SPARC/CPCI-52x(G)
Technical Reference Manual

P/N 208914 Edition 2.0
November 2000

A SOLECTRON COMPANY

World Wide Web: www.forcecomputers.com

24-hour access to on-line manuals, driver updates, and application notes is provided via SMART, our SolutionsPLUS customer support program that provides current technical and services information.

Headquarters

The Americas

Force Computers Inc.
5799 Fontanoso Way
San Jose, CA 95138-1015
U.S.A.

Tel.: +1 (408) 369-6000
Fax: +1 (408) 371-3382
Email support@fci.com

Europe

Force Computers GmbH
Prof.-Messerschmitt-Str. 1
D-85579 Neubiberg/München
Germany

Tel.: +49 (89) 608 14-0
Fax: +49 (89) 609 77 93
Email support@force.de

Asia

Force Computers Japan KK
Shiba Daimon MF Building 4F
2-1-16 Shiba Daimon
Minato-ku, Tokyo 105-0012
Japan
Tel.: +81 (03) 3437 3948
Fax: +81 (03) 3437 3968
Email smiyagawa@fci.com

Table of Contents

List of Tables and Figures

Using This Manual

This section does not provide information on the product, but on standard features of the manual itself:

- its structure,
- special layout conventions,
- and related documents.

Audience of the Manual

This *Technical Reference Manual* is intended for hard- and software developers installing and integrating the SPARC/CPCI-52x(G) into their systems.

Overview of the Manual

This *Technical Reference Manual* provides a comprehensive hardware and software guide to the SPARC/CPCI-52x(G).

The *Technical Reference Manual* includes

- safety notes: see section 1 “Safety Notes” on page 1.
- a brief overview of the product, the specifications, the ordering information: see section 2 “Introduction” on page 5.
- the installation instructions, a mechanical overview of the product, initialization prerequisites and requirements, the default configuration, for example, the default switch setting and the connector pinouts of the SPARC/CPCI-52x(G).

  The installation instructions are separated into 3 sections: one general for the complete SPARC/CPCI-52x(G), one for the Base-520(G), and one for the I/O-52x(G). This modular concept should help you to find the information needed for your SPARC/CPCI-52x(G) configuration easily.

  The installation instructions also appear as the product’s *Installation guide* – a separate manual delivered together with each product shipped.
- a detailed hardware description : see section 6 “Hardware Description” on page 65.
- a detailed description of OpenBoot which controls the CPU board operations: see section 7 “FORCE OpenBoot Enhancements” on page 101.

The Sun OpenBoot 3.x manuals are available on the following web site: http://docs.sun.com.

The following data sheets of board components are relevant to the SPARC/CPCI-52x(G). They contain appropriate information on configuring and integrating the board in systems and can be found on the respective company's webpage.

- Advanced PCI Bus Bridge – Sun SME2411 (http://www.sun.com)
- UltraSPARC-IIi – Sun SME1040 (http://www.sun.com)
- PCI I/O Controller – Sun STP2003QFP (http://www.sun.com)
- PHYceiver – ICS ICS1890 (http://www.icsinc.com)
- PCI-Ultra SCSI (Fast-20) I/O Interface – Symbios SYM53C875 (http://www.lsilogic.com)
- Enhanced Serial Communication Controller – Siemens SAB82532 (http://www.siemens.com)
- Super I/O – NS PC87332VLJ (http://www.national.com)
- Audio Controller – Crystal CS4231A (http://www.crystal.com)
- RTC/NVRAM SGS M48T58Y (http://www.st.com)
- Flash Memory – AMD Am29F0808B and Am29F016B (http://www.amd.com)
- Temperature Sensor – NS LM75 (http://www.national.com)

Publication History of the Manual

Table a History of manual publication

Ed.	Date	Description
1	Feb/1998	First print
1.3	Jan/1999	Thoroughly revised, corrected SCSI-related OpenBoot command description, added descriptions for Miscellaneous Control Register, Miscellaneous Control and Status Register as well as ENUM Interrupt Control Register, extended temperature sensor description Added descriptions for installing Solaris, added description for OpenBoot plcc2tsop command (version 3.10.4 or greater) Added note for I2C_SDAO Battery maintenance safety note changed SPARC/MEM-50-5 information added SMART Service information added OpenBoot 3.10.6 information added
2.0	November 2000	Section “Safety Notes” included, Solaris versions for required software packages specified, maximum power supply data revised, section “Data Sheets” and data sheet cross-references removed

Fonts, Notations and Conventions

Table b **Fonts, notations and conventions**

Notation	Description
	All numbers are decimal numbers except when used with the following notations:
0000.0000_{16}	Typical notation for hexadecimal numbers (digits are 0 through F), e.g. used for addresses and offsets. Note the dot marking the 4th (to its right) and 5th (to its left) digit.
0000_8	Same for octal numbers (digits are 0 through 7)
0000_2	Same for binary numbers (digits are 0 and 1)
`Program`	Typical character format used for names, values, and the like. It is used to indicate when to type literally the same word. Also used for on-screen output.
Variable	Typical character format for words that represent a part of a command, a programming statement, or the like, and that will be replaced by an applicable value when actually applied.

Table c **Product naming conventions**

Used Name	Description
SPARC/CPCI-52x(G)	General name for all available product configurations
Base-520(G)	General name for all available base board configurations
Base-520G	Name for base board with UPA64S slot
Base-520	Name for base board with 1 slot front panel
I/O-52x(G)	General name for all available I/O-board configurations
I/O-52xG	General name for I/O-board configurations, G stands for an additional slot
I/O-522(G)	General name for peripheral slot I/O-board
I/O-523G	Name for system slot I/O-board with second CompactPCI interface

Icons for Ease of Use: Safety Notes and Tips & Tricks

The following 3 types of safety notes appear in this manual. Be sure to always read and follow the safety notes of a section first – before acting as documented in the other parts of the section.

Danger

Dangerous situation: serious injuries to people or severe damage to objects.

Caution

Possibly dangerous situation: slight injuries to people or damage to objects possible.

Note: **No danger encountered. Pay attention to important information marked using this layout.**

---

1 Safety Notes

This section provides safety precautions to follow when installing, operating, and maintaining the SPARC/CPCI-52x(G). For your protection, follow all warnings and instructions found in the following text.

General notes

This *Technical Reference Manual* provides the necessary information to install and handle the SPARC/CPCI-52x(G). As the product is complex and its usage manifold, we do not guarantee that the given information is complete. In case you need additional information, ask your Force Computers representative.

The SPARC/CPCI-52x(G) 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 Force Computers or qualified persons in electronics or electrical engineering are authorized to install, uninstall or maintain the SPARC/CPCI-52x(G). 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.

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

Installation

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

- **Before installing the board, check section 3.1.1 “Requirements” on page 14.**
- **Before touching integrated circuits, ensure that you are working in an electrostatic-free environment.**
- **When plugging the board in or removing it, do not press on the front panel but use the handles.**
- **Before installing or uninstalling the board, read section 3 “Installation” on page 11.**
- **Before installing or uninstalling an additional device or module, read the respective documentation.**

- Ensure that the board is connected to the CompactPCI backplane via both the J1 and the J2 connectors and that power is available on both CompactPCI connectors.

Operation

- While operating the board ensure that the power and environmental requirements as given in table 4 “Maximum power supply values without UPA64S card and PMC modules” on page 15 and table 5 “Environmental requirements of the SPARC/CPCI-52x(G)” on page 15 are met.
- When operating the board in areas of strong electromagnetic radiation ensure that the board is bolted on the CompactPCI rack and shielded by closed housing.

EMC

- If boards are integrated into open systems, always cover empty slots.

Expansion

- Check the total power consumption of all components installed (see the technical specification of the respective components). For the total power consumption of the SPARC/CPCI-52x(G), see table 4 “Maximum power supply values without UPA64S card and PMC modules” on page 15.
- Ensure that any individual output current of any source stays within its acceptable limits (see the technical specification of the respective source).
- Only replace components or system parts with those recommended by Force Computers. In case you use components other than those recommended by Force Computers, you are fully responsible for the impact on EMI and the eventually changed functionality of the product.

Battery change

If a Lithium battery on the board has to be exchanged, observe the following safety notes:

- Incorrect exchange of Lithium batteries can result in a hazardous explosion.
- Always use the same type of Lithium battery as is already installed.

Protect your environment

Always dispose used batteries and/or old boards according to your country’s legislation.

RJ-45 connector

An RJ-45 connector is available on the board. Take into account that the RJ-45 connector type is used for telephone connectors and for twisted pair Ethernet (TPE) connectors. Note that mismatching these 2 connectors may destroy your telephone as well as your SPARC/CPCI-52x(G). Therefore:

- **Make sure that TPE connectors near your working area are clearly marked as network connectors.**
- **Make sure that TPE bushing of the system is connected only to safety extra low voltage (SELV) circuits.**
- **Verify that the length of the electric cable connected to a TPE bushing does not exceed 1 kilometer outside the building.**
- **If in doubt, ask your system administrator.**

2 Introduction

The SPARC/CPCI-52x(G) is a high performance CompactPCI board computer providing a CompactPCI system controller interface including DMA. It is based on

- the UltraSPARC-IIi processor
- the Advanced PCI Bridge (APB) with interfaces to the CompactPCI bus

An UPA64S card can be connected for high performance graphics support.

Memory

- 32 to 1024-Mbyte EDO DRAM
- Up to 1 Mbyte secondary (L2) cache
- 1 Mbyte PLCC boot PROM and 2 Mbyte TSOP boot flash EPROM
- Up to 4 Mbyte user flash EPROM

Interfaces

Interfaces of the Base-520(G):

- PCI bridge for 7 CompactPCI slots
- 10BaseT/100BaseTx Ethernet on front panel and MII on backplane
- Wide Ultra SCSI
- 2 serial I/O ports RS-232
- Floppy disc, parallel and Keyboard/Mouse ports
- UPA64S (only Base-520G)

Interfaces of the I/O-52x(G):

- 10BaseT/100BaseTx Ethernet on front panel and MII on backplane
- Wide Ultra SCSI
- 2 standard PMC card slots
- PCI bridge for additional 7 CompactPCI slots (only I/O-523G)

Real-time clock

A real-time clock with on-board battery backup is also available.

Table 1 Specifications of the SPARC/CPCI-52x(G)

Processor	UltraSPARC-IIi with 300 MHz
Shared main memory	32 MByte to 1 GByte EDO DRAM with ECC
L2 cache	256 KByte or 1 MByte late write SRAM with parity
PMC slots	2 for 32 bit with 33 MHz PMC modules
CompactPCI interface	32 bit with 33 MHz
SCSI	Wide Ultra SCSI I/O on front panel and backplane
Ethernet	MII and 10BaseT/100BaseTx half and full duplex Ethernet Twisted Pair on front panel and MII on backplane
Parallel port with DMA	Centronics compatible, uni- or bidirectional I/O on backplane
Floppy Disk Interface	I/O on backplane
Serial I/O	2 ports with RS-232 configuration (as factory option RS-422) I/O on front panel or backplane
Audio Port	I/O on front panel (microphone and headphone) or backplane
Keyboard/Mouse Port	I/O on front panel or backplane
Boot PROM (PLCC)	1 MByte PROM (OTP)
Boot flash EPROM (TSOP)	2 MByte flash EPROM On-board programmable with hardware write protection
User flash EPROM (TSOP)	Up to 4 MByte (2 MByte in default configuration) On-board programmable with hardware write protection
RTC/NVRAM/Battery	M48T58; NVRAM reserved for OpenBoot
Additional Features	Reset and abort key, status LEDs, hexadecimal display, rotary switch
Firmware	OpenBoot with diagnostics
Power consumption	see “Power supply” on page 14
Environm. Conditions	see “Thermal requirements” on page 15
PCI compliants	CompactPCI Specification PICMG 2.0 R2.1 PCI Specification Rev. 2.1.

2.1 Product Nomenclature

The SPARC/CPCI-52x(G) is available in several variants, with or without I/O-52x(G) as well as several memory and speed options. Consult your local sales representative to confirm availability of specific combinations. The table below explains the general product nomenclature.

Table 2 Nomenclature of the SPARC/CPCI-52x(G)

SPARC/CPCI-52xG/mmm-sss-c-uu-ggg	
x = 0 x = 2 x = 3	Base-520(G) Base-520(G) + I/O-522(G) Base-520G + I/O-523G
G	extra slot (for an UPA64S card)
mmm	DRAM capacity in MByte
sss	CPU speed in MHz
c	L2-cache size in KByte divided by 256 (e.g.: 4=1024 KByte)
uu	User flash EPROM size in MByte
ggg	UPA64S card type if preinstalled

2.2 Ordering Information

The following table is an excerpt from the SPARC/CPCI-52x(G) ordering information at the time of print. Contact your local FORCE COMPUTERS representative for current information.

Table 3 Excerpt from the product's ordering information

Product name	Description
SPARC/CPCI-520…	Base-520(G)
...G/64-300-4-2	with 64 MByte DRAM, 300 MHz CPU, 1 MByte L2 cache, 2 MByte user flash EPROM, and UPA64S slot
SPARC/CPCI-522…	Base-520(G) and I/O-52x(G) (dual SCSI, dual Ethernet, 2 PMC slots)
...G/64-300-4-2	with 64 MByte DRAM, 300 MHz CPU, 1 MByte L2 cache, 2 MByte user flash EPROM, and UPA64S slot
SPARC/MEM-…	
...50M/128	user upgradable memory module for slot 2 and 3, 128 Mbyte
...50M/256	user upgradable memory module for slot 2 and 3, 256 Mbyte
...50U/128	user upgradable memory module for slot 2 or 4, 128 Mbyte
...50U/256	user upgradable memory module for slot 2 or 4, 256 Mbyte
Accessories SPARC/…	
…IOBP-520/CPU	I/O panel for the Base-520(G)
…IOBP-520/IO	I/O panel for the I/O-52x(G)
…CPCI-520/AccKit/CPU	I/O panel for the Base-520(G) with cables: 2 serial splitter cable, 1 Micro D-Sub and 1 flat ribbon SCSI cable, 1 TPE cable.
…CPCI-520/AccKit/IO	I/O panel for the I/O-52x(G) with cables: 1 flat ribbon SCSI cable, 1 TPE cable.
…UPA/C2D	Creator 2D graphic card
…UPA/C3D	Creator 3D graphic card
…CPCI-52x/TM	*Technical Reference Manual Set* for SPARC/CPCI-52x(G).

3 Installation

This section describes the SPARC/CPCI-52x(G) variants you may purchase from FORCE COMPUTERS. It is intended to get an overview over all possible configurations with named components which will help to find the information necessary for your configuration in this manual.

How to begin installation

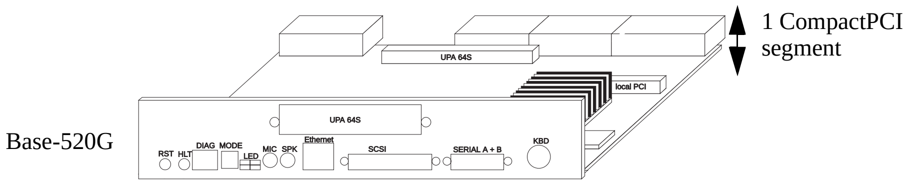
First read the Safety Notes and the Installation Prerequisites and Requirements (see section 1 “Safety Notes” on page 1 and section 3.1 “Installation Prerequisites and Requirements” on page 14).
Then go through the Base-520(G) installation section and the I/O-52x(G) installation section, depending on the variant you have purchased from FORCE COMPUTERS (see section 4 “Base-520(G) Installation” on page 21 and see section 5 “I/O-52x(G) Installation” on page 53).

SPARC/CPCI-52x(G) variants

There are 4 variants available:

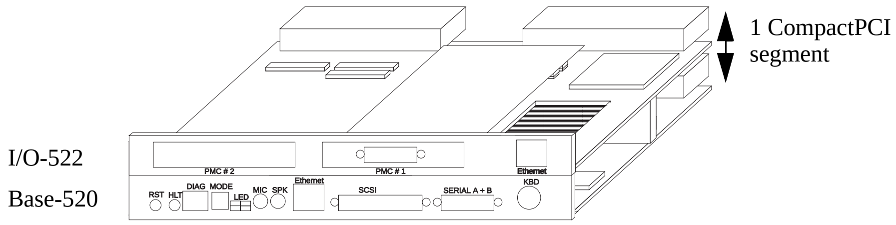
- a SPARC/CPCI-520G
  obtaining the 2 slot high base board with UPA64S card option named in this manual as Base-520G (or in general as Base-520(G)).

Figure 1 SPARC/CPCI-520G (schematic view)



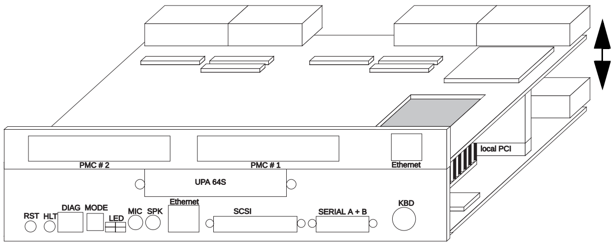
- a SPARC/CPCI-522
  obtaining the 1 slot Base-520 with the peripheral slot I/O-522.

Figure 2 SPARC/CPCI-522 (schematic view)



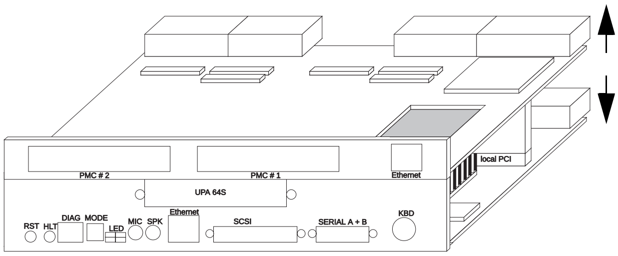
- a SPARC/CPCI-522G
  obtaining the 2 slot base board with UPA64S card option (Base-520G) and with the peripheral slot I/O-522G.

Figure 3 **SPARC/CPCI-522G (schematic view)**



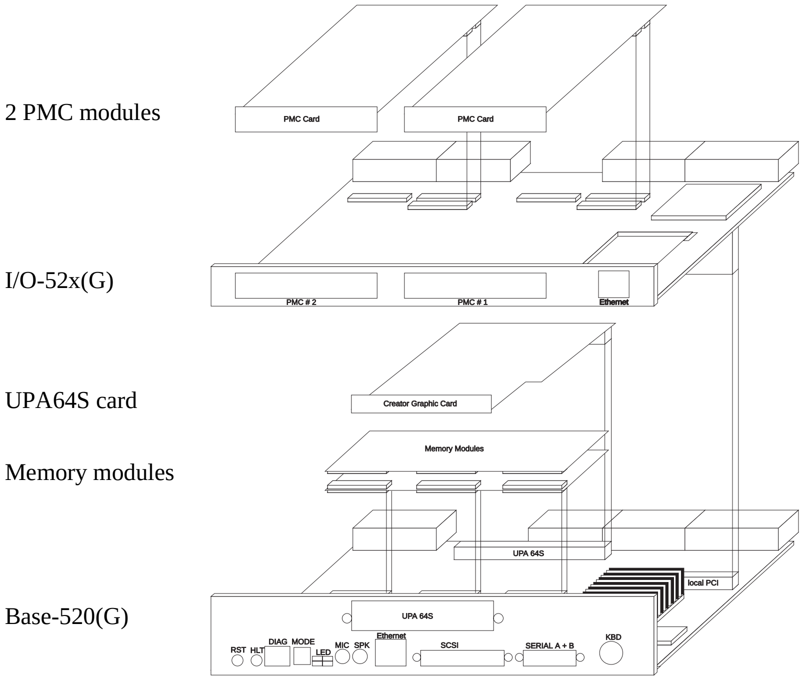
- and a SPARC/CPCI-523G
  which is a dual-segment CompactPCI variant obtaining the 2 slot base board with UPA64S card option (Base-520G) and with the system slot I/O-523G. The dual-segment variant is designed for CompactPCI systems with a backplane consisting of 2 CompactPCI bus segments.

Figure 4 **SPARC/CPCI-523G (schematic view)**



The following figure is intended to get an overview over all available main components of a SPARC/CPCI-52x(G).

Figure 5 **Schematic exploded mechanical construction view**



Caution

The SPARC/CPCI-52x(G) is a sytem board. According to the *CompactPCI Specification PICMG 2.0 R2.1*, the front panel of the SPARC/CPCI-52x(G) shows a triangle. To ensure proper functioning of the board, plug it exclusively in a system slot marked by a triangle.

Danger

The Lithium battery of the RTC/NVRAM provides a data retention of at least 7 years summing up all periods of actual battery use. Therefore FORCE COMPUTERS assumes that there usually is no need to exchange the Lithium battery except for example in the case of long-term spare part handling.
Please observe the following:

- **Exchange the battery before 7 years of actual battery use have elapsed.**
- **Exchanging the battery always results in data loss of the devices which use the battery as power backup. Therefore, back up affected data before exchanging the battery.**
- **Always use the same type of Lithium battery as is already installed.**

- **Use appropriate tools to remove the battery.**
- **When installing the new battery ensure that the marked dot on top of the battery covers the dot marked on the chip.**
- **Used batteries have to be disposed according to your country's legislation.**

3.1 Installation Prerequisites and Requirements

Caution

Before powering up check:

- **this section for installation prerequisites and requirements**
- **and the consistency of the current switch setting (see section 4.4 "Switch Settings" on page 27).**

3.1.1 Requirements

The installation requires at least

- a power supply
- a minimum airflow meeting the thermal requirements,
- and free CompactPCI backplane slots due to your SPARC/CPCI-52x(G) variant:
  - a system CompactPCI bus slot for the Base-520(G)
  - an additional system CompactPCI bus slot for the IO-523G on the right side of the Base-520(G)
  - an additional system or peripheral CompactPCI bus slot for the IO-522(G) on the right side of the Base-520(G).

Peripheral slot — A peripheral slot of a CompactPCI rack is marked by a circle.

System slot — A system slot of a CompactPCI rack is marked by a triangle.

Signaling level — The SPARC/CPCI-52x(G) is a CompactPCI-universal board operational in 3.3 V or 5 V CompactPCI systems. Therefore, no voltage keys are provided on the CompactPCI interface.

Power supply — The power supply must meet the specifications given in the following table. The values in the table below are maximum values without an UPA64S card installed and without PMC modules.

Table 4 Maximum power supply values without UPA64S card and PMC modules

CPU board	+5 V	+3.3 V	+/-12 V	V I/O
SPARC/CPCI-520	6.5 A	4.6 A	not required	200 mA
SPARC/CPCI-520G	6.5 A	4.6 A	not required	200 mA
SPARC/CPCI-522	7.8 A	5.6 A	not required	200 mA
SPARC/CPCI-522G	7.8 A	5.6 A	not required	200 mA
SPARC/CPCI-523G	7.8 A	5.6 A	not required	400 mA
Creator Graphic Card	1.3 A	2.2 A	not required	not required

Thermal requirements

The operating temperature is 0 °C to +55 °C (humidity 5 % to 95 % non-condensing at +40 °C), when operating the SPARC/CPCI-52x(G) in systems providing a minimum forced airflow of 300 LFM (linear feet per minute). The typical operating temperature of the system is 0 °C to +40 °C.

Table 5 Environmental requirements of the SPARC/CPCI-52x(G)

	Operating	Non-operating
Temperature	0 °C to +55 °C	–40 °C to +85 °C
Forced air flow	300 LFM (linear feet per minute)	–
Temp. change	+/– 0.5 °C/min	+/– 1 °C/min
Rel. humidity	5 % to 95 % noncondensing at +40 °C	5 % to 95 % noncondensing at +40 °C
Altitude	–300 m to +3,000 m	–300 m to +13,000 m

Audio interfaces

Simultaneous use of the audio interfaces available on the front panel and on the backplane can damage on-board hardware or connected devices. For example: never use the headphone/line audio output at the backplane, if a headphone is plugged into the front-panel jack.

- Always use at most one of the interfaces if an audio interface is available on both the front panel and the backplane.

