PCI:SBus Comparison

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A

Part No.: 805-4447-10
October 1998, Revision A

Send comments about this document to:
docfeedback@sun.com

Contents

Tables

Preface

The PCI:SBus Comparison document describes feature differences between the PCI interconnect bus and the SBus interconnect bus. This document is intended to help developers who are in transition from developing for SBus to developing for PCI bus. In addition, this document is intended to help field engineers who are assisting these developers.

Related Documents

The following documentation contains the topics related to the information discussed in this *PCI:SBus Comparison* manual.

TABLE P-1 Related Documents

Subject	Title	Part Number
PCI	PCI Local Bus Specification, Revision 2.1	802-2387-02
	PCI System Architecture, by Tom Shanley and Don Anderson, MindShare, Inc.	
	PCI--A Developer Overview, December, 1996 (technical brief)	
	Sun Microsystems™ PCI migration Web page: `http://shasta.corp.sun.com/Rte66/`	
	PCI Developer Kit	
	IEEE 1275-1994 Standard for Boot Firmware	
	PCI Bus Binding to IEEE Std1275-1994 Standard for Boot (Initialization Configuration) Firmware, Rev 2..0 at:`fhttp://playground.sun.com/pub/1275/bindings/pci`	

TABLE P-1 Related Documents

Subject	Title	Part Number
PCI and SBus	*SBus and PCI Bus: a Comparison* (white paper)	
SBus	*1496-1993 IEEE Standard for a Chip Module Interconnect Bus: SBus*, IEEE Computer Society	SH16659-NYF
	1275.2-1994 IEEE Standard for Boot (Initialization Configuration) Firmware:Bus Supplement for IEEE 1496	SH94236-NYF
	SBus Specification B.0, by Edward H. Frank and Jim Lyle, Sun Microsystems, Inc.,December, 1990	800-5922-10
	SBus:Information, Applications, and Experience, James D. Lyle, Springer-Verlag	
	SBus Handbook, by Susan A. Mason, SunSoft Press / Prentice-Hall	

Sun Documentation on the Web

The `docs.sun.com` web site enables you to access Sun technical documentation on the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

`http://docs.sun.com`

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

`docfeedback@sun.com`

Please include the part number of your document in the subject line of your email.

Feature Comparison

This document contains information and tables to show the comparison of the major PCI features to those of SBus, as well as, some PCI *only* features.

Features Tables

Tables 1 through 5 compare the SBus and PCI electrical, firmware, hardware, software, and normal transaction cycle features.

TABLE 1 Electrical Features Comparison

Feature	SBus	PCI	Difference
Power consumption	P=VI	25W (5V x 5A)	None
Power supply	5V +/-.25V 2A max per connector 12V +/-.75V 30mA max per connector	5V +/-5% 5A max per connector 3.3V +/-.3V & .6A max per connector 12V +/-5% 500ma per connector -12V +/-10% 100ma per connector	None

TABLE 2 Firmware Features Comparison

Feature	SBus	PCI	Difference
FCode required	All devices	Boot and console devices	None

TABLE 3 Hardware Features Comparison

Feature	SBus	PCI	Difference
Address allocation	Static mapping. Slot has known start address and length	Dynamic mapping for I/O and memory address spaces. Slot distinguished only by address of configuration space header.	None
Addressing mode	Use virtual addressing. Require MMU in SBus controller and address translation. 32-bit virtual address for masters, 28-bit physical address space per slave.	Normal address mode	None
Address spaces	Standard chunk of memory-mapped space (28 bits/slot)	Three physical address spaces: memory, I/O, configuration.	None
Auto-configuration			
Burst: Transfer mode size	Available. Up to 64 bytes: size declared in advance in SIZ{2:0}.	Yes Variable size, no limit; determined by PCI device and PCI bridge.	None
Bus bandwidth: (data transfer rate)	25MHz/32-bit, max. 100MBytes/sec for 32-bit. 200 MBytes/sec for 64-bit	33MHz (rev 2.0) or 66 MHz for PCI/66, 32/64bit, maximum rate from 132MByte/second (33MHz/32-bit) to 528MByte/second (66 MHz/64-bit).	None
Bus parking	No	Yes. An arbiter may grant the busses to a master when the bus is idle and masters are not generating requests for the bus. If the master that the bus is parked on subsequently issues a request from the bus, it can immediately access it.	None
Bus width			x 32-to 64-bit
Clock frequency/ bus speed	Max CLK 25MHz CLK to out: 22nS Input Set-up: 15nS	33MHz max CLK CLK to out:22ns Input Set-up: 7 ns	Synchronous; signals referenced to rising clock edge
Connectors	One type of connector	Either 5V or 3.3V 32-and 64-bit connectors; the 64-bit being an extension of the 32-bit	None

