OpenSPARC™ T2 Processor Megacell Specification

Sun Microsystems, Inc.
www.sun.com

Part No. 820-2728-10
December 2007, Revision A

Submit comments about this document at: `http://www.sun.com/hwdocs/feedback`

Contents

Figures

Tables

Preface

This document provides information regarding the megacells used in the OpenSPARC™ T2 processor, which is the first chip multiprocessor that fully implements the Sun™ Throughput Computing Initiative.

How This Document Is Organized

Chapter 1 provides a summary of all the megacells in the OpenSPARC T2 design.

Chapter 2 describes the Store buffer Content-addressable Memory (SCM), one of two arrays associated with the store buffer of the load and store unit (LSU).

Chapter 3 describes the translation Lookaside Buffer (TLB), commonly used as an Instruction Translation Lookaside Buffer (I-TLB) or a Data Translation Lookaside Buffer (D-TLB) for instruction fetch and for load and store units.

Chapter 4 describes the Integer Register File (IRF), which is a 32-entry x 72-bit structure, replicated four times for each thread.

Chapter 5 describes the Memory Management Unit (MMU).

Chapter 6 describes an 8-Kbit Floating-point Register File (FRF) that contains 128 entries of 64-bit doublewords.

Chapter 7 describes a 16-Kbyte, 32-byte line size, 4-way set-associative Instruction Cache Data (ICD).

Chapter 8 describes a 128-entry, 4-way, set-associative Instruction and Data Cache Tag (ICT).

Chapter 9 describes the 9-Kbyte 4-way set associative Data Cache Array (DCA).

Chapter 10 describes the 9-Kbyte 4-way set associative Data Cache Tag Array (DTA).

Chapter 11 describes the Data Valid Bit array (DVA)..

Chapter 12 describes the 172-Kbyte, 12-way, set-associative Level 2 cache tag array (L2 Tag), which is split into four banks of 43 Kbytes each.

Chapter 13 describes Level 2 cache data array, an inclusive, 3-Mbyte cache composed of four symmetrical banks.

Chapter 14 describes the 32 x 128 Miss Buffer Data Array.

Chapter 15 describes the 32 x 40 Miss Buffer Request CAM.

Chapter 16 describes the 64 x 64 L2- Cache Instruction and Data Directory CAM.

Chapter 17 describes the 8 x 40 L2- Cache Fill Buffer.

Chapter 18 describes the 16 x 160 L2-Cache Dual Port Register File.

Chapter 19 describes the 32 x 160 dL2-Cache VUAD Register File.

Chapter 20 describes the Memory Controller Register File (MCU) 32x72.

Chapter 21 describes the Data Management Unit IOMMU Array.

Chapter 22 describes the Data Management Unit 128 x 132 DMU Register File.

Chapter 23 describes the Data Management Unit 144 x 149 DMU arrray.

Chapter 24 describes the Data Management Unit 512 x 60 DMU Register File.

Chapter 25 describes the Electronic Fuse Array (EFA), composed of an array of 64x32 poly fuses and address decoders.

Using UNIX Commands

This document might not contain information about basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices. Refer to the following for this information:

- Software documentation that you received with your system
- Solaris™ Operating System documentation, which is at:

  `http://docs.sun.com`

Shell Prompts

Shell	Prompt
C shell	*machine-name*`%`
C shell superuser	*machine-name*`#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	`#`

Typographic Conventions

Typeface*	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`% su` `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this. To delete a file, type `rm` *filename*.

* The settings on your browser might differ from these settings.

Related Documentation

The documents listed as online are available at:

`http://www.opensparc.net`

Application	Title	Part Number	Format	Location
Documentation	*OpenSPARC T2 Core Microarchitecture Specification*	820-2545-11	PDF	Online
Documentation	*OpenSPARC T2 System-On-Chip (SoC) Microarchitecture Specification*	820-2620-11	PDF	Online
Documentation	*OpenSPARC T2 Processor Megacell Specification*	820-2728-10	PDF	Online
Documentation	*OpenSPARC T2 Processor Design and Verification User's Guide*	820-2729-10	PDF	Online

Documentation, Support, and Training

Sun Function	URL
Documentation	http://www.sun.com/documentation/
Support	http://www.sun.com/support/
Training	http://www.sun.com/training/

Third-Party Web Sites

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

`http://www.sun.com/hwdocs/feedback`

Please include the title and part number of your document with your feedback:

OpenSPARC T2 Processor Megacell Specification, part number 820-2728-10

CHAPTER 1

Megacells in the OpenSPARC T2 Processor

This chapter provides a summary of the megacells in the OpenSPARC T2 design. These megacells include various register files, translation lookaside buffers (TLBs), content-addressable memory (CAM), Level 2 cache (L2-cache), and arrays. The following chapters give detailed descriptions of each megacell in the design, including functional descriptions, block diagrams, I/O lists, and timing diagrams.

1.1 Overview

TABLE 1-1 lists the megacells in the OpenSPARC T2 design.

TABLE 1-1 Megacells in the OpenSPARC T2 Processor

Megacell Name	Chapter No.	Description
n2_stb_cm_64x45_cust	2	Store Buffer CAM (SCM)
n2_tlb_tl_128x59_cust	3	Translation Lookaside Buffer (TLB)
n2_irf_mp_128x72_cust	4	Integer Register File (IRF)
n2_mmu_cm_64x34s_cust	5	Memory Management Unit CAM (MMU)
n2_frf_mp_256x78_cust	6	Floating-point register file (FRF)
n2_icd_sp_16p5kb_cust	7	Instruction Cache (I$) Data Array (ICD)
n2_ict_sp_1920b_cust	8	Instruction Cache Tag Array (ICT)
n2_dca_sp_9kb_cust	9	Data Cache Array (DCA)
n2_dta_sp_1920b_cust	10	Data Cache Tag Array (DTA)
n2_dva_dp_32x32_cust	11	Data Valid Bit Array (DVA)

TABLE 1-1 Megacells in the OpenSPARC T2 Processor *(Continued)*

Megacell Name	Chapter No.	Description
n2_l2t_sp_28kb_cust	12	L2-Cache Tag Array
n2_l2d_sp_512kb_cust	13	L2-Cache Data Array
n2_l2t_dp_32x128_cust	14	L2-Cache Miss Buffer Data Array
n2_com_cm_32x40_cust	15	L2-Cache Miss Buffer Request CAM
n2_com_cm_64x64_cust	16	L2-Cache Instruction and Data Directory CAM
n2_com_cm_8x40_cust	17	L2-Cache Fill Buffer
n2_l2t_dp_16x160_cust	18	L2-Cache Dual Port Register File
n2_l2t_dp_32x160_cust	19	L2-Cache VUAD Register File
n2_mcu_32x72async_dp_cust	20	Memory Controller Register File
n2_iom_sp_devtsb_cust	21	Data Management Unit IOMMU Array
n2_dmu_dp_128x132s_cust	22	Data Management Unit 128x132 RegisterFile
n2_dmu_dp_144x149s_cust	23	Data Management Unit 144x149 Array
n2_dmu_dp_512x60s_cust	24	Data Management Unit 512x60 RegisterFile
n2_efa_sp_256b_cust	25	e-Fuse

CHAPTER 2

Store Buffer CAM (SCM)

This chapter describes the following topics:

2.1 Functional Description

2.1.1 General

- The Store Buffer CAM (SCM) stores the address of each pending store of the load/store unit. All loads check for an address match to see if a Read After Write (RAW) hazard occurs.
- The SCM contains 64 logical entries. This is logically divided into 8 groups (each group for a different thread) each containing 8 entries. In a lookup cycle, the CAM thread id input bus will select only one of the 8 groups.
- Each entry contains 37 bits of tag (stored address), and 8 bits of byte-mask which indicate which of the 8 bytes are actually going to be stored into memory.
- All 45 (37 tag + 8 byte-mask) of the bits in each entry are capable of being written to and read from the SCM array.
- There is 1 read and 1 write port which both share the same address input, and a CAM (lookup) port.

- A read and lookup operation are allowed to occur in the same cycle, but all other combinations are not allowed.
- Any access can occur in consecutive clock cycles.

2.1.2 Lookup Operation

- For the lookup operation, the 37 bits of tag in every entry are compared against the compare input bus (aka "key"). If at least one of the 8 byte-mask bits is set to '1' in both the entry and the key, then this is considered a "byte_match". The result of the tag compare and the byte_match results are then ANDed together. Finally these results are ANDed with the line-enables of the corresponding entries to get the CAM-hit results.
- The CAM-hit result is decoded into a 3 bit address indicating which logical entry (out of the 8 entries in the "thread group") had a hit.
- The final CAM-hit signal for the block is the OR of the CAM-hits of all the entries.
- If an entry has a CAM-hit, and there is at least one bit of it's byte-mask which is set to a '0' while the corresponding bit of the byte-mask key is set to '1', then this is considered a partial hit. The partial output signal is activated in this case.
- If more than one entry has a CAM-hit, then the multi-hit output signal is activated. In this case, the output hit address and partial output signals are meaningless.
- The CAM lookup occurs in phase 1. This is flopped outside of the block.
- In 16 byte loads, the LSB of the tag compare needs to be masked. This is controlled by the quad_ld_cam input.

2.1.3 Read Operation

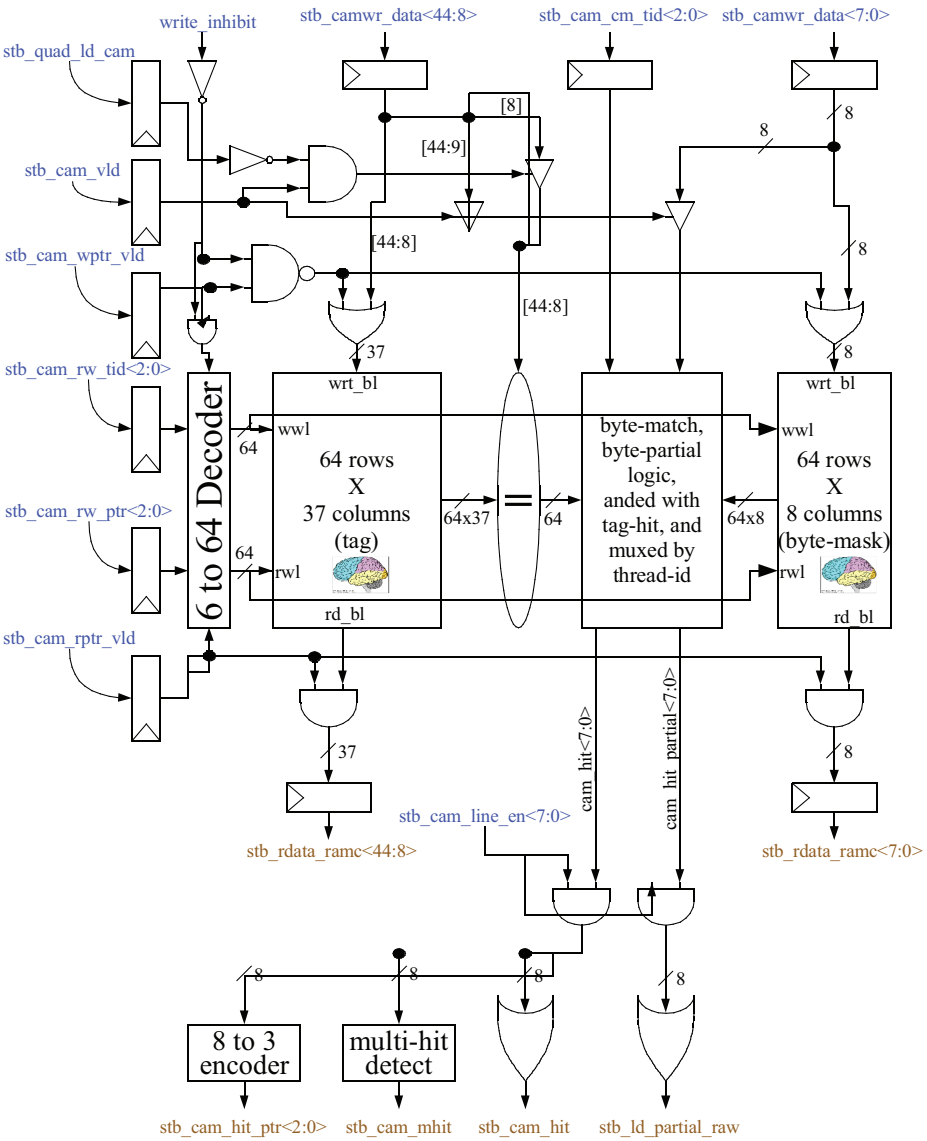
- Read cycle is a single cycle, and occurs in phase 1.
- During a read cycle, all 37 bits of tag and 8 bits of byte-mask are read, flopped and sent to the output bus.

2.1.4 Write Operation

- Write cycle is a single cycle, and occurs in phase 1.
- 37 bits of tag and 8 bits of byte-mask are provided to the block to be written, and are setup to the rising edge of clock.

2.2 Block Diagram

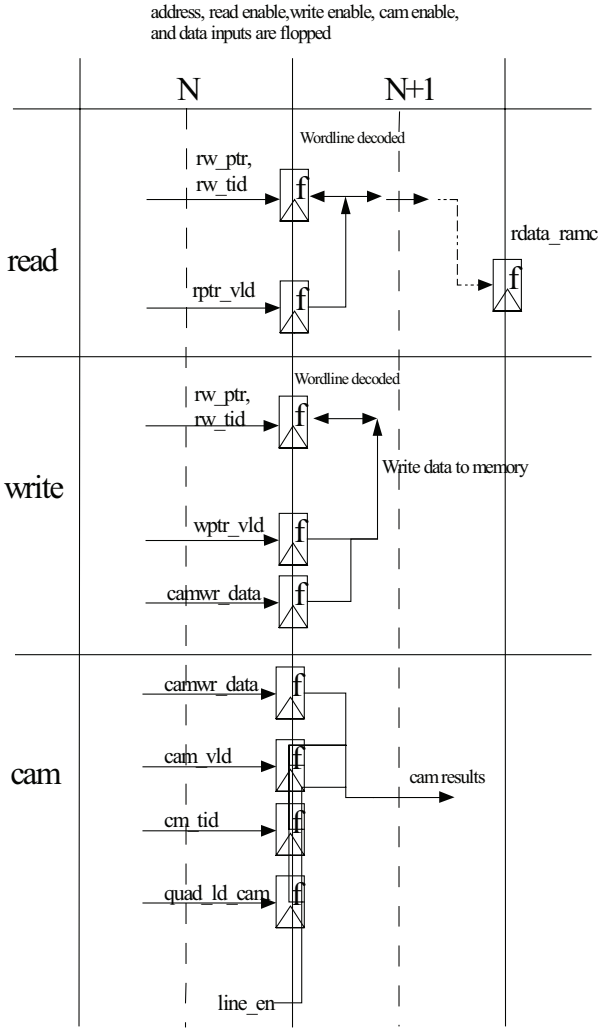
FIGURE 2-1 SCM Block Diagram



- There will be 2 physical banks of arrays. The left array will contain the 64 rows by 8 bits of byte-mask, and the right array will contain the 64 rows by 37 bits of tag. Any 8 threads of the same entry will be physically next to each other.
- There will be dummy rows above and below each portion of 16 entries.
- There will be 2 dummy edge columns at the left and right side of the tag array. There will also be 2 dummy edge columns at the left and right side of the byte-mask array.

2.3 Pipeline Diagram

FIGURE 2-2 SCM Pipeline Diagram



2.4 I/O List

The following table describes the SCM I/O signals.

TABLE 2-1 SCM I/O Signal List

Signal Name	Width	I/O	Flopped	Source/Deal	Description
stb_cam_rw_ptr	[2:0]	IN	Yes	lsu_sbc_ctl	Read/write address
stb_cam_rw_tid	[2:0]	IN	Yes	lsu_sbc_ctl	Thread ID for read/write
stb_cam_wptr_vld		IN	Yes	lsu_sbc_ctl	Write enable
stb_cam_rptr_vld		IN	Yes	lsu_sbc_ctl	Read enable
stb_cam_vld		IN	Yes		Lookup enable
stb_cam_cm_tid	[2:0]	IN	Yes		Thread ID for Lookup operation
stb_cam_line_en	[7:0]	IN	No		Line enables for Lookup results
stb_quad_ld_cam		IN	Yes		Masks LSB of tag in a lookup (for 16 byte loads)
stb_camwr_data	[44:0]	IN	Yes		Data for compare/write
stb_rdata_ramc	[44:0]	OUT	Yes		Read data
stb_ld_partial_raw		OUT	No		Partial Lookup Result
stb_cam_hit_ptr	[2:0]	OUT	No		Address location of matched entry
stb_cam_hit		OUT	No		Store buffer hit
stb_cam_mhit		OUT	No		Multiple hit
Clock and Test Related I/O					
l2clk		IN	No	clock grid	L2 clock input
scan_in		IN	No		Scan input
tcu_pce_ov		IN	No	tcu	Test controller input
tcu_aclk		IN	No	tcu	Test controller scan-in clock
tcu_bclk		IN	No	tcu	Test controller scan-out clock
pce		IN	No	tied high	Clock enable.
tcu_array_wr_inhibit		IN	No	tcu	Disables writes during test modes
tcu_se_scancollar_in		IN	No	tcu	Scan-enable for input flops/latches

TABLE 2-1 SCM I/O Signal List *(Continued)*

Signal Name	Width	I/O	Flopped	Source/Deal	Description
tcu_se_scancollar_out		IN	No	tcu	Scan-enable for output flops
tcu_scan_en		IN	No	tcu	Scan-enable for internal clocking
scan_out		OUT	No		Scan output

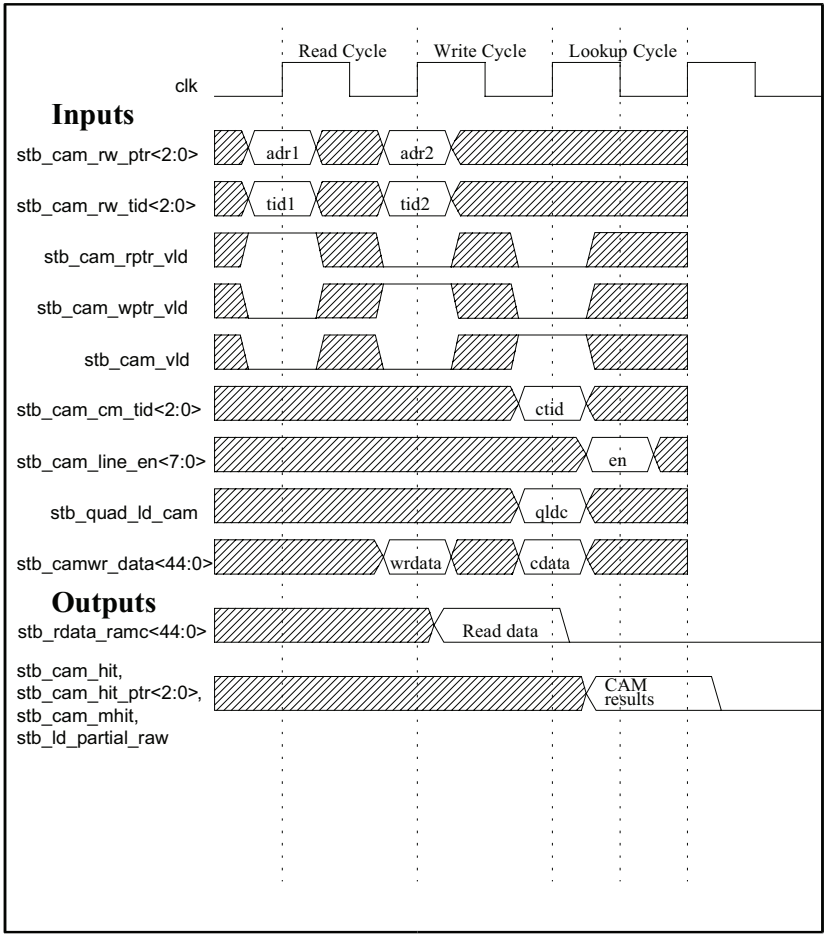
2.4.1 Modes of Operation

TABLE 2-2 SCM Modes of Operation

Enable Values			Memory Operation	Output Results				
read	write	lookup		ramc <44:0>	partial	hit_ptr <2:0>	cam_hit	cam_mhit
1	0	0	Read Cycle	results of read	0	0	0	0
0	1	0	Write Cycle	0	0	0	0	0
0	0	1	Lookup Cycle	0	results of lookup			
1	0	1	Read and Lookup Cycle	results of read	results of lookup			
0	0	0	No Operation	0	0	0	0	0
1	1	0	Monitors in RTL do not allow these scenarios. Memory contents and outputs may be indeterminate.					
0	1	1						
1	1	1						

2.5 Timing Diagram

FIGURE 2-3 SCM Timing Diagram



CHAPTER 3

Translation Lookaside Buffer (TLB)

This chapter describes the following topics:

3.1 Functional Description

In today's computers most programs running on a machine assume that they have the entire memory of the system to themselves. These programs utilize a virtual memory abstraction which requires that the virtual addresses must be converted to real, physical addresses before the physical memory can be actually accessed. The virtual to physical address translation is performed in the CPU by the Translation Lookaside Buffer (TLB). The TLB does not map individual memory locations but instead maps larger regions of memory (pages). In the following paragraphs a description of the Data TLB (DTLB) and Instruction TLB (ITLB) supported on OpenSPARC T2 and its enhancements and differences with respect to its first generation implementation are provided.

3.1.1 OpenSPARC T1 vs OpenSPARC T2

- OpenSPARC T2 has a 128-entry Data TLB (DTLB), while OpenSPARC T1 has a 64-entry DTLB. The Instruction TLB (ITLB) is 64-entry and remains the same.

- The instruction cache is 8-way set associative in OpenSPARC T2 as compared to 4-way in OpenSPARC T1. Hence, 8 cache ptag comparators are required in the ITLB. The data cache is 4-way set associative in OpenSPARC T2. Hence, 4 cache ptag comparators are required in DTLB.
- OpenSPARC T2 supports a pair of primary context registers and a pair of secondary context registers. OpenSPARC T1 supports a single primary context and single secondary context register. Multiple primary and secondary contexts permit different processes to share TTEs within the TLBs. To maintain backwards compatibility with software designed for a single primary and single secondary context register (OpenSPARC T1, writes to primary (secondary) context 0 register also update primary (secondary) context 1 register.
- OpenSPARC T2 does not support a locked bit in the TLBs. OpenSPARC T1 supports a locked bit in the TLBs.
- OpenSPARC T1 and OpenSPARC T2 support the same four page sizes (8 kB, 64 kB, 4 MB, 256 MB).
- OpenSPARC T2 adds a demap real operation, which demaps all pages with R=1 from the TLB.
- Autodemapping of pages in the TLBs only demaps pages of the same size or of a larger size in OpenSPARC T2. In OpenSPARC T1, autodemap demaps pages of the same size, larger size, or smaller size.
- OpenSPARC T1 and OpenSPARC T2 use the same used bit-based pseudo-LRU TLB replacement scheme.
- OpenSPARC T2 supports detection of multiple hits in the TLBs.

3.1.2 DTLB Description

- The DTLB is divided into a tag array (128x66 entries) and a data array (128x38 entries)
- The DTLB has a separate Used bit array, Valid bit array and Real bit array.
- The DTLB has a 128-bit multi-match detect & priority encoder and 4-way comparators.
- Clock headers are distributed along the width of the block.

3.1.3 ITLB Description

- The ITLB is divided into a tag array (64x66 entries) and a data array (64x38 entries)
- The ITLB has a separate Used bit array, Valid bit array and Real bit array.

- The ITLB has a 64-bit multi-match detect & priority encoder and 8-way comparators.
- Clock headers are distributed along the width.

3.2 Block Diagrams

FIGURE 3-1 DTLB Block Diagram

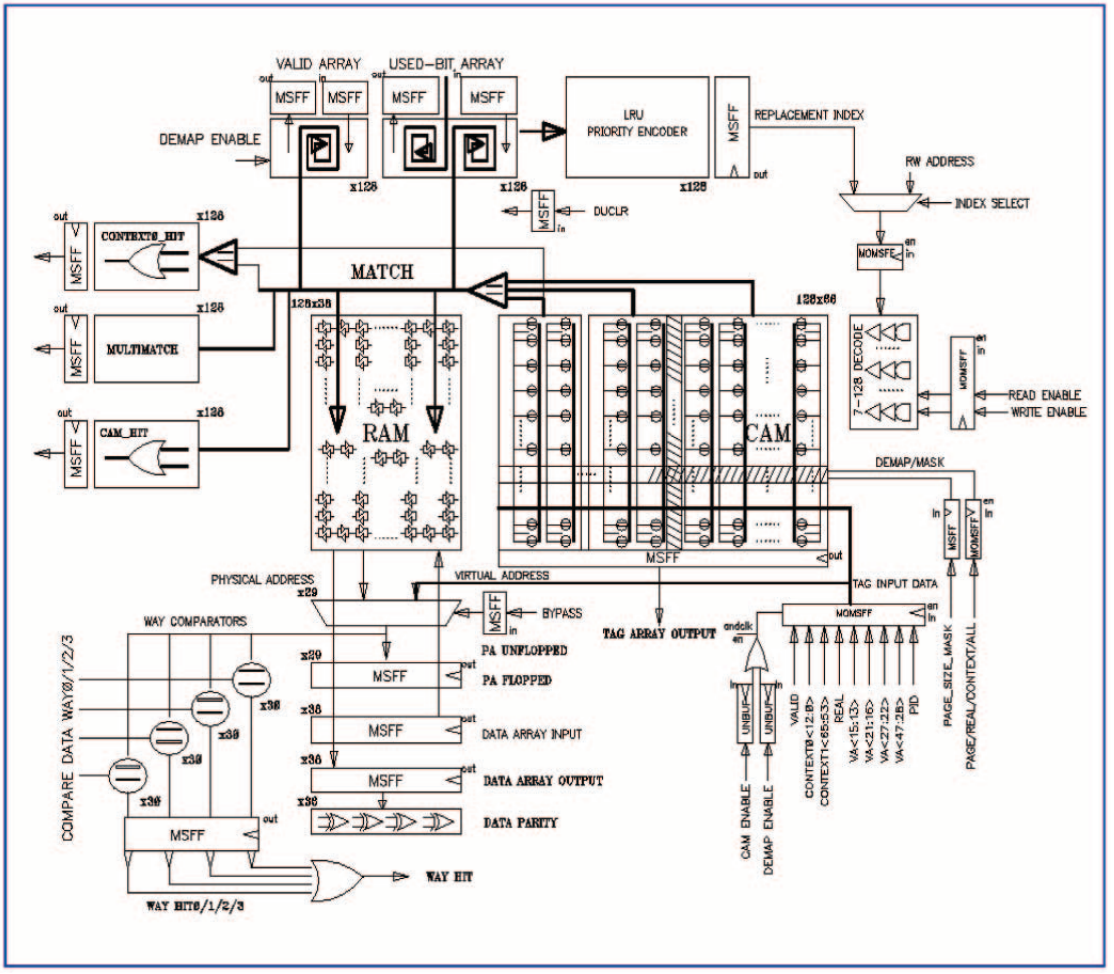
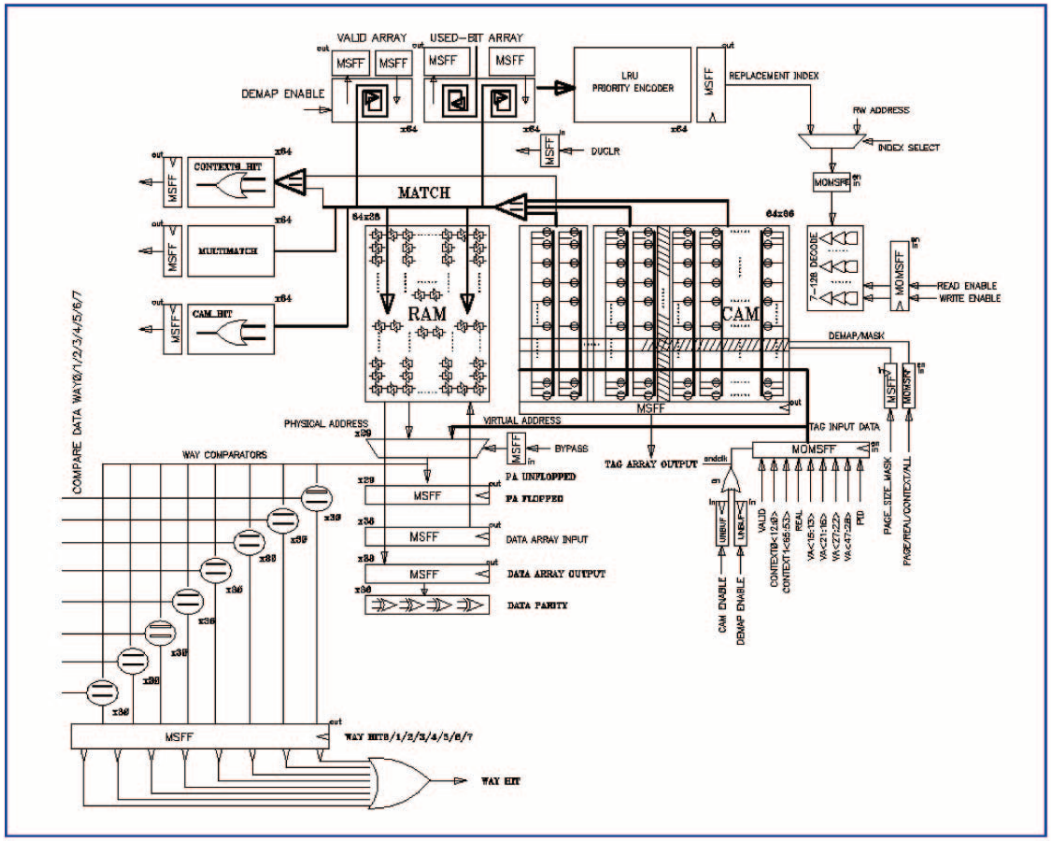


FIGURE 3-2 ITLB Block Diagram



3.3 I/O List

The following table describes the TLB I/O signals.

TABLE 3-1 TLB I/O Signal List

Port Name	Type	UN/FF/LAT	Width	Function
		ADDRESS		
tlb_rw_index	input	flopped	7	read/write address (DTLB)
tlb_rw_index	input	flopped	6	read/write address (ITLB)
		DATA		
tte_tag	input	flopped/NAN	66	D/I TLB tag array write/compare data
tte_data	input	flopped	38	D/I TLB data arraywrite data
tlb_tte_tag	output	flopped	66	D/I TLB tag array read data
tlb_tte_data	output	flopped	38	D/I TLB data array read data
tte_ubit	input	flopped	1	D/I TLB used bit
tlb_tte_u_bit	output	flopped	1	D/I TLB used bit
tlb_va	input	flopped	2	D/I TLB incoming virtual address
tlb_pgnum	output	flopped	27	D/I TLB physical address
tlb_pgnum_crit	output	unflopped	27	D/I TLB physicall address
		CACHE TAG COMPARE DATA		
cache_ptag_w0	input	unflopped	30	D/I TLB Way 0 compare data
cache_ptag_w1	input	unflopped	30	D/I TLB Way 1 compare data
cache_ptag_w2	input	unflopped	30	D/I TLB Way 2 compare data
cache_ptag_w3	input	unflopped	30	D/I TLB Way 3 compare data
cache_ptag_w4	input	unflopped	30	ITLB Way 4 compare data
cache_ptag_w5	input	unflopped	30	ITLB Way 5 compare data
cache_ptag_w6	input	unflopped	30	ITLB Way 6 compare data
		CONTROL		
tlb_cam_vld	input	flopped/NAN	1	D/I TLB compare enable
tlb_demap	input	flopped/NAN	1	D/I TLB demap enable

TABLE 3-1 TLB I/O Signal List *(Continued)*

Port Name	Type	UN/FF/LAT	Width	Function
tlb_demap_context	input	flopped	1	D/I TLB demap context enable
tlb_demap_all	input	flopped	1	D/I TL:B demap all enable
tlb_demap_real	input	flopped	1	D/I TL:B demap real enable
tlb_demap_page	input	flopped	1	D/I TLB demap page enable
tlb_wr_vld	input	flopped/NAN	1	D/I TLB write enable
tlb_rd_vld	input	flopped/NAN	1	D/I TLB read enable
tlb_bypass	input	flopped	1	D/I TLB bypass enable
disable_clear_ubit	input	flopped	1	D/I TLB disable U-bit clearing
tlb_rw_index_vld	input	flopped/NAN	1	D/I TLB regular/replacement select
tte_page_size_mask	input	flopped	1	D/I TLB page mask during write
MISCELLANEOUS SIGNALS				
cache_set_vld	input	unflopped	4	D/I TLB compare enable
cache_way_hit	output	flopped	4	D/I TLB demap enable
cache_hit	output	ff (delayed)	1	D/I TLB demap context enable
tlb_cam_hit	output	flopped	1	D/I TLB 128-bit OR; compare hit
tlb_context0_hit	output	flopped	1	D/I TLB 128-bit OR; context0 hit
tlb_tte_data_parity	output	ff (delayed)	1	D/I TLB data array parity
CLOCK RELATED CONTROLS				
l2clk	input	x	1	D/I TLB global clock
scan_in	input	x	1	D/I TLB scan input
scan_out	output	x	1	D/I TLB scan output
tcu_aclk	input	x	1	D/I TLB a-phase scan input clock
tcu_bclk	input	x	1	D/I TLB b-phase scan input clock
tcu_se_scancollar_in	input	x	1	D/I TLB input clock header control
tcu_se_scancollar_ou	input	x	1	D/I TLB output clock header control
tcu_array_wr_inhibit	input	x	1	D/I TLB write inhibit
tcu_clk_stop	input	x	1	D/I TLB disable clock header control
tcu_pce_ov	input	x	1	D/ITLB test controller input

3.3.1 IDTLB/ITLB Modes of Operation

The TLB supports the following operations per clock cycle:

3.3.1.1 Primary Operations

- Read (tlb_rd_vld)
- Compare (tlb_cam_vld)
- Write (tlb_wr_vld)
- Demap Page (tlb_demap)
- Demap Context (tlb_demap & tlb_demap_context)
- Demap Real (tlb_demap & tlb_demap_real)
- Demap All (tlb_demap & tlb_demap_all)
- Bypass (tlb_bypass)

3.3.1.2 Secondary Operations

- Used Bit Setting
- Used Bit Resetting
- Replacement Index (Valid | Used) generation
- Valid Bit Resetting
- Replacement Index Write (Feedback Loop)
- Array Write Inhibit
- Multimatch Generation
- Context0 Hit Generation
- Data Parity
- TLB Cam Hit)

TABLE 3-2 Primary Control Truth Table

oper	rw_idx	rd	wr	cam	dmp	dmpal	dmpcx	dmprl	byp	duclr	wr_ih	msk	V	R	U
noop	1/0	0	0	0	0	0	0	0	0	0	1/0	0	0	0	0
read	1	1	0	0	0	0	0	0	0	x	1/0	x	1/0	1/0	1/0
write	1/0	0	1	0	0	0	0	0	0	1/0	1/0	1/0	1/0	1/0	1/0
cam	1/0	0	0	1	0	0	0	0	0	1/0	1/0	x	1/0	1/0	1/0
dmppg	1/0	0	0	0	1	0	0	0	0	x	1/0	x	1/0	1/0	1/0

TABLE 3-2 Primary Control Truth Table *(Continued)*

dmpal	1/0	0	0	0	1	1	0	0	0	x	1/0	x	1/0	1/0	1/0
dmpcx	1/0	0	0	0	1	0	1	0	0	x	1/0	x	1/0	1/0	1/0
dmprl	1/0	0	0	0	1	0	0	1	0	x	1/0	x	1/0	1/0	1/0
bypass	1/0	0	0	1/0	0	0	0	0	1	0	1/0	x	1/0	1/0	1/0

Note – *All controls are gated by tcu_array_wr_inihibit* except cam_hit output flop

TABLE 3-3 Output for Primary Operations

oper	pgcrit	pgnum	$hit	$whit	mhit	c0hit	camhit	dataout	tagout	prty	ubit	vld
noop	x0	x0	1/0	1/0	x0	x0	x0	x0	x0	0	x	x
read	1/0	1/0	1/0	1/0	0	0	0	1/0	1/0	1/0	1/0	1/0
write	0	0	1/0	0	x0	x0	x0	x0	x0	0	1/0	1/0
cam	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	x0	1/0	1/0	x
dmppg	x0	x0	1/0	1/0	x0	1/0	x0	x0	x0	0	x	0
dmpal	x0	x0	1/0	1/0	x0	1/0	x0	x0	x0	0	x	0
dmpcx	x0	x0	1/0	1/0	x0	1/0	x0	x0	x0	0	x	0
dmprl	x0	x0	1/0	1/0	x0	1/0	x0	x0	x0	0	x	0
bypass	1/0	1/0	1/0	1/0	0	0	1	x0	x0	0	x	x

Note – x : unaffected; x0 : unaffected and default state 0

- Normal Translation: The TLB receives a virtual address and a context identifier as input and produces a physical address and page attributes as output. This operation is contolled by tlb_cam_vld. This is a single cycle operation resulting in the generation of the physical address and individual way hit.
- Read Operation: The TLB simultaneously reads the CAM and RAM portion of the specified entry. This operation is controlled by tlb_rd_vld and tlb_rw_index_vld (=1 regular index; =0 replacement index). This a single cycle operation with the tag array accessed during the A-phase and the data array accessed during the B-phase.
- Write Operation: The TLB simultaneously writes the CAM and RAM portion of the specified entry, or the entry given by the replacement policy. This operation is controlled by tlb_wr_vld and tlb_rw_index_vld (=1 regular index; =0 replacement index). This a single cycle operation with the tag array accessed during the A-phase and the data array accessed during the B-phase.

- Bypass Operation: The TLB receives a virtual address as input and produces a physical address equal to the truncated virtual address page attributes as output. This operation is controlled by tlb_bypass.
- Demap Operation: The TLB receives a virtual address and a context identifier as input and sets the valid bit to zero for any entry matching the demap page or demap context or demap real or demap all criteria. This operation produces no output.

The purpose of a demap operation is to remove zero, one, or more entries in the TLB. Four types of demap operations are provided: context, real, page and all. The 3-bit partition identifier is compared in all demap operations. All these operations alongwith their implementation have been shown in the tag array datatpath organization.

