VMEbus Compatible

MVCP 16-Line Sync/Async Communications Processor

Product P/N 221025

Manual P/N 341025 Revision B

Revision Date: March 30, 1999

Includes:
Installation Instructions
Programming Specification

Macrolink, Inc. 1500 North Kellogg Drive Anaheim, California 92807
Phone (714)777-8800 — FAX (714)777-8807
info@macrolink.com – http://www.macrolink.com

Table of Contents

Chapter 1 - Installation

Chapter 2 - User Applications

Appendix A - Technical Information

Installation

1.1 Introduction

The Macrolink MVCP VME Multi-Protocol Communications Processor is a sophisticated, high-speed, microprocessor based controller configured with 8 or 16 independent synchronous/asynchronous serial communications ports. Each port supports serial data rates up to 2Mbps, with an aggregate throughput ability of 15Mbps for all 16 ports simultaneously. Status, data movement & detection, port arbitration & selection, and protocols are processed on-board, minimizing driver calls and host intervention while maximizing overall system throughput.

The MVCP allows for user applications and custom interface protocols to be downloaded into its on-board SRAM, or optionally, the user application may be stored in the MVCP's non-volatile Flash EPROM. The MVCP handles all character processing, flow control, status decoding & interpretation, and error handling, offloading this responsibility from the host.

The MVCP uses field-replaceable Single Inline Module (SIM)-style serial line conditioners that allow easy reconfiguration. Up to 4 different line disciplines can be supported on each controller. Available interfaces include RS-232, RS-422, RS-485, buffered TTL, and MIL-STD-188. Base memory is 512KB of dual-ported, 12ns SRAM which can be expanded to 4MB. Serial port connections are made via high-density front panel connectors, or optionally, the backplane P2 connector.

Streams-based drivers are available for many popular operating systems. A comprehensive VxWorks-based software development kit is also available which includes I/O & memory maps, code samples, application notes, and Enhanced Serial Communications Controller (ESCC) data books. Program source is provided in C. Contact Macrolink for more information about these software options.

1.2 Unpacking And Inspection

Each MVCP is shipped in a sturdy, reusable padded carton. Optional accessories will be shipped in a separate carton and include I/O interface panels, interface cables, and an Installation and Programming manual. The shipping carton and packaging material should be retained for future use in moving, shipping, or storage of the board and accessories.

Carefully inspect the shipping carton for signs of damage. If any shipping damage is evident, do not open the package. Notify Macrolink and the freight carrier immediately to receive further instructions. Open the shipping carton and remove the anti-static bag containing the MVCP.

☞ The MVCP uses sensitive electronic circuitry that can easily be damaged by electrostatic discharge. Personnel handling the board must exercise proper static control methods. Open the shipping carton only at an approved static-controlled workstation. An anti-static bag, anti-static bin, or the original packaging material must be used when transporting the board.

Carefully remove the MVCP from the antistatic shielding bag. Inspect the board for damage prior to installation. Claims for shortage or damage must be filed within seven days of receipt of the shipment.

1.3 Board Configuration

Address Switches

The base address of the MVCP is selected via rotary switches SW 1 and SW 2. SW 1 is the most significant nibble of the board's address. Refer to Figure 1-1 to locate these switches. The MVCP must be configured to match the addressing mode used by your system. Most systems support A32 addressing, while some support only A24 or A16 addressing. Install only the addressing mode jumper that corresponds with the addressing mode used by your system. Refer to the table on the next page for address mode jumper information.

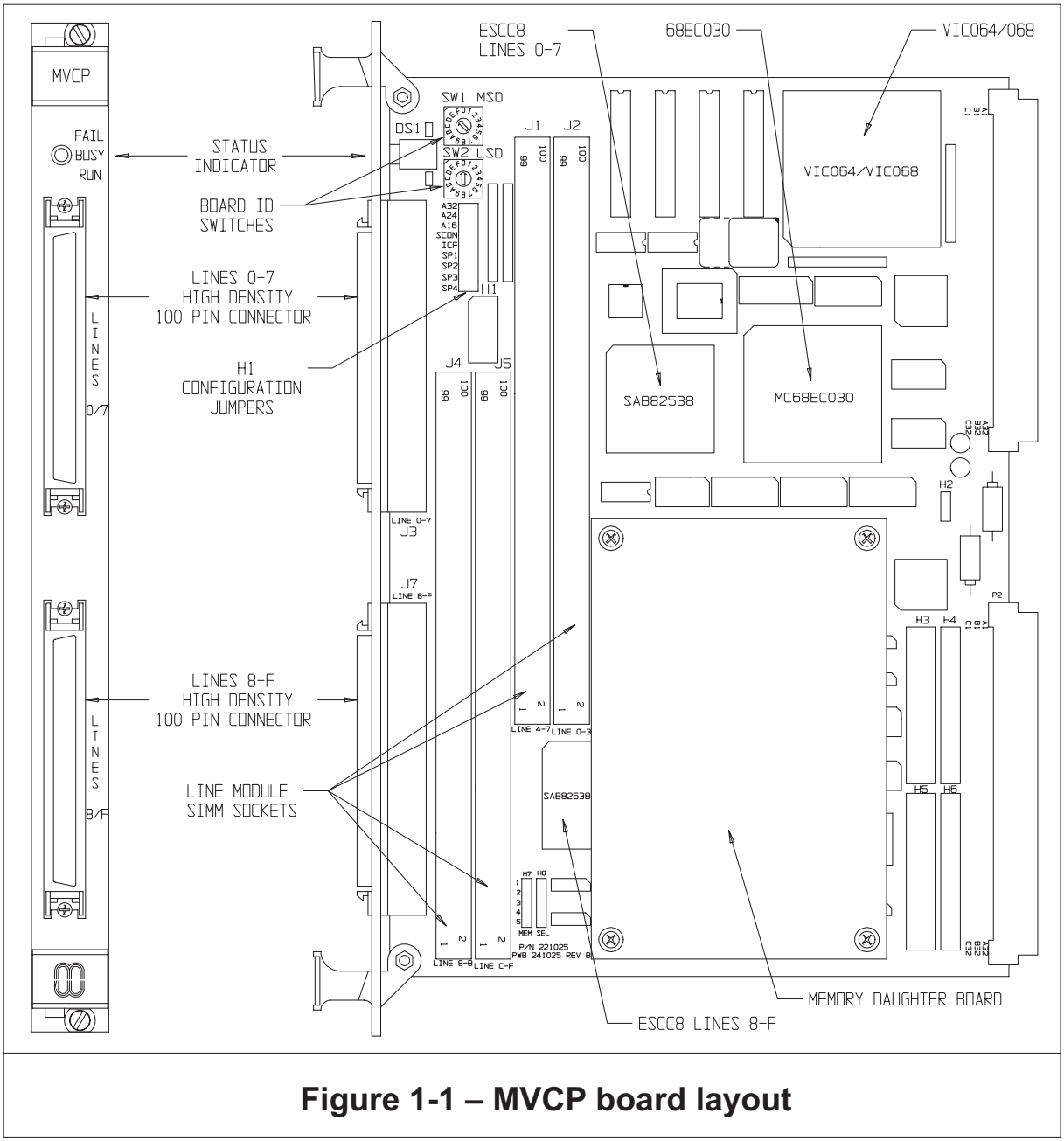


Figure 1-1 – MVCP board layout

MVCP Jumpers

The MVCP has nine user configuration jumpers, plus a bank of jumper blocks (H3-H6) to allow serial I/O through the P2 connector. The remaining jumpers are set at the factory and must not be changed. Refer to figure 1-1 to locate the configuration jumpers.

H1 Option Jumpers

Pins	Function
1 - 2	This jumper selects A32 addressing. The default setting is installed.
3 - 4	This jumper selects A24 addressing. The default setting is removed.
5 - 6	This jumper selects A16 addressing. The default setting is removed.
7 - 8	Enables System Controller mode. The default setting is removed.
9-10	Enables the VIC interprocessor communication facilities. The user must store an ICF compatible user application in the Flash EPROM before using this feature. If installed, the A32, A24, and A16 CONFIG jumpers must be removed. The default setting is removed.
11-12	When installed, this jumper enables the Power On Self Test loop mode. In loop mode the POST will continuously run. On each pass the LED will change color between yellow and green. If an error is detected, the test will halt with the LED set to red. This jumper MUST be removed for normal board operation. The default setting is removed.
13-14	Enables auto-execution of the user application stored in flash EPROM after the Power On Self Test completes. The default setting is removed.
15-16	Enables continuation after a POST failure. This may allow the boot loader or user application to start even if the board failed part of its Power On Self Test. Proper board operation is not guaranteed. Boot loader commands may be issued even though the command register will contain the POST failure code. The default setting is removed.
17-18	Reserved for user application. The default setting is removed.

System Controller Mode

The MVCP can be configured to be the VMEbus System Controller by installing the SCON jumper. The System Controller is the VMEbus system clock generator, the bus request arbiter, the IACK daisy-chain driver, and the bus timer. The System Controller must be installed in the first slot of the backplane. Normally this function is performed by the CPU card. The SCON jumper is not installed when the board is shipped from the factory.

I/O through P2 Connector

An adapter panel is available that allows the MVCP serial I/O signals to be routed through the VME P2 connector. The P2 I/O adapter is equipped with two 100-pin high-density connectors similar to those on the front panel. This allows the interface cables to be kept in the rear of the chassis, and is an ideal solution where front panel space is at premium. The P2 I/O adapter is available with either straight or right-angle connectors. The serial line conditioners are removed from the MVCP and installed in sockets on the P2 I/O adapter. Complete installation documentation is included with the P2 I/O adapter. Contact Macrolink for more information.

1.4 Replaceable Serial Line Conditioners

The MVCP uses field-replaceable 100-pin Single Inline Module (SIM)-style serial line conditioners that allow easy reconfiguration or repair. The most popular serial line conditioners are for the RS-232, RS-422, and RS-485 interfaces, and are documented in the figures and tables below and on the next two pages. Other serial line conditioners are available; contact Macrolink for more information.

To remove a serial line conditioner, release the locking clips at both ends of the socket and tilt the serial line conditioner towards the front panel of the MVCP. Lift the serial line conditioner from the socket. To install a serial line conditioner, gently insert it into the socket, with the top of the serial line conditioner angled about 30 degrees towards the front panel. The pin 1 end of the serial line conditioner is oriented towards the configuration jumper edge of the board. The serial line conditioners are keyed to prevent improper insertion. Tilt the serial line conditioner to the vertical position until the locking clips snap into place.

RS232 Serial Line Conditioner

The RS232 serial line conditioners are equipped with DIP switches that are used to select the direction of the synchronous transmit and receive clocks on a port by port basis. Each clock is controlled by two switches. Setting both switches to OFF disables that port's clock input and output.

RS232 Serial Line Conditioner Configuration

Signal	Direction	Switch Setting
RXCLK 0, 4, 8, C	Send	SW2-1 On, SW2-2 Off
	Receive	SW2-1 Off, SW2-2 On
TXCLK 0, 4, 8, C	Send	SW2-3 On, SW2-4 Off
	Receive	SW2-3 Off, SW2-4 On
RXCLK 1, 5, 9, D	Send	SW2-5 On, SW2-6 Off
	Receive	SW2-5 Off, SW2-6 On
TXCLK 1, 5, 9, D	Send	SW2-7 On, SW2-8 Off
	Receive	SW2-7 Off, SW2-8 On
RXCLK 2, 6, A, E	Send	SW1-1 On, SW1-2 Off
	Receive	SW1-1 Off, SW1-2 On
TXCLK 2, 6, A, E	Send	SW1-3 On, SW1-4 Off
	Receive	SW1-3 Off, SW1-4 On
RXCLK 3, 7, B, F	Send	SW1-5 On, SW1-6 Off
	Receive	SW1-5 Off, SW1-6 On
TXCLK 3, 7, B, F	Send	SW1-7 On, SW1-8 Off
	Receive	SW1-7 Off, SW1-8 On

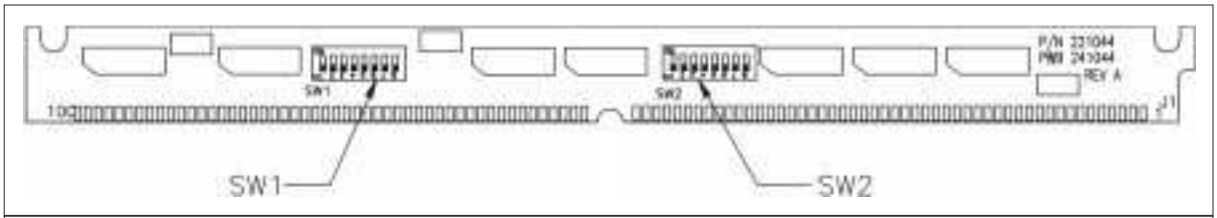


Figure 1 - 2 RS232 Serial Line Conditioner Switch Locations

RS485 Serial Line Conditioner

The RS485 serial line conditioners have terminators for the clocks and data lines that are enabled via DIP switches. Four switches are used to terminate the clock and data signals for each port. Set the switches to the ON position to enable termination on the desired lines.

