USER'S MANUAL

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

INTELLIGENT PMC/PCMCIA CARRIER

MODEL VX411C

Document Part No: 11028584

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

COPYRIGHT

C&H Technologies, Inc. (C&H) provides this manual "as is" without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. C&H may make improvements and/or changes in the product(s) and/or program(s) described in this manual at any time and without notice.

This publication could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of this publication.

Copyright © 2008, 2010 by C&H Technologies, Inc.

The information and/or drawings set forth in this document and all rights in and to inventions disclosed herein which might be granted thereon disclosing or employing the materials, methods, techniques or apparatus described herein, are the exclusive property of C&H Technologies, Inc.

A Reader's Comment Form is provided at the back of this publication. If this form has been removed address comments to:

C&H Technologies, Inc.
Technical Publications
445 Round Rock West Drive
Austin, Texas 78681-5012

Or visit our web site for support information at:

http://www.chtech.com.

C&H may use or distribute any of the information you supply in any way that it believes appropriate without incurring any obligations.

AMENDMENT NOTICE

C&H Technologies, Inc. makes every attempt to provide up-to-date manuals with the associated equipment. Occasionally, changes are made to the equipment wherein it is necessary to provide amendments to the manual. If any amendments are provided for this manual they are printed on colored paper and will be provided with the module and manual. Manual updates may also be found on out web site at www.chtech.com.

NOTE

The contents of any amendment may affect operation, maintenance, or calibration of the equipment.

INTRODUCTION

This manual describes the operation and use of the C&H Model VX411C Intelligent PMC/PCMCIA Carrier Module (Part Number 11028580). This VXI module is one of a number of test and data acquisition/control modules in the VME and VXI format provided by C&H.

Contained within this manual are the physical and electrical specifications, installation and startup procedures, functional description, and configuration guidelines to adequately use the product.

The part numbers covered by this manual are:

Part Number	Description
11028580-0002	VX411C Single Wide PMC/PCMCIA

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

TABLE OF CONTENTS

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

LIST OF FIGURES

LIST OF TABLES

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

1.0 GENERAL DESCRIPTION

The VX411C is an intelligent VXI carrier module that provides an electrical and mechanical interface for up to three PMC modules (two with front panel access) and one PC-Card (Cardbus). It features an on-board 32-bit PowerPC® processor that can perform command translation, data analysis, and many other data processing or process control functions. For a complete list of PMC modules compatible with the VX411C, visit the mezzanine section of www.mezzanines.org or www.vita.com. For detailed information on the PC-CARD standard, visit www.pcmcia.org.

1.1 PURPOSE OF EQUIPMENT

The VX411C was designed for Automated Test Equipment (ATE) applications requiring on-board instrument intelligence or data processing. Some of the more common applications include: legacy instrument emulation, data intensive signal acquisition, high speed signal analysis, and control processing.

1.2 SPECIFICATIONS OF EQUIPMENT

1.2.1 Key Features

- Supports three PMC modules (two with front panel access)
- Supports one PC-Card (Cardbus module)
- 300MHz MPC8245 Integrated Processor
- 128 megabytes SDRAM
- 16 kilobytes dual-ported SRAM accessible by both the processor and VXI
- 8 megabytes flash memory
- VXI A24/A32 access to shared memory
- VXI block transfers to/from shared memory
- DMA transfers between PowerPC, PMC devices, PC-Card, and shared memory
- Direct access to PMC and PC-Card modules from VXI
- Supports common off-the-shelf embedded operating systems

1.2.2 Specifications

Processor:

- MPC8245 300MHZ (MPC603e core)
- 16KB/16KB L1 Integrated Cache

Local PCI Bus:

- 33MHZ 32-bit

Main Memory:

- 128MB SDRAM
- 8MB Flash, VXI programmable
- 64KB Boot ROM, socketed

Shared Memory:

- 16 KB Dual-ported SRAM
- DMA/Burst support
- Internal arbitration
- Fully accessible by both VXI and PowerPC

PMC Interface:

- Support for three PMC modules
- 33MHz 32-bit
- 3.3V/5V signaling level is jumper selectable
- PMC I/O connected to 64-pin headers

Cardbus Interface:

- Automatic detection of 5V/3.3V 16-bit PC Cards and 3.3V 32-bit Cardbus cards
- Hot insertion and removal
- UltraMedia devices, such as Smart Media cards, MultiMedia Cards, Memory Stick devices, and Smart Card devices, defined by the PCMCIA Proposal 0262 are supported with appropriate adapters

Onboard External Relay Control:

- Darlington relay driver, 7-channels, 50V, 350ma
- Internal +5V supply or external power can be used

Onboard External Input:

- Four TTL inputs
- Allows direct external control of application software

Interrupts:

- PMC to PowerPC interrupt support
- Cardbus to PowerPC interrupt support
- PowerPC to VXI interrupt level 1-7 (programmable)
- VXI Host to PowerPC interrupt support

Temperature:

Operating:	0°C to 50°C
Storage:	-40°C to 70°C

Software:

C&H Intelligent Carrier Operating Systems (ICOS or Linux):

- Boot-up, initialization, and PCI bus enumeration
- VXI word serial protocol support
- Firmware download to Flash memory via VXI
- Math Library (ICOS only)

Direct Access:

- Direct VXI access of PMC and Cardbus modules
- Up to 8K of local PCI address space can be directly mapped to VXI A24 or A32 space

Debugging Interface:

- Common On-Chip Processor (COP)/JTAG
- Standard COP header
- Third-party development tools supported

RTOS Support:

- Architecture supports common real-time operating systems, such as VxWorks, OS-9, Linux, and others.

1.2.3 Electrical

The VX411C requires the +5V, ±12V, and ±24V power from the VXI back plane. The +5V supply drives a DC to DC converter supplying +3.3V to carrier components and the PMC positions. The VXI backplane can provide a total of 7.2 amps of +5 volts, of which, the VX411C uses a maximum of 2.3 amps (11.5W) for internal purposes. The remaining 4.9 amps (24W) are available to the PMC and PC-Card through a combination of the +5V and +3.3V supplies. Of that 4.9 amps, an absolute maximum of 4.4 amps (14.5W) can be provided to the PMC via the +3.3V supply. If the PCI bus is configured to operate off +5V, the entire 4.9 amps are available for use.

The ±12 volt supply is not used internally by the carrier, but may be required by an installed PMC module. The carrier can provide up to 1.5 amps (18W) each of both +12 volts and -12 volts to the PMC positions.

The ±24V is neither used by the carrier nor the PMC positions. However, it is available at an external connector for special purpose use. The carrier can supply a maximum of 1 amp (24W) each to the +24V and the -24V pins on the connector.

An external connection to the +5V supply is also provided. A maximum of 2.5 amps (12.5W) can be sourced or sink-ed to/from the carrier. As an output, the power drawn from this connector reduces the total power available to the PMC and PC-Card positions. As an input, the power provided to this connector increases the available power to the PMC and PC-Card positions. Even with an external +5V source an absolute maximum of 6 amps of +5 volts and 4.4 amps of +3.3 volts can be provided to the PMC and PC-Card positions.

For electrical information on individual PMC and PC-Cards, please reference each module's documentation. The power requirements for each installed PMC or PC-Card must be added to the VX411C's requirements for the total module's requirements.

1.2.4 Mechanical

The mechanical dimensions of the VX411C are in conformance with the VXI bus specification for the height and width of Size-C modules. The nominal dimensions are 233.35 mm (9.187 in) high x 340.0 mm (13.386 in) deep. The module is designed for a standard mainframe with 30.48 mm (1.2 in) width between slots.

1.2.5 Environmental

The environmental specifications of the module are:

Operating Temperature:	0°C to +50°C
Storage Temperature:	-40°C to +70°C
Humidity:	<95% without condensation

1.2.6 Bus Compliance

The module complies with the VXIbus Specification Revision 1.4 for C-Size VXI modules and with VMEbus Specification ANSI/IEEE STD 1014-1987, IEC 821.

Manufacturer ID:	FC1$_{16}$ (can also be set by PowerPC)
Model Code:	FD8$_{16}$ (can also be set by PowerPC)
VXI Access Type:	Register Based or Message Based
VXI Addressing:	A16/A24/A32
VXI Data Transfer:	D16/D32
VXI Sysfail:	supported
VXI Interrupts:	ROAK, programmable levels
VXI Local Bus:	Available
TTL Triggers:	SYNC trigger protocol supported
Memory Requirements:	32 Kilobytes

The modules on-board PMC bus complies with the PMC Specification IEEE P1386.1 for 32-bit PMC modules.

Bus Data Width:	32-bit
Bus Speed:	33 MHz
Bus Voltage:	5V or 3.3V (jumper selectable)
Rear Connector I/O:	64-pin Header

The modules on-board PC-Card bus complies with the PC Card Standard 8.0, PCI Local Bus Specification Rev. 2.2, PCI Bus Power Management I/F Specification 1.1, and PC 98/99.

Bus Data Width:	32-bit
Bus Speed:	33 MHz
Bus Voltage:	Automatic detection

2.0 INSTALLATION

2.1 UNPACKING AND INSPECTION

In most cases the VX411C is individually sealed and packaged for shipment. Verify that there has been no damage to the shipping container. If damage exists then the container should be retained as it will provide evidence of carrier caused problems. Such problems should be reported to the carrier immediately as well as to C&H. If there is no damage to the shipping container, carefully remove the module from its box and anti static bag and inspect for any signs of physical damage. If damage exists, report immediately to C&H.

2.2 HANDLING PRECAUTIONS

The VX411C contains components that are sensitive to electrostatic discharge. When handling the module for any reason, do so at a static-controlled workstation, whenever possible. At a minimum, avoid work areas that are potential static sources, such as carpeted areas. Avoid unnecessary contact with the components on the module.

2.3 INSTALLATION OF PMC MODULES

PMC modules must be installed before the VX411C is installed into the VXI system. To install modules, remove the VX411C's top shield and front panel covers as needed. ***There is never a need to remove the VX411C's bottom shield.*** Install a module by firmly pressing the connector on the PMC together with the connector on the carrier as shown in Figure 1. Secure the PMC through the holes on the bottom shield using screws provided with the PMC module.

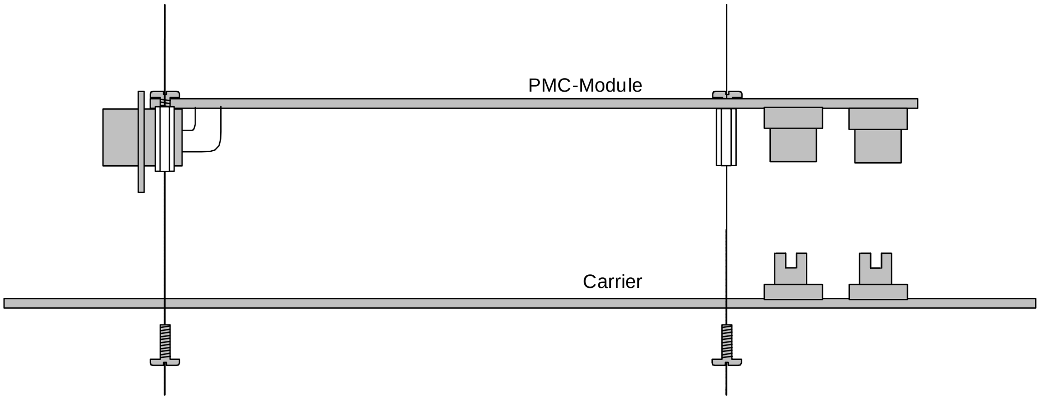


Figure 1 PMC Module Installation

There are three possible mounting locations on the carrier for PMC modules: One internal location without front panel access and two external locations (A & B) with front panel access. PMC Modules may be installed into any of the three locations. The mounting locations are illustrated in Figure 2.

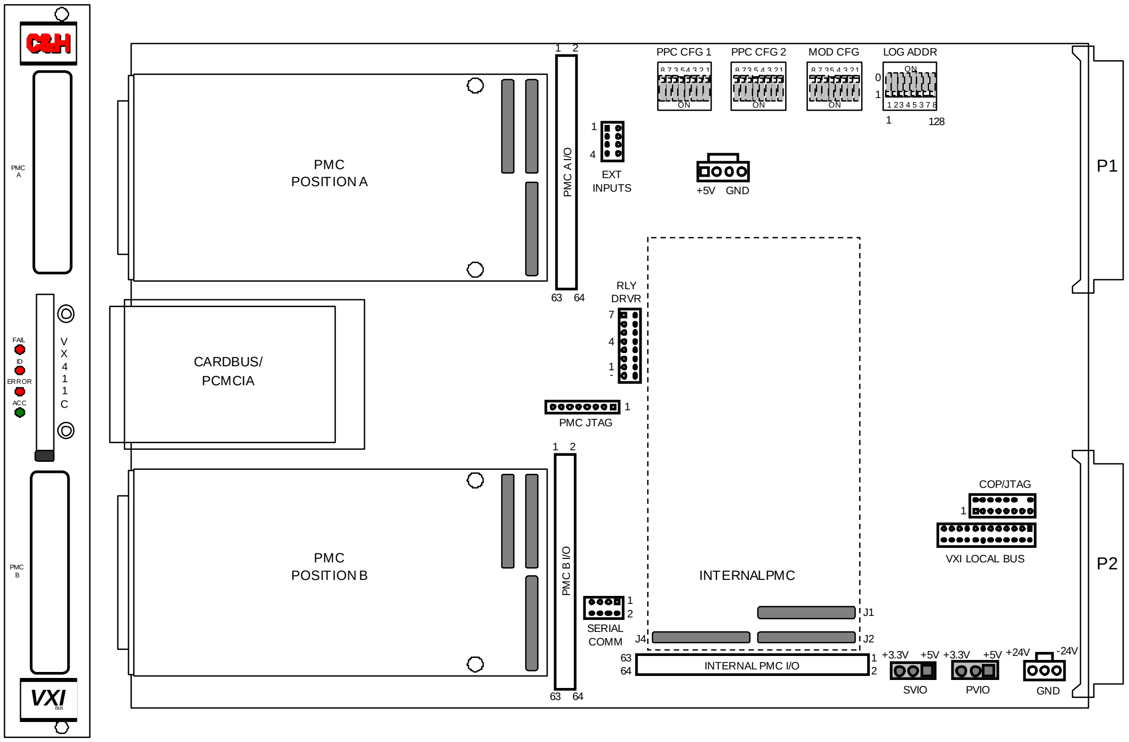


Figure 2 Front Panel and Top View (Top Shield Not Shown)

2.4 INSTALLATION OF PC-CARD MODULES

PC-Cards are hot-swappable and therefore, may be installed at any time before or after power is applied to the VX411C. Access to the PC-Card position is available via a slot in the front panel. Install a PC-Card by simply pushing the PC-Card into the slot on the front panel until it is firmly set. PC-Cards are keyed so that it is not possible to install incorrectly. When properly installed, the ejector button should be flush with the PC-Card. To eject a PC-Card, simply press on the ejector button until it is flush with the front panel. Then pull the card out of the slot.

2.5 INSTALLATION OF VX411C CARRIER

CAUTION: Read the entire User's Manual before proceeding with the installation and application of power.

If necessary, remove the top shield from the VX411C and configure the switches and jumpers. Set the module's logical address and addressing mode as described in section 3.4. Replace the shield and insert the carrier into the appropriate slot according to the desired priority and apply power. If no obvious problems exist, proceed to communicate with the module as outlined throughout the rest of this manual.

2.6 PREPARATION FOR RESHIPMENT

If the module is to be shipped separately it should be enclosed in a suitable water and vapor proof anti static bag. Heat seal or tape the bag to insure a moisture-proof closure. When sealing the bag, keep trapped air volume to a minimum.

The shipping container should be a rigid box of sufficient size and strength to protect the equipment from damage. If the module was received separately from a C&H system, then the original module shipping container and packing material may be re-used if it is still in good condition.

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

3.0 FUNCTIONAL OVERVIEW

3.1 GENERAL

The VX411C provides an intelligent interface between the VXI bus and up to three PMC modules (two with front panel access) and one PC-CARD (PCMCIA). It features an embedded processor system powered by a MPC8245 integrated processor. An on-board PCI bus provides an interface to three PMC modules, a PC-Card position, and 16 kilobytes of shared memory. VXI interface logic provides an interface between the VXI bus and the PowerPC via the shared memory and the PowerPC's local bus.

C&H Intelligent Carrier Operating System (ICOS) is provided to assist an embedded user application in performing necessary tasks or to allow the carrier to function normally when no user application is used. ICOS utilities include: boot-up and initialization routines, system configuration routines, VXI communications routines, an application programming interface (API), and various hardware interface routines to provide a basic interface to the carrier and installed modules and to assist application development.

For more advanced application development, a Linux distribution and Linux embedded development kit is also available from C&H for the VX411C. The Linux distribution includes all the items needed to create an embedded Linux system including a bootloader, patched kernel, root file system and many standard utilities. In addition, the processor architecture supports various other 3rd party commercially available real time operating systems.

3.2 HARDWARE OVERVIEW

The VX411C is powered by a highly integrated MPC8245 microprocessor with a PowerPC 603e core, a built-in Peripheral Component Interconnect (PCI) interface, and an advanced memory controller. The processor along with flash memory, ROM memory, and SDRAM form a complete embedded processing system with all the peripherals necessary for flexible application development.

Dual-ported shared memory and VXI interface logic allow for seamless communication between the VXI host and the PowerPC. Interrupts and handshaking logic is also provided to assist communications between the host and PowerPC.

Three PMC positions and a single PC-Card position residing on the on-board PCI bus allow a variety of instruments and peripherals to be added to the system. Both the VXI host and the PowerPC can communicate with and control the modules.

Finally, relay driver logic and TTL input logic allows special control hardware to be easily added to the overall integrated system. Figure 3 illustrates the system hardware architecture.

