Motorola MVME130

Single Board Computer

In Stock

Used and in Excellent Condition

Open Web Page

https://www.artisantg.com/71423-1

ARTISAN TECHNOLOGY GROUP

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

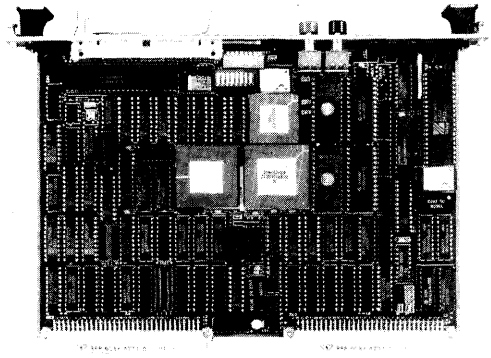
Artisan Technology Group

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

- Critical and expedited services
- In stock / Ready-to-ship
- We buy your excess, underutilized, and idle equipment
- Full-service, independent repair center

MVME130
MVME131

VMEmodule 32-Bit Monoboard Microcomputer



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

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

CPU

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

DEBUG MONITOR FIRMWARE

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

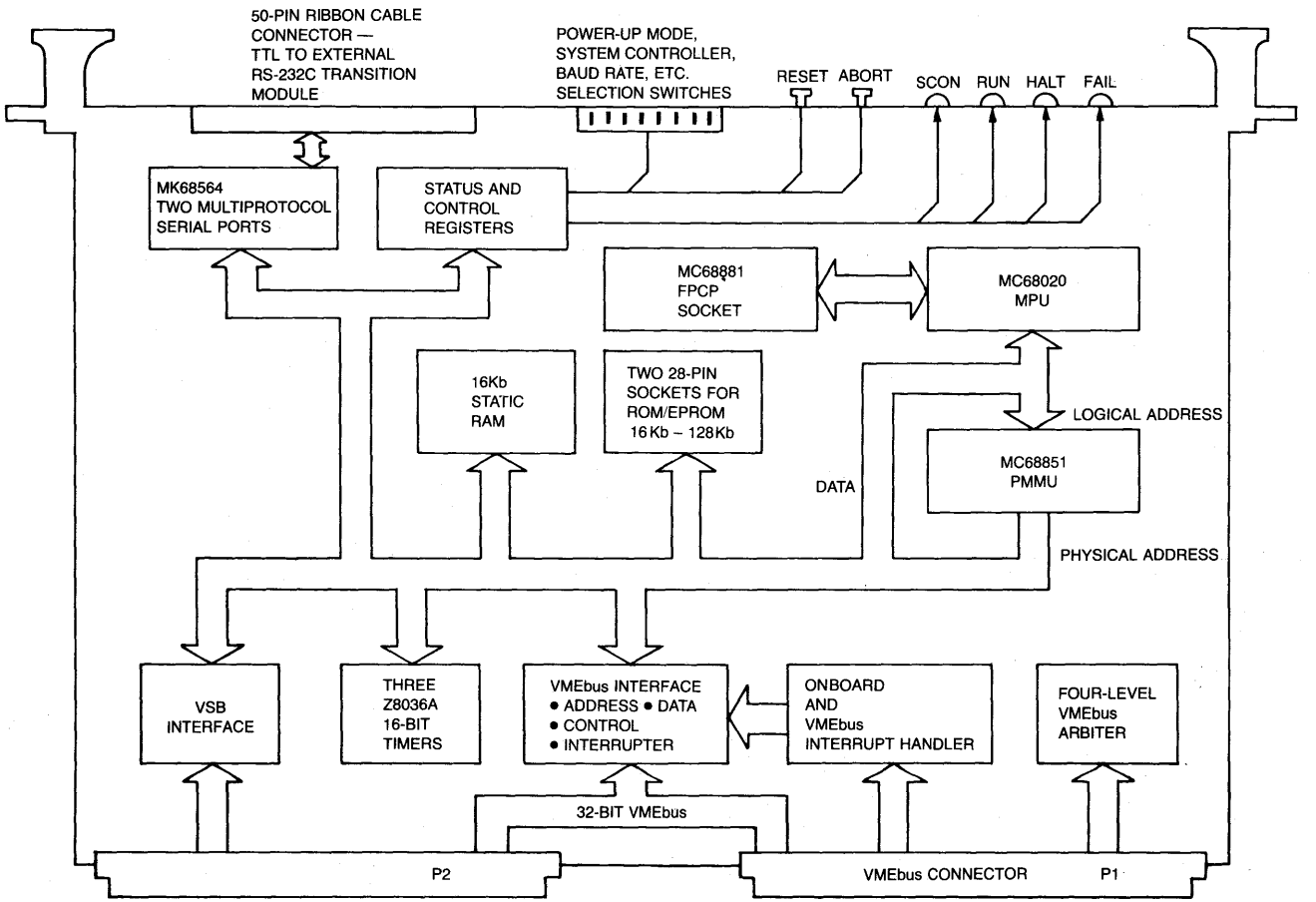
COPROCESSOR

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

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

Figure 1 is a block diagram of the MVME130 module.

FIGURE 1 — MVME130 32-Bit Microcomputer Module



MEMORY MANAGEMENT

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

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

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

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

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

ONBOARD STATIC RAM

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

VSB INTERFACE

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

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

MEMORY MAP

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

TABLE 1 — Address Map

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

PROGRAMMABLE TIMER MODULE

The MVME130 Programmer Timer Module provides the following:

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

SERIAL I/O

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

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

STATUS/CONTROL REGISTERS

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

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

TABLE 2 — Status Register Format

<STAT0>

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

<STAT1>

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

Provides VME130 Networking Base Address
and useful as general purpose status

<STAT2>

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

BERR Source Status Register

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

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

TABLE 3 — Control Register Format

<CNT0>

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

<CNT1>

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

<CNT2>

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

<CNT3>

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

<CNT4>

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

<CNT5>

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

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

VMEbus INTERFACE

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

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

