EPC®-5

Hardware Reference

(includes 100 MHz option)

RadiSys® Corporation

15025 SW Koll Parkway

Beaverton OR 97006

Phone: (503) 646-1800

Fax: (503) 646-1850

07-0059-03 July 1994

April 1990

Hardware Warranty

RadiSys Corporation ("RadiSys") warrants the EPC system and component modules to the original purchaser for two years from the product's shipping date. If an EPC product fails to operate in compliance with its specification during this period, RadiSys will, at its option, repair or replace the product at no charge. The customer is, however, responsible for shipping the product; RadiSys assumes no responsibility for the product until it is received. This warranty does not cover repair of products that have been damaged by abuse, accident, disaster, misuse, or incorrect installation.

RadiSys' limited warranty covers products only as delivered. User modification, such as the addition of memory arrays or other devices, may void the warranty, and if the product is damaged during installation of the modifications, this warranty does not cover repair or replacement.

This warranty in no way warrants suitability of the product for any specific application.

IN NO EVENT WILL RADISYS BE LIABLE FOR ANY DAMAGES, INCLUDING LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT EVEN IF RADISYS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY PARTY OTHER THAN THE PURCHASER.

THE ABOVE WARRANTY IS IN LIEU OF ANY AND ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED OR STATUTORY, INCLUDING THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR USE, TITLE AND NONINFRINGEMENT. Repair or replacement as provided above shall be the Purchaser's sole and exclusive remedy and RadiSys' exclusive liability for any breach of warranty.

Table of Contents

List of Illustrations

1. Product Description

The EPC-5 is a PC/AT compatible embedded CPU module containing the following:

- 25 or 33 MHz Intel486 DX, 66 MHz Intel486 DX2 processor, or a 100 MHz Intel486 DX4 processor
- 8 Kbytes of cache (16 Kbytes on the DX4) and math co-processor on-chip
- 4, 8, 16 or 32 MBytes of DRAM memory on 25, 33, and 66 Mhz EPC-5s, or 8, 16, 32 and 64 MBytes of DRAM memory on the 100 MHz EPC-5
- Keyboard interface
- 2 standard 9-pin DTE serial ports (COM1 & COM2)
- 1 standard parallel port (LPT1)
- Time-of-day clock with user-replaceable battery
- Award 486 BIOS with extensions for a SCSI device driver
- VMEbus interface
- EXM expansion interface

The EPC-5 form factor has been designed to the VMEbus specification (6U). It provides direct VMEbus communication to all three VMEbus address spaces (A32, A24, & A16). The EPC-5 DRAM has dual-ported access from both the PC side and the VME side.

The EXM expansion interface is electrically similar to the 16-bit PC/AT ISA bus. Video is provided through an add-in card called an EXM (Expansion module). Mass storage can be added via a Mass Storage module (EXP-MS/MX) inside the VME chassis or externally via other EXMs that provide SCSI or IDE interfaces. Many other EXMs are available to provide additional peripherals such as: additional serial ports (RS232 or RS422), internal modem, Flash EPROM memory, Timer/Counter, PCMCIA adapter and Ethernet controllers. Also, an adapter module (EXP-AM) can be used to install a single 8-bit PC add-in short card.

1

Specifications

Environmental				
Temperature	operating	0° to 60° C		
	storage	-40° to 85° C		
Humidity	operating	5 - 95% (non-condensing)		
	storage	5 - 95% (non-condensing)		
Vibration	operating	.015"PP 2.5g (max) 5-2000 Hz		
	storage	.030"PP 5g (max) 5-2000 Hz		
Shock	operating	30g 11 msec duration		
	storage	50g 11 msec duration		
Electrical				
		+5V	+12V	-12V
25 MHz - DX	maximum	5.4 A	100 mA	100 mA
	typical	4.4 A	100 mA	100 mA
33 MHz - DX	maximum	5.9 A	100 mA	100 mA
	typical	4.8 A	100 mA	100 mA
66 MHz - DX2	maximum	6.2 A	100 mA	100 mA
	typical	5.1 A	100 mA	100 mA
100 MHz - DX4	maximum	6.5 A	100 mA	100 mA
	typical	5.5 A	100 mA	100 mA

Table 1. EPC-5 Specifications.

2. Before Installation

Unpack the EPC-5 and inspect it for shipping damage.

☛ **DO NOT REMOVE THE EPC-5 MODULE FROM ITS ANTI-STATIC BAG UNLESS YOU ARE IN A STATIC-FREE ENVIRONMENT.**

The EPC-5, like most electronic devices, is susceptible to electrostatic discharge (ESD) damage. ESD damage is not always immediately obvious. It can cause a partial breakdown in semiconductor devices that might not result in immediate failure.

Configuring the EPC-5

The EPC-5 can be user-configured to provide standard VMEbus Slot-1 functionality. The Slot-1 configuration option is enabled (default) by installing the Slot-1 shunt (jumper) on the processor board (see Figure 1, page 4). Removing the jumper disables Slot-1 functionality. When the EPC-5 is configured as the Slot-1 controller, it performs all the standard VMEbus system control functions. See Chapter 5, *Theory of Operation* for more details on Slot-1 controller functions.

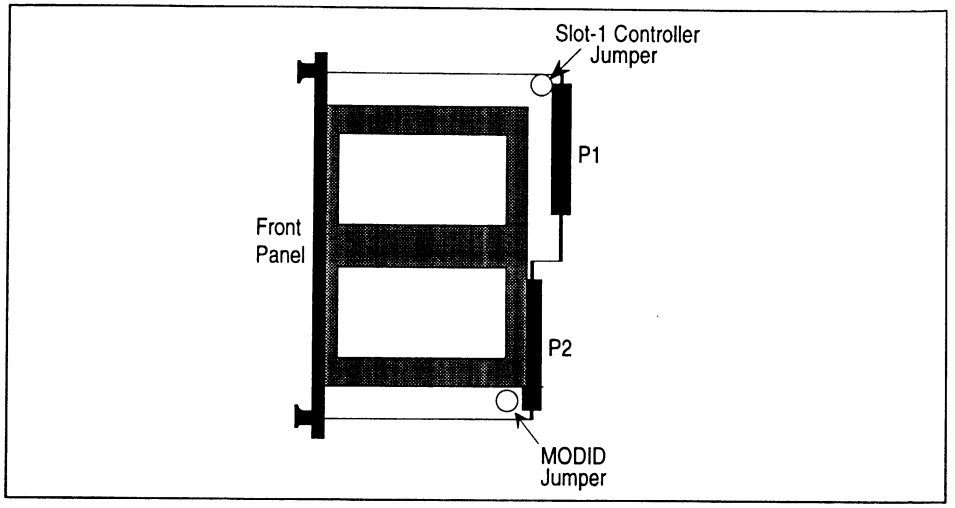


Figure 1. Slot-1 Jumper Location.

Additionally, the EPC-5 has another jumper (see Figure 1 above) that rarely needs to be changed - the MODID jumper. The EPC-5 uses pin 30, Row A of the P2 connector for module identification. If the J2 backplane is other than a standard VME or VXI backplane (e.g., a VSB backplane) or Pin 30, Row A is defined for another purpose, remove this jumper.

Selecting the EPC-5 Slot Location

There are two main considerations in determining where the EPC-5 should be positioned in the chassis.

- Per the VMEbus specification (Rule 3.3), the Slot-1 controller must be in Slot 1. All other boards must be to the right of the Slot-1 controller.
- The EPC-5 connects to its peripherals via a subplane which extends to the right of the EPC-5. Make sure that the location you choose provides sufficient room for all the attached peripherals (EXMs and mass storage module).

The EPC-5 plus EXM expansion modules plus any mass storage module can be considered together as a single subsystem. Use the following worksheet to determine the total number of VME expansion interface slots your particular subsystem configuration requires.

Product	VME Slots	Total
EPC-5 (Includes first two EXM modules)	2	
Additional EXP-MC(s) (Holds additional two EXM modules)	1 each	
EXP-AM	2	
Mass Storage Module (EXP-MS/MX including EXP-MX200A and EXP-MX500) or EXP-MX200	2 3	
Total VMEbus slots used		

Table 2. VME Slots Available.

Once you have determined where the EPC-5 subsystem will be physically located in the chassis, the VME backplane must be jumpered appropriately.

Installing the VMEbus Backplane Jumpers

2

The VMEbus specification provides four bus grant signals (BG0 - BG3) and one interrupt acknowledge signal (IACK) via daisy-chain lines. Per the VMEbus specifications, all boards (that plug into the backplane) are required to correctly handle these signals. All slots that do not have a board plugged into the backplane (i.e. empty slots and slots occupied by EXMs or mass storage modules), need to be jumpered to allow the signals to pass through to other boards in the system.

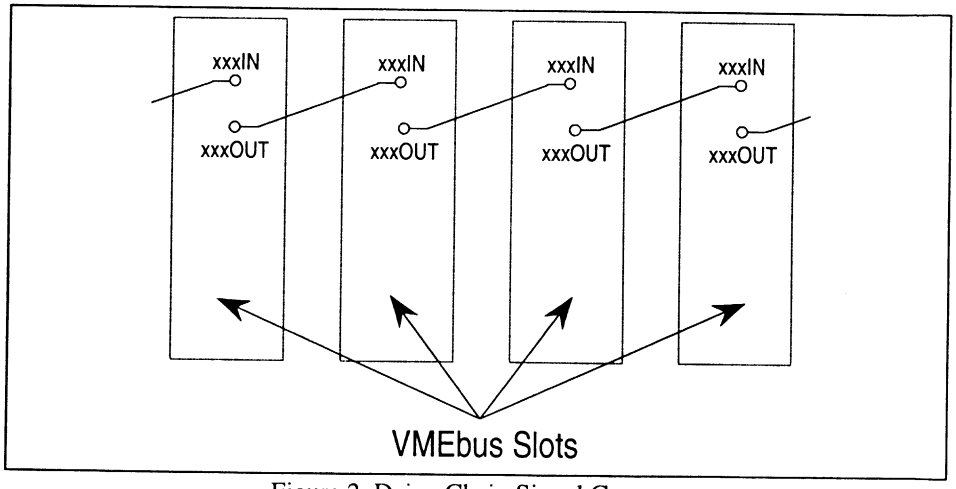


Figure 2. Daisy-Chain Signal Concept.

The Slot-1 controller board initiates each daisy-chain signal. Each VMEbus slot to the right of the Slot-1 controller must pass through each of the daisy-chain signals. For each VMEbus slot, xxxIn pin must be connected to its corresponding xxxOut pin (e.g. BG0In to BG0Out, BG1In to BG1Out,...,IackIn to IackOut) either through the board in that slot or by jumpers. Some boards correctly pass all of these signals, some boards handle some of these signals and not others, and some boards (typically "dumb" slave boards) may not handle any of these signals. Check the manual for each board to be installed to determine if these signals are passed through correctly. If they are not, or if the VMEbus slot is empty, all (or some) of these signals must be jumpered. See Figures 3 and 4 on the following page for examples.

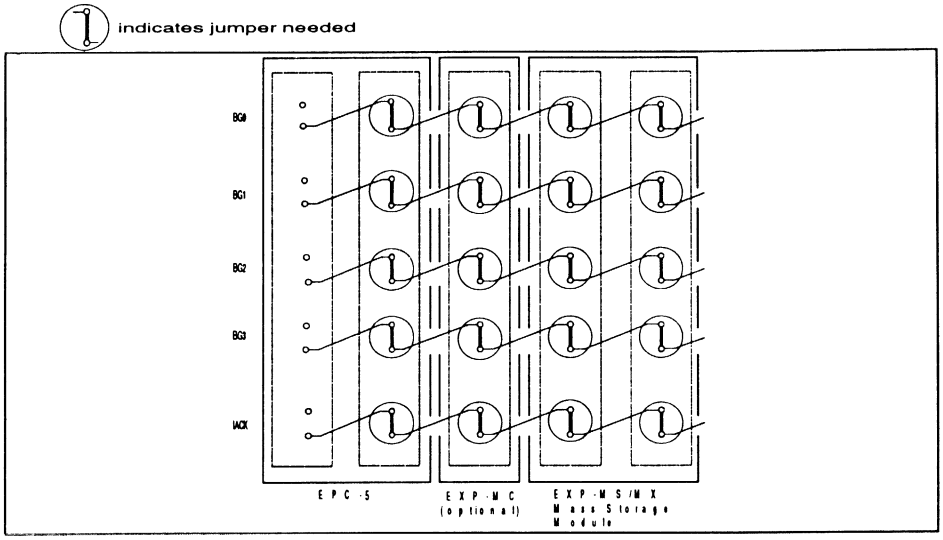


Figure 3. Backplane Jumpers required for EPC-5 subsystem.

The figure above shows the EPC-5 subsystem. Note that the left-most slot does not require any jumpers. All other slots occupied by the subsystem require all five jumpers be installed.

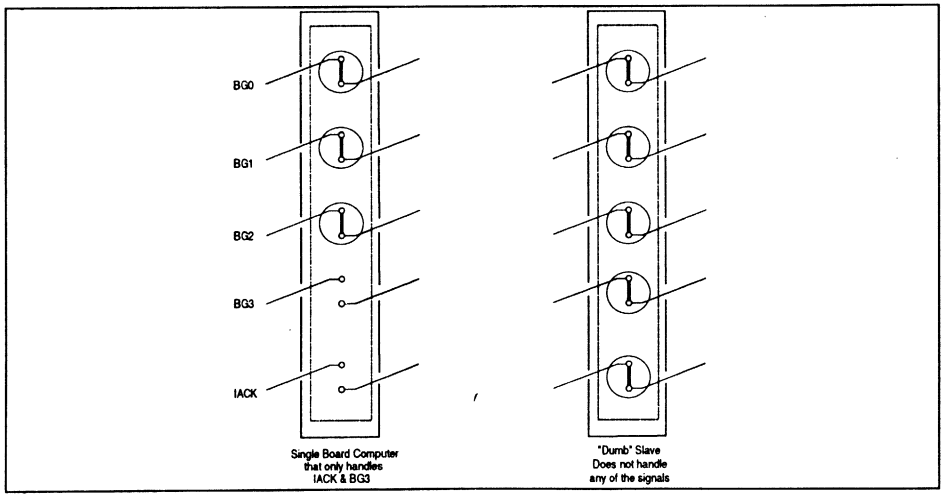


Figure 4. VMEbus Backplane Jumper Examples.

Once you have determined where the jumpers need to be, you must determine how to jumper your particular backplane. Different backplane manufacturers handle this in different ways; some provide stake pins on the rear of the backplane while others provide stake pins on the front of the backplane. These stake pins can be located in several different places.

If the stake pins are on the rear of the backplane, the most common location is in the middle of the J1 connector as shown in Figure 5 below. This can be just these pins extended or all pins extended for wirewrapping.

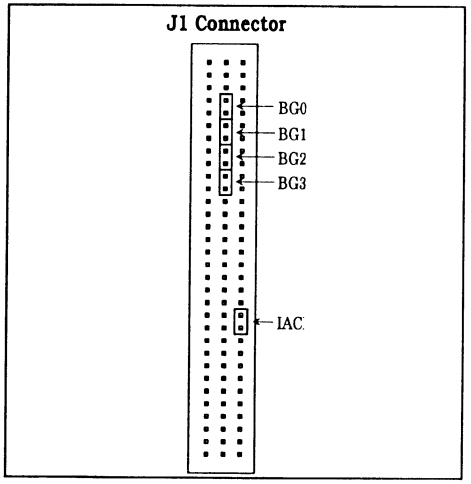


Figure 5. VMEbus Jumpers on Rear Wirewrap Pins.

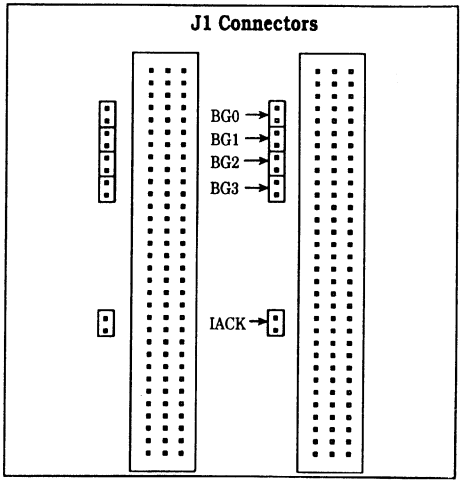


Figure 6. VMEbus Jumpers on Front Stake Pins.

