XVME-689VR7
6U VME Intel® Celeron® M Processor Board

USER'S MANUAL

ACROMAG INCORPORATED
30765 South Wixom Road
P.O. BOX 437
Wixom, MI 48393-7037 U.S.A.

Tel: (248) 295-0885
Fax: (248) 624-9234
Email: xembeddedsales@acromag.com

Copyright 2012, Acromag, Inc., Printed in the USA.

Data and specifications are subject to change without notice.

8500-978D

Revision	*Description*	*Date*
A	Init	02/08
B	Errors in J2 and P0 pin-out	09/09
C	Error Correction	02/10
D	Error Correction	12/10

Part Number 74689-VR7

Trademark Information

Brand or product names are trademarks or registered trademarks of their respective owners.
Intel and Pentium are registered trademarks and Celeron is a trademark of Intel Corporation.
Windows, Windows NT, Windows 2000 and Windows XP are registered trademarks of Microsoft Corporation in the US and in other countries.

Copyright Information

This document is copyrighted by Xembedded, Incorporated (Xembedded) and shall not be reproduced or copied without expressed written authorization from Xembedded.
The information contained within this document is subject to change without notice. Xembedded does not guarantee the accuracy of the information.

WARNING

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

Warning for European Users – Electromagnetic Compatibility

European Union Directive 89/336/EEC requires that this apparatus comply with relevant ITE EMC standards. EMC compliance demands that this apparatus is installed within a VME enclosure designed to contain electromagnetic radiation and which will provide protection for the apparatus with regard to electromagnetic immunity. This enclosure must be fully shielded. An example of such an enclosure is a Schroff 7U EMC-RFI VME System chassis, which includes a front cover to complete the enclosure.

The connection of non-shielded equipment interface cables to this equipment will invalidate European Free Trade Area (EFTA) EMC compliance and may result in electromagnetic interference and/or susceptibility levels that are in violation of regulations which apply to the legal operation of this device. It is the responsibility of the system integrator and/or user to apply the following directions, as well as those in the user manual, which relate to installation and configuration:

All interface cables should be shielded, both inside and outside of the VME enclosure. Braid/foil type shields are recommended for serial, parallel, and SCSI interface cables. Where as external mouse cables are not generally shielded, an internal mouse interface cable must either be shielded or looped (1 turn) through a ferrite bead at the enclosure point of exit (bulkhead connector). External cable connectors must be metal with metal back-shells and provide 360-degree protection about the interface wires. The cable shield must be terminated directly to the metal connector shell; shield ground drain wires alone are not adequate. VME panel mount connectors that provide interface to external cables

(e.g.,
RS232, USB, keyboard, mouse, etc.) must have metal housings and provide direct connection to the metal VME chassis. Connector ground drain wires are not adequate.

Environmental Protection Statement

This product has been manufactured to satisfy environmental protection requirements where possible. Many of the components used (structural parts, printed circuit boards, connectors, batteries, etc.) are capable of being recycled.

Final disposition of this product after its service life must be accomplished in accordance with applicable country, state, or local laws or regulations.

Table of Contents

Appendix A SDRAM and Battery Installation 1

Table of Figures and Tables

Chapter 1 – Introduction

The XVME 689-VR7 VMEbus Intel® Celeron® M PC-compatible VMEbus processor module combines the high performance and ruggedized packaging of the VMEbus with the broad application software base of the IBM PC/AT standard. It integrates the latest processor and chipset technology. The XVME 689-VR7is the lowest power draw of any of our processors while maintaining a very high level of processing power.

NOTE:

The VR7 is a replacement board that is designed to fit into the same chassis as a SBS built VR7. Xembedded has no RTM for the rear I/O. If you find you need an RTM please contact the factory.

Module Features

The XVME 689-VR7 offers the following features:

- The XVME 689-VR7 is configured with the Intel Celeron M Celeron M Processor at 1.0GHZ. Both processor models can be configured with 256MB to 2GB SDRAM.
- 1Mb on die level 2 cache on Celeron M 1.0GB (running at the speed of the processor).
- Integrated Video controller using shared system DRAM for VRAM
- Enhanced IDE controller, capable of driving two EIDE devices on P2 (NOT compatible with the XVME-977 or XVME-979).
- Two channels of SATA-150 out P2. Use the XVME 990-VR7 to provide the connectors needed to connect external SATA drives.
- Floppy disk controller, capable of driving one floppy drive on P2 (**NOT Compatible with XVME-977)**
- Dual 10/100/1000 Base T Ethernet controllers with front panel RJ-45 connectors with isolated ground or selectable out the P0 to support rear Ethernet or Vita 31.1.
- Type I/II Compact Flash site on optional carrier
- VME64X VMEbus interface with programmable hardware byte swapping
- Support for Vita 31.1 Switch Fabric in complaint back planes
- Four serial ports:
  - Two RS-232 serial port on front panel (Com 1 and 4) two RS-232 serial port (Com 2 and 3) on P2. NOTE COM 4 is also available out the P2 and can be configured for RS-232/422/485
- Three Universal Serial Bus (USB 2.0) port one on front the other two out P2.
- EPP/ECP configurable parallel port (P2) on 26-pin header on the XVME 990-VR7
- Combined PS/2 compatible keyboard/mouse port on front panel
- PCI 80-pin Expansion Connectors (NOT Compatible with XVME-976-01 thru XMVE-976-205 Carriers) must use the XVME-976-209.
- PMC (PCI Mezzanine Card) site with front panel I/O 32/64-bit 33/66MHz with rear I/O using optional P0 connector. This site is on the internal PCI-X bus.

- Front panel ABORT/RESET switch with indicating lights. Red for “fail” and green for “pass”
- Electrical isolation and noise immunity on the Ethernet ports, Serial Port, and PMC site.
- Ejector type handles in IEEE 1101.10 (Compact PCI type) or IEEE 1101.1 (legacy VME type).
- VME64 VMEbus interface with programmable hardware byte swapping

Architecture

CPU Chip

The Intel Celeron M processors have a new micro-architecture, but remain software compatible with previous members of the Intel microprocessor family. The Celeron M has longer pipeline stages and thus does more per clock cycle, which allows it to run at a lower clock frequency thus saving power. The Celeron M has a large L2 cache (1MB on 160nm) which boosts performance. A Celeron M is comparable in performance to a Pentium 4 running at 50% higher clock rate, but dissipates less than half the power. With a junction temperature range of 0 to 100C, and a max power dissipation ranging from 10W to 24Watts, the Celeron M is capable of withstanding a great deal of thermal stress while reducing the overall power dissipation for the product.

PCI Local Bus Interface

The Intel 855GME / 6300ESB chipset supports the Celeron M processors with up to 400MHz front side bus. The XVME 689-VR7 incorporates one PCI-X bus which is used to service the two Intel 82546EB Ethernet controllers and the on-board PMC site. The PMC site supports both 32-bit/33MHz and 64-bit/66MHz bus speeds with 5V I/O support. The XVME 689-VR7 supports on PCI bus for 32-bit/33MHz operation, this bus services the PCI to VME bridge chip known as the tundra Universe II chip and the 80-pin expansion connector used to connect the XVME-976-209 Dual PMC carrier modules. These carrier modules can be “stacked” to allow for up to five (5) PMC sites on one XVME 689-VR7.

PCI-X, or **PCI extended,** is an enhanced version of PCI (Peripheral Component Interconnect) computer bus. Although PCI-X is backward-compatible with traditional PCI devices and systems, this specification implements additional features and performance improvements include 3.3V signaling, increased speed grades, and adaptation to other form factors. PCI-X effectively doubles the speed and amount of data exchanged between the computer processor and peripherals. PCI-X bus was designed for and is ideally suited for server cards such as Fibre Channel, RAID, high-speed networking, and other demanding devices.

Onboard Memory

SDRAM Memory

The XVME 689-VR7 has a socket for a single 200-pin SODIMM, providing 256 MB, 512MB, 1GB and 2GB of ECC DDR 266/333MHz SDRAM. Approved SDRAM suppliers are listed in 0.

Flash BIOS

The XVME 689-VR7 system BIOS is contained in a 1MB flash device to facilitate system BIOS updates. Contact Xembedded support for available updates at support@xembedded.com if needed. Be sure to record your current version number and the reason for the request.

Video Controller

The 855GME Graphics and Memory Controller Hub (GMCH) has a built-in 2D/3D graphics controller. The maximum video modes supported are listed in the following table. The highest supported interlaced monitor mode is 1280x1024, 16-bit/65k color, and 43 Hz. Video output is available on the front panel through a standard 15-pin D shell connector. The graphics controller is in the 855GME which uses up to 64MB main memory as video memory. The 855GME has a built-in 3D graphics engine and its display / render core frequency is up to 200MHz.

Table 1-1 Maximum Video Modes Supported

Resolution	Bit Depth/Colors	Vertical Refresh
640x480	24-bit/16M color	100 Hz
800x600	24-bit/16M color	100 Hz
1024x768	24-bit/16M color	100 Hz
1280x1024	24-bit/16M color	75 Hz
1600x1200	16-bit/65k color	60 Hz

Ethernet Controller

The 82546GB dual Giga-bit Ethernet controller provides a pair of 10/100/1000baseT Ethernet interfaces. The 82546GB contains both the MAC and the physical layer. The RJ-45 connectors on the module's front panel provide auto-sensing for 10Base-T, 100Base and 1000Base -TX connections. Each RJ-45 connector has two indicator lights. When mounted vertically, the top light is the link/activity light and the bottom light (the one closer to the COM ports) is the 10Base-T/100Base-TX indicator. When it is off, the connection is 10Base-T; when it is on, the connection is 100Base-TX. When the Ethernet is switched to the rear optional P0 no lights are available to indicate link or speed. Storage Devices (Hard Drive, Floppy, Compact PCI and On-Board Drive)

EIDE and Floppy Drives

The XVME 689-VR7 primary IDE and floppy drive signals are routed through the P2 connector inner three rows (available in a legacy 96-pin back plane), providing a simplified method of connecting up to two IDE devices and one external floppy drive. The secondary IDE master signals support the optional on-board hard drive or on-board Compact Flash site and the secondary IDE slave signals are not supported.

The XVME-689-VR7 is NOT compatible with the XVME-977 and/or the XVME-979 mass storage modules.

Caution

The IDE controller supports enhanced PIO modes, which reduce the cycle times for 16-bit data transfers to the hard drive. Check with your drive manual to see if the drive you are using supports these modes. The higher the PIO mode, the shorter the cycle time. As the IDE cable length increases, this reduced cycle time can lead to erratic operation. As a result, it is in your best interest to keep the IDE cable as short as possible.

The PIO modes can be selected in the BIOS setup. The Auto configuration will attempt to classify the connected drive if the drive supports the auto ID command. If you experience problems, change the **Transfer Mode** to *Standard*.

Caution

The total cable length must not exceed 18 inches. Also, if two drives are connected, they must be no more than six inches apart. See SATA below for longer cable lengths.

Serial ATA Hard Drive

The XVME 689-VR7 features two (2) SATA-150 drive interfaces out the rear P2 VMEbus connector. The use of the optional rear transition module (XVME 990-VR7) allows for the connection of two drives using standard SATA cables. If your application requires the external drives to be mounted in a location that requires a long cable run, the SATA drives are better suited to that application. SATA cable can be up to 1 meters or 39” long, EIDE have be less than 18” long.

On-Board Hard Drive (Optional module XVME-913)

The on-board hard drive resides as a master on the secondary EIDE port. The XVME-913 is a kit of parts including; 1.8” hard drive, cable, 4 brackets, screws and standoffs. There are no unique drivers required. The XVME 689-VR7 can be booted from the on-board hard drive if configured in the BIOS **Boot** menu. ***NOTE: The XVME 689-VR7 module can accept either an on-board 1.8” hard drive (XVME-913) or the Compact Flash carrier (XVME-912) but not both.***

Compact Flash Site (Optional module XVME-912)

The compact flash socket on the optional carrier module will support type I or type II Compact Flash cards. The compact flash resides as a master on the secondary IDE port. There are no unique drivers required. The XVME 689-VR7 can be booted from the compact flash drive if configured in the BIOS **Boot** menu. ***NOTE: The XVME 689-VR7 module can accept either an on-board 1.8” hard drive (XVME-913) or the Compact Flash carrier (XVME-912) but not both.***

VMEbus Interface

The XVME 689-VR7 uses the PCI local bus to interface to the VMEbus via a PCI to VME bridge device (Tundra Universe IID). The VMEbus interface supports full DMA to and from the VMEbus, integral FIFOs for posted writes, block mode transfers, and read-modify-write operations. The interface contains one master and eight slave images that can be programmed in a variety of modes to allow the VMEbus to be mapped into the XVME 689-VR7 local memory. This makes it easy to configure VMEbus resources in protected and real mode programs The XVME 689-VR7 also incorporates onboard hardware byte-swapping (see Table 1-2). For a complete API, the Xembedded Board Support Packages tailored to your operating system will allow quick programming of your application.

Serial and Parallel Ports

XVME-689-VR7 includes four high-speed 16550-compatible serial ports (RS-232C) with Com 4 capable of RS-232 and RS-422/485 configurations. The Parallel port can be configured for ECP or EPP parallel port. This is done in the SMC SCH3114 LPC Super I/O and programmed via the BIOS. Com ports 1 and 4 are RJ-45's on the front panel and Com 2 and 3 are available out the P2 VMEbus. The COM 4 port differ from front to back (P2), the front COM port 4 CAN NOT be configured as a RS-232/422/485 but the rear P2 COM port 4 can be configured in this way.

Keyboard / Mouse Interface

A combined keyboard and mouse port PS/2 connector is provided on the front panel. A PS/2 splitter cable part number 140232 (provided with the module) may be used to separate the two ports so that both devices may be simultaneously connected to the module. IF a mouse is not required, a keyboard can be connected directly to the PS/2 port. The keyboard and mouse are controlled in the SMC SCH3114 LPC Super I/O.

PMC Expansion

The XVME 689-VR7 provides an on-board PMC site for use with standard 32/64-bit, 33/66MHz PMC and PMC-X modules. The PMC site is serviced by the on-board PCI-X bus. For electrical isolation, the PMC front panel bezel is not connected to the main CPU ground. **PCI-X**, or **PCI extended,** is an enhanced version of PCI (Peripheral Component Interconnect) computer bus. Although PCI-X is backward-compatible with traditional PCI devices and systems, this specification implements additional features and performance improvements include 3.3V signaling, increased speed grades, and adaptation to other form factors. PCI-X effectively doubles the speed and amount of data exchanged between the computer processor and peripherals. PCI-X bus was designed for and is ideally suited for server cards such as FPGA, DSP, Fibre Channel, RAID, high-speed networking, and other demanding devices. If a standard PCI PMC card is fitted on the XVME 689-VR7 PMC site, the on-board PCI-X bus reverts to the PCI bus speed.

Additional PMC Expansion Options

The XVME 689-VR7 supports optional PMC (PCI Mezzanine Card) expansion using XVME-976-209 expansion module. The XVME-976-209 provides two PCI Mezzanine Card (PMC) sites. The XVME-976-209 module is designed to plug directly into the XVME 689-VR7 using the 80-pin expansion board connector. Another XVME-976-209 can be used to extend the XVME 689-VR7 and first XVME-976-209 to five PMC card sites.

Watchdog Timer

The XVME 689-VR7 has a long duration watchdog timer which can count from 1 to 255 seconds or from 1 to 255 minutes. The BIOS supports various system events which can be routed to the watchdog timer. The timer when enabled can generate either an interrupt or a master reset depending on how the watchdog timer is configured.

Note

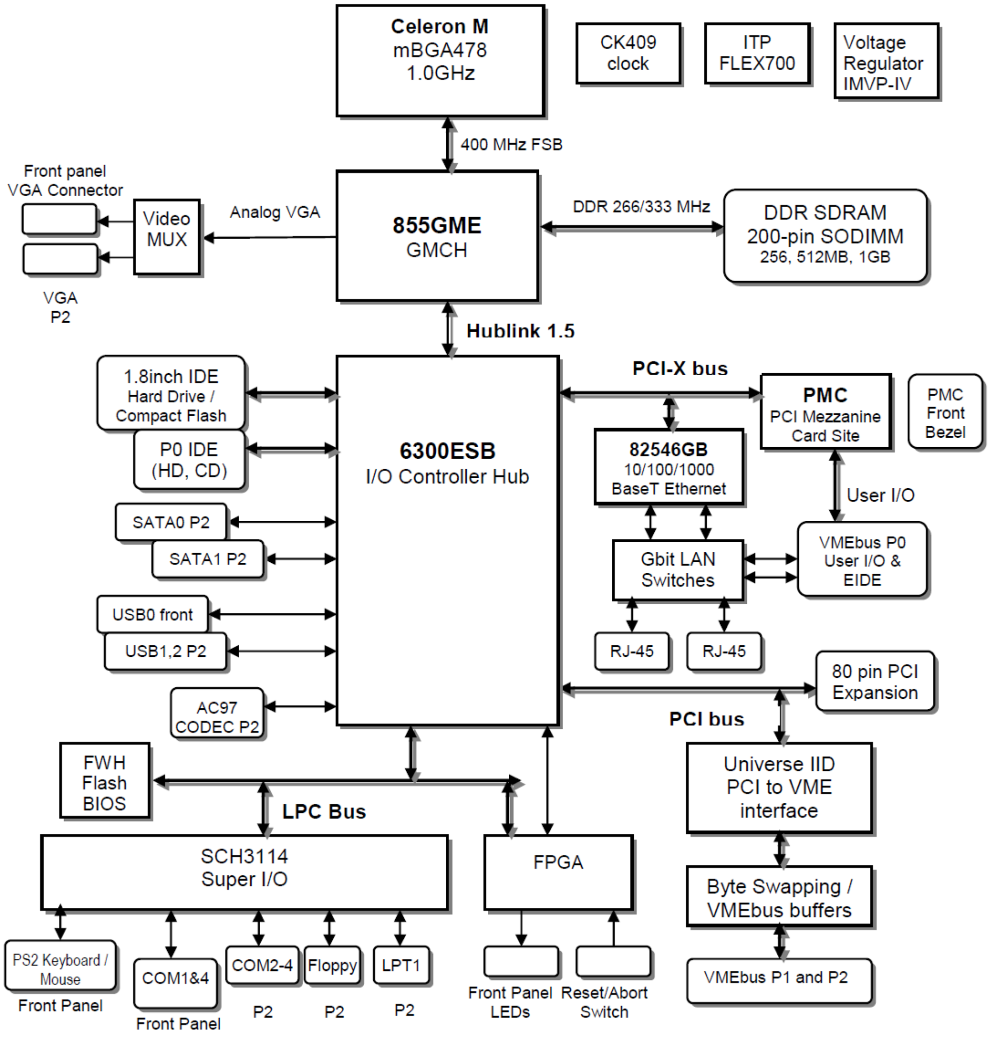
The timeout range is from 1.0 second to 2.25 seconds; it will typically be 1.6 seconds.

Software Support

The XVME 689-VR7 is fully PC-compatible and will run "off-the-shelf" PC software, but most packages will not be able to access the features of the VMEbus. To solve this problem, Xembedded has developed extensive Board Support Packages (BSPs) that simplify the integration of VMEbus data into PC software applications. Xembedded's BSPs provide users with an efficient high-level interface between their applications and the VMEbus-to-PCI bridge device. Board Support Packages are available for MS-DOS®, Windows NT®, Windows 2000®, Windows XP, Windows XP Embedded, Linux, VxWorks, and QNX®.

