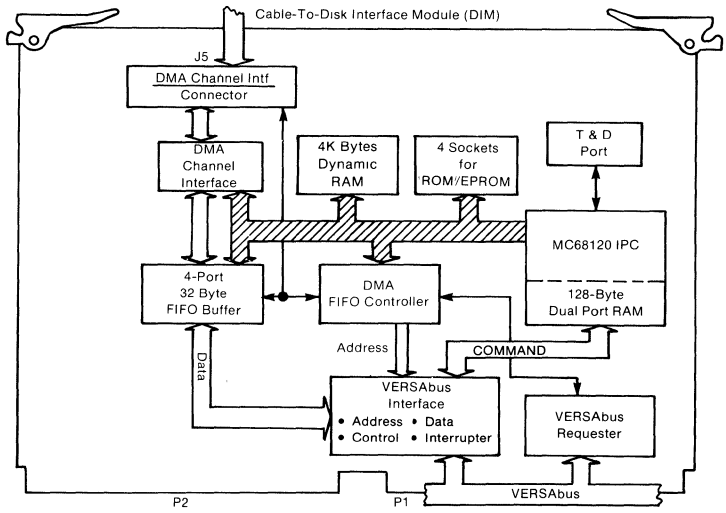


FIGURE 1B — UIPC

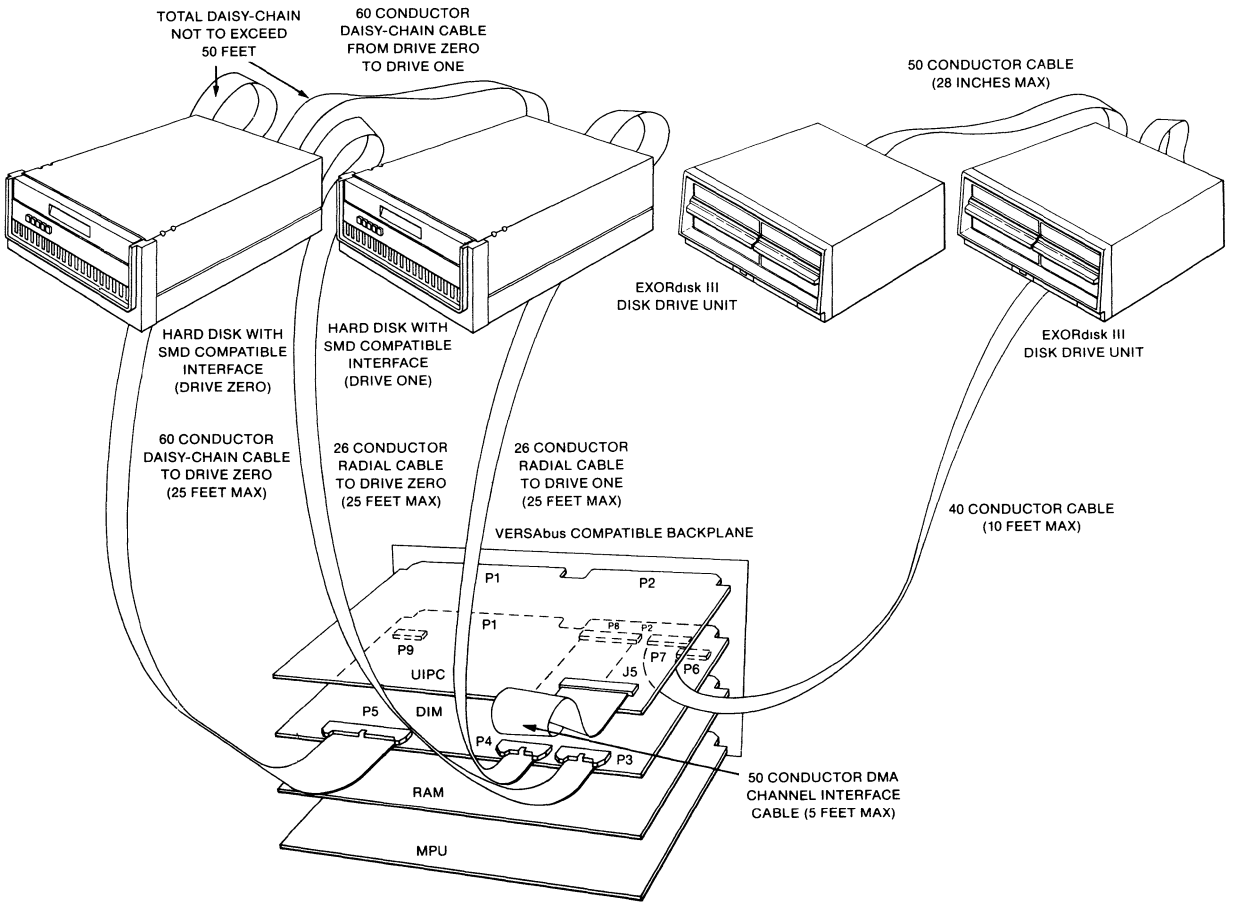


FIGURE 2 — Typical Installation of the Universal Disk Controller

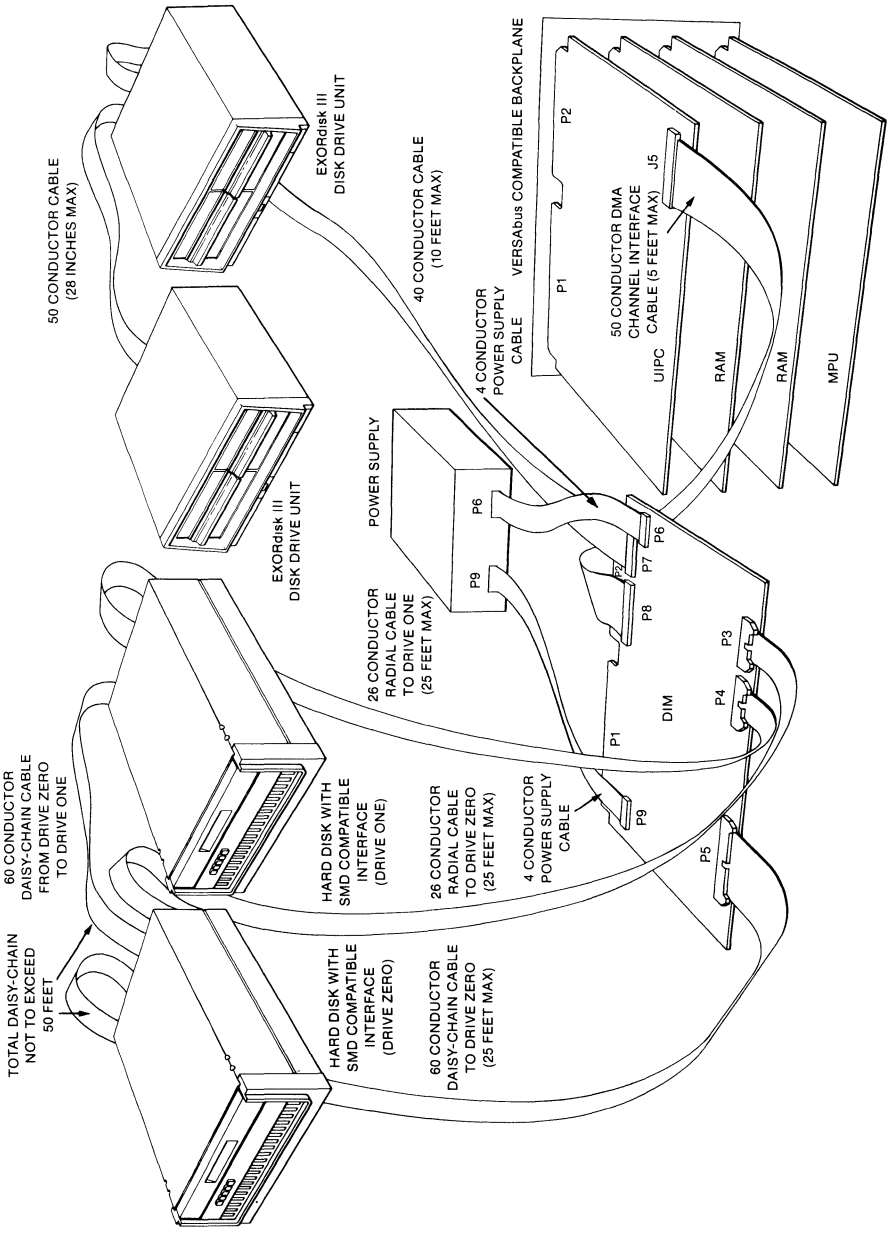


FIGURE 3 — Typical Installation of the Universal Disk Controller Utilizing Standalone DIM

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

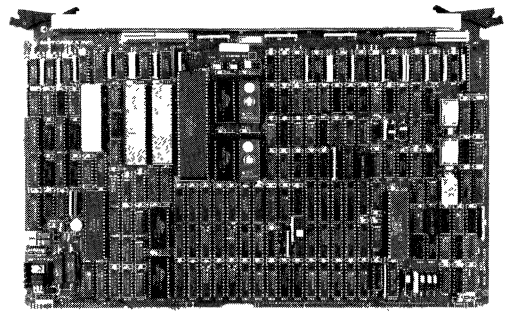
Options

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

2

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

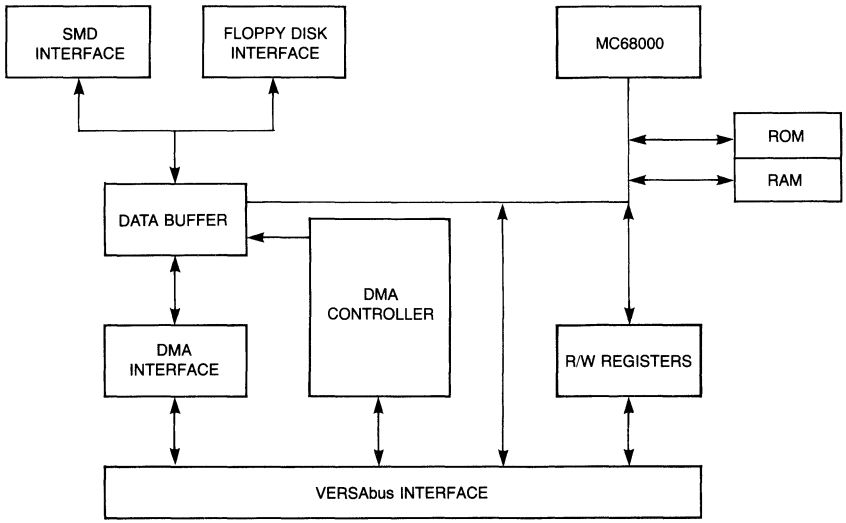


Table 1 — SMD Compatible Disk Drive Parameters

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)

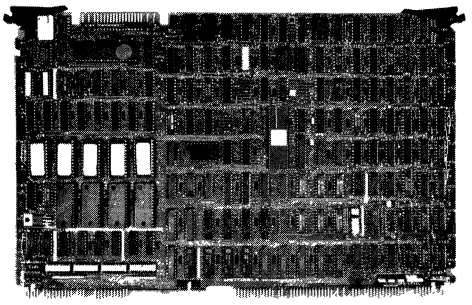
Note:
- The SMD disk format of the VM22 is not compatible with that of the M68KVM21 universal disk controller for the following reasons:
  1. The VM21 uses a 3-byte ID header; the VM22 uses five bytes.
  2. The VM21 uses a 2-byte CRC: the VM22 uses a 4-byte ECC.
- The VM22 also requires a different software 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

2

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.

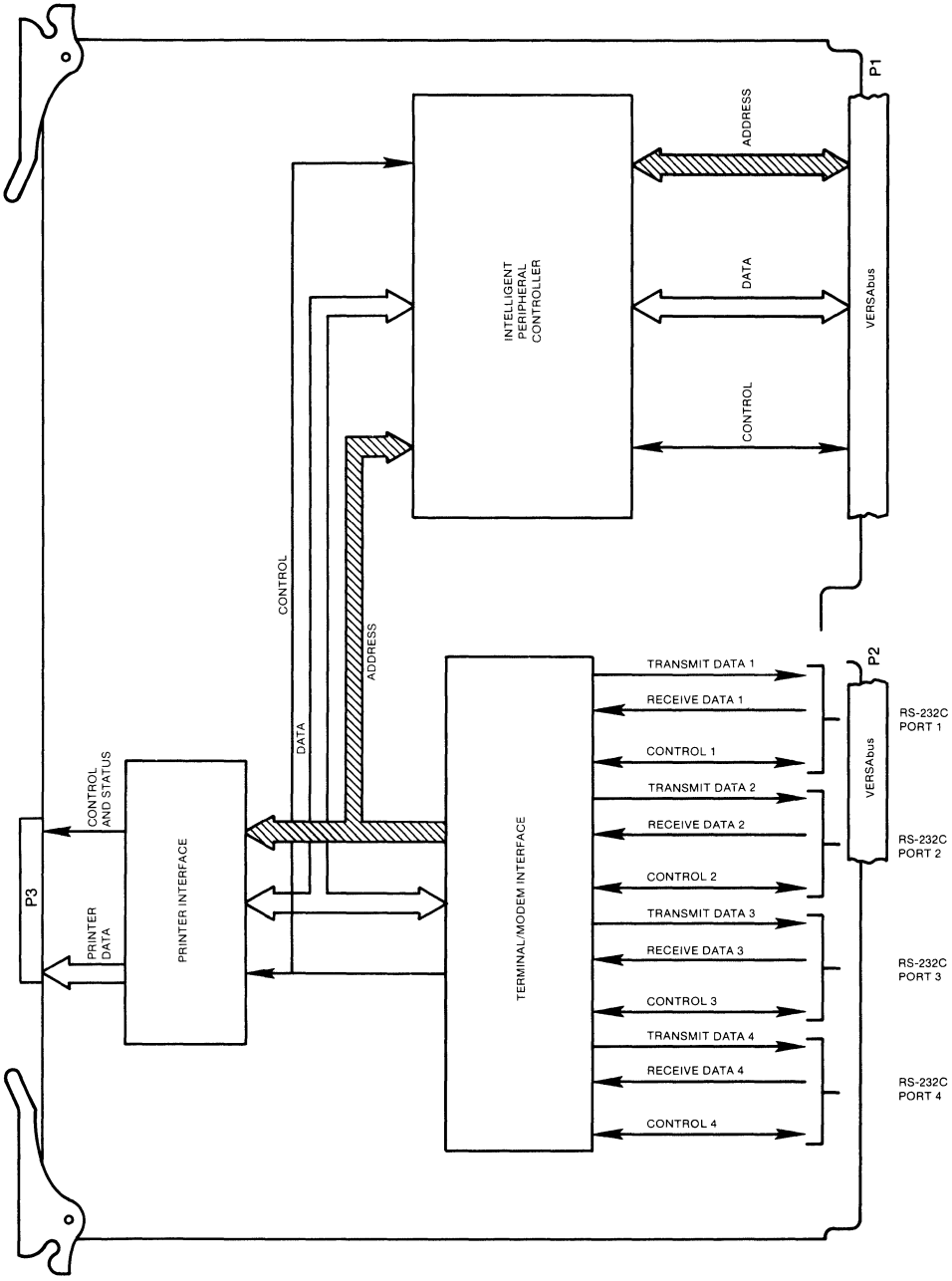


FIGURE 1 — MCCM Functional Block Diagram

2

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

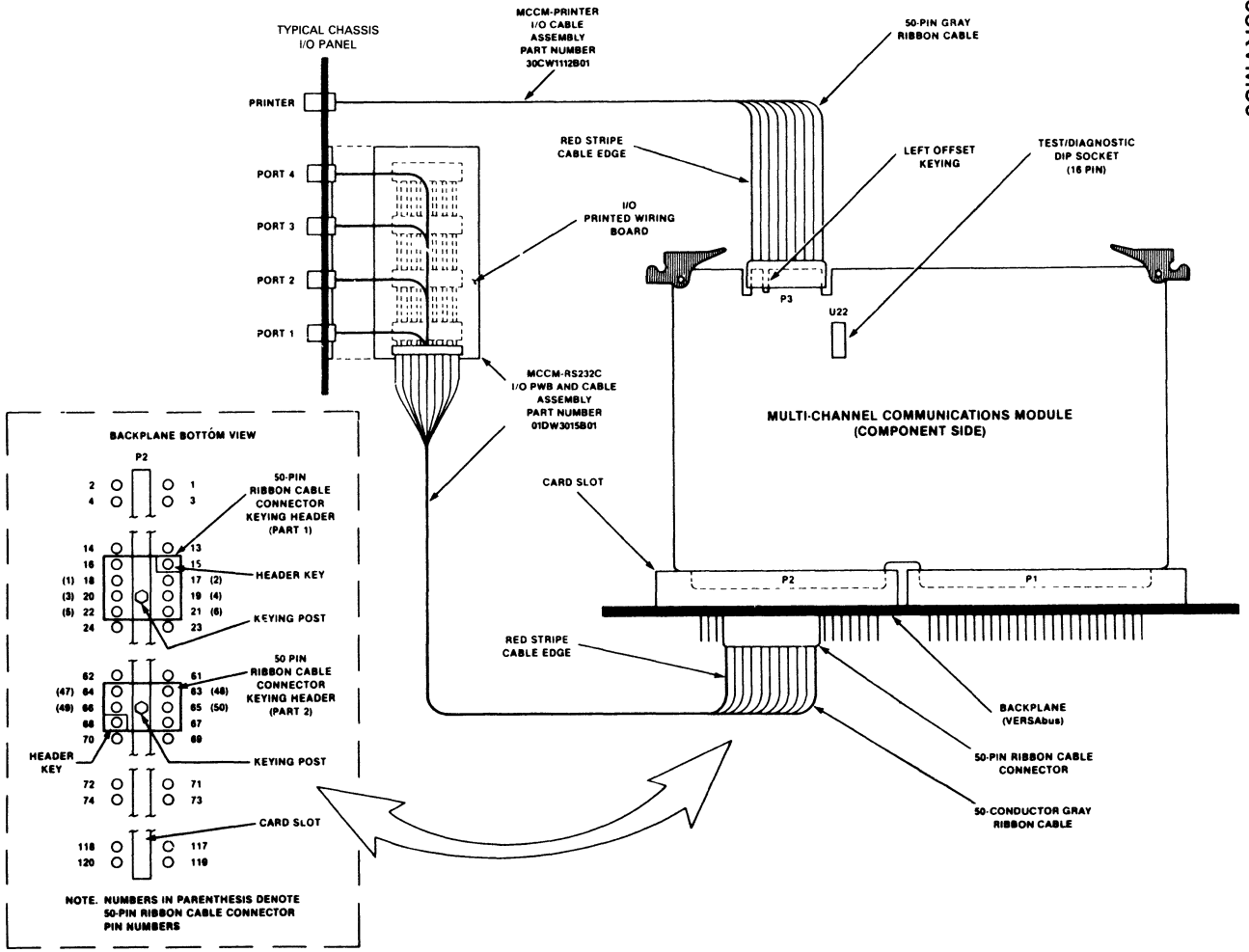
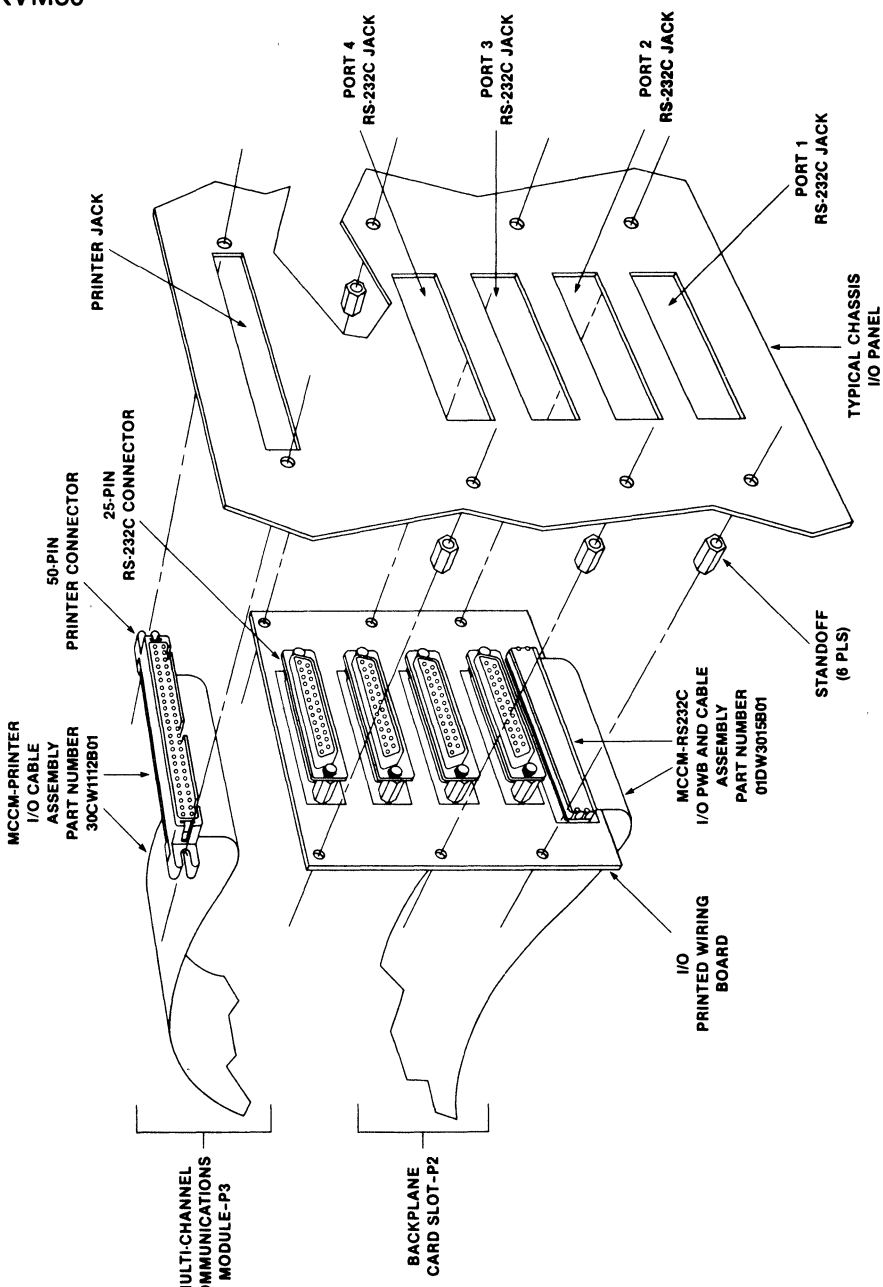


FIGURE 3 — Rear Panel Connector Installation Using The Optional M68KVM232CP Cable Assembly



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

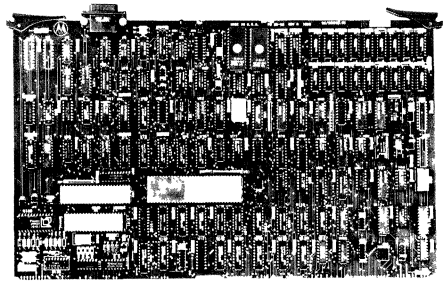
Options

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

M68KVM33

2

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

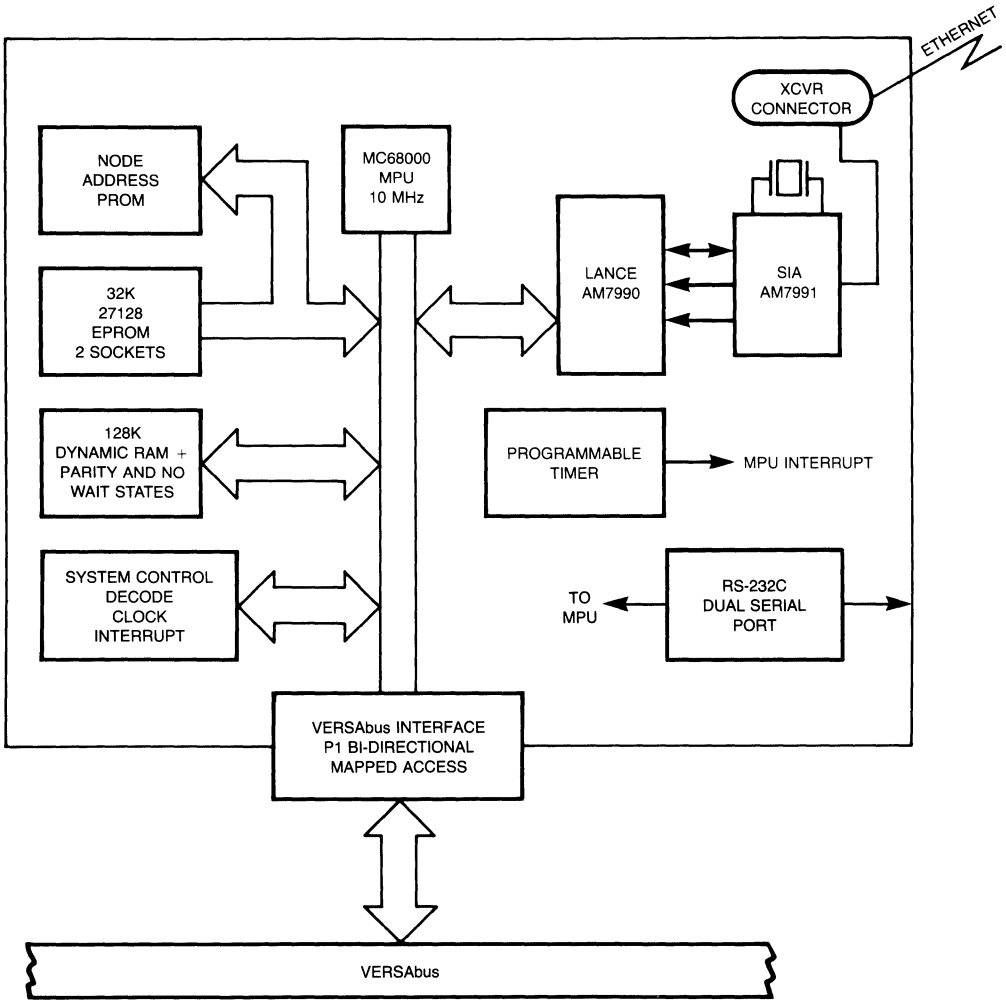
Software Features

- 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

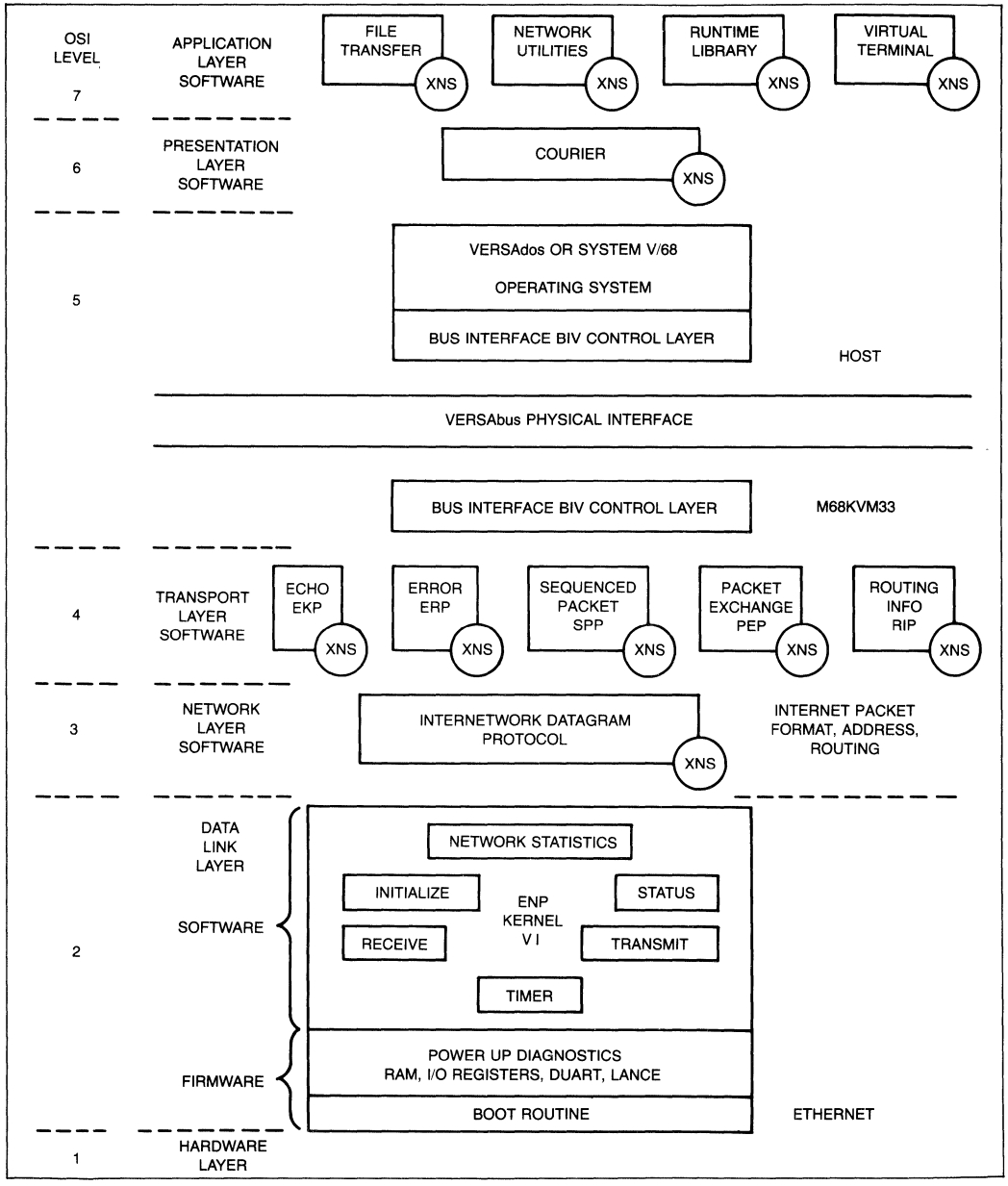


M68KVM33's on-board MC68000 MPU facilitates the downloading of custom protocol implementations. Host-specific drivers also are easily written for interfacing M68KVM33 to another host.

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.

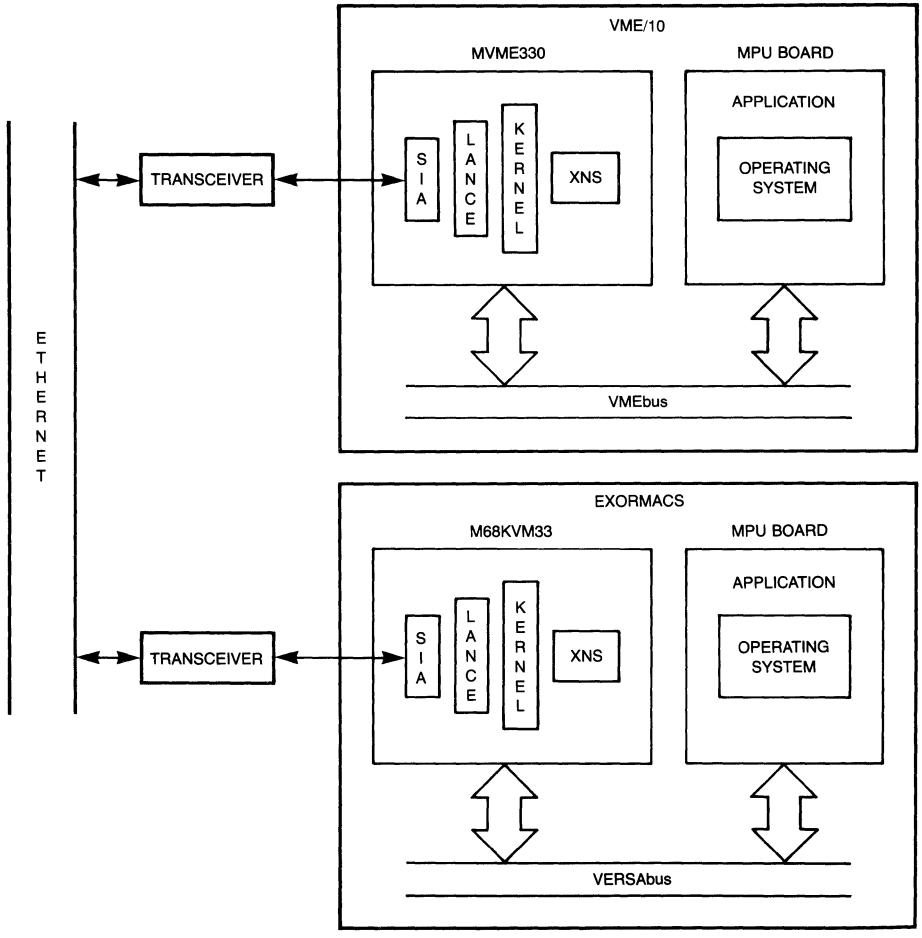
FIGURE 2 — Host/M68KVM33 Ethernet Functions



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



2

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 1/5 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 Systems 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.

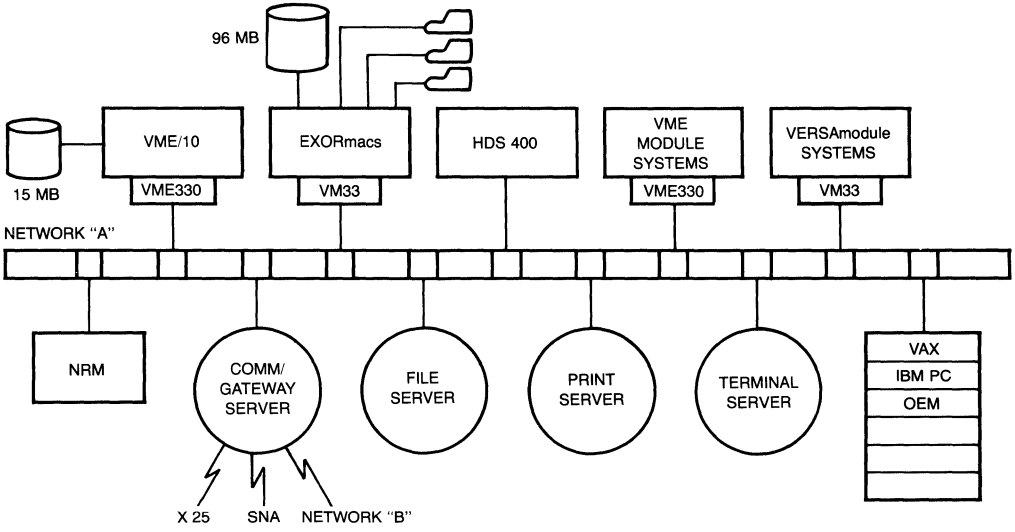
SYSTEMS NETWORK CONCEPT

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

2

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 Micro Plastics Inc.	CPH 7000-140 ST 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

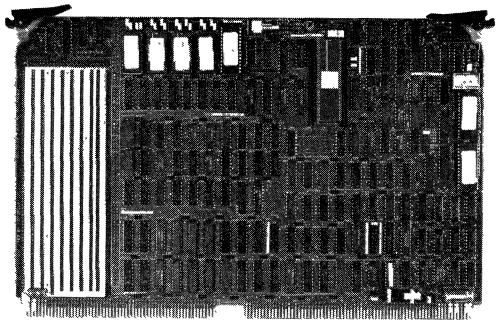
The MVME330 LAN Controller is ordered as a standalone board which includes the executive kernel software; or as a package including the board, executive kernel, a complete set of XNS protocol and network application software, and appropriate host-specific software including device driver and the BIV bus interface. The entire software package (less kernel firmware) may be purchased separately as can the executive kernel firmware and/or the kernel debugger firmware only.

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
M6811RBVMBUG	LAN Controller Debugger in software, kernel, self-test in 32K EPROM (See Note)
M6811RBVMKRN	LAN Controller Kernel, and self test in 16K EPROM
M68NNHSVMLAN for System V/68 Host M68VVHSVMLAN for VERSAdos Host	Source Software supplied on 8" floppy, includes: — Ethernet 2.0 compatible network software device drivers, Host BIV, and M68KVM33 BIV. — Kernel,self-test debugger and documentation (XNS Host-resident utilities and protocol software available from third party vendor).

NOTE: Cable with AMP745094-1 connector, or equivalent, required for serial communications.

2

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.

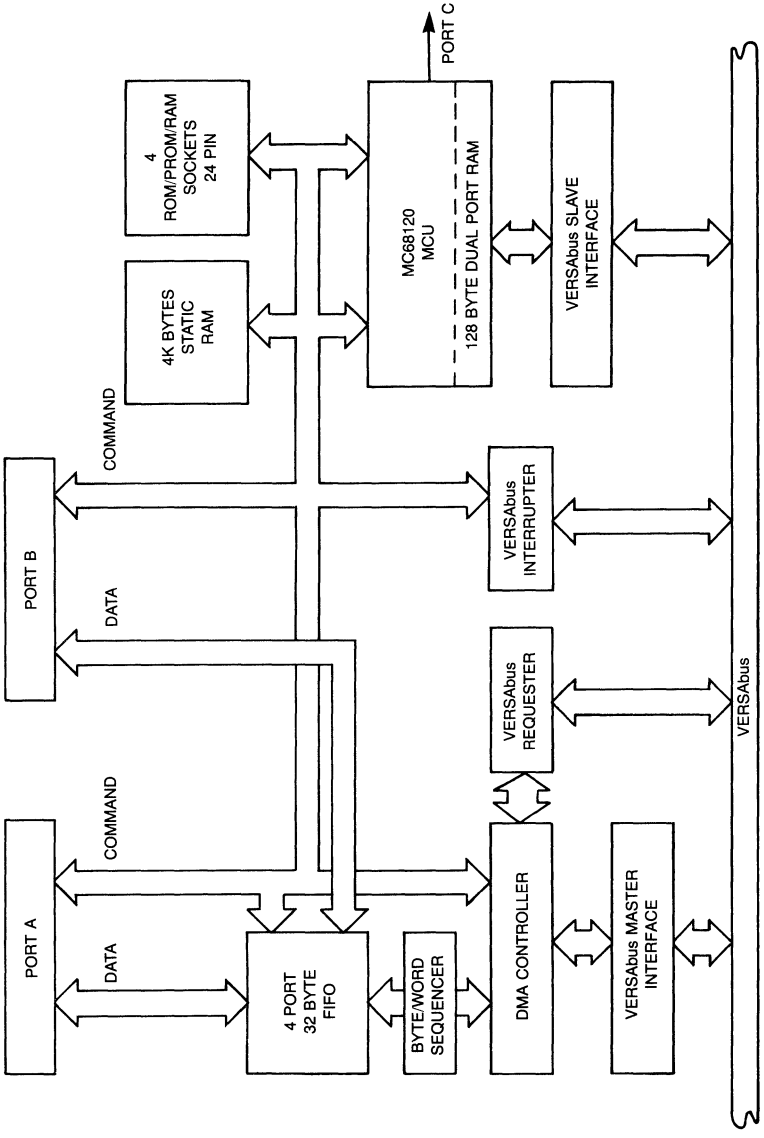


FIGURE 1 — UIPC Functional Block Diagram

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

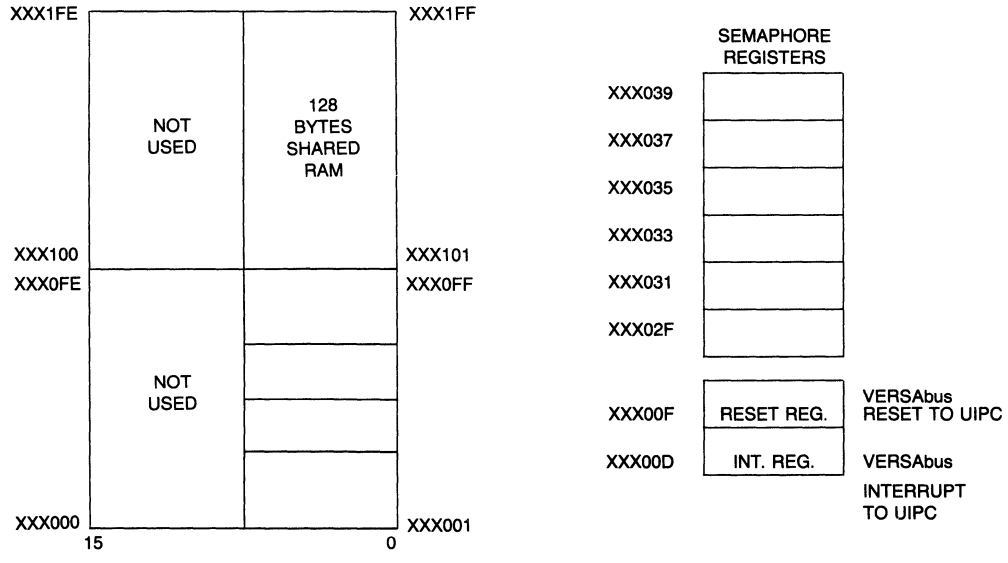


FIGURE 3 — UIPC Memory Map From the MC68120

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.

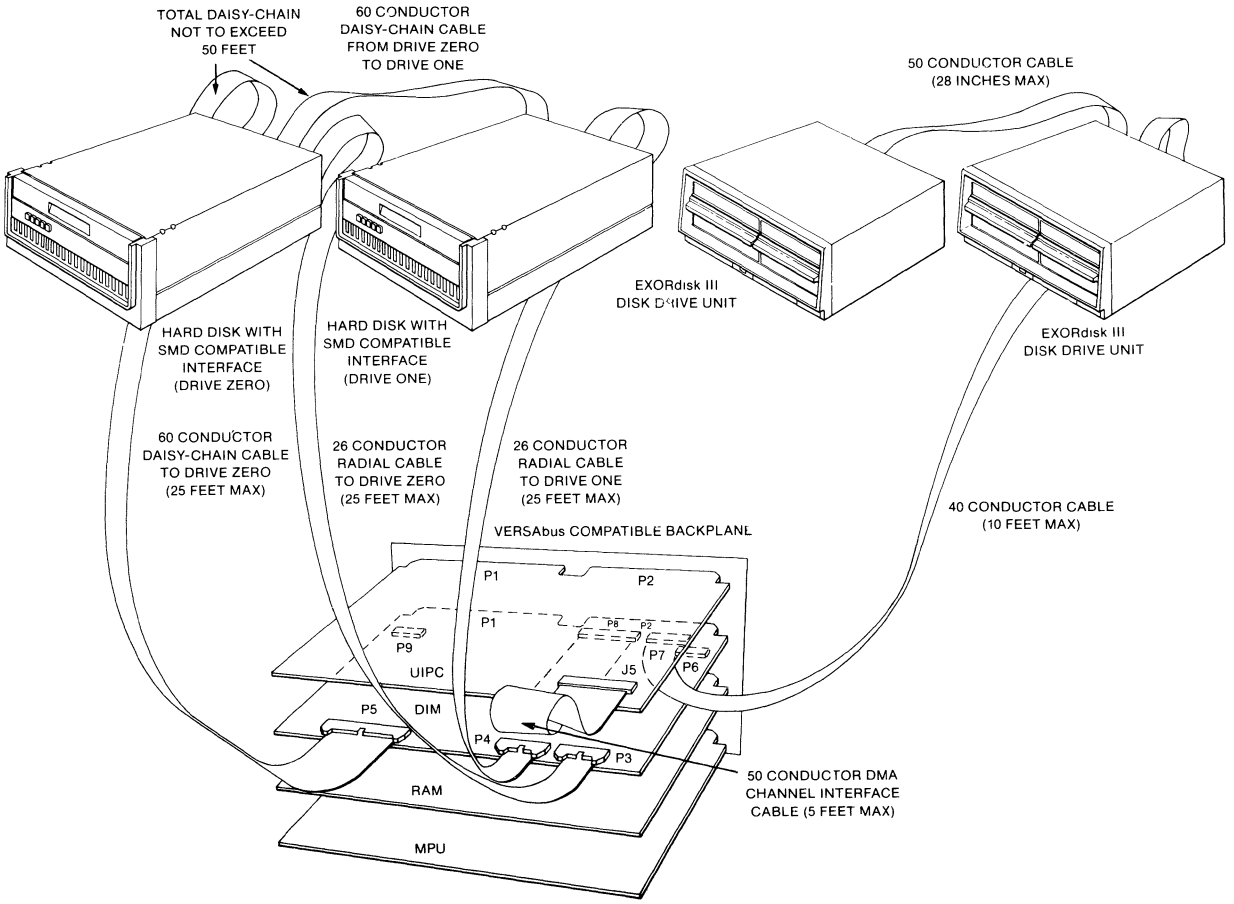


FIGURE 4 — UIPC Interconnection in a Typical Installation of the Universal Disk Controller

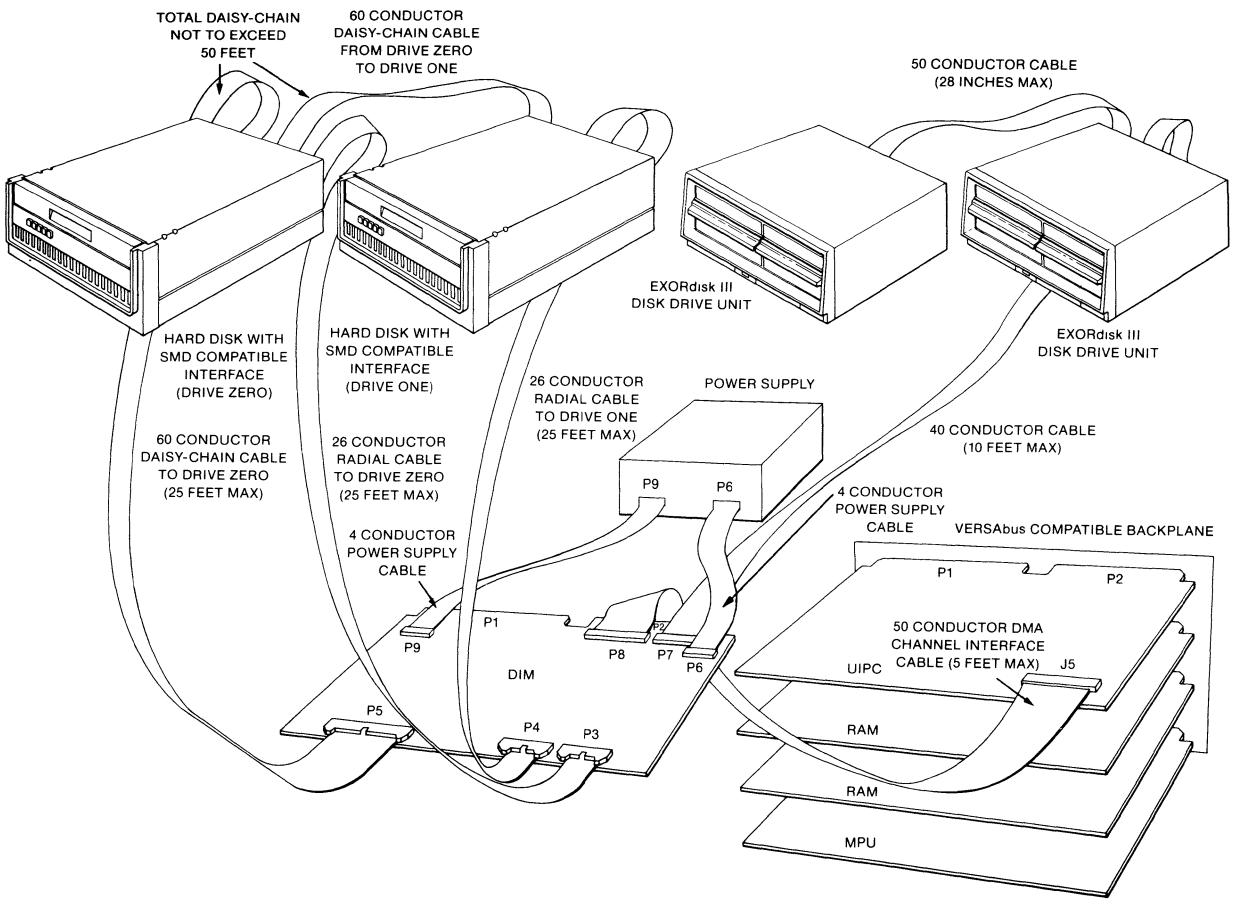


FIGURE 5 — UIPC Interconnection in a Typical Installation of the Universal Disk Controller Utilizing Standalone DIM

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

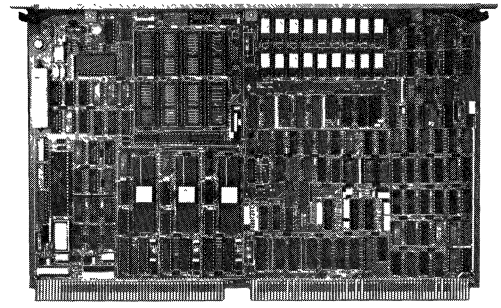
M68KVM60/D2	Universal Intelligent Peripheral Controller Module User's Manual
M68KIPCS	M68000/IPC Command Channel Software Interface Reference Manual
M68KDRVGD	Guide to Writing Device Drivers for VERSAdos

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

Section (Module) of Board	+5 Vdc		+12 Vdc		-12 Vdc		+5 V Stdby	
	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

Section (Module) of Board	+5 Vdc		+12 Vdc		-12 Vdc		+5 V Stdby	
	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	
$FF019B		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

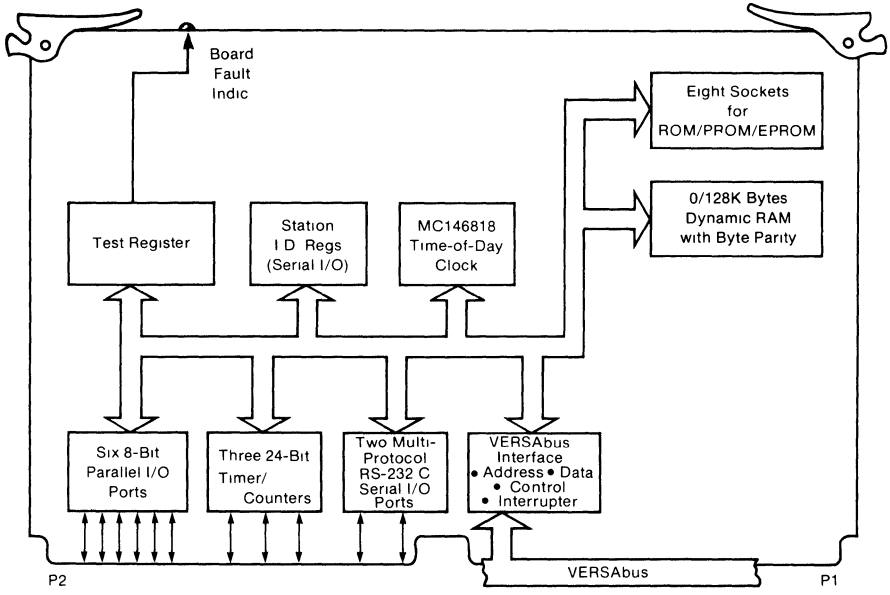
SIGNAL MNEMONIC	PIN NUMBER	TO TERMINAL	TO MODEM	SIGNAL NAME AND FUNCTION
CLOCK1*	77			CLOCK1 — May be used as an input clock for timer 1.
CLOCK2*	114			CLOCK 2 — May be used as an input clock for timer 2.
CLOCK3*	60			CLOCK 3 — May be used as an input clock for timer 3.
CTS1 CTS2	25 53	OUTPUT OUTPUT	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)

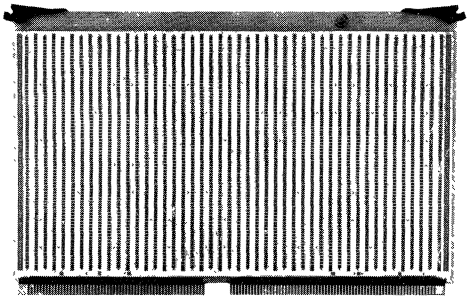
SIGNAL MNEMONIC	PIN NUMBER	TO TERMINAL	TO MODEM	SIGNAL NAME AND FUNCTION
TXC1 TXC2	37 65	OUTPUT OUTPUT	INPUT INPUT	TRANSMIT CLOCK — May be an output from baud rate generator or an input clock for transmitter or receiver.
TXD1 TXD2	19 47	INPUT INPUT	OUTPUT OUTPUT	TRANSMIT DATA — Used for transmit data 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

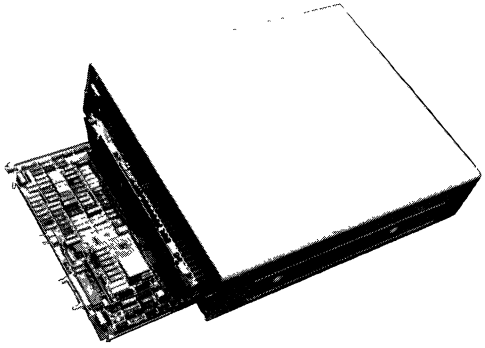
The VERSAbus Wirewrap Module permits the user to construct and incorporate his custom circuits into an EXORmacs Development System, VMC 68/2 Microcomputer System, or any other VERSAbus application Incorporated on the module are the power bus and ground bus printed wiring runs. On the component or front 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

M68KV020BES1
M68KV020BES2

Benchmark 20 System Package



The Benchmark 20 System Package is a 32-bit, high performance system that provides the first-time user of the MC68020 MPU a tool to quickly evaluate the MC68020 and to start code development The Benchmark 20 system has the software tools to allow the user to do benchmarking and code debugging

Paged memory management support is provided by the Memory Management Board (MMB) containing a gate array implementation of the table walking algorithm of the proposed MC68851 PMMU and a 512 word set associative address translation cache

The Benchmark 20 supports Motorola resident assemblers on the EXORmacs and VME/10 systems with upload and download capability The Benchmark 20, being VERSAmodule based, can be user configured with existing VERSAmodule boards to provide user target systems. The Benchmark 20 system comprises the following major sub-elements

- M68KVM04 VERSAmodule 32-bit Monoboard Microcomputer
  - Processor — MC68020
  - Coprocessor — Support for MC68881
  - Memory Management Unit — MMB (Gate Array subset of MC68851)
  - Memory Hierarchy — 4K Entry Cache, VERSAbus/RAMbus
  - Utilities — PTM, SIO, ROM
- M68KVM13-1 VERSAmodule 1024K Byte Dynamic RAM
  - Capacity — 1 Mbyte with 64K-bit Dynamic RAM devices
  - Dual Port — VERSAbus/RAMbus
  - Error Detection — Byte Parity
- MVMCH1-2 VERSAmodule 4-Slot Chassis with Card Cage, Power Supply, and Front Panel
- Software — 020bug
  - Assembler/Disassembler — Supports MC68020 and MC68881 Instructions
  - Benchmarking — Timer Utilities for Program Execution Timing

SYSTEM FEATURES

The Benchmark 20 has the following system features.

- Both boards conform to VERSAbus mechanical and electrical specifications.
- Provides 32-bit data and address computing capability.

SYSTEM SOFTWARE FEATURES

The VM04 incorporates an on-board ROM based debug monitor, 020bug, which allows the user to access the VM04 resources, such as

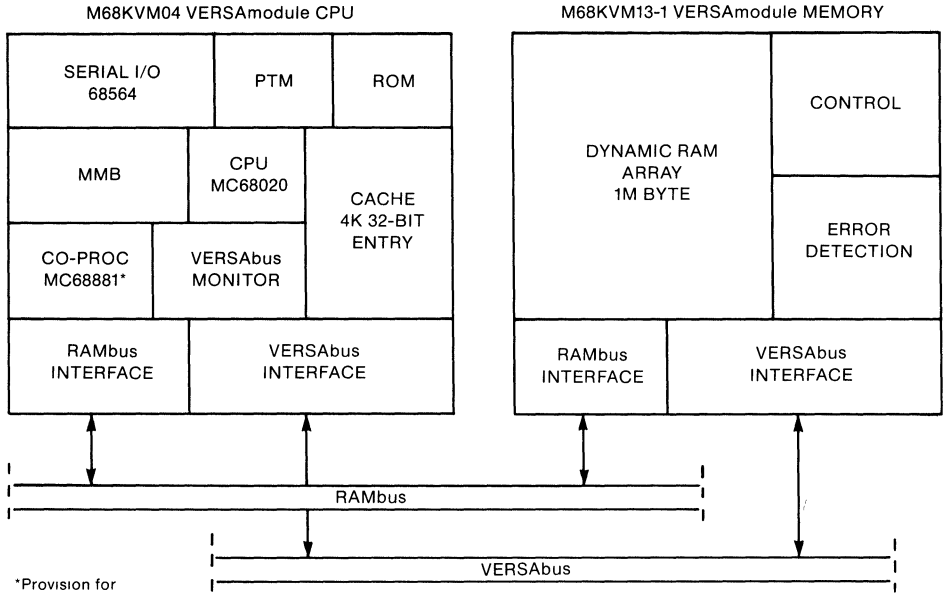
- Any MC68020/68881*/MMB* internal register.
- VM04 Status/Control bits
- Timer for benchmarking applications, etc
- Any memory mapped VERSAbus or RAMbus resource

In addition to these user utilities, 020bug also includes.

- Two types of power-up self test diagnostics, short and extensive The test type is selectable with the TEST1 and TEST0 DIP switches mounted on top of the VM04 board
- Software drivers to accommodate both serial ports, including download and upload capability from a host computer (using Motorola's S-records)
- Functional module drivers for the MC68881* Floating Point Coprocessor and the Memory Management Unit (MMB)
- An operating system bootstrap command
- Breakpoint and trace modes for program debug/development

2

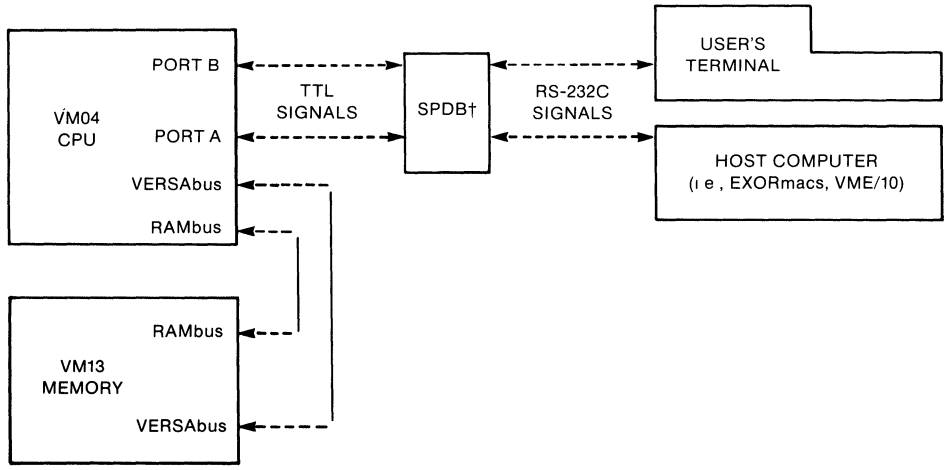
FIGURE 1 — Benchmark 20 Board Set



SYSTEM CONFIGURATIONS

The Benchmark 20 is factory tested in a double-board computer configuration

FIGURE 2 — VM04/VM13 Double-Board Computer Configuration



†SPDB denotes VM04 Serial Port Distribution Board

TABLE 1 — Benchmark 20 Specifications

Characteristics	Specifications
Power Requirements	TBD
Operating Temperature	0° to 70° C
Storage Temperature	-40 to 85° C
Relative Humidity	5% to 95% (non-condensing)

SYSTEM COMPONENT FEATURES

M68KVM04 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 The VM04 offers the following features

- 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 consisting of 32 data bits
- RAMbus Interface provides high-speed data path to/from memory.
- VERSAbus Interface allows user configuration of Microcomputer System to fit application
- VERSAbus Interrupter, Interrupt Handler, and Arbiter on-board.
- Programmable Timer Module
- Dual Multiprotocol Serial I/O Ports
- Two ROM sockets configured for Industry Standard 28-pin ROM/EPROM devices

M68KVM13 VERSAmodule 1024K Byte Dynamic RAM

The VM13 is a 1024K byte dynamic RAM module with parity It attains its capacity and performance using high density HMOS 64K × 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. The VM13 offers the following features.

- 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 — 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).

MVMCH1-2 Chassis

The VERSAmodule 4-Slot Chassis (MVMCH1-2) includes a card cage, one or more power supplies, and one or more power monitors Some of the basic features are:

- Four-slot VERSAmodule card cage in a front-load metal housing.
- Snap-on, removable front panel.
- Two fans for cooling cards and power supply
- Four-slot backplane (VERSAbus compatible).
- Two DB25 connectors installed on back panel.

020bug 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 VM04/VM13 Benchmark 20 system through the RS-232 port on the VM04.
- 32 debug, up/downline load and disk bootstrap load commands.
- Full speed execution of system and user developed programs operating out of the VM04/VM13 Benchmark 20 system package.
- Virtual terminal capability for up/downline load from an EXORmacs-VME/10 Development system.
- Powerful software and system debug 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.
- Disk Bootstrap load/dump from standard Motorola Floppy and Hard Disk Systems.
- Includes all required installation and operation documentation.

2

RAMbus FEATURES

- Provides a high speed dedicated channel for concurrent processing.
- Occupies 50 VERSAbus P2 connector pins
- 32-bit multiplexed address and data
- Supports all MC68020 data bus modes, including Dynamic bus sizing for 8-, 16-, and 32-bit data ports Operand misalignments
- Software transparent — RAMbus and VERSAbus can share same address space
- RAMbus cacheable handshake, which is configured on the slave.
- Circuitry required to interface to RAMbus — 12 to 25 devices

SYSTEM INTERFACE REQUIREMENTS

All VERSAmodules conform to VERSAbus electrical and mechanical specification, so their interface requires a VERSAbus backplane and card cage

VERSAbus INTERFACE

The VM04 uses the P1 and P2 VERSAbus edge connectors in the Expanded Bus Option, which allows a 32-bit data path and full 32-bit addressing range The VERSAbus Expanded Bus Option includes the upper 16-data bus signals, the upper 8 address bus bits, 50 I/O pins, 19 RESERVED pins, V_{CC}, GND, and other assorted P2 connector signals The VERSAbus P1 and P2 signal placement and functionality are defined by the Motorola VERSAbus Specification, document number M68KVBS/D4

Ordering Information

Part Number	Description
M68KV020BES1	Benchmark 20 High Performance Evaluation Computer System Package Includes User's Manual 115 Vac Power
M68KV020BES2	Benchmark 20 High Performance Evaluation Computer System Package Includes User's Manual 230 Vac Power
M68KV020BES/D1	Benchmark 20 User's Guide

020bug 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 Benchmark 20 System Package through the RS-232 port on the VM04
- 32 debug, up/downline load and disk bootstrap load commands
- Full speed execution of system and user developed programs operating out of the Benchmark 20 System Package
- Virtual terminal capability for up/downline load from an EXORmacs/VME/10 Development system or from a cross-computer
- Powerful software and system debug 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
- One line Assembler/Disassembler
- Disk Bootstrap load/dump from standard Motorola Floppy and Hard Disk Systems
- Includes all required installation and operation documentation

020bug Source and Relocatable Object Module Package

- Source and Relocatable Object Modules for 020bug on diskette or cartridge disk
- Relocatable Object Modules allow the user 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 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. 020bug Source and Relocatable Object Modules are available as a separate product on either EXORmacs Development system compatible diskette, or disk cartridge.

020bug provides a powerful evaluation, use and system debugging tool for VERSAmodule Systems. The EPROM Resident package (M68K2RBBUG4) will operate in a minimum of 16K bytes of ROM space. 020bug uses the first 1K byte of RAM storage for Interrupt Vectors and the next 8K bytes for temporary storage. The EPROM resident package is delivered in two 8K byte EPROMs. Table 1 lists the commands available to the user.

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 020bug allows terminal and host access to the two serial ports on the VMM. 020bug 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/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 VMM.

FIGURE 1 — Benchmark 20 with 020bug Connected to a VME/10 Serving Both the Control and Downline Load Functions.

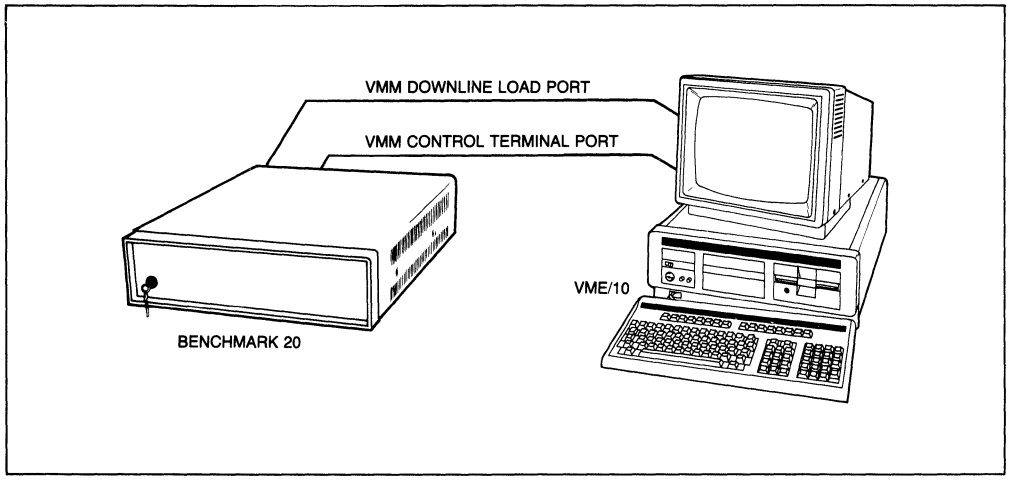
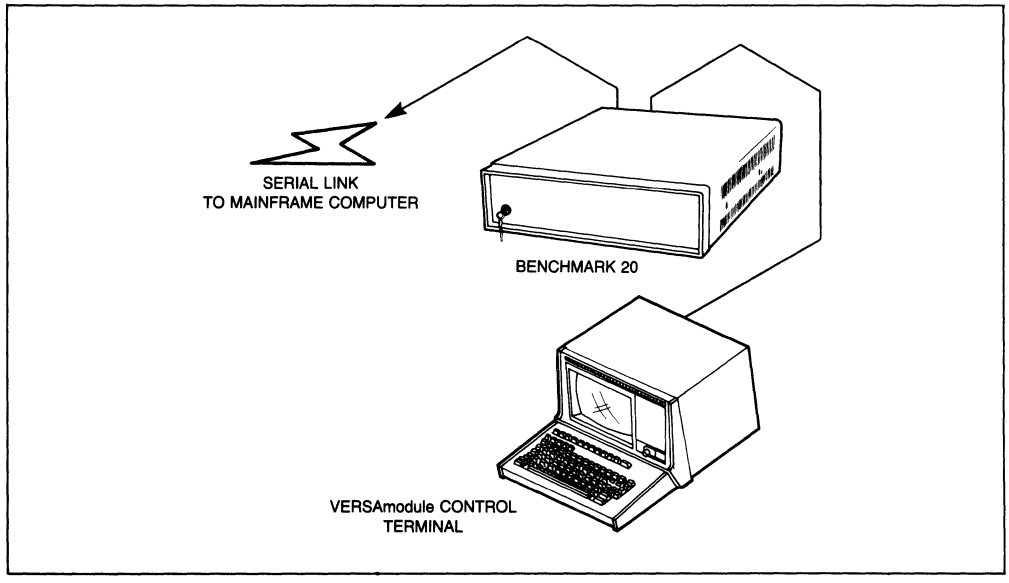


FIGURE 2 — Benchmark 20 with 020bug and Cross-Computer Serial Communications Link (EXORmacs, or Other Computer with "S" Record Files)



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 VME/10 Development System. Following a download, 020bug 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 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 on the serial port debug terminal to 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.

020bug 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 (M68K2CSBUG4, M68K2FSBUG4 and M68K2XSBUG4) provides the user with the information necessary to link 020bug 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 and/or the 020bug disk bootstrap loader. The user may also use the Relocatable Object Modules substituting their own device bootstrap loader for the 020bug disk bootstrap loader.

