SPARCbook™ 3 Series

Technical Reference Manual

980327-02

Trademarks

Issue 2.2 (Draft) of February 10, 1997
Part number 980327-02

Printed in the UK

Contents

Chapter 5 SCSI Controller

Chapter 6 Ethernet Interface

Chapter 7 PCMCIA Interface

Chapter 8 ISDN and 16-bit Audio

About This Guide

The SPARCbook 3 Technical Reference Manual is written for the hardware engineer wishing to carry out service or repairs, and at the software engineer wishing to implement hardware drivers. It is assumed that you are familiar with the operation of SPARCbook 3, as detailed in the SPARCbook 3 User Guide, and that you have an understanding of computer hardware.

Note

The SPARCbook 3 Technical Reference Manual covers all models of SPARCbook 3. Where information for one model differs to information for another model, this is indicated in the text.

Document Summary

The SPARCbook 3 Technical Reference Manual comprises the following chapters:

- Chapter 1 , *Architecture Overview,* discusses the main features of the SPARCbook 3 and introduces the main hardware devices that provide control over the SPARCbook 3's operations. The internal architecture of SPARCbook 3 is described, showing how the major devices are connected together.
- Chapter 2, *Microprocessor*, provides an overview of the SPARC processor.
- Chapter 3, *Memory Map and Interrupts*, describes the addressing architecture and the interrupt architecture of the SPARCbook 3.
- Chapter 4 , *Serial Interface*, discusses the serial interface of the SPARCbook 3.
- Chapter 5, *SCSI Communications,* discusses the SCSI controller.
- Chapter 6, *Ethernet Interfac*e, discusses the Ethernet interface of the SPARCbook 3.
- Chapter 7, *PCMCIA*, discusses the PCMCIA interface implemented in the SPARCbook 3.
- Chapter 8, *ISDN and 16-Bit Audio Controller*, discusses the ISDN and 16-bit audio controller.
- Chapter 9, *Modem*, discusses the internal modem on the SPARCbook 3.
- Chapter 10, *Parallel Interface*, discusses the parallel interface on the SPARCbook 3.
- Chapter 11, *Display Interface*, discusses the display interface implemented in the SPARCbook 3. The discussion is centered on the Brooktree Bt445 RAMDAC, on which the interface is based.
- Chapter 12, *Microcontroller Subsystem*, discusses the microcontroller subsystem. This is used to provide internal control over such things as the display brightness, keyboard and mouse scanning and power management.

Definitions

The following conventions are used in the SPARCbook 3 Technical Reference Manual:

Logic states

The terms *clear* or *low* indicate that the signal being discussed is at the logic level ‘0’.

The terms *set* or *high* indicate that the signal being discussed is at the logic level ‘1’.

The term *asserted* indicates that a signal is in its true or active state regardless of whether that state is high or low.

The term *negated* indicates that a signal is in its false or inactive state regardless of whether that state is high or low.

Data entities

A *halfword* is taken to contain 16 bits.

A *word* is taken to contain 32 bits.

A *doubleword* is taken to contain 64 bits

Typographical conventions

Different typography is used in this guide to distinguish between normal text, examples of SPARCbook responses, and cases where you are required to provide input using the keyboard or mouse.

Key presses, buttons, and field names

Key presses are shown in **Helvetica bold**. In order to perform certain tasks, you need to press two or more keys; for example:

To switch off your SPARCbook, press **Pause-O.**

In this case, you should press the **Pause** key down first, and then, while holding the **Pause** key down, press the **O** key.

Buttons and field names are also shown in **Helvetica bold**. For example:

Click the **Transmit** button.

Type the name of the file that you want to send in the **File Name** field.

Solaris commands

Information displayed on your SPARCbook screen by the Solaris Operating System is shown in `Courier` font. `Courier` is also used to describe system utilities and commands. For example:

The `mail` system will inform you when there is incoming mail from another user.

```
you have mail
```

`Bold Courier` is used in examples to show what you must type in order to perform a specific task. For example:

To report the current time and date you should use the `date` command:

```
% date
```

Notes, cautions and warnings

Notes are used throughout this manual to explain items of related interest to the topic under discussion, and are used to refer the reader to another part of the manual, or to other documentation.

Note

This is an example of a note, used as to provide additional information.

Cautions are used to advise the reader of actions that if carried out may cause damage to the SPARCbook 3.

Caution

This is an example of a caution

Warnings are used to draw your attention to actions that could cause personal injury or pose a hazard to life. For example:

WARNING!

THE AC ADAPTER SUPPLIED WITH YOUR SPARCBOOK 3 CONTAINS HAZARDOUS VOLTAGES. IT CONTAINS NO USER SERVICEABLE PARTS. DO NOT REMOVE THE COVERS.

Architecture Overview

1

This chapter discusses the architecture of the SPARCbook 3. It describes the main system components and how they are packaged together to deliver workstation-class performance in a compact notebook form factor.

1.1 Introduction

At the heart of the SPARCbook 3 design concept is the Tadpole Advanced Notebook Architecture (ANA). This defines a set of goals and guidelines to which the SPARCbook 3 range of systems are designed. It is a modular approach which results in a system that implements highly integrated components to provide the performance and I/O facilities normally associated with desktop workstations. It also results in a system that can be readily upgraded by the user with larger memory (up to 128 MB) or disk capacities or returned to the factory for upgrades with the fastest CPUs available for notebook implementation.

1.2 Main Components

The SPARCbook 3 contains three printed circuit boards. These are the S3-XP Base board, the S3-XP or the S3TX CPU module, and the microcontroller board.

1.2.1 Base board

The S3-XP Base board carries all of the I/O components together with the display controller, RAMDAC and 2MB of Video RAM, and the battery management hardware. It is populated on both sides using mainly surface mount devices in order to keep its physical dimensions to a minimum. The Base board also carries two PCMCIA sockets and the I/O panel which is visible at the rear of the assembled system.

The Base board provides mounting points and sockets to accommodate the CPU module.

1.2.2 CPU module

The CPU module carries the main SPARC CPU. The CPU module is extended to carry the main memory SIMMS. This physical arrangement has the advantage of making the SIMMs very easy to fit to or remove from a fully assembled system through the battery tray without the use of tools.

The CPU module is mounted onto the base board such that the CPU itself is sandwiched between the CPU module and Base board. However, an interesting feature of the Base board is a large hole through which a heatsink fitted to the main CPU is allowed to protrude when the two boards

are fixed together. When the two boards are correctly assembled, the CPU heatsink is brought into contact with the system's magnesium base casting to provide effective heat dissipation, as shown in Figure 1-1.

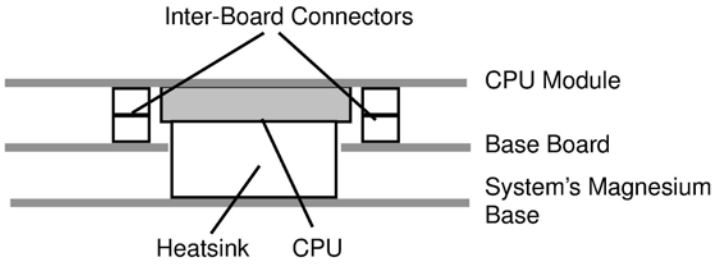


Figure 1-1 CPU Heat Dissipation

1.2.3 Microcontroller module

The microcontroller module is a small board which carries an Hitachi H8 microcontroller, the status display which is visible from the outside of the assembled system, and a number of programmable memory devices. It provides connections for the keyboard and pointing stick for which it provides control and for the Base board for which it provides system control and status monitoring functions.

1.2.4 Main display

The main display is housed within the system's lid along with an inverter board required to drive the display's backlight. Systems use either 9.4 inch 640 x 480 or 10.4 inch 800 x 600 color TFT display to provide a sharp image in a wide range of lighting conditions. The brightness of the backlight is controlled by the microcontroller and can be varied to suit the lighting conditions or can be dimmed or turned off when required to conserve battery power.

1.2.5 Other components

In addition to the main boards and display, the SPARCbook 3 system contains is a 2.5 inch 1.2 GB (or larger when available) SCSI hard disk drive assembled within a removable module. The drive can be removed from the SPARCbook 3 while the system is fully assembled, see your *SPARCbook 3 User Guide*.

1.3 System Architecture

The SPARCbook 3 system architecture is illustrated in Figure 1-2.

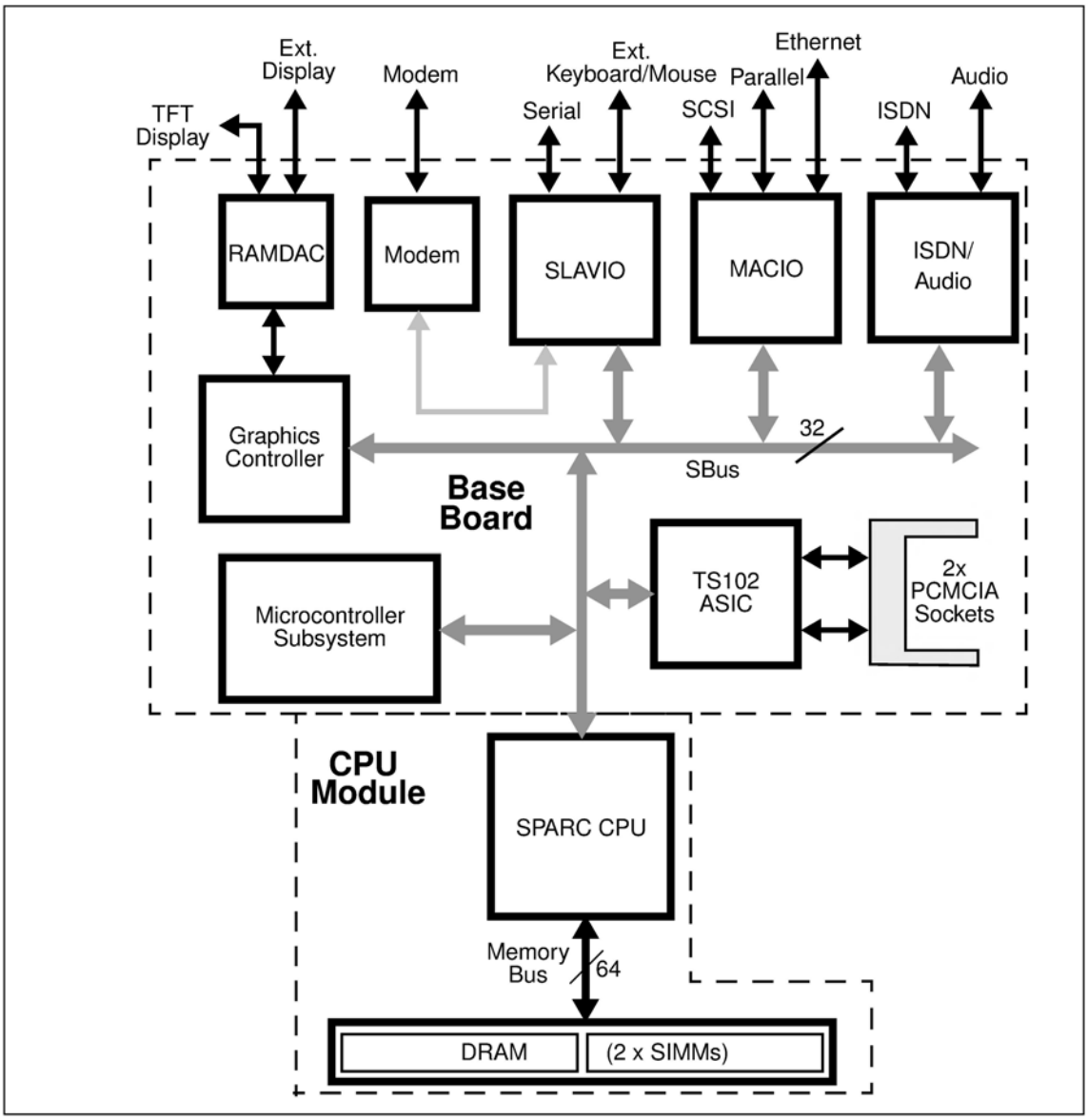


Figure 1-2 SPARCbook 3 Architecture

1.4 Processor

The CPU used in the S3TX is the TurboSPARC and the CPU used in the S3XPand S3GX is the microSPARC II.

The TurboSPARC CPU provides the following key features:

- SPARC compliant V8 Integer Unit core
- SPARC Reference Memory Management Unit
- Floating Point ALU
- FP-Muliply Unit
- FP Divide/Square Root Unit
- 16 Kbyte Instruction Cache
- 16 Kbyte Data Cache
- Secondary Cache Controller
- DRAM Controller
- SBus Controller, Master and Slave Interface.

The TurboSPARC implemented in the S3TX operates at 170 MHz and provides performance figures of 3.5 SPECint95 and 3.0 SPECfp95.

The microSPARC II CPU provides the following key features:

- SPARC-II compliant V8 Integer Unit (IU) core
- SPARC Reference Memory Management Unit (MMU)
- MEIKO Floating Point Unit (FPU)
- 16 Kbyte Instruction Cache
- 8 Kbyte Data Cache
- Memory Controller
- SBus Controller, Master & Slave Interface.

The microSPARC II implemented in the SPARCbook 3XP processor operates at 85 MHz and provides performance figures of 64 SPECint92 and 54.6 SPECfp92.

The microSPARC II implemented in the SPARCbook 3GX processor operates at 105 MHz and provides performance figures of 64 SPECint92 and 54.6 SPECfp92.

1.5 Main System Buses

The SPARCbook 3 architecture is based around three main buses conventional for SPARC-based workstations. These are the Memory bus which connects the CPU to the main memory; the SBus which connects the CPU to the major I/O devices; and the EBus.

1.5.1 Memory bus

The microSPARC II's integral memory controller is connected to the system DRAM directly via a 64 bit high speed memory bus. The microSPARC II provides direct addressing and control for the main memory, illustrated in Figure 1-3, providing the write enable signal and RAS and CAS lines. The smallest data movement is 64 bits; smaller transfers are carried out by using read-modify-write operations. Parity protection is provided by the CPU as 1 bit per word (32 bits) of data. SBus based master I/O devices are able to access the memory bus via the processor's SBus interface.

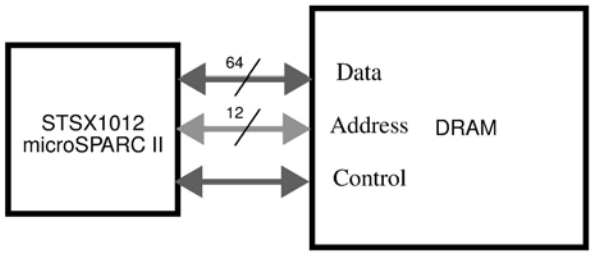


Figure 1-3 Main Memory/CPU Interface

1.5.2 SBus

The microSPARC II incorporates a complete SBus controller. The SBus connects the microSPARC II to the Weitek P9100 graphics controller, NCR89C105 SLAVIO, NCR89C100 MACIO and T725FC ISDN controller. See Figure 1-4.

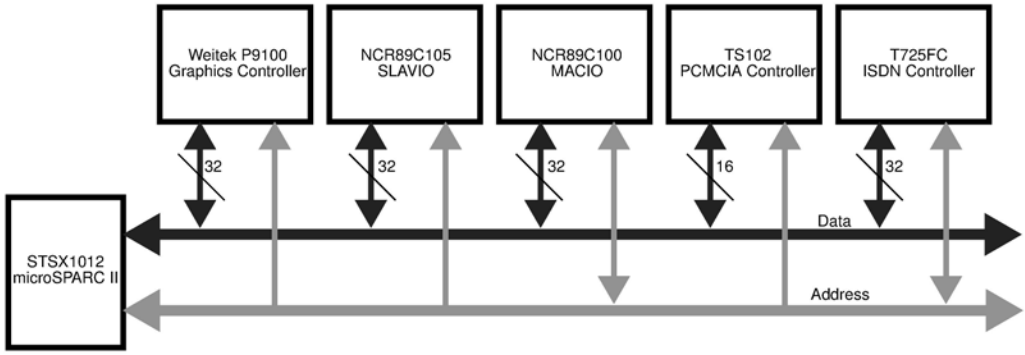


Figure 1-4 SBus Connected Devices

The microSPARC II provides an SBus Master and Slave interface which enables the I/O devices with integrated DMA capability to gain access to the main memory without encroaching unduly on processor bandwidth.

SBus master and slave operations can be single cycle or bursts, and dynamic bus sizing is supported (for single-cycle transfers). Master accesses by the microSPARC II to the SBus cannot be cached, and only double burst accesses are supported.

1.5.3 Ebus

The third system bus within the SPARCbook 3 is the Ebus. This is an 8-bit data bus driven by the SLAVIO. The SLAVIO divides the EBus address space into a number of regions by providing address generated EPROM, RTC/RAM and Generic chip select signals. The EBus interface of the SLAVIO is limited to a data bus and the chip select signals. The EBus address bus is driven by the TS102 ASIC to enable the CPU to gain access to the internal registers of devices on the EBus. Figure 1-5 provides a simplified illustration of the EBus architecture.

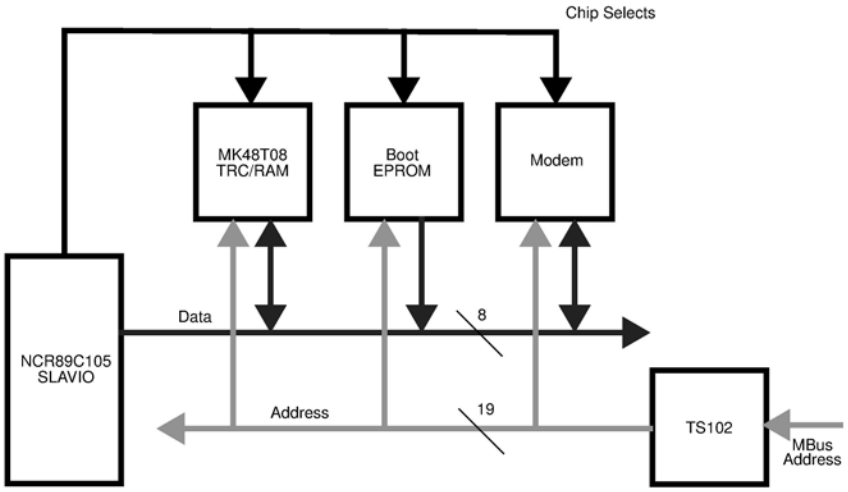


Figure 1-5 EBus Architecture

1.6 DRAM

The SPARCbook 3 provides two SIMM sites which support a range of different capacity modules. The SIMM sites accommodate 72-pin units, which must be fitted in matched pairs to provide a full width 64-bit data interface for the microSPARC II.

The SIMMs are each 33-bits wide (32 bits data and 1 bit parity), and are available in sizes of 8Mbytes x 33, 16Mbytes x 33, 32Mbytes x 33, and 64Mbytesx33. This gives a usable memory capacity of up to of 128. The fast processor clock speed used in SPARCbook 3 series computers requires the use of 60ns SIMMS.

1.7 Slow I/O Subsystem

The Slow I/O subsystem is managed by an NCR89C105 SLAVIO. The SLAVIO is an application specific integrated circuit (ASIC), designed as part of a two-chip set with the NCR89C100 MACIO, which provides two serial channels, keyboard and mouse ports, an interrupt controller and two counter-timers. The key features of the SLAVIO include:

- Two synchronous/asynchronous serial ports (85C30 SCC compatible)
- Keyboard/mouse ports (85C30 SCC sub-set)
- Two programmable counter-timers (500ns period)
- Interrupt controller
- 8-bit expansion bus (EBus) interface/controller for EPROM and 8-bit I/O devices
- Internal 82077 style floppy disk controller
- Miscellaneous I/O functions.

1.7.1 Serial Channels

The two serial ports are used to provide general purpose synchronous or asynchronous RS232 interfaces. The SCC channels A and B are connected to two 8-way mini-DIN connectors, which are marked as Serial Channel A and Serial Channel B on the I/O panel at the rear of the SPARCbook 3 system unit.

The two remaining serial channels provide the keyboard and mouse interfaces. These use transmit and receive data only. The TTL-level output signals connect directly to the combined Keyboard/Mouse mini-DIN connector on the I/O panel at the rear of the SPARCbook 3 unit.

For more information about these channels, refer to Chapter 4, “Serial Interface”. For information about the connections of these channels, refer to Appendix B, “Connector Information”.

1.7.2 Counter-Timers

The SLAVIO contains two counter timers. These are the System Counter and the Processor Counter/User Timer which are clocked at 2MHz and can provide counter-timer functions or periodic interrupts. The System Counter is 22 bits wide, and increments every 500ns.

The Processor Counter/User Timer can be used in either the same mode as the System Counter, or as a free running 54-bit timer. OpenBoot uses the Processor Counter as a system watchdog timer.

1.7.3 Interrupt Controller

The interrupt controller co-ordinates all on-board interrupt functions. These include all internal sources and a number of signals from elsewhere within the system. The microSPARC II uses a 4-bit priority encoded interrupt mechanism. The SLAVIO provides control and priority encoding for all of the system interrupt sources. For more information about the SPARCbook 3 interrupts system, see Section 3.2, “Interrupts”, on page 3-5.

1.7.4 EBus Interface and Controller

The SLAVIO provides an 8-bit bus called the EBus for a number of slower auxiliary devices. The EBus interface of the SLAVIO supports the Boot ROM; the real time clock and SRAM; and the system clock control port. This is illustrated in Figure 1-5.

1.8 Fast I/O Subsystem

The Fast I/O Subsystem includes the SCSI, parallel and network interfaces. These are controlled by the NCR89C105 MACIO. This device is a custom ASIC designed to be operated with the NCR89C100 SLAVIO as a two-chip set. The key features of the MACIO include:

- 53C90 style SCSI controller (Emulex FAS100A compatible)
- 7990 style Ethernet controller
- Parallel port interface
- Dual 64 byte FIFOs
- IEEE-1496 SBus DMA controller

This section describes each of these features.

1.8.1 SCSI Controller

The SCSI controller provides a 10Mbyte/sec 8-bit interface able to support up to eight SCSI devices. The SPARCbook 3 counts as one device, and the hard disk counts as a second, making it possible to add six external devices. The SPARCbook 3 is fitted with a 50-pin high density SCSI-2 connector.

For more information, refer to Chapter 5, “SCSI Controller”. For information about the connections, refer to Appendix B, “Connector Information”.

1.8.2 Ethernet Controller

The Ethernet controller provides a 10Mbit/sec networking interface. The design features an AT&T serial interface encoder to provide the standard AUI interface through a 26-way high density connector. An AUI cable and an Ethernet transceiver can be used to provide access to other physical Ethernet media, including Thick, Thin and Fiber-optic networks.

Note – The AUI interface is DC coupled, and any attachment units used with SPARCbook 3 must feature the network isolation function.

Ethernet data transfers are supported with the MACIO DMA function.

For more information, refer to Chapter 6, “Ethernet Interface”. For information about the connections, refer to Appendix B, “Connector Information”.

1.8.3 Parallel Port

The parallel port breakout cable supplied with the SPARCbook enables connection to a bi-directional Centronics style interface on a standard 25-way D-Type connector. Parallel port data transfers are supported with the MACIO DMA function.

For more information, refer to Chapter 10, “Parallel Interface”. For information about the connections, refer to to Appendix B, “Connector Information”.

1.8.4 FIFOs and DMA Controller

The FIFO and DMAC arrangement provided by the MACIO supports the SCSI, Ethernet and parallel interfaces. The DMAC performs burst transfers on the SBus whenever possible, supported by the FIFO circuitry, to minimize the I/O bandwidth consumed by simultaneous operation of these interfaces.

1.9 Graphics and Video Subsystem

The Graphics and Video Subsystem comprises the Weitek P9100 User Interface Controller, an IBM RGB528 palette DAC (RAMDAC), and a framebuffer provided by a 2MByte array of video RAM (VRAM) devices. All display interface configuration is carried out in software. There are no link adjustments.

The display interface supports the following display resolutions:

- 640 x 480 at 8, 16 or 24 bits per pixel
- 1024 x 768 at 8 or 16 bits per pixel
- 1152 x 900 at 8 bits per pixel
- 1280 x 1024 at 8 bits per pixel

1.9.1 Graphics Controller

The P9100 User Interface Controller provides the graphics control function. This device provides a 32-bit host interface and the following features:

- 32-bit VRAM interface and control signals
- RAMDAC interface and control signals
- Video timing control (up to 165MHz)
- 2D Graphics Accelerator
- Supports X window drawing mode
- Powerful graphics primitives

The Weitek Power 9100 User Interface Controller provides programmable display resolutions, supporting displays from 640 x 480 up to 1280 x 1024 pixels.

1.9.2 VRAM

The SPARCbook 3 has a 2Mbyte framebuffer comprising eight 256K x 8 devices. The VRAM is dual ported to provide a random access port for P9100 and a serial read-only port for the RAMDAC. The random access port is used by the P9100 and host to read and write picture information. The serial port is used to output pixel information to the RAMDAC. The RAMDAC provides timing signals for the serial data port of the framebuffer.

1.9.3 RAMDAC, Panel Driver and Video Clock Generator

The RGB528 combines a video clock generator, RAMDAC and flat panel control circuitry. The primary mode of operation supports the internal TFT panel and provides a display of 800 x 600 pixels (640x480 on some models)

in 256 colors from a palette of 262144. The RAMDAC can be software configured to support display resolutions of up to 1280 x 1024 in 256 colors (from a choice of 16M) on external monitors. 16 bit and 24 bit true color imaging modes are also supported on some configurations. The RAMDAC also provides numerous power-down features.

1.10 MK48T08 RTCRAM

The MK48T18 provides time-keeping facilities and incorporates 8 Kbytes of battery-backed non-volatile RAM. The device appears to software as an ordinary 8K x 8 RAM array. However, the uppermost 8 bytes provides an accurately updated real-time clock. The battery-backed RAM is used to store system configuration information, such as manufacturing data and Ethernet ID, via Tadpole's implementation of the OpenBoot firmware. The real time clock provides second, minute, hour, day, date, month and year information and a calibration register which allows adjustment of the RTC function in 2ppm steps. The device is accessed via the EBus port of the SLAVIO device.

1.11 ISDN and 16-Bit Audio Controller

The ISDN and Audio interface consists of two major components: the AT&T T7259 Dual Basic Rate ISDN Controller; and the Crystal Semiconductor Corporation CS4215 Multimedia Audio CODEC.

The T7259 has the following major features:

- Simultaneous terminal endpoint (TE) and network termination (NT)
- CCITT I.430/ANSI T1.605 support for 4 wire ISDN 2B+D basic access at the S/T reference point
- Multiframing support: S&Q channel operation
- Automatic synchronization of ISDN interfaces
- On-chip HDLC formatter
- On-chip 16-channel DMA address generator and linked list buffer manager
- Supports AT&T Concentration Highway Interface (CHI)
- Sbus master and slave interface

The ISDN controller combines a DMAC and data format converter (Parallel/Serial, Serial/Parallel and Time-Division-Multiplex). It has a number of DMA channels that can be allocated to support the ISDN or audio functions. The DMACs provide linked-list command support, and

FIFOs allow burst data transfers to be performed on the Sbus. Large amounts of ISDN or audio information can be moved to and from the Sbus with a minimum of processor overhead. The data is formatted by the ISDN controller into a composite digital serial stream (the Concentration Highway Interface). This connects to additional on-chip ISDN support circuitry, and to the external audio CODEC. The ISDN interface is implemented as a 2B+D Terminal Endpoint.

The Concentration Highway Interface of the ISDN circuitry provides a variety of different serial digital framing standards and data rates to the Audio CODEC. This supports a majority of the world standard Digital Audio formats. Typical configurations include high-quality stereo 16-bit 44.1KHz (CD), and telephony quality mono 8-bit 8KHz (ISDN).

The CS4215 Audio CODEC has the following major features:

- Stereo analog-to-digital and digital-to-analog conversion
- 4KHz to 48KHz sample rates
- 16-bit linear and 8-bit u-law or A-law coding
- Serial digital interface, compatible with AT&T CHI Concentration Highway Interface
- Microphone and line analog outputs

1.12 PCMCIA Controller

The PCMCIA controller is the Tadpole TS102. This device provides an interface between the SBus and the PCMCIA bus. It performs the additional function of providing a serial link between the CPU and microcontroller. The TS102 supports two PCMCIA Type I, II, and III cards or devices at a time. However, due to space constraints, the SPARCbook 3 unit is able to support one or two Type I and II cards, but only one Type III device.

1.13 Modem Interface

The modem interface consists of a microcontroller, a DSP device and a DAA. The microcontroller is the high-level controlling element and interfaces to the system bus. The DSP device performs all of the high speed data manipulation and data conversion. The DAA provides the line interconnect to the telephone network.

The modem supports a number of high level functions. It implements DTMF dialing, call progression, and is controlled via an enhanced "AT" command set. The data standards supported include V.22 bis, V.23, V.32, V.32 bis, V.42, and V.42 bis. In addition, the modem provides send and receive Fax capabilities to Group 3 standards (at up to 14,400bps).

1.14 Microcontroller Subsystem

The microcontroller subsystem provides system housekeeping support, freeing the main CPU. A Hitachi H8/337 microcontroller is used, offering the following features:

The microcontroller subsystem performs the following functions:

- Internal keyboard and pointing device control
- External keyboard and mouse control
- Serial communication channels to SLAVIO keyboard and mouse ports
- PSU and battery energy management
- System non-volatile storage (RTC and serial EEPROM)
- Environmental parameter control (display brightness and audio volume)
- LCD status display (2 x 16 character) control
- System reset control
- Power management control

The SPARC CPU

2

Processing power for all SPARCbook 3 models is provided by SPARC processors. In the case of he S3XP and S3GX microSPARC II is used; in the case of the S3TX TurboSPARC is used.

This chapter provides a general overview of SPARC CPU. For further information, please refer to Appendix A, “Further Information”.

2.1 SPARC Architecture Overview

The SPARC processor is a highly integrated device which provides the following features:

- SPARC compliant V8 Integer Unit core
- SPARC Reference Memory Management Unit
- MEIKO Floating Point Unit
- 16 Kbyte Instruction Cache
- 8 or 16 Kbyte Data Cache
- Memory Controller
- SBus Controller, Master and Slave Interface

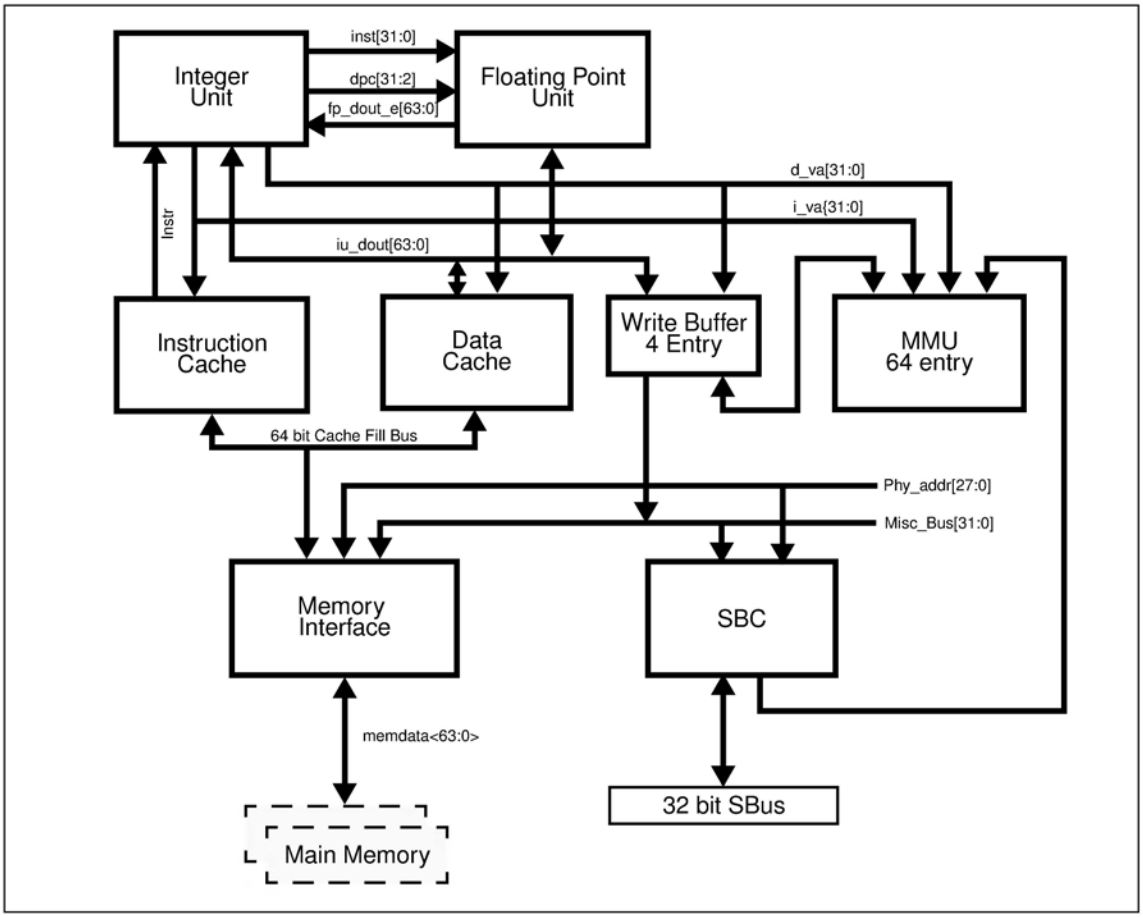


Figure 2-1 MicroSPARC-II Architecture

The SPARC CPU is a RISC (reduced instruction set computer) based processor which uses a simplified command set to carry out operations. It is able to execute most instructions within a single clock cycle.

The high performance of the SPARC CPU is enhanced by the ability of the floating point unit (FPU) to execute instructions simultaneously with the integer unit (IU), and by the provision of cache memory. The cache memory is a specialized area of fast (zero wait state) memory which allows many instructions and operands to be fetched locally by the CPU without it having to access the (comparatively slow) main memory.

2.2 Integer Unit

The IU is the main processing engine, executing all instruction groups except for floating point operations.

2.2.1 Pipeline

The SPA IU has a five-stage pipeline, receiving instructions which complete five cycles later. The five stages are fetch, decode, execute, cache access and write back. These stages are overlapped to allow a peak execution rate of one instruction per cycle.

The one instruction per cycle performance is supported by the IU's 32-bit data bus which interfaces directly with the instruction cache. If an instruction is in the cache, it is returned in the same cycle in which it was requested.

2.2.2 Instruction set overview

The integer instructions supported by the micro SPARC processor fall into the following basic categories:

- Load and Store Instructions
- Arithmetic, Logical and Shift Instructions
- Control Transfer Instructions
- Read/Write Control Registers Instructions.

The load and store instructions are the only instructions that cause the movement of data on the memory interface. They use two registers or a register and a constant to calculate the memory address involved. Halfword accesses must be aligned on 2-byte boundaries, word accesses on 4-byte boundaries, and doubleword accesses on 8-byte boundaries. These alignment restrictions greatly speed up memory access.

The arithmetic, logical and shift instructions compute a result that is a function of one or two source operands and then place the result non-destructively in a register.

The control transfer instruction category includes jumps, calls, traps, and branches. Control transfers are usually delayed until after execution of the next instructions so that the pipeline is not emptied every time a control transfer occurs, allowing compilers to optimize for delayed branching.

The read/write control register instructions include instructions to read and write the contents of various control registers. Generally, the source or destination is implied by the instruction.

INSTRUCTION	CYCLES
Call	1
Single Loads	1
Jump/Return	2
Double Loads	2
Single Stores	1
Double Stores	2
Taken Trap	3
Atomic Load/Store	2
SWAP	2
Integer Multiply	19
Integer Divide	39
All Others	1

Table 2-1 IU Cycles per Instruction

2.2.3 Traps and interrupts

The SPARC design supports a full set of traps and interrupts. They are handled by a table that supports 128 hardware and 128 software traps. Even though floating-point instructions can execute concurrently with integer instructions, floating-point traps are precise because the FPU supplies (from the table) the address of the instruction that failed.

The IU supports both asynchronous traps (interrupts) and synchronous traps (error conditions and trap instructions). Traps transfer control to an offset within the trap table. The base address of the table is specified by the Trap Base Register and the offset is a function of the trap type. Traps are taken before the current instruction causes any changes visible to the programmer and can therefore be considered to occur between instructions.