The stake pins (front or rear) can also be located adjacent to the slot being jumpered as shown in Figure 6 above. Typically, the stake pins are located between the slot being jumpered and the next lower-numbered slot (e.g. jumpers for Slot 6 would be located adjacent to Slot 6 between Slots 5 and 6).

Consult your VME chassis reference manual or contact the chassis manufacturer if you are unsure where to jumper your particular system.

3. Installation

☛ **DURING ALL OF THIS INSTALLATION PROCESS, MAKE SURE THAT POWER TO YOUR VME SYSTEM IS OFF.**

The EPC-5 is not designed to be inserted or removed while the chassis is powered up.

☛ **MAKE SURE THAT THE INSTALLATION PROCESS DESCRIBED HERE IS PERFORMED IN A STATIC-FREE ENVIRONMENT.**

Do not remove any modules from their anti-static bags unless you are in a static-free environment. The EPC-5 module, like most other electronic devices, is susceptible to electrostatic discharge (ESD) damage. ESD damage is not always immediately obvious. It can cause a partial breakdown in semiconductor devices that might not result in immediate failure.

☛ **THE EXP-MS/MX MASS STORAGE MODULE CONTAINS A DELICATE HARD DISK.**

Subplane Installation

Subplanes are printed-circuit boards with connectors on both sides. A subplane provides several functions. Primarily it acts as the PC/AT bus. Additionally, it provides power from the VMEbus backplane to the EPC-5 and expansion modules. How subplanes function is discussed in detail in Chapter 5, *Theory of Operation.*

Depending on the particular EPC-5 subsystem configuration, a specific subplane will need to be installed. Locate the appropriate subsection for the subplane you are using either by name or by picture. Follow the directions in the appropriate subsection. A small bag of bolts, nylon washers, and nuts is provided for optionally securing the subplane to the VME backplane. If these are used, be careful not to over tighten the bolts. Over tightening causes the subplane to bend and may cause EXM failure due to poor contact.

EXP-BP2 Subplane

This subplane is used in the smallest configuration, where only the EPC-5 processor module occupies VME slot space. It provides connectivity for two EXM modules within the EPC-5 (e.g., a graphics controller and a network or disk controller). The EXP-BP2 is an L-shaped board with three connectors on each side.

After jumpering the backplane, plug the subplane into the VMEbus backplane such that the P2 connector on the back of the 4-row DIN is pressed into the J2 connector of the left-most VMEbus slot that the EPC-5 subsystem will occupy.

The subplane has holes for optional bolting to the VMEbus backplane using the screws included.

The lower EXM connector is denoted as EXM slot 0 and the upper as slot 1 as shown in the diagram. This information will be needed later when configuring the installed EXMs.

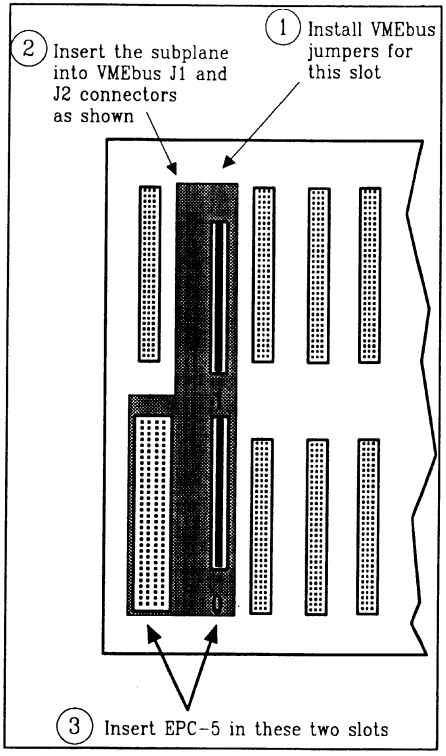


Figure 7. EXP-BP2 Subplane.

EXP-BP4 Subplane

The EXP-BP4 subplane is used to couple an EPC-5 processor module with an EXP-MS/MX Mass Storage module. The EXP-BP4 is a T-shaped board with four connectors on the front side and three on the rear.

After jumpering the backplane, plug the subplane into the VMEbus backplane such that the P2 connector on the back of the 4-row DIN is pressed into the J2 connector of the left-most VMEbus slot that the EPC-5 subsystem will occupy.

The subplane has holes for optional bolting to the VMEbus backplane using the screws included.

The EXM slot numbers are shown in the drawing.

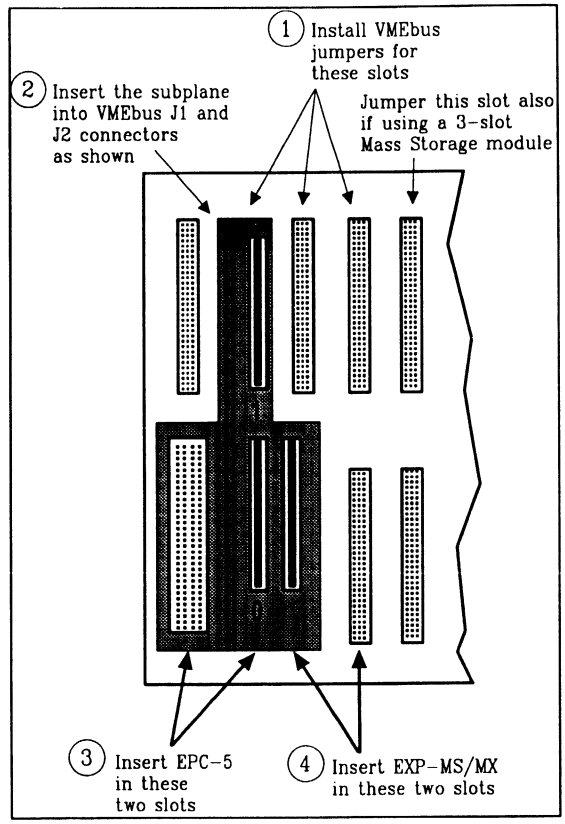


Figure 8. EXP-BP4 Subplane.

EXP-BP3A Subplane

The EXP-BP3A subplane is used to add an EXP-MC Module Carrier for the addition of one or two more EXM modules to an EPC-5 processor module. The EXP-BP3A has five connectors on each side.

After jumpering the backplane, plug the subplane into the VMEbus backplane such that the P2 connector on the back of the 4-row DIN is pressed into the J2 connector of the left-most VMEbus slot that the EPC-5 subsystem will occupy.

The subplane has holes for optional bolting to the VMEbus backplane using the screws included.

The EXM slot numbers are shown in the drawing.

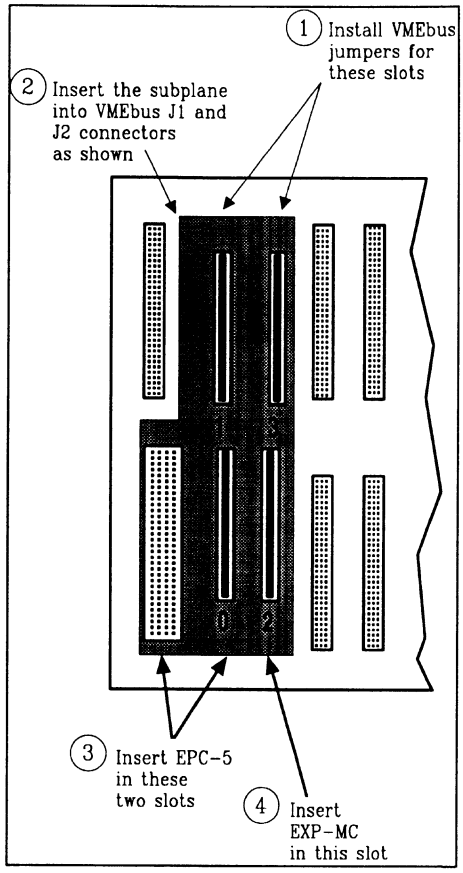


Figure 9. EXP-BP3A Subplane.

EXP-BP5 Subplane

The EXP-BP5 subplane is used in a configuration to couple an EPC-5 processor module with an EXP-MC Module Carrier and an EXP-MS/MX Mass Storage module. The EXP-BP5 has six connectors on the front side and five on the rear.

After jumpering the backplane, plug the subplane into the VMEbus backplane such that the P2 connector on the back of the 4-row DIN is pressed into the J2 connector of the left-most VMEbus slot that the EPC-5 subsystem will occupy.

The subplane has holes for optional bolting to the VMEbus backplane using the screws included.

The EXM slot numbers are shown in the drawing.

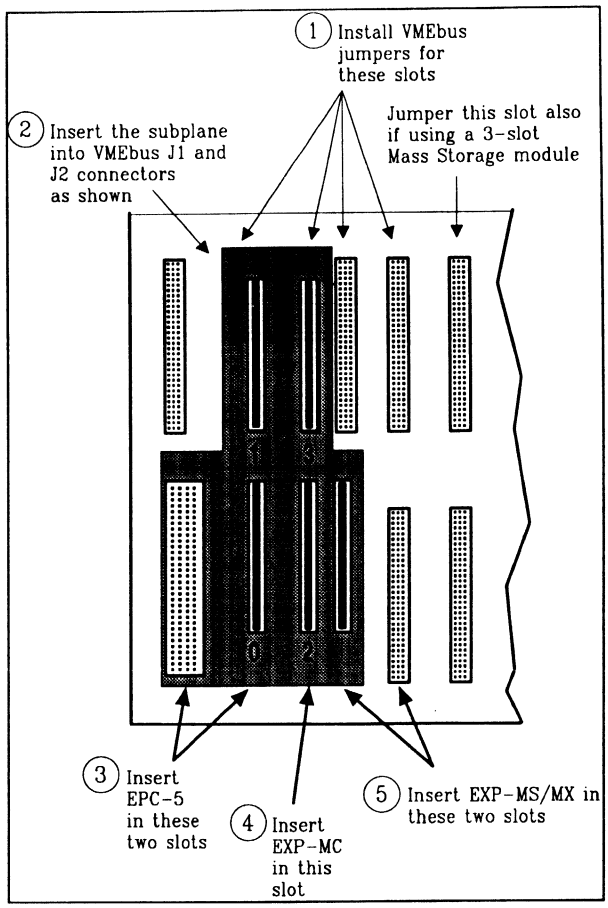


Figure 10. EXP-BP5 Subplane.

3

EXP-BP4A Subplane

The EXP-BP4A subplane is used to add either

- two EXP-MC Module Carriers

or

- one EXP-AM Adapter Module.

The EXP-BP4A has seven connectors on each side.

After jumpering the backplane, plug the subplane into the VMEbus backplane such that the P2 connector on the back of the 4-row DIN is pressed into the J2 connector of the left-most VMEbus slot that the EPC-5 subsystem will occupy.

The subplane has holes for optional bolting to the VMEbus backplane using the screws included.

The EXM slot numbers are shown in the drawing.

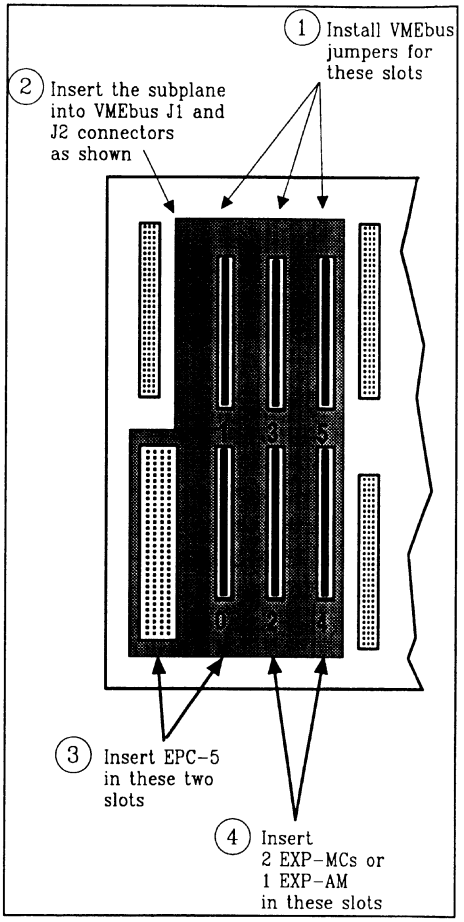


Figure 11. EXP-BP4A Subplane.

EXP-BP6 Subplane

The EXP-BP6 subplane is used in a configuration to couple an EPC-5 processor module with an EXP-MS/MX Mass Storage module and either

- two EXP-MC Module Carriers

  or

- one EXP-AM Adapter Module.

The EXP-BP6 has eight connectors on the front side and seven on the rear.

Plug the subplane into the VMEbus backplane such that the P2 connector on the back of the 4-row DIN is pressed into the J2 connector of the left-most VMEbus slot that the EPC-5 subsystem will occupy.

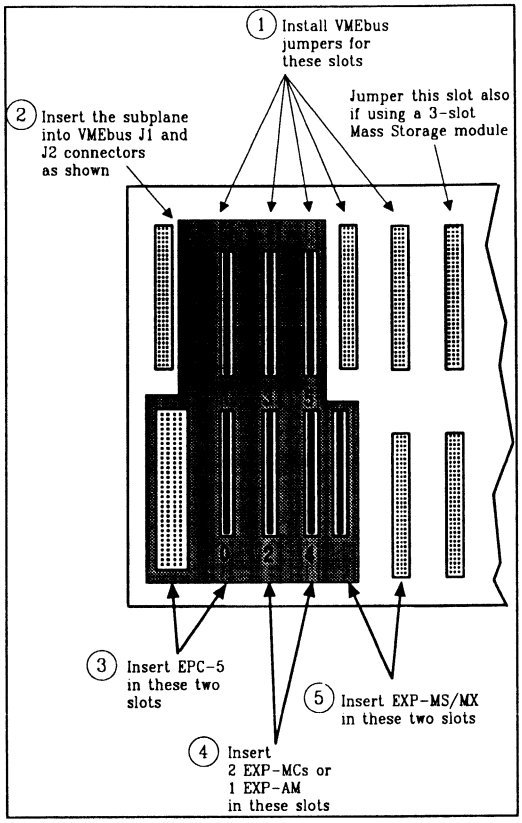


Figure 12. EXP-BP6 Subplane.

The subplane has holes for optional bolting to the VMEbus backplane using the screws included.

The EXM slot numbers are shown in the drawing.

EPC-5 Insertion

After installing the subplane, the EPC-5 processor module can be inserted into the VMEbus chassis.

3

☛ **MAKE SURE THAT POWER TO YOUR VME SYSTEM IS OFF. THE EPC-5 MODULE IS NOT DESIGNED TO BE INSERTED OR REMOVED FROM LIVE BACKPLANES.**

☛ **WHEN INSERTING THE EPC-5 MODULE, AVOID TOUCHING THE CIRCUIT BOARD AND CONNECTOR PINS, AND MAKE SURE THE ENVIRONMENT IS STATIC-FREE.**

θ Make sure the ejector handles are in the normal (non-eject) position. (Push the top handle down and the bottom handle up so that the handles are not tilted.)

θ Slide the EPC-5 module into the left-most slot occupied by the subplane. Use firm pressure on the handles to mate the module with the connectors.

θ Tighten the retaining screws in the top and bottom of the front panel to ensure proper connector mating and prevent loosening of the module due to vibration.

EXP-MC Module Carrier Insertion

If one or more EXP-MC Module Carriers are part of the configuration, they are inserted into the slot(s) immediately to the right of the EPC-5. The Module Carrier can only be used in a VMEbus slot where the subplane has both EXM connectors. Simply slide the Module Carrier into place and tighten the two top and bottom retaining screws.

The following figure shows a side view of an EXP-MC containing two EXMs plugged into a subplane that is plugged into a VMEbus backplane.

