

ARTISAN®
TECHNOLOGY GROUP

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

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

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Emerson / Motorola / Force Computers CPSB-560* at our website: *Click HERE*



SPARC/CPSB-560

Reference Guide

P / N 227474 Revision AA
September 2005

Copyright

©Copyright 2005 Motorola GmbH

All rights reserved.

Motorola and the stylized M logo are trademarks of Motorola,Inc., registered in the U.S. Patent and Trademark Office.

All other product or service names mentioned in this document are the property of their respective owners.

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 The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Motorola GmbH.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

Using this Guide

Other Sources of Information

Safety Notes

Sicherheitshinweise

1 Introduction

About this Manual 1-3
Organization of this Manual 1-3
Feedback 1-4

Features 1-5

Interfaces 1-6

Standard Compliances 1-7

Ordering Information 1-8
Product Nomenclature 1-8
Order Numbers 1-8

2 Installation

Action Plan 2-3

Requirements 2-6

Environmental Requirements 2-6

Power Requirements 2-7

Software Requirements 2-8

Hardware Upgrades and Accessories 2-9

Memory Module 2-9

IDE Devices 2-10

Hard-Disk Drive Accessory Kit 2-11

PMC Module 2-11

Voltage Key 2-12

Installation Procedure 2-13

Rear Transition Board 2-15

Switch Settings 2-16

Board Installation 2-18

Installation in a Nonpowered System 2-18

Installation Procedure 2-18

Installation in a Powered System Supporting Hot Swap 2-20

Installation Procedure 2-21

Powering Up 2-23

Solaris Installation 2-24

Via CD-ROM 2-24

Via Network 2-25

Software Upgrades and Accessories 2-26

Solaris Driver Package 2-26

Driver Names and Instance Numbers of Ethernet Devices 2-27

Driver FRCipmi 2-27

3 Controls, Indicators, and Connectors

Front Panel 3-3
LEDs 3-4
Keys 3-5
Reset Key 3-5
Abort Key 3-5
Connectors 3-6

IDE Connector 3-7

CompactPCI Connectors 3-9
J1 3-10
J2 3-11
J3 3-12
J5 3-13

4 OpenBoot Firmware

Introduction 4-3

CORE 4-4
CORE Workflow 4-5
CORE Key Commands 4-6
Obtaining CORE Help 4-8

POST 4-9

OpenBoot 4-10
Entering OpenBoot 4-10
Booting the Operating System 4-10
Running Diagnostics 4-15
Diagnostic Commands 4-15
OBDIAG 4-20
Displaying System Information 4-24
Resetting the System 4-24
Activating OpenBoot Help 4-24

Adding Drop-In Drivers and Updating OpenBoot 4-26
Drop-In Drivers 4-28
add-dropin 4-29
delete-dropin 4-30
show-dropins 4-31
Updating OpenBoot 4-31

5 Devices, Features and Data Paths

Block Diagram 5-3

PCI Bus A 5-4

PCI Bus B 5-5
IDE Interface 5-5
Ethernet Interface 4 5-5
PCIO-2 5-6

EBus 5-7
Boot PROM and Flash EPROM 5-7
Real-Time Clock/NVRAM 5-8
Serial Interfaces 5-8
Xilinx FPGA 5-8
Watchdog 5-9
Timer 5-9
Local I^2C Bus 5-10
IPMI Controller 5-10
I^2C Slave Addresses 5-11
Available IPMI Drivers 5-11
Temperature Sensor 5-12

PCI Bus C 5-13

6 Maps and Registers

Interrupt Map 6-3

Interrupt Concept 6-3

Interrupt Sources 6-3

Physical Memory Map 6-5

UltraSPARC-IIi+ Physical Address Memory Map 6-5

Memory Address Map 6-6

UltraSPARC-IIi+ Internal CSR Space 6-7

PCI Bus Address Map 6-7

PCIO-2 Address Map 6-9

System Configuration Registers 6-11

Overview of System Configuration Registers 6-11

Overview of IPMI Related Register Bits 6-14

Miscellaneous Control Register 6-14

Display Registers 6-15

- LED 0 Control Register 6-15
- LED 1 Control Register 6-16
- LED 2 Control Register 6-17
- LED 3 Control Register 6-18

External Failure Status Register 6-18

Watchdog Timer Registers 6-19

- Watchdog Timer Control Register 6-20
- Watchdog Timer Trigger Register 6-21
- Watchdog Timer Status Register 6-21

Timer Registers 6-22

- Timer Control Register 6-22
- Timer Clear Control Register 6-23
- Timer Status Register 6-24
- Timer Initial Control Registers 6-25
- Timer Counter Status Register 6-25

Interrupt Registers 6-26

- Interrupt Enable Control Register 6-26
- Interrupt Pending Status Register 6-27

Reset Registers 6-29

- Reset Control Register 6-29
- Reset Clear Control Registers 6-29
- Reset Status Register 6-30

Board Status Registers 6-31
Switch 1 and 2 Register 6-31
Switch 3 Register 6-31
Board Configuration Status register 6-32
FPGA Revision Status Register 6-32
I^2C Register 6-33

Index

Tables

Introduction

Table 1 Interfaces of CPSB-560 1-6
Table 2 Standard Compliances 1-7
Table 3 Ordering Information Excerpt 1-8

Installation

Table 4 Environmental Requirements 2-6
Table 5 Power Requirements without Memory Module 2-7
Table 6 Power Requirements with Memory Module 2-8
Table 7 Switch Settings 2-17
Table 8 Devices and their Appropriate Drivers 2-26
Table 9 Instance Number Assignement 2-27

Controls, Indicators, and Connectors

Table 10 Description of Front Panel LEDs 3-4
Table 11 User LEDs During Power Up 3-4

OpenBoot Firmware

Table 12 Boot Configuration Parameters 4-12
Table 13 Boot Parameters 4-13
Table 14 Device Alias Definitions for SCSI 4-13
Table 15 Device Alias Definitions for IDE 4-14
Table 16 Diagnostic Routines 4-15
Table 17 OBDIAG Commands 4-21
Table 18 Commands to Display System Information 4-24
Table 19 PCI-Based FCODE Drivers Compared to Supported Hardware Devices 4-28
Table 20 Drop-In Drivers 4-29

Devices, Features and Data Paths

Table 21 Slave Addresses of Local I^2C Bus 5-10
Table 22 I^2C Slave Addresses 5-11

Maps and Registers

Table 23 Interrupt Sources 6-3
Table 24 UltraSPARC-IIi+ Main Address Map 6-5
Table 25 Main Memory Address Map 6-6
Table 26 UltraSPARC-IIi+ Internal CSR Space 6-7
Table 27 PCI Bus Address Map 6-8
Table 28 PCIO-2 Address Map 6-9
Table 29 Alphabetical List of System Configuration Registers 6-11
Table 30 System Configuration Registers Sorted by Address Range 6-12
Table 31 Overview of IPMI Related Register Bits. 6-14
Table 32 Miscellaneous Control Register 6-14
Table 33 LED 0 Control Register 6-15
Table 34 LED 1 Control Register 6-16
Table 35 LED 2 Control Register 6-17
Table 36 LED 3 Control Register 6-18
Table 37 External Failure Register 6-19
Table 38 Watchdog Timer Control Register 6-20
Table 39 Watchdog Timer Trigger Register. 6-21
Table 40 Watchdog Timer Status Register. 6-21
Table 41 Timer Control Register 6-22
Table 42 Timer Clear Control Register 6-23
Table 43 Timer Status Register 6-24
Table 44 Timer Initial Control Registers 6-25
Table 45 Timer Counter Status Register 6-25
Table 46 Interrupt Enable Control Register 6-26
Table 47 Interrupt Pending Status Register 6-27
Table 48 Reset Control Register. 6-29
Table 49 Reset Clear Control Register 6-29
Table 50 Reset Status Register 6-30
Table 51 Switch 1 and 2 Register. 6-31
Table 52 Switch 3 Register 6-31
Table 53 Board Configuration Status Register 6-32
Table 54 FPGA Revision Status Register 6-32
Table 55 I^2C Register. 6-33

Figures

Introduction

Figure 1 Function Blocks 1-5

Installation

Figure 2 Memory Module Connectors 2-9
Figure 3 IDE Connector 2-10
Figure 4 Voltage Key 2-12
Figure 5 PMC Connectors 2-13
Figure 6 Position of Mounting Holes 2-14
Figure 7 Location of Switches on CPU Board 2-16
Figure 8 Compatibility Glyph 2-18

Controls, Indicators, and Connectors

Figure 9 Front Panel 3-3
Figure 10 Ethernet 1 Connector Pinout 3-6
Figure 11 Serial Interface A Connector Pinout 3-6
Figure 12 Location of IDE Connector 3-7
Figure 13 IDE Connector Pinout 3-8
Figure 14 Location of CompactPCI Connectors 3-9
Figure 15 J1 Connector Pinout, Rows A-C 3-10
Figure 16 J1 Connector Pinout, Rows D and E 3-10
Figure 17 J2 Connector Pinout, Rows A-C 3-11
Figure 18 J2 Connector Pinout, Rows D and E 3-11
Figure 19 J3 Connector Pinout, Rows A-C 3-12
Figure 20 J3 Connector Pinout, Rows D and E 3-12
Figure 21 J5 Connector Pinout, Rows A-C 3-13
Figure 22 J5 Connector Pinout, Rows D and E 3-14

OpenBoot Firmware

Figure 23 System Overview 4-4
Figure 24 OBDIAG Main Menu Mask 4-20

Devices, Features and Data Paths

Figure 25 Block Diagram 5-3
Figure 26 Location of Board Temperature Measurement 5-12

Maps and Registers

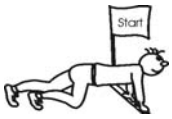
Figure 27 Battery Location A-3

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 signaled
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 used for optional OpenBoot parameters
..	Ranges
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

Notation	Description
	Ordering information
	Action plan for installation or module exchange
	Start of a procedure
	End of a procedure
	Problem / error message
	Possible reason
	Possible solution

Abbreviations

BIB	Board Information Block
BMC	Baseboard Management Controller
API	Application Programming Interface
BMC	Base Board Management Controller
CAS	Column Address Select
COP	Common On-Chip Processor
CPCI	Compact Peripheral Component Interconnect
CPU	Central Processing Unit
CSR	Configuration Space Registers
DMA	Direct Memory Acces
DRAM	Dynamic Random Access Memory
ECC	Error Checking and Correction
EPROM	Erasable Programmable Read Only Memory
ESD	Electrostatic Discharge
FAE	Field Application Engineer
FPGA	Field-Programmable Gate Array
GND	Ground
GPP	General Purpose Pins
I^2C	Intelligent Interface Controller
IBMU	Intelligent Board Management Unit
ICMB	Inter Chassis Management Bus
IDE	Integrated Device Electronics
IOM	I/O Memory Management Unit
IPMI	Intelligent Platform Management Interface
KCS	Keyboard Controller Style
LED	Light Emitting Diode
LFM	Linear Feet per Minute
MAC	Media Access Control Layer

MCU	Memory Control Unit
MII	Media Independent Interface
NMI	Nonmaskable Interrupt
NVRAM	Nonvolatile Read-Only Memory
PBM	PCI bus module
PCI	Peripheral Component Interconnect
PHY	Physical Layer
PIE	PCI Interrupt Engine
PLD	Programmable Logic Device
PM	Peripheral Management Controller
PMC	PCI Mezzanine Card
PROM	Programmable Read Only Memory
PSB	Packed Switching Backplane
RAM	Random Access Memory
ROM	Read-Only Memory
RTB	Rear Transition Board
RTC	Real Time Clock
SBC	Single-Board Computer
SEL	System Event Log
SDR	Sensor Data Record
SDRAM	Synchronous Dynamic Random Access Memory
SELV	Safety Extra Low Voltage
SMB	Serial Management Bus
SMI	System Management Interrupt
SRAM	Static RAM
TPE	Twisted-Pair Ethernet
UART	Universal Asynchronous Receiver / Transmitter
UIC	UPA Interrupt Concentrator
UPA	Ultra Port Architecture

Revision History

Order No.	Rev.	Date	Description
216440	AA	August 2002	First revision of Installation Guide
216440	AB	September 2002	Added pinouts of J1 and J2 to “CompactPCI Connectors” page 3-9
216440	AC	February 2003	Changed pinout description of J3 B, C and E page 3-12; changed pinout description of J5 A, B, D and E page 3-13; changed position of voltage key on Figure 4 “Voltage Key” page 2-12; changed VI/O value of PMC modules page 2-12; added that installing PMC modules with 5V VI/O damages the board; added boot flash segmentation for switch SW2 page 2-17; added that user LED0 indicates if SPARC/CPSB-560 does not run properly page 3-4 and page 6-15; added position and completed function description of temperature sensor page 5-12; editorial changes on pages 4-15, 4-24, 4-29, 4-30, 4-31; added public bus and private bus classification page 5-10; changed IPMI1-3 to KCS0-2 in Table 23 “Interrupt Sources” page 6-3; changed KCS1-3 to KCS0-2 in Table 28 “PCIO-2 Address Map” page 6-9; added signal RTB_GPO to Table 32 “Miscellaneous Control Register” page 6-14; added offset value to Table 37 “External Failure Register” page 6-19; added signal SCSI AUTO-TERMINATION to Table 53 “Board Configuration Status Register” page 6-32; changed EN 55022 Class A to EN 55022 and IEC 68-2-1/2/3/13/14 and IEC 68-2-6/27/32 to IEC 60068-2-1/2/3/13/14 and IEC 60068-2-6/27/32 in Standard Compliances section in the Introduction chapter; changed node slot picture on pages in Installation section in the Safety Notes chapters, in Board Installation section in the Installation section; changed pinout figures in CompactPCI Connectors section in the Controls, Indicators, and Connectors chapter, used _N for negative and M Bus 100 MHz to SDRAM Bus 93 MHz in block diagram in Buses chapter, and added ETH2 and ETH3; added note to section Temperature Sensor in Buses chapter that MAX1617 temperature sensor is not IBMU powered

Order No.	Rev.	Date	Description
216440	AD	September 2003	Changed RTC / NVRAM SGS M48T35A to RTC / NVRAM SGS M48T58Y in Other Sources of Information section, Using this Guide chapter; Changed title of Buses chapter to “Devices’ Features and Data Paths”; Changed M48T35A to M48T58Y in Buses chapter; Removed footnote from Ordering Information Excerpt table in Introduction chapter; Changed heading “Dear Customer,” to “Battery Exchange” and “Error List” in appendices; Editorial changes
222911	AA	March 2004	Extended description of 110596 board variant: mentioned additional SCSI and Ethernet interfaces which this variant provides In the “Abort Key” description: added note on usage restrictions Added section: Entering OpenBoot Extended description of User LED 0
227474	AA	September 2005	Brought manual to Motorola-style (copyright, logo, etc.)

Other Sources of Information

For further information refer to the following documents and data sheets of the following devices.

Company	www.	Document
Motorola	motorola.com / computers	ACC / RTB-505 Installation Guide SPARC / MEM-50 Installation Guide SPARC / CPSB-560 / HDAcc-Kit Installation Guide IPMI Reference Guide, only available via S.M.A.R.T. Solaris Driver Packages Installation and Reference Guide
Intel	developer.intel.com	Ethernet controller 82559ER GigaBit Ethernet controller 8254xEM PHYceiver LXT970A
LSI Logic	lsilogic.com	PCI-to-Ultra160 SCSI controller LSI53C1000
Samsung	samsung.com	256 MBit SDRAM, 512 MBit SDRAM
Silicon Image	siliconimage.com	IDE controller
ST Microelectronics	eu.st.com	RTC / NVRAM SGS M48T58Y
Sun Microsystems	sun.com	Advanced PCI Bridge SME2411 PCIO-2 controller SME2300 UltraSPARC IIi + CPU
Texas Instruments	ti.com	Serial controller 16C554
Vitesse	vitesse.com	IPMI controller VSC215
XICOR	xicor.com	X24C02 serial E^2PROM
Xilinx	xilinx.com	XCS20XL local FPGA
Intel	www.intel.com / design / servers / ipmi / spec_old.htm	Intelligent Platform Management Interface Specification v. 1.0 Rev. 1.1

Company	www.	Document
	developer.intel.com Search for *FRU Specification*	IPMI -Platform Management FRU Information Storage Definition v1.0 Rev. 1.1
-	www.picmg.com	PICMG 2.9 R1.0 System Management Specification



Safety Notes

This section provides safety precautions to follow when installing, operating, and maintaining the SPARC / CPSB-560.

We intend to provide all necessary information to install and handle the SPARC / CPSB-560 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 SPARC/CPSB-560 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 SPARC/CPSB-560. 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 using single-point grounding 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. If you ground the board at multiple points, EMC problems may arise.

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 you will be required to correct the interference at your own expense.

If you use the board without a PMC module, cover the empty slot with a blind panel 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,

- **Incorrect board installation or removal may cause malfunctioning of the board or board damage. 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.**
- **Pressing the front panel when plugging the board in or removing it causes board damage. Do not press on front panel but use handles.**
- **Incorrect installation or removal of additional devices or modules may cause malfunctioning of the board or board damage. Before installing or removing an additional device or module, read the respective documentation.**
- **Disconnected power pins and connectors may cause board malfunction. Make sure that the board is connected to the backplane via all assembled connectors and that power is available on all power pins.**
- **Installing the CPSB-560 into a fabric slot marked by a sign shown below damages the CPSB-560. Do not install CPSB-560 into a fabric slot.**



Only install the CPSB-560 into node slots marked by the following glyphs



Hot Swap

Installing the SPARC/CPSB-560 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.

Removing the board from a powered system with IDE devices attached to the board's secondary IDE interface via the RTB-505 results in data loss.



Do not remove the board from a powered system with IDE devices attached to the board's secondary IDE interface.

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.

Operation

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

High humidity and condensation on the surface causes 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.

Electromagnetic radiation may disturb the board's functions. Ensure that the board is bolted on the CompactPCI system and the system is shielded by enclosure.

The board may not operate properly when contacts and cables of the board are touched during operation. 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).

PMC Module

Installing PMC modules with a VI/O of 5V damages the PMC module and the board itself. Therefore, only install universal or 3.3V PMC modules and do not change the position of the voltage key.



If the power consumption of the PMC module exceeds 7.5W the board and the PMC module are 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).

Switch Settings

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

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.



RJ-45 Connector

Connecting telephones to the RJ-45 connector damages the board and the telephone. Therefore, only connect an Ethernet network to the board's RJ-45 connector. Furthermore, take note of the following:

- **Clearly mark TPE connectors near your working area as network connectors.**
- **TPE bushing of the system has to be connected only to safety extra low voltage (SELV) circuits.**
- **The length of the electric cable connected to a TPE bushing must not exceed 100 meters.**

Battery Exchange

Wrong battery installation may result in a hazardous explosion and board damage. Therefore, make sure the battery is installed correctly, see "Battery Exchange" page A-1.

Exchanging the battery after five years of actual battery use have elapsed results in data loss. Therefore, exchange the battery before five years 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.

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 beim Einbau, Betrieb und bei der Wartung des SPARC / CPSB-560 zu beachten sind.

Wir sind darauf bedacht, alle notwendigen Informationen, die für die Installation und den Betrieb erforderlich sind, in diesem Handbuch bereit zu stellen. Da es sich jedoch um ein komplexes Produkt handelt bzw. viele verschiedene Einsatzmöglichkeiten bestehen, können wir die Vollständigkeit der im Handbuch enthaltenen Informationen nicht garantieren. Falls Sie weitere Informationen benötigen sollten, wenden Sie sich bitte an die für Sie zuständige Geschäftstelle von Motorola.

Das SPARC/CPSB-560 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 entsprechend der PCI Spezifikation 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.
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 den freien PMC-Steckplatz 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:

- **Unsachgemäße Installation oder Deinstallation kann das Board beschädigen oder seine Funktionsfähigkeit beeinträchtigen. Lesen Sie vor Ein- oder Ausbau des Boards das Kapitel “Action Plan”.**
- **Das Berühren des Boards oder elektronischer Komponenten in einem nicht ESD-geschützten Bereich kann zur Beschädigung des Boards oder der Komponenten führen. Bevor Sie Boards oder elektronische Komponenten berühren, vergewissern Sie sich, dass Sie in einem ESD-geschützten Bereich arbeiten.**
- **Durch Drücken der Frontblende, während Sie das Board ein- oder ausbauen, wird das Board beschädigt. Drücken Sie bei Ein- oder Ausbau des Boards nicht auf die Frontblende, sondern benutzen Sie die Griffe.**
- **Unsachgemäße Installation oder Deinstallation von zusätzlichen Geräten und Modulen kann das Board beschädigen oder seine Funktionsfähigkeit beeinträchtigen. Lesen Sie vor dem Ein- oder Ausbau von zusätzlichen Geräten oder Modulen das dazugehörige Benutzerhandbuch.**
- **Nicht angeschlossene Stecker und Versorgungskontakte können die Funktionsfähigkeit des Boards beeinträchtigen. Vergewissern Sie sich, dass das Board über alle Stecker an die Backplane angeschlossen ist und alle Versorgungskontakte mit Strom versorgt werden.**
- **Wenn Sie die CPSB-560 in einen Fabric-Steckplatz installieren, der mit dem folgendem Symbol gekennzeichnet ist, wird die CPSB-560 beschädigt.**

  Installieren Sie die CPSB-560 deshalb nur in Node-Steckplätze, die mit folgenden Symbolen gekennzeichnet sind.



Hot Swap

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

Wenn Sie die CPSB-560 aus einem laufenden System entfernen und am RTB-505 IDE-Festplatten angeschlossen sind, führt das zu Datenverlust. Entfernen Sie die CPSB-560 nicht aus einem laufenden System, wenn am RTB-505 IDE-Festplatten angeschlossen sind.

Wenn Sie das Board im laufenden Betrieb herausziehen, 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 Board-Oberfläche können Kurzschlüsse entstehen. Betreiben Sie das SPARC/CPSB-560 nur innerhalb der angegebenen Grenzwerte für die relative Luftfeuchtigkeit und Temperatur. Stellen Sie vor dem Einschalten des Stroms sicher, dass sich auf dem SPARC/CPSB-560 kein Kondensat befindet.

Elektromagnetische Strahlung kann die Funktionsfähigkeit des Boards beeinträchtigen. Wenn Sie das Board in Gebieten mit elektromagnetischer Strahlung betreiben, stellen Sie sicher, dass das Board mit dem System verschraubt ist und das System durch ein Gehäuse abgeschirmt wird.

Die Funktionsfähigkeit des Boards kann beeinträchtigt werden, wenn Anschlüsse und Kabel während des Betriebs berührt werden. 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).

PMC Modul

Wenn ein PMC Modul mit einer VI/O-Spannung von 5V installiert wird, werden das PMC Modul und das Board beschädigt. Installieren Sie deshalb nur universelle oder 3,3V PMC Module und verändern Sie nicht die Position des Voltage Keys (Codierstift für die Betriebsspannung).

Wenn der Stromverbrauch des PMC Moduls 7,5W übersteigt, werden das Board und das PMC Modul beschädigt. Stellen Sie deshalb sicher, dass der Gesamtstromverbrauch der Spannungspegel +/-12V, 5V und 3,3V 7.5W nicht überschreitet (Summe der verwendeten Spannungen).

Switch-Einstellungen

Das Einstellen der Switches während des Betriebs kann das Board beschädigen. Überprüfen und ändern Sie die Switch-Einstellungen deshalb vor der Installation des Boards.