INTERRUPT HANDLER

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

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

The Interrupt assignments are described in Table 4.

VMEbus INTERRUPTER

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

TABLE 4 — Interrupt Handler Priority Assignments

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

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

Group 3 Interrupts

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

Group 2 Interrupts

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

Group 1 Interrupts

IRQ7-1*	<VMEbus Interrupts>

Mechanical and Environmental Specifications

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

Ordering Information

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

Related Product

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

Advance Information

130bug Debug/ Diagnostic Monitor

130bug RESIDENT PACKAGE

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

130bug SOURCE MODULE PACKAGE

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

130bug DEBUG/DIAGNOSTIC MONITOR

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

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

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

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

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

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

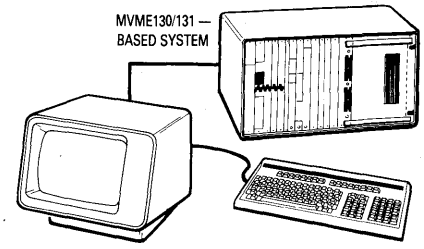


Figure 1. Standalone Operation

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

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

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

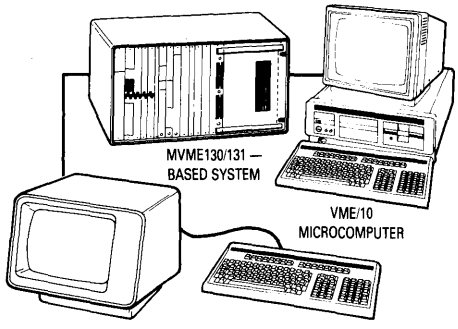


Figure 2. Transparent Mode — Direct Serial Connection With Host Computer

DEBUG/DIAGNOSTIC FACILITIES

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

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

PROGRAM DEBUGGING

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

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

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

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

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

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

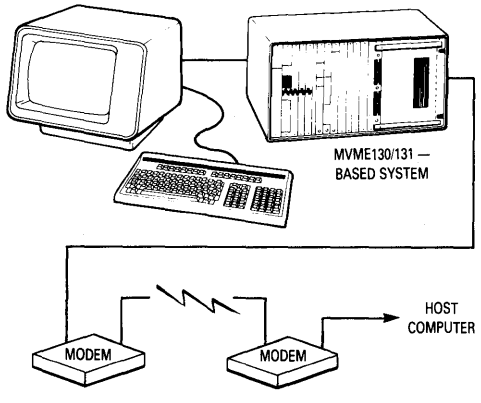


Figure 3. Transparent Mode — Dial-Up Serial Link With Host Computer

Table 1. 130bug Debugging Commands

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

HARDWARE DIAGNOSTICS

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

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

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

CALLABLE 130bug ROUTINES

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

Table 2. 130bug Diagnostic Commands

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

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

Table 3. Callable 130bug Functions

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

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

ORDERING INFORMATION

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