Modular Computers®

VMP60

VME PowerPC™ Fast Ethernet/Serial Communications Controller

Manual ID: 22861, Rev. Index 02
October 2004

The product described in this manual is in compliance with all applied CE standards.

Revision History

Manual/Product Title:	VMP60		
Manual ID Number:	22861		
Rev. Index	**Brief Description of Changes**	**Board Index**	**Date of Issue**
01	Initial Issue	00	Apr 2001
02	Replaced BootLoader chapter with NetBoot-Loader chapter	00	Oct. 2004

Imprint

Copyright © 2004 PEP Modular Computers GmbH. All rights reserved. This manual may not be copied, photocopied, reproduced, translated or converted to any electronic or machine-readable form in whole or in part without prior written approval of PEP Modular Computers GmbH.

Modular Computers ®

Disclaimer:

PEP Modular Computers GmbH rejects any liability for the correctnesss and completeness of this manual as well as its suitability for any particular purpose.

This manual was realized by: **TPD/Engineering, PEP Modular Computers GmbH**.

Preface

Revision History ii
Imprint ii
Proprietary Note xiii
Trademarks xiii
Environmental Protection Statement xiii
Explanation of Symbols xiv
For Your Safety xv
Two Year Warranty xvii

Chapter 1

1. Introduction ..... 1 - 3
1.1 System Overview ..... 1 - 3
1.2 Product Overview ..... 1 - 3
1.3 Board Overview ..... 1 - 4
1.3.1 Board Introduction ..... 1 - 4
1.3.2 Board-Specific Information ..... 1 - 5
1.3.3 Main Memory and BootRom ..... 1 - 5
1.4 System Relevant Information ..... 1 - 5
1.4.1 System Configuration ..... 1 - 6
1.4.2 Bootstrap Loader ..... 1 - 6
1.4.3 Operating Systems ..... 1 - 6
1.5 Board Diagrams ..... 1 - 6
1.5.1 System Level Interfacing ..... 1 - 7
1.5.2 Front Panel ..... 1 - 8
1.5.3 Board Layouts ..... 1 - 9
1.6 Technical Specifications ..... 1 - 10
1.7 Applied Standards ..... 1 - 12
1.7.1 CE Compliance ..... 1 - 12
1.7.2 Mechanical Compliance ..... 1 - 12

 © 2004 PEP Modular Computers GmbH

1.7.3 Environmental Tests ..... 1 - 12
1.8 Related Publications ..... 1 - 12
1.8.1 VMEbus/CXC Systems ..... 1 - 12
1.8.2 Manufacturers' Component Documentation ..... 1 - 12

Chapter 2

2. Functional Description ..... 2 - 3
2.1 General Information ..... 2 - 3
2.1.1 MPC860T PowerQUICC™ Microprocessor CPU ..... 2 - 4
2.1.2 System Interfaces ..... 2 - 5
2.1.3 Memory ..... 2 - 5
2.1.4 Monitor and Control ..... 2 - 6
2.1.5 Software ..... 2 - 6
2.1.6 Board-Level Interfacing Diagram ..... 2 - 6
2.2 System Interfaces ..... 2 - 8
2.2.1 VMEbus ..... 2 - 8
2.2.2 (Enhanced) Controller Extension Connector Bus ((E)CXC Bus) 2 - 10
2.2.3 Communications ..... 2 - 11
2.2.4 Test and Program Development ..... 2 - 15
2.3 Memory ..... 2 - 16
2.3.1 DM60x Memory Module Interface ..... 2 - 16
2.3.2 Socket 1 Memory Expansion ..... 2 - 16
2.3.3 SPI EEPROM ..... 2 - 17
2.4 Monitor and Control (M/C) ..... 2 - 17
2.4.1 Pre-Operation M/C ..... 2 - 17
2.4.2 Operation M/C ..... 2 - 18
2.5 VSBCxxx/VMxx-to-VMP60 System Upgrading ..... 2 - 22
2.6 System Enhancement ..... 2 - 24

Chapter 3

3. Installation ..... 3 - 3

 © 2004 PEP Modular Computers GmbH

3.1 Hardware Installation ..... 3 - 3
3.1.1 Safety Requirements ..... 3 - 3
3.1.2 Installation Procedures ..... 3 - 4
3.1.3 Removal Procedures ..... 3 - 4
3.2 Software Installation ..... 3 - 4

Chapter 4

4. Configuration ..... 4 - 3
4.1 Jumper Settings ..... 4 - 3
4.1.1 Overview ..... 4 - 3
4.1.2 J1 - Boot Device Select ..... 4 - 4
4.1.3 J2 - (E)CXC Mode Select ..... 4 - 4
4.1.4 J3 - VME Bus Reset ..... 4 - 4
4.1.5 J4 - EEPROM Write Protection ..... 4 - 4
4.1.6 J5, J6, and J9 - CPU Clock Speed Select ..... 4 - 5
4.1.7 J7 and J10 - Installed ROM Configuration ..... 4 - 5
4.1.8 J8 - Baud Rate Clock ..... 4 - 6
4.1.9 J11 and J16 - Layer Bridge ..... 4 - 6
4.1.10 J12 - 24 MHz Clock ..... 4 - 6
4.1.11 J13 - CXC Pin A5 Use Select ..... 4 - 6
4.1.12 J14 and J15 - SRAM Size Select ..... 4 - 7
4.1.13 J17 - SRAM Data Retention ..... 4 - 7
4.1.14 C1 - Signal Ground Connection to Chassis (Ground) ..... 4 - 7
4.2 Board Address Map ..... 4 - 8
4.3 Board Registers ..... 4 - 9
4.3.1 Overview ..... 4 - 9
4.3.2 BOARD_CON_STA_REG_1 Register ..... 4 - 10
4.3.3 BOARD_CON_STA_REG_2 Register ..... 4 - 11
4.3.4 BOARD_IRQ_PEND_REG Register ..... 4 - 12
4.4 VME Control and Status Registers ..... 4 - 13
4.4.1 Overview ..... 4 - 13
4.4.2 VME_IRQ_MASK_REG Register ..... 4 - 14

 © 2004 PEP Modular Computers GmbH

4.4.3 VME_IRQ_PEND_REG Register ....4 - 15
4.4.4 VME_CON_STA_REG Register ....4 - 16

Chapter 5

5. NetBootLoader ....5 - 3
5.1 General Operation ....5 - 3
5.2 NetBootLoader Interfaces ....5 - 3
5.2.1 ABT (Abort) Switch ....5 - 4
5.2.2 TERM Serial Interface ....5 - 4
5.2.3 SER0 Serial Interface ....5 - 4
5.2.4 Ethernet Interface ....5 - 4
5.3 NetBootLoader Functions ....5 - 4
5.3.1 NetBootLoader Control ....5 - 5
5.3.2 System Status Monitoring ....5 - 5
5.3.3 ftp Server Access ....5 - 6
5.3.4 FLASH Operation ....5 - 6
5.3.5 Motorola S-Records ....5 - 6
5.4 Operating the NetBootLoader ....5 - 7
5.4.1 Initial Setup ....5 - 7
5.4.2 Accessing the NetBootLoader ....5 - 7
5.4.3 NetBootLoader Configuration ....5 - 8
5.4.4 telnet Login ....5 - 9
5.4.5 FLASH Operations ....5 - 9
5.4.6 Updating the NetBootLoader ....5 - 11
5.4.7 Uploading a FLASH Area ....5 - 12
5.5 Plug and Play ....5 - 12
5.6 Porting an Operating System to the CPU Board ....5 - 12
5.7 Commands ....5 - 13

Annex A

A. DM60x Memory Modules .... A - 3

 © 2004 PEP Modular Computers GmbH

A.1 General ..... A - 3
A.2 DM600 ..... A - 4
A.2.1 Board Layout and Jumper Location ..... A - 4
A.2.2 Jumper Description and Flash Addresses ..... A - 4
A.3 DM601 ..... A - 5
A.3.1 Board Layout and Jumper Location ..... A - 5
A.3.2 Jumper Description and Flash Addresses ..... A - 5
A.4 DM602 ..... A - 6
A.4.1 Board Layout and Jumper Location ..... A - 6
A.4.2 Jumper Description and Flash Addresses ..... A - 6
A.5 DM603 ..... A - 7
A.5.1 Board Layout and Jumper Location ..... A - 7
A.5.2 Jumper Description and Flash Addresses ..... A - 7
A.6 DM604 ..... A - 8
A.6.1 Board Layout and Jumper Location ..... A - 8
A.6.2 Jumper Description and Flash Addresses ..... A - 8
A.7 DM605 ..... A - 9
A.7.1 Board Layout and Jumper Location ..... A - 9
A.7.2 Jumper Description and Flash Addresses ..... A - 9

Annex B

B. (E)CXC ..... B - 3
B.1 (E)CXC Address Ranges ..... B - 3
B.2 CXC Generic Pinouts ..... B - 5
B.3 CXC Pinout Cross Reference to MPC860T ..... B - 6

 © 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

TABLES

1-1 System Relevant Information ..... 1 - 6
2-2 VMP60 Main Specifications ..... 1 - 10
2-1 Address Modifier (AM) Codes ..... 2 - 8
2-2 VMEbus Connector Pinout (CON1) ..... 2 - 9
2-3 CXC Connector Pinout (CON2) ..... 2 - 10
2-4 VMP60 Serial Interface Channel Configurations ..... 2 - 12
2-5 MPC860T Serial Interface Routing ..... 2 - 12
2-6 TERM (CON5) Serial Interface Pinout ..... 2 - 13
2-7 SER0 (CON4) Serial Interface Pinout ..... 2 - 13
2-8 Fast Ethernet Connector (CON3) Pinout ..... 2 - 14
2-9 DEBUG Connector (CON7) Pinout ..... 2 - 15
2-10 JTAG Connector (CON6) Pinout ..... 2 - 15
2-11 Memory Devices Supported by Socket 1 ..... 2 - 17
2-12 System Status Indicators ..... 2 - 18
2-13 VMP60 Reset Sources ..... 2 - 19
2-14 VSBCxxx/VMxx - VMP60 Porting Information ..... 2 - 22
4-1 VMP60 Jumper Settings Overview ..... 4 - 3
4-2 J1 - Boot Device Select ..... 4 - 4
4-3 J2 - (E)CXC Mode Select ..... 4 - 4
4-4 J3 - VME Bus Reset ..... 4 - 4
4-5 J4 - EEPROM Write Protection ..... 4 - 4
4-6 J5, J6, and J9 - CPU Clock Speed Select ..... 4 - 5
4-7 J7 and J10 - ROM Type and Size Select ..... 4 - 5
4-8 J8 - Baud Rate Clock ..... 4 - 6
4-9 J12 - 24 MHz Clock ..... 4 - 6
4-10 J13 - CXC Pin A5 Use Select ..... 4 - 6
4-11 J14 and J15 - SRAM Size Select ..... 4 - 7
4-12 J17 - SRAM Data Retention ..... 4 - 7
4-13 C1 - Signal Ground Connection to Chassis (Ground) ..... 4 - 7
4-14 VMP60 Board Address Map ..... 4 - 8
4-15 Board Registers ..... 4 - 9
4-16 BOARD_CON_STA_REG_1 Register ..... 4 - 10

 © 2004 PEP Modular Computers GmbH

4-17 BOARD_CON_STA_REG_2 Register ..... 4 - 11
4-18 BOARD_IRQ_PEND_REG Register ..... 4 - 12
4-19 VME Control and Status Registers ..... 4 - 13
4-20 VME_IRQ_MASK_REG ..... 4 - 14
4-21 VME_IRQ_PEND_REG Register ..... 4 - 15
4-22 VME_CON_STA_REG Register ..... 4 - 16
A-1 DM60x Memory Module Types and Configurations ..... A - 3
A-2 Jumper: J1 Settings and Flash Memory Address Ranges ..... A - 4
A-3 Jumper: J1 Settings and Flash Memory Address Ranges ..... A - 5
A-4 Jumpers: J1 and J2 Settings and Flash Memory Address Ranges ..... A - 6
A-5 Jumpers: J1 and J2 Settings and Flash Memory Address Ranges ..... A - 7
A-6 Jumpers: J1 and J2 Settings and Flash Memory Address Ranges ..... A - 8
A-7 Jumpers: J1 and J2 Settings and Flash Memory Address Ranges ..... A - 9
B-1 CXC Address Range ..... B - 3
B-2 Enhanced CXC Address Range ..... B - 4
B-3 CXC Connector Pinouts ..... B - 5
B-4 Cross Reference of CXC Signals to MPC860T ..... B - 6

 © 2004 PEP Modular Computers GmbH

FIGURES

1-1 VMP60 System Level Interfacing Diagram ..... 1 - 7
1-2 VMP60 Front Panel ..... 1 - 8
2-3 VMP60 Board (Front View) ..... 1 - 9
3-4 VMP60 Board (Rear View) ..... 1 - 9
2-1 VMP60 Board-Level Interfacing ..... 2 - 7
2-2 VMEbus Connector (CON1) ..... 2 - 9
2-3 (E)CXC Bus Connector (CON2) ..... 2 - 10
2-4 TERM (CON5) and SER0 (CON4) Serial Interfaces ..... 2 - 13
2-5 Fast Ethernet Connector (CON3) ..... 2 - 14
2-6 DEBUG Connector (CON7) ..... 2 - 15
2-7 JTAG Connector (CON6) ..... 2 - 15
2-8 Socket 1 Memory Expansion ..... 2 - 17
2-9 Gold Cap Charge and Discharge Characteristics ..... 2 - 21
A-1 DM600 Memory Module ..... A - 4
A-2 DM601 Memory Module ..... A - 5
A-3 DM602 Memory Module ..... A - 6
A-4 DM603 Memory Module ..... A - 7
A-5 DM604 Memory Module ..... A - 8
A-6 DM605 Memory Module ..... A - 9

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

Preface

Proprietary Note

This document contains information proprietary to *PEP Modular Computers*. It may not be copied or transmitted by any means, disclosed to others, or stored in any retrieval system or media without the prior written consent of *PEP Modular Computers GmbH* or one of its authorized agents.

The information contained in this document is, to the best of our knowledge, entirely correct. However, *PEP Modular Computers* cannot accept liability for any inaccuracies or the consequences thereof, or for any liability arising from the use or application of any circuit, product, or example shown in this document.

PEP Modular Computers reserves the right to change, modify, or improve this document or the product described herein, as seen fit by *PEP Modular Computers* without further notice.

Trademarks

PEP Modular Computers, the *PEP* logo and, if occurring in this manual, “CXM” are trade marks owned by *PEP Modular Computers GmbH*, Kaufbeuren (Germany). In addition, this document may include names, company logos and trademarks, which are registered trademarks and, therefore, proprietary to their respective owners.

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.

© 2004 PEP Modular Computers GmbH

Explanation of Symbols



CE Conformity

This symbol indicates that the product described in this manual is in compliance with all applied CE standards. Please refer also to the section “Applied Standards” in this manual.



Caution, Electric Shock!

This symbol and title warn of hazards due to electrical shocks (> 60V) when touching products or parts of them. Failure to observe the precautions indicated and/or prescribed by the law may endanger your life/health and/or result in damage to your material.

Please refer also to the section “High Voltage Safety Instructions” on the following page.



Warning, ESD Sensitive Device!

This symbol and title inform that electronic boards and their components are sensitive to static electricity. Therefore, care must be taken during all handling operations and inspections of this product, in order to ensure product integrity at all times.

Please read also the section “Special Handling and Unpacking Instructions” on the following page.



Warning!

This symbol and title emphasize points which, if not fully understood and taken into consideration by the reader, may endanger your health and/or result in damage to your material.



Note...

This symbol and title emphasize aspects the reader should read through carefully for his or her own advantage.

PEP Advantage

This symbol and title emphasize advantages or positive aspects of a product and/or procedure.

© 2004 PEP Modular Computers GmbH

For Your Safety

Your new *PEP* product was developed and tested carefully to provide all features necessary to ensure its compliance with electrical safety requirements. It was also designed for a long fault-free life. However, the life expectancy of your product can be drastically reduced by improper treatment during unpacking and installation. Therefore, in the interest of your own safety and of the correct operation of your new *PEP* product, you are requested to conform with the following guidelines.

High Voltage Safety Instructions



Warning!

All operations on this device must be carried out by sufficiently skilled personnel only.



Caution, Electric Shock!

Before installing your new *PEP* product into a system always ensure that your mains power is switched off. This applies also to the installation of piggybacks.

Serious electrical shock hazards can exist during all installation, repair and maintenance operations with this product. Therefore, always unplug the power cable and any other cables which provide external voltages before performing work.

Special Handling and Unpacking Instructions



ESD Sensitive Device!

Electronic boards and their components are sensitive to static electricity. Therefore, care must be taken during all handling operations and inspections of this product, in order to ensure product integrity at all times.

- Do not handle this product out of its protective enclosure while it is not used for operational purposes unless it is otherwise protected.
- Whenever possible, unpack or pack this product only at EOS/ESD safe work stations. Where a safe work station is not guaranteed, it is important for the user to be electrically discharged before touching the product with his/her hands or tools. This is most easily done by touching a metal part of your system housing.
- It is particularly important to observe standard anti-static precautions when changing piggybacks, ROM devices, jumper settings etc. If the product contains batteries for RTC or memory back-up, ensure that the board is not placed on conductive surfaces, including anti-static plastics or sponges. They can cause short circuits and damage the batteries or conductive circuits on the board.

© 2004 PEP Modular Computers GmbH

General Instructions on Usage

- In order to maintain *PEP's* product warranty, this product must not be altered or modified in any way. Changes or modifications to the device, which are not explicitly approved by *PEP Modular Computers* and described in this manual or received from *PEP* Technical Support as a special handling instruction, will void your warranty.
- This device should only be installed in or connected to systems that fulfill all necessary technical and specific environmental requirements. This applies also to the operational temperature range of the specific board version, which must not be exceeded. If batteries are present their temperature restrictions must be taken into account.
- In performing all necessary installation and application operations, please follow only the instructions supplied by the present manual.
- Keep all the original packaging material for future storage or warranty shipments. If it is necessary to store or ship the board please re-pack it as nearly as possible in the manner in which it was delivered.
- Special care is necessary when handling or unpacking the product. Please, consult the special handling and unpacking instruction on the previous page of this manual.

© 2004 PEP Modular Computers GmbH

Two Year Warranty

PEP Modular Computers grants the original purchaser of *PEP* products a ***TWO YEAR LIMITED HARDWARE WARRANTY*** as described in the following. However, no other warranties that may be granted or implied by anyone on behalf of *PEP* are valid unless the consumer has the express written consent of *PEP Modular Computers*.

PEP Modular Computers warrants their own products, excluding software, to be free from manufacturing and material defects for a period of 24 consecutive months from the date of purchase. This warranty is not transferable nor extendible to cover any other users or long-term storage of the product. It does not cover products which have been modified, altered or repaired by any other party than *PEP Modular Computers* or their authorized agents. Furthermore, any product which has been, or is suspected of being damaged as a result of negligence, improper use, incorrect handling, servicing or maintenance, or which has been damaged as a result of excessive current/voltage or temperature, or which has had its serial number(s), any other markings or parts thereof altered, defaced or removed will also be excluded from this warranty.

If the customer's eligibility for warranty has not been voided, in the event of any claim, he may return the product at the earliest possible convenience to the original place of purchase, together with a copy of the original document of purchase, a full description of the application the product is used on and a description of the defect. Pack the product in such a way as to ensure safe transportation (see our safety instructions).

PEP provides for repair or replacement of any part, assembly or sub-assembly at their own discretion, or to refund the original cost of purchase, if appropriate. In the event of repair, refunding or replacement of any part, the ownership of the removed or replaced parts reverts to *PEP Modular Computers*, and the remaining part of the original guarantee, or any new guarantee to cover the repaired or replaced items, will be transferred to cover the new or repaired items. Any extensions to the original guarantee are considered gestures of goodwill, and will be defined in the "Repair Report" issued by *PEP* with the repaired or replaced item.

PEP Modular Computers will not accept liability for any further claims resulting directly or indirectly from any warranty claim, other than the above specified repair, replacement or refunding. In particular, all claims for damage to any system or process in which the product was employed, or any loss incurred as a result of the product not functioning at any given time, are excluded. The extent of *PEP Modular Computers* liability to the customer shall not exceed the original purchase price of the item for which the claim exists.

PEP Modular Computers issues no warranty or representation, either explicit or implicit, with respect to its products' reliability, fitness, quality, marketability or ability to fulfil any particular application or purpose. As a result, the products are sold "as is," and the responsibility to ensure their suitability for any given task remains that of the purchaser. In no event will *PEP* be liable for direct, indirect or consequential damages resulting from the use of our hardware or software products, or documentation, even if PEP were advised of the possibility of such claims prior to the purchase of the product or during any period since the date of its purchase.

Please remember that no *PEP Modular Computers* employee, dealer or agent is authorized to make any modification or addition to the above specified terms, either verbally or in any other form, written or electronically transmitted, without the company's consent.

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

Chapter 1

Introduction

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

1. Introduction

1.1 System Overview

The *PEP Modular Computers* product described in this manual operates with the VME-bus architecture. In addition, this product also supports the CXC and/or Enhanced CXC [(E)CXC] local mezzanine interface standards which represent a streamlined variant of the VMEbus standard itself. Thus a wide range of I/O functions for industrial applications are supplied. Some of the major advantages of the VMEbus standard are:

- Internationally accepted VITA standards
- Broad range of available VMEbus solutions
- Scalable processor performance

In addition, in combination with (E)CXC technology the VMEbus equipment offers the following advantages:

- Lower costs and optimized reliability thanks to reduced design complexity
- Compacter I/O subsystems due to simplified I/O wiring

For detailed information concerning the VMEbus and (E)CXC standards, please consult the VMEbus and CXC Specifications which are available via the *VMEbus International Trade Association (VITA): http://www.vita.com*.