Interrupts from the peripheral devices in SPARCbook 3 are controlled and prioritized by the SLAVIO.

2.2.4 Memory protection

The SPARC design provides memory protection, essential for smooth multi-tasking operation. Memory protection prevents user programs from corrupting the system, other user programs, or themselves.

The IU supports a multi-tasking operating system by providing user and supervisor modes. Some instructions are privileged and can only be executed while the processor is in supervisor mode. Changing from user to supervisor mode requires taking a hardware interrupt or executing a trap instruction. This instruction execution protection ensures that user programs cannot accidentally alter the state of the machine with respect to its peripherals.

2.2.5 IU internal registers

The IU contains working registers (or r registers) and control registers. The r registers are used for storage by processes, and the control registers are used to track and control the state of the IU. The r registers are within a large register file containing one hundred and twenty 32-bit registers. Eight of these are global registers and are always accessible to a program, while the remaining registers are accessed through register windows. The way in which register windows are organized is shown in Figure 2-2.

The register file contains seven register windows, and each window contains twenty-four working registers. Each register window is divided into three sections called ins, outs, and locals, with eight registers in each section. Windows share ins and outs with adjacent windows. The outs of the previous window are the ins of the current window, and the outs of the current window are the ins of the next window. The windows form a circular stack where the outs of the last window are the ins of the first window.

A current window pointer (CWP) in the processor state register keeps track of the currently active window. The CWP is decremented when a program calls a subroutine that causes the processor to make accesses to the next

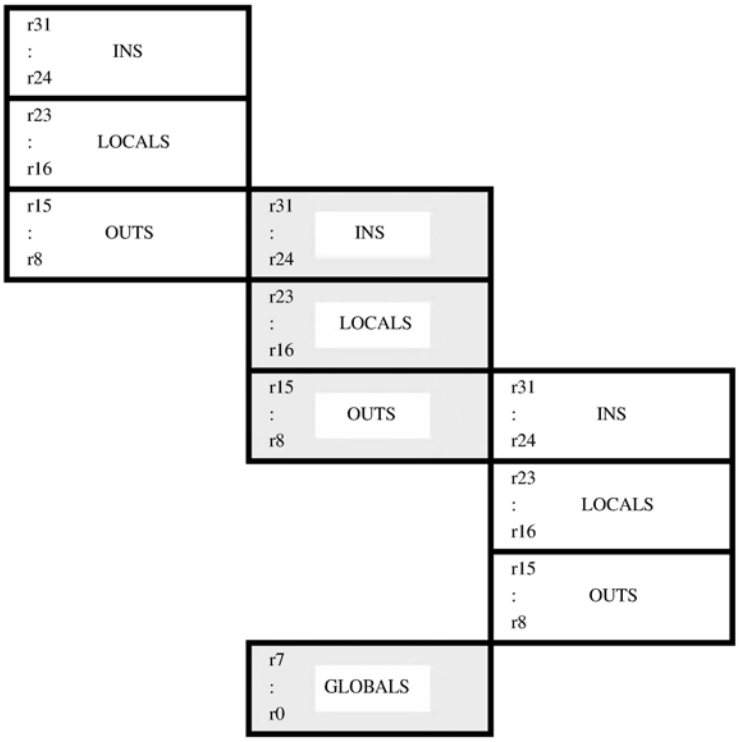


Figure 2-2 Window Register Organization

window, and is incremented when the processor returns to the previous window. Register windows can be marked as invalid in the WIM register, and interrupts can be enabled to signal when movement into an invalid window is caused by an instruction.

2.2.6 IU control registers

These include the Processor Status Register, the Window Invalid Mask Register, the Trap Base Register, the Y Register, and the Program Counter.

2.3 Floating Point Unit

The SPARC FPU is designed to execute all single- and double-precision SPARC Version 8 floating point instructions except fsmuld. All other FP instructions cause an unimplemented floating point operation trap. The FPU contains a 32x32-bit register file.

INSTRUCTION	MIN	TYP	MAX
fads	4	4	17
faddd	4	4	17
fsubs	4	4	17
fsubd	4	4	17
fmuls	5	5	25
fmuld	7	9	32
fdivs	6	20	38
fdivd	6	35	56
fsqrts	6	37	51
fsqrtd	6	65	80
fnegs	2	2	2
fmovs	2	2	2
fabss	2	2	2
fstod	2	2	14
fdtos	3	3	16
fitos	5	6	13
fitod	4	6	13
fstoi	6	6	13
fdtoi	7	7	14
fcmps	4	4	15
fcmpd	4	4	15
fcomes	4	4	15
fcmped	4	4	15
unimplemented	3	3	3

Table 2-2 FPU Execution Timing

2.3.1 Floating Point Registers

The FPU contains thirty-two 32-bit floating-point f registers, as illustrated in Figure 3-8. These form a 32x32-bit register file. The contents of these registers are transferred to and from external memory under control of the IU using floating-point load/store instructions. Addresses and control signals for data accesses during a floating-point load or store are supplied by the IU, while the FPU supplies or receives the data.

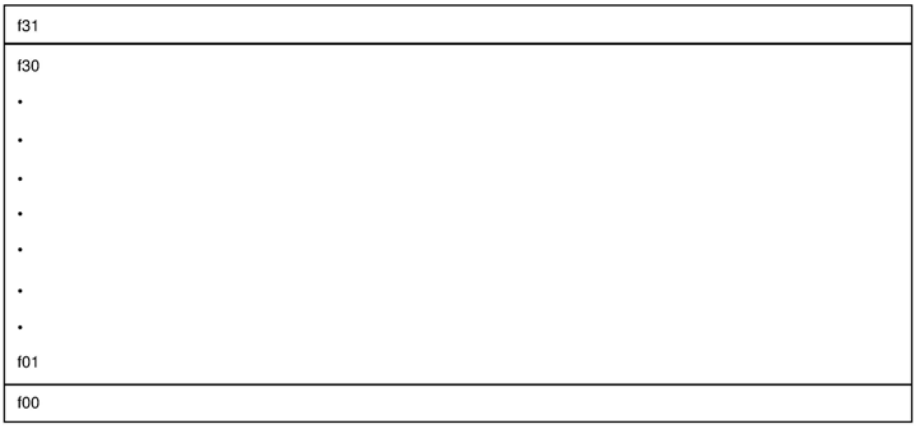


Figure 2-3 f Registers

Although the FPU operates concurrently with the IU, a program containing floating-point computations generates results as if the instructions were being executed sequentially.

A single f register is able to store one single-precision operand. Two registers are required to hold a double precision operand.

2.4 Cache Controller and Memory Management Unit

The SPARC's integral Memory Management Unit (MMU) provides virtual to physical address translation, memory protection and arbitration between I/O, data cache and TLB references to physical memory.

2.4.1 Translation lookaside buffer

The Memory Management Unit (MMU) conforms to the standard SPARC architecture definition for memory management.

The MMU provides virtual to physical address translation using a translation lookaside buffer (TLB). An entry in the TLB has the fields shown in Figure 2-4.

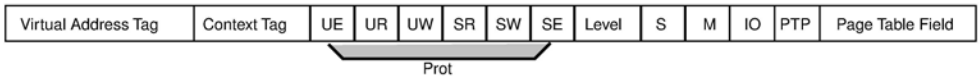


Figure 2-4 TLB Entry

TLB Entry Fields

Virtual Address Tag
: represents the most significant 20 bits, VA(31:12), of the virtual address.

Context Tag
: is compared with the 6-bit context number in the context register written by memory management software.

Prot
: Six protection bits in each TLB entry represent the decoded ACC bits from the matching PTE. These are: User Rd, Wr, Ex, and Supervisor Rd, Wr, Ex.

Level
: This 3-bit field is used to allow the proper tag match of region and segment PTEs. I/O PTEs and PTPs 1 will have this field set to use index 1, 2, and 3. The most significant bit also serves as the TLB Valid Bit because it is set for any valid PTE, I/OPTE or PTP.
  000 = none
  100 = Index 1 –VA(31:24)
  110 = Index 1,2 – VA(31:18)
  111 = index 1,2,3 – VA(31:12)

S
: Supervisor – This bit disables matching of the context field of a page in supervisor level.

M
: This is set to 1 if page is written.

I/O PTE
: This bit, when ‘0’, indicates that an I/O PTE is contained in this entry.

PTP — This bit, when set, indicates that a page table pointer is contained in this entry

Page Table Field — This field can contain a PTE, PTP or I/O PTE. It can be read and written using ASI 0x06 (25 bits).

Page Table Entry

The PTE defines the physical address of a page and its access permission. It contains the following information:

Bits 31:27 — Reserved - always write 0.

Bits 26:8 — PPN – Physical Page Number, which provides the upper 19 bits (30:12) of the 31-bit physical address of the page.

Bit 7 — Cacheable. This bit when set indicates that a page is cacheable.

Bit 6 — Modified. This bit is set when the page is written to.

Bit 5 — Always 1 for a PTE in the TLB. For a PTE in physical memory, this bit is set when the page is accessed.

Bits 4:2 — ACC – Access Permissions. This field indicates whether access is permitted for the transaction being attempted. The Address Space Identifier (ASI) determines whether the an access is an instruction or data access, and whether it is a user or supervisor access.

ACC	User	Supervisor
000	Read Only	Read Only
001	Read/Write	Read/Write
010	Read/Execute	Read/Execute
011	Read/Write/Execute	Read/Write/Execute
100	Execute Only	Execute Only
101	Read Only	Execute Only
110	No Access	Read/Execute
111	No Access	Read/Write/Execute

Table 2-3 Page Table Access Permissions

Bits 1:0 — ET – These are set to 10 indicate an entry type of PTE.
00 = Invalid
01 = Page Table Pointer
10 = Page Table Entry
11 = Reserved in Page Tables

Page Table Pointer

The PTP contains the physical address of a page table in memory, and can be found in the context table, or in a level 1 or 2 page table. Page tables are loaded into the TLB during tablewalks, and are removed by tablewalks or flushing.

Bits 31:27	Reserved – always write 0
Bits 26:4	PTP – Physical address of the next page table.
Bits 3:2	Reserved – always 00.
Bits 1:0	ET – Entry type, contains 01 to denote a PTP.

I/O Page Table Entry

The I/O PTE defines the physical address of a page and its access permission.

Bits 31:27	Reserved – always write 0
Bits 26:8	PPN – Physical Page Number, which provides the upper 19 bits (30:12) of the 31-bit physical address of the page.
Bits 7:3	Reserved – always contains 0000
Bit 2	Writeable 0 = read only 1 = read/write
Bit 1	This bit is set to 1 when the I/OPTE is valid
Bit 0	This bit is to be written as zero (WAZ) in the Memory I/O Page Table by software.

2.4.2 Address translation

During an access by the IU, the virtual address supplied by the IU and the contents of the context register are compared with the virtual section of all TLB entries. When a match is found (or a "hit" occurs), the Physical section supplies the address of a page in memory, or a pointer to a page table in physical memory. Virtual address bits A(11:00) from the IU are passed through unchanged to supply a byte offset. Each hit TLB entry is automatically checked for memory protection attributes and violations are reported to the IU as memory exceptions.

If the virtual address from the IU does not match an entry in the TLB, the MMU automatically performs a search (or table walk) through a translation table in main memory to obtain an address translation. The translation table forms a tree structure in the main memory. An example of this is illustrated in Figure 2-5.

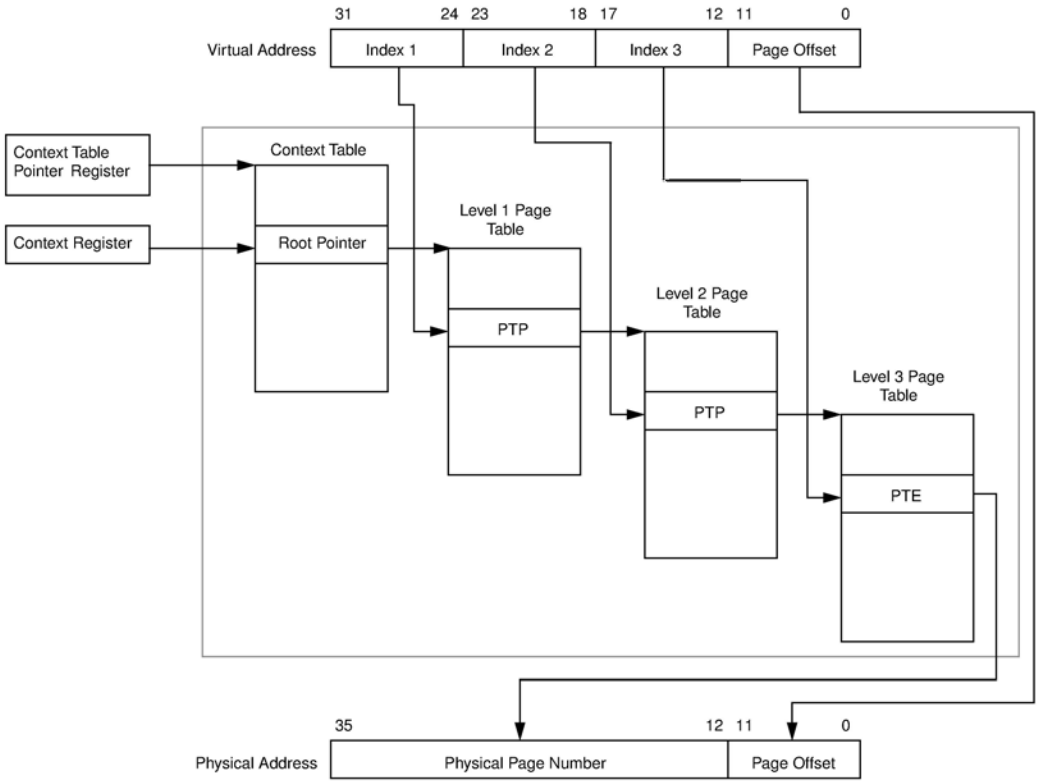


Figure 2-5 A Three Level Table Walk In Memory

The Context Table Pointer register provides a pointer to the context table, and the context register provides an index to the Root Pointer, which in turn points to a level 1 page table. Index 1 from the virtual address selects an entry within Level 1 pointing to a level 2 page table, where Index 2 selects a pointer to a level 3 table. Index 3 then selects one of the entries in the level 3 table which should point to a 4Kbyte memory page. When a page table pointer (PTP) is encountered within the tables, the search continues to the next lower level. If a page table entry (PTE) is found, the search is terminated and the entry is stored in the TLB. If no PTE is found at all, a synchronous fault exception is signalled to the IU. The PTE provides a pointer to a physical page while, in the case of a three level table walk, the lower 12 bits of the virtual address provides an offset.

The level at which a table walk terminates (that is, a PTE is found) is related to the size of addressing region associated with the entry. A table walk which finds a PTE in the context table corresponds to a region of 4Gbytes. A PTE corresponds to: a 16Mbyte region in level 1; a 256Kbyte region in level 2; or a 4Kbyte region in level 3. The virtual address bits not used to index table entries are used to supply the offset address within the page.

A table walk which uses three page table levels is shown in the illustration in Figure 2-5. In this example A(31:12) from the virtual address are used to index the page tables, and A(11:0) supply an offset address into the selected memory page.

Figure 2-6 shows a table walk which terminates at level 2. In this case A(31:18) are used as index bits, and A(17:0) provide an offset address into the selected 256 Kbyte page.

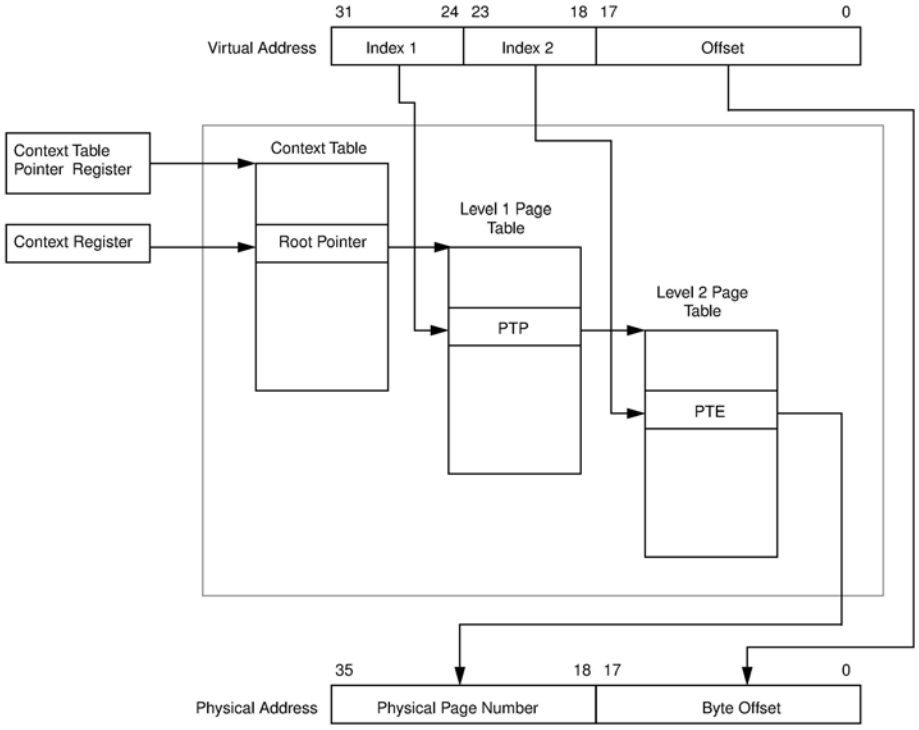


Figure 2-6 A Two Level TableWalk In Memory

2.5 Memory Interface

The SPARC provides a 64-bit memory interface which supports up to 128Mbytes of system memory. The memory is composed of four banks of up to 32Mbytes each. Different density devices are supported, allowing the SPARCbook 3 to be fitted with a range of memory size options.

The SPARC's memory interface provides a 64-bit data bus with 2-bit parity (1 bit parity for each 32 bit word). The memory interface incorporates a DRAM refresh controller.

The minimum memory access width is 32 bits; 8-bit and 16-bit write accesses require read-modify-write operation and correct 32-bit boundary alignment.

2.6 Instruction Cache

The integral instruction cache is a 16Kbyte physically tagged cache. The instruction cache is organized as 512 lines of 32 bytes each.

2.7 Data Cache

The data cache is a 8Kbyte direct mapped physically tagged write through cache with no write allocate. It is organized as 512 lines of 16 bytes each.

Data cache read and write hits take no extra pipe cycles, except for doubleword operations. There are two store buffers which hold data being stored from the IU or FPU to memory or other physical devices. The store buffers are 32-bit registers.

2.8 SBus Controller

The SPARC incorporates an SBus Controller which provides a master and slave interface used to access I/O devices. It connects the CPU core directly to the SBus and allows DMA devices to access the main DRAM located on the SPARC's memory bus. The SBus controller provides 32-bit data, SBD(31:00), and 28-bit physical address, SBA(27:00), interface to other devices in SPARCbook 3. The SBus controller also provides five SBus Slave Select lines, SBSEL(4:0).

2.8.1 Programmed I/O

Programmed I/O transactions consist of an SBus slave cycle only, with address translations being carried out before bus acquisition. The processor executes loads and stores to transfer data between it and devices on the SBus (in I/O Space). The SBus Controller performs write posting during processor writes, allowing processing to continue while the SBus transaction is completed. During reads, processing is stalled until the data becomes valid at the end of the SBus transaction.

2.8.2 DVMA

A direct virtual memory access consists of a translation cycle followed by a slave cycle. During the translation cycle, a master places a virtual address on the SBus data bus. The SPARC's MMU provides a translated physical address on the SBus.

Memory Map and Interrupts

3

This chapter describes the addressing architecture and interrupt architecture of SPARCbook 3.

The SLAVIO incorporates an interrupt controller and is used to coordinate all on-board interrupts. These include interrupts from devices on the board and interrupts from SLAVIO internal source.

3.1 Address Map

The SBus controller contained in the MicroSPARC partitions the SPARCbook 3's memory map into a region for the main memory plus five physical address regions of 256Mbytes each for the SBus. The resulting memory map of the SPARCbook is shown in Table 3-1.

Address Range (Hexadecimal)		Region	SBus Rsource	Size (Mbytes)
70000000	7FFFFFFF	SBus Slot 4	MACIO/SLAVIO	256
60000000	6FFFFFFF	SBus Slot 3	Not allocated	256
50000000	5FFFFFFF	SBus Slot 2	ISDN Controller	256
40000000	4FFFFFFF	SBus Slot 1	TS102 PCMCIA Controller	256
30000000	3FFFFFFF	SBus Slot 0	P9100 Graphics Controller	256
20000000	2FFFFFFF	Unmapped	-	256
10000000	1FFFFFFF	I/O MMU Control Space	-	256
08000000	0FFFFFFF	Unmapped	-	128
06000000	07FFFFF	DRAM Bank 3	-	32
04000000	05FFFFF	DRAM bank 2	-	32
02000000	03FFFFF	DRAM Bank 1	-	32
00000000	01FFFFF	DRAM Bank 0	-	32

Table 3-1 Main Memory Map

3.1.1 MACIO and SLAVIO Space (SBus Slot 4)

SBus slot 4 is sub-divided to provide access to the MACIO and SLAVIO. These two devices are designed to operate as a pair, and each device has a direct SBus connection.

The MACIO device incorporates DMA controllers, an Ethernet controller, a bi-directional parallel interface, and a SCSI bus controller. The DMA controllers provide support for the Ethernet, parallel and SCSI interfaces, including FIFO support, and allow burst transfers to be performed on the SBus.

Connected to the SLAVIO but isolated from the SBus are the Boot ROM, the O/S ROMs, the RTC and NVRAM, modem and Audio devices.

The map for SBus Slot 4 is as shwon in Table 3-2.

Address Range (Hexadecimal)		Resource	Size (Mbytes)
78000000	7FFFFFFF	MACIO	128
70000000	77FFFFFF	SLAVIO	128

Table 3-2 SBus Slot 4 Memory Map

The MACIO provides processor accessible control ports for DMA related operations, for parallel port operations, for SCSI operations and for network interface control.

Base Address (Hexadecimal)	Port
7C800000	Parallel Port Registers
78C00000	Network Controller Registers
78800000	SCSI Registers
78400000	DMA Control and Test Registers
78000000	Internal ID Register

Table 3-3 MACIO Port Locations

The function and access sizes of these locations are discussed in the relevant chapter for each interface.

The SLAVIO contains a number of internal devices. These are a floppy controller (not used in SPARCbook 3), two serial controllers (SCCs), counter/timers and an interrupt controller. The SLAVIO also supports access to external devices via the EBus.

The internal interface devices each present an independent interface to the host. Details of these control interfaces are provided in the relevant chapter for each interface.

Within their assigned spaces in the SLAVIO's address space, the RTC/NVRAM and Boot PROM are mapped repeatedly. In the case of the RTC/NVRAM this means that there are 256 images.

The SLAVIO memory map is shown in Table 3-4.

Offset Range (Hexadecimal)		Resource	Accessibility
1F00000	1FFFFFF	SLAVIO System Controller and Status Register	Word
1E00000	1EFFFFF	SLAVIO Interrupt Controller	Word
1D00000	1DFFFFF	SLAVIO Counter/Timers	Word, Doubleword
1C00000	1CFFFFF	SLAVIO Reserved	
1B00000	1BFFFFF	SLAVIO Modem Register	Byte
1A00000	1AFFFFF	SLAVIO Diagnostic Message Register	Byte
1900000	19FFFFF	SLAVIO Auxiliary I/O Registers	Byte
1800000	18FFFFF	SLAVIO Configuration Registers	Byte
1500000	17FFFFF	SLAVIO Reserved	
1400000	14FFFFF	SLAVIO Floppy Controller	Byte
1380000	13FFFFF	Paged FLASH O/S ROM	Byte
1304000	137FFFF	Echoes of Audio and Auxiliary Port	Byte
1302000	1303FFF	Auxiliary Control/Status Port	Byte
1300000	1301FFF	Audio Device	Byte
1204000	12FFFFF	Echoes of RTC/RAM and Diagnostic LEDS	Byte, Halfword, Word
1202000	1203FFF	RTC/RAM and Diagnostic LEDs	Byte, Halfword, Word
1200000	1201FFF	RTC/RAM	Byte, Halfword, Word
1000000	11FFFFF	SLAVIO Keyboard/Mouse/Serial	Byte
0800000	0FFFFFF	Echo of Boot and O/S ROMs	Read Only, Byte, Halfword, Word
0700000	07FFFFF	O/S ROM 4	Read Only, Byte, Halfword, Word
0600000	06FFFFF	O/S ROM 3	Read Only, Byte, Halfword, Word
0500000	05FFFFF	O/S ROM 3	Read Only, Byte, Halfword, Word
0400000	04FFFFF	O/S ROM 1	Read Only, Byte, Halfword, Word
0100000	03FFFFF	Echoes of Boot ROM	Read Only, Byte, Halfword, Word
0000000	00FFFFF	Boot ROM	Read Only, Byte, Halfword, Word

Table 3-4 SLAVIO Memory Map

3.1.2 DRAM

The DRAM address multiplexers support SIMM units with up to 11 x 11 or (12 x 10) Row/Column multiplex (1Mbit, 4Mbit, and most 16Mbit devices). Each SIMM module can contain one or two banks of memory. SIMM sizes up to 64Mbytes are supported, with a maximum of 32Mbytes per bank.

The memory maps for the different configurations are based on the principle that the individual bank selects (RAS signals) are derived from the address decodes for fixed 32Mbyte memory segments.

Table 3-8 summarizes the possible memory implementations when using identical SIMM pairs.

System Capacity	SIMMs			Address Map (Where memory appears)
	Qty	Organization*	Architecture of Each	
16Mbytes	2	2M by 33	Two banks of 4Mbit	At 0 & 32Mbytes for 8Mbytes each
32Mbytes	2	4M by 33	One bank of 16Mbit	At 0 for 32Mbytes
64Mbytes	2	8M by 33	Two banks of 16Mbit	At 0 for 64Mbytes
128 Mbytes	2	16M by 33		

Table 3-5 DRAM Mapping

*Figures in the **Organization** column can be either by 33 or by 36.

3.2 Interrupts

Interrupts from devices within SPARCbook 3 are signaled to the microSPARC as a 4-bit priority encoded value on S_IRL(3:0). Control of interrupts and prioritization is carried out by the SLAVIO. The microSPARC provides a structure of traps which supports fifteen external interrupts.

The SLAVIO contains interrupt control and mask registers which are used to enable and clear hardware interrupts from devices within the SPARCbook 3 system as well as interrupts from devices internal to the SLAVIO and MACIO. The SLAVIO also provides software generated interrupts on each of fifteen levels.

The interrupts in SPARCbook 3 can be considered as belonging to three categories: SLAVIO interrupts, MACIO interrupts, and SBus interrupts. The SLAVIO interrupts include those from the internal serial ports and timers and the software interrupts. The MACIO interrupts include the Ethernet, SCSI and parallel port interrupts.

Table 3-6 provides a summary of the interrupt request sources within SPARCbook 3.

Level	Source		
	SLAVIO	**MACIO**	**SBus**
0	No Interrupts Pending		
1	SOFTINT 1	-	-
2	SOFTINT 2	-	IRQ1
3	SOFTINT 3	Parallel Port	IRQ2
4	SOFTINT 4	SCSI	-
5	SOFTINT 5	-	IRQ3
6	SOFTINT 6	Ethernet	-
7	SOFTINT 7	-	IRQ4
8	SOFTINT 8	-	-
9	SOFTINT 9	-	IRQ5
10	SOFTINT 10, Sys Counter/Timer	-	-
11	SOFTINT 11, Floppy	-	IRQ6
12	SOFTINT 12, Serial/KBD/MSE	-	-
13	SOFTINT 13, Audio	-	IRQ7
14	SOFTINT 14, Proc Counter/Timer	-	-
15	SOFTINT 15	Async. Hardware Errors. User Power-Down Request	-

Table 3-6 SPARCbook Interrupts

3.2.1 Interrupt Control

The SLAVIO provides a number of interrupts status and control locations. The processor group provides interrupt pending information and control over software interrupts. The system group provides enable and clearing control over the individual hardware interrupt requests.

Address (Hexadecimal)	Device or Register	Access
71E00000	Processor Interrupt Pending	R
71E00004	Processor Clear Pending	W
71E00008	Processor Set Software Interrupt	W
71E10000	System Pending Interrupt Register	R
71E10004	System Interrupt Target Mask Register	R
71E10008	System Interrupt Target Mask Clear	W
71E1000C	System Interrupt Target Mask Set	W

Table 3-7 Interrupt Control Registers

Processor Group

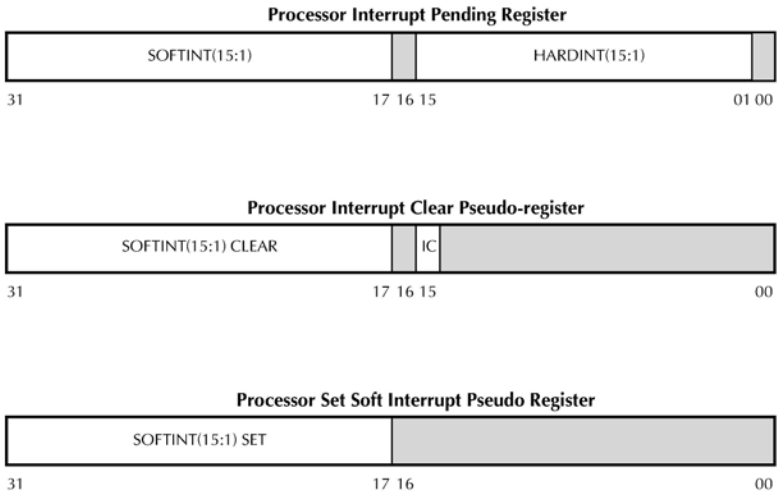


Figure 3-1 Processor Interrupt Registers

SOFTINT(15:1)
Software Interrupt

HARDINT(15:1)
Hardware Interrupt

IC Interrupt Level 15 Clear

Writing a '1' to any of the SOFTINT bits or IC bit in the Interrupt Clear pseudo-register clears the associated interrupt.

The Set Soft Int pseudo-register is used to generate software interrupts. Setting a bit in this register sets the associated bit in the pending register and triggers an interrupt request on the appropriate level.

System Group

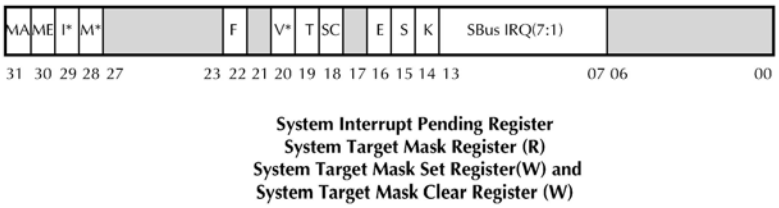


Figure 3-2 System Interrupt Registers

Bit 31 MA – Mask all Interrupts (reserved in the System Interrupt Pending Register)
1= disable all interrupts

Bit 30 ME – Module Error

Bit 29 Reserved on SPARCbook

Bit 28 Reserved on SPARCbook

Bits 27:23 Reserved

Bit 22 F – Floppy Interrupt

Bit 21 Reserved

Bit 20 Reserved on SPARCbook

Bit 19 T – Level 10 Counter/Timer

Bit 18 SC – SCSI Interrupt

Bit 17	Reserved
Bit 16	E – Ethernet Interrupt
Bit 15	Bit S – Serial Port Interrupt (SLAVIO)
Bit 14	K – Keyboard/Mouse Interrupt
Bits 13:07	SBus IRQ (7:1)
Bits 06:00	Reserved

3.3 NCR89C105 SLAVIO Configuration Control

3.3.1 SLAVIO Configuration Register

The NCR89C105 has several software-programmable options which are controlled by its configuration register. This register is located at 0x70180000.

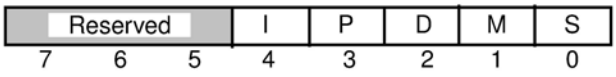


Figure 3-3 The SLAVIO Configuration Register

A system reset clears all of the SLAVIO configuration bits to 0.

Field definitions:

Bits 7:5	Reserved - always read 0.
Bit 4	I - Modem Ring Interrupt Enable. When this bit is set to 1, the modem RI interrupt generation is activated (see also Bit 1, and the description of the modem register). When this bit is cleared, modem interrupts are not generated, regardless of the state of the M bit (Bit 1) or the MSI_IRQ_input.
Bit 3	P - Power Fail Detect Enable. When this bit is set to 1, a low on the PFD_input causes a module error (error 15) interrupt. The interrupt is visible (and clearable) in AuxIO register 2. When this bit is clear, the PFD_input is ignored.

Bit 2 — D - Density Select Source (1 = 82077 density select, 0 = 82077 motor enable #2). - This bit determines which signal drives the external density select pin (FPY_DENSEL).

Bit 1 — M - Modem Ring Select. When this bit is set to 1, a low on the MSI_IRQ_input causes a level 15 MSI interrupt. When this bit is cleared, a transition causes a modem ring indicate interrupt (SBus level 5). Either a low or high transition can cause an interrupt in this mode, depending on the Edge Select bit in the modem register. The unused input is held in its inactive state.

Bit 0 — S - SuperSPARC mode (1 = SuperSPARC, 0 = MicroSPARC-II). This bit determines the function of several multiplexed input pins (the NCR89C105 unsufficient pins to support all functions concurrently). The muxed pins are shown in Table 3-8:

Pin	microSPARC Use	SuperSPARC Use
ser_rtxc_b	ser_rtxc_b_	emc_irq_
iu_error_	iu_error_	video_irq

Table 3-8 microSPARC-II/SuperSPARC Muxed Pins

In microSPARC-II mode, the SuperSPARC interrupts are inactive.

3.3.2 Diagnostic Messages

The Diagnostic Message Register is an 8-bit read/write register provided for diagnostic use. Accesses to this register change the value that is stored in it.

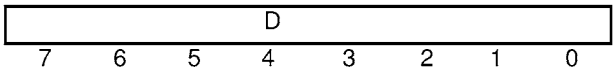


Figure 3-4 Diagnostic Message Register

Field definitions:

Bits 7:0 — (D) - Diagnostic value. This value is read/writeable, and is preserved across resets.

Note

All bits in the Diagnostic Message Register are unaffected by system resetbut contains random information after a power-on reset.

3.3.3 Miscellaneous System Functions

The NCR89C105 contains two 8-bit Auxiliary I/O Registers: one dedicated to controlling system power-down; and one used to support several hardware functions that do not fit well elsewhere. These registers are located at physical addresses 0x7190000 (aux1) and 0x719100000 (Aux 2).

LED/Floppy (Aux 1)Register

Figure 3-5 Auxiliary I/O Register 1

A system reset clears all output bits (Bits 3, 2, 1, and 0) to 0. The FPY_DENSENSE chip pin controls input bit (Bit 5). The unused bits (Bits 4, 6, and 7) are unaffected by writes, and always read 0.

Bits 7:6 Reserved - always read 0.

Bit 5 D - Floppy Density Sense (not used).

Bit 4 Reserved - always read 0.

Bit 3 E - Link Test Enable. This bit is directly reflected in the LINK_TEST_EN pin. It controls the AT&T 7213 LTE pin.

Bit 2 M - Monitor/Mouse Mux. This bit is directly reflected on the MON_MSE_MUX pin.

Bit 1 T - Terminal Count (1 = TC). Writing a 1 to this bit sends a 4 SBus clock wide TC pulse to the 82077 floppy controller. This bit is self-clearing; it always reads 0. Writing a 0 has no effect.

Bit 0 L (1 = on, 0 = off). - This bit controls the system LED on the front panel.

Power Down Control (Aux 2) Register

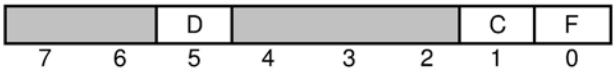


Figure 3-6 Auxiliary I/O Register 2

All Bits are cleared to 0 on system reset.

Bits 7:6 Reserved. These bits should be masked, and their values should be discarded by the software. These bits can be read and written to, but the values have no meaning or effect.

Bit 5 D - Power Failure Detect (1 = power fail). When the power fail detect signal from the power supply is low, this bit is set, and a *module error* interrupt is generated (this is a level 15 interrupt). Writing a 1 to Bit 1 of this register, or disabling PFD in the Config register clears this bit. This bit is also set if BUFP or PWROK are set to 0.

PFD* input is ignored if Bit 5 is disabled in the Config register.