TABLE 3 Hardware Features Comparison *(Continued)*

Feature	SBus	PCI	Difference
Data path	32-bit, except 64-bit for extended transfers	32/64-bit	None
Dynamic Bus Sizing	Yes. Enables slave to control data width it accepts during non-burst transfers	No, but same function is performed by Byte Enable setting during Data Phase	None
Form factors	Single form factor per slot (double and triple-wide can share slots)	Three form factors: short, long, and variable-height short	None
Interrupts	Seven levels	Four levels (INTA-D)	None
Max devices per bus	Usually limited by electrical loading	32 slots	None
Max functions per device	1	8	None
Max masters per bus	8	32	None
Pin count (connector)	96	94 pins for both 32-bit and 64-bit connectors	None

TABLE 4 Software Features Comparison

Feature	SBus	PCI	Difference
Parity	Optional on data and virtual address transfers if parity generation and checking is implemented on controller and installed masters and slaves.	Default during address and data phases; must be performed by all PCI-compliant devices.	None

TABLE 5 Normal Transaction Cycle

Address decode	Slave selected by decode from controller	Each target performs full 32-bit decoding and drives DEVSEL# if selected
Burst size	32 words	No limit
Bus driving and turnaround	None	A turnaround cycle is necessary for signals driven by more than one agent, to enable contention-avoidance when bus driving agents transfer a signal.
Byte ordering and placement	Byte-lane swapping	DWORD swapping puts bytes in correct lane base on byte address

TABLE 5 Normal Transaction Cycle

Byte ordering	Big-endian	Little-endian
Cycle participants	Controller, Master, Slave	PCI Agents: Master/Initiator, Target
Cycle terminology and composition	Transfer = 1. Arbitration Phase 2. Translation Phase 3. Extended Transfer Information Phase 4. Transfer Phase (also called Slave Cycle)	Transfer = 1 or more clock cycles Phase = 1 or more read or write transfers, including (hidden) arbitration while current initiator is performing a data transfer Read or write transaction = 1 address phase + 1 or more data phases

Bus Transaction Participants

The bus transaction participants include the SBus controller, PCI arbiter, SBus and PCI masters, and the SBus and PCI targets.

SBus Controller

The SBus controller arbitrates contention between bus masters during the arbitration phase.

The SBus controller also performs the following:

- Address strobe
- Bus arbitration
- Bus time-outs
- Data transfer count
- SBus system clock
- Slave selects
- Virtual address translation and page size restrictions

PCI Arbiter

The PCI arbiter has no true controller but performs functions equivalent to the controller to arbitrate between bus masters. Bus masters can terminate transactions on completion or time-out; targets can also terminate transactions. The arbiter is typically integrated into the host/PCI or the PCI/expansion bus bridge chip.

SBus Master

The SBus master controls operations that produce error-free data read and write tasks between itself and an SBus slave.

The master-initiated transaction types are:

- Arbitration
- Translation
- Default transfer
- Extended transfer

PCI Master

The PCI master becomes an initiator when it has arbitrated for and gained access to the PCI bus. The initiator starts transfers but can also abort, terminate, and time out.

