EPC-8A

Hardware Reference

RadiSys Corporation

5445 NE Dawson Creek Drive

Hillsboro, OR 97124

(503) 615-1100

FAX: (503) 615-1150

www.radisys.com

07-0961-01 December 1998

December 1998

Before you begin

This guide describes the EPC-8A, a highly integrated PC-compatible computer designed specifically for use in the VMEbus and extended VMEbus (VXIbus) environments.

About this guide

This guide explains how to install, configure, and troubleshoot the EPC-8A, and describes the EPC-8A and how it works.

Guide contents

Chapters

Chapter		Description
1	Product description	Provides an overview and specifications for the EPC-8A
2	Configuration and Installation	Explains how to configure and install the EPC-8A.
3	BIOS Configuration	Details the various menus and sub-menus used to configure the system using the Phoenix BIOS.
4	Theory of Operation	Explains how the EPC-8A works.
5	Programming the VMEbus Interface	Describes the EPC-8A VMEbus interface as seen by a program

Appendices

Appendix		Description
A	Chipset and I/O Map	Defines the I/O addresses decoded by the EPC-8A
B	Interrupts and DMA Channels	Lists the EPC-8A's interrupt assignments.
C	Connectors	Specifies the EPC-8A's connector details.
D	Memory	Lists the EPC-8A's SIMM specifications.
E	Subplanes	Describes how to install subplanes in the EPC-8A.
F	Registers	Defines registers in the I/O space specific to the EPC-8A.
G	XFORMAT Software for the EPC-8A	Explains when to use the XFORMAT utility.
H	AUTOSET Software	Describes how to use the AUTOSET program to configure the Ethernet controller.
I	SVGA	Provides operating instructions for user utilities and installation instructions for the display drivers supplied with your Chips and Technologies SVGA for the EPC-8A

Appendix		Description
J	Error Messages and Diagnosis	Provides troubleshooting information in these areas: • Problems that do not display an error message. • Beep codes (audible codes consist of patterns of beeps and pauses). • Error and warning messages.
K	Configuring the Ethernet Drivers	Provides instructions for configuring network interface drivers after you install the EPC-8A.

Notational conventions

This manual uses the following conventions:

- All numbers are decimal unless otherwise stated.
- Bit 0 is the low-order bit. If a bit is set to 1, the associated description is true unless otherwise stated.
- `Data structures and syntax strings appear in this font.`


Notes indicate important information about the product.


Cautions indicate situations that may result in damage to data or the hardware.


Tips indicate alternate techniques or procedures that you can use to save time or better understand the product.


Warnings indicate situations that may result in physical harm to you or the hardware.

Where to get more information

You can find out more about the EPC-8A from these sources:

- **Readme file**: Lists features and issues that arose too late to include in other documentation.
- **RadiSys web site**: Go to www.radisys.com

Contents

Chapter 1: Product description
Overview 1
VMEbus 2
System controller functions 2
VMEbus master interface 2
VMEbus extended register set 3
Specifications 3
Chapter 2: Configuration and Installation
Before you begin 5
Configuring the EPC-8A 5
Slot-1 Functionality 5
Selecting the EPC-8A Slot Location 6
Installing the VMEbus Backplane Jumpers 7
EPC-8A Insertion 9
Connecting Peripherals to the EPC-8A 10
Remaining Steps 10
Chapter 3: BIOS Configuration
BIOS Setup Screens 11
Main BIOS Setup Menu 12
IDE Adapter Sub-Menus 13
Memory Cache Sub-Menu 15
Memory Shadow Sub-Menu 16
About Shadow Memory Regions 16
Boot Sequence Sub-Menu 17
Keyboard Features Menu 19
Advanced Menu 20
Integrated Peripherals Sub-Menu 22
EXM Menu 23
VME Menu 25
Exit Menu 26
Chapter 4: Theory of Operation
Overview 29
Processor board 30
Processor and Coprocessor 30
Memory 31
Memory Map 31
BIOS ROM and ROM Shadowing 31
CMOS Backup and Restore 32
Video Controller 32
Ethernet Controller 33

Resident Flash/SRAM Memory 33
Resident Flash Memory 33
SRAM 33
Watchdog Timer 33
Battery 34
Peripheral Ports 35
RS-232 Port 35
RS-422/485 Port 35
Parallel Port 35
Keyboard 35
Front Panel LEDs 35
Resetting the EPC-8A 36
Power-off, Power-on 36
Power low 36
Front-panel Reset button 36
Ctrl+Alt+Del 37
VMEbus SYSRESET 37
Watchdog Timer 37
Register State after Reset 37
VME/VXI Soft RESET state and SYSRESET 37
Signal Register FIFO 38
EXM Expansion Interface 38
VME Interface 39
Connectivity 39
VMEbus System (Slot-1) Controller Functions 39
VMEbus Access 39
Byte Ordering 40
VMEbus Interrupt Response 41
VME Extension Registers (VXI) 41
Passing VME Interrupts and Events to the CPU 42

Chapter 5: Programming the VMEbus Interface

Concepts 43
Atomic access 43
Read-Modify-Write Operations 43
Setting the VMEbus Access Bit 44
Real-Mode “E-page” VMEbus Accesses 44
Supported Address Modifiers 45
Low-Level Programming “E” Page Accesses 45
Example #1 45
Example #2 46
Low-Level Handling of VMEbus Interrupts 47
Start of Loop 48
Protected-Mode Direct VMEbus Accesses 48
Generating IACKs in Protected Mode 49
Programming the Watchdog Timer 50

Appendix A: Chipset and I/O Map 51

Appendix B: Interrupts and DMA Channels

Interrupts 57
DMA Channels 57

Appendix C: Connectors
RS-232 Port (COM1) ... 59
RS-422/485 Port (COM2) ... 59
Parallel Port ... 60
Keyboard ... 60
VGA ... 61
Ethernet ... 61
Appendix D: Memory
Memory ... 63
Appendix E: Subplanes
Subplane Installation ... 65
EXP-BP1 Subplane ... 65
EXP-BP2 Subplane ... 66
EXP-BP4 Subplane ... 67
EXP-BP3A Subplane ... 68
EXP-BP5 Subplane ... 69
EXP-BP4A Subplane ... 70
EXP-BP6 Subplane ... 71
Appendix F: Registers
Registers Specific to the EPC-8A ... 73
Configuration (8102h) ... 74
VME A21–16 Address Register (8130h) ... 75
ID Register (8140h and 8141h) ... 75
Device Type Register (8142h and 8143h) ... 75
Status/Control Register (8144h and 8145h) ... 75
Reserved (8146h and 8147h) ... 76
Protocol Register/Signal FIFO (8148h and 8149h) ... 77
Response Register (814Ah and 814Bh) ... 77
Reserved (814C and 814D) ... 79
Message Low Register (814Eh and 814Fh) ... 79
VME A31–24 Address Register (8150h) ... 79
VME Modifier Register (8151h) ... 79
VME Interrupt State Register (8152h) ... 80
VME Interrupt Enable Register (8153h) ... 80
VME Event State Register (8154h) ... 80
VME Event Enable Register (8155h) ... 80
VME Interrupt Generator Register (8158h) ... 81
Unique Logical Address Register (815Ch) ... 81
Module Status/Control Register (815Dh) ... 81
VMEbus Mapped Registers ... 82
Appendix G: XFORMAT Software for the EPC-8A
XFORMAT Software ... 83
Appendix H: AUTOSET Software
AUTOSET Software ... 85
Installing the Software ... 85
Using the AUTOSET Program ... 85
Novell Configuration ... 87
I/O Base Address ... 87
Adapter Architecture ... 88
Interrupts ... 88

Adapter Architecture 88
Boot PROM 89
Configuring Additional Ethernet Controllers 89
Diagnostics 90
Error Messages 90
Appendix I: SVGA
Video Controller Hardware 93
Display Drivers and Utilities 93
Introduction 93
Before you begin 94
Notational conventions 94
Easy installation 94
Windows 95 installation 94
Windows 3.x installation 94
Chips and Technologies web site 94
Utility software 95
Microsoft Windows 3.1 95
Before upgrading from a previous release 95
Microsoft Windows 95 96
Driver Installation Procedure 96
OSR2 Driver Installation Procedure 96
Appendix J: Error Messages and Diagnosis
Troubleshooting 99
BIOS Beep Codes 100
Common Error Messages 100
Appendix K: Configuring the Ethernet Drivers
NetWare IPX Driver for DOS Installation 105
ODI Driver for DOS Installation 105
NDIS Driver for DOS Installation 106
Packet Driver Installation 107
Windows NT Driver Installation 107
Index 109

Figures

Figure 2-1. Jumper Locations 6
Figure 2-2. Daisy-Chain Signal Concept 7
Figure 2-3. Backplane Jumpers Required for EPC-8A Subsystem 8
Figure 2-4. VMEbus Jumpers on Rear Wirewrap Pins 9
Figure 2-5. VMEbus Jumpers on Front Stake Pins 9
Figure 3-1. Main menu 12
Figure 3-2. IDE Adapter sub-menu 13
Figure 3-3. Memory Shadow sub-menu 15
Figure 3-4. Memory Shadow sub-menu 16
Figure 3-5. Boot Sequence sub-menu 17
Figure 3-6. Keyboard features sub-menu 19
Figure 3-7. Advanced menu 20
Figure 3-8. Integrated Peripherals sub-menu 22
Figure 3-9. EXM menu 23
Figure 3-10. Slot Numbering 24
Figure 3-11. VME menu 25
Figure 3-12. Exit menu 26
Figure 4-1. Block Diagram 30
Figure 4-2. Flash Boot Device Memory 32
Figure 4-3. Using little-endian byte ordering 40
Figure 4-4. Using big-endian byte ordering 41
Figure 4-5. Passing VME interrupts and events to the CPU 42
Figure E-1. EXP-BP1 Subplane 65
Figure E-2. EXP-BP2 Subplane 66
Figure E-3. EXP-BP4 Subplane 67
Figure E-4. EXP-BP3A Subplane 68
Figure E-5. EXP-BP5 Subplane 69
Figure E-6. EXP-BP4A Subplane 70
Figure E-7. EXP-BP6 Subplane 71
Figure H-1. AUTOSET Program Main Menu 86
Figure H-2. Manual Configuration Menu 86

Tables

Table 1-1. EPC-8A Environmental Specifications 3
Table 1-2. Additional EPC-8A Specifications 3
Table 2-1. VME Slots Available 7
Table 4-1. Memory at addresses between 0 and 1MB (0FFFFFh) 31
Table 4-2. Extended memory address space for a 4MB EPC-8A 31
Table 4-3. Reset Conditions 36
Table 5-1. Supported Address Modifiers 45
Table 5-2. VME access type 49
Table A-1. First (8-bit) DMA controller: RadiSys R400EX chip emulating 8237 of PC/AT 51
Table A-2. First Interrupt controller: RadiSys R400EX emulating 8259 of PC/AT 51
Table A-3. R400EX controller 51
Table A-4. Counter-Timer functions: RadiSys R400EX emulating 8254 of PC/AT 52
Table A-5. Keyboard Port: RadiSys R400EX emulating 8742 of PC/AT 52
Table A-6. Time-of-Day Clock: RadiSys R400EX emulating MC146818 of PC/AT 52
Table A-7. DMA Page Registers: RadiSys R400EX emulating 74LS612 of PC/AT 52
Table A-8. Port A 53
Table A-9. EXM Configuration 53
Table A-10. Second Interrupt Controller: RadiSys R400EX emulating 8259 of PC/AT 53
Table A-11. Second (16-bit) DMA Controller: RadiSys R400EX emulating 8237 of PC/AT 53
Table A-12. Coprocessor Interface: For the EPC-8A DX, DX replaces the 80287 of PC/AT 54
Table A-13. EXM Configuration 54
Table A-14. Ethernet 54
Table A-15. Serial I/O (Com2) Port: National Semiconductor PC87336 emulating 8251 of PC/AT 54
Table A-16. Parallel I/O (LPT1) Port: National Semiconductor PC87336 emulating 8255 of PC/AT 54
Table A-17. VGA 54
Table A-18. Serial I/O (Com1) Port: National Semiconductor PC87336 emulating 8552 of PC/AT 55
Table A-19. EPC Registers 55
Table A-20. VME Registers 55
Table A-21. VXI, VME/Misc., and On-board EXM-2A Registers 56
Table B-1. Interrupts 57
Table B-2. DMA Channels 57
Table C-1. DB-9 Pin-Out 59
Table C-2. COM2 RS-422/RS485 Port Pin-out 59
Table C-3. DB-25 Pin-out 60
Table C-4. Keyboard Pin-out 60
Table C-5. DB-15 Pin-out 61
Table C-6. RJ45 Phone Jack Pin-out 61
Table I-1. Notational Conventions 94
Table J-1. Troubleshooting Error Messages 99
Table J-2. BIOS Troubleshooting Beep Codes 100

1 Product description

Overview

The EPC-8A, compatible with major PC software environments such as Microsoft† DOS, Microsoft Windows† 3.x, Microsoft Windows 95, and Microsoft Windows NT† is a highly integrated PC-compatible computer designed specifically for use in the VMEbus and extended VMEbus (VXIbus) environments.

The EPC-8A is available s either a Single- or Dual-slot 6U VMEbus module. The two configurations are functionally identical, and differ only in that the two-slot module includes mechanical support for two EXM modules.

The PC-compatible portion of the architecture includes these PC-standard features:

- Intel 486 processors:
  - Dual-slot option: DX4-100 processor
  - Single-slot option: DX4-100 processor
- 8, 16, or 32MB of DRAM, using an industry-standard 80ns fast page mode or EDO 72 pin SIMM.
- 512 KB Flash BIOS EPROM.
- RadiSys standard EXM expansion interface for adding PC-architecture peripherals or interfaces.
- PhoenixBIOS† with RadiSys enhancements to support the additional on-board features and VMEbus environment.
- Standard PC peripheral interfaces, including an RS232 serial port and a bi-directional parallel port, all with standard interface connectors. The PS/2†-style keyboard connector includes an adapter.
- All other standard PC architectural features, including an on-board speaker.

Additional features (some optional) include:

- SVGA interface based on the Chips and Technologies 65545 chipset, with 512 Kbytes of video RAM providing local bus graphics performance and non-interlaced resolutions to 1024 x 768, with 16 colors. The standard 15-pin connector for this port is enhanced with a programmable output on pin 15, which is not connected on a standard VGA interface.
- 10BASE-T Ethernet interface based on the National Semiconductor† AT/LANTIC† chip set, which provides emulation of Western Digital WD8013, Novell† NE2000 and

NE2000+ Ethernet controllers. Interface configuration (Emulation mode and address/ interrupt control) is software controlled; a DOS-based application is provided to perform this operation.

- Second modified RS-422/485 PC-compatible serial port interface implemented with RS-422/485 transmit and receive buffers. CTS, RTS, DSR and DTR are buffered to RS-422 levels. The Transmit lines are controlled by the RTS signal, providing RS-485 multidrop support.
- Watchdog timer that you can configure to generate a VMEbus SYSFAIL signal and reset the processor. The processor can be configured to reboot or halt. The timer is implemented as a software-retriggerable one-shot, with a programmable reset interval ranging from 125 mS to 8000 mS.
- The ability to boot from VMEbus memory or resident Flash memory.
- (Optional) 128 Kbytes of battery-backed memory-mapped static RAM.
- (Optional) 2 Mbytes of flash-EPROM based non-volatile memory. This memory can be protected against accidental overwriting by removing a shorting jumper. Software is available to support this in the DOS/Windows environment. This memory can also be configured as a bootable disk.

VMEbus

The VMEbus implementation provides a complete bus interface with enhancements for multiprocessor environments. For detailed information, continue reading.

System controller functions

With a single hardware configuration jumper, the EPC-8A can provide full VME SLOT-1 arbitration functions. When enabled, the provided functions include priority and round-robin bus arbitration, IACK and bus grant daisy-chain driving, SYSRESET and SYSCLK generation, and bus time-out detection. ROR (release-on-request) or a fair-requester, RONR (release-on-no-request) bus release mechanism is software selectable.

VMEbus master interface

The VMEbus master interface provides a full 32-bit data path to the 16, 24, and 32-bit address spaces of the VMEbus. The bus interface is designed with a minimum number of state registers, which maximizes performance in a multitasking or interrupt-driven environment.

Programmable hardware byte-swapping is provided for ease of communication with other processor architectures that may share the VMEbus. All the VMEbus address spaces can be addressed from both protected-mode and real-mode operating systems.

The EPC-8A can generate or respond to all 7 standard VMEbus interrupts, and can also receive the VMEbus signals ACFail, BERR, and Sysfail as interrupts. When it is used to generate interrupts, it implements 16-bit IACK cycles.

VMEbus extended register set

As an extension to the VMEbus interface, the EPC-8A implements the set of VXIbus (IEEE 1155-1992) standard multiprocessor support registers in the A16 space.

These registers provide a set of standard identification, status, control and communication functions that are useful in multiprocessor environments. They allow dynamic system configuration by providing for board identification, and provide well-defined multi-processor communications channels and protocols.

The EPC-8A includes a complete set of VXIbus-defined message-based device registers. These registers, implemented in a proprietary gate array and mapped into the VMEbus A16 address space, include a device-type identifier register (supporting geographic addressing), bus status and control registers, and a register-based message passing facility.

Specifications

Table 1-1. EPC-8A Environmental Specifications

Characteristic		Value
Temperature	Operating	0 – 60°C at point of entry of forced air derated 2°C per 1000 ft. (300 m) over 6600 ft (2000m)
	Storage	–40 to 85°C 5°C per hour max excursion gradient
Humidity	Operating	5% – 95% noncondensing
	Storage	5% – 95% noncondensing
Altitude	Operating	0 – 10,000 ft (3000 m)
	Storage	0 – 40,000 ft (12,000 m)
Vibration	Operating	0.015 inch (0.38 mm) P-P displacement with 2.5 g peak (max) acceleration over 5–300 Hz
	Storage	0.030 inch (0.76 mm) P-P displacement with 5.0 g peak (max) acceleration over 5–300 Hz
Shock	Operating	30 g, 11 mS duration, half-sine shock pulse
	Storage	50 g, 11 mS duration, half-sine shock pulse
Airflow		6 Liters/sec. for the dual-slot module 3 Liters/sec. for the single-slot module

Note: Specifications apply to only the EPC-8A CPU.

Table 1-2. Additional EPC-8A Specifications

Characteristic		Value
Electrical		
Current	+5V	3.9A typical; 4.2A maximum
	+12V	100mA typical; 100mA maximum
	–12V	100mA typical; 100mA maximum
Other		
Weight	Without EXMs	2.7 lb. (1.3 kg)

Table 1-2. Additional EPC-8A Specifications

Characteristic		Value
VME	Master address	A16, A24, A32
	Master transfer	D08(EO), D16, D32
	Slave address	A16
	Slave transfer	D08(EO), D16
	Interrupter	I(1–7)
	Interrupt handler	D08(O),D16 IH(1–7)
	Requester	ROR, RONR
	Arbiter	RRS, PRI
	System controller	SYSCLK, IACK and bus grant daisy chains, bus time-out error (BERR)
VXI	Device type	Message based
	Protocols	Commander/master/interrupter
	Manufacturer code	4076 RadiSys Corporation
	Model code	196 (if configured for slot 0) 452 (if configured for other than slot 0)

Configuration and Installation

2

To configure and install the EPC-8A, you must complete the following steps:

1. Set jumpers as needed for Flash, BIOS, and Slot-1
2. Select the slot location
3. Install VMEbus backplane jumpers
4. Insert subplane into mainframe
5. Insert EPC-8A into mainframe
6. Connect peripherals
7. Power-up
8. Configure BIOS
9. Boot operating system
10. Install software/configure system

Before you begin

Unpack the EPC-8A and inspect it for shipping damage.

Do not remove the EPC-8A module from its anti-static bag unless you are in a static-free environment.

The EPC-8A, 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.

During the installation process, ensure that power to your system is off.

The EPC-8A is not designed to be inserted or removed while the chassis is powered up.

Configuring the EPC-8A

Slot-1 Functionality

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 and bus grant daisy chains.
- Provides bus timeout error (BERR) function

The EPC-8A 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 2-1). Removing the jumper disables Slot-1 functionality. When the EPC-8A is configured as the Slot-1 controller, it performs all the standard VMEbus system control functions.

Jumper JP2 (LOOP) is reserved for use by RadiSys and should not be installed.

The BIOS Jumper JP2 (-BIOS) enables the BIOS Flash memory for writing.

The Flash Jumper JP2 (-FLSH) enables the resident Flash memory for writing and must be installed when using software such as the XFORMAT utility.

The Recovery Jumper JP2 (RCVR) must be installed only to recover a corrupted BIOS. For example, if the power source was interrupted during a BIOS upgrade, and the corrupt BIOS can only be reflashed by a Flash recovery floppy disk installed on EXM Slot 0.

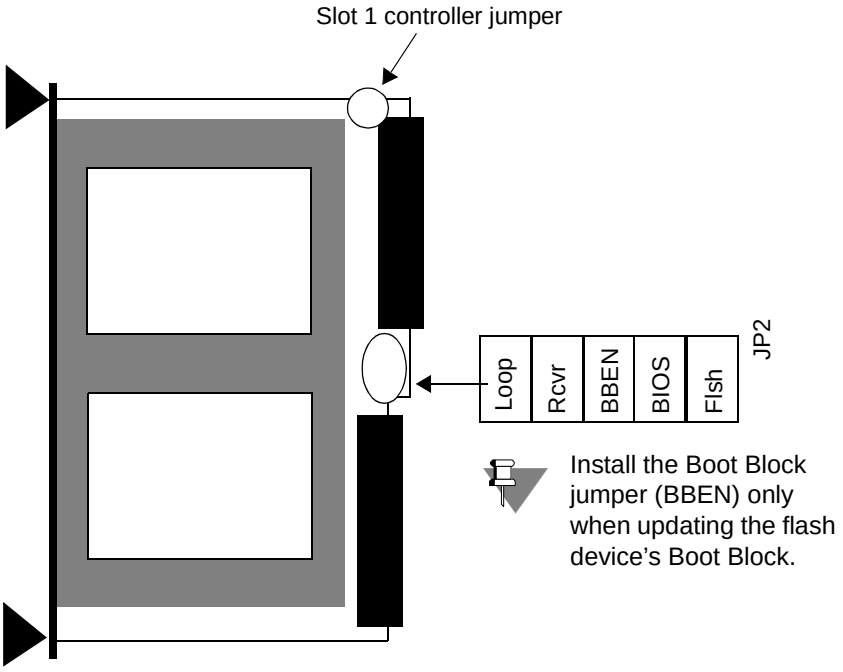


Figure 2-1. Jumper Locations

Selecting the EPC-8A Slot Location

There are two main considerations in determining where the EPC-8A 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-8A connects to its peripherals via a subplane which extends to the right of the EPC-8A. Make sure that the location you choose provides sufficient room for all the attached peripherals (EXMs and mass storage module).

The EPC-8A plus EXM expansion modules plus any mass storage module can be considered together as a single subsystem. Use the following worksheet in Table 2-1 to determine the total number of VME slots your particular subsystem configuration requires.

Table 2-1. VME Slots Available

Product	VME Slots	Total
Single-slot EPC-8A (Includes first EXM module)	1	
Double-slot EPC-8A (Includes first two EXM modules)	2	
Additional EXP-MC(s) (Holds additional two EXM modules)	1 each	
EXP-AM	2	
Mass Storage Module (EXP-MX)	2	
Total VMEbus slots used		

Once you determine where you want to place the EPC-8A subsystem in the chassis, the VME backplane must be jumpered appropriately.