Operational Description

Figure 1-1 XVME 689-VR7 Block Diagram



Environmental Specifications

The XVME 689-VR7 will meet the following environmental requirements:

Environmental Specification	Operating	Non-Operating
Thermal		-40 to 85 C
Humidity	10% to 90% RH, non-condensing	10% to 90% RH, non-condensing
Shock	30 g peak acceleration, 11 msec duration	50 g peak acceleration, 11 msec duration
Vibration 5 – 2000 Hz	0.015” (0.38mm) peak-to-peak displacement, 2.5 g maximum acceleration	0.030” (0.76mm) peak-to-peak displacement, 5 g maximum acceleration
Emissions	EN 55022	EN 55022
Immunity	EN 50082-2	EN 50082-2

Hardware Specifications

Characteristic	Specification
Power Specifications:	5.4 A (typical); 10.5 A (maximum)
Voltage Specifications:	+5V, +12V, -12V; all +5%/-2.5%
CPU speed: Intel Celeron M Low Power Processor	1.8 GHz
L2 Cache: Intel Celeron M Low Power Processor	2 MB
Onboard memory	SDRAM, up to 2 GB (one 200-pin SODIMM)
Integrated Graphics Controller	1600 x 1200 maximum resolution, 24-bit color maximum; 4 MB Shared memory
Ethernet Controllers (2)	Intel 82546GB 10/100/1000Base-TX Gigabit Ethernet; RJ-45
Mass Storage	
Integrated SATA-150 Controller	SATA0 and SATA1 via P2
EIDE Ultra DMA 100 interface	2 channels via P2
	One 1.8" on-board EIDE via optional carrier module
Compact Flash	One, on-board Compact flash site via optional carrier module
Floppy Drive	Via P2 to XVME 977
PMC Site	On board 66 MHz/64 Bit PMC/PCI-X with front and P0 I/O Access. Site is 3.3V interface level
	Optional 32bit / 33 MHz sites available via XVME-976-209 (2 sites, total 3) and XVME-976-210 (4 sites, total 5)
Stereo Audio	AD1981B AC97 CODEC, Line Level Stereo Input and Output Via P2
USB	One USB 2.0 via Front panel
	Two USB 2.0 via P2
Serial Ports	RS-232C, 16550 compatible (4) COM1 Front, Com 2 and 3 Rear I/O, (Com 4 can be configured for RS-232/422/485 but only the COM 4 out the P2, Com 2 and 3 are RS-232 only). The COM 4 out the front is RS-232 data leads only.
Parallel Interface	EPP/ECP compatible (1)
Keyboard and Mouse	Via Front Panel
Regulatory Compliance	European Union – CE; Electromagnetic Compatibility - 89/336/EEC RoHS Compliant product available

VMEbus Specification

VMEbus Compliance

Complies with VMEbus Specification ANSI/VITA 1–1994
A32/A24/A16:D64/D32/D16/D08(EO) DTB Master
A32/A24/A16:D64/D32/D16/D08(EO) DTB Slave
R(0-3) Bus Requester
Interrupter I(1)-I(7) DYN
IH(1)-IH(7) Interrupt Handler
SYSCLK and SYSRESET Driver
PRI, SGL, RRS Arbiter
RWD, ROR bus release
Form Factor: DOUBLE 233.7 mm x 160 mm (9.2" x 6.3")

System Configuration and Expansion Options Tables

Table 1-2 XVME 689-VR7 CPU configurations

Ordering Number	CPU Type
XVME 689-VR7/1XY	**Intel® Celeron® M 1.0GHz**

Handle and P0 configurations	Memory configurations
Y = 1 VME-64 IEEE 1101.1 (Std / legacy) handles w/o VMEbus P0	**X = 1 for 256 MB ECC DRR SDRAM**
Y = 2 VME-64 IEEE 1101.10 (Compact PCI type) handles w/o VMEbus P0	**X = 2 for 512 MB ECC DRR SDRAM**
Y = 3 VME-64 IEEE 1101.1 (Std / legacy) handles with the VMEbus P0	**X = 3 for 1 GB ECC DRR SDRAM**
Y = 4 VME-64 IEEE 1101.10 (Compact PCI type) handles with the VMEbus P0	

Note: Some features on the XVME 689-VR7 are only available in a 160-pin (5-Row) VMEbus P2 backplane.

The ordering number is broken into two parts. The model number is the 689-VR7. The tab number is the three digits after the slash. For the XVME 689-VR7, the tab number indicates the CPU type, amount of SDRAM memory (the middle digit) and the ejector handle type and optional P0 connector.

.

Table 1-3 XVME 689-VR7 Expansion Module Options

Ordering Number	Description
XVME 990-VR7/1	Drive Adapter Module for external drives, cables out back of VME backplane, Primary PIDE (2 PIDE), Floppy, COM2 (Only RS-232), two ports of SATA-150, LPT1, 1 USB port, Audio in/out and Analog Video out plus P0 for rear I/O from PMC site and Ethernet 31.1 or rear RJ-45 Ethernet.
XVME 990-VR7/2	Drive Adapter Module for external drives, cables out back of VME backplane, Primary PIDE (2 PIDE), Floppy, COM2 (Only RS-232), two ports of SATA-150, LPT1, 1 USB port, Audio in/out and Analog Video out
XVME-976-209	PMC Carrier module with two PMC module sites. The XVME-976-209 allows for stacking of a second XVME-976-209, this combination will accommodate a total of five PMC modules
XVME-977/011	Not Compatible with the XVME-689-VR7
XVME-979/1	Not Compatible with the XVME-689-VR7
XVME-979/2	Not Compatible with the XVME-689-VR7
XVME-979/3	Not Compatible with the XVME-689-VR7
XVME-979/4	Not Compatible with the XVME-689-VR7
XVME-979/5	Not Compatible with the XVME-689-VR7
XVME-979/6	Not Compatible with the XVME-689-VR7
XVME-9000-EXF	Not Compatible with the XVME-689-VR7

The XVME 990-VR7 is described in **Error! Reference source not found.**.

Chapter 2 – Installation and Setup

Board Layout

This chapter provides information on configuring the XVME 689-VR7 modules. It also provides information on installing the XVME 689-VR7 into a backplane and enabling the Ethernet controller.

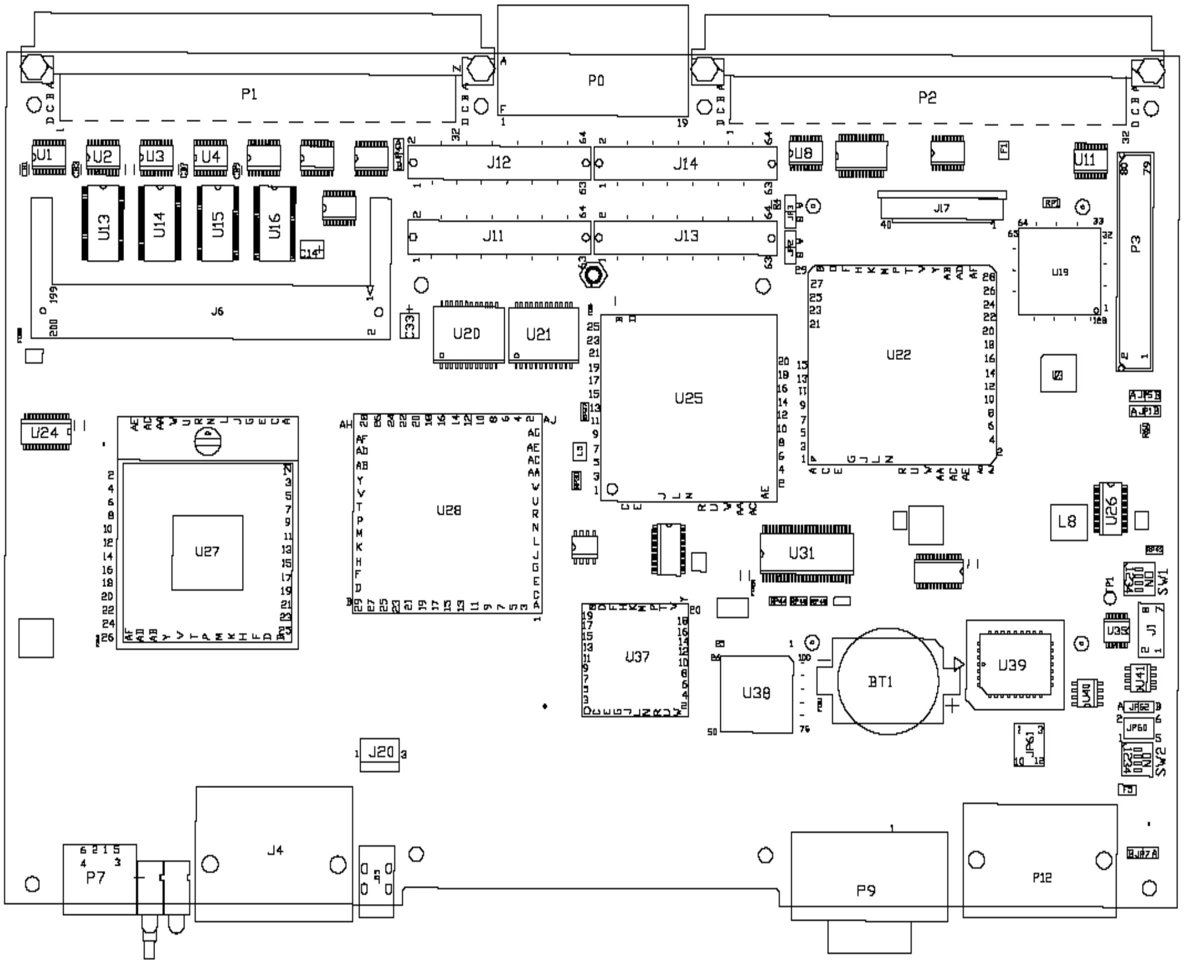


Fig. 2-1 shows the jumper, switch, and connector locations on the XVME 689-VR7.

3Jumper Settings

The following table Lists XVME689-VR7 jumpers, their default positions, and their functions.

Table 2-1 XVME 689-VR7 Jumper Settings

Jumper	Position	Function
JP2	A B √	XVME 689-VR7 cannot generate SYSFAIL* XVME 689-VR7 generates SYSFAIL* normally
JP3	A B √	Disables system resources (no auto SYSCON) Enables system resources (auto SYSCON)
JP4	A √ B	XVME 689-VR7 can reset VMEbus XVME 689-VR7 cannot reset VMEbus
JP5	A √ B	Normal Clear CMOS
JP6	A √ B	Boot from FLASH Boot from ROM
JP7	A B √	Orb ground not connected to logic ground Orb ground connected to logic ground
JP8	A √ B	Normal (Video out Front Panel VGA) Video re-routed to P2 connector (no video out front panel)
JP60	1 ----- 2 3 4 5 6	SIO_COM_RXD 4 = RXD4 SIO_COM_RXD 4 = RXD422 SIO_COM_RXD 4 = RXD485
JP61	1 ----- 2 2 3 4 ----- 5 5 6 7 ----- 8 8 9 10 ----11 11 12	DCD4 485TXD- DSR4 485TXD+ RI4 485RXD- DTR4 485RXD+
JP62	A √ B□	RS-232 RS-422/RS-485

✓ denotes default

Switch Settings

The XVME 689-VR7 has one four-pole switch (SW1) shown in Figure 2-2. The switches functions are explained in table 2-2. This switch controls the system response to the front panel **Abort** switch. Figure 2-2 shows the switch settings required to reset on the XVME 689-VR7 CPU, to reset only the VME backplane, or to reset both. The XVME 689-VR7 is shipped with all four switches in the closed position (which causes the push button reset switch to reset both the XVME 689-VR7 and the VME backplane).

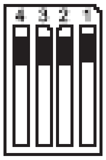


Figure. 2-2

SW1 XVME 689-VR7 has a 4 position DIP switch to control the following functions.

Table 2-2 Four-Pole Switch (SW1) Functions

Position	Open	Closed
1	Do not respond to SYSRESET*	Respond to SYSRESET*
2	Toggle switch does not cause SYSRESET*	Toggle switch causes SYSRESET*
3	SYSFAIL* asserted on Power Up	SYSFAIL* not asserted on Power
4	No local reset	Toggle switch causes local reset

Registers

The XVME 689-VR7 modules contain the following Xembedded-defined I/O registers: 218h, 219h, 233h, and 234h.

Register 218h – Abort/CMOS Clear Register

This register controls the abort toggle switch and allows you to read the CMOS clear jumper (main board J21).

Table 2-3 Abort/CMOS Clear Register Settings

Bit	Signal	Result	R/W
0	RESERVED	Reserved	
1	RESERVED	Reserved	
2	RESERVED	Reserved	
3	RESERVED	Reserved	
4	ABORT_STS	1 = Abort toggle switch caused interrupt	R
5	ABORT_CLR	0 = Clear and disable abort 1 = Enable abort	R/W
6	RESERVED	Reserved	
7	CLRCMOS	0 = Clear CMOS 1 = CMOS okay	R

Register 219h – Flash Control Register

This register controls the following LEDs and signals.

Table 2-4 LED/BIOS Register Settings

Bit	LED/Signal	Result	R/W
0	FAULT	0 = Fault LED on 1 = Fault LED off	R/W
1	PASS	0 = PASS LED off 1 = PASS LED on	R/W
2	N/A	N/A	
3	N/A	N/A	
4	RESERVED	Reserved	
5	RESERVED	Reserved	
6	RESERVED	Reserved	
7	RESERVED	Reserved	

Register 233h – Watchdog Timer Register

This register controls watchdog timer operation.

Table 2-5 Watchdog Timer Register Settings

Bit	Signal	Result
0	RESERVED	Reserved
1	RESERVED	Reserved
2	RESERVED	Reserved
3	RESERVED	Reserved
4	WDOG_EN	1 = Enables the watchdog timer
5	MRESET_EN	1 = Timeout generates 0 = Timeout generates IRQ10
6	WDOG_STS	Watchdog timer status bit
7	WDOG_CLR	Toggling this bit clears the watchdog timer back to a zero count.

Note

Before enabling the watchdog timer for the first time, it is necessary to reset the count back to zero by toggling bit 7 (WDOG_CLR). Toggling implies changing the state of bit (0 to 1 or 1 to 0).

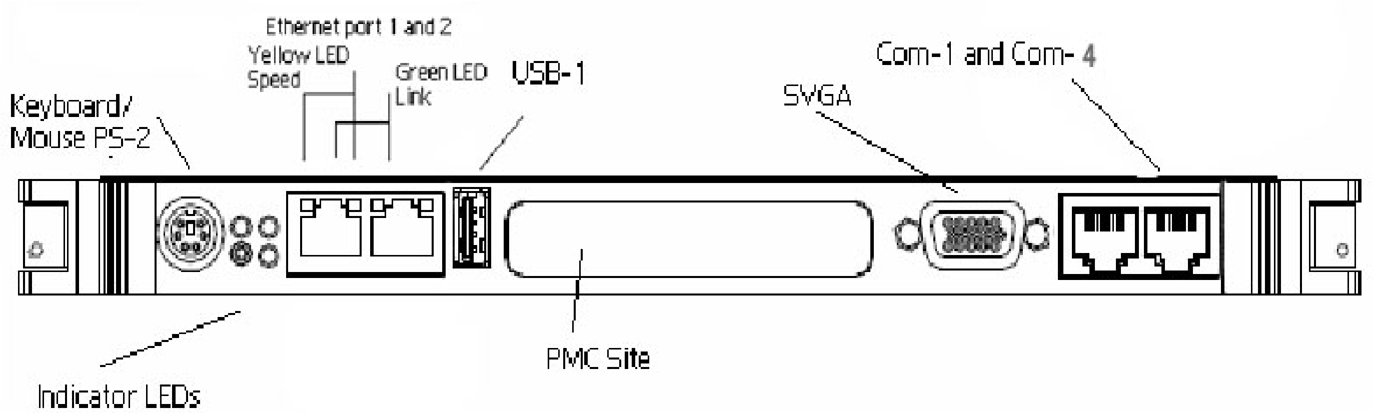
Register 234h – Flash Paging and Byte Swap Register

This register controls access to the Flash paging and byte-swapping functions.

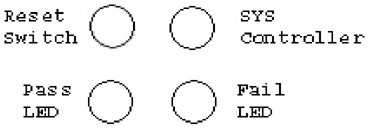
Table 2-6 Flash Paging and Byte Swap Register Settings

Bit	Signal	Result
0	FLB_A15	Flash address 15 - page control bit
1	FLB_A16	Flash address 16 - page control bit
2	FLB_A17	Flash address 17 - page control bit
3	Unused – set to 0	Do not use.
4	Unused – set to 0	Do not use.
5	Unused – set to 0	Do not use.
6	SWAPS	1 = No swapping (data invariant) occurs during slave cycles
7	SWAPM	1 = No swapping (data invariant) occurs during master cycles

Front Panel Layout



Panel LEDs and Switch



The reset switch can be enabled to reset see the setup of Sw-1 shown in Figure 2-2 and table 2-2. This switch can be configured to either just reset the XVME 689-VR7 or to reset both the VMEbus and the XVME 689-VR7. The green pass and red fail LEDs are used as an indication of board health during the BIOS boot up. Both the green pass and red fail LEDs will light during the POST of the board. As the BIOS complete the POST, the red fail LED will be turned off. This is an indication the XVME 689-VR7 has passed the POST. The Green SYS Controller LED is lit when the XVME 689-VR7 is configured as the VMEbus system controller. This is the function that grants bus ownership to multiple bus VME masters and provides the 16MHz clock signal on the back plane.

Connectors

This section provides pin outs for the XVME 689-VR7 connectors. Refer to the EMC warning at the beginning of this manual before attaching cables.

Keyboard/Mouse Port Connector (P7)

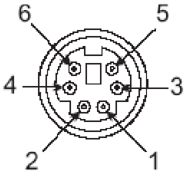


Table 2-7 Keyboard Port Connector Pin out

Pin	Signal
1	Keyboard Data
2	Mouse Data
3	GND
4	+5V
5	Keyboard Clock
6	Mouse Clock

Front panel (P12 or Rear Ethernet Port and Vita 31.1 Ethernet

The Ethernet ports on the XVME 689-VR7 are switch able between the front and the rear of the XVME 689-VR7. When in the rear mode, the Ethernet port can use the PO connector for either Vita 31.1 switch fabric over the Vita 31.1 compliant backplane or Ethernet out the rear or the module.

Table 2-8 RJ-45 10/100/1000 BaseT Connector Pin out

Pin	Signal
1	TX+
2	TX-
3	RX+
4	GND
5	GND
6	RX-
7	GND
8	GND

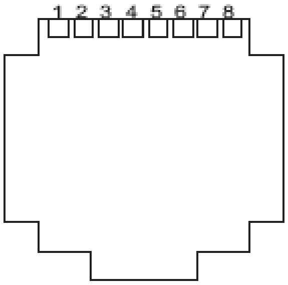


Figure 2-8 RJ-45 10/100/1000Mbps

VGA Connector (P9)

The video is BIOS selectable and is available on either the front panel on a standard SVGA connector or out the VMEbus P2. The table below shows the pin out of the VMEbus P2 pin out for the rear access of video.

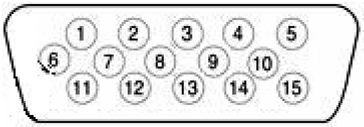


Figure 2-9 SVGA Connector

Table 2-9 VGA Connector Pin out

SVGA Pin out	Signal	VMEbus P2
1	RED	Row d Pin-20
2	GREEN	Row d Pin-21
3	BLUE	Row d Pin-22
4	NC	N/C
5	GND	Row d Pin-31
6	GND	Row d Pin-31
7	GND	Row d Pin-31
8	GND	Row d Pin-31
9	25MIL_VIDA	N/C
10	GND	Row d Pin-31
11	NC	N/C
12	LDDCDAT	Row d Pin-26
13	HSYNC	Row d Pin-23
14	VSYNC	Row d Pin-24
15	LDDCCLK	Row d Pin-25

USB Port Connector (J5)

USB provides an expandable, hot-pluggable Plug and Play serial interface that ensures a standard, low-cost connection for peripheral devices. Devices suitable for USB range from simple input devices such as keyboards, mice, and joysticks, to advanced devices such as printers, scanners, storage devices, modems, and video conferencing cameras. USB 2.0 has a raw data rate at 480Mbps, and it is rated 40 times faster than its predecessor interface, USB 1.1, which tops at 12Mbps.

USB port 1 is available on the front panel using a standard connector as shown in Figure 2-10 below. The other two USB ports USB-2 and 3, are routed out the VMEbus P2 connector and can be accesses either directly off the VMEbus P2 connector using the pin assignment shown in Fig. 2-10. The USB +5 V supplies are protected with a polyswitch. This device will open up if +5 V is shorted to GND. Once the shorting condition is removed, the polyswitch will allow current flow to resume.

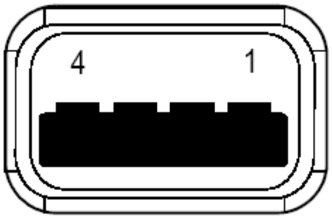


Figure 2-10 USB Connector

VMEbus P2	Signal Name
P2-row-z 17	USB3_GND
P2-row-z 18	GND
P2-row-z 19	USB3+
P2-row-z 20	GND
P2-row-z 21	USB3-
P2-row-z 22	GND
P2-row-z 23	USB3_PWR
P2-row-z 24	GND
P2-row-z 25	USB2_GND
P2-row-z 26	GND
P2-row-z 27	USB2+
P2-row-z 28	GND
P2-row-z 29	USB2-
P2-row-z 30	GND
P2-row-z 31	USB2_PWR
P2-row-z 32	GND

Table 2-11 Rear USB ports 2 and 3

USB Port Connector Pin out on VMEbus P2

Pin	Signal
1	+5V
2	USBP0-
3	USBP0+
4	GND

Table 2-10 USB Port Connector Pin out

COM1 and COM4 (J4) Pin Definitions

The XVME 689-VR7 has two serial ports out the front panel, Com 1 and Com 4, these two com ports use the RJ-45 connector. Two more com ports are out the VMEbus P2 connector. See below for connector layout and pin descriptions.

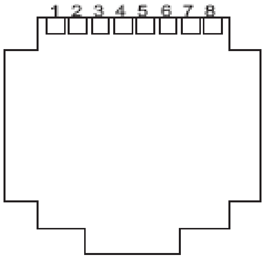


Figure 2-11 RJ-45 Serial Port Connectors

Com ports 1 and 4 Pin Definitions for front connectors

Pin Number	RS232
1	RTS
2	DTR
3	TXD
4	GND
5	GND
6	RXD
7	DSR
8	CTS

Table 2-13 Serial Port Connector Pin out for Comm-1 and 4

On-Board Hard Drive/Compact Flash Site (J17)

A horizontal ZIF connector is used on the board. (Samtec part number ZF5-40-01-TM-WT.) The connector on the board has a reverse pin out because of the connector orientation relative to the hard drive. This allows the flex cable to loop up to the hard drive, with the connector side facing the board.