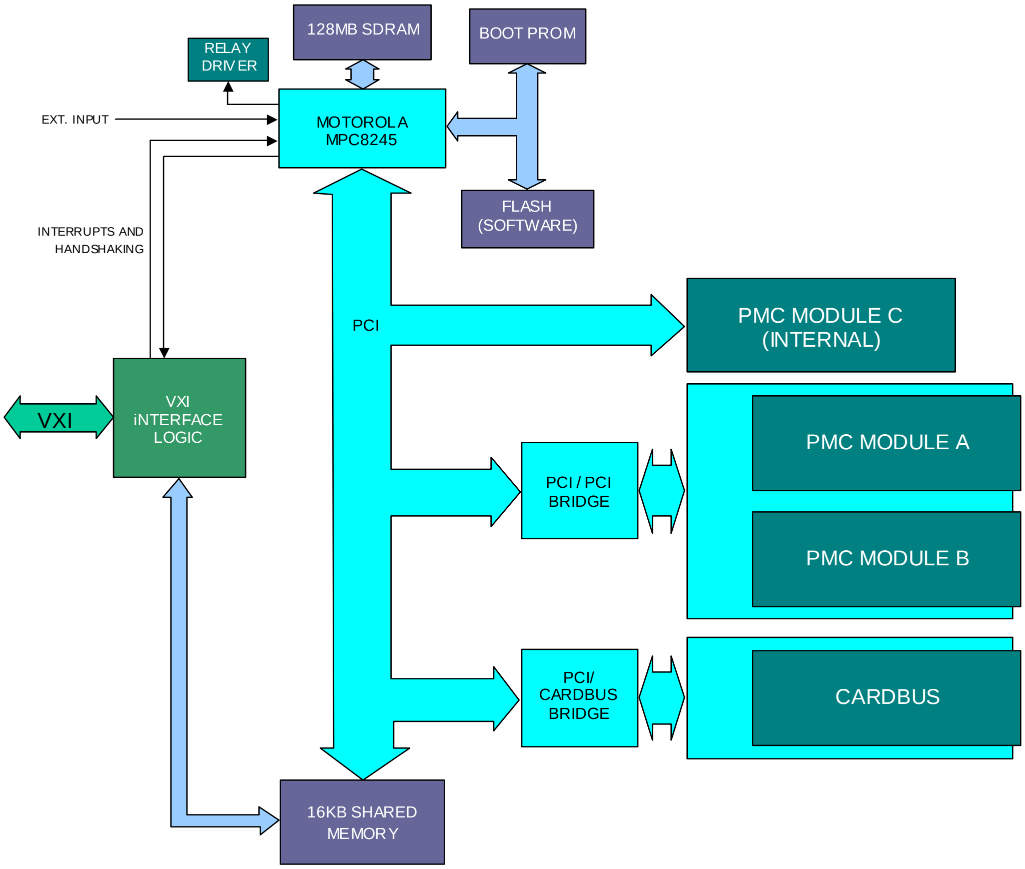


Figure 3 System Hardware Architecture

3.2.1 PMC Modules

The PMC modules provide the measurement, control, communication or other functionality for the given application. The carrier provides two PMC positions with front panel access and a single internal PMC position without front panel access. All three positions also make available rear connector I/O via a 64-pin header. The two external modules reside on a secondary PCI bus and interface to the primary on-board PCI bus via a PCI to PCI Bridge. The internal PMC position, interfaces directly to the primary PCI bus. Both the PowerPC application and the VXI host can communicate with and control the modules. A variety of PMC modules are commercially available from numerous manufacturers. For a list of modules compatible with the VX411C, visit the mezzanine section of www.mezzanines.org.

3.2.2 PC-Card Modules

The PC-Card position can be used to provide further functionality to the system. The PC-Card interfaces to the primary PCI bus via a PCI to Cardbus bridge and is accessible by both the PowerPC application and the VXI host. A Cardbus power switch provides hot-swap capability and distributes the appropriate power (3.3V or 5V) based on auto-detection of the PC-Card. The PC-Card interface is accessible via the front panel.

3.2.3 Shared Memory

The 16 kilobyte shared memory device acts as a buffer between the VXI bus and the on-board PCI bus. The device provides 16 kilobytes of dual-port SRAM and various other communications utilities such as general purpose FIFO’s. It connects to the VXI bus through a local bus interface controlled by the VXI interface logic and to the PowerPC through the PCI bus. The device performs on-chip memory arbitration allowing the 16 kilobytes of memory to be accessed at the same time from both the VXI bus and the PowerPC. It also contains an embedded PCI bus controller allowing the VXI bus to directly access the on-board PCI bus and thus directly access the M-modules and the PMC module.

3.2.4 PowerPC and Peripherals

The PowerPC architecture was designed as a standard embedded processor system. It consists of a MPC8245 PowerPC, a boot ROM device, 128 megabytes of SDRAM, and a flash memory device. This architecture allows the developer to select from standard off-the-shelf development tools and real-time operating systems for application development.

The PowerPC acts as the PCI bus master and can access the PMC modules, the PC-Card module, and the shared memory device. It also can access the VXI interface logic to perform handshaking between itself and the VXI bus.

3.2.5 VXI Interface Logic

The VXI interface logic acts as a transparent interface between the VXI bus and the shared memory device. It translates VXI bus accesses into shared memory local bus accesses by managing all local bus address and control lines. It maps all of the shared memory device’s address space to VXI A24/A32 space.

The VXI interface also handles handshaking between the PowerPC and the VXI bus. It includes a set of registers that are mapped to VXI A16 space and are accessible by the PowerPC to handle host to device communications and handshaking.

Finally the interface logic provides interrupt capabilities. VXI interrupts can be generated by the PowerPC application or shared memory device on any of the 8 VXI interrupt levels.

3.2.6 External Drivers

The PowerPC can control a Darlington sink driver device residing on its local memory bus. The device's outputs are available at a 16 pin header for external use. The device is intended to drive external relays, display LED's, or other high current devices.

3.2.7 External Input

Four external TTL input lines can be read by the PowerPC. The signals are available at an external connector on the carrier. These lines allow direct external control of the application software.

3.2.8 JTAG/COP Interface

The JTAG interface to the PowerPC provides a debug and development interface supported by many standard off-the-shelf developments tools. The interface is used by development tools to communicate with the processor. It provides the developer with the ability to view system registers, view memory, set breakpoints, and use other standard debugging practices.

3.3 SOFTWARE OVERVIEW

The embedded software on the carrier as well as the host software are very application dependant and thus, must be developed specifically to suit the needs of the particular application. However, firmware is provided as part of the Intelligent Carrier Operating System (ICOS) to assist application development and to provide basic functionality when no user application exists. In addition a Linux distribution is available from C&H.

The Intelligent Carrier Operating System (ICOS) is a single-threaded OS kernel specifically designed for use on the VX411C and C&H Technologies' other intelligent carriers. When no user application exists, ICOS provides basic functionality allowing the user to communicate with the carrier and the PMC and PC-Card modules. A limited set of VXI message based commands are available as well as the ability to access all defined registers and the shared memory. In this capacity, the carrier can operate as a fully functional instrument without the existence of a user application.

If a user application is to be provided, ICOS assists the developer in performing several tasks that require advanced knowledge of the carrier architecture and the devices that make up that architecture. For example a system routine is provided to program the flash memory so that the developer does not need to refer to the flash device's data sheet to learn the programming protocol. Also, ICOS automatically handles the communications required for VXI message passing so that the application can concentrate on performing high level tasks and not on the details of the VXI word serial protocol. ICOS is completely independent and fully interrupt and exception driven so that it only takes up a very small amount of processor resources and so the user application can be independently compiled and linked without knowledge of ICOS memory organization.

The Linux distribution available from C&H includes the DAS U-boot bootloader, a customized Linux kernel and a custom root filesystem that contains many familiar Linux utilities. Application development is performed using common open source development and debug tools and the C&H Linux Embedded Development includes the necessary hardware and software to ease application development. Contact C&H Technologies for details on the available Linux distribution and the Linux Embedded Development Kit.

The VX411C architecture also supports various other 3rd party Real-Time Operating Systems (RTOS's). Full support of 3rd party operating systems will depend on the availability of various VX411C specific software components for the RTOS. Contact the RTOS vendor and C&H Technologies for information on support for a particular RTOS.

3.4 HARDWARE CONFIGURATION

There are several switch and jumper selectable settings that configure the VX411C for operation. Configuration options include: the VXI logical address, PowerPC options, programming modes, and operational voltages. Figure 4 shows the layout of all the switches and jumpers on the VX411C.

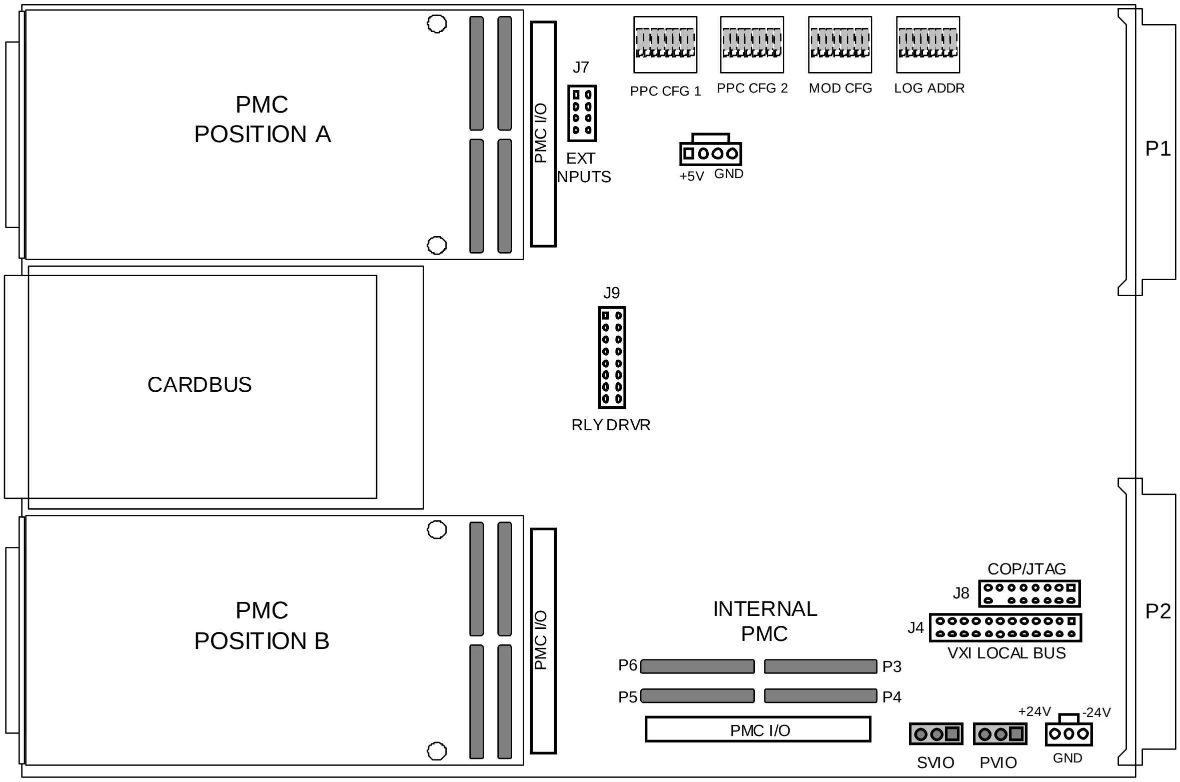


Figure 4 Hardware Layout

3.4.1 Logical Address Switch

The logical address switch determines the logical address for the VX411C. The switch forms a binary weighted decimal value that sets the logical address of the module. The OFF position for each switch represents a binary one in that bit position. For example, the switch settings shown in Figure 5 would result in a logical address of 36.

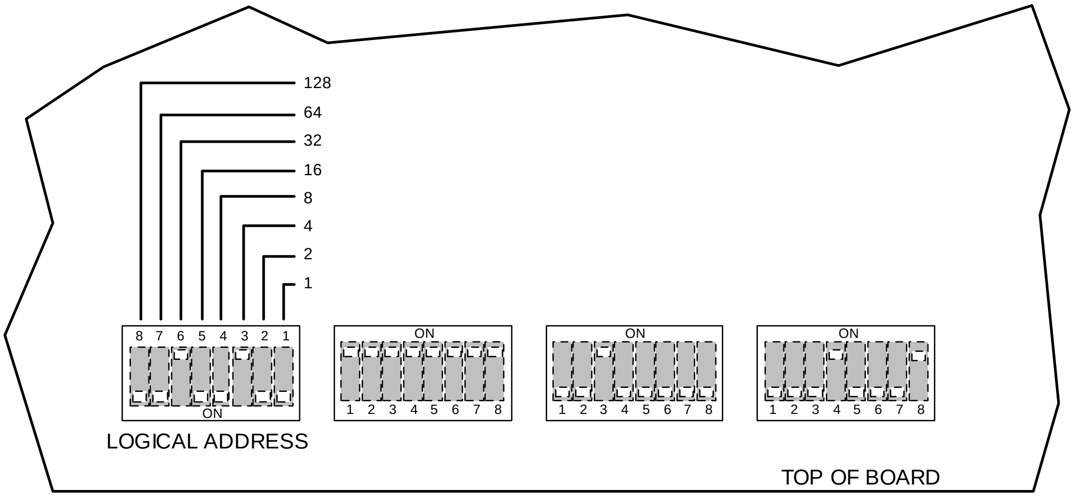


Figure 5 Logical Address Configuration Switch

3.4.2 Module Configuration Switch

The module configuration switch is used to set some of the miscellaneous options on the VX411C. Figure 6 shows the options that are configurable with this switch.

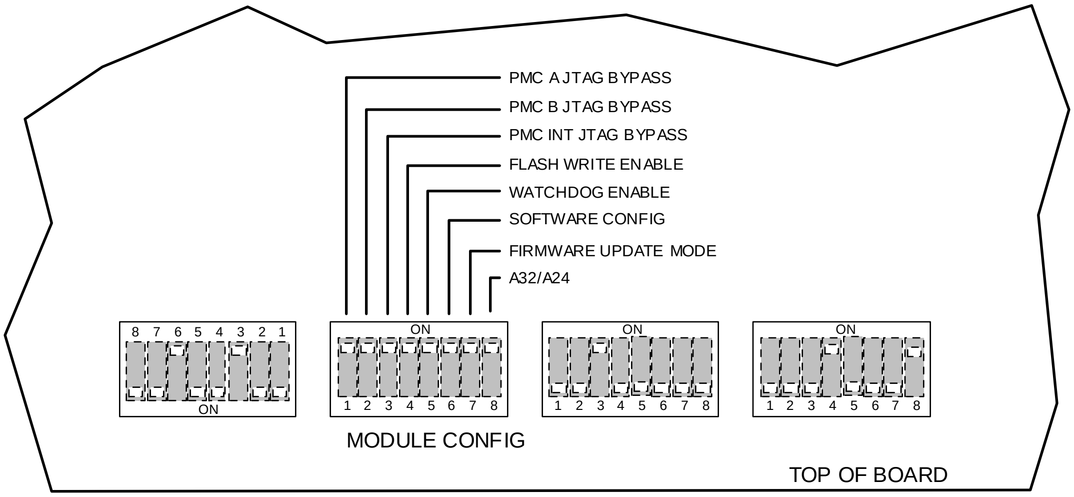


Figure 6 Module Configuration Switch

PMC JTAG Bypass Switches: Setting these switches to ON will bypass the PMC devices' JTAG interface by automatically daisy chaining the JTAG input to the JTAG output at the PMC connector. The JTAG bus is daisy chained between the PMC connector and other devices on the carrier. It is necessary to bypass a PMC position when using the JTAG interface with no PMC module installed in that position. ***The PowerPC's JTAG/COP interface is an independent JTAG interface and is not affected by these switches.***

Flash Write Enable Switch: This switch will enable or disable the ability for software to program flash memory. Setting this switch to ON will enable the flash programming capability. When the switch is OFF the carrier will not allow the PowerPC to program the flash device.

Watchdog Enable Switch: The Watchdog Enable Switch enables/disables the watchdog timer at the hardware level. The watchdog timer may also be disabled by software by writing to the Watchdog Timer Control Register (Offset 0x26). **Disabling the watchdog timer at the hardware level is required to utilize a JTAG debugger on the processor.**

Software Configuration Switch: The software configuration switch is available for use by the user application and system firmware. This switch has no effect on the hardware operation of the VX411C. The value of the switch is copied to the operations registers so that the software can read the value and define the switch's function. Setting the switch to ON results in the corresponding bit of the operations register being set to a binary '0'. OFF corresponds to a binary '1'. ***The operating system or application software may define a function this switch and therefore setting it may affect the operation of the carrier. Refer to the operating system or application software's documentation for details.***

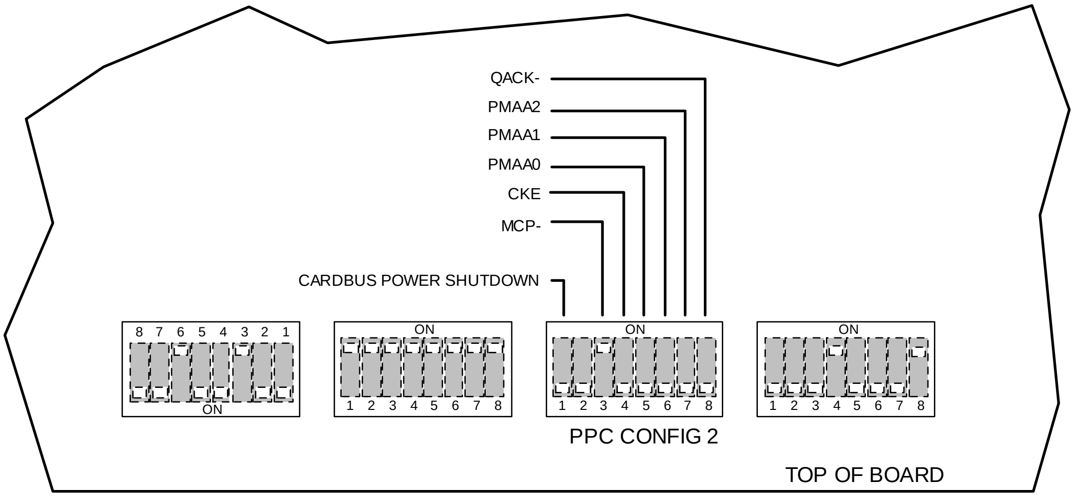
Firmware Update Mode Switch: This switch determines whether the VX411C boots normally or into a mode where the system firmware can be updated. If this switch is set to OFF at power-up, the carrier will go into the firmware update mode and wait for firmware to be downloaded via the VXI bus. If this switch is ON at power-up, the carrier will initialize normally and launch the system firmware.

A32 Switch: This switch selects whether the VX411C performs VXI A24 or A32 address decoding. This address space is used to access the shared memory device. If this switch is set to ON, the carrier requests memory in the systems A32 address space, otherwise A24 address space is requested.

3.4.3 PowerPC Configuration Switches

The PowerPC configuration switches determine the configuration of the MPC8245 after reset. With the exception of the Cardbus Power Shutdown switch, each signal connected to these switches is a reset configuration signal for the PowerPC. The values of these signals at reset, determine the configuration of the processor. Figure 7 shows all available PowerPC configuration switches. Table I briefly describes each reset configuration signal all possible settings. For details on reset configuration refer to the *MPC8245 User's Manual*.

WARNING: The PowerPC configuration switches are preset during manufacturing to the optimal settings for the VX411C. Modifying these settings is rarely necessary and in some cases may cause the VX411C to not function correctly.