Source Modules permit the user to modify, or customize any of the 020bug functions as desired.

The dual port serial I/O cable assembly provided with the EPROM Resident Package is available as a separate product (MVMCH3-104).

TABLE 1 — Terminal Control Functions

Input Function	
CTRL X (Cancel Line)	Cursor is backspaced to the beginning of the line. If the terminal port is configured with the hard copy option, a <CR> <LF> is issued along with another prompt. See PF command.
CTRL H (Backspace)	Cursor is moved back one position. Character above cursor is erased. If the hard copy option is selected, a "/" character is typed along with the deleted character. See PF command.
CTRL D (Redisplay)	Line is redisplayed in the following line.
CTRL W (Wait)	Console output is halted. Any key resumes output.

Prompt

At power up 020bug will go into the auto baud detect mode, waiting for the user to type <CR> until the baud rate is detected. Once the baud rate is detected the following message will be printed:

```
VM04 020bug Diagnostic Package Version V.r
020bug>
```

Also after reset the above message is printed. At any other time the prompt will be:

```
020bug>
```

2

Command Line Format

A command consists of two letters, followed by any number of spaces and then any required parameters. Commands can be entered in upper or lower case.

The commands are shown using a modified Backus-Naur form syntax. The metasymbols used are:

< > The angular brackets enclose a symbol, known as a syntactic variable, that is replaced in a command line by one of a class of symbols it represents.

| This symbol indicates that a choice is to be made. One of several symbols, separated by this symbol, should be selected.

/ This symbol indicates that one or more of the symbols separated by this symbol can be selected.

[ ] Square brackets enclose a symbol that is optional. The enclosed symbol may occur zero or one time.

{ } Curly brackets enclose an optional symbol that may appear zero or more times.

Syntactic Variables

<ADDR> — Address
<ADDRL> — Lower of the two addresses specified
<ADDRH> — Higher of the two addresses specified
<DEL> — Delimiter (a space or comma)
<COUNT> — A number
<FC> — Function codes
<CMD> — A 20bug command name

Address As A Parameter

Many commands use <ADDR><address> as a parameter. The syntax accepted by 020bug is similar to the one accepted by the assembler. A memory indirect mode, and an address + offset register mode is also allowed.

Format	Description
N	Absolute address + contents of automatic offset register
N+Rn	Absolute address + contents of offset register n
(An)	Address register indirect
(An,Xn)	Address register indirect with index
d(An)	Address register indirect with displacement
d(An,Xn)	Address register indirect with displacement + index
[d(An,Xn)]	Address register indirect + displacement with preindex indirect
[d(An)]+Xn	Address register indirect + displacement with postindex indirect
[N]	Memory indirect

X — A or D register
N — a value from 0 to FFFFFFFF
d — a value from 0 to FFFFFFFF
n — a value from 0 to 7
R — offset register

Commands Available

Command Mnemonic Options	Description
MD[n] <ADDR>[:<COUNT>][; [B:W:L:S:D:X:P:DI]]	Memory Display
MM <ADDR>[<DEL><DATA>] [;[[N][B:W:L:O:V:S:D:X:P]]:DI]	Memory Modify

These commands are used to examine and change memory locations. MD and MM accept the following data types:

B — Byte
W — Word
L — Long Word
S — Single Precision
D — Double Precision
X — Extended Precision
P — Packed BCD

The DI option enables the assembler/disassembler. Data type option is ignored if DI is selected. With MM the next and previous locations are opened when <CR> and <LF> are entered respectively. The count option specified with the MD command specifies the number of lines that will be displayed, defaulting to 1 if none are entered. The optional n character

selects the port used for sending the data. The W- and X-off are recognized at port n and will halt the output if received. The N option of the MM command disables the read portion of the command. O,V force odd or even access only. The L option selected with O or V does accesses every 4 bytes (longword boundaries). If the data argument is entered in the MM command, the location will be set to the specified value. The options are ignored Data delimited by ' or " will be taken as an ASCII string.

Command Mnemonic Options	Description
RD <DEL> [A:D:S:F:P[T:B]]	Register Display
RM <REG> [<DATA>][, [B.W.L.S.D:X:P]]	Register Modify

These are the register examine/modify functions. These functions allow you to examine and modify all the CPU registers, including the coprocessor registers that are part of the programmer's model. If the <DATA> argument of the RM command is entered, the options are ignored. Data delimited by ' or " will be taken as an ASCII string. If the data argument does not fit into the specified register, characters will be deleted at the beginning until it fits.

#regs. type

8	A	— Address Registers	(A0–A7)
8	D	— Data Registers	(D0–D7)
10	S	— System Registers	(PC,SR,MSP,ISP,USP,VBR,SFC,SFC,DFC,CACR,CAAR)
11	F	— Floating Point Registers	(FP0–FP7,FCR,FSR,FAR)
36	P	— PMMU Registers	10 Translation Registers: (CRP,SRP,DRP,TC,CS,ST,CAL,VAL,SCC,ACTL)
			26 Breakpoint Registers: (BAD0–BAD7,BAC0–BAC7,LBA0–LBA7,LBC0–LBC3,PBA,PBC)

Total: 72 Registers. (Note that A7 is the active stack pointer)

Command Mnemonic Options	Description
DF	Display Format

DF allows the user to format the display of registers when a breakpoint or trace exception is taken.

Command Mnemonic Options	Description
BF <ADDRL><DEL><ADDRH>[; [B:W:L:S:DD:X:P]]	Block Fill
BM <ADDRL><DEL><ADDRH><DEL><ADDRL1>	Block Move
BS <ADDRL><DEL><ADDRH><DEL><DATA> [; [B:W:L:S:DD:X:P]	Block Search
BI <ADDRL><DEL><ADDRH>	Block Initialize

BF fills a memory area with the specified data. Data types are the same as explained in the MD/MM functions.

BS searches memory for the specified string.

BI initializes parity nondestructively.

2

Command Mnemonic Options	Description
TM [EXIT]	Transparent Mode

TM essentially connects the two on-board serial ports together, allowing the user to communicate with a host computer. The exit code specifies the code that will allow the user to exit the transparent mode. The ports do not have to be at the same baud rate, but the terminal port baud rate should be equal to or greater than the host port baud rate for reliable operation.

Default exit code = A

Command Mnemonic Options	Description
BR[I:L:P] <ADDR>[:,COUNT>]	Breakpoint Insert
BD[I:L:P] <ADDR>	Breakpoint Delete

The breakpoint functions allow the user to interrupt the normal program flow at selected points in his program for debugging purposes. Every time that a breakpoint is found a breakpoint handler routine prints the CPU state on the screen.

Three different breakpoint types are allowed:

Command Mnemonic Options		Description
BR[I] ,ADDR>[:<COUNT>	8	Breakpoint Instruction
BRL <ADDR>[:COUNT>[;S/U/D/P/W]]	4	Logical Breakpoint
BRP <ADDR>[:COUNT>[;S/U/D/P/W]]	1	Physical Breakpoint

The count field specifies how many times the specified location must be accessed before a breakpoint is taken. Logical and physical breakpoints will allow breakpoints on an access to a location, as opposed to BR, which allows breakpoints on instruction fetches only.

The logical and physical breakpoints provide the user with more resolution in specifying an address, with the following options:

S — Supervisor references
U — User references
P — Program references
D — Data references
W — Write or read modify write references

If none of the above options are specified, the conditions will be ignored.

Command Mnemonic Options	Description
TR[C]	Trace

The TRACE function allows execution of one instruction at a time, or on change of control flow (TRC).

Command Mnemonic Options	Description
GO[I:L:P] [<ADDR>]	Go Execute Program
GT[I:L:P] <ADDR>	Go and Set Temporary Breakpoint
GD [<ADDR>]	Go Direct to Program

The GO function initiates execution of user programs. All previously set breakpoints are enabled. The options I (instruction), L (logical), and P (physical), allow the user to enable only a class of breakpoints.

GD disables all breakpoints before starting program execution.

Command Mnemonic Options	Description
DU[n] <ADDRL><DEL><ADDRH><DEL>[<TEXT>]	Dump S Records to Port n
LO[n] [<ADDRL>][;[N:X]=<TEXT>]	Load S Records from Port n
VE[n] [<ADDRL>][;[N:X]=<TEXT>]	Verify S Records

The DU[n] command recognizes the AX-off (or AW) received from port n to stop its output. Any key resumes output.

For the LO command: N = ignore checksum
X = echo data read into terminal

The LO and VE commands have an optional address argument. If specified, the first record is loaded at that address. The difference between this address and the one included in the first record is added to the remaining records.

Command Mnemonic Options	Description
ML <STRING>	Macro Load Command

ML allows the user to load a file of 20bug commands from port 2. The specified string is transmitted to port 2.

Command Mnemonic Options	Description
BH BO	Bootstrap and Halt Bootstrap Operating System
(IOP/IOC/IOT)	Disk Transfer (Sectors)
CD	Configuration Display

CD shows the board configuration, including switch settings, available memory.

Command Mnemonic Options	Description
DC <expression>	Evaluate Expression

This function will allow evaluation of expressions using integer and real expressions in decimal and/or binary.

Command Mnemonic Options	Description
OF Rn;A	Display/Change Offset Reg. n

OF allows the user to access and change pseudo-registers called offset registers. There are eight offset registers (R0–R7), and they are used to aid in address calculations when debugging programs. The A option sets the specified register as an automatic register. This means that its value will be automatically added to the address argument of every command. OF <CR> displays all the registers.

Offset register rules:

1. At power up/reset R7 is the automatic register.
2. R7 is always zero. Used to bypass the automatic register. See (5).
3. Any register can be set as the automatic register.
4. Automatic register is always added to every address argument of every 20bug command, except the OF Rn command (no interaction between offsets).
5. Automatic register is not used when an offset register is explicitly added to an argument.
6. There is always an automatic register. Note that a convenient way to disable an automatic register is by setting R7 as the automatic register.

Command Mnemonic Options	Description
PR[n] PD[n]	Printer Attach to Port n Printer Detach from Port n

PA and PD allow the user to logically connect to a printer, to get hard copy outputs.

It is assumed that communications are through the I/O channel, as there are no on-board printer ports.

Command Mnemonic Options	Description
PF[n]	Port Format

PF allows the user to configure the two on-board serial ports. For port 1, the user can specify if a hard or soft copy terminal is being used. Also a null padding count can be specified for use after a CR and/or LF.

Defaults: soft copy term and 0 null padding
For port 2 the user can specify parity, #data and stop bits, and baud rate.

Command Mnemonic Options	Description
MA <ADDR><DEL><FC>	Map Address

MA returns the physical address that is mapped to the specified logical address. The logical address and the function codes must be specified.

Command Mnemonic Options	Description
AS <ADDRL><DEL><ADDRH><DEL><FC>	Allocate Logical Space

This command takes two logical addresses that specify the beginning and ending addresses of a logical address space and generates the proper descriptors to the MMU to map the logical space in physical memory. Up to 10 logical spaces can be allocated.

Command Mnemonic Options	Description
HE [<COMMAND>]	Help

HE shows all the available commands along with a brief description of each one. If a command is specified only the description for that command will be displayed.

Trap Utilities

Various functions are available to a user program through the TRAP #15 instruction. The command is specified in the word following the TRAP opcode.

Example: to invoke INCH

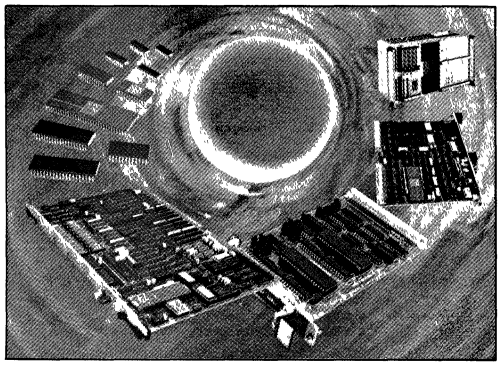
```
TRAP #15
DC.W INCH
```

INCH	Input Character from Port n
INCH1	Poll Input Port n Once
OUTCH	Output Character to Port n
PDATA	Print ASCII String to Port n
DISK	Access Disk Routines
STIM	Start Timer
RTIM	Read Timer
RPTR	Return Pointer to Command
HEXADEC	Convert Hex Number to ASCII Decimal
HEXA	Convert Hex Number to ASCII Hex
ADECHEX	Convert ASCII Decimal to Hex
AHEX	Convert ASCII Hex to Hex
COLD	Cold Entry to 20bug
WARM	Warm Entry to 20bug
ALLOC	Performs the Equivalent of the AS Function
GTBL	Takes as Input a Parameterized Table and Generates the Proper MMU Tables
MAP	Performs the Equivalent of the MA Function

Ordering Information

Part Number	Description
M68K2RBBUG4	020bug, the VERSAmodule Benchmark 20 System Debug Package, includes EPROM set* and Dual RS-232C Serial I/O Cable Assembly 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	Dual RS-232C Serial I/O Cable Assembly for interconnecting between a 020bug 50 Pin I/O arrangement with pin out compatible to the VERSAmodule Benchmark 20 System Package and external RS-232C terminals. May be used to connect a debug terminal and an up/downline load host to a VMM.
M68K2RBBUG4/D1	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.

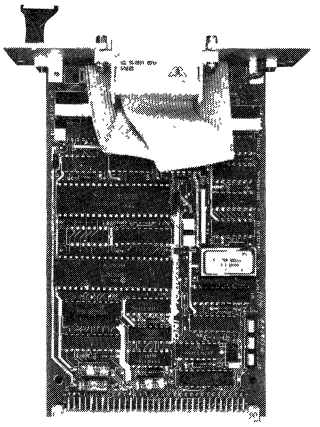
I/Omodule Data Sheets

3

MVME400

3

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

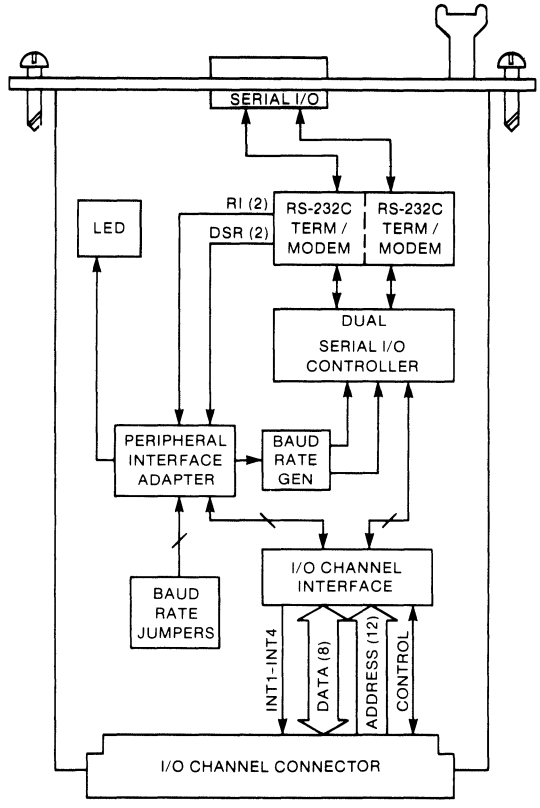
The Motorola I/O Channel is specifically designed to provide efficient, low-cost distributed communications to peripheral and I/O controller boards. It provides a 12-bit address bus, plus an 8-bit bidirectional data bus, and supports asynchronous operation at data rates up to 2 mbytes/sec. For those 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

3

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

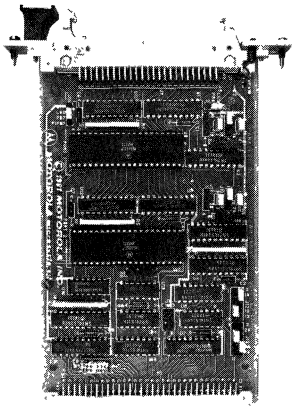
Other Modules in the MVME400 Series

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

3

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

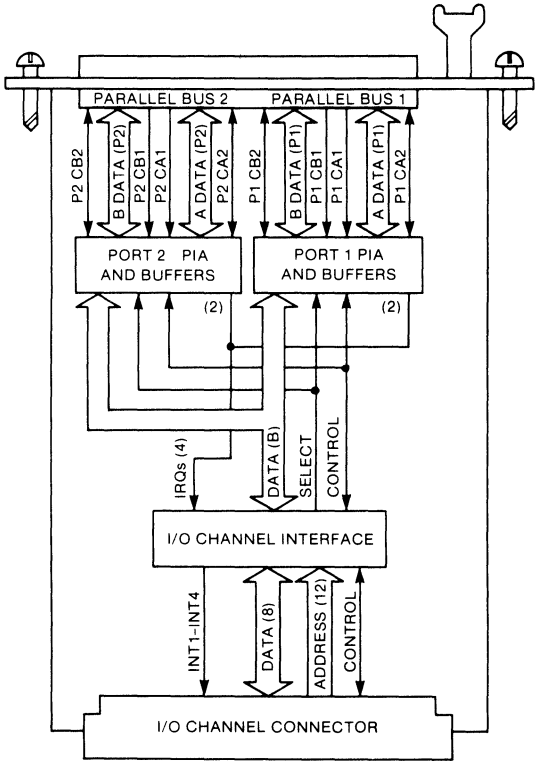
MVME400 Series

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

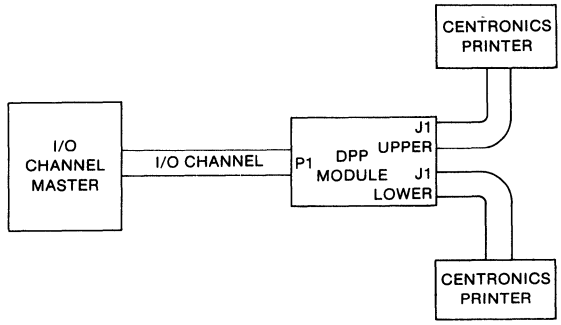


TABLE 1 — Connector J1 Pin Assignments and Signal Descriptions

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

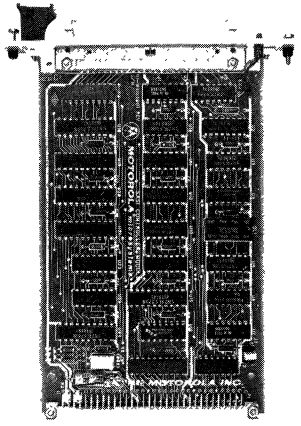
M68RIOCS/D2	Input/Output Channel Specification Manual
MC6821	Peripheral Interface Adapter Data Sheet

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

3

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

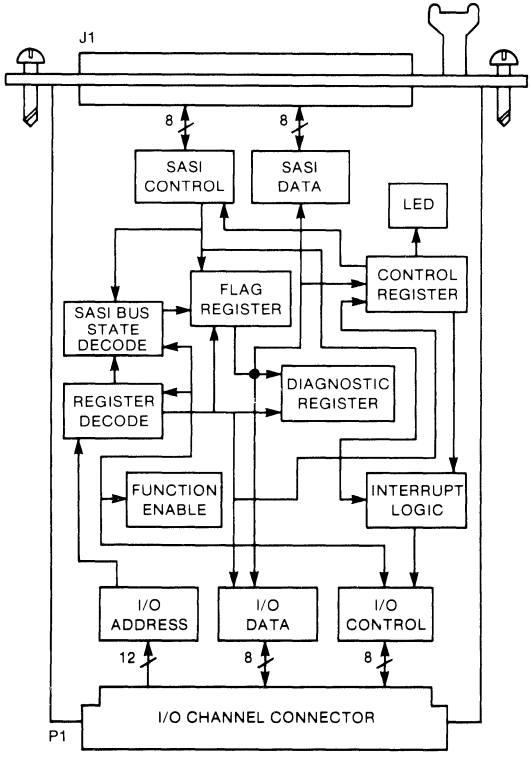
MVME400 Series

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

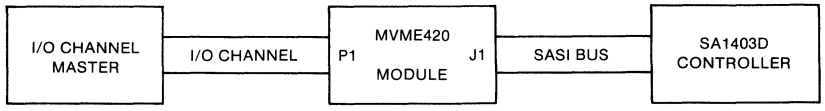
MVME420 — SASI Peripheral Adapter Block Diagram



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

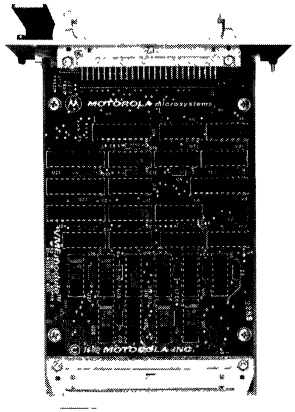
M68RIOCS/D2	Input/Output Channel Specification Manual
SA1403D	Shugart Associates Technical Manual for the SA1403D Controller.

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

MVME435

Buffered 9-Track Magnetic Tape Adapter I/Omodule



- Motorola I/O Channel Interface Compatible
- Interfaces I/O Channel to Industry Standard 9-Track, Dual Density (800/1600 bpi), 1/2" Magnetic Tape Formatters
- Supports Tape Drivers Operating at from 12.5 to 125 ips in the Start/Stop Mode
- Interfaces I/O Channel to One or Two (Daisy Chained) Formatters with Integral Master Drive Each Formatter Can Control Three Additional Slave Drives Providing a Maximum of Eight Drives Per Adapter
- Provides Software Selectable Direct (Interrupt Driven) or Buffered Data Transfer Modes
- 4K Byte FIFO Buffer
- In the Buffered Mode, Allows Direct Data Transfers Between Drives (Non-I/O Channel) of 4K Bytes Per Operation
- Supports On-The-Fly Operation
- Accessible to Host Via 32 Contiguous Read/Write Registers
- Provides 16-Bit Status Register
- Packaged on Two (Sandwiched) Cable Interconnected Single Eurocards (6 3" × 3 9") Requires the Space of Two Card Cage Slots
- Connects to Host I/O Channel Using a Standard 64-Pin DIN 41612 Connector
- Connects to Formatter Via Two Adapter Module Front Panel Mounted 50-Pin Connectors
- Four Front Panel Yellow LEDs Indicate Read/Write Status for I/O Channel and Tape Drive Operations
- Front Panel Red FAIL LED Accessible to Host for Adapter Module Failure or Other Indication
- 0° C-70° C Operating Temperature Range

The Buffered 9-Track Magnetic Tape Adapter (MTA) is used to add magnetic tape data storage capacity to microcomputer system resources on the Motorola I/O Channel It provides the interface for one or two dual-density, 1/2-inch magnetic tape formatters allowing the addition of up to eight compatible tape drives operating at speeds of from 12 5 to 125 inches per second in a start/stop mode.

The MTA is implemented in the single Eurocard form factor and is compatible with industry standard tape formatters such as those offered by Pertec and Kennedy.

The adapter module provides for host program control of the transfer of formatter and adapter commands, status and buffered data. Formatter commands include:

- Formatter DISABLE/RESET
- Select Formatter/Drive/Density
- OFF-LINE
- REWIND
- WRITE and READ FORWARD/REVERSE
- WRITE FILEMARK
- SEARCH FILEMARK FORWARD/REVERSE
- SPACE FORWARD/REVERSE
- ERASE TAPE

Adapter commands include

- Adapter Reset
- Data Transfer mode select (Buffered Block or Direct)
- Last Word Output
- Data Input
- Data Output
- Input Status 1 byte
- Input Status 2 byte

Reads and writes by the host via the I/O Channel to the appropriate register address are used for transferring data and status from/to the MTA buffers.

Typical adapter use can be described as follows First, the desired formatter and drive are selected and the direct or buffered block adapter data transfer mode is selected by host command. Then the desired formatter tape motion command is sent to the adapter. On non-data transfer commands the adapter interrupts the host when the commanded motion is complete. The host may read the adapter status byte registers to determine the cause of completion. On formatter data transfer commands (WRITE or READ FORWARD/REVERSE) the adapter interrupts the host whenever it has need to send/receive data to/from the host or when an error has been detected. The host may read the adapter status registers to determine the cause of the interrupt.

Both data transfer modes pass data through the adapter's 4K byte FIFO buffer. Only the method of host transfer differs. The buffered block mode is intended for I/O Channel host processors using programmed I/O data transfer (although a block DMA may also be used). The direct mode is intended for hosts using DMA data transfers (or capable of very fast programmed I/O).

In the buffered block mode, the host transfers an entire tape block to/from the 4K byte adapter buffer before the command to transfer to/from the next tape block is given This makes host programmed I/O transfer totally asynchronous so that the host processor may handle higher priority tasks concurrently with magnetic tape data transfers. MTA last word output and formatter WRITE commands following the last byte of the tape block cause the tape WRITE to begin. The host is interrupted by the adapter when the tape block transfer to/from the tape drive is completed. Tape block size is limited to a maximum of 4K bytes in this mode.

In the direct mode, the host is interrupted by the adapter during READ/WRITE whenever the 4K byte buffer is half full/empty or a tape block transfer is complete. Thus, the I/O Channel host processor may transfer up to 2K bytes of data in a DMA or fast programmed I/O transfer using tape data blocks that exceed 4K bytes. Tape block size, of course, may be any length less than 4K bytes as in the buffered block mode. The direct mode results in higher tape transfer performance, since physical request for start of tape transfer need not wait for completion of block transfer between the host and the adapter buffer.

A functional block diagram of the adapter module is illustrated in Figure 1 The maximum tape system configuration is illustrated in Figure 2.

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

MVME110-1	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 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 system bus, 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 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 16-bit modular product families: VMEmodule and VERSAmodule.

FIGURE 1 — Buffered 9-Track Magnetic Tape Adapter Module Functional Block Diagram

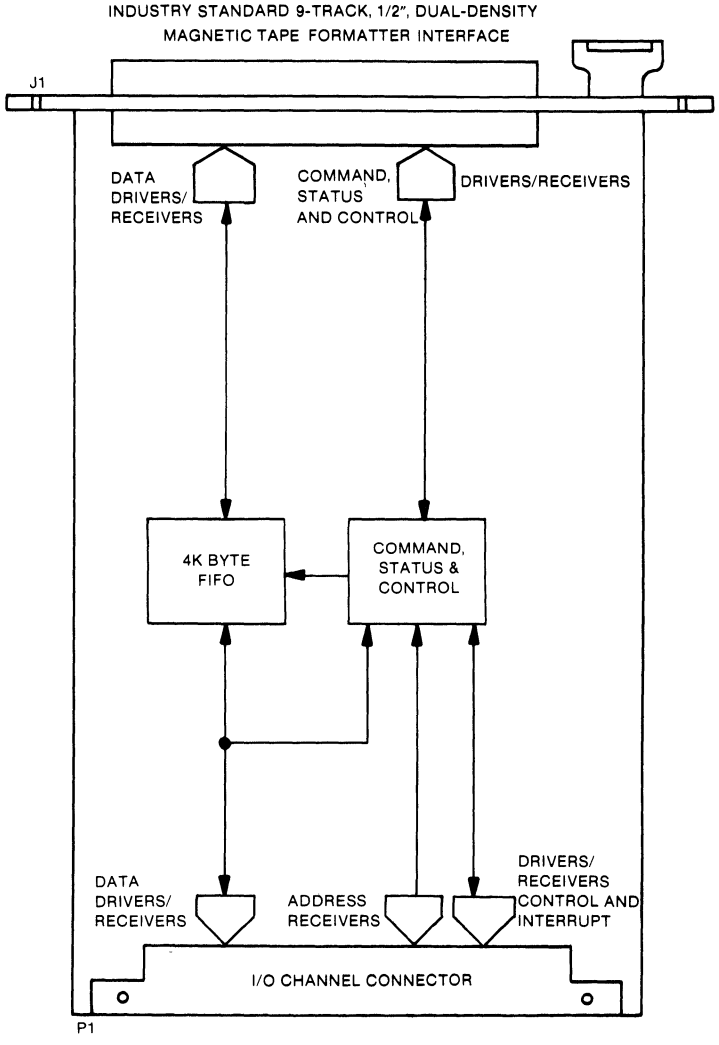
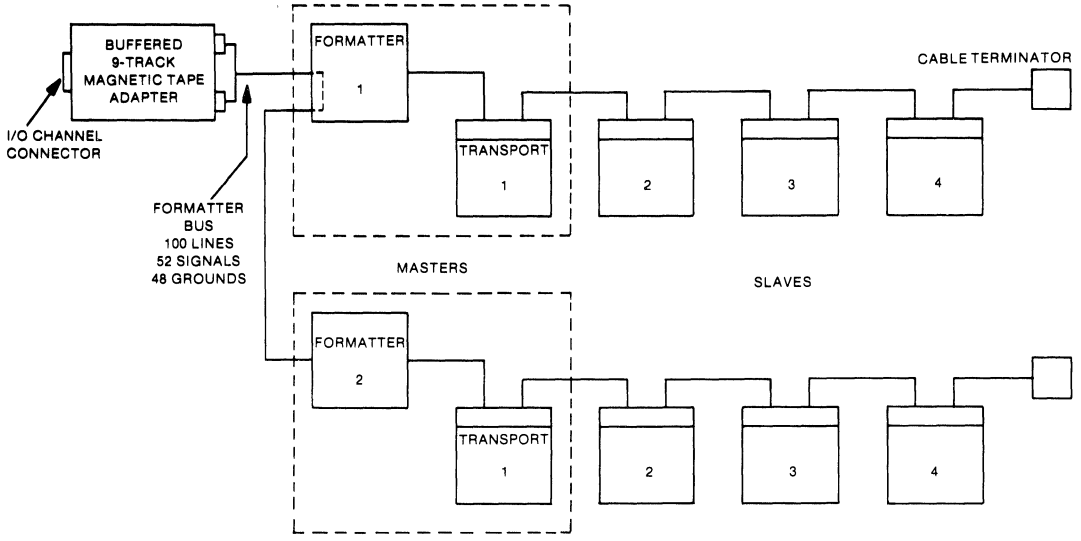


FIGURE 2 — Maximum Magnetic Tape System Configuration



Peripheral Connector J1 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonics	Signal Name And Description
1, 40	(Reserved)	No connection
2	FRRD0*	FORMATTER READ DATA (bit 0) — Input read data bit 0 from formatter to adapter
3	FRRD1*	FORMATTER READ DATA (bit 1) — Same as FRRD0*, but bit 1
4	LDPNT*	LOAD POINT — Input signal that is true when transport is ready and tape is at rest with BOT tab under the photosensor.
5-49 (odd numbers)	GND	Connect to ground plane
6	FRRD4*	FORMATTER READ DATA (bit 4) — Same as FRRD0*, but bit 4
8	FRRD7*	FORMATTER READ DATA (bit 7) — Same as FRRD0*, but bit 7
10	FRRD6*	FORMATTER READ DATA (bit 6) — Same as FRRD0*, but bit 6
12	HER*	HARD ERROR — Input signal which is set low if a read error is detected by formatter
14	FMK*	FILE MARK — Input pulse which indicates formatter read logic has detected a file mark

Peripheral Connector J1 Pin Assignments and Signal Descriptions (continued)

Pin Number	Signal Mnemonics	Signal Name And Description
16	CCG/ID*	CHECK CHARACTER GATE/IDENTIFICATION — In NRZI mode, input signal CCG is set low by formatter when read information being transmitted to adapter is CRCC or LRCC Signal is high when data characters are being transmitted In PE format, input signal ID is pulsed when ID burst is read from tape
18	FRMEN*	FORMATTER ENABLE — Output signal that when high resets formatter(s) to quiescent state
20	FRRD5*	FORMATTER READ DATA (bit 5) — Same as FRRD0*, but bit 5
22	EOT*	END OF TAPE — Input signal that when true indicates the EOT reflective tab is positioned under the photosensor.
24	OFLNE*	OFF-LINE — Output signal pulse that causes selected transport to be placed in off-line mode of operation Signal does not cause formatter to become busy
26	NRZI*	TRANSPORT FORMAT — Input signal that, when true, indicates the selected transport is utilizing NRZI format When signal is false, transport is utilizing PE format
28	RDY*	READY — Input signal that is true only when transport is ready to receive external commands (interlocks made, load or rewind complete, transport on-line and not rewinding)
30	RWNDG*	REWINDING — Input signal that is true when transport is performing any rewind operation
32	FLPRT*	FILE PROTECT — Input signal that is true when power is on, tape is loaded, under tension, and supply reel has write enable ring removed.
34	FRDST*	FORMATTER READ STROBE — Input signal pulsed for each character of read information to be transmitted to the adapter
36	FWTDS*	FORMATTER WRITE STROBE — Input signal pulsed for each data character to be written on tape The strobe copies write data from adapter into formatter
38	DBSY*	DATA BUSY — Input signal that goes low when transport has reached operating speed, traversed the IRG, and formatter is about to read or write data from/to tape Signal remains low until data transfer is finished and post-record delay is completed Signal goes high when capstan starts to declerate tape
42	CER*	CORRECTED ERROR — Input signal used only in PE Mode Signal is set low by single track dropout during read or read-after-write.
44	ONLINE*	ON-LINE — Input signal that is true when the transport is under remote control When false, the transport is under local control
46	TAD1*	TRANSPORT ADDRESS (bit 1) — Output signal used with TAD0* to select one of four transports that may be daisy chained to the formatter.
48	FAD*	FORMATTER ADDRESS — Output signal that selects one of two formatters attached to the MTA Signal high selects address 0, signal low selects address 1
50	DENSE*	DENSITY SELECT — Output signal that when high selects 1600 cpi for PE format, when low, selects 800 cpi for NRZI format

Peripheral Connector J2 Pin Assignments and Signal Descriptions

Pin Number	Signal Mnemonics	Signal Name And Description
1-49 (odd numbers)	GDN	GROUND.
2	FBSY*	FORMATTER BUSY — Input signal that goes low on trailing edge of G0* command and remains low until tape motion ceases after execution of command to formatter.
4	FLWRD*	FORMATTER LAST WORD — Output signal pulse that during write or erase (variable length) indicates that next character to be strobed into formatter is last character of the record.
6	FWRD4*	FORMATTER WRITE DATA (bit 4) — Output write data bit 4 from adapter to formatter
8	GO*	INITIATE COMMAND — Output signal pulse low initiates formatter group commands On trailing edge of GO*, the REFWD*, WRTRD*, WRTFLM*, EDIT*, and ERASE* lines are copied into the formatter
10	FWRD0*	FORMATTER WRITE DATA (bit 0) — Same as FWRD4*, but bit 0
12	FWRD1*	FORMATTER WRITE DATA (bit 1) — Same as FWRD4*, but bit 1
14,16,22 36,44	(Reserved)	Not used by MTA
18	REFWD*	REVERSE/FORWARD — Output signal that when low specifies reverse tape motion, when high, specifies forward motion
20	REWND*	REWIND — Output signal pulse that causes transport to rewind to load point if transport is ready and on-line Signal does not cause formatter to become busy
24	FWRD7*	FORMATTER WRITE DATA (bit 7) — Same as FWRD4*, but bit 7
26	FWRD3*	FORMATTER WRITE DATA (bit 3) — Same as FWRD4*, but bit 3
28	FWRD6*	FORMATTER WRITE DATA (bit 6) — Same as FWRD4*, but bit 6

Peripheral Connector J2 Pin Assignments and Signal Descriptions (continued)

Pin Number	Signal Mnemonics	Signal Name And Description
30	FWRD2*	FORMATTER WRITE DATA (bit 2) — Same as FWRD4*, but bit 2.
32	FWRD5*	FORMATTER WRITE DATA (bit 5) — Same as FWRD4*, but bit 5.
34	WRTRD*	WRITE/READ — Ouptut signal that when low specifies write mode operation, when high, specifies read mode operation.
38	EDIT*	EDIT — Output signal that when low either (a) during read reverse, modifies stop delay to optimize head positioning when editing tapes, or (b) during write, turns off transport write current gradually at end of record to prevent erasing adjacent data record.
40	ERASE*	ERASE — Output signal that when low erases some tape by conditioning formatter to perform dummy write command or dummy write file mark operation Signal is also used to inhibit FRDST* during space forward/ reverse or file mark search operation.
42	WTFLM*	WRITE FILE MARK — Output signal pulse that when low causes a file mark to be written on tape if WRTRD* is also low.
46	TAD0*	TRANSPORT ADDRESS, bit 0 — Output signal used with TAD1* to select one of four transports that may be daisy chained to the formatter. TAD0* TAD1* TRANSPORT high high first high low second low high third low low fourth
48	FRRD2*	FORMATTER READ DATA (bit 2) — Same as FRRD0*, but bit 2.
50	FRRD3*	FORMATTER READ DATA (bit 3) — Same as FRRD0*, but bit 3.

Mechanical and Environmental Specifications

Characteristic	Specification
Temperature Operating Storage	 0° to 70° C -55° to +85° C
Relative Humidity	0% to 95% (non-condensing)
Dimensions Length × Width	Single Eurocard 160 mm × 100 mm (6.3 in. × 3 9 in.)
MTA Connector P1 J1, J2	Mating Part Triple row, 64-pin connector, female, DIN 41612, on I/O Channel backplane Double row, 50-pin connector, female, 3M 3425-5000 or equivalent, on ribbon cable to formatter

Ordering Information

Part Number	Description
MVME435	The Buffered Magnetic Tape Adapter I/Omodule includes an I/O Channel interface and an interface to up to two daisy chained industry standard 9-Track, 1/2" dual-density magnetic tape formatters capable of handling up to four tape drives, each The Adapter includes a 4K byte FIFO buffer Cables to the formatter may be obtained from the formatter supplier Includes User's Manual.
MVME435/D1	Buffered 9-Track Magnetic Tape Adapter Module User's Manual

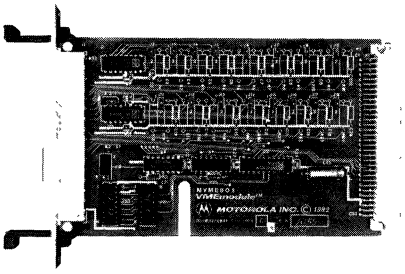
Related Documentation

M68RIOCS/D2	Input/Output Channel Specification Manual
79609	OEM User's Manual for Pertec Microformatted Tape Transports

Other Modules in the MVME400 Series

MVME400	Dual Channel RS-232C Serial Port
MVME410	Dual Channel 16-Bit Parallel Port
MVME420	SASI Peripheral Adapter

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.

3

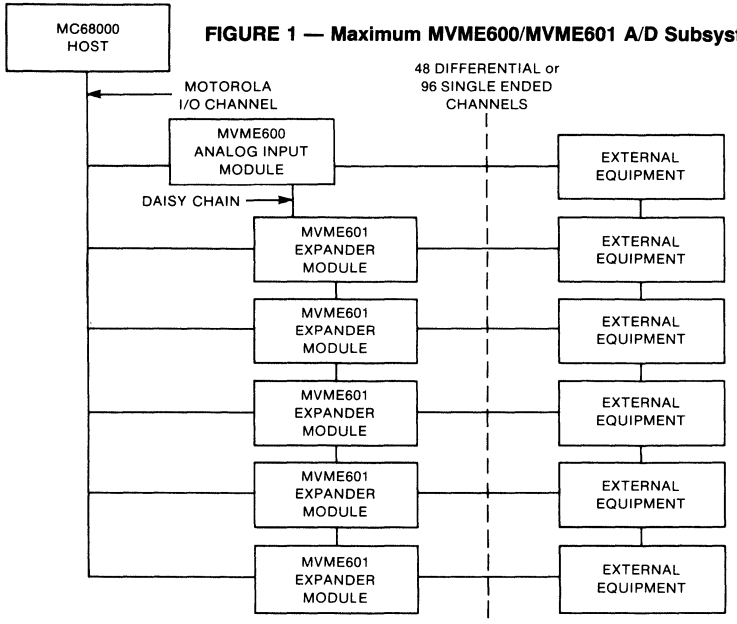


FIGURE 1 — Maximum MVME600/MVME601 A/D Subsystem

FIGURE 2 — MVME600 Functional Block Diagram

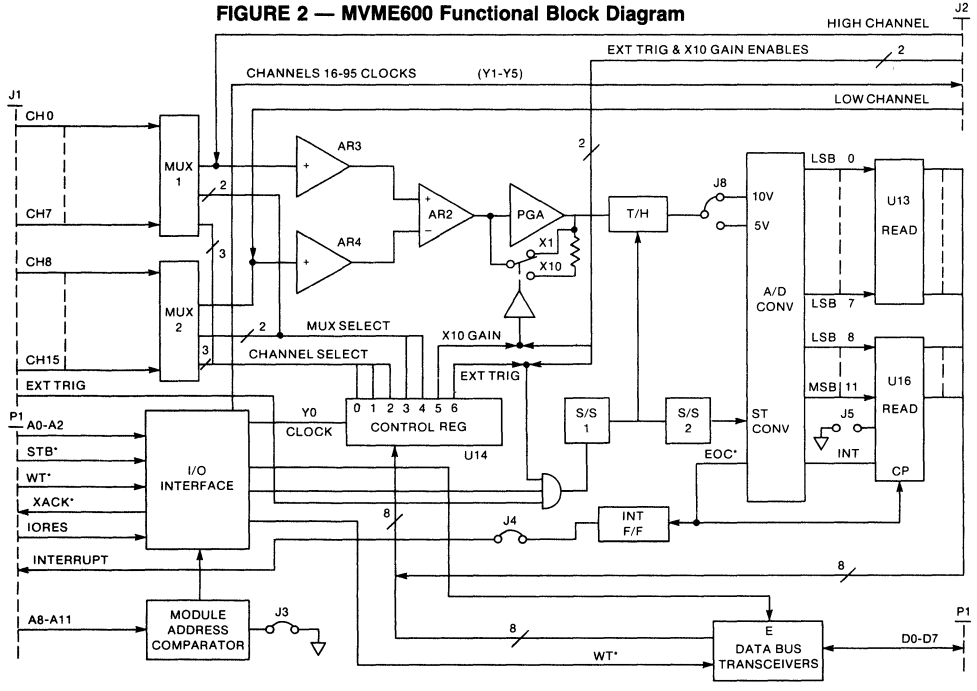
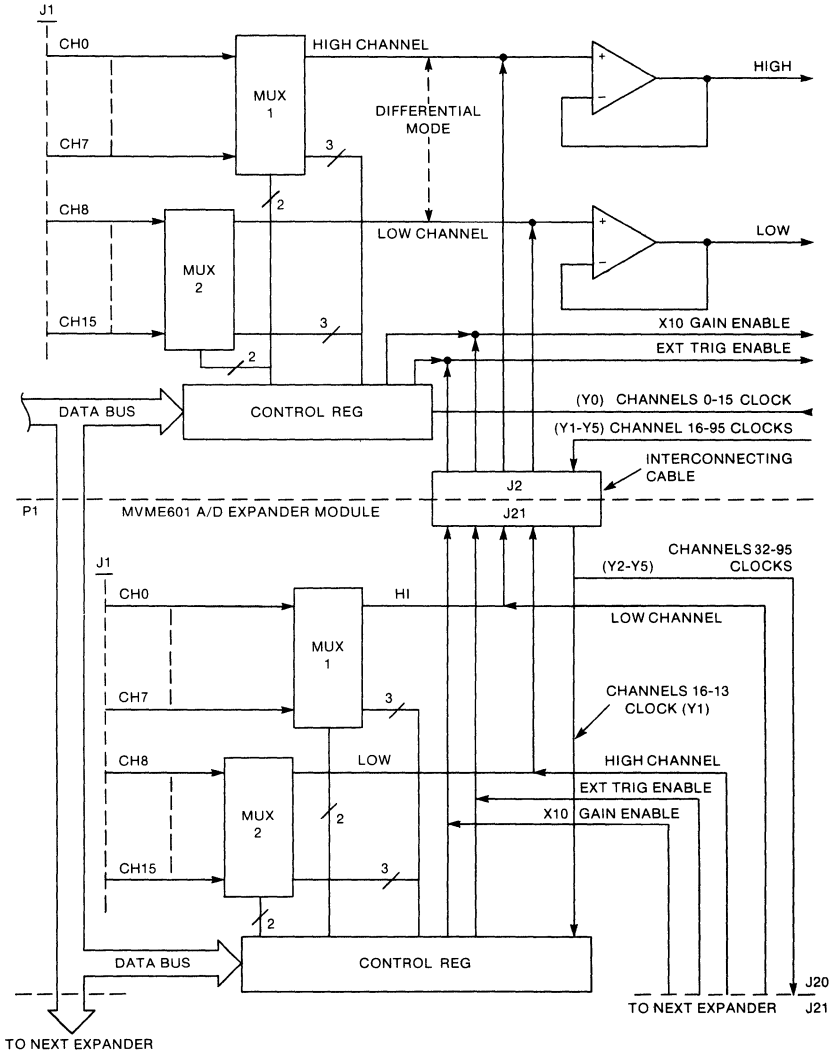


FIGURE 3 — MVME600/MVME601 Interconnection Diagram



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 Figure 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)

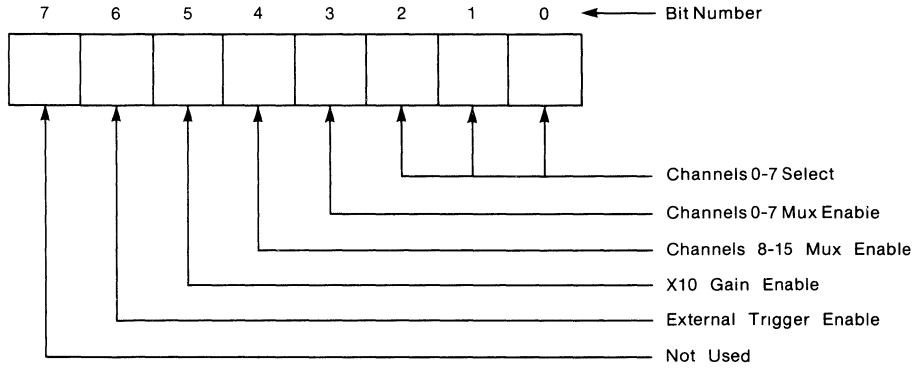
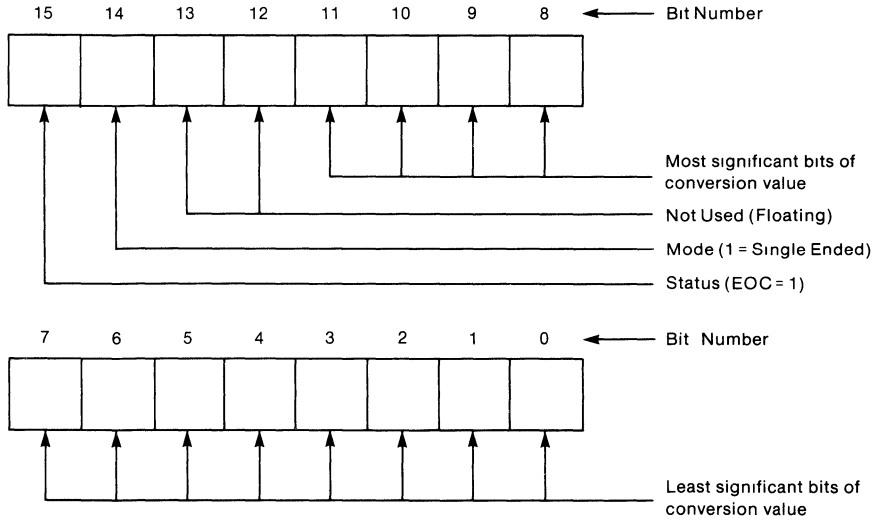


FIGURE 6 — MVME600 Data Registers (Read Only)



3

FIGURE 7 — Channel Selections for Single-Ended Measurements

Channel-Selection Bits / Multiplexer-Selection Bits	4	3	2	1	0	Channel Selected
	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	4	3	2	1	0	Channel Selected
	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

3

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

3

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

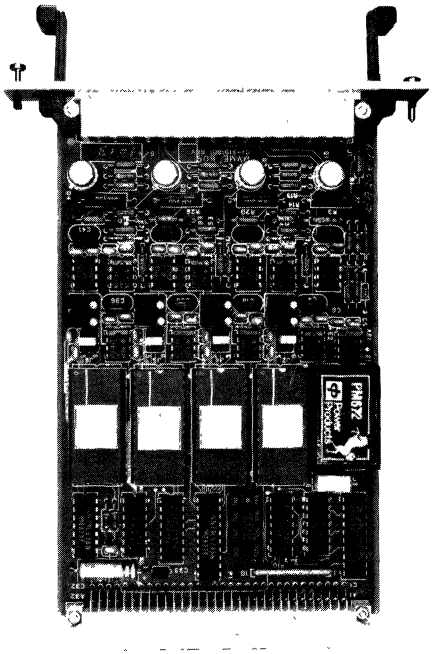
Related Documentation

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
- $\pm$0.1% FS, $\pm$1.0 LSB Accuracy
- Bipolar Voltage Outputs
- Three Full Scale Ranges $\pm$10 V; $\pm$5.0 V, $\pm$2.5 V
- 4–20 mA Current Loop Output on each Channel Simultaneous with Voltage
- Power Requirements $\pm$5.0 V, $\pm$12 V
- On Board DC/DC Converter Supplies $\pm$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: $\pm$2.5 V, $\pm$5.0 V or $\pm$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

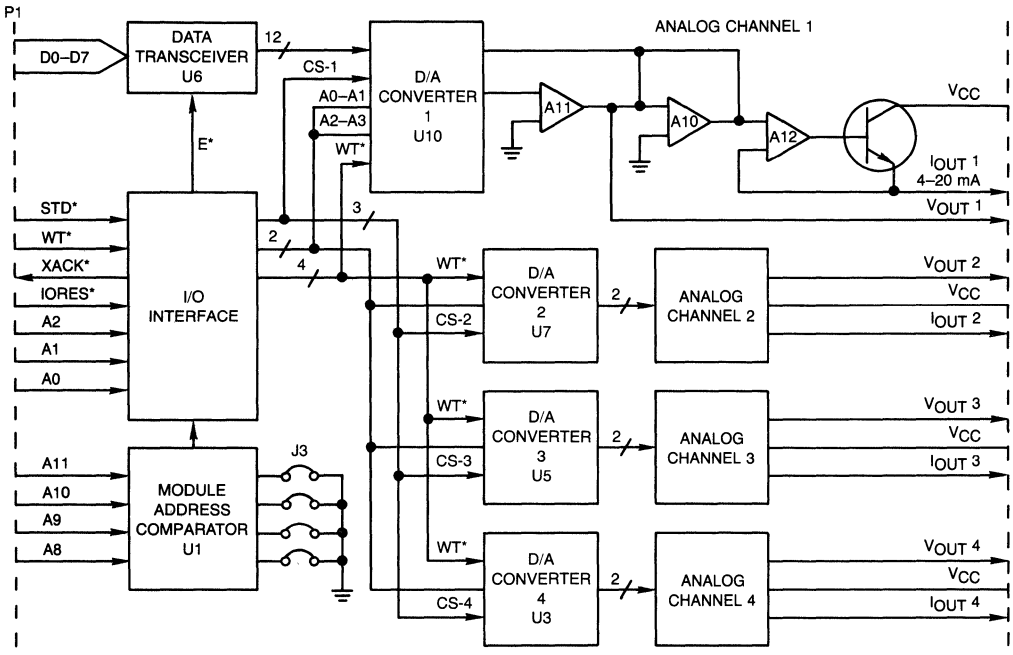
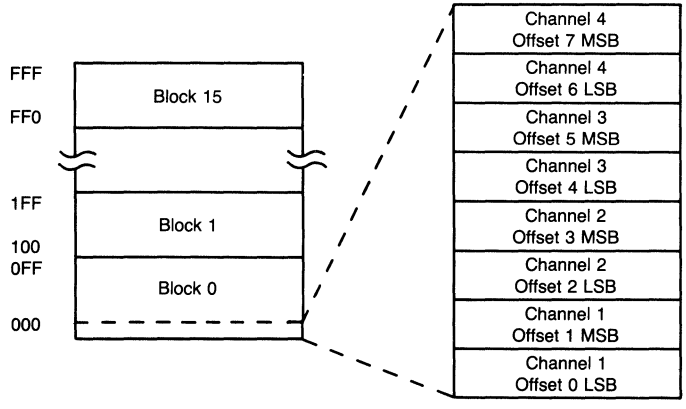


FIGURE 2 — MVME605 Memory Map



NOTE:
As shown, all header J3 jumpers in

TABLE 1 — Address Word Bit Functions

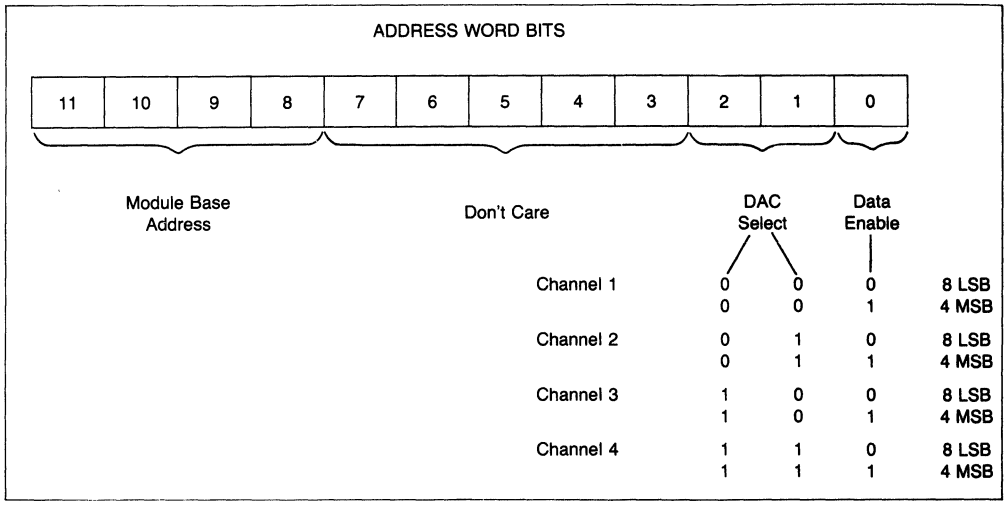


TABLE 2 — Outputs for Selected Input Values

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.

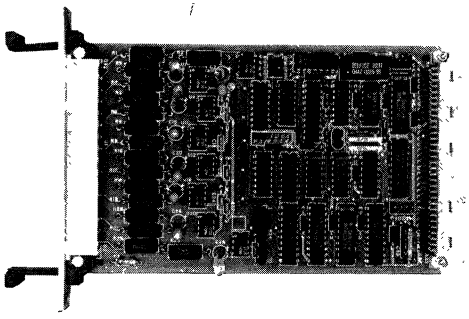
Related Documentation

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

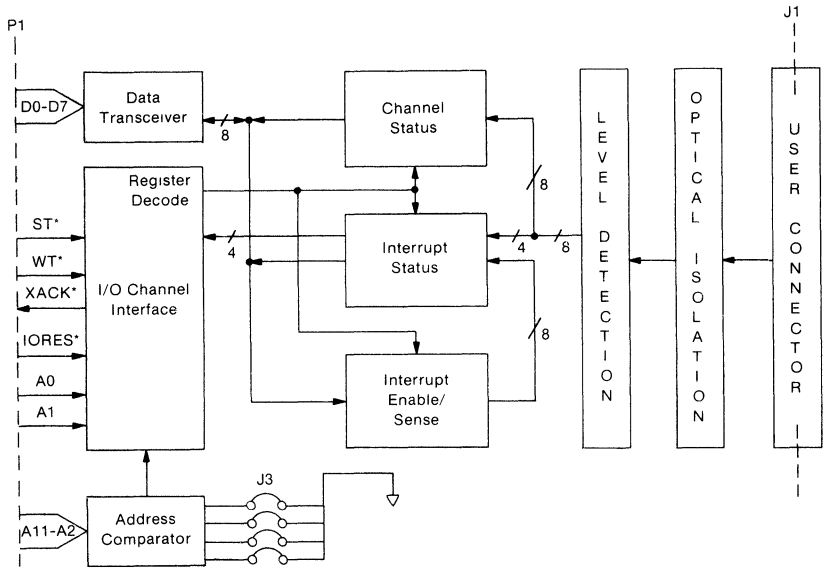
A microcomputer system for the process control, automatic test and other industrial and laboratory automation environments often utilizes both digital and analog equipment, imposing the need for suitable interfaces Such interfaces typically provide conversion into digital form of sampled voltages and 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

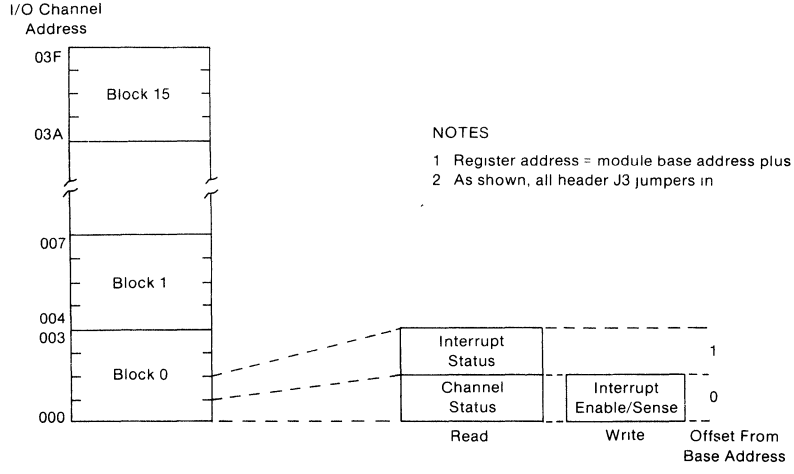
FIGURE 1 — MVME610 Functional Block Diagram



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

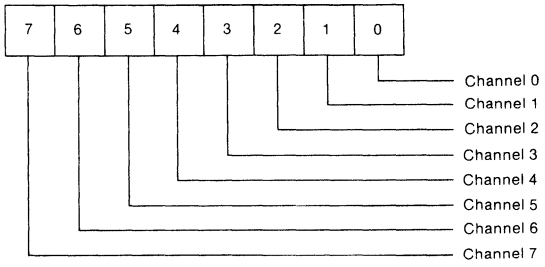
FIGURE 2 — MVME610 Memory Map



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

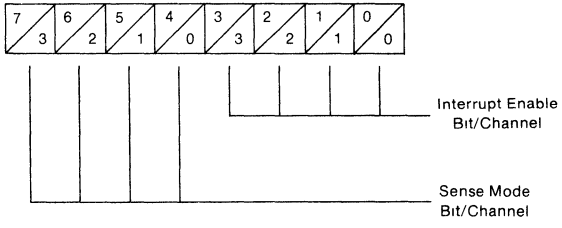


Interrupt Enable/Sense Register

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

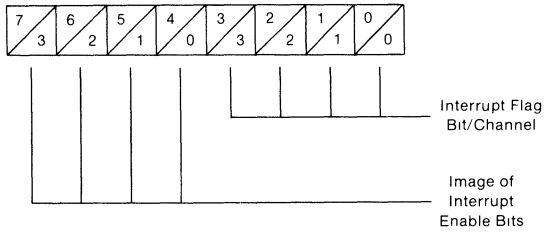
A software driver is available for use in systems utilizing the RMS68K executive kernel or the VERSAdos Operating System. For users desiring to write their own driver, a manual, "Guide to Writing a Device Driver", part number M68DRVGD, is available

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

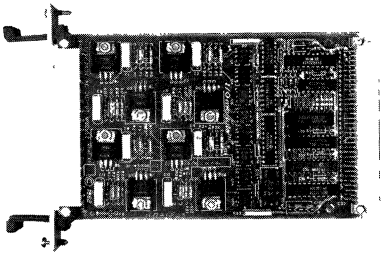
Related Documentation

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

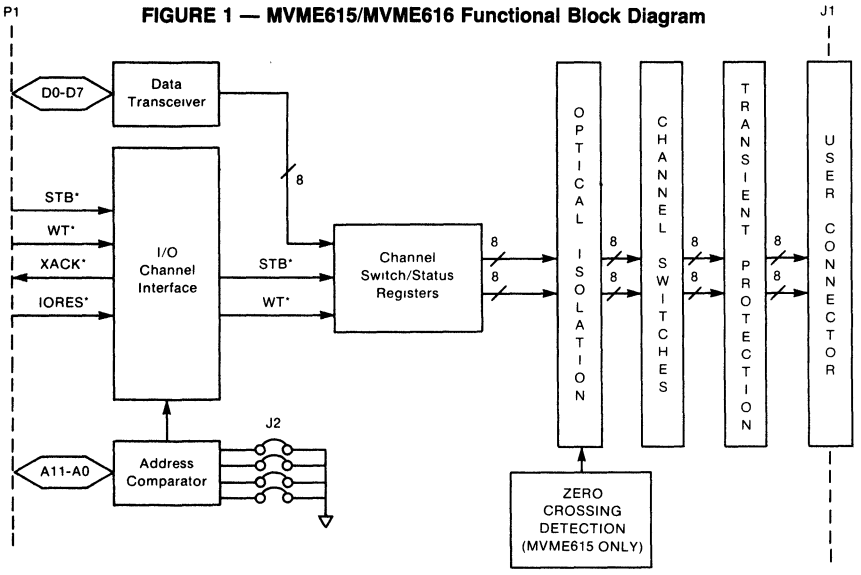
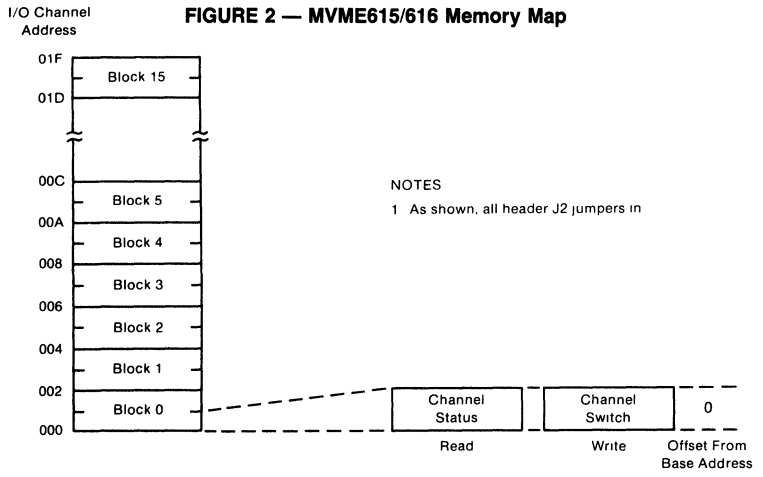


FIGURE 2 — MVME615/616 Memory Map



NOTES

1 As shown, all header J2 jumpers in

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

3

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

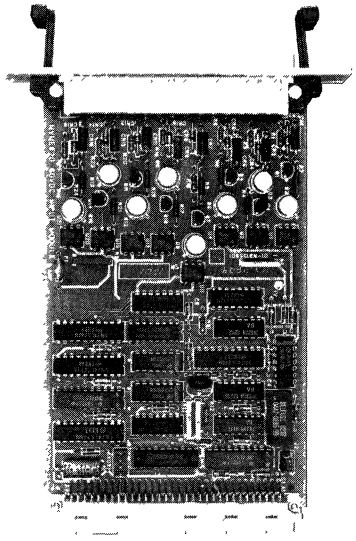
Related Documentation

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

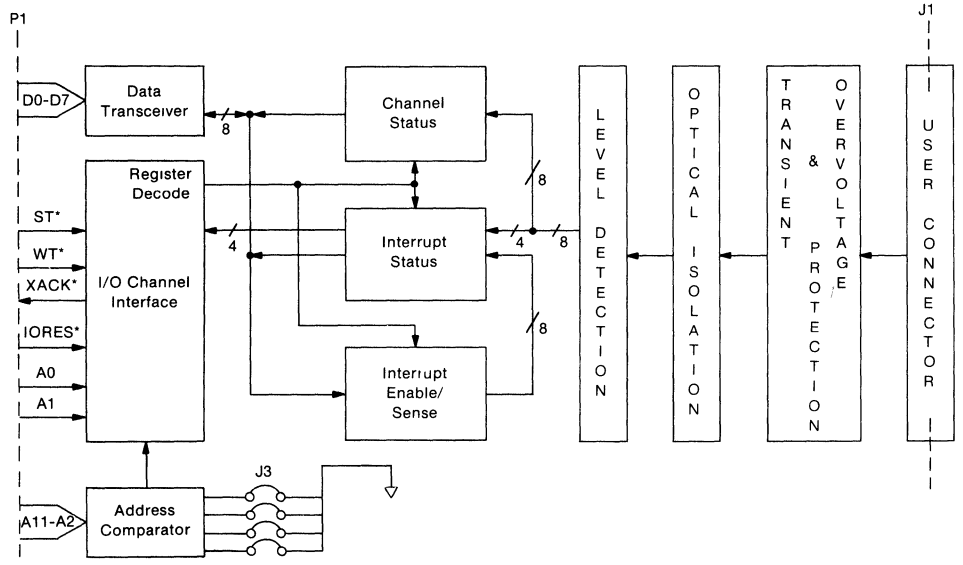
The MVME620 DC Input I/Omodule (DCIM) is a data acquisition device that can indicate to a host on the Motorola I/O Channel the energized/non-energized states of dc equipment connected to its eight input channels All eight DCIM channels set a status flag on sensing an energized input and four channels 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

3

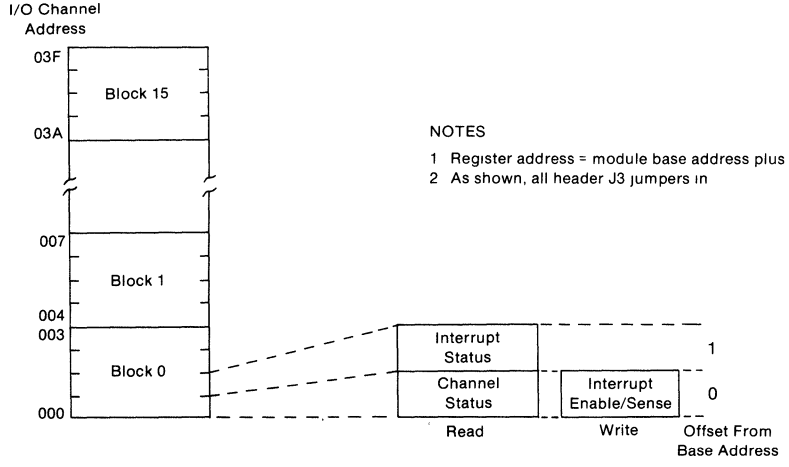
FIGURE 1 — MVME620 Functional Block Diagram



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.

