SUN MICROELECTRONICS

STP1031

July 1997

UltraSPARC™-II

DATA SHEET Second Generation SPARC v9 64-Bit Microprocessor With VIS

DESCRIPTION

The STP1031, UltraSPARC–II, is a high-performance, highly-integrated superscalar processor implementing the SPARC-V9 64-bit RISC architecture. The STP1031 is capable of sustaining the execution of up to four instructions per cycle even in the presence of conditional branches and cache misses. This sustained performance is supported by a decoupled Prefetch and Dispatch Unit with Instruction Buffer. Load buffers on the input side of the Execution Unit, together with store buffers on the output side, completely decouple pipeline execution from data cache misses. Instructions predicted to be executed are issued in program order to multiple functional units and execute in parallel. Such predictively-issued instructions can complete out of order. In order to further increase the number of instructions executed per cycle, instructions from different blocks (for instance, instructions before and after a conditional branch) can be issued in the same group.

UltraSPARC-II is part of a second generation of UltraSPARC pipeline-based products. In addition to using a new process technology, the UltraSPARC-II provides a higher clock frequency, multiple SRAM modes and System-to-Processor clock ratios to accommodate multiple price points for system developers. At the same time, it provides software compatibility with existing UltraSPARC-I based systems. UltraSPARC-II also implements the SPARC-V9 Prefetch instruction.

The STP1031 supports both 2D and 3D graphics as well as image processing, video compression and decompression, and video effects through the sophisticated Visual Instruction Set (VIS). VIS provides high levels of multimedia performance, including real-time H.261 video compression/decompression and a single stream of MPEG-2 decompression at full broadcast quality with no additional hardware support.

Features

- SPARC-V9 Architecture Compliant
- Binary Compatible with all SPARC Application Code
- Multimedia Capable, VIS--Visual Instruction Set
- Multi-Processing Support
  - Glueless 4-Processor Connection--Minimum Latency
  - Snooping or Directory Based Protocol Support
- 4-Way SuperScalar Design with 9 Execution Units
  - 4 Integer Execution Units
  - 3 Floating-Point Execution Units
  - 2 Graphics Execution Units
- Software Prefetch Instruction Support
- Selectable Little- or Big-Endian Byte Ordering
- 64-Bit Address Pointers
- 16-Kilobyte Non-blocking Data Cache
- 16-Kilobyte Instruction Cache
  - Single Cycle Branch Following
  - In-Cache 2-Bit Branch Prediction
- Integrated Second Level Cache Controller
- Supports 0.5 - 16 Megabyte Cache Sizes
- Block Load/Store Instructions
- High Bandwidth Processor-Memory Interconnect
  - 1.3 Gigabyte/Sec Processor-Memory Bandwidth
  - 600 MBPS Sustained Processor-Memory Transfers
- Ease of Use
  - JTAG Boundary Scan
  - Performance Instrumentation
- Technology/Packaging
- 0.35um 5-Layer Metal CMOS Process
- Operates at 3.3V and 2.6 V
- 787-Pin Ceramic Land Grid Array (LGA) Package
- Power Management
- Multiple Clocking Modes -- UltraSPARC-I Compatible
- Multiple Outstanding Requests

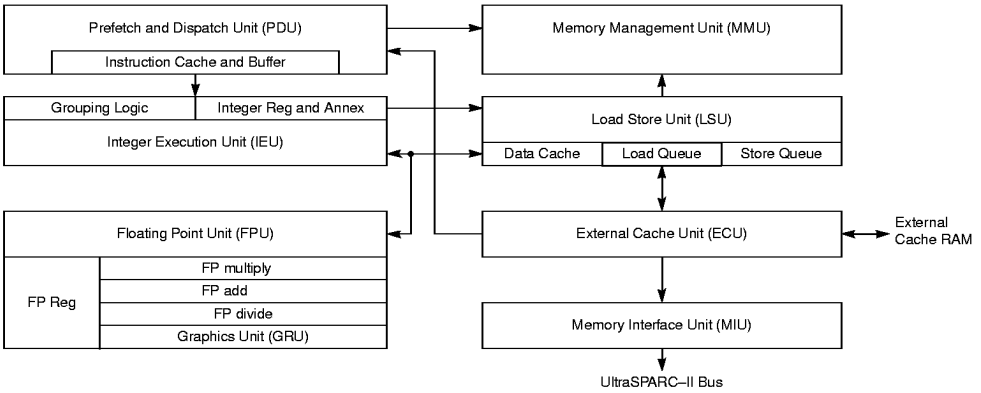


Figure 1. Functional Block Diagram

TECHNICAL OVERVIEW

In a single chip implementation, UltraSPARC–II integrates the following components (see *Figure 1*):

- Prefetch, branch prediction and dispatch unit (PDU)
- 16-Kilobyte instruction cache (I-Cache)
- Memory management unit (MMU) containing two 64-entry buffers
  - a 64-entry instruction translation lookaside buffer (iTLB)
  - a 64-entry data translation lookaside buffer (dTLB)
- Integer execution unit (IEU) with two arithmetic logic units (ALUs)
- Load and store unit with a separate address generation adder
- Load buffer and store buffer decoupling data accesses from the pipeline
- 16-Kilobyte data cache (D-Cache)
- Floating-point unit (FPU) with independent add, multiply and divide/square root sub-units
- Graphics unit (GRU) composed of two independent execution pipelines
- External cache (E-Cache) control unit
- Memory interface unit, responsible for main memory and I/O accesses

Prefetch and Dispatch Unit

The prefetch and dispatch unit fetches instructions a before they are actually needed in the pipeline, so the execution units do not starve. Instructions can be prefetched from all levels of the memory hierarchy, including the instruction cache, the external cache and the main memory. In order to prefetch across conditional branches, a dynamic branch prediction scheme is implemented in hardware. The outcome of a branch is based on a two-bit history of the branch. A "next field" associated with every four instructions in the instruction cache (I-Cache) points to the next I-Cache line to be fetched. The use of the "next field" makes it possible to follow taken branches and basically provides the same instruction bandwidth achieved while running sequential code. Prefetched instructions are stored in the instruction buffer until they are sent to the rest of the pipeline. Up to 12 instructions can be buffered.

Instruction Cache (I-Cache)

The instruction cache is a 16-Kilobyte pseudo-two-way set-associative cache with 32 byte blocks. The cache is physically indexed and contains physical tags. The set is predicted as part of the "next field" so that only the index bits of an address are necessary to address the cache (13 bits, which matches the minimum page size). The instruction cache returns up to 4 instructions from an 8-instruction-wide line.

Memory Management Unit (MMU)

The MMU provides mapping between a 44-bit virtual address and a 41-bit physical address. That is accomplished through a 64-entry iTLB for instructions and a 64-entry dTLB for data, both fully associative. UltraSPARC–II provides hardware support for a software-based TLB miss strategy. A separate set of global registers is available whenever an MMU trap is encountered. Page sizes of 8-, 64-, and 512 Kilobytes and 4 Megabytes are supported.

Integer Execution Unit (IEU)

Two arithmetic logic units (ALUs) form the main computational part of the IEU. An early-out multi-cycle integer multiplier and a multi-cycle integer divider are also part of the IEU. Eight register windows and four sets of global registers are provided (normal, alternate, MMU and interrupt globals). The trap registers (UltraSPARC–II supports five levels of traps) are part of the IEU.

Load/Store Unit (LSU)

The LSU is responsible for generating the virtual addresses of all loads and stores (including atomics and ASI loads), for accessing the data cache, for decoupling load misses from the pipe through the load buffer, and for decoupling the stores through a store buffer. One load or one store can be issued per cycle.

Data Cache (D-Cache)

The data cache is a write-through non-allocating 16-Kilobyte direct-mapped cache with two 16-byte sub-blocks per line. It is virtually indexed and physically tagged. The tag array is dual-ported so that tag updates due to line fills do not collide with tag reads for incoming loads. Snoops to the D-Cache use the second tag port so that an incoming load can proceed without being held up by a snoop.

Floating-Point Unit (FPU)

The separation of the execution units in the FPU allows UltraSPARC–II to issue and execute two floating-point instructions per cycle. Source data and result data are stored in the 32-entry register file, where each entry can contain a 32-bit value or a 64-bit value. Most instructions are fully pipelined (throughput of one per cycle) have a latency of three and are not affected by the precision of the operands. (That is, latency is the same for single or double precision). The divide and square-root instructions are not pipelined. These take 12 cycles (single precision) and 22 cycles (double precision) to execute but they do not stall the processor. Other instructions, following the divide/square root can be issued, executed, and retired to the register file before the divide/square root finishes. A precise exception model is maintained by synchronizing the floating-point pipe with the integer pipe and by predicting traps for long-latency operations.

Graphics Unit (GRU)

UltraSPARC–II introduces a comprehensive set of graphics instructions that provide fast hardware support for two-dimensional and three-dimensional image and video processing, image compression, audio processing, and similar functions. Sixteen-bit and 32-bit partitioned add, boolean and compare are provided. Eight-bit and 16-bit partitioned multiplies are supported. Single cycle pixel distance, data alignment, packing and merge operations are all supported in the GRU.

External Cache Unit (ECU)

The main role of the ECU is to handle I-Cache and D-Cache misses efficiently. The ECU can handle one access per cycle to the external cache.

The ECU also provides support for the 1-1-1 and 2-2 modes of external cache operation. In 1-1-1 mode, the SRAMs are clocked at the processor frequency. In 2-2 mode, the SRAMs are clocked at half the UltraSPARC-II frequency. (See Timing Considerations Section on page 21 for more detail.)

In 1-1-1 mode, accesses to the external cache are pipelined and take three cycles (pin-to-pin), returning 16 bytes of instructions or data per cycle. (In 2-2 mode, the latency is four cycles.) These low latencies can effectively make the external cache a part of the pipeline. For programs with large data sets, this means data can be maintained in the external cache and instructions scheduled with load latencies based on the E-Cache latency. Floating-point applications can use this feature to effectively "hide" D-Cache misses. The size of the external cache can be 512 Kilobytes, 1-, 2-, 4-, 8-, or 16 Megabytes, where the line size is always 64 bytes. A MOESI (modified, own, exclusive, shared, invalid — see Cache Coherence Protocol Section on page 7) protocol is used to maintain coherency across the system. External cache is physically-indexed and physically-tagged.

The ECU provides overlap processing during load and store misses. For instance, stores that hit the E-Cache can proceed while a load miss is being processed. The ECU is also capable of processing reads and writes indiscriminately without a costly turnaround penalty (only 2 cycles are needed). Snoops are also handled by the ECU.

Block loads and block stores load or store a 64-byte line of data from memory to the floating-point register file. These are processed efficiently by the ECU, providing high-transfer bandwidth without polluting the internal or external cache.

Memory Interface Unit (MIU)

The MIU handles all transactions with the system, such as external cache misses, interrupts, snoops, write-backs, and so forth. The MIU communicates with the system at a frequency lower than UltraSPARC–II frequency (the ratio can be 1/2, 1/3, or 1/4).

Both the MIU and the ECU provide support for up to three outstanding loads and up to two outstanding writebacks to the UPA bus.

ULTRASPARC–II SUBSYSTEMS

Subsystem Description

In the following discussion, "system" refers to any other location within the same coherency domain as UltraSPARC–II. For instance, the term includes caches of other processors connected to the interconnect. "Subsystem" refers to the processor and its interface elements.

A complete UltraSPARC–II subsystem consists of the UltraSPARC–II processor, synchronous SRAM components for the external cache tags and data, and the UltraSPARC-II Data Buffer, which has two identical UDB-II devices. The UDB-II isolates the E-Cache from the system and provides data buffers for incoming and outgoing system transactions. The UDB-II also provides ECC generation and checking.