Installing the VMEbus Backplane Jumpers

The VMEbus propagates 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 (for example, 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.

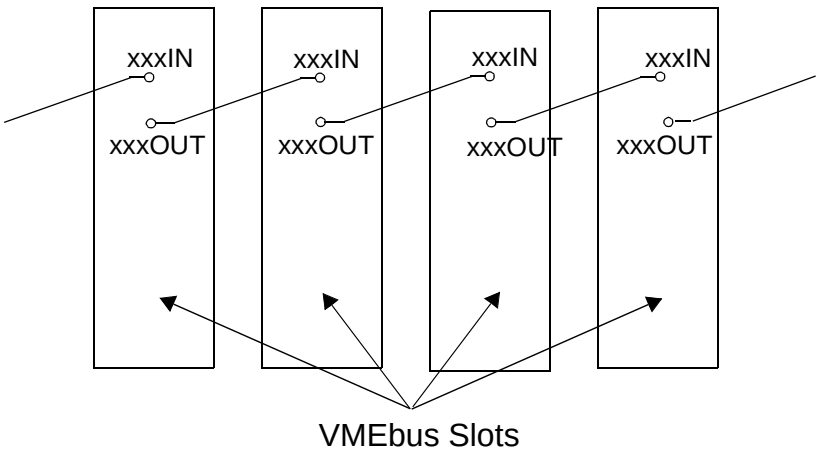


Figure 2-2. Daisy-Chain Signal Concept

The daisy-chain signal concept is shown in Figure 2-2. 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 (For example, BG0In to BG0Out, BG1In to BG1Out,...,IackIn to IackOut) either through the board in that slot or by jumpers. (Boards that meet VMEbus specifications correctly handle the signals and do not need backplane jumpers. However, many early designs may not propagate these signals correctly.) 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 2-3 and 2-4 on the following pages for examples.

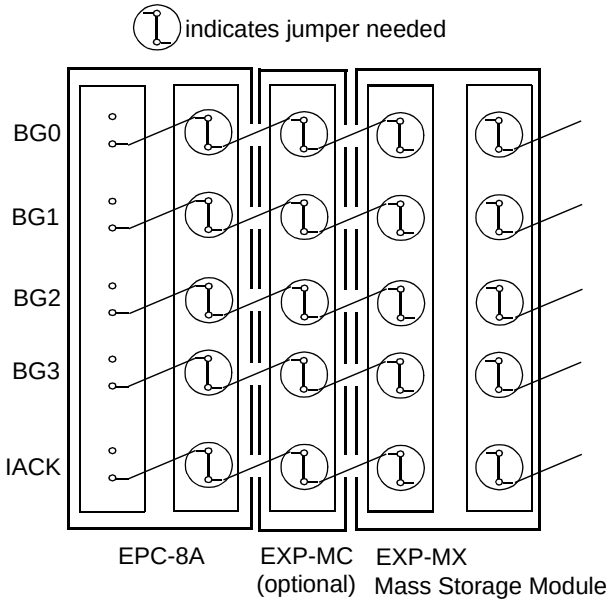


Figure 2-3. Backplane Jumpers Required for EPC-8A Subsystem

The figure above shows the jumpers required for a five-slot EPC-8A subsystem, consisting of a two-slot EPC-8A, an EXP-MC module carrier for two additional EXM modules, and an EXP-MX storage module. Note that the left-most slot does not require any jumpers. All other slots occupied by the subsystem require all five jumpers be installed.

Once you determine 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.

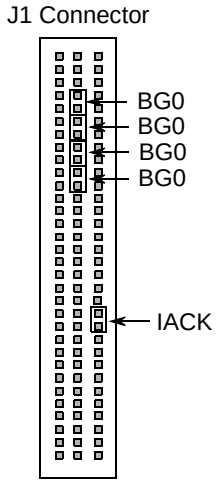


Figure 2-4. VMEbus Jumpers on Rear Wirewrap Pins

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 2-4. The connector may have wire-wrap tails on all pins, or just on the bus-grant and IAC pins.

Stake pins (front or rear) can also be located adjacent to the slot being jumpered as shown in Figure 2-5. Typically, the stake pins are located between the slot being jumpered and the next lower-numbered slot (For example, jumpers for Slot 6 would be located adjacent to Slot 6 between Slots 5 and 6).

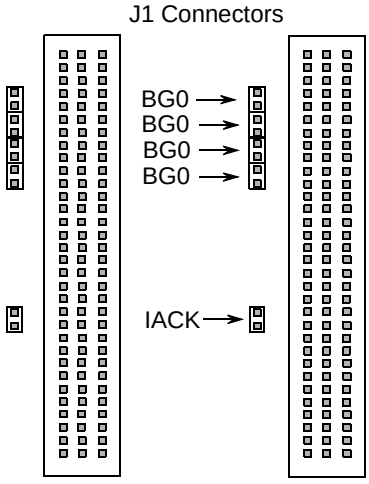


Figure 2-5. VMEbus Jumpers on Front Stake Pins

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

EPC-8A Insertion

The EPC-8A must be installed onto a subplane that fits between the EPC-8A and the VMEbus backplane. Subplanes are discussed in more detail in Appendix E.

The subplane is installed first, connecting to the backplane. After installing the subplane, the EPC-8A processor module can be inserted into the VMEbus chassis.

Make sure that power to your VME system is off. the EPC-8A module is not designed to be inserted or removed from live backplanes.

When inserting the EPC-8A module, avoid touching the circuit board and connector pins, and make sure the environment is static-free.

- 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.)
- Slide the EPC-8A module into the left-most slot occupied by the subplane. Use firm pressure on the handles to mate the module with the connectors.
- 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.

Connecting Peripherals to the EPC-8A

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 next step of installation is connecting peripherals, typically a video display and keyboard, but also perhaps a mouse, modem, printer, etc. Pin-outs for the EPC-8A front-panel connectors are specified in Appendix C, *Connectors*.

Remaining Steps

The remaining configuration steps may include BIOS configuration, driver software installation and application software installation. Your system may be pre-configured by your supplier or you may be required to perform these tasks yourself.

Chapter 3 of this manual describes the BIOS set up program. Many of the hardware setups are accomplished using the BIOS.

Other features of the EPC-8A, such as the Ethernet port, graphics controller, and optional Flash memory require drivers and are configured using DOS programs or specific operating system functions and utilities. Installation of other software drivers that are optionally available with the EPC-8A is described in the appropriate sections of this manual. These include the following:

- A resident Flash memory utility (See *Appendix G, XFORMAT Software for the EPC-8A*)
- An Ethernet configuration utility (See *Appendix H, AUTOSET Software*)
- SVGA display drivers (See *Appendix I, SVGA*)
- Ethernet drivers (See *Appendix K, Configuring the Ethernet Drivers*)

3 BIOS Configuration

This chapter details the various menus and sub-menus you use to configure the system. The EPC-8A uses the PhoenixBIOS program to configure and select various system options. While the chapter is written as though you are encountering each field in sequence and for the first time, your system may be pre-configured and require very little intervention.

Some error messages might occur during the execution of the BIOS initialization sequence. If errors occur during the power-on self-test (POST), the BIOS displays the error on the appropriate line of the screen display and, depending on how your system is configured, either pauses or attempts to continue. Refer to Appendix J, *Error Messages and Diagnosis.*

BIOS Setup Screens

The EPC-8A's BIOS contains a setup function to display and alter the system configuration. This information is maintained in the EPC-8A's nonvolatile CMOS RAM and is used by the BIOS to initialize the EPC-8A hardware.

The BIOS Setup can only be entered during the system reset process, following a power-up, front panel reset, or equivalent. Press the F2 key when prompted to enter Setup.

You can always press the F2 key to enter the BIOS setup screens, even if the prompt is suppressed. You can suppress the F2 key prompt in the BIOS setup.

Press the up and down cursor (arrow) keys to move from field to field. Press the right and left arrows to move from menu to menu, as noted in the menu bar at the bottom of the screen. If you use the arrow keys to leave a menu and then return, your active field is always at the beginning of the menu. If you select a sub-menu and then return to the main menu, you return to that sub-menu heading.

Fields with a triangle to the left are actually sub-menu headings; press Enter when the cursor rests on one of these headings to reach that sub-menu. For most fields, position the cursor at the field and from the numeric keypad, press the + and – keys to rotate through the available choices. Certain numeric fields can also be entered via the keyboard. Once the entry has been changed to appear as desired, use the up and down arrow to move to the next field.

Main BIOS Setup Menu

The next figure shows the main BIOS setup menu.

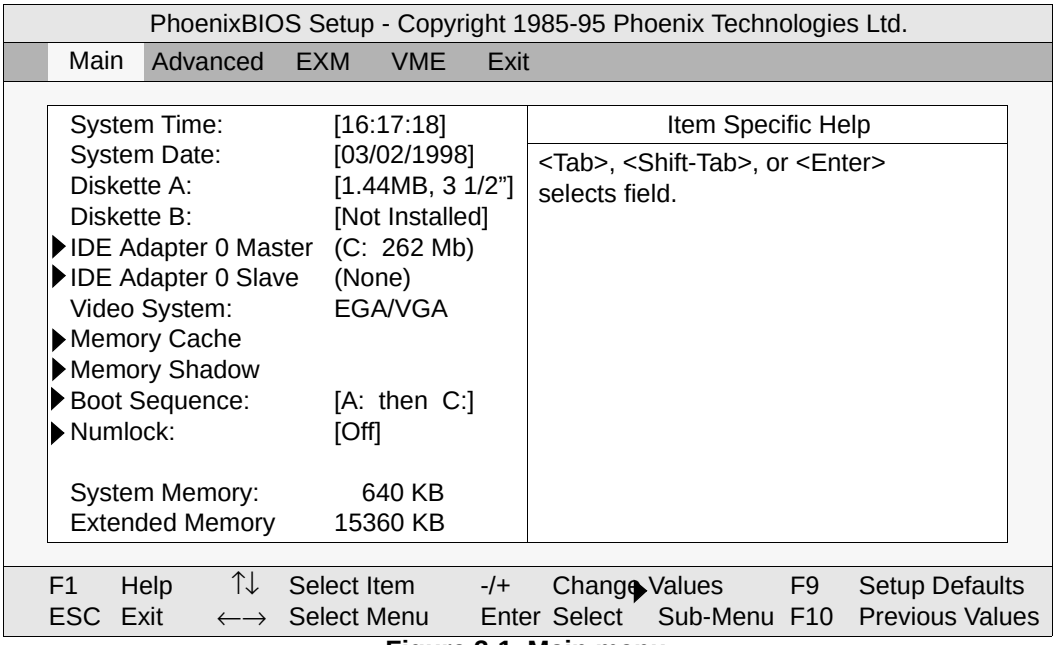


Figure 3-1. Main menu

The rightmost menu in the menu bar is the Exit Menu. Use the options in this menu to save your changes, re-load default BIOS settings, and so on. By pressing the ESC key, you immediately go to the Exit Menu.

The fields in each menu and sub-menu are explained below. Additional help information is available in the help area on the BIOS setup screen.

Field	Description
System Time/System Date	Sets the system time and date. To change these values, go to each field and enter the desired value. Press the tab key to move from hour to minute to second, or from month to day to year.
Diskette A/Diskette B	Identifies the type of floppy disk drive installed as the A drive. If the EPC-8A has a floppy drive installed, the proper setting is usually for a 1.44 MB floppy disk drive. Other options include 360K, 720K, 1.2 Mbyte, and 2.88 Mbytes. If no drive is installed, the proper setting is NOT INSTALLED.
IDE Adapter 0 Master/Slave Sub-Menus	Displays a menu that you use to enter disk drive information. When entered, the Main Menu shows the drive selected. For more information, see IDE Adapter Sub-Menus on page 13.
Memory Cache Sub-Menu	Controls the use of the CPU cache. For more information, see Memory Cache Sub-Menu on page 15.
Memory Shadow Sub-Menu	The term "Memory Shadow" refers to the technique of copying information from ROM into RAM and accessing it in this alternate memory location. For more information, see Memory Shadow Sub-Menu on page 16.
Video System	Identifies the video options you want to use: EGA/VGA, CGA 80x25, or monochrome. The EPC-8A's onboard video is VGA.

Field	Description
Boot Sequence Sub-Menu	Displays a menu that you use to change the boot delay and boot sequence, and disable several displays during the boot process, such as the SETUP prompt, POST errors, floppy drive check, and summary screen. Once you set the boot sequence, your choice displays in the Main menu. For more information, see Boot Sequence Sub-Menu on page 17.
Keyboard Features (Numlock) Sub-Menu	Displays a menu that you use to enable or disable the Numlock key and key click, and set the keyboard auto-repeat rate and delay. The Numlock setting displays for this entry in the Main Menu. For more information, turn to the section concerning the Keyboard Features sub-menu.

Note: System Memory and Extended Memory are display-only fields set by the BIOS. No user interaction is required.

IDE Adapter Sub-Menus

Two IDE adapter sub-menus exist: one for the master drive and one for the slave drive. To use an EXM-HD, EXM-MX, or EXP-MX series mass storage unit, you must configure a master adapter; the slave is optional, and not relevant to most RadiSys hardware. To see the detailed characteristics of the device or to change the device, choose the IDE Adapter 0 Master Sub-Menu to configure the fixed disk. The following screen displays:

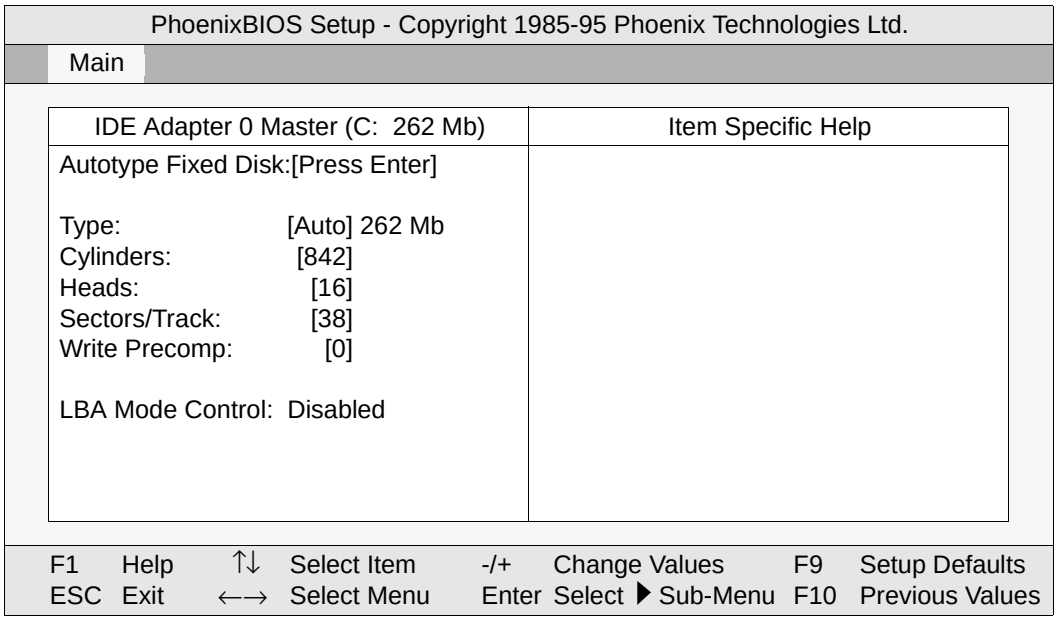


Figure 3-2. IDE Adapter sub-menu

Field	Description
Autotype Fixed Disk	Sets up new disks. This option allows the BIOS to determine proper settings for the disk based on information on the disk, which is detected by the EPC-8A BIOS for drives that comply with ANSI specifications. Press the ENTER key to invoke this function. Existing (formatted) disks must be set up using the same parameters originally used when the disk was formatted. You must enter the specific cylinder, head, sector information as listed on the label attached to the drive at the factory. Use the “User” type described below.
Type	Identifies the disk type. The majority of users who are using a pre-configured system probably have an IDE hard disk drive. • If you have an IDE disk but cannot employ the “Autotype” feature, then select “User” for the Type and enter the correct drive values for cylinders, heads, sectors/track, and write precomp from the label attached by RadiSys at the factory. • If yours is not an IDE hard disk drive, select “None”. • You can also set the "Type" to "Auto" which causes POST to autotype the IDE drive every time it runs. • For disks not supplied by RadiSys, consult the product’s documentation. Note that there are some restrictions when setting up devices on the EPC-8A. If you plan to boot from a non-IDE device, such as the resident Flash memory or VMEbus, set the C: drive as None and use the BIOS extension. Flash and VMEbus BIOS extensions are enabled and configured in the Advanced Menu. Once you complete setup for the IDE Master, you can choose the IDE Adapter 0 Slave Sub-menu to configure your second drive. When finished, press the ESC key to return to the Main BIOS Setup Menu.

Memory Cache Sub-Menu

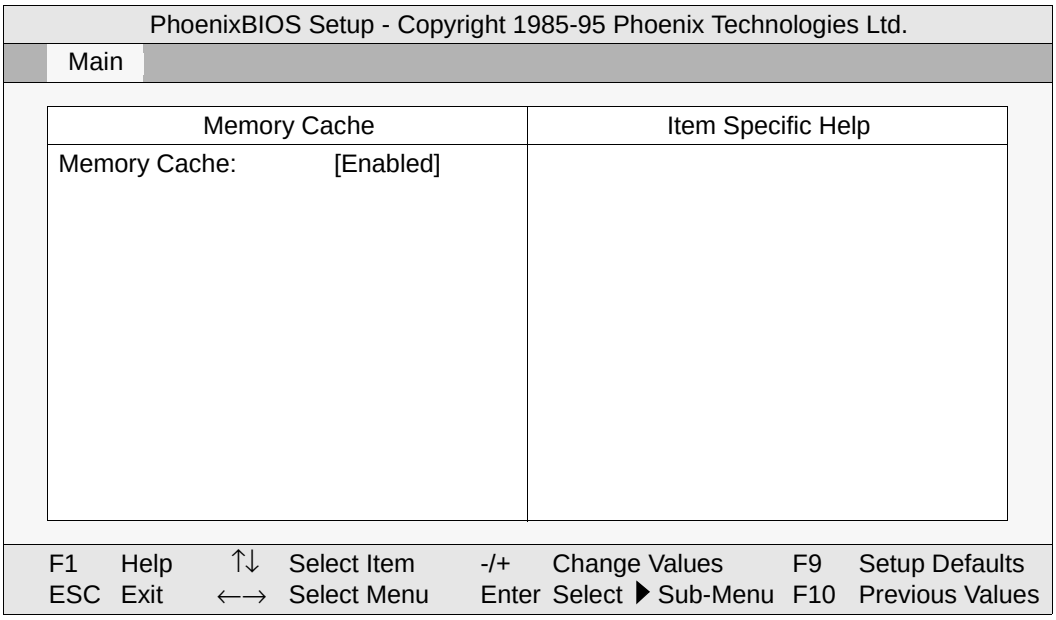


Figure 3-3. Memory Shadow sub-menu

Field	Description
Memory Cache	Enables or disables the Level 1 (L1) cache. The default is “Enabled”.

Memory Shadow Sub-Menu

The term “Memory Shadow” refers to the technique of copying information from ROM into RAM and accessing it in this alternate memory location. The Memory Shadow Sub-Menu is discussed below.

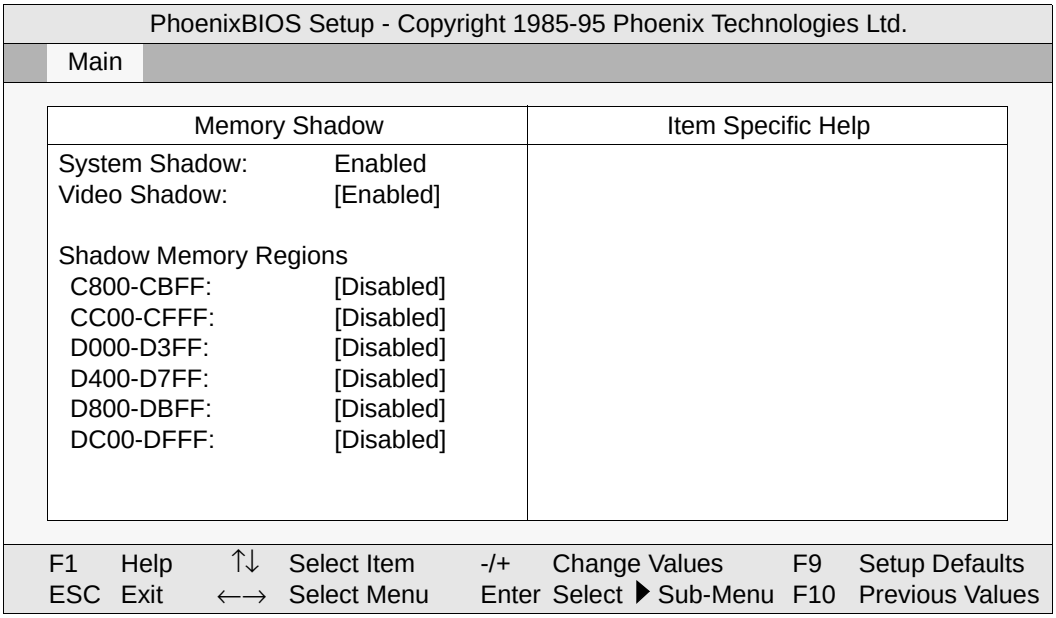


Figure 3-4. Memory Shadow sub-menu

About Shadow Memory Regions

The shadow regions should be used only if an EXM module is installed in the system that contains a BIOS ROM. Enabling shadowing for the region occupied by the ROM increases system performance.

Do not enable shadowing for the region you may have specified for installing Flash or VME disks. When these ROMs are installed, they are automatically shadowed.

Boot Sequence Sub-Menu

The Boot Sequence Sub-Menu allows you to change the boot sequence options. The following displays:

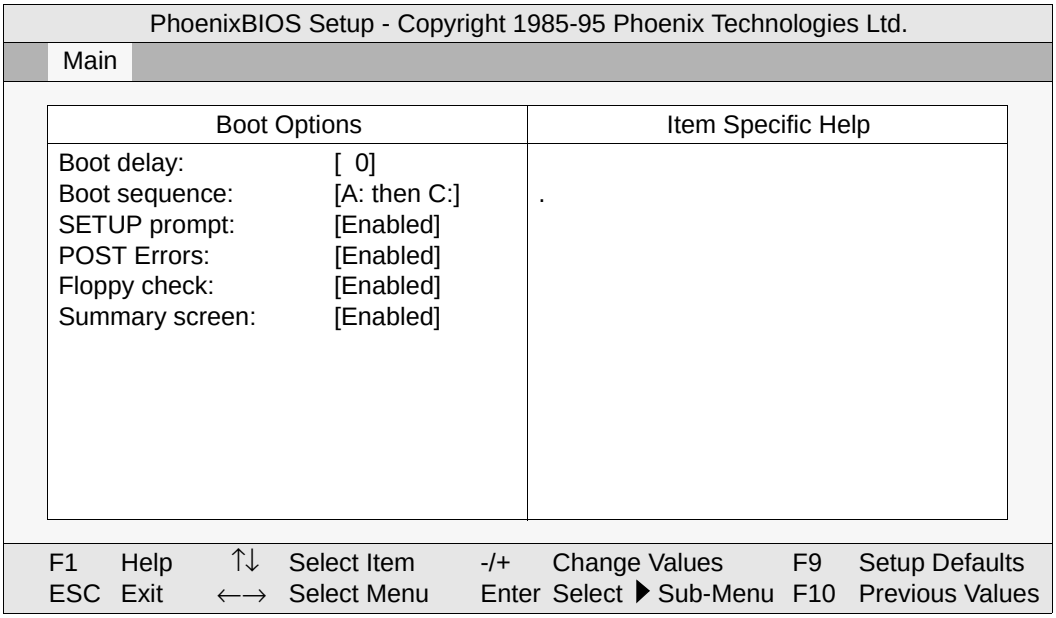


Figure 3-5. Boot Sequence sub-menu

Field	Description
Boot delay	Sets the system to delay booting for the number of seconds that you specify. This allows for long start up times on boot devices that spin up slowly. The default is zero.
Boot sequence	Defines how the system treats floppy drive A: when booting. You can boot from a floppy in the A: drive or boot directly from the fixed disk drive. To reduce the amount of time required to boot, set the boot sequence to use the C: drive only. Note that the C: drive may be an IDE drive, Flash memory, a disk image on the VMEbus, or a SCSI drive. The options are as follows: • **A: then C: (default)**: Boots from the floppy disk drive, or if no floppy is present in the A: drive, boot from the C: drive. • **C: then A**: Used to boot from the C: drive, whether Flash or IDE, or if none is present, boot from the A: drive. • **C: only**: Used to boot from the C: drive without searching for an A: drive. The setting chosen here displays in the Boot Sequence Sub-Menu prompt.
SETUP prompt	Enables or disables the message “Press F2 to enter Setup.” Even if the message is disabled, you can still press F2 to enter the Setup Menu. The default is to enable this prompt.
POST Errors	Pauses the boot process if the system encounters error messages. Otherwise, the system continues to attempt to boot despite any startup error messages that display. The default is to enable this option.