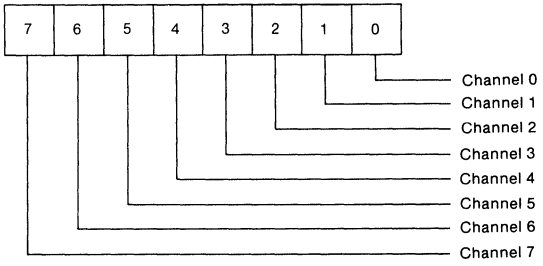
FIGURE 2 — MVME620 Memory Map



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

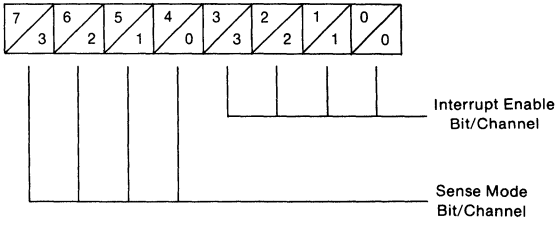


Interrupt Enable/Sense Register

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

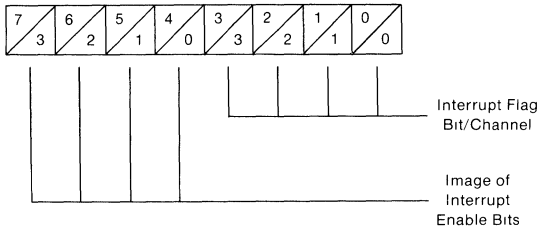
A software driver is available for use in systems utilizing the RMS68K executive kernel or the VERSAdos Operating System For users desiring to write their own driver, a manual, "Guide to Writing a Device Driver", part number M68DRVGD, is available

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

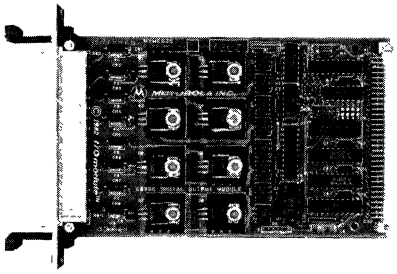
Related Documentation

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

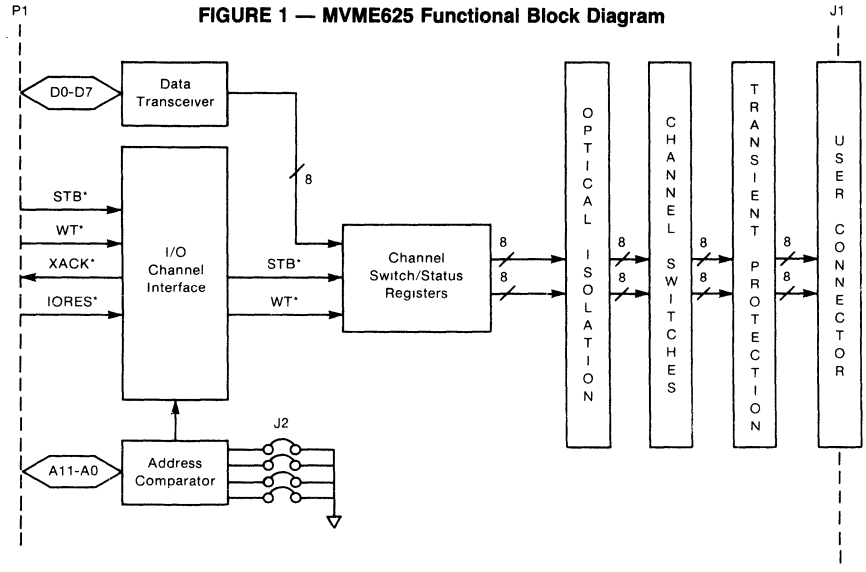
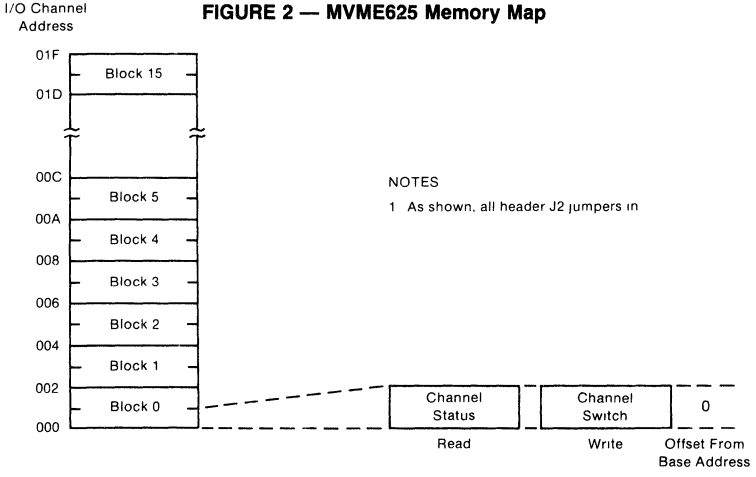


FIGURE 2 — MVME625 Memory Map



NOTES

1 As shown, all header J2 jumpers in

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

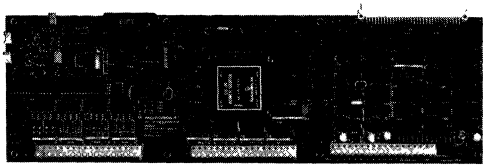
Specifications

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

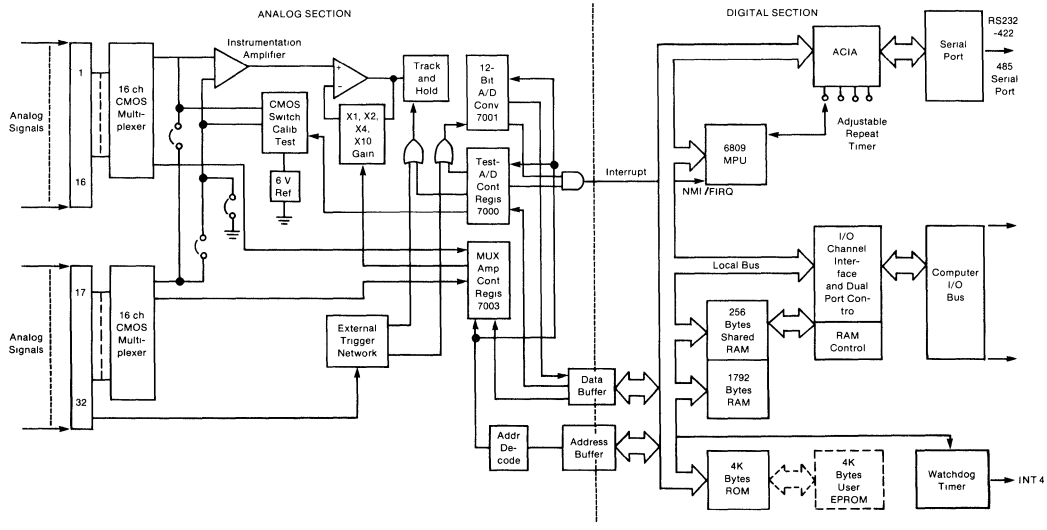
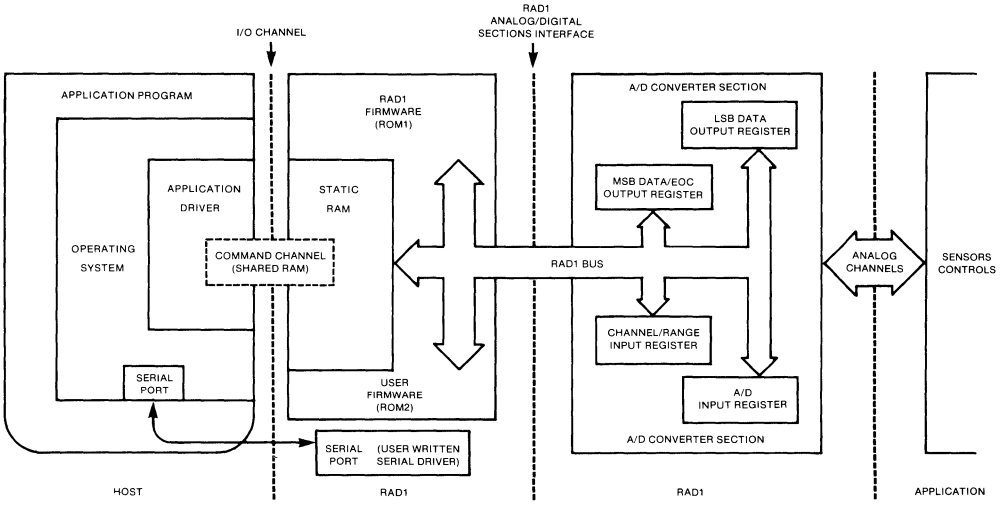


FIGURE 2 — M68RAD1 Digital and Analog Communications



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 × 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.

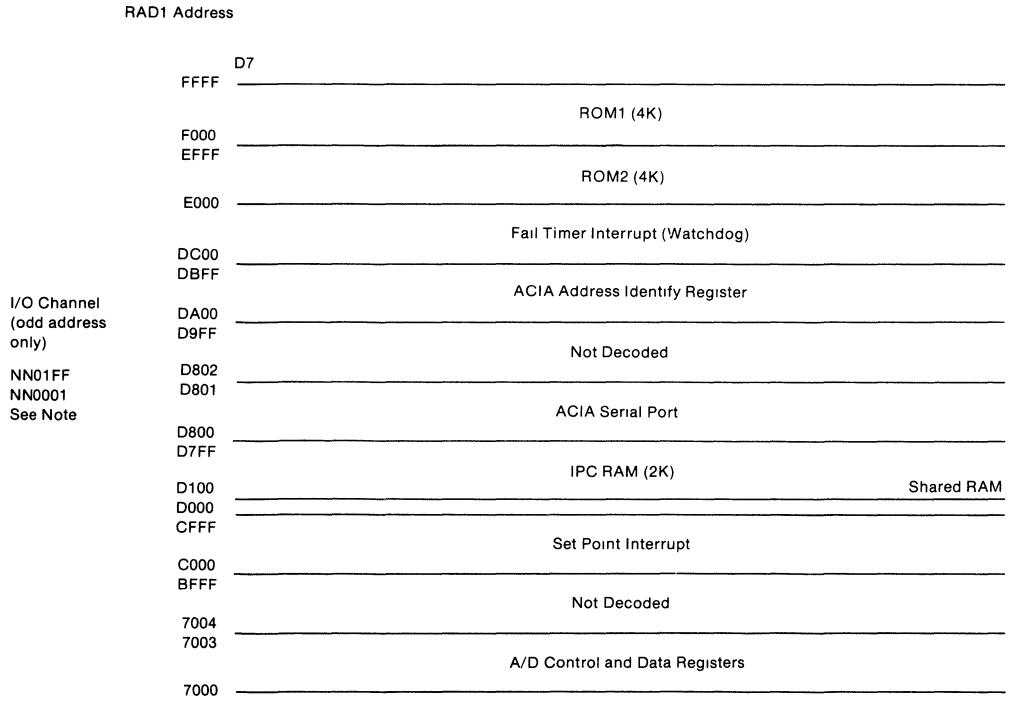
One of the 4K ROM sockets is available for a device containing user-written application routines (ROM2) In the other socket, ROM1 (supplied with the RAD1) contains a number of firmware routines which may be accessed by the driver to perform needed tasks. Additional routines or control software can be 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

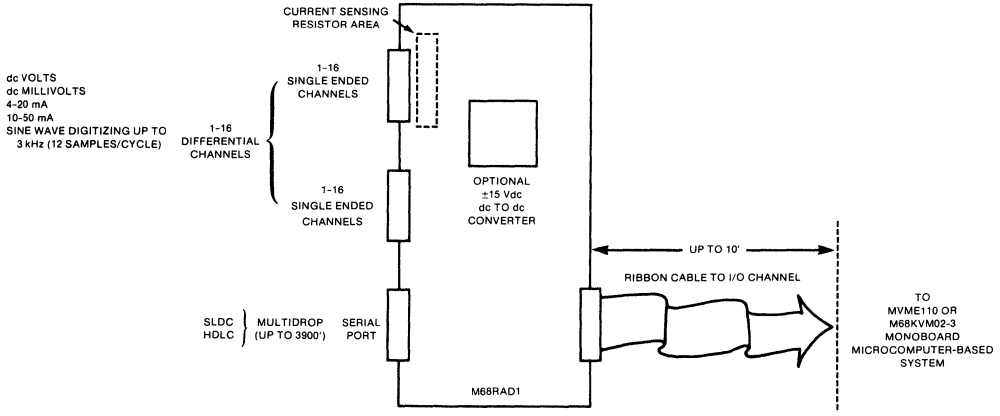
The RAD1 is designed to be used in many common applications in commerce and industry requiring analog to digital conversion for measurement and control. Typical uses include the measurement of voltages, temperature, pressure, strain and force. Almost any process in which a sensor is used to transform 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

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

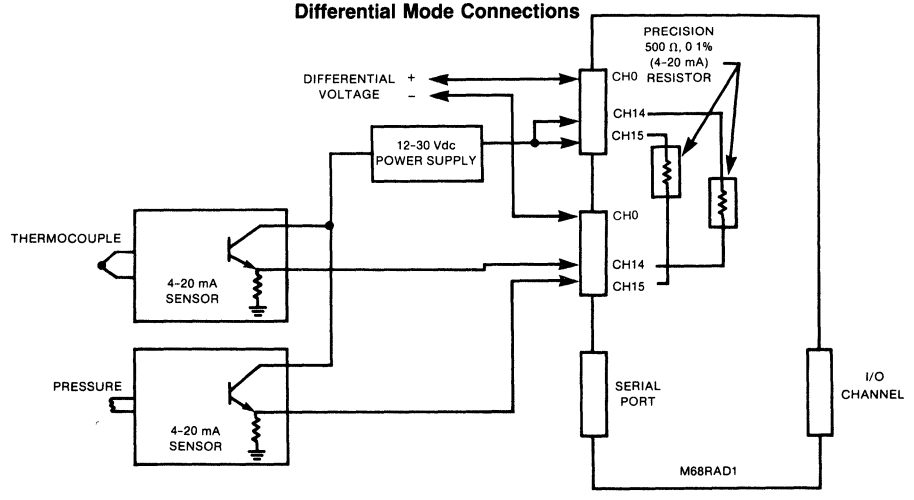
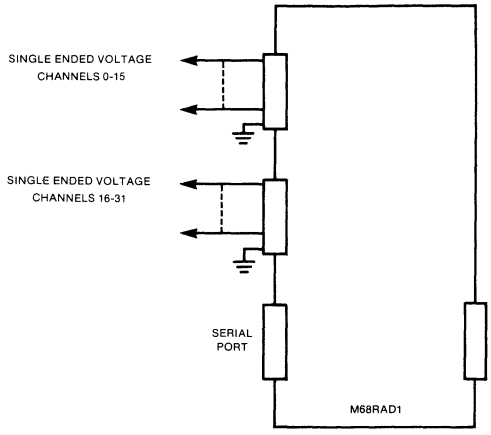
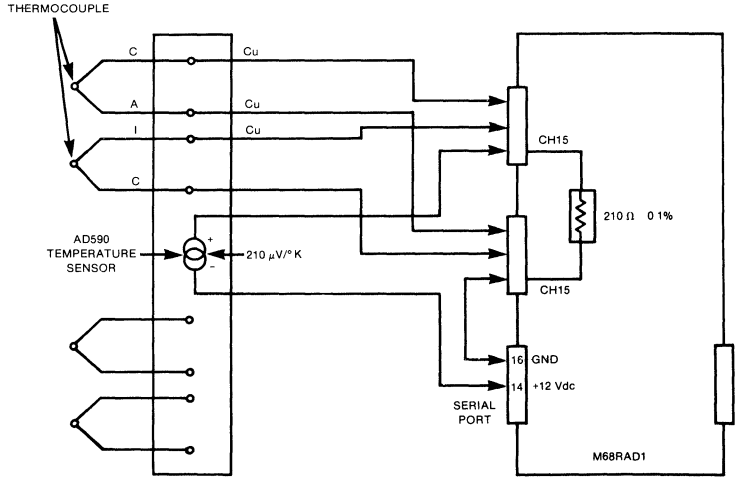


FIGURE 6 — Single Ended Mode (32 Channels)



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 — Electrovert 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

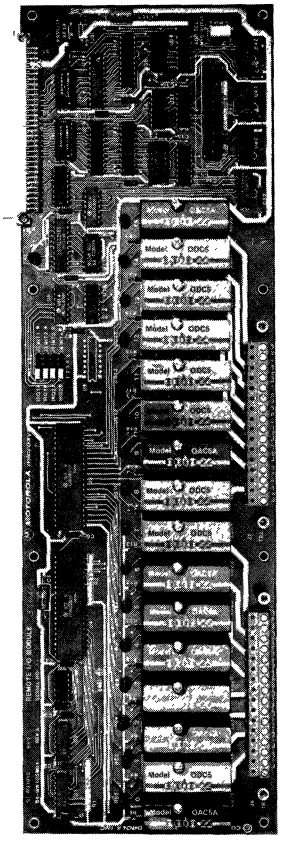
Other Modules in the MVME400 Series

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

3

FIGURE 1 — Remote I/O Module Block Diagram

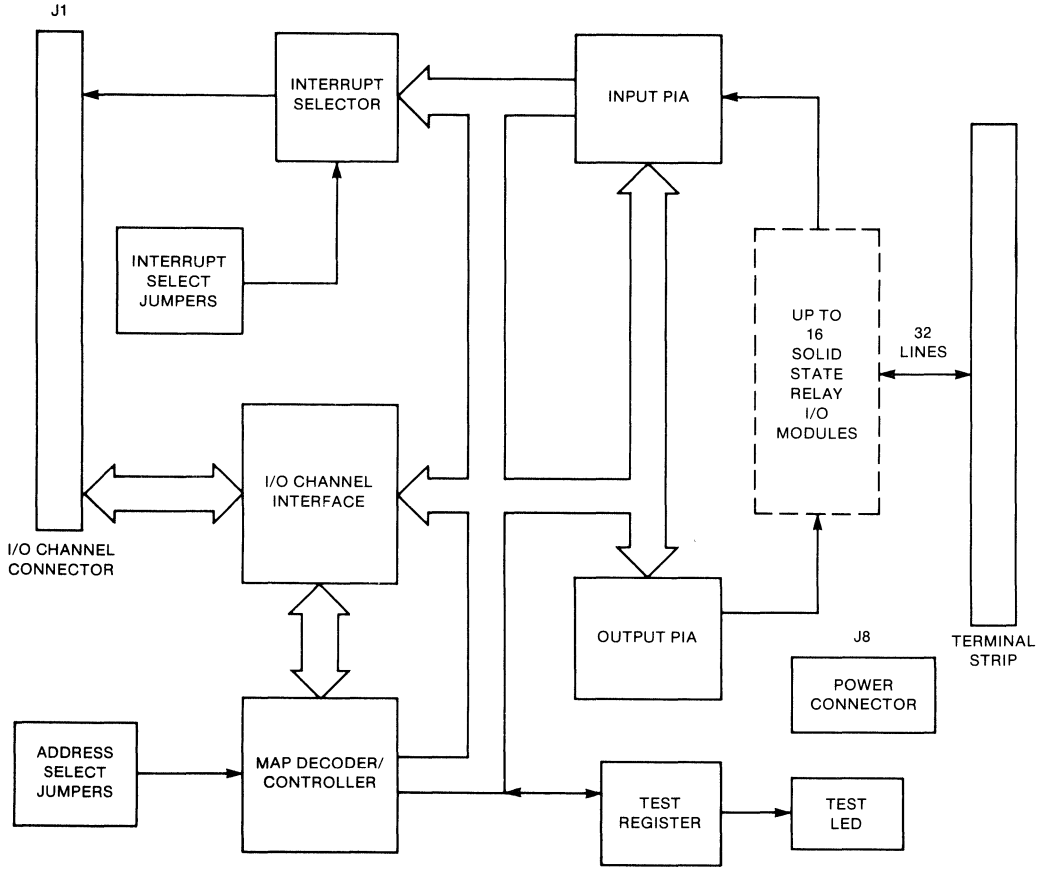


TABLE 1 — RIO1 Specifications — General

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.

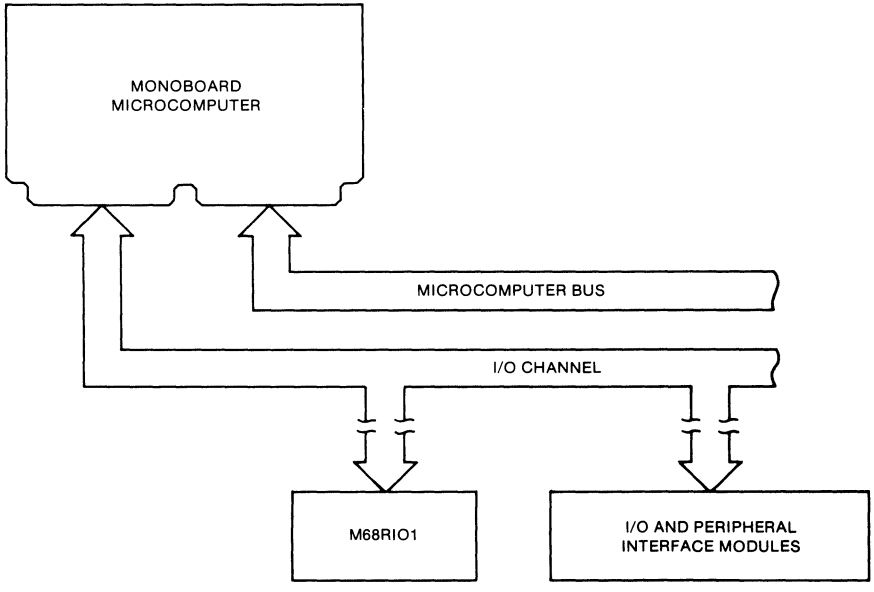
Interrupt Selection

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



3

FIGURE 3 — RIOM I/O Channel Memory Base Address Selection

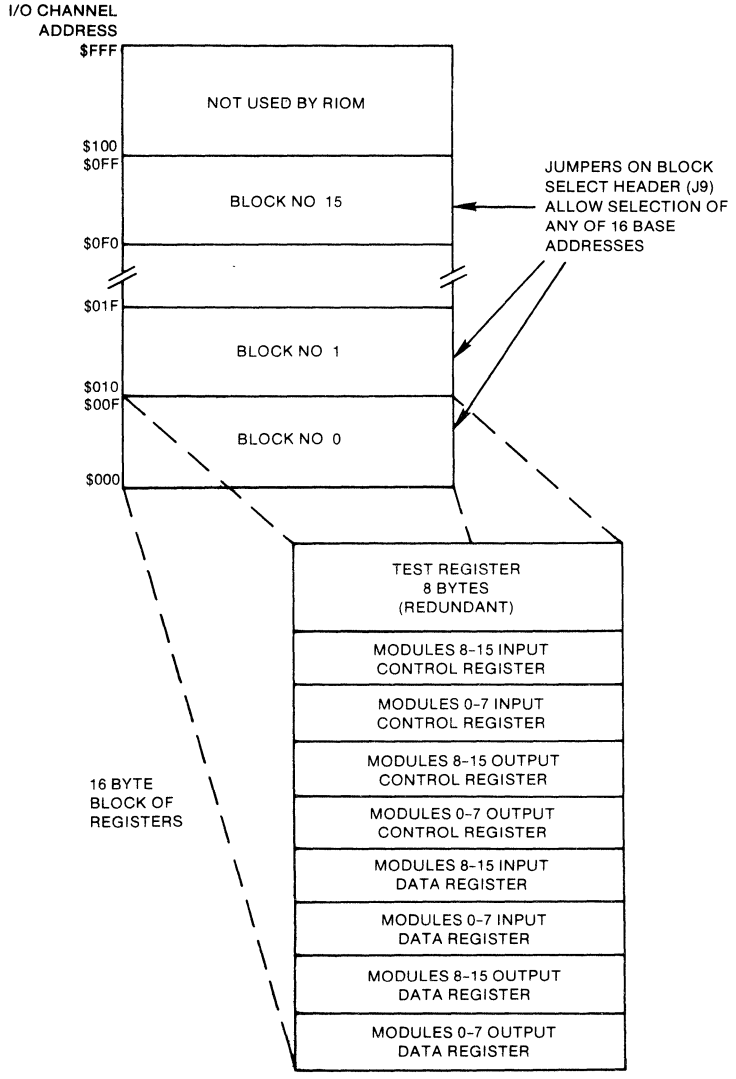


FIGURE 4 — RIOM PIA and Test Register Addresses

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

3

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

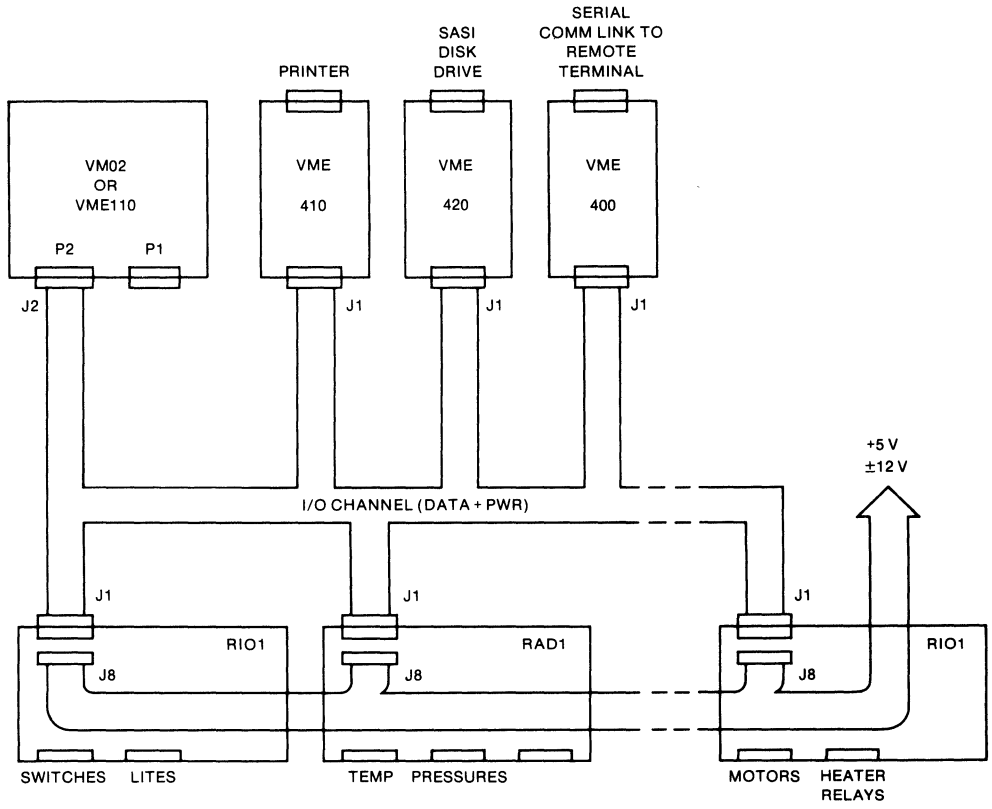
MVME400 Series

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.)

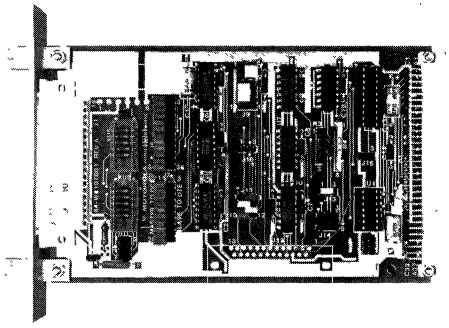
Related Documentation

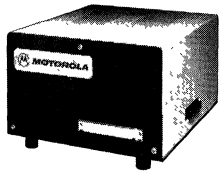
M68RIOCS/D2	Input/Output Channel Specification Manual
M68DRVGD	Guide to Writing a Device Driver

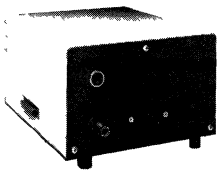
Other Modules in the MVME400 Series

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 6⅜-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.

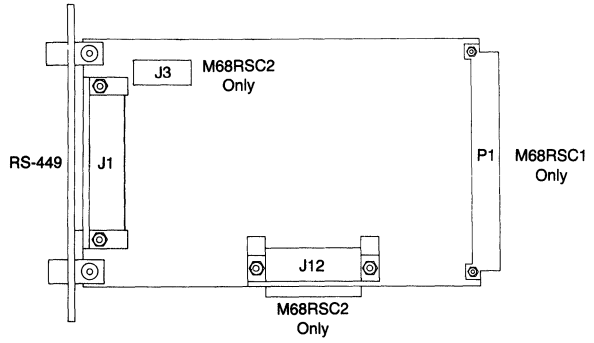


FIGURE 1 — RSCM Connector Locations

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.

3

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).

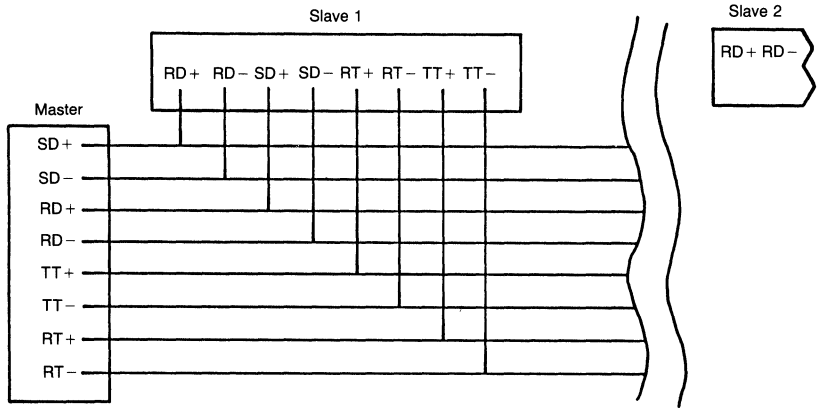
NOTE: Pins 1, 9–13, 19, 21–23, and 25 are not connected.

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

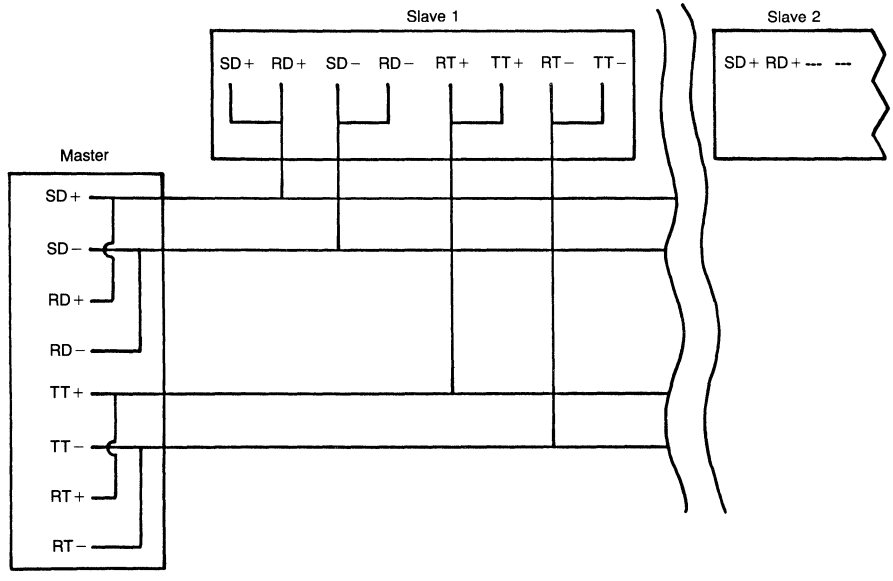
NOTES: See Table 3 for description of this signal. Pins A4, A6, A8–A14, A25, A27, C1, C11–C14, C27 are not connected.

FIGURE 2 — RSCM Interconnections for Full Duplex, Synchronous Operation



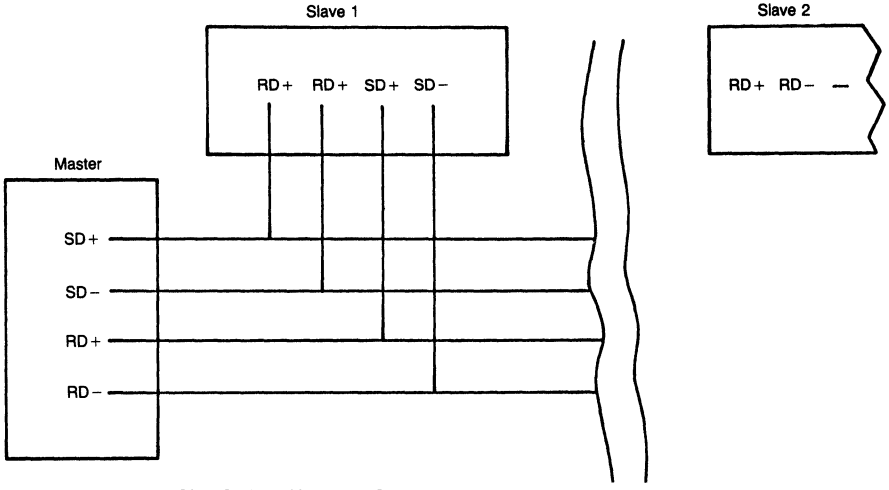
NOTE. Up To A Total Of 32 Stations (Master and Slaves) Can Be Interconnected

FIGURE 3 — RSCM Interconnections for Half-Duplex, Synchronous Operation



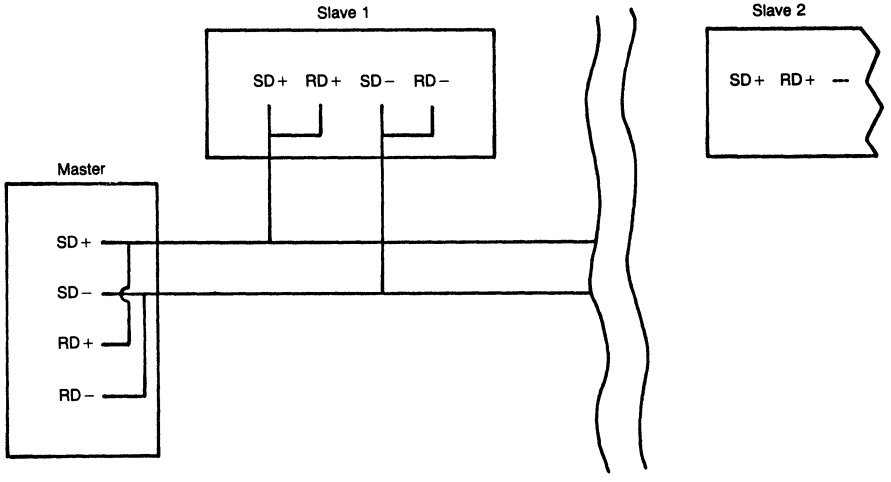
NOTE: Up To a Total Of 32 Stations (Master and Slaves) Can Be Interconnected
(Header J5 Jumpers Are Installed.)

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

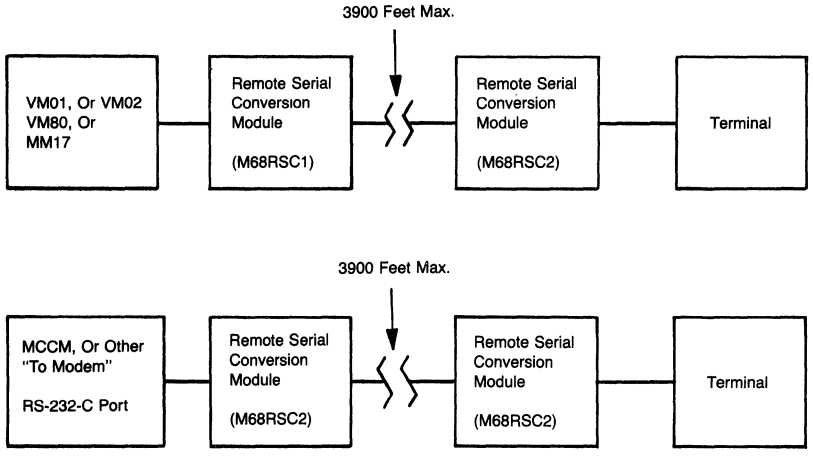


FIGURE 7 — RSCM Application in Multidrop Use

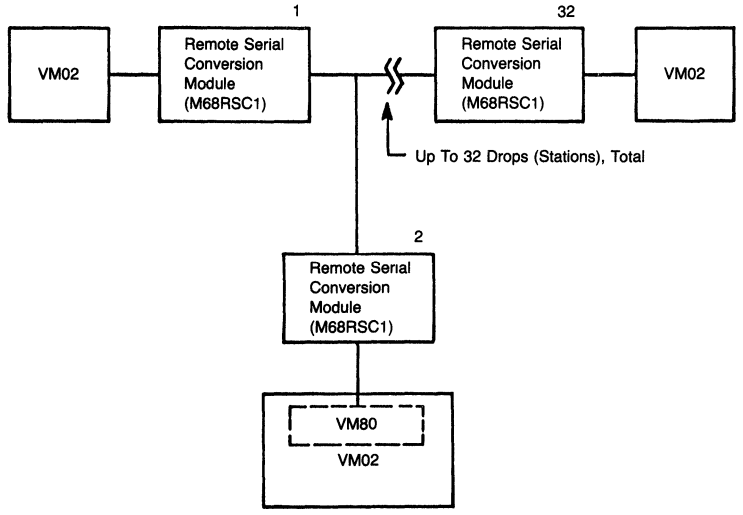
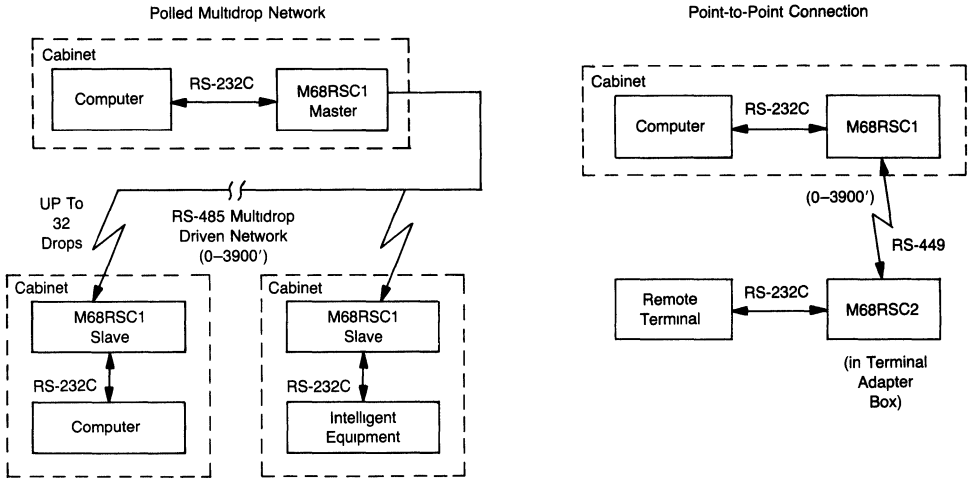


FIGURE 8 — Typical RSCM Systems Applications



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

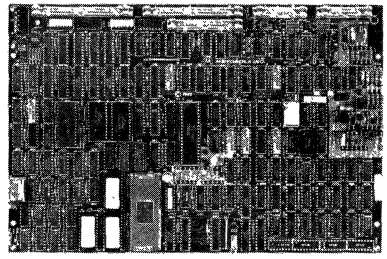
M68RSC1/D1	Remote Serial Conversion Module and Terminal Adapter Enclosure User's Manual.

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

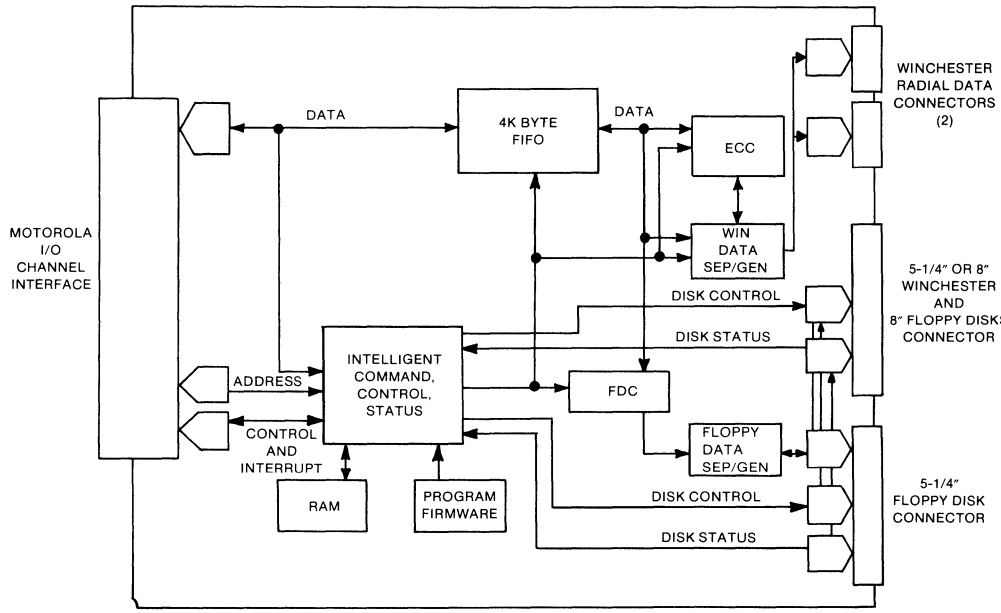
I/O Channel

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

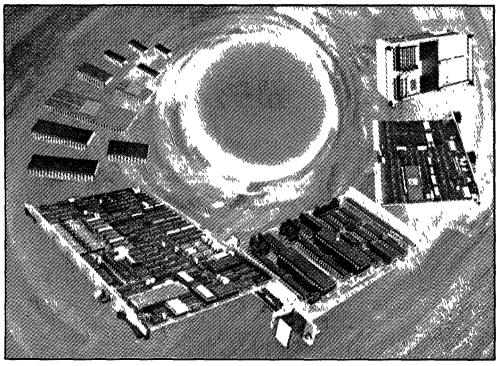


Mechanical and Environmental Specifications

Characteristic	Specification
Power Requirements	+5 Vdc ±5% @ 6 0 A max -5 Vdc ±5% (-8 to -18 Vdc optional) @ 100 mA max
Temperature Operating Storage	 0° C to +70° C -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 UNIX-derived 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

RMS68K M68000 Real-Time Multitasking Software

- Real-Time Multitasking Capability
- Task Scheduling Based on Task Priority Levels (0 to 255)
- Task Synchronization Capability Using Semaphores
- Resource Sharing Using Semaphores
- Support For Any Number of Tasks
- Inter-Task Communication
- Server Task Capability
- Memory Management and Allocation
- Fast Interrupt Response
- Wide Variety of Timer Options
- User Written I/O Device Driver Capability
- Exception Monitoring Facility
- Highly Modularized Functions
- System Generation Capability to Custom-Configure the System
- Trace Table Capability to Ease Debugging of System
- System Initialization Capability
- ROM or RAM Based

The Motorola M68000 Family Real-Time Multitasking System Software, RMS68K, is a multitasking kernel around which real-time application systems can be built. The powerful and universal characteristics of RMS68K allow a wide variety of application systems without large expenditures for design and programming efforts on the complex real-time and multitasking functions.

RMS68K can be used in systems which are built around the Motorola VERSAmodule or VMEmodule Monoboard Microcomputer or in customer designed hardware incorporating the Motorola MC68000 Family microprocessor. RMS68K is compatible with the EXORmacs Development System, the VME/10 Microcomputer System and the VERSAdos operating system. Therefore, programs designed to execute under the control of RMS68K will also execute under the control of VERSAdos. This provides a development environment which is compatible with the target system.

REAL-TIME SYSTEM 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

RMS68K Functions

RMS68K is comprised of a task controller, an inter-task communication facility, an optional memory management facility, and an initialization facility These facilities provide 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
- Provide inter-task communication and synchronization
- Manage and allocate memory
- System initialization capability
- Diagnostic feedback during error conditions

RMS68K Structure

RMS68K is structured in six levels or layers (see Figure 1), with each level performing a particular range of functions

Level 0 supplies processor management functions, including task dispatching, primitive synchronization, exception handling, and interrupt dispatching

Level 1 supplies physical memory management functions

Level 2 supplies utility functions.

Level 3 supplies task address space and memory segment management functions

Level 4 supplies task creation, deletion, and execution controlling functions.

Level 5 (an optional layer of RMS68K) provides the channel-oriented, physical I/O functions

The functions provided by levels 3, 4, and 5 are directly available to user tasks The functions provided by levels 0, 1, and 2 are not directly available to user tasks, but provide support to level 3, 4, and 5 functions This multi-layering allows changes to be made within levels 0, 1, and 2 without affecting the user interface to RMS68K

User tasks can request RSM68K to perform level 3, 4, or 5 functions by using executive directives An executive directive contains all of the information needed by RMS68K to perform the desired function

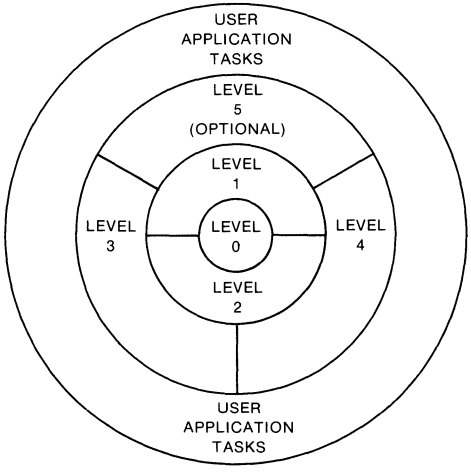


FIGURE 1 — RMS68K Functional Levels

Memory Requirements

RMS68K is supplied in two parts—one part is the level 0 through 4 software; the other part is optional level 5. The maximum memory requirements are given in Table 1. The user may elect to exclude certain functions in his particular application system, which would decrease memory requirements.

TABLE 1 — RMS68K Memory Requirements

Function	RAM (Parameter Data)	ROM/RAM (Program Code)
Level 0 through Level 4	3000 bytes	15K bytes
Level 5	See note below	2 0K bytes

NOTE Dynamic system tables created by levels are not included

Application Hardware Requirements

The hardware requirements for an RMS68K based application system are:

- MC68000 or MC68010 microprocessor
- Appropriate memory
- 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 real-time 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.

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 Figure 2 is the task state transition diagram

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)

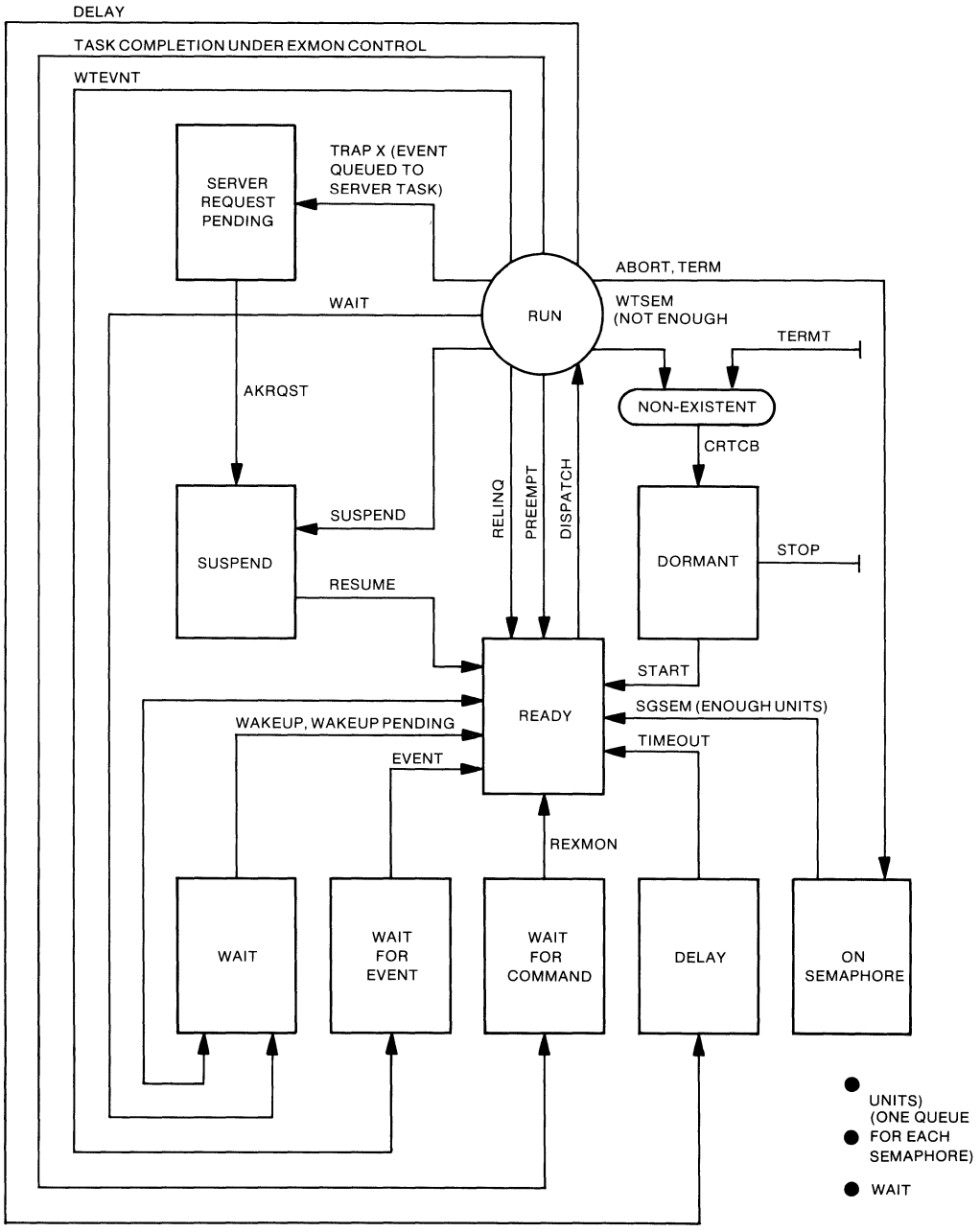


FIGURE 2 — RMS68K Task State Transition Diagram

TASK MANAGEMENT

Task management is one of the broad functional areas of RMS68K Levels 3 and 4 of RMS68K software provide this task management function This task management functional area can be broken down into several smaller service areas:

1 Memory allocation and management
2 Task state control
3 Task event control
4 Task synchronization
5 Special function task control
6 Interrupt service management

There is a set of task management directives which tasks can use to request the performance of task management services by RMS68K.

1. Memory Allocation and Management Directives

A list and brief description of all the directives in the memory allocation and management category

GTSEG	Allocate a new program code or data memory segment to the target task by placing it in the task's address space
DESEG	Delete a program code or code memory segment from the target task's address space
GTASQ	Allocate an asynchronous service queue (ASQ) to the target task
DEASQ	Deallocate the requesting task's ASQ
DCLSHR	Make a memory segment available for shared access
ATTSEG	Place an existing memory segment (shareable) into the requesting task's address space
SHRSEG	Place an existing memory segment (shareable) into another task's address space
TRSEG	Remove a memory segment from the requesting task's address space and place it in the address space of another task
RCVSA	Return a description of the specified segment to the requesting task

2. Task State Control

Directives are provided to control task execution by means of moving a task through various states A task can control its own state transitions or another task's state transitions Following is a list and brief description of each of the directives in the task state control category

CRTCB	Create a task control block (TCB) for a new task and place the task in the DORMANT state
START	Move the target task to the READY state
TERMT	Terminate task by releasing resources and deleting the TCB of the target task so that the task no longer exists in the system.
STOP	Move the target task to the DORMANT state
RESUME	Move the target task from the SUSPEND state to the READY state.
WAKEUP	Move the target task from the WAIT state to the READY state
ABORT	Initiate a task's own abnormal termination Its TCB is deleted and the task no longer exists in the system
TERM	Initiate a task's own normal termination Its TCB is deleted and the task no longer exists in the system
SUSPND	A task moves itself into the SUSPEND state.
WAIT	A task moves itself into the WAIT state
RELINQ	A task moves itself to the READY state so that RTEX can perform a dispatch cycle.
SETASQ	A task changes the status of its ASQ and/or ASR
DELAY	A task moves itself to the DELAY state for a specified period of time
DELAYW	A combination of DELAY WTEVNT and WAIT directives
WTEVNT	A task moves itself into the WAIT-FOR-EVENT state until an event is placed in its ASQ
SETPRI	A task changes its own priority or the priority of another task to the specified value
RQSTPA	A task requests the timed periodic activation of itself or another task
EXPVCT	A task announces the handling of its own exceptions
TRPVCT	A task announces the handling of its own trap instructions

3. Task Event Control

These directives can be used by a task for event queueing and event servicing

QEVNT	An event is placed into the ASQ of the target task
RDEVNT	The next event in the requesting task's ASQ is moved into the task's receiving area
RTEVNT	Return to the point of interruption of a task upon completion of that task's ASR processing

4. Task Synchronization Directives

Tasks can use semaphores to synchronize activities and control resources Several task synchronization directives, which tasks can use to control these semaphores, are available

ATSEM	A task attaches to a semaphore and acquires use of that semaphore The semaphore is created and given an initial signal count if it does not already exist.
CRSEM	A task creates a new semaphore or resets the initial count of an existing semaphore
DESEM	A task releases use of a semaphore.
DESEMA	A task releases use of all semaphores to which it is attached.
WTSEM	A task requests and, if necessary, waits for semaphore-controlled access or execution to be granted.

Task Synchronization Directives (continued)

SGSEM A task signals release of a semaphore-controlled resource or execution.

5. Special Function Task Control

The directives in this category are used to control server tasks and exception monitor tasks.

Server Task Control

A server task is able to receive and process requests from any task in the system A user task issues a request to a server task by executing a TRAP instruction, thus, a server task appears to function as part of RMS68K

SERVER A task establishes itself as a server task

AKRQST A server task acknowledges receipt or 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 Task Control

An exception monitor task provides execution control over a target task or emergency processing when a task fails Exception events which will cause a target task to stop execution and notification to be sent to the target task's exception monitor task are specified in an exception monitor mask associated with the target task An exception monitor task can also control the operation of a target task in a trace mode

EXMON A target task becomes associated with an exception monitor task

DEXMON A target task detaches from an exception monitor task

EXMMSK Events of interest to an exception monitor task are specified

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

REXMON A target task executes as directed by an exception monitor task

6. Interrupt Service Management

A task can include one or more routines which are activated as the result of an external interrupt (or exception), and execute in the MC68000 user hardware state at the priority level of the interrupt Such a routine is called an interrupt service routine (ISR) This mechanism is useful in creating I/O device drivers

CISR A task configures one of its ISR's to respond to a particular exception.

SINT A task simulates an exception (interrupt)

RTE An ISR returns from execution.

Miscellaneous RMS68K Functions

There are some miscellaneous RMS68K functions which do not fit into the categories discussed in the preceding paragraphs.

Date and Time Directives

STDTIM A system task sets the system date and time

GTDTIM A task obtains the current system date and time.

Task or System Information Directives

Directives are provided to retrieve some types of information about other tasks or about system execution

TSKATTR A task receives the user number and attributes of a target task

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

TSKINFO A system task requests a copy of a target task's Task Control Block

SNAPTRAC The contents of the system trace table are copied into a buffer in the calling task's address space

System Enhancement Directive

CDIR This directive provides a means by which new executive directives can be created for use by specific applications

DEVICE I/O HANDLERS

Channel Management Request (CMR) routines reside as part of the Executive, RMS68K, and represent level 5 of the EXEC structure 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, among other things, is defined to be a single contiguous portion of memory mapped I/O space, associated with one or more polled identity conditions of interrupt Up to four conditions of interrupt for a device can be associated with a single channel The conditions of interrupts associated with a channel are defined when the channel is allocated

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

Supervisor/Subordinate Channels

Some devices possess more than one channel (e.g , some serial controller chips) In many cases, such a device gives the same interrupt, regardless of the channel which caused it, and each of the channels must inspect the device to determine whether the interrupt belongs to that channel

This can create a problem if reading the device's registers in some way signals the device (it might clear the interrupt, for example). In order to provide for this situation, CMR provides a scheme for a single channel to read the device and communicate with (and pass control to) other channels

For such a situation, supervisor and subordinate channels must be used A SUPERVISOR CHANNEL is allocated first (option bit 4 on the allocate command), and then the SUBORDINATE CHANNELS are allocated (option bit 3) with the supervisor's channel mnemonic in field eight The subordinate channels receive requests to initiate I/O as usual, but all their interrupts are directed to the supervisor. The supervisor (which must be written to fit the device and subordinate channels) then fields the interrupt and farms it out to the appropriate subordinate in whatever manner the designer desires For example, the details of entry into the subordinate channel for proper interrupt handling are left undefined — all CMR cares about is giving the interrupt to the supervisor

The supporting data structure is a linked list starting from the supervisor channel and containing all (there is no limit) subordinates, via the longword CCBSUB in the CCB It is expected that the supervisor will traverse this list in search of a driver corresponding to the particular interrupt type

The supervisor channel is (of course) linked into the vector chain for the specified vector, but the subordinate channels are not. Hence, the only way they will ever be entered for interrupt service is if the supervisor channel jumps into them. Figure 3 depicts the chaining of subordinate channels in relation to the vector chain

Ordering Information

Part Number	Description
M68K0RMS68K	M68000 Family Real-Time Multitasking Object Software on VERSAdos 8" Diskettes with User's Manual
M68K0RMS68KH	M68000 Family Real-Time Multitasking Object Software on VERSAdos Cartridge Disk with User's Manual
M68VKXBRMS68K	M68000 Family Real-Time Multitasking Object Software on VERSAdos 5¼" diskettes with User's Manual.
M68K0RMS68KT	M68000 Family Real-Time Multitasking Source Software on VERSAdos cartridge with User's Manual.
M68VKXSRMS68K	M68000 Family Real-Time Multitasking Source Software on VERSAdos 5¼" diskettes with User's Manual.
M68VKMSRM68K	M68000 Family Real-Time Multitasking Source Software on VERSAdos 25 Mbyte Lark cartridge with User's Manual.
M68KRMS68K(D8)	M68000 Real-Time Multitasking Software User's Manual

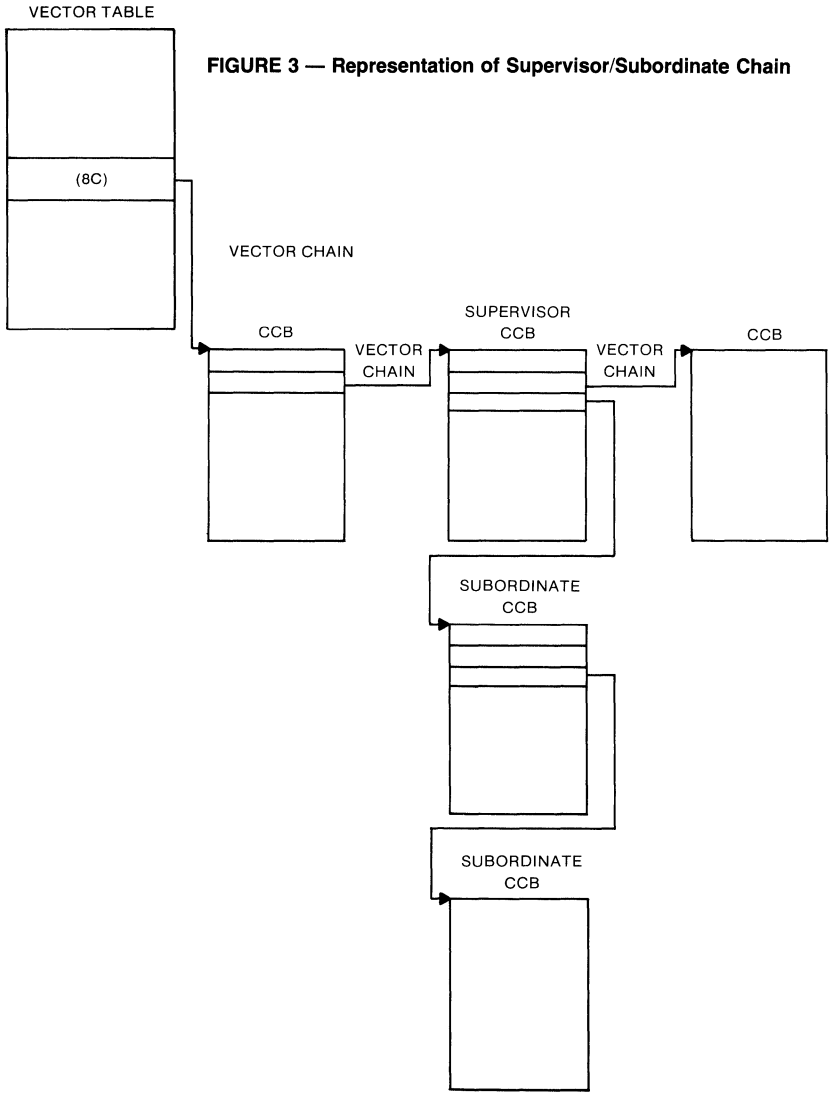


FIGURE 3 — Representation of Supervisor/Subordinate Chain

VERSAdos 4.3 Operating System

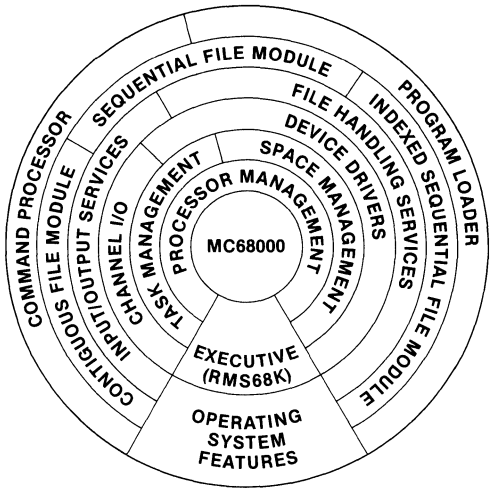
- Modular design of major programs (such as Real-Time Executive Services and Input/Output Control) to allow adding user functions with minimal effort
- System functions which require minimal operator intervention
- System security with redundant safeguards to prevent system failures
- Automatic system generation capability
- Support for VERSAmodule Monoboard Computers
- Support for VMEmodule Monoboard Computers
- Support for VMC 68/2
- Support for EXORmacs
- Support for VME/10
- Hard Disk oriented
- Optimum processor and memory utilization through multitasking and hardware memory mapping

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 VERSAmodule and VMEmodule board products and the MC68000 family VERSAdos provides the basic capability to service a wide range of requirements — including batch, 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 VERSAmodule and VMEmodule 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
- Task management and services
- Support for server tasks (user customized expansion of the Operating System)
- 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 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

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 is the nucleus of the Operating System and is available as a separate product known as RMS68K This is a real-time multitasking kernel 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

There are six layers, with each level performing a particular range of functions (as shown in Figure 1)

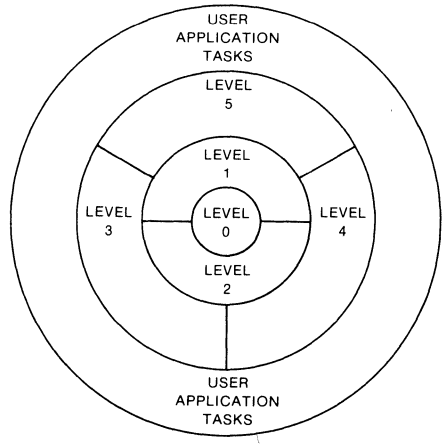


FIGURE 1 — VERSAdos Executive Functional Levels

Level 0 supplies processor management functions, including task dispatching, primitive synchronization, exception handling, and interrupt dispatching

Level 1 supplies physical memory management functions.

Level 2 supplies utility functions.

Level 3 supplies task address space and memory segment management functions.

Level 4 handles the task creation, deletion, and execution controlling functions.

Level 5 provides the channel-oriented, physical I/O functions.

The functions provided by levels 3, 4, and 5 are directly available to user tasks The functions provided by levels 0, 1, and 2 are not directly available to user tasks, but provide support to level 3, 4, and 5 functions This multi-layering allows changes to be made within levels 0, 1, and 2 without affecting the user interface to the RMS68K Executive

User tasks can request the RMS68K Executive to perform level 3, 4, or 5 functions by using executive directives An executive directive contains all of the information required to perform the desired function

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 VERSAdos 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 moving the tasks through various task states Figure 2 is the task state transition diagram

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)

TASK MANAGEMENT

Task management is one of the broad functional areas of RMS68K Levels 3 and 4 of the Executive provide this task management function This task management functional area can be broken down into several smaller service areas

1 Memory allocation and management
2 Task state control
3 Task event control
4 Task synchronization
5 Special function task control
6. Interrupt service management

There is a set of task management directives which tasks can use to request the performance of task management services

1. Memory Allocation and Management Directives

A list and brief description of all the directives in the memory allocation and management category

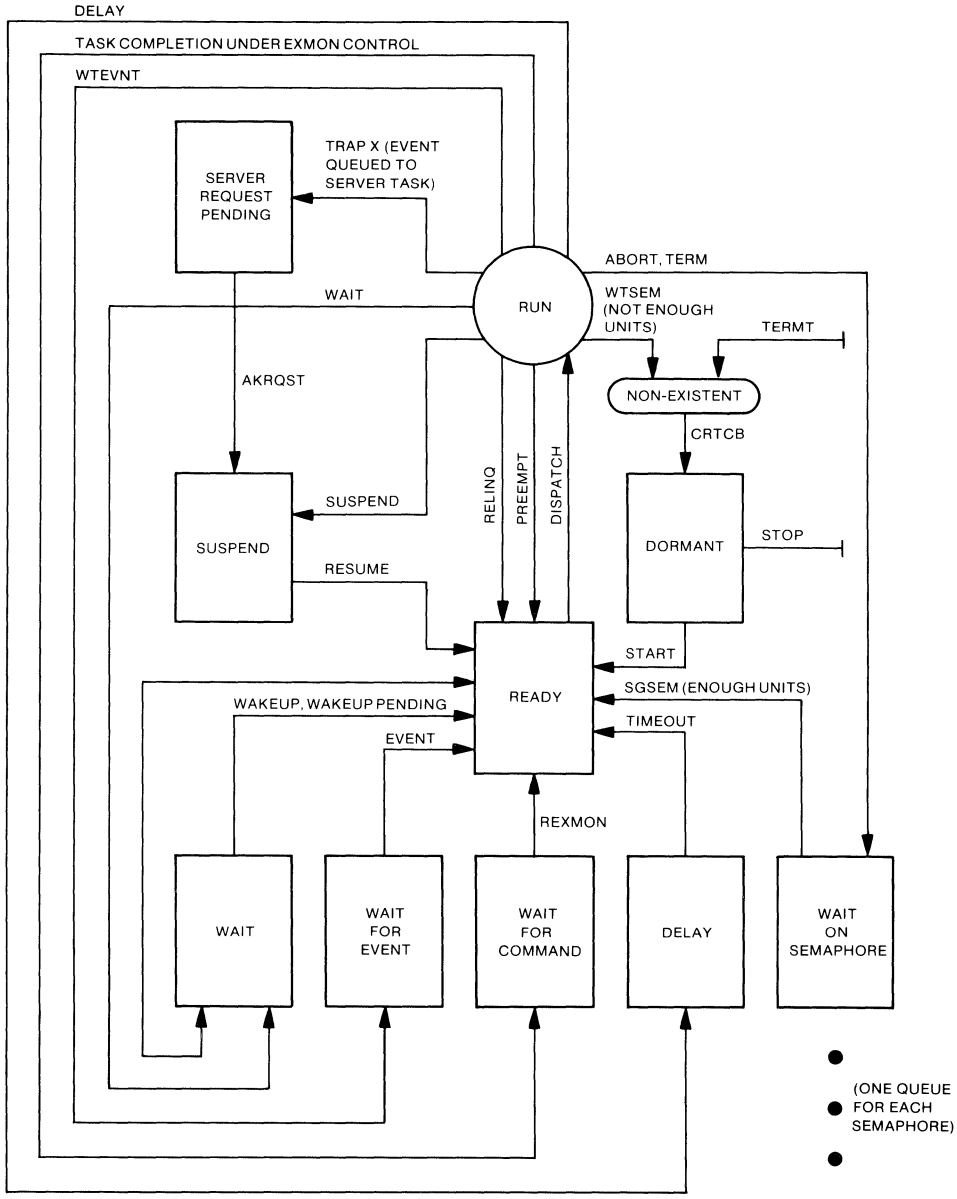
GTSEG	Allocate a new program code or data memory segment to the target task by placing it in the task's address space
DESEG	Delete a program code or code memory segment from the target task's address space
GTASQ	Allocate an asynchronous service queue (ASQ) to the target task
DEASQ	Deallocate the requesting task's ASQ
DCLSHR	Make a memory segment available for shared access
ATTSEG	Place an existing memory segment (shareable) into the requesting task's address space
SHRSEG	Place an existing memory segment (shareable) into another task's address space
TRSEG	Remove a memory segment from the requesting task's address space and place it in the address space of another task
RCVSA	Return a description of the specified segment to the requesting task

2. Task State Control

Directives are provided to control task execution by means of moving a task through various states A task can control its own state transitions or another task's state transitions Following is a list and brief description of each of the directives in the task state control category

CRTCB	Create a task control block (TCB) for a new task and place the task in the DORMANT state
START	Move the target task to the READY state
TERMT	Terminate task by releasing resources and deleting the TCB of the target task so that the task no longer exists in the system
STOP	Move the target task to the DORMANT state
RESUME	Move the target task from the SUSPEND state to the READY state
WAKEUP	Move the target task from the WAIT state to the READY state
ABORT	Initiate a task's own abnormal termination Its TCB is deleted and the task no longer exists in the system
TERM	Initiate a task's own normal termination Its TCB is deleted and the task no longer exists in the system
SUSPND	A task moves itself into the SUSPEND state
WAIT	A task moves itself into the WAIT state
RELINQ	A task moves itself to the READY state so that RTEX can perform a dispatch cycle
SETASQ	A task changes the status of its ASQ and/or ASR

FIGURE 2 — RMS68K TASK STATE TRANSITION DIAGRAM



2. Task State Control (continued)

DELAY — A task moves itself to the DELAY state for a specified period of time.