Table 2-14 On-Board storage devices us the J17 1.8inch Hard Drive Connector pin assignment

Pin #	Description	Pin #	Description	Pin #	Description	Pin #	Description
1	factory use	11	DD4	21	GROUND	31	DA1
2	factory use	12	DD11	22	DMARQ	32	PDIAG-
3	RESET-	13	DD3	23	GROUND	33	DA0
4	GROUND	14	DD12	24	DIOW-	34	DA2
5	DD7	15	DD2	25	DIOR-	35	CS0-
6	DD8	16	DD13	26	GROUND	36	CS1-
7	DD6	17	DD1	27	IORDY	37	DASP-
8	DD9	18	DD14	28	GROUND	38	3.3V
9	DD5	19	DD0	29	DMACK-	39	3.3V
10	DD10	20	DD15	30	INTRQ	40	DEVADR

The Hitachi C4K60 CE has a 40 pin ZIF connector.

Table 2-15 1.8inch Hard Drive Connector pin assignment on hard drive

Pin #	Description	Pin #	Description	Pin #	Description	Pin #	Description
40	factory use	30	DD4	20	GROUND	10	DA1
39	factory use	29	DD11	19	DMARQ	9	PDIAG-
38	RESET-	28	DD3	18	GROUND	8	DA0
37	GROUND	27	DD12	17	DIOW-	7	DA2
36	DD7	26	DD2	16	DIOR-	6	CS0-
35	DD8	25	DD13	15	GROUND	5	CS1-
34	DD6	24	DD1	14	IORDY	4	DASP-
33	DD9	23	DD14	13	GROUND	3	3.3V
32	DD5	22	DD0	12	DMACK-	2	3.3V
31	DD10	21	DD15	11	INTRQ	1	DEVADR

VMEbus Connectors

VMEbus P1 Connector

Table 2-16 P1 Connector Pin out

Pin	Z	A	B	C	D
1*	_MPR_	D00	BBSY*	D08	+5V
2	GND	D01	BCLR*	D09	GND
3*	_MCLK_	D02	ACFAIL*	D10	_+V1_
4	GND	D03	BG0IN*	D11	_+V2_
5*	_MSD_	D04	BG0OUT*	D12	_RSVU1_
6	GND	D05	BG1IN*	D13	_-V1_
7*	_MMD_	D06	BG1OUT*	D14	_-V2_
8	GND	D07	BG2IN*	D15	_RSVU2_
9*	_MCTL_	GND	BG2OUT*	GND	_GAP*_
10	GND	SYSCLK	BG3IN*	SYSFAIL*	_GA0*_
11*	_RESP*_	GND	BG3OUT*	BERR*	_GA1*_
12	GND	DS1*	BR0*	SYSRESET*	
13	_SDB14*_	DS0*	BR1*	LWORD*	_GA2*_
14	GND	WRITE*	BR2*	AM5	
15	_SDB15*_	GND	BR3*	A23	_GA3*_
16	GND	DTACK*	AM0	A22	
17	_SDBP1_	GND	AM1	A21	_GA4*_
18	GND	AS*	AM2	A20	
19	_RSVBUS5_	GND	AM3	A19	_RSVBU1_
20	GND	IACK*	GND	A18	
21	_RSVBUS6_	IACKIN*	NC	A17	_RSVBU2_
22	GND	IACKOUT*	NC	A16	
23	_RSVBUS7_	AM4	GND	A15	_RSVBU3_
24	GND	A07	IRQ7*	A14	
25	_RSVBUS8_	A06	IRQ6*	A13	_RSVBU4_
26	GND	A05	IRQ5*	A12	
27	_RSVBUS9_	A04	IRQ4*	A11	_LI/I*_
28	GND	A03	IRQ3*	A10	
29	_RSVBUS10_	A02	IRQ2*	A09	_LI/O*_
30	GND	A01	IRQ1*	A08	
31	_RSVBUS11_	-12V	NC	+12V	GND
32	GND	+5V	+5V	+5V	+5V

Some pins in columns Z and D are use internally as test points, these are denoted by italics. These pins are not intended to drive any external devices and MUST not be used for any purpose.

VMEbus P2 Connector

Pin Number	Row Z	Row A	Row B	Row C	Row D
1	FD_DRV0	VGA_RED	+5V	USB0-	FD_INDX
2	GND	VGA_GREEN	GND	USB0+	FD_DCHG
3	FD_DIR	VGA_BLUE	VME_RETRY*	USB1-	FD_MTR0
4	GND	VGA_HSYN	A24	USB1+	FD_STEP
5	FD_WGAT	GND	A25	VGA_VSYN	FD_WDAT
6	GND	AUD_LINE_IN_L	A26	NC	FD_TRK0
7	FD_RDAT	AUD_LINE_IN_R	A27	NC	FD_WPRT
8	GND	AUD_LINE_OUT_L	A28	NC	FD_HDSL
9	LPT1_AFED	AUD_LINE_OUT_R	A29	NC	LPT1_STRO
10	GND	GND	A30	GND	LPT1_D0
11	LPT1_ERRO	SATA_TXP0	A31	SATA_TXP1	LPT1_D1
12	GND	SATA_TXN0	GND	SATA_TXN1	LPT1_D2
13	LPT1_INT	GND	+5V	GND	LPT1_D3
14	GND	SATA_RXP0	VD16	SATA_RXP1	LPT1_D4
15	LPT1_SLIN	SATA_RXN0	VD17	SATA_RXN1	LPT1_D5
16	GND	GND	VD18	GND	LPT1_D6
17	LPT1_ACK	AUD_LINE_OUT_L	VD19	\RST_BUT	LPT1_D7
18	GND	KYBD DATA	VD20	KYBD CLK	LPT1_BUSY
19	LPT1_SLCT	MOUSE DATA	VD21	MOUSE CLK	LPT1_PE
20	GND	ETH2 PIN 1(TXA+)	VD22	ETH2 PIN 2(TXA-)	WDG_REL
21	COM3_DSR	ETH2 PIN 3(RXA+)	VD23	ETH2 PIN 4(NC)	COM3_DCD
22	GND	ETH2 PIN 5(NC)	GND	ETH2 PIN 6(RXA-)	COM3_RXD
23	COM3_TXD	ETH2 PIN 7(NC)	VD24	ETH2 PIN 8(NC)	COM3_RTS
24	GND	COM1_DSR	VD25	COM1_DCD	COM3_CTS
25	COM3_RI	COM1_RTS	VD26	COM1_RXD	COM3_DTR
26	GND	COM1_CTS	VD27	COM1_TXD	COM4_DCD(C4_TXD-)
27	COM4_RXD	COM1_RI	VD28	COM1_DTR	COM4_DSR(C4_TXD+)
28	GND	COM2_DCD	VD29	GND	COM4_RTS
29	COM4_CTS	COM2_RXD	VD30	COM2_DSR	COM4_TXD
30	GND	COM2_TXD	VD31	COM2_RTS	COM4_DTR(C4_RXD+)
31	COM4_RI(C4_RXD-)	COM2_DTR	GND	COM2_CTS	GND
32	GND	GND	+5V	COM2_RI	+5V

Table 2-17 VMEbus P2 Connector

VMEbus P0 Connector

PIN	F	E	D	C	B	A
1	GND	IDE_PDDACK#	N/C	N/C	N/C	PMC I/O 32
2	GND	IDE_PDDREQ	N/C	N/C	PMC I/O 17	PMC I/O 1
3	GND	GND	N/C	N/C	PMC I/O 18	PMC I/O 2
4	GND	IDE_PDDCS1#	N/C	N/C	PMC I/O 19	PMC I/O 3
5	GND	IDE_PDDA0	N/C	N/C	PMC I/O 33	PMC I/O 4
6	GND	IDE_PDDA1	N/C	N/C	PMC I/O 20	PMC I/O 5
7	GND	IDE_IRQ14	N/C	N/C	PMC I/O 34	PMC I/O 6
8	GND	IDE_PIORDY	N/C	N/C	PMC I/O 21	PMC I/O 7
9	GND	IDE_PIORDR#	IDE_PDCS3	N/C	PMC I/O 22	PMC I/O 8
10	GND	IDE_PIORDW#	IDE_PDA2	N/C	PMC I/O 23	PMC I/O 35
11	GND	IDE_PDD0	IOCS16#	N/C	PMC I/O 24	PMC I/O 9
12	GND	IDE_PDD1	IDE_PDD15	N/C	PMC I/O 25	PMC I/O 10
13	GND	IDE_PDD2	IDE_PDD14	N/C	PMC I/O 26	PMC I/O 11
14	GND	IDE_PDD3	IDE_PDD13	N/C	GND	PMC I/O 12
15	GND	IDE_PDD4	IDE_PDD12	N/C	PMC I/O 27	PMC I/O 13
16	GND	IDE_PDD5	IDE_PDD11	N/C	PMC I/O 28	PMC I/O 14
17	GND	IDE_PDD6	IDE_PDD10	N/C	PMC I/O 29	GND
18	GND	IDE_PDD7	IDE_PDD9	N/C	PMC I/O 30	PMC I/O 15
19	GND	IDE_PDRST#	IDE_PDD8	N/C	PMC I/O 31	PMC I/O 16

80pin PCI connector (P3)

The P3, high speed micro-strip connector has all the PCI signals along with 2 separate PCI clocks and the 2 request and grants predefined. The CPU board and the Interface boards will be keyed for either 3.3V or 5V signaling. The keying mechanism is based on standoffs. At this point all CPU boards will be 5V PCI signaling. The V/IO pins on the connector are used to define the signaling level to the other PCI boards. This connector is used to attach the XVME-976-209 dual PMC carrier. The XVME-976-209 can support one additional carrier module stacked to create a three slot set of boards that can support five PMC cards.

Table 2-18 PCI bus interconnect for optional carrier module

P3 Pin Number	Name	P3 Pin Number	Name
1	TCLK (P.D.)	41	AD(23)
2	TRST* (P.D.)	42	AD(22)
3	TMS (P.U.)	43	AD(21)
4	TDO (NC)	44	AD(20)
5	TDI (P.U.)	45	AD(19)
6	+12v	46	AD(18)
7	+12V	47	AD(17)
8	NC	48	AD(16)
9	NC	49	BE2*
10	-12V	50	FRAME*
11	-12V	51	IRDY*
12	NC	52	TRDY*
13	NC	53	DEVSEL*
14	NC	54	STOP*
15	NC (PCLKS3) (note 1)	55	PLOCK*
16	PIRQA*	56	PERR*
17	PIRQB*	57	SDONE (P.U.)
18	PIRQC*	58	SBO* (P.U.)
19	PIRQD*	59	SERR*
20	REQ3*	60	PAR
21	NC(PCLKS2) (note 1)	61	BE1*
22	REQ1*	62	AD(15)
23	GNT3*	63	AD(14)
24	PCICLK1	64	AD(13)
25	GNT1*	65	AD(12)
26	PCIRST*	66	AD(11)
27	PCICLK0	67	AD(10)
28	GNT0*	68	AD(9)
29	REQ0*	69	AD(8)
30	REQ2*	70	BE0*
31	AD(31)	71	AD(7)
32	AD(30)	72	AD(6)
33	AD(29)	73	AD(5)
34	AD(28)	74	AD(4)
35	AD(27)	75	AD(3)
36	AD(26)	76	AD(2)
37	AD(25)	77	AD(1)
38	AD(24)	78	AD(0)
39	BE3*	79	ACK64* (P.U.)
40	GNT2*	80	REQ64* (P.U.)

Although not shown, the P3 connector supplies Vi/o = +5v, VCC=+5V, and GND through the center pins. Notes: (1) PCICLK2 and PCICLK3 are not supplied by the XVME 689-VR7. These clocks were needed for on board PCI devices and were not used by any currently supported daughtercards.

PMC Host Connectors

PMC Host Connector 1

Table 2-19 XVME 689-VR7 Daughterboard PMC Host Connector 1 Pin out

Pin	Signal	Pin	Signal
1	TCK	33	FRAME*
2	-12V	34	GND
3	GND	35	GND
4	INTA*	36	IRDY*
5	INTB*	37	DEVSEL*
6	INTC*	38	+5V
7	BUSMODE1*	39	GND
8	+5V	40	PLOCK*
9	INTD*	41	SDONE
10	PCI-RSVD14B	42	SBO*
11	GND	43	PAR
12	PCI-RSVD14A	44	GND
13	PCICLK	45	V_I/O
14	GND	46	AD(15)
15	GND	47	AD(12)
16	GNT*	48	AD(11)
17	REQ*	49	AD(9)
18	+5V	50	+5V
19	V_I/O	51	GND
20	PAD(31)	52	C_BE*(0)
21	PAD(28)	53	AD(6)
22	PAD(27)	54	AD(5)
23	PAD(25)	55	AD(4)
24	GND	56	GND
25	GND	57	V_I/O
26	C_BE*(3)	58	AD(3)
27	AD(22)	59	AD(2)
28	AD(21)	60	AD(1)
29	AD(19)	61	AD(0)
30	+5V	62	+5V
31	V_I/O	63	GND
32	AD(17)	64	REQ64*

PMC Host Connector 2

Table 2-202 XVME 689-VR7 PMC Host Connector 2 Pin out

Pin	Signal	Pin	Signal
1	+12V	33	GND
2	TRST*	34	PMC-RSVD_PN2-34
3	TMS	35	TRDY*
4	TDO	36	+3.3V
5	TDI	37	GND
6	GND	38	STOP*
7	GND	39	PERR*
8	PCI-RSVD9A	40	GND
9	PCI-RSVD10B	41	+3.3V
10	PCI-RSVD11A	42	SERR*
11	BUSMODE2* (V_IO)	43	C_BE*(1)
12	+3.3V	44	GND
13	RST*	45	AD(14)
14	BUSMODE3* (GND)	46	AD(13)
15	+3.3V	47	GND
16	BUSMODE4* (GND)	48	AD(10)
17	PCI-RSVD19A	49	AD(8)
18	GND	50	+3.3V
19	AD(30)	51	AD(7)
20	AD(29)	52	PMC-RSVD_PN2-52
21	GND	53	+3.3V
22	PAD(26)	54	PMC-RSVD_PN2-54
23	PAD(24)	55	NC
24	+3.3V	56	GND
25	IDSEL*	57	NC
26	AD(23)	58	NC
27	+3.3V	59	GND
28	AD(20)	60	NC
29	AD(18)	61	ACK64*
30	GND	62	+3.3V
31	AD(16)	63	GND
32	CE_BE*(2)	64	RES (NC)

CPU Fan Power Connector

The fan +12 V and +5 V supplies are protected with a polyswitch. This device will open up if +12 V or +5 V is shorted to GND. Once the shorting condition is removed, the polyswitch will allow current flow to resume.

Table 2-21 CPU Fan Power Connector Pin out

Pin	Signal
1	GND
2	+12V (fused)
3	+5V pullup

Installing the XVME 689-VR7 into a Backplane

This section provides the information necessary to install the XVME 689-VR7 into the VMEbus backplane. The XVME 689-VR7 is a double-high, single-slot VMEbus module.

Note

Xembedded modules are designed to comply with all physical and electrical VMEbus backplane specifications of VME64x.

Note

The XVME 689-VR7 is available from the factory in two basic configurations, with P0 and without P0. The without P0 would normally be used in a legacy system since most of these racks are equipped with a stiffener bar in the P0 location. Also note that to use the extended features of the XVME 689-VR7, the backplane must use 160-pin P1 and P2.

Caution

Do not install the XVME 689-VR7 on a VMEbus system without a P2 backplane.

Warning

Never install or remove any boards before turning off the power to the bus and all related external power supplies.

1. Disconnect all power supplies to the backplane and the card cage. Disconnect the power cable.
2. Make sure backplane (5 rows) 160-pin connectors P1 and P2 are available.
3. Verify that all jumper settings are correct.
4. Verify that the card cage slot is clear and accessible.
5. Install the XVME 689-VR7 in the card cage by centering the unit on the plastic guides in the slots (P1 connector facing up). Push the board slowly toward the rear of the chassis until the P1 and P2 connectors engage. The board should slide freely in the plastic guides.

Caution

Do not use excessive force or pressure to engage the connectors. If the boards do not properly connect with the backplane, remove the module and inspect all connectors and guide slots for damage or obstructions.

6. Secure the module to the chassis by tightening the machine screws at the top and bottom of the board.
7. Connect all remaining peripherals by attaching each interface cable into the appropriate connector on the front of the XVME 689-VR7 board as shown in Table 2-.

8. Turn on power to the VMEbus card cage.

Table 2-22 Front Panel Connector Labels

Connector	Label
Keyboard/Mouse	KEYBD/ MOUSE
Display cable	VGA
USB cable	USB
Ethernet cable	10/100/1000T
Serial devices	COM 1, COM 3
Parallel device	LPT1
PMC card	PMC

Enabling the PCI Ethernet Controller

Loading the Ethernet Driver

To enable the Ethernet controller, you must load the applicable Ethernet driver for your operating system from the Documentation and Support Library CD included with the XVME 689-VR7. For best results, always use the supplied drivers.

Ethernet RJ-45 10/100/1000 BaseT Connector (P12)

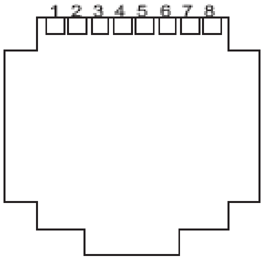


Table 2-23 RJ-45 10/100/1000 BaseT Connector Pin out

Pin	Signal
1	TX+
2	TX-
3	RX+
4	GND
5	GND
6	RX-
7	GND
8	GND

Chapter 3 BIOS Setup Menus

The XVME 689-VR7 customized BIOS is designed to surpass the functionality provided for normal PCs. The custom BIOS allows access to the value-added features on the XVME 689-VR7. Some of the on-board features of the XVME 689-VR7 can also be setup via on-board jumpers, most of the time if a feature can be setup either way, the jumper takes priority over the BIOS settings.

Navigation (moving your cursor around, selecting items, and changing them) is easy in the Setup system. The following chart is a helpful user reference:

Table 3-1 BIOS Keyboard navigation

UP key (also ^E)	Move the cursor to the line above, scrolling the window as necessary.
DOWN key (also ^X)	Move the cursor to the line below, scrolling the window as necessary.
LEFT key	Go back to the menu to the left of the currently-displayed menu in the menu bar.
RIGHT key	Go forward to the menu to the right of the currently-displayed menu in the menu bar.
PGUP key	Move the cursor up several lines (a full window's worth), scrolling the window as necessary.
PGDN key	Move the cursor down several lines (a full window's worth), scrolling the window as necessary.
HOME key	Move the cursor to the first configurable field in the current menu, scrolling the window as necessary.
END key	Move the cursor to the last configurable field in the current menu, scrolling the window as necessary.
ESC key	Exit the Setup system, discarding all changes (except date/time changes, which take place on-the-fly.)
TAB key	Move the cursor down to the next configurable field.
Shift-TAB key (backtab)	Move the cursor up to the last configurable field.
+ key	Toggle an Enable/Disable field, or increase a numeric field's value.
- key	Toggle an Enable/Disable field, or decrease a numeric field's value.
SPACE key	Toggle an Enable/Disable field.
BKSP key	Reset an Enable/Disable or multiple-choice field, or back-up in numeric or string fields.
Digits (0-9)	Used to enter numeric parameters.
Alphabetic (A-Z, a-z)	Used to enter text data on ASCII fields such as email addresses.
Special symbols (!@#$%^&*_-+={}[], etc.)	Used to enter special text on ASCII fields that permit these characters.

The basic idea when using the Setup system is to navigate to the menus containing fields you want to review, and change those fields as desired. When your settings are complete, navigate to the EXIT menu, and select “Save Settings and Restart”. This causes the settings to be stored in nonvolatile memory in the system, and the system will reboot so that POST can configure itself with the new settings.

After rebooting it may be desirable to reenter the Setup system as necessary to adjust settings as necessary.