Field	Description
Floppy check	Enables or disables the floppy drive search during the boot. To speed up booting, you can disable the floppy check. It is still possible to boot from the A: drive even with the floppy check disabled. The default is to enable the floppy check.
Summary screen	Enables or disables a summary of the system configuration, which displays before the operating system starts to load. To save time, you can disable the summary screen. The default is to enable the summary screen display.

About Drive Letter Assignment

The BIOS determines the boot device algorithmically. First it determines where the floppy drive fits into the sequence; however, for simplicity here, assume no A: drive. The BIOS starts by determining if an IDE controller is enabled. If so, this becomes the C: drive and is expected to be the boot device.

The BIOS searches memory for enabled mass-storage devices, and builds a device table. The first device it finds is the C: drive, and thus the boot device. The second device found is the D: drive; only two physical drives are supported.

If an IDE drive is specified in the BIOS setup, it becomes the C: drive. Next, the BIOS looks for BIOS extensions at addresses C8000, CC000, D000, D4000, D8000 and DC000 for more devices, up to the maximum of two.

There are many different boot options. Two supported directly on the EPC-8A hardware and BIOS are booting from a Flash ROMdisk and booting from a VME ROM disk. For more information on booting from a VME ROMdisk or a Flash ROMdisk, refer to setups in the Advanced Menu and Appendix G, XFORMAT Software.

Keyboard Features Menu

Use this sub-menu to enable or disable various keyboard features.

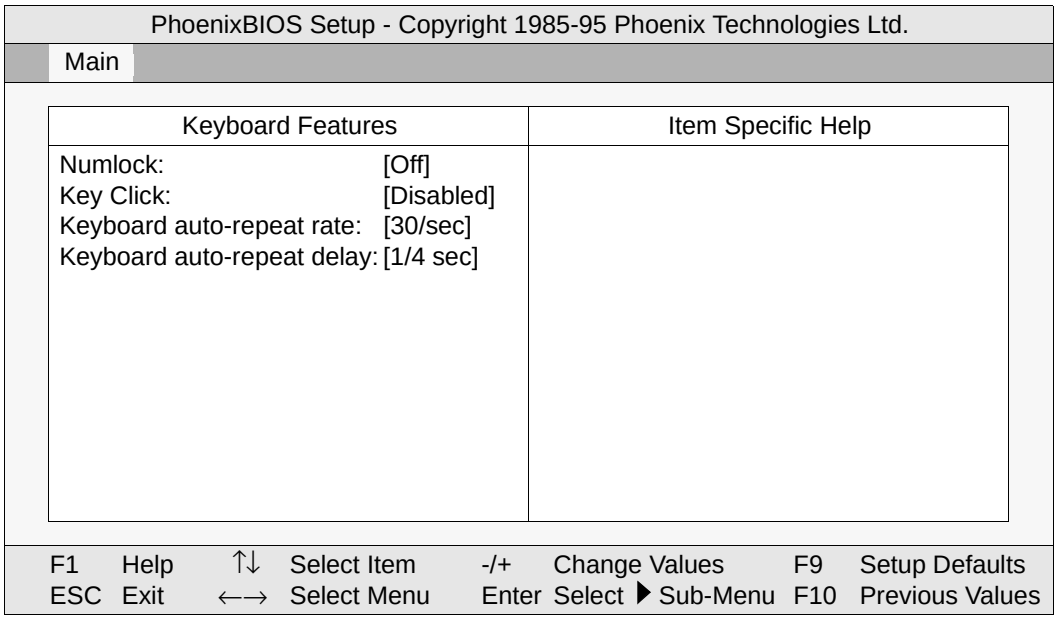


Figure 3-6. Keyboard features sub-menu

Field	Description
Numlock	Enables or disables the keyboard's Numlock feature. This enables the use of the keypad numbers. The default is to automatically disengage the Numlock key at boot-up.
Key Click	Enables or disables the key click feature on the keyboard. If enabled, the keyboard produces an audible click each time a key is pressed.
Keyboard auto-repeat rate	Sets the auto-repeat rate if holding a key down on the keyboard. The rates are from 2–30 per second.
Keyboard auto-repeat delay	Sets the delay between when a key is pressed and when the auto-repeat feature begins. Options are 1/4, 1/2, 3/4, and one second. When finished with this menu, press ESC to exit back to the main BIOS Setup Menu.

Advanced Menu

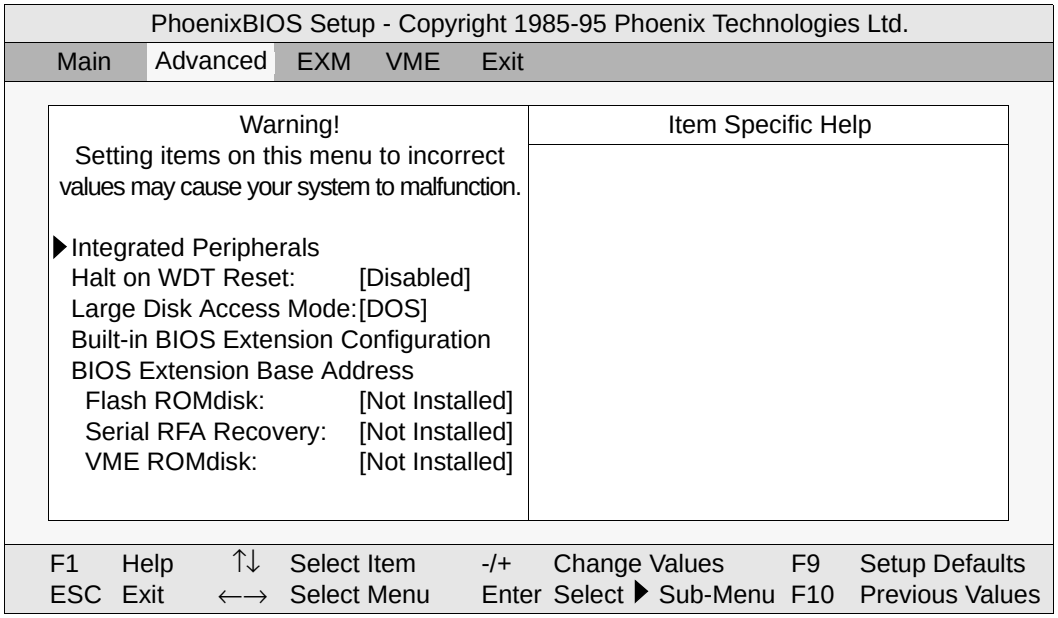


Figure 3-7. Advanced menu

The Advanced Menu contains settings for integrated peripherals, memory shadow, watchdog timer, large disk access mode, and setting the Flash and VME ROMdisk BIOS extension base addresses.

Field	Description
Integrated Peripherals Sub-Menu	Selects the Integrated peripherals sub-menu, to configure the COM and LPT ports. This does not configure Ethernet, video, or Flash memory. For more information, turn to the section concerning the Integrated Peripherals Sub-Menu.
Halt on WDT (watchdog timer) Reset	Sets the action for the BIOS to take following a hardware reset condition caused by a timeout of the watchdog timer on the EPC-8A. When such a timeout occurs, first the system generates a VMEbus SYSRESET. The EPC-8A then either halts, displaying an error message, or simply reboots, depending on the entry here.
Large Disk Access Mode	If using a drive larger than 528 Mbytes, set this to DOS if you are running DOS, or set this to Other if using a different operating system.

Field	Description
Flash ROMdisk	Enables Flash memory on the EPC-8A. This must be selected for the Flash memory to appear as a drive. The base address you select defines where the Flash ROMdisk BIOS extension is installed. Options include: DC000–DFFF0h D8000–DBFF0h D4000–D7FF0h D0000–D3FF0h CC000–CFFF0h C8000–CBFFFh Not Installed **Important**: You cannot have an IDE drive if the resident Flash memory is the boot device. For more information, see About Drive Letter Assignment on page 18.
Serial RFA Recovery	Selects the base address of the Flash ROMdisk (also known as RFA or Resident Flash Array) serial update BIOS extension. Use this option to update a corrupt Flash ROMdisk from the COM1 port. Options include: DC000–DFFF0h D8000–DBFF0h D4000–D7FF0h D0000–D3FF0h CC000–CFFF0h C8000–CBFFFh Not Installed **Important**: Do not place at the same base address as other BIOS extensions.
VME ROMdisk	Enables VME memory on the EPC-8A. This must be selected for the VME memory to appear as a drive. The base address you select defines where the VME ROMdisk BIOS extension is installed. DC000–DFFF0h D8000–DBFF0h D4000–D7FF0h D0000–D3FF0h CC000–CFFF0h C8000–CBFFFh Not Installed Make sure you do not choose an address used for Flash devices. The address chosen affects whether or not this is the boot device.

Field	Description
VME Scan Range	Used for VME booting. To boot from VME: 1. Use XFORMAT with the /F option to first create a file that looks like a FAT/boot record. 2. Upload the file to VME memory on a 100000h boundary in one of the memory ranges below. 3. Set the scan range to match where in VME memory the file was placed. 4. Be sure to enable the VME ROMdisk BIOS extension to enable scanning. 00000000–FFF00000 A32 00000000–00F00000 A24 FF000000–FFF00000 A24

Integrated Peripherals Sub-Menu

Use the options in this sub-menu to configure the COM and LPT ports.

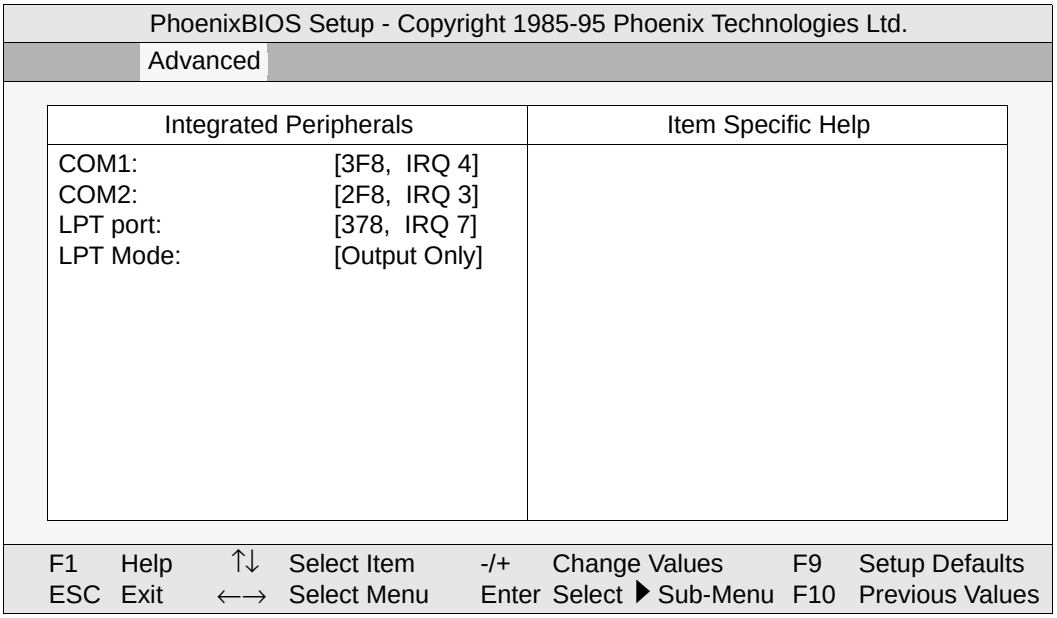


Figure 3-8. Integrated Peripherals sub-menu

Field	Description
COM port	Enables or disables the COM1 and COM2 ports. The defaults include: COM1 3F8 and IRQ4 COM2 2F8 and IRQ3
LPT port	Enables or disables the LPT port. The default is 378 and IRQ 7.

Field	Description
LPT Mode	Options include: • **ECP**: Enables the parallel port's Extended Capabilities Port (ECP). • **Bi-directional**: Enables the parallel port as bi-directional. • **Output only**: Enables the parallel port as output only. When you are finished, press ESC to exit back to the Advanced Menu.

EXM Menu

Use this menu to set up the optional EXM expansion modules in your EPC-8A. Enter the EXM-ID, plus option byte information for OB1 and OB2. This information is found in the hardware reference manual shipped with each EXM expansion module.

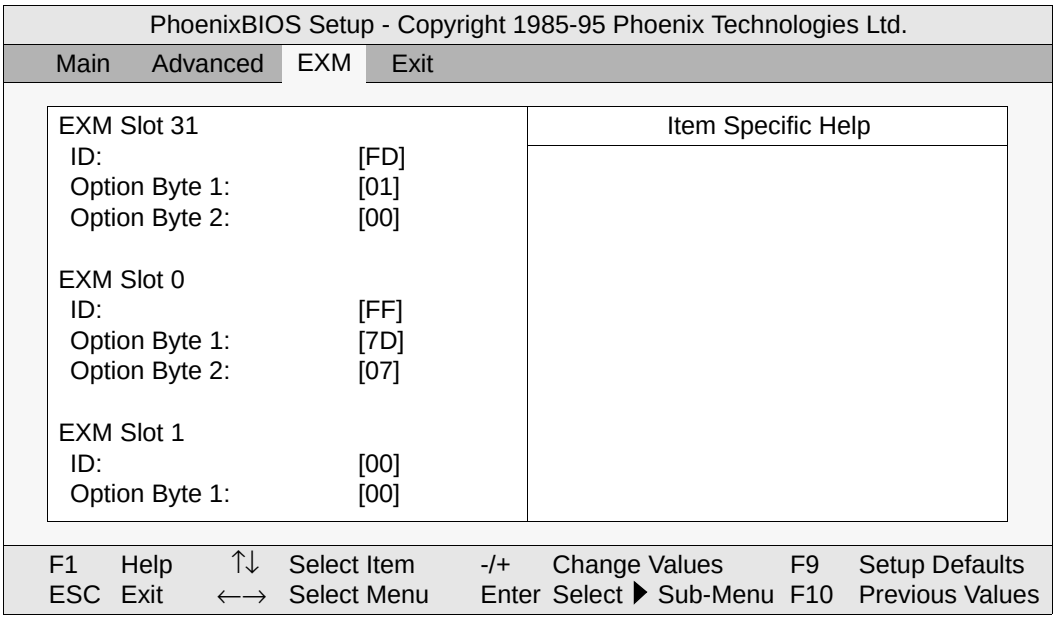


Figure 3-9. EXM menu

Note the following when installing EXMs:

SLOT Indicates the EXM slot in which the EXM is installed.

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

OB1/OB2 Two “option” bytes of configuration information.

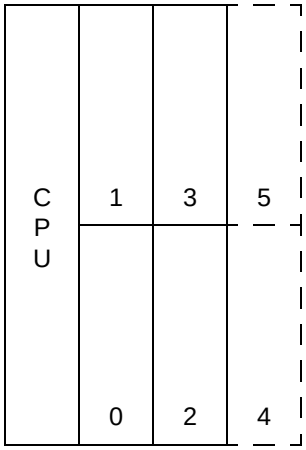


Figure 3-10. Slot Numbering

All slots *not* occupied by an EXM module should show an ID of FF and OB1/OB2 of 00 00 indicating that no EXM is present.

Field	Description
Slots	Identifies the slot to which the configuration information applies: • **EXM Slot 31**: Identifies the Flash memory address. Slot 31 refers to the optional resident Flash memory. The ID should be set to FF if your EPC-8A does not have this feature. If present, then set the ID to “FD” and Option Byte 1 to “01”. • **Slots 0 through 8**: Specifies configuration information for EXM expansion modules. Consult the appropriate EXM manual for the correct configuration information for each EXM expansion module installed. Note that most EXM hardware reference manuals depict a different BIOS setup from the EPC-8A. The ID/OB1/OB2 information is valid. When using EXMs with configurable interrupts, DMA channels, I/O addresses, and/or memory addresses, avoid conflicts with built-in functions of the EPC-8A. Guidelines are: - If an interrupt is needed, use IRQ3, IRQ5, IRQ9, IRQ12, or IRQ15. IRQ7 can be used if the printer port is not being used. IRQ3 should not be used if the COM2 port is being used. - Use DMA channels 1, 3, 6, and 7. - Do not select I/O addresses that conflict with those in the EPC-8A. A complete list appears in Appendix A. For instance, I/O addresses in the 300–33F range can be used. - If the EXM needs to use upper memory addresses, they must be in the 0C8000–0DFFFF range. Note that the E-page is used for VMEbus access and is not available.

Field	Description
ID	Identifies the system's EXMs. Enter the EXM-IDs for the EXMs you intend to install in this system. You can install up to six EXMs.
Option Byte 1/ Option Byte 2	Specifies option byte configuration information. Each EXM expansion module has values you must enter for the option byte 1 and option byte 2 configuration data. When you are finished with this menu, press the right arrow key to move to the VME Menu, or press ESC.

VME Menu

Use the options in this menu to configure the VMEbus interface and addressing.

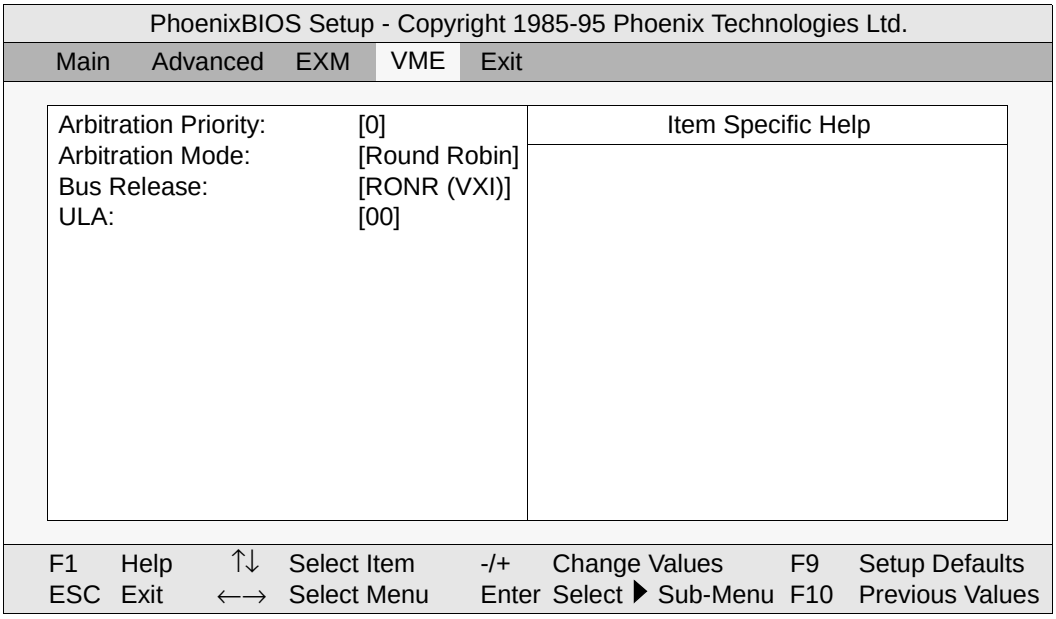


Figure 3-11. VME menu

Field	Description
Arbitration Priority	Sets the arbitration priority. Possible values include 0, 1, 2, and 3, with 0 indicating the highest priority. This determines which VMEbus request signal the EPC-8A uses when it accesses the VMEbus.
Arbitration Mode	Selects the arbitration mode. Possible values are "Round Robin" or "Priority." This determines how the EPC-8A performs bus arbitration when it is the VMEbus Slot-1 controller. If round-robin is selected, the four bus request lines have equal priority. If Priority is selected, bus request 0 has highest priority.
Bus Release	Selects the Bus Release mechanism. Possible values are RONR (VXI) and ROR (VME).

Field	Description
ULA	Selects the unique logical address (ULA). Possible values range from 00 through FF. The logical address controls where in the VME A16SD space the EPC-8A extended registers are mapped. If the ULA=0, the registers are at C000. If the ULA=FF, the registers are at FFF0. In general, the registers map to [C000 + (ULA*40)] (all numbers in hexadecimal).

Exit Menu

Use the options in this menu to save and exit, or abandon your changes and exit to the system.

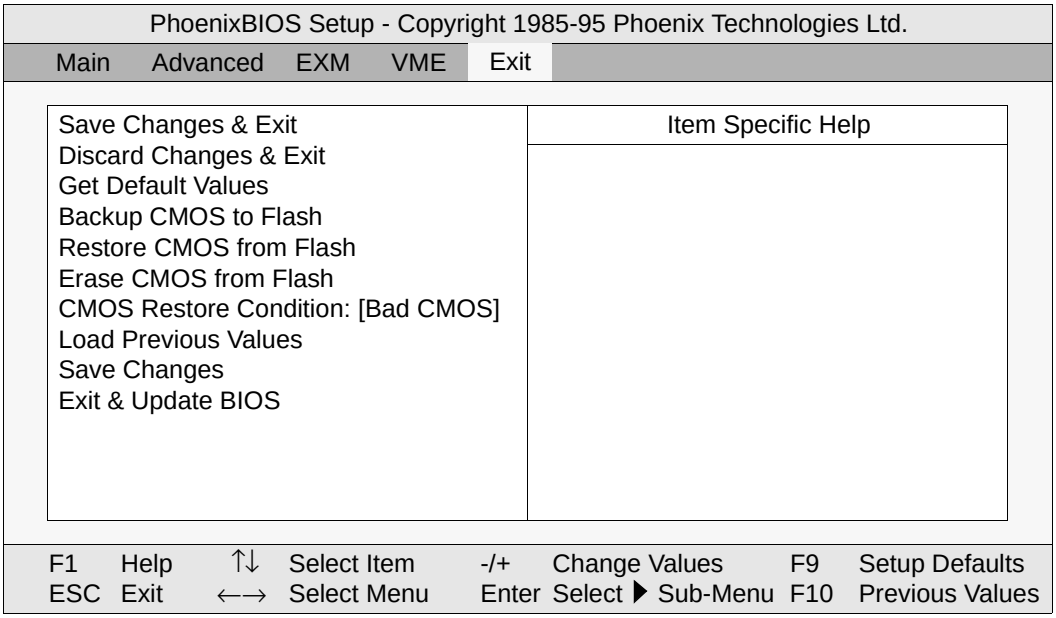


Figure 3-12. Exit menu

Field	Description
Save Changes & Exit	Saves the values you just entered and exits to load the operating system. The new values are loaded, and you exit and reboot.
Discard Changes & Exit	Discards the changes you just made and revert to the BIOS as it was before you started. The system boots with the old values.
Get Default Values	Resets the BIOS values to the original, default values set at the factory, before any suppliers or other end users made changes.
Backup CMOS to Flash	Immediately saves current Setup settings to CMOS RAM and into FBD main block #1.
Restore CMOS from Flash	Immediately restores CMOS RAM and updates current Setup settings from FBD main block #1.
Erase CMOS from Flash	Immediately erases the CMOS image from the flash.
CMOS Restore Condition	Determines under what conditions the System BIOS restores CMOS RAM from FBD main block #1 when booting. Restore conditions include: Always, Never, and CMOS Corruption (default).
Load Previous Values	Loads the system with the previous values before this editing session started. You do not exit.