Table 6 Audio interfaces requirements

Interface	Description
Stereo Micro In (op-amp pre-amp with 18 dB gain)	• Signal level: single-ended condensator microphones with signal level –up to 12 mV with 20 dB gain inside Codec enabled –and up to 120 mV with 20 dB gain inside Codec disabled • Availability: on front panel and as factory option on backplane instead of Aux#2 In
Stereo Head-phone/ Line Out	• Signal level: maximum 2 V_{RMS} line-level signal output (also designed to directly drive headphones) • Availability: on front panel and on backplane
Stereo Line In	• Signal level: typical 47 kΩ audio input impedance; maximum full scale input of 2 V_{RM} • Availability: on backplane
Stereo Aux#1 In	• Signal level: ~10 kΩ input impedance; maximum full scale input of 2 V_{RMS} • Availability: on backplane
Stereo Aux#2 In	• Signal level: ~10 kΩ input impedance; maximum full scale input of 2 V_{RMS} • Availability: on backplane
Mono In	• Signal level: typical 47 kΩ audio input impedance; nominally 1 V_{RMS} maximum (centered around 2.1 V) input signal level • Availability: as factory option on front panel instead of Micro In and as factory option on backplane instead of Aux#1 In
Mono Out	• Signal level depends on the setting of `OLB` which is a bit in the Codecs Alternate Feature Enable I register (I16) –maximum 1 V_{RMS} output (centered around 2.1 V) if `OLB` = `1` –or maximum 0.707 V_{RMS} (centered around 2.1 V) if `OLB` = `0` Default is `OLB` = `0`. • Availability: as factory option on backplane instead of Head-phone/Line Out

3.1.2 Memory Modules

The main memory capacity is adjustable via installation of the appropriate memory modules.
The qualified memory modules depend on the SPARC/CPCI-52x(G) processor frequency. They are given in the following table.

Table 7 Qualified memory modules

Processor frequency	Memory modules
up to 300 MHz	SPARC/MEM-50x
	SPARC/MEM-50x-5
333 MHz and above	SPARC/MEM-50x-5

Caution

Do not install SPARC/MEM-50x and SPARC/MEM-50x-5 memory modules on the same board, otherwise system malfunction may occur.

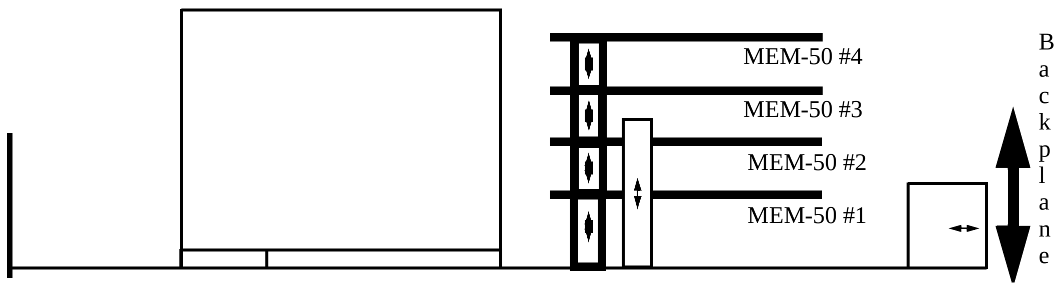
In the following it will be refered to all memory module types as SPARC/MEM-50x.
The Base-520(G) can hold 1 to 4 memory modules providing up to 1 GByte DRAM capacity. 1 memory module can carry 2 memory banks.

Note: **At least 1 lower memory module MEM-50L is required.**

See the following figure for the memory module numbering scheme:

- Memory modules #1 and #2 are located in the first CompactPCI slot the SPARC/CPCI-52x(G) occupies.
- Memory modules #3 and #4 are located in the second CompactPCI slot the SPARC/CPCI-52x(G) occupies.

Figure 6 MEM-50 – memory module numbering scheme



The memory configuration is adjustable to the application's needs via selection of the appropriate memory modules. The memory configuration must fulfill the following requirements:

- The lowest memory module (#1) must be a SPARC/MEM-50L – which is a lower memory module.
- The top memory module (with the greatest number in your configuration given the number scheme in the figure above) can be a SPARC/MEM-50M or SPARC/MEM-50U – which is a middle (M) or upper (U) memory module. The upper module misses the connectors for another memory module to be stacked on top.
- The memory modules between the lowest and the top memory module must be SPARC/MEM-50M, i.e. middle memory modules.
- If a UPA64S card is installed, at most 2 memory modules can be installed and memory module #2 must be a SPARC/MEM-50U, i.e. an upper memory module.
- Note the limitations given by the SPARC/CPCI-52x(G) configuration under consideration (see section 4.2 "Mechanical Construction" on page 23).

Out of the extensive list of possible configurations the following memory module configurations have been qualified (others may be tested and qualified on request):

Table 8 Qualified memory configurations (all data in MByte)

Total capacity	**32**	**64**	**128**	**256**	**384**	**512**	**768**	**1024**
Mem. module #4	–	–	–	–	–	–	–	256
Mem. module #3	–	–	–	–	–	–	256	256
Mem. module #2	–	–	–	–	128	256	256	256
Mem. module #1	32	64	128	256	256	256	256	256

For installation information see the respective *Installation Guide* delivered together with the memory module.

3.1.3 Solaris Installation

When installing Solaris, there are some general installation guidelines to be followed before and during Solaris installation and a specific guideline related to SCSI to be followed after Solaris installation (see "SCSI" on page 20).

General Installation Guidelines

Note: **Solaris versions and hardware updates prior to 2.5.1 11/97 and 2.6 03/98 are not supported.**

Required software packages

In case of Solaris 2.5.1 and Solaris 2.6 the following Solaris software packages must be installed, otherwise Solaris fails to boot.

Table 9 **Required Solaris Packages**

Package	**Description**
`SUNWvplr.u`	SMCC sun4u new platform links
`SUNWvplu.u`	SMCC sun4u new usr/platform links

When setting up Solaris interactively, these packages can be installed by selecting the proper software group in the `Software` dialog. Customize the software groups as follows:

Table 10 **Customizing Solaris**

<table>
<tr><th rowspan="2">Software Group</th><th colspan="2">Customization required for</th></tr>
<tr><th>Solaris 2.5.1</th><th>Solaris 2.6</th></tr>
<tr><td>Entire distribution plus OEM support</td><td colspan="2">No customization is required</td></tr>
<tr><td>Entire distribution</td><td colspan="2" rowspan="4">Select the following clusters:
• SMCC platform links</td></tr>
<tr><td>Developer system support</td></tr>
<tr><td>End user system support</td></tr>
<tr><td>Core system support</td></tr>
</table>

SCSI

The Solaris SCSI driver may revert Wide-SCSI devices, which are connected to the front-panel SCSI connector, to asynchronous mode. However, it is possible to operate such a configuration in synchronous mode also by inserting the following line into `/kernel/drv/glm.conf`:

```
targetn-scsi-options=0x5f8
```

where n is the SCSI ID of the Wide-SCSI device under consideration. In case of several Wide-SCSI devices insert the respective line per device. Terminate the file with a `";".`

3.1.4 Terminal connection

The SPARC/CPCI-52x(G) provides 2 serial interfaces (A and B) which are implemented on the Base-520(G). For the initial power up, a terminal can be connected to interface A via the front-panel 26-pin-MicroD-Sub connector SERIAL A+B. Per default, all serial I/O interfaces provide an RS-232 interface. As factory option the 2 interfaces can be configured as RS-422 interface.
For information on the serial interface connector pinout, see section 4.5.4 “Serial I/O Interface Connector Pinout” on page 34.

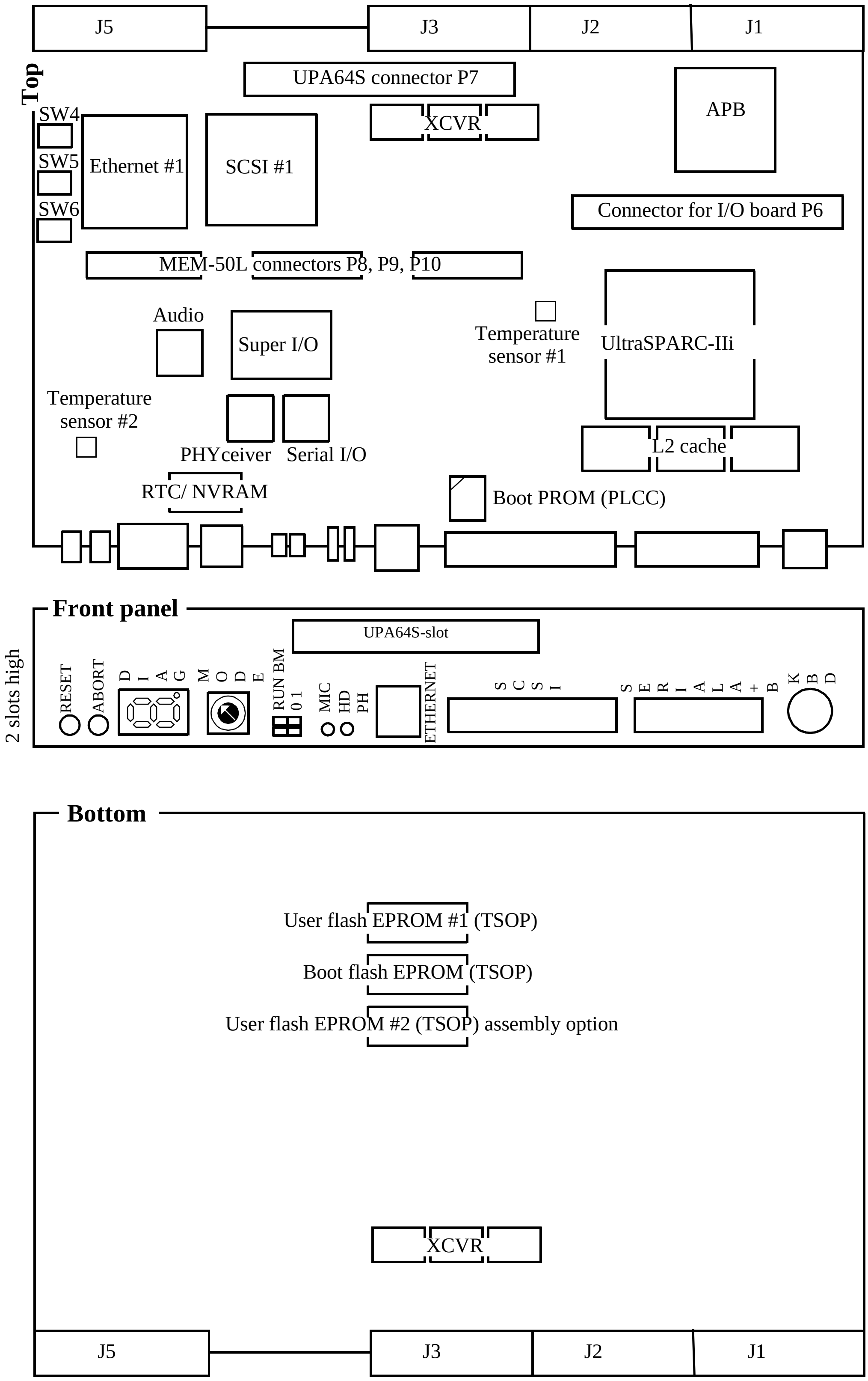
4 Base-520(G) Installation

4.1 Location Overview

The Base-520(G) contains the following main components:

- a UltraSPARC-IIi processor,
- a second level cache (L2 cache),
- a CompactPCI interface,
- a boot PROM (PLCC),
- a boot flash EPROM (TSOP) and an user flash EPROM (TSOP),
- 3 connectors for the memory modules,
- a connector for interfacing to a UPA64S card,
- a connector for interfacing to the I/O-52x(G),
- switches,
- temperature sensors,
- and the following I/O interfaces: SCSI #1, Ethernet #1, floppy, keyboard and mouse, audio, parallel interface as well as the 2 serial interfaces A+B.

Figure 7 **Location diagram of the Base-520(G) (schematic)**



4.2 Mechanical Construction

The Base-520(G) is a CompactPCI computer. It occupies 2 CompactPCI slots and consists of the following major components:

- an I/O connector for the I/O-52x(G),
- an UPA64S connector for an UPA64S card (only Base-520G),
- 3 memory module connectors for up to 4 memory modules. With an installed UPA64S card only 2 memory modules are possible.

The following figures show the Base-520(G) in possible configurations:

Figure 8 **Mechanical construction of a Base-520G**

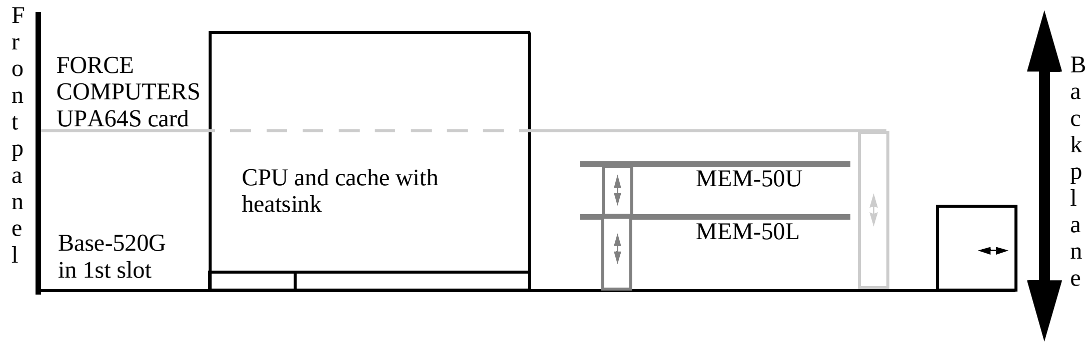


Figure 9 **Mechanical construction of a Base-520G with 4 memory modules**

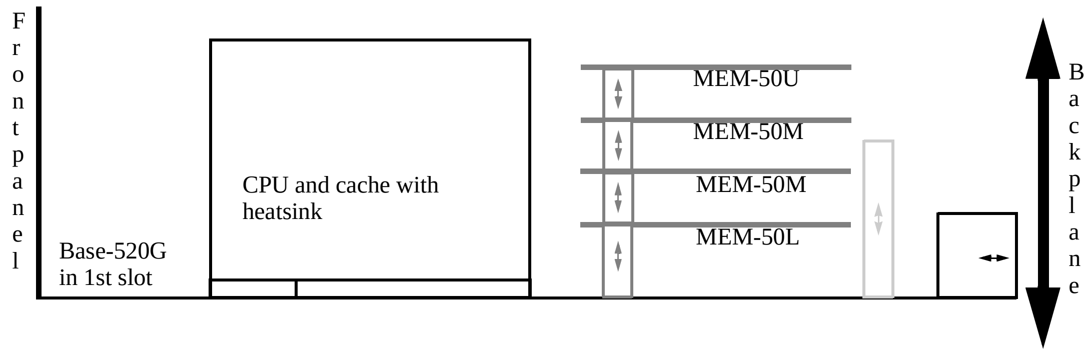
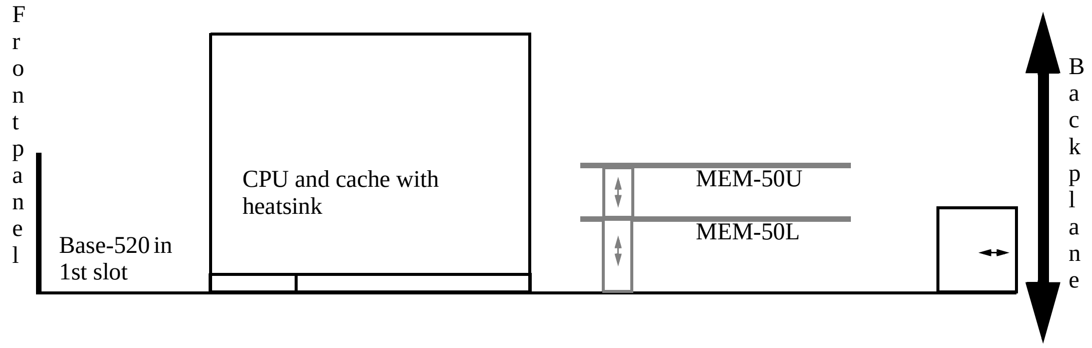


Figure 10 **Mechanical construction of a Base-520**

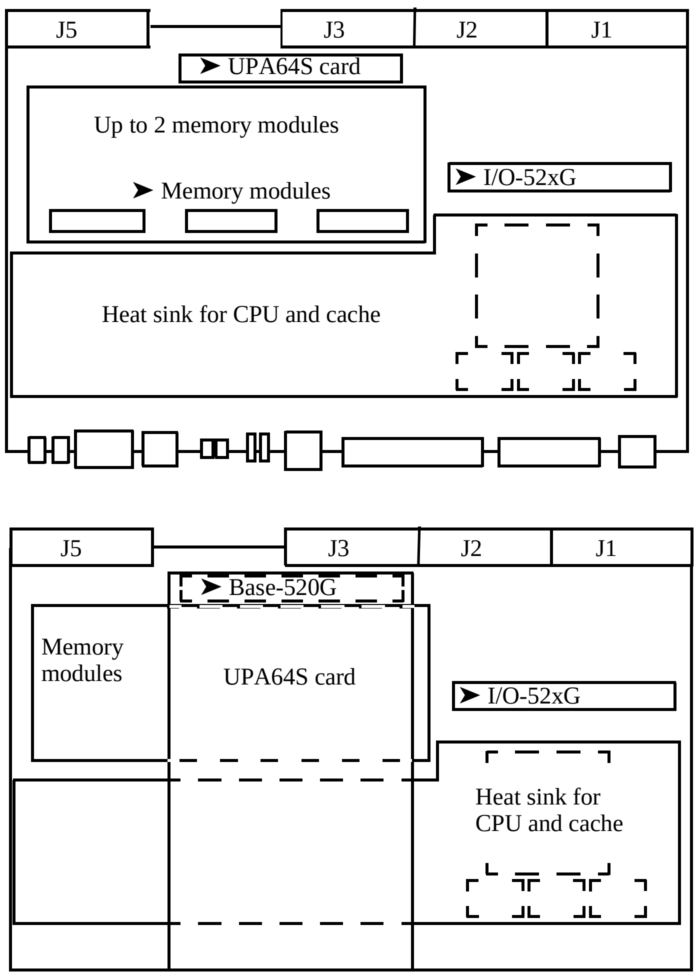


The Base-520 is only available as a 2-slot solution with an I/O-522.

Figure 11 **Components of a 2-slot configuration with UPA64S card (schematic)**

1st of 2 slots Base-520G with up to 2 memory modules

2nd of 2 slots UPA64S card



4.2.1 FORCE COMPUTERS UPA64S Card Installation

You can only install a FORCE COMPUTERS UPA64S card if you purchased a SPARC/CPCI-52xG version. It is connected to the Base-520G via the UPA64S connector P7 (see figure 7 “Location diagram of the Base-520(G) (schematic)” on page 22).

***Note:* Use only UPA64S cards from FORCE COMPUTERS. Throughout this section the term “UPA64S card” always refers to a card purchased from FORCE COMPUTERS and specified for use with a SPARC/CPCI-52xG.**

For the locations mentioned in the description see figure 12 “Installing/Deinstalling an UPA64S card” on page 26.

Installation of a UPA64S card

1. If an I/O-52xG is installed, remove it as described in the I/O-52x(G) installation section. Remove the 2 z-standoffs at location 5 and 6 from the Base-520G by loosing the respective 2 screws. Keep them in a save place to have them available for reusing the I/O-52xG without UPA64S card.
2. If you do not install an I/O-52xG afterwards: Remove the 2 z-standoffs fixed on the UPA64S card by loosing the respective 2 screws and fix the UPA64S card again with 2 of the 4 shorter screws delivered with the UPA64S card on the 2 standoffs which connect it to the Base-520G.
3. Remove the blind panel fixed in the UPA64S front panel slot. Store it in a safe place for later use.
4. Plug the prepared UPA64S card from FORCE COMPUTERS to the respective UPA64S connector on the Base-520G.
5. Fix the UPA64S card with 2 of the 4 short screws at location 5 and 6 on the bottom side of the Base-520G and with the 4 screws and 2 nuts from the blind panel on the front panel at location 1...4.

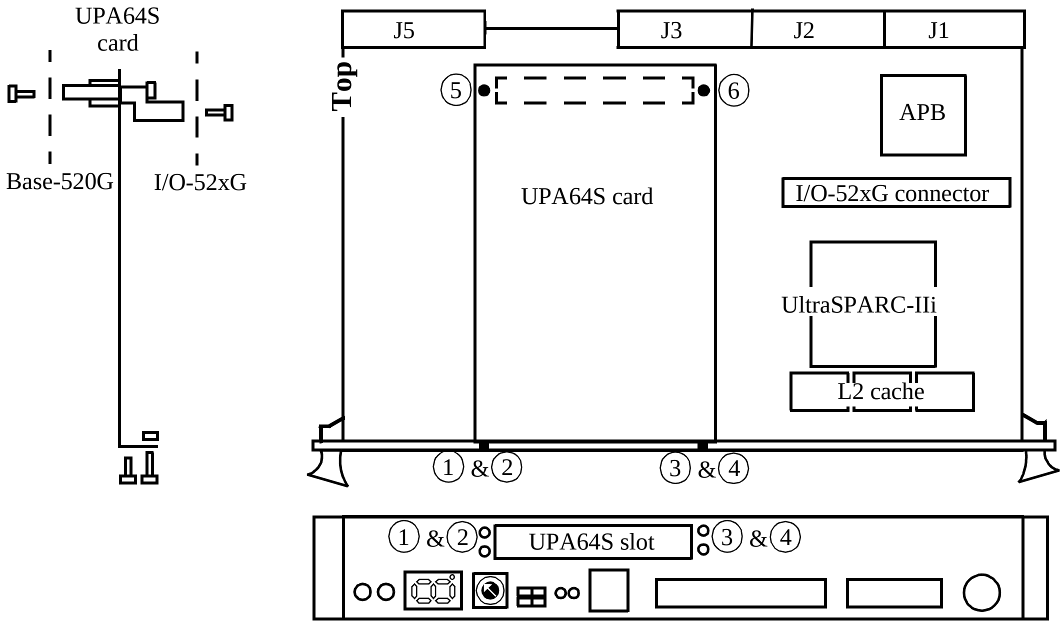
   Now the UPA64S card is installed.
6. If an I/O-52xG was installed, fix it again as described in the I/O-52x(G) installation section.

Uninstalling a UPA64S card

1. If an I/O-52xG is installed, remove it as described in the I/O-52x(G) installation section.
2. Remove the 4 screws and 2 nuts on the front panel at location 1...4. Remove the 2 screws at location 5 and 6 on the bottom side of the Base-520G.
3. Remove the UPA64S card by lifting it.

4. If you do not install the UPA64S card again, fix the blind panel.
5. To install the I/O-52xG again refer to the installation section of the I/O-52x(G).

Figure 12 **Installing/Deinstalling an UPA64S card**



4.3 Powering Up

The initial powering up can be done by connecting a terminal to the front panel serial I/O interface A. The advantage of using a terminal is that you do not need any frame buffer, monitor, or keyboard for initial powering up.

Booting

The SPARC/CPCI-52x(G) boot PROM consists of a 1 MByte PROM (OTP) PLCC socket device (not writeable). Alternatively a 2 MByte TSOP boot flash EPROM device can be enabled by SW6-2. This boot flash EPROM device is writeable if enabled by SW4-3.

Note: **If you have an unformatted floppy disk in a floppy connected to your SPARC/CPCI-52x(G) then the OpenBoot does not come up.**

Per default the SPARC/CPCI-52x(G) is shipped with its boot PROM containing the OpenBoot firmware (see section 4.8 “OpenBoot Firmware” on page 42).

User application

The SPARC/CPCI-52x(G) provides 1 user flash EPROM devices (2M*8) to store user applications. As factory option 2 user flash EPROM devices (2M*8) are possible. For write-protection of the user flash EPROM see SW4-4 in section 4.4 “Switch Settings” on page 27.

4.4 Switch Settings

The following table lists the functions and the default settings of all switches shown in figure 7 “Location diagram of the Base-520(G) (schematic)” on page 22.