Bits 4:2 Reserved. These bits should be masked, and their values should be discarded by the software. These bits can be read and written to, but the values have no meaning or effect.

Bit 1 C - Clear Power Fail Detect (1 = clear). This bit clears the interrupt generated by PFD_ and Bit 5 of this register. Writing 0 has no effect.

Bit 0 F - Power off (1 = off). This bit is reflected in the power output pin. When this bit is set to 1, the power supply turns off.

Modem Register

The NCR89C105 can directly support the RI (Ring Indicate) bit output of a modem, when it is configured for modem use. This mode uses the MSI_IRQ_ input pin for sensing modem RI. When the Modem mode and the Modem interrupt are enabled in the Configuration Register, the NCR89C105 generates an SBus level 5 interrupt on RI transitions.

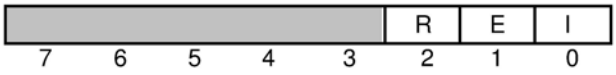


Figure 3-7 Modem Register

A system reset clears Bits 1 and 0 to 0. The input bit 2 is controlled by the MSI_IRQ_ chip pin. Bits 7:3 are unaffected by resets or writes and always read as 0.

Bits 7:3	Reserved - always read 0.
Bit 2	R - RI pin. This pin directly reflects the state of the MSI_IRQ_ pin (which is used for modem RI when in Modem mode). If this pin is low, then this bit is 0.
Bit 1	E - Edge Select. This bit selects which RI edge causes an interrupt. When this bit is cleared to 0, a 1 to 0 transition on the MSI_IRQ_ pin causes an interrupt. Toggling this bit after the first edge of an RI pulse is received, allows interrupts to be obtained on both edges of RI.
Bit 0	I - Modem RI Interrupt. This bit is set to 1 if a modem RI interrupt is pending. When this happens, an SBus level 5 interrupt is set. Writing a 0 to this bit clears the interrupt.

Serial Interface

4

The SPARCbook is equipped with two Z85C30 Serial Communications Controllers (SCC), both of which are contained within the SLAVIO. Although packaged in the SLAVIO, the two SCCs appear to software to be fully independent devices.

One is a fully functional SCC and one has reduced functionality being intended for keyboard and mouse interfacing.

The architecture of the SPARCbook serial interface is shown in Figure 4-1.

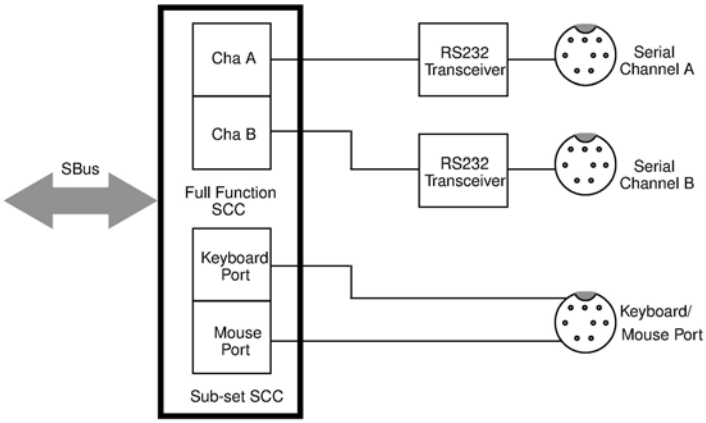


Figure 4-1 Serial Interface Architecture

4.1 Serial Channel Assignment

There are four serial channels on the SPARCbook, which are controlled by SCCs within the SLAVIO. Each channel provides a control interface. Serial Channels A and B are controlled via the Full-Function SCC; the keyboard and mouse port (which is Sun compatible) is controlled via the sub-set SCC.

Connector pinout details are provided in Appendix B, “Connector Information”.

4.1.1 Serial Interface Control

Table 4-1 summarizes the addresses of the four serial channels present on the SPARCbook.

Address (Hexadecimal)	Device Channel	Assignment on SPARCbook	Connector
SLAVIO Full-Function SCC			
71100000	TTYB Control	Serial Channel B	8-pin mini-DIN
71100002	TTYB Transmit and Receive Buffers		
71100004	TTY A Control	Serial Channel A	8-pin mini-DIN
71100006	TTY A Transmit and Receive Buffers		
SLAVIO Sub-Set SCC			
71000000	Mouse Control	Keyboard/Mouse Port	8-pin mini-DIN
71000002	Mouse Data		
71000004	Keyboard Control		
71000006	Keyboard Data		

Table 4-1 Serial Ports Addressing and Assignment

The SCC provides an 8-bit host interface. The SLAVIO internal registers appear on D(7:0); the data on the remainder of the bus during an SCC access is undefined.

A detailed programming discussion is beyond the scope of this manual; for further information, please refer Appendix A, “Further Information”.

4.2 SCC Registers

The SCC internal registers are accessed using a register pointer to perform selection. First, the register pointer bits in WR0 are programmed to specify the register to be accessed. Then, a read or a write is performed at the same address to transfer data into or out of the selected register. When the access to the selected register has been completed, the register pointer bits are reset to '0'.

Three pointer bits in WR0 allow access to the lower eight registers locations, but WR0 also contains 3-bit command word (D5:D3) in which the 'Point High' command, 001(bin), is required to gain access to the upper eight register locations. The receive buffer and transmit buffer, RR8 and WR8 respectively for each channel can be accessed in a single read or write operation.

4.2.1 Register functions

The Z82530 SCC contains fifteen write registers (WR0 to WR15) for each channel. WR8 is the transmit buffer and the remainder are used to configure the SCC for the required operation. Two registers (WR2 and WR9) are shared by both channels. Each channel also has nine read registers (RR0 to RR3, RR8, RR10, RR12, RR13 and RR15). RR8 is the receive buffer and the remainder provide status information.

Table 4-11 shows the full suite of registers present in the external 85C30 SCC and full-function internal SCC. The sub-set SCC does not present the full set of registers.

Register	Function
RR0	Transmit and Receive Buffer status, external status
RR1	Special receive condition status
RR2A, RR2B	Unmodified interrupt vector (Ch A only), Modified interrupt vector (Ch B only)
RR3A, RR3B	Interrupt pending bits (Ch A only), Null (Ch B only)
RR8	Receive Buffer
RR10	Miscellaneous status
RR12	Lower byte of baud-rate time constant
RR13	Upper byte of baud-rate time constant

Table 4-2 SCC Register Summary

Register	Function
RR15	External/Status interrupt information
WR0	CRC initialize, mode initialization, Register Pointer
WR1	Transmit and Receive interrupt, data transfer mode definitions
WR2	Interrupt vector (CH A and B)
WR3	Receive Parameters and controls
WR4	Transmit and receive parameters and controls
WR5	Transmit parameters and control
WR6	Synchronization character or SDLC address field
WR7	Synchronization character or SDLC flag
WR8	Transmit Buffer
WR9	Master Interrupt control and reset (CH A and B)
WR10	Miscellaneous controls
WR11	Clock mode control
WR12	Lower byte of baud-rate time constant
WR13	Upper byte of baud-rate time constant
WR14	Miscellaneous control
WR15	External/Status interrupt control

Table 4-2 SCC Register Summary (Continued)

WR0

Bits 7:6 Resets
- 00 = Null
- 01 = Reset Rx CRC checker
- 10 = Reset Tx CRC generator
- 11 = Reset Tx underrun/OEM latch

Bits 5:3 Commands
- 000 = Null
- 001 = Point High (reg 8-15 select)
- 010 = Reset external/status interrupts
- 011 = Send abort (SDLC mode)
- 100 = Enable int on next Rx character
- 101 = Reset Tx interrupt pending
- 110 = Error reset
- 111 = Reset highest IUS

Bits 2:0 Register 0-7 or 8-15 select

WR1	Bit 7	Wait/DMA request enable
	Bit 6	$\overline{\text{Wait}}$/DMA request
	Bit 5	Wait/DMA request on Rx/$\overline{\text{Tx}}$
	Bit 4:3	Receive interrupt control 00 = Interrupt disable 01 = Interrupt on first char or special condition 10 = Interrupt on all chars or special condition 11 = Interrupt on special condition only
	Bit 2	Parity as special condition enable
	Bit 1	Tx interrupt enable
	Bit 0	External interrupt enable
WR3 – Receive Parameters and Control	Bits 7:6	Bits/character 00 = 5 bits 01 = 7 bits 10 = 6 bits 11 = 8 bits
	Bit 5	Auto enables
	Bit 4	Enter hunt mode
	Bit 3	Receiver CRC enable
	Bit 2	Address search mode
	Bit 1	Sync character load inhibit
	bit 0	Receiver enable
WR4 – Transmit and Receive Parameters and Control	Bits 7:6	Clock mode 00 = x1 01 = x16 10 = x32 11 = x64
	Bits 5:4	Sync Mode 00 = 8-bit sync character 01 = 16-bit sync character 10 = SDLC mode 11 = External sync mode
	Bits 3:2	Stop mode

		00 = Sync modes enabled 01 = 1 stop bit/character 10 = 1.5 stop bits/character 11 = 2 stop bits/character
	Bit 1	Parity Even/$\overline{\text{Odd}}$
	Bit 0	Parity enable
WR5 – Transmit Parameters and Control	Bit 7	DTR
	Bits 6:5	Bits/character 00 = 5 bits 01 = 7 bits 10 = 6 bits 11 = 8 bits
	Bit 4	Send Break
	Bit 3	Transmitter enable
	Bit 2	SDLC/$\overline{\text{CRC-16}}$
	Bit 1	RTS
	Bit 0	Transmitter CRC enable
WR9 – Interrupt Control and Reset	Bits 7:6	Reset control 00 = no reset 01 = Reset Channel B 10 = Reset Channel A 11 = Force hardware reset
	Bit 5	Reserved (Write 0)
	Bit 4	Status High/$\overline{\text{Low}}$
	Bit 3	Master interrupt enable
	Bit 2	Disable lower chain
	Bit 1	No vector or interrupt acknowledge
	Bit 0	Vector include status
WR10 – Miscellaneous Controls	Bit 7	CRC Preset
	Bits 6:5	Mode

		00 = NRZ 01 = NRZI 10 = FM1 11 = FM0
	Bit 4	Active on poll
	Bit 3	Mark/$\overline{\text{Flag}}$ on idle
	Bit 2	Abort/$\overline{\text{Flag}}$ on idle
	Bit 1	Loop mode
	Bit 0	6-bit/$\overline{\text{8-bit}}$ Sync
WR11 – Clock Mode Control	Bit 7	RTXCb XTAL/$\overline{\text{No XTAL}}$
	Bits 6:5	Receive clock source 00 - 01 = Reserved 10 = Baud rate generator 11 = Reserved
	Bits 4:3	Transmit clock source 00 - 01 = Reserved 10 = Baud rate generator 11 = Reserved
	Bit 2	TRXC Enable (Write 0)
	Bits 1:0	Not used
WR14 – Miscellaneous Control	Bit 7:5	Clock Mode 000 = Null 001 = Enter search mode 010 = Reset missing clock 011 = Disable DPLL 100 = Clock source is BR generator 101 = Clock source is RTXCb 110 = FM mode 111 = NZRI Mode
	Bit 4	Local loopback
	Bit 3	Auto echo
	Bit 2	DTRb/Request function
	Bit 1	BR generator source
	Bit 0	BR generator enable

WR15 – External/Status Interrupt Control	Bit 7	Break/Abort Interrupt enable
	Bit 6	Tx Underrun/EOM Interrupt enable
	Bit 5	CTS enable
	Bit 4	Sync/Hunt enable
	Bit 3	DCD interrupt enable
	Bit 2	Reserved (write 0)
	Bit 1	Zero count interrupt enable
	Bit 0	Reserved (write 0)
RR0 – Status	Bit 7	Break/Abort
	Bit 6	Transmit underrun/EOM
	Bit 5	CTS
	Bit 4	Sync/Hunt
	Bit 3	DCD
	Bit 2	Transmit Buffer empty
	Bit 1	Zero count interrupt enable
	Bit 0	Receive character available
RR1 – Special Receive Condition	Bit 7	End of frame
	Bit 6	CRC/Framing error
	Bit 5	Receive overrun error
	Bit 4	Parity error
	Bit 3	Residue Code 0
	Bit 2	Residue Code 1
	Bit 1	Residue Code 2
	Bit 0	All sent
RR3 – Interrupt Pending	RR3 returns zero when read via channel B	
	Bits 7:6	00
	Bit 5	Channel A Rx
	Bit 4	Channel A Tx

	Bit 3	Channel A EXT/STAT
	Bit 2	Channel B Rx
	Bit 1	Channel B Tx
	Bit 0	Channel B EXT/STAT
RR10 – Miscellaneous Status	Bit 7	One clock missing
	Bit 6	Two clocks missing
	Bit 4	Loop sending
	Bit 1	On loop
	Other bits 0	
RR15 – External/Status Interrupt Status	Bit 7	Break/Abort interrupt enable
	Bit 6	Tx Underrun/EOM interrupt enable
	Bit 5	CTS interrupt enable
	Bit 4	Sync/Hunt interrupt enable
	Bit 3	DCD interrupt enable
	Bit 1	Zero count interrupt enable
	Other bits 0	

4.3 Baud Rate Clocks

A 19.66 MHz clock signal generated by the MACIO is used to derive a baud rate clock for all of the SCCs. This is divided by four before being fed to the SCCs at 4.915 MHz. The baud rate can be changed by loading different time constants (which can be different for each channel) into the time constant register. The clocking modes 1, 2, 16, 32 and 64 are supported.

The SCCs can be operated at baud rates of between 75 and 19200. This is controlled by the clocking mode and the contents of the time constant register. The value of the time constant can be determined using the following equation:

$$Tc = \frac{Clock}{2 \times (Mode) \times (Buadrate)} - 2$$

For example, to set a baud rate of 9600 using the x16 clocking mode on the SPARCbook, the following would be used:

$$Tc = \frac{4.915 \times 10^6}{2 \times 16 \times 9600} - 2$$

This gives a time constant of approximately 14 (0xD).

4.4 Handshakes

With the exception of the mouse and keyboard port, the serial channels all support handshaking. The handshake signals are controlled or monitored via the write registers and read registers of the associated SCC channel.

SCSI Controller

The MACIO incorporates an enhanced NCR53C90 Fast SCSI Controller (FSC), which supports SCSI-2 operations at up to 10 Mbytes/sec running synchronously. SCSI transfers are supported by the MACIO's integral DMA controller.

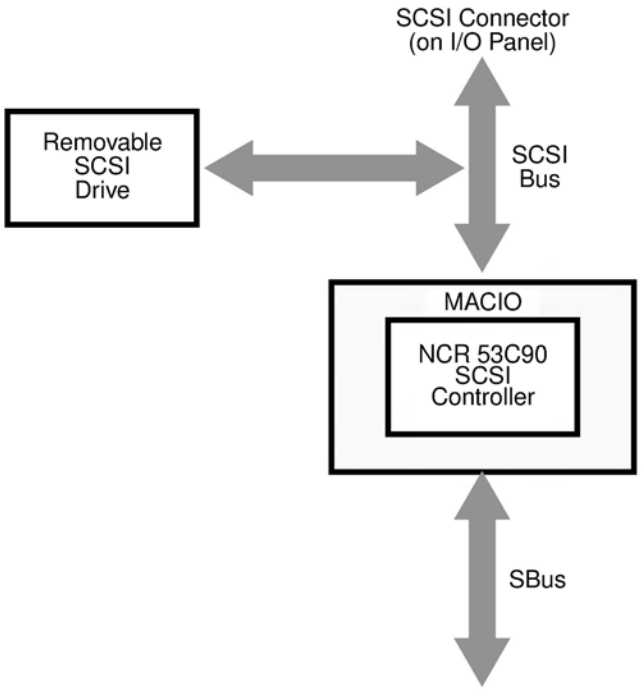


Figure 5-1 SCSI Architecture

5.1 Connecting SCSI Devices

The SPARCbook provides a 30-pin high density (Hosiden style) connector, to which you can connect the supplied SCSI adapter cable. This cable provides a 50-way high density SCSI-2 connector that can be used to make connection to SCSI devices.

5.2 NCR53C9X SCSI Controller

The FSC is a high performance superset of the NCR53C90 ASC and provides fast SCSI operation with additional commands and an additional configuration register. Speed of operation is enhanced by the MACIO's integral DMA controller.

The FSC provides full support of ANSI X3.131 SCSI and SCSI-2 standard.

5.2.1 53C9X register set

The SCSI controller internal registers appear from base address 0x78800000. Table 5-1 shows the address offset of the SCSI internal registers.

Address	Register	Access
0x78800000	Transfer Count Low Register	R/W
0x78800004	Transfer Count Mid Register	R/W
0x78800008	FIFO Data Port Register	R/W
0x7880000C	Command Port Register	R/W
0x78800010	Status Port Register	R
	Select / Reconnect Bus ID register	W
0x78800014	Interrupt Status Register	R
	Select / Reconnect Timeout Register	W
0x78800018	Sequence Step Register	R
	Sync Period Register	W
0x7880001C	FIFO Flags Register	R
	Sync Offset register	W
0x78800020	Configuration 1 Register	R/W

Table 5-1 FSC Register Set

Address	Register	Access
0x78800024	Clock Conversion Register	W
0x78800028	Test Mode	W
0x7880002C	Configuration 2 Register	R/W
0x78800030	Configuration 3 Register	R/W
0x78800038	Transfer Counter High Register	R/W

Table 5-1 FSC Register Set (Continued)

There follows a brief description of the FSC registers. A detailed programming description of the device is beyond the scope of this manual and the user should refer to the bibliography at the rear of this document.

Command Register

The Command Register is a double-buffered register that allows two commands to be written to the FSC consecutively. Bit 7 controls the enabling of DMA operations and bits 6:0 provide a command code. Within the command code, bits 6:4 control the operating mode, of which only one can be selected at any time.

Bit 7 Enable DMA
0 = DMA Mode Disabled
1 = DMA Mode Enabled

Bit 6:4 Select Mode, see Table 5-2

An interrupt is generated if an invalid mode for the FSC is specified by bits 6:4, or if the command is not supported. Table 5-2 shows the commands selectable using bits 6:0.

Command Register Value								Command	Interrupt
7	6	5	4	3	2	1	0		
Immediate Commands									
X	0	0	0	0	0	0	0	No operation	NO
X	0	0	0	0	0	0	1	Flush FIFO	NO
X	0	0	0	0	0	1	0	Reset 53C90A device	NO
X	0	0	0	0	0	1	1	Reset SCSI bus	NO
Disconnect Commands									
X	1	0	0	0	0	0	0	Reselect Sequence	YES
X	1	0	0	0	0	0	1	Select without ATN sequence	YES
X	1	0	0	0	0	1	0	Select with ATN sequence	YES
X	1	0	0	0	0	1	1	Select with ATN and stop sequence	YES
X	1	0	0	0	1	0	0	Enable Selection / Reselection	NO
X	1	0	0	0	1	0	1	Disable Selection / Reselection	YES
X	1	0	0	0	1	1	0	Select with ATN3	YES
X	1	0	0	0	1	1	1	Reselect Sequence	YES
Target Mode Commands									
X	0	1	0	0	0	0	0	Send message	YES
X	0	1	0	0	0	0	1	Send status	YES
X	0	1	0	0	0	1	0	Send data	YES
X	0	1	0	0	0	1	1	Disconnect sequence	YES
X	0	1	0	0	1	0	0	Terminate sequence	YES
X	0	1	0	0	1	0	1	Target command complete sequence	YES
X	0	1	0	0	1	1	1	Disconnect	NO
X	0	1	0	1	0	0	0	Receive message sequence	YES
X	0	1	0	1	0	0	1	Receive command sequence	YES
X	0	1	0	1	0	1	0	Receive data	YES
X	0	1	0	1	0	1	1	Receive command sequence	YES
X	0	1	0	1	1	0	0	Target Abort DMA	NO
Initiator Mode Commands									
X	0	0	1	0	0	0	0	Transfer information	YES
X	0	0	1	0	0	0	1	Initiator command complete sequence	YES
X	0	0	1	0	0	1	0	Accept message	YES
X	0	0	1	1	0	0	0	Transfer pad	YES
X	0	0	1	1	0	1	0	Set ATN	NO
X	0	0	1	1	0	1	1	Reset ATN	NO

Table 5-2 FSC Commands

Status Register

The Status register contains the device and interrupt status flags. The Status register should always be read prior to reading the Interrupt Status register which will cause bits to clear. The Status register contains Error, Transfer count and SCSI phase information.

Bit 7 — Interrupt
- 0 = No interrupt pending
- 1 = Interrupt pending

Bit 6 — Gross Error

This bit is set if the top of the FIFO is overwritten; if the top of the command register is overwritten; if the direction of DMA and SCSI transfer are in opposition; or if there is an unexpected phase change in initiator role during a synchronous data phase.

Bit 5 — Parity Error

Bit 4 — Terminal Count

Bit 3 — Valid Group Code

Bits 2:0 — SCSI Phase
- 000 = Data Out
- 001 = Data In
- 010 = Command
- 011 = Status
- 100 = Reserved
- 101 = Reserved
- 110 = Message Out
- 111 = Message In

Interrupt Status Register

This register can be used in conjunction with the Status and Sequence Step register to determine the cause of an interrupt. It includes bits to indicate SCSI bus reset, disconnect, bus service request, function complete and selected states.

Bit 7 — SCSI Bus

Bit 6 — Illegal Command

Bit 5 — Disconnect

Bit 4 — Bus Service

Bit 3 — Function Complete

Bit 2 — Reselected

Bit 1	Selected with ATN
Bit 0	Selected

Configuration Registers

The CON1, CON2 and CON3 registers allow various operating modes to be set up. CON1 is used to control the slow cable mode, parity enable and test, and the 3-bit SCSI host ID. CON2 provides control of the Tagged queuing, group 2 command and parity control facilities provided by the 53C90A.

Transfer Count Registers

These combine into a 24-bit counter used by a DMA command, and by the Command Sequencer to count the decoded length of a received SCSI command.

FIFO / Flags Register

The FIFO is a 16 byte deep buffer between the SCSI bus and the system memory accessible to the host via the FIFO Register.

The FIFO Flags register is a read only register which indicates the number of bytes remaining in the FIFO.

Select / Reconnect Register

This is a 3-bit wide write only register used to specify the destination SCSI bus ID for a select or reselect command.

Clock Conversion Factor Register

This register specifies the clock conversion factor allowing the FSC to be clocked at different speeds. On the SPARCbook 3, the FSC is clocked at 40 MHz and this register should be loaded with the value 0x00.

Select / Reconnect Timeout Register

This register specifies the time period to wait for a select / reselect response.

Synchronous Transfer Period Register

The Transfer Period register is a 5-bit write-only register that specifies the minimum time between leading edges of successive REQ or ACK pulses. The default time is 5 clock cycles, the maximum is 35 cycles.

Sequence Step Register

This provides an incrementing 3-bit sequence count to indicate which steps of a command sequence have been executed prior to an error or interrupt condition.

5.3 DMA Support

The SCSI controller is provided with DMA support by one channel of the MACIO integral DMA controller. Between the FSC and Sbus the MACIO provides a 64-byte deep FIFO (D-FIFO) which is bypassed by CPU accesses to the FSC's registers.

5.3.1 DMA Transfers

A transfer from SCSI to memory is carried out in two phases. First from the SCSI controller to the DMA controller, and then from the DMA controller to memory. Similarly, transfers from memory to SCSI are composed of a transfer into the D-FIFO, and then a transfer between the D-FIFO and SCSI controller.

Data from the SCSI is written into the D-FIFO until the largest possible Sbus burst write, as specified in the DMA Controller's Control and Status register, to memory can be carried out.

5.3.2 DMA Registers

The DMA controller provides four 32-bit registers used to control DMA operations with the FSC.

Address	Register	Size	Access
0x78400000	Control and Status Register	32	R/W
0x78400004	Address Register	32	R/W
0x78400008	Byte Count Register	24	R/W
0x7840000C	Test Control and Status Register	32	R/W

Table 5-3 SCSI Related DMA Registers

5.3.3 SCSI Interrupts

The SCSI controller signals interrupts to the CPU via the SLAVIO multifunction peripheral on level 4. The interrupt service routine can establish the cause of the interrupt by examining the contents of first the status register, and then the interrupt status register.

Ethernet Interface

6

The Ethernet interface on the SPARCbook is provided by an NCR92C990 integrated into the MACIO. This provides AUI connections via a 15-pin D-shell connector. The MACIO enhances Ethernet operations by providing DMA support.

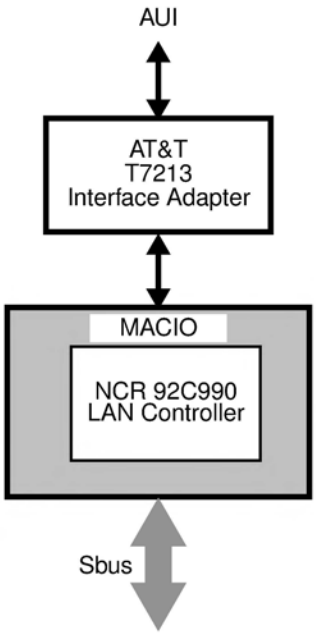


Figure 6-1 Network Interface Architecture

6.1 NCR92C990 Overview

The NCR92C990 is a LAN controller which supports the parameters for an IEEE 802.3 network interface.

6.1.1 Bus Interface

The LAN controller operates as a bus master or a bus slave device.

As a slave, it provides a 16-bit control interface on the SBus with two register locations. The CPU accesses the registers during system initialization, and in response to interrupts. Additional registers in the MACIO's DMA controller provide enhanced support for programming DMA operations.

As a master, it operates independently using DMA operations to transfer data between the network and data structures in memory. The LAN controller incorporates two independent 48-byte FIFOs; one for transmit operations, and one for receive operations.

6.1.2 LAN Interface

The NCR92C990 supports error reporting for diagnostics, addressing, collision, jabbering, framing, underflow and overflow. It allows internal and external loopback modes to be configured via the control registers.

Physical, logical and promiscuous addressing modes are supported. Packets can be received containing the full 48-bit destination address for matching against a physical address programmed during initialization. The LAN controller will also allow the programming of 64 logical addresses for matching with an incoming packet's address header. In the promiscuous mode, all incoming error free packets are accepted and stored in memory.

6.1.3 Descriptor Management

Buffer management for the LAN controller is handled by a recurrent list of assignments in memory called descriptor rings. There are separate descriptor rings for transmit and receive operations. The decriptor rings are searched to determine the location of the next empty buffer. After a buffer is filled, the OWN bit in the corresponding descriptor is set. When the controller detects that the OWN bit is set in a descriptor, it uses the ring buffer pointer in that descriptor to locate the next data buffer.

6.2 LAN Controller Registers

Access to the LAN controller's internal registers is gained via two 16-bit locations, the Register Data Port (RDP) and Register Address Port (RAP). The RAP is loaded with an index to the required register, and then the register data can be read from or written to the RDP.

Address	Register	Size	Access
0x78C00000	Register Data Port (RDP)	16-bit	R/W
0x78C00002	Register Address Port (RAP)	16-bit	R/W

Table 6-1 LAN Controller Register Locations

6.2.1 Register Indexing

The lower 2 bits in the Register Address Port are used to index the internal Control and Status Registers. The remaining bits, D(15:2) are reserved. The encoding of D(1:0) is shown in Table 6-2.

Bit 1	Bit 0	Register
0	0	Control and Status Register 0
0	1	Control and Status Register 1
1	0	Control and Status Register 2
1	1	Control and Status Register 3

Table 6-2 Register Indexing

6.2.2 Control and Status Register 0

This register contains control, status, error and interrupt information.

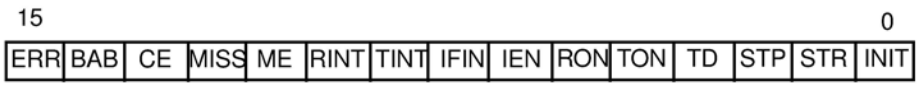


Figure 6-2 Control and Status Register 0

ERR	Error – a logical "OR" of BAB, CE, MISS and ME
BAB	Babble – Transmitter timeout error
CE	Collision Error
MISS	Missed Packet
ME	Memory Error – LAN controller unable to access memory as bus master.
RINT	Receiver Interrupt – set when a packet is received, or if a receive error has occurred.
TINT	Transmit Error – set when a transmission is completed or due to a transmit error.
IFIN	Initialization Finished
INT	Interrupt Pending
IEN	Interrupt Enable – when set, interrupt requests are enabled.
RON	Receiver On – receiver has been enabled
TON	Transmitter On – transmitter has been enabled
TD	Transmit Demand
STP	Stop – setting this bit causes the LAN controller to cease network activity. This bit must also be set to allow access to the other Control and Status registers. STP is reset by INIT or STR
STR	Start – setting this bit enables the transmitter and receiver and buffer management parts of the LAN controller. It also clears the STP bit.
INIT	Initialize – setting this bit causes the LAN controller to begin an initialization process.

6.2.3 Control and Status Register 1 and 2

These two registers combine to provide the 24 bit address of the initialization block. CRS1 contains the low order 16 bits, and CSR2 contains the upper 8 bits. The initialization block should always be aligned on a word boundary; i.e. bit 0 of CSR1 should contain 0.

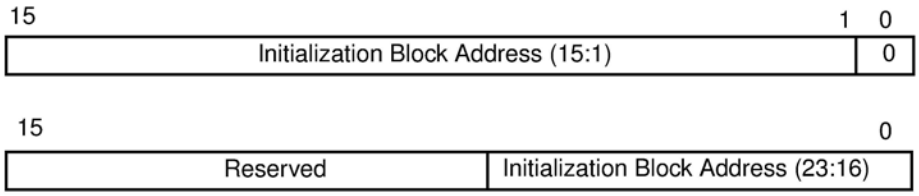


Figure 6-3 Control and Status Registers 1 and 2

6.2.4 Control and Status Register 3

This register is used to set the LAN controller operating parameters to suit its hardware environment. It is programmed by the resident firmware during system start-up and should not be changed.

6.3 DMA Support for Network Operations

The MACIO's DMA core provides additional registers for controlling network related DMA operations. Between the network interface and Sbus interface, there is a 2 line 8 word/line cache with consistency control logic. Each byte in the cache has an associated valid/dirty bit, and each line has a bit which determines the meaning of the valid/dirty bit for that line. These bits can be accessed in the Ethernet cache Valid Register.

Table 6-3 shows the DMA registers for network operations.

Address	Register	Size	Access
0x784000010	Ethernet Control and Status Register	32	R/W
0x784000014	Ethernet Test Control Registers	32	R/W
0x784000018	Cache Valid Bits	32	R/W
0x78400001C	Ethernet Base Address Register	8	R/W

Table 6-3 DMA Registers for Network Operations

The selection between 10Base5, 10Base2 and 10BaseT is determined by a bit port within the MACIO. However, OpenBoot automatically selects between the interfaces.

PCMCIA Interface

7

The PCMCIA inteface is controlled by a custom designed ASIC (application specific integrated circuit), the TS102. The TS102 provides support for two PCMCIA cards and, in addition, an interface between the CPU and microcontroller subsystem which provides battery management, keybaord and mouse interfacing, and initial power sequencing control. The TS102 also provides interfaces for an external keyboard and mouse, but these are not used in the SPARCbook 3 application.

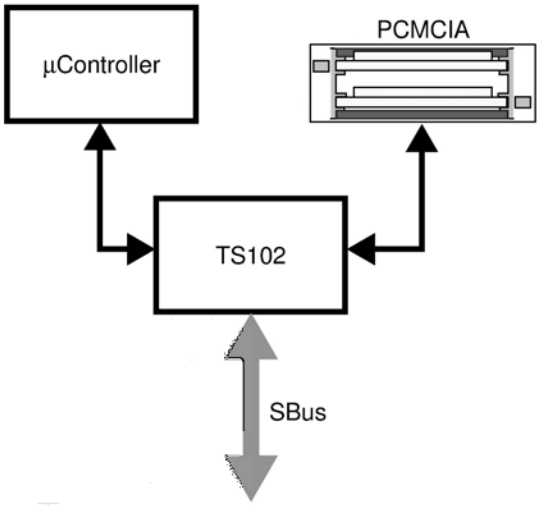


Figure 7-1 PCMCIA Interface Architecture

7.1 TS102 Architecture Overview

The general architecture of the TS102 ASIC is illustrated in Figure 8-5.

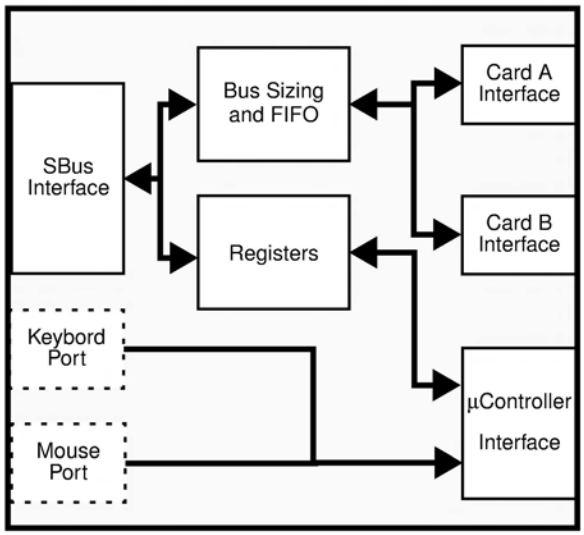


Figure 7-2 TS102 Architecture

7.1.1 SBus interface

The TS102 provides an IEEE-P1496 compliant 32-bit slave interface, with separate address spaces for PCMCIA memory accesses and for register accesses. This feature allows user processes access to the PCMCIA memory resources while restricting access to key systems resources, such as the microcontroller communications link.

The TS102 slave interface supports all SBus transfer sizes up to 8-word transfers, but not 16-word and extended transfer modes. The TS102 performs the required number of accesses to the PCMCIA interface to fulfill any request. For example, if the controller requests a 32-bit (word) access, and the card installed is 16 bit, then the TS102 performs two accesses.

The TS102 supports posted writes to PCMCIA cards of up 8 words. Such posted completed by the TS102 after the transfer has been acknowledged on the SBus. Until the write is complete, further accesses to the TS102 rerun by the TS102.

The TS102 also features a read-ahead capability. Setting the read-ahead bit in the TS102 card register causes the TS102 to pre-fetch an additional 8 words a of data after each 8 word read from the card, starting from the address used in the last transfer. The address used for subsequent PCMCIA I/O cycles as a result of a single SBus transaction may be static or incrementing.

The TS102 supplies a single interrupt request to the CPU. This interrupt request combines requests from the PCMCIA card interface and the microcontroller interface. Status registers allow the CPU to determine the source of the interrupt.

7.1.2 PCMCIA interface

The SPARCbook 3 complies to PCMCIA standards which define the mechanical and electrical specifications for a wide range of removable memory and I/O cards. However, there is no provision in the PCMCIA specification for a DMA master on a PCMCIA card.

The PCMCIA interface supports three memory spaces. These are common memory space, attribute memory space and I/O space. Common memory space is intended for simple read-write data memory; attribute memory typically contains the card configuration registers; and I/O space is used by I/O cards, such as network interface and modem cards.

The transfer cycle time is either fixed, for release 1.0 compatible cards, or is determined by the card itself via a WAIT line, for release 2.0 compatible cards. All card types and slots default to a common memory configuration at power up or following removal of a card. When a card is installed, the host interrogates the card through attribute memory to determine whether any I/O pin functions need to be redefined or whether card operating or programming voltages require adjustment.

A number of PCMCIA cards request low power 3.3 Volts operation via their attribute memory data, or request two different programming voltages for use by programmable memory. The TS102 provides bit-ports, which are used to select the appropiate voltages, as required. However, because some signals are shared between the two PCMCIA cards, it is not possible to operate one card at 3.3V while simultaneously operating the other card at 5V.

7.1.3 Microcontroller interface

A simple 8-bit parallel read-write port supports communications with the microcontroller subsystem. The CPU writes commands to a transmit register. The act of writing sets a busy bit in the microcontroller parallel

port status register, and also asserts an interrupt request to the microcontroller. The microcontroller read of the receive register clears the CPU write busy bit and the interrupt request.

There is a similar protocol for transfers from the microcontroller to the CPU. For improved performance, there is a four entry FIFO in each direction, which allows posted write cycles to be performed.

7.2 TS102 Memory Mapping

The TS102 occupies a 256 Mbyte region of the SPARCbook 3's address space. Its internal memory map is summarized in Table 8-4.