DELAY W — A task moves itself into the DELAY state At the end of a specified period, it moves itself into the WAIT-FOR-EVENT state until an event is placed in its ASQ

WTEVNT — A task moves itself into the WAIT-FOR-EVENT state until an event is placed in its ASQ

SETPRI — A task changes its own priority or the priority of another task to the specified value

RQSTPA — A task requests the timed periodic activation of itself or another task

EXPVCT — A task announces the handling of its own exceptions

TRPVCT — A task announces the handling of its own trap instructions

3. Task Event Control

These directives can be used by a task for event queueing and event servicing

QEVNT — An event is placed into the ASQ of the target task

RDEVNT — The next event in the requesting task's ASQ is moved into the task's receiving area

RTEVNT — Return to the point of interruption of a task upon completion of that task's ASR processing

4. Task Synchronization Directives

Tasks can use semaphores to synchronize activities and control resources Several task synchronization directives, which tasks can use to control these semaphores, are available

ATSEM — A task attaches to a semaphore and acquires use of that semaphore The semaphore is created and given an initial signal count if it does not already exist

CRSEM — A task creates a new semaphore or resets the initial count of an existing semaphore

DESEM — A task releases use of a semaphore

DESEMA — A task releases use of all semaphores to which it is attached

WTSEM — A task requests and, if necessary, waits for semaphore-controlled access or execution to be granted

SGSEM — A task signals release of a semaphore-controlled resource or execution.

5. Special Function Task Control

The directives in this category are used to control server tasks and exception monitor tasks

Server Task Control

A server task is able to receive and process requests from any task in the system A user task issues a request to a server task by executing a TRAP instruction, thus, a server task appears to function as part of VERSAdos

SERVER — A task establishes itself as a server task

AKRQST — A server task acknowledges receipt or 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 Task Control

An exception monitor task provides execution control over a target task or emergency processing when a task fails Exception events which will cause a target task to stop execution and notification to be sent to the target task's exception monitor task are specified in an exception monitor mask associated with the target task An exception monitor task can also control the operation of a target task in a trace mode

EXMON — A target task becomes associated with an exception monitor task

DEXMON — A target task detaches from an exception monitor task

EXMMSK — Events of interest to an exception monitor task are specified

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

REXMON — A target task executes as directed by an exception monitor task

6. Interrupt Service Management

A task can include one or more routines which are activated as the result of an external interrupt (or exception), and execute in the MC68000 user hardware state at the priority level of the interrupt Such a routine is called an interrupt service routine (ISR) This mechanism is useful in creating I/O device drivers.

CISR — A task configures one of its ISR's to respond to a particular exception.

SINT — A task simulates an exception (interrupt).

RTE — An ISR returns from execution.

MISCELLANEOUS RMS68K FUNCTIONS

Some miscellaneous RMS68K functions which do not fit into the preceding task management categories are described below.

1. Date and Time Directives

STDTIM A system task sets the system date and time

GTDTIM A task obtains the current system date and time

2. Task or System Information Directives

These directives are used to retrieve some types of information about other tasks or about system execution

TSKATTR A task receives the user number and attributes of a target task

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

TSKINFO A system task requests a copy of a target task's Task Control Block

SNAPTRAC The contents of the system trace table are copied into a buffer in the calling task's address space

3. System Enhancement Directive

CDIR This directive provides a means by which new executive directives can be created for use by specific applications

I/O DEVICE CHANNEL MANAGEMENT

The Channel Management Request handler (CMR) routines represent level 5 of the RMS68K Executive structure 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 requester commands and command service routines

To facilitate the orderly management of device I/O and interrupts from devices, three control parameters are used. hardware interrupt vector number, hardware priority level and software priority number At boot up, a communications channel is reserved for each device consisting of a contiguous portion of memory mapped I/O space Each channel is fully described by a Channel Control Block (CCB) into which is placed control parameter values for the channel Also, for each hardware interrupt vector number, all CCB's having the same vector number are linked to that vector by the creation of a vector chain in which the CCB with the highest software priority is linked first, the CCB with the second highest priority linked second, and so on

When an interrupt occurs, control is passed through the first CCB (via a JSR) to the CMR interrupt handler which performs a minimum state save, retrieves the CCB address, and calls the appropriate I/O handler. If the I/O handler returns without claiming the interrupt, CMR calls the handler of the next CCB chained to that vector number This continues until the chain is exhausted

Supervisor/Subordinate Channels

Because some I/O devices can control more than one peripheral (e g , a dual serial controller) and, further, because each of these controlled peripherals causes interrupts of the same hardware priority level, a means is required for the system to communicate with each peripheral and of identifying the peripheral which caused the interrupt

The RMS68K method provides for allocation of a supervisor channel for the device, of a subordinate channel for each controlled peripheral and for linking all subordinate channels to the supervisor via the CCB field — CCBSUB Only the supervisor CCB is linked by vector chain to the hardware interrupt vector number

The supervisor channel (code) is written for the corresponding I/O device and is capable of searching the linked list of subordinate channels for an interrupt source In traversing the interrupt vector chain, when the CMR handler encounters a supervisor CCB, control is passed to that channel which then identifies the interrupting peripheral and passes control to the corresponding subordinate channel for interrupt service If no peripheral in the list caused the interrupt, control is returned to the CMR handler Figure 3 illustrates a hardware interrupt vector number chain and a linked supervisor CCB/subordinate CCB list

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

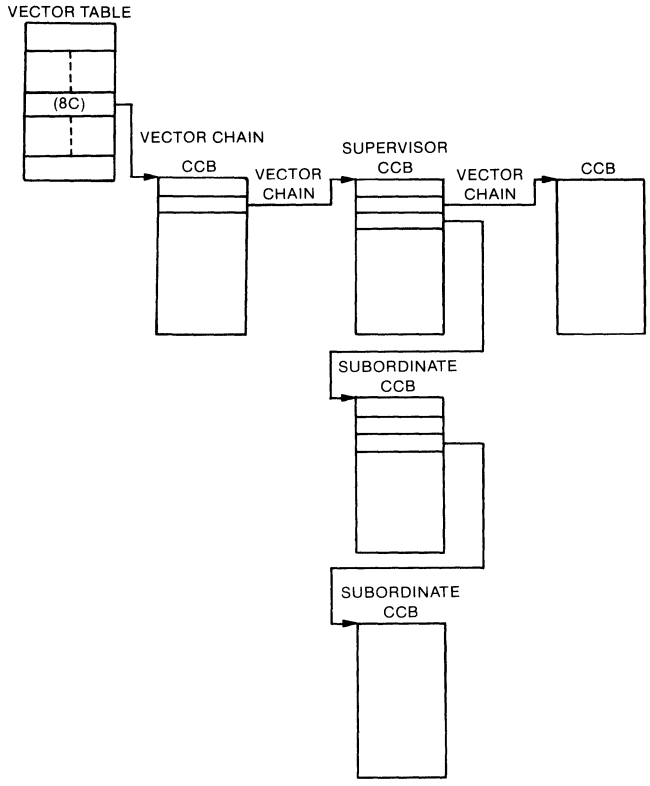
DELETE Remove a channel definition from the system

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

FIGURE 3 — REPRESENTATION OF SUPERVISOR/SUBORDINATE CHAIN



CMR Handler (continued)

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

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 rejected 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

This Processor is made up of a series of modules linked together to efficiently execute user defined commands It allows the user to access a pre-established list of system commands, or to easily build additional commands into the system by adding the ASCII name of the command in the Command List (Cmdlist) Module and providing a pointer to the processing routine in the Command Control Module

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

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

Ordering Information

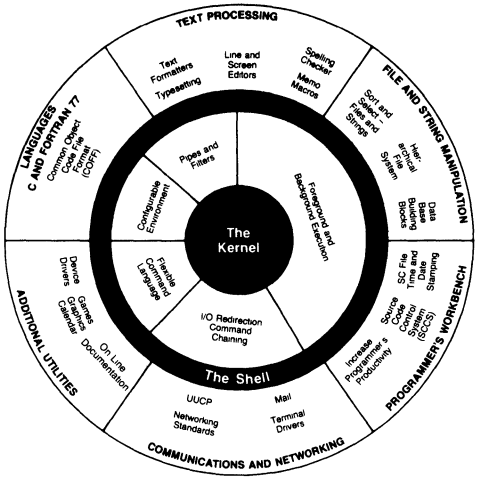
Part Number	Description
	VERSAdos Operating System, the full-feature, real-time, multitasking, multiuser disk operating system for MC68000-based products such as VERSAmodule and VME-module Monoboard Microcomputers and other hardware, including the required memory, I/O, timer function (optional) and a mass storage (disk) subsystem Drivers for VMmodules, VMEmodules, and I/Omodules, utilities, and system generation are also included
M68K0VDOST	VERSAdos source on 14" CMD cartridge
M68VKMSVRDOS	VERSAdos source on 25 Mbyte Lark cartridge
M68K0VDOSH	VERSAdos object on 14" CMD cartridge, bootable on EXORmacs
MVMC682-1SWC	VERSAdos object on 14" CMD cartridge, VMC 68/2 bootable
M68VKMBVERDOS	VERSAdos on 25 Mbyte Lark cartridge, VMC 68/2 bootable
MVMC682-1SWL	VERSAdos on 8 Mbyte Lark cartridge, VMC 68/2 bootable
M68K0VDOS	VERSAdos on 8" diskettes, bootable on EXORmacs, VMC, VM21, VM02, VME110, VME101
M68VKXBVERDOS	VERSAdos on 5¼" diskettes, bootable on VM03, VME110, VME101, VME/10
	RMS68K Real-Time Multitasking executive for MC68000-based products such as VERSAmodule and VMEmodule Monoboard Microcomputers and other hardware, including the required memory and timing function (optional)
M68K0RMS68KT	RMS68K source on 14" CMD cartridge
M68VKMSRM68K	RMS68K source on 24 Mbyte Lark cartridge
M68VKXSRM68K	RMS68K source on 5¼" diskettes
M68K0RMS68KH	RMS68K object on 14" CMD cartridge
M68K0RMS68K	RMS68K object on 8" diskettes
M68VKXBRMS68K	RMS68K object on 5¼" diskettes

Related Documentation

Part Number	Description
M68KVOVER/D3	VERSAdos Overview Manual
M68KVSF/D4	M68000 Family VERSAdos Systems Facilities Reference Manual
M68KRMS68K/D5	MC68000 Real-Time Multitasking Software User's Guide
MVMEVDOS/D1	VERSAdos to VME Hardware and Software Configuration User's Manual
RMS68KIO/D6	VERSAdos I/O and File Management Services Manual

SYSTEM V/68 Operating System

- The Standard **UNIX** derived Operating System for the M68000 Microprocessor Family
  - — The only AT&T Validated UNIX System V port for the MC68000*
  - — Small, Flexible Kernel with Performance Optimized for the M68000 Family
  - — Command Interpreter ("Bourne Shell") Offers Powerful Facilities For Interactive Control. Shell can be replaced by a user tailored command interpreter.
  - — Extensive Set of Programming Languages (C, FORTRAN 77, BASIC and a SNOBOL)
  - — Latest version of C language with flexnames
  - — Extensive set of program development tools cxref, lint, sdb
  - — Text Processing Tools (vi, ex, nroff, troff)
  - — Electronic Mail
  - — Communications and Networking Support
  - — Programmer's Workbench
- Language Support for the MC68000, MC68010, and MC68008 Microprocessors. (Future support for the MC68020 & MC68881)
- Support for Motorola Memory Management Device, MC68451; a future release will have support for the Paged Memory Management Unit, MC68851
- Executes on Motorola Development Systems, EXORmacs, VME/10
- Source Code Available On Motorola Development Systems and Non-Motorola Systems Capable of Reading UNIX cpio Format Media (AT&T UNIX System V/M68000 Version Source Code License Required)



- A total of over 400 utilities
- Files in Common Object File Format, COFF

TABLE 1 — Common Object File Format (coff)

FILE HEADER
OPTIONAL AOUT HEADER
SECTION HEADERS
TEXT SECTION
INITIALIZED DATA SECTION
UN-INITIALIZED DATA SECTION
RELOCATION INFORMATION
LINE NUMBER INFORMATION
SYMBOL TABLE
STRING TABLE (FLEXNAMES)

INTRODUCTION

SYSTEM V/68 is derived from UNIX System V/M68000, a jointly developed product of AT&T Technologies, Inc. and Motorola Inc.

The SYSTEM V/68 Operating System is the standard UNIX derived Operating System for the M68000 family of microprocessors. It offers a small compact 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.

THE KERNEL

The kernel supports a Multitasking Multiuser software development and/or OEM application execution environment. The kernel size is typically 120 K bytes, but for target configurations may be reduced to approximately 80 K bytes depending upon the system features required. SYSTEM V/68 supports a layered or hierarchical file system of arbitrary size. File blocks are 1024 bytes long and a hashed i-node look up feature makes for faster file access; (refer to the more detailed description under FILE SYSTEM). The kernel is of the swapping variety. Processes may share memory segments through the optionally configurable shared memory support.

The kernel features multiple memory management unit support for the MC68451 MMU chip, which is configurable using the config utility. Support for the MC68451 is supplied for the VME/10 system only. The MMU descriptors are demand loaded. The retargetable MMU interface provides for ease of porting. This MC68451 MMU support is for the MC68010 microprocessor only.

The SYSTEM V/68 kernel employs an aged process priority scheduler.

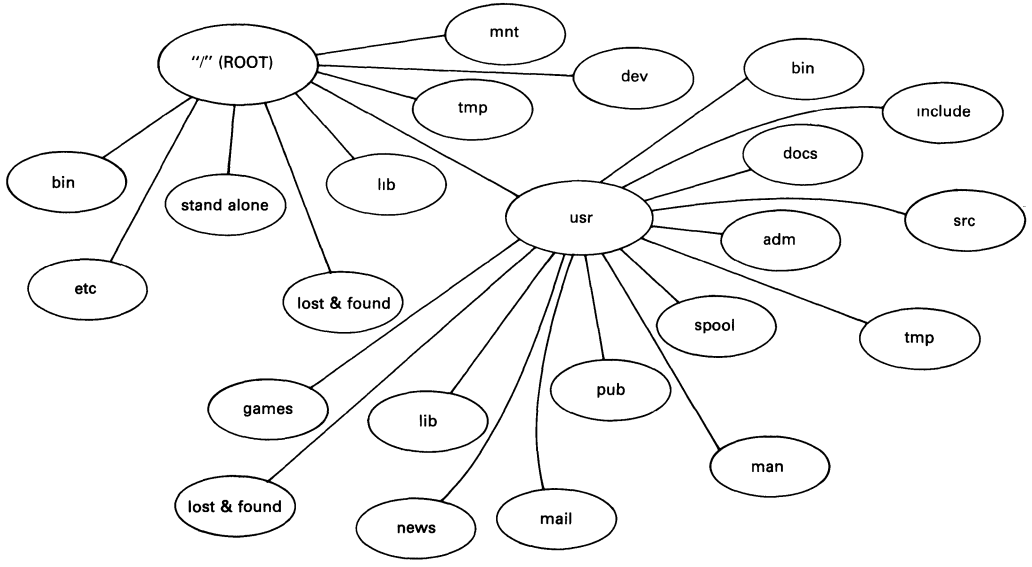
Kernel support is provided for both the MC68000 and MC68010 microprocessors. Support for the MC68020 32-bit microprocessor is under development

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 system of arbitrary size (see figure 1). Major features of the file system are:

- 1024 byte file block size.
- Simple and coherent naming conventions; a name can be absolute or relative to any directory in the file system hierarchy.
- File linking across directory boundaries.

FIGURE 1 — SYSTEM V/68 File Structure As Shipped



- Automatic file space allocation and de-allocation.
- Mountable and unmountable file systems.
- Hashed i-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 dynamically alterable.
- Facilities are provided for creating, moving, accessing and processing groups or single files and directories.

LANGUAGES

Provided as an integral part of SYSTEM V/68 is the C language compiler "cc." The C language has established itself as one of the most popular commercially supported programming languages, and is used most frequently to produce portable application software.

The SYSTEM V/68 OS 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 "efl."

An M68000 assembler "as," and a linker/loader "ld" complements the C and FORTRAN language compilers.

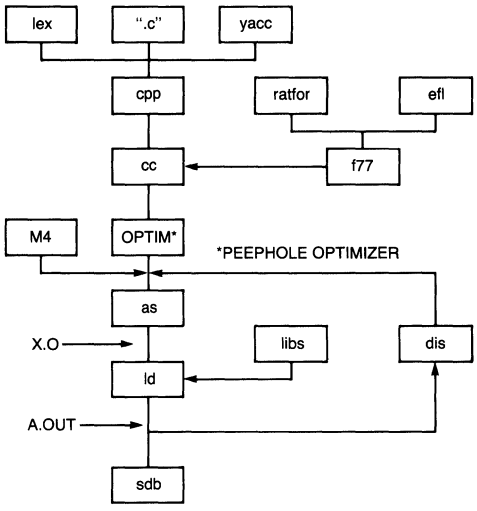
The following list displays the commitment that the SYSTEM V/68 OS provides to the programming function (see figure 2):

- cc is the C language compiler which has been derived from pcc2, the Portable C Compiler version 2. This compiler is based on the Software Generation System (SGS) technology and supports the Common Object File Format (COFF).
- f77 is the FORTRAN 77 language compiler.
- as is the M68000/M68010 assembler.
- ld is the linker/loader and supports COFF.
- sno is a SNOBOL language compiler/interpreter for string manipulation.
- bs is a "BASIC" — like compiler/interpreter for modest size programs.
- awk is a pattern scanning and processing language.
- bc is an arbitrary precision desk calculator language.
- yacc is a compiler compiler.
- lex is a lexical analysis language.
- sdb is a symbolic debugger which can be used on C source code, FORTRAN, and assembler code modules.

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. Such 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 attempts to build a graph charting the external references.

FIGURE 2 — SYSTEM V/68 Language Inter Play Chart



- 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 allows you to dump 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 attempts to build a cross reference table.
- strip is a tool that strips all the symbol table and line numbering information from common object files and archives.

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 the Source Code Control System (SCCS), which provides the facility to store, update, and retrieve all versions, past and present, of source code modules.

The following utilities make up this set of software tools:

- admin is used to create and administer SCCS files, and to change parameters of existing files.
- delta is used to make a delta (change) to an SCCS file.
- cdc is a tool that changes the delta commentary of an SCCS delta.
- comb combines SCCS deltas by generating a shell procedure which when run will reconstruct the given SCCS files.
- get is a utility used to get a copy of a specific version of an SCCS file.
- prs is a tool that will print all or part of an SCCS file.
- rmdel is the complement tool to delta in that it removes a delta (change) from an SCCS file.
- sact is a tool that informs the user of any impending deltas to an SCCS file.
- sccsdiff is a tool used to show up the differences between two SCCS files.
- unget is the complement tool to get, in that it undoes a previous get of an SCCS file.
- val is a tool to validate an SCCS file.
- vc is a tool to affect version control of a file.
- what is a tool used to identify SCCS files.

TEXT PROCESSING

Support for text processing under the SYSTEM V/68 OS is provided by the inclusion of five editors: vi, ed, ex, bfs, and sed. Ex and vi are full screen editors, ed is a line editor, and sed is a non-interactive stream oriented editor. Bfs is a big file scanner and is similar to ed.

There are two text formatting utilities included: nroff for printing on typewriter-like devices, and troff for a Wang Laboratories phototypesetter. 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, including the Motorola EXORterm 155, through the use of the terminal database "termcap." Additional entries to termcap can be easily added by the user to support new terminals.

Support for Text Processing under the SYSTEM V/68 OS includes the following tools and utilities:

- nroff is a tool that formats text suitable for printing on a terminal or printer.
- troff is a text formatting tool suitable for a Wang Laboratories, Inc.,/C/A/T phototypesetter.
- vi is a display oriented text editor based on the underlying line editor ex. It is possible to use the command mode of ex from within vi and vice-versa.
- ex is the root of a family of editors which includes edit, ex, and vi. Ex is a line oriented editor which is a super-set of ed.
- view is a read only version of ex/vi.
- tbl is a preprocessor for formatting tables for nroff, and troff.
- cw and checkcw are two tools used to prepare constant width text for troff.
- eqn, neqn and checkeqn are tools used to format mathematics text for nroff and troff.
- mm, osdd and checkmm are tools to print/check documents formatted with mm macros.
- mmt and mvt are tools used to typeset documents, viewgraphs, and slides.
- tc is a phototypesetter simulator for the Tektronix 4014 terminal.
- deroff is a tool that removes nroff/troff, tbl, and eqn constructs.
- ptx is the tool used to generate permuted indexes.
- spell is used to check a document for spelling errors.
- edit is a variant of the text editor ex and is recommended for new or casual users.
- ed is the standard text editor.
- sed is a stream editor that may be given editing instructions from a file.
- bfs is the big file scanner and is similar to ed except that it is read only.

COMMUNICATIONS

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. Networking support allows several computers to be linked together, either through dedicated links or by dial-up connections. With these facilities Motorola development system users can communicate with one another and with other UNIX systems. Available tools/utilities 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.

In SYSTEM V/68 interprocess communications (IPC) functionality has been added; including shared memory, messages, and semaphores.

GRAPHICS

SYSTEM V/68 provides several graphic support tools. Due to hardware restrictions, graphics support is limited to the Motorola VME/10 desktop computer system only.

- gdev is a set of device routines and filters that enable a user to display Graphical Primitive String, GPS, files, generated by stat and toc (see below). These files can be displayed on a Hewlett-Packard 7221A graphics plotter, a Tektronix 4010 series storage terminal, and a VME/10 system.
- gutil is a collection of device independent graphical utility commands. Capabilities supported in gutil include:
  - — brief online documentation for each SYSTEM V/68 graphic utility.
  - — the ability to dump a GPS file in a human readable format.
  - — the ability to dump a plot(4) file in a human readable format.
  - — the ability to transform a GPS file into plot(4) commands which can be displayed by plot filters.
  - — the ability to transform plot(4) commands into GPS.
- stat contains commands that manipulate and plot numerical data. The statistical network, stat(1G), is a collection of routines that can be interconnected using the SYSTEM V/68 shell to implement numerical processing networks. Included in stat are routines to generate simple statistics and GPS files to be displayed using the gdev utilities.
- toc contains commands that generate graphical tables of contents in the form of an hierarchy chart depicting all subdirectories of a user-chosen directory.

HARDWARE CONFIGURATIONS

VME/10 Host

The recommended minimum configuration of the VME/10 to support SYSTEM V/68 is:

- 15M byte or 40M byte Winchester fixed disk. Dependent upon application.
- 640K bytes RAM (384K on board RAM plus one MVME201 256K byte RAM board).

This configuration is provided by part number M68K102C1 plus one additional memory board MVME201.

Additional 256K byte blocks of memory can be provided with the addition of extra MVME201's. Ethernet LAN support can be supplied by adding MVME330 boards. Additional serial ports can be supported by the addition of MVME400 boards. Dual parallel ports can be integrated by the addition of MVME410 boards, which support a centronics parallel printer interface.

EXORmacs Host

The basic EXORmacs configuration to support SYSTEM V/68 requires 768K bytes of RAM and 96Mbytes of disk space (80M bytes fixed and 16M bytes removable). This configuration is provided for by ordering the following parts:

M68KMACS	MC68000 EXORmacs Development System.
M68KVM11-1	256K VERSAbus RAM module.
M68HDE96-1	EXORmacs 96M byte Hard Disk.

Additional memory can be added by ordering the following part numbers:

M68KVM11-1	256K bytes of dynamic RAM.
M68KVM11-2	512K bytes of dynamic RAM.
M68KVM12	1024K bytes of RAM.

Ethernet LAN support is provided by part number M68KVM33's. Additional serial port support is provided by the addition of M68KV7's (Four channel full-duplex communication controller).

NETWORKING SUPPORT

Ethernet Local Area Networking (LAN) support can be optionally provided by SYSTEM V/68 with the addition of either M68KVM33 VERSAmodule board or MVME330 VMEmodule boards together with their associated software. This option is provided at additional cost.

The software is designed to interconnect VME/10 and EXORmacs systems via the Ethernet using the XEROX NETWORK SYSTEM (XNS) protocol package. 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 be easily upgraded to other protocols, each module corresponds to a specific layer of the OSI interconnection model. Device drivers that run under both Motorola SYSTEM V/68 and VERSAdos operating systems are available, thus allowing systems running under both operating systems to be networked together over the Ethernet LAN. The following applications packages are provided with the LAN software:

- 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 to a remote host and execute commands and programs on that computer as though directly connected to it.
- Electronic Message: users can send and receive messages across the network.
- 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 of sent/received information. From this and hardware-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.

THE PAL PORT

As an additional language product offering, Motorola's Pascal compiler, macro assembler, and linkage editor have been ported to run under the SYSTEM V/68 OS. The assembler and linkage editor are referred to as the PAL assembler and the PAL linker so as to distinguish them from the common assembler (as) and the common linker (ld) available as part of the SYSTEM V/68 product. The PAL Port product will not run under VERSAdos.

The PAL Port provides an environment for software development similar to that provided under VERSAdos. Existing software source code, written in Pascal or M68000 assembly language and previously used on VERSAdos, can be transported to SYSTEM V/68, compiled or assembled, linked and executed. Load modules can be generated that can run under either VERSAdos or SYSTEM V/68.

The PAL Port package is available at additional cost, and contains the following software:

- The M68000/010 Pascal compiler.
- The M68000/010 PAL structured macro assembler.
- The M68000 PAL linkage editor.
- Run Time Libraries to support SYSTEM V/68 on the EXORmacs and VME/10 systems; VERSAdos with MMU; VERSAdos without an MMU; RMS68K (with BIOS); and VERSAbug on M68KVM01.

THE VERSAdos TOOL KIT

The VERSAdos TOOL KIT is a set of software tools. This tool kit has been designed to be particularly useful in the development of VERSAdos target software in a SYSTEM V/68 host environment. This option is available at additional cost.

- cvtu is a tool to convert the file format of SYSTEM V/68 ".o" files to VERSAdos ".RO" file format.
- cvtv is a tool to convert the file format of VERSAdos ".RO" files to SYSTEM V/68 ".o" file format.
- migru is a tool to read a file from a VERSAdos disk and creates a SYSTEM V/68 file.
- sysgenv is a cross SYSGEN utility that will create a VERSAdos/RMS68K operating system on a SYSTEM V/68 host. (Requires the PAL port for assembler and linker)
- ubuilds is a tool that builds an S-record file from a SYSTEM V/68 ".o" file, that file can then be downloaded to a target board/system.
- vbuilds is a tool to convert a VERSAdos ".LO" load module file to an S-Record file.
- vdir is a tool to provide a directory listing of a VERSAdos diskette.
- connect is a tool to allow an HDS-200 or an HDS-400 user to use a VME/10 SYSTEM V/68 as a host.
- pmesg is a generalized message print routine which allows the user to alter messages without recompiling pages that use them.

In later releases of the "Tool Kit" support for the following tools/utilities will be added:

- ppu is a utility to program proms.
- migrv is a tool to read a SYSTEM V/68 file and create a VERSAdos file.
- initv is a tool to format and initialize a VERSAdos disk under SYSTEM V/68.
- loads is a tool to allow a SYSTEM V/68 user to receive S-records from or send S-records to an HDS200 and/or HDS400.

SYSTEM V/68 TECHNICAL TRAINING

For those users not familiar with SYSTEM V/68 or other UNIX derived operating systems, and for those who require additional training, Motorola Technical Training offers a four day training course; MTT18.

The course teaches the student how to use the Motorola UNIX derived operating system, SYSTEM V/68, and the C compiler. As a prerequisite, familiarity with the M68000 family of microprocessors and a knowledge of programming is recommended. A course outline follows:

Introduction
- Schedule
- Purpose

General Concepts
- Overview
- Multitasking
- Operating System Structure
- Conclusion

Command Set
- Overview
- General Commands
- Pipes
- Development Commands
  - Editor
  - Assembler
  - Linkage Editor
- Conclusion

Shells
- Overview
- Concepts
- Passing Arguments
- Conclusion

File System Structure

System calls
- Overview
- C compiler
- Calls
  - SYSTEM
  - System Interface
  - Subroutines
  - Library Calls

Conclusion

The course features hands on training on Motorola development systems. For more information and for schedules of course availability call your local Motorola Semiconductor Sales offices or

Motorola Technical Training,
Attn: Ray Doskocil, Mail Drop HW68,
P.O. Box 2953, Phoenix, AZ, 85062.
(602) 244-7126 or 244-4945

SYSTEM V/68 Ordering Information

PART NUMBER	DESCRIPTION
M68NNCBSV[1]	SYSTEM V/68 Operating System OBJECT CODE ONLY software supplied on CDC 16M byte removable CMD disk cartridges for use on EXORmacs System (68000/EXORmacs MMU). This product contains the following components: • two CMD disk cartridges containing all the SYSTEM V/68 object code files for the kernel, shell, languages, tools, and utilities • One full set of SYSTEM V/68 documentation in three volumes • customer letter • customer's copy of the executed End-user License agreement for SYSTEM V/68 Operating System software. Object code is supplied as bootable load modules The kernel modules are also supplied as relocatable, partitioned, and relinkable modules so that the OEM can RECONFIGURE the SYSTEM V/68 Operating System without purchasing source code.
M68NNXBSV10[1]	Same as above except on 5 1/4" diskettes for use on a VME/10 System (68010/68451)
M68NNCSSV[2]	SYSTEM V/68 Operating System software SOURCE AND OBJECT code supplied on CDC 16M-byte removable CMD cartridges for use with the EXORmacs system (68000) • three CMD disk cartridges containing all the SYSTEM V/68 source and object code files for the kernel, shell, languages, tools, and utilities • One full set of SYSTEM V/68 documentation in three volumes • customer letter • customer's copy of the executed Source Code License agreement for SYSTEM V/68 Operating System software
M68NNXSSV10[3]	Same as above except supplied on 5 1/4" diskettes for use with the VME/10 system (68010/68451).
M68NNQSSV[2]	Same as M68NNCSSV except supplied on one 9 track, 1600 bpi, cpio format magnetic tape suitable for use on a DEC VAX or other system capable of reading this format.
M68NNQSSV10[3]	Same as M68NNXSSV10 except supplied on one 9 track, 1600 bpi, cpio format magnetic tape suitable for use on a DEC VAX or other system capable of reading this format

1 MOTOROLA SYSTEM V/68 object software license required
2 AT&T Technologies UNIX System V/M68000 Version source license required
3 AT&T Technologies UNIX System V/M68010 Version source license required

Networking Products

PART NUMBER	DESCRIPTION
M68NNXBVMELAN	Ethernet object software supplied on 5.25" diskette for use on VME/10 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 MVME330 BIV. • Software documentation, and user manual Object code is supplied as bootable load modules, and relinkable object modules, so that the OEM can reconfigure without need of source code.
MVME330	VMEmodule Ethernet LAN Controller. This module provides high performance, intelligent single board connection of VMEbus systems to Ethernet, a Local Area Network. Includes 128K RAM, LANCE(7990), SIA(7991), MC68000 MPU, Kernel Firmware and power up self-test.
MVME330-UX	MVME330 plus the M68NNXBVMELAN software on 5 25" diskettes.
M68NNHBVMELAN	Same as M68NNXBVMELAN except on 8" diskette for use on the M68KVM33 VERSAmodule board for EXORmacs.
M68KVM33	Same as MVME330 except a VERSAmodule format board.
M68KVM33-UX	M68KVM33 plus the M68NNHBVMELAN software on 8" diskettes.

See the MVME330 and M68KVM33 board data sheets for more detail on compatible Ethernet LAN product

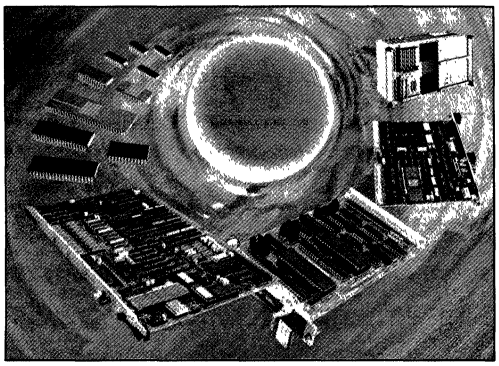
All ETHERNET LAN software requires a license from Motorola

PAL Port Product

PART NUMBER	DESCRIPTION
M68NNCBPASMLK	Object code release of the PAL PORT on CMD media for use on EXORmacs and KVM03 systems. The product contains the following items: • Pascal compiler object code modules, PASCAL, POPTIM, PASCAL2. • PAL Structured macro assembler. • PAL linkage editor. • Pascal, assembler, linker documentation. • Customer letter.
M68NNXBPASMLK	Object release similar to M68NNCBPASMLK except packaged on 5.25" diskettes for use on VME/10 systems.
M68NNCSPASMLK	Source release of M68NNCBPASMLK on CMD media for use on EXORmacs and VM03 systems.
M68NNQSPASMLK	Source release of M68NNCBPASMLK on 1600 bpi cpio format 9 track Magnetic Tape suitable for use on a DEC VAX running UNIX System V.
M68NNXSPASMLK	Source release similar to M68NNCSPASMLK except packaged on 5.25" diskettes for use on VME/10 systems.

VERSAdos Tool Kit Product

PART NUMBER	DESCRIPTION
M68NNCBTLKT	Object code release of the VERSAdos TOOL KIT on CMD media for use on EXORmacs and VM03 systems. Product contains the following items: • Object code modules for each tool, sysgenv; ubuilds, cutv; cvtu; migrv • VERSAdos TOOL KIT documentation. • Customer letter
M68NNCSTLKT	Same as M68NNCBTLKT but including source code release on CMD media for use on EXORmacs and VM03 systems.
M68NNQSTLKT	Same as M68NNCSTLKT but released on 1600 bpi cpio format magnetic tape, suitable for use on a DEC VAX.
M68NNXBTLKT	Same as M68NNCBTLKT except released on 5.25" diskette media for use on a VME/10 system.
M68NNXSTLKT	Same as M68NNXBTLKT except source code on 5.25" diskette media for use on a VME/10.
M68NNFBTLKT	Same as M68NNCBTLKT except object code release on 8" diskette media for use on EXORmacs and VM03 systems.



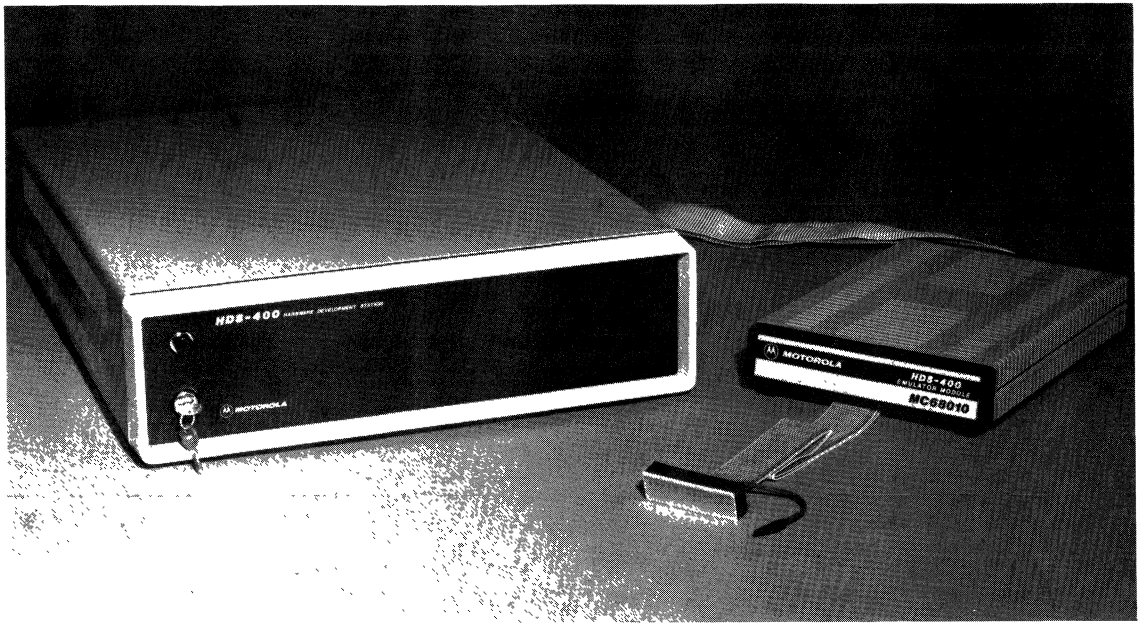
While it's not absolutely necessary, developing a microprocessor-based system is about 99% easier using a computer based on the same microprocessor family. Motorola's 16/32-bit development systems are just that: computers based on the MC68000 but computers especially designed to speed development of hardware and software for an application system.

Both the VME/10 Microcomputer System and the EXORmacs Development System can host either the UNIX-derived SYSTEM V/68 Operating System or the VERSAdos Real-Time Multitasking Operating System. These hosts then provide powerful environments which facilitate the initial development of application software and the parallel development of system hardware.

The development, diagnostic and debugging capabilities of a host environment can be greatly increased through the use of two compatible development tools: the HDS-400 Microprocessor Hardware/Software Development Station and the Real-Time Bus State Analyzer. By performing as a fully functional substitute for the application system microcomputer or microprocessor throughout the development cycle, the HDS-400 facilitates testing of the integrated hardware and software providing quick, easy accommodation of design changes and correction of program bugs. The Bus State Analyzer brings additional diagnostic power to the development process by allowing at various points in the system, the simultaneous sampling and storage of events on each bus line for later analysis.

Whatever the combination of elements, the resulting development system will provide powerful support for the design and development of an application system based on Motorola's lines of board-level, 16/32-bit microcomputer system components.

Development Support Section



HDS-400 MICROPROCESSOR HARDWARE/ SOFTWARE DEVELOPMENT STATION

The HDS-400 Microprocessor Hardware/Software Development Station emulates the Motorola M68000 family of microprocessors (MPU's) including the MC68000, MC68008 and the MC68010 The HDS-400 and its computer host provide tools for debugging and for analyzing microprocessor system performance — both hardware and software.

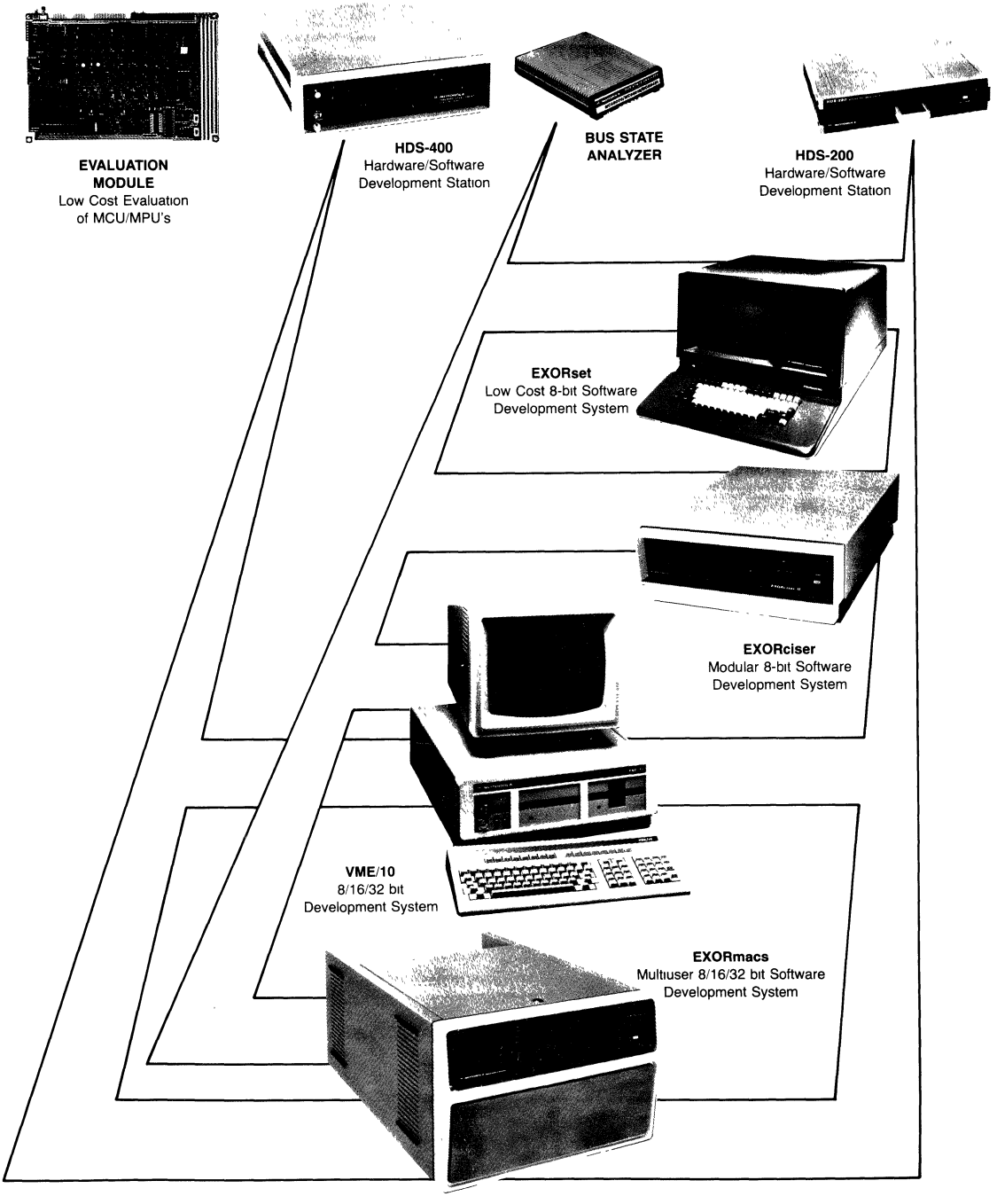
The microprocessor is the nerve center for microcomputer systems; and, by observing conditions at critical times and locations within the microprocessor, the efficiency of system debugging is greatly enhanced. Since a microprocessor is a solid piece of processed silicon, it is difficult to physically observe conditions within the microprocessor itself. Emulators provide the facility through which these conditions can be observed.

An emulator is a circuit that exactly duplicates the functions and performance of the microprocessor. By substituting for the microprocessor in the system being tested, it allows the designer access into the memory, CPU and registers of the microprocessor for debugging purposes The HDS-400 provides this emulation as well as the facilities and tools for actually making the analysis of system performance

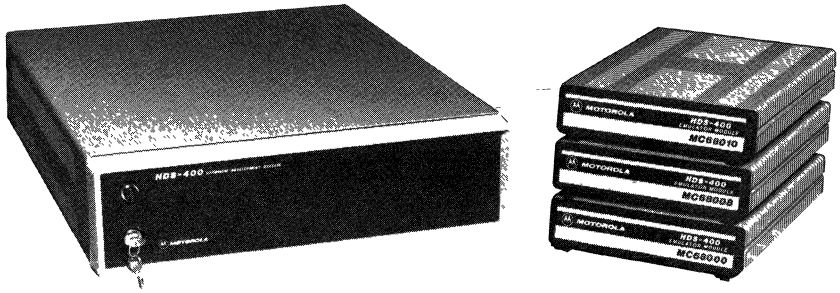
The optional Real-Time Bus State Analyzer may be integrated into the HDS-400 to significantly enhance the features and functionality of the Hardware/Software Development Station. BSA Personality modules are available for the MC68000, MC68008 and MC68010 to match the emulator support offered with the HDS-400. The BSA probe is inserted directly into the target MPU socket. Either the emulator probe or the actual MPU chip may then be inserted into the target through the BSA probe. Thus, the BSA can function alone, or simultaneously with the HDS-400 emulator.

While the HDS-400 provides development support specifically for the MC68000, MC68008 and MC68010, 8 and 16-bit MPU's, other development stations are available for all Motorola microprocessors and microcomputers.

MOTOROLA FAMILY OF MICROCOMPUTER DEVELOPMENT AIDS



MOTOROLA 16/32-BIT MICROCOMPUTER SYSTEM COMPONENTS



HDS-400 MICROPROCESSOR HARDWARE/SOFTWARE DEVELOPMENT STATION

5

DESIGN FEATURES

- 12.5 MHz Real-Time Emulation for MC68000 MPUs
- 10 MHz Real-Time Emulation with no Wait Cycles for MC68000 MPUs
- 8 0 MHz Emulation for MC68008 and MC68010 MPUs
- No User Target System Restrictions
- 32K bytes of 10 MHz No Wait Cycle Emulation RAM is Standard
- Emulation RAM Expandable to 64K, 128K or 256K bytes
- Full Symbolic Debug with EXORmacs and VME/10 Hosts
- Unrestricted User Memory Map
- One-Line Assembler/Disassembler
- Automatic Self-Test of Development Station Hardware
- M68KHDS400 Interfaces with EXORmacs Development System
- M68KHDS400A Interfaces with Motorola VME/10 and DEC VAX Hosts
- Compatible with Real-Time Bus State Analyzer

MAJOR BENEFITS

- Reduces Development Costs
- Shortens Product Development Cycle
- Brings Product to Market Faster
- Versatility Protects Against Obsolescence

The HDS-400 Microprocessor Hardware/Software Development Station, in conjunction with a Motorola EXORmacs Development System or VME/10 Microcomputer System, or a DEC VAX Computer, provides a complete hardware/software development system for the Motorola M68000 family of microprocessors. It consists of a Control Station, with all the support circuitry for complete MPU emulation, and a separate Emulator Module with an internal microprocessor to match the particular MPU it is expected to emulate.

Two key capabilities of the HDS-400 make it very useful as a systems development tool. The first is the ability to serve as a fully functional substitute for the selected MPU in the user's target system By plugging the HDS-400 into the socket on the prototype hardware, it allows efficient testing and debugging of both hardware and software. The second capability is the rapid debug and integration of the target system for the production of prototypes This is accomplished by the use of the powerful set of commands in the HDS-400. The user may execute the commands by either entering the command code and its parameters, or by sequentially depressing function keys which provide a "fill-in-the-blanks" format with parameters such as file name, address, data, etc. When a single function key or a combination of function keys is pressed, a command code is automatically generated and the command syntax is displayed by the system.

SYSTEMS DEVELOPMENT AND INTEGRATION

The initial stages of developing an MPU-based system normally involve two parallel, and yet rather independent, efforts. One is the hardware design — the other the software design. These design efforts are frequently accomplished by two different teams of personnel, resulting in debugging problems that are often difficult and time-consuming.

The Hardware/Software Development Station simplifies the system debugging by bringing the hardware and software development processes into intimate relationship with each other throughout the development cycle. Moreover, with the HDS-400 it becomes economically feasible to test alternate design approaches in order to determine the best solution.

HARDWARE DEVELOPMENT

"Target hardware" refers to the target system microprocessor, all memory (both RAM and ROM), and I/O devices which interface to the MPU. Hardware development involves the architectural design and debugging of the RAM, ROM and the I/O interfacing to the MPU.

Obtaining sample program ROM's with the precise pattern needed for optimum system performance can be expensive and time consuming because each memory IC must be individually programmed and tested. Since the complete development process may require several iterations of either ROM programming or masked ROM's, each with its attendant time delay and/or mask expense, the cost in time and money can become not only unpredictable but sometimes prohibitive.

The HDS-400 has been designed to solve these problems by emulating the target MPU until the target configuration has been fully proven. This capability allows the software and hardware to be tested at an early stage of the development process. The firmware can be quickly changed, and the results may be easily tested. Printed circuit board checkout, debugging and modification are also completed during hardware development and integration with the system software. Once this is done, the HDS-400 will allow a complete evaluation of the target system.

SOFTWARE DEVELOPMENT

The HDS-400 Hardware/Software Development Station starts bringing its advantages into use at a very early stage of the system development The emulator can be organized and software run prior to availability of hardware. This allows an early start on the debugging process. As hardware changes occur, software updating and debugging are readily accomplished When system prototype hardware development is completed, user programs can be thoroughly checked out and modified as required. This allows early determination of the final mask ROM configuration Original system algorithms and I/O parameters can be tested before completing the hardware or software design. After this phase is completed, a source program is prepared using the host's editing facilities (EXORmacs, VME/10, or DEC VAX).

The source program is compiled or assembled and the software modules are linked together by the host linker/loader to create the object software. The host translates the object software into modular "S" record format and downloads the program through the HDS-400 into target memory. The emulator is now ready to act as a substitute MPU in the target system. The "S" record serial transmission format is widely used and highly reliable The format is shown on page 19.

DEVELOPMENT SYSTEM IMPLEMENTATION

The HDS-400 Hardware/Software Development Station has been partitioned with options and part numbers that give the user versatility in defining the development system configuration The user may choose from three host computers EXORmacs, VME/10, or DEC VAX with a variety of operating systems. Each of the HDS-400 Control Stations is delivered prewired to accept the optional Emulation Memory Module and the Real-Time Bus State Analyzer (BSA) The EXORterm 155 is required in HDS-400 systems hosted by the EXORmacs and the VAX. The VME/10 functions as both host and terminal to the HDS-400, eliminating the need for a separate terminal in VME/10-based systems.

Figures 1, 2 and 3 show functional block diagrams of HDS-400 systems with EXORmacs, VME/10 and VAX hosts respectively. All three system configurations show how the optional Emulation Memory Module and the BSA are interconnected.

The EXORmacs-based system shown in Figure 1 uses the M68KHDS400 version of the HDS-400 Control Station Remote Serial Conversion Modules are installed in the I/O card cage of the Control Station, and in the EXORmacs chassis as part of the Data Link Controller (M68KHDSDLC). These two modules are key to implementing the synchronous 56K-Baud Host Link between the EXORmacs and the HDS-400 This link allows one EXORmacs host to support up to four HDS-400 Control Stations and Emulator Modules. These Control Stations can be connected to the host computer with up to 3900 feet of cable. The block diagram also shows the interconnections between the HDS-400, the Emulation Memory Module, the BSA, and the EXORterm 155 Display Console.

FIGURE 1 — EXORmacs TO HDS-400 HARDWARE DEVELOPMENT SYSTEM FUNCTIONAL BLOCK DIAGRAM

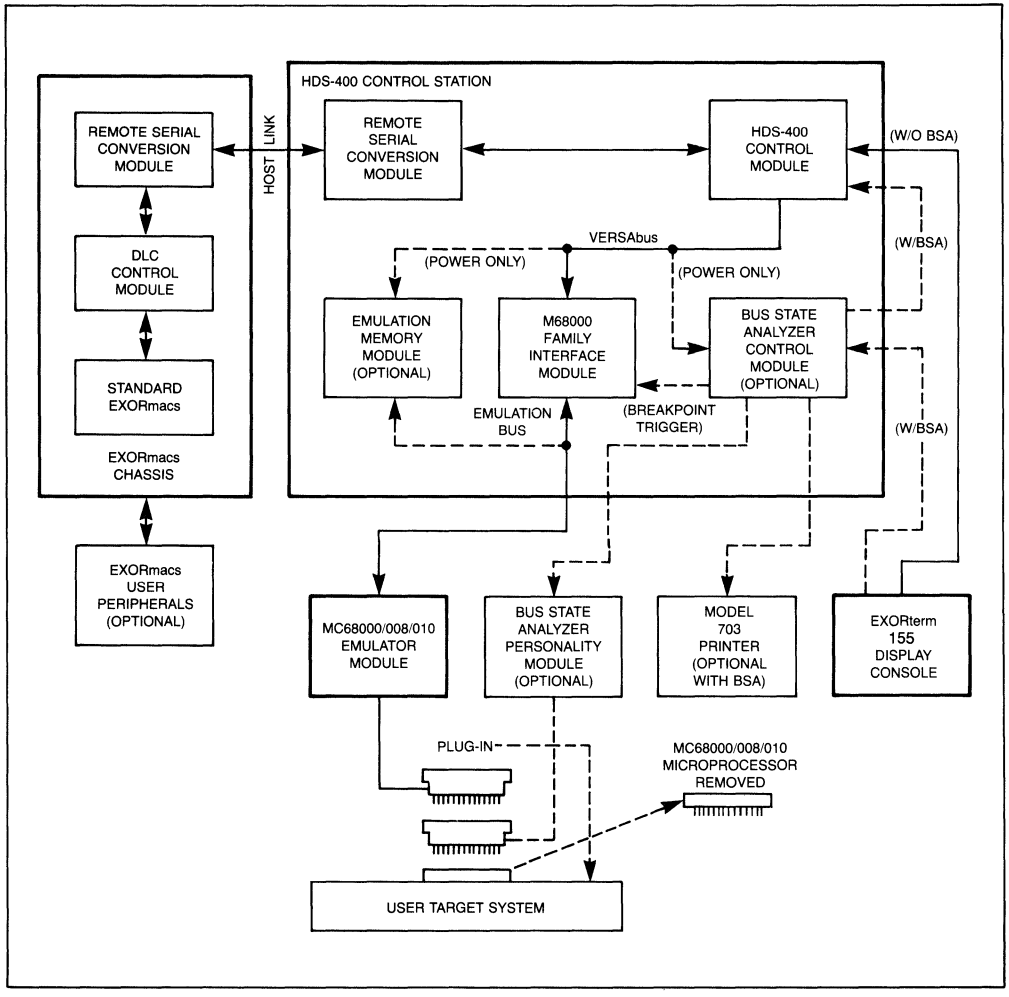
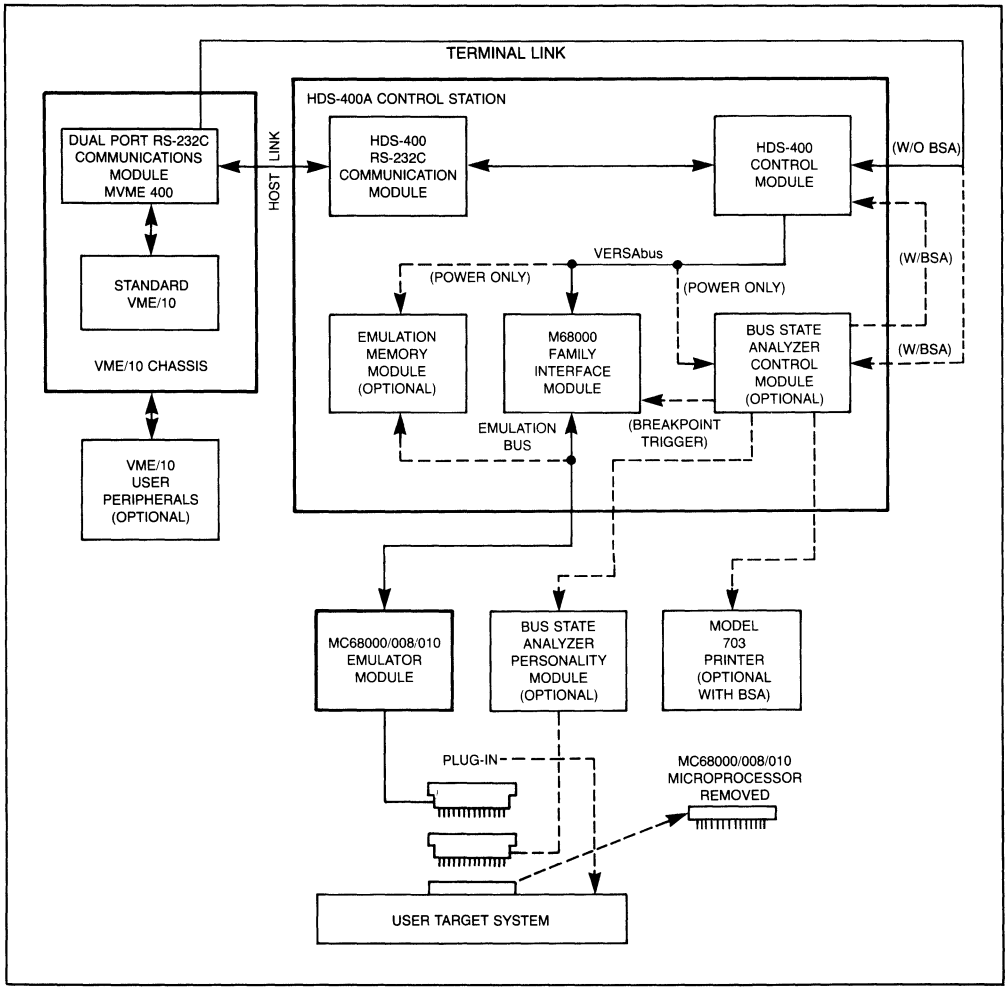


Figure 2 shows the VME/10-based system connected to the HDS-400A Control Station (M68KHDS400A). This version of the HDS-400 Control Station has an HDS-400 RS-232C Communication Module installed in its I/O card cage rather than the Remote Serial Conversion Module. The VME/10 has a Dual Port RS-232C Communications Module (MVME400) installed in its I/O card cage. The MVME400 provides the Host and Terminal Links from the VME/10 to the HDS-400 Hardware/Software Development Station. This block diagram shows the interconnections between the HDS-400A, the Emulation Memory, the BSA and the VME/10.

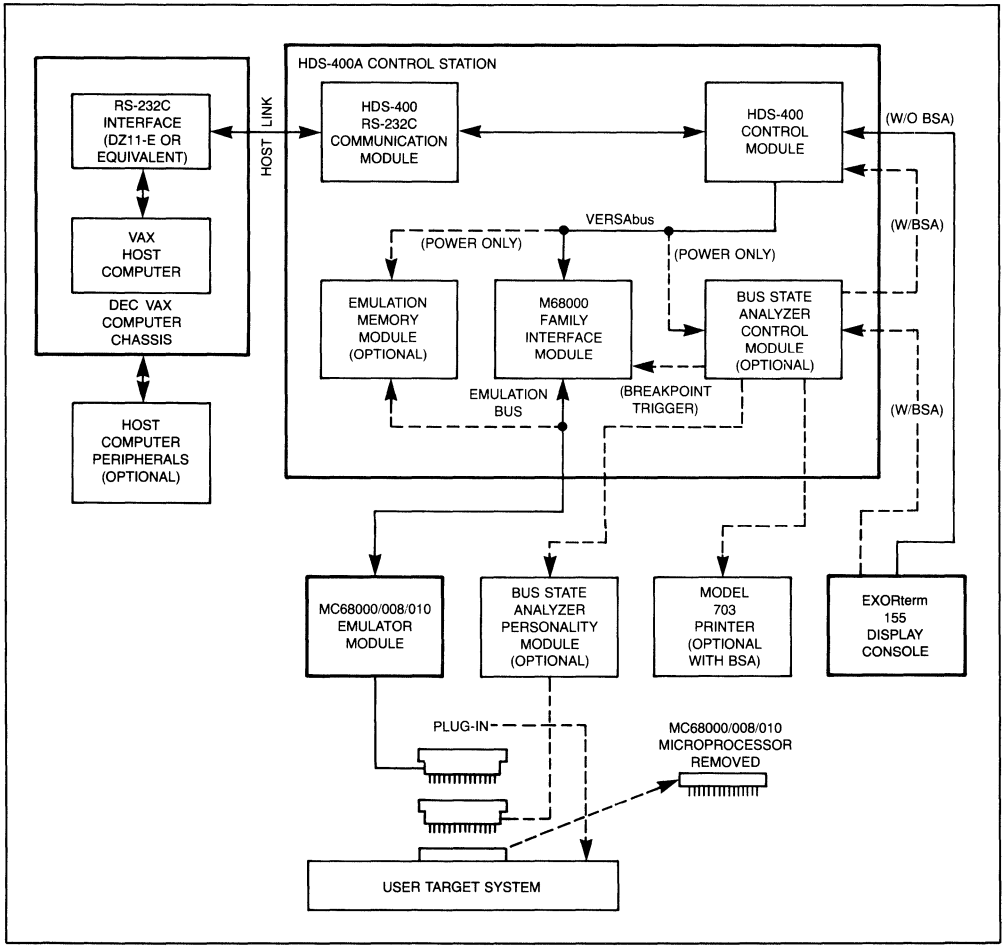
FIGURE 2 — VME/10 TO HDS-400 HARDWARE DEVELOPMENT SYSTEM FUNCTIONAL BLOCK DIAGRAM



The DEC VAX-based system is shown in Figure 3. This system uses the HDS-400A Control Station (M68KHDS400A) and requires an EXORterm 155 Display Console. The Host Link is implemented between the HDS-400 RS-232C Communication Module in the HDS-400A Control Station and an asynchronous RS-232C interface, DZ11-E or equivalent, in the VAX. Figure 3 also shows how the BSA and Emulation Memory are connected in the VAX-based system.

The Motorola-supplied software for VAX hosts includes the HDS-400 control and self-test programs to be downloaded from the VAX, host-resident file servers and HELP messages, plus VAX software installation instructions. The customer will have to provide appropriate cross assemblers and linkers which generate target program load modules in the Motorola "S" record format. These are commonly available from third-party companies, and are listed in the "Motorola Microprocessor Software Catalog."

FIGURE 3 — DEC VAX TO HDS-400 HARDWARE DEVELOPMENT SYSTEM FUNCTIONAL BLOCK DIAGRAM



THE HARDWARE/SOFTWARE DEVELOPMENT STATION CONFIGURATION

The basic HDS-400 Hardware/Software Development Station is illustrated in the functional block diagram shown in Figure 4 It consists of two separate enclosures — the HDS-400 Control Station and the Emulator Module. The MPU Family Interface Module is mounted in a VERSAmodule card slot within the HDS-400 Control Station.

HDS-400 CONTROL STATION

The Control Station contains a four-slot VERSAmodule chassis with a built-in 228 W switching power supply, and a separate I/O card cage. The I/O card cage is used to house the terminal and host interface link cards. Two of the four VERSAmodule card slots in the Control Station are used for the HDS-400 Control Module and the MPU Family Interface Module. The other two card slots are available to accommodate the Real-Time Bus State Analyzer (BSA) Control Module, and the Emulation Memory Module

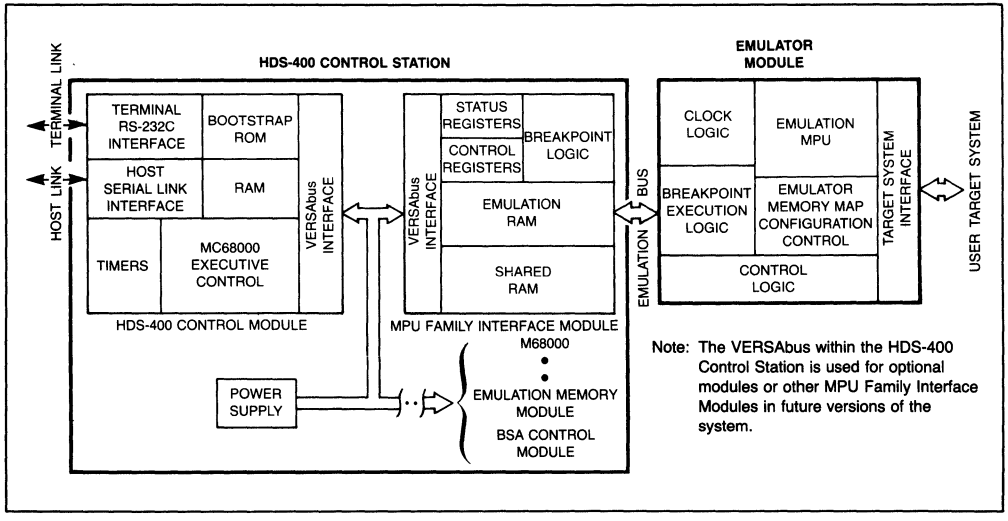
Two versions of the HDS-400 Control Station are available. The M68KHDS400 is configured to interface to an EXORmacs host over a 56K baud synchronous RS-422 serial link. The M68KHDS400A interfaces to a Motorola VME/10 host through a 19 2K baud asynchronous RS-232C serial link, or to a DEC VAX host over a 9600 baud asynchronous RS-232C link

The Control Module within the HDS-400 Control Station contains an MC68000 MPU with ROM, timers, plus a VERSAbus interface. The MC68000 runs the monitor, controls the host and terminal interface links, and interfaces with the Emulator Module through the Family Interface Module. Although the MPU Family Interface Module is located in the HDS-400 Control Station, it is purchased under a separate part number.

MPU FAMILY INTERFACE MODULE

The Family Interface Module functions with the Emulator Module specified in the HDS-400 system. The M68000 Family Module has 32K bytes of 10 MHz no wait cycle emulation RAM which the designer can map in 4K byte blocks to replace or expand the target system's RAM. This 32K bytes of emulation RAM may be expanded to 64K, 128K or 256K bytes with the addition of an optional Emulation Memory Module. These modules are discussed in a later section This HDS-400 RAM can also be used to replace the target system's ROM, and may be fully protected against inadvertent writing The MPU's entire 16 Mbyte address space, including all vectors, is available to the designer The Family Module also contains breakpoint logic which supports up to sixteen program breakpoints, one complex event breakpoint (when the optional BSA is used), one timeout breakpoint, and one breakpoint on a write to protected emulation RAM

FIGURE 4 — HDS-400 FUNCTIONAL BLOCK DIAGRAM



THE EMULATOR MODULE

The Emulator Module's output to the user target system is by a noisefree controlled impedance cable which terminates in a plug to mate with the target hardware MPU socket. Controlled impedance cables also connect the Emulator Module to the Family Interface Module in the HDS-400 Control Station. The emulator contains the emulation MPU, memory map configuration control, and several logic functions including breakpoint execution logic

Emulators are available for the MC68000, MC68008 and MC68010 MPU's The HDS-400 system provides 12.5 MHz real-time and 10 MHz no wait cycle emulation of the MC68000, and 8.0 MHz emulation of the MC68008 and MC68010

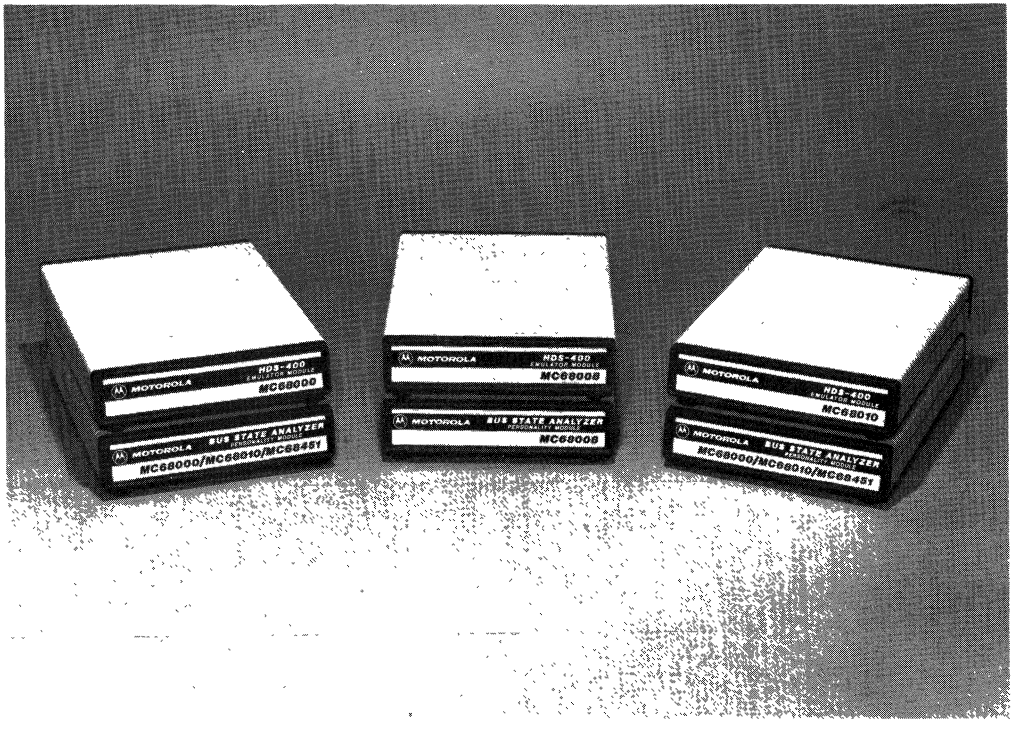
OPTIONAL EMULATION MEMORY MODULE

The standard 32K bytes of emulation RAM provided in the Family Interface Module may be expanded with one of three optional Emulation Memory Modules The three memory expansion modules available increase the 10 MHz no wait cycle emulation RAM to 64K, 128K or 256K bytes. The Emulation Memory Module is installed in an extra VERSAmodule card slot in the HDS-400 Control Station, and receives its power from the VERSAbus. The memory module is connected to the Family Interface Module through the special Emulator Module cable provided with the emulator.