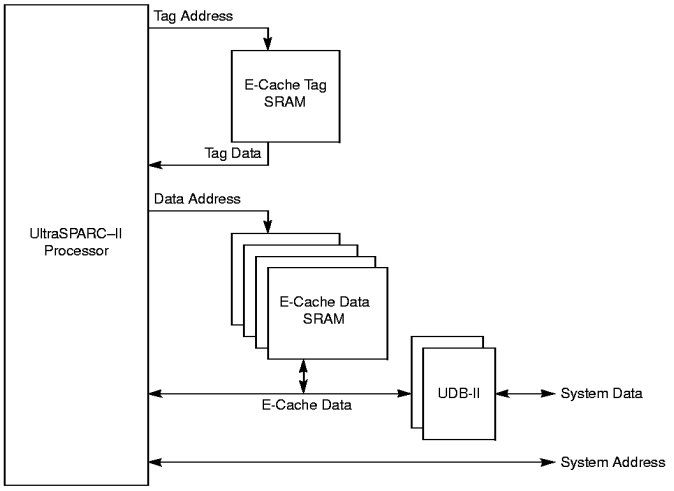


Figure 2. UltraSPARC–II System Interface

Overview of UltraSPARC–II Interface

The main interfaces to and from UltraSPARC–II are shown in *Figure 3*. A typical module includes an external cache composed of the tag unit and the data unit. Both can be implemented using commodity SRAMs. Separate address and data buses are provided from and to the tag and data SRAMs for increased performance.The main role of the UDB-II is to isolate UltraSPARC–II and its external cache from the main system data bus so that the interface can operate at processor speed (reduced capacitance loading). The data buffer also provides overlapping between system transactions and local E-Cache transactions even when the E-Cache transactions need to use part of the data buffer. The logic to control the UDB-II is included on UltraSPARC–II to provide fast data transfers from and to UltraSPARC–II or from and to the external cache and the system. A separate address bus and separate control signals are provided for supporting system transactions. Clock signals, reset pins, observability pins and JTAG support are also part of UltraSPARC–II interfaces.

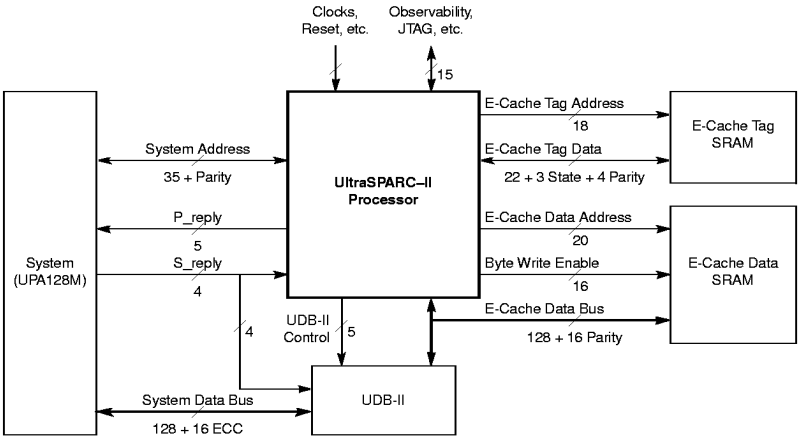


Figure 3. Main UltraSPARC–II Interfaces

Cache Coherence Protocol

This section describes the protocol used to maintain coherency between UltraSPARC–II's internal caches, the external cache and the system.

Inclusion in the E-Cache is maintained for both the I-Cache and the D-Cache. (All lines containing data currently held in the internal caches are in the external cache, even when the caches are turned off.) The state of these lines forms a part of the tag kept in the external tag RAM.

The cache coherence protocol is point-to-point write-invalidate. It is based on the 5 MOESI states maintained in the E-Cache tags of each master port (that is, each UltraSPARC–II). The E-Cache tags have one of the following five states (MOESI):

- Exclusively Modified (M)
- Shared Modified (O)
- Exclusive Clean (E)

- Shared Clean (S)
- Invalid (I)

Three bits in the tag RAM define the state of each line as follows:

TABLE 1: External Cache Coherency State Definition

STATE	State Bit		
	Valid	Modified	Exclusive
Invalid (I)	0	0	0
Shared Clean (S)	1	0	0
Exclusive Clean (E)	1	0	1
Shared Modified (O)	1	1	0
Exclusively Modified (M)	1	1	1

The cache coherence protocol operates only on physically indexed physically tagged (PIPT) writeback caches. The unit of cache coherence is a block size of 64 bytes which corresponds to one E-Cache line. Coherent read/write transactions transfer data in a 64-byte blocks only, using 4 quadwords.

The state diagram representing the allowed transactions is shown in *Figure 4*.

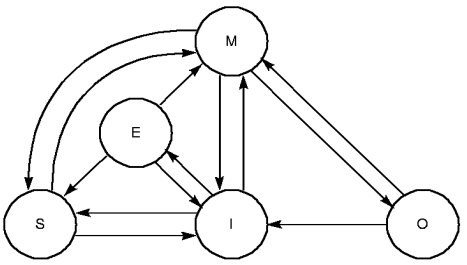


Figure 4. Cache Coherency Protocol State Diagram

Table 2 describes all transitions between the states shown in *Figure 4*. It also shows the transactions that are initiated by either UltraSPARC–II or the system and the acknowledgment that is expected after completion of each transaction.

TABLE 2: Transitions Allowed for Cache Coherency Protocol

Transition	Description	Transition Req to and from Port	Acknowledgment
I → E	Load miss; data coming from memory to an invalid line (no other cache has the data).	P_RDS_REQ	S_RBU
I → S	Load miss; data provided by another cache or memory to an invalid line (another cache has the data or I-Cache miss).	P_RDS_REQ P_RDSA_REQ	S_RBS S_RBS
I → M	Store miss, atomic miss on invalid line.	P_RDO_REQ	S_RBU
E → M	Store hit or atomic hit to Exclusive Clean line.	No Transaction	No Transaction
E → S	Request from system to share this line (load miss from another processor).	S_CPB_REQ, S_CPB_MSI_REQ	P_SACK | P_SACKD *followed by* S_CRAB
E → I	i) A clean line is victimized by the processor. I-Cache miss. Write miss.	P_RDS_REQ or P_RDSA_REQ or P_RDO_REQ	S_RBU or S_RBS S_RBS S_RBU
	ii) Request from system to copyback and invalidate this line (store miss from another processor).	S_CPI_REQ	P_SACK|P_SACKD *followed by* S_CRAB
	iii) Request from SC to invalidate this line (block store from another processor)	S_INV_REQ	P_SACK|P_SACKD
S → M	Store hit, atomic hit to Shared Clean line	P_RDO_REQ	S_OAK
S → I	i) A Shared Clean line is victimized by UltraSPARC. I-Cache miss. Write hit on shared line.	P_RDS_REQ or P_RDSA_REQ or P_RDO_REQ	S_RBU or S_RBS S_RBS S_RBU
	ii) Another processor wants to write this shared line.	S_INV_REQ or S_CPI_REQ	P_SACK|P_SACKD P_SACK|P_SACKD *followed by* S_CRAB
	iii) Request from SC to invalidate this line (block store from another processor).	S_INV_REQ	P_SACK|P_SACKD
M → O	Request from another processor to read a modified line, memory is not updated (as opposed to M → S).	S_CPB_REQ	P_SACK|P_SACKD *followed by* S_CRAB

TABLE 2: Transitions Allowed for Cache Coherency Protocol (Continued)

Transition	Description	Transition Req to and from Port	Acknowledgment
M, O → I	i) A Modified line is victimized by the processor (Writeback).	P_WRB_REQ	S_WAB or S_WBCAN *if system takes ownership before completing Writeback*
	ii) Request from system to copyback and invalidate this line (store miss from another processor).	S_CPI_REQ	P_SACK|P_SACKD *followed by* S_CRAB
	iii) Request from system to invalidate this line (block store from another processor)	S_INV_REQ	P_SACK|P_SACKD
M, O → S	Request from another processor to read this line, memory is updated so line becomes clean (*c.f.* M → O)	S_CPB_MSI_REQ	P_SACK|P_SACKD *followed by* S_CRAB
O → M	Store hit, atomic hit to Modified line, PREFETCH.	P_RDO_REQ	S_OAK

UltraSPARC-II as a Bus Port

The UltraSPARC Port Architecture (UPA) defines protocols for a family of tightly coupled, cache consistent, shared memory multiprocessor systems. The UltraSPARC–II bus provides low latency to memory, high bandwidth and fast multiprocessor data sharing. UltraSPARC–II bus transactions are carried over a packet-switched bus with independent scheduling of separate (and possibly multiple) address and data buses.

UltraSPARC–II and its cache subsystem (including the UDB-II) form a UPA128M module. This module interfaces to the interconnect using the system bus interface definition called the UltraSPARC Port Architecture (UPA). (For more information, refer to the manual entitled *UPA Interconnect Architecture.*) Similarly, the I/O subsystem and the graphics subsystem reside on a UPA bus as UPA64M or UPA64S.

The physical connections between UltraSPARC–II and the UPA mainly consist of the following:

- a bidirectional address bus for transaction requests between UltraSPARC–II and the UPA bus interface
- two unidirectional (one incoming, one outgoing) reply buses for flow control
- a bidirectional request for a distributed address bus arbitration scheme.

The snoop bus, although not part of the UltraSPARC–II module, is present in the system controller ASIC and is used to manage duplicate tags, and for efficient data sharing.

Table 3 shows the UPA bus port interface as specified in the *UPA Interconnect Architecture*, and the corresponding pins for UltraSPARC–II.

TABLE 3: UPA Port Interface

UPA Port Interface		UltraSPARC–II Interface
UPA_DataBus[144]	<--->	EDATA[127:0], EDPAR[15:0]
UPA_ECC_Valid[2]	<--->	
UPA_AddressBus[37]	<--->	SYSADDR[38]
UPA_Addr_Valid	<——>	ADR_VLD
UPA_Addr_Arb[5]	<——>	NODE_RQ[2:0], REQUEST_OUT, SC_RQ
UPA_P_REPLY[5]	<---	P_REPLY[5]
UPA_S_REPLY[5]	--->	S_REPLY[3:0]
UPA_SnoopBus[36]	--->	
UPA_SnoopCntl[13]	<——>	
UPA_Port_ID[5]	——>	
UPA_Reset	——>	RESET_L
UPA_Sys_Clk[2]	——>	CLKA, CLKB
UPA_ClkCntl[4]	<--->	
UPA_JTAG[4]	——>	
UPA_Slave_INT	——>	
UPA_Mode	——>	
UPA_Wakeup_Reset	——>	

UltraSPARC–II is both a bus master and a bus slave. As UPA master it issues read/write transactions to the interconnect using part of the UPA transaction set. UltraSPARC–II splits transactions into two independent classes:

- Class 0 contains read transactions due to cache misses and block loads.
- Class 1 contains writeback requests, write invalidate requests, block stores, interrupt requests, and non-cached read/write requests.

Transactions in each class are strongly ordered by the interconnect. As a bus master, UltraSPARC–II has a physically addressed coherent cache (the E-Cache), which participates in the MOESI cache coherency protocol, and responds to the interconnect for copyback and invalidation requests.

As a bus slave, UltraSPARC–II responds to a non-cached read of its bus port identification.

UltraSPARC–II is both an interrupter and an interrupt receiver. It has the capability to generate interrupt packets to other UPA interrupt receivers, and it can receive interrupts coming from other interrupters.

UPA Transactions Supported by UltraSPARC–II

Transactions Initiated by UltraSPARC–II

The UPA transactions initiated by UltraSPARC–II are sent off through the system address bus. Four bits in the packet identifying the transaction type are encoded according to the UPA definition of the corresponding transaction.

- P_RDS_REQ (Read To Share): Coherent read with intent to share; UltraSPARC–II issues this in response to a load miss.
- P_RDSA_REQ (Read to Share Always): Coherent read with intent to share "always;" UltraSPARC–II issues this in response to an E-Cache miss generated by an instruction fetch.
- P_RDO_REQ (Read to Own): Coherent read with invalidate; UltraSPARC–II issues this in response to a store miss, a store hit on a shared line, or a read-with-intent-to-write for merging partial writes, such as for read-modify-writes.
- P_RDD_REQ (Read to Discard): Coherent read with no intent to cache the data; UltraSPARC–II issues this during block loads.
- P_WRB_REQ (Writeback): Generated when a dirty victimized block from the E-Cache must be written back to its home location. The writeback is associated with a prior coherent read transaction to the same E-Cache location.
- P_WRI_REQ (Write Invalidate): Coherent write and invalidate request; UltraSPARC–II issues this during a block store.
- P_INT_REQ (Interrupt): Interrupt transaction request packet. UltraSPARC–II issues this to deliver a 64 byte interrupt packet to the destination (see ASI Registers definition in *UltraSPARC User's Manual*).
- P_NCRD_REQ (Noncached Read): Used when a load or a block load is issued to a noncacheable location. One, 2, 4, 8, or 16 bytes can be read with this transaction.
- P_NCBRD_REQ (Noncached Block Read): Used when a block read (64 bytes) is made to a noncacheable location.
- P_NCBWR_REQ (Noncached Block Write): Generated by UltraSPARC–II when a block write (64 bytes) is made to a noncacheable location.