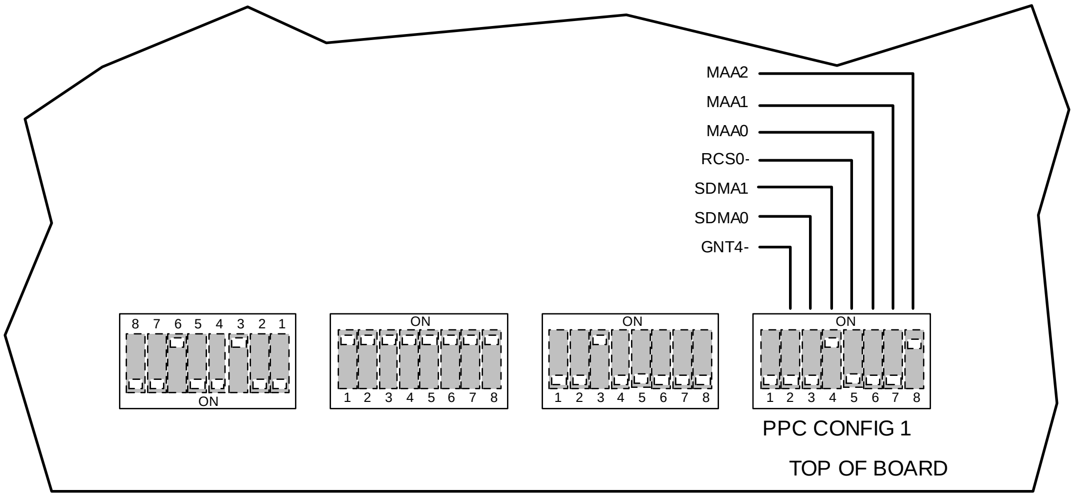


Figure 7 PowerPC Configuration Switches

Cardbus Power Shutdown Switch: This switch, when set to the ON position, will place the Cardbus power logic into shutdown mode. In shutdown mode, the Cardbus power signals (Vcc & Vpp) will be forced to a high-impedance state (HI-Z) and the Cardbus interface will become inoperable.

Table I PowerPC Configuration Signals

Signal	Description	Settings
MCP-, CKE	Sets the PCI output hold delay value (in nanoseconds) relative to PCI_SYNC_IN. Refer to the MPC8245 documentation for details on each setting.	(MCP_ = 0, CKE = 0) **(MCP_ = 0, CKE = 1) Recommended for 33 MHz PCI[1,2]** (MCP_ = 1, CKE = 0) (MCP_ = 1, CKE = 1) Recommended for 66 MHz PCI
PMAA0, PMAA1	Memory signal driver capabilities.	(PMAA0 = 0, PMAA1 = 0) = reserved (PMAA0 = 0, PMAA1 = 1) = 40 Ω drive capability (PMAA0 = 1, PMAA1 = 0) = 20 Ω drive capability **(PMAA0 = 1, PMAA1 = 1) = 6 Ω drive capability[1,3]**
PMAA2	PCI and EPIC controller driver capabilities	**0 = 40 Ω drive capability[1,2,3]** 1 = 20 Ω drive capability (except for IRQ2/S_RST and IRQ3/S_FRAME- signals which have 6 Ω drive capability)
QACK-	Clock Flip Disable	0 = Clock flip enabled **1 = No clock flip[1,2]**
GNT4-	Debug Address Disable	0 = Debug address enabled **1 = Debug address disabled[1,2]**
SDMA0	DUART Signals Disabled	0 = DUART signals enabled **1 = PCI_CLK[0:3] signal used instead of DUART[1,2]**
SDMA1	Extended Addressing Mode	**0 = Extended addressing mode enabled[1,2]** 1 = Extended addressing mode disabled
RCS0-	Boot Memory Location	0 = Boot ROM is located on the PCI bus **1 = Boot ROM is located on the local bus[1,2]**
MAA0	Address Map Setting. The MPC8245 only supports address map B.	0 = Invalid **1 = MPC8245 is configured for address map B[1,2]**
MAA1	PCI Host Mode	0 = MPC8245 is a PCI agent device **1 = MPC8245 is a PCI master device[1,2]**
MAA2	PCI Arbiter Disable	**0 = PCI arbiter enabled[1,2]** 1 = PCI arbiter disabled

Notes: 1. Bold indicates the recommended setting for the VX411C
2. 1=Switch OFF, 0=Switch ON (except for PMAA2 see note 3)
3. For the PMAA2 switch, 1=Switch ON, 0=Switch OFF

3.4.4 VIO Configuration Jumpers

The Primary VIO (PVIO) and Secondary VIO (SVIO) configuration jumpers select the voltage level supplied to the VIO pins on the PMC connector. The internal PMC module runs off the PVIO voltage and the two external PMC modules run off the SVIO voltage. The VIO power signal is used by universal PMC modules that can operate in both +5V and +3.3V systems. On these boards, the power for the I/O buffers is provided by the VIO pin instead of directly from the +3.3V or +5V power pins. Set the jumper according to the PMC modules installed on the carrier as shown in Figure 8.

The other devices on the PCI bus do not use the VIO pin therefore, if no PMC module is installed, either jumper position is acceptable. The jumper must be installed for the PCI bus to operate.

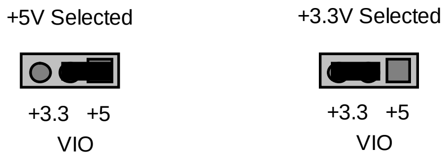


Figure 8 VIO Configuration Jumper

3.5 CONNECTORS

The VX411C incorporates several connectors to provide a physical connection to its various interfaces. Figure 4 shows the general location of each connector on the VX411C. Detailed pin-out information can be found in Appendix A. A short description of each connector is provided in the following sections.

3.5.1 External Power Connectors

Two connectors are provided to connect +5V and ±24V externally. The +5V connection is provided by a Molex 70543 male 4-pin connector. The ±24V connections are provided by a Molex 70543 male 3-pin connector. Refer to Appendix A for details on the connector pin-outs.

3.5.2 External Drivers Connector

The external relay driver output signals are available on a 16-pin header (8x2 with 0.100 inch centers). Refer to Appendix A for details on the header pin-outs.

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

3.5.3 External Input Connector

The four external TTL input lines are available on a standard 8-pin female header (4x2 with 0.100 inch centers). Refer to Appendix A for details on the header pin-outs.

3.5.4 JTAG/COP Connector

Connection to the PowerPC's JTAG/COP debug interface is provided through a keyed 16-pin header (8x2 with 0.100 inch centers). This header is the standard size and employs the standard pin-out used by most JTAG based emulators. The pin-out details of the JTAG/COP header can be found in Appendix A.

3.5.5 PMC Connectors

There are three PMC interfaces provided. Two located such that the front panel of the PMC module is available at the front of the carrier. The other PMC position is internal and does not have front panel access. Each of the PMC positions has three connectors to provide the electrical interface to a PMC module. The connectors are configured in accordance with the PMC specification. Refer to Appendix A for pin-out details.

3.5.6 PMC I/O Connectors

Some PMC modules provide User I/O signals on the PMC's P4 connector. The VX411C provides connection to the signals through a standard 64-pin header (32x2 with 0.100 inch centers) located next to the PMC interface connectors. Refer to Appendix A for pin-out details. The headers can be used for inter-module I/O communication.

3.5.7 PC-Card Connector

The PC-Card connector provides the electrical interface to a PC-Card. The connector is configured in accordance with the latest PC Card standard and supports Type I, II, or III PC Card or Cardbus modules and UltraMedia devices, such as Smart Media cards, MultiMedia Cards, Memory Stick devices, and Smart Card devices.

3.5.8 VXI Connectors

The rear connectors, labeled P1 and P2, provide the electrical interface to the VXI system. They are configured in accordance with the VXI specification. Refer to Appendix A for pin-out details.

3.5.9 VXI Local Bus Connector

The VXI Local Bus signals are available at a standard 24-pin header (12x2 with 0.100 inch centers). The VX411C does not use the VXI Local Bus internally but makes it available for external use at this connector.

4.0 SYSTEM ARCHITECTURE

4.1 OVERVIEW

The system architecture illustrated in Figure 3 is viewed differently from an application running on the embedded PowerPC than from an application running on the VXI host. Most of the carrier's hardware can be accessed by both applications but, the methods for doing so differ. The system architecture is best described by viewing the host-side and the device-side separately. However, it is also important to understand how the resources shared between both applications are used for host to device communications.

4.2 DEVICE-SIDE ARCHITECTURE

The device-side architecture is anchored by a standard embedded processor system powered by the MPC8245 PowerPC. The architecture provides on-board RAM, boot ROM, and flash memory to support the software application. The PowerPC acts as the PCI bus master and has full access to all devices on the PCI bus. A set of operational registers, the external relay driver and the external inputs are available to the application via the processor's local bus. Figure 9 illustrates the device-side architecture.

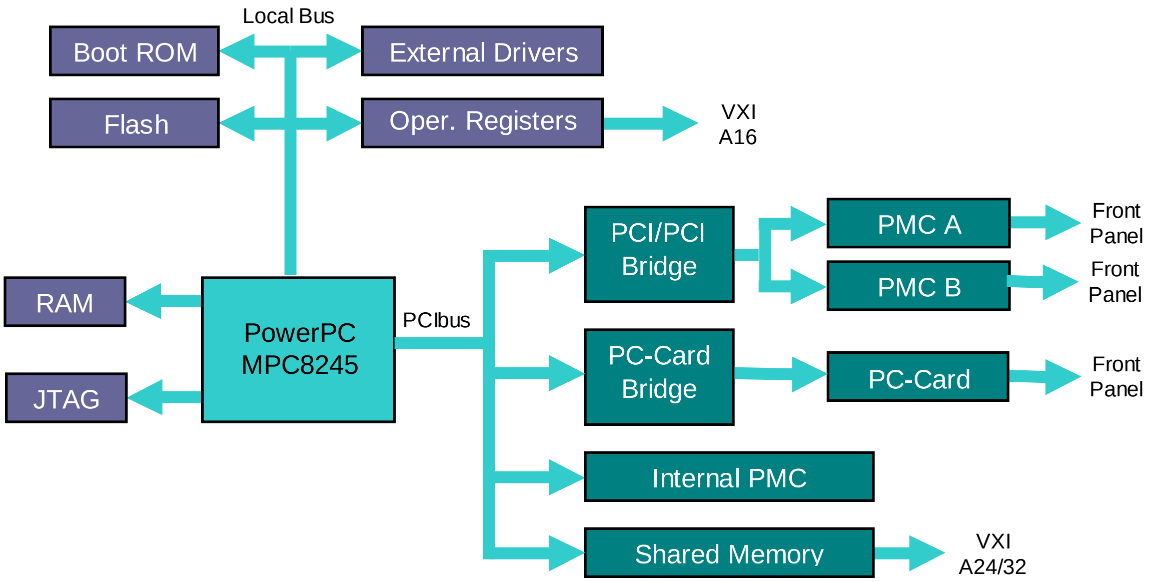


Figure 9 Device-Side Architecture

4.2.1 PowerPC Memory Map

Being a 32-bit processor, the MPC8245 can address up to 4 Gigabytes of physical memory. On the VX411C, the processor maps this 4 Gigabytes of memory into a configuration designated as Address Map B. The address map B configuration divides the memory space into sections that, when accessed, translate the operation to a local memory, PCI memory, PCI I/O, or ROM access. Figure 10 shows the general layout of address map B.

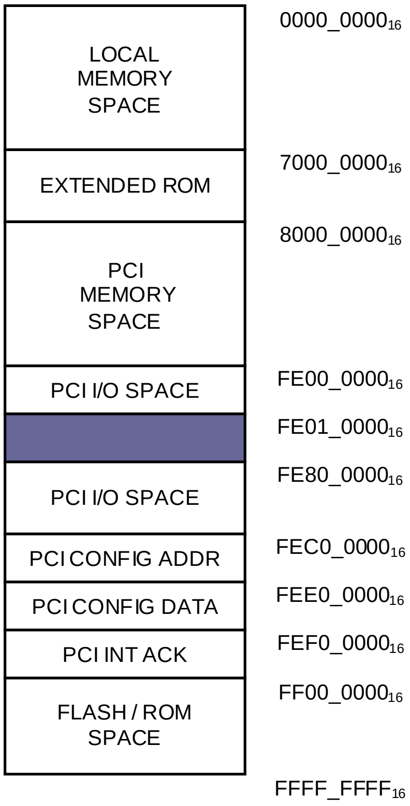
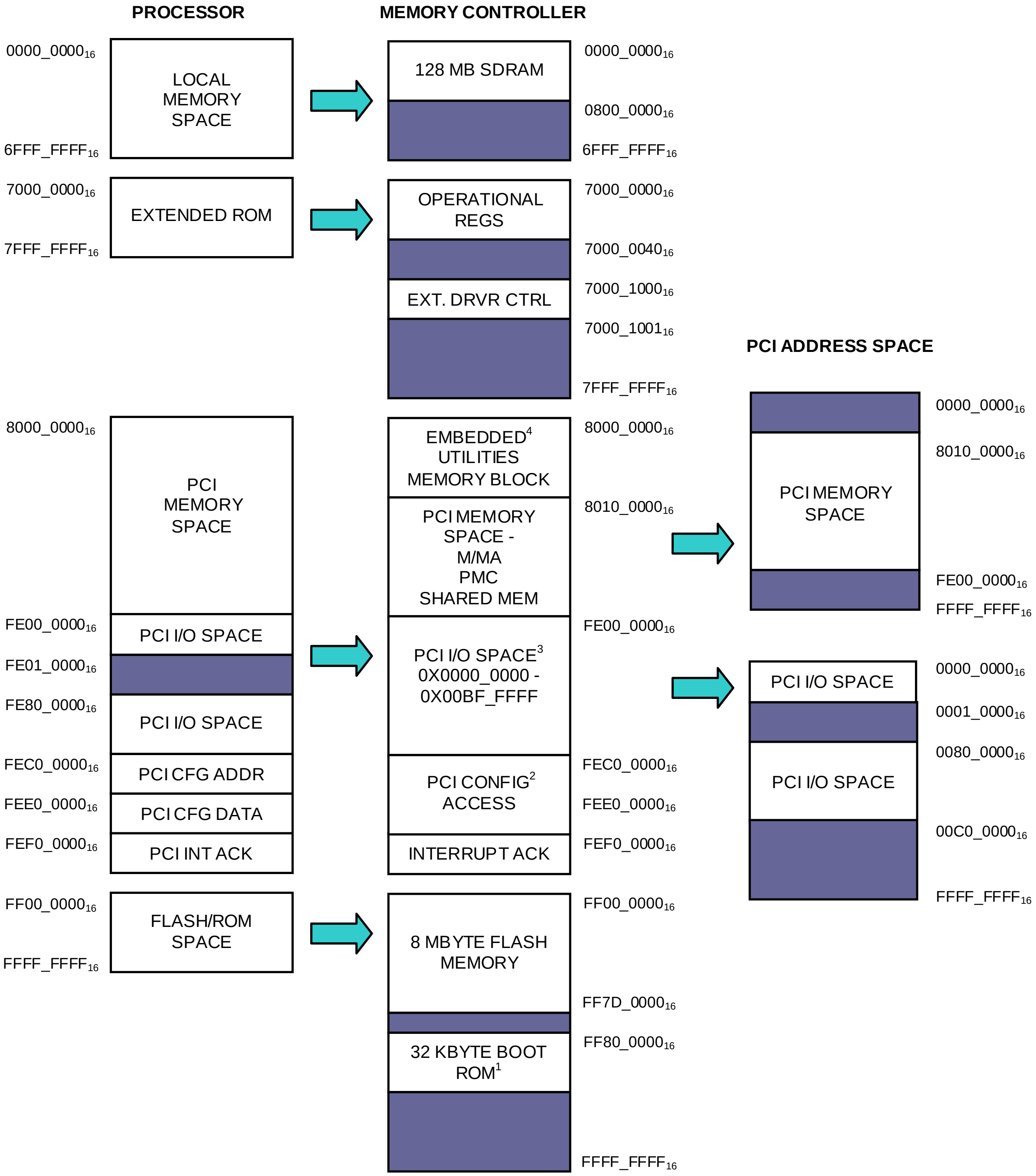


Figure 10 Address Map Overview

Details of the address map B implementation for the VX411C architecture are shown in Figure 11. Each section in the address map directly addresses a resource in the system architecture. Further details of each address block are provided throughout this document. Other address map B options and settings are also available but generally not used on the VX411C. Refer to the MPC8245 User's Manual for details.



Notes:

1. The boot ROM device only decodes 15 address lines. Therefore, the boot ROM is repeated throughout the address space. For example, address FF80_0000$_{16}$ is the same location as FFF0_0000$_{16}$.
2. Addresses FEC0_0000$_{16}$ and FEE0_0000$_{16}$ are used to perform PCI configuration accesses as described in section 6.2.
3. PCI I/O accesses are forwarded to the PCI bus with the 8 most significant bits of the address cleared. (i.e. processor address FE80_0000$_{16}$ = PCI I/O address 0080_0000$_{16}$)
4. In reality, the embedded utilities memory block can be located at any location between 8000_0000$_{16}$ and FDF0_0000$_{16}$ by setting the EUMBAR PowerPC configuration register.

Figure 11 Detailed PowerPC Address Map

4.2.2 SDRAM

The SDRAM provides 128 Megabytes of temporary storage for the application and system software. The memory is organized in a 13 rows x 10 columns x 4 banks configuration. It has a 10ns access time and a 32 bit wide data bus. It is accessed through the PowerPC's addresses space starting at offset 0.

4.2.3 Boot ROM

The boot ROM provides 64 kilobytes of non-volatile, read-only memory. It is normally programmed during the manufacturing process to contain boot code and initialization routines. It can not be reprogrammed in circuit. The boot ROM is mapped to PowerPC address FF80_0000$_{16}$ and has an 8-bit data bus. Only 15 address bits are decoded so that the 64 kilobytes are repeated throughout the PowerPC's ROM/Flash space between addresses FF80_0000$_{16}$ and FFFF_FFFF$_{16}$. Consequently, the default exception vector table starting at address FFF0_0000h resides in the boot ROM device.

4.2.4 Flash Memory

The flash device provides 8 megabytes of non-volatile storage for code and data. Unlike the boot ROM, flash is programmable in circuit and may be used by the system firmware. The flash device is accessed starting at PowerPC address FF00_0000$_{16}$ and has an 8-bit wide data bus. Reads from flash are performed as standard PowerPC memory accesses. Programming and erasing the device, however, requires a sequence of commands to be sent to the device. In addition, a hardware configuration switch is available to enable/disable programming of the flash device. If disabled, firmware can neither write-to or erase the device.