Field	Description
Save Changes	Saves the edits you made during this session but does not exit.
Exit & Update BIOS	Updates the BIOS from a floppy disk. **Note**: Select this exit option only if you obtained BIOS update replacement software from your supplier and reviewed the documentation and procedures provided with that distribution. If you select this option by mistake, changes made to the BIOS are lost unless already saved using the Save Current Values option. The system automatically seaches for the update program that should be on the floppy disk inserted in drive A. If no floppy exists, two series of beep codes sound: a long and two short beeps, followed by three short beeps that repeat. Cycle power to reset the system to its previous state.

4 Theory of Operation

Overview

The EPC-8A is a PC/AT compatible computer with standard PC peripherals, a VMEbus interface, and modular expansion capability via the EXM expansion interface. The Block Diagram on page 30 provides an diagrammatic overview of the system functional blocks. Most of the standard functions of the PC architecture are embodied in the RadiSys R400EX chipset. DRAM and VGA are interfaced to the processor by a 32-bit local bus. The customary PC peripheral interfaces for keyboard, two serial ports, a parallel port, and a battery-backed real-time clock are connected by the EXM expansion interface which is electrically similar to the standard PC ISA bus. An Ethernet controller is built onto the EPC-8A board.

The VMEbus interface includes special VME byte swapping hardware to aid the software when dealing with different processor and memory organization. This is described in detail later in this chapter.

The EPC-8A maps a standard set of VXI configuration registers onto the VMEbus A16 space. These are dual-ported and accessible both by other VMEbus modules and the EPC-8A in its I/O space.

Resident Flash memory (on the EPC-8A board) is I/O mapped and appears as an EXM expansion module as if in EXM slot 31. SRAM is memory mapped. The PC BIOS is shadowed into main memory at start-up.

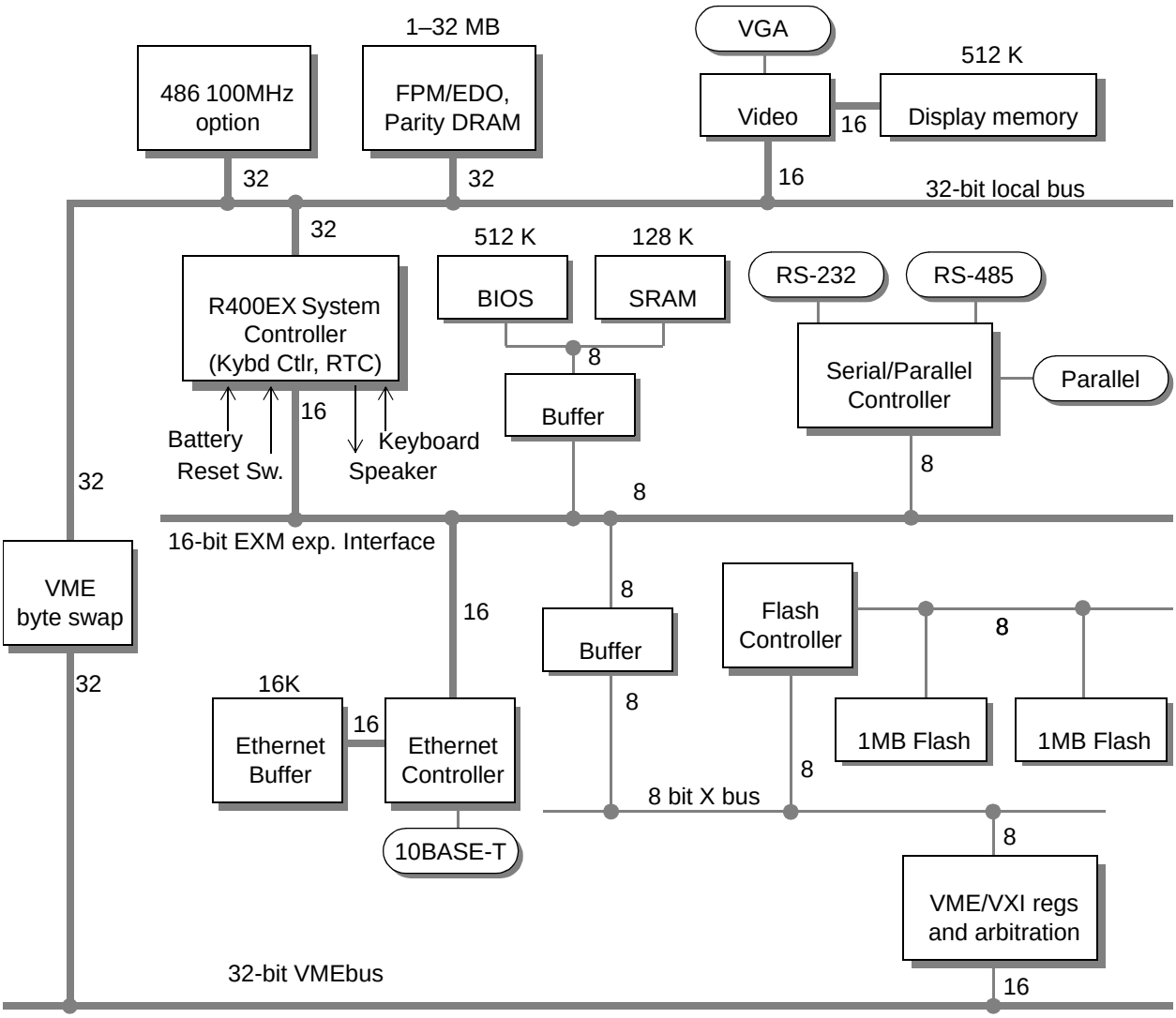


Figure 4-1. Block Diagram

Processor board

The EPC-8A processor board conforms with the VMEbus standard 6U form-factor.

Processor and Coprocessor

The Single-slot EPC-8A contains an Intel486 DX2 (32-bit bus interface) running at 66 MHz, with a built-in math coprocessor.

The Dual-slot EPC-8A contains an Intel486 DX4 (32-bit bus interface) running at 100 MHz, with a built-in math coprocessor.

Memory

The following memory options are supported: 8 Mbytes, 16 Mbytes, and 32 Mbytes. One SIMM socket is available. For memory upgrade instructions, see Appendix D, *Memory*.

Memory Map

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

- Addresses between 0 and 1 Mbytes, which are largely defined by the IBM PC/AT architecture.
- Addresses between 1 Mbytes and 256 Mbytes, which largely depend on how much DRAM is installed in the EPC-8A.
- Addresses above 256 Mbytes, which provide direct mapping to the VMEbus with a variety of address modifiers and byte orderings. For more information about this feature, see Chapter 5, *Programming the VMEbus Interface*.

Table 4-1. Memory at addresses between 0 and 1MB (0FFFFFh)

Range		Content
000000	– 09FFFF	DRAM (first 640 KB)
0A0000	– 0BFFFF	Almost always used by a video controller as video RAM. If the onboard VGA is populated and enabled, then this memory is accessed over the CPU local bus. Otherwise, it is mapped to the EXM interface.
0C0000	– 0C7FFF	Write-protected DRAM containing video BIOS
0C8000	– 0DFFFF*	Uncommitted; mapped to EXM interface.
0E0000	– 0EFFFF	Either is mapped to EXM interface or is window into the VMEbus. Mapping controlled by bit-1 of register 8102h.
0F0000	– 0FFFFF	Write-protected DRAM containing system 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 E is installed or may be used for BIOS extensions.

Table 4-2. Extended memory address space for a 4MB EPC-8A

Extended memory	Range		Content
8MB EPC-8A	00100000	007FFFFF	7 Mbytes DRAM extended memory
	00800000	00FFFFFF	Uncommitted; mapped to EXM interface
16MB EPC-8A	00100000	00FFFFFF	15 Mbytes DRAM extended memory
32MB EPC-8A	00100000	01FFFFFF	31 Mbytes DRAM extended memory

BIOS ROM and ROM Shadowing

The EPC-8A contains a Flash boot device (chip) as its BIOS ROM. The BIOS ROM is mapped into the top of the processor's 32-bit address space. The BIOS consists of an 16K boot block and a System BIOS combined with a VGA BIOS in a 128 Kbyte partition. The Flash boot device is memory addressed and resides in the last 512 Kbytes of system memory at address FFF80000H to FFFFFFFFH. The layout is described in Figure 4-2.

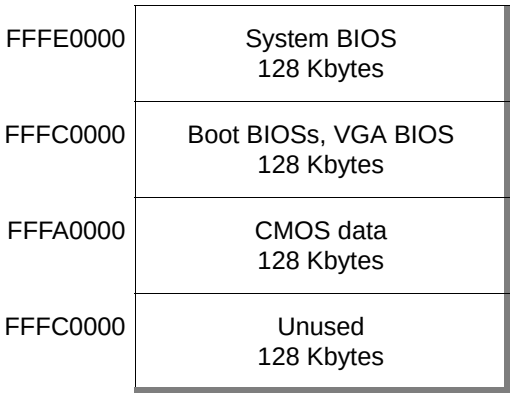


Figure 4-2. Flash Boot Device Memory

The BIOS initialization software copies the ROM contents into DRAM (a process called *shadowing*) at addresses 0F0000–0FFFFF (also called the "F" page). The VGA BIOS is copied into 0C0000–0C7FFF of DRAM. If the user has configured an EXM VGA card (For example, an EXM-13A) and enabled it, the EXM's BIOS is used for the copy. Otherwise, the internal VGA BIOS is copied and enabled.

After copying into these areas, the BIOS write-protects them. Subsequent writes to these areas complete successfully but do not alter the data in DRAM. However, the proper sequence of writes alters data in the Flash boot device (BIOS ROM) itself, if the jumper is installed.

There are two parameter blocks, each 8 Kbytes in size. Both blocks are used to store BIOS code.

CMOS Backup and Restore

CMOS memory is backed up to and restored from Main Block 2 of the FBD as determined by the settings in the BIOS Setup Exit Menu. This allows you to save your settings to nonvolatile flash memory and to specify the conditions under which CMOS is to be restored from the FBD. For more information, see the Exit Menu section of Chapter 3, *BIOS Configuration*, on page 26.

Jumper JP2 (-BIOS) must be installed for Save to work. Be sure to remove the jumper after using this function.

Video Controller

The EPC-8A hardware includes a VGA graphics controller implemented using the Chips and Technologies 65545. This is connected to the CPU local bus to give the best possible graphics performance. VGA memory is 512 Kbytes, resulting in the following resolutions:

Dimensions	Resolution
640 x 480	16 or 256 colors
800 x 600	16 or 256 colors
1024 x 768	16 colors

The BIOS does not enable (using bit 2 of register 8102h) the VGA controller if another VGA controller is enabled on the EXM expansion interface.

Ethernet Controller

The EPC-8A contains an on-board Ethernet controller connected through the 16-bit EXM expansion interface, which is compatible with Western Digital 8013, Novell NE2000 and NE2000+ cards through the use of National Semiconductor's DP83905 (AT/LANTIC chip). The default configuration for the Ethernet port is as a WD8013-compatible card. The I/O base address is 240. Interrupts are signaled on IRQ5. These parameters can be changed by running the AUTOSET.EXE program, optionally available from your supplier. The possible alternative interrupts are IRQ3, IRQ9 and IRQ15.

The EPC-8A Ethernet port has two 8 Kbytes x 8 RAM chips for packet buffering and a 10BASE-T interface. You cannot use eight-bit network drivers with the EPC-8A.

For more information about the AUTOSET.EXE program, refer to Appendix H, *AUTOSET Software*.

Resident Flash/SRAM Memory

The EPC-8A can optionally be built with 2 Mbytes of Flash memory and 128 Kbytes of Static RAM resident on the processor board.

Resident Flash Memory

The resident Flash memory is accessed via 8-bit read/write registers and physically appears on the EXM expansion interface as if it is in slot 31. It is enabled as if it were a standard RadiSys EXM expansion card in that slot.

The EPC returns an ID of FD to a read of address 100. This is the same ID that would be returned from a RadiSys EXM-2A card. You can use the optional **XFORMAT.EXE** utility to format and load the resident Flash memory under MS DOS. The EXM-2A and the resident Flash memory cannot both be used in a single EPC-8A system. Note: You must set jumper JP2 (-FLASH) before the Flash can be written. Jumper locations are described in Figure 2-1.

SRAM

The battery-backed SRAM is memory mapped. The resident SRAM resides on the EXM expansion interface and, when populated, can be accessed at 0xFX3XXXXX or 0xFXBXXXXX. If there is less than 16 Mbytes of DRAM, it may also be accessed at 0x00BXXXXX. A battery low indicator (low when 0) is returned in bit 0 of register 8387.

Watchdog Timer

The watchdog timer is a binary counter which, upon overflow, signals a watchdog timer event. The counter causes a watchdog event after approximately 125 mS, 1 second or 8 seconds (depending on the value of FWDT and SWDT, bits 2 and 1 in register 815Dh) if the application software does not reset the timer.

An I/O read to address 815Dh resets the counter. If WDTR (bit 3 of register 815Dh) is set, the following occurs in response to a timeout event:

WDT (bit 3 of register 8154h) is set. A local “warm” hardware reset occurs. Bits 1, 2, and 3 of register 815Dh are cleared to prevent the watchdog timer from expiring on a warm reset that is initiated from a source other than a watchdog timeout. The BIOS must set the BTOE bit. VME SYSFAIL* is also asserted. When exiting a hardware reset condition, the BIOS can check the WDT bit. If this bit is set (0), then a watchdog timeout caused the hardware reset (as opposed to SYSRESET or power-on reset). Then depending on the value of a setup option the BIOS either HALTs the CPU or allows the boot process to continue. At this point, software may deassert the VME SYSFAIL* condition by reading the register at 815Dh.

Note that a watchdog hardware reset results in are “warm” hardware reset. A warm hardware reset clears all register bits except for the upper four bits of the Configuration register (these control Slot-1 arbitration functions) and bits 4 and 6 of the Module Status/Control registers (these control bus timeout function and watchdog timer functions). A warm reset does clear WDTR (bit 3 of the Module Status/Control register) to allow the hardware to be released from the warm reset state, but SYSFAIL continues to be driven until the WDT bit is cleared by either reading the Module Status/Control Register or by a power-on reset.

If WDTR is clear, WDT mask (bit 3 of register 8155h) enables an interrupt if a timeout event occurs (SYSFAIL is not driven). The clock is disabled to the counter if the interrupt is pending and not serviced. Service of the interrupt is signaled to the counter by reading register 815Dh. This resets the counter value and resumes counting. The interrupt is signaled on IRQ10. The timer event also clears WDT bit in the BES register (bit 3 of register 8154h).

Application software that utilizes this timer should take care to reset the counter just prior to enabling the interrupt bit in register 8155h. This inhibits a spurious timer event from occurring just after enabling the timer.

Battery

The battery powers the CMOS RAM and TOD clock when system power is not present. At 60ºC, 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 60ºC, the battery is estimated to have a life of 10 years.

If system power is present, the VME +5V STDBY voltage also powers the CMOS RAM and TOD clock. This is done with isolation diodes, so that either the onboard battery or the VME voltage supplies power. Neither power source affects the other.

The 3.0V lithium battery supplied with the EPC-8A is a Panasonic BR2330 “coin cell” or equivalent. It is mounted behind the reset button at the top of the EPC-8A. Should the battery fail, you may obtain and install a replacement.

Replacing the battery is a simple task. However, removing the battery invalidates the CMOS setup parameters if you do not put the new battery into the empty socket within about 60 seconds. It is recommended that all setup parameters be written down while the battery is still good, or saved using the CMOS Save and Restore feature. For a method to backup CMOS parameters into a Flash memory feature block, see the description of the CMOS Backup and Restore feature on page 32.

To replace the battery, lay the EPC-8A flat on an ESD-protected surface. Remove the SIMM module, following proper ESD protection procedures. Now slide the battery out

and quickly place the new battery into the empty socket. Replace the SIMM module and reinsert the EPC-8A into the system.

Peripheral Ports

The I/O address and IRQ of the peripheral ports are determined by the CMOS parameters established by default and modified via BIOS Setup screens.

RS-232 Port

The RS-232 port is a standard PC COM port based on the 16550 architecture. It is normally configured as COM1 (I/O address = 3F8–3FF, IRQ4). “COM1” is an alias for an address and is determined by the first COM port found. If not needed, COM1 can be disabled in the BIOS Setup screen to free up the I/O address and interrupt for usage by other expansion products.

RS-422/485 Port

The modified RS-422/RS-485 port is normally configured as COM2 (I/O address = 2F8–2FF, IRQ3). This port is also a standard PC COM port based on the 16550 architecture. If not needed, COM2 can be disabled in the setup screen to free up the I/O address and interrupt for usage by other expansion products. The RXD and TXD signals are RS422 compatible with differential receiver and driver, respectively. Handshake signals RTS, DTR, CTS and DSR are single-ended RS422 compatible. Outputs TXD+ and TXD- are tristated if RTS is unasserted for RS485 multidrop compatibility. The port has 10K ohm termination.

Parallel Port

The printer port is a standard PC printer port. The parallel port supports bi-directional communication compatible with the PS/2 definition. It is configured as LPT1 (I/O address 378–37F, IRQ7). If not needed, LPT1 can be disabled in the setup screen to free up the I/O address and interrupt for usage by other expansion products.

Keyboard

The keyboard controller is integrated into the RadiSys chipset. It can interface to most standard PC keyboards with a PS/2-style connector. An adapter cable is shipped with the EPC-8A to allow use of standard PC/AT keyboards with larger 5-pin connectors.

Front Panel LEDs

The EPC-8A has three LEDs in the top left corner of the front panel. These LEDs are described below:

RUN — This LED is lit whenever a write access to DRAM is made. This provides a normally lit LED indicating that the CPU is operating. This LED is not lit if the CPU is halted or is executing entirely out of the on-chip cache.

FAIL — This LED is lit whenever the VME SYSFAIL line is asserted if the EPC-8A is jumpered to be the Slot-1 controller. This occurs independent of which VME board is driving the SYSFAIL line. It comes on whenever the system receives a hardware reset and remains on until the initial power-on self-tests complete.

TEST — This LED is lit whenever the system is running its power-on self-test, as reflected in the PASS bit in the VXI registers. If PASS is 0, then this LED is lit. This only occurs during a hardware reset.

Resetting the EPC-8A

You can reset (reboot) the EPC-8A in a number of ways. Resets are summarized in the table below, with details in the paragraphs that follow.

Table 4-3. Reset Conditions

Power-On Reset	Power < 3.0 V	Power 3–4.5 V	Front-Panel Reset Button	Ctrl-Alt-Del	VMEbus SYSRESET Asserted	Watchdog Timer
"Cold" reset	"Cold" reset	"Warm" reset	"Warm" reset	Software reset	"Warm" reset	"Warm" reset
POST runs	POST runs	POST runs	POST runs	No POST	POST runs	POST runs
R400EX, all VXI registers reset	R400EX, all VXI registers reset	R400EX, most VXI registers reset	R400EX, most VXI registers reset		R400EX, most VXI registers reset	R400EX, most VXI registers reset
		bits 4–7 of 8102h not reset; bits1,2,4 & 6 of 815Dh not reset	bits 4–7 of 8102h not reset; bits1,2,4 & 6 of 815Dh not reset		bits 4–7 of 8102h not reset; bits 1, 2, 4 & 6 of 815Dh not reset	bits 4–7 of 8102h not reset; bits 1, 2, 4 & 6 of 815Dh not reset
AT. RESET generated	AT. RESET generated	AT. RESET generated	AT. RESET generated	No AT. RESET	AT. RESET generated	AT. RESET generated
SYSRESET* generated	SYSRESET* generated	SYSRESET* not generated unless bit 6 of 815Dh is set	SYSRESET* not generated unless bit 6 of 815Dh is set	No SYSRESET*	SYSRESET* not generated unless bit 6 of 815Dh is set	SYSRESET* generated
						SYSFAIL asserted until watchdog timer is reset

Power-off, Power-on

Known as a cold hardware reset. This causes all boards in the VMEbus to reset. The system runs the power-on self-test and reboots the OS.

Power low

"Warm" hardware reset. When power is detected between ~3.0 and ~4.5V, the system runs the Power On Self Test (POST) and reboots the operating system. When power is less than ~3.0V, the system performs a cold hardware reset.

Front-panel Reset button

"Warm" hardware reset. The Reset button causes the EPC-8A to perform a hardware reset. The system runs the power-on self-tests and reboots the operating system.

Ctrl+Alt+Del

"Warm" software reset. This keyboard sequence is also called a "warm boot". The EPC-8A 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

"Warm" hardware reset. The EPC-8A can be software-configured to respond or not respond to the VMEbus SYSRESET* line. Asserting bit 7(SRIE) of register 8144h allows the VME SYSRESET* signal to reset the EPC-8A. The reset semantics are the same as the front panel reset.

Watchdog Timer

"Warm" hardware reset. Same as a front panel reset button except that SYSFAIL is asserted until the watchdog timer is cleared.

Register State after Reset

A "cold" or hardware reset of the EPC-8A (not a keyboard Ctrl+Alt+Del reset) clears all of the register bits to 0, except for RELM, ARBM, and ARBPRI, 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.

VME/VXI Soft RESET state and SYSRESET

"Soft Reset" is a capability that allows another VME master to disable the EPC-8A's connection to the VME bus, without interrupting (or resetting) the 486 processor on the EPC-8A.

The Soft Reset state is entered when the SRST bit is set. In this state the EPC-8A removes any asserted interrupts (clears the Interrupt Generator register, disables its VME master logic, asserts both the VMER and BERR sticky bits in the VME Event State register, disables watchdog timer resets and interrupts, and clears the PASS bit SYSFAIL is also asserted if the NOSF (SYSFAIL inhibit) bit is clear. The Slot-1 arbitration and control logic and the bus timeout function, if it is enabled, is unaffected by the SRST bit. Software on the EPC-8A can detect that another board has set the EPC-8A into Soft Reset state by several different methods:

1. Enable the interrupt events associated with either the VMER bit or BERR bit.
2. After each VME master access poll the BERR bit. If this bit is set either a true bus timeout occurred (VMER is not asserted in this case) or the Soft Reset state has been entered (both VMER and the BERR bits are asserted).

The Soft Reset state can be exited by a push-button reset, a power-on reset, by simply writing a 0 to the SRST bit, or by the assertion of SYSRESET when the SRIE bit is also set. When SRST is cleared by writing the bit to 0, the VMER and BERR bits should also be cleared by writing to the VME Event State register, VME master accesses are again enabled (if they were enabled prior to SRST being asserted), and the watchdog functions,

that were enabled prior to SRST being asserted, resume (with the counter starting in the cleared state).