- Demap Page (Type 0 demap operation) removes the TTE from the specified TLB matching the specified virtual page number, real bit, partition identifier register, and context register. Virtual page offset bits <15:13>, <21:13> and <27:13> for 64 kB, 4 MB, and 256 MB page TLB entries, respectively, are stored in the TLB, but are always set to zero and do not participate in the match for that entry. This is the same condition as for a translation match. This operation is controlled by tlb_demap.
- Demap Context (Type 1 demap operation) removes all TTEs from the specified TLB having the specified context, a real bit of 1'b0 and matching the partition identifer register. This operation is controlled by tlb_demap and tlb_demap_context.
- Demap All (Type 2 demap operation) removes all TTEs from the specified TLB matching the partition identifier register. This operation is controlled by tlb_demap and tlb_demap_all.
- Demap Real (Type 3 demap operation) removes all TTEs from the specified TLB matching the specified real bit and the partition identifier register. This operation is controlled by tlb_demap and tlb_demap_real.

Replacement Policy: OpenSPARC T2 uses a 1-bit LRU scheme. Each TLB entry has an associated valid and used bit. Used bits are set on each TLB translation and also on the initial TLB write of an entry. When setting the used bit for a translation or TLB write would result in all used bits being set, the used bits for all TLB entries are cleared instead. This is a 2-cycle operation.

3.4 Timing Diagrams

FIGURE 3-3 I/O Timing Diagrams (CAM

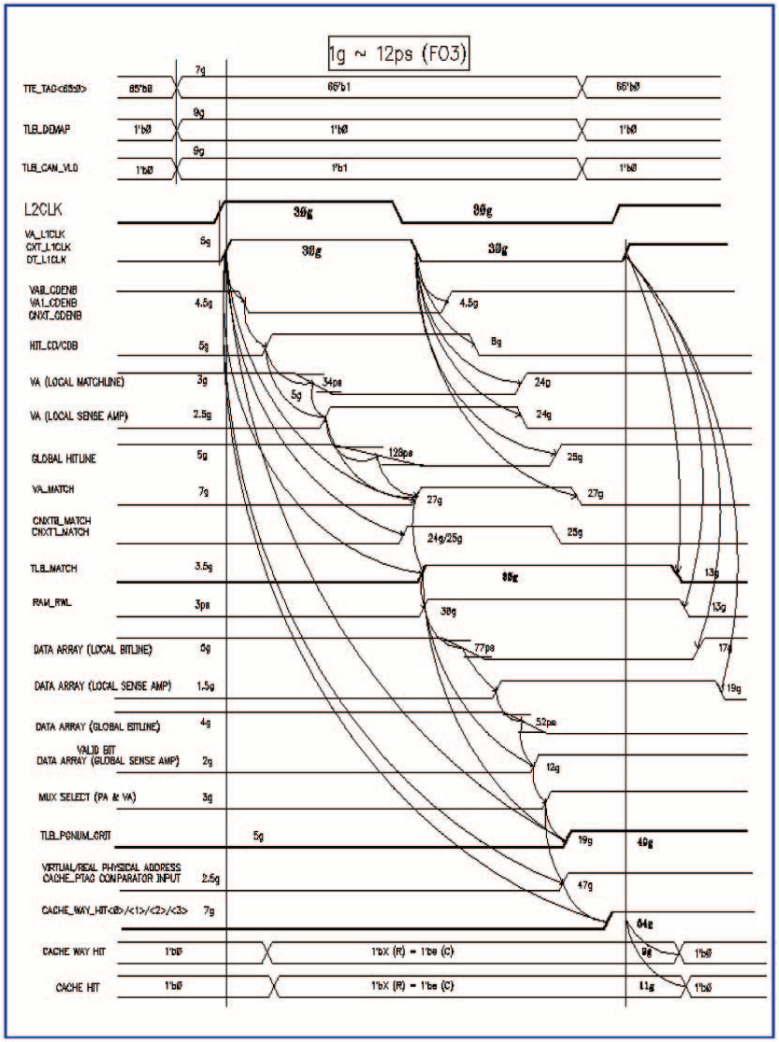


FIGURE 3-4 I/O Timing Diagrams (Demap)

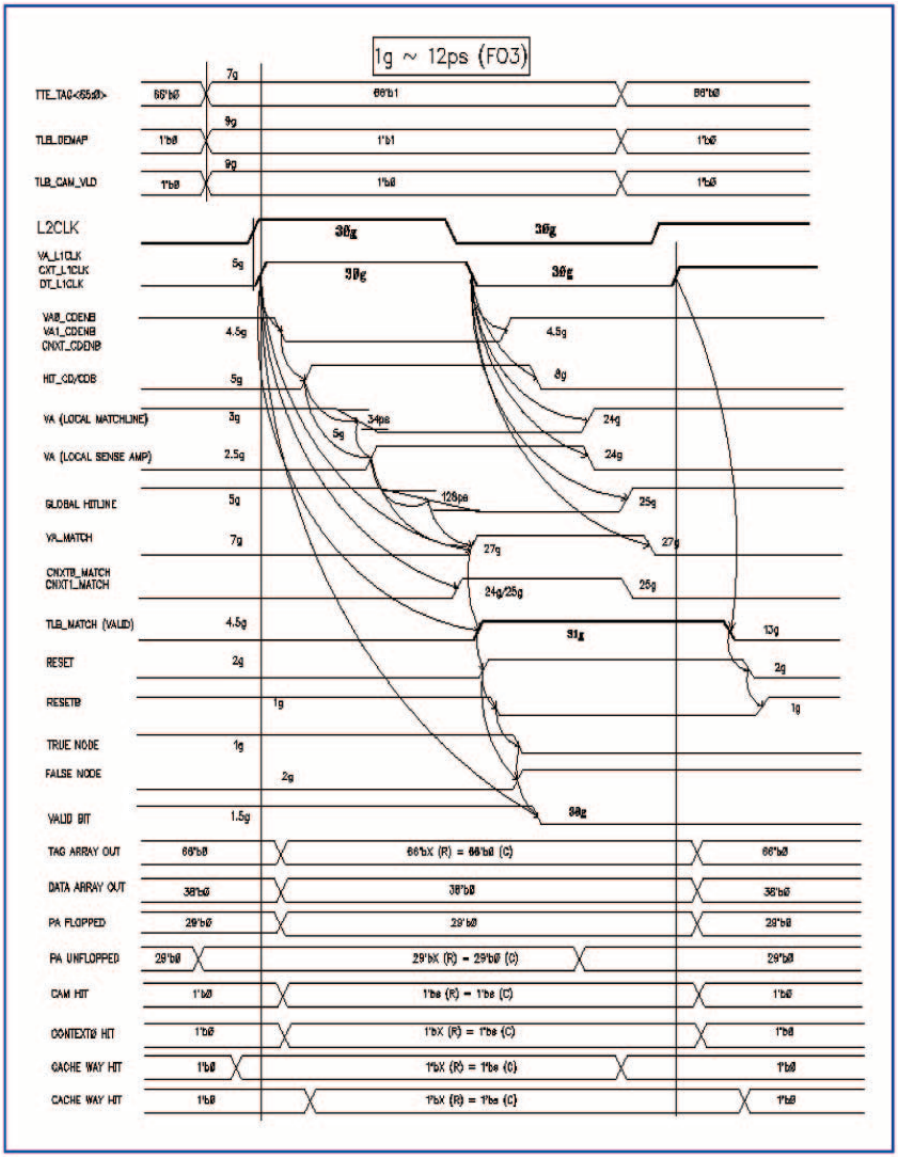


FIGURE 3-5 I/O Timing Diagrams (Replacement Index)

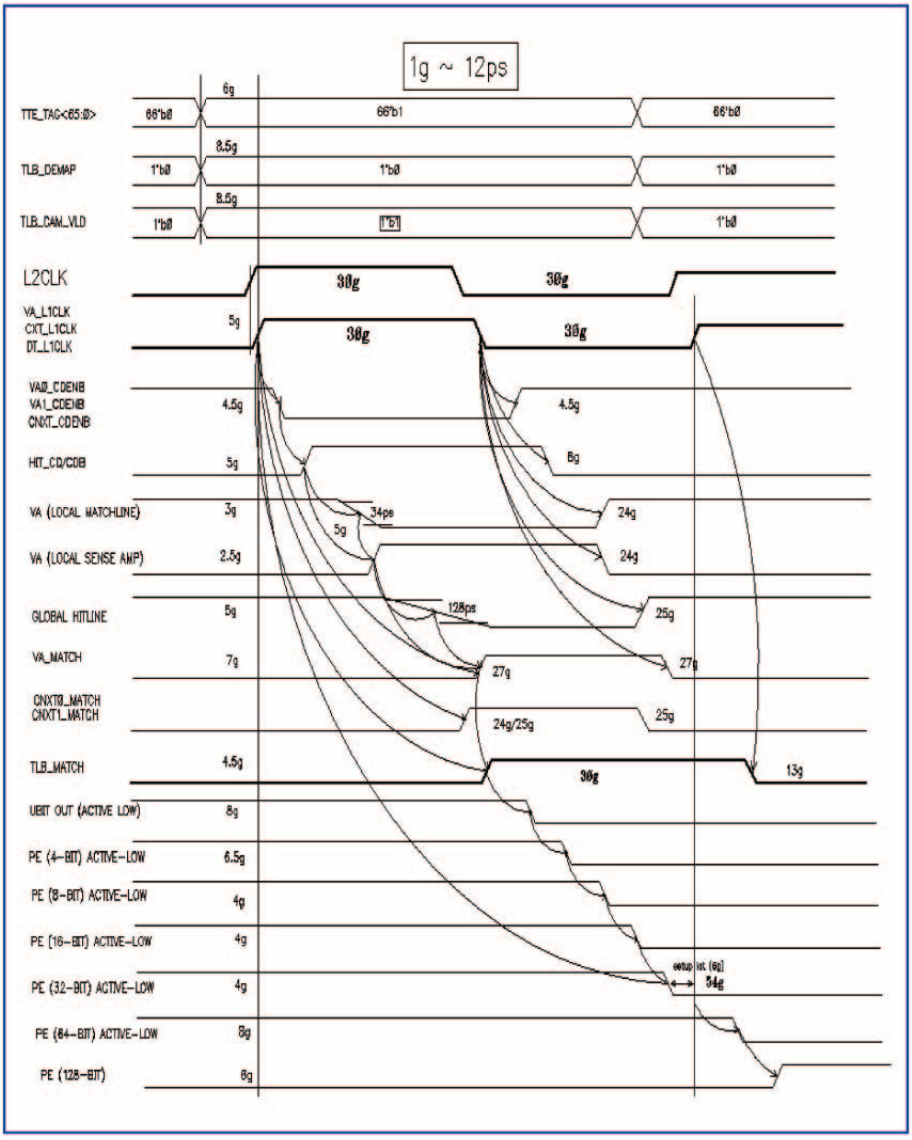


FIGURE 3-6 I/O Timing Diagrams (Mutlimatch/Context0 Hit/Cam Hit)

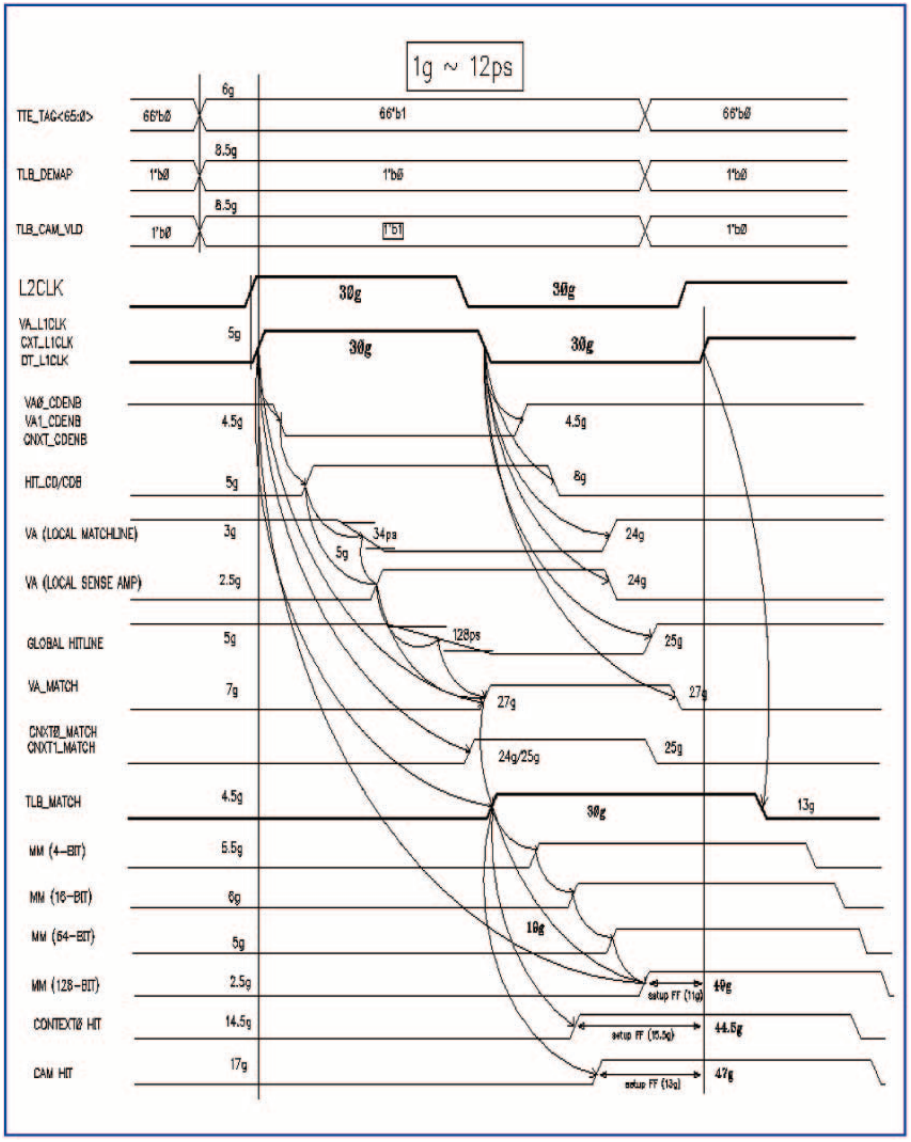
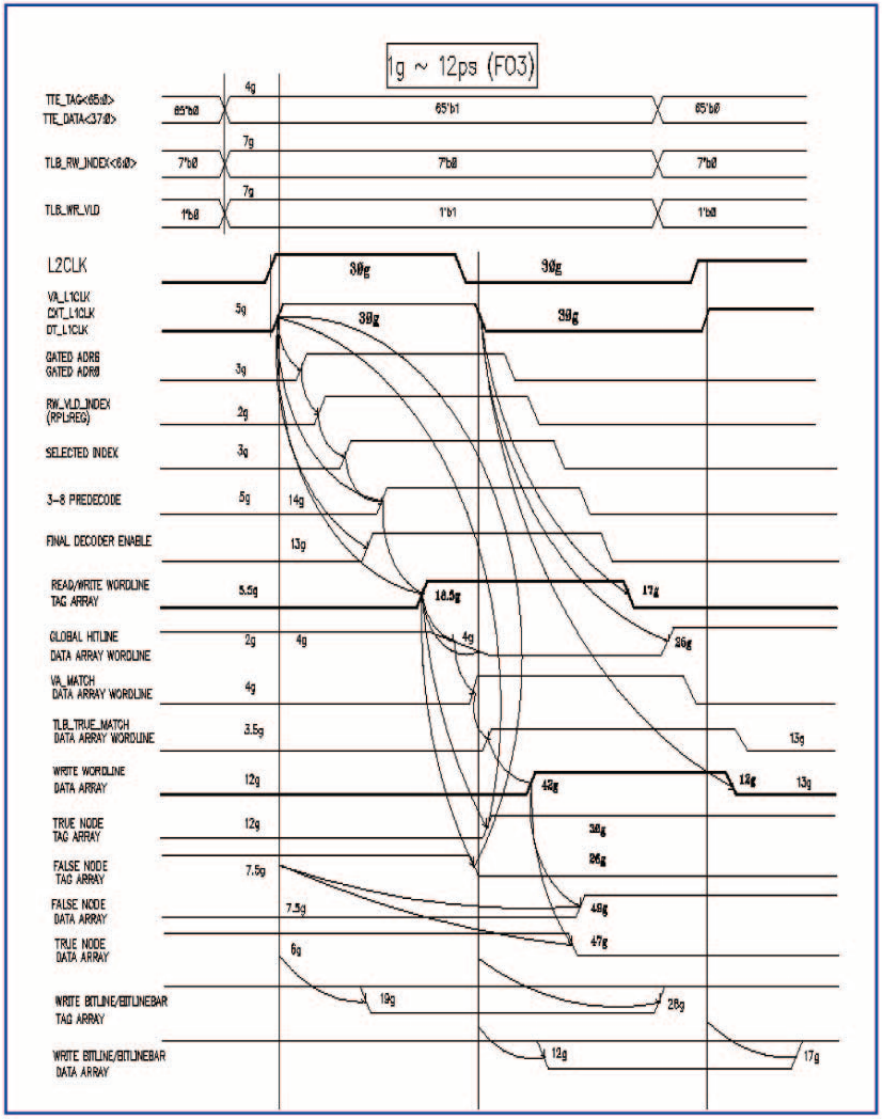


FIGURE 3-7 I/O Timing Diagrams (Write)


1g ~ 12ps (F03)
TTE_TAG<65:0>
TTE_DATA<37:0>
65'h0
65'b1
65'h0
4g
TLB_RW_INDEX<6:0>
7'h0
7'b0
7'h0
7g
TLB_WR_VLD
1'h0
1'b1
1'h0
7g
L2CLK
VA_L1CLK
CXT_L1CLK
DT_L1CLK
GATED ADR8
GATED ADR0
RW_VLD_INDEX
(RPLREG)
SELECTED INDEX
3-8 PREDECODE
FINAL DECODER ENABLE
READ/WRITE WORDLINE
TAG ARRAY
GLOBAL HITLINE
DATA ARRAY WORDLINE
VA_MATCH
DATA ARRAY WORDLINE
TLB_TRUE_MATCH
DATA ARRAY WORDLINE
WRITE WORDLINE
DATA ARRAY
TRUE NODE
TAG ARRAY
FALSE NODE
TAG ARRAY
FALSE NODE
DATA ARRAY
TRUE NODE
DATA ARRAY
WRITE BITLINE/BITLINEBAR
TAG ARRAY
WRITE BITLINE/BITLINEBAR
DATA ARRAY
30g
30g
30g
30g
5g
3g
2g
3g
5g
14g
13g
5.5g
18.5g
17g
2g
4g
4g
36g
4g
3.5g
13g
12g
42g
12g
13g
12g
30g
26g
7.5g
7.5g
48g
47g
6g
19g
28g
12g
17g

CHAPTER 4

Integer Register File (IRF)

This chapter describes the following topics:

4.1 Functional Description

The integer register file (IRF) is a 32-entry x 72-bit structure, replicated four times for each thread. It has three read ports and two write ports. All three reads occur to the same thread, while the writes may be to different threads.

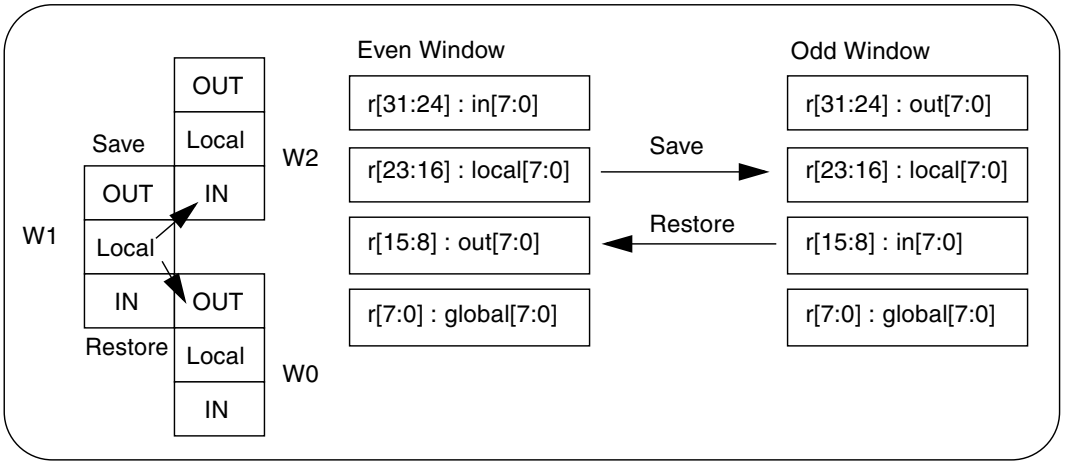
The first write port is for data from the normal integer pipeline or loads that do not return as long-latency operations. The second write port is for data from long-latency operations (load, division, multiplication). Because a load will always get access, no acknowledgement is required. However, both the multiplication (MUL) and division (DIV) require acknowledgements and will stall until their data has been written.

The 32 entries are split into 16 I/O registers – eight local registers and eight global registers. The register file supports eight windows per thread. Each local register is made up of eight basic registers, one per window. In addition, they contain one active register for each thread, which has the contents of the current window. Each I/O register has four basic registers, which will be shared between even and odd windows, and one active register for each thread.

A SWAP operation involves both SAVE and RESTORE operations pipelined internally and takes three cycles. A SWAP of locals, globals, evens, or odds can occur. However, only two SWAPs out of four sets of registers will occur due to power constraints.

A SAVE is done by transferring the contents of the active register to one of the basic registers attached to it. Later when a RESTORE is done, the contents of this basic register are transferred back to the active register. The “in” register in an odd window becomes the “out” register in an even window and vice versa. Input register addresses must be translated to real register addresses before they are sent to the register file as shown in FIGURE 4-1. In an even window, the input and real address are the same. In the odd window, address range [31:24] must be translated to [15:8], and the address range [15:8] will be translated to [31:24]. The global registers are the same across all windows. They are replicated to provide three sets of alternate global registers for use by privileged software only.

FIGURE 4-1 Register File Window Structure



4.1.1 General

Integer Register File functions include the following:

- 1.4Ghz read and write, based on simulation with spl_noise netlist.
- Three single ended read ports - Two dual ended write ports.
- Data width is 72 bits. Array uses 2 -> 1 column mux.
- 32 active registers for each type : even, local, odd, and global.
- Even, odd, and global have four shadow registers for each active register.

- Local has eight shadow registers for each active register.
- Write to active reg in 2nd phase. Read/Save/Restore to active reg in 1st phase.
- Read/Write is a single cycle operation. Save/Restore is a two cycle operation.
- Multiple Writes to same address (tid) disallowed via external logic
- Read/Write to same address produces "X" on output.
- Both Save and Restore to same thread in same cycle disallowed via external logic.
- Save/Restore allowed in all 4 reg types (even, odd, local, global) at same time.
- Flop bound inputs. No flops for Dout.
- All BIST external to array.
- Library flops with custom layout used on inputs.
- Custom non-scanable latches for swap wordline cntrl between 1st and 2nd cycle.

4.1.2 Read/Write Operations

- Data width is 72 bits. Array uses 2 -> 1 column mux.
- Write to active reg in 2nd phase. Read to active reg in 1st phase.
- Read/Write is a single cycle operation.
- Multiple Writes to same address (tid) disallowed via external logic
- Read/Write to same address produces "X" on output.

4.1.3 Save/Restore Operations

- 32 active registers for each type : even, local, odd, and global.
- Even, odd, and global have four shadow registers for each active register.
- Local has eight shadow registers for each active register.
- Save/Restore to active reg in 1st phase.
- Save/Restore is a two cycle operation.
- Both Save and Restore to same thread in same cycle disallowed via external logic.
- Save/Restore allowed in all 4 reg types (even, odd, local, global) at same time.

4.1.4 Illegal Operations

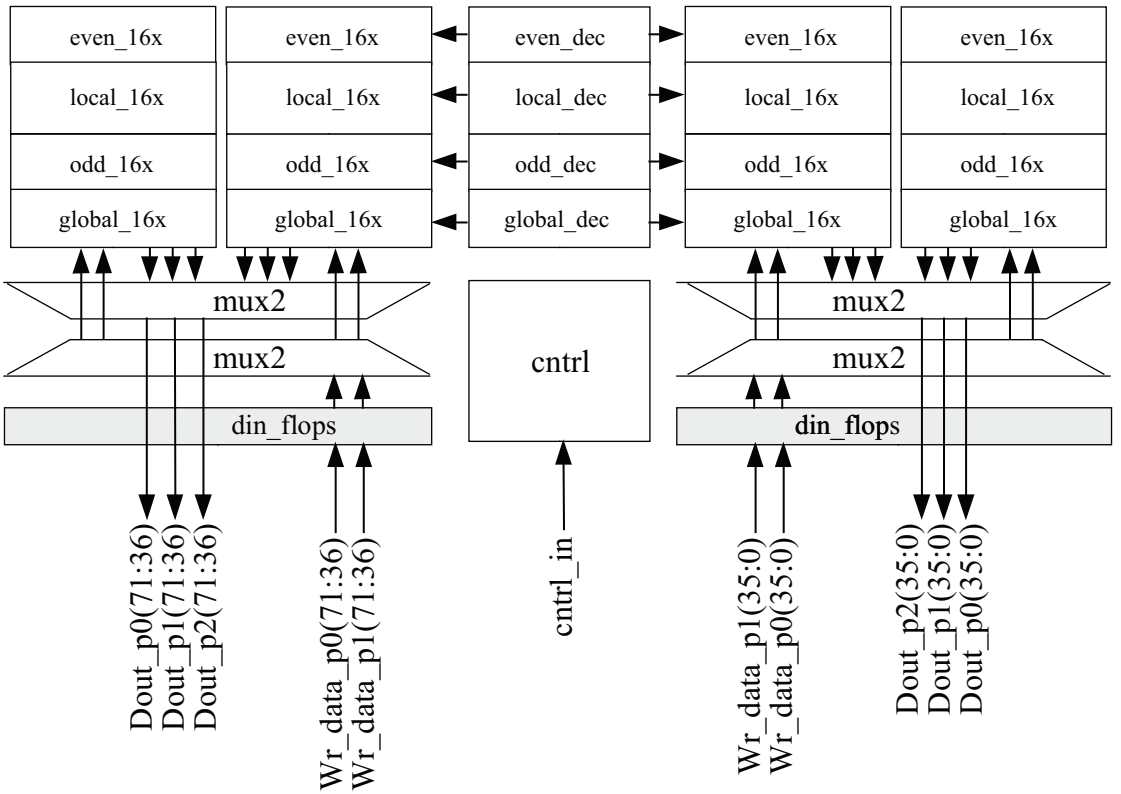
- Read and Write to the same TID and same address in the same cycle allowed. Write will complete. Read will produce an 'X' in RTL and possible high power state via metastability. (read collision)

- Both write ports can not write to the same TID in the same cycle. (write collision)
- Save and Restore to the same TID in the same cycle not allowed. (swap collision)
- No phases with back to back operations to the same TID. No swaps immediately preceeded or followed by a read or write to the same TID. (access overlap)

4.2 Block Diagram

The OpenSPARC T2 processor has eight SPARC® cores. IRF is part of each SPARC core.

FIGURE 4-2 IRF Block Diagram



4.3 I/O List

In the following IRF I/O list, all timing is with regard to rclk.

- Signal naming is source_destination_signalname.
  - *ifu* is instruction fetch unit.
  - *ecl* is exu control logic.
  - *byp* is bypass unit.
  - *rml* is register management.
  - *logic, clk, si, so, se* are global signals.
- -12 r means – 12 ps setup with regard to rclk clock.
- rclk(L2clk) – 12 f means -12 ps setup with regard to falling clock.
- rclk(L2clk) – Output delay is with regard to falling rclk with 4x o/p load.

TABLE 4-1 IRF Functional and DFT I/O Signal List

Name	Width	I/O	Flopped	Source/ Dest	Function
dout_p[2/1/0]	72	O			Data to be read on port[2/1/0]
rd_addr_p[2/1/0]	5	I	Y		Determines which entry is read on port[2/1/0]. Bits(4:1) are used to index into one of sixteen wordlines. Decodes of 0-3 access global registers, 4-7 odd registers, 8-11 local registers, and 12-15 even registers. Bit(0)is used for mux2 selection of 72 out of 144 columns.
rd_en_p[2/1/0]	1	I	Y		Enables a specific port for reading.
rd_tid	2	I	Y		Determines which thread is made available to the read port. All ports see the same tid selection for reads
wr_data_p[1/0]	72	I	Y		Data to be written on port[1/0]. All BIST muxing external.
wr_addr_p[1/0]	5	I	Y		Determines which entry is written on port[1/0]. Bits(4:1) are used in conjunction with the wr_tid_p[1/0] bits for that specific port to index into one of sixty-four wordlines. Bit(0) is used for mux2 selection of 72 out of 144 columns.
wr_en_p[1/0]	1	I	Y		Enables a specific port for writing.

TABLE 4-1 IRF Functional and DFT I/O Signal List *(Continued)*

wr_tid_p[1/0]	2	I	Y		Determines which thread is made available to the write port. Unlike reads separate write ports can access different threads. Indeed the two write ports must have a different tid.
Save_(even)_addr (odd/global)	2	I	Y		Determines which shadow register (one of four) receives the active register contents for the (even/odd/global) bits. Only a single thread is saved in any cycle. All active even registers for a given tid are saved at the same time. Saves are a two cycle operation. The save does not actually start until one cycle after the address is flopped.
save_local_addr	3	I	Y		Determines which shadow register (one of eight) receives the active register contents for the local bits (See save_even_addr for more detail).
save_(even)_en (local/odd/global)	1	I	Y		Enables saves for (even/local/odd/global) registers. All enables can be asserted in same cycle.
save_tid	2	I	Y		Determines which thread will be saved for even/odd/local registers.
save_global_tid	2	I	Y		Determines which thread will be saved for global registers.
restore_(even)_addr (odd/global)	2	I	Y		Determines which shadow register (one of four) updates the active register contents for the (even/odd/global) bits. Only a single thread is restored in any cycle. All active even registers for a given tid are restored at the same time. Restores are a two cycle operation. The restore does not actually start until one cycle after the address is flopped.
restore_local_addr	3	I	Y		Determines which shadow register (one of eight) updates the active register contents for the local bits (See restore_even_addr for more detail).
restore_(even)_en (local/odd/global)	1	I	Y		Enables restores for (even/local/odd/global) registers. All enables can be asserted in same cycle.
restore_tid	2	I	Y		Determines which thread will be restored for even/odd/local registers.
restore_global_tid	2	I	Y		Determines which thread will be restored for global registers.
l2clk	1	I	N	IO clk	Core clock
tcu_aclk	1	I	N	tcu	Scan In clock
tcu_bclk	1	I	N	tcu	Scan Out clock
tcu_se_scancollar_in	1	I	N	tcu	Scan enable for input flops. Controls shutting off input masters during macrotest.

TABLE 4-1 IRF Functional and DFT I/O Signal List *(Continued)*

tcu_se	1	I	N	tcu	Scan enable for array with exception of input flops. This is the same as the scan enable to RTL flops. This allows array to run for both macrotest and LBIST but will shut off the L1clk during scanning.
clk_en	1	I	N		pce to clock header used for local clock control to shut off clock for functional or power saving purposes
tcu_pce_ov	1	I	N	tcu	pce_ov to clock header used in override for scan of clk_en (pce) having turned clock off.
tcu_clk_stop	1	I	N	tcu	stop to clock header. This is an at speed signal
tcu_array_wr_inhibit	1	I	N	tcu	Prevents array from writing, reading, or swaping. Effectively "anded" with wr_en, rd_en, save_en, and rsto_en. Turns off wordline and holds precharge for swaps. Turns off wordline, column selects, and holds precharge for writes. Turns off wordline and holds precharge (not column select) for reads.
scan_in	1	I	N		Block scan input. Latched on falling tcu_aclk edge.
scan_out	1	I	N		Block scan output. Launched on falling tcu_bclk edge.

4.4 Functional Operation

TABLE 4-2 IRF Modes of Operation

Enable Values				Memory Operation	Output Results (dout_p[2/1/0])
read	write	save	restore		
L	L	L	L	No Operation	L
H	L	L	L	Read Cycle	Results of Read
L	H	L	L	Write Cycle	L
H	H	L	L	Read/Write Cycle with DOUT = X for the same address	Results of Read
L	L	H	L	Save Cycle	L
H	L	H	L	Illegal	
L	H	H	L	Illegal	
H	H	H	L	Illegal	
L	L	L	H	Restore Cycle	L
H	L	L	H	Illegal	
L	H	L	H	Illegal	
H	H	L	H	Illegal	
L	L	H	H	Save/Restore Cycle with different TID	L
H	L	H	H	Illegal	
L	H	H	H	Illegal	
H	H	H	H	Illegal	

4.5 Timing Diagram

- A SWAP instruction will send the current window pointer (CWP) and decoded SWAP (partial or full) or alternate_global control signals to the register file in E stage.

- The cancellation of the transition must be sent to the register file before the end of the W stage for the transition in the first (rising) phase of W2 stage and the switched window is available for read/write in the second (falling) phase.
- The predecoded READ (WRITE) addresses and the associated thread are sent to the register file in the S (M) stage.
- Write enable will arrive late and set up to the falling edge of the W stage and have the (WRITE) operations done in the second phase of the (W) stage. Read occurs in the second phase of D stage.

FIGURE 4-3 IRF Read Timing Diagram

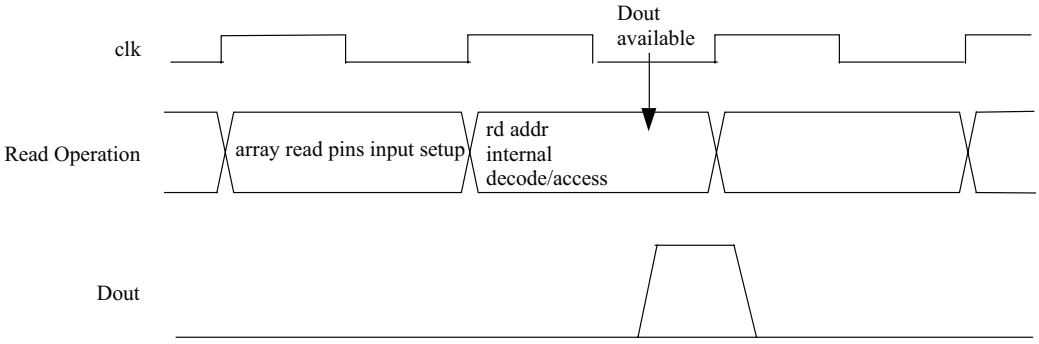
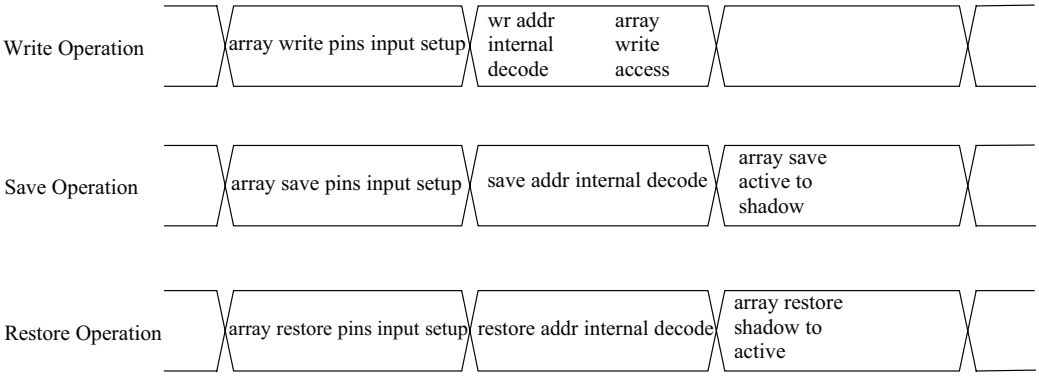


FIGURE 4-4 IRF Write Timing Diagram



CHAPTER 5

Memory Management Unit CAM

This chapter describes the following topics:

5.1 Functional Description

The mmu_cm_64x34s_cust is a single read port, single write port cam. Read, and match operations occur in the first phase of the clk. Write operations occur in the second phase of the clk. The cam produces a set of matches (hits). This CAM operates at 375MHz.

The Tag Buffer (DMU_MMU_TB) CAM array is the underlying custom array of two CAM structures in the IOMMU block, residing in DMU, the PCI-Express Data Management Unit. The 2 instances are:

1. Virtual Tag Buffer: a 32-bit wide, 64-entry array that stores the virtual tags of the 64 TTE entries
2. Physical Tag Buffer: a 33-bit wide, 64-entry array that stores the physical tags of the TTE entries.

Organization:

- OpenSPARC 2 Tag Buffer (TB) CAM array is structured as a synchronous 33-bit wide, 64-entry array.

This CAM supports the following 3 functions:

1. Lookup Reference Address: key[32:0] and lkup_en are driven from control logic to the CAM. They are registered at the input of the CAM. If the hld signal is asserted, then the flops of the key input holds its previous value. The registered key is compared to stored data in each of the 64 entries, and generates a decoded 64-bit vector, hit[63:0]. The hit output is unregistered.

In normal operation, the key should match at most 1 entry in the CAM. If due to S/W errors, a CAM operation results in multiple hits, the hit[63:0] output from the CAM will have multiple 1's accordingly. Tentatively, it does NOT need to generate a separate multi-hit flag signal. If needed, the priority encoder outside the CAM will generate it.

2. Write access: Write to a entry specified by wr_addr and wr_en. The write address and write enable are registered at the input of the array.
3. Read access: Read from a entry specified by rd_addr and rd_en. The read address and read enable are registered at the input of the array, and the data output is unregistered.

- Number of entries.

  64

- Number of bits per entry.

  33

- Column folding (read or write column muxing).

  None

- How many read/write/cam ports.

  One read, one write, and one cam

- Operations which are allowed or not allowed in same cycle.

  All operations allowed

- Operations which are allowed or not allowed to occur in consecutive cycles.