4.2.5 PCIbus Architecture

The VX411C has two onboard PCI buses. The primary on-board PCI bus contains five devices including the PowerPC, which acts as the bus master. The other devices on the primary bus are the shared memory device, the internal PMC, a PCI to PCI Bridge, and the Cardbus Bridge. The secondary PCI bus contains the two external PMC interfaces. Both buses operate at 33 MHz and 5V or 3.3V (jumper selectable).

PCI memory, configuration, and I/O space is memory mapped directly into the PowerPC's address map as shown in Figure 11. Approximately 2 gigabytes of PCI memory space is mapped starting at address 8000_0000$_{16}$. Each device requiring memory will have a base address within this mapped area. If the processor's Embedded Utilities Memory Block (EUMB) is enabled, it occupies 1 megabyte of PCI memory space effectively reducing the amount of memory space available to the PCI devices. About 4 megabytes of PCI I/O space is mapped to PowerPC addresses FE00_0000$_{16}$. When performing a PCI I/O access, the processor clears the upper 8 bits of the address before forwarding the transaction to the PCI bus. So, for example, accessing processor address FE80_0000$_{16}$ will read or write PCI I/O address 0080_0000$_{16}$. The base address of each device is determined by the PCI enumeration routines during initialization. The base address of a particular device can be determined by reading its Base Address Register (BAR) register in PCI configuration space for that device.

To perform a single PCI configuration write or read, two processor accesses are required. First, the PCI configuration address register at PowerPC address FEC0_0000$_{16}$ must be set to point to the correct device and offset. Then the data can be read from or written to the PCI configuration data register at PowerPC address FEE0_0000$_{16}$. The PCI configuration address register value is determined by the bus number, IDSEL signal routing, device function number, and the register offset. Configuration writes and reads to the primary PCI bus are performed using Type 0 configuration accesses. Configuration writes and reads to the secondary and other subordinate PCI buses are performed using Type 1 configuration accesses. For details on performing PCI configuration accesses refer to the MPC8245 User's Manual.

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

4.2.5.1 IDSEL Signal Routing

Each device on the PCI bus has a unique ID select line used to specify the destination of a configuration access. The PCI specification does not stipulate the source of each ID select line; however, the upper 16-bits of the address bus are normally used. On the VX411C each device has its IDSEL line tied to a specific address line as shown in Table II for the primary PCI bus and Table III for the secondary PCI bus. The device number, normally provided to software routines, is also system dependant. Table II and Table III also show the device numbering used on the VX411C. This information must be incorporated into a configuration access by software when performing a write or read.

Table II Primary Bus IDSEL Signal Routing

Device	IDSEL	DevNum
• Shared Memory	AD16	16_{10}
• PowerPC	AD17	17_{10}
• Internal PMC	AD18	18_{10}
• PCI to PCI Bridge	AD19	19_{10}
• Card Bus Interface	AD20	20_{10}

Table III Secondary Bus IDSEL Signal Routing

Device	IDSEL	DevNum
• PMC A	AD30	15_{10}
• PMC B	AD31	14_{10}

Note: Secondary bus number are determined by PCI/PCI Bridge

4.2.6 PCI Interrupts

The PowerPC's Embedded Programmable Interrupt Controller (EPIC) acts as the PCI interrupt controller. The interrupt lines from the PCI devices including the PMC positions and the Cardbus interface are routed to the EPIC controller's five interrupt inputs as shown in Table IV. The PCI to PCI bridge device and the PCI to Cardbus bridge do not generate interrupts. This architecture requires that several devices may share a common interrupt line. It is up to software to determine the source of an interrupt on shared lines. Refer to the MPC8245 User's Manual for information on programming the EPIC controller to handle interrupts.

Table IV PCI Interrupt Signal Routing

PCI Int	**A**	**B**	**C**	**D**
Device	**EPIC IRQ**			
PMC Internal	0	1	2	3
PMC A	1	2	3	0
PMC B	2	3	0	1
Cardbus	3	-	-	-
Shared Memory	4	-	-	-

4.2.6.1 Shared Memory Device

The shared memory device's entire address space is mapped to PCI memory space including all registers, the I_2O messaging unit, and the general purpose shared memory. The offset into PowerPC memory space is determined at boot up by the system software.

Figure 12 shows the shared memory device's address map. All addresses are offsets from the device's base address. To determine the shared memory device's base address, read offset 10_{16} (BAR0 Register) of the shared memory device's PCI configuration space. This value is the base address of the shared memory device. The general purpose shared memory begins at an offset of 4000_{16} from this address. Full details of the shared memory device's address map, including register details, can be found in the *CY7C09449PV Data Sheet* from Cypress Semiconductor, Inc.

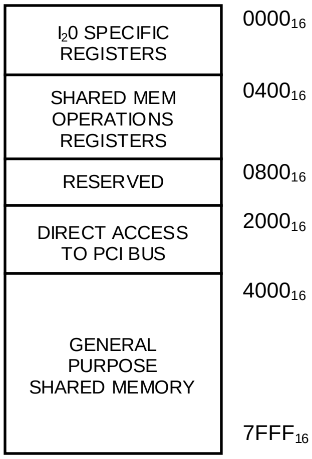


Figure 12 Shared Memory Organization

4.2.6.2 Internal PMC Device

The internal PMC position resides on the on-board primary PCI bus. Its address space is mapped directly to the PowerPC's address map. The bus mode signals are implemented to inform the PMC module of the PCI bus configuration. The PMC module may contain a PCI to PCI Bridge whose secondary bus is fully accessible by the PowerPC. Interrupts from the PMC device are supported as described in section 4.2.6.

4.2.6.3 PCI to PCI Bridge

The PCI to PCI Bridge interfaces the primary PCI bus controlled by the PowerPC to a secondary PCI bus containing the two PMC devices with front panel access. The PCI to PCI Bridge device is a Texas Instruments PCI2250. As shown in Table II the PCI to PCI Bridge resides at device 19 on the primary PCI bus.

4.2.6.4 External PMC Devices

The two external PMC positions reside on the on-board secondary PCI bus. Access to these devices is transparently routed through the PCI to PCI Bridge. The address space of the external PMC devices is mapped directly to the PowerPC's address map. The bus mode signals are implemented to inform the PMC module of the PCI bus configuration. The PMC module may contain a PCI to PCI Bridge whose secondary bus is fully accessible by the PowerPC. Interrupts from the PMC device are supported as described in section 4.2.6.

4.2.6.5 Cardbus Interface

A PCI Cardbus controller interfaces the primary PCI bus to a PCMCIA or Cardbus device. The Cardbus controller device is a Texas Instruments PCI1620 Dual Socket PC-Card and Ultramedia controller. Only one of the available PC-Card sockets is electrically connected on the VX411C. The Cardbus controller resides at device 20 on the primary PCI bus. The low-level Cardbus interface software is part of the installed operating system.

4.2.7 Operations Registers

The operations registers are a shared resource between the PowerPC and VXI host and are used to facilitate communication between the two sides and to allow software to perform other operations on the VX411C such as read the current state of the TTL inputs. The standard set of VXI defined registers are part of the operations registers. The PowerPC has access to these operations registers via the local bus. These registers are mapped to the PowerPC's extended ROM space starting at address 7000_0000_{16}. The data bus width between the PowerPC and the operations registers is 8-bits and reads and writes are performed as standard memory accesses. For register definitions in this space refer to section 4.4.1.

4.2.8 External Drivers

The architecture includes an 7-bit Darlington sink driver device residing on the PowerPC's local memory bus. The device is intended to drive external relays, display LED's, or other devices with high current requirements. The device's outputs are available at a 16 pin header for external use. Refer to section 3.5.2 and Appendix A for details on the header

Access to the device is provided at address 7000_1000_{16}. The data bus width to the device is 7-bits wide and each bit corresponds to one of the 7 channels. The device can only be written to. Figure 13 shows the external driver control register.

External Driver Control

7000_1000

Bit	7	6	5	4	3	2	1	0
Write	CH7	CH6	CH5	CH4	CH3	CH2	CH1	-
Read	(write-only)							

CHx ⇨ Channel value (1 = driven, 0 = not drivern)

Figure 13 External Driver Control Register

4.2.9 Watchdog Timer

A watchdog timer is available to reset the processor in the event that an errant software flow occurs. The watchdog can be disabled using the Module Configuration switch described in section 3.4.2 or by software by writing to the Watchdog Timer Control Register (Offset 0x26). ***Disabling the watchdog timer at the hardware level is required to utilize a JTAG debugger on the processor.***

4.2.10 External TTL Inputs

The PowerPC has access to four general purpose TTL inputs. The inputs can be driven externally at a standard 8-pin header. Refer to section 3.5.3 and APPENDIX A for details on the header. The values of the inputs are stored in the operations registers and can be read by the PowerPC. Refer to section 4.4.1 for details on the operations registers.

4.2.11 JTAG/COP Interface

The JTAG interface to the PowerPC provides support for several standard off-the-shelf developments tools. Most development environments for the PowerPC support JTAG based communications with the processor. It provides the developer with the ability to view system registers, view memory, set breakpoints, and use other standard debugging practices. ***To utilize a JTAG debugger on the processor, the watchdog timer must be disabled using the Module Configuration switch described in 3.4.2.***

Connection to the JTAG/COP interface is provided through a standard 16 pin header. Refer to section 3.5.4 and APPENDIX A for details on the header.

4.3 HOST-SIDE ARCHITECTURE

The host-side architecture is anchored by the VXI system including a VXI chassis and a host computer. Standard off-the-shelf VXI controllers from several different manufacturers are available to interface the carrier to the host computer, including high performance embedded controllers. The VXI host has access to the entire address space of the shared memory device as well as to a set of the operations registers. The shared memory device provides a utility to directly access devices on the PCI bus from the VXI host. Therefore, the VXI host application can control on-board PCI devices and thus the M-modules without the assistance of the PowerPC. The standard VXI registers required by the VXI specification are implemented as part of the operations registers. These include the registers required to implement the VXI word serial protocol. The VXI host has the ability to fully access all devices on the on-board PCI bus and to fully utilize all host-to-device communications utilities. Figure 14 illustrates the intelligent carrier architecture as viewed from the VXI host computer.

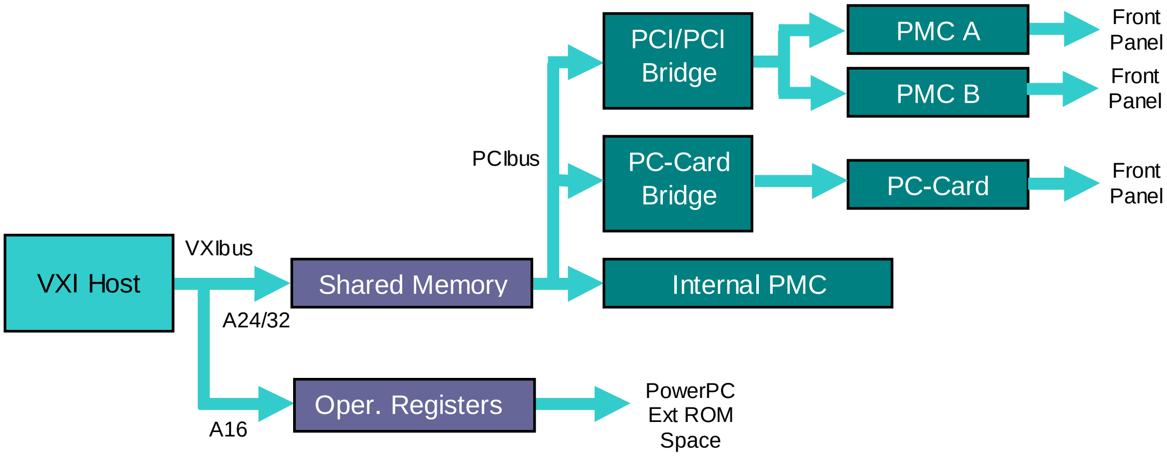


Figure 14 Host-Side Architecture

4.3.1 VXI Memory Map

Figure 15 shows the host-side memory organization for the intelligent carrier. The operations registers are accessed via VXI A16 space. These registers include the VXI required registers and the VXI message based communication registers as defined by the VXIbus specification.

A24/A32 memory space is a direct mapping of the shared memory device's memory map. This architecture gives the host full access to the shared memory and its registers including direct access to the PCI bus and other miscellaneous communications utilities.

A24 or A32 addressing is switch selectable as described in section 3.4.2. The VXI resource manager will write a base address to the offset register at address 06_{16} in A16 space. If the carrier is configured for A32 addressing, the carrier will use the value of the offset register as the upper 16 bits of its 32-bit base address. If A24 addressing is selected, the carrier will use the value in the offset register as the upper 16 bits of its 24-bit base address. This behavior is illustrated at the bottom of Figure 15.

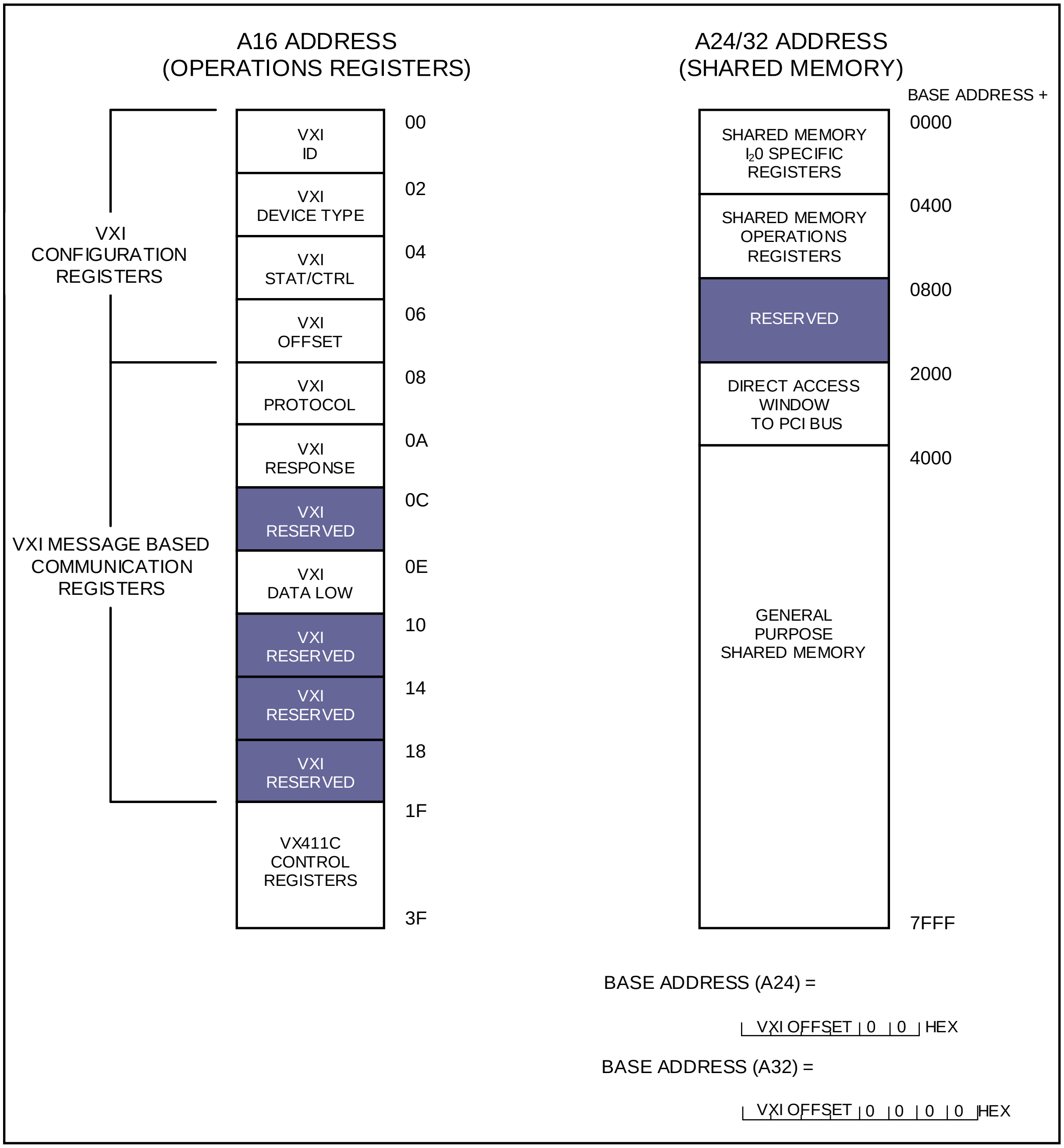


Figure 15 VXI Memory Organization

4.3.2 Data Bus Width

The intelligent carrier supports 16 and 32-bit wide data transactions to the shared memory device in the A24/A32 address space. However the device must be configured to be either 16 or 32-bits, but not both. Differences in the byte lane assignments between the VXI bus and the shared memory's local bus make dynamically switching between D16 and D32 impossible. Configuration of the data bus width is performed by configuring the shared memory device's local bus to either 16 or 32 bits. This is done using a single register in the shared memory device. Refer to the *CY7C09449PV Data Sheet* from Cypress Semiconductor, Inc. for details on configuring the width of the data bus.

Only 16-bit accesses to A16 address space are supported.

4.3.3 PCI Bus Mastering and Direct Access

The shared memory device provides an 8 Kilobyte window directly into PCI memory space. This window is accessible by the VXI host at offset 2000_{16} in A24/A32 space. This window gives the host PC direct access to any PCI device residing on the on-board primary or secondary PCI bus. To point the 8 Kilobyte window to the correct PCI bus address, the host must control the direct access control register defined by the shared memory device. The register is part of the shared memory operation registers accessible by the VXI host in A24/A32 space. Refer to section 6.5 for details on using the direct access capabilities of the carrier.

4.4 SHARED RESOURCES AND DEVICE COMMUNICATIONS

Communication between the host-side application and the device-side application is accomplished using a couple of resources available to both the host and the device. Namely, these shared resources are the operations registers and the shared memory device as shown in Figure 16. These shared resources are used to perform VXI communications, device configurations, block data transfers, and other miscellaneous functions.