Wenn Sie die Einstellungen von Switches, die als „reserved" markiert sind, ändern, kann die Funktionsfähigkeit des Boards beeinträchtigt werden. Ändern Sie nicht die Einstellungen von Switches, die als „reserved" markiert sind, weil Sie mit produktionsspezifischen Funktionen belegt sein könnten.



RJ-45 Stecker

Wenn Sie Telefone an den RJ-45 Stecker anschließen, kann das Telefon und das Board beschädigt werden. Schließen Sie deshalb ausschließlich Ethernet-Netze an den RJ-45 Stecker an. Beachten Sie deshalb folgende Punkte:

- **Vergewissern Sie sich, dass Anschlüsse deutlich als Netzwerkanschlüsse gekennzeichnet sind.**
- **Schließen Sie TPE-Stecker/Netzwerkstecker Ihres Systems nur an Sicherheits-Kleinspannungs-Kreise (SELV) an.**
- **Vergewissern Sie sich, dass die an einem TPE-Anschluss angeschlossene Leitung eine Gesamtlänge von 100 Metern nicht überschreitet.**

Batteriewechsel

Wenn die Batterie falsch installiert wird, kann das eine gefährliche Explosion und die Beschädigung des Boards zur Folge haben. Stellen Sie deshalb sicher, dass die Batterie so installiert wird wie im Kapitel "Battery Exchange" im Appendix A beschrieben.

Wenn die Batterie länger als 5 Jahre in Gebrauch ist, führt dies zu Datenverlust. Ersetzen Sie die Batterie deshalb innerhalb von 5 Jahren.

Der Austausch der Batterie hat immer einen Datenverlust der Geräte zur Folge, die die Batterie als Reserve zur Stromversorgung verwenden. Sichern Sie deshalb Ihre Daten vor dem Austausch der Batterie.

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
OpenBoot	Describes the main features of the board's OpenBoot
Devices, Features and Data Paths	Provides detailled information on the devices, such as controllers, CPU etc., used on the board and how they are interconnected
Maps and Registers	Provides information that is relevant for programmer's. This includes registers, interrupt and memory maps

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 SPARC / CPSB-560 is a high-performance single-board computer and is based on the 650 MHz UltraSPARC-IIi + processor. It supports PICMG 2.16 and can be installed in hot-swap and high-availability systems. Other features of the SPARC / CPSB-560 are:

- Two GBit Ethernet interfaces routed in the backplane
- Two 10 / 100 MBit Ethernet interfaces for external connection
- One Ultra-3 Wide 160 SCSI interface
- 512 KByte L2 cache for fast processing
- IPMI controller for system management
- 512 MByte or 1 GByte on-board SDRAM memory

The figure below shows the function blocks of the SPARC / CPCI-560.

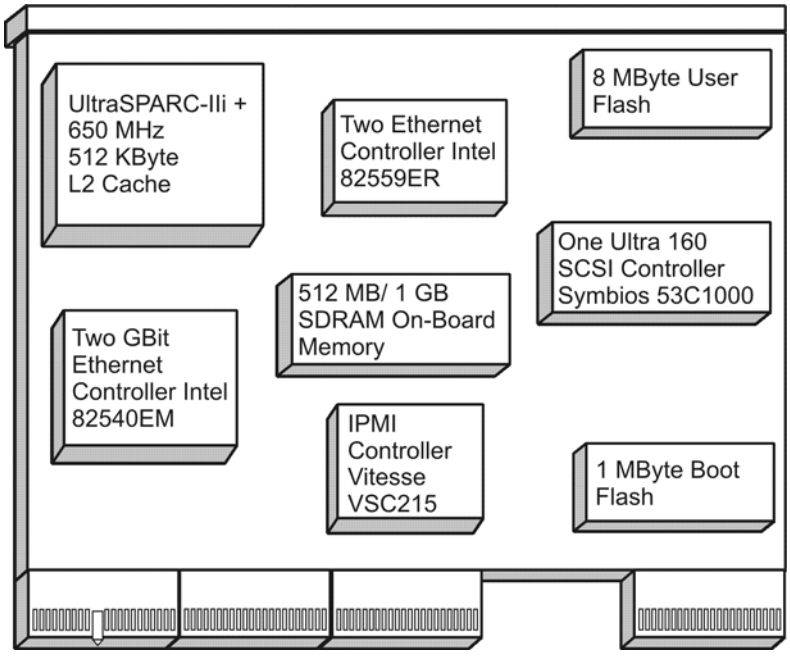


Figure 1: *Function Blocks*

Interfaces

The SPARC / CPSB-560 provides the following interfaces:

Table 1: *Interfaces of CPSB-560*

Interface	Description
Ethernet	Four Ethernet interfaces: Ethernet 1: 10 / 100 MBit interface, alternatively available on front panel or on RTB-505 as Ethernet 5 Ethernet 2 and 3: 1 GBit interface, routed in backplane Ethernet 4: 10 / 100 MBit interface available on RTB-505
IDE	One on-board IDE interface
PMC	One PMC interface
SCSI	One Ultra-3 Wide 160 SCSI interface available on RTB-505
Serial I / O	Four serial interfaces: A: Available on front panel B: Available on front panel of RTB-505 C: Available on RTB-505 as SUN keyboard interface D: Available on RTB-505 as SUN mouse interface
USB	Two interfaces available on RTB-505

Standard Compliances

The SPARC/CPSB-560 meets the following standards:

Note: EN 55022 and FCC Part 15 Class A are achieved by using single-point grounding. If you ground the CPSB-560 at multiple points EMC problems may occur.

Table 2: *Standard Compliances*

Standard	Description
IEC 60068-2-1/2/3/13/14	Climatic environmental requirements The CPSB-560 can only be used in a restricted temperature range. See “Environmental Requirements” page 2-6 for details.
IEC 60068-2-6/27/32	Mechanical environmental requirements
EN 609 50/UL 60950 UL 94V-0/1 (predefined Motorola system)	Legal requirements
EN 61000-6-2 EN 55022, EN 55024, FCC Part 15 Class A	EMC requirements on system level
ANSI/IPC-A-610 Rev.C Class 2, ANSI/IPC-7711, ANSI/IPC-7721, ANSI-J-001...003	Manufacturing Requirements

Ordering Information

When ordering the SPARC / CPSB-560 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.

SPARC/CPSB-560/xxxx-ccc-Lyyy-z	
xxxx	Memory size in MByte
ccc	Processor clock frequency in MHz
Lyyy	L2 cache capacity in KByte
z	Flash EPROM capacity in MByte

Order Numbers

Depending on the SPARC / CPSB-560 type, the available upgrades and accessories may differ. Consult your local sales representative to check the possibility of combinations.

Table 3: *Ordering Information Excerpt*



Order No.	SPARC/CPSB-560/...	Description
110596	… / 1024-650-L512-8	650 MHz CPU with 512 KByte L2 cache, 1 GByte SDRAM on-board memory, 8 MByte user flash, space for one PMC module and space for one memory module or one IDE hard-disk drive, PICMG 2.16 compliant, IPMI support
110595	... / 512-650-L512-8	650 MHz CPU with 512 KByte L2 cache, 512 MByte SDRAM on-board memory, 8 MByte user flash, space for one PMC module and space for one memory module or one IDE hard-disk drive, PICMG 2.16 compliant, IPMI support **Note: this variant does not provide the following interfaces routed to the RTB: Ethernet interface #4 and SCSI.**

Table 3: *Ordering Information Excerpt (cont.)*

Order No.	SPARC/CPSB-560/...	Description
Hardware Accessories SPARC/CPSB-560		
110598	ACC / RTB-505 / PSB	Rear transition board for SPARC / CPSB-560
110599	ACC / CPSB-560 / HD	2.5“ hard-disk drive Accessory Kit
109045	SPARC / MEM-550	1 GByte memory upgrade module for SPARC / CPSB-560 CPU board
Software Accessories SPARC/CPSB-560		
105608	SPARC / SOL / DRV R.2.x	Solaris driver package for Solaris 8 for SPARC / CPSB-560

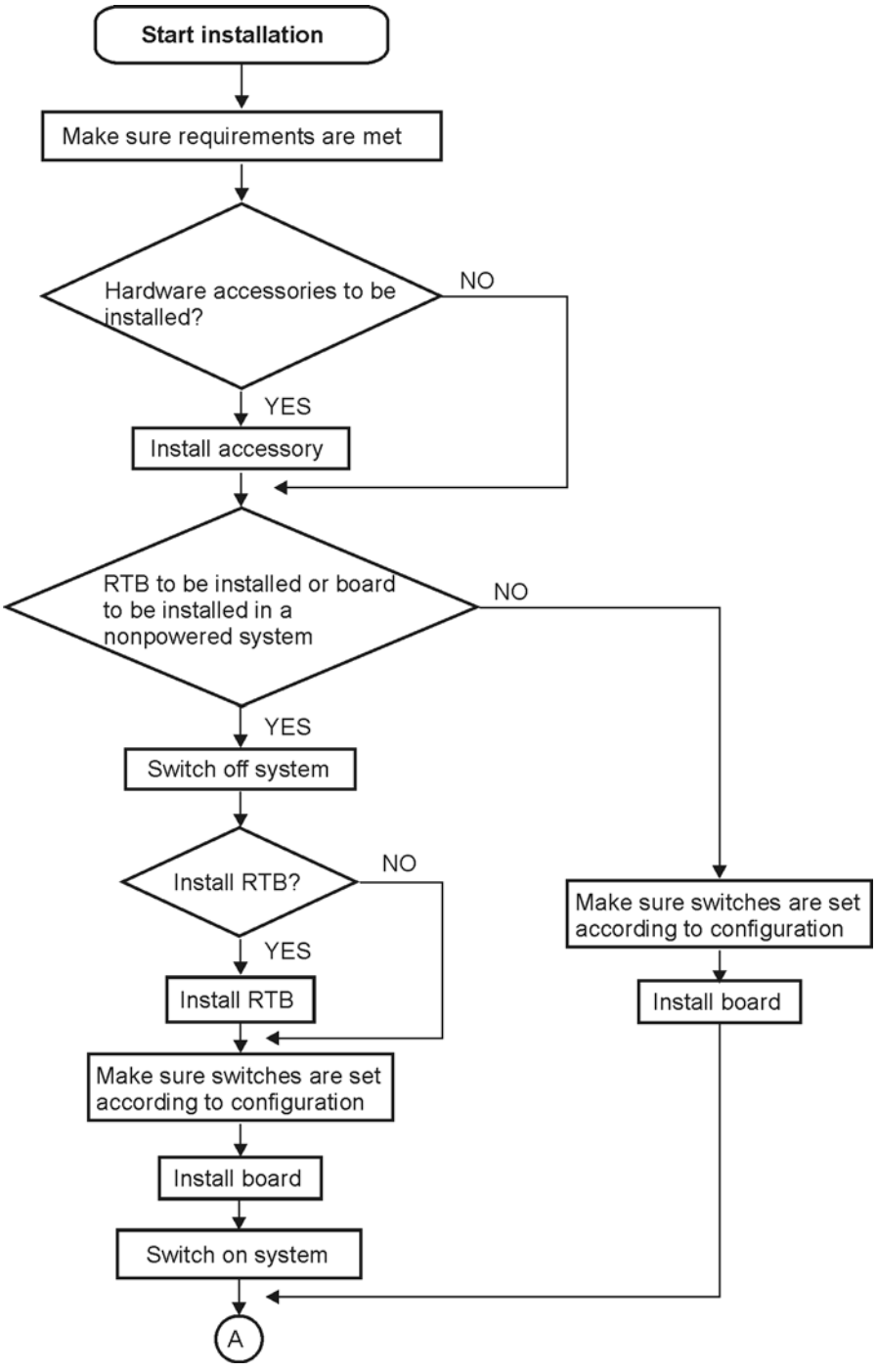
2

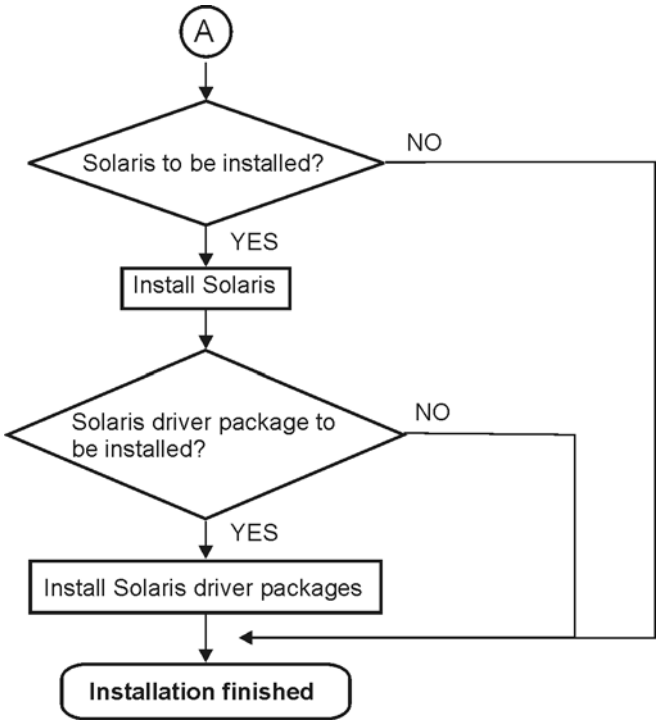
Installation

Action Plan

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




Start installation
Make sure requirements are met
Hardware accessories to be installed?
NO
YES
Install accessory
RTB to be installed or board to be installed in a nonpowered system
NO
YES
Switch off system
Install RTB?
NO
YES
Install RTB
Make sure switches are set according to configuration
Install board
Switch on system
A
Make sure switches are set according to configuration
Install board


A
Solaris to be installed?
NO
YES
Install Solaris
Solaris driver package to be installed?
NO
YES
Install Solaris driver packages
Installation finished

Requirements

To meet the environmental requirements, the SPARC/CPSB-560 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.**
- **The environmental values given in the table below only apply to the CPSB-560 without any accessories. If installing accessories, their environmental requirements must also be taken into account. If you use the CPSB-560 together with the SPARC/CPSB-560/HD-AccKit, make sure the environmental values given in the *SPARC/CPSB-560/HD-AccKit Installation Guide* are met.**

Caution



Board damage
Board surface
High humidity and condensation on the surface causes 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 4: *Environmental Requirements*

Feature	Operating	Non-Operating
Temperature	0°C to +50°C	–40°C to +85°C
Forced Airflow	300 LFM (linear feet per minute)	-

Table 4: *Environmental Requirements (cont.)*

Feature	Operating	Non-Operating
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	10 Hz to 15 Hz: 2 mm amplitude 15 Hz to 150 Hz: 2 g	10 Hz to 15 Hz: 5 mm amplitude 15 Hz to 150 Hz: 5 g
Shock	5 g / 11 ms halfsine	15 g / 11 ms halfsine
Free Fall	100 mm / 3 axes	1200 mm / all edges and corners (packed state)

Power Requirements

The board's power requirements depend on the installed hardware accessories. The following tables give typical power requirements for 5V and 3.3V with and without a memory module. If you want to install further accessories, the load of the respective accessory has to be added to the load of the SPARC / CPSB-560.

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

Table 5: *Power Requirements without Memory Module*

SPARC/CPSB-560	3.3V	5V	5V IPMI
Typical current	2.5A	4.0A	0.1A
Max. current	3.0A	4.5A	0.4A
Typical power requirements	8.25W	20W	0.5W
Min. voltage	3.2V	4.85V	4.85V
Max. voltage	3.45V	5.25V	5.25V

Table 6: *Power Requirements with Memory Module*

SPARC/CPSB-560	3.3V	5V	5V IPMI
Typical current	3.0A	4.0A	0.1A
Max. current	3.5A	4.5A	0.4A
Typical power requirements	9.9W	20W	0.5A
Min. voltage	3.2V	4.85V	4.85V
Max. voltage	3.45V	5.25V	5.25V

Software Requirements

If you wish to use Solaris and one of the CPU board devices listed below you need to install the Motorola Solaris Driver Package Rel. 2.17:

- Intel 82559ER Ethernet device
- Intel 8254xEM GBit Ethernet device
- On-board flash EPROM
- Vitesse IPMI controller VSC215
- LEDs
- Ejector switch

For information on the driver package itself and which drivers have to be installed, refer to section “Solaris Driver Package” page 2-26.

Hardware Upgrades and Accessories

The SPARC/CPSB-560 itself allows an easy and cost-efficient way to adapt the board to your application needs. The following accessories can be installed on the SPARC/CPSB-560:

- Memory module
- IDE hard disk
- PMC module
- RTB-505

Memory Module

The main memory capacity is upgradeable by installing the Motorola qualified memory module SPARC/MEM-550.

The CPU board already incorporates 512 MByte or 1 GByte of SDRAM memory which can be upgraded with one SPARC/MEM-550 module of 1 GByte altogether providing 1.5 or 2 GByte of SDRAM capacity.

Note: There is only space for one memory module or one IDE hard-disk drive.

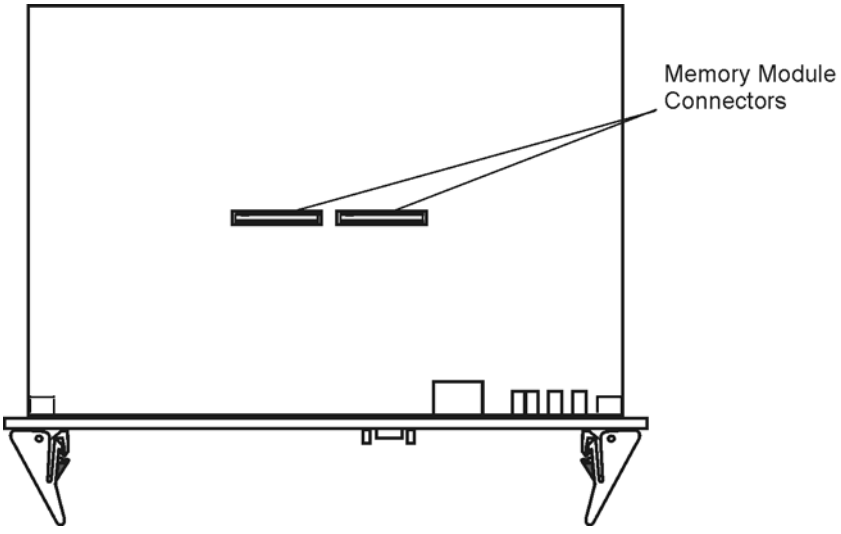


Figure 2: *Memory Module Connectors*

For information on the memory module's installation, see the *SPARC/MEM-550 Installation Guide.*

IDE Devices

The IDE controller allows to install up to two IDE devices on the primary IDE bus. However, only one IDE device can be installed because the CPSB-560 only has mounting holes for one device.

Note:

- **There is only space for one IDE device or one memory module. If you want to install both, you can install an IDE hard-disk PMC module into the PMC slot.**
- **We do not recommend to use the following hard-disk types: IBM DDRS-39130, IBM DDRS-34560 and IBM DNES-309170. Booting from these hard-disk types is not supported.**

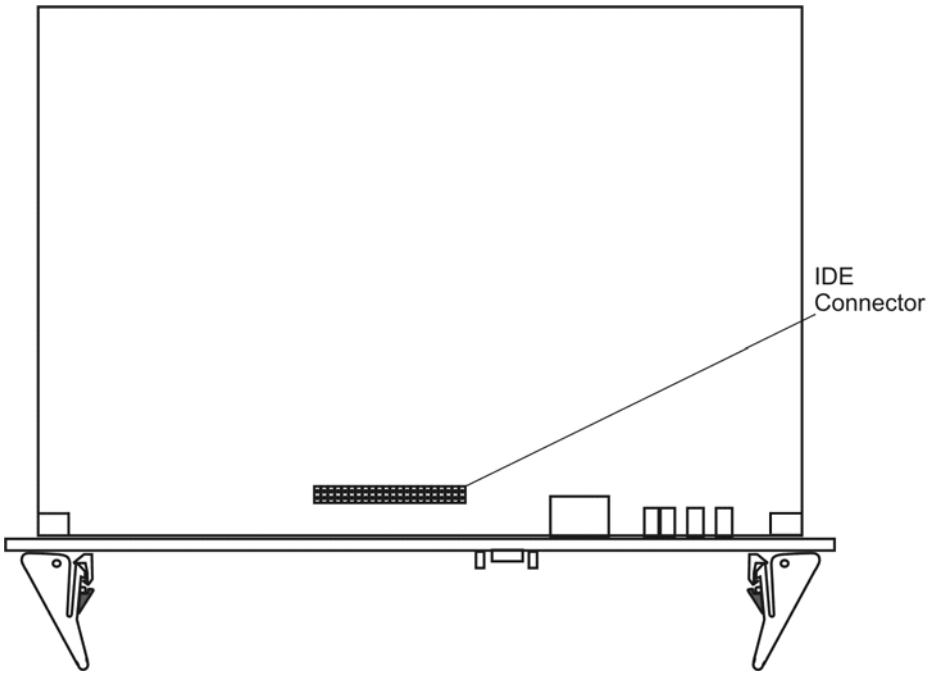


Figure 3: *IDE Connector*

Further devices can be connected to the secondary IDE interface via the RTB-505. For further information, refer to the *ACC/RTB-505 Installation Guide.*

Hard-Disk Drive Accessory Kit

The Motorola hard-disk accessory kit ACC / CPSB-560 / HD-AccKit provides a local mass storage device. It can be connected to the primary IDE port via the on-board IDE connector.

For informationon installation and environmental requirements, refer to the *ACC/CPSB-560/HD-AccKit Installation Guide*.

PMC Module

The board provides one PMC slot. It supports a 32-bit data bus width with a maximum frequency of 33 MHz and supports +/–12V.

Note:

- **To ensure proper EMC shielding, either operate the board with the blind panel or with a module installed.**
- **If the board is upgraded with a PMC module, ensure that the blind panel is stored in a safe place in order to be used again when removing the PMC module.**
- **Processor PMC modules are only supported in non-monarch mode.**

Voltage Key

The PCI bus applies 3.3V VI/O on the PMC slots and it allows to install universal and 3.3V PMC modules. The SPARC/CPSB-560 provides a voltage key which prevents 5V PMC modules from being installed.

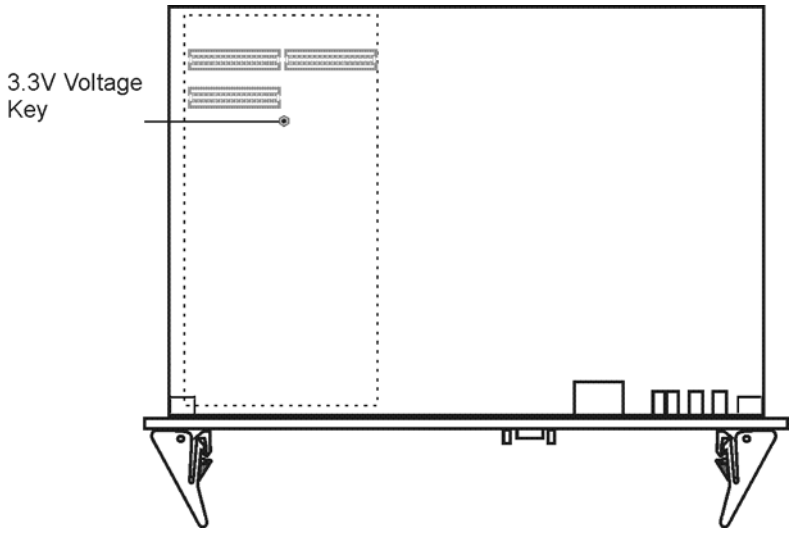


Figure 4: *Voltage Key*

Caution

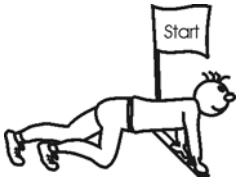
PMC module and board damage
Installing PMC modules with a VI/O of 5V damages the PMC module and the board itself. Therefore, only install universal or 3.3V PMC modules and do not change the position of the voltage key.