Note: **Before powering up the board check the current switch settings for consistency. Do not switch during operation.**

Table 11 **Default switch settings**

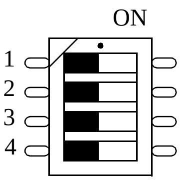
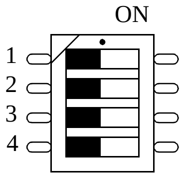
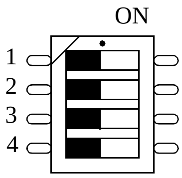
Name and default setting		Function
	SW4-1 OFF	Reset key on front-panel control OFF = RESET key enabled ON = RESET key disabled
	SW4-2 OFF	Abort key control OFF = ABORT key enabled ON = ABORT key disabled
	SW4-3 OFF	Boot flash EPROM write protection (only relevant if SW6-2 = ON) OFF = boot flash EPROM write protected ON = boot flash EPROM write enabled
	SW4-4 OFF	User flash EPROM write protection OFF = user flash EPROM write protected ON = user flash EPROM write enabled

Table 11 Default switch settings (cont.)

Name and default setting		Function
	SW5-1 OFF	SCSI termination for SCSI #1 on front panel OFF = front panel termination automatic ON = front panel termination disabled
	SW5-2 OFF	SCSI termination for SCSI #1 on backplane OFF = backplane termination disabled ON = backplane termination enabled
	SW5-3 OFF	Reserved, must be OFF
	SW5-4 OFF	Reserved, must be OFF
	SW6-1 OFF	Reserved, must be OFF
	SW6-2 OFF	Select boot device OFF = boot from boot PROM ON = boot from boot flash EPROM
	SW6-3 OFF	Reserved, must be OFF
	SW6-4 OFF	Watchdog enable switch OFF = disabled ON = enabled

4.5 Front Panel and Connectors

Front panel features

The features of the front panel are described in the following table. For a location diagram see figure 7 “Location diagram of the Base-520(G) (schematic)” on page 22.

Table 12 Front panel features

Device	Description
RESET	Mechanical reset key: When enabled and toggled it instantaneously affects the SPARC/CPCI-52x(G) by generating a push-button Power On Reset (POR) to the UltraSPARC-IIi. Push-button Power On Reset has the same effect as a Power On Reset from the power supply, with the only difference, that the corresponding status bit (`B_POR`) in the UltraSPARC-IIi `Reset_Control` Register is set and the DRAM refresh is not influenced. For information on disabling the reset key, see “SW4-1” on page 27.
ABORT	Mechanical abort key: When enabled and toggled it instantaneously affects the SPARC/CPCI-52x(G) by generating a push-button external initiated reset (XIR). Push-button external 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 external initiated reset is detected. For information on disabling the abort key, see “SW4-2” on page 27.
DIAG	Software programmable hexadecimal display for diagnostics.
MODE	Hexadecimal rotary switch, decoded with 4 bit. Default setting: F_{16}.
RUN	CPU status LED: green normal operation red the processor is halted or reset is active; it starts blinking to signal that the processor did not access the PCI bus for more than 1 second.
BM	CompactPCI busmaster LED: green if the SPARC/CPCI-52x(G) accesses the CompactPCI as master off otherwise

Table 12 **Front panel features (cont.)**

Device	Description
0, 1	2 software programmable user LEDs. Possible status: off, red, yellow, or green, all colors either permanent or with a blinking frequency of approximately 0.5, 1, or 2 Hz.
MIC	Standard 3.5 mm microphone jack
HDPH	Standard 3.5 mm headphone jack
ETHERNET	Standard Twisted-Pair-Ethernet RJ45 connector for 10BaseT/100BaseTX Ethernet.
SCSI	50-pin shielded fine-pitch connector for standard SCSI
SERIAL A+B	26-pin shielded fine-pitch connector for 2 serial interfaces
KBD	Standard 8-pin mini-DIN connector for keyboard and mouse

On-board connectors

In addition to the front-panel connectors, the Base-520(G) provides on-board connectors for memory modules and for the I/O-52x(G), only the Base-520G provides the UPA64S interface connector. An overview of the on-board connectors is shown in the following table.

Table 13 **On-board connectors**

Connector description and location	Connector type
CompactPCI backplane connector J1, J2, J5	Standard CompactPCI metric, 5 row shielded connectors female
I/O-52x(G) connector P6	100-pin MBus connector male
UPA64S interface connector P7	120-pin UPA connector female
Memory module connectors P8, P9, P10	80-pin SMD connector

Available interfaces on J5

The following list shows the available interfaces on the J5 backplane connector. For the J5 connector pinout see Figure 13, “CompactPCI J5 connector pinout,” on page 36.

- Ultra Wide SCSI #1
- MII #1 Ethernet interface

- Floppy interface
- Parallel interface
- Serial interface A and B
- Keyboard and mouse
- Audio In:

  Stereo Line In,

  Stereo Aux#1 In,

  Stereo Aux#2 In (or Microphone In as factory option)
- Audio Out:

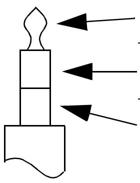
  Stereo Line Out

4.5.1 Audio Interface

The 2 front panel audio interfaces use standard 3.5-mm-phono jacks supporting

- 1 single-ended condenser microphone
- 1 line level signal output, also designed to directly drive low impedance headphones

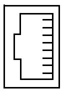
Table 14 Audio interface signals

Connector		Headphone	Microphone
	Tip	Left channel	
	Ring	Right channel	
	Shield	Analog GND	

4.5.2 Ethernet Interfaces

The full duplex Ethernet interface is available at the front panel via a 10BaseT/100BaseTx Twisted-Pair-Ethernet connector.

Table 15 Twisted-Pair-Ethernet #1 connector pinout

Connector	Pin	Signal
RJ-45 TPE 	1	TX+
	2	TX–
	3	RX+
	4	GND
	5	GND
	6	RX–
	7	GND
	8	GND

The Ethernet #1 interface is also accessible at the J5 back panel connector via an MII #1 interface. If Ethernet #1 gets accessed via I/O panel, the front panel connector is normally disabled automatically, for other configurations see the respective jumper settings in the *SPARC/IOBP-520 Installation Guide*. For the J5 connector pinout see Figure 13, “CompactPCI J5 connector pinout,” on page 36.

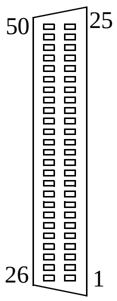
4.5.3 SCSI #1 Connector Pinout

TERMPWR The SCSI #1 interface is single-ended and supports `TERMPWR`.

AUTOTERM Automatic termination mode means the respective termination is disabled when you connect a standard SCSI cable to the front panel connector.

Table 16 50-pin SCSI connector pinout

Signal	Pin	Connector	Pin	Signal
GND	1		26	D0
GND	2		27	D1
GND	3		28	D2
GND	4		29	D3
GND	5		30	D4
GND	6		31	D5
GND	7		32	D6
GND	8		33	D7
GND	9		34	DP0
GND	10		35	GND
GND	11		36	AUTOTERM
n.c.	12		37	n.c.
n.c.	13		38	TERMPWR
n.c.	14		39	n.c.
GND	15		40	GND
GND	16		41	ATN
GND	17		42	GND
GND	18		43	BSY
GND	19		44	ACK
GND	20		45	RST
GND	21		46	MSG
GND	22		47	SEL
GND	23		48	CD
GND	24		49	REQ
GND	25		50	IO

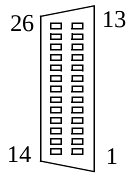


4.5.4 Serial I/O Interface Connector Pinout

Both serial I/O interfaces of the Base-520(G) are independent full-duplex channels. For each of them the 4 signals RXD, TXD, RTS, and CTS are also provided via the respective CompactPCI J5 connector (for interface A and B see figure 13 “CompactPCI J5 connector pinout” on page 36).

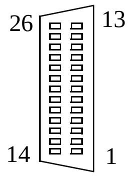
SERIAL A+B on the Base-520(G)’s front panel holds the signals for the 2 serial interfaces A and B.

Table 17 **26-pin serial A+B connector pinout RS232**

Signal	Pin	Connector	Pin	Signal
n.c.	1		14	TxD_B (Output)
TxD_A (Output)	2		15	RxC_A (Input)
RxD_A (Input)	3		16	RxD_B (Input)
RTS_A (Output)	4		17	RTxC_A (Input)
CTS_A (Input)	5		18	RxC_B (Input)
DSR_A (Input)	6		19	RTS_B (Output)
GND_A (Ground)	7		20	DTR_A (Output)
DCD_A (Input)	8		21	DSR_B (Input)
n.c.	9		22	RTxC_B (Input)
n.c.	10		23	GND_B (Ground)
DTR_B (Output)	11		24	TxC_A (Output)
DCD_B (Input)	12		25	TxC_B (Output)
CTS_B (Input)	13		26	n.c.

SERIAL A+B on the Base-520(G)’s front panel holds the signals for the 2 serial interfaces A and B.

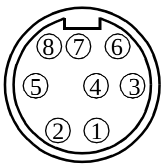
Table 18 **26-pin serial A+B connector pinout RS422 (factory option)**

Signal	Pin	Connector	Pin	Signal
n.c.	1		14	CTS+_B (Input))
CTS+_A (Input))	2		15	nc
RTS–_A (Output)	3		16	RTS–_B (Output)
RTS+_A (Output)	4		17	nc
CTS–_A (Input)	5		18	nc
nc	6		19	RTS+_B (Output)
RxD–_A (Input)	7		20	RxD+_A (Input)
TxD–_A (Output)	8		21	nc
n.c.	9		22	nc
n.c.	10		23	RxD–_B (Input)
RxD+_B (Input)	11		24	TxD+_A (Output)
TxD–_B (Output)	12		25	TxD+_B (Output)
CTS–_B (Input)	13		26	n.c.

4.5.5 Keyboard/Mouse Connector

The SUN-type keyboard/mouse interface is available at the front panel via an 8-pin mini-DIN connector.

Table 19 **Keyboard/mouse connector pinout**

Connector	Pin	Function
	1	GND
	2	GND
	3	+5 V DC
	4	Mouse In
	5	Keyboard Out
	6	Keyboard In
	7	Mouse Out
	8	+5 V DC

4.5.6 CompactPCI Backplane Connector Pinout

J1 and J2

The J1 and J2 connectors implement the CompactPCI 64-bit connector pinout as specified by the CompactPCI Specification. Therefore, this section only documents the pinout of the J5 connector.

J3

J3 is reserved.

J5

Besides the CompactPCI specific pinout the following interfaces are available on the CompactPCI J5 connector (the names used in the following pinout is given in brackets):
SCSI (`SCSI`), MII (`MII`), parallel (`LPT`), floppy (`FDC`), serial interface A (`SerA`), serial interface B (`SerB`), audio (`AUD`), keyboard (`KBD`), mouse (`MSE`), fused 5 V power for the I/O panel (`VP5`).

The pinout shown in the figure below applies to RS-232 configuration of the Base-520(G)'s serial I/O interfaces. Serial I/O interfaces configured for RS-422 (factory option) are only available via the front panel connector.

Figure 13 **CompactPCI J5 connector pinout**

A	B	C		D	E
SCSI #1 D8	SCSI #1 D9	SCSI #1 D10	1	SCSI #1 D11	n.c.
SCSI #1 SEL	SCSI #1 CD	SCSI #1 REQ	2	SCSI #1 IO	WIDETERMPWR
SCSI #1 ATN	SCSI #1 BSY	SCSI #1 ACK	3	SCSI #1 RST	SCSI #1 MSG
SCSI #1 D4	SCSI #1 D5	SCSI #1 D6	4	SCSI #1 D7	TERMPWR
SCSI #1 D0	SCSI #1 D1	SCSI #1 D2	5	SCSI #1 D3	SCSI #1 DP0
SCSI #1 D12	SCSI #1 D13	SCSI #1 D14	6	SCSI #1 D15	SCSI #1 DP1
MII #1 RXD3	MII #1 RXD2	MII #1 RXD1	7	MII #1 RXD0	MII #1 RX_CLK
MII #1 RX_DV	MII #1 COL	MII #1 CRS	8	MII #1 RX_ER	MII #1 MGT_DIO
MII #1 TXD3	MII #1 TXD2	MII #1 TXD1	9	MII #1 TXD0	MII #1 TX_CLK
FDC HDSEL	FDC DSKCHG	MII #1 TX_EN	10	MII #1 TX_ER	MII #1 MGT_CLK
FDC WDATA	FDC WGATE	FDC TRK0	11	FDC WP	FDC RDATA
FDC DR0	FDC DR1	FDC MTR0	12	FDC DIR	FDC STEP
FDC EJECT	FDC DENSEL	FDC DSENS	13	FDC INDEX	VP5_IOBP
LPT BSY	LPT ERR	LPT SLIN	14	LPT INIT	n.c.
VP5_IOBP	LPT PE	LPT SLCT	15	LPT AFD	n.c.
LPT D4	LPT D5	LPT D6	16	LPT D7	LPT ACK
LPT D0	LPT D1	LPT D2	17	LPT D3	LPT STB
SerA RXD	SerA CTS	SerB DCD	18	SerB CTS	SerB RXD
SerA TXD	SerA RTS	SerA DCD	19	SerB RTS	SerB TXD
SerA DTR	KBD DOUT	KBD DIN	20	MSE DIN	SerB DTR
AUD RLINEIN	AUD RAUX2IN	AUD RAUX1IN	21	AUD ROUT	AUD MOUT
AUD LLINEIN	AUD LAUX2IN	AUD LAUX1IN	22	AUD LOUT	AUD AGND

Audio factory option

As factory option the following signals are routed to the mentioned pins instead of the signals mentioned in the connector pinout above:

- Pin 21 row C: AUD MIN (Mono In)
- Pin 21 row B: AUD RMICIN (Right Micro In)
- Pin 22 row B: AUD LMICIN (Left Micro In)

I/O panel

As a separate price list item an I/O panel is available for the Base-520(G), the SPARC/IOBP-520/CPU. An extended variant is the SPARC/CPCI-520/AccKit/CPU which contains additionally to the I/O panel the following cables:

–a serial splitter cable for the front panel and the I/O panel

–a flat ribbon SCSI cable for the I/O panel

–a Micro D-Sub SCSI cable for the front panel

–and a Twisted-Pair-Ethernet cable for the front panel or I/O panel.

The I/O panel supports the following interfaces:

- Fast/Wide SCSI #1,
- MII #1 Ethernet,
- Serial A/B interface,
- Audio interface,
- Keyboard/Mouse,
- Parallel interface,
- and Floppy interface.

Danger

The SPARC/IOBP-520/CPU and the SPARC/CPCI-520/AccKit/CPU is especially designed for the Base-520(G). Do not use any other I/O panels on the Base-520(G).
Use only the front panel or the backpanel Ethernet interface, not both. Check the configuration of your I/O panel.
All switches on the Base-520(G) concerning the SCSI-bus termination must be configured so, that the corresponding backplane terminator (SW5-2) is disabled! This is necessary as the I/O panel includes automatic termination for the backplane SCSI-bus.

4.6 SCSI #1 Configuration

Note: **Correct SCSI bus selection: The Base-520(G) provides 1 SCSI bus, SCSI #1. A further SCSI controller, SCSI #2, is available with the I/O-52x(G). Its termination is described in the I/O-52x(G) installation section.**

SCSI #1 termination

The Base-520(G)'s SCSI #1 bus is accessible via the Base-520(G)'s front-panel SCSI #1 connector (8-bit SCSI) and via the Base-520(G)'s J5 connector (Wide SCSI). Therefore, the Base-520(G) holds 2 distinct SCSI bus terminations to enable correct termination of the SCSI #1 bus. Associated to the 2 terminations there are 2 switches – SW5-1 and SW5-2 – which allow easy selection of a valid SCSI #1 bus configuration.

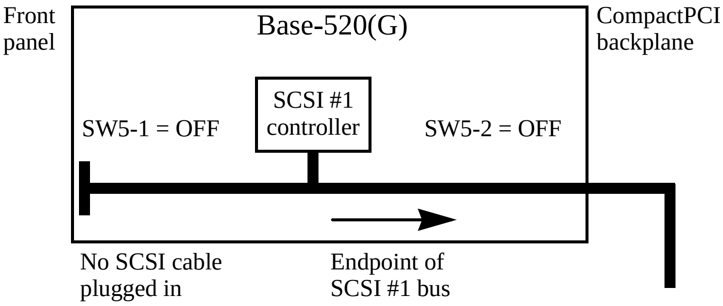
There are 4 valid Base-520(G) switch settings corresponding to valid SCSI #1 bus configurations. The following factors differentiate the valid SCSI #1 bus configurations:

- the Base-520(G)'s location within the SCSI #1 bus: Is the Base-520(G) located at an endpoint of the SCSI #1 bus?
- the connector(s) being used from the SCSI #1 bus:
  - Is a SCSI cable plugged into the Base-520(G)'s front-panel SCSI connector?
  - Is the Base-520(G)'s CompactPCI J5 connector used by the SCSI #1 bus?
  - Are both Base-520(G) connectors used by the SCSI #1 bus?
- the SCSI device type being connected to the SCSI #1 bus: Is a Wide-SCSI device connected to the J5 connector?

Each of the following configuration descriptions starts with identifying the SCSI #1 bus configuration being covered and ends with defining the correct switch setting corresponding to the configuration under consideration.

Default configuration 1 for 8 bit SCSI

• The default configuration 1 is covered by the default switch setting: The Base-520(G) is located at an endpoint of the SCSI #1 bus, the SCSI #1 bus is extended via the CompactPCI backplane (J5 connector), but no SCSI cable is plugged into the front-panel SCSI connector:



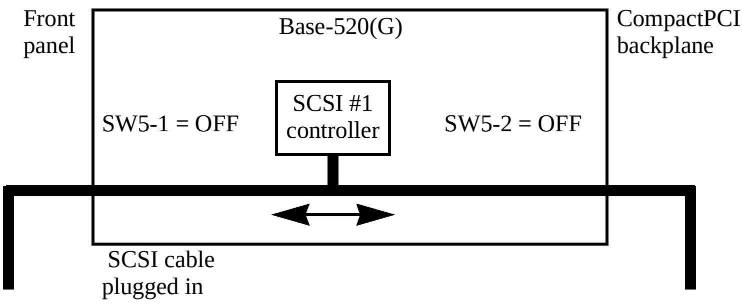
In this configuration (default switch setting):

– SW5-1 must be set to OFF = front panel termination automatic (automatic enabling or disabling of termination by sensing whether a SCSI cable is plugged in)

– and SW5-2 must be set to OFF = backplane termination disabled.

Default configuration 2 for 8 bit SCSI

• The default configuration 2 is also covered by the default switch setting: the Base-520(G) is not located at an endpoint of the SCSI #1 bus, the SCSI 1 bus is extended via the CompactPCI backplane and via the front-panel SCSI connector:



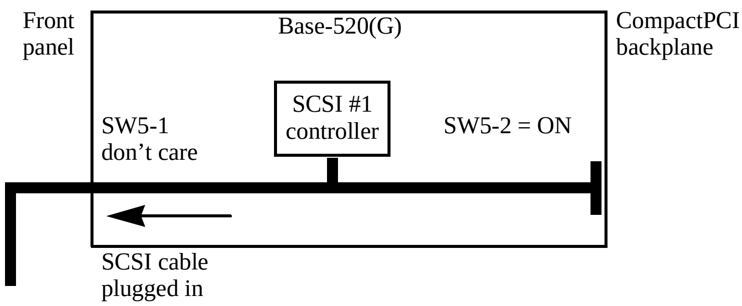
In this configuration (default switch setting):

– SW5-1 must be set to OFF = front panel termination automatic (automatic enabling or disabling of termination by sensing whether a SCSI cable is plugged in)

– and SW5-2 must be set to OFF = backplane termination disabled.

Alternative configuration for 8 bit SCSI

- Alternative configuration: the Base-520(G) is located at an endpoint of the SCSI #1 bus and the CompactPCI backplane is not used for SCSI #1 bus signalling, but the SCSI #1 bus is extended via the front panel connector:



In this configuration

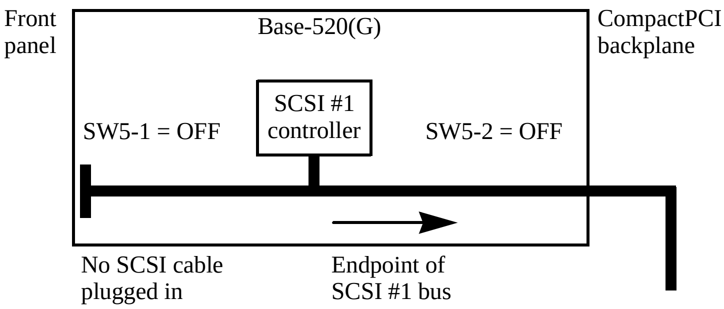
– both settings of SW5-1 are valid

– and SW5-2 must be set to ON = backplane termination enabled.

Default configuration for Wide SCSI

Wide SCSI is only available on the J5 connector of the CompactPCI backplane.

- The Wide SCSI configuration is covered by the default switch setting: The Base-520(G) is located at an endpoint of the SCSI #1 bus, the SCSI #1 bus is extended via the CompactPCI backplane, but no SCSI cable is plugged into the front-panel SCSI connector:



The Wide SCSI termination is always enabled and it is located near the SCSI #1 controller.

In this configuration (default switch setting):

– SW5-1 must be set to OFF = automatic enabling or disabling of termination by sensing whether a SCSI cable is plugged in,

– and SW5-2 must be set to OFF = backplane termination disabled.

4.7 Ethernet Address and Host ID

In order to see the Ethernet address and host ID, type the following command at the prompt:

```
ok banner
```

The information below explains how the SPARC/CPCI-52x(G) Ethernet address and the host ID are determined.

Figure 14 The 48-bit (6-byte) Ethernet address

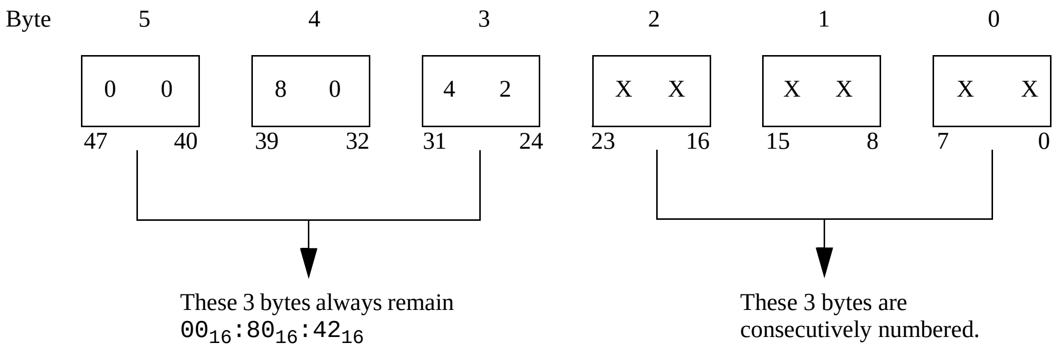
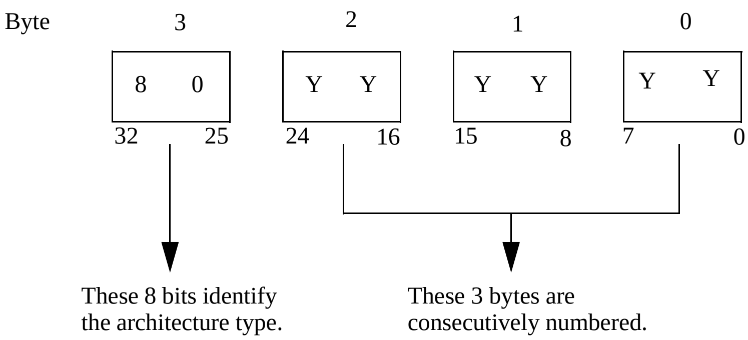


Figure 15 The 32-bit (4-byte) host ID



4.8 OpenBoot Firmware

This chapter describes the use of the OpenBoot firmware. The following tasks will be described in detail:

- Boot the system
- Run diagnostics
- Display system information
- Reset the system
- OpenBoot help

Note: **The examples in this section can differ from the appearance on your monitor according to your device tree (CPU architecture).**

For more information on the OpenBoot firmware see the *Open Boot 3.x Manual Set*.

The OpenBoot firmware is subject to changes. For newest version and how to upgrade refer to the SMART service accessible via the FORCE COMPUTERS World Wide Web site.