OPTIONAL BSA

The addition of the Real-Time Bus State Analyzer (BSA) greatly enhances the features and functionality of the HDS-400 Hardware/Software Development Station The BSA Control Module is installed in a VERSAmodule card slot in the Control Station A Personality Module which matches the MPU being emulated is selected and connected to the BSA Control Module The BSA probe may be used with either the emulator probe, or with the actual target system MPU in the hardware socket Functionality of the HDS-400 with and without the BSA is described in the following sections.

FULL EMULATION AND BSA SUPPORT FOR MC68000, MC68008, AND MC68010



The HDS-400 Hardware/Software Development Station, when substituted for the MPU chip in the target system being debugged, performs the functions of the microprocessor being emulated — exactly as the MPU would have performed were it still in the circuit being tested. The emulator provides the interfacing with the RAM, ROM, and I/O devices and operates at the same speed as the MPU. There are no restrictions on the use of emulation memory that are not imposed by the MPU itself, and the memory may be mapped to the target system or to the emulator module.

To demonstrate the versatility and effectiveness of the HDS-400, a few of the system's development capabilities are described in the following paragraphs.

HELP DISPLAY

A complete command set listing for the HDS-400 is provided on pages 14, 15, 16, 17 and 18. It is not necessary, however, to continually refer to this printed table, because the command set listing can be displayed on the screen of the terminal simply by calling for HELP. The command "HELP" causes a brief command syntax summary for all commands to be displayed. And . should you require more detailed information regarding a selected command, typing "HELP" followed by the command code brings up a more detailed summary of the specific command (Figure 5). If the code for a desired command function is unknown, it can be displayed and entered by pressing the descriptive Function Keys after typing "HELP".

TRANSPARENT MODE OPERATION

The TM command allows the user to communicate with the host, the HDS-400, and the optional BSA using a single EXORterm 155 in EXORmacs and DEC VAX-based systems. The Transparent Mode is also used in VME/10-based HDS-400 systems, although a separate terminal is not required.

EMULATOR CONFIGURATION

The user may selectively enable/disable hardware signals between the emulator and the target system (Figure 6). The user can save the emulator configuration setup on the host computer, to be recalled for a later debug session. This eliminates the need to redefine the emulator memory map and the enable/disable status of the emulator signals at each debug session.

EMULATOR MEMORY MAP CONFIGURATION

The total emulation memory, up to 256K bytes with an optional Emulation Memory Module, may be mapped in 4K byte blocks to overlay or expand the target system's RAM (Figure 6). The emulation memory may also be write-protected to simulate the target system ROM. The MPU's entire 16 Mbyte (1.0 Mbyte for MC68008) address space, including all vectors, is available to the emulator without restrictions.

MEMORY MANIPULATION

The contents of all memory, or selected portions of memory, can be displayed or modified via a powerful set of commands. Mass block memory moves, data updates, and memory scans (to locate a specific data pattern) are among the many operations available. The Memory Display Command and Memory Modify Command can display and modify the specified memory contents in either disassembled form, or in both hexadecimal and ASCII form. Support of target systems with protection or memory management de-

5

FIGURE 5 — COMMAND HELP DISPLAY

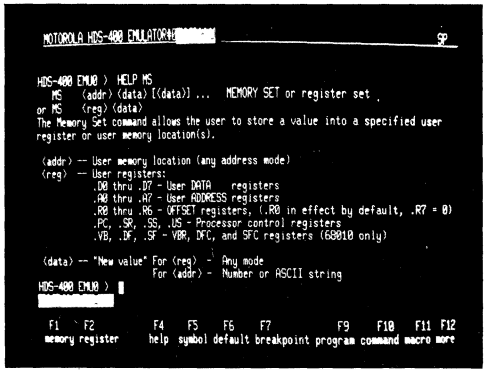
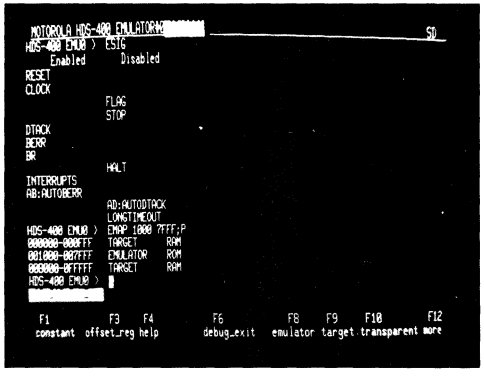


FIGURE 6 — EMULATOR SIGNAL AND MEMORY MAP CONFIGURATION



vices is accomplished by specifying the memory space to be accessed as Supervisor Program, Supervisor Data, User Program or User Data. The memory manipulation commands may be used with assembler symbols when the HDS-400 has a Motorola host (EXORmacs or VME/10).

REGISTER MANIPULATION

The contents of the emulator MPU registers may be displayed or modified. The user can store a value in a specified register location, or can display the contents of MPU registers and all eight offset registers. The offset registers may be used to simplify debugging of relocatable code. The register manipulations may also be accomplished using assembler symbols when the HDS-400 is used with a Motorola host.

5

LOAD AND SAVE PROGRAMS

Target programs assembled on the host computer may be downloaded via the HDS-400 into the target or emulation memory. The object code is transmitted in "S" record format with a checksum performed on each record to verify the data as it is received.

One to seven areas of target memory may be saved in a single file on the host disk in "S" record format. The saved memory can later be loaded, in the same "S" record format, from the host to the target or emulation memory

BREAKPOINT DEFINITION AND DELETION

One of the most powerful debugging tools is the ability to insert temporary breakpoints at selected locations in the program — usually at the ends of loops or after selected system operational steps The HDS-400 supports up to sixteen program breakpoints, within four address ranges, in the user's target program.

Program Breakpoints in the HDS-400 cause the processor to stop *prior* to executing the breakpoint instruction. The HDS-400 emulators are not "fooled" into breaking on instruction prefetches . . . an important improvement over other emulators. Because of the unique architecture in the HDS-400, program breakpoints may be set in either RAM or ROM, since a breakpoint does not cause any memory locations to be altered.

The breakpoints may be displayed, deleted, and can be entered as absolute addresses, local symbols, or assembler symbols (Figure 7). Assembler symbols are available in HDS-400 systems used with Motorola hosts. Emulation may also be stopped when the target bus times out, or when an attached optional BSA triggers. The BSA can provide complex breakpoints on other than program instructions This enhanced functionality is described in the section entitled "Additional Features with BSA."

PROGRAM EXECUTION CONTROL

The target program execution may be started in a free-running mode or at a specified starting address. If no address is specified, the program will start executing at the current program counter address. The execution of the target program can be stopped from the display console

Tracing is an effective and commonly used debugging aid. Trace commands cause instructions to be executed, one at a time, between any selected program steps. After each instruction is executed, the next instruction is disassembled, allowing analysis and modifications to be made as required (Figure 8).

SYMBOLIC ASSEMBLY AND DISASSEMBLY

Assembly language greatly speeds program development by substituting easily remembered mne-

FIGURE 7 — BREAKPOINT DEFINITION

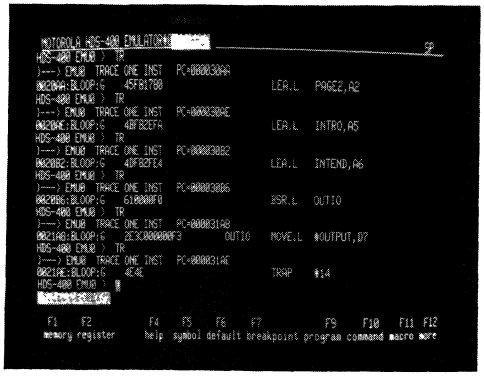
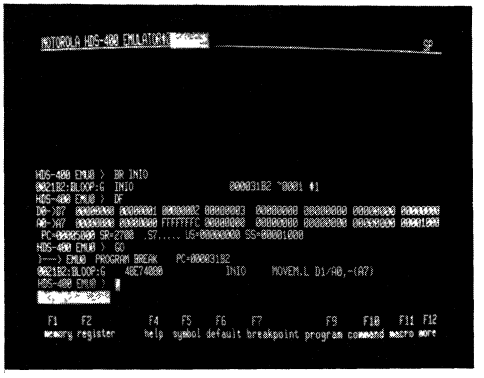


FIGURE 8 — INSTRUCTION TRACE DISPLAY



monic labels for low-level machine code instructions. Developing any assembly language program is an iterative process Many changes must be made to the program during product development to correct errors, compact code, or modify code to accommodate new specifications. Relocatable labels and symbols are much more convenient and efficient to use in developing and debugging code than absolute addressing and data values.

The HDS-400 provides a single-line assembler/ disassembler to expedite changes and additions to program code. HDSdebug is a resident program which generates and maintains information (assembler symbol names, module names, and section numbers) about the program being debugged. HDSdebug automatically equates this symbolic information to absolute addresses for the user (Figure 9). As discussed above, Memory Manipulation, Register Manipulation, and Breakpoint features may be used with assembler symbols when the HDS-400 has a Motorola host (EXORmacs or VME/10)

When used with non-Motorola hosts, assembler symbols are not available. However, local symbols may be defined at the HDS-400 and saved on the host. These local symbols may also be downloaded from the host disk to the HDS-400, and then displayed for use in expediting the debugging process

COMMAND MACROS

A macro is a sequence of commands that can automatically be executed when the macro name is entered as a command Macros allow complex custom commands to be designed by the user. The user may define the macro using any number of standard HDS-400 debugger commands Several standard debug commands allow the user to define, display, edit and delete macros In addition, all defined macros may be saved on the host disk, and at a later time be loaded to the HDS-400 from the host.

ADDITIONAL FEATURES WITH BSA

The HDS-400 Hardware Development Station was designed to function with the Real-Time Bus State Analyzer (BSA). System functionality is significantly enhanced when the optional BSA is installed in the HDS-400 Control Station. The BSA probe is inserted directly into the user's target MPU socket Either the Emulator Module probe or the actual MPU chip may then be inserted into the target through the BSA probe Thus, the BSA can function alone, or simultaneously with the HDS-400 emulator.

The BSA connects to all signals which appear at the target system MPU socket Accesses to emulation memory do not appear at the target MPU socket, and are not visible to the BSA. The BSA also connects to six optional signals and one external clock through ball clip probes

The following paragraphs describe the added HDS-400 functionality and features provided with the BSA. All of the features described are provided in the BSA alone except the complex breakpoints.

COMPLEX BREAKPOINTS

The standard HDS-400 allows implementation of Program Breakpoints without the addition of the BSA. Program Breakpoints allow the user to set a breakpoint on a program statement, causing the microprocessor to stop *prior* to executing the designated instruction These breakpoints were discussed earlier in the "Breakpoint Definition and Deletion" section.

The addition of the BSA to the HDS-400 allows the implementation of another powerful type of breakpoint . . . an event breakpoint. These Complex Breakpoints may be specified by a combination or sequence of events defined in terms of address, data, control, status, and external signals. This allows the user to define

FIGURE 9 — SYMBOLIC DISASSEMBLY DISPLAY

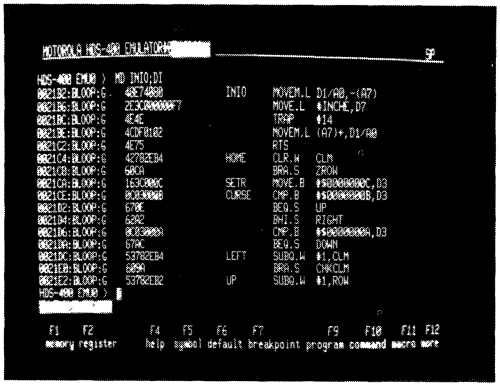
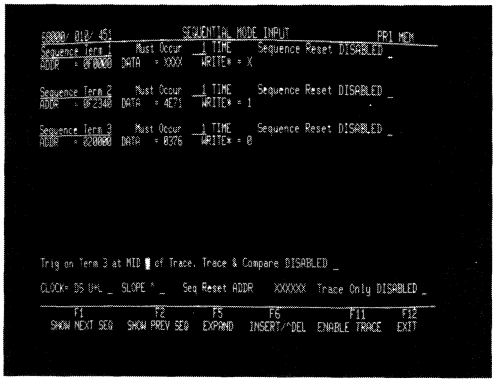


FIGURE 10 — BSA COMPLEX BREAKPOINT SETUP



breakpoints generated by very complex situations which stop the microprocessor after the specified event occurs.

The following examples indicate some of the Complex Breakpoints which can be implemented when the BSA is added to the HDS-400 system (Figure 10). The examples have been arranged in a progressively more detailed sequence. The HDS-400/BSA combination allows the user to define a breakpoint:

- when the program has left a specified memory area,
- on a write to a specific address location,
- when a specific interrupt vector is fetched,
- when interrupt request level 4 is presented to the MPU,
- when address location $2000 has been addressed 10 times, but only after the program has executed a subroutine located at address $F0124 and written data equal to $77 to location $1030,

  .
  .
  .

- or when a condition specified by up to seven events from 79 qualifier lines has occurred.

REAL-TIME TRACE

The Bus State Analyzer has a Sequential Trigger Mode which gathers, or stops gathering, real-time trace data after a specified series of events occur. Sequence Terms, as these events are called, must occur in the specified order or triggering will not take place. A Sequence Reset Term can also be defined which will reset the BSA and cause the instrument to begin looking for the entire set of Sequence Terms again. This real-time trace capability is most useful for debugging complex software including loops, nested subroutines and complex branches.

PROGRAM EXECUTION TIMING

The time necessary to execute a program or subroutine may be measured using the Sequential Trigger Mode. The beginning and ending instructions of the program or subroutine can be defined as Sequence Terms. The BSA will then display the time which elapses between these two trigger points.

WINDOW TRIGGER MODE

The Window Trigger Mode in the BSA provides a means of causing signal states to be stored if address accesses occur inside *or* outside of a particular address range. Both the upper and lower bounds of the range are programmable, and the size is variable from a single address to the full range of the memory map (Figure 11). Window triggering is useful for following programs that suddenly and unexpectedly leave the memory area in which they should be operating. It is also applicable for observing access violations in a multiple user environment.

PERFORMANCE ANALYSIS

The BSA can also provide Performance Histogram displays for analysis of software and hardware system performance. Software (Address) Histograms give an indication of the relative frequency of memory accesses within a particular memory address range, with the exact range specified by the user (Figure 12). This histogram provides a means of determining where a program spends the greatest amount of time The resulting information can then be used to optimize inefficient code. A Hardware (Discrete) Histogram is also provided which displays the relative frequency of combinations of four of the optional user-selected signals within the target system.

FIGURE 11 — BSA WINDOW TRIGGER MODE

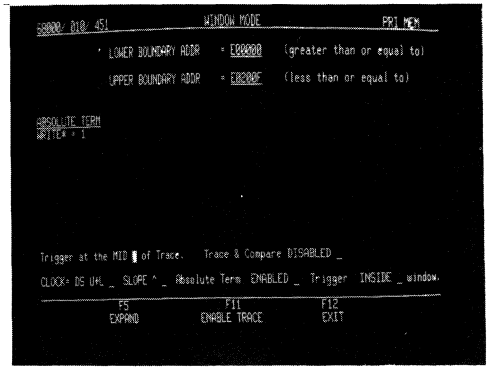
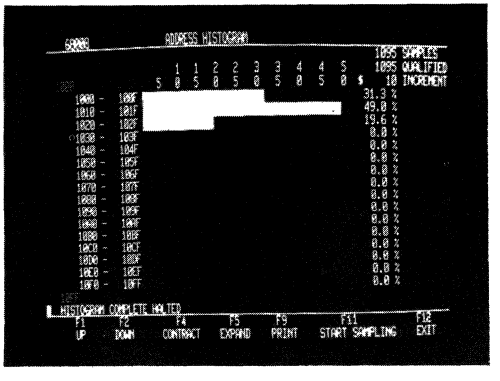


FIGURE 12 — BSA ADDRESS HISTOGRAM



HDS-400 COMMAND SET

The general syntax for commands is:
HDS-400 EMU0 > [NO]<command> [<parameters>][;<options>]

HDS-400 EMU0 >	Prompt signal for HDSdebug	< >	Angular brackets enclose a symbol (syntactic variable) that is replaced in a command line by one of a class of symbols it represents.
NO	Performs the inverse function of the command (if applicable)	|	Indicates that a choice is to be made. One of several entries, separated by this symbol, should be selected.
command	The command mnemonic		
parameters	Expressions or addresses, separated by a space when used.	[ ]	Square brackets enclose a symbol that is optional.
options	Multiple options may be selected, separated from parameters by a semicolon when used.	[ ]. . .	Square brackets followed by periods enclose a symbol that is optional and repetitive.

5

SYSTEM CONTROL COMMANDS

COMMAND NAME/DESCRIPTION	Press: FUNCTION KEYS	Enter: or CODE	SYNTAX
HELP DISPLAY Allows the user to display a list of HDSdebug commands, or obtain detailed information on a specific command For help on a specific command, the user may type "HELP" then press the appropriate function keys to enter the command	F4 help	HELP	[<command>]
MORE Redefines the function key menu If a desired function is not displayed, the user should press the "more" function key	F12 more		
TRANSPARENT Allows the user to communicate to the host computer To communicate with the HDS-400 again, the exit character must be inputted The default exit character is CTRL-S, however, an optional control character may be inputted	F10 transparent	TM	[<EXITcharacter>]
DISPLAY DEFAULTS Displays the current set of defaults used by HDSdebug for I/O and control operations	F6 F1 default display	DE	
MODIFY DEFAULTS Allows the user to modify the I/O and control specification defaults used by HDSdebug The space to be accessed via Memory Manipulation Commands may be set to Supervisor Program, Supervisor Data, User Program or User Data by using the Modify Defaults command A complete list of default options may be found in the HDS-400 operations manual	F6 F2 default modify	DE	<defopt>
DEBUG — EXIT Causes termination of the debug session Control is returned to the power-up program	F6 debug-exit	QUIT	

EMULATOR CONFIGURATION COMMANDS

These commands allow the user to configure the emulator memory map, signal paths from the user target system, and enable special functions.

COMMAND NAME/DESCRIPTION	Press: FUNCTION KEYS	Enter: or CODE	SYNTAX
EMULATOR MEMORY MAP Allows the user to specify the addresses where emulation memory will substitute for target system memory Emulation memory may be allocated in blocks of 4K bytes The "P" option causes the emulation RAM to be write protected, simulating ROM The Emulator Memory Map command with no parameters displays the current map specification	F8 emulator F5 map	EMAP	[⟨LOWaddr⟩ ⟨HIGHaddr⟩[,P]]
TARGET MEMORY MAP Allows the user to specify the addresses where the emulator will access target system memory This command is the complement of the Emulator Memory Map command The "P" option will not physically protect target memory, it will cause a warning message to be displayed whenever a write to protected target memory occurs The Target Memory Map command with no parameters displays the current map specification	F9 target	TMAP	[⟨LOWaddr⟩ ⟨HIGHaddr⟩[,P]]
EMULATOR SIGNAL Allows signals between the target system and Emulator Module to be enabled or disabled Signals that may be controlled are DTACK (DT), BERR (BE), BR/BGACK (BR), HALT (HA), INTERRUPTS (IN), CLOCK (CL), RESET (RE) Additional emulator functions are available AUTO-DTACK (AD) and AUTO-BERR (AB) will terminate a bus cycle when the target system fails to generate the appropriate signal within 160 processor clocks The LONG TIMEOUT (LO) will generate the AUTO-DTACK or AUTO-BERR after 160K processor clocks The FLAG (FL) and STOP (ST) functions will alert the user to a write to ROM memory as specified in the Emulator, or Target Memory Map commands The Emulator Signal command with no parameters will display the current status of the emulator signals and functions	F8 emulator F2 enable	ESIG	[⟨signal⟩|⟨function⟩]
	F8 emulator F3 disable	NOESIG	[⟨signal⟩|⟨function⟩]
REAL-TIME STATUS DISPLAY Provides a display of the emulation system's current status	F8 emulator F6 status	STAT	
EMULATOR CONFIGURATION Allows specification of a file to be used to load the emulator configuration The emulator configuration consists of the memory map definition and enable/disable status of the emulator signals If the FILEname specified does not already exist, a new file will be created and the current emulator configuration saved in that file The Emulator Configuration command with no parameters displays the current emulator type and configuration file name	F8 emulator F11 load or F8 emulator F8 select or F8 emulator F1 display	EMUL	[⟨emulator#⟩|⟨FILEname⟩]

MEMORY MANIPULATION COMMANDS

These commands allow the user to change memory, download the target program, and examine internal processor registers.

COMMAND NAME/DESCRIPTION	Press: FUNCTION KEYS		Enter: or CODE	SYNTAX
MEMORY DISPLAY Displays memory contents If the "DI" option is entered, the specified number of instructions are shown in disassembled form, otherwise the specified number of bytes are shown in both hexadecimal and ASCII form	F1 memory	F1 display	MD	⟨addr⟩[⟨count⟩][,DI]
MEMORY MODIFY Allows memory contents to be examined and altered Data length to be displayed or modified may be specified as Byte (B), Word (W), or Long Word (L) The type of display or modificaton may be Disassembled (DI), No Read (N) for I/O devices, Odd Addresses Only (O), and Even Addresses Only (V) To replace an instruction in disassembled display, type a space, then enter the assembler mnemonic and operand fields Entering a " " will terminate the Memory Modify command	F1 memory	F2 modify	MM	⟨addr⟩[,[⟨type⟩][⟨length⟩]]
MEMORY (BLOCK) MOVE A block of memory between the specified "LOW" and "HIGH" memory addresses is copied to a new location starting at the specified "TO" address	F1 memory	F4 move	BM	⟨LOWaddr⟩⟨HIGHaddr⟩ ⟨TOaddr⟩
MEMORY (BLOCK) SEARCH The user specifies a given data pattern and address range Memory is then searched from the "LOW" address to the "HIGH" address specified, and the address of each location where the given data pattern is found will be displayed	F1 memory	F5 search	BS	⟨LOWaddr⟩⟨HIGHaddr⟩ ⟨data⟩
MEMORY (BLOCK) FILL The user specifies a given data pattern and memory range Memory is then written to between the "LOW" address and "HIGH" address with the specified data The data size may optionally be set to Byte (B), Word (W), or Long Word (L)	F1 memory	F6 fill	BF	⟨LOWaddr⟩⟨HIGHaddr⟩ ⟨data⟩[,⟨opt⟩]
MEMORY SET Deposits given data in memory, starting at the address specified	F1 memory	F10 define	MS	⟨addr⟩⟨data⟩[⟨data⟩]
REGISTER DISPLAY Displays the contents of the emulator processor registers The DF command displays all processor registers, the MD command displays a specified register Valid registers are Address Registers (A0 through A7), Data Registers (D0 through D7) and Control Registers (PC, SR, SS, US) For the MC68010 only Vector Base Register (VB), Source Function Code Register (SF), Destination Function Code Register (DF) Offset Registers (R0 through R7) for use in debug of relocatable code may also be displayed when the "MD" command is used	F2 register	F1 display	DF or MD	 ⟨reg⟩
REGISTER MODIFY Allows setting of a specified register to a desired value (see Register Display command for list of register definitions)	F2 register or F3 offset-reg	F2 modify F10 define	MS	⟨reg⟩⟨data⟩
OFFSET REGISTER DISPLAY Displays the contents of the eight offset registers Offset registers may be used in address expressions to ease debug of relocatable code	F3 offset-reg	F1 display	OF	
DISPLAY CONSTANT Allows the user to resolve arithmetic expressions and/or symbolic values The result is displayed in symbolic, hexadecimal, decimal and binary formats	F1 constant		DC	⟨expr⟩

(continued)

MEMORY MANIPULATION COMMANDS — (Continued)

COMMAND NAME/DESCRIPTION	Press: FUNCTION KEYS		Enter: or CODE	SYNTAX
LOAD PROGRAM Causes "S" records from the host to be loaded into target or emulation memory as defined by the Emulator Memory Map command A checksum is performed on each record as it is received	F9 program	F11 load	LOAD	⟨FILEname⟩
DUMP PROGRAM Allows user to save one to seven areas of target memory on the host disk in "S" record format Each area is specified by an address range and is stored in the specified file name	F9 program	F12 save	DUMP	⟨FILEname⟩⟨LOWaddr⟩ ⟨HIGHaddr⟩

5

DEBUG COMMANDS

These commands allow the user to direct and observe the program during execution. These commands include starting and stopping the program, setting breakpoints, and defining macro programs to facilitate the creation of user-specific commands

COMMAND NAME/DESCRIPTION	Press: FUNCTION KEYS		Enter: or CODE	SYNTAX
DEFINE BREAKPOINT Sets breakpoints at target program locations Up to 16 breakpoint locations may be specified within 4 address ranges Each breakpoint may be specified with a decimal count to effectively debug program loops The BSA option (BSA) is used to enable a complex breakpoint generated by the BSA to cause a break in the emulator The Timeout Option (TO) will cause a breakpoint whenever the emulator generates AUTO-DTACK or AUTO-BERR (see Emulator Signal command)	F7 breakpoint	F10 define	BR	⟨addr⟩[,⟨count⟩] | [BSA| TO]
DELETE BREAKPOINT Removes a specified breakpoint If no parameters are given, all breakpoints are removed	F7 breakpoint	F3 delete	NOBR	[⟨addr⟩] |[BSA| TO]
DISPLAY BREAKPOINTS Displays the current list of breakpoints	F7 breakpoint	F1 display	BR	
GO Starts target program execution at a specified address If no address is specified, the program will begin execution at the location pointed to by the current program counter Only a Breakpoint, Write Violation ("STOP" function enabled in an Emulator Signal command) or a user-generated STOP command will cause the emulator to cease execution in the target system The HDS-400 prompt will not be displayed after the GO command is entered To restore the prompt or to enter another command, hit the Break key	F9 program	F9 execute	GO	[⟨addr⟩]
STOP Terminates target program execution After a GO command is entered, it will be necessary to hit the Break key prior to entering the STOP command	F8 emulator	F10 stop	STOP	
TRACE PROGRAM Causes target program execution, one instruction at a time, for a specified count of instructions After execution, the next instruction is displayed in disassembled format	F9 program	F8 trace	TR	[⟨count⟩]
DEFINE LOCAL SYMBOL Allows the user to create local symbols A symbol name consists of one to eight alphanumeric characters Symbol values are normally entered in hexadecimal To enter a symbol in decimal, the ⟨value⟩ should be preceeded with an ampersand (&)	F5 symbol	F10 define	SD	⟨name⟩⟨value⟩

(continued)

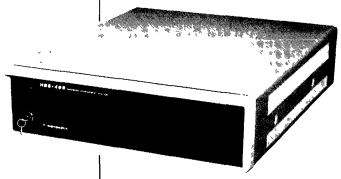
DEBUG COMMANDS (Continued)

COMMAND NAME/DESCRIPTION	Press: FUNCTION KEYS		Enter: or CODE	SYNTAX
DELETE LOCAL SYMBOL Deletes a specified local symbol from the current list If entered with no parameters, all local symbols are deleted	F5 symbol	F3 delete	NOSD	[⟨name⟩]
DISPLAY LOCAL SYMBOLS Displays the current list of local symbols and their values When a name is specified, the value of that symbol will be displayed	F5 symbol	F1 display	SD	[⟨name⟩]
SAVE LOCAL SYMBOLS Allows user to save local symbols in a specified file on the host disk If the file name specified already exists, it will be overwritten, otherwise, a new file will be created This file will also contain the current macros The result of this command is the same as the Save Macros command	F5 symbol	F12 save	FS	⟨FILEname⟩
LOAD LOCAL SYMBOLS Allows a previous list of local symbols and macros to be loaded from a file on the host disk The result of this command is the same as the Load Macros command	F5 symbol	F11 load	FR	⟨FILEname⟩
DEFINE MACRO Allows the user to create complex personalized commands consisting of any number of debugger commands Macros may be created by pressing function keys or entering the command codes directly Parameters may be passed to the macro to substitute for the arguments normally associated with a particular command Hitting a carriage return in response to the "M=" prompt will exit the Define Macro mode and return the HDS-400 prompt Typing the macro name followed by a carriage return will invoke execution of that macro	F11 macro	F10 define	MA	⟨name⟩
DELETE MACRO Deletes a specified macro from the current list If the delete Macro Command is entered with no parameters, all macros will be deleted	F11 macro	F3 delete	NOMA	[⟨name⟩]
DISPLAY MACRO Displays a specified macro(s) If no name is specified, the entire current list of macros is displayed	F11 macro	F1 display	MA	[⟨name⟩]
EDIT MACRO Allows modification of a macro by insertion, deletion, or replacement of a line Specifying an existing line number will cause the replacement of that line with the specified data If no data is provided, the line will be deleted Insertion is performed by specifying a non-existing line number Renumbering is automatic	F11 macro	F2 modify	MAE	⟨name⟩⟨line #⟩[,⟨data⟩]
SAVE MACROS Allows user to save the current list of macros in a specified file on the host disk If the file name already exists, it will be overwritten, otherwise, a new file will be created This file will also contain the current local symbols The result of this command is the same as the Save Local Symbols command	F11 macro	F12 save	FS	⟨FILEname⟩
LOAD MACROS Allows a previous list of macros and local symbols to be loaded from a file on the host disk The result of this command is the same as the Load Local Symbols command	F11 macro	F11 load	FR	⟨FILEname⟩
COMMAND REPEAT Allows the user to specify multiple invocations of the next command entered The next command will be repeated the number of times specified by the count The Break key will terminate the process	F10 command	F7 repeat	CR	[⟨count⟩]

5

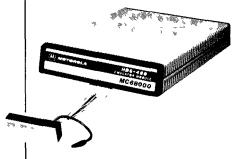
HDS-400 CONTROL STATION SPECIFICATIONS

CHARACTERISTICS	SPECIFICATIONS
Baud rates	
Terminal Link	50, 75, 110, 134 5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600
Host Link	Up to 56K synchronous 19 2K or 9600 asynchronous
Temperature	
Operating	0° to 40°C
Storage	−40° to 85°C
Physical	
Length	23.6 in (59.8 cm)
Width	19 0 in. (48 2 cm)
Height	5 7 in (14 5 cm)
Weight	42 lbs (19 kg)
Power requirements (HDS-400 Control Station)	90–128 Vac, 47–63 Hz, 380 W max 180–256 Vac, 47–63 Hz, 380 W max



EMULATOR MODULE SPECIFICATIONS

CHARACTERISTICS	SPECIFICATIONS
Microprocessor	MC68000, MC68008, or MC68010
Power requirements (Provided by HDS-400 Control Station)	+5 Vdc @ 5 A (max) ±12 Vdc @ 100 mA (max)
Temperature	
Operating	0° to 40°C
Storage	−40° to 70°C
Physical	
Length	12 8 in (32 5 cm)
Width	9 0 in (22 86 cm)
Height	2.5 in (6 4 cm)



MOTOROLA "S" RECORD FORMAT

The HDS-400 transmits and receives "S" record serial data on its host link When viewed by the user, "S" records are essentially character strings consisting of several fields which identify the record type, record length, memory address, code/data, and checksum. Each byte of binary data is encoded as a two-character hexadecimal number the first character represents the high-order four bits, and the second the low-order four bits of the byte

There are three types of "S" records; sign-on records, data records, and end-of-file records The format of each record contains five fields·

Field 1 — Two characters used to indicate sign-on (S0), start of data record (S2), or end-of-file (S9)

Field 2 — Two characters, containing the byte count of the record (including data, address and checksum).

Field 3 — Six characters indicating the first byte address in the record

Field 4 — Variable number of characters representing the data

Field 5 — Two characters representing the checksum The checksum is the one's complement of the binary summation of the bytes in the byte count, address and data fields in hex notation.

FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5
S1	HH	HHHHHH	HHHHHHHH	HH
SIGN ON S0 START S2 EOF S9	BYTE COUNT	ADDRESS	DATA	CHECKSUM

H = Indicates hexadecimal data

A detailed description of the "S" record format is contained in Appendix B of the HDS-400 Operations Manual (M68HDS4OM)

HDS-400 ORDERING INFORMATION

The HDS-400 Hardware/Software Development Station has been partitioned with options and part numbers which give the user versatility in defining his development system configuration. The user may choose from three host computers with a variety of operating systems.

The Hardware and Software Configuration Guides shown below define the Motorola part numbers which must be ordered to define EXORmacs, VME/10, or DEC VAX-based HDS-400 systems. Each of the HDS-400 Control Stations is delivered pre-wired to accept the optional Real-Time Bus State Analyzer (BSA) and an Emulation Memory Module. The Recommended HDS-400 Options Configuration Guide shows the BSA part numbers which apply in HDS-400 applications, and the three Emulation Memory Modules which are available. Detailed descriptions of each part number shown in the three Configuration Guides are also included below.

HOST/HDS-400 SYSTEM HARDWARE CONFIGURATION GUIDE

5

HOST	OPERATING SYSTEM	**M68KHDS400** CONTROL STATION ("DLC" INTERFACE)	**M68KHDS400A** CONTROL STATION (RS-232C INTERFACE)	**M68KHDS16FB** 16-BIT FAMILY INTERFACE MODULE	**M68000HDS4** MC68000 EMULATOR MODULE	**M68008HDS4-8** MC68008 EMULATOR MODULE (8 MHz)	**M68010HDS4-8** MC68010 EMULATOR MODULE (8 MHz)	**M68KHDSDLC** 56K-BAUD DATA LINK CONTROLLER	**MVME 400** DUAL RS-232C SERIAL PORT MODULE	**M68SXD10155A** EXORterm 155 TERMINAL	**SOFTWARE** ORDERING INFORMATION
EXORmacs (1st Station)	VERSAdos	X		X	X or	X or	X	X		X	See Below
EXORmacs (Addt'l Stations)	VERSAdos	X		X	X or	X or	X			X	See Below
VME/10	VERSAdos		X	X	X or	X or	X		X		See Below
VAX	VMS		X	X	X or	X or	X			X	See Below
VAX	UNIX SYSTEM V		X	X	X or	X or	X			X	See Below

HOST/HDS-400 SYSTEM SOFTWARE CONFIGURATION GUIDE

HOST	OPERATING SYSTEM	**M68KHDS4-1F** EXORmacs/VERSAdos S/W ON 8" FLOPPY		**M68KHDS4-1H** EXORmacs/VERSAdos S/W ON 14" HARD DISK		**M68KHDS4-1L** EXORmacs/VERSAdos S/W ON 8" LARK DISK	**M68KHDS4-2** VME/10 — VERSAdos S/W ON 5¼" FLOPPY	**M68KHDS4-3T** VAX/VMS S/W ON 9-TRACK TAPE	**M68KHDS4-4T** VAX/UNIX S/W ON 9-TRACK TAPE
EXORmacs (1st Station)	VERSAdos	X	or	X	or	X			
EXORmacs (Addt'l Stations)	VERSAdos								
VME/10	VERSAdos						X		
VAX	VMS							X	
VAX	UNIX SYSTEM V								X

RECOMMENDED HDS-400 OPTIONS CONFIGURATION GUIDE

HOST	OPERATING SYSTEM	REAL-TIME BUS STATE ANALYZER: **M68BSAC** BUS STATE ANALYZER CONTROL MODULE	REAL-TIME BUS STATE ANALYZER: **M68BSA1-1** MC68000, 68010, 68451 PERSONALITY MODULE		REAL-TIME BUS STATE ANALYZER: **M68BSA3** MC68008 PERSONALITY MODULE		EMULATION MEMORY EXPANSION: **M68KHDS4EMM1** EMULATION MEMORY MODULE (EXPANDS TOTAL EMULATION RAM TO 64K)		EMULATION MEMORY EXPANSION: **M68KHDS4EMM2** EMULATION MEMORY MODULE (EXPANDS TOTAL EMULATION RAM TO 128K)		EMULATION MEMORY EXPANSION: **M68KHDS4EMM3** EMULATION MEMORY MODULE (EXPANDS TOTAL EMULATION RAM TO 256K)
ALL HOST/OPERATING SYSTEM COMBINATIONS SHOWN ABOVE		X	X	or	X		X	or	X	or	X

HDS-400 CONTROL STATIONS

PART NUMBER	DESCRIPTION
M68KHDS400	HDS-400 Control Station (90–120 V, 50/60 Hz) for use with Data Link Controller (M68KHDSDLC) interface to EXORmacs host. Includes four-slot chassis with built-in 228 W power supply, HDS-400 Control Module, a host interface cable, connectors and manuals. Order appropriate MPU Family Interface Module, Emulator Module, Software, and EXORmacs Update Requirements from tables below.
M68KHDS402	Same as M68KHDS400 except for use with 200–260 V, 50 Hz.
M68KHDS400A	HDS-400A Control Station (90–120 V, 50/60 Hz) for use with RS-232C interface to Motorola VME/10 or DEC VAX host Includes four-slot chassis with built-in 228 W power supply, HDS-400 Control Module, two RS-232C interface cables, connectors and manuals Order appropriate MPU Family Interface Module, Emulator Module(s), Software, and MVME400 Communications Module if a VME/10 host is used.
M68KHDS402A	Same as M68KHDS400A except for use with 220–260 V, 50 Hz.
M68HDS4UM/D1	HDS-400 Hardware Development Station User's Manual (for extra copies) The User's Manual provides general information, hardware preparation and installation instructions, plus a functional description for the HDS-400 Hardware Development Station.
M68HDS4OM/D3	HDS-400 Hardware Development Station Operations Manual (for extra copies) This manual describes how to operate the HDS-400.

MPU FAMILY INTERFACE MODULE

PART NUMBER	DESCRIPTION
M68KHDS16FB	16-bit M68000 Family Interface Module for use with MC68000, MC68008, and MC68010 Emulator Modules. Includes necessary cable, connectors and user's manual. The MPU Family Interface Module for the emulator being used in the system should be installed in the HDS-400 Control Station.
M68HDS4FB1/D1	16-bit Family Interface Module User's Manual (for extra copies)

EMULATOR MODULES

(Each module is shipped with necessary cable/connector assemblies and user's manual)

PART NUMBER	DESCRIPTION
M68000HDS4	MC68000 Emulator Module (12.5 MHz)
M68HDS4EM1/D1	MC68000 Emulator User's Manual (for extra copies)
M68008HDS4-8	MC68008 Emulator Module (8 MHz)
M68HDS4EM2/D1	MC68008 Emulator User's Manual (for extra copies)
M68010HDS4-8	MC68010 Emulator Module (8 MHz)
M68HDS4EM3/D1	MC68010 Emulator User's Manual (for extra copies)

EXORmacs UPDATE REQUIREMENTS

PART NUMBER	DESCRIPTION
M68KHDSDLC	56K-Baud Data Link Controller. Includes printed circuit boards, necessary cables, connectors and hardware. Installation in the EXORmacs is performed by Motorola Field Service This link provides a synchronous serial local communications link between the EXORmacs host and the HDS-400 Hardware/ Software Development Station(s). One link is required for each EXORmacs host.
SMM1193	Update of EXORmacs software and firmware for floppy disk system. Includes VERSAdos 3 0, assembler, SYMbug/A and linkage editor software on floppy diskettes; plus a firmware set for MPU and DEbug. This update is required if the EXORmacs host has not been updated from VERSAdos 2.1 to 3.0.
SMM1194	Update of EXORmacs software and firmware for hard disk system. Includes VERSAdos 3.0, assembler, SYMbug/A and linkage editor software on 14-inch hard disk cartridge; plus a firmware set for MPU, DEbug and disk controller. This update is required if the EXORmacs host has not been updated from VERSAdos 2.1 to 3.0.
SMM1195	Update of Multi-Channel Communications Module (MCCM) firmware for VERSAdos 3.0/4.1 or later compatibility. One set of update firmware must be ordered for each MCCM installed in the EXORmacs host. Existing hosts may have zero, one or more MCCM(s) installed. This update is required if the EXORmacs host has not been updated from VERSAdos 2.1 to 3.0.

VME/10 UPDATE REQUIREMENTS

PART NUMBER	DESCRIPTION
MVME400	Dual RS-232C Serial Port Module. Includes printed circuit board, I/O connectors, cables, front panel, and user's manual. The MVME400 is configured and installed in the VME/10 I/O card cage by the user. Instructions are provided in the HDS-400 User's Manual. This module provides an asynchronous serial local communications host link, and an asynchronous serial terminal link between the VME/10 host and the HDS-400 Hardware/Software Development Station.
MVME400/D1	User's Manual (for extra copies).

HDS-400 SOFTWARE

PART NUMBER	DESCRIPTION
M68KHDS4-1F	HDS-400 system software for use with EXORmacs host and VERSAdos operating system on 8-inch floppy diskettes (Requires VERSAdos 4.3 or later).
M68KHDS4-1H	HDS-400 system software for use with EXORmacs host and VERSAdos operating system on 14-inch hard disk cartridge (Requires VERSAdos 4.3 or later).
M68KHDS4-1L	HDS-400 system software for use with EXORmacs host and VERSAdos operating system on Lark 8-inch hard disk cartridge (Requires VERSAdos 4.3 or later).
M68KHDS4-2	HDS-400 system software for use with Motorola VME/10 host and VERSAdos operating system on 52-inch floppy diskettes.
M68KHDS4-3T	HDS-400 system software for use with DEC VAX host and VMS operating system on 9-track magnetic tape.
M68KHDS4-4T	HDS-400 system software for use with DEC VAX host and UNIX SYSTEM V operating system on 9-track magnetic tape.
BR126R1	Please refer to the "Motorola Microprocessor Software Catalog" for available VAX cross software.

NOTE Future enhancements and problem corrections will be provided at no extra charge to customers who have signed a Motorola Software Maintenance Agreement Hardware On-site Maintenance Agreements are also available, which provide for service at the customer site covering labor and materials Please contact a Motorola salesperson, distributor, or representative for agreement forms and further details

EXORterm 155 TERMINAL

PART NUMBER	DESCRIPTION
M68SXD10155A	EXORterm 155 (90–120 V, 50/60 Hz) is a M6800-based display terminal and console. It consists of a Video Display Enclosure and an expanded Keyboard. A user's manual is included.
M68SXD155/D2	User's Manual (for extra copies).

OPTIONAL BUS STATE ANALYZER SUPPORT

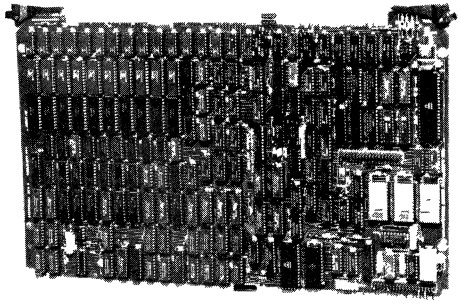
PART NUMBER	DESCRIPTION
M68BSAC	Real-Time Bus State Analyzer Control Module for use with BSA Personality Module. Contains analyzer hardware, control firmware, and I/O ports User's Manual included
M68BSA1-1	MC68000, MC68010, MC68451 Personality Module requires a BSA Control Module. Includes associated cabling, chip probes, and documentation.
M68BSA3	MC68008 Personality Module requires a BSA Control Module. Includes associated cabling, chip probe, and documentation.
M68KBSA/D6	User's Manual (for extra copies).

OPTIONAL EMULATION MEMORY MODULES

PART NUMBER	DESCRIPTION
M68KHDS4EMM1	Emulation Memory Module provides additional 10 MHz no wait cycle emulation RAM when connected to the M68000 Family Interface Module. Total emulation memory is 64K bytes when this module is installed in a VERSAmodule card slot in the HDS-400 Control Station Includes a User's Manual
M68KHDS4EMM2	Total emulation RAM is 128K bytes when this Emulation Memory Module is installed in the HDS-400 Control Station Includes a User's Manual.
M68KHDS4EMM3	Total emulation RAM is 256K bytes when this Emulation Memory Module is installed in the HDS-400 Control Station Includes a User's Manual
M68HDS4MM1/D1	Emulation Memory Module User's Manual (for extra copies)

To order Motorola literature, including product User's Manuals, Catalogs, Data Sheets, please contact our Literature Distribution Center, P.O. Box 20912, Phoenix, Arizona 85036 Telephone (602) 994-6561.

Real-Time Bus State Analyzer Control Module And Enclosure



M68BSAC

M68BSACE

Analyzer System Features

- Connects Directly to the Motorola MPU and/or MMU in the User's System
- Up to 79 Channels by 128 States Trace Memory
- MPU and MMU Signals Displayed Using Their Mnemonics
- Single Trace
- Trace and Compare with Previous Trace
- Trace a Selected State Only
- Trace Inside or Outside an Address Window
- Trace Through Processor Reset
- Up to Seven Step Qualifier Sequence on a Given Address State
- Trigger Pulse Outputs Capable of Triggering Complex Breakpoints in HDS-400 Development Stations
- Inverval Timer
- Address Histogram Display for Software Performance Analysis
- Discrete Signal Histogram Display for Hardware Analysis
- Self-Test on Power Up
- Hard Copy Capability with Optional Printer
- Easy Menu-Type Control by Function Keys of EXORterm 155 or VME/10
- EXORmacs, VME/10, and HDS-400 Hardware/Software Development Station Compatible

Enclosure Features:

- Protects Control Module within Attractive, Compact Enclosure
- Ruggedized Glass-Reinforced Plastic (ABS)
- Sturdy Construction with Stabilized Base
- Allows BSA Use Where Space Restrictions Exist
- Enhances BSA Use in a Non-VERSAbus Environment
- Provisions Made for Easy Attachment of User Power Supply
- Host/Terminal RS-232C and Printer Port Connectors for Direct Cable Hook-Up

The Real-Time Bus State Analyzer (BSA) is a highly intelligent diagnostic tool that is designed specifically for use with microprocessors It consists of a Control Module and one of several "Personality Modules" The Control Module contains the analyzer hardware, control software, and I/O ports The Bus State Analyzer Personality Module interfaces to selected MPU and/or MMU signals The BSA stores data that appears on up to 79 different lines Six or eight of these lines are available for external connection, and the remainder are prewired to chip lines

To facilitate the gathering of data from the device(s), a set of qualified triggering modes are provided Continuous Trace Mode, Sequential Trigger Mode, and Window Trigger Mode In order to service these triggering modes and provide a complete set of operating features, an MC6809 microprocessor is located on the Control Module with local intelligence running from an operating system based in ROM This operating system provides the data analysis and formatting functions for the operator, including the interface to the hardware sampling the bus

To reduce system redundancy, the terminal used by the operator to communicate with the development system also serves to link him to the intelligence aboard the BSA (it is a requirement of the CRT-based analyzer operating system that the terminal used be an EXORterm 155 Display Console or a VME/10) This communication is acheived by means of a phantom or transparent serial link feeding from the terminal through the Bus State Analyzer Control Module, and then to the normal terminal input channel of the development system The logic onboard the Control Module determines whether the information traveling over the link is destined for the development system, the Bus State Analyzer or the terminal This allows the operating system or the user's software to run in the development system while analysis is being performed Additionally, a means is provided for the

BSA to operate in a stand alone mode with only a terminal connected.

The BSA is simple to program through the use of the EXORterm's or VME/10's function key capability A menu defining the function keys is displayed at the bottom of the CRT Initially (after power-up), when the BSA is talking to the system terminal, the analyzer displays a message regarding the result of the internal self test After the self test has been completed, the BSA may be programmed to a specific set of trace conditions.

CONTINUOUS TRACE MODE samples signals and stores signal information continuously on each occurrence of the clocking signal. It is primarily a default mode which the BSA automatically enters when power is first applied. There are no qualifications for the BSA to begin gathering information, so it will always be storing the processor and/or MMU signal states This default mode is particularly useful when a sudden catastrophic failure occurs during a debugging session, before the user is able to configure the BSA It is very likely that the events leading up to the failure will be stored in the BSA's trace store buffer

SEQUENTIAL TRIGGER MODE requires that a series of events occur before the instrument triggers and starts to gather real-time trace data, or conversely triggers and stops gathering data Sequence Terms, as these events are called, must occur in order of specification, or triggering will not take place Up to seven Sequence Terms can define a BSA trigger Each event is defined from up to 79 qualifier lines. A Sequence Reset Term can also be specified to reset the BSA and cause the instrument to begin looking for the Sequence Terms again. Sequential triggering is most useful for debugging complex software including loops, nested subroutines and complex branches

WINDOW TRIGGER MODE provides a means of causing signal states to be stored if address accesses occur inside or outside of a particular address range Both the upper and lower bounds of the range are programmable, and the size is variable from a single address to the full range of the memory map Window triggering is useful for following programs that suddenly and unexpectedly leave the memory area in which they should be operating. It is also applicable for observing access violations in a multiple user environment

PERFORMANCE HISTOGRAMS provide displays for the analysis of software and hardware system performance The **address (software)** histograms give an indication of the relative frequency of memory accesses within a particular memory address range, with the exact range specified by the user This histogram provides a means of determining where a program spends the greatest amount of time The resulting information can then be used to optimize inefficient code A **discrete (hardware)** performance histogram is also provided to display the relative frequency of combinations of four of the optional user-selected signals within a user system

Bus State Analyzer Control Module Specifications

Characteristics	Specifications
Trigger Outputs	One Scope Trigger from the final occurrence of each Sequence Term found One Scope Trigger from the final occurrence of Trigger Term
Trace Store Buffer	128 events deep by 79 bits wide with a second comparison array of equal size
Trigger Qualifiers	Up to 7 events from 79 qualifier lines, 1 level of magnitude comparison, any bits may be "don't cared"
Selective Trace Store	Single Combination (Trace only)
Trace Data Outputs	Selectable — binary, decimal, octal
Input Device Requirements	EXORterm 150, 155, 220 or VME/10, Transparent Mode supported with variable baud rates
Output Requirements	Transparent Mode provides RS-232C output with variable baud rates
Onboard Intelligence	MC6809
RAM (scratch)	2K × 8
Processor ROM	20K × 8 Operating System (Control Module)
Power Required	+5.0 Volts at 9 0 Amps +12 Volts at 1.0 Amp -12 Volts at 0 1 Amp
Environmental	0 to 50° C operating temperature
Module Dimensions Width × Height	14.5 in. × 9.25 in. (36.8 cm × 23 5 cm)

M68BSAC
M68BSACE

The Bus State Analyzer Control Module is a VERSAmodule size board (14 5 in × 9 25 in) When used with an HDS-400 Hardware Development Station or an EXORmacs System, it requires one VERSAbus slot but merely receives power and ground from the bus Personality Modules for eight-bit and sixteen-bit devices and buses are now available The ordering information below summarizes the BSA part numbers for the Control Module, the Control Module with Enclosure, and the Personality Modules currently available

CONTROL MODULE ENCLOSURE

The BSA Control Module with Enclosure provides a convenient and inexpensive way to house the Control Module With the Control Module housed separately in its own container, one more slot is free for use in the EXORmacs System chassis or the Hardware Development Station chassis

Because the Control Module follows VERSAmodule mechanical specifications, it cannot be enclosed in a non-VERSAbus backplane For such Personality Modules as the EXORbus, which is used in a non-VERSAbus EXORciser, this alternative housing is suggested

A power connector is built into the enclosure back panel for easy connection of the user's own supply The power required is +5 0 V at 9 5 Amps, +12 V at 1 5 Amps, and -12 V at 0 1 Amp Enclosure dimensions are 17 25 inches × 12 87 inches × 2 69 inches (width × depth × height).

The BSA Control Module is 14 5 inches × 9 25 inches × 1 6 inches (width × depth × height) This module can plug into a slot of the VERSAmodule System chassis, the EXORmacs Development System, or an HDS-400 chassis The BSA Control Module enclosure is not required for operation in these systems, but its use is recommended if another slot is needed for expansion

5

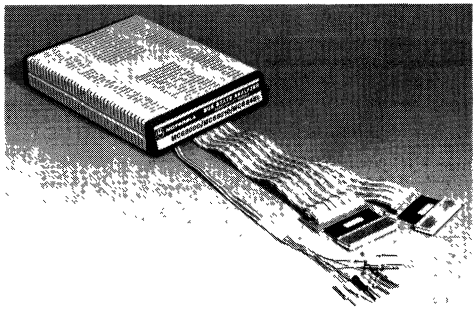
Ordering Information

Part Number	Description	Remarks
M68BSAC	Real-Time Bus State Analyzer Control Module for use with a VME/10 or EXORterm 155 and Bus State Analyzer Personality Modules User's Manual included	Without Enclosure
M68BSACE	Real-Time Bus State Analyzer Control Module in Standalone Enclosure for use with a VME/10 or EXORterm 155 and Bus State Analyzer Personality Modules User's Manual included	Enclosure Included
M68BSA1-1	Real-Time MC68000/MC68010/MC68451 Bus State Analyzer Personality Module Includes associated cabling, chip probes, and documentation NOTE A Control Module is required to complete the system	Enclosure suggested unless used with HDS-400, EXORmacs or VERSAmodule chassis.
M68BSA2	Real-Time MC6800/MC6802/MC6808/MC6809/MC6809E/ MC6829 Bus State Analyzer Personality Module Includes associated cabling, chip probes, and documentation NOTE A Control Module is required to complete the system	Enclosure Suggested
M68BSA3	Real-Time MC68008 Bus State Analyzer Personality Module Includes associated cabling, chip probe, and documentation NOTE A Control Module is required to complete the system	Enclosure suggested unless used with HDS-400, EXORmacs or VERSAmodule chassis
M68BSA4	Real-Time MC6801/MC6803/MC68120/MC68701 Bus State Analyzer Personality Module Includes associated cabling, chip probes, and documentation NOTE A Control Module is required to complete the system	Enclosure Suggested
M68BSA5	Real-Time VERSAbus State Analyzer Personality Module Includes associated cabling and documentation NOTE A Control Module is required to complete the system	Enclosure suggested unless used with an EXORmacs or VERSAmodule chassis.
M68BSA6	Real-Time EXORbus State Analyzer Personality Module Includes associated cabling and documentation NOTE A Control Module is required to complete the system	Enclosure Suggested
M68KBSA	Real-Time Bus State Analyzer User's Manual with Personality Module Appendixes (for extra copies)	Enclosure documented in User's Manual

Related Motorola Development Systems Products

Part Number	Description
M68SXD10155A	EXORterm 155 is a M6800-based display terminal and console. It consists of a Video Display Enclosure and an expanded Keyboard.
HDS-400 (configured with several part numbers)	HDS-400 Microprocessor Hardware/Software Development Station emulates the Motorola M68000 family of MPU's including the MC68000, MC68008 and the MC68010 The BSA Control Module may easily be mounted within the HDS-400 Control Station
M68KMACS	EXORmacs is a MC68000-based development system. It utilizes the standard VERSAbus, and is designed to allow expansion to multiuser capability
M68K102B1	VME/10 is a compact minicomputer workstation which allows expansion and customization through VMEbus and I/O channel card cages. It includes a 10 MHz MC68010 MPU and MC68451 MMU, 384K bytes of RAM, 5-1/4" Winchester and floppy disk subsystems, a 15 inch video display, and an expanded ASCII keyboard.
MVMC682	VMC68/2 microcomputer system is built upon the MC68000-based VERSAmodule family of board-level products. It is designed to allow expansion and functional tailoring of both hardware and software system elements.

Real-Time MC68000/ MC68010/MC68451 Bus State Analyzer Personality Module



Analyzer System Features:

- Connects Directly to the MC68000 or MC68010 MPU in the User's System
- Additional Probe Connects Directly to an MC68451 MMU in the User's System
- 79 Channels by 128 States Trace Memory
- Designed Specifically for the MC68000 and MC68010 MPU's and the MC68451 MMU
- MPU and MMU Signals Displayed Using Their Mnemonics
- Single Trace
- Trace and Compare with Previous Trace
- Trace a Selected State Only
- Trace Inside or Outside an Address Window
- Trace Through Processor Reset
- Up to Seven Step Qualifier Sequence on a Given Address State
- Trigger Pulse Outputs Capable of Triggering Complex Breakpoints in HDS-400 Development Stations
- Interval Timer
- Address Histogram Display for Software Performance Analysis
- Discrete Signal Histogram Display for Hardware Analysis
- Self-Test on Power Up
- Hard Copy Capability with Optional Printer
- Easy Menu-Type Control by Function Keys of EXORterm 155 or VME/10
- EXORmacs, VME/10, and HDS-400 Hardware/Software Development Station Compatible

The Real-Time Bus State Analyzer (BSA) is a highly intelligent diagnostic tool that is designed specifically for use with microprocessors It consists of a Control Module and one of several "Personality Modules" The Control Module contains the analyzer hardware, control software, and I/O ports The MC68000/68010/68451 Bus State Analyzer "Personality Module" interfaces to selected MC68000 or MC68010 MPU and MC68451 MMU signals The BSA stores data that appears on 79 different lines Six of these lines are available for external connection, and 73 are prewired to MC68000/MC68010 and MC68451 lines

To facilitate the gathering of data from the MC68000/68010/68451, a set of qualified triggering modes are provided· Continuous Trace Mode, Sequential Trigger Mode, and Window Trigger Mode

CONTINUOUS TRACE MODE samples signals and stores signal information continuously on each occurrence of the clocking signal It is primarily a default mode which the BSA automatically enters when power is first applied There are no qualifications for the BSA to begin gathering information, so it will always be storing the processor and/or MMU signal states This default mode is particularly useful when a sudden catastrophic failure occurs during a debugging session, before the user is able to configure the BSA It is very likely that the events leading up to the failure will be stored in the BSA's trace store buffer

SEQUENTIAL TRIGGER MODE requires that a series of events occur before the instrument triggers and starts to gather real-time trace data; or conversely triggers and stops gathering data Sequence Terms, as these events are called, must occur in order of specification, or triggering will not take place. Up to seven Sequence Terms can define a BSA trigger. Each event is defined from 79 qualifier lines. A Sequence Reset Term can also be specified to reset the BSA and cause the instrument to begin looking for the Sequence Terms again. Sequential triggering is most useful for debugging complex software including loops, nested subroutines and complex branches

WINDOW TRIGGER MODE provides a means of causing signal states to be stored if address accesses occur inside or outside of a particular address range Both the upper and lower bounds of the range are programmable, and the size is variable from a single address to the full range of the memory map Window triggering is useful for following programs that suddenly and unexpectedly leave the memory area in which they should be operating It is also applicable for observing access violations in a multiple user environment

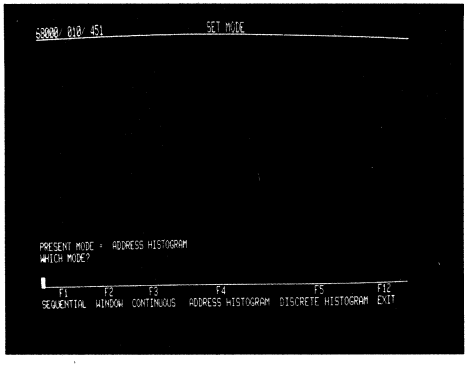
PERFORMANCE HISTOGRAMS provide displays for the analysis of software and hardware system performance The **address (software)** histograms give an indication of the relative frequency of memory accesses within a particular memory address range, with the exact range specified by the user This histogram provides a means of determining where a program spends the greatest amount of time The resulting information can then be used to optimize inefficient code A **discrete (hardware)** performance histogram is also provided to display the relative frequency of combinations of four of the optional user-selected signals within a user system

In order to service these triggering modes and provide a complete set of operating features, an MC6809 microprocessor is located on the Control Module with local intelligence running from an operating system based in ROM This operating system provides the data analysis and formatting functions for the operator, including the interface to the hardware sampling the bus

To reduce system redundancy, the terminal used by the operator to communicate with the development system also serves to link him to the intelligence aboard the BSA (it is a requirement of the CRT-based analyzer operating system that the terminal used be an EXORterm 150 or 155 Display Console) This communication will be achieved by means of a phantom or transparent serial link feeding from the terminal through the Bus State Analyzer Control Module, and then to the normal terminal input channel of the development system The logic onboard the Control Module determines whether the information traveling over the link is destined for the development system, the Bus State Analyzer or the terminal This allows the operating system or the user's software to run in the development system while analysis is being performed Additionally, a means is provided for the BSA to operate in a stand alone mode with only a terminal connected

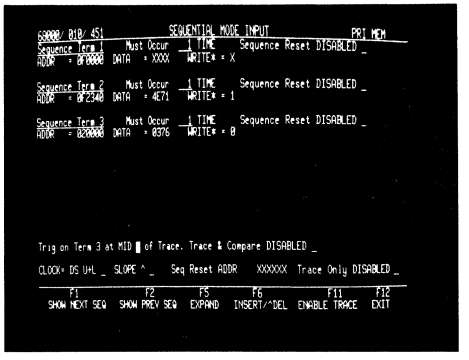
The BSA is simple to program through the use of the EXORterm's function key capability A menu defining the function keys is displayed at the bottom of the CRT Initially, (after power-up) when the BSA is talking to the system terminal, the analyzer displays a message regarding the result of the internal self test After the self test has been completed, the BSA may be programmed to a specific set of trace conditions

SET MODE



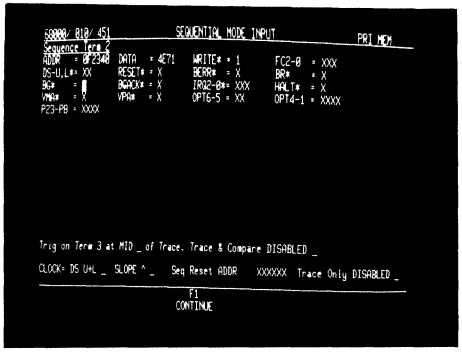
All operations of the BSA are controlled by the EXORterm function keys Here, the operating mode is being selected

SEQUENTIAL MODE INPUT



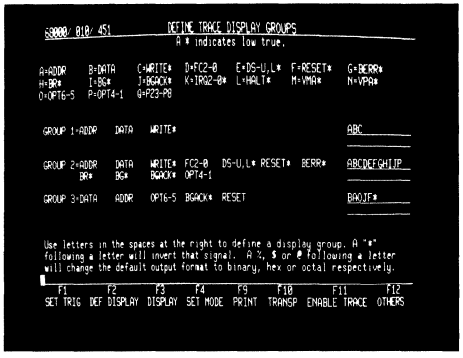
To control trace triggering, up to seven Sequence Terms, each consisting of 79 bits, may be entered Control options are selectable at the bottom of the screen "Soft" keys allow easy user control of display data and BSA operation

SEQUENTIAL MODE INPUT (Sequence Term 2)



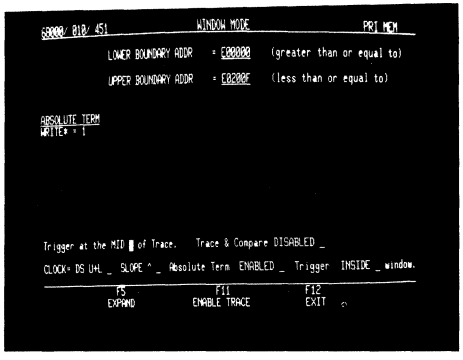
Sequence Terms may be "expanded" to display all 79 bits. Those left as "don't cares" are not normally displayed. Standard MC68000/MC68010 mnemonics are used for signal labels.

DEFINE TRACE DISPLAY GROUPS



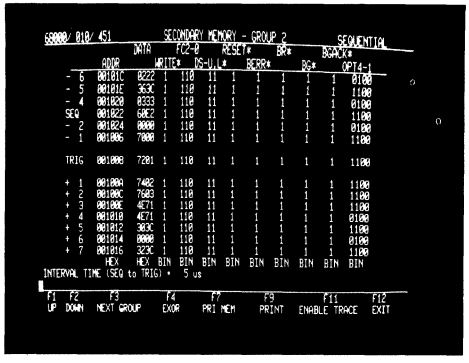
The 79 signals may be grouped and formatted according to user needs. Group 1 defaults to the most used signals. Groups 2 and 3 can be viewed when the user requires additional signals or different grouping.

WINDOW MODE



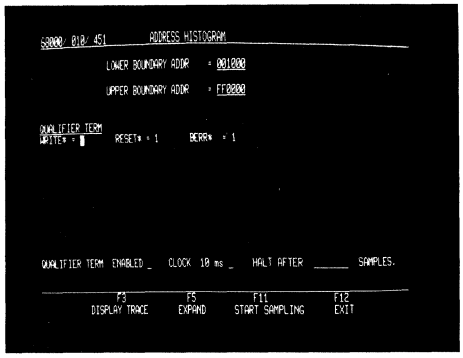
Address boundaries have been set. If a program strays into this area the BSA will trigger.

