PCI:SBus Comparison

THE NETWORK IS THE COMPUTER™

Sun Microsystems Computer Company
A Sun Microsystems, Inc. Business
2550 Garcia Avenue
Mountain View, CA 94043 USA
415 960-1300 fax 415 969-9131

Part No.: 802-7681-10
Revision A, April 1997

Contents

Tables

Preface

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

How This Book Is Organized

Chapter 1, "Feature Comparison", compares PCI features with SBus features.

Related Documents

The following documents are referred to in the text or contain topics related to the information discussed in *PCI:SBus Comparison*.

Subject	Title	Part Number
PCI	*PCI Local Bus Specification, Revision 2.1*	802-2387-02
PCI	*PCI System Architecture*, by Tom Shanley and Don Anderson, MindShare, Inc.	n/a
PCI	*PCI--A Developer Overview*, December, 1996 (technical brief)	n/a
PCI	Sun Microsystems™ PCI migration web page: `http://shasta.corp.sun.com/Rte66/`	n/a
PCI	PCI Developer Kit	
PCI	IEEE 1275-1994 Standard for Boot Firmware	
PCI	PCI Bus Binding to IEEE Std1275-1994 Standard for Boot (Initialization Configuration) Firmware, Rev 2.0 found at `http://playground.sun.com/pub/1275/bindings/pci`	
PCI and SBus	*SBus and PCI Bus: a Comparison* (white paper)	n/a
SBus	*1496-1993 IEEE Standard for a Chip Module Interconnect Bus: SBus*, IEEE Computer Society	SH16659-NYF
SBus	*1275.2-1994 IEEE Standard for Boot (Initialization Configuration) Firmware:Bus Supplement for IEEE 1496*	SH94236-NYF
SBus	*SBus Specification B.0*, by Edward H. Frank and Jim Lyle, Sun Microsystems, Inc.,December, 1990	800-5922-10
SBus	*SBus:Information, Applications, and Experience*, James D. Lyle, Springer-Verlag	
SBus	*SBus Handbook*, by Susan A. Mason, SunSoft Press/Prentice-Hall	

Ordering Sun Documents

SunDocs℠ is a distribution program for Sun Microsystems technical documentation. Easy, convenient ordering and quick delivery is available from SunExpress™. You can find a full listing of available documentation on the World Wide Web: `http://www.sun.com/sunexpress/`

Country	Telephone	Fax
United States	1-800-873-7869	1-800-944-0661
United Kingdom	0-800-89-88-88	0-800-89-88-87
France	05-90-61-57	05-90-61-58
Belgium	02-720-09-09	02-725-88-50
Luxembourg	32-2-720-09-09	32-2-725-88-50
Germany	01-30-81-61-91	01-30-81-61-92
The Netherlands	06-022-34-45	06-022-34-46
Sweden	020-79-57-26	020-79-57-27
Switzerland	155-19-26	155-19-27
Japan	0120-33-9096	0120-33-9097

Sun Welcomes Your Comments

Please use the *Reader Comment Card* that accompanies this document. We are interested in improving our documentation and welcome your comments and suggestions.

If a card is not available, you can email or fax your comments to us. Please include the part number of your document in the subject line of your email or fax message.

- Email: `smcc-docs@sun.com`
- Fax: SMCC Document Feedback
  1-415-786-6443

Feature Comparison 1

This chapter compares the more important features of PCI to those of SBus. Topics covered in the tables which need further discussion are covered in the last part of this chapter.

PCI Features of Particular Note

Performance Improvement Using Cache Line Size

The cache line 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 information about related questions, refer to “PCI Developer’s Frequently-Asked Questions”at

`http://shasta.corp.sun.com/Rte66/`

1

Features Tables

These tables compare SBus and PCI features within the categories stated. Topics with additional information later in this chapter have asterisks after them.

Table 1-1 Electrical Features Comparison

Feature	No Difference	SBus	PCI
Power consumption		P=VI	25W (5V x 5A)
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

Table 1-2 Firmware Features Comparison

Feature	No Difference	SBus	PCI
FCode required		All devices.	Boot and console devices.