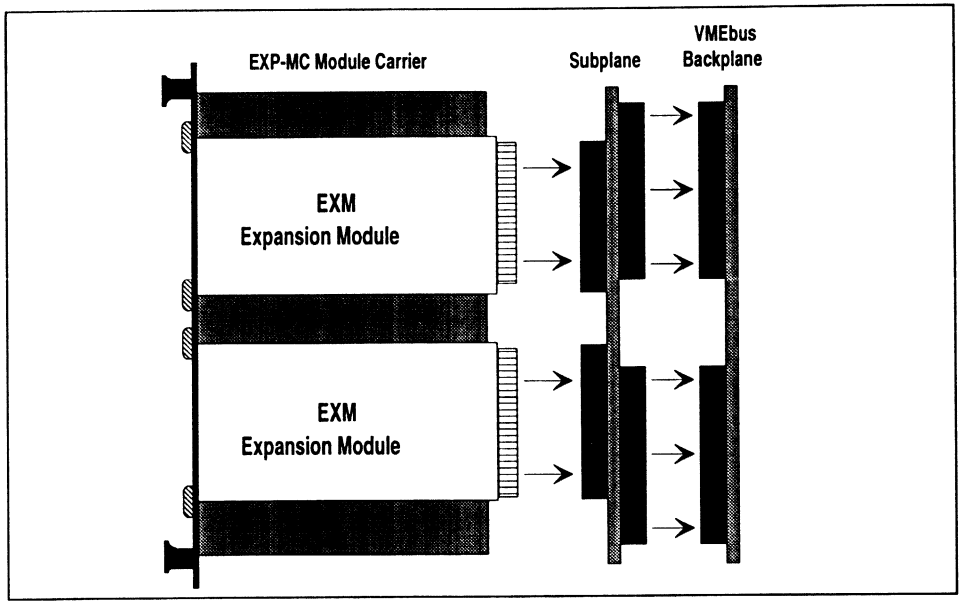


Figure 13. EXP-MC Module Carrier (side view).

EXP-MS/MX Mass Storage Module Insertion

☛ **HANDLE THE MASS STORAGE MODULE WITH CARE, AVOIDING SUDDEN DROPS AND JOLTS.**

☛ **WHEN INSERTING THE MODULE, AVOID TOUCHING THE CIRCUIT BOARD AND CONNECTOR PINS, AND MAKE SURE THE ENVIRONMENT IS STATIC-FREE.**

The Mass Storage module is always inserted as the rightmost module of the EPC-5 subsystem. Insert it so that its rear connector mates with the lower rightmost connector of the subplane. Insert it using adequate continuous force rather than tapping or hammering on it. Tighten the top and bottom front-panel screws to hold it firmly in place.

EXM Module Insertion

One or two EXMs may be installed through the front panels of the EPC-5 and each EXP-MC Module Carrier. To install an EXM:

θ Remove and save the blank face plate from the desired slot.

θ Slide the EXM into place in the card guides. Push firmly on the EXM front panel until the EXM card-edge connector is firmly seated in the subplane connector.

θ Tighten the thumb screws on the EXM's face plate.

Each EXM must be configured in the EPC-5's BIOS to set how the EXM should be initialized on power-up. This information is slot specific. Although EXMs can be installed in any available carrier slot, once an EXM is installed, it cannot be moved without re configuring the BIOS setup. Configuring the BIOS setup is discussed in the next chapter.

Connecting Peripherals to the EPC-5

WARNING!

DO NOT PLUG IN ANY CABLE OR CONNECTOR INTO THE FRONT PANEL CONNECTORS WHILE THE SYSTEM IS POWERED UP. IN GENERAL, ELECTRONICS EQUIPMENT IS NOT DESIGNED TO WITHSTAND POTENTIAL DAMAGE THAT COULD ARISE FROM FLUCTUATIONS IN POWER. **NEVER** PLUG IN A SERIAL OR PARALLEL DEVICE, KEYBOARD, TRANSCEIVER, MONITOR OR OTHER COMPONENT WHILE THE SYSTEM IS ON.

The final step of installation is connecting peripherals, typically a video display and keyboard, but also perhaps a mouse, modem, printer, etc. Unless otherwise noted, all connectors are compatible with those found on IBM-compatible desktop PCs, and therefore pin-by-pin details are not given in this chapter. Pin-outs are specified in Chapter 7, *Connectors*.

Monitor

Connection of a monitor requires the use of an EXM video controller. Consult the video controller manual for details.

The monitor should be attached and powered on prior to applying power to the EPC-5. If this is not done, the EPC-5 cannot detect the monitor type and the video adapter may not be initialized correctly.

Keyboard

The front panel contains a round 6-pin DIN jack for connecting a keyboard. The jack is compatible with that of some newer PCs, and is not compatible with the previous style of larger 5-pin PC/AT keyboard connectors. However, an adapter cable is provided with the EPC-5 so either type of PC keyboard can be used with the EPC-5.

To operate your system without a keyboard, start with a keyboard and invoke the BIOS setup screen to change the configuration-errors field to "ignore keyboard errors." The system can then be booted with or without a keyboard.

Serial Ports

The front panel contains two DB-9 DTE serial-port connectors. They are identical to the standard serial ports COM1 and COM2 in the PC/AT and compatibles. They may be used for connecting a mouse, modem, serial printer, RS-232 link, etc.

The driver software for most types of mice detect the presence of the mouse dynamically, so it usually doesn't matter which port the mouse is connected to. A common mistake is connecting the mouse too late (e.g., Microsoft Windows looks for a mouse when Windows is first invoked, so plugging a mouse in after Windows has been started has no effect).

Some PCs and PC peripherals contain DB-25 serial connectors instead of DB-9 connectors. Adapters converting between the two are readily available.

Parallel Printer Port

The parallel port on the front panel is a DB-25 connector that is completely compatible with the corresponding LPT1 connector on IBM PCs and compatibles. Typically it is used to connect printers and software security keys.

EXP-MS/MX Ports

In addition to containing floppy and hard disk drives, the EXP-MS and EXP-MX Mass Storage modules contain front-panel connectors for adding external devices. Consult the EXP-MS/MX manual for further information.

4. Configuring the BIOS Setup

Power-On Screen Display

Whenever a hardware reset of the EPC-5 occurs (power-on or front panel reset), the system performs a power-on self-test (POST) which displays information on the attached monitor showing the status of the BIOS self tes. If everything proceeds normally, the screen image should appear approximately as shown below.

```
486 Modular BIOS Version 3.03, Copyright Award Software Inc.
Copyright 1988,1990 RadiSys Corporation  BIOS V3.05

TESTING INTERRUPT CONTROLLER #1 ..........  OK
TESTING INTERRUPT CONTROLLER #2 ..........  OK
TESTING CMOS BATTERY .....................  OK
TESTING CMOS CHECKSUM ....................  OK
SIZING SYSTEM MEMORY .....................  640K FOUND
TESTING SYSTEM MEMORY ....................  640K    OK
CHECKING UNEXPECTED INTERRUPTS AND STUCK NMI .....  OK
TESTING PROTECTED MODE ...................  OK
SIZING EXPANSION MEMORY ..................  3072K FOUND
TESTING MEMORY IN PROTECTED MODE .........  3712K FOUND
TESTING PROCESSOR EXCEPTION INTERRUPTS ...  OK
TESTING VME INTERFACE ....................  OK
TESTING VXI INTERFACE ....................  OK
TESTING SERIAL PORT #1 ...................  OK
TESTING SERIAL PORT #2 ...................  OK
TESTING PARALLEL PORT ....................  OK

RadiSys EPC Setup Program
```

Figure 14. EPC Setup Program.

For an 8 MB EPC-5 system, expect to see the following:

```
SIZING EXPANSION MEMORY ..........  7168K FOUND
TESTING MEMORY IN PROTECTED MODE .  7808K FOUND
```

For a 16 MB EPC-5 system, you'll see 15360K and 16000K respectively.

If errors occur during the power-on self-test, the BIOS displays the error on the appropriate line of the screen display and attempt to continue. For instance, if an error is discovered accessing COM1, the BIOS disables COM1, display the error on the line "TESTING SERIAL PORT #1 ....." and then continue as if the error did not occur. It is important to watch the POST display to verify that no errors occur.

If error messages are displayed during or after the POST display, see Chapter 9, *Troubleshooting & Error Messages.*

BIOS Setup Screen

The EPC-5's BIOS contains a setup function to display and alter the system configuration. This information is maintained in the EPC-5's nonvolatile CMOS RAM and is used by the BIOS to initialize the EPC-5 hardware.

The setup function can be invoked any time after the POST completes and first clears the screen. Simultaneously press the CTRL+ALT+ESC keys. This may be done during system operation in most, but not all circumstances. Some programs that take control of the keyboard at a low level, such as Microsoft Windows, cause this key sequence to be interpreted differently, or not at all. It should always work, however, when the DOS operating system prompt is shown on the screen.

The main setup screen looks like the following:

```
RadiSys EPC-5 CMOS Setup, BIOS V3.05
25 MHz 486, 4 MBytes memory

Date (mm/dd/yy) ..... ..... 01/12/90
Time (hh:mm:ss) ..... ..... 09:34:56
Configuration Errors .. ..... Halt on all errors

Diskette Drive A ..... ..... 1.4M 3.5 inch
Diskette Drive B ..... ..... None
Fixed Disk Drive C .... AT
Fixed Disk Drive D .... None

Bus Priority ..... ..... Pri 3          F2  = EXM Menu
Bus Release Method .... ..... ROR        F3  = Fixed Disk Menu
Slot 1 Arbitration .... ..... Priority   F10 = Save CMOS and EXM data
VXI Register Base (ULA) .....             FE00 (F8) ESC      = Exit
without saving
Slave Memory Offset ... ..... 18000000 (A32)    ↑ ↓ ↵ move between items
                                          ← → select values
```

Figure 15. EPC Main Setup Screen.

Use the up and down cursor (arrow) keys to move from field to field. For most fields, once you are positioned at the field, pressing the left and right cursor (arrow) keys will rotate through the available choices. Once the screen has been changed to appear as you desire, press the F10 function key to save the changes in nonvolatile (battery-backed) CMOS RAM and then the F5 function key to confirm the changes and reboot. Alternately, press ESC to ignore any changes and exit.

The fields are explained below.

Date and Time

These values are changed by moving to them and typing in the format shown.

Configuration Errors

This field provides several choices about the situations under which the BIOS should wait for user input if a configuration error is found. The selections are:

1) `Halt on all errors`
2) `Ignore all errors`
3) `Ignore keyboard errors` (allows operation without a keyboard)
4) `Ignore disk errors`
5) `Ignore keyboard and disk errors.`

Diskette Drive

This field gives you several choices about the type of floppy disk drives installed as the A and B drives. Typically when using the EXP-MS or EXP-MX Mass Storage modules you would select 1.4M 3.5 inch (1.44 MB 3.5" disk) for A and NONE for B.

Fixed Disk Drive

This display-only field shows the type of disk selected from the fixed disk menu. Possible values are None, SCSI, AT, Flash, and VME. To see the detailed characteristics of the device or to change the device, use the F3 function key to go to the fixed disk menu. See the section *Fixed Disk Menu.*

Bus Priority

This field allows selection among the four VMEbus priority levels. This is the level at which the EPC-5 contends for the bus when it performs a VMEbus access. Priority 3 is the highest priority; priority 0 is the lowest.

Bus Release Method

This field entry toggles between two bus-release modes: ROR (release on request) and RONR (request on no request, also known as the VXI fair-requester mode). ROR results in slightly better EPC-5 performance when accessing the VMEbus; RONR directs the EPC-5 to not "park" on the bus and thus slightly improves the access time of other VMEbus masters to the bus.

There is a caveat to using RONR. If the EPC-5 is set to RONR, then all other bus masters should also be set to RONR. If this is not done, the EPC-5 could be effectively "locked" off the bus due to frequent bus requests from the other bus masters.

Slot 1 Arbitration

This field toggles between the two arbitration algorithms provided by the EPC-5 when it is configured as the Slot-1 system controller: priority arbitration and round-robin arbitration. This only has meaning when the EPC-5 is the Slot-1 controller.

VXI Register Base (ULA)

The EPC-5 has a set of configuration registers consistent with the VXIbus specification that are mapped into the VMEbus A16 address space. This field allows you to choose among eight locations for these registers; FE00h, FE40h, FE80h, ... , FFC0h. In VXIbus terminology, the base address is associated with the device's ULA (unique logical address).

In a system with multiple EPCs, configure each EPC with a unique value.

Slave Memory Offset

This field specifies whether the EPC-5's DRAM is accessible from the VMEbus and, if so, whether the DRAM is mapped into the VMEbus A24 or A32 address space and the base address. If mapped to A24 space, the lowest 4 MBytes of PC memory is mapped to the VMEbus. If mapped to the A32 space, up to 16 MBytes of PC memory is mapped to the VMEbus.

You can select from the following 13 choices.

- DISABLED (DRAM is not mapped into the VMEbus address space)

- 18000000h (A32)
- 19000000h (A32)
- 1A000000h (A32)
- 1B000000h (A32)
- 1C000000h (A32)
- 1D000000h (A32)
- 1E000000h (A32)
- 1F000000h (A32)
- 000000h (A24)
- 400000h (A24)
- 800000h (A24)
- C00000h (A24)

This field has no effect on the EPC-5's behavior as an interrupter and interrupt handler, and does not affect the EPC-5's ability to act as a slave to VMEbus A16 requests that access the EPC-5's configuration and message registers.

EXM Setup Screen

A separate EXM setup screen is used to configure the EXM expansion modules in the system. It is displayed by pressing the F2 function key from the main setup screen. The EPC-5's nonvolatile (battery-backed) RAM holds identification and configuration information for six EXM modules. The BIOS displays the configuration information in low-level hexadecimal format.

The EXM Setup screen looks like the following:

```
RadiSys EPC-5 EXM Setup, System BIOS V3.05
25 MHz 486, 4 MBytes memory

Slot    ID   OB1   OB2
 0      FF    00    00
 1      DB    C1    00          F10 = Save and return
 2      DA    93    00          ESC = Return without saving
 3      DE    00    39
 4      F5    05    00              ↑ ↓ ← → move between items
 5      DC    F5    00
```

Figure 16. EXM Setup Screen.

EXMs must be defined in this screen so the BIOS can correctly identify and initialize each one at boot-up. Each EXM must be listed by slot number, ID and two option bytes as defined below.

SLOT — indicates the EXM slot in which the EXM is installed. See Figure 17 below to determine which slot each EXM occupies. Dotted lines indicate EXM slots that may or may not be present in all system configurations.

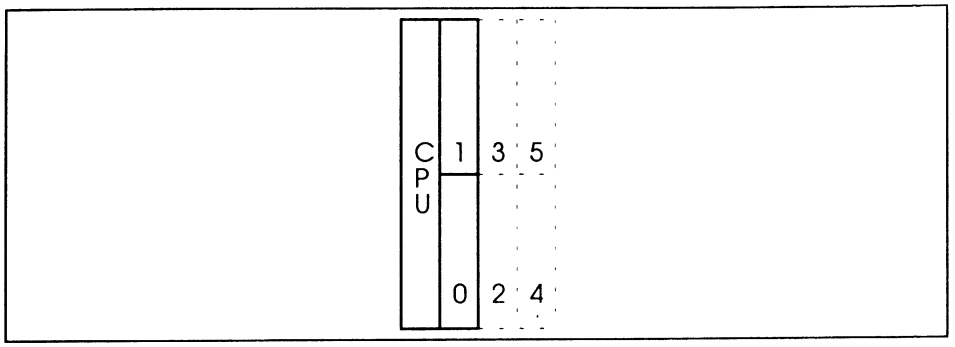


Figure 17. EXM Slot Numbering.

ID — is a hard-wired ID value. Each type of EXM has a unique ID value.

OB1/OB2 — are two "option" bytes of configuration information.

Note that all slots are listed even if the system configuration does not have 6 available EXM slots. All slots **not** occupied by an EXM module must show an ID of FF and OB1/OB2 of 00 00 indicating that no EXM is present. Note that if an EXM module is removed, this field must be manually re configured to FF for that EXM slot.

Consult the EXM manual for the correct configuration information for each EXM expansion module installed.

After all EXMs have been configured, press F10 to save the data or ESC to ignore the changes. In either case you will be returned to the main setup screen.

Fixed Disk Menu

The fixed disk menu is used to define the type of hard disk(s) installed in the system. Enter the fixed disk menu screen by pressing the F3 function key from the main setup screen.

The fixed disk menu screen looks like the following:

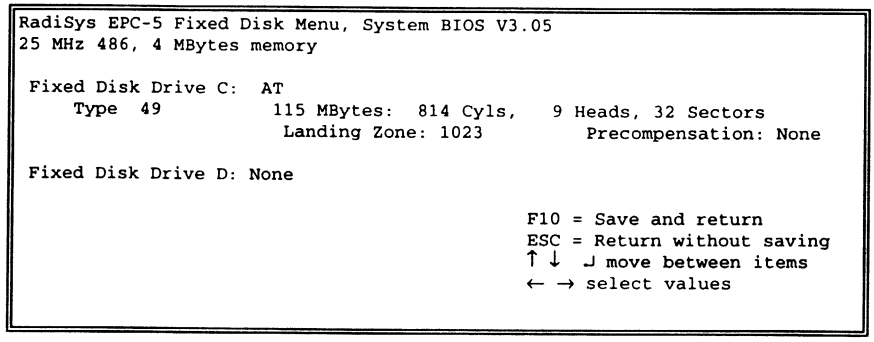


Figure 18. Fixed Disk Menu Screen.