  All operations allowed

TABLE 5-1 Operations

Operation	cycles	phase	bits
Read	1	A	32:0
Write	1	B	32:0
Match (cam)	1	A	32:0

5.2 Block Diagram

FIGURE 5-1 MMU Block Diagram

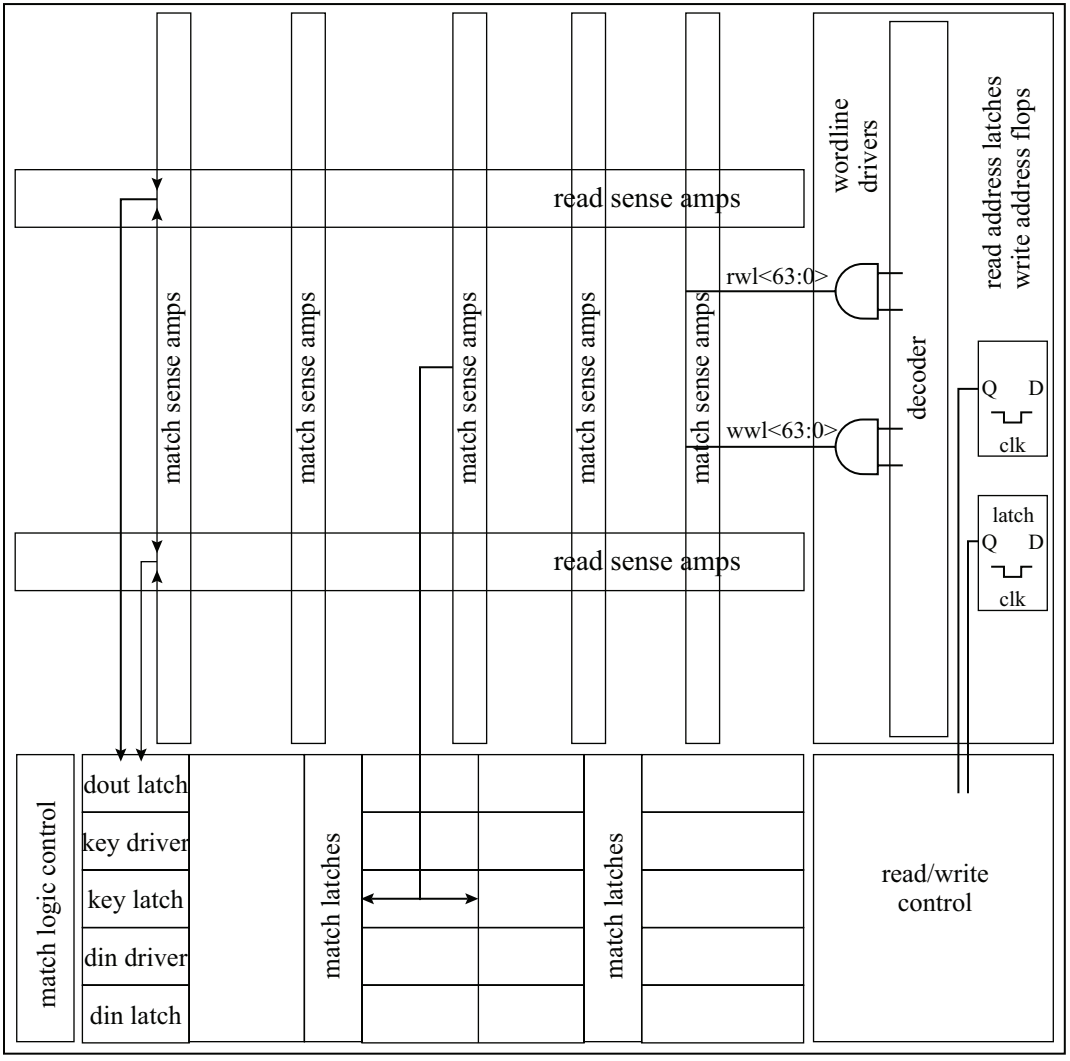
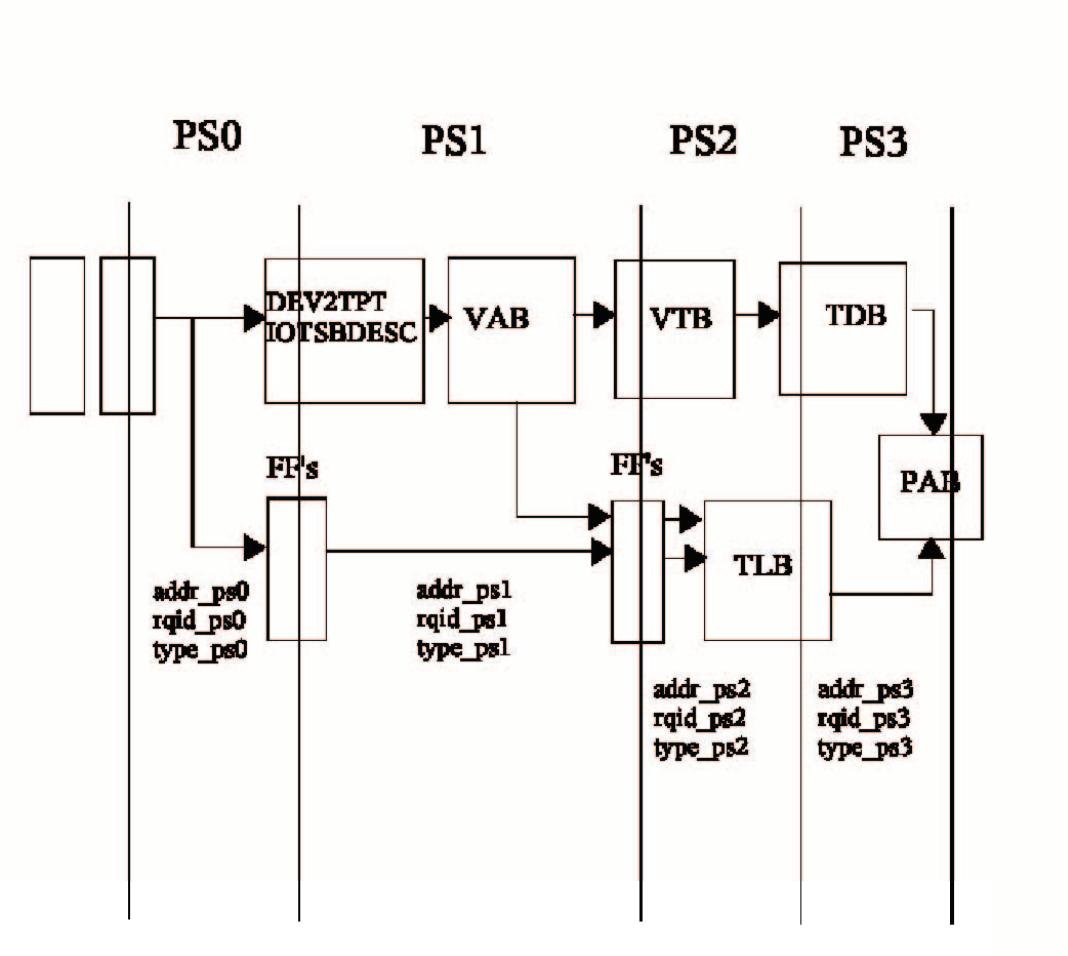


FIGURE 5-2 Pipeline Diagram



5.3 I/O List

TABLE 5-2 I/O List

Signal Name	I/O	Flopped	Source/Dest.	Description
rd_addr [5:0]	In	Yes	dmu	Read Address
wr_addr [5:0]	In	Yes	dmu	Write Address
rd_en	In	Yes	dmu	Read Enable
wr_en	In	Yes	dmu	Write Enable
din [32:0]	In	Yes	dmu	Write Data
key [32:0]	In	Yes	dmu	cam data for matching
lkup_en	In	Yes	dmu	cam operation enable
hld	In	Yes	dmu	virtual tag buffer: if hld=1 i/p flops retain previous key value. Previous hit values are held on o/p physical tag buffer: hld=0
Clock and Test Related I/Os				
clk	In	No	grid	main clock
hit [63:0]	Out	No	dmu	cam result for key [32:0]
dout [32:0]	Out	No	dmu	Read Output Data
pce	In	No	tied high	clock enable
tcu_array_wr_inhibit	In	No	tcu	inhibits operations in the array
tcu_se_scancollar_in	In	No	tcu	controls input flops
tcu_pce_ov	In	No	tcu	overrides the clock enable
tcu_adk	In	No	tcu	scan latch clock
tcu_bdk	In	No	tcu	scan latch clock
tcu_clk_stop	In	No	tcu	clock header control
tcu_array_bypass	In	No	tcu	bypass mode control
scan_out	In	No	dmu	output of scan chain
scan_in	In	Yes	dmu	input of scan chain
tcu_scan_en	In	No	tcu	control internal strobes

5.4 Functional Operation

TABLE 5-3 Modes of Operation

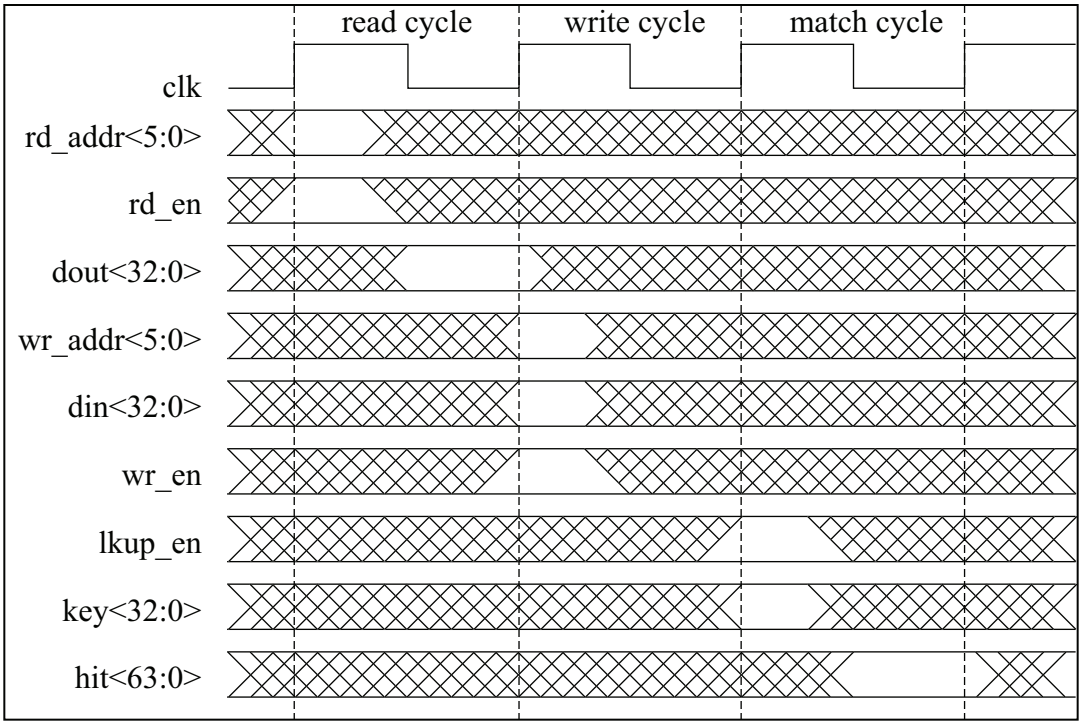
Enable values			Operation	Results	
read_en	write_en	Lookup_en	loperation	dout	match
L	L	L	no operation	previous data	64'b0
L	L	H	look up	previous data	match results
L	H	L	write	previous data	64'b0
L	H	H	lookup(A) then write (B)	previous data	match results on old data
H	L	L	read	read data	64'b0
H	L	H	read (A) and lookup (A)	read data	match results
H	H	L	read (A) then write (B)	read data	64'b0
H	H	H	read (A) and lookup (A) then write (B)	read data	match results on old data

TABLE 5-4 Functional Operational Table

Enables				Comments
read_en	write_en	Lookup_en	hld	
X	0	1	0	key[32:0] is compared to all entries in the array, and the hit[63:0] vector is valid during the next cycle
X	0	X	1	Input flops retain the previous key value
X	1	1	0	A write update and a CAM operation occur during the same cycle. The CAM results are based on data in the array before the write
X	1	0	X	Data written to entry specified by wr_addr at rising edge of the clk
1	X	1	X	Data read from entry specified by rd_addr

5.5 Timing Diagrams

FIGURE 5-3 I/O Timing Diagram



CHAPTER 6

Floating-point Register File (FRF)

This chapter describes the following topics:

6.1 Functional Description

256 entry 64bits floating point register file(FRF) 2R/2W ports, supports eight-way multithreading (eight threads) by dedicating 32 entries for each registe file also includes 14 bits of ECC for a total of 78 bits per entry. Correctable ECC errors (CEs), and uncorrectable ECC errors (UEs) result in a trap if the corresponding enables are set. CEs are never corrected by hardware, but may be corrected by software following a trap.

One FRF write port (W2) is dedicated to floating-point loads and FPD (divide and square root) floating-point results. FPD results always have highest priority for W2 and are not required to arbitrate. The other FRF write port (W1) is dedicated to FPX (floating point execution pipeline) and FGX (graphics execution pipeline) results. Arbitration is not necessary for the FPX/FGX write port because of single instruction issue and fixed execution latency constraints. FPX, FGX, and FPD pipelines never stall.

Independent upper/lower half-word (32 bit) write enables are provided for each write port.

Can only read from one thread in a given cycle. Can write to multiple threads in a given cycle.

An attempt to read and write the same half word concurrently will result in a successful write, but read enable is turned off (Note, in this case, FRF read results are bypassed externally to FRF).

Floating-point store instructions share an FRF read port with the execution pipelines.

ASI read access is provided for all FRF ECC check bits stored in the FRF. Write access via ASI is not provided for the FRF ECC check bits stored in the FRF. Write access is provided only via the 7-bit ECC mask used for error injection. ECC check bits are always read via ASI using the FRF rs1 port.

This CAM supports the following 3 functions:

1. Lookup Reference Address: key[32:0] and lkup_en are driven from control logic to the CAM. They are registered at the input of the CAM. If the hld signal is asserted, then the flops of the key input holds its previous value. The registered key is compared to stored data in each of the 64 entries, and generates a decoded 64-bit vector, hit[63:0]. The hit output is unregistered.

   In normal operation, the key should match at most 1 entry in the CAM. If due to S/W errors, a CAM operation results in multiple hits, the hit[63:0] output from the CAM will have multiple 1's accordingly. Tentatively, it does NOT need to generate a separate multi-hit flag signal. If needed, the priority encoder outside the CAM will generate it.

2. Write access: Write to a entry specified by wr_addr and wr_en. The write address and write enable are registered at the input of the array.

3. Read access: Read from a entry specified by rd_addr and rd_en. The read address and read enable are registered at the input of the array, and the data output is unregistered.

- Number of entries.

64

- Number of bits per entry.

33

- Column folding (read or write column muxing).

None

- How many read/write/cam ports.

One read, one write, and one cam

- Operations which are allowed or not allowed in same cycle.

All operations allowed

- Operations which are allowed or not allowed to occur in consecutive cycles.

All operations allowed

TABLE 6-1 Operations

Operation	cycles	phase	bits
Read	1	A	32:0
Write	1	B	32:0
Match (cam)	1	A	32:0

6.2 Block Diagrams

FIGURE 6-1 FRF Block Diagram

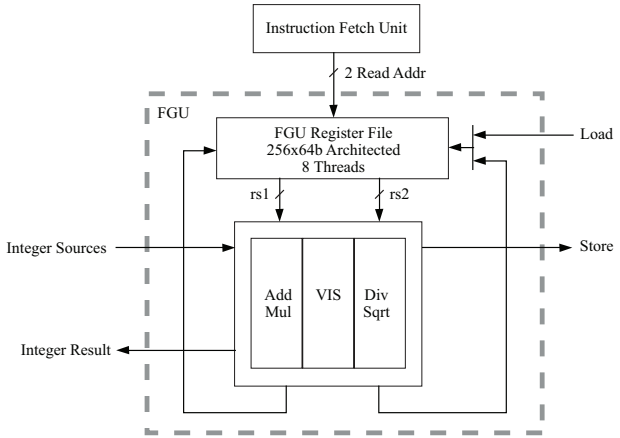
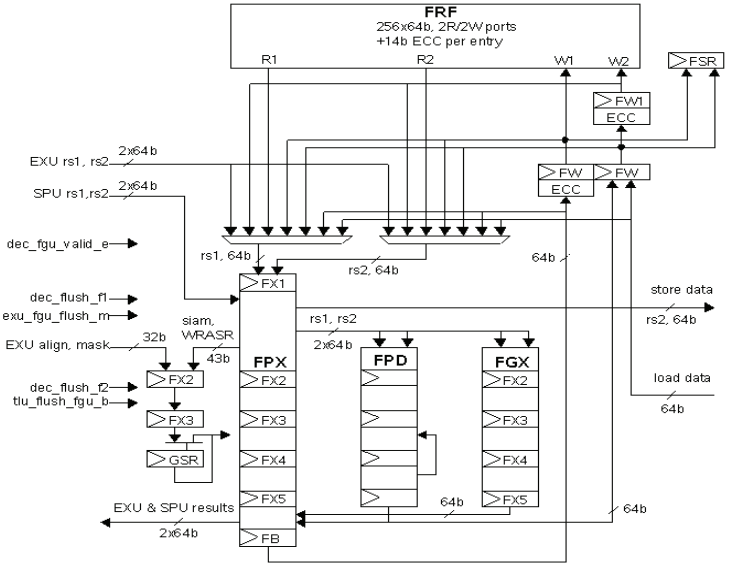


FIGURE 6-2 Pipeline Diagram



6.3 I/O List

TABLE 6-2 I/O List

Signal Name	Width	I/O	Flopped	Source/Dest.	Description
frf_r2_data	[63:0]	In	Yes	dmu	Output data
frf_r1_ecc	[13:0]	In	Yes	dmu	Output data
frf_r2_ecc	[13:0]	In	Yes	dmu	Output data
frf_r1_data	[63:0]	In	Yes	dmu	Output data
r1_addr	[4:0]	In	Yes	dmu	Read port one address
r2_addr	[4:0]	In	Yes	dmu	Read port two address
w1_tid	[2:0]	In	Yes	dmu	Write port one address
w2_tid	[2:0]	In	Yes	dmu	Write port two address
r_tid	[2:0]	In	No	grid	Read tid shared
w1_valid	[1:0]	Out	No	dmu	Write port one enable
w2_valid	[1:0]	Out	No	dmu	Write port two enable
main_clken		In	No	tied high	Power save clk_en
scan_out		In	No	Global	Scan output
scan_in		In	No	Global	Scan input
tcu_pce_ov		In	No	Global	Test control
Pce		In	llll	Global	Test control
tcu_aclk		In		Global	Scan clock
tcu_bclk		In		Global	Scan clock
tcu_scan_en				Global	Core scan en
tcu_se_scancollar_in		In	No	Global	Test control
w1_data	[63:0]	In	No	Fad	Write data input
w2_data	[63:0]	In	No	Fec	Write data input port two
w1_ecc	[13:0]	In	No	Fad	Write ecc bits port one
w2_ecc	[13:0]	In	No	Fec	Write ecc bits port two
r1_valid		In	Yes	Fac	Port one read_en
r2_valid		In	No	Dec	Port two read enable

6.4 Functional Operation

TABLE 6-3 Modes of Operation

Enable values			Operation	Results	
read_en	**write_en**	**Lookup_en**	**loperation**	**dout**	**match**
L	L	L	no operation	previous data	64'b0
L	L	H	look up	previous data	match results
L	H	L	write	previous data	64'b0
L	H	H	lookup(A) then write (B)	previous data	match results on old data
H	L	L	read	read data	64'b0
H	L	H	read (A) and lookup (A)	read data	match results
H	H	L	read (A) then write (B)	read data	64'b0
H	H	H	read (A) and lookup (A) then write (B)	read data	match results on old data

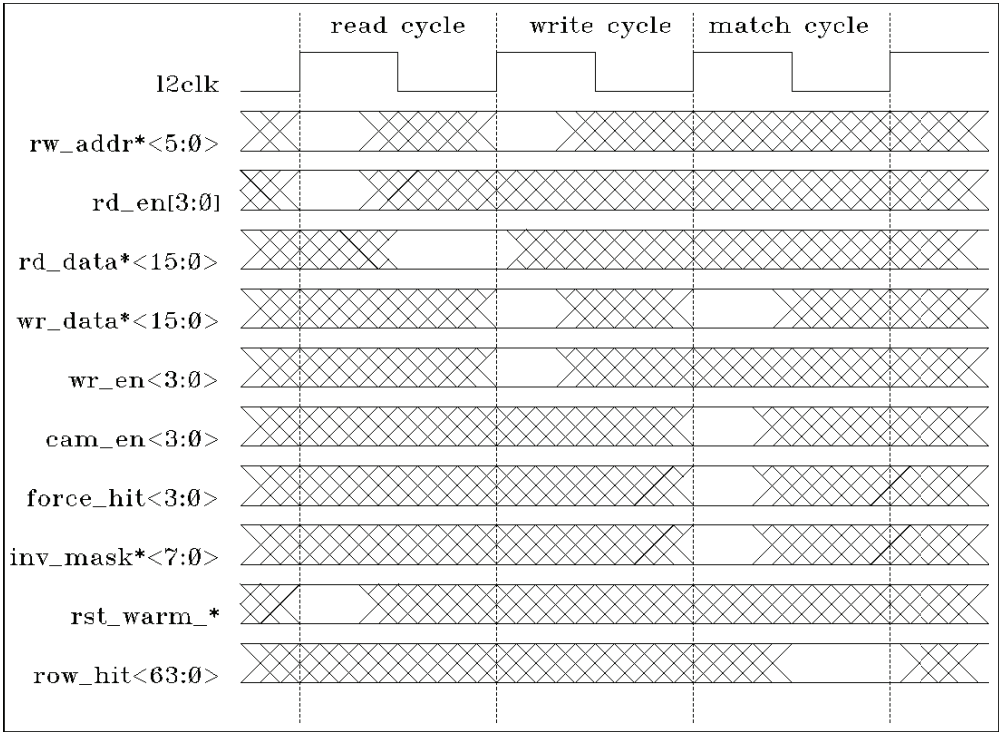
TABLE 6-4 Functional Operational Table

Enables				Comments
read_en	**write_en**	**Lookup_en**	**hld**	
X	0	1	0	**key[32:0] is compared to all entries in the array, and the hit[63:0] vector is valid during the next cycle**
X	0	X	1	Input flops retain the previous key value
X	1	1	0	A write update and a CAM operation occur during the same cycle. The CAM results are based on data in the array before the write
X	1	0	X	Data written to entry specified by wr_addr at rising edge of the clk
1	X	1	X	Data read from entry specified by rd_addr

0in monitor in RTL ; if read address = write address in the same cycle output will be

6.5 Timing Diagram

FIGURE 6-3 I/O Timing Diagram



CHAPTER 7

Instruction Cache Data Array (ICD)

This chapter describes the following topics:

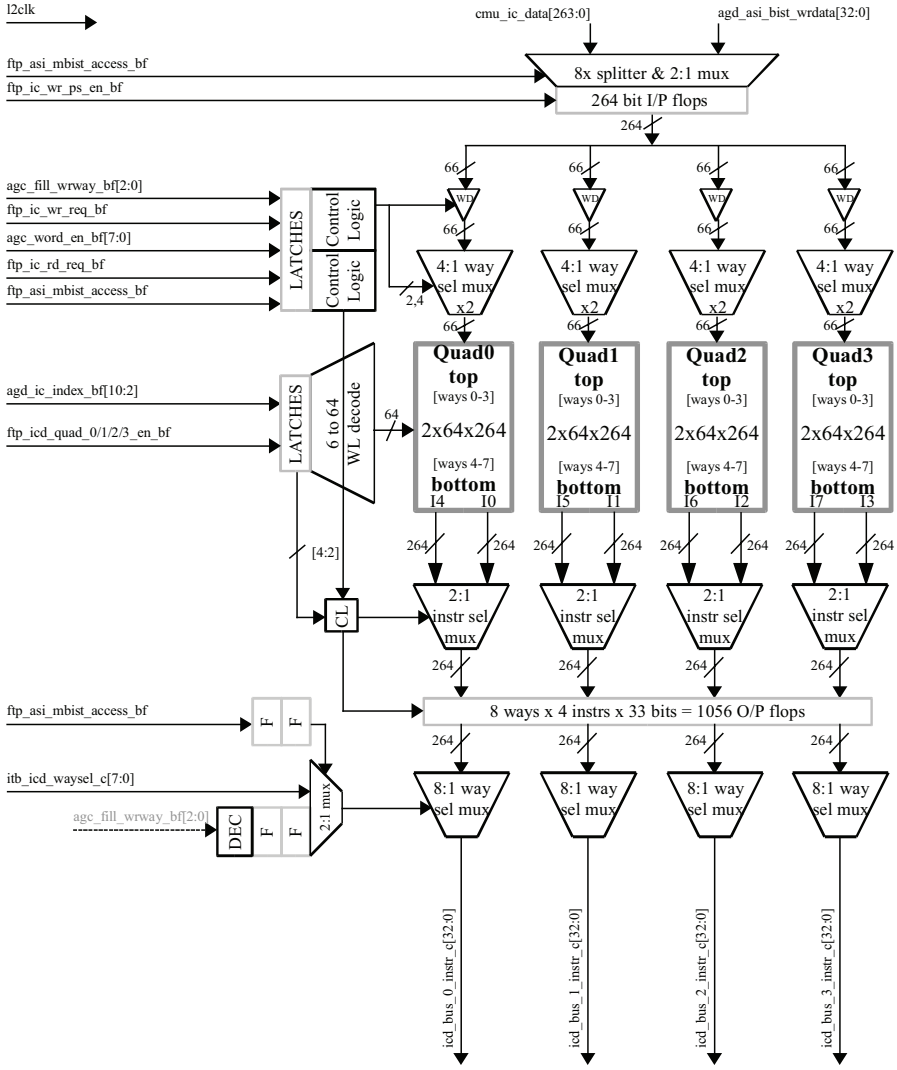
7.1 Functional Description

- OpenSPARC T2 uses 16.5KB (16KBytes data and 0.5KBytes parity), 32B line size, 8-way set associative instruction cache.
- Each way contains 64 entries of 264 bits (256 instruction bits and 8 parity bits).
- Each line contains 8 words of instruction data and one parity bit per instruction.
- Parity protection is implemented as one parity bit per word.
- During normal fetches, all ways in the ICD are accessed in parallel during the F cycle. In this cycle, four instructions are read from each way based on the index bits [4:2] according to TABLE 7-1 and are then flopped.
- In the C cycle, the instructions from all 8 ways are muxed using signals itb_icd_waysel_c [7:0]. Four instructions from the selected way are output.
- Write data bus provides 264 bits for a single cycle write into 1 of 8 selected ways.
- Read operation has a double cycle latency and single cycle throughput.
- Back-to back read or write is possible.
- Read and Write operation cannot occur in the same cycle.
- The ICD can also be accessed for ASI/MBIST reads and writes through normal read/write ports.

- The read data is output the same way for ASI/MBIST and functional modes. The ftp_asi_mbist_access_bf signal chooses which mode to operate in.
- During ASI write, only 1 of 8 instruction words are written into the selected way. Eight bit word write enable selects which word is written in selected way. During MBIST write, all 8 instruction words are written into the selected way. Eight bit word write enable is all-hot. Instruction data input bus is 264 bits wide and ASI/MBIST write data is 33 bits wide. Each 33 bit portion of instruction data is muxed with the 33 bits of ASI/MBIST write data.
- The outputs of the read way select muxes in the ICD are sent to the Datapath Block, which physically resides on top of the ICD. These outputs are also sent to the Instruction Buffers, which too reside on top of the ICD.
- All inputs are latched except for itb_icd_waysel_c [7:0].

7.2 Block Diagram

FIGURE 7-1 ICD Block Diagram



7.3 I/O List

The following table describes the ICD I/O signals.

TABLE 7-1 I/O List

Signal name	I/O	Flopped	Source/Destination	Description
Functional Signals				
agd_asi_bist_wrdata[32:0]	IN	YES	address gen. (agd_dp)	ASI/BIST write data
agd_ic_index_bf[10:2]	IN	YES	agd_dp	Read/Write index
agc_fill_wrway_bf[2:0]	IN	YES	add. gen. ctrl (agc_ctl)	Write way select
agc_word_en_bf[7:0]	IN	YES	agc_ctl	Write word enable for ASI
ftp_asi_mbist_access_bf	IN	YES	fetch pick unit (ftp_ctl)	ASI access request
ftp_ic_rd_req_bf	IN	YES	ftp_ctl	Read request signal
ftp_ic_wr_req_bf	IN	YES	ftp_ctl	Write request signal
ftp_icd_quad_0_en_bf	IN	YES	ftp_ctl	Quad enable signal for quad 0
ftp_icd_quad_1_en_bf	IN	YES	ftp_ctlWay Select	Quad enable signal for quad 1
ftp_icd_quad_2_en_bf	IN	YES	ftp_ctl	Quad enable signal for quad 2
ftp_icd_quad_3_en_bf	IN	YES	ftp_ctl	Quad enable signal for quad 3
ftp_ic_wr_ps_en_bf	IN	NO	ftp_ctl	Power save control signal for din flops
itb_icd_waysel_c[7:0]	IN	NO	instr_tlb	way select for instruction fetch
cmu_ic_data[263:0]	IN	YES	Load/Store unit (lsi_dp)	data for instruction fill
icd_bus_0_instr_c[32:0]	OUT	YES	bypass data (byp_dp)	Data out Bus 0
icd_bus_1_instr_c[32:0]	OUT	YES	byp_dp	Data out Bus 1
icd_bus_2_instr_c[32:0]	OUT	YES	byp_dp	Data out Bus 2
icd_bus_3_instr_c[32:0]	OUT	YES	byp_dp	Data out Bus 3
Redundancy/Repair Signals				
red_arst	IN	YES	sram header	fuse repair register reset
red_wen	IN	YES	sram header	fuse repair register write enable

TABLE 7-1 I/O List *(Continued)*

Signal name	I/O	Flopped	Source/Destination	Description
rid_in	IN	YES	sram header	fuse repair register address
red_en_in	IN	YES	sram header	fuse repair enables
red_d_in	IN	YES	sram header	fuse repair values
red_en_out	OUT	YES	sram header	fuse repair enables from repair registers
red_d_out	OUT	YES	sram header	fuse repair values from repair registers
Clock and Scan Signals				
l2clk	IN	NO	---	global clock
tcu_pce_ov	IN	NO	TCU	scan network
tcu_array_wr_inhibit	IN	NO	TCU	memory write protection during scan
tcu_se_scancollar_in	IN	NO	TCU	scan enable for input flops
tcu_se_scancollar_out	IN	NO	TCU	scan enable for output flops
tcu_scan_en	IN	NO	TCU	scan enable for all other flops
tcu_aclk	IN	NO	TCU siclk	scan clock for scan_in
tcu_bclk	IN	NO	TCU soclk	scan clock for scan_out
scan_in	IN	NO	scan chain	scan input
scan_out	OUT	NO	scan chain	scan output
Signal name	I/O	Flopped	Source/Destination	Description

7.4 Functional Table

TABLE 7-2 ICD Array Modes of Operation

Inputs			Mode Power Savings	Output at DOUT flops	Cache Operation	Supplementary Comments
rd_req	wr_req	asi_mbist_access				
0	0	Don't Care	No	Previous output	NOP	WL fires, dout
			Yes		NOP	WL, din, dout flops disabled
				Previous output		
0	1	0	No	Previous output	Write	Dout flops
			Yes		Write functional data	Dout flops disabled
				Previous output		
0	1	1	No	Previous output	Write ASI/MBIST	Dout flops disabled
			Yes			
				Previous output	Write ASI/MBIST data	Dout flops disabled
1	0	0	No	Valid	Read data	Way select comes from
			Yes	Valid	Read data	Din flops disabled
1	0	1	No	Valid	Read data	Way select comes from decoded
			Yes	Valid	Read data	Din flops disabled

TABLE 7-3 Way Select Decode During Read Operation

itb_icd_waysel_c[7:0]	agc_fill_wrway_bf[2:0]	ftp_asi_mbist_access_bf	Way Select
8'b0000_0001	Don't Care	0	0
8'b0000_0010	Don't Care	0	1
8'b0000_0100	Don't Care	0	2
8'b0000_1000	Don't Care	0	3

TABLE 7-3 Way Select Decode During Read Operation *(Continued)*

itb_icd_waysel_c[7:0]	agc_fill_wrway_bf[2:0]	ftp_asi_mbist_access_bf	Way Select
8'b0001_0000	Don't Care	0	4
8'b0010_0000	Don't Care	0	5
8'b0100_0000	Don't Care	0	6
8'b1000_0000	Don't Care	0	7
Don't Care	3'b000	1	0
Don't Care	3'b001	1	1
Don't Care	3'b010	1	2
Don't Care	3'b011	1	3
Don't Care	3'b100	1	4
Don't Care	3'b101	1	5
Don't Care	3'b110	1	6
Don't Care	3'b111	1	7

TABLE 7-4 Way Select Decode During Write Operation

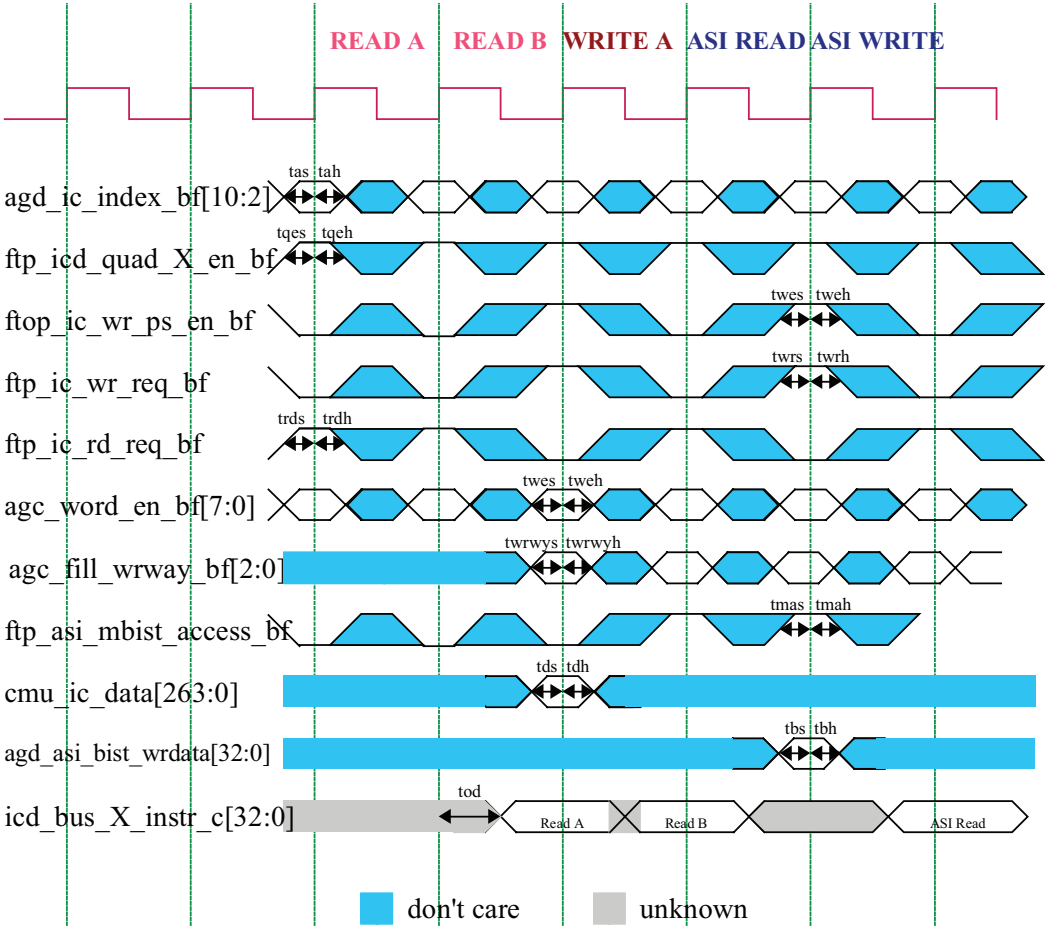
agc_fill_wr_way_bf[2:0]	Way Select
3'b000	0
3'b001	1
3'b010	2
3'b011	3
3'b100	4
3'b101	5
3'b110	6
3'b111	7

TABLE 7-5 Word Enable During Write Operation

agc_word_en_bf[7:0]	ftp_asi_mbist_access_bf	Data Source	Word to Write
8'b1111_1111	0	cmu_ic_data[263:0]	All
8'b1111_1111	1 – MBIST	agd_asi_bist_wrdata[32:0]	All
8'b0000_0001	1 – ASI	agd_asi_bist_wrdata[32:0]	0
8'b0000_0010	1 – ASI	agd_asi_bist_wrdata[32:0]	1
8'b0000_0100	1 – ASI	agd_asi_bist_wrdata[32:0]	2
8'b0000_1000	1 – ASI	agd_asi_bist_wrdata[32:0]	3
8'b0001_0000	1 – ASI	agd_asi_bist_wrdata[32:0]	4
8'b0010_0000	1 – ASI	agd_asi_bist_wrdata[32:0]	5
8'b0100_0000	1 – ASI	agd_asi_bist_wrdata[32:0]	6
8'b1000_0000	1 – ASI	agd_asi_bist_wrdata[32:0]	7

7.5 Timing Diagram

FIGURE 7-2 Read/Write Timing Diagram for ICD Array



CHAPTER 8

Instruction Cache Tag Array (ICT)

This chapter describes the following topics:

8.1 Functional Description

- OpenSPARC T2 L1 Instruction Cache Tag (ICT) is logically organized as 64 entry x 30 bits x 8 ways and it is 15360 bits/1920Bytes (1856B data and 64B parity).
- Each way contains 64 entries of 30 bits (29 bits data, 1 bit parity).
- It is a single-ported SRAM configured as 2 banks of 64 entries x 120 bits. Each bank is composed with 4 sub-blocks (16 x 120) and ioslice.
- Back-to-back read or write is possible.
- During a line fill, when a new Instruction cache(I$) line is being written to one way in the L1 I$, the same way is in the tag is also written into.
- During a read access, all 6 bits of the index are fully decoded to select one of 64 entries. All the ways in the tag are accessed at the same time (during instruction fetches). The tags are sent to the ITLB (Instruction cache TLB) to match against the physical address read from the ITLB. Also, the tags go through partial parity check and get flopped (in the datapath). The reminder of the parity checking is done in the following cycle.
- Read access and line fill are both single cycle operations.
- 30 bit flopped line fill data inputs are provided for a single cycle write into one out of the eight ways. The way to which the fill happens is selected by the wayselect signals.

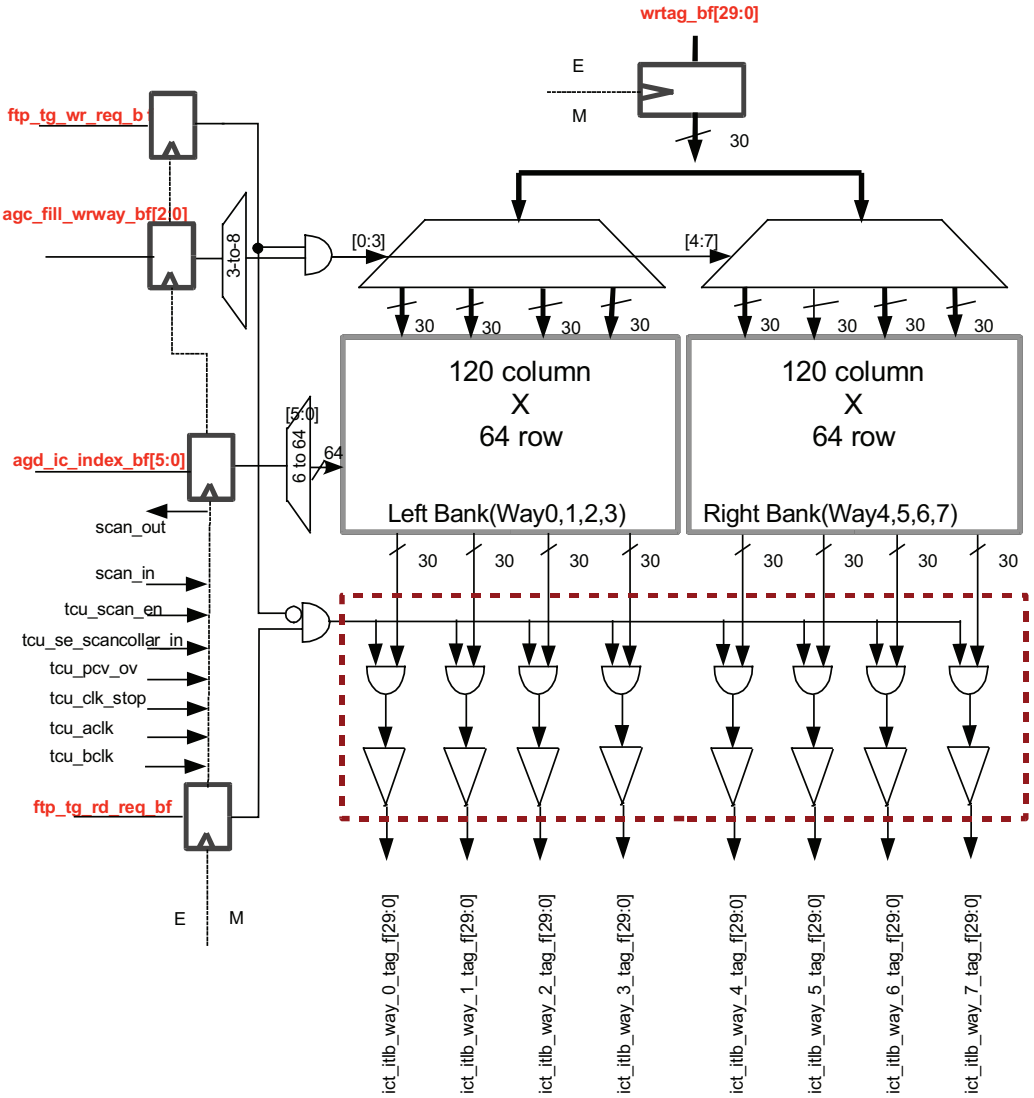
- Supports testing through BIST and macrotest. Ramtest is not supported.
- In scan /macrotest mode, 'tcu_wr_array_inhibit' is asserted during scan shifting to avoid writing in memory or any disturbance in memory circuits.
- Power saving feature is implemented. There're 2 l1clk domains for power saving which are controlled by 'pce' signals and l1clks are active only when they are needed. One of l1clk is not part of power saving which means 'pce' is tied to vdd.

In summary, the ICT outputs go to three separate location:

- To ITLB to be compared with the physical address.
- To the parity check logic (in datapath).
- To a mux for the ASI/MBIST reads.

8.2 Block Diagrams

FIGURE 8-1 ICT Block Diagram



8.3 I/O List

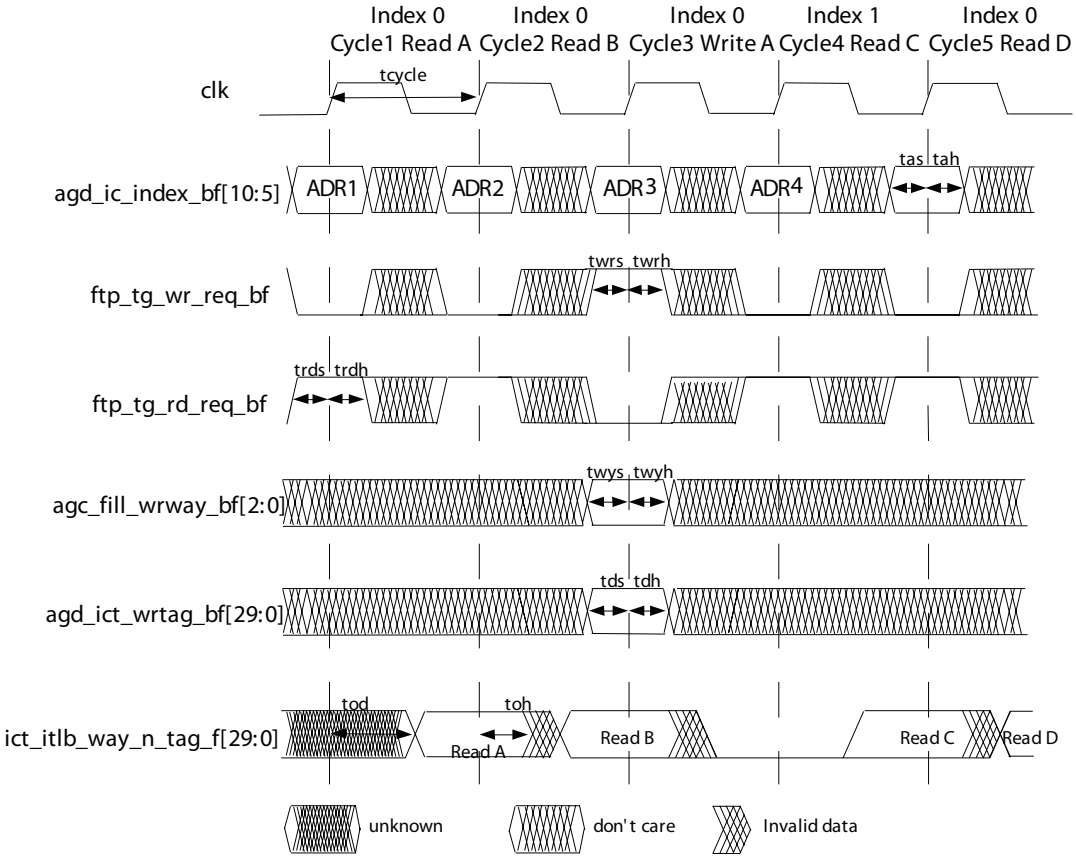
TABLE 8-1 I/O List

Signal Name	Width	I/O	Source/Destination	Description
agd_ic_index_bf	[10:5]	IN	datapath(agd)	read/write index (address input)
agd_fill_wrway_bf	[10:5]	IN	datapath(agc)	write way select
ftp_tg_rd_req_bf		IN		read request signal
ftp_tg_wr_req_bf		IN		write request signal
agd_ict_wrtag_bf	[29:0]	IN	datapath(agd)	linefill data input
ict_itlb_way_0_tag_f	[29:0]	OUT*	ITLB/datapath	Output tag w0 and parity
ict_itlb_way_1_tag_f	[29:0]	OUT*	ITLB/datapath	Output tag w1 and parity
ict_itlb_way_2_tag_f	[29:0]	OUT*	ITLB/datapath	Output tag w2 and parity
ict_itlb_way_3_tag_f	[29:0]	OUT*	ITLB/datapath	Output tag w3 and parity
ict_itlb_way_4_tag_f	[29:0]	OUT*	ITLB/datapath	Output tag w4 and parity
ict_itlb_way_5_tag_f	[29:0]	OUT*	ITLB/datapath	Output tag w5 and parity
ict_itlb_way_6_tag_f	[29:0]	OUT*	ITLB/datapath	Output tag w6 and parity
ict_itlb_way_7_tag_f	[29:0]	OUT*	ITLB/datapath	Output tag w7 and parity
l2clk		IN*		Global clock
tcu_pce_ov		IN*		Chip enable override, control l1 clk heaer
ftp_tg_clk_en		IN*		Clock enable signal for power saving
tcu_se_scancollar_in		IN*		Scan enable for input flops
tcu_scan_en		IN*		Scan enable for array
tcu_aclk		IN*		Scan in clock
tcu_bclk		IN*		Scan out clock
tcu_array_wr_inhibit		IN*		Memory write protection in scan mode
Scan_in		IN*	scan chain	Scan input
Scan_out		OUT*	scan chain	Scan output

Non-registered I/O is marked with an *

8.4 Timing Diagrams

FIGURE 8-2 Read/Write Timing Diagram of ICT Array



CHAPTER 9

Data Cache Array (DCA)

This chapter describes the following topics:

9.1 Functional Description

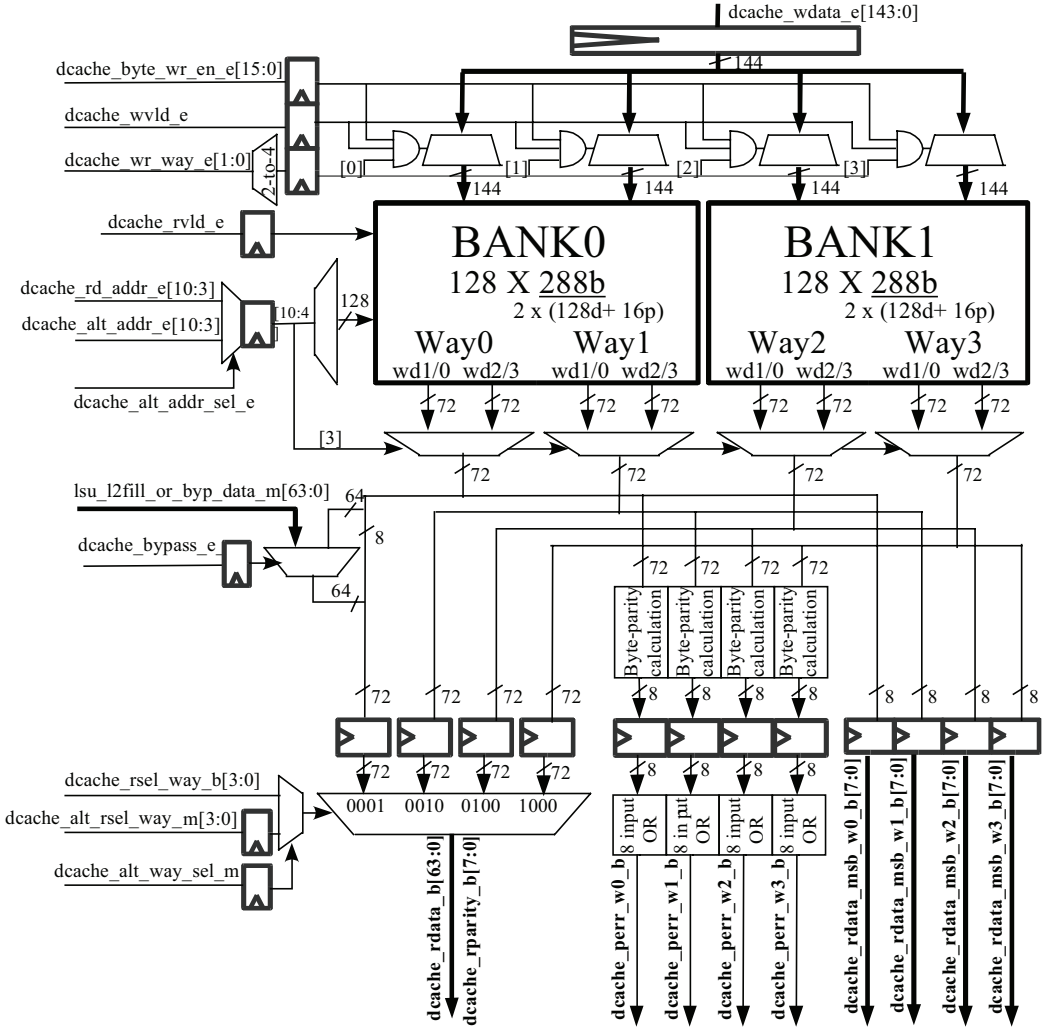
- OpenSPARC T2 uses a 9kByte (8KB data and 1KB parity), 16Byte line size, 4-way set associative data cache.
- Each way contains 128 entries of 144 bits. (16 bytes with parities)
- It is a single-ported SRAM configured as 2 banks of 128 entries x 288 bits. Each bank is composed with two sub-banks (128 x 144) and a common decoder.
- Parity protection is implemented as one parity per byte.
- Read operation has single cycle latency and single cycle throughput.
- From 8 bits address, 7 bits are decoded to 128 entries and 1 bit (LSB) is used to select upper or lower double words in each way. All four ways read out double word simultaneously and one of four ways selected by read way select sent to output bus.
- The output bus is not floating even all the read way select inputs are not active. In that case, all outputs will be '0's.
- With power saving feature, output data is preserved in write/noop cycles after read. Rdata_b & rparity_b go '0's in noop/write cycles through 4-to-1 mux(AOI) if rway_sel<3:0> is reset to 0's.

- The MSB's of each byte in each way are outputs in cycle 'B'.
- Parity calculation and check is provided. Final parity check result of each way is an output. So DCA sends out total of 4 parity error outputs.
- 144 bit write data inputs are provided for a single cycle write into the selected way.
- 16 byte write enable signals allow byte level writing from 16byte inputs.
- Back-to-back read or write is possible.
- Supports testing through BIST and macrotest. Ramtest is not supported.
- In scan /macrotest mode, 'tcu_wr_array_inhibit' is asserted during scan shifting to avoid writing in memory or any disturbance in memory circuits.
- Supports 4 redundant columns per sub-array. 4 columns are repaired together as a unit when needed. There's no redundant row available.
- Power saving feature is implemented. There're 3 l1clk domains which are controlled by 3 separate 'pce' signals and l1clks are active only when they are needed. Some of l1clks are not part of power saving which means 'pce' is tied to Vdd.
- A 144-bit write data bus is provided for a single-cycle write into either of the banks. The write way inputs select which one of the ways to write in the cycle.
- Sixteen-byte write-enable signals enable writing to any or all of the 16 bytes.
- Write is inhibited when scan is enabled.
- A read or write can occur during any cycle, but they cannot both occur during the same cycle.
- The address input to the cache can be selected from a normal or alternate bus through a 2-to-1 mux. The way select for read can be similarly selected.
- Supports testing through built-in self test (BIST), macrotest, and ramtest.
- Design and layout are shared with the floating-point register file (FRF), the instruction cache data (ICD), and Level 2 cache tag array (L2 Tag).

9.2 Block Diagram

This section provides a functional block diagram of the data cache array.

FIGURE 9-1 DCA Block Diagram



9.3 I/O List

TABLE 9-1 I/O List

Signal name	Width	I/O	Flopped	Source / Destination	Description
Functional Signals					
dcache_rd_addr_e	[10:3]	IN	YES	EXU	read cache index [10:4] + bit [3] offset
dcache_alt_addr_e	[10:3]	IN	YES	lsu_dcc_ctl	write/bist/diagnostic read cache index + offset
dcache_alt_addr_sel_e		IN	YES	lsu_dcc_ctl	address select (1:alt. Addr., 0: normal addr.)
dcache_rvld_e		IN	YES	lsu_dcc_ctl	read accesses dcache
dcache_wvld_e		IN	YES	lsu_dcc_ctl	write accesses dcache
dcache_wdata_e	[143:0]	IN	YES	lsu_dcp_dp	write data 16byte data + 16bit parities
dcache_wr_way_e	[1:0]	IN	YES	lsu_dcc_ctl	replacement way for load miss/store (encoded)
dcache_byte_wr_en_e	[15:0]	IN	YES	lsu_dcc_ctl	16b byte wr enable for stores
dcache_rsel_way_b	[3:0]	IN	NO	lsu_tlb_oust	load way select, connect to cache_way_hit
dcache_alt_rsel_way_m	[3:0]	IN	YES	lsu_dcc_ctl	bist/diagnostic read way select
dcache_alt_way_sel_m		IN	YES	lsu_dcc_ctl	read way select (1 for alt. rsel, 0 for normal rsel)
lsu_l2fill_or_byp_data_m	[63:0]	IN	YES	lsu_lmd_dp	L2fill data for bypass. Mux with way0 rdata.
dcache_bypass_e_		IN	YES	lsu_dcc_ctl	bypass select (1 for way0 rdata, 0 for l2fill)
dcache_rdata_b	[63:0]	OUT	YES/NO	lsu_dcd_dp	read data out. (8 bytes, selected way)
dcache_rparity_b	[7:0]	OUT	YES/NO	lsu_dcs_dp	read parity out. (8 bits, selected way)
dcache_perr_w0_b		OUT	YES	lsu_dcc_ctl	way0 parity error

TABLE 9-1 I/O List *(Continued)*

Signal name	Width	I/O	Flopped	Source / Destination	Description
dcache_perr_w1_b		OUT	YES	lsu_dcc_ctl	way1 parity error
dcache_perr_w2_b		OUT	YES	lsu_dcc_ctl	way2 parity error
dcache_perr_w3_b		OUT	YES	lsu_dcc_ctl	way3 parity error
dcache_rdata_msb_w0_b	[7:0]	OUT	YES	lsu_dac_ctl	way0 rdata msb's
dcache_rdata_msb_w1_b	[7:0]	OUT	YES	lsu_dac_ctl	way1 rdata msb's
dcache_rdata_msb_w2_b	[7:0]	OUT	YES	lsu_dac_ctl	way2 rdata msb's
dcache_rdata_msb_w3_b	[7:0]	OUT	YES	lsu_dac_ctl	way3 rdata msb's
Clock and Scan Related Signals					
l2clk		IN	NO	TCU	clock
dcache_clk_en_e		IN	NO	lsu_dcc_ctl	array clock enable
dcache_wclk_en_e		IN	NO	lsu_dcc_ctl	write data/byte_wr_en flops clock enable
dcache_rclk_en_m		IN	NO	lsu_dcc_ctl	read flops clock enable
tcu_pce_ov		IN	NO	TCU	chip enable override. clkhdr is enabled in scan
tcu_se_scancollar_in		IN	NO	TCU	scan enable for input flops
tcu_se_scancollar_out		IN	NO	TCU	scan enable for output flops
tcu_scan_en		IN	NO	TCU	scan enable for memory control logics
tcu_aclk		IN	NO	TCU	scan in clock
tcu_bclk		IN	NO	TCU	scan out clock
tcu_array_wr_inhibit		IN	NO	TCU	memory write protection in scan mode
scan_in		IN	NO	lsu_dac_ctl	scan input
scan_out		OUT	NO	dva_dp_32x32 _cust	scan output
Redundancy repair related signals					
fuse_dca_repair_value	[5:0]	IN	YES	lsu_red_ctl	fuse repair values
fuse_dca_repair_en	[1:0]	IN	YES	lsu_red_ctl	fuse repair enables
fuse_dca_rid	[1:0]	IN	YES	lsu_red_ctl	fuse repair register address
fuse_dca_wen		IN	YES	lsu_red_ctl	repair register write (1 for write, 0 for read)

TABLE 9-1 I/O List *(Continued)*

Signal name	Width	I/O	Flopped	Source / Destination	Description
fuse_red_reset		IN	YES	lsu_red_ctl	repair register reset
dca_fuse_repair_value	[5:0]	OUT	YES	lsu_red_ctl	fuse repair values from repair registers
dca_fuse_repair_en	[1:0]	OUT	YES	lsu_red_ctl	fuse repair enables from repair registers
l2clk		IN	NO	TCU	clock
dcache_clk_en_e		IN	NO	lsu_dcc_ctl	array clock enable
dcache_wclk_en_e		IN	NO	lsu_dcc_ctl	write data/byte_wr_en flops clock enable
dcache_rclk_en_m		IN	NO	lsu_dcc_ctl	read flops clock enable

9.4 Functional Table

TABLE 9-2 Functional Table

Inputs					Outputs	Operation
rv	wvld_e	bypass_e_	alt_addr_sel	alt_rsel_way[3:0]	Rdata[63:0]	rvld_e
1	1	x	x	x	0	Not allowed. If happens, same as Noop. wcs & sae are disabled.
0	1	x	1	x	0	Write Operation
1	0	1	0	0	valid	Read Operation in normal mode.
1	0	1	1	valid	valid	Read Operation in test mode.
0	x	0	1	valid	valid	Bypass Operation*
0	0	1	x	x	0	Noop.\

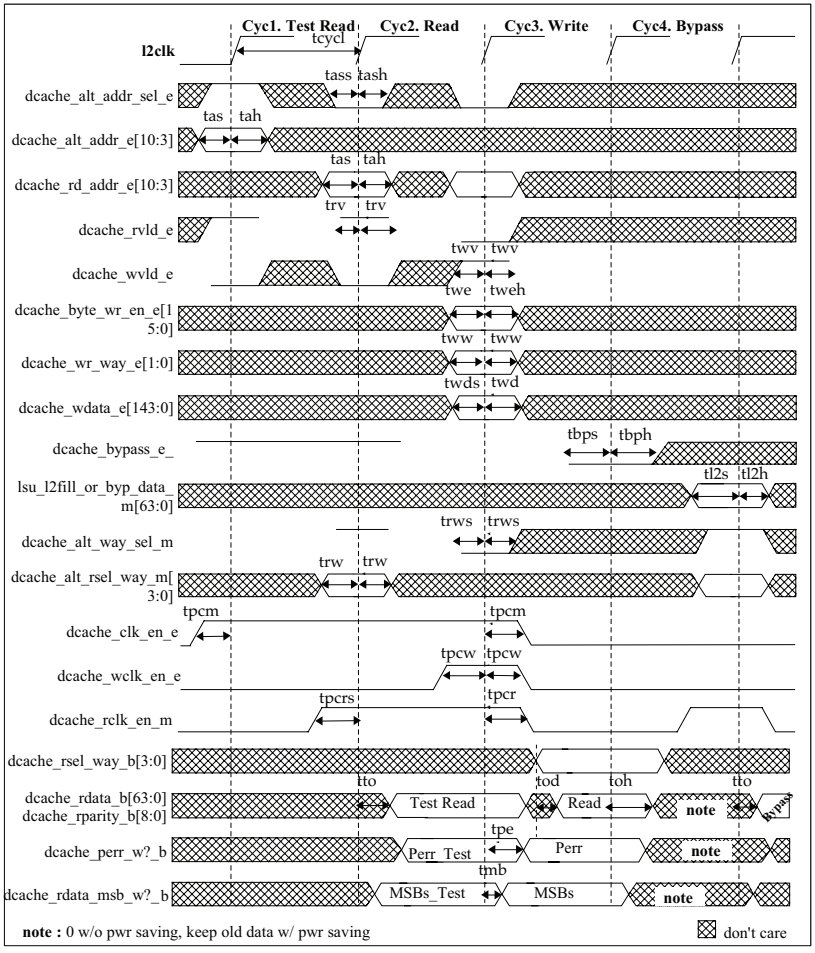
* To the DCA point of view, bypass operation is performed as long as dcache_bypass_e_ is '0', but dcache_bypass_e_ is not an indicator for bypass operation. This means tt could also be '0' during Noop.

\ There's no special constraints for control signals except for rsel_way[3:0]/alt_rsel_way[3:0], which are one-hot signal in normal operation. If multiple -hot happens for these signals, outputs are not valid and not used.

9.5 Timing Diagram

The following figure provides a read/write input/output timing diagram for the data cache array.

FIGURE 9-2 I/O Timing



CHAPTER 10

Data Cache Tag Array (DTA)

This chapter describes the following topics:

10.1 Functional Description

- OpenSPARC T2 L1 Data Cache Tag (DTA) is logically organized as 128 entry x 30 bits x 4 ways and it is 15360 bits / 1920Bytes (1856B data and 64B parity). There is one parity bit for 29 tag bits.
- Each way contains 128 entries of 30 bits. (29 bits data, 1 bit parity)
- It is a single-ported SRAM configured as 2 banks of 64 entries x 120 bits. Each bank is composed with 4 sub-block (16 x 120) and ioslice.
- The 2 x 30 bits (29b tag + 1b parity) of each way are contained in 2 banks (left and right). Each bank stores 60 bits of tag entries for 2 ways which gives a total of 120 bits per bank.
- Back-to-back read or write is possible.
- During a line fill, when a new Data cache(I$) line is being written to one way in the L1 D$, the same way is in the tag is also written into.
- During a read access, all 7-bits are muxed and flopped, 6-bits of 7-bits from read index are fully decoded to select one of 64 entries. the LSB of read index selects which 30 bits from each way should read out. All the ways in the tag are accessed

at the same time. The tags are sent to the TLB (Data cache TLB) to match against the physical address read from the TLB. Also, the tags go through partial parity check and get flopped (in the datapath).

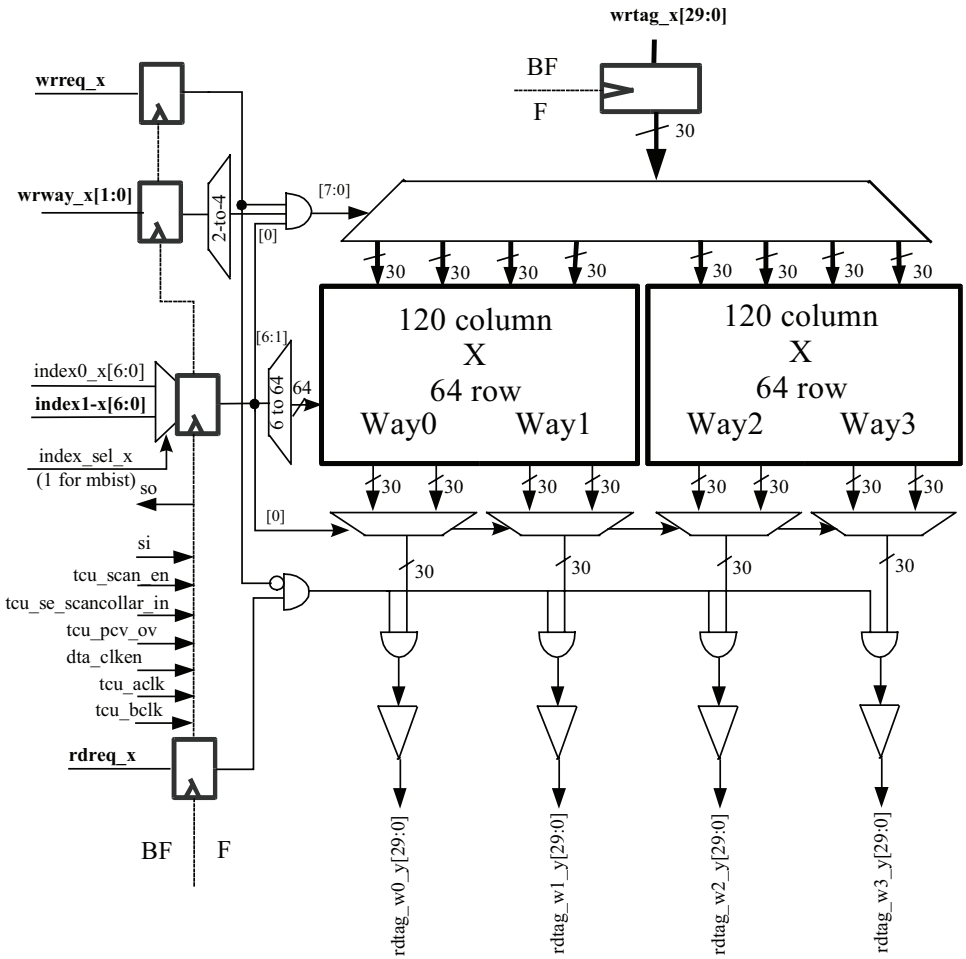
- Read access and line fill are both single cycle operations.
- Back to back reads and writes are allowed.
- Supports testing through BIST and macrotest.
- In scan /macrotest mode, 'tcu_wr_array_inhibit' is asserted during scan shifting to avoid writing in memory or any disturbance in memory circuits.
- Power saving feature is implemented. There're 2 l1clk domains for power saving which are controlled by 'pce' signals and l1clks are active only when they are needed. One of l1clk is not part of power saving which means 'pce' is tied to vdd.

In summary, the DTA outputs go to three separate location:

- To TLB to be compared with the physical address .
- To the parity check logic (in datapath)
- To a mux for the read diagnosis (in datapath).

10.2 Block Diagrams

FIGURE 10-1 DTA Block Diagram



10.3 I/O List

TABLE 10-1 DTA I/O List

Signal name	Width	I/O	Source / Destination	Description
index0_x	[6:0]	IN	EXU	Address inputs
index1_x	[6:0	IN	lsu_dcc_ctl	Address inputs
index_sel_x		IN*	lsu_dcc_ctl	Address input mux select
wrway_x	[1:0	IN	lsu_dcc_ctl	Write way select
wrtag_x	[29:0	IN	lsu_dcc_ctl/ lsu_lmd_dp	Linefill data input
rdreq_x		IN	lsu_dcc_ctl	Read request signal
wrreq_x		IN	lsu_dcc_ctl	Write request signal
rdtag_way0_y	[29:0	OUT*	TLB/datapath	
rdtag_way1_y	[29:0	OUT*	TLB/datapath	
rdtag_way2_y	[29:0	OUT*	TLB/datapath	
rdtag_way3_y	[29:0	OUT*	TLB/datapath	
l2clk		IN*		
tcu_pce_ov		IN*		chip enable override, control l1clk_header
dta_clken		IN*		power saving for array and datapath
tcu_se_scancollar_in		IN*		scan enable for input flops
tcu_scan_en		IN*		scan enable for array
tcu_aclk		IN*		scan in clock
tcu_bclk		IN*		scan out clock
tcu_array_wr_inhibit		IN*		memory write protection in scan mode
scan_in		IN*		scan input

10.4 Functional Table

TABLE 10-2 DTA Modes of Operation

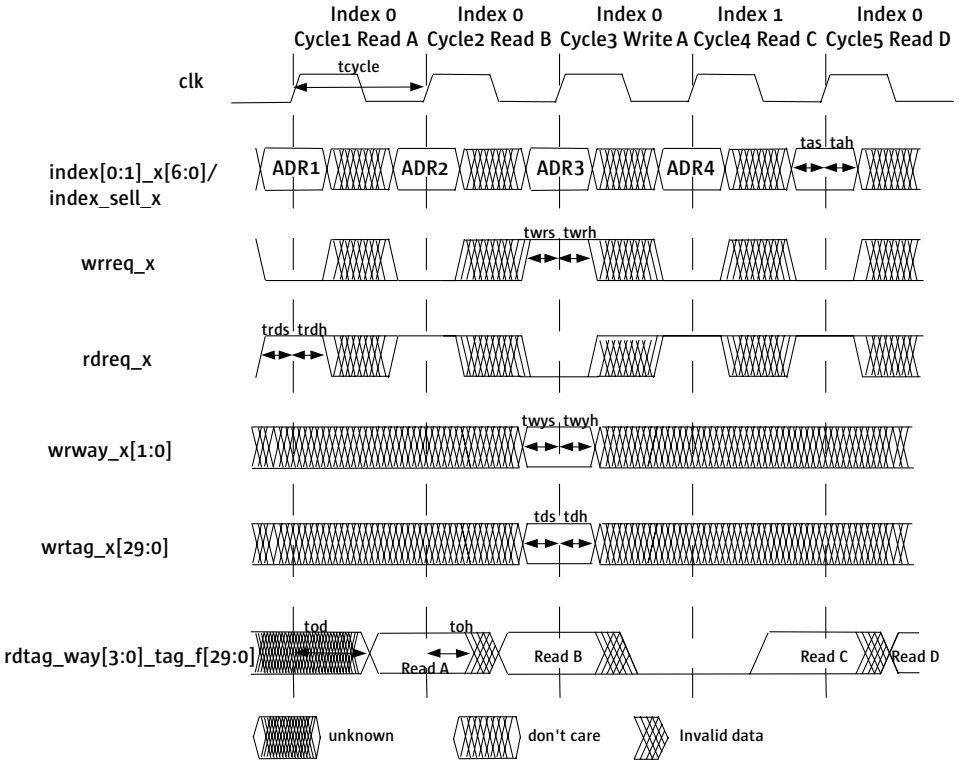
rd_req	wr_req	ict_itbl_way[7:0]_tag_f{29:0] (ckt output)	Description
1	1	0	Even though this is an invalid condition, the write takes priority, and the read is invalidated. No read data output is generated . Precharge will follow.
0	1	0	Normal write cycle
1	0	valid	Normal read cycle, perform read according to input control signals
0	0	0	No-operation(no-op). Wordline will fire, but since the write column switches are off,memory contents are not affected. Precharge will follow.

TABLE 10-3 DTA Write Way Select Decode

wrway_x[1:0]	Way select for write
0 0	0
0 1	1
1 0	2
1 1	3

10.5 Timing Diagram

FIGURE 10-2 Read/Write Timing Diagram of DTA



CHAPTER 11

Data Valid Bit Array (DVA)

This chapter describes the following topics:

11.1 Functional Description

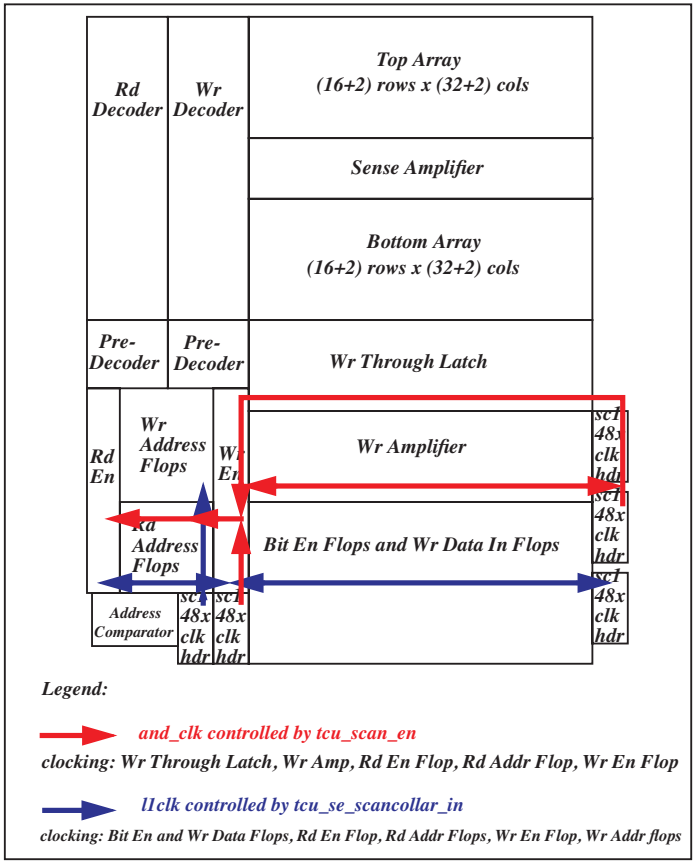
- The dva_dp_32x32 is a dual port (1r1w) register file used for storing the valid bits of the instruction cache and data cache. It is also used for storing the LRU bits of the data cache, which implements a true LRU policy. The selective write through allows for predictability on the data accessed from the array.
- The dva_dp_32x32_cust is physically structured as 2 sub-arrays of 512 bits each (16x32 array).
- Designed to meet the speed target of 1.4 GHz (714ps cycle time or 357ps phase time)
- All inputs are flopped. Read address and Read enable are captured with a gated latch cl_mc1_sram_msff_mo_16x from mc1_l.
- Write address and Write enable are captured with a standard cell library MSFF.
- Read operation occurs during the A-phase of l2clk. Read data is captured by a glitch latch. Single ended full swing read bitline is used.
- Write operation occurs during the B-phase of l2clk. Differential full swing write bitlines. Bitwise write is supported.

- The block has a selective write through mode which is enabled when the read and write addresses are equal. The read data is a logical AND between the previous data and the write through data. The write through data are logically HIGH when the new data being written are '1' or when not writing.

11.2 Block Diagrams

This section provides a functional block diagram of the data valid array.

FIGURE 11-1 DVA Block Diagram

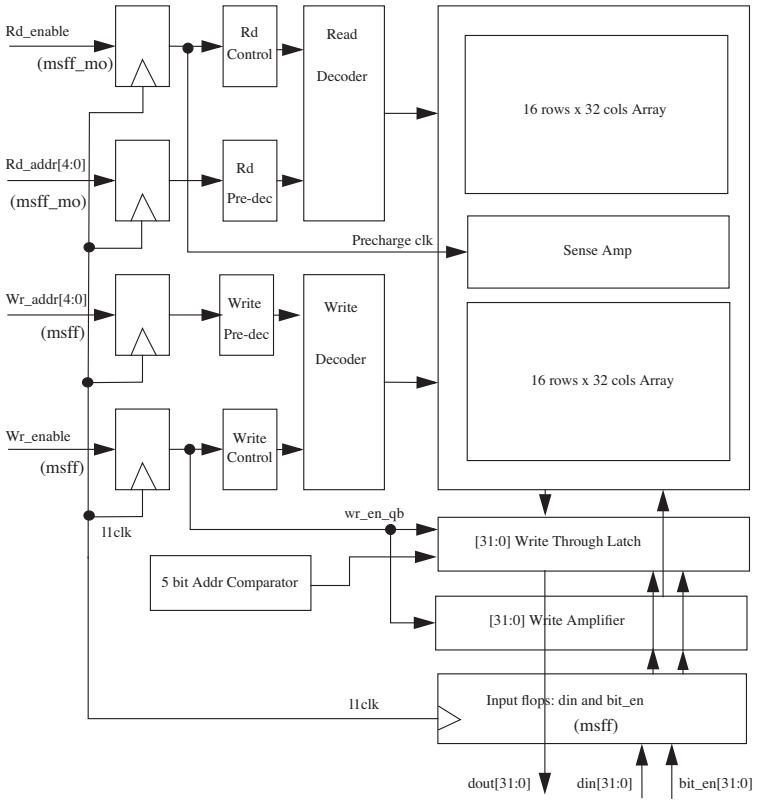


11.3 I/O List

TABLE 11-1 DVA I/O List

Signal name	Width	I/O	Description
Functional Signals			
l2clk		Input	input clock
rd_addr	[4:0]	Input	read address (Flopped - NAND gate)
rd_en		Input	read enable (Flopped - NAND gate)
wr_addr	[4:0]	Input	write address (Flopped)
din	[31:0]	Input	write data (Flopped)
bit_wen	[31:0]	Input	bit write enable (Flopped)
wr_en		Input	write enable for 32 bits(Flopped)
dout	[31:0]	Output	read data out (Latched)
Test Related Signals			
tcu_se_scancollar_in		Input	scan enable for l1clk hdr
tcu_scan_en		Input	scan_enable for and_clk hdr
scan_in		Input	scan input data
tcu_pce_ov		Input	test control signal for l1clk hdr
tcu_aclk		Input	scan input clock
tcu_bclk		Input	scan output clock
scan_out		Output	scan output
tcu_array_wr_inhibit		Input	Read/Write Inhibit
pce		Input	test control signal for l1clk hdr

FIGURE 11-2 DVA Logic Symbol



11.4 Functional Table

TABLE 11-2 DVA Functional Table

rd_addr = wr_addr	bitwen	wr_en	rd_en	wt_data	operation (RTL and Circuit)
0	x	0	0	1	No-op; dout[31:0]=0
0	x	0	1	1	read
0	1/0	1	0	1	write to bits w/ bitwen=1
0	1/0	1	1	1	read; write
1	x	0	0	~bitwen | din | ~wren	no operation
1	1/0	0	1	~bitwen | din | ~wren	rd:wt_data&array_data
1	1/0	1	0	~bitwen | din | ~wren	write to bits w/ bitwen=1
1	1/0	1	1	~bitwen | din | ~wren	rd: wt_data&array_data;write

When tcu_array_wr_inhibit =1, it forces Rd_en=Wr_en=0 and dout[31:0]=0

11.5 Timing Diagram

The following figure provides a read/write input/output timing diagram for the data valid array.

FIGURE 11-3 DVA I/ORead Timing

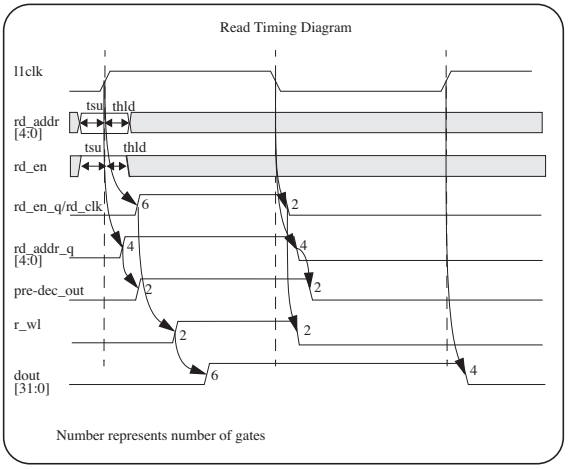


FIGURE 11-4 DVA I/O Write Timing

CHAPTER 12

L2-Cache Tag Array

This chapter describes the following topics:

12.1 Functional Description

This section describes Level 2 cache tag array (L2 Tag) physical descriptions and functions.

- OpenSPARC T2 L2 Tag has 224KB of 16-way set-associative Level 2 cache tag array. This is split into 8 banks of 28KB each.
- L2T Array is a single-ported SRAM and has a logical organization of 16 ways x 512 entries x 28bits/entry.
- Each 28 bit tag entry is split into 22 tag data bits, 5 bits ECC and 1 bit parity.
- Each L2T bank is built using 8 arrays. Each array has 128 rows and 224 (split into 120 and 104) columns and stores 512 tags for 2 different ways.

L2T supports two operations: Lookup (Read) access and Write access.

During a read access, index[8:0] are used to select one of 512 entries for each of the 16 ways. Data is read from all 16 ways.

Data output from each way, consisting of address bits [27:6] and ECC bits [5:1], is compared with the lookup tag address (lkup_tag[27:1]) to generate a 1 bit way_sel signal for each way. Parity bits (bit[0]) are not compared.

lkup_tag[27:1] input is not registered inside the L2 tag. All other inputs are registered internally. lkup_tag[27:1] signals are timed as having a setup time w.r.t the negative edge of the clock as they go to the dynamic comparator.

During a write access, an entry in the tag ram specified by index[8:0] and way[15:0] is updated with the wrdata[27:0].

Single cycle back to back read and write operations are allowed.

In case of multiple way hit, L2T will produce multiple way_hit, but it will not detect the occurrence.

All outputs are flopped at each array boundary.

All input/output latches and flops (including redundancy flops) have built-in scan.

We provide 8 redundant columns per 224 columns for each array. Column redundancy is implemented after the sense amplifier, hence this amounts to 2 redundant bits for every 56 bits in each array.

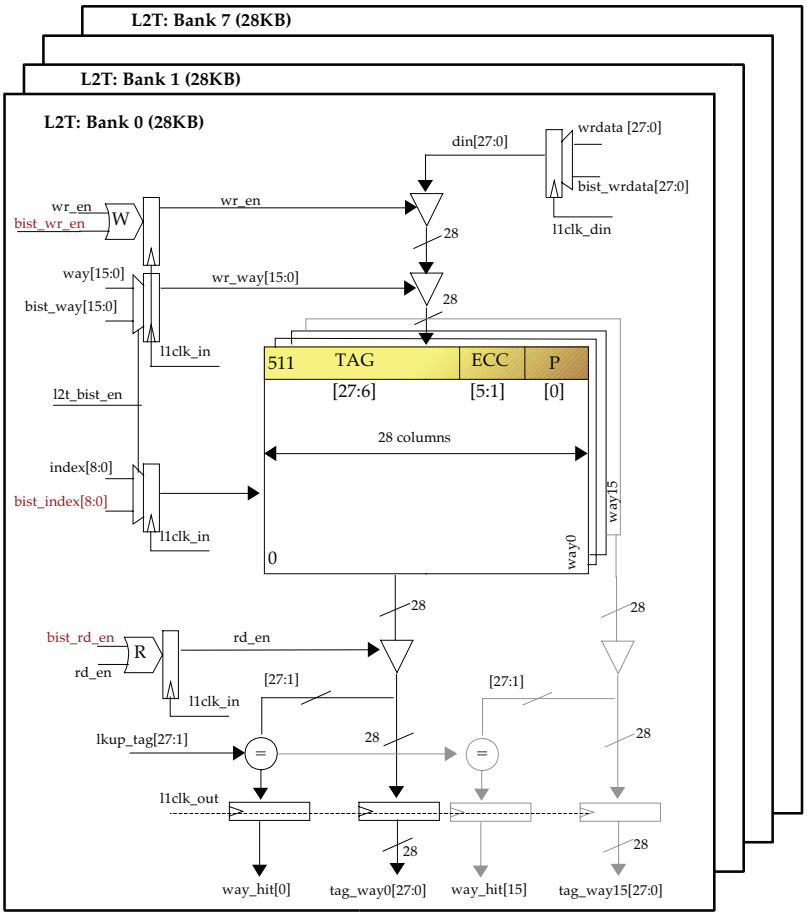
Row redundancy is not implemented for L2T.

Write is inhibited using w_inhibit signal during scan operation.

Along with scan, L2Tag array supports BIST. The BIST and functional ports are muxed and sent to the tag array.

12.2 Block Diagram

FIGURE 12-1 L2 Tag Array Block Diagram



12.3 I/O List

The following table describes the L2 Tag I/O signals.

TABLE 12-1 Functional and Redundancy Repair Related I/O Signal List .

Signal	Width	I/O	Source or Destination	Description
Functional I/O Signal List				
l2clk		Input	lib_clust_hdr_cust	Input Clock
index0	[8:0]	Input	l2t_tagd_dp	Address input for normal read/write
index1	[8:0]	Input	l2t_tagd_dp	Address input for normal read/write
bist_index0	[8:0]	Input	l2t_tagd_dp	Address input used during BIST operation
bist_index1	[8:0]	Input	l2t_tagd_dp	Address input used during BIST operation
rd_en0		Input	l2t_tagd_dp	Read enable control input for normal read
rd_en1		Input	l2t_tagd_dp	Read enable control input for normal read
bist_rd_en0		Input	l2t_tagd_dp	Read enable control input during BIST operation
bist_rd_en1		Input	l2t_tagd_dp	Read enable control input during BIST operation
way	[15:0]	Input	l2t_tagd_dp	Way select during a fill/tag write
bist_way	[15:0]	Input	l2t_tagd_dp	Way select during BIST operation
wr_en0		Input	l2t_tagd_dp	Write enable control input
wr_en1		Input	l2t_tagd_dp	Write enable control input
bist_wr_en0		Input	l2t_tagd_dp	Write enable control used during BIST opeartion
bist_wr_en1		Input	l2t_tagd_dp	Write enable control used during BIST opeartion
wrdata0	[27:0]	Input	l2t_tagd_dp	Tag data to be written into the RAM
wrdata1	[27:0]	Inout	l2t_tagd_dp	Copy of wrdata0[27:0]
bist_wrdata0	[27:0]	Input	l2t_tagd_dp	Tag write data during BIST operation
bist_wrdata1	[27:0]	Input	l2t_tagd_dp	Copy of bist_wrdata0[27:0]
lkup_tag0		Input	l2t_tagd_dp	Compare data input to the comparator.
lkup_tag1		Input	l2t_tagd_dp	Compare data input to the comparator.
l2t_bist_en0		Input	l2t_tagd_dp	Bist enable control
l2t_bist_en1		Input	l2t_tagd_dp	Bist enable control

TABLE 12-1 Functional and Redundancy Repair Related I/O Signal List *(Continued)*.

way_hit	[15:0]	Output	l2t_tagl_dp	Tag compare output for each way
tag_way	[27:0]	Output	l2t_tagl_dp	Tag data output for all the 16 ways
Power Saving Feature Related Signals				
clk_en0		Input	l2t_tagd_dp	Enable signal (pce) for control flops
clk_en1		Input	l2t_tagd_dp	Enable signal (pce) for control flops
clk_en_ov		Input	l2t_tagd_dp	Override enable signal (pce) for control flops
wr_en_ov		Input	l2t_tagd_dp	Override enable signal (pce) for data input flops
Redundancy repair related signals				
hdr_l2t_rvalue	[5:1]	Input	sram header	Fuse repair values
hdr_l2t_rvalue	[0:0]	Input	sram header	Fuse repair enable (duplicated internally)
hdr_l2t_rid	[3:0]	Input	sram header	Fuse repair register address.
hdr_l2t_wr_en		Input	sram header	Repair register write enable.
hdr_l2t_red_clr		Input	Sram header	Repair register reset
l2t_hdr_read_data	[5:1]	Output	Sram header	Fuse repair value from repair registers.
l2t_hdr_read_data	[0]	Output	Sram header	Fuse repair enable from repair registers.
Scan Related Signals				
tcu_pce_ov		Input	lib_clust_hdr_cust	Chip enable override, controls L1 clk. header
tcu_aclk0		Input	lib_clust_hdt_cust	Scan in clock.
tcu_aclk1		Input	lib_clust_hdt_cust	Scan in clock.
tcu_bclk0		Input	lib_clust_hdt_cust	Scan out clock.
tcu_bclk1		Input	lib_clust_hdr_cust	Scan out clock.
tcu_scan_en0		Input	lib_clust_hdt_cust	Scan enable for internal flops/latches
tcu_scan_en1		Input	lib_clust_hdt_cust	Scan enable for internal flops/latches.
tcu_se_scancollar_in0		Input	lib_clust_hdt_cust	Scan enable for input flops
tcu_se_scancollar_in1		Input	lib_clust_hdt_cust	Scan enable for input flops
tcu_se_scancollar_out0		Input	lib_clust_hdt_cust	Scan enable for output flops
tcu_se_scancollar_out1		Input	lib_clust_hdt_cust	Scan enable for output flops
scan_in		Input	l2t_usaloc_dp	Scan input
scan_out		Output	l2t_tagl_dp	Scan out

TABLE 12-1 Functional and Redundancy Repair Related I/O Signal List *(Continued)*.

w_inhibit0		Input	lib_clust_hdt_cust	Write inhibit signal (during scan)
w_inhibit1		Input	lib_clust_hdt_cust	Write inhibit signal (during scan)
Unused Signals (terminated internaly)				
pce		Input	lib_clust_hdr_cust	Chip enable override, controls L1 clk. header
tcu_clk_stop		Input	TCU	Clock kill signal.

12.4 Functional Operation

TABLE 12-2 Functional Operation Table

Normal Operation (l2t_bist_en = 0)										Description	Circuit behaviour during illegal / noop case
Inputs								Outputs			
Index /bist_	rd_ en	bist _rd _en	wr_ en	bist _wr _en	din/bi st_din	way[15: 0]/bist_ way[15:	w_inhi bit	way_hi t[15:0]	tag_wayn[27:0]		
X/X	0	0	1	0	X/X	0/X	0	0	Previous Data	Data not written, no way selected	
Valid/X	0	0	1	0	Valid/ X	Valid/X	0	0	Previous Data	Data written to selected way	
Valid/X	1	0	0	0	X/X	X/X	0	Valid	Valid	Data read out from all ways	
X/X	0	0	0	0	X/X	X/X	0	0	Previous Data	No operation	No read or write
X/X	1	0	1	0	X/X	X/X	0	FFFF	Previous Data	Illegal operation	No read or write (*)
X/X	0	0	1	0	X/X	X/X	1	0	Previous Data	Write is disabled during scan	
X/X	1	0	0	0	X/X	X/X	1	FFFF	Previous Data	Read is disabled during scan	
Valid/X	0	0	1	1	Valid/ X	Valid/X	0	0	Previous Data	Data written to selected way	Normal write
Valid/X	1	1	0	0	X/X	X/X	0	Valid	Valid	Data read out from all ways	Normal read
Bist Operation (l2t_bist_en = 1)											
X/Valid	X	0	X	1	X/Vali d	X/Valid	0	0	Previous Data	Data written to selected way	
X/Valid	X	1	X	0	X/X	X/X	0	Valid	Valid	Data read out from all ways	
X/Valid	X	1	X	1	X/X	X/X	0	FFFF	Previous Data	Illegal operation	No read or write (*)
X/Valid	X	0	X	1	X/Vali d	X/Valid	1	0	Previous Data	Write is disabled by w_inhibit	
X/Valid	X	1	X	0	X/X	X/X	1	FFFF	Previous Data	Read is disabled by w_inhibit	
Inconsistent Implementation for normal mode operation (l2st_bist_en=0) (See Note 1)											
Valid/X	0	1	1	0	Valid/ X	X/X	0	Valid	Updated	Data read from all ways	Read (Note2)
Valid/X	1	0	0	1	X/X	Valid/X	0	0	Previous Data	Data written to valid ways	Write(Note2)

Note – (*)Inside the L2Tag, the read operation is gated with wr_en, and write operation is gated with rd_en. This ensures that we do neither a read nor a write during any cycle in which both rd_en and wr_en are simultaneously.

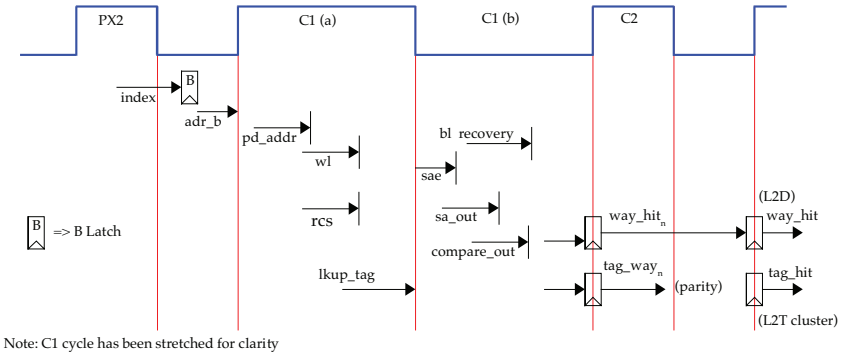
Note – For the case when bist mode read/write control signals (bist_rd_en and bist_wr_en) are asserted during functional mode (l2t_bist_en =0), the rtl and schematic implementation is same. In this case priority is given to bist operation. However, this is inconsistent with the other blocks, where the priority is given to functional mode signal.

12.5 Timing Diagrams

The following figures show L2-cache tag read and write timing diagrams.

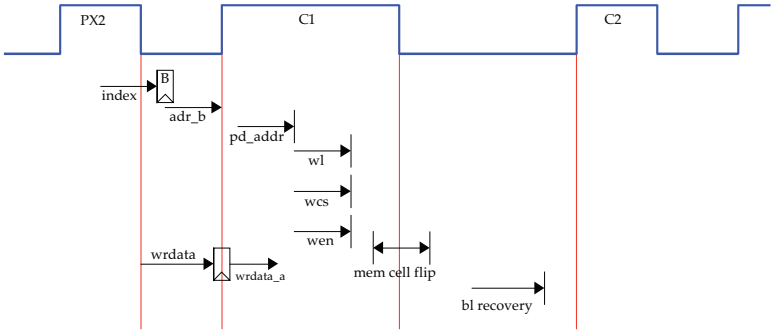
12.5.1 Read Timing Diagram

FIGURE 12-2 I/O Read Timing Diagram of L2 Tag



12.5.2 Write Timing Diagram

FIGURE 12-3 I/O Write Timing Diagram of L2 Tag



CHAPTER 13

L2-Cache Data Array

This chapter describes the following topics:

13.1 Functional Description

- OpenSPARC T2 L2 cache is incusive, 4 MB in size, and is composed of 8 symmetrical blocks.
- Each L2 bank (l2d_sp_512kb_cust) contain 512 entries 16-way set associative 64 Byte cache line as part of L2 Data
- Each L2 bank also contains L2 Tag, L2 VUAD, Input Queue, Output Queue, SIU Queue, Directory, Miss Buffer, Fill Buffer, Writeback Buffer, Snoop Response Buffer, physically located in the l2tag cluster.
- More than 1 bank can be accessed at the same time.
- Each L2 data array bank is a single ported structure that support the following operation:
- 16 Byte/ 64 Byte read.
- 4 Byte / 8 Byte / 64 Byte write with combination of any word enable.
- Each L2 bank is further divided into 4 logical sub-banks. Only one of the logical sub-banks is turned on for 16 Byte access.
- L2 Cache data array accesses takes 3 cycles (c4 c5 c52).
- The logical sub-banks 0,2 and 1,3 are grouped together.

- All access can be pipelined except, back to back accesses to the same logical sub-bank group.
- Throughput is single cycle for different sub-bank group access and one every two cycles for the same sub-bank group (0,2 or 1,3) access.
- The 64 Byte accesses need to be preceeded and followed by an access bubble. This is architecturally transparent for the L2d array.
- Each 32 bits word is protected by 7 bits of ECC.
- Physically, each logical sub-bank group is partitioned into 8 blocks of size 32 kB each.

13.2 Block Diagrams

FIGURE 13-1 L2 Cache Data Array Block Diagram

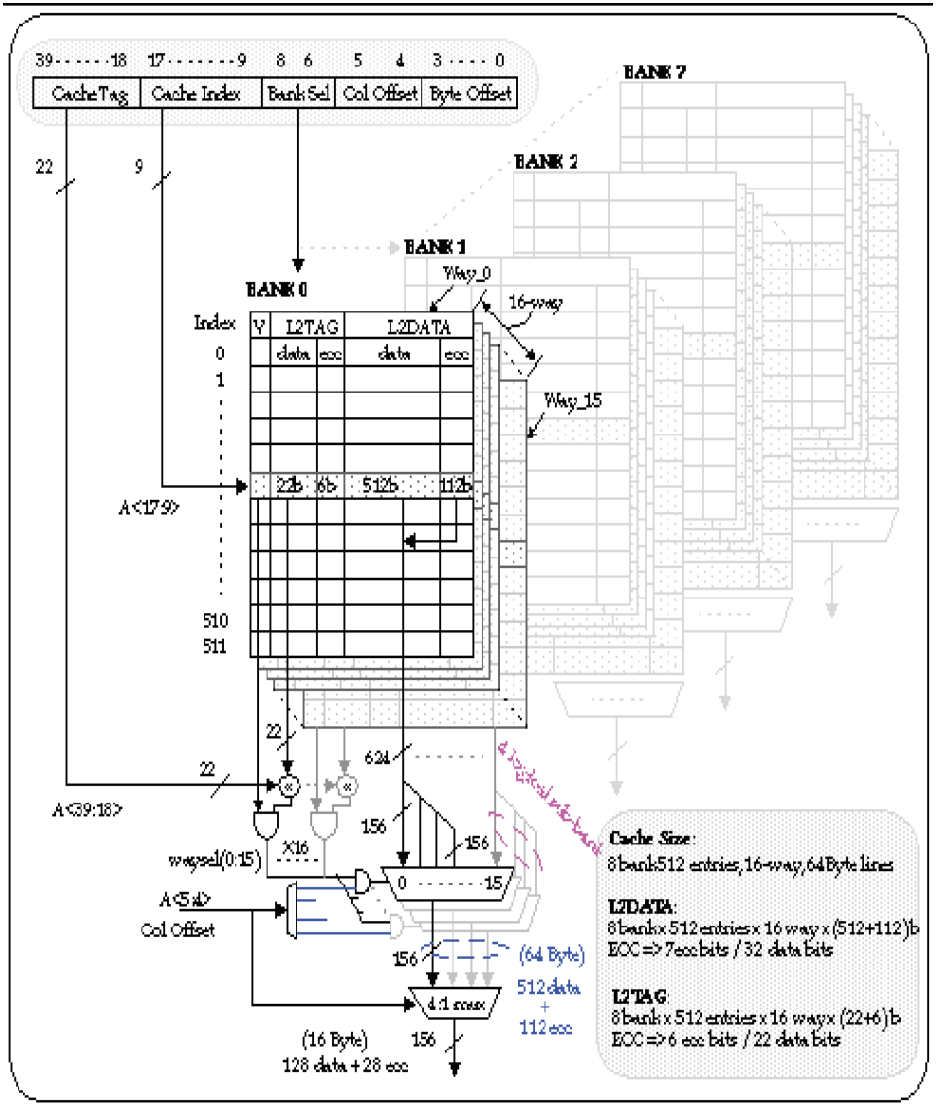
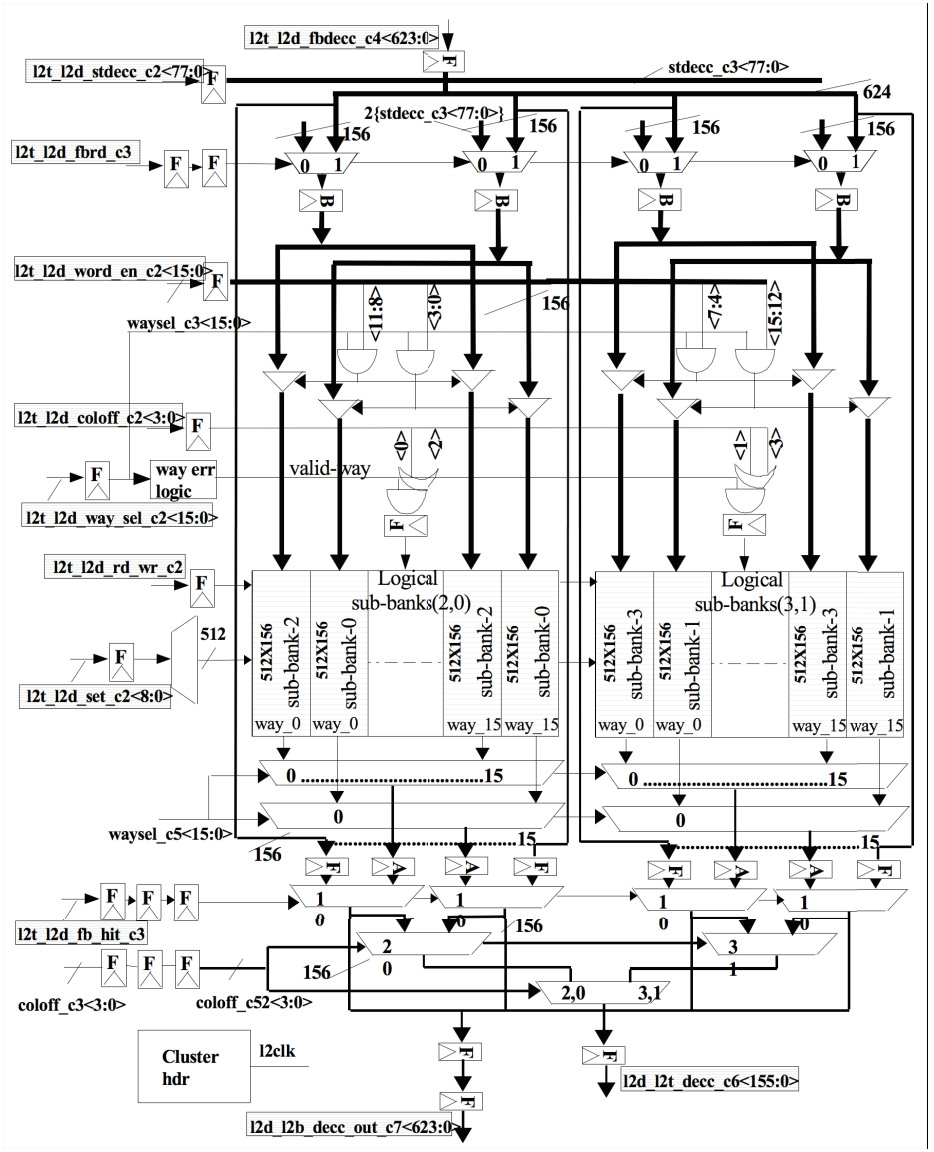


FIGURE 13-2 Functional Block Diagram of 1 Bank of L2 Data Array



13.3 I/O List

TABLE 13-1 L2 Cache Data Array I/O List

Signal name	Width	I/O	Source / Destination	Description
l2t_l2d_way_sel_c2	[15:0]	I	l2t	Way select inputs, decoded
l2t_l2d_col_offset_c2	[3:0]	I	l2t	Select 16 byte quarter line, decoded
l2t_l2d_fb_hit_c3		I	l2t	MUX select for fb_data or l2data out, 1 = fb_data
l2t_l2d_fbrd_c3		I	l2t	MUX select for fb_data or stdata write, 1 = fb_data
l2t_l2d_rd_wr_c2		I	l2t	read or write operation select, low during write
l2t_l2d_set_c2	[8:0]	I	l2t	set index inputs
l2t_l2d_word_en_c2	[15:0]	I	l2t	Enables for word writes, decoded, qualified externally by write so they're stable during read
l2t_l2d_stdecc_c2	[77:0]	I	l2t	data/ecc for store operations (writes)
l2b_l2d_fbdecc_c4	[623:0]	I	l2b	Fb_data/ecc to be MUXed with L2 output data for reads, and with store input data for writes
l2b_l2d_en_fill_clk_v0		I	l2b	Clock enable (fill) to save power
l2b_l2d_en_fill_clk_v1		I	l2b	Clock enable (fill) to save power
l2t_l2d_en_fill_clk_ov		I	l2t	overide for fill power save
l2t_l2d_pwrsav_ov		I	l2t	Overide for c7-flop power
rst_wmr_protect		I	ccu	warm reset (for cluster header)
rst_wmr_		I	ccu	warm reset (for cluster header)
rst_por_		I	ccu	power on reset / asynchronous reset (for cluster header)
gclk		I	ccu	global clock
tcu_aclk		I	tcu	scan a-clock
tcu_bclk		I	tcu	scan b-clock
tcu_scan_en		I	tcu	scan enable
tcu_pce_ov		I	tcu	pce override
tcu_ce		I	tcu	block enable
tcu_clk_stop		I	tcu	clock stop (for cluster header)
tcu_atpg_mode		I	tcu	Control signal to cluster header

TABLE 13-1 L2 Cache Data Array I/O List *(Continued)*

tcu_se_scancollar_in		I	tcu	scan enable for input flops
tcu_se_scancollar_out		I	tcu	scan enable for output flops
tcu_array_wr_inhibit		I	tcu	To protect array contens during scan shift (disables bank)
scan_in		I		scan chain data input
l2b_l2d_fuse_l2d_data_in	[9:0]	I	l2b	Address for bad row / column
l2b_l2d_fuse_rid	[6:0]	I	l2b	Select one of the 96 redundancy registers set in the 512 MB block
l2b_l2d_fuse_reset		I	l2b	Reset the fuse register contents
l2b_l2d_fuse_l2d_wren		I	l2b	When asserted, the fuse_data-in will be loaded into the redundant register selected by rid address
l2d_l2b_decc_out_c7	[623:0]	O	l2b	L2 output data/ecc for evictions (reads)
l2d_l2t_decc_c6	[155:0]	O	l2t	L2 output data/ecc for loads (reads)
l2d_l2b_efc_fuse_data	[9:0]	O	l2b	Fuse data output (contents of redundancy register selected by rid)
scan_out		O		l2d scan out

TABLE 13-2 L2 Cache Data Array Functional Table

way_sel	col_offset	rd_wr	fbrd	fb_hit	word_en	Operation/function	Description
All 0's	X	X	X	0	X	No Op	Nothing happens, no way selected
X	All 0's	X	X	0	X	No Op	Nothing happens, no bank selected
1 of 16 = 1	1 of 4 = 1	1	X	0	X	Load	Read from L2 data to l2t (156b)
1 of 16 = 1	All 1's	1	X	0	X	Evict	Read from L2 data to wbb or rdma (624b)
All 0's	X	1	X	1	X	Cache Bypass "Fill Op"	Fill buffer data sent instead of wbb data (624b)
All 0's	1 of 4 = 1	1	X	1	X	Load data Bypass	Fill buffer data sent to l2t (156b)

TABLE 13-2 L2 Cache Data Array Functional Table *(Continued)*

way_sel	col_offset	rd_wr	fbrd	fb_hit	word_en	Operation/function	Description
1 of 16 = 1	1 of 4 = 1	0	0	X	(1 or 2) of16 =1	Store	Write into L2 from l2t (#b from word_en's)
1 of 16 = 1	4 of 4 = 1	0	1	X	All 1's	Fill	Write into L2 from fbb
>1 of 16 = 1						ILLEGAL	Way_sel's must be mutually exclusive.

13.4 Timing Diagrams

FIGURE 13-3 L2data Load Operation

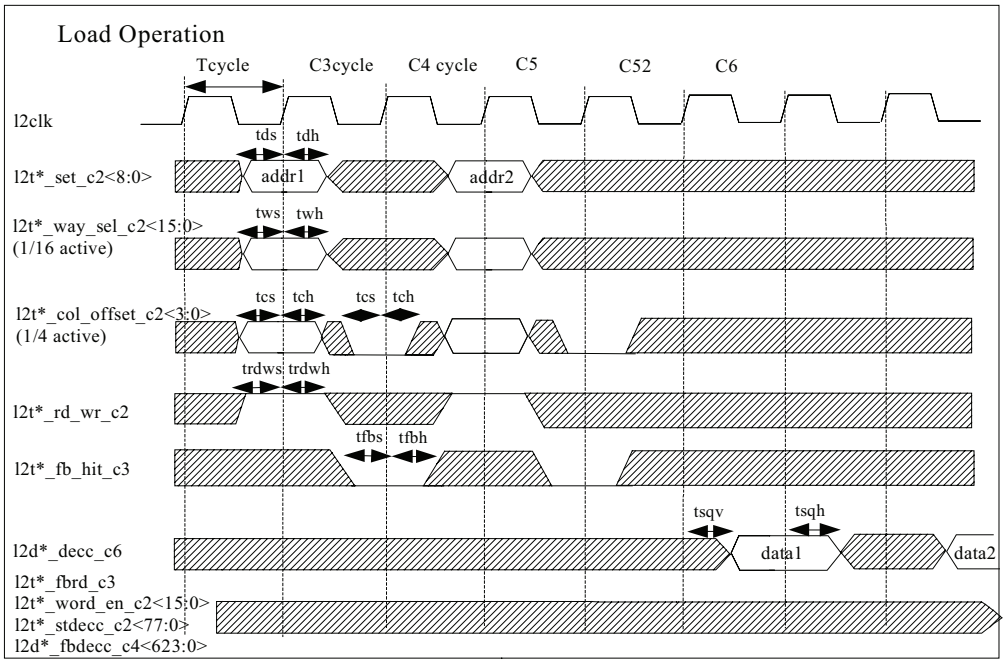


FIGURE 13-4 L2data Evict Operation

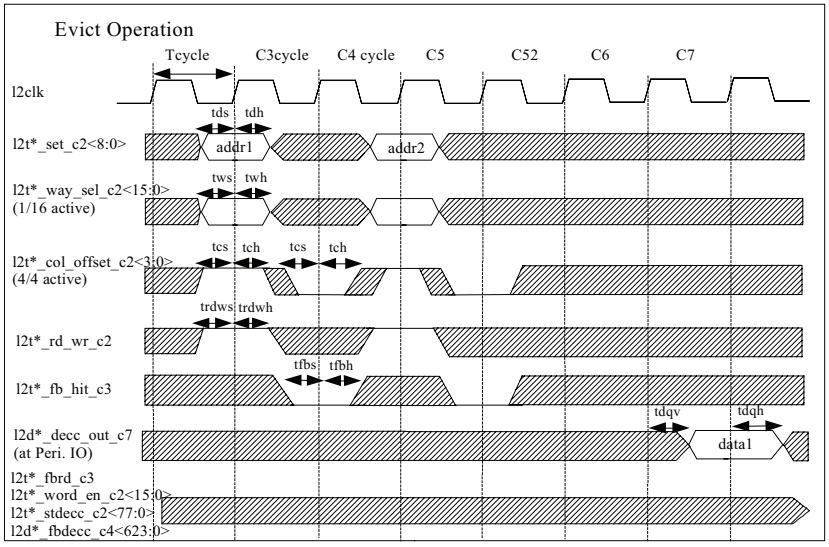


FIGURE 13-5 L2data Store Operation

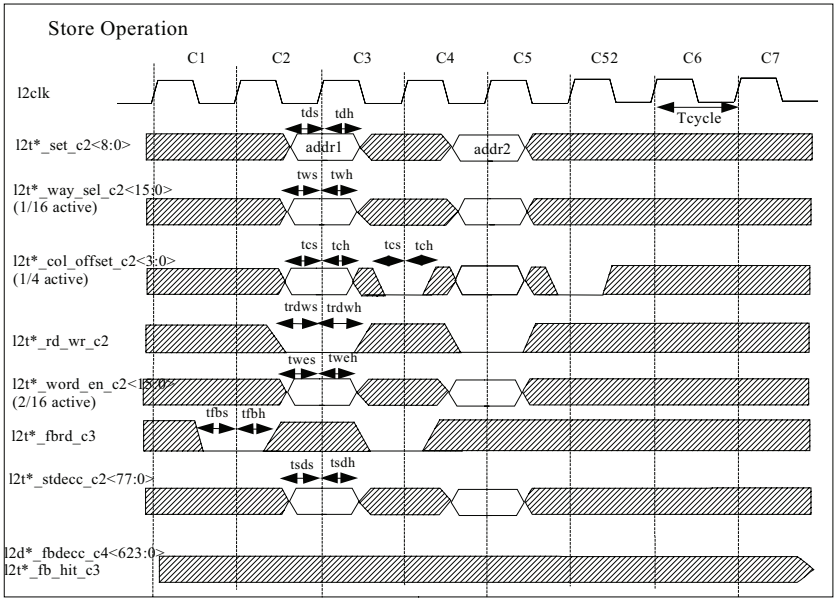


FIGURE 13-6 L2data Fill Operation

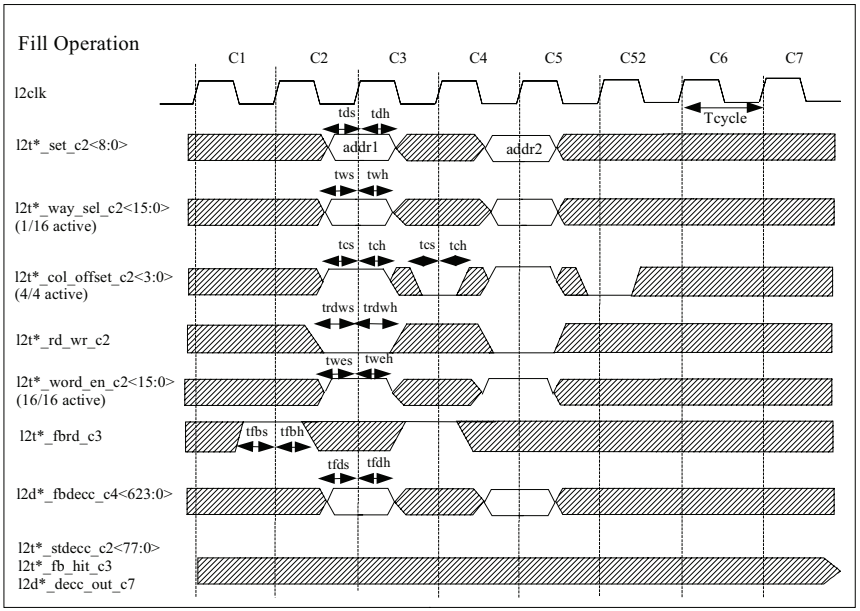


FIGURE 13-7 L2data Load Data Bypass Operation

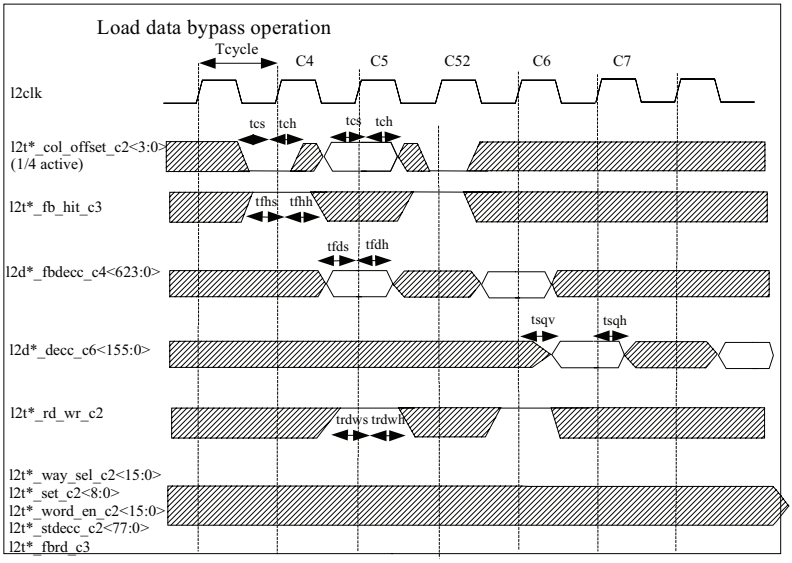
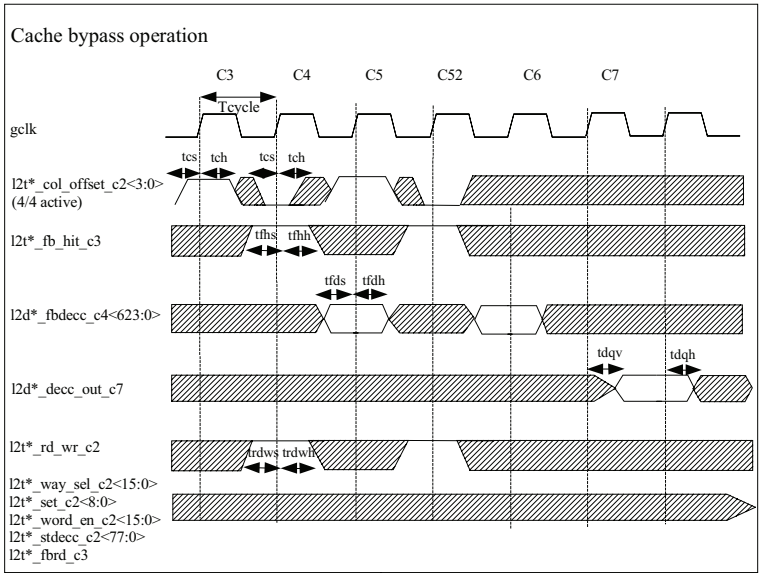


FIGURE 13-8 L2data Cache Bypass Operation


Cache bypass operation
C3
C4
C5
C52
C6
C7
Tcycle
gclk
tcs
tch
tcs
tch
l2t*_col_offset_c2<3:0>
(4/4 active)
l2t*_fb_hit_c3
tfhs
tfhh
tfds
tfdh
l2d*_fbdecc_c4<623:0>
tdqv
tdqh
l2d*_decc_out_c7
l2t*_rd_wr_c2
trdws
trdwh
l2t*_way_sel_c2<15:0>
l2t*_set_c2<8:0>
l2t*_word_en_c2<15:0>
l2t*_stdecc_c2<77:0>
l2t*_fbrd_c3

CHAPTER 14

L2-Cache Miss Buffer Data Array

This chapter describes the following topics:

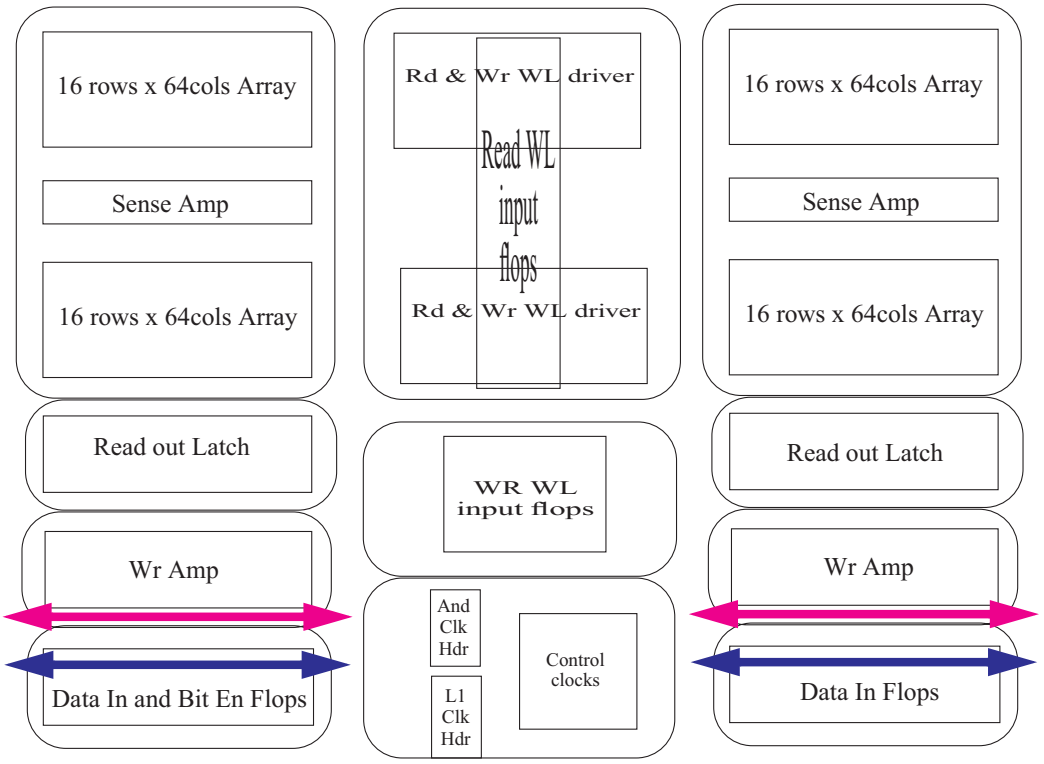
14.1 Functional Description

The l2t_dp_32x128_cust Register File is used as MissBuffer data array in L2 Cache. This is a 1R1W register file array with input addresses as one-hot decoded and flopped.

- The l2t_dp_32x128_cust is physically structured as 4 sub-arrays of 1024 bits each (16x64 array).
- Designed to meet the speed target of 1.4 GHz (714ps cycle time or 357ps phase time)
- All inputs are flopped. Read enable is captured with a gated latch cl_mc1_sram_msff_mo_8x from mc1_l library.
- Write address and Write enable are captured with a standard cell library MSFF.
- Read operation occurs during the A-phase of l2clk. Read data is captured by a glitch latch. Single ended full swing read bitline is used.
- Write operation occurs during the B-phase of l2clk. Differential full swing write bitlines.

14.2 Block Diagrams

FIGURE 14-1 Miss Buffer Data Array Block Diagram



14.3 I/O List

TABLE 14-1 Miss Buffer Data Array Functional Input / Output Signal List

Signal Name	Width	I/O	Description
l2clk		Input	input clock
rd_wl	[31:0]	Input	read address (Flopped)
read_en		Input	read enable (Flopped - NAND gate)
wr_wl	[31:0]	Input	write address (Flopped)
din	[127:0]	Input	write data (Flopped)
wr_en		Input	write enable (Flopped)
dout	[127:0]	Output	read data out (Latched)

TABLE 14-2 Miss Buffer Data Array Test Related Signals

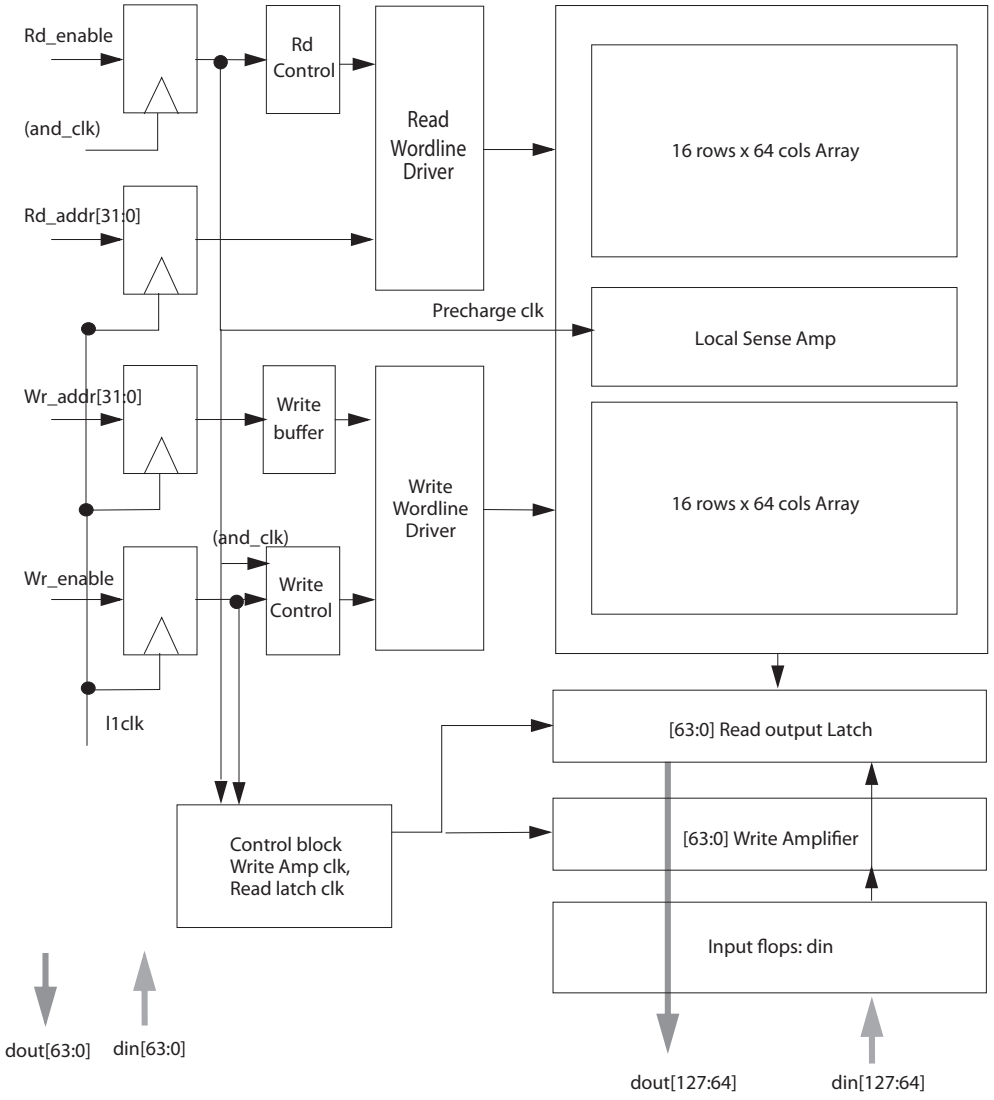
Signal Name	I/O	Description
tcu_se_scancollar_in	Input	scan enable for l1clk hdr
tcu_scan_en	Input	scan_enable for and_clk hdr
scan_in	Input	scan input data
tcu_pce_ov	Input	test control signal for l1clk hdr
pce	Input	test control signal for l1clk hdr
tcu_aclk	Input	scan input clock
tcu_bclk	Input	scan output clock
scan_out	Output	scan output
tcu_array_wr_inhibit	Input	Read/Write Inhibit

14.4 Functional Operations

TABLE 14-3 Miss Buffer Data Array Operations

Inputs				Outputs		Comments
read_en	write_en	tcu_array_wr_inhibit	write_data [127:0]	read_data [127:0]		
				RTL	Schematic	
0	0	0	X	Qn-1	Qn-1	NOP (data_out = previous read data)
1	0	0	X	array contents for read_addr[31:0]	array contents for read_addr[31:0]	Data read out from read_addr[31:0]
0	1	0	Valid	Qn-1	Qn-1	write_addr[31:0] location is updated with write_data[127:0]
X	X	1	Valid	Qn-1	Qn-1	Write & Read is disabled with tcu_array_wr_inhibit
1	1	0	Valid	array contents for read_addr[31:0]	array contents for read_addr[31:0]	Valid if read_addr[31:0] and write_addr[31:0] are different. (0-In monitor. see section 3.10.2)

FIGURE 14-2 Miss Buffer Data Array Logical Diagram (right half of block)



14.4.1 Pipeline Diagrams

MBDATA is a single array array. All Inputs are flopped.

14.5 Timing Diagrams

FIGURE 14-3 Miss Buffer Data Array Read Timing Diagram

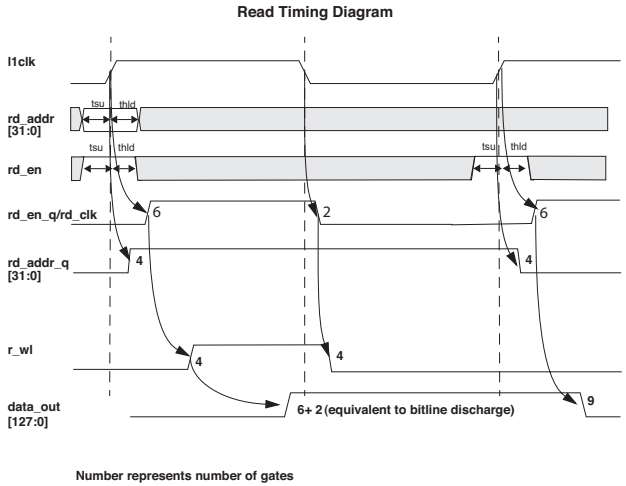
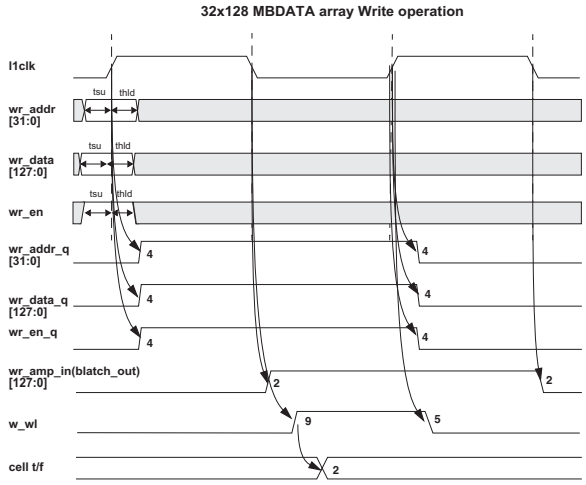


FIGURE 14-4 Miss Buffer Data Array Write Timing Diagram



CHAPTER 15

L2-Cache Miss Buffer Request CAM

This chapter describes the following topics:

15.1 Functional Description

The Miss Buffer contains miss requests as well as multi-pass L2 operations. The buffer contains data and address (tag) entries. The tag array is an 32 entry CAM of 40 bits each. A false hit on a tag can result in data corruption. A false miss can also result in data corruption. OpenSPARC T1 does not protect the tags or data. OpenSPARC T2 also does not protect the data or tags due to their small contribution to the FIT rate.

- Number of entries.

32

- Number of bits per entry.

42

- Column folding (read or write column muxing).

None.

- How many read/write/cam ports.

One read, one write, and one cam.

- Operations which are allowed or not allowed in same cycle.

Write and cam operations can not be done in the same cycle - the requirement for this has been updated to allow write and cam operations to occur in the same cycle, but the match output for the written row is undefined.

Read and write operations to the same address are not allowed.

- Operations which are allowed or not allowed to occur in consecutive cycles.

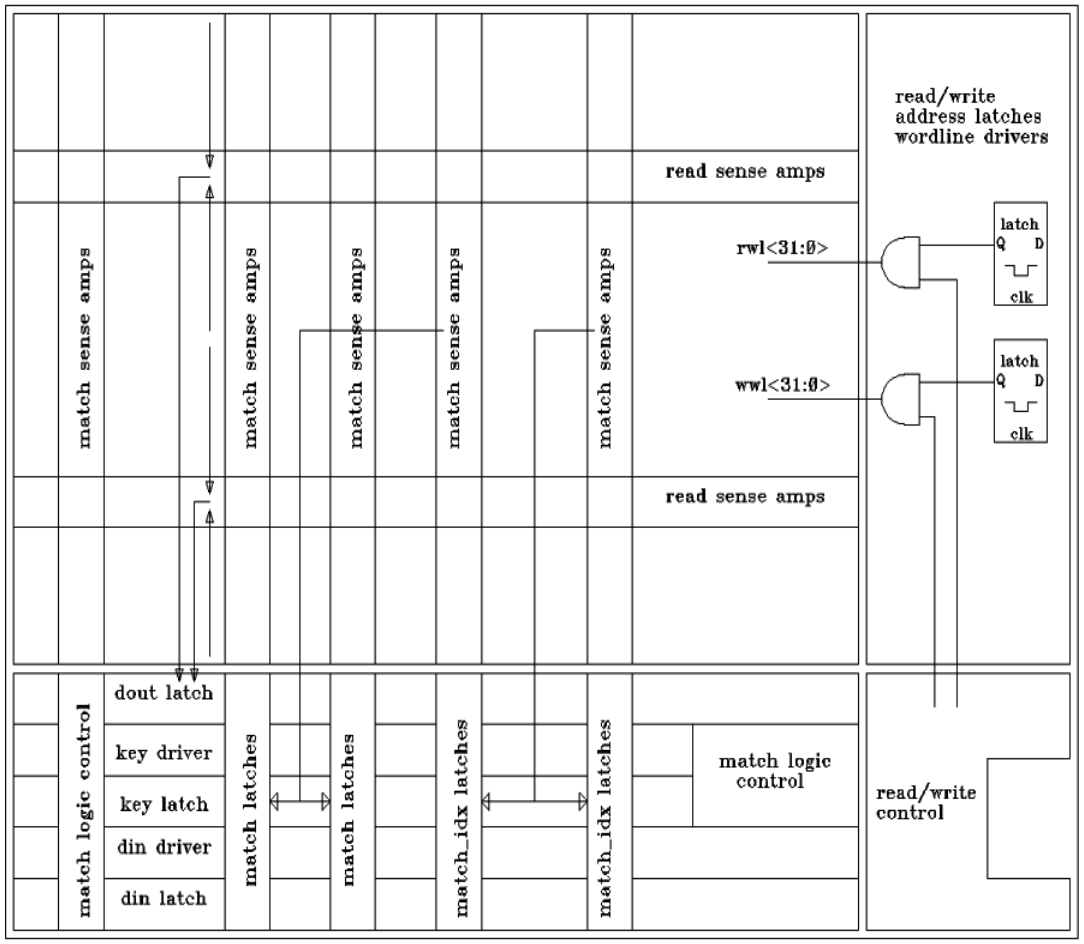
No restrictions.

- For each operation describe the following:
  - How many cycles does operation take.
  - What phase of cycle does operation occur.
  - How many bits are operated upon.

Operation	cycles	phase	bits
Read	1	A	41:0
Write	1	A	41:0
Match (cam)	1	A	41:7

15.2 Block Diagram

FIGURE 15-1 Miss Bufferrequest CAM Block Diagram



15.3 Pipeline Diagram

TABLE 15-1 Miss Buffer Reauest CAM Pipeline Diagram

C1	C2	C3	C4	C5	C52	C6	C7	C8
tag, VUAD read	way sel logic	way sel xmit in l2d	data array read cyc1	data array read cyc2	data array read cyc3	data xmit cycle	request to the dest cpx queue	Mux Data/In val. Vector
VUAD bypass	xmit way sel to l2d		FB data read cycle					
tag compare	rd/wr! Gen. xmit		Xmit inputs to directory	stage FB data	4:1 mux	gen inval. vector	check ECC on data	data return to dest. cpx
Check ECC for Tags	vaud ecc check			D$ directory write	mux with FB data			
MB cam and MB hit logic				I$ directory CAM				
FB cam				VUAD write				
WBB cam								

15.4 I/O List

TABLE 15-2 Miss Buffer Request CAM I/O List

Signal Name	Width	I/O	Source/Dest	Description
adr_w	[31:0]	IN/flopped	l2t_misbuf_ctl	Write Address
adr_r	[31:0]	IN/flopped	l2t_misbuf_ctl	Read Address
din	[41:0]	IN/flopped	l2t_evctag_dp	write data
key	[41:7]	IN/flopped	l2t_arbadr_dp	cam data for matching
read_en		IN/flopped	l2t_misbuf_ctl	Read Enable
write_en		IN/flopped	l2t_misbuf_ctl	Write Enable

TABLE 15-2 Miss Buffer Request CAM I/O List *(Continued)*

Signal Name	Width	I/O	Source/Dest	Description
lookup_en		IN/flopped	l2t_arb_ctl	cam operation enable
l2clk		IN/No	grid	main clock
match	[31:0]	Out/No	l2t_misbuf_ctl	cam result for key [41:7]
match_idx	[31:0]	Out/No	l2t_misbuf_ctl	cam result for key [17:9]
dout	[41:0]	Out/No	l2t_evctag_dp	read data out
pce		IN/No	tied high	clock enable
tcu_array_wr_inhibit		IN/No	tcu	inhibits operations in the array
tcu_se_scancoller_in		IN/No	tcu	controls input flops
tcu_pce_ov		IN/No	tcu	overides the clock enable
tcu_aclk		IN/No	tcu	scan latch clock
tcu_bclk		IN/No	tcu	scan latch clock
tcu_clk_stop		IN/No	tcu	clock header control
tcu_array_bypass		IN/No	tcu	bypass mode control
scan_out		IN/No	l2t_filbuf_ctl	output of scan chain
scan_in		IN/flpped	l2t_dp_32x128_cust	input of scan chain
tcu_scan_en		IN/No	tcu	control internal strobes

15.5 Functional Operations

TABLE 15-3 Functional Operations

enables			operation	results		
read_en	write_en	lookup_en	operation	dout	match	match_idx
L	L	L	no operation	previous data	32′b0	32′b0
L	L	H	lookup	previous data	match results	match results
L	H	L	write	previous data	32′b0	32′b0
L	H	H*	write/lookup	read data	match results	match results

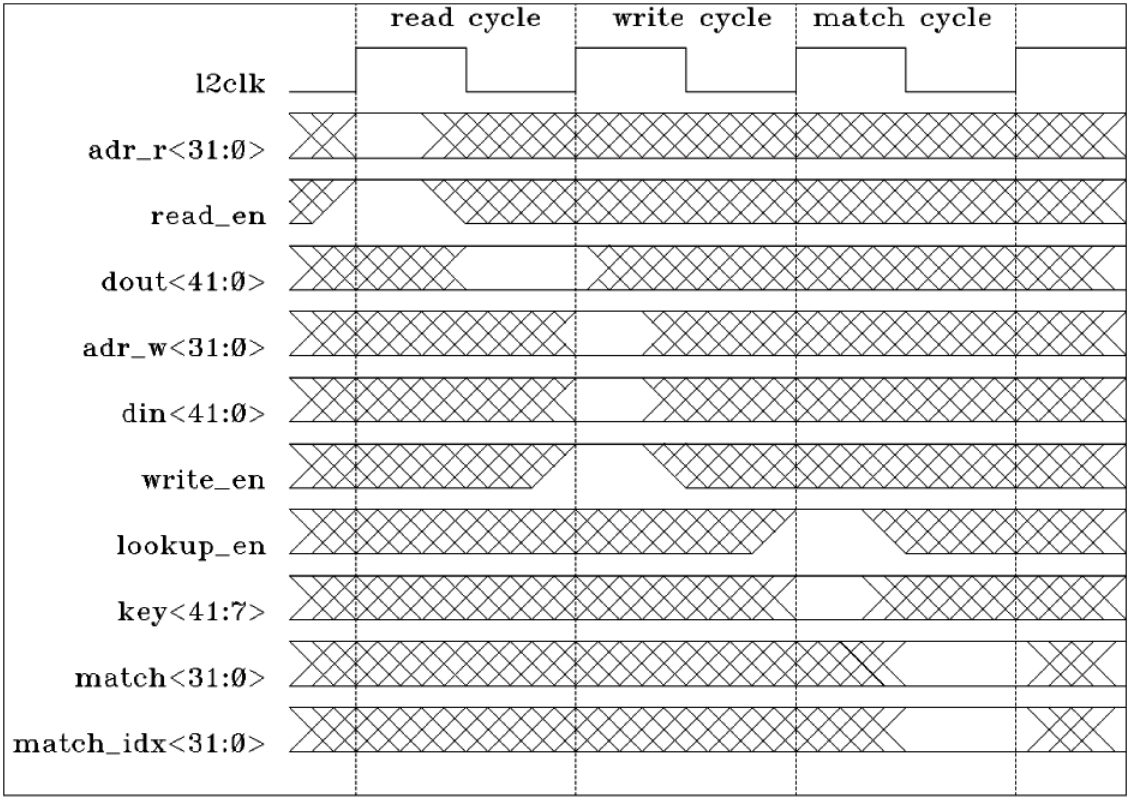
TABLE 15-3 Functional Operations *(Continued)*

enables			operation	results		
read_en	**write_en**	**lookup_en**	**operation**	**dout**	**match**	**match_idx**
H	L	L	read	previous data	32′b0	32′b0
H	L	H	read/lookup	**previous data**	match results	match results
H	H	L	read/write	read data- indeterminate for same address	32′b0	32′b0
H	H	H*	read/write/ lookup	read data- indeterminate for same address	match results	match results

* The X is the logicaal output. The circuit protects itself by producing a 0 on the match outputs. Since the timing for this is slower, it remains defined as X.

15.6 Timing Diagram

FIGURE 15-2 Miss Buffer Request CAM Timing Diagram



CHAPTER 16

L2-Cache Instruction and Data Directory CAM

This chapter describes the following topics:

16.1 Functional Description

16.1.1 Instruction Cache Directory

L1 Icache for each core has 64 sets, each set has 8 ways. Since L1 Icache line size is 32B, this gives a total of (8x64x8x32) bytes = 128 KB or 4K L1 Icache lines in all cores together. Each L1 Icache is 16KB.

Thus each L1 cache will map (64/8)= 8 sets to each L2 bank. So for each L2 bank, the Icache directory will consist of (8x8) sets of L1 Icache lines for all cores combined. This gives a total of 512 L1 Icache lines per bank.

These 512 L1 Icache line mappings gets organized physically in the ICache directory as follows:

Each Icachedirectory has 16 panels arranged as 4 rows and 4 columns. Row gets accessed by {address[5], I$ replacement way [2]}, column by address [10,9]. Each panel has 32 entries indexed by {cpu_id(3 bits), I$ replacement way [1:0].

For an update related to a Ifetch hit, the panel is accessed by {address {10,9,5}, I$ replacement way [2]}, and the entry within the panel to be updated is sselected by {cpu_id (3 bits) column by address [10,9]. Each panel has 32 entries indexed by {cpu_id(3 bits), I$ replacement way [1:0]}.

For an update related to a Ifetch hit, the panel is accessed by {address {10,9,5}, I$ replacement way [2]} and the entry within the panel to be updated is selected by {cpu_id(3bits), I$ replacement way [1:0]} for the Ifetch hit.

For a store or a Dcache mutual-exclusivity check on a load, which can potentially invalidate a maximum 8 L1 Icache lines(1 line per core), 2 panels gets selected by adddress {10,9,5} and all 32 entries within each panel get CAMed against the sotre, based on which a invalidation vector gets generated for a max of 8 L1 Icache line invalidation's (one per core). The way number for each L1 Icache line will be encoded as a 3 bit field in the inval vector.

For an eviction, since the L2 cache line is 64 bytes, 4 panels out of 16 will get CAMed based on address [10,9] (i.e. 1 column). This would mean a tottal of 64x2=128 compares to invalidate a max of 2 Icache lines per L1, i.e. a max of 2x8=16 L1 Icache lines for all cores combined. This will come out as 16 I$ L1 lines eviction vector from L2. The way number for each L1 Icache line will be encoded as a 3 bit fiels in the inval vector.

Each entry in the directory will store {L2 index [9 bits], L2 way [4 bits], parity, valid} i.e. a total of 15 bits corresponding to the location in L2 that lthe L1 line maps to.

For a Ifetch hit, the entry gets updated with {L2 index [9 bits], L2 way [4 bits], parity}, while on a store or eviction or a Dcache mutual-exclusivity check on a load, the {L2 index [9 bits], L2 way [4 bits]} gets CAMed against the stored value of each entry.

16.1.2 Data Cache Directory

There are 8 cores in OpenSPARC T2. L1 Dcache foe each core has 128 sets, each set has 4 ways. Since L1 Dcache line size is 16 B, this gives a total of (8x128x4x16) bytees=64 KB of 4 KL1 lines in all cores together. Each L1 Dcache is 8 KB.

Thus each L1 Dcache will map (128/8) 16 sets to each L2 bank. So for each L2 bank, the Dcache directory will consist of (16x8) sets of L1 Dcache lines for all cores combined. This gives a total of 512 Dcache lines per bank.

These 512 L1 Dcache line mappings gets organized physically in the Dcache directory as follows:

Each Dcache directory has 16 panels organized as 4 rows and 4 columns. Row gets accessed by address[5:4]=, column by address [10:9]. Each panel has 32 entries indexed by {cpu_id (3 bits), replacement way (2 bits)}.

For an update related to a load, the panel is accessed by address {10,9,5,4} and the entry within the panel to be updated is selected by {cpu_id (3 bits), replacement way (2 bits)} for the load.

For a store or a Icache mutual-exclusivity check on a Ifetch, which can potentially invalidate a maximum 8 L1 cache lines (one L1 line per core), the panel gets selected by address {10,9,5,4} and all 332 entries within that panel get CAMed against the store, based on which a invalidation vector gets generated for a max of 8 L1 Dcache line invalidations (one per core). the way number for each L1Dcache will be encoded as a 2 bit field in the inval vector.

For an eviction, since the L2 cache line is 64 bytes, 4 panels out of 16 will get CAMed based on address [10,9] (i.e. 1 column). This would mean a total of 64x2=128 compares to invalidate a max of 4 cache lines per L1, i.e. a max of 4x8=32 L1 Dcache lines for all cores combined. This will come out as 32 D$ L1 lines eviction vector from L2. The way number for each L1 Dcache line will be encoded as a 2 bit field in the inval vector.

Each entry in the directory will store {L2 index [9 bits], L2 way [4 bits], parity, valid} i.e. a total of 15 bits corresponding to the location in L2 that the L1 line maps to.

For a load hit, the entry gets updated with {L2 index [9 bits], L2 way [4 bits], parity}, while on a store or eviction or a Icache mutual-exclusivity check on a Ifetch, the {L2 index [9 bits], L2 way [4 bits]} gets CAMed against the stored value of each entry.

- Number of entries.

64

- Number of bits per entry.

4x16

- Column folding (read or write column muxing).

None.

- How many read/write/cam ports.

One read/write port, one write/cam port. Addresses are shared between the read and write ports, while write data and cam data are also shared.

- Operations which are allowed or not allowed in same cycle.

Write and cam operations can not be done in the same cycle.

Read and write operations can not be done in the same cycle.

- Operations which are allowed or not allowed to occur in consecutive cycles.

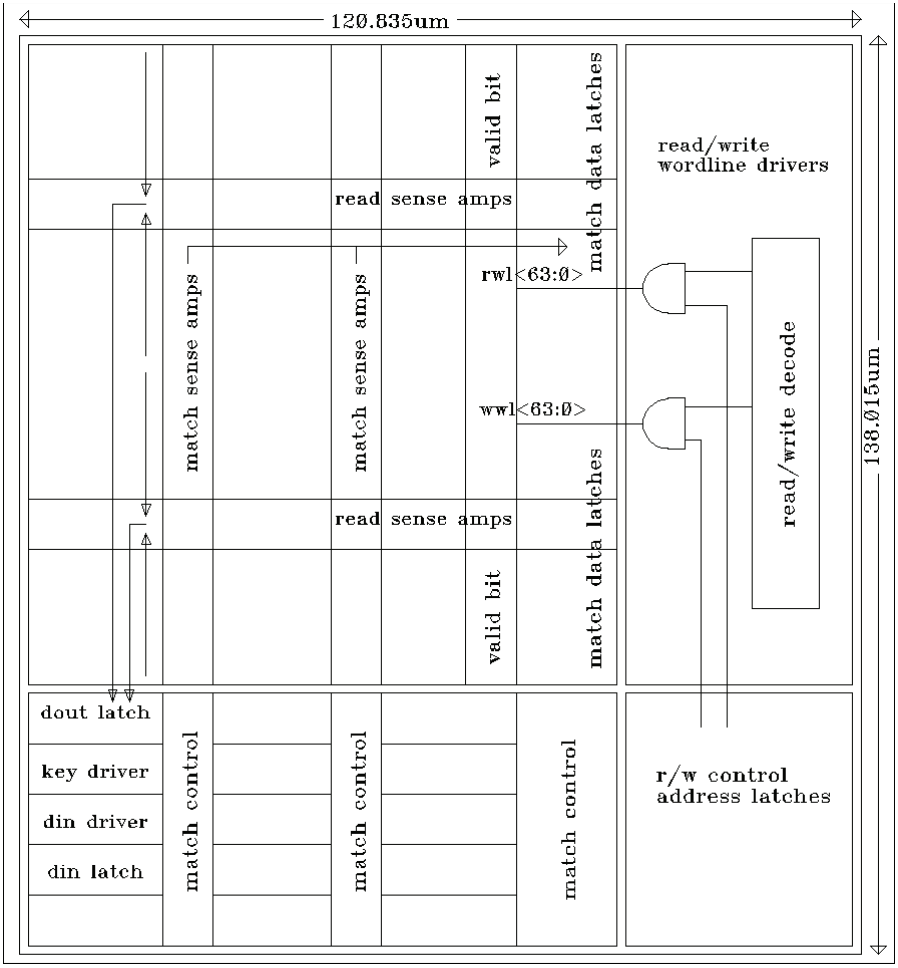
No restrictions.

- For each operation describe the following:
  - How many cycles does operation take.
  - What phase of cycle does operation occur.
  - How many bits are operated upon.

Operation	cycles	phase	bits
Read	1	A	4x<15:0>
Write	1	A	4x<15:0>
Match (cam)	1	A	4x<13:0>
Invalidate	1	B	last match data

16.2 Block Diagrams

FIGURE 16-1 Instruction Cache and Data Cache Block Diagram



16.3 Pipeline Diagram

TABLE 16-1 Instruction Cache and Data Cache Pipeline Diagram

C1	C2	C3	C4	C5	C52	C6	C7	C8
tag, VUAD read	way sel logic	way sel xmit in l2d	data array write cyc1	data array write cyc2	data array wr cyc3	gen inval. vector	request to the dest cpx queue (ack for write)	Mux Data/In val. Vector
VUAD bypass	xmit way sel to l2d							
tag compare	rd/wr! Gen. xmit		Xmit inputs to directory	I$ and D$ Directory CAM			check ECC on data	
Check ECC for Tags	vaud ecc check							
MB cam and MB hit logic				VUAD write				
perform store data ECC								
FB cam								
WBB cam								

16.4 I/O List

TABLE 16-2 Instruction Cache and Data Cache Directory I/O List

Signal Name	Width	I/O	Source/Dest	Description
rw_addr0 rw_addr1 rw_addr2 rw_addr3	[5:0]	IN/flopped	l2t_dirtop_ctl	read/write address
wr_data0 wr_data1 wr_data2 wr_data3	[15:0]	IN/flopped	l2t_dirtop_ctl	write data cam data
read_en	[3:0]	IN/flopped	l2t_dirout_dp	Read Enable
write_en	[3:0]	IN/flopped	l2t_dirout_ctl	Write Enable
cam_en	[3:0]	IN/flopped	l2t_dirtop_ctl	cam operation enable
l2clk		IN/No	grid	main clock
row_hit	[63:0]	Out/No	l2t_dirvec_dp	cam results
rd_data0 rd_data1 rd_data2 rd_data3	[15:0]	Out/No	l2t_dirout_dp	read output data
inv_mask0 inv_mask1 inv_mask2 inv_mask3	[7:0]	IN/flopped	l2t_dirctl_top	invalidate mask clears valid bit of last matched
force_hit [3:0]		IN/flopped	l2t_dirctl_top	force hit on wr_data*[13]
rst_warm_0		IN/flopped	l2t_dirctl_top	clear valid bits for 0,1
rst_warm_1		IN/flopped	l2t_dirctl_top	clear valid bits for 2,3
pce		IN/No	tied high	clock enable
tcu_array_wr_inhibit		IN/No	tcu	inhibits operations in the array
tcu_se_scancoller_in		IN/No	tcu	controls input flops
tcu_pce_ov		IN/No	tcu	overides the clock enable
tcu_aclk		IN/No	tcu	scan latch clock
tcu_bclk		IN/No	tcu	scan latch clock
tcu_clk_stop		IN/No	tcu	clock header control

TABLE 16-2 Instruction Cache and Data Cache Directory I/O List *(Continued)*

Signal Name	Width	I/O	Source/Dest	Description
tcu_array_bypass		IN/No	tcu	bypass mode control
scan_out		IN/No		output of scan chain
scan_in		IN/flopped		input of scan chain
tcu_scan_en		IN/No	tcu	control internal strobes

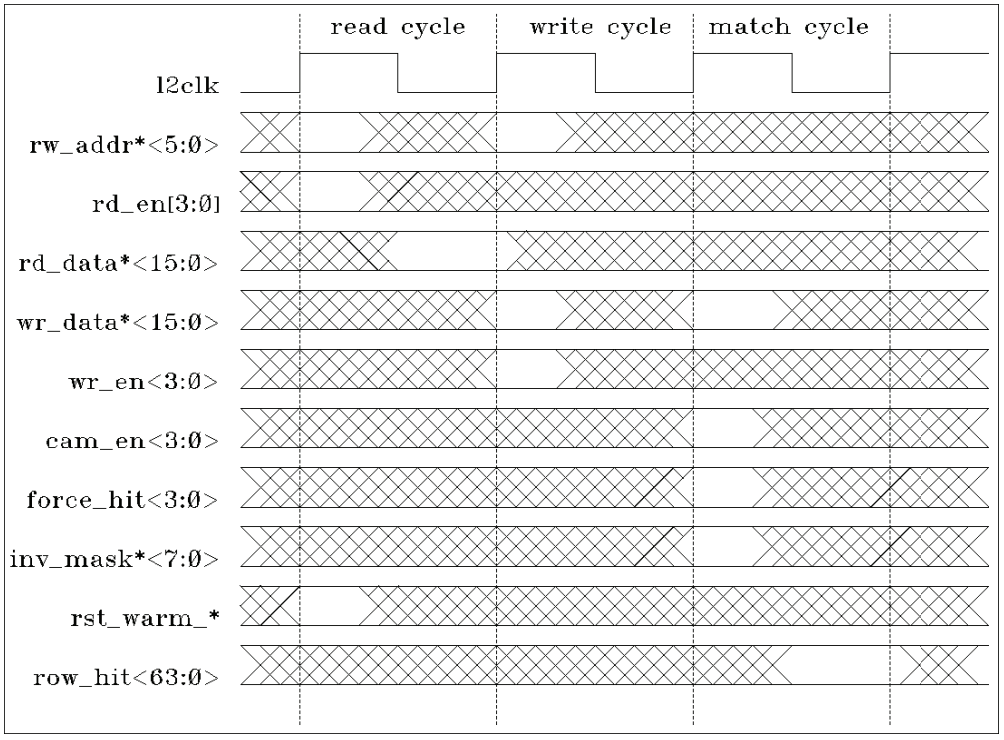
16.5 Functional Operations

TABLE 16-3 Instruction Cache and Data Cache Directory Functional Operations

enables			operation	results		
rd_en	wr_en	cam_en	operation	dout	match	
L	L	L	no operation	previous data	64'b0	
L	L	H	lookup	previous data	match results	
L	H	L	write	previous data	64'b0	
L	H	H	illegal	previous data	match results, x for written row	
H	L	L	read	previous data	64'b0	
H	L	H	read/lookup	previous data	match results	
H	H	L	illegal	indeterminate	64'b0	
H	H	H	illegal	indeterminate	match results, x for written row	

16.6 Timing Diagram

FIGURE 16-2 I/O Timing Diagram



CHAPTER 17

L2-Cache Fill Buffer

This chapter describes the following topics:

17.1 Functional Description

The Fill Buffer contains memory read data. This data is either cacheable reads to be written to the L2, or non-allocating cacheable data forwarded to the I/O interface. The buffer contains data and address (tag) entries. The data is protected by SEC/DED ECC and ECC is checked on the way from Fill Buffer to L2 pipe. The tag array is an 8 entry CAM of 40 bits each. A false bit on a tag can result in data corruption. A false miss can result in multiple fills for the same line outstanding, reducing performance. OpenSPARC T1 does not protect the tags or data. OpenSPARC T2 also does not protect the data or tags due to their small contribution to the FIT rate.

A correctable data ECC error is logged to a global ESR and, if enabled, generates a disrupting trap request to the core identified in the ERRORSTEER field of the L2 Control Register. Hardware corrects the error before writing into the L2.

An uncorrectable data ECC error is logged to a global ESR, and, if enabled, generates a disrupting trap request to the core identified in the ERRORSTEER field of the L2 Control Register.

17.1.1 Writeback Buffer

The Writeback Buffer contains modified evicted L2 data to be written back to memory. The data portion is protected by SEC/DED ECC and ECC is checked on the way from WBB to MCU. The tag is implemented as an 8 entry CAM with 40 bits per entry. OpenSPARC T1 does not protect the tag. OpenSPARC T2 also does not protect the tag due to its small contribution to the FIT rate.

If a correctable ECC error occurs on the data, the error is logged, and, if enabled, a disrupting trap request is generated to the core identified by the ASI_CMP_ERROR_STEERING register. Hardware corrects the error before writing the data to memory.

If an uncorrectable ECC error occurs on the data, the error is logged, and, if enabled, a disrupting trap request is generated to the core identified by the ASI_CMP_ERROR_STEERING register.

- Number of entries.

8

- Number of bits per entry.

40

- Column folding (read or write column muxing).

None.

- How many read/write/cam ports.

One read, one write, and one cam.

- Operations which are allowed or not allowed in same cycle.

Read and write operations to the same address are not allowed.

- Operations which are allowed or not allowed to occur in consecutive cycles.

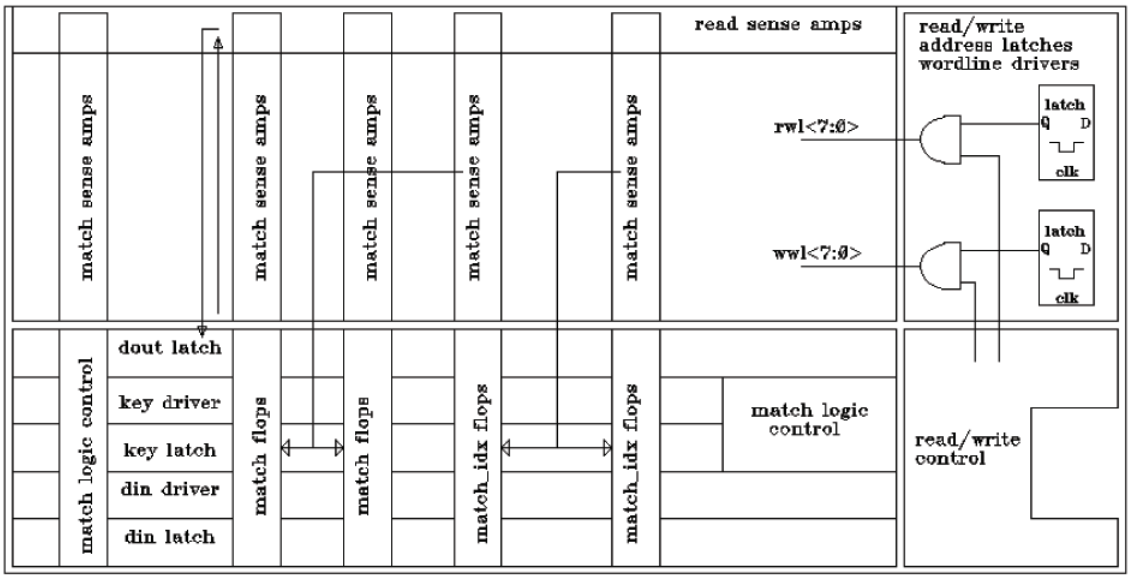
No restrictions.

- For each operation describe the following:
  - How many cycles does operation take.
  - What phase of cycle does operation occur.
  - How many bits are operated upon.

Operation	cycles	phase	bits
Read	1	A	39:0
Write	1	A	39:0
Match (cam)	1	A	39:7

17.2 Block Diagram

FIGURE 17-1 Fill Buffer Block Diagram



17.3 I/O List

TABLE 17-1 Fill Buffer I/O List

Signal Name	Width	I/O	Source/Dest	Description
adr_w	[7:0]	IN/flopped	l2t_filbuf_ctl, l2t_wbuf_ctl, l2t_rdmat_ctl	Write Address
adr_r	[7:0]	IN/flopped	l2t_filbuf_ctl, l2t_wbuf_ctl	Read Address
din	[39:0]	IN/flopped	l2t_evctag_dp, l2t_snpd_dp	write data

TABLE 17-1 Fill Buffer I/O List *(Continued)*

Signal Name	Width	I/O	Source/Dest	Description
key	[39:7]	IN/flopped	l2t_evctag_dp	cam data for matching
read_en		IN/flopped	l2t_filbuf_ctl, l2t_wbuf_ctl	Read Enable
write_en		IN/flopped	l2t_filbuf_ctl, l2t_wbuf_ctl, l2t_rdmat_ctl	Write Enable
lookup_en		IN/flopped	l2t_arb_ctl	cam operation enable
l2clk		IN/No	grid	main clock
match	[7:0]	Out/No	l2t_filbuf_ctl, l2t_wbuf_ctl, l2t_rdmat_ctl	cam result for key [41:7]
match_idx	[7:0]	Out/No	unused x 3	cam result for key [17:9]
dout	[39:0]	Out/No	l2t_evctag_dp	read data out
pce		IN/No	tied high	clock enable
tcu_array_wr_inhibit		IN/No	tcu	inhibits operations in the array
tcu_se_scancoller_in		IN/No	tcu	controls input flops
tcu_pce_ov		IN/No	tcu	overides the clock enable
tcu_aclk		IN/No	tcu	scan latch clock
tcu_bclk		IN/No	tcu	scan latch clock
tcu_clk_stop		IN/No	tcu	clock header control
tcu_array_bypass		IN/No	tcu	bypass mode control
scan_out		IN/No	l2t_arbdat_dp, l2t_ique_dp, l2t_rdmat_ctl	output of scan chain
scan_in		IN/flpped	l2t_filbuf_ctl, l2t_wbuf_ctl, l2t_evctag_dp	input of scan chain
tcu_scan_en		IN/No	tcu	control internal strobes

17.3.1 Pipeline Diagram

TABLE 17-2 Fill Buffer Pipeline Diagram

C1	C2	C3	C4	C5	C52	C6	C7	C8
tag, VUAD read	way sel logic	way sel xmit in l2d	data array read cyc1	data array read cyc2	data array read cyc3	data xmit cycle	request to the dest cpx queue	Mux Data/In val. Vector
VUAD bypass	xmit way sel to l2d		FB data read cycle					
tag compare	rd/wr! Gen. xmit		Xmit inputs to directory	stage FB data	4:1 mux	gen inval. vector	check ECC on data	data return to dest. cpx
Check ECC for Tags	vaud ecc check			D$ directory write	mux with FB data			
MB cam and MB hit logic				I$ directory CAM				
FB cam				VUAD write				
WBB cam								

17.4 Functional Operations

TABLE 17-3 Functional Operations

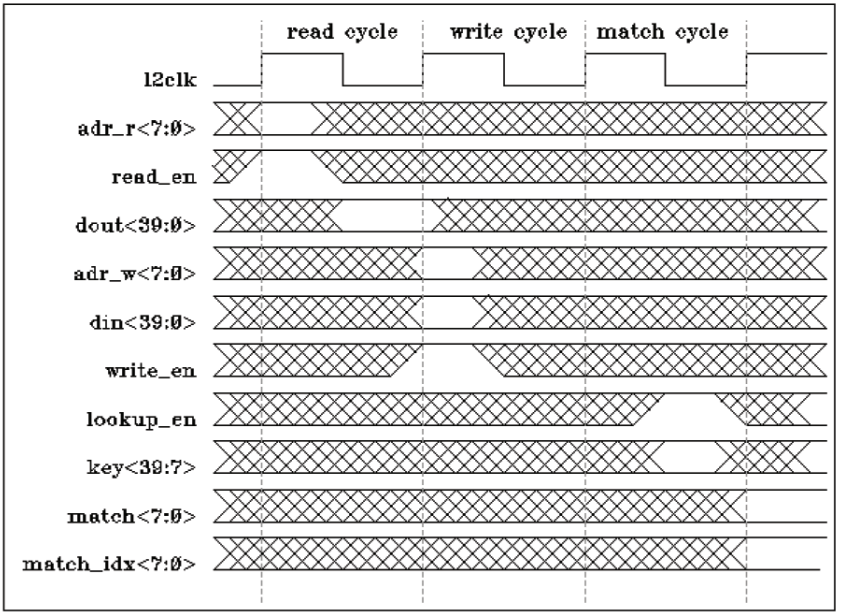
enables			operation	results		
read_en	write_en	lookup_en	operation	dout	match	match_idx
L	L	L	no operation	previous data	8'b0	8'b0
L	L	H	lookup	previous data	match results	match results
L	H	L	write	previous data	8'b0	8'b0
L	H	H	write/lookup	read data	match results	match results
H	L	L	read	previous data	8'b0	8'b0

TABLE 17-3 Functional Operations *(Continued)*

enables			operation	results		
read_en	write_en	lookup_en	operation	dout	match	match_idx
H	L	H	read/lookup	previous data	match results	match results
H	H	L	read/write	read data- indeterminate for same address	8'b0	8'b0
H	H	H	read/write/ lookup	read data- indeterminate for same address	match results	match results

17.5 Timing Diagram

FIGURE 17-2 Fill Buffer I/O Timing Diagram



CHAPTER 18

L2-Cache Dual Port Register File (16x160)

This chapter describes the following topics:

18.1 Functional Description

The l2t_dp_16x160_cust register file is a generic dual port register file which support byte and word write operations. TABLE 18-1 shows the various names and the number of instances the block is used. FIGURE 18-1 shows the L2 block diagram and interaction with other blocks and outside world.

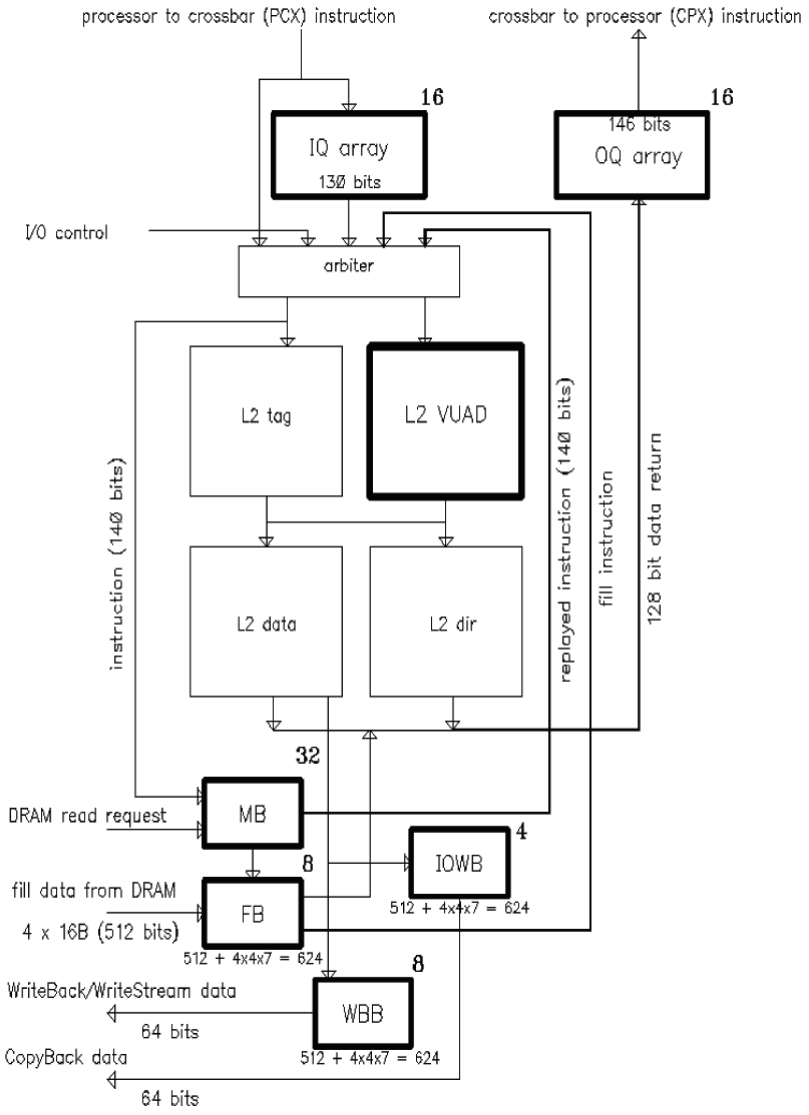
TABLE 18-1 Block Instances in L2

Input Queue Array	1
Output Queue Array	1
Fill Buffer Array	4
Write Back Array	4
IO Write Buffer Array	4
Total	14

- Input Queue (IQ) Array (iqarray) : It a 16 entry FIFO implemented as a dual port register file, which queues packets arriving on the processor to cache crossbar (PCX) when they cannot be immediately accepted into the L2 pipeline. The architectural requirement is 131 bits but implemented here as a 160 bit wide entry. The write port is for writing from the IQ to PCX and the read port is for reading the contents for issue into the L2 pipeline.
- Output Queue (OQ) Array (oqarray) : It is a 16 entry FIFO implemented as a dual port register file, which queues operations waiting for access to the cache to processor crossbar (CPX). The architectural requirement is 146 bits but implemented here as a 160 bit wide entry. The write port is for writing into the OQ from the L2 pipeline and the read port for reading the contents for issue to the CPX.
- Fill Buffer (FB) Array (fb_array_x) : It is a 16 entry buffer used for storing data coming from the DRAM. The architectural requirement is a 8 entry dual port array which can store 4 16B (128 bits) chunks of incoming DRAM data. After all 4 chunks arrive, the fill instruction enters the pipeline (L2 cache) and fills the cache. The FB has a corresponding CAM portion which contains the address for the data returned from the DRAM for filling the L2 cache. When the FB is full the Miss Buffer (MB) cannot make requests to the DRAM.
- Write Back Buffer (WBB) Array (wb_array_x) : It is a 16 entry buffer used for storing the dirty evicted data from the L2 cache. The architectural requirement is a 8 entry dual port array which can store the 4 16B chunks of evicted data which is opportunistically written to the DRAM. WBB has a corresponding CAM portion which contains the address of the evicted data stored in the WBB.
- I/O Write Buffer (IOWB) Array (iow_array_x) : It is a 16 entry buffer which stores incoming data from the PCI-EX interface in the case of a 64B write operation. Since the PCI-EX interface bus width is 32 bits (4B) wide, the data must be collected over 16 chunks (4 x 16B) before writing to the DRAM. The architectural requirement is a 4 entry dual port array which can store 4 16B chunks of PCI-EX data. It is the responsibility of the I/O interface to use a handshaking protocol to track the state of the IOWB. IOWB has a corresponding CAM portion which contains the address of the data stored in the IOWB (obtained from the I/O interface wating to be written to the DRAM).

18.2 Block Diagrams

FIGURE 18-1 Dual Port Register File (16x160) Block Diagram



18.3 I/O List

TABLE 18-2 Dual Port Register File (16x160) I/O List

Signal Name	Width	I/O	Description
rd_addr	[8:0]	IN/latched	Read Address
wr_addr	[8:0]	IN/flopped	Write Address
rd_en	1 bit	IN/latched	Read Enable
wr_en	1 bit	IN/flopped	Write Enable
din	[59:0]	IN/flopped	Data in
dout	[59:0]	OUT/latched	Data out
Clock and Test Related I/Os			
clk		IN/No	L2 clock input
scan_in		IN/No	Scan input
tcu_scan_en		IN/No	Test controller input for scan enable
tcu_se_scancoller_in		IN/No	Test controller input
tcu_pce_ov		IN/No	Test controller input
Pce		IN/No	Test controller input
tcu_adk		IN/No	Test controller scan-in clock
tcu_bdk		IN/No	Test controller scan-out clock
scan_out		OUT/No	Scan output

0in monitor in RTL ; if read address = write address in the same cycle output will be x.

When tcu_array_wr_inhibit goes high writes and reads are both disabled to the block.

18.4 Functional Operation

TABLE 18-3 Dual Port Register File (16x160) Modes of Operation

Enable values		
Read	**Write**	**Memory Operation**
1	1	RTL does not allow this to the same address
0	1	Write cycle at phase 2
1	0	Read cycle at phase 1
0	0	No Operation

18.5 Timing Diagrams

FIGURE 18-2 Dual Port Register File (16x160) I/O Timing Diagram

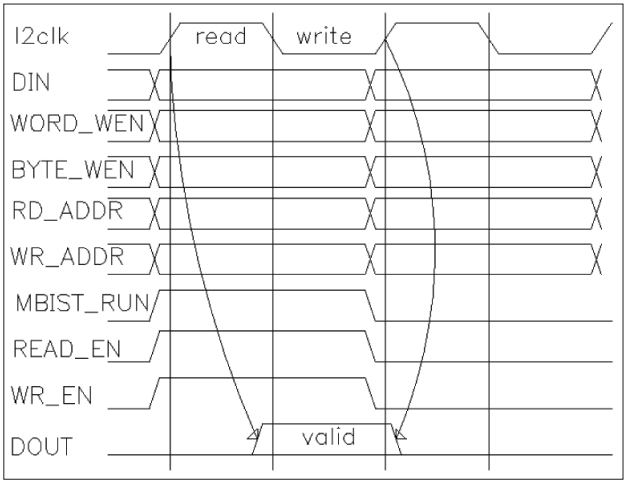


FIGURE 18-3 Dual Port Register File (16x160) Internal Timing, Read

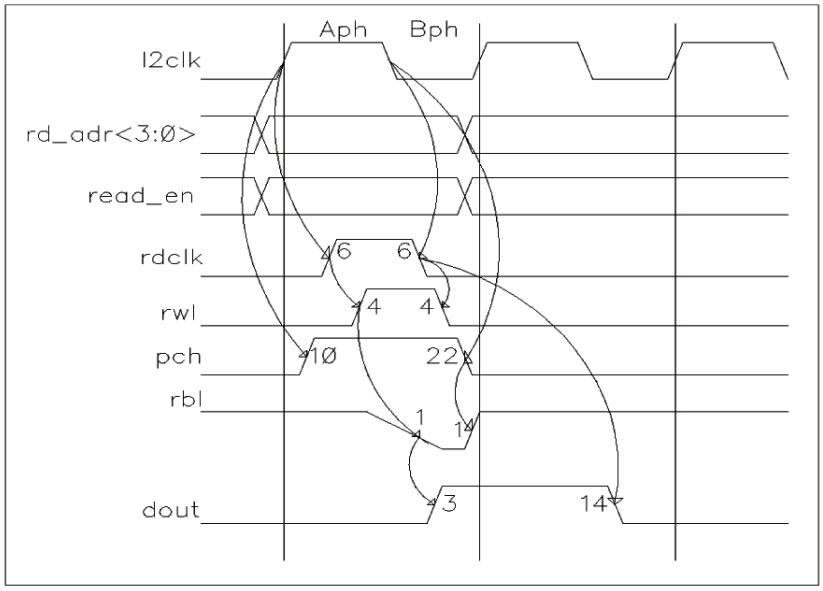
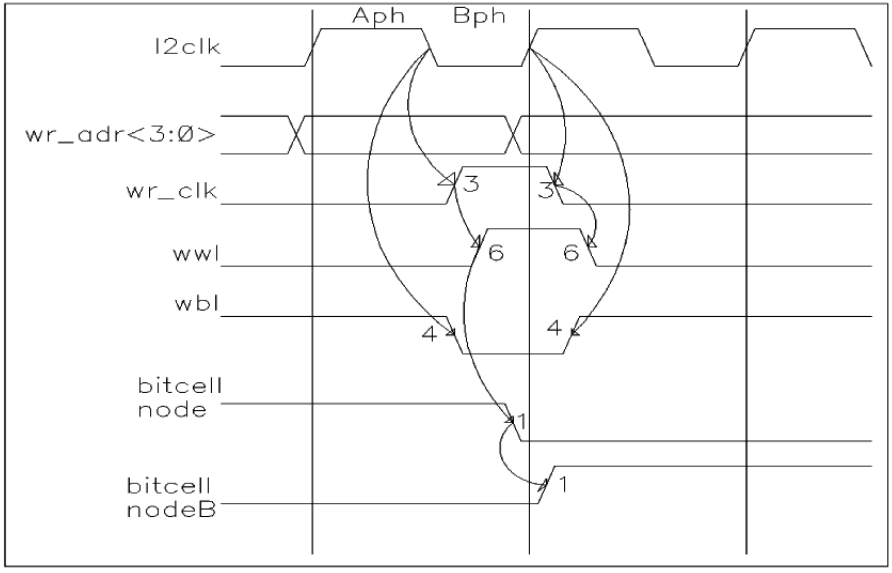


FIGURE 18-4 Dual Port Register File (16x160) Internal Timing, Write



CHAPTER 19

L2-Cache VUAD Register File (32x160)

This chapter describes the following topics:

19.1 Functional Description

This is 5Kb dual ported array used to maintain the state of every line in the L2 cache for each bank. The state of each line is maintained using the Valid (V), Used (U), Allocate (A) and Dirty (D) bits.

The Allocate bit indicates that a marked line has been allocated to a miss. It is set when a line gets picked for replacement.

The Used bit is a reference bit used in the replacement algorithm. It is set when any store/load hits (1 per way).

The Dirty bit gets set when a store modifies the line.

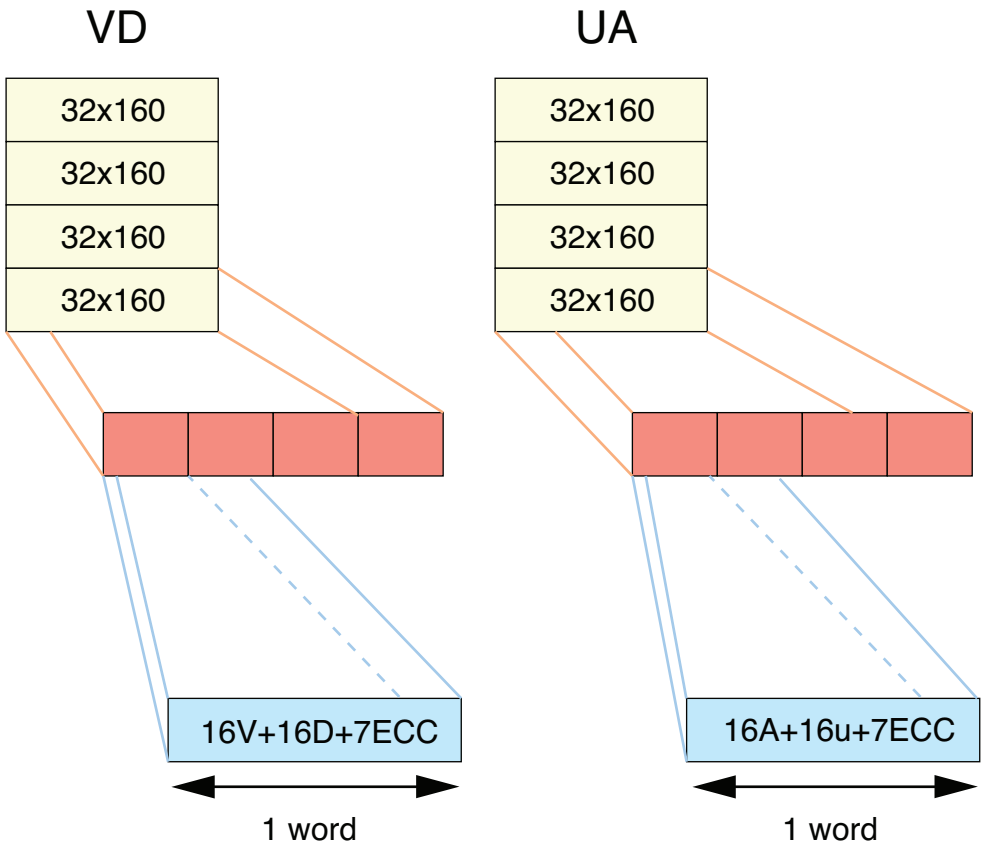
The Valid bit gets set when a new line is installed.

- The 32x160 RF is a dual ported array.
- The array is expected to operate at 1.4GHz frequency.

- Reads are done in phase A (1/2 cycle operation).
- Writes in phase B (1/2 cycle operation).
- Read access: reads from an entry specified by read_addr[4:0] and drives read_data[159:0] as output.
- Write access: Updates an entry specified by write_addr[4:0] with write_data[159:0].
- Supports word enables for writes (four words of 40 bits each).
- Supports read and write access in the same cycle, different entries or words.
- RTL team will ensure that when read and write addresses are the same, the word bits written are not needed in that cycle. Those output bits need to be X and 0in checkers/monitors may be in place.

19.2 Block Diagram

FIGURE 19-1 VUAD (32x160) Register File Block Diagram



19.3 Functional Operation

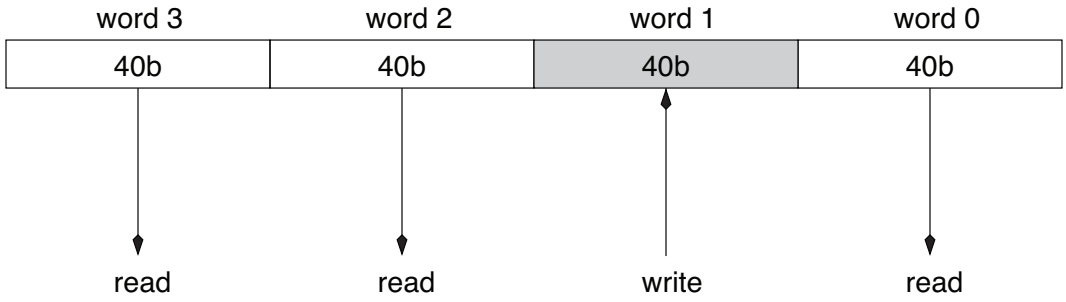
TABLE 19-1 Functinal Operation

read_en	word_en [3:0]	write_en	wr_inh	write_data [159:0]	read_data[159 :0]	Comments
0	X	0	0	X	160'h0	NOOP
1	X	0	0	X	Valid	Data read out from read_addr[4:0]
0	Valid	1	0	Valid	Hold Prev Phase or 160'h0	Qualified write_data[159:0] data written to write_addr[4:0]
X	X	X	1	Valid	160'h0	write_data for scan capture
1	Valid	1	0	Valid	Valid	Valid if read_addr[4:0] and write_addr[4:0] are different.

19.3.1 Partial Entry Read and Write

The partial entry read and write operations can happen when both the read and write row addresses are the same. This isn't really an address collision because those words written can't be read (logic X, don't care). Likewise those words read are not being written. So we need not worry about the write after read margin. The diagram below depics the situation:

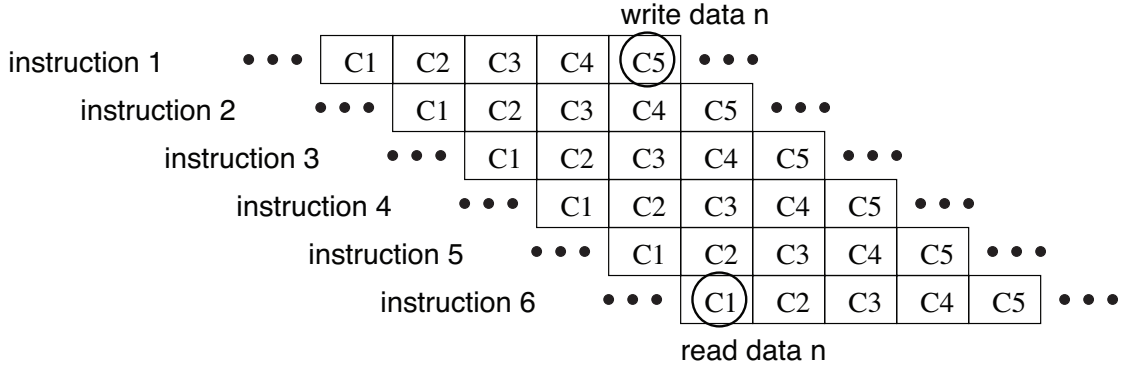
FIGURE 19-2 Partial Entry Read and Write



True for any one entry in the array.

19.4 Pipeline Diagram

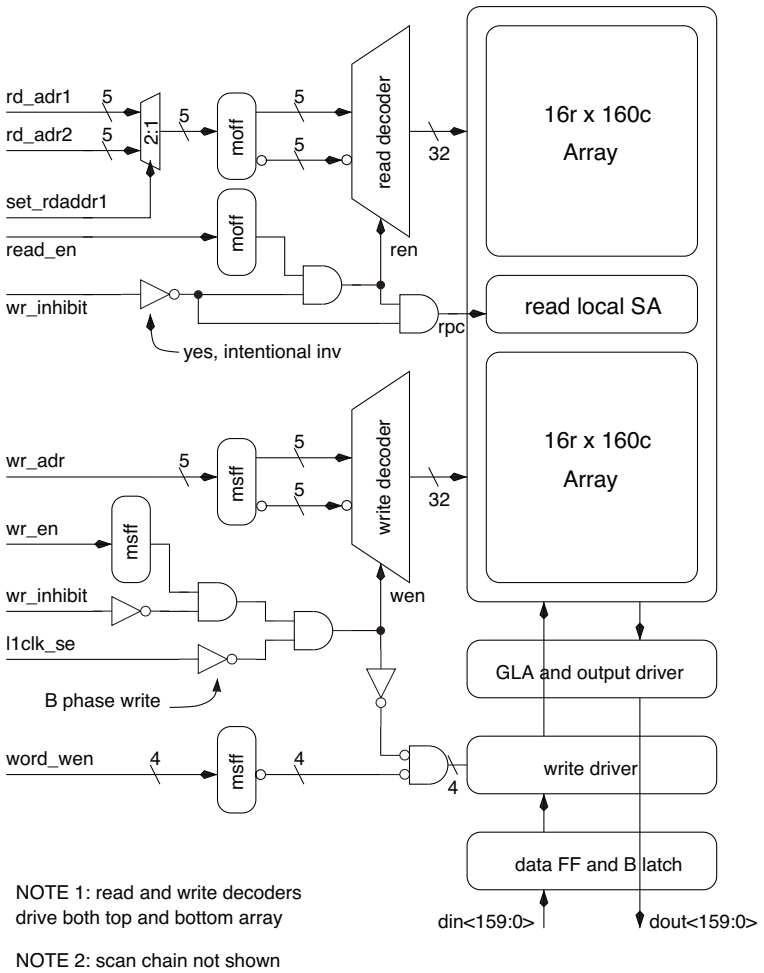
FIGURE 19-3 Pipeline Diagram



Note: read in phase A, write in phase B

19.5 Logic Diagram

FIGURE 19-4 Logic/Functional Diagram



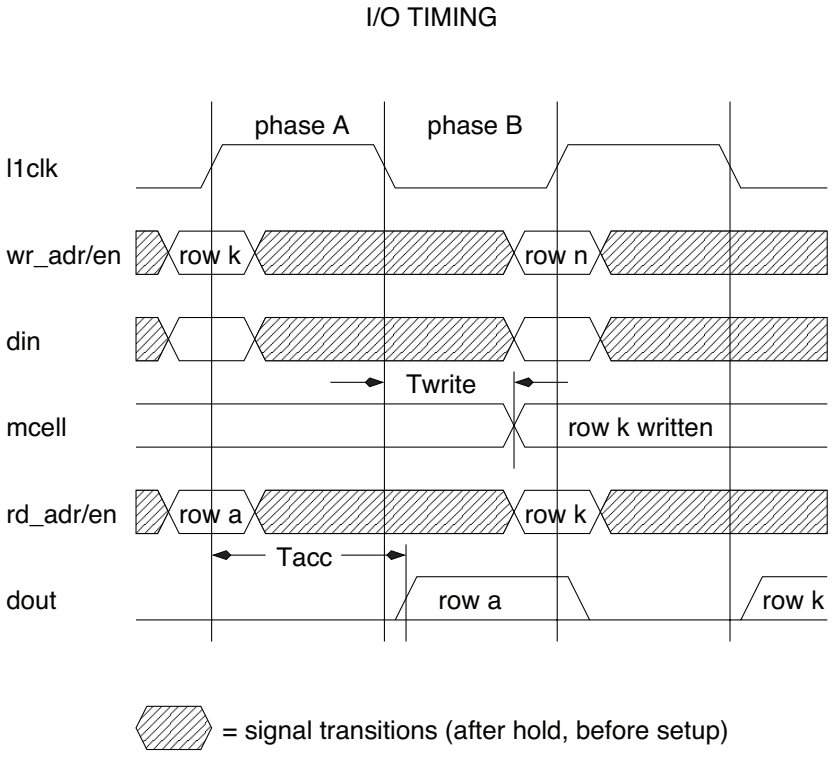
19.6 I/O Signal List

TABLE 19-2 I/O Signal List

Signal	Width	I/O	Op.	Comments
l2clk	1	I	clock	Clock
din	160	I	write	write data to array
wr_addr	5	I	write	write address
wr_en	1	I	write	write enable to array
word_wen	4	I	write	word enable for writes
rd_adr1	5	I	read	read address
rd_adr2	5	I	read	read address
sel_rdaddr1	1	I	read	select read address
read_en	1	I	read	read enable to array
dout	160	O	read	read data output
tcu_pce_ov	1	I	power	power savings
pce	1	I	power	power savings
tcu_aclk	1	I	test	scan in clock
tcu_bclk	1	I	test	scan out clock
tcu_se_scancollar_in	1	I	test	hold scan in data
tcu_scan_en	1	I	test	scan enable
tcu_array_wr_inhibit	1	I	test	array operation inhibited
scan_in	1	I	test	scan in data
scan_out		O	test	test scan out

19.7 Timing Diagram

FIGURE 19-5 VUAD I/O Timing Diagram



CHAPTER 20

Memory Controller Register File (MCU 32x72)

This chapter describes the following topics:

20.1 Functional Description

- Simple register file functionality - 1 read 1 write
- All inputs are flopped. Outputs are not flopped
- One template RTL for all compiler blocks
- One array used
- Single ended read (local and global bitlines)
- Full-swing complementary write bit lines
- Supports read and write access to different locations in the same cycle
- Collision is not supported at any frequency. In case of collision, the value of read data can not be predicted
- RTL team will ensure that read and write addresses are different when requesting read and write access in the same cycle
- Frequencies of operation are 1.4Ghz write and 375Mhz read

20.1.1 Read Operations

- Read operation in a single cycle, and occurs in phase 1
- Read entries are specified by flopped rd_adr[] and rd_en, then driven to dout[] . dout is latched inside the async block

20.1.2 Write Operations

- Write operation in a single cycle, and occurs in phase 2
- Write entries specified by flopped wr_adr[], wr_en and din[] . Inputs have to setup to the rising edge of the clock

20.2 I/O List

TABLE 20-1 MCU I/O List

Signal name	Width	I/O	Flopped/Latched	Description
Functional Signals				
rd_addr	[4:0]	Input	Flopped	Read Address
rd_en		Input	Flopped	Read Enable
wr_addr	[4:0]	Input	Flopped	Write Address
din	[71:0]	Input	Flopped	Data In
wr_en		Input	Flopped	Write Enable
dout	[71:0]	Output	Latched	Data Out
Test Related Signals				
tcu_se_scancollar_in		Input	No	Test Controller Input
wrclk		IN	No	Write Clock Input
rdclk		IN	No	Read Clock Input
scan_in		Input	No	Scan Input
bist_clk_mux_sel		IN	No	Test Controller Input
tcu_pce_ov		Input	No	Test Controller Input
tcu_aclk		Input	No	Test Scan_in Clock
tcu_bclk		Input	No	Test Scan_in Clock
scan_out		Output	No	Scan Output
tcu_array_wr_inhibit		Input	No	Test Controller Input
wr_pce		Input	No	Test Controller Input
rd_pce		Input	No	Test Controller Input

20.3 Functional Table

TABLE 20-2 MCU Functional Operation Table

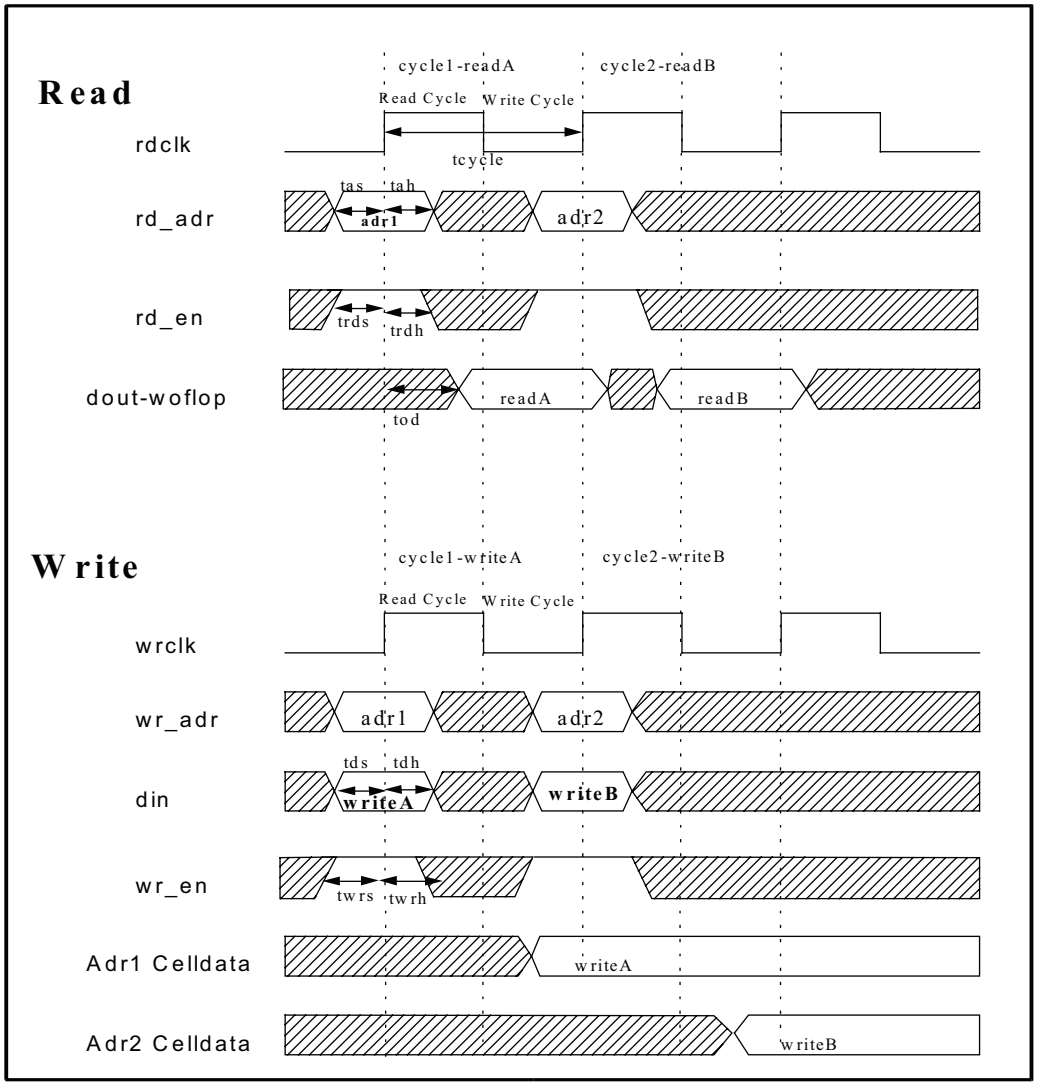
tcu_array_wr_inhibit	Read	Write	Memory Operation	Circuit
0	1	1	Asynchronous operation with non-overlapping addresses(*)	Read old data
0	0	1	Write Cycle in Phase-B	Writes new data
0	1	0	Read Cycle in Phase-A	Reads new data
0	0	0	No Operation	Holds last read data
1	x	x	No Operation	Holds last read data

(*) 0in checks in RTL monitor address collisions ..