RS485 Serial Line Conditioner Signal Termination Switches

Signal	Switch	Description
TXDATA 0,4,8,C	SW1-1	Transmit Data
RXDATA 0,4,8,C	SW1-2	Receive Data
TXCLK 0,4,8,C	SW2-1	Transmit Clock
RXCLK 0,4,8,C	SW2-2	Receive Clock
TXDATA 1,5,9,D	SW1-3	Transmit Data
RXDATA 1,5,9,D	SW1-4	Receive Data
TXCLK 1,5,9,D	SW2-3	Transmit Clock
RXCLK 1,5,9,D	SW2-4	Receive Clock
TXDATA 2,6,A,E	SW1-5	Transmit Data
RXDATA 2,6,A,E	SW1-6	Receive Data
TXCLK 2,6,A,E	SW2-5	Transmit Clock
RXCLK 2,6,A,E	SW2-6	Receive Clock
TXDATA 3,7,B,F	SW1-7	Transmit Data
RXDATA 3,7,B,F	SW1-8	Receive Data
TXCLK 3,7,B,F	SW2-7	Transmit Clock
RXCLK 3,7,B,F	SW2-8	Receive Clock

Another set of switches enables TXCLK to be gated on and off by the RTS signal. Set the switch to the ON position to enable TXCLK control on a given port.

Enable TXCLK Control By RTS

Switch	Port
SW3-1	0, 4, 8, C
SW3-2	1, 5, 9, D
SW3-3	2, 6, A, E
SW3-4	3, 7, B, F

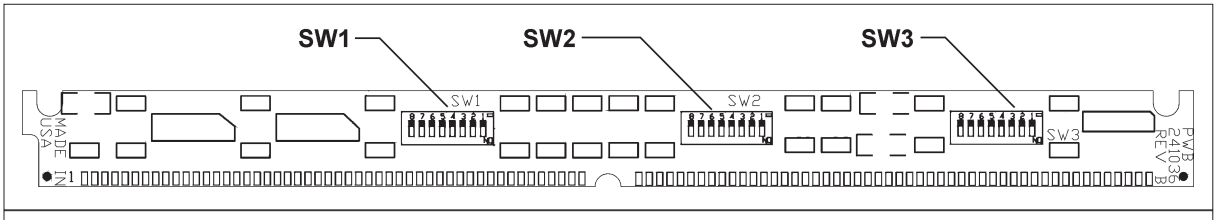


Figure 1 - 3 RS485 Serial Line Conditioner Switch Locations

RS422 Serial Line Conditioner

Each clock is controlled by three switches. Setting all switches to OFF disables that port's clock input and output.

RS422 Serial Line Conditioner Configuration		
Signal	**Direction**	**Switch Setting**
RXCLK 0, 4, 8, C	Send	SW3-1 & 2 On, SW2-1 Off
	Receive	SW3-1 & 2 Off, SW2-1 On
TXCLK 0, 4, 8, C	Send	SW3-3 & 4 On, SW2-2 Off
	Receive	SW3-3 & 4 Off, SW2-2 On
RXCLK 1, 5, 9, D	Send	SW3-5 & 6 On, SW2-3 Off
	Receive	SW3-5 & 6 Off, SW2-3 On
TXCLK 1, 5, 9, D	Send	SW3-7 & 8 On, SW2-4 Off
	Receive	SW3-7 & 8 Off, SW2-4 On
RXCLK 2, 6, A, E	Send	SW2-5 & 6 On, SW1-5 Off
	Receive	SW2-5 & 6 Off, SW1-5 On
TXCLK 2, 6, A, E	Send	SW2-7 & 8 On, SW1-6 Off
	Receive	SW2-7 & 8 Off, SW1-6 On
RXCLK 3, 7, B, F	Send	SW1-1 & 2 On, SW1-7 Off
	Receive	SW1-1 & 2 Off, SW1-7 On
TXCLK 3, 7, B, F	Send	SW1-3 & 4 On, SW1-8 Off
	Receive	SW1-3 & 4 Off, SW1-8 On

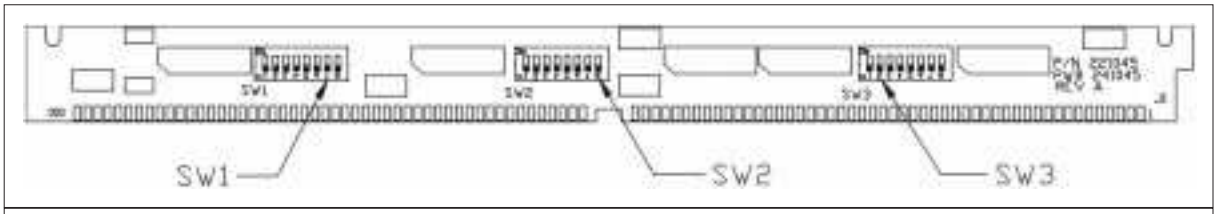


Figure 1 - 4 RS422 Serial Line Conditioner Switch Locations

1.5 Installing The MVCP

The location of the MVCP on the DMA priority chain may be critical to the overall performance of your system. DMA priorities must be balanced to achieve maximum system performance. If the MVCP is not placed high enough on the chain, it may not be able to get enough of the available bus bandwidth to maintain a high data throughput. Likewise, if the board is too high on the chain, it may starve other boards that have smaller buffers than it.

The MVCP must be installed in a backplane that has both P1 and P2 connectors. If your backplane has jumpers or switches on the VME serial lines, remove or open the bus request (BR0 - BR3 IN/OUT) and the interrupt acknowledge (IACK IN/OUT) jumpers or switches for the slot in which the MVCP is to be installed. Slide the board into the slot, ensuring that it is fully seated into the backplane connectors. Tighten the board lock down screws to secure the board and guarantee proper RF shielding.

1.6 Cabling

The MVCP uses high-density 100 pin connectors for the I/O ports. The upper connector provides the interface for ports 0 through 7. The lower connector provides the interface for ports 8 through F. See Figure 1-1 for the location of these connectors. If the MVCP is configured as an 8 port board, only the upper connector is used.

☞ The pins used on the connectors are delicate. Be sure the connectors are properly aligned when installing the cables to avoid bending or breaking the pins. NEVER FORCE THE CONNECTORS TOGETHER.

The standard interface cable sold for use with the MVCP is 6 feet long. Cables in other lengths are available; contact Macrolink for ordering information. Macrolink supplied interconnect cables should be used to avoid excessive RF radiation in violation of FCC limits.

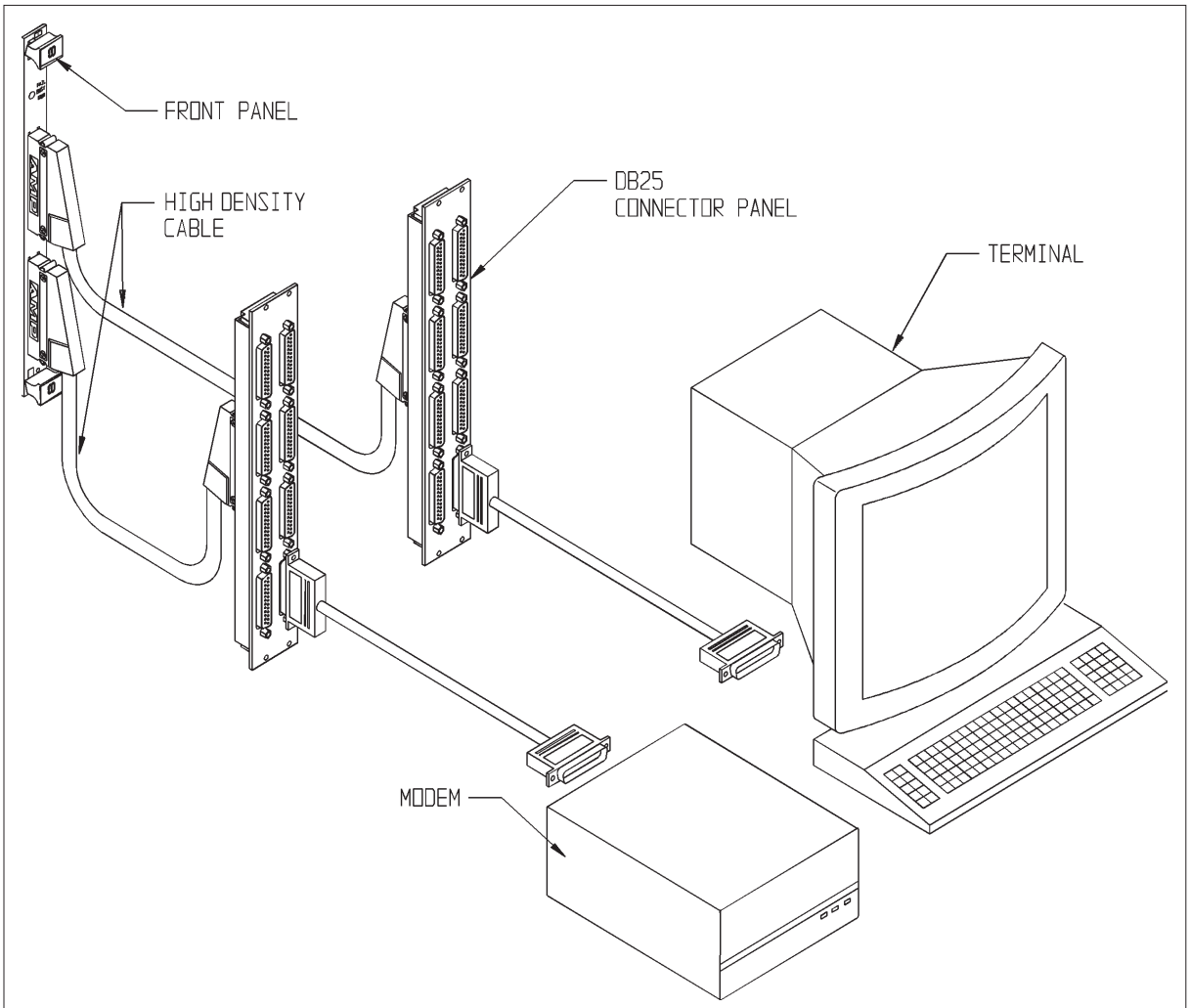


Figure 1 - 5 DB25 I/O Panel Connection

Figure 1-5 illustrates connection of two DB25 I/O panels to the MVCP. Figure 1-6 illustrates connection of an RJ45 I/O panel to the MVCP. The I/O panels may be installed in free space in either the front or the back of your CPU chassis. Contact Macrolink for cable and I/O panel ordering information.

1.7 Front Panel Status Indicator

The MVCP uses a tri-color LED for status indication. The status indicated by this LED varies according to the mode that the board is in. The LED is under the control of the user application while it is running. The application programmer must define their own status indications in this mode. Refer to Figure 1-1 to locate the board status indicator.

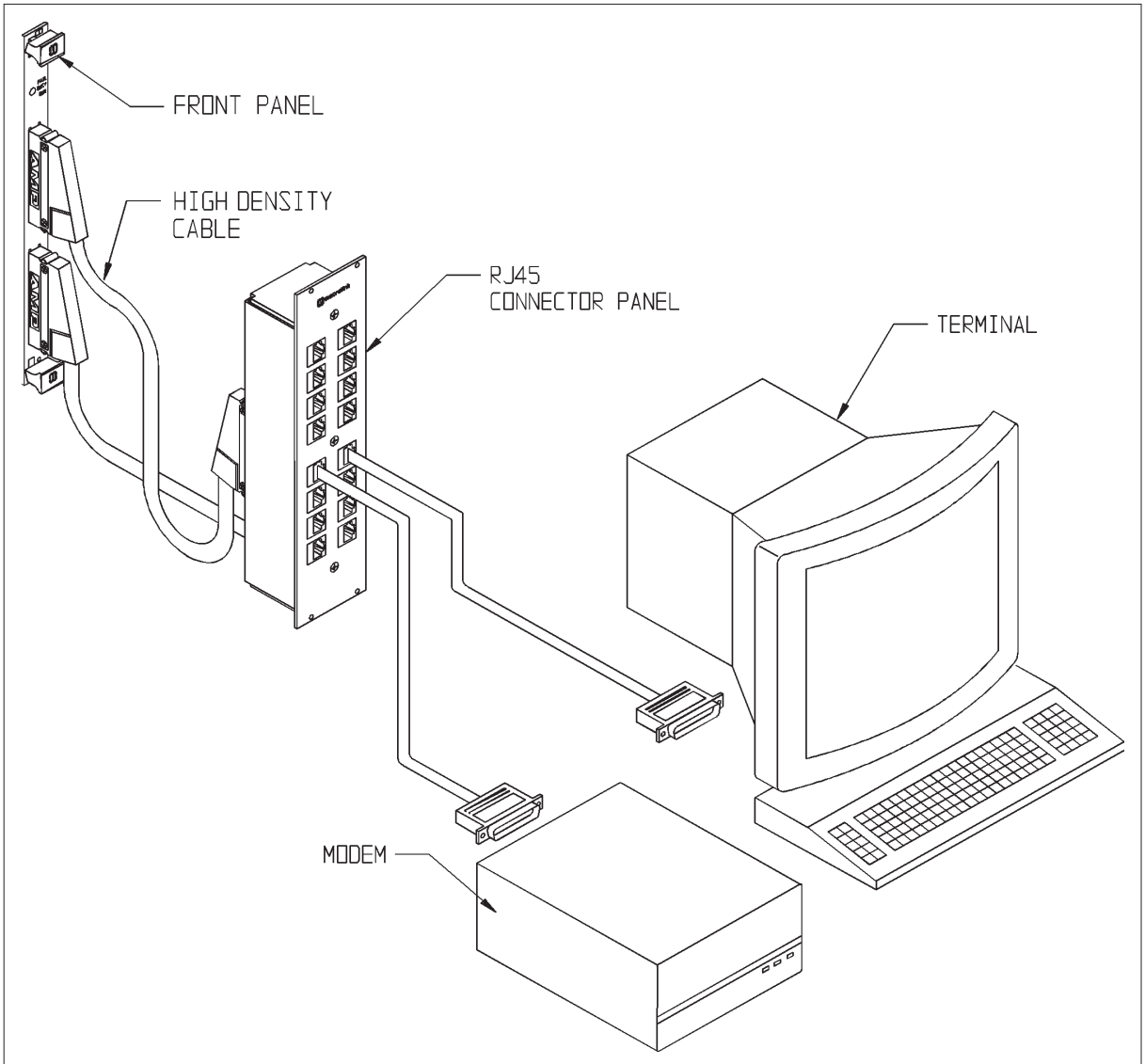


Figure 1 - 6 RJ45 I/O Panel Connection

1.8 Software Development Kit

The MVCP Software Development Kit (SDK) is designed to assist users in creating their own application firmware for the MVCP by providing a working example of a driver and firmware interface. The application, driver, or demo may be used as a starting point for development of a specific application or a reference for implementing features on the MVCP.