Table 1-3 Hardware Features Comparison

Feature	No Difference	SBus	PCI
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.
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.	Doesn't care.
Address spaces		Standard chunk of memory-mapped space (28 bits/slot).	3 physical address spaces: memory, I/O, configuration.
Auto-configuration	x		

Table 1-3 Hardware Features Comparison *(Continued)*

Feature	No Difference	SBus	PCI
Burst: Transfer Mode Size		Available, Up to 64 bytes; size declared in advance in SIZ[2:0].	Always. Variable size, no limit; determined by PCI device and PCI bridge.
Bus bandwidth (data transfer rate)		25MHz / 32-bit, max. ~100MBytes / sec for 32-bit. ~200MBytes for 64-bit.	33MHz (rev 2.0) or 66 MHz for EPCI, 32 / 64-bit, maximum rate from 132MByte / second (33MHz / 32 bit) to 528MByte / second (66MHz / 64-bit).
Bus parking		No	Yes. An arbiter may grant the busses to a master when the bus is idle and no masters are generating a request for the bus. If the master that the bus is parked on subsequently issues a request for the bus, it can immediately access it.
Bus width	x 32-to 64-bit		
Clock frequency / bus speed	Synchronous; signals referenced to rising clock edge.	Max CLK 25MHz CLK to out: 22nS Input Set-up: 15nS	33MHz max CLK CLK to out: 11nS Input Set-up: 7nS
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.
Data path		32-bit, except 64-bit for extended transfers	32 / 64-bit
Dynamic Bus Sizing		Yes. Allows slave to control data width it accepts during non-burst transfers.	No, but same function is performed by Byte Enable setting during Data Phase.
Form factors		Single form factor per slot (double and triple-wide can share slots).	3 form factors: short, long, and variable-height short
Interrupts		7 levels.	4 levels (INTA-D)
Max devices per bus		Usually limited by electrical loading.	32 slots

Table 1-3 Hardware Features Comparison *(Continued)*

Feature	No Difference	SBus	PCI
Max functions per device		1	8
Max masters per bus		8	32
Pin count (connector)		96	94 pins for both 32-bit and 64-bit connectors

Table 1-4 Software Features Comparison

Feature	No Difference	SBus	PCI
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.

Table 1-5 Normal Transaction Cycle

Feature	No Difference	SBus	PCI
Address decode		Slave selected by decode from controller.	Each target performs full 32-bit decoding, drives DEVSEL# if selected.
Burst size		32 words	No limit.
Bus driving and turnaround		No.	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 based on byte address.

Table 1-5 Normal Transaction Cycle *(Continued)*

Feature	No Difference	SBus	PCI
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 and Their Functions

Controller

SBus

The SBus controller arbitrates when there is contention between bus masters (during the arbitration phase); it also performs these tasks:

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

PCI

The PCI arbiter performs functions equivalent to the controller's (PCI has no controller per se): arbitrates 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.

Master

SBus

The SBus master controls operations producing error-free data read and write tasks between itself and an SBus slave. These are the master-initiated transaction types:

- Arbitration
- Translation
- Default transfer
- Extended transfer

PCI

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

- Is responsible for starting the Address Phase
- Inserts wait states during data transfer
- Terminates transactions

Slave/Target

SBus

The SBus slave

- 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; master does.
- Participates in burst transfers, dynamic bus-sizing, extended transfer phases.
- Controls the data transfer rate by controlling the data acknowledgment rate (inserts wait states during data transfer).
- Terminates transactions.

PCI

PCI target performs

- Determination that it is the target of a transaction.
- Reception of a data object from the initiator.
- Address decoding
- Participates in special cycles
- Inserts wait states during data transfer(controls data transfer rate)
- Terminates transactions

Protocol

Basic Cycles

These are definitions of the basic transaction cycle for each bus.

SBus

Arbitration Phase

During this phase, masters request bus access. When there is contention between masters, the controller determines which 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 used by the master, and to be used by 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 allows 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 for the fact 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

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.

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

SBus

SBus allows concurrent arbitration and transfer execution. The only requirement made by the specification is that the arbitration method be “fair”, in order to meet the masters’ latency requirements.

When more than one SBus master requests bus access, the controller grants access to one of the requesters. Allowing arbitration concurrent with the master making a data transfer is known as hidden arbitration.

PCI

To minimize access latency, the PCI arbitration approach is access-based, not time-slot-based. Therefore, a bus master must arbitrate for each bus access.

The PCI specification doesn’t 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. They are both low-latency, high-throughput busses. The number of wait states that targets and masters can add to a transaction is limited. Also, masters have programmable

timers which limit their times on the bus during heavy-traffic periods. The limits, plus bus arbitration order, ensure that bus acquisition latencies can quite 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.

Table 1-6 describes the components of bus access latency. 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 1-6 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 elapsed time from the master's issuing a request to the time when the arbiter asserts the master's grant.
Bus acquisition latency	The elapsed time from the requesting master's reception of the grant until the current master surrenders the bus.
Target latency	The elapsed time from transaction start until the currently-addressed target is ready to finish the transactions's first data transfer

Index

E

F

I

L

M

P

R

S

T

V

W