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

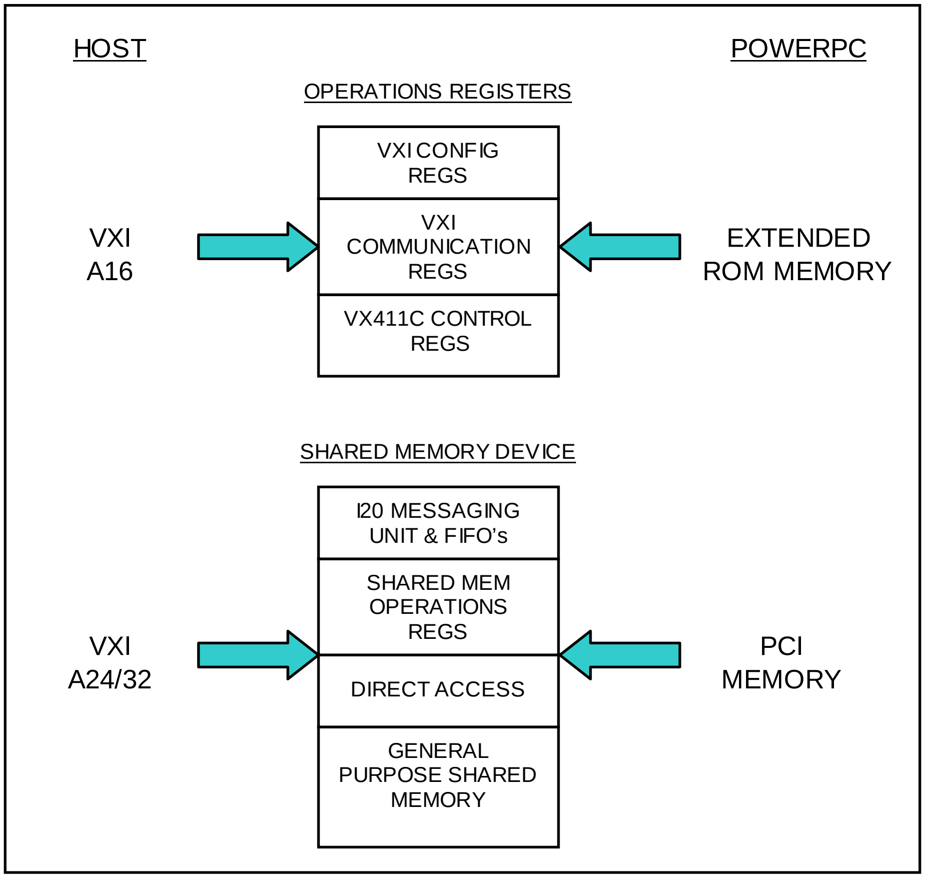


Figure 16 Shared Resources

4.4.1 Operations Registers

The operations registers combine the required VXI configuration registers, VXI communication registers and a set of carrier control registers. Table V lists all available registers along with their offset. The VXI host can access each registers at its specified offset in the A16 address space. The PowerPC can access each register at its specified offset in its extended ROM space starting at address 7000_0000_{16}. There may be access restrictions on individual registers or individual bits within a register depending on the whether it is being accessed by the host-side or the device-side application. Refer to the register descriptions in Figure 17, Figure 18, and Figure 19 for details on each register.

NOTE: The PowerPC's interface to the operations registers is only 8-bits wide. In Figure 17, Figure 18, and Figure 19, the least significant bits reside in the low PowerPC address. For example, VXI ID bits 0-7 reside at PowerPC address 0x7000_0000$_{16}$ and bits 8-15 reside at address 0x7000_0001$_{16}$.

Table V Operations Registers Map

Offset (hex)	Register Description
VXI Configuration Registers	
00	VXI ID
02	VXI Device Type
04	VXI Status/Control
06	VXI Offset Register
VXI Communication Registers	
08	VXI Protocol
0A	VXI Response
0C	Reserved
0E	VXI Data Low
10 – 1F	Reserved
VX411C Control Registers	
20	VX411C Status/Control
22	Interrupt Control
24	Reserved
26	Watchdog Register
28-3F	Reserved

4.4.1.1 VXI Configuration Registers

The VXI configuration registers contain basic information needed to configure a VXI system as required by the VXIbus specification. The configuration information includes: manufacturer identification, product model code, device type, memory requirements, device status, and device control. The registers are briefly described below and are detailed in Figure 17.

<u>VXI Identification (ID) Register</u> (00$_{16}$): This register provides the manufacturer identification, device classification (i.e., register based or message based), and the addressing mode (i.e. A32 or A24). It is a read only register from the VXI host. The PowerPC can write this register however it should be done immediately after reset prior to running VXI resource manager.

VXI Device Type Register (02_{16}): This register provides the model code identifier and required memory information. It is a read only register from the VXI host. The PowerPC can write this register however it should be done immediately after reset prior to running VXI resource manager.

VXI Status/Control Register (04_{16}): A read of this register provides the state of the VXI MODID* line and the pass, ready, and self-test status bits. A write to this register allows disabling of the SYSFAIL function and performing a reset of the carrier. This register is readable and writeable from both the VXI host and the PowerPC however, there are several access restrictions on individual bits depending on the source of the access.

VXI Offset Register (06_{16}): This register controls the offset value for addressing the A24/A32 address space. The VXI system resource manager or control module sets this value according to the memory requirements specified for this module and the memory requirements of the other instruments in the system. This register is readable and writeable from the VXI host. The PowerPC only has read capability of this register.

VXI ID

00_{16}			
Bit	15 14	13 12	11 0
PPC Write	Device Class	(read only)	Manufacturer ID
VXI Write	(read only)		
Read	Device Class	Address Space	Manufacturer ID

Device Class ⇨ Device Class (10 = Message Based, 11 = Register Based, 00 & 01 = reserved)
Address Space ⇨ Address Space (00 = A16/A24, 01 = A16/A32, 10 = reserved, 11 = A16 Only))
Manufacturer ID ⇨ Manufacturer Identification (default = $FC1_{16}$)

VXI Device Type

02_{16}		
Bit	15 12	11 0
PPC Write	(read only)	Model Code
VXI Write	(read only)	
Read	Required Memory	Model Code

Required Memory ⇨ 32 Kbytes required (F_{16} if A32, 8_{16} if A24)
Model Code ⇨ Model Code (default = $FD8_{16}$)

Figure 17 VXI Configuration Registers

VXI Status/Control

04$_{16}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPC Write	-								POST Result				RDY	PASS	-	
VXI Write	AAA	-													SI	RST
Read	AAA	MID	-						POST Result				RDY	PASS	-	

AAA ⇨ A24/A32 Access (0 = disabled)
MID ⇨ Module ID Status(0=MODID* line is asserted)
POST Result ⇨ Power On Self Test Result

0000	Passed
0001	SDRAM Failure
0010	Shared Memory Failure
0011	Flash Memory Failure
0100	Operations Register Failure
0101- 1111	Reserved

RDY ⇨ Ready(1=ready)
PASS ⇨ Pass/Fail Indicator(0=executing or failed, 1=passed)
SI ⇨ Sysfail Inhibit(1=inhibit)
RST ⇨ Reset(writing a '1' to this bit resets the carrier; after a minimum of 100μs a ‘0’ must be written to resume normal operation)

VXI Offset

06$_{16}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPC Write	(read only)															
VXI Write	A24/A32 Offset															
Read	A24/A32 Offset															

A24/A32 Offset ⇨ Offset to the carriers A24/A32 memory space

Figure 17 VXI Configuration Registers (continued)

4.4.1.2 VXI Communication Registers

The VXI communication registers are defined by the VXI specification for message based devices. They provide all the functionality necessary to perform the VXI word serial protocol. The on-board software must manage these registers to perform message passing. A word serial protocol handler should be part of any standard offering of operating system software for the VX411C. Only on very rare occasions should a user application need to access these registers directly. Figure 18 shows these registers in detail.

CAUTION: It is rarely necessary for either the host-side or the device-side user application to access the VXI communications registers directly. The host-side VXI libraries and the device-side operating system software will automatically manage these registers when performing message passing functions. Directly accessing these registers is not advised without prior knowledge of the VXI specification for message based devices.

VXI Protocol Register (08_{16}): This register indicates which message based protocols the carrier supports and indicates additional communication capabilities of the carrier. This register is a read only register by both the host and device software.

VXI Response Register ($0A_{16}$): This register indicates status of the carrier's communications capabilities. The register is read-only from the host-side application. The PowerPC can write to this register to perform message passing.

VXI Data Low Register ($0E_{16}$): This register is used to pass VXI commands and data to and from the carrier over the VXI bus. The status of this register is indicated in the VXI response register. This register is readable and writeable by both the host and the device-side software; however, the rules for message based communications as set by the VXI specification must be followed to ensure data integrity. Neither the PowerPC user application nor the VXI host applications should need to write directly to this register. The on-board operating system software should automatically manage this register and the VXI host should use standard VXI libraries to communicate with the carrier.

VXI Protocol

08_{16}

Bit	15	14	13	12	11	10	9 0
PPC Write	(read only)						
VXI Write	(read only)						
Read	CMDR*	SIG*	MSTR*	INT	FHS*	SMEM*	reserved

CMDR* ⇨ Commander (default 1=Servant only capabilities)
SIG* ⇨ Signal Register (default 1=No signal register)
MSTR* ⇨ Master (default 1=No VME bus master capabilities)
INT ⇨ Interrupter (default 1=Has interrupter capabilities)
FHS* ⇨ Fast Handshake (default 1=Does not support the Fast Handshake Mode)
SMEM* ⇨ Shared Memory (default 1=Does not support the shared memory protocol)

VXI Response

$0A_{16}$

Bit	15	14	13	12	11	10	9	8	7	6 0
PPC Write	0	rsvd	DOR	DIR	ERR*	RRDY	WRDY	FHS*	LCK*	reserved
VXI Write	(read only)									
Read	0	rsvd	DOR	DIR	ERR*	RRDY	WRDY	FHS*	LCK*	reserved

DOR ⇨ Data Out Ready (1 = message byte available to be read by VXI host)
DIR ⇨ Data In Ready (1 = carrier ready to receive message byte)
ERR* ⇨ Error (0 = error occurred, 1 = no error)
RRDY[1] ⇨ Read Ready (1 = data has been place in the data low register for read by the VXI host)
WRDY[2, 3] ⇨ Write Ready (1 = data low register is empty and ready for VXI host to write a command)
FHS* ⇨ Fast Handshake Active (0 = Fast handshake mode is active)
LCK* ⇨ Locked (0 = a commander has locked the carrier from being accessed by other sources)

Notes:

1. The Read Ready (RRDY) bit automatically cleared by the VXI interface logic when the data low register is read by the VXI host.
2. The write ready (WRDY) bit is automatically cleared by the VXI interface logic when the data low register is written by the VXI host.
3. **The Write Ready (WRDY) bit is logically tied to the PowerPC's System Management Interrupt (SMI) such that when the bit is cleared by the VXI host writing data to the Data Low register, a SMI interrupt is generated signaling to the PowerPC that data is available. This can be disabled using the VXDIS bit in the VX411C Control/Status register.**

Figure 18 VXI Communications Registers

VXI Data Low

$0E_{16}$ Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPC Write	Data Low															
VXI Write	Data Low															
Read	Data Low															

Data Low ⇨ Low Data Word

Notes:

1. The VXI specification for message based devices must be followed for reading and writing the data low register
2. The Read Ready (RRDY) bit in the VXI response register is automatically cleared by the VXI interface logic when the data low register is read by the VXI host.
3. The Write Ready (WRDY) bit in the VXI response register is automatically cleared by the VXI interface logic when the data low register is written by the VXI host.

Figure 18 VXI Communications Registers (continued)

4.4.1.3 VX411C Control Registers

The VX411C control registers provide miscellaneous configuration, status, and control functionality for the carrier. Refer to the register descriptions and Figure 19 for details.

VX411C Control/Status (20_{16}): This register provides miscellaneous status information and control functionality for the carrier. Each bit has individual read/write restrictions depending on whether the host-side or the device-side application is accessing it.

Interrupt Control (22_{16}): This register is used to configure and control the interrupt capabilities of the carrier. The PowerPC or the shared memory device can interrupt the VXI host. The host or PowerPC application can enable/disable the ability of the carrier to interrupt the VXI host and can configure the interrupt level. The PowerPC application can also set a vector that'll be passed to the VXI host and generate the interrupt.

Watchdog Timer Control (26_{16}): This register is used to configure and control the Watchdog Timer. The watchdog timer can be enabled or disabled by the PowerPC using the enable bit of this register. If enabled, the Watchdog must be reset within every 250 ms to avoid a processor reset. This register is not writable by the VXI host.

VX411C Status/Control

20_{16} Bit	15	14	13	12	11 8	7	6	5	4	3	2	1	0
PPC Write	-	-	-	-	reserved	-	VXDIS	PMC RST	BRST	-	-	-	-
VXI Write	-	-	-	-	reserved	-	VXDIS	PMC RST	BRST	-	-	-	-
Read	IN3	IN2	IN1	IN0	reserved	-	VXDIS	PMC RST	BRST	-	WDE	CFG1	USF

USF[1] ⇨ Update System Firmware (Value of the Update System Firmware configuration switch)
0 = Boot normally
1 = Boot into update system firmware mode

WDE[2] ⇨ Watchdog Enable Switch (Value of the Watchdog Enable configuration switch)
0 = Watchdog enabled
1 = Watchdog disabled

CFG1[3] ⇨ Software Configuration Switch (Value of configuration switch)
0 = Switch closed
1 = Switch open

BRST ⇨ Bridge Reset (writing a ‘1’ to this bit resets the PCI to PCI Bridge and the PCI to Cardbus Bridge; after a minimum of 100μs a ‘0’ must be written to resume normal operation)

PMC RST ⇨ PMC Reset (writing a ‘1’ to this bit resets the module in PMC slot; after a minimum of 100μs a ‘0’ must be written to resume normal operation)

VXDIS[4] ⇨ VXI Disable (writing a ‘1’ to this bit will disable the interrupt generated whenever a message byte is received over the VXI interface)

INx ⇨ TTL Input (Current level of the TTL Input signals)

Notes:

1. This bit is used by the boot code to determine whether to go into firmware update mode immediately on power-up or whether to boot normally
2. The Watchdog Enable Switch enables/disables the watchdog timer at the hardware level. The watchdog timer may also be disabled by software by writing to the Watchdog Timer Control Register (Offset 0x26). **Disabling the watchdog timer at the hardware level is required to utilize a JTAG debugger on the processor.**
3. The user configuration switch has no effect on the carrier operation. The PowerPC firmware or operating system software can use this value. The operating system may define the function of this switch. Otherwise the user application can use it however it wishes.
4. The Write Ready (WRDY) bit in the VXI response register is logically tied to the PowerPC’s System Management Interrupt (SMI) such that when the bit is cleared by the VXI host writing data to the Data Low register, a SMI interrupt is generated signaling to the PowerPC that data is available. The VXDIS bit will disable the interrupt generation. Disabling the VXI interrupt may cause the VXI message based firmware to not operate and may not be recoverable. Make sure the DIR and DOR bits in the VXI Response register are cleared before setting this bit to avoid problems.

Figure 19. VX411C Control Registers

Interrupt Control

22_{16}

Bit	15 8	7	6	5	4	3 1	0
PPC Write	Vector	PIP	-	PIE	SMIE	VXI Level	MIE
VXI Write	(read only)		-	PIE	SMIE	VXI Level	MIE
Read	Vector	PIP	SMIP	PIE	SMIE	VXI Level	MIE

Vector[1] ⇨ Upper 8 bits of the status-id value returned during an interrupt acknowledge cycle

00_{16} – $7F_{16}$	VXI response interrupt
80_{16}	reserved for shared memory interrupt
81_{16}-BF_{16}	user defined interrupt
FC_{16}	VXI request false event interrupt
FD_{16}	VXI request true interrupt
FE_{16}	reserved
FF_{16}	no cause given

PIP ⇨ Processor interrupt pending (if PIE=1 & MIE=1 then writing a 1 to this bit will generate an interrupt)

SMIP[2] ⇨ Shared memory interrupt pending (a value of 1 indicates that the shared memory device has asserted its interrupt line)

PIE ⇨ Processor interrupt enable (1 = a value of 1 in the PIP bit will generate an interrupt)

SMIE ⇨ Shared memory interrupt enable (1 = enable interrupts from the shared memory device)

VXI Level ⇨ VXI Interrupt Level (0= disabled, 1-7=IRQ 1-7)

MIE[3] ⇨ Master Interrupt Enable (1 = interrupts enabled)

Notes:

1. The vector value specifies the type of interrupt that is pending. The 'user defined interrupt' vector range may be used by the user application when generating processor interrupts. All other interrupt vector values are defined by the VXI specification or reserved by the VX411C. The operating system software may manage this register field when generating VXI defined interrupts.
2. If both the PIP and SMIP bits are set, the shared memory interrupt will have priority and a vector value of 80_{16} will be returned regardless of the value in the vector field.
3. All interrupts are Release On Acknowledge (ROAK) interrupts. This is achieved by clearing the MIE bit during the interrupt acknowledge cycle. This bit should be re-enabled prior to returning from the interrupt service routine in order for interrupts to continue to occur.

Reserved

24_{16}

Bit	15	14	13	12	11	8	7	6	5	4	3	2	1	0
PPC Write	reserved													
VXI Write	reserved													
Read	reserved													

Reserved ⇨ This register is reserved for reasons involving software compatibility with other C&H intelligent VXI carriers.

Watchdog Timer Control

26_{16}

Bit	15	14	13	12	11	8	7	6	5	4	3	2	1	0
PPC Write	reserved												RST	EN
VXI Write	(read only)													
Read	reserved												0	EN

RST[1] ⇨ Reset Watchdog Timer (Write a 1 to this bit to reset the watchdog timer)
EN[2] ⇨ Watchdog Timer Enable
0 = Disabled
1 = Enabled (default)

Notes:
1. The RST will automatically be cleared after the timer is reset.
2. The watchdog timer is enabled by default.
3. If enabled, the watchdog timer expires every 250ms unless reset. Upon expiration, the watchdog will reset the processor.

Figure 19. VX411C Control Registers (continued)

4.4.2 VXI Word Serial Protocol

Most communications between the host and the PowerPC is via the VXI word serial protocol defined by the VXI bus specification. Message passing is implemented via the VXI communication registers as described in section 4.4.1.2. The PowerPC handles the device-side of the communication protocol with the on-board system firmware.

The word serial protocol implementation is dependant upon the software environment running on the carrier. A word serial protocol handler that defines all standard word serial commands specified in the VXIbus specification. System commands are also defined to provide general access to the PMC and Cardbus modules, PowerPC utilities, and carrier configuration options. Application dependant commands can be developed per application.

4.4.3 General Purpose Shared Memory

The 16 kilobytes of general purpose shared memory can be used to pass large amounts of data between the host application and the PowerPC application. High performance, data intensive applications can take advantage of burst access to this memory space from the host and DMA access to this memory space from the PowerPC or other device on the PCI bus. The shared memory device will provide low level arbitration for this memory space. High level handshaking can be provided through message based commands and/or VXI interrupts.

The host-side application can access the general purpose shared memory at offset 4000_{16} in VXI A24/A32 address space. The on-board PowerPC application can access this memory at offset 4000_{16} in shared memory device's PCI BAR0 space. The device's BAR0 space is determined during PCI bus enumeration and can be read by performing a PCI configuration read of the shared memory device's BAR0 register.