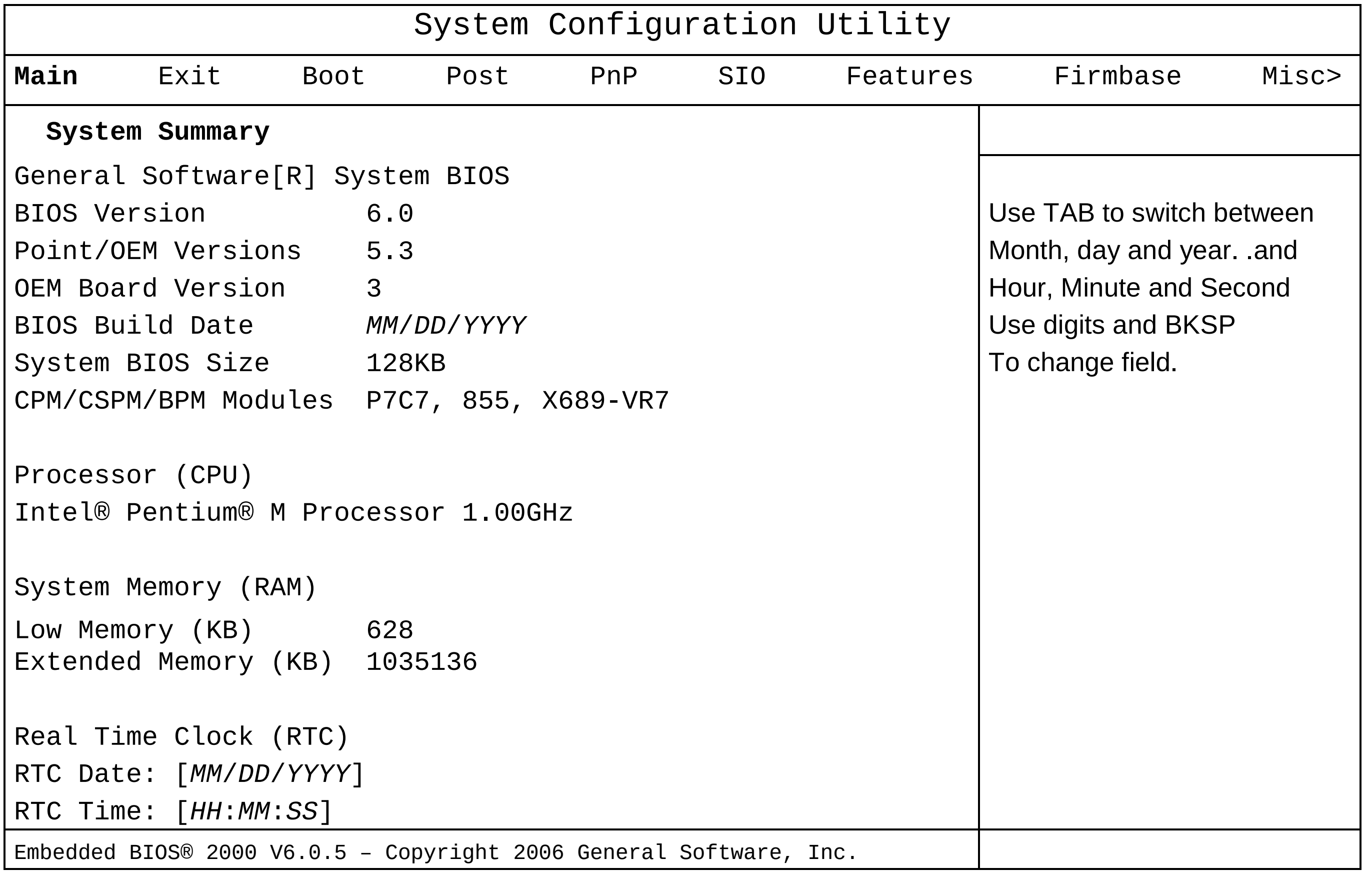
Once the system boots, the Setup system cannot be entered; this is because the memory used by the BIOS configuration manager is deallocated by the system BIOS, so that it can be used by the OS when it boots. To reenter the Setup system after boot, simply press the reset on the front panel or cycle the system power.

Note

The default values given in the descriptions are for the XVME 689-VR7 board with no peripheral devices attached. If drives are connected, their values will be shown.

3.1 Main Setup Menu

This Main menu (System Summary) provides information about the BIOS, processor, system memory, and allows the setup of date and time. Only the date and time fields are user definable in this menu.



```
                       System Configuration Utility
Main    Exit    Boot    Post    PnP    SIO    Features    Firmbase    Misc>

  System Summary
General Software[R] System BIOS
BIOS Version          6.0
Point/OEM Versions    5.3
OEM Board Version     3
BIOS Build Date       MM/DD/YYYY
System BIOS Size      128KB
CPM/CSPM/BPM Modules  P7C7, 855, X689-VR7

Processor (CPU)
Intel® Pentium® M Processor 1.00GHz

System Memory (RAM)
Low Memory (KB)       628
Extended Memory (KB)  1035136

Real Time Clock (RTC)
RTC Date: [MM/DD/YYYY]
RTC Time: [HH:MM:SS]
Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

Use TAB to switch between
Month, day and year. .and
Hour, Minute and Second
Use digits and BKSP
To change field.

Figure 3-1 Main Setup Menu

BIOS Version	Indicates the major and minor core architecture versions (6.x, where x is a number from 0 to 999.)
BIOS Build Date	Date in MM/DD/YY format on which the OEM built the system BIOS binary file.
System BIOS Size	Size of BIOS exposed in low memory below the 1MB boundary. Commonly, 128KB would mean that the BIOS is visible in the address space from E000:0000 to F000:FFFF.
CPM/CSPM/BPM Modules	Indicates the names of the key architectural modules used to create the system BIOS binary file. The CPM module provides the CPU family support; the CSPM module provides the northbridge support; and the BPM module provides the board-level support.
Option	**Description**
Real Time Clock (*HH*:*MM*:*SS*)	Sets the real-time clock for hour (*HH*), minute (*MM*), and seconds (*SS*). The hour is calculated according to the 24 hour military clock (00:00:00 through 23:59:59). Use TAB or ENTER to move the cursor right, and SHIFT-TAB to move it left. Use the number keys, 0-9, to change the field values. It is not necessary to enter the seconds or type zeros in front of numbers.
Real Time Clock (*MM*/*DD*/*YYYY*)	Sets the real-time clock for the month (*MM*), day (*DD*), and year (*YYYY*). The valid values in this field are 01/01/1981 through 12/31/2099. Use TAB or to ENTER move the cursor right, and SHIFT-TAB to move it left. Use the number keys, 0-9, to change the field values. It is not necessary to type zeros in front of numbers.

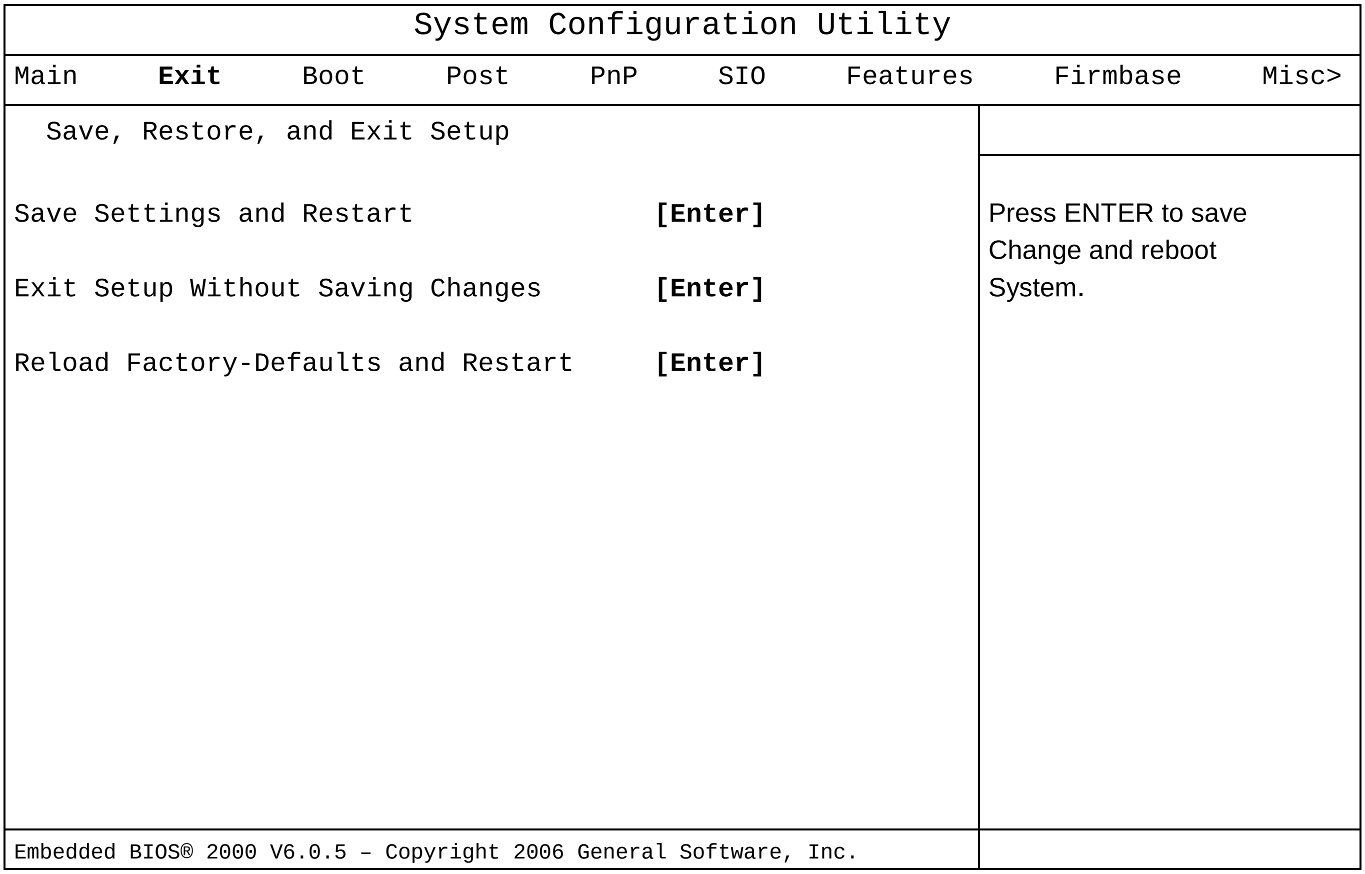
The CPU information is automatically obtained.

The system memory information does not describe physical RAM; rather it describes the RAM as configured, subtracting RAM used for System Management Mode, Shadowing, Video buffers, and other uses. This provides realistic values about how much memory is actually available to operating systems and applications.

The Real Time Clock fields are editable with keystrokes. To navigate through the MM/DD/YYYY and HH:MM:SS fields, use the TAB and BACKTAB keys. The hours are normally specified in military time; thus 13 means 1pm, or one hour after noon, whereas 01 means 1am, or one hour after midnight. When the cursor leaves RTC fields, they either affect the battery-backed RTC right away, allowing the system to continue with your new settings, or they revert back to old values if the new values are not valid entries.

3.2 Exit Menu

The Exit menu provides methods for saving changes made in other menus, discarding changes, or reloading the standard system settings.   This menu is shown in Figure 3.2 below.



```
                         System Configuration Utility
Main     Exit     Boot     Post     PnP     SIO     Features     Firmbase     Misc>

  Save, Restore, and Exit Setup

Save Settings and Restart                  [Enter]        Press ENTER to save
                                                          Change and reboot
Exit Setup Without Saving Changes          [Enter]        System.

Reload Factory-Defaults and Restart        [Enter]

Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

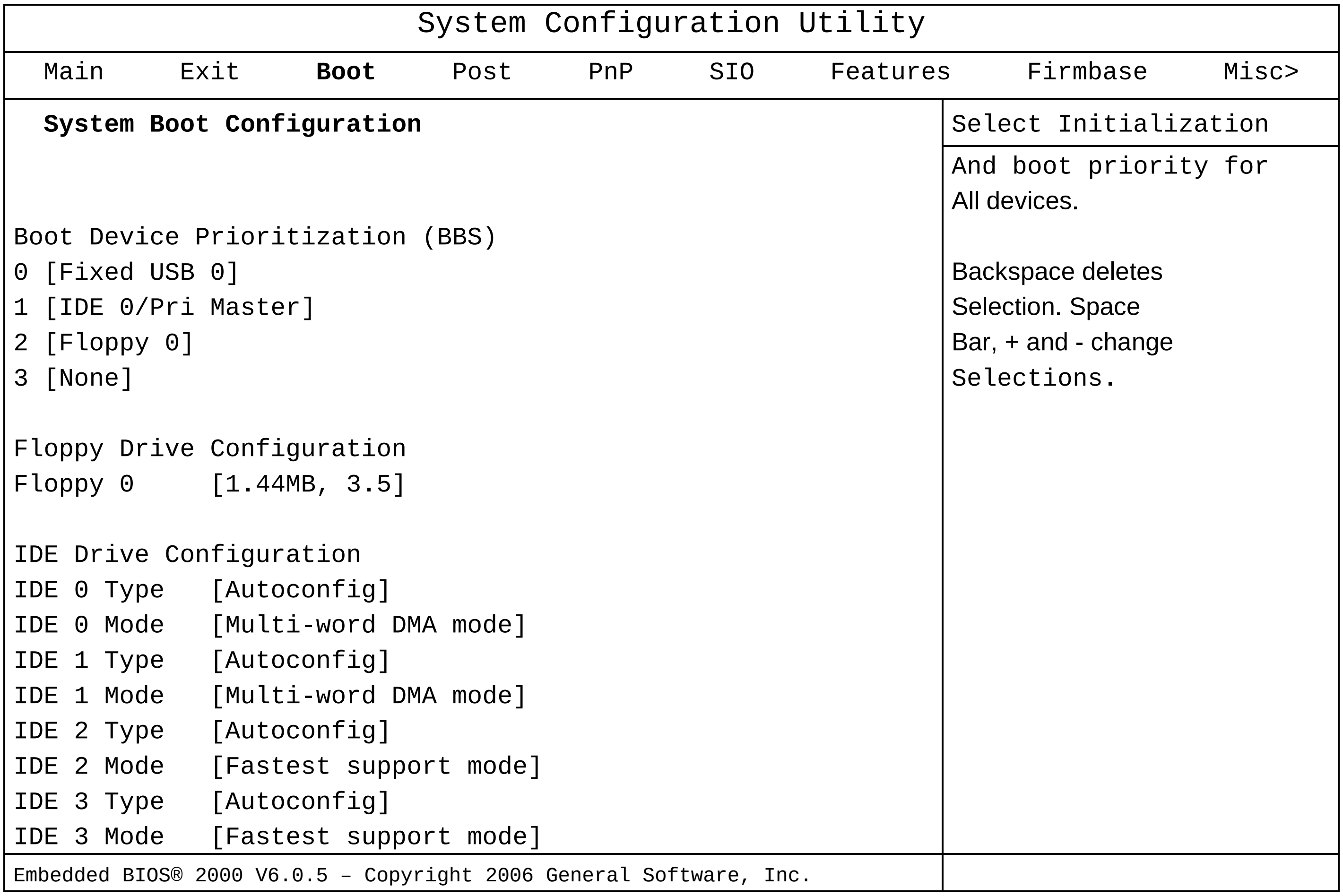
Figure 3-2 Save, Restore and Restart Menu

To select any of these options, position the cursor over the option and press the ENTER key. Pressing the ESC key at any time within the Setup system is equivalent to requesting “Exit Setup without Saving Changes.”

All three options request verification before performing the selected action, otherwise, the system configuration might be saved or lost by accident. Figure 3.3 illustrates the verification popup for saving and exiting; the other options are similar.

3.3 System Boot Menu

The Boot menu allows the system's boot actions and boot devices to be configured. This menu is shown in Figure 3.3.



```
                          System Configuration Utility
  Main     Exit     Boot     Post     PnP     SIO     Features     Firmbase     Misc>

  System Boot Configuration                                Select Initialization
                                                           And boot priority for
                                                           All devices.
Boot Device Prioritization (BBS)
0 [Fixed USB 0]                                            Backspace deletes
1 [IDE 0/Pri Master]                                       Selection. Space
2 [Floppy 0]                                               Bar, + and - change
3 [None]                                                   Selections.

Floppy Drive Configuration
Floppy 0     [1.44MB, 3.5]

IDE Drive Configuration
IDE 0 Type   [Autoconfig]
IDE 0 Mode   [Multi-word DMA mode]
IDE 1 Type   [Autoconfig]
IDE 1 Mode   [Multi-word DMA mode]
IDE 2 Type   [Autoconfig]
IDE 2 Mode   [Fastest support mode]
IDE 3 Type   [Autoconfig]
IDE 3 Mode   [Fastest support mode]
Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

Figure 3-3 System Boot Configuration Menu

When the BIOS completes POST, it follows this list, attempting to process each item. Some items are drives, such as an ATA/IDE drive, or a USB hard disk, or CDROM.

The ordering of the drives in the list the BIOS controls the process in several ways.

First, it is the list of drives that are scanned and assigned BIOS unit numbers for DOS (0, 1, 2 for floppy-type devices, and 80h, 81h, 83h, and so on for hard drives.)   If a drive on the list is not plugged in or working properly, the BIOS moves on to the next drive, skipping the inoperative device.

Second, once the drives in the list have been verified, POST attempts to boot from them in that order as well. Drives without bootable partitions might be configured, but skipped over in the boot phase, so that other drives on the list become candidates for booting the OS.

This list can also contain other boot actions, as boot from network ports. When deciding what boot action to do first and then next in succession, POST first scans all the drives in the list to verify they are present and operating properly (as described earlier in this section) and then goes down the list and tries to perform the actions in order. During this boot phase, if the list item is a drive, an attempt is made to boot from the boot record of that drive. If the list item is a device like a network PMC card, an attempt is made to boot from that device. If the list item is a non-bootable device, it moves on to the next item in the boot list.

The following table shows a list of boot devices and their configuration parameters.

Boot Device Prioritization (BBS) 0 [Fixed USB 0] 1 [IDE 0/Pri Master] 2 [Floppy 0] 3 [None]	This is the order in which the BIOS will look for the Operating System. Place the device to which you want to Boot at the top on the list. The device Names are loaded by the BIOS as it finds the hardware.
Floppy Drive Configuration	
Floppy 0 [1.44MB, 3.5]	Not Installed, 360KB 5,25”,1.2MB 5.25”, 720KB 3.5”, 1.44MB 3.5” and 2.88MB 3.5”
IDE Drive Configuration	
IDE 0 Type [Autoconfig]	AutoConfig, AutoConfig-Physical, AutoConfig-LBA, AutoConfig-Phoenix and Not-Installed
IDE 0 Mode [Multi-word DMA mode]	PIO-Mode, MULTI-Word-DMA-Mode, UDMA_Mode (40-Conductor Cable), UDMA_Mode (80-Conductor Cable), Fastest-Supported Mode
IDE 1 Type [Autoconfig]	AutoConfig, AutoConfig-Physical, AutoConfig-LBA, AutoConfig-Phoenix and Not-Installed
IDE 1 Mode [Multi-word DMA mode]	PIO-Mode, MULTI-Word-DMA-Mode, UDMA_Mode (40-Conductor Cable), UDMA_Mode (80-Conductor Cable), Fastest-Supported Mode
IDE 2 Type [Autoconfig]	AutoConfig, AutoConfig-Physical, AutoConfig-LBA, AutoConfig-Phoenix and Not-Installed
IDE 2 Mode [Fastest support mode]	PIO-Mode, MULTI-Word-DMA-Mode, UDMA_Mode (40-Conductor Cable), UDMA_Mode (80-Conductor Cable), Fastest-Supported Mode
IDE 3 Type [Autoconfig]	AutoConfig, AutoConfig-Physical, AutoConfig-LBA, AutoConfig-Phoenix and Not-Installed

IDE 3 Mode [Fastest support mode]	PIO-Mode, MULTI-Word-DMA-Mode, UDMA_Mode (40-Conductor Cable), UDMA_Mode (80-Conductor Cable), Fastest-Supported Mode
Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.	

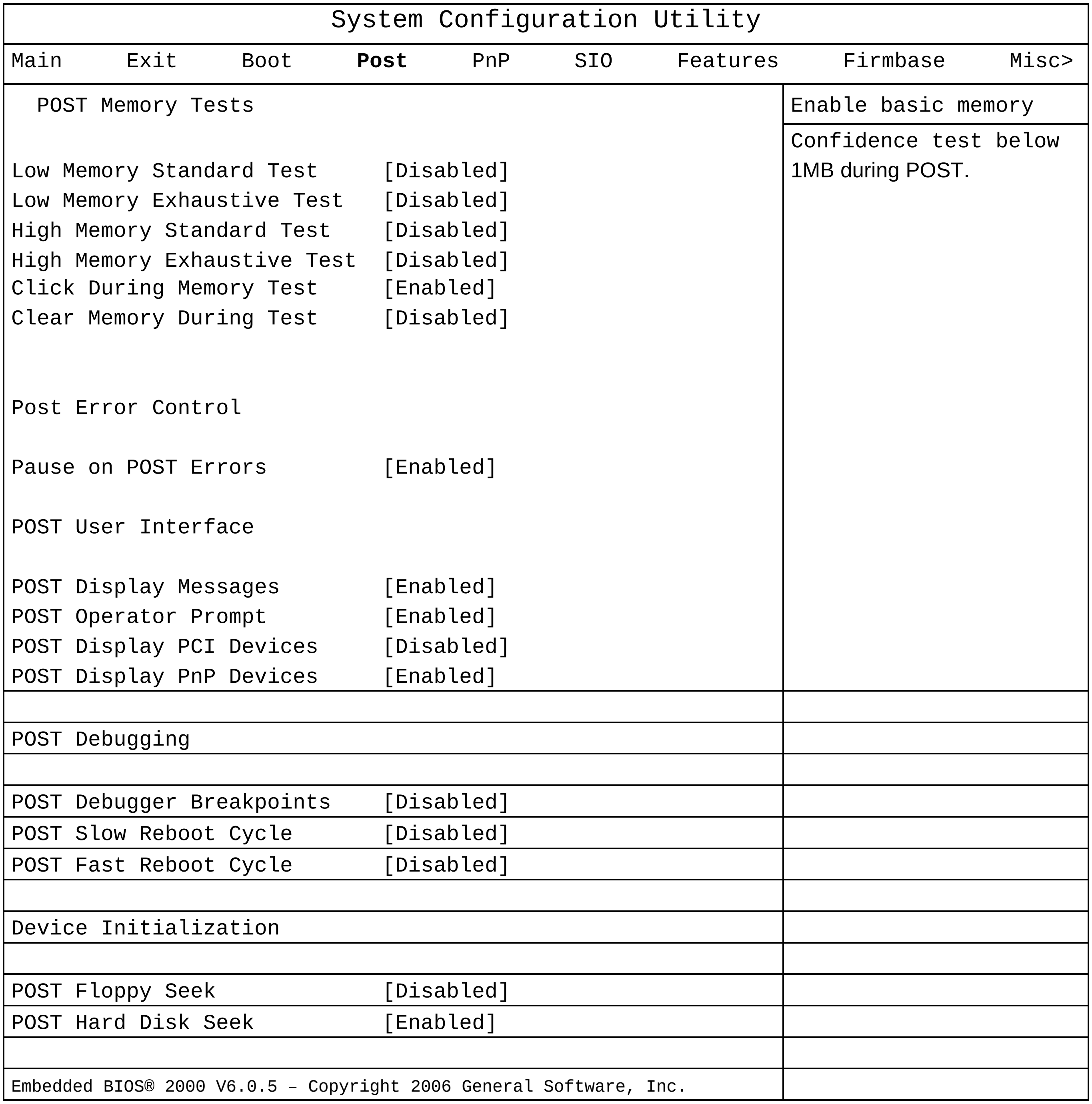
In addition to the boot device list, there are two more sections in the BOOT menu; namely, the Floppy Drive Configuration and IDE Drive Configuration sections. Both of these sections tell the BIOS what kind of equipment is connected to the motherboard, so that the BIOS can inspect the equipment. For example, the floppy drive section allows you to specify whether a floppy drive is a 5¼” 360KB floppy, a 5¼” 1.2MB floppy, a 3½” 720KB floppy, or a 3½” 1.44MB floppy (the first three are largely supplied for compatibility, since these floppy drives are no longer available in stores.)

Similarly, the IDE Drive Configuration section describes the type of hard drive equipment that is connected to the motherboard, including the cable type. IDE drives, or actually more properly Parallel ATA (PATA) drives, are connected to the motherboard with a flat cable with either 40 or 80 wires running in parallel (hence, Parallel ATA, as opposed to Serial ATA.) The 40-pin connector supports speeds up to UDMA2, whereas 80-pin cables are needed for higher transfer rates to eliminate noise. The BIOS can be told what type of cable is available, so that it knows whether higher transfer rates are allowed; or, it can be told to autodetect the cable type, in which case the drive and the motherboard must both support the hardware protocol used to autodetect the drive’s cable type.

IDE CABLE SETTINGS: PATA cable autodetection sometimes fails with older drives, so 40-pin is the default, to ensure data integrity. If higher performance is desired (and it normally is), you should change this setting to 80-pin or AUTO if you’re sure an 80-pin cable is installed.

3.4 POST Memory Tests

The POST menu is used to configure POST. This menu is shown in Figure 3.4 (scrolled down more so the full set of options can be seen.).