The EPC-8A's reaction to SYSRESET* being asserted on the VME bus depends upon whether the SRIE bit (SYSRESET Input Enable, bit 7 of the Status/Control register) SYSRESET is set or clear. When SRIE is asserted (set), the assertion of SYSRESET* results in the same "warm" hardware reset that a watchdog timer reset causes. When SRIE is not asserted, the EPC-8A responds to the assertion of SYSRESET* by placing the EPC-8A into a state almost identical to the Soft Reset state. The only difference between the two states is that the PASS bit (bit 2 of Status/Control register) is not cleared by the assertion of SYSRESET* (as long as SRIE is 0—to avoid a warm reset), but the PASS bit is cleared by the assertion the SRST bit.

Signal Register FIFO

To spell out the operation of the Signal Register FIFO and associated control bits, the operations are explained in algorithmic fashion. SIG, FSIG and LSIG are fields in the Response register. The signal FIFO, called SRFIFO henceforth, is a two-element array with indexes.

A write to VPR from the VXI is a write to the signal register (and FIFO), and does the following:

```
if (SIG && (FSIG != LSIG)) {    /* FIFO full */
  assert BERR;
}
else {
   if (SIG) LSIG = !LSIG;
   SRFIFO[LSIG] = data_bus;
...SIG = 1
}
```

A read from SRFL returns the low-order byte of SRFIFO(FSIG). In all cases of accesses to SRFL and SRFH, if SIG = 0 (empty FIFO), the result is an access to SRFIFO(0). A read from SRFH returns the high-order byte of SRFIFO(FSIG), and does the following:

```
if (FSIG == LSIG)
  {SIG = LSIG = FSIG = 0}
else
  FSIG = !FSIG;
```

Writes to SRFL and SRFH (I/O address 8148h,8149h) are identical to reads except for the direction of the data flow (For example, writing to SRFH alters the high-order byte of SRFIFO(FSIG) and does the above action). The only apparent purpose of these writes is testability.

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–105h 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 3, BIOS Configuration in the section on the EXM Menu.

The EXM expansion interface is provided on rows A, C, and D of the EPC-8A's 4-row DIN P2 connector. The subplane carries the EXM interface to other modules, such as to EXM modules and the EXP-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.

VME Interface

Connectivity

The EPC-8A module connects to the VMEbus J1 connector directly and uses all of the defined VMEbus lines except SERCLK, and SERDAT. Connection to the J2 connector is through the subplane's 4-row DIN connector B row. Connections to the VME J2 backplane are on the B row and are power and ground, address lines A31–A24, data lines D16–D31.

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 bus grant daisy chain.
- Provides Bus time-out function.

When configured as the Slot-1 controller, the EPC-8A detects and terminates data transfer bus timeout. 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-8A asserts the VMEbus BERR signal until both data strobes are deasserted. The duration of the VMEbus timeout counter is approximately 100–120 µsecs.

When the EPC-8A is configured as the slot-1 controller, this timeout cannot be disabled and the duration cannot be changed.

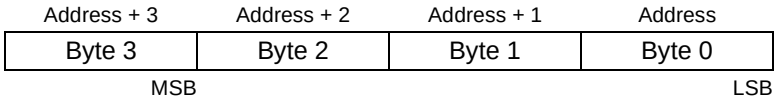
VMEbus Access

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

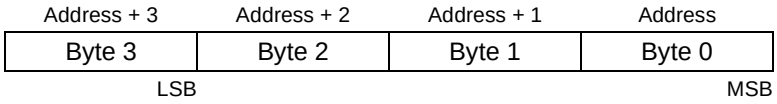
Byte Ordering

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-8A 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 (8151h) When using direct memory mapping, the order is address-range dependent (For example, 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.

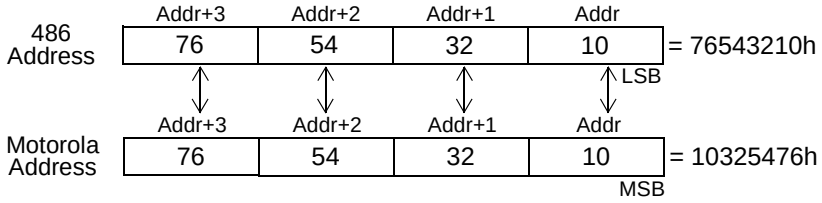


Figure 4-3. Using little-endian byte ordering

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 as shown in the next diagram.

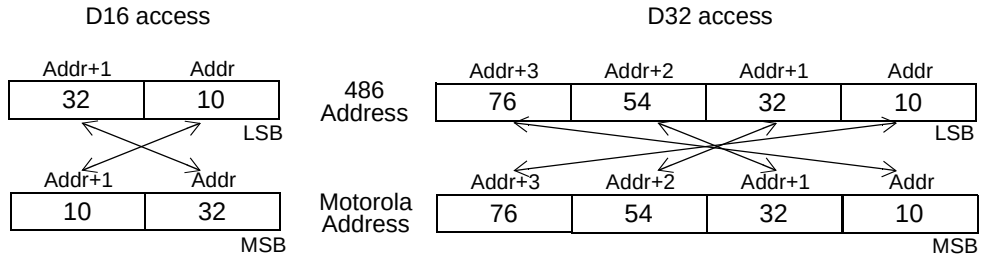


Figure 4-4. Using big-endian byte ordering

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 are "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 is "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; that both systems expect the mantissa and exponent in the same byte locations. As long as this is correct, transferring a floating-point number works 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.

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

VMEbus Interrupt Response

When the EPC-8A's Interrupt Generator register (815Fh) is used to assert an interrupt, the EPC-8A formulates a status/ID value that is transmitted on the bus as the response to a matching interrupt acknowledgment cycle. The EPC-8A acts as both a D08(O) and D16 interrupter. For D08 interrupt acknowledge cycles, the status/ID value is the EPC-8A's logical address (`11111aaa`, where `aaa` is the value of ULA as defined in port 814Ah). 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 814Bh) and the lower eight bits are the logical address.

VME Extension Registers (VXI)

EPC-8A maps a standard set of VXI configuration registers onto 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-8A's logical address. The VME-mapped registers are a subset of those defined as I/O ports in the EPC-8A. The registers are dual-ported in that they are accessible both from VME and from within the EPC-8A as ports in its I/O space.

Refer to Appendix F, *VMEbus Mapped Registers* for additional information.

Passing VME Interrupts and Events to the CPU

The diagram below shows how VME interrupts and VME events are generated and passed to the CPU:

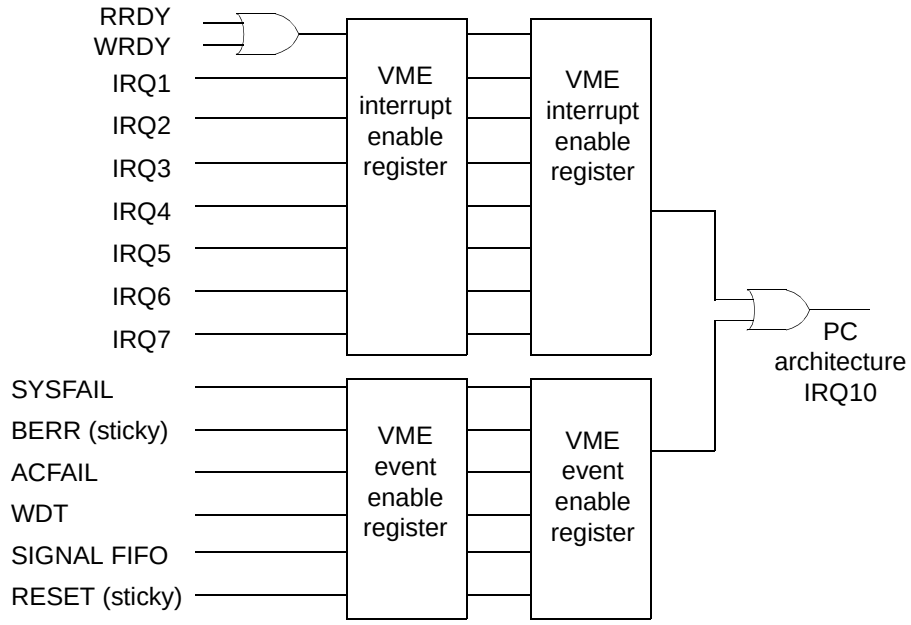


Figure 4-5. Passing VME interrupts and events to the CPU

Interrupt-causing signals are visible in two state registers. Most of these are unlatched, meaning that a read of the state register shows the actual state of the signals at the instant of the read. The exceptions are (1) BERR, which is a “sticky” bit, meaning that the bit signifies whether BERR had ever been asserted (the SBER register bit), and (2) RESET, another sticky bit. The convention used is that a 0 bit signifies an asserted (interrupting) state.

The primary purpose of the state registers is to let the interrupt handler software determine which interrupts and events generated the IRQ10 interrupt to the processor. The state registers can also be read by non-interrupt-handler software to poll for the state of these signals.

The enable registers allow one to mask selectively these 14 states. A 0 state bit and a corresponding 1 enable bit causes the PC architecture IRQ10 interrupt to be asserted.

Programming the VMEbus Interface

5

This chapter describes the EPC-8A 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 or other higher level programming facilities.

The EPC-8A VMEbus interface registers are defined in Appendix F, *Registers*. For specific bit definitions, refer to that appendix.

Concepts

The VMEbus interface on the EPC-8A is primarily for master accesses out onto the VMEbus. The only exceptions to this are the VXI-defined registers accessible in the A16 space discussed in the previous chapter. There is no slave memory accessible in the A32 or A24 address spaces.

VMEbus accesses are performed in either of two ways. For real-mode operating systems, a 64Kbyte “window” is provided in high memory accessed at 0E0000–0EFFFF. Any 64Kbyte region of the VMEbus (all address spaces) can be mapped behind this “E page” window such that the VMEbus address space is visible to the operating system memory map. For protected mode operating systems, the EPC-8A provides direct mapping of the VMEbus into the PC memory space above 256 MByte.

Atomic access

The EPC-8A supports atomic access to the VMEbus data for data lengths not greater than 32 bits and only then if it is aligned to its natural boundary; for instance, 32-bit data must be aligned on a 32-bit address.

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-8A 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 do not behave as expected. Also, many compilers do not actually generate a 32-bit access. Instead, two 16-bit accesses are generated. This can also cause an 32-bit RMW cycle to terminate prematurely with unexpected results.

Setting the VMEbus Access Bit

Before any VMEbus accesses can occur, the VMEbus access bit must be set. The EPC-8A provides two separate VMEbus access bits corresponding to the two access methods described above. Both of these access bits are part of the configuration register at port 8102h. In both cases setting the bit (1) enables accesses and clearing the bit (0) disables accesses. Bit 0 is used to enable direct VMEbus accesses above 256 MByte. Bit 1 enables E-page accesses. In some cases, the programmer may wish to enable both access methods. However, if bit 1 is set (E-page accesses enabled), then the 64KByte of upper memory located from E000:0000 to EFFF:FFFF is not available for use as system memory. If the application program is going to enable the E-page window, care must be taken to ensure that the Operating System does not use this address space. Otherwise a memory conflict occurs that will cause the Operating System to fail at some point.

Real-Mode “E-page” VMEbus Accesses

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

A32

31 24	23 22	21 16	15 0
From port 8150h	From port 8151h	From port 8130h	From 486 address bits 15–0

A24

23 22	21 16	15 0
From port 8151h	From port 8130h	From 486 address bits 15–0

A16

15 0
From 486 address bits 15–0

It should be noted that the EPC-8A drives all 32 address lines even when performing an A24 or A16 access. Although the VME specification states that boards should not decode address lines outside their respective address spaces, some boards do anyway. Because of this, all “unused” high address lines should be set via their respective registers to all 1’s before a VME data transfer is executed.

Supported Address Modifiers

The table below lists supported address modifiers:

Table 5-1. Supported Address Modifiers

Address modifier	Description
29h	A16 non-privileged
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 “E” Page Accesses

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

Example #1

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

1. Set the VME E-page enable bit in Register 8102h.
2. Determine the correct address modifier for A16 supervisory access (2Dh)
3. The unused address lines A31–A16 are driven to unspecified levels when not used.
4. Set the access mode in the VME Modifier Register (8151h) 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, 8151h should be set to `1100 1010` or `CAh`.