Address	Resource
0x40000000 - 0x41FFFFFF	FCode PROM
0x42000000 - 0x43FFFFFF	TS102 registers
0x44000000 - 0x44FFFFFF	Attribute space, card A
0x45000000 - 0x45FFFFFF	Attribute space, card B
0x46000000 - 0x46FFFFFF	I/O space, card A
0x47000000 - 0x47FFFFFF	I/O space, card B
0x48000000 - 0x4BFFFFFF	Common memory space, card A
0x4C000000- 0x4FFFFFFF	Common memory space, card B

Table 7-1 TS102 Address map

7.2.1 Acceses to PCMCIA Resources

The low order SBus address lines S_A(25:0) are passed directly through to PCM_A(25:0), permitting access to the full 64Mbyte of common memory address space. For attribute memory and I/O space, PCM_A(25:24) are driven from paging bits in the TS102's register set.

The TS102 accepts byte, halfword, word, doubleword, quadword, and octalword transfer requests, and responds with error acknowledgment on request for other sizes.

The TS102 detects the WAIT signal from any card that drives it. This can be used to extend the transfer cycle on memory and I/O accesses. The wait can be asserted for up to 12µs which is longer than the maximum 255 SBus clocks (10.2µs @ 25MHz) allowed for an SBus transfer. If a PCMCIA card

asserts its WAIT signal, the TS102 slave interface responds with a retry acknowledgment to the master, and then continues with the PCMCIA transfer cycle.

Similarly, burst accesses to slow PCMCIA cards can cause a single burst access to take longer than 10µs, and in these cases the TS102 responds with a retry acknowledgment. The TS102 checks subsequent cycles by comparing the address, size and direction to ensure that the new cycle is the retried cycle, and not a new and different cycle. Any other new cycle attempted to the PCMCIA slave interface must be rerun as it cannot be attempted until after the retried PCMCIA cycle is complete.

7.2.2 Byte Swapping

The TS102 supports programmable byte-swapping, to allow the use of big or little-endian PCMCIA cards. The two PCMCIA card interfaces in the TS102 have individually programmable byte swapping, although it is not advisable to program the two differently.

Four byte-swapping modes are implemented in the data routing block of the TS102, of which three are are of interest here. They are dependent upon the transfer size, and the two status bits per card indicating the byte ordering mode. One status bit provides endian control for the normally big-endian SBus interface, and the other status bit provides endian control for the PCMCIA card. The details of these modes are shown in theTable 7-2. Data on the left is the SBus data, and data on the right is the PCMCIA data.

Size	SBus BE PCMCIA LE	SBus LE PCMCIA LE	Sbus BE PCMCIA BE	SBus LE PCMCIA LE
Byte	D[31:24] <-> D[7:0]	D[31:24} <-> D[7:0]	D[31:24] <-> D[7:0]	D[31:24] <-> D[7:0]
Halfword	D[31:16] <-> D[15:0]	D[31:24] <-> D[7:0] D[23:16] <-> D[15:8]	D[31:24] <-> D[7:0] D[23:16] <-> D[15:8]	D[31:16] <-> D[15:0]
Word	D[31:0] <-> D[31:0]	D[31:24] <-> D[7:0] D[23:16] <-> D[15:8] D[15:8] <-> D[23:16] D[7:0] <-> D[31:24]	D[31:24] <-> D[7:0] D[23:16] <-> D[15:8] D[15:8] <-> D[23:16] D[7:0] <-> D[31:24]	D[31:0] <-> D[31:0]

Table 7-2 Sbus to PCMCIA Data Routing

Accesses to the I/O and attribute spaces are handled differently. The attribute space is byte wide and only even addressed bytes exist, while accesses to I/O space must be treated as bytes unless the I/O card responds

with an IOIS16* acknowledgement. The TS102 monitors the IOIS16* signal during I/O accesses in order to decide how many PCMCIA accesses to perform for a given master cycle.

7.2.3 SLAVIO expansion interface

The SLAVIO controls access to the boot EPROM and parallel port. It also has an expansion bus that allows additional I/O devices to be added. This expansion bus consists of an 8-bit data path, controlled by the SLAVIO, and a 20-bit address bus. To support posted write cycles, this address bus requires an external address latch . This latch is implemented in the TS102.

The expansion bus has a single "generic" chip select. The TS102 uses this, together with some high order address lines, to select the boot PROM, or and modem controller. There are three additional selects but these are not used in the SPARCbook 3. The address map of the decode is shown Table 7-3.

A[19:17]	Select
0 x x	Boot ROM
1 0 0	Spare
1 0 1	Modem
1 1 0	Interrupt register
1 1 1	Spare

Table 7-3 Generic Chip Select Decoding

The boot PROM is placed on the expansion bus because the SLAVIO only permits read accesses to the EPROM in its conventional location. However, because the boot PROM is a FLASH device, it is necessary to have write access to it so that can be reprogrammed in circuit. The TS102 takes the normal boot PROM select from the SLAVIO and ORs this with the decode of the generic select to produce a composite PROM select.

The FLASH device implements a software write protection to enhance with a hardware protection mechanism, in which the TS102 only generates a boot PROM select for a write cycle if the microcontroller has enabled EPROM write cycles in a command register in the TS102.

Generation of the boot PROM selects by the TS102 also allows the implementation of a boot PROM redirection facility. When in operation, the mechanism inhibits generation of an PROM select and passes the memory access to the PCMCIA card in slot A. Thus the CPU reads code from the PCMCIA memory instead of the boot EPROM. This feature is

intended primarily for recovery from accidental erasure of the boot PROM contents, but may also be used for such applications as field diagnostics, or even OS upgrades.

It is also necessary to inhibit the generation of the SBus select to the SLAVIO device when boot redirection is used, in order to prevent the SLAVIO from from responding to accesses that are intended for the PCMCIA interface, while still allowing access to all other SLAVIO resources. When redirection is selected, the TS102 inhibits the SLAVIO select if a decode of the SBus physical address indicates that the SLAVIO would normally direct the cycle to the boot PROM.

7.3 TS102 Registers

There are two separate register blocks within the TS102. The first gives access to PCMCIA card specific resources, and the second gives access to the microcontroller interface. Table 7-4 summarizes the TS102'2 register address map.

Address	Function
0x42000000	Card A Interrupt Register
0x42000004	Card A Status Register
0x42000008	Card A Control Register
0x4200000C	Not used
0x42000010	Card B Interrupt Register
0x42000014	Card B Status Register
0x42000018	Card B Control Register
0x4200001C	Not used
0x42000020	Microcontroller Interrupt Register
0x42000024	Microcontroller Data Register
0x42000028	Microcontroller Status Register
0x4200002C - 0x4200003C	Not used

Table 7-4 CPU Register Address Map

All registers are 16 bits wide and are can be read or written.

The PCMCIA specification relies on software negotiation between the host system and each card to configure the hardware interface. As a result of this negotiation, some signals on the PCMCIA interface can, if required, be redefined. This is handled via the TS102 register interface.

7.3.1 Card A and B interrupt registers

There is one 16 -bit interrupt register for each card. Each register contains an interrupt status (read) and clear (write) bits and an interrupt mask bit for each of four interrupt sources.

The request bit is the logical AND of the status and the mask bit, and indicates that an interrupt is being requested. The mask bits allow masking of individual interrupts. An interrupt is enabled when the mask is set to 1 and is cleared by writing a 1 to the associated clear bit.

The card interrupt registers also contain the soft reset flag. Setting this bit to 1 will cause the SPARCbook 3 to be reset.

Bit	Read	Write
0	IRQ request	Clear IRQ request
1	WP status changed request	Clear WP status changed request
2	Battery status changed request	Clear Battery status changed
3	Card detect status changed request	Clear card detect status changed
4	IRQ status	Not used
5	WP status changed status	Not used
6	Battery/card status changed status	Not used
7	Card detect status changed status	Not used
8	IRQ mask	IRQ mask
9	WP status mask	WP status mask
10	Battery/card status mask	Battery status mask
11	Card detect status mask	Card detect status mask
12	Soft reset	Soft reset
13	Not used	Not used
14	Not used	Not used
15	Not used	Not used

Table 7-5 Card Interrupt Register

I/O cards replace the two battery voltage signals with card status changed and audio waveform input signals. In this case the battery status changed interrupt becomes a card status changed interrupt, and the card status must then be read from the card.

7.3.2 Card status register

The card status register contains card status and control bits.

Bit	Name	Function	Asserted	Reset
0	PRES	Card is present	1	0
1	IO	Card is memory or I/O	1	0
2	TYP3	Card is type 3 (disk)	1	0
3	Vcc	Vcc voltage control		0
5:4	Vpp1[1:0]	Vpp1 voltage control		00
7:6	Vpp2[1:0]	Vpp2 voltage control		00
8	WP	Write protect status	1	0
10:9	BVD[1:0]	Battery voltage detect		
11	LVL	Card IRQ interrupt level/edge control	1	0
12	RDY	Card ready/not busy status	1	0
13	Vccen	Card Vcc enable	0	0
14	RIEN	Ring indicate enable	1	0
15	ACEN	Card access enable	1	0

Table 7-6 Card Status Register

Bit 15 ACEN – Card Access Enable
Accesses to the card are enabled when this bit is set (1)

Bit 14 RIEN – Ring Indicator Enable.This is no longer supported.

Bit 13 VccEN – Vcc enable. When this bit is cleared (0) power is enabled to the card.

Bit 12 RDY – Card ready/not busy status. This bit indicates the status of the card RDY/BSY* signal. Note that this signal becomes IREQ* on I/O cards, and the RDY/BSY* status may be available in the card pin replacement register, if this bit is implemented in it. Since the RDY/BSY* signal and IREQ* are one and the same, it is possible to generate an interrupt on BSY* becoming asserted, since the interrupt

request is not automatically masked when the card is not I/O. The interrupt request should normally be explicitly masked when the card is not I/O using the interurpt mask bit in the card interrupt register.

Bit 11 — LVL – IRQ Leve/Edge Control. This bit controls whether the card IRQ input is edge or level sensitive. By default it is edge sensitive.
- 0 = edge sensitive
- 1 = level sensitive

Bits 10:9 — BVD(1:0) – Battery Voltage Detect. These bits indicate the status of the card battery voltage detect bit. These are not valid for an I/O card, and the battery voltage status may be read from the card pin replacement register, if these bits are implemented in the card.
- 00 = battery low, data suspect
- 01 = battery low, data suspect
- 10 = battery low, data OK
- 11 = battery good

Bit 8 — WP – Write Protect. The WP bit indicates that the card is write protected, as the card has its WP signal asserted.

Bits 7:6 — Vpp2(1:0) – Programming Voltage Control

Bits 5:4 — Vpp1(1:0) – Programming Voltage Control. The Vpp1 and Vpp2 bits control the Vpp1 and Vpp2 voltages, and allow them to be set to:
- 00 = 0V
- 01 = Vcc (5V or 3.3V)
- 10 = Vpp (12V)
- 11 = NC

Bit 3 — Vcc – The Vcc bit controls the card Vcc voltage, allowing it to be set to either 5V or 3.3V:
- 0 = 5V
- 1 = 3.3V

Bit 2 — TYP3 – The TYP3 bit configures the card as a disk. This may not be implemented if Type-3 PCMCIA data is not available.

Bit 1 — IO – The IO bit controls whether the card is configured for I/O or memory:

0 = Memory card
1 = I/O card

Bit 0 PRES – Present. The PRES bit indicates that both card detects are active and that the card is correctly inserted.

7.3.3 Card Control Register

Bit	Name	Function	Asserted	Reset	Comments
1:0	AA[25:24]	Attribute address A(25:24)			
3:2	IA[25:24]	I/O address A(25:24)		00	
6:4	CES[2:0]	CE/address setup time		111	000 = 1 clocks 001 = 2 clocks 010 = 3 clocks 011 = 4 clocks 100 = 5 clocks 101 = 6 clocks 110 = 7 clocks 111 = 8 clocks
9:7	OEW[2:0]	OE/WE width		111	000 = 2 clocks 001 = 3 clocks 010 = 4 clocks 011 = 5 clocks 100 = 6 clocks 101 = 7 clocks 110 = 8 clocks 111 = 9 clocks
10	CEH	Chip enable hold time		1	0 = 1 clocks 1 = 2 clocks
11	SBLE	SBus little endian	1	0	1 indicates SBus operation in 'little' endian mode
12	PCMBE	PCMCIA big endian	1	0	1 indicates PCMCIA interface should be 'big' endian
13	RAHD	Read ahead enable	1	0	Set to 1 to enable read-ahead
14	INCDIS	Address increment disable	1	0	Set to 1 to disable PCMCIA address incrementing in I/O PCMCIA accesses from SBus
15	PWRD	Power down	1	0	Set to 1 to power down

Table 7-7 Card Control Register

7.3.4 Microcontroller Interrupt Register

Bit	Name	Function	Asserted	Reset
0	TXE_REQ	Transmit FIFO empty interrupt request	1	0
1	TXNF_REQ	Transmit FIFO not full interrupt request	1	0
2	RXNE_REQ	Receive FIFO not empty interrupt request	1	0
3	RXO_REQ	Receive FIFO overflow interrupt request	1	0
4	TXE_MSK	Transmit FIFO empty mask	1	0
5	TXNF_MSK	Transmit FIFO not full mask	1	0
6	RXNE_MSK	Receive FIFO not empty mask	1	0
7	RXO_MSK	Receive FIFO full mask	1	0

Table 7-8 Microcontroller Interrupt Register

7.3.5 Microcontroller data register

Bit	Read	Write
D[7:0]	Data from microcontroller	Data to microcontroller

Table 7-9 CPU Data Register

7.3.6 Microcontroller status register

Bit	Name	Function	Asserted	Reset
0	TXE_STA	Transmit FIFO empty status	1	1
1	TXNF_STA	Transmit FIFO not full status	1	1
2	RXNE_STA	Receive FIFO not empty status	1	0
3	RXO_STA	Receive FIFO overflow status	1	0

Table 7-10 Microcontroller Status Register

7.4 Microcontroller Registers

The microcontroller has access to TS102 registers within its own address space. The content of these registers and protocols for use are under the control of software programmed into the microcontroller's onboard ROM during system manufacture.

The registers within the microcontroller's address space are shown in the table below.

Address	Function
0x00	Host data register
0x01	Host interrupt register
0x02	Host status register
0x03	Not used
0x04	Command register
0x05-0x07	Not used
0x08	Keyboard data register
0x09	Keyboard interrupt status register
0x0A	Keyboard status/control register
0x0B	Not used
0x0C	Mouse data register
0x0D	Mouse interrupt status/control register
0x0E	Mouse status register
0x0F	Not used

Table 7-11 Microcontroller register set

ISDN and 16-bit Audio

8

The ISDN and 16-bit Audio interfaces on the SPARCbook 3 are provided by a pair of coupled components: the AT&T T7259 Dual Basic Rate ISDN Controller (DBRI); and the Crystal Semiconductor Corporation CS4215 Multimedia Audio Coder-Decoder (CODEC). These provide a terminal endpoint (TE) ISDN interface at the S/T reference point, and also high quality stereo audio. The architecture of the ISDN and audio interface is illustrated in Figure 8-1.

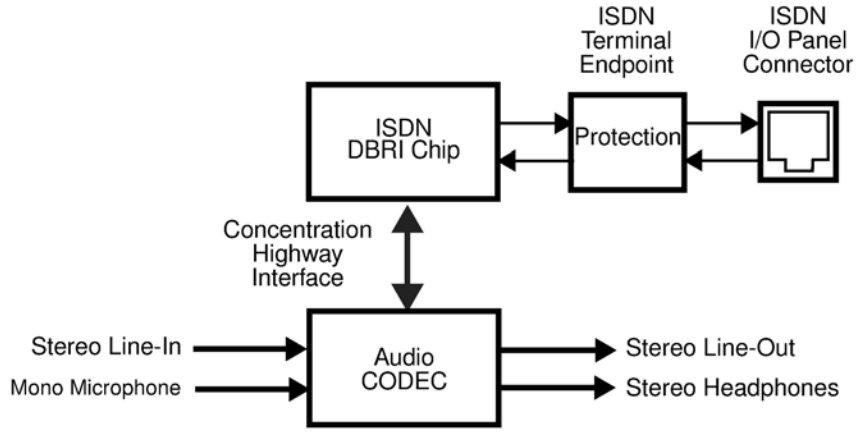


Figure 8-1 ISDN and Audio Interface

8.1 ISDN Overview

The integrated services digital network (ISDN) is an advanced telephone system which allows computers to communicate together at a higher data rate than they could using a modem. Information carried on the ISDN is digital and uses a circuit switched system at a *basic* rate of 64 KB/s on each of two bearer channels, or B channels, and 16 KB/s on a signaling and control channel, or D channel. The B channels can be combined to provide a maximum throughput of 128 KB/s.

The ISDN interface on SPARCbook 3 provides a 4-wire terminal endpoint (TE) connection via an RJ45 connector on the I/O panel, providing transmit and receive differential signal pairs. This is connected to the ISDN at the S (or S/T) interface point. Signals at the S point are then converted by an adapter called a network termination (NT1) to use the more common two wire telephone connection for which most premises are currently wired. The connection can be point to point between an NT1 and TE or multipoint, with up to eight TE devices attached to a single NT1. These eight devices may share a single ISDN line to achieve virtually simultaneous operation.

Time division multiplexing is employed to superimpose both B channels and D channel together onto a differential wire pair. The data carried on the B channels is arbitrary and could contain digitized voice and picture information, or raw computer data. A collision detection and backoff strategy allows multiple TEs to share the D channel for control and signaling functions.

Basic rate ISDN uses a 48-bit frame and defines several time slots within the frame. Among these are the D channel and two B channels. There are additional bits of information which are required by the physical layer electronic circuits to support the framing, synchronization and multipoint capabilities of ISDN.

8.2 DBRI Overview

The the AT&T T7259 Dual Basic Rate ISDN Controller (DBRI) functions as a multiport tap into an ISDN line and has four ports through which data is passed. In addition to the synchronous serial TE port, there is an NT port (not used in SPARCbook 3), a serial Concentration Highway Interface (CHI) port, and the 32-bit SBus DMA port. In general, the DBRI is a digital data transport, formatter and multiplexer that dynamically routes bits within time slots between ports.

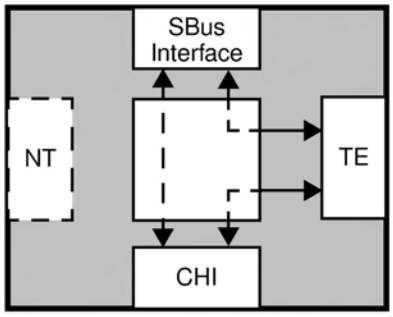


Figure 8-2 DBRI Internal Architecture

8.2.1 TE and NT Ports

The TE port provides the basic network connection to the ISDN. The NT port is not used in SPARCbook 3.

An internal FIFO buffer is used as a pipe to hold data transfering between interfaces. Data is moved between the various ports by assigning pipes and time slots to connect them. The D channel input and output streams use the High Data Level Link Control (HDLC) protocol, as required by ISDN. Once a connection is made through a pipe, little or no service is required to maintain it other than to manage buffer pools of data received and for transmission.

8.2.2 CHI Port

The CHI port is used to connect the audio CODEC. It provides a synchronous serial link which is used to transfer frames of audio data between the DBRI and audio CODEC. The CHI uses four wires to communicate: the data transport clock (CHIK); a frame synchronization pulse (CHIFS); a data transmit data wire; and a receive data wire. Data frames exchanged between the two devices contain digital audio or control information. Specific time slots within each frame are assigned to carry particular types of data.

8.2.3 SBus Interface

The 32-bit bus port is a fully compliant SBus DMA master that supports up to 64-byte DMA bursts. Up to sixteen pipes, each 80 bytes long, can be used to support DMA operations.

8.2.4 DBRI Programming Model

This section gives a brief overview of the DBRI programming interface. It is not, however, intended to provide a detailed programming guide.

The DBRI operates as a coprocessor, maintaining its own data structures in memory and operating on its own instruction set. The host assembles lists of instructions, services interrupts and maintains receive and transmit data buffers in main memory. The DBRI accesses memory using DMA operations.

The DBRI can be configured to select time slots and route selected data between the three serial interfaces and the SBus. It can be dynamically reconfigured to provide different routing strategies for different operations. For example, audio data carried on one B channel from the TE interface could be routed through the CHI for audio output, while data on the other B channel could be routed to the SBus interface for DMA into main memory. Figure 8-3 shows this configuration

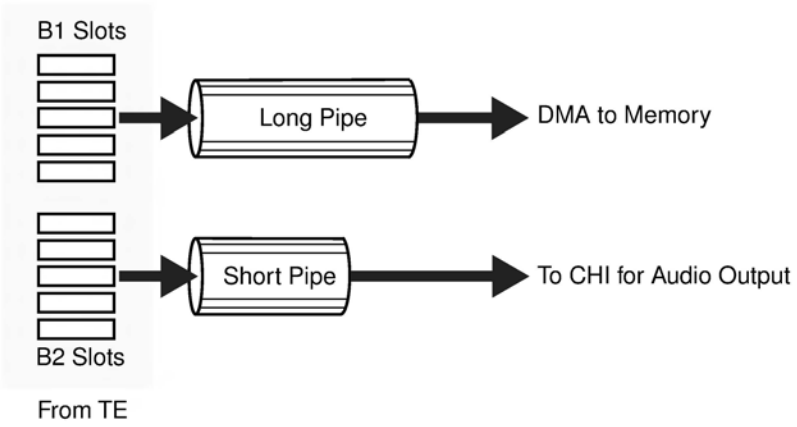


Figure 8-3 Long and Short Pipes

Data Pipes

The DBRI provides 32 data pipes through which data is channelled between the interfaces. These are divided into two types: sixteen long data pipes, so called because they have enough depth to buffer data for DMA (80 bytes); and sixteen short data pipes of 32 bits each. The long data pipes can be used for DMA with optional HDLC formatting in either direction. The short data pipes are used for data transfer between two serial interfaces (CHI, NT, and TE) and can have time slots assigned to both ends.

Long pipes can be configured to pass data through transparently or to code or decode data as HDLC frames. The CCITT Q921 protocol requires that protocol information is coded as HDLC formatted packets whereas payload data on the B channels does not require processing.

In HDLC mode, long pipes perform CRC generation and checking and automatically handle abort conditions.

Short pipes are only used to connect serial interfaces (e.g. CHI and TE). They cannot be used for DMA. The 32 bits of buffering provided by short pipes is adequate for transfer between serial interfaces because they are synchronized to a master clock with little unpredictable latency between the ports.

To maintain synchronization, some pipes are assigned as anchor pipes. With two serial interfaces active, there needs to be four anchor pipes. These pipes are the beginning points of linked lists maintained within the DBRI that define time slots for each interface. A number of pipes are predefined as anchor pipes: pipe 0 for the TE receive list; pipe 1 for the TE transmit list; and pipe 16 for the CHI receive and transmit lists (for DMA to the CHI, link a long pipe to the anchor pipe).

Data can enter a pipe in three ways:

- From a data structure in memory via DMA.
- From a time slot on one of the serial interfaces, as defined by a time-slot descriptor (TSD) associated with the data pipe.
- Fixed data can be set by the Set Short Pipe Data (SSP) command.

Similarly, data can leave a pipe in three ways:

- To a data structure in memory via DMA.
- To a time slot on one of the serial interfaces as defined by a TSD associated with the data pipe.
- A change in fixed data can be reported on the interrupt queue.

Any data pipe which is DMA or fixed on one end must be serial on the other end.

In the monitor mode, serial input from one time slot can go to two pipes. In the noncontiguous mode, multiple time slots from a serial interface can be assigned to the same pipe

Time Slots

Each of the DBRI's three serial interfaces (CHI, NT and TE) can have time slots defined for them. A time-slot descriptor (TSD) must be assigned to one end of the data pipe in order for data to enter or leave that end of the data pipe via a serial interface

Each basic rate interface (BRI) is viewed as a 19-bit long bit string consisting of the B1 and B2 channels (8 bits each), followed by an S or Q channel bit, and then 2 bits of D channel data. A time slot on a basic rate interface can be between 1 bit and 16 bits long. B1 and B2 can be concatenated and viewed as a single 16-bit time slot. For the BRI B channels, different time-slot descriptors can be used to overlap transmit and receive time slots. Multiple time slots from the same interface can connect to the same data pipe in the same direction (noncontiguous mode). The time slots do not have to be contiguous.

A time slot on the CHI can be up to 255 bits long.

There are a number of restrictions on the use of time slots:

- Time slots from different interfaces cannot be connected to the same data pipe in the same direction.
- For the CHI, the time-slot descriptors must not define overlapping time slots.
- CHI time slots that are greater than 32 bits long must connect to long data pipes.
- No CHI time slot can start within 0.975 µs after the start of the previous time slot.

The DBRI does not enforce these restrictions, but peculiar actions result from their violation.

Linked Lists

The time-slot descriptors are assembled into circular linked lists, one in each direction for the CHI and the two BRI interfaces. These linked lists must be maintained when using the Define Time Slot (DTS) command. When using any serial interface, the first time slot defined is assigned to the anchor pipe. That is: the TE receive time-slot list starts with the D channel in pipe 0; the TE transmit time-slot list starts with the D channel in pipe 1; the NT receive time-slot list starts with the D channel in pipe 3; and the NT transmit time-slot list starts with the D channel in pipe 2.

Both CHI time-slot linked lists (transmit and receive) start in pipe 16 (CHI anchor pipe). When initiating pipe 16, the CHI anchor mode must be used in the time-slot descriptor (TSD) field of the DTS command.

Fixed I/0 Channels

S and Q channels are supported on the basic rate interfaces. They can be connected to short data pipes, the other ends of which are in fixed I/O mode. Commands set a value for S and Q which is output repeatedly until changed. If an incoming S or Q channel changes, the change is reported on the interrupt queue. Note that when receiving fixed data, the FXDT interrupt reports only the least. significant 20 bits.

Frame Synchronization

The DBRI TE interface always synchronizes to its receive signal (from the network). The CHI can be either a slave or a master. If the CHI is master, it generates CHICK and CHIFS, and synchronizes these to the TE interface, if active. In CHI slave mode, the CHI synchronizes to the CHIFS input. If the TE interface is also active, data transmission between the CHI and the network can be unreliable. If the TE interface is synchronized with the network before the NT interface or CHI is active, then the NT CHI is frequency-locked with the TE when it is activated. It is recommended for data transfer between TE and CHI that TE is activated first. In this way, all interfaces are synchronized to TE and the network.

The DBRI provides a number of registers through which it is initialized, and a command set through which it is, mostly, controlled. Initialization performed by writing to the internal registers and command sequences are started by writing to REG8.

8.2.5 DBRI Internal Registers

The DBRI provides six registers through which its operations can be controlled. These are shown in Table 10-3.

Address	Register	Function
0x50000000 - 3F	ID Code	Read-only FCode
0x50000040	REG0	Status and Control Register
0x50000044	REG1	Mode and Interrupt Control Register
0x50000048	REG2	Parallel I/O Register
0x5000004C	REG3	Test Register
0x50000060	REG8	Command Queue Pointer
0x50000064	REG9	Interrupt Queue Pointer

Table 8-1 DBRI Internal Registers

REG0: Status and Control Register

Bit 15	P – This bit is set by the host or by the DBRI when REG8 (command pointer) is written. It is cleared by the DBRI when a WAIT command is encountered on the command list. When the bit is set, the DBRI begins executing commands from system memory, starting at the location pointed by the value in REG8.
Bit 14	G – This bit is used to enable 4-word burst transfers. This bit is cleared when a 4-word burst fails on the SBus. This bit is set by a hardware or software reset.
Bit 13	S – This bit is used to enable 16-word burst transfers. It is cleared when a 16-word burst fails on the SBus and cleared by a hardware or software reset.
Bit 12	E – This used to enable 8-word burst transfers. This bit is cleared when an 8-word burst fails on the SBus. This bit is cleared by a hardware or software reset.
Bit 7	X – This bit is set and cleared by the host. If this bit is set, the sanity timer is reset and disabled (TO pin held high). Clearing this bit restarts the sanity timer.
Bit 6	T – Permit Activation of the TE Interface. This bit is set and cleared by the host.
Bit 5	N – Permit Activation of the NT Interface. This bit is set and cleared by the host.
Bit 4	C – Permit Activation of the CHI Interface. This bit is set and cleared by the host.
Bit 3	F – This bit is set and cleared by the host. Setting this bit forces an immediate time-out of the sanity timer (overrides X bit). With the F bit set, the TO pin is low and remains low until this bit is cleared. This bit is set by a hardware or software reset. Unless the F bit is cleared, the sanity timer remains in the time-out state (TO remains low) even when registers are accessed.
Bit 2	D – This bit is set and cleared by the host. Setting this bit immediately disables any activity initiated by the DBRI. Since DMA is disabled, the DBRI stalls, and if left set long enough, the D8RI could have overrun and underrun conditions.

Bit 1	H – This bit is set and cleared by the host. Setting this bit halts all DMA associated with the long data pipes (these can be delayed) but enables the DBRI to continue execution of commands from the command queue. Any command which controls a data pipe (SDP and DTS commands) is delayed in its execution until the H bit is cleared. Issuing these commands with the H bit set is not recommended since overrun and underruns may occur.
Bit 0	R – This bit is set by the host CPU to RESET the DBRI. This so1tware reset generates a short internal reset signal which is active for two clock cycles. This bit is cleared internally when the ID code is available approximately 64 clock cycles after the software reset. This bit is also set by the hardware reset and cleared internally approximately 64 system clock cycles after the rising edge of the hardware reset. Setting this bit initiates an internal initialization procedure that releases all time slots, clears all pipes, and clears REG0 (except the R bit and F bit), REG1, REG2, and REG3.

REG1: Mode and Interrupt Register

This register is used to set the byte ordering mode (little or big endian) of the DBRI and also record interrupt events. The byte ordering applies only to data in data buffers and not to descriptors or commands. The function of this register is summarized below.

Bits 31:9	Reserved
Bit 8	BO – Byte Order
Bits 7:5	Reserved
Bit 4	MRR – Multiple Error Ack on SBus
Bit 3	MLE – Multiple Late Error on SBus
Bit 2	LBG – Lost Bus Grant on SBus
Bit 1	MBE – Burst Error on SBus
Bit 0	IR – Interrupt Indicator

REG2:Parallel I/O Register

This register is associated with parallel I/O port (PIO) operations. Three of the PIO pins are used to control D/C, PDN and RESET pins of the audio codec. The bits in this register are assigned as shown in Table 10-6 (all other bits are reserved).

Bit 7:4	Enable for PIO(3:0) – Reaserved
Bit 3	Parallel Input/Output 3 – Used to control the CODEC D/$\overline{C}$ pin
Bit 2	Parallel Input/Output 2 – Used to control CODEC PDN pin
Bit 1	Parallel Input/Output 1 – Used to control CODEC RESET pin
Bit 0	Parallel Input/Output 0 – Not connected

A bit port can be configured as an output by setting the associated enable bit. Setting a PIO bit takes the associated bit-port pin high.

REG3:Test Register

This register is reserved for manufacturing test purposes.

REG8: Command Queue Pointer

This register contains a pointer to the current command. When a value is written into this register, bit 15 in REG0 is set and the command pointed to by the value is executed. After reset, the DBRI does not perform I/O until REG8 is written. REG8 can be read to see which command is being executed.

REG9 Interrupt Queue Pointer

This register pints to the top of the current interrupt queue. It is initialized by the IIQ command.

8.2.6 DBRI Commands

Control of the DBRI is mainly accomplished using commands. Table 8-2 lists the DBRI command set.

Opcode	Command	Function
0x0	WAIT	Wait command
0x1	PAUSE	Pause command
0x2	JMP	Jump to new command queue
0x3	IIQ	Initialize interrupt queue
0x4	REX	Report execution of command via interrupt
0x5	SDP	Setup data pipe
0x6	CDP	Continue data pipe
0x7	DTS	Define
0x8	SSP	Set short data pipe
0x9	CHI	Set CHI global mode
0xA	NT	NT command
0xB	TE	TE command
0xC	CDEC	Codec setup command
0xD	Test	Test command
0xE	CDM	CHI data mode command
0xF		Reserved

Table 8-2 DBRI Command Set

Command execution is started when a pointer to the first command is written into REG8. Each command is at least one word in length. The first word contains the command opcode and control bits and the second (if needed) contains a pointer to the next command. Bits 31:28 within the first command word contain the opcode. If bit 27 (labeled I) is set, a CMDI interrupt is issued with the corresponding opcode in the interrupt field. Table 10-8 illustrates the format of each command.

The more commonly used instructions are as follows:

- **IIQ** – Initialize interrupt queue
- **SDP** – Setup data pipe. Initialize a data pipe to carry time slots.
- **DTS** – Define time slot. Specify a time slot by its start bit and length.

- **TE** – Set TE interface operating modes.
- **NT** – Set NT interface operating modes.
- **CDM** – Set CHI operating modes.
- **PAUSE** – Pause command execution to allow previous SDP commands to take effect.
- **CDP** – Reexamine transmit or receive descriptor pointer, to extend queue

For a detailed description of these commands, please refer to the technical documentation for the DBRI, see Appendix A, “Further Information”.

BIT																														
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WAIT																														
0	0	0	0	I	Reserved											Value														
PAUSE																														
0	0	0	1	I	Reserved											Value														
JMP (Jump to new command queue)																														
0	0	1	0	I	Reserved																									
Pointer to New Command																														
IIQ (Initialize interrupt queue)																														
0	0	1	1	I	Reserved											Value														
Pointer to New Interrupt Queue																														
REX (Report command execution with interrupt)																														
0	1	0	0	I	Reserved											Value														
SDP (Setup data pipe)																														
0	1	0	1	I	Reserved							IRM				MODE			D	B	P	-	A	C	-			PIPE		
Pointer to Transmit or Receive Descriptor																														
CDP (Continue data pipe)																														
0	1	1	0	I	RESERVED																							PIPE		
DTS (Define time slot)																														
0	1	1	1	I	Reserved									VI	VO	ID	Prev In PIPE					Prev Out PIPE					PIPE			
Input Time Slot Descriptor																						Monitor PIPE					Next PIPE			
Output Time Slot Descriptor																						Monitor PIPE					Next PIPE			
SSP (Set short data pipe)																														
1	0	0	0	I	Reserved																							PIPE		
Data																														
CHI (Set CHI global mode)																														
1	0	0	1	I				CHICM								IRM		OD	FE	FD	BPF									
NT																														
1	0	1	0	I	Control Bits																									
TE																														
1	0	1	1	I	Control Bits																									
CDEC (Codec setup command)																														
1	1	0	0	I		CK			FSCOD Falling Edge Delay												FSCOD Rising Edge Delay									
TEST																														
1	1	0	1	I		RAM Pointer										Size					Test Type									
Pointer to System Memory (SRC Pointer for COPY)																														
DEST Pointer (Only used for COPY)																														
CDM (CHI data mode)																														
1	1	1	0	I	Reserved																									
Control Bits																														

Table 8-3 DBRI Command Format Summary

8.2.7 Data structures

DBRI instructions, descriptors and transmit and receive data buffers are stored in data structures in main memory. Instruction execution is initiated by loading a pointer to the first instruction into REG8. The DBRI then continues by fetching instructions from data structures in memory.