SECONDARY MEMORY — GROUP 2



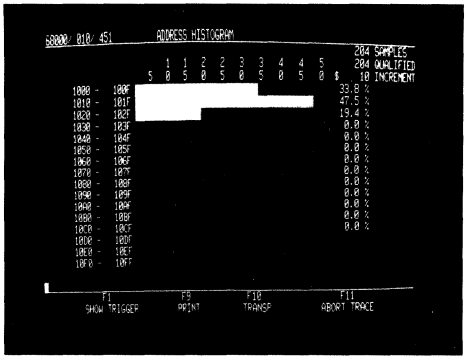
Typical trace display with interval time

ADDRESS HISTOGRAM (Set Up)



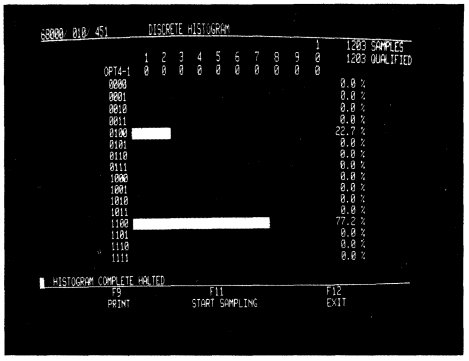
This histogram display has been set up to sample the top 1 MB of memory Samples taken will be considered only when WRITE*, DS1*, and DS0* are low, plus RESET* and BERR* are high Other sample qualifying terms are available

ADDRESS HISTOGRAM (Operation)



This software performance monitor displays relative percentage of accesses to the address ranges displayed Parameters are under the user's control Upper 500K bytes shown

DISCRETE HISTOGRAM (Operation)



This histogram shows the relative percentage of the indicated states of four of the optional user selectable lines

MC68000/MC68010/MC68451 Bus State Analyzer Personality Module Specifications

Characteristics	Specifications
Input Channels	79 total — 24 address, 16 data, 33 general purpose, 6 optional
Clock Inputs	6 total with selectable edges Free running 1 MHz (internally generated) External TTL Address strobe Data strobes DTACK or BUS ERROR (with switch-selectable delays to analyze system returning an "EARLY" DTACK)
Sample Rate	5 MHz maximum — supports zero wait state 16 MHz MC68000 or MC68010 operation
Sample Pulse Width	20 ns minimum
Power Required	+12 V from Control Module at 1 0 A
Environmental	0 to 50°C operating temperature
Module Dimensions Length × Width × Height	12 75 in × 9 00 in × 2 25 in (32 38 cm × 22 86 cm × 5 72 cm)

Bus State Analyzer Control Module — Specifications

Characteristics	Specifications
Trigger Outputs	One Scope Trigger from the final occurrence of each Sequence Term found One Scope Trigger from the final occurrence of Trigger Term
Trace Store Buffer	128 events deep by 79 bits wide with a second comparison array of equal size
Trigger Qualifiers	Up to 7 events from 79 qualifier lines, 1 level of magnitude comparison, any bits may be "don't cared"
Selective Trace Store	Single Combination (Trace only)
Trace Data Outputs	Selectable — binary, decimal, octal
Input Device Requirements	EXORterm 150, 155, 220 or VME/10, Transparent Mode supported with variable baud rates
Output Requirements	Transparent Mode provides RS-232C output with variable baud rates
Onboard Intelligence	MC6809
RAM (scratch)	2K × 8
Processor ROM	20K × 8 Operating System (Control Module)
Power Required	+5 0 Volts at 9 0 Amps +12 Volts at 1 0 Amp −12 Volts at 0 1 Amp
Environmental	0 to 50°C operating temperature
Module Dimensions Width × Height	14 5 in × 9 25 in (36 8 cm × 23 5 cm)

Ordering Information

Part Number	Description
M68BSA1-1	Real-Time MC68000/MC68010/MC68451 Bus State Analyzer Personality Module. Includes associated cabling, chip probes, and documentation. NOTE: A Control Module is required to complete the system
M68BSAC	Real-Time Bus State Analyzer Control Module for use with a VME/10 or EXORterm 155 and Bus State Analyzer Personality Modules User's Manual included
M68BSACE	Real-Time Bus State Analyzer Control Module in Standalone Enclosure for use with a VME/10 or EXORterm 155 and Bus State Analyzer Personality Modules User's Manual included
M68KBSA	Real-Time Bus State Analyzer User's Manual with Personality Module Appendixes (for extra copies)

Related Motorola Development Systems Products

Part Number	Description
M68SXD10155A	EXORterm 155 is a M6800-based display terminal and console. It consists of a Video Display Enclosure and an expanded Keyboard.
HDS-400 (configured with several part numbers)	HDS-400 Microprocessor Hardware/Software Development Station emulates the Motorola M68000 family of MPU's including the MC68000, MC68008 and the MC68010. The BSA Control Module may easily be mounted within the HDS-400 Control Station
M68KMACS	EXORmacs is a MC68000-based development system It utilizes the standard VERSAbus, and is designed to allow expansion to multiuser capability.
M68K101B1	VME/10 is a compact minicomputer workstation which allows expansion and customization through VMEbus and I/O channel card cages It includes a 10 MHz MC68010 MPU and MC68451 MMU, 384K bytes of RAM, 5 1/4" Winchester and floppy disk subsystems, a 15 inch video display, and an expanded ASCII keyboard.
MVMC682	VMC68/2 microcomputer system is built upon the MC68000-based VERSAmodule family of board-level products. It is designed to allow expansion and functional tailoring of both hardware and software system elements.

Real-Time VERSAbus State Analyzer Personality Module

Analyzer System Features:

- Connects Directly to the VERSAbus in the User's System or in the EXORmacs System
- 79 Channels by 128 States Trace Memory
- VERSAbus Signals Displayed Using Their Mnemonics
- Single Trace
- Trace and Compare with Previous Trace
- Trace a Selected State Only
- Trace Inside or Outside an Address Window
- Trace Through Reset
- Up to Seven Step Qualifier Sequence for Trigger
- Restart Trigger Qualifier Sequence on a Given Address State
- Trigger Pulse Outputs
- Interval Timer
- Address Histogram Display for Software Performance Analysis
- Discrete Signal Histogram Display
- Self-Test on Power Up
- Hard Copy Capability
- Easy Menu-Type Control by Function Keys of EXORterm 155
- EXORmacs and Remote Development Station Compatible

The Real-Time Bus State Analyzer is a highly intelligent diagnostic tool that is designed specifically for use with microprocessors It consists of a control module and one of several "personality" modules. The control module contains the Analyzer hardware, control software, and I/O ports The VERSAbus State Analyzer "personality" module interfaces to selected VERSAbus signals. The Analyzer stores data that appears on 79 different lines. Eight of these lines are available for external connection, and 71 are prewired to VERSAbus lines

Eight of these lines are available for external connection, and 71 are prewired to VERSAbus lines.

To facilitate the gathering of data from the bus, a set of qualified triggering modes are provided Continuous Trace Mode, Sequential Trigger Mode, and Window Trigger Mode

CONTINUOUS TRACE MODE samples signals and stores signal information continuously on each occurrence of the clocking signal It is primarily a default mode which the Analyzer automatically enters when power is first applied There are no qualifications for the Analyzer to begin gathering information, so it will always be storing the bus signal states This default mode is particularly useful when a sudden catastrophic failure occurs during a debugging session, before the user is able to configure the Analyzer It is very likely that the events leading up to the failure will be stored in the Analyzer's trace store buffer

SEQUENTIAL TRIGGER MODE requires that a series of events occur before the instrument triggers and starts to gather data, or conversely triggers and stops gathering data. Sequence Terms, as these events are called, must occur in order of specification, or triggering will not take place A Sequence Reset term can also be specified to reset the Analyzer and cause the instrument to begin looking for the sequence terms again. Sequential triggering will be most useful for debugging complex software including loops, nested subroutines and complex branches

WINDOW TRIGGER MODE provides a means of causing signal states to be stored if address accesses occur inside or outside of a particular address range. Both the upper and lower bounds of the range are programmable, and the size is variable from a single address to the full range of the memory map. Window triggering is useful for following programs that suddenly and unexpectedly leave the memory area in which they should be operating. It is also applicable for observing access violations in a multiple user environment.

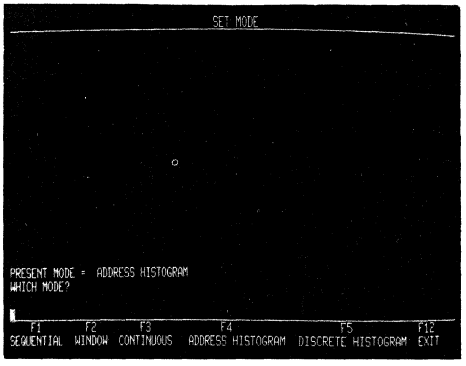
SOFTWARE PERFORMANCE HISTOGRAMS are also provided to give an indication of the relative frequency of memory accesses within a particular memory range, with the exact range specified by the user. This histogram provides a means of determining where a program spends the greatest amount of time The resulting information can then be used to compress inefficient code. A hardware performance histogram is provided to display the relative frequency of combinations of four user-selected signals within a user system

In order to service these triggering modes and provide a complete set of operating features, an MC6809 microprocessor is located on the control module with local intelligence running from an operating system based in ROM. This operating system provides the data analysis and formatting functions for the operator including the interface to the hardware sampling the bus

To reduce system redundancy, the terminal used by the operator to communicate with the development system will also serve to link him to the intelligence aboard the analyzer (it is a requirement of the CRT-based analyzer operating system that the terminal used be an EXORterm 150 or 155 Display Console). This communication will be achieved by means of a phantom or transparent serial link feeding from the terminal through the Bus State Analyzer control board and then to the normal terminal input channel of the development system. The logic onboard the analyzer determines whether the information traveling over the link is destined for the development system, the Bus State Analyzer or the terminal This allows the operating system or the user's software to run in the development system while analysis is being performed. Additionally, a means is provided for the analyzer to operate in a standalone mode with only a terminal connected

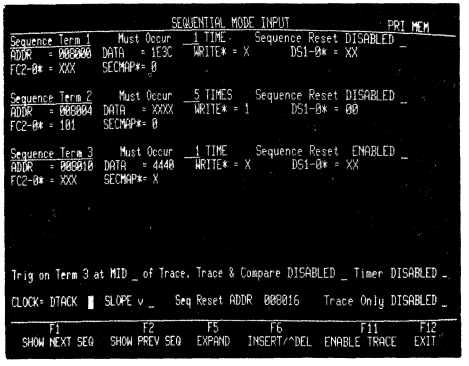
The analyzer is simple to program through the use of the EXORterm's function key capability A menu defining the function keys is displayed at the bottom of the CRT Initially, (after power-up) when the analyzer is talking to the system terminal, the analyzer will display a message regarding the result of the internal self test After the self test is completed, the analyzer may be programmed to a specific set of trace conditions

SET MODE



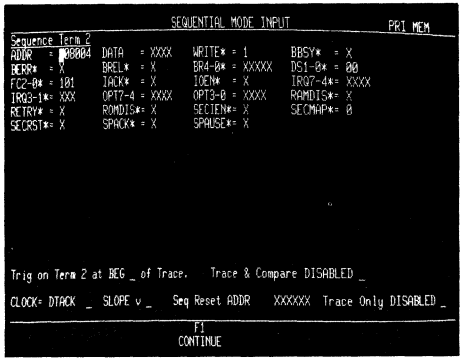
All operations of the Analyzer are controlled by the EXORterm function keys Here the operating mode is being selected

SEQUENTIAL MODE INPUT (Sequence Term 1)



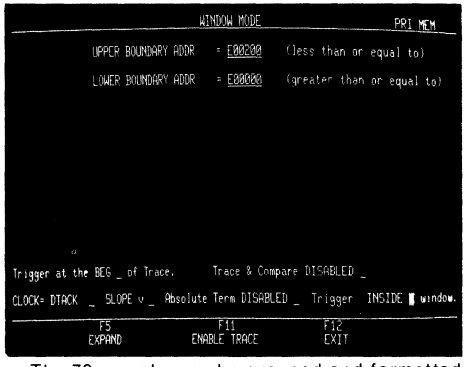
To control trace triggering, up to seven sequence terms each consisting of 79 bits may be entered. Control options are selectable at the bottom of the screen "Soft" keys allow easy user control of display data and Analyzer operation.

SEQUENTIAL MODE INPUT
(Sequency Term 2)



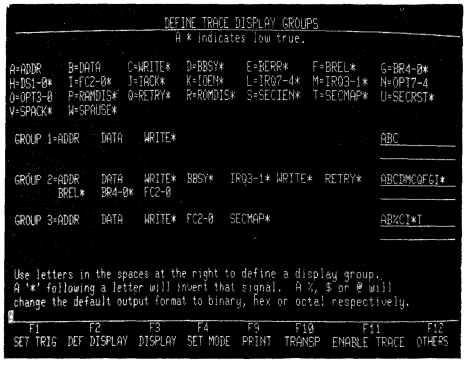
Sequence terms may be "expanded" to display all 79 bits Those left as "don't cares" are not normally displayed Standard VERSAbus mnemonics are used for signal labels

DEFINE TRACE DISPLAY GROUPS



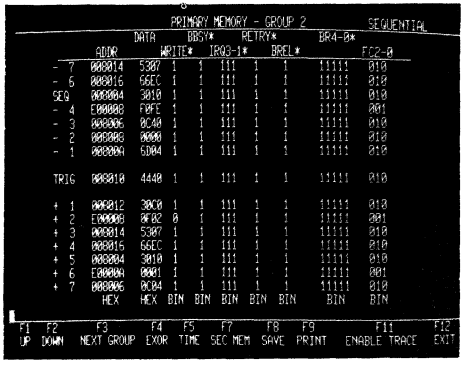
The 79 signals may be grouped and formatted according to user needs Group 1 is always displayed. Groups 2 and 3 are selected for display only when required by user for a more detailed display.

WINDOW MODE



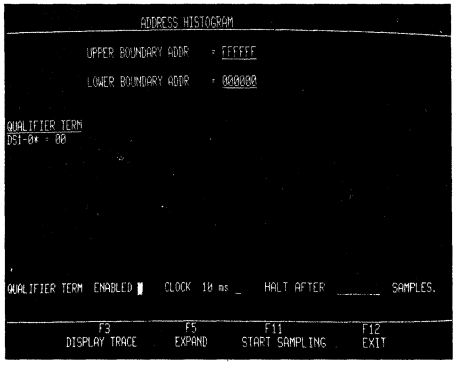
Address boundaries have been set If a program strays into this area the analyzer will trigger.

PRIMARY MEMORY — GROUP 2



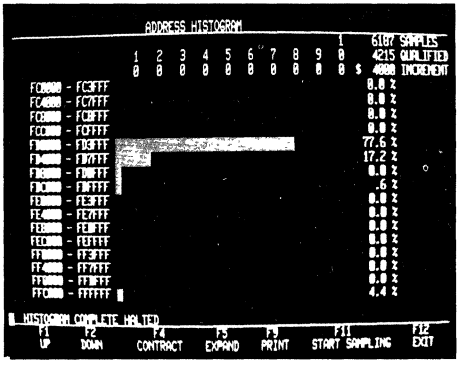
Typical trace display.

ADDRESS HISTOGRAM
(Setup)



Histogram display is set up to sample all of memory. Samples taken will be considered only when DS1* and DS0* are low. Other sample qualifying terms are available.

ADDRESS HISTOGRAM
(Operation)



This software performance monitor displays relative percentage of accesses to the address ranges displayed. Parameters are under the user's control.

VERSAbus STATE ANALYZER PERSONALITY MODULE
Specifications

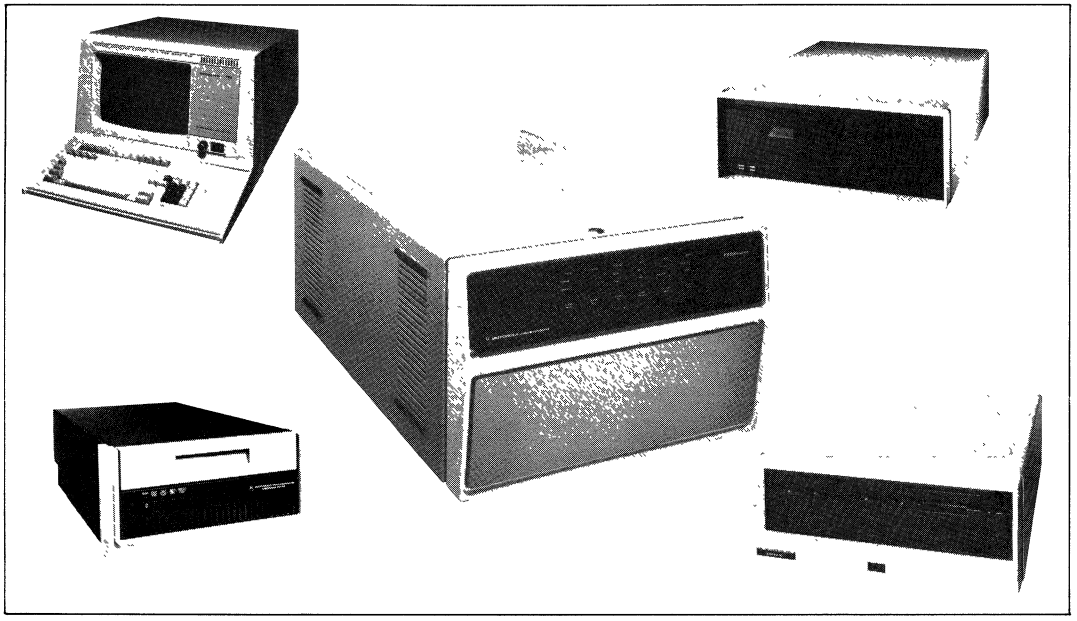
Input Channels	79 total — 24 address, 16 data, 31 general purpose, 8 optional
Clock Inputs	4 total with selectable edges — 2 bus signals, 1 internal unsynchronized, 1 external TTL
Clock Rate	5 MHz (200 ns period active edge to active edge)
Clock Width	20 ns minimum
Personality ROM	2K × 8 Personality
Power Required	+5 V from VERSAbus at 1.0 A
Environmental	0 to 50° operating temperature
Module Dimensions Length × Width × Height	14.5 in. × 9.25 in × 0.25 in. (36.8 cm × 23.5 cm × 0.6 cm)

BUS STATE ANALYZER CONTROL MODULE
Specifications

Input Channels	79 total — 24 address, 16 data, 39 general purpose
Clock Inputs	7 total with selectable edges — 5 personality board signals, 1 internal unsynchronized, 1 external TTL
Clock Rate	5 MHz (200 ns period active edge to active edge)
Clock Width	20 ns minimum
Trigger Outputs	1 Scope Trigger from the final occurrence of each sequence term found 1 Scope Trigger from the final occurrence of Trigger term
Trace Store Buffer	128 events deep by 79 bits wide with a second comparison array of equal size
Trigger Qualifiers	Up to 7 events from 79 qualifier lines, 1 level of magnitude comparison, any bits may be "don't cared"
Selective Trace Store	Single Combination (Trace only)
Input Device Requirements	EXORterm 150, 155 or 220 with RS-232 interface, Transparent Mode supported with variable baud rates
Output Requirements	Transparent Mode provides RS-232 output with variable baud rates
Onboard Intelligence	MC6809
RAM (scratch)	2K × 8
Processor ROM	20K × 8 Operating System (Control Module)
Power Required	+5.0 Volts at 9 0 Amps +12 Volts at 1 0 Amps -12 Volts at 0 1 Amps
Environmental	0 to 50° C operating temperature
Module Dimensions Length × Width × Height	14 5 in × 9 25 in × 0.25 in (36 8 cm × 23.5 cm × 0 6 cm)

Ordering Information

Part Number	Description
M68BSA5	Real-Time VERSAbus State Analyzer Personality Module. Includes associated cabling, and documentation. NOTE A control module is required to complete the system.
M68BSAC	Real-Time Bus State Analyzer Control Module for use with Bus State Analyzer Personality Modules. User's Guide included.
M68KBSA(D1)	Real-Time Bus State Analyzer User's Guide with Personality Module Appendixes



EXORmacs M68000 DEVELOPMENT SYSTEM HOST

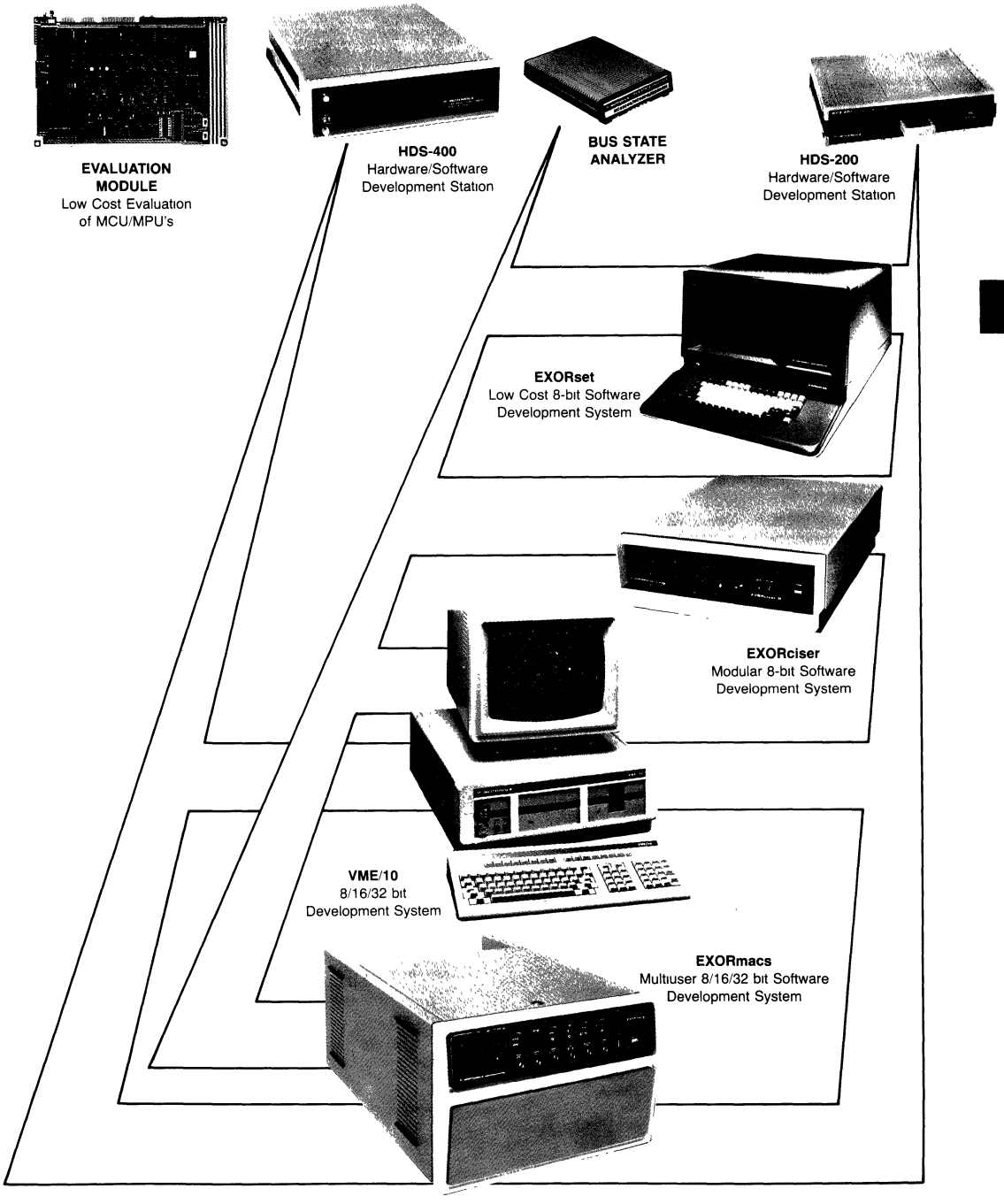
The EXORmacs Development System is a state-of-the-art instrument for designing and developing advanced 8- and 16-bit microprocessor based systems using Motorola M6800 and M68000 families of microprocessors, microcomputers, and peripheral components

It offers flexible operating environments with a choice of VERSAdos or System V/68 Operating Systems for systems development With VERSAdos, it is a true realtime multi tasking development system capable of fully supporting Motorola's 8/16/32-bit MPU and MCU product lines With System V/68, it is a development system that offers a powerful operating system that contains all system calls, commands, and utilities included in AT&T Technologies' UNIX System V Operating System, except those features dependent upon hardware not supported by Motorola

Coupled with the Motorola HDS-400 Microprocessor Hardware/Software Development Station it is also ideally suited for developing applications using the VERSAmodule and VMEmodule families of 16-bit board level application products and accessories

Designed for flexibility and ease of use, the EXORmacs Development System takes advantage of the power and features of the MC68000 microprocessor unit (MPU). It reduces cost and development time by incorporating features which support 16-bit and future 32-bit microprocessor designs, as well as providing high-level language support through 'C, Pascal and FORTRAN Compilers With an appropriate number of accessories, such as terminals, multiple-channel communications modules and hardware development stations, up to eight users may simultaneously develop and debug M68000 programs

MOTOROLA FAMILY OF MICROCOMPUTER DEVELOPMENT AIDS



MOTOROLA 16/32-BIT MICROCOMPUTER SYSTEM COMPONENTS

EXORmacs M68000 DEVELOPMENT SYSTEM HOST

5

The EXORmacs Development System is intended to aid in the design of systems based on the MC68000 16-bit microprocessor. But it is also compatible with previous development tools and microprocessors, and, more importantly, is designed to support Motorola's line of compatible 8-16-32 bit processors based on the M68000 architecture.

In readiness for these 32-bit devices, the EXORmacs system incorporates a 5.0 MHz bus structure called VERSAbus with 32 address lines and 32 data lines (see Figure 1). On this bus reside the modules that constitute the system hardware. Besides the MC68000, the central-processing-unit board contains memory management logic to smooth software development. A bus arbiter scheme allowing multiple processors to share the system bus and an intelligent peripheral controller concept has been used in order to further boost system performance.

FUNCTIONAL FEATURES

- Software Development System for MC68000
- Powerful Memory Management Unit
- Up to eight simultaneous users
- Flexible Operating Environments
  - — System V/68 O/S
  - — VERSAdos O/S
- Background Batch Capability
- Multi-Processor Bus Arbitration
- Host for Hardware Development
  - — 16 Bit: HDS-400
  - — 8 Bit: HDS-200 (VERSAdos O/S only)
- Symbolic DEbug w/VERSAdos
- Self-Testing using Diagnostic Firmware
- Intelligent Display Console with System Function Keys
- Versatile Mass Storage Options
  - — 1 Mbyte Floppy Disk
  - — 16, 50 or 96 Mbyte Hard Disk
- Industry-Standard VERSAbus
- Down-Line Load Host Serial Port
- Key Lock Chassis Security
- Front Panel Status/Error Indication
- Provisions for Future 32-Bit Microprocessors

SYSTEMS DEVELOPMENT AND INTEGRATION

The initial stages of developing a microprocessor-based system normally involve two parallel, rather independent, efforts One is the hardware design — the other the software design These efforts are frequently accomplished by two different teams of personnel.

The EXORmacs Development System, when used as the host along with Motorola's new line of Hardware Development Stations (HDS), simplifies the design process because of its ability to bring the hardware and software development processes into intimate relationship with each other throughout the development cycle.

Using the multiuser feature of the EXORmacs System, a team of designers can accomplish the software development phase of the design cycle. The resultant object code is then ready for execution on the prototype hardware, and can be downloaded into the appropriate HDS.

The HDS, in conjunction with the EXORmacs software development system, provides a complete hard-

FIGURE 1 — VERSAbus DESCRIPTION

A 32-BIT BUS SYSTEM

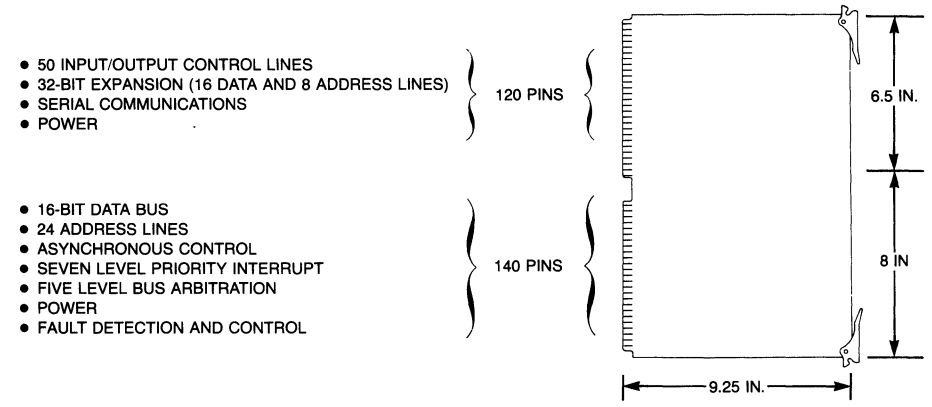
VERSAbus is designed to serve as a comprehensive foundation for 8-bit to 32-bit microprocessor architectures having five MHz data transfer rates. It is specifically intended to serve industrial control, communications, and general-purpose business applications and to allow for system architectures involving multiple processors A final objective of VERSAbus is to exploit to the fullest the latest computer and semiconductor technologies but without sacrificing ease of use.

To satisfy these objectives, the following features are provided by VERSAbus:

- A very fast bus cycle time
- Asynchronous, bidirectional operation.
- 8-, 16-, or 32-bit data transfers, with single-byte designation possible in 16- and 32-bit data transfers.
- 32 address lines for direct access to four billion words of memory.
- Direct memory access and multiprocessor support
- Seven-level priority interrupt control
- Five-level daisy-chained bus arbitration
- Up to 50 peripheral input/output lines with mating ground returns.
- Standard +5 V and +12 V power for logic as well as +15 V for process control and standby power support.
- Serial communications.
- Distinct I/O mapping.
- Bus error and retry signals.
- Separate analog ground.

To accommodate the signal lines required for these features, VERSAbus uses 260 pins in two connectors, as shown in the diagram Why should a microprocessor bus be so large? First, as densities increase in semiconductor devices, greater amounts of memory and peripherals will be controlled from a microprocessor and wider data widths will be required.

Secondly, since one of the goals of VERSAbus is to provide a comprehensive basis for future microprocessor systems, it must be capable of supporting the largest envisioned applications and single-board designs.



ware and software development system for the Motorola families of microprocessors. Two major factors contribute to the HDS's usefulness as a systems development tool The first is the ability to serve as a fully functional substitute for the selected microprocessor or microcomputer in the user's target system. By plugging the HDS emulator into the socket on the prototype hardware, it allows efficient testing of hardware as well as software. The second is its ability to interface with the Real-Time Bus Analyzer which speeds the debugging process and optimizes program code.

The Motorola Real-Time Bus State Analyzer (BSA) is a development tool which allows monitoring at different points in a system Interfacing through the system bus, the BSA tracks events occurring on each line of the bus, storing the information for later analysis and interpretation.

The emulator can be organized and software run prior to availability of hardware. This allows an early start on the debugging process. As hardware changes occur, software updating and debugging is readily accomplished. Moreover, with the HDS it becomes economically feasible to test alternate design approaches in order to determine the best solution.

INTEGRAL EXORmacs BASIC SYSTEM HARDWARE

The functional block diagram of the EXORmacs System is shown in Figure 2. The following paragraphs give general descriptions of the components incorporated in the basic system.

EXORmacs CHASSIS

The chassis provides the housing for the development system card complement, a switching power supply, cooling fans, and a front control panel. The VERSAbus/card cage accommodates 15 cards A switching power supply is included to handle all power needs for system configuration. The front panel provides to the user both manual control input capability and visual indication of current system status, along with any existing error conditions.

DEbug MODULE

There are three ports associated with this module. The two RS-232C serial ports are used for connections to the EXORterm 155 Display Console and a host computer. The second of the RS-232C ports, which is configured for a host computer, can be reconfigured for connection to a second EXORterm 155 Display Console. The third port is used for the printer This module also contains resident firmware (MACSbug), which provides a vehicle for system start-up and control

MPU/MMU MODULE

This module is the central intelligence of the EXORmacs System. It contains a four-segment Memory Management Unit (MMU) and diagnostic firmware. The MMU provides the multitasking operating system protection from the user programs, and aids in allocating memory to each task. The multitasking operating system allows tasks to be executed concurrently, which aids in decreasing the time associated in the user program development cycle. An example of this advantage is an assembly in progress that requires the printer, but allows the use of the CRT display console for editing by other modules.

512K BYTE RAM MODULE

This 512K Byte Memory utilizes 64K x 1 HMOS RAM technology for operating system and support software storage. High reliability information storage is provided through on-board error detection and correction circuitry. Optionally, up to seven more modules may be added to increase system performance. Total on-line RAM can be up to eight Mbytes.

UNIVERSAL INTELLIGENT PERIPHERAL CONTROLLER (UIPC)

The UIPC is one module of a two-module system called a Universal Disk Controller (UDC). The UIPC has an MC68120-based microcomputer and the necessary operating firmware and hardware interface to control both hard disk and floppy disk drives through the Disk Interface Module (the second module of the UDC). The UDC reduces disk I/O for the operating system to a matter of simple, short command packets. The UDC performs the remaining tasks of accessing the disk(s), setting up the Direct Memory Access (DMA) controller, and transferring the data. This distributed processing approach provides increased performance as well as standardization of the I/O channel interface, which minimizes system impact when I/O device configuration changes occur.

DISK INTERFACE MODULE (DIM)

The DIM interfaces to the UIPC through a standard DMA channel. This module provides the control and interface for the floppy disk and the hard disk. Two compatible Storage Module Drives (SMDs) are supported by the SMD interface, and up to two drive units are supported by the floppy disk channel.

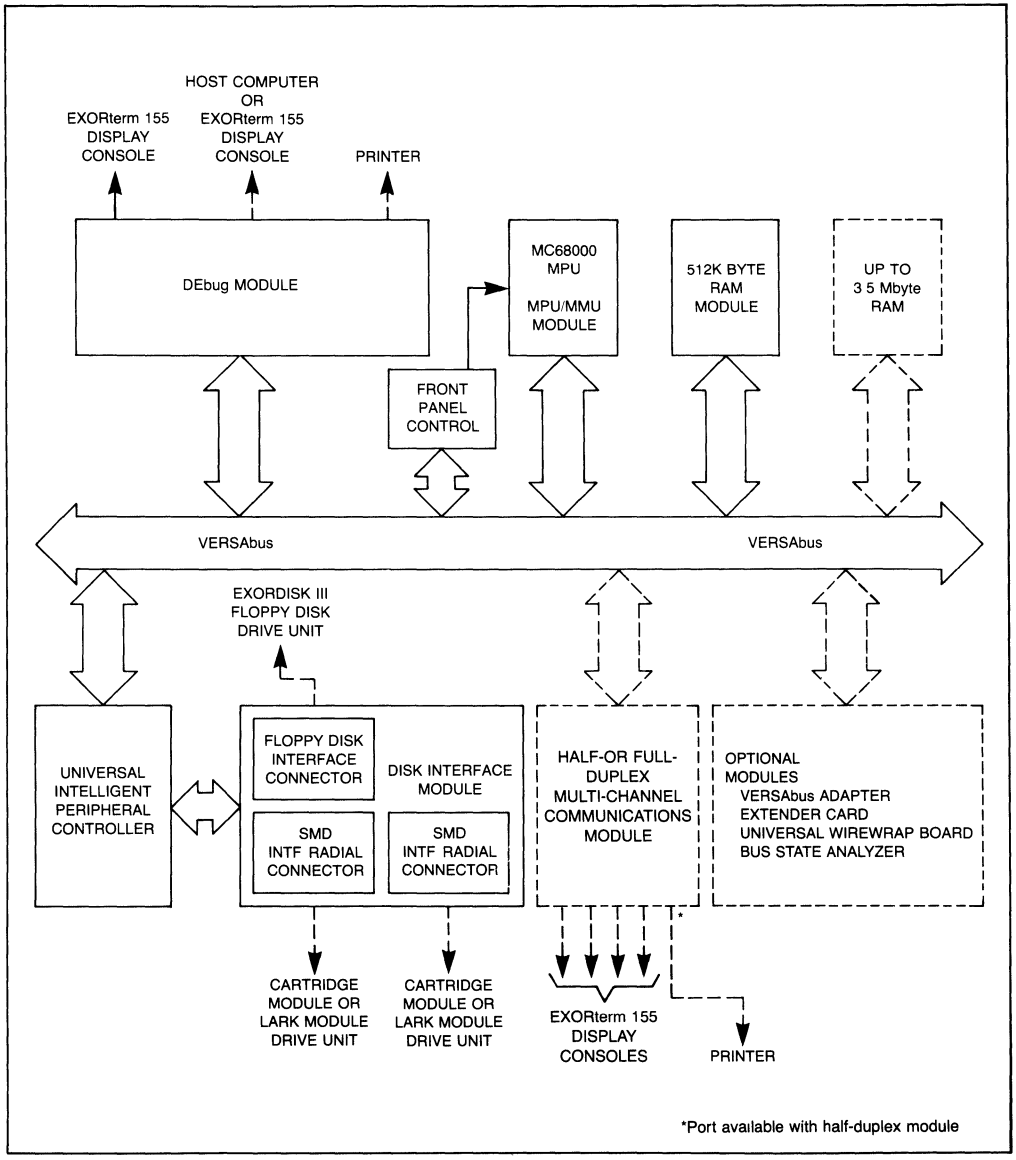
EXORterm 155 DISPLAY CONSOLE

This console is an intelligent CRT station providing the user with enhanced editing and symbolic debugging capabilities through screen formatting, cursor control, and soft key entry. The console also provides control of the display attributes, communication facility, terminal switch/indicator control, and keyboard inputs. In conjunction with control and application task firmware, the microexecutive firmware coordinates the functions of the display console in the system.

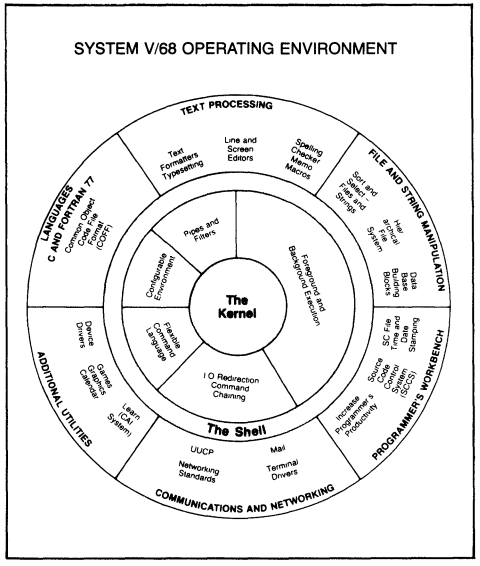
MACSbug DEbug MONITOR

MACSbug is a resident firmware debug monitor. It has its own I/O and requires only a terminal connected to port one. MACSbug is initiated by the test and diagnostic firmware during a power-up sequence, provided the ENABLE/DISABLE key switch is in the ENABLE position. It is used strictly in a single-user environment.

FIGURE 2 — FUNCTIONAL BLOCK DIAGRAM OF EXORmacs SYSTEM HARDWARE (DASHED LINES INDICATE OPTIONAL PRODUCT)



SYSTEM V/68
OPERATING SYSTEM SOFTWARE



SYSTEM V/68 OPERATING SYSTEM

The System V/68 Operating System is the standard UNIX-derived Operating System for the M68000 family of microprocessors. It offers a small compact kernel, which provides process scheduling and I/O facilities to all programs. In addition, a powerful command shell for interactive system controls and an extensive set of utility programs for many tasks, such as program development, text processing, electronic mail, and networking support are included.

TEXT PROCESSING

Text processing utilities include the *ex/vi* full-screen editor, NROFF and TROFF text formatters, a spelling checker, and programs for formatting tables and mathematical equations. The *ex/vi* editor supports a large number of existing terminals, including the Motorola EXORterm 155, through the use of the *termcap* terminal data base. *Termcap* entries for new terminals may be added by the user.

INSTRUMENTATION SUPPORT

Communications support for the Motorola HDS-400 Hardware Development Station is included in System V/68. This provides customers with the full system development capability (both hardware and software) that they have come to expect from Motorola.

LANGUAGES

As an integral part of System V/68, C Language is offered. C Language has developed into one of the most popular commercial programming languages, and is used frequently in developing portable application software. System V/68 offers significant enhancements to C Language, along with several new language utilities. CXREF, a new cross reference program, and CFLOW, a new flow analysis program, are just two of the new utilities offered. System V/68 also includes a FORTRAN 77 compiler as well as an M68000 assembler and linker/loader.

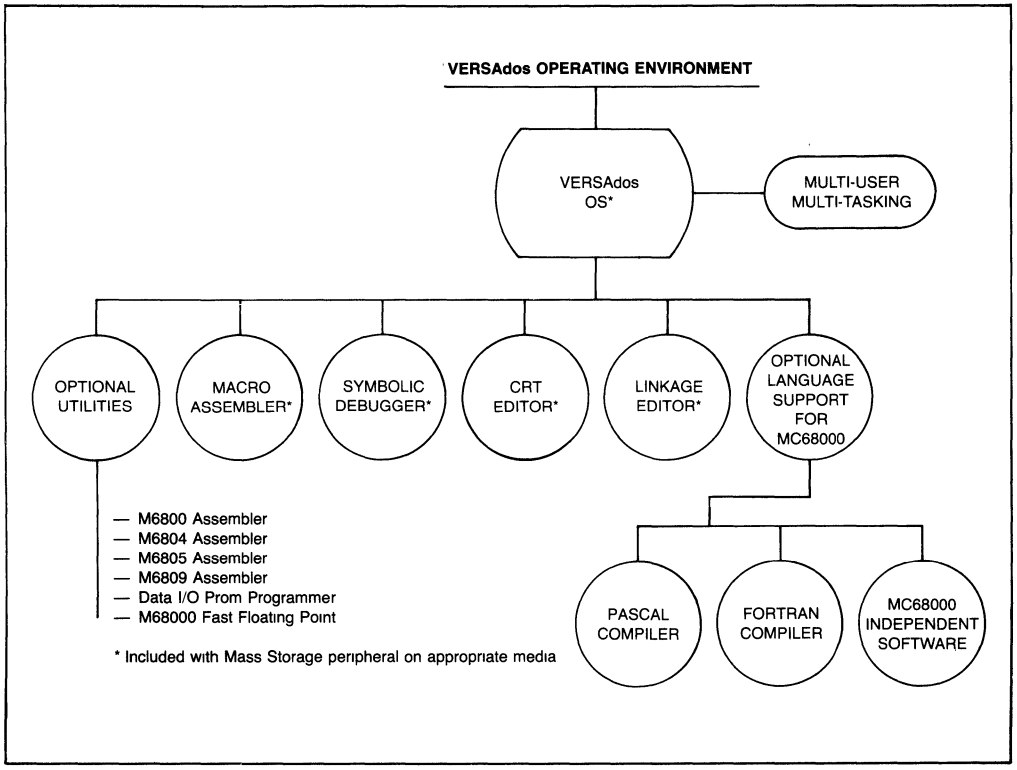
COMMUNICATIONS

System V/68 utilities provide support for electronic mail, communications, and networking. Electronic mail allows users to communicate with one another, using the system as a mailbox or as a bulletin board. The communications utilities allow a System V/68 user to communicate to mainframe computers. Networking support allows several computers to be linked together, either through dedicated links or by dial-up telephone connections. With these utilities, Motorola development system users can communicate with one another In addition, target systems may be developed with similar capabilities. In System V/68, interprocess communications routines have been added. These include shared memory, messages, and semaphores. Also included is an IPC/remove command which removes message queues, semaphores, and shared memory identifiers from the system.

PROGRAMMER'S WORKBENCH

The Programmer's Workbench utilities support the development of large software systems in a professional manner. They include the Source Code Control System (SCCS), which provides facilities to store, update and retrieve all versions of source code modules; YACC, which generates parsers; LEX, which builds lexical analyzers; and other utilities which enhance programmer productivity and the quality of work.

VERSAdos OPERATING SYSTEM AND DEVELOPMENT TOOLS



VERSAdos AND STANDARD DEVELOPMENT SOFTWARE TOOLS

VERSAdos OPERATING SYSTEM

VERSAdos is a multitasking, multiprogramming system. Programs execute in dynamically-assigned, variable-length segments with read/write privileges. Instructions and data are located in separate memory segments.

The heart of VERSAdos is a real-time executive which provides task services and supports memory management. It also allows inter-task communication, provides exception monitor facilities, and handles system interrupts.

The input/output (I/O) subsystem of VERSAdos supports device independence, logical I/O, overlapped computation, and physical I/O. Both sequential and random record access are supported by VERSAdos.

A powerful file management system supports three file structures — contiguous, variable length, and index sequential. Other features include disk and file protection, and dynamic file access permission.

RESIDENT STRUCTURED ASSEMBLER

The M68000 resident structured macro assembler translates source statements into relocatable machine code, assigns storage locations to instructions and data, performs auxiliary assembler actions designated by the programmer, and optionally produces a cross reference listing. The M68000 resident assembler includes macro and conditional assembly capabilities, plus certain "structured programming" control structures such as "for, repeat, while" (loops) and "if-then, if-then-else" (conditional branches).

SYMBOLIC DEBUGGER

The SYMbug/A program is used to debug other programs whose source code may have been written in Motorola-provided assembler language for execution on the MC68000 The language processors, in cooperation with the linkage editor, supply information to SYMbug/A This permits the user to describe the debugging requirements to SYMbug/A in terms close to the language in which the source program was written. SYMbug/A allows the user to debug in symbolic terminology SYMbug/A allows the user to perform the following:

- Examine, insert, and modify program elements such as instructions, numeric values, and coded information (i.e., data in all its representations and formats)
- Control executing, including the insertion of breakpoints into a program and requests for breaks or changes in elements of data.
- Trace execution by displaying information at designated points in a program
- Search programs and data for specific elements and sub-elements
- Create macro commands allowing user-defined formats and commands

CRT TEXT EDITOR

The CRT-oriented text editor provides the capability to create and modify source programs. The editor supports both command-and-page editing, utilizing the cursor capability, control characters, and function keys of the EXORterm 155 display console to insert, alter, or delete characters and lines within a user file

LINKAGE EDITOR

The linkage editor provides the capability of converting one or more separately-compiled object units into a loadable object module file. The editor determines segment attributes, calculates address space, searches libraries, resolves external references, relocates object code, and issues error messages. The editor prints a report that contains a module map, a table of externally-defined symbols, and any unresolved or duplicate defined symbol.

VERSAdos OPERATING SYSTEM OPTIONAL LANGUAGE SUPPORT

OPTIMIZING PASCAL COMPILER

Pascal is a high level, user oriented language for MC68000 MPU, based on the language as defined by Niklaus Wirth. Pascal is a highly structured language which promotes good programming techniques, is self-documenting, and its user-oriented statement forms simplify program writing. Extensions provided by Motorola include: Address specification for variables, alphanumeric labels, string types, exit, non-decimal integers, runtime error checking, runtime file assignment, and separate compilation and linking. The optimizer produces compact efficient code. Library routines include both IEEE floating point and a single precision fast (multiply 44 microseconds) floating point

FORTRAN COMPILER

The FORTRAN compiler translates source programs written in FORTRAN into M68000 machine language, using the EXORmacs Development System. FORTRAN is a high-level programming language widely used for scientific and engineering problem solving with features also useful for certain business-related applications. The FORTRAN compiler is the 1977 ANSI subset standard In addition, extensions designed specifically for microprocessor applications are included such as bit manipulations and assembly language interface

MC68000 INDEPENDENT SOFTWARE

A broad range of independent software suppliers support Motorola's 16/32 bit microprocessors. See the latest issue of the "Motorola Microprocessor Software Directory (BR126R1)" for a list of applicable software and independent suppliers

EXORmacs SYSTEM PERIPHERALS AND I/O

OPTIONAL EXORmacs SYSTEM PERIPHERALS

One EXORterm 155 is an integral part of the basic EXORmacs System. Up to seven more may be added for a total of eight simultaneous users.

EXORdisk III FLOPPY DISK DRIVE

This system consists of two disk drives and power supply. The disk drive unit is a high speed, one Mbyte random access storage unit, utilizing a flexible disk cartridge as the storage media The disk control and interface are provided by the Universal Disk Controller, which consists of the Disk Interface Module and the Universal Intelligent Peripheral Controller A total of two Mbytes may be added to the basic EXORmacs System.

CARTRIDGE MODULE DRIVE UNIT (96 Mbyte)

This drive consists of a front loading removable 16-Mbyte cartridge and a fixed pack which provides 80 Mbytes of fixed capacity It is a high-performance, random-access, mass-memory device with average access time of 30 ms and data transfer rate of 9.67 MHz. Optionally, an additional 96-Mbyte drive may be added to an existing system, resulting in a total of up to 192 Mbyte of rotating hard disk storage

LARK MODULE DRIVE UNIT (16 Mbyte or 50 Mbyte)

These drives consist of a front loading removable 8 or 25-Mbyte cartridge and a fixed pack which provides another 8 or 25 Mbytes of fixed capacity This unit is a high performance random access, mass memory device with average access time of 42 ms and data transfer rate of 9 67 MHz Two drive units may be connected to a basic EXORmacs System giving a total of 32 or 100 Mbytes of hard disk storage

NOTE Cartridge module drive units and Lark module drive units may be used together on the same system

OPTIONAL EXORmacs SYSTEM INPUT/OUTPUT

SERIAL CHANNELS

The EXORmacs basic system supports two full duplex serial channels through the DEbug Module System V/68 and VERSAdos support up to eight channels (users).

HALF-DUPLEX MULTI-CHANNEL COMMUNICATIONS MODULE (MCCM)

This is an optional module that interfaces four asynchronous serial communication devices and one printer to VERSAdos through the VERSAbus. This module utilizes the Intelligent Peripheral controller (IPC) to interface directly with VERSAbus, while the serial ports and printer port interface with the IPC bus The IPC contains an MC6801 which provides high thoughput via the I/O ports and also provides self-contained module diagnostics The data channel is a high speed direct memory access controller which transfers data to and from VERSAbus without MC68000 microprocessor intervention

FULL-DUPLEX MULTI-CHANNEL COMMUNICATIONS MODULE

This is an optional module that interfaces four asynchronous serial communication devices to System V/68 Operating System through the VERSAbus.

RECOMMENDED EXORmacs CONFIGURATION GUIDE

STEP 1

Specify EXORmacs Basic System.

ORDERING INFORMATION

Description	Part Number
Basic System includes EXORmacs Chassis, DEbug Module, MPU/MMU Module, 512 K Byte RAM Module, Universal Intelligent Peripheral Controller (two modules) and an EXORterm 155 Display Console.	M68KMACS

5

STEP 2

Select Mass Storage

<table>
<tr><th>NUMBER OF USERS</th><th>1 MB FLOPPY</th><th>SECOND 1 MB FLOPPY</th><th>16 MB DRIVE 8+8</th><th>50 MB DRIVE 25+25</th><th>50 MB DRIVE + 96 MB DRIVE</th><th>192 MB DRIVE 96+96 DRIVE</th></tr>
<tr><td>1</td><td>Most economical single-user system.</td><td rowspan="2">OK for Floppy Disk-Based Systems</td><td rowspan="2">Minimum Recommended Mass Storage</td><td rowspan="4">Recommended</td><td colspan="2" rowspan="3">Additional Capacity Suggested for Future Growth and/or Large User Data Bases</td></tr>
<tr><td>2</td><td rowspan="6">Recommended only for Hard Disk-Based Systems that Desire Auxiliary Floppy Disk Storage</td></tr>
<tr><td>3</td><td rowspan="5">Not Recommended</td></tr>
<tr><td>4</td><td rowspan="2">Not Recommended</td></tr>
<tr><td>5</td><td colspan="2" rowspan="3">Recommended</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td><td rowspan="2">Limited Mass Storage Space Per User</td></tr>
<tr><td>8</td></tr>
<tr><td>First* Unit</td><td>M68KFD1102</td><td>—</td><td>M68KHDD16-1</td><td>M68KHDD50-1</td><td>M68KHDD50-1</td><td>M68KHDD96-1</td></tr>
<tr><td>EXP Unit</td><td>—</td><td>M68SFDU1102E</td><td>M68KHDE16-1</td><td>M68KHDE50-1</td><td>M68KHDE96-1</td><td>M68KHDE96-1</td></tr>
</table>

*Includes VERSAdos Operating System and Development Tools. For SYSTEM V/68 (only) applications, select M68KHDE96-1 as both "First Unit" and "Exp Unit."

ORDERING INFORMATION

For use with System V/68 and M68000 Development Software Tools:

Note: Does *Not* Include System V/68

Description	Part Number
1 MB Floppy Disk	M68SFDU1102E
16 MB Drive	M68KHDE16-1
50 MB Drive	M68KHDE50-1
96 MB Drive	M68KHDE96-1

Includes VERSAdos and standard M68000 Development Software Tools: Macro Assembler, Linkage Editor, Editor and SYMbug/A.

Description	Part Number
1 MB Floppy Disk	M68KFD1102
16 MB Drive	M68KHDD16-1
50 MB Drive	M68KHDD50-1
96 MB Drive	M68KHDD96-1

System Expansion for use with *either* VERSAdos or System V/68 Operating Systems.

Description	Part Number
1 MB Floppy Disk	M68SFDU1102E
16 MB Drive	M68KHDE16-1
50 MB Drive	M68KHDE50-1
96 MB Drive	M68KHDE96-1

5

STEP 3

Select Additional Software Development Tools

ORDERING INFORMATION

SYSTEM V/68 OPERATING SYSTEM SOFTWARE

Description	Part Number
System V/68 Software Supplied on CMD Cartridge System V/68 Includes the following Object Code Modules — M68000 System V/68 OS — M68000 C Language Compiler Assembler, and Linker — Instrumentation Support Utilities — Full Set of System V/68 Documentation — OEM Configuration Guide — Users Guide	M68NNCBSV

VERSAdos SOFTWARE DEVELOPMENT TOOLS

Description	Part Number		
	FLOPPY	CARTRIDGE	16 MB LARK
FORTRAN Compiler	M68K0FORTRN	M68K0FORTRNH	M68K0FORTRNG
Pascal Compiler	M68K0PASCAL	M68K0PASCALH	M68K0PASCALL
M6800 Assembler	M6800XASM	M6800XASMH	M6800XASML
M6804 Assembler	M68V4EBASM04	M68V4CBASM04	M68V4LBASM04
M6805 Assembler	M6805XASM	M6805XASMH	M6805XASML
M6809 Assembler	M6809XASM	M6809XASMH	M6809XASML
Data I/O Prom Programmer Software	M68KDIOPP	M68KDIOPPH	M68KDIOPPL
M68000 Fast Floating Point	M68KFFP	N/A	N/A

STEP 4

Select Number of Concurrent Terminals (See Step 1)

NUMBER OF USERS	TERMINALS	MULTI-CHANNEL COMMUNICATIONS MODULES	
		MCCM #1	MCCM #2
1	First EXORterm 155* included in BASIC System		
2	Order 2nd EXORterm 155		
3	Order 3rd EXORterm 155	USE FIRST MCCM	
4	Order 4th EXORterm 155		
5	Order 5th EXORterm 155		
6	Order 6th EXORterm 155		
7	Order 7th EXORterm 155		USE SECOND MCCM
8	Order 8th EXORterm 155		

ORDERING INFORMATION

Description	Part Number
Full Duplex Multi-Channel Communications Module (4 channel, RS-232C). Provides serial ports for adding up to four additional user terminals for System V/68 Operating System on an EXORmacs M68000 Development System Host. NOTE: The basic EXORmacs System supports two user terminals.	M68KV7
Half Duplex Multi-Channel Communications Module (4 channel, RS-232C). Provides serial ports for adding up to four additional user terminals for VERSAdos Operating System.	M68KMCCM

STEP 5

Select memory (total bytes on-line RAM)

NUMBER OF USERS	512 K	640 K	768 K	896 K	1 MBYTE
1	Minimum System				
2	Recommended	Improves System Thruput			
3					
4					
5					
6	Not Recommended	Recommended			
7					
8					

Available for System Expansion

Improves System Thruput

ORDERING INFORMATION

Description	Part Number
128 K Bytes Dynamic RAM w/parity	M68KVM10-3
256 K Bytes Dynamic RAM w/ECC	M68KVM11-1
512 K Bytes Dynamic RAM w/ECC	M68KVM11-2
1024 K Bytes Dynamic RAM W/Parity	M68KVM12

STEP 6

Select Optional Assemblies

ORDERING INFORMATION

Part Number	Description
M68KVAM	VERSAbus Adapter Module
M68KEXTM	VERSAbus Extender Module
M68KWW	VERSAbus Wirewrap Module
M68BSA5	Real-Time VERSAbus State Analyzer Personality Module
M68HDS201	HDS-200 Control Station (Requires Emulator)
M68KHDS400	HDS-400 Control Station (Requires Emulator)
M68CART	Spare Cartridge for Cartridge Module Drive
MLD16CART	Spare Cartridge for LARK Module Drive (16 MB)
MLD50CART	Spare Cartridge for LARK Module Drive (50 MB)
M68KMACSRK	EXORmacs Rack Mount Kit

NOTE: Above Options Refer to U.S Availability. In Other Countries Alternative Configurations may be offered

STEP 7

PRINTER SELECTION

The Debug Module and the optional Multi-Channel Communications Module have parallel printer ports to which a Centronics type compatible printer supplied by the user should be connected Printer desirability depends on total printer output However, a 180 cps printer is recommended as a minimum for up to two users A high speed line printer (400–600 lpm) is recommended for more than two users

NOTE: Refer to page 19 for a list of applicable EXORmacs support literature

EXORmacs SYSTEM CHASSIS

CHARACTERISTIC	SPECIFICATION
Dimensions Height Width Depth	 12.625 inches (32.0 cm) (with bottom feet) 19.5 inches (49.5 cm) (sides with feet) 28.75 inches (73.0 cm)
Weight	45 pounds (20.4 kg)
Input Voltage	90 to 126.5 Vac, 47 to 63 Hz, single phase, 6.0 A, 600 watts (10 A fuse) 103.5 to 126.5 Vac, 47 to 63 Hz, single phase, 6.0 A, 600 watts (10 A fuse) 208 to 240 Vac, 50 Hz, single phase, 3.0 A, 600 watts (5.0 A fuse)
Temperature Storage Operating	 −50°C to 70°C 0°C to 45°C
Humidity	0–85%
Card Loading	15 modules max. (min. card slot separation .90 inch, center to center) One card slot provides 1.4 inches to accommodate wirewrap card.
dc Power Supply	+5.0 V at 75 Amps +12 V at 8.0 Amps −12 V at 8.0 Amps

CARTRIDGE MODULE DRIVE UNIT OPTIONS

Format	256 Bytes/Sector 64 Sectors/Track 808 Tracks/Surface
Capacity	12.928 Mbytes/Surface
Spare Tracks per Surface	15
Track Density (tracks/in)	384
Track Spacing	0.0026 in (0.0660) mm
Data Surfaces	Two for 32 MB version: Six for 96 MB version
Servo Surfaces	Two
Recording Mode	MFM
Bit Rate, Nominal	9.67 MHz
Diameter	14 in (355 mm)
Coating	Magnetic oxide
Performance (processing speed) Data Transfer Spindle Speed Bit Density inner track outer track	 1.2 MBs 3600 rpm 6038 bits/in. 4038 bits/in.
Accessing Time Maximum Full Stroke Average Maximum One Track	 55 ms 30 ms 6.0 ms
Latency Time Maximum Average	 17.3 ms 8.33 ms

CARTRIDGE MODULE DRIVE UNIT OPTIONS — Continued

Recording Heads	
Read/Write	Two for 32 MB version: Six for 96 MB version
Servo Type	Two
Read/Write Width	0.002 in (0.0508 mm)
Type	Self-loading, single gap, no erase
Data Error Rate	
Recoverable	Not more than one error in 10^{10} bits transferred
Reliability and Service	
MTBF	4000 hours
MTTR	1.5 hours
Service Life	Five Years
Operator Control Panel	Start/Stop switch with indicator Logic plug with Ready indicator Fault Clear switch with Fault Status Indicator Write-Protect switch (cartridge/fixed) with Indicator
Voltage	100, 120 Vac
Frequency	50/60 Hz
Phase	Single
Operating Current	8.2 Amps
Dimensions	
Height	10.65 inches (27.0 cm)
Width	19.00 inches (48.2 cm)
Depth	30.5 inches (77.4 cm)
Weight	170 pounds (77 kg)

LARK MODULE DRIVE UNIT OPTIONS

Drive Type	Eight-inch fixed and removable cartridge disks on common spindle; disk compartment sealed when operating, flying head air bearing and cooling supplied via internally recirculated and filtered air.
Head Positioner	Liner voice coil actuator and precision servo system
Cartridge	Eight-inch LARK Module Cartridge
Interface	SMD compatible
Capacity, Formatted	6.75 Mb or 20.4 Mb fixed disk 6.75 Mb or 20.4 Mb removable cartridge 13.5 Mb or 40.8 Mb Total
Format	
Bytes/Sector	256 plus CRC check characters
Sectors/Track	64
Tracks/Surface	206 or 624
Surfaces/Disk	two
Heads/Drive	four
Performance	
Rotational Speed	3150 RPM
Average Access Time	42 ms on 16 Mb drives or 35 ms on 50 Mb drives
Transfer Rate	9.67 Mbits per second
Error Detection and Recovery	CRC check and Early/Late Data Strobes
Max Start/Stop Delay	Two-minute disk purge cycle from Start switch to operation ready; one minute from Stop switch until cartridge can be removed.

LARK MODULE DRIVE UNIT OPTIONS — Continued

Reliability	7500 hour MTBF
Service Required Installation Preventative Repair	None, embedded servo information eliminates head alignment and cartridge interchange problems None; sealed drive Depot Repair
Service Life	Five years or 20,000 hours operating before need for factory overhauls.
Air Filter Change	Not required between factory overhauls
Environment, Operating Air Cleanliness Temperature, Operating Humidity, Operating Altitude, Operating	 Normal Office Conditions +10°C to +40°C (50°F to 104°F) 20% to 80% (non-condensing) 983 feet below sea level to 6560 feet above sea level
Power Requirement	138 watts running 175 watts starting
Dimensions Height Width Depth	 7.55 inches (19.2 cm) 18.96 inches (48.2 cm) 24.56 inches (62.4 cm)
Weight	55 pounds (25.9 kb)

EXORdisk III FLOPPY DISK OPTION

Performance	Rotational Speed — 360 RPM Track-to-Track Access Time — 10 ms (max) Head Load — 40 ms (max) Sector Read/Write Time — 2.4 ms (nom) Average Latency Time — 83.3 ns Transfer Rate — 426 Kbits/sec (nom)
Power Requirements	115 Vac/60 Hz
Physical Characteristics (two-drive system)	L x H x D — 17.75 x 6.96 x 23.5 inches (45.1 x 17.7 x 59.7 cm) Weight — 48 pounds
Electrical	All interface signals are TTL, Low-True 16 Output Lines, 8 Input Lines
Format Bits/Diskette (Soft Sector)	4M
Bytes/Diskette (Soft Sector)	512 K
Tracks/Diskette	154
Sectors/Track	26
Bytes/Sector	128

Specifications (Continued)

EXORterm 155 OPTION

Display	12″ diagonal, P4 phosphor with antiglare shield 24 lines, 80 characters per line 1920 character full display
Character Configuration	9 x 12 dot matrix 7 x 9 character size
Character Set	ASCII-96-character subset with Greek alphabet and six special graphics
Cursor	Blinking inverted video, nondestructive, on 9 x 12 block Incremental and absolute positioning (addressable, readable)
Keyboard	Encoded, n-key rollover 12 function keys LED mode indicators 86-key alphanumeric and control Full ASCII Typewriter layout Movable and detachable All keys typamatic
Interface	Asynchronous RS-232C compatible serial data control Input lines ±30 Vdc signal range Output lines ±10 mA current limited Half-duplex and full duplex Selectable baud rates — 110, 150, 300, 600, 1200, 1800, 2400, 4800, or 9600 Parity selection — odd, even, or none Word size — 7 or 8-bit Framing — 1 or 2-stop bits 103/202 modem compatibility (optional)
Power Requirements	100–130 Vac, 60 Hz @ 4 A 200–240 Vac, 50/60 Hz @ 2A 90–110 Vac, 50/60 Hz @ 4 A
Physical Dimensions Height Width Depth	 Display — 13.0″ H (33.02 cm) Keyboard — 3.45″ (8.76 cm) Display — 18.54″ W (47.09 cm) Keyboard — 18.54″ (47.09) Display — 20.0″ (51.54 cm) Keyboard — 8.54″ (21.69 cm)
Environmental	Operating temperature — +10° to +40°C Nonoperating temperature — −20° to +75°C Altitude — not to exceed 10,000 feet U.L. and C.S.A. approvable

EXORmacs Support Literature

M68KCHAS/D3	EXORmacs Chassis User's Guide
M68KDIOPP/D2	EXORmacs S/W Interface for Data I/O PROM Programmer User's Guide
M68KEMM/D3	EXORmacs Maintenance Manual
M68KMACS/D9	EXORmacs System Operations Manual
M68KMACSCM/D1	EXORmacs Development System Hardware/Software Configuration Manual
M68DMACSRK/D1	EXORmacs Chassis Rack Mounting
M68KMACSBG/D1	EXORmacs System MACSbug Monitor Reference Manual
M68DMBUG/D2	MACSbug Initialization and I/O Routines
M68KRDS/D2	EXORmacs System Remote Development Station
M68KUSE/D2	EXORmacs User's System Emulator (Single-User)
M68KVBS/D4	VERSAbus Specification Manual
M68RMSE/D2	M68RMWE1-1, MLD1-16, MLD1-50 Mass Storage Enclosures User's Manual
M68SXD155/D3	EXORterm 155 Display Console User's Manual
M68KRMS68K/D7	M68000 Family Real-Time Multitasking Software User's Manual
M68KVSF/D4	M68000 Family VERSAdos System Facilities Reference Manual
M68PRM/D	M68000 Programmer Reference Manual
M68KEDIT/D7	Resident Editor Reference Manual
M68DUNM/D1	System V/68 User's Manual

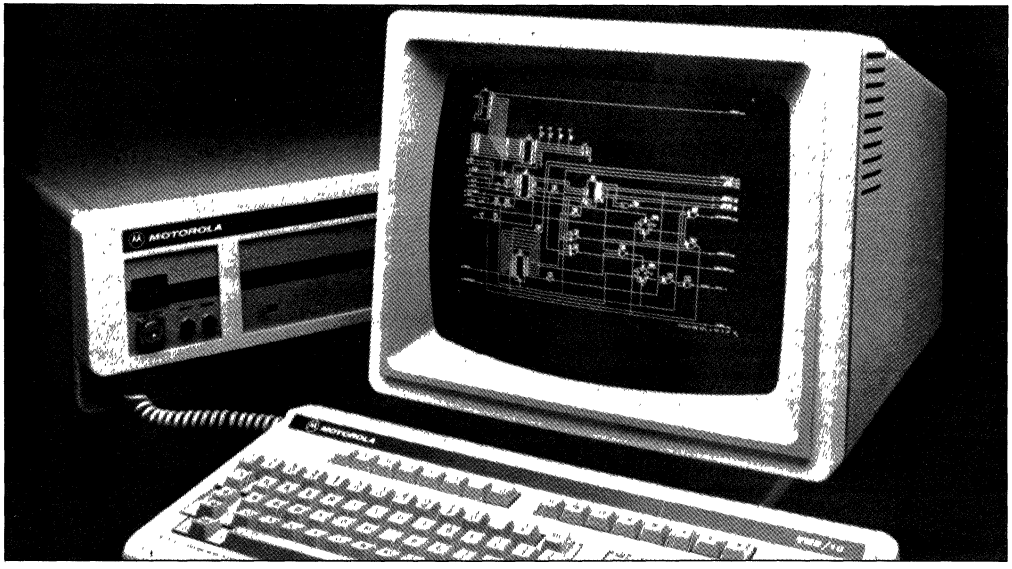
M68K102-B1
M68K102C1
M68K102D1

Technical Data

VME/10
OEM MICROCOMPUTER SYSTEM

FEATURES	OEM BENEFITS
• Designed Around Worldwide Standard Hardware and Software	• Lowers Overall Product Cost • Provides Compatibility • Increases Reliability • Lengthens Product Life
• Incorporates VMEbus System Architecture Standard Methodology	• Provides Choice from Over 400 Expansion Products • Allows Customization to Meet OEM Special Requirements • Reduces Risk and Investment Considerably • Allows "Quicker Time to Market"
• Field Proven Motorola Support, Reliability and Quality	• Provides Choice of Service Plans to Fit the Need • Reduces Learning Curve Because of Large Selection of Training Programs

THE VME/10 OEM MICROCOMPUTER SYSTEM



5

BASIC DESIGN FEATURES

- CONTROL UNIT CHASSIS
  - — MC68010 16/32-bit Microprocessor Unit
  - — MC68451 Memory Management Unit
  - — 8K Byte Static RAM for storage of user-definable character sets and display attributes.
  - — Two 28-pin sockets for ROM/PROM/EPROM storage of up to 64K bytes for custom applications.
  - — Battery backed-up time-of-day clock with 50 bytes of CMOS RAM storage.
  - — Software Controllable Display Formats
    1. 25 lines by 80 characters — 8 x 10 characters with descenders (10 x 12 character field)
    2. 800 x 300 pixel for low resolution graphics
    3. 800 x 600 pixel for medium resolution graphics
    4. Pixel graphics with overlaid character displays
  - — Industry-standard VMEbus interface with full bus arbitration logic and software controllable interrupter
  - — I/O Channel Interface for adding off-board resources such as A/D converters, serial and parallel I/O ports, etc.
  - — 384K Byte Dynamic RAM (multiported between graphics controller and local bus, and VMEbus)
  - — Mass Storage Subsystem providing both 5¼" Floppy Disk and 5¼" Winchester Disk Storage Units.