Many system relevant features that are specific for VMEbus systems can be found in the ANSI/VITA VME64 Standard and in the VITA/*PEP* CXC MPI Specification which also applies to VMEbus MPI carriers.

The VME64 Standard includes the following information:

- VMEbus specification
- Signal lines, bus modules, typical operation
- Electrical specifications
- Mechanical specifications

The CXC MPI Specification includes the following information:

- Mechanical dimensions
- Electrical specifications
- Interface description
- ID Byte assignment

With reference to the (E)CXC aspects of mixed VMEbus/CXC systems please refer also to the *PEP* CXC Reference Manual.

1.2 Product Overview

The VMP60 is a comprehensive single-height PowerQUICC™ computing platform which combines a modern microprocessor with flexible I/O through the use of VMEbus and (E)CXC bus interfaces and excellent communication ability afforded by the Motorola PowerQUICC™ RISC controller.

The VMP60 with its embedded PowerPC™ core (EPC) not only achieves high computational performance but dispenses with the usual restrictions associated with serial com-

 © 2004 PEP Modular Computers GmbH

munication using currently available off-the-shelf products. The PowerQUICC™ processor's internal Communications Processor Module (CPM) combined with the standard built-in Fast Ethernet interface makes the VMP60 ideally suited for data communication and networking applications.

The VMP60 employs an OS independent boot loader that enables the loading of any operating system available for the PowerPC™. This boot loader allows for updating of the Flash contents and automatically downloads from Flash to DRAM before booting the OS. For performance reasons the OS and user programs are started from the DRAM.

1.3 Board Overview

1.3.1 Board Introduction

The VMP60 is a 3U/4HP VME/(Enhanced)CXC combined system and communications controller board based on Motorola's MPC860T PowerQUICC™ (Quad Integrated Communications Controller) microprocessor. This processor combines four high-speed multi-protocol serial I/O channels, two RS232 data input/terminal interfaces, and a Fast Ethernet controller to provide outstanding flexibility and computing power for advanced computer controlled communication applications. The VMP60 realizes the advantages of the MPC860T in a single board which provides two serial I/O channels and one Fast Ethernet interface onboard with additional expansion capability being provided by either the VMEbus or the CXCbus.

Some of the outstanding features of this product are:

- VME system and communications controller board
- Master or slave operation, and as VMEbus system controller
- 32-bit Motorola MPC860T PowerQUICC™ microprocessor with integrated CPU, communications controller, and Fast Ethernet controller (FEC)
- 80 MHz CPU frequency
- CPU on-chip background debugging
- Up to 64 MB DRAM
- Up to 4 MB Flash memory
- 1 MB SRAM (backed-up)
- 2 kBit serial EEPROM
- Up to 1 MB of EPROM or up to 512 kB of Flash memory as optional boot device via an onboard socket
- Real-time clock (backed-up)
- Five different communication standards are supported:
  - Serial I/O (RS232, RS485; RS422 on request)
  - Ethernet (10baseT or 100baseT Fast Ethernet)
- Expandability through external serial interface modules
- Reset and Abort control (front panel buttons)
- Front panel system status indicators
  - Two general purpose and one watchdog
  - Three Fast Ethernet status indicators
- VxWorks® drivers

 © 2004 PEP Modular Computers GmbH

1.3.2 Board-Specific Information

The VMP60 mainboard is provided with the following standard connectors:

- Two non-optoisolated RS232 serial interfaces on front panel: TERM (RJ12) and SER0 (RJ45)
- One Fast Ethernet interface on front panel (RJ45)
- One set of memory piggyback interface connectors (two 50-pin female connectors)
- One EPROM/Flash 32-pin DIP socket
- A background debug mode (BDM) interface (one 10-pin, 2-row, male connector)
- A JTAG/ISP programming and testing interface (one 10-pin, 2-row, male connector)
- An Enhanced CXC mezzanine interface (one 96-pin, 3-row, male connector)
- A VMEbus backplane interface (one 96-pin, DIN 41612 style C, male connector)

Applications requiring further communication interfaces may be upgraded by means of external serial interface modules which provide the following interface extension possibilities:

- RS232 serial interface connectors (non-optoisolated)
- Serial interface piggyback ports
- Serial communications piggyback port

Refer to chapter 2, Functional Description for more information on system enhancements.

1.3.3 Main Memory and BootRom

VMP60 main memory and bootROM is provided using DM60x Memory Modules. By means of these modules the following memory configurations are possible:

- Up to 64 MB of DRAM
- Up to 4 MB of Flash memory which can be used for the Bootstrap Loader and application.

For a detailed description of the DM60x Memory Modules please refer to appendix A of this manual.

In addition to the Flash memory available on the DM60x Memory Modules there is a single 32-pin DIP socket on the VMP60 main board which accommodates up to 1 MB of EPROM or 512 kB of Flash memory which can be used for the Bootstrap Loader instead of the Flash memory located on the DM60x Memory Module.

1.4 System Relevant Information

The following system relevant information is general in nature but should still be considered when developing applications using the VMP60.

© 2004 PEP Modular Computers GmbH

Table 1-1: System Relevant Information

SUBJECT	INFORMATION
System Configuration	A VMP60-based system is made up of up to a maximum of 21 VME boards whereby one board must function as a system controller.
Master / Slave Functionality	The VMP60 can operate either as a master or a slave.
Board Location in the System	The VMP60 board may be installed in any available slot.
Operating Systems	The VMP60 is provided with support for the VxWorks® operating system. Support for other operating systems is available on request.

1.4.1 System Configuration

As a combined VMEbus system and communications controller board the VMP60 will also function as a system controller if installed in the system controller slot (slot 1).

The only limitation to the number of VMP60's which may be installed in a VMEbus system is the availability of slots. The VMP60 may also be installed in any available slot. The only restriction here is if it is used with a CXC mezzanine board there must be enough space to the right of the VMP60 to also accommodate the CXC board.

1.4.2 Bootstrap Loader

Via the VMP60 front panel serial interface connectors the flash memory of the board's memory piggyback can be re-programmed by means of the Bootstrap Loader which is delivered already installed in the DM60x memory piggybacks. This stand-alone software has the capability of loading Flash memory from Motorola S-records or from any absolute address. If the downloaded image does not work properly, the Bootstrap Loader can be re-entered, the memory contents analyzed and a further programming cycle initiated



Warning!

To avoid damaging of your Bootstrap Loader and consequently leaving your board unusable, please read chapter 5 before resetting the flash contents of your VMP60 board.

1.4.3 Operating Systems

The VMP60 can operate under the following operating system:

- VxWorks®

Porting to other operating systems can be done on request.

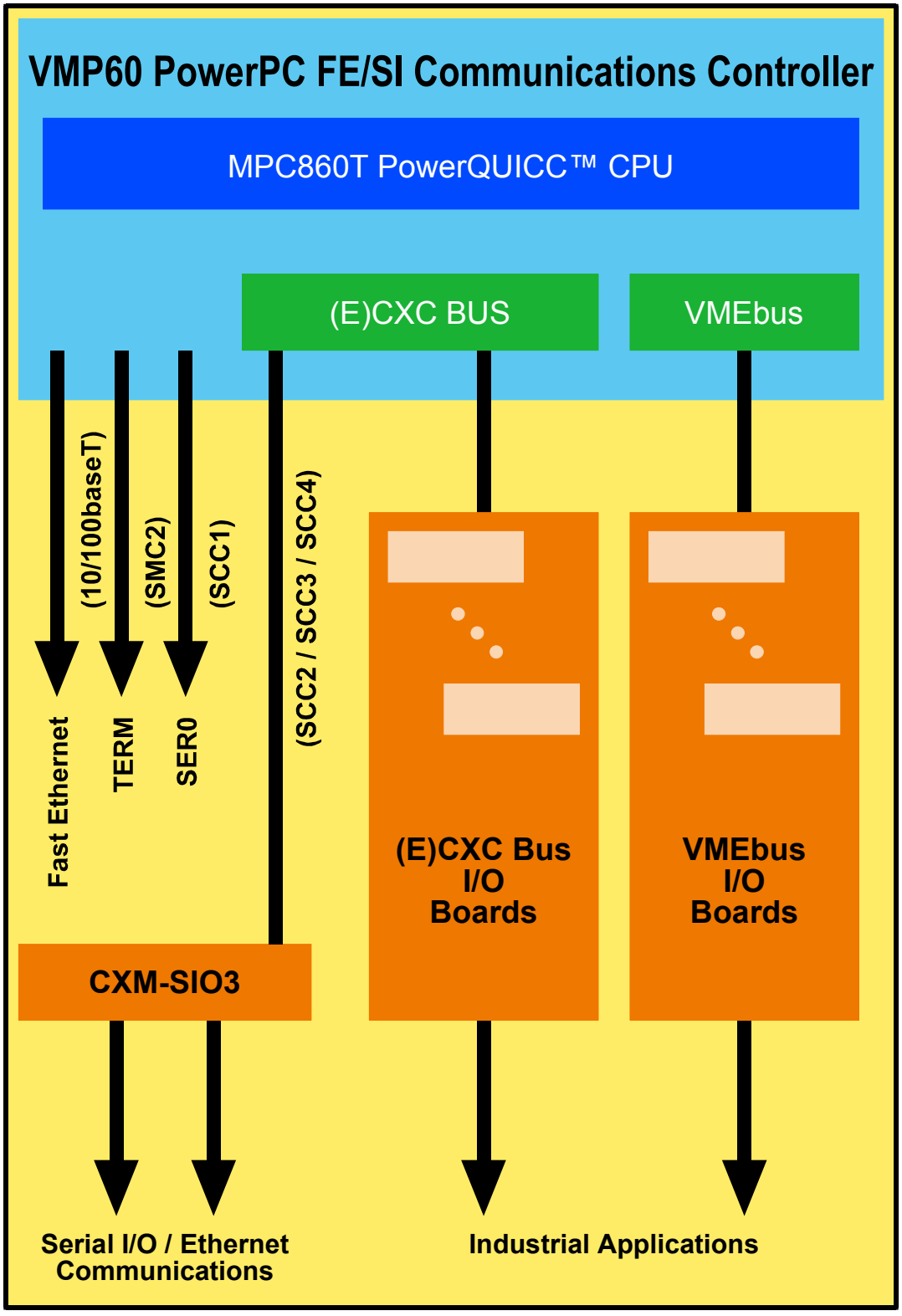
1.5 Board Diagrams

The following diagrams provide additional information concerning board functionality and component layout.

 © 2004 PEP Modular Computers GmbH

1.5.1 System Level Interfacing

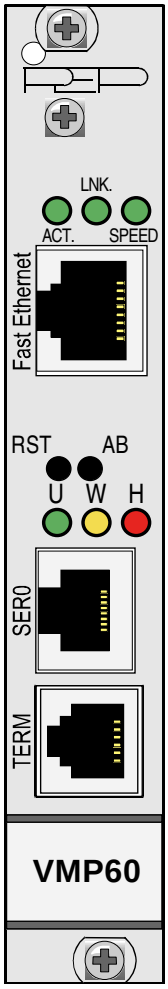
Figure 1-1: VMP60 System Level Interfacing Diagram



© 2004 PEP Modular Computers GmbH

1.5.2 Front Panel

Figure 1-2: VMP60 Front Panel



Legend:

LED1 Group

- Green (“ACT.”): Data send/receive
- Green (“LNK.”): Link established in both directions
- Green (“SPEED”): Data rate: on = 100 Mbps; off = 10 Mbps

Push Buttons:

- RST (left): Reset
- AB (right): Abort

LED2 Group

- Green (“U”): General purpose
- Yellow (“W”): Watchdog
- Red (“H”): General purpose

© 2004 PEP Modular Computers GmbH

1.5.3 Board Layouts

Figure 2-3: VMP60 Board (Front View)

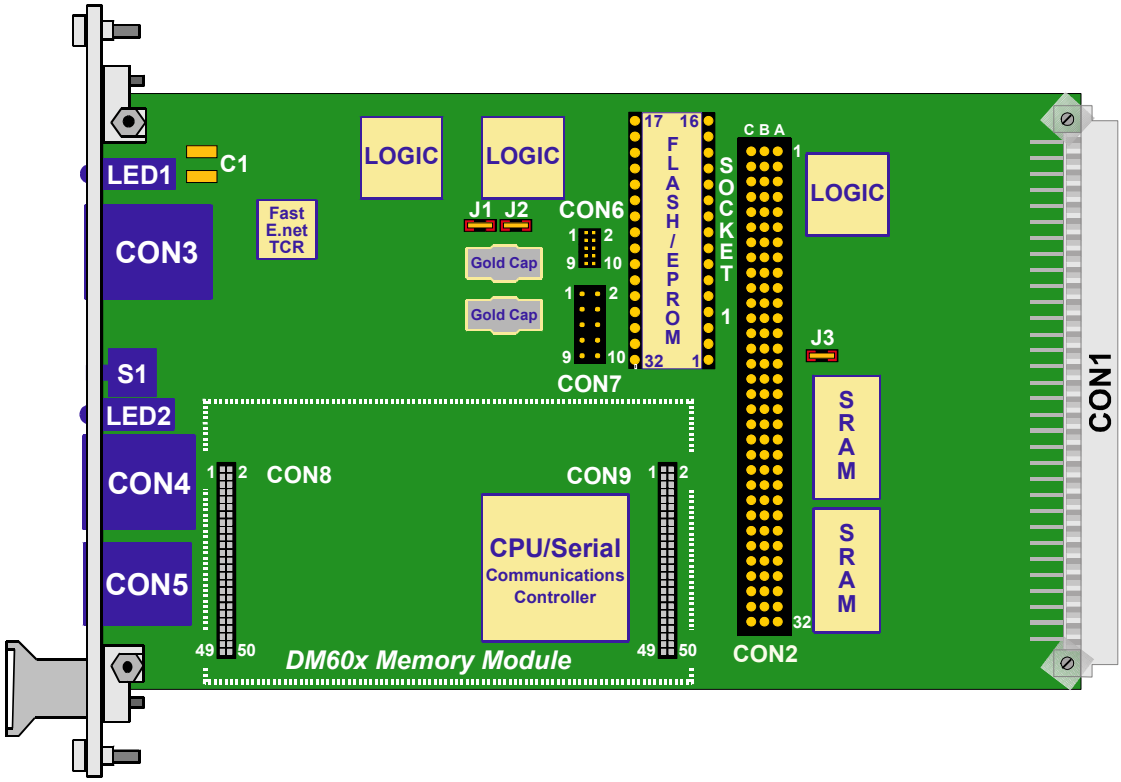
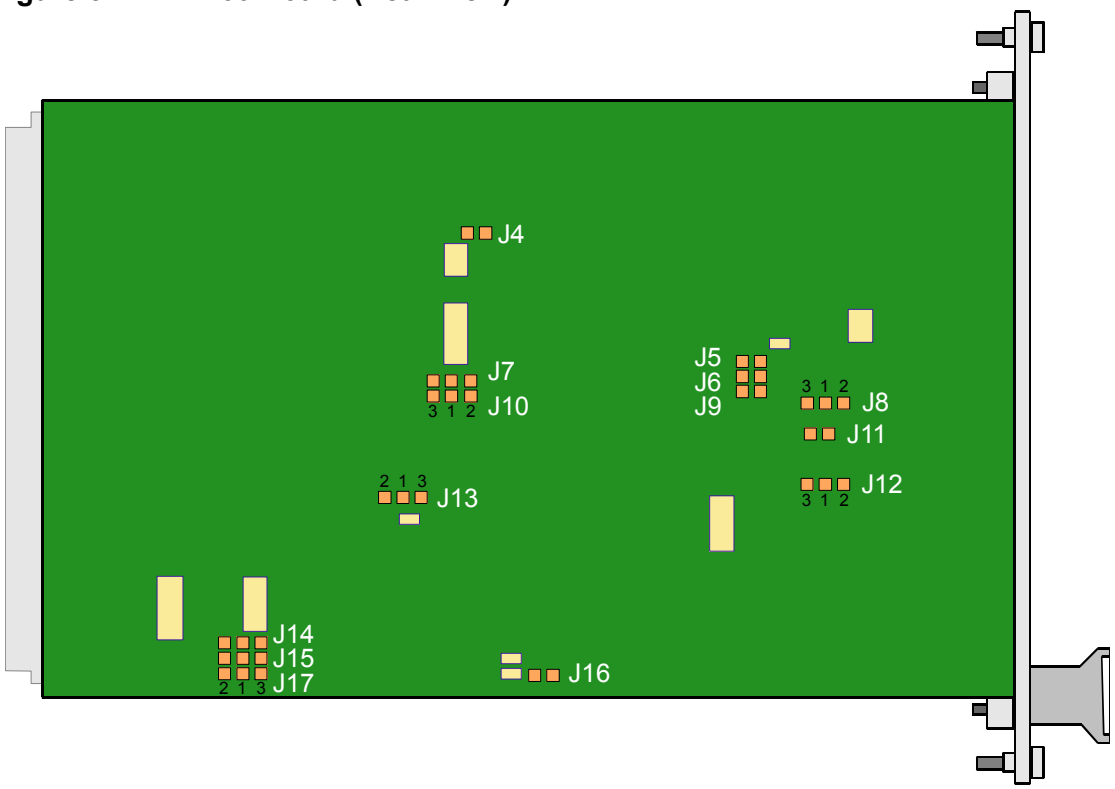


Figure 3-4: VMP60 Board (Rear View)



© 2004 PEP Modular Computers GmbH

1.6 Technical Specifications

Table 2-2: VMP60 Main Specifications

	VMP60	Specifications
Processor and Related	Processor	Motorola Power QUICC MPC860T, 80MHz
	Main Memory	Up to 64 MB of add-on DRAM via DM60x memory modules
	Cache Structure	4 KB instruction cache and 4 KB data cache
	FEC	Independent IEEE 802.3u compliant 10/100baseT Fast Ethernet Controller
	Watchdog	Watchdog generates a system reset
	RTC	Real-time clock, backed up using Gold Caps with the data retention being typically about 6 to 7 days
Peripheral Memory	Onboard (soldered)	
	SRAM	1 MB: backed up using Gold Cap with the data retention being typically about 6 to 7 days
	EEPROM (SPI)	2 kBit: 1 kBit for system and 1 kBit for applications (User EEPROM write protection possible)
	DM60x Memory Module (Plug-in)	
	Flash	Up to 4 MB of Flash memory
	Memory Expansion Socket 1 (Plug-in)	
	EPROM/Flash	Up to 1 MB of EPROM or 512 kB of Flash memory
External Interfaces	Fast Ethernet	10baseT, 100baseTX via RJ45 connector
	Serial Ports	Two RS-232 ports on front panel: TERM: standard RS232 SER0: full modem RS232 Three RS-232/485 full modem ports of MPC860T to CXC : SCC2 : SCC3 : SCC4
	VME	VMEbus expansion connector
	(E)CXC	(Enhanced) Controller eXtension Connector: CXC expansion bus
Internal Interfaces	Memory Sockets	Two 50-pin SMD connectors for one DM60x Memory Module, and one 32-pin DIP socket for EPROM or Flash memory
	Debug Interface	BDM 10-pin twin row connector
	JTAG/ISP	JTAG/ISP 10-pin SMD connected for programming and testing purposes
Indicators / Switches	LED's	3 LED's for indicating system status (two user programmable, one fixed); 3 LED's indicating the status of the Fast Ethernet link
	Switches, Front Panel	Two non-latching push-button type switches for resetting the system or accessing the bootstrap loader during booting of the operating system

© 2004 PEP Modular Computers GmbH

Table 2-2: VMP60 Main Specifications (Continued)

	VMP60	Specifications
General	Mechanical	Conforms with IEEE 1101.1
	Power Consumption	Typ. 6.2 watts at 5V (measured with DM601 Memory Module)
	Temperature Range	Operational: 0ºC to +70ºC Standard -40ºC to +85ºC E2 Storage: -55ºC to +125ºC
	Humidity	0% to 95% non-condensing
	Dimensions	100 mm x 160 mm, 4 HP / 3U VME format
	Board Weight	149 g without add-on memory 178 g with DM600 Memory Module
Software	Operating System Support	Initial boot loader with capability to load VxWorks® operating system or any other realtime operating system. A board support package (BSP) for VxWorks® is available. BSP's for other operating systems available on request.

© 2004 PEP Modular Computers GmbH

1.7 Applied Standards

1.7.1 CE Compliance

The *PEP Modular Computers'* VMEbus and (E)CXC systems comply with the requirements of the following CE-relevant standards:

- Emission EN50081-1
- Immunity EN50082-2
- Electrical Safety EN60950

1.7.2 Mechanical Compliance

- Mechanical Dimensions IEEE 1101.1

1.7.3 Environmental Tests

- Vibration IEC68-2-6
- Random Vibration, Broadband IEC68-2-64 (3U boards)
- Permanent Shock IEC68-2-29
- Single Shock IEC68-2-27

1.8 Related Publications

1.8.1 VMEbus/CXC Systems

- ANSI/VITA: VME64 Draft Specification 1-1994, Rev.1.9
- VITA: CXC Specification, Rev. 2.0
- VITA/*PEP Modular Computers* CXC MPI Draft Specification, Rev. 3.10 (ID 12190)
- *PEP Modular Computers* CXC Reference Manual (ID 16819)

1.8.2 Manufacturers' Component Documentation

- Motorola: MPC860T Power QUICC User's Manual
- EM Microelectronic: V3021 1-Bit Real-Time Clock Data Sheet
- XICOR: X25C02 SPI Serial EEPROM Data Sheet

 © 2004 PEP Modular Computers GmbH

Chapter 2

Functional Description

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

2. Functional Description

The following chapters present more detailed, board level information about the VMP60 PowerPC™ Fast Ethernet - Serial Communications Controller whereby the board components and their basic functionality are discussed in general. Additional information concerning the upgrading of VSBCxxx systems to implement usage of the VMP60 is also provided.