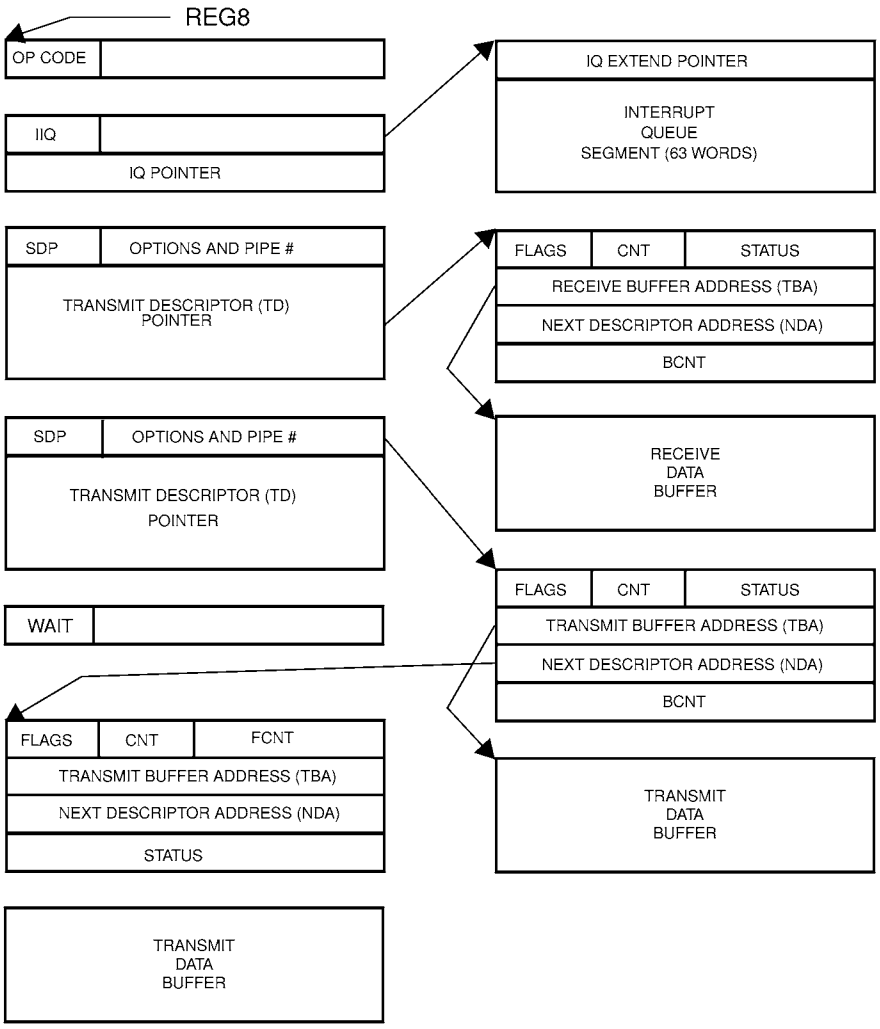


Figure 8-4 DBRI Data Structures

All DBRI data structures are aligned on 32-bit boundaries and transmit and receive descriptor buffers must be aligned on 4-word boundaries.

8.3 Audio CODEC

A Crystal Semiconductor Corporation CS4215 Multimedia Audio Coder-Decoder (CODEC) provides the SPARCbook 3 with stereo audio capabilities. It operates in conjunction with the DBRI with which it is connected via a synchronous serial link. The DBRI's concentration highway interface (CHI) provides a specialized serial interface which is used to support the audio codec's operations. Figure 8-5 shows how the DBRI and audio codec are connected.

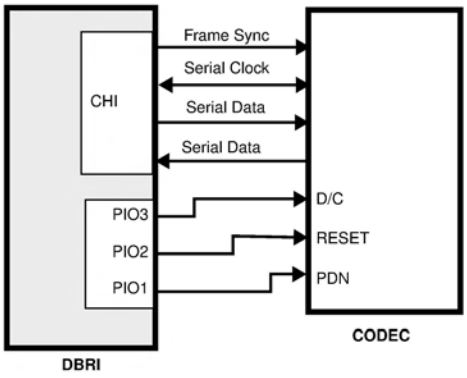


Figure 8-5 DBRI and CODEC Interface

8.3.1 Clocking and Data Synchronization

The SPARCbook 3 implementation of the DBRI and audio codec provides flexible operation of the audio interface. The codec has two clock inputs, as illustrated in Figure 8-6, which are used to generate a sampling clock for the internal analog-to-digital and digital-to-analog converters and can be programmed supply serial port timing. The codec is able to function as a timing master or timing slave for serial transfers between at it and the DBRI. It can generate frame synchronization and serial data clocks or can receive them from the DBRI.

The audio CODEC is not supplied with a clock on its CLKIN pin.

The CODEC operates in two modes: control mode, during which command and status information are transferred; and data mode, during which audio data is transferred. The operating mode is controlled via one if the DBRI's bit I/O pins.

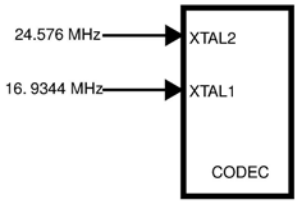


Figure 8-6 Audio Codec Clock Inputs

Data transferred between the two devices is assembled into frames carried between the tow devices during times slots specifically assigned to different types of data, as illustrated in Figure 8-7.

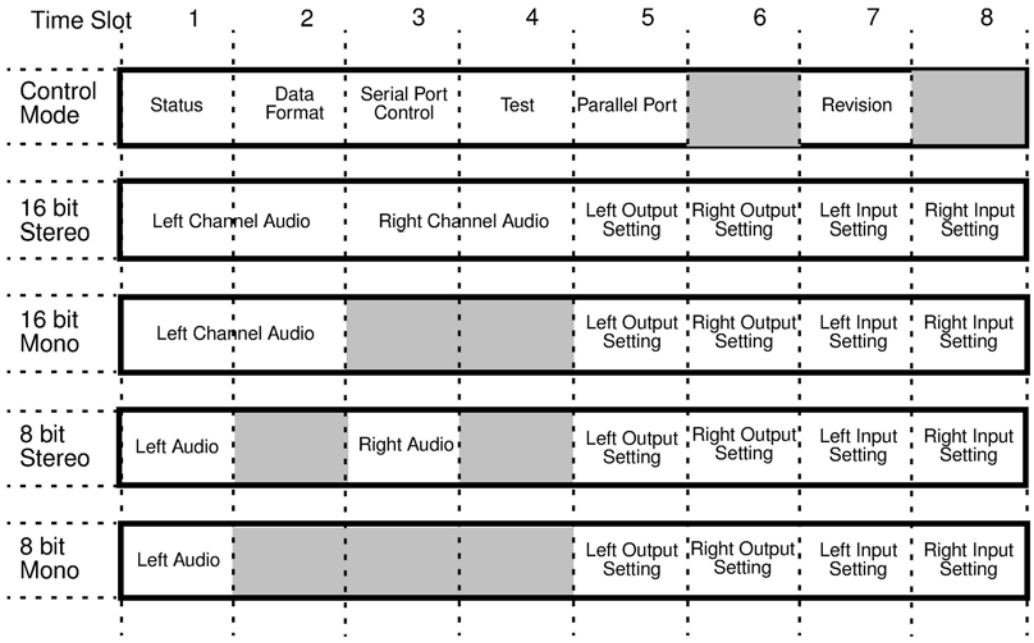


Figure 8-7 Data and Control Mode Time Slots

The serial data exchanged between the DBRI and codec always contains eight time slots and all time slots contain 8 bits (64 bits in total). Figure 8-8 shows the timing relationship between the serial clock (SCLK) and frame sync (FSYNC) signals and the serial data stream. (The TSOUT pin of the codec is not connected.)

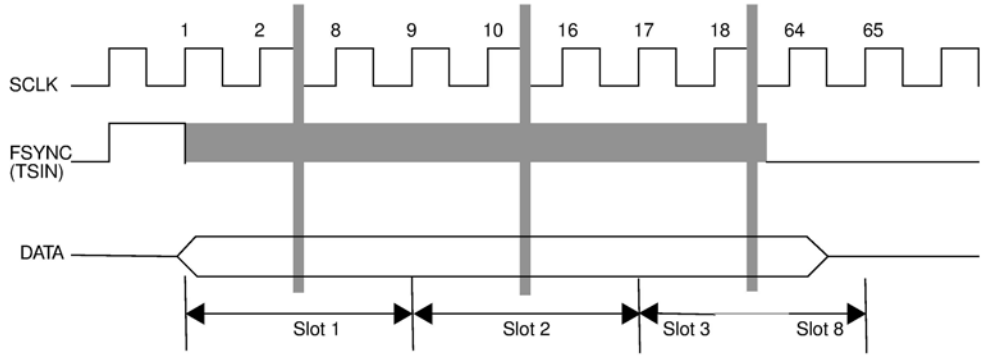


Figure 8-8 Frame Timing

The steps required to move the audio codec from data to control mode and then back again are as follows:

1. Lower the output level to maximum attenuation.
2. Mute the speaker output
3. Take the D/C pin low.
4. If the codec is timing master and the ITSD bit in the serial port control register is ‘0’, wait at least 12 SCLK periods to allow SCLK and FSYNC to tri-state
5. Drive the serial clock and frame sync signals into the audio codec.
6. Send control data as required with the CLB bit in the Status register low.
7. Read back and verify the control information from the codec. mask reserved bits. Wait for the CLB bit to go low.
8. Set the CLB bit high and send at least to more frames. This causes the codec to ignore any further activity on the bus. Also, the serial data out pin will be held in a highimpedence state afrter transmitteing one frame with the CLB bit high.
9. If the codec is progarmmed to be timing master, set the DBRI to receive SLK and FSYNC from the codec.
10. Set up new audio data to be transmitted to the codec

11. Set the D/C line high to place the codec in data mode. The codec will execute an offset calibration cycle
12. Transmit or receive audio data.

8.3.2 Control Mode

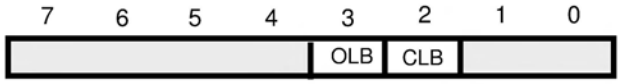
Conrol information can be written to the audio codec, and status can be read while is is in the control mode. The audio codec is placed in control mode via the DBRI bit port PIO3. There are six 8-bit internal registers which are each assigned to a time slot. The slots to which the registers are assigned are shown in Table 8-4.

Time Slot	Register
1	Status
2	Data Format
3	Serial Port Control
4	Test
5	Parallel Port – Not used
6	Reserved
7	Revision
8	Reserved

Table 8-4 Audio Codec Control Registers

The registers of most interest are described briefly below.

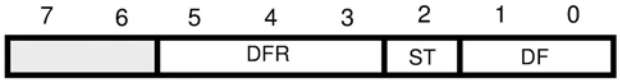
Status Register



Bit 3 OLB – Output Level Bit.
0 = Full scale outputs are:
Line 2.8 Vpp; Headphone 4.0 Vpp; Speaker 8.0 Vpp.
1 = Full scale for Line and headphone are 2.0 Vpp;
Speaker 4.0 Vpp

Bit 2 CLB – Control Latch Bit. This bit is used to ensure proper transition between control and data modes

Data Format Register



Bits 5:3 Data conversion frequency. The bits in this field are used to select the frequency have the following function:

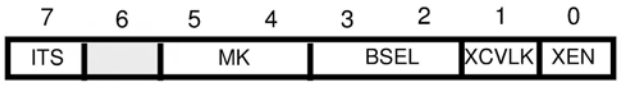
DFR	Crystal Source	
	XTAL1	XTAL2
0	8 kHz	5.5125 kHz
1	16 kHz	11.025 kHz
2	27.42857 kHz	18.9 kHz
3	32 kHz	22.05 kHz
4	NA	37.8 kHz
5	NA	44.1 kHz
6	48 kHz	33.075 kHz
7	9.6 kHz	6.615 kHz

Table 8-5

Bit 2 Stereo bit
0 = mono
1 = stereo

Bits 1:0 Data Format selection
00 = 16 bit law
01 = 8 bit μ law
10 = 8 bit A law
11 = Reserved

Serial Port Control Register



Bit 7 ITS – Immediate Tri-State Bit

0 = FSYNC and SCLK tri-sate 12 clock after D/$\overline{C}$ goes low
1 = FSYNC and SCLK tri-sate immediately after D/$\overline{C}$ goes low

Bits 5:4 MCK1:0 – Clock source select
00 = SCLK is master clock, 256 bits per frame. BSEL must be set to 2 and XCLK must be set to 0.
01 = XTAL 1 is clock source
10 = XTAL 2 is clock source
11 = CLKIN is clock source and must be 256 Fs

Bits 3:2 BSEL1:0 – Select bit rate
00 = 64 bits per frame
01 = 128 bits per frame
10 = 256 bits per frame
11 = reserved

Bit 1 XCLK – Transmit clock
0 = Receive SCLK and FSYNC from external source
1 = Generate SCLK and FSYNC

Bit 0 Transmitter enable
0 = Enable serial data output
1 = Disable serial data output

Test Register

Bits 7:2 Test – These bits must be written with zero

Bit 1 ENL – Enable Loopback
0 = disabled
1 = enabled

Bit 0 DAD – Loopback Mode
0 = Digital-digital loopback
1 = Digital-analog-digital loopback

8.3.3 Data Mode

The data mode is used during conversions to pass digital audio data between the DBRI and codec. The frame synchronization rate is equal to the value of the conversion frequency set by the by the DFR bits in the Data Fromat register. Each frames has either 64, 128 or 256 bit times, controlled by the BSEL bits in the Serial Control register. Control of gain, attenuation, input slection and output are embedded in the audio data stream.

All data time slots contain 8 bits with the MSB (D7) transmitted or recieved first. Data time slots are assigned to particular sections of the audio data bit-stream, as shown in Table 10-10.

Time Slot	Data Type
1	Left Audio most significant 8 bits
2	Left Audio least significant 8 bits
3	Right Audio most significant 8 bits
4	Right Audio least significant 8 bits
5	Output setting
6	Output setting
7	Input settings
8	Input settings

Table 8-6 Audio Data Time Slots

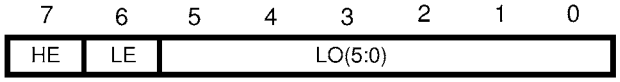
Left Channel Audio

Time slot 1 and 2 contain the audio data for the left channel. Audio data is transmitted MSB first and uses 2's compliment coding. In mono mode, only the left channel data is used, although both right and left DACs are driven. In 8-bit modes, only time slot 1 is used for the data.

Right Channel Audio

Time slot 3 and 4 contian the audio data for the right channel. Audio data is transmitted MSB first and uses 2's compliment coding. In mono modes, the right ADC zeros and the right DAC used the left digital data. channel data is used, although both right and left DACs are driven. In 8-bit modes, only time slot 3 is used for the data.

Output Setting

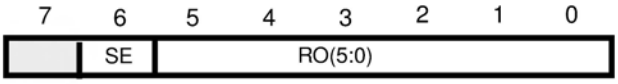


Bit 7 HE – Headphone output enable
0 = Headphone disabled
1 = Headphone enabled

Bit 6 LE – Line Output Enable
0 = Line Output disabled
1 = Line Output enabled

Bits 5:0 Output attenuation for left channel. LO(5) is MSB. LO(0) represents 1.5 dB.
0 = no attenuation

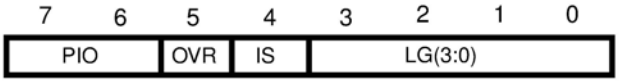
Output Setting



Bit 6 SE – Speaker Enable
0 = Speaker disabled
1 = Speaker enabled

Bits 5:0 Output attenuation for right channel. LO(5) is MSB. LO(0) represents 1.5 dB. 0 = no attenuation

Intput Setting



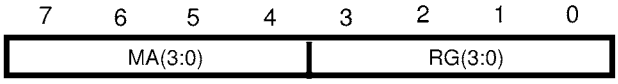
Bits 7:6 PIO bits – Not used

Bit 5 OVR – Overrange. This bit when set indiactes that an overrange condition has occured. It remains set until written with a zero. Writing a 1 enables the overrange detection. The bit remains at 0 until an overrange occurs.

Bit 4 IS – Input selection
0 = Line level input
1 = Microphone level input

Bits3:0 LG(3:0) – Input gain for left channel
LG(3) is the MSB. LG(O) represents 1.5 dB. Full gain is 22.5 dB.
0 = no gain

Intput Setting



Bits 7:4 MA(3:0) – Monitor path attenuation
MA3 is the MSB. MA0 represents 6 dB Full gain is at least 22.5 dB. Not used in mono modes.

Bits 3:0 RG(3:0) – Input gain for right channel
RG3 is the MSB. LGO represents 1.5 dB. Full gain is 22.5 dB.
0 = no gain

MODEM

9

This chapter discusses the SPARCbook 3's internal modem. It uses a two-chip set comprising the R65C39 microcontroller unit (MCU) and RC144APL modem data pump (MDP). Connection to a telephone line is made via a data access arrangement (DAA) which provides the required line isolation

Figure 9-1 shows the implementation of the modem interface in the SPARCbook 3.

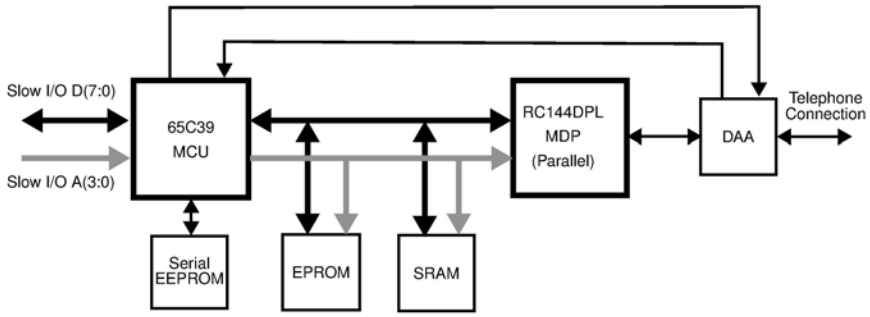


Figure 9-1 SPARCbook Internal Modem

9.1 Internal Modem Overview

The MCU provides the interface between the host system and DPL. It incorporates a processor core and a 16C450 UART-type parallel interface, and supports an enhanced AT command set as well as Fax Class 2 commands.

The MDP supports data and facsimile communications. As a data modem, it operates at up to 14400 bps (V32.bis), and with data compression (V42) can increase this by a factor of four. As a fax modem it supports V29, V27 ter and V21 channel 2 recommendations.

The modem data pump interfaces to the telephone line via a data access arrangement (DAA) which provides the required line isolation and impedance matching.

Note

The DAA is only approved for connection to public telephone systems in the U.S.A. and Canada.

The EPROM provides storage for the MCU's operational software, and the SRAM provides storage for parameters and the S-registers.

The modem interface can be controlled using Hayes compatible commands written to the parallel interface of the MCU. Commands are sent to the interface using character strings, and these are interpreted by the MCU and acted on appropriately.

Command characters and data for transmission onto the telephone line are written into the MCU's transmit buffer. Data received from the telephone line and status information from the DPL can be read from the MCU's receive buffer.

The modem interface operates in two modes: Command Mode or On-Line Mode. In Command Mode, characters written to the transmit buffer are regarded as being command characters. In the On-line mode, data written to this location (apart from the escape sequence +++) are transmitted onto the telephone line.

9.2 Interface Control

The SPARCbook implementation of Solaris provides the user with complete control of the modem via a graphical user interface. Behind this interface, the host controls the modem via a set of registers which are part of the parallel interface of the MCU.

The MCU registers are accessible at base address 0x0F0280000 and are shown in Table 11-1.

ADDRESS	REGISTER	ACCESS
0F0280000	Receive Buffer	Read-only
	Transmit Buffer	Write-only
	Divisor Latch (LSB)	Read-Write
0F0280001	Interrupt Enable	Read-Write
	Divisor Latch (MSB)	Read-Write
0F0280002	Interrupt Identity	Read-only
0F0280003	Line Control	Read-Write
0F0280004	Modem Control	Read-Write
0F0280005	Line Status	Read-Write
0F0280006	Modem Status	Read-Write
0F0280007	Scratch Pad Register	Read-Write

Table 9-1 Modem Control Registers

Bit 7 in the Line Control register is used as a pointer to either the Transmit and Receive Buffers and the Interrupt Enable register, or to the Divisor Latch registers. Bit 7 must be at '0' for access to the Transmit and Receive Buffers, or at '1' for access to the Interrupt Enable register and Divisor Latch.

The host is responsible for managing the transmit and receive buffers. During transmit operations, the host must supply data at a sufficient rate to ensure that there is always a new character for the modem to transmit; the modem is unable to wait and will regard an empty buffer as an error condition and disconnect the telephone line. During receive operations, the host must read the data in the buffer in time for the next incoming character from the telephone line.

The Modem generates interrupts when the receive buffer contains a character, when the transmit buffer is empty, and also to signal error conditions. All interrupt requests within the SPARCbook are maskable and are prioritized by the SLAVIO, see Section 3.2, "Interrupts", on page 3-5.

9.3 Modem Registers

This section describes the more significant registers from the point of view of managing the flow of commands and data between main memory and the modem.

9.3.1 Interrupt Enable Register

This register contains interrupt control bits. Setting one of the interrupt enable bits has the effect of enabling the associated interrupt request.

Bits 7:4	Reserved
Bit 3	Enable Modem Status Interrupt
Bit 2	Enable Receiver Line Status Interrupt
Bit 1	Enable Transmitter Holding Register Empty Interrupt
Bit 0	Enable Received Data Available Interrupt

9.3.2 Interrupt Identification Register

This register provides the identity of the highest priority pending interrupt condition, and a flag which indicates whether or not there is an interrupt pending. The MCU prioritizes the internal interrupt requests as shown below.

Bit 7:3	Reserved
Bit 2:1	Interrupt ID 11 = Receiver Line Status – highest priority 10 = Receiver Data Available 01 = Transmit Buffer Empty 00 = Modem Status – lowest priority
Bit 0	Interrupt Pending Flag 1 = No Interrupt Pending 0 = Interrupt Pending

9.3.3 Line Status Register

The Line Status register contains information which allows the condition of the transmit and receive buffers to be monitored.

Bit 7	Reserved
Bit 6	Transmitter Empty This bit when set indicates that both the transmit buffer and the transmit shift register are empty
Bit 5	Transmitter Buffer Empty This bit when set indicates that the transmit holding register is empty. It is cleared automatically when the host writes data into the transmit holding register
Bit 4	Break Interrupt Flag
Bit 3	Framing Error Flag
Bit 2	Parity Error
Bit 1	Overrun Error
Bit 0	Receiver Data Ready

9.4 AT Command Set

Normally, a user interacts with the modem through Mail Tool or FAXtool in the OpenWindows environment, or with Unix commands from the Solaris command line. It is also possible to interact with the modem directly using the Hayes compatible AT command set. This section describes some of the more commonly used commands in AT command set supported by SPARCbook modem, as summarized in Table 9-2.

To respond to AT commands, the modem must be placed in Command Mode. This can be done from the Solaris command line using the `cu` command:

```
% cu modem
```

When in Command Mode, the modem accepts instructions in the form of command lines and in many instances returns responses. The modem can be instructed to perform functions such as originating or answering calls, or can be configured to change its mode of operation. The SPARCbook modem is provided with an EEPROM which allows two configuration profiles and four telephone numbers to be retained.

CODE	DESCRIPTION
A	Go Off-hook in Answer Mode
A/	Re-execute Previous Command
AT	Attention Characters
B	Bell/CCITT Protocol
D	Dial Telephone Number
En	Command Echo
Fn	Select Line Modulation
Hn	Switch Hook Control
In	Identification
Nn	Automode
O	Return to Online
P	Set Pulse Dialling Default
Qn	Quiet Command Reset Code
Sn	Read/Write From Selected S-Register
T	Set Tone Dialling Default
V	Enable Short-form Result Code
Wn	Error Correction Message Control
X	Enable Extended Result Code Set
Y	Long Space Disconnect
&Fn	Restore Factory Configuration
&V	Display Current Configuration and Stored Profiles
&W	Store Current Configuration
&Yn	Designate a Default Profile
&Zn=x	Store Telephone Number
%Cn	Enable/Disable Data Compression
%En	Enable/Disable Auto Retrain
*B	Display Blacklisted Number
*C	Remote Configuration Password
*D	Display Delayed Numbers
*E	Exit Remote Configuration Mode
*L	Display Secure Access Directory
*P	Store Callback Password
*R	Request Remote Configuration

Table 9-2 AT Command Set Summary

With the exception of A/, all command lines begin with the attention characters "AT" followed by one or more command characters, and are terminated with a **RETURN**. Command lines may contain up to 56 characters, including A and T. All characters before the AT string, and all characters that follow an errant command are ignored.

The modem can be returned to normal operation by typing in **~.** (tilde period).

Note

Normally, a host computer and modem communicate via a serial link, but in the SPARCbook application, the MCU and DPL chipset present a parallel interface to the host and are themselves interconnected via an asynchronous parallel interface. The commands which relate to the control of a serial link between a host and modem have no effect on the operation of the SPARCbook modem.

A **Answer Incoming Call**
This forces the modem to go off-hook in answer mode.

A/ **Re-execute Previous Command**
This repeats the last command. It is not preceded by the AT characters or terminated by pressing **RETURN**.

AT **ATtention Characters**
These characters must appear at the beginning of all command lines.

Bn **Bell/CCITT Protocol**
This command selects the communication standard:

n =	0 CCITT operation
n =	1 Bell

Dn **Dial Telephone Number**
This command causes the modem to dial up a remote modem. The following modifiers may be added:

L	Redial last number
P	Pulse Dialling
T	Touch-tone Dialling
R	Originate Call in Answer Mode
W	Wait for dial tone for a time defined by S6
,	Delay dial sequence for a time defined by S8
@	Wait for Quiet for a time defined by S7
!	Go On-hook for a time defined by S29
;	Return to Command Mode
S=n	Dial a Stored Number

En **Echo Command Characters**

This controls whether the modem echoes command characters back to the host:

- n = 0 Disable Character Echo
- n = 1 Enable Character Echo

Fn **Select Line Modulation**

This selects line modulation:

- n = 0 Auto Dial mode
- n = 1 V.21 or bell 103 (according to Bn)
- n = 2 Reserved
- n = 3 V.23
- n = 4 V.22 at 1200 bps
- n = 5 V.22 bis
- n = 6 V.32 bis at 4800 bps
- n = 7 V.32 bis at 7200 bps
- n = 8 V.32 bis at 9600 bps
- n = 9 V.32 bis at 12000 bps
- n = 10 V.32 bis at 14400 bps

Hn **Switch Hook Control**

- n = 0 Go on-hook (hang up)
- n = 1 Go off-hook to access the telephone line

In **Identification**

This causes the modem to respond with identification codes:

- n = 0 Request Product Code
- n = 1 ROM Checksum
- n = 2 Return OK Response
- n = 3 Manufacturers ID
- n = 4 Configuration Mode
- n = 33 Sierra ID

O **Return to Online**

This command returns to the modem to the Data Mode.

Qn **Command Response Control**

This controls whether the modem provides responses to commands:

- n = 0 Return Response
- n = 1 Do Not Send Response

Sr=n **Change Register Value**

This selects an S register and changes its contents:

- r = S register
- n = New Contents

Sn? **Read S Register**
This returns the contents of an S register
n = S-register

Vn **Response Format**
This command is used to select the format of response made by the modem to the host:
n = 0 Single Digit Response
n = 1 Extended Response

Xn **Select Extended Response Set**

&Fn **Fetch Factory Configuration**
This recalls the factory settings of the modem.
n = 0 Recall factory profile 0
n = 1 Recall factory profile 1

&V **Display Current Configuration and Stored Profiles**
This displays the currently active configuration, the stored profiles and first four stored telephone numbers.

&Wn **Store Current Configuration**
Store the currently active configuration, including S registers, as a profile n (0 or 1).

&Yn **Designate a Default Profile**
Selects the profile used after a hardware reset.

&Zn=x **Store Telephone Number**
The modem can store up to 20 numbers:
n = Number memory 0 - 19
x = Dial string of up to 40 characters

%Cn **Enable/Disable Data Compression**
n = 0 Disable data compression
n = 1 Enable MNP 5 data compression
n = 2 Enable V.42 bis data compression
n = 3 Enable V.42 bis and MNP 5 compression

%En **Enable/Disable Auto Retrain**
This controls whether the modem automatically monitors the line quality and requests a retrain when needed:
n = 0 Disable auto retrain
n = 1 Enable auto retrain

***B** **Display Blacklisted Numbers**
This causes the modem to return a list of blacklisted numbers.

***C** **Remote Configuration Password**
This instructs the modem to store a password. By supplying a matching password, a remote modem may reconfigure the local modem. Supplied by a remote modem.

***D** **Display Delayed Numbers**
This causes the modem to send a list of delayed numbers and the delay associated with each.

***E** **Exit Remote Configuration Mode**
This causes a remote modem to exit remote configuration mode and transmit OK onto the telephone line.

***L** **Display Callback Directory**
This causes the modem to supply a list of all callback directory entries.

***P** **Store Callback Password**
This causes the modem to store a password and to store or delete a corresponding telephone number. The password is used to match that supplied by a remote modem when secure access is used. The number is used to dial back the remote modem.

***R** **Request Remote Configuration**
This causes the modem to attempt to place a remote modem into remote configuration mode. This is only possible if the local modem is in the online command mode and is connected to the remote modem by an MNP error corrected link.

9.5 S Registers

The SPARCbook modem provides a set of S registers which can be used to control the activity and configuration of the modem, and can be used to obtain status information. A set of default values (or profile) are loaded into these registers during system power-on. The modem can be loaded with one of two profiles with the **AT&Yn** command. Table 9-3 provides a summary of the function of the SPARCbook modem S registers.

The current contents of an S-Register can be read using the command:

```
ATS0?
```

will cause the modem to respond by returning the contents of the S-Register 0 in three-digit form followed by OK.

The contents of an S-Register can be changed using the Change S-Register Command. For example, the command line:

```
ATS0=2
```

causes the modem to change the contents of S-Register 0 to 2.

S0 **Number of Rings to Auto-Answer**
Sets the number of rings required before the modem automatically answers a call. Setting this register to zero disables auto-answer mode.

Range: 0-255 rings, Default: 2

S1 **Ring Counter**
Sets the number of rings required before the modem answers a call. If no rings occur over an eight second interval, this register is cleared.

Range: 0-255 rings, Default: 0

S2 **Escape Character**
Holds the decimal value of the ASCII character used as the escape character. The default value corresponds to an ASCII value. A value over 127 disables the escape process, i.e., no escape character will be recognized.

Range: 0-255, ASCII decimal, Default: 43 (+)

REGISTER	FUNCTION
S0	Rings to Auto-answer
S1	Incoming Ring Counter
S2	Escape Character
S3	Carriage Return Character
S4	Line Feed Character
S5	Backspace Character
S6	Dial Tone Wait Time
S7	Wait Time for Remote Carrier
S8	Pause Time for Dial Delay Modifier
S9	Carrier Detect Response Time
S10	Carrier Loss Disconnection Time
S11	DTMF Dialing Speed
S12	Escape Guard Time
S13	Reserved
S14	Command Options Status
S15	Reserved
S16	Modem Loopback Tests (&T)
S17	Reserved
S18	Test Timer
S19-20	Reserved
S21	Command Options Status
S22	Command Options Status
S23	Command Options Status
S24	Sleep Inactivity Timer
S25	DTR Delay for Synchronous Operation
S26	RTS/CTS Delay
S27	Command Options Status
S28	Command Options Status
S29	Flash Dial Modifier Time
S30	Disconnect Inactivity Timer
S31	Command Options Status
S32	XON Character
S33	XOFF Character
S34-35	Reserved
S36	LAPM Failure Control
S37	Line Connection Speed
S38	Delay Before Forced Hangup
S39	Flow Control
S40	Command Options Status
S41	Command Options Status
S42-45	Reserved
S46	Data Compression Control
S48	V42 Negotiation Control
S80	Soft Switch Functions
S82	LAPM Break Control
S86	Call Failure Reason Code
S91	Fax Transmit Level
S95	Result Code Messages Control
S99	Leased Line Transmit Level

Table 9-3 S Register Summary

S3 **Carriage Return Character**
Sets the command line and result code terminator character. Pertains to asynchronous operation only.

Range: 0-127, ASCII decimal, Default: 13 (Carriage Return)

S4 **Line Feed Character**
Sets the character recognized as a line feed. Pertains to asynchronous operation only. The Line Feed control character is output after the Carriage Return control character if verbose result codes are used.

Range: 0-127, ASCII decimal, Default: 10 (Line Feed)

S5 **Backspace Character**
Sets the character recognized as a backspace. Pertains to asynchronous operation only. The modem will not recognize the Backspace character if it Is set to a value that is greater than 32 ASCII. This character can be used to edit a command line. When the echo command is enabled, the modem echoes back to the local DTE the Backspace character, an ASCII space character and a second Backspace character; this means a total of three characters are transmitted each time the modem accesses the Backspace character.

Range: 0-127, ASCII decimal, Default: ^ (Backspace)

S6 **Dial Tone Wait Time or Dial Tone Before Blind Dialing**
Sets the length of time, in seconds, that the modem must wait (pause) after going off-hook before dialing the first digit of the telephone number. The modem always pauses for a minimum of 2 seconds, even if the value of S6 is less than 2 seconds. The "Wait for Dial Tone" call progress feature (W dial modifier in the dial string) will override the value in register S6. This operation, however, may be affected by some ATX options, according to country restrictions.

Range: 0-255 seconds, Default: 4

S7 **Wait Time for Carrier After Dial**
Sets the time, in seconds, that the modem must wait before hanging up because carrier is not detected. The timer is started when the modem finishes dialing (originate), or goes off hook (answer). In originate mode, the timer is reset upon

detection of answer tone if allowed by country restrictions. This timer also specifies the wait for silence time of the @ dial modifier in seconds. S7 is not associated with the W dial modifier.

Range: 0-255 seconds, Default: 50

S8 **Pause Time for Dial Delay**

Sets the time, in seconds, that the modem must pause when the "." dial modifier is encountered in the dial string.

Range: 0-255 seconds, Default: 2

S9 **Carrier Detect Response Tim**

Sets the time, in tenths of a second, that the carrier must be present before the modem considers it valid and turns on RLSD. As this time is increased, there is less chance to detect a false carrier due to noise from the telephone line.

Range: 1-255 tenths of a second, Default: 6 (0.6 second)

S10 **Lost Carrier To Hang Up Delay**

Sets the length of time, in tenths of a second, that the modem waits before hanging up after a loss of carrier. This allows for a temporary carrier loss without causing the local modem to disconnect. When register S10 is set to 255, the modem functions as if a carrier is always present.

The actual interval the modem waits before disconnecting is the value in register S10 minus the value in register S9. Therefore, the S10 value must be greater than the S9 value or else the modem disconnects before it recognizes the carrier.

Range: 1-255 tenths of a second, Default: 14 (1.4 seconds)

S11 **DTFM Durations**

Default 0

S12 **Escape Code Guard Time**

Defines the maximum period, in tenths of a second, allowed between consecutive asynchronous escape characters (+) for the escape sequence to be considered valid.

Range: 0-255 tenths of a second, Default: 50 (5 seconds)

S13 **Reserved**

S14 **General Bit Mapped Options**

Indicates the status of command options.

Default: 138 (8Ah) (10001010b)

- Bit 0 This bit is ignored
- Bit 1 Command echo (En)
  - 0 = Disabled (E0)
  - 1 = Enabled (E 1) (default)
- Bit 2 Quiet mode (Qn)
  - 0 = Send result codes (Q0) (default)
  - 1 = Do not send result codes (Q1)
- Bit 3 Result codes (Vn)
  - 0 = Numeric (V0)
  - 1 = Verbose (V1) (default)
- Bit 4 Reserved
- Bit 5 Tone (T)/Pulse (P)
  - 0 = Tone (T) (default)
  - 1 = Pulse (P)
- Bit 6 Reserved
- Bit 7 Originate/Answer
  - 0 = Answer
  - 1 = Originate (default)

S15 **Reserved**

S16 **General Bit Mapped Test Options**

Indicates the test in progress status.

Default: 0

- Bit 0 Local analog loopback
  - 0 = Disabled (default)
  - 1 = Enabled (&T1)
- Bit 1 Not used
- Bit 2 Local digital loopback
  - 0 = Disabled (default)
  - 1 = Enabled (&T3)
- Bit 3 Remote digital loopback (RDL) status
  - 0 = Modem not in RDL
  - 1 = RDL In progress
- Bit 4 RDL requested (AT&T6)
  - 0 = RDL not requested (default)
  - 1 = RDL requested (&T6)
- Bit 5 RDL with self test
  - 0 = Disabled (default)

1 = Enabled (&T7)

Bit 6 Local analog loopback (LAL) with self test
0 = Disabled (default)
1 = Enabled (&T8)

Bit 7 Not used

S17 Reserved

S18 Test Timer

Sets the length of time, in seconds, that the modem conducts a test (commanded by &Tn) before returning to the command line. If this register value is zero, the test will not automatically terminate; the test must be terminated from the command line by issuing an &T0 or H command.

Range: 0-255 seconds, Default: 0

S19-S20 Reserved

S21 V21/General Bit Mapped Options

Indicates the status of command options.