1. Map the address.

   Add the A16 address to the "E page" address

   `Addr ← segment:offset + A16 address`

2. Read the data.

   `Data ← value pointed to by Addr`

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(0x8102,(inp(0x8102)|2));/* set VME E-page enable access bit */
outp(0x8151,0xCA);  /* Set address modifier to A16 supervisory
access */
wptr = (WORD far *) (0xE0000000L + addr);
data = *wptr;       /* Read through window */
```

Example #2

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 E-page enable bit in Register 8102h.
2. Set register 8150h 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 8151h 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 8130h 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

C code for Example 2

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

outp(0x8102,(inp(0x8102)|2));  /* set VME E-page enable 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 (8154h) after each access.

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, WDT, and so on.) should be handled on the EPC-8A. Note that, in general, the use of EPConnect is highly recommended for handling interrupts.

- Enable the appropriate registers (VME Interrupt enable (8153h) and VME Event enable (8155h) 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-8A or another master) writes to the Message Low register (16-bit) from the VMEbus. A message interrupt does *not* occur when the EPC-8A writes to its 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 are 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 are recognized.

Protected-Mode Direct VMEbus Accesses

Addresses above 256 MB map directly 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.

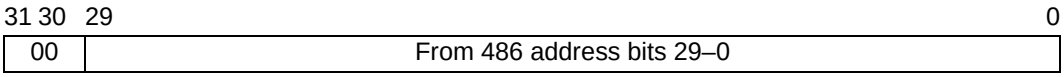
VME access is enabled by two bits in register 8102h. Bit 1, if set, enables the E-page access mechanism, and bit 0, if set, enables the 32-bit access mechanism. If the E-page mechanism is disabled, then the E-page is available for EXMbus options that require more DOS memory below 1MB. The type of VME access performed by the EPC-8A is a function of the address supplied.

Table 5-2. VME access type

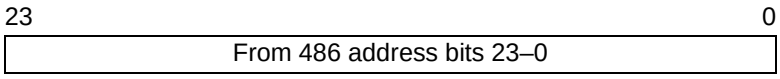
Address Range			Access Mode & Access Modifiers	Byte Order
1xxx0000	–	1xxxFFFF	A16M, AM = 1,0,1,~A25,A24,~A24	Little endian
2x000000	–	3xFFFFFF	A24M, AM = 1,1,1,~A25,A24,~A24	Little endian
40000000	–	7FFFFFFF	A32M-Supervisor (AM = 001101)	Llittle endian
80000000	–	BFFFFFFF	A32M-Supervisor (AM = 001101)	Big endian
Cxxx0000	–	CxxxFFFF	A16M, AM = 1,0,1,~A25,A24,~A24	Big endian
	or			
DxxxFFFF	–	DxxxFFFF		
Ex000000	–	ExFFFFFF	A24M, AM = 1,1,1,~A25,A24,~A24	Big endian

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

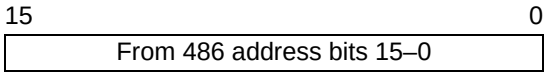
A32



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 must coordinate their use of the E-page mapping registers.

When using the EPC-8A this way to perform VMEbus accesses, you 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 you can use the E-page mechanism if needed to provide these.

Generating IACKs in Protected Mode

As noted in the description of protected mode direct VMEbus accesses, the values of address bits 24 and 25 determine the access mode. Setting bit 26 causes an IACK (if the

high-order address nibble is 1, C, or D, i.e., only in A16 direct access mode). Note that an IACK overrides any of the other access modes.

With the EPC-8A programmers are not restricted to using the E-page to signal an IACK; direct VME memory access is possible. In protected mode you cannot access physical addresses—only virtual addresses. Thus for addressing, map the flat 32-bit physical address to a virtual pointer.

To acknowledge a VME interrupt and acquire its 16-bit status/ID:

1. Multiply the VME interrupt number by two and add the result to 0xC4000000.
2. Map the resulting physical address 0xC4000000 to a protected mode address.
3. Read the 16-bit status/ID value from the resulting virtual mode address.

Whether or not a protected mode environment uses virtual memory is an attribute of the operating system, not of being in protected mode. There are protected mode environments that do not use virtual memory (like most DOS extenders).

Programming the Watchdog Timer

The watchdog timer on the EPC-8A can be set to either halt the system or reboot when a watchdog timer event occurs. The counter causes a watchdog event after a specified time.

Bits 1 and 2 of the Module Status/Control register (815Dh) are cleared by a “warm reset”. This keeps the watchdog timer from expiring on a “warm reset” that is not initiated from a source other than a watchdog timeout. ENSYSO (bit 6 of 815Dh) also has to be cleared by the BIOS must set the BTOE bit (the VME Bus Timeout Enable bit).

The watchdog timer is enabled by setting the WDTR bit (bit 3 of register 815Dh). Note that a watchdog hardware reset results in a “warm” hardware reset. An I/O read to address 815D resets the counter.

To program the watchdog timer, follow these steps:

1. Determine if you want the watchdog timer to reset the EPC-8A or signal a watchdog timer event using IRQ10. Use bit 3 in the Module Status/Control Register (815Dh). If set to 1, the EPC-8A resets. If set to 0, the event is signaled.
2. Determine if you want the system to 1) halt or 2) continue rebooting on the watchdog timer event. From the EPC-8A BIOS Advanced Menu, choose the option you prefer enabled.
3. Set the speed of your watchdog timer. Options are 8.2 seconds, 128 mS, or 1.02 seconds. Use bits 1 and 2 of register 815Dh. Bit 1 is the slow timer and bit 2 is the fast timer. When used in conjunction, the settings are as follows:

   01 8.2 seconds

   10 128 mS

   11 1.02 seconds

The timer is reset to its maximum value by an I/O read of the module status/control register. Application software that utilizes this timer should take care to reset the counter just prior to enabling the interrupt bit in register 8155h. This inhibits a spurious timer event from occurring just after enabling the timer.

Chipset and I/O Map

A

The following defines the I/O addresses decoded by the EPC-8A. It does not define addresses that might be decoded by EXMs and the EXP-MX.

Table A-1. First (8-bit) DMA controller: RadiSys R400EX 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 req counter (R)
		Clear all DMA req mask(W)
00F		All DMA request mask

Table A-2. First Interrupt controller: RadiSys R400EX emulating 8259 of PC/AT

I/O Addr	Functional group	Usage
020	Interrupt controller 1	Port 0
021		Port 1

Table A-3. R400EX controller

I/O Addr	Functional group	Usage
024	R400EX Controller	Index register
026		Data register

Table A-4. Counter-Timer functions: RadiSys R400EX emulating 8254 of PC⁄AT

I/O Addr	Functional group	Usage
040	Timer	Counter 0
041		Counter 1
042		Counter 2
043		Control (W)

Table A-5. Keyboard Port: RadiSys R400EX emulating 8742 of PC/AT

I/O Addr	Functional group	Usage
060	Keyboard controller	Data I/O register
061	Port B	R400EX Miscellaneous control bits
064	Keyboard controller	Command/status register

Table A-6. Time-of-Day Clock: RadiSys R400EX emulating MC146818 of PC⁄AT

I/O Addr	Functional group	Usage	
070	Real-time clock	RTC index reg / R400EX 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

Table A-7. DMA Page Registers: RadiSys R400EX emulating 74LS612 of PC/AT

I/O Addr	Functional group	Usage
080		Phoenix BIOS status information
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

Table A-7. DMA Page Registers: RadiSys R400EX emulating 74LS612 of PC/AT

I/O Addr	Functional group	Usage
08B		Channel 5 page register
08F		Refresh page register

Table A-8. Port A

I/O Addr	Functional group	Usage
092	Port A	Fast A20 and reset control

Table A-9. EXM Configuration

I/O Addr	Functional group	Usage
096	EXM Config	EXM slot register (part of the subplane)

Table A-10. Second Interrupt Controller: RadiSys R400EX emulating 8259 of PC⁄AT

I/O Addr	Functional group	Usage
0A0	Interrupt controller 2	Port 0
0A1		Port 1

Table A-11. Second (16-bit) DMA Controller: RadiSys R400EX 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
0EC		R400EX index register
0ED		R400EX data register

Table A-12. Coprocessor Interface: For the EPC-8A DX, DX replaces the 80287 of PC/AT

I/O Addr	Functional group	Usage
0F0	Coprocessor	Clear coprocessor busy
0F1		Reset coprocessor

Table A-13. EXM Configuration

I/O Addr	Functional group	Usage
100	EXM Configuration	EXM IDs (on each EXM module)
102		EXM option byte 1 (slot-specific)
103		EXM option byte 2 (slot-specific)

Table A-14. Ethernet

I/O Addr	Functional group	Usage
240–25F	Default Ethernet registers (programmable)	

Table A-15. Serial I/O (Com2) Port: National Semiconductor PC87336 emulating 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

Table A-16. Parallel I/O (LPT1) Port: National Semiconductor PC87336 emulating 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

Table A-17. VGA

I/O Addr	Functional group	Usage
3B4	VGA	CRT Controller index
3B5		CRT Controller data
3BA		Feature control output, Input status
3C0		Attribute controller Index/Data
3C1		Attribute controller Index/Data

Table A-17. VGA

I/O Addr	Functional group	Usage
3C2		Miscellaneous output, Input status
3C3		Sleep
3C4		Sequencer Index
3C5		Sequencer Data
3C6		Video DAC pixel mask, Hidden DAC register
3C7		Pixel address read mode, DAC state
3C8		Pixel mask write mode
3C9		Pixel data
3CA		Feature control readback
3CC		Miscellaneous output readback
3CE		Graphics controller index
3CF		Graphics controller data
3D4		CRT controller index
3D5		CRT controller data
3DA		Feature control, input status
3F0		Configuration PC87366 Super I/O Combo chip
3F1		Configuration PC87366 Super I/O Combo chip
46E8	VGA	Adapter Sleep

Table A-18. Serial I/O (Com1) Port: National Semiconductor PC87336 emulating 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

Table A-19. EPC Registers

I/O Addr	Functional group	Usage
8102	EPC-8A registers	Peripheral Enables

Table A-20. VME Registers

I/O Addr	Functional group	Usage
8130	VME and misc. control	VME map WA21–16
8132	VME and misc. control	Alias VME map WA21–16
8133	VME and misc. control	Alias VME map WA21–16
8144	VME and misc. control	Alias VME map WA21–16

Table A-21. VXI, VME/Misc., and On-board EXM-2A Registers

I/O Addr	Functional group	Usage
8140	VXI Registers	ID low
8141		ID high
8142		Device type low
8143		Device type high
8144		Status/Control low
8145		Status/Control high
8148		Protocol low
8149		Protocol high
814A		Response low
814B		Response high
814C		Message low low
814D		Message low high
814E		Message high low
814F		Message high high
8150	VME and misc. control	VME map WA31–24
8151		VME modifier
8152		VME interrupt state
8153		VME interrupt enable
8154		VME event state
8155		VME event enable
8158		VME interrupt generator
815C		Unique Logical Address Register
815D		Module status control
815E		Signal FIFO lower
815F		Signal FIFO upper
8380	On-board EXM-2A	Flash low-order address (if present)
8381		Flash low-middle address (if present)
8382		Flash high-middle address (if present)
8383		Flash data address
8384		Reserved
8385		Reserved
8386		Flash high-order address (reserved)
8387		Battery status

Interrupts and DMA Channels

B

Interrupts

The assignment of interrupts for the EPC-8A is shown in the following table:

Table B-1. Interrupts

Interrupt	Description
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	Real-time clock
IRQ9	Unassigned
IRQ10	VME interrupt/event
IRQ11	Unassigned
IRQ12	Unavailable (you must use external keyboard option to free IRQ12.)
IRQ13	Coprocessor
IRQ14	IDE disk controller
IRQ15	Unassigned

DMA Channels

The assignment of DMA channels for the EPC-8A is shown in the following table.

Table B-2. DMA Channels

Channel	Description
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)

C

Connectors

This Appendix specifies the details of the connectors on the EPC-8A. 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-MX are described in the separate manuals for those products.

Pins are labeled from the point of view of looking into the front of the connector on the EPC-8A.

RS-232 Port (COM1)

The RS-232 serial port is a male DB-9 DTE.

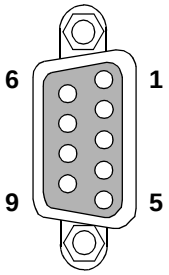


Table C-1. DB-9 Pin-Out

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		

RS-422/485 Port (COM2)

The modified RS-422/485 serial port is a female DB-9 connector.

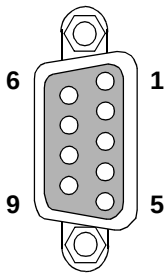


Table C-2. COM2 RS-422/RS485 Port Pin-out

Pin	Signal	Pin	Signal
1	RTS	6	CTS
2	DTR	7	DSR
3	RXD–	8	RXD+
4	TXD–	9	TXD+
5	GND		

Parallel Port

The DB-25 LPT1 parallel port connector is defined as:

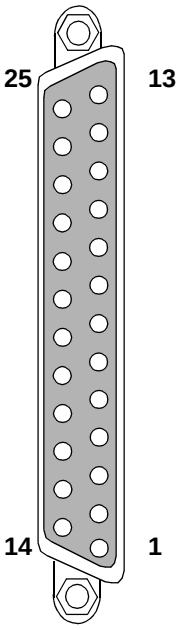


Table C-3. DB-25 Pin-out

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	26	

Keyboard

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

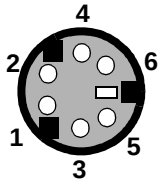


Table C-4. Keyboard Pin-out

Pin	Signal	Pin	Signal
1	Data	4	+5V
2	Not used	5	Clock
3	Ground	6	Not used
1	Reference voltage	2	Speaker tone

VGA

The SVGA DB-15 monitor connector is defined as follows:

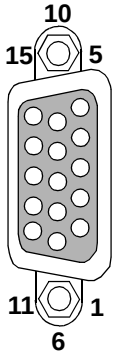


Table C-5. DB-15 Pin-out

Pin	Signal	Pin	Signal
1	Red	9	(key)
2	Green	10	Ground
3	Blue	11	(not used)
4	(not used)	12	(not used)
5	Ground	13	Horizontal sync
6	Ground	14	Vertical sync
7	Ground	15	programmable
8	Ground		output

Ethernet

The DTE RJ-45 phone jack supplies the 10BASE-T interface to the Ethernet controller.

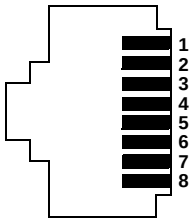


Table C-6. RJ45 Phone Jack Pin-out

Pin	Signal	Pin	Signal
1	Tx+	5	No connect
2	Tx–	6	Rx+
3	Rx–	7	No connect
4	No connect	8	No connect

Memory

D

Do not handle the EPC-8A or memory modules unless you are in a static-free environment.

Memory

The EPC-8A has a single SIMM socket that can handle a SIMM as large as 32 MB. The memory configurations use a SIMMs with the following specifications:

- 72 pin
- Fast page mode or EDO (parit or non-parity).
- 80 ns (or better).
- Single-sided or double-sided

For 8 MB, Use a 2M x 36 SIMM.

RadiSys P/N 70-0041

We recommend Toshiba THM362020ASG-80

For 16 MB, Use a 4M x 36 SIMM.

RadiSys P/N 70-0053

We recommend Toshiba THM364020SG-70

For32 MB, Use a 8 MB x 36 SIMM.
RadiSys P/N 70-0150

We recommend Microhn MT24D836G-6

After upgrading the memory, reboot the system. An error message displays concerning memory. Press F2 to enter the Main Setup Menu. Verify that the memory size on this screen shows the correct amount of memory.

Subplanes

E

Subplane Installation

A subplane is a printed-circuit board 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 backplaneVME to the EPC-8A and expansion modules.

Depending on the particular EPC-8A subsystem configuration, a specific subplane may need installing. Locate the appropriate subsection for the subplane you are using either by name or by picture. Follow the directions in the appropriate subsection.

EXP-BP1 Subplane

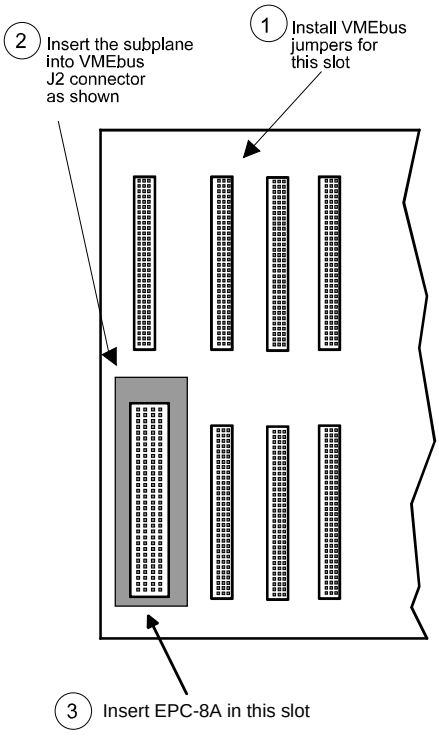


Figure E-1. EXP-BP1 Subplane

This subplane is used in the smallest configuration, where a single-slot EPC-8A processor is used by itself.

The EXP-BP1 is a rectangular-shaped board with two connectors, on each side.

No backplane jumpers are required.

EXP-BP2 Subplane

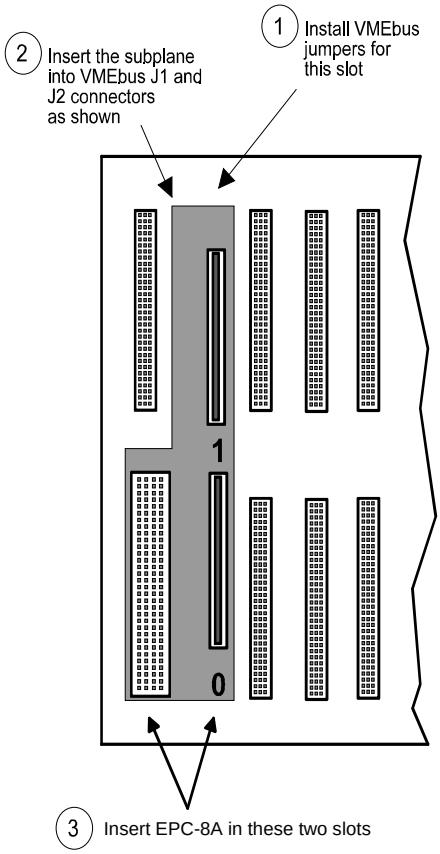


Figure E-2. EXP-BP2 Subplane

This subplane is to provide connectivity for the processor and two EXM modules.

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-8A subsystem will occupy.

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

EXP-BP4 Subplane

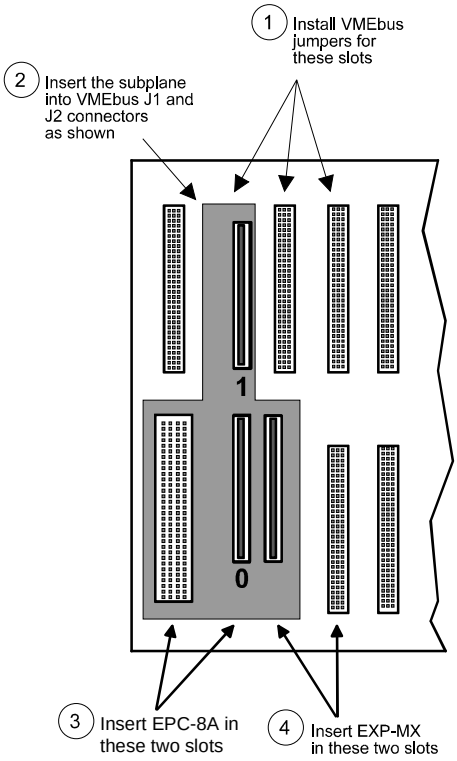


Figure E-3. EXP-BP4 Subplane

The EXP-BP4 subplane is used to couple an EPC-8A processor module, two TXM modules, and an EXP-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-8A subsystem will occupy.

The EXM slot numbers are shown in the drawing.

EXP-BP3A Subplane

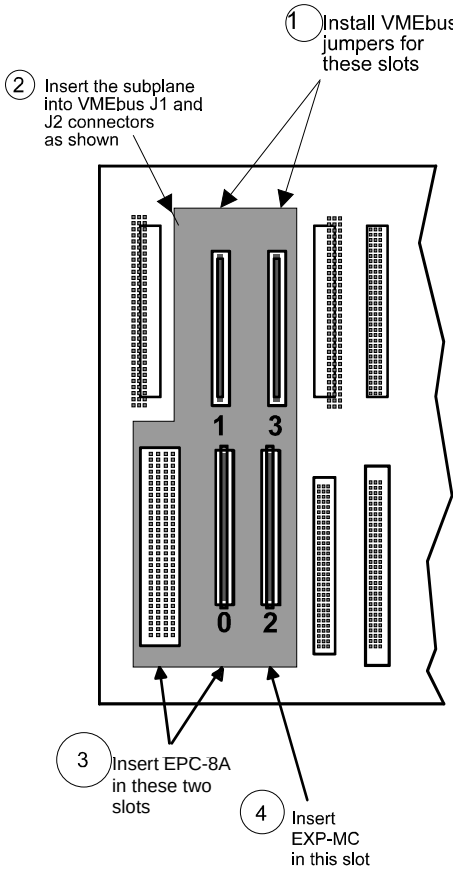


Figure E-4. EXP-BP3A Subplane

The EXP-BP3A subplane allows for a processor and a total of four EXM modules.

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-8A subsystem will occupy.

The EXM slot numbers are shown in the drawing.

EXP-BP5 Subplane

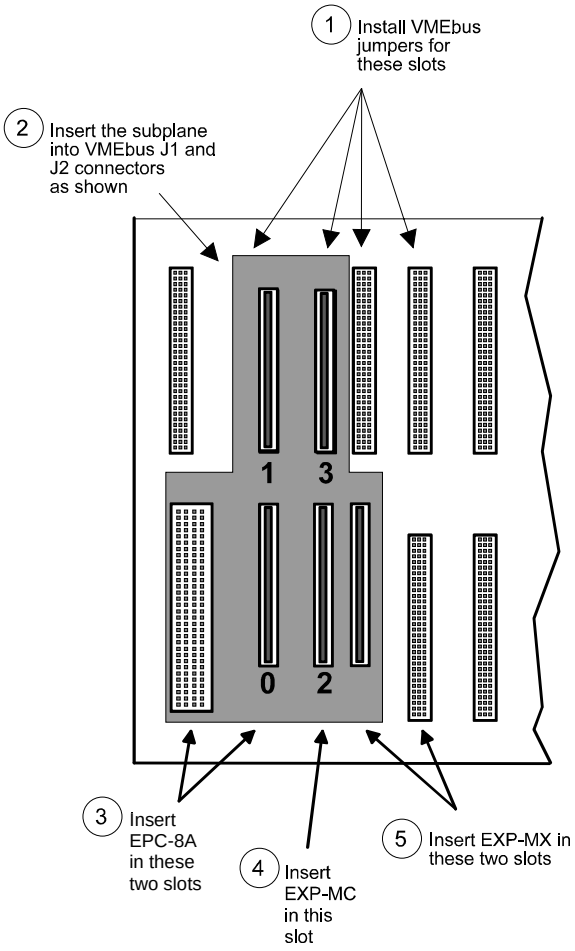


Figure E-5. EXP-BP5 Subplane

The EXP-BP5 subplane is used to build a system with a processor, four EXM modules, and an EXP-MS 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-8A subsystem will occupy.

The EXM slot numbers are shown in the drawing.

EXP-BP4A Subplane

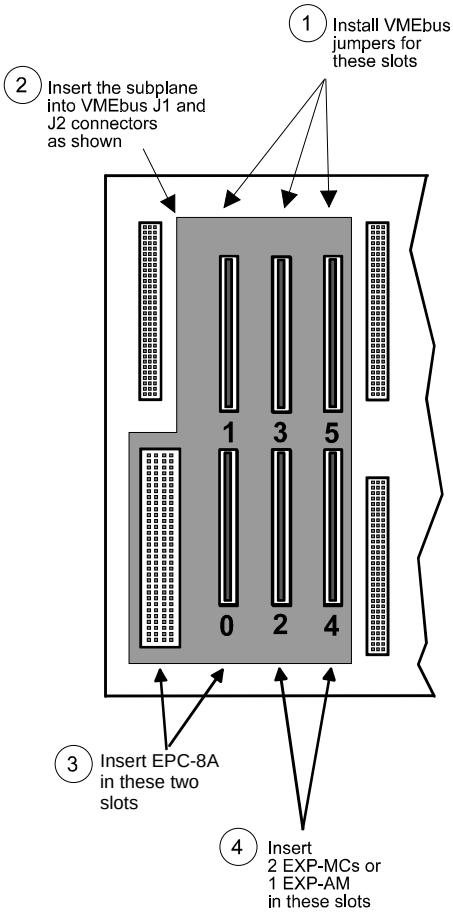


Figure E-6. EXP-BP4A Subplane

The EXP-BP4A subplane is used in systems needing a CPU and six EXM modules.

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-8A subsystem will occupy.

The EXM slot numbers are shown in the drawing.

EXP-BP6 Subplane

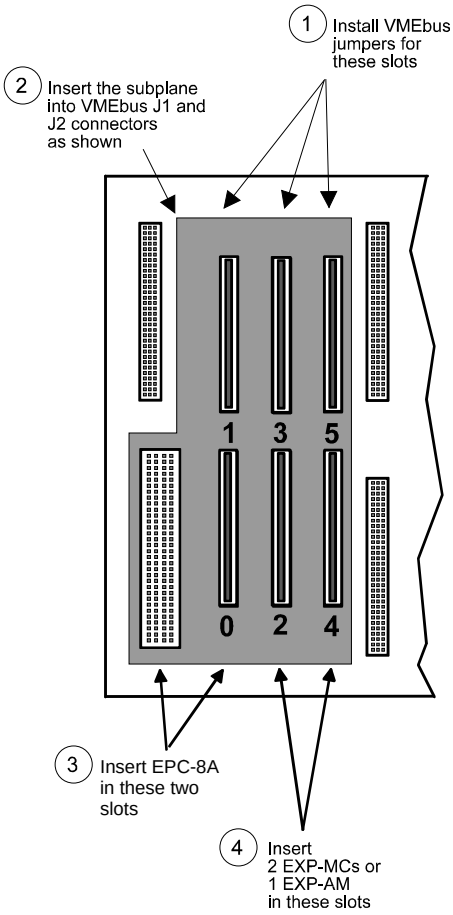


Figure E-7. EXP-BP6 Subplane

The EXP-BP6 subplane is used in a configuration with an EPC-8A processor module, six EXM modules, and an EXP-MX Mass Storage 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-8A subsystem will occupy.

The EXM slot numbers are shown in the drawing.

Registers

F

Registers Specific to the EPC-8A

Registers in the I/O space that are specific to the EPC-8A are defined below. Only registers 8140h–814Fh are “dual-ported” to both the PC and VME bus. The addresses shown below are used by the PC port. The VME addresses for the registers 8140h–814Fh are described later.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	I/O port
Configuration	ARBPRI		RELM	ARBM	GPO	VGA	VME-E	VME-32	8102h
VME A21–16 Address Reg	VMEbus address bits 21–16						res	res	8130h
ID Register, lower	1	1	1	0	1	1	0	0	8140h
ID Register, upper	1	0	1	1	1	1	1	1	8141h
Device Type Reg, lower	1	1	0	0	0	1	0	0	8142h
Device Type Reg, upper	0	0	0	0	0	0	0	S	8143h
Status/Control Reg, lower	SRIE	1	SYSC	1	READY	PASS	NOSF	SRST	8144h
Status/Control Reg, upper	0	MODID	SYSR	R	RESDET	1	1	1	8145h
Reserved, lower	1	1	1	1	1	1	1	1	8146h
Reserved, upper	1	1	1	1	1	1	1	1	8147h
Protocol Register, lower	1	1	1	1	1	1	1	1	8148h
Protocol Register, upper	0	0	0	1	1	1	1	1	8149h
Response Register, lower	R	RRIEN	1	SIG	MLCK	WRCP	FSIG	LSIG	814Ah
Response Register, upper	0	1	DOR	DIR	ERR	RRDY	WRDY	1	814Bh
Reserved, lower	1	1	1	1	1	1	1	1	814Ch
Reserved, upper	1	1	1	1	1	1	1	1	814Dh
Message Low Reg, lower	RAM								814Eh
Message Low Reg, upper	RAM								814Fh
Message A31–24 Address Reg	VMEbus address bits A31–24 Address register (WA31-24)								8150h
VME Modifier Register	VME WA23–22		BORD	IACK	AM5	AM4	AM2	AM1	8151h
VME Interrupt State Reg	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	MSGR	8152h
VME Interrupt Enable Reg	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	MSGR	8153h
VME Event State Register	1	1	VMER	SIGR	WDT	ACFA	BERR	SYSF	8154h
VME Event Enable Register	1	1	VMER	SIGR	WDT	ACFA	BERR	SYSF	8155h
Interrupt Generator Register	SLOT1*	1	1	1	1	INTERRUPT-OUT			8158h
Unique Logical Address Reg	ULA								815Ch
Module Status/Control Reg	DONE	ENSYS0	1	BTOE	WDTR	FWDT	SWDT	1	815Dh
Signal FIFO Register, lower	RAM								815Eh
Signal FIFO Register, upper	RAM								815Fh

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.

Configuration (8102h)

ARBPRI	RELM	ARBM	GPO	VGA	VME-E	VME-32

This register controls VGA controller enables and the VMEbus interface. Only the lower four bits in this register are cleared by an “AT reset” (that is, when the RESET button is pushed or WDT reset occurs or SYSRESET occurs). All eight bits are cleared by a power-on reset.

VME-32 VME 32-bit enable. This bit is automatically set by the BusManager software when using EPConnect. Please note the VMER bit (bit 5 of the VME Event State register), if asserted, also disables this function, but does not clear the VME-32 bit.

1 Enables VME access through the 32-bit addressing mechanism.

0 Disabled.

VME-E VME E-page enable. This bit is automatically set by the BusManager software when using EPConnect. Please note the VMER bit (bit 5 of the VME Event State register), if asserted, also disables this function, but does not clear the VME-32 bit.

1 Enables VME access through the DOS “E page”.

0 The “E page” is available for DOS use.

VGA VGA enable. If set (1), this bit enables the VGA controller. This bit powers up clear and if the BIOS detects another VGA controller in the system it is not set. Once this bit is set via write of “1” to this register only a hardware reset can clear it.

GPO VGA General Purpose output control. This inversion of this bit is tied to pin 15 of the VGA connector through a 150 ohm resistor.

ARBPRI Arbitration priority. This defines the level at which the EPC-8A arbitrates for the VMEbus:

This value...	Means...
11	3
10	2
01	1
00	0

RELM Bus release mode:

1 ROR (Release on Request), if set.

0 RONR (Release on No Request) The “fair requester” mode.

ARBM Arbitration mode. This bit is pertinent only if the EPC-8A is jumpered to be the slot 1 controller.

1 Priority

0 Round Robin

VME A21–16 Address Register (8130h)

VMEbus Address bits 21–16	1	1

When the EPC-8A performs an access 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 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.

Register bit	VME address bit
7	21
6	20
2	16

The two low-order bits are ROM bits. For compatibility with older EPCs, this register is aliased at I/O port addresses 8132, 8134, and 8136.

ID Register (8140h and 8141h)

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

This read-only register adheres to the VXIbus specification. It defines the EPC-8A as a message-based device and the manufacturer as RadiSys Corporation.

Device Type Register (8142h and 8143h)

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

This register adheres to the VXIbus specification. The value defines the EPC-8A as having a model code of 0C4h if it is a slot 0 controller and 1C4h if it is not a slot 0 controller). Note that the S bit is a read/write bit that must be set by BIOS very early in the boot sequence. This bit may also be read/written from the VME port.

Status/Control Register (8144h and 8145h)

Lower	SRIE	1	SYSC	1	READY	PASS	NOSF	SRST
Upper	0	MODID	SYSR	R	RESDET	1	1	1

This register adheres to the VXIbus specification and also contains EPC-8A specific bits.

SRIE SYSRESET input enable. If set, assertion of VME SYSRESET generates a reset of the EPC-8A. One use of this bit is having EPC-8A software reset other VME devices (via bit SYSR) without resetting the EPC-8A. This bit may be read/written from the PC port, but is read-only from the VME port.

SYSCLK SYSCLK status bit. Only PC port writes to this register can clear this bit. The

bit is then set if four rising edges of the SYSCLK signal are detected. This bit is intended to be used to detect that SYSCLK is being generated on the backplane.

READY — This is a RAM bit defined by the VXI specification. In a VXIbus software environment, if READY=1 and PASS=1, the EPC-8A is ready to accept VXI-defined messages. This bit is read-only from the VME port and may be read/written from the PC port. This bit is also held clear while the SRST bit is asserted. When deasserting SRST via an I/O write to this register, a second write is required to reassert the READY bit, since the READY bit is held in reset until just after the first write completes.

PASS — This bit is read-only from the VME port and may be read/written from the PC port. This bit is also held clear while the SRST bit is asserted. When deasserting SRST via an I/O write to this register, a second write is required to reassert the PASS bit since the PASS bit is held in reset until just after the first write completes.

- 1 — The EPC-8A completed its self test successfully.
- 0 — The Test LED on the EPC-8A front panel is lit.

NOSF — SYSFAIL inhibit. If set, the EPC-8A does not assert the VMEbus SYSFAIL line due to the PASS bit being cleared. If the PASS bit is clear and this bit is clear, then SYSFAIL is asserted. SYSFAIL is also asserted when a Watchdog timeout reset occurs, independent of the setting of this bit. This bit may be read/written from both the VME and PC ports.

SRST — Soft Reset. Setting this bit places the EPC-8A into the soft reset state. This bit may be read/written from both the VME and PC ports.

MODID — This read-only 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.) Note, this bit is defined but not implemented in the EPC-8A and always returns a value of 1. If future versions of the product need this capability it can be provided by installing a resistor.

SYSR — SYSRESET. The EPC-8A 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. This bit may be read/written from the PC port, but is read-only from the VME port.

R — A read/write bit that is available for software use (For example, SURM).

RESDET — This bit is cleared by a hardware reset. Once this bit is written to "1" from the PC-port (read-only from the VME port) it can only be set to zero by a hardware reset. This bit is used by the firmware to determine if a software or hardware reset is in progress and it set to "1" before any OS or application is invoked.

Reserved (8146h and 8147h)

Lower	1	1	1	1	1	1	1	1
Upper	1	1	1	1	1	1	1	1

These registers are reserved and return all ones if read.

Protocol Register/Signal FIFO (8148h and 8149h)

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

A read of this register from either the PC or VME ports reads the ROM constants stored in the protocol register. A write from either the PC or VME port writes the signal register.

The protocol register (the read value) defines the EPC-8A as being a servant and commander, having a signal register, being a bus master and an interrupter, not providing the shared-memory protocol, and not providing fast handshake mode.

When written from the VXIbus, this register is the signal register. The value written enters the signal FIFO (two deep) or returns a bus error (BERR) if the FIFO is already full.

A write to the signal register is a happening of some significance for the EPC-8A, since it potentially asserts an EPC interrupt, shuffles a signal-register FIFO, and may return BERR if the FIFO is already full. For these reasons, the full semantics of writing to the signal register are discussed separately in a later section.

Response Register (814Ah and 814Bh)

Lower	R	RRIEN	1	SIG	MCLK	WRCP	FSIG	LSIG
Upper	0	1	DOR	DIR	ERR	RRDY	WRDY	1

This register contains some VXI-defined state bits associated with message handling, and several EPC-8A dependent bits. All of these bits may read/written (except where noted below) from both the PC and VME ports. Some of this bits may also be cleared by certain hardware events as described below.

DOR RAM bit available to software for VXI communication protocols.

DIR RAM bit available to software for VXI communication protocols.

ERR RAM bit available to software for VXI communication protocols.

RRDY Read ready. 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.

R RAM bit available to software.

RRIEN This EPC-8A specific bit is used to enable RRDY interrupt signaling.

- 0 Hardware reset state. only the deassertion of WRDY causes the MSGR interrupt to be asserted.
- 1 The “OR” of the deasserted RRDY, WRDY bits is used to assert the interrupt.

This bit would normally only be set for protocols that require multiple reply data to be sent in response to a single command.

SIG If this EPC-8A specific bit is 0, the signal register FIFO is empty. This bit is

read-only.

MLCK — This EPC-8A specific bit is used for synchronization of messages from multiple senders, something not provided for in the VXI specification.

- 1 — The message register can be locked for the sending of a message.
- 0 — The message register is locked.

WRCP — This EPC-8A specific bit is a read-only copy of the WRDY bit.

FSIG — Defined only when SIG=1, in which case FSIG is the number (0 or 1) of the register in the FIFO holding the earliest signal. This is a read-only bit.

LSIG — Defined only when SIG=1, in which case LSIG is the number (0 or 1) of the register in the FIFO holding the most recent signal. This is a read-only bit.

FSIG and LSIG have no utility to software. They exist as read-only bits for tests of the EPC-8A during manufacture.

The protocol for sending a message to the EPC, if there are multiple potential senders, is the following. The sender first reads register ABR (described in greater detail later). If both WRDY and MLCK are 1, he may then proceed to send the message; if not, he must spin or wait for this condition. For 16-bit messages, he writes into the Message Low register.

The bits RRDY, WRDY, and MLCK in the response register are altered by hardware-detected conditions. A read from the message-low clears RRDY. A write into all or the lower 8 bits of the message low register clears WRDY. A read from the VME bus port of the Alternate Bus Response register clears MLCK if WRDY is set. A read from the Alternate Bus Response register also returns the value in the Response register. Please note that the Alternate Bus Response register is accessed at offset 0x2A from the VME A16 base address of these registers.

Bits MLCK, DIR, DOR, ERR can be set or cleared by using a write to the response register from either the PC or VME ports. RRDY and WRDY may only set via a write to the response register. For these two bits, a 0 written into the respective bit position does not change the value of the register bit. A

1 written into the respective bit position sets the value of the register bit to 1.

Supporting software on the EPC must be aware of how to set the bits initially. Valid states include:

RRDY	WRDY	MLCK	State
X	1	1	Write ready (awaiting incoming msg
X	1	0	Write ready, locked by a sender
1	X	X	Read ready (outgoing data present)
0	0	X	Not ready for write or read

Reserved (814C and 814D)

Lower	1	1	1	1	1	1	1	1
Upper	1	1	1	1	1	1	1	1

These registers are reserved and return 1s when read. Writes to these registers have no effect.

Message Low Register (814Eh and 814Fh)

Lower	RAM
Upper	RAM

There are actually two 16-bit registers at this address, outgoing and incoming 16 bit registers (UART model).

- **Outgoing register**: A write from the PC side fills the outgoing register. This register can only be read from the VME port and when this occurs the RRDY bit is deasserted in the Response register.
- **Incoming register**: A read from the PC side reads the incoming register. The incoming register can only be written to from the VME bus port and when this occurs, the WRDY bit is deasserted in the Response register.

VME A31–24 Address Register (8150h)

VMEbus A31–24 Address register (WA31–24)

This register is one of several that supply the VMEbus address bits when the EPC-8A makes an access in its “E page.” This register supplies VME address bits A31–A24.

VME Modifier Register (8151h)

VME WA23–22	BORD	IACK	AM5	AM4	AM2	AM1

This register is also used when the EPC-8A 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-8A 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 (8152h)

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 (8153h)

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-8A IRQ10 interrupt is asserted. Software may then examine the interrupt and event state registers to determine the cause.

VME Event State Register (8154h)

1	1	VMER	SIGR	WDT	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. All bits are read-only except for the VMER and BERR signals.

ACFA VMEbus ACFAIL is asserted.

BERR This bit is cleared (asserted low) when an access from the EPC-8A to the VMEbus is terminated with a BERR (bus error). It is also held clear when the SRST bit is set. This bit may be deasserted by writing a “1” (provided SRST is not asserted) into this bit position.

SYSF VMEbus SYSFAIL is asserted.

WDT Watchdog timer expired

VMER A SYSRESET or soft reset has occurred. This bit is held clear while SYSRESET is asserted or the SRST bit is asserted. This bit may be deasserted by writing a “1” into it once the reset conditions are removed.

SIGR Signal register FIFO is not empty.

All bits are read-only except BERR and VMER. BERR is a sticky bit that is cleared whenever an access from the EPC-8A is terminated by a bus error or is held clear, and remains clear (0) unless changed by software (by writing any value to this register).

VME Event Enable Register (8155h)

1	1	VMER	SIGR	WDT	ACFA	BERR	SYSF

The low-order six bits are 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-8A IRQ10 interrupt is asserted. Software may then examine the interrupt and event state registers to determine the cause.

VME Interrupt Generator Register (8158h)

SLOT1*	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-8A. 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 is sent to the EPC-8A, the INTERRUPT-OUT bits are cleared. You can also deassert a previously asserted interrupt by writing 0 into the register. Finally, this register is cleared whenever SYSRESET* is asserted or when the SRST (soft reset) bit is asserted.

Bit 7 of register 8158 is read-only and returns the value of the slot1 jumper setting. If the slot 1 shunt is installed (slot1 operation), then 0 is returned. If the slot1 shunt is not installed, then 1 is returned.

Unique Logical Address Register (815Ch)

ULA

This register contains the EPC-8A's ULA. The ULA contents are used to map the EPC-8A's register set into VME A16 space as described below in the VMEbus Mapped Registers section. The ULA is changed by writing into this register or into the ID register.

Module Status/Control Register (815Dh)

DONE	0	1	BTOE	WDTR	FWDT	SWDT	1

This register contains the following miscellaneous status and control bits: Only bit 3, WDTR, is cleared by a warm reset. All bits of this register, except for the read-only DONE status bit, are cleared by a power-on reset.

DONE — This read-only bit is 0 whenever the EPC-8A has a VMEbus access outstanding. It is used for determining when a pipelined VMEbus write is complete.

BTOE — Bus timeout enable. Enables the slot-0 bus timeout timer. This is used by the BIOS.

WDTR — Watchdog timer reset enable. If 1, expiration of the watchdog timer generates a reset of the EPC-8A. If 0, only the WDT event is signaled. A read of the module status register should be performed before enabling the watchdog timer reset. This clears the watchdog counter to zero so that a PC reset does not occur immediately after enabling the watchdog timer reset.

FWDT — Fast watchdog timer.

SWDT — Slow watchdog timer. FWDT and SWDT produce the following timeout values:

00	Disables events from the watchdog timer
01	8.2 S
10	128 ms
11	1.02 S

A read of the module status/control register also has a side effect of resetting the watchdog timer. Therefore, if you are using the watchdog timer, the intention is that you are required to read this register within the defined period of the timer to prevent its generating an interrupt.

VMEbus Mapped Registers

The EPC-8A maps a standard set of VXI configuration registers onto 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-8A's logical address.

The base address is

`11aa aaaa aa00 0000`

where `aaaaa aaaa` is the value of the ULA field in the response register at I/O port 815C.

The VME-mapped registers are a subset of those defined previously as I/O ports in the EPC-8A. The registers are dual-ported in that they are accessible both from VME and from within the EPC-8A as ports in its I/O space. The VME mapped registers are defined below. Please note that the odd addresses from VME port accesses the lower byte (registers addressed by even PC I/O addresses). The registers may be accessed using D08 and/or D16 accesses from the VME port.

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	Reserved (8147)	Reserved (8146)
8	Protocol/Signal (8149)	Protocol/Signal (8148)
A	Response (814B)	Response (814A)
C	Message high (814D)	Message high (814C)
E	Message low (814F)	Message low (814E)
2A	Alternate Response	Alternate Response

The registers occupy the first 16 bytes of the 64-byte space, but DTACK (BERR in the case of an LWORD or Signal FIFO overflow access) are signaled for accesses within the entire 64 byte region. Note that the registers may only be written by using the lower 16 addresses. Writes between address offsets 16–64 have no effect. For reads, the registers are aliased every 16 bytes (For example, a read at offset 0x10,0x20,0x30 return the data in the ID register). The lone exception to this rule occurs when accessing the Alternate Response register.

Reads and writes of the registers from VME and as I/O ports have identical results and effects except where noted in the register descriptions above.

XFORMAT Software for the EPC-8A

G

XFORMAT Software

You use the XFORMAT utility program to:

- Build DOS file structures on the EPC-8A's optional resident flash memory.
- Build file system images that can be used in VME RAM disks.

For complete information about XFORMAT, see the *XFORMAT Software User's Manual.*

AUTOSET Software

AUTOSET Software

The AUTOSET program is used to configure the Ethernet controller. Since this may be set up by your supplier, you may never be required to use this program. The software is contained on the optional Net 1 software diskette that is an optional item for the EPC-8A. The AUTOSET program must be run from DOS or a DOS shell and enables these actions:

- Select one adapter to be configured. Up to four adapters may be installed.
- Manually select configuration options (such as I/O base address and IRQ) for a specific installation.

If configuring multiple Ethernet controller cards in a single system, refer to the section, “Configuring Additional Ethernet Controllers.”

Installing the Software

Under DOS 5.0 or higher, or in a DOS window, insert the optional Net1 distribution diskette in the floppy drive and switch the command line to that floppy drive.

The DOS INSTALL.BAT file creates a subdirectory on the hard disk and copies the collection of software drivers and configuration files from the distribution disks to the hard disk. Type **Install** and press Enter. Note that AUTOSET can be run from a floppy disk drive.

The **Net1** distribution diskette contains the setup program AUTOSET, and a collection of drivers for various network interfaces and operating systems.

Using the AUTOSET Program

Make sure that network software is *not* currently running on the computer system. If network software is running, reboot or unload the network software.

To start the program, move to the Ethernet controller subdirectory and type **AUTOSET**, then press Enter. The menu depicted in the figure below displays.

Figure H-1. AUTOSET Program Main Menu

The following options are in the main menu: Display/Change Adapter Configuration, Diagnostics, and Quit.

To initialize an Ethernet controller for the first time, to change the configuration, or to configure multiple Ethernet controllers, select “Display/Change Adapter Configuration” in the main menu.

To diagnose problems or verify the configuration, select “Diagnostics.”

If more than one Ethernet controller requires configuration for the first time, see the section *Configuring Additional Ethernet Controllers*.

Configuring an Ethernet controller with AUTOSET.

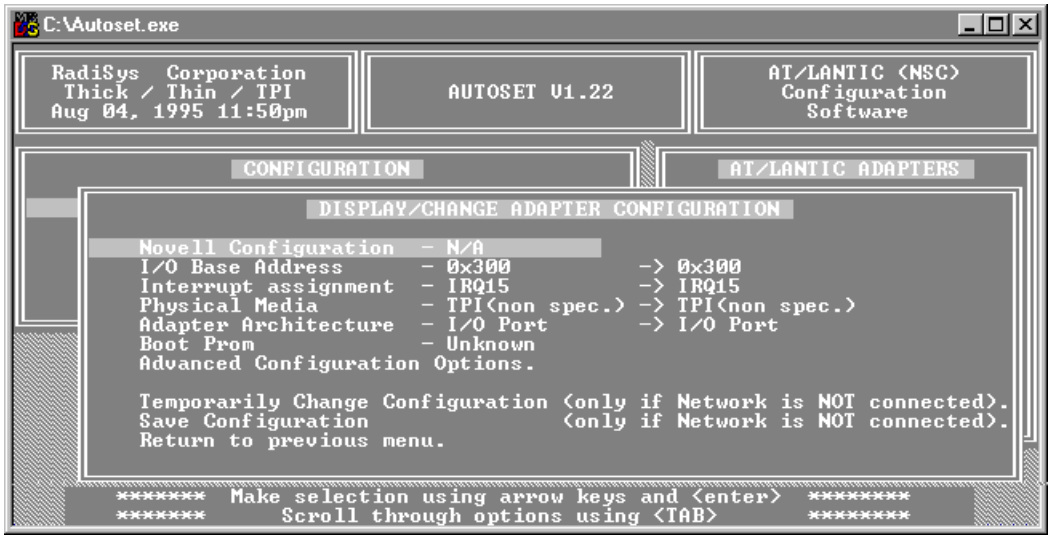


Figure H-2. Manual Configuration Menu

For a single Ethernet controller installation, follow these steps:

1. From the AUTOSET program, select “Display/Change Adapter Settings” to set up the Ethernet controller just installed.
2. Skip the Novell Configuration prompt. It does not apply to the EPC-8A.
3. Select the I/O base address. Use the tab key to change the Ethernet controller's I/O base address to any available location.

   300h is recommended because it allows the use of most network software drivers without modification or configuration.

4. Select the Interrupt Assignment. Be sure to avoid any interrupts used by other interfaces in the system.
5. Select the Physical Media cabled up: only 10BASE-T is available for the EPC-8A.
6. Select the Adapter architecture. Use “I/O Port” for NE2000 mode, or use “Shared Memory” for Western Digital mode.
7. Skip the Boot PROM prompt. It does not apply to the EPC-8A.
8. Save the configuration to the on-board memory only if your system is currently not connected to the network. If you are experiencing difficulties, save as *temporary* changes to determine if they work correctly, then run AUTOSET again and save the changes as *permanent*.
9. Run Diagnostics to verify the setup. If using Western Digital mode, specify a shared memory base address to be used during diagnostics. Also, to guard against conflicts, exit Diagnostics and check that any EMM386 entries in the CONFIG.SYS file and SYSTEM.INI file (if using Windows) both exclude the memory assignment to the Ethernet controller. The default memory exclusion range is DC00–EFFF. If the system does not pass the Diagnostics, try again, changing the indicated parameters.
10. Press ESC to return to the previous menu.

If only one Ethernet controller is to be installed, the configuration procedure is complete. Exit the AUTOSET program and turn to Appendix K, *Configuring the Ethernet Drivers*.

The responses entered from the AUTOSET configuration menu depicted in the previous figure are discussed in detail below.

Novell Configuration

There are several pre-set Novell Configuration options programmed into the AUTOSET program that are not applicable to the EPC-8A. The Novell Configuration option must be set to None.

I/O Base Address

The Ethernet controller uses 32 bytes in I/O space. The I/O Base Address *must* be selected using the AUTOSET program. The AUTOSET program can only access the Ethernet controller when the I/O space occupied by the Ethernet controller does not conflict with I/O space previously assigned. In addition, the network interface drivers used with the

Ethernet controller must use the same I/O base address parameter as the Ethernet controller.

The Ethernet controller is programmed at the factory with a default I/O base address of 300h. If necessary, use the tab key to choose an optional address according to the guidelines set out in the following sections. AUTOSET allows selection of the following I/O base addresses: 240h, 280h, 2C0h, 300h, 320h, and 340h.

If the Ethernet controller is mistakenly programmed to use an I/O address that conflicts with another I/O device, the AUTOSET program does not pass the diagnostics test. In this case, disable the other device or select a different I/O address for the Ethernet controller. For example, an Ethernet controller installed in an EPC system that also contains an EXM-16 SCSI controller must not be set to I/O base address 340h.

Adapter Architecture

The Ethernet controller has up to 16Kbytes of shared memory buffer. The default shared memory base on the Ethernet controller is DC00h.

If using the DOS expanded memory manager or Microsoft Windows, be sure to exclude the area of memory reserved for the Ethernet controller base memory.

For instance, if the Ethernet controller shared memory selected is 16 Kbytes and is located at DC00h, when you finish using AUTOSET, include the following statement in the CONFIG.SYS file:

`device=c:\dos\emm386.exe x=dc00-dfff` ... *other parameters* ...

For Microsoft Windows, include the following statement in the [386Enh] section of the SYSTEM.INI file:

`EMMExclude=dc00-dfff`

Note that the area excluded may be greater than that required by the Ethernet controller due to other system requirements. For example, VME and VXI EPCs also require excluding the E page (E000 to EFFF).

If you are unsure how to make these changes, refer to the *Microsoft DOS User's Guide and Reference* and *Microsoft Windows Operating System* manuals.

Interrupts

Use the tab key to scroll through the displayed options and select one of the listed interrupts. Note that AUTOSET lists all interrupts, not simply those that are available. Use the Diagnostics routine in AUTOSET to detect invalid interrupts and IRQs.

Physical Media

The only physical media option for the Ethernet controller on the EPC-8A is 10BASE-T.

Adapter Architecture

There are two options for the method of data exchange between the CPU and the Ethernet controller. Use the tab key to toggle the displayed option and choose the I/O Port or Shared Memory for data exchange.

NE2000 and NE2000+ users should select the I/O Port mode.

WD8003 and WD8013 users should select the Shared Memory mode.

Systems with limited amounts of memory should select I/O Port mode to avoid conflicts with other memory-mapped devices.

Systems with ample memory, where increased performance is important, should select Shared Memory mode.

Boot PROM

(This option is for systems with an EXM-10A installed. The EPC-8A does not support the boot PROM option of AUTOSET. For more information, refer to the *EXM-10A Hardware Reference Manual*.)

Configuring Additional Ethernet Controllers

To configure multiple Ethernet controllers for use in a single EPC, each Ethernet controller must be inserted in the computer and configured, one at a time, *before* inserting the next Ethernet controller. EXM-10A Ethernet controllers are shipped with the default I/O base address of 240h. A conflict occurs if more than one unmodified Ethernet controller is installed at a time. If I/O base address 240h is not available, disable the conflicting EXM and configure the Ethernet controller. To install more than one Ethernet controller, follow these steps:

1. Turn the EPC off, install one Ethernet controller, and turn the EPC on.
2. Modify the EXM configuration data using the BIOS setup screen.
3. Invoke the AUTOSET program and select “Display/Change Configuration.”
4. Manually select an available I/O base address.
5. Specify the other configuration options for this Ethernet controller.
6. Run Diagnostics to validate the configuration options.
7. Save the configuration, then exit the AUTOSET program.

Repeat this procedure until all Ethernet controllers are installed and configured. Make sure that no parameter conflicts occur among the Ethernet controllers. With the exception of the last Ethernet controller configured, each card must be set to an I/O address other than 240h. That way, as each succeeding Ethernet controller is installed with its factory default of 240h, no conflict occurs with preceding Ethernet controllers. The software driver for each Ethernet controller must be configured to match that Ethernet controller's actual IRQ and I/O base address.

A unique I/O address for each Ethernet controller is required so that AUTOSET can distinguish which card is to configure. Two cards cannot share the same I/O base address; if they do, neither are configurable. If this occurs, remove one card and change the I/O address on the one still in the system, then return the first card.

When using an EPC-8A with an EXM-10A installed, you cannot start AUTOSET while already connected to the network. The system must be logged off.

Diagnostics

Once the LAN adapter is installed and cabled, use the AUTOSET diagnostic program to check the adapter installation.

Note that the AUTOSET diagnostic program test requires that the adapter be attached to a properly-terminated network or to a BNC “T” connector that has two terminators connected (if thin Ethernet is used). Once the cable is installed, use the AUTOSET diagnostic program to check the network interconnection.

Select the Diagnostics option in the AUTOSET menu. The Initialization and Diagnostics menu displays. The configuration options currently chosen are listed in parentheses. The following displays while several options are checked:

Network Interface Controller (12-byte node address).....OK

Buffer Memory Check.....OK

Check Cable Connection (Cable Connected).....OK

Interrupt Assignment (5) .....OK

Boot PROM Check (No Boot PROM) .....OK

Press ESC to return to the previous menu. If Diagnostics reveals an error, return to the Display/Change configuration menu and make the necessary changes. Run Diagnostics again.

Error Messages

The AUTOSET software has several associated error messages that may display during operation. These error messages are explained below.

- `Out of memory error`

  This message displays only during development and should never display during normal operation. If this message displays, contact RadiSys Technical Support.

- `No adapter at this address`

  This message displays when the address selected does not have an attached Ethernet controller network adapter card. Check the hardware to make sure the Ethernet controller is properly seated and functioning. If the message continues to display, select another address such as 300h, 280h, etc. If an address change does not fix the problem, the Ethernet controller should be replaced.