Installation Procedure

Caution



PMC module and board damage
If the power consumption of the PMC module exceeds 7.5W the board and the PMC module are 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).



1. Remove blind panel of PMC slot from front panel
2. Store blind panel in a safe place
3. Plug PMC module into connectors of PMC slot

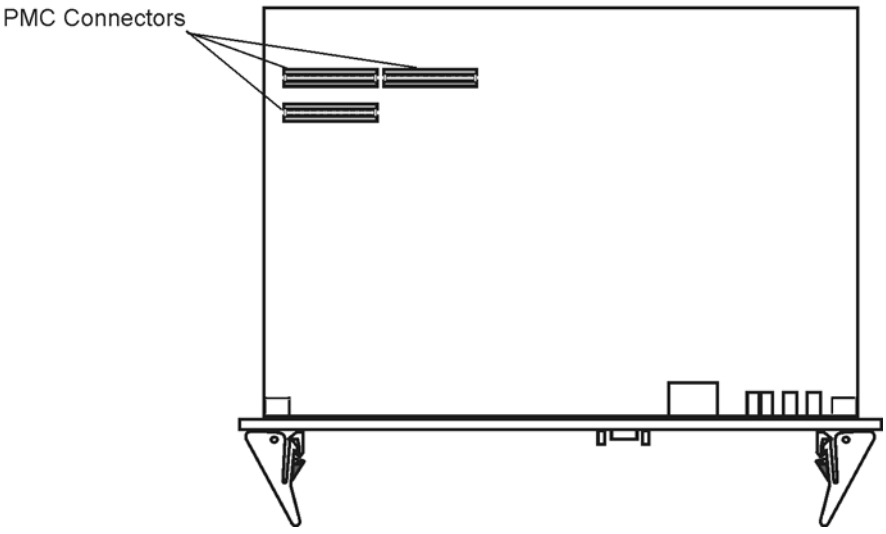


Figure 5: *PMC Connectors*

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

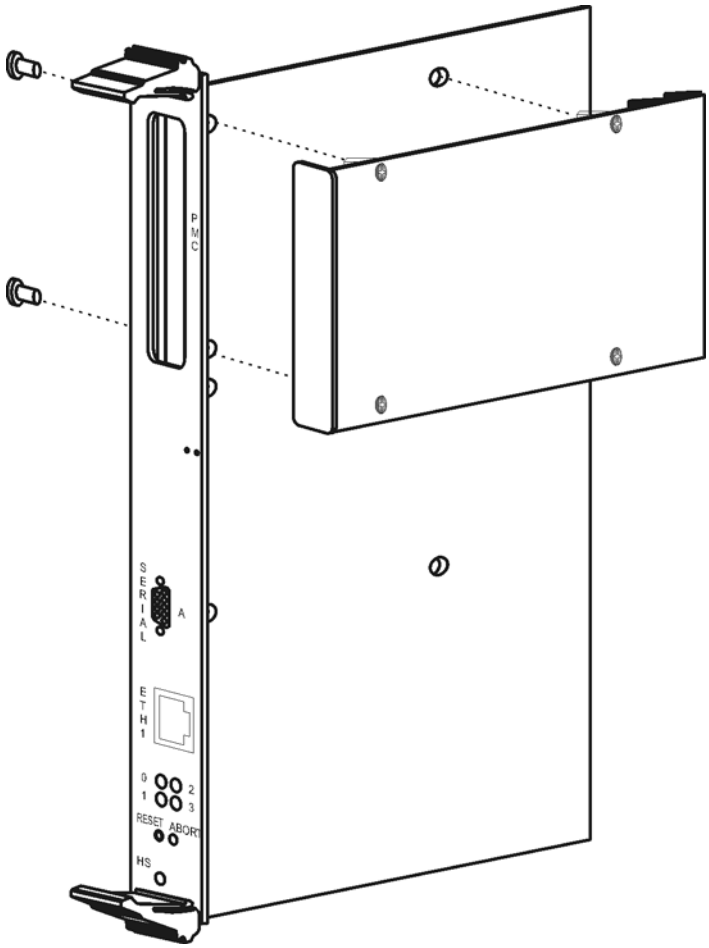
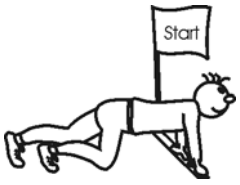


Figure 6: *Position of Mounting Holes*

5. Place screws delivered with PMC module into the mounting holes
6. Fasten screws



Removal Procedure



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



Rear Transition Board

As a separate price list item, Motorola offers a rear transition board (RTB), the RTB-505. The RTB provides access to the board's user I/O interfaces via industry standard connectors. The ACC/RTB-505, the accessory kit for the CPSB-560 board, contains the RTB itself and the user's documentation. The RTB-505 provides a IPMB1 and ICMB connector which can be used if your backplane does not provide these connectors.

Note: Only use the RTB-505 for boards of the SPARC/CPSB-560.

For information on the RTB's features and installation, refer to the *ACC/RTB-505 Installation Guide.*

Switch Settings

The CPSB-560 provides three switches which are located on the top side of the board. They can be switched without having to remove any module.

Caution



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

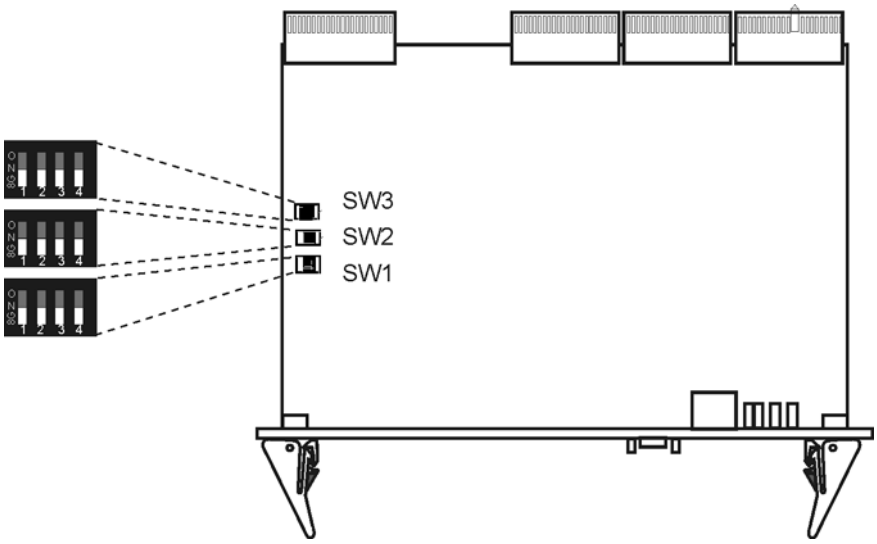
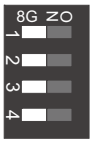
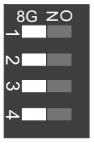


Figure 7: *Location of Switches on CPU Board*

Table 7: *Switch Settings*

Switch	No.	Description
SW1 	1	Disable CompactPCI reset OFF (default): CompactPCI reset of board enabled ON: CompactPCI reset of board disabled
	2	FORCE_PM[1)] OFF (default): The IPMI controller is BMC (SYSEN active) or PM (SYSEN inactive) ON: The IPMI controller is PM
	3	FORCE IPMI SYSEN[1)] OFF (default): The IPMI controller senses the CPCI_SYSEN ON: The IPMI SYSEN is active
	4	Ethernet interface 1 / 5 selection OFF (default): Ethernet 1 is available on CPSB-560's front panel ON: Ethernet 5 is available on RTB-505 front panel
SW2 	1	Boot device selection / Boot flash segmentation OFF (default): Boot from boot PROM / 8 MByte user flash ON: Boot from flash EPROM / 1 MByte boot flash, 7 MByte user flash
	2	Flash EPROM write protection (whole device) OFF (default): Whole flash EPROM is write-protected ON: Writing enabled for whole flash EPROM
	3	Flash EPROM write protection for the first MByte OFF (default): Flash EPROM writing disabled for the first MByte (boot section) ON: Flash EPROM writing enabled for the first MByte (boot section)
	4	Disable reset / abort key OFF (default): Reset / abort key enabled ON: Reset / abort key disabled
SW3 	1	Disable watchdog OFF (default): Watchdog disabled ON: Watchdog enabled
	2	**Reserved, must be OFF**
	3	**Reserved, must be OFF**
	4	**Reserved, must be OFF**

1) The CPSB-560 is BMC if switch SW1-2 is OFF and if switch SW1-3 is ON. All other settings of SW1-2 and SW1-3 result in the CPSB-560 acting as PM.

Board Installation

The SPARC / CPSB-560 supports PICMG 2.16, provides no CompactPCI bus and carries the appropriate compatibility glyph (see figure below).



Figure 8: *Compatibility Glyph*

Furthermore, it provides hot-swap and high-availability support, i.e it may be installed in or removed from a powered system. This section is divided into two subsections 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 CPSB-560 at multiple points EMC problems may occur.

Installation in a Nonpowered System

The SPARC / CPSB-560 can be inserted into systems with a signaling level of 3.3V and 5V.

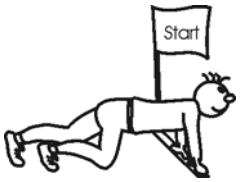
Note: Before installing the board, install the upgrades and accessories, if necessary (refer to "Hardware Upgrades and Accessories" page 2-9).

Installation 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. Check switch settings for consistency (see “Switch Settings” page 2-16)

Caution



Board damage
Fabric backplane slots marked with sign below



Installing the CPSB-560 into a fabric slot marked by the sign shown above damages the CPSB-560. Only install the CPSB-560 into node slots marked with the following glyphs.



3. Plug board into free node slot
4. Press handles inwards to lock board on backplane
5. Fasten board with screws
6. Plug in interface cables into front panel connectors, if applicable
7. Turn on system power

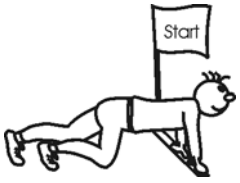


Removal Procedure

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 at front panel
3. Press red release button on both handles
4. Press handles outwards to disconnect board from backplane
5. Remove board from rails of slot position



6. Turn on system power

Installation in a Powered System Supporting Hot Swap

The board supports hot swap and high-availability. The basic purpose of hot-swap support is to allow the board to be installed and removed in a powered system without adversely affecting system operation. With hot-swap 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 SPARC/CPSB-560 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.

The SPARC / CPSB-560 can be inserted into systems with a signaling level of 3.3V and 5V.

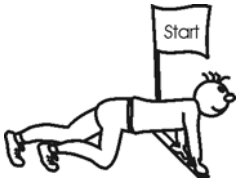
Installation Procedure

Before installing the board, observe the following:

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. Check board configuration e.g. switch settings
2. Check that you are using an appropriate rear transition board, if applicable

Caution



Board damage
Fabric backplane slots marked with sign shown below



Installing the CPSB-560 into a fabric slot marked by the sign shown above damages the CPSB-560. Only install the CPSB-560 into node slots marked with the following glyphs.



3. Insert board into free node slot of powered system
   The hot-swap LED stays blue until the board goes healthy.
4. Press handles inwards to lock board on backplane
5. Fasten board with screws on front panel



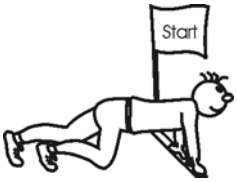
Removal Procedure

Before removing the board, observe the following:

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.
- **Board damage and data loss**
  Removing the SPARC/CPSB-560 from a powered system not supporting hot swap or high availability causes board damage and data loss. Therefore, only 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.
- **Data loss**
  Removing the board from a powered system with IDE devices attached to the board's secondary IDE interface via the RTB-505 results in data loss.



1. Loosen screws on front panel
2. Press red release button of lower handle
3. Open lower handle
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. Open upper handle by pressing red button
6. Remove board from powered system

Powering Up

For initial power up and configuration, a terminal can be connected to the front panel serial I/O connector A. The advantage of using a terminal is that you do not need any graphic card, monitor or keyboard. The board has successfully booted OpenBoot if LED 2 shines green.

By default, the SPARC/CPSB-560 is shipped with a 1 MByte boot PROM (PLCC) containing the OpenBoot firmware. To boot from boot PROM, set switch SW2-1 to OFF (default).

The boot PROM is not writeable, therefore, the CPSB-560 provides a 8 MByte flash EPROM which can be used as follows:

- As 1 MByte boot section and 8 MByte user area
  This configuration is needed if you want to use your own executable image or if you want to update the OpenBoot firmware (see "Adding Drop-In Drivers and Updating OpenBoot" page 4-26). The first MByte (used as boot section) can be write-protected by setting switch SW2-3 to OFF. To boot from boot section of the flash EPROM, set switch SW2-1 to ON (page 2-17).
- Completely as user area
  It can be programmed with an executable image which can be loaded and executed from user flash. To write protect the complete flash EPROM set switch SW2-2 to OFF.

Solaris Installation

The SPARC/CPSB-560 is designed to run with Solaris 8 Version 2/02 or higher with the 64-bit kernel.

Note:

- **Solaris versions prior to version 8 2/02 are not supported.**
- **SPARC/CPSB-560 runs with 64-bit kernel only.**

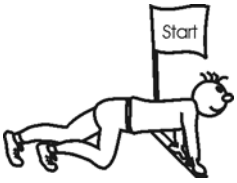
The following devices of the SPARC/CPSB-560 are not supported by the Solaris operating system:

- Intel 82559ER Ethernet device
- Intel 8254xEM GBit Ethernet device
- On-board flash EPROM
- Vitesse IPMI controller VSC215
- LEDs
- Ejector switch

If you wish to use these devices you need to install the Motorola Solaris Driver Package. For information on its availability, contact your local Motorola representative. For information on which driver must be installed for a particular device, see section "Solaris Driver Package" page 2-26.

You can install Solaris via an IDE or SCSI CD ROM drive attached to the RTB-505 or via Ethernet.

Via CD-ROM

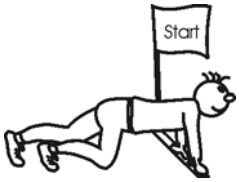


1. Insert Solaris CD labeled "Software 1 of 2" into CD drive
2. If you install Solaris via SCSI CD-ROM drive, enter
   `boot cdrom`
   If you install Solaris via IDE CD-ROM drive, enter
   `boot cdrom-2`



3. Follow on-screen instructions
   For further information on installing Solaris, refer to the Solaris documentation.

Via Network



1. Create an Install Server
   For information on how to create an Install Server, refer to the Sun Solaris documentation.
2. Enter **`boot net - install`**



3. Follow on-screen instructions
   For further information on installing Solaris, refer to the Solaris documentation.

Software Upgrades and Accessories

Motorola offers a Solaris Driver Package.

Solaris Driver Package

The Solaris Driver Package supports the following devices:

- Intel 82559ER Ethernet device
- Intel 8254xEM GBit Ethernet device
- On-board flash EPROM
- Vitesse IPMI controller VSC215
- LEDs
- Ejector switches

If you wish to use the devices above you need to install the Motorola Solaris Driver Package. For information on which driver must be installed for a particular device, see table below.

Note: Before installing the Solaris Driver Package uninstall the Solaris package SUNWdmfex by entering the command `pkgrm SUNWdmfex` into the Solaris prompt.

Table 8: *Devices and their Appropriate Drivers*

Device	Driver to be installed
Intel 82559ER Ethernet controller	FRCiprb
Intel 8254xEM Ethernet controller	FRCgei
On-board flash EPROM	FRCflash
LEDs, ejector switch	FRCctrl
IPMI managment controller	FRCipmi

For information on the instance numbers of the two Ethernet drivers, refer to the following section. For information on the driver installation, refer to the *Solaris Driver Package Rel. 2.17 Installation and Reference Guide*.

Driver Names and Instance Numbers of Ethernet Devices

The following table shows the driver names and instance numbers assigned to Ethernet interfaces. The names and instance numbers are needed to configure the network interfaces, i.e. for assigning an IP address to the Ethernet interfaces. For information on how to configure network interfaces, refer to the Solaris user's documentation.

Table 9: *Instance Number Assignement*

Ethernet Interface	Location	Driver Name and Instance Number
ETH 1[1)]	CPSB-560 front panel	eri0
ETH 5[1)]	RTB-505 front panel	eri0
ETH 2	Backplane	frcgei0
ETH 3	Backplane	frcgei1
ETH 4	RTB-505 front panel	fciprb0

1) This interface is either available as ETH 1 on the CPU board's front panel or as ETH 5 on the RTB's front panel. The selection is done via SW3-2.

Driver FRCipmi

The FRCipmi driver provides an application programming interface (API) for communication with the on-board IPMI controller Vitesse VSC215. The API can be used to program a system management application software, e.g. to:

- Execute IPMI commands
- Read the System Event Log (SEL)
- Read sensor values
- Set sensor threshold values
- Set the IPMI watchdog
- Read the geographical address

For further information on the Solaris IPMI driver package, refer to the *IPMI Reference Guide.*

3

Controls, Indicators, and Connectors

Front Panel

The following figure shows the position of the PMC cutout, the keys, the connectors and the LEDs on the CPSB-560 front panel.

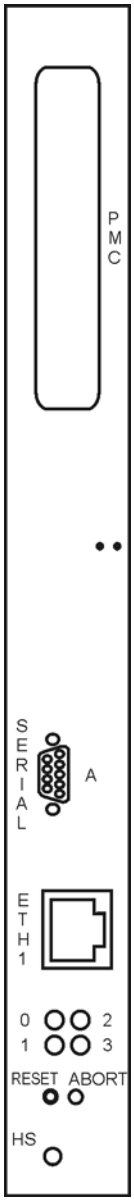


Figure 9: *Front Panel*

LEDs

The front panel provides five LEDs whose position can be seen in the following figure.

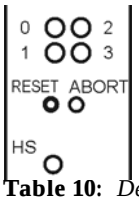


Table 10: *Description of Front Panel LEDs*

LED	Description
0	User LED0 Programmable via "LED 0 Control Register" page 6-15. **Note: Independent of its configuration, user LED 0 blinks red if PCI bus A has been idle for more than 1 second, i.e. the last assertion of PCI_FRAME has occurred more than 1 second ago.**
1	User LED1 Programmable via "LED 1 Control Register" page 6-16.
2	Shines green if board has sucessfully booted OpenBoot Otherwise, functions as user LED2 Programmable via "LED 2 Control Register" page 6-17.
3	User LED3 Programmable via "LED 3 Control Register" page 6-18.
HS	Hot swap LED: Indicates hot-swap status Blue: Board may be removed from the system OFF: Board must not be removed from the system

During power up, the settings of the user LEDs are as follows:

Table 11: *User LEDs During Power Up*

State	User LED	Status
Start CORE	All	LEDs are turned on and off consecutively
CORE execution	2	Flashing green (2 Hz)
CORE client (e.g. OpenBoot (FVM) or POST) execution	2	Flashing green (1 Hz)
OpenBoot (FVM) loading process finished	2	Green

Keys

The front panel provides two keys, the mechanical reset key and the abort key.



Reset Key

If the reset key is enabled via switch SW2-4 (default) and is toggled, the processor, all on-board and attached I/O devices are reset.

By default, the reset key is enabled. To disable the key, set switch SW2-4 to ON.

Abort Key

When enabled and toggled it instantaneously affects the SPARC/CPSB-560 by generating a push-button externally initiated reset (XIR). Push-button externally initiated reset allows a user-reset (abort) of part of the processor without resetting the whole system. UltraSPARC-IIi+ sets the B_XIR bit in the Reset Control register when a push-button externally initiated reset is detected.

Note: Only use the abort key if an application under Solaris or OpenBoot hangs. Do not use it to enter OpenBoot or to bypass the diagnostic routine during power up. After pressing the abort key, the board is in diagnostic mode and the OpenBoot ok prompt appears. In this mode, you can diagnose what caused the program to hang. However, the board is not fully initialized and therefore is not fully functional. To regain the full functionality, you need to press the reset key.

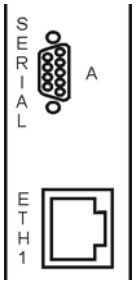
By default, the reset key is enabled. To disable the key, set switch SW2-4 to ON.

Connectors

The front panel provides the following connectors:

Note: Ethernet interface 1 is either available on the board's front panel as ETH1 or on the RTB-505 as ETH5. They cannot be used at the same time. The selection is done via switch SW3-2 (see Table 7 "Switch Settings" page 2-17).

- RJ-45 for Ethernet interface 1
- Serial interface A



If the board is to be incorporated in larger systems and adapted to specific needs, the following connector pinouts may be useful to give information on which signal is assigned to which pin.



Figure 10: *Ethernet 1 Connector Pinout*

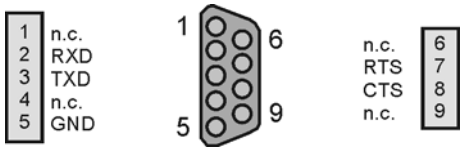


Figure 11: *Serial Interface A Connector Pinout*

IDE Connector

The on-board IDE connector is connected to the primary IDE interface. The figure below shows the IDE connector's position.

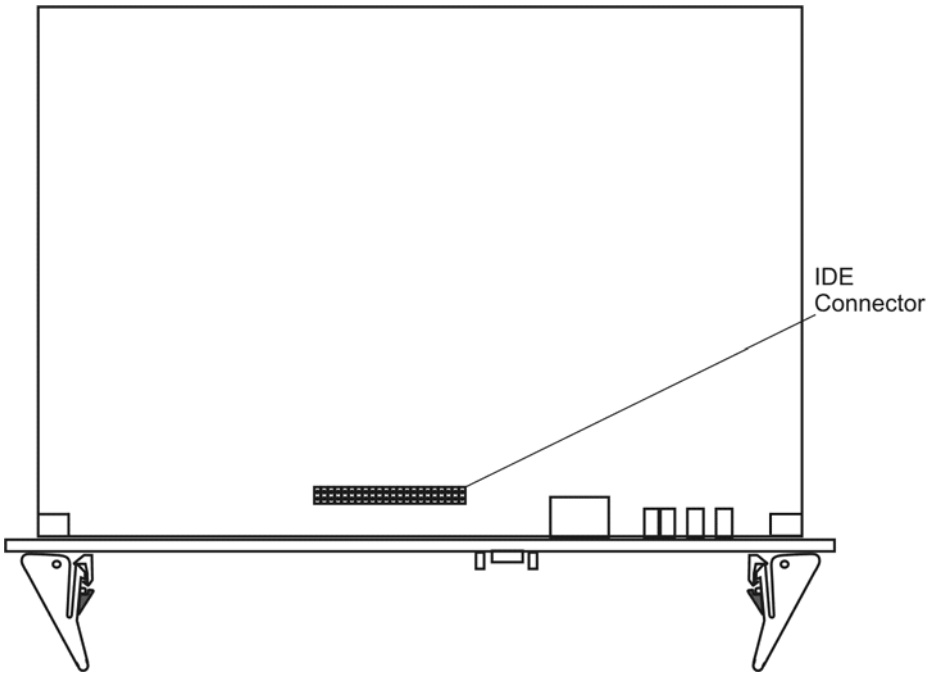


Figure 12: *Location of IDE Connector*

The pinout of the IDE connector is shown below.