    Floppy Disk

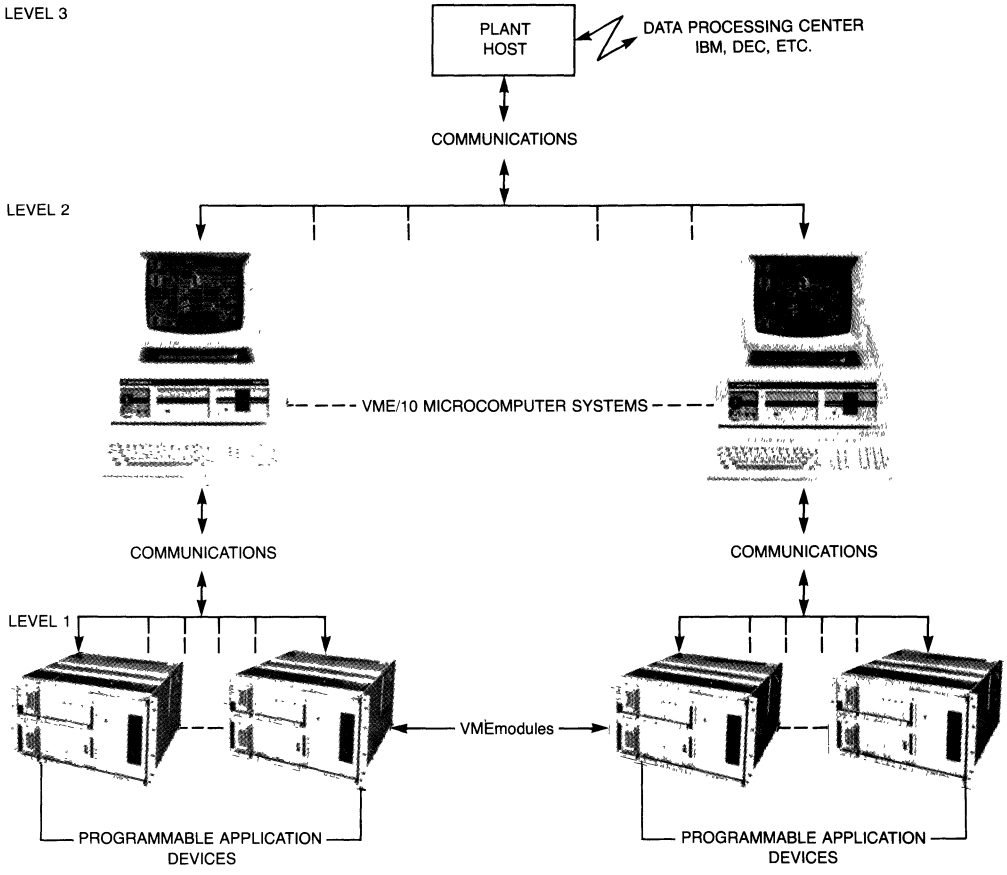
    1 Mbyte Unformatted Capacity (655K Byte Formatted)

    Winchester Disk

    Choice of (a) 19 1 Mbyte Unformatted Capacity (15 Mbyte Formatted)
    (b) High performance disk with 51.9 Mbyte Unformatted Capacity (40 Mbyte Formatted)
  - — Card cage for feature expansion capability.

    Five VMEbus Card Cage Slots with VMEbus backplane, plus four I/O Channel Slots
- OPERATING SYSTEMS
  - — SYSTEM V/68 Operating System the standard UNIX-derived Operating System for the M68000 family of microprocessors
  - — VERSAdos Real-Time, Multitasking Operating System with M68000 Family Macro Assembler, plus tools and utilities
  - — CP/M-68K including C, Pascal and CBASIC
- OPTIONAL VIDEO DISPLAY AND KEYBOARD
  - — 15" Monochrome Video Display Unit with 7-level gray scaling.
  - — 14" Color Video Display Unit with 8-Color Capability
  - — Detachable full ASCII keyboard with cursor control keys, numeric pad and 16 function keys
  - — Conformance to ergonomic standards applicable to video display and keyboard

TYPICAL DISTRIBUTED SYSTEM APPLICATION USING VME/10 AND VMEmodules

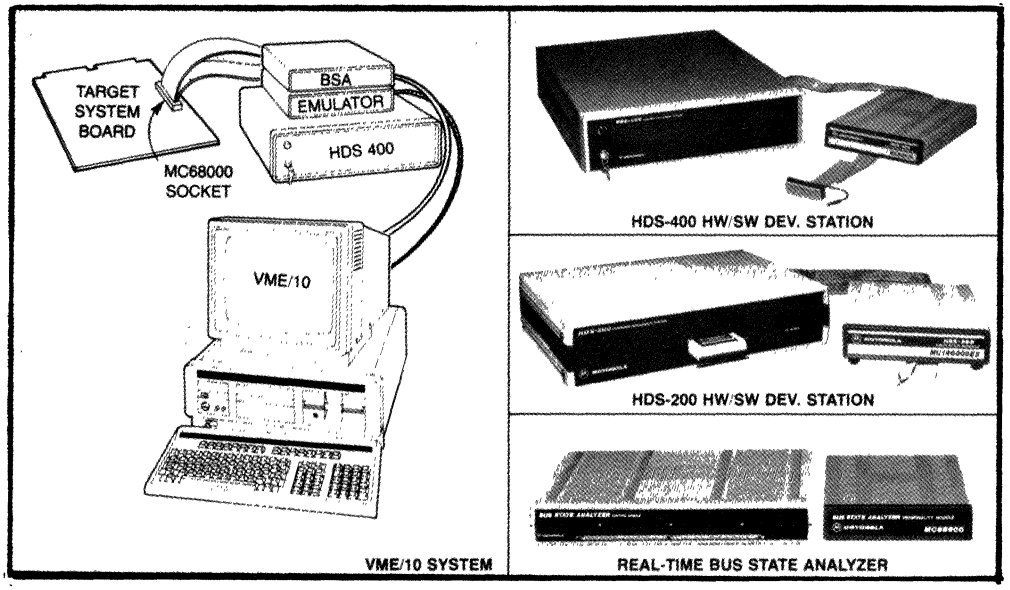


5

FLEXIBILITY FOR A RANGE OF APPLICATIONS

Flexibility is a key feature of the VME/10 System. Unlike many microcomputer systems which are optimized for specific end uses, the VME/10 System is designed to meet the needs of a broad spectrum of OEM's and system developers who are targeting the industrial and technical markets. Another significant aspect of the VME/10 System is the combination of both VMEbus and I/O Channel interfaces along with supporting card cages for expansion modules. The system level VMEbus allows system expansion via high-performance VMEmodules; and, in a complementary fashion, the I/O Channel permits addition of a variety of modular I/O interface boards without incurring any additional load on the system bus.

TYPICAL HARDWARE DESIGN STATION HOOK-UP



5

SYSTEMS DEVELOPMENT AND INTEGRATION

The initial stages of developing a microprocessor-based system normally involve two parallel, rather independent, efforts. One is the hardware design — the other the software design.

The VME/10 System, when used as the host along with Motorola's new line of Hardware Development Stations (HDS), simplifies the design process because of its ability to bring the hardware and software development processes into intimate relationship with each other throughout the development cycle.

The HDS, in conjunction with the VME/10 software development system's ability to download generated object code into the HDS, provides a complete hardware and software development system for the Motorola families of microprocessors. Two major factors contribute to the HDS's usefulness as a systems development tool. The first is the ability to serve as a fully functional substitute for the selected microprocessor or microcomputer in the user's target system. By plugging the HDS emulator into the socket on the prototype hardware, it allows efficient testing of hardware as well as software. The second factor is its ability to interface with the Real-Time Bus State Analyzer which speeds the debugging process and allows program code optimization

The Motorola Real-Time Bus State Analyzer (BSA) is a development tool which allows simultaneous monitoring of different points in a system Interfacing through the system bus, or the "native" MPU bus, the BSA tracks events occurring on each line of the bus, storing the information for later analysis and interpretation

The emulator can be configured and software run prior to availability of prototype target system hardware. This allows an early start on the debugging process As hardware changes occur, software updating and debugging is readily accomplished Moreover, with the HDS it becomes economically feasible to test alternate design approaches in order to determine the best solution.

VME/10 SYSTEM CONFIGURATION

BASIC VME/10 SYSTEM HARDWARE

The basic VME/10 System consists of the Control Unit Chassis plus Motorola supplied keyboard and display unit. The Control Unit Chassis can be purchased standalone and the OEM can add monitor or terminal capability to meet his needs.

The following paragraphs provide detailed descriptions of the features and functions incorporated into these three sub-elements.

CONTROL UNIT CHASSIS

This chassis, as shown in Figure 1, provides the housing for a 400 watt switching power supply sufficient to handle all power needs for system configuration, a cooling fan, a front control panel for user manual control as well as.

A. System Control Module
B Mass Storage Subsystem
C. Expansion Card Cage

A The **System Control Module (SCM)** contains the central intelligence of the VME/10 System It consists of a two-board set: the Processor/MMU Board and the Graphics/Interface Board.

(1) **Processor/MMU Board**

The combination of the MC68010 MPU and MC68451 Memory Management Unit provides processing power sufficient to permit several development tasks to proceed simultaneously — editing, program development, system debugging — with full protection for each task This fundamental processor/memory architecture also provides designers with the protection features required in multitasking OEM systems where security and protection of both programs and data are essential. The Processor/MMU Board contains one MC68451 MMU device, affording up to 32 separate program/data segments.

(2) **Graphics/Interface Board**

This second board of the two-board set contains the major elements of high-speed semiconductor memory in the system, plus the graphics subsystem and interfaces to various off-board devices. This board incorporates the following specific features

a. **384K bytes RAM** — utilizing 64Kx1 HMOS RAM technology for operating system, support software, user programs and Graphics Subsystem display buffer storage (See section (i) for detail on the Graphics Subsystem) For increased performance and minimum contention, the on-board RAM is multiported to allow shared access from the local intercommunication bus, the VMEbus, and the graphics controller.

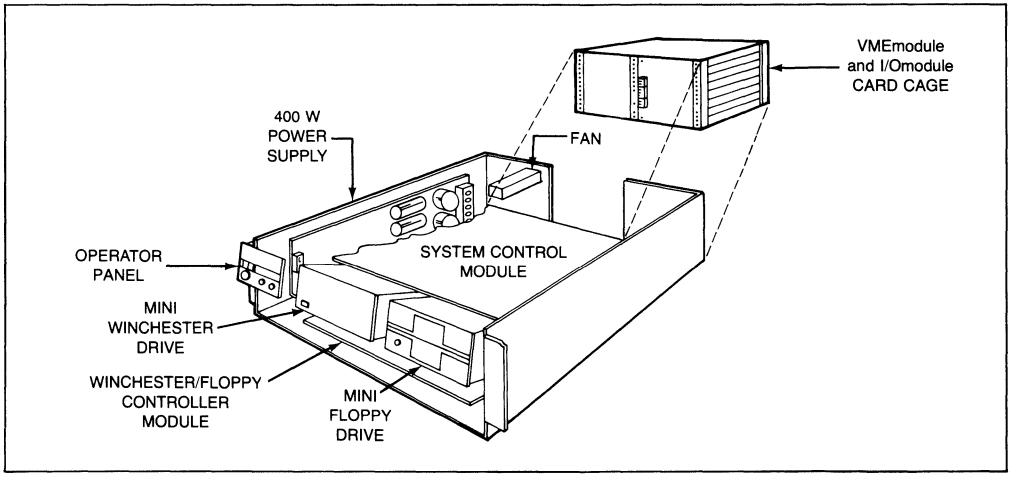
b **32K bytes ROM/PROM/EPROM** — used for bootload and test and diagnostic routines for debugging and system start-up and control.

c. **An interrupt handler** to allow 22 sources of interrupts to the MC68010 Microprocessor.

d. **Time-of-day Clock** (MC146818), an 8-bit real-time clock including 50 bytes of general purpose RAM for saving critical information. Both the clock and RAM are battery backed-up allowing up to five days data retention with fully charged batteries at power down.

e. **Keyboard Interface** (MC68661) using RS-422 type buffers to the multidrop 5 Volt differential serial line to the keyboard.

FIGURE 1 — CONTROL UNIT CHASSIS



f. **Local On-Board Bus** providing communication between the MPU, ROM, RAM, CRT Controller, Keyboard Interface, battery backed-up Time-of-day Clock, I/O Channel, and the VMEbus. This architecture allows the on-board processor to continue operating at full speed on-board while other (optional) VMEbus masters operate simultaneously.

g. **I/O Channel Interface** linking the local intercommunication bus to the I/O Channel cable for communication to the Mass Storage Subsystem and any optional I/O cards installed

h **Industry-standard VMEbus Compatibility** is supported by three functions.

1) The VMEbus interface which provides the data and address path from the on-board MPU via the local bus to the VMEbus to allow VMEbus use in a system requiring additional off-board resources such as additional memory, processors, or intelligent device controllers.

2) The VMEbus Arbiter which arbitrates all four bus request priority levels, with operation transparent to software.

3) The VMEbus Requester which is used to gain access to the resources on the VMEbus Bus requests can be made either indirectly (software transparent) or directly by specific request through corresponding Status and Control Registers under program control. The associated VMEbus Interrupter logic permits the MPU to place an interrupt on one of the seven VMEbus interrupt request lines The related Interrupt Handler can be software configured to respond to any subset of the seven VMEbus interrupt request lines, thus allowing several boards with like interrupt handlers to respond to different interrupt levels from the VMEbus

i **Graphics Subsystem**

Increased emphasis on video display graphics at the user interface in microcomputer-based systems has highlighted the requirement for systems with graphics display capabilities which are both low cost and efficient The VME/10 System offers a solution which provides quality, efficient graphics support hardware at low cost to the user.

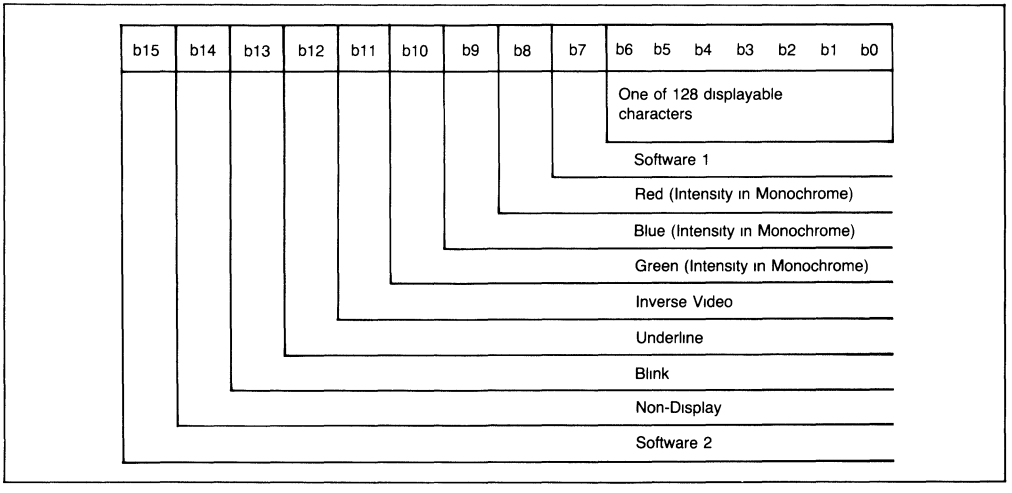
The Video Graphics Subsystem generates all video and display synchronization signals required by both monochrome and color display units The standard VME/10 System incorporates a monochrome video display, with a color video display available as an option

Fundamentally, the VME/10 System displays characters in a 25-line by 80-column format, or medium-resolution pixel graphics in an 800 x 600 pixel matrix, or a specialized combination display of both character and pixel graphics simultaneously Within this basic framework, additional capabilities are provided for (a) alternate low-resolution pixel graphics within an 800 x 300 point field, and (b) user definition through software of specialized character sets, fonts, and display attributes An MC6845 CRT Controller device is used by software to define the video screen

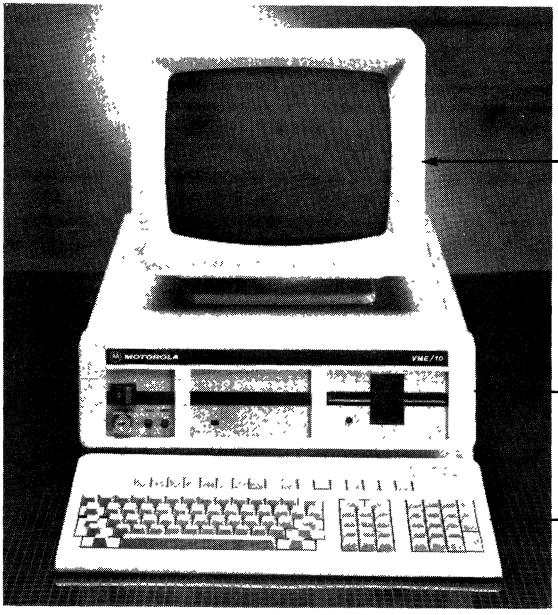
The 384K byte RAM in the System Control Module has the dual purpose of general software storage and of graphics data storage If no pixel graphics capability is being used, then all of the RAM on the SCM is available as system RAM If graphics is being used in the normal (low resolution) mode, then the graphics display RAM is located in the high-address 96K bytes of RAM. Alternately, in the medium-resolution mode, the high-address 192K bytes of RAM are dedicated to graphics.

The VME/10 graphics subsystem combines separate character and graphic display memories. The character memory is 16 bits wide and is 3K bytes deep. The 16 bits of each word are defined in Figure 2.

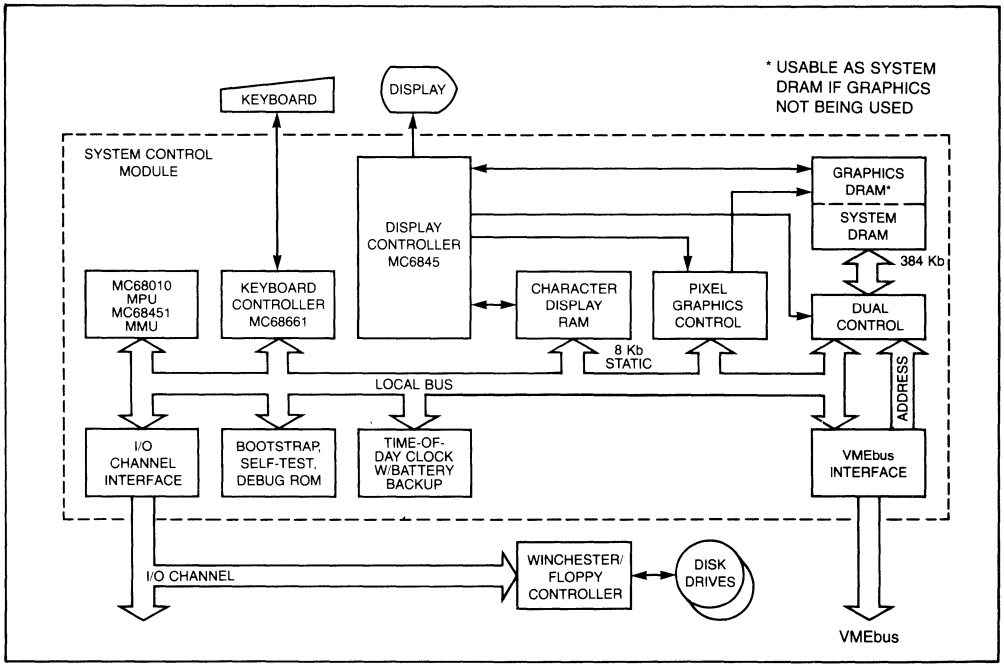
FIGURE 2 — CHARACTER MEMORY WORD FORMAT DEFINITION



VME/10 MICROCOMPUTER SYSTEM BASIC SUB-ELEMENTS



VME/10 SYSTEM BLOCK DIAGRAM



The character generator RAM is initialized with the standard ASCII character definitions, but can be modified by the user to define alternate special symbols required by the application The font is 8 x 16 in a 10 x 24 character field for the medium resolution monochrome display and 8 x 10 in a 10 x 12 character field for the normal resolution/color display.

Two alternate modes of pixel graphics operation are available on the VME/10 System: the Group Access Mode and the Pixel Access Mode. The application determines the most appropriate mode to use The Group Access Mode, so-called because a small group of contiguous pixels may be accessed simultaneously, utilizes three separate, contiguous banks of memory

Each bank represents a primary color for color applications, or an intensity level for monochrome applications The graphics display buffer RAM block in the memory map is organized so that the first third of the graphics RAM locations is Bank 1, the second third of the graphics RAM is Bank 2, and the third is Bank 3. This organization allows the MPU to change 8 or 16 contiguous pixels on the screen at one time in one color/intensity bank. This mode is particularly useful for drawing bar graphs, color filling and object, or blanking the screen

The graphics RAM is accessible in another mode called the Pixel Access mode. In the Pixel Access Mode, read/write hardware exists to allow the processor the ability to change one pixel at a time (per memory cycle) in all three banks In this mode the processor uses only word accesses, and writes a special "pixel access word" to addresses defined in non-existent memory.

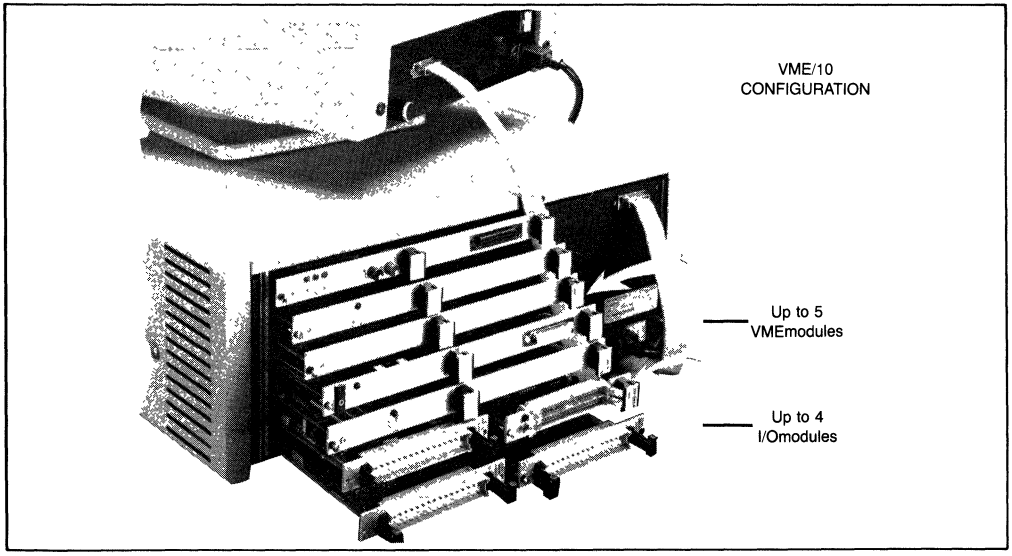
The organization of the Pixel Access Word is as shown in Figure 3

The Pixel Access Mode is oriented toward drawing lines or changing a portion of a display. The mask bits allow the user to avoid disturbing the contents of a given plane (or planes) while changing the contents of another plane (or planes) This technique minimizes the software design effort required and improves system performance It further eliminates the necessity to rewrite data into bank addresses when the data remains unchanged.

FIGURE 3 — PIXEL ACCESS WORD FORMAT DEFINITION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A	N/A	N/A	N/A	N/A	MASK3	MASK2	MASK1	N/A	N/A	N/A	N/A	N/A	BANK3	BANK2	BANK1

VMEbus AND I/O CHANNEL MODULE EXPANSION CARD CAGE OPTIONS



B **The Mass Storage Subsystem** consists of a disk controller board, plus one 5¼" Floppy Disk Drive Unit and one 5¼" Winchester Disk Drive Unit The Floppy Disk Drive affords storage of 1 0 Mbyte unformatted (655K byte formatted) capacity, incorporating reliable dual-density, dual-sided 96 TPI technology. The associated Winchester disk drive unit is available in two storage capacities, depending on the specific VME/10 System model

Option #1 — 19 1 Mbyte Unformatted Capacity (15 Mbyte Formatted)

Option #2 — 51 9 Mbyte Unformatted Capacity (40 Mbyte Formatted)

C An **Expansion Card Cage** is provided with card plug-in access from the rear panel of the Control Unit Chassis This combination VMEbus and I/O Channel Card Cage, with 5-slot VMEbus backplane, accommodates up to five double Eurocard format VMEmodule boards, plus up to four single Eurocard format I/Omodules All necessary I/O channel cabling and connectors are installed to serve the four I/O channel card slots This card cage is standard with the 19 1 and 51 9 Mbyte Winchester disk model

Note A 5 Mbyte formatted Winchester disk with 5-slot single Eurocard card cage is available for special orders Consult factory for more information

These card cages allow convenient system expansion and customization through addition of I/O Channel and VMEbus compatible cards. Common applications are additional global system memory, serial and parallel ports, analog conversion functions, and IEEE-488 interface to intelligent instrumentation

DISPLAY UNIT

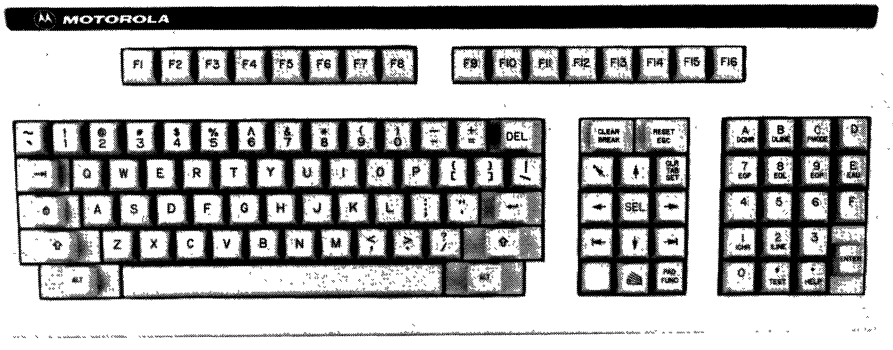
The monochrome video display is a 15" (diagonal) monochrome unit with antiglare, P39 (green) phosphor screen A 14" (diagonal) RGW color monitor with long persistence phosphor is available as an option. The Display Unit is mounted on a tilt and swivel stand that allows freedom of movement in vertical and horizontal directions This configuration conforms to all ergonomic standards in order to increase user comfort and productivity.

The standard ASCII character set with 8 x 10 characters with lower case descenders in a 10 x 12 font is used over a 25 line x 80 character display in normal mode 800 x 600 pixel medium-resolution graphics, 800 x 300 pixel low-resolution graphics, or a combination of characters and graphics may be displayed The character set and attributes displayed can be easily changed by altering the contents of the character/attribute static RAM. Underlining, blinking, user-defined masks and window capability are programmable options Characters or individual pixels can be displayed in any of seven levels of the gray scale on the monochrome display

KEYBOARD

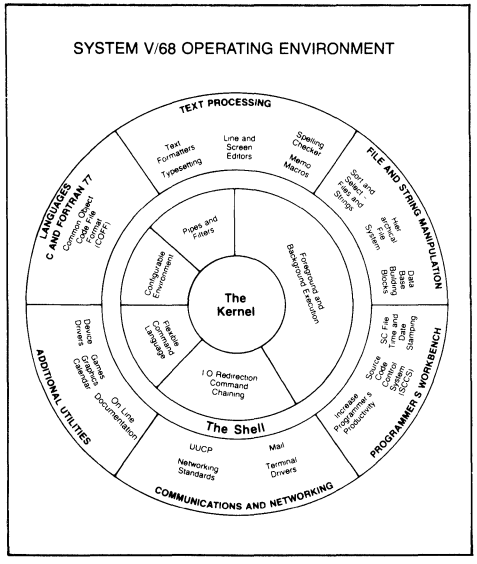
The keyboard is connected to the Control Unit Chassis via a serial link using a coiled telephone cable with quick disconnect "modular" connectors The keypad consists of a full ASCII character set with interchangeable keycaps, 16 function keys for selection of pre-defined program sequences, a 7-key cursor/tab control pad and a 10-key numeric pad. The keyboard is also designed to conform to all ergonomic design principles

VME/10 SYSTEM KEYBOARD



VME/10 SYSTEM SOFTWARE

SYSTEM V/68 OPERATING SYSTEM SOFTWARE



SYSTEM V/68 OPERATING SYSTEM

The System V/68 Operating System is the standard UNIX-derived Operating System for the M68000 family of microprocessors It offers a small compact kernel, which provides process scheduling and I/O facilities to all programs. In addition, a powerful command shell for interactive system controls and an extensive set of utility programs for many tasks, such as program development, text processing, electronic mail, and networking support are included

TEXT PROCESSING

Text processing utilities include the *ex/vi* full-screen editor, NROFF and TROFF text formatters, a spelling checker, and programs for formatting tables and mathematical equations. The *ex/vi* editor supports a large number of existing terminals, including the Motorola EXORterm 155, through the use of the *termcap* terminal data base *Termcap* entries for new terminals may be added by the user

LANGUAGES

As an integral part of System V/68, C Language is offered C Language has developed into one of the most popular commercial programming languages, and is used frequently in developing portable application software. System V/68 offers significant enhancements to C Language, along with several new language utilities. CXREF, a new cross reference program, and CFLOW, a new flow analysis program, are just two of the new utilities offered. System V/68 also includes a FORTRAN 77 compiler as well as an M68000 assembler and linker/loader.

COMMUNICATIONS

System V/68 utilities provide support for electronic mail, communications, and networking. Electronic mail allows users to communicate with one another, using the system as a mailbox or as a bulletin board The communications utilities allow a System V/68 user to communicate to mainframe computers. Networking support allows several computers to be linked together, either through dedicated links or by dial-up telephone connections. With these utilities, Motorola development system users can communicate with one another. In addition, target systems may be developed with similar capabilities. In System V/68, interprocess communications routines have been added These include shared memory, messages, and semaphores Also included is an IPC/remove command which removes message queues, semaphores, and shared memory identifiers from the system.

PROGRAMMER'S WORKBENCH

The Programmer's Workbench utilities support the development of large software systems in a professional manner They include the Source Code Control System (SCCS), which provides facilities to store, update and retrieve all versions of source code modules, YACC, which generates parsers; LEX, which builds lexical analyzers, and other utilities which enhance programmer productivity and the quality of work

MEMORY REQUIRED

In addition to the standard 384K, SYSTEM V requires the addition of one 256K Byte VMEbus RAM module Order part number MVME201.

VERSAdos OPERATING SYSTEM AND DEVELOPMENT TOOLS

To achieve most efficient use of the varied system resources provided by the VME/10 Microcomputer System, efficient system control and management software is required. This need is met superbly by the VERSAdos Operating System and its related family of development tools and utilities for the M68000 and M6800 MPU families. VERSAdos incorporates a modular, multilayer design supporting a variety of application environments, and is especially well-suited to real-time control system and applications. Since it provides a convenient, friendly interface between the user and the system resources, including a wealth of development support software, VERSAdos has emerged as the operating system environment of choice for increasing numbers of system developers who are incorporating the Motorola 8 and 16-bit microprocessor families into various end applications. VERSAdos is particularly well-suited to providing a common host/target environment, thereby minimizing the task of system integration.

As packaged with the VME/10 System, the complete basic VERSAdos operation environment includes, in addition to the basic operating system, an M68000 Family Structured Macro Assembler, Symbolic Debugger, CRT-Oriented Text Editor, and M68000 Family Linkage Editor. In addition, a Diagnostics package, TENbug, is provided for use in helping to isolate suspected hardware system problems.

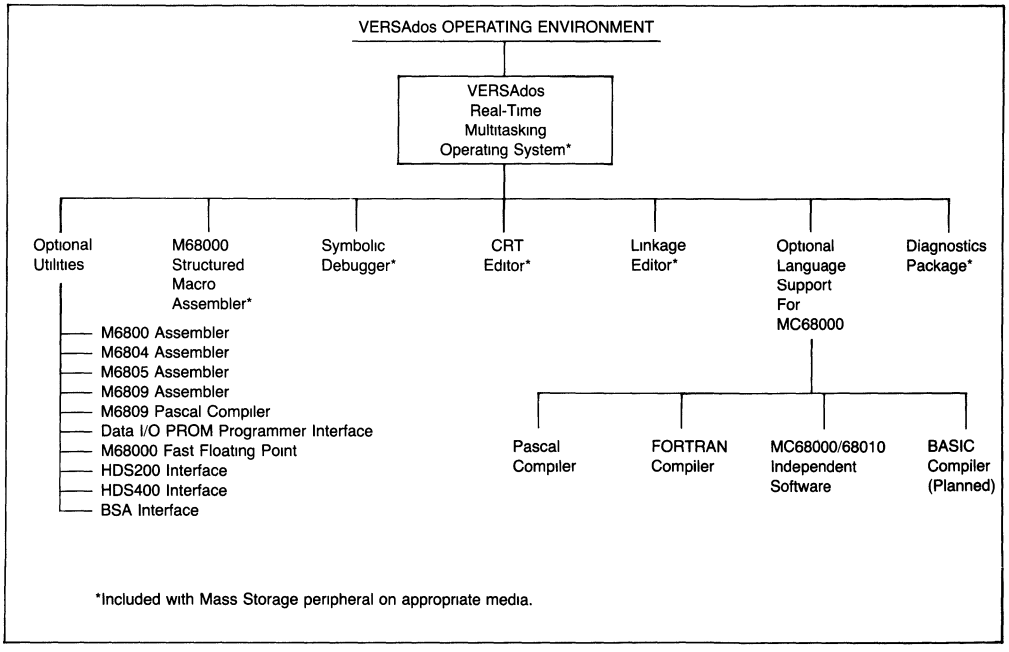
VERSAdos OPERATING SYSTEM CAPABILITIES

VERSAdos is a multitasking, multiprogramming system executing on the MC68010 MPU in the VME/10 Microcomputer System. Programs execute in dynamically-assigned, variable-length segments with read/write privileges. Instructions and data are located in separate memory segments.

The heart of VERSAdos is the RMS68K Real-time Executive which provides task services and supports memory management. It also allows inter-task communication, provides exception monitor facilities, and handles system interrupts.

The Input/Output (I/O) Subsystem of VERSAdos supports device independence, logic I/O, overlapped computation, and physical I/O. Both sequential and random record access are supported by VERSAdos.

The powerful VERSAdos File Management System supports three file structures — contiguous, sequential length, and indexed sequential. Other file system features include disk and file protection, shared file access, dynamic file allocation, and fixed or active protection



VERSAdos OPERATING SYSTEM AND DEVELOPMENT TOOLS (Continued)

RESIDENT STRUCTURED ASSEMBLER

The M68000 Resident Structured Macro Assembler translates source statements into relocatable machine code, assigns storage locations to instructions and data, performs auxiliary assembler actions designated by the programmer, and optionally produces a cross reference listing The M68000 Resident Assembler includes macro and conditional assembly capabilities, plus certain "structured programming" control structures such as "for, repeat, while" (loops) and "if-then, if-then-else" (conditional branches)

SYMBOLIC DEBUGGER

The Symbolic Debugger (SYMbug) program is used to debug other programs whose source code is written in Motorola's structured Macro Assembler for execution on the M68000 Family MPU's. The language processors, in cooperation with the linkage editor, supply information to SYMbug. This permits the user to describe the debugging requirements to SYMbug in terms close to the language in which the source program was written. SYMbug allows the user to debug in symbolic terminology SYMbug allows the user to perform the following:

- Examine, insert, and modify program elements such as instructions, numeric values, and coded information (i.e., data in all its representations and formats).
- Control execution, including the insertion of breakpoints into a program and request for breaks or changes in elements of data
- Trace execution by displaying information at designated points in a program.
- Search programs and data for specific elements and subelements.
- Create macro commands allowing user-defined formats and commands

CRT TEXT EDITOR

The CRT-oriented Text Editor provides the capability to create and modify source programs The editor supports both command and page editing, utilizing the cursor control keys, control characters, and function keys of the VME/10 keyboard to insert, alter, or delete characters and lines within a user text file.

LINKAGE EDITOR

The Linkage Editor provides the capability of converting one or more separately-compiled object units into a loadable object module file. The Editor determines segment attributes, calculates address space, searches libraries, resolves external references, relocates object code, and issues error messages. At the end of a linkage process, the editor prints a report that contains a module map, a table of externally-defined symbols, and any unresolved or multiply-defined symbols.

TENbug DIAGNOSTIC PACKAGE

The purpose of the TENbug Diagnostics Package is to verify the overall functionality of the VME/10 system by exposing it to a set of off-line tests. The package provides a diagnostic monitor and two levels of diagnostics The first level is the firmware resident power-up/reset test and the second level is comprised of disk resident diagnostics for more extensive hardware testing and symbolic assembly/disassembly debugging. The governing guideline for TENbug is to provide a user-friendly, comprehensive test and debug package that will isolate a malfunction to a functional block and at least down to the faulty module.

OPTIONAL LANGUAGE SUPPORT

OPTIMIZING PASCAL COMPILER

Pascal is a high-level, user-oriented language for the MC68000/68010 MPU, based on the language as defined by Niklaus Writh. Pascal is a highly structured language which promotes good programming techniques, is self-documenting, and its user-oriented statement forms simplify program writing Extensions provided by Motorola include: Address specification for variables, alphanumeric labels, string types, exit, non-decimal integers, runtime error checking, runtime file assignment, and separate compilation and linking. The optimizer produces compact efficient code. Library routines include both IEEE floating point and a single precision fast (multiply: 44 microseconds) floating point

FORTRAN COMPILER

The FORTRAN Compiler translates source programs written in FORTRAN into MC68000 machine language, object code. FORTRAN is a high-level programming language widely used for scientific and engineering problem solving with features also useful for certain business-related applications.

MC68000/68010 INDEPENDENT SOFTWARE SOURCES

A broad range of independent software suppliers support Motorola's 16/32 bit microprocessors. See the latest issue of the "Motorola Microprocessor Software Catalog" for a list of applicable software and addresses of independent suppliers.

VME/10 SYSTEM SOFTWARE (Continued)

CP/M-68K OPERATING SYSTEM

CP/M-68K extends the popular CP/M operating system to the Motorola MC68000 family of microprocessors. It is a high performance single user, single-tasking operating system CP/M-68K is ideal for the business user — it is easy to learn and use. CP/M-68K also features a flexible application program interface and powerful system utilities combined with a C compiler to provide a complete software development environment

CP/M-68K includes a fast, reliable file system It has a time tested, modular design which allows it to be easily customized to run in a particular hardware environment. System dependent input/output device handlers are located in a module called the Basic Input/Output System (BIOS). The BIOS interfaces to the logical, hardware independent portion of the operating system, and is the only module which differs from machine to machine.

The RAM resident portion of CP/M-68K is small, requiring about 40K bytes of memory Its size is dependent on the size of the BIOS The actual size of the customized BIOS is dependent on the number of peripheral devices in the system. Although, it is compact, CP/M-68K can manage up to 16 Mb of RAM, and up to 16 disk drives, making it a good match for today's MC68000-based microcomputer systems

CP/M-68K Operating System features the following.

- CP/M and CP/M-86 Compatible
- Supports from 64K bytes to 16 Mbytes of RAM
- Supports from 1 to 16 disk drives of up to 512 Mbytes for a maximum of 8 gigabytes of on-line storage
- Languages including C, Pascal MT+, and CBASIC Compiler
- Standard CP/M utilities including PIP, ED, DDT, STAT, SUBMIT, and DUMP
- Multiple Programs can co-exist in RAM
- Resident System Extensions
- Allows full access to the VME/10
- Cross Development Tools including:
  - A C compiler and C run-time library compatible with UNIX software
  - An assembler which supports standard Motorola MC68000 assembly language
  - A linker which produces both relocatable and absolute load modules
  - An object module librarian
  - A utility to convert CP/M-68K load modules to Motorola S-Record form
  - A relocation utility to convert relocatable load modules to absolute form
  - A utility to list the symbol table contained in a load module file
  - A utility which prints the memory size required by the load module

These tools can run on any CP/M-68K system with at least 128K of memory.

5

MECHANICAL AND ENVIRONMENTAL SPECIFICATIONS

Characteristic	Specification		
Overall Dimensions	Control Unit	Display	Keyboard
Length	22.8" (57.9 cm)	15.0" (38.1 cm)	8.3" (21.1 cm)
Height	7.0" (17.8 cm)	13.0" (33.1 cm)	2.0" (5.1 cm)
Width	19.0" (48 3 cm)	14.0" (35 7 cm)	19.0" (48.3 cm)
Weight	35 lbs (16 1 kg)	15 lbs (6.9 kg)	5 lbs (2.3 kg)
Power Requirements	90-132/180-264 Vac 50-60 Hz	Power: Control limit — 500 Watts; Color Display — 60 Watts	
Temperature			
Operating	10°–40°C		
Storage	10°–47°C		
Humidity	20%–80% Non-Condensing		
DC Power available for VMEmodules and I/O modules	23 A @ 5 Vdc, 4 A @ – 12 Vdc, 6 A @ + 12 Vdc		

ORDERING INFORMATION

STANDARD SYSTEM CONFIGURATIONS

Features	Part Number					
	M68K102B1	*M68K102C1	M68K102D1	M68K102B2	*M68K102C2	M68K102D2
Control Unit Chassis	15 Mbyte	40 Mbyte	40 Mbyte	15 Mbyte	40 Mbyte	40 Mbyte
I/O Channel Card Slots	4	4	4	4	4	4
VMEbus Backplane Slots	5	5	5	5	5	5
14" Color Display Unit	—	—	X	—	—	X
15" Monochrome Display Unit	X	X	—	X	X	—
VME/10 System Keyboard	X	X	X	X	X	X
VERSAdos Operating-System and Development Tools	X	X	X	X	X	X
TENbug Debug/Monitor	X	X	X	X	X	X
Winchester Seek Time (ms avg)	70	33	33	70	33	33
Electrical Characteristics	115 Vac 60 Hz	115 Vac 60 Hz	115 Vac 60 Hz	230 Vac 50 Hz	230 Vac 50 Hz	230 Vac 50 Hz

Note Motorola software is supplied under license agreement
*When this hardware configuration is intended to be used with SYSTEM V/68 Operating System Software an MVME201 VMEbus RAM module must be purchased

SOFTWARE OPTIONS FOR VERSAdos OPERATING SYSTEM

M68VVXBPASCAL SPD Only	M68000 Pascal Compiler on 5¼" Diskette
M68K1XBDIOPP SPD Only	Data I/O Software Interface on 5¼" Diskette
M68VKXSVMEBUG SPD Only	VMEbug Source and Relocatable Object Modules on 5¼" Diskette
M68V9XBPASCAL SPD Only	M6809 Cross Pascal Compiler on 5¼" Diskette
M68VCXBASM SPD Only	M6800/04/05/09 Cross Macro Assembler on 5¼" Diskette
M68VVXBABSFTN	M68000 FORTRAN Compiler on 5¼" Diskette
M68VKXBPTOM	PDP-11 to 68000 Translator on 5¼" Diskette

Note Requires Software License Agreement applicable to the individual software product

OPERATING SYSTEM OPTIONS

Part Number	Description
M68NNXBSV10	SYSTEM V/68 software supplied on 5¼" diskette SYSTEM V/68 includes the following object code modules. — M68000 SYSTEM V/68 Operating System — Motorola M68000 C Language Compiler, Assembler, and Linker — Full set of SYSTEM V/68 documentation — Graphics Support Object code will be supplied as bootable load modules, and as relocatable, partitioned, and unlinked modules so that the OEM can reconfigure the SYSTEM V/68 operating system without purchasing source code. In addition, a sample source code driver is included.
M68K11XBCPM68K	CP/M-68K software supplied on 5¼" diskette, CP/M-68K includes the following. — CP/M-68K Operating System — CP/M Compatible — C Language — Cross Development Tools — Standard CP/M Utilities

Note It is intended that SYSTEM V/68 software be used with the 40 MB option Also required is a 256K Byte VMEbus RAM Module Order part number MVME201

MODULAR EXPANSION OPTIONS — VMEbus COMPATIBLE

Part Number	Description
MVME025	System Controller
MVME050	System Controller
MVME101	Monoboard Microcomputer
MVME110-1	Monoboard Microcomputer
MVME115	Monoboard Microcomputer
MVME120	Monoboard Microcomputer with 4K Bytes CACHE
MVME200	64K Dynamic RAM with Byte Parity
MVME201	256K Dynamic RAM with Byte Parity
MVME202	512K to 2048K Dynamic RAM with byte parity
MVME211	Static ROM/RAM Module
MVME300	High Performance IEEE-488 GPIB Controller with DMA
MVME310	Intelligent Controller for Custom Interfacing
MVME315	Intelligent Floppy Controller/SASI Interface
MVME320	Winchester/Floppy Disk Controller
MVME330	Ethernet LAN Controller

MODULAR I/O EXPANSION OPTIONS — I/O CHANNEL COMPATIBLE

Part Number	Description
MVME400	Dual RS-232C Serial Port
MVME410	Dual 16-bit Parallel Port NOTE. This module recommended for Printer interface port applications with the VME/10 System
MVME420	SASI Adapter
MVME435A	Buffered 9-Track Magnetic Tape Adapter
MVME600	12-Bit Analog Input Module
MVME601	16 Channel Expander for MVME600
MVME605	12-Bit Analog Output Module
MVME610	Opto-Isolated 120V/240 Vac Input
MVME615/616	Opto-Isolated 120V/240 Vac Output
MVME620	Opto-Isolated 60 Vdc Input
MVME625	Opto-Isolated 60 Vdc Output
MVME701/702	I/O Translation for use with MVME050
MVME935	Remote I/O Channel Extender Cable Connection Module

REMOTE I/O CHANNEL MODULES

M68RAD1	Remote Intelligent Analog Conversion Module
M68RIO1	Remote Input/Output Solid State Relay Module

INSTRUMENTATION OPTIONS — 16-BIT EMULATORS

Part Number	Description
M68KHDS400A	HDS-400 Control Station for use with RS-232C serial interface to VME/10 host system. For 90–120 V, 50/60 Hz operation.
M68KHDS402A	Same as M68KHDS400A, except for 220 V, 50 Hz operation.
M68KHDS16FB	16-bit M68000 Family Interface Module
M68000HDS4	MC68000 Emulator Module (12.5 MHz)
M68008HDS4-8	MC68008 Emulator Module (8.0 MHz)
M68010HDS4-8	MC68010 Emulator Module (8.0 MHz)
M68KHDS4-2	HDS-400 System Software for use with VME/10 host and VERSAdos Operating System Software on 52" floppy diskette.

Note: For additional HDS-400 product details and ordering information, see the HDS-400 data sheet

INSTRUMENTATION OPTIONS — 8-BIT EMULATORS

Part Number	Description
M68HDS201	HDS-200 Control Station, for 90–120 V, 50/60 Hz operation.
M68HDS202	Same as M68HDS201, but for 200–260 V, 50 Hz operation
M6804P2HM	MC6804P2 Emulator Module
M6805P234HM	MC6805P2, P4, P6, MC68705P3, P5 Emulator Module
M6805RU23HM	MC6805R2, R3, U2, U3, MC68705R3, R5, U3, U5 Emulator Module
M6805S2HM	MC6805S2 Emulator Module
M6805T2HM	MC6805T2 Emulator Module
M68HC05C4HM	MC68HC05C4 Emulator Module
M146805E2HM	MC146805E2 Emulator Module
M146805F2HM	MC146805F2, MC1468705F2 Emulator Module
M146805G2HM	MC146805G2, MC1468705G2 Emulator Module

Note: For additional HDS-200 product details and ordering information, see the HDS-200 data sheet

INSTRUMENTATION OPTIONS — BUS STATE ANALYZERS

Part Number	Description
M68BSAC	Bus State Analyzer Control Module
M68BSA1-1	MC68000/68010/68451 Personality Module
M68BSA2	MC6800/6802/6808/6809/6809E/6828 Personality Module
M68BSA3	MC68008 Personality Module
M68BSA4	MC6801/6803/68120/68701 Personality Module
M68BSA5	VERSAbus Personality Module
M68BSA6	EXORbus Personality Module

Note: For additional Bus State Analyzer product details, and ordering information, see the related BSA Control Module and Personality Module data sheets.

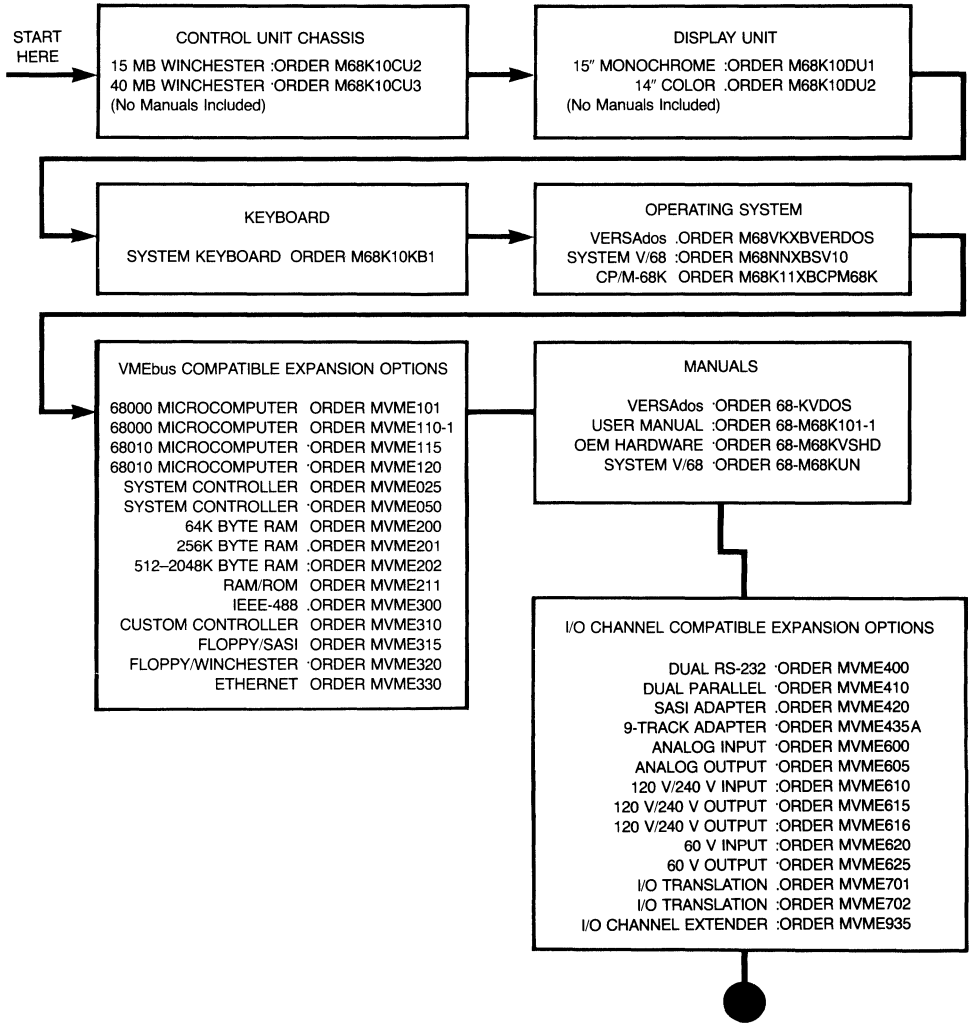
SUPPLEMENTAL DOCUMENTATION

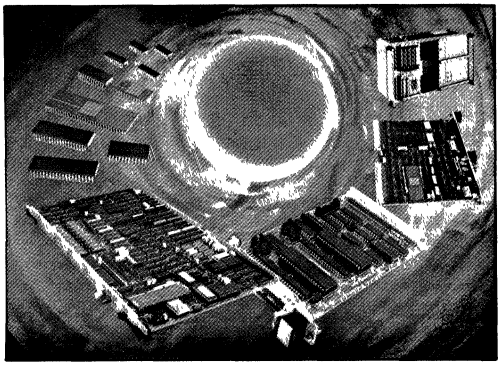
Part Number	Description
M68MVMEBS/D	VMEbus Specification Manual
M68RIOCS/D2	I/O Channel Specification Manual
68-M68KVSHDV1/V2	VME/10 Hardware Documentation Volume 1 and Volume 2.

Contact your local Motorola Semiconductor Sales Office for additional ordering information.

VME/10 OEM CONFIGURATION GUIDE

The VME/10 computer system is offered in two basic configurations including software development tools, operating system, Motorola user guides, etc However, an OEM that is supplying the VME/10 to his end customers in quantity might not want our manuals or software with each and every unit For instance, the OEM may purchase one copy of SYSTEM V/68, add value to the hardware and software and offer this product to his customer Basically, the following guide allows the OEM to pick and choose the optional VME/10 components along with VMEbus and I/O channel expansion options and personalize a system to meet his market needs





Bus Technical Summaries

6

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

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.

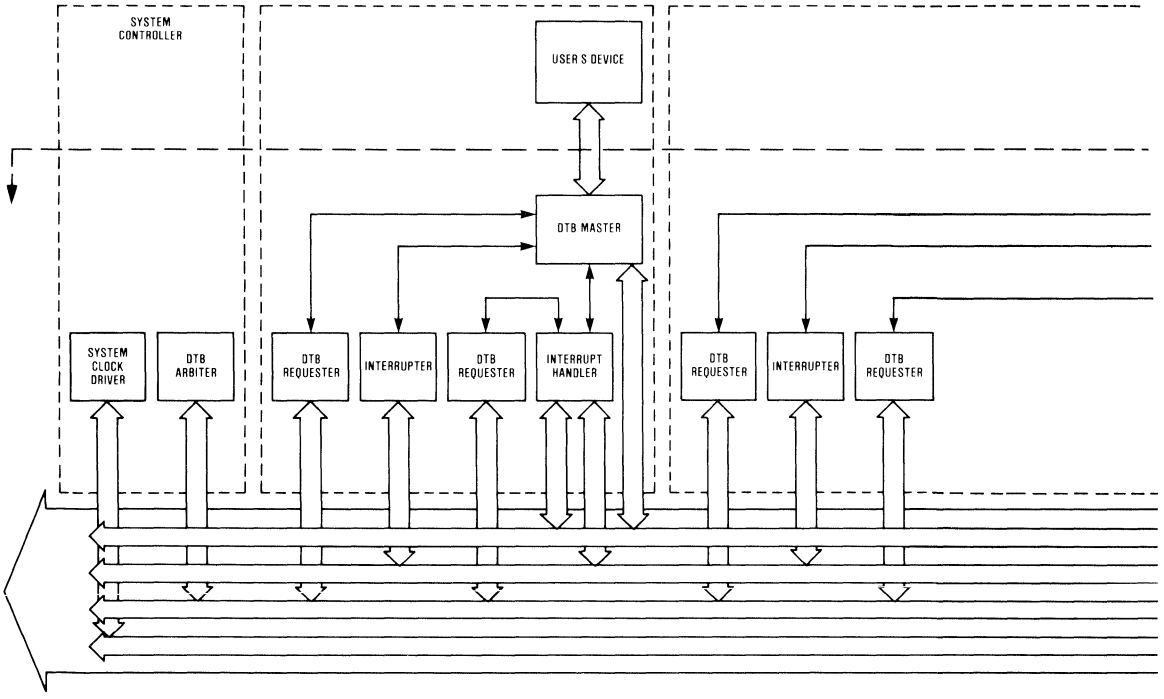
An example of a pair of interlocked signals is DS0* (or DS1*) and DTACK*, which provide interlocking between an addressed SLAVE and the active MASTER.

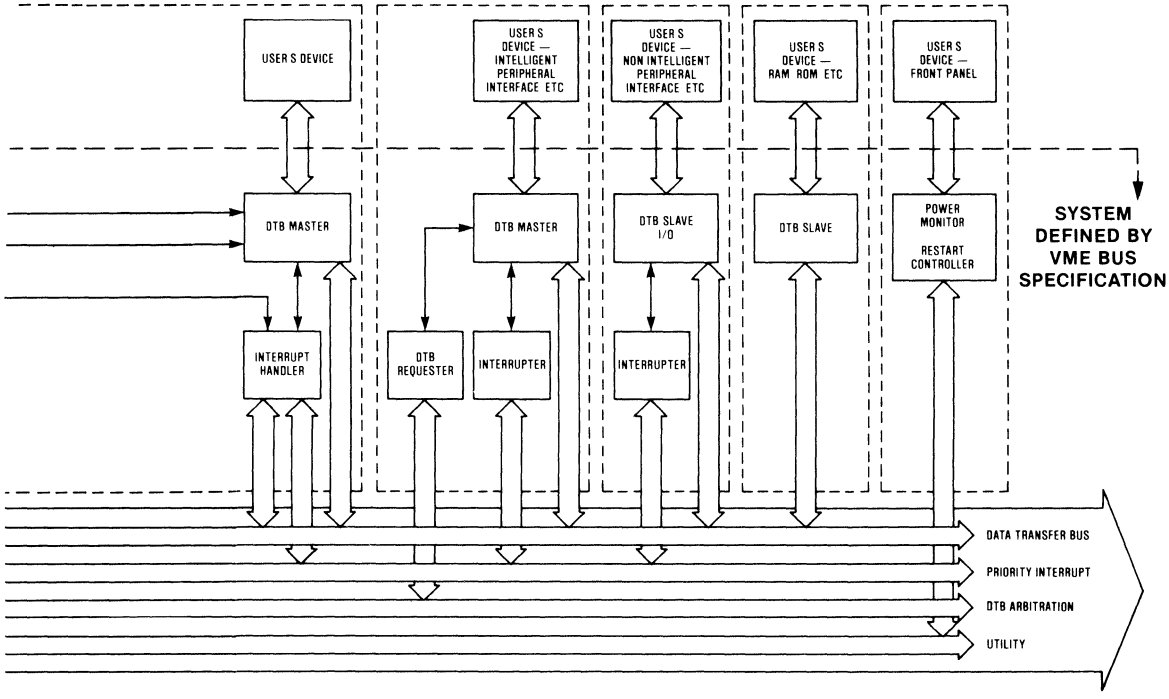
Broadcast Bus Signal

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

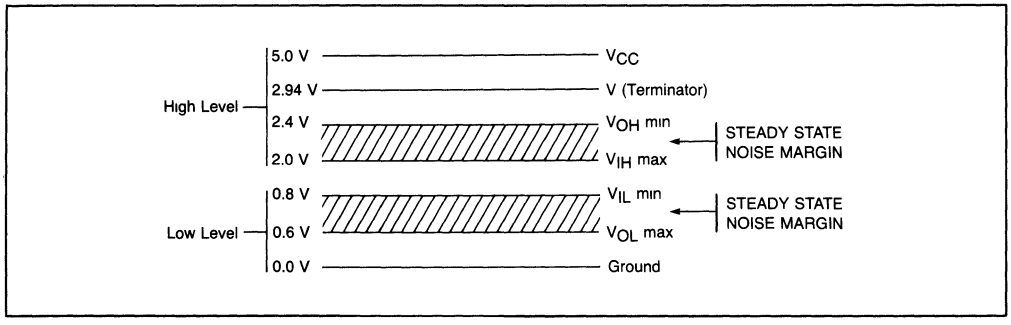
— 0.0 V ≤ Receiver low input level ≤ 0.8 V

— 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

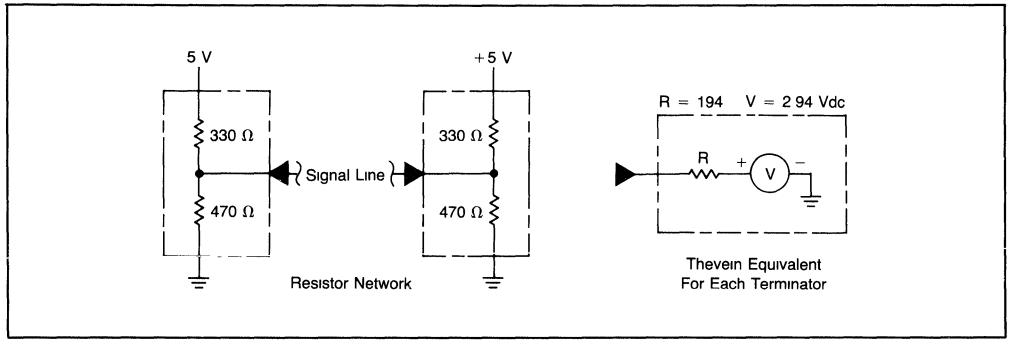
A standard termination is used on each end of all VMEbus signal lines except daisy-chain lines. The termination serves four purposes:

- 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.

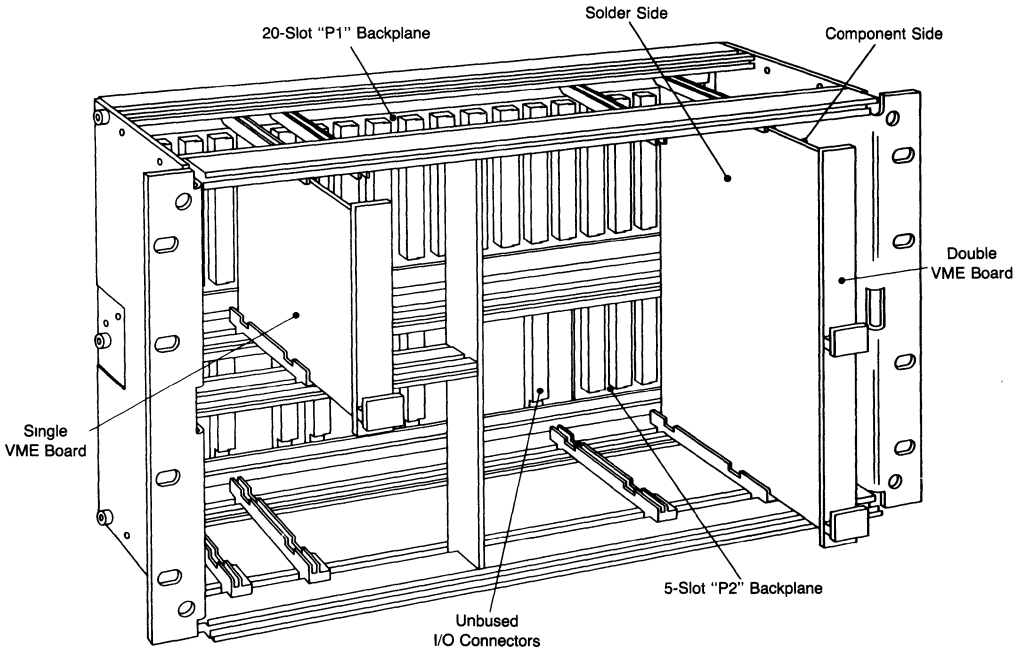
Front Panels

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.

6

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

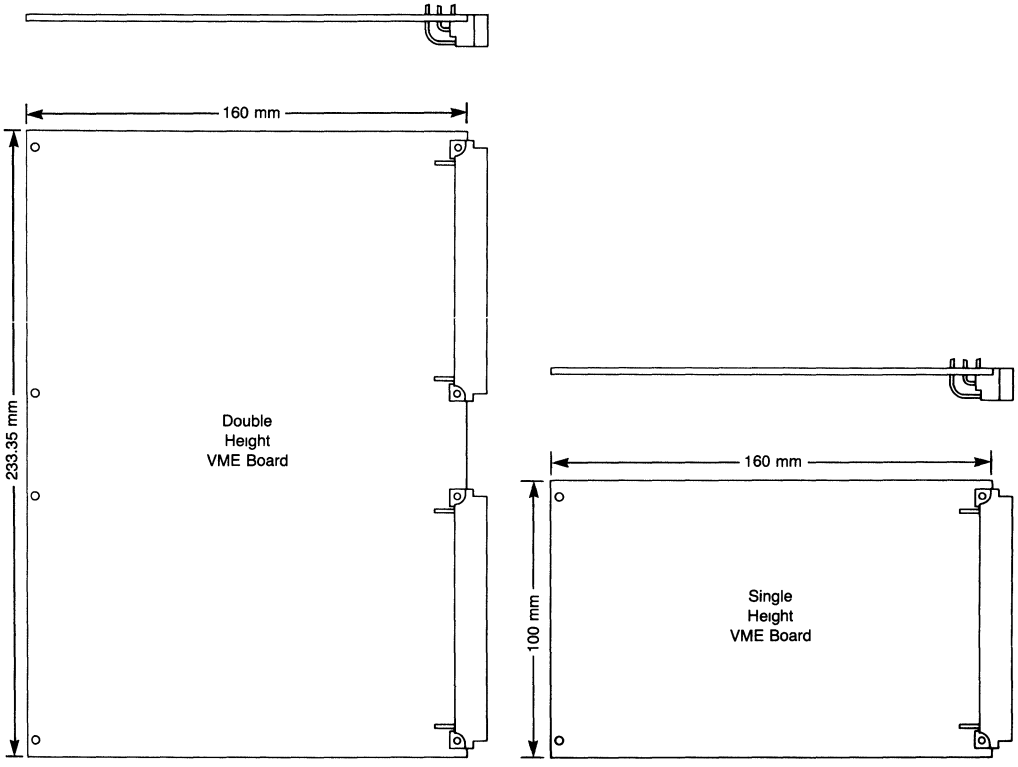
The grid format on the rear is consistent with the board grid and is used to standardize the location of the front panel components for appearance sake.

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.

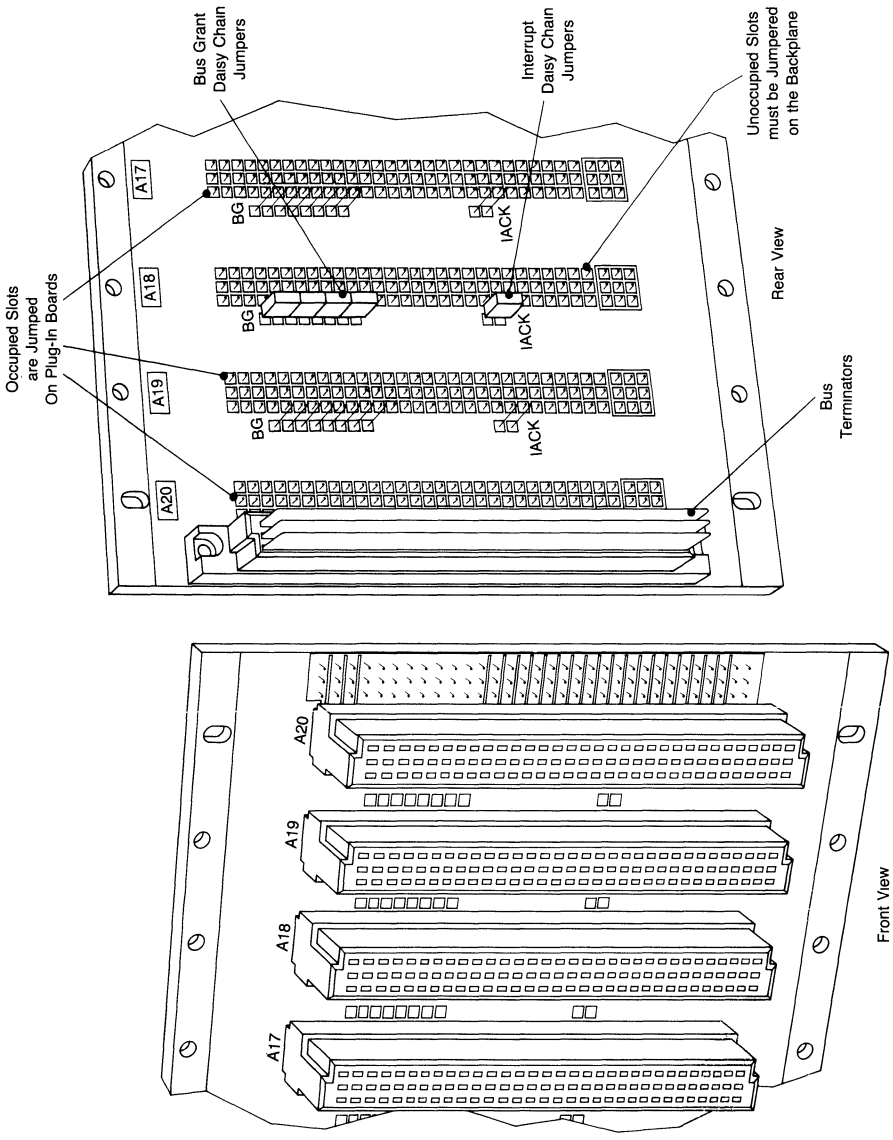


FIGURE 6 — VMEbus Backplane

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

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:

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"

VMXbus High-Speed Local Memory Expansion Bus

- High-Speed Secondary Memory Path for Multiprocessing Efficiency
- Up to 80 mB/Sec Transfer Rate
- 32-bit Data Path
- 16 Mbyte Direct Addressing Range
- Asynchronous Operation
- Up to Six Modules in VMXbus Group
- Allows Module Interconnection by either Ribbon Cable or Hard Backplane
- Accommodates one Primary and (Optionally) one Secondary Master
- VME-based System May Contain Multiple VMXbus Groups

VMXbus OVERVIEW*

The VMXbus is a "subsystem bus" which has been designed for use with VMEbus. It provides a high speed secondary path which is optimized for connecting up to six boards in a subsystem configuration. This subsystem can transfer data from board to board over its VMXbus interface without waiting for and without burdening the primary bus (VMEbus).