System Transactions Accepted by UltraSPARC–II

- S_INV_REQ (Invalidate): Invalidate request from the system controller to UltraSPARC–II following a Read To Own (P_RDO_REQ) or Write Invalidate (P_WRI_REQ) request for a block from another UltraSPARC–II.
- S_CPB_REQ (Copyback): Copyback request from the system controller to UltraSPARC–II following a Read To Share (P_RDS_REQ) or Read To Share Always (P_RDSA_REQ) request for a block from another UltraSPARC–II.
- S_CPI_REQ (Copyback Invalidate): Copyback and Invalidate request from the system controller to UltraSPARC–II in response to a Read To Own (P_RDO_REQ) request for a block from another UltraSPARC–II.
- S_CPD_REQ (Copyback To Discard): Sent at the UltraSPARC–II bus interface to UltraSPARC–II in order to service a Read To Discard (P_RDD_REQ) issued by another UltraSPARC–II. This transaction does not generate a state change for the E-Cache and does not require a flush of the store buffer tag check.

- P_NCRD_REQ (Noncached Read): The only slave read transaction that should be sent to UltraSPARC–II. UltraSPARC–II responds to this request by sending the value of its bus port identification to the data bus. The transaction starts as a P_NCRD_REQ from an UPA master, and is forwarded by the system controller to UltraSPARC–II. UltraSPARC–II replies through a P_RAS; then the system controller issues an S_SRS to drive the data on SYSDATA. Finally the requesting master gets the data after the system controller issues S_RAS.
- P_INT_REQ (Interrupt): Interrupt transaction request packet sent by another UltraSPARC. In parallel, the 64-byte interrupt data packet is placed on SYSDATA, and S_SWIB is sent to instruct the UDB-II to accept the data.

Responses to Transactions Initiated by the System (P_Reply)

P_reply is an acknowledgment from UltraSPARC–II to the system, in response to a request that the system sent to UltraSPARC–II previously. There are five unidirectional (output only) pins on UltraSPARC–II connected directly to the system.

- P_IDLE (Idle): This is the default state of the wires. It indicates no reply.
- P_SNACK (Non-Existent Block): Reply by UltraSPARC–II indicating that the requested block from a snoop does not exist in the external cache (only set when DTAGs are not present).
- P_RAS (Read Acknowledge Single): 16 bytes of read data is ready in the output data queue on the UDB-II. Sent following a single noncacheable read request from an UltraSPARC–II (reply to P_NCRD_REQ).
- P_SACK (Coherent Read Acknowledge Block): Asserted for a coherent (snoop) S_REQ when the data is in the cache and not pending a writeback in response to victimization. Indicates data is available in the UDB-II. (Reply to S_CPB_REQ, S_CPD_REQ, S_CPI_REQ, or S_INV_REQ.)
- P_SACKD (Coherent Read Acknowledge Block for Dirty Victim): Asserted for a coherent (snoop) S_REQ when the data has been victimized and is pending a writeback. Indicates data is available in the UDB-II. (Reply to S_CPB_REQ, S_CPD_REQ, S_CPI_REQ, or S_INV_REQ.)
- P_IAK (Interrupt Acknowledge): UltraSPARC–II sends a P_IAK to acknowledge that the interrupt transaction delivered by the system has been serviced. This implies that there is room on the UDB-II for another interrupt request and its 64 bytes of data.
- P_RERR (Read Error): Returned by UltraSPARC–II in response to a noncached block read request (P_NCBRD) sent to it. No data is transferred. Cacheable read requests produce undefined results.
- P_FERR (Fatal Error): Sent when UltraSPARC–II detects a parity error on the SYSADDR bus or E-Cache tags. Indicates that the system should generate a system-wide power-on reset.

TABLE 4: P_REPLY Encodings

Type	Cycles	Name	Reply to Which Transaction	Class	Type
P_IDLE	1	Idle	Default State	0	00000
P_FERR	1	Fatal Error	All Transactions, Any Time	X	0100
P_RERR	2	Read Data Error	P_NCBRD_REQ	C	0101
P_SNACK	2	Coherent S_REQ Non-Existent ACK	S_REQ	C	0111
P_RAS	2	Read Acknowledge Single	P_NCRD_REQ	C	1000
P_SACK	2	Coherent S_REQ ACK	S_REQ	C	1010
P_IAK	2	Interrupt Acknowledge	P_INT_REQ	C	1100
P_SACKD	2	Coherent S_REQ Dirty Victim ACK	S_REQ	C	1101

The class values are indicated as follows:

- 0 = hardwired to 0
- X = don't care
- C = Copied from the associated P_REQ packet

System Responses (S_REPLY) to Transaction Requests (P_REQ) or Acknowledgment (P_REPLY) from UltraSPARC–II

This is also a unidirectional point-to-point connection between the system and UltraSPARC–II.

- S_IDLE (Idle): The default state of the wires. It indicates no reply.
- S_RTO (Read Time-out): Forwards the Read Time Out (P_RTO) reply from the slave that UltraSPARC–II tried to access. Note that time-outs on writes are reported asynchronously via interrupt by the detecting slave UltraSPARC–II.
- S_ERR (Error): Asserted by the system if the error conditions described in the *UPA Interconnect Architecture* occur.
- S_WAS (Write Acknowledge Single): Generated by the system following a noncacheable write request from UltraSPARC–II (P_NCWR_REQ). It causes 16 bytes of data from UDB-II to be put on SYSDATA.
- S_WAB (Write Acknowledge Block): Generated by the system following a noncacheable block store (P_NCBWR_REQ), a writeback request (P_WRB_REQ), or a write invalidate request during a block store with invalidate (P_WRI_REQ). It causes 64 bytes of data to be put on SYSDATA.
- S_OAK (Ownership Acknowledge Block): Generated by the system when UltraSPARC–II wants permission to write to a block that is already in the E-Cache. No data transfer occurs.
- S_RBU (Read Block Unshared Acknowledge): The system commands the input data queue of UDB-II to accept 64 bytes of unshared or noncached data from SYSDATA. This is in response to a P_RDS_REQ, a P_RDO_REQ, or a P_NCBRD_REQ.
- S_RBS (Read Block Shared Acknowledge): The system commands the input data queue of the UDB-II to accept 64 bytes of shared data from SYSDATA. This is a response to a P_RDS_REQ or a P_RDSA_REQ.

- S_RAS (Read Acknowledge Single): The system commands UDB-II to accept 16 bytes of data from SYSDATA. This is a response to a noncacheable read request (P_NCRD_REQ).
- S_SRS (Read Single Acknowledge): The system commands UDB-II to drive 16 bytes of data onto SYSDATA. This follows a P_RAS from UltraSPARC–II indicating that the data was ready in the UDB-II.
- S_CRAB (Copyback Read Block Acknowledge): The system commands UDB-II to drive 64 bytes of copyback data onto SYSDATA. This follows a P_SACK or P_SACKD from UltraSPARC–II indicating that the copyback data was ready in the UDB-II.
- S_SWIB (Interrupt Write Block Acknowledge): The system commands UDB-II to accept 64 bytes of interrupt data from SYSDATA. In parallel the P_INT_REQ packet that was initiated by the interrupting UPA port is sent to the target UltraSPARC–II on SYSADDR.
- S_WBCAN (Writeback Cancel Acknowledge): The system generates this to cancel a previous writeback by UltraSPARC–II (P_WRB_REQ).
- S_INAK (Interrupt NACK): The system generates this if the receiver of an UltraSPARC–II interrupt request (P_INT_REQ) cannot accept another interrupt packet at the moment. This reply effectively removes the interrupt packet from the UDB-II queue; software on the originator should retry later. This is the only transaction that is NACK'ed by the system; S_INAK sets a bit in an ASI register on UltraSPARC (see *UltraSPARC User's Manual*).

TABLE 5: S_REPLY Encodings

S_REPLY	Name	Reply to Which Transaction	Type
S_IDLE	Idle	Default State	0000
S_ERR	Error	Report Error to Master	0001
S_CRAB	Coherent Read Acknowledge Block	To Slave for P_CRAB reply	0010
S_WBCAN	Writeback Cancel	To Master for P_WRB_REQ	0011
S_WAS	Write Acknowledge Single	To Master for P_NCWR_REQ	0100
S_WAB	Write Acknowledge Block	To Master for Any Block Write	0101
S_OAK	Ownership Acknowledge	To Master for P_RDO_REQ	0110
S_INAK	Interrupt Nack	To Master for P_INT_REQ	0111
S_RBU	Read Block Acknowledge Unshared	To Master for Any Block Read	1000
S_RBS	Read block Acknowledge Shared	To Master for Coherent Shared Read	1001
S_RAS	Read Acknowledge Single	To Master for P_NCRD_REQ	1010
S_RTO	Read Time Out	To Master, Forwarding of P_RTO	1011
S_SRS	Slave Read Single	Read 16 Bytes of Data from Slave	1110
S_SWIB	Slave Write Interrupt Block	Write 64 Bytes of Interrupt Data to Slave	1101
Reserved	—	—	1111

Interaction between the E-Cache and UltraSPARC-II Data Buffer (UDB-II)

External cache accesses, although synchronous to the internal clock (for instance, ALU operations), are not closely coupled to the pipeline. Full throughput to the external cache is supported and can make the E-Cache look like a very large D-Cache. The micro architecture used to support this feature consists of the load buffer, dual ported tags, separate address buses for tag and data, and so forth.

The UDB-II isolates the system data bus from UltraSPARC–II. It allows data transfers between UltraSPARC–II and the memory system (that is, noncacheable stores) or I/O. It also allows data transfers between the E-Cache and the memory system (that is, writebacks) to occur much more rapidly since system arbitration and system throughput are hidden by the internal buffering of the UDB-II. Overlapping of transactions is also possible, which increases overall bandwidth. Interrupt "packets" are also handled by the UDB-II, which also generates and checks error correction code (ECC).

The external cache consists of two parts:

- E-Cache Tag RAM, which contains the physical tags of the cached lines and 3 bits of state information
- E-Cache Data RAM, which contains the data for each cache line

Both these parts can be built out of commodity RAMs. The parts operate synchronously with UltraSPARC–II (Synchronous Static RAMs). The external cache sizes supported by UltraSPARC–II are: 512 Kilobytes, 1-, 2-, 4- and 16 Megabytes. The size of the cache is established at boot time by software.

Each byte in the RAMs is accompanied by a parity bit (three bits for the tags and 16 bits for data).

The clients for the external cache are UltraSPARC–II and the UDB-II. More specifically for UltraSPARC-II they are the load buffer, the store buffer, the prefetch unit, and the data buffer. If the working set is too large for the D-Cache, it may still fit the E-Cache. Therefore, loads that miss the D-Cache are sent to the E-Cache. All cacheable stores go to the E-Cache (the D-Cache is write-through) but not necessarily in order with respect to load accesses. All I-Cache misses generate a request for the E-Cache. The UDB-II returns data from main memory during an E-Cache miss or a load to noncacheable locations. Writebacks (the process of writing a dirty line back to memory before a fill), generate data transfers from the E-Cache to the data buffer, controlled entirely by the CPU. Copybacks (responses to snoop hits) also generate transfers from the E-Cache to the UDB-II.

Each UDB-II has a 12-entry-by-16-byte read buffer that can hold three 64-byte lines coming from main memory in response to an E-Cache read miss or a noncacheable read. The outgoing buffer (that is, the buffer receiving data from the UltraSPARC–II and sending it to the system) is divided into three parts. There are two 8×16-byte writeback buffers, an 8×16-byte noncacheable store buffer, and a 4×16-byte snoop buffer. Note that the writeback buffer can be snooped; consequently, internal bypass is provided to send the writeback data to the port requesting the snoop on the interconnect. Three 64-bit registers are provided to hold an incoming interrupt data packet, while three more are provided to hold an interrupt data packet waiting to be sent.

SIGNAL DESCRIPTIONS

Quick Pin Reference - UPA Interface (3.3 Volts)

Symbol	Type	Name and Function
SYSADR[35:0]	I/O	Bidirectional UltraSPARC-II transaction request bus. Maximum of 3 other masters and 1 system controller can be connected to this bus. Synchronous to UDBCLK A/B. (3.3V, UPA)
ADR_VLD	I/O	Bidirectional radial UltraSPARC-II bus signal between UltraSPARC-II and the system. Driven by UltraSPARC-II to initiate SYSADR transactions to the system. Driven by the system to initiate coherency, interrupt or slave transactions to UltraSPARC-II. Synchronous to the system clock. (3.3V, UPA)
NODE_RQ[2:0]	I	UltraSPARC-II system address bus arbitration request from up to 3 other UltraSPARC-II bus ports that might be sharing the SYSADR. Used by UltraSPARC-II for the distributed SYSADR arbitration protocol. Connection to other UltraSPARC-II bus ports is strictly dependent on the master identification allocation. Synchronous to system clock. (3.3V, UPA)
SC_RQ	I	UltraSPARC-II system address bus arbitration request from the system. Used by UltraSPARC-I for the distributed SYSADR arbitration protocol. Synchronous to system clock. (3.3V, UPA)
S_REPLY[3:0]	I	UltraSPARC-II system reply packet, driven to UltraSPARC-II. Bit 4 of the UltraSPARC-II bus S_REPLY is not used by UltraSPARC-II. Synchronous to system clock. (3.3V, UPA)
DATA_STALL	I	This is asserted with or after an S_REPLY to hold output system data or signal the delay in arrival of input data from the system. (3.3V, UPA)
P_REPLY[4:0]	O	UltraSPARC-II processor reply packet, driven by UltraSPARC-II to the system. Synchronous to system clock. (3.3V, UPA)
NODEX_RQ	O	UltraSPARC-II system address bus arbitration request. Asserted when UltraSPARC-II needs to drive SYSADR. Connected to all other UltraSPARC-II bus ports that share this address bus, and to the system. Synchronous to system clock. (3.3V, UPA)

Quick Pin Reference - External Cache Interface (2.6 Volts)

Symbol	Type	Name and Function
EDATA[127:0]	I/O	E-Cache data bus. Connects UltraSPARC–II to the E-Cache data SRAMs and the UDB-II. Synchronous to UDBCLK A/B.
EDPAR[15:0]	I/O	Data bus parity. Odd parity is driven for all EDATA transfers, and checked if UltraSPARC–II or the UDB-II is the receiver. The most significant bit serves as the parity for the most significant byte of EDATA. Synchronous to processor clock.
TDATA[24:0]	I/O	Bidirectional data bus for E-Cache tag SRAMs. Bits [24:22] carry the MOESI state: Dirty, Exclusive, Valid. Bits [21:0] carry the physical address bits [40:19]. This allows a minimum cache size of 512 Kilobytes. All of the TDATA bits are used, even when the E-Cache is more than 512 Kilobytes. This is because there is no sizing in the tag compare for E-Cache hit generation. Synchronous to processor clock.
TPAR[3:0]	I/O	Bidirectional data bus for E-Cache tag SRAMs. Odd Parity for TDATA[24:0]. TPAR[3] covers TDATA[24:22]. TPAR[2] covers TDATA[21:16]. TPAR[1] covers TDATA{15:8], TPAR[0] covers TDATA[7:0]. Synchronous to processor clock.
BYTEWE_L[15:0]	O	Byte write enables for synchronous pipelined E-Cache SRAMs. Bit [0] controls EDATA[127:120]. Bit [15] controls EDATA[7:0]. Byte write control is necessary because the first-level data cache is write-through. Synchronous to processor clock.
ECAD[19:0]	O	Address for E-Cache data SRAMs. Corresponds to physical address [23:4]. Allows a maximum 4 Megabyte E-Cache. Synchronous to processor clock.
ECAT[17:0]	O	Address for E-Cache tag SRAMs. Corresponds to physical address [23:6]. Allows a maximum 4 Megabyte E-Cache. Synchronous to processor clock.
DSYN_WR_L	O	Write enable for E-Cache data SRAMs. Active low. Synchronous to processor clock.
DOE_L	O	Active low for all SRAM data reads and writes. Synchronous to processor clock.
TSYN_WR_L	O	Write enable for E-Cache tag SRAMs. Active low. Synchronous to processor clock.
TOE_L	O	Active low for all tag data SRAM reads and writes. Synchronous to processor clock.
ECACHE_22_MODE	I	Selects E-Cache 2-2 mode operation.
PHASE_DET_CLK	I	Phase detection clock for PLL bypass operation.
MCAP[3:0]	I	Module speed capability.
SPARE_OUT	O	Spare output. Not connected on the module.

Quick Pin Reference - Clock Interface

Symbol	Type	Name and Function
CLKA, CLKB	I	These are differential inputs of the primary CPU clock. CLKA is the positive differential clock input and CLKB is the negative differential clock input. The clock reference for the device timing occurs when the rising edge of CLKA crosses the falling edge of CLKB.
UDBCLKA, UDBCLKB	I	These are differential inputs of the system clock. They are used to generate the phase signal which allows UltraSPARC–II to synchronize communication to the system with respect to the system clock.
PLLBYPASS	I	When asserted this pin causes the phase-lock loop to be bypassed. The clock from the differential receiver is directly passed to the clock trunk.
STOP_CLOCK	O	Indicates clock has stopped.
L5CLK	O	A buffered version of the internal level 5 clock of UltraSPARC–II. Used to determine PLL lock or clock tree delay when UltraSPARC–II is in PLL bypass mode.

Quick Pin Reference - JTAG/Debug Interface (3.3 Volts)

Symbol	Type	Name and Function
TDO	O	IEEE 1149 test data output. A three-state signal driven only when that TAP controller is in the shift-DR state.
TDI	I	IEEE 1149 test data input. This pin is internally pulled to logic one when not driven.
TCK	I	IEEE 1149 test clock input. This pin if not hooked to a clock source must always be driven to a logic 1 or a logic 0.
TMS	I	IEEE 1149 test mode select input. This pin is internally pulled to logic one when not driven.
TRST_L	I	IEEE 1149 test reset input (active low). This pin is internally pulled to logic one when not driven.
RAM_TEST	I	When asserted this pin forces the processor into SRAM test mode allowing direct access to the cache SRAMs for memory testing.
MISC_BIDIR[14:0]	I/O	These are miscellaneous bidirectional signals used for test, debug and instrumentation. Some of them are used to improve internal operation observability, such as pipeline monitoring signals. Their exact functions are TBD.
EXT_EVENT	I/O	This signal is used to indicate the clock should be stopped. It is a debug signal which is set inactive on production systems.
PM_OUT	O	Used for on-chip process monitors (reserved exclusively for IC manufacturing use).
TEMP_SEN[1:0]	O	Defines the end points of the temperature sense element on the module used to measure the processor temperature (reserved exclusively for IC manufacturing use).

Quick Pin Reference - Initialization Interface (3.3 Volts)

Symbol	Type	Name and Function
RESET_L	I	Driven for POR (power-on) resets. Asserted asynchronously. Deasserted synchronous to system clock. Active low.
XIR_L	I	Driven to signal XIR_L resets. Actually acts like a non-maskable interrupt. Synchronous to system clock. Active low.
EPD	O	Asserted when UltraSPARC–II is in power-down mode.

Quick Pin Reference - UDB-II Chip Interface (2.6 Volts)

Symbol	Type	Name and Function
UDB_UEH	I	Asserted when the High UDB-II drives EDATA[127:64], if there is an uncorrectable ECC error associated with that data. Synchronous to system clock.
UDB_UEL	I	Asserted when the Low UDB-II drives EDATA[63:0], if there is an uncorrectable ECC error associated with that data. Synchronous to system clock.
UDB_CEH	I	Asserted when the High UDB-II drives EDATA[127:64], if the data has a corrected single-bit error. Synchronous to system clock.
UDB_CEL	I	Asserted when the Low UDB-II drives EDATA[63:0], if the data has a corrected single-bit error. Synchronous to system clock.
UDB_CNTL[4:0]	O	UltraSPARC–II controls the drive and receive of the UDB-II EDATA. Asserted with valid EDATA when driving data to UDB-II. Asserted the cycle before the UDB-II should drive data. Synchronous to system clock.

TIMING CONSIDERATIONS

This section describes the logical timing for the transactions occurring between UltraSPARC–II, the external cache, and the data buffer. The diagrams are based on a clock with a 50% duty cycle. The transitions represented in the diagrams show what occurs at the pins of UltraSPARC–II. The position of the transitions relative to the clock transitions is correct but not drawn to scale. (For instance, a set up time of 1 nanosecond is represented by showing the transition of the incoming signal changing slightly before the rising edge of the clock.)

The examples shown in the following pages assume the 1-1-1 mode of operation for the external cache. In this mode, the SRAMs are clocked at the processor frequency. (See Figure 5 on page 21, Figure 6 on page 22, Figure 7 on page 23, Figure 8 on page 23, and Figure 9 on page 24.)

Coherent Read Hit in 1-1-1 Mode

Coherent reads that hit the E-Cache are represented in *Figure 5*. With UltraSPARC–II there is no difference between burst reads and two consecutive reads. The signals used for a single read are simply duplicated for each subsequent read.

The timing diagram shows three consecutive reads that hit the E-Cache. The control signals (TSYN_WR_L, TOE_L) and the address for the tag read (ECAT) as well as the control signals (DSYN_WR_L, DOE_L) and the address for the data (ECAD) are shown to transition shortly after the rising edge of the clock. Two cycles later, the data for both the tag read and data read is back at the pins of the CPU shortly before the next rising edge. (This meets set up time and clock skew.) Notice that the reads are fully pipelined and thus full throughput is achieved. (There are three requests made before the data of the first request comes back and the latency of each request is three cycles).

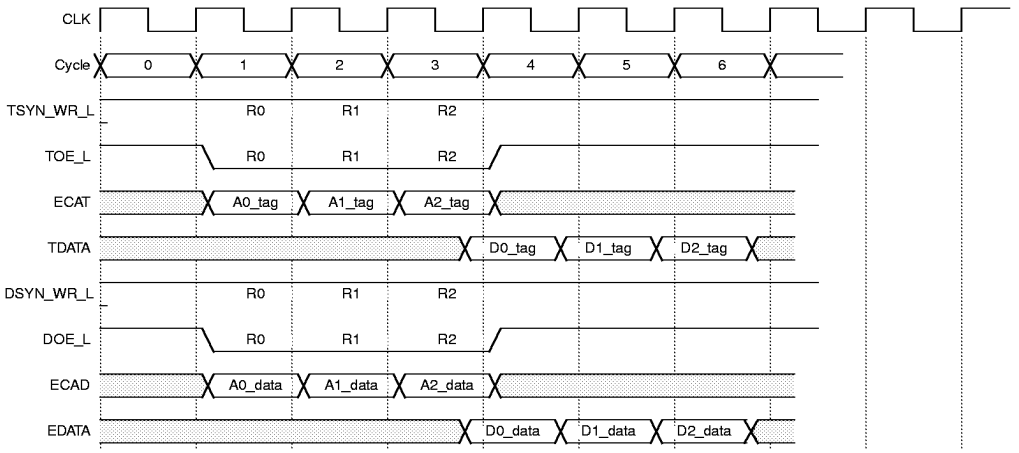


Figure 5. Coherent Read Hit Timing

Coherent Write Hits in 1-1-1 Mode

Writes to the external cache are processed through independent tag and data transactions. First the tag and the state bits of the E-Cache line corresponding to the write are read. If the access is a hit and the state is exclusive or modified, the data is written to the data RAM.

In the timing diagram in *Figure 6*, three consecutive write hits to M state lines are shown. Access to the first tag (D0_tag) is started by asserting TSYN_WR_L and TOE_L and by sending the tag address (A0_tag). In the cycle after the tag data (D0_tag) comes back, it is determined by UltraSPARC–II that the access is a hit and that the line is in M state (Modified). In the next clock cycle, a request is made to write the data. The data address is presented on the ECAD pins in the cycle after the request (cycle 7 for W0) and the data is sent in the following cycle (cycle 8), as shown in *Figure 6*. Separating the address and the data by one cycle reduces the turn around penalty when reads are immediately followed by writes (discussed in "Coherent Read Followed by a Coherent Write in 1-1-1 Mode" on page 24).

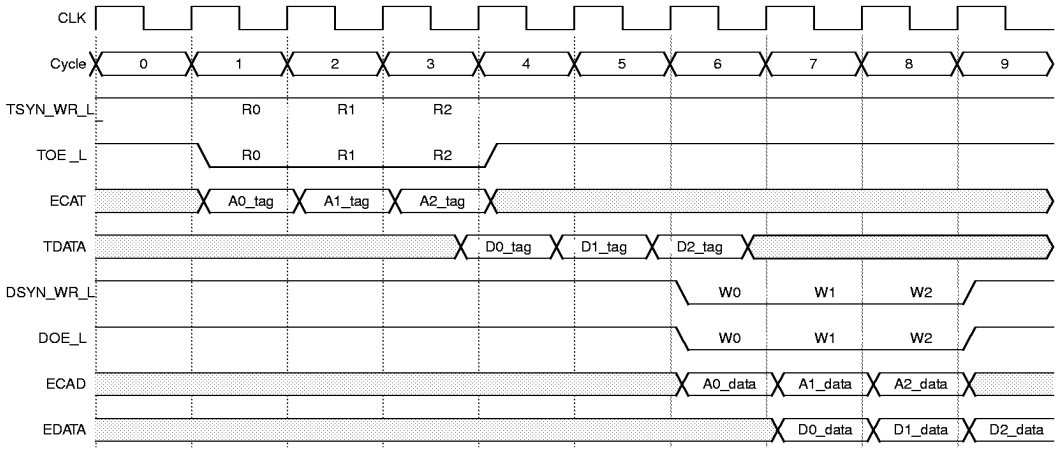


Figure 6. Coherent Write Hit to M State Line

If the line is in exclusive state then the tag is updated to Modified at the same time as the data is written as shown in Figure 7 on page 23. Otherwise, the tag port is available for a tag check of a younger store during the data write. In the timing diagram, the store buffer is empty when the first write request is made. That is why there is no overlap between the tag accesses and the write accesses. In normal operation the tag access for one write can be done in parallel with the data write of the previous write. This independence of the tag and data buses make the peak store bandwidth as high as the load bandwidth (one per cycle).

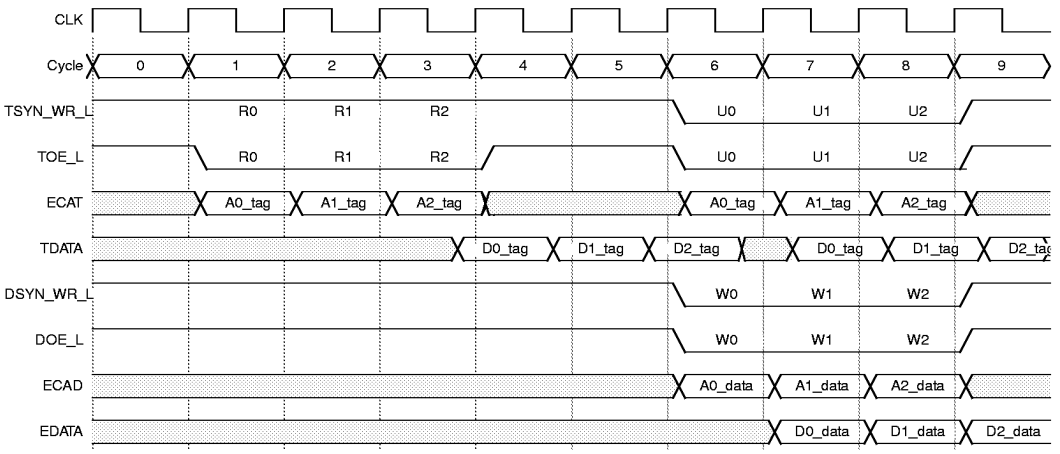


Figure 7. Coherent Writes with E to M Updates

Figure 8 shows the overlap of tag and data accesses. The data for three previous writes (W0, W1 and W2) is written while three tag accesses (reads) are made for three younger stores (R3, R4 and R5). If the line is in Shared or Owned state, then a read for ownership is performed before writing the data. If the access is a miss then a line is victimized and the data is written after the new line is brought in.

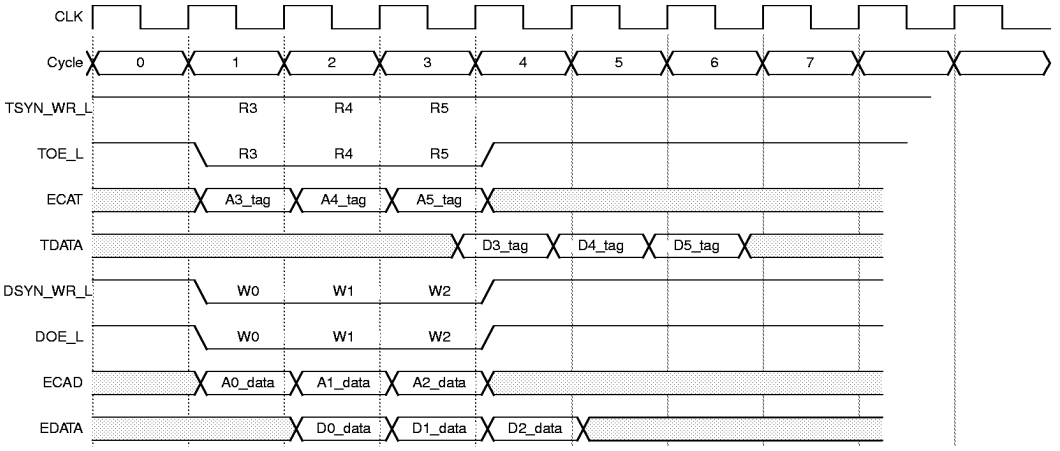


Figure 8. Overlap Between Tag Access and Data Write for Coherent Writes

Coherent Read Followed by a Coherent Write in 1-1-1 Mode

When a read is made to the E-Cache, the three-cycle latency causes the data bus to be busy two cycles after the address appears at the pins. For a processor without delayed writes, writes have to be held for two cycles in order to avoid collisions between the data of the write and the data coming back from the read. Additionally, an extra cycle is necessary to switch the driver of the E-Cache data bus from the SRAMs to UltraSPARC–II (due to electrical considerations). UltraSPARC–II uses a one-deep write buffer in the data SRAMs to reduce the turn around penalty to two cycles (going from reads to writes). The data of a write is sent one cycle after the address. Note that there is no penalty for going from writes to reads. *Figure 9* shows the two-cycle penalty between reads and writes. The figure represents three reads followed by two writes and two tag updates.The two cycle penalty applies to both tag accesses and data accesses (two cycles between A2_tag and A3_tag as well as between A2_data and A3_data).

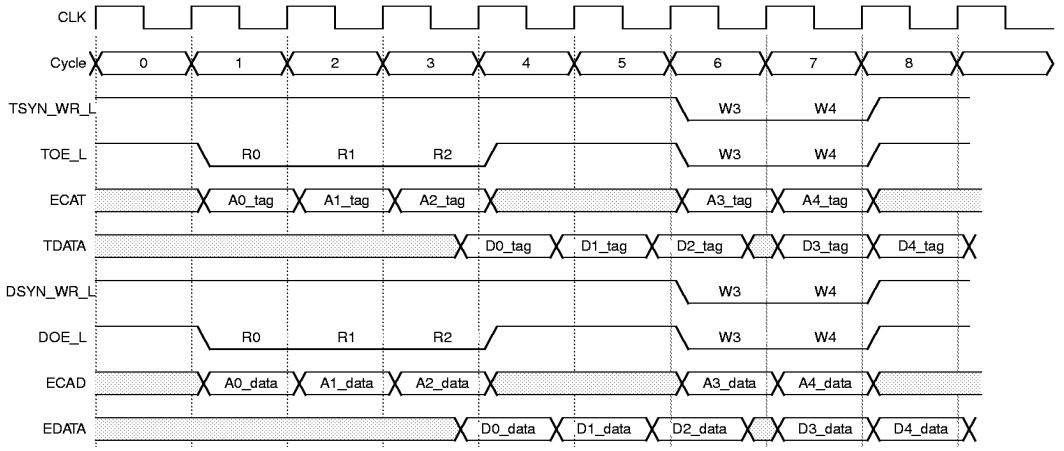


Figure 9. Reads Followed by Writes; Turn Around Penalty

Timing Diagrams for 2-2 Mode

UltraSPARC-II supports another mode of SRAM operation, called flow-through access, or the 2-2 mode. In this case the SRAMs are clocked at half the UltraSPARC-II frequency. The pin ecahe_22_mode is set to "1" in this mode. The examples on the following pages show transactions on the E-Cache bus, with the external cache operating in the 2-2 mode. (See Figure 10 on page 25, Figure 11 on page 25, and Figure 12 on page 26.)

Coherent Read Hit in 2-2 Mode

Three consecutive coherent reads are shown hitting the E-Cache in *Figure 10*. The control signal (TOE_L) and the address for the tag read (ECAT) are shown in transition shortly after the rising edge of the clock -- along with the control signal DOE_L and address for the data (ECAD). One cycle later, the data for both the tag read and data read is back at the pins of the CPU. The data arrives shortly before the next rising edge, which meets the set up time and clock skew requirements. The reads are fully pipelined and full throughput is achieved. Two requests are made before the data of the first request returns, and the latency of each request is two cycles.

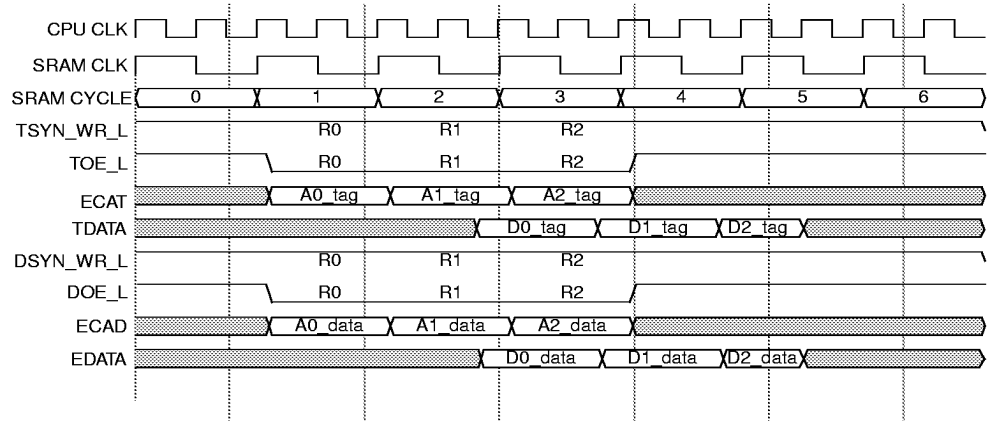


Figure 10. Timing for Coherent Read Hit (2-2 mode)

Coherent Write Hit to M State Line in 2-2 Mode

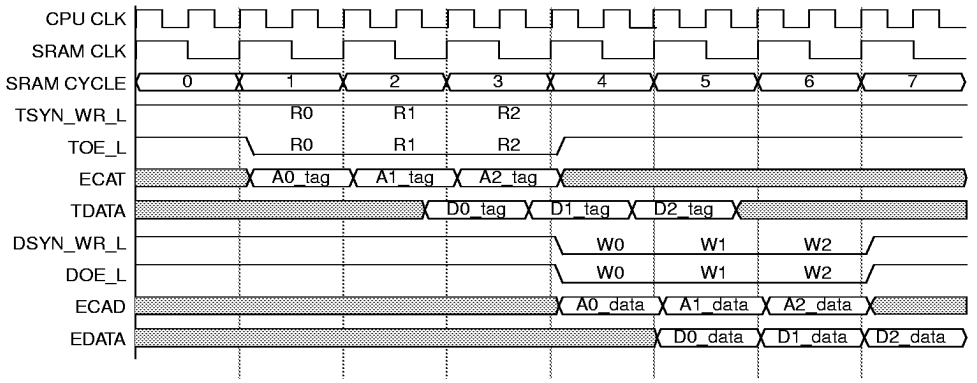


Figure 11. Timing for Coherent Write Hit to M State Line (2-2 mode)

In *Figure 11*, three consecutive write hits to the M state lines are diagramed in 2-2 mode. Access to the first tag (D0_tag) is started by asserting TOE_L and by sending the tag address (A0_tag). In the cycle after the tag data (D0_tag) comes back, UltraSPARC-II determines that the access is a hit and that the line is in Modified (M) state. In the next clock, a request is made to write the data. The data address is presented on the ECAD pins in the cycle after the request (cycle 4 for W0) and the data is sent in the following cycle (cycle 5). Systems running in 2-2 mode incur no read-to-write bus turnaround penalty.

Read-to-Write Transition in 2-2 Mode

Figure 12 shows the read-to-write transition for 2-2 mode. The figure shows three reads followed by two writes and two tag updates. Since the tag and data appear in the same cycle as the write, there is no turn-around penalty for reads followed by writes in 2-2 mode.

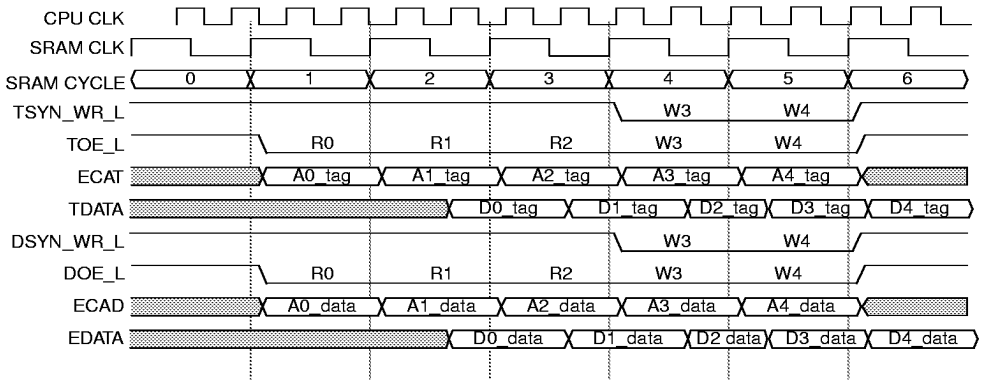


Figure 12. No Turn-Around Penalty: Read-to-Write Transition (2-2 mode)

ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings[1]

Symbol	Parameter	Rating	Units
VDD_C	Supply voltage range, core (includes VDD_{PLL} & VDD_{PECL})	0 to 3.2	V
VDD_O	Supply voltage range, peripherals	0 to 3.2	V
VDD_H	Supply voltage range (System Interface)	0 to 3.8	V
V_I	Input voltage range[2] (System Interface and JTAG)	-0.5 to VDD_H + 0.5	V
	Input voltage range (Except System Interface and JTAG)	-0.5 to VDD_C + 0.5	V
V_O	Output voltage range (System Interface and JTAG)	-0.5 to VDD_H + 0.5	V
	Output voltage range (Except System Interface and JTAG)	-0.5 to VDD_C + 0.5	V
I_{IK}	Input clamp current (V_I < 0 or V_I > VDD)	±20	mA
I_{OK}	Output clamp current (V_O < 0 or V_O > VDD)	±+/- 50	mA
I_{OL}	Current into any output in the low state	50	mA
T_{STG}	Storage temperature	-40 to 150	°C

1. Operation of the device at values in excess of those listed above will result in degradation or destruction of the device. All voltages are defined with respect to ground. Functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
2. Unless otherwise noted, all voltages are with respect at V_{SS}.

Recommended Operating Conditions

Symbol	Parameter		Min	Typ	Max	Units
VDD_C	Supply voltage, core (includes VDD_{PLL} & VDD_{PECL})		2.52	2.6	2.73	V
VDD_O	Supply voltage, peripheral		2.52	2.6	2.73	V
VDD_H	Supply voltage(UPA)		3.2	3.3	3.465	V
V_{SS}	Ground		–	0	–	V
V_{IH}	High-level input voltage	All except CLK &UPA	2.0	–	VDD_C + 0.3	V
		UPA	2.4	–	VDD_H + 0.3	V
V_{IL}	Low-level input voltage	All except CLK & UPA	-0.3	–	0.8	V
		UPA	-0.3	–	0.8	V
V_{CM}	DC Common Mode Input Range	CLK	VDD_{PECL} - 0.85	VDD_{PECL} - 0.5	VDD_{PECL} - 0.15	V
V_{IN}	Minimum Input Signal Amplitude	CLK	+/- 0.155	-	-	V
I_{OH}	High-level output current		–	–	-4.0	mA
I_{OL}	Low-level output current		–	–	8.0	mA
T_J	Operating junction temperature		–	–	105	°C
T_A	Operating ambient temperature		0	–	[1]	°C

1. Maximum ambient temperature is limited by air flow such that the maximum junction temperature does not exceed T_J.

DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OH}	High-level output voltage (except UPA)	VDD_O = Min, I_{OH} = Max	2.0	–	–	V
V_{OH}	High-level output voltage (UPA)	VDD_H = Min, I_{OH} = Max	2.4	–	–	V
V_{OL}	Low-level output voltage [1](except UPA)	VDD_O = Min, I_{OL} = Max	–	–	0.6	V
V_{OL}	Low-level output voltage [2](UPA)	VDD_H = Min, I_{OL} = Max	–	–	0.4	V
V_{IH}	High-level input voltage (except CLKA, CLKB, UDBCLKA, UDBCLKB, UPA)	VDD_C = Max	1.65	–	–	V
	High -level input voltage (CLKA, CLKB, UDBCLKA, UDBCLKB)	VDD_C = Max	2.135	–	2.42	V
	High -level input voltage (UPA)	VDD_H = Max	2.0	–	–	V
V_{IL}	Low-level input voltage (except CLKA, CLKB, UDBCLKA, UDBCLKB, UPA)	VDD_C = Min	–	–	1.15	V
	Low-level input voltage (CLKA, CLKB, UDBCLKA, UDBCLKB)	VDD_C = Min	1.49	–	1.825	V
	Low-level input voltage (UPA)	VDD_H = Min	–	–	0.8	V
$IDD_C + IDD_O + IDD_{PECL}$	Supply current	VDD_C, VDD_O & VDD_{PLL} = Max, frequency = 250 MHz	–	–	9.4	A
IDD_H		VDD_H = Max, frequency = 250 MHz	–	–	0.5	A
IDD_{PLL}		VDD_{PLL} = Max, frequency = 250 MHz	–	–	0.1	A
I_{OZ}	High-impedance output current [3]	VDD_O & VDD_H= Max, V_O = VDD	-30	–	30	uA
		VDD_O& VDD_H = Max, V_O = V_{SS}	-30	–	30	uA
I_I	Input current (Input-Only Pins)	VDD_C & VDD_H = Max, V_O = V_{SS} to VDD	-10	–	10	uA
C_I	Input capacitance [4] (Input-Only Pins)		–	5	–	pF
C_O	Output capacitance [4]		–	10	–	pF

1. STOP_CLK has no V_{OL} specification.
2. STOP_CLK has no V_{OL} specification.
3. Only bidirectional lines can be three-state. Output-only are not three-state. All bidirectional lines will be three-stated when $\overline{\text{RESET\_L}}$ is held LOW and SRAM_TEST is held high.
4. This specification is provided as an aid to board design. This specification is not assured during manufacturing testing.

AC Characteristics - Signal Timing (Except Clock and JTAG)[1]

Symbol	Parameter	Signals	Conditions	250 MHz		Units
				Min	Max	
t_{SU} (1)	Input setup time to CLK	SYSADR[35:0], ADR_VLD, SCLK_MODE, NODE_RQ[2:0], SC_RQ, S_REPLY[3:0], DATA_STALL, EDATA[127:0], EDPAR[15:0], TDATA[24:0], TPAR[3:0], RESET_L[2], XIR_L, UDB_UEH,[3] UDB_UEL, UDB_CEH, UDB_CEL ECACHE_22_MODE		1.38	–	ns
t_H(1)	Input hold time to CLK			1.07	–	ns
t_{PD}(1)	Output delay from CLK	EDATA[127:0], EDPAR[15:0], TDATA[24:0], TPAR[3:0], BYTEWE, ECAD[19:0], ECAT[17:0], DSYN_WR, DOE, TSYN_WR, TOE	I_{OL} = 8 mA I_{OH} = -4 mA C_L = 35 pF V_{LOAD} = 1.5V	–	1.40	ns
t_{OH}(1)	Output hold time from CLK			-0.37	–	ns
t_{PD}(1)	Output delay from CLK	UPA pins		–	1.66	ns
t_{OH}(1)	Output hold time from CLK	UPA pins		-0.37	–	ns
t_{PD}(2)	Output delay from CLK	STOP_CLK, EPD		–	4.0	ns
t_{OH}(2)	Output hold time from CLK	EPD		0.2	–	ns
t_{LOCK}	PLL acquisition time			10		us
				2500		cycle
t_{SKEW}	UDBCLK skew to CLK (see *Figure 17*)					ns

1. All timing requirements are specified with PLL enabled. Internal PLL skew can cause outputs to change before the clock edge at a device pin.
2. RESET is asserted asynchronously but deasserted synchronously to the system clock.
3. "UDB" was referred to as "SDB" in earlier documentation.

AC Characteristics - Clock Timing

Symbol	Parameter	250 MHz			Units
		Min	Typ	Max	
t_{CYC} (CLK)	Processor clock cycle time [1]	4	–	–	ns
t_W(CLK)	Processor clock duty cycle [1]	40	50	60	%
t_{SLEW}(CLK)	Clock input slew rate [1]	–	–	1.0	V/ns
t_{CYC} (TCK)	TCK clock cycle time	20	–	100	ns
t_W(TCK)	TCK clock duty cycle	40	50	60	%
t_{CYC} (UDBCLK)	UDBCLK [2] clock cycle time (DIVIDE BY 2 MODE)	t_{CYC} (CLK) x2	–	–	ns
t_{CYC} (UDBCLK)	UDBCLK clock cycle time (DIVIDE BY 3 MODE)	t_{CYC} (CLK) x3	–	–	ns
t_{CYC} (UDBCLK)	UDBCLK clock cycle time (DIVIDE BY 4 MODE)	t_{CYC} (CLK) x4	–	–	ns
t_W (UDBCLK)	UDBCLK duty cycle	33	50	66	%
t_W (RESET_L)	RESET_L pulse width LOCK MODE (See *Figure 18*)	10	–	–	ns
t_W (RESET_L)	RESET_L pulse width BYPASS MODE (See *Figure 18*)	10	–	–	ns

1. This is for the PLL enabled.
2. "UDB" was referred to as "SDB" in earlier documentation.

AC Characteristics - JTAG Timing

Symbol	Parameter	Signals	Conditions	250 MHz			Units
				Min	Typ	Max	
t_W(TRST_L)	Test reset pulse width	TRST_L [1]		5	–	–	ns
t_{SU}(TDI)	Input setup time to TCK	TDI		–	3	–	ns
t_{SU}(TMS)	Input setup time to TCK	TMS		–	4	–	ns
t_H(TDI)	Input hold time to TCK	TDI		–	1.5	–	ns
t_H(TMS)	Input hold time to TCK	TMS		–	1.5	–	ns
t_{PD}(TDO)	Output delay from TCK [2]	TDO	I_{OL} = 8 mA, I_{OH} = -4 mA C_L = 35 pF, V_{LOAD} = 1.5V	–	6	–	ns
t_{OH}(TDO)	Output hold time from TCK [2]	TDO		–	–	3	ns

1. TRST_L is an asynchronous reset.
2. TDO is referenced from falling edge of TCK.

AC Characteristics - TPD (Output) Capacitive Derating Factor[1]

Symbol	Parameter	250 MHz			Units
		Min	Typ	Max	
t_{PD}	Capacitive derating factor	0.2	0.3	0.4	ns/10pF

1. Derating factors are shown to aid in board design. This specification is not verified during manufacturing testing.

Thermal Resistance vs. Air Flow[1]

Symbol	Air Flow (ft/min)				Units
	100	200	300	500	
$Theta_{JA}$	1.5	1.2	1.0	0.85	(°C/W)

1. T_J can be calculated by: $T_J = T_A + P_D \times Theta_{JA}$.
Thermal resistance measured using UltraSPARC-II heatsink. P_D = Power Dissipation.

Power Consumption

At 250 MHz, the UltraSPARC-II consumes a maximum of 26W of power.

PARAMETER MEASUREMENT

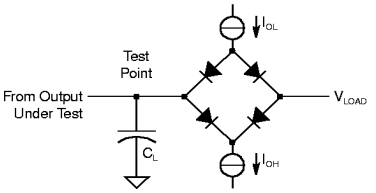


Figure 13. Load Circuit

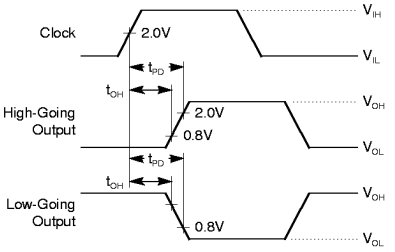


Figure 14. Voltage Waveforms - Propagation Delay Times

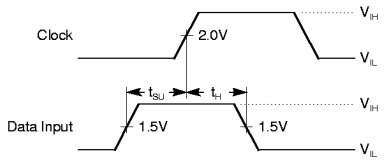


Figure 15. Voltage Waveforms - Setup and Hold Times

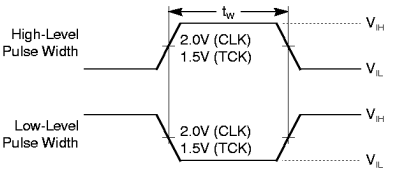


Figure 16. Voltage Waveforms - Clock Pulse Duration

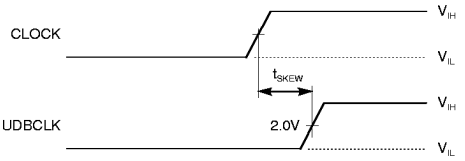


Figure 17. Voltage Waveforms - Clock Skew

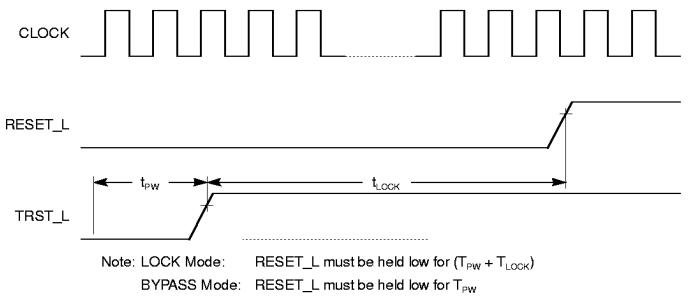


Figure 18. Reset Timing

PIN ASSIGNMENTS

Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name
A2	VSSO	B22	VSSO	D10	VSSC	E30	BYTEWE_L[13]	G18	VSSO
A3	VDDO	B23	VSSO	D11	VSSC	E31	EDATA[4]	G19	VSSO
A4	VDDO	B24	ECAT[1]	D12	TDATA[23]	E32	VSSC	G20	ECAT[8]
A5	VSSC	B25	ECAT[2]	D13	TDATA[24]	F1	VDDO	G21	ECAT[14]
A6	DSYN_WR_L	B26	VDDO	D14	VDDC	F2	VDDO	G22	VDDO
A7	ECAD[0]	B27	VDDO	D15	VDDC	F3	VSSC	G23	VDDO
A8	ECAD[3]	B28	BYTEWE_L[5]	D16	NOCONN	F4	VDDC	G24	ECAT[16]
A9	ECAD[4]	B29	VDDO	D17	NOCONN	F5	VSSO	G25	BYTEWE_L[11]
A10	VSSO	B30	VDDC	D18	VSSC	F6	VDDO	G26	NOCONN
A11	VSSO	B31	VDDO	D19	VSSC	F7	VSSO	G27	VSSO
A12	ECAD[6]	B32	VDDO	D20	ECAT[10]	F8	TDATA[8]	G28	VDDO
A13	ECAD[5]	C1	VDDO	D21	ECAT[15]	F9	TDATA[10]	G29	VSSC
A14	VDDO	C2	VSSO	D22	VSSC	F10	VDDO	G30	VDDC
A15	VDDO	C3	VDDO	D23	VSSC	F11	VDDO	G31	VSSO
A16	ECAD[12]	C4	VDDC	D24	BYTEWE_L[1]	F12	TDATA[14]	G32	EDATA[7]
A17	ECAD[17]	C5	TDATA[12]	D25	ECAT[12]	F13	TDATA[18]	H1	SYSADR[31]
A18	VSSO	C6	VDDO	D26	VDDC	F14	VSSO	H2	P_REPLY[2]
A19	VSSO	C7	VDDO	D27	VDDC	F15	VSSO	H3	SYSADR[34]
A20	UDB_CNTL[0]	C8	PMO	D28	VSSO	F16	ECAD[10]	H4	P_REPLY[4]
A21	ECAD[19]	C9	TPAR[3]	D29	VDDO	F17	ECAD[16]	H5	TDATA[0]
A22	VDDO	C10	VSSO	D30	VSSO	F18	VDDO	H6	TDATA[2]
A23	VDDO	C11	VSSO	D31	VDDO	F19	VDDO	H7	TPAR[0]
A24	UDB_CNTL[1]	C12	ECACHE_22_MODE	D32	VSSC	F20	ECAT[9]	H8	NOCONN
A25	UDB_CNTL[2]	C13	TOE_L	E1	VSSC	F21	ECAT[17]	H9	TDATA[6]
A26	ECAT[0]	C14	VDDO	E2	TDATA[1]	F22	VSSO	H10	VDDO
A27	VDDO	C15	VDDO	E3	TDATA[3]	F23	VSSO	H11	VDDO
A28	BYTEWE_L[0]	C16	ECAD[9]	E4	TDATA[4]	F24	BYTEWE_L[4]	H12	TDATA[15]
A29	VSSC	C17	ECAD[15]	E5	NOCONN	F25	BYTEWE_L[9]	H13	TDATA[20]
A30	VDDO	C18	VSSO	E6	VSSO	F26	VDDO	H14	VSSO
A31	VSSO	C19	VSSO	E7	VSSO	F27	VSSO	H15	VSSO
A32	VSSO	C20	ECAT[5]	E8	TPAR[1]	F28	VDDO	H16	ECAD[7]
B1	VSSO	C21	ECAT[3]	E9	TDATA[13]	F29	VSSC	H17	ECAD[13]
B2	VDDO	C22	VDDO	E10	VDDC	F30	VDDC	H18	VDDO
B3	VSSO	C23	VDDO	E11	VDDC	F31	VSSO	H19	VDDO
B4	VDDC	C24	ECAT[7]	E12	TDATA[17]	F32	EDPAR[0]	H20	ECAT[6]
B5	TDATA[11]	C25	ECAT[4]	E13	TDATA[19]	G1	P_REPLY[1]	H21	ECAT[13]
B6	VSSO	C26	VSSO	E14	VDDC	G2	VDDO	H22	VSSO
B7	VSSO	C27	VSSO	E15	VDDC	G3	VSSC	H23	VSSO
B8	DOE_L	C28	BYTEWE_L[12]	E16	NOCONN	G4	VDDC	H24	BYTEWE_L[8]
B9	TSYN_WR_L	C29	VDDC	E17	NOCONN	G5	VSSO	H25	BYTEWE_L[3]
B10	VDDO	C30	VDDO	E18	VSSC	G6	VDDO	H26	BYTEWE_L[14]
B11	VDDO	C31	VSSO	E19	VSSC	G7	NOCONN	H27	BYTEWE_L[15]
B12	ECAD[2]	C32	VDDO	E20	ECAT[11]	G8	TDATA[7]	H28	EDATA[3]
B13	ECAD[1]	D1	VDDO	E21	BYTEWE_L[2]	G9	TDATA[9]	H29	EDATA[2]
B14	VSSO	D2	VSSO	E22	VDDC	G10	VSSO	H30	EDATA[35]
B15	VSSO	D3	VSSO	E23	VDDC	G11	VSSO	H31	EDATA[37]
B16	ECAD[11]	D4	VDDO	E24	BYTEWE_L[7]	G12	TDATA[16]	H32	EDATA[38]
B17	ECAD[18]	D5	TDATA[5]	E25	BYTEWE_L[10]	G13	TPAR[2]	J1	SYSADR[28]
B18	VDDO	D6	VDDC	E26	VSSO	G14	VDDO	J2	SYSADR[32]
B19	VDDO	D7	VDDC	E27	VDDO	G15	VDDO	J3	SYSADR[30]
B20	UDB_CNTL[4]	D8	TDATA[21]	E28	VSSO	G16	ECAD[8]	J4	SYSADR[33]
B21	UDB_CNTL[3]	D9	TDATA[22]	E29	BYTEWE_L[6]	G17	ECAD[14]	J5	ADR_VLD

PIN ASSIGNMENTS (CONTINUED)

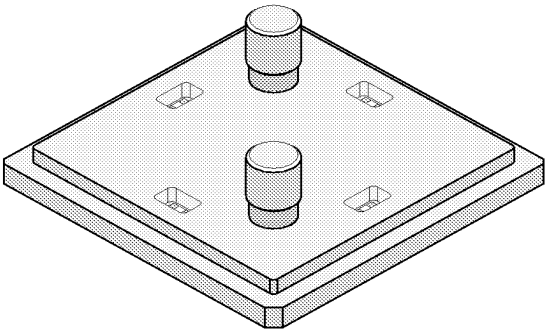
Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name
J6	SYSADR[35]	M26	EDATA[33]	R26	VDDO	V26	VSSO	AA26	EDATA[78]
J7	P_REPLY[3]	M27	EDATA[66]	R27	VSSO	V27	VDDO	AA27	EDATA[106]
J8	P_REPLY[0]	M28	EDATA[65]	R28	VDDC	V28	VSSC	AA28	EDATA[74]
J25	EDATA[5]	M29	EDATA[70]	R29	VDDC	V29	VSSC	AA29	EDATA[46]
J26	EDATA[6]	M30	EDATA[96]	R30	VDDO	V30	VSSO	AA30	EDATA[47]
J27	EDATA[1]	M31	EDATA[98]	R31	VSSO	V31	VDDO	AA31	EDATA[42]
J28	EDATA[39]	M32	EDPAR[8]	R32	VDDO	V32	VSSO	AA32	EDATA[43]
J29	EDATA[0]	N1	RESET_L	T1	S_REPLY[3]	W1	VDDH	AB1	VSSO
J30	EDATA[36]	N2	PLLBYPASS	T2	UDBCLKA	W2	VSSO	AB2	VDDH
J31	EDATA[32]	N3	MCAP[3]	T3	UDBCLKB	W3	VDDH	AB3	VSSO
J32	EDATA[64]	N4	PHASE_DET_CLK	T4	NOCONN	W14	VSSC	AB4	VDDC
K1	VDDH	N5	CLKA	T5	NOCONN	W20	VDDC	AB5	VSSC
K2	VSSO	N6	CLKB	T6	SYSADR[19]	W25	VDDO	AB6	VDDH
K3	VDDH	N7	SYSADR[22]	T7	S_REPLY[2]	W26	VSSO	AB7	VSSO
K4	VDDC	N8	SYSADR[26]	T8	PECL_VDD	W27	VDDO	AB8	VDDH
K5	VSSC	N14	VDDC	T14	VDDC	W28	VSSC	AB25	VSSO
K6	VSSO	N20	VSSC	T20	VSSC	W29	VSSC	AB26	VDDO
K7	VDDH	N25	EDATA[34]	T25	EDATA[10]	W30	VSSO	AB27	VSSO
K8	VSSO	N26	EDATA[67]	T26	EDATA[8]	W31	VDDO	AB28	VSSC
K25	VDDO	N27	EDATA[69]	T27	EDATA[9]	W32	VSSO	AB29	VDDC
K26	VSSO	N28	EDATA[68]	T28	NOCONN	W4	VDDC	AB30	VDDO
K27	VDDO	N29	EDATA[97]	T29	NOCONN	W5	VDDC	AB31	VSSO
K28	VSSC	N30	EDATA[71]	T30	EDATA[103]	W6	VSSO	AB32	VDDO
K29	VDDC	N31	EDPAR[12]	T31	EDATA[101]	W7	VDDH	AC1	VSSO
K30	VSSO	N32	EDATA[100]	T32	EDATA[99]	W8	VSSO	AC2	VDDH
K31	VDDO	P1	VSSO	U1	SYSADR[16]	Y1	SYSADR[14]	AC3	VSSO
K32	VSSO	P2	VDDH	U2	S_REPLY[1]	Y2	SYSADR[11]	AC4	VDDC
L1	VDDH	P3	VSSO	U3	SYSADR[15]	Y3	SYSADR[13]	AC5	VSSC
L2	VSSO	P4	VSSC	U4	NOCONN	Y4	SYSADR[12]	AC6	VDDH
L3	VDDH	P5	VSSC	U5	NOCONN	Y5	SYSADR[10]	AC7	VSSO
L4	VDDC	P6	VDDH	U6	SYSADR[18]	Y6	SYSADR[7]	AC8	VDDH
L5	VSSC	P7	VSSO	U7	S_REPLY[0]	Y7	SYSADR[8]	AC25	VSSO
L6	VSSO	P8	VDD_PLL	U8	SYSADR[17]	Y8	SYSADR[2]	AC26	VDDO
L7	VDDH	P14	VSSC	U14	VDDC	Y14	VDDC	AC27	VSSO
L8	VSSO	P20	VDDC	U20	VSSC	Y20	VSSC	AC28	VSSC
L25	VDDO	P25	VSSO	U25	EDATA[14]	Y25	EDPAR[9]	AC29	VDDC
L26	VSSO	P26	VDDO	U26	EDATA[15]	Y26	EDATA[77]	AC30	VDDO
L27	VDDO	P27	VSSO	U27	EDATA[13]	Y27	EDATA[72]	AC31	VSSO
L28	VSSC	P28	VDDC	U28	NOCONN	Y28	EDATA[44]	AC32	VDDO
L29	VDDC	P29	VDDC	U29	NOCONN	Y29	EDPAR[5]	AD1	DATA_STALL
L30	VSSO	P30	VDDO	U30	EDPAR[1]	Y30	EDATA[40]	AD2	NODE_RQ0
L31	VDDO	P31	VSSO	U31	EDATA[12]	Y31	EDATA[41]	AD3	NODE_RQ2
L32	VSSO	P32	VDDO	U32	EDATA[102]	Y32	EDATA[11]	AD4	STOP_CLOCK
M1	MCAP[2]	R1	VSSO	V1	VDDH	AA1	SYSADR[0]]	AD5	XIR_L
M2	SYSADR[20]	R2	VDDH	V2	VSSO	AA2	SYSADR[9]	AD6	UDB_UEH
M3	SYSADR[21]	R3	VSSO	V3	VDDH	AA3	SYSADR[6]	AD7	UDB_CEL
M4	SYSADR[23]	R4	VSSC	V4	VDDC	AA4	SYSADR[5]	AD8	MISC_BIDIR[13]
M5	SYSADR[24]	R5	VSSC	V5	VDDC	AA5	SYSADR[4]	AD25	EDATA[110]
M6	SYSADR[25]	R6	VDDH	V6	VSSO	AA6	SYSADR[3]	AD26	EDATA[20]
M7	SYSADR[27]	R7	VSSO	V7	VDDH	AA7	SYSADR[1]	AD27	EDPAR[2]
M8	SYSADR[29]	R8	VSS_PLL	V8	VSSO	AA8	NODE_RQ1	AD28	EDATA[107]
M14	VDDC	R14	VSSC	V14	VSSC	AA14	VDDC	AD29	EDATA[105]
M20	VSSC	R20	VDDC	V20	VDDC	AA20	VSSC	AD30	EDATA[76]
M25	EDPAR[4]	R25	VSSO	V25	VDDO	AA25	EDATA[79]	AD31	EDATA[73]