4.4.3.1 Shared Memory Arbitration

The shared memory device provides arbitration logic so that any location can be accessed at the same time by both the VXI host via the shared memory device's local bus interface and the PowerPC via the shared memory device's PCI bus interface.

The shared memory device also provides an Arbitration Utility Flag Register accessible to the host through VXI A24/A32 space and to the PowerPC through PCI memory space. Software can use this register to implement high level memory arbitration. As shown in Figure 20, the register provides four arbitration flags that can be owned by either the local bus (VXI) or the PCI bus (PowerPC) but never both at the same time.

Arbitration Utility Flag Register

$04C0_{16}$

Bit	31 26	25	24	23 18	17	16	15 10	9	8	7 2	1	0
Write	-	L3	P3	-	L2	P2	-	L1	P1	-	L0	P0
Read	-	L3	P3	-	L2	P2	-	L1	P1	-	L0	P0

Lx ⇨ Local bus ownership (This bit can only be set by the VXI host and only if the corresponding Px bit is not set)

Px ⇨ PCI bus ownership (This bit can only be set by the PowerPC and only if the corresponding Lx bit is not set)

Figure 20 Shared Memory Arbitration Utility Flag Register

4.4.3.2 DMA/Burst

The PowerPC contains an embedded DMA controller than can be used to burst data into and out of shared memory. The destination or source of the DMA transfer can be local memory or another PCI device. Details on using the embedded DMA controller are beyond the scope of this document. Refer to the MPC8245 User's Manual for further information.

The shared memory device also contains an embedded DMA controller that can burst between the shared memory device and any PCI device. The shared memory can be programmed to perform the DMA transfer then interrupt the VXI host when the transfer is complete. The shared memory's DMA controller is fully accessible without the help of a PowerPC application.

The VXI host can burst data into and out of shared memory using VXI block transfer cycles. If supported, the host's VXI library should provide functions to perform block transfers. The data width of block transfers can be 16 or 32 bits however, the data bus width must be configured as discussed in section 4.3.2.

4.4.4 I$_2$0 Message Unit

The shared memory device has an on-board I$_2$O messaging unit that can be used to communicate between the host and the PowerPC. However, the I$_2$O messaging unit will only be used in special circumstances since the VXI Word Serial Protocol provides full message passing capabilities. Full access to the I$_2$O messaging unit is provided through VXI A24/A32 space and through PCI memory space. Refer to the Cypress CY7C09449PV Data Sheet for further details.

4.4.5 General Purpose FIFOs

The I$_2$O messaging unit contains four FIFOs that are available for general purpose use when the I$_2$O messaging unit is not being used. Each FIFO is 32 deep x 32 bits and can be accessed by both the host and the PowerPC applications. Access to the FIFOs is achieved through 4 registers that are part of the I$_2$O Specific Registers section mapped to VXI A24/A32 space and to PCI memory space. Figure 21 describes the FIFO registers. Each register is a shared port such that on a write it places data on one FIFO and on a read it reads data from a different FIFO. This way each FIFO can be accessed by both the host and the PowerPC simultaneously.

Inbound Free/Post FIFO

0040$_{16}$

Bit	31 … 0
Write	Inbound Post FIFO Write
Read	Inbound Free FIFO Read

Inbound Post FIFO Write	⇨	Place data into the Inbound Post FIFO
Inbound Free FIFO Read	⇨	Pull data from the Inbound Free FIFO

Outbound Post/Free FIFO

0044$_{16}$

Bit	31 … 0
Write	Outbound Free FIFO Write
Read	Outbound Post FIFO Read

Outbound Free FIFO Write	⇨	Place data into the Outbound Free FIFO
Outbound Post FIFO Read	⇨	Pull data from the Outbound Post FIFO

Figure 21 General Purpose FIFO Registers

Inbound Post/Free FIFO

0048_{16}

Bit	31	0
Write	Inbound Free FIFO Write	
Read	Inbound Post FIFO Read	

Inbound Free FIFO Write	⇨	Place data into the Inbound Free FIFO
Inbound Post FIFO Read	⇨	Pull data from the Inbound Post FIFO

Outbound Free/Post FIFO

$004C_{16}$

Bit	31	0
Write	Outbound Post FIFO Write	
Read	Outbound Free FIFO Read	

Outbound Post FIFO Write	⇨	Place data into the Outbound Post FIFO
Outbound Free FIFO Read	⇨	Pull data from the Outbound Free FIFO

Notes:

1. Writing to a full FIFO will result in loss of data. The contents of the FIFO will not change.
2. Reading from an empty FIFO will return $FFFFFFFF_{16}$.
3. All FIFOs are empty at reset.

Figure 21 General Purpose FIFO Registers (continued)

5.0 SOFTWARE ARCHITECTURE

For a typical application, the system software will consist of both an on-board application running on the PowerPC and a host application running on the VXI host computer. The two applications will communicate over the VXI bus using the shared resources of the VX411C described in section 4.4. Figure 22 illustrates the system architecture for a typical application.

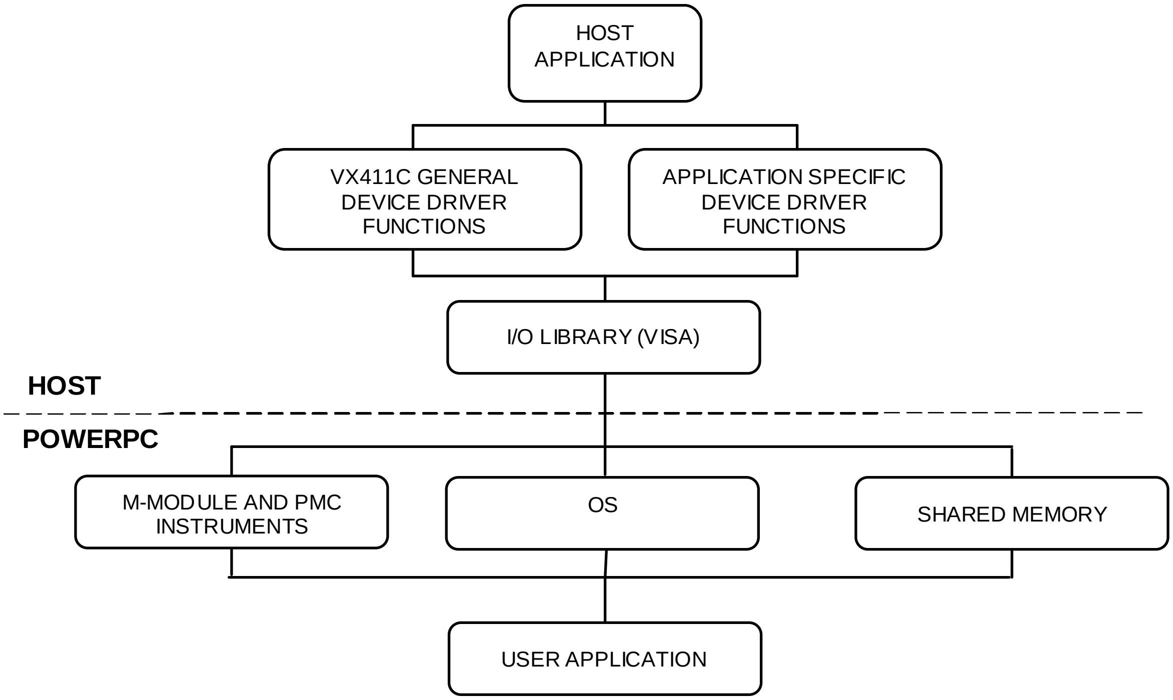

HOST APPLICATION
VX411C GENERAL DEVICE DRIVER FUNCTIONS
APPLICATION SPECIFIC DEVICE DRIVER FUNCTIONS
I/O LIBRARY (VISA)
HOST
POWERPC
M-MODULE AND PMC INSTRUMENTS
OS
SHARED MEMORY
USER APPLICATION

Figure 22 System Software Architecture

5.1 HOST SYSTEM SOFTWARE

The host-side application will normally run on a standard PC or an embedded VXI controller and will communicate with the device using standard off-the-shelf VXI interfaces and software libraries. The application can be part of a large automated test system responsible for controlling the VX411C along with numerous other instruments, or it can be an independent diagnostic application allowing the user to interact with the VX411C only. The typical application will be responsible for sending high level commands to the PowerPC application and retrieving, analyzing, and reporting data. It might also provide a soft front panel that a user may use to interact with the instrument.

The device driver library will provide high level functions to communicate with the instrument. The library can be separated into VX411C general carrier driver functions and application specific driver functions. The general carrier functions will provide functionality that exists regardless of the M/MA-modules and PowerPC application residing on the board. The application specific functions will be developed along with the PowerPC software and will be specific to the given application.

This document does not attempt to define the scope or the architecture of the host application. It is only mentioned here to illustrate how the application would interact with the software running on-board the VX411C's PowerPC system.

5.2 ON-BOARD SYSTEM FIRMWARE

The on-board system firmware will usually consist of boot code, an operating system or kernel, and a user application. The boot code is responsible for initializing the processor and launching the operating system. Once launched, the operating system is responsible for initializing of the carrier, launching the user application, and performing various system level tasks during execution. The user application will perform many of the high level tasks that are unique to the function of the carrier in the overall system application.

5.2.1 Boot Code

The VX411C boot code is located in the carrier's 32 kilobyte boot ROM device. At reset, the PowerPC automatically jumps to the reset exception vector at address FFF0_0100$_{16}$ located in boot ROM and begins running the boot code. The boot code is responsible for initializing the PowerPC to a state where a minimal application can run, and launching the operating system or other system firmware. The boot ROM also contains code to download and update the on-board firmware via the VXI bus.

5.2.1.1 PowerPC Initialization

The first action the VX411C boot code performs is to initialize the PowerPC and all its associated peripheral interfaces. The MPC8245 is a highly integrated processor with the capability of interfacing to many different peripherals. An extensive set of configuration registers are provided to configure the processor for all the different possible architectures. During the boot sequence, these registers are initialized to values appropriate for the VX411C hardware architecture. Detailed settings for each configuration option are beyond the scope of this document. However, a list of configuration registers and the value that each is initialized to, is provided in APPENDIX

B. For further details on the processor configuration, refer to the MPC8245 User's Manual.

5.2.1.2 Launching the Operating System

Once the boot code has initialized the PowerPC to a state where it can run a minimal application, it will attempt to launch the operating system. This is accomplished by initializing the PowerPC's registers that determine how the processor returns from an exception and performing a *return from interrupt* (rfi) instruction. The registers are setup such that after the rfi instruction is executed, the processor immediately begins executing the instruction at address FF00_1000$_{16}$. This address is located in the first sector of the flash device. The first instruction of the operating system or other system firmware must exist at this location.

Essentially, the boot code performs a blind jump to address FF00_0000$_{16}$ leaving the carrier in a fairly raw, un-initialized state. Besides initializing the PowerPC processor, the boot code does not configure the carrier for operation. It is the responsibility of the system firmware to configure the carrier including: enumerating the PCI bus, initializing the VXI interface, configuring the shared memory device, and initializing stack, data, and other memory pointers. The register initializations shown in APPENDIX B outline the extent of initialization efforts performed by the boot code.

5.2.1.3 Firmware Download Utility

If the 'Firmware Update Mode' hardware switch is set to the OFF position at reset, the boot code will automatically launch the firmware download utility instead of the system firmware as described in the previous section. The firmware download utility allows the user to download firmware and data into the flash device via the VXI bus using the shared memory device as a buffer for the data. This mode is useful for programming a blank flash device or updating the embedded operating system. Once in the download utility mode, the carrier must be reset and the 'Firmware Update Mode' switch must be set back to the ON position for normal operation to continue. For details on the download protocol for updating firmware refer to section 6.3. For details on the 'Firmware Update Mode' hardware switch, refer to section 3.4.2.

5.2.2 User Application

The user application will be responsible for performing higher level tasks specific to the integrated carrier and M-modules. The user application will normally communicate with the carrier and associated M-modules via the operating system. It will typically be stored in Flash memory or downloaded over the VXI bus at reset. Launching of the user application and restrictions on the resources available to the user application is dependant upon the operating system or other system firmware running on the carrier.

6.0 PROGRAMMING INSTRUCTIONS

6.1 FLASH PROGRAMMING

Programming a single byte in flash requires sending a stream of commands to the device. The OS may provide API system calls and VXI commands to perform flash programming so that the user does not need to know the specifics of the device protocol. Refer to the *Am29LV065D Data Sheet* from Advanced Micro Devices (AMD) for details on programming the flash device.

The device is organized into sectors that are 64 kilobytes each. The sectors are organized sequentially in memory so that sector 0 is from address 00_0000_{16} – 00_FFFF_{16}, sector 1 is from address 01_0000_{16} – 01_FFFF_{16} and so forth. The 64 kilobytes are mapped to the PowerPC address space starting at address $FF00\_0000_{16}$. The operating system or other system firmware may reserve some of the sectors for system use. Refer to the firmware's documentation for details.

Programming operations can be performed on any address in the flash device. Only 8-bit accesses are supported. ***The programming operation can only toggle a bit from '1' to '0'***. To set a bit back to '1' an erase operation must be performed. Erase operations can only be performed on a sector by sector basis. Therefore, in most cases it is necessary to erase an entire sector and rewrite it to change a single byte within that sector.

6.2 PCI ACCESSES

Devices on the PCI bus are mapped into the PowerPC's address space. Therefore, accessing these devices (except for configuration accesses) is as simple as performing a standard memory read or write. The base address for the device, in the PowerPC's memory space, is determined during PCI bus enumeration and is dependant upon the resources required by all the devices on the bus. A PCI configuration read can be used to determine the base address of a specific device. Every PCI device is required to have a set of Base Address Registers (BAR) that the PCI controller configures during bus initialization. These registers determine the base address(s) of the resource(s) on the device. Each BAR can point to either PCI memory space or PCI I/O space. If bit 0 (the least significant bit) of the base address register is a '1', the resource is mapped to PCI I/O space. Otherwise the resource is mapped to PCI memory space. PCI memory space is mapped directly into the PowerPC's address map (i.e. PowerPC address 9000_0000_{16} is mapped to the same address in PCI memory space). PCI I/O space, however, is mapped relative to a base address of $FE00\_0000_{16}$ (i.e. PowerPC address $FE80\_0000_{16}$ is mapped to PCI I/O address 0080_0000_{16}). For details on the PowerPC address map refer to Figure 11. For information on a particular device's BAR registers, refer to the PCI device's documentation.

Two accesses are required to perform a single PCI configuration write or read. The PCI configuration address register at PowerPC address FEC0_0000$_{16}$ must first be set to point to the correct device and offset. Then the data can be read from or written to the PCI configuration data register at PowerPC address FEE0_0000$_{16}$. The PCI configuration address register value is determined by the device number, IDSEL signal routing, device function number, and the register offset. IDSEL signal routing information can be found in section 4.2.5.1 of this document. For further details on performing PCI configuration accesses refer to the MPC8245 User's Manual.

6.3 FIRMWARE UPDATE MODE

The firmware update mode allows the user to download code and data over the VXI bus via shared memory. The destination of the download can be flash, or any other storage location in the PowerPC's address space. The PowerPC will automatically go into firmware update mode when the 'Firmware Update Mode' hardware switch is set to the OFF position at reset. When the update is complete, the VX411C must be restarted and the switch must be set back to the ON position for normal operation to continue.

6.3.1 Firmware Update Mode Protocol

The firmware update routines use the general purpose shared memory as a buffer between the host and the PowerPC. The protocol divides the 16 kilobytes of memory into four 4 kilobyte banks as shown in Figure 23. Each bank has two associated ownership bits in the arbitration utility flag register at shared memory offset 4C0$_{16}$. One bit signifies ownership of the associated bank by the host and the other signifies ownership of the bank by the PowerPC. The register functions such that the host cannot take ownership of a bank that the PowerPC has ownership of and vice-versa. Software must guarantee that it does not write to or read from a bank that it does not have ownership of. Further details of the arbitration utility flag register are discussed in section 4.4.3.1.

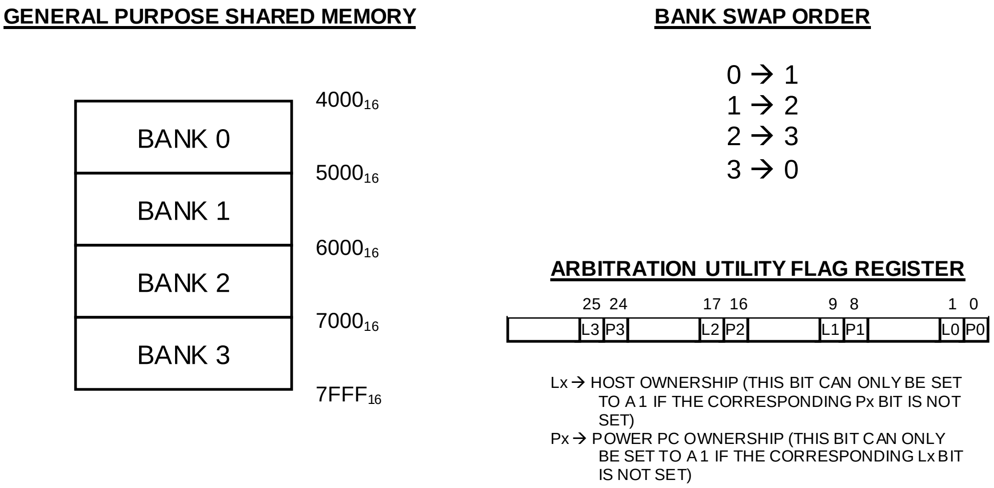

GENERAL PURPOSE SHARED MEMORY
BANK 0
BANK 1
BANK 2
BANK 3
4000_{16}
5000_{16}
6000_{16}
7000_{16}
$7FFF_{16}$
BANK SWAP ORDER
0 → 1
1 → 2
2 → 3
3 → 0
ARBITRATION UTILITY FLAG REGISTER
25 24
17 16
9 8
1 0
L3
P3
L2
P2
L1
P1
L0
P0
Lx → HOST OWNERSHIP (THIS BIT CAN ONLY BE SET TO A 1 IF THE CORRESPONDING Px BIT IS NOT SET)
Px → POWER PC OWNERSHIP (THIS BIT CAN ONLY BE SET TO A 1 IF THE CORRESPONDING Lx BIT IS NOT SET)

Figure 23 Shared memory banks for firmware update

The download protocol is illustrated in the flow chart of Figure 24. The host computer takes ownership of bank 0 to signify that it is ready to download data. The PowerPC application will wait until it has verified that bank 0 is no longer free. The PowerPC application then takes ownership of bank 3 to signify that it is ready. Like the PowerPC application, the host application must wait until it has verified that bank 3 is no longer free. At this point the download can begin.

The format of the data depends on the type of download command being performed. In all cases the data starts with a command code and ends with a Cyclic Redundancy Check (CRC) value. To transfer the data, the host simply places it in consecutive address locations of shared memory. When the host fills an entire bank, it must perform a bank swap by first, verifying the next bank is free, second, taking ownership of the bank, and finally, releasing ownership of the completed bank. When a bank becomes free, the PowerPC will take ownership of it prior to reading the data. Like the host, the PowerPC application will verify that it has ownership of the next bank prior to releasing the current bank. Bank swapping must be done in the order illustrated in Figure 23 in order to preserve data integrity. When the download is complete the PowerPC application verifies the CRC value and returns an acknowledge (ACK) or negative acknowledge (NACK) value to the host. Both the host and the PowerPC must release all bank ownerships prior to returning from their respective download routines.

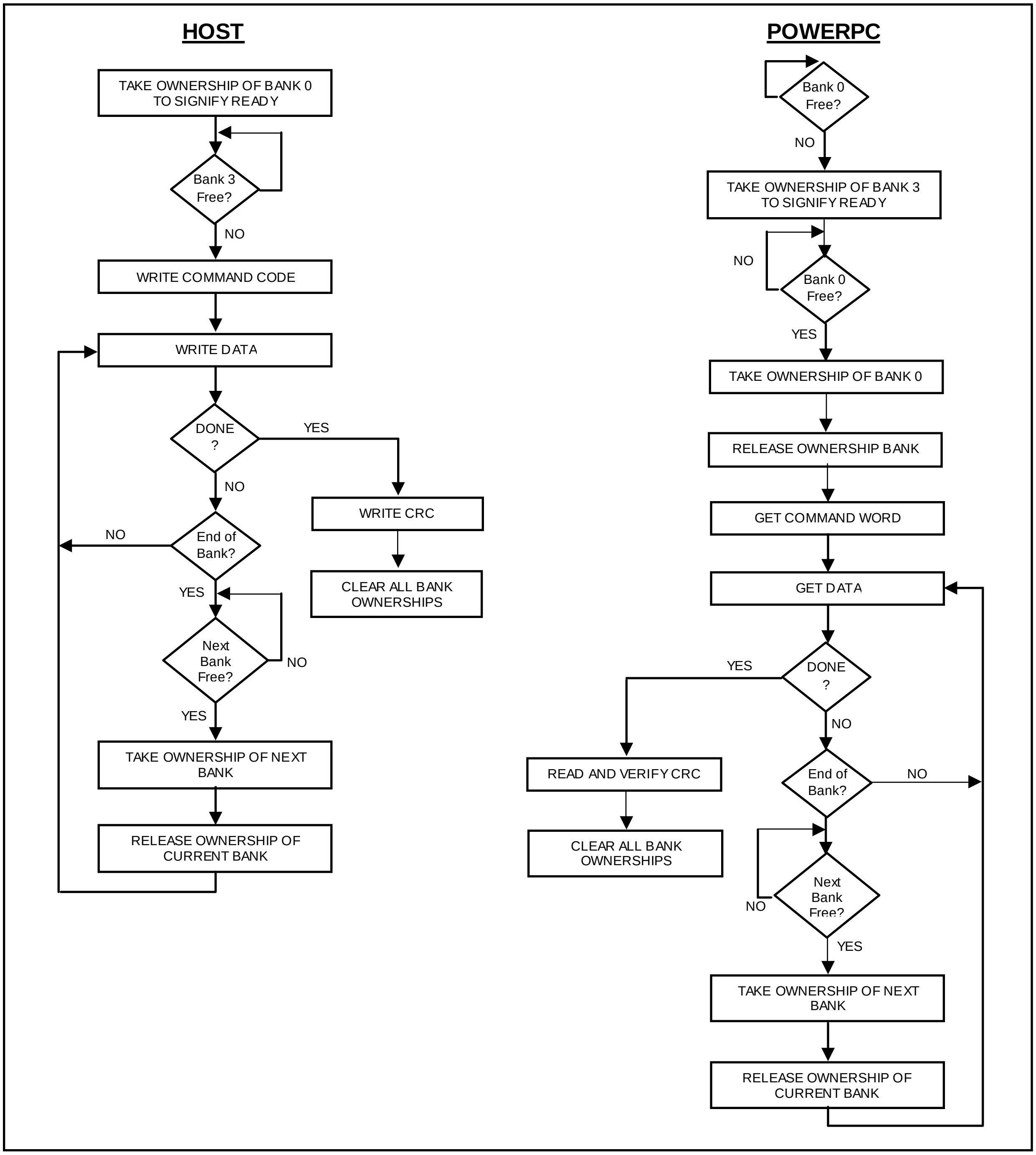


Figure 24 Firmware Update Protocol

6.3.2 Download Commands

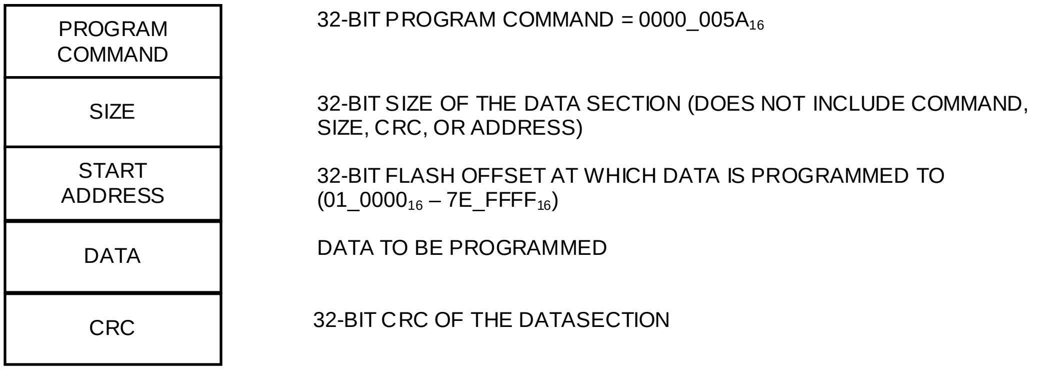
Download commands identify the format of the data being downloaded and instruct the PowerPC what to do with the data. Each command format begins with a 32-bit command code and ends with a 32-bit CRC value. The data between the command code and the

CRC can be a variable sizes and meaning depending on the command. The download protocol illustrated in Figure 24 must be followed for each download command. The behavior of the PowerPC firmware after the command is complete is dependant upon the type of command.

6.3.2.1 Flash Program Command

The flash program command allows the user to download data to be written to the flash device. The PowerPC behaves exactly as it does with the generic download command except that instead of performing a simple memory write, it performs a flash write. The start address value must point to an offset within the flash device and not an absolute PowerPC address. Also, the start address must not be within the first sector of the flash device unless the PowerPC is in a special update system firmware mode as discussed in section 6.3.

The PowerPC does not erase the sector prior to programming the flash unless it is in the update system firmware mode. The flash sector erase command is provided for that purpose. Since the flash write function cannot toggle a bit from a ‘0’ to a ‘1’ the operation might fail if the sector has not been erased before this command is received. The flash program command is illustrated in Figure 25.


PROGRAM COMMAND
32-BIT PROGRAM COMMAND = 0000_005A_{16}
SIZE
32-BIT SIZE OF THE DATA SECTION (DOES NOT INCLUDE COMMAND, SIZE, CRC, OR ADDRESS)
START ADDRESS
32-BIT FLASH OFFSET AT WHICH DATA IS PROGRAMMED TO (01_0000_{16} – $7E\_FFFF_{16}$)
DATA
DATA TO BE PROGRAMMED
CRC
32-BIT CRC OF THE DATASECTION

***Note:** The start address may be within the first sector of flash (00_0000_{16} – 00_FFFF_{16}) only if the processor is in the update system firmware mode.*

Figure 25 Flash Program Command

6.3.3 Flash Sector Erase Command

The flash sector erase command instructs the PowerPC to erase the specified sector in the flash device. A flash write operation can only toggle a bit from a ‘1’ to a ‘0’. An erase operation has to be performed to toggle and bits back to a ‘1’. A sector is the smallest block of flash memory that can be erased by the erase operation. It is recommended that the sectors being programmed are erased prior to sending the flash program command. The format of the flash sector erase command is illustrated in Figure 26.

SECTOR ERASE COMMAND	32-BIT SECTOR ERASE COMMAND = 0000_0096_{16}
SECTOR #	32-BIT SECTOR # 0000_0001_{16} – 0000_007F_{16}
1'S COMP OF SECTOR #	32-BIT 1'S COMPLEMENT OF SECTOR # $FFFF\_FFFE_{16}$ – $FFFF\_FF80_{16}$
CRC	32-BIT CRC OF SECTOR # AND 1'S COMPLEMENT OF SECTOR #

Note: ***The sector # may specify the first sector (sector 0) only if the processor is in the update system firmware mode.***

Figure 26 Flash Sector Erase Command

6.3.3.1 Calculating the CRC

The Cyclic Redundancy Check (CRC) value must be a 32-bit value that is calculated based on the 'Data' section of the 'Flash Program' command. The command code, size and start address field must not be used in the CRC calculation. An accurate calculation of the CRC must be performed in order for the firmware update to complete successfully. The example source code in Figure 27 can be used to calculate an accurate CRC. The exact CRC Polynomial shown in the example must be used.

```
#define  CRC_POLYNOMIAL  0x04C11DB7

void CalculateCRC(unsigned char *buffer, int size, int *crc)
{  int i, j;
   int *crc_table;
   int index;

   crc_table = malloc(sizeof(int) * 256);

   /*  Build CRC Table  */
   for(i=0; i<256; i++)
   {  *crc = i;
      for(j=8; j>0; j--)
      {  if (((*crc) & 0x1) == 1)
            *crc = ((*crc) >> 1) CRC_POLYNOMIAL;
         else
            *crc = (*crc) >> 1;
      }
      crc_table[i] = *crc;
   }

   /*  Calculate CRC  */
   *crc = 0;
   i = 0;
   while (size-- != 0)
   {  index = ((*crc) >> 24) & 0xFF;
      *crc = ((*crc) << 8) | buffer[i++];
      *crc ^= crc_table[index];
   }

   free(crc_table);
}
```

Figure 27. CRC Calculation - Example Source Code

6.4 INTERRUPTS

The PowerPC application will handle all interrupts from the on-board PCI devices. In addition, the PowerPC can generate a VXI interrupt by accessing an operations register. Finally, the shared memory device can also interrupt the VXI host on various programmable conditions such as the completion of a DMA transfer.

6.4.1 PCI Interrupts

PCI interrupts are handled by the Embedded Programmable Interrupt Controller (EPIC) of the MPC8245. The interrupt lines are routed between the PCI devices and the EPIC's interrupt lines as described in Table IV. Each interrupt line must be set to direct input mode so that the EPIC controller can respond to interrupts from the PCI devices. The interrupt lines can also be prioritized and can be programmed for level or edge sensitivity and either polarity.

Further details on programming the EPIC controller can be found in the MPC8245 User's Manual.

6.4.2 VXI Interrupts

The source of VXI interrupts can be the on-board firmware, or the shared memory device. In all cases the ‘Interrupt Control’ register at offset 22_{16} in the VX411C’s operations register provides the control for the interrupt. All cases share a single interrupt line and the cause of the interrupt can be determined from the status/ID value returned to the host during the interrupt acknowledge cycle.

The interrupt priority level used by the VX411C is programmable using the ‘VXI Level’ field of the interrupt control register. Any level between 1 and 7 can be selected. Writing a value of ‘0’ to this field will disable the VXI interrupts. The Master Interrupt Enable (MIE) bit must also be set to a value of ‘1’for interrupts to occur. In addition, there is an interrupt enable bit for shared memory interrupts and one for processor based interrupts.

The VXI interrupt is always automatically released by the VX411C during the interrupt acknowledge cycle (i.e. ROAK). To achieve this behavior, the carrier automatically clears the MIE bit during the acknowledge cycle. The interrupt handler routine should re-enable this bit, after performing the interrupt handling functions, if interrupts are to continue to occur. ***If an interrupt is still pending when the MIE bit is re-enabled another interrupt will immediately be generated.*** The method of clearing a pending interrupt is dependent upon the type of interrupt.

The on-board firmware can generate VXI interrupts by setting the interrupt vector value in the ‘Interrupt Pending’ register and setting the PIP bit to a ‘1’. Some vector values are reserved for specific VXI defined interrupts. Others are available for definition by the user. Figure 19 show the defined vector values. The Processor Interrupt Enable (PIE) bit must also be set to a ‘1’. Clearing the interrupt is achieved by setting the PIP bit to a ‘0’. The method for instructing the PowerPC to clear the interrupt is application specific.

The shared memory device can interrupt the VXI host for several reasons including to signify the completion of a DMA transaction. If the Shared Memory Interrupt Enable (SMIE) bit is set to a ‘1’ all interrupts from the shared memory device will be forwarded to the VXI bus. Shared memory interrupts have priority of processor based interrupts. The vector returned during the interrupt acknowledge cycle will always be 80_{16}. Configuring the shared memory device to generate interrupts is done through the device’s operation registers.

6.5 HOST-SIDE PCI BUS MASTERING AND DIRECT ACCESS

The shared memory device provides an 8 Kilobyte window into PCI memory space. By accessing this window in VXI A24/A32 space, the host PC has direct access to the PMC module and all four M-modules. To point the 8 Kilobyte window to the correct PCI bus address the host must control the direct access operational register defined by the shared memory device. Figure 28 shows a description of the control register.

Direct Access Register

0460_{16}

Bit	31 13	12	11	10	9 8	7 4	3	2 1	0
Write	PCI Physical Base Address	-	F	-	A1A0	Byte Enables for Reads	-	Type	-
Read	PCI Physical Base Address	-	F	-	A1A0	Byte Enables for Reads	-	Type	-

PCI Physical Base Address ⇨ PCI physical base address of the 8K byte direct access window at VXI A24/A32 offset 0x2000

F ⇨ Force contents of A1A0 to PCI during PCI address phase (0 = don't force, 1 = force)

A1A0 ⇨ Value of PCI A1 and PCI A0 to be placed on the PCI bus if F = 1

Byte Enables for Reads ⇨ Data Byte Enables for PCI Master Reads, C/BE#[3:0]

Type ⇨ PCI Command Type for PCI Master Access

00 Interrupt Acknowledge (read) (PCI command 0x0)
Special Cycle (write) (PCI command 0x1)
01 I/O Cycle (read/write) (PCI command 0x2 or 0x3)
10 Memory Cycle (read/write) (PCI command 0x6 or 0x7)
11 Configuration Cycle (read/write) (PCI command 0xA or 0xB)

Figure 28 Direct Access Control Register

Before the shared memory device can generate a PCI access its Master Enable bit must be set in the PCI configuration registers. This bit is set by default after power-up during the configuration of the shared memory device. The master enable bit can only be cleared by the PowerPC. If for some reason this occurs the VXI host must request that the PowerPC re-enable this bit before it can perform a direct access. Even if the PowerPC's boot procedure fails, the Master Enable bit will be set by default and the VXI host will automatically gain control of the PCI bus.

The procedure for directly accessing the onboard PCI bus is as follows:

1) Make sure the Master Enable Bit in the shared memory's PCI configuration space is set.
2) Program the Direct Access Register with the desired PCI system's physical base address. This is the 8 kilobyte address block that the Direct Access Window points to.
2a) In the Direct Access Register, set the type of PCI command to be generated.
2b) In the Direct Access Register, set the byte enables for the desired accesses if they are to be read accesses.
3) Access the PCI bus by reading or writing to the 8 kilobyte Direct Accesses Window.

Note: Step 2 should be repeated if a different 8 kilobyte area of PCI space needs to be accessed or if a different access type or byte enables are needed.

6.5.1.1 PCI Configuration Accesses

PCI configuration cycles use a different addressing method than normal PCI command cycles. During the cycle, each device is selected by asserting a unique IDSEL line. The PCI specification does not stipulate the routing of the IDSEL signals however in most systems the IDSEL line for a given device is connected to one of the upper PCI address bits. Refer to Table II in Section 4.2.5 for details on IDSEL signal routing on the VX411C.

To perform a type 0 PCI configuration cycle perform the following steps:

1) Determine which address bit must be asserted in order to assert the IDSEL line of the desired device. Use Table II as a reference.

2a) Set the correct bit in the PCI Physical Base Address section of the Direct Access Register. Only set one bit to select the PCI device.

2b) Set the 'F' bit (bit 11) in the Direct Access Register to 0.

2c) Set the access type in the Direct Access Register to '11' which specifies a configuration cycle

3) Access the 8 kilobyte direct access window at an offset that incorporates the function number (bits 10-8) and the register offset (bits7-0). Determine the A24/A32 offset to read or write to using the following formula:

$$\textit{A24/A32 Offset} = 2000_{16} + (\textit{FuncNum} \times 256) + \textit{Reg Offset}$$

6.5.1.2 Byte Enables in a Direct Access Cycle

The byte enable field of the Direct Access Register is only used during a read access. Byte enable for write accesses are determined by the type of VXI bus access. The byte enable field is applied directly to the C/BE# signals on the PCI bus during direct access reads from the VXI host. Since the PCI byte enables are active low, the byte enable bits of the control register are active low. If a target supports pre-fetching, it will return all bytes regardless of byte enables.