4.8.1 Boot the System

The most important function of OpenBoot firmware is the booting of the system. Booting is the process of loading and executing a stand-alone program such as the operating system. After it is powered on, the system usually boots automatically after it has passed the power-on self-test (POST). This occurs without user intervention.
If necessary, you can explicitly initiate the boot process from the Open-Boot command interpreter. Automatic booting uses the default boot device specified in nonvolatile RAM (NVRAM); user initiated booting uses either the default boot device or one specified by the user.
To boot the system from the default boot device, enter the following command at the Forth monitor prompt `ok`:

```
ok boot
```

The boot command has the following format:
`boot [device-specifier] [filename] [-bootoption]`

Optional Boot Parameters

***Note:* These options are specific to the operating system and may differ from system to system.**

[device-specifier] The name (full path or alias) of the boot device. Typical values are cdrom, disk, floppy, net, or tape.

[*filename*] The name of the program to be booted. *filename* is relative to the root of the selected device. If no filename is specified, the boot command uses the value of *boot-file* NVRAM parameter. The NVRAM parameters used for booting are described in the following section.

[-*bootoption*] Boot option may be one of the following:

[-a] -a prompt interactively for the device and name of the boot file.

[-h] -h halt after loading the program.

[-r] -r reconfigure Solaris device drivers after changing the hardware configuration.

[-v] -v print verbose information during boot procedure.

Devices to Boot from

To explicitly boot from the internal disk using the Forth monitor enter:

```
ok boot disk
```

To retrieve a list of all device alias definitions, type devalias at the Forth Monitor command prompt. The following table lists some typical device aliases:

Table 20 Device alias definitions

Alias	Description
	Defined for SCSI
`scsi`	SCSI
`disk`	Default disk SCSI-target-ID 0
`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)`	1st tape drive SCSI-target-ID 4
`tape1`	2nd tape drive SCSI-target-ID 5
`cdrom`	CD-ROM partition f, SCSI-target-ID 6
	Defined for Ethernet
`net`	Ethernet
`floppy`	Floppy disk
`audio`	Audio
`keyboard`	Keyboard
`mouse`	Mouse
`ebus`	EBus2
`pcia`	secondary PCI bus A
`pcib`	secondary PCI bus B
`pci`	primary PCI bus
`flash-prog`	Flash EPROM programming mode
`flash`	Flash EPROM
`ttya`	Serial interface A
`ttyb`	Serial interface B

4.8.2 NVRAM Boot Parameters

The OpenBoot firmware holds its configuration parameters in NVRAM. At the Forth monitor prompt enter `printenv` to see a list of all available configuration parameters.

***Note:* Per default the SPARC/CPCI-52x(G) boots the OS automatically. If not, ensure that the `auto-boot?` parameter is always set to `true`.**

To set parameters

The OpenBoot command `setenv` may be used to set specific parameters in the order below:
`setenv [`*`configuration_parameter`*`] [`*`value`*`]`

The configuration parameters in Table 2 are involved with the boot process.

Table 21 Setting configuration parameters

Parameter	Default value	Description
`auto-boot?`	`true`	If `true`, automatic booting after power on or reset
`boot-device`	`disk`	Device from which to boot
`boot-file`	`empty string`	File to boot
`diag-switch?`	`false`	If `true`, run in diagnostic mode
`diag-device`	`net`	Device from which to boot in diagnostic mode
`diag-file`	`empty string`	File to boot in diagnostic mode

When booting an operating system or another stand-alone program, and neither a boot device nor a filename is supplied, the `boot` command of the Forth monitor takes the omitted values from the NVRAM configuration parameters. If the parameter `diag-switch?` is `false`, `boot-device` and `boot-file` are used. Otherwise, the OpenBoot firmware uses `diag-device` and `diag-file` for booting.

4.8.3 Diagnostics

At Hardware Power On or Button Power On the OpenBoot firmware executes POST. The extent of certain tests executed within the POST depend on the state of the configuration parameter `diag-level.` The operator can choose between minimal or maximal testing by setting this configuration parameter to `min` or `max`. Furthermore an enhanced diagnostic menu is available if setting this parameter to `menu`. If the NVRAM configuration parameter `diag-switch?` is `true` 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 are displayed which indicate the problem. After POST the OpenBoot firmware boots an operating system or enters the Forth monitor, if the NVRAM configuration parameter `auto-boot?` is `false`.

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

The table below lists several diagnostic routines followed by examples for each of these routines:

Table 22 **Diagnostic routines**

Command	**Description**
`probe-scsi`	Identifies devices connected to the primary SCSI bus
`probe-scsi-all [`*`device-path`*`]`	Performs probe-SCSI on all SCSI buses installed in the system below the specified device tree node. If *`device-path`* is omitted, the root node is used.
`test` *`device-specifier`*	Executes the specified device's self-test method. *`device-specifier`* may be a device path name or a device alias. **Example:** • `test net` – test network connection
`test-all [`*`device-specifier`*`]`	Tests all devices that have a built-in self-test method and that reside below the specified device tree node. If *`device-path`* is omitted, the root node is used.
`watch-clock`	Monitors the clock function.
`watch-net-all`	Monitors network connection via all Ethernet interfaces installed in the system.
`watch-net`	Monitors network connection via primary Ethernet.

Examples:

SCSI bus

To check the SCSI #1 for connected devices enter:

```
ok probe-scsi
Target 3
Unit 0 Disk FUJITSU M2952ESP SUN2.1G2545
ok
```

All SCSI buses

To check all the SCSI buses installed in the system enter the following (The actual response depends on the devices on the SCSI buses).:

```
ok probe-scsi-all
/pci@1f,0/scsi@2

Target 6
Unit 0 Disk Removable Read Only Device SONY CD-ROM CDU-8012 3.1a

/pci@1f/pci@4,1/scsi@2

Target 3
Unit 0 Disk FUJITSU M2952ESP SUN2.1G2545
ok
```

Note: **The command `probe-scsi-all` can last up to 2 minutes without terminal message.**

Single device

To test a single installed device enter:

```
ok test device-specifier
```

This executes the `self-test` device method of the specified device node.
`device-specifier` may be a device path name or a device alias as described in Table 20, “Device alias definitions,” on page 45. The response depends on the self-test of the device node.

Group of devices

To test a group of installed devices enter:

```
ok test-all
```

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

Clock

To test the clock function enter:

```
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.
22
ok
```

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

Network

To monitor the network connection enter:

```
ok watch-net
Internal loopback test -- succeeded.
Transceiver check -- Using Onboard transceiver -- Link Up.
passed
Using Onboard transceiver -- Link Up.
Looking for Ethernet packets.
'.' is a good packet. 'X' is a bad packet.
Type any key to stop.
..........X..........................X.............
ok
```

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

4.8.4 Display System Information

The Forth monitor provides several commands to display system information. These commands let you display the system banner, the Ethernet address for the Ethernet controller, the contents of the ID PROM, and the version number of the OpenBoot firmware.
The ID PROM contains specific information to the individual machine, including the serial number, date of manufacture, and assigned Ethernet address.
The following table lists these commands:

Table 23 Commands to display system information

Command	**Description**
`banner`	Displays system banner
`show-pci-devs-all`	Displays list of installed and probed PCI Bus devices
`.enet-addr`	Displays the Ethernet address
`.idprom`	Displays ID PROM contents, formatted
`.traps`	Displays a list of SPARC trap types
`.version`	Displays version and date of the boot PROM
`show-devs`	Displays a list of all device tree nodes
`devalias`	Displays a list of all device aliases

4.8.5 Reset the System

If your system needs to be reset, you either press the reset button on the front panel or, if you are in the Forth Monitor, type **reset** on the command line.

```
ok reset
```

The system immediately begins executing the initialization procedures and executes the POST if having pressed the reset button. Then the system either boots automatically or enters the Forth Monitor, just as it would have done after a power-on cycle.

4.8.6 OpenBoot Help

The Forth Monitor contains an online help which can be activated by entering:

```
ok help
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:
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 words or subcategories just type **help [name]**.

- The online help shows you the Forth word, the parameter stack before and after execution of the Forth word (before -- after), and a short description.
- The online help of the Forth monitor is located in the boot PROM, that means that there is not an online help for all Forth words.

Example:

How to get help for special Forth words or subcategories:

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