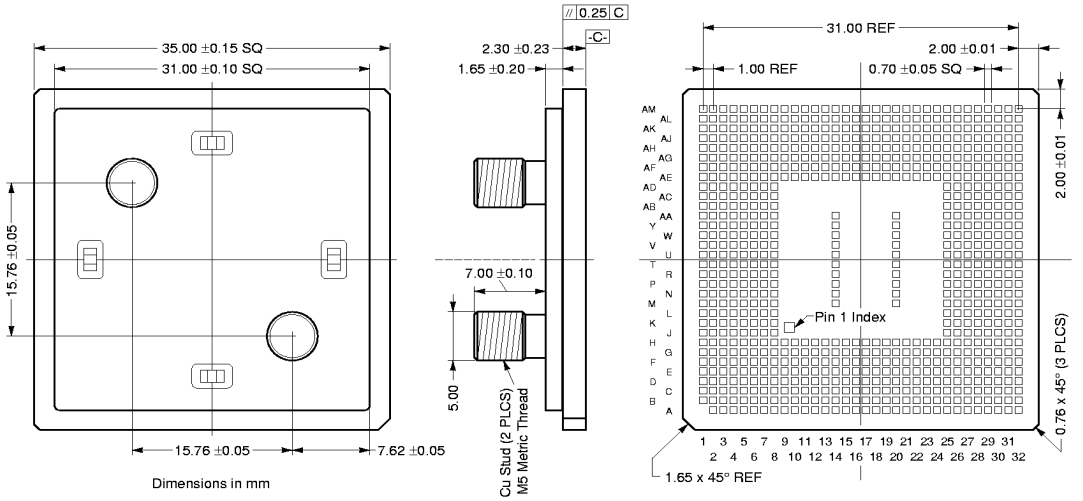
PIN ASSIGNMENTS (CONTINUED)

Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name
AD32	EDATA[45]	AF19	VDDO	AH6	VSSC	AJ25	EDATA[29]	AL13	EDATA[115]
AE1	VSSC	AF20	EDATA[60]	AH7	VSSC	AJ26	VDDC	AL14	VDDO
AE2	NODEX_RQ	AF21	EDATA[59]	AH8	MISC_BIDIR[6]	AJ27	VDDC	AL15	VDDO
AE3	SC_RQ	AF22	VSSO	AH9	MISC_BIDIR[3]	AJ28	EDATA[18]	AL16	EDATA[91]
AE4	TEMP_SENSE0	AF23	VSSO	AH10	VDDC	AJ29	VDDO	AL17	EDATA[88]
AE5	UDB_CEH	AF24	EDATA[28]	AH11	VDDC	AJ30	VSSO	AL18	VSSO
AE6	UDB_UEL	AF25	EDATA[31]	AH12	EDATA[121]	AJ31	VDDO	AL19	VSSO
AE7	TEMP_SENSE1	AF26	NOCONN	AH13	EDATA[120]	AJ32	VDDO	AL20	EDATA[86]
AE8	MISC_BIDIR[14]	AF27	VDDO	AH14	VSSC	AK1	VDDO	AL21	EDATA[85]
AE9	IO_SPARE_OUT	AF28	VSSO	AH15	VSSC	AK2	VDDC	AL22	VDDO
AE10	VSSO	AF29	VDDC	AH16	NOCONN	AK3	VSSO	AL23	VDDO
AE11	VSSO	AF30	VSSC	AH17	NOCONN	AK4	VDDO	AL24	EDATA[56]
AE12	MISC_BIDIR[2]	AF31	VDDO	AH18	VDDC	AK5	MISC_BIDIR[7]	AL25	EDATA[52]
AE13	MISC_BIDIR[0]	AF32	VSSO	AH19	VDDC	AK6	VSSO	AL26	VSSO
AE14	VDDO	AG1	L5CLK	AH20	EDATA[80]	AK7	VSSO	AL27	VSSO
AE15	VDDO	AG2	VSSO	AH21	EDATA[61]	AK8	MISC_BIDIR[8]	AL28	EDPAR[3]
AE16	EDATA[112]	AG3	VDDO	AH22	VDDC	AK9	EPD	AL29	VDDC
AE17	EDATA[92]	AG4	VSSC	AH23	VDDC	AK10	VDDO	AL30	VSSO
AE18	VSSO	AG5	VDDO	AH24	EDATA[30]	AK11	VDDO	AL31	VDDO
AE19	VSSO	AG6	VDDO	AH25	EDATA[27]	AK12	EDATA[119]	AL32	VSSO
AE20	EDATA[57]	AG7	VDDO	AH26	VDDO	AK13	EDATA[116]	AM1	VSSO
AE21	EDATA[55]	AG8	MISC_BIDIR[5]	AH27	VDDO	AK14	VSSO	AM2	VDDO
AE22	VDDO	AG9	TCK	AH28	EDATA[17]	AK15	VSSO	AM3	VDDO
AE23	VDDO	AG10	VSSO	AH29	EDATA[111]	AK16	EDATA[95]	AM4	VSSO
AE24	EDATA[48]	AG11	VSSO	AH30	EDATA[22]	AK17	EDATA[89]	AM5	VSSC
AE25	EDATA[23]	AG12	EDATA[126]	AH31	EDATA[109]	AK18	VDDO	AM6	TRST_L
AE26	EDATA[21]	AG13	EDATA[123]	AH32	VSSC	AK19	VDDO	AM7	MCAP[0]
AE27	EDATA[19]	AG14	VDDO	AJ1	VSSO	AK20	EDATA[84]	AM8	EDATA[127]
AE28	EDATA[16]	AG15	VDDO	AJ2	VSSO	AK21	EDATA[82]	AM9	EDPAR[15]
AE29	EDPAR[13]	AG16	EDATA[94]	AJ3	VDDC	AK22	VSSO	AM10	VDDO
AE30	EDATA[108]	AG17	EDPAR[11]	AJ4	VSSO	AK23	VSSO	AM11	VDDO
AE31	EDATA[104]	AG18	VSSO	AJ5	MISC_BIDIR[10]	AK24	EDATA[53]	AM12	EDPAR[14]
AE32	EDATA[75]	AG19	VSSO	AJ6	VDDC	AK25	EDATA[49]	AM13	EDATA[114]
AF1	VDDO	AG20	EDATA[63]	AJ7	VDDC	AK26	VDDO	AM14	VSSO
AF2	VSSO	AG21	EDPAR[7]	AJ8	TDI	AK27	VDDO	AM15	VSSO
AF3	VDDO	AG22	VDDO	AJ9	MCAP[1]	AK28	EDATA[24]	AM16	EDATA[90]
AF4	VSSC	AG23	VDDO	AJ10	VSSC	AK29	VSSO	AM17	EDATA[87]
AF5	VDDO	AG24	EDATA[50]	AJ11	VSSC	AK30	VDDO	AM18	VDDO
AF6	VSSC	AG25	EDATA[26]	AJ12	EDATA[122]	AK31	VSSO	AM19	VDDO
AF7	NOCONN	AG26	VSSO	AJ13	EDATA[117]	AK32	VSSO	AM20	EDPAR[10]
AF8	MISC_BIDIR[9]	AG27	VDDO	AJ14	VSSC	AL1	VDDO	AM21	EDATA[81]
AF9	MISC_BIDIR[4]	AG28	VSSO	AJ15	VSSC	AL2	VSSO	AM22	VSSO
AF10	VDDO	AG29	VDDC	AJ16	NOCONN	AL3	VDDO	AM23	VSSO
AF11	VDDO	AG30	VSSC	AJ17	NOCONN	AL4	VSSO	AM24	EDATA[58]
AF12	MISC_BIDIR[1]	AG31	VDDO	AJ18	VDDC	AL5	TMS	AM25	EDATA[54]
AF13	EDATA[124]	AG32	VSSO	AJ19	VDDC	AL6	VDDO	AM26	EDPAR[6]
AF14	VSSO	AH1	VSSC	AJ20	EDATA[83]	AL7	VDDO	AM27	EDATA[25]
AF15	VSSO	AH2	MISC_BIDIR[11]	AJ21	EDATA[62]	AL8	TDO	AM28	VSSC
AF16	EDATA[113	AH3	EXT_EVENT	AJ22	VSSC	AL9	EDATA[125]	AM29	VDDC
AF17	EDATA[93]	AH4	RAM_TEST	AJ23	VSSC	AL10	VSSO	AM30	VSSO
AF18	VDDO	AH5	MISC_BIDIR[12]	AJ24	EDATA[51]	AL11	VSSO	AM31	VSSO
-	-	-	-	-	-	AL12	EDATA[118]	AM32	VDDO

PACKAGE DIMENSIONS

787-Pin LGA Package





ORDERING INFORMATION [1]

Part Number	Speed	Description
STP1031LGA	250 MHz	Second Generation SPARC v9 64-bit Microprocessor With VIS

1. Standard parts are not shipped with sockets or heat sinks. Contact Sun Microsystems for more information.

Document Part Number: 802-7430-03