2.1 General Information

The VMP60 is comprised basically of the following:

- Motorola MPC860T PowerQUICC™ Microprocessor
  - PowerPC™ Quad Integrated Communications Controller (PowerQUICC™)
  - One-chip integrated microprocessor and peripheral CPU
  - Embedded PowerPC™ core
  - System Interface Unit (SIU)
  - Communications Processor Module (CPM)
  - Fast Ethernet Controller (FEC)
- System interfaces for:
  - VMEbus
    - 96-pin, male, 3 pinrow, J1/P1 connector
    - VMEbus master interface logic
    - VMEbus slave interface logic
  - (Enhanced) Controller Extension Connector bus ((E)CXC bus)
    - 96-pin, male, 3 pinrow, header style, CXC connector
    - (E)CXC bus interface logic
  - Communications
    - Serial (TERM and SER0 on front panel)
    - Serial (SCC 2, 3, and 4 to (E)CXC bus)
    - Network (Fast Ethernet on front panel)
  - Testing and program development (BDM, JTAG/ISP)
- Memory
  - Main memory (DRAM on DM60x Memory Module)
  - Boot memory (Flash or EPROM on DM60x Memory Module or Socket 1)
  - SRAM semi-non-volatile memory (Gold Cap backup)
    - Onboard power-off data retention
    - VMEbus slave access
  - Serial Peripheral Interface (SPI) (EEPROM)
- Monitor and Control
  - Front panel LED’s (LED1 and LED2 groups)
  - Front panel switches (AB and RST)
  - Jumpers (J1 through J17 and C1)
  - Registers (Board and VME)
  - Realtime clock (RTC)
  - Onboard Bus Error Timer (BERRB)
  - VMEbus Bus Error Timer (BERRV)

© 2004 PEP Modular Computers GmbH

- Watchdog timer (WDOG Timer)
- Reset logic

- Software
  - Board Support Package (BSP)
  - Bootstrap Loader (BSTL)
  - Operating System (OS)

2.1.1 MPC860T PowerQUICC™ Microprocessor CPU

The heart of the VMP60 is the MPC860T PowerQUICC™microprocesor, a PowerPC™ architecture-based derivative of the Motorola MC68360 Quad Integrated Communications Controller (QUICC™). This microprocessor combines an embedded PowerPC™ core with communications peripherals to form a very versatile unit for communications and networking applications.

The following information concerning the MPC860T is provided as a general guide to the user for orientation purposes. More comprehensive information concerning the functioning and programming of the MPC860T can be obtained either from Motorola direct or from their Internet Web site.

2.1.1.1 PowerPC™ Embedded Core

The embedded PowerPC™ core is a 32-bit implementation which supports all PowerPC™ user-level instructions except for floating-point instructions. The core is made up of the following:

- Data cache: 4 kB
- Instruction cache: 4 kB
- Data Memory Management Unit (DMMU)
- Instruction Memory Management Unit (IMMU)
- Integer Unit
- Load/Store Unit (LSU)
- GPR File: 32, 32-bit registers
- Instruction Unit

Additional support is provided for: Power Dissipation Control, Time Base Counter, Decrementer, JTAG, BDM interface, and Clock Multiplier.

2.1.1.2 System Interface Unit (SIU)

The SIU provides very extensive external interfacing and operational control functionality required to support operation of the core and other peripherals. It is responsible for controlling the system start-up, initialization and operation protection, and the external system bus. In addition it provides numerous timing and clock functions, power consumption control, external memory interfacing, system test and development capability (debugging), and JTAG support.

 © 2004 PEP Modular Computers GmbH

2.1.1.3 Communications Processor Module (CPM)

As the second element of the dual-processor design of the MPC860T, the CPM, is responsible for low-level communications functions, providing external data access and distribution (serial communications, Fast Ethernet, and DMA), providing timing and timer facilities, and interrupt control. The CPM consists of a RISC Communications Processor, a comprehensive set of serial I/O peripheral devices, timers and baud rate generators, dual-port RAM for interfacing to the core and DMA, a multi-function DMA controller, and an interrupt controller. The discrete capabilities provided by the CPM for communications tasks free-up the resources of the core making it available for higher priority, higher level tasks. This concept ensures the suitability of the MPC860T for the most demanding of communications applications.

2.1.1.4 Fast Ethernet Controller (FEC)

The FEC is an independently implemented controller within the MPC860T which is responsible for providing all the functionality required to support 10/100baseT Fast Ethernet network communications. It consists of DMA's, FIFO's, 10/100baseT Media Access Control, and the media-independent interface (MII).

2.1.2 System Interfaces

The VMP60 provides interfacing capability for the following system elements:

- VMEbus
- (Enhanced) Controller Extension Connector bus ((E)CXC bus)
- Serial communications
- Fast Ethernet
- Test and program development
    - BDM
    - JTAG

Interfacing to the VMEbus and the (E)CXC bus is accomplished via a programmable logic controller and corresponding board connectors. Test and program development is supported with appropriate board connectors. Serial communications interfacing is achieved via two front panel connectors and their associated transceiver, and the (E)CXC bus connector. The Fast Ethernet interface consists of the Ethernet connector and the Fast Ethernet Transceiver.

2.1.3 Memory

Main memory for the VMP60 is provided by using a single, mezzanine type DM60x Memory Module. This family of memory modules allows for various configurations of up to 64 MB of DRAM. Boot memory is a function of the application requirements. There are two possibilities for providing Boot memory: 1) as Flash memory on a DM60x Memory Module, or 2) through the use of either Flash or EPROM memory installed on Socket 1. For power off data retention and transferring of data to and from the VMP60

 © 2004 PEP Modular Computers GmbH

on the VMEbus in slave mode there is 1 MB SRAM memory available. The SPI interfaces to a 2 kBit EEPROM of which 1 kBit is available for application relevant data.

2.1.4 Monitor and Control

Various monitor and control functions are available for the operation of the VMP60. Two switches (AB and RST) and various LED's are available on the front panel for operator interaction. Pre-operation configuration is accomplished via jumpers. In addition, applications have access to six board specific registers as well as numerous timers, clocks, and reset sources.

2.1.5 Software

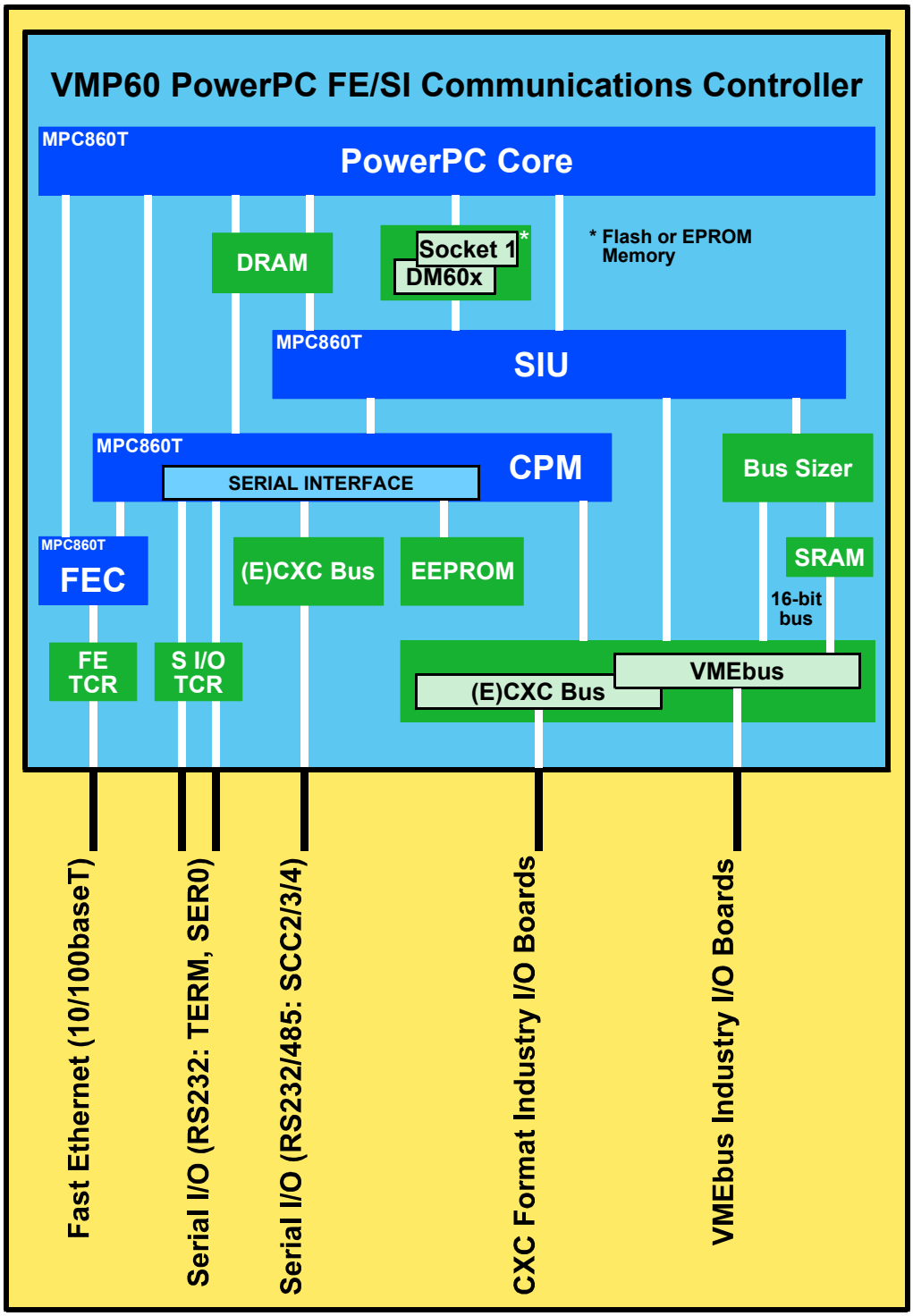
Application software support is provided by a board support package and a bootstrap loader. Currently the VxWorks operating system is supported.

2.1.6 Board-Level Interfacing Diagram

The following figure demonstrates the interfacing structure between the internal processing modules of the MPC860T PowerQUICC™ microprocessor and other major VMP60 system components. Where VMP60 system elements have common interfacing they are grouped into a block. Interfacing common to only one element of a block is indicated with a direct connecting line. The MPC860T modules are shown in blue and the rest of the VMP60 elements in green. The interfacing lines are shown in white where they are onboard and in black for board external interfacing.

 © 2004 PEP Modular Computers GmbH

Figure 2-1: VMP60 Board-Level Interfacing



© 2004 PEP Modular Computers GmbH

2.2 System Interfaces

2.2.1 VMEbus

The VMP60 is designed to provide comprehensive VMEbus support:

- Master functions
- Slave functions
- System controller functions
- Standard VMEbus backplane interfacing

The functionality required of the VMP60 depends on the application as well as the system configuration. When configured as a VMEbus master its functions basically consist of a VMEbus requester and a seven level VMEbus interrupt handler. In addition, if the VMP60 is detected as the system controller within a VMEbus system, the master functions also include VMEbus arbitration, system clock driver, and an IACK daisy chain driver which controls the interrupt acknowledgment. The arbitration function as system controller is single level FAIR on BR3*. Contrastingly, the VMP60's slave functions consist of a programmable board address decoder, dual-ported SRAM access, and a mailbox/doorbell interrupt facility. Rounding up, the VMP60 is equipped with a 96 pin, male, 3 row connector for interfacing to the VMEbus backplane. This connector provides all of the J1/P1 VMEbus standard signals to the VMEbus.

As a VMEbus master, the VMP60 supports A24 and A16 addressing modes in any combination with D16 and D8 data bus width. Table 2-1 indicates the address modifier (AM) codes supported. For the user defined AM codes (1F-18 and 17-10) A24/D16 cycles are generated by the VMP60.

Table 2-1: Address Modifier (AM) Codes

AM Code	Function
3E	A24 supervisory program access
3D	A24 supervisory data access
3A	A24 non-privileged program access
39	A24 non-privileged data access
2D	A16 supervisory access
29	A16 non-privileged access
1F - 18	User defined
17 - 10	User defined

Detection of installation of the VMP60 in slot 1 of the VMEbus backplane (the system controller position) is done automatically. This status is also indicated by the bit 4 setting of the VME_CON_STA_REG register.

In the event that the VMP60 functions as the system controller, it provides the VMEbus system clock (SYSCLK). For this reason, system integrators must ensure that there are no other SYSCLK drivers active in the system.

On the VMP60, the signal, SYSRESET*, is available for routing to the VMEbus. This is accomplished through the setting of jumper, J3. Based on the application requirements, the system integrator must ensure the proper setting of this jumper.

Further, the signals, SYSFAIL* and ACFAIL*, from the VMEbus are processed by the VMP60. SYSFAIL* generates a maskable onboard level 6 interrupt, whereas ACFAIL* generates a non-maskable onboard level 0 interrupt to the MPC860T (highest priority interrupt of the MPC860T PowerQUICC).

For further details regarding the VMEbus and signals refer to the appropriate publications listed in chapter 1.8.

 © 2004 PEP Modular Computers GmbH

Figure 2-2: VMEbus Connector (CON1)

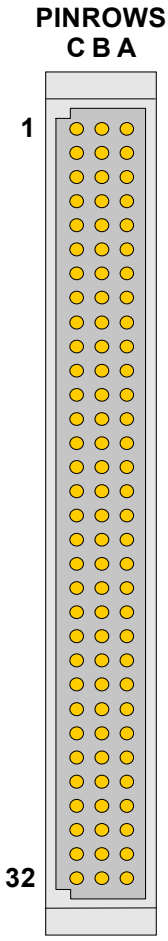


Table 2-2: VMEbus Connector Pinout (CON1)

Pin	Pinrow A	Pinrow B	Pinrow C
1	D00	BBSY* 1)	D08
2	D01	BCLR* 1)	D09
3	D02	ACFAIL* 1)	D10
4	D03	BG0IN* 1)	D11
5	D04	BG0OUT* 1)	D12
6	D05	BG1IN* 1)	D13
7	D06	BG1OUT* 1)	D14
8	D07	BG2IN* 1)	D15
9	GND	BG2OUT* 1)	GND
10	SYSCLK	BG3IN* 1)	SYSFAIL* 1)
11	GND	BG3OUT* 1)	BERR* 1)
12	DS1* 1)	BR0* 1)	SYSRESET* 1)
13	DS0* 1)	BR1* 1)	LWORD* 1)
14	WRITE* 1)	BR2* 1)	AM5
15	GND	BR3* 1)	A23
16	DTACK* 1)	AM0	A22
17	GND	AM1	A21
18	AS* 1)	AM2	A20
19	GND	AM3	A19
20	IACK* 1)	GND	A18
21	IACKIN* 1)	SERA	A17
22	IACKOUT* 1)	SERB	A16
23	AM4	GND	A15
24	A07	IRQ7* 1)	A14
25	A06	IRQ6* 1)	A13
26	A05	IRQ5* 1)	A12
27	A04	IRQ4* 1)	A11
28	A03	IRQ3* 1)	A10
29	A02	IRQ2* 1)	A09
30	A01	IRQ1* 1)	A08
31	-12V	+5VSTDBY	+12V
32	+5V	+5V	+5V

1) By signal assertion is low.

© 2004 PEP Modular Computers GmbH

2.2.2 (Enhanced) Controller Extension Connector Bus ((E)CXC Bus)

The VMP60 is equipped with a 96-pin, male, 3 row header connector (CON2) for providing a (E)CXC bus interface. This connector allows for interfacing to CXC mezzanine modules without requiring a CXC backplane.

Figure 2-3: (E)CXC Bus Connector (CON2)

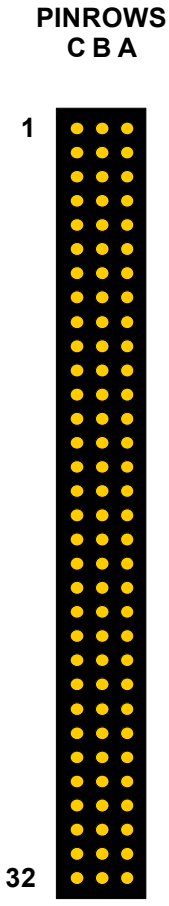


Table 2-3: CXC Connector Pinout (CON2)

Pin	Row A	Row B	Row C
1	_IRQ_1 1)	SER1_RCLK	User-defined
2	_IRQ_2 1)	SER1_TCLK	_DMA_ACK 1)
3	_IRQ_3 1)	GND	_DMA_REQ 1)
4	_IRQ_4 1)	SER1_TXD	User-defined
5	User-defined	SER1_RTS	SER3_TCLK
6	User-defined	GND	SER3_RCLK
7	VCC	SER3_RTS	VCC
8	User-defined	SER3_CD	SER3_TXD
9	User-defined	GND	SER3_RXD
10	User-defined	SER1_RXD	User-defined
11	SER2_DTR	User-defined	SER2_CD
12	SER3_DTR	GND	SER2_RTS
13	SER1_DTR	SER1_CTS	SER2_CTS
14	VCC	SER1_CD	VCC
15	_CS-CXC 1)	GND	SER2_TCLK
16	_AS 1)	SER3_CTS	SER2_RCLK
17	R/_W 1)	_SYSR 1)	SER2_TXD
18	_UDS 1)	GND	SER2_RXD
19	_LDS 1)	_EDTACK 1)	VCC
20	VCC	CXC-CLK	_CS2 1)
21	A1	GND	_CS3 1)
22	A2	_CS0 1)	_CS4 1)
23	A3	_CS1 1)	_CS5 1)
24	A4	GND	_CS6 1)
25	A5	A6	_CS7 1)
26	VCC	A7	VCC
27	D0	GND	D10
28	D1	D6	D11
29	D2	D7	D12
30	D3	GND	D13
31	D4	D8	D14
32	D5	D9	D15

1) By signal assertion is low (by R/_W, only _W).

© 2004 PEP Modular Computers GmbH

CXC and ECXC both contain a 16-bit data bus and eight decoded chip select lines. In total, there are eight control signals (CXC_CS0...CXC_CS7). The main difference between the two VITA standards is the amount of address space available for peripheral devices:

- CXC 7 address lines, 8 x 128 words (256 bytes) per slot (address room available = 0x400 or 1024 bytes per slot with a total of 8192 bytes),
- ECXC 23 address lines, 16 MB per slot, where 16 lines are multiplexed over the data bus (address room available = 0x1000 000 or 16 MB per slot with a total of 128 MB).

Furthermore, the CXC contains a 4-IRQ capability (4 edge sensitive interrupt requests), DMA capability (1 channel, DREQ + DACK), serial ports (3 channels, full MODEM) and a set of parallel port signals. These special CXC functions are based on the MPC860T controller resources.

For general CXC information including addressing, generic pinouts, and a cross reference to the MPC860T CPU pinouts refer to appendix B of this manual. For further details refer to the appropriate publications listed in chapter 1.8.

2.2.3 Communications

Under the aspect of communications control, a major advantage of the MPC860T's Communications Processor Module (CPM) is its compatibility with all important communication standards. This provides the VMP60 user with practically unlimited application configurability. In addition to the onboard interfaces, the VMP60 allows for extensive system enhancement using (E)CXC mezzanine modules. Chapter 2.6, System Enhancement, describes the many possibilities available.

The VMP60 provides support for five communication standards and six communication channels:

- Serial I/O
  - RS232
  - RS485
  - RS422
- Ethernet
  - 10baseT
  - 100baseT
- Channels: SCC1 to SCC4, SMC2, and Fast Ethernet

In the standard configuration, the VMP60 provides two RS232 serial communications interfaces and one 100baseT Fast Ethernet interface on the front panel. In addition, the communication channels: SCC2, SCC3, and SCC4 are routed to the (E)CXC connector CON2. These channels are configurable for RSxxx and 10baseT communications.

The VMP60 mainboard is provided with TxD and RxD signals by the controller's SMC1 and SMC2 channels and supply RS232 interface software handshake (XON/XOFF) capability. Connector CON5 (TERM) is configured for SMC2. SMC1 is not used by the VMP60. The full modem interface provided by connector CON4 (SER0) on the front panel is configured for SCC1 with RxD, TxD, RTS, CTS, CD, and DTR.

 © 2004 PEP Modular Computers GmbH

The following table provides an overview of how the various serial communications channels on the VMP60 are configured.

Table 2-4: VMP60 Serial Interface Channel Configurations

SERIAL I/O CHANNEL	DEDICATED FUNCTION	CONFIGURED VIA	PHYSICAL STANDARDS	CONNECTOR	PORT NAME ON FRONT PANEL
SCC1	Serial I/O	ADM211 transceiver	RS232	CON4	SER0
SCC2	Serial I/O	CXM	RS232 RS485	CON2	not available
SCC3	Serial I/O	CXM	RS232 RS485	CON2	not available
SCC4	Serial I/O	CXM	RS232 RS485	CON2	not available
SMC1 (not used)	-	-	RS232	-	-
SMC2	Terminal Port	ADM211 transceiver	RS232	CON5	TERM

To achieve flexibility, a full modem RS232 interface on the SER0 port, and to make use of the Fast Ethernet controller of the MPC860T the internal routing of the general purpose pins of the MPC860T has been realized in the following way.