The master also does the following:

- Starts the Address Phase
- Inserts wait states during data transfer
- Terminates transactions

SBus Slave/Target

The SBus slave/target performs as follows:

- Monitors the SBus to determine if a master is requesting a data transfer
- Provides the data requested to the SBus master during the transfer phase
- Does not perform address decoding; the master does this
- Participates in burst transfers, dynamic bus-sizing, extended transfer phases
- Controls the data transfer rate by controlling the data acknowledgment rate
- Terminates transactions

PCI Target

The PCI target performs the following functions:

- Determines that it is the target of a transaction
- Receives a data object from the initiator
- Decodes addresses
- Participates in special cycles
- Inserts wait states during data transfer (controls data transfer rate)
- Terminates transactions

Protocol

Protocol contains the SBus and PCI basic transaction cycles for each bus, and bus arbitration.

SBus Basic Cycles

The following are definitions of the SBus basic transaction cycle for each bus.

Arbitration Phase—During this phase, masters request bus access. When there is contention between masters, the controller determines which master performs the next transfer. After arbitration, the controller is responsible for monitoring the transfer.

Translation Phase—The master and controller participate in conversion of the virtual addresses to physical address and selection signals that are used by the master and the slaves.

Extended Transfer Information Phase—This phase is used only for the cycles requested by the master that have a SIZ[2:0] value of Extended Transfer.

Transfer Phase—During the Transfer Phase (slave cycle), data is moved to or from the slave.

Dynamic Bus-sizing—This feature enables a master to communicate more easily with slaves of varying widths.

Burst Transfers—SBus burst transfer protocol is the same as that for SBus basic transactions, except that multiple words are transferred. Neither dynamic bus-sizing or varying-width slaves are allowed.

Extended Transfer Mode—These 64-bit transfers enable increased performance. Up to twice the bandwidth is possible, as the data path is twice as wide.

PCI Basic Cycles

The following are definitions of the PCI basic transaction cycle for each bus.

Address Phase—Every PCI transaction begins with this phase, which includes concurrent hidden arbitration.

The initiator identifies the target device and transaction type.

Data Phase—At the end of the address phase, the address/data bus transfers data in one or more data phases. The clock immediately following the Address Phase begins the data phase.

During the data phase, a data object is transferred between initiator and target during every rising edge of the PCI bus clock.

Note – Bus Idle State – When the last data transfer has finished, the initiator returns the bus to the idle state, which is the condition of having no transaction in progress on the bus.

Bus Arbitration

Bus arbitration by the SBus enables concurrent arbitration and transfer execution. The PCI arbitration is access-based, not time-slot-based, to minimize access latency.

SBus

When more than one SBus master requests bus access, the controller grants access to one of the requesters. Enabling arbitration concurrent while the master makes a data transfer referred to as hidden arbitration.

PCI

In PCI arbitration, a bus master must arbitrate for each bus access.

The PCI specification does not define the PCI bus arbitration scheme. The arbiter may use any scheme, but the 2.1 specification requires that the arbiter implement a fairness algorithm to avoid deadlocks.

Latency

SBus and PCI bus latency characteristics are similar. Both are low-latency, high-throughput buses. The number of wait states that targets and masters can add to a transaction is limited. Also, masters have programmable timers that limit their times on the bus during heavy-traffic periods. The limits, plus bus arbitration order, ensure that bus acquisition latencies can accurately be predicted for any bus master.

PCI and SBus latencies are functions of:

- The number of bus masters
- The arbitration method and its overhead time
- The length of an SBus translation phase or a PCI data phase
- The time the slave or target takes to finish the transfer
- The occurrence of retries and errors.

Bus access latency is the elapsed time from the moment that a bus master requests bus access until it finishes the first data transfer of the transaction. Table 6 describes the bus access latency components.