The MVCP SDK contains source and executables for the following:

- An MVCP executable application.
- A VxWorks driver (Motorola-based host) for the MVCP.
- A VxWorks application demo which utilizes the firmware application and driver.
- Support files for these programs.

Installing the SDK

The SDK is in a tar archive named mvcpsdkB.tar.Z. This file is compressed using the Unix "compress" utility. After copying the file to your system from the distribution media, use the uncompress utility to extract the archive. Please note that the uncompressed file will still be an archive in tar format, so tar must be run to extract the files within it.

READMEs are included under each subdirectory which describe the contents of that directory. The VxWorks demo and driver READMEs contain information on loading or installing them into the VxWorks RTOS. These should be reviewed before running the SDK or modifying any source code contained within it.

Bootcode Electronic Users Manual

The manuals.zip file included with the distribution contains the User's Manual for use with the MVCP Bootcode Firmware in plain text and HTML format. If using the HTML manual, load the man_html/index.htm file into your HTML browser and follow the links of interest.

The bootcode firmware resides on the MVCP and is executed after the board's Power On Self Test (POST) has completed. A complete description of the bootcode firmware implementation and its interface appears in Chapter 2.

Special Notes for DOS-based File System Users

If the files extracted from the archived SDK file will reside on a DOS-based file system, please be aware the following file names will be modified from their original name:

File Names Under DOS

Original	DOS
vxapp/Makefile.MC68040gnu	vxapp/Makefile.MC6
vxdrv/Makefile.MC68040gnu	vxdrv/Makefile.MC6
vxdrv/srecmake/Makefile.MC68040gnu	vxdrv/srecmake/Makefile.MC6
vxdrv/h/mvcpLibErr.h	vxdrv/h/mvcpLibE.h

1.9 Returning Products For Repair

Before returning a product for repair, you must obtain a Return Material Authorization (RMA) number from Macrolink. This number must appear on the outside of the shipping container. Products returned without an RMA number will be refused. When requesting an RMA number, please be prepared to provide the following information:

- Product name or part number.
- Serial number.
- Failure description. An inspection fee may be assessed on items returned without an adequate failure description.

Shipping expenses to Macrolink are to be paid by the customer. Macrolink will pay for standard return shipping for products under warranty.

1.10 FCC Compliance

This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with this manual, may cause interference to radio and television reception. This equipment has been tested and found to comply with the limits for a Class A computing device in accordance with the specifications in Subpart J of Part 15 of the FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference requiring the user, at the user's own expense, to take whatever measures may be necessary to correct the interference.

If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference in one or more of the following ways:

1. Reorient the receiving antenna.
2. Relocate the computer with respect to the receiver.
3. Move the computer away from the receiver.
4. Plug the computer into a different outlet, so that the computer and receiver are on different circuit branches.
5. Ensure that the mounting screws, attachment connector screws, and grounding connections are securely tightened.
6. Ensure that good quality, shielded and grounded cables are used for data communications.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the booklet *How To Identify And Resolve Radio-TV Interference Problems* prepared by the Federal Communications Commission helpful. This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20420, Stock No. 004-000-0345-4.

User Applications

This chapter of the manual is provided to assist in the creation of a user application to run on the MVCP. It provides a description of the MVCP register interface, but is not meant to be an all-encompassing guide. A level of familiarity with 'C' and/or assembly language programming is required, as well as familiarity with the serial protocol(s) you wish to implement. Reference works on hand should include data books for the Motorola 68EC030 CPU and Siemens SAB8532 ESCC 8. The Siemens data is available on the Macrolink web page at http://www.macrolink.com.

A Software Development Kit package is available that runs in the VxWorks environment. The MVCP SDK is designed to assist users in creating their own application firmware to be executed on the MVCP by providing a working example of an MVCP driver and firmware interface. The SDK contains source and executables for an MVCP-based application, a VxWorks (Motorola based host) MVCP driver, an application demo which utilize the firmware application and driver, support files for these programs, and application notes. Program source, with annotation, is provided in C. The application, driver or demo may be used as a starting point for development of a specific application or a reference for implementing certain features on the MVCP. For more information, refer to Chapter 3 - *Software Development Kit*.

2.1 MVCP Operation

The MVCP transitions through three different modes after it is powered up or reset. The modes are Power On Self Test (POST), Boot Load, and Application. The POST runs automatically. When it completes, the board will switch to Boot Load mode. If the POST fails, the board status indicator will be set to red. Registers on the board can be read to help determine the cause of the failure.

In Boot Load mode, the user application is downloaded to the board. Extended diagnostic tests can be invoked to verify operation of the board beyond the level tested by the POST. If the user application has been stored in the MVCP's Flash EPROM, Boot Load mode may be bypassed by installing configuration jumper H1 pins 13&14 (Auto-execute).

2.2 Power On Self Test

The Power On Self Test (POST) executes every time the board is reset or powered up. The board status indicator is yellow while the POST is running. The POST takes up to four seconds to complete, depending upon the amount of RAM installed. Upon successful completion of the POST, the board status indicator LED is set to green and the board switches to Boot Load mode.

In the event of a POST failure, the status LED will be red, and the Command register will contain the value 0xBADTTTSS (see Section 2.5 -*Boot Load Mode Memory Map*). TTT specifies the POST test number and SS specifies the POST sub-test number that failed. The test numbers are identified in the table on the next page. If the POST fails, do not issue any boot loader commands except Software Reset to the MVCP. Additional failure information may be present in the ASCII Print buffer.

The POST may be run continuously by installing jumper H1 pins 11&12. Extended diagnostics are also available by use of the boot loader diagnostic test commands. See Section 2.6 – *Boot Load Commands*.

POST Test Numbers	
Test	**Description**
Test 0x001	RAM data bus test.
Test 0x002	RAM data equals address test.
Test 0x003	RAM data equals complement of address test.
Test 0x004	RAM byte lane test.
Test 0x005	RAM word lane test.
Test 0x006	RAM longword lane test.
Test 0x007	Firmware checksum test.
Test 0x008	RAM size test.
Test 0x009	Extended RAM data equals address test.
Test 0x010	Extended RAM data equals complement of address test.
Test 0x011	VIC data bus test.
Test 0x012	VIC address bus test.
Test 0x013	VIC generated interrupts to CPU test.
Test 0x014	Timer data bus test.
Test 0x015	Timer address bus test.
Test 0x016	Timer 0 connected to Timer 1 test.
Test 0x017	Timer generated interrupts test.
Test 0x018	Configuration test.
Test 0x019	ESCC 0 data bus test.
Test 0x020	ESCC 0 byte lane test.
Test 0x021	ESCC 0 address bus test.
Test 0x022	ESCC 0 generated interrupt test.
Test 0x023	ESCC 0 receive clock source test.
Test 0x024	ESCC 0 internal loop back test.
Test 0x025	ESCC 1 data bus test.
Test 0x026	ESCC 1 byte lane test.
Test 0x027	ESCC 1 address bus test.
Test 0x028	ESCC 1 generated interrupt test.
Test 0x029	ESCC 1 receive clock source test.
Test 0x030	ESCC 1 internal loop back test.
Test 0x031	External loop back test.
Test 0x032	Command interrupt test.
Test 0x033	VME bus master DMA test.

2.3 Boot Load Mode

After the MVCP has successfully completed its POST, it enters Boot Load mode. The board is ready to accept commands to download the user application to its on-board RAM, or optionally, to the non-volatile Flash EPROM. Also, extended diagnostic tests can be invoked by issuing the proper commands.

If configuration jumper H1 pins 13&14 (Auto-execute) is installed, the MVCP will bypass Boot Load mode and automatically start the user application stored in the Flash EPROM.

2.4 Board Status LED in Boot Load Mode

The Board Status LED is used to indicate the current state of the MVCP. The states indicated by the status LED during Boot Load mode are as follows:

Boot Load Mode LED Codes

Color	Meaning
Green	Ready for boot loader command
Yellow	Processing boot loader command
Red	Boot loader command failed
Off	Started user application

The LED will be turned off before starting the user application. It is recommended that one of the first user application instructions set the LED to yellow or green as a visual cue that the application has started.

2.5 Boot Load Mode Memory Map

During Boot Load mode, several registers are used to exchange information between the MVCP and the host system. The registers are defined in the table on the next page. The address of a register is its offset added to the MVCP's VMEbus base address. For example, if the board's base address is 0xC2000000, the Command register is at address 0xC2000000 and the ASCII Print Buffer starts at 0xC2000400.

The Command register is used for both issuing boot loader commands and receiving command completion status. This register may only be accessed as a longword. Commands may use some, none, or all of the Parameter registers.

Writing the Command Interrupt register as a byte with the value of 3 causes an active low strobe on the VIC LIRQ1 input pin to be generated. The user application can program the VIC Local Interrupt Control register to use this strobe to generate a CPU interrupt. This interrupt can be used for any user defined purpose, such as starting a command.

The ASCII Print Buffer contains text that may assist in diagnosing problems. The buffer is filled primarily with configuration and diagnostic information.

The user application may be downloaded anywhere into the User Application buffer. The Auto-execute feature (jumper H1 13&14 installed) will begin execution of user application code stored in the Flash EPROM.

Do not write to any Reserved register.

Some registers have limitations on the mode in which they can be accessed. The access mode codes as shown in the table are:

R = Read access allowed.
W = Write access allowed.
B = Byte (8 bit) access allowed.
S = Short (16 bit) access allowed.
L = Longword (32 bit) access allowed.

Boot Load Mode Memory Map

Board offset	R	W	B	S	L	Description
0x0000-0x0003	●	●			●	Command Register
0x0004-0x0007	●	●	●	●	●	Parameter 0 Register
0x0008-0x000B	●	●	●	●	●	Parameter 1 Register
0x000C-0x000F	●	●	●	●	●	Parameter 2 Register
0x0010-0x0013	●	●	●	●	●	Parameter 3 Register
0x0014-0x0017	●	●	●	●	●	Parameter 4 Register
0x0018-0x001B	●	●	●	●	●	Parameter 5 Register
0x001C-0x001F	●	●	●	●	●	Parameter 6 Register
0x0020-0x0023	●	●	●	●	●	Parameter 7 Register
0x0024-0x0027	●	●	●	●	●	Parameter 8 Register
0x0028-0x002B	●	●	●	●	●	Parameter 9 Register
0x002C-0x002F	●	●	●	●	●	Parameter 0xA Register
0x0030-0x0033	●	●	●	●	●	Parameter 0xB Register
0x0034-0x0037	●	●	●	●	●	Parameter 0xC Register
0x0038-0x003B	●	●	●	●	●	Parameter 0xD Register
0x003C-0x003F	●	●	●	●	●	Parameter 0xE Register
0x0040-0x00FB	●		●	●	●	Reserved
0x00FC	●	●	●			Command Interrupt Register
0x00FD-0x00FF	●		●	●	●	Reserved
If VME A24 or VME A32 slave address space is selected:						
0x0100-0x03FF	●		●	●	●	Reserved
0x0400-0x1FFF	●		●	●	●	ASCII Print Buffer
0x2000-0x3FFF	●		●	●	●	Reserved
0x4000-0xFFFF	●	●	●	●	●	User Application Buffer.
If VME A32 slave address space is selected:						
0x10000-End of RAM	●	●	●	●	●	Additional User Application Buffer.

2.6 Boot Load Commands

The Command register is used for both issuing boot loader commands and receiving command completion status. This register may only be accessed as a longword. Commands may use some or all of the Parameter registers. To issue a command to the MVCP, follow this sequence:

- Test the value in the Command register. The MVCP will not accept a new command until a previously issued command has completed. The Command register will contain a 0 if the previous command completed successfully. If a bad command or invalid parameter was issued, the Command register will contain the value 0x00000BAD, and the Parameter 0 register will contain the offending command. The ASCII Print buffer will contain the invalid parameter and value. If the Command register contains a Bad Command status, a new command may be issued, but it is advisable that the cause of the failure be investigated before issuing further commands.
- Write any required parameters to the appropriate Parameter registers.
- Write the command code to the Command register. The Command register will clear to indicate successful command completion. After the command completes, the host may read any returned parameters.

Boot Load Commands

Command	Code
Software Reset	1
NOP	2
Board Information	4
Write Longword	5
Read Longword	6
Start Application	7
Flash Sector Load	8
External Loop Back Diagnostic	9
Command Interrupt Diagnostic	0xA
VME Bus Master DMA Diagnostic	0xB
VMEbus Interrupt Diagnostic	0xC
Move Board VME Location	0xF

Boot Load Status Messages

Status	Code
Command Complete	0x00
Bad Command	0xBAD

The commands are described in detail on the following pages.

Software Reset Command

Command Code

0x01

Summary

Reset the MVCP to the powered-on state.

Description

This command initializes the MVCP to its powered-on state. It can be used to re-run the board's Power On Self Test.