Table 2-5: MPC860T Serial Interface Routing

SIGNAL	SCC1	SCC2	SCC3	SCC4	SMC1	SMC2
TXD	PA14	PA12	PA10	PA8	PB25	PB21
RXD	PA15	PA13	PA11	PA9	PB24	PB20
CTS	PC11	PC9	PC7	PC5	-	-
RTS	PB19	PB18	PB17	PB16	-	-
CD	PC10	PC8	PC6	PC4	-	-
DTR	PA7	PB26	PB27	PB31	-	-
RCLK	PA7	PA6	PA2	PA0	-	-
TCLK	PA4	PA5	PA3	PA1	-	-

Notes: General purpose pin PA7 of the MPC860T can be used either as RCLK or DTR, but, however, on the VMP60 it must be configured as DTR because SCC1 is routed to the SER0 front panel interface where no clocks (RCLK or TCLK) are used.

PB20, PB21, PB24 and PB25 are also routed to the CXC connector and connected to user defined pins:

- PB20 to C4
- PB21 to B11
- PB24 to C10
- PB25 to C1

© 2004 PEP Modular Computers GmbH

2.2.3.1 TERM and SER0 Serial I/O Interfaces

The TERM front panel connector (CON5) is provided with TxD and RxD signals via the CPM's SMC2 channel with software handshake (XON/XOFF) capability. It is configured by default for use with a terminal.

The SER0 front panel connector (CON4) is provided with TXD, RXD, RTS, CTS, CD, and DTR signals by the CPM's SCC1 channel. As such it functions as a full modem interface.

The signal/pin assignments of these connectors are shown as follows:

Figure 2-4: TERM (CON5) and SER0 (CON4) Serial Interfaces

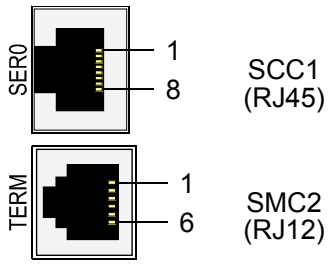


Table 2-6: TERM (CON5) Serial Interface Pinout

PIN	SIGNAL
1	N/C
2	GND
3	TXD
4	RXD
5	N/C
6	N/C

Table 2-7: SER0 (CON4) Serial Interface Pinout

PIN	SIGNAL
1	N/C
2	RTS
3	GND
4	TXD
5	RXD
6	CD
7	CTS
8	DTR

© 2004 PEP Modular Computers GmbH

2.2.3.2 Fast Ethernet Interface

The Fast Ethernet interface is based on the MPC860T on-chip Fast Ethernet Controller (FEC) and the LXT970ATC PHY transceiver chip from LEVEL ONE.

The main features of the Fast Ethernet are as follows:

- Full compliance with IEEE 802.3 standard for 10baseT and 100baseT
- Configurable via MII or external control pins
- Configurable for DTE, repeater, or switch operation
- Full duplex support at both 10 and 100 Mbps
- IEEE 802.3u auto-negotiation support
- Integrated FIFO's and bursting DMA

The signal/pin assignments for this connector are shown as follows:

Figure 2-5: Fast Ethernet Connector (CON3)

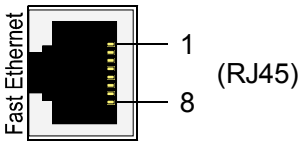


Table 2-8: Fast Ethernet Connector (CON3) Pinout

PIN NUMBER	SIGNAL
1	TX+
2	TX-
3	RX+
4	N/C
5	N/C
6	RX-
7	N/C
8	N/C

© 2004 PEP Modular Computers GmbH

2.2.4 Test and Program Development

2.2.4.1 DEBUG Interface and Pinout

The MPC860T microprocessor has an integrated serial on-chip debug interface which is accessible on the VMP60 through a separate 10-pin, male, dual row connector (2.54 mm pin pitch).

Figure 2-6: DEBUG Connector (CON7)

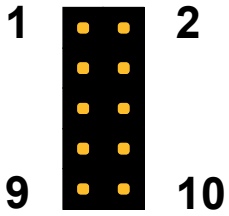


Table 2-9: DEBUG Connector (CON7) Pinout

SIGNAL	PIN	I / O	DESCRIPTION
VFLS0	1	O	MPC860 debug signal (visible history buffer flushes status)
RESETIN	2	I	Debug port reset input (leads to power on reset)
GND	3		
DSCK	4	I	Debug clock
GND	5		
VFLS1	6	O	MPC860 debug signal (visible history buffer flushes status)
RESETOUT	7	O	Debug port reset output (MPC860 is in reset state)
DSDI	8	I	Input serial data
VDD (3.3V)	9		
DSDO	10	O	Output serial data

2.2.4.2 JTAG/ISP Interface and Pinout

A JTAG/ISP interface is provided on the VMP60 for the manufacturer's use (logic programming, JTAG test). This is a SAMTEC 10-pin, male, dual row connector (0.050" pin pitch).

Figure 2-7: JTAG Connector (CON6)

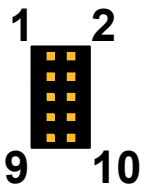


Table 2-10: JTAG Connector (CON6) Pinout

SIGNAL	PIN	PIN	SIGNAL
TCK	1	2	GND
TDO	3	4	VCC
TMS	5	6	GND
OE_CLKS	7	8	N/C
TDI	9	10	GND

© 2004 PEP Modular Computers GmbH

2.3 Memory

The memory architecture of the VMP60 is comprised of four different elements:

- Main memory for application program code and data
- Boot memory for the Boot Strap Loader, Board Support Package (BSP), and the operating system
- SRAM semi-non-volatile memory for onboard power-off data retention and VMEbus slave access
- Serial Peripheral Interface (SPI) memory

Main memory (DRAM) and standard BootRom for the VMP60 is made available through the use of *PEP's* family of DM60x Memory Modules.

In addition to the DM60x Memory Module there is a DIP socket located on the VMP60 mainboard which can accommodate an EPROM/Flash device for use as boot memory. This memory is particularly suited for situations which require separation of the boot-strap loader and the application relevant software.

For backed up program and data storage on the VMP60 there is 1 MB of soldered SRAM available. The SRAM also provides for the exchange and retention of system relevant data either from or to the VME bus when the VMP60 is operated in slave mode. Backup of the SRAM is accomplished using Gold Caps.

System configuration data is stored in an onboard EEPROM accessed through the SPI. A portion of this memory can also be used for application specific configuration data.

2.3.1 DM60x Memory Module Interface

The VMP60 is equipped with an interface for PEP's DM60x series of mezzanine memory modules. Two 50-pin twin row connectors (CON8 and CON9) provide connections for one module. These modules allow for customized memory configurations of up to 64 MB of DRAM and 4 MB of Flash memory depending on the type of memory module installed.

Use of the DM60x Flash memory for boot ROM or as additional Flash memory is primarily a function of the application requirements. The boot device selection, either DM60x Flash or Socket 1 EPROM/Flash, is done through the use of the J1 jumper. Actual area addressing is done through separate CS lines of the CPU which are controlled by the boot decoder logic. Addressing and jumper configuration information is provided in chapter 4, and Appendix A contains further details of the various available DM60x memory modules.

The flexibility achieved by this type of memory configuration allows the user to take advantage of the onboard programming facility to produce low cost upgrades by simply overwriting existing stored data.

2.3.2 Socket 1 Memory Expansion

One 32-pin DIP memory expansion socket is provided for the addition of various types of memory expansion devices (up to 1 MB EPROM or 512 kB Flash) with 8-bit wide data access. Use of this memory is controlled by jumper J1 which allows it to be used either as the bootstrap loader source or as additional memory if required. Jumpers J7 and J10

 © 2004 PEP Modular Computers GmbH

are used to indicate to the system the type of device installed. Addressing and jumper configuration information is provided in chapter 4.

Figure 2-8: Socket 1 Memory Expansion

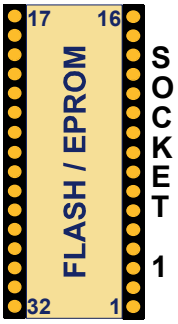


Table 2-11: Memory Devices Supported by Socket 1

DESCRIPTION		
DEVICE	**SPEED**	**SIZE**
EPROM 27C010	120 ns	128 kByte
EPROM 27C020	120 ns	256 kByte
EPROM 27C040	120 ns	512 kByte
EPROM 27C080	120 ns	1 MByte
FLASH 28F010	120 ns	128 kByte
FLASH 28F020	120 ns	256 kByte
FLASH 29F010	120 ns	128 kByte
FLASH 29F040	120 ns	512 kByte

2.3.3 SPI EEPROM

The serial SPI EEPROM is a 2 kBit device which is accessible over the three-wire Interchip SPI interface of the MPC860T. 1 kBit of the EEPROM is reserved for factory data, including Board ID codes, Internet and Ethernet addresses, boot information etc. The rest of the EEPROM is available to the user.

Write protection of the SPI EEPROM is achieved through the use of jumper J4. See chapter 4, Configuration, for details. For usage/configuration of the EEPROM refer to the documentation supplied with the board support package (BSP). For further information on the EEPROM itself refer also to the XICOR X25C02 data sheet.

2.4 Monitor and Control (M/C)

Monitor and Control functions are divided essentially into Pre-operation and Operation. Pre-operation M/C deals with memory configuration aspects, board jumper configurations, and system requirements. Operation M/C covers direct operator interfaces, register handling, and the use of clocks, timers, and counters.

2.4.1 Pre-Operation M/C

Pre-operation M/C is a direct function of the application and the system requirements as a whole. These requirements dictate the operational memory configuration, the jumper settings, and system integration. Memory and its configuration has been addressed in this chapter. The various jumper settings are described in chapter 4. System integration aspects such as board installation and bootstrap loading are covered in chapter 3 and 5 respectively. Overall system integration and compliance with its requirements is beyond the scope of this manual.

© 2004 PEP Modular Computers GmbH

2.4.2 Operation M/C

Operation M/C makes extensive use of the VMP60's various clocks, timers, and control registers. In addition, the VMP60 front panel provides switches and LED's for direct interaction by the operator. The following sections describe the major M/C elements. For details of registers refer to chapter 4, Configuration.

2.4.2.1 System Status Indicators

Six system status indicators divided into two groups of three LED's each are provided on the front panel of the VMP60. The first group (LED1) is dedicated to and controlled by the Ethernet interface whereas the second group (LED2) is application oriented. The table below provides an overview of the functionality associated with these indicators.

Table 2-12: System Status Indicators

G	FP DES.	COLOR	NAME	DESCRIPTION
LED1	ACT	GREEN	LED1AC	This LED indicates that data are being transmitted or received via the Ethernet link. It blinks when there is traffic on the link.
	LNK	GREEN	LED1LN	This LED indicates the integrity of the Ethernet link. When on, the link is established in both directions.
	SPEED	GREEN	LED1SP	This LED indicates the data rate of the Ethernet link. When on the speed is 100 Mbps; off it is 10 Mbps.
LED2	U	GREEN	LED2G	This LED blinks during start-up indicating that the system is being initialized (bootstrap loading in progress). After system initialization has been completed it is on steady and is available for use by the application program. Refer to chapter 4 (BOARD_CON_STA_REG_1 register) for setting of this LED.
	W	YELLOW	LED2Y	This LED is used to indicate the status of Watchdog Timer enabling. When on, the Watchdog Timer is enabled.
	H	RED	LED2R	This LED is available for general use and is application dependent. It is off if not used. Refer to chapter 4 (BOARD_CON_STA_REG_1 register) for setting of this LED.

2.4.2.2 Reset/Abort Switches

On the VMP60 front panel there are two push button switches for interacting with the system: RST for reset and AB for abort.

Pressing of the RST button initiates an immediate hardware reset of the system.

During normal operation pressing of the AB button causes a non-maskable interrupt (NMI) to be generated. Pressing the AB button during system start-up when the green user LED (U) is blinking causes the bootstrap loader to enter the interactive command mode. Commands can then be entered for processing by the bootstrap loader. Refer to chapter 5 for Bootstrap Loader information.

© 2004 PEP Modular Computers GmbH

2.4.2.3 Onboard Bus Monitor

The VMP60 is provided with an onboard bus monitor. A 128 µs timeout timer monitors the cycle lengths of data transfers to and from locations beyond the CPU, including onboard I/O, CXC, dual ported SRAM, and VME arbitration phase. When a timeout occurs, it generates an onboard BERR signal for error termination. This timer is enabled or disabled via the BOARD_CON_STA_REG_1 register which also supplies a timeout status bit in order to identify bus errors generated by the onboard bus error timer.

During VME bus cycles the onboard bus error timer is reset as soon as the VMP60 gains VME bus ownership. Thus the arbitration phase between a VME bus request and the start of a VME bus cycle is monitored by the onboard bus timer. VME bus cycles themselves are monitored by the VME bus monitor

2.4.2.4 VME Bus Monitor

For monitoring of VME bus activity there is a separate VME bus monitor available. This 128 µs timeout timer monitors VME bus data transfer cycle lengths and generates a VME bus BERRV signal for error termination. This timer is enabled or disabled via the EN_BERRV bit in the VME_CON_STA_REG register. A timeout status bit BERRV is provided in the BOARD_CON_STA_REG_1 register to identify bus errors generated by the VME bus monitor.

2.4.2.5 Watchdog Timer

A 512 ms watchdog timer triggers the onboard reset generator at timeout. It is enabled via the BOARD_CON_STA_REG_1 register, and, once enabled, it cannot be reset by software. It must be re-triggered periodically via the corresponding bit in the BOARD_CON_STA_REG_1 register within the timeout period.

‘Watchdog timer activated’ is a status that is indicated by the yellow front panel LED. At timeout the watchdog timer triggers the onboard system reset which also causes a reset to the CXC bus and, if jumper J3 is set, to the VME bus.



Note...

The internal MPC860T watchdog timer is not used on the VMP60. Therefore, it should remain disabled (default setting).

2.4.2.6 Reset Sources

The VMP60 interacts with the following reset sources.

Table 2-13: VMP60 Reset Sources

MPC860T Reset Input	Reset Source
PORESET	Onboard Power Monitor Watchdog Front panel push button: RST Debug port (CON7, pin 2) VMEbus reset (external reset)

© 2004 PEP Modular Computers GmbH

2.4.2.7 Tick Generator

The MPC860T internal Periodic Interrupt Timer is used by the PEP supported real-time operating system as Tick generator.

For further information refer to the Motorola MPC860T User's Manual.

2.4.2.8 Real-Time Clock

The real-time clock, V3021, is a 1-bit device which is accessible over the CS4 of the MPC860T. Its timekeeping features are as follows:

- Seconds, minutes, hours, day of month, month, year, week day, and week number in BCD format
- Leap year and week number correction

For further information regarding the V3021 refer to the EM Microelectronic data sheet.

2.4.2.9 RTC and SRAM Data Retention

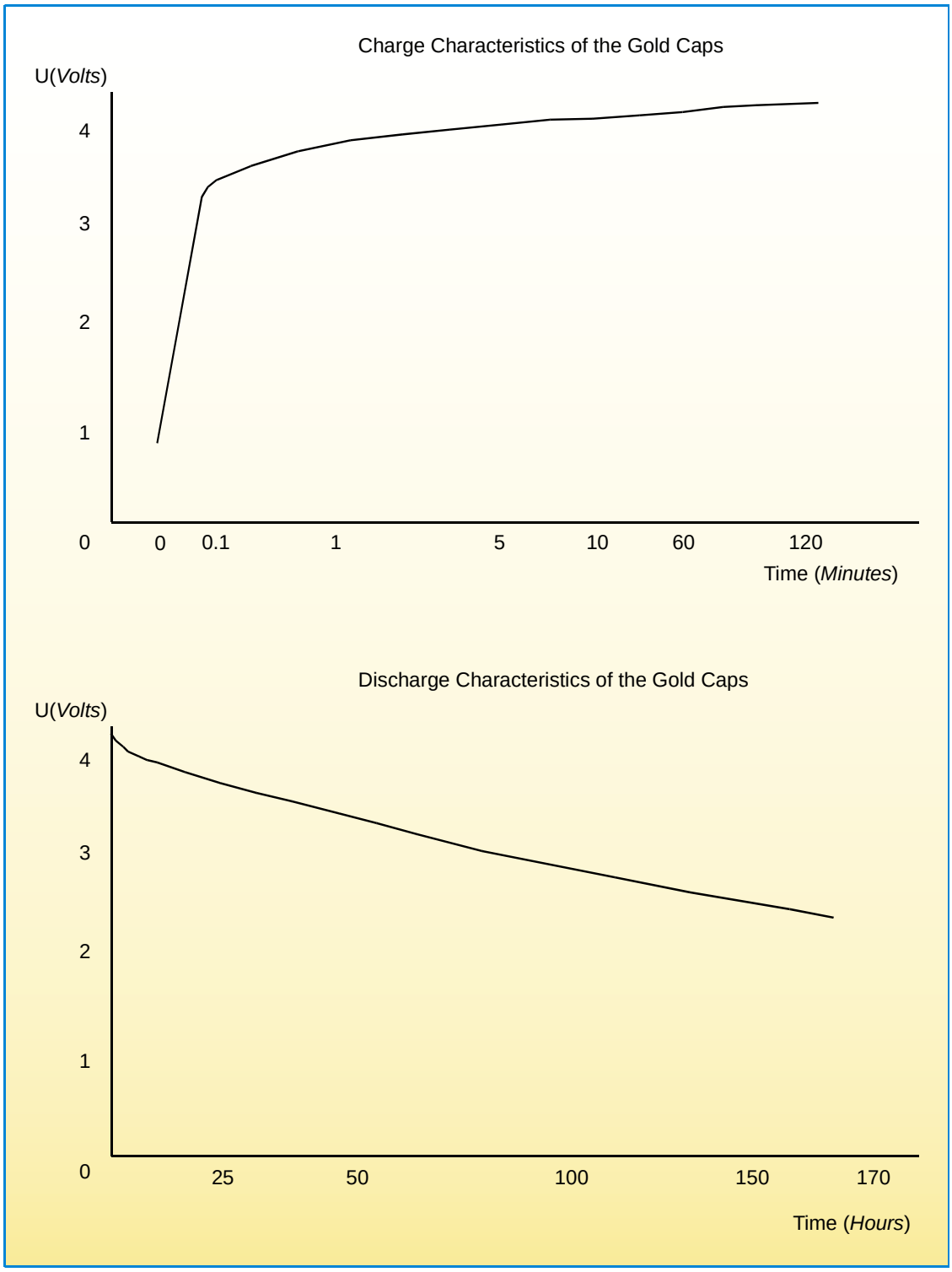
Short-term data retention for RTC and SRAM is achieved through the use of two Gold Caps, each with a value of 0.22 Farad. In contrast to Lithium cells, Gold Caps do not require servicing. This short-term backup is intended for short power failures or for reconfiguring systems. An empty Gold Cap needs approximately three hours to charge up: with backup times dependent on the temperature, memory size, and memory manufacturer tolerances. A well charged Gold Cap provides a minimum of 10 hours backup time.

Laboratory tests at *PEP* indicate a typical backup time of 6 to 7 days for 1MByte SRAM plus RTC (the typical onboard backup current is below 2 µA). The charge and discharge behavior of Gold Caps is documented on the following page.

If a longer data retention time is required, the VMP60 can be equipped optionally with a battery instead of the Gold Caps.

 © 2004 PEP Modular Computers GmbH

Figure 2-9: Gold Cap Charge and Discharge Characteristics



© 2004 PEP Modular Computers GmbH

2.5 VSBCxxx/VMxx-to-VMP60 System Upgrading

The following the porting information is provided for customers who have systems which use the (E)CXC bus interface and require upgrading of their VSBCxxx/VMxx based systems to a VMP60 based system. For the VMEbus interface, the VMEbus implementation is the same as the VSBCxxx/VMxx based systems.

The CXC ports SER1, SER2 and SER3 of the MC68302/MC68360 are equivalent to ports SCC2, SCC3 and SCC4 respectively on the MPC860T controller chip.

With regard to special CXC capabilities, the CXC pinout on the VMP60 has been developed to provide maximum compatibility between the standard CXC functions.

Multifunction pins with incompatible functions with regard to the MC68302/MC68360 and MPC860T, which are called “user-defined” in the generic CXC Specification, are not part of the VMP60 CXC specification.

Although the SMC2 is configured on the mainboard, this port and the unused SMC1 port are also integrated on the CXC because of possible ISDN applications where SMC’s can be integrated and other protocols supported by the MPC860T.

