# UltraSPARC® IIe Processor

## User's Manual

## Supplement to the UltraSPARC IIi User's Manual

Version 1.1

February 2003

# Table of Contents

# List of Figures

# List of Tables

# Preface

The UltraSPARC® IIe processor manual contains information about the architecture and programming of the UltraSPARC IIe processor. It describes the details of the processor's new features.

The UltraSPARC IIe processor is part of Sun Microsystems' UltraSPARC II Processor family, an enhanced 64-bit, SPARC V9 architecture implementation. The UltraSPARC IIe processor includes an SDRAM memory controller that supports SDRAM DIMMs and a 32-bit, 66 MHz PCI bus interface, compatible with the PCI Specification, Version 2.1. The processor integrates a 256 KB L2-cache onto the chip, includes a clock frequency controller and new STICK timer, and operates at a lower processor core voltage than previous processors.

## *SPARC V9 Architecture Manual*

*The SPARC Architecture Manual, Version 9* defines the processor architecture and is available from many technical bookstores or directly from its copyright holder:

SPARC International, Inc., 535 Middlefield Road, Suite 210
Menlo Park, CA 94025, (415) 321-8692

*The SPARC Architecture Manual, Version 9* provides a complete description of the SPARC V9 architecture. Since SPARC V9 is an open architecture, many of the implementation decisions have been left to the manufacturers of SPARC-compliant processors. These "implementation dependencies" are introduced in *The SPARC Architecture Manual, Version 9*.

## *UltraSPARC IIi User's Manual*

Since the UltraSPARC IIe processor is very similar to the UltraSPARC IIi processor, the UltraSPARC IIi User's Manual is a necessary companion to UltraSPARC IIe User's Manual Supplement.

## *UltraSPARC I/II User's Manual*

The original UltraSPARC II*s*-series processor is described in the UltraSPARC I/II User's Manual. In some cases, this manual may provide additional information concerning the operation of the processor. Normally, the UltraSPARC IIi User's Manual is sufficient as a supplement.

## *Other UltraSPARC User's Manuals*

All other processor UltraSPARC II User Manuals may be helpful.

# Textual Conventions

### *Font Usage:*

- *Italic font* is used for emphasis, book titles, and the first instance of a word that is defined. Italics are also used for Assembly Language terms.
- `Courier` font is used for register fields (named bits), instruction fields, signals, and read-only register fields. Courier is also used for literals, instruction names, and software examples.
- **Bold font** is used for emphasis.
- UPPERCASE items are acronyms, instruction names, or writable register fields, and external signals. **Note**: Names of some instructions contain both uppercase and lowercase letters.
- Underbar character (_) joins words together in registers, register fields, and signal names.

### *Notation:*

- Square brackets '[ ]' indicate the bits of a register field or external signal name.
- Angle brackets '< >' indicate a textual substitution.
- h7'03C indicates first 7 least significant bits in the hex number 03C are relevant.

### *Examples:*

- `SIGNAL_NAME`, `BUS_SIGNALS`[31:0], `ACTIVE_LOW_SIGNAL_L`
- *Register_Bit_Field*, *Range_Of_Bits[3:0]*
- <enter_filename>, **Emphasis**
- `BERR` bit

# Where to Find Things

The following table can be used to find discussions about the UltraSPARC IIe processor.

The UltraSPARC II processor User's Manual includes the UltraSPARC IIi User's Manual and the UltraSPARC I/II User's Manual.

**TABLE 0-1** Documentation List

| **Item** | **Document** | | |
|---|---|---|---|
| | **Name**<br>**Chapter/Section** | | **Reference** |
| Architecture, Operation, and CSRs of processor / MMU | User's Manual | UltraSPARC II | |
| Architecture, Operation, and CSRs of L1-caches | User's Manual | UltraSPARC II | |
| Architecture, Operation, and CSRs of L2-caches | User's Manual | UltraSPARC IIe | Chapter 3, *Level 2 Cache Subsystem*, page 19 |

**TABLE 0-1** Documentation List *(Continued)*

| **Item** | **Document** | | |
|---|---|---|---|
| | **Name Chapter/Section** | | **Reference** |
| Architecture, Operation, and CSRs of Memory Controller | User’s Manual | UltraSPARC IIe | Chapter 5, *Memory Control Unit (MCU)*, page 43 |
| Architecture, Operation, & CSRs of PCI Subsystem | User’s Manual | UltraSPARC IIi | UltraSPARC IIi User’s Manual, Chapter 19 |
| Clock Operations | User’s Manual | UltraSPARC IIe | Section 2.1, *Clocks,* on page 11 |
| Errata upto UltraSPARC IIi | User’s Manual | UltraSPARC II | UltraSPARC IIi User’s Manual,, Appendix K |
| Glossary | User’s Manual | UltraSPARC II | |
| Interrupts and Traps | User’s Manual | UltraSPARC II | UltraSPARC IIi User’s Manual,, Chapter 11 |
| Memory ASI Definitions | User’s Manual | UltraSPARC II | UltraSPARC IIi User’s Manual,, Chapter 6 and this document |
| Memory Transaction Ordering | User’s Manual | UltraSPARC II | |
| Power Management Energy Star (E-Star) Operation | User’s Manual | UltraSPARC IIe | Section 2.2, *Clock Frequency Control,* on page 13 |
| Programming Code Generation Guidelines | User’s Manual | UltraSPARC II | UltraSPARC IIi User’s Manual,, Chapter 21 |
| Programming Grouping Rules and Stalls | User’s Manual | UltraSPARC II | UltraSPARC IIi User’s Manual,, Chapter 22 |
| System Memory Map | User’s Manual | UltraSPARC IIe | Section 4.3, *Memory Space,* on page 39 |

## CHAPTER 1

# UltraSPARC IIe Processor Overview

The UltraSPARC IIe processor integrates a 256 KB L2-cache, an SDRAM memory controller, a 66 MHz 32-bit PCI Bus Interface, and a power management feature. The processor is very similar to all other UltraSPARC II processors. It implements the 64-bit SPARC V9 architecture and the VIS™ instruction set. The SPARC V9 architecture provides binary compatibility across all SPARC processors. The VIS instruction set performs parallel execution on multiple pixel data widths of 8 and 16 bits to accelerate the most common operations related to processing, 2D and 3D graphics, compression algorithms, and numerous network operations. The VIS instruction set enables high bandwidth for memory-to-processor and memory-to-memory transfers by providing 64-byte block load and block store operations.

### *Integrated Features*

The SDRAM DIMM memory controller supports up to 2 GB of memory using four double-sided, 512 MB DIMMs with 128 Mb SDRAMs or four single sided, 512 MB DIMMs with 256 Mb SDRAMs.

The PCI Bus subsystem provides command and data buffering, and an I/O memory management unit (IOM) for PCI bus masters accessing main memory. The processor's host bus interface is PCI Bus 2.1 compatible, 32 bits wide, operates at up to 66 MHz, sends and receives 3.3 V signals, and is often connected to Sun's Advanced PCI Bridge (APB). The APB extends the PCI Bus structure to include two additional bus segments of 32 bits at 33 MHz with 3.3 or 5.0 V signaling.

The fully integrated L2-cache contains up to 256 KB of space for instructions and data. The L2-cache allocates space in 4-way set-associative and direct-mapped mode.

Power Management Logic provides a mechanism to slow down the processor clock rate. This reduces power consumption while running the operating system.

The JTAG interface supports boundary scan for systemboard interconnect testing.

Each functional area on the UltraSPARC IIe processor maintains decentralized control, allowing many activities to overlap.

# 1.1 UltraSPARC IIe Processor Implementation

## 1.1.1 New Features

The following list of items are features in the UltraSPARC IIe processor that are not necessarily found in previous UltraSPARC processors (s-series and II), but will impact the system software and some of the application software.

- Memory Controller (SDRAM) – New, impacts initialization code (firmware)
- Clock Control Unit (1/2 and 1/6 frequency modes) – New, enables Energy Star (E-Star) mode
- STICK Timer – New, impacts Operating System (OS) time base when E-Star mode is used
- Traps – Minor software changes for the STICK timer support
- L2-cache – New internal 256 KB cache replaces external L2-cache. New cache flushing method required, no other impact to software code
- Four General Purpose Output (GPO) signals – New, available for PCI clock control or other functions

## 1.1.2 Features Removed

The following list of items are *not* supported in the UltraSPARC IIe processor that were supported in previous UltraSPARC processors.

- UPA Bus (all port types, including UPA64S)
- External tag and data L2-cache SRAMs (replaced by internal cache RAM arrays)
- EDO DRAM memory controller (replaced by SDRAM memory controller)

## 1.1.3 Processor Comparison

All processors listed below include the UltraSPARC II pipeline and the VIS instruction set. The MMU and L1-caches structures are very similar. TABLE 1-1 shows a comparison of processor implementations.

**TABLE 1-1** Processor Implementation Comparison

| | **UltraSPARC II*s*-series** | **UltraSPARC IIi** | **UltraSPARC IIe** |
|---|---|---|---|
| Sun Platforms | Ultra$^{\text{TM}}$ 1, Ultra 2,<br>Sun Enterprise$^{\text{TM}}$ Servers | Ultra 10, Ultra 20,<br>UltraAXi | Sun Blade$^{\text{TM}}$ 100,<br>Netra$^{\text{TM}}$ t1120, t1125,<br>t1400, t1405,<br>CP2060, CP2080,<br>AX1105 |
| Year of first system | 1996 | 1998 | 2000 |
| Clock Frequency | 167 to 480 MHz | 270 to 440 MHz | 400, 500 MHz |
| Process Technology | 0.35 and 0.25 µm Al | 0.35 and 0.25 µm Al | 0.18 µm Al |
| System Bus | UPA64M (up to 64-way) | UPA64S (graphics only) | PCI |
| I/O Bus | S-bus and PCI bus via UPA system bridge | PCI 66 MHB, 32-bit | |
| Memory Bus | EDO DRAM | EDO DRAM | SDRAM |

**TABLE 1-1** Processor Implementation Comparison *(Continued)*

| | **UltraSPARC IIs-series** | **UltraSPARC IIi** | **UltraSPARC IIe** |
|---|---|---|---|
| Maximum Memory | | 1 GB | 2 GB |
| L2-cache | 1 to 8 MB, external, module dependent | 256 KB to 1 MB, external, module dependent | 256 KB On-Chip,<br>4-way Set-Associative |
| Energy Star Mode | No | No | 1/2 and 1/6 |

# 1.2 Processor Architecture

The UltraSPARC IIe processor consists of six major components. The components are listed with their interconnections in FIGURE 1-1.

The central compute engine and primary caches in the UltraSPARC IIe processor provides very similar functionality as all other UltraSPARC II processors. The UltraSPARC IIe processor has integrated features to further reduce systemboard size, board cost, and power dissipation.

## 1.2.1 Processor/MMU/Primary Level 1 Caches

**Compatibility Note –** The primary level 1 caches are the same as all other UltraSPARC II processors, with an enhancement for trap generation to serve the new STICK timer.

**Documentation Note –** See the other UltraSPARC II processor manuals for the description of the processor, MMU, and primary caches.

## 1.2.2 Integrated Level 2 Cache

- Secondary Cache (L2-Cache) – Unified Instruction-Data Memory, 256 KB, 4-way set-associative or direct-mapped mode
- Cache Control Unit (ECU) – Interfaces the L2-cache to the processor, Memory, and PCI subsystems

## 1.2.3 SDRAM Memory Subsystem

- Memory Interface Unit (MIU) – Accepts, buffers, checks for data coherency, and arbitrates memory requests
- SDRAM Memory Control Unit (MCU) – 72-bit interface

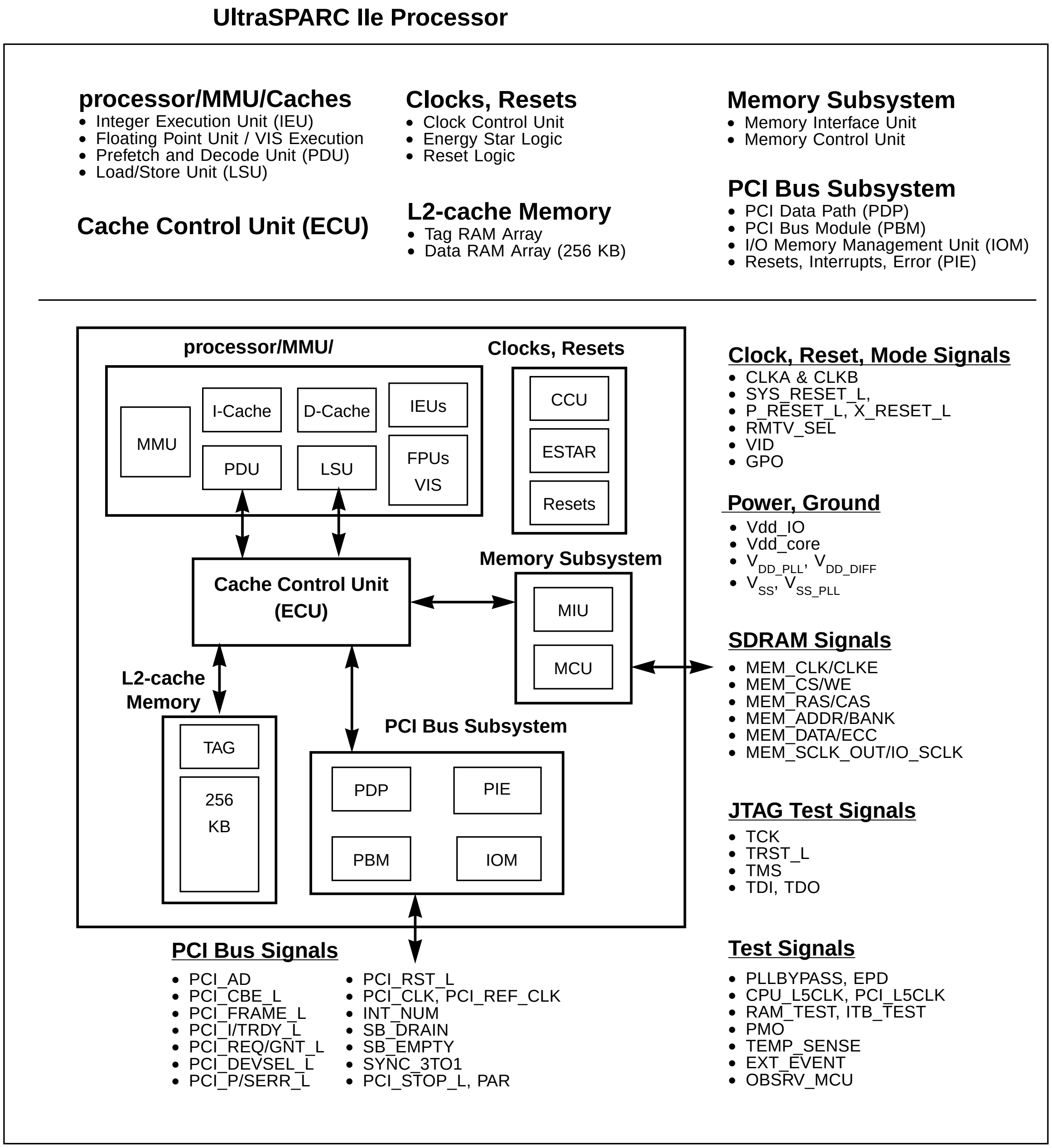


**FIGURE 1-1** Simplified Processor Block Diagram and I/O Signals

## 1.2.4 PCI Bus Subsystem

---

**Compatibility Note –** The PCI Bus subsystem is same as the UltraSPARC IIi processor.

---

The major blocks of the PCI Subsystem includes:

- PCI Bus Module (PBM) – 33/66 MHz, 32-bit, 3.3 V PCI Host Bus Interface
- I/O Memory Management Unit (IOM) – Translates PCI addresses to memory's physical address

- PCI Data Path (PDP) – Dual 64-byte data buffer (one for PIO and one for DMA)
- PCI Resets, Interrupts, and Error (PIE) – External interrupts processed

The PCI Subsystem is clocked independently from the processor. A 2-entry, bidirectional command buffer is at the PCI to processor clock domain boundary to decouple activities from the processor to improve PCI data transfer bandwidth.

### 1.2.5 System Control

#### *Clocks*

- Processor Clock Input – Differential CLKA/B
- *New:* Clock Control Unit (CCU) – PLL, 1/2 and 1/6 divider select
- Internal Clock Distribution – Utilizes internal PLL to reduce on-chip clock skew
- Memory Clock derived from CLKA/B – Programmable divider
- PCI Clock – 66 MHz Subsystem Clock, 33/66 MHz PCI Interface Clock

#### *Resets*

- POR, system (hardware), and XIR (software)
- `Red_State` Mode Trap Address Vector Select (`RMTV_SEL`)
- Test Interfaces: JTAG, Factory Tests

#### *Diagnostics*

- Control Status Registers (CSRs) – Most processor subsystems
- TAP Controller, JTAG – Boundary Scan

## 1.3 System Perspective

The UltraSPARC IIe processor interfaces directly to industry standard SDRAM DIMMs for memory. The processor also contains a PCI 2.1 compatible bus interface for system I/O functions. These interfaces provide a high degree of compatibility with standard design practices and device interfacing.