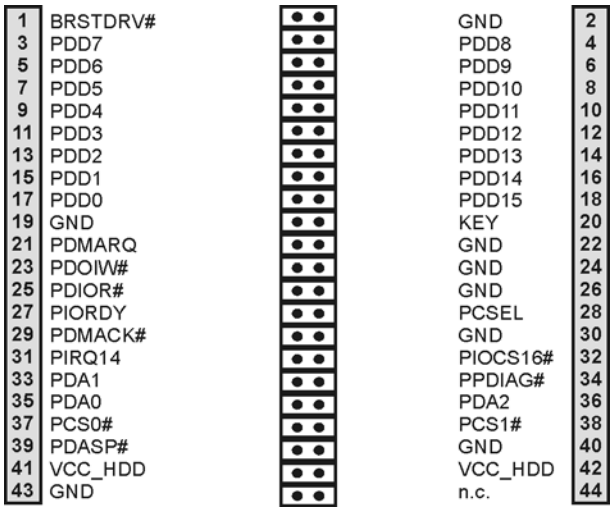


Figure 13: *IDE Connector Pinout*

CompactPCI Connectors

Note: The CPSB-560 has no CompactPCI interface.

The board provides the CompactPCI connectors J1, J2, J3, and J5.

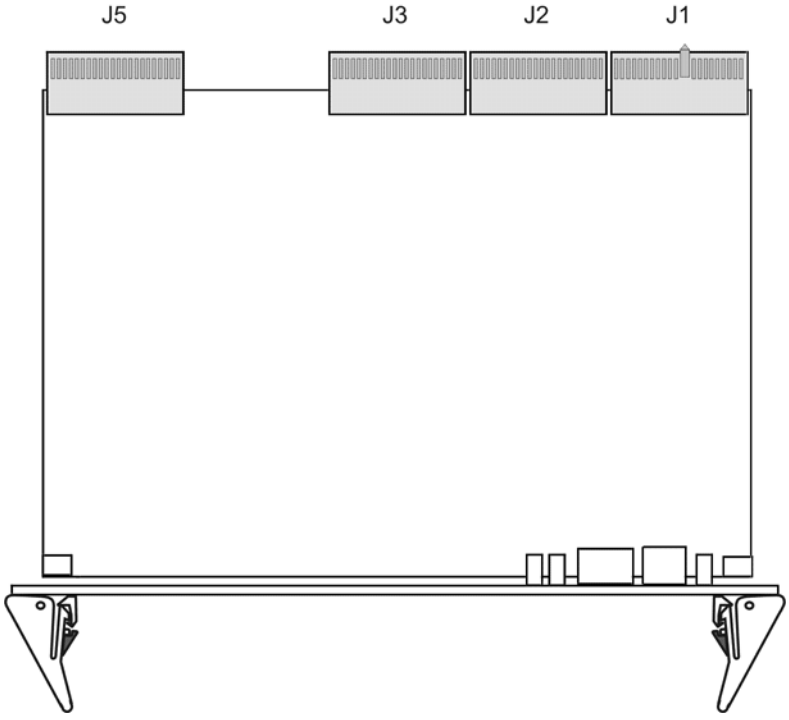


Figure 14: *Location of CompactPCI Connectors*

J1

Connector J1 provides the IPMB and IPMB0 signals.

	A	A B C D E	B	C	
25	5V		n.c.	n.c.	25
24	n.c.		5V	V(I/O) EARLY	24
23	3.3V		n.c.	n.c.	23
22	n.c.		GND	3.3V EARLY	22
21	3.3V		n.c.	n.c.	21
20	n.c.		GND	V(I/O)	20
19	3.3V		n.c.	n.c.	19
18	n.c.		GND	3.3V	18
17	3.3V		IPMB0 SCL	IPMB0 SDA	17
16	n.c.		n.c.	V(I/O)	16
15	3.3V		n.c.	n.c.	15
14	Key		Key	Key	14
13	Key		Key	Key	13
12	Key		Key	Key	12
11	n.c.		n.c.	n.c.	11
10	n.c.		GND	3.3V	10
9	n.c.		n.c.	n.c.	9
8	n.c.		GND	V(I/O)	8
7	n.c.		n.c.	n.c.	7
6	n.c.		PCIPRESENT_N	3.3V EARLY	6
5	n.c.		n.c.	RST_N	5
4	IPMB PWR		HEALTHY_N	V(I/O) EARLY	4
3	n.c.		n.c.	n.c.	3
2	CLK		5V	n.c.	2
1	5V		-12V	n.c.	1

Figure 15: *J1 Connector Pinout, Rows A-C*

	D	A B C D E	E	
25	3.3V		5V	25
24	n.c.		n.c.	24
23	5V EARLY		n.c.	23
22	n.c.		n.c.	22
21	n.c.		n.c.	21
20	n.c.		n.c.	20
19	GND		n.c.	19
18	n.c.		n.c.	18
17	GND		n.c.	17
16	n.c.		n.c.	16
15	BDSEL_N		n.c.	15
14	Key		Key	14
13	Key		Key	13
12	Key		Key	12
11	GND		n.c.	11
10	n.c.		n.c.	10
9	GND		n.c.	9
8	n.c.		n.c.	8
7	GND		n.c.	7
6	n.c.		n.c.	6
5	GND		n.c.	5
4	n.c.		n.c.	4
3	5V EARLY		n.c.	3
2	n.c.		n.c.	2
1	+12V		5V	1

Figure 16: *J1 Connector Pinout, Rows D and E*

J2

Connector J2 provides the IPMB1 signals.

	A	B	C	
22	GA<4>	GA<3>	GA<2>	22
21	n.c.	GND	n.c.	21
20	n.c.	GND	n.c.	20
19	GND	GND	IPMB1 SDA	19
18	n.c.	n.c.	n.c.	18
17	n.c.	GND	PRST_N	17
16	n.c.	n.c.	DEG_N	16
15	n.c.	GND	FAL_N	15
14	n.c.	n.c.	n.c.	14
13	n.c.	GND	V(I/O)	13
12	n.c.	n.c.	n.c.	12
11	n.c.	GND	V(I/O)	11
10	n.c.	n.c.	n.c.	10
9	n.c.	GND	V(I/O)	9
8	n.c.	n.c.	n.c.	8
7	n.c.	GND	V(I/O)	7
6	n.c.	n.c.	n.c.	6
5	n.c.	n.c.	V(I/O)	5
4	V(I/O)	n.c.	n.c.	4
3	n.c.	GND	n.c.	3
2	n.c.	n.c.	SYSEN_N	2
1	n.c.	GND	n.c.	1

A B C D E

Figure 17: *J2 Connector Pinout, Rows A-C*

	D	E	
22	GA<1>	GA<0>	22
21	n.c.	n.c.	21
20	GND	n.c.	20
19	IPMB1 SCL	IPMB1 ALERT_N	19
18	GND	n.c.	18
17	n.c.	n.c.	17
16	GND	n.c.	16
15	n.c.	n.c.	15
14	GND	n.c.	14
13	n.c.	n.c.	13
12	GND	n.c.	12
11	n.c.	n.c.	11
10	GND	n.c.	10
9	n.c.	n.c.	9
8	GND	n.c.	8
7	n.c.	n.c.	7
6	GND	n.c.	6
5	n.c.	n.c.	5
4	GND	n.c.	4
3	n.c.	n.c.	3
2	n.c.	n.c.	2
1	n.c.	n.c.	1

A B C D E

Figure 18: *J2 Connector Pinout, Rows D and E*

J3

Connector J3 provides interfaces to the following signals:

- PICMG 2.16 Packet Switched Ethernet interfaces 2 and 3
- SCSI (not available on all board variants, see “Ordering Information” for details)

	A	A B C D E	B	C	
19	GND		GND	GND	19
18	ETH2_MD0_P		ETH2_MD0_N	GND	18
17	ETH2_MD1_P		ETH2_MD1_N	GND	17
16	ETH3_MD0_P		ETH3_MD0_N	GND	16
15	ETH3_MD1_P		ETH3_MD1_N	GND	15
14	GND		GND	GND	14
13	SCSI_D12_P		SCSI_D12_N	V2P5_ETH2	13
12	SCSI_D14_P		SCSI_D14_N	V2P5_ETH3	12
11	SCSI_PAR1_P		SCSI_PAR1_N	SCSI_RESET_P	11
10	SCSI_D1_P		SCSI_D1_N	SCSI_RESET_N	10
9	SCSI_D3_P		SCSI_D3_N	TERM_PWR	9
8	SCSI_D5_P		SCSI_D5_N	TERM_PWR	8
7	SCSI_D7_P		SCSI_D7_N	TERM_OFF	7
6	SCSI_ATN_P		SCSI_ATN_N	n.c.	6
5	SCSI_ACK_P		SCSI_ACK_N	V3P3_RTB	5
4	SCSI_SEL_P		SCSI_SEL_N	V3P3_RTB	4
3	SCSI_REQ_P		SCSI_REQ_N	GND	3
2	SCSI_D8_P		SCSI_D8_N	SC_DIFFSENS	2
1	SCSI_D10_P		SCSI_D10_N	RTB_Present_N	1

Figure 19: *J3 Connector Pinout, Rows A-C*

	D	A B C D E	E	
19	GND		GND	19
18	ETH2_MD2_P		ETH2_MD2_N	18
17	ETH2_MD3_P		ETH2_MD3_N	17
16	ETH3_MD2_P		ETH3_MD2_N	16
15	ETH3_MD3_P		ETH3_MD3_N	15
14	GND		GND	14
13	SCSI_D13_P		SCSI_D13_N	13
12	SCSI_D15_P		SCSI_D15_N	12
11	SCSI_D0_P		SCSI_D0_N	11
10	SCSI_D2_P		SCSI_D2_N	10
9	SCSI_D4_P		SCSI_D4_N	9
8	SCSI_D6_P		SCSI_D6_N	8
7	SCSI_PAR0_P		SCSI_PAR0_N	7
6	SCSI_BSY_P		SCSI_BSY_N	6
5	SCSI_MSG_P		SCSI_MSG_N	5
4	SCSI_CD_P		SCSI_CD_N	4
3	SCSI_IO_P		SCSI_IO_N	3
2	SCSI_D9_P		SCSI_D9_N	2
1	SCSI_D11_P		SCSI_D11_N	1

Figure 20: *J3 Connector Pinout, Rows D and E*

J5

Connector J5 provides interfaces to:

- Ethernet 4 and 5 (Ethernet 4 is not available on all board variants, see “Ordering Information” for details)
- USB 1 and 2
- COM 1-3
- SUN Keyboard / Mouse
- IPMB 1
- Secondary IDE bus
- ICMB

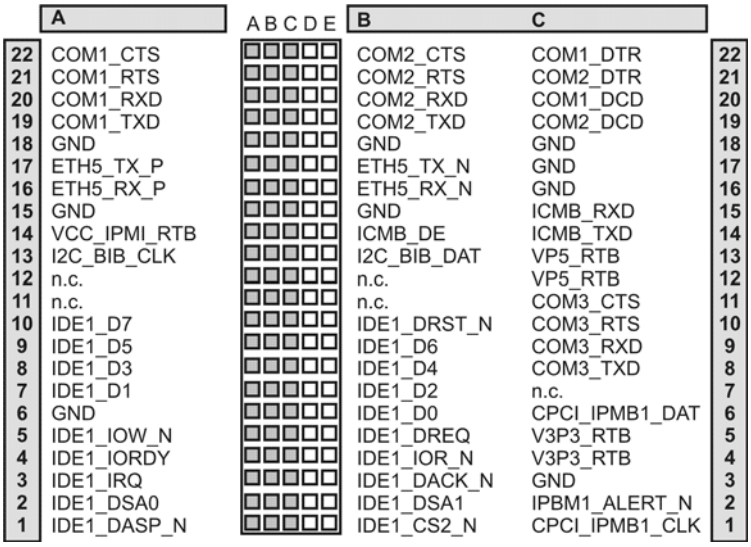


Figure 21: *J5 Connector Pinout, Rows A-C*

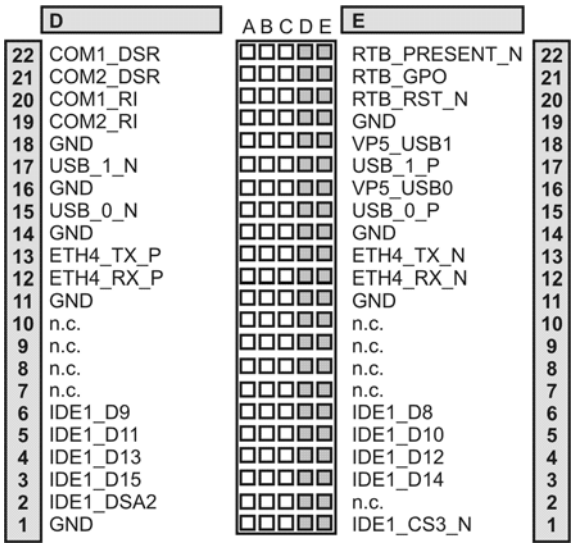


Figure 22: *J5 Connector Pinout, Rows D and E*

4

OpenBoot Firmware

Introduction

The OpenBoot firmware consists of the Common Operations and Reset Environment (CORE), the power-on self test (POST), and the OpenBoot itself which is also called Forth Virtual Machine (FVM).

During power up the CORE initializes the board and transfers control to one of its clients (e.g. POST, OpenBoot, or VxWorks BSP). In case of power up, the POST client is executed first if the NVRAM variable diag-switch? is set to true. Then the OpenBoot client Forth Virtual Machine (FVM) is started which finally is responsible to load the operating system Solaris.

The FVM contains a set of drop-in drivers needed either for booting the operating system, accessing and writing to the flash or initializing the onboard devices. Furthermore, a special diagnostics driver is included which is called OBDIAG (OpenBoot DIAGnostics). The OBDIAG provides a set of additional tests to those already performed during POST.

The functionality of the firmware parts is described in further detail in the following sections.

Note: The OpenBoot firmware is subject to changes. For the newest version and how to update refer to the Motorola website or the former Force Computers SMART service.

CORE

The CORE is responsible for setting up proper environments for booting purposes. It first performs a basic hardware initialization, loads a client (such as OpenBoot, VxWorks BSP, cPOST, ...), and transfers control to this client during power up.

Furthermore, it provides a unified interface for using public CORE functions. Thus, the CORE unifies system initialization and minimizes modifications in the upper level firmware.

The following figure gives a system overview of which clients and operating systems can be called by CORE.

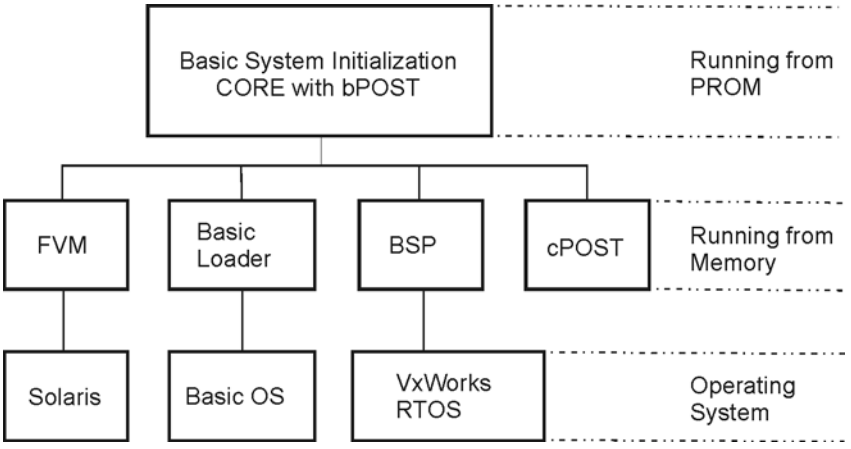


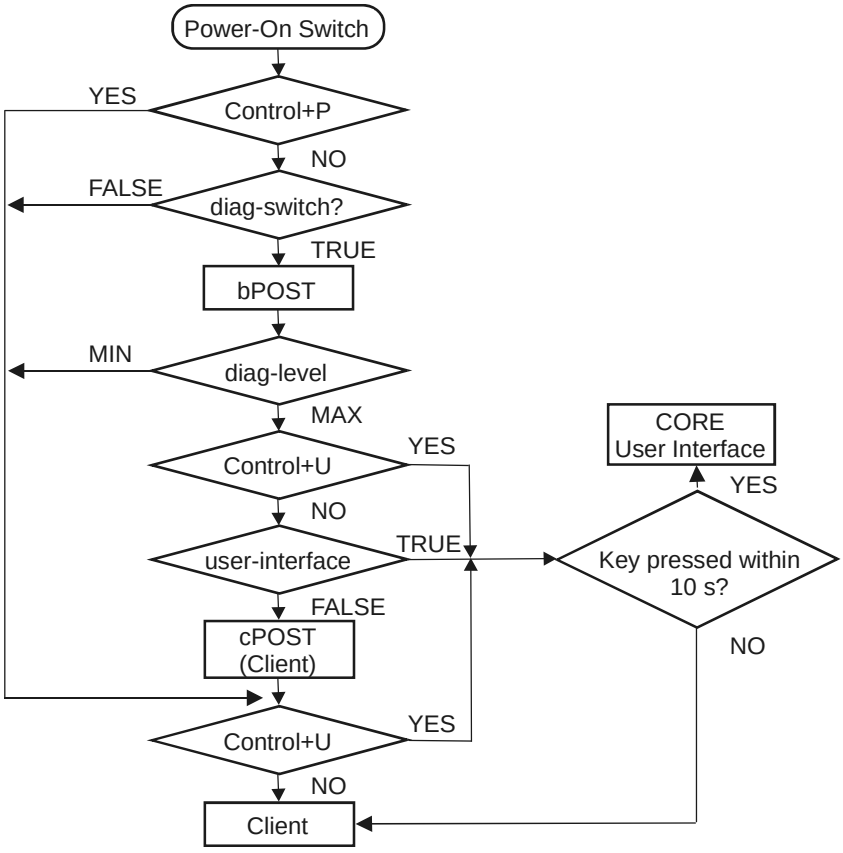
Figure 23: *System Overview*

Additionally, CORE is designed to perform the following tasks:

- Ability to use I/O devices including serial port, flash, and net early on the cold boot sequence of a firmware client.
- Basic system tests that can replace existing POST in min. mode.
- System testing may be done using the POST drop-in in max. mode.
- Developing standard validation test suites that could prevent major bugs in CORE and clients
- Sample client codes that could facilitate any client porting

CORE Workflow

The figure below shows the steps CORE executes during power up depending on the configuration variables.



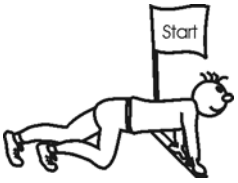
CORE Key Commands

In order to change or interrupt the boot process in CORE, the following key commands can be used if a terminal is used:

Note: If you use a SUN keyboard, use the key sequences below without <Control>.

- Skip POST: <Control> + <P>
- Enter CORE user interface after rebooting system: <Control> + <U>
- Use default NVRAM variables for this run: <Control> + <N>
- Turn on messages: <Control> + <M>

Booting a Specific Client



1. Reset or power on system

Note: If you use a SUN keyboard, use the key sequence below without <Control>.

---

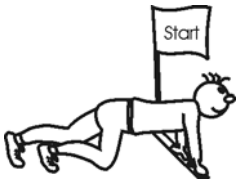
2. <Control> + <U>
3. Wait until user interface prompt appears
4. Press any key within 10 seconds to prevent automatic power up
5. View drop-ins by entering **`show-dropins`**



6. Start client with command **`execute <clientname>`**
   For <clientname> FVM (OpenBoot) or cPOST (extended POST) can be used.

Permanently Booting Another Client

By default, OpenBoot is activated. In order to permanently activate another boot client in CORE, do the following:



1. Reset or power on system

Note: If you use a SUN keyboard, use the key sequence below without <Control>.

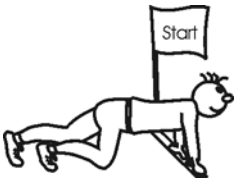
---

2. <Control> + <U>
3. Wait until user interface prompt appears
4. Press any key within 10 seconds to prevent automatic power up
5. Set NVRAM variable kernel by entering command
   `set-nvram kernel <Name>`
   where <name> is the name of the new client



6. Reset system by entering **`reset`**

Obtaining CORE Help



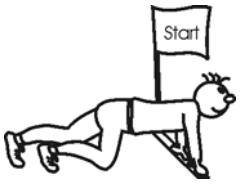
1. Reset or power on system

Note: If you use a SUN keyboard, use the key sequence below without <Control>.

2. <Control> + <U>
3. Wait until user interface prompt appears
4. Press any key within 10 seconds to prevent automatic power up
5. Show CORE help by entering **`help`**



POST



To execute POST when turning on the system or pressing the reset key, do the following:

1. Enter at ok prompt: **`setenv diag-switch? true`**
2. Optional: To set minimal testing, enter **`setenv diag-level min`**
   To set maximal testing, enter **`setenv diag-level max`**



3. Reboot board
   For each test a message is displayed on a terminal connected to the serial I/O interface A. If the system does not work correctly, error messages will be displayed which indicate the problem.

OpenBoot

The most important function of OpenBoot firmware is booting the operating system Solaris. This chapter explains how to boot the operating system and provides a list of the boot device aliases available on the CPSB-560.

Entering OpenBoot

To enter OpenBoot while the operating system boots or when the operating system is already running, press <~>+<#>.

Note: Do not press the abort key to enter OpenBoot. After pressing the abort key, the board is not fully initialized, and therefore, is not fully functional.

Booting the Operating System

The OpenBoot firmware holds its configuration parameters in NVRAM. Depending on these parameters the system is able to boot automatically from the specified device and with the specified file.

By default, it boots automatically the operating system after it is powered on and after it has passed the POST.

The NVRAM configuration variable auto-boot? is used to enable or disable the automatic boot process. The NVRAM configuration variable diag-switch? is used to decide whether to boot from the contents of the NVRAM configuration variables boot-device and boot-file or from variables diag-device and diag-file.

If the value of diag-switch? is false (default), the contents of boot-device and boot-file are evaluated for booting. Otherwise, the OpenBoot firmware uses the contents of diag-device and diag-file for booting.

Note: By default, the SPARC/CPSB-560 boots the operating system automatically. If this is not the case, ensure that the auto-boot? parameter is set to true.

To see a list of all available configuration parameters, enter the command printenv at the Forth Monitor prompt.

To set specific parameters, use the setenv command as shown below:

setenv <configuration_parameter> <value>

The configuration parameters in the following table are involved in the boot process.