TABLE 6 PCI Bus Access Latency Components

Component	Description
Bus access latency	The elapsed time from the moment a master requests bus access until it finishes the transactions's first data transfer
Arbitration latency	The time that the master issues a request to the time when the arbiter asserts the master's grant
Bus acquisition latency	The elapsed time that the requesting master receives the grant until the current master surrenders the bus
Target latency	The elapsed time that transaction starts until the currently-addressed target is ready to finish the transactions's first data transfer

PCI Bridge Information

PCI Bridge Information

PCI bridge device PCI characteristics information is contained in this section. See Table 7 for bus type A and bus type B specific breakdowns

Improving Performance with Cache Line Size

The cacheline size for Sun™ SPARC™ platforms is 64 bytes. PCI devices for SPARC platforms should use 64 bytes for best performance.

PCI-PCI Bridge Device Supported

For PCI-PCI bridge device support, refer to “PCI Developer’s Frequently-Asked Questions” at `http://shasta.corp.sun.com/Rte66/`.

TABLE 7 Sun Host/PCI Bridge Characteristics

Characteristics	Bus A	Bus B
Data Transfer Width	64-bit	64-bit
Clock Frequency	33/66 MHz capable	33 MHz capable
Burst Size	64 bytes	64 bytes
Number of Read Buffers	One 64-byte for DMA	One 64-byte for DMA
Number of Write Buffers	Two 64-byte for DMA One 64-byte for PIO	Two 64-byte for DM One 64-byte for PIO
Dual Address Cycles	Bypass DMA only	Bypass DMA only
Fast back-to-back device	In target mode only	In target mode only
Byte Swapping	Yes for DMA	Yes for DMA
Interrupt Latency	6 Cycles inside the IDU	6 Cycles inside the IDU
Cache Line Size	64 bytes	64 bytes
Number of cache lines	16	16
Disconnected on Cache Line	Target mode (DMA) only	Target mode (DMA) only
Configuration Mechanism (per Section 3.7.4 of PCI 2.1 Specification)	Configuration Mechanism #2	Configuration Mechanism #2
Configuration Space	256 byte; starting at physical address 1FE.0101.0000	256 bytes; starting at physical address 1FE.0100.0000
I/O Space	8K; at physical address 1FE.0200.0000	8K; at physical address 1FE.0201.0000
Memory Space	2G; at physical address 1FF.0000.0000	2G; at physical address 1FF.8000.0000
Configuration Cycles	Master mode only	Master mode only
Special Cycle	Master mode only	Master mode only
Arbitrary byte enables	Consistent DMA only	Consistent DMA only
Peer-to-peer DMA	On a single segment	On a single segment
Interrupt	Four interrupt lines shared among PCI devices	Four interrupt lines shared among PCI devices

TABLE 7 Sun Host/PCI Bridge Characteristics *(Continued)*

Characteristics	Bus A	Bus B
IOMMU page size	8K and 64K.; only 8K page size is used in the STC.	8K and 64K.; only 8K page size is used in the STC.
DVMA addressing space (set by pci nexus driver)	64 Mbyte in Solaris 2.5.1; may be changed in later release of Solaris	64 Mbyte in Solaris 2.5.1; may be changed in later Solaris releases
PIO read size	1, 2, 4, 8, 16, 64 bytes for memory cycles 1,2, and 4 bytes for I/O or configuration cycles	1, 2, 4, 8, 16, 64bytes for memory cycles 1,2, and 4 bytes for I/O or configuration cycles
PIO write size	0-16 arbitrary byte enable and 64-byte aligned for memory cycles 0-4 arbitrary byte enables for I/O or configuration cycles	0-16 arbitrary byte enable and 64-byte aligned for memory cycles 0-4 arbitrary byte enables for I/O or configuration cycles
Cache-line wrap addressing mode	Not supported	Not supported
Local (on-PCI) cache	Not supported	Not supported
Exclusive access to main memory	LOCK# signal not connected	LOCK# signal not connected
Address/data stepping	Not supported	Not supported
DOS compatibility hole	Not supported	Not supported
External arbiter	Not supported	Not supported
Subtractive decode	Not supported	Not supported

Index

S

T

V