Parameters Used/Returned

None.

Results Upon Completion

Board is initialized to the powered-on state.

Notes

Do not access the board for four seconds after issuing this command. Do not poll this command for completion.

NOP Command

Command Code

0x02

Summary

No operation.

Description

This command can be used to test the boot loader command interface. It performs no action other than causing the CPU to execute its command service routine.

Parameters Used/Returned

None.

Results Upon Completion

The Command register is cleared. No other action is taken.

Notes

Board Information Command

Command Code

0x04

Summary

Report board information.

Description

This command is used to read configuration and revision information stored in the MVCP.

Parameters Used/Returned

Parameter 0 = Returned board part number in BCD.
Parameter 1 = Returned board revision letter in ASCII.
Parameter 2 = Returned board rework level in BCD.
Parameter 3 = Returned board serial number in BCD.
Parameter 4 = Returned firmware part number in BCD.
Parameter 5 = Returned firmware dash number in BCD.
Parameter 6 = Returned firmware revision letter in ASCII.
Parameter 7 = Returned RAM size.
Parameter 8 = Returned size of Flash EPROM available to the user.
Parameter 9 = Returned VIC revision.
Parameter 0xA = Returned ESCC 0 revision.
Parameter 0xB = Returned ESCC 1 revision (0 if not installed)

Results Upon Completion

Parameter registers 0-0xB will contain the values noted above.

Notes

Write Longword Command

Command Code

0x05

Summary

Write data to the board.

Description

This command can be used to download a user application to the MVCP. Write the Parameter 0 register with the longword address relative to the MVCP CPU. This only has to be done once for sequential longword writes because this command automatically increments the Parameter 0 register to the next longword address. The valid address range is from 0x4000 to the end of on board RAM. Once started, the application may modify an address below 0x4000 (such as the interrupt vector table).

Write the data to the Parameter 1 register. When the command completes, the Parameter 0 register will auto-increment to the next longword address.

Parameters Used/Returned

Parameter 0 = Longword address relative to the MVCP CPU.
Parameter 1 = Data.

Results Upon Completion

When the command completes, the Parameter 0 register will auto-increment to the next longword address. This facilitates the issuance of successive Write Longword commands without the need to continually update the Parameter 0 value.

Notes

Read Longword Command

Command Code

0x06

Summary

Read data from the board.

Description

This command can be used to verify the user application download. Write the Parameter 0 register with the longword address relative to the MVCP CPU. This only has to be done once for sequential longword reads because this command automatically increments the Parameter 0 register to the next longword address. A valid longword address is anywhere from 0x4000 to the end of on board RAM.

Upon completion, the Parameter 0 register will auto-increment to the next longword address. the Parameter 1 register will contain the data longword read from the board.

Parameters Used/Returned

Parameter 0 = Longword address relative to the MVCP CPU.
Parameter 1 = Returned data.

Results Upon Completion

The Parameter 0 register will auto-increment and contain the address of the next longword. This facilitates the issuance of successive Read Longword commands without the need to continually update the Parameter 0 value.
The Parameter 1 register will contain the data read.

Notes

Start Application Command

Command Code

0x07

Summary

Start the downloaded application.

Description

This command is used to initiate the application on the MVCP after it has been downloaded to the board.

Write the Parameter 0 register with the application start address. The application may be located in either RAM or Flash EPROM. The Command register will clear just before starting the application. The application can reset the board and restart the boot loader at any time by allowing the watchdog timer to expire.

The front panel LED will be turned off before starting the application. It is recommended that one of the first application instructions set the LED to yellow or green as a visual cue that the application has started.

Parameters Used/Returned

Parameter 0 = Application start address.

Results Upon Completion

The Command register is cleared just before the application starts.

Notes

Flash Sector Load Command

Command Code

0x08

Summary

Store 32K of user application into the Flash EPROM.

Description

This command is used to store the user application into the MVCP's non-volatile Flash EPROM. Write the Parameter 0 register with the command key of 0xA1B2C3D4. This key is used to prevent accidental execution of this command. An invalid command key will generate a Bad Command status.

Write the Parameter 1 register with the Flash EPROM address that the User Application Buffer will be stored into. The Flash EPROM address must be 32K aligned. Addresses 0x80010000 and 0x80018000 are valid for the standard size Flash EPROM. Addresses 0x80010000 through 0x80078000 are valid for the optional larger Flash EPROM. An invalid address will generate a Bad Command status.

Write the User Application buffer with up to 32K of the user application. This buffer may be directly written if the board is configured as a VME A32 or A24 slave. If the board is configured as a VME A16 slave, the Write Longword command must be used to write to the buffer. After the buffer is written, it is recommended that the host read it back and verify the data.

The command will verify that the application buffer was loaded without error by comparing it to the data loaded into the Flash EPROM (the Flash EPROM data can also be verified using the Read Longword command).

User applications larger than 32K will require several of these commands. The standard Flash EPROM can be used to store up to 64K of a user application (0x80010000-0x8001FFFF). An optional larger Flash EPROM may be ordered to allow up to 448K of storage (0x80010000-0x8007FFFF).

While the command is executing, the LED will be set to yellow.

Parameters Used/Returned

Parameter 0 = Command key (0xA1B2C3D4).

Parameter 1 = Flash EPROM address.

User Application buffer = 32K of user application.

Results Upon Completion

If the store fails, the LED will be set to red and the command register will contain 0xBAD102SS (SS = Failure code). The ASCII Print buffer will contain failure information. The board should be reset. If this status occurs repeatedly, the Flash EPROM should be replaced.

Notes

This command typically takes less than five seconds to complete. However, as Flash EPROM programming times vary with temperature, this command may take up to two minutes to complete.

External Loop Back Diagnostic Command

Command Code

0x09

Summary

Run the External Loop Back Diagnostic test.

Description

This diagnostic is used as a functional test of the MVCP serial interface. An external loop back connector must be installed for the ports being tested. Serial line conditioners with switches should be configured to send TXCLK and receive RXCLK. Parameters 0-3 specify the serial line conditioner types. Valid parameters are:

Value	Serial Line Conditioner Type
0	No serial line conditioner installed
0x188	IEEE-188C serial line conditioner
0x232	RS-232 or buffered TTL serial line conditioner
0x422	RS-422 serial line conditioner
0x422A	RS-422 serial line conditioner on P2 I/O Adapter
0x422B	RS-422I serial line conditioner, no loopback connector
0x485	RS-485 serial line conditioner

Write the Parameter 4 register with the repeat count. A repeat count of 0 will loop forever. An invalid parameter will generate a Bad Command status.

While the test is running, the LED will be set to yellow. A Software Reset command may be issued to stop the diagnostic and reset the board.

Parameters Used/Returned

Parameter 0 = Ports 0-3 serial line conditioner type.
Parameter 1 = Ports 4-7 serial line conditioner type.
Parameter 2 = Ports 8-B serial line conditioner type.
Parameter 3 = Ports C-F serial line conditioner type.
Parameter 4 = Repeat count.

Results Upon Completion

If the diagnostic fails, the LED will be set to red and the command register will be set to 0xBAD031SS (SS = Failure code). The ASCII Print Buffer will contain failure information. The board should be reset.

This command will take a long time to complete if a large repeat count is specified.

Notes

See Appendix A for a pinout of the loopback connectors.

Command Interrupt Diagnostic Command

Command Code

0x0A

Summary

Run the VMEbus Generated Command Interrupt Diagnostic test.

Description

This diagnostic test will check that the Command Interrupt register can generate a CPU interrupt. To run the diagnostic:

- Write the Command register with 0xA. Wait at least 1 ms.
- Write the Command Interrupt register (0xFC) with a byte of 3. Wait at least 10 ms.

The Command register should now be cleared if the diagnostic passed.

The LED will be set to yellow when the Command register is written. The command will not complete until the Command Interrupt register is written. A Software Reset command may be issued to stop the diagnostic and reset the board.

Parameters Used/Returned

Command Interrupt register = 3.

Results Upon Completion

The Command register will be cleared if the diagnostic passed. If the diagnostic fails, the LED will be set to red and the command register will be set to 0xBAD032SS (SS = POST sub-test number). The ASCII Print Buffer will contain failure information. The board should be reset.

Notes

VMEbus Master DMA Diagnostic Command

Command Code

0x0B

Summary

Run the VMEbus Master DMA Diagnostic test.

Description

The MVCP will become VMEbus master and run several tests using 64K of external VMEbus memory. This memory must support unaligned transfers as required by the VMEbus specification. Write the Parameter 0 register with the beginning address of the external memory block. The address must be longword aligned. Write the Parameter 1 register with the VMEbus request level. Some VME backplanes support only a specific level. Valid parameters are:

Value	Bus Request Level Tested
0	VMEbus request level 0
1	VMEbus request level 1
2	VMEbus request level 2
3	VMEbus request level 3
4	Test all four bus request levels (0-3)

Write the Parameter 2 register with the VMEbus address modifier. The VMEbus address modifier determines the transfer's address size and space. Valid parameters:

Value	Addressing Mode	Value	Addressing Mode
0x00	Test all A32 address modifiers	0x30	Test all A24 address modifiers
0x08	A32 user D64 block space (VIC64 installed)	0x38	A24 user D64 block space (VIC64 installed)
0x09	A32 user data space	0x39	A24 user data space
0x0A	A32 user program space	0x3A	A24 user program space
0x0B	A32 user D32 block space	0x3B	A24 user D32 block space
0x0C	A32 supervisor D64 block space (VIC64 installed).	0x3C	A24 supervisor D64 block space (VIC64 installed)
0x0D	A32 supervisor data space	0x3D	A24 supervisor data space
0x0E	A32 supervisor program space.	0x3E	A24 supervisor program space
0x0F	A32 supervisor D32 block space	0x3F	A24 supervisor D32 block space

Write the Parameter 3 register with the repeat count. A repeat count of 0 will loop forever. An invalid parameter will generate a Bad Command status.

While the diagnostic is running, the LED will be set to yellow. A Software Reset command may be issued to stop the diagnostic and reset the board.

Parameters Used/Returned

Parameter 0 = Address of the external 64K VMEbus memory block.

Parameter 1 = VMEbus request level.

Parameter 2 = VMEbus address modifier.

Parameter 3 = Repeat count.

Results Upon Completion

This command may take a long time to complete with multiple parameters being tested or with a large repeat count. If the diagnostic fails, the LED will be set to red and the command register will be set to 0xBAD033SS (SS = Failure code). The ASCII Print Buffer will contain failure information. The board should be reset.

Notes

VMEbus Interrupt Diagnostic Command

Command Code

0x0C

Summary

Run the VMEbus Interrupt Logic Diagnostic test.

Description

This test will verify that the MVCP can issue VMEbus interrupts at a user specified level and vector. Write the Parameter 0 register with the desired VMEbus interrupt level (1-7). Write the Parameter 1 register with the desired VMEbus interrupt vector (0-0xFF). An invalid parameter will generate a Bad Command status.

Parameters Used/Returned

Parameter 0 = VMEbus Interrupt level.
Parameter 1 = VMEbus Interrupt vector.

Results Upon Completion

The Command register is cleared. The interrupt is generated to the VMEbus.

Notes

Move Board VME Location Command

Command Code

0x0F

Summary

Modify the MVCP's VMEbus base address and shared memory size.

Description

The MVCP's VMEbus base address and shared memory size are set by the configuration jumpers and rotary switches SW1 and SW2. This command can be used to change these parameters dynamically.

Write the Parameter 0 register with the board's new VMEbus base address. Only the most significant address byte may be modified as shown below:

Example parameter for VME A32 slave address space: 0xE2000000

Example parameter for VME A24 slave address space: 0x7A0000

Example parameter for VME A16 slave address space: 0xF500

Write the Parameter 1 register with the VMEbus shared memory window size. The size must be a power of 2 and at least 256 bytes. A value of 0 will share the maximum size of memory possible for the slave address space selected (A32 max = 16M, A24 max = 64K, A16 max = 256 bytes).

Valid parameter examples:

0x00000000 Maximum size of memory for the slave address space.

0x00000100 256 bytes shared with the VMEbus (A16 maximum).

0x00010000 64K bytes shared with the VMEbus (A24 maximum).

0x01000000 16M bytes shared with the VMEbus (A32 maximum).

Parameters Used/Returned

Parameter 0 = New VMEbus base address.

Parameter 1 = New VMEbus shared memory window size.

Results Upon Completion

The board can be accessed at the new VMEbus location.

Notes

While the command is executing, do not access the board for at least 1 ms. Do not poll this command for completion.

2.7 Boot Loader Command Examples

```
/* Declare the board's shared memory interface. */
typedef     struct
{
        volatile unsigned long          command;
        volatile unsigned long          parameter[0xF];
}       * Mvcp_ptr;
#define     MVCP                                    ( (Mvcp_ptr) 0xE0000000L )
/* Commands. */
#define     NOP                                     2
#define     WRITE_LONGWORD                          5
#define     START_USER_APPLICATION                  7

/* Example use of NOP command. */
void    Nop_command(void);                          /* Prototype. */
void    Nop_command(void)
{
        volatile long delay;

        /* Poll with a delay of about 10us. */
        while( MVCP->command ) for(delay=10; delay; delay--);

        MVCP->command = NOP;
}

/* Example use of WRITE_LONGWORD command. */
void    Write_3_longwords(void);                    / * Prototype. */
void    Write_3_longwords(void)

{
        volatile long delay;

        while( MVCP->command ) for(delay=10; delay; delay--);
        MVCP->parameter[0] = 0x4000;                /* Set up address only once. */
        MVCP->parameter[1] = 0x12345678L;           /* Fake instruction. */
        MVCP->command = WRITE_LONGWORD;

        while( MVCP->command ) for(delay=10; delay; delay--);
        MVCP->parameter[1] = 0xABCD1234L;           /* Fake instruction. */
        MVCP->command = WRITE_LONGWORD;

        while( MVCP->command ) for(delay=10; delay; delay--);
        MVCP->parameter[1] = 0x000000A1L;           /* Fake instruction. */
        MVCP->command = WRITE_LONGWORD;
}
/* Example use of START_APPLICATION command. */
void Start_application_command(void);               /* Prototype. */
void Start_application_command(void)

{
        volatile long delay;

        while( MVCP->command ) for(delay=10; delay; delay--);
        MVCP->parameter[0] = 0x4000;                /* Application start address. */
        MVCP->command = START_APPLICATION;
}
```

2.8 User Application Notes

The Power On Self Test (POST) is executed every time the board is reset or powered up and will take up to four seconds to complete, depending upon the amount of RAM installed on the board. After the POST completes, read the Command register (offset 0) to verify successful completion status. If the POST failed, do not issue any boot loader commands except Software Reset to the MVCP.

The user application should be linked to run at the downloaded address. The user application may be written to any location in the User Application Buffer, but it is preferred that the application start at address 0x4000.

If the board is in A32 space, all of the on board RAM is shared with the VMEbus. The user application may be directly written into the VMEbus shared memory. If board is in A24 space, only the first 64K of the MVCP's RAM is shared with the VMEbus. If the user application will fit between 0x4000 and 0xFFFF, it may be directly written into the VMEbus shared memory. The Write Longword command must be used to write to addresses above 0xFFFF. If board is in A16 space, only the first 256 bytes of RAM is shared on the VMEbus. The user application must be downloaded using Write Longword commands.

Download the application to the board by either writing the code directly into the shared RAM region or by using the Write Longword command. It is recommended that the application code be read back and verified. Issue the Start User Application command. The LED will be turned off before starting the application. Once started, a user application takes total control of the board. The application should never return. (If for some reason the application does return, the board will perform a Software Reset command and become operational after four seconds.) The user application can reset the board and return to the boot loader by allowing the watchdog timer to expire.

The user application can use the CPU to access the VMEbus using the 'MOVES' instruction. The user application must set up a bus error and deadlock interrupt handler to use this feature.

Before starting the user application, the boot loader initializes the CPU registers as follows:

- The CPU is set to run in supervisor state by register SR.
- The stack pointer is set to use 8K of memory from 0x2000 to 0x3FFF by register A7. The user application may change the stack location.
- Interrupts are disabled by register SR.
- The interrupt vector base address is set to 0 by register VBR.
- MMU (tt0, tt1) set to disable caching for any address with A27 set.

The user application may change any of the following:

- CPU interrupt level (sr).
- CPU vector base register (vbr).
- CPU stack (a7) location and size.
- CPU MMU (tt0, tt1) caching.
- Any address below 0x4000 (such as the interrupt vector table).

The instruction and data caches are invalidated with the 'CINVA BC' instruction. The instruction and data caches are disabled for all addresses with A27 set by registers IACR0 and DACR0 (all on board hardware registers have A27 set in their memory map). The data cache is also disabled for all VMEbus accesses by register DACR1. The instruction and data caches are enabled by register CACR. The user application should not change these registers.

The source and destination function code registers SFC and DFC are set to 3 to allow access to the VMEbus using the 'MOVES' instruction. The user application should not change these registers. The default VMEbus address space is A32 supervisor data. The user application may change the VMEbus address space by using the VIC address_modifier_source register.

It is recommended that one of the first user application instructions set the LED to yellow or green as a visual cue that the application has started.

If the user application has been stored in the Flash EPROM, downloading is not required. The Start User Application command can be used to start an application located anywhere in the Flash EPROMs If jumper H1 pins 13&14 is installed, the user application will be automatically executed after the POST completes. For fastest execution, the application stored in the Flash EPROMs may copy itself to and execute from the faster RAM memory.

The following test application is provided as an example of the downloading and starting procedure. The test will loop forever changing the LED color from off, green, yellow, and red. The board must be reset to stop the test application.

Download Test Application Source Code

```
#define LOOP_FOREVER    while(1)

#define     LED         ( *(volatile unsigned long *)0x88200004 )
#define     RED         0x1C
#define     YELLOW      0x1D
#define     GREEN       0x1F
#define     OFF         0x1E

/* Since the watchdog must be written, write it with the DELAY macro. */
#define     RESET_WATCHDOG_TIMER *(volatile unsigned long *)0x88200000 = 0
#define     DELAY                                                   \
{                                                                   \
    long delay = 2000000;                                           \
    do RESET_WATCHDOG_TIMER;    while(—delay); \
}

void   Download_test_application(void);             /* Prototype. */
void Download_test_application(void)
{
    LOOP_FOREVER
    {
        LED = OFF;          DELAY;
        LED = GREEN;        DELAY;
        LED = YELLOW;       DELAY;
        LED = RED;          DELAY;
    }
}
```

Download Test Application Binary Data

Address	Data		Address	Data
4000	2C3C8820		4030	20805381
4004	00002A3C		4034	66F62045
4008	88200004		4038	701D2080
400C	283C001E		403C	22042046
4010	84802045		4040	70002080
4014	701E2080		4044	538166F6
4018	22042046		4048	2045701C
401C	70002080		404C	20802204
4020	538166F6		4050	20467000
4024	2045701F		4054	20805381
4028	20802204		4058	66F660B6
402C	20467000			

2.9 MVCP Hardware Description

The MVCP features a 40 MHz Motorola 68EC030 32-bit microprocessor (MPU), two Siemens SAB82538 Enhanced Serial Communication Controllers (ESCC8), an integrated VME interface controller (VIC), 32 bit wide high-speed EDO Dynamic RAM, and up to four serial line conditioners. These components are described in the following sections.

32-Bit Microprocessor

The Motorola 68EC030 CPU used on the MVCP performs startup tests of the entire board, initializes the VIC, accepts commands from the host system, returns command and interface status, and supervises the ESCC8 chips. In normal operation, the host system passes commands to registers on the MVCP. These commands can be either interface commands, which are processed by the 68EC030, or protocol commands, which are passed to the appropriate protocol handler. The CPU is also responsible for providing graceful handling of interface error conditions.

The table below is the hardware register memory map as viewed by the MVCP's CPU. Some registers have limitations on the mode in which they can be accessed. The access mode codes as shown in the table are:

R = Read access allowed.
W = Write access allowed.
B = Byte (8 bit) access allowed.
S = Short (16 bit) access allowed.
L = Longword (32 bit) access allowed.

Memory Map

Board offset	R	W	B	S	L	Description
0x00000000-0x003FFFFF	●	●	●	●	●	Installed Memory (256K to 4M)
0x000000FC		●	●			VMEbus Generated Command Interrupt
0x00004000-0x003FFFFF	●	●	●	●	●	User Application download area
0x80000000-0x8000FFFF	●		●			Flash EPROM - Macrolink firmware only
0x80010000-0x8001FFFF	●	●	●			Flash EPROM for user application
0x80020000-0x8007FFFF	●	●	●			Optional Flash EPROM area
0x88100100-0x88101FFF	●	●	●			8Kx8 EEPROM (0-0xFF Macrolink only)
0x88200000-0x88200003		●			●	Watchdog Timer
0x88200004-0x88200007		●			●	LED
0x88200004-0x88200007	●				●	Board Options
0x88200008-0x8820000F		●			●	VME address mask and compare
0x88200014-0x88200017		●			●	Port Clock Select
0x88200018-0x8820001B	●	●	●			82C54-2 Timer
0x88300008-0x883000FF	●	●	●			Cypress VIC64 or optional VIC68
0x884FFE00-0x884FFFFF	●	●	●	●		ESCC0 (Ports 0-7)
0x88500000-0x885001FF	●	●	●	●		ESCC1 (Ports 8-F)

Address Decoding

The board's VMEbus address is selected by two hexadecimally-coded switches and decoded by an 8 bit comparator. The addressing mode (A32, A24, or A16) used by the board is selected by one of the positions of Jumper H1. This enables corresponding bus transceiver circuits on the board.

EPROM, EEPROM, and On-Board Memory

The MVCP uses high-speed EDO DRAM arranged in two 32-bit wide banks. A shared memory interface allows access by both the MVCP's CPU and the host. Shared memory is used for data and program storage, commands, parameters, and board status. The CPU uses the memory for stack, variable, and program storage.

A Flash EPROM provides storage for the MVCP's firmware and a user application. The first 64K of Flash EPROM stores the Macrolink boot loader firmware. The user must not alter this part of the Flash EPROM. The standard size Flash EPROMs can store up to 64K of a user application. Larger Flash EPROMs may be ordered to allow up to 448K of user application storage. If configuration jumper H1 pins 13 & 14 is installed, the user application stored in the Flash EPROM will be automatically executed after the POST completes. The Start User Application command can also be used to start the stored user application at any address. For faster application execution, it is recommended that the user application be copied from the Flash EPROM into the User Application Buffer in the shared memory area.

Programming the Flash EPROM typically takes less than five seconds to complete. However, as Flash EPROM programming times vary with temperature, programming may take as long as two minutes.

An 8K x 8 EEPROM is used to store information specific to the board (configuration, part number, revision, etc.). The first 256 bytes (88100000-881000FF) are reserved for factory use. The remainder (88100100-88101FFF) is available for any user-defined purpose.

VMEbus Generated Command Interrupt Register

This register is used to generate an interrupt to the MVCP from the host system. It resides in the shared memory region. Writing the byte value 0x03 to this register causes a CPU interrupt using VIC LIRQ1 to be generated. The user application can use this interrupt to start a command or for any other user defined purpose.

Watch Dog Timer

The watchdog timer is used to ensure that the board's hardware and software continue to operate properly. The watchdog timer resets the board if the user application fails to write to this register at least every 1.5 seconds. This is a write only longword register. The watchdog timer can not be disabled.

The user application can reset the board and return to the boot loader by allowing the watchdog timer to expire.

LED Latch & Board Options Registers

The longword format LED Latch is a write-only memory-mapped hardware register that is used to drive the tri-color Board Status LED. The value in bits 0 and 1 determine the color of the LED (green/yellow/red/off). The board initially comes up with all bits cleared. The LED will be red. After the POST (Power On Self Test) has completed the LED will be set to yellow. After the board has been initialized, the LED will be changed to green.

LED Latch (Write)	
Value	**LED Color**
1F	Green
1D	Yellow
1C	Red
1E	Off

A longword read at address 0x882000004 will access the Board Options register. The firmware uses this information to control operating characteristics of the board. The function of each bit is defined in the table on the next page.

The user application may read this register to determine the state of configuration jumper H1 17&18. This jumper is reserved for any use by the user application. Configuration jumper bits will read 1 if the jumper is installed.

Address Switches and CONFIG Jumpers Register		
Bits	**Description**	**Comments**
0-7	Address switches SW1 and SW2	Most significant byte of the board s VMEbus base address.
8	H1 15&16	Continue after POST failure
9	H1 13&14	Enable auto-execution of the user application
10	H1 11&12	Power On Self Test loop mode
11	H1 9&10	VIC interprocessor interface mode
12	H1 7&8	VME system controller mode
13	H1 5&6	VME A16 slave address space
14	H1 3&4	VME A24 slave address space
15	H1 1&2	VME A32 slave address space
16	H1 17&18	Reserved for any use by the user application
17-19	These bits are reserved.	
20-31	Not used (The application should mask these bits out)	

VME Address Mask and Compare Registers

These registers control the compare logic used to manipulate the upper address bits on the VMEbus. Refer to the Cypress Semiconductor CY7C964 data sheet for more information.

Port Clock Select

The baud rate clock is derived from the 10 MHz ESCC clock, or an on-board 3.6854 MHz oscillator. The selection between these two sources is determined by the clock mode selected for the ESCCs. Refer to the *Clock Modes* section in the ESCC8 User's Manual for more information.

The ESCC8 can also derive the baud rate clock from an external source. The write-only longword Port Clock Select register is used to select the external source for the baud rate generator clock for each port. Data bit zero (0) selects the source for port 0. Data bit 15 selects the source for port F (15). Setting the bit (1) selects the on-board oscillator as the source. Clearing the bit (0) selects the RXCLK line as the source.

Data Bit 0 = Port 0,
Data Bit 1 = Port 1,
|
|
Data Bit 15 = Port 15 (F)
Data Bit setting: 1 = On-board oscillator, 0 = RXCLK.

82C54-2 Timer

The 82C54 is a triple programmable timer. It can be used to generate precise CPU interrupts. Timer 0 and 2 clocks are connected to a 10MHZ oscillator. Timer 0's output is connected to the clock of timer 1 and acts as a prescaler. Timer 1 can generate an interrupt using VIC LIRQ6. Timer 2 can generate an interrupt using VIC LIRQ5. The timers should be initialized before their interrupts are enabled.

Refer to the 82C54 in the AMD peripherals data manual for register definitions.

VIC64/VIC68

A VIC64 VMEbus Interface Controller controls data transfers between the MVCP and the VMEbus. The VIC64 supports high speed master and slave D64 block transfers. Using D64 block transfers the board can transfer VMEbus data at over 45MB per second. The MVCP may be equipped with a VIC68 if D64 ability is not required.

The VIC registers can only be accessed by the local (MVCP CPU) bus. The ESCCs, VIC, and CPU all share a common local bus. The user application can program the VIC local_interrupt_control_4 register to generate a CPU interrupt when the host writes the Command Interrupt register (board offset 0xFC) as a byte with the value of 3. This interrupt can be used for any user defined purpose, such as starting a command.

The VIC interprocessor communication facilities are enabled by installing jumper H1 positions 9&10. The user must store an ICF compatible user application in the Flash EPROM before using this feature. The H1 13&14 Auto-execute jumper must also be installed.

The VME bus is mapped at address hex 00000000 - FFFFFFFF with a Function code of 3. Before accessing the VME bus the firmware must set the VIC64 bus controller to the proper values. The address space (address modifier) to be used in master mode will be set up by the CPU prior to the initiation of data transfers on the VME bus.

The VIC64 provides a tic timer that can be used to time certain software functions. This timer interrupts on level 2.

All of the VIC's LIRQs must be programmed for active low input. Except for LIRQ3, each of the VIC's LIRQs can be programmed to interrupt the MC68EC030 CPU on any level (1-6). LIRQ3 must be programmed to issue a level 3 interrupt.

The seven VIC LIRQs are connected as follows:

- LIRQ1 VME bus generated command interrupt.
- LIRQ2 VIC tic timer.
- LIRQ3 SAB82538 ESCCs.
- LIRQ4 Unused.
- LIRQ5 82C54-2 timer 2.
- LIRQ6 82C54-2 timer 0 chained to timer 1.
- LIRQ7 Unused.

The VIC interface configuration register is set up by the firmware to support metastability interval (bit 2) and deadlock with halt signaling (bit 4).

The following VIC registers are set by the boot loader to ensure proper hardware timing when using the VIC to transfer block data:

- local bus timing = EXTRA_DS_DEASSERTION_CLOCK
- interface configuration = METASTABILITY | DEDLK_AND_LBERR
- slave select 0 control 0 = D32 | A32_SPACE | ACCELERATE_TRANSFERS
- slave select 0 control 1 = FIRST_2_5_CLOCKS_SECOND_2_5_CLOCKS
- block transfer definition = D64_SLAVE_ENABLE | AMSR_ENABLE | VME_BUS_256_BYTE_CROSSINGS | LOCAL_256_BYTE_CROSSINGS;

When transferring block D64 data, a minimum interleave period of one must be used in VIC register block transfer control.

Refer to the Cypress VIC64 data manual for the VIC register definition.

Enhanced Serial Communications Controllers

The Siemens SAB82538 (ESCC8) is an intelligent data communications controller with eight independent full duplex serial ports. The ESCC8 supports asynchronous and synchronous communication with data rates of up to 2 megabits per second. Each port contains a 64 byte FIFO per direction (transmit and receive). These parts have a 16 bit interface, which allows for a very high data throughput. The MVCP supports two ESCC8 chips for a total of sixteen serial ports. An eight port board will have one ESCC chip (ESCC 0). A sixteen port board will have two ESCC chips (ESCC 0 and ESCC 1). ESCC 0 controls ports 0-7, ESCC 1 controls ports 8-F. Each of the ESCC chips also have 4 programmable ports A-D. For a complete description of the ESCC 8 parts, refer to the Siemens SAB8532 data sheets.

All bits in port A of both chips should be configured for output. These bits are used by the serial line conditioners for DTR. ESCC chip 0, port A, bit 0, is DTR 0. ESCC chip 1, port A, bit 7, is DTR F. All bits in port B should be configured for input. These bits are used by the serial line conditioners for DSR. ESCC chip 0, port B, bit 0, is DSR 0. ESCC chip 1, port B, bit 7, is DSR F. Configure port C bits for input to support RING on RS-232 serial line conditioners. Configure port C bits for output to support other serial line conditioners. ESCC chip 0, port C, bit 0, is RS-232 RING 0. ESCC chip 1, port C, bit 7, is RS-232 RING F. All bits in port D should be configured for output. These bits are reserved for Macrolink use.

The receive clock source register bits 0-F are used to select the clock source for ESCC's RXCLK ports 0-F. A set bit selects the on-board 3.6864MHZ oscillator for RXCLK. A clear bit selects the serial line conditioner's receive clock for RXCLK.

After an ESCC line is powered up (channel configuration 0 register bit 7 is set) do not access the ESCC for at least 3 transmit clocks.

To generate an interrupt, the ESCCs must be programmed for interrupt slave mode with a slave address of seven and the INT pin should be programmed for push-pull active low output.

The interrupt request of each ESCC chip is connected through a ping-pong select PAL to the VIC's LIRQ3. Because the ESCC supplies the interrupt vector, the VIC must be programmed to issue a level 3 interrupt to the MC68EC030 CPU for LIRQ3.

Port Priority

The two types of interrupt priority supported are:

Fixed interleaved Channel 0 will have the highest priority and channel 7 the lowest for each ESCC8 device. The order of priority by port is 0,8,1,9,2,10,3,11,4,12,5,13,6,14,7,15 if two ESCC8's are used.

Rotating The channel last serviced drops to the lowest priority and the ports following are rotated up. For example assuming a 0,1,2,3,4,5,6,7 starting order, when channel 4 asserts an interrupt without any higher channel requesting service then channel 4 would be serviced and a new order would follow: 5,6,7,0,1,2,3,4. The priority will "ping-pong" between ports 0-7 and 8-15 when two ESCC8's are used. The ESCC8s are on interrupt level 3. The interrupt slave address in the Interrupt Port Configuration Register should be set to hex 7 for all ports.

Technical Information

This appendix provides signal name and pinout information for the interface connectors used on the MVCP, information regarding Macrolink's implementation of the RS485 interface, and loopback connector pinouts.

A.1 DB25 Interface Panel Signals

RS232/MIL188C Signals

RS232/MIL188C DB25 Signals

Signal	Pin	Description
CGND	1	Protective Ground
TXD	2	Transmit Data
RXD	3	Receive Data
RTS	4	Request to Send
CTS	5	Clear to Send
DSR	6	Data Set Ready
SGND	7	Signal Ground
DCD	8	Data Carrier Detect
RXCLK	17	Receive Clock
DTR	20	Data Terminal Ready
RI	22	Ring Indicate
TXCLK	24	Transmit Clock

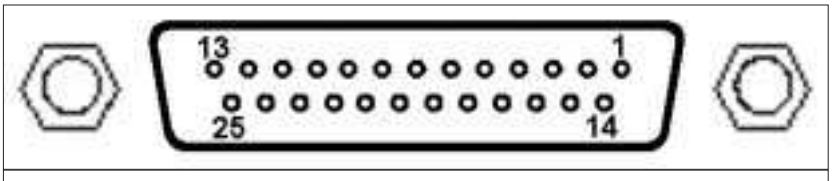


Figure A - 1 – DB-25 Connector Pinout

RS422/485 Signals

RS422/485 DB25 Signals		
Signal	**Pin**	**Description**
CGND	1	Protective Ground
TXDA	2	Transmit Data A
RXDA	3	Receive Data A
RTSA	4	Request to Send A
CTSA	5	Clear to Send A
SGND	7	Signal Ground
RXCLKB	9	Receive Clock B
TXCLKB	11	Transmit Clock B
CTSB	13	Clear to Send B
TXDB	14	Transmit Data B
RXDB	16	Receive Data B
RXCLKA	17	Receive Clock A
RTSB	19	Request to Send B
TXCLKA	24	Transmit Clock A

Note: Request To Send and Clear To Send are not provided as outputs on the RS485 serial line conditioners.

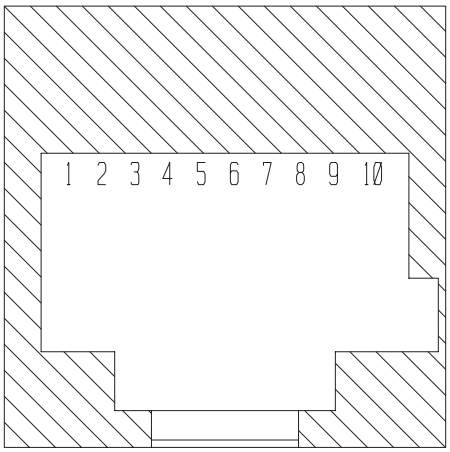


Figure A - 2 – RJ45 Connector Front View

A.2 RJ45 Interface Panel Signals

RS232/MIL188C Signals

RS232/MIL188C RJ45 Signals		
Signal	**Pin**	**Description**
RI /RXCLK	1	Ring Indicate / Receive Clock *
DSR	2	Data Set Ready
RTS	3	Request to Send
CGND	4	Protective Ground
TXD	5	Transmit Data
RXD	6	Receive Data
SGND	7	Signal Ground
CTS	8	Clear to Send
DTR	9	Data Terminal Ready
DCD/TXCLK	10	Data Carrier Detect / Transmit Clock *

* This signal is determined by jumper selection in the I/O panel. Either the RI/DCD signals or the TXCLK/RXCLK clocks can be selected. Move the jumper blocks to the positions corresponding to the signals to be supported. The jumpers are moved in two port increments.

RS422/485 Signals

RS422/485 RJ45 Signals		
Signal	**Pin**	**Description**
CTSB/RXCLKA	1	Clear to Send B / Receive Clock A *
RXDB	2	Receive Data B
RTSA/TXCLKB	3	Request to Send A / Transmit Clock B *
CGND	4	Protective Ground
TXDA	5	Transmit Data A
RXDA	6	Receive Data A
SGND	7	Signal Ground
CTSA/RXCLKB	8	Clear to Send A / Receive Clock B *
TXDB	9	Transmit Data B
RTSB/TXCLKA	10	Request to Send B / Transmit Clock A *

Note: Request To Send and Clear To Send are not provided as outputs on the RS485 serial line conditioners.

* This signal is determined by jumper selection in the I/O panel. Either the RTS/CTS signals or the TXCLK/RXCLK clocks can be supported. Move the jumper blocks to the positions corresponding to the signals to be supported. The jumpers are moved in two port increments.

A.3 MVCP RS485 Interface

The RS-485 standard specifies the electrical characteristics of drivers and receivers that can be used to implement a balanced multi-point line, also known as a "party line." The devices attached to this line will support 32 driver/receiver connect points. The transmission line used is 120 Ohm twisted pair terminated at both ends with a 120 Ohm resistor. A line length of 50 meters can be supported with this interface. When a driver is not actively driving the line it will be taken to a high impedance condition.

There is no implied protocol supported by RS-485 nor is there any hardware flow control. RS-485 as implemented by Macrolink does not have any collision recovery. It is intended to be used in a system employing one "talker or initiator" and multiple "listeners or targets." The jumpers on the I/O transition panel must be strapped to the RS-422 setting.

The MVCP supports 2-wire or 4-wire RS485 configurations.

2-Wire

The 2-wire configuration transmits and receives data on the same pair of wires in a half duplex operation. A device that wants to transmit data must wait until the line is available before its data can be sent. The transmit and receive data signals are connected together to operate in 2-wire mode; TXDATA A is wired to RXDATA A and TXDATA B is wired to RXDATA B.

The receiver can be turned off under software control to keep from receiving the transmitted data. The RTS signal is used to tri-state the transmit data and clock drivers when they are idle. This function must be performed by the application.

4-Wire

In 4-wire mode the data is transmitted on 1 pair of wires and received on another pair in a full duplex operation. The master device sends data down 1 pair of wires to slave devices (TXDATA A and B). Slave devices send data to the master device on the other pair of wires (RXDATA A and B).

Clocks

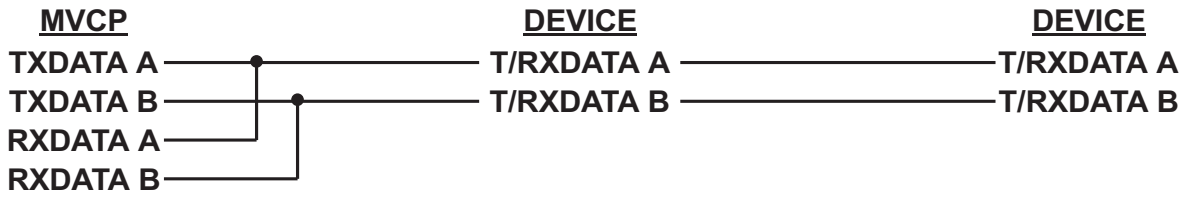
The MVCP can be used to generate transmit and receive clocks. The transmit clock is used to synchronize the data that it is being transmitted by the MVCP to another device. It is output on the transmit clock lines (TXCLK A and B). The MVCP uses the RTS signal to tri-state the transmit clock when it is idle. This function must be performed by the application. The device's transmit clock lines connect to the receive clock lines of the MVCP (RXCLK A and B). The MVCP receives synchronized data from the device using this clock.

Termination

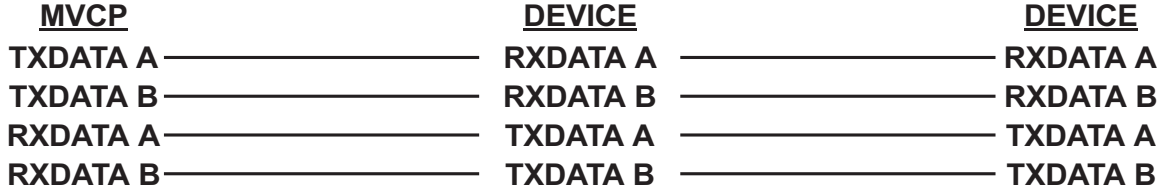
Termination is provided for the data and clock lines. The RS485 bus should be terminated at both ends. Switches are provided for each line on the RS485 serial line conditioners for this purpose. The signals are terminated when the switch is set to the ON position. In 2-wire mode, only one termination switch should be ON for each transmit/receive data and clock pair.

RS485 serial line conditioner Signal Termination Switches			
Signal	**Switch**	**Description**	**Ports**
TXDATA	SW1-1	Transmit Data	0,4,8,C
RXDATA	SW1-2	Receive Data	0,4,8,C
TXCLK	SW2-1	Transmit Clock	0,4,8,C
RXCLK	SW2-2	Receive Clock	0,4,8,C
TXDATA	SW1-3	Transmit Data	1,5,9,D
RXDATA	SW1-4	Receive Data	1,5,9,D
TXCLK	SW2-3	Transmit Clock	1,5,9,D
RXCLK	SW2-4	Receive Clock	1,5,9,D
TXDATA	SW1-5	Transmit Data	2,6,A,E
RXDATA	SW1-6	Receive Data	2,6,A,E
TXCLK	SW2-5	Transmit Clock	2,6,A,E
RXCLK	SW2-6	Receive Clock	2,6,A,E
TXDATA	SW1-7	Transmit Data	3,7,B,F
RXDATA	SW1-8	Receive Data	3,7,B,F
TXCLK	SW2-7	Transmit Clock	3,7,B,F
RXCLK	SW2-8	Receive Clock	3,7,B,F

RS485 2-WIRE INTERFACE



RS485 4-WIRE INTERFACE



A.4 Front Panel I/O Connectors

The MVCP uses AMP brand 100 pin high-density connectors for the serial interface. The mating connector is AMP catalog number 749621-9. The recommended back shells are AMP catalog number 749206-1 (right angle) or 749197-1 (straight).

The upper connector on the front of the MVCP (J3) interfaces to ports 0 through 7. The lower connector (J7) interfaces to ports 8 through F.

Note: Request To Send and Clear To Send are not provided as outputs on the RS485 serial line conditioners.

	RS232/MIL188C Interface		RS422/485 Interface	
J3, J7 Pin	**Signal**	**Description**	**Signal**	**Description**
1	TXD0/8	Transmit Data Port 0/8	TXDA0/8	Transmit Data A Port 0/8
2	DTR0/8	Data Terminal Ready Port 0/8	TXDB0/8	Transmit Data B Port 0/8
3	RXD0/8	Receive Data Port 0/8	RXDA0/8	Receive Data A Port 0/8
4	DSR0/8	Data Set Ready Port 0/8	RXDB0/8	Receive Data B Port 0/8
5	TXCLK0/8	Transmit Clock Port 0/8	TXCLKA0/8	Transmit Clock A Port 0/8
6	1SGND0/8	Signal Ground 0/8	TXCLKB0/8	Transmit Clock B Port 0/8
7	RXCLK0/8	Receive Clock Port 0/8	RXCLKA0/8	Receive Clock A Port 0/8
8	2SGND0/8	Signal Ground 0/8	RXCLKB0/8	Receive Clock B Port 0/8
9	RTS0/8	Request to Send Port 0/8	RTSA0/8	Request to Send A Port 0/8
10	CD0/8	Carrier Detect Port 0/8	RTSB0/8	Request to Send B Port 0/8
11	CTS0/8	Clear to Send Port 0/8	CTSA0/8	Clear to Send A Port 0/8
12	RING0/8	Ring Indicate Port 0/8	CTSB0/8	Clear to Send B Port 0/8
13	TXD1/9	Transmit Data Port 1/9	TXDA1/9	Transmit Data A Port 1/9
14	DTR1/9	Data Terminal Ready Port 1/9	TXDB1/9	Transmit Data B Port 1/9
15	RXD1/9	Receive Data Port 1/9	RXDA1/9	Receive Data A Port 1/9
16	DSR1/9	Data Set Ready Port 1/9	RXDB1/9	Receive Data B Port 1/9
17	TXCLK1/9	Transmit Clock Port 1/9	TXCLKA1/9	Transmit Clock A Port 1/9
18	1SGND1/9	Signal Ground 1/9	TXCLKB1/9	Transmit Clock B Port 1/9
19	RXCLK1/9	Receive Clock Port 1/9	RXCLKA1/9	Receive Clock A Port 1/9
20	2SGND1/9	Signal Ground 1/9	RXCLKB1/9	Receive Clock B Port 1/9
21	RTS1/9	Request to Send Port 1/9	RTSA1/9	Request to Send A Port 1/9
22	CD1/9	Carrier Detect Port 1/9	RTSB1/9	Request to Send B Port 1/9
23	CTS1/9	Clear to Send Port 1/9	CTSA1/9	Clear to Send A Port 1/9
24	RING1/9	Ring Indicate Port 1/9	CTSB1/9	Clear to Send B Port 1/9
25	GND	Signal Ground	GND	Signal Ground
26	GND	Signal Ground	GND	Signal Ground
27	TXD2/A	Transmit Data Port 2/A	TXDA2/A	Transmit Data A Port 2/A
28	DTR2/A	Data Terminal Ready Port 2/A	TXDB2/A	Transmit Data B Port 2/A
29	RXD2/A	Receive Data Port 2/A	RXDA2/A	Receive Data A Port 2/A
30	DSR2/A	Data Set Ready Port 2/A	RXDB2/A	Receive Data B Port 2/A
31	TXCLK2/A	Transmit Clock Port 2/A	TXCLKA2/A	Transmit Clock A Port 2/A
32	1SGND2/A	Signal Ground 2/A	TXCLKB2/A	Transmit Clock B Port 2/A
33	RXCLK2/A	Receive Clock Port 2/A	RXCLKA2/A	Receive Clock A Port 2/A

	RS232/MIL188C Interface		RS422/485 Interface	
J3, J7 Pin	**Signal**	**Description**	**Signal**	**Description**
34	2SGND2/A	Signal Ground 2/A	RXCLKB2/A	Receive Clock B Port 2/A
35	RTS2/A	Request to Send Port 2/A	RTSA2/A	Request to Send A Port 2/A
36	CD2/A	Carrier Detect Port 2/A	RTSB2/A	Request to Send B Port 2/A
37	CTS2/A	Clear to Send Port 2/A	CTSA2/A	Clear to Send A Port 2/A
38	RING2/A	Ring Indicate Port 2/A	CTSB2/A	Clear to Send B Port 2/A
39	TXD3/B	Transmit Data Port 3/B	TXDA3/B	Transmit Data A Port 3/B
40	DTR3/B	Data Terminal Ready Port 3/B	TXDB3/B	Transmit Data B Port 3/B
41	RXD3/B	Receive Data Port 3/B	RXDA3/B	Receive Data A Port 3/B
42	DSR3/B	Data Set Ready Port 3/B	RXDB3/B	Receive Data B Port 3/B
43	TXCLK3/B	Transmit Clock Port 3/B	TXCLKA3/B	Transmit Clock A Port 3/B
44	1SGND3/B	Signal Ground 3/B	TXCLKB3/B	Transmit Clock B Port 3/B
45	RXCLK3/B	Receive Clock Port 3/B	RXCLKA3/B	Receive Clock A Port 3/B
46	2SGND3/B	Signal Ground 3/B	RXCLKB3/B	Receive Clock B Port 3/B
47	RTS3/B	Request to Send Port 3/B	RTSA3/B	Request to Send A Port 3/B
48	CD3/B	Carrier Detect Port 3/B	RTSB3/B	Request to Send B Port 3/B
49	CTS3/B	Clear to Send Port 3/B	CTSA3/B	Clear to Send A Port 3/B
50	RING3/B	Ring Indicate Port 3/B	CTSB3/B	Clear to Send B Port 3/B
51	TXD4/C	Transmit Data Port 4/C	TXDA4/C	Transmit Data A Port 4/C
52	DTR4/C	Data Terminal Ready Port 4/C	TXDB4/C	Transmit Data B Port 4/C
53	RXD4/C	Receive Data Port 4/C	RXDA4/C	Receive Data A Port 4/C
54	DSR4/C	Data Set Ready Port 4/C	RXDB4/C	Receive Data B Port 4/C
55	TXCLK4/C	Transmit Clock Port 4/C	TXCLKA4/C	Transmit Clock A Port 4/C
56	1SGND4/C	Signal Ground 4/C	TXCLKB4/C	Transmit Clock B Port 4/C
57	RXCLK4/C	Receive Clock Port 4/C	RXCLKA4/C	Receive Clock A Port 4/C
58	2SGND4/C	Signal Ground 4/C	RXCLKB4/C	Receive Clock B Port 4/C
59	RTS4/C	Request to Send Port 4/C	RTSA4/C	Request to Send A Port 4/C
60	CD4/C	Carrier Detect Port 4/C	RTSB4/C	Request to Send B Port 4/C
61	CTS4/C	Clear to Send Port 4/C	CTSA4/C	Clear to Send A Port 4/C
62	RING4/C	Ring Indicate Port 4/C	CTSB4/C	Clear to Send B Port 4/C
63	TXD5/D	Transmit Data Port 5/D	TXDA5/D	Transmit Data A Port 5/D
64	DTR5/D	Data Terminal Ready Port 5/D	TXDB5/D	Transmit Data B Port 5/D
65	RXD5/D	Receive Data Port 5/D	RXDA5/D	Receive Data A Port 5/D
66	DSR5/D	Data Set Ready Port 5/D	RXDB5/D	Receive Data B Port 5/D
67	TXCLK5/D	Transmit Clock Port 5/D	TXCLKA5/D	Transmit Clock A Port 5/D
68	1SGND5/D	Signal Ground 5/D	TXCLKB5/D	Transmit Clock B Port 5/D
69	RXCLK5/D	Receive Clock Port 5/D	RXCLKA5/D	Receive Clock A Port 5/D
70	2SGND5/D	Signal Ground 5/D	RXCLKB5/D	Receive Clock B Port 5/D
71	RTS5/D	Request to Send Port 5/D	RTSA5/D	Request to Send A Port 5/D
72	CD5/D	Carrier Detect Port 5/D	RTSB5/D	Request to Send B Port 5/D
73	CTS5/D	Clear to Send Port 5/D	CTSA5/D	Clear to Send A Port 5/D
74	RING5/D	Ring Indicate Port 5/D	CTSB5/D	Clear to Send B Port 5/D
75	GND	Signal Ground	GND	Signal Ground
76	GND	Signal Ground	GND	Signal Ground

	RS232/MIL188C Interface		RS422/485 Interface	
J3, J7 Pin	**Signal**	**Description**	**Signal**	**Description**
77	TXD6/E	Transmit Data Port 6/E	TXDA6/E	Transmit Data A Port 6/E
78	DTR6/E	Data Terminal Ready Port 6/E	TXDB6/E	Transmit Data B Port 6/E
79	RXD6/E	Receive Data Port 6/E	RXDA6/E	Receive Data A Port 6/E
80	DSR6/E	Data Set Ready Port 6/E	RXDB6/E	Receive Data B Port 6/E
81	TXCLK6/E	Transmit Clock Port 6/E	TXCLKA6/E	Transmit Clock A Port 6/E
82	1SGND6/E	Signal Ground 6/E	TXCLKB6/E	Transmit Clock B Port 6/E
83	RXCLK6/E	Receive Clock Port 6/E	RXCLKA6/E	Receive Clock A Port 6/E
84	2SGND6/E	Signal Ground 6/E	RXCLKB6/E	Receive Clock B Port 6/E
85	RTS6/E	Request to Send Port 6/E	RTSA6/E	Request to Send A Port 6/E
86	CD6/E	Carrier Detect Port 6/E	RTSB6/E	Request to Send B Port 6/E
87	CTS6/E	Clear to Send Port 6/E	CTSA6/E	Clear to Send A Port 6/E
88	RING6/E	Ring Indicate Port 6/E	CTSB6/E	Clear to Send B Port 6/E
89	TXD7/F	Transmit Data Port 7/F	TXDA7/F	Transmit Data A Port 7/F
90	DTR7/F	Data Terminal Ready Port 7/F	TXDB7/F	Transmit Data B Port 7/F
91	RXD7/F	Receive Data Port 7/F	RXDA7/F	Receive Data A Port 7/F
92	DSR7/F	Data Set Ready Port 7/F	RXDB7/F	Receive Data B Port 7/F
93	TXCLK7/F	Transmit Clock Port 7/F	TXCLKA7/F	Transmit Clock A Port 7/F
94	1SGND7/F	Signal Ground 7/F	TXCLKB7/F	Transmit Clock B Port 7/F
95	RXCLK7/F	Receive Clock Port 7/F	RXCLKA7/F	Receive Clock A Port 7/F
96	2SGND7/F	Signal Ground 7/F	RXCLKB7/F	Receive Clock B Port 7/F
97	RTS7/F	Request to Send Port 7/F	RTSA7/F	Request to Send A Port 7/F
98	CD7/F	Carrier Detect Port 7/F	RTSB7/F	Request to Send B Port 7/F
99	CTS7/F	Clear to Send Port 7/F	CTSA7/F	Clear to Send A Port 7/F
100	RING7/F	Ring Indicate Port 7/F	CTSB7/F	Clear to Send B Port 7/F

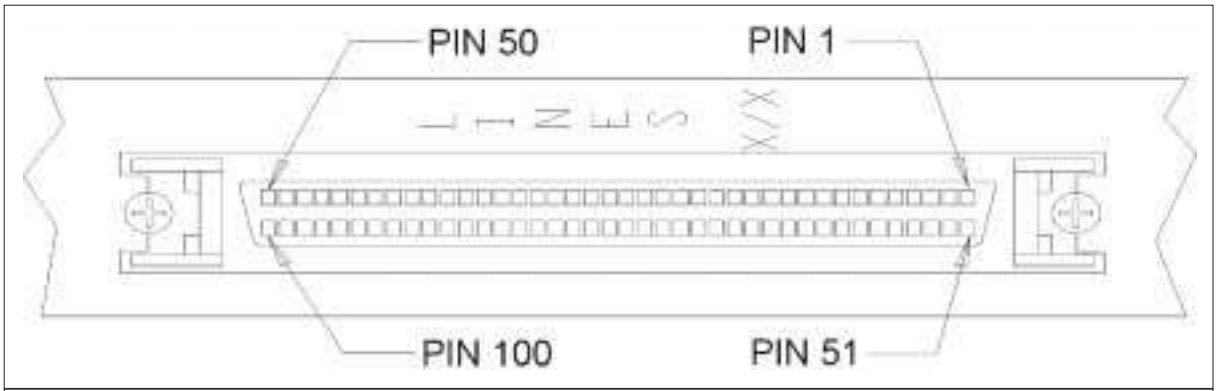


Figure A - 3 – High-Density 100 Pin Connector

A.5 VMEbus Interface Signals

VMEbus Interface Signals

Signal	Pin	Description
A01	P1A-30	Address Bus
A02	P1A-29	Address Bus
A03	P1A-28	Address Bus
A04	P1A-27	Address Bus
A05	P1A-26	Address Bus
A06	P1A-25	Address Bus
A07	P1A-24	Address Bus
A08	P1C-30	Address Bus
A09	P1C-29	Address Bus
A10	P1C-28	Address Bus
A11	P1C-27	Address Bus
A12	P1C-26	Address Bus
A13	P1C-25	Address Bus
A14	P1C-24	Address Bus
A15	P1C-23	Address Bus
A16	P1C-22	Address Bus
A17	P1C-21	Address Bus
A18	P1C-20	Address Bus
A19	P1C-19	Address Bus
A20	P1C-18	Address Bus
A21	P1C-17	Address Bus
A22	P1C-16	Address Bus
A23	P1C-15	Address Bus
A24	P2B-4	Address Bus
A25	P2B-5	Address Bus
A26	P2B-6	Address Bus
A27	P2B-7	Address Bus
A28	P2B-8	Address Bus
A29	P2B-9	Address Bus
A30	P2B-10	Address Bus
A31	P2B-11	Address Bus
D00	P1A-1	Data Bus
D01	P1A-2	Data Bus
D02	P1A-3	Data Bus
D03	P1A-4	Data Bus
D04	P1A-5	Data Bus
D05	P1A-6	Data Bus
D06	P1A-7	Data Bus
D07	P1A-8	Data Bus

VMEbus Interface Signals		
Signal	**Pin**	**Description**
D08	P1C-1	Data Bus
D09	P1C-2	Data Bus
D10	P1C-3	Data Bus
D11	P1C-4	Data Bus
D12	P1C-5	Data Bus
D13	P1C-6	Data Bus
D14	P1C-7	Data Bus
D15	P1C-8	Data Bus
D16	P2B-14	Data Bus
D17	P2B-15	Data Bus
D18	P2B-16	Data Bus
D19	P2B-17	Data Bus
D20	P2B-18	Data Bus
D21	P2B-19	Data Bus
D22	P2B-20	Data Bus
D23	P2B-21	Data Bus
D24	P2B-23	Data Bus
D25	P2B-24	Data Bus
D26	P2B-25	Data Bus
D27	P2B-26	Data Bus
D28	P2B-27	Data Bus
D29	P2B-28	Data Bus
D30	P2B-29	Data Bus
D31	P2B-30	Data Bus
Strobes, Active Low (*=0)		
AS*	P1A-18	Address Strobe
DS0*	P1A-13	Data Strobe Zero
DS1*	P1A-12	Data Strobe One
DTACK*	P1A-16	Data Transfer Acknowledge
Master Mode Signals, Active Low		
BBSY*	P1B-1	Bus Busy
BCLR*	P1B-2	Bus Clear
BERR*	P1C-11	Bus Error
BG0IN*	P1B-4	Bus Grant In Zero
BG1IN*	P1B-6	Bus Grant In One
BG2IN*	P1B-8	Bus Grant In Two
BG3IN*	P1B-10	Bus Grant In Three
BG0OUT*	P1B-5	Bus Grant Out Zero
BG1OUT*	P1B-7	Bus Grant Out One
BG2OUT*	P1B-9	Bus Grant Out Two
BG3OUT*	P1B-11	Bus Grant Out Three
BR0*	P1B-12	Bus Request Zero

VMEbus Interface Signals		
Signal	**Pin**	**Description**
BR1*	P1B-13	Bus Request One
BR2*	P1B-14	Bus Request Two
BR3*	P1B-15	Bus Request Three
Interrupt Lines, Active Low		
IRQ1*	P1B-30	Interrupt Request One
IRQ2*	P1B-29	Interrupt Request Two
IRQ3*	P1B-28	Interrupt Request Three
IRQ4*	P1B-27	Interrupt Request Four
IRQ5*	P1B-26	Interrupt Request Five
IRQ6*	P1B-25	Interrupt Request Six
IRQ7*	P1B-24	Interrupt Request Seven
IACK*	P1A-20	Interrupt Acknowledge
IACKIN*	P1A-21	Interrupt Acknowledge In
IACKOUT*	P1A-22	Interrupt Acknowledge Out
Miscellaneous Signals, Active Low		
LWORD*	P1C-13	Longword, 32 Bit Transfer
WRITE*	P1A-14	Write
SYSRESET*	P1C-12	System Reset
Power		
+5V	P1A,B,C-32	Plus Five Volts DC
+5V	P2B-1,13,32	Plus Five Volts DC
GND	P1A-9,11,15	Signal Ground
GND	P1A-17,19	Signal Ground
GND	P1B-20,23	Signal Ground
GND	P2B-2,12,22,31	Signal Ground
GND	P1C-9	Signal Ground

A.6 P2 I/O Interface Signals

The following tables list the TTL-level signals available via the backplane P2 connector when headers H3-H6 are installed on the MVCP. An optional interface adapter is available to adapt the P2 connector signals to the Macrolink interface panels. The serial line conditioners are removed from the MVCP and installed on the interface adapter. See Chapter 1 or contact Macrolink for more information.

The positioning of headers H3 and H5 select between RTS/CTS or TXCLK/RXCLX being presented on row C pins 1-32.

P2 I/O Interface Signals

Signal	Pin	Description
0TXD0	A1	Channel 0 Transmit Data
0RXD0	A2	Channel 0 Receive Data
1TXD0	A3	Channel 1 Transmit Data
1RXD0	A4	Channel 1 Receive Data
2TXD0	A5	Channel 2 Transmit Data
2RXD0	A6	Channel 2 Receive Data
3TXD0	A7	Channel 3 Transmit Data
3RXD0	A8	Channel 3 Receive Data
4TXD0	A9	Channel 4 Transmit Data
4RXD0	A10	Channel 4 Receive Data
5TXD0	A11	Channel 5 Transmit Data
5RXD0	A12	Channel 5 Receive Data
6TXD0	A13	Channel 6 Transmit Data
6RXD0	A14	Channel 6 Receive Data
7TXD0	A15	Channel 7 Transmit Data
7RXD0	A16	Channel 7 Receive Data
8TXD0	A17	Channel 8 Transmit Data
8RXD0	A18	Channel 8 Receive Data
9TXD0	A19	Channel 9 Transmit Data
9RXD0	A 20	Channel 9 Receive Data
ATXD0	A 21	Channel A Transmit Data
ARXD0	A 22	Channel A Receive Data
BTXD0	A 23	Channel B Transmit Data
BRXD0	A 24	Channel B Receive Data
CTXD0	A 25	Channel C Transmit Data
CRXD0	A 26	Channel C Receive Data
DTXD0	A 27	Channel D Transmit Data
DRXD0	A 28	Channel D Receive Data
ETXD0	A 29	Channel E Transmit Data
ERXD0	A 30	Channel E Receive Data
FTXD0	A 31	Channel F Transmit Data
FRXD0	A 32	Channel F Receive Data

P2 I/O Interface Signals				
	H3 & H5 Set For RTS/CTS		H3 & H5 Set For TXCLK/RXCLK	
Pin	Signal	Description	Signal	Description
C1	0RTS0	Channel 0 Request To Send	0TXCLK0	Channel 0 Transmit Clock
C2	0CTS0	Channel 0 Clear To Send	0RXCLK0	Channel 0 Receive Clock
C3	1RTS0	Channel 1 Request To Send	1TXCLK0	Channel 1 Transmit Clock
C4	1CTS0	Channel 1 Clear To Send	1RXCLK0	Channel 1 Receive Clock
C5	2RTS0	Channel 2 Request To Send	2TXCLK0	Channel 2 Transmit Clock
C6	2CTS0	Channel 2 Clear To Send	2RXCLK0	Channel 2 Receive Clock
C7	3RTS0	Channel 3 Request To Send	3TXCLK0	Channel 3 Transmit Clock
C8	3CTS0	Channel 3 Clear To Send	3RXCLK0	Channel 3 Receive Clock
C9	4RTS0	Channel 4 Request To Send	4TXCLK0	Channel 4 Transmit Clock
C10	4CTS0	Channel 4 Clear To Send	4RXCLK0	Channel 4 Receive Clock
C11	5RTS0	Channel 5 Request To Send	5TXCLK0	Channel 5 Transmit Clock
C12	5CTS0	Channel 5 Clear To Send	5RXCLK0	Channel 5 Receive Clock
C13	6RTS0	Channel 6 Request To Send	6TXCLK0	Channel 6 Transmit Clock
C14	6CTS0	Channel 6 Clear To Send	6RXCLK0	Channel 6 Receive Clock
C15	7RTS0	Channel 7 Request To Send	7TXCLK0	Channel 7 Transmit Clock
C16	7CTS0	Channel 7 Clear To Send	7RXCLK0	Channel 7 Receive Clock
C17	8RTS0	Channel 8 Request To Send	8TXCLK0	Channel 8 Transmit Clock
C18	8CTS0	Channel 8 Clear To Send	8RXCLK0	Channel 8 Receive Clock
C19	9RTS0	Channel 9 Request To Send	9TXCLK0	Channel 9 Transmit Clock
C20	9CTS0	Channel 9 Clear To Send	9RXCLK0	Channel 9 Receive Clock
C21	ARTS0	Channel A Request To Send	ATXCLK0	Channel A Transmit Clock
C22	ACTS0	Channel A Clear To Send	ARXCLK0	Channel A Receive Clock
C23	BRTS0	Channel B Request To Send	BTXCLK0	Channel B Transmit Clock
C24	BCTS0	Channel B Clear To Send	BRXCLK0	Channel B Receive Clock
C25	CRTS0	Channel C Request To Send	CTXCLK0	Channel C Transmit Clock
C26	CCTS0	Channel C Clear To Send	CRXCLK0	Channel C Receive Clock
C27	DRTS0	Channel D Request To Send	DTXCLK0	Channel D Transmit Clock
C28	DCTS0	Channel D Clear To Send	DRXCLK0	Channel D Receive Clock
C29	ERTS0	Channel E Request To Send	ETXCLK0	Channel E Transmit Clock
C30	ECTS0	Channel E Clear To Send	ERXCLK0	Channel E Receive Clock
C31	FRTS0	Channel F Request To Send	FTXCLK0	Channel F Transmit Clock
C32	FCTS0	Channel F Clear To Send	FRXCLK0	Channel F Receive Clock

A.7 Loopback Connectors

If you wish to run the loopback tests using the External Loopback Diagnostic command (see Chapter 2 - *User Applications*), you will need loopback connectors. The information in this section is for those who wish to fabricate their own. The signals connect to each local port, for example, TXD for port 0 connects to RXD for port 0.

RS-232C and MIL-188-C Loopback Connections

RS-232C/MIL-188-C Loopback Connections	
Signal	**Connects To**
TXD	RXD
TXCLK	RXCLK
RTS	CTS and RING
DTR	DSR and CD

RS-422 Loopback Connections

RS-422 Loopback Connections	
Signal	**Connects To**
TXD	RXD
TXCLK	RXCLK
RTS	CTS

RS-485 Loopback Connections

RS-485 Loopback Connections	
Signal	**Connects To**
TXD	RXD
TXCLK	RXCLK

A.8 Specifications

Specifications	
VMEbus	Compliant with the VMEbus revision C.1 specification IEC 821 & IEEE P1014/01.2 Supports both Master & Slave modes.
Master Data Transfer	A32/A24/A16 - D32/D08. Optional D64 support. Programmable address modifiers. UAT/BLT/RMW. DMA transfer support with scatter/gather.
Slave Data Transfer	A32/A24/A16 - D32/D16/D08. Optional D64 support. Short Supervisory (2D). Standard Supervisory (3D, 3E, 3F). Extended Supervisory (0D, 0E, 0F). Short non-privileged (29). Standard non-privileged (39, 3A, 3B). Extended non-privileged (09, 0A, 0B).
Bus Request Lines	BR(0) through BR(3). Programmable selection of all lines.
Interrupt Request	IRQ1 through IRQ7. Programmable selection of all lines.
P2 Serial Interface	Optional full serial interface available via unused pins on rows A & C of the P2 connector.
Dimensions	6U Dual-height EuroCard (160mm x 233mm). Front Panel: Single-width.
Connectors	Front Panel: Two 100-pin high-density.
Status LED	Tri-color LED used to indicate overall board status.
Power	5 VDC @ 5 Amps (maximum) - 75 BTU/hr. +12 VDC @ .25 Amps (maximum) -12 VDC @ .10 Amps (maximum)
Temperature	0” to 50”C (32” to 120” F) operating -40” to 68” C (0” to 150” F) storage.
Humidity	10% to 95% non-condensing

This page has no technical content.