- ```
  Error — multiple boot ROMs detected
  This error has been automatically corrected
  Please re-power the PC to enable the corrections
  ```

  This message displays when more than one installed Ethernet controller adapter has enabled a boot ROM. Only one network boot ROM should ever be installed in a PC.

  To correct this, the adapter at the lowest address should remain enabled, and all other adapters should be set for no boot ROM. After saving the changes in the software, invoke a hardware reset—either push the reset button, or turn the system off, wait ten seconds, then turn it back on.
  ```

- `Boot PROM changes occur only after cold boot of PC`

  This message displays following a saved change to the boot ROM configuration. Boot ROM changes are not in effect until a hardware reset is initiated—either push the reset button, or turn the system off, wait ten seconds, then turn it back on.

- `Error – no new adapter`

  This message displays when the user has attempted to enable a disabled Ethernet controller, but the software never found the disabled adapter. Either the address is wrong or the adapter card is faulty.

- `No interrupt available for configuration`

  This message displays when there is no free interrupt to assign to the Ethernet controller. The only solution is to determine where all the interrupts are configured, and to free an interrupt that is redundant or unnecessary.

- `Cannot initiate an adapter with a disabled interrupt`

  This message displays when a disabled interrupt has been selected. Retry with a valid interrupt selection.

- `No NIC.`
  `DMA Failure`

  These messages display when a fault is detected in the AT/LANTIC† chip on the Ethernet controller. This is a fatal hardware error. Contact RadiSys Technical Support.

- `Incorrect PROM ID Byte`

  This message displays when there is a hardware problem in the EEPROM. I/O mode is thus disabled. Contact RadiSys Technical Support.

- `(xfer to memory)`
  `(xfer from memory)`
  `Failed after X bytes with X`

  These messages display if there is a problem during the Buffer Memory test. The first two messages highlight errors in the transfer of data. The third message identifies where the data transfer error occurred. This is a fatal hardware error. Contact RadiSys Technical Support.

- `Cable Disconnected`
  `Cable Unterminated`

  These messages indicate faults in the cabling or connection. Check to see if the network cable is in proper working order, properly connected and terminated. If the connections are proper and the termination is acceptable, the cables are faulty. If the cable tests good in another system, the connection is faulty.

- `No Interrupt`

  This message displays during the Interrupt Assignment test if the interrupt selected is not active. Return to the setup function and select another interrupt listed as available.

- `Boot ROM Failed`

  This message displays during the Boot ROM Check if the boot ROM does not pass the checksum test. The user must replace the boot ROM.