Use the up and down cursor keys (↑ ↓) to move between items. Use the left and right cursor keys (← →) to toggle through the available choices for each item.

Choose **None** if there is no disk present or if you are using a SCSI device via the EXM-16 SCSI controller. In the case of no disk present, this avoids the timeout period used by the BIOS to conclude that there is no responding device. In the case of the EXM-16 device, the boot BIOS is provided by the EXM-16 BIOS ROM.

Choose **SCSI** for operation with the EXP-MS, EXP-MX front-panel SCSI connector (drive D: only), or EXM-3. There are no fixed disk characteristics to select because the BIOS determines them dynamically. If Drive C is set to SCSI, Drive D must also be set to SCSI even if no second drive is installed. If this is not done, the system will prompt you to correct the problem before exiting this screen.

Choosing disk type **VME** tells the BIOS to use an area of the VMEbus address space as the boot image. A set of addresses is given from which to select (actually, a set of starting addresses and search points). This facility allows the system to boot from a boot image located in a nonvolatile memory VMEbus board, or even from a boot image constructed in memory by another processor. RadiSys provides a utility program upon request to construct such a boot image.

Choosing disk type **Flash** tells the BIOS to use the EXM-2 flash memory device as a drive. An EXM-2 can be made the boot device by making Flash the drive C type. However, if Drive C is Flash, Drive D **must** be set to None.

Disk type **AT** denotes the many types of non-SCSI PC/AT compatible drives including IDE. You can scroll through a set of numbered types; the physical configuration is displayed for each. For operation with the EXP-MX or EXM-9, scroll through the numeric drive types to find the one matching the characteristics of the hard drive.

User-Definable Drive Types

If the correct **AT** disk type is not listed, the EPC-5 provides user-editable drive types 48 and 49. Select either of these drive types. Use the TAB key (→|)or the left and right cursor keys (← →) to move to the next (or previous) field. Note that the default settings for MBytes, Cylinders, Heads, and Sectors is 1. MBytes is a display-only field calculated by the BIOS. Move the cursor to each field (Cyls, Heads, and Sectors) and type the value for that field. The BIOS allows use of the following maximum values:

Cylinders	1023	Heads	63	Sectors	16

The hard disk you are using may have parameters larger than the allowable maximum. If the drive parameters are greater than the allowable maximum, divide the actual number of cylinders by 2 and multiply the actual number of heads by 2. IDE drives use Universal translation mode. That is, each sector is addressed as an absolute sequential sector number. Since the embedded "intelligent" controller converts the sector data to an absolute number, these "false" cylinder and head numbers will still allow the drive to be used. The example on the next page shows how this is done.

Example:

	Actual parameters	Conversion factor	Numbers to Use
Cylinders	1350	divide by 2	675
Heads	5	multiply by 2	10
Sectors	32	(none)	32
Total Sectors	216,000		216,000

Table 3. Fixed Disk Conversion Factors.

After the Fixed disk(s) have been configured, press F10 to save the data or ESC to ignore the changes. In either case you will be returned to the main setup screen.

NOTES

5. Theory of Operation

The EPC-5 is a PC/AT compatible processor. Most of the standard functions of the PC architecture is embodied in the TI83000 chip set. In addition, the EPC-5 has two proprietary interfaces: one for the EXM expansion interface and the other for the VMEbus.

Processor board

The EPC-5 processor board conforms with the VMEbus standard 6U form-factor.

Processor and Coprocessor

The processor in the EPC-5 is an Intel486 DX (32-bit bus interface) running at 25 or 33 MHz, or an Intel486-DX2 running internally at 66 MHz, with an external interface at 33 MHz, or an Intel486-DX4 running internally at 100 MHz with an external interface at 33 MHz.

The EPC-5 has a built-in math coprocessor and 8K cache.

Memory

There are four memory options available; 4 MBytes, 8 MBytes, 16 MBytes, and 32 MBytes (32 MB is factory installed only). Four SIMM sockets are available. See Chapter 8, *Upgrades* for memory upgrade instructions.

Memory Map

The 2^{32} byte physical address space seen by the Intel486 occupies three areas:

1. Addresses between 0 and 1 MB, which are largely defined by the IBM PC/AT architecture.

2. Addresses between 1 MB and 256 MB, which largely depend on how much DRAM is installed in the EPC-5.

3. Addresses above 256 MB, which provide direct mapping to the VMEbus with a variety of address modifiers and byte orderings. See Chapter 6, *The VMEbus Interface* for more information about this feature.

Memory at addresses between 0 and 1 MB (0FFFFFh) is mapped as follows:

Range	Content
000000 - 09FFFF	DRAM (first 640 KB)
0A0000 - 0BFFFF	mapped to EXM interface; almost always used by a video controller as video RAM
0C0000 - 0C7FFF*	Write-protected DRAM containing video BIOS
0C8000 - 0DFFFF*	Uncommitted; mapped to EXM interface
0E0000 - 0EFFFF	User-mappable hardware window onto VMEbus
0F0000 - 0FFFFF	Write-protected DRAM containing BIOS

* 0C8000 - 0DFFFF may be used either as page frames (i.e. for Ethernet, etc.) or may be used by DOS as upper memory blocks if an EMM driver is installed or may be used for BIOS extensions..

For a 4 MB EPC-5, the extended memory address space is defined as

00100000	003FFFFF	3 MB DRAM extended memory
00400000	00FEFFFF	Uncommitted; mapped to EXM interface
00FF0000	00FFFFFF	Mapped to BIOS ROM for PC compatibility

For an 8 MB EPC-5, the extended memory address space is defined as

00100000	007FFFFF	7 MB DRAM extended memory
00800000	00FEFFFF	Uncommitted; mapped to EXM interface
00FF0000	00FFFFFF	Mapped to BIOS ROM for PC compatibility

For a 16 MB EPC-5, the extended memory address space is defined as

00100000	00FEFFFF	15296 KB DRAM extended memory
00FF0000	00FFFFFF	Mapped to BIOS ROM for PC compatibility

For a 32 MB EPC-5, the extended memory address space is defined as

00100000	00FEFFFF	15296 KB DRAM extended memory
00FF0000	00FFFFFF	Mapped to BIOS ROM for PC compatibility
01000000	01FFFFFF	16384 KB DRAM extended memory

For a 64 MB EPC-5, the extended memory address space is defined as

00100000	00FEFFFF	15296 KB DRAM extended memory
00FF0000	00FFFFFF	Mapped to BIOS ROM for PC compatibility
01000000	01FFFFFF	16384 KB DRAM extended memory

Note that since the EXM expansion interface has 24 address lines, some of the "uncommitted; mapped to EXM interface" address areas map repeatedly, or wrap-around, in the EXM interface's address space.

ROM and ROM Shadowing

The EPC-5 contains a 27C010 EPROM (32-pin 128K x 8). The EPROM is mapped into the top of the processor's 32-bit address space, and also just below the 16 MB boundary for PC/AT compatibility. The EPROM contains the PC BIOS, SCSI disk BIOS, self-test functions, and the setup screen program.

For best possible performance, the BIOS initialization software copies the ROM contents into DRAM (called shadowing) at addresses 0F0000-0FFFFF (also called the "F" page). The BIOS also searches for the existence of a video adapter containing a video BIOS (e.g., an EXM-13A). If a video BIOS is found, it is copied into the 0Cxxxx ("C" page) area of DRAM.

After copying into these areas, the BIOS write-protects them. Subsequent writes to these areas complete successfully but do not alter the data.

Battery

The battery powers the CMOS RAM and TOD clock when system power is not present. At 60°C, the battery should have a shelf life of over four years. In a system that is powered on much of the time and where the ambient power-off temperature is less than 60°C, the battery is estimated to have a life of 10 years.

The battery supplied with the EPC-5 is a Tadiran TL-5242/W or equivalent. It is mounted on the underside of the metal frame and connected to a header on the processor board. Should the battery fail, you may obtain and install a replacement.

Replacing the battery is a simple task. However, removing the battery **will** invalidate the CMOS setup parameters. It is recommended that all setup parameters be written down while the battery is still good.

Video Controllers

The EPC-5 can operate with or without a video controller (such as the EXM-13 or EXM-13A). The BIOS searches for an EXM having an EXM ID in the range E8h-EFh (a range reserved for video controllers). The search is done by EXM slot number, beginning at slot 0. If no EXM video adapter is found, the BIOS looks for a PC add-in card video controller in an EXP-AM Adapter module. The error message EXM CONFIGURATION ERROR may appear if the video controller EXM or the EXP-AM hasn't been configured via the setup screen.

In either case, the BIOS automatically initializes and uses the first one found.

If no video controller is present, the BIOS operates without one. Programs that use the standard operating system and BIOS character output functions can be run successfully (the output is ignored). However, programs that rely on specific video modes, that write directly into the video RAM, or that directly call video BIOS functions will fail.

Front Panel LEDs

The EPC-5 has five LEDs in the top left corner of the front panel. These LEDs are described below:

RUN — This LED is lit whenever the EPC-5's memory is being accessed. It first comes on at power-up and should remain lit as long as the system is running.

SYSFAIL — This LED is only active when this EPC-5 is jumpered to be the Slot-1 controller. It comes on whenever the system receives a hardware reset and remains on until the initial power-on self-tests have completed. It also comes on whenever the VMEbus SysFail line is asserted.

TEST — This LED is lit whenever the system is running its power-on self-test. This only occurs during a hardware reset.

MASTER — The Master LED is lit whenever the EPC-5 is accessing the VMEbus.

SLAVE — The Slave LED is lit whenever another master on the VMEbus is accessing the EPC-5's memory.

Resetting the EPC-5

There are a number of ways to reset (reboot) the EPC-5.

Power-off, Power-on

This causes all boards in the VMEbus to reset. The system runs the power-on self-tests and reboots the operating system.

Front-panel Reset button

The Reset button causes the EPC-5 to perform a hardware reset. The system runs the power-on self-tests and reboots the operating system.

Ctrl+Alt+Del

This keyboard sequence is called a "warm boot". The EPC-5 does not reinitialize all of the processor's hardware. The power-on self-test does not run. However, the operating system is reloaded.

VMEbus SysReset

The EPC-5 can be software-configured to respond or not respond to the VMEbus SysReset line. Asserting this bit causes a hard reset of the system. See bit 7 (SRIE), register 8144 on page 49 for details.

VMEbus Register Reset

The EPC-5 can also be reset by another master asserting the reset bit of a register mapped to the VMEbus. Asserting this bit causes a hard reset of the system. See bit 0 (RSTP), register 8144 on page 49.

EXM Expansion Interface

The EXM expansion interface is electrically similar to the PC/AT ISA (16-bit data) bus. In addition, it contains a signal -EXMID used for dynamic recognition and configuration of EXMs. EXMs respond to one or more I/O addresses in the range 100h - 107h only when their -EXMID line is asserted. EXMs are required to return a unique EXM ID byte in response to a read from I/O address 100h.

This ID byte is the same identification byte discussed earlier in Chapter 4, *Configuring the BIOS Setup* in the section on the *EXM Setup Menu*, page 25.

The EXM expansion interface is provided on rows A, C, and D of the EPC-5's 4-row DIN P2 connector. The subplane carries the EXM interface to other modules, such as to EXM modules and the EXP-MS/MX Mass Storage module. These EXM interface signals are not passed through to the VMEbus.

Further information on the EXM expansion interface, its connectors, and standards for building EXMs is available upon request.

6. The VMEbus Interface

This chapter describes the EPC-5 VMEbus interface as seen by a program. Users should avoid direct use of most of these facilities. Whenever possible, the VMEbus interface should be accessed through the EPConnect software, an easy-to-use, high-level interface that frees you from most machine-dependent considerations.

Connectivity

The EPC-5 module connects to the VMEbus J1 connector directly and uses all of the defined VMEbus lines except SERCLK, SERDAT, and +5V STDBY. Connection to the J2 connector is through the subplane's 4-row DIN connector B row. The only connections to the VME J2 backplane on the B row are power and ground, address lines A31-A24, data lines D16-D31. Pin 30, Row A, the VXI-defined module identification (MODID) line is also connected. Pin A30 is an input driving one gate input and an 825-ohm pull-down resistor. It may be disabled by removing the MODID jumper on the EPC-5. If necessary, see *Configuring the EPC-5*, page 3.

VMEbus System (Slot-1) Controller Functions

Every VMEbus system must have a System (Slot-1) Controller. The Slot-1 controller provides the following functionality:

- Serves as the bus arbiter (priority or round-robin)
- Drives the 16 MHz SYSCLK signal
- Starts the IACK daisy chain.
- Provides Bus Timer function

When configured as the Slot-1 controller, the EPC-5 detects and terminates data transfer bus timeouts. Once it sees either the DS0 or DS1 lines asserted, a counter is started. If the counter expires before both DS0 and DS1 are deasserted, the EPC-5 asserts the VMEbus BERR signal until both data strobes are deasserted. The duration of the VMEbus timeout counter is 100-120 µsecs. When the EPC-5 is configured as the slot-1 controller, this timeout cannot be disabled and the duration cannot be changed.

Although the EPC-5 provides the required timeout function for data transfer time-out, it does not provide the optional bus grant timeout. If another master has been granted permission to use the data bus but does not access (or relinquish) the data bus, the bus will be "hung" indefinitely.

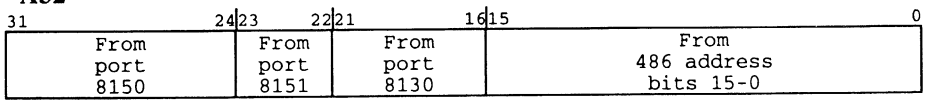
Concepts

Memory Map

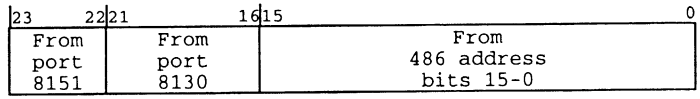
VMEbus accesses are available either by mapping a 64K segment of the VMEbus through the 0E0000-0EFFFF "E page" window or by direct mapping above 256 MB.

The following summarizes the source of the VMEbus address lines for accesses through the E page.

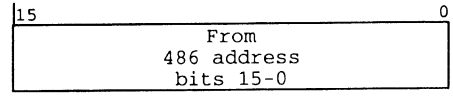
A32



A24



A16



It should be noted that the EPC-5 drives all 32 address lines even when performing an A24 or A16 access. Therefore, all the above registers (8150, 8151, 8130) should be set for every access using the E-page window. Make sure that those registers not directly supplying address lines are set to "FF" values in the appropriate bit positions.

Direct VMEbus Accesses

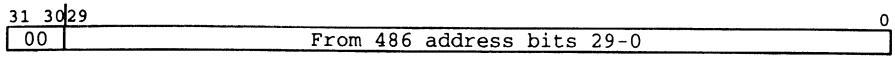
An alternate way to perform VMEbus accesses, providing that the EPC-5 is running in protected mode is to perform reads and writes at 486 addresses above 10000000h (256 MB). For instance, a 4-byte read at address 40000000h will result in a 4-byte VMEbus read access at address 00000000 with an address modifier specifying A32, supervisory data and no byte-swapping (little-endian mode).

With the EPC-5, addresses above 256 MB, with one exception for PC compatibility, map onto the VMEbus. When direct "protected-mode" addressing of A24 or A16 space, the high-order nibble is used to define the access mode and byte ordering. For A32 space, the high-order 2 bits define the access mode leaving 30 bits available for addressing. Thus, only the first 1 Gigabyte of VMEbus A32 space is directly addressable. All A24 and A16 space is directly addressable. The chart following shows how this direct mapping is used.

