Systran BHAS-CPCISC4

# CompactPCI Slave 4-Slot IP Module Carrier Board


Your **definitive** source for quality pre-owned equipment.

**Limited Availability**
**Used and in Excellent Condition**

**Open Web Page**

https://www.artisantg.com/97641-1

**Artisan Technology Group**
(217) 352-9330 | sales@artisantg.com | artisantg.com



- Critical and expedited services
- In stock / Ready-to-ship
- We buy your excess, underutilized, and idle equipment
- Full-service, independent repair center

# CPCISC4

## CompactPCI® Slave 4-Slot IP Module Carrier Hardware Reference

Document No. B-T-MR-CPCISC4#-A-0-A5

# FOREWORD

The information in this document has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Systran reserves the right to make changes without notice.

Systran makes no warranty of any kind with regard to this printed material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.



Systran® is a registered trademark of Systran Corporation.
ALTERA® and Altera® are registered trademarks of Altera Corporation.
Littelfuse® registered trademark of Littelfuse, Inc.
VIEW*logic*® is a registered trademark of Mentor Graphics Corporation (formerly Innoveda, Inc.).
PICMG® and CompactPCI® are registered trademarks of the PCI Industrial Computer Manufacturer's Group

Systran is an Associate Level member of PICMG and as such may use the PICMG and CompactPCI logos.



Any reference made within this document to equipment from other vendors does not constitute an endorsement of their product(s).

Revised: April 15, 2004

**Systran Corporation**
4126 Linden Avenue
Dayton, OH 45342-3068 USA
(800) 252-5601 (U.S. only)
(937) 252-5601

**FCC**

This product is intended for use in industrial, laboratory or military environments. This product uses and emits electromagnetic radiation, which may interfere with other radio and communication devices. The user may be in violation of FCC regulations if this device is used in other than the intended market environments.

**CE**

As a component part of another system, this product has no intrinsic function and is therefore not subject to the European Union CE EMC directive 89/336/EEC.

# TABLE OF CONTENTS

# APPENDICES



# FIGURES



# TABLES

# 1. INTRODUCTION

## 1.1 How To Use This Manual

### 1.1.1 Purpose

This is a reference manual for the Systran CPCISC4 CompactPCI Slave 4-slot IP Module Carrier herein referred to as the CPCISC4 board, part number BHAS-CPCISC4.

**NOTE**: In this manual CompactPCI will hereafter be referred to as CPCI. Also note that the acronym "CPCI" will be used to refer to any characteristics of the CPCISC4 that are mechanically based; this includes the J1 connector for interfacing to the host via a backplane. The acronym "PCI" will be used when relating to electrical characteristics of the board or relating to signals that are covered in the PCI specification separate from CompactPCI.

### 1.1.2 Scope

This reference manual covers the physical and operational description of the CPCISC4, both from hardware and software perspectives. This reference manual also contains detailed technical information about CPCISC4 performance characteristics, and a few typical applications. A general understanding of computer processing, software and/or hardware applications experience; and a working knowledge of using IP Modules, is necessary to effectively use this manual.

### 1.1.3 Style Conventions

- Called functions are italicized. For example, *OpenConnect()*.
- Data types are italicized. For example, *int*.
- Function parameters are bolded. For example, **Action**.
- Path names are italicized. For example, *utility/sw/cfg*.
- File names are bolded. For example, **config.c**.
- Path file names are italicized and bolded. For example, ***utility/sw/cfg/config.c***.
- Hexadecimal values are written with a "0x" prefix. For example, 0x7E.
- An 'Active Low' signal on a hardware product has a slash (/) prefix. For example, /SYNC.
- Code and monitor screen displays of input and output are boxed and indented on a separate line. Text that represents user input is bolded. Text that the computer displays on the screen is not bolded. For example:

```
c:\>ls
file1          file2          file3
```

- Large samples of code are Courier font, at least one size less than context, and are usually on a separate page or in an appendix.

## 1.2 Related Information

- Systran's *PCI/CPCI IP Module Carrier Software Installation and DLL Reference for Windows NT 4.0* (Doc. No. B-T-ML-NTASP)

- PCI Local Bus Specification Rev. 2.1- PCI Special Interest Group
- *American National Standard for IP Modules (ANSI/VITA4 – 1995)* published by the VMEbus International Trade Association, 7825 Gelding Dr., Suite 104, Scottsdale, AZ 85260, Telephone: (602) 951-8866
- QuickLogic (previously V3 Semiconductor) www.quicklogic.com
- CompactPCI Specification, PICMG 2.0 R2.1, Sept. 1997 by the PCI Industrial Computer Manufacturer's Group

## 1.3 Quality Assurance

Systran Corporate policy is to provide our customers with the highest quality products and services. In addition to the physical product, the company provides documentation, sales and marketing support, hardware and software technical support, and timely product delivery. Our quality commitment begins with product concept, and continues after receipt of the purchased product.

Systran's Quality System conforms to the ISO 9001 international standard for quality systems. ISO 9001 is the model for quality assurance in design, development, production, installation and servicing. The ISO 9001 standard addresses all 20 clauses of the ISO quality system, and is the most comprehensive of the conformance standards.

Our Quality System addresses the following basic objectives:

- Achieve, maintain and continually improve the quality of our products through established design, test, and production procedures.
- Improve the quality of our operations to meet the needs of our customers, suppliers, and other stakeholders.
- Provide our employees with the tools and overall work environment to fulfill, maintain, and improve product and service quality.
- Ensure our customer and other stakeholders that only the highest quality product or service will be delivered.

The British Standards Institution (BSI), the world's largest and most respected standardization authority, assessed Systran's Quality System. BSI's Quality Assurance division certified we meet or exceed all applicable international standards, and issued Certificate of Registration, number FM 31468, on May 16, 1995. The scope of Systran's registration is: "Design, manufacture and service of high technology hardware and software computer communications products." The registration is maintained under BSI QA's bi-annual quality audit program.

Customer feedback is integral to our quality and reliability program. We encourage customers to contact us with questions, suggestions, or comments regarding any of our products or services. We guarantee professional and quick responses to your questions, comments, or problems.

## 1.4 Technical Support

Technical documentation is provided with all of our products. This documentation describes the technology, its performance characteristics, and includes some typical applications. It also includes comprehensive support information, designed to answer any technical questions that might arise concerning the use of this product. We also publish and distribute technical briefs and application notes that cover a wide assortment of topics. Although we try to tailor the applications to real scenarios, not all possible circumstances are covered.

Although we have attempted to make this document comprehensive, you may have specific problems or issues this document does not satisfactorily cover. Our goal is to offer a combination of products and services that provide complete, easy-to-use solutions for your application.

If you have any technical or non-technical questions or comments, contact us. Hours of operation are from 8:00 a.m. to 5:00 p.m. Eastern Standard/Daylight Time.

- Phone: (937) 252-5601 or (800) 252-5601
- E-mail: support@systran.com
- Fax: (937) 252-1349
- World Wide Web address: www.systran.com

## 1.5 Ordering Process

To learn more about Systran products or to place an order, please use the following contact information. Hours of operation are from 8:00 a.m. to 5:00 p.m. Eastern Standard/Daylight Time.

- Phone: (937) 252-5601 or (800) 252-5601
- E-mail: info@systran.com
- World Wide Web address: www.systran.com

*This page intentionally left blank*

# 2. PRODUCT OVERVIEW

## 2.1 Overview

The Systran CPCISC4 Slave 4-slot IP Module Carrier board (part number BHAS-CPCISC4) is a CompactPCI slave add-in board that supports four singlewide IP Modules, two doublewide IP Modules or one doublewide with two singlewide IP Modules. The CPCISC4 supports 8 MHz IP transfers of ID, I/O, Interrupt Acknowledge and Memory for all slots. The CPCISC4's PCI interface design is compliant with the PCI bus specification V2.1 and the IP interface is designed in accordance with the *American National Standard for IP Modules (ANSI/VITA 4-1995)* specification. The bus connector interface and board physical layout are in accordance with the CompactPCI specification PICMG 2.0 R2.1.

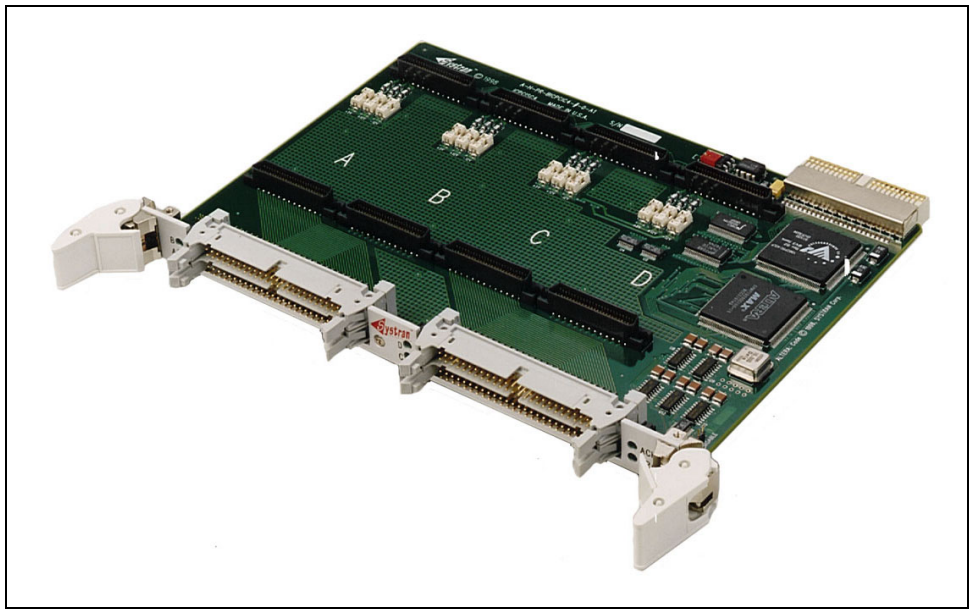


**Figure 2-1 CPCISC4 Carrier**

The CPCISC4 encompasses a design that does not insert IP hold cycles. This guarantees the highest throughput possible for this PCI-to-IP Module bus-coupling carrier. The CPCISC4 provides up to 200 I/O points (50 per IP) per CPCI slot with the area under two of the IP Modules as a solid ground plane to provide "quiet" operating conditions for sensitive analog IP Modules. LEDs on the CPCISC4 provide diagnostic information on accesses to the IPs, the CPCISC4 local registers, and an IP acknowledgment signal. The CPCISC4 has a very easy-to-use and flexible interrupt capability where the individual interrupts from each IP Module can be programmed to assert a common interrupt request to the PCI bus.

### 2.1.1 Features

- Up to 200 I/O points in a single, CPCI slot.
- I/O, ID, INT and MEM transfers on all slots with no HOLD cycles inserted.

- Precise PCI local memory map for easy access to IP Modules and local registers.
- Board base addresses automatically configured by system BIOS at power-up.
- Supports four singlewide IPs, or any combination of doublewide and singlewide cards adding up to four slots.
- Each of eight IP interrupt requests (two per IP Module slot) can individually assert a common PCI interrupt.
- Local bus interface time-out can be enabled via software.
- Board is memory mapped via two apertures in the PCI bridge.
- Independent field-replaceable fuses, and 'T'-type filters are used for the +5, +12, and -12 V power feeding each IP.
- Six IP/Carrier Access and IP Acknowledge indicators (with pulse stretchers).
- Pure ground plane under two of four IP Modules (slots A and B) for "quiet" operation.
- Posts IP error signals as status.
- Supports writes to ID space.
- Resets individual IP Modules via software.
- Four "standard" I/O ribbon cable connections can be made without board removal.
- Supports four "strobe" connections (one per slot).

# 2.2 Functional Description

## 2.2.1 Memory Mapping Scheme

The mapping scheme for the ID and I/O address spaces on all four IP Modules as well as the carrier's on-board CSRs are all made at fixed addresses relative to the board base address, which is contained in the PCI Configuration Space Registers. These registers are found inside the V962PBC PCI Bridge IC (U6 on the CPCISC4). For more information on these registers, contact QuickLogic (previously V3 Semiconductor) as shown in Section 1.2 Related Information.

IP Module memory accesses are performed relative to base addresses set within an aperture register in the Configuration Space and on-board address decoding. Memory Size as well as base address is controlled by the aperture settings. Refer to Chapter 3, INSTALLATION and Appendix B, PROGRAMMING GUIDE, for more details on memory mapping.

## 2.2.2 PciCfg Utility

The **PciCfg** Utility facilitates the modification of the PCI configuration EEPROM on up to 16 installed IP Carriers on a single machine. This software allows for quick, easy disabling or enabling of the IP module memory space. If the IP memory space is disabled, the carrier occupies much less PCI memory space. Instructions for installation and use of the **PciCfg** program disk are included in Appendix E of this manual. The **PciCfg** Utility software is provided with this manual in a sleeve on the inside back cover.