- `Using AUTOSET on Large Systems`

  If you are using AUTOSET on a system with more than eight (8) EXM slots, you must use a switch when starting AUTOSET to tell it how many slots are present. For example, on a system with 20 EXM slots, enter the following:

  ```
  AUTOSET /20
  ```

SVGA

I

Video Controller Hardware

The EPC-8A contains an SVGA graphics controller using the Chips and Technologies 65545. This is connected to the CPU local bus to give the best possible graphics performance. SVGA memory is 512 Kbytes, resulting in the following resolutions:

- 640 x 480 16 colors
- 640 x 480 256 colors
- 800 x 600 16 colors
- 800 x 600 256 colors
- 1024 x 768 16 colors

The BIOS does not enable (using bit 2 of register 8102) the SVGA controller if another VGA/SVGA controller is enabled on the EXM expansion interface.

Display Drivers and Utilities

The SVGA Driver and Utilities Software is supplied by Chips and Technologies, and has the following nomenclature (where X is the version number):

```
Windows 3.x:
Product: DR655XX
Software Revision: 3.3.3
Document Release Date: April 7, 1997

Windows 95:
Product: DR655XX
Software Revision: 2.0.3
Document Release Date: June 3, 1997
```

Drivers for other operating systems are available on the Chips and Technologies website at **www.chips.com**.

Introduction

This section provides operating instructions for user utilities and installation instructions for the display drivers supplied with your Chips and Technologies SVGA for the EPC-8A.

Before you begin

The following instructions assume that the user is familiar with DOS and certain DOS commands. Please review the associated DOS commands before performing the installation.

Notational conventions

Throughout this manual, the term 'DOS' refers to both MS-DOS and PC-DOS, except when noting features that are unique to one or the other.

Table I-1 shows the typographic conventions that are used throughout this section:

Table I-1. Notational Conventions

Type style	Description
User Input	This text must be typed exactly as it appears. Text within brackets indicates certain keyboard keys (such as [Enter], [F10], etc.).
Bold	System output. Any message that displays by the computer.
ALL CAPITALS	Directory names, files and acronyms.

Easy installation

The installation utility is provided to facilitate the smooth installation of the display drivers and utility software. The installation program is menu-driven and allows you to select and install only those display drivers for software and applications currently in use.

It is important to note that some display drivers need the associated vendor's application program already installed on the system prior to loading the Chips and Technologies SVGA display drivers. In other cases, the loading of the display driver may be an integral part of the vendor's product installation process. Please review the driver product section below for specific instructions prior to running the installation program.

Windows 95 installation

Follow standard procedures for installing new drivers under Windows 95. For detailed instructions, please see the Display drivers section of this manual.

Windows 3.x installation

The installation utility is located on the diskette labeled "VGA Disk 2 of 2". To install the display dirvers and utilities, insert the diskette into the A: drive, and type the following:

```
A:
SETUP
```

Follow the instructions on the screen to install the display drivers.

Selected drivers are simply copied to the specified disk and directory. Applications will require additional installation as described in the Display drivers section of this manual.

Chips and Technologies web site

If there are any newer versions of the software provided with your SVGA adapter, they are available on the Chips and Technologies web site at www.chips.com.

Utility software

The Chips and Technologies SVGA software for Windows 95 provides several functions for setting screen resolution and color depths. These are selected under the Display tab as normally accessed through Windows 95.

Microsoft Windows 3.1

Before upgrading from a previous release

Before installing the new drivers you should use Windows Setup to select the VGA or SUPERVGA video driver so that when you install the new drivers, there is no chance of overwriting the driver that Windows is using to control your screen. Next go to the system directory, and find a file that is named OEM?.INF where the question mark is a number. There may be more than one of these. These files are the different OEMSETUP.INF files used to configure Windows for different devices. Using a text editor, such as Notepad or Edit, look at them until you find the one that is for the previous version of the Cirrus Logic video drivers and delete it. This is not completely necessary, but if you don't delete old files the drop down box for Setup soon becomes very cluttered with different versions of the same files. Also, in many cases the old files were overwritten by newer ones, so they no longer exist anyway.

Installing Windows 3.1 display drivers

To install the Windows 3.1 drivers from the DOS prompt:

1. Ensure that Windows 3.1 is already installed on your computer.
2. From your Windows directory, at the DOS prompt, type SETUP to run the Windows SETUP.EXE program. Follow the instructions on the screen.

   When you come to the screen which lists the hardware and software components such as display adapter (For example, VGA, CGA, etc.), keyboard type, mouse type, etc., go to the Display selection by using cursor keys to move the highlighted bar and press [Enter].
3. From the next menu listing of display options, scroll through the list until the Chips and Technologies drivers are found (marked with an *):

   * CHIPS 655DGX-VL/ISA 1024x768x16
   * CHIPS 655DGX-VL/ISA 640x480x16
   * CHIPS 655DGX-VL/ISA 640x480x256
   * CHIPS 655DGX-VL/ISA 800x600x16
   * CHIPS 655DGX-VL/ISA 800x600x256
4. Highlight the desired choice by moving the cursor to the correct display driver, and then press [Enter].
5. Continue with the remainder of the setup procedure.

To install Windows 3.1 drivers from within Windows, proceed as follows:

1. Ensure that Windows 3.1 is already installed on your computer and start Windows
2. From the Main window of the Program Manager run the Windows 3.1 Setup program.
3. Highlight the desired choice by moving the cursor to the correct display driver, and then press [Enter].
4. Select Change System Settings... from the Options menu of Setup.
5. Click on the down arrow at the right side of the Display: line. Scroll to the end of the list of available display drivers and select from the following resolutions:

   * CHIPS 655DGX-VL/ISA 1024x768x16
   * CHIPS 655DGX-VL/ISA 640x480x16
   * CHIPS 655DGX-VL/ISA 640x480x256
   * CHIPS 655DGX-VL/ISA 800x600x16
   * CHIPS 655DGX-VL/ISA 800x600x256
6. Highlight by moving the cursor to the desired display driver, and then click OK.
7. Continue with the remiander of the setup procedure. The changes do not take effect until Windows restarts.

Microsoft Windows 95

Driver Installation Procedure

1. Click Start, then Settings, then Control Panel.
2. Start the "Display" applet program.
3. Click the Settings tab, then click the Change Display Type button
4. Click the Change button in the Adapter area.
5. Click the Have Disk button, then click the OK button.
6. Specify the path to the new driver and press the <ENTER> key.
7. Insert the drivers disk labeled “VGA Disk 1 of 2” in the A: floppy drive, and enter A:\. The “Select Device” dialog box appears.
8. Select the adapter that corresponds to the one you installed in your machine and click the OK button

   Windows 95 copies the display drivers to the proper directories on your system.

   Continue choosing Close until asked to restart your machine from the Systems Settings Change dialog box.

   After the system restarts, you can go back to the Display applet and select alternate screen resolutions and color depths.

OSR2 Driver Installation Procedure

1. Open the Control Panel by selecting the Start menu, then Settings, then Control Panel.
2. Double-click the Display icon.

3. Click the Settings tab, then click the Advanced Properties button.
4. Click the Change button in the Adapter area.
5. Click the Have Disk button, then click the OK button.
6. Specify the path to the new driver and press the <ENTER> key.
7. Insert the drivers disk labeled “VGA Disk 1 of 2” in the A: floppy drive, and enter A:\. The “Select Device” dialog box appears.
8. Select the adapter that corresponds to the one you installed in your machine and click the OK button.

   Windows OSR2 copies the display drivers to the proper directories on your system.

   Continue choosing Close until asked to restart your machine from the Systems Settings Change dialog box.

   After the system restarts, you can go back to the Display applet and select alternate screen resolutions and color depths.

Error Messages and Diagnosis

J

Troubleshooting

This section deals with problems that you may encounter that do not provide an error message. If an error message displays, see the *Error Messages and Diagnosis* section of this Appendix, starting on page 99.

Table J-1. Troubleshooting 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.	If not using the built-in VGA, then the EXM-based video adapter may not be 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 (with or without onboard video).	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.
	Subplane failure or other hardware failure (with or without onboard video).	Call RadiSys Technical Support.
	EPC-8A cannot talk to EXM expansion interface.	Call RadiSys Technical Support.
System fails at power-up—does not run power-on self-test. May be accompanied by combinations of beep tones from the speaker.	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-8A is fully seated in the subplane.
	BIOS detected a failure	See *BIOS Beep Codes* table below.
	SIMM missing or loose.	Check that the SIMM is still present or connected.
	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-8A subsystem is using IRQ3 or IRQ4.
	Port hardware failure.	Call RadiSys Technical Support.

Table J-1. Troubleshooting Error Messages

Symptoms	Possible cause(s)	Solution
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 Grantbus jumpers are not installed.
System does not talk across VMEbus.	The VMEbus backplane may not be jumpered correctly.	See the section *Installing the VMEbus Backplane Jumpers* in Chapter 2.
	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.
	There may be no Slot-1 controller providing bus arbitration.	Determine if the system is in the left-most position and that the Slot-1 controller jumper is set.
	EPC-8A or subplane may have bent pins.	Remove the EPC-8A and the subplane and verify that no pins are bent. Then reinsert the subplane and the EPC-8A.
	VMEbus interface failure.	Call RadiSys Technical Support.

BIOS Beep Codes

If the BIOS detects a critical error condition while running the Power On Self Test (POST) code, it may halt after issuing a beep code and attempting to display the error code in the upper left corner of the screen. The audible codes consist of patterns of beeps and pauses. If this occurs, you should contact your supplier for technical support.

Table J-2. BIOS Troubleshooting Beep Codes

Beep Pattern	Condition
1 short	POST OK
2 short	POST error, such as EXM configuration or keyboard error. Refer to the message on the screen to determine the cause.
one long, two short	Insert BIOS update disk now.
one long, three short	Video BIOS has detected an error with the video hardware.
3 short	BIOS update disk missing. Either insert the BIOS update disk or turn the system off and then back on.

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).

`Operating system not found`

Problem: No boot disk could be found.

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

Your hard disk may not be 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. Or the hard disk drive has no active partition set. 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  drive A error or Diskette drive B error`

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): 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-MX module, drive A should be set to "1.44M". 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.

`Failure - Fixed 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, run the BIOS setup program. Change the drive type to match the device being used.

`EXM configuration error (with two short beeps)`

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

Solution(s): Run the BIOS setup program. Enter the EXM menu. Verify the information listed on the screen, save any changes and reboot. If necessary, refer to the Chapter 3, *BIOS Configuration* of this manual and/or your EXM manual(s) for more details.

`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.

```
Incorrect Drive A type - run SETUP
Incorrect Drive B type - Run SETUP
```

Problem: Type of floppy drive (A: or B:) not correctly identified in BIOS Setup.

Solution(s): Run BIOS Setup and validate correct settings. Also, see “Diskette drive A error” above.

`Invalid drive specification`

Problem: You are trying to access a logical drive (For example, 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 (with two short beeps)`

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 (For example,

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 or head/cylinder combination in the EPC-8A's IDE Adapter 0 Master Menu that does not match the type number used to format the hard disk. Run the BIOS setup program. Select the Autotype fixed disk option, or if known, correct drive type or User Editable parameters 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 does not contain the operating system files.

```
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 nonsystem 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.

`Real time clock error`

Problem: The battery-backed TOD clock fails the BIOS test.

Solution(s): Run the BIOS setup program to determine what is wrong, and correct it. If the error occurs repeatedly, the EPC-8A's battery may have failed.

`System CMOS checksum bad - run SETUP`

Problem: Something in the nonvolatile CMOS RAM is incorrect. It has been corrupted or modified incorrectly, possibly by an application program that changes stored data in CMOS.

Solution(s): Run the BIOS setup program to determine what is wrong, and correct it. If the error occurs repeatedly, the EPC-8A's battery has failed.

`System battery is dead - Replace and run SETUP`

Problem: The CMOS clock battery indicator shows the battery is dead.

Solution(s): Run the BIOS setup program to determine what is wrong, and correct it. If the error occurs repeatedly, the EPC-8A's battery has failed and needs to be replaced.

Configuring the Ethernet Drivers

K

After the EPC-8A is installed, cabled and configured, follow the instructions in this appendix to configure the network interface drivers. The distribution diskettes supplied with the EPC-8A contain drivers for IPX, ODI, NDIS and a packet driver under DOS/Windows.

The majority of users running DOS/Windows select the ODI driver in 16-bit mode, if supported by their network software.

The I/O base address, shared memory base address, and IRQ channel used by each LAN adapter must agree with the values for those parameters that are used by the adapter's network interface driver. The method used to change the driver parameters depends on the network operating system software in use.

NetWare IPX Driver for DOS Installation

The Internet Packet Exchange (IPX) drivers provided with the EPC-8A distribution diskettes are NE2000.COM and ATLANTIC.COM. The NE2000.COM driver is used in 8-bit, NE2000 mode with the IPX interface. Once the EPC-8A is configured using AUTOSET, and the distribution diskettes are loaded onto the system, use a text editor to modify the AUTOEXEC.BAT file, entering these commands at the end of the file:

```
ne2000 /o0 (run ne2000 /d to display available options)
netx.com(provided by network software or operating
    system distribution diskettes)
```

The NE2000.COM driver supplied on the EPC-8A distribution diskettes is preconfigured for use with the Ethernet 802.3 interface. If your network requires operating multiple protocols on top of a single card, use the ECONGIF program provided by Novell to change the NE2000.COM interface setting to Ethernet II.

The ATLANTIC.COM driver is used in 16-bit Western Digital mode with the IPX interface. Once the EPC-8A is configured using AUTOSET, and the distribution diskettes are loaded onto the system, modify the AUTOEXEC.BAT file by entering these commands at the end of the file:

```
atlantic /o0 (run atlantic /d to display available options)
netx.com provided by network software or operating
    system distribution diskettes)
```

ODI Driver for DOS Installation

The Open Data-Link Interface (ODI) adds functionality to NetWare and network computing environments by supporting multiple protocols and multiple LAN adapters in a

single workstation. Refer to the Novell *NetWare ODI Shell for DOS* manual for additional installation instructions.

Once the EPC-8A is configured using AUTOSET, and the distribution diskettes are loaded onto the system, use a text editor to modify the NET.CFG file to match the AUTOSET parameters. The interrupt, I/O address, mode, and interface type must all match.

Then add the following programs into the AUTOEXEC.BAT file:

LSL.COM Link support layer

ATLANTIC.COMMultiple Link Interface Driver

IPXODI.COM IPX/SPX Protocol

NETX.COM (provided by operating system or workstation distribution diskettes)

While editing the AUTOEXEC.BAT file, make sure that the path= statement includes the directory containing the NET.CFG file.

The ATLANTIC.COM driver is used for the ODI interface in 16-bit Western Digital mode and 16-bit NE2000 mode.

The ODI interface does not support IRQ15.

NDIS Driver for DOS Installation

The NDIS ethat2 driver is used for 16-bit Western Digital mode and 16-bit NE2000 mode for the NDIS interface. It is used by various protocol stacks and applications, such as Windows for Workgroups (WFW). NDIS 3.0 protected mode support is already included in WFW. The steps below would be part of a typical setup for NDIS 2.0 real mode.

1. Set up PROTOCOL.INI according to the environment. To set up ethat2 for NE2000 using IRQ3 and I/O base address 300, refer to the example below.

```
[EXM10A]
DriverName=ETHAT20$
IOBASE=0x300
INTERRUPT=3

[MS$NETBEUI]
DriverName=netbeui$
SESSIONS=10
NCBS=32
LANABASE=1
BINDINGS=EXM10A
[NETBEUI]
LANABASE=1
BINDINGS=EXM10A
```

2. At the command line, enter NET START WORKSTATION.
3. At the command line, enter NET VIEW to verify access to other computers.

Packet Driver Installation

The diskettes distributed with the EPC-8A contain two packet drivers named ATDRIVE.COM (for 16-bit NE2000 mode) and WD8003E.COM (for 16-bit Western Digital mode). RadiSys also supplies files named TELBIN (a Telnet terminal emulation) and FTPBin (an FTP or File Transfer Program) that work with TCP/IP.

To install the packet drivers, first run AUTOSET to initialize the adapter (IRQ, I/O port, shared memory base, and so on.)

Once AUTOSETAUTOSET has been run, install the packet driver. Add the following lines to the end of the AUTOEXEC.BAT file:

For WD8003:

```
REM WD8003 <s/w interrupt> <IRQ> <IO_Base> <shared memory base>
wd8003e 0x60 5 0x240 0xDC000
ipxpkt.com(supplied with EPC-8A distribution diskettes)
netx.com(supplied with network software or operating system)
```

For NE2000:

```
REM ATDrive -i <software interrupt> -b <I/O Base> -q <IRQ> -n
ATDrive -i 60 -b 240 -q 5 -n
ipxpkt.com(supplied with EPC-8A distribution diskettes)
netx.com(supplied with network software or operating system)
```

Windows NT Driver Installation

This section explains how to use the NE2000 driver provided with Windows NT to access the network using the EPC-8A. Be sure to use a 16-bit data width, and run AUTOSET first (in DOS) to set up the hardware parameters in I/O Port mode.

1. Log on to Windows NT.
2. Select the Network icon in the Control Panel.
3. Select the Adapters button and install the NE2000 adapter.
4. Select and install the NE2000.SYS driver. When completed, restart the system in order for the changes to take effect.
5. Select the File Manager icon. Select “Network Connection” under the Disk Menu. Additional drives that are now available display in the list, verifying the installation.

Index

A

A16 41, 44, 45, 49, 82
A24 44, 45, 49, 75
A32 44, 45, 46, 49, 75
ACFAIL 47, 80
address lines 44
address modifier 45, 49, 79
Altitude 3
Arbitration mode 74
Arbitration priority 74
ATLANIC.COM driver 105
AUTOSET 105, 106, 107

B

backplane 8
backplane jumpers 7
battery 34, 104
BERR 39, 47, 77, 80
BG0 - BG3 7
BG0In - BG3In 8
BG0Out - BG3Out 8
big-endian 40, 41, 49
BIOS 101
  setup 11–27
BIOS Jumper (JP2) 6
BP2 65, 66
BP3A 68
BP4 67
BP4A 70
BP5 69
BP6 71
bus
  arbiter 5, 39
  error 80
  grant 100
  grant signals 7
  release 74
  timeout 39, 81
Bus Timer function 6, 39
byte order 40
byte ordering 40, 79
byte-swapping 40, 79

C

cache 30
chassis 9
Chips and Technologies
  VGA GD5428 93
  web site 94
CMOS
  backup 26
  RAM 11, 34
  restore 26
  setup 102
  setup parameters 34
COM1 27
COM2 24, 27
Commander 77
configuration options 6, 85
configuration registers 41, 82
Configuring a Single EXM-10 86
Configuring Additional EXM-10 89
connectors 59
conventions, notational iv
Ctrl+Alt+Del 37
Current 3

D

D08 40, 41
D16 40, 41
D32 40, 41
daisy chain 6, 39
daisy-chain lines 7
DB-25 60
DB-9 59
Display Drivers and Utilities 93
DMA channels 24, 57

E

E page 39, 40, 49, 75
electrostatic discharge 5
EMM driver 31
endian, little 40, 49
EPC-1 compatibility 75
EPConnect 41, 43, 47

error messages 100
ESD 5
EXM 7, 38, 69
  configuration error 102
  expansion interface 38, 59
  setup menu 39
  slot numbers 69
EXMID signal 38
EXP-AM 7
EXP-BP2 65, 66
EXP-BP3A 68
EXP-BP4 67
EXP-BP4A 70
EXP-BP5 69
EXP-BP6 71
EXP-MC 7
EXP-MS/MX 7, 69
extended memory 31
EZSetup
  I/O base address 87, 88
  Interrupt 88
  Other Parameters 88, 89
  Program 85, 89

F

Fast handshake mode 77
Fixed disk menu 12
Flash Jumper (JP2) 6
floating-point numbers 41
floppy disk drives 12
floppy diskette errors 101
front panel LEDs 35

G

GD5428, VGA 93
general failure 102

H

hardware reset 36, 37
help iv
Humidity 3

I

I/O addresses 24
I/O base address 87, 88
I/O space map 51
IACK 6, 7, 39
IACK daisy chain 6, 39
IackIn 8
IackOut 8
ID register 81
insertion 9
Installation 5
Interrupt 88
  acknowledge 49, 79, 81
  acknowledge cycle 41
  acknowledge signal 7
  assignments 24
  generator register 41
  mapping 57
  reset 80
interrupts 47
IPX 105, 106
IRQ10 interrupt 80, 81

J

J1 connector 9, 39
J2 connector 39
jumpers 6–*??*, 8, *??*–9, 100

K

keyboard 10
keyboard connector 60
keyboard errors 102

L

LEDs 76
little-endian 40, 49
Location monitor 77
Lock 78
LOCK instruction prefix 43
LPT1 27, 60

M

main setup screen 63
Mass Storage module 7, 67, 69, 71
math co-processor 30
memory 31, 63
  map 31
  options 31
  upgrades 31
Memory cache sub-menu 12
message interrupt 80
Message low register 77
Message register 78
  high 77
  low 77
message-based device 75
model code 75
MODID 76
module identification 76
Module status/control register 81
mouse 10
MSGR interrupt 77
Multiple EXM-10s 89

N

NDIS 105, 106
NET.CFG file 106
network interface drivers 105
Non-system disk error 103
notational conventions iv

O

ODI 105, 106

P

P2 39
packet driver 105, 107
parallel port 24, 60
Parity error 103
part numbers 63
PASS bit 37
PC/AT bus 65
peripherals 7, 10
power 3, 65
power-on self-test 36
Printer port 24
priority 5, 39
Priority arbiter 74
processor 30
Protocol register 77
PROTOCOL.INI 106

R

Read ready 77
Real time clock error 104
reboot 36
Recovery Jumper (JP6) 6
Release on request 74
Request on no request 74
reset 34, 36, 37, 50, 74
    interrupt 80
    watchdog timer 81
Response register 77
ROM 31
ROM shadowing 32
RONR 74
ROR 74
round-robin 5, 39
round-robin arbitration 74
Run 35

S

SBER bit 42
selftest 76
serial port 59
Servant 77
setup parameters 34
Setup screen 11–27
shadowing 32
Shock 3
Signal register 77
Signal register FIFO 78, 80
SIMMs 31, 63
slave boards 8
Slot 0
    arbitration 74
    timeout 81
Slot-1 100
    controller 5, 6, 8, 39, 100
    jumper 6
soft reset 37, 76, 81
specifications 63
SRST bit 34, 37
status/ID 41
Sticky BERR 42
subplane 7, 65, 66, 67, 68, 69, 70
support iv
SYSCLK 5, 39, 76
SYSFAIL 35, 76, 80
SYSRESET 38, 75, 76
system control functions 6
system controller 5, 39

T

Telbin 107
Temperature 3
Test 36
Test LED 76
TI83000 29
TOD clock 34, 104
troubleshooting iv, 99

U

ULA 81
unique logical address 41, 82
unique logical address register 81
Using the EZSetup Program 85

V

VGA controller 93
    two controllers in system 93
VGA GD5428 93
Vibration 3
video 10
video BIOS 31
Video controller
    Windows drivers 95
Video options 12
VME

address bits 75
chassis 9
event enable register 81
mapped registers 82
modifier register 40
VME access bit 74
VMEbus 49
accesses 39, 43, 45, 49
addressing 4
arbiter 4
backplane 65
daisy-chain signals 8
direct mapping 39, 43
interface 39, 74
interrupt handler 4
interrupter 4
interrupts 80
master data transfer 4
requester 4
slave data transfer 4
slots 7
specifications 3
system controller 4
timeout duration 39
VXI
device type 4
manufacturer code 4
model code 4
protocols 4
registers 41, 82

W

warm boot 37
warm reset 34, 50
Watchdog timer 81
web sites
Chips and Technologies 94
Western Digital mode 105, 106, 107
Windows 3.1 display drivers 95
Write ready 77