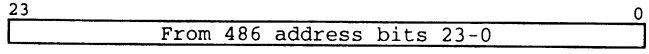
Address Range	Access Mode	Byte Order
1xxx0000 - 1xxxFFFF	VME A16 supervisory data	little endian
2x000000 - 3xFFFFFF	VME A24 supervisory data	little endian
40000000 - 7FFFFFFF	VME A32 supervisory data (mapped to VME 00000000-3FFFFFFF)	little endian
80000000 - BFFFFFFF	VME A32 supervisory data (mapped to VME 00000000-3FFFFFFF)	big endian
Cxxx0000 - DxxxFFFF	VME A16 supervisory data	big endian
Ex000000 - ExFFFFFF	VME A24 supervisory data	big endian
F0000000 - FFFEFFFF	Mapped to EXM expansion interface	
FFFF0000 - FFFFFFFF	486 upper ROM area	

When accessing the VMEbus in this manner, the source of the VMEbus address lines is defined below.

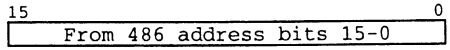
A32



A24



A16



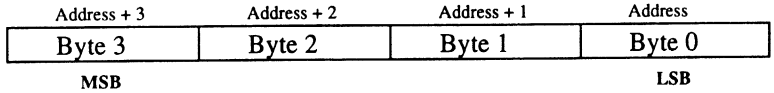
The main purpose of the direct VMEbus access mechanism, as opposed to the E-page mechanism, is for multitasking 32-bit operating-system environments, where multiple tasks need to make VMEbus accesses. Without this, the tasks would have to coordinate their use of the E-page mapping registers.

When using the EPC-5 this way to perform VMEbus accesses, one would typically set up the E-page window for interrupt acknowledge accesses. Also note that the direct access mappings do not cover the entire VMEbus A32 address range and do not provide all VMEbus-defined address modifier encodings, but one can use the E-page mechanism if needed to provide these.

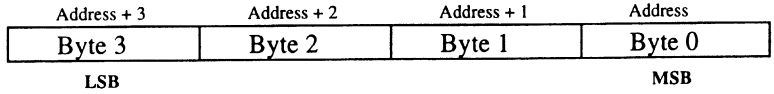
Byte Ordering

There are two fundamentally different ways of storing numerical values in byte locations in memory:

- Little endian, characteristic of Intel microprocessors, where the **least**-significant data byte (LSB) is stored in the lowest byte address



- Big endian, characteristic of Motorola microprocessors and the VMEbus environment in general, where the **most**-significant data byte (MSB) is stored in the lowest byte address



The EPC-5 contains programmable byte-swapping hardware to allow programs to read or write VMEbus memory in either byte order. When using the E-page to access the VMEbus, the order is selected by bit 5 (BORD) in the VME modifier register (8151). When using direct memory mapping, the order is address-range dependent (e.g., E0000000-E0FFFFFF accesses the A24 space with big endian byte ordering, and 20000000-20FFFFFF accesses the A24 space with little endian byte ordering).

When performing a single byte (D08) access, the byte order makes no difference. However, word (D16) or double-word (D32) accesses may require byte-swapping.

When little-endian is selected, bytes pass straight through unchanged. Little endian should only be used when reading or writing data between two Intel processor systems. The results of using little-endian byte ordering to transfer a double-word integer between an Intel processor and a Motorola processor are shown below.

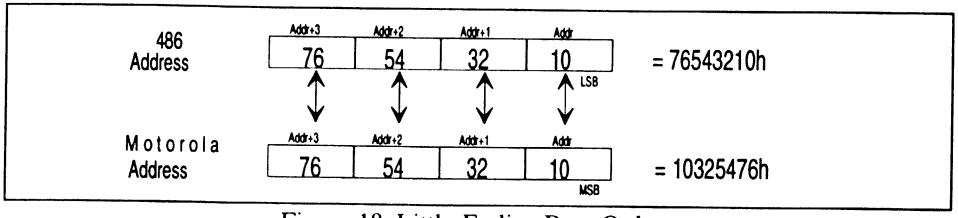


Figure 18. Little-Endian Byte Order.

Since the 486 processor uses Addr as the least-significant byte and the Motorola processor uses Addr as the most-significant byte, the processor receiving the data gets a "scrambled" value.

When big-endian is selected, the bytes are swapped between the 486 and VME. See the diagram below.

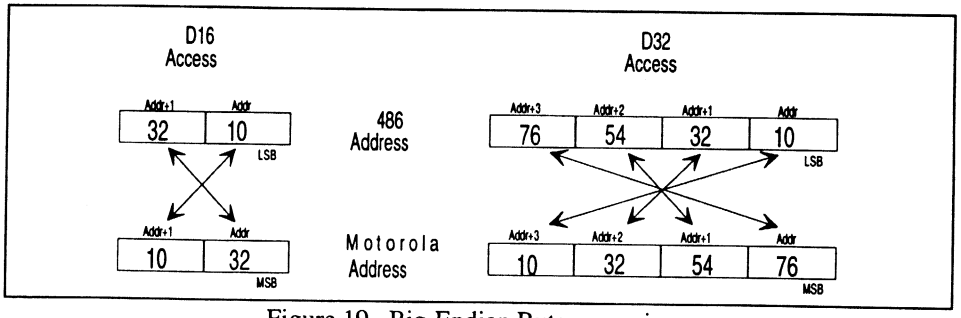


Figure 19. Big-Endian Byte-swapping.

When using big-endian byte ordering, care must be taken to assure that the VME address is aligned on a boundary; for D16 accesses the VME address must be on a word boundary (address evenly divisible by 2) and for D32 accesses the VME address must be on a double-word boundary (evenly divisible by 4). If this is not done, the results will be "scrambled" data. Although the VMEbus address must be boundary-aligned to match the data width (word or double-word), the 486 address does not need to be boundary-aligned.

Another consideration is the compiler being used. Some compilers produce two 16-bit accesses when a 32-bit access is desired. When this occurs, again the data will be "scrambled."

When transferring a 32-bit floating-point number, special care must be taken to assure that both processors use the same floating-point format; and that both systems expect the mantissa and exponent in the same byte locations. As long as this is correct, transferring a floating-point number will work correctly. Since transferring a 64-bit floating-point number is not supported in hardware, two 32-bit transfers must be used with little-endian byte order and then byte-swapping must be accomplished in software.

Byte swapping applies only to EPC-5 initiated (master) **accesses; it does not apply to slave accesses** (from other VMEbus masters to the EPC-5's DRAM).

The EPConnect Bus Manager software provides a means of selecting the byte ordering during memory-copy operations.

Slave Accesses from the VMEbus

When SLE (Slave Enable) in the status/control register (8145) is set, the EPC-5's dual-ported memory will respond to accesses from other VMEbus masters.

All types of VME accesses (reads, writes, and read-modify-writes of all lengths) are supported, except for block transfer cycles. The EPC-5 responds to supervisory, non privileged, program, or data access modes.

The amount of memory that will be dual-ported is limited to the first (lowest address) 4 MBytes in A24 space or all available memory in A32 space. In both cases, the slave memory's local (PC) address starts at Segment 0000, Offset 0000. This, of course, means that it is possible to overwrite the memory space occupied by the operating system. As such, care must be taken in writing to the EPC-5's memory.

When such an access is fielded by the EPC-5, the EPC-5's A24 or A32 base address is effectively subtracted from the VMEbus address value, and the result is treated as if the access came from the 486. However, note the following:

1. Any access that maps to local addresses 000A0000-000BFFFF, 000D0000-000EFFFF, to addresses mapped to the EPC-5's EXM expansion interface, and to addresses beyond the extent of the installed DRAM cause the EPC-5 to respond with BERR (bus error).

2. Write accesses to write-protected DRAM terminate normally (DTACK response), but with no effect on the DRAM.

Enabling of the EPC-5 as a slave and specification of the address space (A24 or A32) and the base address is controlled by the registers discussed in the following section, *Registers Specific to the EPC-5*. The easiest way to set up these registers is to do so via the BIOS setup screen.

Self Accesses Across the VMEbus

Since the EPC-5's DRAM can be mapped into the VMEbus A24 or A32 address space, the EPC-5 can access its DRAM in an alternate way - by generating VMEbus accesses to addresses mapped as the EPC-5's VME slave memory. This can be of use in multiple-processor systems where some of the EPC-5's DRAM is used as shared global memory; it means that the EPC-5 can access the global memory with the same addresses as used by other processors without needing to understand that the memory is actually on-board.

This ability is also useful in system checkout (i.e., checking operation of the backplane) and in giving an EPC-5 program the ability to view its memory in big endian format.

A24 and A32 slave accesses result in accesses to the on-board DRAM and never to the cache. Because the EPC-5's cache is a write-through cache, there is never a discrepancy between data in the cache and the DRAM. When a slave access results in a *write* into the DRAM, the EPC-5 automatically purges the cached entry, if it exists.

Given the above, another subtle use for the ability of the EPC-5 to access its own DRAM via a VMEbus access is selective purging of the cache. For instance, if the EPC-5 is mapped at address base 18000000h in the A32 space and a program is meant to purge location 0000AB00h from the cache, a read from 0000AB00h followed by a write of the read data back to 1800AB00h will accomplish the task.

Read-Modify-Write Operations

VMEbus RMW (read-modify-write) cycles can be performed through use of the LOCK instruction prefix with certain instructions. All of these instructions perform a read followed by a write. When such a read occurs that is mapped to the VMEbus, the EPC-5 treats it as the start of a VME RMW cycle. The next VME access from the CPU is treated as the write that terminates the RMW cycle. Keep in mind that accesses that cross a 32-bit boundary are actually performed as two accesses. For this reason, RMW accesses that cross a 32-bit boundary will not behave as expected.

accesses that cross a 32-bit boundary are actually performed as two accesses. For this reason, RMW accesses that cross a 32-bit boundary will not behave as expected.

The EPC-5 provides synchronization integrity in its local DRAM between accesses from the CPU into the DRAM and RMW VME accesses from other masters into the DRAM.

When a VMEbus slave read access occurs to the local DRAM, the EPC-5 watches the VMEbus data and address strobes to determine if the cycle is an RMW cycle. If it is, accesses by the CPU are held up until the terminating access of the RMW cycle occurs.

When the CPU performs a locked access (e.g., via an instruction using the LOCK instruction prefix) to the local DRAM or the cache, VMEbus slave accesses are held up until the last locked access completes.

One more case of interest is when the EPC-5 performs a locked access that results in a self access. These function correctly (i.e., as if the access was not a self access), providing that operating-system tables (e.g., page tables) that are accessed by the CPU by implicit locked accesses are not mapped into VME. This would only be a concern for user-written operating systems.

6

VMEbus Interrupt Response

When the EPC-5's Interrupt Generator register (815F) is used to assert an interrupt, the EPC-5 formulates a status/ID value that is transmitted on the bus as the response to a matching interrupt acknowledgeent cycle. The EPC-5 acts as both a D08(O) and D16 interrupter. For D08 interrupt acknowledge cycles, the status/ID value is the EPC-5's logical address (11111aaa, where aaa is the value of ULA as defined in port 814A). For D16 and D32 interrupt acknowledge cycles, the status/ID value consists of 16 bits. The upper eight bits are the upper half of the response register (the value in I/O port 814B) and the lower eight bits are the logical address.

Registers Specific to the EPC-5

Registers in the I/O space that are specific to the EPC-5 are defined below.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	I/O Port
M00FF	M000F	M000C	MEMSIZE			VME	A16	8104

VMEbus and Memory Controller Configuration

VMEbus Address bits 21-16						Res	Res	8130

VME A21-16 Address Register

1	1	1	0	1	1	0	0	8140

ID Register, lower

1	0	0	A32	1	1	1	1	8141

ID Register, upper

1	1	0	0	0	1	0	0	8142

Device Type Register, lower

0	Slave Size		1	1	1	1	1	8143

Device Type Register, upper

SRIE	RELM	ARBPRI		READY	PASS	NOSF	RSTP	8144

Status/Control Register, lower

SLE	MODID	SYSR	SYSF	ARBM	1	1	1	8145

Status/Control Register, upper

1	1	1	1	1	1	1	1	8146

Slave Offset Register, lower

0	0	0	1	1	SLAVE BASE			8147

Slave Offset Register, upper

1	1	1	1	1	1	1	1	8148

Protocol Register, lower

0	1	0	1	1	1	1	1	8149

Protocol Register, upper

LOCK	1	ABMH	1	1	ULA	814A

Response Register, lower

0	0	0	0	1	RRDY	WRDY	1	814B

Response Register, upper

RAM	814C

Message High Register, lower

RAM	814D

Message High Register, upper

RAM	814E

Message Low Register, lower

RAM	814F

Message Low Register, upper

VMEbus Address bits 31-24 (WA31-24)	8150

Message A31-24 Address Register

VME WA23-22	BORD	IACK	AM5	AM4	AM2	AM1	8151

VME Modifier Register

IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	MSGR	8152

VME Interrupt State Register

IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	MSGR	8153

VME Interrupt Enable Register

1	1	1	1	1	ACFA	BERR	SYSF	8154

VME Event State Register

1	1	1	1	1	ACFA	BERR	SYSF	8155

VME Event Enable Register

DONE	AS	DS0	DS1	1	1	(res.)	1	8156

Module Status/Control Register

1	1	1	1	1	INTERRUPT-OUT	815F

Interrupt Generator Register

Where a bit position has been described by a 0 or 1, the bit is a ROM bit, and writing to it has no effect. Unless otherwise noted below, all registers and bit values are readable and writeable.

VMEbus and Memory Controller Configuration (8104)

M00FF	M000F	M000C	MEMSIZE	VME	A16

This register controls the DRAM memory controller and certain aspects of BIOS write protection and the VMEbus interface. The bits in this register are cleared by an "AT reset" (that is, when the RESET button is pushed or power is applied to the system).

M00FF — Mode 00FF. If clear (0), reads of addresses in the range 00FF0000-00FFFFFF refer to BIOS ROM and writes in that range refer to cached DRAM (if it is present) or to the EXM expansion interface.

NOTE: If this bit changes from 1 to 0 without a concurrent processor reset, the 486 cache must be flushed.

M000F — Mode 000F. If clear (0), reads of addresses in the range 000F0000-000FFFFF refer to uncached DRAM and writes in this range refer to DRAM. If set (1), reads in this range refer to uncached DRAM and writes refer to the EXM expansion interface.

M000C — Mode 000C. If clear (0), reads of addresses in the range 000C0000-000CFFFF refer to the EXM expansion interface and writes in that range refer to DRAM. If set (1), reads is this range refer to uncached DRAM and writes refer to the EXM expansion interface.

MEMSIZE — Memory Size. These bits tell the memory controller how much SIMM DRAM memory is present on the EPC-5. This field is set by the BIOS when power is applied to the EPC-5, and will contain one of the following values: 001 = 4 MB, 010 = 8 MB, 111 = 16 MB, 110 = 32 MB.

VME — If set (1), this bit allows VME accesses through the DOS "E page" or through shared memory above 256 MBytes in protected mode. When clear (0), the VMEbus cannot be accessed. This bit is automatically set by the BusManager software when using EPConnect.

A16 — ROM A16. The 128K bytes of ROM on the EPC-5 are addressed through a 64K "window". If this bit is set (1), the high 64K of ROM is selected; if clear (0), the low 64K of ROM is selected.

Note: Except for the VME access bit, the bits in this register are manipulated by the BIOS. User software should manipulate only the VME access bit in this register.

VME A21-16 Address Register (8130)

VMEbus Address bits 21-16	Res	Res

When an access is performed by the EPC-5 in its "E page" (address range 0E0000-0EFFFF), the access is mapped onto the VMEbus. The least-significant sixteen of the VME address bits are provided directly (from the 486), and the remaining 8 (for an A24 access) or 16 (for an A32 access) bits must come from somewhere else. Six of them come from this register. Bit 7 of this register is used as VME address bit 21, bit 6 as VME address bit 20, ..., and bit 2 as VME address bit 16.

The two low-order bits are reserved RAM bits. On writes, assign them the value 0.

For compatibility with EPC-1, this register is aliased at I/O port addresses 8132, 8134, and 8136.

ID Register (8140 & 8141)

1	1	1	0	1	1	0	0	Lower
1	0	0	A32	1	1	1	1	Upper

This read-only register adheres to the VXIbus specification. It defines the EPC-5 as a message-based device and the manufacturer as RadiSys Corporation.

A32 If set (1), the EPC-5's DRAM is mapped into the VMEbus A32 address space. If clear, the DRAM is mapped into the A24 address space. This read-only bit is influenced by the value stored in the SLAVE-SIZE field of the next register.