The structure of the VMXbus can be described from two points of view: its physical structure and its functional structure.

The physical structure of the VMXbus is composed of an "expanded" VMEbus card rack, one to six plug-in PC boards, and a backplane ribbon cable. The expanded card rack provides both a P1 backplane and a P2 backplane. The P2 backplane buses only the "b" row of pins on the three-row connectors. (The outer two rows of connector pins are left unbused by the P2 backplane: they typically protrude through the rear of the P2 backplane as wirewrap pins and are bused for VMXbus use with a ribbon cable which is pushed onto the wirewrap pins. This cable has from two to six connectors on it. The cable buses the two outer rows of pins of adjacent P2 connectors allowing the boards in these slots to transfer data over the cable.

*This summary was prepared from Revision A, 10/83 of the VMXbus specification.

A board that is inserted from the front of the card rack may have a VMEbus interface, a VMXbus interface, or both. For example, a global I/O board might have only a VMEbus interface, a CPU or a memory board might have both, and a math processor might have only a VMXbus interface. Any board that has a VMXbus interface uses the two outer rows of the J2 connector.

The use of a ribbon cable to bus the P2 connectors allows any group of up to six adjacent slots to function as a subsystem. The user can install two or more cables to create several VMXbuses in a single card rack. Each of these VMXbus subsystems can operate independently of the primary system bus (VMEbus) and independently of each other. It also permits some slots to be used for other purposes such as I/O signals.

Functional Structure

The VMXbus consists of two groups of signal lines, called the ***VMXbus Data Transfer Bus*** (VMX-DTB), an arbitration bus, and a collection of "functional modules" which are configured as required to interface devices to these buses. The functional modules communicate with one another by means of bus signal lines provided by the ribbon cable.

The VMXbus Data Transfer Bus consists of the data and address pathways and associated control signals. Functional modules called MASTERS and SLAVES use the VMX-DTB to transfer data between each other.

Since there can be two MASTERS in a VMXbus group a means must be provided to transfer control of the VMX-DTB between these two MASTERS in an orderly manner, and to guarantee that only one MASTER controls the VMX-DTB at a given time. The Arbitration Bus is defined to do this.

VMXbus DATA TRANSFER BUS

The VMXbus contains a high speed asynchronous parallel Data Transfer Bus (VMX-DTB). This VMX-DTB is used by a PRIMARY MASTER and a SECONDARY MASTER to select SLAVE locations and to transfer data to or from those locations. The PRIMARY MASTER is typically a processor, the SECONDARY MASTER is typically an intelligent I/O controller, and the SLAVE is typically a memory array.

Note The terms PRIMARY MASTER and SECONDARY MASTER are used to distinguish between the MASTER that arbitrates the VMX-DTB and the MASTER that must request, and be granted, the VMX-DTB before using it

The VMX-DTB can be logically divided into address, data, and control line groups:

LWORD/A12 – A11/A23	(23) Multiplexed Address lines + longword (LDS* + UDS* provide a 24th bit)
D00–D31	(32) Data lines
LAS	Lower Address Strobe
UAS*	Upper Address Strobe
LDS*	Lower (Odd Byte) Data Strobe
UDS*	Upper (Even Byte) Data Strobe
ACK*	Acknowledge
DERR*	Data Error
READ	Read/Write Control

6

Data is transferred 8-bits, 16-bits, or 32-bits at a time. Eight bit transfers are called "byte transfers " Byte locations which are addressed with an even address are called "even byte locations" and byte locations addressed with an odd address are called "odd byte locations." Byte transfers to even byte locations are conveyed by D08–D15, and byte transfers to odd byte locations are conveyed by D00–D07. Sixteen bit transfers are called "word transfers." These words are conveyed by D00–D15. Thirty-two bit transfers are called "longword transfers." Longwords are conveyed by D00–D31.

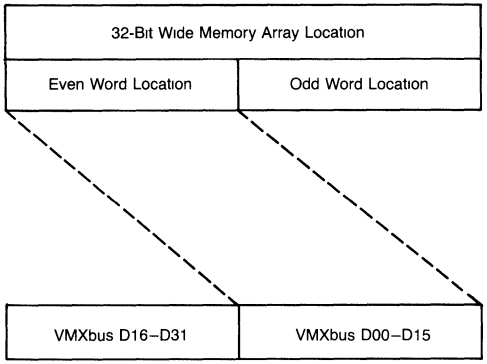
Data is stored into memory 8-, 16-, or 32-bits a time and then read 8-, 16-, or 32-bits at a time. Figure 1 shows that when the most significant 16-bits of a longword location is read from (or written to) the data is transferred over D00–D15 instead of D16–D31 as one might expect This means that 32-bit wide memory boards must have on-board logic to translate the most significant 16-bits of the memory array to the lower 16 data lines of the VMXbus (D00–D15).

When 32-bits of data are written they are stored in a single longword location. When reading this data 16 bits at a time the even word address (A01 = 0) corresponds to the longword location bits D16–D31.

Likewise if two 16-bit words of data are stored in adjacent word locations (an even word location and the next higher odd word location) they may be read as a longword with the even word (A01 = 0) forming the most significant 16-bits (D16–D31).

Two bytes of data may be stored in each word location. The even byte location is the most significant byte of the word location. The odd byte is the least significant. The even byte is selected when UDS* is driven low by the MASTER, the odd byte is selected when LDS* is driven low, and both are selected for a word access when both LDS* and UDS* are low.

FIGURE 1 — When a word access is done the data is always transferred on VMXbus lines D00–D15.



MASTER AND SLAVE MODULES

All VMXbus MASTERS and SLAVES can do 8-bit and 16-bit data transfers. Some MASTERS and SLAVES can also do 32-bit transfers. MASTERS that can only do 8- and 16-bit transfers are called D16 MASTERS, while those that can also do 32-bit transfers are called D32 MASTERS. SLAVES that can handle only 8- and 16-bit transfers are called D16 SLAVES while those that can also handle 32-bit transfers are called D32 SLAVES.

There are four categories of cycles used to transfer data on the VMXbus: single cycles, sequential cycles, indivisible single address cycles, and indivisible multiple address cycles. In order to allow VMXbus board vendors to specify which categories of cycles their boards can do, the following abbreviated notation has been developed:

SGL	— Single cycles supported
SEQ	— Sequential cycles supported
ISA	— Indivisible single address cycles supported
IMA	— Indivisible multiple address cycles supported

These abbreviations are added, as suffixes, to the data width to give a complete description of a board. For example a D32 MASTER capable of all cycles would be called a D32, SGL, SEQ, ISA, IMA MASTER.

Note All MASTERS must support, as a minimum, single cycles All SLAVES must support, as a minimum, single cycles, indivisible single address cycles, and indivisible multiple address cycles

D16 MASTERS and D16 SLAVES

The D16 MASTER definition allows 16-bit processors and 16-bit DMA devices to be interfaced to the VMXbus. The D16 SLAVE provides 16-bit memory. Since 16-bit processors sometimes access memory one byte at a time, D16 SLAVES must allow their even and odd byte locations to be accessed one byte at a time. Whenever a byte transfer is done to or from an odd byte location, it is done over D00–D07. Whenever a byte transfer is done to or from an even byte location it is done over D08–D15. (Transferring the even and odd bytes over the same data lines that are used in 16-bit transfers minimizes the multiplexing logic required on the D16 memory slave.)

D32 MASTERS and D32 SLAVES

D32 MASTERS can do LONGWORD (32-bit) transfers over D00–D31, but they can also act like D16 MASTERS (i.e. they can do byte transfers and word transfers: they do odd byte transfers over D00–D07, even byte transfers over D08–D15, and word transfers over D00–D15). This means that they can access D16 slaves as well as D32 SLAVES.

D32 SLAVES allow each of their locations to be accessed one byte at a time, one word at a time, or one longword at a time. When doing byte transfers they transfer even bytes over D08–D15, and their odd bytes over D00–D07. When doing word transfers they always use D00–D15. This means that D32 SLAVES must have logic that allows data in the upper 16-bits of its 32-bit memory array to be accessed through D00–D15. Because D32 SLAVES allow access in all of these ways they can be accessed by D16 MASTERS as well as D32 MASTERS.

Mixing D16 and D32 MASTERS and SLAVES

VMXbus systems can be built with mixtures of D16 and D32 MASTERS and SLAVES. This can be done in any combination, and these boards can still communicate data successfully, with one qualification: if a D32 MASTER attempts a 32 read or write on D16 SLAVE that SLAVE must respond by driving DERR* low, indicating an error. (The D32 MASTER can still transfer data to and from this SLAVE by doing it 16 bits at a time.)

SEQ MASTERS and SEQ SLAVES

SEQ MASTERS are able to do sequential access cycles. Most SEQ MASTERS are DMA controllers, but there is no reason why a processor board couldn't be designed to use sequential cycles. (e.g. a processor board with a cache might read blocks of data using the sequential mode.)

SEQ SLAVES are able to respond to sequential access cycles. During a sequential access cycle a SEQ SLAVE latches the initial address from the bus and then increments that address each time data is read or written. Since there are actually three types of sequential access cycles (byte, word, and longword) the SEQ SLAVE must increment its internal counter in such a way that the next access results in a transfer on the next location in memory. (Since memory SLAVES may have 16-, 32- or even 64-bit wide internal memory arrays, the VMXbus spec doesn't dictate how the internal address counter is incremented.)

Mixing SEQ and non-SEQ MASTERS and SLAVES

VMXbus systems can be built with mixtures of SEQ and non-SEQ MASTERS and SLAVES. If an SEQ MASTER attempts a sequential cycle on a non-SEQ SLAVE that SLAVE will respond by driving DERR* low, indicating an error. (The SEQ MASTER can still transfer data to and from this SLAVE by doing single cycles.)

IMA MASTERS

IMA MASTERS are able to do indivisible multiple address cycles, which allow a VMXbus MASTER to access VMXbus SLAVE locations while excluding all other MASTERS from access to the same SLAVE locations. All SLAVES must be able to handle IMA cycles. Many VMXbus SLAVES are "dual ported", allowing VMXbus MASTERS to access their locations through one port and VMEbus MASTERS to access the same locations through the second port. These dual ported boards must be designed so that whenever they respond to an indivisible multiple address cycle on the VMXbus they "lock out" all accesses from VMEbus until all of the "reads" and "writes" of the cycle are completed.

DATA TRANSFER CYCLE TYPES

Putting aside, for the moment, the fact that data may be transferred over VMXbus one byte, two bytes, or four bytes at a time there are six basic cycle types:

1. Single write cycle
2. Single read cycle
3. Sequential write cycle
4. Sequential read cycle
5. Indivisible single address cycle
6. Indivisible multiple address cycle

There are some cases in which a MASTER may direct a SLAVE to do something that it cannot do. For example, a D16 SLAVE may be told to do a 32-bit read, or a non-SEQ SLAVE may be told to do a sequential cycle. When this happens the SLAVE "aborts" the cycle. This results in six more cycle types:

7. Aborted single write cycle
8. Aborted single read cycle
9. Aborted sequential write cycle
10. Aborted sequential read cycle
11. Aborted indivisible single address cycle
12. Aborted indivisible multiple address cycle

A SLAVE may acknowledge a cycle from the MASTER and then find itself unable to complete the cycle. For example, it may detect a parity error when retrieving data from its memory array. To indicate this to the MASTER it aborts the cycle This sequence results in six more cycle types:

13. Late aborted single write cycle
14. Late aborted single read cycle
15. Late aborted sequential write cycle
16. Late aborted sequential read cycle
17. Late aborted indivisible single address cycle
18. Late aborted indivisible multiple address cycle

If no SLAVE responds (e.g. the SLAVE has malfunctioned, or there is no SLAVE at the address provided) a timer on the MASTER board terminates the cycle, resulting in six more cycle types:

19. Timed-out single write cycle
20. Timed-out single read cycle
21. Timed-out sequential write cycle
22. Timed-out sequential read cycle
23. Timed-out indivisible single address cycle
24. Timed-out indivisible multiple address cycle

A MASTER may begin a cycle BEFORE it has determined whether the location it needs to access is on the VMXbus or elsewhere. This feature can improve performance on processor boards where some addresses result in accesses over the VMXbus while others result in on-board accesses or accesses to other buses, such as VMEbus.

The processor begins such a cycle in the usual manner, by sending the "lower address" to the VMXbus SLAVES, before it knows whether or not a VMXbus access is required. If, after decoding the address, it determines that a VMXbus access is not required, then it doesn't proceed with the rest of the cycle. Such a cycle is said to be "withdrawn" by the MASTER. Since all six of the basic cycle types begin in the same way (with the transmission of the lower address) they all result in similar timing when they are withdrawn. This results in one more additional cycle type:

25. Withdrawn cycle

VMXbus ARBITRATION BUS

The VMXbus contains two arbitration lines which are used to transfer control of the VMX-DTB between the PRIMARY MASTER and SECONDARY MASTER:

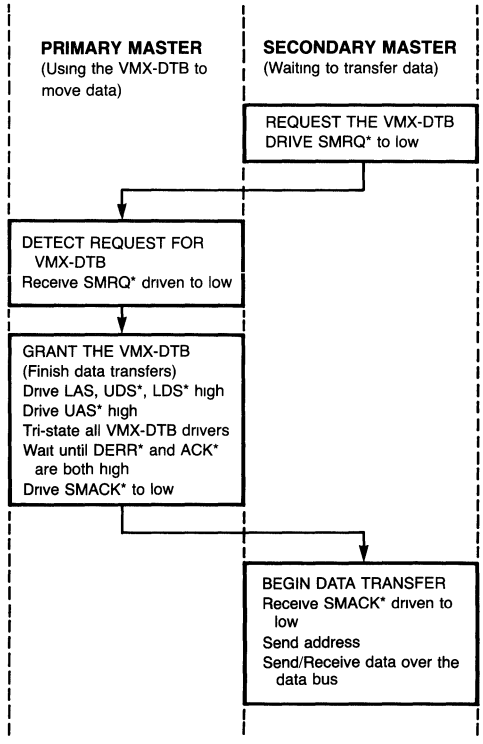
SMRQ*	SECONDARY MASTER bus request
SMACK*	SECONDARY MASTER acknowledge

These lines are collectively called the "arbitration bus". The SECONDARY MASTER drives SMRQ* low to when it needs to read or write data to VMXbus memory. The PRIMARY MASTER may drive SMACK* low in response to the low level on SMRQ* to permit the SECONDARY MASTER to use the VMX-DTB.

The PRIMARY MASTER drives SMACK* high whenever the system is reset by the VMXbus SYSRESET* line and maintains it high at least until SMRQ* is driven low by the SECONDARY MASTER. The SECONDARY MASTER drives SMRQ* high when the system is reset and maintains it high at least until SYSRESET* goes high again

Since only two MASTERS are allowed on each VMXbus, (a PRIMARY MASTER and a SECONDARY MASTER) there are only two basic arbitration sequences. The first is when the PRIMARY MASTER grants control of the VMX-DTB to

FIGURE 2



the SECONDARY MASTER. The second is when the SECONDARY MASTER gives control of the VMX-DTB back to the PRIMARY MASTER.

PRIMARY MASTER grants VMX-DTB to SECONDARY MASTER.

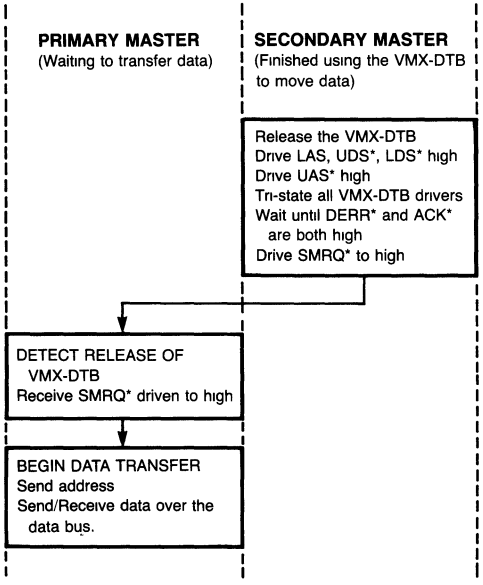
Figure 2 shows the flow diagram for the case where the SECONDARY MASTER is granted the bus. When the sequence begins, the PRIMARY MASTER is using the VMX-DTB. When the SECONDARY MASTER drives SMRQ* to low and when the PRIMARY MASTER has finished using the VMX-DTB, it drives SMACK* low, granting the VMX-DTB to the SECONDARY MASTER.

As long as the SECONDARY MASTER holds SMRQ* low it may continue to use the VMX-DTB without interruption. This allows it to transfer data to or from the memory at the maximum data rate, since it need not check before each transfer to see whether it still has control of the VMX-DTB.

PRIMARY MASTER receives VMX-DTB from SECONDARY MASTER

Figure 3 shows the flow diagram for the case where the SECONDARY MASTER relinquishes control of the data transfer lines. When the sequence begins, the SECONDARY MASTER is transferring data between itself and VMXbus memory. When the transfer is complete, and it has tri-stated all of the VMX-DTB drivers, it drives SMRQ* high, signaling to the PRIMARY MASTER that it has finished using the VMX-DTB. The PRIMARY MASTER then drives SMACK* high, signaling to the SECONDARY MASTER that its high level on the SMRQ* line has been detected and that it may request the bus again when needed.

FIGURE 3



ELECTRICAL SPECIFICATIONS

VMXbus signals are normally driven with TTL family drivers, although any technology which complies with the VMXbus specification may be used. Recommendations and examples are included in the full specification to aid the designer in obtaining optimum system performance.

Power Distribution

Power and power-return distribution for the VMXbus is done through the VMEbus P1 and P2 backplanes. (This leaves all of the P2 ribbon cable lines available for signals and signal grounds.) The pins of the P1 and P2 connectors, which are used for power distribution, the voltage tolerances, the maximum ripple, and the current carrying capacity of each power pin are the same as those described in the VMEbus Specification Manual.

Electrical Signal Characteristics

Other than the power supply lines, all VMXbus signals are limited to positive levels between 0.0 and 5.0 volts. The signal levels are:

Signal	Minimum	Maximum
Driver low output level		0.6 V
Receiver low input level		0.8 V
Driver high output level	2.4 V	
Receiver high input level	2.0 V	

Line Driver Requirements

There are three types of signal line drivers used on the VMXbus: ***totem-pole drivers, tri-state drivers,*** and ***open collector drivers.***

Totem-pole drivers sink current when driving a signal line to its low level and source current when driving it to its high level. They are used to drive signal lines which are always driven by the same VMXbus board. SMRQ* and SMACK* are the two VMXbus lines driven by totem-pole drivers.

VMXbus tri-state drivers differ from totem-pole drivers in two respects: they can sink and source more current than

the totem-pole drivers and they can go to a high impedance state (driver turned off) in addition to the low and high logic states. Tri-state drivers are used to drive lines that can be driven by several different boards on the VMXbus. (Only one of these drivers is ever driving a signal line at any one time.) The address bus, the data bus and most of the other signal lines on the VMXbus are driven by tri-state drivers.

Open collector drivers sink current when driving a signal line to its low level but source no current to drive the signal line high. Pull-up resistors on the PRIMARY MASTER ensure that the signal line voltage rises to a high level whenever the open collector line is not driving it to a low level. ACK* and DERR* are the two VMXbus lines driven by open collector drivers.

Signal Line Terminations

The following ten signal lines of the VMXbus are terminated on PRIMARY MASTER boards with 330 ohm pull-up resistors. Other boards must not pull these lines up.)

READ Tri-state
LOCK* Tri-state
LAS Tri-state
UAS* Tri-state
UDS* Tri-state
LDS* Tri-state
DERR* Open-collector
ACK* Open-collector

These pull-up resistors pull their respective lines to a high level whenever a tri-state or open collector driver turns off.

Isolation Lines

VMXbus includes ten signal lines called Static Isolation Lines (STIL0–STIL9). The purpose of these lines is to prevent cross-talk between adjacent signal lines whose transitions take place at different times. All VMXbus boards must provide a 0.01 plus a 0.1 μF by-pass capacitor between each of these lines and its on-board ground grid. The length of the trace connecting this capacitor to the connector plus the length of the trace connecting the other side of the capacitor to the ground grid must not exceed 2 inches.

MECHANICAL SPECIFICATIONS

Certain requirements must be met to assure that VMXbus boards will be mechanically compatible with the VMEbus systems into which they are installed. As in the case of the VMEbus, the following terminology applies:

Backplane — A board into which 96-pin connectors are installed. The backplane of primary interest to VMXbus users is the VMEbus "P2 backplane." This is the lower backplane in a double-high VMEbus card rack which buses only the center row pins of each "P2 connector."

Ribbon cable — A 64 wire cable which has installed on it two to six 64-pin connectors at 0.8" intervals. This cable is installed on the back of the "P2 backplane" to bus the two outside rows of pins on adjacent P2 connectors together.

VMXbus board — A PC board which is plugged into the backplane and communicates with other VMXbus boards installed in the same backplane.

The "front" of a backplane is the side from which the VMXbus boards are inserted into the connectors.

The "rear edge" of a VMXbus board is the edge which provides the on-board connector for mating with the connector on the backplane.

VMXbus Backplane

The VMXbus has been designed for use with a double high VMEbus card. This double high rack has two backplanes: an upper one which buses all three rows of pins on the P1 connectors and a lower one which buses only the center row of pins on the P2 connectors. (This VMEbus "P2 backplane" does not bus the two outer rows of the P2 connectors.) Since these two outer rows aren't bused by the P2 backplane they may be used as I/O pins. When a user wants to bus these two rows of pins for a secondary bus, such as VMXbus, a ribbon cable is installed to form a "private bus."

VMXbus is defined to provide just such a private bus. One or more 64 wire ribbon cables can be installed on the P2 connectors to "bus" the two outer rows of pins. Each ribbon cable may have from two to six connectors on it, and several of these cables can be used in a SINGLE CARD RACK to provide several VMXbuses.

The ribbon cable provides all of the conductors needed to send 24-bit addresses from the MASTERS to the SLAVES and transfer up to 32-bits of data at a time. This ribbon cable connects a maximum of six VMXbus compatible boards which are installed at 0.8 inch intervals. The length of the cable used to connect these boards must not exceed 5.0 inches.

All VMXbus configurations assume the existence of a P2 backplane. Power for the VMXbus boards is provided by this P2 backplane through the center row (row "b") of its VMXbus connector. (Double-high VMXbus boards typically also draw power through their P1 connector.)

VMXbus Eurocard Description

VMXbus boards may be either single or double high, just like VMEbus boards. The dimensions of these boards must conform to those given for the single and double high boards in the VMEbus specification. All of the dimensions given there including maximum board warpage, maximum component height, etc. apply to VMXbus boards also. (All PRIMARY MASTER and SECONDARY MASTERS are double-high

boards, since they monitor the level of SYSRESET* on the VMEbus.)

VMXbus Pin Assignment

VMXbus uses the two outside rows (rows a and c) of the P2 connector for all signal transmissions. VMXbus boards may draw power from the center row (row b) of P2. (These +5 V power and GND pins are the same as the ones specified in the VMEbus specification.) In addition to these pins, a double-high VMXbus board may also draw power from its P1 connector. (See the VMEbus specification for the +5 V and GND pin locations on P1.) Pin assignments by function are summarized in Figure 4.

FIGURE 4 — VMXbus Connector Pin Assignments

Pin #	ROW a	ROW b	ROW c
1	DB00	+5 V	DB01
2	DB02	GND	DB03
3	DB04	VMEbus signal	DB05
4	DB06	VMEbus signal	DB07
5	DB08	VMEbus signal	DB09
6	DB10	VMEbus signal	DB11
7	DB12	VMEbus signal	DB13
8	DB14	VMEbus signal	DB15
9	DB16	VMEbus signal	DB17
10	DB18	VMEbus signal	DB19
11	DB20	VMEbus signal	DB21
12	DB22	GND	DB23
13	DB24	+5 V	DB25
14	DB26	VMEbus signal	DB27
15	DB28	VMEbus signal	DB29
16	DB30	VMEbus signal	DB31
17	WRITE*	VMEbus signal	STIL0
18	LOCK*	VMEbus signal	STIL1
19	LWORD/A12	VMEbus signal	STIL2
20	A02/A14	VMEbus signal	A01/A13
21	A04/A16	VMEbus signal	A03/A15
22	A06/A18	GND	A05/A17
23	A08/A20	VMEbus signal	A07/A19
24	A10/A22	VMEbus signal	A09/A21
25	STIL3	VMEbus signal	A11/A23
26	STIL4	VMEbus signal	LAS*
27	STIL5	VMEbus signal	UAS*
28	STIL6	VMEbus signal	UDS*
29	STIL7	VMEbus signal	LDS*
30	STIL8	VMEbus signal	DERR*
31	STIL9	GND	ACK*
32	SMRQ*	+5 V	SMACK*

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.

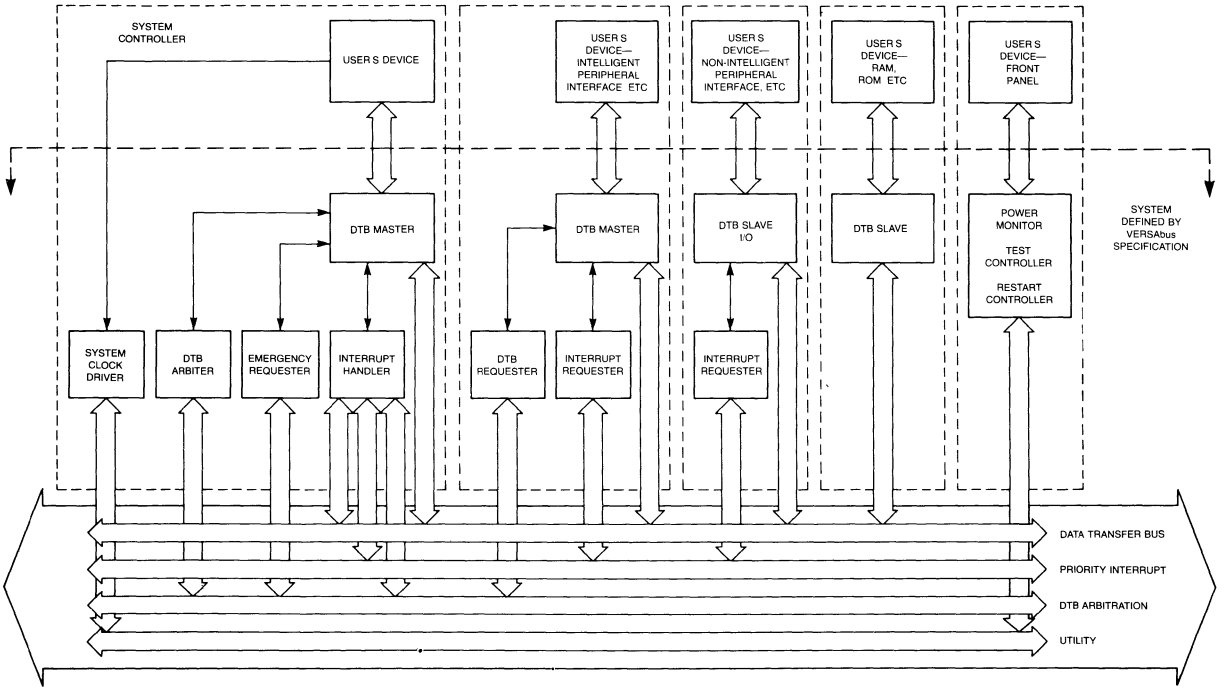
Each VERSAbus based system has a designated "System Controller." The controller function can contain the following functional modules:

- 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.

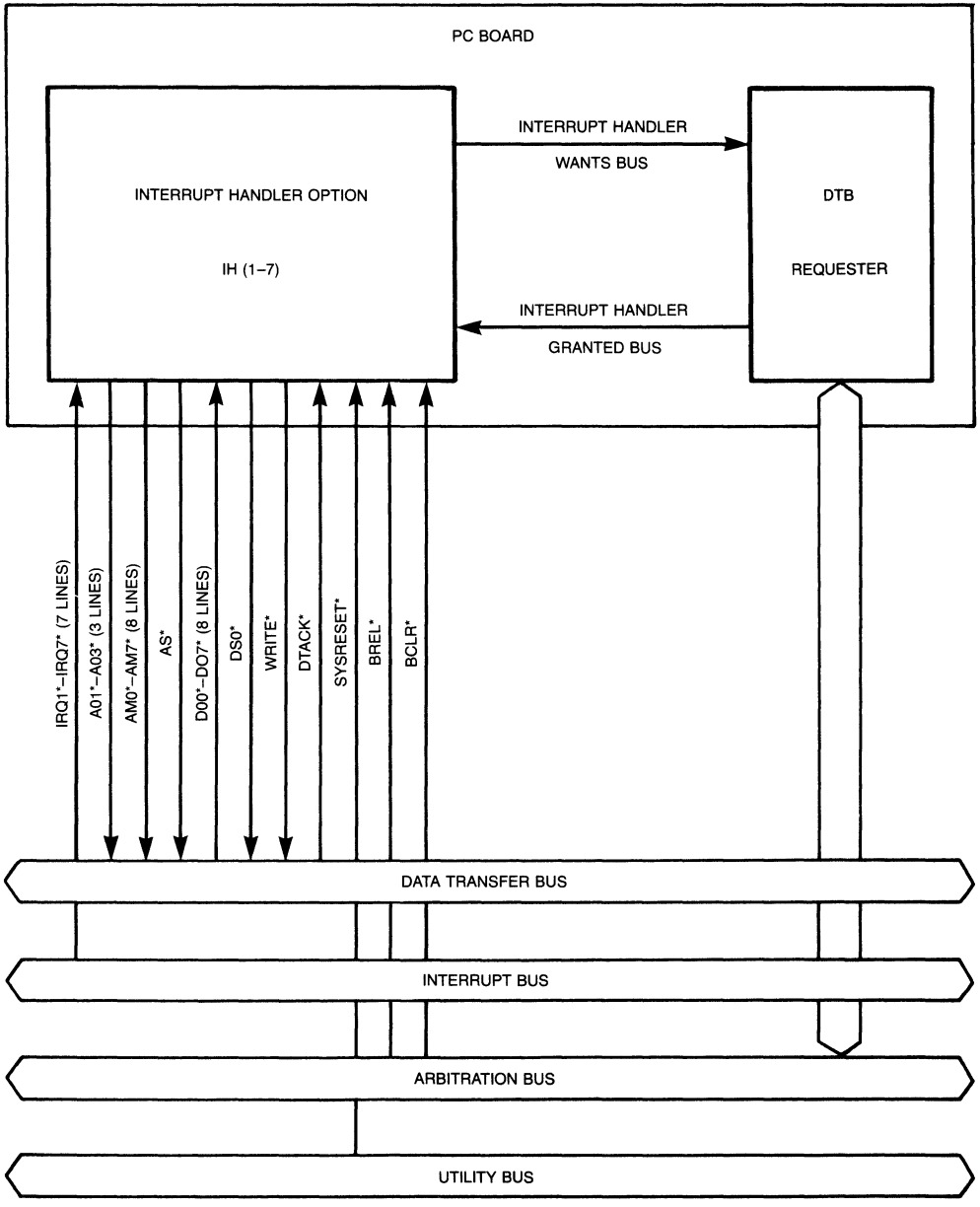
In a single processor or single handler system, the executive software and all the interrupt routines are executed by one processor.

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 Main logic power supply Primary auxiliary supply Standby power Analog application supply	Regulated dc voltages available on VERSAbus: +5 Vdc ± 12 Vdc ± 5 V STDBY ± 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 Board Thickness Component Lead Length Component Height Board Warpage	 PCB: 0.900 minimum WWB. 1 4 minimum 0.062 ± 0.005 inch Component leads protruding through back of VERSAboards not to exceed 0.100 inch Component height on front of each VERSAboard not to exceed 0.50 inch 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 2.4		V V	Source 15 mA terminated 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 2.4		V V	Source 15 mA terminated 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.)				

6

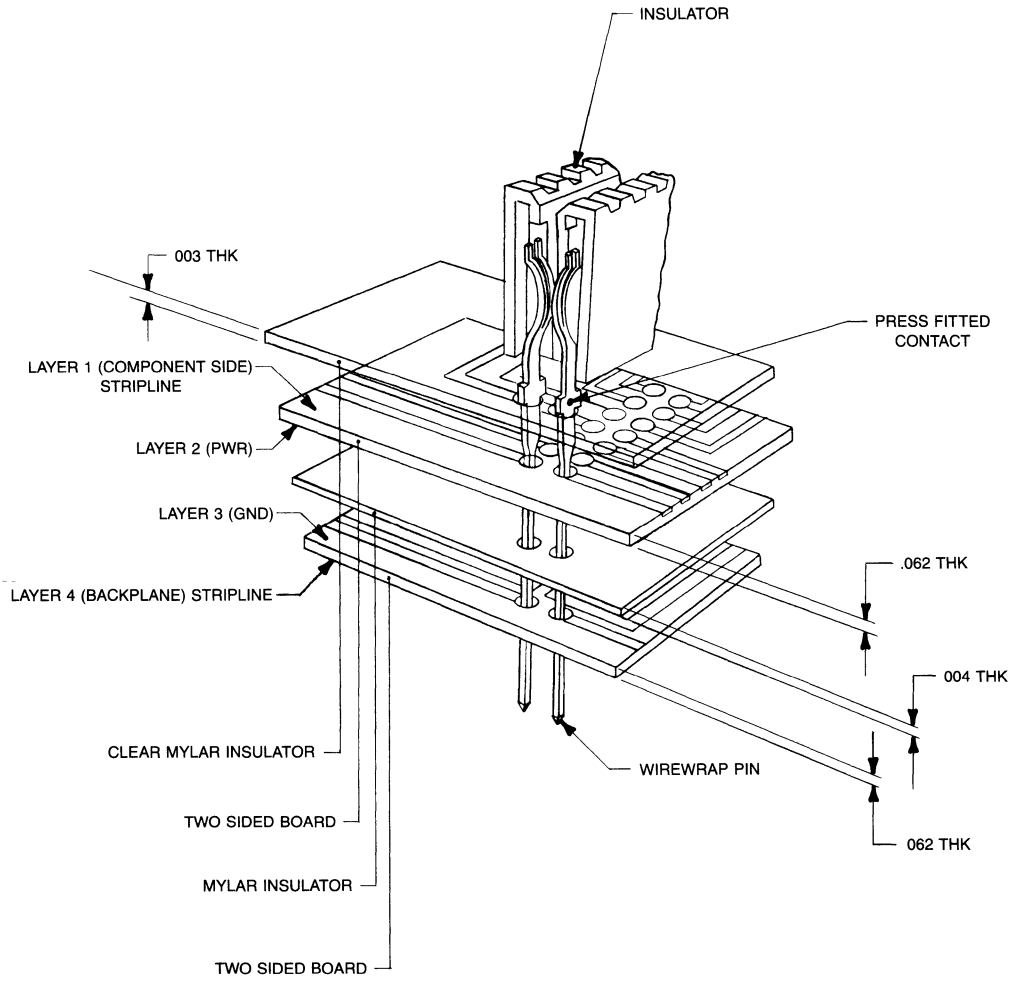
Mechanical Considerations

Backplane Construction Techniques

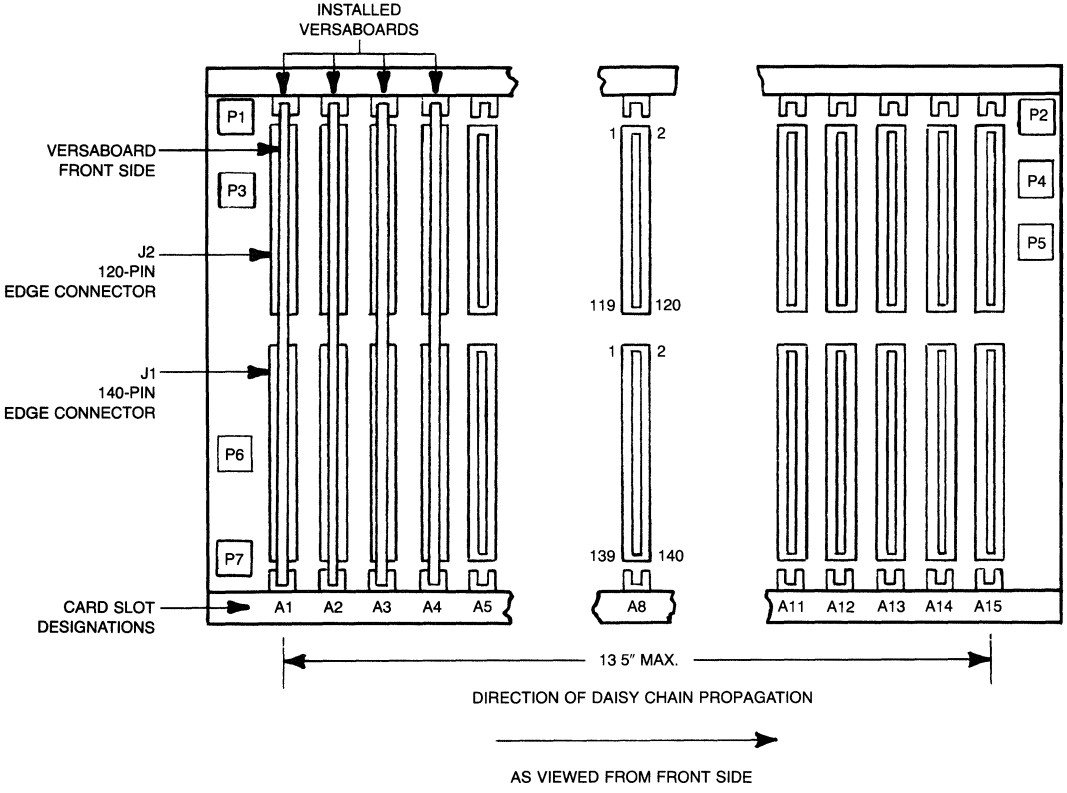
Many backplane construction techniques are available to the designer. Backplane construction can be of a multilayer laminated or unlaminated design. Following is a description of a multiple layer design composed of two discrete, two-sided, glass epoxy boards separated by a mylar insulator with a mylar 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.

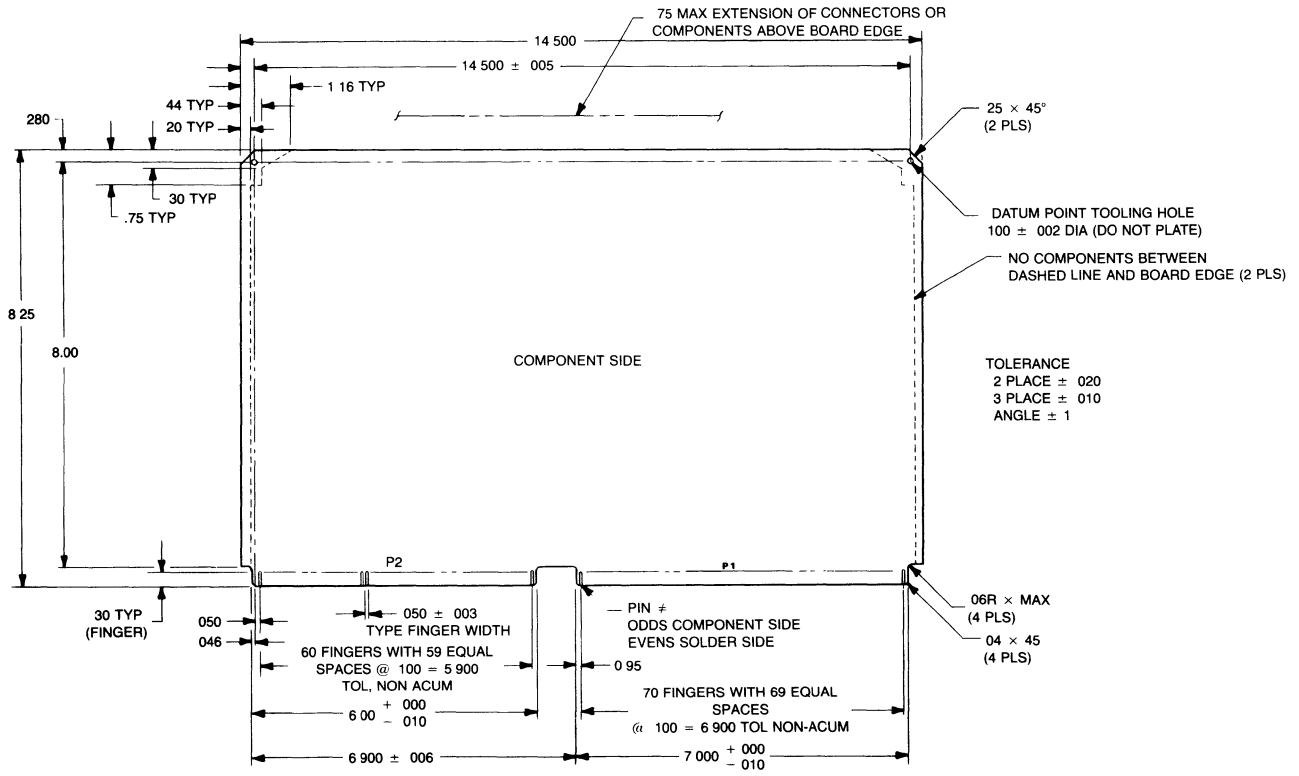
Typical Multilayer Backplane/PCB Construction Technique



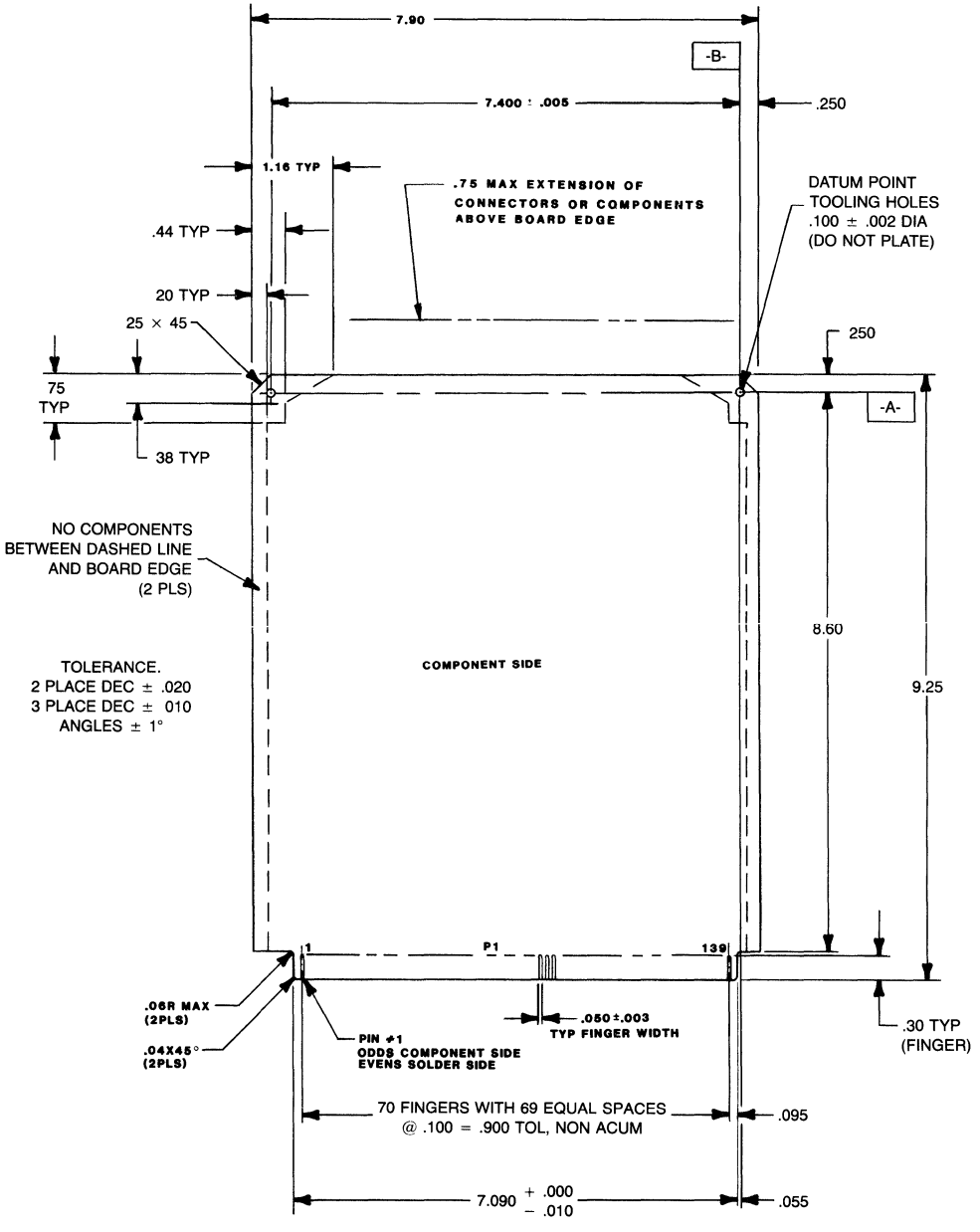
Backplane Reference Designations and Pin Numbering Standard



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

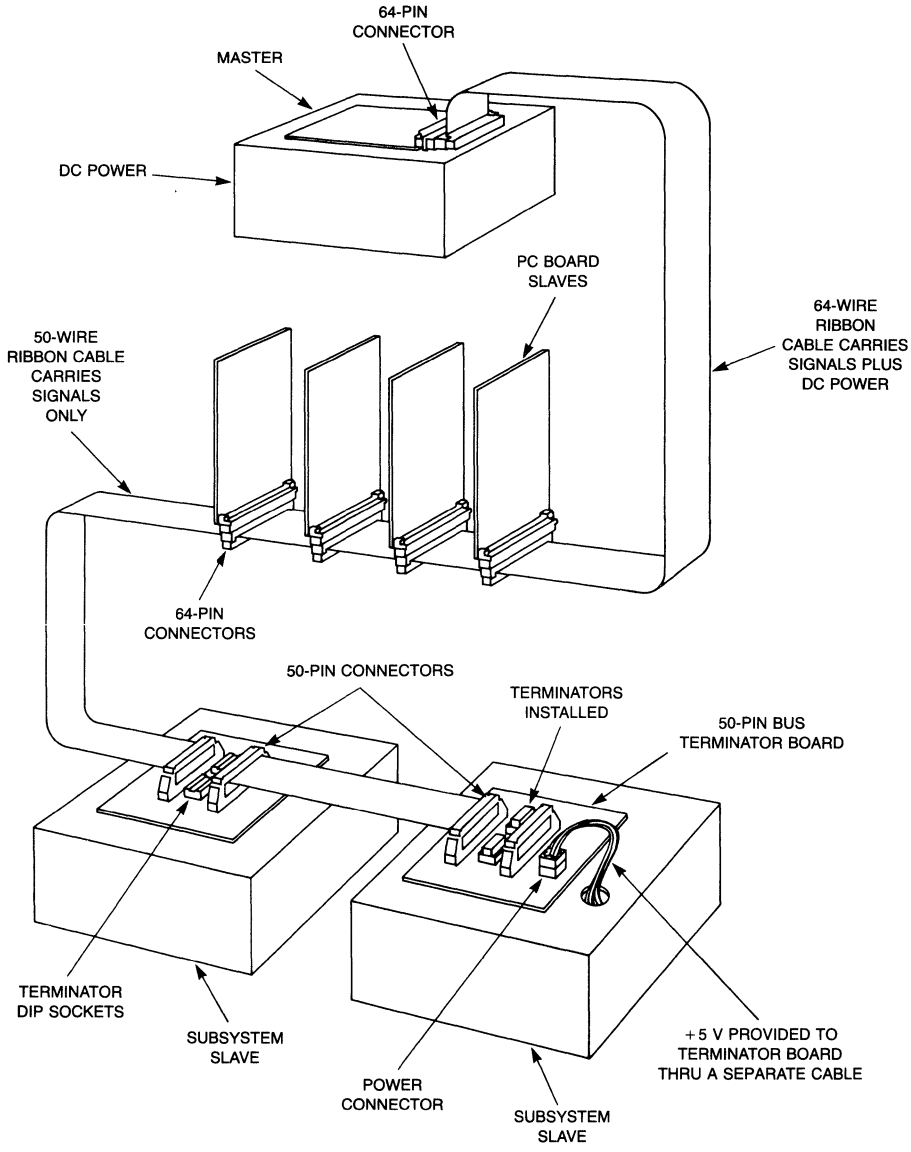
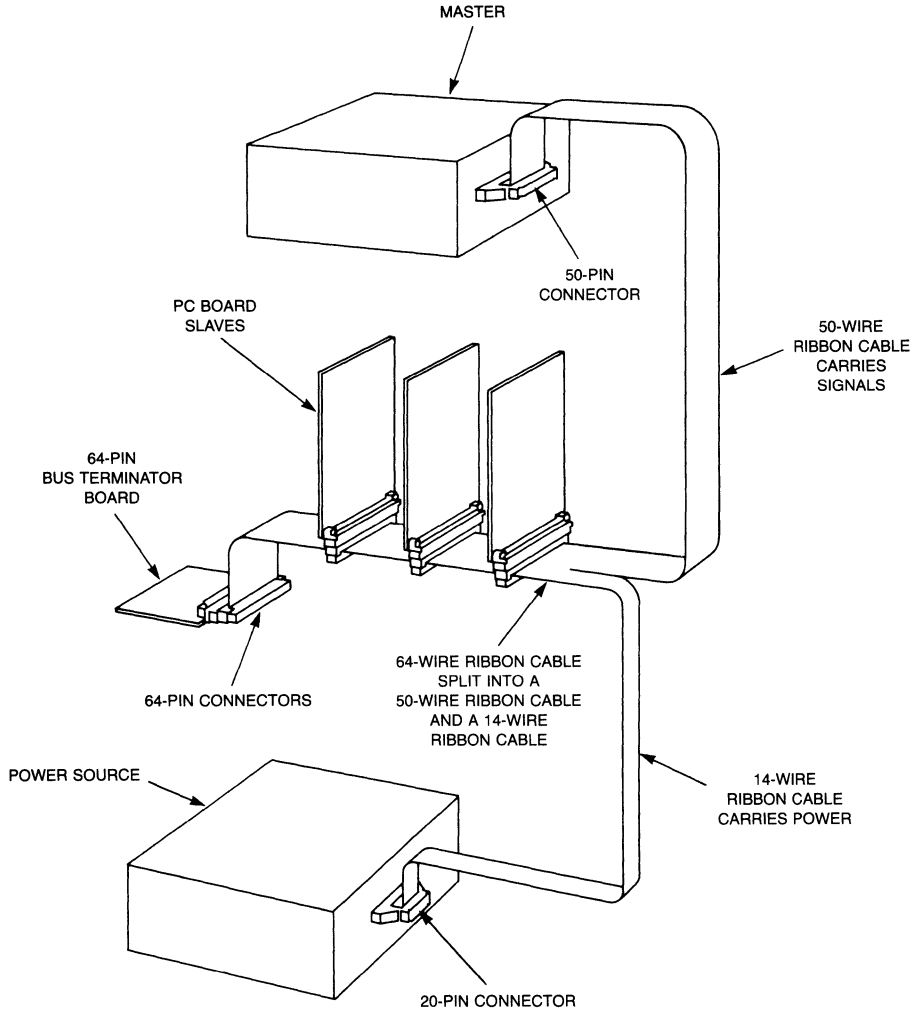


FIGURE 2–Alternate Power Wiring Configuration



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 $\pm$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

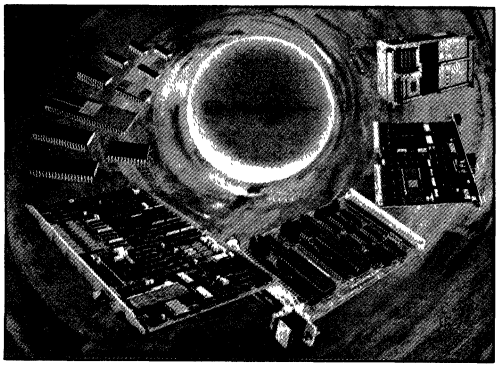
For additional information pertaining to the I/O Channel, refer to the I/O Channel Specification Manual, Motorola publication number M68RIOCS.

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

USER DOCUMENTATION
16/32-Bit
Microcomputer System Components

VMEbus COMPATIBLE MODULES

MVMEBUG/D3	VMEbug Debugging Packages User's Manual
MVME025/D1	MVME025 System Controller User's Manual
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
MVME115M/D1	VMEmodule Monoboard Microcomputer 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
MVME300/D1	MVME300 GPIB Controller With DMA User's Manual
MVME310/D1	VMEmodule Intelligent Controller For Custom Interfacing User's Manual
MVME315/D1	MVME315 Intelligent Disk Controller User's Manual
MVME330/D1	VMEmodule LAN Controller User's Manual
MVME435	VMEmodule 9-Track Magnetic Tape Adapter User's Manual
MVME900	940 941 942 943 944 911

VERSAbus COMPATIBLE MODULES

M68KVAM/D1	VERSAbus Adapter Module User's Guide
M68KVBUG/D2	VERSAbug Debugging Package User's Manual
M68KVBUG2/D1	VERSAbug 2.n Debugging Package
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

VERSAbus (cont.)

M68KVMCHU/D1	Installation Instructions for VERSAmodule System Unified Power Control
M68KVMESH/D2	VERSAmodule Chassis, Card Cage, Power Supply and Power Monitor 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
M68KVMTPS/D1	Todd Products Corporation Switching Power Supply User's Manual
M68KVM01/D1	VERSAmodule Monoboard Microcomputer User's Guide
M68KVM02/D2	VERSAmodule Monoboard Microcomputer User's Manual
M68KVM03/D1	VERSAmodule Monoboard Microcomputer User's Manual
M68KVM04/D1	VERSAmodule 32-Bit Monoboard Microcomputer User's Manual
M68KVM10/D1	32K/64K/128K Byte Dynamic Memory Module User's Manual
M68KVM11/D2	M68KVM11-1/-2 256K/512K Byte Dynamic RAM Memory Module User's Manual
M68KVM12/D1	VERSAmodule 1024K/4096K Byte RAM User's Manual
M68KVM13/D1	VERSAmodule 1024K/4096K Byte Dynamic RAM W/RAMbus User's Manual
M68KVM20/D2	Floppy Disk Controller Module User's Manual
M68KVM21/D2	Universal Disk Controller User's Manual
M68KVM30/D2	Multichannel Communications Module User's Manual
M68KVM31/D1	Eight-Channel Intelligent Communications Module User's Manual
M68KVM33/D1	VERSAmodule LAN Controller User's Manual
M68KVM60/D2	Universal Intelligent Peripheral Controller Module User's Manual
M68KVM80/D2	VERSAmodule Combination Memory, I/O, and Time-of-Day Clock 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 Adapater Module User's Manual
MVME435/D2 Buffered 9-Track Magnetic Tape Adapter Module User's Manual
MVME600/D1 MVME600 Analog Input Module — MVME601 A/D Expander Module User's Manual
MVME605/D1 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
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/D8 M68000 CRT Text Editor User's Manual
M68KIPCS/D2 M68000/IPC Command Channel Software Interface Reference Manual

VERSAdos (cont.)

M68KRIODRV/D1 RIO1 Device Driver Software User's Manual
M68KRMS68K/D8 M68000 Family Real-Time Multitasking Software User's Manual
M68KSYMBG/D3 SYMbug/A and DEbug Monitor Reference Manual
M68KSYSGEN/D6 System Generation Facility User's Manual
M68KVMSG/D2 VERSAdos Messages Reference Manual
M68KVOVER/D4 VERSAdos Overview
M68KVSEDT/D1 VME/10 Text Editor User's Manual
M68KVSF/D5 M68000 Family VERSAdos System Facilities Reference Manual
MVMECNFG1/D1 VME101 System VERSAdos Hardware and Software Configuration 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 Referrence 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
M68MASR/A8 8-Bit Cross Assembler on EXORmacs
M68XLINK/D1 8-Bit Cross Linkage Editor on EXORmacs User's Guide
M6809XPASC/D1 Cross Pascal Compiler On EXORmacs User's Manual

VME/10

M68KTENBG/D1 TENbug Debugging Package User's Manual
M68KVSDM/D1 VME/10 Microcomputer System Diagnostics Manual

VME/10 (cont.)

M68KVSGM/D1	VME/10 Microcomputer System Command and Graphics Primitives Reference Manual
M68KVSIG/D1	VME/10 Microcomputer System Installation Guide
M68KVSOM/D1	VME/10 Microcomputer System Overview Manual
M68KVSREF/D1	VME/10 Microcomputer System Reference Manual
68-M68KVSHD/V1, V2	VME/10 Hardware Documentation *(Optional)*

EXORmacs

M68KCHAS/D4	EXORmacs Chassis User's Manual
M68KDIOPP/D2	EXORmacs S/W Interface for Data I/O PROM Programmer User's Guide
M68KEMM/D3	EXORmacs Maintenance Manual
M68KMACS/D9	EXORmacs System Operations Manual
M68KMACSCM/D1	EXORmacs Development System Hardware/Software Configuration Manual
M68KMACSRK/D1	EXORmacs Chassis Rack Mounting
M68KRDS/D2	EXORmacs System Remote Development Station

EXORmacs (cont.)

M68KUSE/D2	EXORmacs User's System Emulator (Single-User)
M68KMACSBG/D1	EXORmacs System MACSbug Monitor Ref. Manual

HDS-400

M68HDS4EM1/D1	HDS-400 MC68000 Emulator Module User's Manual
M68HDS4EM2/D1	HDS-400 MC68008 Emulator Module User's Manual
M68HDS4EM3/D1	HDS-400 MC68010 Emulator Module User's Manual
M68HDS4FB1/D1	HDS-400 MC68000 Family Interface Module User's Manual
M68HDS4MM1/D1	HDS-400 Emulation Memory Module User's Manual
M68HDS4OM/D3	HDS-400 Hardware Development Station Operations Manual
M68HDS4UM/D3	HDS-400 Hardware Development Station User's Manual

BUS STATE ANALYZER

M68KBSA/D6	Real-Time Bus State Analyzer User's Manual

FIELD AND FACTORY SERVICE

Consistent with its responsibility as a leading producer of quality components for 16/32-Bit Microcomputer Systems, Motorola offers its customers both field and factory service of the highest quality. This service is provided in North America by Four Phase Systems, a unit of the Motorola Information Systems group. Four Phase Systems utilizes a support staff of more than 1500 to provide service at 162 locations nationwide. Four Phase offers on-site installation and maintenance, scheduled maintenance and factory exchange and repair services.

QUICK RESPONSE REPAIR/SPARE PARTS

For Microsystems products, a quick response nationwide to requests for service and for selected repair/spare parts is provided by the Four-Phase National Dispatch Center at the address given below. Telephone assistance can be obtained through the Four-Phase Technical Phone Support Center at (800) 528-1908 which has access to the extensive data bank of the Four-Phase Product Support Center. Quotations on repair/spare parts and service can be obtained from:

Four Phase Parts Sales
2200 Martin Avenue
Santa Clara, CA 95052
(800) 538-9660
(800) 662-9211 (California Only)

APPLICATIONS ENGINEERING

Applications engineering assistance is available through a network of over 50 Motorola Semiconductor Sales Offices in North America. In addition, the Four-Phase toll-free hotline provides a means of obtaining answers to questions on the use of Motorola-supported hardware and software products.

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
MTT6	MC6805 Introductory Course
MTT7	Understanding Microprocessor Basics MC68000 Minicourse on Cassette
MTT8	MC68000 16/32-bit Microprocessor Family
MTT10	8-bit Development Systems
MTT11	Basic MC6809
MTT12	Pascal
MTT13	EXORmacs VERSAdos Course
MTT15	RMS68K/VERSAdos Operating System
MTT16	Virtual System Course (MC68010)
MTT17	MCA Basic CAD Course
MTT18	SYSTEM V/68 (UNIX*)
MTT19	Designing with VERSAmodules/VMEmodules
MTT20	MC68020 Course
MTT21	To Be Announced
MTT22	To Be Announced
MTT23	VME/10 VERSAdos Course

COURSES AT YOUR OWN LOCATION:

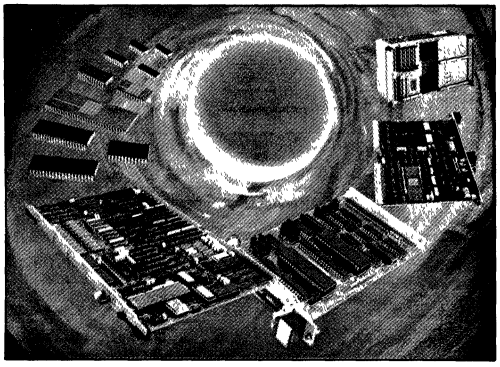
A special session of any Motorola Technical Training course may be held at your facility. This can be one of our standard courses or a course designed to fulfill your particular needs. For detailed information and scheduling, contact training headquarters.

MOTOROLA TECHNICAL TRAINING HEADQUARTERS

Telephone (602) 244-7126 or (602) 244-4945
Ray Doskocil, 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.
MC68174 VMEbus Arbiter (E-BAM)	The E-BAM performs round-robin and four level (VMEbus systems) and round-robin and five level (VERSAbus systems) priority arbitration.
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)	5.0 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	—	—	
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.
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	—
MVME200	VMEmodule 64K Byte Dynamic RAM	2.0 A/2.7 A	84 mA/500 mA	5 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	—	—	
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.
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	
MVME330	VMEmodule Ethernet Controller	3.8 A/4.6 A	600 mA/1.0 A	100 mA/200 mA	

8

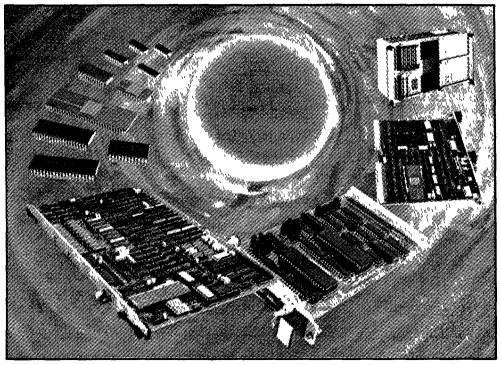
Table of Module Current Requirements (continued)

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 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 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	1A
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	1A

8



Index

9

Device No.	Description	Page No.
HDS-400	Hardware/Software Development Station	5-2
MVMCC3	VERSAmodule 8-Slot Card Cage	2-3
MVMCH3-1	VERSAmodule 8-Slot Chassis	2-3
MVMCH3-2	VERSAmodule 8-Slot Chassis	2-3
MVMEBUG	VMEbug Debugging Package for MVME110-1	1-2
MVMEBUG1	VMEbug Source Object Modules on VERSAdos Diskette	1-2
MVMEBUG2	VMEbug Source Object Modules on VERSAdos Cartridge	1-2
MVME025	System Controller VMEmodule	1-6
MVME101	VMEmodule Monoboard Microcomputer	1-8
MVME101BUG	Debugging Package for MVME101 MPU Module	1-17
MVME101BUGLF	101bug Source Object Module on VERSAdos Diskette	1-17
MVME101BUGLC	101bug Source Object Module on VERSAdos Cartridge	1-17
MVME110-1	VMEmodule Monoboard Microcomputer	1-20
MVME115M	VMEmodule Monoboard Microcomputer	1-27
MVME200	64K Byte Dynamic RAM Module	1-31
MVME201	256K Byte Dynamic RAM Module	1-31
MVME202	Dynamic RAM Modules	
MVME222-1	See MVME202	1-37
MVME222-2	See MVME202	1-37
MVME211	Static RAM/ROM Module	1-41
MVME300	IEEE-488 Listener/Talker/ Controller Module	1-44
MVME310	Intelligent Peripheral Controller Module	1-52
MVME315	Intelligent Peripheral Interface Module	1-57
MVME320	Winchester/Floppy Disk Controller	1-63
MVME330	Ethernet Node Processor/LAN Controller Module	1-67
MVME400	Dual Port RS-232C Communications Module	3-2
MVME410	Dual 16-Bit Parallel I/O Module	3-6
MVME420	SASI Peripheral Adapter Module	3-10
MVME435	Buffered 9-Track 1/2" Magnetic Tape Adapter	3-15
MVME600	12 bit A/D, I/O Channel Module	3-23
MVME601	Expander Module for MVME600	3-23
MVME605	12-Bit D/A I/O Channel Module	3-35

Device No.	Description	Page No.
MVME610	A/C Input I/O Channel Module	3-41
MVME615	A/C Output I/O Channel Module (zero x-over)	3-47
MVME616	A/C Output I/O Channel Module (Phase Angle)	3-47
MVME620	D/C Input I/O Channel Module	3-51
MVME625	D/C Output I/O Channel Module	3-57
MVME910-3	VMEmodule 200 Watt Switching Power Supply	1-77
MVME920	20-Slot VMEbus Backplane	1-77
MVME921	9-Slot VMEbus Backplane	1-77
MVME922	5-Slot I/O Channel Backplane	1-77
MVME930	VMEbus Extender Module, Double High	1-77
MVME931-1	VMEbus Wirewrap Module, Double High	1-77
MVME932	I/O Channel Extender Module	
MVME933-1	I/O Channel Wirewrap Module	
MVME935	Remote I/O Connector Board	
MVME940-1	9-Slot VMEbus, 10 Slot I/O Channel Chassis with 200 Watt Switching Power Supply	1-77
MVME941	9-Slot VMEbus, 10-Slot I/O Channel Card Cage	1-77
MVME942	20-Slot VMEbus Card Cage	1-77
M68BSAC	Bus State Analyzer Control Module	5-27
M68BSACE	Bus State Analyzer Control Module with Enclosure	5-27
M68BSA1-1	MC68000/68010/68451 Personality Module	5-31
M68BSA5	Real-Time VERSAbus State Analyzer Personality Module	5-37
M68KEXTM	VERSAbus Extender Module	2-9
M68KMACS	MC68000 EXORmacs Development System	5-42
M68KVAM	VERSAbus Adapter Module	2-10
M68KVBUG	VERSAbug Debugging Package for M68KVM01	2-14
M68KVBUGLC	VERSAbug Source/Object on Cartridge	2-14
M68KVBUGLF	VERSAbug Source/Object on Diskette	2-14
M68KVBUG2	VERSAbug Debugging Package for M68KVM02	2-14
M68KVBUG2LC	VERSAbug Source & Object on VERSAdos Cartridge	2-14
M68KVBUG2LF	VERSAbug Source & Object on VERSAdos Diskette	2-14
M68KVBUG2LMC	VERSAbug Source & Object on VERSAdos Lark Cartridge	2-14
M68KVBUG3	VERSAbug Debugging Package for M68KVM03	2-20

MOTOROLA *Semiconductor Products Inc.*

BOX 20912 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.