Table 2-14: VSBCxxx/VMxx - VMP60 Porting Information (Sheet 1 of 2)

CXC Function	Pin	MC68302/ MC68360 HW Comp.	MPC860T Port	Comment
IRQ_1	A1	Yes	PC12	—
IRQ_2	A2	Yes	PC13	—
IRQ_3	A3	Yes	PC14	—
IRQ_4	A4	Yes	PC15	—
DMA_ACK	C2	Yes	PB23	—
DMA_REQ	C3	Yes	PC15	—
SER1_RCLK	B1	Yes	PA6	If jumper J8 is set to 1-3. Cannot be used if jumper J8 set to 1-2.
SER1_TCLK	B2	Yes	PA5	—
SER1_TXD	B4	Yes	PA12	—
SER1_RXD	B10	Yes	PA13	—
SER1_RTS	B5	Yes	PB18	—
SER1_DTR	A13	Yes	PB26	—
SER1_CTS	B13	Yes	PC9	—
SER1_CD	B14	Yes	PC8	—
SER2_RCLK	C16	Yes	PA2	If jumper J12 is set to 1-3. Cannot be used if J12 is set to 1-2
SER2_TCLK	C15	Yes	PA3	—
SER2_TXD	C17	Yes	PA10	—
SER2_RXD	C18	Yes	PA11	—

© 2004 PEP Modular Computers GmbH

Table 2-14: VSBCxxx/VMxx - VMP60 Porting Information (Sheet 2 of 2)

CXC Function	Pin	MC68302/ MC68360 HW Comp.	MPC860T Port	Comment
SER2_RTS	C12	Yes	PB17	—
SER2_DTR	A11	Yes	PB27	—
SER2_CTS	C13	Yes	PC7	—
SER2_CD	C11	Yes	PC6	—
SER3_RCLK	C6	Yes	PA0	—
SER3_TCLK	C5	Yes	PA1	—
SER3_TXD	C8	Yes	PA8	—
SER3_RXD	C9	Yes	PA9	—
SER3_RTS	B7	Yes	PB16	—
SER3_DTR	A12	Yes	PB31	—
SER3_CTS	B16	Yes	PC5	—
SER3_CD	B8	Yes	PC4	—
User-Defined	A5	No	PB14	If jumper J13 is set to 1-3. If J13 is set to 1-2, +5VSTDBY is connected to CXC pin A5.
	A6	No / Yes	PB30	SPI Clk: can be used if an ‘SPI SEL’ for example PB14 is used.
	A8	No / Yes	PB29	SPI TxD: can be used if an ‘SPI SEL’ for example PB14 is used.
	A9	No / Yes	PB28	SPI RxD: can be used if an ‘SPI SEL’ for example PB14 is used.
	A10	No	PB22	*See MPC860 User Manual*
	B11	No / Yes	PB21	Used on board SMC2 (Transmit)
	C1	No / Yes	PB25	SMC1 (Transmit)
	C4	No / Yes	PB20	Used on board SMC2 (Receive)
	C10	No / Yes	PB24	SMC1 (Receive)

© 2004 PEP Modular Computers GmbH

2.6 System Enhancement

The comprehensive design of the VMP60 offers many possibilities for system enhancement. Providing full support for both the VME and CXC worlds, *PEP Modular Computers* offers a wide range of complementary industrial I/O products for customizing VMP60 applications.

For example, *PEP*'s CXM-SIO3 Serial Port Extension board is specifically designed to take advantage of the MPC860T's SCC2, SCC3, and SCC4 signal routing to the (E)CXC bus expansion connector. Variants of the CXM-SIO3 which interface directly with this connector thus eliminating the need for a separate CXC backplane can support all of the standard communications protocols except for 100baseT and provide many different types of application interfacing connectors.

This is just one example of the extensive opportunities for employing off-the-shelf system components to satisfy VMP60-based application requirements. For further information on enhancing your system please contact your *PEP* representative.

© 2004 PEP Modular Computers GmbH

Chapter 3

Installation

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

3. Installation

The VMP60 has been designed for easy installation. However, the following standard precautions and installation information and procedures must be observed to ensure proper installation and to preclude damage to the board or injury to personnel.

3.1 Hardware Installation

The product described in this manual can be installed in any available slot of a VMEbus system. However, to function as a system controller it must be installed in slot 1 (far left).

3.1.1 Safety Requirements

The board must be securely fastened to the chassis using the front panel retaining screws to avoid loosening of the board through vibration and to ensure proper grounding and operation of the board.



Caution, Electric Shock Hazard!

Switch off the VME system main power before installing the board. Ensure that there are no other external voltages being applied to system. This includes signals to all other boards within the system. Failure to comply with the above could endanger your life or health and may damage your board or system.



ESD Equipment!

This VME board contains electrostatically sensitive devices. Please observe the following precautions to avoid damage to your board:

- Discharge your clothing before touching the assembly. Tools must be discharged before use.
- Do not touch components, connector pins, or conductive circuits.
- If working at an anti-static workbench with professional discharging equipment, ensure compliance with its usage when handling this product.

© 2004 PEP Modular Computers GmbH

3.1.2 Installation Procedures

To install the board proceed as follows:

- Ensure that the safety requirements indicated above are observed.



Warning!

Failure to comply with the instruction below may cause damage to the board or result in improper system operation. Please refer to chapter 4 for configuration information.

- Ensure that the board is properly configured for operation before installing.



Note...

Care must be taken when applying the procedures below to ensure that when the board is inserted it is not damaged through contact with other boards in the system.

- Install the board in the appropriate slot and ensure that it is properly seated in the backplane (front panel is flush with the rack front).
- Fasten the front panel retaining screws.
- Connect external interfacing cables to the board as required
- Ensure that the board and interfacing cables are properly secured.

3.1.3 Removal Procedures

To remove the board proceed as follows:

- Ensure that the safety requirements indicated above are observed.



Warning!

Care must be taken when applying the procedures below to ensure that when the board is removed it is not damaged through contact with other boards in the system. This applies in particular when a CXC mezzanine board such as the CXM-SIO3 is connected directly to the VMP60 via the CON2 connector. In this case, both boards must be removed together as a unit and not separately.

- Disconnect any interfacing cables that may be connected to the board.
- Loosen the front panel retaining screws.
- Disengage the board from the backplane by pressing down on the front panel handle and pull the board out of the slot.

3.2 Software Installation

Software installation is a function of the Bootstrap Loader and is described in chapter 5 of this manual.

© 2004 PEP Modular Computers GmbH

Chapter 4

Configuration

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

4. Configuration

The following chapters provide information for configuring the VMP60 board jumpers and register addressing and usage.

4.1 Jumper Settings

Prior to installation of the VMP60 the board jumpers must be configured appropriately to ensure proper operation of the board. Three jumpers: J1, J2, and J3 are wire bridge type jumpers, whereas the remaining jumpers, J4 to J17, are solder or 0 ohm resistor type jumpers. Please refer to Figures 1-3 and 1-4 in chapter 1 for the location of the jumpers on the board.

4.1.1 Overview

The following table provides an overview of the VMP60 jumpers including a short description of the default settings.

Table 4-1: VMP60 Jumper Settings Overview

JUMPER	FUNCTION	DEFAULT SETTING	SETTING DESCRIPTION
J1	Boot device select	set	Boot from Flash on DM60x memory PB
J2	(E)CXC mode select	open	CXC mode selected
J3	VME Bus Reset select	set	Onboard Reset generator connected to VME bus
J4	EEPROM write protection	open	Serial EEPROM not write protected
J5	CPU clock speed select	open	Depends on processor type used
J6	CPU clock speed select	open	Depends on processor type used
J7	Installed ROM configuration	open	Depends on the ROM device installed
J8	Baud rate clock	1 - 2	1.843 MHz clock connected to pin PA6 of the MPC860T
J9	CPU clock speed select	set	Depends on processor type used
J10	Installed ROM configuration	open	Depends on the ROM device installed
J11	Layer bridge	set	Factory set, do not change
J12	24 MHz clock	1 - 2	24 MHz clock connected to pin PA2 of MPC860T
J13	CXC pin A5 use select	1 - 3	MPC860T pin PB14 connected to CXC pin A5
J14	SRAM size select	1 - 2	1 MByte of SRAM
J15	SRAM size select	1 - 2	1 MByte of SRAM
J16	Layer bridge	set	Factory set, do not change
J17	SRAM data retention	1 - 3	Data retention on
C1	Connection of signal ground to chassis (ground)	open	Signal ground disconnected from chassis (ground)

© 2004 PEP Modular Computers GmbH

4.1.2 J1 - Boot Device Select

The purpose of the J1 jumper is to select the memory position from which the VMP60 fetches its boot code. It establishes the address location of the DM60x memory module Flash window and the memory expansion socket EPROM/Flash memory if installed.

Table 4-2: J1 - Boot Device Select

JUMPER	SETTING	DESCRIPTION
J1	open	VMP60 fetches boot code from memory expansion socket EPROM or Flash memory
	set	VMP60 fetches boot code from DM60x memory module Flash memory

4.1.3 J2 - (E)CXC Mode Select

The purpose of the J2 jumper is to select either CXC or ECXC mode of operation.

Table 4-3: J2 - (E)CXC Mode Select

JUMPER	SETTING	DESCRIPTION
J2	open	CXC mode operation enabled
	set	ECXC mode operation enabled

4.1.4 J3 - VME Bus Reset

The purpose of the J3 jumper is to make the output of the onboard reset generator available to the VME bus.

Table 4-4: J3 - VME Bus Reset

JUMPER	SETTING	DESCRIPTION
J3	open	Onboard Reset generator disconnected from the VME bus
	set	Onboard Reset generator connected to the VME bus

4.1.5 J4 - EEPROM Write Protection

The purpose of the J4 jumper is to provide write protection for the serial EEPROM if required.

Table 4-5: J4 - EEPROM Write Protection

JUMPER	SETTING	DESCRIPTION
J4	open	Serial EEPROM not write protected
	set	Serial EEPROM write protected

© 2004 PEP Modular Computers GmbH

4.1.6 J5, J6, and J9 - CPU Clock Speed Select

The purpose of the J5, J6, and J9 jumpers is to select the CPU clock speed. As there is currently only one clock speed available, these jumpers are factory configured and may not be changed.

Table 4-6: J5, J6, and J9 - CPU Clock Speed Select

JUMPER	SETTING	DESCRIPTION
J5	open	Factory setting **- do not change -**
	-	-
J6	open	Factory setting **- do not change -**
	-	-
J9	-	-
	set	Factory setting **- do not change -**

4.1.7 J7 and J10 - Installed ROM Configuration

The purpose of the J7 and J10 jumpers is to configure the VMP60 board with respect to the type of memory device installed in the memory expansion socket. If a memory device is installed, care must be taken to ensure the proper setting of jumpers J7 and J10 otherwise improper operation of the VMP60 may result. The J7/J10 jumpers are set to open/open at the factory if no memory device is installed. This is also the default setting.

Table 4-7: J7 and J10 - ROM Type and Size Select

SETTING		DESCRIPTION			
J7	J10	DEVICE	SPEED	ORG.	SIZE
open	open	EPROM 27C010	120 ns	128k x 8	128 kByte
open	open	EPROM 27C020	120 ns	256k x 8	256 kByte
open	1 - 3	EPROM 27C040	120 ns	512k x 8	512 kByte
1 - 2	1 - 3	EPROM 27C080	120 ns	1M x 8	1 MByte
open	open	FLASH 28F010 (read only)	120 ns	128k x 8	128 kByte
open	open	FLASH 28F020 (read only)	120 ns	256k x 8	256 kByte
open	1 - 2	FLASH 29F010 (read & write)	120 ns	128k x 8	128 kByte
1 - 3	1 - 2	FLASH 29F040 (read & write)	120 ns	512k x 8	512 kByte

© 2004 PEP Modular Computers GmbH

4.1.8 J8 - Baud Rate Clock

The purpose of the J8 jumper is to select the source and frequency of the baud rate clock.

Table 4-8: J8 - Baud Rate Clock

JUMPER	SETTING	DESCRIPTION
J8	1 - 2	1.843 MHz clock connected to pin PA6 of the MPC860T
	1 - 3	CXC SER1RCLK connected to MPC860T PA6 pin

4.1.9 J11 and J16 - Layer Bridge

The purpose of the J11 and J16 jumpers is to bridge layers in the VMP60 board. They are set at the factory and may not be changed.

4.1.10 J12 - 24 MHz Clock

The purpose of the J12 jumper is to connect the 24 MHz clock to pin PA2 of the MPC860T.

Table 4-9: J12 - 24 MHz Clock

JUMPER	SETTING	DESCRIPTION
J12	1 - 2	24 MHz clock connected to pin PA2 of the MPC860T
	1 - 3	CXC SER2RCLK connected to MPC860T PA2 pin

4.1.11 J13 - CXC Pin A5 Use Select

The purpose of the J13 jumper is to select the usage of pin A5 of the CXC bus.

Table 4-10: J13 - CXC Pin A5 Use Select

JUMPER	SETTING	DESCRIPTION
J13	1 - 2	+5VSTDBY connected to CXC pin A5
	1 - 3	MPC860T pin PB14 connected to CXC pin A5

© 2004 PEP Modular Computers GmbH

4.1.12 J14 and J15 - SRAM Size Select

The purpose of the J14 and J15 jumpers is to indicate the amount of SRAM installed on the VMP60 board. Both jumpers are required to be set together to properly indicate the amount of SRAM installed.

Table 4-11: J14 and J15 - SRAM Size Select

JUMPER	SETTING	DESCRIPTION
J14	1 - 2	1 MByte SRAM installed
	1 - 3	reserved
J15	1 - 2	1 MByte SRAM installed
	1 - 3	reserved

4.1.13 J17 - SRAM Data Retention

The purpose of the J17 jumper is to provide the ability to retain SRAM data if required.

Table 4-12: J17 - SRAM Data Retention

JUMPER	SETTING	DESCRIPTION
J17	1 - 2	SRAM data retention off
	1 - 3	SRAM data retention on

4.1.14 C1 - Signal Ground Connection to Chassis (Ground)

The purpose of the C1 pads is to allow for various signal ground to chassis (ground) connections to satisfy different EMC/EMI requirements.

Table 4-13: C1 - Signal Ground Connection to Chassis (Ground)

JUMPER	SETTING	DESCRIPTION
C1	open	Signal ground disconnected from chassis (ground)
	set (wire bridge)	Signal ground connected directly to chassis (ground)
	set (capacitor)	Capacitive connection of signal ground to chassis (ground) through the use of a special high voltage capacitor

© 2004 PEP Modular Computers GmbH

4.2 Board Address Map

The following table and figure illustrate the address mapping of the VMP60. Whereas the table provides a detailed description, the figure provides an overall view of the mapping. The address map depends on the configuration of the VMP60 memory expansion socket and the requirements of the application.

Table 4-14: VMP60 Board Address Map

ADDRESS RANGE		MCS*	AREA NAME	SIZE	DESCRIPTION
FROM	TO				
0x0000 0000	0x3FFF FFFF	CS2	DRAM	1 GB	Main memory (DM60x Module)
0x4000 0000	0x7FFF FFFF	CS1	reserved	1 GB	
0x8000 0000	0x81FF FFFF	CS4	VME	32 MB	reserved
0x8200 0000	0x82FF FFFF	CS4	VME	16 MB	User definable AM code (11, 12, 15, 16)
0x8300 0000	0x83FF FFFF	CS4	VME	16 MB	User definable AM code (19, 1A, 1D, 1E)
0x8400 0000	0x84FF FFFF	CS4	VME	16 MB	reserved
0x8500 0000	0x8500 FFFF	CS4	VME	64 kB	Short VME (A16/D16)
0x8501 0000	0x86FF FFFF	CS4	VME	32 MB	reserved
0x8700 0000	0x87FF FFFF	CS4	VME	16 MB	Standard VME (A24/D16)
0x8800 0000	0x88FF FFFF	CS4	VME	16 MB	reserved
0x8900 0000	0x89FF FFFF	CS4	VME	16 MB	IACK cycle generation (0x89XXXXX1)
0x8A00 0000	0x8AFF FFFF	CS4	RTC	16 MB	Realtime clock
0x8B00 0000	0x8BFF FFFF	CS4	BCS-REG	16 MB	Board control/status registers
0x8C00 0000	0x8FFF FFFF	CS4	reserved	64 MB	
0x9000 0000	0x9FFF FFFF	CS0	**J1 open**: SOCKET / **J1 set**: DM60x	256 MB	Boot area
0xA000 0000	0xAFFF FFFF	CS3	**J1 open**: DM60x / **J1 set**: SOCKET	256 MB	Additional memory area
0xB000 0000	0xB7FF FFFF	CS5	(E)CXC	128 MB	(E)CXC memory area
0xB800 0000	0xB8FF FFFF	CS7	SRAM	16 MB	SRAM memory area
0xB900 0000	0xBBFF FFFF		reserved	48 MB	
0xBC00 0000	0xBCFF FFFF	CS6	reserved	16 MB	
0xBD00 0000	0xBD00 FFFF		IMP (MPC-REG)	64 kB	IMMR register, 64 kBytes resolution
0xBD01 0000	0xBFFF FFFF		reserved	48 MB	
0xC000 0000	0xFFFF FFFF	CS1	reserved	1 GB	

* MPC860T chip select

© 2004 PEP Modular Computers GmbH

4.3 Board Registers

The VMP60 has three board registers available to the user which are used to initiate, control, or monitor various onboard functions. The following chapters provide a basic overview of the registers. For more information concerning the functions addressed by these registers refer to the appropriate section in chapter 2.

4.3.1 Overview

The board registers may be accessed through byte-wide read and write operations to the address space 0x8B00 000x.

Table 4-15: Board Registers

REGISTER	ADDRESS	ACCESS	
		READ	WRITE
BOARD_CON_STA_REG1	0x8B00 0007	X	X
BOARD_CON_STA_REG2	0x8B00 0009	X	X (not all bits)
BOARD_IRQ_PEND_REG	0x8B00 000B	X	X (not all bits)

© 2004 PEP Modular Computers GmbH

4.3.2 BOARD_CON_STA_REG_1 Register

The BOARD_CON_STA_REG1 register is a multipurpose register which is used to initiate, control, or monitor various onboard functions.

Table 4-16: BOARD_CON_STA_REG_1 Register

REGISTER NAME		BOARD_CON_STA_REG1						ACCESS	
ADDRESS		0x8B00 0007						R	W
BIT POSITION		MSB 7	6	5	4	3	2	1	0 LSB
CONTENT		res.	BERRV	BERRB	EN_ WDG	TR_ WDG	EN_ BERRB	LED_R	LED_G
DEFAULT		0	0	0	0	1	1	0	1
BIT	**NAME**	**VAL**	**DESCRIPTION**						
0	LED_G	0	Used to switch **off** the green general purpose front panel LED						
		1	Used to switch **on** the green general purpose front panel LED						
1	LED_R	0	Used to switch **off** the red general purpose front panel LED						
		1	Used to switch **on** the red general purpose front panel LED						
2	EN_ BERRB	0	Used to **disable** the on-board bus error timer						
		1	Used to **enable** the on-board bus error timer						
3	TR_ WDG	0	Used to indicate that the watchdog timer has not been re-triggered						
		1	Used to re-trigger the watchdog timer						
4	EN_ WDG	0	Used to indicate that the watchdog timer has not been enabled						
		1	Used to enable the watchdog timer						
5	BERRB	0	Used to indicate that the onboard bus error timer **has not** timed out						
		1	Used to indicate that the onboard bus error timer **has** timed out						
6	BERRV	0	Used to indicate that the VME bus error timer **has not** timed out						
		1	Used to indicate that the VME bus error timer **has** timed out						
7		0	reserved						
		1	reserved						

© 2004 PEP Modular Computers GmbH

4.3.3 BOARD_CON_STA_REG_2 Register

The BOARD_CON_STA_REG2 register is a multipurpose register which is used to initiate, control, or monitor various onboard functions.

Table 4-17: BOARD_CON_STA_REG_2 Register

REGISTER NAME		BOARD_CON_STA_REG2						ACCESS	
ADDRESS		0x8B00 0009						R	W (bits 0 to 6 read only)
BIT POSITION		MSB 7	6	5	4	3	2	1	0 LSB
CONTENT		RMW	ECXC	CPUSP1	CPUSP0	BOARDID3	BOARDID2	BOARDID1	BOARDID0
DEFAULT		0	0	1	1	0	0	0	1
BIT	**NAME**	**VAL**	**DESCRIPTION**						
0	BOARDID0	0	Board Identification: Bits: 3 2 1 0 Setting: 0 0 0 1 = 0x1 = VMP60 board						
		1							
1	BOARDID1	0							
		1							
2	BOARDID2	0							
		1							
3	BOARDID3	0							
		1							
4	CPUSP0	0	CPU (clock) Speed: Bits: 5 4 Setting: 1 1 40 MHz bus clock (80 MHz CPU clock) (This is the only valid setting for these bits.)						
		1							
5	CPUSP1	0							
		1							
6	ECXC	0	Used to indicate that the CXC bus protocol is being used						
		1	Used to indicate that the enhanced CXC ((E)CXC) bus protocol is being used						
7	RMW	0	Used to indicate normal read or write cycles						
		1	Used to generate an unsplittable read modify write (RMW) cycle						

© 2004 PEP Modular Computers GmbH

4.3.4 BOARD_IRQ_PEND_REG Register

The BOARD_IRQ_PEND_REG register is a multipurpose register which is used to initiate, control, or monitor various onboard functions.

Table 4-18: BOARD_IRQ_PEND_REG Register

REGISTER NAME		BOARD_IRQ_PEND_REG						ACCESS	
ADDRESS		0x8B00 000B						R	W (bit 4 read only)
BIT POSITION		MSB 7	6	5	4	3	2	1	0 LSB
CONTENT		EN_DBELL	res.	res.	P_ACF	res.	res.	res.	res.
DEFAULT		0	0	0	0	0	0	0	0
BIT	**NAME**	**VAL**	**DESCRIPTION**						
0		0	reserved						
		1	reserved						
1		0	reserved						
		1	reserved						
2		0	reserved						
		1	reserved						
3		0	reserved						
		1	reserved						
4	P_ACF	0	Used to indicate that a VME ACFAIL IRQ **is not** pending						
		1	Used to indicate that a VME ACFAIL IRQ **is** pending						
5		0	reserved						
		1	reserved						
6		0	reserved						
		1	reserved						
7	EN_DBELL	0	Used to **disable** the VME doorbell IRQ						
		1	Used to **enable** the VME doorbell IRQ						

© 2004 PEP Modular Computers GmbH

4.4 VME Control and Status Registers

The VMP60 has three VME control and status registers available to the user which are used to initiate, control, or monitor various VME bus functions. The following chapters provide a basic overview of the registers. For more information concerning the functions addressed by these registers refer to the appropriated section in chapter 2.