Device Type Register (8142 & 8143)

1	1	0	0	0	1	0	0	Lower
0	Slave Size		1	1	1	1	1	Upper

This register adheres to the VXIbus specification. Only bit 6 is writeable. Bit 5 is automatically set to match bit 6. If bit 6 is set, the value of the register is 7Fh and the A32 bit in the previous register is 1. This denotes that the EPC-5 responds to a 16 MB range in the A32 space.

If bit 6 is clear, the value of the register is 1Fh and the A32 bit in the previous register is 0. This denotes that the EPC-5 responds to a 4 MB range in the A24 space.

The remaining ROM bits define the EPC-5 as having a model code of 4036.

Status/Control Register (8144 & 8145)

SRIE	RELM	ARBPRI		READY	PASS	NOSF	RSTP	Lower
SLE	MODID	SYSR	SYSF	ARBM	1	1	1	Upper

This register adheres to the VXIbus specification and also contains EPC-5 specific bits.

SRIE SYSRESET input enable. If set, assertion of VME SYSRESET generates a reset of the EPC-5. One use of this bit is having EPC-5 software reset other VME devices (via bit SYSR) without resetting the EPC-5.

RELM Bus release mode. If set, the bus release mode is ROR (release on request); otherwise it is the VXI RONR "fair requester" mode (request on no request). Altering this bit via the VME-mapped location of this register has no effect.

ARBPRI Arbitration priority. This defines the level at which the EPC-5 will arbitrate for the VMEbus. 11 means 3, 10 means 2, 01 means 1, 00 means 0. Like for RELM, altering this field via the VME-mapped location of this register has no effect.

READY This is a RAM bit defined by the VXI specification. In a VXIbus software environment, if READY=1 and PASS=1, the EPC-5 is ready to accept VXI-defined messages. *In the EPC-1 and in early versions of the EPC-3 manual, this bit was named EXTE. Its implementation hasn't changed, but it was renamed to correspond to the renaming of the bit in revision 1.3 of the VXIbus specification.*

PASS If set (1), the EPC-5 has completed its self test successfully. If this bit is clear, the Test LED on the EPC-5 front panel is lit.

NOSF SYSFAIL inhibit. If set, the EPC-5 cannot assert the VMEbus SYSFAIL line.

RSTP Reset EPC. Setting this bit resets the EPC-5.

SLE Slave enable. If set, the EPC-5 responds to certain A24 or A32 accesses on the VMEbus.

MODID This readable bit is connected to pin 30 in row A of the VMEbus P2 connector. If clear (0), it denotes that the pin is being pulled high. (This is used in VXI systems for module identification.)

SYSR SYSRESET. The EPC-5 asserts the VME SYSRESET line while this bit is 1. When using this bit, it is the software's responsibility to ensure that the VME-specified minimum assertion time of SYSRESET is met.

SYSF SYSFAIL. The EPC-5 asserts the VME SYSFAIL line while this bit is 0. (The polarity of the bit is reversed from that of SYSRESET so that an EPC-5 reset - which clears this bit - causes SYSFAIL to be asserted until the BIOS stores a 1 in this bit.)

ARBM Arbitration mode. This bit is pertinent only if the EPC-5 is jumpered to be the VMEbus system controller. If set, the EPC-5 is a priority arbiter; otherwise it is a round-robin arbiter. Like for RELM, altering this field via the VME-mapped location of this register has no effect.

Slave Offset Register (8146 & 8147)

1	1	1	1	1	1	1	1	Lower
0	0	0	1	1	SLAVE BASE			Upper

If A32 and SLE are set, the value in port 8147 defines the base address of the EPC-5's memory in the VMEbus A32 address space. This register can hold the values 18 - 1F, which correspond to the base addresses 18000000 - 1F000000.

If A32 is clear and SLE is set, the two low-order bits of SLAVE BASE define the base address of the EPC-5's memory in A24 as follows: 00 - 000000, 01 - 400000, 10 - 800000, 11 - C00000.

Protocol Register (8148 & 8149)

1	1	1	1	1	1	1	1	Lower
0	1	0	1	1	1	1	1	Upper

This read-only register is defined by the VXIbus specification. In VXI systems, it defines the EPC-5 as being a servant and commander, having no signal register, being a bus master, and not providing fast handshake mode.

Response Register (814A & 814B)

LOCK	1	ABMH	1	1	ULA			Lower
0	0	0	0	1	RRDY	WRDY	1	Upper

With the exception of LOCK, this register is defined by the VXIbus specification. It contains control bits associated with the message registers.

LOCK If set, the message register can be locked for the sending of a message. If clear, the message register has been locked.

ABMH This bit is cleared when the message high register is read or written. It serves as a location monitor for determining whether a message is 16 or 32 bits in length.

ULA Unique logical address. This determines the base of the registers in the VMEbus A16 space. 0 denotes FE00, 1 denotes FE40, 2 denotes FE80, 3 denotes FEC0, 4 denotes FF00, 5 denotes FF40, 6 denotes FF80, and 7 denotes FFC0.

RRDY Read ready. As defined by VXI, a 1 denotes that the message registers contain outgoing data to be read by another device. RRDY is cleared when the message low register is read.

WRDY Write ready. If set, the message registers are armed for an incoming message. When a write occurs into the message-low register, WRDY is cleared and the MSGR interrupt condition is asserted.

When the response register is read from the VMEbus, the current value of the register is read, and then LOCK is cleared. The protocol for sending a message to the EPC-5, if there are multiple potential senders, is the following. The sender first reads the response register. If both WRDY and LOCK are 1, the sender may then proceed to send the message. For a 16-bit message, the sender writes into the message-low register. For a 32-bit message, the sender writes first into the message-high register and then the message-low register.

Message High Register (814C & 814D)

RAM	Lower
RAM	Upper

This register is an extension of the following register for 32-bit messages. An access to this register clears flag ABMH in the response register.

Message Low Register (814E & 814F)

RAM	Lower
RAM	Upper

This register is typically used as an incoming message register by doing a D16 write into it from the VMEbus (this register, as are many others, are mapped into the VMEbus A16 address space, as discussed later).

VME A31-24 Address Register (8150)

VMEbus Address bits 31-24 (WA31-24)

This register is one of several that supply the VMEbus address bits when the EPC-5 makes an access in its "E page." This register supplies VME address bits A31-A24.

VME Modifier Register (8151)

VME WA23-22	BORD	IACK	AM5	AM4	AM2	AM1

This register is also used when the EPC-5 makes an access through its E page to the VMEbus. Bits 7 and 6 provide VME address bits A23 and A22, respectively. Bits 3-0 define the value placed on the associated VMEbus address-modifier lines. Register bits are not defined for the VMEbus address-modifier AM3 and AM0 lines since, for all defined address-modifier values in the VMEbus specification, AM3 is 1 and AM0 is the inverse of AM1. Therefore these two bit values are generated by hardware. Note that because AM3 and AM0 are hardware generated, the EPC-5 does not support user-defined address-modifiers.

BORD Byte order. This bit controls the ordering of data bytes for D16 and D32 VMEbus accesses. If 0, the bytes are transmitted in little endian (Intel) order; if 1, byte-swapping hardware transmits the bytes in big endian (Motorola) order. Refer to the previous section in this chapter on byte ordering.

IACK This bit, when set, is used to define the VMEbus access as an interrupt acknowledge cycle. The interrupt being acknowledged must be encoded by software as a value on VME address lines A1-A3.

VME Interrupt State Register (8152)

IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	MSGR

This read-only register defines the state of the VMEbus and message interrupts.

IRQx If clear (0), the associated VMEbus interrupt line is asserted.

MSGR If clear (0), a message interrupt is being signaled. MSGR is clear if both bits RRDY and WRDY in the response register are clear.

VME Interrupt Enable Register (8153)

IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	MSGR

This is a mask of the interrupt conditions in the interrupt state register. A 1 denotes that the corresponding interrupt is enabled. If any bit in this register is a 1 and the corresponding bit in the interrupt state register is a 0, the EPC-5 IRQ10 interrupt is asserted. Software may then examine the interrupt and event state registers to determine the cause.

VME Event State Register (8154)

1	1	1	1	1	ACFA	BERR	SYSF

Similar to the interrupt state register, this register defines additional conditions that may result in an IRQ10 interrupt. If the bit is 0, the condition is present.

ACFA VMEbus ACFAIL is asserted.

BERR An access from the EPC-5 to the VMEbus was terminated with a BERR (bus error).

SYSF VMEbus SYSFAIL is asserted.

All bits are read-only except BERR. BERR is a sticky bit that is cleared whenever an access from the EPC-5 is terminated by a bus error, and remains clear (0) unless changed by software (by writing any value to this register).

VME Event Enable Register (8155)

1	1	1	1	1	ACFA	BERR	SYSF

This is a mask of the interrupt conditions in the event state register. A 1 denotes that the corresponding event is enabled as an interrupt. If any bit in this register is a 1 and the corresponding bit in the event state register is a 0, the EPC-5 IRQ10 interrupt is asserted. Software may then examine the interrupt and event state registers to determine the cause.

Module Status/Control Register (8156)

DONE	AS	DS0	DS1	1	1	(res.)	1

This register contains miscellaneous status and control bits.

DONE This read-only bit is 0 whenever the EPC-5 has a VMEbus access outstanding. It is used for determining when a pipelined VMEbus write is complete.

AS This read-only bit is 1 whenever the VMEbus AS (address strobe) signal is asserted. It may be used for bus monitoring.

DS0 This read-only bit is 1 whenever the VMEbus DS0 (data strobe) signal is asserted. It may be used for bus monitoring.

DS1 This read-only bit is 1 whenever the VMEbus DS1 (data strobe) signal is asserted. It may be used for bus monitoring.

(res.) This bit should always be set (1).

The EPC-3 contains two additional bits in this register - ENMI and DEAD - for breaking deadlock situations on its dual-port DRAM. These situations cannot exist in the EPC-5 so the signals are not implemented.

VME Interrupt Generator Register (815F)

1	1	1	1	1	INTERRUPT-OUT

This register is used to assert one of the VMEbus interrupt signals. If the INTERRUPT-OUT bits are zero, no interrupt line is asserted by the EPC-5. If lower three bits are set to 001, VMEbus IRQ1 is asserted. If set to 010, VMEbus IRQ2 is asserted, and so on. If and when an interrupt acknowledge cycle is sent to the EPC-5, the INTERRUPT-OUT bits are cleared.

You can also deassert a previously asserted interrupt by writing 0 into the register.

VMEbus Mapped Registers

The EPC-5 follows the lead of the VXIbus specification in defining a standard set of configuration registers that are mapped into the VMEbus A16 space and thus accessible by other VMEbus modules. These registers are 16-bit registers occupying

64 bytes of A16 space at a base address defined by the EPC-5's logical address. The base address is

1111 111a aa00 0000

where aaa is the value of the ULA field in the response register at I/O port 814A.

The VME-mapped registers are a subset of those defined previously as I/O ports in the EPC-5. The registers are dual-ported in that they are accessible both from VME and from within the EPC-5 as ports in its I/O space. The VME mapped registers are defined below.

Offset from ULA	Upper byte	Lower byte
0	ID (8141)	ID (8140)
2	Device type (8143)	Device type (8142)
4	Status/control (8145)	Status/control (8144)
6	Slave offset (8147	Slave offset (8146)
8	Protocol (8149)	Protocol (8148)
A	Response (814B)	Response (814A)
C	Message high (814D)	Message high (814C)
E	Message low (814F)	Message low (814E)

The registers occupy the first 16 bytes of the 64-byte space; the remainder of the space is undefined. (Actually, the registers are mapped into each 16-byte chunk of the 64-byte space.)

Reads and writes of the registers from VME and as I/O ports have identical results and effects except for the following:

1. Changing the RELM, ARBPRI, and ARBM fields of the status/control register from VME will appear to have changed the fields (i.e., if the register is then read), but the new values will not effect the EPC-5's bus-control logic. To use these fields for their intended purpose, they must be set by I/O port accesses.
2. A read of the response register from VME clears the LOCK bit (immediately after the current value of the response register is returned).

Register State after Reset

A hardware reset of the EPC-5 (not a keyboard CTRL+ALT+DEL reset) clears all of the register bits to 0, except for RELM, ARBM, ARBPRI, and the registers at ports 8130, 8150, and 8151, which may be in an undefined state. (All bits, however, are cleared by a power-on reset.) However, this may not be apparent because the BIOS initialization sequence then reinitializes values in these register fields, largely as a result of the non-volatile configuration information specified in the setup screen.

The BIOS clears the interrupt enable and event enable registers.

Supported Address Modifiers

2Dh	A16 supervisor
39h	A24 non-privileged data
3Ah	A24 non-privileged program
3Dh	A24 supervisor data
3Eh	A24 supervisor program
09h	A32 non-privileged data
0Ah	A32 non-privileged program
0Dh	A32 supervisor data
0Eh	A32 supervisor program

Low-Level Programming the VMEbus Interface

It is recommended that rather than performing accesses in this low-level hardware dependent form, the Bus Manager component of the EPConnect software package be used instead.

VMEbus Accesses

Two examples are given here including both a verbal description and the Microsoft C source code for performing VMEbus accesses through the "E" page.

Example #1 performs a 16-bit read from the VMEbus A16 space.

1. Set the VME access bit in Register 8104.

2. Determine the correct address modifier for A16 supervisory access (2Dh)

3. The unused address lines A31-A16 may float when not being used. Registers 8150 and 8130 must be set so that each line is a 1.

   Set register 8130 to FCh and register 8150 to FFh.

4. Set the access mode in the VME Modifier Register (8151) as follows:

VME WA23-22	BORD	IACK	AM5	AM4	AM2	AM1

(Note that register bits are not defined for the VMEbus address modifier lines AM3 and AM0 since, for all defined address modifier values in the VMEbus specification, AM3 is 1 and AM0 is the inverse of AM1. Therefore these two bit values are generated by hardware.)

Bits 7 & 6 Since the A16 space does not use VMEbus address lines A23 & A22, set these values to 1.

VME WA 23-22 = 11

Bit 5 Set the byte order to "little endian".

BORD = 0

Bit 4 Clear the IACK bit so this is not an interrupt acknowledge cycle.

IACK = 0

Bits 3-0 Use the address modifier (in binary form) to determine the appropriate values for these bits. 2Dh = 00**10**1**10**1b

Bit 3 (Address Modifier bit 5) = 1
Bit 2 (Address Modifier bit 4) = 0
Bit 1 (Address Modifier bit 2) = 1
Bit 0 (Address Modifier bit 1) = 0

Thus, 8151 should be set to 1100 1010 or CAh.

5. Map the address.

Add the A16 address to the "E page" address

Addr ← E0000000 + A16 address

6. Read the data.

Data ← value pointed to by Addr

Microsoft C code for Example 1 -

```
#define WORD  unsigned short
#define LWORD unsigned long

WORD addr;    /* 16-bit A16 address */
WORD data;
WORD far * wptr;

outp(0x8104,(inp(0x8104)|2));          /* set VME access bit */
outp(0x8130,0xFC);
outp(0x8150,0xFF);
outp(0x8151,0xCA);  /* Set address modifier to A16 supervisory access */
wptr = (WORD far *) (0xE0000000L + addr);
data = *wptr;       /* Read through window */
```

Example #2 performs a byte (8-bit) write into the VMEbus A32 space. Here the upper 16 bits of the VME address need to be stored in the appropriate registers.

1. Set the VME access bit in Register 8104.

2. Set register 8150 with the value corresponding to the 8 high-order address bits.

VMEbus Address bits 31-24 WA31-24

3. Determine the correct address modifier for A32 supervisory access.

4. Calculate the value and set register 8151 as follows:

VME WA23-22	BORD	IACK	AM5	AM4	AM2	AM1

Bits 7 & 6	VME address bits 23-22
Bit 5	BORD = 0
Bit 4	IACK = 0
Bits 3-0	Bit 3 (Address Modifier bit 5)
	Bit 2 (Address Modifier bit 4)
	Bit 1 (Address Modifier bit 2)
	Bit 0 (Address Modifier bit 1)

5. Set register 8130 with the value corresponding to bits 21-16 of the VMEbus address with the two low order bits of the register set to 0.