```
                      System Configuration Utility
Main     Exit     Boot     Post     PnP     SIO     Features     Firmbase     Misc>

  POST Memory Tests                                           Enable basic memory
                                                              Confidence test below
Low Memory Standard Test     [Disabled]                       1MB during POST.
Low Memory Exhaustive Test   [Disabled]
High Memory Standard Test    [Disabled]
High Memory Exhaustive Test  [Disabled]
Click During Memory Test     [Enabled]
Clear Memory During Test     [Disabled]


Post Error Control

Pause on POST Errors         [Enabled]

POST User Interface

POST Display Messages        [Enabled]
POST Operator Prompt         [Enabled]
POST Display PCI Devices     [Disabled]
POST Display PnP Devices     [Enabled]

POST Debugging

POST Debugger Breakpoints    [Disabled]
POST Slow Reboot Cycle       [Disabled]
POST Fast Reboot Cycle       [Disabled]

Device Initialization

POST Floppy Seek             [Disabled]
POST Hard Disk Seek          [Enabled]

Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

Figure 3-4 POST memory test Menu

The following table describes the settings associated with the POST setup menu's Memory Test section.

Low Memory Standard Test	Enable basic memory confidence test, of memory below 1MB address boundary (conventional memory, or memory normally used by DOS.)
Low Memory Exhaustive Test	Enable exhaustive memory confidence test of memory below 1MB address boundary.
High Memory Standard Test	Enable basic memory confidence test, of memory between 1MB and 4.2GB address boundaries (extended memory.)
High Memory Exhaustive Test	Enable exhaustive memory confidence test, of memory between 1MB and 4.2GB address boundaries.
Huge Memory Standard Test	Enable basic memory confidence test, of memory above 4.2GB address boundary (available using PAE technology.)
Huge Memory Exhaustive Test	Enable exhaustive memory confidence test, of memory above 4.2GB address boundary.
Click During Memory Test	Enable/disable speaker click when testing each block.
Clear Memory During Test	Enable storing 0's in all memory locations tested. Only necessary when some legacy DOS programs are run, as they may rely on cleared memory to operate properly.

The following table describes the settings associated with the POST setup menu's Error Control section:

Pause on POST Errors	Enable pause when errors are detected during POST, so that the user can view the error message and enter Setup or continue to boot the OS.

The following table describes the settings associated with the POST setup menu's POST User Interface section:

POST Display Messages	Enable display of text messages during POST. When disabled, POST is "quiet."
POST Operator Prompt	Enable operator prompts if POST is configured to ask interactive questions of the user about whether to load specific features; i.e., whether or not to load SMM.
POST Display PCI Devices	Enable display of PCI devices.
POST Display PnP Devices	Enable display of ISA PnP devices.

The following table describes the settings associated with the POST setup menu's Debugging section:

POST Debugger Breakpoints	Enable processing of INT 3 (breakpoint) instructions embedded into option ROMs. When enabled, if an INT 3 instruction is encountered, control is transferred to the BIOS debugger, so that the option ROM can be debugged. When disabled, these instructions perform no action.

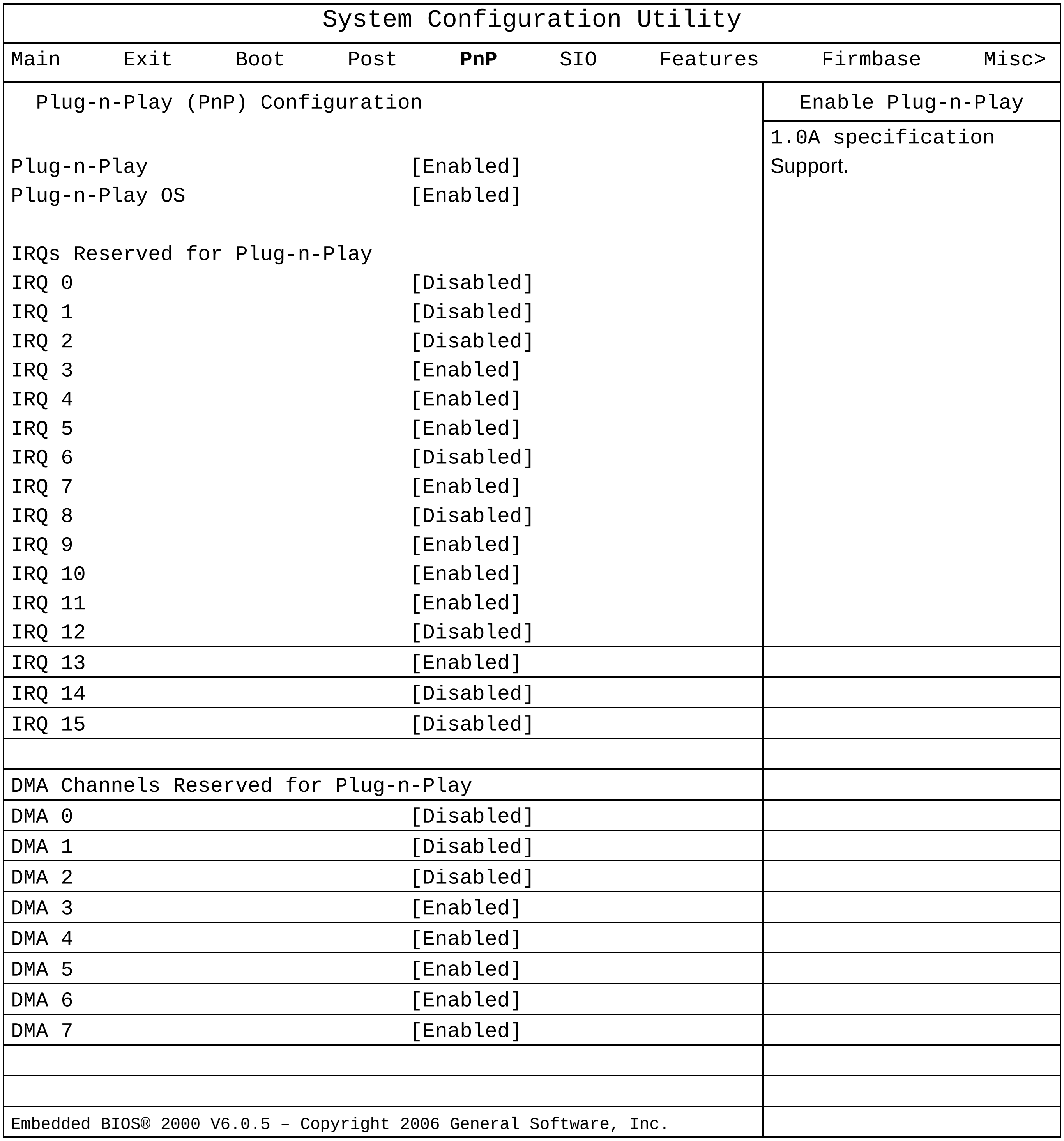
POST Fast Reboot Cycle	Enable early reboot in POST, allowing service technician to verify that the hardware can reboot very quickly many times in succession. Platform will continue to reboot after every boot until the system's CMOS is reset, as there is no way to enter Setup from this early point during POST.
POST Slow Reboot Cycle	Enable late reboot in POST, allowing service technician to cause the system to move through POST and then reboot, causing POST to be reexecuted, over and over, until Setup is reentered and this option is disabled. When left unattended, this is a straightforward way of having POST exercise system memory and peripherals without requiring a boot to a drive with an operating system installed.

The following table describes the settings associated with the POST setup menu's Device Initialization section:

POST Floppy Seek	Enable head seek on each floppy drive configured in the system. Used to recalibrate the drive in some systems with older DOS operating systems.
POST Hard Disk Seek	Enable head seek on each hard drive configured in the system. This is a way of extending the standard testing performed on each drive during POST, by requesting that the drive actually move the head. Not available with all drives.

3.5 Plug and Play Configuration Menu

The PnP menu is used to configure Plug-n-Play, a legacy BIOS initiative used to support operating systems such as Windows95, Windows98, and WindowsNT. ACPI has largely replaced this feature; however, it is necessary for platforms to support older operating systems. Figure 3.5 shows the PnP Setup menu.



```
                    System Configuration Utility
Main    Exit    Boot    Post    PnP    SIO    Features    Firmbase    Misc>

  Plug-n-Play (PnP) Configuration                  |  Enable Plug-n-Play
                                                   |  1.0A specification
Plug-n-Play                      [Enabled]         |  Support.
Plug-n-Play OS                   [Enabled]         |
                                                   |
IRQs Reserved for Plug-n-Play                      |
IRQ 0                            [Disabled]        |
IRQ 1                            [Disabled]        |
IRQ 2                            [Disabled]        |
IRQ 3                            [Enabled]         |
IRQ 4                            [Enabled]         |
IRQ 5                            [Enabled]         |
IRQ 6                            [Disabled]        |
IRQ 7                            [Enabled]         |
IRQ 8                            [Disabled]        |
IRQ 9                            [Enabled]         |
IRQ 10                           [Enabled]         |
IRQ 11                           [Enabled]         |
IRQ 12                           [Disabled]        |
IRQ 13                           [Enabled]         |
IRQ 14                           [Disabled]        |
IRQ 15                           [Disabled]        |
                                                   |
DMA Channels Reserved for Plug-n-Play              |
DMA 0                            [Disabled]        |
DMA 1                            [Disabled]        |
DMA 2                            [Disabled]        |
DMA 3                            [Enabled]         |
DMA 4                            [Enabled]         |
DMA 5                            [Enabled]         |
DMA 6                            [Enabled]         |
DMA 7                            [Enabled]         |
                                                   |
                                                   |
Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

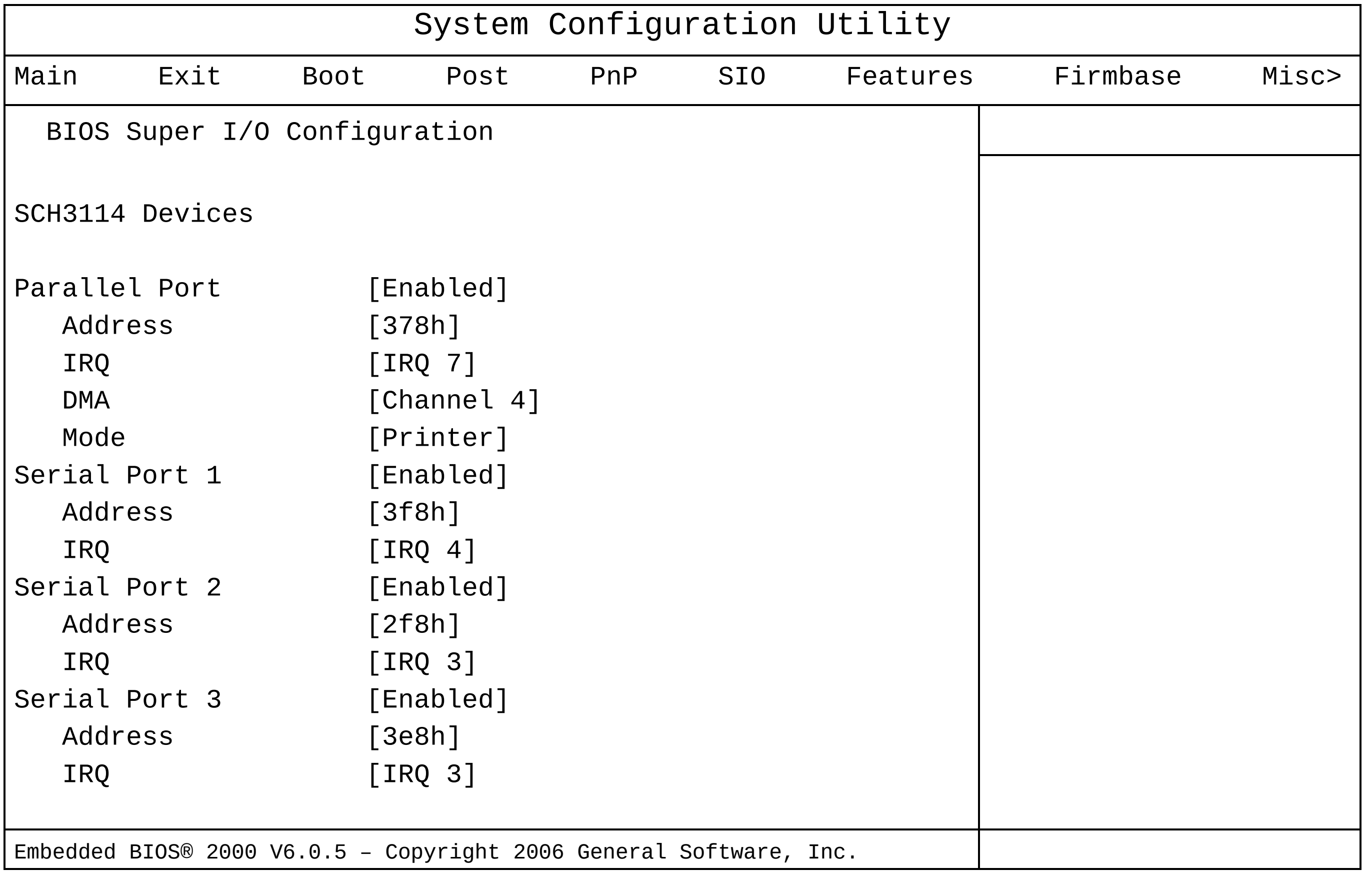
Figure 3-5 Plug-n-Play Configuration Menu

The PnP menu consists of two sections; basic configuration that enables Plug-n-Play and identifies if a PnP should perform configuration or let the OS do it; and then, another section that defines which system IRQs should be reserved for PnP's use, so that PCI doesn't use them. The following table presents the fields in the PnP menu.

Plug-n-Play	Enable PnP feature. When disabled, a PnP-aware OS will not find any PnP services in the BIOS, and all other configuration parameters in the menu will be greyed out. Enable to support legacy OSes like DOS, Windows95, Windows98, and WindowsNT. Disable for operating systems like WindowsXP or Windows Vista, or for Linux operating systems with ACPI support.
Plug-n-Play OS	Enable delay of configuration of PnP hardware and option ROMs. When enabled, BIOS will NOT configure the devices, and instead defer assignment of resources, such as DMA, I/O, memory, and IRQs, to the PnP OS. When disabled, the BIOS performs conflict detection and resolution, and assigns resources for the OS. Disable this parameter when running non-PnP OSes like DOS. Enable this parameter when running PnP OSes like Windows95, Windows98, and WindowsNT.
IRQ0	Enable exclusive use of IRQ0 by PnP.
IRQ1	Enable exclusive use of IRQ1 by PnP.
IRQ2	Enable exclusive use of IRQ2 by PnP.
IRQ3	Enable exclusive use of IRQ3 by PnP.
IRQ4	Enable exclusive use of IRQ4 by PnP.
IRQ5	Enable exclusive use of IRQ5 by PnP.
IRQ6	Enable exclusive use of IRQ6 by PnP.
IRQ7	Enable exclusive use of IRQ7 by PnP.
IRQ8	Enable exclusive use of IRQ8 by PnP.
IRQ9	Enable exclusive use of IRQ9 by PnP.
IRQ10	Enable exclusive use of IRQ10 by PnP.
IRQ11	Enable exclusive use of IRQ11 by PnP.
IRQ12	Enable exclusive use of IRQ12 by PnP.
IRQ13	Enable exclusive use of IRQ13 by PnP.
IRQ14	Enable exclusive use of IRQ14 by PnP.
IRQ15	Enable exclusive use of IRQ15 by PnP.

3.6 BIOS Super I/O Configuration Menu

The SIO menu is used to configure Super I/O components on the XVME 689-VR7/689. These components commonly are serial and parallel port controllers to floppy disk and keyboard controllers. The I/O, DMA, and IRQ assignments of each peripheral are configurable, so these values are also brought out to the SIO Setup menu. Figure 3.6 shows the SIO Setup menu for the XVME 689-VR7/689.



```
                     System Configuration Utility
Main     Exit     Boot     Post     PnP     SIO     Features     Firmbase     Misc>

  BIOS Super I/O Configuration

SCH3114 Devices

Parallel Port          [Enabled]
   Address             [378h]
   IRQ                 [IRQ 7]
   DMA                 [Channel 4]
   Mode                [Printer]
Serial Port 1          [Enabled]
   Address             [3f8h]
   IRQ                 [IRQ 4]
Serial Port 2          [Enabled]
   Address             [2f8h]
   IRQ                 [IRQ 3]
Serial Port 3          [Enabled]
   Address             [3e8h]
   IRQ                 [IRQ 3]

Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

Figure 3-6 BIOS Super I/O Configuration Menu

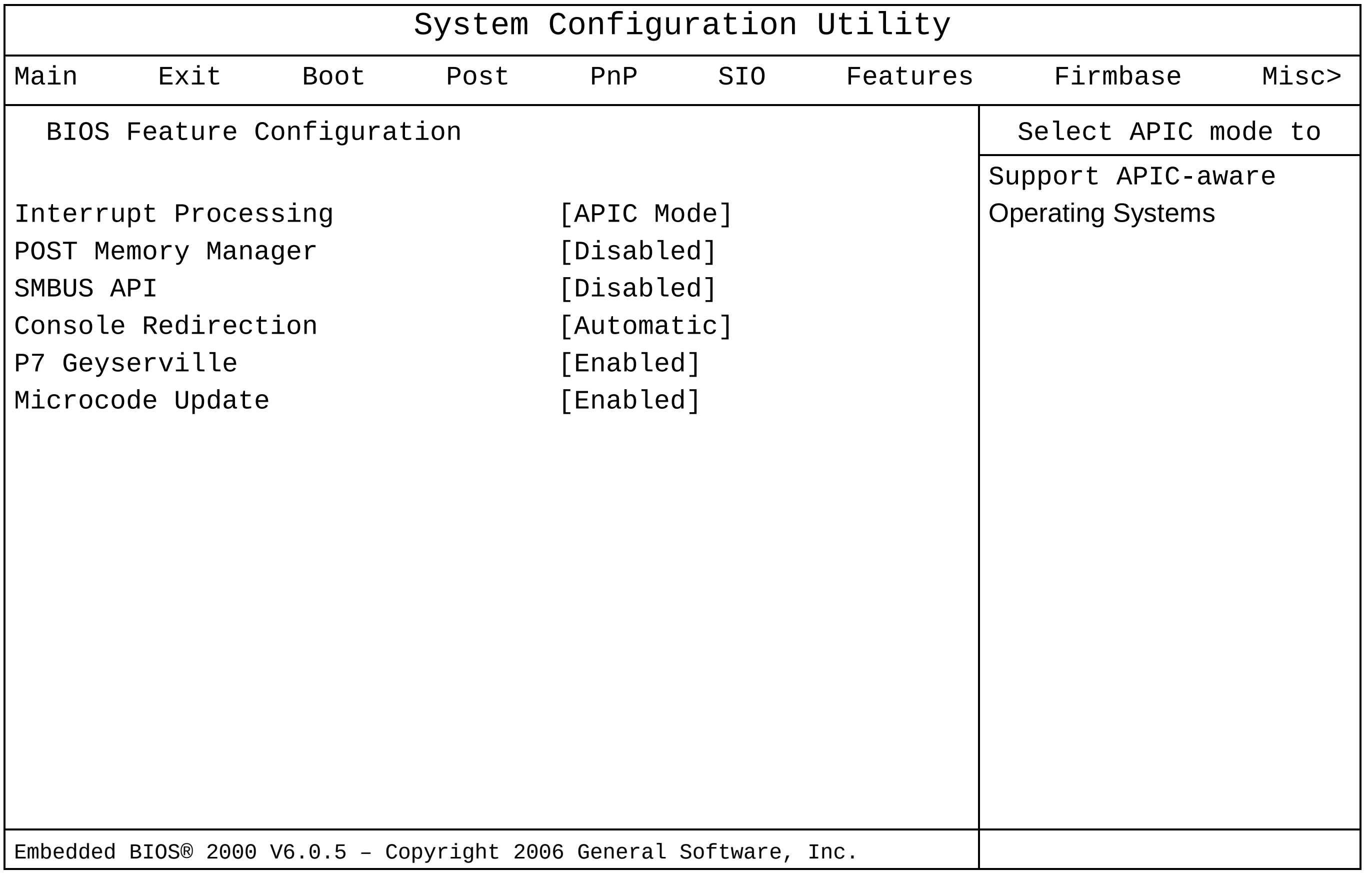
The following table provides the basic types of devices that might appear in a typical SIO menu:

Parallel Port	Enable parallel (LPT/printer) port. Legacy I/O addresses are 378h, 278h, and 3bch. IRQ7 was originally used on the IBM PC.
Serial Port	Enable serial (COM/communications) port. Legacy I/O addresses and IRQs are as follows: COM1 – I/O 3f8h, IRQ4. COM2 – I/O 2f8h, IRQ3. COM3 – I/O 3e8h, IRQ4. COM4 – I/O 2e8h, IRQ3. It should be noted that these are not the only possible addresses, but they are the ones that will ensure compatibility with the most legacy software,

	especially early DOS programs that do not use BIOS to access the COM ports.
Keyboard	Enable PC/AT or PS/2 keyboard controller.
Mouse	Enable PC/AT or PS/2 mouse portion of keyboard controller.
GPIO [device name]	Enable GPIO device.
ACPI [device name]	Enable ACPI device.