```
ok help memory
dump ( addr length -- ) display memory at addr for length bytes
fill ( addr length byte -- ) fill memory starting at addr with byte
move ( src dest length -- ) copy length bytes from src to dest address
map? ( vaddr -- ) show memory map information for the virtual address
x? ( addr -- ) display the 64-bit number from location addr
l? ( addr -- ) display the 32-bit number from location addr
w? ( addr -- ) display the 16-bit number from location addr
c? ( addr -- ) display the 8-bit number from location addr
x@ ( addr -- n ) place on the stack the 64-bit data at location addr
l@ ( addr -- n ) place on the stack the 32-bit data at location addr
w@ ( addr -- n ) place on the stack the 16-bit data at location addr
c@ ( addr -- n ) place on the stack the 8-bit data at location addr
x! ( n addr -- ) store the 64-bit value n at location addr
l! ( n addr -- ) store the 32-bit value n at location addr
w! ( n addr -- ) store the 16-bit value n at location addr
c! ( n addr -- ) store the 8-bit value n at location addr
ok
```

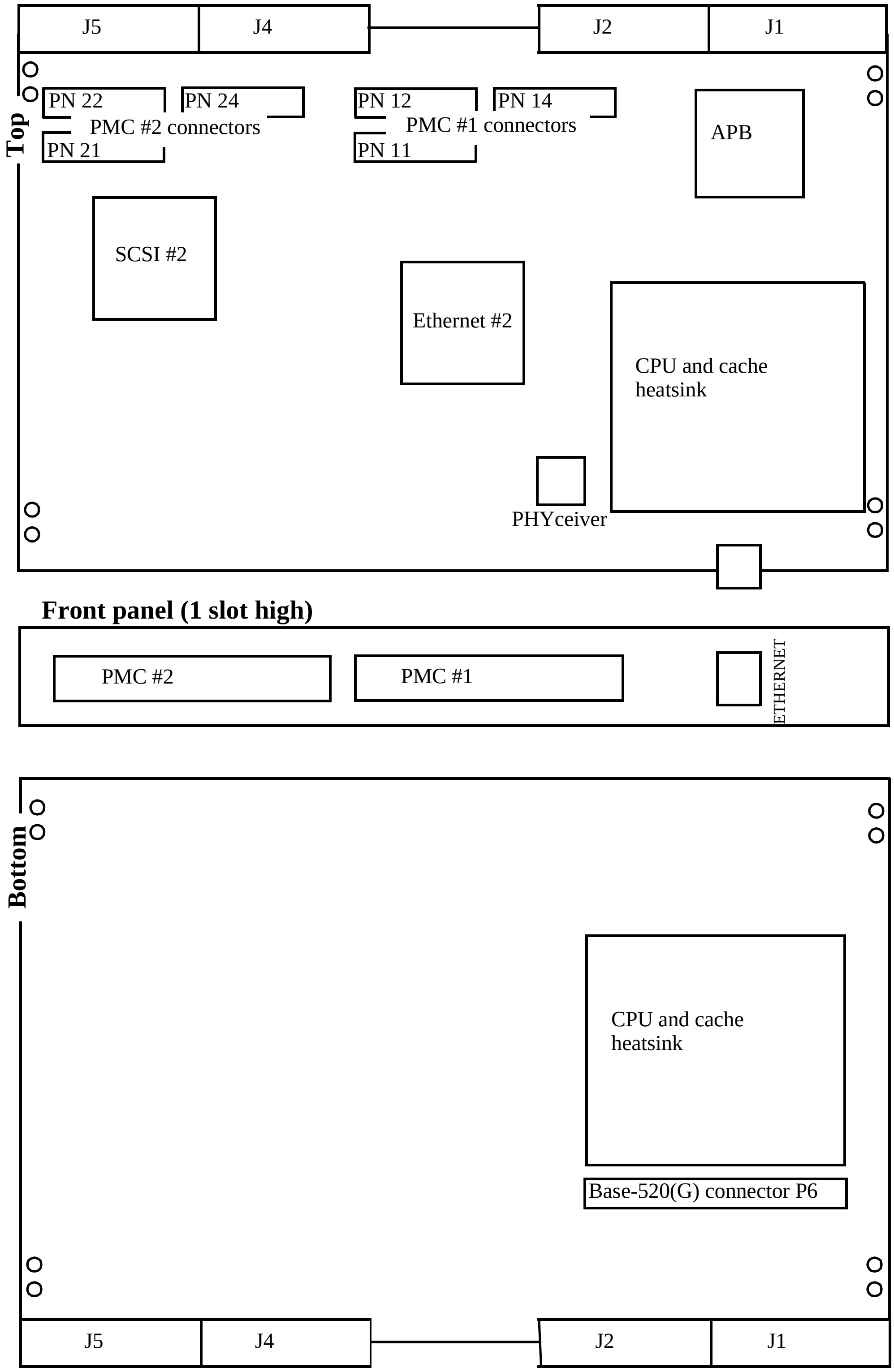
5 I/O-52x(G) Installation

5.1 Location Overview

The I/O-52x(G) contains the following main I/O interfaces:

- SCSI #2,
- Ethernet #2,
- PMC #1 and PMC #2.

Figure 16 Location diagram of the I/O-52x(G) (schematic)



5.2 Mechanical Constructions

The I/O-52x(G) is an extension to the Base-520(G). It occupies 1 CompactPCI slot and consists of the following major components:

- 2 PMC connectors,
- 1 SCSI #2 interface,
- and 1 Ethernet #2 interface.

The following figures show the SPARC/CPCI-52x(G) in 2-slot and 3-slot configurations.

Figure 17 Mechanical construction of the SPARC/CPCI-522

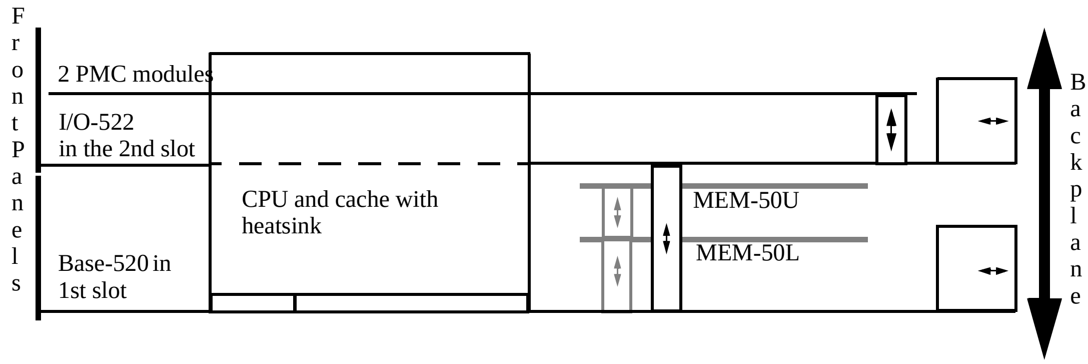


Figure 18 Mechanical construction of the SPARC/CPCI-52xG

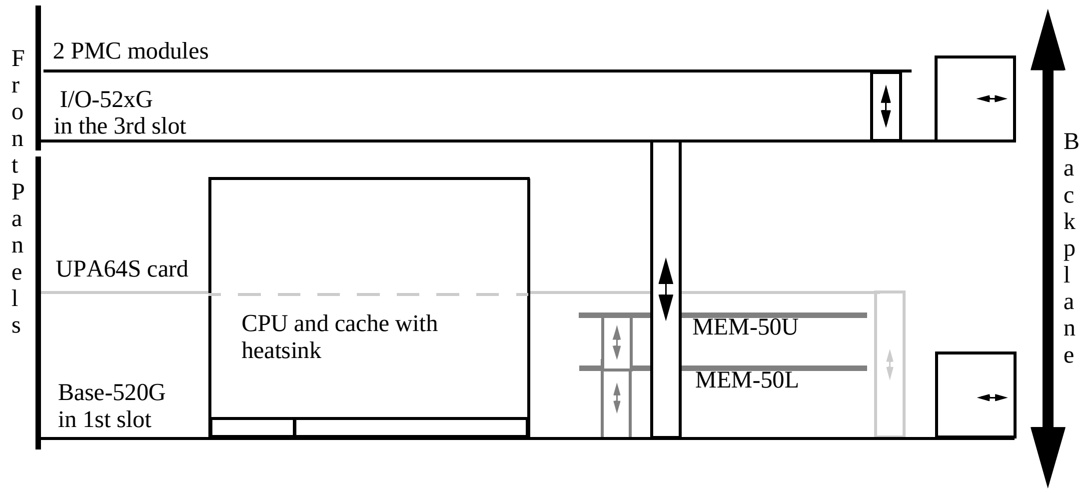
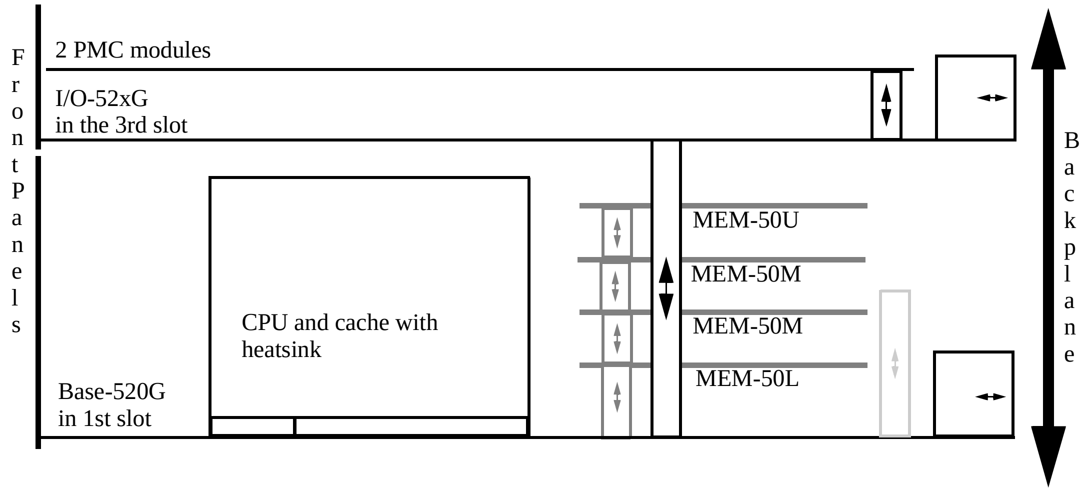


Figure 19 Mechanical construction of the SPARC/CPCI-52xG (option)



5.2.1 Installation/Deinstallation of the I/O-52x(G)

This section describes the installation and deinstallation procedure for the I/O-52x(G) with the mentioned location shown in the figure below.

Installation of the I/O-52x(G)

To install the I/O-52x(G) follow the steps below:

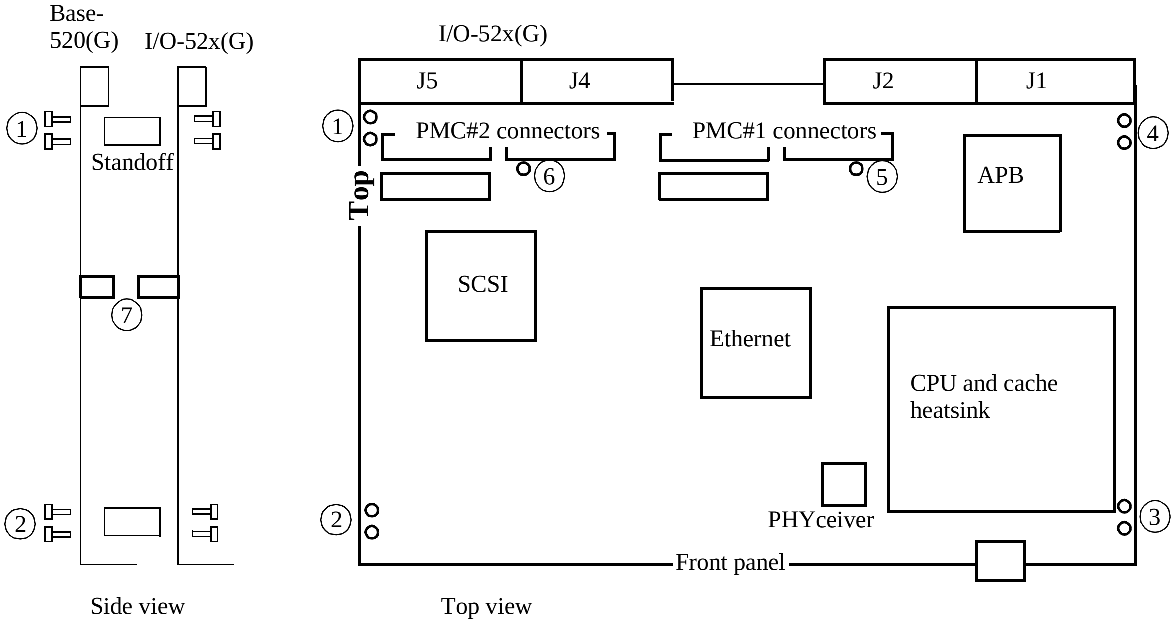
1. If there is an UPA64S card installed on your Base-520(G) ensure that you use the 2 z-standoffs delivered with the UPA64S card.
2. Remove the 10 screws at location 1...6 on the open end off the stand-offs of the I/O-52x(G).
3. Plug the I/O-52x(G) to the Base-520(G) via the I/O-52x(G) to Base-520(G) connector at position 7 and fix it with the 10 removed screws on the standoffs at location 1...6.

Deinstallation of the I/O-52x(G)

To deinstall the I/O-52x(G) follow the steps below:

1. Remove the 8 screws at location 1...4 on the bottom side of your Base-520(G).
2. Remove the 2 screws at location 5 and 6 on the top side of the I/O-52x(G).
3. Remove the I/O-52x(G) from the Base-520(G) by lifting it.
4. Fix the removed 10 screws on the open ends of the standoffs to have them available when installing the I/O-52x(G) again.

Figure 20 **Installation/Deinstallation of the I/O-52x(G)**



5.3 Powering Up

For powering up see the respective installation section of the Base-520(G).

5.4 Front Panel and Connectors

Front panel features

The features of the front panel are described in the following table. For a location diagram see figure 14 “Location diagram of the I/O-board (schematic)” on page 47.

Table 24 **Front panel features**

Device	**Description**
ETHERNET	Standard Twisted-Pair-Ethernet RJ45 connector for 10BaseT/100BaseTX Ethernet
PMC #1	Hole for the PMC #1 front panel
PMC #2	Hole for the PMC #2 front panel

On-board connectors

In addition to the front-panel connectors, the I/O-52x(G) provides on-board connectors for connection to the Base-520(G), to the CompactPCI bus and for 2 PMC modules. An overview is shown in the following table.

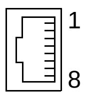
Table 25 **On-board connectors**

Connector description and location	**Connector type and sample manufacturer part number**
CompactPCI J1, J2, J4, J5	Standard CompactPCI metric, 5-row shielded connectors female
I/O-52x(G) extension connector P6	100-pin MBUS connector female low: for 2 slot solution high: for 3 slot solution
PMC #1 PN11, PN12, PN14	64-pin SMD connector
PMC #2 PN21, PN22, PN24	64-pin SMD connector

5.4.1 Ethernet #2 Interfaces

The full duplex 10BaseT/100BaseTx Ethernet #2 interface is available at the front panel via a Twisted-Pair-Ethernet connector.

Table 26 **Twisted-Pair-Ethernet connector pinout**

Connector	**Pin**	**Signal**
RJ-45 TPE 	1	TX+
	2	TX-
	3	RX+
	4	GND
	5	GND
	6	RX-
	7	GND
	8	GND

The Ethernet #2 interface is also accessible at the J5 back panel connector via an MII #2 interface. If Ethernet #2 gets accessed via I/O panel, the front panel connector is normally disabled automatically, for other configurations see the respective jumper settings in the *SPARC/IOBP-520*

Installation Guide. For the J5 connector pinout see figure 22 “CompactPCI J5 connector pinout” on page 60.

5.4.2 PMC Slots

The I/O-52x(G) provides 2 PMC slots compliant with IEEE P1386 ("Draft Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC"). The PCI bus, a high speed local bus, connects different high speed I/O cards with the SPARC/CPCI-52x(G). Both PMC slots support 32-bit data bus width with a maximum frequency of 33 MHz.

PMC Voltage Keys

The PCI bus uses a 5V voltage to signal bus levels. The voltage keys prevent 3.3V PMC cards from being plugged into the PMC slots.

Connector Configuration

The 32-bit PCI bus requires 2 PMC connectors. The 3rd PMC connector (PNx4) connects additional user I/O signals of PMC slot 1 and PMC slot 2 to the CompactPCI J4 connector.

PMC slot 1 connectors

- for the PCI bus: PN11 and PN12
- for 64 user I/O signals: PN14

PMC slot 2 connectors

- for the PCI bus: PN21 and PN22
- for 32 user I/O signals: PN24

5.4.3 CompactPCI Backplane Connector Pinout

J1 and J2

The J1 and J2 connectors implement the CompactPCI 64-bit connector pinout as specified by the CompactPCI specification. Therefore, this manual only documents the pinout of the J4 and J5 connector.

J4 and J5

Besides the CompactPCI specific pinout the following interfaces are available on the CompactPCI J4 and J5 connector.

- SCSI #2, MII #2
- User I/O pins for PMC #1 and #2 (`PMC #1, PMC #2`)

Figure 21 **CompactPCI J4 connector pinout**

Pin	A	B	C	D	E
1	PMC #1 I/O 61	PMC #1 I/O 62	PMC #1 I/O 63	PMC #1 I/O 64	n.c.
2	PMC #1 I/O 56	PMC #1 I/O 57	PMC #1 I/O 58	PMC #1 I/O 59	PMC #1 I/O 60
3	PMC #1 I/O 51	PMC #1 I/O 52	PMC #1 I/O 53	PMC #1 I/O 54	PMC #1 I/O 55
4	PMC #1 I/O 46	PMC #1 I/O 47	PMC #1 I/O 48	PMC #1 I/O 49	PMC #1 I/O 50
5	PMC #1 I/O 41	PMC #1 I/O 42	PMC #1 I/O 43	PMC #1 I/O 44	PMC #1 I/O 45
6	PMC #1 I/O 36	PMC #1 I/O 37	PMC #1 I/O 38	PMC #1 I/O 39	PMC #1 I/O 40
7	PMC #1 I/O 31	PMC #1 I/O 32	PMC #1 I/O 33	PMC #1 I/O 34	PMC #1 I/O 35
8	PMC #1 I/O 26	PMC #1 I/O 27	PMC #1 I/O 28	PMC #1 I/O 29	PMC #1 I/O 30
9	PMC #1 I/O 21	PMC #1 I/O 22	PMC #1 I/O 23	PMC #1 I/O 24	PMC #1 I/O 25
10	PMC #1 I/O 16	PMC #1 I/O 17	PMC #1 I/O 18	PMC #1 I/O 19	PMC #1 I/O 20
11	PMC #1 I/O 11	PMC #1 I/O 12	PMC #1 I/O 13	PMC #1 I/O 14	PMC #1 I/O 15
12	Coding key area				
13					
14					
15	PMC #1 I/O 6	PMC #1 I/O 7	PMC #1 I/O 8	PMC #1 I/O 9	PMC #1 I/O 10
16	PMC #1 I/O 1	PMC #1 I/O 2	PMC #1 I/O 3	PMC #1 I/O 4	PMC #1 I/O 5
17	PMC #2 I/O 61	PMC #2 I/O 62	PMC #2 I/O 63	PMC #2 I/O 64	VP5_IOBP
18	PMC #2 I/O 56	PMC #2 I/O 57	PMC #2 I/O 58	PMC #2 I/O 59	PMC #2 I/O 60
19	PMC #2 I/O 51	PMC #2 I/O 52	PMC #2 I/O 53	PMC #2 I/O 54	PMC #2 I/O 55
20	PMC #2 I/O 46	PMC #2 I/O 47	PMC #2 I/O 48	PMC #2 I/O 49	PMC #2 I/O 50
21	PMC #2 I/O 41	PMC #2 I/O 42	PMC #2 I/O 43	PMC #2 I/O 44	PMC #2 I/O 45
22	PMC #2 I/O 36	PMC #2 I/O 37	PMC #2 I/O 38	PMC #2 I/O 39	PMC #2 I/O 40
23	PMC #2 I/O 31	PMC #2 I/O 32	PMC #2 I/O 33	PMC #2 I/O 34	PMC #2 I/O 35
24	PMC #2 I/O 26	PMC #2 I/O 27	PMC #2 I/O 28	PMC #2 I/O 29	PMC #2 I/O 30
25	PMC #2 I/O 21	PMC #2 I/O 22	PMC #2 I/O 23	PMC #2 I/O 24	PMC #2 I/O 25

As factory option the PMC #1 I/O 1...32 signals can be connected to the PMC #2 I/O 33...64 signals.

Figure 22 **CompactPCI J5 connector pinout**

Pin	A	B	C	D	E
1	SCSI #2 D8	SCSI #2 D9	SCSI #2 D10	SCSI #2 D11	n.c.
2	SCSI #2 SEL	SCSI #2 CD	SCSI #2 REQ	SCSI #2 IO	WIDETERMPWR
3	SCSI #2 ATN	SCSI #2 BSY	SCSI #2 ACK	SCSI #2 RST	SCSI #2 MSG
4	SCSI #2 D4	SCSI #2 D5	SCSI #2 D6	SCSI #2 D7	TERMPWR
5	SCSI #2 D0	SCSI #2 D1	SCSI #2 D2	SCSI #2 D3	SCSI #2 DP0
6	SCSI #2 D12	SCSI #2 D13	SCSI #2 D14	SCSI #2 D15	SCSI #2 DP1
7	MII #2 RXD3	MII #2 RXD2	MII #2 RXD1	MII #2 RXD0	MII #2 RX_CLK
8	MII #2 RX_DV	MII #2 COL	MII #2 CRS	MII #2 RX_ER	MII #2 MGT_DIO
9	MII #2 TXD3	MII #2 TXD2	MII #2 TXD1	MII #2 TXD0	MII #2 TX_CLK
10	PMC #2 I/O 19	PMC #2 I/O 20	MII #2 TX_EN	MII #2 TX_ER	MII #2 MGT_CLK
11	PMC #2 I/O 14	PMC #2 I/O 15	PMC #2 I/O 16	PMC #2 I/O 17	PMC #2 I/O 18
12	PMC #2 I/O 9	PMC #2 I/O 10	PMC #2 I/O 11	PMC #2 I/O 12	PMC #2 I/O 13
13	PMC #2 I/O 5	PMC #2 I/O 6	PMC #2 I/O 7	PMC #2 I/O 8	VP5_IOBP
14	PMC #2 I/O 1	PMC #2 I/O 2	PMC #2 I/O 3	PMC #2 I/O 4	n.c.
15	VP5_IOBP	n.c.	n.c.	n.c.	n.c.
16	n.c.	n.c.	n.c.	n.c.	n.c.
17	n.c.	n.c.	n.c.	n.c.	n.c.
18	n.c.	n.c.	n.c.	n.c.	n.c.
19	n.c.	n.c.	n.c.	n.c.	n.c.
20	n.c.	n.c.	n.c.	n.c.	n.c.
21	n.c.	n.c.	n.c.	n.c.	n.c.
22	n.c.	n.c.	n.c.	n.c.	n.c.

I/O panel

As a separate price list item an I/O panel is available for the I/O-52x(G), the SPARC/IOBP-520/IO. An extended variant is the SPARC/CPCI-520/AccKit/IO which contains additionally to the I/O panel the following cables:

– a flat ribbon SCSI cable for the I/O panel

– and a Twisted-Pair-Ethernet cable for the front panel or the I/O panel.

The I/O panel supports the following interfaces:

- Fast/Wide SCSI #2,
- MII #2 Ethernet,
- and PMC user I/O.

Danger

The SPARC/IOBP-520/IO and the SPARC/CPCI-520/AccKit/IO is especially designed for the I/O-52x(G).
Do not use any other I/O panels on the I/O-52x(G). Use only the front panel or the backpanel Ethernet interface, not both. Check the configuration of your I/O panel.

5.5 SCSI #2 Configuration

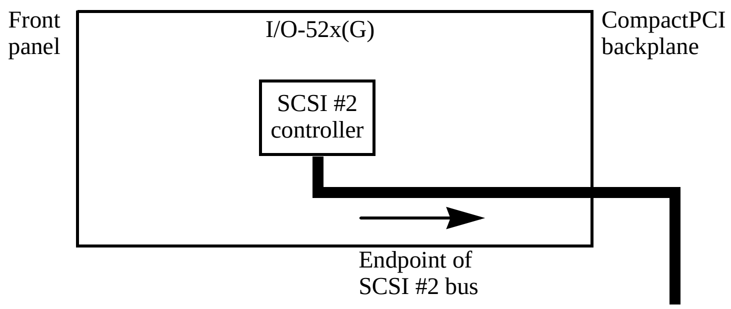
***Note:* Correct SCSI bus selection: The I/O-52x(G) provides a second SCSI bus, SCSI #2. Its configuration is described as follows.**

The I/O-52x(G)'s SCSI #2 bus is only available at the I/O-52x(G)'s CompactPCI J5 connector.

Valid configuration

There is only 1 valid I/O-52x(G) SCSI #2 bus configuration:

- The I/O-52x(G) is located at an endpoint of the SCSI #2 bus, the SCSI #2 bus is extended via the CompactPCI backplane:



The SCSI #2 bus is always terminated at the SCSI #2 controller.

5.6 Ethernet #2 Configuration

***Note:* Correct Ethernet selection: The I/O-52x(G) provides the following 2 Ethernet #2 interfaces:**

- via a TPE #2 interface connected to a front-panel RJ-45 connector
- or an MII #2 interface available at the CompactPCI J5 connector

Ethernet address and host ID

For the SPARC/CPCI-52x(G) exists only 1 ethernet address and host ID, see "Ethernet Address and Host ID" section of the Base-520(G)'s installation section. Therefore you can use the Ethernet #2 TPE or MII interface of the I/O-52x(G) only in a separate network according to Ethernet #1 TPE or MII of the Base-520(G).

5.7 OpenBoot Firmware Alias Definitions for I/O-52x(G)

This chapter describes additional features used with reference to the I/O-52x(G) enhancements.

Table 27 Device alias definitions

Alias	Description
	Defined for SCSI #2:
`scsi-2`	SCSI #2
`disk26`	disk SCSI #2-target-ID 6
`disk25`	disk SCSI #2-target-ID 5
`disk24`	disk SCSI #2-target-ID 4
`disk23`	disk SCSI #2-target-ID 3
`disk22`	disk SCSI #2-target-ID 2
`disk21`	disk SCSI #2-target-ID 1
`disk20`	disk SCSI #2-target-ID 0
`tape2 (or tape20)`	1st tape drive SCSI #2-target-ID 4
`tape21`	2nd tape drive SCSI #2-target-ID 5
`cdrom2`	CD-ROM partition f, SCSI #2-target-ID 6
	Defined for Ethernet #2:
`net`	Ethernet #2
`pcia-io`	secondary PCI bus A
`pcib-io`	secondary PCI bus B

6 Hardware Description

The SPARC/CPCI-52x(G) is a high performance CompactPCI board computer providing a CompactPCI system controller interface including DMA. It is based on

- the UltraSPARC-IIi processor supporting 3 high-speed interfaces concurrently operating:
  - the memory interface with ECC
  - the external (L2) cache interface
  - the 66 MHz PCI interface.
- and the APB (Advanced PCI Bridge) with interfaces to the Compact-PCI bus

The base board carries the components of the UltraSPARC-IIi chip set:

- the XCVR data multiplexers,
- the UIC (UPA Interrupt Concentrator) and
- the PCIO (PCI IO) chip which interfaces to the local I/O bus.

Described features of the Base-520(G)

The Base-520G is fully functional without the I/O-52x(G). Besides the CompactPCI interface the SPARC/CPCI-52x(G) provides the following components on the 2 on-board buses – the PCI bus and the EBus2:

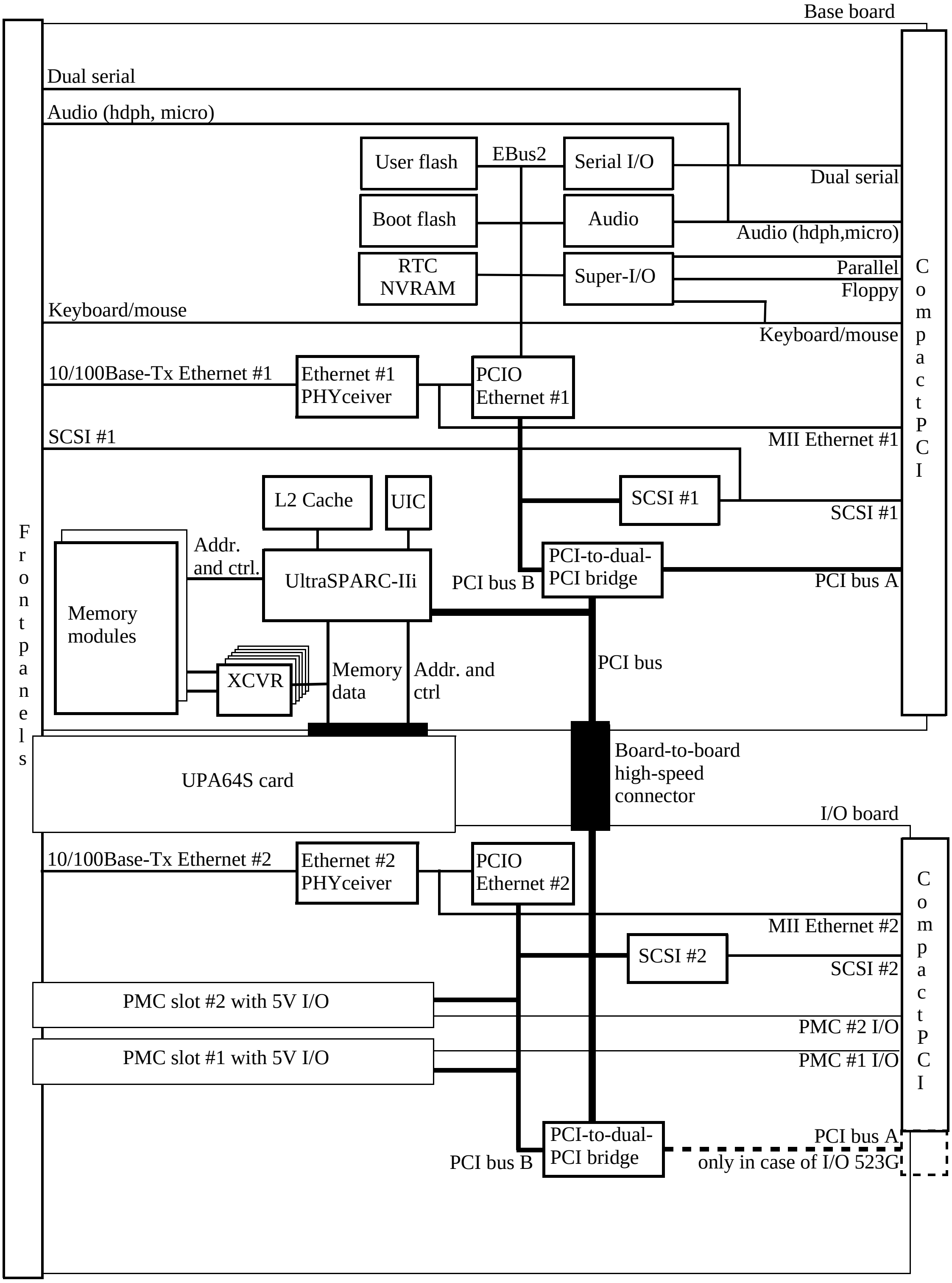
- Ethernet interface (Twisted-Pair and MII) Ethernet #1
- Ultra-wide SCSI interface SCSI #1
- Boot PROM, boot flash EPROM, and user flash EPROM
- 2 Serial Interfaces – SAB 82532
- Keyboard/Mouse, FDC and Parallel Interface – Super I/O
- RTC/NVRAM – M48T58
- 16-bit stereo audio interface
- System configuration registers
- 7- segments LED, rotary-switch, system and user LEDs

Described features of the I/O-52x(G)

The I/O-52x(G) adds

- 2 PMC Slots with Busmode Support,
- the second SCSI interface SCSI #2,
- and a second Ethernet interface Ethernet #2.

Figure 23 Block Diagram of the SPARC/CPCI-52x(G)



The following table gives an overview of the different buses, their bus modes, and the connected devices.

Table 28 Buses, bus modes, and connected devices

Bus and bus mode	Connected Devices
EC bus, big endian	• UltraSPARC-IIi (see page 68) • L2 cache with tag (see page 70)
Memory bus, big endian	• UltraSPARC-IIi (see page 68) • Memory banks (see page 70)
UPA bus, big endian	• UltraSPARC-IIi (see page 68) • UPA64S card
PCI bus, little endian	• UltraSPARC-IIi (see page 68) • APB on Base-520(G) and on I/O-52x(G) (see page 75)
PCI A bus on Base-520(G), little endian	• APB (see page 75) • CompactPCI interface (see page 75)
PCI bus B on Base-520(G), little endian	• APB (see page 75) • Ethernet controller with EBus2 interface (PCIO) (see page 75) • SCSI controller (see page 97)
EBus2, little endian	• Serial I/O interfaces (see page 80) • Keyboard/Mouse, parallel and floppy interface (see page 81) • Audio interface (see page 84) • RTC/NVRAM (see page 83) • Boot PROM, boot and user flash EPROM (see page 79) • System Configuration Registers (see page 85)
PCI bus A on I/O-52x(G), little endian	• APB (see page 75) • only I/O-523G: CompactPCI interface (see page 75)
PCI bus B on I/O-52x(G), little endian	• APB (see page 75) • Ethernet controller (PCIO) (see page 75) • SCSI controller (see page 97) • 2 PMC modules (see page 98)

6.1 Processor – UltraSPARC-IIi

UltraSPARC-IIi is a highly integrated 64-bit SPARC V9 superscalar processor. The interfaces have been optimized to typical uniprocessor system requirements.

Features

- Binary compatible with all SPARC application codes
- VIS instruction set
- 4-way SuperScalar design with 9 execution units (SPARC V9)
  - 4 integer execution units
  - 3 floating-point execution units
  - 2 graphics execution units
- Directly addresses little- or big-endian data
- 64-bit address pointers
- 16-KByte non-blocking data cache
- 16-KByte instruction cache
- Integrated L2 cache controller
- Integrated control of 400-MByte/s EDO DRAM memory subsystem
- 64-Byte block load and block store instructions
- Supports software data prefetch into L2 cache
- Supports up to 3 outstanding L2 cache misses
- Supports UPA64S interface
- High sustained PIO and DMA PCI I/O bandwidth
- Read prefetch and write gathering and posting
- PCI DMA is cache coherent
- Dedicated TLB provides mapping and protection

6.1.1 Physical Memory Map

Table 29 UltraSPARC-IIi physical address map (41-bit physical addresses)

Address range in PA<40:0>	Size	Addressed interface	Access type
$000.0000.0000_{16}$ …$000.3FFF.FFFF_{16}$	1 GByte	Main memory	cacheable
$000.4000.0000_{16}$ …$1FF.FFFF.FFFF_{16}$	reserved	undefined	cacheable
$000.0000.0000_{16}$ …$1FB.FFFF.FFFF_{16}$	reserved	undefined	noncacheable
$1FC.0000.0000_{16}$ …$1FD.FFFF.FFFF_{16}$	8 GByte	UPA64S	noncacheable
$1FE.0000.0000_{16}$ …$1FF.FFFF.FFFF_{16}$	8 GByte	PCI	noncacheable

Table 30 UltraSPARC-IIi internal CSR space (16 MByte)

Address range in PA<40:0>	Size	Description/owner
$1FE.0000.0000_{16}$ …$1FE.0000.01FF_{16}$	512 Byte	PBM (PCI bus module)
$1FE.0000.0200_{16}$ …$1FE.0000.03FF_{16}$	512 Byte	IOM (IO memory management unit)
$1FE.0000.0400_{16}$ …$1FE.0000.1FFF_{16}$	7 KByte	PIE (PCI interrupt)
$1FE.0000.2000_{16}$ …$1FE.0000.5FFF_{16}$	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	MCU (memory control unit)
$1FE.00FF.F020_{16}$	8 Byte	PIE
$1FE.00FF.F028_{16}$ …$1FE.00FF.FFFF_{16}$	4 KByte	MCU

6.1.2 External Cache Control Unit

The UltraSPARC-IIi obtains an integrated L2 cache controller providing a backside interface with a 72-bit wide data bus and 150 MHz speed. The L2 cache capacity is either 256 KByte or 1 MByte. 3 synchronous late write SRAM devices (2 for data and 1 for the tags) are provided.

6.1.3 Memory Controller Unit, Memory Modules, and Main Memory Configuration

Memory controller unit

The memory controller unit of the SPARC/CPCI-52x(G) is included in the UltraSPARC-IIi. The memory interface provides full EDO DRAM support including refresh. It uses 8 RAS lines to select 8 DRAM banks and 2 identical CAS signals. The memory interface is 72 bit wide, 64 bits are shared with the UPA64S interface and 8 bits are used for ECC. 6 bidirectional registered multiplexers and demultiplexers (XCVRs) are used to extend the memory interface from 72 bit to 144 bit. The control of the XCVRs is also included in the UltraSPARC-IIi. 60 ns EDO DRAMs with 10-bit column address (CAS) are supported.

The CPU supports the following accesses to main memory:

- Refresh: Refresh is a 4-way staggered CAS before RAS refresh. One refresh at a time refreshes the 2 memory banks of 1 memory module.
- 64-byte read to fill one L2 cache line: containing 1 burst access (EDO fast page mode) with 4 data (CAS) cycles each 128 bit wide (16 byte).
- 64-byte write to flush one L2 cache line: containing 1 burst access (EDO fast page mode) with 4 data (CAS) cycles each 128 bit wide (16 byte). To write data words smaller than 64 byte,
  - fill 1 cache line,
  - modify this cache line,
  - and write it back.

Memory modules

The main memory capacity is adjustable via installation of the appropriate memory modules. For naming conventions, see figure 6 “MEM-50 – memory module numbering scheme” on page 17.

Table 31 Relating memory capacity to device type and number of banks

Memory module	Memory capacity	Device type	No. of banks
MEM-50L	32 MByte	2M*8	1
	64 MByte		2
	128 MByte	8M*8	1
	256 MByte		2
MEM-50M or -50U	128 MByte		1
	256 MByte		2

Up to a total of 8 memory banks are possible. Every bank has a fixed physical starting address and an end address depending on the bank size.

Table 32 Physical memory addresses for memory modules

Physical address range	Module number	Bank	Size in MByte	Memory module MEM-…
$000.0000.0000_{16}$- $000.01FF.FFFF_{16}$	1	1	32	…50L/32 or …50L/64
$000.2000.0000_{16}$- $000.21FF.FFFF_{16}$		2	32	…50L/64
$000.0000.0000_{16}$- $000.07FF.FFFF_{16}$		1	128	…50L/128 or …50L/256
$000.2000.0000_{16}$- $000.27FF.FFFF_{16}$		2	128	…50L/256
$000.0800.0000_{16}$- $000.0FFF.FFFF_{16}$	2	1	128	…50M or U/128 or /256
$000.2800.0000_{16}$- $000.2FFF.FFFF_{16}$		2	128	…50M or U/256
$000.1000.0000_{16}$- $000.17FF.FFFF_{16}$	3	1	128	…50M or U/128 or /256
$000.3000.0000_{16}$- $000.37FF.FFFF_{16}$		2	128	…50M or U/256
$000.1800.0000_{16}$- $000.1FFF.FFFF_{16}$	4	1	128	…50M or U/128 or /256
$000.3800.0000_{16}$- $000.3FFF.FFFF_{16}$		2	128	…50M or U/256

6.1.4 Interrupt Map

Interrupt concept

The UltraSPARC-IIi provides a 6-bit wide interrupt vector for 63 interrupt sources. A separate device, the UPA interrupt concentrator (UIC), provides the inputs for all necessary interrupts. The UIC monitors all interrupts by a round-robin-scheme with 33 MHz, converts them to an own vector and transmits this vector to the processor. The processor interrupt PCI unit (PIE) reflects every vector in 1 state bit. From the state bit a new vector is generated and transmitted to the processor's execution unit. If more than 1 interrupt state bit is active, the transmitting sequence of the new interrupt vector is priority controlled.

Enabling interrupts

Every interrupt routed to the UIC can be separately enabled or disabled 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 33 Interrupt sources from the Base-520(G)

Function	Device	UIC vector	CPU internal vector	CPU internal priority
Ethernet #1	PCIO	21_{16}	21_{16}	3
SCSI #1	SYM53C875	20_{16}	20_{16}	3
Serial	SAB82532	$2d_{16}$	$2b_{16}$	7
Keyboard	Super I/O	$2b_{16}$	29_{16}	4
Mouse		$2c_{16}$	$2a_{16}$	4
Floppy		29_{16}	27_{16}	8
Parallel		22_{16}	22_{16}	2
UPA64 S	UPA64S card	23_{16}	software controlled	5
Audio playback	CS4231	$1f_{16}$	24_{16}	7
Audio capture		24_{16}	23_{16}	8
WatchDog	EBus2 CTRL	18_{16}	$0c_{16}$	6
Temperature		02_{16}	03_{16}	2
ENUM		$1a_{16}$	$0f_{16}$	1
CompactPCI #A	CompactPCI	$0f_{16}$	04_{16}	7
CompactPCI #B		$0d_{16}$	05_{16}	5
CompactPCI #C		$1d_{16}$	06_{16}	5
CompactPCI #D		$0a_{16}$	07_{16}	2

Table 34 Interrupt sources from the I/O-52x(G)

Function	Device	UIC vector	CPU internal vector	CPU internal priority
Ethernet #2	PCIO	15_{16}	02_{16}	5
SCSI #2	SYM53C875	05_{16}	01_{16}	5
PMC1 #A	PMC1	$0e_{16}$	14_{16}	6
PMC1 #B		$0c_{16}$	15_{16}	4
PMC1 #C		$0b_{16}$	16_{16}	3
PMC1 #D		09_{16}	17_{16}	1
PMC2 #A	PMC2	16_{16}	18_{16}	6
PMC2 #B		14_{16}	19_{16}	4
PMC2 #C		13_{16}	$1a_{16}$	3
PMC2 #D		11_{16}	$1b_{16}$	1
CompactPCI #A	CompactPCI – only for I/O-523G	06_{16}	10_{16}	6
CompactPCI #B		04_{16}	11_{16}	4
CompactPCI #C		03_{16}	12_{16}	3
CompactPCI #D		01_{16}	13_{16}	1

6.1.5 UltraSPARC-IIi PCI Bus Interface

The CPU uses a 66 MHz PCI bus as its bus for I/O extensions. This bus is 32 bits wide.

Table 35 UltraSPARC-IIi PCI address space (8 GByte)

Address range in PA<40:0>	Size	Description	Generated PCI commands
`1FE.0000.0000`$_{16}$ …`1FE.00FF.FFFF`$_{16}$	16 MByte	CPU internal CSR space	n.a.
`1FE.0100.0000`$_{16}$ …`1FE.01FF.FFFF`$_{16}$	16 MByte	PCI configuration space	Configuration read or write (may be special cycle)
`1FE.0200.0000`$_{16}$ …`1FE.02FF.FFFF`$_{16}$	16 MByte	PCI bus I/O space	I/O read or write
`1FE.0300.0000`$_{16}$ …`1FE.FFFF.FFFF`$_{16}$	4 GByte minus 48 MByte	reserved	reserved
`1FF.0000.0000`$_{16}$ …`1FF.FFFF.FFFF`$_{16}$	4 GByte	PCI bus memory space	Memory read or write

For a list of devices connected to the UltraSPARC-IIi PCI bus, see table 28 “Buses, bus modes, and connected devices” on page 67.

6.2 APB (Advanced PCI Bridge) and CompactPCI Interface

The APB (Advance PCI Bridge) is a PCI-to-dual-PCI bus bridge and is assembled on the base and on the I/O board (see figure 23 "Block Diagram of the SPARC/CPCI-52x(G)" on page 66). Both drive 2 secondary PCI buses A and B.

- The Base-520(G) APB supports the CompactPCI interface #1 on PCI bus A and the local PCI interface on PCI bus B. On PCI bus B there are additional local PCI devices providing:
  - SCSI #1 (see section 6.4 "SCSI Interface – SYM53C875" on page 97),
  - as well as Ethernet #1 and access to interfaces implemented by EBus2 devices (see section 6.3 "Ethernet and EBus2 Devices – PCIO" on page 75).
- The I/O-523G APB supports the CompactPCI interface #2 on PCI bus A. On the I/O-522(G) this bus is not used. On PCI bus B there are additional local PCI devices in both cases providing:
  - 2 PMC Slots with Busmode Support (see page 98),
  - SCSI #2 (see section 6.4 "SCSI Interface – SYM53C875" on page 97),
  - and Ethernet #2 (see section 6.3 "Ethernet and EBus2 Devices – PCIO" on page 75).

The backplane CompactPCI bus is connected via the PCI bus A of the APBs. A separate arbiter is used to support up to 7 CompactPCI I/O devices. A separate PLL is used to deliver 7 individual clocks for the CompactPCI I/O devices.

6.3 Ethernet and EBus2 Devices – PCIO

PCIO features

PCIO is a high integration, high performance single chip IO subsystem using a single PCI bus load. As a dual PCI device, PCIO delivers Ethernet and EBus2 functionality to the SPARC/CPCI-52x(G) (see section 6.3.1 "Ethernet Interface – PCIO" on page 76 and section 6.3.2 "EBus2 Interface – PCIO" on page 77).

- PCI Local Bus specification 2.1 compliant master/slave interface
- 10/100BaseT Ethernet using a derivative of Media Access Control (MAC), with fully buffered transmit and receive channels; media independent interface (MII)

- Expansion bus 2 interface (EBus2), supporting up to 8 external devices and 4 buffered slave DMA channels.

6.3.1 Ethernet Interface – PCIO

The PCIO on the Base-520(G) delivers the Ethernet #1 interface and the PCIO on the I/O-52x(G) delivers the Ethernet #2 interface. As described above, the PCIO provides Ethernet via a media independent interface (MII). An additional on-board PHYceiver transforms the MII into a 10/100-BaseT interface.

The Ethernet interface consists of 2 major function blocks:

- High performance two-channel DVMA host interface between the MAC and the PCI bus with interrupt generation capability
- Media Access Control (MAC) function for a 10/100 Mbit/s CSMA/CD protocol based network compatible with IEEE 802.3/Ethernet

DVMA

The PCIO DVMA controller enables the Ethernet interface to transfer data to and from the main memory. PCIO supports full duplex operation and provides 2 KByte local on-chip buffers (FIFOs) in each direction.

On-board PHYceiver

The Twisted Pair Ethernet interface is realized via a PHYceiver device, the PHYceiver - ICS1890. It is directly connected to the MII interface of the PCIO. The PHYceiver is a fully integrated physical layer device supporting 10 and 100 Mbit/s CSMA/CD Ethernet applications. The PHYceiver is compliant with ISO/IEC 8802-3 Ethernet standard for 10- and 100-Mbit/s operation. A station management interface (MII management interface) is provided to enable command and status information exchange between PCIO and PHYceiver. The PHYceiver supports shielded twisted pair (STP) and unshielded twisted pair (UTP) category 5 cables up to 105 m. Operation in half duplex or full duplex mode at either 10 or 100 Mbit/s is possible with control by auto-negotiation or manual selection. By employing auto-negotiation the technology capabilities of the remote link partner may be determined and operation automatically adjusted to the highest performance operating mode common to both. The on-board PHYceiver address is hardwired to 01_{16} as defined by the MII management interface IEEE specification.

Ethernet interrupt

The Ethernet controller uses the Ethernet interrupt on the UIC for interrupting the UltraSPARC-IIi (see table 33 “Interrupt sources from the Base-520(G)” on page 72 and table 34 “Interrupt sources from the I/O-52x(G)” on page 73).

6.3.2 EBus2 Interface – PCIO

The PCIO also provides the interface to the EBus2. EBus2 is a generic slave 8-bit wide DMA bus (pseudo ISA bus) to which off-the-shelf peripherals can be connected.

Base addresses of PCIO chip select signals

The base addresses of all 8 PCIO chip select signals in the 4 GByte PCI address space is determined by the following 2 registers of PCIO's configuration address space:

Table 36 PCIO EBus2 base address registers

PCIO configuration space address	Size	Description	Reset Value
010_{16}	32	EBus2 base address register 0: base address for EB_CS#0 (16 MBytes)	$F000.0000_{16}$
014_{16}	32	EBus2 base address register 1: base address for EBus_CS#1…EBus_CS#7 (each 1 MByte)	$F100.0000_{16}$

PCIO chip select signals

The PCIO PCI-to-EBus2 controller delivers 8 decoded chip select signals:

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

It thereby supports up to 8 single- or multi-function Intel-style 8-bit devices with a minimum of glue logic. The resulting memory map in the PCI address space (with the base address registers in the reset value configuration) is described in the table below.
After power up, the Base-520(G) PCIO is in boot mode and uses EBus_CS#0 for the initial code fetch (OpenBoot). Therefore, the boot PROM is hardwired to EBus_CS#0.

Table 37 EBus2 memory map in the PCI bus 4 GByte address space

PCI addr. range	Description	EBus_ CS#
$F000.0000_{16}$ …$F01F.FFFF_{16}$	Boot PROM or boot flash EPROM, see "Boot PROM, Boot and User Flash EPROM" on page 79	0
$F020.0000_{16}$ …$F0FF.FFFF_{16}$	User flash, see "Boot PROM, Boot and User Flash EPROM" on page 79	0
$F100.0000_{16}$ …$F10F.FFFF_{16}$	"RTC/NVRAM – M48T58" on page 83	1

Table 37 EBus2 memory map in the PCI bus 4 GByte address space (cont.)

PCI addr. range	Description	EBus_ CS#
`F110.0000`$_{16}$ …`F11F.FFFF`$_{16}$	reserved	2
`F120.0000`$_{16}$ …`F12F.FFFF`$_{16}$	"Audio Interface – CS4231A" on page 84	3
`F130.0000`$_{16}$ …`F13F.FFFF`$_{16}$	"Keyboard/Mouse, FDC and Parallel Interface – Super I/O" on page 81	4
`F140.0000`$_{16}$ …`F14F.FFFF`$_{16}$	"Serial Interfaces – SAB 82532" on page 80	5
`F150.0000`$_{16}$ …`F15F.FFFF`$_{16}$	reserved	6
`F160.0000`$_{16}$ …`F16F.FFFF`$_{16}$	"System Configuration Registers – SCR" on page 85	7
`F170.0000`$_{16}$ …`F17F.FFFF`$_{16}$	EBus2 controller configuration registers	n.a.

PCIO EBus2 DMA channels

Additionally, 4 DMA channels for floppy, parallel interface, audio in, and audio out are provided by the EBus2 interface. Each of the 4 DMA channels supports 128-byte deep FIFOs for data stream buffering.

Table 38 PCIO EBus2 DMA channels

PCIO EBus2 DMA channel no.	Associated device
0	Parallel interface (see page 81)
1	Audio playback (out) (see page 84)
2	Audio capture (in) (see page 84)
3	Floppy disk controller (see page 81)

6.3.3 Boot PROM, Boot and User Flash EPROM

The PCIO’s 16-MByte chip select signal `EBus_CS#0` is decoded to 3 chip select signals for

- 1 boot PROM device or 1 boot flash EPROM device (2 MByte address space).
- and up to 2 user flash EPROM devices (remaining 14 MBytes address space).

For the base address of `EBus_CS#0` see section 6.3.2 “EBus2 Interface – PCIO” on page 77.

Flash decoding

The Boot and User Flash Size Control Register indicates the `EBus_CS#0` decoding according to the assembled flash devices (see section 6.3.10 “SCR: Boot and User Flash” on page 89). The decoding is shown in the table below.

Table 39 Boot and user flash address space configuration

PCI addr. range	Configuration	Device type
F000.0000$_{16}$ …F00F.FFFF$_{16}$	Default config. with SW6-2 = OFF	1 MByte boot PROM one 27C080, 1Mbx8 read-only device
F020.0000$_{16}$ …F03F.FFFF$_{16}$		2 MByte user flash EPROM one 29F016, 2Mbx8, 5V write-protectable via SW4-4
F040.0000$_{16}$ …F0FF.FFFF$_{16}$		12 MByte unused
F000.0000$_{16}$ …F01F.FFFF$_{16}$	Alternative config. with SW6-2 = ON	2 MByte boot flash EPROM one 29F016, 2Mbx8, 5V write-protectable via SW4-3
F020.0000$_{16}$ …F03F.FFFF$_{16}$		2 MByte user flash EPROM one 29F016, 2Mbx8, 5V write-protectable via SW4-4
F040.0000$_{16}$ …F0FF.FFFF$_{16}$		12 MByte unused

Table 39 Boot and user flash address space configuration

PCI addr. range	Configuration	Device type
F000.0000$_{16}$ …F00F.FFFF$_{16}$	Default config. with SW6-2 = OFF in case of 4-MByte user flash fact. opt.	1 MByte boot PROM one 27C080, 1Mbx8 read-only device
F020.0000$_{16}$ …F05F.FFFF$_{16}$		4 MByte user flash EPROM two 29F016, 2Mbx8, 5V write-protectable via SW4-4
F060.0000$_{16}$ …F0FF.FFFF$_{16}$		10 MByte unused
F000.0000$_{16}$ …F01F.FFFF$_{16}$	Alternative config. with SW6-2 = ON in case of 4-MByte user flash fact. opt.	2 MByte boot flash EPROM one 29F016, 2Mbx8, 5V write-protectable via SW4-3
F020.0000$_{16}$ …F05F.FFFF$_{16}$		4 MByte user flash EPROM two 29F016, 2Mbx8, 5V write-protectable via SW4-4
F060.0000$_{16}$ …F0FF.FFFF$_{16}$		10 MByte unused

Programming the boot or user flash EPROM

Boot flash EPROM and user flash EPROM are on-board programmable if the switch-selectable hardware write protection is disabled (**see section 4.4 “Switch Settings” on page 27**).

Caution

Before programming the boot flash EPROM on-board, save the area containing the OpenBoot image for reprogramming purposes. For example, damage to the image in the boot flash EPROM can occur, if power fails during on-board reprogramming.

To reprogram the boot flash EPROM, see also section 7.3.1 “Copying the OpenBoot Image from Boot PROM to Boot Flash EPROM” on page 116.

6.3.4 Serial Interfaces – SAB 82532

The Base-520(G) provides 2 independent full-duplex serial I/O interfaces (A and B). They are implemented via the Enhanced Serial Communication Controller – SAB 82532 User’s Manual and Addendum at PCI bus base address F140.000016 on the EBus2.

Device features

- 2 independent full-duplex serial channels
- 2 independent baud rate generators
- Hardware handshake support

- Protocol support (HDLC/SDLC)
- Interrupt controlled

6.3.5 Keyboard/Mouse, FDC and Parallel Interface – Super I/O

To implement a major part of the UltraSPARC-IIi architecture's I/O-subsystem a standard PC Super I/O device is utilized, the Super I/O – PC87332VLJ at PCI bus base address `F130.000016` on the EBus2.

Device features

The Super I/O is a single chip solution for most commonly used I/O peripherals in ISA based computers. It incorporates

- a floppy disk controller (FDC, see "Floppy interface" on page 81),
- 2 UARTs which are fully NS16450 and NS16550 compatible and which are used for the SUN style keyboard/mouse interface,
- and an IEEE1284 compatible parallel interface with EPP (Enhanced Parallel Port) and ECP (Enhanced Capabilities Port) compatibility (see "Parallel interface" on page 82).

Standard PC-AT address decoding for all the on-chip peripherals and a set of configuration registers are also implemented together with advanced power management features.

Floppy interface