Default: 4 (00000100b)

Bit 0 Set by &Jn command but ignored otherwise
0 = J10 (default)
1 = &J1

Bit 1 Reserved

Bit2 CTS behavior (&Rn)
0 = CTS always on (&R0)
1 = CTS tracks RTS (&R1) (default)

Bit 3,4 DTR behavior (&Dn)
0 = &DO selected (default)
1 = &D1 selected
2 = &D2 selected
3 = &D3 selected

Bit5 RLSD (DCD) behavior (&Cn)
0 = &C0 selected (default)
1 = &C1 selected

Bit 6 DSR behavior (&Sn)
0 = &S0 selected (de1ault)
1 = &S1 selected

Bit 7 Long space disconnect (Yn)
0 = Y0 (default)
1 = Y1

S22 **Speaker/Results Bit Mapped Options**

Indicates the status of command options.

Default 117 (75h) (01110101b)

Bit 0,1 Speaker volume (Ln)
- 0 = (L0)
- 1 = Low (L1) (default)
- 2 = Medium (L2)
- 3 = High (L3)

Bit 2,3 Speaker control (Mn)
- 0 = Disabled (M0)
- 1 = Off on carrier (M1) (default)
- 2 = Always on (M2)
- 3 = On during handshake (M3)

Bit 4,5,6 Limit result codes (Xn)
- 0 = X0
- 4 = X1
- 5 = X2
- 6 = X3
- 7 = X4 (default)

Bit 7 Reserved

S23 **General Bit Mapped Options**

Indicates the status of command options.

Default 183 (B7h) (10110111b)

Bit 0 Grant RDL
- 0 = RDL not allowed (&T5)
- 1 = RDL allowed (&T4) (default)

Bit 1,2,3 Assumed DTE Rate
- 0 = 0 - 300 bps
- 1 = 600 bps
- 2 = 1200 bps
- 3 = 2400 bps (default)
- 4 = 4800 bps
- 5 = 9600 bps
- 6 = 19200 bps

Bit 4,5 Assumed DTE parity
- 0 = even
- 1 = not used
- 2 = odd
- 3 = none (default)

Bit 6,7 Guard tone (&Gn)

0 = None (&G0)
1 = None (&G1)
2 = 1800 Hz (&G2) (default)

S24 **Sleep Inactivity Timer**

Sets the length of time, in units of 10 seconds, that the modem will operate in normal mode with no detected telephone line DTE line activity before entering low-power sleep mode. The timer is reset upon any DTE line or telephone line activity. If S24 value is zero, neither DTE line nor telephone inactivity will cause the modem to enter the sleep mode.

Range: 0-255 tens of seconds , Default: 0

S25 **Delay To DTR**

Sets the length of time that the modem will ignore DTR for before hanging up. Its units are seconds for synchronous mode and one hundredths of a second for other modes.

Range: 0-255 (1 second for synchronous mode 1; 0.01 second otherwise) , Default: 5

S26 **RTS-to-CTS Delay**

Sets the time delay, in hundredths of a second, before the modem turns CTS ON after detecting an OFF-to-ON transition on S when &R0 is commanded. Pertains to synchronous operation only.

Range: 0-255 hundredths of a second, Default: 1

S27 **Bit Mapped Option**

Indicates the status of command options.

Default: 9 (00001001b)

Bit 0,1,3 Synchronous/asynchronous selection (&Mn/&Qn)

0, 0 = &M0 or &Q0
1,0 = &M1 or &Q1
2,0 = &M2 or &Q2
3,0 = &M3 or &Q3
0,1 = &Q4
1,1 = &Q5 (default)
2,1 = &Q6

Bit 2 Leased line control (&Ln)

0 = Dial up line (&L0) (default)
1 = Leased line (&L1)

Bit 4,5 Internal clock select (&Xn)
0 = Internal clock (&X0) (default)
1 = External clock (&X1)
2 = Slave clock (&X2)

Bit 6 CCITT/Bell mode select (Bn)
0 = CCITT mode (B0) (default)
1 = Bell mode (B1)

Bit 7 Reserved

S28 Bit Mapped Options

Default: 0

Bit 0 V.23 split screen (\Wn)
0 = Disabled (\W0) (default)
1 = Enabled (\W1)

Bit 1 V.23 split screen direction
0 = 75 Tx (%F0) (default)
1 = 1200 Tx (%F1)

Bit 2 V.23 half-duplex
0 = Disabled (default)
1 = Enabled (%F3)

Bit 3, 4 Pulse dialing (&Pn)
0 = 512 (&PO) (default)
1 = 1024 (&P1)
2 = 2048 (&P2)
3 = 4096 (&P3)

Bit 5 Reserved

Bit 6, 7 Reserved

S29 Flash Dial Modifier Time

Sets the length of time, in units of 10 ms, that the modem will go on-hook when it encounters the flash (|) dial modifier in the dial string. The time can be limited as it is a country dependent parameter.

Range: 0-255 10 ms intervals , Default: 0 (disabled)

S30 Disconnect Inactivity Timer

Sets the length of time, in tenths of a second intervals, that the modem will stay online before disconnecting when no data is sent or received. In error-correction mode, any data transmitted or received will reset the timer. The timer is

inoperative in synchronous mode.

Range: 0-255 tenths of a second, Default:0 (disabled)

S31 **Bit Mapped Options**

Default: 2 (00000010b)

Bit 0 Reserved

Bit 1 Controls auto line speed detection (Nn)
0 = Disabled (N0)
1 = Enabled (N1) (default)

Bit 2, 3 (Wn) Controls error correction progress messages
0 = DTE speed only (W0) (default)
1 = Full reporting (W1)
2 = DCE speed only (ATW2)

Bit 3 Reserved

Bit 4-7 Reserved

S32 **XON Character**

Sets the value of the XON character.

Range: 0-255, ASCII decimal, Default: 17

S33 **XOFF Character**

Sets the value of the XOFF character.

Range: 0-255, ASCII decimal, Default: 19

S34-35 **Reserved**

S36 **LAPM Failure Control**

Default: (00000111b)

Bit 0 - 2 This value indicates what should happen upon an LAPM failure. These fallback options are initiated immediately upon connection if S48=128. If an invalid number is entered, the number is accepted into the register, but S36 will act as if the default value has been entered.
0 = Modem disconnects
1 = Modem stays on-line and a Direct mode connection is established
2 = Reserved
3 = Modem stays on-line and a Normal mode connection is established

4 = An MNP connection is attempted and if it fails, the modem disconnects
5 = An MNP connection is attempted and it fails, Direct mode connection is established
6 = Reserved
7 = An MNP connection is attempted and if it fails, Normal mode connection is established (Default)
Bit 3 -7 Reserved

S37 Desired Line Connection Speed

Default: 0

Bit 0 - 3 Desired line connection speed. This is interlinked with the Fn command. If an invalid number is entered, the number is accepted into the register, but S37 will act as if the default value has been entered.
0 = Attempt auto mode connection (F0) (Default)
1-3 = Attempt to connect at 300 bps (F1)
4 = Reserved
5 = Attempt to connect at 1200 bps (F4)
6 = Attempt to connect at 2400 bps (F5)
7 = Attempt to connect at V.23 (F3)
8 = Attempt to connect at 4800 bps (F6)
9 = Attempt to connect at 9600 bps (F8)
10 = Attempt to connect at 12000 bps (F9)
11 = Attempt to connect at 14400 bps (F10).
12 = Attempt to connect at 7200 bps (F7)
Bit 4 - 7 Reserved

S38 Delay Before Forced Hang Up

This register specifies the delay between the modem's receipt of the H command to disconnect (or ON-to-OFF transition of DTR if the modem is programmed to follow the signal), and the disconnect operation. Applicable to error-correction connection only. This register can be used to ensure that data in the modem buffer is sent. If S38 is set to a value between 0 and 254, the modem will wait that number of seconds for the remote modem to acknowledge all data in the modem buffer before disconnecting. If time expires before all data is sent, the NO CARRIER result code will be issued to indicate that data has been lost. If all data is transmitted prior to time-out, the response to the H0 command will be OK.

If S38 is set to 255, the modem does not time-out and continues to attempt to deliver data in the buffer until the connection is lost or the data is delivered.

Range: 0-255 seconds, Default: O

S39 **Flow Control**

Default: 3 (00000011 b)

Bits 0-2 Status of command options
- 0 = No flow control
- 3 = RTS/CTS (&K3) (default)
- 4 = XON/XOFF (&K4)
- 5 = Transparent XON (&K5)
- 6 = Both methods (&K6)

Bits 3-7 Reserved

S40 **General Bit Mapped Options**

Indicates the status of command options.

Default: 105 (69h) (01101001b)

Bit 0 MNP Extended Services (-Kn)
- 0 = Disable extended services (-K0)
- 1 = Enable extended services (-K1) (default)

Bit 1 (Mn) Power Level Adjustment for Cellular Use
- 0 = Auto-adjustment (M0) (default)
- 1 = Force adjustment (M1)

Bit 2 (default) MNP Link Negotiation Speed (*Hn)
- 0 = Link negotiation at highest speed (*H0)
- 1 = Link negotiation at 1200 bps (*H1)

Bits 3-5 Break Handling (\Kn)
- 0 = \K0
- 1 = \K1
- 2 = \K2
- 3 = \K3
- 4 = \K4
- 5 = \K5 (default)

Bits 6-7 MNP block size (\An)
- 0 = 64 chars (\A0)
- 1 = 128 chars (\A1) (default)
- 2 = 192 chars (\A2)
- 3 = 256 chars (\A3)

S41 **General Bit Mapped Options**

Indicates the status of command options.

Default: 3 (00000011b)

Bits 0 - 1 Compression selection (%Cn)
- 0 = Disabled (%C0)
- 1 = MNP5 (%C1)
- 2 = V.42bis (%C2)
- 3 = MNP5 and V.42bis (%C2) (default)

Bit 2 Auto retrain
- 0 = Retrains disabled (%E0) (default)
- 1 = Retrains enabled (%E)

Bit 3 Modem-to-modem flow control
- 0 = Disabled (\G0) (default)
- 1 = Enabled (\G1)

Bit 4 Block mode control (\Ln)
- 0 = Stream mode (\LO) (default)
- 1 = Block mode (\L1)

Bit 5-7 Reserved

S41-45 **Reserved**

S46 **Data Compression Control**

Controls selection of compression. The following actions are executed for the given values:

Range: 136 or 138 , Default: 138

S46 = 136 Execute error correction protocol with no compression

S46 = 138 Execute error correction protocol with no compression (default)

S48 V.42 **Negotiation Action**

The V.42 negotiation process determines the capabilities of the remote modem. However, when the capabilities of the remote modem are known and negotiation is unnecessary, this process can be bypassed if so desired.

Range: 0, 7, or 128 if an invalid number is entered, it is accepted into the S register, but S48 will act as if 128 has been entered.

Default: 7

S48=0 Disable negotiation; bypass the detection and negotiation phases; and proceed with LAPM

S48 = 7 Enable negotiation (default)

S48 = 128 Disable negotiation; bypass the detection and negotiation phases; and proceed at once with the fallback action specified in S36. Can be used to force MNP.

S80 **Soft Switch Functions**

S80 is applicable only if the EPROM has been so customized by ConfigurACE. S80 bits are the soft equivalent of four hardware input signals. The hardware signal is indicated corresponding to each bit.

Default: 0

Bit 0 signal)	V.25 bis/AT command mode select (AT/V25B
	0 = AT selected
	1 = V.25 bis selected
Bit 1 signal)	Remote configuration permitted (REMCONF
	0 = Remote configuration not permitted
	1 = Remote configuration permitted
Bit 2 signal)	Call back security enforcement (SECACC
	0 = Call back security disabled
	1 = Call back security enabled
Bit 3	Originate /Answer select (O/A signal)
	0 = Originate selected for connections
	1 = Answer selected for connections
Bits 4-7	Reserved

S82 **Break Handling Options**

Break signals provide a way for the user to get the attention of the remote modem. The break type depends on the specific application. LAPM specifies three methods of break signal handling: in sequence, expedited, and destructive. If an invalid number is entered, it is accepted into the S register, but S82 will act as if the default value has been entered.

Range: 0, 7, or 128, Default: 128

S82 = 3 Expedited: Modem sends a break immediately; data integrity is maintained both ahead of and after the break.

S82 = 7 Destructive: Modem sends a break immediately; data being processed by each modem at the time of the break is destroyed.

S82 = 128 In sequence: Modem sends a break in sequence with any transmitted data; data integrity is maintained both ahead of and after the break. (Default.)

S86 **Call Failure Reason Code**

When the modem issues a NO CARRIER result code, a value is written to this S register to help determine the reason for a failed connection. S86 records the first event that contributes to a NO CARRIER message. The cause codes are:

Range 0, 4, 5, 9,12,13, or 14, Default: -

S86 = 0 Normal disconnect, no error occurred

S86 = 4 Loss of carrier

S86 = 5 V.42 negotiation failed to detect an error-correction modem at the other end

S86 = 9 The modems could not find a common protocol

S86 = 12 Normal disconnect initiated by the remote modem

S86 = 13 Remote modem does not respond after 10 re-transmissions of the same message

S86 = 14 Protocol violation

S91 **PSTN Transmit Level**

Sets the transmit level, in dBm, for the PSTN mode. In some countries, this level cannot be changed and there are checks to prevent transmit level change.

Range: 0 to -15 dBm, Default: 10

S92 **Fax Transmit Level**

Sets the transmit level, in dBm, for the fax mode. In some countries, this cannot be changed and there are checks to prevent transmit level change.

Range: 0 to -15 dBm, Default: 10

S95 **Extended Result Code**

The bits in this register can be set to override some of the Wn command options. A bit set to a 1 in this register will enable a corresponding result code regardless of the Wn setting.

Default: 0

Bit 0 = CONNECT result code indicates DCE speed instead of DTE speed.
Bit 1 = Append/ARQ to CONNECT XXXX result code if error correction is on.
Bit 2 = Enable CARRIER XXXX result code
Bit 3 = Enable PROTOCOL XXXX result code
Bit 4 = Reserved
Bit 5 = Enable COMPRESSION result code
Bit 6 = Reserved
Bit 7 = Reserved

S99 **Leased Line Transmit Level**

Sets the transmit level, in dBm, for the leased line mode. In some countries this cannot be changed and there are checks to prevent transmit level change.

Range: 0 to -15 dBm, Default: 10

9.6 Class 2 Fax Command Set

The SPARCbook internal modem is able to execute extended Class 2 Fax Commands, summarized in Table 9-4. These commands must be preceded by the AT characters, and may be terminated with a semicolon (;) or **Return**.

COMMAND	FUNCTION
Service Class ID	
+FCLASS=	Service Class
Class 2 Action Commands	
D	Originate a Call
A	Answer a Call
+FDT=	Data Transmission
+FET=N	Transmit Page Punctuation
+FDR	Begin or Continue Phase C Receive Data
+FK	Session Termination
Class 2 DCE Responses	
+FCON	Facsimile Connection Response
+FDCS	Report Current Session
+FDIS	Report Remote Identification
+FCFR	Indicate Confirmation to Receive
+FTSI	Report the Transmit Station ID
+FCSI	Report the Called Station ID
+FPTS	Page Transfer Status
+FET	Post Page Message Response
+FHNG	Call Termination with Status
Class 2 Session Parameters	
+FMFR?	Identify Manufacturer
+FMDL?	Identify Model
+FREV?	Identify Revision
+FDCC=	DCE Capabilities Parameters
+FDIS=	Current Sessions Parameters
+FDCS=	Current Sessions Results
+FLID=	Local ID String
+FCR	Capability to Receive
+FPTS=	Page Transfer Status
+FCR=	Capability to Receive
+FAA	Adaptive Answer
+FBUF?	Buffer Size (Read Only)
+FPHCTO	Phase C Time Out
+FAXERR	Fax Error Value
+FBOR	Phase C Data Bit Order

Table 9-4 Class 2 Fax Command Summary

Parallel Interface

10

The parallel interface on the SPARCbook is provided by the MACIO.

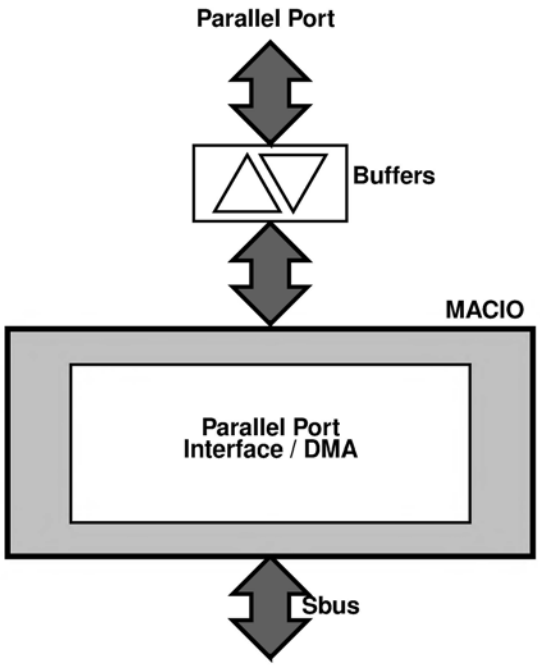


Figure 10-1 Parallel Port Architecture

10.1 Parallel Port Overview

The parallel port of the MACIO comprises four 8-bit port registers, three 16-bit configuration registers and a 64 byte FIFO. Parallel communications can be carried out using programmed I/O or DMA operations; the MACIO's internal DMA controller provides the necessary hardware support.

The direction, timing and protocol are programmable to support the wide variety of Centronics interfaces on peripheral devices.

Connection to the parallel interface on the SPARCbook is made via the supplied parallel breakout cable, which makes available a standard 25 way D-type connector. Pinout details for this connector are provided in Appendix B.

10.1.1 Parallel Port DMA Operations

The parallel port DMA can operate in either chained or unchained transfer mode. Mode selection, interrupt control and DMA control are carried out via the Parallel Port Control Register (P_CSR). The direction of transfer is controlled via the Parallel Port Transfer Register (P_TCR), and is reflected in the P_CSR.

Unchained Mode

In the unchained transfer mode, a single address register and byte counter are used. The byte counter is enabled via the P_CSR and decrements each time a byte is transferred. When the byte counter expires (from 0x1 to 0), bit 9 in the P_CSR becomes set and an interrupt generated. Unchained DMA transfers terminate in two ways: either when the byte counter expires, or when software clears bit 9 in the P_CSR.

Chained Mode

Chained DMA transfers use two additional registers to control operations. These are the NEXT Address Register and NEXT Byte Count Register. The NEXT registers provide a new address and byte count for the next DMA operation in a sequence of operations. Chaining is enabled via bits13 and 24 in the P_CSR. When the byte counter expires in the chained mode, the contents of the NEXT registers are transferred into the corresponding address and byte count registers.

The NEXT registers are located at the same addresses as their current counterparts, and can be accessed only when bit 24 in the P_CSR is set.

The NEXT registers need to be updated before the current byte count expires in order for the DMA controller to load valid contents into the address and byte counter registers. However, the NEXT Byte Counter does not need to be reprogrammed for each DMA operation when setting up a

sequence of DMA transfers of equal size from different addresses; only the NEXT Address Register need be updated and the DMA controller will simply load its new contents and the existing contents of the NEXT Byte Counter.

10.2 Parallel Port Control Registers

The parallel port registers are located at the addresses shown in Table 10-1.

Address	Register	Size	Access
0x7C800000	Control and Status Register	32	R/W
0x7C800004	Address Register	32	R/W
	NEXT Address Register	32	R/W
0x7C800008	Byte Count Register	32	R/W
	NEXT Byte Count Register	32	R/W
0x7C80000C	Test Control and Status Register	32	R/W
0x7C800010	Hardware Configuration Register	16	R/W
0x7C800012	Operation Control Register	16	R/W
0x7C800014	Parallel Data Register	8	R/W
0x7C800015	Transfer Control Register	8	R/W
0x7C800016	Output Register	8	R/W
0x7C800017	Input Register	8	R/W
0x7C800018	Interrupt Control Register	16	R/W

Table 10-1 Parallel Port Control Registers

Control and Status Register

This register provides DMA status and control for the parallel port. It is a 32-bit read-write register, which contains a number of read-only status bits and read-write control bits.

Bits 31:28 Device ID (= 1010)

Bit 27 Next Address Loaded (read-only) – this bit is set when the NEXT Address and Byte Count registers have been written but have not been used.

Bit 26 Address Loaded (read-only) – is set when the contents of the address and byte count are considered valid during chained transfers.

Bit 25	DMA On (read-only) – when set, indicates that DMA transfers are not disabled due to any hardware or software condition.
Bit 24	Enable Next (R/W) – when set enables DMA chaining, and autoloading of NEXT registers.
Bit 23	Terminal Count Interrupt Disable (R/W) – when set disables terminal count from generating an interrupt.
Bits 22:21	Unused
Bit 20	When set, disables draining and resetting of FIFO when the address register is loaded.
Bits 19:18	Pbus Burst Size – defines size of Pbus bursts
Bits 17:15	Unused
Bit 14	Terminal Count – set when the byte count expires. It is cleared by writing to logic '1'.
Bit 13	Enable Count – when set enables the byte counter.
Bits 12:10	Unused
Bit 9	Enable DMA – when set, enables DMA
Bit 8	DMA Direction 0 = to memory 1 = from memory
Bit 7	Parallel Port Reset
Bit 6	Slave Error – when set indicates that an access-size error occurred to a parallel port register
Bit 5	Invalidate FIFO (write-only).
Bit 4	Interrupt Enable – when set, enables interrupts generated by the parallel port.
Bits 3:2	Draining – both bits are set when the FIFO is draining, otherwise both bits are '0'.
Bit 1	Error Pending
Bit 0	Interrupt Pending

Display Interface

11

This chapter discusses the SPARCbook 3's display interface.

The display interface drives the built-in LCD flat panel display and is able to drive an an external CRT display with resolutions of up to 1600 x 1200 pixels. SPARCbook 3 is able to display simultaneously in the LCD and external display if they operate at the same resolution.

11.1 Display Interface Overview

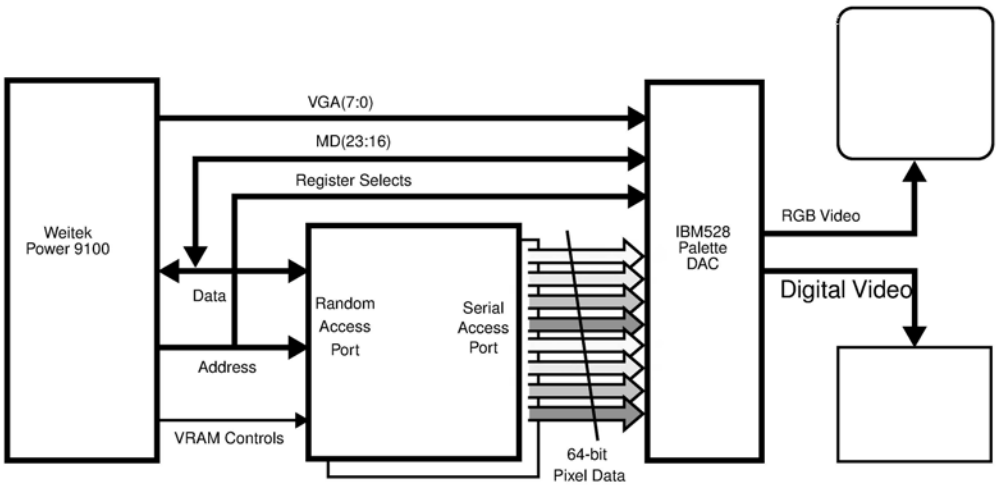


Figure 11-1 SPARCbook 3 Display Interface Architecture

11.1.1 Architecture

The display interface, illustrated in Figure 11-1, is based around two major components, the Weitek Power 9100 User Interface Controller and an IBM RGB528 palette DAC (RAMDAC). These provide advanced graphics functions and all timing and control signals required to control 2 MBytes of dual ported Video RAM (VRAM), which functions as frame buffer memory, and to drive the internal and external displays.

The Power 9100 provides the host interface between the SBus and the frame buffer, providing accelerated graphics operations for 8, 16 and 32 bits per pixel.

The frame buffer comprises four 256K x 16 bit page mode VRAM devices, although these appear to the Power 9100 hardware as eight 256K x 8 bit devices. These are organized to into a 2 Mbyte array to provide storage for image data.

The page-mode VRAM device is a type of memory chip specifically designed for use in display memories which provides a read-write random access port through which the CPU and P9100 can perform graphics operations, and a serial port through which image data is output for display by RAMDAC. Picture information is moved a page at a time from the random access area of the VRAM to the serial access memory (SAM) area within the VRAM chips and then shifted out at high speed to the RAMDAC under the control of a timing signals from the Power 9100.

The RAMDAC and frame buffer are connected via a 64-bit pixel data bus. Pixel data can be read in by the RAMDAC as eight 8-bit, four 16-bit or two 32-bit pixels at a time. Athough not implemented by the Solaris X Server, the display hardware also supports 24-bit packed pixels. The RAMDAC translates pixel data from the frame buffer into digital color information which is used to drive the LCD display directlty, or routed through three D-to-A converters to produce the analog signals required to drive an external video display.

The control register interface, which is used to configure the RAMDAC for the required operation, is accessed via locations within the Power9100 address space.

11.1.2 Display Interface Timing

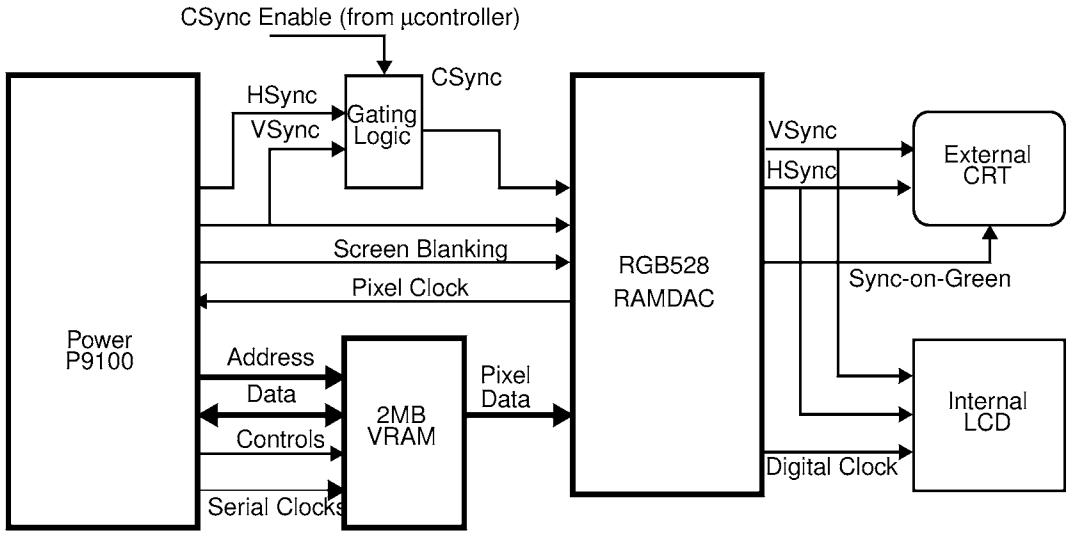


Figure 11-2 Display Interface Timing

Pixel clocks

The RAMDAC incorporates a programmable pixel clock generator, which allows the pixel clock to be programmed to suit a large range of display formats. In the SPARCbook 3 application, the RAMDAC generates a master pixel clock controlled by values in the Pixel PLL Rate Registers. The master pixel clock is fed into the P9100 which generates the horizontal and vertical synchronization signals.

The Power 9100 also provides serial clock signals which are used to clock pixel information out of the serial access port of the frame buffer. Thse are generated from divided pixel clock supplied by the RAMDAC. The frequency of divided pixel clock is a function of the number of pixels within each 64-bit transfer from the frame buffer. For example, if the pixel bus is organized as eight 8-bit pixels, the require serial clock rate must be supplied at $^1/_8$ of the dot clock frequency.

The serial clock and multiplexer rates are configurable properties via registers within the RAMDAC and Power 9100 which must be correctly programmed for the SPARCbook display interface to function correctly.

Display Synchronization

The horizontal and vertical sync signals from the Power 9100 are connected to some gating logic where they are combined to provide a composite synch signal. The composite sync and vertical sync are both fed into the RAMDAC to control display timing.

The SPARCbook 3 is able to supply an external display with separate horizontal and vertical sync signals or supply a composite sync superimposed on the green video channel (sync-on-green). Sync-on-green operation is controlled by the RAMDAC and can be enabled using the microcontroller's Control Bitport command, see Section 12.2.3, "Read/Write/Modify Commands", on page 12-11.

11.1.3 LCD Power

Power to the built-in LCD display is controlled by the microcontroller using the microcontroller's Control Bitport command, see Section 12.2.3, "Read/Write/Modify Commands", on page 12-11. In order for the display to function correctly, the digital outputs from the RAMDAC should be enbabled before enabling the TFT display.

11.2 Power 9100 User Interface Controller

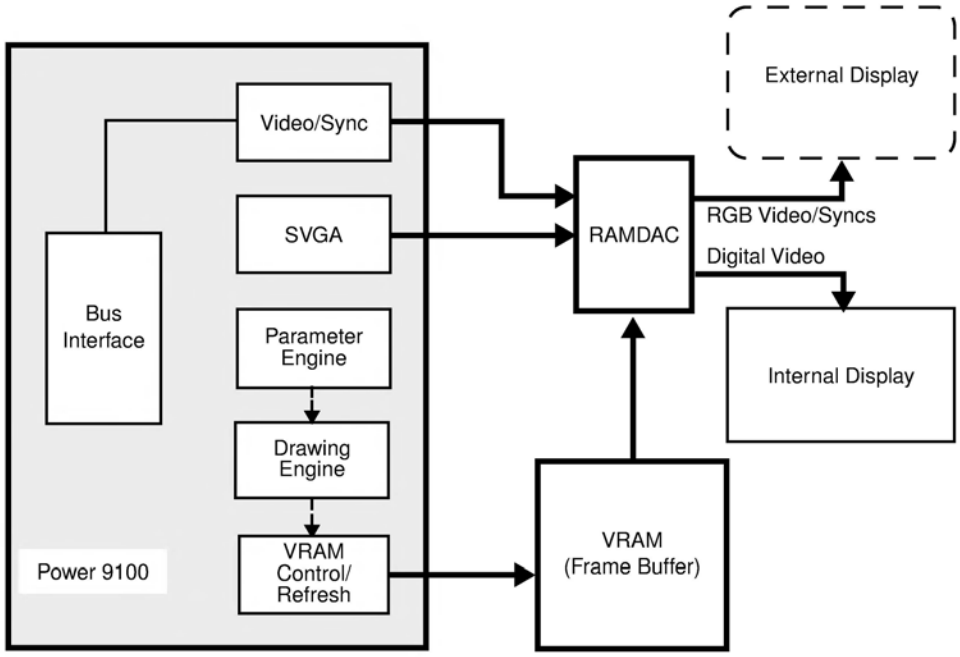


Figure 11-3 Weitek Power 9100 Block Diagram

The WEITEK Power 9100 User Interface Controller is an accelerated 2-D graphics processor which supports draw, fill, and bit block-transfer operations at the full speed of the page-mode VRAM employed in SPARCbook 3. Its operation is controlled by via internal registers and by command sent over the host interface bus. It operates as a SBus slave.

The Power 9100 contains the following functional units:

- Parameter engine
- Drawing engine
- Host interface
- Frame buffer controller
- Video controller

11.2.1 Parameter engine

The parameter engine prepares drawing operations for the drawing engine; its function is to take input coordinates from the host and convert them to a form usable by the drawing engine. The input parameters include the x,y vertices of polygons and the corners of bit block-transfer (BitBlt) regions. The parameter engine tests the vertices against window and screen boundaries, tests for exceptions, and performs trivial rejection. It transfers commands that pass these tests to the drawing engine for execution.

The parameter engine prepares four kinds of polygons for drawing. These are quadrilaterals, triangles, lines, and points. It converts points, lines, and triangles into quadrilaterals by automatically replicating vertices. For example, a point is a quadrilateral with all four vertices at the same x,y location. Only a single vertex is required to draw a point; the parameter engine passes four copies of that vertex to the drawing engine and instructs it to draw a quad.

The parameter engine also handles screen-to-screen BitBlt and two kinds of host-to-screen BitBlt operations; one is optimized for text and the other is optimized for graphics. The parameter engine handles all exception testing and host accesses to parameter engine registers and passes operations that write to the display to the drawing engine.

11.2.2 Drawing engine

The drawing engine performs three functions. These are quadrilateral drawing (quad operations), screen-to-screen BitBlt (blit operations), and host-to-screen (pixel operations).

The quad operation draws quadrilaterals in one of two modes: X11 mode, an X-Windows compatible mode; and oversized mode. Triangles, lines, and points can always be rendered correctly, but the drawing engine cannot draw horizontally convex quadrilaterals. That is, it cannot cross from the inside to the outside of the same object more than once per scan line.

The blit operation copies a rectangular area on the display from one screen location to another.

The pixel8 operation takes 8-bits-per-pixel color data (up to four pixels per word), and writes them to the frame buffer.

11.2.3 Frame buffer controller

The Power 9100 controls the frame buffer directly. Frame buffer control registers in the P9100 determine the VRAM refresh rate, the screen size, and single or double-buffering; they also select interleaved or

non-interleaved VRAM modes. These registers are initialized during system startup and but can be reprogrammed discretely via a user interface provided by the NCE Display panel.

11.2.4 SVGA unit

The Power 9100 contains an SVGA unit which is functionally independent of the main graphics engine. Although implemented in hardware, the software drivers provided for the Power 9100 for the SPARCbook 3 Solaris implementation do not support SVGA operations.

11.2.5 Power 9100 host interface

The Power 9100 provides a memory mapped host interface which supports accesses into three distinct functional areas. These are:

- Configuration registers
- Status and control registers and command locations
- Display frame buffer

Configuration registers

Basic configuration for the Power 9100's operating environment (such as host bus type, RAMDAC type and VRAM sihift register size) is controlled during power-up by a number of pullup resistors connected to the frame buffer data ouptut pins. This results in a 32-bin configuration word which is stored in the Power-up Configuration Register at 0x38000198.

During subesequent system initailization sequence, four 8-bit configuration registers must be written to set up the Power 9100 correctly for its SPARCbook 3 operating environment. These configuration registers are accessible via an index register at 0x30009100 and data register at 0x30009104. Table 11-1 summarizes these registers and the values that must be (S3 Values) used to ensure correct operation. Note that the Command register CONFIG[4], is the last register written to to enable chip operation.

Register Index	Name	Function	S3 Value
04	CONFIG[4]	Command Register (7:0)	0xA3
13	CONFIG[19]	Display Memory Base Address Register	0x00
30	CONFIG[48]	Expansion ROM Base Address Register	0x00
41	CONFIG[65]	Configuration Register	0x00

Table 11-1 Required Configuration Register Values

Status and control registers and commands

The Power 9100 contains a number of registers which are used to control device operations. They are used, for example, to enable and handle interrupts, to control parameter and drawing engine operations and to configure video and VRAM operations.

In addition status and control registers, this area of the Power 9100's host interface provides several write-only or read-only locations which are accessed to generate commands (i.e., it is the address that is interpreted as the command). In the case of write commands, any data required to execute the command is written to the command location.

Table 11-2 lists the addresses of the internal register sets and command locations.

Base Address	Register/Command Type	Access
38000000	System Control Registers	R/W
38000100	Video Control Registers	R/W
38000180	VRAM Control Registers	R/W
38000200	RAMDAC Control	R/W
38000400	Video Coprocessor Interface	R/W
38002000	Parameter Engine Status Register	R
38002004	Blit Command	R
38002008	Quad Command	R
3800200C	Pixel8 Command (Power9000 Format)	W
38002014	Next_pixels Command	W
38002080	Pixel1 Command	W
38002180	Parameter Engine Registers	R/W
38002200	Drawing Engine Pixel Processing Registers	R/W
38003000	Device Coordinate Registers	R/W
38003200	Load Coordinate Commands	R/W
38004xxxx	Pixel8 Command (Power 9100 Format)	W
38800000	Frame Buffer Memory	R/W

Table 11-2 Power 9100 Register Locations

These registers are described briefly on the following pages. However, for a detailed description of the Power 9100's registers and capabilities, please refer to the Power 9100 data book, see Appendix A, "Further Information".

Note

Due to the behavior of the Weikek Power 9100 chip, writes to any register group must be preceded by a read from the framebuffer with address bits 14:07 matching the register group's offset. Further accesses to registers within the same group can then be carried out in the normal way.

For example, you must precede a write to any of the Video Control registers at 0x380001xx, with a read from the frame buffer at 0x388001xx.