3.7 BIOS Super I/O Configuration Menu

The Features menu is used to configure the system BIOS' major features, including Quick Boot, APM, ACPI, PMM, SMBUS, SMBIOS, Manufacturing Mode, Splash Screen, Console Redirection, and others added by the OEM. Figure 3.7 shows a typical Setup.



```
                          System Configuration Utility
Main     Exit     Boot     Post     PnP     SIO     Features     Firmbase     Misc>

  BIOS Feature Configuration                                Select APIC mode to
                                                           Support APIC-aware
Interrupt Processing                  [APIC Mode]          Operating Systems
POST Memory Manager                   [Disabled]
SMBUS API                             [Disabled]
Console Redirection                   [Automatic]
P7 Geyserville                        [Enabled]
Microcode Update                      [Enabled]

Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

Figure 3.7 BIOS Feature Configuration

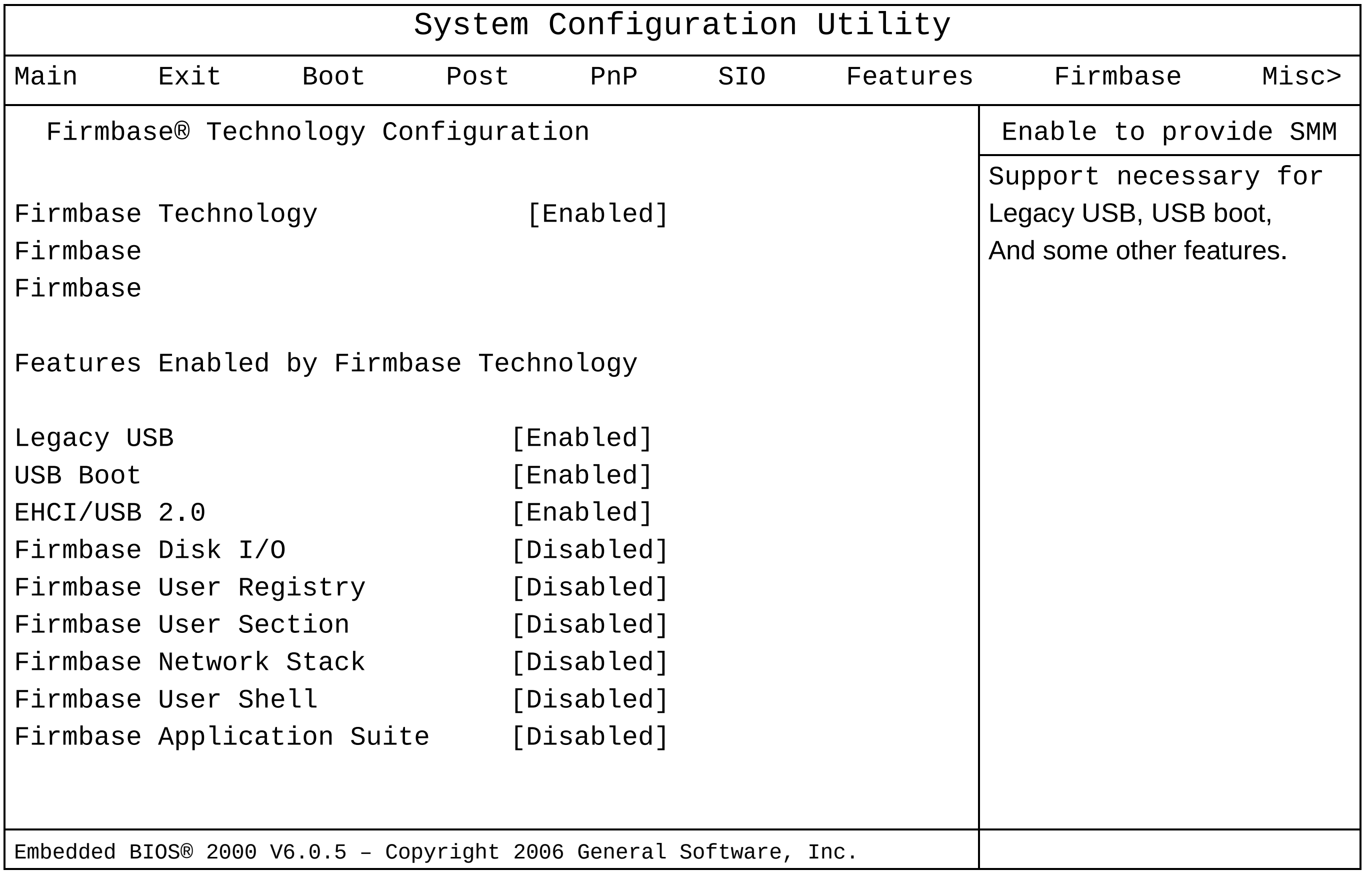
The following table describes each setting in the Features menu.

Quick Boot	Enable time-optimized POST, causing certain preconfigured OEM optimizations to be made when the system boots. Depending on the system, Quick Boot can reach the DOS prompt in as little as 85ms (milliseconds.)
Advanced Power Management (APM)	Enable legacy power management, used by the system when an ACPI-aware operating system is not running (during POST, such as when the system is running the preboot environment, or while running DOS, Windows95, Windows98, or Linux kernels below version 2.6.) Uses the SMM feature (see Firmbase) to operate properly.

ACPI	Enable ACPI system description and power management (ACPI replaces PnP and APM.) Used with ACPI-aware OSes such as Linux kernels version 2.6 and above, Windows XP, and Windows Vista. Commonly also uses the SMM feature (see Firmbase) to operate properly.
POST Memory Manager (PMM)	Enable memory allocation services for option ROMs, especially network cards running PXE. Some option ROMs may use this interface incorrectly, causing system crashes. Other PXE option ROMs may not run if PXE is not supported. Because of the state of these option ROMs, the setting is provided as an option to the user.
SMBUS API	Enable INT 15h services that permit certain software to access devices on the system's SMBUS without having knowledge of the SMBUS controller itself. Such devices include TV/radio tuners, volume controls, brightness and contrast controls, etc.
SMBIOS	Enable System Management BIOS interface specification support, exposing information about the type of hardware, including the chassis, motherboard layout, type of CPU and DRAM sticks, to applications such as WfM, which runs on PXE in the preboot environment.
Manufacturing Mode	Enable automatic entry into manufacturing mode when POST encounters a critical error. Used in closed device settings such as smart phones that need access to docking stations when they don't boot.
Splash Screen	Enable graphical POST, including animation, sound, icons, advertisements, and other multimedia objects that may be configured by the OEM.
Console Redirection	Configure the console redirection feature over a serial port. Automatic – causes POST, the debugger, and the preboot environment to use the system's first serial port (COM1) when an RS232 cable is detected with DSR and CTS modem signals active, indicating a terminal emulation program is likely to be attached ot the other end of the cable. Always – causes the BIOS to always use the serial port as the console, without testing for the presence of the terminal emulation program. Never – causes the BIOS to never invoke console redirection, but instead always use the main keyboard and video display. If there is no keyboard or video display, the system operates headless.

3.8 Firmbase® Technology Configuration

This menu is highly configurable by the OEM who may elect to eliminate some of the Firmbase Technology tuning parameters in more fixed-function devices. To illustrate what all of the standard Firmbase Technology configuration parameters are.



```
                      System Configuration Utility
Main     Exit     Boot     Post     PnP     SIO     Features     Firmbase     Misc>

  Firmbase® Technology Configuration                         Enable to provide SMM
                                                             Support necessary for
Firmbase Technology             [Enabled]                    Legacy USB, USB boot,
Firmbase                                                     And some other features.
Firmbase

Features Enabled by Firmbase Technology

Legacy USB                     [Enabled]
USB Boot                       [Enabled]
EHCI/USB 2.0                   [Enabled]
Firmbase Disk I/O              [Disabled]
Firmbase User Registry         [Disabled]
Firmbase User Section          [Disabled]
Firmbase Network Stack         [Disabled]
Firmbase User Shell            [Disabled]
Firmbase Application Suite     [Disabled]

Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

Figure 3.8 Firmbase® Technology Configuration

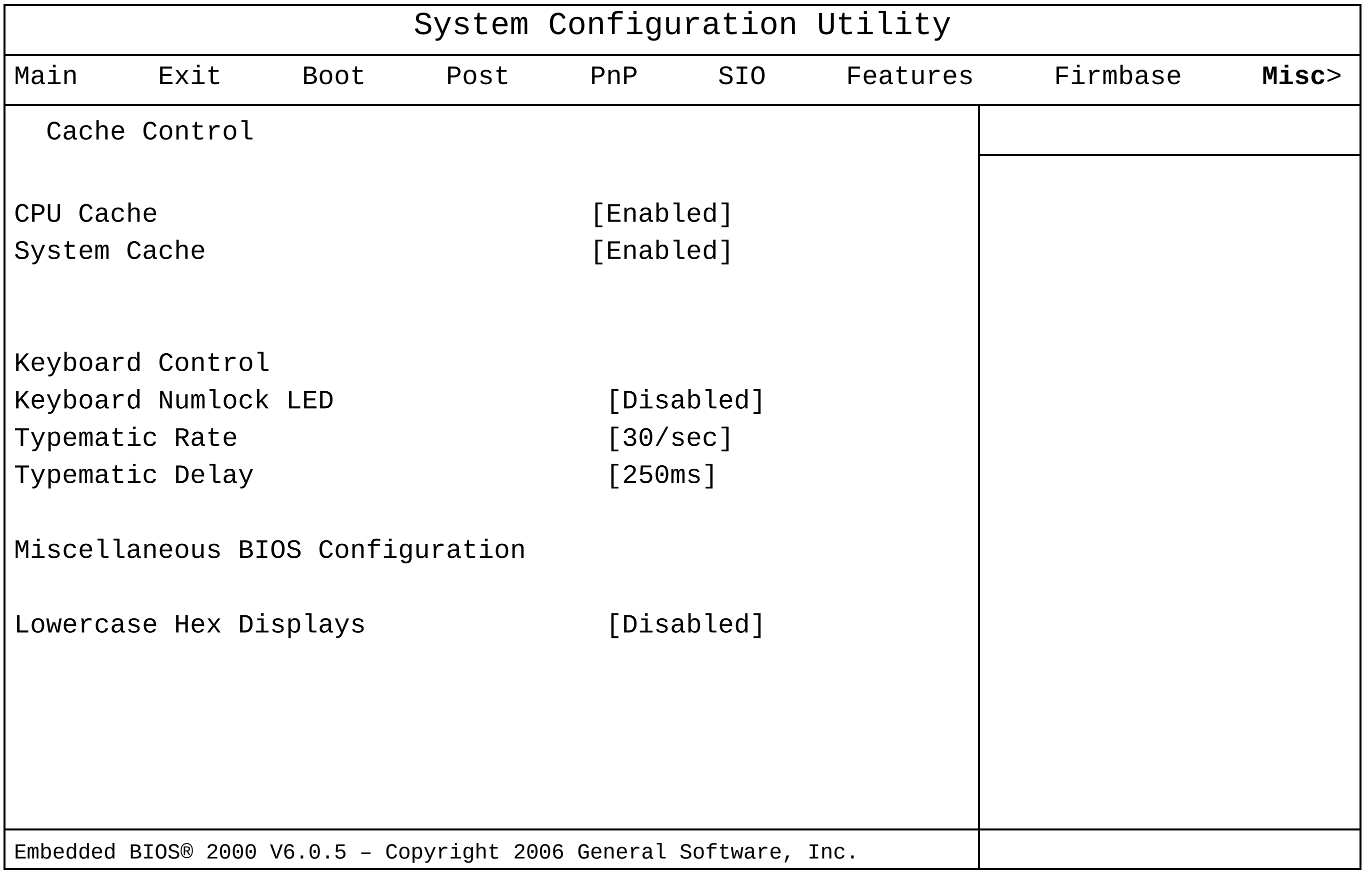
The following table presents the settings that enable high-level features enabled by Firmbase Technology.

Legacy USB	Enables BIOS support for USB keyboards and mice. Up to 8 USB keyboards and 8 USB mice may be supported at a time. Use of PS/2 keyboard and mouse concurrently with USB devices is discouraged, as the legacy PS/2 keyboard controller cannot easily separate simultaneous data streams from both device classes.
USB Boot	Enables BIOS support for accessing USB mass storage devices and emulating legacy floppy, hard drive, and CDROM drive devices with them.

	Enable this option in order for USB devices to be supported in the BBS device list (see the BOOT menu.)
EHCI/USB 2.0	Enables EHCI Firmbase Technology driver, allowing USB Boot feature to use high speed transfers on USB 2.0 ports in the system.
Firmbase Disk I/O	Enables Firmbase Technology FAT file system driver, so that Firmbase applications such as Boot Security, Platform Update Facility, and HA Monitor, as well as the HA and TCB components of the kerne, have access to files residing on drives containing FAT file systems. Also turn on this option if you wish to run Firmbase applications from FAT file systems on either ATA or USB mass storage devices.
Firmbase User Registry	Enables execution of the [USER] Firmbase registry section, which is preconfigured by the OEM. This may run OEM-specific applications.
Firmbase Network Stack	Enables Firmbase Technology internet protocol stack, including MAC driver for the platform, TCP, UDP, as well as presentation-level drivers configured by the OEM, which may include SNMP, SMTP, TELNETD, and HTTP server.
Firmbase Desktop	Enables Firmbase Technology graphical shell.
Firmbase User Shell	Enables Firmbase Technology command line interpreter, a multi-user command shell with DOS-like and Unix-like command structure; can be used to start Firmbase applications written with the Firmbase SDK, a General Software product.
Firmware Application Suite	Enables Firmbase applications configured for the system by the OEM. Typically includes Boot Security, Platform Update Facility, and High Availability Monitor.
Firmbase Technology	Enables Firmbase Technology as a whole, the industry's most comprehensive and full-featured System Management Mode (SMM) operating environment. Some hardware platforms require Firmbase Technology to run, as they may use it to virtualize hardware such as virtual video and audio PCI devices. Some BIOS features, such as ACPI and APM, may require Firmbase Technology to operate.

3.9 Misc. Menu

The Misc menu provides for configuration of BIOS settings that don't easily fit in any other category. They include Cache Control, Keyboard Control, Debugger Settings, and System Monitor Utility Configuration parameters. Figure 3.9 shows the Misc Setup men



```
                          System Configuration Utility
Main     Exit     Boot     Post     PnP     SIO     Features     Firmbase     Misc>

  Cache Control

CPU Cache                                    [Enabled]
System Cache                                 [Enabled]


Keyboard Control
Keyboard Numlock LED                          [Disabled]
Typematic Rate                                [30/sec]
Typematic Delay                               [250ms]

Miscellaneous BIOS Configuration

Lowercase Hex Displays                        [Disabled]

Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

Figure 3.9 Misc. Setup Menu

The following table presents the settings in the Misc Setup menu.

System Cache	Enables POST's support for cache in the system. Modern processors virtually require cache to be enabled to achieve acceptable performance. However, to diagnose certain problems related to caching in the system, such as multiprocessing systems, it may be desirable to disable this setting.
Keyboard Numlock LED	Enables the Numlock key when POST initializes the PS/2 keyboard.
Typematic Rate	Specify the rate at which the PS/2 keyboard controller repeats characters when most keys are pressed down. USB typematic is automatic and does not use this parameter.
Typematic Delay	Specifies the amount of time a repeating key may be pressed on a PS/2 keyboard until the key repeat feature begins repeating the keystroke. USB typematic is automatic and does not use this parameter.
Lowercase Hex Displays	Enables the display of hexadecimal numbers in the debugger with lowercase letters instead of

	uppercase letters (ie, 2f8ah instead of 2F8AH.)
Proprietary Stimulation	Enables System Monitor's callout to the OEM's BPM adaptation code to execute code that causes stimulation of the SMM environment for measurement purposes.
Hard Disk Read Stimulation	Enables System Monitor's read of a preconfigured number of sectors from a location on the first hard disk in the system in order to stimulate the SMM environment. This is useful when measuring code path lengths in USB boot, when the first hard drive is configured in the BBS list as a USB hard drive.
Hard Disk Write Stimulation	Enables System Monitor's write of a preconfigured number of sectors to a location on the first hard disk in the system in order to stimulate the SMM environment. This is useful when measuring code path lengths in USB boot, when the first hard drive is configured in the BBS list as a USB hard drive. Please note that when this parameter is selected, the system automatically enables reading, so that the stimulation of the system includes reading a range of sectors into a memory buffer, and writing the same data back to the same range of sectors for safety. Thus, this feature is theoretically nondestructive. WARNING: BECAUSE THE OEM'S ADAPTATION AND OTHER FACTORS ARE BEYOND GENERAL'S CONTROL, YOU ARE ADVISED THAT THIS FEATURE COULD CAUSE DATA LOSS AT YOUR SOLE EXPENSE; ACCORDINGLY, IT IS PROVIDED AS-IS WITHOUT WARRANTY OF ANY KIND. ALWAYS BACKUP YOUR DATA BEFORE PERFORMING DIAGNOSTICS ON ANY SYSTEM, AS THEY COULD CAUSE DATA LOSS.
Floppy Disk Read Stimulation	Enables System Monitor's read of a preconfigured number of sectors from a location on the first floppy disk in the system in order to stimulate the SMM environment. This is useful when measuring code path lengths in USB boot, when the first floppy drive is configured in the BBS list as a USB floppy drive.
Floppy Disk Write Stimulation	Enables System Monitor's write of a preconfigured number of sectors to a location on the first floppy disk in the system in order to stimulate the SMM environment. This is useful when measuring code path lengths in USB boot, when the first floppy drive is configured in the BBS list as a USB floppy

	drive. Please note that when this parameter is selected, the system automatically enables reading, so that the stimulation of the system includes reading a range of sectors into a memory buffer, and writing the same data back to the same range of sectors for safety. Thus, this feature is theoretically nondestructive. WARNING: BECAUSE THE OEM’S ADAPTATION AND OTHER FACTORS ARE BEYOND GENERAL’S CONTROL, YOU ARE ADVISED THAT THIS FEATURE COULD CAUSE DATA LOSS AT YOUR SOLE EXPENSE; ACCORDINGLY, IT IS PROVIDED AS-IS WITHOUT WARRANTY OF ANY KIND. ALWAYS BACKUP YOUR DATA BEFORE PERFORMING DIAGNOSTICS ON ANY SYSTEM, AS THEY COULD CAUSE DATA LOSS.

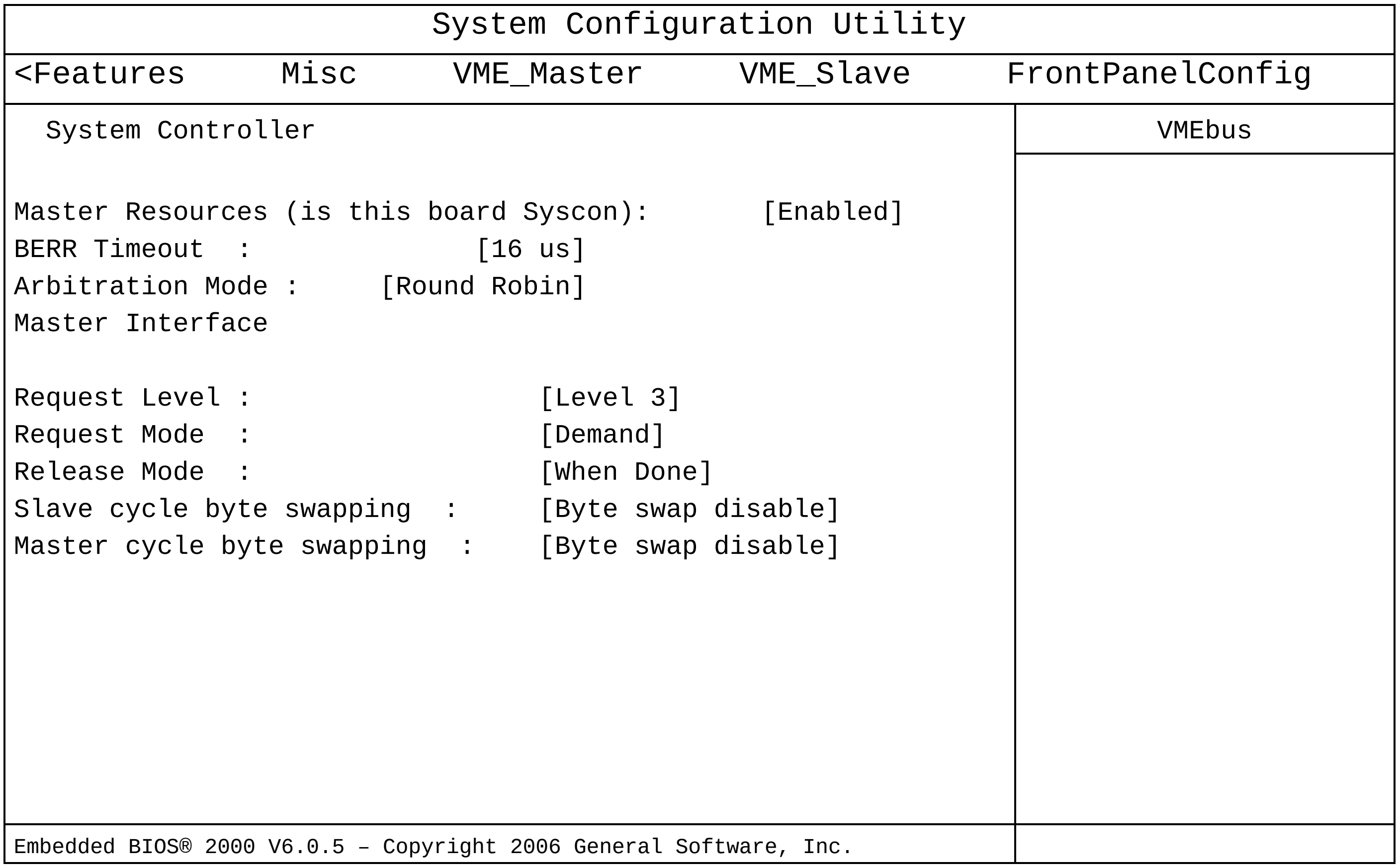
3.10 VMEbus Master

System Controller Submenu

The XVME 689-VR7 automatically provides slot 1 system resource functions. The system resource functions are explained in the Universe manual. (Contact Tundra at *www.tundra.com* for a PDF version of the Universe manual.) This function can be disabled using XVME 689-VR7 jumper J3. Refer to **Jumper Settings** in Chapter 2 (p. 2-2) for more information. System resources are VMEbus Arbiter, BERR timeout, SYSCLK, and IACK daisy chain driver. These resources must be provided by the module installed in the system controller slot (left most slot). The status of the XVME 689-VR7 system resources is reported in a read-only field.

Note

The BERR timeout is the VMEbus error timeout value.



```
                         System Configuration Utility
<Features     Misc     VME_Master     VME_Slave     FrontPanelConfig

  System Controller                                            VMEbus

Master Resources (is this board Syscon):       [Enabled]
BERR Timeout  :              [16 us]
Arbitration Mode :     [Round Robin]
Master Interface

Request Level :                  [Level 3]
Request Mode  :                  [Demand]
Release Mode  :                  [When Done]
Slave cycle byte swapping  :     [Byte swap disable]
Master cycle byte swapping  :    [Byte swap disable]

Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

Figure 3.10 System Controller VMEbus Master setup

Option	Description
System Resources	This read-only field displays the status (*Enabled* or *Disabled*) of the XVME-661 system resources. This value is automatically detected.
BERR Timeout*	This field is used to set the VMEbus error timeout. Choices are *16µs*, *32µs*, *64µs* (default), *128µs*, *256µs*, *512µs*, *1024µs*, and *Disabled*.
Arbitration Mode*	This field is used to set the VMEbus arbitration mode. Choices are *Priority/Single* (default) or *Round Robin*.

*Note

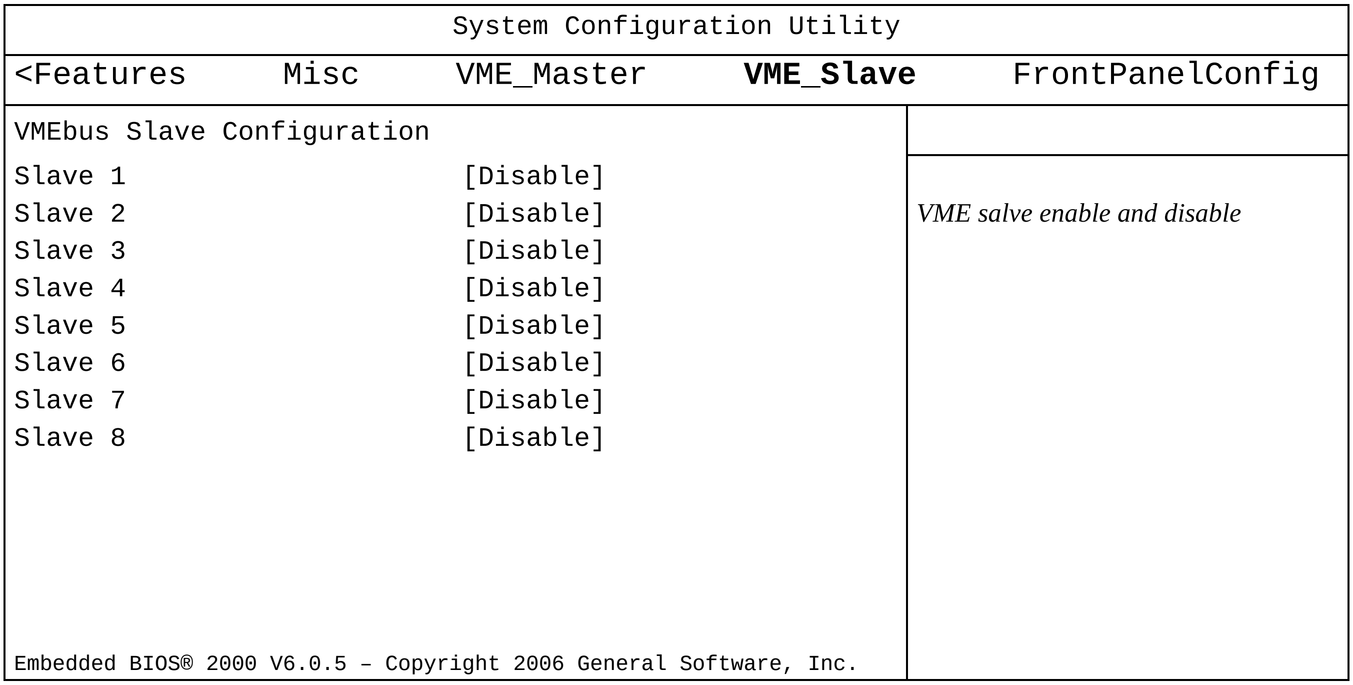
These fields are only referenced if the board is the system controller. If it is not, the setup field values are ignored, **BERR Timeout** is set to *Disabled* (0), and **Arbitration Mode** is set to *Round Robin*, with an Arbitration timeout value of 0 (*Disabled*).

3.11 VMEbus Slave Configuration

The VMEbus slave setup allows configuration of the XVME processor board's VMEbus slave interfaces.

Note

When the Slave 1 & 2 Operational Mode setting is *Compatible*, slave images 0 and 1 are reserved for BIOS use. See p. **Error! Bookmark not defined.** for more details.



```
VMEbus Slave Configuration
Slave 1                     [Disable]
Slave 2                     [Disable]
Slave 3                     [Disable]
Slave 4                     [Disable]
Slave 5                     [Disable]
Slave 6                     [Disable]
Slave 7                     [Disable]
Slave 8                     [Disable]
```

VME salve enable and disable

Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.

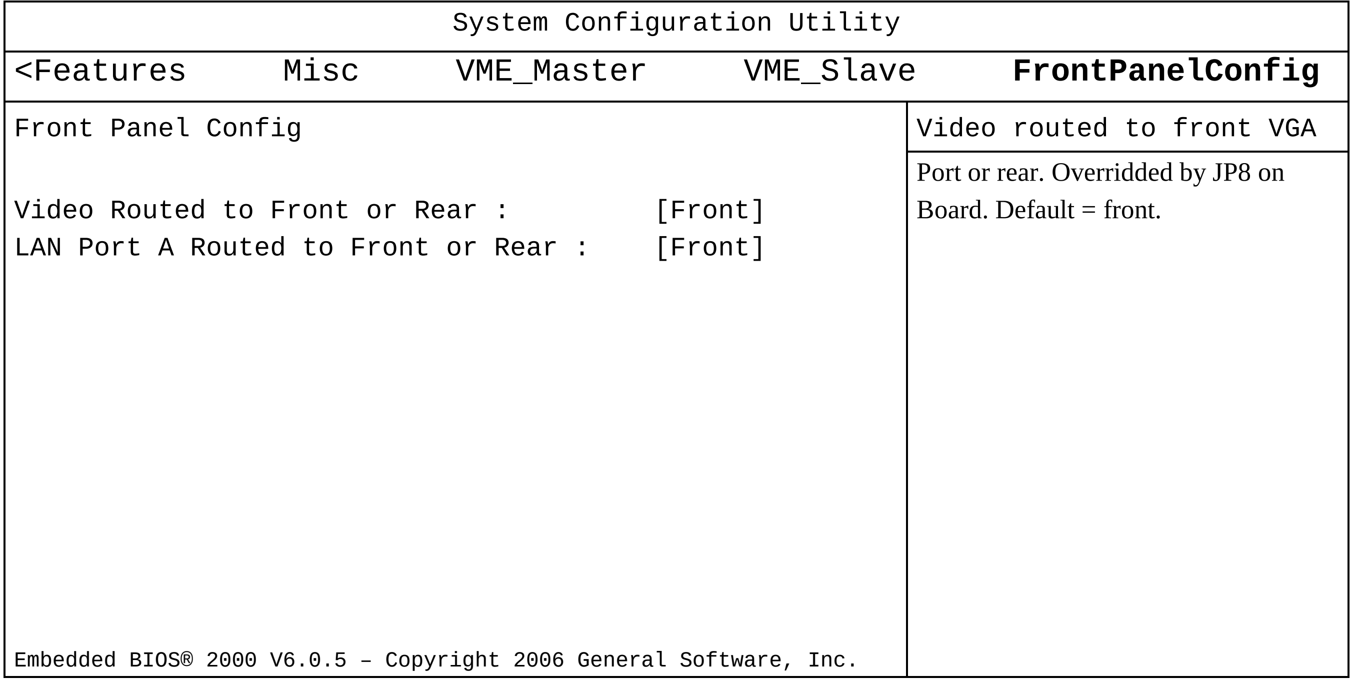
Figure 3-2 Slave Interface Submenu

Table 3-1 Slave Interface Submenu

Option	**Description**
Slave Interface	Used to turn the slave interface boot state *On* or *Off* (default). When turned *Off*, other VME masters cannot access memory on the XVME 689-VR7.
Address Modifiers	Determines which type of VMEbus slave access is permitted to read or write to the XVME 689-VR7 dual-access memory. The first field determines whether the slave interface responds to *Data* access only (default), *Program* access only, or *Both*. The second field determines whether the slave interface responds to *Supervisory* access only, *Non-Privileged* access only (default), or *Both*.
Address Space	Determines if VME masters access the slave's dual-access memory in the *VMEbus Standard* (A24) or VMEbus Extended (A32) address space. The default is *VMEbus Extended*.
Size	Determines the amount of dual-access memory that is available to external VMEbus masters. The slave memory size cannot be more than the total memory size, or greater than 16 MB for *VMEbus Standard* Address Space. The choices are *1MB* (default), *2MB*, *4MB*, *8MB*, *16MB*, and *32MB* (unavailable for *VMEbus Standard* Address Space).
Base Address High Nibble Base Address Med. Nibble Base Address Low Nibble	These fields determine the base VMEbus address prefix for the first 12 bits of the address to which the VMEbus slave interface will respond. The three fields are the high (H), middle (M), and low (L) nibbles of these 12 bits. The address is HML00000h. In the default screen configuration H is *A*, M is *A*, and L is *4*, so the address is AA400000h. The values change depending on the Size and Address Space field values. When the Address Space value is *VMEbus Standard*, the dual-access memory must be located on a 1 MB boundary and the upper two nibbles are ignored, so the high and medium nibbles are changed to *0* and are made read-only. When the Address Space value is *VMEbus Extended*, the slave address must be a multiple of the slave memory size. When the Size is greater than 1 MB, the low nibble is truncated to an even value. Note: The address that is set with these fields is the address that is used by the VMEbus processors. The PC/AT processor on the XVME 689-VR7 will see a translated address. This translation (and the amount of translation) is calculated by the BIOS and is not user-configurable in the BIOS setup. See p. 4-5 for a discussion of translation addresses.

3.12 Front Panel resources control

Use this menu to select front panel or rear I/O for the Video and the two Ethernet ports.



```
                         System Configuration Utility
<Features     Misc     VME_Master     VME_Slave     FrontPanelConfig

Front Panel Config                                    Video routed to front VGA
                                                      Port or rear. Overridded by JP8 on
Video Routed to Front or Rear :          [Front]      Board. Default = front.
LAN Port A Routed to Front or Rear :     [Front]

Embedded BIOS® 2000 V6.0.5 – Copyright 2006 General Software, Inc.
```

Chapter 4 Programming

Memory Map

The preliminary memory map of the XVME 689-VR7 as seen by the CPU is shown below. The I/O designation refers to memory which is viewed as part of the AT bus or as part of VMEbus depending on how the Universe is programmed.

XVME 689-VR7 MEMORY MAP

Table 4-1 Memory Map

ADDRESS RANGE (HEX)	SIZE	DEVICE
FFFC0000 - FFFFFFFF	256K	SYSTEM BIOS
end of DRAM -FFFBFFFF	xxxK	I/O MEMORY**
00100000 – end of DRAM	xxxK	DRAM *
000F0000 – 000FFFFF	64K	SYSTEM BIOS
000E0000 – 000EFFFF	64K	SYSTEM BIOS
000D8000 – 000DFFFF	32K	Universe Real Mode Window
000D0000 – 000D7FFF	32K	Open memory block
000CC000 – 000CFFFF	16K	Open memory block
000C0000 – 000C7FFF	32K	VGA BIOS
000A0000 – 000BFFFF	128K	VGA DRAM MEMORY
00000000 – 0009FFFF	640K	DRAM

*See Intel 6300ESB data sheet for a description for optional settings for setting memory holes or gaps within Memory map area

**The PCI devices are located at the very top of memory just below the system BIOS.

I/O Map

This Preliminary I/O map for the XVME 689-VR7 contains I/O ports of the IBM AT architecture plus some additions for PCI I/O registers and Xembedded specific I/O registers.

Hex Range	Device
000-01F	DMA controller 1, 8237A-5 equivalent
020-021	Interrupt controller 1, 8259 equivalent
022-023	Available
025-02F	Interrupt controller 1, 8259 equivalent (note 3)
040-05F	Timer, 8254-2 equivalent
060-06F	8742 equivalent (keyboard)
070-07F	Real Time Clock bit 7 NMI mask (note 3)
080-091	DMA page register (note 3)
92	Fast GateA20 and Fast CPU Init
93-9F	DMA page register (note 3)
0A0-0BF	Interrupt controller 2, 8259 equivalent (note 3)

0C0-0DF	DMA controller 2, 8237A-5 equivalent (note 3)
0F0	N/A
0F1	N/A
0F2-0FF	N/A
170-177	Secondary IDE Controller (Generates CS1*)
1F0-1F7	Primary IDE Controller (Generates CS1*)
219	Xembedded LED control register
234	Byte Swap port
235-277	Available
278-27F	Parallel Port 2 (note 1)
280-2F7	Available
2F8-2FF	Serial Port 2 (note 1)
300-36F	Available
376	Secondary IDE Controller (Generates CS3*)
378-37F	Parallel Port 1 (note 1)
380-3BF	Available
3C0-3DF	VGA/EGA2 (note 2)
3E0-3EF	Available
3F0-3F5	Primary Floppy disk controller
3F6	Primary IDE Controller (Generates CS3*)
3F8-3FF	Serial port 1 (note 1)
400-47F	Industry Pack (IP) I/O
480-4BF	Industry Pack (IP) ID
4D0h	ELCR1 (Edge or level triggered)
4D1h	ELCR2 (Edge or level triggered)
CF8	PCI configuration address register (note 4)
CF9	Reset Control Register
CFC	PCI configuration data register (note 4)

Note 1:The serial and parallel port addresses may be changed or the port may be disabled. Therefore these address maybe used for some applications and not for others.

Note 2: Reference the Intel 855GME datasheet for detailed information.

Note 3: Reference the Intel 6300ESB datasheet for detailed information

Note 4: Reference "The PCI local bus specification rev 2.3", 6300ESB datasheet for PCI configuration information.

IRQ Map

Table 4-2 IRQ Map

INT#	Function
IRQ0	System Timer
IRQ1	Keyboard
IRQ2	Interrupt Cascade (reserved)
IRQ3	COM2
IRQ4	COM1
IRQ5	Ethernet 1
IRQ5	PCI Expansion to PMC 2
IRQ6	Floppy
IRQ7	Parallel Port (LPT1)
IRQ8	Real Time Clock
IRQ9	Universe IID
IRQ9	PCI-X Video
IRQ10	Onboard PMC-X
IRQ11	PCI Expansion to PMC 1
IRQ11	Ethernet 2
IRQ12	Mouse
IRQ13	Math Coprocessor (reserved)
IRQ14	Primary IDE
IRQ15	Secondary IDE

The above interrupt mapping is one possible scenario. The user or operating system may choose a different mapping for some of these interrupts based on what devices are actually in the system and require interrupts. If COM2 or LPT1 are not used, then these would free up IRQ3 and IRQ7 respectively.

PCI Device Map –

Table 4-3 PCI device Map

Device	ID MFG		CF8 DWORD	R/G	INT# A B C D	AD Line	Device
82546GB (Ethernet1)	1079 8086	Ethernet Controller Function 0	8000 8000	PCI-X R/G 0	PX_IRQ#0	AD(20)	09H
82546GB (Ethernet2)	1079 8086	Ethernet Controller Function 1	8000 8000	PCI-X R/G 0	PX_IRQ#1	AD(20)	09H
PMC	–	XVME 689-VR7 PMC-X site	8007 8000	PCI R/G 1	3 0 1 2	AD(26)	0FH
855GME	3580 8086 3584 8086 3885 8086	Host-Hub interface, DDR SDRAM I/F, Legacy control Device #0 Host-to-AGP Bridge (Virtual PCI-to-PCI) Device #1 (Intel 855GME GMCH Only) Integrated Graphics Controller (IGD) Device #2	0000 0000				
6300ESB	244e 25A1 25A2 25A9 25AD 25A4 25A6 25A7 25AB 25AC 25AE 25A3 or 25B0 8086	Hub to PCI bridge LPC Interface IDE Controller USB UHCI Controller USB EHCI Controller SMBus Controller AC’97 Audio AC’97 Modem Watchdog Timer APIC1 HUB to PCI-X Bridge Serial ATA Controller					
Universe IID	0000 10E3	PCI/VMEbus Bridge	8000 5000	PCI R/G 0	A B C D	AD(21)	0AH
PMC 1	–	70976-201	8000 A000	R/G 2	B C D A	AD(31)	14H
PMC 1	–	70976-203	8000 A000	R/G 2	B C D A	AD(31)	14H
PMC 2	–	70976-203	8000 9000	R/G 3	D A B C	AD(29)	12H
PCI Card	–	70976-205	8000 A000	R/G 2	B C D A	AD(31)	14H

VME Interface

The VME interface is the Tundra Universe IID chip, which is a PCI bus-to-VMEbus bridge device. The XVME 689-VR7 implements a 32-bit PCI bus and a 32/64-bit VMEbus interface. The Universe chip configuration registers are located in a 4 KB block of PCI memory space. This memory location is programmable and defined by PCI configuration cycles. The VMEbus controller has four main functions; System Resources or the "traffic cop of the bus" Master interface which "starts conversation on the bus", Slave interface which responds to a bus master's question, and the interrupt functions which uses seven (7) levels of interrupt control.

Note

For your frame of reference, the left side below is the XVME 689-VR7 board and the right side below is the VMEbus.

