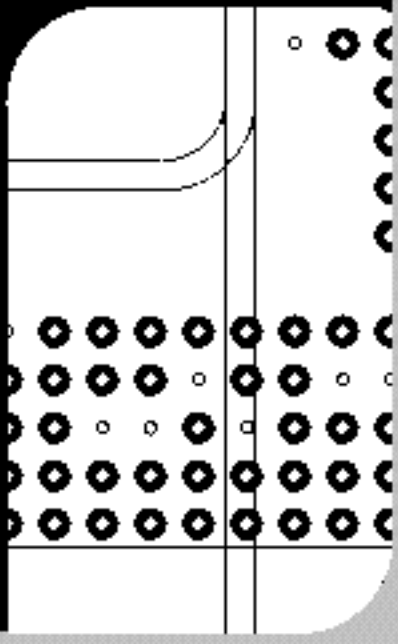
Sun Microelectronics



UltraSPARC™-I

High-Performance, 167 & 200 MHz, 64-bit RISC Processor

Data Sheet

October 1996

STP1030A

SUN MICROELECTRONICS

STP1030A

October 1996

UltraSPARC™-I

DATA SHEET

High-Performance, 167 & 200 MHz,
64-bit RISC Processor

DESCRIPTION

The STP1030A, UltraSPARC–I, is a high-performance, highly-integrated superscalar processor implementing the SPARC V9 64-bit RISC architecture. The STP1030A 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 (PDU) with an Instruction Buffer to feed the Execution Unit. 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. To further increase the number of instructions executed per cycle, instructions from different blocks (for instance, those before and after a conditional branch) can be issued in the same group.

The STP1030A supports 2D as well as 3D graphics, 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 two streams 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 Visual Instruction Set (VIS)
- 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
- Selectable Little- or Big-Endian Byte Ordering
- 64-Bit Address Pointers
- 16-Kilobyte Non-Blocking Data Cache
- 16-Kilobyte Instruction Cache
  - In-Cache 2-bit Branch Prediction
  - Single Cycle Branch Following
- Integrated Second Level Cache Controller
  - Supports 0.5 to 4 Megabyte Cache Sizes
  - Sustained throughput of 1 load/cycle
  - 2.6 Gigabyte/sec Processor-Cache Bandwidth (167 MHz)
  - 3.2 Gigabyte/sec Processor-Cache Bandwidth (200 MHz)
- Block Load/Store Instructions
  - 1.3 Gigabyte/sec Processor-Memory Bandwidth (167 MHz)
  - 1.6 Gigabyte/sec Processor-Memory Bandwidth (200 MHz)
- Ease of Use
  - JTAG Boundary Scan
  - Performance Instrumentation
- Technology/Packaging
  - 0.4um 4-Layer Metal CMOS Process
  - Operates at 3.3V
  - 521 Pin Plastic Ball Grid Array (BGA)
- Power Management

BLOCK DIAGRAM

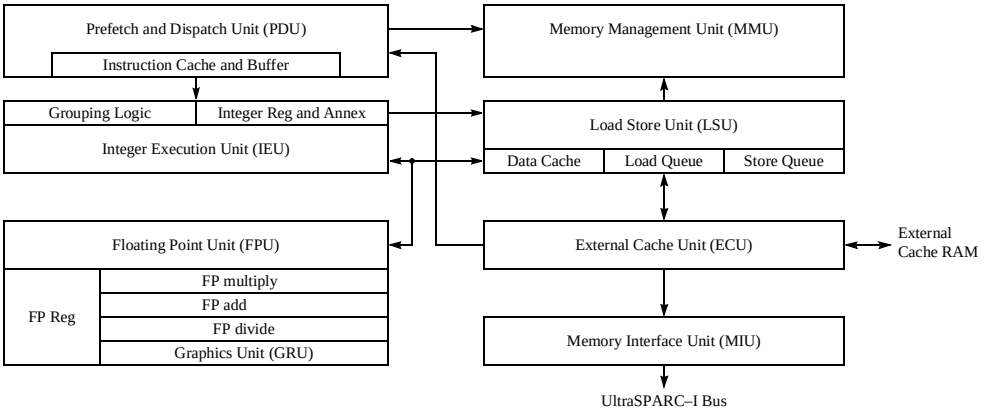


Figure 1. Functional Block Diagram

COMPONENT OVERVIEW

In a single chip implementation, UltraSPARC–I 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 (PDU)

The prefetch and dispatch unit fetches instructions 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 groups of 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–I provides hardware support for a software-based TLB miss strategy. A separate set of global registers is available to process an MMU trap. 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–I supports five levels of traps) are part of the IEU.

Load/Store Unit (LSU)

The LSU is responsible for generating the virtual address 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)

Separating the execution units in the FPU allows UltraSPARC–I 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) or 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–I 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 (E-Cache). Accesses to the external cache are pipelined, take three cycles (pin-to-pin) and return 16 bytes of instructions or data per cycle. This can effectively make the external cache a part of the pipeline. Programs with large data sets, data can keep data in the external cache and schedule instructions 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, or 1-, or 2-, or 4 Megabytes, but the line size is always 64 bytes. A MOESI protocol (“modified, owned, exclusive, shared, invalid”-- see "Cache Coherence Protocol" on page 11) helps to maintain coherency across the system.

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). The ECU also handles snoops.

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

Memory Interface Unit (MIU)

The MIU handles all transactions with the system, such as external cache misses, interrupts, snoops, writebacks, and so forth. The MIU communicates with the system at a frequency lower than that of UltraSPARC–I (either 1/2 or 1/3).

SIGNAL DESCRIPTIONS

Quick Pin Reference - UPA Interface

Symbol	Type	Name and Function
SYSADR[35:0]	I/O	Bidirectional UltraSPARC-I transaction request bus. Maximum of 3 other masters and 1 system controller can be connected to this bus. (3.3V, UPA)[1]
ADR_VLD	I/O	Bidirectional radial UltraSPARC-I Bus signal between UltraSPARC-I and the system. Driven by UltraSPARC-I to initiate SYSADR transactions to the system. Driven by the system to initiate Coherency, Interrupt or Slave transactions to UltraSPARC-I. Synchronous to the system clock. (3.3V, UPA)
NODE_RQ[2:0]	I	UltraSPARC-I system address bus arbitration request from up to 3 other UltraSPARC-I bus ports that might be sharing the SYSADR. Used by UltraSPARC-I for the distributed SYSADR arbitration protocol. Connection to other UltraSPARC-I bus ports is strictly dependent on the Master ID allocation. Synchronous to system clock. (3.3V, UPA)
SC_RQ	I	UltraSPARC-I 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-I system Reply packet, driven to UltraSPARC-I. Bit 4 of the UltraSPARC-I bus S_REPLY is not used by UltraSPARC-I. 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-I processor reply packet, driven by UltraSPARC-I to the system. Synchronous to system clock. (3.3V, UPA)
NODEX_RQ	O	UltraSPARC-I system address bus arbitration request. Asserted when UltraSPARC-I needs to drive SYSADR. Connected to all other UltraSPARC-I bus ports which share this address bus, and the system. Synchronous to system clock. (3.3V, UPA)

1. 3.3V UPA pins: SYSADR[35:0], ADR_VLD, NODE_RQ[2:0], SC_RQ, S_REPLY[[3:0], DATA_STALL, P_REPLY[4:0], NODEX_RQ, STOP_CLOCK, TDI, TCK, TMS, TRST, EXT_EVENT, RESET, XIR.

Quick Pin Reference - External Cache Interface

Symbol	Type	Name and Function
EDATA[127:0]	I/O	E-Cache data bus. Connects UltraSPARC–I to the E-Cache data SRAMs and the UDB. Synchronous to processor clock.
EDPAR[15:0]	I/O	Data bus parity. Odd parity is driven for all EDATA transfers, and checked if UltraSPARC–I or the UDB 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 greater 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[17:0]	O	Address for E-Cache data SRAMs. Corresponds to physical address [21:4]. Allows a maximum 4 Megabyte E-Cache. Synchronous to processor clock.
ECAT[15:0]	O	Address for E-Cache tag SRAMs. Corresponds to physical address [21:6]. 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.

Quick Pin Reference - Clock Interface

Symbol	Type	Name and Function
CLKA	I	This pin provides STP1030A with its primary clock source and is the positive differential clock input.
CLKB	I	This pin provides STP1030A with its primary clock source and is the negative differential clock input.
LOOPCAP	I	The external PLL loop filter connects to this pin to filter the analog voltage which controls the PLL VCO.
SCLK_MODE	I	Indicates clock divider mode - system frequency is /2 or /3 of the processor frequency
UDBCLKA, UDBCLKB	I	These are differential inputs of the system clock. They are used to generate the phase signal which allows UltraSPARC–I 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 UltraSPARC–I's internal level 5 clock. Used to determine PLL lock or clock tree delay when UltraSPARC–I is in PLL bypass mode.

Quick Pin Reference - JTAG/Debug Interface

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 is an open drain bidirectional signal used to indicate the clock should be stopped. This signal is wired-or with the other EXT_EVENT signals from other devices so that once one is activated all are activated. It is a debug signal which is set inactive on production systems.
PM_OUT	O	Used for on-chip process monitors (reserved 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 for IC manufacturing use).

Quick Pin Reference - Initialization Interface

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 resets. Actually acts like a non-maskable interrupt. Synchronous to system clock. Active low.
EPD	O	Asserted when UltraSPARC–I is in power-down mode.

Quick Pin Reference - UDB Chip Interface

Symbol	Type	Name and Function
UDB_UEH	I	Asserted when the High UDB 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 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 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 drives Edata[63:0], if the data has a corrected single-bit error. Synchronous to system clock.
UDB_CNTL[4:0]	O	UltraSPARC–I controls the UDB's drive and receive of EDATA. Asserted with valid EDATA when driving data to UDB. Asserted the cycle before the UDB should drive data. Synchronous to system clock.

ULTRASPARC–I SUBSYSTEM OPERATION

Subsystem Description

In the discussion which follows, “system” refers to any other location within the same coherency domain as UltraSPARC–I. For instance, the term includes caches of other processors connected to the interconnect.

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

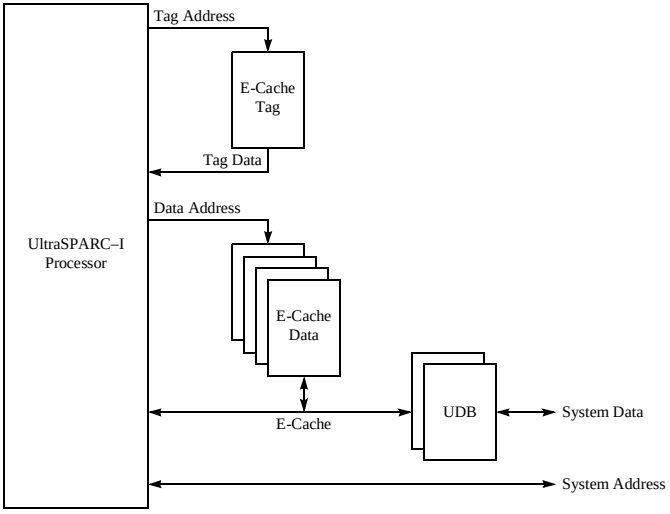


Figure 2. UltraSPARC–I System Interface

Overview of UltraSPARC–I Interface

The main interfaces to and from UltraSPARC–I are shown in *Figure 3*. A typical module includes an external cache composed of the tag unit and the data unit. Both of them 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 is to isolate UltraSPARC–I 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 overlap between system transactions and local E-Cache transactions even when the latter need to use part of the data buffer. The logic to control the UDB is included on UltraSPARC–I to provide fast data transfers between UltraSPARC–I and 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–I interfaces discussed here.

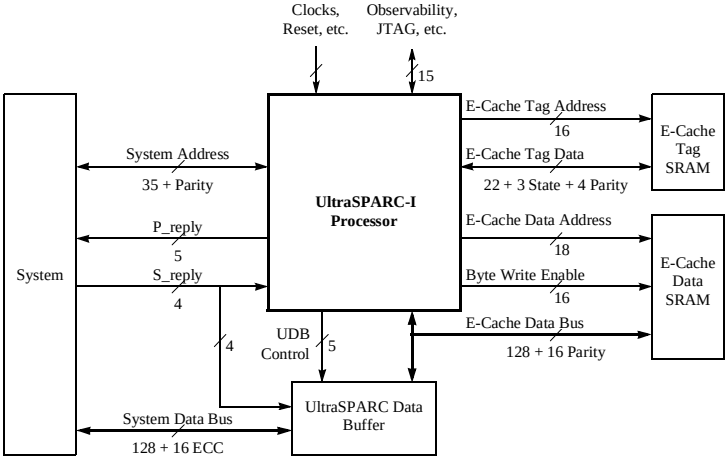


Figure 3. Main UltraSPARC–I Interfaces

Cache Coherence Protocol

This section describes the protocol used to maintain coherency between the internal caches of UltraSPARC–I, on the one hand, and the external cache and the system, on the other.

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–I). 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) write-back 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 64-byte blocks only, using 4 quadwords.

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

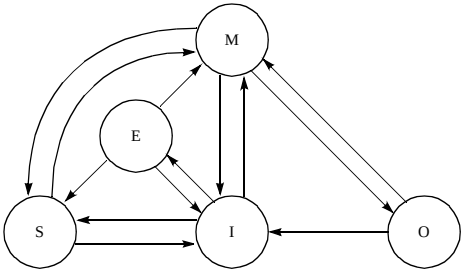


Figure 4. Cache Coherency Protocol State Diagram

Table 2 describes all the transitions between the states, as shown in *Figure 4*. It also shows the transactions that are initiated by either UltraSPARC–I 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/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 the 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
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–I as a Bus Port

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

The UltraSPARC–I processor and its cache subsystem (including the UDB) form an UltraSPARC–I bus module. This module interfaces to the interconnect using the UltraSPARC–I bus interface definition called the UltraSPARC Port Architecture (UPA). (For more information, refer to the manual entitled *UPA Interconnect Architecture.*) The I/O subsystem and the graphics subsystem may also reside on an UltraSPARC–I bus module.

The physical connections between UltraSPARC–I and the UltraSPARC–I bus mainly consist of the following:

- a bidirectional address bus for transaction requests between UltraSPARC–I and the UltraSPARC–I 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–I module, is present and is used to manage duplicate tags, and for efficient data sharing.

Table 3 shows the UltraSPARC–I Port interface as specified in the *UPA Interconnect Architecture* and the corresponding pins for UltraSPARC–I.

TABLE 3: UPA Port Interface

UPA Port Interface		UltraSPARC–I 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]	——>	

TABLE 3: UPA Port Interface (Continued)

UPA Port Interface		UltraSPARC–I Interface
UPA_Reset	——>	RESET
UPA_Sys_Clk[2]	——>	CLKA, CLKB
UPA_ClkCntl[4]	<——>	
UPA_JTAG[4]	——>	
UPA_Slave_INT	——>	
UPA_Mode	——>	
UPA_Wakeup_Reset	——>	

UltraSPARC–I is both a bus master and a bus slave. As a bus master UltraSPARC–I issues read/write transactions to the interconnect using part of the UltraSPARC-I bus transaction set. UltraSPARC–I 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–I has a physically addressed coherent cache (the E-Cache), which participates in the MOESI cache coherence protocol, and responds to the interconnect for copyback and invalidation requests.

As a bus slave, UltraSPARC–I responds to a noncached read of its bus port ID.

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

UPA Transactions Supported by UltraSPARC–I

Transactions Initiated by UltraSPARC–I

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

- P_RDS_REQ (Read To Share): Coherent read with intent to share; UltraSPARC–I issues this in response to a load miss.
- P_RDSA_REQ (Read to Share Always): Coherent read with intent to share “always;” UltraSPARC–I 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–I 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–I 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–I issues this during a block store.
- P_INT_REQ (Interrupt): Interrupt transaction request packet. UltraSPARC–I 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–I when a block write (64 bytes) is made to a noncacheable location.

System Transactions Accepted by UltraSPARC–I

- S_INV_REQ (Invalidate): Invalidate request from system controller to UltraSPARC–I following a Read To Own (P_RDO_REQ) or Write Invalidate (P_WRI_REQ) request for a block from another UltraSPARC–I.
- S_CPB_REQ (Copyback): Copyback request from the system controller to UltraSPARC–I following a Read To Share (P_RDS_REQ) or Read To Share Always (P_RDSA_REQ) request for a block from another UltraSPARC-I.
- S_CPI_REQ (Copyback Invalidate): Copyback and Invalidate request from the system controller to UltraSPARC–I in response to a Read To Own (P_RDO_REQ) request for a block from another UltraSPARC–I.
- S_CPD_REQ (Copyback To Discard): Sent at the UltraSPARC–I bus interface to UltraSPARC–I in order to service a Read To Discard (P_RDD_REQ) issued by another UltraSPARC–I. 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–I. UltraSPARC–I responds to this request by sending the value of its bus port ID to the data bus. The transaction starts as a P_NCRD_REQ from an UltraSPARC–I bus master, and is forwarded by the system controller to UltraSPARC–I. UltraSPARC–I replies through a P_RAS.

The system controller issues an S_SRS to drive the data on SYSDATA. Finally the requesting master gets the data when the system controller issues an S_RAS.

- P_INT_REQ (Interrupt): Interrupt transaction request packet sent by another UltraSPARC-I. In parallel, the 64-byte interrupt data packet is placed on SYSDATA, and S_SWIB is sent to instruct the UltraSPARC Data Buffer to accept the data.

Responses to Transactions Initiated by the System (P_Reply)

P_reply is an acknowledgment from UltraSPARC–I to the system, in response to a request that the system sent to UltraSPARC–I previously. There are five unidirectional (output only) pins on UltraSPARC–I 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–I 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. Sent following a single non-cacheable read request from an UltraSPARC–I (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 due to victimization. Indicates data is available in the UDB. (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. (Reply to S_CPB_REQ, S_CPD_REQ, S_CPI_REQ, or S_INV_REQ.)
- P_IAK (Interrupt Acknowledge): UltraSPARC–I 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 for another interrupt request and its 64 bytes of data.
- P_RERR (Read Error): Returned by UltraSPARC–I 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–I 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 Encoding

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

TABLE 4: P_REPLY Encoding (Continued)

Type	Cycles	Name	Reply to Which Transaction	Class	Type
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 Request (P_REQ) or Acknowledgment (P_REPLY) from UltraSPARC–I

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

- 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–I tried to access. Note that time-outs on writes are reported asynchronously via interrupt by the detecting slave UltraSPARC–I.
- S_ERR (Error): Asserted by the system if the situations described in the UltraSPARC–I bus specification occur. (See the manual on *UPA Interconnect Architecture.*)
- S_WAS (Write Acknowledge Single): Generated by the system following a noncacheable write request from UltraSPARC–I (P_NCWR_REQ). It causes 16 bytes of data from the UDB 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 Acknowledged Block): Generated by the system when UltraSPARC–I 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 the UDB 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 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 the UDB 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 the UDB to drive 16 bytes of data onto SYSDATA. This follows a P_RAS from UltraSPARC–I indicating the data to be ready in the UDB.
- S_CRAB (Copyback Read Block Acknowledge): The system commands the UDB to drive 64 bytes of copyback data onto SYSDATA. This follows a P_SACK or P_SACKD from UltraSPARC–I indicating the copyback data to be ready in the UDB.
- S_SWIB (Interrupt Write Block Acknowledge): The system commands the UDB to accept 64 bytes of interrupt data from SYSDATA. In parallel the P_INT_REQ packet that was initiated by the interrupting UltraSPARC–I is sent to the target UltraSPARC–I on SYSADDR.
- S_WBCAN (Writeback Cancel Acknowledge): The system generates this to cancel a previous writeback by UltraSPARC–I (P_WRB_REQ).
- S_INAK (Interrupt NACK): The system generates this if the receiver of an UltraSPARC–I interrupt request (P_INT_REQ) cannot accept another interrupt packet at the moment. This reply effectively removes the interrupt packet from the UDB 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 Encoding

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

TABLE 5: S_REPLY Encoding

S_REPLY	Name	Reply to Which Transaction	Type
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 Data Buffers (UDB)

External cache accesses, although synchronous to the internal clock, 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 consists of the load buffer, dual ported tags, separate address buses for tag and data, and so forth.

The UDB isolates the system data bus from UltraSPARC–I. The UDB allows data transfers between UltraSPARC–I and the memory system or I/O (for example, noncacheable stores). The data buffer also enables transfers between the E-Cache and the memory system (for instance, writebacks) to occur much more rapidly since system arbitration and system throughput are hidden by the internal buffering of the UDB. Overlapping of transactions is also possible which increases overall bandwidth. Interrupt “packets” are handled by the UDB, 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 SRAMs. The parts operate synchronously with UltraSPARC–I (and so are known as Synchronous Static RAMs). The external cache sizes supported by UltraSPARC–I are: 512 Kilobytes, and 1-, 2-, or 4 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–I and the UDB. More specifically for Ultra-SPARC-I, 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. So 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 UltraSPARC Data Buffer 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 UDB, controlled entirely by the CPU. Copybacks (responses to snoop hits) also generate transfers from the E-Cache to the UDB.

Each UltraSPARC data buffer microchip has a 4-entry-by-16-byte read buffer that can hold a 64-byte line coming from main memory due to an E-Cache read miss or a non-cacheable read. The outgoing buffer (that is, the buffer receiving data from the UltraSPARC–I and sending it to the system) is divided into three parts. There is an 8×16 byte writeback buffer, an 8×16 byte non-cacheable 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.

TIMING DIAGRAMS

This section describes the logical timing for the transactions occurring between UltraSPARC–I, 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–I. 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 ns is represented by showing the transition of the incoming signal changing slightly before the rising edge of the clock).

Coherent Read Hit

Coherent reads that hit the E-Cache are represented in *Figure 5*. With UltraSPARC–I 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.)

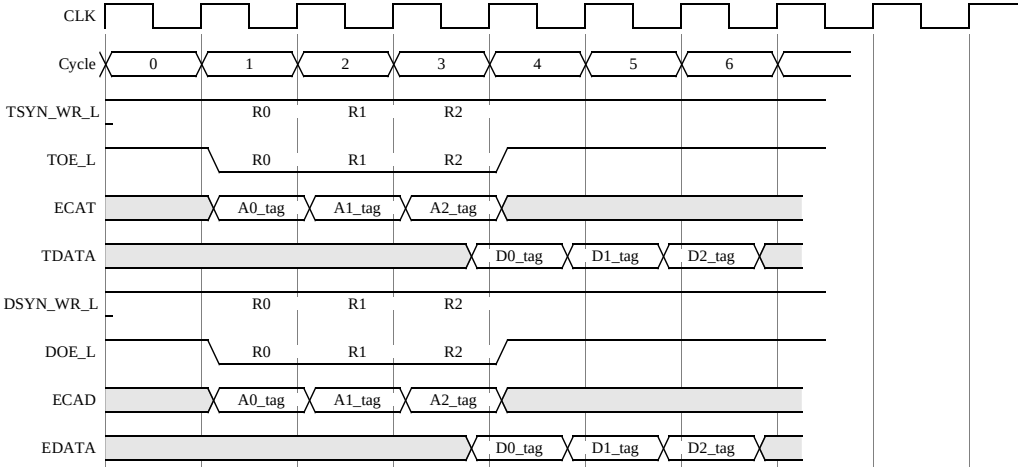


Figure 5. Coherent Read Hit Timing

Coherent Write Hits

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 shown 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–I 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" on page 26).

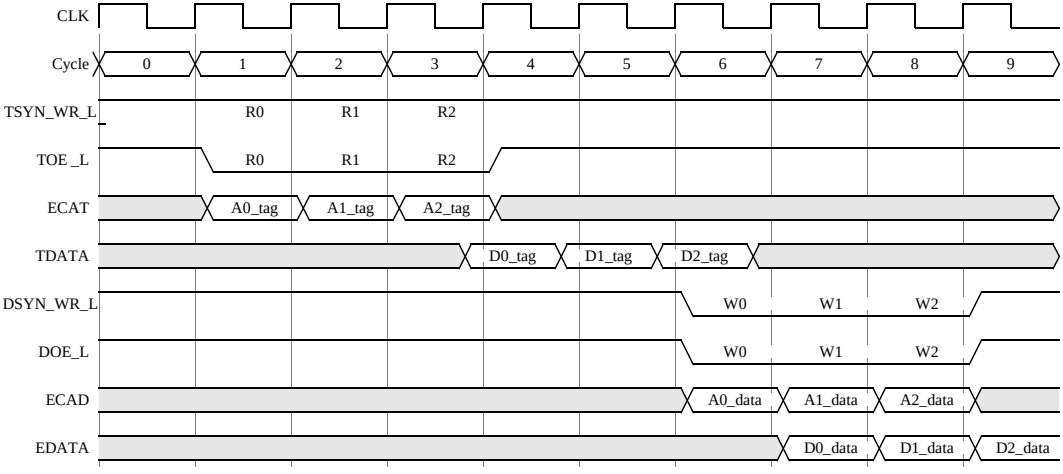


Figure 6. Coherent Write Hit to M State Line

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*. 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).

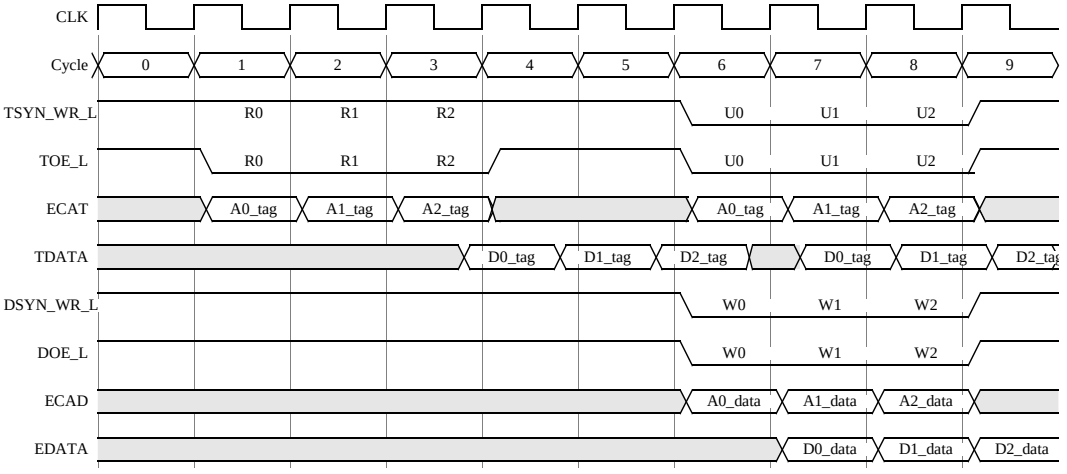


Figure 7. Coherent Writes with E to M Updates

Overlap of Tag and Data Access

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 (discussed in a later section).

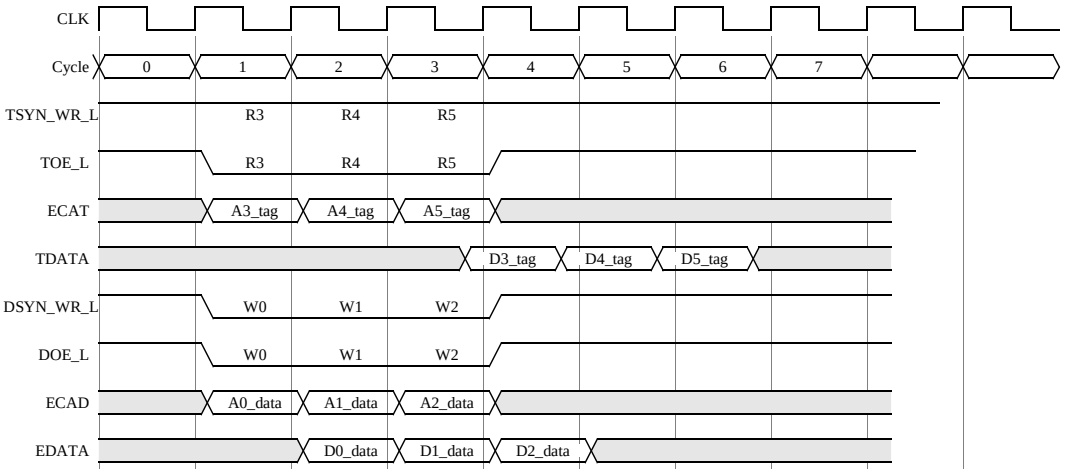


Figure 8. Overlap Between Tag Access and Data Write for Coherent Writes

Coherent Read Followed by a Coherent Write

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–I (due to electrical considerations). UltraSPARC–I uses a one-deep write buffer in the data SRAMs to reduce the turnaround penalty (going from reads to writes) to two cycles. The data of a write is sent one cycle after the address (*Figure 9*). 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 dead cycles between A2_tag and A3_tag as well as between A2_data and A3_data).

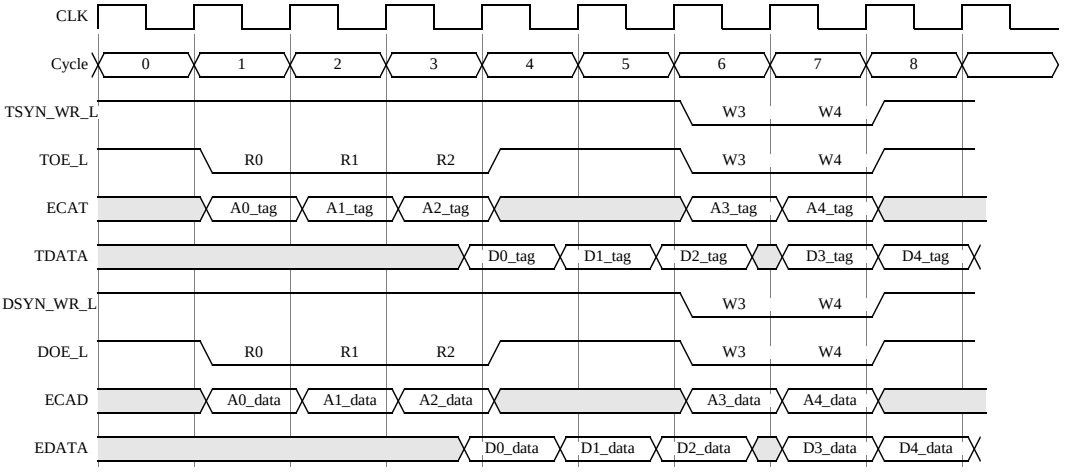


Figure 9. Reads Followed by Writes; Turn Around Penalty

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings[1]

Symbol	Parameter	Rating	Units
V_{CC}	Supply voltage range	0 to 4.0	V
V_I	Input voltage range[2]	-0.5 to V_{CC} + 0.5	V
V_O	Output voltage range	-0.5 to V_{CC} + 0.5	V
I_{IK}	Input clamp current ($V_I < 0$ or $V_I > V_{CC}$)	±20	mA
I_{OK}	Output clamp current ($V_O < 0$ or $V_O > V_{CC}$)	±50	mA
	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
V_{CC}	Supply voltage		3.2	3.3	3.465	V
V_{SS}	Ground		–	0	–	V
V_{IH}	High-level input voltage	All except CLK	2.0	–	V_{CC} + 0.3	V
		CLK	2.4	–	V_{CC} + 0.3	V
V_{IL}	Low-level input voltage	All except CLK	-0.3	–	0.8	V
V_{IL}		CLK	-0.3	–	1.6	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	V_{CC} = Min, I_{OH} = Max	2.4	–	–	V
V_{OL}	Low-level output voltage [1]	V_{CC} = Min, I_{OL} = Max	–	–	0.4	V
V_{IH}	High-level input voltage (except CLKA, CLKB, UDBCLKA, UDBCLKB, LOOPCAP)	V_{CC} = Max	2.0	–	–	V
	High -level input voltage (CLKA, CLKB, UDBCLKA, UDBCLKB)	V_{CC} = Max	2.4	–	–	V
V_{IL}	Low-level input voltage (except CLKA, CLKB, UDBCLKA, UDBCLKB, LOOPCAP)	V_{CC} = Min	–	–	0.8	V
	Low-level input voltage (CLKA, CLKB, UDBCLKA, UDBCLKB, LOOPCAP)	V_{CC} = Min	–	–	0.8	V
V_{IH}, V_{IL}	High-level, low-level input voltage for LOOPCAP	Pin should be grounded	–	0	–	V
I_{DDD}	Supply current	V_{CC} = Max, freq = 167 MHz	–	8	–	A
		V_{CC} = Max, freq = 200MHz	–	10	–	A
I_{OZ}	High-impedance output current [2]	V_{CC} = Max, V_O = V_{CC}	-10	–	10	µA
		V_{CC} = Max, V_O = V_{SS}	-10	–	10	µA
I_I	Input current	V_{CC} = Max, V_O = V_{SS} to V_{CC}	-10	–	10	µA
C_I	Input capacitance [3]		–	5	–	pF
C_O	Output capacitance		–	10	–	pF

1. STOP_CLK has no V_{OL} specification.
2. Only bidirectional lines can be three-state. Output-only cannot be three-state. All bidirectional lines will be three-stated when RESET_L is held LOW and SRAM_TEST is held high.
3. 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)

Symbol	Parameter	Signals	Conditions	167 MHz		200Mhz		Units
				Min	Max	Min	Max	
t_{SU} [1]	Input setup time to CLK	SYSADDR[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	Figure 12	2.4	–	2.2	-	ns
t_{H} [1]	Input hold time to CLK			0.9	–	0.9	-	ns
t_{PD} [1]	Output delay from CLK	SYSADDR[35:0], ADR_VLD, EDATA[127:0], EDPAR[15:0], TDATA[24:0], TPAR[3:0], BYTEWE_L, ECAT[15:0], TSYN_WR_L, TOE_L, P_REPLY[4:0], NODEX_RQ	I_{OL} = 8 mA I_{OH} = -4 mA C_L = 35 pF V_{LOAD} = 1.5V	–	2.55	-	1.95	ns
t_{OH} [1]	Output hold time from CLK			0.2	–	0.2	-	ns
t_{PD} [1]	Output delay from CLK	ECAD[17:0], DOE_L,DSYN_WR		-	2.55	-	1.55	ns
t_{OH} [1]	Output hold time from CLK	ECAD[17:0],DOE_L,DSYN_WR		0.2	-	0.2	-	ns
t_{PD} [1]	Output delay from CLK	STOP_CLK, EPD		–	4.8	-	4.4	ns
t_{OH} [1]	Output hold time from CLK	EPD		0.5	–	0.5	-	ns
t_{LOCK}	PLL acquisition time			6.0		5.0		uµs
				1000		1000		cycle
t_{SKEW}	UDBCLK skew to CLK [3]		Figure 14	0	1.3	0	1.3	ns

1. All timing requirements are specified with PLL enabled.
2. RESET_L is asserted asynchronously but deasserted synchronously.
3. UDBCLK is before CLK as shown in *Figure 14*.

AC Characteristics - Clock Timing

Symbol	Parameter	167 MHz			200Mhz			Units
		Min	Typ	Max	Min	Typ	Max	
t_{CYC}(CLK)	Processor clock Tcycle time [1]	6	–	–	5	-	-	ns
t_{w}(CLK)	Processor clock duty cycle [1]	40	50	60	40	50	60	%
t_{SLEW}(CLK)	Clock input slew rate [1]	3	–	–	3	-	-	V/ns
t_{CYC}(TCK)	TCK clock cycle time	0	–	20	0	-	20	ns
t_{w}(TCK)	TCK clock duty cycle	40	50	60	40	50	60	%
t_{CYC}(UDBCLK)	UDBCLK clock cycle time (Divide-by-2 Mode)	t_{CYC} (CLK) x2	–	–	t_{CYC} (CLK) x2	-	–	ns
t_{CYC} (UDBCLK)	UDBCLK clock cycle time (Divide-by-3 Mode)	t_{CYC} (CLK) x3	–	–	t_{CYC} (CLK) x3	-	–	ns
t_{w} (UDBCLK)	UDBCLK duty cycle	40	50	60	40	50	60	%
t_{w} (RESET_L)	RESET pulse width LOCK MODE	10	–	–	10	-	–	ns
t_{w} (RESET_L)	RESET pulse width BYPASS MODE	t_{CYC} (CLK) x3	–	–	t_{CYC} (CLK) x3	-	–	ns

1. This is for the PLL enabled.

AC Characteristics - JTAG Timing

Symbol	Parameter*	Signals	Conditions	167 MHz			200MHz			Units
				Min	Typ	Max	Min	Typ	Max	
t_{SU}(TRST_L)	Input setup time to TCK	TRST_L		10.0	–	–	10.0	–	–	ns
t_{SU}(TDI)	Input setup time to TCK	TDI		1.5	-	–	1.5	-	–	ns
t_{SU}(TMS)	Input setup time to TCK	TMS		3.0	-	–	3.0	-	–	ns
t_{H}(TRST_L)	Input hold time to TCK	TRST_L		0	-	–	0	-	–	ns
t_{H}(TDI)	Input hold time to TCK	TDI		1.0		–	1.0		–	ns
t_{H}(TMS)	Input hold time to TCK	TMS		1.0	-	–	1.0	-	–	ns
t_{PD}(TDO)	Output delay from TCK [1]	TDO	I_{OL} = 8 mA	–	-	8.0	–	-	8.0	ns
t_{OH}(TDO)	Output hold time from TCK [1]	TDO	I_{OH} = -4 mA C_{L} = 35 pF V_{LOAD} = 1.5V	2.0	–	-	2.0	–	-	ns

1. TDO is referenced from falling edge of TCK.

TABLE 6: AC Characteristics - TPD (Output) Capacitive Derating Factor[1]

Symbol	Parameter	167 MHz			200 MHz			Units
		Min	Typ	Max	Min	Typ	Max	
t_{PD}	Capacitive derating factor	0.2	0.3	0.4	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.

TABLE 7: Thermal Resistance vs. Air Flow[1]

Symbol	Air Flow (ft/min)				Units
	200	400	600	800	
$Theta_{JA}$	1.8	1.2	1.0	0.9	(×°C/W)

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

PARAMETER MEASUREMENT INFORMATION

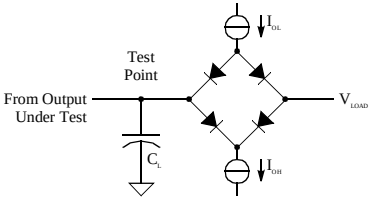


Figure 10. Load Circuit

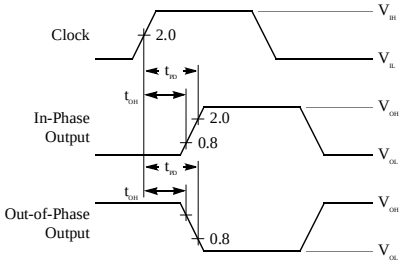


Figure 11. Voltage Waveforms - Propagation Delay Times

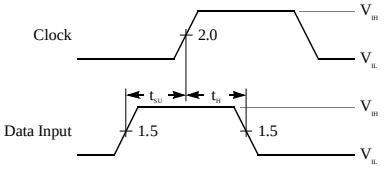


Figure 12. Voltage Waveforms - Setup and Hold Times

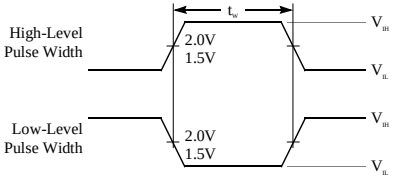


Figure 13. Voltage Waveforms - Clock Pulse Duration

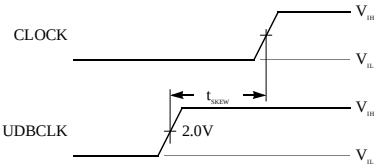


Figure 14. Voltage Waveforms - Clock Skew

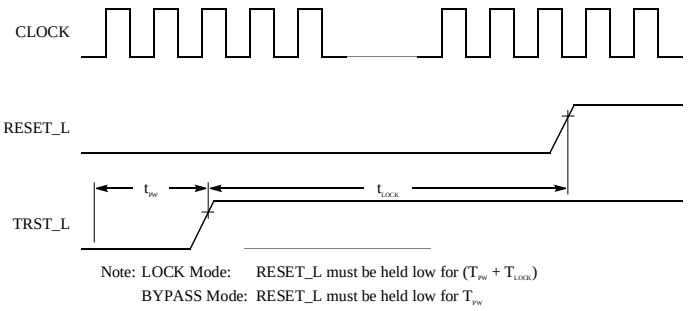


Figure 15. Reset Timing

PACKAGING INFORMATION

PBGA 521 Pin Assignment [1] [2] [3]

Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name
A5	VSSO	B30	VDDO	D23	EDATA[58]	F32	EDATA[111]	L33	EDATA[74]	S1	SYSADDR[14]
A6	MISC_BIDIR[9]	C3	VSSC	D24	VDDO	F33	EDATA[108]	M1	NODEX_RQ	S2	SYSADDR[15]
A7	MISC_BIDIR[5]	C4	TDO	D25	EDATA[51]	G1	SC_RQ	M2	ADR_VLD	S3	VSSC
A8	MISC_BIDIR[1]	C5	MISC_BIDIR[[14]	D26	EDATA[48]	G2	XIR_L	M3	VSSO	S4	VSSO
A9	EDATA[125]	C6	MISC_BIDIR[12]	D27	VSSO	G3	UDB_CEL	M4	VSSC	S5	SYSADDR[17]
A10	EDATA[121]	C7	VSSO	D28	EDATA[22]	G4	UDB_UEL	M5	VDDC	S29	VSSC
A11	EDATA[120]	C8	MISC_BIDIR[2]	D29	spare	G5	UDB_UEH	M29	VSSC	S30	EDATA[12]
A12	EDATA[116]	C9	EDATA[127]	D30	VDDC	G29	EDATA[110]	M30	VDDO	S31	EDATA[13]
A13	EDATA[113]	C10	VDDO	D31	VSSC	G30	EDATA[109]	M31	EDATA[73]	S32	VSSO
A14	EDATA[93]	C11	EDATA[122]	D32	VDDO	G31	VSSO	M32	EDATA[47]	S33	EDPAR[1]
A15	EDATA[91]	C12	EDATA[118]	E1	VSSO	G32	EDPAR[13]	M33	EDATA[72]	T1	SYSADDR[16]
A16	VDDC	C13	VSSO	E2	L5CLK	G33	EDATA[107]	N1	SYSADDR[1]	T2	SYSADDR[19]
A17	EDATA[87]	C14	EDATA[94]	E3	EPD	H1	S_REPLY[0]	N2	SYSADDR[0]	T3	SYSADDR[18]
A18	EDATA[86]	C15	VSSC	E4	VSS_QUIET	H2	DATA_STALL	N3	SYSADDR[2]	T4	RAM_TEST
A19	EDATA[84]	C16	VDDO	E5	VDD_QUIET	H3	NODE_RQ[2]	N4	VDDO	T5	LOOPCAP
A20	EDATA[81]	C17	VDDC	E6	MISC_BIDIR[13]	H4	NODE_RQ[1]	N5	VDDC	T29	EDATA[11]
A21	EDATA[61]	C18	VSSC	E7	MISC_BIDIR[8]	H5	NODE_RQ[0]	N29	VDDC	T30	VDDO
A22	EDPAR7	C19	VSSO	E8	MISC_BIDIR[4]	H29	VSSC	N30	VDDC	T31	EDATA[9]
A23	EDATA[56]	C20	EDATA[82]	E9	MISC_BIDIR[0]	H30	EDATA[106]	N31	VSSO	T32	EDATA[10]
A24	EDATA[52]	C21	EDATA[63]	E10	VDDC	H31	VDDC	N32	EDATA[45]	T33	EDATA[8]
A25	EDATA[49]	C22	VDDO	E11	EDPAR[15]	H32	VDDO	N33	EDATA[46]	U1	UDBCLKA
A26	EDATA[29]	C23	EDATA[57]	E12	EDATA[119]	H33	EDATA[105]	P1	SYSADDR[3]	U2	UDBCLKB
A27	EDATA[28]	C24	EDATA[54]	E13	EDPAR[14]	J1	S_REPLY[2]	P2	SYSADDR[4]	U3	RESET_L
A28	EDATA[26]	C25	VSSO	E14	EDATA[112]	J2	S_REPLY[1]	P3	SYSADDR[5]	U4	CLKA
A29	VSSO	C26	EDATA[30]	E15	VDDC	J3	VDDC	P4	SYSADDR[6]	U5	VSS_PLL
B4	VDDO	C27	EDATA[27]	E16	EDPAR[11]	J4	VSSC	P5	VSSC	U29	VDDC
B5	MISC_BIDIR[11]	C28	EDATA[23]	E17	EDATA[90]	J5	S_REPLY[3]	P29	VSSC	U30	VSSC
B6	MISC_BIDIR[10]	C29	EDPAR[2]	E18	EDATA[89]	J29	VSSC	P30	EDPAR[5]	U31	VSSO
B7	MISC_BIDIR[6]	C30	EDATA[19]	E19	VDDC	J30	VSSO	P31	EDATA[44]	U32	EDATA[103]
B8	VDDO	C31	VSSO	E20	EDPAR[10]	J31	EDATA[104]	P32	VDDO	U33	EDATA[102]
B9	EDATA[126]	D2	VDDO	E21	EDATA[80]	J32	EDATA[79]	P33	EDATA[41]	V1	PLLBYPASS
B10	EDATA[124]	D3	VDDC	E22	VDDC	J33	EDATA[78]	Q1	SYSADDR[9]	V2	TRST_L
B11	VSSO	D4	VSSC	E23	EDATA[59]	K1	P_REPLY[0]	Q2	VSSO	V3	CLKB
B12	EDATA[117]	D5	VDDC	E24	EDATA[55]	K2	P_REPLY[1]	Q3	SYSADDR[7]	V4	SCLK_MODE
B13	EDATA[114]	D6	VDDO	E25	EDPAR[6]	K3	P_REPLY[2]	Q4	SYSADDR[10]	V5	VDD_PLL
B14	VDDO	D7	MISC_BIDIR[7]	E26	EDATA[31]	K4	VSSC	Q5	SYSADDR[8]	V29	EDATA[101]
B15	EDATA[92]	D8	MISC_BIDIR[3]	E27	EDATA[25]	K5	VSSC	Q29	EDATA[15]	V30	EDPAR[12]
B16	VSSC	D9	VSSO	E28	EDATA[20]	K29	VDDC	Q30	VSSO	V31	EDATA[100]
B17	VSSO	D10	VSSC	E29	spare	K30	VSSC	Q31	EDATA[42]	V32	VDDO
B18	EDATA[88]	D11	EDATA[123]	E30	EDATA[18]	K31	VDDO	Q32	EDATA[43]	V33	EDATA[99]
B19	EDATA87[85]	D12	VDDO	E31	EDATA[16]	K32	EDATA[76]	Q33	EDATA[40]	W1	SYSADDR[20]
B20	VDDO	D13	EDATA[115]	E32	EDATA[17]	K33	EDATA[77]	R1	SYSADDR[13]	W2	VDDO
B21	EDATA[62]	D14	EDATA[95]	E33	VSSO	L1	P_REPLY[3]	R2	SYSADDR[11]	W3	SYSADDR[21]
B22	EDATA[60]	D15	VSSO	F1	UDB_CEH	L2	VDDO	R3	VDDO	W4	SYSADDR[23]
B23	VSSO	D16	VDDC	F2	TMS	L3	P_REPLY[4]	R4	SYSADDR[12]	W5	SYSADDR[22]
B24	EDATA[53]	D17	VSSC	F3	TCK	L4	VSSC	R5	VDDC	W29	EDATA[98]
B25	EDATA[50]	D18	VDDO	F4	TDI	L5	VDDC	R29	VDDC	W30	VSSO
B26	VDDO	D19	VSSC	F5	EXT_EVENT	L29	VDDC	R30	VDDC	W31	EDATA[97]
B27	EDPAR[3]	D20	EDATA[83]	F29	VSSC	L30	EDATA[75]	R31	VDDO	W32	EDATA[96]
B28	EDATA[24]	D21	VSSO	F30	VDDO	L31	EDPAR[9]	R32	VSSC	W33	EDATA[71]
B29	EDATA[21]	D22	VSSC	F31	VDDC	L32	VSSO	R33	EDATA[14]	X1	SYSADDR[24]
X2	SYSADDR[25]	AA31	VSSO	AE4	TDATA[12]	AF8	VDDC	AG14	ECAD[9]	AH22	ECAT[6]
X3	VSSO	AA32	EDATA[37]	AE5	VDDC	AF9	VSSC	AG15	VDDO	AH23	VDDO
X4	SYSADDR[26]	AA33	EDATA[36]	AE6	TDATA[18]	AF10	VSSO	AG16	VDDC	AH24	ECAT[13]
X5	SYSADDR[27]	AB1	TDATA[3]	AE7	TDATA[21]	AF11	VDDC	AG17	VDDC	AH25	BYTEWE_L[0]
X29	VDDC	AB2	TDATA[4]	AE8	TDATA[24]	AF12	ECAD[3]	AG18	VSSO	AH26	VSSO
X30	VSSC	AB3	VDDO	AE9	VDDC	AF13	VDDO	AG19	ECAD[17]	AH27	BYTEWE_L[5]

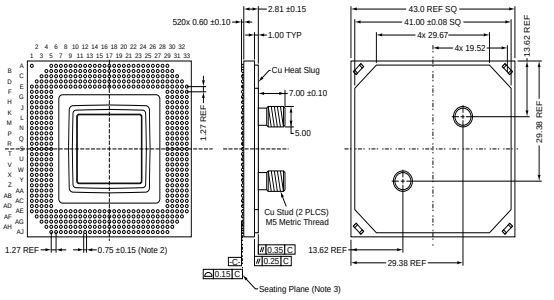
PBGA 521 Pin Assignment (Continued) [1] [2] [3]

Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name	Pin	Pin Name
X31	VDDO	AB4	TDATA[5]	AE10	DOE_L	AF14	VSSC	AG20	UDB_CNTL[4]	AH28	BYTEWE_L[10]
X32	EDATA[70]	AB5	TDATA[9]	AE11	ECAD[1]	AF15	ECAD[11]	AG21	VDDO	AH29	BYTEWE_L[14]
X33	EDATA[69]!	AB29	EDATA[33]	AE12	ECAD[2]	AF16	VSSO	AG22	VDDC	AH30	VSSO
Y1	SYSADDR[28]	AB30	EDATA[34]	AE13	VSSC	AF17	VSSC	AG23	ECAT[9]	AJ5	VDDO
Y2	SYSADDR[30]	AB31	EDATA[35]	AE14	VDDC	AF18	VDDC	AG24	VSSO	AJ6	TDATA[19]
Y3	SYSADDR[31]	AB32	VDDO	AE15	VDDC	AF19	VDDO	AG25	ECAT[15]	AJ7	TDATA[22]
Y4	VDDO	AB33	EDPAR[4]	AE16	VSSC	AF20	UDB_CNTL[3]	AG26	VSSC	AJ8	TOE_L
Y5	SYSADDR[32]	AC1	TPAR[0]	AE17	VDDC	AF21	ECAT[1]	AG27	VDDO	AJ9	TSYN_WR_L
Y29	EDATA[68]	AC2	TDATA[6]	AE18	ECAD[15]	AF22	VSSO	AG28	BYTEWE_L[11]	AJ10	ECAD[0]
Y30	EDPAR[8]	AC3	TDATA[7]	AE19	VSSC	AF23	ECAT[7]	AG29	BYTEWE_L[12]	AJ11	VDDC
Y31	EDATA[67]	AC4	VSSO	AE20	UDB_CNTL[2]	AF24	ECAT[12]	AG30	VDDC	AJ12	ECAD[5]
Y32	VSSO	AC5	TDATA[11]	AE21	ECAT[2]	AF25	VDDO	AG31	VDDO	AJ13	ECAD[8]
Y33	EDATA[66]	AC29	EDATA[5]	AE22	ECAT[5]	AF26	BYTEWE_L[3]	AH4	VSSO	AJ14	ECAD[10]
Z1	SYSADDR[29]	AC30	VSSO	AE23	VSSC	AF27	BYTEWE_L[7]	AH5	TDATA[17]	AJ15	ECAD[13]
Z2	SYSADDR[33]	AC31	EDATA[7]	AE24	ECAT[11]	AF28	VSSO	AH6	TPAR[2]	AJ16	ECAD[12]
Z3	SYSADDR[34]	AC32	EDATA[6]	AE25	VDDC	AF29	BYTEWE_L[15]	AH7	TDATA[23]	AJ17	VSSC
Z4	VDD_PECL	AC33	EDATA[32]	AE26	BYTEWE_L[4]	AF30	VSSC	AH8	VSSO	AJ18	ECAD[14]
Z5	TDATA[0]	AD1	TDATA[8]	AE27	BYTEWE_L[9]	AF31	TEMP_SEN[0]	AH9	TPAR[3]	AJ19	UDB_CNTL[1]
Z29	EDATA[65]	AD2	TDATA[10]	AE28	BYTEWE_L[13]	AF32	VDDO	AH10	DSYN_WR_L	AJ20	ECAT[0]
Z30	VDDO	AD3	TPAR[1]	AE29	TEMP_SEN[1]	AG3	VDDC	AH11	VDDO	AJ21	ECAT[3]
Z31	EDATA[64]	AD4	TDATA[13]	AE30	VSSC	AG4	PM_OUT	AH12	ECAD[4]	AJ22	ECAT[8]
Z32	EDATA[39]	AD5	TDATA[15]	AE31	EDATA[0]	AG5	SPARE	AH13	ECAD[7]	AJ23	ECAT[10]
Z33	EDATA[38]	AD29	EDPAR[0]	AE32	EDATA[1]	AG6	VDDC	AH14	VSSO	AJ24	ECAT[14]
AA1	SYSADDR[35]	AD30	EDATA[2]	AE33	VSSO	AG7	TDATA[20]	AH15	VSSC	AJ25	BYTEWE_L[1]
AA2	VSSO	AD31	VDDC	AF2	VSSO	AG8	VSSC	AH16	VSSC	AJ26	BYTEWE_L[2]
AA3	STOP_CLOCK	AD32	EDATA[3]	AF3	VDDC	AG9	VDDO	AH17	VDDO	AJ27	BYTEWE_L[6]
AA4	TDATA[1]	AD33	EDATA[4]	AF4	VSSC	AG10	VSSC	AH18	ECAD[16]	AJ28	BYTEWE_L[8]
AA5	TDATA[2]	AE1	VDDO	AF5	VSSC	AG11	VSSC	AH19	UDB_CNTL[0]	AJ29	VDDO
AA29	VDDC	AE2	TDATA[14]	AF6	VSSC	AG12	VSSO	AH20	VSSO		
AA30	VSSC	AE3	TDATA[16]	AF7	VDDO	AG13	ECAD[6]	AH21	ECAT[4]		

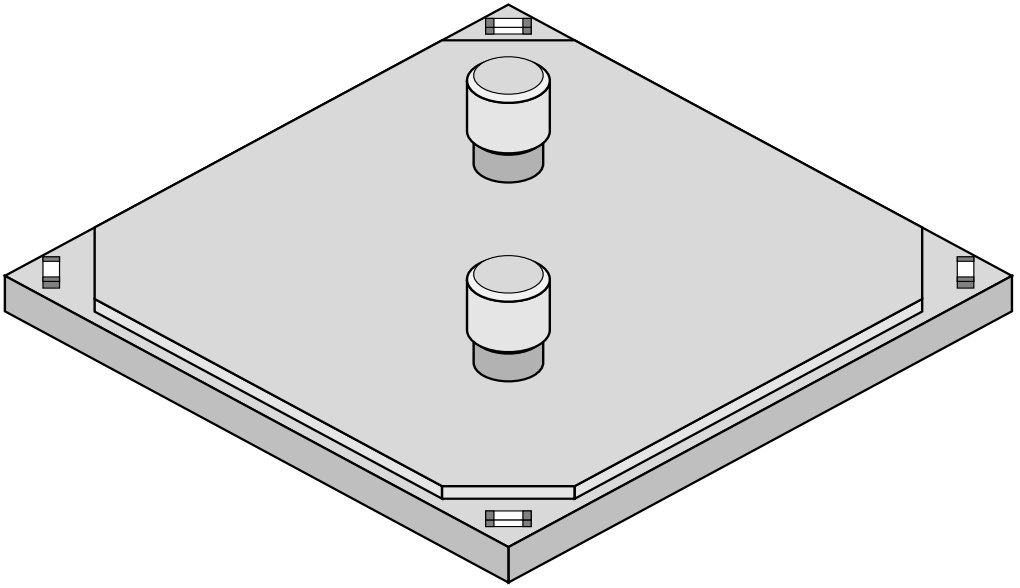
1. V_{DD}/V_{SS} Core: V_{DD}/V_{SS} for the Input, Core and Memory are tied together on the die to the same power plane. The V_{DD}/V_{SS} core pins are attached to these planes.
2. V_{DD}/V_{SS} Out: V_{DD}/V_{SS} outputs on the die have their own separate planes which are accessed through the V_{DD}/V_{SS}–OUT pins.
3. V_{DD}/V_{SS} Quite, PLL and PECL are bonded directly to the package pins.

PACKAGE INFORMATION

256-Pin PBGA Package Dimensions



Notes: 1. Dimensions in mm.
2. Measured at maximum solder ball diameter parallel to primary datum C.
3. Primary datum C and seating plane are defined by the spherical crowns of the solder balls.



ORDERING INFORMATION

Part Number	Speeds	Description
STP1030ABGA-167	167 MHz	High Performance 64-Bit RISC Processor
STP1030ABGA-200	200 MHz	

SUN MICROELECTRONICS

2550 Garcia Avenue
Mountain View, CA, USA 94043
800-681-8845
www.sun.com/sparc

Part Number: 802-7432-01