This process must be repeated each time a new register group is accessed.

11.2.6 System Control Registers

Address	Register	Function
38000004	System Configuration Register	Specifies shift control, endian swapping for pixel accesses, buffer selection for pixel accesses, test control (reserved), and P9100 version information. Note: This register must be written to first before the chip's revision can be read.
38000008	Interrupt Register	Bits in this register indicate the occurrence of interrupt conditions.
3800000C	Interrupt Enable Register	Used to enable or disable interrupts from the P9100
38000010	Alternate Read Bank Register	Alternate bus read bank select.
38000014	Alternate Write Bank Register	Alternate bus write bank select

Table 11-3 System Control Registers

System Configuration Register

The bits in this register are assigned as follows:

Bits 31 Reserved

Bits 30:29 Shift Control 3
00 = no add
01 = add 1024
01 = add 2048
11 = add 4096

Bits 28:26 Pixel size
010 = 8 bits per pixel
011 = 16 bits per pixel
111 = 24 bits per pixel
101 = 32 bits per pixel

Bit 25 Reserved

Bit 24 Drive Load 2
0 = Normal drive load
1 = Double drive load

Bit 23 Clocking – should be zero

Bits 22:20 Shift control 0

Bits 19:17 Shift control 1

bits 16:14 Shift control 2
The four shift control fields are used to define the resolution, in bytes, of the horzontal display.

To determine which values to place in these feilds: first determine the number of pixels in a horizontal scan line; next, multiply this by the number of bytes per pixel (set in Pixel Size field); finaly, set the shift control fields so that the total adds up to the number of bytes per scan line. There are many possible combinations of which four examples are shown below:

Horizontal Resolution	Bits per Pixel	Horizontal Bytes	Shift 0	Shift 1	Shift 2	Shift 3
640	8	64	101	011	000	00
			011	000	101	00
1280	24	3840	110	101	100	10
			111	100	101	01

Table 11-4 Example Horizontal Resolution Settings

Bits 13:11 Endian swapping for frame buffer access– set as follows:
Use 000 for 8 bits per pixel mode
Use 010 for 16 bits per pixel mode
Use 110 for 24 and 32 bits per pixel modes

Bit 10 Buffer select for read
0 = buffer 0
1 = buffer 1

Bit 9 Buffer select for write
0 = buffer 0
1 = buffer 1

Bits 8:3	reserved
Bits 2:0	P9100 version

Interrupt register

This register records events that have triggered an interrupt. A bit when it is asserted to indicate the cause of interrupt stays asserted until cleared by a host write. Each bit is preceded by a field control bit which must be set to change the field . The bits in this register are assigned as follows:

Bits 31:06	Reserved
Bits 5	Vblank field write control
Bit 4	Vblank 0 = not done 1 = Vblank done
Bits 3	Pick field write control
Bit 4	Pick 0 = not done 1 = Pick done
Bits 1	Idle field write control
Bit 0	Drawing engine idle 0 = Drawing engine busy 1 = Drawing engine idle

Interrupt enable register

This used to enable interrupts. Each bit is preceded by a field control bit which must be set to change the field.The bits in this register are assigned as follows:

Bits 31:08	Reserved
Bit 7	Master enable write control
Bit 6	Master enable 0 = disable all interrupts 1 = enable interrupts
Bits 5	Vblank write control
Bit 4	Vblank 0 = Disable Vblank interrupts 1 = Enable Vblank interrupts
Bits 3	Pick write control
Bit 2	Pick

0 = Disable Pick interrupts
1 = Enable Pick Interrupts

Bits 1:0 Drawing engine idle
0 = Disable drawing engine idle interrupt
1 = Enable drawing engine idle interrupt

Alternate Read and Alternate Write Bank registers

These two registers specify the seven high-order address bits when the host uses an alternate apperture to read from or write to the frame buffer.

Bits 31:23 Reserved

Bits 22:16 Alternate Bank

Bits 15:0 Reserved

11.2.7 Video Control Registers

These registers control the video interface of the P9100. They are summarized below:

Address	Register	Function
	Horizontal Timing	
38000104	Horizontal Counter	This read-only register is used by the P9100 to count dot clocks to derive the current pixel along a horizontal trace.
38000108	Horizontal total	This is used by the host to specify the total horizontal line length.
3800010C	Horizontal sync rising edge	This register is used to specify where along the horizontal trace the rising edge of the horizontal sync signal occurs.
38000110	Horizontal blank rising edge	This register is used to specify where along the horizontal trace the rising edge of the horizontal blanking signal occurs.
38000114	Horizontal blank falling edge	This register is used to specify where along the horizontal sweep the falling edge of the horizontal blanking signal occurs.
38000118	Horizontal counter preload	This read-write register is used by the host to specify the value to load into the Horizontal Counter when the horizontal or vertical sync signal occurs.
	Vertical Timing	
3800011C	Vertical Counter	This read-only register is used by the P9100 to count horizontal sync signal to derive the current pixel along a vertical trace.

Table 11-5 Video Control Registers Summary

Address	Register	Function
38000120	Vertical Length	This is used by the host to specify the number of lines in a vertical trace.
38000124	Vertical sync rising edge	This register is used to specify where along the vertical trace the rising edge of the horizontal sync signal occurs.
38000128	Vertical blank rising edge	This register is used to specify where along the vertical trace the rising edge of the vertical blanking signal occurs.
3800012C	Vertical blank falling edge	This register is used to specify where along the vertical trace the falling edge of the vertical blanking signal occurs.
38000130	Vertical counter preload	This read-write register is used by the host to specify the value to load into the Vertical Counter when the vertical sync signal occurs.
	Screen repaint	
38000134	Screen repaint address	This read-only register specifies the next VRAM address to be loaded into SAM. The address corresponds to bits (21:10) of the linear address (the lower address bits are always zero). The value occupies the lower 12 bits.
38000138	Screen repaint timing control1	Specifies controls for screen refresh – **must contain 0x1A3** – see below.
3800013C	QSF Counter	This read-only register counts the QSF signals from the VRAMs to keep track of which part of the SAM is being shifted out. It is loaded with zero after every read transfer and is incremented by each dot clock
38000140	Screen repaint timing control2	Specifies controls for screen refresh – **must contain 0x5** – see below.

Table 11-5 Video Control Registers Summary (Continued)

Screen Repaint Timing Control Register 1

The bits in this register are assigned as follows (required settings are shown in bold type):

Bits 31:11 Reserved

Bits 10:9 SRADDR increment size
- **00 = 256**
- 01 = 512
- 10 = 1024

Bit 8 VSYNC source
- 0 = External
- **1 = Internal**

Bit 7 HSYNC Source
0 = External
1 = Internal

Bit 6 Reserved - must be set to 0

Bit 5 Enable Video
0 = Blanks asserted, HSYNC and VSYNC are disabled
1= Normal operation

Bit 4 Screen repaint mode
0 = Normal
1 = Restricted

Bit 3 Buffer for display
0 = Buffer 0
1 = Buffer 1

Bits 2:0 QSF counter position
Set to **011**

Screen Repaint Timing Control Register

The bits in this register are assigned as follows (required settings are shown in bold type):

Bits 31:4 Reserved

Bits 3:2 External VSYNC polarity control
00 = High true polarity
01 = Low true polarity
10 = Forced low
11 = Forced high

Bits 1:0 External HSYNC polarity control
00 = High true polarity
01 = Low true polarity
10 = Forced low
11 = Forced high

11.2.8 VRAM control registers

The VRAM control registers specify the timing and chip configuration of the P9100 VRAM interface. These are summarized below:

Address	Register	Function
38000184	Memory Configuration	This read-write register specifies how the framebuffer is configured. **Must contain 0xC008007D**.
38000188	Refresh Period	This read-write register uses the lower 10 bits to specify the refresh period for the VRAM array.
3800018C	Refresh Count	The lower 10 bits of this read-only register contains the number of elapsed cycles between refreshes.
38000190	RAS Low Maximum	This read-write register uses the lower 10 bits to specify the maximum amount of time that the RAS signal can be asserted. **Must contain 0x3FF**.
38000194	RAS Low Current	The lower 10 bits in this register controls the amount of time that the RAS signal can be asserted.

Table 11-6 VRAM Control Register Summary

11.2.9 Parameter engine registers

The parameter engine registers include the Device Coordinate registers, Status Register and Control and Condition register.

Device Coordinate Registers

The Device Coordinate registers supply the screen coordinates for a drawing operation. Coordinates can be absolute or reltaiveto the window offset. Located at base address 0x38003000, there are four X and four Y registers each for absolute and relative coordinates. These are summarized below in Table 11-7.

Address	Function
38003008	32-bit Absolute value for X[0]
38003010	32-bit Absolute value for Y[0]
38003018	16-bit Absolute value for X[0] and 16-bit Absolute value Y[0]
38003028	32-bit Relative value for X[0]
38003030	32-bit Relative value for Y[0]
38003038	16-bit Relative value for X[0] and 16-bit Absolute value Y[0]

Table 11-7 Device Coordinate Register Addresses

Address	Function
38003048	32-bit Absolute value for X[1]
38003050	32-bit Absolute value for Y[1]
38003058	16-bit Absolute value for X[1] and 16-bit Absolute value Y[1]
38003068	32-bit Relative value for X[1]
38003070	32-bit Relative value for Y[1]
38003078	16-bit Relative value for X[1] and 16-bit Absolute value Y[1]
38003088	32-bit Absolute value for X[2]
38003090	32-bit Absolute value for Y[2]
38003098	16-bit Absolute value for X[2] and 16-bit Absolute value Y[2]
380030A8	32-bit Relative value for X[2]
380030B0	32-bit Relative value for Y[2]
380030B8	16-bit Relative value for X[2] and 16-bit Absolute value Y[2]
380030C8	32-bit Absolute value for X[3]
380030D0	32-bit Absolute value for Y[3]
380030D8	16-bit Absolute value for X[3] and 16-bit Absolute value Y[3]
380030E8	32-bit Relative value for X[3]
380030F0	32-bit Relative value for Y[3]
380030F8	16-bit Relative value for X[3] and 16-bit Absolute value Y[3]

Table 11-7 Device Coordinate Register Addresses (Continued)

Status register

The Status register is a 32-bit read-only register located at 0x38002000 which provides information about the current state of the drawing engine.

Bit 31 — Quad/blit command
0 = ready for a quad or blit operation
1 = do not start a quad or blit operation

Bit 30 — 1 = drawing engine busy (do not start pixel1 or pixel8 operations)

Bits 29:8 — Not used, all zeros.

Bit 7 — 1 = Pick detected

Bit 6 — 1 = Pixel exception, operation must be done in software

Bit 5 — 1 = Blit command exception, operation must be done in software

Bit 4	1 = Quad exception, operation must be done in software
Bit 3	1 = Requested quad is concave
Bit 2	1= Source coordinates for quad draw entirely outside clipping window
Bit 1	1= Source coordinates for quad draw entirely inside clipping window
Bit 0	1= Source coordinates for quad draw straddle clipping window

Control and Condition Registers

These registers control and monitor P9100 operations. The addresses and function of these registers is summarized in Table 11-8.

Address	Name	Function
38002184	Out of Range Register	This read-only register identifies x and y values that are out of range for the current operation
3800218C	Index Register	This read-write register supplies the current index into the x and y coordinates as a 2-bit binary integer. The value is used for meta-coordinate computation.
38002190	Window Offset XY	This read-write register supplies the x and y offsets of the current window on the display as two packed 16-bit binary integers.
38002194	Parameter Engine Window Minimum	These read-write registers contains an exact copy of the window minimum and maximum registers respectively.
38002198	Parameter Engine Window Maximum	
380021A0	Y Clip Register	These read-only registers provide the results of clip checks. The P9100 updates these registers for every new set of x and y values loaded into the coordinate registers.
380021A4	X Clip Register	
380021A8	X Edge Less Than Register	These read-only registers provide the results of checks on the vertices of a requested drawing operation. The P9100 updates these registers for every new set of x and y values loaded into the coordinate registers.
380021AC	X Edge Greater Than Register	
380021B0	Y Edge Less Than Register	
380021B4	Y Edge Greater Than Register	

Table 11-8 Control and Condition Registers

The control and condition registers are all 32-bit registers with valid data contained only within the least significant byte – bits (7:0). A bit contains a '1' when the test condition is true.

Register	Bit							
	7	6	5	4	3	2	1	0
Out of Range (oor)	x_oor				y_oor			
	x[3]	x[2]	x[1]	x[0]	y[3]	y[2]	y[1]	y[0]
X Clip	x less than min				x greater than max			
	x[3]	x[2]	x[1]	x[0]	x[3]	x[2]	x[1]	x[0]
Y Clip	y less than min				y greater than			
	y[3]	y[2]	y[1]	y[0]	y[3]	y[2]	y[1]	y[0]
X Edge Less Than	0	0	x[0]>x[2]	x[1]>x[3]	x[3]>x[0]	x[2]>x[3]	x[1]>x[2]	x[0]>x[1]
X Edge Greater Than	0	0	x[0]<x[2]	x[1]<x[3]	x[3]<x[0]	x[2]<x[3]	x[1]<x[2]	x[0]<x[1]
Y Edge Less Than	0	0	y[0]>y[2]	y[1]>y[3]	y[3]>y[0]	y[2]>y[3]	y[1]>y[2]	y[0]>y[1]
Y Edge Greater Than	0	0	y[0]<y[2]	y[1]<y[3]	y[3]<y[0]	y[2]<y[3]	y[1]<y[2]	y[0]<y[1]

Table 11-9 Vertex Checking Register (Significant Bits)

11.2.10 Drawing engine registers

The drawing engine registers control pixel processing. They are used define foreground and background colors, plane masking, pattern and pattern origins, pixel drawing window limits, and raster operation parameters. These registers can only be read or written while the drawing engine is idle (when the Busy bit in the status register contains ‘0’.

The registers in this category are summarized in Table 11-10.

Address	Name	Function
38002200	Foreground Color	Contains an 8-bit value that specifies the foreground color
380020204	Background Color	Contains an 8-bit value that specifies the background color
38002208	Plane Mask	Contains an 8-bit mask in bits 7:0. Bit 0 corresponds to plane 0. A plane is enabled when the associated bit is set.
3800220C	Draw Mode	Controls writing within a picked window and selects the destination buffer for drawing operations.
38002210	Pattern Origin x	Contain the 4-bit x and y origins of a 16 x 16-bit pattern to be used by the quad command to fill polygons on the display.
38002214	Pattern Origin y	
38002218	Raster	Specifies the parameters for a raster operation
3800221C	Pixel8	Provides temporary storage for excess bits from a Pixel8 operation. The data is stored in the lower part of the register
38002220	Window Minimum	These supply the window limits that the pixel processor hardware uses to determine whether a pixel is to be drawn
38002224	Window Maximum	

Table 11-10 Drawing Engine Registers Summar

Address	Name	Function
38002280	Pattern [0]	Used to specify the pattern for a quad fill.
38002284	Pattern [1]	
38002288	Pattern [2]	
3800228C	Pattern [3]	
38002290	User [0]	User defined registers
38002294	User [1]	
38002298	User [2]	
3800229C	User [3]	
380022A0	Byte Window Minimum	
380022A4	Byte Window Minimum	

Table 11-10 Drawing Engine Registers Summar (Continued)

Draw Mode Register

The bits in the draw mode register are assigned as follows:

Bits 31:04 Reserved (contain zeroes)

Bits 3:2 Pick Write Enable
0 = Suppress any writes outside of the window
1 = Suppress all writes; set the picked bit in the interrupt register if the write is inside the window.

Bits 1:0 Definition buffer
0 = Select buffer 0 as destination for drawing engine operations
1 = Select buffer 1 as destination for drawing engine operations

Raster Register

Bits 31:18 Reserved (contain zeroes)

Bit 17 Pattern Enable
0 = Disable pattern
1 = Enable pattern

Bit 16 Quick Draw Mode
0 = X11 Mode
1 = Oversized mode

Bits 15:0 Minterms for boolean raster operations

Window Minimum/Maximum Registers

The bits in these registers are assigned as follows:

Bits 31:29 Reserved (contains zeroes)

Bits 28:16 Minimum or maximum x values

Bits 15:13 Reserved (contains zeroes)

Bits 12:00 Minimum or Maximum y values

11.2.11 RAMDAC register accesses

Accesses to the RAMDAC's internal registers are made through locations within the Power 9100's address space, see Section 11.4.1, "RAMDAC host interface", on page 11-22

11.3 Direct frame buffer access

To the host, the framebuffer appears as a normal 2 Mbyte linear memory array. With 8 bit pixels, each byte in the framebuffer represents 1 pixel on the display and the lowest addressed location in the framebuffer appears at the top left of the display; that is the pixel at the top left of the display (pixel 0) is addressed at 0x30200000. The top left pixel on the display is fixed at this address in memory. Panning and zooming is carried out by redrawing the image in the framebuffer.

Pixel data is packed into the framebuffer so that adjacent lines on the display start at line-length address intervals. For example, the first pixel on the second line of a 640 x 480 pixel display would be addressed at 0x38800280. Figure 11-4 illustrates the framebuffer addressing for two

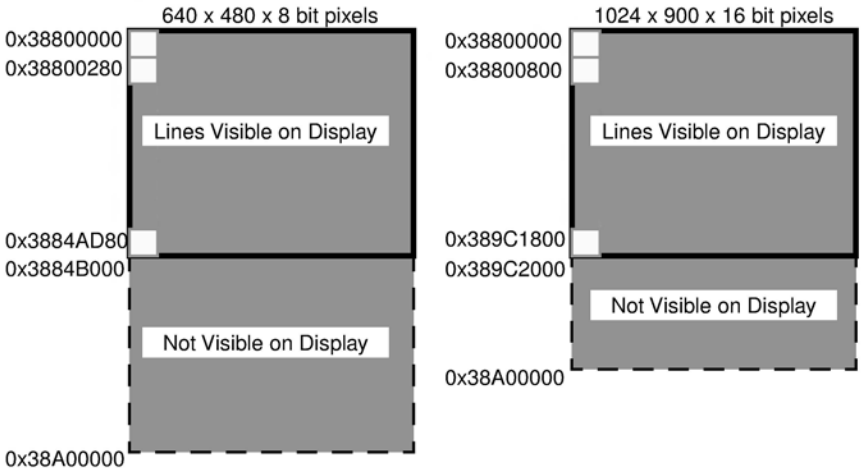


Figure 11-4 Frame Buffer Addressing

sizes of display: a 640x480 pixel display using 8 bits per pixel; and a 1024 x 900 display using 16 bits per pixel. The framebuffer is considered in the illustration as a rectangular area with a horizontal dimension equivalent to the horizontal resolution of the display. In both examples there results an undesplayed area of the framebuffer memory which can be used for other data storage.

11.4 RAMDAC

The RGB528 Palette DAC (RAMDAC) is a highly integrated device that combines color palettes, a hardware cursor, digital outputs to drive an LCD display panel, three digital to analog converters to drive a CRT display, and a clock generator.

The the SPARCbook 3 implementation, the device supports a 64-bit pixel data bus from the frame buffer; twenty-four of upper 64 data pins are dual-function and are used as digital outputs to support the TFT display.

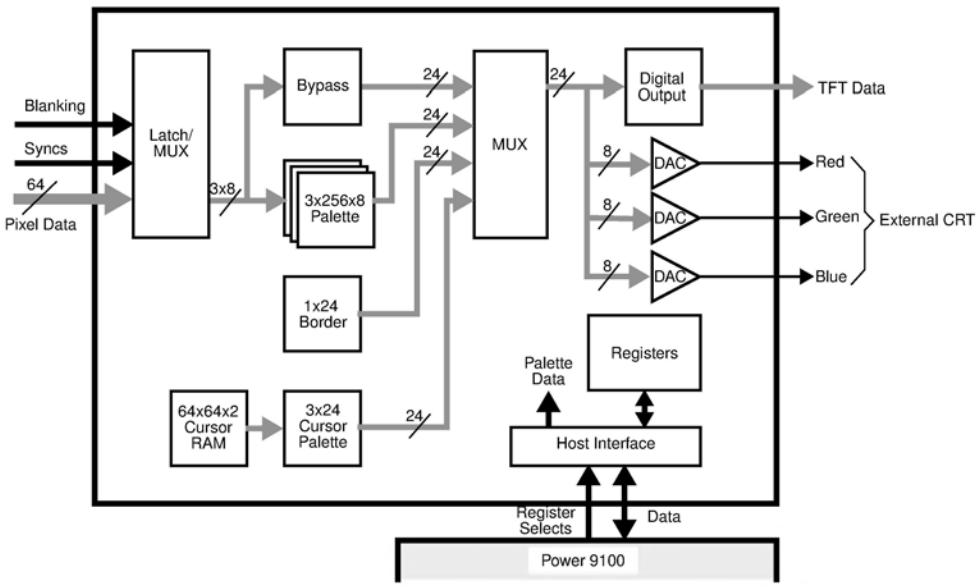


Figure 11-5 RAMDAC Architecture

11.4.1 RAMDAC host interface

The RAMDAC host interface is mapped through the RAMDAC space of the P9100 at 0x38000200. In response to addresses in the RAMDAC region, the Power 9100 generates register select signals to select the target register.

The RAMDAC provides a byte-wide control interface allowing control registers and color lookup tables to be accessed. There are eight register locations that can be accessed directly. All other registers are accessed by first writing to internal index registers to specify the control register or cursor register to be accessed.

Note

It is necessary to insert software delays between accesses to the RAMDAC registers and palettes. A convenient way of doing this is to read the Power 9100 configuration register between each RAMDAC access.

The RAMDAC internal locations can be accessed using the addresses shown in Table 6-13.

CPU Address	Register
38000200	Palette Address Register (write mode)
38000208	Palette Data
38000210	Pixel Mask
38000218	Palette Address (read mode)
38000220	Register Index Low
38000228	Register Index High
38000230	Register Data
38000238	Index Control

Table 11-11 Directly Addressed RAMDAC Registers

11.4.2 Control register accesses

The control registers are used to program the RAMDAC to suit its operating environment. Setable properties include pixel unpacking and pixel depth, serial clock and multiplexor rates, and blinking.

The column ‘INDEX’ in Table 11-12 shows the value that should be written into the Index High and Index Low Registers in order to read or write internal locations. For example, to write to the Power Management register at index 0x0005, first the value 0x05 would be written Index Low

register at 0x38000220 and the value 0x00 would be written into the Index High register at 0x38000228, and then the read enable register could be accessed at 0x38000230.

The column ‘CONTENTS’ shows the values which must be used for the SPARCbook 3 to operate correctly. Values are shown for 8 , 16, 24 and 32 bits per pixel (BPP), where these differ.

The column ‘ORDER’ shows the order in which the registers are programmed during initialization. It is recommended that this sequence is adhered to to ensure correct operation.

Note

The RAMDAC provides a register index autoincrement facility which simplifies accesses to the control registers. This mechansim is usefull when writing to the cursor array, for example, but is not recommended when initailizing SPARCbook 3 because the registers are not programmed in index sequence.

Order Accessed	Register		Contents			
	Index	Name	8 BPP	16 BPP	24 BPP	32 BPP
	0000	Revisions Register				
	0001	ID Register				
4	0002	Miscellaneous Clock Register	0x7	0x5	0x27	0x3
5	0003	Synchronization Control	0			
6	0004	Horizontal Sync Position	0			
7	0005	Power Management				
8	0006	DAC Operation	Sync on green etc..			
9	0007	Palette Control	0			
15	0008	System Clock Control	0x01			
	0009	Reserved				
10	000A	Pixel Format	0x03	0x04	0x05	0x06
11	000B	8 BPP Control	0x0			
	000C	16 BPP Control	0xC6			
	000D	24 BPP Packed Control	0x01			
	000E	32 BPP Control	0x03			

Table 11-12 RAMDAC Indexed Registers

Order Accessed	Register		Contents			
	Index	Name	8 BPP	16 BPP	24 BPP	32 BPP
	000F	Buffer A/B Select				
12	0010	Pixel PLL Control 1	0x2			
13	0011	Pixel PLL Control 2	0x00			
	0012 - 0013	Reserved				
14	0014	Fixed Pixel PLL Reference Divider	0x05 (Manditory for 10 MHz)			
16	0015	System PLL Reference Divider	0x05			
17	0016	System PLL VCO Divider	0x63 (for 50 MHz			
	0017 - 001F	Reserved				
	0020	F1 (M0)				
	0021	F1 (N0)				
	0022	F2 (M1)				
	0023	F3 (N1)				
	0024	F4 (M2)				
	0025	F5 (N2)				
	0026	F6 (M3)				
	0027	F7 (N3)				
	0028	F8 (M4)				
	0029	F9 (N4)				
	002A	F10 (M5)				
	002B	F11 (N5)				
	002C	F12 (M6)				
	002D	F13 (N6)				
	002E	F14 (M7)				
	002F	F15 (N7)				
	0030	Cursor Control				
	0031	Cursor X Low				
	0032	Cursor X High				
	0033	Cursor Y Low				
	0034	Cursor Y High				

Table 11-12 RAMDAC Indexed Registers (Continued)

Order Accessed	Register		Contents			
	Index	Name	8 BPP	16 BPP	24 BPP	32 BPP
	0035	Cursor X Hotspot				
	0036	Cursor Y Hotspot				
	0037 - 003F	Reserved				
	0040	Cursor Color 1 Red				
	0041	Cursor Color 1 Green				
	0042	Cursor Color 1 Blue				
	0043	Cursor Color 2 Red				
	0044	Cursor Color 2 Green				
	0045	Cursor Color 2 Blue				
	0046	Cursor Color 3 Red				
	0047	Cursor Color 3 Green				
	0048	Cursor Color 3 Blue				
	0049 - 005f	Reserved				
	0060	Border Color Red				
	0061	Border Color Green				
	0062	Border Color Blue				
	0063 - 006F	Reserved				
1	0070	Miscellanous Control 1	0x11			
2	0071	Miscellanous Control 2	0x45			
3	0072	Miscellanous Control 3	0x00	0x00	0x80	0x80
	0073	Miscellanous Control 4				
	0074 - 0081	Reserved				
	0082	DAC Sense				
	0083	Reserved				
	0084	MISR Red				
	0085	Reserved				
	0086	MISR Green				
	0087	Reserved				

Table 11-12 RAMDAC Indexed Registers (Continued)

Order Accessed	Register		Contents			
	Index	Name	8 BPP	16 BPP	24 BPP	32 BPP
	0088	MISR Blue				
	0089 - 008d	Reserved				
	008E	Pixel PLL VCO Divider Input				
	008F	Pixel PLL Reference Divider Input				
	0090	VRAM Mask 0				
	0091	VRAM Mask 1				
	0092	VRAM Mask 2				
	0093	VRAM Mask 3				
	0094-00ff	Reserved				
	0100 - 04FF	Cursor Array				
	0500 - 07FF	Reserved				

Table 11-12 RAMDAC Indexed Registers (Continued)

A brief description is given below of the register accessed during initialization. For a full description, please refer to the manufacturer's data sheet, see Appendix A, "Further Information".

Miscellaneous Control 1 – Index 0x0070

This register must contain the value 0x11. This configures the RAMDAC to disable the SENSE pin and for a 64-bit pixel bus.

Miscellaneous Control 2 – Index 0x0071

This register mus contain 0x45. This configures the RAMDAC to use the internal PLL clock source, to use 8-bit color resolution (as apposed to 6-bit for VGA compatibility), and to use the VRAM pixel port inputs.

Miscellaneous Control 3 – Index 0x0072

This register must contain 0x00 when using 8 or 16 BPP but must contain 0x80 when using 24 or 32 BPP. This causes the RAMDAC to swap the red and blue pixels to correct for endian differences between the frame buffer and RAMDAC.

Miscellaneous Clock Control – Index 0x0002

Bits 3:1 Control the clock divider which means that the setting of this register depends on the pixel depth in use. Valid settings are:

001 Pixel Out/2 = 32 BPP
010 Pixel Out/4 = 16 BPP
011 Pixel Out/8 = 8 BPP

Bit 0 Used to enable the internal PLL. Always = 1.

Synchronization Control – Index 0x0003

This register controls how the RAMDAC uses the sync input signals and how it drives the sync outputs. This register must contain 0x00. to operate correclty within its hardware environment.

Horizontal Sync Position – Index 0x0004

This register specifies the number of pixel clocks by which the horizontal or composite sync are delayed. Initailly this register contains 0x00 which configures the RAMDAC to drive the TFT display and most CRT displays correctly.

Power Management – Index 0x0005

This register is used to control the power management facilities. Initally it contains 0x00 to activate all internal circuitry.

DAC Operation – Index 0x0006

Only bits 3 and 0 have any function within the SPARCbook 3 environment. Bit 3 is used to enable or disable sync-on-green, and bit 0 controls the blanking pedestal. Initially, this register is contains 0x00

Palette Control – Index 0x0007

This register is used to control the way the palette is addressed by incoming pixel data. initially this register contains 0x00 to select linear palette mode (bit 7).

System Clock Control – Index 0x0008

This register is used to control the SYSCLK phase locked loop and signal pin. Bit 6 is programmed to zero to enable the SYSCLK pin, bit 1 is programmed to zero to select SYSCLK PLL as an output, and bit 0 is set to enble SYSCLK PLL programming.

Pixel Format – Index 0x00A

Bits 2:0 Defines how many bits there are per pixel (BPP) as follows:

011 = 8 BPP
100 = 16 BPP
101 = 24 BPP
111 = 32 BPP

Pixel PLL Control 1 and 2 – Index 0x0010 and 0x0011

Pixel PLL control 1 is programmed with 0x 02. This selects a mode in which the pixel PLL control 2 selects one of the F0-F15 registers to use as the pxel PLL VCO divider. The divider register is used to prescale the PLL pixel PLL reference clock.

Fixed Pixel PLL Reference Divider – Index 0x014

This register is programmed with the value 0x05 to supply to correct divider for ths SPARCbook 3 application.

System PLL Reference Divider – Index 0x015

This register is programmed with the value 0x05 to supply to correct divider for ths SPARCbook 3 application.

System PLL VCO Divider – Index 0x016

This register is programmed with the value 0x63 to supply to correct divider for ths SPARCbook 3 application.

11.4.3 Color palette accesses

The RAMDAC contains three 256 x 8 bit color lookup tables (palettes) which are accessed as a single 256 x 24 bit palette via an internal control mechanism. Before reading from the palette, the Palette Address (Read Mode) register must be written to specify the starting address. Conversely, the Palette Address (Write Mode) must be written before writting to the palette. It should be noted that writing to either Palette Address register changes the contents of both.

The RAMDAC contains address counters which simplify palette accesses by keeping track of access cycles. The first counter is reset to zero whenever either Palette Address Register is written to and incremented each time an access to the color lookup table is made. The second address count increments the address register after each access to the blue entry. This is illustrated in Figure 6-8.

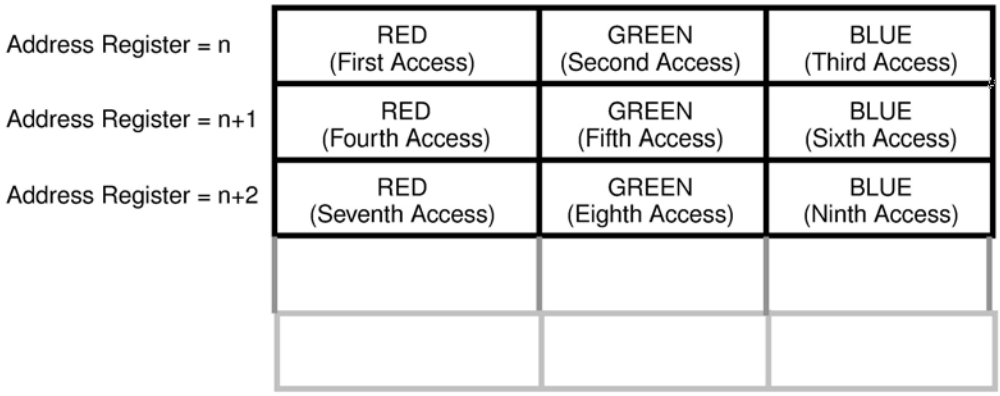


Figure 11-6 Multiple Access Support

For example, to write a new set of color information to the main palette would require the following steps.

First, an initial value is written to the Palette Address (Write Mode) address register at 0x38000200 to specify the first location in the color palette to be accessed. Next, a byte of red color information is written to the Palette Data location at 0x38000208 followed by a green byte, followed by a blue byte. The RAMDAC transfers all 24 bits of color information into the addressed

palette location in a single transfer. This mechanism allows successive byte writes to the Palette Data location to step through the 256 palette entries without the need to supply a new address for each access.

The address register is reset to zero following an access to the blue color lookup entry at 0xFF.

11.4.4 Pixel Formats

The RAMDAC is able to operate with 8, 16, 24 or 32 bit pixel data.

8 BPP

With 8 bits per pixel format, eight pixels are obtained for each pixel port data cycle. The 8 bit pixel data can used to index entries in the color palettes (indirect color mapping) or can be passed directly through to the DACs (direct color mapping). However, in this case the DAC for each color channel receives the same data resulting in a monochrome image on the external display.

16 BPP

When 16 bit pixels are in use, four pixels are obtainde for each pixel port data cycle. Color information can be organized as 5 bits for each color (555) or 5 bits red, 6 bits green, and 5 bits blue (565 format). The 16 bits are expanded to 24 bits under control of the 16 BPP control register.

Bit 15 pixel data can be used to dynamically select between indriect or direct color mapping.

24 BPP

When 24 BPP packed is selected, each pixel port cycle reads in $2^2/_3$ pixels. Every three consecutive cycles contains 8 pixels of 3 bytes each. Color mapping can be direct or indirect.

32BPP

With 32 BPP selected, two pixels are read in for each pixel port cycle. Color data is contained in the lower 24 bits, providing 8 bits each for red, green and blue channels. Bit 24 can be used to dynamically select between direct and indirect color mapping on a pixel by pixel basis.

Microcontroller Subsystem

12

This chapter decribes the microcontroller subsystem which provides low-level system management for the SPARCbook 3. System management includes the following:

- Monitoring and controlling initial system power-up
- Handling keyboard and pointing stick input
- Monitoring battery condition and charging
- Monitoring the unit's thermal envorinment.
- Handling RTC and EEPROM accesses
- Controlling TFT screen brightness
- LCD status screen character generation
- Case microswitch monitoring

12.1 Microcontroller subsystem overview

The architecture of the microcontroller subsystem is illustrated in Figure 12-1.

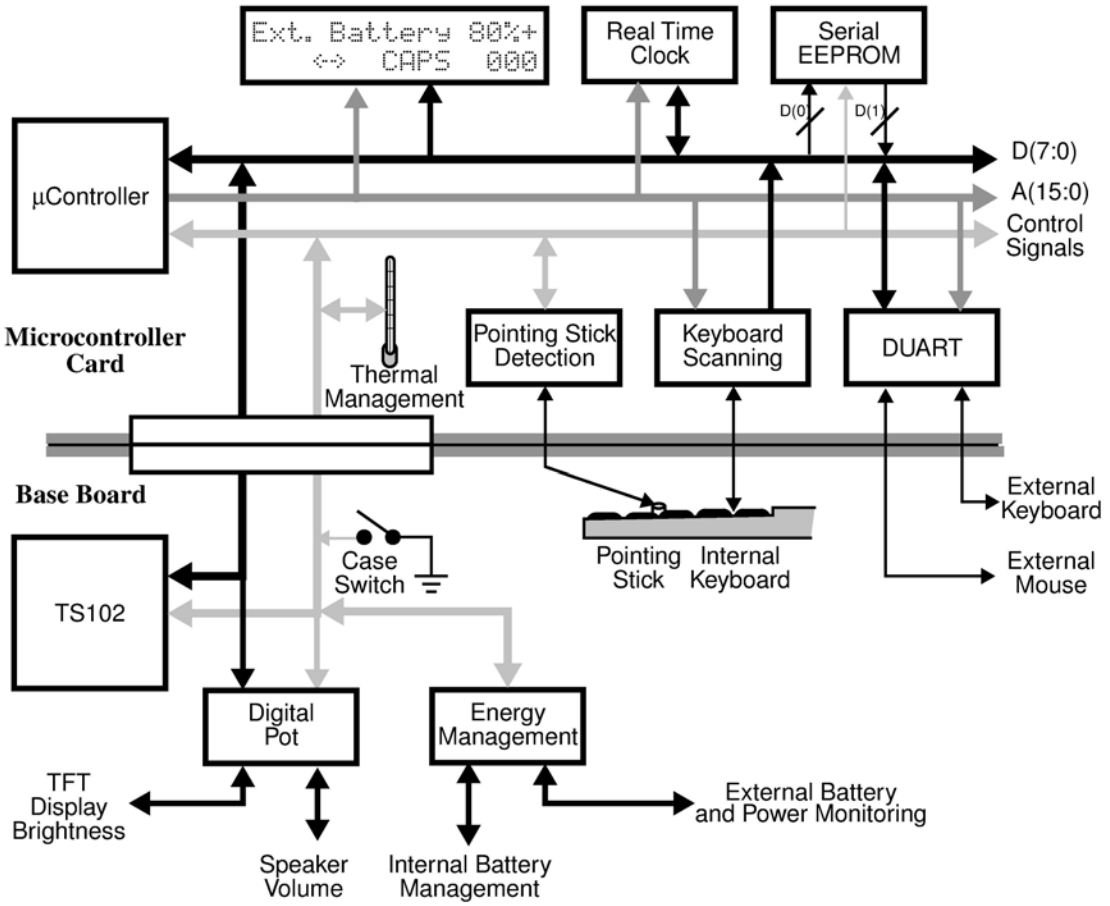


Figure 12-1 Microcontroller Subsystem Architecture

The illustration above shows scope of the microcontroller subsystem's role. The microcontroller incorporates on-chip non-volatile storage and is factory programmed with dedicated system management software.