APPENDIX A CONNECTORS

PIN	C	B	A
1	D08	-	D00
2	D09	-	D01
3	D10	-	D02
4	D11	BG0IN*	D03
5	D12	BG0OUT*	D04
6	D13	BG1IN*	D05
7	D14	BG10UT*	D06
8	D15	BG2IN*	D07
9	GND	BG20UT*	GND
10	SYSFAIL*	BG3IN*	-
11	-	BG3OUT*	-
12	SYSRESET*	-	DS1*
13	LWORD*	-	DS0*
14	AM5	-	WRITE*
15	A23	-	-
16	A22	AM0	DTACK*
17	A21	AM1	_
18	A20	AM2	_
19	A19	AM3	_
20	A18	GND	IACK*
21	A17	-	IACKIN*
22	A16	-	IACKOUT*
23	A15	GND	AM4
24	A14	IRQ7*	A07
25	A13	IRQ6*	A06
26	A12	IRQ5*	A05
27	A11	IRQ4*	A04
28	A10	IRQ3*	A03
29	A09	IRQ2*	A02
30	A08	IRQ1*	A01
31	+12 V	-	-12 V
32	+5 V	+5 V	+5 V

Figure A-1 VXI P1 Pin Configuration

PIN	C	B	A
1	-	+5V	-
2	-	GND	-
3	GND	-	-
4	-	A24	GND
5	-	A25	-
6	-	A26	-
7	GND	A27	-
8	-	A28	-
9	-	A29	-
10	GND	A30	GND
11	-	A31	-
12	-	GND	-
13	-	+5V	-
14	-	D16	-
15	-	D17	-
16	GND	D18	GND
17	-	D19	-
18	-	D20	-
19	-	D21	-
20	-	D22	-
21	-	D23	-
22	GND	GND	GND
23	TTLTRG1*	D24	TTLTRG0*
24	TTLTRG3*	D25	TTLTRG2*
25	GND	D26	+5V
26	TTLTRG5*	D27	TTLTRG4*
27	TTLTRG7*	D28	TTLTRG6*
28	GND	D29	GND
29	-	D30	-
30	GND	D31	MODID
31	-	GND	GND
32	-	+5V	-

Figure A-2 VXI P2 Pin Configuration

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

P3				P4			
PIN	Signal	PIN	Signal	PIN	Signal	PIN	Signal
1	TCK	2	-12V	1	+12V	2	TRST#
3	GND	4	INTA#	3	TMS	4	TDO
5	INTB#	6	INTC#	5	TDI	6	GND
7	BUSMODE1#	8	+5V	7	GND	8	*PCI-RSVD*
9	INTD#	10	*PCI-RSVD*	9	*PCI-RSVD*	10	*PCI-RSVD*
11	GND	12	+3.3Vaux	11	BUSMODE2#	12	+3.3V
13	CLK	14	GND	13	RST#	14	BUSMODE3#
15	GND	16	GNT#	15	+3.3V	16	BUSMODE4#
17	REQ#	18	+5V	17	*PME#*	18	GND
19	V(I/O)	20	AD[31]	19	AD[30]	20	AD[29]
21	AD[28]	22	AD[27]	21	GND	22	AD[26]
23	AD[25]	24	GND	23	AD[24]	24	+3.3V
25	GND	26	C/BE[3]#	25	IDSEL	26	AD[23]
27	AD[22]	28	AD[21]	27	+3.3V	28	AD[20]
29	AD[19]	30	+5V	29	AD[18]	30	GND
31	V(I/O)	32	AD[17]	31	AD[16]	32	C/BE[2]#
33	FRAME#	34	GND	33	GND	34	*PMC-RSVD*
35	GND	36	IRDY#	35	TRDY#	36	+3.3V
37	DEVSEL#	38	+5V	37	GND	38	STOP#
39	GND	40	LOCK#	39	PERR#	40	GND
41	*PCI-RSVD*	42	*PCI-RSVD*	41	+3.3V	42	SERR#
43	PAR	44	GND	43	C/BE[1]#	44	GND
45	V(I/O)	46	AD[15]	45	AD[14]	46	AD[13]
47	AD[12]	48	AD[11]	47	*M66EN*	48	AD[10]
49	AD[09]	50	+5V	49	AD[08]	50	+3.3V
51	GND	52	C/BE[0]#	51	AD[07]	52	*PMC-RSVD*
53	AD[06]	54	AD[05]	53	+3.3V	54	*PMC-RSVD*
55	AD[04]	56	GND	55	*PMC-RSVD*	56	GND
57	V(I/O)	58	AD[03]	57	*PMC-RSVD*	58	*PMC-RSVD*
59	AD[02]	60	AD[01]	59	GND	60	*PMC-RSVD*
61	AD[00]	62	+5V	61	*ACK64#*	62	+3.3V
63	GND	64	*REQ64#*	63	GND	64	*PMC-RSVD*

Note: Italicized words are unused signals.

Figure A-3 PMC Pin Configuration

P6				P5			
PIN	Signal	PIN	Signal	PIN	Signal	PIN	Signal
1	*PCI-RSVD*	2	*GND*	1	I/O	2	I/O
3	*GND*	4	*C/BE[7]#*	3	I/O	4	I/O
5	*C/BE[6]#*	6	*C/BE[5]#*	5	I/O	6	I/O
7	*C/BE[4]#*	8	*GND*	7	I/O	8	I/O
9	*V(I/O)*	10	*PAR64*	9	I/O	10	I/O
11	*AD[63]*	12	*AD[62]*	11	I/O	12	I/O
13	*AD[61]*	14	*GND*	13	I/O	14	I/O
15	*GND*	16	*AD[60]*	15	I/O	16	I/O
17	*AD[59]*	18	*AD[58]*	17	I/O	18	I/O
19	*AD[57]*	20	*GND*	19	I/O	20	I/O
21	*V(I/O)*	22	*AD[56]*	21	I/O	22	I/O
23	*AD[55]*	24	*AD[54]*	23	I/O	24	I/O
25	*AD[53]*	26	*GND*	25	I/O	26	I/O
27	*GND*	28	*AD[52]*	27	I/O	28	I/O
29	*AD[51]*	30	*AD[50]*	29	I/O	30	I/O
31	*AD[49]*	32	*GND*	31	I/O	32	I/O
33	*GND*	34	*AD[48]*	33	I/O	34	I/O
35	*AD[47]*	36	*AD[46]*	35	I/O	36	I/O
37	*AD[45]*	38	*GND*	37	I/O	38	I/O
39	*V(I/O)*	40	*AD[44]*	39	I/O	40	I/O
41	*AD[43]*	42	*AD[42]*	41	I/O	42	I/O
43	*AD[41]*	44	*GND*	43	I/O	44	I/O
45	*GND*	46	*AD[40]*	45	I/O	46	I/O
47	*AD[39]*	48	*AD[38]*	47	I/O	48	I/O
49	*AD[37]*	50	*GND*	49	I/O	50	I/O
51	*GND*	52	*AD[36]*	51	I/O	52	I/O
53	*AD[35]*	54	*AD[34]*	53	I/O	54	I/O
55	*AD[33]*	56	*GND*	55	I/O	56	I/O
57	*V(I/O)*	58	*AD[32]*	57	I/O	58	I/O
59	*PCI-RSVD*	60	*PCI-RSVD*	59	I/O	60	I/O
61	*PCI_RSVD*	62	*GND*	61	I/O	62	I/O
63	*GND*	64	*PCI-RSVD*	63	I/O	64	I/O

Note: Italicized words are unused signals.

Figure A-4 PMC Pin Configuration (continued)

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

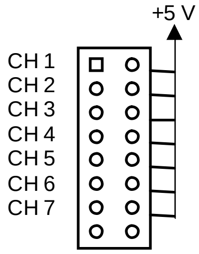


Figure A-5 External Driver Outputs (J9)

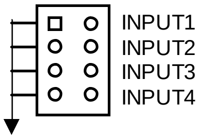


Figure A-6 External Inputs (J7)

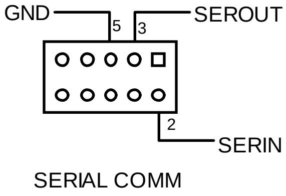


Figure A-7. Serial Comm

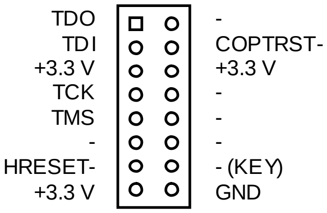


Figure A-8 JTAG/COP Header (J8)

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

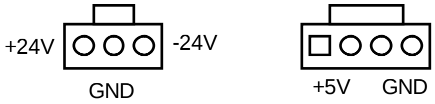


Figure A-9 External Power Connectors

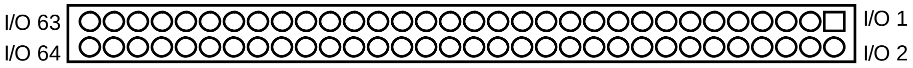


Figure A-10 PMC I/O Connector

APPENDIX B CONFIGURATION REGISTERS

Table B-1 PowerPC Configuration Registers

Device ID		Vendor ID		00
PCI Status		PCI Command		04
Class Code	Subclass Code	Standard Programming	Revision ID	08
BIST Control	Header Type	Latency Timer	Cache Line Size	0C
Local Memory Base Address Register				10
Peripheral Control And Status Register Base Address Register				14
Local Memory Base Address Register 1				18
Subsystem ID		Subsystem Vendor ID		2C
Expansion Rom Base Address				30
MAX LAT	MIN GNT	Interrupt Pin	Interrupt Line	3C
Reserved		Subordinate Bus #	Bus Number	40
PCI Arbiter Control		PCI General Control		44
Output Driver Control	PMCR2	PMCR1		70
Misc Driver Control Reg 2	Misc Driver Control Reg 1	Clock Driver Control Register		74
Embedded Utilities Memory Block Base Address Register				78
Memory Starting Address				80
Memory Starting Address				84
Extended Memory Starting Address				88
Extended Memory Starting Address				8C
Memory Ending Address				90
Memory Ending Address				94
Extended Memory Ending Address				98
Extended Memory Ending Address				9C
Memory Page Mode	Reserved		Memory Bank Enable	A0
Processor Interface Configuration Register 1				A8
Processor Interface Configuration Register 2				AC
Reserved		ECC Single-Bit Trigger	ECC Single-Bit Counter	B8
Proc. Bus Error Status	Reserved	Error Detection 1	Error Enabling 1	C0
PCI Bus Error Status	Reserved	Error Detection 2	Error Enabling 2	C4
Processor/PCI Error Address				C8
Extended ROM Configuration Register 1				D0
Extended ROM Configuration Register 2				D4
Extended ROM Configuration Register 3				D8
Extended ROM Configuration Register 4				DC
Reserved	PLL Configuration	Reserved	Addr. Map B Options	E0
Memory Control Configuration Register 1				F0
Memory Control Configuration Register 2				F4
Memory Control Configuration Register 3				F8
Memory Control Configuration Register 4				FC

B.1 PCI INTERFACE CONFIGURATION REGISTERS

00_{16}	**Device ID (Read)**	**Vendor ID (Read)**
Default	0006_{16}	1057_{16}
Initialized	(read only)	(read only)

04_{16}	**PCI Status**	**PCI Command**
Default	$00A0_{16}$	0004_{16}
Initialized	(bit reset only)	0004_{16}

08_{16}	**Class Code**	**Subclass Code**	**Standard Programming**	**Revision ID**
Default	06_{16}	00_{16}	00_{16}	MPC8245 revision
Initialized	(read only)	(read only)	(read only)	(read only)

$0C_{16}$	**BIST control**	**Header type**	**Latency timer**	**Cache line size**
Default	00_{16}	00_{16}	00_{16}	00_{16}
Initialized	(read only)	(read only)	01_{16}	08_{16}

10$_{16}$	Local memory Base Address Register 0
Default	0000_0008$_{16}$
Initialized	0000_0008$_{16}$

14$_{16}$	Peripheral control and status register base address register
Default	0000_0000$_{16}$
Initialized	0000_0000$_{16}$

18$_{16}$	Local memory Base Address Register 1 (Read/Write)
Default	0000_0008$_{16}$
Initialized	0000_0008$_{16}$

2C$_{16}$	Subsystem ID	Subsystem Vendor ID
Default	0000$_{16}$	0000$_{16}$
Initialized	0000$_{16}$	0000$_{16}$

30$_{16}$	Expansion ROM base address
Default	0000_0000$_{16}$
Initialized	(read only)

3C$_{16}$	MAX LAT	MIN GNT	Interrupt Pin	Interrupt line
Default	00$_{16}$	00$_{16}$	01$_{16}$	00$_{16}$
Initialized	(read only)	(read only)	(read only)	01$_{16}$

40$_{16}$	Reserved	Reserved	Subordinate Bus Number	Bus Number
Default	-	-	00$_{16}$	00$_{16}$
Initialized	-	-	00$_{16}$	00$_{16}$

44$_{16}$	PCI arbiter control register	PCI general control register
Default	0000$_{16}$	0000$_{16}$
Initialized	C080$_{16}$	0020$_{16}$

B.2 PERIPHERAL POWER MANAGEMENT CONFIGURATION REGISTERS

70$_{16}$	Output Dr Cntl	PMCR2	PMCR1
Default	switch configured	switch configured	0000$_{16}$
Initialized	B7$_{16}$	44$_{16}$	C007$_{16}$

B.3 OUTPUT/CLOCK DRIVER AND MISC I/O CONTROL REGISTERS

74$_{16}$	Misc. Dr. Cntl 2	Misc. Dr. Cntl 1	CLK Driver Control register
Default	00$_{16}$	00$_{16}$	0000$_{16}$
Initialized	00$_{16}$	FC$_{16}$	0000$_{16}$

B.4 EMBEDDED UTILITIES MEMORY BLOCK ADDRESS REGISTER

78$_{16}$	Embedded utilities memory block base address register
Default	0000_0000$_{16}$
Initialized	8000_0000$_{16}$

B.5 MEMORY INTERFACE CONFIGURATION REGISTERS

80$_{16}$ Memory Starting Address

	Bank 3	Bank 2	Bank 1	Bank 0
Default	0000_0000$_{16}$			
Initialized	8080_8000$_{16}$			

84$_{16}$ Memory Starting Address

	Bank 7	Bank 6	Bank 5	Bank 4
Default	0000_0000$_{16}$			
Initialized	8080_8080$_{16}$			

88$_{16}$ Extended Memory Starting Address

	Reserved	Bank 3	Reserved	Bank 2	Reserved	Bank 1	Reserved	Bank 0
Default	0000_0000$_{16}$							
Initialized	8080_8000$_{16}$							

8C$_{16}$ Extended Memory Starting Address

	Reserved	Bank 7	Reserved	Bank 6	Reserved	Bank 5	Reserved	Bank 4
Default	0000_0000$_{16}$							
Initialized	8080_8080$_{16}$							

90$_{16}$ Memory Ending Address

	Bank 3	Bank 2	Bank 1	Bank 0
Default	0000_0000$_{16}$			
Initialized	8080_807C$_{16}$			

94$_{16}$ Memory Ending Address

	Bank 7	Bank 6	Bank 5	Bank 4
Default	0000_0000$_{16}$			
Initialized	8080_8080$_{16}$			

98$_{16}$ Extended Memory Ending Address

	Reserved	Bank 3	Reserved	Bank 2	Reserved	Bank 1	Reserved	Bank 0
Default	0000_0000$_{16}$							
Initialized	8080_8000$_{16}$							

9C$_{16}$ Extended Memory Ending Address

	Reserved	Bank 7	Reserved	Bank 6	Reserved	Bank 5	Reserved	Bank 4
Default	0000_0000$_{16}$							
Initialized	8080_8080$_{16}$							

A0$_{16}$	**Memory Page Mode**	**Reserved**	**Reserved**	**Memory Bank Enable**
Default	00$_{16}$	-	-	00$_{16}$
Initialized	00$_{16}$	-	-	01$_{16}$

B.6 PROCESSOR INTERFACE CONFIGURATION REGISTERS

A8$_{16}$ Processor interface configuration register 1

Default	00n4_0010$_{16}$
Initialized	0014_1310$_{16}$

AC$_{16}$ Processor interface configuration register 2

Default	000C_000C$_{16}$
Initialized	0000_0000$_{16}$

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

B.7 ERROR HANDLING REGISTERS

B8$_{16}$	Reserved	ECC Single-Bit Trigger	ECC Single-Bit Counter
Default	-	00$_{16}$	00$_{16}$
Initialized	-	00$_{16}$	00$_{16}$

C0$_{16}$	Proc. Bus Error Status	Reserved	Error Detection 1	Error Enabling 1
Default	00$_{16}$	-	00$_{16}$	01$_{16}$
Initialized	(bit reset)	-	(bit reset)	00$_{16}$

C4$_{16}$	PCI Bus Error Status	Reserved	Error Detection 2	Error Enabling 2
Default	00$_{16}$	00$_{16}$	00$_{16}$	00$_{16}$
Initialized	(bit reset)	-	(bit reset)	00$_{16}$

C8$_{16}$	Processor/PCI Error Address Register
Default	0000_0000$_{16}$
Initialized	(read only)

B.8 EXTENDED ROM CONFIGURATION REGISTERS

D0$_{16}$	Extended ROM Configuration Register 1
Default	B5FF_8000$_{16}$
Initialized	8C1F_8180$_{16}$

D4$_{16}$	Extended ROM Configuration Register 2
Default	B5FF_8000$_{16}$
Initialized	84FF_8000$_{16}$

D8$_{16}$	Extended ROM Configuration Register 3
Default	0C00_000E$_{16}$
Initialized	0000_0000$_{16}$

DC$_{16}$	Extended ROM Configuration Register 4
Default	0C00_000E$_{16}$
Initialized	0000_1000$_{16}$

B.9 ADDRESS MAP B OPTIONS AND PLL CONFIGURATION REGISTER

E0$_{16}$	PLL Config Register	Reserved	Reserved	Addr. Map B Options
Default	config setting	-	-	C0$_{16}$
Initialized	(read only)	-	-	40$_{16}$

B.10 MEMORY CONTROLCONFIGURATION REGISTER

F0$_{16}$	Memory Control Configuration Register 1
Default	FFn2_0000$_{16}$
Initialized	0B88_0002$_{16}$

F4$_{16}$	Memory Control Configuration Register 2
Default	0000_0000$_{16}$
Initialized	A660_0B3C$_{16}$

F8$_{16}$	Memory Control Configuration Register 3
Default	0000_0000$_{16}$
Initialized	0700_0000$_{16}$

FC$_{16}$	Memory Control Configuration Register 4
Default	0010_0000$_{16}$
Initialized	25B2_3220$_{16}$

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

C&H Technologies, Inc. <> 445 Round Rock West Drive <> Round Rock, TX 78681 <> www.chtech.com

READER'S COMMENT FORM

Your comments assist us in improving the usefulness of C&H's publications; they are an important part of the inputs used for revision.

C&H Technologies, Inc. may use and distribute any of the information that you supply in any way that it believes to be appropriate without incurring any obligation whatsoever. You may, of course, continue to use the information, which you supply.

Please refrain from using this form for technical questions or for requests for additional publications; this will only delay the response. Instead, please direct your technical questions to your authorized C&H representative.

COMMENTS: ____

____

____

____

____

____

____

____

Thank you for helping C&H to deliver the best possible product. Your support is appreciated.

Sincerely,

F. R. Harrison
President and CEO

INSTRUCTIONS

In its continuing effort to improve documentation, C&H Technologies, Inc. provides this form for use in submitting any comments or suggestions that the user may have. This form may be detached, folded along the lines indicated, taped along the loose edge (DO NOT STAPLE), and mailed. Please try to be as specific as possible and reference applicable sections of the manual or drawings if appropriate. Also, indicate if you would like an acknowledgment mailed to you stating whether or not your comments were being incorporated.

> NOTE: This form may not be used to request copies of documents or to request waivers, deviations, or clarification of specification requirements on current contracts. Comments submitted on this form do not constitute or imply authorization to waive any portion of the referenced document(s) or to amend contractual requirements.

———————————— (Fold along this line) ————————————

———————————— (Fold along this line) ————————————

____

____

____

Place
Stamp
Here

C&H Technologies, Inc.
Technical Publications
445 Round Rock West Drive
Round Rock, Texas 78681-5012