DECmpp 12000/Sx Model 100

Parallel VME Reference Manual

Part Number: EK–DECAB–PM. B01

September 1992

This document describes the DECmpp 12000/Sx Parallel VME PCB option (KF100–AA), its installation, and configuration. It also describes the Parallel VME 6U Adapter PCB, its installation, and configuration.

Revision/Update Information:	This document has been revised for DECmpp Version 1.1.
Operating System and Version:	ULTRIX Version 4.2A. Future releases may require higher versions.
Software Version:	DECmpp 12000/Sx Version 1.1.

Digital Equipment Corporation
Maynard, Massachusetts

First Printing, January 1992
Revised, September 1992

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

This document was prepared using VAX DOCUMENT, Version 2.0.

Contents

Figures

Tables

Preface

The DECmpp 12000/Sx Parallel VME (PVME) PCB serves as a buffer between the processing elements of the Data Parallel Unit (DPU) and a VME device. The software routines that allow access to the VMEbus are described in the *DECmpp 12000/Sx Programming Language User's Guide* and the *DECmpp 12000/Sx Programming Language Reference Manual*.

Intended Audience

This guide is for use by Digital Services personnel and self-maintenance customers who install and service DECmpp 12000/Sx systems. It is also a reference for DECmpp 12000/Sx system programmers.

If necessary, refer to the *DECmpp 12000/Sx Hardware Installation Guide* for site preparation information.

Required Tools

The tools required to install the PVME and 6U adapter are:

- Scissors or knife for opening packages
- Conventional flat blade screwdriver
- Allen wrench, 1/4-inch, to open cabinet doors
- Pliers; a thin, long-nose to add or remove jumpers

Document Structure

This manual contains two chapters and two appendixes:

- Chapter 1 describes the physical DECmpp 12000/Sx Parallel VME (PVME) PCB and its installation and diagnostic procedures and explains the PCB's internal registers.
- Chapter 2 describes the DECmpp 12000/Sx Parallel VME 6U Adapter and how to install it.
- Appendix A explains the PVME address maps.
- Appendix B describes the swap mode definitions.

Related Documents

Table 1 lists documents that provide additional information about the DECmpp 12000/Sx system.

Table 1 Related Documents

Document Title	Order Number
DECmpp 12000/Sx Parallel Disk Array Reference Manual	EK-DECAB-RM
DECmpp 12000/Sx Programming Language User's Guide	AA-PMAUB-TE
DECmpp 12000/Sx Programming Language Reference Manual	AA-PMAVB-TE
DECmpp 12000/Sx Architecture Specification	AA-PMASB-TE
DWTVX-Ax VME I/O Subsystem Pocket Service Guide	EK-DWTVX-PS
T6000 Module Installation/Owner's Card	EK-T6000-IN
DECmpp 12000/Sx Hardware Service Manual	EK-DECAC-SM

Conventions

The following conventions are used throughout the DECmpp 12000/Sx documentation set.

Convention	Meaning
Return	In examples, a key name shown within a box indicates that you press a key on the keyboard. In text, a key name is not enclosed in a box but is printed with an initial capital letter, like Return.
Ctrl/*x*	A key combination, shown with a slash separating the two key names, indicates that you hold down the first key while you press the second key.
MB1, MB2, MB3	The buttons on a mouse. MB1 is the left button, MB2 is the center button, and MB3 is the right button of a mouse whose button arrangement is right-handed. It is possible to redefine the mouse buttons.
%	A percent sign (%) represents the default user prompt for your system.
#	A number sign (#) represents the default superuser (root) prompt for your system.
. . .	In examples, a horizontal series of dots, or ellipsis, indicates that additional parameters, values, or other information can be entered.
⋮	In examples, a vertical series of dots, or vertical ellipsis, indicates that a portion of the example is intentionally omitted.
[ ]	In syntax descriptions and functional descriptions, brackets indicate optional items.
`dpumanager(6)`	Cross-references to the *ULTRIX Reference Pages*, which include the appropriate section number in parentheses.
italicized text	In examples, italicized text denotes parameters, values, or other information that will change from either session to session or user to user. In text, italicized words or phrases are used to add emphasis to important words, concepts, or titles of manuals.
`ULTRIX keywords`	This typeface is used to indicate system output or the exact name of a command, option, partition, pathname, directory, or file.
`Code examples`	This typeface is used to display program coding examples.
UPPERCASE and lowercase strings	The ULTRIX system differentiates between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function descriptions must be entered exactly as shown.

1
Parallel VME

The DECmpp 12000/Sx Parallel VME (PVME) option (KF100–AA) serves as a buffer between processing elements (PEs) and a VME device. The PVME is a peripheral to the front-end server or to other I/O processors in the DECmpp 12000/Sx or DECmpp 12000–LC/Sx systems. It is connected to these other processors by the Massively Parallel VMEbus (MPVMEbus).

1.1 IORAM Capabilities

The PVME has an input/output random access memory (IORAM) buffer of 8 MB of parity-checked memory. Its performance is as follows:

- For PVME only block-mode accesses to IORAM, peak VME performance is:
  - 13.5 MB/s for I/O write
  - 11.5 MB/s for I/O read
- For router-only interface accesses to IORAM, peak router transfer rates into IORAM are:
  - 50 MB/s for I/O write
  - 60 MB/s for I/O read
- For VME if constant router and VME interface accesses to the IORAM— 8.0 MB/s.

The IORAM is available on the MPVMEbus as either a bus master or a bus slave. Master operation is combined with a DMA controller to provide normal single-transfer bus operations and VME block-mode operations. The IORAM buffers data transfers between the PE array and the MPVMEbus.

The MPVMEbus port to the IORAM is capable of D32, D16, and D8 transfers. VME access to the IORAM includes a byte-swapping feature to provide compatibility with little- and big-endian VME devices for byte, word, and longword object sizes.

1.2 PVME Features

The PVME has the following features:

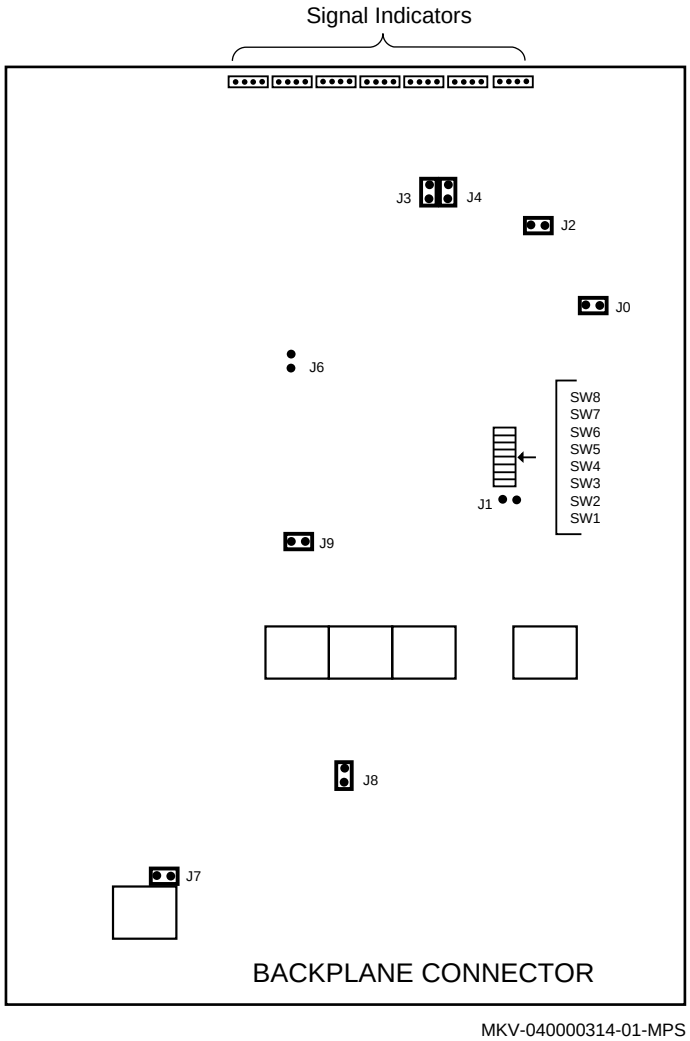
- 8 MB of parity-checked RAM for use as an I/O buffer
- IORAM and all registers fully memory-mapped into MPVMEbus address space
- MPVMEbus interrupts for VME DMA transaction completion and error conditions
- VME I/F supports both A24 and A32 addressing modes as both master and slave

- VME block-mode access supported for all PVME addresses as slave; (block mode as master-only, supported in conjunction with IORAM DMA controller)
- D32 and D16 transfers supported to all control registers, as well as D32, D16, and D8 access to IORAM
- Four modes of byte-swapping supported by IORAM for compatibility with little- and big-endian VME devices

1.3 Hardware Description

Figure 1–1 shows the DECmpp 12000/Sx Parallel VME PCB, including switches and jumpers. Table 1–1 lists factory settings for the PVME PCB switches and jumpers.

Figure 1–1 Parallel VME PCB Diagram



MKV-040000314-01-MPS

Table 1–1 PVME Switch/Jumper Factory Settings

Switch/Jumper	Factory Setting	Function
SW8	Off	Base address modifier (A32)
SW7	Off	Base address modifier (A32)
SW6	On	Base address modifier (A32)
SW5	Off	Base address modifier (A32)
SW4	Off	Base address modifier (A24)
SW3	Off	Base address modifier (A24)
SW2	On	Base address modifier (A24)
SW1	Off	Base address modifier (A24)
J0	In	
J1	Out	
J2	In	Interrupt priority request level
J3	In	
J4	In	
J6	Out	
J7	In	
J8	In	
J9	In	

1.3.1 PVME Switches

The PVME independently supports base address selection for A24 and A32 address modes. In both cases, the base address of the PCB is selected by four DIP switches that set the four most significant address bits within each address space.

Note

A32 and A24 addressing modes both support VME block-mode and data-mode address modifiers, for both supervisor and nonprivileged accesses.