VMEbus Address bits 21-16	Res	Res

6. Map the address.

7. Write the data

Microsoft C code for Example 2 -

```
LWORD addr;   /* 32-bit A32 address */
BYTE  data;
BYTE  far * wptr;


outp(0x8104,(inp(0x8104)|2));                  /* set VME access bit */
outp(0x8150,(WORD)(addr >> 24));            /* A31-A24 */
outp(0x8151,2 | (((addr << 8) >> 30) << 6));
  /* A23-A22 and address modifier for A32 supervisory data access */
outp(0x8130,(WORD)((addr << 10) >> 24);  /* A21-A16 */
wptr = (BYTE far *) (0xE0000000L + (addr & 0X0000FFFFL));
*wptr = data;                              /* Write through window */
```

The success of the access can be checked either by enabling BERR as an interrupt or by looking at the BERR bit in the event state register (8154) after each access. Since writes are pipelined, software that looks at the BERR bit should first wait until the DONE bit is set.

Low-Level Handling of VMEbus Interrupts

The following is a description of how VMEbus interrupts (IRQ1-IRQ7), VXIbus message interrupts and error interrupts (BERR, ACFAIL, WDTG, etc.) should be handled on the EPC-5. Note that, in general, the use of EPConnect is highly recommended to handling interrupts.

- Enable the appropriate registers (VME Interrupt enable (8153) and VME Event enable (8155) registers) to allow the interrupts you want to respond to.

- Enable IRQ10 on the EPC's equivalent of the 8259 interrupt controller.

- A VXIbus message interrupt is generated when a master (this EPC-5 or another master) writes to the Message Low register (16-bit) or the Message High and Message Low registers (32-bit) from the VMEbus. A message interrupt does not occur when the EPC-5 writes to it's own message register(s) from the PC I/O space.

- Keep in mind that while PC/AT interrupts are edge sensitive, VMEbus interrupts are level sensitive. As such, you must ensure that

  1) The 8259 interrupt controller is enabled to capture interrupts before a VMEbus interrupt occurs (otherwise VMEbus interrupts will be totally missed) and

  2) You must handle all pending VMEbus interrupts before returning from the interrupt handler.

- When an interrupt occurs, first acknowledge the interrupt to the PC/AT 8259 interrupt controllers by sending both interrupt controllers an End-of-Interrupt (EOI).

- You must make sure that your interrupt handler code is not re-entered while dispatching interrupts. Either all interrupts should be disabled or IRQ10 should be masked after doing the EOI to the interrupt controller. Remember to re-enable them prior to leaving the interrupt handler.

- If you are using DOS, you may need to switch to an internal stack. This may or may not be necessary in other environments and applications. You should also store the state of the VMEbus (i.e., current byte ordering, bus mappings and address modifiers) if you expect the state to change. Be sure to restore the state before leaving the interrupt handler.

Start of Loop

- Determine the source of the interrupt or event. This can be done by reading the VME Interrupt State register which should be ANDed with the VME Interrupt Enable register. As described above, the VME Event State register and VME Event enable register may also be potential sources for the generation of IRQ10. Keep in mind that all pending interrupts must be handled.

- If the interrupt is a VMEbus interrupt 1-7;

  Acknowledge the interrupt to the VMEbus device generating the interrupt as follows:

  1- Set the IACK bit in the VME Modifier register

  2- Establish a byte-ordering for the status/ID to be read. Whether this is an 8-bit or 16-bit read is dependent on the card issuing the interrupt

  3- The address modifiers and transfer length are dependent on the hardware generating the interrupt.

  4- Perform a read of the VMEbus where the address being read reflects the interrupt level being responded to. Address lines A3-A1 must reflect the interrupt level in binary form. Multiply the interrupt level by 2 and use that as the address of the read operation.

  5- After the read operation, clear the IACK bit in the VME Modifier register.

- If the interrupt is a VXIbus message interrupt, the interrupt is acknowledged and cleared by reading the appropriate register(s), followed by setting the WRDY bit in the VME Response register.

- Call your interrupt handling routine.

- Upon returning from the interrupt handling routine, go back to the beginning of the loop until no more interrupts are active. In other words, you must handle all other active interrupts. This includes all other interrupts and errors which come in prior to calling the interrupt handling routine as well as any new interrupts and errors which may occur during this process. Only when all interrupts and error conditions are handled may you return from the overall interrupt handler. Again, if you miss any interrupts or errors, no other interrupts or errors will be recognized.

NOTES

7. Connectors

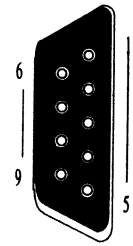
This chapter specifies the details of the connectors on the EPC-5. Please note, however, that all the connectors adhere to existing standards. The EXM expansion interface connectors are not defined here; their definition is available upon request. Connectors on EXMs and the EXP-MS/MX are described in the separate manuals for those products.

All but the battery and speaker headers are on the front panel. Pins are labeled from the point of view of looking into the front of the connector on the EPC-5.

Serial Ports

The COM1 and COM2 serial ports are DB-9 DTE connectors defined in the following table.

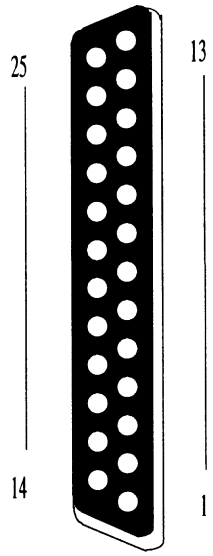
Pin	Signal	Pin	Signal
1	Carrier detect	6	Data set ready
2	Receive data	7	Request to send
3	Transmit data	8	Clear to send
4	Data terminal ready	9	Ring indicator
5	Signal ground		



Parallel Port

The DB-25 LPT1 parallel port connector is an Output-Only device defined as:

Pin	Signal	Pin	Signal
1	Strobe	14	Auto line feed
2	DB0	15	Error
3	DB1	16	Initialize printer
4	DB2	17	Select in
5	DB3	18	Signal ground
6	DB4	19	Signal ground
7	DB5	20	Signal ground
8	DB6	21	Signal ground
9	DB7	22	Signal ground
10	Acknowledge	23	Signal ground
11	Busy	24	Signal ground
12	Paper end	25	Signal ground
13	Select		

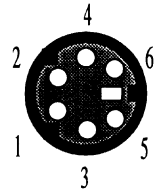


7

Keyboard

The keyboard connector is a 6-pin DIN defined as:

Pin	Signal	Pin	Signal
1	Data	4	+5V
2	not used	5	Clock
3	Ground	6	not used



Speaker Header

The speaker header is located on the EPC-5 circuit board and is defined as:

Pin	Signal	Pin	Signal
1	Reference voltage	2	Speaker tone

Battery Header

The battery header is located on the EPC-5 circuit board and is defined as:

Pin	Signal	Pin	Signal
1	VBATT	3	Ground
2	(key)	4	Ground

NOTES

8. Upgrades

☛ **DO NOT HANDLE THE EPC-5 MODULE UNLESS YOU ARE IN A STATIC-FREE ENVIRONMENT.**

Memory

The EPC-5 can be configured for various memory sizes. Not all are field-upgradable. The EPC-5 SIMMs work in pairs. When configuring an EPC-5 for use with 2 SIMMs, the 2 SIMMs should be placed in sockets 1 and 3. Sockets 2 and 4 should remain empty. See the figure below for SIMM socket locations.

The 25 Mhz, 33 MHz, and 66 MHz EPC-5 memory configurations use SIMMs with the following specifications:

- 72 pin
- fast page mode
- 80 nanosec. (or better)
- single-sided

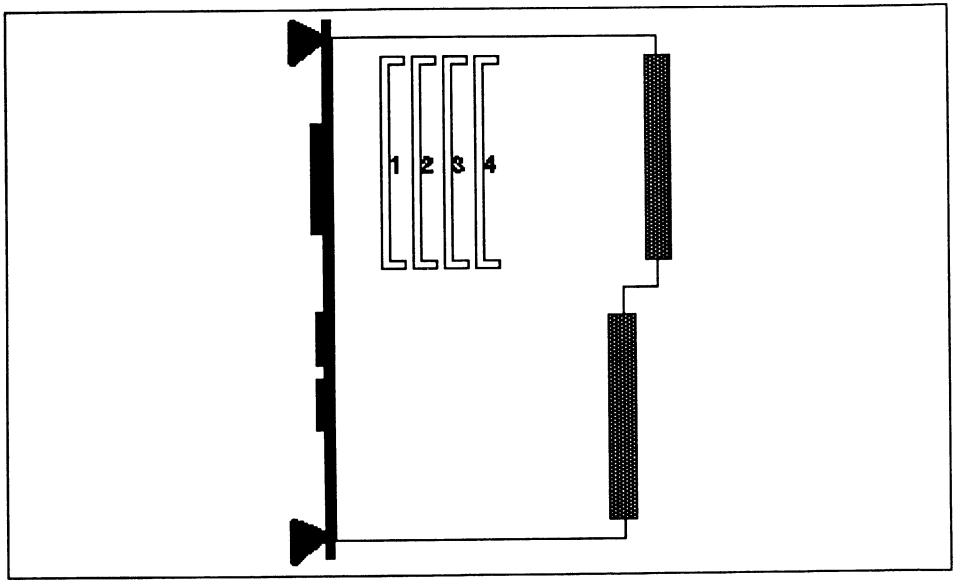
For 4 MB:	Use 4 each 256K x 36 SIMMs For example, Toshiba THM362500ASG-80 (RadiSys P/N 70-0032)
For 8 MB:	Use 2 each 1M x 36 SIMMs For example, Toshiba THM361000ASG-80 (RadiSys P/N 70-0042)
For 16 MB:	Use 4 each 1M x 36 SIMMs For example, Toshiba THM361000ASG-80 (RadiSys P/N 70-0042)
For 32 MB:	Not field upgradable unless using two 60 nanosecond SIMMs

The 100 MHz EPC-5 memory configurations use SIMMs with the following specifications:

- 72 pin
- fast page mode
- 60 nanosec. (or better)
- single-sided

For 8 MB:	Use 2 each 1M x 36 SIMMs RadiSys P/N 70-0074
For 16 MB:	Use 4 each 1M x 36 SIMMs. RadiSys P/N 70-0074
For 32 MB:	Use 2 each 4M x 36 SIMMs. RadiSys P/N 70-0075
For 64 MB:	Use 4 each 4M x 36 SIMMs. RadiSys P/N 70-0075

The SIMMs used in the 100 Mhz EPC-5 are custom-built for RadiSys, and at the time of this manual's printing, were not publicly available. Contact RadiSys Technical Support for more information.



After upgrading the memory, reboot the system. The error message "`Memory Size error - run setup`" will display after the power-on self-test completes. Press CTRL+ALT+ESC to enter the Main Setup Menu. Verify that the top line of this screen shows the correct amount of memory. Press F10 to save and F5 to confirm and reboot. The system will reboot and no error messages should be displayed.

9. Troubleshooting & Error Messages

Troubleshooting

This section deals with problems that you may encounter that do not provide an error message. If an error message is displayed, see the next section of this chapter, *Common Error Messages*.

Symptoms	Possible cause(s)	Solution
System appears to boot (evidenced by RUN LED being on, floppy and hard disk being accessed) but provides no video.	Video adapter not fully seated in subplane.	Remove the video adapter. If the subplane is secured to the VMEbus backplane by retaining screws, verify that the subplane is not warped from over tightening the screws. Reinsert the video adapter and verify seating into the subplane.
	Monitor or cable problem.	Verify that the cable pins are not bent and the cable is fully seated in the video adapter. If necessary, try the monitor on another system to verify that the monitor is good.
	Video adapter failure.	Call RadiSys Technical Support.
	Subplane failure.	Call RadiSys Technical Support.
	EPC-5 cannot talk to EXM expansion interface.	Call RadiSys Technical Support.

Symptoms	Possible cause(s)	Solution
System fails at power-up - will not run power-on self-test.	The system is not getting power.	Check the backplane and verify that +5V power is good. Verify that the subplane is fully seated in the VME backplane and the EPC-5 is fully seated in the subplane.
	Hardware failure.	This cannot be diagnosed in the field. Call RadiSys Technical Support.
Serial port(s) do not work.	Bad power.	Verify that backplane +12V and -12V are good.
	Interrupt conflicts	An EXM module is using the same interrupts as COM1 and/or COM2. Verify that no other card in the EPC-5 subsystem is using IRQ3 or IRQ4.
	Port hardware failure.	Call RadiSys Technical Support.
System hangs during boot process (Master LED on; RUN LED off)	VMEbus has no Slot-1 controller providing bus timeout.	You are probably loading an expanded memory manager (for example, EMM386.EXE) in your CONFIG.SYS file. This can cause the system to hang if - there is no Slot-1 controller - the Slot-1 controller is not providing the proper bus timeout - the Bus Grant jumpers are not installed.

System will not talk across VMEbus.	The VMEbus backplane may not be jumpered correctly.	See the section *Installing the VMEbus Backplane Jumpers*, on page 6.
	More than 1 master may be set to provide Slot-1 functions.	Make sure that only 1 system is configured as the Slot-1 controller and that it is the left-most system in the chassis.
	EPC-5 or subplane may have bent pins.	Remove the EPC-5 and the subplane and verify that no pins are bent. Then reinsert the subplane and the EPC-5.
	VMEbus interface failure.	Call RadiSys Technical Support.

Common Error Messages

This section contains a summary of error and warning messages alphabetized by message text. These are messages generated by the BIOS and MS-DOS that may be related to your hardware configuration.

BAD OR MISSING COMMAND INTERPRETER

Problem: The DOS operating system cannot find the Command line interpreter.

Solution(s): Either COMMAND.COM is not present at the specified (or default) directory level of the boot disk or the "SHELL=" statement in your CONFIG.SYS lists the file incorrectly (wrong directory or misspelled).

CMOS CHECKSUM INVALID

Problem: Something in the nonvolatile CMOS RAM is incorrect.

Solution(s): Run the BIOS setup program to determine what is wrong, and correct it. If the error occurs repeatedly, the EPC-5's battery has failed.

CMOS RAM ERROR, CHECK BATTERY / RUN SETUP

Problem: Something in the nonvolatile CMOS RAM is incorrect.

Solution(s): Run the BIOS setup program to determine what is wrong, and correct it. If the error occurs repeatedly, the EPC-5's battery has failed.

DISK BOOT FAILURE, INSERT SYSTEM DISK AND PRESS ENTER

Problem: No boot disk could be found.

Solution(s): This could occur in several different ways.

Your hard disk may not have been partitioned into logical drive(s). PCs look for logical drives to boot from. Hard disks are physical drives; partitions are logical drives.

Your BIOS setup screen has all disks disabled, or if your hard disk is disabled and no floppy diskette is inserted in the A: drive. Run the BIOS setup program and verify that all disk parameters are correct. If they are, insert a bootable floppy disk in the A: drive and press enter. If a hard disk is present, verify that it is properly partitioned and formatted as a system disk and one partition is set active.

DISKETTE DRIVES OR TYPES MISMATCH ERROR - RUN SETUP

Problem: The floppy diskette(s) installed in the system do not match the configuration information listed in the BIOS setup screen. This may be due to incorrect entries in the BIOS setup screen or one or both drives may not be responding at power-up.

Solution(s): Press CTRL+ALT+ESC to run the BIOS setup program. Make sure the BIOS setup entries relating to floppy drives correctly reflect the attached floppy drives. If you are using the EXP-MS/MX module, drive A should be set to "1.4M". If no second floppy drive is attached, set drive B to NONE. If you have no floppy drives, both drive A and drive B should be set to none.

Also, verify that all floppy drives are firmly connected (via subplane or ribbon cable) and that each drive has power.

If you are using an external floppy drive via a front panel connector, verify that the end of the ribbon cable is not shorting to the front panel and pin 1 on the front panel connector is connected to pin 1 on the drive.

ERROR INITIALIZING HARD DISK 0

Problem: The IDE disk controller for drive C cannot be initialized.

Solution(s): If you are using an EXP-MX mass storage module, ensure that the module is fully seated in the subplane and that the +5V and +12V LEDs indicate that the module has power.

If you are using the EXM-9 to cable to an external disk, make sure that you have power to the disk, the ribbon cable is good and correctly oriented, and that the end of the ribbon cable is not shorting to the front panel of the EXM-9.

If you are not using an IDE drive, press CTRL+ALT+ESC to enter the BIOS setup program. Press F3 to enter the Fixed disk menu. Change the drive type to match the device being used.

EXM CONFIGURATION ERROR

Problem: The EXMs installed (or not installed) do not match the configuration information in the CMOS Setup.

Solution(s): Press CTRL+ALT+ESC to run the BIOS setup program. Press F2 to enter the EXM menu. Verify the information listed on the screen, save any changes and reboot.