Table 12: *Boot Configuration Parameters*

Parameter	Default Value	Description
auto-boot?	true	Defines whether to boot automatically or to enter into the OpenBoot prompt True: Automatic booting of operating system after power on or reset False: OpenBoot prompt appears
diag-switch?	false	Defines the operation mode (normal / diagnostic mode) True: Run in diagnostic mode, execute POST and boot from values set in variables <boot-device> and <boot-file> False: Run in normal operation mode, do not execute POST and boot from values set in variables <diag-device> and <diag-file>
boot-device	disk	Device from which to boot in normal mode By default, the device alias disk is set and therefore, OpenBoot boots from a SCSI disk with SCSI-target-ID 0. OpenBoot also provides device aliases for other boot devices. For information on other device aliases, see Table 14 “Device Alias Definitions for SCSI” page 4-13 and Table 15 “Device Alias Definitions for IDE” page 4-14.
boot-file	-	File to boot
diag-device	net	Device from which to boot in diagnostic mode For information on other device aliases, see Table 14 “Device Alias Definitions for SCSI” page 4-13 and Table 15 “Device Alias Definitions for IDE” page 4-14.
diag-file	-	File to boot in diagnostic mode

If necessary, you can explicitely initiate the boot process from the OpenBoot command interpreter if the variable auto-boot? is set to false. User-initiated booting either uses the default boot device or one specified by the user.

In order to boot the system from the default boot device, enter the command boot at the Forth Monitor ok prompt. The boot command has the following format:

boot <device-specifier> <filename> <-bootoption>

Possible parameters for the variables above are given in the table below.

Table 13: *Boot Parameters*

Parameter	Description	
<device-specifier>	Name (full path or alias) of the boot device Typical values are cdrom, disk, net or tape (see the boot device section page 4-13)	
<filename>	Name of program to be booted The filename parameter is relative to the root of the selected device. If no filename is specified, the boot command uses the value of the boot file NVRAM parameter. The NVRAM parameters used for booting are described in the following section.	
<-bootoption>	Bootoption may be one of the following:	
	Option	**Description**
	-a	Prompts interactively for device and name of boot file
	-h	Halts after loading program
	-r	Reconfigures Solaris device drivers after changing hardware configuration
	-v	Prints verbose information during boot procedure

To explicitly boot from the default disks using the Forth Monitor, enter **`boot disk`** or **`boot disk-2.`** This requires that the disk target ID is set to 0.

Otherwise, enter the following command at the Forth Monitor command prompt, to retrieve a list of all device alias definitions and device paths:

```
devalias
```

The following tables list some typical device aliases.

Table 14: *Device Alias Definitions for SCSI*

Alias	Description
scsi	SCSI
disk	Default disk SCSI-target-ID 0
diskf	Disk SCSI-target-ID f
diske	Disk SCSI-target-ID e
diskd	Disk SCSI-target-ID d
diskc	Disk SCSI-target-ID c

Table 14: *Device Alias Definitions for SCSI (cont.)*

Alias	Description
diskb	Disk SCSI-target-ID b
diska	Disk SCSI-target-ID a
disk9	Disk SCSI-target-ID 9
disk8	Disk SCSI-target-ID 8
disk7	Disk SCSI-target-ID 7
disk6	Disk SCSI-target-ID 6
disk5	Disk SCSI-target-ID 5
disk4	Disk SCSI-target-ID 4
disk3	Disk SCSI-target-ID 3
disk2	Disk SCSI-target-ID 2
disk1	Disk SCSI-target-ID 1
disk0	Disk SCSI-target-ID 0
tape (or tape0)	First tape drive SCSI-target-ID 4
tape1	Second tape drive SCSI-target-ID 5
cdrom	CD-ROM partition f, SCSI-target-ID 6

Table 15: *Device Alias Definitions for IDE*

Alias	Description
ide	IDE hard disk connected to primary IDE bus
disk-2	Default disk IDE disk 0
disk23	IDE disk 3
disk22	IDE disk 2
disk21	IDE disk 1
disk20	IDE disk 0
cdrom-2	CD-ROM partition f on IDE CD-ROM

Running Diagnostics

Besides several diagnostic commands executed from the user interface (ok prompt) there is also a special drop-in driver called OBDIAG. OBDIAG provides a terminal menu which allows the user to select a device-specific test and to influence the test depth as well as whether messages are printed during the test execution.

Diagnostic Commands

The Forth Monitor includes several diagnostic routines. These diagnostic routines let you check devices such as network controller, SCSI devices, memory, clock, and keyboard. User-installed devices can be tested if their firmware includes a self-test routine.

In the following, the diagnostic-specific OpenBoot commands are described in detail. The commands can be used to:

- Identify devices connected to the primary SCSI bus
- Probe SCSI buses
- Test device functions
- Monitor the clock function
- Monitor the network connection

Table 16: *Diagnostic Routines*

Task		Command	Page
Probe	Devices connected to the primary SCSI bus	probe-scsi	4-16
	All SCSI buses	probe-scsi-all [device-path]	4-16
	IDE devices at the on-board controller	probe-ide	4-17
	All IDE devices	probe-ide-all	4-17
Test	Specified device's self-test method	test [device-specifier]	4-18
	All devices with a built-in self-test method	test-all [device-specifier]	4-18
Monitor	Clock function	watch-clock	4-19
	Network connection via primary Ethernet	watch-net	4-19

probe-scsi

DESCRIPTION Identifies devices connected to the primary SCSI bus.

SYNTAX `probe-scsi`

PARAMETERS None

RETURNS None

EXAMPLE

```
ok probe-scsi
Primary UltraSCSI bus:
Target 0
  Unit 0   Disk     IBM     DCAS-32160W     S65A
Target 1
  Unit 0   Disk     IBM     DCAS-32160W     S65A
-
ok
```

probe-scsi-all

DESCRIPTION Identifies installed devices on all SCSI buses in the system below the specified device tree node. If device-path is omitted, the root node is used.

SYNTAX `probe-scsi-all (device path)`

PARAMETERS None

RETURNS The actual response depends on the devices on the SCSI buses.

Note: A terminal message as answer to the command probe-scsi-all can take up to two minutes.

EXAMPLE

```
ok probe-scsi-all
/pci@1f,0/scsi@2
Target 0
  Unit 0   Disk     IBM     DCAS-32160W     S65A
Target 1
  Unit 0   Disk     IBM     DCAS-32160W     S65A
-
ok
```

probe-ide

DESCRIPTION Identifies IDE devices at the on-board controller.

SYNTAX `probe-ide`

PARAMETERS None

RETURNS None

EXAMPLE

```
ok probe-ide
  Device 0  ( Primary Master )
          ATA Model: FUJITSU MHM2100AT

  Device 1  ( Primary Slave )
         Not Present

  Device 2  ( Secondary Master )
         Not Present

  Device 3  ( Secondary Slave )
         Not Present

ok
```

probe-ide-all

DESCRIPTION Identifies installed IDE devices on all IDE buses in the system below the specified device tree node. If device-path is omitted, the root node is used.

SYNTAX `probe-ide-all (device path)`

PARAMETERS None

RETURNS The actual response depends on the IDE devices attached.

EXAMPLE

```
ok probe-ide-all
/pci@1f,0/pci@1,1/ide@3
  Device 0  ( Primary Master )
          ATA Model: FUJITSU MHM2100AT

  Device 1  ( Primary Slave )
         Not Present

  Device 2  ( Secondary Master )
         Not Present

  Device 3  ( Secondary Slave )
         Not Present


ok
```

test

DESCRIPTION Executes the specified device's self-test method. <Device-specifier> may be a device path name or a device alias.

SYNTAX `test (device-specifier)`

PARAMETERS None

RETURNS None

EXAMPLE To test the network connection, enter test net.

test-all

DESCRIPTION All devices below the root node of the device tree are tested. The response depends on the devices having a self-test method. If a device specifier option is supplied at the command line, all devices below the specified device tree node are tested.

SYNTAX `test-all`

PARAMETERS None

RETURNS None

watch-clock

DESCRIPTION Monitors the clock function.

SYNTAX `watch-clock`

EXAMPLE

```
ok watch-clock
Watching the 'seconds' register of the real time clock chip.
It should be 'ticking' once a second.
Type any key to stop.
26
ok
```

The system responds by incrementing a number every second. Press any key to stop the test.

watch-net

DESCRIPTION Monitors the primary network connection.

SYNTAX `watch-net`

PARAMETERS None

RETURNS None

EXAMPLE

```
ok watch-net
gme register test --- succeeded.
Internal loopback test -- succeeded.
Checking Link  -- Using Onboard Transceiver - Link Up.
Using Onboard Transceiver - Link Up.
Looking for Ethernet Packets.
'.' is a Good Packet.  'X' is a Bad Packet.
Type any key to stop.
.........................
ok
```

The system monitors the network traffic. It displays a dot (.) each time it receives a valid packet and displays an X each time it receives a packet with an error which can be detected by the network hardware interface.

OBDIAG

There are two different methods to execute OBDIAG:

a) Open OBDIAG by entering the command obdiag at the ok prompt

b) Set the configuration variable mfg-mode to chamber and set the variable diag-switch? to true. To set the variable mfg-mode to chamber, enter:
`setenv mfg-mode chamber`

When setting the variable mfg-mode to chamber a script of additional diagnostic tests is executed automatically after each POST from OBDIAG provided the POST has been running during power up without failure.

During the start-up sequence, OpenBoot searches for the presence of devices on all expansion busses and evaluates their characteristics such as device ID, device type, vendor ID, and revision ID. In order to test the hardware, OBDIAG requires self-test methods for the discovered devices. If OBDIAG does not find any self-test methods in the device nodes, it looks for its own self-test methods.

The more devices are detected, the more devices will appear in the OBDIAG main menu. This means that OBDIAG automatically adapts itself to the number of present hardware devices.

The OBDIAG main menu can be called by entering obdiag after the ok prompt of the OpenBoot firmware. The following figure shows the main menu mask of OBDIAG.

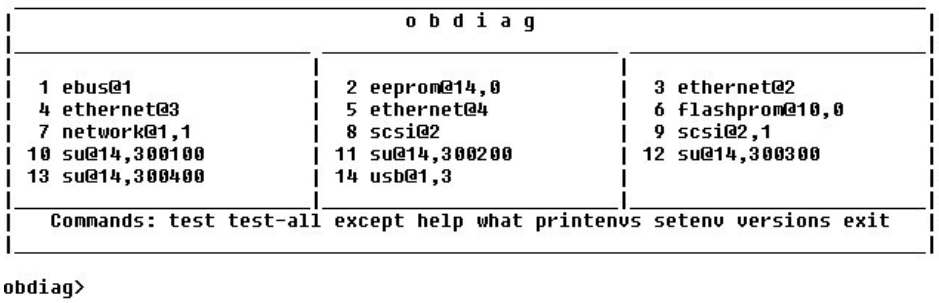


Figure 24: *OBDIAG Main Menu Mask*

When OBDIAG is entered, the obdiag test prompt appears and you can now choose the required test. You can run single tests, a number of tests, all tests, or all tests with exceptions. Apart from testing the hardware, you can also call several commands which are available from the ODBIAG main menu. The following table provides an overview of these commands.

Table 17: *OBDIAG Commands*

Command	Description	
exit	Exits obdiag tool	
help	Prints this help information	
setenv	Sets diagnostic configuration variable to new value (see values of printenv below)	
printenvs	Prints values for diagnostic configuration variables Possible variables are as follows:	
	diag-verbosity	Extent to which the test results are printed on the screen. 0 (default): Only error messages are displayed. 1: Extra test information is displayed. 2: Subtest names are printed. 3: Test debugging information is printed.
	diag-continue?	Defines whether the test is stopped if an error was found. 0: The self-test will be aborted if an error was detected. 1: If errors are found the test will be continued.
	diag-switch?	True: CPU stands in diagnostic mode and the POST will run with a lot of screen output. False: No POST and no OBDIAG script will be executed. However, OBDIAG can be executed if entering obdiag at the prompt.

Table 17: *OBDIAG Commands (cont.)*

Command	Description	
	diag-targets	Specifies how wide the OpenBoot diagnostics will reach in testing the devices, e.g. if a serial interface will be tested internally. For an external test, use a loopback connector. 0: Internal testing only 1: Bus path to devices 4: Perform I/O to media if possible 10: External loopback 1 20: External loopback 2 30: External loopback 1 and 2 40: External loopback 3 80: Test without main memory
	diag-passes	Specifies the number of executions or loops OBDIAG will perform for each self-test.
	diag-level	Specifies the quality of the test. off: No test will be performed. min: Minimal or quick level of testing max: Maximum or extensive level of testing menus: POST menu controlled diagnostics
versions	Prints self-tests, library, and obdiag tool versions	
test-all	Tests all devices displayed in the main menu	
test x,y,z	Tests devices x, y, and z	
except x,y	Tests all devices except for devices x and y	
what x,y,z	Prints some selected properties for devices x, y, and z	

EXAMPLE If you want to test all but devices 5 and 8, enter:

```
except 5,8
```

You can decide whether the chosen test will either stop at the occurrence of the first error or continue testing the hardware. It is also possible to run the test more than once or produce a detailed print-out of the test.

If the test has passed successfully, a short test comment will appear on the screen. In order to return to the main menu after the test has finished, enter

```
cr
```

In order to terminate OBDIAG and return to OpenBoot, enter at the obdiag prompt

exit

The OpenBoot prompt will then reappear.

The example below shows the detailed print-out of an OBDIAG test.

```
obdiag> setenv diag-verbosity 2
diag-verbosity =      2

Hit any key to return to the main menu <cr>

obdiag> setenv diag-continue? 0
diag-continue? =      0

Hit any key to return to the main menu  <cr>

obdiag> test 1
Hit the spacebar to interrupt testing
Testing /pci@1f,0/pci@1,1/ebus@1
SUBTEST: vendor-id-test
SUBTEST: device-id-test
SUBTEST: mixmode-read
SUBTEST: e2-class-test
SUBTEST: status-reg-walk1
SUBTEST: line-size-walk1
SUBTEST: latency-walk1
SUBTEST: line-walk1
SUBTEST: dma-reg-test
SUBTEST: dma-func-test
SUBTEST: tcr-reg-test
SUBTEST: flauxio-reg-test
SUBTEST: modauxio-reg-test
Selftest at /pci@1f,0/pci@1,1/ebus@1
..................................... passed

Hit any key to return to the main menu  <cr>
obdiag> exit <cr>
ok
```

Displaying System Information

The Forth Monitor provides several commands to display system information such as the system banner, the primary Ethernet address, the contents of the ID PROM and the version number of the OpenBoot firmware. The following table lists these commands.

Table 18: *Commands to Display System Information*

Task		Command
Display	System banner with Ethernet address and host ID	banner
	Ethernet address	.enet-addr
	ID PROM contents, formatted	.idprom
	List of SPARC trap types	.traps
	Version and date of the boot PROM	.version
	List of all device tree nodes	show-devs
	List of all device aliases	devalias

Resetting the System

If your system needs to be reset, there are two possibilities:

- Software reset
  For this type of reset, use the command reset at the Forth command line. The system begins with the initialization procedures but no POST is executed.
- Button power-on reset
  The system begins with the initialization procedures and POST is executed if the NVRAM configuration variable diag-switch? is set to true.

Activating OpenBoot Help

The Forth Monitor contains an online help which can be activated by entering the command help. The following screen output or a similar one will appear.

```
Enter 'help command-name' or 'help category-name' for more help
(Use ONLY the first word of a category description)
Examples:  help select   -or-   help line
    Main categories are:
SMC (System Management Control)
Breakpoints (debugging)
Repeated loops
Defining new commands
Numeric output
Radix (number base conversions)
Arithmetic
Memory access
Line editor
System and boot configuration parameters
Select I/O devices
Floppy eject
Power on reset
Diag (diagnostic routines)
Resume execution
File download and boot
nvramrc (making new commands permanent)
ok
```

A list of all available help categories is displayed. These categories may also contain subcategories. To get help for special Forth commands or subcategories, enter

help <name>

The online help shows you the Forth commands, the parameter stack before and after execution of the Forth command (before -- after), and a short description.

The online help of the Forth monitor is located in the boot PROM. This means that an online help is not available for all Forth commands.

Typical examples for how to get help for special Forth commands or subcategories are given below.

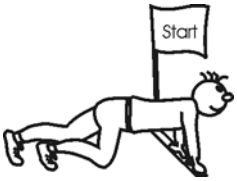
```
ok help power
reset-all   reset machine, ( simulates power cycling )
power-off   Power Off
ok
```

Adding Drop-In Drivers and Updating OpenBoot

At board delivery, OpenBoot is stored in the boot PROM. Due to the fact that the boot PROM is a read-only device, it is not possible to update the OpenBoot firmware, to add or delete an OpenBoot driver into the boot PROM.

If the writeable flash is used as the boot device, the OpenBoot firmware can be modified or a single drop-in driver can be added. In this case the flash is used as boot and user flash. If you want to add or delete drop-in drivers or to update OpenBoot, the contents of the boot PROM must first be copied to the boot section of the flash since the boot PROM is not writeable.

To copy the OpenBoot image from the boot PROM into the flash, do the following:



1. Set SW2-1 and SW2-3 to ON to disable boot flash write protection
2. If not already done, set SW2-2 to OFF to select boot PROM as boot device
3. Install board
4. Switch on system
   The board is booted from boot PROM



5. Copy OpenBoot image into flash by entering
   `plcc2tsop`
   The OpenBoot image is copied from the boot PROM into the flash while printing the following messages to the screen.

```
ok plcc2tsop


  COPY OBP from PLCC to TSOP boot flash
  -------------------------------------

  Please enter 'y' to execute boot flash copy
  or any other key to abort
Step1: Mapping memory space at addr 0x1.0000
       Done
Step2: Erasing TSOP Boot flash ...
BOOT flash memory is selected for programming
No BOOT flash memory is available
8192 Kbyte USER flash memory is available at 0xfe800000

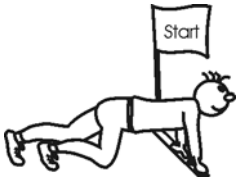
Erasing selected part of flash memory ...  DONE
       Done
Step3: Updating TSOP Boot flash ...
       Done

  COPY OBP into TSOP successful
  -----------------------------

  Now you have to switch SW 2-2 to the ' ON' position
  and thereafter reset the system
ok
```

Before you proceed with adding / deleting a drop-in driver or updating OpenBoot, set switch SW2-2 to ON and reset the board.

In case of an error during one of the steps described above, proceed as follows:



1. Select booting from boot PROM
2. Reset system
3. Try again



Drop-In Drivers

OpenBoot supports drop-in drivers, i.e. drivers that may be added to OpenBoot during start-up. The drop-in drivers are placed inside a specific drop-in driver area within the EPROM. Thus, they are available even after the system has been powered down.

The drop-in drivers are special FCode drivers having a unique drop-in driver header. At certain instants during start-up, OpenBoot scans the drop-in driver area for specific drivers to be loaded at specific instants. Each drop-in driver may be dedicated to a specific device and is loaded to the corresponding device node if the probing algorithm has identified a device whose device ID and vendor ID is equal to a specific drop-in driver.

OpenBoot contains commands to display all available drop-in drivers, to add them and to remove them. Thus, the drop-in drivers can be loaded at any time to the OpenBoot image from net or any other external media.

The table below illustrates the PCI-based drop-in drivers already present and gives useful information on the supported hardware.

Table 19: *PCI-Based FCODE Drivers Compared to Supported Hardware Devices*

Driver Name	Hardware Support for <device,vendor-ID>	Description
pci1000,3	1000,0003 1000,000c 1000,000f 1000,008f 1092,8000 1000,1588 1000,1001 1000,1000	Symbios SCSI FCODE device driver, e.g. Symbios 53C875, 53C895, etc.
pci1002,5654	1002,5654	ATI Mach64 VT graphic FCODE device driver
pci1002,4755	1002,4755,1002,4756	ATI Mach64 GU graphic FCODE device driver
pci1002,4750	1002,4750	ATI Mach64 GP graphic FCODE device driver
pci8086,1004	8086,1004	Intel Ethernet FCODE device driver, e.g. Intel i8254x, i82543, i82544
pci8086,1209	8086,1209	Intel Ethernet FCODE device driver, e.g. Intel i82559, i82559ER
class010100	-	Generic IDE FCODE class driver (see note below)

Table 19: *PCI-Based FCODE Drivers Compared to Supported Hardware Devices (cont.)*

Driver Name	Hardware Support for <device,vendor-ID>	Description
class060400	1011,0022	Generic PCI-to-PCI FCODE class driver, e.g. Intel DEC21150
pci108e,1101	108e,1101	RIO Ethernet FCODE device driver

Note: The generic IDE FCODE class driver is not supported by Motorola. Therefore, Motorola cannot guarantee that it works properly with all IDE controllers available on the market.

The following commands allow to add, delete or to display a drop-in driver.

Table 20: *Drop-In Drivers*

Task		Command	Page
Add	Drop-in driver to OpenBoot	add-dropin	4-29
Delete	Drop-in driver from OpenBoot	delete-dropin	4-30
Show		show-dropins	4-31

add-dropin

DESCRIPTION Adds a drop-in driver to OpenBoot. This can be done at any time. To use the command add-dropin, the device from which the driver is to be loaded must be specified. For a list of device aliases, refer to Table 14 “Device Alias Definitions for SCSI” page 4-13 and Table 15 “Device Alias Definitions for IDE” page 4-14.

After the command add-dropin has been executed, it is recommended to enter reset in order to update the new value.

SYNTAX `add-dropin <[device-alias:,]> <file name>`

PARAMETERS `<[device-alias:,]> <file name>`

RETURNS None

EXAMPLE To add the drop-in driver obdiag.di to the current OpenBoot from network port 3, for example, enter the following command:

```
add-dropin net3:, obdiag.di
```

```
fe40e9f0
Server IP address: 192.168.60.16
Client IP address: 192.168.60.18
ebb0 Bytes
BOOT flash memory is selected for programming
1024 Kbyte BOOT flash memory is available at 0xfe200000
7168 Kbyte USER flash memory is available at 0xfdb00000
Erasing FLASH...
Erasing selected part of flash memory ...  DONE
Done.
Programming FLASH... Done.
Verifying FLASH PROM Done.
ok
```

By entering show-dropins, it is visible whether the new driver was saved.

delete-dropin

DESCRIPTION Deletes a drop-in driver from OpenBoot. This can be done at any time. The <dropin-name> has to be entered without the extension .di. Otherwise, you get an error message.

SYNTAX `delete-dropin <dropin-name>`

PARAMETERS `dropin-name`

RETURNS None

EXAMPLE

```
delete-dropin obdiag
liag deleted!!
lating changes to PROM...
IT flash memory is selected for programming
!4 Kbyte BOOT flash memory is available at 0xfe200000
;8 Kbyte USER flash memory is available at 0xfdb00000
ısing FLASH...
ısing selected part of flash memory ...  DONE
ıe.
ıgramming FLASH... Done.
'ifying FLASH PROM Done.
```

show-dropins

DESCRIPTION Displays a list of all existing drop-in drivers.

SYNTAX `show-dropins`

PARAMETERS None

RETURNS None

Updating OpenBoot

After copying the OpenBoot firmware into the flash, the OpenBoot image can easily be updated.

The following OpenBoot command provides the possibility to update the OpenBoot image from various devices: disk, CD-ROM or network. To update the image, enter the command:

`flash-update <[device-alias:,]> <filename>`

For available device aliases, see Table 14 “Device Alias Definitions for SCSI” page 4-13 and Table 15 “Device Alias Definitions for IDE” page 4-14. If you do not enter a device alias (“[]” means the parameter device-alias is optional) OpenBoot automatically uses the primary network for updating the OpenBoot firmware. In this case, a TFTP server is required and has to be set up before using this command. During command execution a file is transferred from the server’s directory /tftpboot and is stored into the local memory to update the flash.