Assigning low level activites activities to a dedicated microcontroller in this way releives the main CPU of the burden of trivial activities, ensuring that it is employed effeciently running applications.

The host CPU and microcontroller communicate via register locations within the TS102 address space. The CPU places command bytes into the microcontroller write register, the microcontroller interprets and executes the command, if possible, and then returns status information via the microcontroller read register.

12.1.1 Normal operation

Once the initial bootstrapping procedure is complete, the microcontroller executes its main code loop. This loop incorporates the the following operations:

1. Checking for host commands from TS102.
2. Checking for keyboard and mouse commands.
3. Scanning the internal keyboard
4. Scanning the internal mouse
5. Detecting characters from an external keyboard
6. Detecting external mouse input
7. Decrementing and checking internal timers
8. Sleep until next interrupt (maximumperiod one centisecond).

Several periodic functions, which are scheduled by the internal counters, include:

1. Check watchdog timeout
2. Sample temperature
3. Sample battery charge level
4. Update status display

Administration of the link between microcontroller and host is performed under interrupt control. The main line code transfers any relevant data for the host into the TS102's 'read' buffer under an asynchronous interrupt-controlled transmit mechanism.

12.1.2 Internal keyboard scanning

The microcontroller address bus signals are driven out to the internal keyboard drive lines and depressed keys are sensed via the data bus. The microcontroller implements debounce and key repeat control in software.

12.1.3 External keyboard and mouse

The microcontroller subsystem incorporates a dual universal asynchronous receiver transmitter (DUART). This handles serial transfers and from the keyboard and mouse ports on the baseboard.

Input from the internal and external mouse and internal and external keyboards is multiplexed by the microcontroller. This allows internal devices to remain active while an external mouse and/or keyboard are connected.

12.1.4 Pointing stick

The pointing stick uses strain gauge technology to detect and quantify vertical and horizontal pressure. It produces two outputs (for the *x* and *y* directions) which are amplified by a high-gain amplifier before being applied to analog-to-digital converters in the microcontroller. The dc bias of the amplifiers is software adjustable via a digital potentiometer, allowing pointing stick recalibration and sensitivity adjustment to be carried out in software.

12.1.5 Real time clock

The real time clock is used solely to provide the timed "wake up" alarm; a facility that allows the user to set (via the NCE Save Panel, see the *NCE User Guide*) a time at which the SPARCbook 3 system powers up automatically. All accesses to this real time clock are handled by the microcontroller in response to commands from the host.

The microconroller subsystem's real time clock is in addition to the MK48T08 real time clock which is fitted to the base board. The latter is used by the OpenBoot firmware as in a conventional SPARCstation.

12.1.6 EEPROM

The EEPROM is used to store system manufacturing information. The EEPROM is a serially accessed device which is read and written using its own command set. Commands are written to it along with any new data via D(0). Data is read from the device via D(1).

12.1.7 LCD status display

The status screen is managed by the microcontroller. The display can be programmed for continuous loop or static message display. A stringcode table defines characters and text strings to be used for any messages that the microcontroller generate's to indicate, for example, that a low-power shutdown is imminent.

Using a command set, the host is able to select the display contents and cause them to be displayed. Messages may be altered by sending a new definition for the status display characters to the microcontroller.

The display of microcontroller generated messages take precedence over host messages, although once the microcontroller message sequence has completed the host sequence is displayed.

12.2 Command Set

The microcontroller command set is organized into several functional groups. The command structure is the same for all groups.

12.2.1 Command synchronization

The administartion of the host-microcontroller link must be single threaded; only one data exchange may be active at any time. The sync command (0x00) can be used to bring the microntroller and host into phase. This allows the data stream from the microcontroller to be discarded until the synchronizing byte is found.

{how do I know when I have received the synchronizing byte}

12.2.2 System Information Commands

Opcode	Command	Argument	Returned	Function
0x01	Read Serial Number	none	ack + 4 bytes	The Unit Serial Number is represented by a four byte value held in EEPROM. These are returned in the order msb...lsb.
0x02	Read Ethernet Address	none	ack + 6 bytes	The Ethernet Hardware Address for the unit is represented by six byte values, held in EEPROM. These are returned in ordermsb .. lsb.
0x03	Read Hardware Version	none	ack +2 bytes	The Hardware Version is held in two EEPROM byte values. These are returned in order major .. minor.
0x04	Read Microcontroller Version	none	ack + 2 bytes	The Microcontroller Version is hardwired into the microcontroller firmware. It is described by two byte values, returned in order major .. minor.
0x05	Read Max Temperature	none	ack + 1 byte	The Maximum Recorded temperature is held in EEPROM as a byte value. This value is returned.
0x06	Read Min Temperature	none	ack + 1 byte	The Minimum Recorded temperature is held in EEPROM as a byte value. This value is returned.
0x07	Read Current Temperature	none	ack + 1 byte	The Current Temperature is read from the unit temperature sensor and returned as a byte value.
0x08	Read System Variant	none	ack + 4 bytes	System variant information is held in EEPROM as four byte values. This includes, for example, non-US keyboard and base board type. The encoding of these values is understood by both microcontroller and host system software.
0x09	Read Power-on Cycles		ack + 4 bytes	This command returns the number of times that the unit has been powered-on. This value is held in EEPROM as a four byte value which are returned in order msb...lsb. Assuming an average of ten power-on cycles per day, this value will overflow in just around eighteen years.

Table 12-1 System Information Command Summary

Opcode	Command	Argument	Returned	Function
0x0A	Read Power-on Seconds		ack + 4bytes	This command returns the total number of seconds that the unit has been powered-on during its life. This value is held in EEPROM as a four byte value which are returned in order msb ... lsb. This value will overflow in 139 years of constant operation.
0x0B	Read Reset Status		ack + 1 byte	The reset status of the microcontroller is derived from the cause of the reset. This information is updated at the time of reset, and remains until the next reset.The interpretation of this value is shared with the system software. The defined values are as follows: 0x00 Reserved 0x01 Power-on reset 0x02 Keyboard (user) reset 0x03 Watchdog reset 0x04 Timeout reset 0x05 Software reset 0x06 Brownout reset 0x07 Reserved
0x0C	Read Event Status		ack + 2 bytes	The microcontroller status is a reflection of the current state of the unit and its environment. The interpretation placed on this bitmask is shared with system software. The defined values are as follows: Bit 0 Shutdown request Bit 1 Battery event - low power warning Bit 2 Battery event - very low power warning Bit 3 Battery event - battery changed Bit 4 Change in external keyboard status Bit 5 Change in external mouse status Bit 6 Change in external VGA status Bit 7 Change in case-closed microswitch status Bit 8 Microcontroller error Bit 9 Reserved Bit 10 Change in external battery status Bit 11 Change in external battery charging state Bit 12 External battery event - low power warning Bit 13 Change in DC state Bit 14 Change in charging state Bit 15 Power-On button pressed The status is returned in order msb...lsb,with bits set to indicate that the associated event has occurred. After returning status to the host, the microcontroller normally resets its internal status flags. This can be avoided by using command 0x1B

Table 12-1 System Information Command Summary (Continued)

Opcode	Command	Argument	Returned	Function
0x0D	Read Real Time Clock		ack + 7 bytes	The microcontroller reads the values from the RTC and returns them in the order: seconds, minutes, hour, day, date monthe, year. This command will fail (return NAck) if the RTC reports an internal error.
0x0E	Read External VGA Port		ack + 1 byte (bits 3:0 valid)	The VGA port allows the unit to sense the type of monitor attached. No interpretation is placed on these bits by the microcontroller.
0x0F	Read Microntroller ROM Checksum		ack + 2 bytes	This command returns the checksum of the microcontroller's internal ROM. The checksum is calculated in the same way as the Tadpole EPROM Programmer and is returned in order msb…lsb.
0x10	Read Error Status		ack + 2 bytes	Internal microcontroller errors and failures of devices (such as the RTC) which are administered by the microcontroller, are indicated to the host by the microcontroller setting the Microcontroller Error bit in the Event Status register and sending the interrupt character (0xFA). The host uses the Read Event Status command (0x0C), to read the error status and clear the error bit. 0x00 No error 0x01 Command error 0x02 Execution error 0x04 Physical error
0x11	Read External Status		ack + 2 bytes	The host can determine the external configuration of the unit by using this command. The microcontroller returns a bitmask with bits being set to indicate the current conditionof the unit. Bit 0 Mains Power Available Bit 1 Internal Battery Attached Bit 2 External Battery Attached Bit 3 External VGA Attached Bit 4 External Keyboard Attached Bit 5 External Mouse Attached Bit 6 Microswitch closed (ie lid down) Bit 7 Internal battery currently charging Bit 8 External battery currently charging Bit 9 Internal battery currently discharging Bit 10 External battery currently discharging **NOTE**: A Sun mouse cannot be polled to confirm its presence. Bit 5 is set if packets are detected at the external mouse port, but it is not possible to reliably detect subsequent removal of the external mouse.

Table 12-1 System Information Command Summary (Continued)

Opcode	Command	Argument	Returned	Function
0x12	Read User Configuration Area and Address		ack + 2 bytes	The User Configuration Area of the EEPROM is used to store long-lived system configuration information. This area can be read and written by using the read and write EEPROM commands, provided that the location of the record within the EEPROM can be ascertained. The first byte returned is the offset of record within the EEPROM, the second is the total lengthof the record.
0x13	Read Microcontroller Voltage (MICROMONV)		ack + 1 byte	This command reads the voltage being supplied to the microcontroller.
0x14	Read Internal Battery Voltage (INTBATV)		ack + 1 byte	This command reads the voltage of the internal battery.
0x15	Read DC-IN Voltage (DCINV)		ack + 1 byte	This command reads the voltage being supplied to the system.
0x16	Read Horizontal Pointer Voltage (PTRHORV)		ack + 1 byte	This command reads the voltage on the bridge for the horizontal pointer.
0x17	Read Vertical Pointer Voltage (PTRVERV)		ack + 1 byte	This command reads the voltage on the bridge for the vertical pointer.
0x18	Read Internal Battery Charge Level (EXTCHGLVL)		ack + 1 byte	This command reads the level of the charge for the internal battery.
0x19	Read External Battery Charge Level (EXTCHGLVL)		ack + 1 byte	This command reads the level of the charge for the external battery.
0x1A	Read Real Time Clock Alarm		ack + 7 bytes	This command reads the current alarm setting for the real time clock and returns them in the order: seconds, minutes, hour, day, date month, year. Unsupported fields return 0xFF.
0x1B	Read Event Status with no Reset		ack + 2 bytes	This command returns microcontroller status without causing the microcontroller to reset its internal status flags. Refer to command 0x0C for details of the status flags.

Table 12-1 System Information Command Summary (Continued)

Opcode	Command	Argument	Returned	Function
0x1C	Read Internal Keyboard Layout		ack + 2 bytes	This command returns the keyboard type (first byte) and layout information (second byte) for the unit's internal keyboard. This value is recorded in EEPROM, but defaults to Sun Type 5 US layout if the EEPROM is corrupt.
0x1D	Read External Keyboard Infromation		ack + 2 bytes	This command returns the keyboard type (first byte) and layout information (second byte) for an external keyboard attached to the unit. If no external keyboard is attached, both fields are set to KB_UNKNOWN (0xFF).
0x1E	Read EEPROM Status		ack + 1 byte	This command forces the internal EPROM to be checksummed and returns the result in a bitfield. Bit 0 Factory area checksum fail Bit 1 Consumer area checksum fail Bit 2 User area checksum fail Bit 3 VPD area checksum fail

Table 12-1 System Information Command Summary (Continued)

12.2.3 Read/Write/Modify Commands

These commands are used to read or modify the byte attribute addressed. Each command uses the same form, where the value written to the attribute is given by:

```
new_attribute = (old_attribute AND mask) EOR value
```

The old attribute value is returned in all cases. Therefore, to read an attribute the parameters are:

```
mask = 0xff, value = 0
```

For example, to write an attribute to 0x5A, the parameters are:

```
mask = 0x00, value = 0x5A
```

To change bit n to value v (either 1 or 0) while leaving all other bits unchanged:

```
mask = NOT(1<<n), value = (v<<n)
```

The following table summarizes the Read/Write/Modify Commands

Opcode	Command	Argument	Returned	Function
0x20	Control LCD	mask	ack + 2 bytes	The host controls the LCD via this command. Setting a bit activates the approriate symbol on the unit's LCD panel. Bit 0 Caps lock Bit 1 Scroll lock Bit 2 Numlock Bit 3 Disk active Bit 4 LAN active Bit 5 WAN active Bit 6 PCMCIA active Bit 7 DC OK Bit 8 Compose
0x21	Control Bitport	mask	ack + 1 byte	The microcontroller administers several bit ports which are used to control things within the system. Setting a bit in the bitport sets the corresponding signal to a high level. The power-up signal setting is given in parentheses. Bit 1 TFTPWR - S3XP TFT power (high) Bit 2 SYNCINVA - external monitor sync (low) Bit 3 SYNCINVB - external monitor sync (low) Bit 4 BP_DIS - bootprom disable via TS102 - boot from PCMCIA. (low) Bit 5 ENCSYNC - enable composite sync (low)

Table 12-2 Read/Write/Modify Commands

Opcode	Command	Argument	Returned	Function
0x23	Speaker Volume	mask	ack + 1 byte	The volume is controlled via a digital potentiometer which is varied with this command. Values in the range 0-255 (dec) are acceptable when setting this parameter. The value is stored in EEPROM and is used to determine the power-up value of this variable.
0x24	Control TFT Brighttness	mask	ack + 1 byte	The brightness of the built-in TFT display is controlled via a digital potentiometer which is varied with this command. Values in the range 0-255 (dec) are acceptable when setting this parameter. The value is stored in EEPROM and is used to determine the power-up value of this variable.
0x25	Control Watchdog	mask	ack + 1 byte	The microcontroller implements a watchdog timer so as to detect host failure which can be usually due to hardware or software problems. If the host fails to retrigger the watchdog timer before the timer expires, then the microcontroller resets the system. The value returned by Read Reset Status will be watchdog reset. The watchdog period is expressed in seconds from 1-255 (dec). A setting of, zero (0) indicates that the watchdog is to be suspended. This is the default state after power up.
0x26	Control Factory EEPROM Area	mask	ack + 1 byte	The factory and vital product data areas of the EEPROM are write-protected after power up. This command is used to remove the write-protection so that changes can be made to values held in these areas. The parameter must be set to 'w' (0x77).
0x28	Control KBD Time Until Repeat	mask	ack + 1 byte	Time til autorepeat
0x29	Control KBD Time Between Repeats	mask	ack + 1 byte	Time between repeats
0x2A	Control Timezone	mask	ack + 1 byte	This command is used to determine the adjustment required to derive local time from the internal GMT. This is expressed as a signed value in 30 minute increments. This variable is slaved in EEPROM.
0x2B	Control Mark Space Ratio	mask	ack + 1 byte	This command is used to determine the mark space ratio used when flashing the led (which LED – Battery?)

Table 12-2 Read/Write/Modify Commands (Continued)

Opcode	Command	Argument	Returned	Function
0x2E	Control Diagnostic Mode	mask	ack + ((mark_space_ratio AND mask) EOR value)	This command is used to control various features used during diagnostic test of the system. By setting the appropriate bit in this simulated "port", a diagnostic feature is activated. The following bit values are currently defined: 0x1 Display command interface diagnostics on LCD. 0x2 Set keyboard and mouse ports to 9600 baud.
0x2F	Control Screen Contrast	mask	ack + 1 byte	This command, which is only valid on the LC system variant, is used to vary the LCD screen contrast value between 0 and 0xFF

Table 12-2 Read/Write/Modify Commands (Continued)

12.2.4 Commands Returning no Status

The following commands are provided to simplify the use of the micromcontroller by system software, particularly form interrupt routines. No status is returned for these commands and in many cases the commands do not require arguments

OpCode	Command	Argument	Returned	Function
0x30	Ring Bell	duration_msb duration_lsb	ack	The argument to this command expresses the duration in centiseconds for which the bell is sounded. A duration of zero indicates that any current bell should be terminated.
0x32	Set Diagnostic Status	msb lsb	ack	This command displays a three-digit hex value on the unit's status display. The following special sequences are defined: 0xFF Clear Status LCD 0xAxx Display "Cxx" Values between 0x000 and 0xFFF only are accepted.
0x33	Clear Key Combination Table	none	ack	This command is used by system software clear the key combination table prior to defining new character combinations.
0x34	Perform Software Reset	none	ack	This command will cause the microcontroller to reset the host CPU. The value returned by Read Reset Status after the unit is reset will be software reset.
0x35	Set Real Time Clock	7 bytes (sec, min, hr, day, date, mon, yr)	ack	This command is used by system software to initialize the RTC. By convention the RTC represents GMT, although it is a system software function to maintain this.
0x36	Recalibrate Pointing Stick	none	ack	This command is used by system software to request the microcontroller to rebalance the pointing stick by adjusting the digital potentiometers
0x37	Set Bell Frequency	frequency_msb frequency_lsb	ack	This command is used by system software to define the frequency to be used when the bell is sounded.
0x39	Set Internal Battery Charge Rate	charge_level	ack	This command is used by system software to set the level at which the internal battery is to be charged.
0x3A	Set External Battery Charge Rate	charge_level	ack	This command is used by system software to set the level at which the external battery is to be charged.
0x3B	Set Real Time Clock Alarm	7 bytes (sec, min, hr, day, date, mon, yr)	ack	This command is used by system software to set the RTC alarm. By convention the RTC represents GMT, although it is a system software function to maintain this.

Table 12-3 Commands Returning no Status

12.2.5 Block Transfer Commands

This gorup of commands handle transfers of arbitrary amounts of data, up to a maximum of 255 bytes, between the CPU and microcontroller. A length parameter of zero transfers no data. Attenpts to transfer excess data results in an error.

Opcode	Command	Argument	Returned	Function
0x40	Read EEPROM	length offset	ack + <data>	This command reads arbitrary amounts of data, up to a maximum of 255 bytes, from the EEPROM. A length parameter of 0 (zero) transfers no data. Attempts to transfer excess data returns an error.
0x41	Write EEPROM	length offset <data>	ack	This command writes an area of EEPROM. If the area to be written includes part of the factory area, and this is write-protected, then the status byte will indicate the error condition.
0x42	Write to Status Display	length offset <data>	ack	This command writes a string to the status display. The offset argument is used to determine the position of the string on the LCD display, which is made up of two lines, each of 16 characters. The top line of the display is addressed with offsets 0x0 to 0xE, the bottom line with offsets 0x40 to 0x4E
0x43	Define Special Character	length offset <data>	ack	This command writes a string to the microcontroller which is interpreted as a definition of a special character to be displayed on the status screen. 8 bytes of data are required per character, and the following offsets may be used to define special status symbols: 0 WAN1 active 1 WAN2 active 2 LAN1 active 3 LAN2 active 4 PCMCIA active 5 DC good 6 Backslash (used for disk spinner)

Table 12-4 Block Transfer Commands

12.2.6 Generic Commands

This group of commands perform related functions and are of variable length.

Opcode	Command	Argument	Returned	Function
0x50	Define Key Combination Entry	(sequence and combination_length)	ack	The built-in keyboard features an 86-key pad. Normal Sun 4 type 5 keyboards offer up to 106 keys, including a separate numeric keypad, some of which must be simulated with key-combinations. In addition, extra 'hot-keys' which are used to control the behavior of the unit. These simulated keys aredefined in terms of microcontroller action (e.g. reset). The microcontroller actions which can be invoke by key-sequence are: 0x90 System Reset 0x91 System Powerdown 0x92 TFT Brightness UP 0x93 TFT Brightness DOWN 0x94 LCD Contrast UP 0x95 LCD Contrast DOWN 0x96 Speaker Volume UP 0x97 Speaker Volume DOWN 0x98 Display Next Status It should be noted that scan codes in this range and above are invalid. Complete key-sequences are sent to the host in the usual way. Key sequences are given in terms of scancode, the number returned by a Sun 4 type5 keyboard when that key is pressed. The microcontroller interprets key combinations sequentially, i.e. the first key must be down before the second key is pressed.
0x51	Define String Table Entry	stringcode length <string>	ack	To support language variants without the need to change the microcontroller, the messages displayed on the status screen are defined by the system software and downloaded to the microcontroller. These messages are discriminated by stringcode. Additional messages, not usually displayed by the microcontroller, may also be defined by this method.

Table 12-5 Generic Commands

Opcode	Command	Argument	Returned	Function
0x52	Define Status Screen Display	length <message bytes and puase descriptor>	ack	The LCD status screen can be set up to display a continuous loop of messages separated by definable pauses, or to display a single message. All messages must be predefined entries in the string table, identified by their index number. Message descriptors are discriminated from pauses by the lsb, 0x1 indicating a pause. Pauses are defined in odd multiples of 0.1 seconds, thus 0.1, 0.3, 0.5.. . Two special characters, loop (0xFF) and stop (0xFD) are defined. The status screen retains the last message displayed.

Table 12-5 Generic Commands (Continued)

12.2.7 Generic Commands with Optional Status

Opcode	Command	Argument	Returned	Function
0x64	Perform EMU Command	<command>	ack	The host controls the Energy Management Unit by sending commands to it through this interface. The microcontroller does not interpret these commands.
0x65	Read EMU Register	register	ack +1 byte	The host may read the Energy Management Unit registers with this command.
0x66	Write EMU Register	register value	ack	The host may write to an the Energy Management Unit register with this command.
0x67	Read EMU's RAM	ram_address	ack + 1 byte	The host may read from the Energy Management Unit's RAM with this command.
0x68	Write EMU's RAM	ram_address value	ack	The host may write to the Energy Management Unit's RAM with this command.
0x69	Read BQ Register	register	ack + 1 byte	The host may read registers in the external battery's EMU with this command.
0x6A	Write BQ Register	register value	ack	The host may write to registers in the external battery's EMU with this command.

Table 12-6 Generic Commands with Optional Status

12.2.8 Administration Commands

Opcode	Command	Arguments	Returned	Function
0x70	Set User Password	length <password>	ack	The supplied user password is encrypted and stored in EEPROM. The microcontroller fails attempts to use this command if a password not been supplied when required.
0x71	Verify User Password	length <password>	ack + status	The unencrypted user password is encrypted by the microcontroller and verified against the version held i n EEPROM. The status returned indicates whether the supplied password does encrypt to the same value as the stored version. A status value of zero indicates a correct comparison.
0x72	Get System Password Key	none	ack + <7 byte key>	The microcontroller will return the value from the RTC. It will also retain the values which are returned for use in the Verify System Password command.
0x73	Verify System Password	length <password>	ack + status	The microcontroller derives a system password, using the unit serial number and system password keys. This value is verified against the value passed in. Where a successful comparison is achieved the current RTC value is retained by the microcontroller. The indication that the system password is set expires 24 hours from the current RTC value.
0x82	Power Off	none	none	This command causes the microcontroller to power down the system.
0x83	Power Restart	period_msb period_xx period_lsb	none	The microcontroller configure the unit to cause an automatic power-on after the specified number of seconds has elapsed.

Table 12-7 Adminstration Commands

Further Information

A

This appendix provides a list of publications or websites to which you can refer for further information about components used in the SPARCbook 3.

Device	Facility Provided	Contact
STSX1012 microSPARC II	CPU, SBus controller, memory controller	http://www.sun.com/sparc
TurboSPARC	CPU, SBus controller, memory controller (S3TX only)	TurboSPARC Microprocessor User's Guide Revision 1.0, October 1996 Fujitsu Microelectronic Inc., 3545 North First Street, San José CA 95134-1804 http://www.fujitsumicro.com
NCR89C100 MACIO	Ethernet, SCSI, Parallel	NCR SBUS I/O Chipset Data Manual March 1993 NCR Corporation, Dayton, Ohio http://www.symbios.com/
NCR89C105 SLAVIO	Serial I/O, External keyboard/mouse interface, interrupt controller, EBus controller.	
MK48T08	Real time clock /SRAM	Memory Products Databook, 3rd Edition, June 1994, SGS-Thomson Electronics http://www.st.com
Weitek P9100	Graphics Controller	Power 9100 Programmer's Manual Weitek http://www.weitek.com
IBM RGB528 Palette DAC	Display controller, color palette RAM	RGB528A Product Desciption http://www.chips.ibm.com/
AT&T T725FC	ISDN controller	Lucent Technologies, Inc. http://www.lucent.com/micro
CS4215	Audio CODEC	Audio Databook Crystal Corporation PO Box 17847, Austin, TX 788760
R65C39 MCU	Microcontroller Unit, provides control of the modem	http://www.pb.rockwell.com
RC144DPL	Modem data pump	

Connector Information

B

This appendix provides a pinouts for the connectors used the SPARCbook 3.

B.1 I/O Panel Connectors

This chapter provides details of the connector-pin signal assignments for the interfaces on the I/O panel.

SPARCbook 3 systems are supplied with cable adapters, where required, so that interfaces appear on the appropriate industry standard connector.

Note

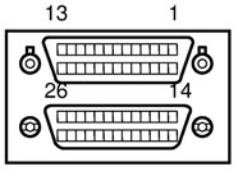
Connections between the SPARCbook 3 and any peripherals must be made using shielded cables to maintain compliance with FCC radio frequency emission limits. All adapter cables supplied with your SPARCbook 3 are shielded types. Refer to page iii of this User Guide if interference is suspected.

B.1.1 DCIn

Connector	Pin	Signal
	1	+12 V
	2	0 V

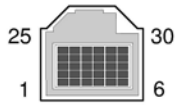
B.1.2 Parallel (S3XP, S3GX and S3TX)

The following connector is fitted on the I/O panel of the S3XP, S3GX and S3TX. It is the he upper connector of a double-stacking pair.

Connector	Pin	Signal	Pin	Signal
	1	/STROBE	14	/AUTOFEED
	2	DATA (0)	15	/ERROR
	3	DATA(1)	16	/INIT
	4	DATA(2)	17	/SELECT_IN
	5	DATA(3)	18	Signal Ground
	6	DATA(4)	19	Signal Ground
	7	DATA(5)	20	Signal Ground
	8	DATA(6)	21	Signal Ground
	9	DATA(7)	22	Signal Ground
	10	/ACK	23	Signal Ground
	11	/BUSY	24	Signal Ground
	12	PE	25	Signal Ground
	13	SLCT	26	Signal Ground

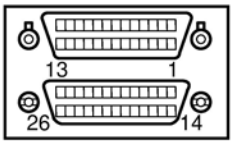
B.1.3 Parallel (S3 and S3LC)

The following connector is fitted on the I/O panel of the S3 and S3LC models.

Connector	Pin	Signal	Pin	Signal
	1	Signal Ground	16	SELECT_IN
	2	/STROBE	17	Signal Ground
	3	Signal Ground	18	DATA(0)
	4	DATA(1)	19	DATA(2)
	5	+5 V	20	Signal Ground
	6	Signal Ground	21	DATA(4)
	7	DATA(3)	22	DATA(6)
	8	DATA(5)	23	Signal Ground
	9	Signal Ground	24	ACK
	10	DATA(7)	25	Signal Ground
	11	Signal Ground	26	PE
	12	BUSY	27	Signal Ground
	13	SELECT	28	AUTOFEED
	14	Signal Ground	29	INIT
	15	ERROR	30	Signal Ground

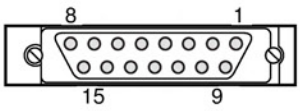
B.1.4 Ethernet (S3XP, S3GX and S3TX)

The following connector is fitted on the I/O panel of the S3XP, S3GX and S3TX. It is the lower connector of a double-stacking pair.

Connector	Pin	Signal	Pin	Signal
	1	TRANSMIT DATA–	14	TRANSMIT DATA+
	2	RECEIVE DATA+	15	RECEIVE DATA–
	3	COLLISION–	16	COLLISION+
	4	+12V	17	Signal Ground
	5	Not connected	18	Not connected
	6	Not connected	19	Not connected
	7	Not connected	20	Not connected
	8	Not connected	21	Not connected
	9	Not connected	22	Not connected
	10	+5 V	23	Signal Ground
	11	+5V	24	Signal Ground
	12	Not connected	25	Not connected
	13	+5V	26	Signal Ground

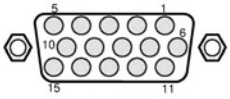
B.1.5 Ethernet (S3 and S3LC)

The following connector is fitted on the I/O panel of the S3 and S3LC models. It is also present on a cable adapter supplied with the S3XP, S3GX and S3TX models.

Connector	Pin	Signal	Pin	Signal
	1	Chassis Ground	9	COLLISION–
	2	COLLISION+	10	TRANSMIT DATA–
	3	TRANSMIT DATA+	11	Chassis Ground
	4	Chassis Ground	12	RECEIVE DATA–
	5	RECEIVE DATA+	13	+12V[a]
	6	Chassis Ground	14	Chassis Ground
	7	Not Connected	15	Not Connected
	8	Chassis Ground		

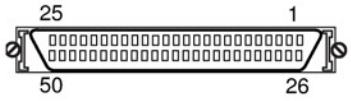
a.Protected internally by a self-resetting fuse

B.1.6 Video

Connector	Pin	Signal	Pin	Signal
	1	RED	9	Not Connected
	2	GREEN	10	Signal Ground
	3	BLUE	11	DISPLAYID(0)
	4	DISPLAYID(2)	12	DISPLAYID(1)
	5	Signal Ground	13	HSYNC
	6	Signal Ground	14	VSYNC
	7	Signal Ground	15	DISPLAYID(3
	8	Signal Ground		

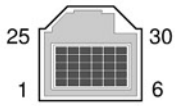
B.1.7 SCSI (S3XP, S3GX and S3TX)

The following connector is fitted on the I/O panel of S3 XP, S3GX and S3TX and is present on the a cable adapter supplied with S3 and S3LC

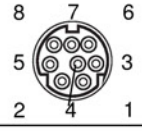
Connector	Pin	Signal	Pin	Signal
	1	Signal Ground	26	SCSI D (0)
	2	Signal Ground	27	SCSI D (1)
	3	Signal Ground	28	SCSI D (2)
	4	Signal Ground	29	SCSI D (3)
	5	Signal Ground	30	SCSI D (4)
	6	Signal Ground	31	SCSI D (5)
	7	Signal Ground	32	SCSI D (6)
	8	Signal Ground	33	SCSI D (7)
	9	Signal Ground	34	SCSI D(PARITY)
	10	Signal Ground	35	Signal Ground
	11	Signal Ground	36	Signal Ground
	12	Not Used	37	Not Used
	13	Not Used	38	SCSI TERMPWR
	14	Not Used	39	Not Used
	15	Signal Ground	40	Signal Ground
	16	Signal Ground	41	/SCSI ATN
	17	Signal Ground	42	Signal Ground
	18	Signal Ground	43	/SCSI BSY
	19	Signal Ground	44	/SCSI ACK
	20	Signal Ground	45	/SCSI RST
	21	Signal Ground	46	/SCSI MSG
	22	Signal Ground	47	/SCSI SEL
	23	Signal Ground	48	/SCSI CD
	24	Signal Ground	49	/SCSI SREQ
	25	Signal Ground	50	/SCSI IO

B.1.8 SCSI (S3 and S3LC)

The following connector is fitted to the I/O panel of S3 and S3LC models.

Connector	Pin	Signal	Pin	Signal
	1	Not connected	16	SCSI D(6)
	2	SCSI D(0)	17	Signal Ground
	3	Signal Ground	18	SCSI D(7
	4	SCSI D(1)	19	SCSI D(PARITY)
	5	SCSITERMPWR	20	Signal Ground
	6	SCSI D(2)	21	/SCSI REQ
	7	SCSI D(3)	22	Signal Ground
	8	Signal Ground	23	/SCSI BSY
	9	/SCSI ACK	24	Signal Ground
	10	Signal Ground	25	/SCSI ATN
	11	SCSI D(4)	26	/SCSI CD
	12	Signal Ground	27	/SCSI RST
	13	Signal Ground	28	/SCSI MSG
	14	SCSI D(5)	29	/SCSI SEL
	15	Signal Ground	30	/SCSI ID

B.1.9 Keyboard/Mouse

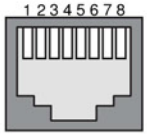
Connector	Pin	Signal	Pin	Signal
	1	0V	5	Keyboard Tx
	2	0V	6	Keyboard Rx
	3	5V	7	Mouse Tx
	4	Mouse Rx	8	5V

B.1.10 Serial (x2)

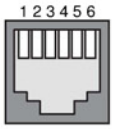
Connector	Pin	Signal	Pin	Signal
	1	DTR	5	RECEIVE DATA
	2	CTS	6	RTS
	3	TRANSMIT DATA	7	DCD
	4	Chassis Ground	8	TX CLOCK

B.1.11 ISDN

The following connector is not present on the S3LC.

Connector	Pin	Signal	Pin	Signal
	1	Not Connected	5	TE IN–
	2	Not Connected	6	TE OUT–
	3	TE OUT+	7	Not Connected
	4	TE IN+	8	Not Connected

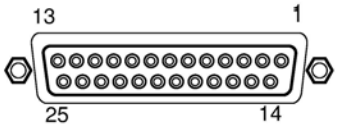
Modem

Connector	Pin	Signal	Pin	Signal
	1	Not Connected	4	TIP
	2	Not Connected	5	Not Connected
	3	RING	6	Not Connected

B.2 Cable Adapter Connectors

B.2.1 Parallel Cable Adapter

The following connector is fitted on the parallel cable adapter supplied with all SPARCbook 3 models.

Connector	Pin	Signal	Pin	Signal
	1	/STROBE	14	/AUTOFEED
	2	DATA (0)	15	/ERROR
	3	DATA (l)	16	/INIT
	4	DATA (2)	17	/SELECT_IN
	5	DATA (3)	18	Signal Ground
	6	DATA (4)	19	Signal Ground
	7	DATA (5)	20	Signal Ground
	8	DATA (6)	21	Signal Ground
	9	DATA (7)	22	Signal Ground
	10	/ACK	23	Signal Ground
	11	/BUSY	24	Signal Ground
	12	PE	25	Signal Ground
	13	/SELECT		

B.3 Removable Hard Drive SCSI Connector

Signal	Pin	Pin	Signal
MOTORVCC	1	21	MOTORVCC
MOTORGND	2	22	MOTORGND
/SCSIIO	3	23	GND
/SCSICD	4	24	SCSID(7)
/SCSISEL	5	25	GND
/SCSIREQ	6	26	SCSID(6)
/SCSIRST	7	27	GND
/SCSIMSG	8	28	SCSID(5)
/SCSIIACK	9	29	GND
/SCSIATN	10	30	SCSID(4)
GND	11	31	GND
/SCSIBSY	12	32	SCSID(3)
GND	13	33	GND
/SCSITPWR	14	34	SCSID(2)
/SCSID(P)	15	35	GND
NOT CONNECTED	16	36	SCSID(1)
NOT CONNECTED	17	37	GND
/MOTOROFF	18	38	SCSID(0)
GND	19	39	GND
MOTORVCC	20	40	MOTORVCC

Index

L

M

N

P

R

S

T

V

W