The floppy disk controller uses a high performance digital data separator, eliminating the need for any external filter components. One of the 4 DMA channels of the PCIO EBus2 is used for the Super-I/O floppy interface (see table 38 "PCIO EBus2 DMA channels" on page 78).

- Software compatible with the PC8477
- Superset of DP8473, the 765A and the N82077
- 16-byte FIFO (disabled by default)
- Burst and non-burst modes
- Perpendicular recording drive support
- High-performance internal digital data separator (no external filter components required)
- Low-power CMOS with enhanced power-down mode
- Automatic media-sense support
- Support of all popular 5.25" and 3.5" floppy drives, including the 2.88 MByte 3.5" floppy drive
- Support of fast 2 Mbps and standard 1 Mbps/500 Kbps/250 Kbps tape drives

Parallel interface

The parallel interface is Centronics compatible. One of the 4 DMA channels of the PCIO EBus2 is used for the Super-I/O parallel interface (see table 38 “PCIO EBus2 DMA channels” on page 78).

- Uni- or bidirectional parallel interface
- Centronics compliant and operation in either programmed I/O or DMA mode (software or hardware control).
- EPP (Enhanced Parallel Port) and ECP (Enhanced Capabilities Port) compatibility
- Includes protection circuit to prevent damage to the parallel interface when a connected printer is powered up or operated at a higher voltage

Control of Super I/O power-down mode

For information on controlling the Super I/O power-down mode see “SUPIO_PWDN (r/w)” on page 93.

6.3.6 RTC/NVRAM – M48T58

The Base-520(G) provides a RTC/NVRAM – M48T58 at PCI bus base address `F100.000016` on the EBus2.

Table 40 Address map of the RTC/NVRAM

Address offset range	Access
0000_{16} ...$1FF7_{16}$	NVRAM with 8 KByte minus 8 bytes capacity
$1FF8_{16}$ ...$1FFF_{16}$	RTC registers with clock information in 24-hour BCD format: year, month, date, day, hour, minute, second

Device features

- 8 KByte ultra low power CMOS SRAM
- Byte-wide accessible real-time clock and power-fail control circuit for automatic power-fail chip deselect and write protection
- Long-life lithium carbon monofluoride battery
- Year-2000 compliant RTC with own crystal

6.3.7 Audio Interface – CS4231A

The Base-520(G) provides an Audio Controller – CS4231A at PCI bus base address `F120.000016` on the EBus2. 2 DMA channels of the PCIO EBus2 are used for the audio interface (see table 38 “PCIO EBus2 DMA channels” on page 78): one for capture (Micro In, Line/Aux In) and one for playback (Line/Headphone Out).

Device features

- 16-bit stereo audio converters and complete on-chip filtering for record and playback of 16-bit audio data
- Windows sound system compatible
- ADPCM compression and decompression
- Extensive software support
- MPC level 2 compatible mixer
- Dual DMA registers support full duplex operation for capture and playback
- On-chip FIFOs for higher performance
- Included analog mixing and programmable gain and attenuation.

6.3.8 System Configuration Registers – SCR

The Base-520(G) implements a set of system configuration registers via a field programmable gate array XC4003E (FPGA Xilinx LCA) at PCI bus base address F160.000016 on the EBus2. The table below gives an overview of all SPARC/CPCI-52x(G) system configuration registers:

Table 41 System configuration register set (SCR), all 8-bit wide

PCI bus addr.	Reset value	Description
F160.000016 (LED 1) and F160.000116 (LED 2)	F0$_{16}$	User LED x Control Registers, x = 1, 2 (see page 86)
F160.000216	F3$_{16}$	Boot and User Flash Size Control Register (see page 90)
F160.000316	F0$_{16}$	I2C Bus Control and Status Register (see page 96)
F160.000416	F0$_{16}$	Miscellaneous Control Register (see page 93)
F160.000516	F0$_{16}$	Miscellaneous Control and Status Register (see page 91)
F160.000616	F0$_{16}$	Watchdog and Temperature Control and Status Register (see page 91)
F160.000716	F0$_{16}$	Watchdog Timer Trigger Register (see page 91)
F160.0008$_{16}$	F0$_{16}$	reserved
F160.0009$_{16}$		
F160.000A16	F0$_{16}$	ENUM Interrupt Control Register (see page 94)
F160.000B$_{16}$	FF$_{16}$	reserved
F160.000C$_{16}$		
F160.000D$_{16}$		
F160.000E16	FX$_{16}$	Reset Status Register (see page 92)
F160.000F16	FX$_{16}$	System Configuration Identification Register (see page 86)
F160.001016	00$_{16}$	7-Segment LED Display Control Register (see page 87)
F160.001116	XX$_{16}$	Rotary Switch Status Register (see page 87)
F160.001216	XX$_{16}$	SW4 and SW5 Status Register (see page 89)
F160.0013$_{16}$	XF$_{16}$	reserved

Table 42 System Configuration Identification Register

$F160.000F_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	ID			

ID (ro) ID indicates the hardware ID of the device containing the system configuration registers.

6.3.9 SCR: Front Panel and Switches

The following registers control front-panel or switch related features:

- “User LED x Control Registers, x = 1, 2” on page 86
- “Rotary Switch Status Register” on page 87
- “7-Segment LED Display Control Register” on page 87
- “SW4 and SW5 Status Register” on page 89

Table 43 User LED *x* Control Registers, *x* = 1, 2

$F160.0000_{16}$ (LED 1) and $F160.0001_{16}$ (LED 2)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	BLINK_FREQ		COLOR	

BLINK_FREQ (r/w) BLINK_FREQ specifies the blink frequency:

- = 00_2 no blinking
- = 01_2 blinking at appr. 0.5 Hz (slow)
- = 10_2 blinking at appr. 1 Hz (moderate)
- = 11_2 blinking at appr. 2 Hz (fast)

COLOR (r/w) COLOR specifies the status and color of the LED:

- = 00_2 off
- = 01_2 green
- = 10_2 red
- = 11_2 yellow

Table 44 7-Segment LED Display Control Register

F160.0010$_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	SEG DP	SEG G	SEG F	SEG E	SEG D	SEG C	SEG B	SEG A

DP and SEG_G … SEG_A (w) — DP and SEG_G … SEG_A control the status of the decimal point (DP) and the segments (SEG_G...SEG_A) in the hexadecimal display (see figure below for naming conventions).

= 0 The respective part of the display is turned OFF.

= 1 The respective part of the display is turned ON.

Figure 24 Naming the parts of the 7-segment LED display

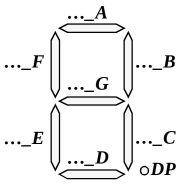


Table 45 Rotary Switch Status Register

F160.0011$_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	ROTARY_SW			

ROTARY_SW (ro) — ROTARY_SW indicates the current state of the rotary switch:

= 0000_2 Rotary switch set to 0_{16} (= 0000_2)

= 0001_2 Rotary switch set to 1_{16} (= 0001_2)

= 0010_2 Rotary switch set to 2_{16} (= 0010_2)

… …

= 1110_2 Rotary switch set to E_{16} (= 1110_2)

= 1111_2 Rotary switch set to F_{16} (= 1111_2)

Table 46 SW4 and SW5 Status Register

F160.0012$_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	SW5-4	SW6-2	SW5-3	SW5-2	SW5-1	SW4-4	SW4-3

SWx-y (ro), except SW5-2

SWx-y indicates the setting of the respective switch (see table 11 “Default switch settings” on page 27):

- SW5-4 – Reserved, must be OFF (see page 28)
- SW6-2 – Boot device selection (OFF = boot from boot PROM, see page 28)
- SW5-3 – Reserved, must be OFF (see page 28)
- SW5-1 – SCSI Termination for SCSI #1 on front panel (OFF = front panel termination automatic, see page 28)
- SW4-4 – User flash EPROM write protection (OFF = user flash EPROM write protected, see page 27)
- SW4-3 – Boot flash EPROM write protection (OFF = boot flash EPROM write protected, see page 27)

= 0 Switch is ON.

= 1 Switch is OFF.

SW5-2 (ro)

SW5-2 indicates the setting of the SW5-2: SCSI Termination for SCSI #1 on P2 (OFF = backplane termination disabled, see page 28).

= 0 Switch is OFF.

= 1 Switch is ON.

6.3.10 SCR: Boot and User Flash

The Boot and User Flash Size Control Register within the system configuration register set indicates the decoding of EBus_CS#0 according to the assembled flash device type (see section 6.3.3 “Boot PROM, Boot and User Flash EPROM” on page 79).

Note: **OpenBoot initializes the Boot and User Flash Size Control Register during power up with the correct value. Note also that BOOTROM_SIZE only concerns the boot flash EPROM, whereas the boot PROM is always decoded to 1 MByte. Therefore never reprogram this register.**

Table 47 Boot and User Flash Size Control Register

$F160.0002_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	USERROM_SIZE		BOOTROM_SIZE	

*x*ROM_SIZE (ro) — USERROM_SIZE and BOOTROM_SIZE indicate the decoding of the user and boot flash EPROM, respectively.

= 00_2, 01_2	reserved
= 10_2	Offset range 00.0000_{16} … $0F.FFFF_{16}$
= 11_2 (default)	Offset range 00.0000_{16} … $1F.FFFF_{16}$

Boot device selection — For information on selecting the boot device, see “SW_PLCC_TSOP (r/w)” on page 91.

6.3.11 SCR: Watchdog, Temperature Sensors, and Reset

Watchdog — An on-board watchdog can be enabled by SW6-4 (ON = enabled, see page 28). To start the watchdog timer after enabling it via SW6-4, it is necessary to trigger WDI in the Watchdog Timer Trigger Register once. The watchdog monitors the processor activity. When the watchdog timer interval expires, i.e. the watchdog timer is no longer triggered periodically, the watchdog timer activates its WDO output and a watchdog timer interrupt can be generated if IE_WDT in the Watchdog and Temperature Control and Status Register is set appropriately. The generation of an interrupt is indicated by the Reset Status Register.

Temperature sensors — The 2 temperature sensors connected to the I^2C Bus can be programmed to generate a temperature control interrupt (see also section 6.3.13 “SCR: I2C-Bus” on page 95).
The temperature sensors may be programmed in such a way that either the O.S. output signal is operating in the comparator mode or the interrupt mode. The state of the O.S. output signal is indicated by the IS_TEMPn (ro) bits in the Miscellaneous Control and Status Register.

- In the comparator mode O.S.
  - is cleared (0) when the current temperature exceeds an upper temperature limit T_{OS}.
  - is set (1) only when the current temperature falls below a lower limit T_{HYST}.
- In the interrupt mode O.S.

– is cleared (0) whenever the current temperature exceeds an upper temperature limit (T_{OS}) or falls below a lower limit (T_{HYST}).

– is set (1) only upon reading one of the temperature sensor's internal registers via the I^2C-Bus.

An interrupt is generated if O.S. is cleared. The generation of an interrupt is controlled and indicated by the Watchdog and Temperature Control and Status Register. OpenBoot initialises the SPARC/CPCI-52x(G) temperature sensors for interrupt mode operation.

Table 48 **Miscellaneous Control and Status Register**

$\mathtt{F160.0005}_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	IS_TEMP2	IS_TEMP1	0	SW_PLCC_TSOP

IS_TEMP*n* (ro) IS_TEMP1 and IS_TEMP2 indicate the state of the O.S. output signal of the respective temperature sensor (#1 or #2).

= 0 (default) O.S. output signal is low (0).

= 1 O.S. output signal is high (1).

SW_PLCC_TSOP (r/w) SW_PLCC_TSOP controls the selection of the boot device: the boot PROM or the boot flash EPROM. After reset this bit is cleared (0).

= 0 (default) The boot device specified by SW6-2 is selected.

= 1 The boot flash EPROM is selected.

Table 49 **Watchdog Timer Trigger Register**

$\mathtt{F160.0007}_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	WDI	1	1	1

WDI (r/w) WDI is used to trigger the watchdog timer by changing the value of WDI. Triggering the watchdog timer clears the watchdog timer. Default is 0.

Table 50 **Watchdog and Temperature Control and Status Register**

$\mathtt{F160.0006}_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	IP_TEMP	IE_TEMP	IS_WDT	IE_WDT

IP_TEMP (ro) IP_TEMP indicates whether one of the 2 temperature sensors signals an alarm condition. Only if enabled via appropriate setting of IE_TEMP, an interrupt is generated.

= 0 No temperature control interrupt is pending.

= 1 A temperature control interrupt is pending.

IE_TEMP (r/w) IE_TEMP specifies whether the generation of temperature control interrupts is enabled or disabled. A pending temperature control interrupt is indicated by IP_TEMP.

= 0 Interrupt generation disabled (default after reset).

= 1 Interrupt generation enabled.

IE_WDT (r/w) IE_WDT specifies whether the generation of watchdog timer interrupts is enabled or disabled. A pending watchdog timer interrupt is indicated by WDT_RESET in the Reset Status Register.

= 0 Interrupt generation disabled (default after reset).

= 1 Interrupt generation enabled.

IS_WDT (ro) IS_WDT indicates status of the WDO output signal and thereby indicates the status of the watchdog timer if started (see “Watchdog” on page 90).

= 0 Watchdog timeout.

= 1 Watchdog timer has not expired.

Table 51 Reset Status Register

$F160.000E_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	1	WDT_ RESET	BUS_ RESET	KEY_ RESET

The Reset Status Register allows to identify the on-board reset source which generated the latest hardware reset:

- watchdog – WDT_RESET,
- CompactPCI reset – BUS_RESET,
- front-panel reset key – KEY_RESET,
- or power-on reset: If all status bits in the Reset Status Register are cleared (0) after a reset, the reset has been generated due to a power-on reset. A power-on reset occurs when the power supply unit is turned on, or the power supply sensor detects that one of the required power supply voltages falls below a tolerable limit.

Once one of the bits has been set to 1, it is cleared (0) by setting RESET_STAT_CLR in the Miscellaneous Control Register (see page 93).

KEY_RESET (ro) — KEY_RESET indicates that a reset has been generated via the front-panel reset key if KEY_RESET = 1.

BUS_RESET (ro) — BUS_RESET indicates that a reset has been generated because the CompactPCI reset signal has been asserted if BUS_RESET = 1.

WDT_RESET (ro) — WDT_RESET indicates that a reset has been generated because of a watchdog timeout if WDT_RESET = 1.

Table 52 Miscellaneous Control Register

$F160.0004_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	RESET_STAT_CLR	reserved	SUPIO_PWDN	reserved

RESET_STAT_CLR (r/w) — RESET_STAT_CLR specifies to clear all reset status bits in the Reset Status Register when set to 1.

SUPIO_PWDN (r/w) — SUPIO_PWDN controls the Super I/O power-down mode.

- = 0 Turns off the power-down mode.
- = 1 Turns on the power-down mode.

6.3.12 SCR: ENUM Interrupt

Note: **All SPARC/CPCI-52x(G) variants provide the ENUM #1 interrupt since this is the ENUM interrupt of the CompactPCI interface related to the base board of the SPARC/CPCI-52x(G). However, note that in case of using the I/O-523G and only in this case, there is a second ENUM interrupt related to the CompactPCI interface of the I/O board: ENUM #2. ENUM #1 and ENUM #2 utilize the same interrupt request pin. Therefore the interrupt handler must determine the actual interrupt source by reading `IP_ENUM1` and `IP_ENUM2`.**

Table 53 ENUM Interrupt Control Register

F160.000A$_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	IP_ ENUM2	IE_ ENUM2	IP_ ENUM1	IE_ ENUM1

IE_ENUM*x* (r/w) — IE_ENUM*x* specifies if the respective ENUM #*x* interrupt is enabled (*x*=1 or 2). Only if enabled an interrupt is generated to the processor in case of a pending ENUM interrupt.

= 0 (default) — ENUM #*x* interrupt is disabled.

= 1 — ENUM #*x* interrupt is enabled.

IP_ENUM*x* (r/w) — IP_ENUM*x* indicates the status of the ENUM #*x* signal on the respective CompactPCI interface

- on the base board’s CompactPCI interface in case of IP_ENUM1
- or on the I/O-523G CompactPCI interface in case of IP_ENUM2.

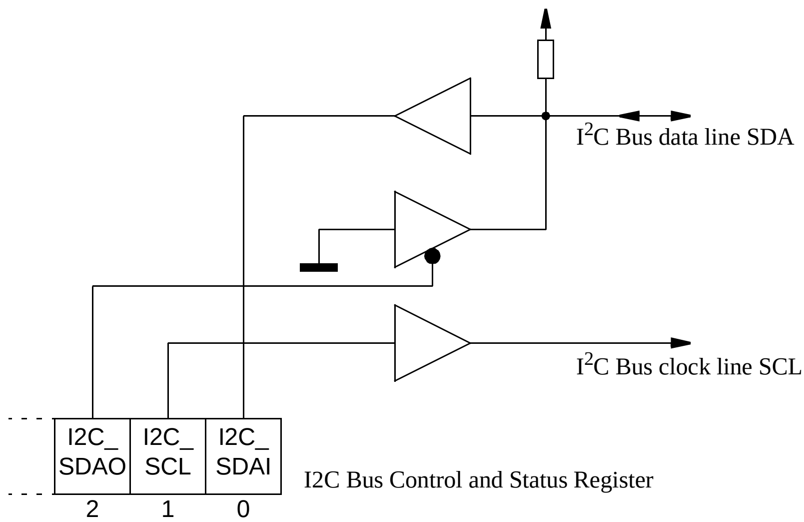
= 0 — The respective ENUM signal is high that is it is inactive.

= 1 — The respective ENUM signal is low that is it is active.

6.3.13 SCR: I^2C-Bus

The SPARC/CPCI-52x(G) is equipped with an I^2C bus consisting of a data line and a clock line which are software-controlled. For further information on the I^2C Bus, refer to the I^2C Bus specification. The figure below shows the I^2C Bus interface implementation in detail.

Figure 25 **I^2C Bus interface**



Accessible devices

The devices listed in the table below can be accessed via the I^2C Bus. They are I^2C Bus slaves and are identified by unique addresses also listed in the table below:

Table 54 **I^2C Bus slave addresses**

I^2C Bus Slave Address	Description
1001.111_2	Temperature sensor #1 LM75
1001.110_2	Temperature sensor #2 LM75
1010.000_2	ID PROM of the Base-520(G) – XICOR X24C04 Serial E^2PROM
1010.010_2	ID PROM of I/O-52x(G) – XICOR X24C04 Serial E^2PROM

The ID-PROMs are used to store board specific parameters. The temperature sensors can be programmed for temperature monitoring and protection against overheating. To locate the temperature sensors see figure 7 “Location diagram of the Base-520(G) (schematic)” on page 22.

The devices can be accessed via the I^2C Bus and can be read and written via the I2C Bus Control and Status Register.

Table 55 I^2C Bus Control and Status Register

F160.0003$_{16}$								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value	1	1	1	1	reserved	I2C_SDAO	I2C_SCL	I2C_SDAI

See I2C_SDAO for information how to write data to I^2C Bus slaves. See I2C_SDAI for information how to read data from I^2C Bus slaves.

I2C_SDAO (r/w) I2C_SDAO is used to force low (0) or high (1) level on the I^2C Bus data line.

Note: **Whenever data is read from another I^2C Bus participant, I2C_SDAO must be set to 1, otherwise data is corrupted.**

= 0 Low level (0) is forced on the data line.

= 1 High level (1) is forced on the data line.

Note: **Since the I2C_SDAO is realized as an open collector output onto the I^2C Bus data line, it is necessary to be aware of the typical RC-time constant. This time constant is 5 usec. It is important that the read-out of the data line via I2C_SDAI, which immediately follows the setting of the SDA data line to the high level 1, strictly respects the 5 usec time constant before executing the read-out. Otherwise the data will be read incorrectly on I2C_SDAI.**

I2C_SCL (r/w) I2C_SCL is used to control the state of the clock line of the I^2C bus.

= 0 Low level (0) is forced on the data line.

= 1 High level (1) is forced on the data line.

I2C_SDAI (ro) I2C_SDAI is used to read data from the I^2C Bus data line. However, to do so I2C_SDAO must be set to 1, otherwise data is corrupted.

- If I2C_SDAO is set (1), the state of I2C_SDAI indicates the state of the data line of the I^2C bus.
- If I2C_SDAO is cleared (0), low level (0) is forced on the data line of the I^2C bus and I2C_SDAI is cleared, too.

6.4 SCSI Interface – SYM53C875

Ultra/wide SCSI #1 and #2 (single-ended) are implemented on the Base-520(G) and on the I/O-52x(G) (SCSI), respectively. Ultra SCSI (Fast-20) is an extension of the SCSI-3 standard that expands the bandwidth of the SCSI bus and allows faster synchronous SCSI transfer rates (approximately double the synchronous transfer rates of Fast SCSI-2).
Each – SCSI #1 or #2 – is implemented by a PCI-Ultra SCSI (Fast-20) I/O Interface – SYM53C875, both independent from each other. 8-bit Ultra SCSI is provided via the front panel connector only on the Base-520(G). Ultra/wide (16-bit) SCSI is provided only via the backplane connector of Base-520(G) and I/O-52x(G).

Device features

- Single-chip high-performance PCI-Wide Ultra SCSI I/O Processor
- Enhanced PCI performance and flexibility
- SCSI SCRIPTS high-level programming interface and SCRIPTS instruction prefetch, allowing tailored SCSI sequences to be executed locally (SCRIPTS – Symbios Logic-developed SCSI programming language)
- 536-byte DMA FIFO buffer allowing bursts of up to 128 transfers
- Support for PCI extended access cycles: memory read multiple, memory read line, as well as memory write and invalidate
- Supports wide high-speed SCSI bus transfers in single-ended mode up to 40 MB/s synchronous Ultra SCSI (Fast-20) transfers and 14 MB/s asynchronous transfers
- Provides full SCSI-2 capabilities
- SCAM (SCSI configured automatically) Level 1 functionality
- 32 additional scratchpad registers for user-defined functions
- Direct PCI-to-SCSI connection
- Features Symbios Logic Tolerant SCSI driver and receiver technology for reliable operation in all cabling environments
- Solaris driver support for hard disk, tape, CD-ROM, and removable media peripherals

SCSI interrupt

The SCSI controller uses SCSI interrupt on the UIC for interrupting the UltraSPARC-IIi (see table 33 “Interrupt sources from the Base-520(G)” on page 72 and table 34 “Interrupt sources from the I/O-52x(G)” on page 73).

6.5 PMC Slots with Busmode Support

The PMC busmode signals are supported for both PMC slots via 5 general purpose I/O pins of the SCSI #2 controller on-board the I/O-52x(G) (see section 6.4 “SCSI Interface – SYM53C875” on page 97). The busmode signals allow detection of installed PMC cards and proper initialization of the PMC card according to the protocol supported by the installed PMC module.

Table 56 I/O pins for PMC busmode function

Busmode signal	Pin name at SCSI#2 controller
BUSMODE[4]	GPIO[0] / FETCH
BUSMODE[3]	GPIO[1] / MASTER
BUSMODE[2]	GPIO[2] / MAS2
PMC#1 BUSMODE[1]	GPIO[3]
PMC#2 BUSMODE[1]	GPIO[4]

BUSMODE[4…2] are driven by the host and specify the busmode command transferred to the PMC modules as described in table 57 “BUSMODE [4..2] (r/w) commands” on page 98. The answer of PMC module #*x* is transferred via PMC#*x* BUSMODE[1] as described in table 58 “PMC#x BUSMODE[1] (ro) response encoding” on page 99.

Table 57 BUSMODE [4..2] (r/w) commands

Busmode [4..2]	Command
000_2	The modules at PMC slot 1 and 2 shall return "Card Present", if they are plugged into the slot and no bus protocol is used. This is the default setting.
001_2	The modules at PMC slot 1 and 2 shall return "Card Present" if they are PCI capable and PCI protocol is used (default).
010_2	The modules at PMC slot 1 and 2 shall return "Card Present" if they are SBus capable and SBus protocol is used.
111_2	No host present
011_2, 100_2, 101_2, 110_2 are reserved	

Table 58 **`PMC#x BUSMODE[1]` (ro) response encoding**

PMC#*x* BUSMODE[1]	Description
0	“Card Present”: PMC module present which has the requested capability and uses the requested protocol
1	no PMC module present or PMC module does not have the requested capability or PMC module does not use the requested protocol

7 FORCE OpenBoot Enhancements

The OpenBoot ported to the SPARC/CPCI-52x(G) is based upon OpenBoot V3.10 obtained from Sun Microsystems. This section describes the enhancements to the standard OpenBoot firmware that have been done for the SPARC/CPCI-52x(G). Examples are given when it seems necessary to convey the usage of a particular word or a group of words.

Note: **OpenBoot is subject to changes. Some features are only available with specific versions of the software. Features not available on all OpenBoot versions are marked with the version given, e.g. ... (OpenBoot 3.10.4 and above) ... . For information on the latest OpenBoot version and how to upgrade refer to the SMART service accessible via the FORCE COMPUTERS World Wide Web site.**

Base information

For a description of standard OpenBoot 3.x firmware features, see the *OpenBoot 3.x Manual Set*.

Besides the commands already provided by the standard OpenBoot firmware, the OpenBoot firmware available on the SPARC/CPCI-52x(G) includes further words for

- accessing and configuring system specific components (see section 7.1 “System Configuration” on page 102),
- accessing and programming available flash EPROMs (see section 7.2 “Flash EPROM Support” on page 110).

For information on additional hardware dependencies, see section 7.3 “Hardware Dependencies” on page 116.

Notation

In general, each word is described using the following notation:

name (*stack-comment*) *description*

The *name* field identifies the name of the word being described.

The *stack-comment* notation which is enclosed in parentheses describes the stack parameters passed to and returned from a word. It shows the effect of the word on the evaluation stack. The parameters passed and returned to the word are separated by the “—” within the *stack-comment*:

parameters before execution — parameters after execution

The description body describes the semantics of the word. It also conveys the purpose and effect of the particular word.

7.1 System Configuration

This section consists of the following parts:

- section 7.1.1 “System Configuration Register Accesses” on page 102,
- section 7.1.2 “LEDs, Seven Segment Display and Rotary Switch” on page 106,
- section 7.1.3 “ID PROM” on page 107,
- section 7.1.4 “Viewing the Switch Status and Controlling the Temperature Sensors” on page 108,
- and section 7.1.5 “PCI-Probing – NVRAM Configuration Variables” on page 109.

7.1.1 System Configuration Register Accesses

The following commands are available to read data from and store data in the System Configuration Registers.

`led1-ctrl@` (— *byte*) returns the contents – an 8-bit data – of the First User LED Control Register.

`led1-ctrl!` (*byte* —) stores the 8 bit data *byte* in the First User LED Control Register.

`led2-ctrl@` (— *byte*) returns the contents – an 8-bit data – of the Second User LED Control Register.

`led2-ctrl!` (*byte* —) stores the 8 bit data *byte* in the Second User LED Control Register.

`idprom-ctrl@` (— *byte*) returns the contents – an 8-bit data – of the IDPROM Control Register.

`idprom-ctrl!` (*byte* —) stores the 8 bit data *byte* in the IDPROM Control Register.

`rotary-switch-stat@` (— *byte*) returns the contents – an 8-bit data – of the Rotary Switch Status Register.

`boot-rom-size-ctrl@` (— *byte*) returns the contents – a 2-bit data – of the Boot ROM Size Control Register.

`boot-rom-size-ctrl!` (*byte* —) stores the 2-bit data in the Boot ROM Size Control Register.

`user-rom-size-ctrl@` (— *byte*) returns the contents – a 2-bit data – of the User ROM Size Control Register.

`user-rom-size-ctrl!` (*byte* —) stores the 2-bit data in the User ROM Size Control Register.

`led-display@` (— *byte*) returns the contents – an 8-bit data – of the LED Display Control/Status Register. Since the LED Display Control Register is only writable, the command returns the contents of the LED Display Control Shadow Register.

`led-display!` (*byte* —) stores the 8-bit data *byte* in the LED Display Control/Status Register. Since the LED Display Control Register is only writable, the command stores the given data in the LED Display Control Shadow Register, too.

`lca-id@` (— *byte*) returns the contents – an 8-bit data – of the LCA ID Register.

`supio_pwdn@` (— `true` | `false`) returns the state of the SuperI/O Power Down Mode Register. The SuperI/O is put into Power Down Mode if `true` is returned.

`supio_pwdn!` (`true` | `false` —) controls the Power Down Mode Register of the SuperI/O. The SuperI/O can be put into Power Down Mode if `true` is used with this command.

`eject_fd@` (— `true` | `false`) returns the state of the automatic Floppy Disk Eject Register. The floppy disk is ejected if `true` is returned.

`eject_fd!` (`true` | `false` —) controls the automatic Floppy Disk Eject Register. The floppy can be ejected if `true` is used with this command. Immediately after setting the Floppy Disk Eject Register to eject the floppy via `eject_fd! true`, it should be cleared again via `eject_fd! false`.

`reset_stat_clr!` (`true` | `false` —) controls the Reset Status Control Register. The register is used to clear the status bits in the Reset Status Register after a reset has occurred. All status bits are cleared if `true` is used with this command. Once the Reset Status Control Register is set (`true`), it is cleared automatically.

`boot_wp@` (— `true` | `false`) returns the state of the Boot Flash EPROM Write Protection Switch (SW4-3). The boot flash is write protected if `true` is returned.

`user_wp@` (— `true` | `false`) returns the state of the User Flash EPROM Write Protection Switch (SW4-4). The user flash is write protected if `true` is returned.

`scsi_front@` (— `true` | `false`) returns the state of the SCSI Front Panel Termination Switch (SW5-1). The SCSI front panel termination is automatic if `true` is returned. In this case the termination is only enabled if no SCSI cable is connected to the front panel. Otherwise, the SCSI front panel termination is disabled.

`scsi_bp@` (— `true` | `false`) returns the state of the SCSI Backplane Termination Switch (SW5-2). The SCSI backplane termination is disabled if `true` is returned.

`ie_sysfail_enum@` (— `true` | `false`) returns the state of the ENUM#1 Interrupt Enable Register. The ENUM#1 interrupt is enabled if `true` is returned. After Reset ENUM#1 is disabled.

`ie_sysfail_enum!` (`true` | `false` —) controls the ENUM#1 Interrupt Enable Register. The ENUM#1 interrupt is enabled if `true` is used with this command. After Reset ENUM#1 is disabled.

`ip_sysfail_enum@` (— `true` | `false`) returns the state of the ENUM#1 Interrupt Pending Register. The ENUM#1 interrupt is pending if `true` is returned. In that case an interrupt is generated to the processor provided that the ENUM#1 interrupt is enabled by the ENUM#1 Interrupt Enable Register. This signal is low level sensitive.

`ip_sysfail_enum!` (`true` | `false` —) controls the SYSFAIL- Interrupt Pending Register. The SYSFAIL- interrupt is pending if `true` is used with this command. In that case an interrupt is generated to the processor provided that the SYSFAIL- interrupt is enabled by the SYSFAIL- Interrupt Enable Register. This signal is rising edge sensitive.

`ip_sysfail_enum!` (`true` | `false` —) controls the ENUM#1 Interrupt Pending Register. The ENUM#1 interrupt is pending if `true` is used with this command. In that case an interrupt is generated to the processor provided that the ENUM#1 interrupt is enabled by the ENUM#1 Interrupt Enable Register. This signal is low level sensitive.

`ie_acfail@` (— `true` | `false`) returns the state of the ENUM#2 Interrupt Enable Register. The ENUM#2 interrupt is enabled if `true` is returned. After Reset the ACFAIL- is disabled.

`ie_acfail!` (`true` | `false` —) controls the ENUM#2 Interrupt Enable Register. The ENUM#2 interrupt is enabled if `true` is used with this command. After Reset the ACFAIL- is disabled.

`ip_acfail@` (— `true` | `false`) returns the state of the ENUM#2 Interrupt Pending Register. The ENUM#2 interrupt is pending if `true` is returned. In that case an interrupt is generated to the processor provided that the ENUM#2 interrupt is enabled by the ENUM#2 Interrupt Enable Register. This signal is low level sensitive.

`ip_acfail!` (`true` | `false` —) controls the ENUM#2 Interrupt Pending Register. The

ENUM#2 interrupt is pending if `true` is used with this command. In that case an interrupt is generated to the processor provided that the ENUM#2 interrupt is enabled by the ENUM#2 Interrupt Enable Register. This signal is low level sensitive.

`ie_wdt@` (— `true` | `false`) returns the state of the Watchdog Timer Interrupt Register. The watchdog timer interrupt is enabled if `true` is returned. After Reset the Watchdog Timer Interrupt is disabled.

`ie_wdt!` (`true` | `false` —) controls the Watchdog Timer Interrupt Register. The watchdog timer interrupt is enabled if `true` is used with this command. After Reset the Watchdog Timer Interrupt is disabled .

`is_wdt@` (— `true` | `false`) returns the state of the output signal WDO of the MAX815 Watchdog Timer device. The watchdog timer intervall is not expired if `true` is returned.

`wdi!` (`true` | `false` —) controls the Watchdog Timer Trigger Input Register. The watchdog timer is cleared by changing the value of the WDI input. If the watchdog timer intervall expires, the watchdog timer device MAX815 activates its WDO output and a Watchdog timer interrupt may be generated. To start the watchdog timer it is necessary to trigger WDI once.

`ie_temp@` (— `true` | `false`) returns the state of the Temperature Control Interrupt Register. The temperature control interrupt is enabled if `true` is returned. After Reset the temperature control interrupt is disabled.

`ie_temp!` (`true` | `false` —) controls the Temperature Control Interrupt Register. The temperature control interrupt is enabled if `true` is used with this command. After Reset the temperature control interrupt is disabled.

`ip_temp@` (— `true` | `false`) returns the state of both output signals of the temperature sensors. A temperature interrupt is pending and `true` is returned if one of them has put its output signal active. An interrupt is generated to the processor provided that the Temperature Enable Interrupt Register is enabled.

`is_temp1@` (— `true` | `false`) returns the state of the output signal of the first digital temperature sensor and thermal watchdog. The state of the temperature's output signal is high (`1`) if `true` is returned.

`is_temp2@` (— `true` | `false`) returns the state of the output signal of the second digital temperature sensor and thermal watchdog. The state of the temperature's output signal is high (`1`) if `true` is returned.

7.1.2 LEDs, Seven Segment Display and Rotary Switch

The commands described below are available to control the seven segment LED display, the user LEDs, as well as to get information about the state of the rotary switch.

`diag-led!` (*byte* —) stores the data *byte* passed to the command in the register used to control the seven segment display.

`>7-seg-code` (*u* — *7-seg-code*) converts the value *u* to its corresponding seven segment code *7-seg-code*. Only the least significant four bits of the value *u* are considered.

`led!` (*colour freq led#* —) controls the user LED identified by *led#*. *led#* = `0` specifies the first user LED, `1` specifies the second user LED. The command only considers the state of bit `0` of the value *led#*. *colour* and *freq* define the colour of the LED and the frequency at which the LED is blinking. The following constants are defined for *colour*: `black` (= LED is turned off), `green`, `red`, and `yellow`. The following constants are defined for *freq*: `no-blinking` (= LED is turned on permanently), `slow`, `moderate`, and `fast`.

Example:

The following command makes the second user LED blink with a moderate frequency in red:

```
ok red moderate 1 led!
```

`led-on` (*led#* —) turns on the user LED identified by *led#*. *led#* = `0` specifies the first user LED, `1` specifies the second user LED. The command only considers the state of bit `0` of the value *led#*.

`led-off` (*led#* —) turns off the user LED identified by *led#*. *led#* = `0` specifies the first user LED, `1` specifies the second user LED. The command only considers the state of bit `0` of the value *led#*.

`led?` (*led#* — `true` | `false`) determines the state of the LED identified by *led#*, and returns either `true` or `false` to indicate if the LED is turned on or off. *led#* = `0` specifies the first user LED, `1` specifies the second user LED. The command only considers the state of bit `0` of the value *led#*.
If the LED is turned on, `true` is returned; otherwise `false` is returned.

`toggle-led` (*led#* —) determines the state of the user LED identified by *led#*, and turns the LED on or off. The LED is turned on when it was turned off before, and vice versa. *led#* = `0` specifies the first user LED, `1` specifies the second user LED. The command only considers the state of bit `0` of the value *led#*. Regardless of the colour having been set, the LED shines yellow after using this command.

`rotary-switch@` (— *byte*) returns the current state of the rotary switch. The value of *byte* may be one of the values in the range `0`…`15`. `0` corresponds to position `0` of the rotary switch, `1` corresponds to position `1`, and so forth.

7.1.3 ID PROM

Depending on the SPARC/CPCI-52x(G) variant under consideration there are several ID PROMs, all connected via an I^{2}C bus. At least 1 ID PROM is present, the one assembled on the base board. Each ID PROM if assembled is accessible as I^{2}C bus slave via the I^{2}C bus commands listed below:

`select-idprom` (*slv#* — *flag*) selects an ID PROM by adjusting the I^{2}C bus slave address. If the ID PROM to be selected is not accessible or not available, an error message is displayed. The resulting *flag* is `true` only if the ID PROM is present and the selection was successful. The number *slv#* of the I^{2}C bus device can be one of the values listed in the table below:

slv#	**I^{2}C bus slave address**	**Description**
x in the I^{2}C bus slave address depends on the action to be done: • *x* = `1` for read access and *x* = `0` for write access.		
1	$1010.000x_2$	ID PROM on base board
2	$1010.010x_2$	ID PROM on I/O-board
3	$1010.100x_2$	reserved (related to CompactPCI connector J3)
4	$1010.110x_2$	

`i2c!` (*addr data i^{2}c-slave-addr* —) transmits a byte *data* to the ID PROM which is identified by its *i^{2}c-slave-addr*. For *i^{2}c-slave-addr* values see the `select-idprom` description above. *addr* specifies the offset within the ID PROM's address range, at which *data* is stored in the ID PROM.

`i2c@` (*addr i^{2}c-slave-addr* — *data*) reads a byte *data* from the ID PROM which is identified by its *i^{2}c-slave-addr*. For *i^{2}c-slave-addr* values see the `select-idprom` description above. *addr* specifies the offset within the ID PROM's address range, at which *data* is stored in the ID PROM.

`mem>idprom` (*src-addr dest-addr size* —) copies a number of bytes from the on-board memory to the ID PROM selected via `select-idprom`. The number of bytes to be copied is specified by *size*, *size* = `1` …`512`. The source data start at the virtual address *src-addr* in the on-board memory. The start address for the copied data in the ID PROM is specified by *dest-addr*, *dest-*

addr = `0` …`511`.

`mem>idprom` (*src-addr dest-addr size* —) copies a number of bytes from the on-board memory to the ID PROM selected via `select-idprom`. The number of bytes to be copied is specified by *size*, *size* = `1` …`512`. The source data start at the virtual address *src-addr* in the on-board memory. The start address for the copied data in the ID PROM is specified by *dest-addr*, *dest-addr* = `0` …`511`.

`idprom>mem` (*src-addr dest-addr size* —) copies a number of bytes from the ID PROM which has been selected via `select-idprom` to the on-board memory. The number of bytes to be copied is specified by *size*, *size* = `1` …`512`. The source data start at the address *src-addr* in the ID PROM, *src-addr* = `0`…`511`. The start address for the copied data in on-board memory is specified by *dest-addr*.

7.1.4 Viewing the Switch Status and Controlling the Temperature Sensors

---

Note: **All temperature values are measured in °C and specified as *degree* degrees plus *1/10-degree* 1/10-degrees.**

`.cpu-switch-stat` (—) displays the current state of all switches on the SPARC/CPCI-52x(G).

`get-t` (*sensor# — 1/10-degree degree*) reads the result of the last temperature conversion for the specified temperature sensor *sensor#*, *sensor#* = `1`, `2`.

`get-t-low` (*sensor# — 1/10-degree degree*) returns the current setting of the THYST temperature register of the temperature sensor specified by *sensor#, sensor#* = `1`, `2`.

`get-t-high` (*sensor# — 1/10-degree degree*) returns the current setting of the TOS temperature register of the temperature sensor specified by *sensor#*, *sensor#* = `1`, `2`.

`get-conf` (*sensor# — byte*) reads the status register of the temperature sensor specified by *sensor#, sensor#* = `1`, `2`. The contents of the status register data is left on the evaluation stack.

`set-t-high` (*1/10-degree degree sensor#* —) sets the TOS temperature register of the temperature sensor specified by *sensor#, sensor#* = `1`, `2`.

`set-t-low` (*1/10-degree degree sensor#* —) sets the THYST temperature register of the temperature sensor specified by *sensor#, sensor#* = `1`, `2`.

`set-conf` (*byte sensor#* —) stores the value *byte* in the configuration register of the temperature sensor specified by *sensor#*, *sensor#* = `1`, `2`. The contents of the status register data is left on the evaluation stack.

`watch-temperature` (—) puts both temperature sensors into interrupt mode and initiates continous temperature conversion.

`monitor-t` (*sensor#* —) monitors the actual temperature of the temperature sensor specified by *sensor#*, *sensor#* = `1`, `2`.

7.1.5 PCI-Probing – NVRAM Configuration Variables

The current state of the configuration variables are displayed using the `printenv` command, and are modified using either the `setenv`, or the set-default command provided by OpenBoot.

The following NVRAM configuration variables are related to probing the PCI buses:

`pcia-probe-list` specifies the probe-list containing the device numbers to be probed at the CompactPCI bus controlled by the on-board arbiter.

`pcib-probe-list` specifies the probe-list containing the device numbers to be probed onboard.

`pcia-io-probe-list` specifies the probe-list containing the device numbers to be probed at the second CompactPCI bus controlled by the arbiter of the I/O board. This variable is only effective if the I/O-523 is used as I/O-board because only the I/O-523 provides an additional arbiter due to the support of the second CompactPCI segment.

`pcib-io-probe-list` specifies the probe-list containing the device numbers to be probed on the I/O-board.

7.2 Flash EPROM Support

The boot and user flash available on the SPARC/CPCI-52x(G) are both programmable via the appropriate hardware setup and the appropriate OpenBoot commands (see section 7.2.1 “Flash EPROM Programming” on page 110 which includes an example for programming the user flash on page 111).
This allows to program the user flash with an executable image and use the FORCE OpenBoot enhancements to load and execute such an image from user flash (see section 7.2.3 “Loading and Executing Programs from User Flash EPROM” on page 115). For general information on NVRAM configuration parameters and methods available for flash EPROM, see section 7.2.2 “Flash EPROM Device Node” on page 112.

Note: **Note the hardware dependencies for the flash memory driver documented in section 7.3 “Hardware Dependencies” on page 116.**

7.2.1 Flash EPROM Programming

The commands listed below are available to access and program the on-board flash EPROM.

`flash-messages` (— *vaddr*) returns the virtual address of the variable `flash-messages`. The state of this variable controls whether the words to erase and program the flash memories display messages while erasing or programming the flash memories. Messages are not displayed after entering `flash-messages off`<eol>. They are displayed after entering `flash-messages on`<eol>. The virtual address is returned after entering `flash-messages` without further text.

`flash-va` (— *vaddr*) returns the virtual base address *vaddr* of the flash EPROM programming window. The address is only valid, if the flash EPROM has previously been selected using `select-flash`.

`boot-flash-va` (— *vaddr*) returns the virtual base address *vaddr* of the boot flash EPROM.

`user-flash-va` (— *vaddr*) returns the virtual base address *vaddr* of the user flash EPROM. When the user flash EPROM is not accessible directly, but only through the flash EPROM programming window, then the address returned is zero. On the SPARC/CPCI-52x(G) the user flash EPROM is accessible only through the flash EPROM programming window. Thus, the commands described above have to be used to access the user flash EPROM.

`select-flash` (`USER` | `BOOT` —) selects the flash EPROM to be programmed and prepares the selected flash EPROM for programming. No further words may follow in the same command line (see “Example: Programming the User Flash” on page 111). `USER` selects the user flash, `BOOT` the boot flash.
In detail, the number and size of the available flash devices are determined, as well as the size of the flash EPROM programming window. The flash EPROM programming window is mapped and the virtual base address of the window is stored internally. The address may be obtained by `flash-va`.

`move>flash` (*source-addr dest-addr count* —) programs the selected flash EPROM beginning at *dest-addr* with *count* number of Bytes which are fetched from *source-addr*. Use `select-flash` to select the flash.

`flash>move` (*source-addr dest-addr count* —) copies *count* number of Bytes from the selected flash EPROM beginning at *source-addr* to *dest-addr*. Use `select-flash` to select the flash.

`fill-flash` (*dest-addr count pattern* —) fills the selected flash EPROM beginning at *dest-addr* with a particular *pattern*. The number of Byte to be programmed in the flash EPROM is given by *count*. Use `select-flash` to select the flash.

`erase-flash` (*device-number* —) erases the contents of a device of the selected flash EPROM. The device is identified by *device-number*. Device numbering starts at `0`. Use `select-flash` to select the flash.

`c!-flash` (*byte addr* —) stores *byte* at the location *addr* within the selected flash EPROM. Use `select-flash` to select the flash.

`w!-flash` (*half-word addr* —) stores the *half-word* (16 bits) at the location *addr* within the selected flash EPROM. Use `select-flash` to select the flash EPROM.

`l!-flash` (*word addr* —) stores the *word* (32 bits) at the location *addr* within the selected flash EPROM. Use `select-flash` to select the flash.

Example: Programming the User Flash

The user flash EPROM is prepared for programming by:

```
ok select-flash USER
USER flash memory is selected for programming
2048 Kbyte BOOT flash memory is available at 0xff550000
2048 Kbyte USER flash memory is available at 0xff350000
ok
```

`select-flash` informs the user that the user flash EPROM has been made accessible. It displays the available boot flash EPROM and user flash EPROM.

After the user flash has been selected, all following commands operate on the user flash. For example, to read data from the user flash EPROM, the command `flash>move` is used as follows:

```
ok flash-va h# 10.0000 h# 20.0000 flash>move
ok
```

Thereby, the contents of the entire user flash are copied to main memory beginning at address 10.0000_{16}.

A specific area within the selected flash EPROM is read as follows:

```
ok flash-va h# 6.8000 + h# 10.0000 h# 5.8c00 flash>move
ok
```

The source data start at address `flash-va` + 6.8000_{16}. They are copied to main memory starting at address 10.0000_{16} . And the amount of data copied is 363520 Bytes.

7.2.2 Flash EPROM Device Node

The device tree of OpenBoot for the SPARC/CPCI-52x(G) contains a device node associated with the user and boot flash EPROM. The device alias `flash` is available as an abbreviation of the flash EPROM device path.
Thereby, it is possible to load an executable image stored in the available user flash EPROM into memory and start such an executable (see section 7.2.3 “Loading and Executing Programs from User Flash EPROM” on page 115).

Vocabulary

The vocabulary of the flash EPROM device node includes the standard commands recommended for a *byte* device. This vocabulary is only available when the flash EPROM device node has been selected using one of the following 2 methods:

- `cd flash` selects the flash EPROM device and makes it the current node:

```
ok cd flash
```

- `select-dev` can also be used to select the flash EPROM device node. However, before using this command, the NVRAM configuration parameters `bootflash-#megs` and `bootflash-#devices` have to be set properly (see “NVRAM Configuration Parameters” on page 113).

After selecting the flash EPROM device node, the word `words` displays the names of the methods of the flash EPROM device.

```
ok words
close         open         selftest     reset     load
write-blocks  read-blocks  seek         write     read
max-transfer  block-size
```

To unselect the current device node, i.e. leaving no node selected, use `device-end`.

```
ok device-end
ok
```

NVRAM Configuration Parameters

The NVRAM configuration parameters listed below are available to control loading of an image from the user flash EPROM, i.e. boot from the user flash EPROM.

Note: **Note that the parameters indicate the purpose of booting from the user flash EPROM and therefore are called `bootflash`...**

The current state of the configuration parameters

- is displayed using `printenv`,
- and is modified using either `setenv`, or `set-default`.

`bootflash-#megs` specifies the amount of available user flash EPROM in MByte. Default: 0 MByte.

`bootflash-#devices` specifies the number of available user flash EPROM devices. Default: no devices.

Methods

The methods listed below are available in the flash EPROM vocabulary:

`open` (— `true` | `false`) prepares the package for subsequent use. `true` is returned if the device has been opened successfully. Otherwise, `false` is returned. Usually, the fail state is indicated when the NVRAM configuration parameters `bootflash-#megs` and `bootflash-#devices` are not consistent.

`close` (—) frees all resources allocated by `open`.

`reset` (—) puts the flash EPROM device into quiet state.

`selftest` (— *error-number*) always returns `0` as *error-number*.

`read` (*addr length — actual*) reads at most *length* Bytes from the flash EPROM and copies it to memory beginning at address *addr*. If *actual* is `0` or negative, the read failed. The value of *length* can be chosen independently of the device's block size. For information on the start address within the flash EPROM see the description of the `seek` command.

`write` (*addr length — actual*) discards the information passed to the command. It always returns `0` to indicate that the device does not support this function. However, this command is available to be standard compliant.

`seek` (*offset file# — error?*) seeks to Byte *offset* within the file identified by *file#*. An internal position counter is maintained and updated whenever a method is called to read data from or to store data in the flash EPROMs. The position counter can be adjusted using the `seek` command.

- If *offset* and *file#* are both `0`, the internal position counter is reset to offset `0`.
- Otherwise, the value of *file#* is ignored and the value of *offset* is assigned to the internal position counter. A subsequent access to the flash EPROM then starts at the adjusted offset.

After a successful seek, *error?* is `0`, otherwise `-1` is returned to indicate the fail state.

`read-blocks` (*addr block# #blocks — #read*) reads *#blocks* number of blocks where each block is of length *block-size* Byte. The blocks are read from the device beginning at the device block *block#*. The read data are copied to memory at address *addr*. `read-blocks` returns the number of blocks actually read (*#read*).

`write-blocks` (*addr block# #blocks — #written*) discards the information passed to the command and always returns zero to indicate that the device does not support this function. However, this command is available to be standard compliant.

`block-size` (*— bytes*) returns the current setting of the block size *bytes* in Byte. This always is the size of the flash EPROM programming window.

`max-transfer` (*— bytes*) returns the size *bytes* in Byte of the largest single transfer the device can perform. This always is a multiple of the value returned by `block-size`.

`load` (*addr — length*) reads a stand-alone program from the flash EPROM beginning at offset 0_{16} and stores it beginning at address *addr*. It returns the number of Bytes *length* read from the flash EPROM.

7.2.3 Loading and Executing Programs from User Flash EPROM

Besides the ability to load and execute an executable image from disk, from a network component, or from other components, the SPARC/CPCI-52x(G) OpenBoot also provides a convenient way to load and execute an executable image from available user flash EPROM. The executable image to be loaded has to be

- a binary image in `a.out` format,
- a FORTH program,
- or an FCode program.

Manual loading

To load and execute an image from the flash EPROM use the device alias `flash` together with the `boot` command:

```
ok boot flash
```

Automatic loading

The following NVRAM configuration parameters can be modified to determine whether or not the system loads an executable image automatically after a power-up cycle or system reset:

```
auto-boot?
boot-device
```

Example:

Assume that the CPU board provides 1 user flash EPROM device which is 1 MByte in size. The following commands load and execute an image from the flash EPROM automatically after a power-up cycle or system reset:

```
ok setenv bootflash-#devices 1
bootflash-#devices =  1
ok setenv bootflash-#megs 1
bootflash-#megs =     1
ok setenv boot-device flash
boot-device =         flash
ok setenv auto-boot? true
auto-boot? =          true
ok reset
```

7.3 Hardware Dependencies

Note: **To make use of the features described in this section proceed as follows:**

- copy the OpenBoot image from the boot PROM which is a read-only device to the boot flash EPROM (see section 7.3.1 "Copying the OpenBoot Image from Boot PROM to Boot Flash EPROM" on page 116)
- and use the boot flash EPROM for booting (see SW6-2).

7.3.1 Copying the OpenBoot Image from Boot PROM to Boot Flash EPROM

The following section describes how to copy the OpenBoot image from the boot PROM into the boot flash EPROM. The copy is done via some memory space and makes use of the ability to switch between the boot PROM and boot flash EPROM. There are 2 versions of this procedure. The first version uses the OpenBoot command `plcc2tsop` which is available in OpenBoot versions 3.10.4 or greater whereas the second version applies in any case.

`plcc2tsop` (—) copies the contents of the boot PROM into the boot flash EPROM. A sample dialog is given below. To be successful, the boot flash write protection has to be disabled by setting SW4-3 to `ON`.

```
ok
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
2048 Kbyte BOOT flash memory is available at 0xff550000
No USER flash memory is available
       Done
Step3: Updating TSOP Boot flash ...
       Done
    COPY OBP into TSOP successful
    -----------------------------

    Now you have to switch SW 6-2 to the ’ ON’ position
    and thereafter reset the syste
ok
```

Note: **In case of an error during one of the 2 following procedures proceed as follows:**

- select booting from the boot PROM,
- reset the system,
- and try again.

1. version: OpenBoot 3.10.4 or greater

To copy the OpenBoot image from the boot PROM into the boot flash EPROM, do the following in case of OpenBoot version 3.10.4 or greater:

1. Make sure to have the CPU board installed with SW4-3 set to `ON` and and SW6-2 set to `OFF`.
2. Copy the OpenBoot image by entering:

```
ok plcc2tsop
```

The OpenBoot image is copied from the boot PROM into the boot flash EPROM. To operate the CPU board, install the CPU board with the boot flash EPROM used for subsequential booting by setting SW6-2 to `ON`.

2. version: any OpenBoot version

In case of an OpenBoot version previous to 3.10.4, copy the OpenBoot image from the boot PROM into the boot flash EPROM as follows:

1. Make sure to have the CPU board installed with SW4-3 set to `ON` and and SW6-2 set to `OFF`.
2. Map the boot PROM and a specific memory space, and copy the OpenBoot image from the boot PROM into memory by entering:

```
ok 1.0000 0 1meg memmap value my-mem
ok 1ff.f000.0000 0 1meg memmap value my-plcc-flash
ok my-plcc-flash my-mem 1meg move
```

3. Select the boot flash EPROM for booting by entering:

```
ok 1ff.f160.0005 1d spacec@
ok 1 or 1ff.f160.0005 1d spacec!
```

4. Select the boot flash EPROM for programming, erase the boot flash EPROM, and copy the OpenBoot image into the boot flash EPROM by entering:

```
ok select-flash BOOT
BOOT flash memory is selected for programming
2048 Kbyte BOOT flash memory is available at 0xff550000
…
ok 0 erase-flash
Erasing selected flash memory ... passed!
ok my-mem boot-flash-va 1meg move>flash
```

The OpenBoot image is copied from the boot PROM into the boot flash EPROM. To operate the CPU board, install the CPU board with the boot flash EPROM used for subsequential booting by setting SW6-2 to `ON`.

7.3.2 Drop-in Drivers

OpenBoot supports drop-in drivers, i.e. drivers that may be added to OpenBoot during start-up time. The drop-in drivers are placed inside a specific drop-in driver area within the OpenBoot image so that they cannot be erased during a power-up once they are added.

The drop-in drivers are special FCode drivers having a unique drop-in driver header. At certain points during start-up, OpenBoot scans the drop-in driver area for specific drivers to be loaded at specific points. 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 other external media.

For example, drop-in drivers can automatically be loaded from a floppy disk when OpenBoot detects that a driver for a present PCI device is not present. In this case, the drop-in driver is loaded from floppy disk and is stored in the drop-in driver area. Thereby, after the next power-up sequence the driver is available in ROM.

7.3.3 Flash Memory Driver

The functionality of the flash memory driver depends on the type of boot device used:

- If the CPU board boots from the boot flash EPROM, the full functionality of the flash memory driver can be used as documented in section 7.2 “Flash EPROM Support” on page 110.
- If the CPU board boots from the boot PROM, the driver cannot write to the boot PROM because it is a read-only device. However, a write access is done to identify the flash EPROM device when selecting the boot flash EPROM. For example, selecting the boot flash EPROM via `select-flash BOOT` results in the following error message:

```
Flash memory either is not available or
protected against writing!
```

Product Error Report

PRODUCT:	SERIAL NO.:
DATE OF PURCHASE:	ORIGINATOR:
COMPANY:	POINT OF CONTACT:
TEL.:	EXT.:
ADDRESS: __________ __________ __________	
PRESENT DATE:	
AFFECTED PRODUCT: ❏ HARDWARE ❏ SOFTWARE ❏ SYSTEMS	AFFECTED DOCUMENTATION: ❏ HARDWARE ❏ SOFTWARE ❏ SYSTEMS
ERROR DESCRIPTION:	

THIS AREA TO BE COMPLETED BY FORCE COMPUTERS:

DATE:

PR#:

RESPONSIBLE DEPT.: ❏ MARKETING ❏ PRODUCTION
ENGINEERING ➠ ❏ BOARD ❏ SYSTEMS

✉ Send this report to the nearest Force Computers headquarter listed on the back of the title page.