## 2.2.3 ID Transfers

IP Module ID transfers are supported as 8-bit or 16-bit local transfers. ID transfers are performed by a byte or word read/write operation of the slot IP Module ID address space from the PCI Bus. The ID ROM values are located every other byte at IP Module odd-byte locations (IP Module byte-lane 0). For 8-bit reads the even byte is discarded. The ID

ROM area allows read or write accesses to support future IP Modules requiring writes to the ID address space.

The IP Module data bus has been made big endian to be compatible with VME. Since PCI is 32-bit little endian, accesses to the IP Modules occur for PCI even addresses on four byte boundaries only. On PCI, the odd byte is the upper byte (D15 - D8) and the even byte is the lower byte (D0 - D7). For this reason, on 16-bit reads and writes, the IP Module odd byte is transferred on the PCI even byte, and the IP Module even byte is transferred on the PCI odd byte. Single-byte transfers are performed on PCI even addresses only and access the IP Module odd byte.

### 2.2.4 I/O Transfers

IP Module I/O transfers are supported as 8-bit or 16-bit PCI transfers. I/O transfers are performed by a byte or word read or write operation of the slot IP Module I/O address space from the PCI Bus.

### 2.2.5 Interrupt Transfers

Interrupts are requested by the IP Module by assertion of N_INTREQ0 or N_INTREQ1. An independent Local-to-PCI interrupt circuit handles each of these IP Module interrupt request lines which asserts the interrupt request line INTA# to the PCI bus. This provides a minimum delay in asserting the interrupt request to the CPU board.

In detail, the hardware and firmware in the CPCISC4 are configured such that any of nine interrupts (Intreq0 and Intreq1 from each of four IP Modules and a time-out interrupt) will generate an INTB# input to the on-board PCI bridge which will generate an INTA# to the PCI bus. The INTB# from the IP Interface controller will only be generated if enabled to do so in software. Interrupts are disabled upon power-up to prevent a possible interrupt cycle from occurring until the system becomes stable. Upon completion of power-up, bits in the Control Register on the CPCISC4 can then be set to enable IP and time-out interrupts.

### 2.2.6 Interrupt Configuration

The interrupt-select tiers in Figure 2-1 show the interrupt “flow” from the IP Modules to the PCI bus.

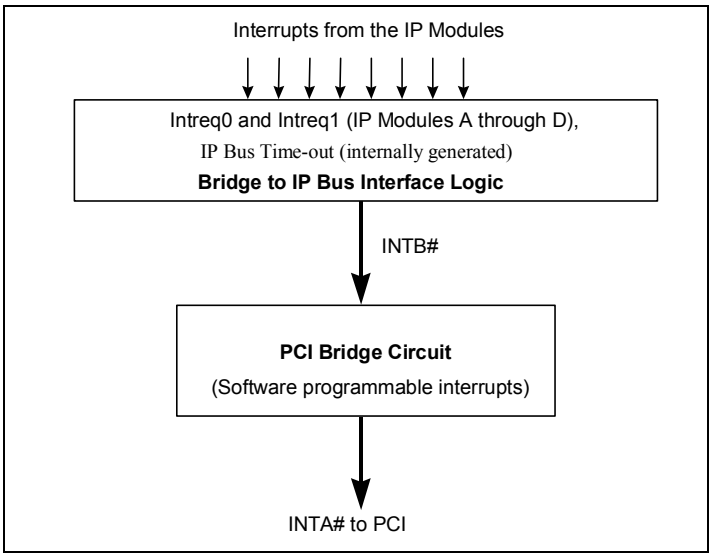


**Figure 2-1 Interrupt-Select Tiers**

### 2.2.7 Diagnostic LEDs

The CPCISC4 has three "double" LEDs used for diagnostic purposes. The first unit, DS1, indicates a transaction to IP Module "A" on the left LED (marked "A") and IP Module "B" on the right LED (marked "B"). The second unit, DS2, indicates a transaction attempt to IP Module "C" on the left LED (marked "C") and IP Module "D" on the right LED (marked "D"). The third unit, DS3, indicates a transaction attempt to the carrier registers on the left LED (marked "CRA") and successful acknowledgment of any transaction on the right LED (marked "ACK"). This scheme provides independent verification of the access and the acknowledgment.

**Table 2-1 LED Descriptions**

| LED | Marking | Description |
|---|---|---|
| DS1 | A | Transactions to IP Module "A" |
| | B | Transactions to IP Module "B" |
| DS2 | C | Transactions to IP Module "C" |
| | D | Transactions to IP Module "D" |
| DS3 | CRA | Transactions to carrier registers |
| | ACK | Successful acknowledgement of any transaction |

### 2.2.8 IP Module Reset

Independent or simultaneous resetting of the IP Modules is facilitated through writes to the Reset register. Monitoring of the pulse-stretched reset signals is made possible by reading the Reset register. See Appendix B, PROGRAMMING GUIDE, for a detailed description of this register.

### 2.2.9 Error Status

Monitoring of the IP Module ERROR signals is made possible by reading the Error Status register. See Appendix B, PROGRAMMING GUIDE, for a detailed description of this register.

### 2.2.10 Strobe Signals

The IP Module STROBE signals can be connected via an 8-pin header (J4).

### 2.2.11 Power Supply Filtering

Independent 'T'-type filters are used for the +5 V, +12 V, and -12 V power feeding each IP Module. This arrangement provides superior power supply filtering.

### 2.2.12 Fuses

Independent field-replaceable fuses are used for the +5V, +12 V, and -12 V power feeding each IP Module. This provides superior short-circuit protection.

**NOTE**: Replacement fuses can be purchased directly from Littelfuse, Inc. Phone (708) 829-0400.

1 Amp Fuse Manufacturer's Part Number R451001
2 Amp Fuse Manufacturer's Part Number R451002

# 2.3 Theory of Operation

## 2.3.1 Block Diagram Description

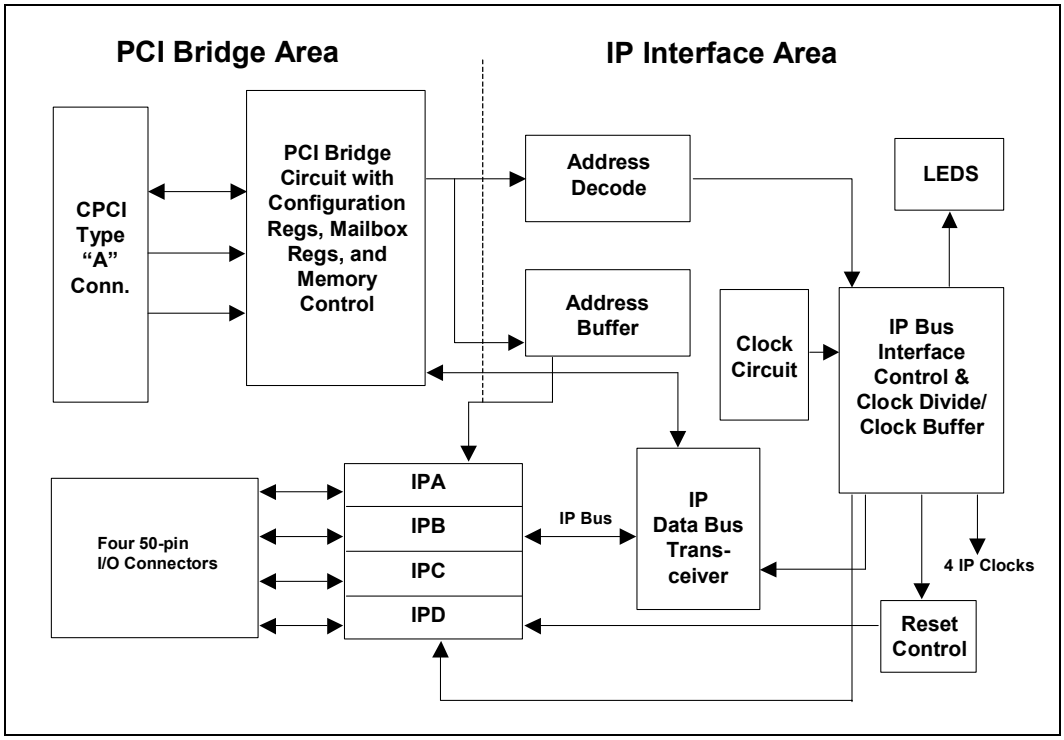


**Figure 2-2 CPCISC4 Block Diagram**

Figure 2-2 is a simplified block diagram of the CPCISC4 representing the signal flow between the CPCI bus backplane, IP Module Logic Bus and the IP Module I/O connectors. On the left side of the diagram is the PCI bus connector through which all transfers between the backplane and the Bridge registers, CSR registers or IP Modules flow. The PCI-to-IP transfers **do not** insert any IP Module hold cycles into the accesses to the IP Modules.

The block labeled "ADDRESS DECODE" handles decoding of the aperture address generated from the bridge circuit. This circuit determines which select strobe to generate to a particular IP Module based on the address bits LA26 through LA2.

The block labeled "ADDRESS BUFFER" provides the common IP address bits IPA1 through IPA6 to all the IP Modules as well as address bits 7 through 22 during the select cycle for memory transactions.

The block labeled "IP DATA BUS TRANSCEIVER" allows for buffering between the PCI bridge device and the IP Module data bus where the IP Interface Control controls the direction on the bus, and enabling to the bus.

The block labeled "IP INTERFACE CONTROL" provides timing and control of the data flow between the PCI bus bridge and the IP Module bus. It also provides control of the address buffers such that the IP Module data bus will contain an address only during the select cycle of a memory transfer. This block also has a selectable time-out circuit that prevents the local side of the PCI bridge circuit from "hanging". This block will also produce the reset pulses for the IP Modules and the required synchronous select cycles for ID, I/O, Interrupt Acknowledge and Memory transactions.

The RESET CONTROLLER (Reset Cont.) block consists of a bank of pulse stretchers that convert the narrow trigger from the reset register in the IP INTERFACE CONTROL part to a 200 ms pulse for each of the IP Modules. The wide pulses will then be synchronized to the IP bus clock such that they will be deasserted synchronously.

The “LEDS” block contains pulse stretchers and SIX LEDs to indicate transaction attempts to each of four IP Modules and the on board CSR registers as well as an acknowledge indicator.

The block labeled “CLOCK CIRCUIT” consists of a 16 MHz oscillator, divider flip-flop and low-skew buffers to allow a separate clock for each IP Module.

The block labeled “PCI BRIDGE CIRCUIT” consists of a large-scale IC that provides an interface between the PCI bus backplane and the interface to the IP Module bus.

# 3. INSTALLATION

## 3.1 Unpack the CPCISC4

Table 3-1 lists the contents of the CPCISC4 shipping packages.

**Table 3-1 Contents of CPCISC4 Shipping Packages**

| Qty | Description |
|---|---|
| 1 | CPCISC4 Printed Circuit Assembly |
| 2 | Surface mount spare fuses (one-1 Amp, one-2 Amp) |
| 1 | CPCISC4 Hardware Reference* |

The Printed Circuit Assembly is enclosed in an anti-static box. The box and the manual are packaged together in a larger box. Save the shipping material in case the board needs to be returned.

* One manual is shipped for each board ordered. Extra manuals may be purchased by calling Systran or by mail. Use the prefix "BTMR-" followed by the product order part number (for example, BTMR- CPCISC4).

## 3.2 Visually Inspect the CPCISC4

Examine the CPCISC4 to determine if any damage occurred during shipping.

## 3.3 Install the CPCISC4

**NOTE**: The CPCISC4 is an Electrostatic Sensitive Device (ESD). The hardware installation of the CPCISC4 must be conducted on an anti-static workbench to protect the IP Modules and carrier board.

The tools required for the CPCISC4 installation are listed in Table 3-2.

**Table 3-2 CPCISC4 Installation Tools**

| Qty | Description |
|---|---|
| 1 | ESD Static Control Kit/Ground Strap/Etc. |
| 1 | Standard Flat Head Screwdriver |
| 1 | Standard Phillips Screwdriver (Optional) |

## 3.3.1 Jumper and Strobe Header Locations

Figure 3-1 shows the locations of the Reset Mode jumper (J2), the V3 EEPROM Write Protect jumper (J3), the Time-out Jumper (J5) and the Strobe header (J4).

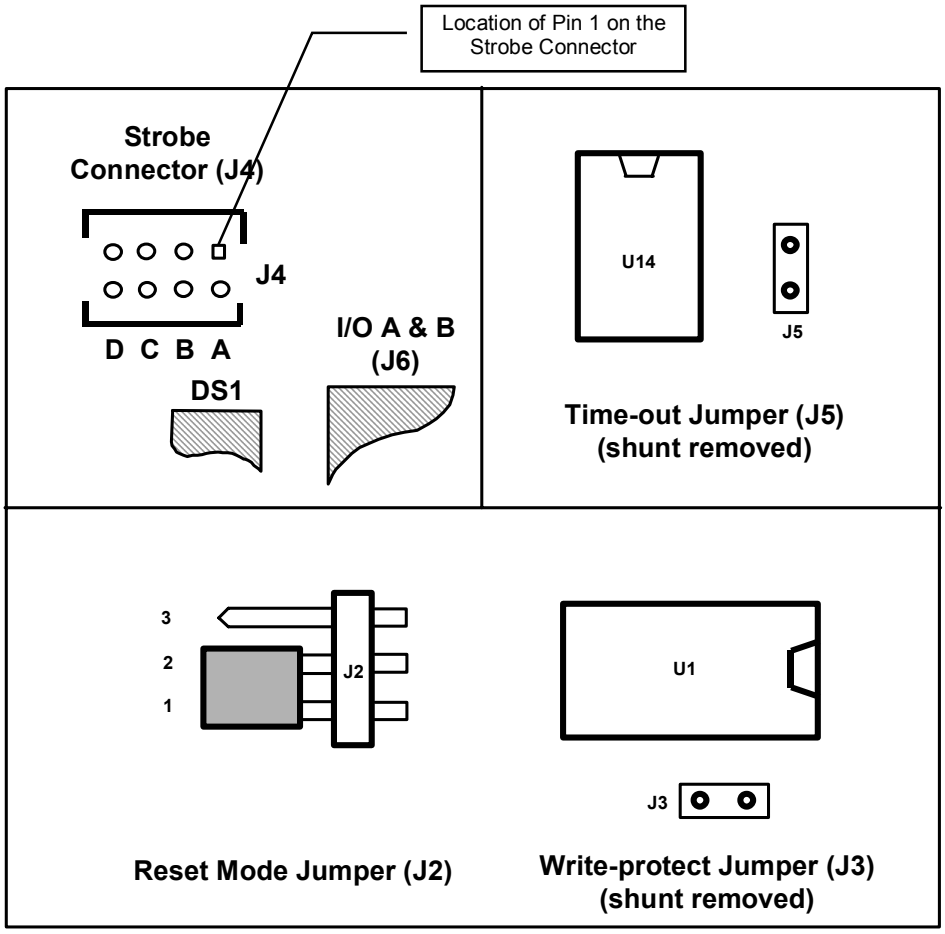


**Figure 3-1 Jumper and Strobe Header Locations**

### RESET MODE JUMPER

Reset Mode jumper (J2) is a special jumper that is used if the host system exhibits "glitches" on the PCI Reset line that causes the board to not configure itself properly during initialization. Figure 3-1 shows the factory default (normal mode) jumper location for J2 (shunt pins 1 and 2). This is for systems that do not exhibit this problem. If your system is exhibiting the described problem, then the jumper shunt may be moved to pins 2 and 3. Contact Systran Customer Support for more information.

### V3 EEPROM WRITE PROTECT JUMPER

The V3 EEPROM Write Protect jumper (J3) enables/disables the write protect for the V3 PCI Bridge chip (U6) configuration EEPROM (U1). This EEPROM contains the information necessary to initialize the V3 chip in the PCI system. The factory setting for J3 is write-protect enabled (no shunt installed). Normally, it is not necessary to modify the configuration EEPROM. However, the EEPROM may be altered by installing a shunt to J3 that permits write-access to the EEPROM. Note that this jumper is used in the procedure called out in Appendix E, PCICFG UTILITY FOR PCI-BASED CARRIERS.

## TIME-OUT JUMPER

Although the PCI bus allows a maximum number of clock cycles before a retry, it is still possible for an incorrectly addressed IP Module to "hang" the local side of the bridge IC by not returning an acknowledge signal. Consequently, there is a potential for a wrongly addressed, faulty or missing IP Module to cause an endless string of PCI retries unless the carrier produces its own local time-out to release the bus.

In other cases, it may not be desirable for the carrier to abort a transaction. For example, when used for diagnostic purposes or for use as an IP Module with a very high number of specified wait states. For this reason, the CPCISC4 allows a jumper-selectable enable/disable of the time-out feature. The enable/disable selection determines the time-out operation for **all** IP Module positions on the CPCISC4—either the time-out feature is enabled for all positions or it is disabled for all positions.

To enable the time-out feature, remove the shunt from header J5 (factory default) and set the TO_EN bit in the Carrier Control Register to a '1.' The CPCISC4 will now abort any transaction to an IP Module after a maximum of 29 wait states (slightly less than four microseconds) if the selected IP Module fails to assert an acknowledge. After this time has elapsed, any data returned to the PCI bus is invalid. To disable the time-out feature, place the shunt across both pins on J5 (or leave the shunt for header J5 off and set the TO_EN bit in the Carrier Control Register to a '0'). The CPCISC4 will now allow an indefinite number of IP Module wait states.

**NOTE**: The CPCISC4 is shipped with the time-out feature **disabled**. Verify that an IP Module selected for use on the CPCISC4 produces the correct acknowledge signal before placing it on the carrier. In addition, verify that the software address used for accessing an IP Module on the CPCISC4 agrees with the physical address of the IP Module in the system. Attempting to access an IP Module that is either not physically present or not correctly addressed on the CPCISC4, or is faulty, could cause an endless string of PCI retries.

## CPCISC4 IP MODULE STROBE CONNECTOR

IP Module Logic Bus connector has a signal called "$\overline{\text{STROBE\_X}}$" and is user-defined by the IP Module Logic Bus Specification. On the CPCISC4, each of the strobe signals is provided through J4 with the IP Module slot "A" strobe signal on pin 1, "B" strobe on pin 2, "C" strobe on pin 3 and "D" strobe on pin 4.

**Table 3-3 CPCISC4 STROBE Connector Pin Assignments**

| Pin Number | Signal Name |
|---|---|
| 1 | $\overline{\text{STROBE\_A}}$ |
| 2 | $\overline{\text{STROBE\_B}}$ |
| 3 | $\overline{\text{STROBE\_C}}$ |
| 4 | $\overline{\text{STROBE\_D}}$ |
| 5 | GROUND |
| 6 | GROUND |
| 7 | GROUND |
| 8 | GROUND |

Pins 5 through 8 are connected to ground, as indicated in Table 3-3. Each of the strobe signals on the CPCISC4 is pulled-up to +5 V via 10 KΩ resistors. The strobe connector (J4) is located as shown in Figure 3-1.

**NOTE**: The silk-screen numbering for the J4 header is such that pin one is in the upper-right corner when facing the component side of the CPCISC4 board and the screened lettering is right side up. Pin 5 is just below pin 1 in the numbering sequence.

## 3.3.2 Installation of I/O Cables

Installation of I/O cables to the CPCISC4 is similar to procedures for VME-based I/O carriers including Systran's VMESC4M. Based on the 6U VME form factor, the CPCISC4 CPCI carrier has two 100-pin "double-stack" headers. Refer to Figure 3-2 for the following discussion.

Note that the CPCISC4 has two 100-pin "double-stacked" headers, one for connection to IP Modules A and B while the other is for connection to IP Modules C and D. The positions of the four headers is clearly marked on the CPCISC4 front panel and are as shown below in Figure 3-2. This allows for insertion of the I/O cables into the headers with the board either removed from the CPCI chassis or secured into it. After the IP Modules have been installed as detailed in the next section and the carrier has been secured into the chassis, should the cables require removal, simply pull the securing tabs outward on the header to pull the connector out. Reinstall the connector by pushing into the header and securing the tabs behind it.

Tagging each cable with a unique code using adhesive labels or permanent ink marker is recommended.

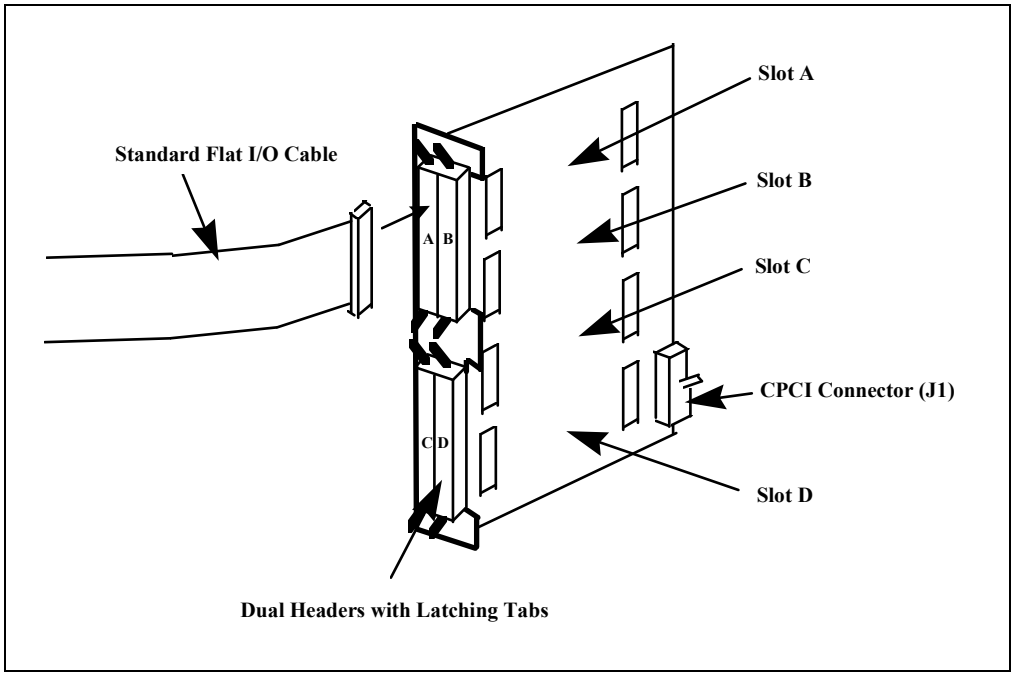


**Figure 3-2 I/O Cable Installation**

## 3.3.3 Installation of IP Modules on the CPCISC4

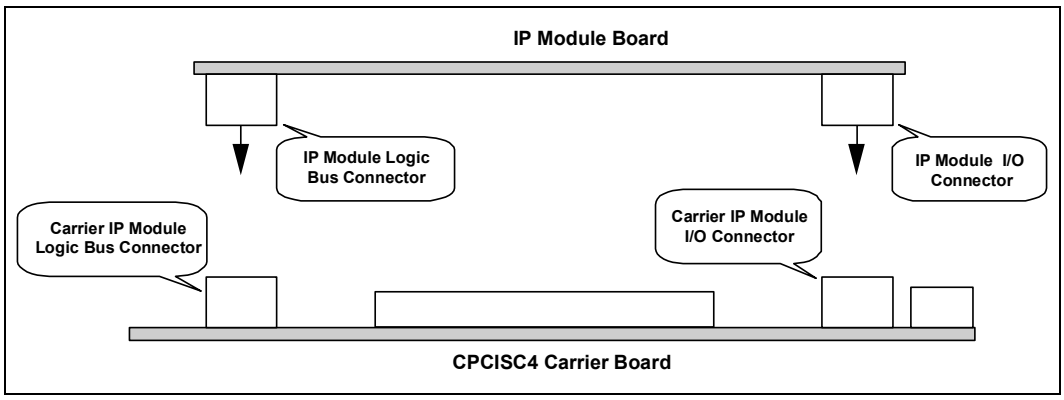


**Figure 3-3 Installation of an IP Module on the CPCISC4**

Referring to the appropriate figures and table described below, perform the following steps in removing the CPCISC4 from its shipping container (or from an existing installation) and installing IP Modules. The asterisk (*) denotes optional items.

1. Turn off all power to the CPCI backplane.
2. Remove the CPCISC4 from its shipping container and move it to the ESD-controlled area where the installation of the IP Module(s) can be made.

*2a. Remove the CPCISC4 from the CPCI backplane if modifications are necessary, and move it to an ESD controlled area where the installation or changing of IP Modules can be made.

3. Set the Time-out jumper (J5), the EEPROM Write Protect jumper (J3), and the Reset Mode jumper (J2) to the desired positions.
4. Install the IP Module(s) onto the carrier board by applying adequate and equal pressure to the IP Module(s) at both ends and on the CPCISC4 board.

*5. Install four M2x5mm flat head machine screws onto the IP Module's connectors.

*6. Install any required I/O cables onto the CPCISC4.

7. Slide the CPCISC4 into a desired peripheral slot of a 6U CPCI backplane while applying enough force to ensure a tight fit into the backplane connector. Note that the J1 connector on the CPCISC4 has a blue colored plastic key guide in the connector's center. This denotes the board as 5 V compatible. If the CPCISC4 does not fit into the chassis, check to make sure the chassis has a 5 V backplane. After insertion, use a standard flathead screwdriver to secure the CPCISC4 panel onto the frame of the CPCI chassis.

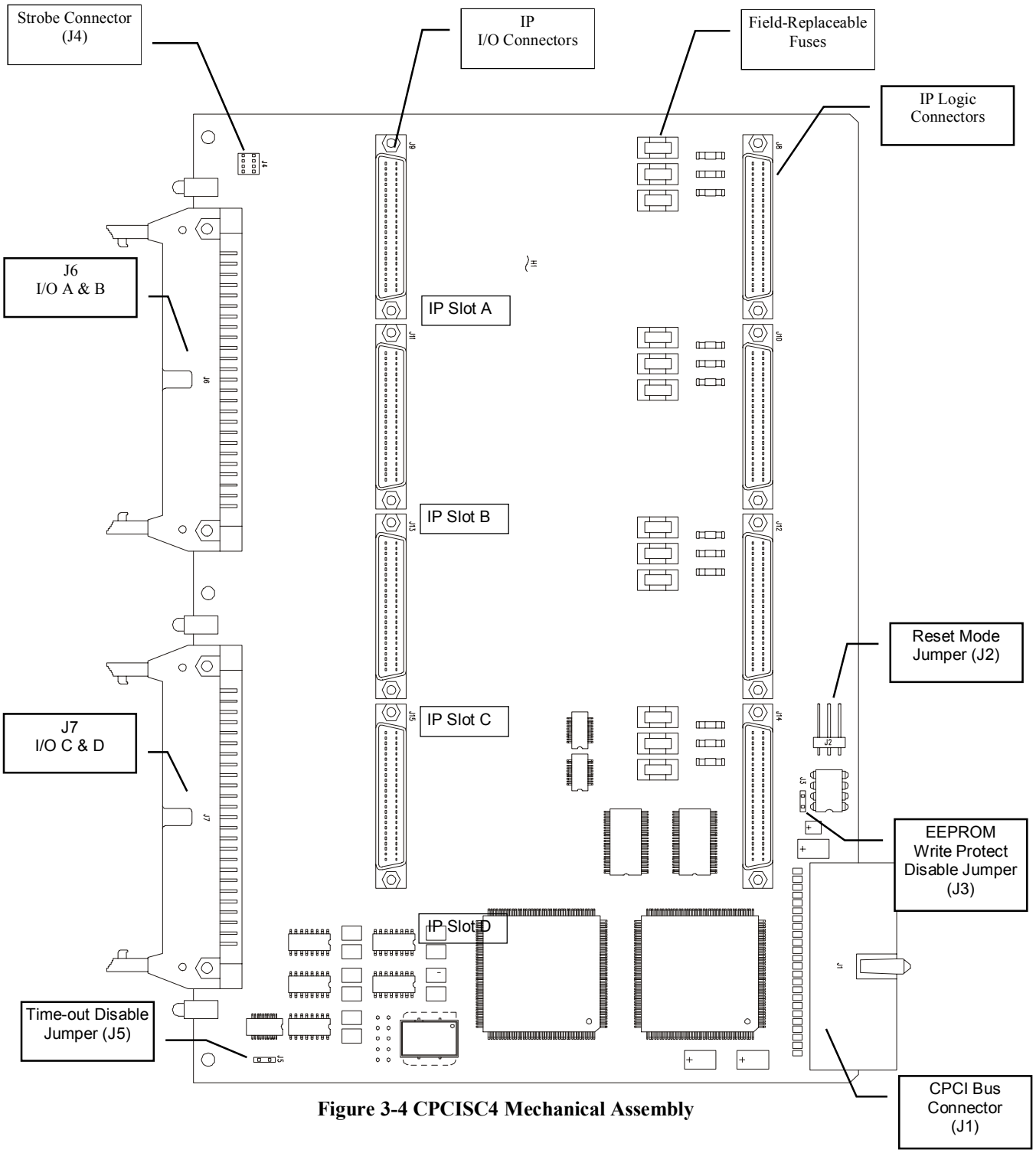


**Figure 3-4 CPCISC4 Mechanical Assembly**

## 3.3.4 Connector Pin Assignments

Table 3-4 contains the pin assignments for the PCI signals, power, ground and other pertinent connections made to the J1 connector, which interfaces to the CPCI backplane. Note that the J1 connector has six columns of pin positions labeled A through F with column F providing connections for a shield ground. There are twenty-five rows of pins with rows 12 through 14 not used since this area of the connector is designated for the voltage key. Also note that Table 3-4 shows both the letter-number designation for each pin as well as its number-only equivalent shown on schematic #2 in the appendix.

**Table 3-4 PCI Bus Pin Assignments**

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 25 | 5 V | (25) | REQ64# | (50) | ENUM# | (75) | 3.3 V | (100) | 5 V | (125) | GND | (150) | | |
| 24 | AD[1] | (24) | 5 V | (49) | V(I/O) | (74) | AD[0] | (99) | ACK64# | (124) | GND | N/C | | |
| 23 | 3.3 V | (23) | AD[4] | (48) | AD[3] | (73) | 5 V | (98) | AD[2] | (123) | GND | (148) | | |
| 22 | AD[7] | (22) | GND | (47) | 3.3 V | (72) | AD[6] | (97) | AD[5] | (122) | GND | N/C | | |
| 21 | 3.3 V | (21) | AD[9] | (46) | AD[8] | (71) | M66EN | (96) | C/BE[0]# | (121) | GND | (146) | | |
| 20 | AD[12] | (20) | GND | (45) | V(I/O) | (70) | AD[11] | (95) | AD[10] | (120) | GND | N/C | | |
| 19 | 3.3 V | (19) | AD[15] | (44) | AD[14] | (69) | GND | (94) | AD[13] | (119) | GND | (144) | | J |
| 18 | SERR# | (18) | GND | (43) | 3.3 V | (68) | PAR | (93) | C/BE[1]# | (118) | GND | N/C | | 1 |
| 17 | 3.3 V | (17) | SDONE | (42) | SBO# | (67) | GND | (92) | PERR# | (117) | GND | (142) | | |
| 16 | DEVSEL# | (16) | GND | (41) | V(I/O) | (66) | STOP# | (91) | LOCK# | (116) | GND | N/C | | |
| 15 | 3.3 V | (15) | FRAME# | (40) | IRDY# | (65) | GND | (90) | TRDY# | (115) | GND | (140) | | C |
| 12-14 | **KEY AREA** | | | | | | | | | | | | | O |
| 11 | AD[18] | (11) | AD[17] | (36) | AD[16] | (61) | GND | (86) | C/BE[2]# | (111) | GND | (136) | | N |
| 10 | AD[21] | (10) | GND | (35) | 3.3 V | (60) | AD[20] | (85) | AD[19] | (110) | GND | N/C | | N |
| 9 | C/BC[3]# | (9) | IDSEL | (34) | AD[23] | (59) | GND | (84) | AD[22] | (109) | GND | (134) | | E |
| 8 | AD[26] | (8) | GND | (33) | V(I/O) | (58) | AD[25] | (83) | AD[24] | (108) | GND | N/C | | C |
| 7 | AD[30] | (7) | AD[29] | (32) | AD[28] | (57) | GND | (82) | AD[27] | (107) | GND | (132) | | T |
| 6 | REQ# | (6) | GND | (31) | 3.3 V | (56) | CLK | (81) | AD[31] | (106) | GND | N/C | | O |
| 5 | BRSVP1A5 | (5) | BRSVP1B5 | (30) | RST# | (55) | GND | (80) | GNT# | (105) | GND | (130) | | R |
| 4 | BRSVP1A4 | (4) | GND | (29) | V(I/O) | (54) | INTP | (79) | INTS | (104) | GND | N/C | | |
| 3 | INTA# | (3) | INTB# | (28) | INTC# | (53) | 5 V | (78) | INTD# | (103) | GND | (128) | | |
| 2 | TCK | (2) | 5 V | (27) | TMS | (52) | TDO | (77) | TDI | (102) | GND | N/C | | |
| 1 | 5 V | (1) | -12 V | (26) | TRST# | (51) | +12 V | (76) | 5 V | (101) | GND | (126) | | |
| Pin | **A** | | **B** | | **C** | | **D** | | **E** | | **F** | | | |

Table 3-5 shows the pin assignments for each IP Module Logic Bus connector. The signals on the left side of the table are of the original IP Module signal nomenclature, and the signals on the right are those used by Systran Corp. The upper case "X" in the Systran signal name represents the "A", "B", "C" or "D" mnemonic for each IP Module slot for memory transfers, I/O, ID, or interrupt transfers.

**Table 3-5 IP Module Logic Bus Pin Assignments**

| Original IP Module Signals Names | IP Module Logic Bus Pin # | Systran Signal Names | Bussed or Unique | X = Pulled Up Via 10 KΩ Resistor |
|---|---|---|---|---|
| GND | 50 | GND | GND | GND |
| Reserved | 49 | RESERVED1 | U | X |
| Ack* | 48 | $\overline{\text{ACK\_X}}$ | U | X |
| A6 | 47 | IPA6 | B | |
| Strobe* | 46 | $\overline{\text{STROBE\_X}}$ | U | X |
| A5 | 45 | IPA5 | B | |
| Intreq1* | 44 | $\overline{\text{INTREQ\_X1}}$ | U | X |
| A4 | 43 | IPA4 | B | |
| Intreq0* | 42 | $\overline{\text{INTREQ\_X0}}$ | U | X |
| A3 | 41 | IPA3 | B | |
| Error* | 40 | $\overline{\text{ERROR\_X}}$ | U | X |
| A2 | 39 | IPA2 | B | |
| DMAEnd* | 38 | $\overline{\text{DMAEND\_X}}$ | U | X |
| A1 | 37 | IPA1 | B | |
| Reserved | 36 | RESERVED2 | U | X |
| IOSel* | 35 | $\overline{\text{IOSEL\_X}}$ | U | |
| DMAck0* | 34 | $\overline{\text{DMACK\_X}}$ | U | X |
| IntSel* | 33 | $\overline{\text{INTSEL\_X}}$ | U | |
| DMAReq1 | 32 | $\overline{\text{DMAREQ\_X1}}$ | U | X |
| MemSel* | 31 | $\overline{\text{MEMSEL\_X}}$ | U | |
| DMAReq0 | 30 | $\overline{\text{DMAREQ\_X0}}$ | U | X |
| IDSel* | 29 | $\overline{\text{IDSEL\_X}}$ | U | |
| R/W* | 28 | R/W_X | B | |
| +5V | 27 | +5VDC | +5 Vdc | +5 Vdc |
| GND | 26 | GND | GND | GND |
| GND | 25 | GND | GND | GND |
| +5V | 24 | +5VDC | +5 Vdc | +5 Vdc |
| +12V | 23 | +12VDC | +12 Vdc | +12 Vdc |
| -12V | 22 | -12VDC | -12 Vdc | -12 Vdc |
| BS1* | 21 | $\overline{\text{BS\_X1}}$ | B | |
| BS0* | 20 | $\overline{\text{BS\_X0}}$ | B | |
| D15 | 19 | DATA_X15 | B | X |
| D14 | 18 | DATA_X14 | B | X |

| Original IP Module Signals Names | IP Module Logic Bus Pin # | Systran Signal Names | Bussed or Unique | X = Pulled Up Via 10 KΩ Resistor |
|---|---|---|---|---|
| D13 | 17 | DATA_X13 | B | X |
| D12 | 16 | DATA_X12 | B | X |
| D11 | 15 | DATA_X11 | B | X |
| D10 | 14 | DATA_X10 | B | X |
| D9 | 13 | DATA_X9 | B | X |
| D8 | 12 | DATA_X8 | B | X |
| D7 | 11 | DATA_X7 | B | X |
| D6 | 10 | DATA_X6 | B | X |
| D5 | 9 | DATA_X5 | B | X |
| D4 | 8 | DATA_X4 | B | X |
| D3 | 7 | DATA_X3 | B | X |
| D2 | 6 | DATA_X2 | B | X |
| D1 | 5 | DATA_X1 | B | X |
| D0 | 4 | DATA_X0 | B | X |
| Reset* | 3 | $\overline{\text{RESET\_X}}$ | U | |
| CLK | 2 | ICLK_X | U | |
| GND | 1 | GND | GND | GND |

**NOTE**: The IP Module data bus, usually defined as IPD[15:0] for all of our IP Modules, is a subset of the information that appears on these signal lines. Since carrier register information also uses these signal lines, the bus was defined as the "CPCISC4 Local Data Bus" designated as DATA_X[15:0].

# 3.4 CPCISC4 PCI Base Address

The CPCISC4 PCI base address space is divided into two sections called apertures. The first aperture set in the board's PCI configuration space is aperture 0 which is set for a total of 1 MB of PCI memory space. This aperture contains the address space for I/O, ID, Interrupt Acknowledge and all of the CSR registers on the carrier itself. The second aperture, aperture 1, is set for 64 MB and is for exclusively for memory accesses to the four IP modules. 16 MB is allotted to each IP module since the transactions between the PCI bridge circuit and the IP modules themselves are 32-bit. Note that all accesses to the CPCISC4 will be performed on four-byte boundaries. Also note that the base address set from the BIOS for aperture 0 is termed Base 1 while the address set for aperture 1 will be termed Base 2.

**NOTE**: The mapped PCI memory space is controlled by the BIOS on the system board, and is not normally controlled by the user. Use caution in changing the base address of the CPCISC4 after power-up. The amount of memory space required for the CPCISC4 is provided by the EEPROM upon power-up. This space can be disabled, if necessary, by using the information in Appendix E, PCICFG UTILITY FOR PCI-BASED CARRIERS.

Refer to Appendix B, PROGRAMMING GUIDE, for details of PCI accesses to the CPCISC4 on-board registers or IP Modules.

The CPCISC4 memory base addresses are set in hardware for all IP Modules as below:

- PCI Base2 + 0x0000 0000 for the first location of IP A Memory.
- PCI Base2 + 0x0100 0000 for the first location of IP B Memory.
- PCI Base2 + 0x0200 0000 for the first location of IP C Memory.
- PCI Base2 + 0x0300 0000 for the first location of IP D Memory.

### ACCESS EXAMPLE

Assume it is desired to access the fourth memory location on IP Module D. This would be address 0x0000 000C (four byte boundary) on IP Module D. This equates to address 0x0300 0000 + 0x0000 000C or 0x0300 000C being accessed from the host system. If the BIOS system gives the CPCISC4 a base2 address of, say, 0xF800 0000, then the address on PCI when ~FRAME is asserted will be 0xFB00 000C. The bridge IC on the CPCISC4 will strip away the PCI base address and present 0x0300 000C to the IP bus controller. The controller in turn will decode the upper address bits and begin a memory transaction to address 0x0000 000C on IP Module D.

Accesses to I/O, ID and interrupt acknowledge cycles are done in exactly the same manner. For addresses to these areas, consult Table C-1 in section C-5 of Appendix C.

# APPENDIX A

# SPECIFICATIONS

# A.1 Specifications

## MECHANICAL

Measurements ..... 6.30 inches x 9.19 inches,
160 cm x 233.35 cm
Weight ..... 11.2 oz., 317.52 grams
(includes front panel)
Board thickness ..... 0.062 inches, 0.157 cm, nominally,
(8 layers)

## POWER REQUIREMENTS

Power: ..... +5 Vdc (±5%) @ 260 mA
(No IPs installed)
+12 Vdc @ 0.0 mA
-12 Vdc @ 0.0 mA

**NOTE**: The CPCI bus ground and IP Module ground are not isolated through this board.

## PERFORMANCE CHARACTERISTICS

Electrical Characteristics ..... Refer to Appendix C, PERFORMANCE

## ABSOLUTE MAXIMUM RATINGS

Supply voltage with respect to ground
+5 Vdc ..... -0.5 V Minimum
+7.0 V Maximum
+12 Vdc ..... Dependent on the IP Modules installed
-12 Vdc ..... Dependent on the IP Modules installed

## RECOMMENDED OPERATING CONDITIONS:

Supply voltage
With respect to ground ..... +4.75 Vdc to +5.25 Vdc
Logic Interface ..... ANSI/VITA 4-1995 specification
compliant carrier

## ENVIRONMENTAL SPECIFICATIONS:

Temperature (Operating) ..... 0°C to +70°C
Temperature (no bias, storage) ..... -40°C to +85°C
Humidity (Non-condensing) ..... 5% to 95%

## MEAN TIME BETWEEN FAILURES (MTBF)

MIL-HDBK-217F ..... 312,891 hours
Bellcore 332, Issue 6 ..... 791,666 hours

The MTBF numbers are based on calculations using MIL-HDBK-217F, Appendix A; and Bellcore 332, Issue 6, for a ground-benign environment.

*This page intentionally left blank*

# APPENDIX B

# PROGRAMMING GUIDE

## TABLE OF CONTENTS



## TABLES

# B.1 Overview

This section of the manual describes the operation of the CPCISC4 from the software perspective, detailing the CPCISC4 registers and the overall mapping and addressing scheme for the board, and provides programming examples. A more detailed description of the hardware can be found in Chapter 2, PRODUCT OVERVIEW, and chapter 3, INSTALLATION; and application concepts are discussed in Appendix D, TYPICAL APPLICATIONS.

# B.2 Description

The CPCISC4 is an easy-to-use CPCI slave card that holds four singlewide, two singlewide and one doublewide, or two doublewide IP Modules. In addition to providing access to the I/O and ID space of the IP Modules via the CPCI backplane, the CPCISC4 has eight locations to perform interrupt acknowledge cycles on the IP Modules, and four on-board control/status registers. The CPCISC4 also provides memory space access to all four IP Modules.

# B.3 PCI Configuration Space

The CPCISC4, like all PCI-compliant devices, contains an address space specifically devoted to device identification, as well as memory and I/O configuration. For a comprehensive listing of the configuration register space contents, see the information presented in Table B-1.

# B.4 Board Base Address

The CPCISC4 uses a total of 65 Mbytes of PCI memory space. This space is divided into two parts: a 1 MB section for I/O, ID, INT ACK and board registers (starting with Base1) and a 64 MB section for memory transactions to all four IP modules (starting with Base2). These different addressable areas are set in two different apertures in the PCI configuration space of the CPCISC4. Other configurations are possible. Contact Systran for information regarding additional setups for memory space other than those mentioned in Appendix D.

# B.5 Address Map

The IP Module I/O and ID, and the carrier's control, status, and general-purpose registers, reside at a fixed relative address from the CPCISC4 base1 address.

The IP module memory space for all four IP modules reside at fixed Base2 address locations relative to the CPCISC4 Base2 address.

Table B-1 shows the complete memory map for the board.

**Table B-1 CPCISC4 Address Map**

| Address Space or Register Name | Local Address (relative to board base address) | Size |
|---|---|---|
| **Base1** | | |
| IP Module A I/O | Base1 + 0x0000 | 128 bytes* |
| IP Module A ID | Base1 + 0x0100 | 128 bytes* |
| IP Module B I/O | Base1 + 0x0200 | 128 bytes* |
| IP Module B ID | Base1 + 0x0300 | 128 bytes* |
| IP Module C I/O | Base1 + 0x0400 | 128 bytes* |
| IP Module C ID | Base1 + 0x0500 | 128 bytes* |
| IP Module D I/O | Base1 + 0x0600 | 128 bytes* |
| IP Module D ID | Base1 + 0x0700 | 128 bytes* |
| IP Module Reset Register | Base1 + 0x0A00 | 4 bits |
| Error Status Register | Base1 + 0x0A04 | 4 bits |
| Carrier Control Register | Base1 + 0x0A08 | 4 bits |
| IP Module A IRQ0 Register | Base1 + 0x0B00 | 8 bits |
| IP Module A IRQ1 Register | Base1 + 0x0B04 | 8 bits |
| IP Module B IRQ0 Register | Base1 + 0x0B08 | 8 bits |
| IP Module B IRQ1 Register | Base1 + 0x0B0C | 8 bits |
| IP Module C IRQ0 Register | Base1 + 0x0B10 | 8 bits |
| IP Module C IRQ1 Register | Base1 + 0x0B14 | 8 bits |
| IP Module D IRQ0 Register | Base1 + 0x0B18 | 8 bits |
| IP Module D IRQ1 Register | Base1 + 0x0B1C | 8 bits |
| Interrupt Steering Register | Base1 + 0x0B28 | 9 bits |
| **Base2** | | |
| IP Module A Memory Base | Base2 + 0x0000 0000 | 8 MB* |
| IP Module B Memory Base | Base2 + 0x0100 0000 | 8 MB* |
| IP Module C Memory Base | Base2 + 0x0200 0000 | 8 MB* |
| IP Module D Memory Base | Base2 + 0x0300 0000 | 8 MB* |

* This denotes the actual number of bytes used by the respective address space. For each 32-bit memory location, only the lower 16 bits are actually used. The CPCISC4 and any installed IP Modules are typically accessed via 8-bit or 16-bit operations. If 32-bit operations are used, the upper 16-bits are discarded by software.

**NOTE**: Base1 refers to the base address of the first aperture in the PCI-to-Local bridge IC and Base2 refers to the base address of the second aperture. All addresses shown in Table B-1 are those on the local side of the PCI interface.

## B.6 Register Descriptions

The CPCISC4 registers can be accessed as byte (8-bit) or word (16-bit) values. The unused bits are driven as 0's on reads. All unused bits are discarded on writes. The hexadecimal addresses for the registers shown below are relative to the board base address and are in the WORD format for word and byte accesses, respectively.

## B.7 Word Access Address Translation

The following table shows the relationship between PCI space addresses and the IP Module address for word accesses. In the table, BASE represents the I/O, ID, or memory base address. All addresses are in hexadecimal.

**NOTE:** By default, the PCI_BASE0 and PCI_BASE1 aperture's byte swapping is disabled by the appropriate bits in the PCI_MAP0 and PCI_MAP1 registers. If these bits are changed to implement byte swapping in the PCI interface, the following table will no longer be correct.

**Table B-2 Word Access Address Translation Table**

| PCI bus Address | IP Module Address |
|---|---|
| BASE + 0x00 | IPA = 0x00 |
| BASE + 0x04 | IPA = 0x01 |
| BASE + 0x08 | IPA = 0x02 |
| BASE + 0x0C | IPA = 0x03 |
| BASE + 0x10 | IPA = 0x04 |
| BASE + 0x14 | IPA = 0x05 |
| BASE + 0x18 | IPA = 0x06 |
| BASE + 0x1C | IPA = 0x07 |
| BASE + 0x20 | IPA = 0x08 |
| BASE + 0x24 | IPA = 0x09 |
| BASE + 0x28 | IPA = 0x0A |
| … | … |
| BASE + 0xFC | IPA = 0x3F |

# B.8 Byte Access Address Translation

The following table shows the relationship between PCI space addresses and the IP Module address for byte accesses. In the table, BASE represents the I/O, ID, and memory base address. All addresses are in hexadecimal.

**NOTE:** By default, the PCI_BASE0 and PCI_BASE1 aperture's byte swapping is disabled by the appropriate bits in the PCI_MAP0 and PCI_MAP1 registers. If these bits are changed to implement byte swapping in the PCI interface, the following table will no longer be correct.

**Table B-3 Byte Access Address Translation Table**

| PCI bus Address | IP Module Address | Byte Lane* |
|---|---|---|
| BASE + 0x00 | IPA = 0x00 | 0 |
| BASE + 0x01 | IPA = 0x00 | 1 |
| BASE + 0x04 | IPA = 0x01 | 0 |
| BASE + 0x05 | IPA = 0x01 | 1 |
| BASE + 0x08 | IPA = 0x02 | 0 |
| BASE + 0x09 | IPA = 0x02 | 1 |
| BASE + 0x0C | IPA = 0x03 | 0 |
| BASE + 0x0D | IPA = 0x03 | 1 |
| BASE + 0x10 | IPA = 0x04 | 0 |
| BASE + 0x11 | IPA = 0x04 | 1 |
| … | … | … |
| BASE + 0xFC | IPA = 0x3F | 0 |
| BASE + 0xFD | IPA = 0x3F | 1 |

* Byte-lane 1 not applicable for ID Space

## B.8.1 Reset Register (0xA00)

Setting one of the bits RES_D through RES_A to '1' asserts the corresponding IP Module's reset line. The bits can be set individually by writing a '1' to that bit or simultaneously with a single write of '0xF.' When set, a one-shot is triggered supplying a minimum 200 ms reset pulse as required by the IP Module specification. When read, these bits return the status of the corresponding IP Module's reset line (a '0' means the IP Module is not being reset, a '1' means it is in a reset condition).

**Table B-4 Reset Register Bit Description**

| **Bit #** | B15-B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|
| **Bit Name** | Not Used | RES_D | RES_C | RES_B | RES_A |
| **R/W** | Reads '0's Writes discarded | R/W | R/W | R/W | R/W |
| **Power-up State** | N/A | 0 | 0 | 0 | 0 |

**NOTE**: RES_D-RES_A: Reset bits for IP Module slots D to A.

## B.8.2 Error Status Register (0xA04)

These bits indicate the state of the corresponding IP Module $\overline{\text{ERROR}}$_X signal (pin #40 on a given IP Module). If the $\overline{\text{ERROR}}$_X signal is asserted (active low) for a particular IP Module, then the corresponding ERR_X bit will be set to a '1.' If the error signal is not asserted, then the corresponding bit will be cleared.

**Table B-5 Error Status Register Bit Description**

| **Bit #** | B15-B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|
| **Bit Name** | Not Used | ERR_D | ERR_C | ERR_B | ERR_A |
| **R/W** | Reads '0's Writes discarded | R | R | R | R |
| **Power-up State** | N/A | State of IP Module D ERR* Signal | State of IP Module C ERR* Signal | State of IP Module B ERR* Signal | State of IP Module A ERR* Signal |

**NOTE**: ERR_D-ERR_A: Error Status bits for IP Module slots D to A.

## B.8.3 Carrier Control Register (0xA08)

This register contains bits that are used to enable interrupts and implement the time-out function on the CPCISC4. Table B-6 shows the bits in the Carrier Control register. A brief description of each bit follows.

- The IP_INT_EN bit must be set to '1' to enable IP Module interrupt requests to assert a PCI interrupt. This is necessary to prevent interrupts from occurring before the application software has installed an appropriate interrupt service routine.
- The TO_EN bit is used to enable the time-out function. When this bit is set to '1' and the Time-out jumper (J5) is in the enabled position (shunt removed), the carrier completes the local bus transaction after 30 IP Module clock cycles have elapsed from the start of an IP Module access. This time-out feature is used to prevent the local bus of the CPCISC4 from "locking up" if an IP Module access fails.
- The TO_INT_EN bit is used in conjunction with the TO_EN bit to generate a PCI interrupt when a time-out occurs. When this bit is set to '1' and the Time-out jumper (J5) is in the enabled position (shunt removed), a PCI interrupt will be generated to alert the software that an IP Module access was unsuccessful.
- The TO_STAT bit indicates if a time-out occurred. This bit is set to a '0' when an IP Module access is successful and it is set to a '1' when a time-out occurs.

**Table B-6 Carrier Control Register Bit Description**

| **Bit #** | B15-B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|
| **Bit Name** | Not Used | TO_STAT | TO_INT_EN | TO_EN | IP_INT_EN |
| **R/W** | Reads '0's Writes discarded | R | R/W | R/W | R/W |
| **Power-up State** | N/A | 0 | 0 | 0 | 0 |

**NOTE**: Do not enable the time-out feature if an IP Module that asserts more than 28 wait states is used.

## B.8.4 Interrupt Vector Registers for IP Modules A through D (0xB00 through 0xB1C)

The eight interrupt vector registers (two per IP Module) are 8-bit, read-only registers. When one of these registers is read, an appropriate interrupt acknowledge cycle is performed on the corresponding IP Module. These registers reflect the vector driven on the IP Module data bus during the appropriate interrupt acknowledge cycle. These registers are useful for IP Modules that have a ROAK (release on acknowledge) interrupt mechanism, as the PCI interrupt cycle is not vectored, unlike some other buses (for example, VME). This means a ROAK-type IP Module will need to use these registers to clear the interrupt source. Software for RORA (release on register access) IP Modules will typically not use these registers.

**Table B-7 Interrupt Vector Registers**

| Address | Interrupt Vector Register | Address | Interrupt Vector Register |
|---|---|---|---|
| Base + 0xB00 | IP Module 'A' Intreq0 | Base + 0xB04 | IP Module 'A' Intreq1 |
| Base + 0xB08 | IP Module 'B' Intreq0 | Base + 0xB0C | IP Module 'B' Intreq1 |
| Base + 0xB10 | IP Module 'C' Intreq0 | Base + 0xB14 | IP Module 'C' Intreq1 |
| Base + 0xB18 | IP Module 'D' Intreq0 | Base + 0xB1C | IP Module 'D' Intreq1 |

## B.8.5 Interrupt Steering Register (0xB28)

The Interrupt Steering Register is an 9-bit register consisting of an interrupt-status bit for each of the two interrupt-request lines for all four IP Modules, and one local time-out status bit. This register is typically read in the Interrupt Service Routine to determine which interrupt source(s) is (are) being asserted. A table of the bits in the Interrupt Steering Register is shown below. A write to this register will clear a pending time-out-interrupt request.

**Table B-8 Interrupt Steering Register**

| Bit # | B15-B9 | B8 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Bit Name | Not Used | TO_INT | IRQ1D | IRQ0D | IRQ1C | IRQ0C | IRQ1B | IRQ0B | IRQ1A | IRQ0A |
| R/W | Reads '0's Writes discarded | R/W | R | R | R | R | R | R | R | R |
| Power-up State | N/A | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**NOTE**: Each bit will be a '1' when the corresponding interrupt is set.

# B.9 Programming Examples

The following examples illustrate how to program the CPCISC4 to achieve various operational modes.

## B.9.1 Reset Example

This example resets one, then all FOUR IP Modules, and then monitors the reset signals to determine when they are de-asserted. Setting a bit in the reset register causes a one-shot to assert the corresponding IP Module reset signal (the one-shot asserts the reset signal for approximately 240 ms). The reset signal can then be monitored by reading the Reset register to determine when it is de-asserted.

- Write '0x01' to the Reset register. This resets the IP Module in slot A by asserting the IP Module A reset signal.

### LOOP1:

- Read the Reset register.
- If Bit 0 is set then go to LOOP1.
- Write '0xF' to the Reset register. This resets all four IP Modules by asserting their corresponding reset signals.

### LOOP2:

- Read the Reset register.
- If any of bits [3:0] are set then go to LOOP2.

## B.9.2 Interrupt Initialization Example

Interrupts in the PCI-based CPCI system are sensed by the system board at an active-low level with the PCI INTA# interrupt-request line remaining at a logic-low until the system CPU has accessed any necessary registers and removed the interrupt source. On the CPCISC4, interrupts are disabled at power-up and must remain disabled until the entire system is properly initialized. After this, IP Module interrupts and time-out interrupt can then be enabled by writing the appropriate bits to the Control Register on the CPCISC4. The interrupt line from the CPCISC4 (PCI INTA#) is asserted from any one of the nine possible sources. Eight of the sources are from N_INTREQ0 and N_INTREQ1 from each of the four IP Modules and the ninth source is from a time-out interrupt-status bit that is set in the IP Module interface controller. The status of these interrupt requests can be determined by a read of the Interrupt Steering Register on the CPCISC4.

An example is given in Appendix D, TYPICAL APPLICATIONS.

# APPENDIC C

# PERFORMANCE

# TABLE OF CONTENTS



# FIGURES

# C.1 Overview

The purpose of this section is to provide several sets of empirical data that represent typical performance parameters beyond those provided in the specification. It is important to note that these are typical responses for the configuration cited, and do not supplant the maximum and minimum envelopes presented in Appendix A.

These parameters were taken with the CPCISC4 installed in a 166 MHz Pentium, 33 MHz PCI bus PC system. Test software was running in DOS mode. Results from any other system configuration will vary from those shown here. The CPCISC4 hardware and the IP Module bus times (assuming no WAIT states) will remain relatively the same as those shown.

# C.2 PCI Data Transfer Performance

Data transfers to three different areas on the CPCISC4 are performed via the PCI bus: Control/Status Register (CSR) area, IP Module (I/O, ID, and Memory) area, and the configuration registers of the V3 PCI-to-Local Bridge chip. Each area has it its own unique access time and therefore, performance aspects.

Performance information for the CSR and IP Module is provided in the subsequent subsections. Performance information for the V3 chip register area is included in the discussion in section C.3.

## C.2.1 CSR Write/Read Cycles

CSR accesses involve the registers internal to the Altera IC in an Altera EPLD. These registers include the Reset Register, the Error Status Register, Carrier Control Register and the Interrupt Steering Register (ISR).

### WRITE CYCLES

Write-cycle performance is aided by the Write Posting feature being enabled on the V3 chip. This, coupled with the V3's write FIFO, allows for maximum write performance from the PCI bus to the CPCISC4 CSR area. This means that PCI write cycles are acknowledged almost immediately by the V3 chip on the PCI bus. However, the write to the addressed CSR register does not actually take place until an acknowledge is generated by the Altera IC on the V3's local bus side. Figure 4-1 is a graphical representation of the cycles involved during a CSR Write followed by the times for the cycles involved.

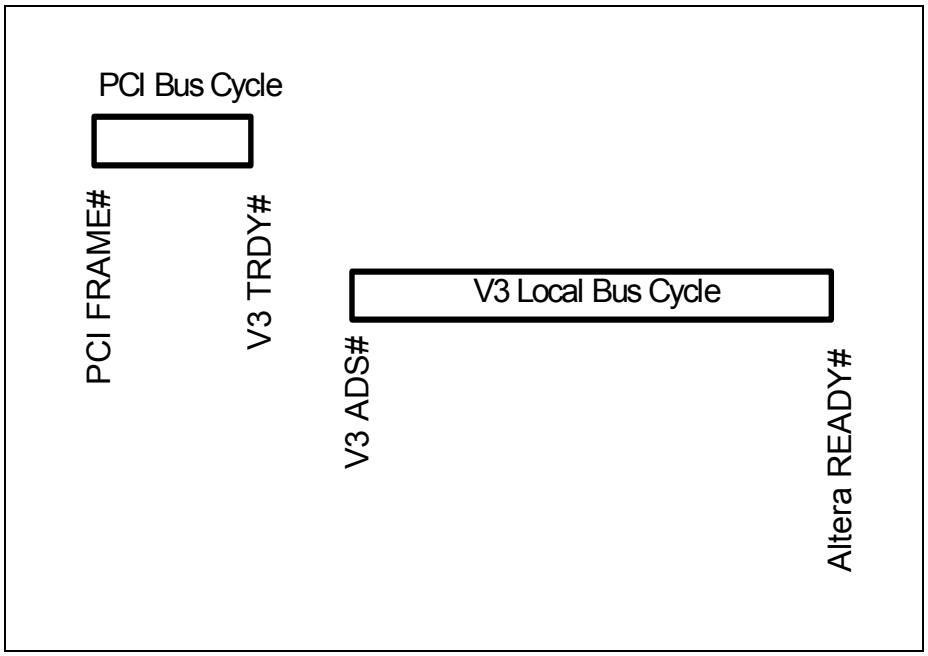


**Figure C-1 CSR Write Cycle**

*

| | | |
|---|---|---|
| PCI bus FRAME# to V3 TRDY# | (data Write Posted) | 121.2 ns |
| PCI FRAME # to V3 ADS# | (begin local bus cycle) | 187.5 ns |
| V3 ADS# to Altera READY# | (end local bus cycle) | 236.8 ns |
| Total cycle time | | 545.5 ns |

* PCI bus is free at this time.

## READ CYCLES

Read accesses to the CSR registers will be faster than the IP Module areas but not as fast as the V3 register area. Transfers to this area go through the V3 first and then the Altera IC. Unlike Write Posting, the PCI bus must wait until the read data is ready from the Altera IC and placed on the bus and the V3 asserts TRDY# before the cycle is finished. Figure 4-2 is a graphical representation of the cycles involved during a CSR Read followed by the times for the cycles involved.

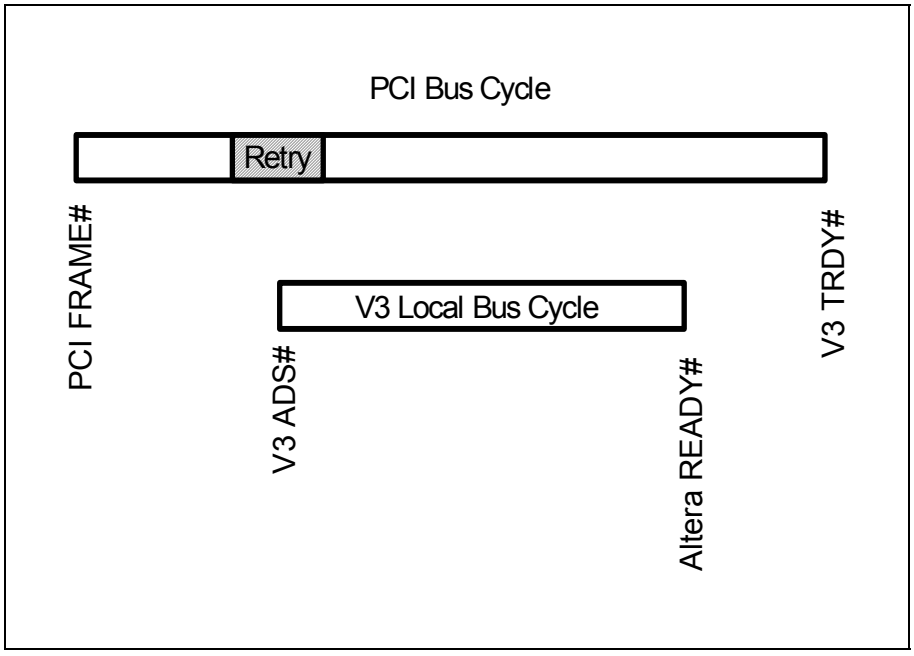


**Figure C-2 CSR Read Cycle**

| | | |
|---|---|---|
| PCI bus FRAME# to V3 ADS# | (begin local bus cycle) | 200.0 ns |
| V3 ADS# to Altera READY# | (end local bus cycle) | 125.0 ns |
| Altera READY# to V3 TRDY# | (end PCI bus cycle) | 219.0 ns |
| Total cycle time | | 540.0 ns |

## C.2.2 I/O, ID, and Memory Write/Read Cycles

IP Module accesses will have the slowest access time of the three areas. This is because any PCI access must go through the V3 chip, the Altera IC and then to the IP Module bus and wait for the IP Module to generate an "acknowledge".

### WRITE CYCLES

As stated before, write-cycle performance is aided by Write Posting and the V3's write FIFO. However, the write does not actually take place until the addressed IP Module generates an N_ACK to the Altera IC and, subsequently, the Altera IC signals the end of the cycle with a READY to the V3 chip. Figure 4-3 is a graphical representation of the cycles involved during a IP Module Write followed by the times for the cycles involved.

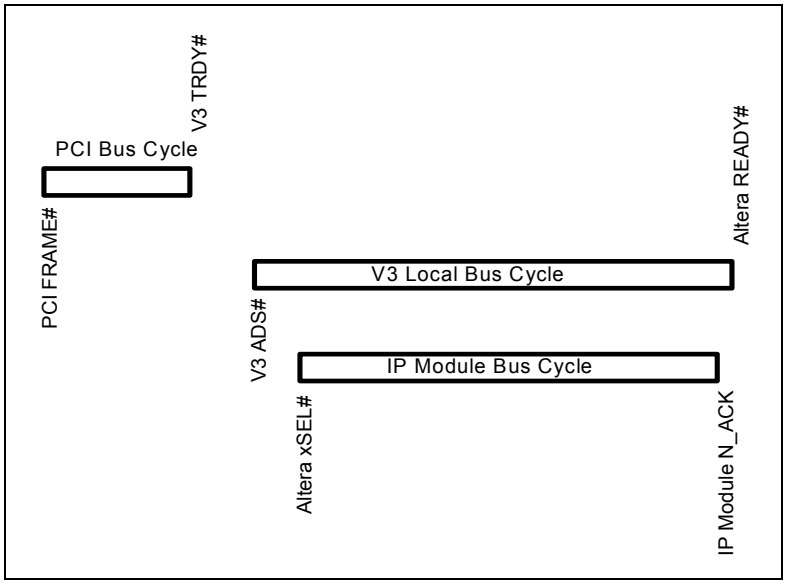


**Figure C-3 IP Module Write Cycle**

| | | |
|---|---|---|
| PCI bus FRAME# to V3 TRDY# | (data Write Posted) | 121.2 ns |
| PCI FRAME # to V3 ADS# | (begin local bus cycle) | 90.9 ns |
| V3 ADS# to Altera xSEL# | (begin IP Module bus cycle) | 125.0 ns |
| Altera xSEL# to IP N_ACK | (end IP Module bus cycle) | 125.0 ns |
| IP N_ACK to IP Interface READY# | (end local bus cycle) | 62.5 ns |
| * Total cycle time | | 403.4 ns |

*

*

* PCI bus is free at this time.

## READ CYCLES

One of the biggest limitations to PCI performance will be the number of PCI-bus retries caused by the PCI Target not being ready with the requested data during a Read cycle. In the case of the CPCISC4, this is a function of the speed at which the Local bus is running and the number of IP Module bus WAIT states that are inserted by the IP Module during an IP Module bus cycle. Another factor is HOLD states inserted by the IP Module carrier into the IP Module bus cycle. The CPCISC4 does not insert any HOLD states. As in the CSR reads, the PCI cycle cannot complete until the data from the addressed device (the IP Module in this case) is placed on the PCI bus and TRDY# is asserted. Figure 4-4 is a graphical representation of the cycles involved during a IP Module Read followed by the times for the cycles involved.

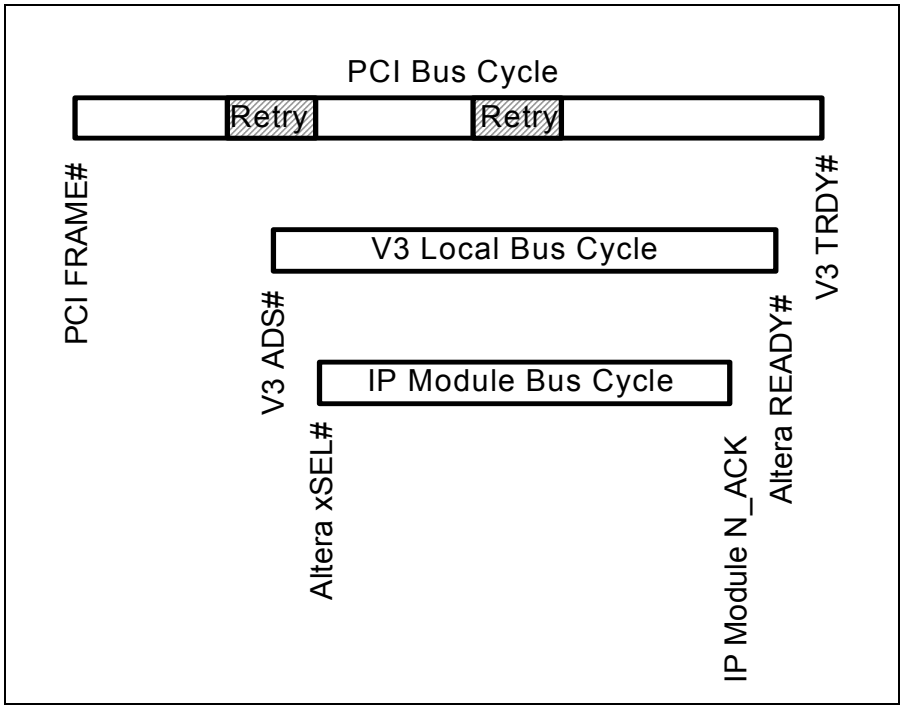


**Figure C-4 IP Module Read Cycle**

| | | |
|---|---|---|
| PCI bus FRAME# to V3 ADS# | (begin local bus cycle) | 90.9 ns |
| V3 ADS# to Altera xSEL# | (begin IP Module bus cycle) | 187.5 ns |
| Altera xSEL# to IP N_ACK | (end IP Module bus cycle) | 125.0 ns |
| IP N_ACK to Altera READY# | (end local bus cycle) | 62.5 ns |
| Altera READY# to V3 TRDY# | (end PCI bus cycle) | 321.9 ns |
| Total cycle time | | 787.8 ns |

# C.3 Interrupt Latency

Interrupt latency is a measure of the time that has elapsed from the generation of an interrupt to that interrupt being serviced. For the CPCISC4 the interrupt service includes a read of the Altera's Interrupt Steering Register to determine which interrupt on which IP Module occurred. An additional access to the IP Module is required in order to clear the IP Module interrupt. Also, a read and a write to the V3 chip's internal registers is required to clear the PCI interrupt. All register read cycle times includes the time for PCI Retries. The interrupting source is from an IP Module. Figure 4-5 is a graphical representation of the cycles involved during an Interrupt followed by the times for the cycles involved.

**NOTE**: The system Interrupt Acknowledge and Interrupt Handler software execution time is part of the overall latency and is highly dependent upon the speed of the system used and the size and efficiency of the Interrupt Handler software.

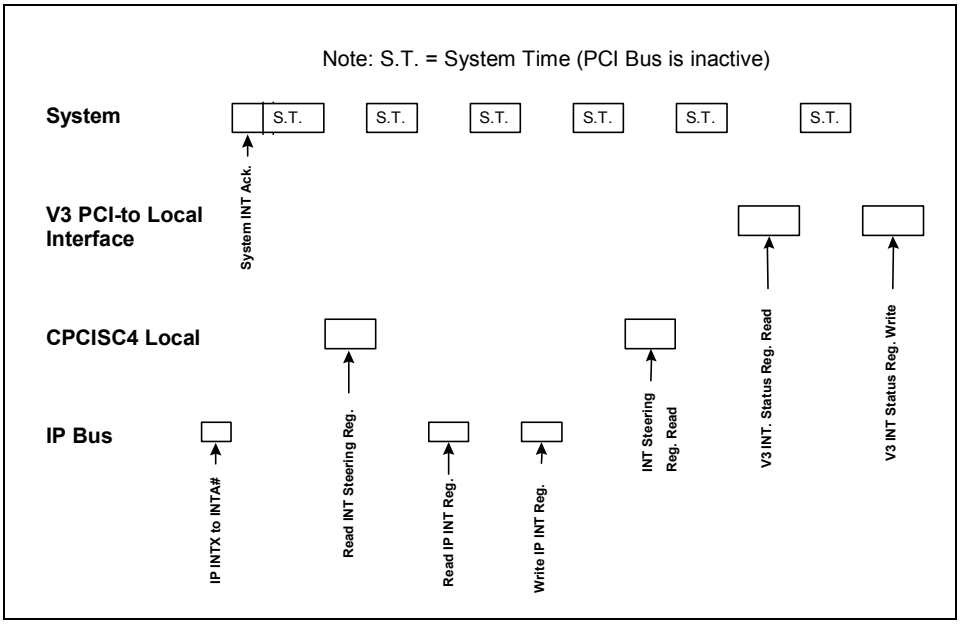


**Figure C-5 Interrupt Cycle**

| | | | |
|---|---|---|---|
| IP INTX# to PCI INTA# | 00.034 µs | IP INT Reg. Write | 00.121 µs |
| System INT ACK | 01.424 µs | System Time | 00.909 µs |
| System Time | 05.999 µs | INT Steering Reg. Read | 00.545 µs |
| Read INT Steering Reg. | 00.576 µs | System Time | 00.454 µs |
| System Time | 02.242 µs | V3 INT Reg. Read | 00.152 µs |
| Read IP INT Reg. | 00.788 µs | System Time | 00.212 µs |
| System Time | 02.060 µs | V3 INT Reg. Write | 00.182 µs |
| Total Time | 15.694 µs | | |

* System times are included in Total Time.

# APPENDIX D

# TYPICAL APPLICATIONS

# TABLE OF CONTENTS



# TABLES

# D.1 Applications

Systran extends an open invitation to all users to freely submit their applications that might, or do, use the CPCISC4 Slave Carrier to solve a problem. This section of the manual will be revised periodically to include new application ideas for all users to consider. Help advance the level of technology by participating with the Systran team, while simultaneously publishing your ideas. Submission constitutes permission to publish without additional consent or compensation, and Systran reserves the right to modify submissions to provide for more generic appeal, when necessary.

# D.2 Typical Software Considerations in PC Environments

The use of the CPCISC4 may initially appear daunting due to the many options provided by the PCI-interface chip V3 962PBC-B2 (hereafter referred to as "PBC") on the CPCISC4. Fortunately, most of the registers are initialized by the on-board EEPROM and/or the BIOS on the system board and are consequently transparent to the user. Typically, the only PBC registers used by user software are the following: PCI_IO_BASE, PCI_BASE0, PCI_BASE1, PCI_BPARAM, and PCI_INT_STAT. The following examples discuss how these registers may be used to access the CPCISC4 and handle interrupts appropriately. This description is generic and does not specifically discuss operating system-specific implementation issues.

1. Verify that a PCI-compliant BIOS is present on the system board.
2. Verify that at least one CPCISC4 is present in the system. The Device ID and Vendor ID are typically used for this purpose. For each CPCISC4 card found in the system, perform steps 3-6.
3. Perform a 32-bit read of the PCI_IO_BASE register. The result obtained by masking off the lower 8 bits (AND with 0xFFFFFF00) of this value is the base address used to access the PBC's internal registers. For purposes of this discussion, this address will be designated **CPCISC4_Config_Pointer**. This base address is used for all subsequent accesses to the PBC.
4. Perform a 32-bit read of the PCI_BASE0 register (**CPCISC4_Config_Pointer** + 0x14). The result obtained by masking off the lower 8 bits (AND with 0xFFFFFF00) of this value is the base1 address used to access the CPCISC4 registers and the IP Modules (see Appendix C for the entire address map). For the discussion that follows, this address will be designated **CPCISC4_IP_Space_Pointer**. This base address is used for I/O, ID, Interrupt Acknowledge and carrier register accesses. It is not used for memory accesses.
5. Perform a 32-bit read of the PCI_BASE1 register (**CPCISC4_Config_Pointer** + 0x18). The result obtained by masking off the lower 8 bits (AND with 0xFFFFFF00) of this value is the base 2 address used to access the CPCISC4 IP Modules for memory transactions (see Appendix C for the entire address map). This address will be designated **CPCISC4_MEM_Space_Pointer** and is used for IP memory accesses only.
6. Perform a 32-bit read of the PCI_BPARAM register (**CPCISC4_Config_Pointer** + 0x3C). The lower eight bits of this value denote the interrupt line that will be used by the CPCISC4 during interrupts. For an x86-based computer, a valid interrupt line will be in the range '0' though '15'. Use this value when installing an Interrupt Service Routine because the system has been configured to generate this level of interrupt to the microprocessor when the CPCISC4 asserts its interrupt request.

7. IP Modules or CPCISC4 internal registers can now be accessed as appropriate by using appropriate offsets from the board base1 or base2 address (**CPCISC4_IP_Space_Pointer** or **CPCISC4_MEM_Space_Pointer**), as listed in Appendix C.

**NOTE**: Steps 1 though 3 are platform and/or operating-system specific. For instance, BIOS calls (or higher-level API) may be used to perform steps 1-3. After the **CPCISC4_Config_Pointer** is determined in step 3, do not continue to use BIOS calls to access the hardware because they must not be invoked from an Interrupt Service Routine

**NOTE**: Do not overwrite the values in the PCI_IO_BASE, PCI_BASE0, PCI_BASE1 or PCI_BPARAM registers, as initialized by the BIOS. These values cannot be simply changed without having potential effects on the operability of the CPCISC4 due to the "plug-and-play" nature of PCI.

## D.3 Typical Interrupt Handling in a PC Environment

The previous example illustrated how to access the PCI interface chip's internal registers, the CPCISC4's registers, and any installed IP Modules. This example demonstrates how interrupts can be handled using the CPCISC4.

- Upon initialization of program, save the current address of the currently installed Interrupt Service Routine (if any) for applicable interrupt line. The interrupt line used by the CPCISC4 was determined in Step 5 of the previous example.
- Install the desired ISR for the appropriate interrupt line.
- Initialize the applicable programmable interrupt controller (8259A-type device)
- Enter the main loop of program here.
- During program's termination routine, restore the old ISR before exiting.

The following is a suggested ISR for the CPCISC4:

- If applicable for ISR, set the programmable interrupt controller (PIC) to enable higher-priority interrupts to preempt this ISR.
- Read the steering register (**CPCISC4_IP_Space_Pointer** + 0xB28) to verify that the interrupt was asserted from the CPCISC4.
- If not, call the previous ISR at this interrupt line, if any, and exit.
- If one or more interrupts from the CPCISC4, service appropriately.

The following is a typical routine, which prioritizes the IP Module interrupt requests in the following order:

**IP Module A Interrupt Request 0 through IP Module D Interrupt Request 1.**

```
      Read Steering Register
      "AND" result with 1FFh
      if (non-zero)
            while result non-zero
                  if (bit 0 set)
                        handle IPA - IRQ0
                        clear source
                  else if (bit 1 set)
                        handle IPA - IRQ1
                        clear source
                  .
                  .
                  .

                  else if (bit 7 set)
                        handle IPD - IRQ 1
                        clear source
                  reread the steering register and "AND" result with
1FFh
            Perform 32-bit read of PCI_INT_STAT
(CPCISC4_Config_Pointer)
            Write this value back to PCI_INT_STAT to clear interrupt
from PBC
            Issue "end-of-interrupt" command to PIC
      else
            call previous ISR if exists
      return from exception
```

**NOTE**: When processing of the interrupt(s) source(s) is complete, read the PCI_INT_STAT register and write this value immediately back to the register to clear the interrupt from the PCI interface chip. Disable the Compiler optimization to avoid removal of this code.

# D.4 962PBC Register Map

As mentioned in previous sections, the PCI_IO_BASE, PCI_BASE0, PCI_BASE1, PCI_BPARAM, and PCI_INT_STAT registers in the PBC Configuration space are necessary to access the CPCISC4. Since the above registers represent a small subset of the entire PBC space, Table D-1 provides a complete listing of the PBC registers as initialized by the on-board EEPROM. For a more detailed description of each register, contact QuickLogic (V3 Semiconductor) at the addresses shown in Section 1.2.

**Table D-1 PBC Default Register Values**

| Name | Width | Offset | Value As Set by EEPROM |
|---|---|---|---|
| PCI_VENDOR | 16 bits | 0x00 | 0x1387 |
| PCI_DEVICE | 16 bits | 0x02 | 0x4903 |
| PCI_CMD | 16 bits | 0x04 | 0x0000 |
| PCI_STAT | 16 bits | 0x06 | 0xF080 |
| PCI_CC_REV | 32 bits | 0x08 | 0x08800000 |
| PCI_HDR_CFG | 32 bits | 0x0C | 0x00000000 |
| PCI_IO_BASE | 32 bits | 0x10 | 0x00000000 |
| PCI_BASE0 | 32 bits | 0x14 | 0x00000008 |
| PCI_BASE1 | 32 bits | 0x18 | 0x04000008 |
| | | | |
| PCI_SUB_VENDOR | 16 bits | 0x2C | 0x0000 |
| PCI_SUB_ID | 16 bits | 0x2E | 0x0000 |
| PCI_ROM | 32 bits | 0x30 | 0x00000000 |
| | | | |
| PCI_BPARAM | 32 bits | 0x3C | 0x00000100 |
| PCI_MAP0 | 32 bits | 0x40 | 0x00000003 |
| PCI_MAP1 | 32 bits | 0x44 | 0x04000063 |
| PCI_INT_STAT | 32 bits | 0x48 | 0x00000000 |
| PCI_INT_CFG | 32 bits | 0x4C | 0x00020002 |
| | | | |
| LB_BASE0 | 32 bits | 0x54 | 0x00000000 |
| LB_BASE1 | 32 bits | 0x58 | 0x00000000 |
| | | | |
| LB_MAP0 | 16 bits | 0x5E | 0x0006 |
| LB_MAP1 | 16 bits | 0x62 | 0x0006 |
| LB_SIZE | 32 bits | 0x68 | 0x00000000 |
| LB_IO_BASE | 16 bits | 0x6E | 0x0000 |
| FIFO_CFG | 16 bits | 0x70 | 0x0A00 |
| FIFO_PRIORITY | 16 bits | 0x72 | 0x0F0F |
| FIFO_STAT | 16 bits | 0x74 | 0x0000 |
| LB_ISTAT | 8 bits | 0x76 | 0x00 |
| LB_IMASK | 8 bits | 0x77 | 0x00 |
| SYSTEM | 16 bits | 0x78 | 0xC400 |

# APPENDIX E

# PCICFG UTILITY FOR PCI-BASED CARRIERS

## TABLE OF CONTENTS



## FIGURES

# E.1 Overview

PCs are continuing to become popular IP module carrier platforms. PCI is currently the primary PC local bus standard and is widely used as slots on a PC motherboard or as the CompactPCI form factor. One reason for such popularity is the "plug-and-play" nature of PCI-BIOS. However, the way PC BIOS systems map add-in card memory tends to differ from one system to another and in some cases can be problematic. Some add-in boards can be ignored or can cause a system to "hang" if they require a greater amount of memory to operate than the particular BIOS allows.

# E.2 Purpose

The **PciCfg** utility allows quick and easy modification of the PCI configuration EEPROM. The modification entails an enable or disable of the memory space of the IP carrier. When memory space is disabled, only one megabyte of system space is required to place the carrier which allows proper operation in systems with a "memory sensitive" BIOS. However, if memory space is necessary, it can be enabled, which will require an additional 32 MB to 128 MB depending on which PCI-based carrier is in use.

# E.3 Product Coverage

This software was designed for use with the following Systran PCI-based IP module carriers:

- BHAS-PCISC5
- BHAS-PCISC3
- BHAS-CPCISC4
- BHAS-CPCISC2

**NOTE**: This software is not intended for use with Systran's VME-based IP carriers or with any competitor's carrier.

## E.3.1 OS Platform

This software operates under Windows NT 4.0. It was not designed to operate from another version of Windows or from DOS.

# E.4 Hardware Installation

The **PciCfg** utility supports up to 16 installed IP Module carrier cards on a single machine. The cards to be configured are inserted into any available PCI or CPCI slots on the host system. Once the software is installed, cards may be inserted or removed and the software will automatically identify changes upon reboot.

It will also be necessary to install a jumper plug across the header connected to the write-protect-disable function on the carrier's EEPROM. The location of the jumper header for each carrier is listed below:

BHAS-PCISC5.....Install a plug across J17 located next to the EEPROM in the upper-left end of the board with the mounting bracket facing left.

BHAS-PCISC3.....Install a plug across J9 located next to the EEPROM in the upper-right end of the board with the mounting bracket facing left.

BHAS-CPCISC4.....Install a plug across J3 located above the EEPROM in the lower-left end of the board with the panel and I/O connectors facing away.

BHAS-CPCISC2.....Install the plug across both pins on J3 located in the lower-right end of the board with the panel and I/O connectors facing away.

**NOTE**: The **PciCfg** program will be blocked from re-programming the EEPROM if the jumper is not properly installed.

## E.5 Software Installation

The **PciCfg** utility is distributed on a standard 3.5" diskette. To install, insert the diskette into the host computer's removable-diskette drive. Select "Run" from the Windows NT "Start" menu and type the following in the Open edit box.

```
a:setup
```

The **PciCfg** installation utility will guide you through the setup process. Upon completion, the host system must be rebooted and **PciCfg** is then ready for operation.

## E.6 Software Operation

To run **PciCfg**, select the appropriate icon in the **PciCfg** group of the Windows NT start menu. This brings up the main application window as shown in Figure E-1.

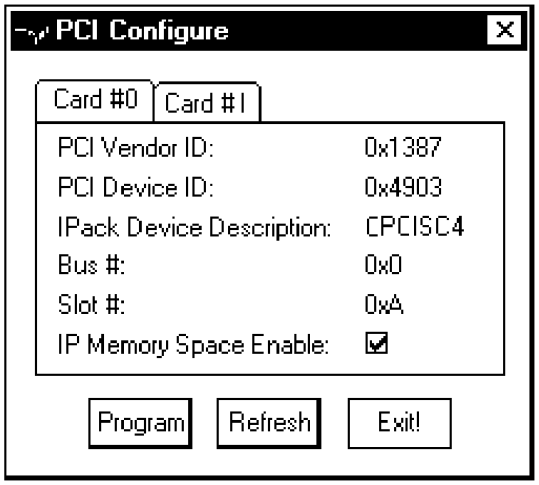


**Figure E-1 Application Main Window**

The main window displays the information for the first Systran IP Module carrier card found on the system. To display other devices installed on the system, simply select the appropriate "Card #" tab at the top of the window.

Various device characteristics are displayed for the selected card including PCI vendor and device ID, Systran IP Module model number as well as the PCI bus and slot numbers.

In addition to the general PCI information, a selection checkbox is provided to turn off the IP Memory Space. To disable, de-select the checkbox and press the **Program** button. This reconfigures the PCI EEPROM on the selected device, effectively disabling the IP Memory Space. The host system must then be powered down and rebooted for the new settings to take effect.

The **Refresh** button on the main display window re-reads the selected devices EEPROM and refreshes the display. The **Exit** button closes **PciCfg**.

*This page intentionally left blank*

# GLOSSARY

[x:y]----------Nomenclature designating a bit-range, where “x” is the left-most bit and “y” is the right-most bit (for example, data bus [7:0] refers to the least significant eight bits).

aperture----------A region or set of addresses.

byte-lane ----------Eight bits of a data bus on octal boundaries.

CSR----------Control and Status Register.

doublewide ----------An IP Module that is twice the size of the singlewide board.

EPLD----------Erasable Programmable Logic Device.

IP Module----------Business-card size mezzanine-type subsystems designed with a common digital interface known as the IP bus. These field-installable plug-and-play modules are automatically recognized by system software. An open industry standard defines the mechanical and electrical interface to the carrier board.

IP Module logic bus----------A synchronous, 4 MTransfers/sec, 16-bit wide bus that includes I/O, memory, ID PROM, and interrupts. The address bus is 6-bits wide, except in memory mode. Then the data bus is multiplexed for the upper portion of the address bus, resulting in 22 bits of address. This results in up to 4 Mwords of memory space per IP Module.

ISR----------Interrupt Service Routine, Interrupt Status Register, or Interrupt Steering Register

MTBF ----------Mean Time Between Failures.

ns, µs, ms ----------Nanoseconds, microseconds, and milliseconds respectively.

singlewide ----------An IP Module printed circuit board (3.9 inches by 1.8 inches). Each module has two 50-pin connectors.

VHDL. ----------Very high-speed integrated circuit Hardware Description Language.

*This page intentionally left blank*

# INDEX