The entire system is memory mapped with Address Space Identifiers (ASI) that add functionality to each load/store transaction from the processor. This expands the effective address space of the processor and reveals special registers for system control.

Sun offers and recommends a number of system devices, including:

- Advanced PCI Bridge (APB) that expands the UltraSPARC IIe PCI Bus Interface to two PCI Bus segments.
- PCIO-2 Multifunction PCI I/O controller that supports Ethernet, Sun’s 8-bit E-Bus, USB, and IEEE1394
- IChip2 System ASIC for I/O interrupt concentration, and PCI clocks

Older devices compatible with the UltraSPARC IIe processor includes:

- PCIO Multifunction PCI I/O controller that supports Ethernet and Sun's 8-bit E-Bus
- RIC System ASIC for I/O interrupt concentration, reset control, and JTAG clocks

These devices provide Sun-proven hardware and software compatibility. System designers can choose from a number of architectures based on these and standard PCI devices. Design requirements and software efforts need to be considered in addition to device functionality when choosing the best devices for an architecture.

FIGURE 1-2 illustrates a typical system block diagram.

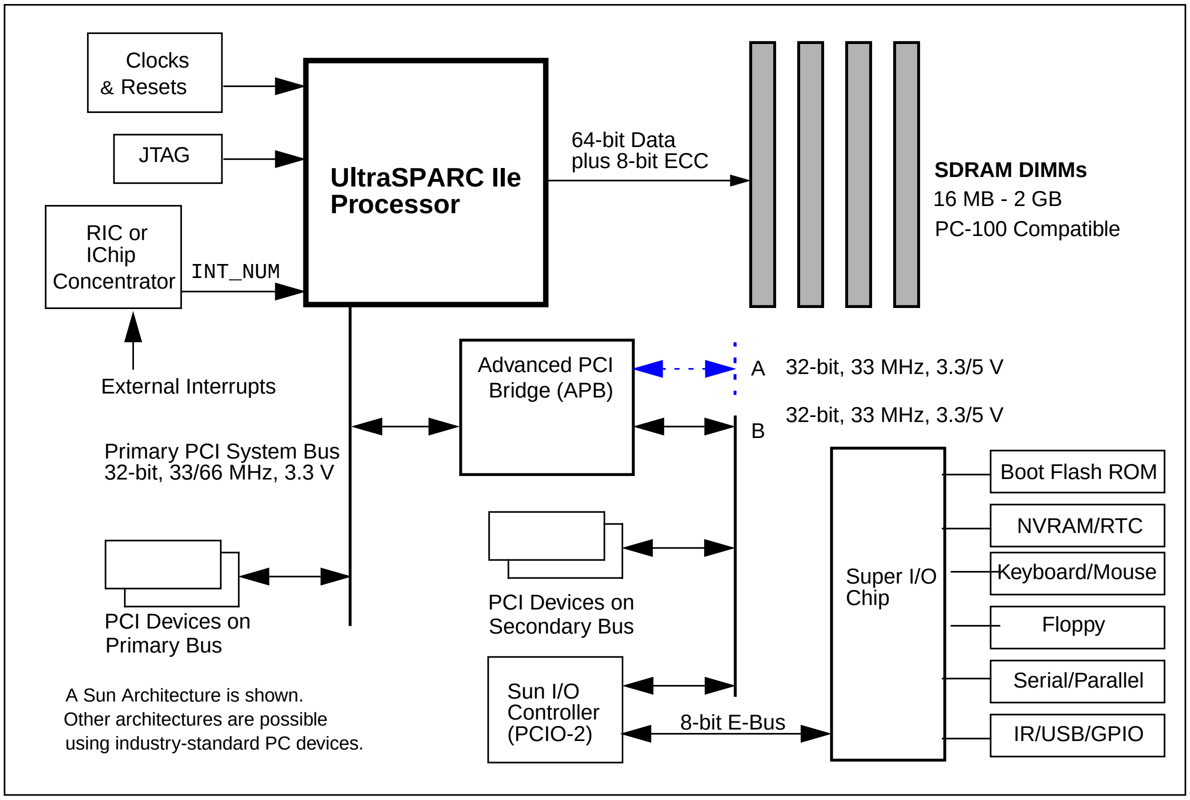


**FIGURE 1-2** Typical System Block Diagram

## 1.3.1 Power Management

The processor can be slowed to 1/2 and under certain operating conditions, 1/6 the normal operating frequency. The memory controller can put the SDRAMs into Self Refresh mode. Software can further reduce system power consumption by controlling system devices with power down capabilities.

## 1.3.2 Memory Subsystem

The processor supports up to four double-sided PC-100 style SDRAM DIMMs (8 banks, total). The processor clock to SDRAM clock ratio is selectable (4 to 7).

Each DIMM can have one or two physical banks and they can all be of a different address size and configuration. Modes and timing parameters are shared across the DIMMs. The memory interface has programmable I/O buffer strengths to adjust the DC current output drive on separate groups of signals to optimize signal transmission integrity over various capacitive loading conditions. SDRAM memories can be operated in Self Refresh mode to reduced power consumption.

## 1.3.3 PCI Bus Architecture

The PCI Bus subsystem directly interfaces the processor to a 32-bit, Version 2.1 compliant PCI Bus running at speeds up to 66 MHz (which yields a maximum theoretical transfer rate of 264 MB/s). The PCI Bus Arbiter can support up to four external PCI Bus Masters. The number of devices that can be attached depends on the physical limits and the bus clock frequency. A built-in I/O Memory Management Unit (IOM) will translate PCI memory space addressees from the PCI Bus Master to the physical addresses of the main memory. The processor is a PCI Slave in this DMA transfer mode to and from memory. The IOM also supports hardware tablewalk in the case of a TLB miss in the IOM. All memory reads and writes initiated by a PCI Bus Master (DMA) are cache coherent with the processor.

The processor boots by initiating a 32-bit memory read request on the PCI Bus Interface. The UltraSPARC IIe processor has two sets of trap vectors to be compatible with Sun and the PC boot address modes.

### *Advanced PCI Bridge (APB) Chip*

The APB extends the UltraSPARC IIe PCI Bus to two PCI 2.1 bus segments of 33 MHz, 32-bit each. The APB drives to 3.3 V levels. The secondary bus segments have configurable I/O buffers to be 5 V tolerant. The APB supports DMA from up to four bus masters on each secondary bus segment.

The APB interfaces seamlessly with the UltraSPARC IIe processor. Software is available to support the APB and the 2115x class of PCI bridges.

### *System Interrupts (INT_NUM Bus)*

The PCI subsystem processes I/O interrupts from the systemboard that are received on its 6-bit `INT_NUM` bus. Dozens of interrupt lines are scanned, encoded or concentrated onto the `INT_NUM` bus by a system ASIC containing an “Interrupt Concentrator.” The UltraSPARC IIe processor uses software interlocks and hardware write buffer (store buffer) flushing logic to synchronize a DMA transfer to the interrupt handler.

System interrupts are considered part of the PCI Subsystem because they service PCI devices or devices indirectly attached to the PCI Bus.

### *PCIO Multifunction PCI I/O Controller*

The PCI I/O controller (PCIO, STP2003QFP) chip is a multifunction PCI Controller that includes a 10/100 Ethernet interface and an E-Bus host controller.

### *PCIO-2 Multifunction PCI I/O Controller (Enhanced)*

The second generation PCI I/O Controller (PCIO-2, SME2300BGA) chip is a multifunction PCI Controller that includes a 10/100 Ethernet interface, an E-Bus host controller, an IEEE 1394 Firewire Interface, and four USB bus interfaces.

### 1.3.4 System ASICs

#### *RIC Reset/Interrupt/Clock ASIC*

The RIC System Controller (SME2210) supports the system resets, system interrupts, system scans, and system clock control functions for UltraSPARC II *s*-series processors. Its features includes:

- Resets from power supply, reset buttons, and scan chain
- Interrupt Concentrator – 41 signals in, 6-bit encoded `INT_NUM` bus out
- Directs scan inputs and outputs through scan chains
- Combinational logic for UPA bus speed
- 160 pin PQFP

#### *IChip2 Interrupt Controller ASIC (Enhanced)*

The IChip2 System Controller (SME2212QFP) provides similar Interrupt Concentrator function as the RIC chip. The rest of the IChip2 includes a PCI clock controller requiring a differential voltage input signal.

- Interrupt Concentrator – 48 signals in, 6-bit encoded `INT_NUM` bus out
- PCI Clock Controller – Compatible with asynchronous dual bus structures
- 128-pin TQFP package
- Newer device than RIC

The IChip and IChip2 controllers are functionally equivalent. The IChip System Controller is packaged in a 120-pin MQFP.

## 1.4 Software Perspective

There are new ASIs for accessing the memory controller, the L2-cache RAMs, and the PCI Bus Interface Controllers. Main memory (SDRAMs) is mapped as cacheable. All the PCI memory spaces are non-cacheable memory mapped. This includes configuration, I/O, and memory.

---

**Compatibility Note –** The processor architecture is similar to the processor architecture of all other UltraSPARC II processors.

The PCI Bus architecture is similar to the PCI architecture in the UltraSPARC IIi processor.

---

#### *Endianess Note*

The UltraSPARC IIe processor uses the big-endian addressing format. The code space and all processor registers are big-endian except the PCI Configuration Space Header in the PCI subsystem and the PCI Bus itself.

The processor supports little-endian data structures using a combination of the byte swapper in the PCI Bus subsystem and the ASI descriptors of the processor.

### *System Bus Hierarchy Model Note*

The UltraSPARC system architecture is bus hierarchy-based. The processor’s I/O system bus is the PCI Bus Interface. The optional Advanced PCI Bridge (APB) provides two secondary PCI Bus segments.

Sun’s *PCIO-2 PCI Multifunction I/O Controller* provides an interface to Ethernet, IEEE 1394, E-Bus and USB type busses to further define the system’s bus hierarchy which originates at the processors primary Host PCI Bus Interface.

CHAPTER **2**

# Clocks, System Timer, GPO, and Resets

## 2.1 Clocks

There are three root clock domains in normal operation.

- Processor clock (`CLKA`, `CLKB`, differential signal pair, 400/500 MHz)
- PCI (`PCI_CLK LVTTL` signal, 66 MHz)
- JTAG (`JTAG_TCK LVTTL` signal)

All three sets of clocks are normally asynchronous to each other. Synchronizers are used to transfer address data, and control signals between the PCI and processor clock domains. FIGURE 2-1 illustrates a clocks block diagram.

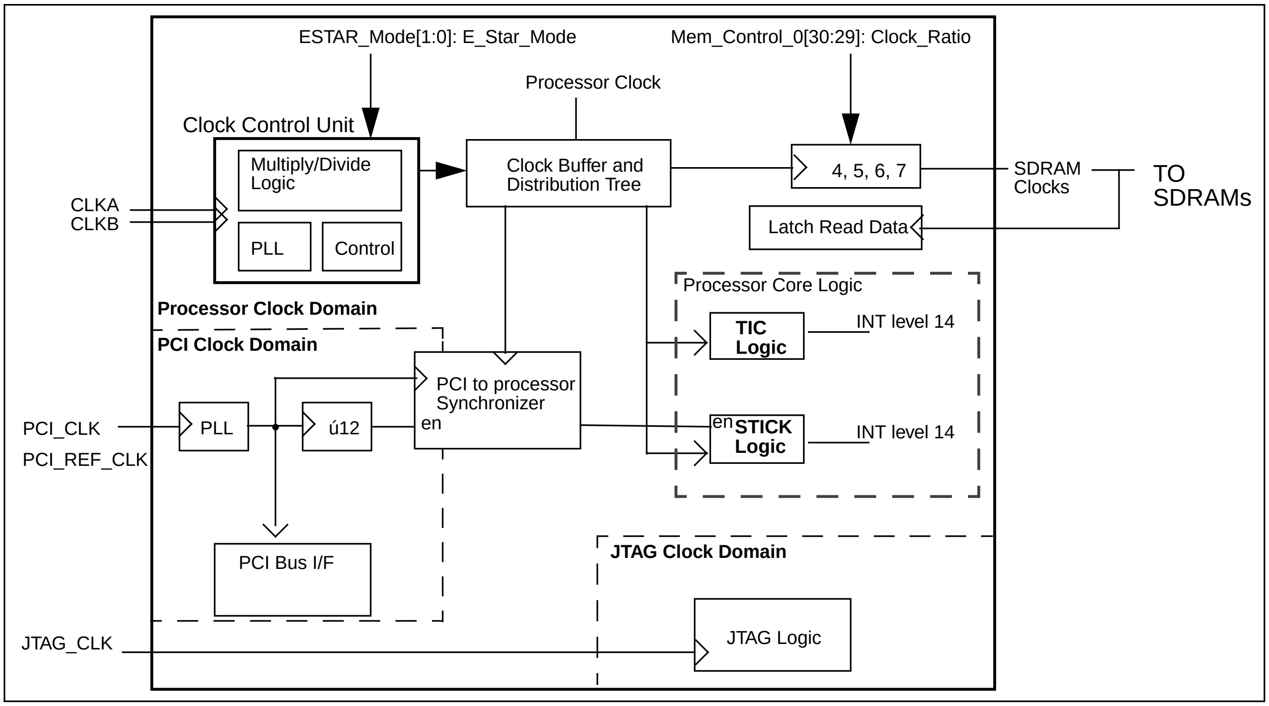


**FIGURE 2-1** Clocks Block Diagram

## 2.1.1 CLKA and CLKB Processor Clock Signal

The `CLKA` and `CLKB` clock pair are driven continuously and at a constant rate of 1/2 the processor's normal operating frequency.

### *Clock Control Unit (CCU)*

The processor clock input signal on the processor is a differential signal pair. The Clock Control Unit (CCU) converts this to a CMOS signal, uses it to drive its PLL, and operates high speed dividers to provide three processor frequency mode settings to reduce power dissipation.

The processor clock is driven at a constant frequency by system logic and runs continuously while operating the processor. The clock is driven at one-half the processor operating frequency in normal operating mode. The processor frequency can be reduced to 1/2 (same frequency as input clock signal) or 1/6 the normal operating frequency, by writing to the Energy Star (E-Star) register.

### *Timebase for Software – TICK and STICK*

The processor contains two clock timers that can be read by software or be used to generate interrupts at fixed intervals of time. Each timer contains a counter, a count value register, and a compare register. The counter updates the count value register. When the count value register equals the compare register value, an interrupt is generated. The TICK logic is incremented by the processor clock.

The STICK logic (new in the UltraSPARC IIe processor) uses the PCI clock for a constant time base. The PCI clock provides a constant time base to the processor STICK logic when the TICK logic is affected by the switch in processor frequency. The `PCI_REF_CLK` clock input must remain at a constant rate for the STICK logic to keep good time. The system software can use the original TICK or the new STICK logic, or a combination of both to maintain a time reference. The TICK logic is affected by the processor operating frequency and the STICK logic is affected by the PCI clock frequency.

The operation of TICK timer is described in the UltraSPARC IIi User's Manual.

## 2.1.2 Memory Clocks

The `MEM_SCLK`[7:0] signals are derived by dividing the processor clock by 4, 5, 6, or 7. The memory controller is discussed in detail in Chapter 5, *Memory Control Unit (MCU)*, page 43.

## 2.1.3 PCI Subsystem Clocks

The `PCI_CLK` clock is driven at the PCI Bus Interface frequency, typically 66 MHz or 33 MHz, although intermediate frequencies are also supported.

The `PCI_CLK` clock is divided and synchronized to the processor clock for the STICK logic. The STICK logic is read by software to maintain accurate time using the PCI clock as a time basis (independent of power down states in a processor).

## 2.1.4 JTAG Clock

The JTAG clock is independent of the other two clock domains.

## 2.2 Clock Frequency Control

Power Management consists of software detecting a system that has been idle for a prolonged period of time and then lowering the processor clock frequency to 1/2 or 1/6 the normal operating mode and optionally programming the SDRAM devices into their power down, self-refresh mode. Additional power savings in the system I/O is possible.

### 2.2.1 PCI/Processor Frequency Restrictions

The processor core frequency must be at least twice the frequency of the primary PCI Bus to ensure that the processor core correctly detects signals driven by the PCI data path inside the processor. This is further explained in the datasheet.

This requirement makes the 1/6 mode unusable when the primary bus frequency is 66 MHz.

### 2.2.2 Frequency Transitions

An example of state transitions for power management are shown in FIGURE 2-2. Consider the need to set a new auto-refresh interval with each change of processor frequency. After software changes the processor frequency, the software should, as a precaution, execute enough NOP instructions so at least 16 processor clocks occur before any memory or PCI references take place. The PCI subsystem should also be quiescent. There is no transition supported from 1/1 to 1/6 mode or visa-versa.

#### *Impact of PLL Enabled DIMMs*

The buffered and registered DIMM types contain PLL circuits on the DIMM to reduce clock skew. When the processor changes frequency, the memory clock frequencies changes, too. If this happens, the PLL enabled DIMMs lose their PLL lock causing the DIMM to be unusable until it stabilizes. Since there is no way to block memory accesses, one may occur while the PLL is locking. If this happens there is a chance the memory transaction gets corrupted and the system fails.

For this reason, we recommend not using the power down modes with registered and buffered DIMMs. Use unbuffered DIMMs when power management is required.

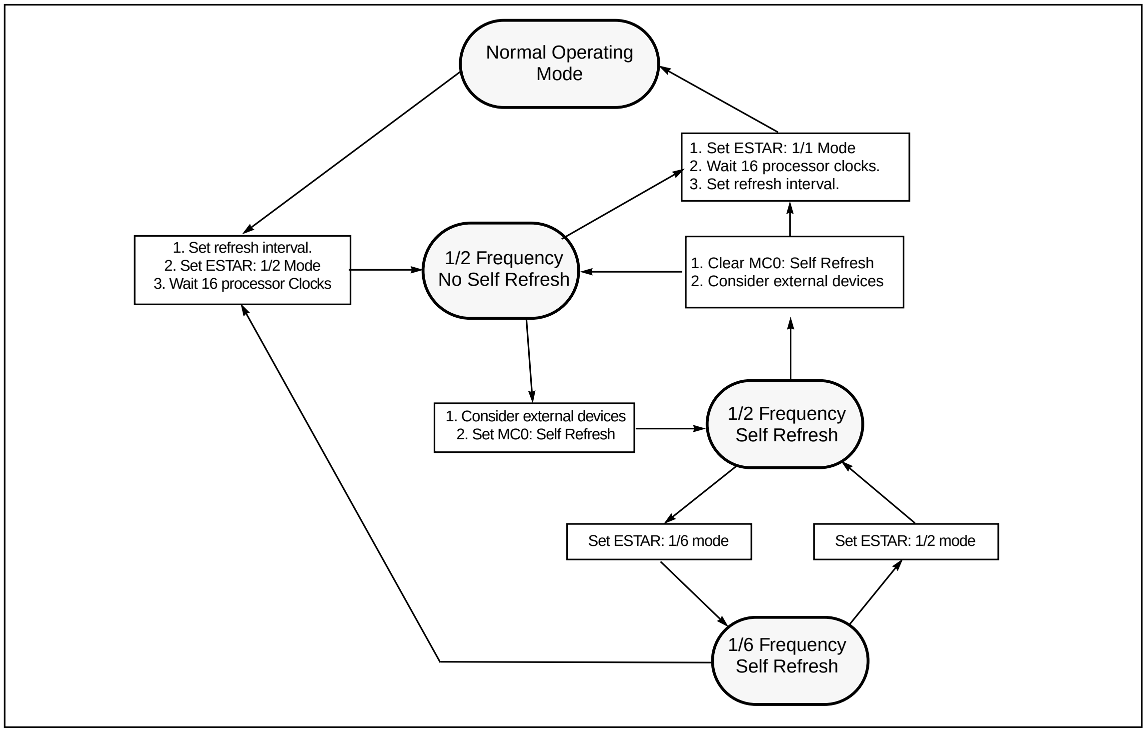


**FIGURE 2-2** Power Management State Transitions Driven by Software

## 2.2.3 Power Management (Energy Star Register)

Power Management is controlled by writing to the E-Star register.

---

**Note –** The UltraSPARC IIe processor clocking must be kept active (1/1, 1/2, or 1/6 mode). The PCI clock to the UltraSPARC IIe processor must remain active, but the PCI clock to the system devices can be stopped if proper care is taken with the PCI Bus system devices. Control of the PCI clock generator can be done by using two of the GPO signals that are driven directly by the UltraSPARC IIe processor and controlled by software. Some of Sun's architectures use GPO[1:0] for this purpose.

---

FIGURE 2-3 and TABLE 2-1 illustrates and describes the E-Star register data field, respectively.

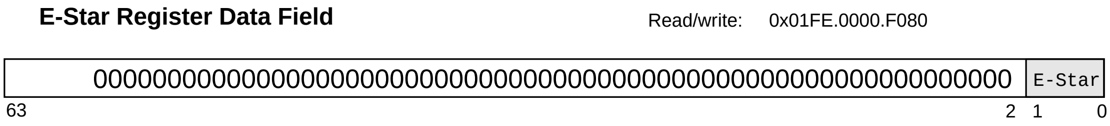


**FIGURE 2-3** Energy Star Register Data Field

**TABLE 2-1** Energy Star Register Data Field

| Field | Bits | Description | POR | Type | Documentation Reference |
|---|---|---|---|---|---|
| *Reserved* | 63:02 | *Reserved* | | | |
| `E_Star_Mode` | 1:0 | 00: Full Operating Frequency<br>01: 1/2 Operating Frequency<br>10: 1/6 Operating Frequency<br>11: *Reserved* | 00 | R/W | |

## 2.3 System Interrupt Timer

When the processor frequency is lowered (via E-Star modes) the time base for the TICK logic in the processor is affected. A new STICK timer has been created that is driven at the `PCI_CLK` signal input frequency rate which must remain constant for the PCI Bus Interface Clock PLL.

The System Tick (STICK) can provide a constant time base for the operating system because the `PCI_CLK` must be driven at a constant rate.

The STICK has an associated compare register (`STICK_CMP`) to generate a periodic interrupt for the operating system. The STICK alarm signal is gated with the TICK alarm signal. Either alarm (if enabled) will generate a level-14 (0x4e offset) trap.

The functionality is similar to the processor Tick (TICK) and Tick Compare (`TICK_CMP`) logic except it is not subject to variations in the processor clock rate.

The STICK counter is clocked by the internal processor clock, but is enabled by a pulse derived from a constant PCI Bus clock source. This means the PCI clock must remain on and at a known constant rate for the operating software to maintain accurate time when using STICK. Similarly the processor clock must remain active, but can be at a reduced rate.

The PCI clock is divided by 12 and fed into a synchronizer. The synchronizer issues an enabling pulse to the STICK counter at 1/12 the PCI Bus clock speed and does so in the processor clock domain. The enable rate is 5.5 MHz using a 66 MHz PCI Bus. The pulse is used to enable the STICK to make one count. The processor clock rate is 67 MHz for a 400 MHz processor in 1/6 power down mode so the enabling pulses from the synchronizer are always detected.

When the `STICK_CMP` logic determines that the timer has timed out, a level-14 interrupt (`STICK_ALARM`) is generated to cause a trap in the processor, same as the `TICK_CMP` logic, only separate. One, both, or neither timer can be enabled. We recommend enabling one timer at a time to simplify software.

TABLE 2-2 and TABLE 2-3 describes the STICK Register and the STICK Compare Register, respectively.

**TABLE 2-2** STICK Register

| Field | Bits | Description | POR | Type |
|---|---|---|---|---|
| *Reserved* | 63 | Reads 0, No Write | 0 | R |
| `Stick_Count` | 62:0 | STICK Register Count Value | 0 | R/W |

**TABLE 2-3** STICK Compare Register

| **Field** | **Bits** | **Description** | **POR** | **Type** |
|---|---|---|---|---|
| `Stick_Alarm_Enable` | 63 | 0 = Enable Stick Alarm (Int 14h)<br>1 = Disable Stick Alarm | 0 | R |
| `Stick_Compare_Value` | 62:0 | Field is compared to `Stick_Count`. If alarm is enabled and count matches, then Int 14h is asserted. | 0 | R/W |

## 2.4 General Purpose Outputs (GPO)

The UltraSPARC IIe processor has four general purpose output signals that come directly from the processor and are controlled by software writable registers. Two of these outputs are designated by Sun software for PCI clock control, but can otherwise be used for any purpose.

For software controlled output signals, set to 1 to drive output to 3.3 V. Set to 0 to drive output to 0 V. Output is clocked by `CLKA`/`CLKB`.

FIGURE 2-4 and TABLE 2-4 illustrates and describes the general purpose outputs data field and register, respectively.

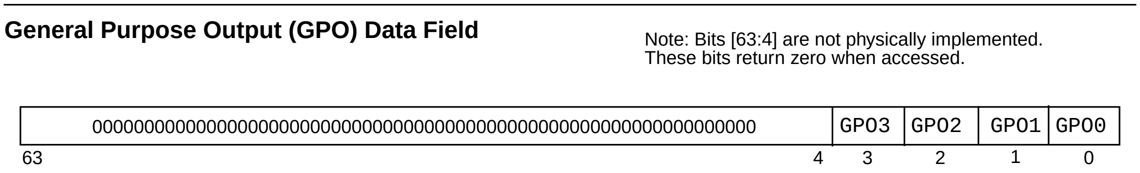


**FIGURE 2-4** General Purpose Outputs Data Field

**TABLE 2-4** General Purpose Outputs Register

| **Field** | **Bits** | **Description** | **POR** | **Type** |
|---|---|---|---|---|
| *Reserved* | 63:4 | Reads 0, No Write | 0 | RO |
| `GPO3` | 3 | Controls state of `GP3` signal | 1 | R/W |
| `GPO2` | 2 | Controls state of `GP2` signal | 1 | R/W |
| `GPO1` | 1 | Controls state of `GP1` signal | 1 | R/W |
| `GPO0` | 0 | Controls state of `GP0` signal | 1 | R/W |

## 2.5 Resets

The processor has two groups of resets – power-on and system resets, and software resets. The power-on and system resets affect the entire processor and PCI Bus subsystem. A software reset simply causes a processor trap. In each case, the cause of the reset is recorded in the Reset Control (RC) register, the processor is put into its `RED_State` condition, and the processor code execution jumps to non-cacheable ROM memory space.

The Reset Control (RC) register contains bits to enable software to generate soft resets and to record the highest level reset which the processor is responding to and recovering from.

---

**Documentation Note –** All of the Processor Reset information in this section is provided as an overview. The operation of resets has not changed significantly from that of the UltraSPARC IIi processor. The manual for this processor provides an additional source of information about processor resets.

---

### *POR Reset (Hardware Reset)*

The POR Reset is a hard reset that resets the processor and PCI Bus subsystem. The POR Reset is caused by the assertion of the `SYS_RESET_L` signal pin, the `P_RESET_L` signal pin, or by writing to the `Soft_POR` bit in the Reset Control Register. The POR Reset affects most of the processor and propagates out to the `PCI_RST_L` signal pin to reset the PCI Bus subsystem.

The POR Reset causes the processor to immediately stop its current activity. The de-assertion of reset allows a sequence of events to occur. During this sequence, the hardware is initialized, the processor is put in its `RED_State` condition, the `PCI_RST_L` signal is released, and the processor begins instruction execution to ROM memory space.

## CHAPTER 3

# Level 2 Cache Subsystem

The Level 2 Cache (L2-cache) subsystem includes the L2-tag and L2-data memory arrays and various Control, Status, and Diagnostic registers (CSRs). The L2-cache responds to the commands of the “ECU.” The ECU manages the flow of the data and control signals and is driven by memory requests and the cache states.

The ECU controls write buffers and monitors the addresses they contain in order to maintain data coherency with the caches and main memory. The ECU interfaces to the PCI subsystem to support DMA transfer requests from the PCI Bus into the coherent data domain of the processor memory.

The L2-cache memory is physically indexed and physically tagged. The cache line size in the L2-cache and main memory is 64 bytes. The L2-cache can operate in 4-way set-associative mode, or direct-mapped mode. The purpose of having two modes is to provide flexibility in operation for performance considerations (4-way), predictable behavior (direct-mapped), and to flush the cache of modified data (direct-mapped).

The L2-cache operates in a write-back mode. The primary I-cache and D-cache operate in write-through mode.

**Compatibility Note –** The UltraSPARC IIe processor includes the L2-cache tag and Data RAM arrays. Previous processors, like the UltraSPARC IIi processor, contain a cache controller that interfaces to external tag and Data SRAMs. In addition, the L2-cache in the UltraSPARC IIe processor is enhanced with a 4-way set-associative operating mode.

**Documentation Note –** The UltraSPARC IIi Processor User’s Manual contains information for the processor, MMU, I-cache/PDU, D-cache/LSU, the cache controller (ECU), and the PCI Bus subsystem. Since the operation of the UltraSPARC IIe processor is nearly identical to the UltraSPARC IIi processor for these functions, please refer to the UltraSPARC IIi Processor User’s Manual. The L2-cache in the UltraSPARC IIe processor is unique and not found on other processors.

## 3.1 Level 2 Cache Features

- 256 KB of Data Storage
  - Cache address space – PA[30:0] = 2 GB
  - Line Entry – 64-byte (8 data transfers)
  - Diagnostic Organization – 8192 64-bit data words per bank × 4 banks

- Tag RAM Array:
  - Line Entry – 15-bit tag + 2-bit status + 2-bit parity (single transfer)
  - Organization – 1024 cache line entries per bank × 4 physical banks × 20 bits
- Line Replacement Selection RAM Array (Rand Array):
  - Usage – 4-way set-associative mode only
  - Line Entry – 2-bit random replacement number (Rand)
  - Organization – 1024 cache line entries per bank × 4 banks × 64-byte cache line

### *Performance Features*

The L2-cache is pipelined and operates in the 2-2 mode as defined by previous UltraSPARC products. This enables the L2-cache to sustain the bandwidth of one 64-bit data transfers every two processor clocks from the Data RAM array. The 64-bit datapath width exists throughout the L2-Cache subsystem.

- Separate Tag and Data memory arrays support simultaneous access
- Supports delayed write, byte-write, and bank-write
- Access mode – 2-2 mode

The read access time of the tag RAM array is optimally designed to enable quick lookups of the L2-cache.

The Cache Control Unit (ECU) is fully pipelined. For programs with large data sets, instructions are scheduled with load latencies based on the L2-cache latency, therefore, the L2-cache acts as a large primary cache. Floating-point applications use this feature to effectively “hide” D-cache misses.

Separate L2-cache miss and hit operations can overlap. Stores that hit the L2-cache can proceed while a load miss is being processed. The L2-cache controller is also capable of processing reads and writes without a bus turnaround penalty.

Block loads and block stores (these load or store a 64-byte line of data from memory or L2-cache to the floating-point register file) provide high transfer bandwidth. By not caching block load/store operations (they are still in the data coherent domain) into the L2-cache on a miss, the cache is available for other data structures that are expected to be accessed more than once.

The ECU also provides support for multiple outstanding data transfer requests to the Memory subsystem and the PCI subsystem.

The peak internal bandwidth to and from the processor and the I-cache or D-cache is 2.0 GB/s at 500 MHz.

The 4-way set-associative mode tends toward better performance. The direct-mapped mode has other advantages, including a more friendly debug environment, and provides the mode to flush the cache lines to main memory.

## 3.2 Architecture

The L2-tag array contains cache control and tag bits for the contents of the L2-data array. The L2-data array contains 256 KB of data in four physical banks. These become a linear address space in direct-mapped mode and each bank maps to one of the four ways in a 4-way set-associative mode.

A high-level diagram of the L2-cache in the UltraSPARC IIe processor is shown in FIGURE 3-1. The operation of the L2-cache is explained in Section 3.3, *Cache Operating Modes,* on page 23.

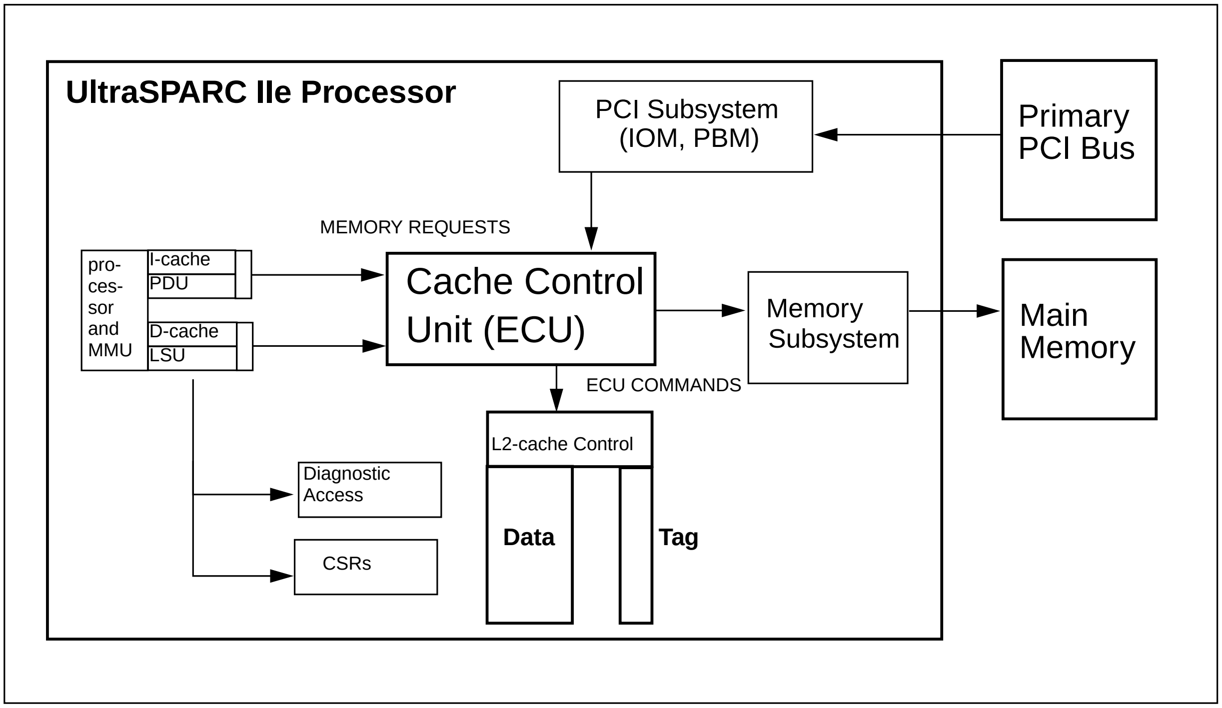


**FIGURE 3-1** Subsystem Interfaces Block Diagram

## 3.2.1 Physical Address

There is no virtual address or context information in the L2-cache. The ASIs are decoded before reaching the ECU. The fully pipelined L2-cache interface supports speculative loads and instruction prefetch requests.

The L2-cache responds to the entire main memory address range and wraps above the 2 GB physical address limit of the UltraSPARC IIe processor back to 0. See TABLE 4-2 on page 40 for a system memory map.

## 3.2.2 CSR Summary Table

All L2-cache control, status and diagnostic registers are accessed as 64-bit data quantities. A non-64-bit access causes a *mem_access_exception* trap. A non-aligned access causes a *mem_address_not_aligned* trap. The CSR registers are listed in TABLE 3-1. The registers are not 64-bit wide, but are accessed with 64-bit load and store operations.

**TABLE 3-1** Level 2 Cache Related CSR Registers

| **Register Name** | **Access Method** | | **Changed** [1] | **Documentation Manual** | **Section** |
|---|---|---|---|---|---|
| | **ASI** | **VA[40:0]** | | | |
| LSU Control | 0x45 | 0 | No | UltraSPARC IIi Manual | Appendix A.6 |
| UPA_Config | 0x4A | 0 | Yes | UltraSPARC IIe Manual | Section 3.6.1 |
| Tag RAM Diagnostic | 0x7E (read)<br>0x76 (write) | [40:39] = 10 | Yes | UltraSPARC IIe Manual | Section 3.9.1 |
| Data RAM Diagnostics | 0x7E (read)<br>0x76 (write) | [40:39] = 01 | Yes | UltraSPARC IIe Manual | Section 3.9.2 |
| Async Fault Address | 0x4D (r/w) | 0 | No | UltraSPARC IIi Manual | Section 16.6.3 |
| Async Fault Status | 0x4C (r/w) | 0 | Minor | UltraSPARC IIe Manual | Section 3.9.3 |
| | | | | UltraSPARC IIi Manual | Section 16.6.2 |

[1] *Changed* refers to differences between the UltraSPARC IIi processor and the UltraSPARC IIe processor.

## 3.2.3 Diagnostic Support

Each RAM array is accessible for diagnostics as described in Section 3.9, *Level 2 Cache Control and Status Registers (CSRs),* on page 32.

All the CSRs are listed in the UltraSPARC IIi Processor User's Manual. A subset of CSRs for the L2-cache are listed in TABLE 3-1.

## 3.2.4 Data Formats

The L2-cache uses the physical address, cache line, and register formats shown in FIGURE 3-2.

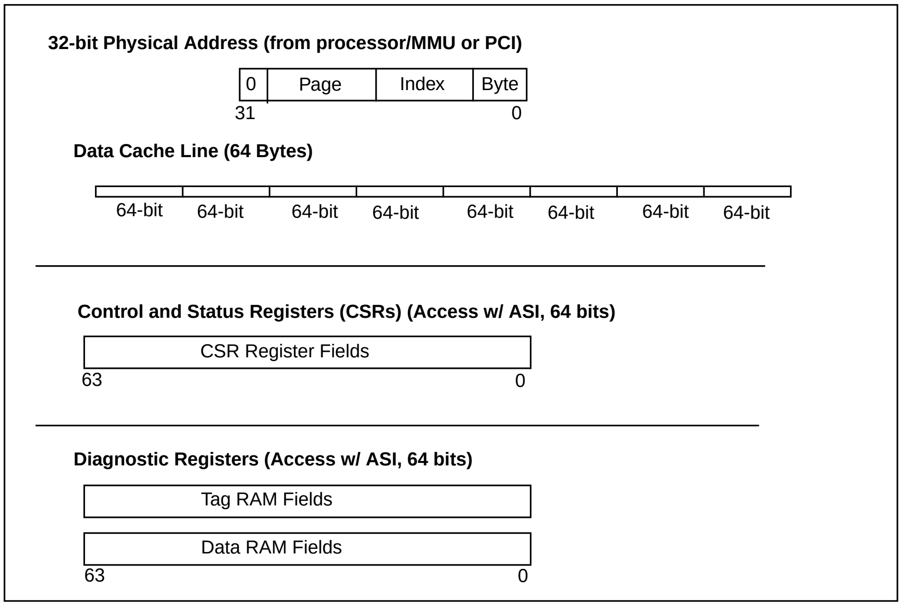


**FIGURE 3-2** Physical Address, Cache Line, and Register Formats

## 3.3 Cache Operating Modes

The L2-cache has two normal operating modes: 4-way set-associative and direct-mapped. The L2-cache also supports a diagnostic access path.

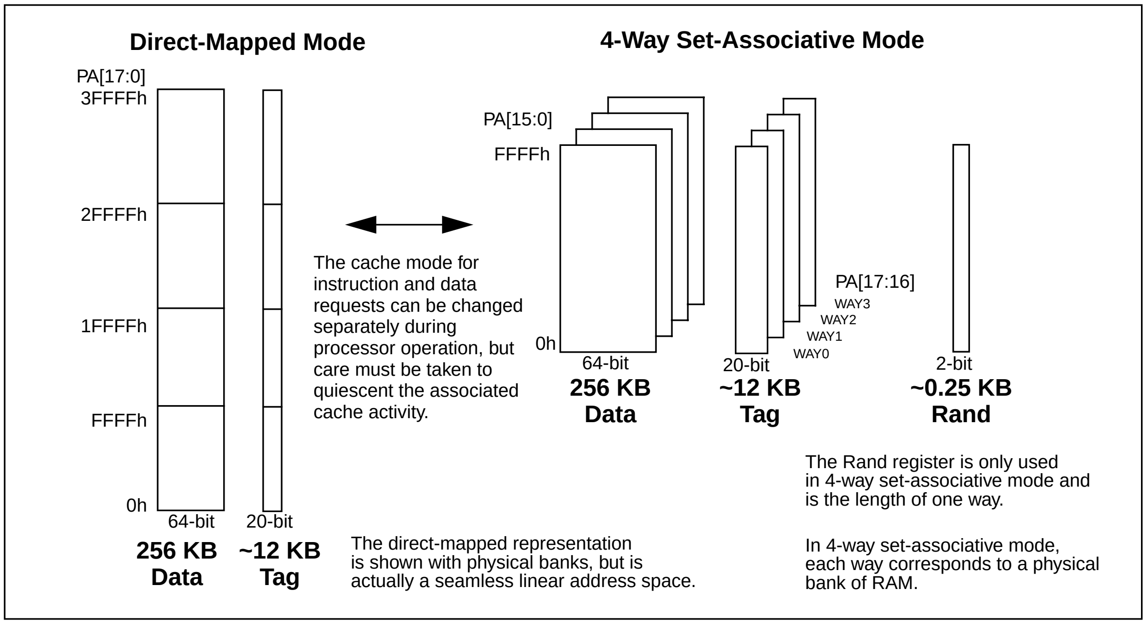
The L2-cache can operate completely in one mode or in a split mode operation. The cache mode defines the cache line replacement algorithm.

To flush a cache operating in 4-way set-associative mode, program the L2-cache so that the D-cache/LSU requests use the cache in direct mode temporarily. I-cache/PDU requests allocate in 4-way set-associative mode.

The mode selection for instruction and data are controlled separately by the `UPA_Config`<37:36> register bits (`dm_instruction` and `dm_data`).

A comparison in the arrangement of the cache arrays in 4-way set-associative and direct-mapped mode are shown in FIGURE 3-3.

The Physical Address (PA) mapping into the RAM arrays using diagnostics accesses is shown in FIGURE 3-8 on page 33.



The cache mode for instruction and data requests can be changed separately during processor operation, but care must be taken to quiescent the associated cache activity.

The direct-mapped representation is shown with physical banks, but is actually a seamless linear address space.

The Rand register is only used in 4-way set-associative mode and is the length of one way.

In 4-way set-associative mode, each way corresponds to a physical bank of RAM.

**FIGURE 3-3** RAM Array Configurations for 4-Way and Direct-Mapped Modes

## 3.3.1 Cache Line Tag RAM Entries

### *Tag Value Field*

The Tag value is compared to the index field of the physical address.

### *Line State (V and M bits)*

The cache lines are in one of three states: *modified*, *exclusive* or *invalid*. See TABLE 3-3 on page 35.

A *modified* state means the data line is valid and has the latest copy of the data. In this case, the L2-cache will source the data on a read hit. When a line replacement is needed, a modified line is flushed to memory.

*Exclusive* is an older term. In the case of the UltraSPARC IIe processor, it means the data line is valid and has not been modified.

*Invalid* cache lines do not contain valid data and are immediately available for a new entry. All cache lines need to be initiated after reset to the invalid state after reset or power-up.

### *Parity Bits*

The tag line is odd parity protected as is described in Section 3.9.3, *Asynchronous Fault Status Register Addendum,* on page 36.

#### *Rand Bits*

The Rand bits selects the way in 4-way set association replacement. The two Rand bits are considered part of the tag line and determine the next way when a displacement is required.

### 3.3.2 Data RAM Organization

An L2-cache line consists of a 64-byte quantity that is accessed from the Data RAM array using eight 64-bit transactions.

There is one line state per 64-byte cache line (invalid, exclusive, or modified). If any byte is modified in a cache line, then the whole cache line is considered modified.

## 3.4 Memory Requests

Requests to the L2-cache are generated by the ECU on behalf of the I-cache/PDU and D-cache/LSU, and by the PCI Bus subsystem: all are cacheable.

Non-cacheable requests are forwarded to the PCI subsystem by the ECU.

When a cache line is displaced to allocate a new one, the old one is written to memory if it is in the modified state. Otherwise the cache line is simply overwritten.

**Documentation Note –** Below are short descriptions on the types of requests serviced by the L2-cache. Refer to the UltraSPARC IIi Processor User’s Manual for complete and detailed discussions about these topics.

### 3.4.1 Instruction Cache/PDU Read Request

All cacheable instruction requests (including prefetch instruction fetches) that miss in the I-cache become an I-cache/PDU read request to the L2-cache. This I-cache line fill operation is always 32 bytes.

The I-cache/PDU requests read-only accesses.

### 3.4.2 Data Cache/LSU Read and Write Requests

#### *Load*

Load instructions that miss in the D-cache are forwarded to the L2-cache.

A *hit* in the L2-cache generates a 16-byte read using two consecutive 8-byte accesses to support cache line fills in the D-cache sub-block.

A *miss* causes the L2-cache to request a 64-byte cache line read of main memory. The 16 bytes of data requested by the D-cache are sourced to the D-cache and the entire 64-byte cache line from memory is put in the L2-cache, displacing an existing line.

### *Block Load*

Block load operations behave slightly different than load operations.

A *hit* in the L2-cache will cause the L2-cache to source the 64 bytes of data. No change to the cache state is made.

A block load *miss* is forwarded to main memory and the data is returned to the processor without allocating in the L2-cache.

---

**Programming Note –** Block load operations do not allocate cache memory space. Block loads are always 64 bytes and aligned to a cache line boundary. Block loads are not ordered, but are within the data coherent domain. Use the `MEMBAR#Sync` instruction to order block loads, if necessary.

---

### *Store*

Cacheable stores are queued in the LSU and update both the D-cache and the L2-cache.

Store operations are 1, 2, 4, 8, or 16 bytes long. These transactions are always aligned on their natural boundary.

A *miss* in the L2-cache will cause a fetch of a 64-byte cache line from memory and displacement of an existing cache line. The L2-cache is then updated with the byte(s) waiting to be written.

### *Block Store*

Block store operations behave slightly differently than store operations. Block store operations do not allocate space in the L2-cache. The L2-cache is checked to see if there is a hit.

A *hit* will cause the data to be written into the L2-cache.

A *miss* causes the request to go directly to the memory and the cache is not allocated.

Block stores are always 64 bytes and aligned to a cache line boundary. Block stores are not ordered.

Block stores with commit force the data to be written to memory and invalidate copies in all caches, if present.

---

**Programming Note –** Execute a `MEMBAR#Sync` after a block store and before using a load instruction that references the data from the block store. Alternatively, a second block store will force the previous block store into memory.

---

## 3.4.3 PCI DMA Read Request

The L2-cache will source data for DMA reads generated by a PCI Bus Master when a hit in the L2-cache is detected. On a *hit*, the access does not affect main memory.

On a *miss*, the access is forwarded to main memory where the memory read transaction takes place. There is no further involvement from the L2-cache.

### 3.4.4 PCI DMA Write Request

When a *hit* is detected and the cache line is modified, then the PCI DMA data byte(s) are written to the L2-cache.

When a *miss* occurs, the write request is forwarded to main memory and the L2-cache is unaffected.

## 3.5 Level 2 Cache Operating Modes

### 3.5.1 Direct-Mapped Mode

*Direct-Mapped Operation of the Tag and Data RAM Array*

A simplified diagram for the direct-mapped cache mode is shown in FIGURE 3-4.

On a read or write hit, the cache line can be in one of four locations regardless of the cache mode. This is because the cache line could be written to the cache when the cache was in 4-way mode.

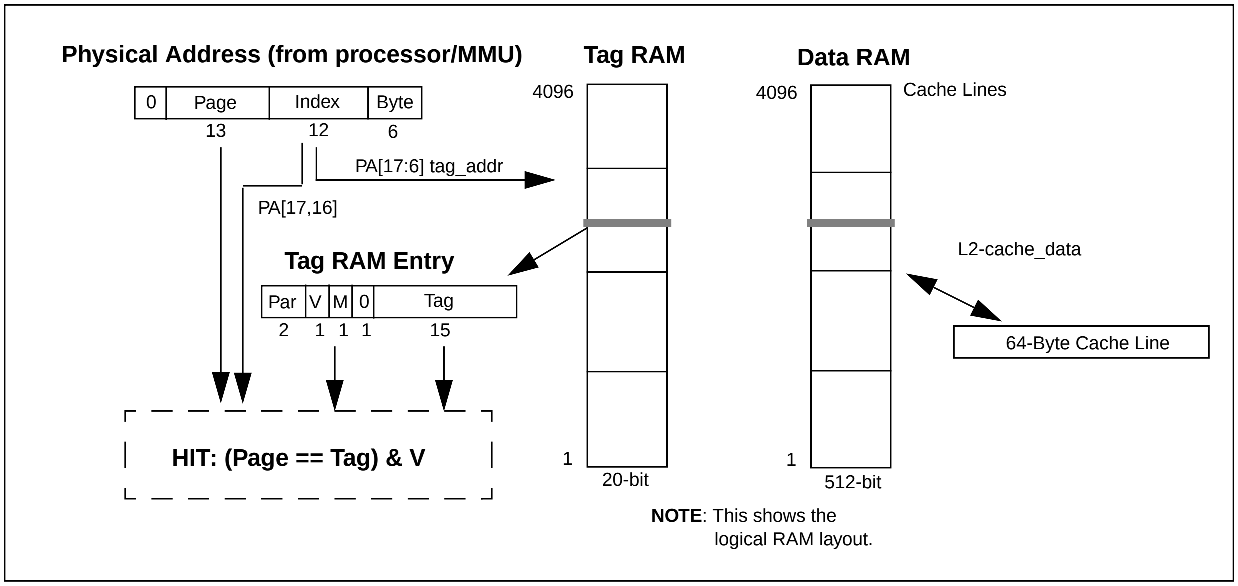


**FIGURE 3-4** Direct-Mapped Cache Mode

*Direct-Mapped Cache Line Replacement Algorithm*

The allocation of a new cache line for misses is determined by the cache mode. The direct-mapped cache line replacement algorithm has only one location that it can use. This is defined by the PA[17:6] offset address.

Data at this location must be displaced before writing the new cache line. This may involve writing the old cache line to memory (if modified), or simply invalidated.

Cache lines can also be systematically flushed out to memory under software control using a flush displacement algorithm with the cache in direct-mapped mode. This is explained in the Section 3.7, *Level 2 Cache Flush Procedure - Programming Guide,* on page 31.

## 3.5.2 4-Way Set-Associative Mode

### *4-Way Set-Associative Operation of the Tag RAM Array*

In 4-way set-associative mode, the PA[15:6] physical address points to an offset in each of the 4 ways. In parallel, the tag value in each of these line entries are compared to the PA[30:16] page address.

A *hit* to a way causes that way to be selected for the subsequent operation.

FIGURE 3-5 illustrates the 4-Way Set-Associative Operation of the Tag RAM Array.

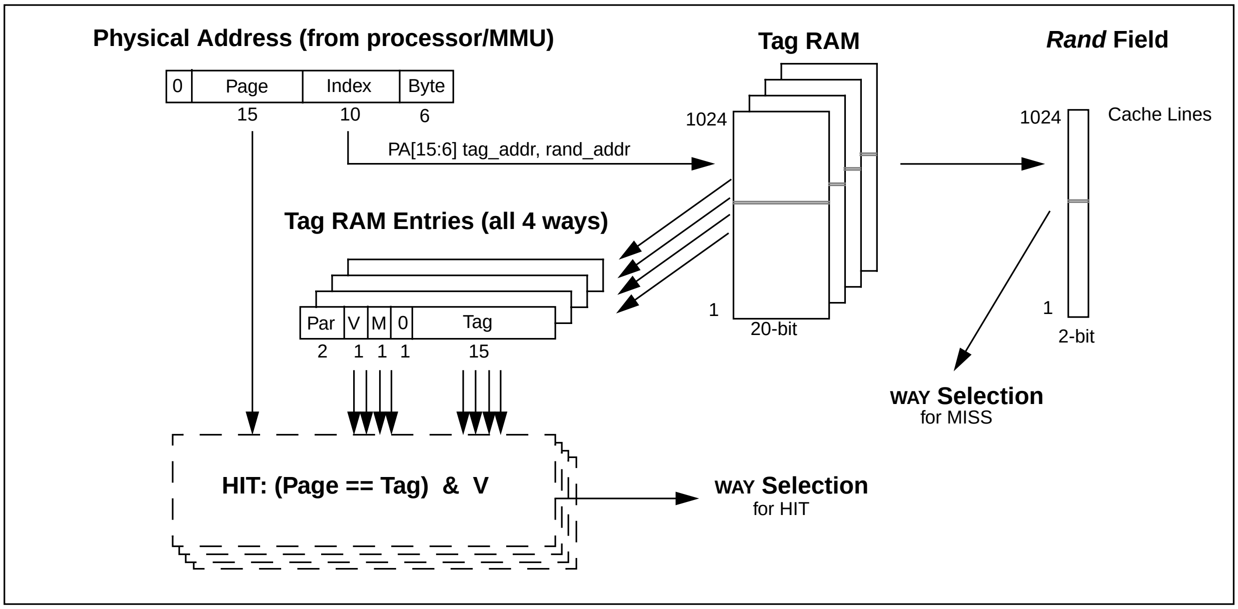


**FIGURE 3-5** 4-Way Set-Associative Cache Mode – Tag RAM Operation

### *4-Way Set-Associative Cache Line Replacement Algorithm*

In the 4-way set-associative mode, the cache line can be stored in one of four places (for example, *way* within the cache). The *Rand* value selects which way to replace when room for a new cache line is needed.

### *4-Way Set-Associative Operation of the Data RAM Array*

FIGURE 3-6 illustrates the 4-Way set-associative operation of the Data RAM access.

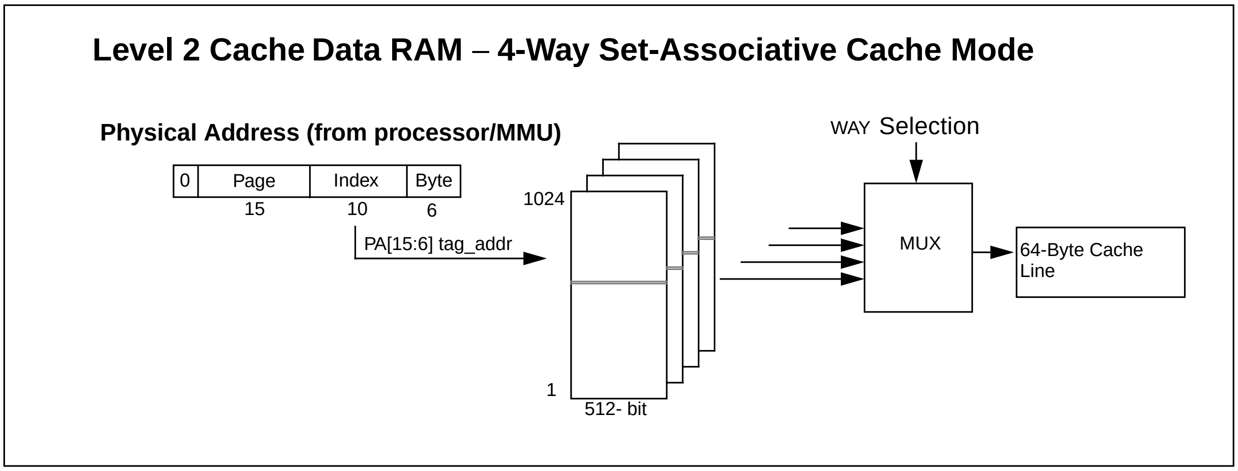


**FIGURE 3-6** 4-Way Set-Associative Cache Mode – Data RAM Access

## 3.6 Level 2 Cache Control Bits

There are two separate mode bits to control the allocation algorithm of the L2-cache. One bit provides the mode for I-cache/PDU requests. The other mode is for D-cache/LSU and PCI DMA memory requests.

The two bits allow the instruction fetches to allocate in 4-way mode while the cache allocates in direct-mapped mode for D-cache/LSU requests. This is often the case when the cache lines are being flushed.

The mode bit fields are defined in Section 3.6.1, *UPA Configuration Register,* on page 29.

### 3.6.1 UPA Configuration Register

**Compatibility Note –** The UltraSPARC IIe processor does not include a UPA bus interface. Previously unassigned bit fields in the UPA_Config Register have been assigned to control the L2-cache. Other bit fields are no longer used.

FIGURE 3-7 illustrates the UPA_Config data field.

**UPA_Config Data Field** Read/write: ASI_UPA_CONFIG_REG (0x04A), VA = 0

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 00000000000000000000000000000 | `rr` | `dm_i` | `dm_d` | `elim` | `pcon` | `mid` | `pcap` |
| 63 39 | 38 | 37 | 36 | 35 33 | 32 22 | 21 17 | 16 0 |

**FIGURE 3-7** UPA_Config Data Field

The UPA_Config register fields are described in TABLE 3-2.

**TABLE 3-2** UPA_Config Register Data Fields

| Field | Bits | Description | POR | Type | Documentation Reference |
|---|---|---|---|---|---|
| *Reserved* | 63:39 | *Reserved* | Unknown | RZ | |
| `rr` | 38 | Normally set to 0 to enable the random line replacement number for the Rand RAM array. Set to 1 to hold the number generator in its reset state. | 0 | R/W | Line Replacement Control (UPA_Config Bit 37), on page 31 |
| `dm_instruction` | 37 | Determines L2-cache line control mode for instruction misses:<br>0: 4-way set-associative<br>1: Direct-mapped | 0 | R/W | Instruction Cache/PDU Request Cache Mode (UPA_Config Bit 37), on page 30 |
| `dm_data` | 36 | Determines L2-cache line control mode for processor Load/Store misses:<br>0: 4-way set-associative<br>1: Direct-mapped | 0 | R/W | |
| `elim` | 35:33 | *Reserved*, used by previous processors. | 111 | Read Only, Writes Ignored. | UltraSPARC IIi Processor User's Manual |
| `pcon` | 32:22 | | Unknown | | |
| `mid` | 21:17 | | | | |
| `pcap` | 16:0 | | | | |

### *Instruction Cache/PDU Request Cache Mode (UPA_Config Bit 37)*

Setting the `dm_instruction` bit causes instruction fetches to use the L2-cache in a direct-mapped mode.

Direct-mapped instruction caching aids in performance modeling.

---

**Programming Note –** When switching the cache mode for I-cache requests, all instruction fetches (regular and prefetch) must occur to non-cache memory space while the effects of changing the `dm_instructions` bit takes effect.

---

### *Data Cache/LSU Request Cache Mode (UPA_Config Bit 36)*

Setting the `dm_data` bit (UPA_Config Register bit [36]) causes processor load/store operations (missed in primary cache) to use the L2-cache in a direct-mapped mode.

A direct-mapped cache provides predictable behavior and a configuration to have software flush cache lines.

---

**Programming Note –** When switching the line replacement mode for loads and stores, a `MEMBAR#Sync` instruction must be executed before and after executing the instruction that changes the operating mode of the cache. The `MEMBAR#Sync` instruction guarantees that there are no outstanding loads or stores in the L2-cache pipeline before switching cache modes.

---

### *Line Replacement Control (UPA_Config Bit 37)*

The `rr` bit is normally cleared to enable the operation of the Rand logic. The random number generator is held in its reset state until the `rr` bit is cleared. Line replacements in the L2-cache with the `rr` bit asserted will be done to the 0x01 way.

When `rr` is cleared, 0x01 is the first number written to the Rand array on the first cache line fill. After that, a new random number is loaded into the Rand field of the cache line tag after each cache line fill.

---

**Note –** A `MEMBAR#Sync` instruction must be executed before and after the setting of this bit.

---

## 3.7 Level 2 Cache Flush Procedure – Programming Guide

The L2-cache lines are flushed under software control. Cache flushing occurs by performing multiple load operations to each cache index in the direct-mapped mode. This is known as *displacement cache line flushing*.

The system software changes the cache to direct-mapped mode for loads and stores (`dm_data` bit) and reads all cache line offsets. This forces the cache to fill all the cache lines with new, unmodified cache lines, flushing the existing data to main memory, as needed.

To flush all the cache lines in the L2-cache to memory, use the following procedure:

- Execute the `MEMBAR#Sync` instruction.
- Do *not* execute load/store instructions. PCI DMA accesses are acceptable because they do not cause cache allocation.
- Set the `dm_data` bit (UPA_Config Register bit [36]) to put the L2-cache in direct-mapped mode.
- Execute the `MEMBAR#Sync` instruction.

Once the L2-cache is in direct mode, the software reads a range of addresses that map to the corresponding cache lines being flushed, forcing modified entries out to main memory. Software must read a range of addresses that map to the entire cache range (PA[17:0], 256 KB).

- Execute the `MEMBAR#Sync` instruction.
- Clear the `dm_data` bit (UPA_Config Register bit [36]).
- Execute the `MEMBAR#Sync` instruction.

# 3.8 Level 2 Cache Initialization – Programming Guide

L2-cache initialization is required after reset to prepare the L2-cache for operation.

The tag and data RAM memories are in an unknown state after resets. Software is responsible for initializing the tag RAM such that no collisions occur between any of the four ways.

Software uses the diagnostic registers to initialize the L2-cache.

To initialize the L2-cache, clear the tag values to zero and set both of the parity bits to a 1 (odd parity).

After initialization, the L2-cache works without the intervention of the operating system unless an error is detected or cache flushing is desired.

## 3.8.1 Error Conditions

Please refer to Section 16.6 in the UltraSPARC IIi User's Manual.

### *Parity Errors*

Please refer to Appendix A.6.3 in the UltraSPARC IIi User's Manual.

# 3.9 Level 2 Cache Control and Status Registers (CSRs)

ASI descriptors are used with 64-bit load and store instructions to address the RAM arrays. The diagnostic access request competes to get access to the L2-cache RAM arrays. The caching requests and the diagnostic access requests are arbitrated.

---

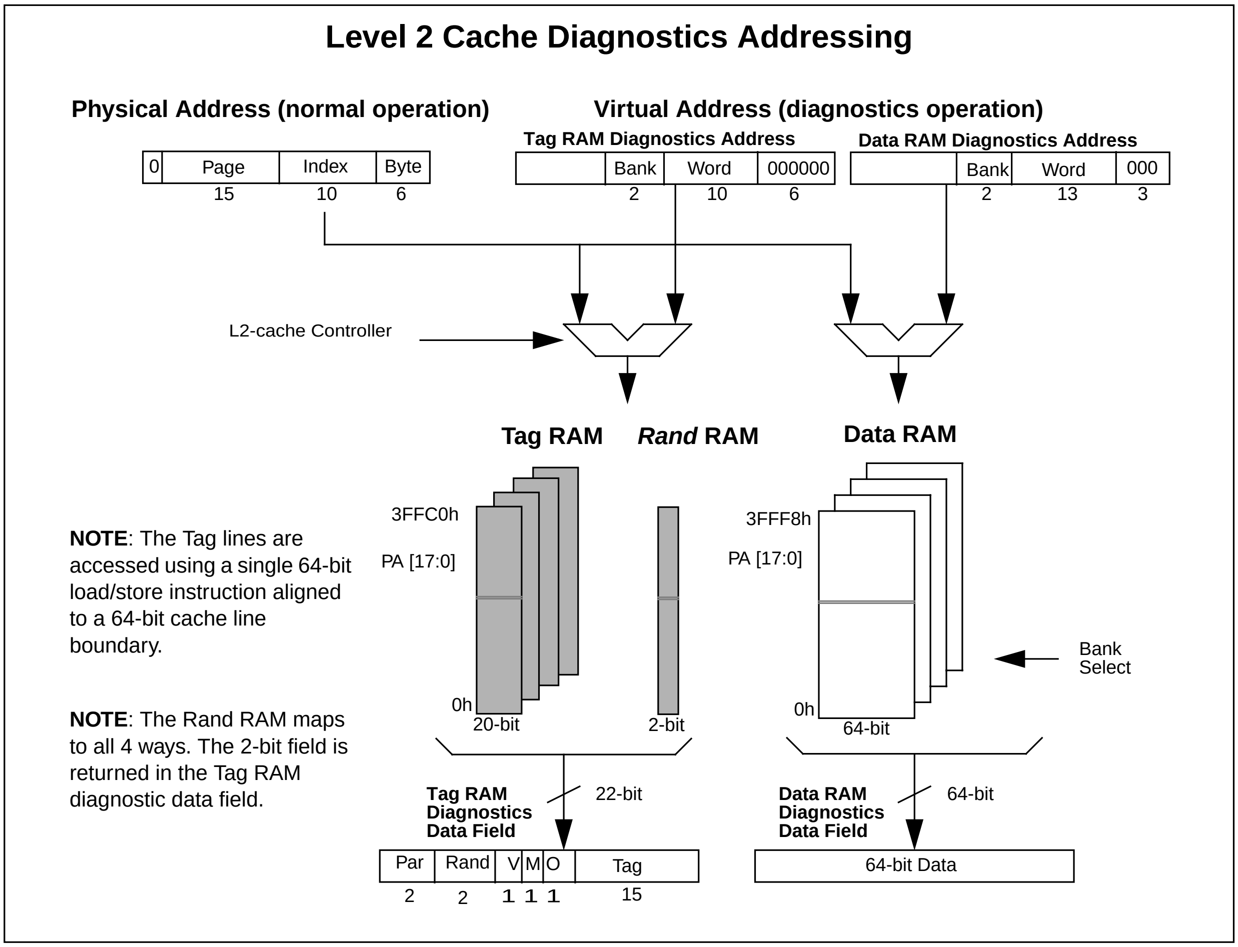
**Documentation Note –** See Appendix A of the UltraSPARC IIi User's Manual for general debug and diagnostic support. For programming guidance, see Section A.9 of the UltraSPARC IIi User's Manual which discusses the L2-cache diagnostic accesses (also known as E-cache). The UltraSPARC IIe register definitions found in this manual take precedence over those in the UltraSPARC IIi User's Manual.

---

**Programming Note –** In general, all cache activity needs to be quiescent to perform the diagnostics. Diagnostic accesses to the L2-cache should be avoided during the normal caching operation.

---

FIGURE 3-8 on page 33 shows the Level 2 Cache diagnostic addressing.



**NOTE**: The Tag lines are accessed using a single 64-bit load/store instruction aligned to a 64-bit cache line boundary.

**NOTE**: The Rand RAM maps to all 4 ways. The 2-bit field is returned in the Tag RAM diagnostic data field.

**FIGURE 3-8** Level 2 Cache Diagnostics Addressing

The L2-cache uses a delayed write buffer for both the tag and data RAM memories. If a particular index is written to and immediately read, then the read data will come from the write buffer, not the memory array. This may be important when writing a RAM diagnostic test.

## 3.9.1 Tag RAM Diagnostics Register

The Tag RAM diagnostics access will return the value of the Tag RAM line along with the associated Rand RAM entry. Since four tag RAM locations correspond to one Rand entry, the same Rand entry is returned for each of the four tag RAM accesses.

The address stepping from one 22-bit entry to the next in the Tag RAM diagnostics access is 64-bytes.

The Tag RAM entries are accessible for diagnostics read and write operations using a two step sequence which must be executed atomically.

### *Sequence to Write to the Tag RAM*

The first step is to use a 64-bit store instruction to “stage” the Tag RAM data.

- Register – Tag RAM Diagnostics Data Register

- ASI – 0x04E
- Address – 0
- Data – Tag RAM Data (see L2-cache Tag RAM Diagnostic Data Field definition)

Next, use a 64-bit store instruction to initiate the write.

- Register – Tag RAM Diagnostic Address Register
- ASI – 0x076
- Address – See L2-cache Tag RAM Diagnostic Address Register definition
- Data – Don’t care.

### *Sequence to Read from the Tag RAM*

The first step is to use a 64-bit load instruction to initiate the read of the Tag RAM.

- Register – Tag RAM Diagnostic Address Register
- ASI – 0x07E
- Address – See L2-Cache Tag RAM Diagnostic Address Register definition
- Data – Don’t care.

Next, use a 64-bit load instruction to retrieve the Tag RAM data.

- Register – Tag RAM Diagnostics Data Register
- ASI – 0x04E
- Address – 0
- Data – Tag RAM data (see L2-cache Tag RAM Diagnostic Data Field definition)

FIGURE 3-9 illustrates the Level 2 Cache Tag RAM Diagnostic Register formats.

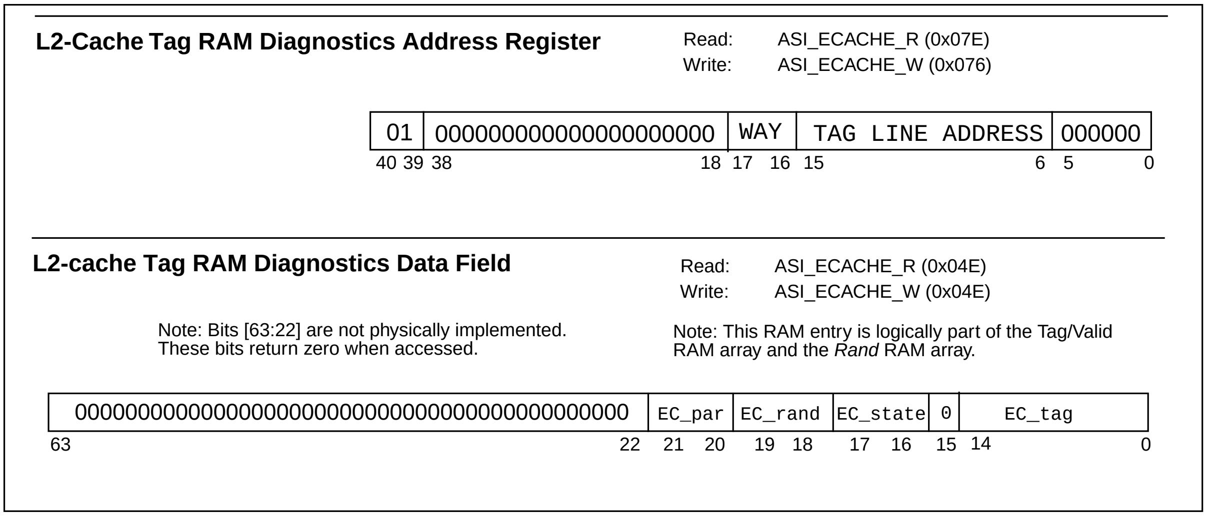


**FIGURE 3-9** Level 2 Cache Tag RAM Diagnostic Register Formats

An L2-cache Tag RAM Diagnostics Data Field is shown in TABLE 3-3.

**TABLE 3-3** L2-Cache Tag RAM Diagnostics Data Field

| Field | Bits | Description | POR | Type |
|---|---|---|---|---|
| *Reserved* | 63:22 | *Reserved* | Unknown | R/W |
| `EC_par` | 21 | `EC_state`[17:16], and `EC_tag`[15:9] Parity | Unknown | R/W |
| | 20 | `EC_tag`<8:0> Parity | Unknown | R/W |
| `EC_rand` | 19:18 | 2-bit L2-cache Rand field to support random way selection for allocation in 4-way set-associative mode | Unknown | R/W |
| `EC_state` | 17:16 | 00: Invalid Entry (line available)<br>01: *Reserved*<br>10: Exclusive (valid, unmodified)<br>11: Modified (valid, modified) | Unknown | R/W |
| `Zero` | 15 | Reads zero. | Unknown | RZ |
| `EC_tag` | 14:0 | Physical Address [30:16] tag | Unknown | R/W |

## 3.9.2 Data RAM Diagnostics Register

The Data RAM diagnostics access returns 64-bit of data based on the aligned word address. It does not return the entire cache line. The Data RAM is accessed using single load or store operations to the L2-cache Data RAM Address port.

FIGURE 3-10 illustrates the Level 2 Cache Data RAM Diagnostic Register formats.

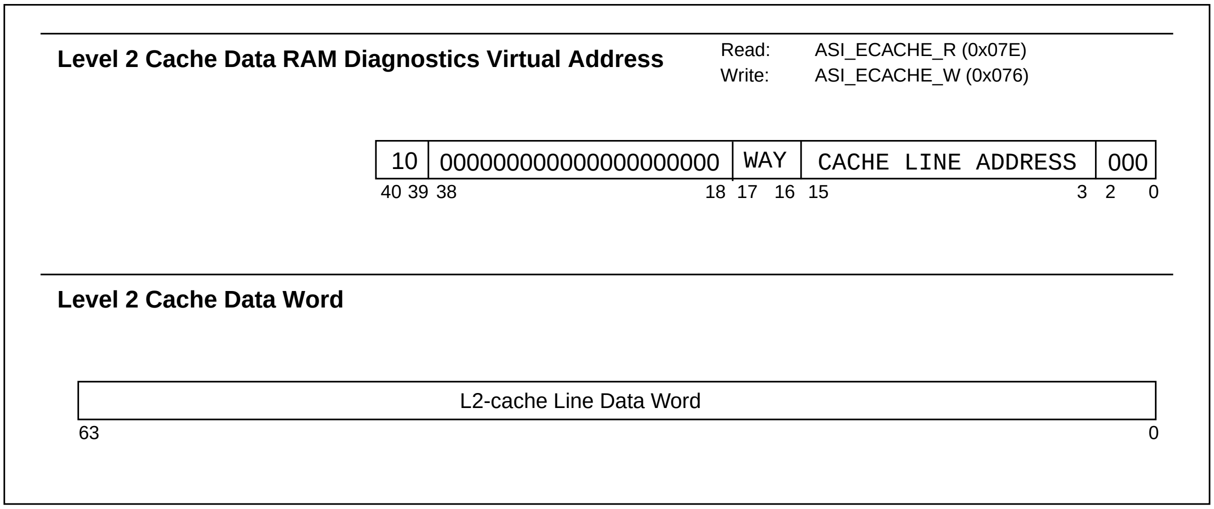


**FIGURE 3-10** Level 2 Cache Data RAM Diagnostic Register Formats

### 3.9.3 Asynchronous Fault Status Register Addendum

The L2-cache Tag Parity Syndrome bits in AFSR[17:16] are defined in the following table:

**TABLE 3-4** Level 2 Cache Asynchronous Fault Status Register (AFSR) Addendum

| L2-cache Tag Fields | Number of bits | Syndrome Bit | R/W |
|---|---|---|---|
| Tag[8:0] | 9 | AFSR[16] | R |
| Tag [17:9] + EC_state | 9 | AFSR[17] | R |

CHAPTER 4

# Memory Address Space

All transactions to the memory subsystem are handled by the Memory Interface Unit (MIU). The MIU operates directly from the processor clock. The external pins are controlled by the Memory Control Unit (MCU). The MCU operates synchronously to the processor, but at a reduced rate to match SDRAM DIMM clock rates.

The MIU manages all the requests from inside the processor and from PCI Bus Masters PCI Bus Masters accessing main memory.

**Documentation Note –** The MIU is similar to the one in the UltraSPARC IIi processor. Refer to the UltraSPARC IIi Processor User's Manual for more information.

## 4.1 Memory Interface Unit (MIU)

The Memory Interface Unit (MIU) has data and command queues and control logic to buffer memory requests from the ECU and PCI subsystem.

Data coherency is maintained by the use of address comparators in the request queues. The address of each new memory request is compared to the addresses in the queues to determine if there is a match.

FIGURE 4-1 on page 38 illustrates the memory request path of the Memory Interface Unit (MIU).

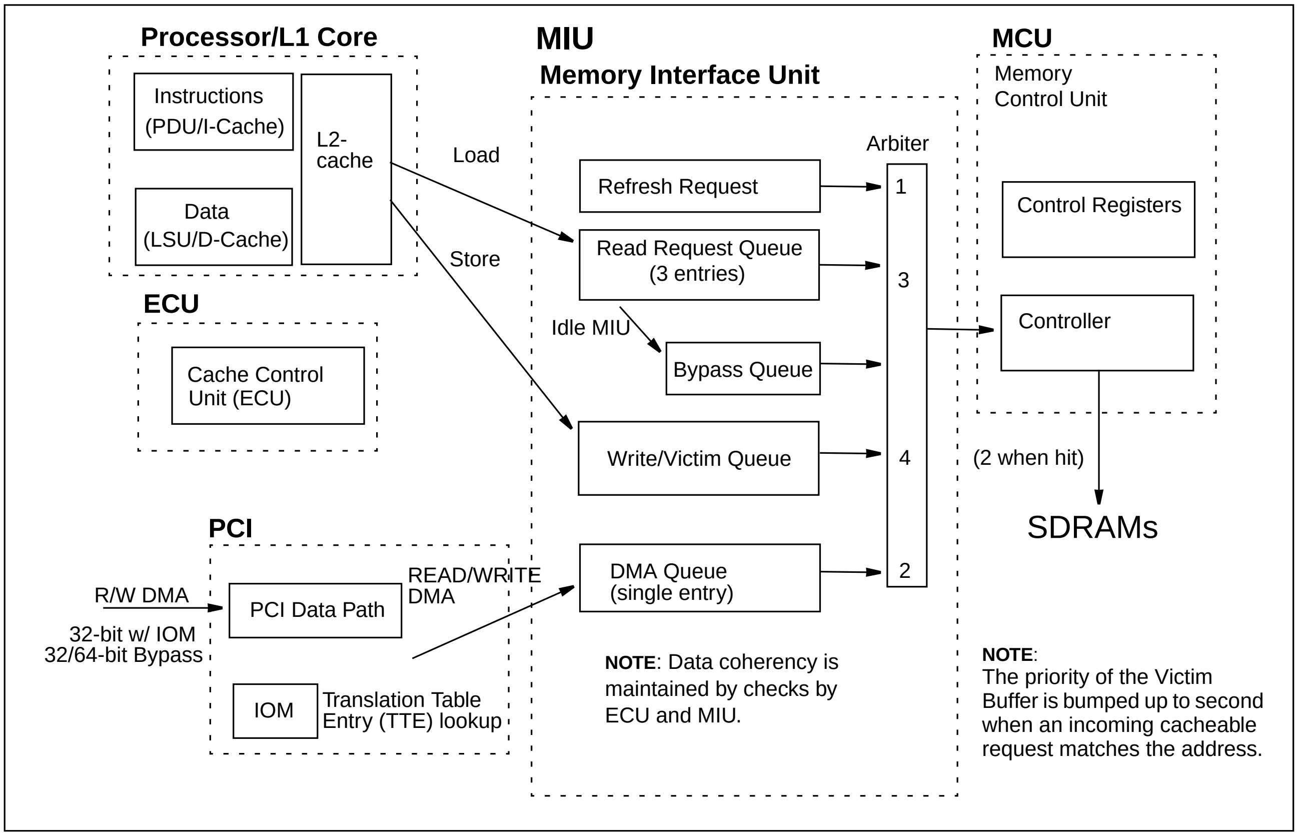


**FIGURE 4-1** Memory Request Paths

## 4.1.1 Memory Requests

The MIU accepts requests from many sources. FIGURE 4-2 illustrates MCU Memory Requests to the MIU. All these requests are cacheable.

**FIGURE 4-2** MCU Memory Requests

| Request | Source | MIU Queue | R/W | Size |
|---|---|---|---|---|
| Instruction Load | ECU | Read Request | R | 64 Bytes |
| L2-cache Line Fill | ECU | Read Request | R | 64 Bytes |
| Block Load | ECU | Read Request | R | 64 Bytes |
| L2-cache Line Flush | ECU | Write / Victim | W | 64 Bytes |
| Block Store | ECU | Write / Victim | W | 64 Bytes |
| PCI DMA Read | PCI Bus Interface | DMA | R | 64 Bytes |
| IOM Table Walk | IOM in PCI Subsystem | DMA | R | 16 Bytes |
| PCI DMA Writes A | PCI Bus Interface | DMA | W | 8 Bytes to 64 Bytes in 8-Byte Multiples |
| PCI DMA Writes B | PCI Subsystem | DMA | W | 16-Byte for DMA Writes < 8-Byte and non-8-Byte Multiples |

### *ECU Request Sources*

The ECU requests are described in Chapter 3, *Level 2 Cache Subsystem*, page 19 and in the UltraSPARC IIi User's Manual.

### *PCI DMA Request Sources*

The memory requests are generated by a PCI Bus Master attached to the PCI Host Bus Interface of the processor. The PCI subsystem buffers the requests in a command queue and presents its request to the ECU first. If needed, the request is forwarded to the main memory to complete.

### *Translation Storage Buffer Accesses from PCI Subsystem*

The PCI subsystem logic also generates a memory request to maintain the Translation Table Entries (TTE) in its I/O MMU (IOM).

## 4.2 SDRAM Memory Control Unit (MCU)

The Memory Control Unit (MCU) drives the signals to the SDRAM memories. The operation of the MCU is described in the next chapter.

## 4.3 Memory Space

The virtual address space in the UltraSPARC IIe processor has multiple physical address spaces. Three physical spaces are used to map system resources: the main memory, the PCI Bus, the internal control and status registers (CSRs), and the diagnostic registers.

---

**Documentation Note –** These sections contain new content concerning the UltraSPARC IIe processor and content from the UltraSPARC IIi Processor User's Manual. Refer to the UltraSPARC IIi Processor User's Manual for additional information.

---

### 4.3.1 Addressable Memory Space

TABLE 4-1 illustrates the Accessible Memory Space.

**TABLE 4-1** Accessible Memory Space

| Addressable Resource | Instructions | ASIs | Cacheable | Endian |
|---|---|---|---|---|
| Control, Status, Error, and Interrupt Registers (CSRs) | 64-bit Load-Store Instructions | CSR Address | Non-Cacheable | Big |
| | | Physical Address | | |
| Main Memory | Load-Store Instructions | Physical Address | Cacheable | Big |
| | | | | Little |

**TABLE 4-1** Accessible Memory Space

| Addressable Resource | Instructions | ASIs | Cacheable | Endian |
|---|---|---|---|---|
| PCI Configuration Space | Load-Store Instructions | Physical Address | Non-Cacheable | Little |
| PCI Bus I/O Space | | | | |
| PCI Bus Memory Space | | | | |

## 4.3.2 Physical Memory Space

The Physical Address (PA) selects among the main memory (SDRAM controller), the entire PCI Bus subsystem, and CSR Registers within the processor.

Cacheable memory requests (from the ECU) are sent to the memory controller. Non-cacheable requests from the processor are sent to the PCI subsystem, the Control, the Status, or the Diagnostic Registers (CSRs).

TABLE 4-2 lists the Physical Address Space.

**TABLE 4-2** Physical Address Space

| Address Range in PA[40:0] | | Destination | Size | Access Type |
|---|---|---|---|---|
| 0x000.0000.0000 | 0x000.7FFF.FFFF | SDRAM Main Memory | 2 GB | Cacheable |
| 0x000.8000.0000 | 0x000.FFFF.FFFF | *Reserved* for other processor implementations, do not use. | 2 GB | |
| 0x001.0000.0000 | 0x007.FFFF.FFFF | | – | |
| 0x008.0000.0000 | 0x1FB.FFFF.FFFF | | – | Non-Cacheable |
| 0x1FC.0000.0000 | 0x1FD.FFFF.FFFF | *Reserved*, do not use (previously UPA64S). | – | |
| 0x1FE.0000.0000 | 0x1FF.FFFF.FFFF | Processor Subsystems (PCI, memory, clock control, GP outputs, and ECU) | 8 GB | |
| 0x000.0000.0000 | 0x000.7FFF.FFFF | SDRAM Main Memory | 2 GB | Cacheable |

---

**Compatibility Note –** For compatibility with previous UltraSPARC systems, software should use PA[40:34] equal to all ‘1’s for non-cacheable space, and all ‘0’s for cacheable space. The UltraSPARC IIe processor does not detect any errors associated with using a PA[40:34] that violates this convention. The UltraSPARC IIe processor also does not detect the error of using PA[33:32] in violation of the above cacheable/non-cacheable partitioning. Consequently, all possible physical addresses decode to a destination. SDRAM accesses wrap at the 2 GB boundary.

---

## 4.3.3 I/O Subsystem Memory Map

The non-cacheable memory map for processor subsystems is shown in TABLE 4-3.

**TABLE 4-3** I/O Subsystem Address Map

| PA[40:0] | | Destination | | Description | Transaction Types |
|---|---|---|---|---|---|
| 0x1FE.0000.0000 | 0x1FE.0000.01FF | PBM | PCI Subsystem, Memory Subsystem, and miscellaneous subsystems internal to the processor. | UPA_Config, DMA Error Registers | Non-Cacheable Read and Write (64-bit only) |
| 0x1FE.0000.0200 | 0x1FE.0000.03FF | IOM | | Control and Status | |
| 0x1FE.0000.0400 | 0x1FE.0000.1FFF | PIE | | Interrupt Mapping and Clearing; Write Sync Registers | |
| 0x1FE.0000.2000 | 0x1FE.0000.5FFF | PBM | | Control, Status, and Diagnostic Registers | |
| 0x1FE.0000.6000 | 0x1FE.0000.9FFF | PIE | | | |
| 0x1FE.0000.A000 | 0x1FE.0000.A7FF | IOM | | Diagnostic Registers | |
| 0x1FE.0000.A800 | 0x1FE.0000.EFFF | PIE | | Diagnostic Registers | |
| 0x1FE.0000.F000 | 0x1FE.0000.F080 | Misc. CSRs | | See Processor Subsystems Memory Mapped CSRs. | |
| 0x1FE.0000.F088 | 0x1FE.00FF.FFFF | | | *Reserved* | |
| 0x1FE.0100.0000 | 0x1FE.0100.0041 | PBM | | PBM PCI Configuration Space Registers | Non-Cacheable Read and Write (8-, 16-, 32-, 64-bit) |
| 0x1FE.0100.0042 | 0x1FE.0100.00FF | | | *Reserved* | |
| 0x1FE.0100.0100 | 0x1FE.01FF.FFFF | PCI Bus | Type 0 and Type 1 Configuration Bus Cycles | PCI Bus Configuration Space. See PCI Configuration Cycles section. | Non-Cacheable 32-bit Writes. |
| 0x1FE.0200.0000 | 0x1FE.02FF.FFFF | PCI Bus | I/O Space | I/O Read and<br>I/O Write | Non-Cacheable Read and Write (8-, 16-, 32-, 64-bit) |
| 0x1FE.0300.0000 | 0x1FE.FFFF.FFFF | | | *Reserved* | |
| 0x1FF.0000.0000 | 0x1FF.FFFF.FFFF | PCI Bus | Memory Space | Memory Read<br>Memory Read Multiple<br>Memory Read Line<br>Memory Write<br>Memory Write<br>Memory Read | NC Read (4-byte)<br>NC Read (8-byte)<br>NC Block Read<br>NC Write<br>NC Block Write<br>NC Instruct Fetch |

## 4.3.4 I/O Programmable Registers (CSRs)

The control and status registers (CSRs) for the subsystems integrated onto the processor are listed in TABLE 4-4.

**TABLE 4-4** Processor Subsystems Memory Mapped CSRs

| Address PA[40:0] | Description | Destination | Reference |
|---|---|---|---|
| 0x1FE.0000.F000 | `FFB.Config` (no UPA64s) | L2-cache | |
| 0x1FE.0000.F008 | *Reserved* | | |
| 0x1FE.0000.F010 | `Mem_Control_0` (MC0) | MCU | |
| 0x1FE.0000.F018 | `Mem_Control_1` (MC1) | MCU | |

**TABLE 4-4** Processor Subsystems Memory Mapped CSRs *(Continued)*

| Address PA[40:0] | Description | Destination | Reference |
|---|---|---|---|
| 0x1FE.0000.F020 | Reset Control (RC) | PIE | |
| 0x1FE.0000.F028 | `Mem_Control_2` (MC2) | MCU | |
| 0x1FE.0000.F030 | `Mem_Control_3` (MC3) | MCU | |
| 0x1FE.0000.F038 | *Reserved* | | |
| 0x1FE.0000.F040 | *Reserved* | | |
| 0x1FE.0000.F048 | General Purpose Output (GPO) | GPO | |
| 0x1FE.0000.F050 | *Reserved* | | |
| 0x1FE.0000.F058 | *Reserved* | | |
| 0x1FE.0000.F060 | `Stick_Cmp_Low` | STICK | |
| 0x1FE.0000.F068 | `Stick_Cmp_High` | STICK | |
| 0x1FE.0000.F070 | `Stick_Reg_Low` | STICK | |
| 0x1FE.0000.F078 | `Stick_Reg_High` | STICK | |
| 0x1FE.0000.F080 | `E-Star_Mode` | CCU | |

## CHAPTER 5

# Memory Control Unit (MCU)

The external pins are controlled by the Memory Control Unit (MCU) and operates synchronously to the processor, but at a reduced rate to match SDRAM DIMM clock rates.

The MCU must be programmed to provide a continuous physical memory space to the processor. The Serial Presence Detection (SPD) mechanism of the SDRAM DIMMs is used by the processor to read DIMM configuration information.

**Compatibility Note –** The SDRAM controller is new to the UltraSPARC IIe processor and is not documented in previous manuals.

## 5.1 SDRAMs and DIMMs

There can be four SDRAM DIMMs ranging in size from 8 MB to 128 MB. An alternate mode for supporting DRAM with 11-bit column addressing allows four DIMMs ranging in size from 8 MB to 512 MB. Each DIMM can have two banks of SDRAMs, controlled by separate chip select signals.

Parameters that affect the address assignments of each DIMM module are DIMM size, SDRAM component configuration (x4, x8, x16), and SDRAM component capacity (16 Mb to 256 Mb). Software probes the DIMMs via the I2C bus to identify the type and size of a DIMM.

### *PC-100/133 Type DIMMs*

The SDRAM bus interface supports standard PC-100/133 type SDRAM DIMMs. The MCU is programmable to support either unbuffered or registered DIMMs. Main memory is protected by the ECC. The MCU supports up to eight physical banks (typically four dual banked DIMMs). Each bank is 72 bits wide.

The MCU uses four control registers to support the SDRAM operating parameters.

### *Buffered and Registered DIMMs*

Buffered and registered DIMMs contain PLLs that are not compatible with Energy Star (E-Star) modes because the memory clock changes frequency as E-Star modes are changed. This frequency change causes the PLLs in the DIMMs to lose synchronization in an environment where the processor may access memory at anytime, including before the time the PLL frequency locks. This may cause system failure.

Use unbuffered (no PLL) DIMMs when E-Star modes are used.

### *SDRAM Self Refresh*

Putting the memory devices in self refresh mode is accomplished by writing a one to the `Mem_Control_0` Register, `Self_Refresh` bit. When the MCU hardware state machine recognizes this bit set, the memory is put in self refresh mode by the hardware. The MCU continues to service memory requests by taking the SDRAMs out of self refresh and putting the memories back into self refresh when the MCU has no other request and the `Self_Refresh` bit is still set.

When the `Self_Refresh` bit is clear (normal mode), the MCU needs to have its `Auto_Refresh` bit enabled and have an appropriate `Refresh_Interval` value written to keep the memory refreshed. In this mode, the MCU is ready for peak performance.

### *Error Correction Code (ECC)*

In normal operation, the ECC to the SDRAM memory is enabled. The UltraSPARC IIe processor performs these functions and requires a 72-bit data path to the memory devices.

The ECC of the MCU is enabled after a Power-On Reset (POR) but the ECC trap in the processor is disabled by POR.

## 5.2 SDRAM Command Set

The memory bus interface supports the SDRAM memory commands shown in TABLE 5-1.

**TABLE 5-1** SDRAM Memory Commands Supported

| Command | Symbol |
|---|---|
| No Operation, Idle | NOP |
| Active | ACT |
| Read Select (Select Bank and Active Row) | READ |
| Write | WRITE |
| Precharging (Precharge All) | PRAL |
| Auto Refresh | ARFSH |
| Self Refresh Entry / Exit (CLKE = 0 and NOP command) | SLFRSH / SLFRSHX |
| Mode Register Set | MRS |

### SDRAM Memory Commands Not Supported

The following commands are not supported:

- Read with auto recharge
- Write with auto recharge
- Write recovering
- Write recovering with auto precharge
- Precharge Select Bank: Burst Read/Write terminate

### SDRAM MRS Field

The `MRS` field for the SDRAMs is written by the processor when the software transitions the `MRS_Initiate` bit of `Mem_Control_0` Register from a 0 to a 1. The `MRS` value is determined by hardware using the parameters previously loaded into the `Mem_Control_0` Register.

TABLE 5-2 lists the MRS Field for the SDRAMs.

**TABLE 5-2** MRS Field

| MRS Field | MRS Field Name | Source |
|---|---|---|
| `MRS`[11:7] | *Reserved* | Hardwired at 00000 |
| `MRS`[6:4] | Latency Mode | From Memory Control Register bits [3:1] |
| `MRS`[3] | Wrap Type | 0 |
| `MRS`[2:0] | Burst Length | Hardwired at 000 |

### SDRAM Operating Parameters

The UltraSPARC IIe processor supports programmable SDRAM parameters shown in TABLE 5-4 on page 47.

### SDRAM Precharge and Refresh Operations

When a memory page is accessed, it is left open (no precharge) until another page is accessed. This is done to anticipate multiple access to the same page.

When an Auto-Refresh cycle is requested, the Precharge All (PRAL) command is issued to the DIMM with the open page before the refresh cycles are initiated. The DIMMs are refreshed on consecutive clock cycles to stagger the power drain due to refresh activity. The Precharge All command is also issued to all SDRAMs before putting the SDRAMs into Self-Refresh mode.

## 5.3 DIMM Configuration

The DIMM configuration information is read over an I2C bus. The bus host controller must be supported by system logic and interface via the PCI Bus Interface.

The information from the DIMMs can be broken down into 3 groups: *addressing*, *timing*, and *number of capacitive loads*. These characteristics can be analyzed by software to set appropriate values in the memory control unit and the SDRAM mode registers.

After the DIMM configuration information reads from the DIMM over an I2C bus, the initialization firmware calculates the memory mapping and configures the `Mem_Control_1` and `Mem_Control_2` Registers.

Mixed DIMM sizes and configurations can be supported in all sockets.

The SDRAM `MEM_CS_L`[7:0] signals select up to 8 banks of physical SDRAM memory (typically 4 double-banked DIMMs). These signals are configured based on the size of the SDRAM devices (MC0), the CS Mask fields (MC1), and whether or not the DIMMs are single or double-banked. A double sided DIMM is not necessarily double-banked. These banks do not refer to the banks within the SDRAM, but instead the bank of SDRAMs on the DIMM.

The hardware attempts to create a contiguously addressable block of main memory starting with the largest DIMM capacity (irrespective of its DIMM socket position) and continuing with the next largest DIMMs, if present. A continuous memory address space is required by the processor.

### *Two Gigabyte Main Memory*

The UltraSPARC IIe processor addresses up to 2 GB of memory. This can be accomplished with any of the following configurations using 256 MB SDRAMs:

- Four 512 MB DIMMs (64 Mb x4, Single bank, 4x CS#)
- Four 512 MB DIMMs (32 Mb x8, Double bank, 8x CS#)

## 5.4 Control and Status Registers (CSRs)

The MCU is programmable via four memory control registers. These registers control operation of the MCU and provide status to the software.

A listing of MCU Control and Status Registers are shown in TABLE 5-3.

**TABLE 5-3** MCU Control and Status Registers

| Physical Address | Description | Read/Write | Size |
|---|---|---|---|
| 0x1FE_0000_F010 | Mem_Control_0 (MC0)<br>Timing and Control | R/W | 32-bit |
| 0x1FE_0000_F018 | Mem_Control_1 (MC1)<br>SDRAM Chip Select Mask | R/W | 32-bit |
| 0x1FE_0000_F028 | Mem_Control_2 (MC2)<br>Miscellaneous: SDRAM enables, DIMM present, SS/DS, and SDRAM size. | R/W | 32-bit |
| 0x1FE_0000_F030 | Mem_Control_3 (MC3)<br>I/O Buffer Strength | R/W | 32-bit |

### *Memory_Control_0 (MC0) Register: Timing and Control*

The Mem_Control_0 (MC0) Register controls SDRAM timing and functional operations. The DIMM parameters are read by software by assessing the MRS DIMM registers via an I2C bus connected to the Serial Presence Detect (SPD) mechanism of the DIMM.

TABLE 5-4 and TABLE 5-5 shows the Mem_Control_0 (MC0) Register and its bit definitions, respectively.

**TABLE 5-4** Memory_Control_0 (MC0) Register

| Register Name | Description | Register Address | POR Reset Value |
|---|---|---|---|
| Memory_Control_0 (MC0) | Timing and Control | 1FE.0000.F010 | 32’h77B0.A486 |

**TABLE 5-5** Memory_Control_0 (MC0) Register Bit Definitions

| Register Field | Symbol | Bits | Description | POR | Type |
|---|---|---|---|---|---|
| *Reserved* | | 31 | *Reserved* | 0 | R |
| `Clock_Ratio` | | 30:29 | Processor to SDRAM clock ratio:<br>00 = 4 to 1<br>01 = 5 to 1<br>10 = 6 to 1<br>11 = 7 to 1 | 11 | R/W |
| $T_{RAS}$ | RAS | 28:26 | RAS Active to Precharge Time. 3 to 6 SDRAM clocks:<br>000 = *Reserved*<br>010 = *Reserved*<br>011 = 3<br>100 = 4<br>101 = 5<br>110 = 6<br>111 = *Reserved* | 101 | R/W |
| $T_{RP}$ | RP | 25:24 | Precharge Command Period<br>2 or 3 SDRAM clocks | 11 | R/W |
| $T_{WR}$ | WR | 23:22 | Write Recovery Time<br>1 or 2 SDRAM clocks | 10 | R/W |
| $T_{RCD}$ | RCD | 21:20 | RAS to CAS Delay<br>2 or 3 SDRAM clocks | 11 | R/W |
| *Reserved* | | 19:18 | | 00 | RO |
| `DIMM_Registered` | | 17 | DIMM type:<br>0 = Unregistered<br>1 = Registered | 0 | R/W |
| `Self_Refresh` | | 16 | SDRAM Self Refresh Enable<br>0 = Disabled, 1 = Enabled | 0 | R/W |
| `Auto_Refresh` | | 15 | Enables the MCU to perform SDRAM refreshes at the specified refresh intervals | 1 | R/W |
| `Refresh_Intervals` | | 14:8 | Interval between MCU initiated refreshes. Each encoding is 64 processor clocks. The E-Star mode setting affects the processor clock frequency. | 7’h24 | R/W |
| `Enable_ECC` | | 7 | All ECC functions<br>0 = Disabled, 1 = Enabled | 1 | R/W |

**TABLE 5-5** Memory_Control_0 (MC0) Register Bit Definitions

| Register Field | Symbol | Bits | Description | POR | Type |
|---|---|---|---|---|---|
| *Reserved* | | 6:4 | | 0 | R/W |
| $T_{CL}$ | CL | 3:1 | CAS Latency.<br>010 = 2 SDRAM clocks,<br>011 = 3 SDRAM clocks,<br>All others *Reserved*. | 011 | R/W |
| `MRS_Initiate` | | 0 | Software must transition this bit from a low to a high to initiate the hardware to write the MRS value to the SDRAMs. This bit can be left a 1 or be immediately returned to a 0. | 0 | R/W |

### *Clock Ratio*

The memory clocks are derived from the processor clock and are divided down as shown in FIGURE 2-1 on page 11.

## 5.4.1 Memory_Control_1 (MC1) Register: DIMM Chip Select

The Memory_Control_1 (MC1) Register and its bit definitions are shown in TABLE 5-6 and TABLE 5-7, respectively.

**TABLE 5-6** Memory_Control_1 (MC1) Register

| Register Name | Description | Register Address | POR Value |
|---|---|---|---|
| Mem_Control_1 (MC1) | DIMM Chip Select Base Address | 1FE.0000.F010 | 0 |

**TABLE 5-7** Memory_Control_1 (MC1) Register Bit Definitions: DIMM Chip Select

| Register Field | Bits | Description | POR | Type |
|---|---|---|---|---|
| `DIMM_3_CS_Addr` | 31:24 | CS Base Address for DIMM 3 | 0x0 | R/W |
| `DIMM_2_CS_Addr` | 23:16 | CS Base Address for DIMM 2 | 0x0 | R/W |
| `DIMM_1_CS_Addr` | 15:8 | CS Base Address for DIMM 1 | 0x0 | R/W |
| `DIMM_0_CS_Addr` | 7:0 | CS Base Address for DIMM 0 | 0x0 | R/W |

### *Chip Select Base Address*

The chip base address field corresponds to the beginning address of the DIMM (even bank, when two are present). The largest DIMM is configured first followed by the others in decreasing DIMM capacity.

The `DIMM_X_CS_Addr` field corresponds to the physical address [30:23] (8 MB minimum granularity). The `DIMM_X_CS_Addr` field for the largest DIMM is zero (can be any slot).

The second largest DIMM (if present) is addressed immediately after the largest DIMM.

TABLE 5-8 lists the Memory_Control_1 (MC1) DIMM chip select base address.

Examples of the MC1 DIMM chip select base address is given in TABLE 5-9.

**TABLE 5-8** MC1 DIMM Chip Select Base Address

| Largest DIMM Size | Entry for Second Largest DIMM Size |
|---|---|
| 32 MB | 0000.0100 |
| 64 MB | 0000.1000 |
| 128 MB | 0001.0000 |
| 256 MB | 0010.0000 |
| 512 MB | 0100.0000 |

**TABLE 5-9** MC1 DIMM Chip Select Base Address – Examples

| Largest DIMM Size | Second Largest DIMM Size | Entry for DIMM Slot with Largest DIMM Size | Entry for DIMM Slot with Second Largest DIMM Size | Entry for DIMM Slot with Third Largest DIMM Size |
|---|---|---|---|---|
| 128 MB | 64 MB | 0000.0000 | 0001.0000 | 0001.1000 |
| 256 MB | 32 MB | 0000.0000 | 0010.0000 | 0010.0100 |
| 128 MB | 128 MB | 0000.0000 | 0001.0000 | 0010.0000 |

## 5.4.2 Memory_Control_2 (MC2) Register: Miscellaneous

The Memory_Control_2 (MC2) Register and its bit definitions are illustrated inTABLE 5-10 and TABLE 5-11, respectively.

**TABLE 5-10** Memory_Control_2 (MC2) Register

| Register Name | Description | Register Address | POR Value |
|---|---|---|---|
| Mem_Control_2 (MC2) | Miscellaneous DIMM controls | 1FE.0000.F018 | 32’b0 |

**TABLE 5-11** Memory_Control_2 (MC2) Register Bit Definitions: Miscellaneous

| Register Field | Bits | Description | POR | Type |
|---|---|---|---|---|
| *Reserved* | 31:28 | *Reserved* | 0x0 | |
| `DIMM_3_SCLK_Enable` | 27 | Enable `MEM_SCLK` 3, 7 to operate.<br>0 = Disabled, no activity<br>1 = Enabled, clock is active | 0 | R/W |
| `DIMM_2_SCLK_Enable` | 26 | Enable `MEM_SCLK` 2, 6 to operate | 0 | R/W |
| `DIMM_1_SCLK_Enable` | 25 | Enable `MEM_SCLK` 1, 5 to operate | 0 | R/W |
| `DIMM_0_SCLK_Enable` | 24 | Enable `MEM_SCLK` 0, 4 to operate | 0 | R/W |
| `DIMM_3_Present` | 23 | Occupied DIMM slot 3<br>0 = Empty, 1 = Populated | 0 | R/W |
| `DIMM_2_Present` | 22 | Occupied DIMM slot 2 | 0 | R/W |
| `DIMM_1_Present` | 21 | Occupied DIMM slot 1 | 0 | R/W |
| `DIMM_0_Present` | 20 | Occupied DIMM slot 0 | 0 | R/W |

**TABLE 5-11** Memory_Control_2 (MC2) Register Bit Definitions: Miscellaneous*(Continued)*

| Register Field | Bits | Description | POR | Type |
|---|---|---|---|---|
| `DIMM_3_Double` | 19 | Double Sided DIMM in slot 3 has two physical banks of SDRAMs on it:<br>0 = Single Sided (banked) DIMM<br>1 = Double Sided (banked) DIMM | 0 | R/W |
| `DIMM_2_Double` | 18 | Double Sided DIMM in slot 2 | 0 | R/W |
| `DIMM_1_Double` | 17 | Double Sided DIMM in slot 1 | 0 | R/W |
| `DIMM_0_Double` | 16 | Double Sided DIMM in slot 0 | 0 | R/W |
| `DIMM_3_SDRAM_Size`[1] | 15:14 | Size of SDRAM devices on DIMM 3<br>TOTAL SIZE:<br>00xxh = 16 Mb<br>01xxh = 64 Mb<br>10xxh = 128 Mb<br>11xxh = 256 Mb | 0x0 | R/W |
| | 13:12 | WIDTH:<br>xx00h = *Reserved*<br>xx01h = by 16 bits<br>xx10h = by 8 bits<br>xx11h = by 4 bits | 0x0 | R/W |
| `DIMM_2_SDRAM_Size` | 11:8 | Size of SDRAM devices on DIMM 2 | 0 | R/W |
| `DIMM_1_SDRAM_Size` | 7:4 | Size of SDRAM devices on DIMM 1 | 0 | R/W |
| `DIMM_0_SDRAM_Size` | 3:0 | Size of SDRAM devices on DIMM 0 | 0 | R/W |

1. The SDRAM size does not convey any information about the DIMM sizes. SDRAM size refers to the size and organization of the SDRAM devices used on the DIMM.

## 5.4.3 Mem_Control_3 (MC3) Register: I/O Buffer Strength

TABLE 5-12 lists the Memory_Control_3 (MC3) Register. TABLE 5-13 shows the signal grouping and defines the I/O buffer strength bit definitions.

**TABLE 5-12** Memory_Control_3 (MC3) Register Address

| Name | Description | Register Address | POR Value |
|---|---|---|---|
| Mem_control_3 (MC3) | I/O Buffer DC Current Strength | 1FE.0000.F020 | 10’b0 |

**TABLE 5-13** Memory_Control_3 (MC3) Register Bit Definitions: I/O Buffer Strength

| Field | Bits | Description | Function | POR | Type |
|---|---|---|---|---|---|
| *Reserved* | 31:10 | *Reserved* | *Reserved* | 0x0 | |
| `Mem_Cntl_3_Buffer` | 9 | I/O buffer strengths:<br>`MEM_CLKE`[3],<br>`MEM_RAS_L`[3],<br>`MEM_CAS_L`[3],<br>`MEM_WE_L`[3]. | 0: Low,<br>1: High | 0 | R/W |
| `Mem_Cntl_2_Buffer` | 8 | I/O buffer strengths:<br>`MEM_CLKE`[2],<br>`MEM_RAS_L`[2],<br>`MEM_CAS_L`[2],<br>`MEM_WE_L`[2]. | 0: Low,<br>1: High | 0 | R/W |

**TABLE 5-13** Memory_Control_3 (MC3) Register Bit Definitions: I/O Buffer Strength*(Continued)*

| Field | Bits | Description | Function | POR | Type |
|---|---|---|---|---|---|
| `Mem_Addr_A_Buffer` | 7:6 | I/O buffer strengths:<br>`MEM_ADDR_A` bus,<br>`MEM_BA_A[1:0]` | 00: Low,<br>01: Medium High,<br>10: Medium Low,<br>11: High | 0 | R/W |
| `Mem_Addr_B_Buffer` | 5:4 | I/O buffer strengths:<br>`MEM_ADDR_B` bus,<br>`MEM_BA_B[1:0]` | 00: Low,<br>01: Medium High,<br>10: Medium Low,<br>11: High | 0 | R/W |
| `Mem_Cntl_1_Buffer` | 3 | I/O buffer strengths:<br>`MEM_CLKE[1]`,<br>`MEM_RAS_L[1]`,<br>`MEM_CAS_L[1]`,<br>`MEM_WE_L[1]`. | 0: Low,<br>1: High | 0 | R/W |
| `Mem_Cntl_0_Buffer` | 2 | I/O buffer strengths:<br>`MEM_CLKE[0]`,<br>`MEM_RAS_L[0]`,<br>`MEM_CAS_L[0]`,<br>`MEM_WE_L[0]`. | 0: Low,<br>1: High | 0 | R/W |
| `Mem_SCLK_Buffer` | 1 | I/O buffer strength:<br>`MEM_SCLK[7:0]` | 0: Low,<br>1: High | 0 | R/W |
| `Mem_Data_ECC_Buffer` | 0 | I/O buffer strengths (writes):<br>`MEM_DATA[63:0]`<br>`MEM_ECC[7:0]` | 0: Low,<br>1: High | 0 | R/W |

### *Programmable I/O Buffer Strength*

The DC current strength of the I/O signal buffers are programmed to match the requirements of the DIMMs that are installed in the system. DIMM configuration information is read by the processor using an I2C bus to calculate capacitive loading on the memory control signals. The DC current strength specifications for the I/O buffers are included in the data sheet.

## 5.5 Physical Address Mapping of DIMMs

The highest address bit generated by the UltraSPARC IIe processor is bit 30. The 31 bits provide a total addressing range of 2 GB. Notice that the highest 3 address bits are for the 8 chip selects, except in the 64 MB x4 case where address bit 28 is used for the internal bank address and only the four even chip selects (`MEM_CS_L[0, 2, 4, 6]`) are supported.

The assignment of address bits to SDRAM signals depends on the DIMM configuration and is programmable for each individual DIMM. Address bits 23 through 28 may be assigned to a Row, Column, Internal Bank or External Bank address. Different DIMMs have different assignments. It is possible, for example, that address bit 28 can be used as an Internal Bank select for one DIMM and as a physical bank select for another DIMM in the same system.

TABLE 5-14 outlines the Bank and Row/Column SDRAM Address Multiplexing Schemes for various DIMM configurations. The first columns specify the corresponding pins on SDRAM DIMM. This table shows flexible support. Some manufactures uses x16 components versus x8 components for the same size DIMM (for example, 32 MB). The configuration register of the DIMM is read by software to program the memory controller.

## *SDRAM Bank Addressing*

TABLE 5-14 shows the `MEM_BA` usage (during row active) for the bank selects. The `MEM_ADR` [12:0] signals are all driven; the table shows the meaningful address bits driven for the various SDRAM configurations.

## *SDRAM Row/Column Addressing*

TABLE 5-14 shows SDRAM Row/Column address multiplexing. TABLE 5-15 provides a legend identifying the meaning of the shade usage for various SDRAM device widths. Notice that x4 SDRAMs use all address bits listed.

**TABLE 5-14** SDRAM Row/Column Address Multiplexing

| **DIMM Pin Number** | **Signal Name** | **16 Mb** | | **64 Mb** | | **128 Mb** | | **256 Mb** | |
|---|---|---|---|---|---|---|---|---|---|
| | | **ROW** | **COL** | **ROW** | **COL** | **ROW** | **COL** | **ROW** | **COL** |
| Number Of Banks in SDRAM | | 2 | | 4 | | 4 | | 4 | |
| 39 | `BA1` | | | A24 | | A24 | | A24 | |
| 122 | `BA0` | A22 | | A23 | | A23 | | A25 | |
| 126 | `MEM_ADDR[12]` | | | | | | | A23 | |
| 123 | `MEM_ADDR[11]` | | | A22 | | A22 | A27 | A22 | A28 |
| 38 | `MEM_ADDR[10]` | A21 | 0 | A21 | 0 | A21 | 0 | A21 | 0 |
| 121 | `MEM_ADDR[9]` | A20 | A24 | A20 | A26 | A20 | A26 | A20 | A27 |
| 37 | `MEM_ADDR[8]` | A19 | A23 | A19 | A25 | A19 | A25 | A19 | A26 |
| 120 | `MEM_ADDR[7]` | A18 | A10 | A18 | A10 | A18 | A10 | A18 | A10 |
| 36 | `MEM_ADDR[6]` | A17 | A9 | A17 | A9 | A17 | A9 | A17 | A9 |
| 119 | `MEM_ADDR[5]` | A16 | A8 | A16 | A8 | A16 | A8 | A16 | A8 |
| 35 | `MEM_ADDR[4]` | A15 | A7 | A15 | A7 | A15 | A7 | A15 | A7 |
| 118 | `MEM_ADDR[3]` | A14 | A6 | A14 | A6 | A14 | A6 | A14 | A6 |
| 34 | `MEM_ADDR[2]` | A13 | A5 | A13 | A5 | A13 | A5 | A13 | A5 |
| 117 | `MEM_ADDR[1]` | A12 | A4 | A12 | A4 | A12 | A4 | A12 | A4 |
| 33 | `MEM_ADDR[0]` | A11 | A3 | A11 | A3 | A11 | A3 | A11 | A3 |

**TABLE 5-15** Address Bit Usage

| | **x16,x8, x4 SDRAM** | **x8, x4 SDRAM** | **x 4 SDRAM Only** |
|---|---|---|---|
| Shading | ▓ | ░ | □ |

An example of an address field using 128 Mb SDRAMs on double-banked DIMM is illustrated in FIGURE 5-1.

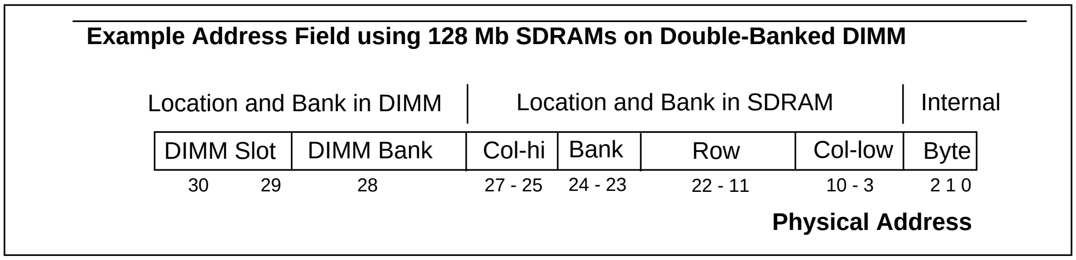


**FIGURE 5-1** Example Address Field Using 128 Mb SDRAMs on Double–Banked DIMM

## *SDRAM Parameters for DIMM Configurations*

TABLE 5-16 shows the mapping of `dev_size` to `cs_mask` and size of one side of a DIMM using such SDRAM device. The index field of each DIMM from MCU CSR represents PA[30:23] of the starting location of that DIMM. If the DIMM is double-sided, the processor determines the offset of the starting location of the second side, and toggles the corresponding bit in the index to generate PA bit [30:23] of the second side.

**TABLE 5-16** SDRAM Parameters for DIMM Configurations

| Base SDRAM Device Configuration | | Number of Devices per Side | Capacity per Side | Mem_Control_2: SDRAM_X_Size | |
|---|---|---|---|---|---|
| | | | | [15:14] | [13:12] |
| 16 Mb | 1 Mx16 | 5 | 8 MB | 00 | 01 |
| | 2 Mx8 | 9 | 16 MB | 00 | 10 |
| | 4 Mx4 | 18 | 32 MB | 00 | 11 |
| 64 Mb | 4 Mx16 | 5 | 32 MB | 01 | 01 |
| | 8 Mx8 | 9 | 64 MB | 01 | 10 |
| | 16 Mx4 | 18 | 128 MB | 01 | 11 |
| 128 Mb | 8 Mx16 | 5 | 64 MB | 10 | 01 |
| | 16 Mx8 | 9 | 128 MB | 10 | 10 |
| | 32 Mx4 | 18 | 256 MB | 10 | 11 |
| 256 Mb | 16 Mx16 | 5 | 128 MB | 11 | 01 |
| | 32 Mx8 | 9 | 256 MB | 11 | 10 |
| | 64 Mx4 | 18 | 512 MB | 11 | 11 |

# Index