Note:

- **For information on how to set up a TFTP server, refer to the *OpenBoot Application Note Upgrading OpenBoot* which can be found on the Motorola website or the former Force Computers SMART server.**
- **The read-only flash device which contains the OpenBoot image and is plugged into the PLCC socket is always available. Therefore, you can always select it again for booting in case any problems have occurred during the updating procedure.**

The screen output after entering the update command could look like this:

```
ok flash-update obp.image
Getting new FLASH PROM image
Boot device: /pci@1f,0/pci@1,1/network@1,1:,obp.image  File and args:
Using Onboard Transceiver - Link Up.
100000
Server IP address: 192.168.60.16
Client IP address: 192.168.60.18
BOOT flash memory is selected for programming
1024 Kbyte BOOT flash memory is available at 0xfe200000
7168 Kbyte USER flash memory is available at 0xfdb00000
Erasing   FLASH PROM
Erasing selected part of flash memory ...  DONE
Done.
Updating  FLASH PROM Done.
Verifying FLASH PROM Done.
ok
```

5

Devices, Features and Data Paths

Block Diagram

The block diagram shows to which buses the CPSB-560 devices are attached.

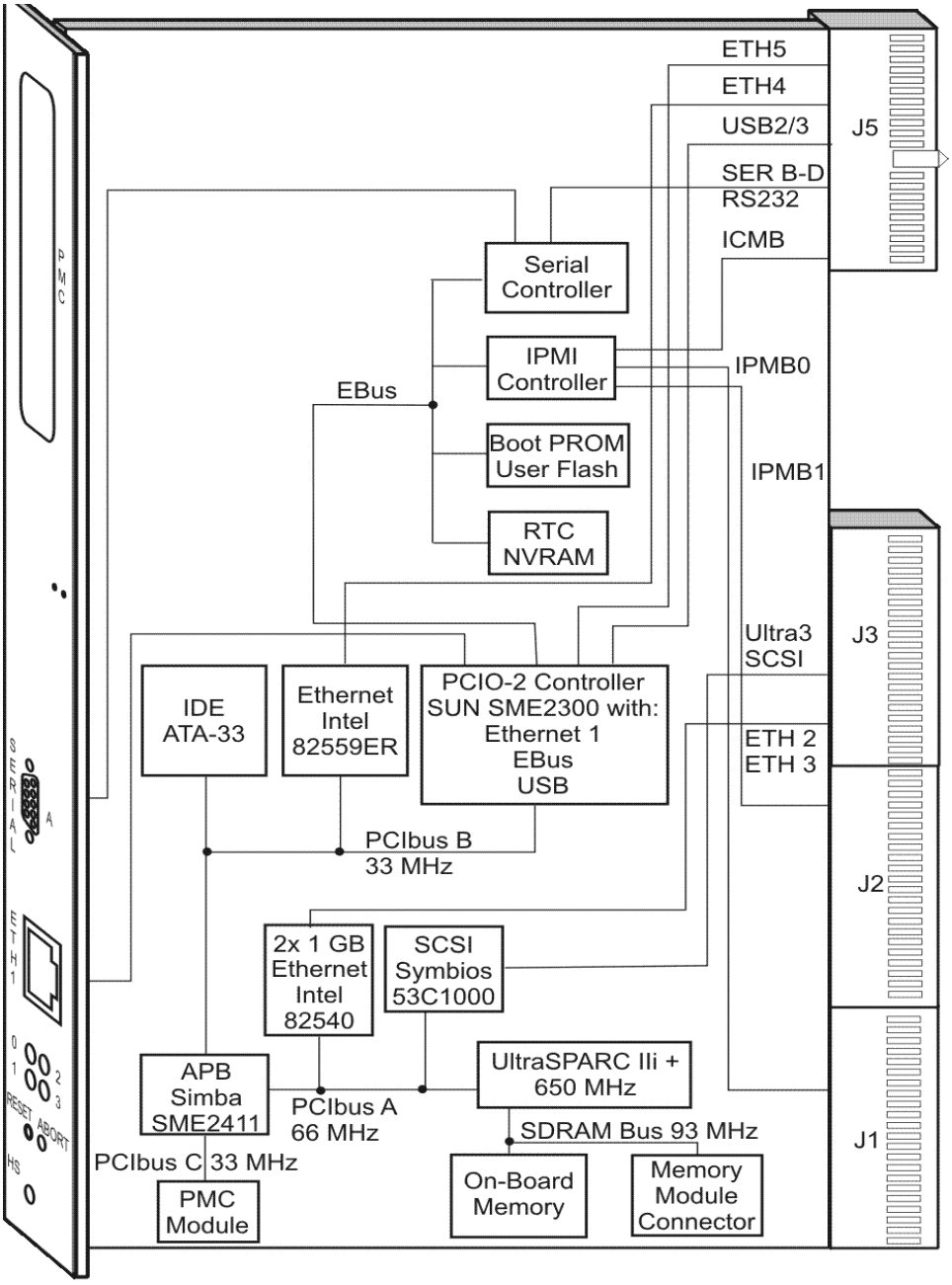


Figure 25: *Block Diagram*

PCI Bus A

PCI bus A runs at 66 MHz and is 32-bit wide. The following devices are connected to PCI bus A:

- UltraSPARC IIi + with a core frequency of 650 MHz and a 66 MHz PCI bus as bus for I/O extensions.
- Two GigaBit Ethernet controllers Intel 8254xEM for Ethernet interfaces 2 and 3 routed in the PSB backplane
  A driver for this controller is available in the Motorola Solaris Driver Package. The user LEDs 1 and 2 on the front panel can be configured by software to indicate link activity of the 1 GBit Ethernet interfaces 2 and 3. The Ethernet address is stored in the respective ID ROM. After power-on the Ethernet address is copied into the CSR space where it can be read by software.
- Symbios PCI-to-Ultra160 SCSI controller 53C1000
  The controller is a 32/64-bit PCI device. The SCSI interface is available on the RTB-505 via J3.
  Note: the SCSI controller is not available on all board variants. See the "Ordering Information" for details.
- Advanced PCI Bridge (APB) SME2411 from SUN Microsystems.
  The APB is a PCI-to-dual-PCI bridge which drives the two secondary PCI buses B and C.

For further information on these components, refer to the respective data sheets.

PCI Bus B

PCI bus B runs at 33 MHz and is 32-bit wide. The following devices are connected to PCI bus B:

- ATA-33 IDE controller
- Ethernet controller Intel 82559ER for 10/100BaseT Ethernet interface 4
- PCIO-2 controller RIO (SUN SME2300)

For further information, refer to the following sections and the respective data sheets.

IDE Interface

A 2,5" hard-disk drive (e.g. the Motorola ACC/CPSB-560/HD Accessory Kit) can be attached to the primary IDE interface via the on-board IDE connector. It is directly mounted on the CPU board via four standoffs and is connected with a 44-pin flat ribbon cable.

Note: Either an IDE hard disk or a memory module can be installed on the CPSB-560. If you want to install both, you can use an IDE PMC module in the PMC slot.

Another IDE device can be attached to the secondary IDE interface which is routed to the RTB-505 via J5.

Ethernet Interface 4

Ethernet interface 4, which is not available on all board variants, is accessible via RTB-505. For information on board variants that provide Ethernet interface 4, refer to the "Ordering Information".

A driver for the Ethernet controller is available in the Motorola Solaris Driver Package.

User LED 3 on the front panel can be configured to indicate link activity of the Ethernet interface 4 via the LED 3 Control register (see page 6-18).

The Ethernet address is stored in the respective ID ROM. After power-on the Ethernet address is copied into the CSR space where it can be read by software.

PCIO-2

The PCIO-2 provides the following:

- 10/100 Mbit BaseT Ethernet interface 1
  Ethernet 1 is alternatively available on the CPSB-560's front panel or on the RTB-505 as Ethernet 5. The two interfaces cannot be used at the same time and are selected via switch SW1-4. User LED 0 on the front panel can be configured to indicate link activity of the Ethernet interface 1/5 via the LED 0 Control register (see page 6-15).
- Twisted-pair Ethernet PHYceiver Intel LXT970
  The PHYceiver address for the front panel is 01_{16} and the PHYceiver address for the RTB-505 is 01_{16}.
- USB interfaces 0 and 1
  Available on the RTB-505 via J5.
- Interface to the EBus

Information on the address range of the PCIO-2, see Table 27 "PCI Bus Address Map" page 6-8.

EBus

The EBus is a generic slave 8-bit wide Direct Memory Acces (DMA) bus (pseudo ISA bus) to which the following devices are connected:

- Boot PROM
- Flash EPROM
- Real time clock and NVRAM
- Serial communication controller
- Xilinx FPGA with
  - Watchdog
  - Timer
  - On-board registers
- IPMI controller

The PCIO-2 PCI-to-EBus controller delivers eight decoded chip select signals:

- EBus_CS#0 (16 MByte space)
- EBus_CS#1..EBus_CS#7 (each 1 MByte space)

It thereby supports eight single or multi-function 8-bit devices with a minimum of glue logic. The memory map can be found in Table 28 “PCIO-2 Address Map” page 6-9.

Boot PROM and Flash EPROM

The PCIO-2 16 MByte chip select signal EBus_CS#0 is decoded into two chip select signals for:

- One boot PROM with 1 MByte address space
- One flash EPROM memory device with up to 15 MByte address space It can either be completely used as user flash or as flash with a 1-MByte boot section and 7-MByte user section. The first MByte can be used as an alternative boot area, for example, if a drop-in driver is added or the OpenBoot image is updated.

To select whether to boot from boot PROM with the OpenBoot image or from the boot area of the flash EPROM, use switch SW2-1.

Before copying the OpenBoot image from the boot PROM into the boot area (first MByte) of the flash EPROM, set switch SW2-1 to OFF and SW2-3 to ON.

To protect the boot image in the boot section of the flash EPROM, set switch SW2-3 to OFF.

To protect the whole flash EPROM from writes, set switch SW2-2 to OFF (default).

For the base addresses of EBus_CS#0 and EBus_CS#2, see Table 28 “PCIO-2 Address Map” page 6-9.

Real-Time Clock/NVRAM

The CPSB-560 provides the M48T58Y with a real-time clock (RTC) and an NVRAM which stores the OpenBoot boot variables.

For the address range of the real-time clock and the NVRAM, see Table 28 “PCIO-2 Address Map” page 6-9. For information on the M48T58Y, see the device’s data sheet.

Serial Interfaces

The CPSB-560 provides four independent full-duplex serial I/O interfaces which are implemented via four Enhanced Serial Communication Controllers 16C554 by Texas Instruments.

Interface A is available on the front panel of the CPU board. Interface B is routed to the front panel of the RTB-505. Interfaces C and D are available on the RTB-505’s front panel and can be used for a SUN type keyboard and/or mouse.

For the address ranges of the serial controllers, see Table 28 “PCIO-2 Address Map” page 6-9. For further information on the M48T58Y, see the device’s data sheet.

Xilinx FPGA

The Xilinx FPGA supports the following functions:

- Watchdog
- Timer
- Local I^2C bus

Watchdog

The watchdog timer is used to reset the board after a configurable time, if no software trigger occurred.

The watchdog can be enabled by setting SW3-1 to ON. A reset generation is indicated by the Reset Status register.

If enabled in the Interrupt Enable Control register, an interrupt will be generated before the watchdog timer runs out. The watchdog starts with the first trigger of the watchdog trigger bit in the Watchdog Trigger register. After the watchdog was started, it is not possible to stop the watchdog.

The watchdog timer can be configured to reset the board after 125 ms..1 hour of the last trigger in 15 steps. The value of each following step is increased by a factor of between 1.5 and 3. To be compatible to the CPCI-550 a fix reset value of 2.5 seconds is set. After the watchdog timer is running, it is only possible to reduce the watchdog run out time.

Timer

The timer can be used as two independent 16-bit count-down timers with a timer interval of 10 µs and a maximum run-out time of 655.35 ms. Two independent interrupts are possible which can be enabled or disabled with the Interrupt Enable Control register. One counter read-back register set is also available which shows the correct timer values.

The timer can also be run as one 32-bit count-down timer with a timer interval of 10 µs and a total run-out time of 42949.67295 s (or 11 h, 55 min, 49 s and 672.95 ms). In this mode only one interrupt is possible.

The timer counts down from its initial value to zero in steps of 10 µs. The initial value can be set by software from 1 to 65535 in the 16-bit mode or from 1 to 4294967295 in the 32-bit mode, which results in a timer period of 10 µs to 655.35 ms in the 16-bit mode or of 10 µs to 42949.67295s in the 32-bit mode. If the timer has reached zero, an interrupt is generated, if enabled, and the timer loads its initial value to count down again.

In total the timer has eleven registers:

- The first register is used to control the timer mode.
- One register is used to clear timer overruns.
- One register is used to read the timer overrun status.
- Four registers are used to set the initial timer values.
- The last four registers are used to read the current value of the count down registers.

Local I^2C Bus

To the local I^2C bus, the on-board and memory module's Serial Presence Detect (SPDs) are connected. They are I^2C bus slaves and are identified by unique addresses which are given in the table below.

Table 21: *Slave Addresses of Local I^2C Bus*

I^2C Bus Slave Address	Description
$101000x_2$	SPD CPSB-560 PROM bank 1-4 XICOR X24C04 Serial E^2PROM
$101001x_2$	SPD MEM-550 PROM bank 5-8 XICOR X24C04 Serial E^2PROM

IPMI Controller

The IPMI controller unit consists of the micro controller Vitesse VSC215, 512 KByte SRAM and 4 MByte flash memory and can be accessed via the Ebus. Furthermore, four I^2C buses are connected to the IPMI controller with the following buses / devices:

- Public buses:
  - I^2C bus 0: Connected to IPMB0
  - I^2C bus 1: Connected to IPMB1
- Private buses:
  - I^2C bus 2: Connected to the board information block (BIB) of the SPARC / CPSB-560 and of the attached boards / modules.
  - I^2C bus 3: Connected to on-board and module sensors

ICMB signals are available via J5 and RTB-505, IPMB0 signals via J1 and IPMB1 signals are available via J2, J5 and RTB-505.

IPMI offers the possibility to explicitly identify the product. This is done by means of the global IPMI command "Get Device ID". The product ID for SPARC / CPSB-560 is 0888_{16}.

I^2C Slave Addresses

The table below shows the I^2C bus slave addresses of devices attached to the I^2C buses 2 and 3.

Table 22: *I^2C Slave Addresses*

I^2C Bus	Device	Slave Address
Bus 2	BIB CPSB-560 XICOR X24C02 serial EEPROM	1010000_2
	BIB MEM-550 XICOR X24C02 serial EEPROM	1010100_2
	BIB RTB-505 XICOR 24C02 Serial EEPROM	1010.110_2
	BIB IPMI-Controller 24C02 Serial EEPROM	1010.111_2
Bus 3	Temperature sensor MAX1617	0011000_2

Available IPMI Drivers

Motorola offers two IPMI drivers to access the IPMI controller:

- Solaris Driver Package Rel. 2.17 (special price list item) providing an application programming interface (API).
  For information on the driver installation and the API, refer to the *Solaris Driver Packages Rel. 2.17 Installation and Reference Guide* delivered together with the driver package.
- On-board OpenBoot firmware IPMI driver
  It provides commands to access the IPMI controller. If you have written your IPMI application software and errors occur you can enter these commands at the OpenBoot prompt. If the IPMI commands are successfully executed from the prompt, it can be deducted that the problem is not hardware-related.

Temperature Sensor

The MAX1617 temperature sensor is connected to the IPMI controller's I^2C bus 3 and measures the temperature inside the CPU and on the board. For the location where the board temperature is measured, see figure below.

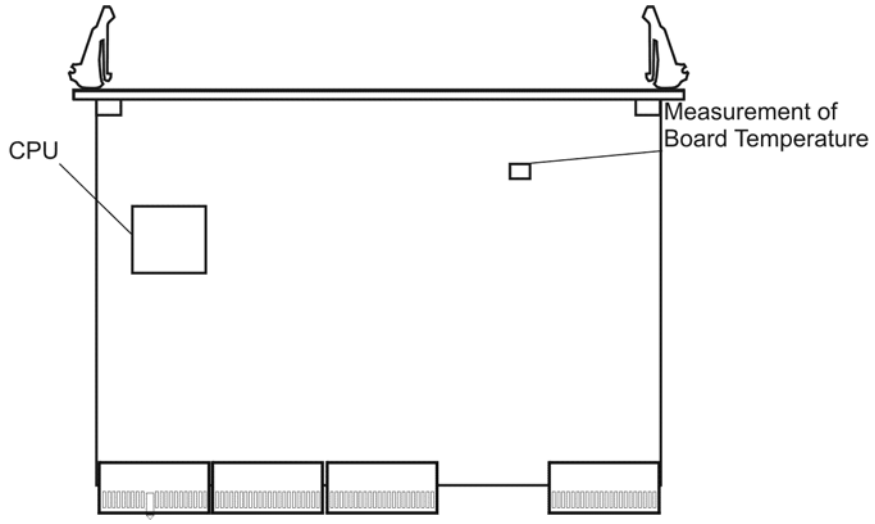


Figure 26: *Location of Board Temperature Measurement*

Note: The MAX1617 sensor is not powered by the IBMU. This means that if the power supply is interrupted, the sensor status at the time of power supply interruption is logged but the current sensor value cannot be read. The current sensor value can be read as soon as the board power is up again.

The temperature values can be read via the IPMI commands "Get Sensor Reading" and "Master Write-Read I^2C" with the Motorola Solaris driver Application Programming Interface (API). For the command "Master Write-Read I^2C" you need the temperature sensor's I^2C slave address (see Table 22 "I2C Slave Addresses" page 5-11).

The IPMI firmware provides a sensor data record (SDR) for the CPU and board temperature sensor containing configuration data such as threshold values. For the CPU temperature an upper critical threshold of 85°C is set. If the CPU temperature exceeds this value, the IPMI controller initiates an event which is written into the system event log (SEL) of the base board management controller (BMC). For the board temperature no threshold values are set and therefore, no events are generated. You can set the board temperature thresholds with the IPMI command "Set Sensor Threshold".

PCI Bus C

Only the PMC interface is connected to the PCI bus C. The PMC slot is 32-bit wide and runs at 33 MHz. The signaling level is 3.3V so the voltage key is installed in the 3.3V position to prevent 3.3V-only PMC modules from being installed.

Note: The PMC slot also supports processor PMC modules, but only in non-monarch mode.

6

Maps and Registers

Interrupt Map

The UltraSPARC-IIi+ provides a 6-bit wide interrupt vector for 63 interrupt sources.

Interrupt Concept

The separate device UPA interrupt concentrator (UIC) provides the inputs for all necessary interrupts. The UIC monitors all interrupts using a round-robin-scheme with 33 MHz, converts them to a device-own vector and transmits this vector to the processor. The PCI interrupt engine (PIE) reflects every vector in one state bit. From the state bit a new vector is generated and transmitted to the processor's execution unit. If more than one interrupt state bit is active, the transmitting sequence of the new interrupt vector is priority controlled.

Interrupt Sources

Every interrupt routed to the UIC can be enabled or disabled separately in the interrupt source and in the processor.

The following table lists all interrupt sources, their vectors from the UIC to the PIE, their vectors from the PIE to the processor's execution unit and the respective priority.

Table 23: *Interrupt Sources*

Segment	Function/ Interface	Device	ID-SEL/AD	CPU Internal Vector	UIC Vector	CPU Internal Priority
PCI A	PCI-to-PCI bridge	Advanced PCI Bridge SME2411	1_{16}/12	-	-	-
PCI A	SCSI	LSI53C1000	2_{16}/13	20_{16}	20_{16}	-
PCI A	Ethernet 2	Intel 8254xEM	3_{16}/14	01_{16}	05_{16}	7
PCI A	Ethernet 3	Intel 8254xEM	4_{16}/15	02_{16}	15_{16}	5

Table 23: *Interrupt Sources (cont.)*

Segment	Function/ Interface	Device	ID-SEL/AD	CPU Internal Vector	UIC Vector	CPU Internal Priority
PCI B	Ethernet 1	PCIO-2 controller SUN SME2300	1_{16}/12	21_{16}	21_{16}	3
PCI B	USB			$1C_{16}$	$1E_{16}$	6
PCI B	Ethernet 4	Intel 82559ER	2_{16}/13	00_{16}	07_{16}	5
PCI B	IDE	Si646	3_{16}/14	14_{16}	$0E_{16}$	6
PCI C	PMC	PMC slot I/O board	1_{16}/17	04_{16}	$0F_{16}$	7
			2_{16}/18	05_{16}	$0D_{16}$	5
			-	06_{16}	$1D_{16}$	5
			-	07_{16}	$0A_{16}$	2
System	Sun keyboard or serial interface C	Quad serial controller	-	26_{16}	28_{16}	7
	Sun mouse or serial interface D	Quad serial controller	-	28_{16}	$2A_{16}$	4
	Serial interface A	Quad serial controller	-	29_{16}	$2B_{16}$	7
	Serial interface B	Quad serial controller	-	$2A_{16}$	$2C_{16}$	6
	Timer	FPGA	-	24_{16}	$1F_{16}$	7
	Watchdog	FPGA	-	$0C_{16}$	18_{16}	6
	Eject	Eject Handle	-	$0B_{16}$	12_{16}	1
	KCS0	IPMI controller	-	03_{16}	02_{16}	2
	KCS1	IPMI controller	-	10_{16}	06_{16}	6
	KCS2	IPMI controller	-	11_{16}	04_{16}	4

Physical Memory Map

The UltraSPARC-IIi+ has a 41-bit wide physical address range. This address range is divided into some specified areas for e.g. the main memory or the PCI bus.

Each area is subdivided into other areas, e.g. the main memory area is subdivided into the different memory module areas with the memory banks. Some areas are subdivided further down to one register with one byte, i.e. the System Control registers in the EBus area are byte-oriented.

The tables on the following pages describe the areas with the related address maps. If an address map is subdivided into other areas, a separate table is available below and a reference to this table can be found in the description column.

UltraSPARC-IIi+ Physical Address Memory Map

The main address map gives an overview of the whole address space of the UltraSPARC-IIi+ CPU. This address range is used for the main memory, and the PCI bus. Each defined address space is divided into subspaces which are described in the next sections.

Table 24: *UltraSPARC-IIi+ Main Address Map*

Physical Address Range PA<40..0>	Size	Description	Access
$000.0000.0000_{16}$ - $000.7FFF.FFFF_{16}$	2 GByte	Main memory (see Table 25 “Main Memory Address Map” page 6-6)	Cacheable
$000.8000.0000_{16}$ - $007.FFFF.FFFF_{16}$		Reserved	Cacheable
$008.0000.0000_{16}$ - $1FB.FFFF.FFFF_{16}$		Reserved	Noncacheable
$1FC.0000.0000_{16}$ - $1FD.FFFF.FFFF_{16}$	8 GByte	Reserved, do not use	Noncacheable
$1FE.0000.0000_{16}$ - $1FF.FFFF.FFFF_{16}$	8 GByte	PCI bus, processor subsystem, memory, clock control, and ECU (see Table 26 “UltraSPARC-IIi+ Internal CSR Space” page 6-7 and see Table 27 “PCI Bus Address Map” page 6-8)	Noncacheable

Memory Address Map

The main memory address range is divided between on-board memory and the MEM-550 memory modules. Two banks of 512 MByte and four banks of 256 MByte result in a total amount of up to 2 GB main memory.