4.4.1 Overview

The VME control and status registers may be accessed through byte-wide read and write operations to the address space 0x8B00 000x.

Table 4-19: VME Control and Status Registers

REGISTER	ADDRESS	ACCESS	
		READ	WRITE
VME_IRQ_MASK_REG	0x8B00 0001	X	X
VME_IRQ_PEND_REG	0x8B00 0003	X	
VME_CON_STA_REG	0x8B00 0005	X	X (not all bits)

© 2004 PEP Modular Computers GmbH

4.4.2 VME_IRQ_MASK_REG Register

The VME_IRQ_MASK_REG register is used to enable or disable VME IRQ's.

Table 4-20: VME_IRQ_MASK_REG

REGISTER NAME		VME_IRQ_MASK_REG						ACCESS	
ADDRESS		0x8B00 0001						R	W
BIT POSITION		MSB 7	6	5	4	3	2	1	0 LSB
CONTENT		ENVI7	EN_VI6	EN_VI5	EN_VI4	EN_VI3	EN_VI2	EN_VI1	EN_SYF
DEFAULT		0	0	0	0	0	0	0	0
BIT	**NAME**	**VAL**	**DESCRIPTION**						
0	EN_SYF	0	Used to **disable** VME SYSFAIL IRQ (VME level 0 IRQ)						
		1	Used to **enable** VME SYSFAIL IRQ (VME level 0 IRQ)						
1	EN_VI1	0	Used to **disable** VME IRQ1						
		1	Used to **enable** VME IRQ1						
2	EN_VI2	0	Used to **disable** VME IRQ2						
		1	Used to **enable** VME IRQ2						
3	EN_VI3	0	Used to **disable** VME IRQ3						
		1	Used to **enable** VME IRQ3						
4	EN_VI4	0	Used to **disable** VME IRQ4						
		1	Used to **enable** VME IRQ4						
5	EN_VI5	0	Used to **disable** VME IRQ5						
		1	Used to **enable** VME IRQ5						
6	EN_VI6	0	Used to **disable** VME IRQ6						
		1	Used to **enable** VME IRQ6						
7	EN_VI7	0	Used to **disable** VME IRQ7						
		1	Used to **enable** VME IRQ7						

© 2004 PEP Modular Computers GmbH

4.4.3 VME_IRQ_PEND_REG Register

The VME_IRQ_PEND_REG register is used to indicate the type or level of VME IRQ pending. This register is necessary for the special VME IACK cycle protocol. It is only set during a VME IACK cycle. The contents of the register always show the level of the last VME IACK cycle.

Table 4-21: VME_IRQ_PEND_REG Register

REGISTER NAME		VME_IRQ_PEND_REG						ACCESS	
ADDRESS		0x8B00 0003						R	
BIT POSITION		MSB 7	6	5	4	3	2	1	0 LSB
CONTENT		P_VI7	P_VI6	P_VI5	P_VI4	P_VI3	P_VI2	P_VI1	P_SYSF
DEFAULT		0	0	0	0	0	0	0	0
BIT	**NAME**	**VAL**	**DESCRIPTION**						
0	P_SYSF	0	Indicates VME SYSFAIL IRQ **is not** pending						
		1	Indicates VME SYSFAIL IRQ **is** pending						
1	P_VI1	0	Indicates VME IRQ1 **is not** pending						
		1	Indicates VME IRQ1 **is** pending						
2	P_VI2	0	Indicates VME IRQ2 **is not** pending						
		1	Indicates VME IRQ2 **is** pending						
3	P_VI3	0	Indicates VME IRQ3 **is not** pending						
		1	Indicates VME IRQ3 **is** pending						
4	P_VI4	0	Indicates VME IRQ4 **is not** pending						
		1	Indicates VME IRQ4 **is** pending						
5	P_VI5	0	Indicates VME IRQ5 **is not** pending						
		1	Indicates VME IRQ5 **is** pending						
6	P_VI6	0	Indicates VME IRQ6 **is not** pending						
		1	Indicates VME IRQ6 **is** pending						
7	P_VI7	0	Indicates VME IRQ7 **is not** pending						
		1	Indicates VME IRQ7 **is** pending						

© 2004 PEP Modular Computers GmbH

4.4.4 VME_CON_STA_REG Register

The VME_CON_STA_REG register is a multipurpose register which is used to initiate, control, or monitor various VME bus functions.

Table 4-22: VME_CON_STA_REG Register

REGISTER NAME		VME_CON_STA_REG						ACCESS	
ADDRESS		0x8B00 0005						R	W (bit 4 read only)
BIT POSITION		MSB 7	6	5	4	3	2	1	0 LSB
CONTENT		P_DBELL	EN_DPR	EN_BERRV	SYSCON	BADR3	BADR2	BADR1	BADR0
DEFAULT		0	0	1	0 or 1	0	0	0	0
BIT	**NAME**	**VAL**	**DESCRIPTION**						
0	BADR0	0	Board's Address Code for VME bus access to the dual ported SRAM: Bits: 3 2 1 0 BA VME Address Setting: 0 0 0 0 = 0x0 = 0x00 00 00 = lowest board address 0 0 0 1 = 0x1 = 0x10 00 00 . . . 1 1 1 1 = 0xF = 0xF0 00 00 = highest board address Equivalent to VME address lines A23 - A20 in 1 MB window; enabled with EN_DPR						
		1							
1	BADR1	0							
		1							
2	BADR2	0							
		1							
3	BADR3	0							
		1							
4	SYSCON	0	Used to indicate that the VMP60 **is not** installed in slot one						
		1	Used to indicate that the VMP60 **is** installed in slot one						
5	EN_BERRV	0	Used to **disable** the VMEbus bus monitor						
		1	Used to **enable** the VMEbus bus monitor						
6	EN_DPR	0	Used to **disable** dual ported SRAM for VME requester						
		1	Used to **enable** dual ported SRAM for VME requester						
7	P_DBELL	0	Used to indicate that a doorbell IRQ **has not** been generated						
		1	Used to indicate that a doorbell IRQ **has** been generated						

© 2004 PEP Modular Computers GmbH

Chapter 5

NetBootLoader

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

5. NetBootLoader

The CPU board is delivered with the NetBootLoader software already programmed into the on-board soldered Flash memory. The NetBootLoader itself is a software utility which initializes the CPU board for operation before turning control over to either an application or to an operator. This software also provides the capability to monitor and control the operation of the NetBootLoader itself, display system status information, to program executable code and data to the Flash memory, and to load and start application software.

To attain full operational capability, the NetBootLoader FLASH must be programmed by the user with application software. Once the application has been programmed to Flash memory, the NetBootLoader will support the complete boot operation. The following chapters describe the functioning of the NetBootLoader and how to program the Flash memory.

5.1 General Operation

Upon power on or a system reset, the NetBootLoader is started. The CPU board is configured for operation and control is either passed to an application or an operator. In the event a valid application has been programmed into the Flash memory and no operator intervention takes place, the application is copied from FLASH into DRAM and control is passed to the application. If the NetBootLoader does not find a valid application or operator intervention has occurred, control is passed to the operator. The operator now has control to determine the system status, make configuration changes, read or program the Flash memory, or to restart or shut down the system.

The operator command interfacing with the NetBootLoader is accomplished either via the TERM serial port or the Ethernet port. During the boot operation a command interpreter is started which allows the operator to input commands to the NetBootLoader. Prior to interfacing via the Ethernet port the network must be configured. This is done via the TERM port.

5.2 NetBootLoader Interfaces

There are four possibilities to interface with the NetBootLoader:

- Via the ABT (Abort) switch
- Via the TERM serial interface
- Via the SER0 serial interface
- Via the Ethernet interface

Gaining access to the NetBootLoader is a function of the contents of the Flash memory and the “BootWaitTime” setting. If there is no valid application programmed into the Flash memory, the boot operation automatically terminates after the CPU board has been initialized and control is passed to the command interpreter. If there is a valid application in the Flash memory the boot operation is delayed according to the setting of the boot wait time. The green user LED (U) on the front panel flashes indicating that the boot operation is in a wait state. During this time the operator may intervene in the boot operation either by pressing the ABT (Abort) switch, entering the “abort” command via the TERM interface, or by performing a successful telnet login via the Ethernet interface. If the operator does not intervene, the boot operation is continued after the boot wait time has been exceeded.

© 2004 PEP Modular Computers GmbH

5.2.1 ABT (Abort) Switch

The ABT switch, located on the CPU board front panel, provides the operator with the ability to directly terminate the boot operation during the boot wait time which is indicated by the “U” LED blinking. This is the sole purpose of the ABT switch during the NetBootLoader operation.

5.2.2 TERM Serial Interface

The TERM serial port is used to provide direct operator interfacing to the NetBootLoader. As soon as the CPU board has been initialized this port is activated and the operator may input commands. During the boot wait time the operator may terminate the boot operation and take control of the NetBootLoader. Once the boot wait time is exceeded the command interpreter is deactivated and the operator no longer has access to the NetBootLoader.

The TERM serial interface may either be directly connected to a terminal device or may interface with a terminal emulator.

5.2.3 SER0 Serial Interface

The SER0 serial port is used to provide the NetBootLoader with the ability to access Motorola S-Records for programming an application to FLASH. No command interpreter is available for this interface.

5.2.4 Ethernet Interface

The Ethernet interface provides the capability of remotely interfacing with the NetBootLoader. Prior to using this interface it is necessary to configure the NetBootLoader network settings. This is accomplished via the TERM interface. Once the network settings have been made, the remote operator has the same capabilities as with the TERM interface. During the boot wait time the operator gains control of the NetBootLoader by logging into it via the Ethernet interface. This causes the boot operation to be terminated and gives control to the remote operator.

The Ethernet interface uses the telnet protocol for operator interfacing with the NetBootLoader. In addition to the operator interface via Ethernet, the NetBootLoader also uses the Ethernet interface for ftp server access.

5.3 NetBootLoader Functions

In addition to initializing the CPU board for operation and the loading and starting of applications, the NetBootLoader provides the following operator monitor and control functions:

- NetBootLoader control
- system status monitoring
- ftp server access
- FLASH reading and programming operations
- Motorola S-Record acquisition

These functions are described in detail in the following chapters.



NOTE ...

The command title (CMD TITLE) is expressed in capital letters and is not the same as the syntax of the command. The command syntax is always written using small letters

 © 2004 PEP Modular Computers GmbH

5.3.1 NetBootLoader Control

The NetBootLoader provides various functions for controlling the operation of the NetBoot-Loader itself as well as the setting of operational parameters. The following table provides an overview of available NetBootLoader control functions.

Table 5-1: NetBootLoader Control Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
ABORT	-	Terminate boot wait	
BW	Boot Wait	Set or display BootWaitTime	
HELP or ?	-	Display online HELP pages	
LOGOUT	-	Terminate telnet session	
NET	-	Set network parameters	Must be set before attempting telnet login
PASSWD	Password	Set telnet password	
PF	Port Format	Set serial port parameters	Used for both TERM and SER0 ports
RS	Reset	Resets system	

5.3.2 System Status Monitoring

The NetBootLoader provides various functions for monitoring the overall status of the system during the operation of the NetBootLoader. The following table provides an overview of avail-able system status monitoring functions.

Table 5-2: System Status Monitoring Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
CHECK	-	Application validation	Verifies validity of user image programmed to FLASH
INFO	-	Display system information	
MD	Memory Display	Display memory contents	Applies to all visible memory
PING	-	Verify network status	
VER	Version	Display version number of NetBootLoader	

© 2004 PEP Modular Computers GmbH

5.3.3 ftp Server Access

The NetBootLoader provides various functions for interfacing with an ftp server. The following table provides an overview of available ftp server functions.

Table 5-3: ftp Server Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
BYE	-	Terminate session with ftp server	
CD	Change Directory	Change ftp server directory	
GET	-	Download a file from ftp server	Only for executable applications. Data buffer is target.
LOGIN	-	Login to ftp server	
LS	List Directory	List ftp server directory	Lists contents of directory.
PUT	-	Upload a file to ftp server	Data buffer is source.
PWD	Print Working Directory	Display current ftp server directory	Lists name of directory

5.3.4 FLASH Operation

The NetBootLoader provides various functions for performing operations with Flash memory. The following table provides an overview of available FLASH operation functions.

Table 5-4: FLASH Operation Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
CLONE	-	Program NetBootLoader to FLASH	Uses data buffer or socket as source
LF	Load FLASH	Program application to FLASH	Uses data buffer as source
SF	Store FLASH	Reads FLASH to data buffer	Uses data buffer as target

5.3.5 Motorola S-Records

The NetBootLoader provides one function for acquiring Motorola S-Records. The following table provides an overview of this function.

Table 5-5: Motorola S-Records Commands

CMD TITLE	ALIAS	FUNCTION	REMARKS
SL	SLoad	Download Motorola S-Records	Uses data buffer as target

© 2004 PEP Modular Computers GmbH

5.4 Operating the NetBootLoader

5.4.1 Initial Setup

The CPU board is delivered with the NetBootLoader already installed in the onboard soldered FLASH and is ready for operation. However, in order for the CPU board to be used in a system, application software must be made available for use. This is accomplished by programming the application also to the onboard soldered Flash memory where the NetBootLoader is located.

Upon initial power up the NetBootLoader is started automatically. As soon as the NetBootLoader has completed initialization of the CPU board, it checks to see if there is a valid application programmed in FLASH and at the same time initiates a command interpreter which the operator can access either via the TERM or telnet interfaces. If there is no valid application in memory, the NetBootLoader terminates the boot operation, and waits for operator intervention. As this is the case when the CPU board is first powered up, the operator now has the opportunity to program an application.

Prior to programming an application it may be necessary to configure the NetBootLoader or perform other functions depending on the user's application development environment or application requirements. Once this has be accomplished and the application has been programmed, the CPU board is ready for operation.

The following chapters provide information on how to set up and operate the NetBootLoader itself, initiation of the telnet interface, and how to program an application to FLASH.

5.4.2 Accessing the NetBootLoader

Initial access to the NetBootLoader can only be achieved via the TERM interface. Prior to using the telnet interface, the Ethernet parameters must be set and this can only be accomplished initially via the TERM interface. Once valid Ethernet parameters and the telnet login password have be set, the telnet interface is available for operation.

Use of the TERM interface requires either a terminal or a terminal emulator. Use of the telnet interface requires a remote telnet login to the NetBootLoader.

Availability of the command interpreter depends on the system status. If there is no valid application programmed, the command interpreter is available as long as the operator requires it. If a valid application is programmed, the command interpreter is only available for the duration of the boot wait time. If the operator requires the command interpreter for a longer time he must terminate the boot operation before the boot wait time is exceeded.

Upon initiation of the command interpreter, a prompt is sent to the TERM interface and commands may be entered. To gain access to the NetBootLoader from a remote location via Ethernet a telnet login must be performed. If the boot wait time has not been exceeded, a telnet login automatically terminates the boot operation and a command prompt is sent to the telnet remote interface.

Once the operator has control of the NetBootLoader, he may perform any required action. To continue with the operation of the CPU board, the system must either be cold started or the operator must issue a "reset" command. In either event, the NetBootLoader is restarted and the boot operation begins anew.

© 2004 PEP Modular Computers GmbH

5.4.3 NetBootLoader Configuration

There are several NetBootLoader commands which provide the operator with the capability to configure specific parameters which are used by the NetBootLoader for interfacing operations. These commands are:

- BW (BootWait)
- NET
- PASSWD
- PF (Port Format)

Default settings are available for all the above commands except for “net” which is dependent on the application environment.

5.4.3.1 BW

This command is used to display or set the actual boot wait time used by the NetBootLoader to delay the boot operation before proceeding with the loading and starting of an application. If this time is set too short it may only be possible to gain access to the NetBootLoader via the “ABT” switch.

The BootWaitTime value is stored in the boot section of the serial EEPROM. This section is validated with a CRC code to avoid the setting of random parameters.



Note ...

If the CRC of the boot section is not valid, changing the BootWaitTime will have no effect because the “bw” command does not validate an invalid CRC. In this case, a default timing of 5 seconds is always used.

To validate an invalid CRC, an operating system utility must be used, or, alternatively, the “-f” option of the “bw” command must be issued.



Warning !!!

Using the “bw -f” command to validate invalid entries may adversely impact the operation of the operating system.

5.4.3.2 NET

This command is used to set or display the parameters for the configuration of the Ethernet interface of the CPU board. The Ethernet interface is only available after these settings have been made. Once these settings have been made, the system must be cold started or reset for them to take effect.

5.4.3.3 PASSWD

This command is used to set the password used by the NetBootLoader for the operation of the telnet interface. No password is required for access from the TERM interface.

© 2004 PEP Modular Computers GmbH

5.4.3.4 PF

This command is used to set the port parameters for the TERM and SER0 serial interfaces only for the current operator session. The next system restart will cause these settings to revert to the default settings of: 9600 Baud, 8 bits per character, 1 stop bit, and no parity. This is done to preclude a system lockout when restarting due to incompatible settings.

5.4.4 telnet Login

A telnet login to the NetBootLoader is only possible during the boot wait time and only after the Ethernet network parameters have been set.

To effect a telnet login the operator performs the standard telnet login procedure during the boot wait time. The NetBootLoader responds by suspending the boot wait and requests a login password. The operator then enters a password. If the password is valid, the boot wait is terminated and the operator can now access the NetBootLoader. If the password is invalid, the telnet login procedure is terminated and the boot operation continues.

In the case of an invalid password, the login procedure may be repeated as often as required within the boot wait time. Once the boot wait time is exceeded, a telnet login is no longer possible.

5.4.5 FLASH Operations

To achieve an operable system for an application, the application software must be programmed to FLASH. The NetBootLoader supports the programming of the application to FLASH. In addition to this, it also supports the updating of the NetBootLoader itself as well as data transfer from the FLASH to the data buffer and from the data buffer to an ftp server. The following chapters provide information on performing the various types of FLASH operations.

5.4.5.1 FLASH Offsets

All FLASH is treated as one uniform FLASH, regardless of the physical addresses of the devices involved. If a DIL FLASH is plugged in the Memory Expansion socket, it is added to the FLASH area as well. All offsets are based from the beginning of the FLASH area. This means that 0x0 is the beginning of the first FLASH bank. The NetBootLoader itself is located at the beginning of the FLASH area and for this reason this area cannot be used for application image programming. To display an overview of the current FLASH organization use the “info” command.

If the application image is an operating system (which is the default case), it must be programmed without an offset. When such an image is programmed to FLASH, the image length and CRC information is also programmed along with the image to FLASH. This information is used by the NetBootLoader to determine the validity of the image during the boot operation. During system startup, a valid image is copied to DRAM address 0x0 and started at offset 0x100 after the boot wait time is exceeded.

If an offset is specified, the image will be programmed exactly at this offset without adding length or CRC information. This option is intended for the storing of configuration information which is required to be located in FLASH.

© 2004 PEP Modular Computers GmbH

5.4.5.2 Programming an Application

The application image itself must be compiled and linked to run from the DRAM base address 0x0 of the CPU. The image must contain executable PPC code at offset 0x100 which is the usual case with ROM/Flash images.

Gaining access to the image for programming to FLASH depends on where it is located. The NetBootLoader can access three different sources for images:

- ftp server
- Motorola S-Records
- memory within the visible address range of the CPU board

The NetBootLoader uses a single data buffer for downloading an image from an ftp server or an image as Motorola S-Records. These images must first be downloaded to the data buffer prior to being programmed to FLASH. An image located within the visible address range of the CPU board is directly accessible for programming.

To access an image located on an ftp server, the “get” command is used. To perform Motorola S-Record acquisition, the “sl” (SLoad) command is used. Once the image is in the data buffer, the FLASH is programmed using the “lf” (Load Flash) command. For an image within visible memory, the “lf” (LoadFlash) command is used to program directly to FLASH.

5.4.5.3 ftp Server Access

To gain access to an application image file stored on an ftp server the Ethernet interface is used. Images are downloaded to the data buffer using the ftp protocol. To use this interface the Ethernet parameters must first be set and then the system must be restarted. During boot wait the operator must gain control of the NetBootLoader and perform an ftp server login. After a successful login, the operator then locates the image file required and downloads it to the data buffer. As with any type of server session, the operator should logout when the session is finished.



Note ...

The commands “get” and “ls” use the same data buffer. Therefore if an “ls” command is issued after a “get” command the data buffer will be overwritten. If an “lf” command follows the “ls” the NetBootLoader refuses to program the overwritten data buffer to the FLASH.

5.4.5.4 Motorola S-Records

The NetBootLoader will also accept Motorola S-Records as an application image. The “sl”’ command accepts S1, S2 and S3 records. Operation is terminated by the appropriate S9, S8 or S7 record. Other types of records are ignored.

The checksum of every record except end records is checked. Bad records are rejected by the NetBootLoader. The address range of every record is also checked. Records which fall outside of the internal buffer are rejected.

The records must be 0-based. This means that it’s address must correspond to the address where they will be loaded in the data buffer relative to its start. If necessary, the base address can be modified with the -o option of the “sl” command.

 © 2004 PEP Modular Computers GmbH



Note ...

If the data buffer is programmed to FLASH without the -o option (program a startable image) the downloaded image is copied to RAM during startup and is executed there. For this reason application images which require to be programmed must start at the address 0x0.

The image must start at the absolute address 0x0 and must contain executable PPC code at the absolute address 0x100. If S1 or S2 record input is preferred, please note that these records only include 16 and 24-bit wide addresses. If no switch to another record type is included it must be ensured that the code is not larger than the address range covered.



Note ...

Neither the “sl” nor “lf” command can be used to program Motorola S-Records to RAM areas.

For accessing the Motorola S-Records, both the TERM and SER0 interfaces can be used. The user LED (“U”) flashes slowly while downloading indicating that the transfer is in progress. The transfer itself may take several minutes to complete.

Ensure that the XON/XOFF protocol is used on the host side. This is a fixed setting and cannot be changed. Additionally, ensure that the host does not stop transmission after a number of lines (e.g. OS-9: use the ‘nopause’ attribute).

The TERM and SER0 serial interface parameters can be modified with the “pf” command.

5.4.6 Updating the NetBootLoader

In addition to programming an application to FLASH, the NetBootLoader itself can be updated. The new version of the image can be made available in one of two ways:

- Via an ftp server
- Via a DIL FLASH device

5.4.6.1 Updating With an Image Loaded Via an ftp Server

The image is downloaded in the same way as an application image (refer to chapter 5.4.5.3). The new version of NetBootLoader image is then programmed using the “clone -n” command.

5.4.6.2 Updating Via a Separate DIL FLASH

The new version of the NetBootLoader image must be programmed to a separate DIL FLASH device (e.g. AM29F040) with an external programmer and then plugged on the Memory Expansion socket of the CPU board. The NetBootLoader must be started and operator must gain control of the NetBootLoader. Either the current NetBootLoader (Boot Jumper removed) or the new NetBootLoader version (BootJumper inserted) may be used for this operation.

The “clone” command is then used to program the new NetBootLoader version to FLASH.

If for some reason the NetBootLoader programmed in the soldered FLASH has been improperly modified (e.g. by a program which accidentally erases or overwrites the chips), this is the only method to reprogram the onboard soldered FLASH. In this case, the boot operation must be started with the Boot Jumper inserted (boot from DIL FLASH).

 © 2004 PEP Modular Computers GmbH

5.4.7 Uploading a FLASH Area

The NetBootLoader also has the possibility to upload certain areas of the FLASH to a host using the Ethernet interface. To use this interface the Ethernet parameters must first be set and then the system must be restarted. During boot wait the operator must gain control of the NetBootLoader and perform an ftp server login. After a successful login, the operator then stores the FLASH area to be uploaded to the local data buffer using the “sf” command. Using the “put” command transfers the contents of the data buffer to the ftp server. As with any type of server session, the operator should logout when the session is finished.

5.5 Plug and Play

On the CPU board the NetBootLoader includes “Plug and Play” functionality. The operating system itself does not include the board initialization code.

5.6 Porting an Operating System to the CPU Board

The image for the absolute address 0x0 should be linked with an entry point at the absolute address 0x100.

Downloaded images are never executed from the FLASH. The programmed image is always downloaded to DRAM, the absolute address 0x0 being downloaded first. There is no configuration option available to amend this process. If it is necessary to relocate the image to another address after download, simply add a small assembly routine at the beginning of the code which will move the image to the correct address.

 © 2004 PEP Modular Computers GmbH

5.7 Commands

The following commands are available with the NetBootLoader. Where an ellipsis (…) appears in the command syntax it means that the command is continued from the previous line. Observe any spaces that may be between the ellipsis and the remainder of the command.

ABORT

FUNCTION:	Terminate the NetBootLoader boot operation
SYNTAX:	**abort**
DESCRIPTION:	This command is used by the operator to to terminate the boot operation during the boot wait time to allow the operator to perform other NetBootLoader operations. To be asserted it must be issued during the boot wait time which is indicated by the flashing green user LED (U) on the CPU board front panel.

BW

FUNCTION:	Set or display the parameters of the boot wait function of the NetBootLoader
SYNTAX:	**bw [<time>| -f]** where: bw command <time> parameter: value: seconds 1, 2, 5, 10, 20, 50 -f option: force CRC update
DESCRIPTION:	The command “bw” displays the parameter “<time>” setting. The parameter “<time>” stipulates the waiting time in seconds that the boot operation is delayed before the application is loaded and started. No values other than these are supported. Bear in mind when setting the boot wait time that the green user LED (U) flashes at the rate of two times a second. Therefore, if the boot wait is set to 1 second the LED will only flash two times.

© 2004 PEP Modular Computers GmbH

BW

DESCRIPTION:	The option “-f” is used to force updating of the CRC value of boot section of the EEPROM. For further information refer to chapter 5.4.3.1.
USAGE:	Display setting of “<time>” parameter COMMAND / RESPONSE: **bw** **WaitTime: 20**
	Set boot wait time to 50 seconds COMMAND / RESPONSE (none): **bw 50**

BYE

FUNCTION:	Terminate an ftp server session
SYNTAX:	**bye**
DESCRIPTION:	An ftp server session which has been established with the command “login” is terminated with the command “bye”.

CD

FUNCTION:	Change the current ftp server directory

© 2004 PEP Modular Computers GmbH

CD

SYNTAX:	**cd <new-path>** where: cd command <new-path> parameter: string new directory path
DESCRIPTION:	If an ftp server session has been established with the “login” command, the command “cd” is used to change the current ftp server directory. The argument “<new-path>” may be an absolute or relative path. The format depends on what the server accepts. For example, UNIX hosts require that the directory names must be entered exactly in the same case.

CHECK

FUNCTION:	Verify validity of application programmed to FLASH
SYNTAX:	**check**
DESCRIPTION:	When an application is programmed to FLASH, a CRC is performed and the results are stored in FLASH along with the application. The “check” command is used to verify that the current application image in FLASH is valid.
USAGE:	Veriy valid application is stored in FLASH COMMAND / RESPONSE: **check** **Check userimage CRC: ok**

© 2004 PEP Modular Computers GmbH

CLONE

FUNCTION:	Program the NetBootLoader to FLASH
SYNTAX:	**`clone [-n]`** where: clone command -n option: program from data buffer
DESCRIPTION:	To update the NetBootLoader itself, the command “clone” is used. The application image source for programming may either be the data buffer or a DIL FLASH installed in the Memory Expansion socket. If the source is the data buffer, the image must first be downloaded to the data buffer from an ftp server. If the image is in the
DESCRIPTION:	DIL FLASH it is programmed directly to the onboard soldered FLASH. To program directly from the DIL FLASH, the command “clone” is used without the “-n” option. The update will be programmed even if the CPU board has been initialized from the DIL FLASH (Boot jumper installed) with the new image. To program from the data buffer, the command “clone -n” is used. In both cases, the new image is checked for validity. If an image is invalid, the update is aborted. Additionally, the operation must be confirmed by typing the word “yes”. Any other or no input will cancel the operation.

© 2004 PEP Modular Computers GmbH

CLONE

USAGE:	Program NetBootLoader from DIL FLASH (normal operation) COMMAND / RESPONSE: **NetBtLd> clone** **clone: Fixup FLASH info from socket** **This will overwrite the current ... NetBootLoader, are you sure? [no] yes** **clone: System transferred; Start again, ... assure that Boot jumper is removed.** **NetBtLd>** Note: When responding to the overwrite query, “yes” must be spelled out. Any other response will terminate the cloning operation.
	Program NetBootLoader from DIL FLASH (image not valid) COMMAND / RESPONSE: **NetBtLd> clone** **clone: Fixup FLASH info from socket** **Image CRC invalid, image is damaged, abort.** **NetBtLd>**

GET

FUNCTION:	Download file from ftp server
SYNTAX:	**get <filename>** where: get command <filename> parameter: string name of image file to be downloaded, or path and name of image file to be downloaded

© 2004 PEP Modular Computers GmbH

GET

DESCRIPTION:	To download a file from the ftp server to the local data buffer, the command “get” is used. A successful ftp server login must be carried out before a file can be downloaded and the file must be in binary format. The argument “<filename>” must refer to an existing and accessible file on the server and the syntax must follow the requirements on the server, e.g. case sensitiveness. The argument may also include a path specification, if the server supports this.

HELP or ?

FUNCTION:	Display online help pages
SYNTAX:	**help** **?**
DESCRIPTION:	This command displays the online help pages. The display of the help text varies between the different CPU’s reflecting their differences. The syntax of every command and a brief description is shown. The display output pauses after every page. The output can be continued with any key. Entering a “.” (period) aborts the help function.

INFO

FUNCTION:	Display system information
SYNTAX:	**info**
DESCRIPTION:	The command “info” is used to display an information summary for the running system. The CPU type, the board type, and the detected FLASH layout are displayed.

© 2004 PEP Modular Computers GmbH

LF

FUNCTION:	Load Flash
SYNTAX:	**`lf [-o[=]<offset> [-k]]`** **`… [-m[=]<adr> -l[=]<len>]`** where: lf command -o option: offset <offset> parameter: value: hexadecimal program to FLASH offset of ... -k option: keep retain surrounding contents -m option: memory (address) <adr> parameter: value: hexadecimal absolute address of image to be programmed -l option: length <len> parameter: value: hexadecimal length of image to be programmed
DESCRIPTION:	Without parameters, the FLASH is programmed using the contents of the data buffer. If no image is available in the data buffer, the FLASH programming is terminated.

© 2004 PEP Modular Computers GmbH

LF

DESCRIPTION:	If no offset option (“-o”) is specified the image is considered to be valid and is therefore added along with CRC and length information. If the CRC is determined to be valid during the next startup, the image is copied to the absolute address 0x0 and started at 0x100 after the boot wait time has been exceeded. Normally, the local data buffer holds the image to be programmed. However, if the “-m” and “-l” parameters are specified, the image is programmed from the absolute address specified. If “<offset>” is specified, the contents are programmed exactly at this offset in FLASH. No length and no CRC information is added. The “-k” option can be specified to prevent deletion of the surrounding FLASH contents. FLASH memory can only be erased sector-wise. If an image is programmed to a certain offset with the “-o” option, at least this sector (and maybe one or more of the following sectors depending on the size of the image) will be erased. The “-k” option can be used to retain the surrounding data, however, this slows down the operation significantly. To achieve fast programming of parameter images without destroying other FLASH contents, the data should be placed at a sector boundary and the sector(s) must not contain any other data or executable images. If organized this way, use of the “-k” option can be avoided. Note: The “lf” command cannot be used to program the NetBootLoader.
USAGE:	Program FLASH from data buffer and add CRC and image length COMMAND / RESPONSE (none): **`lf`**
	Program FLASH from data buffer to offset 0xF4240 COMMAND / RESPONSE (none): **`lf -o=f4240`**
	Program FLASH from visible address at 0x87000000 for length of 0x123456 COMMAND / RESPONSE (none): **`lf -m=87000000 -l=123456`**

© 2004 PEP Modular Computers GmbH

LF

USAGE:	Program FLASH from data buffer to offset 0xF4240 and retain adjacent FLASH contents COMMAND / RESPONSE (none): **`lf -o=f4240 -k`**

LOGIN

FUNCTION:	Initiate ftp server session
SYNTAX:	**`login <ip-of-host> <username> [<password>]`** where: login — command <ip-of-host> — parameter: value: numerical string IP address of host: nnn.nnn.nnn.nnn <username> — parameter: value: string ftp server “username” <password> — parameter: value: string user’s password
DESCRIPTION:	The command “login” is used to establish an ftp server session. The “<ip-of-host>” must be specified as four numbers separated by single dots. The “<password>” parameter is not necessary if the server does not request one.
USAGE:	Initiate ftp server session COMMAND / RESPONSE: **`login 192.168.47.12 johndoe mypassword`** (Response is dependent on the server accessed)

LOGOUT

FUNCTION:	Terminate telnet session with NetBootLoader

© 2004 PEP Modular Computers GmbH

LOGOUT

SYNTAX:	**logout**
DESCRIPTION:	A remote telnet session will be terminated with the command “logout”. No application is loaded and started if the session is terminated with “logout”. The NetBootLoader waits for a new session to be initiated or for a command entry from the serial console.

LS

FUNCTION:	Display listing of the current ftp server directory
SYNTAX:	**ls**
DESCRIPTION:	To display a listing of the current ftp server directory the command “ls” is used. This command downloads the listing to the data buffer and then the listing is displayed. Any previously loaded image in the data buffer is overwritten. If an attempt is then made to program the FLASH after the “ls” command has been issued it will fail.

MD

FUNCTION:	Display visible memory
SYNTAX:	**md [<adr>]** where: md command <adr> parameter: value: hexadecimal starting address of a visible memory area
DESCRIPTION:	To display a visible memory area the command “md” is used. The first time the command “md” is issued, visible memory contents starting at the address 0x0 are displayed if no “<adr>” parameter is used. If issued again without the “<adr>” parameter, the display starts with the end address of the previous display. Data is displayed as hexadecimal 32-bit words and as ASCII dump.

© 2004 PEP Modular Computers GmbH

NET

FUNCTION:	Set or display the parameters for the Ethernet interface
SYNTAX:	**net [<ip-addr>][-netmask <netmask>]** **…[-gw <gateway>][-f]** where: net — command <ip-addr> — parameter: value: numerical string IP address of CPU board: nnn.nnn.nnn.nnn -netmask — option: netmask <netmask> — parameter: value: numerical string netmask of CPU board: nnn.nnn.nnn.nnn -gw — option: gateway <gateway> — parameter: value: numerical string gateway address for network: nnn.nnn.nnn.nnn -f — option: force CRC update
DESCRIPTION:	To set or display the parameters of the Ethernet interface the command “net” is used. Initially the CPU board does not have a valid Ethernet interface configuration, and, therefore, this interface is inoperable. The initial configuration must be done from the TERM interface using the command “net ... -f”. Using the “-f” option forces a CRC to be performed and stored along with the other configuration parameters in the serial EEPROM. Once the initialization of the Ethernet interface is done, the CPU board must be restarted for the parameters to take effect. Later changes to the parameters do not require the use of the “-f” option to force a CRC. This is done automatically. Only in the event that the Ethernet interface does not properly initialize, may it be necessary to re-enter the parameters using the “-f” option.

© 2004 PEP Modular Computers GmbH

PASSWD

FUNCTION:	Set the telnet password
SYNTAX:	**passwd [-f | -d]** where: passwd command -f option: if password is not known -d option: disable disable telnet login (remote access)
DESCRIPTION:	To set the password for telnet sessions with the NetBootLoader the command “passwd” is used. This command is interactive, meaning that after it is issued, the NetBootLoader responds with an appropriate request to the operator which must be properly acknowledged or the operation fails (refer to USAGE below). To set the password in the event it is unknown, use the option “-f”. This is can only be accomplished from the TERM interface and not from the Ethernet interface. With the option “-d”, the remote telnet login can be disabled by invalidating the password.
USAGE:	Set password COMMAND / RESPONSE: **NetBtLd> passwd** **Old Password: ***** **New Password: ***** **Type again : ***** **NetBtLd>** (The old password must be known)
	Set password when the old password is not known COMMAND / RESPONSE: **NetBtLd> passwd** **New Password: ***** **Type again : ***** **NetBtLd>**

© 2004 PEP Modular Computers GmbH

PF

FUNCTION:	Set or display the serial port parameters (format)
SYNTAX:	**pf [<port> [<baud>][/[<bitschar>]** **…[/[<parity>][/<stops>]]]]** where: pf command <port> parameter: string: “term” or “ser0” defines serial port to be configured <baud> parameter: value: numeric: “50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 115200” defines the baud rate for the port <bitschar> parameter: value: numeric: “7” or “8” defines the number of bits per character <parity> parameter: string: “n” (none), “o” (odd), “e” (even) defines parity to be used <stops> parameter: value: number: “1”, “2” defines number of stop bits
DESCRIPTION:	To set or display the operational parameters for the available serial interfaces the command “pf” is used. At startup the settings for the “TERM” and “SER0” interfaces are always set to the default values (9600/8/n/1). This is to avoid a possible system lockout. If other settings are required during operation of the NetBootLoader they may be made. If changes are made, it must be ensured that corresponding parameters are used for the operator consle. Issuing this command without parameters being specified will display the current serial port settings. Syntax-wise, no spaces are permitted between the parameters and they must be separated with a slash. Not all parameters must be specified, but the “/” characters must be present to distinguish the different parameters from each other. The sequence can be aborted after every option.

© 2004 PEP Modular Computers GmbH

PF

USAGE:	Set “TERM” to 300 Baud, 7 Bits/char, odd parity, and 2 stop bits COMMAND / RESPONSE (none): **pf term 300/7/o/2**
	Set the bits per character parameter of “SER0” to 7 COMMAND / RESPONSE (none): **pf ser0 //7**
	Set the stop bits parameter of “SER0” to 2 COMMAND / RESPONSE (none): **pf ser0 ///2**

© 2004 PEP Modular Computers GmbH

PING

FUNCTION:	Verify operability of the Ethernet interface
SYNTAX:	**ping <ip_addr> [-c <count>] [-s <size>]** **… [-w <wait>]** where: ping — command <ip-addr> — parameter: value: numerical string IP address of target: nnn.nnn.nnn.nnn -c — option: count <count> — parameter: value: numeric: “[n ...]n” number of packets to send -s — option: size <size> — parameter: value: numeric: “[n ...]n”: bytes size of packet to send -w — option: wait <wait> — parameter: value: numeric: “[n ...]n”: seconds wait time between packets
DESCRIPTION:	To verify the operational status of the Ethernet interface the command “ping” is used. This command tests the network connection and target server’s ability to respond. If no other parameters are specified, four requests will be sent. This can be changed with the parameter “-c”. The typical size of a ping packet can be changed with the parameter “-s” and the time between requests, which is typically one second, can be changed with the parameter “-w”. Reponses to the “ping” command are dependent on the performance of the network.
USAGE:	Send four packets COMMAND / RESPONSE: **ping 192.192.158.7**
	Send ten packets, 100 bytes long, and wait two seconds between packets COMMAND / RESPONSE: **ping 192.192.158.7 -c 10 -s 100 -w 2**

© 2004 PEP Modular Computers GmbH

PUT

FUNCTION:	Upload contents of the data buffer to the ftp server.
SYNTAX:	**put <filename>** where: put command <filename> parameter: string file name to be used for contents of data buffer to be uploaded
DESCRIPTION:	To upload the contents of the data buffer to a file on an ftp server, the command “put” is used. The file indicated by the parameter “<filename>” is created on the server. In the event that a file with this name already exists, its contents will be overwritten.

PWD

FUNCTION:	Display the current ftp server directory.
SYNTAX:	**pwd**
DESCRIPTION:	If a ftp connection has been established with the “login” command, the command “pwd” is used to display the complete path of the current directory on the ftp server.

RS

FUNCTION:	Reset the system
SYNTAX:	**rs**

© 2004 PEP Modular Computers GmbH

RS

DESCRIPTION:	To permit the operator to force a restart of the system, the command “rs” is used. This command terminates the NetBootLoader command interpreter and resets the entire system, generating a system reset with the onboard watchdog. If this command is issued over a remote telnet connection, the telnet session is terminated prior to the generation of the reset.

SF

FUNCTION:	Store FLASH contents to data buffer
SYNTAX:	**sf -o[=]<offset> -l[=]<length>** where: sf command -o option: offset <offset> parameter: value: hexadecimal relative offset to start of FLASH contents to be stored to the data buffer -l option: length <length> parameter: value: hexadecimal length of FLASH contents to be stored to the data buffer
DESCRIPTION:	With the command “sf” a selected portion of the FLASH contents may be copied to the local data buffer, e.g. for a subsequent upload to the ftp server with the “put” command. The “<offset>” parameter refers to the relative offset within the FLASH area similar to the “lf” command. The parameter “<length>” specifies the length to store.
USAGE:	Store 64 kB of FLASH contents to the data buffer beginning at an offset of 1 MB COMMAND / RESPONSE (none): **sf -o=100000 -l=10000**

© 2004 PEP Modular Computers GmbH

SL

FUNCTION:	Download Motorola S-Records to data buffer
SYNTAX:	**sl [-o[=]<offset>] [-u]** where: sl command -o option: offset <offset> parameter: value: hexadecimal: unsigned offset to be subtracted from each record's address -u option: upper
DESCRIPTION:	With the command “sl” Motorola S-Records are downloaded to the data buffer and the record addresses modified accordingly as required for DRAM operation (for copying to 0x0). The “<offset>” parameter may be used to change the record base to 0x0. The “-u” option selects the SER0 interface as source for the S-Records.
USAGE:	Download S-Records to data buffer and reduce each record’s address by 0x10000. COMMAND / RESPONSE (none): **sl -o=10000**

VER

FUNCTION:	Display version number
SYNTAX:	**ver**
DESCRIPTION:	The command “ver” displays the actual version number of the NetBootLoader.

© 2004 PEP Modular Computers GmbH

Appendix A

DM60x Memory Modules

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

A. DM60x Memory Modules

A.1 General

The DM60x Memory Modules described herein provide main memory capability for the storage of program code and data either in DRAM or Flash memory. Various configurations of DRAM and Flash memory as indicated in the table below are available for a wide variety of PEP CPU boards. All configurations have 32-bit access and a maximum address range of 64 MB. In addition jumpers are available for providing write protection.

Table A-1: DM60x Memory Module Types and Configurations

Type	Memory Configuration	
	DRAM	FLASH
DM600	4 MB	2 MB
		4 MB
DM601	16 MB	2 MB
		4 MB
DM602	1 MB	2 MB
DM603	32 MB	2 MB
		4 MB
DM604	8 MB	2 MB
		4 MB
DM605	64 MB	2 MB
		4 MB

© 2004 PEP Modular Computers GmbH

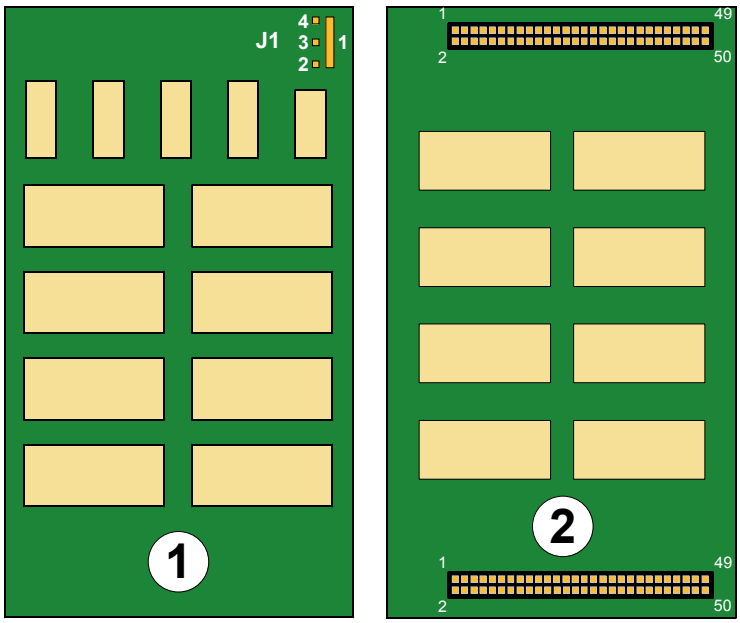
A.2 DM600

The DM600 memory module is fitted with:

- DRAM: 4 MB
- Flash Memory: 2 or 4 MB

A.2.1 Board Layout and Jumper Location

Figure A-1: DM600 Memory Module



Legend:

1. Flash Memory
2. DRAM

A.2.2 Jumper Description and Flash Addresses

Table A-2: Jumper: J1 Settings and Flash Memory Address Ranges

Setting	Description	2 MB Flash	4 MB Flash
Open	All Flash EPROM's write protected		
1 - 2	*No Protection*		
1 - 3	Flash Bank 1: Write protected Flash Bank 1: Address Range		Upper 2 MB 0x(9/A)0200000 - 0x(9/A)03FFFFF
1 - 4	Flash Bank 0: Write protected Flash Bank 0: Address Range	0x(9/A)0000000 - 0x(9/A)01FFFFF	Lower 2 MB 0x(9/A)0000000 - 0x(9/A)01FFFFF

Default settings are in italics.

© 2004 PEP Modular Computers GmbH

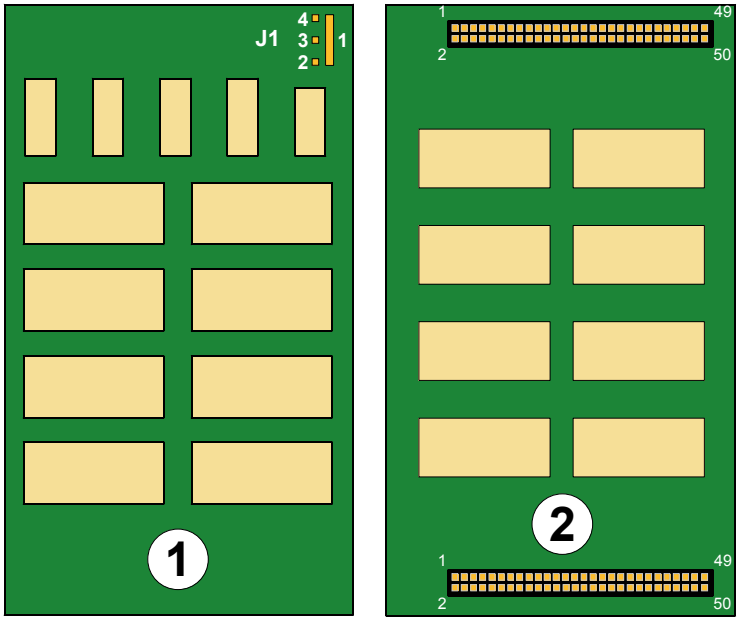
A.3 DM601

The DM601 memory module is fitted with:

- DRAM: 16 MB
- Flash Memory: 2 or 4 MB

A.3.1 Board Layout and Jumper Location

Figure A-2: DM601 Memory Module



Legend:

1. Flash Memory
2. DRAM

A.3.2 Jumper Description and Flash Addresses

Table A-3: Jumper: J1 Settings and Flash Memory Address Ranges

Setting	Description	2MB Flash	4MB Flash
Open	All Flash EPROM's write protected		
1 - 2	*No Protection*		
1 - 3	Flash Bank 1: Write protected Flash Bank 1: Address Range		Upper 2 MB 0x(9/A)0200000 - 0x(9/A)03FFFFF
1 - 4	Flash Bank 0: Write protected Flash Bank 0: Address Range	 0x(9/A)0000000 - 0x(9/A)01FFFFF	Lower 2 MB 0x(9/A)0000000 - 0x(9/A)01FFFFF

Default settings are in italics.

© 2004 PEP Modular Computers GmbH

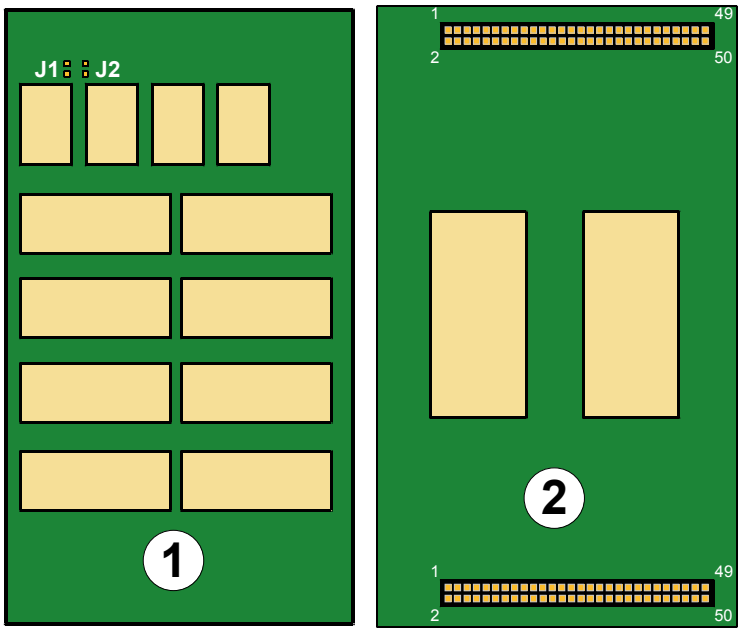
A.4 DM602

The DM602 memory module is fitted with:

- DRAM: 1 MB
- Flash Memory: 2 MB

A.4.1 Board Layout and Jumper Location

Figure A-3: DM602 Memory Module



Legend:

1. Flash Memory
2. DRAM

A.4.2 Jumper Description and Flash Addresses

Table A-4: Jumpers: J1 and J2 Settings and Flash Memory Address Ranges

Setting		Description	2MB Flash
J1	*Set*	*No Protection*	
	Open	Flash Bank 1: Write protected Flash Bank 1: Address Range	
J2	*Set*	*No Protection*	
	Open	Flash Bank 0: Write protected Flash Bank 0: Address Range	0x(9/A)0000000 - 0x(9/A)01FFFFF

Default settings are in italics.

© 2004 PEP Modular Computers GmbH

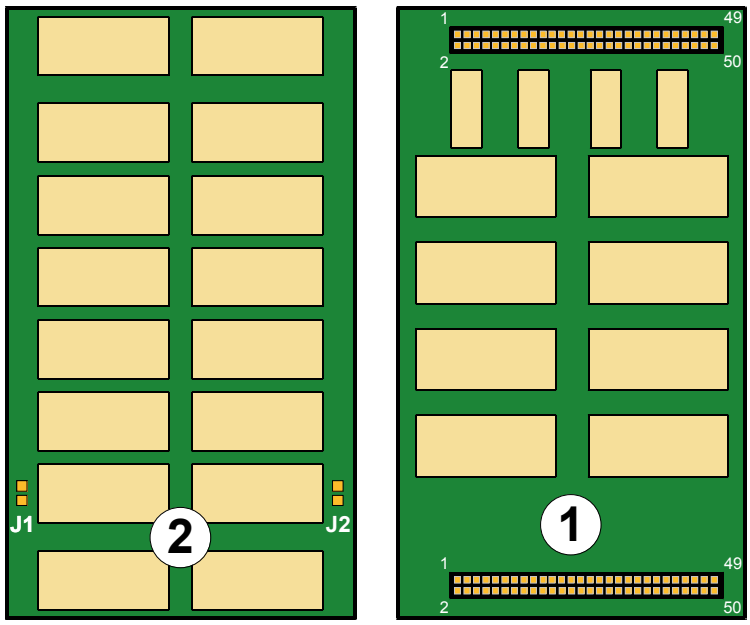
A.5 DM603

The DM603 memory module is fitted with:

- DRAM: 32 MB
- Flash Memory: 2 or 4 MB

A.5.1 Board Layout and Jumper Location

Figure A-4: DM603 Memory Module



Legend:

1. Flash Memory
2. DRAM

A.5.2 Jumper Description and Flash Addresses

Table A-5: Jumpers: J1 and J2 Settings and Flash Memory Address Ranges

Setting		Description	2MB Flash	4MB Flash
J1	*Set*	*No Protection*		
	Open	Flash Bank 0: Write protected		Lower 2 MB
		Flash Bank 0: Address Range	0x(9/A)0000000 - 0x(9/A)01FFFFF	0x(9/A)0000000 - 0x(9/A)01FFFFF
J2	*Set*	*No Protection*		
	Open	Flash Bank 1: Write protected		Upper 2 MB
		Flash Bank 1: Address Range		0x(9/A)0200000 - 0x(9/A)03FFFFF

Default settings are in italics.

© 2004 PEP Modular Computers GmbH

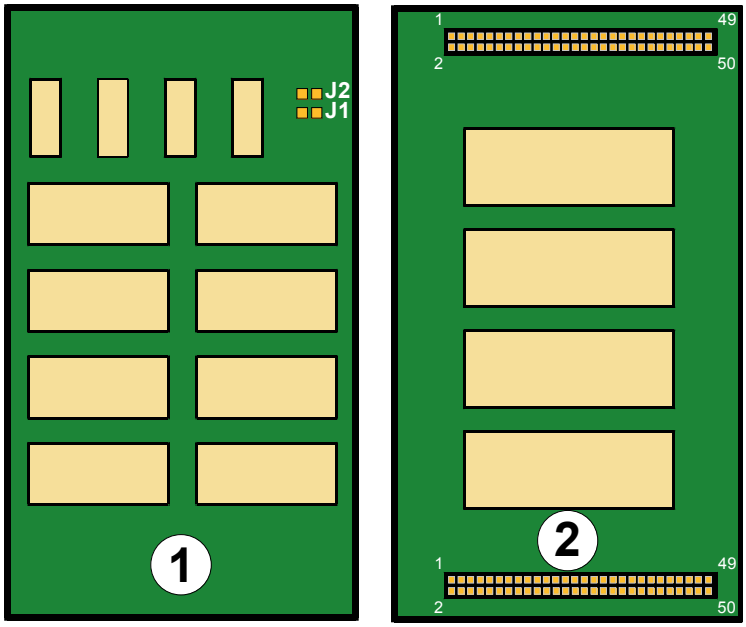
A.6 DM604

The DM604 memory module is fitted with:

- DRAM: 8 MB
- Flash Memory: 2 or 4 MB

A.6.1 Board Layout and Jumper Location

Figure A-5:DM604 Memory Module



Legend:

1. Flash Memory
2. DRAM

A.6.2 Jumper Description and Flash Addresses

Table A-6: Jumpers: J1 and J2 Settings and Flash Memory Address Ranges

Setting		Description	2MB Flash	4MB Flash
J1	*Set*	*No Protection*		
	Open	Flash Bank 0: Write protected		Lower 2 MB
		Flash Bank 0: Address Range	0x(9/A)0000000 - 0x(9/A)01FFFFF	0x(9/A)0000000 - 0x(9/A)01FFFFF
J2	*Set*	*No Protection*		
	Open	Flash Bank 1: Write protected		Upper 2 MB
		Flash Bank 1: Address Range		0x(9/A)0200000 - 0x(9/A)03FFFFF

Default settings are in italics.

© 2004 PEP Modular Computers GmbH

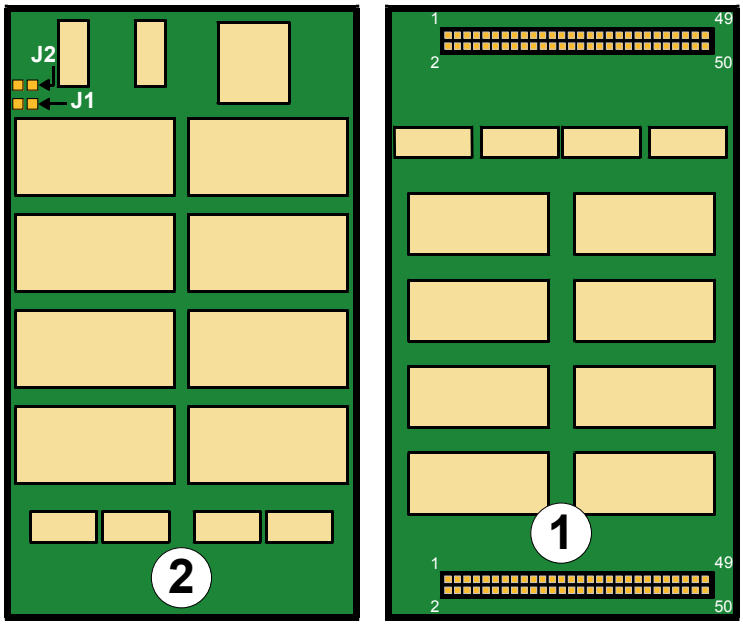
A.7 DM605

The DM605 memory module is fitted with:

- DRAM: 64 MB
- Flash Memory: 2 or 4 MB

A.7.1 Board Layout and Jumper Location

Figure A-6: DM605 Memory Module



Legend:

1. Flash Memory
2. DRAM

A.7.2 Jumper Description and Flash Addresses

Table A-7: Jumpers: J1 and J2 Settings and Flash Memory Address Ranges

Setting		Description	2MB Flash	4MB Flash
J1	*Set*	*No Protection*		
	Open	Flash Bank 0: Write protected		Lower 2 MB
		Flash Bank 0: Address Range	0x(9/A)0000000 - 0x(9/A)01FFFFF	0x(9/A)0000000 - 0x(9/A)01FFFFF
J2	*Set*	*No Protection*		
	Open	Flash Bank 1: Write protected		Upper 2 MB
		Flash Bank 1: Address Range		0x(9/A)0200000 - 0x(9/A)03FFFFF

Default settings are in italics.

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

 © 2004 PEP Modular Computers GmbH

Appendix B

(E)CXC

© 2004 PEP Modular Computers GmbH

This page was intentionally left blank.

© 2004 PEP Modular Computers GmbH

B. (E)CXC

The Controller eXtension Connector (CXC) is a local mezzanine interface. The CXC contains a 16-bit data bus, seven address lines, and eight decoded chip select lines. Addressing of the individual CXC backplane slots is achieved via the CS5 line of the MPC860T processor. The eight CXC chip selects (_CS0 to _CS7) make it possible to select the individual backplane slots.

Additionally, the CXC provides four IRQ's, DMA capability, three serial ports, and one set of parallel port signals.

B.1 (E)CXC Address Ranges

The following tables provide addressing information for both the CXC and ECXC operational modes for the CPU boards indicated. In the case of CXC mode operation, the address range is 256 bytes, whereas, in ECXC mode there are 16 MBytes addressable.

Table B-1: CXC Address Range

Slot	Chip Select	[V]IUC VSBC-4	VM30	VSBC-32 IUC32	VM(6)42 VM(6)62	VSBC-860 IUC860 VMP60
BU2	CS0	0xF70000	0xC1F70000	0x0BF70000	0xCBF70000	0xB0000000
BU3	CS1	0xF70400	0xC1F70400	0x0BF70400	0xCBF70400	0xB0000400
BU4	CS2	0xF70800	0xC1F70800	0x0BF70800	0xCBF70800	0xB0000800
BU5	CS3	0xF70C00	0xC1F70C00	0x0BF70C00	0xCBF70C00	0xB0000C00
BU6	CS4	0xF71000	0xC1F71000	0x0BF71000	0xCBF71000	0xB0001000
BU7	CS5	0xF71400	0xC1F71400	0x0BF71400	0xCBF71400	0xB0001400
BU8	CS6	0xF71800	0xC1F71800	0x0BF71800	0xCBF71800	0xB0001800
BU0**	CS7	0xF71C00	0xC1F71C00	0x0BF71C00	0xCBF71C00	0xB0001C00

* *BU1 is the system slot*

** *On 5S, 8S, and 8ES for CXM-STAT1 only.*

© 2004 PEP Modular Computers GmbH

Table B-2: Enhanced CXC Address Range

Slot	Chip Select	VSBC-860 IUC860 VMP60	VSBC-32 IUC32	VM642 VM662
BU2	CS0	0xB0000000	0x10000000	0x10000000
BU3	CS1	0xB1000000	0x11000000	0x11000000
BU4	CS2	0xB2000000	0x12000000	0x12000000
BU5	CS3	0xB3000000	0x13000000	0x13000000
BU6	CS4	0xB4000000	0x14000000	0x14000000
BU7	CS5	0xB5000000	0x15000000	0x15000000
BU8	CS6	0xB6000000	0x16000000	0x16000000
BU0*	CS7	0xB7000000	0x17000000	0x17000000

* *BU1 is the system slot*
** *On 5S, 8S, and 8ES for CXM-STAT1 only.*

© 2004 PEP Modular Computers GmbH

B.2 CXC Generic Pinouts

Table B-3: CXC Connector Pinouts

Pin	Signals		
	Row A	Row B	Row C
1	_IRQ_1[1)]	SER1_RCLK	User-defined
2	_IRQ_2[1)]	SER1_TCLK	_DMA_ACK[1)]
3	_IRQ_3[1)]	GND	_DMA_REQ[1)]
4	_IRQ_4[1)]	SER1_TXD	User-defined
5	User-defined	SER1_RTS	SER3_TCLK
6	User-defined	GND	SER3_RCLK
7	VCC	SER3_RTS	VCC
8	User-defined	SER3_CD	SER3_TXD
9	User-defined	GND	SER3_RXD
10	User-defined	SER1_RXD	User-defined
11	SER2_DTR	User-defined	SER2_CD
12	SER3_DTR	GND	SER2_RTS
13	SER1_DTR	SER1_CTS	SER2_CTS
14	VCC	SER1_CD	VCC
15	_CS-CXC[1)]	GND	SER2_TCLK
16	_AS[1)]	SER3_CTS	SER2_RCLK
17	R/_W[1)]	_SYSR[1)]	SER2_TXD
18	_UDS[1)]	GND	SER2_RXD
19	_LDS[1)]	_EDTACK[1)]	VCC
20	VCC	CXC-CLK	_CS2[1)]
21	A1	GND	_CS3[1)]
22	A2	_CS0[1)]	_CS4[1)]
23	A3	_CS1[1)]	_CS5[1)]
24	A4	GND	_CS6[1)]
25	A5	A6	_CS7[1)]
26	VCC	A7	VCC
27	D0	GND	D10
28	D1	D6	D11
29	D2	D7	D12
30	D3	GND	D13
31	D4	D8	D14
32	D5	D9	D15

1) *Normally active low (by R/_W, only _W).*

© 2004 PEP Modular Computers GmbH

B.3 CXC Pinout Cross Reference to MPC860T

Table B-4: Cross Reference of CXC Signals to MPC860T

PIN	CXC SIGNALS	MPC860T
A1	IRQ_1	PC[12]
A2	IRQ_2	PC[13]
A3	IRQ_3	PC[14]
A4	IRQ_4	PC[15]
A5	User-defined	CXC A5 jumper J13 (PB[14] / +5VSTDBY)
A6	User-defined	PB30, SPICLK
A8	User-defined	PB29, SPITXD
A9	User-defined	PB28, SPIRXD
A10	User-defined	PB22
A11	SER2_DTR	PB[27]
A12	SER3_DTR	PB[31]
A13	SER1_DTR	PB[26]
B1	SER1_RCLK	SER1RCLK jumper J8 (open / PA[6])
B2	SER1_TCLK	PA[5]
B4	SER1_TXD	PA[12]
B5	SER1_RTS	PB[18]
B7	SER3_RTS	PB[16]
B8	SER3_CD	PC[4]
B10	SER1_RXD	PA[13]
B11	User-defined	PB[21]/SMC2TXD
B13	SER1_CTS	PC[9]
B14	SER1_CD	PC[8]
B16	SER3_CTS	PC[5]
C1	User-defined	PB[25]/SMC1TXD
C2	DMA_ACK	PB[23]
C3	DMA_REQ	PC[15]
C4	User-defined	PB[20]/SMC2RXD
C5	SER3_TCLK	PA[1]
C6	SER3_RCLK	PA[0]
C8	SER3_TXD	PA[8]
C9	SER3_RXD	PA[9]
C10	User-defined	PB[24]/SMC1RXD
C11	SER2_CD	PC[6]
C12	SER2_RTS	PB[17]
C13	SER2_CTS	PC[7]
C15	SER2_TCLK	PA[3]
C16	SER2_RCLK	SER2RCLK jumper J12 (open / PA[2])
C17	SER2_TXD	PA[10]
C18	SER2_RXD	PA[11]

© 2004 PEP Modular Computers GmbH