PCI memory slave access = VMEbus master access

PCI memory master access = VMEbus slave access

System Resources

The XVME 689-VR7 automatically provides slot 1 system resource functions (also referenced as SysCon) if the Bus Grant 3 jumpers are set correctly on the VMEbus backplane. The system resource functions are explained in the Universe manual. (Contact Tundra at *www.tundra.com* for a PDF version of the Universe manual.) This function can be disabled using the XVME 689-VR7's jumper J3. See **Jumper Settings** in Chapter 2 (p. 2-2).

VMEbus Master Interface

The XVME 689-VR7 can be either a VMEbus master by accessing a **PCI slave** channel or the DMA channel initiates a transaction. There are 8 PCI slave images. The first PCI slave image has a 4K resolution the other have 64K resolution. The master can generate A16, A24, A32 VMEbus cycles for each PCI slave image. The address mode and type are also programmed on a PCI slave image basis. The PCI memory address location for the VMEbus master cycle is specified by the Base and Bound address. The VME address is calculated by adding the Base address to the Translation offset address.   All PCI slave images are located in the PCI bus Memory Space. The master cycles are all byte swapped maintaining address coherency.

Caution: PCI slave images mapped to a system DRAM area will access the system DRAM not the PCI slave image. Also the Universe configuration register has a higher priority than the PCI slave images. This means if the PCI slave image and the Universe configuration registers are mapped in to the same memory area the configuration registers will take precedence.

VMEbus Slave Interface

The XVME 689-VR7 can be either a VMEbus slave by being accessing a VMEbus slave image or the DMA channel initiates a transaction. There are eight PCI slave images. The first slave image has a 4K resolution the others (2-4,6-8) have 64K resolution. Slave images 1-8 have been implemented on the XVME 689-VR7. The slave can respond to A16, A24, A32 VMEbus cycles for each VMEbus slave image. The address mode and type are also programmed on a VMEbus slave image basis. The VMEbus memory address location for the VMEbus slave cycle is specified by the Base and Bound address. The PCI address is calculated by adding the Base address to the Translation offset address.

The XVME 689-VR7 DRAM memory is based on the PC/AT architecture and is not contiguous. The VMEbus Slave Images may be setup to allow this DRAM to appear as one Contiguous block. The first VMEbus slave Image must have Base and Bound register set to 640K.

(Example: VMEbus Slave Image 0 BS= 0000000h BD= A0000h TO = 0000000h)

The second VMEbus Slave Image must have the Base register set to be contiguous with the Bound register from the first VMEbus Slave Image. The Bound register is limited by the Total XVME 689-VR7 DRAM. The Translation Offset register is offset by 384K which is equivalent to the A0000h-FFFFFh range on the XVME 689-VR7 board.

(Example: VMEbus Slave Image 1 BS=A0000h BD= 400000h TO = 060000h)

This rather awkward mapping defined by the PC/AT architecture can also be over come if the VMEbus Slave Image window is always configured with a 1Mbyte Translation Offset. From a user and software standpoint this is always more desirable because the interrupt vector table, system parameters, and communication buffers (keyboard) are placed in low DRAM. This provides for more system protection.

Caution: When setting up slave images the address and other parameters should be set first. Then only after the VMEbus slave image is set up correctly should the VMEbus slave image be enabled. If a slave image is going to be remapped disable the slave image first then reset the address. After the image is configured correctly enable the image again.

The VMEbus slave cycle becomes a master cycle on the PCI bus. The PCI bus arbiter is the 6300ESB chip. It arbitrates between the various PCI masters, the Pentium, and the Local bus IDE bus mastering controller. Because the VMEbus can not be retried, all VMEbus slave cycles must be allowed to be processed. This becomes a problem when a Pentium cycle to the PCI slave image is in progress while a VMEbus slave cycle to the onboard DRAM is in progress. The Pentium cycle will not give up the PCI bus and the VMEbus slave cycle will not give up the VMEbus thus the XVME 689-VR7 becomes deadlocked. If the XVME 689-VR7 is to be used as a master and a slave at the same time, the VMEbus master cycles must obtain the VMEbus prior to initiating VMEbus cycles.

All Slave interface cycles are byte swapped to maintain address coherency.

VMEbus Interrupt Handling

The XVME 689-VR7 can service IRQ[7:1]. A register in the Universe enables which interrupt levels will be serviced by the XVME 689-VR7. When a VMEbus IRQ is asserted the Universe requests the VMEbus and generates and IACK cycle. Once the IACK cycle is complete a PCI bus interrupt is generated to allow the proper ISR(Interrupt service routine) to be executed. The Universe connects to all 4 PCI bus interrupts. These interrupts may be shared by other PCI bus devices. The BIOS maps the PCI bus interrupts to the AT-bus Interrupt controllers. The AT-bus interrupts must be uniquely mapped to each device.

Because the PCI devices share interrupt lines, all ISR routines must be prepared to chain the interrupt vector to allow the other devices to be serviced.

Note: The 6300ESB allows multiple PCI bus Interrupts to be mapped to one AT-bus interrupt.

Example: In the BIOS setup menu map the VMEbus IRQ(1) to PCI IRQ(11).

VMEbus Interrupt Generation

The XVME 689-VR7 can generate VMEbus interrupts on all 7 levels. There is a unique STATUS/ID associated with each level. The upper bits are programmed in the STATUS/ID register. The lowest bit is cleared if the source of the interrupt is a software Interrupt, and set for all other interrupt sources. Consult the Universe Users Manual for a more in depth explanation.

VMEbus Reset Options

When the front panel **Reset** switch is toggled, the XVME 689-VR7 can perform the following reset options:

1. Reset the VME backplane only.
2. Reset the XVME 689-VR7 CPU only.
3. Reset both.
4. Reset neither.

See **Switch Settings** in section 3 of this manual for information on how to configure SW1 for the **Reset** options.

Software-Selectable Byte-Swapping Hardware

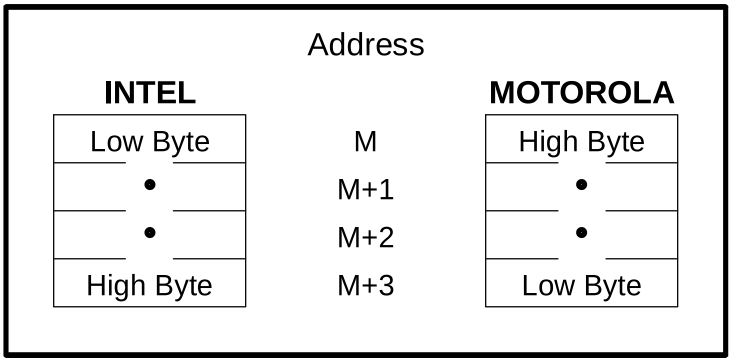
The VMEbus can be used to communicate to either Intel based modules or a Motorola based modules, these two companies have created data transaction that use different byte ordering in their data storage. A hardware approach to swapping these byte orders is a faster solution when compared to a software only byte swapping method. Software selectable byte-swapping hardware is integrated into the XVME 689-VR7 to allow for the difference between the Intel and Motorola byte-ordering schemes, allowing easy communication over the VMEbus. The byte-swapping package incorporates several buffers either to pass data straight through or to swap the data bytes as they are passed through.

Note

The configurable byte-swapping hardware does not support 64-bit byte-swapping. If needed, this should be implemented through software.

Byte-Ordering Schemes

The Motorola family of processors stores data with the least significant byte located at the highest address and the most significant byte at the lowest address. This is referred to as a big-endian bus and is the VMEbus standard. The Intel family of processors stores data in the opposite way, with the least significant byte located at the lowest address and the most significant byte located at the highest address. This is referred to as a little-endian (or PCI) bus. This fundamental difference is illustrated in Figure 4-1, which shows a 32-bit quantity stored by both architectures, starting at address *M.*



INTEL	Address	MOTOROLA
Low Byte	M	High Byte
•	M+1	•
•	M+2	•
High Byte	M+3	Low Byte

Figure 4-1 Byte Ordering Schemes

> **Note**
>
> The two architectures differ only in the way in which they store data into memory, not in the way in which they place data on the shared data bus.

The XVME 689-VR7 contains a Universe chip that performs address-invariant translation between the PCI bus (Intel architecture) and the VMEbus (Motorola architecture), and byte-swapping hardware to reverse the Universe chip byte-lane swapping. (Contact Tundra at *www.tundra.com* for a PDF version of the Universe manual.) Figure 4-2 shows address-invariant translation between a PCI bus and a VMEbus.

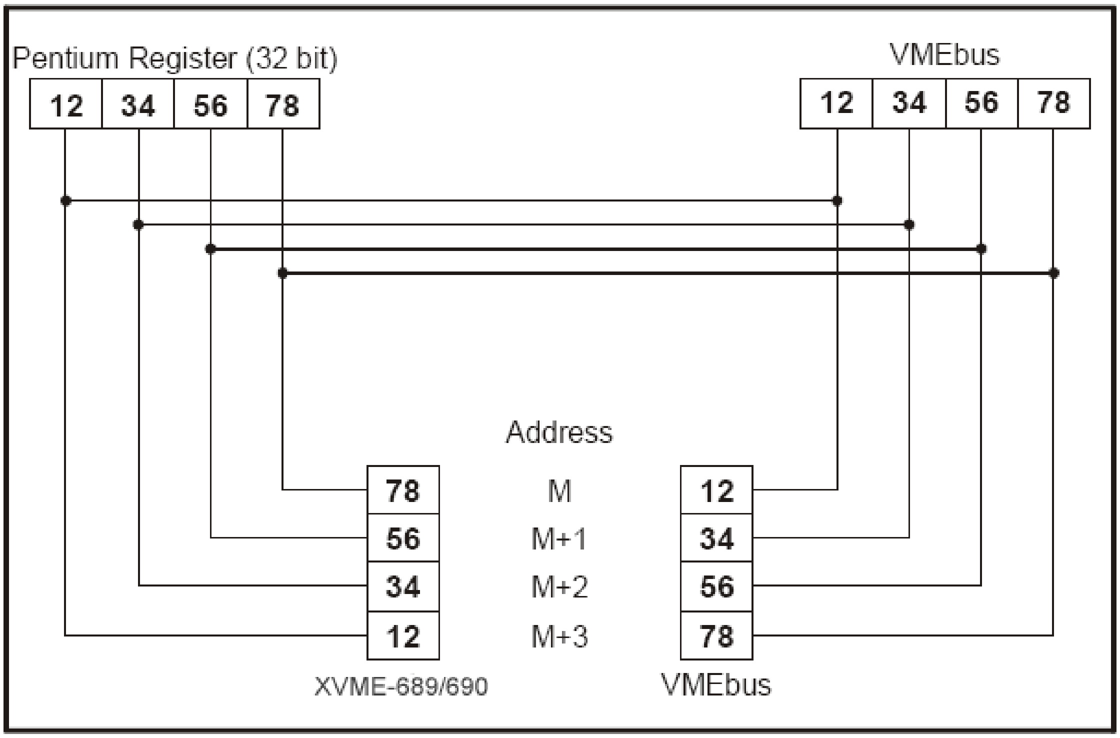


Figure 4-2 Address-Invariant Translation

Notice that the internal data storage scheme for the PCI (Intel) bus is different from that of the VME (Motorola) bus. For example, the byte **78** (the least significant byte) is stored at location *M* on the PCI machine while the byte **78** is stored at the location *M+3* on the VMEbus machine. Therefore, the data bus connections between the architectures must be mapped correctly.

Numeric Consistency

Numeric consistency, or data consistency, refers to communications between the XVME 689-VR7 and the VMEbus in which the byte-ordering scheme described above is maintained during the transfer of a 16-bit or 32-bit quantity. Numeric consistency is achieved by setting the XVME 689-VR7 buffers to pass data straight through, which allows the Universe chip to perform address-invariant byte-lane swapping. Numeric consistency is desirable for transferring integer data, floating-point data, pointers, etc. Consider the long word value **12345678h** stored at address *M* by both the XVME 689-VR7 and the VMEbus, as shown in Figure 4-3.

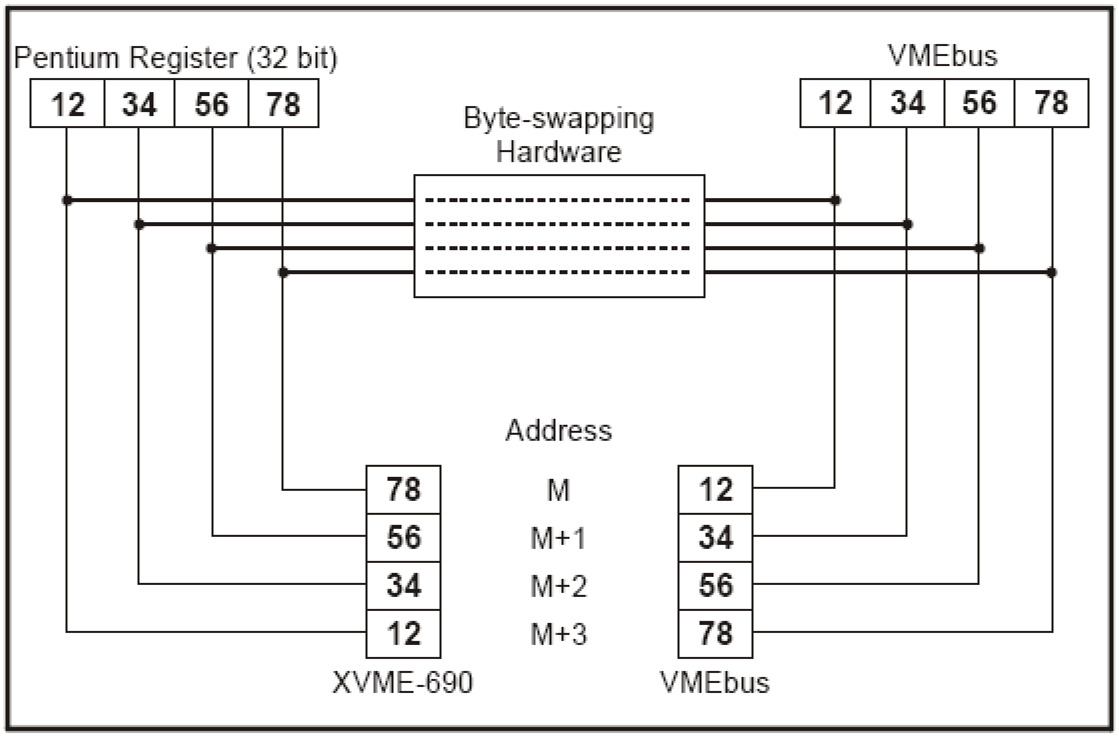


Figure 4-3 Maintaining Numeric Consistency

Due to the Universe chip, the data must be passed straight through the byte-swapping hardware. To do this, maintaining numeric consistency, enable the straight-through buffers by setting bits 6 and 7 of the Flash Paging and Byte Swap register (register 234h) to **1** (see p. 2-4).

Note

With the straight-through buffers enabled, the XVME 689-VR7 does not support unaligned transfers. Sixteen-bit or 32-bit transfers must have an even address.

Address Consistency

Address consistency, or address coherency, refers to communications between the XVME 689-VR7 and the VMEbus in which both architectures' addresses are the same for each byte. In other words, the XVME 689-VR7 and the VMEbus memory images appear the same. Address consistency is desirable for byte-oriented data such as strings or video image data. Consider the example of transferring the string `Text` to the VMEbus memory using a 32-bit transfer in Figure 4-4.

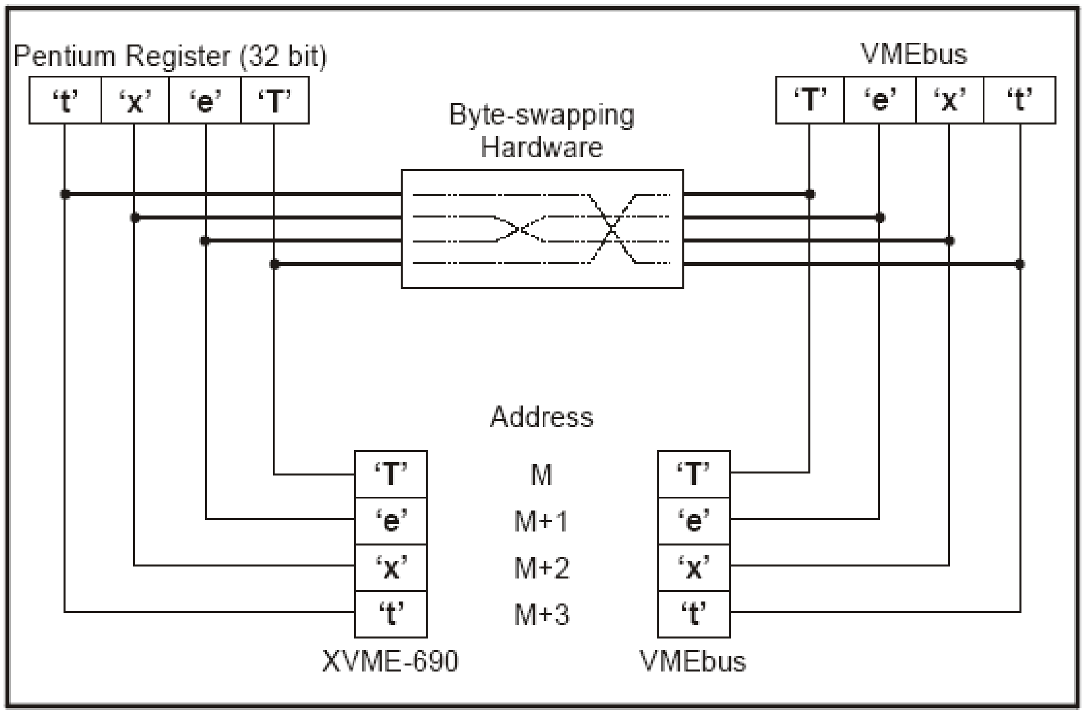


Figure 4-4 Maintaining Address Consistency

Notice that the data byte at each address is identical. To achieve this, the data bytes need to be swapped as they are passed from the PCI bus to the VMEbus. To maintain address consistency, enable the byte-swapping buffers by setting bits 6 and 7 of the Flash Paging and Byte Swap register (register 234h) to **`0`** (see p. 2-4).

Appendix A SDRAM and Battery Installation

Memory Type

The XVME 689-VR7 has one 200-pin DDR333 SDRAM memory module (SODIMM) site in which memory is inserted.

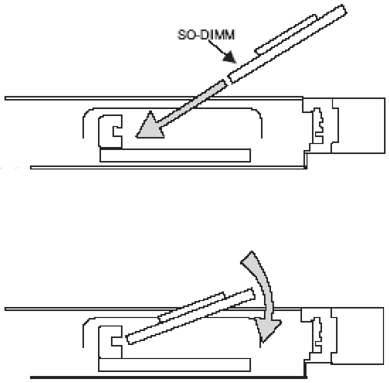
The XVME 689-VR7 supports 256MB, 512MB, 1MB and 2MB of PC2700 SDRAM.. Table A-1 lists the SODIMM configurations.

Table A-1 SDRAM SODIMM Configurations

Xembedded Part Number	Device Type and Size	Vendor	Vendor part number
200193	SODIMM 256MB DDR333	Micron	MT9VDDT3272HY-335F2
		Virtium	VL485L3223C-B3
200194	SODIMM 512MB DDR333	Micron	MT9VDDT6472HY-335F2
		Virtium	VL485L6523C-B3
200195	SODIMM 1GB DDR333	Micron	MT18VDDT12872HY-335F1
		Virtium	VL485L2925C-B3
200196	SODIMM 2GB DDR333	Virtium	

Installing SDRAM

Follow these steps to install the SODIMM:

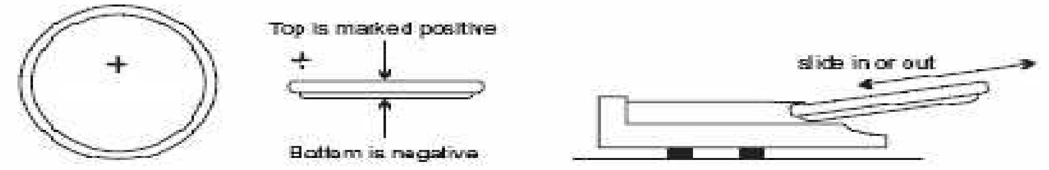


1. Follow standard antistatic procedures using a wrist strap to minimize the chance of damaging the XVME 689-VR7 and its components.
2. Power off the XVME 689-VR7, remove it from the VME backplane, and place it on a safe antistatic (grounded) surface.
3. Remove all connectors if not already removed.
4. Locate the PX connector on the XVME 689-VR7 slightly in front of the P1 VME backplane connector.
5. Pull the metal clips on either side of the SODIMM until it pops up at an angle (roughly 30° from horizontal).
6. Grasping the upper two corners or the edges of the SODIMM, gently pull it out of the socket and set it to the side.
7. Insert the new SODIMM until seated into the connector assuring it fits snugly into the connector retainer clips.

8. Gently push the SODIMM down until the metal clips snap into place to hold it. If you cannot gently push the SODIMM into position, you may need to redo step 7.
9. Replace the XVME 689-VR7 module, reconnect all connectors, etc.
10. Power up the unit and make sure that the memory is recognized (during boot up on the Boot-time diagnostic screen that can be turned on in the BIOS, see p. **Error! Bookmark not defined.**).

Module Battery Installation

During battery replacement, polarity must be observed in installing the coin battery. Please be sure to dispose of the spent battery in an environmentally correct manner. The replacement battery must be a CR2032 or equivalent type.



Index