Table 25: *Main Memory Address Map*

Physical Address Range PA<41..0>	Size	Bank	Memory location	DIMM Type
$000.0000.0000_{16}$ - $000.1FFF.FFFF_{16}$	512 MByte	0	On-board memory	DIMM 0
	Not used	1		
$000.8000.0000_{16}$ - $000.9FFF.FFFF_{16}$	512 MByte	2	On-board memory	DIMM 1
	Not used	3		
$001.0000.0000_{16}$ - $001.0FFF.FFFF_{16}$	256 MByte	4	SPARC / MEM-550	DIMM 2
$001.1000.0000_{16}$ - $001.1FFF.FFFF_{16}$	256 MByte	5		
$000.8000.0000_{16}$ - $000.8FFF.FFFF_{16}$	256 MByte	6		DIMM 3
$000.9000.0000_{16}$ - $000.9FFF.FFFF_{16}$	256 MByte	7		

UltraSPARC-IIi+ Internal CSR Space

The UltraSPARC-IIi+ internal configuration space registers (CSR) are used for the configuration of the peripheral parts of the CPU, e.g. the PCI bus module, the I/O memory management unit and the interrupt unit.

Table 26: *UltraSPARC-IIi+ Internal CSR Space*

Physical Address Range	Size	Description
$1FE.0000.0000_{16}$ - $1FE.0000.01FF_{16}$	512 Byte	PCI bus module (PBM)
$1FE.0000.0200_{16}$ - $1FE.0000.03FF_{16}$	512 Byte	I/O memory management unit (IOM)
$1FE.0000.0400_{16}$ - $1FE.0000.1FFF_{16}$	7 KByte	PCI interrupt engine (PIE)
$1FE.0000.2000_{16}$ - $1FE.0000.5FFF_{16}$	16 KByte	PBM
$1FE.0000.6000_{16}$ - $1FE.0000.9FFF_{16}$	12 KByte	PIE
$1FE.0000.A000_{16}$ - $1FE.0000.A7FF_{16}$	2 KByte	IOM
$1FE.0000.A800_{16}$ - $1FE.0000.EFFF_{16}$	22 KByte	PIE
$1FE.0000.F000_{16}$ - $1FE.00FF.F018_{16}$	23 MByte	Memory control unit (MCU)
$1FE.00FF.F020_{16}$	8 Byte	PIE
$1FE.00FF.F028_{16}$ - $1FE.00FF.FFFF_{16}$	4 KByte	MCU
$1FE.0100.0000_{16}$ - $1FE.0100.0041_{16}$		PBM

PCI Bus Address Map

The PCI bus address space is divided into areas for the different PCI accesses, e.g. configuration access, I/O access or memory access. These areas are distributed to the PCI devices on the SPARC/CPSB-560.

The address allocation of the devices is made dynamically during the PCI configuration cycles after reset in OpenBoot. The allocation depends on the availibility of PCI devices (I/O board, PMC module).

The PCI device PCIO, part of the UltraSPARC-IIi+ chip set, must be available at power up for booting and has a fixed PCI address space. It has an interface to the EBus, where the boot PROM is located. Additionally, it has an interface to the MII bus from where the twisted-pair Ethernet interfaces are generated.

Table 27: *PCI Bus Address Map*

Address Range in PA<40:0>	Size	Description
$1FE.0100.0100_{16}$- $1FE.01FF.FFFF_{16}$	24 MByte - 256 Byte	PCI bus configuration space
$1FE.0200.0000_{16}$- $1FE.02FF.FFFF_{16}$	24 MByte	PCI bus I/O space
$1FE.0300.0000_{16}$- $1FE.FFFF.FFFF_{16}$	4 GByte - 48 MByte	Reserved
$1FF.0000.0000_{16}$- $1FF.FFFF.FFFF_{16}$	4 GByte	PCI bus memory space
$1FF.F000.0000_{16}$- $1FF.F17F.FFFF_{16}$	24 MByte	PCI bus memory space for the PCIO-2 (see Table 28 “PCIO-2 Address Map” page 6-9)
$1FF.F180.0000_{16}$- $1FF.FFFF.FFFF_{16}$	256 MByte - 24 MByte	PCI bus memory space

PCIO-2 Address Map

The PCIO-2 has an address space of 24 MByte in total. It is divided into:

- 16 MByte for the boot PROM or flash memory on the EBus (CS#0)
- Seven address spaces for other EBus devices (CS#1 - CS#7), e.g. RTC / NVRAM, the SPARC / CPSB-560 System Configuration registers or a serial controller.
- The PCIO-2 System Configuration registers

Table 28: *PCIO-2 Address Map*

Address Range in PA<40:0>	Size	EBus CS#	Description
$1FF.F000.0000_{16}$ - $1FF.F00F.FFFF_{16}$	1 MByte	0	Boot PROM on the EBus (if SW2-1 is OFF and if bit 0 of the Miscellaneous Control register is set to 0)
$1FF.F010.0000_{16}$ - $1FF.F0FF.FFFF_{16}$	15 MByte	0	Reserved for the EBus (if SW2-1 is OFF and if bit 0 of the Miscellaneous Control register is set to 0)
$1FF.F000.0000_{16}$ - $1FF.F00F.FFFF_{16}$	1 MByte	0	Boot section of flash EPROM on the EBus (if SW2-1 is ON or if bit 0 of the Miscellaneous Control register is set to 1)
$1FF.F010.0000_{16}$ - $1FF.F0FF.FFFF_{16}$	15 MByte	0	User flash memory on the EBus (if SW2-1 is ON or if bit 0 of the Miscellaneous Control register is set to 1)
$1FF.F100.0000_{16}$ - $1FF.F10F.FFFF_{16}$	1 MByte	1	RTC / NVRAM on the EBus
$1FF.F110.0000_{16}$ - $1FF.F11F.FFFF_{16}$	1 MByte	2	Boot PROM mirror area (independent of SW2-1 and bit 0 of the Miscellaneous Control register)
$1FF.F120.0000_{16}$ - $1FF.F12F.FFFF_{16}$	1 MByte	3	Reserved for the EBus
$1FF.F130.0100_{16}$ - $1FF.F130.0108_{16}$	8 Byte	4	Serial controller on the EBus Serial interface A
$1FF.F130.0200_{16}$ - $1FF.F130.0208_{16}$	8 Byte	4	Serial controller on the EBus Serial interface B
$1FF.F130.0300_{16}$ - $1FF.F130.0308_{16}$	8 Byte	4	Serial controller on the EBus Serial interface C
$1FF.F130.0400_{16}$ - $1FF.F130.0408_{16}$	8 Byte	4	Serial controller on the EBus Serial interface D

Table 28: *PCIO-2 Address Map (cont.)*

Address Range in PA<40:0>	Size	EBus CS#	Description
$1FF.F130.1000_{16}$ - $1FF.F130.1004_{16}$	4 Byte	4	KCS0
$1FF.F130.1100_{16}$ - $1FF.F130.1104_{16}$	4 Byte	4	KCS1
$1FF.F130.1200_{16}$ - $1FF.F130.1204_{16}$	4 Byte	4	KCS2
$1FF.F140.0000_{16}$ - $1FF.F14F.FFFF_{16}$	1 MByte	5	Reserved for the EBus
$1FF.F150.0000_{16}$ - $1FF.F15F.FFFF_{16}$	1 Myte	6	Reserved for the EBus
$1FF.F150.0010_{16}$ - $1FF.F15F.FFFF_{16}$	1 MByte - 16 Byte	6	Reserved for the EBus
$1FF.F160.0100_{16}$ - $1FF.F160.01FF_{16}$	256 Byte	7	System Configuration register on the EBus
$1FF.F160.0200_{16}$ - $1FF.F16F.FFFF_{16}$	1 MByte - 256Byte	7	Reserved for the EBus
$1FF.F170.0000_{16}$ - $1FF.F17F.FFFF_{16}$	1 MByte	n.a.	PCIO configuration registers

System Configuration Registers

The SPARC/CPSB-560 implements a set of system configuration registers via the field-programmable gate array (FPGA) Spartan XCS20XL, which is accessible via the EBus.

The SPARC/CPSB-560 System Configuration registers are used to control the on-board functions and receive status information of the board.

Overview of System Configuration Registers

The table below shows all SPARC/CPSB-560 registers in alphabetical order. The second table shows these registers sorted by address range. Every register is described separately in the following chapters.

Table 29: *Alphabetical List of System Configuration Registers*

Register	Description
Board Configuration Status register	Page 6-32
External Failure Status register	Page 6-19
FPGA Revision Status register	Page 6-32
I^2C register	Page 6-33
Interrupt Enable Control register	Page 6-26
Interrupt Pending Status register	Page 6-27
LED 0 Control register	Page 6-15
LED 1 Control register	Page 6-16
LED 2 Control register	Page 6-17
LED 3 Control register	Page 6-18
Miscellaneous Control register	Page 6-14
Reset Clear Control register	Page 6-29
Reset Control register	Page 6-29
Reset Status register	Page 6-30
Switch 1 and 2 Status register	Page 6-31
Switch 3 Status register	Page 6-31
Timer Clear Control register	Page 6-23

Table 29: *Alphabetical List of System Configuration Registers (cont.)*

Register	Description
Timer Control register	Page 6-22
Timer Counter Status register	Page 6-25
Timer Initial Control register	Page 6-25
Timer Status register	Page 6-24
Watchdog Timer Control register	Page 6-20
Watchdog Timer Status register	Page 6-21
Watchdog Timer Trigger register	Page 6-21

Table 30: *System Configuration Registers Sorted by Address Range*

Address Range in PA<40:0>	Description	Reset Value	Page
$1FF.F160.0100_{16}$	**Function Unit Miscellaneous Control**		
$1FF.F160.0100_{16}$	Miscellaneous Control register	00_{16}	6-14
$1FF.F160.0110_{16}$	**Function Unit Display**		
$1FF.F160.0110_{16}$	LED 0 Control register	00_{16}	6-15
$1FF.F160.0111_{16}$	LED 1 Control register	00_{16}	6-16
$1FF.F160.0112_{16}$	LED 2 Control register	00_{16}	6-17
$1FF.F160.0113_{16}$	LED 3 Control register	00_{16}	6-18
$1FF.F160.0120_{16}$	**Function Unit External Failure**		
$1FF.F160.0120_{16}$	External Failure Status register	00_{16}	6-19
$1FF.F160.0130_{16}$	**Function Unit Watchdog**		
$1FF.F160.0130_{16}$	Watchdog Timer Control register	08_{16}	6-20
$1FF.F160.0131_{16}$	Watchdog Timer Trigger register	FF_{16}	6-21
$1FF.F160.0134_{16}$	Watchdog Timer Status register	00_{16}	6-21
$1FF.F160.0140_{16}$	**Function Unit Timer**		
$1FF.F160.0140_{16}$	Timer Control register	00_{16}	6-22
$1FF.F160.0141_{16}$	Timer Clear Control register	FF_{16}	6-23
$1FF.F160.0144_{16}$	Timer Status register	00_{16}	6-24

Table 30: *System Configuration Registers Sorted by Address Range (cont.)*

Address Range in PA<40:0>	Description	Reset Value	Page
$1FF.F160.0148_{16}$	Timer 1 Initial Control register U	00_{16}	6-25
$1FF.F160.0149_{16}$	Timer 1 Initial Control register L	00_{16}	6-25
$1FF.F160.014A_{16}$	Timer 2 Initial Control register U	00_{16}	6-25
$1FF.F160.014B_{16}$	Timer 2 Initial Control register L	00_{16}	6-25
$1FF.F160.014C_{16}$	Timer 1 Counter Status register U	00_{16}	6-25
$1FF.F160.014D_{16}$	Timer 1 Counter Status register L	00_{16}	6-25
$1FF.F160.014E_{16}$	Timer 2 Counter Status register U	00_{16}	6-25
$1FF.F160.014F_{16}$	Timer 2 Counter Status register L	00_{16}	6-25
$1FF.F160.0150_{16}$- $1FF.F160.0170_{16}$	Reserved		-
$1FF.F160.0180_{16}$	**Function Unit Interrupts**		
$1FF.F160.0180_{16}$	Interrupt Enable Control register	00_{16}	6-26
$1FF.F160.0184_{16}$	Interrupt Pending Status register	00_{16}	6-27
$1FF.F160.0190_{16}$- $1FF.F160.01D0_{16}$	Reserved		-
$1FF.F160.01D0_{16}$	**Function Unit Reset**		
$1FF.F160.01D0_{16}$	Reset Control register	22_{16}	6-29
$1FF.F160.01D1_{16}$	Reset Clear Control register	FF_{16}	6-29
$1FF.F160.01D4_{16}$	Reset Status register	00_{16}	6-30
$1FF.F160.01E0_{16}$	**Function Unit Board Status**		
$1FF.F160.01E0_{16}$	Switch 1 and 2 Status register	$E3_{16}$	6-31
$1FF.F160.01E2_{16}$	Switch 3 Status register	$0F_{16}$	6-31
$1FF.F160.01E4_{16}$	Board Configuration Status register	00_{16}	6-32
$1FF.F160.01EE_{16}$	FPGA Revision Status register	00_{16}	6-32
$1FF.F160.01FF_{16}$	**Function Unit I^2C**		
$1FF.F160.01FF_{16}$	I^2C register	00_{16}	6-33

Overview of IPMI Related Register Bits

The table below gives an overview of IPMI related register bits. These bits can be used if you wish to use interrupts (e.g. OBF interrupts) in your own IPMI driver.

Table 31: *Overview of IPMI Related Register Bits*

Function	Signal	Description
To enable interrupts from IPMI controller	IE_IPMI	Page 6-27
To indicate whether an interrupt from the IPMI controller is pending (only if IE_IPMI is set)	IP_IPMI	Page 6-28

Miscellaneous Control Register

The Miscellaneous Control register is used to define whether the boot PROM or the flash EPROM is available in the CS#0 address space.

Table 32: *Miscellaneous Control Register*

Base Address: 1FF.F160.0100$_{16}$
Offset: 0$_{16}$

Bit	Signal	Description	Access
7..6	Reserved	Reserved	r / w
5	HSLED_ON	Used to switch hot swap LED on 0: (default) Hot swap LED is OFF 1: Hot swap LED is shining blue	r / w
4	RTB_GPO	Used to set the GPO on the RTB-505 0: (default) GPO on RTB-505 is 0 1: GPO on RTB-505 is 1	r / w
3..1	Reserved	Reserved	r / w
0	TSOP EN	Used to switch between boot PROM access and flash EPROM access in the address space for CS#0. After reset this bit is cleared (0) 0 (default): If SW2-1 is OFF, the boot PROM is available in the CS#0 address space If SW2-1 is ON the flash EPROM is available in the CS#0 address space 1: The flash EPROM is available in the CS#0 address space	r / w

Display Registers

The display registers are used to control the four user LEDs.

LED 0 Control Register

The user LED 0 can be configured as Ethernet interface 1/5 LED or as user LED.

Note: Independent of your configuration user LED 0 flashes red if SPARC/CPSB-560 does not run properly (i.e. the last assertion of PCI_FRAME has been more than 1.5 s ago).

Table 33: *LED 0 Control Register*

Base Address: $1FF.F160.0100_{16}$
Offset: 10_{16}

Bit	Signal	Description	Access
7..5	Reserved	Reserved	r/w
4	ETH 1/5	This bit controls whether LED 0 acts as Ethernet 1/5 LED or as user LED configured with bits 3..0. After reset, this bit is cleared (0). 0 (default): User LED 0 configured as user LED 1: User LED 0 acts as Ethernet LED for ETH1/ETH5. It will shine green when link runs at 100/10 MBit. The LED will flash when it is active.	r/w
3..2	BLINK_FREQ	The two bits control the frequency at which the user LED 0 is flashing. The settings below are only effective if bit 4 is set to zero. 00_2 (default): User LED is permanently on. 01_2: User LED 0 flashes at 0.5 Hz (slow). 10_2: User LED 0 flashes at 1 Hz (moderate). 11_2: User LED 0 flashes at 2 Hz (fast).	r/w
1..0	COLOR	The two bits are used to turn the user LED 0 on or off and to set the color. After reset these bits are cleared (00_2). The settings below are only effective if bit 4 is set to zero. 00_2 (default): User LED 0 is off. 01_2: User LED 0 is on (green). 10_2: User LED 0 is on (red). 11_2: Reserved	r/w

LED 1 Control Register

The user LED 1 can be configured as Ethernet interface 2 LED or as user LED.

Table 34: *LED 1 Control Register*

Base Address: 1FF.F160.0100$_{16}$
Offset: 11$_{16}$

Bit	Signal	Description	Access
7..5	Reserved	Reserved	r / w
4	ETH 2	This bit controls whether LED 1 acts as Ethernet 2 LED or as user LED (default) configured with bits 3..0. After reset, this bit is cleared (0). 0 (default): LED 1 is configured as user LED. 1: LED 1 acts as Ethernet LED for ETH 2. It shines green when the link runs at 10 / 100 / 1000 MBit. The LED flashes when the link is active.	r / w
3..2	BLINK_FREQ	The two bits control the frequency at which the user LED 1 is flashing. The settings below are only effective if bit 4 is set to zero. 00_2 (default): User LED 1 is permanently on. 01_2: User LED 1 flashes at 0.5 Hz (slow). 10_2: User LED 1 flashes at 1 Hz (moderate). 11_2: User LED 1 flashes at 2 Hz (fast).	r / w
1..0	COLOR	The two bits are used to turn the user LED on or off and to control the color. After reset these bits are cleared (00_2). The settings below are only effective if bit 4 is set to zero. 00_2 (default): User LED 1 is off. 01_2: User LED 1 is on (green). 10_2: User LED 1 is on (red). 11_2: Reserved	r / w

LED 2 Control Register

The user LED 2 can be configured as Ethernet interface 3 LED or as user LED.

Table 35: *LED 2 Control Register*

Base Address: $1FF.F160.0100_{16}$
Offset: 10_{16}

Bit	Signal	Description	Access
7..5	Reserved	Reserved	r / w
4	ETH 4	This bit controls whether LED 2 acts as Ethernet 3 LED or as user LED (default) configured with bits 3..0. After reset, this bit is cleared (0). 0 (default): LED 2 is configured as user LED. 1: LED 2 acts as Ethernet LED for Ethernet interface 3. It shines green when link runs at 10 / 100 / 1000 MBit. The LED flashes when the link is active.	r / w
3..2	BLINK_FREQ	The two bits control the frequency at which the user LED is flashing. The settings below are only effective if bit 4 is set to zero. 00_2 (default): User LED 2 is permanently on. 01_2: User LED 2 flashes at 0.5 Hz (slow). 10_2: User LED 2 flashes at 1 Hz (moderate). 11_2: User LED 2 flashes at 2 Hz (fast).	r / w
1..0	COLOR	The two bits are used to turn the user LED on or off and to set the color. After reset these bits are cleared (00_2). The settings below are only effective if bit 4 is set to zero. 00_2 (default): User LED2 is off. 01_2: User LED 2 is on (green). 10_2: User LED 2 is on (red). 11_2: Reserved	r / w

LED 3 Control Register

The user LED 3 can be configured as Ethernet interface 4 LED or as user LED.

Table 36: *LED 3 Control Register*

Base Address: $1FF.F160.0100_{16}$
Offset: 13_{16}

Bit	Signal	Description	Access
7..5	Reserved	Reserved	r / w
4	ETH 3	This bit controls whether LED 3 acts as Ethernet 4 LED or as user LED (default) configured with bits 3..0. After reset, this bit is cleared (0). 0 (default): LED 3 is configured as user LED. 1: LED 3 acts as Ethernet LED for ETH 4. It shines green when link runs at 100 / 10 MBit. The LED flashes when the link is active.	r / w
3..2	BLINK_FREQ	The two bits control the frequency at which the user LED is flashing. The settings below are only effective if bit 4 is set to zero. 00_2 (default): User LED is permanently on. 01_2: User LED 3 flashes at 0.5 Hz (slow). 10_2: User LED 3 flashes at 1 Hz (moderate). 11_2: User LED 3 flashes at 2 Hz (fast).	r / w
1..0	COLOR	The two bits are used to turn the user LED on or off and to control the color. After reset these bits are cleared (00_2). The settings below are only effective if bit 4 is set to zero. 00_2 (default): User LED 3 is off. 01_2: User LED 3 is on (green). 10_2: User LED 3 is on (red). 11_2: Reserved	r / w

External Failure Status Register

The External Failure Status register is used to receive information of external failure conditions, for example, overheating or power supply problems. All failure conditions can also be configured as an interrupt (see “Interrupt Registers” page 6-26).

Table 37: *External Failure Register*

Base Address: 1FF.F160.0100$_{16}$
Offset: 20$_{16}$

Bit	Signal	Description	Access
7	STAT FAL	This bit reflects the state of the CompactPCI low active FAL signal (J2.C15). This bit shows a failure of the power supply. 0 (default): The FAL signal is inactive (high). 1: The FAL signal is active (low).	r
6	STAT DEG	This bit reflects the state of the CompactPCI low active DEG signal (J2.C16). This bit shows a failure of the power supply. 0 (default): The DEG signal is inactive (high). 1: The DEG signal is active (low).	r
5..4	Reserved	Reserved	r
3	EJECT	This bit reflects whether the eject handle is opne 0(default): Eject Handle is closed 1: Eject handle is open	r
2..0	Reserved	Reserved	r

Watchdog Timer Registers

The watchdog timer is used to reset the board after a defined time interval, if no software trigger occurred. Before the watchdog timer runs out, an interrupt will be generated if it is enabled in the Interrupt Enable Control register.

The watchdog starts with the first trigger of the watchdog trigger bit in the Watchdog Trigger register. After the watchdog is enabled, it is not possible to stop the watchdog.

The watchdog timer can be configured to reset the board after a certain time interval which can vary between 125 ms and 1 hour. After reset, the time is set to 2.5 s for the reset and to 1.25 s for the interrupt, which is compatible to the SPARC / CPCI-550.

Watchdog Timer Control Register

The Watchdog Timer Control register is used to set the time out for the watchdog timer.

Table 38: *Watchdog Timer Control Register*

Base Address: 1FF.F160.0100$_{16}$
Offset: 30$_{16}$

Bit	Signal	Description	Access
7..5	Reserved	Reserved After reset, these bits are set to (000_2).	r / w
4..0	WDOG LENGTH	These bits are used to set the time out for the watchdog timer. **If the watchdog is running, you can only change the watchdog time to a smaller value.** The tolerance of the time delay is 100ppm or +10 ms / -10 ms whichever is greater. After reset the bits are set to 08_{16}.	r / w

	Reset after:	Interrupt after:
00_{16}	125 ms	62 ms
02_{16}	250 ms	125 ms
04_{16}	500 ms	250 ms
06_{16}	1 s	500 ms
08_{16}	2.5 s	1.25 s
$0A_{16}$	5 s	3 s
$0C_{16}$	10 s	8 s
$0E_{16}$	30 s	25 s
10_{16}	1 min	50 s
12_{16}	3 min	2 min
14_{16}	5 min	4 min
16_{16}	10 min	8 min
18_{16}	20 min	18 min
$1A_{16}$	30 min	25 min
$1C_{16}$	60 min	50 min
$1F_{16}$	Watchdog timer off	

Watchdog Timer Trigger Register

The Watchdog Timer Trigger register is used to trigger the watchdog timer.