The address modifier codes are:

- 0x3F — A24 Supervisory block transfer
- 0x3D — A24 Supervisory data access
- 0x3B — A24 Nonprivileged block transfer
- 0x39 — A24 Nonprivileged data access
- 0x0F — A32 Supervisory block transfer
- 0x0D — A32 Supervisory data access
- 0x0B — A32 Nonprivileged block transfer
- 0x09 — A32 Nonprivileged data access

1.3.1.1 A32 Addressing Mode

DIP switches SW8 to SW5 select the base address that the PVME responds to when addressed with supported A32 address modifiers, as shown in Table 1–2. (Factory-set address is D000 0000.)

Table 1–2 A32 Addressing Mode

SW8	SW7	SW6	SW5	Base Address (in HEX)
On	On	On	On	0000 0000
On	On	On	Off	1000 0000
On	On	Off	On	2000 0000
On	On	Off	Off	3000 0000
On	Off	On	On	4000 0000
On	Off	On	Off	5000 0000
On	Off	Off	On	6000 0000
On	Off	Off	Off	7000 0000
Off	On	On	On	8000 0000
Off	On	On	Off	9000 0000
Off	On	Off	On	A000 0000
Off	On	Off	Off	B000 0000
Off	Off	On	On	C000 0000
Off	Off	On	Off	D000 0000
Off	Off	Off	On	E000 0000
Off	Off	Off	Off	F000 0000

1.3.1.2 A24 Addressing Mode

DIP switches SW4 to SW1 select the base address that the PVME responds to when addressed with supported A24 address modifiers, as shown in Table 1–3. (Factory-set address is xxD0 0000.)

Table 1–3 A24 Addressing Mode

SW4	SW3	SW2	SW1	Base Address (in HEX)
On	On	On	On	xx00 0000
On	On	On	Off	xx10 0000
On	On	Off	On	xx20 0000
On	On	Off	Off	xx30 0000
On	Off	On	On	xx40 0000
On	Off	On	Off	xx50 0000
On	Off	Off	On	xx60 0000
On	Off	Off	Off	xx70 0000
Off	On	On	On	xx80 0000
Off	On	On	Off	xx90 0000

(continued on next page)

Table 1–3 (Cont.) A24 Addressing Mode

SW4	SW3	SW2	SW1	Base Address (in HEX)
Off	On	Off	On	xxA0 0000
Off	On	Off	Off	xxB0 0000
Off	Off	On	On	xxC0 0000
Off	Off	On	Off	xxD0 0000
Off	Off	Off	On	xxE0 0000
Off	Off	Off	Off	xxF0 0000

1.3.2 PVME Jumpers

Refer to Figure 1–1 for installed jumper locations. With the exception of J2 described below, all jumpers are factory set and should not be modified (Table 1–1).

The PVME has two types of interrupts:

- PVME error conditions — asynchronous interrupts
- DMA controller completion — synchronous interrupts

Error interrupts request service on either MPVMEbus priority 5 or 6, as selected by jumper J2 shown below. The factory setting is level 5 (jumper installed).

J2	Interrupt Request Priority Level
In	5
Out	6

All asynchronous interrupts use the same interrupt vector, ERVEC<7:0>. The value for ERVEC<7:0> is derived by prepending IVBASE<1:0> (from OPCSR) with 6 bits of 0:

```
ERVEC = 000000 IVBAS<1> IVBAS<0>
```

Refer to Section 1.6.2 for a description of the OPCSR register.

Note

The asynchronous interrupts share the interrupt vector space with the programmable, synchronous interrupts. Indeterminate operation can occur if a synchronous interrupt uses a vector level of 0.

1.3.3 PVME Signal Indicators

The PVME PCB has 28 signal indicators along its outside edge, as shown in Figure 1–1. Figure 1–2 shows a detail of the indicator bank and the associated signal name. From the top of the PCB (as installed), Table 1–4 provides the signal name and the signal abbreviation used on the PCB.

When the indicator is on (green), the signal is asserted.

Figure 1–2 Parallel VME Signal Indicators

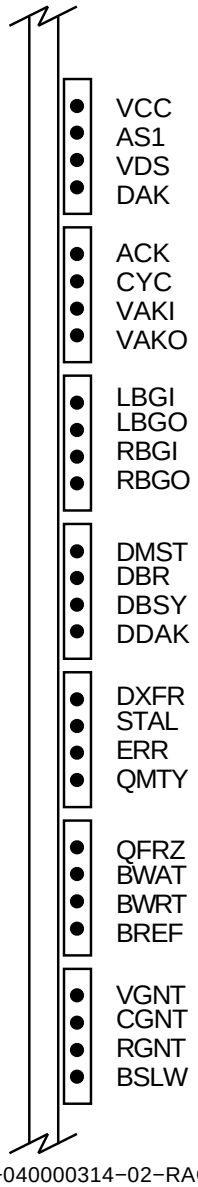


Table 1–4 Parallel VME Signal Indicator Descriptions

Signal	Abbreviation	Description
Vcc	VCC	ON: +5 V power supply is on
AS1_	AS1	ON: MPVMEbus AS (address strobe) is true
vmeDS	VDS	ON: MPVMEbus DS (data strobe) is true
DTACK_	DAK	ON: IOCTLR generated DTACK is true
bIACK	ACK	ON: MPVMEbus IACK (interrupt acknowledge) is true
IACKCYC	CYC	ON: IOCTLR is responding to an IACK MPVMEbus cycle
vmeIACKIN_	VAKI	ON: MPVMEbus interrupt acknowledge daisy chain-in is true
vmeIACKOUT_	VAKO	ON: MPVMEbus interrupt acknowledge daisy chain-out is true
vmeLBGIN_	LBGI	ON: Local MPVMEbus grant daisy chain-in is true
dmaLBBGOUT_	LBGO	ON: Local MPVMEbus grant daisy chain-out is true
vmeRBGIN_	RBGI	ON: Remote MPVMEbus grant daisy chain-in is true
dmaRBGOUT_	RBGO	ON: Remote MPVMEbus grant daisy chain-out is true
dmaMASTER	DMST	ON: IOCTLR is dma bus master
dmaBR_	DBR	ON: IOCTLR is asserting request for MPVMEbus
dmaBBSY_	DBSY	ON: IOCTLR is asserting MPVMEbus Busy
vmeDTACK_	DDAK	ON: MPVMEbus DTACK is true
chDXFR_	DXFR	N/A — May be ON or OFF (value is X)[1]
chSTALL_	STAL	N/A — May be ON or OFF (value is X)
chERR	ERR	N/A — May be ON or OFF (value is X)
queueEMPTY_	QMTY	N/A — May be ON or OFF (value is X)
queueFROZ	QFRZ	N/A — May be ON or OFF (value is X)
babWAIT_	BWAT	ON: IORAM is holding off access; address has crossed page boundary
babWRT_	BWRT	ON: IORAM is performing a write cycle
refGNT_	BREF	ON: IORAM is performing a refresh cycle
vmeGRANT_	VGNT	ON: VME Interface has been granted IORAM access
chGRANT_	CGNT	ON: N/A — may be on or off (value is X)
rioGRANT_	RGNT	ON: RIO Interface has been granted IORAM access
babSLOWDEV_	BSLW	ON: Slow VME device has accessed IORAM

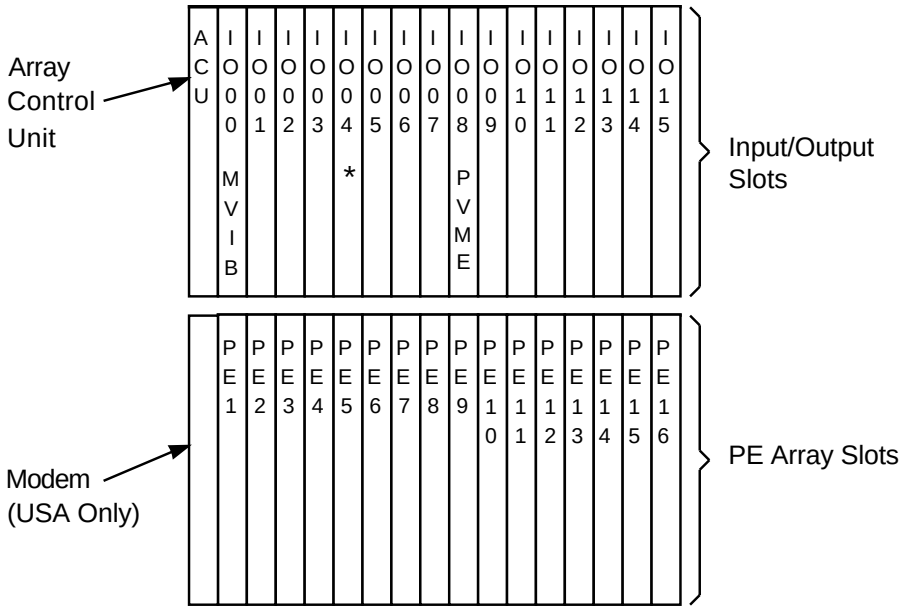
[1] N/A = not applicable to PVME

1.4 PVME Installation

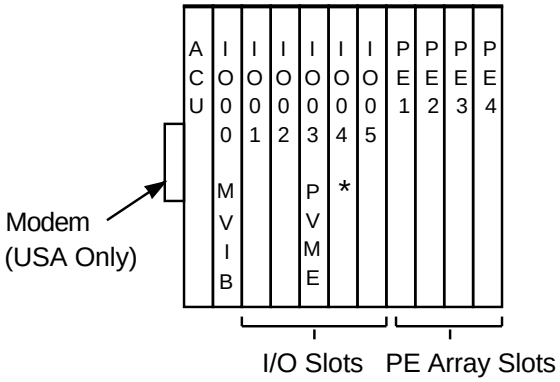
Follow the steps below to install the PVME PCB in the data parallel unit (DPU):

1. Shut down the system, following the procedures in Chapter 1 of the *DECmpp 12000/Sx Hardware Service Manual*.
2. Open the DPU front door and turn the DPU keyswitch to OFF.
3. Open the DPU rear door and turn the circuit breaker OFF.
4. Referring to Figure 1–3, identify the I/O slot in which you insert the PVME PCB. For the DECmpp 12000 DPU, this is slot IO08 (tenth physical slot from left). For the DECmpp 12000–LC DPU, this is slot IO03 (fifth physical slot from left).
5. Install the PVME PCB by sliding it into the rails for the appropriate card cage slot. Seat it firmly in the backplane with the PCB extraction levers.
6. Close and latch the DPU back door.
7. Open the DPU inner front door (–LC configuration only).
8. Remove the clear plastic backplane protector (DECmpp 12000/Sx only).
9. Referring to Figure 1–4, remove the three jumpers above backplane slot IO08 (DECmpp 12000/Sx) or slot IO05 (DECmpp 12000–LC/Sx).
10. Replace the clear plastic backplane protector (DECmpp 12000/Sx only).
11. Close the inner front door (–LC only).
12. Turn the circuit breaker on.
13. Set the DPU keyswitch to ON.
14. Close the DPU front and rear doors.
15. Turn on and boot the server.
16. Run the PVME confidence test as described in Section 1.5.

Figure 1–3 DPU Card Cage Slots



DECmpp 12000
Card Cage (Rear View)



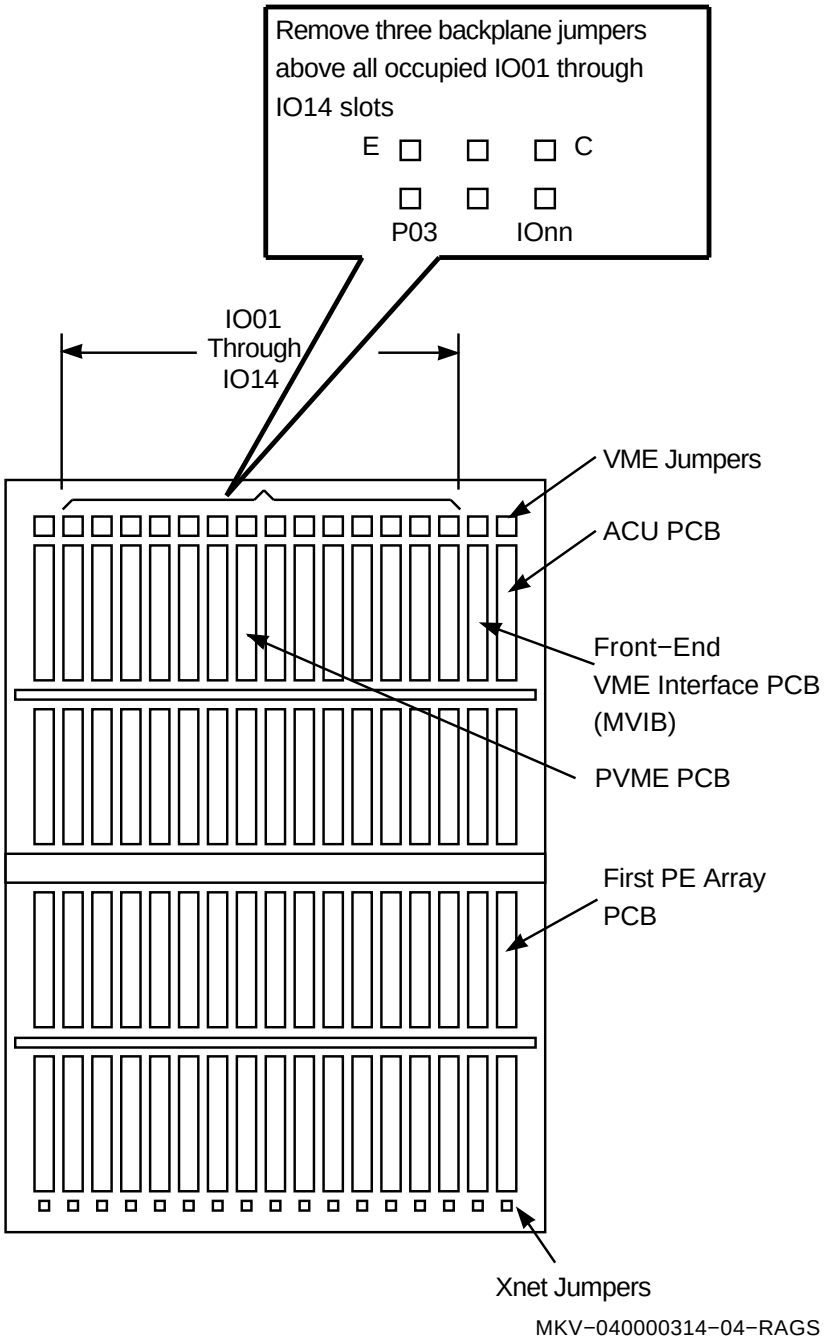
DECmpp 12000−LC
Card Cage (Rear View)

* Slot IO04 Reserved for Optional PDA Interface

MKV−040000314−03−RAGS

1.4 PVME Installation

Figure 1–4 DPU Backplane I/O Jumpers



1.5 Diagnostics

The diagnostics for the PVME PCB test the PCB RAM's addressing, data, and access modes. The diagnostics also test the data transfers from the front-end system to the RAM on the PVME PCB. The six byte-swapping modes providing access to the IORAM through the MPVMEbus are also tested for correct functioning.

The test results are appended to a LOG file in the directory in which the test is run. (If no LOG file exists, one is created.)

To test the PVME PCB, log in as the superuser. The dpumanager should not be running any other jobs. At the system prompt, enter:

```
# pvme_diag [-v]
```

The command line option `-v` displays all test messages. The default is a terse mode, displaying only essential messages (pass, fail, and important error messages).

If available at your site, you can also check the VME memory access paths. In response to the system display, you can specify the VME memory device address or use the default address of 0xe8000000.

You should avoid interrupting the diagnostic test while it is running.

1.6 Register Fields

This section describes control/status (CSR) registers, interrupt registers, and a register used for diagnostics.

1.6.1 Reserved and Spare Addresses and Bits

Spare bits and addresses in the address map correspond to physical hardware that currently has no defined function. Spare bits usually can be set to 1 or 0. Spare registers respond to a MPVMEbus access. Writing a spare register has no effect. Reading a spare register returns 0.

Reserved bits and registers are blocked out for implementation reasons. Accessing a reserved bit has undefined results (there are no reserved bits on the PVME). Accessing a reserved register causes a MPVMEbus Bus Error (BERR).

Unused bits have no defined function and do not correspond to any physical hardware. Reading an unused bit returns an undefined value; writing to one has no effect.

Undefined registers have no meaning to the PVME. These are available to other MPVMEbus devices. The PVME does not drive the MPVMEbus Data Bus or DTACK_ in response to an access to this region.

The bit register descriptions use the following abbreviations:

- R/W — Read/Write
- R/O — Read only
- W/O — Write only
- R/C — Read/Clear

W/O and unused bits return an undefined value for read operations.

1.6.2 Parallel VME Operation CSR (OPCSR)

This is a general-purpose configuration register, shown in Figure 1–5, that affects overall Parallel VME operation. Table 1–5 defines this register.

Figure 1–5 Parallel VME Operation CSR (OPCSR)

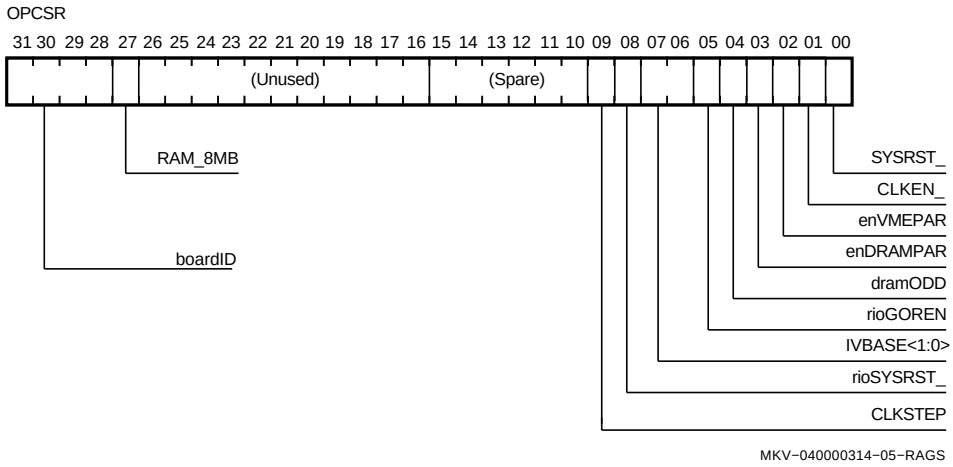


Table 1–5 Parallel VME Operation CSR

Extent	Name	Description
<31:28>	boardID	(R/O) Slot number containing the Parallel VME, derived from physical connections on the backplane.
<27>	RAM_8MB	(R/O) Defines available amount of IORAM on Parallel VME. Set to 1, 8 MBytes of IORAM. Set to 0, 32 MBytes of IORAM.
<26:16>		Unused
<15:10>		Spare
<09>	CLKSTEP	(R/W) No effect for PVME.
<08>	rioSYSRST_	(R/W) Used as Parallel VME analog of sysRst on the ACU. It affects the serial (or router) side of the RIO only. When set to 0, forces a reset to the serial side of the RIO; when set to 1, releases the reset. Resets to 0 on system reset.
<07:06>	IVBASE<1:0>	(R/W) Sets the base value for all interrupts generated by the Parallel VME. When an interrupt vector is placed on the MPVMEbus, IVBASE supplies bits 7 and 6 (corresponding to IVBASE<1> and IVBASE<0>, respectively), while the specific interrupter on the Parallel VME contributes the lower 6 bits. Resets to 0 on system reset.

(continued on next page)

Table 1–5 (Cont.) Parallel VME Operation CSR

Extent	Name	Description
<05>	rioGOREN_	(R/W) When set to 1, prevents the Parallel VME RIO from driving the I/O sum-OR tree; when set to 0, the RIO drives it. Resets to 0 on system reset.
<04>	dramOdd	(R/W) Used for diagnostics only. Specifies the type of parity used when writing to the Parallel VME IORAM. When set to 1, uses odd parity; when set to 0, uses even parity. Resets to 0 on system reset.
<03>	enDRAMPAR	(R/W) Controls the parity checking on the Parallel VME IORAM. When set to 1, enables parity checking; when set to 0, disables parity checking. Resets to 0 on system reset.
<02>	enVMEPAR	(R/W) Controls MPVMEbus address and data parity checking for both slave and master (DMA) operation. Parity generation is always enabled and is not affected by this bit. When set to 1, enables parity checking; when set to 0, disables parity checking. Resets to 0 on system reset.
<01>	CLKEN_	(R/W) No effect for PVME.
<00>	SYSRST_	(R/W) When set to 0, all non-VME functions are held in a reset state. This forces reset to the IORAM and the parallel side of the RIO. Setting this bit to 1 permits further operation. Resets to 0 on system reset.

1.6.3 Parallel VME State CSR (OPSTATE)

This is a general-purpose control/status register, shown in Figure 1–6, that affects overall Parallel VME operation; Table 1–6 defines this register.

Figure 1–6 Parallel VME State CSR (OPSTATE)

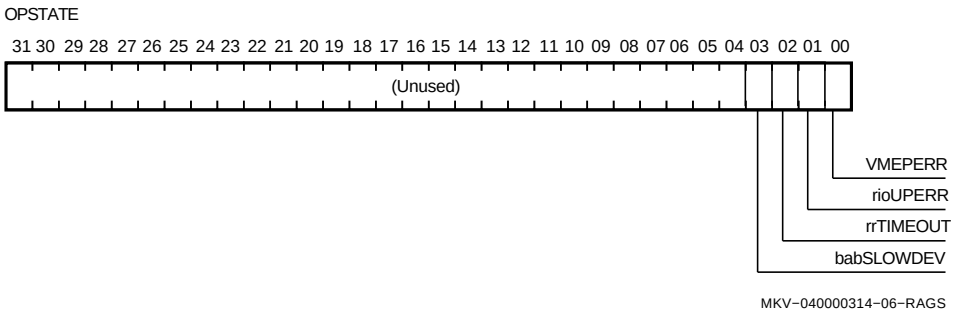


Table 1–6 Parallel VME State CSR

Extent	Name	Description
<31:04>		Unused
<03>	babSLOWDEV	(R/C) Used for advisory only. When set to 1, the IORAM bank access bus (BAB) control has detected a device (typically the VME) that has held the BAB for an excessive length of time. Once set, this bit is held until cleared by writing a 1 to it. Resets to 0 on system reset.
<02>	rrTIMEOUT	(R/C) When read as 1, indicates that a MPVMEbus access to IORAM has exceeded 256 35-ns clocks due to a hardware failure by the IORAM bank access bus (BAB) control. If a DMA/Bus Master operation is in progress, the DMA aborts, and status is reflected in IODMACSR. If rrTIMEOUT occurs during a slave operation, the MPVMEbus eventually times out, and BERR (IODMACSR, bit 11) is asserted, indicating that no slave responded to the address cycle. Writing 1 to this bit clears it. Writing 0 has no effect. Resets to 1 on system reset.
<01>	rioUPERR	(R/C) When read as 1, the RIO chip detected an ACU microcode parity error. The RIO interface remains frozen until this bit is cleared. Writing 1 to this bit clears it. Writing 0 has no effect. Resets to 0 on system reset.
<00>	VMEPERR	(R/C) When read as 1, the Parallel VME generated a MPVMEbus system bus parity error (Bi_VSysParErr_). It remains set until a 1 is written to it. Resets to 0 on system reset.

1.6.4 IORAM DMA (IODMACSR)

This register, shown in Figure 1–7, initiates and terminates the DMA operation, as well as gives completion status. Writing 1 to the GO bit (bit 0) resets all status bits (bits 15–10) to 0. Table 1–7 defines this register.

Figure 1–7 Parallel VME IORAM DMA Register (IODMACSR)

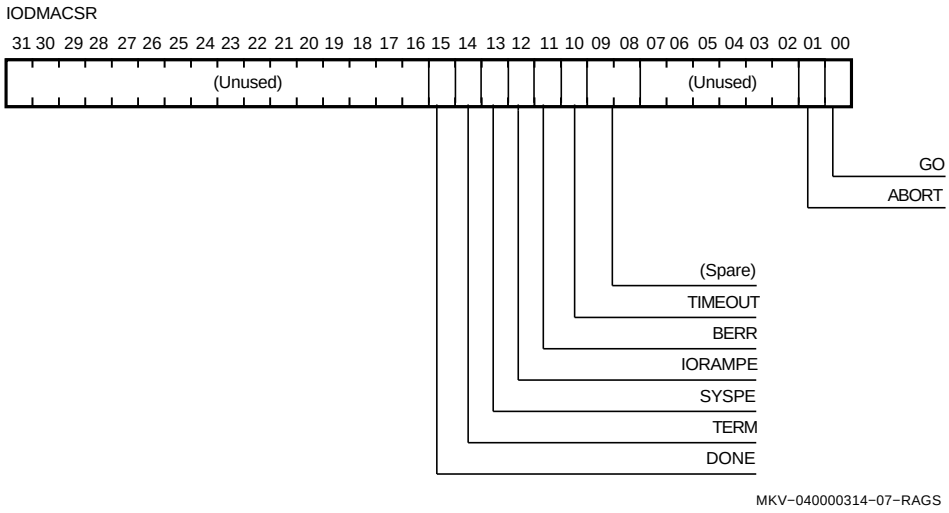


Table 1–7 IORAM DMA

Extent	Name	Description
31:16>		Unused
<15>	DONE	(R/O) When 1, the requested DMA operation has completed. When 0, a DMA is in progress. Resets to 1 on system reset.
<14>	TERM	(R/O) When DONE (bit 15) is true (reads as 1), the requested DMA operation was terminated prematurely by software request (for example, the ABORT bit was set at 1). Reads as 1 when true. Resets to 0 on system reset.
<13>	SYSPE	(R/O) When DONE (bit 15) is true (reads as 1), a MPVMEbus system bus parity error terminated the requested DMA operation prematurely. Reads as 1 when true. Resets to 0 on system reset.
<12>	IORAMPE	(R/O) When DONE (bit 15) is true (reads as 1), a parity error was detected on the Parallel VME IORAM, and the requested DMA operation was terminated prematurely. Reads as 1 when true. Resets to 0 on system reset.
<11>	BERR	(R/O) When DONE (bit 15) is true (reads as 1), indicates that a MPVMEbus Bus Error terminated the requested DMA operation prematurely. Reads as 1 when true. Resets to 0 on system reset.
<10>	TIMEOUT	(R/O) When DONE (bit 15) is true (reads as 1), indicates that the DMA operation was aborted because the DMA controller could not access IORAM within 256 35-ns clocks. This indicates a hardware failure in the IORAM BAB Control. Reads as 1 when true. Resets to 0 on system reset.
<09:08>		Spare
<07:02>		Unused
<01>	ABORT	(W/O) When set to 1, aborts any DMA operations in progress. Reads as 1 when true.
<00>	GO	(W/O) When set to 1, initiates a DMA operation. Reads as 1 when true.

1.6.5 DMA Configuration Control (IODMACONF)

This register, shown in Figure 1–8, sets the operational configuration of the address and byte count registers: as byte, 16-bit short-word, or 32-bit long-word counters. (It must be set before loading the address or byte-count counters.) Resets to 0 on system reset. Table 1–8 defines this register.

Figure 1–8 DMA Configuration Control Register (IODMACONF)

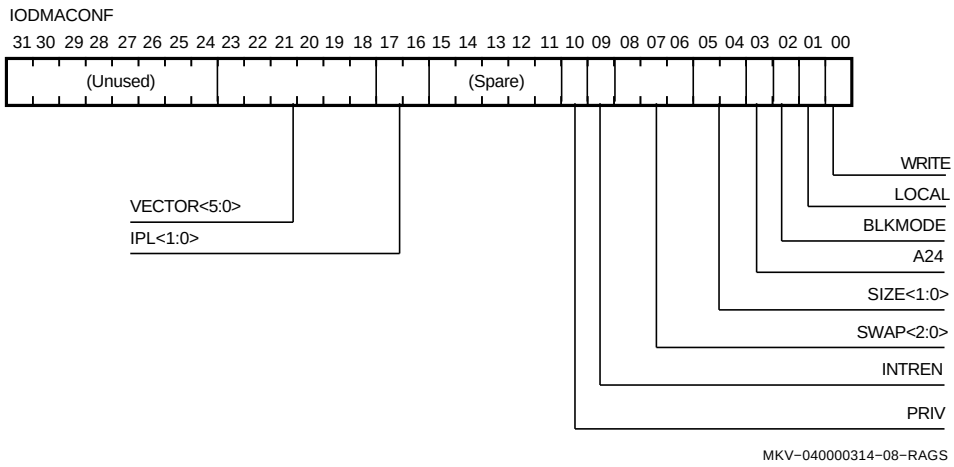


Table 1–8 DMA Configuration Control Register

Extent	Name	Description
<31:24>		Unused
<23:18>	VECTOR<5:0>	(R/W) Appending this value to IVBASE<1:0> from OPCSR creates the DMA DONE interrupt vector. If an interrupt is requested on any DMA completion, the DMA DONE vector is driven onto the VME data bus in response to the interrupt acknowledge.
<17:16>	IPL<1:0>	(R/W) Specifies the MPVMEbus interrupt request level used for a DMA DONE interrupt: • 0 — 1 • 1 — 2 • 2 — 3 • 3 — 4
<15:11>		Spare

(continued on next page)

Table 1–8 (Cont.) DMA Configuration Control Register

Extent	Name	Description
<10>	PRIV	(R/W) Controls the Address Modifier used during a DMA operation. When set to 1, indicates supervisor mode; when set to 0, indicates user mode.
<09>	INTREN	(R/W) When set to 1, on completion, generates an interrupt at the level and vector specified, including error and abort conditions; when set to 0, inhibits interrupt generation.
<08:06>	SWAP<2:0>	(R/W) This field selects the byte-swapping mode used for DMAs. • 0 — Swap Mode 0 • 1 — Swap Mode 1 • 2 — Swap Mode 2 • 3 — Swap Mode 3 • 4 — Swap Mode 4 • 5 — Swap Mode 5 • 6 — (Reserved) • 7 — (Reserved)
<05:04>	SIZE<1:0>	(R/W) This field sets the MPVMEbus transfer size: • 0 — Byte • 1 — 16-bit short word • 2 — 32-bit long word • 3 — (Reserved)
<03>	A24	(R/W) When set to 1, the DMA uses A24 MPVMEbus addressing; when set to 0, uses A32 addressing. In both cases, uses the address modifier for DATA ACCESS.
<02>	BLKMODE	(R/W) When set to 1, the DMA uses the MPVMEbus block mode protocol; when set to 0, the DMA uses the normal address /data mode.
<01>	LOCAL	(R/W) When set to 1, arbitrates for the internal DPU MPVMEbus (not the front end); when set to 0, corresponds to a remote, front-end bus.
<00>	WRITE	(R/W) When set to 1, initiates a DMA write operation; when set to 0, initiates a read operation. Write is defined as a write to the MPVMEbus from IORAM. A read is defined as a read from the MPVMEbus to IORAM.

1.6.6 Interrupt Control Registers

All interrupts are controlled by a common pair of registers:

- IRQCSR: Interrupt Request Control and Status Register
- IRQMASK: Interrupt Request Mask Register

1.6.6.1 Interrupt Request CSR (IRQCSR)

This register, shown in Figure 1–9, holds status for all interrupt sources. A MPVMEbus interrupt request is asserted when any bit in IRQCSR is asserted and its corresponding mask bit in IRQMASK is 1. All bits are reset to 0 on MPVMEbus reset. Table 1–9 defines this register.

Note

The PVME uses the MPVMEbus interrupt protocol release on register access (RORA); once an interrupt has been asserted, it remains asserted until the appropriate status bit is cleared in IRQCSR.

Figure 1–9 Interrupt Request CSR (IRQCSR)

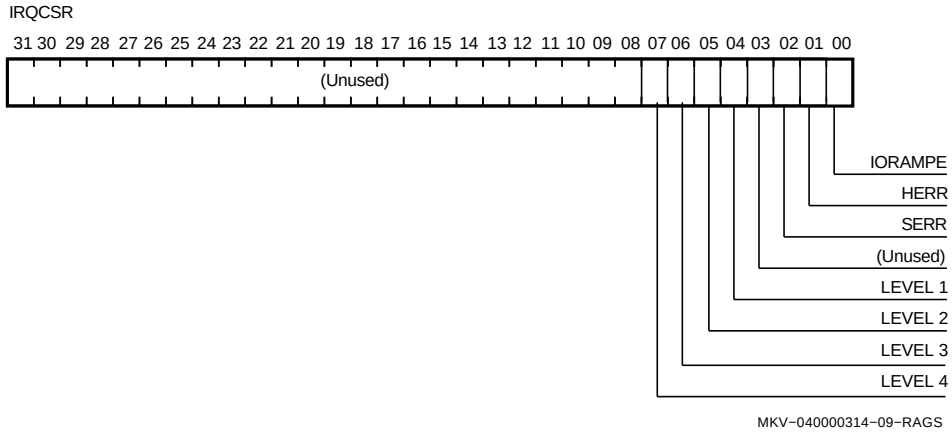


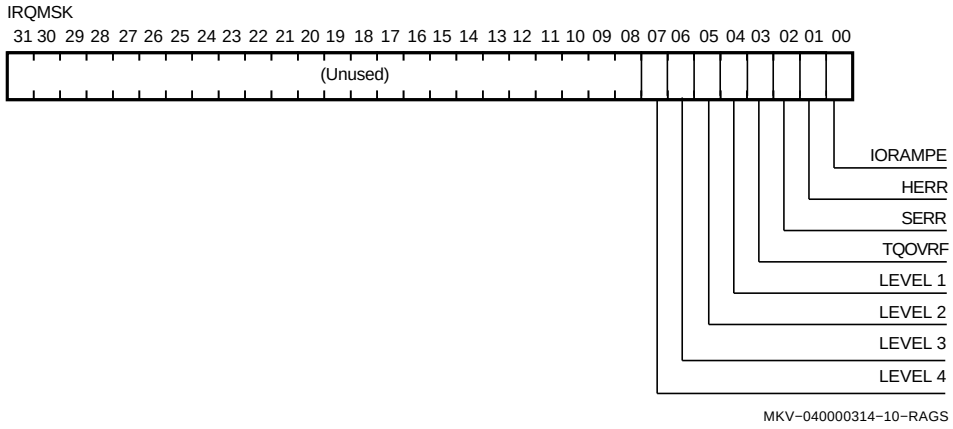
Table 1–9 Interrupt Request CSR (IRQCSR)

Extent	Name	Description
<31:08>		Unused
<07>	LEVEL4	(R/C) MPVMEbus Level 4 Interrupt Request. When set to 1, indicates a level 4 interrupt request by the DMA controller. Write 1 to clear it.
<06>	LEVEL3	(R/C) MPVMEbus Level 3 Interrupt Request. When set to 1, indicates a level 3 interrupt request by the DMA controller. Write 1 to clear it.
<05>	LEVEL2	(R/C) MPVMEbus Level 2 Interrupt Request. When set to 1, indicates a level 2 interrupt request by either the Transaction Controller or the DMA controller. Write 1 to clear it.
<04>	LEVEL1	(R/C) MPVMEbus Level 1 Interrupt Request. When set to 1, indicates a level 1 interrupt request by either the Transaction Controller or the DMA controller. Write 1 to clear it.
<03>		Unused
<02>	SERR	(R/C) Soft Error Remote IORAM. When set to 1, indicates detection of a correctable ECC error on a remote IORAM PCB during a transfer with RIO, the MPIOC, or the MPVMEbus. (This bit might be set if HERR is also set. This would not necessarily indicate that a soft error had occurred.) Write 1 to clear it.
<01>	HERR	(R/C) Hard Error Remote IORAM. When set to 1, indicates detection of an uncorrectable ECC error on a remote IORAM PCB during a transfer with RIO, the MPIOC, or the MPVMEbus. Write 1 to clear it.
<00>	IORAMPE	(R/C) Parity Error PVME IORAM. When set to 1, indicates detection of a parity error on the PVME IORAM during a transfer with RIO or the MPVMEbus. Write 1 to clear it.

1.6.6.2 Interrupt Mask Register (IRQMSK)

When set to 1, each bit in this register, shown in Figure 1–10, acts as an interrupt-enable for a corresponding status bit in IRQCSR. These mask bits override the request bits generated by the Transaction Controller and the DMA Controller. All bits are read/write and are reset to 0 by the MPVMEbus Reset.

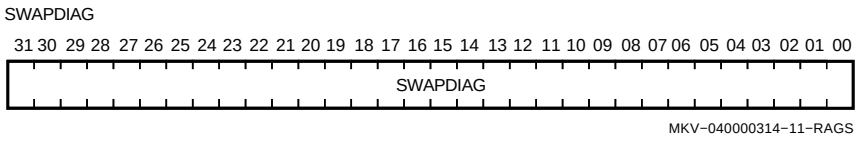
Figure 1–10 Interrupt Mask Register (IRQMSK)



1.6.7 SWAP Diagnostic CSR (SWAPDIAG)

This register, shown in Figure 1–11, is for diagnostic use only. It provides read-only access to data between IORAM and the byte swapper. It samples data on every read or write to IORAM for both slave and DMA access. For writes to IORAM, it provides the output of the byte swap transformation. On reads from IORAM, it contains the data from the last read of IORAM prior to transformation.

Figure 1–11 SWAP Diagnostic CSR (SWAPDIAG)



2

Parallel VME 6U Adapter PCB

The DECmpp 12000/Sx Parallel VME 6U adapter PCB provides hardware and interface functions that allow PCBs designed to standard VMEbus specifications to operate on the DECmpp 12000/Sx extended bus, here called the Massively Parallel VMEbus, or MPVMEbus.

The MPVMEbus is different from a standard VMEbus in two important ways:

- It generates parity on data, addresses, and address modifiers.
- It provides a jumper-selectable, nonstandard voltage on specified connector pins.

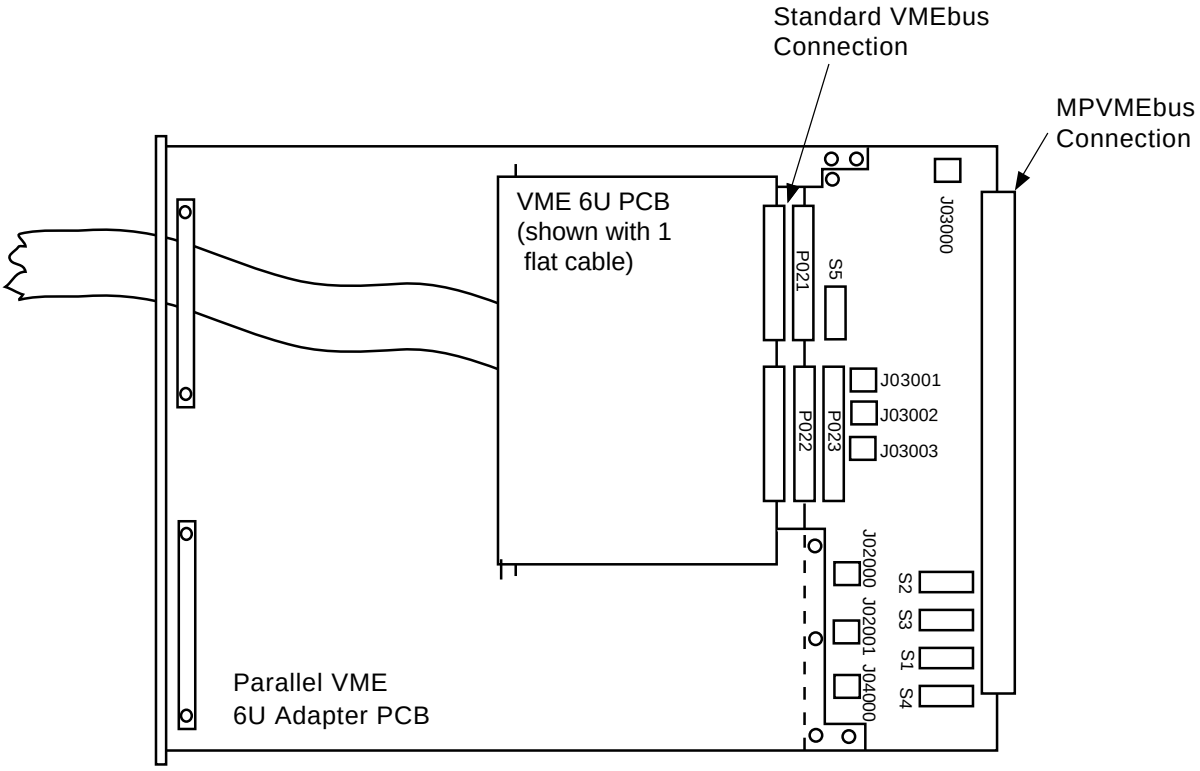
The Parallel VME 6U adapter PCB supports the MPVMEbus parity checking, and it provides a (jumper-selectable) option to output –5.2 V on specified MPVMEbus pins. A control/status register (CSR) provides control and status functions, switches on the Parallel VME 6U adapter PCB, and allows the CSR to be located at any 32-bit address in MPVMEbus space.

The Parallel VME 6U adapter PCB with the interface cable, together with the DPU enclosure, provides support for cabling between the DPU and other I/O devices. A mechanical cutout area on the Parallel VME 6U Adapter PCB front panel allows cables to leave the card cage. Removable sheet metal panels on the DPU lower rear (DECmpp 12000–LC configuration only) can be fitted with connectors and an external connector-to-connector. This combination supports cabling between the standard VMEbus 6U PCB and the exterior of the DPU.

2.1 Hardware Description

Figure 2–1 shows the Parallel VME 6U adapter PCB, including all switches and jumpers. Figure 2–2 shows the standard settings for the jumpers on the adapter PCB.

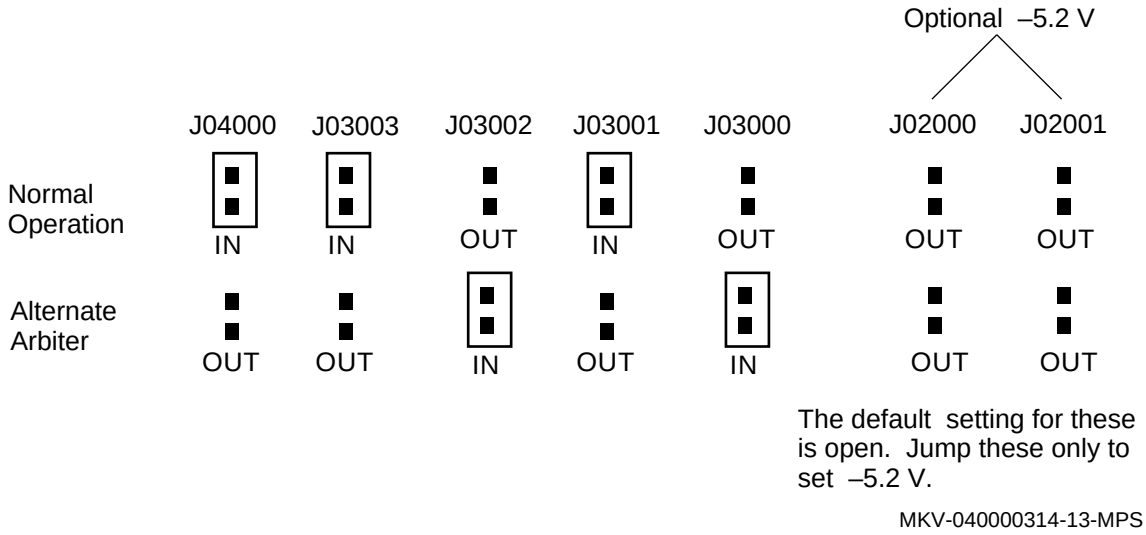
Figure 2–1 Parallel VME 6U Adapter Hardware



Note that J-numbers are located on the back of the board.

MKV-040000314-12-MPS

Figure 2–2 Parallel VME 6U Adapter Jumper Settings



2.1.1 Switches and Jumpers

Switch packs S1 through S4 set the CSR address to any 32-bit location in MPVMEbus space (ON = 1 and OFF = 0). Not used in setting the address, the last two switches of S1 are not required. Table 2–1 shows how the switches are assigned.

S1, switch 1, enables a local 16-MHz oscillator when it is ON. S1, switch 2, is unused. S5 switches enable interrupts when ON. Table 2–2 shows the switch settings for enabling interrupts.

Jumpers J02000 and J02001 connect optional –5.2 V from the PCB onto P022 and P023, rows A and C. The default for these is OFF.

Table 2–1 Parallel VME Adapter PCB CSR Address Selection

Switch Pack	Switch Number	Function	Switch Pack	Switch Number	Function
S4	8	A31	S2	8	A15
S4	7	A30	S2	7	A14
S4	6	A29	S2	6	A13
S4	5	A28	S2	5	A12
S4	4	A27	S2	4	A11
S4	3	A26	S2	3	A10
S4	2	A25	S2	2	A9
S4	1	A24	S2	1	A8
S3	8	A23	S1	8	A7
S3	7	A22	S1	7	A6
S3	6	A21	S1	6	A5
S3	5	A20	S1	5	A4
S3	4	A19	S1	4	A3
S3	3	A18	S1	3	A2
S3	2	A17	S1	2	Unused
S3	1	A16	S1	1	Oscillator

Table 2–2 Switch Settings for Interrupt Enabling

Switch Pack	Switch Number	Function
S5	8	Unused
S5	7	Interrupt 7
S5	6	Interrupt 6
S5	5	Interrupt 5
S5	4	Interrupt 4
S5	3	Interrupt 3
S5	2	Interrupt 2
S5	1	Interrupt 1

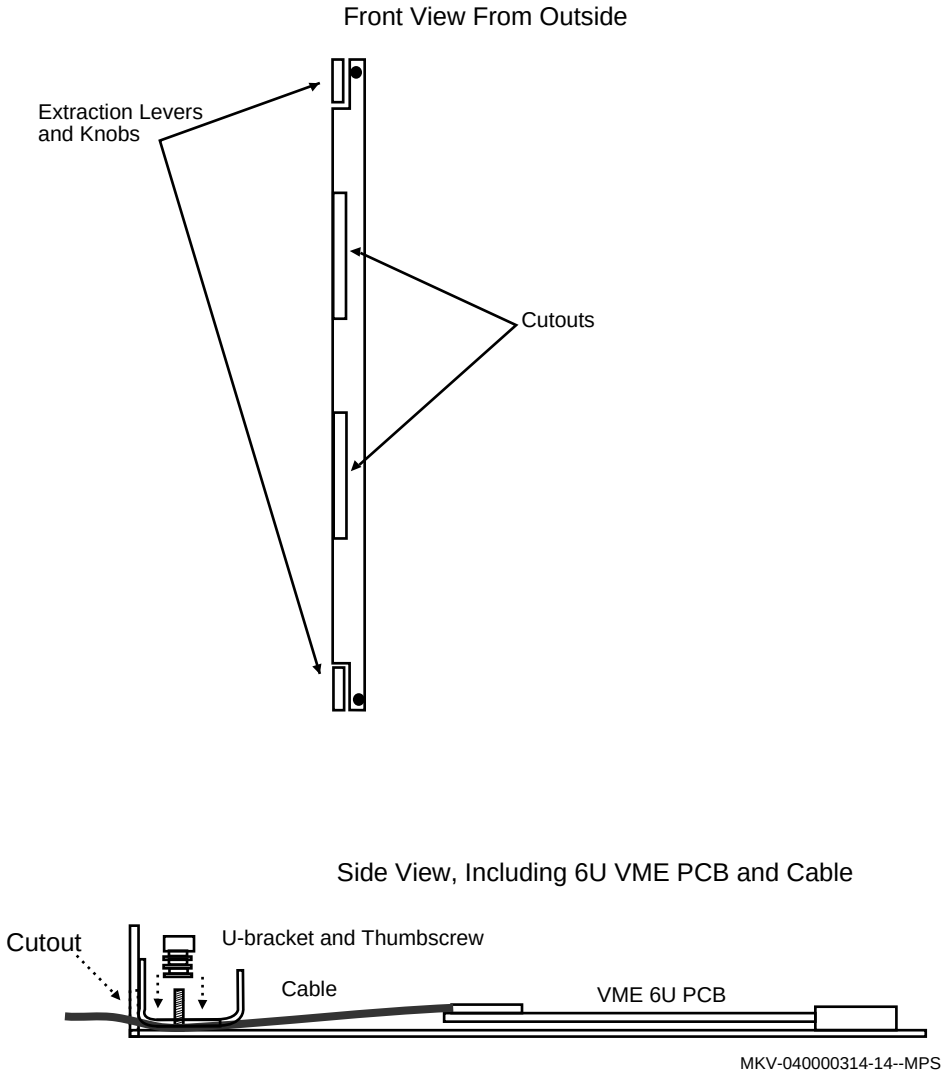
2.1.2 Card Cage Slot Assignment

The Parallel VME 6U Adapter PCB may be installed in any one of the I/O slots identified in Figure 1–3 except for the slots labeled ACU, IO00 (front-end VME interface) or IO08 (PVME). When installing the Parallel VME 6U adapter as an interface for the disk array, slot IO04 (sixth physical slot from the left) is recommended.

2.1.3 Cable Support Feature

Mechanical features on the Parallel 6U Adapter PCB and on the DPU cabinet facilitate cabling between the standard VME 6U PCB and other I/O devices in the DPU, or between the standard VME 6U PCB and I/O devices outside the DPU. Figure 2–3 shows the mechanical cutouts on the Parallel VME 6U Adapter PCB front panel.

Figure 2–3 Mechanical Cutouts



The front panel of the Parallel VME 6U adapter PCB has two cutouts; each cutout has a corresponding U-bracket behind the front panel. This bracket closes off the cutout when it is not being used and secures a flat cable when one is present. Holes in one of the U-brackets can be used to pass a round cable.

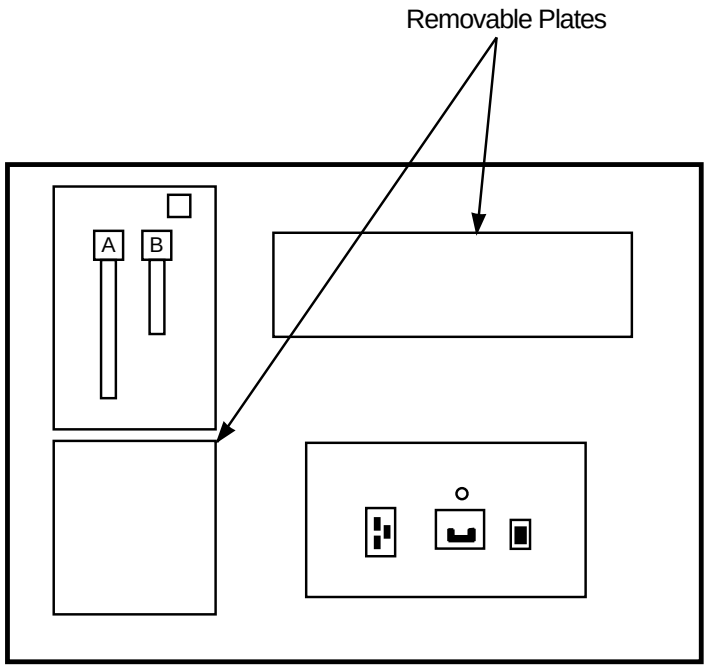
To mount a flat cable or a ribbon cable:

1. Each U-bracket is attached to the Parallel VME 6U Adapter by two nuts attached to studs. Loosen the nuts until you can move the U-bracket out of the way enough to feed the cable through.
2. Feed the cable under the U-bracket and through the cutout.
3. Tighten the nuts until the U-bracket is holding the cable firmly.

To mount a round cable, feed it through the holes in the U-bracket that covers the smallest cutout. It is not necessary to move the U-bracket.

The cable plates on the bulkhead at the bottom rear of the DECmpp 12000–LC/Sx DPU can be easily removed and fitted with connectors. Holes on the panels mate to threaded studs on the inside of the DPU. Figure 2–4 shows the location of the cable plates.

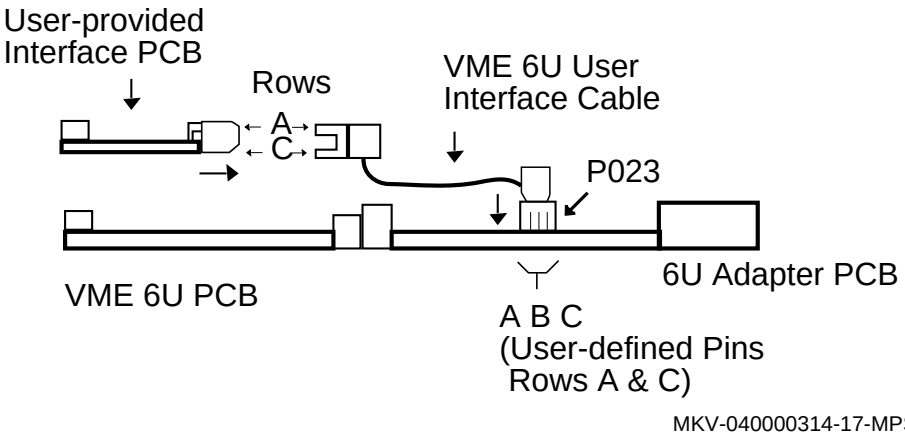
Figure 2–4 DECmpp 12000–LC/Sx DPU Cable Plates



MKV−040000314−22−RAGS

The Parallel VME 6U Adapter PCB also has a user interface cable to support interfaces from the user-defined pins on connector P023, as shown in Figure 2–5, to a user-supplied interface PCB. Rows A and C are brought out on the interface cable and appear as they would on a backplane. Row B is available at connector P023 with the standard VME signals for users who wish to customize their own interface using all three pin rows.

Figure 2–5 Interface Cable



2.2 Installing the Parallel VME 6U Adapter PCB

To install the Parallel VME 6U Adapter PCB, do the following:

1. Shut down the system, following the instructions in Chapter 1 of the *DECmpp 12000/Sx Hardware Service Manual*.
2. Set the switches on the PCB to identify its CSR address. Refer to Section 2.1.1, Table 2–1, and Table 2–2 for switch assignments.
3. Select the slot in which you insert the adapter PCB. The Parallel VME 6U Adapter PCB may be installed in any one of the I/O slots identified in Figure 1–3 except for the slots labeled ACU, IO00 (front-end VME interface), or IO08 (PVME). When the Parallel VME 6U Adapter and a 6U VME controller are installed as a disk array interface, slot IO04 is recommended.
4. If adding an optional user-interface VME 6U PCB, connect it to the Parallel VME 6U Adapter PCB as shown in Figure 2–5.
5. Remove the air baffle in the selected slot.
6. Slide the PCB into place.
7. Referring to Figure 1–4, remove the three jumpers above backplane slot IO 04.
8. Power on the system and boot it.
9. Test your installation by writing to your PCB's address and then reading the information back.
10. To test communications with the VME adapter, enter the following:

    ```
    # cd $MP_PATH/field/bin
    # hdb
    ```

    Note that `hdb` must be version 0,05.25 or greater.

    ```
    hdb > VME.0########
    ```

    Note that the address is dependent on the setting in switch packs S1–S4.

    If you encounter problems, check that you have properly set the switches and jumpers, as described earlier.

2.3 MPVMEbus Features

The Parallel VME 6U Adapter PCB supports VMEbus 6U PCBs. Power and ground are provided on the P021 and P022 connectors, as specified in the VMEbus specification. All VMEbus connectors to the 6U slot are VMEbus-standard except for J2-30; jumpers J02000 and J02001 on the 6U adapter PCB provide –5.2V at 1.2 A (maximum) to rows A and C.

The following is a summary of the features of the DECmpp 12000/Sx VMEbus:

- Support is provided for standard VMEbus 6U PCBs only.
- Support is provided for A32/A24 and D32/D16 addressing and data modes for attached VME PCB remote VME-to-Front-End-Bus (FE-Bus) transactions, as well as A32/A24 and D32/D16 addressing and data modes for the Parallel VME 6U Adapter PCB CSR.
- Unrestricted support is provided for addressing and data modes for local VME-to-VMEbus transactions; no support is provided for automatic read-modify-write transactions to the FE.
- Support is provided for user-defined external cable assemblies.
- Support is provided for both VMEbus master and slave interfaces.
- Masters can perform local VME-to-VMEbus or remote VME-to-FE-Bus transactions.
- Support is provided for VMEbus single-level arbiters only. BR3* and the corresponding bus grant daisy-chain signals BG3IN and BG3OUT are delivered to the VMEbus PCB attached to the VMEbus adapter. Support is provided for ROR (release on request) bus release protocol; all unused grant and request signals are pulled up.
- When a standard VMEbus PCB bids for bus mastership, a MPVMEbus register (CSR) is used to configure the Parallel VME 6U Adapter PCB for local VME-to-VMEbus and remote VME-to-FE-Bus transactions. (The 24-bit addressing mode is selected using standard address modifiers.)
- Parity generation and checking is provided for the address, data, and address modifier portions of the MPVMEbus.
- MPVMEbus slave CSR also provides a mechanism to disable parity checking for diagnostics.
- Address and data parity is always generated across 32 bits, even for A24 and D16 modes.
- The Parallel VME 6U Adapter CSR can be placed in any 32-bit word boundary within either A24 or A32 address space.

VMEbus masters can access MPVMEbus resources that physically reside on the local MPVMEbus or on the remote front-end bus. This is selected by the local/remote bit in the adapter's CSR.

The standard VME PCB must be configured for single-level bus arbitration. The Parallel VME 6U Adapter PCB uses only the VMEbus BR3*, BG3IN, and BG3OUT signals, in combination with the adapter's CSR bit, to generate and monitor the appropriate MPVMEbus bus request and grant control signals.

BR2* can be asserted by the Parallel VME 6U PCB to the attached standard VME PCB to signal a release request by another VME PCB (ROR protocol). BR0* and BR1* are terminated using pull-ups on the Parallel VME 6U Adapter PCB. All VMEbus addressing modes and all data transfer modes are supported for MPVMEbus-to-VME transfers.

2.3.1 Parity Detection and Reporting

The Parallel VME 6U Adapter PCB generates odd-bit parity for each byte of the data, address, and address modifiers. When it generates an address or data, it generates parity; when it inputs the address and data, it checks the parity.

Parity errors are reported using front-end interrupt level IPL17. Note that a MPVMEbus interrupt is not generated for parity errors. Information about the error is latched into the CSR.

Parity errors are reported on all PCBs on the MPVMEbus. To identify the cause, look for the PCB that was driving the MPVMEbus when the parity error occurred.

2.3.2 Voltage Selection

Jumpers J02000 and J02001 connect –5.2 V to P022 and P023, pin 30 in rows A and C. When J02000 and J02001 are IN, power is connected; when J02000 and J02001 are OUT, power is disconnected. Note that the maximum current drain is 1.2 A. The default is OUT.

2.3.3 Control and Status Register (6UCSR)

A register on the Parallel VME 6U Adapter PCB provides control and status bits. Most of them deal with the parity generation/detection system. One of them enables and disables access to the front-end space.

The CSR address is set by switch packs S1 through S4. It can be set to any 32-bit boundary in VMEbus space. Each switch position represents a single VMEbus address bit. Incoming addresses are compared with the switch positions; when they match, the CSR is accessed.

The CSR bits shown in Figure 2–6 are defined in Table 2–3.

Figure 2–6 Control and Status Register (6UCSR)

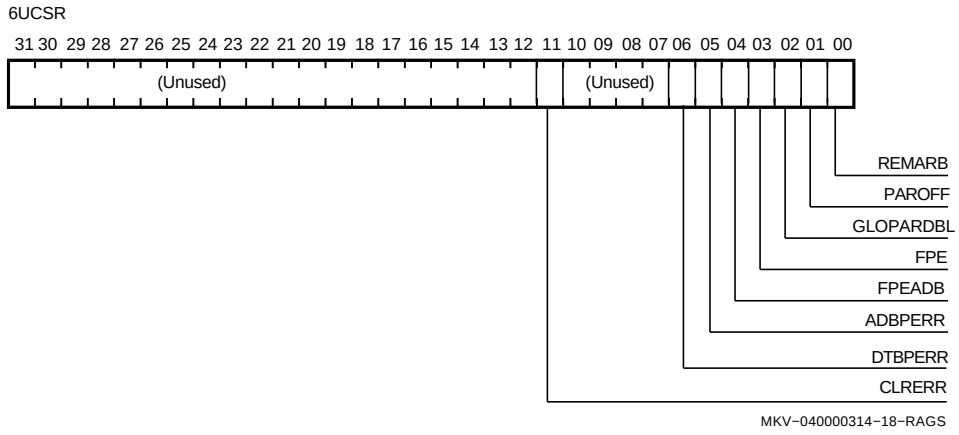


Table 2–3 Control and Status Register (6UCSR)

Extent	Name	Description
<31:12>		Unused
<11>	CLRERR	When this bit is set to 1, all the parity error latches are cleared.
<10:07>		Unused
<06>	DTBPERR	When a parity error is detected on the data bus, this bit is latched into the CSR. Only the first error is latched; it remains latched until it is cleared by writing a 1 to CLRERR (<11>). This bit is initialized to 0 on power-up and when VMEReset* is asserted. For data bus parity errors, only the 6U adapter PCBs involved in the transaction perform parity checking. This occurs during DS* for write operations and DTACK* for read operations.
<05>	ADBPERR	When a parity error is detected on the address or address modifier portion of the MPVMEbus, this bit is latched into the CSR. Only the first error is latched; it remains latched until it is cleared by writing a 1 to CLRERR (<11>). This bit is initialized to 0 on power-up and when VMEReset* is asserted. All 6U adapter PCBs monitor the address bus and address modifier bus. They perform parity checking whenever AS* is active, except during block-mode operations, when only the first address is checked.
<04>	FPEADB	When FPE (<03>) is asserted, this bit selects whether the resultant parity errors are forced on the address bus (and address modifier bus) or on the data bus (1 = data bus, 0 = address buses).
<03>	FPE	When this bit is set to 1, it forces incorrect parity on either the address bus and the address modifier bus, or on the data bus. FPEADB (<04>) selects between these buses.
<02>	GLOPARDBL	When this bit is set to 0, it disables systemwide parity checking for data asserted by the VMEbus device supported by this PCB.
<01>	PAROFF	When this bit is set to 0, it disables the mechanism that reports parity errors. Note that information about the parity error is latched, but the resulting interrupt is not generated.
<00>	REMARB	When this bit is set to 0, the VMEbus device can access both VMEbus addresses and front-end addresses. When it is set to 1, it can access only VMEbus addresses. Only at initialization does the Parallel VME 6U adapter PCB change this value by setting it to 0.

A

PVME Address Maps

The address map specifies where memory and control registers exist relative to the base address of the board. The PVME has two separate address maps, depending on the address mode: A24 or A32.

All PVME addresses are accessible using D16 or D32. IORAM is also accessible as D8. Addressing an address in a D16/D32 register as D8 or addressing outside of the valid address space causes a VMEbus Bus Error (BERR).

Table A–1 shows the addresses for the A24 address mode; Table A–2 shows the addresses for the A32 address mode. Map addresses are given in hex notation.

Table A–1 A24 Address Map

Start	End	Description	Access
xx00 0000	xx01 FFFF	IORAM (Swap Mode 0)	D8/D16/D32
xx02 0000	xx03 FFFF	IORAM (Swap Mode 1)	D8/D16/D32
xx04 0000	xx05 FFFF	IORAM (Swap Mode 2)	D8/D16/D32
xx06 0000	xx07 FFFF	IORAM (Swap Mode 3)	D8/D16/D32
xx08 0000	xx09 FFFF	IORAM (Swap Mode 4)	D8/D16/D32
xx0A 0000	xx0B FFFF	IORAM (Swap Mode 5)	D8/D16/D32
xx0C 0000	xx0E 0003	(Spare)	D16/D32
xx0E 0004	xx0E 0007	Queue Manager CSR	D16/D32
xx0E 0008	xx0E 000B	VME Interrupt Mask	D16/D32
xx0E 000C	xx0E 000F	VME Interrupt CSR	D16/D32
xx0E 0010	xx0E 0013	PVME Operation CSR	D16/D32
xx0E 0014	xx0E 0017	PVME State CSR	D16/D32
xx0E 0018	xx0E 001B	Swap Diagnostic CSR	D16/D32
xx0E 001C	xx0E 001F	(Spare)	D16/D32
xx0E 0020	xx0E 0023	IORAM DMA CSR	D16/D32
xx0E 0024	xx0E 0027	IORAM DMA Configuration	D16/D32
xx0E 0028	xx0E 002B	IORAM DMA VME Addr	D16/D32
xx0E 002C	xx0E 002F	IORAM DMA IORAM Addr	D16/D32
xx0E 0030	xx0E 0033	IORAM DMA Byte Count	D16/D32
xx0E 0034	xx0E 0037	IORAM DMA Semaphore	D16/D32
xx0E 0038	xx0E 003F	(Spare)	D16/D32
xx0E 0040	xx0F FFFF	(Reserved)	D16/D32

Table A–2 A32 Address Map

Start	End	Description	Access
0000 0000	007F FFFF	IORAM (Swap Mode 0)	D8/D16/D32
0080 0000	00FF FFFF	IORAM (Swap Mode 1)	D8/D16/D32
0100 0000	017F FFFF	IORAM (Swap Mode 2)	D8/D16/D32
0180 0000	01FF FFFF	IORAM (Swap Mode 3)	D8/D16/D32
0200 0000	027F FFFF	IORAM (Swap Mode 4)	D8/D16/D32
0280 0000	02FF FFFF	IORAM (Swap Mode 5)	D8/D16/D32
0300 0000	0BFF FFFF	(Undefined)	N/A
0C00 0000	0C02 0007	(Spare)	D16/D32
0C02 0008	0C02 000B	VME Interrupt Mask	D16/D32
0C02 000C	0C02 000F	VME Interrupt CSR	D16/D32
0C02 0010	0C02 0013	PVME Operation CSR	D16/D32
0C02 0014	0C02 0017	PVME State CSR	D16/D32
0C02 0018	0C02 001B	Swap Diagnostic CSR	D16/D32
0C02 001C	0C02 001F	(Spare)	D16/D32
0C02 0020	0C02 0023	IORAM DMA CSR	D16/D32
0C02 0024	0C02 0027	IORAM DMA Configuration	D16/D32
0C02 0028	0C02 002B	IORAM DMA VME Addr	D16/D32
0C02 002C	0C02 002F	IORAM DMA IORAM Addr	D16/D32
0C02 0030	0C02 0033	IORAM DMA Byte Count	D16/D32
0C02 0034	0C02 0037	IORAM DMA Semaphore	D16/D32
0302 0038	0C02 203F	(Spare)	D16/D32
0302 0040	0C02 FFFF	(Reserved)	D16/D32

B

Swap Mode Definitions

This section provides descriptions and tables of the swap modes.

B.1 Mapping IORAM to VME Addresses

Each entry in IORAM consists of 8 bytes. For purposes of this discussion, assume that the byte corresponding to bits 7:0 is labeled byte 0 and contains data identified as A. The byte corresponding to bits 15:8 is labeled byte 1 and contains data identified as B.

Table B–1, Table B–2, and Table B–3 demonstrate how the byte labeling and swap modes are defined.

Table B–1 Byte Labeling

Bits	IORAM Byte Offset	Data
07:00	0	A
15:08	1	B
23:16	2	C
31:24	3	D
39:32	4	E
47:40	5	F
55:48	6	G
63:56	7	H

For each swap mode, VMEbus addressing of this IORAM location produces the output in these tables:

Table B–2 IORAM Swap Mode Definitions

Swap Mode	Byte Addr 0	Byte Addr 1	Byte Addr 2	Byte Addr 3	Byte Addr 4	Byte Addr 5	Byte Addr 6	Byte Addr 7
0	A	B	C	D	E	F	G	H
1	A	B	C	D	E	F	G	H
2	A	B	C	D	E	F	G	H
3	A	B	C	D	E	F	G	H
4	A	B	C	D	E	F	G	H

(continued on next page)

Table B–2 (Cont.) IORAM Swap Mode Definitions

Swap Mode	Byte Addr 0	Byte Addr 1	Byte Addr 2	Byte Addr 3	Byte Addr 4	Byte Addr 5	Byte Addr 6	Byte Addr 7
5	A	B	C	D	E	F	G	H

Table B–3 VMEbus Swap Mode Definitions

Swap Mode	Byte Addr 0	Byte Addr 1	Byte Addr 2	Byte Addr 3	Byte Addr 4	Byte Addr 5	Byte Addr 6	Byte Addr 7
0	A	B	C	D	E	F	G	H
1	B	A	D	C	F	E	H	G
2	D	C	B	A	H	G	F	E
3	H	G	F	E	D	C	B	A
4	E	F	G	H	A	B	C	D
5	F	E	H	G	B	A	D	C

Index