If necessary, refer to the section *EXM Setup Screen*, page 25 of this manual and/or your EXM manual(s) for more details.

FLOPPY DISK CNTRLR ERROR OR NO CNTRLR PRESENT

Problem: The configuration information in the BIOS setup says that one or more floppy disk drives are expected, but a floppy disk controller could not be found.

Solution(s): If you have no floppy diskette drives, enter the setup program and set both floppy drives to "NONE."

If you are using an EXP-MS/MX module, verify that the EPC-5, subplane, and EXP-MS/MX are properly seated, and check the LEDs on the front panel of the EXP-MS/MX to ensure that both the +5V and +12V supplies are available.

GENERAL FAILURE READING DRIVE ...

Problem: This almost always indicates the presence of an unformatted hard disk partition or diskette.

Solution(s): Format the partition or diskette using the utilities supplied by your operating system.

INVALID DRIVE SPECIFICATION

Problem: You are trying to access a logical drive (e.g., A:, B:, ...) that is not known to the operating system.

Solution(s): Select a different logical drive. If you are trying to access a hard disk, you may need to create the logical partition.

KEYBOARD ERROR OR NO KEYBOARD PRESENT

Problem: This message indicates that the system did not recognize a keyboard at power-up or you pressed a key during the power-on self test.

Solution(s): Check the integrity of the keyboard connector.

If you think you pressed a key during power-up, reboot the system using the front panel reset button.

Some keyboards are designed with a switch (or jumper) to allow the user to configure the keyboard for use with an AT machine or an XT machine. If this is the case with your keyboard, verify that the switch is in the AT position.

The keyboard may not be a valid PC/AT keyboard (e.g., it is a PC/XT-only or PS/2 keyboard). If this is the case, replace the keyboard with a PC/AT style keyboard.

MEMORY PARITY INTERRUPT AT ...

Problem: This could be a software error (reading a nonexistent memory area) or a true hardware failure.

Solution(s): Attempt to repeat the error. If the error occurs during the execution of your own proprietary software, verify that the memory location specified in your software is valid.

MISSING OPERATING SYSTEM

Problem: Although the system could read the hard disk and find the active partition, the operating system files could not be found.

Solution(s): This is can be caused by using a drive type number in the Fixed Disk Menu that does not match the type number used to format the hard disk. Press CTRL+ALT+ESC to run the BIOS setup program. Press F3 to enter the Fixed Disk Menu. Select the correct drive type to match the type used to format the disk originally. Save the changes and reboot the system.

This can also occur if the hard disk is partitioned and one partition is set active, but the partition was not formatted.

NON-SYSTEM DISK OR DISK ERROR
REPLACE AND PRESS ANY KEY WHEN READY

Problem: This is caused by an attempt to boot from a disk or diskette that is not recognized as a system disk; that is no system files exist on the disk or diskette.

Solution(s): Most often it results when you reboot with a non-system diskette in the floppy drive, because the BIOS always attempts to boot from the floppy drive if a diskette is installed.

If you are trying to boot from the hard disk, make sure that you do not have a diskette in the A: drive and press any key.

If you are trying to boot from floppy, insert a known good bootable system diskette in the A: drive and press any key.

NOT READY READING DRIVE ...

Problem: This is usually caused by not fully inserting a diskette into the floppy drive.

Solution(s): Eject the floppy diskette and reinsert making sure that the diskette seats completely into the floppy drive.

PARITY ERROR IN SEGMENT ...

Problem: This could be a software error (reading a nonexistent memory area) or a true hardware failure.

Solution(s): Attempt to repeat the error. If the error occurs during the execution of your own proprietary software, verify that the memory location specified in your software is valid.

PRESS A KEY TO REBOOT

Problem: A C: drive partition exists but is not set active.

Solution(s): Run your operating system disk partitioning program (like FDISK) and set the primary partition active.

REAL TIME CLOCK ERROR - RUN SETUP

Problem: The battery-backed TOD clock is incorrect.

Solution(s): Run the BIOS setup program to determine what is wrong, and correct it. If the error occurs repeatedly, the EPC-5's battery has failed.

SCSI CONTROLLER FAILED

Problem: The configuration information in the BIOS setup Fixed disk menu specifies that SCSI is enabled, but no SCSI controller could be found.

Solution(s): If you have no SCSI disk, press CTRL+ALT+ESC to run the setup program, and set the SCSI parameter to disabled.

If you are using the SCSI disk in the EXP-MS module, verify that the EPC-5, subplane, and EXP-MS are fully seated, and check the LEDs on the front panel of the EXP-MS to ensure that both the +5V and +12V supplies are available.

If you are using an external SCSI drive through a front panel connector, ensure that the SCSI cable is good, that it is not inserted backwards, and that the end of the ribbon cable is not shorting to the front panel.

SCSI DISK FAILED

Problem: The path between the SCSI controller and the disk is bad, or the drive itself has failed.

Solution(s): If you are using an EXP-MS40, check the LEDs on the front panel to ensure that both the +5V and +12V supplies are available.

If you are using an external SCSI disk cabled to the EXP-MS/MX or EXM-3 module, the most-likely cause is a bad cable or connection. Ensure that the SCSI cable is good, that it is not inserted backwards, and that the end of the ribbon cable is not shorting to the front panel.

Another possible cause is that the SCSI drive is not jumpered as target unit 0 for the first SCSI disk or target unit 1 for the second SCSI disk. If you are using an EXP-MX mass storage unit and an external SCSI disk, the SCSI disk must be jumpered as target unit 0.

Attempt to solve the problem yourself. If you are unable to solve the problem, call RadiSys Technical Support at (503) 646-1800.

10. Support and Service

In North America

Technical Support

RadiSys maintains a technical support phone line at (503) 646-1800 that is staffed weekdays (except holidays) between 8 AM and 5 PM Pacific time. If you have a problem outside these hours, you can leave a message on voice-mail using the same phone number. You can also request help via electronic mail or by FAX addressed to RadiSys Technical Support. The RadiSys FAX number is (503) 646-1850. The RadiSys E-mail address on Internet is support@radisys.com. If you are sending E-mail or a FAX, please include information on both the hardware and software being used and a detailed description of the problem, specifically how the problem can be reproduced. We will respond by E-mail, phone or FAX by the next business day.

Technical Support Services are designed for customers who have purchased their products from RadiSys or a sales representative. If your RadiSys product is part of a piece of OEM equipment, or was integrated by someone else as part of a system, support will be better provided by the OEM or system vendor that did the integration and understands the final product and environment.

Bulletin Board

RadiSys operates an electronic bulletin board (BBS) 24 hours per day to provide access to the latest drivers, software updates and other information. The bulletin board is not monitored regularly, so if you need a fast response please use the telephone or FAX numbers listed above.

The BBS operates at up to 14400 baud. Connect using standard settings of eight data bits, no parity, and one stop bit (8, N, 1). The telephone number is (503) 646-8290.

Repair Services

Factory Repair Service is provided for all RadiSys products. Standard service for all RadiSys products covers factory repair with customers paying shipping to the factory and RadiSys paying for return shipment. Overnight return shipment is available at customer expense. Normal turn-around time for repair and re-certification is five working days.

Quick Exchange services (immediate shipment of a loaner unit while the failed product is being repaired) or other extra-cost services can be arranged, but need to be negotiated in advance to allow RadiSys to pool the correct product configurations. RadiSys does not maintain a general "loaner" pool: units are available only for customers that have negotiated this service in advance.

RadiSys does not provide a fixed-price "swap-out" repair service, as customers have indicated that issues of serial number tracking and version control make it more convenient to receive their original products back after repair.

Warranty Repairs

Products under warranty (see warranty information in the front of this manual) will have manufacturing defects repaired at no charge. Products sent in for warranty repair that have no faults will be subject to a recertification charge. Extended Warranties are available and can be purchased at a standard price for any product still under warranty. RadiSys will gladly quote prices for Extended Warranties on products whose warranties have lapsed; contact the factory if this applies.

Customer induced damage (resulting from misuse, abuse, or exceeding the product specifications) is not covered by the standard product warranty.

Non-Warranty Services

There are several classes of non-warranty service. These include repair of customer induced problems, repairs of failures for products outside the warranty period, recertification (functional testing) of a product either in or out of warranty, and procurement of spare parts.

All non-warranty repairs are subject to service charges. RadiSys has determined that pricing repairs based on time and materials is more cost-effective for the customer than a flat-rate repair charge. When product is received, it will be analyzed and, if appropriate, a cost estimate will be communicated to the customer for authorization. After the customer authorizes the repair and billing arrangements have been made, the product will be repaired and returned to the customer.

A recertification service is provided for products either in or out of warranty. This service will verify correct operation of a product by inspection and testing of the product with standard manufacturing tests. There is a product-dependent charge for recertification.

There are only a few components that are generally considered field-repairable, but, because RadiSys understands that some customers want or need the option of repairing their own equipment, all components are available in a spares program. There is a minimum billing charge associated with this program.

Arranging Service

To schedule service for a product, please call RadiSys Technical Support directly at (503) 646-1800. Have the product model and serial numbers available, along with a description of the problem. A Technical Support representative will issue a Returned Materials Authorization (RMA) number, a code number by which we track the product while it is being processed. Once you have received the RMA number, follow the instructions of the Technical Support representative and return the product to us, freight prepaid, with the RMA number clearly marked on the exterior of the package. If possible re-use the original shipping containers and packaging. In any case, be sure you follow good ESD-control practices when handling the product, and ensure that anti-static bags and packing materials with adequate padding and shock-absorbing properties are used.

Ship the product, freight prepaid, to

Product Service Center
RadiSys Corporation
15025 SW Koll Parkway
Beaverton, Oregon 97006-6902

When shipping the product, include the following information: return address, contact names and phone numbers in purchasing and engineering, and a description of the suspected problem. Any ancillary information that might be helpful with the debugging process will be appreciated.

Other Countries

Contact the sales organization from which you purchased your RadiSys product for service and support.

Appendix A: Chip Set & I/O Map

The following defines the I/O addresses decoded by the EPC-5. It does not define addresses that might be decoded by EXMs and the EXP-MS/MX.

First (8-bit) DMA controller: TI 83443 chip emulating 8237 of PC/AT		
I/O Addr	Functional group	Usage
000	DMA	Channel 0 address
001		Channel 0 count
002		Channel 1 address
003		Channel 1 count
004		Channel 2 address
005		Channel 2 count
006		Channel 3 address
007		Channel 3 count
008		Command/status
009		DMA request
00A		Command register (R)
		Single-bit DMA req mask(W)
00B		Mode
00C		Set byte pointer (R)
		Clear byte pointer (W)
00D		Temporary register (R)
		Master clear (W)
00E		Clear mode reg counter (R)
		Clear all DMA req mask(W)
00F		All DMA request mask

First Interrupt controller: TI 83443 emulating 8259 of PC/AT		
I/O Addr	Functional group	Usage
020	Interrupt controller 1	Port 0
021		Port 1

83443 controller:		
I/O Addr	Functional group	Usage
024	83443 Controller	Data register
026		Index register

A

Counter-Timer functions: TI 83443 emulating 8254 of PC/AT		
I/O Addr	Functional group	Usage
040	Timer	Counter 0
041		Counter 1
042		Counter 2
043		Control (W)

Keyboard Port: Intel 8242 emulating 8742 of PC/AT		
I/O Addr	Functional group	Usage
060	Keyboard controller	Data I/O register
061	NMI status	NMI status
064	Keyboard controller	Command/status register

Time-of-Day Clock: TI 83443 emulating MC6818 of PC/AT			
I/O Addr	Functional group	Usage	
070	Real-time clock	RTC index reg / NMI enable	
071		RTC data register	
		0	seconds
		1	seconds alarm
		2	minutes
		3	minutes alarm
		4	hours
		5	hours alarm
		6	day of week
		7	date of month
		8	month
		9	year
		A	status A
		B	status B
		C	status C
		D	status D
		E	RAM
		...	
		3F	RAM

DMA Page Registers: TI 83443 emulating 74LS612 of PC/AT		
I/O Addr	Functional group	Usage
081	DMA	Channel 2 page register
082		Channel 3 page register
083		Channel 1 page register
087		Channel 0 page register
089		Channel 6 page register
08A		Channel 7 page register
08B		Channel 5 page register
08F		Refresh page register

Second Interrupt Controller: TI 83443 emulating 8259 of PC/AT		
I/O Addr	Functional group	Usage
0A0	Interrupt controller 2	Port 0
0A1		Port 1

Second (16-bit) DMA Controller: TI 83443 emulating 8237 of PC/AT		
I/O Addr	Functional group	Usage
0C0	DMA	Channel 4 address
0C2		Channel 4 count
0C4		Channel 5 address
0C6		Channel 5 count
0C8		Channel 6 address
0CA		Channel 6 count
0CC		Channel 7 address
0CE		Channel 7 count
0D0		Command/status
0D2		DMA request
0D4		Command register (R)
		Single-bit DMA req mask(W)
0D6		Mode
0D8		Set byte pointer (R)
		Clear byte pointer (W)
0DA		Temporary register (R)
		Master clear (W)
0DC		Clear mode reg counter (R)
		Clear all DMA req mask (W)
0DE		All DMA request mask

A

Coprocessor Interface: On the EPC-5, an 80387 replaces the 80287 of PC/AT		
I/O Addr	Functional group	Usage
0F0	Coprocessor	Clear coprocessor busy
0F1		Reset coprocessor

Serial I/O (Com2) Port: VLSI 16C452 emulates 8251 of PC/AT		
I/O Addr	Functional group	Usage
2F8	COM2 serial port	Receiver/transmitter buffer
		Baud rate divisor latch (LSB)
2F9		Interrupt enable register
		Baud rate divisor latch (MSB)
2FA		Interrupt ID register
2FB		Line control register
2FC		Modem control register
2FD		Line status register
2FE		Modem status register

Parallel I/O (LPT1) Port: VLSI 16C452 emulates 8255 of PC/AT		
I/O Addr	Functional group	Usage
378	LPT1 parallel port	Printer data register
379		Printer status register
37A		Printer control register

Serial I/O (Com1) Port: VLSI 16C452 emulates 8552 of PC/AT		
I/O Addr	Functional group	Usage
3F8	COM1 serial port	Receiver/transmitter buffer
		Baud rate divisor latch (LSB)
3F9		Interrupt enable register
		Baud rate divisor latch (MSB)
3FA		Interrupt ID register
3FB		Line control register
3FC		Modem control register
3FD		Line status register
3FE		Modem status register

EPC-5 Memory Mapping Registers: No PC/AT equivalent		
I/O Addr	Functional group	Usage
8130	VME and misc control	VME map WA21-16
8132		Alias of 8130
8134		Alias of 8130
8136		Alias of 8130
8140		ID low
8141		ID high
8142		Device type low
8143		Device type high
8144		Status/control low
8145		Status/control high
8146		Slave offset low
8147		Slave offset high
8148		Protocol low
8149		Protocol high
814A		Response low
814B		Response high
814C		Message high low
814D		Message high high
814E		Message low low
814F		Message low high
8150		VME map WA31-24
8151		VME modifier
8152		VME interrupt state
8153		VME interrupt enable
8154		VME event state
8155		VME event enable
8156		Module status/control
815F		VME interrupt generator

NOTES

Appendix B: Interrupts and DMA Channels

B

Interrupts

The assignment of interrupts for the EPC-5 is shown in the following table:

NMI	DRAM parity error, EXM expansion interface I/O channel check
IRQ0	timer
IRQ1	keyboard
IRQ2	IRQ8 - IRQ15 cascade through IRQ2
IRQ3	COM2 serial port
IRQ4	COM1 serial port
IRQ5	unassigned
IRQ6	usually needed for floppy disk controller
IRQ7	LPT1 parallel port
IRQ8	clock
IRQ9	unassigned
IRQ10	VME interrupt/event
IRQ11	unassigned
IRQ12	used by optional SCSI disk controller
IRQ13	coprocessor
IRQ14	used by optional IDE disk controller
IRQ15	unassigned

DMA Channels

The assignment of DMA channels for the EPC-5 is shown in the following table.

0	unassigned (8-bit)
1	unassigned (8-bit)
2	usually needed for floppy disk (8-bit)
3	usually needed for SCSI disk (8-bit)
4	(Channel 0 - Channel 3 cascade through Channel 4)
5	unassigned (16-bit)
6	unassigned (16-bit)
7	unassigned - not connected to EXM expansion interface (16-bit)

Index

W

NOTES