Table 39: *Watchdog Timer Trigger Register*

Base Address: $1FF.F160.0100_{16}$
Offset: 31_{16}

Bit	Signal	Description	Access
7..4	Reserved	Reserved	w
3	WDOG TRIG	This bit is used to trigger the watchdog timer. If the watchdog is enabled through switch SW3-1 your software must set this bit within the time period configured in the Watchdog Control register. If a watchdog interrupt is pending it will be cleared by triggering the watchdog. However, the Interrupt Pending Status register must be cleared separately. 0: Watchdog timer is not triggered. 1: Watchdog timer is triggered.	w
2..0	Reserved	Reserved	w

Watchdog Timer Status Register

The Watchdog Timer Status register reflects the watchdog timer status.

Table 40: *Watchdog Timer Status Register*

Base Address: $1FF.F160.0100_{16}$
Offset: 34_{16}

Bit	Signal	Description	Access
7..1	Reserved	Reserved	r
0	STAT WDOG	This bit reflects the status of the watchdog timer. 0: The watchdog timer has not reached the interrupt time. 1: The watchdog timer has exceeded the interrupt time. It is necessary to trigger the watchdog timer.	r

Timer Registers

The timer can be used as two independent 16-bit countdown timers with a timer interval of 10 µs and a total maximum run-out time of 655.35 ms. Two independent interrupts are possible, which can be enabled or disabled (see “Interrupt Registers” page 6-26). A counter read-back register set is also available which always shows the correct timer value.

The timer can also be used as one 32-bit countdown timer with a timer interval of 10 µs and a total run-out time of 42949.67295 s or 11h, 55min, 49 s and 67.295 ms. In this mode only one interrupt is available and possible.

The timer counts down from its initial value to zero in intervals of 10 µs. The initial value can be set by software from 1 to 65535 in the 16-bit mode or to 4294967296 in the 32-bit mode, which results in a timer period of 10 µs to 655.35 ms in 16-bit mode or 42949.67295 s in 32-bit mode. If the timer has reached zero, an interrupt is generated, if enabled, and the timer loads his initial value to count down again.

The timer has eleven registers in total. The first register is used to control the timer mode, one register is used to clear timer overruns, one register is used to read the timer overrun status, four registers are used for setting the initial timer values and the last four registers are used to read the current value of the countdown timers.

Timer Control Register

This register is used to set up the timer. If the timer is set to zero, the timer is off and no interrupts are generated. However, the Timer Status register will not be cleared. In the first countdown after the timer activation the tolerance of the timer is 100 ppm plus a maximum of 10 µs.

Table 41: *Timer Control Register*

Base Address: $1FF.F160.0100_{16}$
Offset: 40_{16}

Bit	Signal	Description	Access
7..5	Reserved	Reserved After reset the bits are set to 00_2.	r / w
4	EN TIM2	This bit is used to control timer 2. After reset, this bit is set to 0. 0 (default): Timer 2 is disabled. 1: Timer 2 is enabled.	r / w
3..2	Reserved	Reserved After reset, the bits are set to 00_2.	r / w

Table 41: *Timer Control Register (cont.)*

Base Address: $1FF.F160.0100_{16}$ Offset: 40_{16}			
Bit	**Signal**	**Description**	**Access**
1	EN MOD32	This bit is used to switch between two 16-bit-wide timers and one 32-bit-wide timer. After reset this bit is set to 0. 0 (default): 32-bit mode is disabled. 1: 32-bit mode is enabled.	r / w
0	EN TIM1	This bit is used to control timer 1. After reset this bit is set to 0. 0 (default): Timer 1 is disabled. 1: Timer 1 is enabled.	r / w

Timer Clear Control Register

This register is used to control the status bits of both timers in the Timer Status register.

Table 42: *Timer Clear Control Register*

Base Address: $1FF.F160.0100_{16}$ Offset: 41_{16}			
Bit	**Signal**	**Description**	**Access**
7..5	Reserved	Reserved	w
4	CLR TIM2	This bit is used to clear the status bits of timer 2 in the Timer Status register. 0: Timer 2 status bits will not be cleared. 1: Timer 2 status bits will be cleared.	w
3..1	Reserved	Reserved	w
0	CLR TIM1	This bit is used to clear the status bits of timer 1 in the Timer Status register. 0: Timer 1 status bits will not be cleared. 1: Timer 1 status bits will be cleared.	w

Timer Status Register

The Timer Status register is used to recognize timer underrun conditions.

Table 43: *Timer Status Register*

Base Address: 1FF.F160.0100$_{16}$
Offset: 44$_{16}$

Bit	Signal	Description	Access
6..7	Reserved	Reserved	r
5	ERR TIM2	This bit is used to recognize more than one time underrun without clearance. It is a status for a missed timer underrun. This can only occur, if timer 2 is enabled, the initial value is higher than 0 and if the 32-bit mode is disabled. 0 (default): Only one time underrun of timer 2 has occurred. 1: More than one time underrun of timer 2 has occurred.	r
4	STAT TIM2	This bit is used to recognize an underrun of timer 2. This can only occur if timer 2 is enabled, the initial value is higher than 0 and if the 32-bit mode is disabled. 0 (default): No underrun of timer 2 has occurred. 1: An underrun of timer 2 has occurred.	r
3..2	Reserved	Reserved	r
1	ERR TIM1	This bit is used to recognize more than one time underrun without clearance. It is a status for a missed timer underrun. This can only occur if timer 1 is enabled and the initial value is higher than 0. 0 (default): Only one time underrun of timer 1 has occurred. 1: More than one time underrun of timer 1 has occurred.	r
0	STAT TIM1	This bit is used to recognize an underrun of timer 1. This can only occur if timer 1 is enabled and the initial value is higher than 0. 0 (default): No underrun of timer 1 has occurred. 1: An underrun of timer 1 has occurred.	r

Timer Initial Control Registers

The following four registers are used to set up the run-out time of both timers. The 32 bits are distributed as big endian, which means the first register ($1FF.F160.0148_{16}$) represents the bits 31..24 and so on.

Table 44: *Timer Initial Control Registers*

Base Address: $1FF.F160.0100_{16}$
Offset: 48_{16} - $4B_{16}$

Bit	Signal	Description	Access
		16-Bit-Mode:	
31..16	TIMER1 INIT	Initialization time of timer 1 After reset, this bit is set to 0000_{16} 0000_{16}: Timer 1 is disabled 0001_{16}: Timer 1 run-out time is 10 µs $FFFF_{16}$: Timer 1 run-out time is 655.35 ms	r / w
15..0	TIMER2 INIT	Initialization time of timer 2 After reset, this bit is set to 0000_{16} 0000_{16}: Timer 2 is disabled. 0001_{16}: Timer 2 run-out time is 10 µs $FFFF_{16}$: Timer 2 run-out time is 655.35 ms	r / w
		32-Bit-Mode:	
31..0	TIMER1 INIT	Initialization time of timer 1 After reset, this bit is set to 0000_{16} 0000.0000_{16}: Timer 1 disabled 0000.0001_{16}: Timer 1 run-out time is 10 µs $FFFF.FFFF_{16}$: Timer 1 run-out time is 42949.67295 s	r / w

Timer Counter Status Register

The following four registers are used to read the current timer value of both timers. The 32 bits are also distributed as big endian.

Table 45: *Timer Counter Status Register*

Base Address: $1FF.F160.0100_{16}$
Offset: $4C_{16}$ - $4F_{16}$

Bit	Signal	Description	Access
		16-Bit Mode:	
31..16	TIMER1 VALUE	Current value of timer 1 0000_{16} (default): Timer 1 is not running. 0001_{16}: Timer 1 will initialize again during the next 10 µs. $7FFF_{16}$: Timer 1 needs 327.68 ms until next initialization. $FFFF_{16}$: Timer 1 needs 655.36 ms until next initialization.	r

Table 45: *Timer Counter Status Register (cont.)*

Base Address: 1FF.F160.0100$_{16}$
Offset: 4C$_{16}$ - 4F$_{16}$

Bit	Signal	Description	Access
15..0	TIMER2 VALUE	Current value of timer 2 0000_{16} (default): Timer 2 is not running. 0001_{16}: Timer 2 will initialize again during the next 10 µs. $7FFF_{16}$: Timer 2 needs 327.68 ms until next initialization. $FFFF_{16}$: Timer 2 needs 655.36 ms until next initialization.	r
		32-Bit Mode:	
31..0	TIMER1 VALUE	Current value of timer 1 0000.0000_{16} (default): Timer 1 is not running. 0000.0001_{16}: Timer 1 will initialize again during the next 10 µs. $0000.7FFF_{16}$: Timer 1 needs 327.68 ms until next initialization. $FFFF.FFFF_{16}$: Timer 1 needs 42949.67295 s until next initialization.	r

Interrupt Registers

The interrupt registers are used to distribute all possible failures or status information to the UPA interrupt concentrator (UIC). These registers are essential to enable the interrupts and read back the status of a pending interrupt. Interrupts must be cleared in the respective function. To clear the timer 1 interrupt, for example, set the CLR_TIM1 bit in the Timer Clear Control register.

Interrupt Enable Control Register

This register is used to enable or disable the interrupt sources.

Table 46: *Interrupt Enable Control Register*

Base Address: 1FF.F160.0100$_{16}$
Offset: 80$_{16}$

Bit	Signal	Description	Access
7..6	Reserved	Reserved After reset, the bits are set to 0.	r / w
5	IE_TIMER2	This bit is used to enable the timer 2 interrupt. After reset, this bit is set to 0. 0 (default): Timer 2 interrupt is disabled. 1: Timer 2 interrupt is enabled.	r / w

Table 46: *Interrupt Enable Control Register (cont.)*

Base Address: 1FF.F160.0100$_{16}$
Offset: 80$_{16}$

Bit	Signal	Description	Access
4	IE_TIMER1	This bit is used to enable the timer 1 interrupt. After reset, this bit is set to 0. 0 (default): Timer 1 interrupt is disabled. 1: Timer 1 interrupt is enabled.	r / w
3	IE_EJECT	This bit is used to enable the eject interrupt. After reset, this bit is set to 0. 0 (default): Eject interrupt is disabled. 1: Eject interrupt is enabled.	r / w
2	Reserved	Reserved After reset, the bits are set to 0.	r / w
1	IE_IPMI	This bit is used to enable interrupts from the IPMI controller. After reset, this bit is set to (0). 0 (default): Interrupts from IPMI controller are disabled. 1: Interrupts from IPMI controller are enabled.	r / w
0	IE_WDT	This bit is used to enable the watchdog timer interrupt. After reset, this bit is set to 0. 0 (default): Watchdog timer interrupt is disabled. 1: Watchdog timer interrupt is enabled.	r / w

Interrupt Pending Status Register

This register reflects whether a certain interrupt is pending.

Note: A write access to this register clears the register by setting all bits to 0.

Table 47: *Interrupt Pending Status Register*

Base Address: 1FF.F160.0100$_{16}$
Offset: 84$_{16}$

Bit	Signal	Description	Access
7..6	Reserved	Reserved	r
5	IP_TIMER2	This bit reflects whether a timer 2 interrupt is pending. 0 (default): No timer 2 interrupt is pending. 1: Timer 2 interrupt is pending.	r

Table 47: *Interrupt Pending Status Register (cont.)*

Base Address: $1FF.F160.0100_{16}$
Offset: 84_{16}

Bit	Signal	Description	Access
4	IP_TIMER1	This bit reflects whether a timer 1 interrupt is pending. 0 (default): No timer 1 interrupt is pending. 1: Timer 1 interrupt is pending.	r
3	IP_EJECT	This bit reflects whether an eject interrupt is pending. 0 (default): No eject interrupt is pending. 1: Eject interrupt is pending.	r
2	Reserved	Reserved	r
1	IP_IPMI	This bit reflects if an interrupt from the IPMI controller is pending and if the IE_IPMI bit in the Interrupt Enable Control register is enabled. 0 (default): No interrupt from the IPMI controller is pending. 1: An interrupt from the IPMI controller is pending.	r
0	IP_WDT	This bit reflects whether a watchdog timer interrupt is pending. 0 (default): No watchdog timer interrupt is pending. 1: Watchdog timer interrupt is pending.	r

Reset Registers

The reset registers are used to enable or disable reset sources and to identify the source of the last reset.

Reset Control Register

The Reset Control register masks the resets coming from CompactPCI connector J2.

Table 48: *Reset Control Register*

Base Address: $1FF.F160.0100_{16}$
Offset: $D0_{16}$

Bit	Signal	Description	Access
7..2	Reserved	Reserved	r / w
1	RSTEN J2	This bit is used to mask the push-button reset from the CompactPCI pin J2-C17 on the SPARC / CPSB-560. After reset, this bit is set to 0. 0 (default): Push-button reset from J2 is disabled. 1: Push-button reset from J2 is enabled.	r / w
0	Reserved	Reserved After reset, the bit is set to 0.	r / w

Reset Clear Control Registers

This register is used to clear the Reset Status register.

Table 49: *Reset Clear Control Register*

Base Address: $1FF.F160.0100_{16}$
Offset: $D1_{16}$

Bit	Signal	Description	Access
7..4	Reserved	Reserved	w
3	RESET_CLEAR	This bit is used to clear the Reset Status register. 0: Reset Status register will not be cleared. 1: Reset Status register will be cleared.	w
2..0	Reserved	Reserved	w

Reset Status Register

The Reset Status register is used to identify the source of the last reset. If all bits are cleared (0) the last reset was a power-on reset. It is not possible to set more than one bit at the same time. Once one of the bits has been set (1), it can be cleared (0) by setting the RESET_CLEAR bit in the Reset Clear Control register.

Table 50: *Reset Status Register*

Base Address: $1FF.F160.0100_{16}$
Offset: $D4_{16}$

Bit	Signal	Description	Access
7..5	Reserved	Reserved	r
4	RST CPCI	This bit reflects whether the last reset has been generated through a CompactPCI reset of the CPSB-560. 0 (default): No CompactPCI reset has been triggered. 1: The CompactPCI reset has been triggered.	r
3	RST J3	This bit reflects whether the last reset has been generated through the reset key on the RTB-505 front panel. 0 (default): Reset key of RTB-505 was not pressed. 1: Reset key of RTB-505 was pressed.	r
2	RST WD	This bit reflects whether the last reset was caused by a watchdog timer time out. 0 (default): No watchdog timer reset has been triggered. 1: The watchdog timer reset has been triggered.	r
1	RST J2	This bit reflects whether the last reset has been generated through a reset of the CPSB-560 J2 connector (J2-17). 0 (default): No reset from J2 has been triggered. 1: A reset from J2 has been triggered.	r
0	RST KEY	This bit reflects whether the last reset has been generated through the reset key on the CPSB-560 front panel. 0 (default): Front panel reset key has not been pressed. 1: Front panel reset key has been pressed.	r

Board Status Registers

The Board Status registers are used to identify the current configuration of the board. The switch settings can be read from two registers.

Switch 1 and 2 Register

This register is used to read the switch settings of the switches 1 and 2. The functionality of the switches can be seen in “Switch Settings” page 2-16.

Table 51: *Switch 1 and 2 Register*

Base Address: $1FF.F160.0100_{16}$
Offset: $E0_{16}$

Bit	Signal	Description	Access
7	SW2-4	0: Respective switch is set to ON. 1: Respective switch is set to OFF.	r
6	SW2-3		r
5	SW2-2		r
4	SW2-1		r
3	SW1-4		r
2	SW1-3		r
1	SW1-2		r
0	SW1-1		r

Switch 3 Register

This register is used to read the switch settings of switch SW3.

Table 52: *Switch 3 Register*

Base Address: $1FF.F160.0100_{16}$
Offset: $E2_{16}$

Bit	Signal	Description	Access
7..4	Reserved	Reserved	r
3	SW3-4	0: Respective switch is set to ON. 1: Respective switch is set to OFF.	r
2	SW3-3		r
1	SW3-2		r
0	SW3-1		r

Board Configuration Status register

This register reflects the hardware configuration of the SPARC/CPSB-560.

Table 53: *Board Configuration Status Register*

Base Address: $1FF.F160.0100_{16}$
Offset: $E4_{16}$

Bit	Signal	Description	Access
7	Reserved	Reserved	r
6	SCSI-ACT	This bit displays the status of the SCSI controller activity signal for the SCSI channel. 0: Activity signal set to 0 1: Activity signal set to 1	r
5	Reserved	Reserved	r
4	SCSI AUTO-TERMINATION	This bit is used to read the switch setting of the SW 1-2 on the RTB-505. 0: Switch SW 1-2 is on, i.e. SCSI autotermination is disabled 1: Switch SW 1-2 is off, i.e. SCSI autotermination is enabled	r
3..2	Reserved	Reserved	r
1	RTB-PRESENT	Shows whether a RTB-505 is plugged at the rear side of the SPARC/CPSB-560. 0: No RTB-505 present 1: RTB-505 present	r
0	Reserved	Reserved	r

FPGA Revision Status Register

The table below shows the implementation of the FPGA Revision Status register. The FPGA revision is BCD encoded, therefore, the maximum will be 99 decimal.

Table 54: *FPGA Revision Status Register*

Address: $1FF.F160.0100_{16}$
Offset: EE_{16}

Bit	Signal	Description	Access
3..0	Lower BCD digit	Lower digit of FPGA revision	r
7..4	Upper BCD digit	Upper digit of FPGA revision	r

I^2C Register

The I^2C register implemented in the FPGA is used to access the local I^2C bus with attached SPDs.

Table 55: *I^2C Register*

Base Address: $1FF.F160.0100_{16}$
Offset: FF_{16}

Bit	Signal	Description	Access
7..3	Reserved	Reserved	r / w
2	I2C-DATAOUT	This bit is used by software to write to the I^2C-data-line. 0: I^2C data line is driven low. 1: I^2C data line is driven high by an external pullup.	r / w
1	I2C-CLK	Corresponds to the I^2C clock line and must be set by software to toggle the I^2C clock. 0: I^2C clock is 0. 1: I^2C clock is 1.	r / w
0	I2C-DATAIN	Reflects the current status of the I^2C data line. 0: I^2C data line is 0. 1: I^2C data line is 1.	r / w

A

Battery Exchange

Battery Exchange

The battery provides a data retention of five 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 example in the 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**
  Exchange the battery before five years of actual battery use have elapsed. Therefore, exchange the battery before five years have elapsed. Exchanging the battery always results in data loss of the devices which use the battery as power backup. Therefore, back-up affected data before exchanging the battery.

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

1. Remove board from backplane according to “Board Installation” page 2-18
2. Remove battery

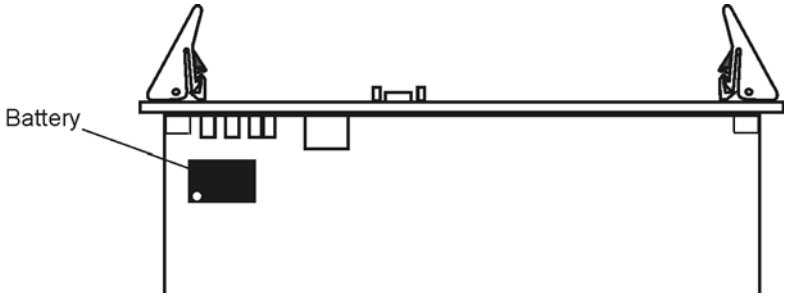


Figure 27: *Battery Location*

3. Place CPSB-560 in front of you with backplane connectors pointing towards you
4. Install new battery with white point on lower left corner of battery (see figure above)
5. Install board back into backplane according to the “Board Installation” chapter

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		
Unable to insert board into backplane	Damaged plugs, bent or broken pins: backplane defect	1. Check backplane 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	Replace backplane
After Power-On		
Powering on the board fails	Backplane voltages for board 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
	Board defect	Replace board
	Damaged plugs, bent or broken pins: backplane defect	1. Check backplane slot position to be used for bent or broken pins 2. Replace backplane

		
During Boot-up Procedure		
Board does not boot	Boot device is not partitioned according to used operating system	Check partition according to operating system's needs.
	Wrong boot variable settings in NVRAM	1. Make sure the variable <auto-boot?> is set to true 2. Make sure variables <boot-device> and <boot-file> are set according to your needs 3. Make sure that boot file name set in variable <boot-file> is present.
	Boot sequence not correct	Correct boot sequence in variable <boot-device>
	Interrupts are not set correctly	Set interrupts correctly
	Wrong configuration of boot devices	Configure boot devices correctly
	Wrong switch settings If switch SW2-1 is set to ON and there is no or no executable file in the boot section of the flash EPROM, the board does not boot.	1. Remove board from system 2. Set switch SW2-1 to OFF to boot from boot PROM 3. If applicable, copy executable file into boot section of flash EPROM. 4. Install board into system 5. Reset board
During Board Operation		
Application software does not work	Not enough disk capacity on mass storage device	Add disk capacity
	Not enough system memory	Add system memory
Connected devices do not work	Device defect	Replace device
	Device not connected to power supply	Connect device to power supply

		
	Driver missing or wrong driver	Install appropriate device driver. For information which driver of the Solaris Driver Package must be installed for which device, see Table 8 “Devices and their Appropriate Drivers” page 2-26.
	Wrong device settings	If one IDE device is connected to a bus, this device must be master. If two devices are connected to the same bus, one must be the master and the other device must be the slave. For information on how to set the device as master / slave, refer to the device’s user’s documentation.
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. For information which driver of the Solaris Driver Package must be installed for which device, see Table 8 “Devices and their Appropriate Drivers” page 2-26.
	Board defect	Replace board
Ethernet interface on CPSB-560’s front panel or on RTB does not work	Wrong switch settings	If you want to use the interface on the CPSB-560 front panel, set switch SW1-4 to OFF. If you want to use the interface on the RTB-505, set switch SW1-4 to ON.

		
	Ethernet drivers from Solaris and from the Motorola Solaris Driver Package are installed.	Uninstall the Solaris Ethernet driver SUNWdmfex by entering into the Solaris prompt: **`pkgrm sunwdmfex`**
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 match with interface specification of board. 2. Replace module if specifications do not match
	Module not installed correctly	Check if module fits perfectly in socket.
RTB does not work	RTB defect	Replace RTB
	RTB installed on wrong slot position	Install RTB on adjacent slot position of the CPSB-560.
	RTB not defined for the CPSB-560	Install RTB defined for the CPSB-560, for example, the RTB-505.

Index

A

add-dropin 4-29
Automatic boot 4-12

B

Board temperature threshold 5-12
boot 4-12
Boot configuration parameters 4-12
Boot devices 4-13
Boot parameters 4-13
Booting automatically 4-12
Booting with POST 4-9
Booting without POST 4-9

C

CPU temperature threshold 5-12

D

Default boot configuration parameters 4-12
delete-dropin 4-30
Device alias definitions for SCSI1 4-13
Device aliases
    IDE devices 4-14
    SCSI devices 4-13
Displaying
    Device alias definitions 4-13
    Device paths 4-13
Displaying system information 4-24

E

Ethernet driver problems 2-26

F

flash-update 4-31, 4-32

H

help 4-24
High-availability support 2-18
Hot swap support 2-18

I

I2C register 6-33
I2C slave addresses 5-11
IDE device alias definitions 4-14
IPMI drivers
    Solaris 2-27

L

Local I2C bus 5-10

O

OpenBoot boot parameters 4-12

P

PHYceiver address 5-6
PMC slot VI/O 2-12
Problems with Ethernet driver 2-26
Problems with SUNWdmfex 2-26

S

SCSI device alias definitions 4-13
SUNWdmfex driver 2-26
System Configuration register address map 6-11
System configuration register address map .6-12

T

Threshold values
    Board temperature 5-12
    CPU temperature 5-12

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP