Force Computers cPCI-690

PowerCore CompactPCI Single-Board Computer

In Stock

Used and in Excellent Condition

Open Web Page

https://www.artisantg.com/60554-1

ARTISAN TECHNOLOGY GROUP

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

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

PPC/CPCI-690

Reference Guide

P / N 227356 Revision AC
June 2006

Copyright

Notice

While reasonable efforts have been made to assure the accuracy of this document, Motorola GmbH assumes no liability resulting from any ommissions in this document, or from the use of the information obtained herein. Motorola reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of Motorola to notify any person of such revision or changes.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to the Motorola Embedded Communications Computing Web site. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Motorola GmbH.

It is possible that this publication may contain reference to or information about Motorola products (machines and programs), programming, or services that are not available in your country. Such references or information must not be construed to mean that Motorola intends to announce such Motorola products, programming, or services in your country.

Contents

2 Installation

3 Controls, Indicators, and Connectors

4 PowerBoot

5 Buses

6 Maps and Registers

Tables

Maps and Registers

Figures

Introduction

Installation

Controls, Indicators, and Connectors

Buses

Battery Exchange

Using This Guide

This Reference Guide is intended for users qualified in electronics or electrical engineering. Users must have a working understanding of Peripheral Component Interconnect (PCI), Compact Peripheral Component Interconnect (CPCI), and telecommunications.

Conventions

Notation	Description
57	All numbers are decimal numbers except when used with the notations described below
00000000_{16}	Typical notation for hexadecimal numbers (digits are 0 through F), e.g. used for addresses and offsets
0000_2	Same for binary numbers (digits are 0 and 1)
x	Generic use of a letter
n	Generic use of numbers
n.nn	Decimal point indicator is signalled
Bold	Character format used to emphasize a word
`Courier`	Character format used for on-screen output
`Courier+Bold`	Character format used to characterize user input
Italics	Character format for references, table, and figure descriptions
<text>	Typical notation used for variables and keys
[text]	Typical notation for buttons and optional parameter in PowerBoot
...	Repeated item
. . .	Omission of information from example/command that is not necessary at the time being
..	Ranges
:	Extents
|	Logical OR

Notation	Description
Note:	No danger encountered. Pay attention to important information marked using this layout
Caution 	Possibly dangerous situation: slight injuries to people or damage to objects possible
Danger 	Dangerous situation: injuries to people or severe damage to objects possible

Abbreviations

ARP	Address Resolution Protocol
BCD	Binary-Coded Decimal
BIB	Board Information Block
BMC	Baseboard Management Controller
CAS	Column Address Select
COP	Common On-Chip Processor
CPCI	Compact Peripheral Component Interconnect
CPU	Central Processing Unit
DRAM	Dynamic Random Access Memory
ECC	Error Checking and Correction
EPROM	Erasable Programmable Read-Only Memory
ESD	Electrostatic Discharge
FIFO	First In First Out
GND	Ground
GPP	General Purpose Pins
I2C	Intelligent Interface Controller
IBMU	Intelligent Board Management Unit
ICMB	Inter Chassis Management Bus
IDE	Integrated Device Electronics

ID-ROM	Identification Read-Only Memory
IPMI	Intelligent Platform Management Interface
LED	Light Emitting Diode
LFM	Linear Feet per Minute
MAC	Media Access Control Layer
MII	Media Independent Interface
MPP	Multi Purpose Pins
NMI	Nonmaskable Interrupt
NVRAM	Nonvolatile Read-Only Memory
PCI	Peripheral Component Interconnect
PHY	Physical Layer
PLD	Programmable Logic Device
PM	Peripheral Management Controller
PMC	PCI Mezzanine Card
PROM	Programmable Read-Only Memory
PSB	Packet Switching Backplane
PTC	Positive Temperature Coefficient
RAM	Random Access Memory
RARP	Reverse Address Resolution Protocol
ROM	Read-Only Memory
RTB	Rear Transition Board
RTC	Real Time Clock
SBC	Single Board Computer
SDR	Sensor Data Record
SDRAM	Synchronous Dynamic Random Access Memory
SELV	Safety Extra Low Voltage
S.M.A.R.T.	Software Maintenance and Repair Tool
SMB	Serial Management Bus
SMI	System Management Interrupt

SRAM	Static RAM
TPE	Twisted-Pair Ethernet
UART	Universal Asynchronous Receiver / Transmitter

Revision History

Order No.	Revision	Date	Description
216049	AA	March 2002	Preliminary Manual
216049	AB	October 2002	New version of Preliminary Manual: Changed manual type from Installation Guide to Reference Guide, modified "Other Sources of Information" page xx, changed CPU frequency value in whole document from 700 to 667 MHz, modified "About this Manual" page 1-3, updated "Order Numbers" page 1-7, updated Table 4 "Power Requirements for Boards with 750FX Processor (667 MHz)" page 2-7, modified Table 6 "Possible On-Board Upgrades and Accessories Combinations" page 2-8, modified Figure 4 "PMC Connectors and Slots" page 2-10, added safety notes to "Switch Settings" page 2-13, changed SW2-4 from reserved to PCI bus 0 reset, removed section "Testing the Board"; removed note from "Front Panel" page 3-3 and modified figure, modified Figure 12 "COM 1 and 2 Connector Pinout" page 3-7, corrected Figure 15 "J5 Connector Pinout Rows A to C" page 3-10, changed Pin D15 from CPCI_IPMB1_CLK to IPMB1_SCL, Pin E 15 from CPCI_IPMB1_DAT to IPMB1_SDA and Pin E 16 from SMB_ALERT to IPMB1_ALERT in Table 16 "J5 Connector Pinout Rows D and E" page 3-10 added "PowerBoot" chapter added "Buses" chapter, added "Maps and Registers" chapter, editorial changes
216049	AC	January 2003	Modified "Other Sources of Information" page xx, added note to "Environmental Requirements" page 2-5,

Order No.	Revision	Date	Description
216049	AC	January 2003	corrected “Power Requirements” page 2-6, corrected note in “Hardware Upgrades and Accessories” page 2-8 and “IDE Devices” page 2-12, changed SW1-4 from Abort Key to Reserved and SW4-4 from EndurX CO 21k system relevant switch to reserved in Table 7 “Switch Settings” page 2-14, removed section “Abort Key” from “Key” page 3-5 and all front panel drawings, changed description of I2C bus 2 in “The device bus is a 32-bit wide multiplexed bus that is used for connecting boot and user flash memory, RTC / NVRAM, IPMI controller and board registers that are implemented in logic.” page 5-14, changed description of Table 21 “Slave Addresses of Devices Attached to IPMI Controller’s I2C Buses” page 5-17, removed description of Table 22 “On-Board I2C Devices” page 5-18, changed RTB BIB to RTB ID-ROM in Table 23 “Off-Board I2C Devices” page 5-18, modified Figure 18 “Connection to I2C Devices” page 5-19, added section “Interrupt Map” page 6-6, added bit 6 to “LED Control Register” page 6-10, removed row from table Table 22 “On-Board I2C Devices” page 5-18, added sentence to “SENTINEL I2C Bus” page 5-19, editorial changes
221400	AA	September 2003	Added IPMI firmware upgrade procedure In “Environmental Requirements”: updated requirements for vibration Updated list of deliverable board variants

Order No.	Revision	Date	Description
222176	AA	March 2004	Added information on 750GX CPU to "About this Manual" page 1-3; updated Table 2 "Ordering Information Excerpt" page 1-7; added power requirements for 750GX boards to "Power Requirements" page 2-6; renamed section "Voltage Key" to "Signaling Level" page 2-9 and modified it; added step regarding PMC module voltage key position to "Installation Procedure" page 2-9; changed "power-on test" and "POT" to "power-on self test" and "POST" throughout the PowerBoot chapter; corrected syntax and/or command description of the following commands: "bt" page 4-15, "cache" page 4-21, "userled" page 4-26, "temp" page 4-34, "config_wr" page 4-36, "eeprom_read" page 4-37, "eeprom_write" page 4-38, "fprog" page 4-39, "netload" page 4-43, "netsave" page 4-45, "bs" page 4-50, "ferase" page 4-52; changed examples 1-3 on page 4-43; changed example screen output in "help" page 4-31; added information on 750GX CPU to "PPC Bus" page 5-6; modified section "Memory Bus" page 5-7; added 750GX CPU to section "Interrupt Map" page 6-6 Editorial changes
223827	AA	July 2004	Replaced IPMI firmware update procedure in "ipmi_flsupd" with reference to *IPMI Firmware for PPC/CPCI-690 and PPC/CPCI-695 Installation Guide*, updated "Other Sources of Information", editorial changes
227356	AA	September 2005	Brought manual to Motorola-style (copyright, logo etc.); updated description of PowerBoot commands netload and netsave.
227356	AB	January 2006	Corrected P/N number on titlepage. It is now: 227356.
227356	AC	June 2006	Corrected board power consumption values in section Installation->Requirements->Power Requirements

Other Sources of Information

For further information refer to the following documents:

Company	www.	Document
AMD	amd.com search for Am79C875	Am79C875 Quad 10/100-TX/FX Ethernet Transceiver
Elpida	elpida.com search for HM5257805B	HM5257805B 512MBit LVTTL SDRAM
Motorola	motorola.com/computers	SENTINEL Rev.1 Reference Guide Only available via SMART
		IPMI Reference Guide Only available via Motorola literature catalog
		IPMI Firmware for PPC/CPCI-690 and PPC/CPCI-695 Installation Guide Only available via Motorola literature catalog
Galileo	marvell.com/products/communication/discovery/GT-64260A.jsp	GT-64260A System Controller for PowerPC Processors
Intel	intel.com search for 28F128J3A and /28F640J3A	28F128J3A/28F640J3A 3 Volt StrataFlash Memory
IBM	ibm.com search for PowerPC 750FX RISC Microprocessor	PowerPC 750FX RISC Microprocessor
PCI Industrial Computer Manufacturers Group	picmg.com	System Management Specification PICMG 2.9
		CompactPCI Specification PICMG 2.0
		Hot Swap Specification PICMG 2.1
		Packet Switching Backplane Specification PICMG 2.16

Company	www.	Document
RAMiX	-	DDC no. Rx-URMH 090 Rev A Hardware Reference Manual PMC233/4/5 & PMC243/4/5 High Capacity Disk Solutions
Samsung	samsungsemi.com search for K4S560832C	K4S560832C 256MBit LVTTL SDRAM
Vitesse	vitesse.com search for VSC215	VSC215 Baseboard Management Controller

Safety Notes

This section provides safety precautions to follow when installing, operating, and maintaining the PPC/CPCI-690.

We intend to provide all necessary information to install and handle the board in this Reference Guide. However, as the product is complex and its usage manifold, we do not guarantee that the given information is complete. If you need additional information, ask your Motorola representative.

The board has been designed to meet the standard industrial safety requirements. It must not be used except in its specific area of office telecommunication industry and industrial control.

Only personnel trained by Motorola or persons qualified in electronics or electrical engineering are authorized to install, maintain, and operate the PPC/CPCI-690. The information given in this manual is meant to complete the knowledge of a specialist and must not be taken as replacement for qualified personnel.

EMC

The board has been tested in a standard Motorola system and found to comply with the limits for a Class A digital device in this system, pursuant to part 15 of the FCC Rules respectively EN 55022 Class A. These limits are designed to provide reasonable protection against harmful interference when the system is operated in a commercial, business or industrial environment.

The board generates and uses radio frequency energy and, if not installed properly and used in accordance with this Reference Guide, may cause harmful interference to radio communications. Operating the system in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

If you use the board without a PMC module, cover empty slots with blind panels to ensure proper EMC shielding. If boards are integrated into open systems, always cover empty slots.

Installation

Electrostatic discharge and incorrect board installation and removal can damage circuits or shorten their life. Therefore:

- **Before installing or removing the board, read “Action Plan” page 2-3.**
- **Touching the board or electronic components in a non-ESD protected environment causes component and board damage.**
  Before touching boards or electronic components, make sure that you are working in an ESD-safe environment.
- **When plugging the board in or removing it, do not press or pull on the front panel but use the handles.**
- **Before installing or removing an additional device or module, read the respective documentation.**
- **Make sure that the board is connected to the CompactPCI backplane via all assembled connectors and that power is available on all power pins.**

Hot Swap

Installing the PPC/CPCI-690 into or removing it from a powered system not supporting hot swap or high availability causes board damage and data loss.
Therefore, only install or remove it from a powered system if the system itself supports hot swap or high availability and if the system documentation explicitly includes appropriate guidelines.

Installing the board under hot-swap conditions in and removing it from system slots causes board damage and data loss.
Therefore, only install or remove the board under hot-swap conditions if the board is to be operated in slots for which hot swap is explicitly permitted by the system documentation (in most hot-swap systems peripheral slots only).

Removing the board from a powered system with IDE devices attached to the board’s primary or secondary IDE interface via the ACC/RTB-602 results in data loss.
Only remove the board from a powered system without IDE devices attached to the board’s primary or secondary IDE interface via the ACC/RTB-602

Removing the board from the backplane while the hot-swap LED is still off causes data loss.
Wait until the blue hot-swap LED is on before removing the board.

Operation

While operating the board ensure that the environmental and power requirements are met.

High humidity and condensation on the surface cause short circuits. Do not operate the product outside the specified environmental limits and do not operate the product below 0°C. Make sure the product is completely dry and there is no moisture on any surface before applying power.

When operating the board in areas of electromagnetic radiation ensure that the board is bolted on the CompactPCI system and the system is shielded by enclosure.

Make sure that contacts and cables of the board cannot be touched while the board is operating.

Replacement/Expansion

Only replace or expand components or system parts with those recommended by Motorola. Otherwise, you are fully responsible for the impact on EMC and the possibly changed functionality of the product.

Check the total power consumption of all components installed (see the technical specification of the respective components). Ensure that any individual output current of any source stays within its acceptable limits (see the technical specification of the respective source).

If the power consumption of the PMC module exceeds 7.5W the board and the PMC module may be damaged. Therefore, make sure that the total max. power consumption at +/–12V, 5V and 3.3V level does not exceed 7.5W (total over all used voltages).

Installing another RTB than the ACC/RTB-602 causes short circuits and damage to the CPU board and the RTB.
Only use the ACC/RTB-602 for CPCI-690 boards.

Switch Settings

Changing the setting of switches marked as 'reserved' causes the board to malfunction.
Do not change the settings of switches marked as 'reserved' for they might carry production-related functions.

Setting/resetting the switches during operation causes board damage. Therefore, check and change switch settings before you install the board.

If you change the boot block flash write protection you may unintentionally overwrite boot flash 2.
Do not change the boot flash write protection unless it is explicitly stated.

Data loss and board malfunction
Disabling the PCI bus 0 reset and resetting the board when data transfer via SENTINEL is performed result in data loss and malfunction of the board.

RJ-45 Connector

The board provides an RJ-45 connector. Connecting the wrong interfaces (e.g. Ethernet and RS-485 or Ethernet and telephone) may damage the board. Therefore:

- **In order to avoid damage to the board make sure you only connect to an Ethernet network.**
- **Clearly mark TPE connectors near your working area as network connectors.**
- **Connect TPE bushing of the system to safety extra low voltages (SELV) circuits only.**
- **Make sure that the length of the electric cable connected to a TPE bushing does not exceed 100 meters.**

If in doubt, ask your system administrator.

PowerBoot

The ferase command erases the whole 32 MByte storage of the 32 MByte user flash device. If you want to erase smaller parts of the user flash device, use the offset and length parameters described in “ferase” page 4-52.

SENTINEL may not work if configuration has been changed.
Only change the configuration if you are familiar with the consequences. If after a change the board does not work, reboot the board.

The I2C EEPROM contains data for ID ROM and IPMI ID ROM. Overwriting this data results in a malfunction of the board. Do not overwrite the data for ID ROM and IPMI ID ROM.

Updating the IPMI flash with wrong data will damage the IPMI controller. Only update the IPMI flash with data provided by Motorola.

Battery

Wrong battery installation may result in a hazardous explosion and board damage.

Using the battery longer than ten years results in data loss. Therefore, exchange the battery before ten years of actual battery use have elapsed.

Exchanging the battery always results in data loss of the devices which use the battery as power backup. Therefore, backup affected data before exchanging the battery.

Always use the same type of lithium battery as is already installed.

Environment

Always dispose of old boards and batteries according to your country’s legislation, if possible in an environmentally acceptable way.

Sicherheitshinweise

Dieser Abschnitt enthält Sicherheitshinweise, die bei Einbau, Betrieb und Wartung des PPC/CPCI-690 zu beachten sind.

Wir sind darauf bedacht, alle notwendigen Informationen zum Einbau und zum Umgang mit dem Board in diesem Handbuch bereit zu stellen. Da es sich bei dem PPC/CPCI-690 um ein komplexes Produkt mit vielfältigen Einsatzmöglichkeiten handelt, wird die Vollständigkeit der im Handbuch enthaltenen Informationen nicht garantiert. Falls Sie weitere Informationen benötigen sollten, wenden Sie sich bitte an die für Sie zuständige Geschäftstelle von Motorola.

Das Board erfüllt die für die Industrie geforderten Sicherheitsvorschriften und darf ausschließlich für Anwendungen in der Telekommunikationsindustrie und im Zusammenhang mit Industriesteuerungen verwendet werden.

Einbau, Wartung und Betrieb dürfen nur von durch Motorola ausgebildetem oder im Bereich Elektronik oder Elektrotechnik qualifiziertem Personal durchgeführt werden. Die in diesem Handbuch enthaltenen Informationen dienen ausschließlich dazu, das Wissen von Fachpersonal zu ergänzen, können es aber in keinem Fall ersetzen.

EMV

Das Board wurde in einem Motorola Standardsystem getestet. Es erfüllt die für digitale Geräte der Klasse A gültigen Grenzwerte in einem solchen System gemäß den FCC-Richtlinien Abschnitt 15 bzw. EN 55022 Klasse A. Diese Grenzwerte sollen einen angemessenen Schutz vor Störstrahlung beim Betrieb des Boards in Gewerbe- sowie Industriegebieten gewährleisten.

Das Board arbeitet im Hochfrequenzbereich und erzeugt Störstrahlung. Bei unsachgemäßem Einbau und anderem als in diesem Handbuch beschriebenen Betrieb können Störungen im Hochfrequenzbereich auftreten.

Warnung! Dies ist eine Einrichtung der Klasse A. Diese Einrichtung kann im Wohnbereich Funkstörungen verursachen. In diesem Fall kann vom Betreiber verlangt werden, angemessene Maßnahmen durchzuführen.

Wenn Sie das Board ohne PMC Modul verwenden, schirmen Sie freie Steckplätze mit einer Blende ab, um einen ausreichenden EMV Schutz zu gewährleisten. Wenn Sie Boards in Systeme einbauen, schirmen Sie freie Steckplätze mit einer Blende ab.

Installation

Elektrostatische Entladung und unsachgemäßer Ein- und Ausbau des Boards kann Schaltkreise beschädigen oder ihre Lebensdauer verkürzen. Beachten Sie deshalb die folgenden Punkte:

- **Lesen Sie vor Ein- oder Ausbau des Boards den Abschnitt "Action Plan" auf Seite 2-3.**
- **Berühren Sie das Board oder elektronische Komponenten in einem nicht ESD-geschützten Bereich, kann dies zu einer Beschädigung des Boards führen.**
  Bevor Sie Boards oder elektronische Komponenten berühren, vergewissern Sie sich, dass Sie in einem ESD-geschützten Bereich arbeiten.
- **Drücken bzw. ziehen Sie bei Ein- oder Ausbau des Boards nicht auf die Frontblende, sondern benutzen Sie die Griffe.**
- **Lesen Sie vor dem Ein- oder Ausbau von zusätzlichen Geräten oder Modulen das dazugehörige Benutzerhandbuch.**
- **Vergewissern Sie sich, dass das Board über alle Stecker an die CompactPCI Backplane angeschlossen ist und Spannung an allen Versorgungskontakten anliegt.**

Hot Swap

Wenn Sie das Board im laufenden Betrieb in ein System, das weder Hot Swap noch High Availability unterstützt, installieren bzw. herausziehen, wird das Board beschädigt und es gehen Daten verloren. Installieren/entfernen Sie das Board nur im laufenden Betrieb, wenn das System Hot Swap oder High-Availability unterstützt und wenn die Systembeschreibung dies ausdrücklich erlaubt.

Installieren Sie das Board im laufenden Betrieb in einen Systemsteckplatz oder ziehen sie es aus einem Systemsteckplatz heraus, führt dies zu Datenverlusten und zu einer Beschädigung des Boards. Installieren Sie das Board deshalb nur im laufenden Betrieb in

Steckplätze und entfernen Sie Boards nur aus Systemsteckplätzen, die laut Systembeschreibung ausdrücklich hot-swap fähig sind (in den meisten Hot Swap Systemen sind dies periphäre Steckplätze).

Entfernen Sie das Board mit über das RTB angeschlossenen IDE-Festplatten aus einem laufenden System, führt dies zu Datenverlust. Entfernen Sie das Board nur, wenn keine IDE-Festplatten über das RTB angeschlossen sind.

Ziehen Sie das Board im laufenden Betrieb heraus, obwohl die Hot-Swap LED noch nicht leuchtet, führt das zu Datenverlust. Warten Sie deshalb bis die Hot-Swap LED blau leuchtet, bevor Sie das Board herausziehen.

Betrieb

Achten Sie darauf, dass die Umgebungs- und die Leistungsanforderungen während des Betriebs eingehalten werden.

Durch hohe Luftfeuchtigkeit und Kondensat auf der Boardoberfläche können Kurzschlüsse entstehen. Betreiben Sie das Board nur innerhalb der angegebenen Grenzwerte für die relative Luftfeuchtigkeit und Temperatur. Stellen Sie vor dem Einschalten des Stroms sicher, dass sich auf dem Board kein Kondensat befindet.

Wenn Sie das Board in einer Umgebung mit elektromagnetischer Strahlung betreiben, stellen Sie sicher, dass das Board mit dem CompactPCI System verschraubt ist und das System durch ein Gehäuse abgeschirmt wird.

Stellen Sie sicher, dass Anschlüsse und Kabel des Boards während des Betriebs nicht berührt werden können.

Austausch/Erweiterung

Verwenden Sie bei Austausch oder Erweiterung nur von Motorola empfohlene Komponenten und Systemteile. Andernfalls sind Sie für mögliche Auswirkungen auf EMV und geänderte Funktionalität des Produktes voll verantwortlich.

Überprüfen Sie die gesamte aufgenommene Leistung aller eingebauten Komponenten (siehe die technischen Daten der entsprechenden Komponente). Stellen Sie sicher, dass die Stromaufnahme jedes Verbrauchers innerhalb der zulässigen Grenzwerte liegt (siehe die technischen Daten des entsprechenden Verbrauchers).

Übersteigt der Gesamtstromverbrauch pro PMC Modul 7,5W, können das Modul und das Board beschädigt werden.
Stellen Sie sicher, dass der Gesamtstromverbrauch pro PMC Modul bei +/- 12V, 5V und 3,3V 7,5W nicht übersteigt (Summe aller Spannungen).

Verwenden Sie ein anderes RTB als das ACC/RTB-602, kann dies zu Kurzschlüssen und yu einer Beschäding sowohl des Boards als auch des RTBs führen.
Verwenden Sie für CPCI-690 board nur das ACC/RTB-602.

Schaltereinstellungen

Das Ändern der mit “reserved” gekennzeichneten Schalter kann zu Störungen im Betrieb des Boards führen.
Ändern Sie Schaltereinstellungen der mit “reserved” gekennzeichneten Schaltern nicht, da diese Schalter mit produktionsrelevanten Funktionen belegt sein können, die im normalen Betrieb Störungen auslösen könnten.

Ein Ändern der Schaltereinstellungen während des laufenden Betriebs kann das Board beschädigen.
Prüfen und ändern Sie die Schaltereinstellungen, bevor Sie das Board installieren.

Wenn Sie den Schreibschutz des Boot Flashes aufheben, ist es möglich, dass Sie Boot Flash 2 unabsichtlich überschreiben.
Um unabsichtliches Überschreiben von Boot Flash 2 zu verhindern, heben Sie den Schreibschutz des Boot Flashes nur dann auf, wenn es ausdrücklich in der Dokumentation gefordert wird.

Wenn Sie den PCI bus 0 Reset ausschalten, führt dies beim Zurücksetzen des Boards und gleichzeitig ausgeführten Datentransfers über SENTINEL zu einem Datenverlust des SENTINEL und zu einer Fehlfunktion des Boards.
Betreiben Sie das Board deshalb nur mit der Schaltereinstellung SW2-4: OFF.

RJ-45 Stecker

Das Board stellt eine RJ-45 Ethernet Verbindung zur Verfügung. Werden falsche Schnittstellen miteinander verbunden (z.B. Ethernet und RS-485 oder Ethernet und Telefon), kann das Board beschädigt werden. Daher:

- **Um Schäden zu vermeiden, verbinden Sie ausschließlich Ethernet Schnittstellen.**
- **Kennzeichnen Sie TPE-Anschlüsse in der Nähe Ihres Arbeitsplatzes deutlich als Netzwerkanschlüsse.**
- **Schliessen Sie an TPE-Buchsen ausschließlich SELV-Kreise (Sicherheitskleinspannungsstromkreise) an.**
- **Stellen Sie sicher, dass die Länge der an einer TPE-Buchse angeschlossenen Leitung nicht mehr als 100 Meter beträgt.**

Falls Sie Fragen haben, wenden Sie sich bitte an Ihren Systemadministrator.

PowerBoot

Der Befehl ferase löscht den gesamten Speicher des 32 MByte User Flashs. Wenn Sie kleinere Teile des User Flashs löschen möchten, verwenden Sie zusätzlich die Offset und Length Parameter, die im Abschnitt “ferase” auf Seite 4-52 beschrieben werden.

Ändern Sie die Konfiguration, ist es möglich, dass SENTINEL nicht mehr funktioniert. Ändern Sie die Konfiguration nur, wenn Sie sich über die Konsequenzen einer Konfigurationsänderung im klaren sind. Hat das Board nach einer Änderung eine Fehlfunktion, booten Sie es erneut.

Das I2C EEPROM enthält ID ROM und IPMI ID ROM relevante Daten. Überschreiben Sie diese Daten, führt dies zu einer Fehlfunktion des Boards. Überschreiben Sie deshalb keine ID ROM und IPMI ID ROM relevanten Daten.

Aktualisieren Sie das IPMI Flash mit falschen Daten, führt dies zur Beschädigung des IPMI Controllers. Aktualisieren Sie das IPMI Flash nur mit Daten, die Sie von Motorola bekommen.

Batterie

Fehlerhafter Austausch von Lithiumbatterien kann zu lebensgefährlichen Explosionen führen.

Verwenden Sie die Batterien länger als zehn Jahre, kann dies zu Datenverlusten führen. Tauschen Sie deshalb die Batterie aus, bevor zehn Jahre reiner Betrieb vorüber sind.

Der Austausch der Batterie bringt immer einen Datenverlust bei den Komponenten mit sich, die sich durch die Batterie die Stromversorgung sichern. Sichern Sie deshalb vor dem Batterieaustausch Ihre Daten.

Verwenden Sie beim Batteriewechsel immer den selben Batterietyp, der bereits eingesetzt wurde.

Umweltschutz

Entsorgen Sie alte Boards und Batterien gemäß der in Ihrem Land gültigen Gesetzgebung, wenn möglich immer umweltfreundlich.

1

Introduction

About this Manual

This reference Guide provides the information that you need to install, access and operate the board.

Organization of this Manual

The Reference Guide is organized as follows.

Chapter	Description
Using This Guide	Lists all conventions and abbreviations used in this manual and outlines the revision history
Other Sources of Information	Lists related documentation and specifications
Safety Notes	Provides safety relevant information when handling the board
Sicherheitshinweise	German translation of the Safety Notes chapter
Introduction	Provides a basic overview of the features of the board and this manual
Installation	Outlines the installation requirements, hardware accessories, switch settings, installation and removal procedures
Controls, Indicators and Connectors	Describes the LEDs, keys, and connectors of the board
PowerBoot	Describes the main features of the board's PowerBoot
Maps and Registers	Provides information that is relevant for programmer's. This includes registers, interrupt and memory maps
Device' s Features and Data Paths	Provides detailled information on the devices, such as controllers, CPU etc., used on the board and how they are interconnected

Chapter	Description
Battery Exchange (Appendix)	Describes how to exchange the on-board battery
Troubleshooting (Appendix)	Describes how to deal with problems related to the operation of the board

Feedback

Motorola welcomes and appreciates your comments on its documentation. We want to know what you think about our manuals and how we can make them better. Mail comments to:

- Motorola GmbH
  ECC Embedded Communication Computing
  Lilienthalstr. 15
  85579 Neubiberg-Munich / Germany
- reader-comments@mcg.mot.com

Features

The PPC/CPCI-690 is a single-slot 6U universal CompactPCI hot-swap board using Motorola' SENTINEL PCI-to-CompactPCI universal bridge technology. As it is based on an IBM 750FX or 750GX PowerPC processor of up to 1 GHz, the board offers high performance. Combined with full hot-swap capabilities it is an ideal component for multi-node processor systems.

Other important features of the CPCI-690 are:

- Three 10/100MBit Ethernet ports
- Up to 2 GByte on-board SDRAM
- Two 64-bit/66-MHz PMC interfaces
- CompactFlash and hard disk support
- Non-monarch mode ProcessorPMC support
- Packet switching backplane (PSB) support
- System management support via IPMI (Intelligent Platform Management Interface)

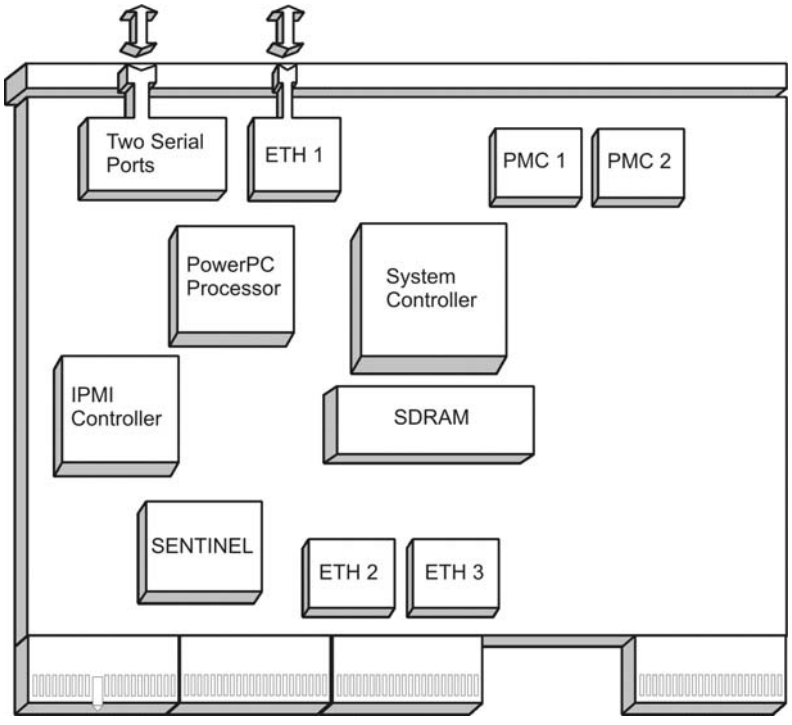


Figure 1: *Function Blocks*

Standard Compliance

The PPC/CPCI-690 meets the following standards:

Table 1: *Standard Compliance*

Standard	Description
EN 60950 UL 60950 UL 94V-0/1 (predefined Motorola system)	Legal safety requirements
EN 55022 EN 55024 FCC Part 15 Class A	EMC requirements on system level
ANSI/IPC-A610 Rev. C Class 2 IPC-7711 and 7721 ANSI-J-001...003	Manufacturing requirements

Ordering Information

When ordering the PPC/CPCI-690 board variants, upgrades and accessories, use the order numbers given below.

Product Nomenclature

In the following you find the key for the product name extensions.

PPC/CPCI-690/xxx-ccc-zz/PSB	
xxx	SDRAM size in MByte
ccc	Processor clock frequency in MHz
zz	User flash capacity in MByte
PSB	Packet switching backplane variant

Order Numbers

Depending on the PPC/CPCI-690 type, the available upgrades and accessories may differ. Consult your local sales representative to check the possibility of combinations.

Table 2: *Ordering Information Excerpt*

Order No.	PPC/CPCI-690	Description
110190	.../256-700-32	IBM 750FX PowerPC processor, 667 MHz with 256 MByte main memory, user flash 32 MByte
110192	.../512-700-32	IBM 750FX PowerPC processor, 667 MHz with 512 MByte main memory, user flash 32 MByte
110193	.../1024-700-32	IBM 750FX PowerPC processor, 667 MHz with 1024 MByte main memory, user flash 32 MByte
110194	.../256-700-32/PSB	IBM 750FX PowerPC processor, 667 MHz with 256 MByte main memory, user flash 32 MByte, for PICMG 2.16 Packet Switching Backplane systems

Table 2: *Ordering Information Excerpt (cont.)*

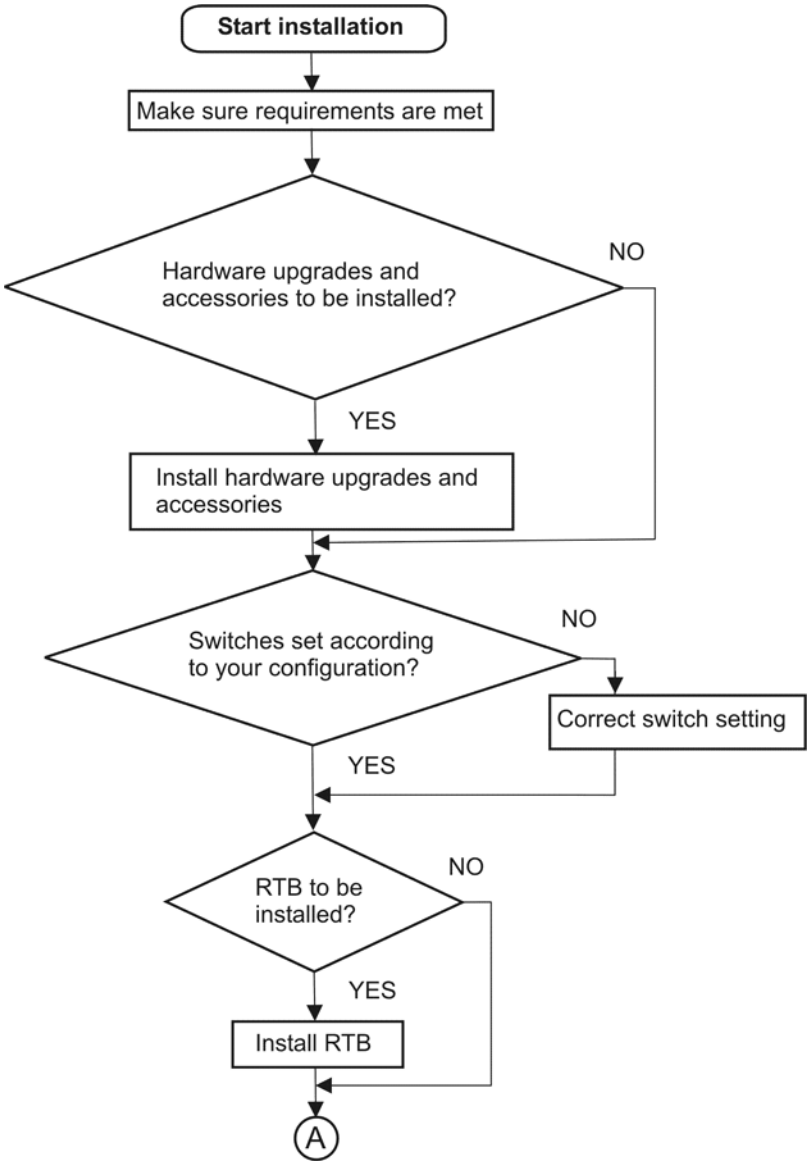
Order No.	PPC/CPCI-690	Description
120922	.../512-700-32/PSB	IBM 750FX PowerPC processor, 667 MHz with 512 MByte main memory, user flash 32 MByte, for PICMG 2.16 Packet Switching Backplane systems
120911	.../1024-700-32/PSB	IBM 750FX PowerPC processor, 667 MHz with 1024 MByte main memory, user flash 32 MByte, for PICMG 2.16 Packet Switching Backplane systems
121077	.../256-1000GX-32	IBM 750GX PowerPC processor, 1 GHz with 256 MByte main memory, user flash 32 MByte
121078	.../512-1000GX-32	IBM 750GX PowerPC processor, 1 GHz with 512 MByte main memory, user flash 32 MByte
121222	.../512-1000GX-32/PSB	IBM 750GX PowerPC processor, 1 GHz with 512 MByte main memory, user flash 32 MByte, for PICMG 2.16 Packet Switching Backplane systems
Hardware Accessories PPC/CPCI-690		
110196	ACC/RTB-602	Rear transition board and Installation Guide
110197	ACC/RTB-602/PSB	Rear transition board and Installation Guide for PICMG 2.16 Packet Switching Backplane systems
111170	PMC-243D	Hard disk on PMC carrier, without secondary IDE interface routed to backplane
111173	PMC-244FP	CompactFlash disk on PMC carrier, with front-panel access and secondary IDE interface routed to backplane
111136	PPC/COP-ADAPTER-69X	Adapter to connect a standard COP/JTAG debugger or emulator

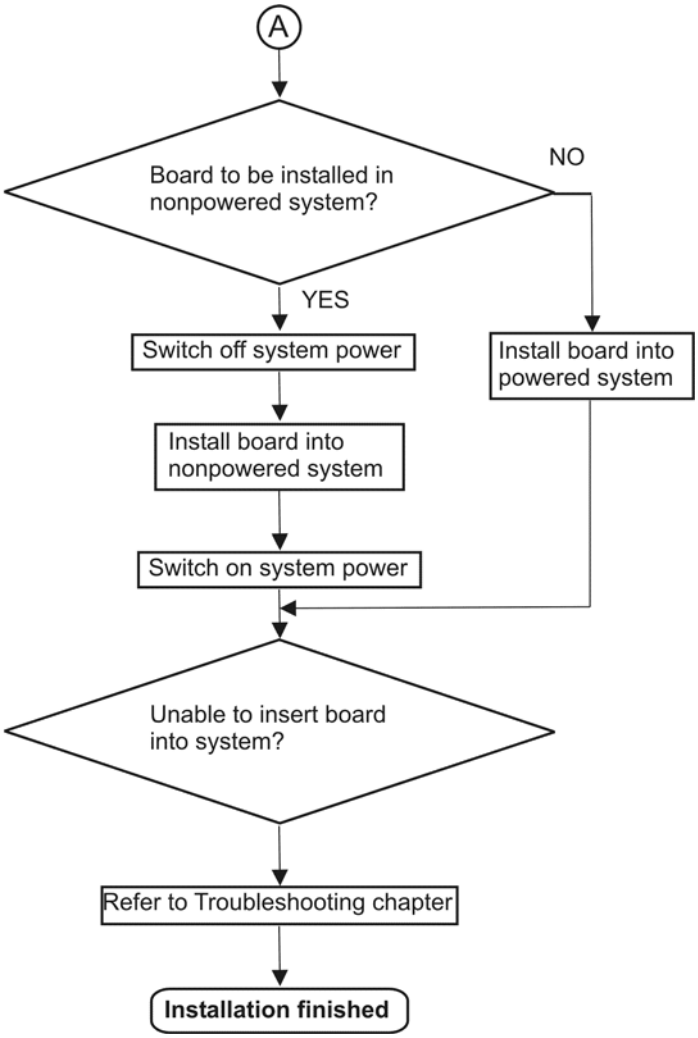
2

Installation

Action Plan

To install the board, the following steps are necessary and described in detail in the sections of this chapter.




A
Board to be installed in nonpowered system?
NO
YES
Switch off system power
Install board into powered system
Install board into nonpowered system
Switch on system power
Unable to insert board into system?
Refer to Troubleshooting chapter
Installation finished

Requirements

To meet the environmental requirements, the board has to be tested in the system where it is to be installed. Before you power up the board, calculate the power needed according to your combination of board upgrades and accessories.

Environmental Requirements

The environmental values must be tested and proven in the used system configuration. The conditions listed below refer to the surroundings of the board within the user environment.

Note:

- **Operating temperatures refer to the temperature of the air circulating around the board and not to the component temperature.**
- **In order to meet the operating conditions, forced air cooling is required at both sides of the board.**
- **The environmental values given in the table below only apply to the board without any accessories. If installing accessories, their environmental requirements must also be taken into account. If you use the board together with an PMC-243D or PMC-245FD, make sure the environmental values given in the *Hardware Reference Manual PMC233/4/5 & PMC243/4/5 High Capacity Disk Solutions* are met.**

Caution

Board damage
High humidity and condensation on the surface cause short circuits. Do not operate the product outside the specified environmental limits and do not operate the product below 0°C. Make sure the product is completely dry and there is no moisture on any surface before applying power.

Table 3: *Environmental Requirements*

Feature	Operating	Non-Operating
Temperature	0°C to +55°C (may be further limited by hard disk)	–40°C to +85°C
Forced air flow	300 LFM	–
Temp. change	+/– 0.5°C / min	+/– 1°C / min
Rel. humidity	5% to 95% non-condensing at +40°C	5% to 95% non-condensing at +40°C
Altitude	–300 m to +3,000 m	–300 m to +13,000 m
Vibration 5 - 500 Hz 20 - 2000 Hz	 2 g sinusodial -	 - 3 g random
Shock	5 g / 11 ms halfsine	15 g / 11 ms halfsine
Free fall	100 mm / 3 axis	1,200 mm / all edges and corners (packed state)

Power Requirements

The board's power requirements depend on the installed hardware accessories. The following tables give examples of typical power requirements for the 750FX and the 750GX board variants without any accessories. If you want to install accessories, the load of the respective accessory has to be added to the load of the used board variant.

For information on the accessories' power requirements, refer to the documentation coming with the respective accessory or ask your local Motorola representative.

The following table gives the typical power requirements for a board with:

- 750FX processor running at 667 MHz
- 1 GByte SDRAM
- No accessories

Table 4: *Power Requirements for Boards with 750FX Processor (667 MHz)*

Requirement	+3.3V	+5V	VIO[1)]	
			3.3V	5V
Maximum power dissipation	9.6W	7.5W	1.32W	2.0W
Min. Voltage	3.20V	4.85V	3.20V	4.85V
Max. Voltage	3.45V	5.25V	3.45V	5.25V
Max. Current	2.9A	1.5A	0.4A	0.4A

1) If 3.3V or 5V are used depends on used system

The table below gives the typical power requirements for a board with:

- 750GX processor running at 1 GHz
- 512 MByte SDRAM
- No accessories

Table 5: *Power Requirements for Boards with 750GX Processor (1 GHz)*

Requirement	+3.3V	+5V	VIO[1)]	
			3.3V	5V
Maximum power dissipation	7.92W	17.6W	1.32W	2.0W
Min. Voltage	3.20V	4.85V	3.20V	4.85V
Max. Voltage	3.45V	5.25V	3.45V	5.25V
Typ. Current	1.7A	2.86A		
Max. Current	2.4A	3.52A	0.4A	0.4A

1) If 3.3V or 5V are used depends on used system

The board provides inrush current protection which allows the board to be installed into the backplane of a powered system. The voltages are current protected, i.e. all board voltages will be turned off if the current of the 5V or 3.3V system environment exceeds the limits stated above. The +/-12V of the PMC slots are passively protected with positive temperature coefficient (PTC) fuses.

Hardware Upgrades and Accessories

The PPC/CPCI-690 itself allows an easy and cost-efficient way to adapt the system board to your application needs.

The following table gives an overview on the possible product combinations, described in Table 2 “Ordering Information Excerpt” page 1-7.

Note: If you use the PMC-244FD and if you want to use the IDE interface via the RTB, the PMC module has to be installed into PMC slot 1.

Table 6: *Possible On-Board Upgrades and Accessories Combinations*

Upgrade/Accessory	PMC Slot
1 PMC module	1 or 2
2 PMC modules	1 and 2
1 PMC module, 1 Hard disk	1 and 2, respectively
1 PMC module, 1 CompactFlash disk	1 and 2, respectively
1 Hard disk, 1 CompactFlash disk	1 and 2, respectively

PMC Module

The board provides two PMC slots supporting a 64-bit data bus width with a maximum frequency of 66 MHz. Both PMC slots provide +/–12V and support non-monarch mode Processor PMC modules.

The slots can be used to install a PMC module with front panel interfaces and rear interfaces (see “Connectors” page 3-7). The user I/O signals of slot 1 are routed to the J3 connector, the user I/O signals of slot 2 are routed to the J5 connector enabling access to these pins on the RTB.

Note:

- **To ensure proper EMC shielding, either operate the board with the blind panels for the PPC/CPCI-690 front panel or with the PMC modules installed.**
- **If the board is upgraded, ensure that the blind panels are stored in a safe place in order to be used again when removing the PMC modules.**

Signaling Level

The PMC slots provide a 3.3V PCI interface, therefore, only PMC modules which support a V I/O of 3.3V can be installed. Since 5V PMC modules do not have drill holes at the 3.3 V voltage key position, the voltage keys prevent these modules from being installed into the PMC slots.

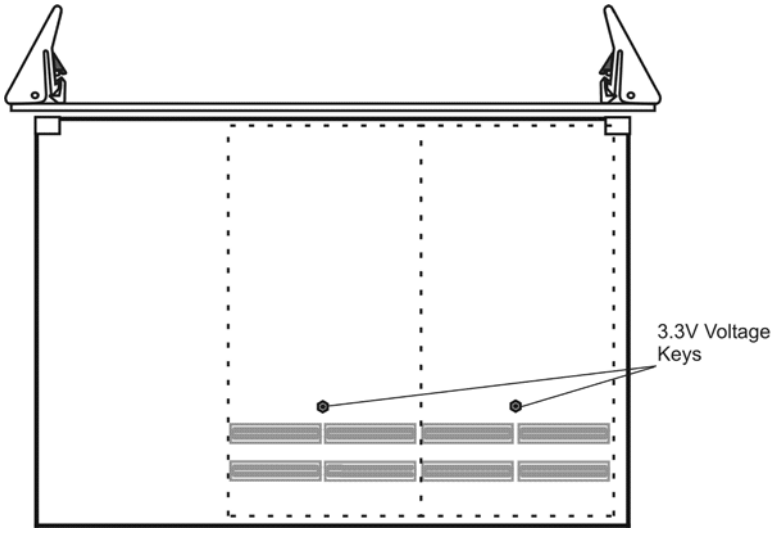


Figure 2: *Voltage Keys*

Installation Procedure

1. Remove blind panel of respective PMC slot from front panel
2. Store blind panel in safe place
3. Check that PMC module has a drill hole at 3.3V position
   If not, the PMC module cannot be installed.

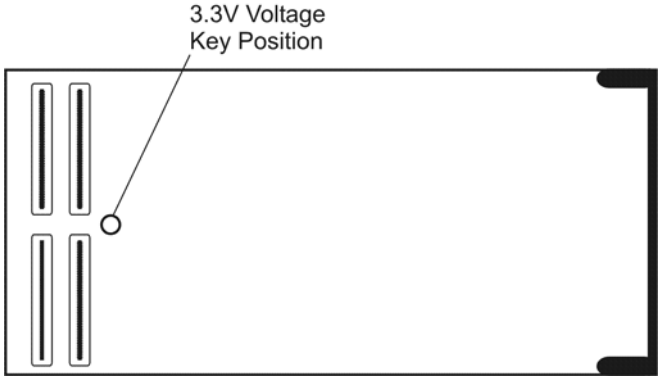


Figure 3: *Location of 3.3V Voltage Key on PMC Module*

Caution

PMC module and board damage
If the power consumption of the PMC module exceeds 7.5W, the board and the PMC module may be damaged.
Therefore, make sure that the total max. power consumption at +/–12V, 5V and 3.3V level does not exceed 7.5W (total over all used voltages).

4. Plug PMC module into connectors of PMC slot 1 (PN11, 12, 13, 14) or PMC slot 2 (PN21, 22, 23 and 24)

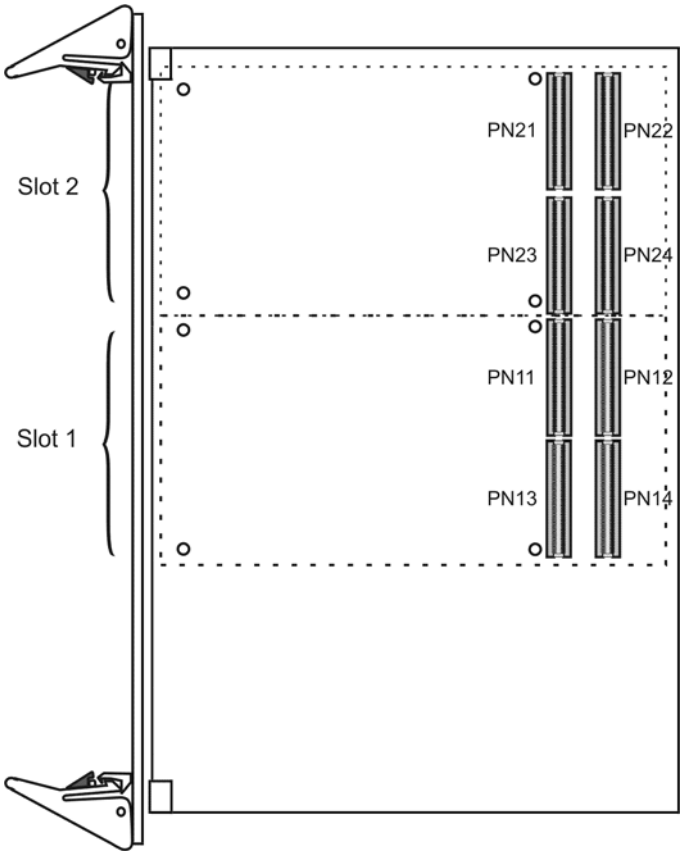


Figure 4: *PMC Connectors and Slots*

5. Check whether standoffs of module cover mounting holes of board

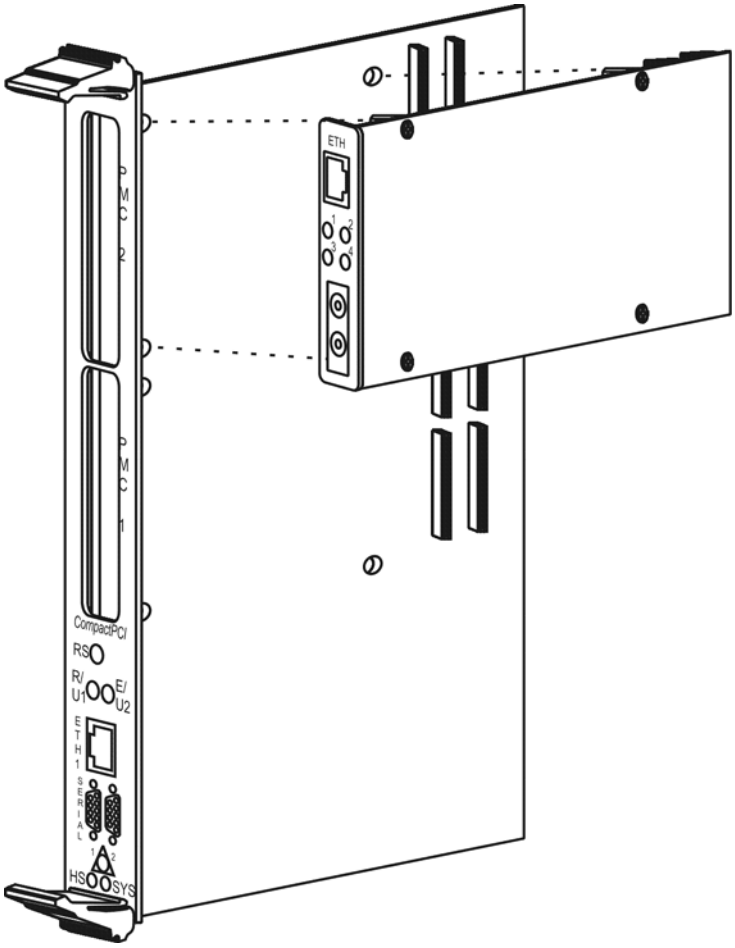


Figure 5: *Position of Mounting Holes*

6. Place screws delivered with PMC module from bottom side into mounting holes
7. Fasten screws

Removal Procedure

1. Remove screws
2. Disconnect PMC module from slot
3. Close front panel gap at free slot with blind panel

IDE Devices

It is possible to have IDE devices on the board which are connected to the local PCI bus. A hard disk, a CompactFlash disk or both can be installed. The devices are fixed on a carrier PMC module which also includes an IDE controller. Hard disk or CompactFlash disk are connected to the primary IDE interface of the controller. The PMC module with the IDE device has to be installed into one of the PMC slots.

For the available carrier PMC modules with IDE devices, refer to "Order Numbers" page 1-7. For installation of the PMC modules, refer to "PMC Module" page 2-8.

Further devices can be connected to the secondary IDE interface via a rear transition board if a PMC-244FP is plugged in PMC slot 1.

Note: If you want to connect further IDE devices via RTB, install the PMC-244FP module into PMC slot 1 since only PMC slot 1 has routed the secondary IDE interface of the IDE controller to the backplane. If you install it into PMC slot 2 the IDE devices on the secondary IDE interface do not work. For further information, refer to the *ACC/RTB-602 Installation Guide*.

Rear Transition Board

As a separate price list item, Motorola offers the ACC/RTB-602. The RTB provides access to the board's CompactPCI user I/O interfaces via industry standard connectors. For further information, refer to the *ACC/RTB-602 Installation Guide.*

Caution

Damage to board and RTB
Installing another RTB than the ACC/RTB-602 causes short circuits and damage to the CPU board and the RTB.
Only use the ACC/RTB-602 for CPCI-690 boards.

Switch Settings

The board provides three configuration switches, SW1, SW2, and SW4.

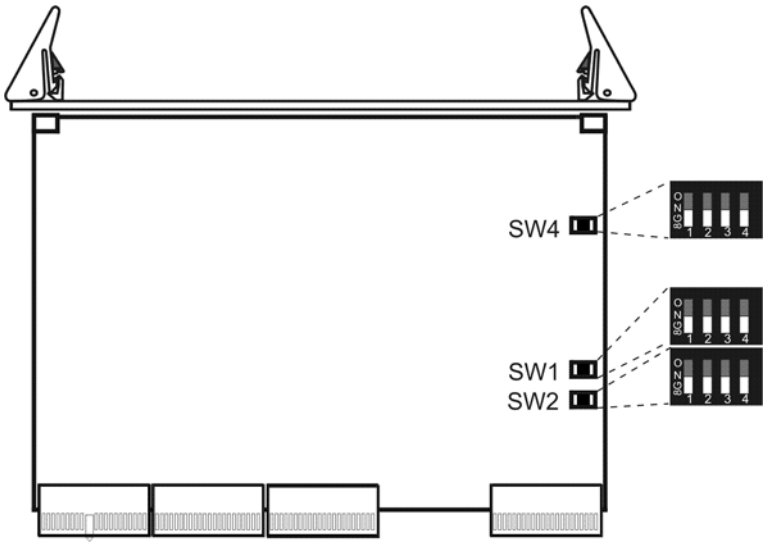


Figure 6: *Location of Switches*

For default settings, the white switches are moved to the OFF position.

Caution

- **Board malfunction**
  Changing the setting of switches marked as 'reserved' causes the board to malfunction.
  Do not change the settings of switches marked as 'reserved' for they might carry production-related functions.
- **Boot block damage**
  If you change the boot flash write protection you may unintentionally overwrite boot flash 2.
  Only change the boot flash write protection if it is explicitly stated.
- **Board damage**
  Setting/resetting the switches during operation causes board damage. Therefore, check and change switch settings before you install the board.
- **Data loss and board malfunction**
  Disabling the PCI bus 0 reset and resetting the board when data transfer via SENTINEL is performed result in data loss and malfunction of the board.
  Only operate the board with SW2-4 in its default position (OFF).

Table 7: *Switch Settings*

Switch	Number	Description
SW1		
	1	IPMI FORCE_PM OFF: Automatic PM/BMC detection (default)[1)] ON: IPMI controller is PM
	2	FORCE IPMI SYSEN OFF: The IPMI controller senses the CPCI_SYSEN (default) ON: The IPMI SYSEN is active
	3	Front panel reset OFF: Reset key enabled (default) ON: Reset key disabled
	4	Reserved OFF: Default
SW2		
	1	Boot flash selection OFF: Booting from boot flash 1 (default) ON: Booting from boot flash 2
	2	Boot flash write enable OFF: Write disabled (default) ON: Write enabled
	3	User flash write protection OFF: Protection disabled (default) ON: Protection enabled
	4	PCI bus 0 reset OFF: PCI bus 0 reset enabled (default) ON: PCI bus 0 reset disabled

Table 7: *Switch Settings*

Switch	Number	Description
SW4		
	1	Reserved OFF: Default
	2	Reserved OFF: Default
	3	Reserved OFF: Default
	4	Reserved OFF: Default

1) The IPMI controller is BMC (SYSEN active) or PM (SYSEN inactive)

Board Installation

The board's front panel provides a compatibility glyph indicating whether you use a CompactPCI standard or a PSB board variant:

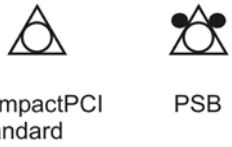


Figure 7: *Compatibility Glyphs for CompactPCI Standard and PSB Board Variants*

The standard CompactPCI board can be used either as system controller in a system slot or as an intelligent I/O board in a peripheral slot. The PSB variant must be installed into a node slot.

Both board variants provide hot-swap support, i.e. they may be installed in or removed from a powered system. This section is divided into two sub-sections for installing the board in a nonpowered system and in a powered system supporting hot swap.

Note: EN 55022 Class A and FCC Part 15 Class A are achieved by using single-point grounding. If you ground the CPCI-690 at multiple points EMC problems may occur.

Installation in a Nonpowered System

As a universal hot-swap board, the CPCI-690 can be installed in both 3.3V and 5V systems.

Note: Before installing the board install the upgrades and accessories, if necessary (refer to "Hardware Upgrades and Accessories" page 2-8).

Installation Procedure

1. Turn off system power

Caution

Board damage
Touching the board or electronic components in a non-ESD protected environment causes component and board damage.
Before touching boards or electronic components, make sure that you are working in an ESD-safe environment.

2. Check switch settings for consistency (see "Switch Settings" page 2-13)
3. Open handles
4. **PSB variant**: Install board into node slot
   Standard CompactPCI variant: Install board depending on intended function either in system slot marked with a triangle or in peripheral slot marked with a circle
5. Press handles inwards to lock board on rack frame
6. Fasten board with screws
7. Connect interface cables to front panel connectors, if applicable
8. Turn on system power

Removal Procedure

Caution

Board damage
Touching the board or electronic components in a non-ESD protected environment causes component and board damage.
Before touching boards or electronic components, make sure that you are working in an ESD-safe environment.

1. Turn off system power
2. Unfasten screws of front panel until board is detached from rack frame
3. Press button on handle
4. Press handles outwards to disconnect board from backplane
5. Remove board from rails of slot position

Installation in a Powered System Supporting Hot Swap

The board supports full hot swap and high availability. The basic purpose of hot-swap support and high availability is to allow the board to be installed in and removed from a powered system without adversely affecting system operation. With high availability support, defective boards can be repaired and systems can be reconfigured without stopping system operation and with minimum operator interaction.

Caution

Board damage and data loss
Installing the board into or removing it from a powered system not supporting hot swap or high availability causes board damage and data loss. Therefore, only install or remove it from a powered system if the system itself supports hot swap or high availability and if the system documentation explicitly includes appropriate guidelines.

As a universal hot-swap board, the CPCI-690 can be installed in both 3.3V and 5V systems.

Man-Machine Interface

The board provides the following elements as man-machine interface:

- Hot-swap switch
  Integrated in the lower handle of the front panel, it allows the user to indicate the intention to remove the board to the system controller.
- Blue hot-swap LED
  If ON, it indicates that you are allowed to remove the board.

Signals

For information on board status concerning the hot-swap conditions, the board provides ENUM# and HEALTHY# signals for hot-swap-controller (HSC) notification and SENTINEL's Control and Status registers.

After installing the board into or before removing it from a powered system, the interrupt ENUM# signal is generated and passed to the system board of the CompactPCI system to indicate a service request. The Control and Status registers allow to determine the source of the ENUM# signal and to control the hot-swap LED. The HEALTHY# signal is active whenever all backplane and on-board voltages are within their operating range.

Installation Procedure

Caution

- **Board damage and data loss**
  Installing the board under hot-swap conditions in system slots causes board damage and data loss.
  Therefore, only install the board under hot-swap conditions if the board is to be operated in slots for which hot swap is explicitly permitted by the system documentation (in most hot-swap systems peripheral slots only).
- **Board damage**
  Touching the board or electronic components in a non-ESD protected environment causes component and board damage.
  Before touching boards or electronic components, make sure that you are working in an ESD-safe environment.

The installation procedure depends upon the system the board is to be installed in.

Into Basic Hot-Swap System

1. Check board configuration, e.g. switch settings
2. Check that you are using an ACC / RTB-602, if applicable
3. **PSB variant**: Install board into node slot
   Standard CompactPCI variant: Install board into peripheral slot of powered system
4. Close handles
5. Fasten board with screws
6. Connect software manually according to system documentation

Into Full Hot-Swap or High-Availability System

1. Check board configuration, e.g. switch settings
2. Check that you are using an ACC / RTB-602, if applicable
3. **PSB variant**: Install board into node slot
   Standard CompactPCI variant: Install board into peripheral slot of powered system
   The hot-swap LED stays blue until the board software connection process has been completed.
4. Close handles
5. Fasten board with screws

Removal Procedure

Before removing the board, observe the following:

Caution

- **Board damage and data loss**
  Removing the board under hot-swap conditions from system slots, causes board damage and data loss.
  Therefore, only remove the board under hot-swap conditions if the board is to be operated in slots for which hot swap is explicitly permitted by the system documentation (in most hot-swap systems peripheral slots only).
- **Data loss**
  Removing the board from a powered system with IDE devices attached to the board's primary or secondary IDE interface via the ACC/RTB-602 results in data loss.
  Only remove the board from a powered system without IDE devices attached to the board's primary or secondary IDE interface via the ACC/RTB-602.

- **Board damage**
  Touching the board or electronic components in a non-ESD protected environment causes component and board damage.
  Before touching boards or electronic components, make sure that you are working in an ESD-safe environment.

The removal procedure depends on the system the board is to be removed from.

From Basic Hot-Swap System

1. Carry out software disconnection using system documentation
2. Wait until software disconnection process has been completed
3. Remove screws from front panel
4. Open handles
5. Remove board from peripheral slot of powered system

From Full Hot-Swap or High-Availability System

1. Remove screws from front panel
2. Press red button to unlock handles

Figure 8: *Button on Handle*

3. Open handles
   The hot-swap switch will be opened automatically.
4. Wait until blue hot-swap LED is illuminated

Caution

Data loss

Removing the board from the backplane while the hot-swap LED is still off causes data loss.
Therefore, wait until the blue hot-swap LED is on before removing the board.

5. Remove board from peripheral slot of powered system

3

Controls, Indicators, and Connectors

Front Panel

The front panel provides a compatibility glyph indicating whether you use a CompactPCI standard or a PSB board variant:

Figure 9: *Compatibility Glyphs for CompactPCI Standard and PSB Board Variants*

The following figure highlights the position of the cutouts for the PMC modules, the keys, the connectors and the LEDs on the CPCI-690 front panel.

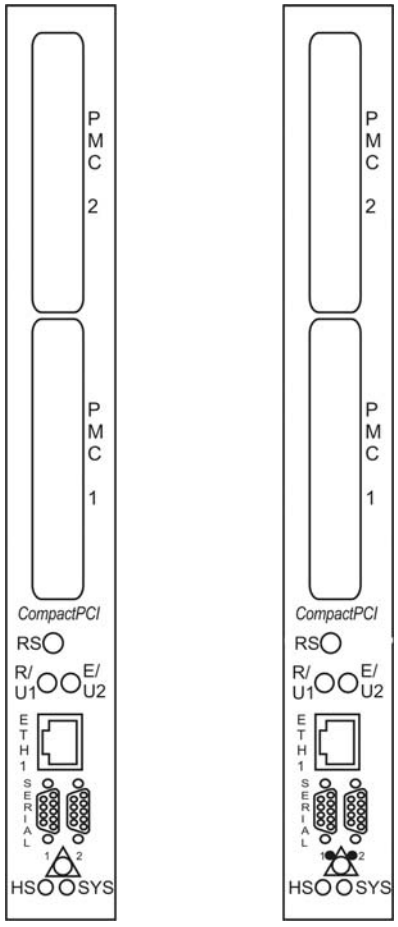
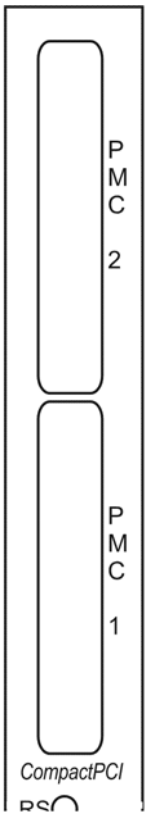


Figure 10: *CompactPCI Standard and PSB Front Panel*

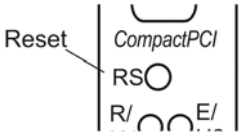
PMC Cutouts

The front panel provides two cutouts to install PMC modules.



Key

The only key available at the front panel is the mechanical reset key.



If the board is installed into a peripheral slot of a standard CompactPCI system or a node slot of a PSB backplane using the CompactPCI bus, the reset key instantaneously affects the board by generating a main reset when enabled and toggled. If the board is installed into a system slot, the main reset additionally generates a CompactPCI reset, i.e. a reset of the whole system.

If a board is installed into a PSB backplane that does not use the CompactPCI bus, pressing the reset key results in a board reset.

When the reset key is pressed, all on-board I/O devices and the CPU are reset. Reset is held active until the key is released.

For information on how to disable the key, see Table 7 “Switch Settings” page 2-14. For information on performing a reset via PowerBoot, refer to “Reset” page 4-47.

LEDs

The front panel provides the following four LEDs:

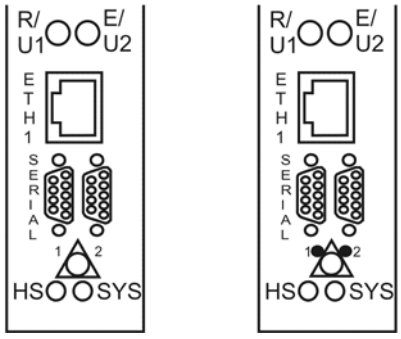


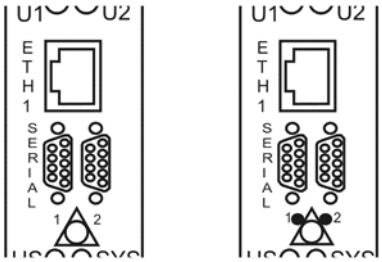
Table 8: *Description of Front Panel LEDs*

LED	Description
R / U1	**Hardware-LED mode** (default): Reset LED indicates board status Green: Normal operation Red: Reset is active **User-LED mode** (see "userled" page 4-26): User LED 1 can be programmed to be red, green or OFF
E / U2	**Hardware-LED mode** (default): Blinking Ethernet LED indicates the status of the Ethernet 1 (default), 2 and 3 interfaces respectively, depending on configuration by software (programmable, see "userled" page 4-26). Green: Ethernet link activity OFF: No link activity **User-LED mode** (programmable, see "userled" page 4-26): User LED 2 can be programmed to be red, green or OFF
HS	Hot swap LED: Indicates hot-swap status Blue: Board is installed and may be removed from the system OFF: Board must not be removed from the system
SYS	System controller LED Yellow: Board is installed in a system slot and acts as system controller OFF: Board is installed in a peripheral slot and acts as intelligent I/O board

Connectors

The front panel provides the following connectors:

- RJ45 for Ethernet 1
- Micro D-SUB for serial ports 1 and 2



The following connector pinouts provide information on signal assignments.

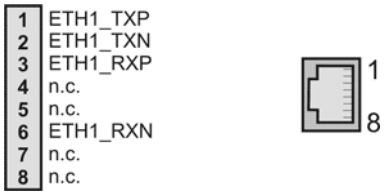


Figure 11: *Ethernet 1 Connector Pinout*

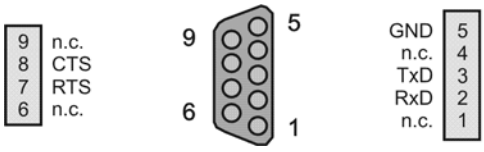


Figure 12: *COM 1 and 2 Connector Pinout*

CompactPCI Connectors

The board provides the CompactPCI connectors J1, J2, J3, and J5. The CompactPCI interface is clocked with 33MHz and is 32-bit wide.

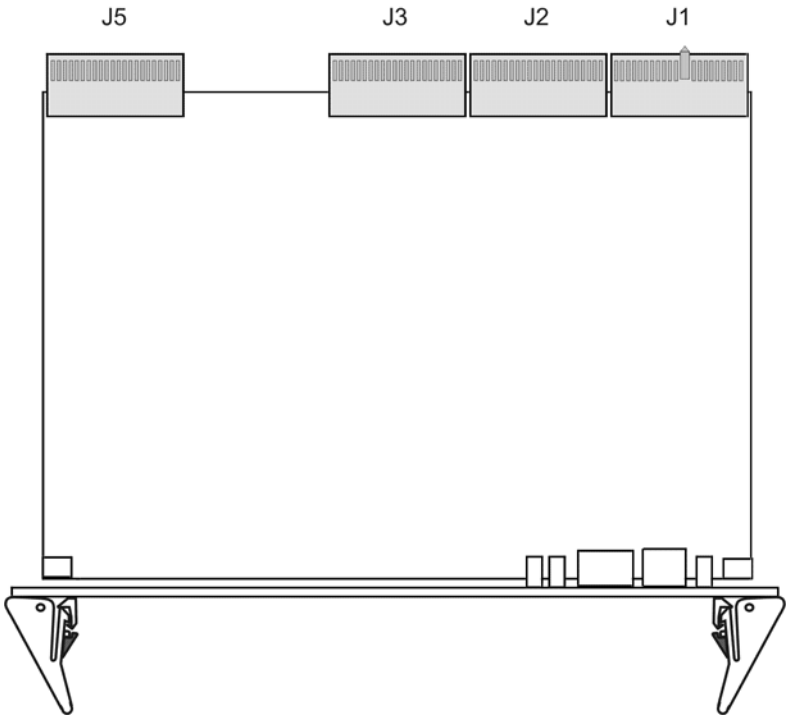


Figure 13: *Location of CompactPCI Connectors*

J1 and J2

The J1 and J2 connectors implement the CompactPCI 64-bit connector pinout as specified by the CompactPCI specification PICMG 2.0 R3.0. Therefore, these pinouts are not documented in this Reference Guide.

J3

Connector J3 provides interfaces to

- Ethernet 2 and 3
- PMC1 user I/O

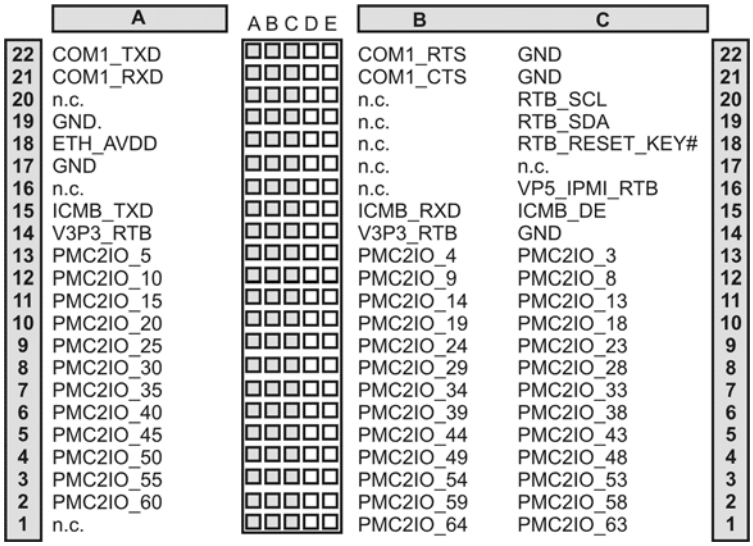
	A	B	C	A B C D E	D	E	
19	GND	GND	GND		GND	GND	19
18	ETH2_TXP	ETH2_TXN	GND		Reserved	Reserved	18
17	ETH2_RXP	ETH2_RXN	GND		Reserved	Reserved	17
16	ETH3_TXP	ETH3_TXN	GND		Reserved	Reserved	16
15	ETH3_RXP	ETH3_RXN	GND		Reserved	Reserved	15
14	GND	GND	GND		GND	GND	14
13	PMC1IO_5	PMC1IO_4	PMC1IO_3		PMC1IO_2	PMC1IO_1	13
12	PMC1IO_10	PMC1IO_9	PMC1IO_8		PMC1IO_7	PMC1IO_6	12
11	PMC1IO_15	PMC1IO_14	PMC1IO_13		PMC1IO_12	PMC1IO_11	11
10	PMC1IO_20	PMC1IO_19	PMC1IO_18		PMC1IO_17	PMC1IO_16	10
9	PMC1IO_25	PMC1IO_24	PMC1IO_23		PMC1IO_22	PMC1IO_21	9
8	PMC1IO_30	PMC1IO_29	PMC1IO_28		PMC1IO_27	PMC1IO_26	8
7	PMC1IO_35	PMC1IO_34	PMC1IO_33		PMC1IO_32	PMC1IO_31	7
6	PMC1IO_40	PMC1IO_39	PMC1IO_38		PMC1IO_37	PMC1IO_36	6
5	PMC1IO_45	PMC1IO_44	PMC1IO_43		PMC1IO_42	PMC1IO_41	5
4	PMC1IO_50	PMC1IO_49	PMC1IO_48		PMC1IO_47	PMC1IO_46	4
3	PMC1IO_55	PMC1IO_54	PMC1IO_53		PMC1IO_52	PMC1IO_51	3
2	PMC1IO_60	PMC1IO_59	PMC1IO_58		PMC1IO_57	PMC1IO_56	2
1	GND	PMC1IO_64	PMC1IO_63		PMC1IO_62	PMC1IO_61	1

Figure 14: *J3 Connector Pinout*

J5

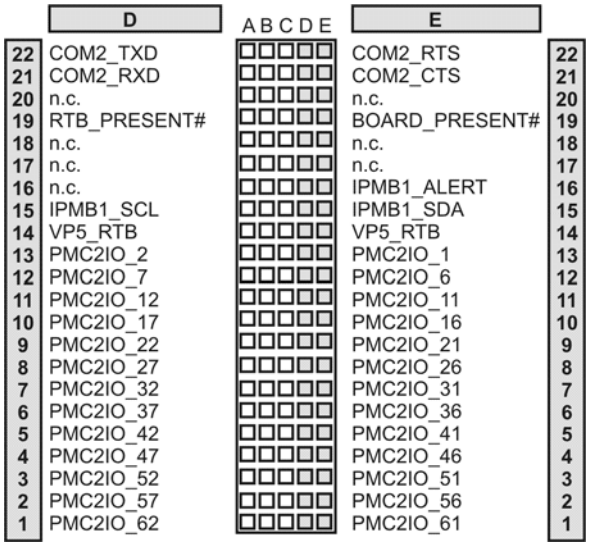
Connector J5 provides interfaces to:

- COM 1 and 2
- PMC2 user I/O
- ICMB and IPMB1 port
- RTB reset key and I2C devices



	A	A B C D E	B	C	
22	COM1_TXD		COM1_RTS	GND	22
21	COM1_RXD		COM1_CTS	GND	21
20	n.c.		n.c.	RTB_SCL	20
19	GND.		n.c.	RTB_SDA	19
18	ETH_AVDD		n.c.	RTB_RESET_KEY#	18
17	GND		n.c.	n.c.	17
16	n.c.		n.c.	VP5_IPMI_RTB	16
15	ICMB_TXD		ICMB_RXD	ICMB_DE	15
14	V3P3_RTB		V3P3_RTB	GND	14
13	PMC2IO_5		PMC2IO_4	PMC2IO_3	13
12	PMC2IO_10		PMC2IO_9	PMC2IO_8	12
11	PMC2IO_15		PMC2IO_14	PMC2IO_13	11
10	PMC2IO_20		PMC2IO_19	PMC2IO_18	10
9	PMC2IO_25		PMC2IO_24	PMC2IO_23	9
8	PMC2IO_30		PMC2IO_29	PMC2IO_28	8
7	PMC2IO_35		PMC2IO_34	PMC2IO_33	7
6	PMC2IO_40		PMC2IO_39	PMC2IO_38	6
5	PMC2IO_45		PMC2IO_44	PMC2IO_43	5
4	PMC2IO_50		PMC2IO_49	PMC2IO_48	4
3	PMC2IO_55		PMC2IO_54	PMC2IO_53	3
2	PMC2IO_60		PMC2IO_59	PMC2IO_58	2
1	n.c.		PMC2IO_64	PMC2IO_63	1

Figure 15: *J5 Connector Pinout Rows A to C*



	D	A B C D E	E	
22	COM2_TXD		COM2_RTS	22
21	COM2_RXD		COM2_CTS	21
20	n.c.		n.c.	20
19	RTB_PRESENT#		BOARD_PRESENT#	19
18	n.c.		n.c.	18
17	n.c.		n.c.	17
16	n.c.		IPMB1_ALERT	16
15	IPMB1_SCL		IPMB1_SDA	15
14	VP5_RTB		VP5_RTB	14
13	PMC2IO_2		PMC2IO_1	13
12	PMC2IO_7		PMC2IO_6	12
11	PMC2IO_12		PMC2IO_11	11
10	PMC2IO_17		PMC2IO_16	10
9	PMC2IO_22		PMC2IO_21	9
8	PMC2IO_27		PMC2IO_26	8
7	PMC2IO_32		PMC2IO_31	7
6	PMC2IO_37		PMC2IO_36	6
5	PMC2IO_42		PMC2IO_41	5
4	PMC2IO_47		PMC2IO_46	4
3	PMC2IO_52		PMC2IO_51	3
2	PMC2IO_57		PMC2IO_56	2
1	PMC2IO_62		PMC2IO_61	1

Figure 16: *J5 Connector Pinout Rows D and E*

4

PowerBoot

Introduction

PowerBoot is the firmware providing an interface between the operating system and the hardware of the board. It is used for hardware configuration. Before loading the operating system, PowerBoot performs basic hardware tests and prepares the board for the initial boot-up procedure.

PowerBoot can be used for the following tasks:

- Booting the board from an operating system, e.g. Linux
- Testing the board

Booting the board is described in detail in “Booting the Board” page 4-5. The other tasks can be performed by means of entering commands described briefly in “Command Set” page 4-12.

Address Mapping

After you power on the board, the on-board devices are initialized. The following table lists the default addresses of the board's devices mapped by PowerBoot.

Address	Device	Size
00000000_{16} .. $7FFFFFFF_{16}$	SDRAM: On-board memory and memory modules	max. 2 GByte
80000000_{16} .. $8FFFFFFF_{16}$	CompactPCI memory space	1 GByte
90000000_{16} .. $9FFFFFFF_{16}$	On-board PCI memory space (PMC)	1 GByte
$F0000000_{16}$ .. $F000000F_{16}$	Board registers (CS #0)	16 Bytes
$F0100000_{16}$ .. $F0107FFF_{16}$	RTC / NVRAM (CS #1)	32 KByte
$F0200000_{16}$ .. $F020000F_{16}$	IPMI Controller (CS #2)	12 Bytes
$F0200000_{16}$ ..$F0200001_{16}$	KCS0[1)]	2 Bytes
$F0200008_{16}$ ..$F0200009_{16}$	KCS1[1)]	2 Bytes
$F1000000_{16}$ .. $F100FFFF_{16}$	Discovery registers	64 KByte
$F4000000_{16}$ .. $F7FFFFFF_{16}$	User flash (CS #3)	max. 64 MByte
$FFF00000_{16}$ .. $FFFFFFFF_{16}$	Boot flash (CSBootCS)	1 MByte (2 x 512)

1) Interface of IPMI controller

Booting the Board

When the board is turned on or rebooted, the presence and functionality of the board's components is tested by the power-on self test (POST). After POST has finished, the board can either boot automatically or after entering the autoboot command. The necessary boot parameters for the possibilities have to be set via the setboot command. The parameters are stored in the on-board NVRAM which keeps its contents during power-off and checks them after the next power-on or after reset.

Reading POST Results

If POST is executed before the boot-up procedure or not, can be set in the POST option in setboot. If POST is enabled and executed, its results are stored in the NVRAM at offset $7CF8_{16}$.

Table 9: *POST Bit Layout Results*

Bit	Description	Result
0	DRAM testbit	1: Okay 0: Failure
1	PCI testbit	1: Okay 0: Failure
2	Not used	0
3	Boot flash test bit	1: Okay 0: Failure
4	Ethernet testbit	1: Okay 0: Failure
5	NVRAM testbit	1: Okay 0: Failure
6	Not used	0
7	Not used	0

A fully functional CPCI-690 will show the value 0x3B for all tests which are okay. A failure will not stop the power-up process since debug functionality is provided via console serial interface 1.

Example:

```
Testing RAM.......................done
Testing Boot FLASH....CSUM 0x20A7..done
Testing PCI Bus ..................done
Testing Discovery Ethernet.........done
```

If POST is disabled, a value of 00_{16} will be stored at NVRAM offset $7CF8_{16}$ to indicate that no test results are available.

Setting Boot Parameters

In order to set the boot parameters, enter the command setboot at the prompt.

The following screen describes briefly the possible options and their current setting. The defined parameters become valid after the next power-on or when a RESET is executed.

Table 10: *Setboot Options*

Option	Description	Parameter	Required to be set when Booting from
Boot select	Location of the binary image which will then be loaded to the load address	0: Net 1: Go 2: Flash+Copy 3: ATA-IDE	-
Auto boot	Auto boot option to boot automatically with a power up	0: Disable 1: Enable	Net, Flash, ATA-IDE
Auto boot delay	Time value to delay boot process. Auto booting countdown can be stopped by pressing any user key on the serial console line.	0..99s	Optional for all
Load address	Address where to copy the binary image	-	Net, Flash, ATA-IDE
Boot address	Address where to execute the go command	-	Net, Go, Flash, ATA-IDE
Ethernet 1, 2 or 3	To select the controller for autoboot	1, 2, 3	Net
Additional tftp retries	Sets a higher retry value for netload	-	-

Table 10: *Setboot Options*

Option	Description	Parameter	Required to be set when Booting from
RARP (Reverse Address Resolution) or ARP (Address Resolution) protocol	To select the protocol used to connect a server to the board for autoboot. If you use RARP, set the IP in NIS.	1: RARP 2: ARP	Net
Server-IP#	Protocol number which selects the TFTP file server. If you use the internet protocol RARP, the server IP# will be ignored.	aaa.bbb.ccc.ddd	Net; for use of ARP protocol only
Target IP#	Set the board IP to define the protocol number identifying the board at internet layer. If you use the internet protocol RARP, the target IP# will be ignored.	aaa.bbb.ccc.ddd	Net; for use of ARP protocol only
Select boot disk	Boot disk to boot from	-	ATA-IDE
TFTP / ATA-IDE boot file name	Binary image name which will be downloaded via net or atapi	image.bin	Net, ATA-IDE
CompactPCI bus probe list	Probe list for CompactPCI	8, 7, 6, 5, 4 , 3, 2	-
Power-on self test (POST)	To enable or disable the power-on self test	0: Enable 1: Disable	-
Primary Booter	To enable or disable the primary booter	0: Disable 1: Enable	-
Watchdog during booting	To enable or disable the watchdog during the boot process.	0:Disabled 1:Enabled	Optional for all
Enable ECC checking	To enable or disable the SDRAM ECC option	0: Disable 1: Enable	Optional for all
Enable shared memory serial	To enable the CompactPCI shared memory communication interface	0: Disable 1: Enable	Only setup on peripheral slot

Selecting the Boot Device

You can chose the board to boot automatically from four different locations:

- Network
- Flash
- ATA / IDE
- Go command

However, for three boot sources prerequisites are necessary:

Table 11: *Boot Source Prerequisites*

Boot Source	Prerequisite
Network	Set up the host as TFTP server. It has to be able to provide the file via Ethernet (TFTP and front panel connector)
Flash	Programming the user flash
ATA / IDE	Partitioning the hard disk

Booting the board via one of the four locations requires different information or setboot options enabled. For further information, refer to Table 10 “Setboot Options” page 4-6.

Programming the User Flash

1. Turn power on
2. Wait until the PowerBoot prompt appears
   ```
   PowerBoot> _
   ```

Caution

Data Loss
The ferase command erases the whole 32 MByte storage of the 32 MByte user flash device. If you want to erase smaller parts of the user flash device, use the offset and length parameters described in “ferase” page 4-52.

3. Enter ferase command to erase user flash
   `ferase user_flash`
   The following message appears:
   ```
   Erasing flash memory ... done
   Erasing flash memory ... done
   Erasing flash memory ... done
   Erasing flash memory ... done
   ```

4. Enter netload command to load image into DRAM memory

```
PowerBoot> netload this-is-my-file-name 100000
PHY-Device at 100MB/s and full duplex negotiated
WANCDM MAC ADDRESS : 00:80:42:11:6A:C3
Transmitting RARP-REQUEST... Reception of RARP-REPLY
Transmitting TFTP-REQUEST to server 02:80:42:0A:0D:79, IP
192.168.41.1
PACKET:307 - loaded $00100000..$001265CF (157136 bytes)
PowerBoot> _
```

5. Enter bf command if you want to change the image

The word “FORCE” is loaded into the user flash:

```
PowerBoot> bf 100000 500000 “FORCE ” p
```

6. Enter md command to verify DRAM memory contents

```
PowerBoot> md 100000
00100000:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
00100010:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
00100020:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
00100030:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
00100040:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
00100050:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
00100060:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
00100070:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
00100080:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
00100090:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
001000a0:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
001000b0:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
001000c0:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
001000d0:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
001000e0:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
001000f0:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
More (cr) ? .
PowerBoot>
```

7. Enter fprog command to program image from DRAM memory into user flash

   The DRAM memory contents from 100000_{16}..$20FFFFF_{16}$ will be programmed.

```
PowerBoot> fprog user_flash 100000
Programming flash memory
0 ... 100%
0 ... 100%
0 ... 100%
0 ... 100%
Done.
```

8. Enter md command to display contents of programmed user flash

```
PowerBoot> md f4000000
f4000000:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
f4000010:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
f4000020:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
f4000030:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
f4000040:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
f4000050:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
f4000060:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
f4000070:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
f4000080:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
f4000090:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
f40000a0:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
f40000b0:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
f40000c0:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
f40000d0:  45 20 46 4f 52 43 45 20  46 4f 52 43 45 20 46 4f
f40000e0:  52 43 45 20 46 4f 52 43  45 20 46 4f 52 43 45 20
f40000f0:  46 4f 52 43 45 20 46 4f  52 43 45 20 46 4f 52 43
More (cr) ?
PowerBoot>
```

Partitioning the Hard Disk

PowerBoot only supports the Ramix IDE PMC modules with hard disk. In order to boot from ATA / IDE, the hard disk must have either a PREP or a DOS 4.0 partition.

Autoboot

There are three possibilities to boot the board:

- Via the autoboot command
  The board boots directly without executing a board reset
- Via the autoboot option in setboot
  The board boots automatically after a reset has been executed
- Via the go option in setboot
  The image loaded to the flash can directly be executed

Restoring Default Values

After parameters have been edited, a checksum is calculated to protect the NVRAM contents from offset $7C00_{16}$ to $7FF0_{16}$ containing all edited parameters. If you have changed a value, restoring the default values is only possible by means of either removing the battery from the board or writing a wrong value into the protected NVRAM range. This results in a wrong NVRAM checksum and leads to a restoration of the default values after the next boot up.

If you want to restore the default values via the NVRAM, proceed as follows:

1. `PowerBoot: m f0107c00`
2. <Enter>
3. `f0107c00  32313130 :1`
4. <Enter>
5. <Ctrl+C>
6. `PowerBoot: reset`

Command Set

In the following, the CPCI-690-specific PowerBoot commands are described in detail. These commands can be used to:

- Test the board
- Get information on devices and settings
- Exchange data
- Enable, disable and change configuration settings

Table 12: *Command Set Overview*

Task		Command	Page
Test	Interface specified by setboot command	atatest	4-14
	Memory	bt	4-15
	Memory read performance	dread	4-17
	Memory write performance	drw	4-18
	Memory read and write performance	dwrite	4-19
Settings	Change baud rate	baud	4-20
	Enable available cache	cache	4-21
	Disable all caches and interrupts and start an executable image	execute	4-21
	Modify or display memory contents	m	4-33
	Modify boot options	setboot	4-24
	Set parameters for SENTINEL SROM	setcpci	4-25
	Set function of user LEDs	userled	4-26
	Enable watchdog	wdog	4-26
Copy	Contents of a source memory area to a destination memory area	bm	4-27
	NVRAM contents to a memory region	nvramrd	4-27
	Memory region into the NVRAM and generates the correct checksum	nvramwr	4-28

Table 12: *Command Set Overview*

Task		Command	Page
Display	Information on the board's PCI bus structure	devshow	4-29
	Supported I2C EEPROMs on the board	eeprom_info	4-31
	Help message	help	4-31
	HID0/1, MSR and L2CR register values	hid	4-32
	Information on the on-board IPMI controller	ipmi_info	4-33
	Memory contents	md	4-33
	Board temperature	temp	4-34
Exchange Data	Make PCI configuration read to a specific PCI location	config_rd	4-35
	Make a PCI configuration write to a specific PCI location	config_wr	4-36
	Read data from the on-board EEPROMs	eeprom_read	4-37
	Write data to an on-board I2C EEPROM	eeprom_write	4-38
	Program flash memory	fprog	4-39
	Send values to the IPMI controller	ipmi_request	4-41
	Load s-records into memory	lo	4-41
	Load binary image from a remote system via network interface	netload	4-43
	Save a memory area via TFTP into a file on a host system.	netsave	4-45
Reset	Hardware reset	hreset	4-47
	Software reset	reset	4-48
Miscella-neous	Fill a memory area with values or an ASCII string	bf	4-49
	Search memory areas for values or an ASCII string	bs	4-50
	Compare two memory areas	bv	4-51
	Erase flash memory devices	ferase	4-52
	Start binary image located in DRAM memory	go	4-53

Testing

The following commands can be used to:

- Test the interface specified by setboot
- Perform basic binary tests to locate memory errors
- Perform memory read performance
- Perform memory write performance
- Perform memory read and write performance

atatest

DESCRIPTION Command to test the interface which is specified with the setboot command. Shows if a disk is present and its format.

SYNTAX `atatest`

PARAMETERS None

EXAMPLE

```
PowerBoot> atatest

Testing ATA/IDE interface...

Found 2 devices on this ATA interface:
   Device 0 type: ATA.
   Device 1 type: unknown type.
Device 0 is the selected device for this run...

Soft Reset...
Seek, CHS, polling...
Seek, LBA, polling...
ATA Identify, CHS, polling...
ATA Identify, LBA, polling...

-Identify ATA/IDE device:
 Signature     : 0x848A
 Serial Number :          i5809080307
 Model Number  : SanDisk SDCFB-64
 Firmware Rev. : vdd 1.00
 Buffer type   : 2 [0x2]
 Buffer size   : 1024 [0x400]
 Cylinders     : 490 [0x0]
 Heads         : 8 [0x0]
 Sect. p. track: 32 [0x0]
 Capabilities  : 512 [0x200]
 Number of sectors LBA : 60160 [0xEB00]
 Number of sectors/card: 125440 [0x1EA00]
```

```
 Bytes per Sector      : 576 [0x240]
 Total Capacity in Bytes: 72253440 [0x44E8000]

ATA Write Buffer, CHS, polling...
ATA Read Buffer, CHS, polling...
Checking Buffer R/W contents...
ATA Read-4-Sectors, LBA, polling...

ATA/IDE-test okay

PowerBoot>
```

bt

DESCRIPTION This commands executes a set of basic binary tests to locate memory errors. The following test types are executed:

Table 13: *BT Test Types*

Test Name	Access Size in Bit
Checkerboard test (reverse)	8, 16, 32
Walking ones test (reverse)	8, 16, 32
Walking zeroes test (reverse)	8, 16, 32
Increment test	8, 16, 32
Prime test	32
March-b test	32

All tests, except the prime test, fill the memory area specified by `begin` and `end`. After the memory area has been filled, the testing starts indicated by

- test forward, i.e. memory is tested from `begin` to `end`
- test backward, i.e. memory is tested from `end` to `begin`

If an error occurs during the test, the following error message will appear:

```
error at location 0x12345678 value found:0xAB should be:0xBA
```

SYNTAX `bt <begin> <end> [e] [c]`

PARAMETERS

begin	Specifies the first byte of the memory area to be tested
end	Specifies the first area of the memory area not to be tested
e	Achieves that the test runs endlessly
c	Achieves that the test uses the cache

EXAMPLE

```
PowerBoot> bt 100000 200000
Blocktest at single mode, testing 65536 bytes from 0x10000..0x1FFFF

Checkerboard test 8bit......done
Checkerboard test reverse...done

Walking ones test 8bit......done
Walking ones test reverse...done

Walking zeros test 8bit.....done
Walking zeros test reverse..done

Increment test 8bit.........done

Checkerboard test 16bit.....done
Checkerboard test reverse...done

Walking ones test 16bit.....done
Walking ones test reverse...done

Walking zeros test 16bit....done
Walking zeros test reverse..done

Increment test 16bit........done

Checkerboard test 32bit.....done
Checkerboard test reverse...done

Walking ones test 32bit.....done
Walking ones test reverse...done

Walking zeros test 32bit....done
Walking zeros test reverse..done

Increment test 32bit........done

Prime test 32bit............done
Prime test 32bit R/T........done

March-b test ...............done

--- Blocktest runs 1 times - 00000000 errors ---

PowerBoot>_
```

dread

DESCRIPTION Command to test memory read performance.

SYNTAX `dread`

PARAMETERS None

EXAMPLE

```
PowerBoot> dread
Memory Read Performance in ns
address range: 0x00010000..0x00410000

    n =       4     8     16    32    64   128   256   512

d = 2^02 :   227    12    10     9     9     9     9     9
d = 2^03 :    14    11    10     9     9     9     9     9
d = 2^04 :    14    11    10     9     9     9     9     9
d = 2^05 :    14    11    10     9     9     9     9     9
d = 2^06 :    14    11    10     9     9     9     9     9
d = 2^07 :    16    13    12    11    10    10    10    18
d = 2^08 :    16    13    12    11    10    11    18    18
d = 2^09 :    16    13    12    11    12    18    18    18
d = 2^10 :    16    13    12    13    18    18    18    20
d = 2^11 :    17    13    16    19    18    18    21   159
d = 2^12 :    18    23    21    20    19    25   160   160
d = 2^13 :    18    23    21    20    32   161   161   160
d = 2^14 :    18    23    21    46   163   162   161   160
PowerBoot>
```

drw

DESCRIPTION Command to test memory write performance.

SYNTAX `drw`

PARAMETERS None

EXAMPLE

```
PowerBoot> drw
Memory Read/Write Performance in ns
address range: 0x00010000..0x00410000

    n =       4     8     16    32    64   128   256   512

d = 2^02 :    85    77    73    71    69    69    69    69
d = 2^03 :    86    77    73    71    69    69    69    69
d = 2^04 :    86    77    73    71    69    69    69    69
d = 2^05 :    86    77    73    71    69    69    69    69
d = 2^06 :    86    77    73    71    69    69    69    69
d = 2^07 :    86    77    73    71    69    69    69    69
d = 2^08 :    86    77    73    71    69    69    69    69
d = 2^09 :    86    77    73    71    69    69    69    69
d = 2^10 :    86    77    73    71    69    69    69    70
d = 2^11 :    85    77    73    71    69    69    71   242
d = 2^12 :    87    78    74    72    70    74   250   250
d = 2^13 :    87    78    74    72    79   251   251   250
d = 2^14 :    87    78    74    89   253   252   251   250
PowerBoot>
```

dwrite

DESCRIPTION Command to test memory read and write performance.

SYNTAX `dwrite`

PARAMETERS None

EXAMPLE

```
PowerBoot> dwrite
Memory Write Performance in ns
address range: 0x00010000..0x00410000

    n =       4     8     16    32    64   128   256   512

d = 2^02 :   171   154   146   142   140   138   138   138
d = 2^03 :   172   155   146   142   140   138   138   138
d = 2^04 :   172   155   146   142   140   138   138   138
d = 2^05 :   172   155   146   142   140   138   138   138
d = 2^06 :   172   155   146   142   140   138   138   138
d = 2^07 :   172   155   146   142   140   138   138   138
d = 2^08 :   172   155   146   142   140   138   138   138
d = 2^09 :   172   155   146   142   140   138   138   138
d = 2^10 :   172   155   146   142   140   138   138   138
d = 2^11 :   172   155   146   142   140   138   138   138
d = 2^12 :   172   155   146   142   140   138   138   138
d = 2^13 :   172   155   146   142   140   138   138   138
d = 2^14 :   172   155   146   142   140   138   138   138
PowerBoot>
```

Setting

The following commands are used to set:

- Baud rate
- Cache
- Interrupts
- Memory contents
- Boot Options
- Parameters for the SENTINEL SROM
- User LED
- Watchdog

baud

DESCRIPTION Command to change the actual baud rate to the specified value. To apply the new baud rate, a reboot is necessary.

Note: If you chose a baud rate the terminal is not able to handle, you will not get any output on screen. Inform yourself about the settings of your screen and choose baud rates which provide a screen output.

SYNTAX `baud <value>`

PARAMETERS `600|1200|2400|4800|9600|19200|38400|115200`

EXAMPLE

```
PowerBoot> baud 9600
PowerBoot> reset
```

cache

DESCRIPTION Enables or disabled the available caches.

SYNTAX `cache <d|i|2> <e|d>`

PARAMETERS

`d`	Selects the data cache
`i`	Selects the instruction cache
`2`	Selects the 2nd level cache
`e,d`	e: Enables the cache d: Disables the cache

EXAMPLE

```
PowerBoot> cache d d
Data cache disabled

PowerBoot>
```

execute

DESCRIPTION Command to disable all caches and interrupts and start an executable image.

SYNTAX `execute <address>`

PARAMETERS

`address`	start address of the executable image.

EXAMPLE

```
PowerBoot> execute 100000
```

m

DESCRIPTION This command provides an editor with which you can display and modify the memory contents of an address. Three tasks can be carried out:

- Displaying and modifying the memory contents of an address
- Specifying the memory access size and displaying and modifying the memory contents of an address
- Specifying the memory access size and type and displaying and modifying the memory contents of an address

To navigate between the addresses, a key and different commands can be entered. They do not change the access option in the command line. The following key and commands are supported:

Table 14: *Navigation Commands*

Key/Command	Description
<Ctrl+Enter>	Current address + 1
=	Same as current address
-	Current address – 1 x size
-<count>	Current address – <*count*>
+	Current address + 1 x size
+<count>	Current address + <*count*>
#address	Address
.	Exit to the PowerBoot command line

SYNTAX `m <address> [<size>] [<type>]`

PARAMETERS **Note: Type options O and E override the options B, W, and L. Access options B, W, L, N, and E check whether the write access has been successful by performing a read access after the write access. If the written and the read data do not match, the command is terminated and an error message displayed.**

`address` First byte which is read for displaying and modifying the memory contents

`size` Specifies the memory access size. Possible values are:

- B: Memory access is byte-sized (8 bit)
- W: Memory access is word-sized (16 bit)
- L: Memory access is long-word-sized (32 bit)

`type` Specifies the memory access type. Possible values are:

- N: Memory access is limited to writing, the current contents are not displayed
- E: Memory access is byte-sized and refers only to even addresses
- O: Memory access is byte-sized and refers only to odd addresses

EXAMPLE

```
PowerBoot> m 100000 b
00100000 00 : 4f
00100001 00 : =
00100001 00 : =
00100001 00 : -
0010000000 : #2000
00102000 00 : .
PowerBoot>
```

setboot

DESCRIPTION Command to set boot options.

SYNTAX setboot

PARAMETERS None

EXAMPLE

```
PowerBoot> setboot

 -General Boot Parameters-

Boot select [0=Net,1=Go,2=Flash+Copy,3=ATA-IDE (0) :
Auto boot [0=disable, 1=enable], (0) :
Auto boot delay [0..99s], (0) :
Load address (00000000) :
Boot address (00000000) :

 -TFTP Ethernet/SCSI/ATA-IDE boot file parameters-

Ethernet 1, 2 or 3 : (0) :
Additional tftp retries (00000000):
RARP [1] or ARP [2] protocol : (1) :
Server-IP# [aaa.bbb.ccc.ddd] :  :
Target-IP# [aaa.bbb.ccc.ddd] :  :
Select bootdisk (0) :
TFTP/SCSI/ATA-IDE Boot file name :
 :


 -PCI-to-PCI bridge host-mode parameters-

Compact PCIbus probe list [8,7,6,5,4,3,2]  (8,7,6,5,4,3,2) :

 -Power On Self Test (POST) parameters-

Power ON Self Test POST [1=disable, 0=enable], (0)   :

 -Primary Booter parameters [FORCE internal use only]-

Primary Booter [0=disable, 1=enable], (0)   :
Watchdog during booting (0) :
Enable ECC checking (0) :
Enable Shared Memory Serial (0):
CSUM : 0xCEE
PowerBoot>
```

setcpci

DESCRIPTION Command to set parameter for the SENTINEL SROM.

SYNTAX `setcpci [-e]`

PARAMETERS

`-e` Extended options

EXAMPLE

```
PowerBoot> setcpci -e
Downstream BAR 0 Setup (0xfffff000) :
Downstream BAR 0 translate (0x0) :
Downstream BAR 2 Setup (0xfff00000) :
Downstream BAR 2 translate (0x0) :
Downstream BAR 3 setup (0x0) :
Downstream BAR 3 translate (0x0) :
Downstream BAR 4 setup (0x0) :
Downstream BAR 4 translate (0x0) :
Upstream BAR 2 setup (0xfff00000) :
Upstream BAR 2 translate (0x0) :
Upstream BAR 3 setup (0x0) :
Upstream BAR 3 translate (0x0) :
Scratchpad (0x0) :

Expert settings:

Chip control 0 (0x0) :
Chip control 1 (0x0) :
Arbiter control and status (0x8020) :
Primary SERR# control (0x7) :
Secondary SERR# control (0x7) :
User capability ID (0x0) :
Message control (0x0) :
MSI next item ptr (0x0) :
Primary class code (0x6800000) :
Secondary class code (0x6800000) :

Do you really want to write data to the SROM? [y/n]:n

 Command -NOT- executed

PowerBoot>
```

userled

DESCRIPTION Command to set the function of the user LEDs.

SYNTAX `userled <1|2> <red|green|dis|HW|ethX>`

PARAMETERS

`1`	Selects user LED 1
`2`	Selects user LED 2
`red`	USERLED is red
`green`	USERLED is green
`dis`	USERLED is disabled
`HW`	USERLED shows default state after power up
`ethX`	USERLED shows Ethernet activity. Possible values for X are 1, 2 or 3 for Ethernet ports 1-3.

EXAMPLE

```
PowerBoot> userled 2 eth3
PowerBoot>
```

wdog

DESCRIPTION Command to enable the watchdog.

SYNTAX `wdog [<sec>]`

PARAMETERS

`sec`

Note: The parameter is interpreted as a hexadecimal value.

Countdown in seconds before the reset

EXAMPLE

```
PowerBoot> wdog 10
Setting watchdog counter to 16 sec
Enable Watchdog!

PowerBoot>
```

Copying

The following commands are used to copy:

- Memory area
- NVRAM contents

bm

DESCRIPTION Command to copy the contents of a specified source memory area to a memory area starting at a specified destination address. However, if the source and the destination area overlap, only the destination area will be complete.

SYNTAX `bm <begin> <end> <destination>`

PARAMETERS

`begin` Specifies the start address of the source memory area, i.e. the first byte of the source memory area used for copying

`end` Specifies the end address of the source memory area, i.e. the first byte of the source memory area not used for copying

`destination` Specifies the start address of the destination memory area where the first byte of the source memory area is copied to

EXAMPLE

```
PowerBoot>bm 2D00 3D00 5E00
PowerBoot>
```

nvramrd

DESCRIPTION Command to copy the NVRAM contents to a memory region to modify it.

SYNTAX `nvramrd <dest-address>`

PARAMETERS

`dest-address` Memory address where to copy the NVRAM contents

EXAMPLE

```
PowerBoot> nvramrd 100000
Read NVRAM from offset 0x0000..0x7fff to 0x00100000
CSUM calculated : 0xCEC
CSUM NVRAM read : 0xCEC
PowerBoot>
```

nvramwr

DESCRIPTION Command to copy a memory area into the NVRAM and generate the correct checksum.

SYNTAX `nvramwr <scr-address>`

PARAMETERS

scr-address	Location of the new NVRAM contents

EXAMPLE

```
PowerBoot> nvramwr 100000
Do you really want to write data to NVRAM [y/n]: y
Write 32KB data at address 0x00100000 to NVRAM
CSUM : 0xCEC
PowerBoot>
```

Displaying

The following commands are used to display information on:

- The board's PCI bus structure
- The board's I2C EEPROMs
- Commands provided by PowerBoot
- Processor register values
- On-board IPMI controller
- Memory contents
- Board temperature

devshow

DESCRIPTION Command to display information on the board PCI bus structure.

SYNTAX `devshow [1|2|3|4] [<node>] [<verbose level>]`

PARAMETERS

`1|2|3|4`	Selects the information level 1 to 4
`node`	Selects a specific node
`verbose level`	Shows if child nodes exist or not

RETURNS Displays all found PCI devices and resources.

EXAMPLE

```
PowerBoot> devshow 4
o Discovery@0            [inst 0, loc 0/0/0, node 0x1ff2125c, ref 3]
  - Device type: root
  - Secondary location: 0x0/0x0/0x0
  - Driver Rtn / Parm: 0xfff17374 / 0x00000000
  - ID: 0x00000000
  - Flags: 0x0000011b
  o PCI@0                [inst 0, loc 0/0/0, node 0x1ff21d68, ref 1]
    - Implements Classes: PCI
    - Implements bus type: PCI bus
    - Secondary location: 0x0/0x0/0x0
    - Driver Rtn / Parm: 0xfff17458 / 0x1ff21efc
    - ID: 0x00000000
    - Flags: 0x00000013
    o CompactPCI@20 [inst 0, loc 0/14/0, node 0x1ff226e0, ref 0]
      - Implements Classes: PCI CPCI
      - Implements bus type: PCI bus
      - Device type: PCI
      - Expands parent bus domain
      - Driver Rtn / Parm: 0xfff1756c / 0x1ff25c88
      - DevVen: 0x00011146  SubSys: 0x00000000  Base/SubClass:
        0x06/0x04
      - No device information available
      - BAR 0  :          87ffe000 /   4KB   MEM, 32bit
      - BAR 1  :          8fffff00 / 128Byte IO
      - Prefetchable Memory : fff00000 to 000fffff (DISABLED)
      - Memory Mapped IO    : fff00000 to 000fffff (DISABLED)
      - IO Range            : 0000f000 to 00000fff (DISABLED)
      - Secondary Bus=1 Subordinate Bus=1
      - Flags: 0x00000321
  o PCI@1                [inst 1, loc 0/1/0, node 0x1ff213f0, ref 0]
    - Implements Classes: PCI
    - Implements bus type: PCI bus
    - Secondary location: 0x0/0x0/0x0
    - Driver Rtn / Parm: 0xfff17458 / 0x1ff21584
    - ID: 0x00000000
    - Flags: 0x00000013

PowerBoot>
```

eeprom_info

DESCRIPTION Command to display all supported I2C EEPROMs on the board.

SYNTAX `eeprom_info`

PARAMETERS None

EXAMPLE

```
PowerBoot> eeprom_info
MSG: EEPROM information list
ID_ROM_0              16384 Byte: ID-ROM of the base board
ID_ROM_1                256 Byte: ID-ROM of the RTB
ID_ROM_2                256 Byte: ID-ROM of the IPMI
I2C_PCIBRIDGE_0         256 Byte: I2C-EEPROM of PCI-to-PCI Bridge
MSG: End of list

PowerBoot>
```

help

DESCRIPTION This command displays all commands with syntax provided by PowerBoot.

SYNTAX `help`

PARAMETERS None

EXAMPLE

```
PowerBoot> help

 --- Command words ---

     ATATEST - Test of ATA/IDE interface connection
    AUTOBOOT - Init SETBOOT command parameter settings
        BAUD - BAUD 600|1200|2400|4800|9600|19200|38400|115200
          BF - BF <begin>,<end>,<value>[,B|W|L|P]
          BM - BM <begin>,<end>,<destination>
          BS - BS <begin>,<end>,[/]<value>[,B|W|L|P]
          BT - BT <begin>,<end>,[e], [c]
          BV - BV <begin>,<end>,<destination>
       CACHE - CACHE <d,i,2> <e,d>
    DEVSHOW - Show board device tree [1|2|3|4] [node] [verbose level]
       DREAD - Read test to check caches
         DRW - R/W test to check caches
      DWRITE - Write test to check caches
 EEPROM_INFO - Display information about EEPROMs
 EEPROM_READ - <device> <offs> <byteCnt> [dstAddr]
EEPROM_WRITE - [-e] <device> [offs byteCnt srcAddr]
     EXECUTE - EXECUTE <address>
```

```
      FERASE - FERASE <flashbank>[,flashoffset,length]
       FPROG - FPROG <flashbank>,<source>[,flashoffset[,length]]
          GO - GO <address>
        HELP - HELP print this help message
         HID - Show HID0/1, MSR, L2CR register
      HRESET - Hardware Reset execution
 IPMI_FLSUPD - IPMI_FLSUPD <begin> <end> [addr]
   IPMI_INFO - Show IPMI info
IPMI_REQUEST - IMPI_REQUEST <HEX> [HEX] [HEX] ...
          LO - Load S-Rec: LO [offset][,V][,commands]
           M - M  <address>[,B|W|L&N|E|O]
          MD - MD <address>[,count]
    NETLOAD - NETLOAD <file> <dest> [Trgt-Enet#] [Trgt-IP# Server-IP#]
    NETSAVE - NETSAVE <file><start><end> [Trgt-Enet#] [Trgt-IP# Srvr-IP#]
     NVRAMRD - NVRAMRD <dest-address>
     NVRAMWR - NVRAMWR <src-address>
   PRIMABOOT - Start Primary Booter - [FORCE internal use]
       RESET - RESET board
     SETBOOT - SETBOOT edit autoboot parameters
     SETCPCI - SETCPCI [-e] edit cpci bridge boot parameters
        TEMP - Show board temperature
     USERLED - USERLED <1|2> <red|green|dis|HW|ethX>
        WDOG - WDOG [sec] enables watch dog

CONFIG_RD - Host:PCIbus cfg RD <dom#><bus#><dev#><fun#><reg#> [B|W|L]
CONFIG_WR - Host:PCIbus cfg WR <dom#> <bus#> <dev#> <fun#> <reg#> <data> [B|W|L]

PowerBoot>
```

hid

DESCRIPTION Command to display the HID0/1, MSR, PVR and L2CR processor register values.

SYNTAX `hid`

PARAMETERS None

EXAMPLE

```
PowerBoot> hid

 -CPU-
 HID0 = 0x8000C0A4
 HID1 = 0x50000000
 MSR  = 0x00009030
 PVR  = 0x70000102
 L2CR = 0x80080001
PowerBoot>
```

ipmi_info

DESCRIPTION Command to display information on the on-board IPMI controller.

SYNTAX `ipmi_info`

PARAMETERS None

EXAMPLE

```
PowerBoot> ipmi_info
IPMI-Informations

Device-ID: 1
Device-Revision: 1
Firmware-Revision: 0.9
IPMI-Version: 1
Geographical Addr: 0x1
IPMI-I2C_Addr: 0x20
IPMI-Role: 0

PowerBoot>
```

md

DESCRIPTION This command displays the memory contents. The data is displayed in hexadecimal notation and ASCII code. If the data cannot be displayed in ASCII code, a full stop is displayed instead.

The memory contents are displayed in blocks of 16 lines, each block representing 16 byte. If you want to display the next block of lines press <Enter>. If you want to return to the command line, enter the full stop (.) character.

SYNTAX `md <address> [<size>]`

PARAMETERS

`address` First byte which is read in order to display the memory contents.

`size` Specifies the memory access size

EXAMPLE

```
PowerBoot> md f1000000

f1000000:  ff 20 00 70 ff 20 00 70  00 00 00 00 00 00 00 00   . .p. .p........
f1000010:  ff 00 00 00 ff 00 00 00  ff 0f 00 00 ff 0f 00 00   ................
f1000020:  00 00 00 00 00 00 00 00  00 0f 00 00 00 0f 00 00   ................
f1000030:  00 0f 00 00 00 0f 00 00  40 0f 00 00 40 0f 00 00   ........@...@...
f1000040:  5f 0f 00 00 5f 0f 00 00  e0 08 00 01 e0 08 00 01   _..._...........
f1000050:  ff 08 00 00 ff 08 00 00  00 08 00 01 00 08 00 01   ................
f1000060:  3f 08 00 00 3f 08 00 00  10 0f 00 01 10 0f 00 01   ?...?...........
f1000070:  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
f1000080:  40 08 00 01 40 08 00 01  7f 08 00 00 7f 08 00 00   @...@...........
f1000090:  e0 09 00 01 e0 09 00 01  ff 09 00 00 ff 09 00 00   ................
f10000a0:  00 09 00 01 00 09 00 01  3f 09 00 00 3f 09 00 00   ........?...?...
f10000b0:  40 09 00 01 40 09 00 01  7f 09 00 00 7f 09 00 00   @...@...........
f10000c0:  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
f10000d0:  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff   ................
f10000e0:  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff   ................
f10000f0:  e0 08 00 00 e0 08 00 00  00 08 00 00 00 08 00 00   ................
More (cr) ? .
PowerBoot>
```

temp

DESCRIPTION Command to display the board temperature. For the location where the temperature is measured on the board, refer to section “Temperature Sensor” page 5-16.

SYNTAX

```
temp
```

PARAMETERS None

EXAMPLE

```
PowerBoot> temp
Board temperature: +27C

PowerBoot>
```

Exchanging Data

The following commands are used to:

- Perform a PCI configuration read or write
- Read data from the on-board EEPROMs
- Write data to an on-board I2C EEPROM
- Program flash memory devices
- Start an IPMI flash update
- Load S-records or a binary image via TFTP
- Save a specified memory area via TFTP

config_rd

DESCRIPTION Command to perform a PCI configuration read to a specific PCI location.

SYNTAX `config_rd <dom#> <bus#> <dev#> <fun#> <reg#> [B|W|L]`

PARAMETERS

`dom#`	Domain number of first or second PCI bus
`bus#`	Bus number
`dev#`	Device number
`fun#`	Function number
`reg#`	Register offset
`B|W|L`	Size of the read: byte, word or long

EXAMPLE

```
PowerBoot> config_rd 0 0 14 0 4
address: 0x8000A004
0x02100007
PowerBoot>
```

config_wr

DESCRIPTION Command to perform a PCI configuration write to a specific PCI location.

Caution

Overwriting PCI configuration space
SENTINEL may not work if configuration has been changed.
Only change the configuration if you are familiar with the consequences.
If after a change the board does not work, reboot the board.

SYNTAX `config_wr <dom#> <bus#> <dev#> <fun#> <reg#> <data> [B|W|L]`

PARAMETERS

`dom#`	Domain number of first or second PCI bus
`bus#`	Bus number
`dev#`	Device number
`fun#`	Function number
`reg#`	Register offset
`data`	Data to write
`B|W|L`	Size of the read: byte, word or long

EXAMPLE

```
PowerBoot> config_wr 0 0 14 0 4 7
address: 0x8000A004
PowerBoot>
```

eeprom_read

DESCRIPTION Command to read data from the on-board EEPROMs.

SYNTAX `eeprom_read <device> <offs> <byteCnt> [<dstAddr>]`

PARAMETERS

`device`	Name of the device is given by the eeprom_info command
`offs`	Offset from which the read starts
`byteCnt`	Count of bytes to be read
`dstAddr`	Destination of the read data

EXAMPLE

```
PowerBoot> eeprom_read ID_ROM_0 0 30
MSG: Begin of S-record
S11300000C200C851C2500344653626A6C6E720009
S1130010464F52434520434F4D505554455253008B
S113002009435043492D3639300010323131313032D2
S9030000FC
MSG: End of S-record

PowerBoot>
```

eeprom_write

DESCRIPTION Command to write data to an on-board I2C EEPROM.

Caution

Board malfunction
The I2C EEPROM contains data for ID ROM and IPMI ID ROM. Overwriting this data results in a malfunction of the board.
Do not overwrite the data for ID ROM and IPMI ID ROM.

SYNTAX `eeprom_write [-e] <device> [[<offs> <byteCnt>] <srcAddr>]`

PARAMETERS

`-e`	Echo mode
`device`	Name of the device (given by the eeprom_info command)
`offs`	Offset to write to
`byteCnt`	Count of the bytes
`srcAddr`	Data address

EXAMPLE

```
PowerBoot> eeprom_write ID_ROM_0
MSG: Waiting for S-record data
.
.
.
MSG: End of S-record

PowerBoot>
```

fprog

DESCRIPTION This command programs flash memory devices. Three tasks can be carried out:

- Programming a whole flash memory device
- Programming the space between a specified offset and the end of the flash memory device
- Programming a specified number of bytes starting at a specified destination offset

SYNTAX `fprog <flashDevice> <source> [<flashOffset> [<length>]]`

PARAMETERS

`flashDevice` Specifies the name of the flash memory device to be programmed

`source` Specifies the source address containing the data with which the flash memory device is to be programmed

`flashOffset` Specifies the destination offset within the flash memory device

`length` Specifies the number of bytes to be programmed

EXAMPLE 1 Suppose you want to reprogram the write-protected boot flash 1 with the data stored at source address 00100000_{16}. Since FPROG checks whether the flash is write-protected or not, the following message appears:

```
PowerBoot> fprog boot_flash1 100000
Do not reprogram BOOT_FLASH1, this would destroy the booter
Device is write protected
```

EXAMPLE 2 Boot flash 2 is programmed with the data stored at source address 00100000_{16}:

```
PowerBoot> fprog boot_flash2 100000
Programming flash memory

  0 |################################################| 100%

Done.

PowerBoot> _
```

ipmi_flsupd

Danger

Defect IPMI Flash
When updating the IPMI flash with wrong data, it becomes unusable. Only update the IPMI flash with data provided by Motorola.

DESCRIPTION Command to start an IPMI flash update. An update via IPMB is possible with the address option. For a detailed description of the update procedure refer to the current version of *IPMI Firmware for PPC/CPCI-690 and PPC/CP-CI-695 Installation Guide* available via S.M.A.R.T.

SYNTAX

```
IPMI_FLSUPD <begin> <end> [<addr>]
```

PARAMETERS

Parameter	Description
begin	Start of the S-record
end	End of the S-record
addr	IPMB I2C address of the connected board.

EXAMPLE

```
PowerBoot  netload ipmicore_all_0_96.s3r 100000 192.168.41.111
192.168.41.1
 PHY-Device at 100MB/s and full duplex negotiated
 WANCOM MAC ADDRESS : 00:80:42:11:E7:3C
 Transmitting ARP-REQUEST... Reception of ARP-REPLY
 Transmitting TFTP-REQUEST to server 00:80:42:10:5B:C7, IP
192.168.41.1
 PACKET:871 - loaded $00100000..$0016CD13 (445716 bytes)

PowerBoot> ipmi_flsupd 100000 162b3b
  0 |#####################################| 100%
Firmware update finished waiting for IPMI reset
Firmware update successful
PowerBoot>
```

ipmi_request

DESCRIPTION Command to send values to the IPMI controller and display the response.

SYNTAX `ipmi_request <HEX> [<HEX>][<HEX>] ...`

PARAMETERS

HEX	Values in hexadecimal format

EXAMPLE

```
PowerBoot> ipmi_request 18 01
1c 01 00 01 81 63 84 01 5f 48 0e 00 04 08
PowerBoot>
```

lo

DESCRIPTION This command loads S-records from the console port to the absolute addresses of the memory which are specified by the S-records. The transfer speed is 9600 bit/s with 1S8D1S (1 start bit, 8 data bits, 1 stop bit) data size. No handshake protocol is used. Five tasks can be carried out:

- Loading S-records from the console port to the memory
- Modifying the storage address while loading S-records from the console port to the memory
- Verifying the S-records
- Modifying the storage address while verifying the S-records
- Displaying the number of errors which occurred during the last loading

SYNTAX `lo [<offset>] [v] [e] [<host commands>]`

PARAMETERS

host commands	Specifies a list of commands that will be sent to the host to initiate the data transfer
offset	Specifies the value added to the absolute addresses of the S-records. This allows to modify the storage address while loading or verifying.
v	Verifies the S-records, i.e. no data is loaded to the memory, but the memory contents and the S-records are compared
e	Displays the number of errors occurred during the last loading

EXAMPLE

```
PowerBoot> lo 100200

PowerBoot> md 100200 10
00100200:  54 68 69 73 20 69 73 20  61 20 74 65 73 74 00 00   This is
a test..
PowerBoot> _
```

netload

DESCRIPTION This command loads a binary image via TFTP (Trivial File Transfer Protocol) from a host acting as server for a specified Ethernet address. To create a connection to the server, the following two protocols are supported:

- RARP (Reverse Address Resolution Protocol)
- ARP (Address Resolution Protocol)

For these protocols the CPU board is referred to as target.

To receive the image from the server, NETLOAD broadcasts a RARP or an ARP request via Ethernet. If RARP is supported, NETLOAD waits for the Ethernet frame of the server responding. If ARP is supported, the specified server will be connected. Then the image is downloaded octet-wise via TFTP.The memory used for the binary image will be made available to the CPU if necessary.

SYNTAX `netload [-c controller] filename address [<Ethernet#>] [<targetIP# serverIP#>]`

PARAMETERS

`controller` Specifies the Ethernet interface
Possible values are: ETH1, ETH2 and ETH3

`filename` Absolute file name (including the path name) to access the file to be loaded. The length of the file name is limited to 128 characters. The file has to be a binary image in big endian notation.

`address` First byte of the memory used for the binary image. The size of the loaded file is the only parameter which determines the number of bytes written to the memory. The binary image must be adapted to the address, because after the loading no relocation will be done. address must be long-aligned.

`Ethernet#` Specifies the Ethernet address of the CPU board written in the following manner aa:bb:cc:dd:ee:ff.

`targetIP#` Defines the Internet IP address of the CPU board (target) written in the following manner ggg.hhh.iii.kkk.

`serverIP#` Defines the Internet IP address of the server accessed for downloading the TFTP file.

EXAMPLE 1 The file “test” is downloaded to 00100000_{16} on the CPU board with the Ethernet address from the BIB using RARP.

```
PowerBoot> netload test 100000
PHY-Device at 100MB/s and full duplex negotiated
WANCOM MAC ADDRESS : 00:80:42:11:67:C3
Transmitting RARP-REQUEST... Reception of RARP-REPLY
Transmitting TFTP-REQUEST to server 00:80:42:10:5B:C7, IP
192.168.41.1
PACKET:2049 - loaded $00100000..$00200000 (1048577 bytes)
PowerBoot>
```

EXAMPLE 2 The file “test” is downloaded to 00100000_{16} on the CPU board with the Ethernet address 00:80:42:0E:88:88 by using RARP.

```
PowerBoot> netload test 100000 00:80:42:0E:88:88
PHY-Device at 100MB/s and full duplex negotiated
WANCOM MAC ADDRESS : 00:80:42:0E:88:88
Transmitting RARP-REQUEST... Reception of RARP-REPLY
Transmitting TFTP-REQUEST to server 00:80:42:10:5B:C7, IP
192.168.41.1
PACKET:2049 - loaded $00100000..$00200000 (1048577 bytes)
PowerBoot>_
```

EXAMPLE 3 The file “test” is downloaded to 00100000_{16} on the CPU board with the Ethernet address 00:80:42:0E:88:88 by using ARP.

```
PowerBoot> netload test 100000 192.168.41.111 192.168.41.1
PHY-Device at 100MB/s and full duplex negotiated
WANCOM MAC ADDRESS : 00:80:42:0E:88:88
Transmitting ARP-REQUEST... Reception of ARP-REPLY
Transmitting TFTP-REQUEST to server 00:80:42:10:5B:C7, IP
192.168.41.1
PACKET:2049 - loaded $00100000..$00200000 (1048577 bytes)
PowerBoot>
```

netsave

DESCRIPTION This command saves a specified memory area via TFTP into a file located in a host system. The file cannot be created, i.e. it must already exist on the host with correct write permissions. NETSAVE overrides the content of the file. For creating a connection to the server, the following 2 procedures are supported:

- RARP (Reverse Address Resolution Protocol)
- and ARP (Address Resolution Protocol).

For these protocols the CPU board is referred to as target.

SYNTAX

```
netsave [-c controller] filename startAddr endAddr [<Ethernet#>]
[<targetIP#> <serverIP#>]
```

PARAMETERS

`controller` Specifies the Ethernet interface
Possible values are: ETH1, ETH2 and ETH3

`filename` Specifies the name of the file in the host system

`startAddr` Specifies the starting address of the memory area to be saved into the file on the host.

`endAddr` Specifies the end address of the memory area to be saved into the file on the host

`Ethernet#` Specifies the Ethernet address of the CPU board

`targetIP#` Defines the Internet IP address of the CPU board (target) written in the following manner ggg.hhh.iii.kkk

`serverIP#` Defines the Internet IP address of the server accessed for uploading the TFTP file

EXAMPLE 1

```
PowerBoot> netsave test 100000 1fffff
PHY-Device at 100MB/s and full duplex negotiated
WANCOM MAC ADDRESS : 00:80:42:11:67:C2
Transmitting RARP-REQUEST... Reception of RARP-REPLY
Transmitting TFTP-REQUEST to server 00:80:42:10:5B:C7, IP
192.168.41.1
PACKET:2049 - saved $00100000..$001FFFFF (1048576 bytes)
PowerBoot>
```

EXAMPLE 2

```
PowerBoot> netsave test 100000 1fffff 00:80:42:11:67:c2
PHY-Device at 100MB/s and full duplex negotiated
WANCOM MAC ADDRESS : 00:80:42:11:67:C2
Transmitting RARP-REQUEST... Reception of RARP-REPLY
Transmitting TFTP-REQUEST to server 00:80:42:10:5B:C7, IP
192.168.41.1
PACKET:2049 - saved $00100000..$001FFFFF (1048576 bytes)
PowerBoot>
```

EXAMPLE 3

```
PowerBoot> netsave test 100000 1fffff 192.168.41.111 192.168.41.1
PHY-Device at 100MB/s and full duplex negotiated
WANCOM MAC ADDRESS : 00:80:42:11:E7:24
Transmitting ARP-REQUEST... Reception of ARP-REPLY
Transmitting TFTP-REQUEST to server 00:80:42:10:5B:C7, IP
192.168.41.1
PACKET:2049 - saved $00100000..$001FFFFF (1048576 bytes)
PowerBoot>
```

Reset

The following commands are used to perform either a hardware or a software reset.

hreset

DESCRIPTION Command to initiate a hardware reset of the board.

SYNTAX `hreset`

PARAMETERS None

EXAMPLE

```
PowerBoot> hreset

Init GT MPSC0 as UART
Copy ROM to RAM
Started at phys. address: 0x1ff80000
Init DTLB/ITLB for block translation, enable MMU
Init L1-Icache
Init L1-Dcache
Init L2-Cache
Found IBM750FX at 667 MHz
Init exception vectors starting at address: 0x00000100
Onboard SDRAM      : 512MB RAM
Init device library
Discovery.0 is /
PCI.1 is /PCI@1
PCI.0 is /PCI@0
CompactPCI.0 is /PCI@0/CompactPCI@20
Ethernet 1:0080:42:11:67:C1
Ethernet 2:0080:42:11:67:C2
Ethernet 3:0080:42:11:67:C3
Found RTB-602

<<PowerBoot V4.00 for PowerCoreCPCI CPCI-690>>
<<        c by FORCE COMPUTERS          >>
<< (This is a nonreleased Version of  >>
<<  PowerBoot for PowerCore) >>
PowerBoot>
```

reset

DESCRIPTION Command to initiate a software reset.

SYNTAX `reset`

PARAMETERS None

EXAMPLE

```
PowerBoot> reset

Init GT MPSC0 as UART
Copy ROM to RAM
Started at phys. address: 0x1ff80000
Init DTLB/ITLB for block translation, enable MMU
Init L1-Icache
Init L1-Dcache
Init L2-Cache
Found IBM750FX at 667 MHz
Init exception vectors starting at address: 0x00000100
Onboard SDRAM      : 512MB RAM
Init device library
Discovery.0 is /
PCI.1 is /PCI@1
PCI.0 is /PCI@0
CompactPCI.0 is /PCI@0/CompactPCI@20
Ethernet 1:0080:42:11:67:C1
Ethernet 2:0080:42:11:67:C2
Ethernet 3:0080:42:11:67:C3
Found RTB-602

<<PowerBoot V4.xx for PowerCoreCPCI CPCI-690>>
<<        c by FORCE COMPUTERS          >>
<< (This is a nonreleased Version of  >>
<<  PowerBoot for PowerCore) >>
PowerBoot>
```

Miscellaneous

The following commands are used to:

- Fill a specified RAM memory area
- Search a memory area
- Compare two memory areas on a byte-organized level
- Erase a flash memory device or a part of it
- Start executing a binary image

bf

DESCRIPTION Command to fill a specified RAM memory area.

SYNTAX `bf <begin> <end> <value> [B|W|L|P]`

PARAMETERS

`begin` First byte of the memory area used for filling

`end` First byte of the memory area not used for filling

`value` Specifies the constant for filling. Possible constants are:

- Byte value
  A Byte constant followed by the character B, e.g. A5 B
- Word value
  A Word constant followed by the character W, e.g. A5B6 W
- Long value
  A long constant followed by the character L, e.g. A5B6C7D8 L
- String value
  An ASCII string followed by the character P. The ASCII string has to be set in inverted commas, e.g. “Hello” P

EXAMPLE

```
PowerBoot> bf 100000 200000 FFFF
PowerBoot>
```

bs

DESCRIPTION Command to search a memory area for a byte, a word, or a long constant or for a user defined ASCII string. There are two tasks which can be done by means of bs:

- Displaying the locations where the searched byte, word, long constant, or ASCII string is found
- Displaying the locations where the searched byte, word, long constant, or ASCII string is not found

SYNTAX `bs <begin> <end> [/]<value> [B|W|L|P]`

PARAMETERS

`begin` Specifies the start address of the memory area which is searched. , i.e. the first byte of the memory area which is read

`end` Specifies the end address of the memory area which is searched, i.e. the first byte of the memory area which is not read

`value` Specifies the constant to search for. Possible constants are:

- byte value
  A byte constant followed by the character B, e.g. 3F B
- word value
  A word constant followed by the character W, e.g. 3F3F W
- long value
  A long constant followed by the character L, e.g. EFEFEFEF L
- string value
  An ASCII string followed by the character P. The ASCII string has to be set in inverted commas, e.g. “Force” P

EXAMPLE 1 A 4-KByte memory from $00002D00_{16}$ .. $00003D00_{16}$ is searched for the byte constant $3F_{16}$. The byte constant is found at addresses $00002E01_{16}$ and $00002E05_{16}$.

```
PowerBoot> bs 2D00 3D00 3F B
Search: 00002E01 = 3F
Search: 00002E05 = 3F
PowerBoot>
```

EXAMPLE 2 An 8-KByte memory from $00004C00_{16}$ .. $0006C00_{16}$ is searched for the long constant $EFEFEFEF_{16}$. However, PowerBoot displays the address at which the long constant is not found ($00004C10_{16}$) and the value which is found instead.

```
PowerBoot>bs 4C00 6C00 /EFEFEFEF L
Search: 00004C10 = EEEEEEEE
PowerBoot>
```

bv

DESCRIPTION This command compares two memory areas on a byte-organized level. If the compared memory areas are not equal, the different values and the memory location will be displayed. An overlap of memory areas is not allowed.

SYNTAX `bv <begin> <end> <destination>`

PARAMETERS

`begin` Specifies the start address of the source memory area, i.e. the first byte of the source memory area which is read

`end` Specifies the end address of the source memory area, i.e. the first byte of the source memory area which is not read

`destination` Specifies the start address of the destination memory area, i.e. the first byte of the destination memory area which is compared with the source memory

EXAMPLE 1 A 4-KByte memory from $00002D00_{16}$..$00003D00_{16}$ is compared with a destination memory starting at address $00005E00_{16}$. There is no output since the two compared memory areas are identical.

```
PowerBoot>bv 2D00 3D00 5E00
PowerBoot>
```

EXAMPLE 2 An 8-KByte memory from $00004C00_{16}$..$00006C00_{16}$ is compared with a destination memory starting at address $00007C00_{16}$. The two compared memory areas are not identical at the locations $00004C10_{16}$ and $00007C10_{16}$.

```
PowerBoot>bv 4C00 6C00 7C00
Verify:00004C10 = EE 00007C10 =3F
PowerBoot>
```

ferase

Caution

Data loss
Using this command without specifying an area will erase the whole flash memory.
If you do not want to erase the whole flash memory, specify an area by adding the destination offset and the number of bytes to be erased.

DESCRIPTION This command erases a whole flash memory device or a specified area of the flash memory device. The specified area must exactly match the page boundaries of the flash memory device. If a user flash consists, for example, of a 28F008 device (1 Mbit * 8 bit), the minimum size of the erasable area is 64 KByte, i.e. the size of one sector.

There are two tasks which can be done by means of FERASE:

- Erasing a whole flash memory device
- Erasing a specified area of a flash memory device

SYNTAX `ferase <flashDevice> [<flashOffset> [<length>]]`

PARAMETERS

`flashDevice` Specifies the name of the flash memory device to be erased. If only flashDevice is defined, the whole flash memory device will be erased.

`flashOffset` Specifies the destination offset within the flash memory device area to be erased.

`length` Specifies the number of bytes within the flash memory device area to be erased.

EXAMPLE 1 Suppose you want to erase boot flash 1 which is write-protected via switch setting. Since FERASE checks whether the flash is write-protected, the following message appears:

```
PowerBoot> ferase BOOT_FLASH1
Do not reprogram BOOT_FLASH1, this would destroy the booter
Device is write protected
```

EXAMPLE 2 Only a specified area of user flash 1 is erased:

```
PowerBoot> ferase USER_FLASH1 30000 40000
Erasing flash device...   done.
PowerBoot> _
```

go

DESCRIPTION This command starts executing a binary image located in the DRAM memory. After entering go you exit from PowerBoot. At the end of the binary image enter the opcode blr to return to PowerBoot.

SYNTAX `go <address>`

PARAMETERS

`address` Specifies the starting address of the binary image

EXAMPLE

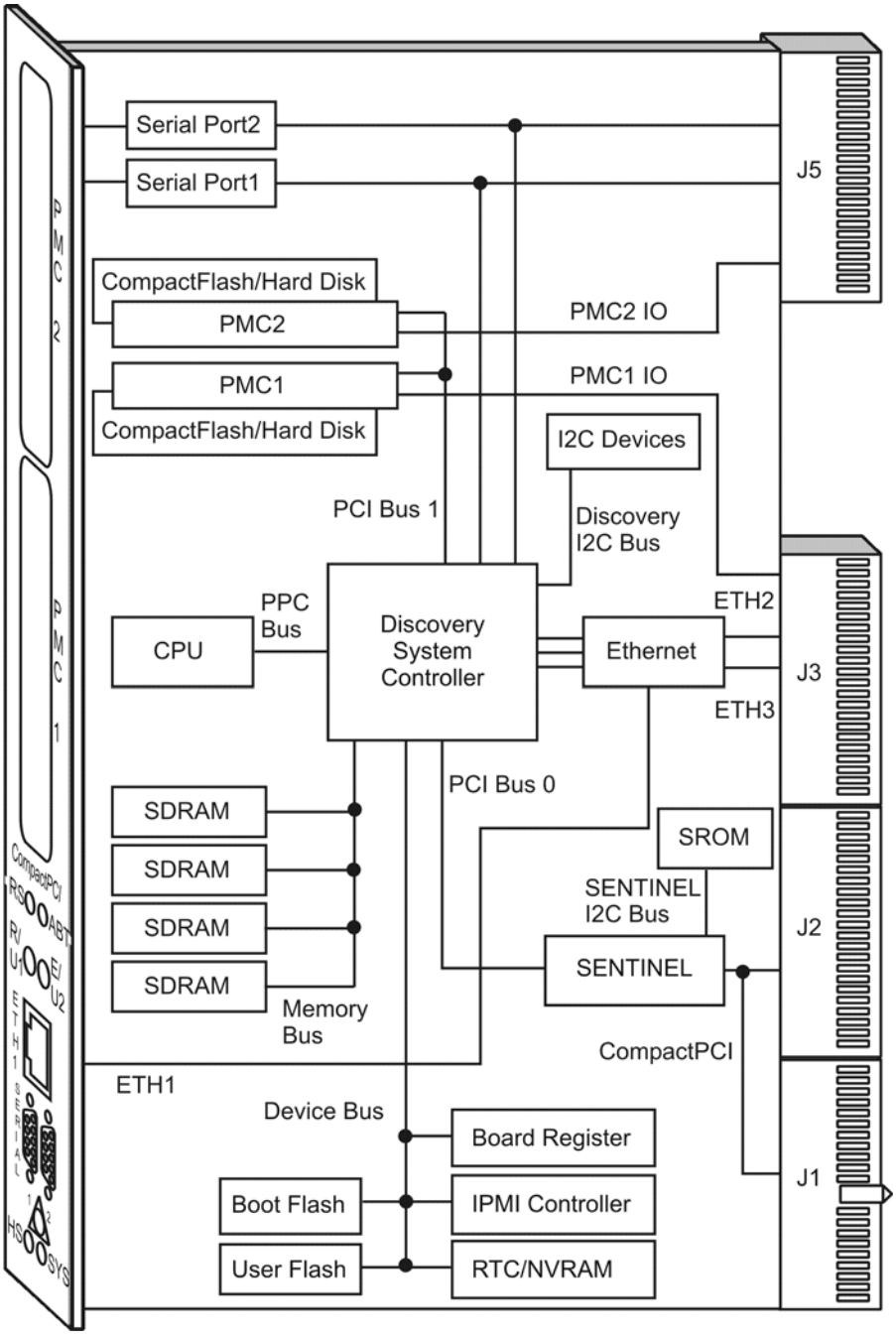
```
PowerBoot> go 10000
```

5

Buses

Block Diagram

The block diagram shows how the board's devices work together and which data paths they use.



System Controller

The Discovery system controller is a single chip solution for PowerPC based systems. The chip has a five bus architecture with:

- 64-bit / 133 MHz interface to CPU
- 64-bit and 8bit for ECC / 133 MHz interface to SDRAM
- Two 64-bit / 66 MHz PCI interfaces
- 32-bit interface to devices

Ethernet Ports

The system controller contains three 10Base-T / 100Base-T Ethernet controllers which are able to operate in either full duplex or half-duplex mode.

Ethernet port 1 is routed to a standard RJ-45 connector at the front panel. Ethernet ports 2 and 3 are routed to connector J3. On the standard variants the magnetics for port 2 and 3 are implemented on the RTB.

On the front-panel, a green LED is available which indicates an Ethernet link. For further information on Ethernet LEDs, refer to “LEDs” page 3-6.

Serial Ports

The system controller includes two multi-protocol serial controllers (MPSC) which are configured for Universal Asynchronous Receiver-Transmitter (UART) mode. The controller can operate simultaneously and have dedicated serial direct memory access (SDMA) engines to transfer data between port and memory. Receive and transmit clock are generated by the internal baud rate generator.

The UART ports are connected with two RS232 connectors at the front-panel and two on the RTB.

Note: The serial port can only be used either via front panel or via RTB, but not simultaneously.

Watchdog

The system controller includes a bi-level watchdog timer which monitors the processor activity. The watchdog timer generates a non-maskable interrupt (NMI) after the first time-out period and a system reset pulse after the second time-out period.

The watchdog can be enabled and disabled via Watchdog Configuration register by a write sequence of 01b followed by 10b to field CTL1. If enabled, the watchdog is a free running counter that needs to be triggered periodically in order to prevent expiration. Watchdog service is done by writing 01b followed by 10b to field CTL2 of the Watchdog configuration register. For further information, refer to the GT-64260A System Controller for PowerPC Processors data sheet.

The watchdog is running with the system controller's core frequency and time-out ranges are from 1.92 µs. to 32 s. Both time-out values are software programmable and can be changed during watchdog operation.

Note: PowerBoot implements time-out ranges from 1 to 32 s only.

PPC Bus

The 60x PowerPC bus connecting the system controller with the CPU is a 64-bit bus running at 133 MHz.

The PPC/CPCI-690 is equipped with one of the following CPUs:

- PowerPC 750FX
  This CPU has an internal 64-KByte L1-cache and 512 KByte of L2-cache, both with 256-bit wide cache paths. Both caches are running at the full processor core frequency of 667 MHz or higher.
- PowerPC 750GX
  This CPU has an internal 64-KByte L1-cache and 1 MByte of L2-cache, both with 256-bit wide cache paths. Both caches are running at the full processor core frequency of up to 1 GHz.

Memory Bus

The system controller has a 3.3V SDRAM interface running at frequencies up to 133 MHz. The interface consists of a 15-bit wide address bus and a 64-bit data bus plus eight additional bits for error checking and correction (ECC). Two bits of the address bus are used for bank select, 13 bits are used for row and column address.

The board is equipped with PC133 compliant unregistered SDRAM memory devices and has a maximum capacity from 256 MByte up to 1 GByte. If the maximum capacity is 256 MByte, one bank is equipped with 256 MBit devices and if it is 1 GByte, two banks are equipped with 512 MBit devices using dense-pack three level stacking technology.

Table 15: *Memory Bus Characteristics*

Memory Characteristics	Memory Capacity below 1 GByte	Memory Capacity 1 GByte
Memory speed	133 MHz	100 MHz
CAS support	CAS 3	CAS 2

PCI Bus 0

PCI bus 0 is running at 32-bit and 33 MHz.

Table 16: *Devices on PCI Bus 0*

PCI Device	Device Type	ID SEL	Dev. No.	PCI IRQ	REQ/GNT
Discovery PCI Interface 0	GT-64260A	31	21	INTA#	Internal
PCI-to-CompactPCI bridge	SENTINEL	30	20	INTA# .. INTD#	4

Table 17: *PCI Bus 0 Interrupt Routing*

PCI Device	Device Interrupt	Interrupt Received by System Controller
Discovery PCI Interface 0	INTA#	INTD#
PCI-to-CPCI Bridge SENTINEL	INTA#	INTC#
	INTB#	INTD#
	INTC#	INTA#
	INTD#	INTB#

SENTINEL

Motorola SENTINEL is used as PCI-to-CompactPCI universal bridge that enables the board to work as either CompactPCI system (host) or peripheral board. SENTINEL is capable to run in either 3.3V or 5V VIO environment. For further information, refer to “Installation in a Powered System Supporting Hot Swap” page 2-18.

The table below lists all fields in the SPROM of the SENTINEL, the contents loaded into the registers at power up.

Note: Invalid SPROM contents may cause malfunction of the board.

Table 18: *SENTINEL SPROM Contents*

Register Name	Offset	Configurable	Default Value	Comment
Magic word	0_{16}	No	1146_{16}	-
Downstream IR mask	2_{16}	No	$FFFF_{16}$	-
Upstream IR mask	6_{16}	No	$FFFF_{16}$	-
Downstream BAR 0 DSR setup	8_{16}	Yes	$FFFFF000_{16}$	Maps 4 KByte DSR registers by default. Larger values provide access to on-board resources. A value less than FFFFF000 is not allowed.
Downstream BAR 0 DSR translate	C_{16}	Yes	00000000_{16}	-
Downstream BAR 2 DSR setup	10_{16}	Yes	$FFF00000_{16}$	This defines a default PCI window of 1 MByte.
Downstream BAR 2 DSR translate	14_{16}	Yes	00000000_{16}	-
Downstream BAR 3 DSR setup	18_{16}	Yes	00000000_{16}	-
Downstream BAR 3 DSR translate	$1C_{16}$	Yes	00000000_{16}	-
Downstream BAR 4 DSR setup	20_{16}	Yes	00000000_{16}	-
Downstream BAR 4 DSR translate	24_{16}	Yes	00000000_{16}	-
Upstream BAR 2 DSR setup	28_{16}	Yes	$FFF00000_{16}$	Defines a default PCI window of 1 MByte.
Upstream BAR 2 DSR translate	$2C_{16}$	Yes	00000000_{16}	-
Memory base preload	30_{16}	No	0001_{16}	These two settings effectively disable the memory window (base>limit).
Memory limit preload	32_{16}	No	0000_{16}	-

Table 18: *SENTINEL SPROM Contents*

Register Name	Offset	Configurable	Default Value	Comment
I_2O outbound post list interrupt mask	34_{16}	No	$FFFFFFFF_{16}$	-
Subsystem vendor ID preload	38_{16}	No	1146_{16}	-
Subsystem ID preload	$3A_{16}$	No	0690_{16}	Board-specific subsystem ID
I_2O setup mask	$3C_{16}$	No	00000000_{16}	-
Secondary min. grant preload	42_{16}	No	01_{16}	-
Secondary max. latency preload	43_{16}	No	01_{16}	-
Primary min. grant preload	46_{16}	No	01_{16}	-
Primary max. latency preload	47_{16}	No	01_{16}	-
Upstream BAR 3 setup	48_{16}	Yes	00000000_{16}	-
Upstream BAR 3 translate	$4C_{16}$	Yes	00000000_{16}	-
Chip control 0	54_{16}	Yes	0000_{16}	-
Chip control 1	56_{16}	Yes	0000_{16}	-
Arbiter control/status	$5A_{16}$	Yes	8000_{16}	-
Primary SERR# control	$5D_{16}$	Yes	07_{16}	-
Secondary SERR# control	$5F_{16}$	Yes	07_{16}	-
GPIO control	60_{16}	No	$F0_{16}$	-
GPIO write data	61_{16}	No	80_{16}	-
GPIO IR mask	63_{16}	No	FF_{16}	-
User capability ID	68_{16}	Yes	00_{16}	-
MSI next item pointer	71_{16}	Yes	00_{16}	User capability is disabled.

Table 18: *SENTINEL SPROM Contents*

Register Name	Offset	Configurable	Default Value	Comment
Message control	72_{16}	Yes	0000_{16}	-
Scratchpad	76_{16}	Yes	0000_{16}	-
Primary programming interface	79_{16}	Yes	00_{16}	-
Primary sub class code	$7A_{16}$	Yes	80_{16}	This field defines “Other Bridge”.
Primary base class code	$7B_{16}$	Yes	06_{16}	This field defines “Bridge Device”.
Secondary programming interface	$7D_{16}$	Yes	00_{16}	-
Secondary sub class code	$7E_{16}$	Yes	80_{16}	This field defines “Other Bridge”.
Secondary base class code	$7F_{16}$	Yes	06_{16}	This field defines “Bridge Device”.

PCI Bus 1

PCI bus1 is capable to run at 64 bit and 66 MHz.

Table 19: *Devices on PCI Bus 1*

PCI Device	Device Type	ID SEL	Dev. No.	PCI IRQ	REQ/GNT
Discovery PCI interface 1	GT-64260A	31	21	INTA#	Internal
PMC slot 1	PMC1	30	20	INTA# .. INTD#	0
PMC slot 1 (optional secondary PCI agent)	PMC1	29	19	INTA# .. INTD#	1
PMC slot 2	PMC2	28	18	INTA# .. INTD#	2
PMC slot 2 (optional secondary PCI agent)	PMC2	27	17	INTA# .. INTD#	3

Table 20: *PCI Bus 1 Interrupt Routing*

PCI Device	Device Interrupt	Interrupt Received by System Controller
Discovery PCI Interface 1	INTA#	INTD#
PMC Slot 1	INTA#	INTC#
	INTB#	INTD#
	INTC#	INTA#
	INTD#	INTB#
PMC Slot 1 - Secondary PCI Agent (ProcessorPMC specification)	INTA#	INTB#
	INTB#	INTC#
	INTC#	INTD#
	INTD#	INTA#
PMC Slot 2	INTA#	INTB#
	INTB#	INTC#
	INTC#	INTD#
	INTD#	INTA#

Table 20: *PCI Bus 1 Interrupt Routing*

PCI Device	Device Interrupt	Interrupt Received by System Controller
PMC Slot 2 - Secondary PCI Agent (ProcessorPMC specification)	INTA#	INTA#
	INTB#	INTB#
	INTC#	INTC#
	INTD#	INTD#

PMC 1

The PMC 1 interface is a 3.3V compliant PCI interface with its user I/O signals routed towards CompactPCI connector J3. PMC slot 1 can be used for Motorola's PMC-8260 module. Non-Monarch mode ProcessorPMC modules and Ramix IDE modules (on-module primary IDE interface) are supported.

PMC 2

The PMC 2 interface is a 3.3V compliant PCI interface with its user I/O signals routed towards CompactPCI connector J5. PMC slot 2 can be used for Motorola's PMC-8260 module. Non-Monarch mode ProcessorPMC modules and Ramix IDE modules (on-module primary IDE interface as well as secondary interface via RTB) are supported.

Device Bus

The device bus is a 32-bit wide multiplexed bus that is used for connecting boot and user flash memory, RTC / NVRAM, IPMI controller and board registers that are implemented in logic.

Boot Flash

Boot flash memory consists of two 8-bit wide 512 KByte flash devices. The boot block memory range is write-protected (default) and can be reprogrammed on-board (see “fprog” page 4-39). Write protection can be removed and the boot flash address space can be exchanged via switch setting (see Table 7 “Switch Settings” page 2-14).

Device Type	Boot Flash	Memory Address Space
M29W040B-90K1	0	FFF00000..FFF7FFFF
	1	FFF80000..FFFFFFFF

User Flash

The user flash consists of four 8-bit wide flash devices offering up to 64 MByte user flash memory. 32 MBytes are implemented on the orderable variants per default. 64 MBytes of user flash memory are available as an assembly option.

The user flash memory is write-enabled by default and can be reprogrammed on-board (see “Programming the User Flash” page 4-8). Write protection can be enabled via switch setting (see Table 7 “Switch Settings” page 2-14).

Device Type	Memory Address Space
E28F640J3A-120 (64 Mbit) E28F128J3A-150 (128 Mbit)	32 MByte: F4000000..F5FF.FFFF (default) 64 MByte: F4000000..F7FF.FFFF

RTC/NVRAM

The board carries a device where 8-bit wide non-volatile SRAM (NVRAM) for backup functionality of 32 KByte size and a real time clock (RTC) are implemented.

A SNAPHAT package with the battery and a 32.768 Hz crystal is mounted on top of the device.

Device Type	Memory Address Space
M48T35AV-10MH1 M4T32-BR12SH	F0100000..F01FFFF7 (NVRAM / RTC) F01FFFF8..F01FFFFF (Battery)

IPMI Controller

The IPMI unit consists of the micro controller Vitesse VSC215, 512 KByte SRAM, 3 MByte flash memory and 8 KByte RAM in the controller. It can be accessed via the device bus.

ICMB is connected to the IPMI controller via an RS485 interface. Furthermore, four I2C buses are connected to the IPMI controller with the following buses / devices:

- I2C bus 0: Connected to IPMB0
- I2C bus 1: Connected to IPMB1
- I2C bus 2: Connected to the ID-ROM of the PPC / CPCI-690 which contains the board information block (BIB) and of the attached boards / modules
- I2C bus 3: Connected to on-board temperature sensor

IPMB0 signals are available at CompactPCI connector J1. IPMB1 signals are available at CompactPCI connector J2 and at J5 via ACC / RTB-602. ICMB signals are available at J5 via ACC / RTB-602.

The operational mode (BMC / PM) can be set manually via switch (see Table 7 “Switch Settings” page 2-14). If the manual selection is disabled, automatic selection of PM or Base Board Management Controller (BMC) functionality is activated.

IPMI related commands are also provided by PowerBoot:

- ipmi_info (see page 4-33)
  Displays information on the on-board IPMI controller
- ipmi_request (see page 4-41)
  Command to send values to the IPMI controller
- ipmi_flsupd (see page 4-40)
  Starts an IPMI flash update

For information on the IPMI firmware and the supported IPMI commands, refer to the *IPMI Reference Guide* available via SMART.

Temperature Sensor

The board provides the temperature sensor LM75.

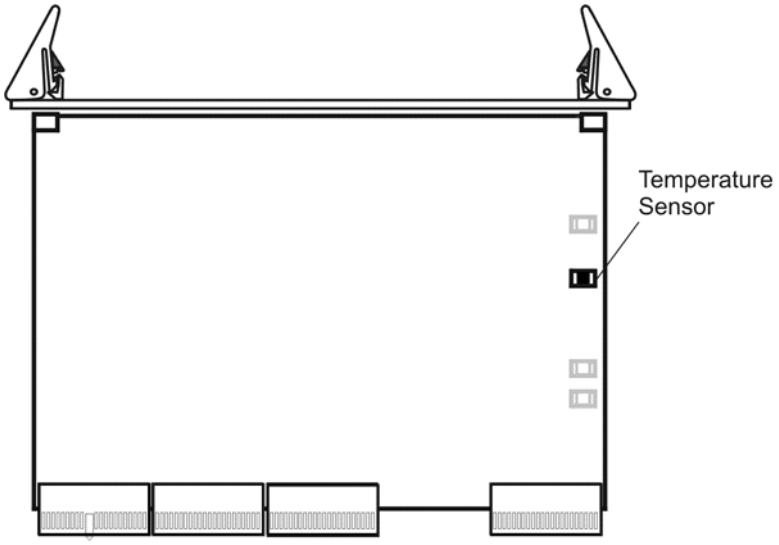


Figure 17: *Location of Temperature Sensor*

The temperature sensor is connected to the IPMI controller's I2C bus 3 and can be configured and read via the IPMI controller. The Motorola IPMI firmware already provides sensor data records (SDRs) containing sensor configuration data such as temperature threshold values. To check how many SDRs are provided, use the IPMI command "Get Device SDR Info" together with the API of the Motorola VxWorks IPMI driver. To read the SDRs into your system management software, use the IPMI command "Get Device SDR". To add the provided SDRs to the repository use the IPMI command "Add SDR".

The temperature values can be read via the Master Write-Read I2C IPMI command and the I2C slave address together with the API of the Motorola VxWorks IPMI driver and via the Get Sensor Reading command.

For further information, refer to the *VxWorks 5.4/Tornado 2.0 BSP Rel. 2-1 Installation Guide and Release Notes* and the IPMI Specification V. 1.0.

I2C Bus Slave Addresses

The table below shows the I2C bus slave addresses of devices attached to the I2C buses 2 and 3. These addresses are e.g. necessary to get information on the temperature sensor via the IPMI command Master Write / Read I2C.

Table 21: *Slave Addresses of Devices Attached to IPMI Controller's I2C Buses*

I2C Bus	I2C Bus Slave Address	Description
2	$1010000x_2 / A0_{16}$	ID-ROM CPCI-690 XICOR X24LC128 serial E^2PROM
2	$1010010x_2 / AA_{16}$	RTB ID-ROM CPCI-690 XICOR X24C02 serial E^2PROM
2	$1010111x_2 / AE_{16}$	IPMI ID-ROM CPCI-690 XICOR X24C02 serial E^2PROM
3	$1001000x_2 / 90_{16}$	Temperature sensor LM75

Available IPMI Drivers

Motorola offers an IPMI driver to access the IPMI controller. It is part of the VxWorks 5.4 / Tornado 2.0 BSP Rel. 2-1. For further information, refer to the *VxWorks 5.4/Tornado 2.0 BSP Rel. 2-1 Installation Guide and Release Notes* and the IPMI Specification V. 1.0.

Local I2C Buses

The board provides two local I2C buses, the Discovery and the SENTINEL I2C bus.

Discovery I2C Bus

The Discovery I2C bus provides access to board information memory devices of the board itself and attached accessories like an RTB or PMC modules.

Table 22: *On-Board I2C Devices*

Device	Type	Size in KByte
Board ID-ROM	24LC128	128
IPMI ID-ROM	24C02	2

Table 23: *Off-Board I2C Devices*

Device	Type	Size in KByte	Description
PIB1	Depends on used PMC module	-	PMC 1 information block
PIB2	Depends on used PMC module	-	PMC 2 information block
RTB ID-ROM	24C02	2	-

This I2C bus can be accessed via:

- IPMI controller and Master write / read command
- JTAG connector
- I2C interface integrated in system controller

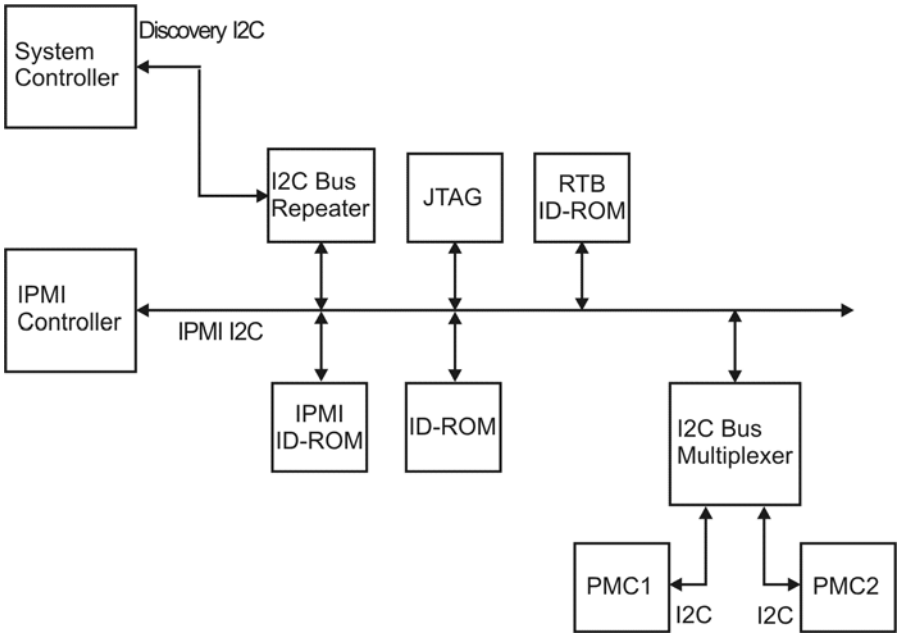


Figure 18: *Connection to I2C Devices*

SENTINEL I2C Bus

This bus connects the SENTINEL with the SROM for configuration registers. A 24C04 device with 4 KBytes is used.

6

Maps and Registers

Overview

This section gives an overview on the memory maps and describes all board-specific registers.

Table 24: *Register Overview*

Register	Description
Last Reset Status register 1	Page 6-14
Last Reset Status register 2	Page 6-15
LED Control register	Page 6-10
Memory Configuration register	Page 6-13
Miscellaneous Status register	Page 6-11
Software Reset register	Page 6-11
Switch Status register	Page 6-12

Address Maps

The address map is determined by the system controller's internal structure. Each address window must have an address space larger than 1 MByte.

Caution

Board malfunction
Accesses to the reserved address areas and overlapping of address windows result in unpredictable board behavior.
Only access address areas not marked as reserved. Make sure that address windows do not overlap.

Table 25: *Processor Memory Address Map*

Base Address	End Address	Size	Device	Bus Width
00000000_{16}	$7FFFFFFF_{16}$	2048 MBytes	Main memory	64 bit
80000000_{16}	$EF7FFFFF_{16}$	1784 MBytes	PCI memory 1)	64 bit
$EF800000_{16}$	$EFFFFFFF_{16}$	8 MBytes	PCI I/O space 2)	64 bit
$F0000000_{16}$	$F0000007_{16}$	8 Bytes	Board registers	8 bit
$F0000008_{16}$	$F00FFFFF_{16}$	-	Reserved	-
$F0100000_{16}$	$F0107FFF_{16}$	32 KBytes	RTC / NVRAM	8 bit
$F0108000_{16}$	$F01FFFFF_{16}$	-	Reserved	-
$F0200000_{16}$	$F020000F_{16}$	16 Bytes	IPMI controller	8 bit
$F0200000_{16}$	$F0200001_{16}$	2 Bytes	KCS0 3)	8 bit
$F0200008_{16}$	$F0200009_{16}$	2 Bytes	KCS1 3)	8 bit
$F0200010_{16}$	$F0FFFFFF_{16}$	-	Reserved	-
$F1000000_{16}$	$F100FFFF_{16}$	64 KBytes	Discovery registers	64 bit
$F1010000_{16}$	$F3FFFFFF_{16}$	-	Reserved	-
$F4000000_{16}$	$F7FFFFFF_{16}$	64 MBytes	User flash	8 bit
$F8000000_{16}$	$FFEFFFFF_{16}$	-	Reserved	-
$FFF00000_{16}$	$FFFFFFFF_{16}$	1 MBytes	Boot flash 4)	8 bit

1) Contains up to four PCI memory windows for each PCI bus segment
2) Contains one PCI I/O window for each PCI bus segment
3) Interface of IPMI controller
4) Access to boot flash 1 and 2 is decoded by external logic

The PCI memory address map only differs from the processor memory map in the two PCI address windows.

Table 26: *PCI Memory Address Map*

Base Address	End Address	Size	Device	Bus Width
00000000_{16}	$7FFFFFFF_{16}$	See Table 25 “Processor Memory Address Map”		
80000000_{16}	$EF7FFFFF_{16}$	1784 MByte	PCI-to-PCI memory	64 bit
$EF800000_{16}$	$EFFFFFFF_{16}$	8 MByte	PCI-to-PCI I/O space	64 bit
$F0000000_{16}$	$FFFFFFFF_{16}$	See Table 25 “Processor Memory Address Map”		

Table 27: *I2C Address Map*

Bus	Address	Device
SENTINEL I2C	$1010000x_2$ / $A0_{16}$	Serial ROM for SENTINEL registers
Local I2C	$1010000x_2$ / $A0_{16}$	Board information block
Local I2C	$1010101x_2$ / AA_{16}	RTB information block
Local I2C	$1010111x_2$ / AE_{16}	IPMI information block

Interrupt Map

The PowerPC 750FX/750GX processor provides three low active interrupt inputs that are driven by the system controller:

- MCP# - machine check interrupt (NMI#)
- SMI# - system management interrupt
- INT# - standard CPU interrupt

The MCP# interrupt has highest priority followed by SMI# and INT#.

Interrupt Controller

The system controller includes an interrupt controller that routes internal and external interrupt requests to both, the processor and the PCI buses 1 or 2.

Each of the internal or external interrupt sources can be enabled, disabled, and routed completely by software. The interrupt controller can be configured via the Interrupt Controller registers (for further information, refer to the GT-64260A System Controller for PowerPC Processors datasheet).

System Controller MPP Configuration

The system controller contains 32 multi purpose pins (MPPs). Each one can be assigned to a different functionality through the MPP Control registers (for further information, refer to the GT-64260A System Controller for PowerPC Processors datasheet). The MPP pins can be used as hardware control signals to the system controller's different interfaces or as general purpose ports (GPP).

Each of the GPP pins can be assigned to act as a general purpose input or output pin. They can be used to register external interrupts when assigned as input pin. The GPP pins can be configured via the General Purpose Port registers. The required CPCI-690 settings for the registers are shown in the table below.

Table 28: *MPP and GPP Register Settings*

Register	Address	Data
MPP Control 0	$F100F000_{16}$	88888888_{16}
MPP Control 1	$F100F004_{16}$	00669977_{16}

Table 28: *MPP and GPP Register Settings*

Register	Address	Data
MPP Control 2	$F100F008_{16}$	00000000_{16}
MPP Control 3	$F100F00C_{16}$	00000000_{16}
GPP I/O Control	$F100F100_{16}$	00000000_{16}
GPP Level Control	$F100F110_{16}$	$FFC7C000_{16}$

The following table gives an overview on the MPP configuration for the board.

Table 29: *System Controller MPP Configuration*

MPP	Function	Description	Direction	Polarity
[31]	GPP[31]	PCI bus 1 interrupts	Input	Active low
[30]	GPP[30]			
[29]	GPP[29]			
[28]	GPP[28]			
[27]	GPP[27]	PCI bus 0 interrupts	Input	Active low
[26]	GPP[26]			
[25]	GPP[25]			
[24]	GPP[24]			
[23]	GPP[23]	Watchdog interrupt (in)	Input	Active low
[22]	GPP[22]	Reserved	Input	Active low
[21]	GPP[21]	IPMI interrupts	Input	Active high
[20]	GPP[20]			
[19]	GPP[19]			
[18]	GPP[18]	Ejector interrupt	Input	Active low
[17]	GPP[17]	CompactPCI interrupt DEG#	Input	Active low
[16]	GPP[16]	CompactPCI interrupt FAL#	Input	Active low
[15]	GPP[15]	Reserved	Input	Active low
[14]	GPP[14]	Reserved	Input	Active low
[13]	INT#[1]	PowerPC interrupt SMI#	Output	Active low
[12]	INT#[0]	PowerPC interrupt MCP#	Output	Active low
[11]	WDE#	Watchdog expired reset	Output	Active low
[10]	WDNMI#	Watchdog interrupt	Output	Active low
[9]	REQ0[4]#	PCI bus 0 REQ#	Input	Active low

Table 29: *System Controller MPP Configuration (cont.)*

MPP	Function	Description	Direction	Polarity
[8]	GNT0[4]#	PCI bus 0 GNT#	Output	Active low
[7]	REQ1[3]#	PCI bus 1 REQ#	Input	Active low
[6]	GNT1[3]#	PCI bus 1 GNT#	Output	Active low
[5]	REQ1[2]#	PCI bus 1 REQ#	Input	Active low
[4]	GNT1[2]#	PCI bus 1 GNT#	Output	Active low
[3]	REQ1[1]#	PCI bus 1 REQ#	Input	Active low
[2]	GNT1[1]#	PCI bus 1 GNT#	Output	Active low
[1]	REQ1[0]#	PCI bus 1 REQ#	Input	Active low
[0]	GNT1[0]#	PCI bus 1 GNT#	Output	Active low

External Interrupt Sources

The following GPP pins of the system controller are used to connect external interrupts to the internal interrupt controller. For a correct functionality, the GPP pins must be configured according to “System Controller MPP Configuration” page 6-7.

Table 30: *External Interrupt Sources*

GPP	Interrupt	Description	Source	Polarity
[31] [30] [29] [28]	PCI1_INTD# PCI1_INTC# PCI1_INTB# PCI1_INTA#	Shared PCI interrupts from PCI bus 1	See “PCI Bus 1” page 5-12	Active low
[27] [26] [25] [24]	PCI0_INTD# PCI0_INTC# PCI0_INTB# PCI0_INTA#	Shared PCI interrupts from PCI bus 0	See “PCI Bus 0” page 5-8	Active low
[23]	WDNMI#	NMI# interrupt from system controller’s internal watchdog	Watchdog	Active low
[21] [20] [19]	IPMI_IRQ2 IPMI_IRQ1 IPMI_IRQ0	Interrupts from the on-board IPMI controller	IPMI controller	Active high

Table 30: *External Interrupt Sources (cont.)*

GPP	Interrupt	Description	Source	Polarity
[18]	EJECT_INT#	Interrupt from the hot-swap switch that is integrated into the lower handle of the front panel. The interrupt only is asserted if the board is installed into a PSB backplane without CompactPCI bus.	Hot-swap switch	Active low
[17]	CPCI_DEG#	Power supply status interrupt from the CompactPCI backplane	CompactPCI	Active low
[16]	CPCI_FAL#	Power supply status interrupt from the CompactPCI backplane	CompactPCI	Active low

Register

The board provides a set of 8-bit wide board-specific registers. The registers can be used by software to read board-specific status and configuration data. Software can also write to some registers to generate a software reset or to set the two user LEDs.

LED Control Register

The board provides two bicolored (red / green) LEDs and a hot-swap LED. The bicolored LEDs can be configured as user LEDs or as hardware LED to display active reset or Ethernet 1, 2 and 3 link activity.

Note: If the board is plugged into a PSB backplane without PCI bus, opening the handle generates an EJECT_INT# interrupt which indicates that the board is to be removed from the backplane. Therefore, bit 6 has to be set to 1.

Table 31: *LED Control Register*

Base Address: F0000000$_{16}$		**Offset: 00$_{16}$**	
Bit	**Signal**	**Description**	**Access**
1..0	USER_LED1	Controls User LED 1 00: OFF 01: Green 10: Red 11: Reset LED (default)	r / w
3..2	USER_LED2	Controls User LED 2 00: OFF 01: Green 10: Red 11: Ethernet LED (default)	r / w
5..4	ETH_LEDx	Controls Ethernet LED 00: Ethernet 1(default) 01: Ethernet 2 10: Ethernet 3 11: Reserved	r / w
6	HS_LED	Controls blue hot-swap LED. 0: OFF (default) 1: Blue	r / w

Table 31: *LED Control Register (cont.)*

Base Address: $F0000000_{16}$		Offset: 00_{16}	
Bit	**Signal**	**Description**	**Access**
7	Reserved	Reserved 0: Default	r / w

Software Reset Register

The Software Reset register enables you to reset the board via software.

Table 32: *Software Reset Register*

Base Address: $F0000000_{16}$		Offset: 01_{16}	
Bit	**Signal**	**Description**	**Access**
7..0	SW_RESET	Type of software initiated reset 11_{16}: Board Reset	r / w

Miscellaneous Status Register

The Miscellaneous Status register provides information on whether PMC modules or an RTB is installed on the board.

Table 33: *Miscellaneous Status Register*

Base Address: $F0000000_{16}$		Offset: 02_{16}	
Bit	**Signal**	**Description**	**Access**
0	RTB_PRESENT	Rear transition board 0: Not present 1: Present	r
1	PMC1_PRESENT	ProcessorPMC in PMC slot 1 0: Not present 1: Present	r
2	PMC2_PRESENT	ProcessorPMC in PMC slot 2 0: Not present 1: Present	r
7..3	Reserved	Reserved 00000: Default	r

Switch Status Register

The Switch Status register shows the status of the configuration switches SW2 and SW4.

Table 34: *Switch Status Register*

Base Address: F0000000$_{16}$		**Offset: 03$_{16}$**	
Bit	**Signal**	**Description**	**Access**
0	SW2_BOOT_SELECT	0: Boot flash 0 is selected (switch is OFF) 1: Boot flash 1 is selected (switch is ON)	r
1	SW2_BOOT_WE	0: Boot flash write protected (switch is OFF) 1: Boot flash write enabled (switch is ON)	r
2	SW2_FLASH_WP	0: User flash write enabled (switch is OFF) 1: User flash write protected (switch is ON)	r
3	SW2_PCI0_RST_DIS	0: PCI bus 0 reset enabled (switch is OFF) 1: PCI bus 0 reset disabled (switch is ON)	r
4	SW4_RSV1	Reserved Switch 0: Default (switch is OFF)	r
5	SW4_RSV2	Reserved Switch 0: Default (switch is OFF)	r
6	SW4_RSV3	Reserved Switch 0: Default (switch is OFF)	r
7	Reserved	Reserved 0: Default	r

Memory Configuration Register

The Memory Configuration Register provides information on memory configuration options.

Table 35: *Memory Configuration Register*

Base Address: $F0000000_{16}$		**Offset: 04_{16}**	
Bit	**Signal**	**Description**	**Access**
2..0	SDRAM_SIZE	SDRAM configuration 000: Reserved 001: 256 MByte 010: 512 MByte 011: 768 MByte 100: 1 GByte 101: 1.5 GByte 110: 2 GByte 111: Reserved	r
3	FLASH_SIZE	User flash configuration 0: 32 MByte 1: 64 MByte	r
4	SDRAM CAS	SDRAM CAS latency 0: CAS latency is 3 1: CAS latency is 2	r
7..5	FPGA_REVISION	FPGA revision status 000: Revision 1.0	r

Last Reset Status Register 1

The Last Reset Status register 1 provides information on the type of the last hardware reset.

Table 36: *Last Reset Status Register 1*

Base Address: $F0000000_{16}$		**Offset: 05_{16}**	
Bit	**Signal**	**Description**	**Access**
0	PON_RST	Power-on reset 0: This type was not reason for last reset. 1: This type was reason for last reset.	r
1	FP_RST	Front panel push button reset 0: This type was not reason for last reset. 1: This type was reason for last reset.	r
2	CPCI_PRST	CompactPCI push button reset 0: This type was not reason for last reset. 1: This type was reason for last reset.	r
3	CPCI_RST	CompactPCI reset 0: This type was not reason for last reset. 1: This type was reason for last reset.	r
4	WD_RST	Discovery watchdog reset 0: This type was not reason for last reset. 1: This type was reason for last reset.	r
5	IPMI_RST	IPMI reset 0: This type was not reason for last reset. 1: This type was reason for last reset.	r
6	PMC1_RST	PMC slot 1 reset (if PPMC is plugged) 0: This type was not reason for last reset. 1: This type was reason for last reset.	r
7	PMC2_RST	PMC Slot 1 Reset (if PPMC is plugged) 0: This type was not reason for last reset. 1: This type was reason for last reset.	r

Last Reset Status Register 2

The Last Reset Status register 2 provides information on the type of the last hardware reset.

Table 37: *Last Reset Status Register 2*

Base Address: $F0000000_{16}$		**Offset: 06_{16}**	
Bit	**Signal**	**Description**	**Access**
0	SW_RST	Software requested board reset 0: This type was not reason for last reset. 1: This type was reason for last reset.	r
7..1	Reserved	Reserved 0000000: Default	r

A

Battery Exchange

Battery Exchange

The battery provides a data retention of ten years summing up all periods of actual battery use. Motorola therefore assumes that there usually is no need to exchange the lithium battery except for exchange in case of long-term spare part handling.

Caution

- **Board damage**
  Wrong battery installation may result in a hazardous explosion and board damage. Therefore, make sure the battery is installed as described below.
- **Data loss**
  Exchanging the battery after ten years of actual battery use have elapsed results in data loss. Therefore, exchange the battery before ten years have elapsed.
- **Data loss**
  Exchanging the battery always results in data loss of the devices which use the battery as power backup. Therefore, backup affected data before exchanging the battery.
- **Always use the same type of lithium battery as is already installed.**

The battery is located on the board such that it can easily be accessed.

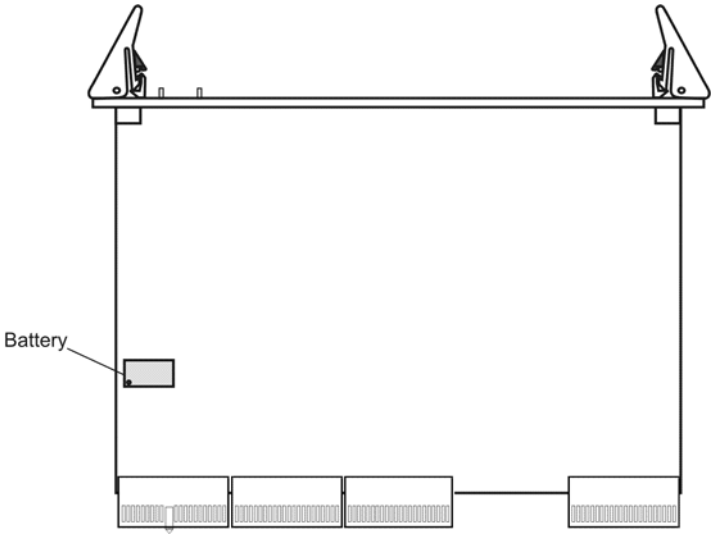


Figure 19: *Battery Location*

In order to exchange the battery, follow the instructions below:

1. Remove board from backplane
   For board removal procedure, see “Board Installation” page 2-16.
2. Remove battery
3. When installing new battery, ensure that dot on battery is in correct position (see figure below).

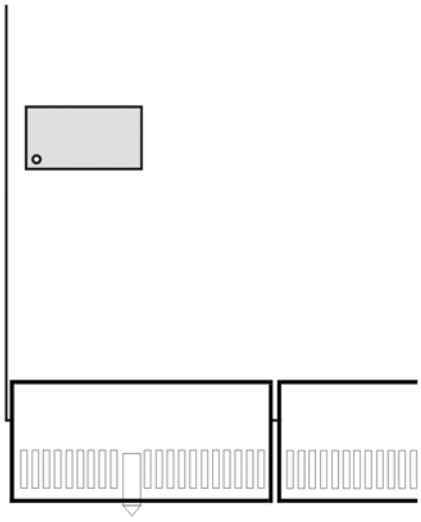


Figure 20: *Dot Position on Battery*

B

Troubleshooting

Error List

A typical CompactPCI system is highly sophisticated. This chapter can be taken as a hint list for detecting erroneous system configurations and strange behaviors. It cannot replace a serious and sophisticated pre- and post- sales support during application development.

If it is not possible to fix a problem with the help of this chapter, contact your local sales representative or FAE for further support.

Mechanical

Problem	Possible Reason	Solution
Unable to insert board into backplane	Damaged plugs, bent or broken pins: backplane defect	1. Check CompactPCI slot position to be used for bent or broken pins 2. Replace backplane
	Board defect	Replace board
	Keying of backplane does not fit to board	1. Check if you use correct board variant and replace board if necessary 2. Replace backplane

After Power-On

Problem	Possible Reason	Solution
Powering-on the board fails	Backplane voltages for device not within the specified range	1. Check that all backplane voltages are within their specific ranges 2. Check that power supply is capable to drive the respective loads

Problem	Possible Reason	Solution
	Board defect	Replace board
	Damaged plugs, bent or broken pins: backplane defect	1. Check CompactPCI slot position to be used for bent or broken pins 2. Replace backplane

During Boot-Up Procedure

Problem	Possible Reason	Solution
Board does not boot	Boot device is not partitioned according to used operating system	Check partition according to operating system's needs.
	Boot sequence not correct	Correct boot sequence, e.g. via PowerBoot
	Interrupts are not set correctly	Set interrupts correctly
	Wrong configuration of boot devices	Configure boot devices correctly, e.g via PowerBoot setboot command

During Board Operation

Problem	Possible Reason	Solution
Application software does not work	Memory ranges of system and peripheral boards do not match	Change application software so that memory ranges match I/O cards and host.
	Not enough disk capacity on mass storage device	Add disk capacity
	Not enough system memory	Add system memory
	Used I/O ranges do not match	Change application software so that I/O ranges match I/O cards and host.
Connected devices do not work	Device defect	Replace device
	Device not connected to power supply	Connect device to power supply

Problem	Possible Reason	Solution
	Wrong board configuration, faulty switch setting	Configure the board correctly for the respective device
	Devices are disabled	Configure board correctly
Board runs unstable	Disregard of environmental requirements	1. Check that temperature inside system stays within specified ranges for all system devices 2. Check for hot-spots within system Improve cooling system if necessary. 3. Check that other environmental values like moisture or altitude are kept within specified ranges
	Drivers are missing, faulty or do not match hardware	1. Check that all used hardware parts have a driver matching the hardware 2. Reinstall hardware drivers
	Board defect	Replace board
Low system performance	Caches are disabled	Enable caches
Memory / PMC module does not work	Module defect	Replace module
	Module not defined for the used board	1. Check if module specification matches with interface specification of board. 2. Replace module if specifications do not match
	Module not installed correctly	Check if module fits perfectly in socket.
	Wrong board configuration, faulty switch setting	Configure the board correctly for the respective module

Problem	Possible Reason	Solution
RTB does not work	RTB defect	Replace RTB
	RTB installed on wrong slot position	Install RTB on adjacent slot position of the used board.
	RTB not defined for the used peripheral or system board	Install RTB defined for the used peripheral or system board.

Index