20.4 Timing Diagram

The following figure provides a read/write input/output timing diagram for the MCU.

FIGURE 20-1 MCU I/O Timing



CHAPTER 21

Data Management Unit IOMMU Array

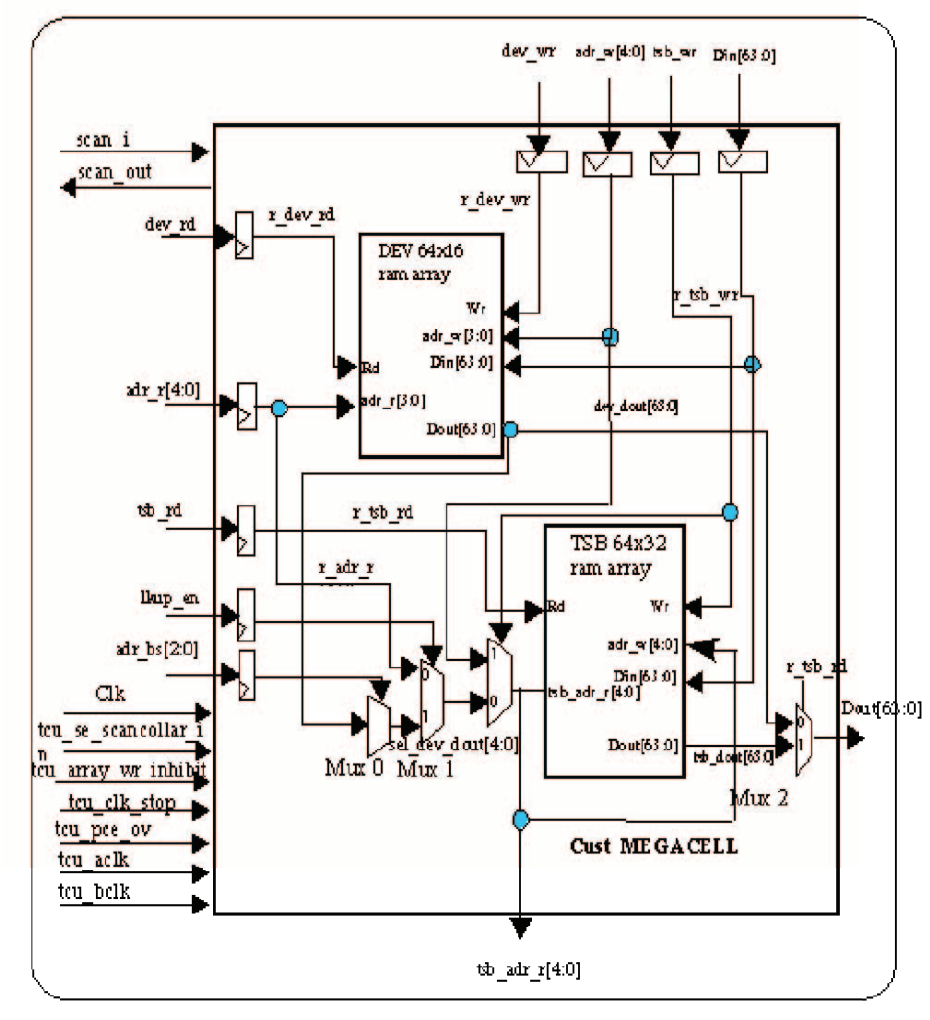
This chapter describes the following topics:

21.1 Functional Description

- The iom_sp_devtsb_cust array is part of the IOMMU block, residing in the DMU(Data Management Unit) of the PCI-Express Subsystem. It provides the one cycle two stage lookup function. It's purpose is to map PCI-Express virtual addresses to OpenSPARC T2 system physical addresses during DMA (Direct Memory Access) transfers. It has two embedded rams. The first ram Device Array (DEV ram) is structured as a synchronous 16 rows x 64 bits. The second embedded ram Translation Storage Buffer (TSB ram) is a synchronous 32 rows x 64 bits.
- Both arrays have 64 bits entry and 64 bits data_out. All inputs are flopped and outputs are latched.
- All reads and writes happen in phase A. In phase B bl/blb go in precharge. Either read(s) or one write can happen in any cycle. Means - either Dev_rd or Tsb_rd or both Dev_rd and followed by Tsb_rd, call look up (lkup_en) can happen in a cycle (phase A). Or either Dev_wr or Tsb_wr can happen in a cycle (phase A). No Read and Write will happen in any cycle.
- Frequency of operation is 350Mhz.
- All BIST are external to array.

21.2 Block Diagram

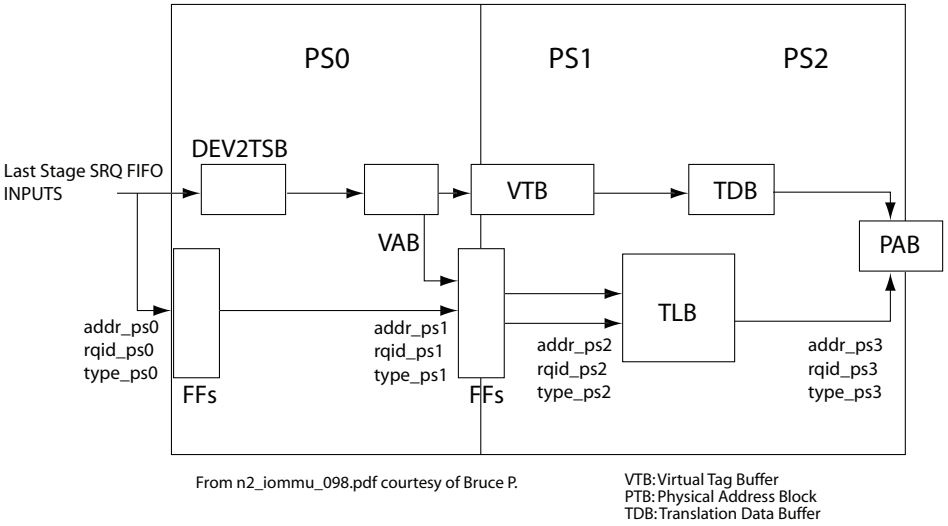
FIGURE 21-1 IOM Block Diagram



PS: 1. DEV 64 x16 means the array size is 64 bits and 16 rows

2. TSB 64 x 32 means the array size is 64 bits and 32 rows

FIGURE 21-2 IOM Pipeline Diagram



TDB: sends out Actual Physical Address to DMU

PAB checks for valid translation

21.3 I/O List

TABLE 21-1 IOM I/O List

Port Name	Width	Type	Comments	FF/Latch
clk		Input		
adr_r	[4:0]	Input		FF
adr_w	[4:0]	Input		FF
adr_bs	[2:0]	Input		FF
dev_rd		Input		FF
dev_wr		Input		FF
tsb_rd		Input		FF
tsb_wr		Input		FF
lkup_en		Input		FF
din	[63:0]	Input		FF
scan_in		Input		
tcu_se_scancollar_in		Input		
tcu_scan_en		Input		
tcu_array_wr_inhibit		Input		
tcu_pce_ov		Input		
Pce		Input		
tcu_aclk		Input		
tcu_bclk		Input		
efu_bits	[3:0]	Input		
Dout	[63:0]	Output	210ff load	Latched
tsb_adr_r	[4:0]	Output	210ff	Latched
scan_out		Output	210ff	

21.4 Functional Operation

TABLE 21-2 Functional Operation

And_ clk	Lku p_ en	Dev_ rd	Tsb_ rd	adr_r <4:0>	Adr_ bs<2:0>	Dev_ wr	Tsb_ wr	Adr_ wr<4:0>	Wr_ inhibit	Memory operation	Comments
1	0	0	0	X	X	1	0	<3:0>	0	Dev_Wr	Dout=holdslast rd
1	0	0	0	X	X	0	1	<4:0>	0	Tsb_Wr	Dout=holdslast rd
1	0	1	0	<3:0>	X	0	0	X	0	Dev_Rd	Dout=dev_dout
1	0	0	1	<4:0>	X	0	0	X	0	Tsb_Rd	Dout=tsb_dout
1	1	1	1	<4:0>	<2:0>	0	0	X	0	Lkup	Dout=tsb_dout
0	X	X	X	X	X	X	X	X	1	Wr_inhib= 1 NO-OP	All rd & wr disabled. Holds last dout
1	X	X	X	X	X	X	X	X	1	Wr_inhib= 1 scanin	All rd & WR disabled. Holds last dout

21.5 Timing Diagrams

FIGURE 21-3 Logical Timing Diagram

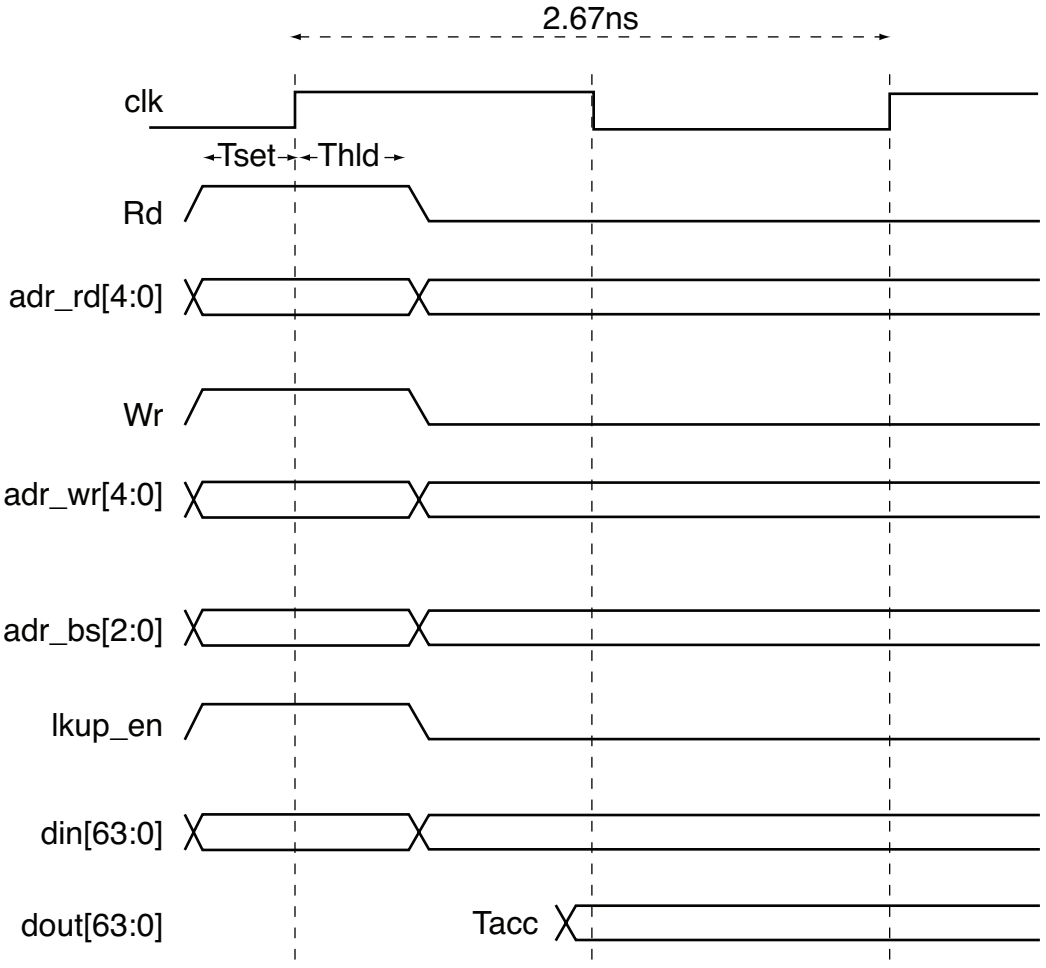


FIGURE 21-4 Timing Diagram Write

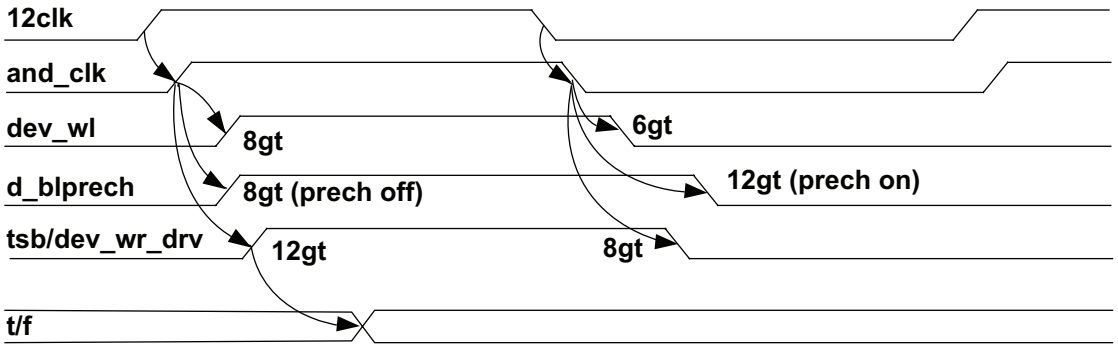
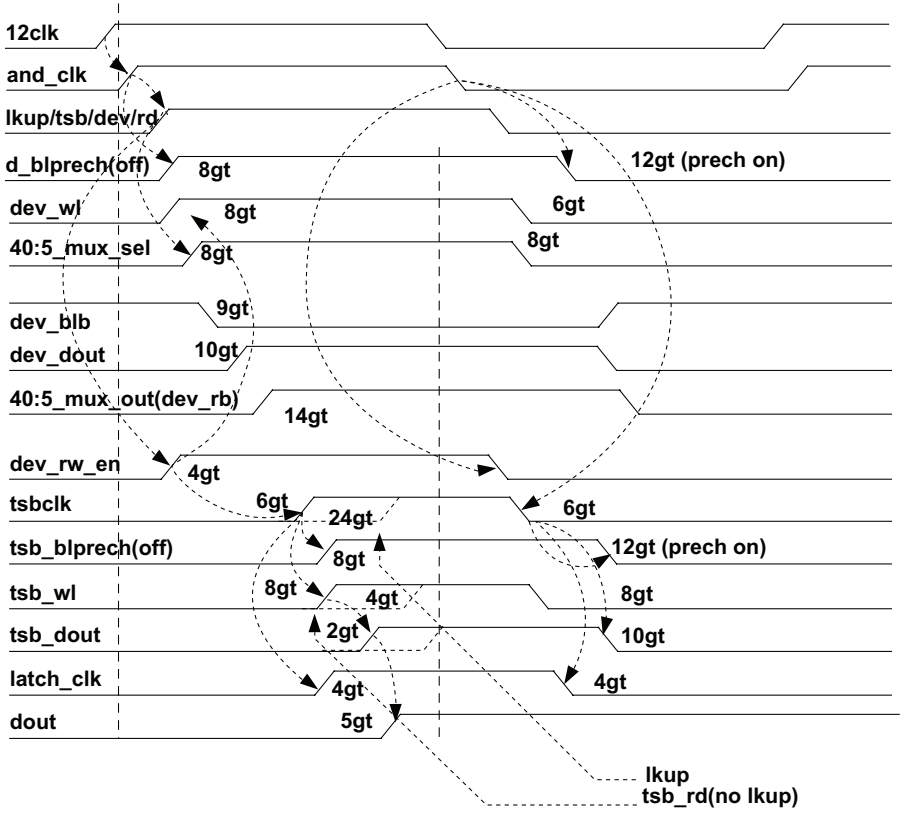


FIGURE 21-5 Timing Diagram Read



CHAPTER 22

Data Management Unit (128x132) Register File

This chapter describes the following topics:

22.1 Functional Description

- Simple register file functionality :1 read, 1 write
- All inputs are flopped. Outputs are latched within the block to hold read data,
- There is combinational logic outside the block and then it is flopped.
- It is a 128x132 1r1w Register File.
- Single ended reads. Full swing for read bit lines
- Full-swing complementary write bit lines
- There is a 0in monitor in RTL which will give out xx's if read and write to the same location occurs in 1 cycle. Physical implementation allows read and write to the same location without any problems.
- Frequency of operation 375 Mhz.
- Both read and write have separate enables .
- All BIST external to array. The block supports both BIST and Macrotest.

22.1.1 Read Operation

- Read operation in a single cycle, and occurs in phase 1
- Reads from entries specified and flopped by rd_adr[] and rd_en, then drives to dout[] . The data from dout will have some combinational logic outside the block and flopped after this logic.

22.1.2 Write Operation

- Write operation in a single cycle, and occurs in phase 2
- Writes from entries specified and flopped by wr_adr[], wr_en and din[] . The inputs have to setup to the rising edge of the clock.

22.2 Block Diagrams

FIGURE 22-1 DMU (128x132) Block Diagram

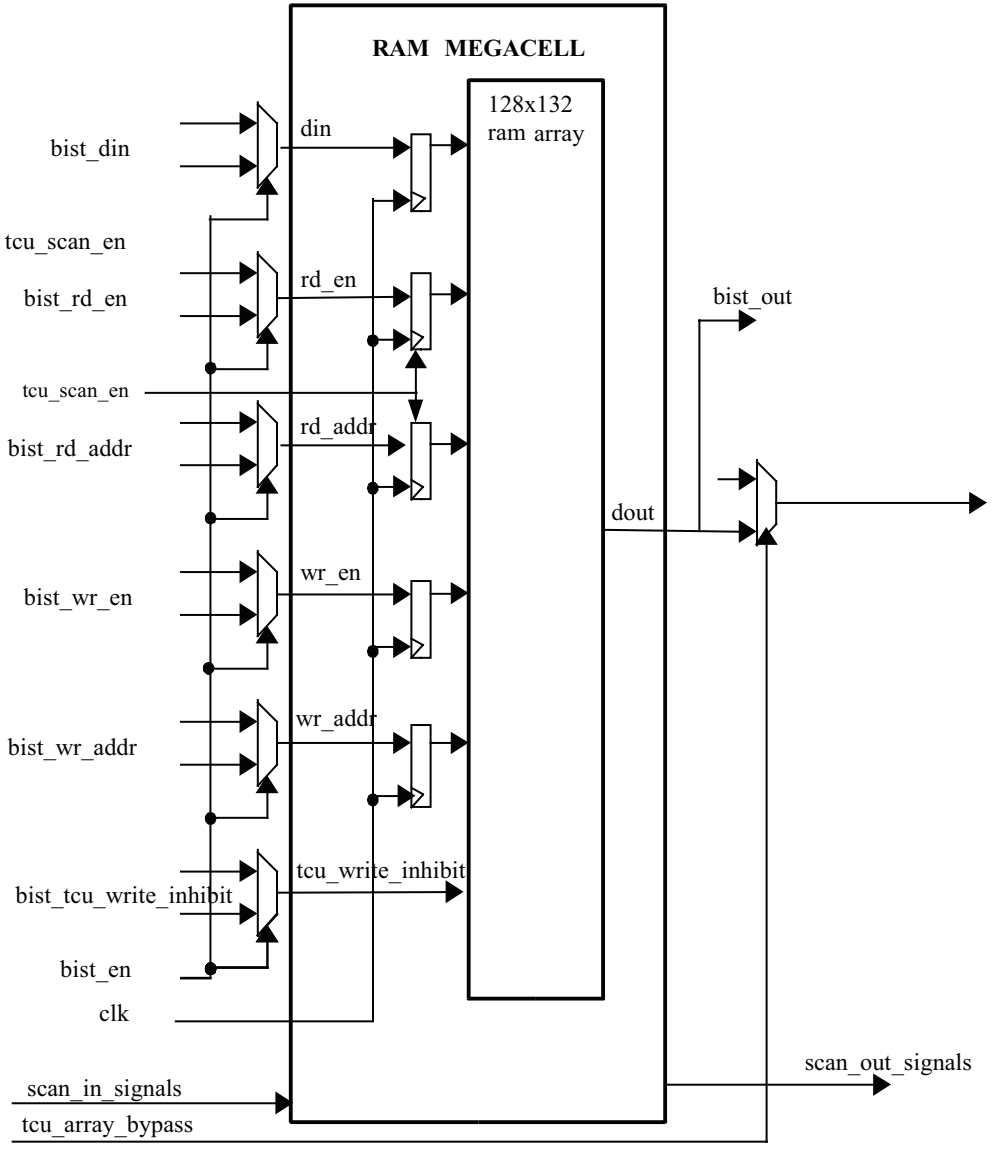
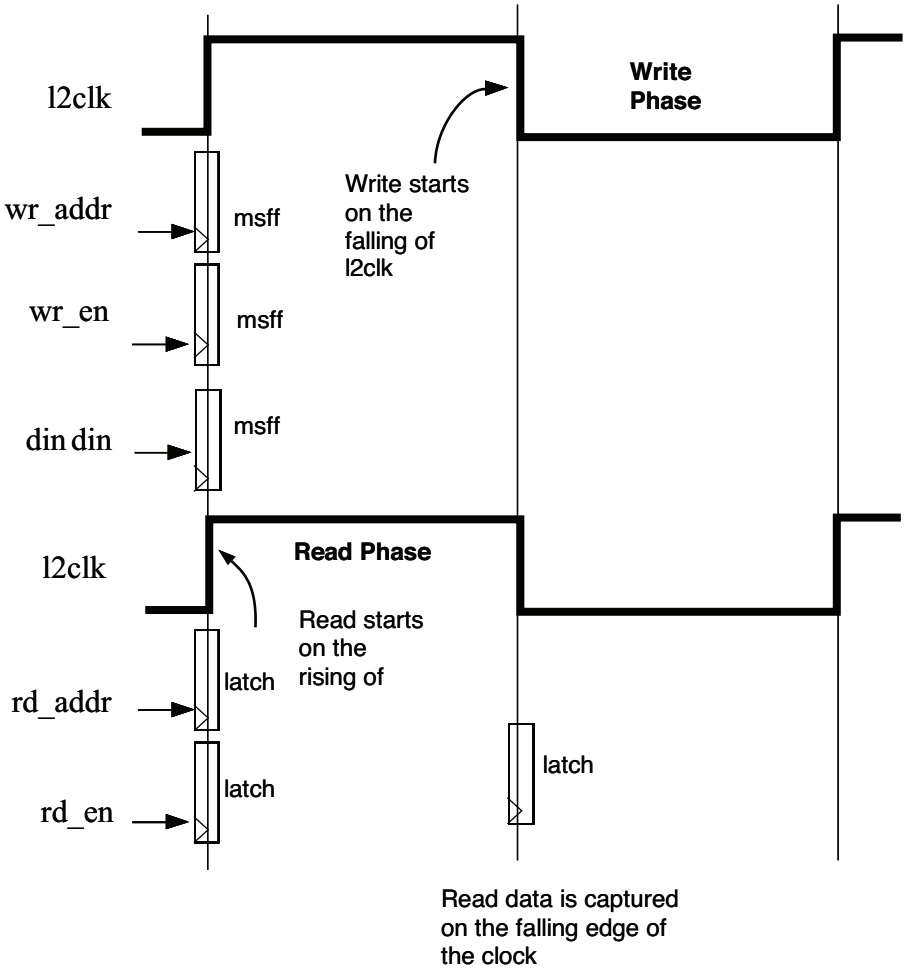


FIGURE 22-2 Internal Operational Diagram



22.3 I/O List

TABLE 22-1 I/O List

Signal Name	Width	I/O	Flopped/Latched	Description
rd_addr	[6:0]	IN	Latched	Read Address
wr_addr	[6:0]	IN	Flopped	Write Address
rd_en		IN	Latched	Read Enable
wr_en		IN	Flopped	Write Enable
din	[131:0]	IN	Flopped	Data In
dout	[131:0]	OUT	Latched	Data Out
		Clock and Test Related I/Os		
clk		IN	No	L2 clock input
scan_in		IN	No	Scan input
tcu_scan_en		IN	No	Test controller input for scan enable
tcu_se_scancoller_in		IN	No	Test controller input
tcu_pce_ov		IN	No	Test controller input
Pce		IN	No	Test controller input
tcu_adk		IN	No	Test controller scan-in clock
tcu_bdk		IN	No	Test controller scan-out clock
scan_out		OUT	No	Scan output

22.4 Functional Operation

TABLE 22-2 Modes of Operation

Enable Values		
Read	**Write**	**Memory Operation**
1*	1	RTL does not allow this to the same address in the same cycle
0	1	Write cycle at phase 2
1	0	Read cycle at phase 1
0	0	No operation

* 0in monitor in RTL ; if read address = write address in the same cycle output will be x.

22.5 Timing Diagrams

FIGURE 22-3 I/O Timing Diagram

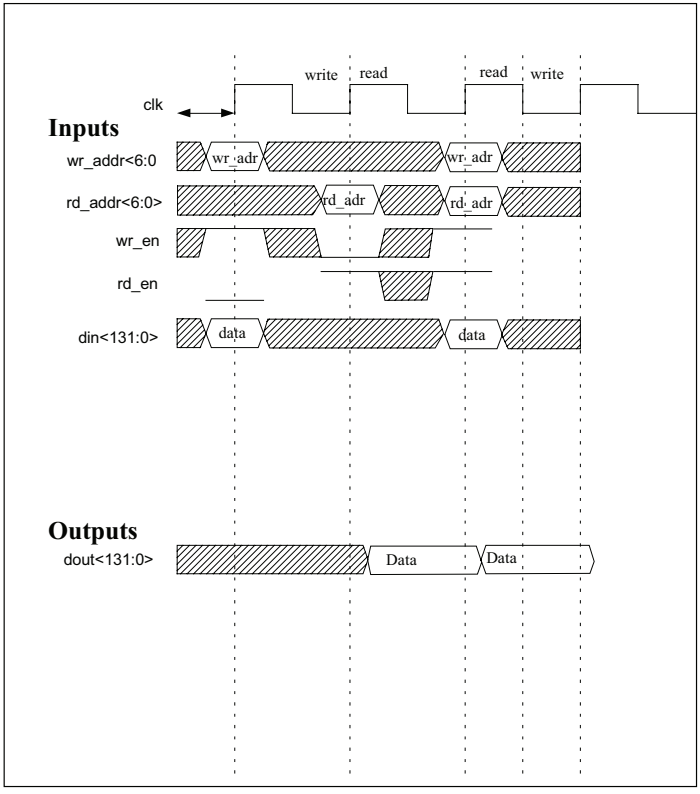


FIGURE 22-4 Internal Timing, Read

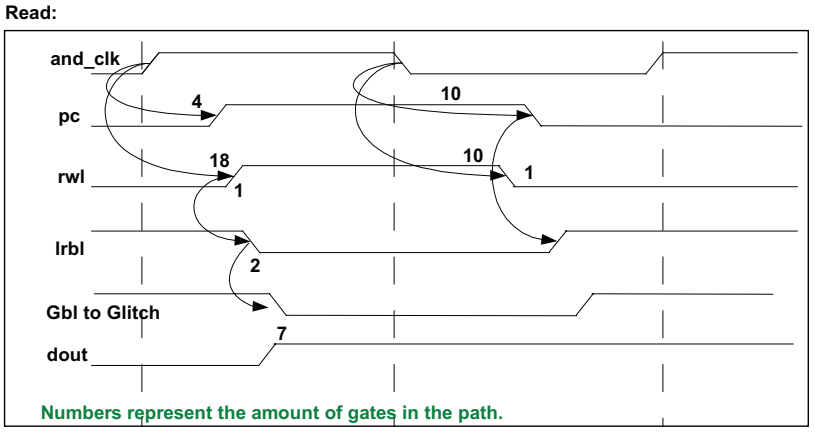
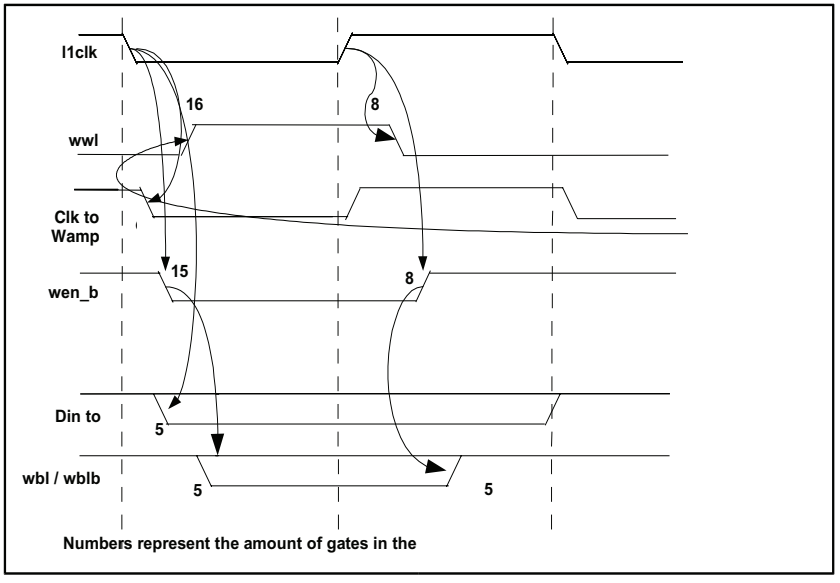


FIGURE 22-5 Internal Timing, Write



CHAPTER 23

Data Management Unit (144x149) Array

This chapter describes the following topics:

23.1 Functional Description

- dmu_dp_144x149s_cust_l is a sychronous 1 read port and 1 write port static ram array with 144 rows and 149 bits.
- It operates at clock frequency of 375Mhz.
- The functionality is summarized as:
  - Data In - 149 bits all registered at positive CLK edge with the scannable MSFFs
  - Data Out-149 bits all latched at CLK low
  - Read/Write Address-8 bits for both write and read addresses with the scannable latches and MSFFs
  - Enables-Separate 1 read and 1 write enables all flopped and scannable
  - Clock- 375Mhz

23.1.1 Read Operation

- Read Address is registered at clock rising edge and ANDing with Read enable and clock A-phase (clock signal = high)
- Read operation is finished within a single cycle and occurs in phase One
- Precharge in local and global bitline in clock B-phase
- Decode and Wordline are activated in clock A-phase
- Evaluations of the local and global bitlines in clock A-phase
- The output latch store the value of global bitline in clock B-phase

23.1.2 Write Operation

- Write Address is latched at clock rising edge and ANDing with Write Enable and inverted clock.
- Precharge the write bitline in clock A-phase
- Activate write wordline and write bitline Driver in clock B-phase

23.2 Block Diagram

FIGURE 23-1 DMU (144x149) Block Diagram

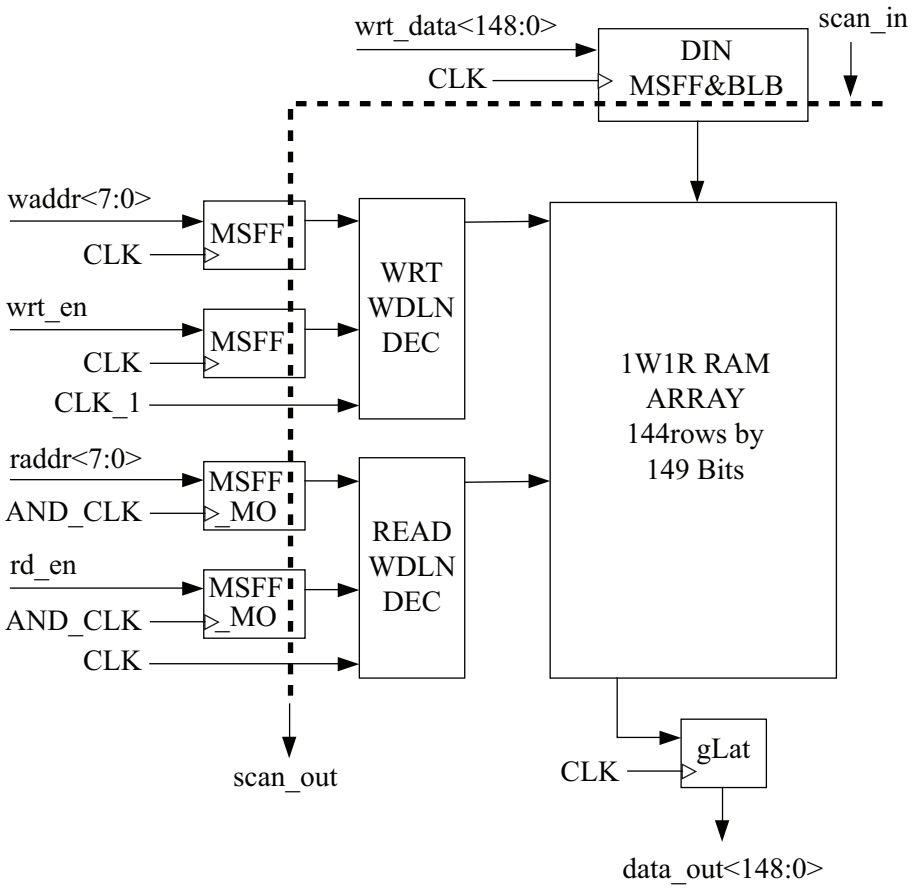
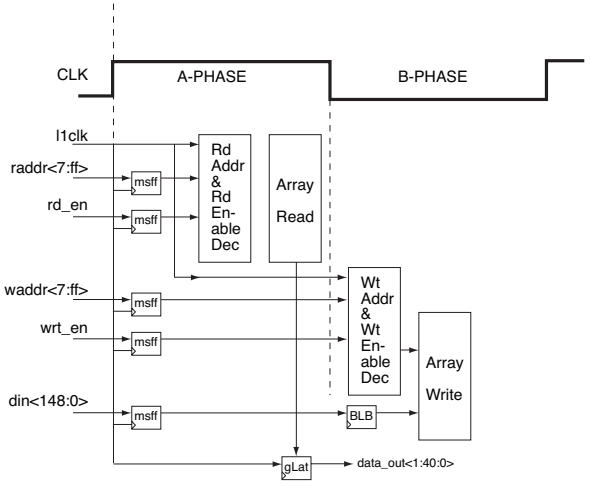


FIGURE 23-2 Pipeline Diagram



23.3 I/O List

TABLE 23-1 I/O List

Signal Name	Width	I/O	Description
rd_addr	[7:0]	IN/flopped	Read Address
wr_addr	[7:0]	IN/flopped	Write Address
rd_en	1 bit	IN/flopped	Read Enable
wr_en	1 bit	IN/flopped	Write Enable
din	[7:0]	IN/flopped	Write input data
dout	[7:0]	OUT/latched	Read output data
		Clock and Test Related I/Os	
clk*		IN	clock = 375 MHz
scan_in	1 bit	IN/flopped	Scan input data
scan_out	1 bit	OUT/flopped	Scan output data
tcu_se_scancoller_in	1 bit	IN	DFT scan enable for scan chain input
tcu_pce_ov		IN	Clock header inputs

TABLE 23-1 I/O List *(Continued)*

Signal Name	Width	I/O	Description
pce		IN	Clock header inputs
tcu_adk		IN	Scan Clock A phase
tcu_bdk		IN	Scan Clock B phase
tcu_array_wr_inhibit		IN	
tcu_scan_en		IN	Clock header inputs

* Here, input signal "clk" is the same as "l2clk" in the OpenSPARC T2 designs.

2.5 Functional Operation

TABLE 23-2 Modes of Operation

Read_en	Write_en	Wr_inhibit	Memory Operation	Read Data Output
0	1	0	Write cycle at phase B	149'b1
1	0	0	Read cycle at phase A	Din of previous writing
0	0	0	No operation	149'b1
1	1	0	(1)Within the same cycle, Write and Read at DIFFERENT addresses; (2)The control logic prohibits the design from the same address read and write, but the circuit is designed in such a way that if the same address Write/Read happens, the read data will be the content of last write operation in the same address and the current write content will be stored into the memory cells at B-phase, i.e., Dout = 149'bx (waddr==raddr)	Din of Previous writing on the same addresses.
x	x	1	No Operation	Dout = 149'bX

23.4 Timing Diagrams

FIGURE 23-3 I/O Timing Diagram

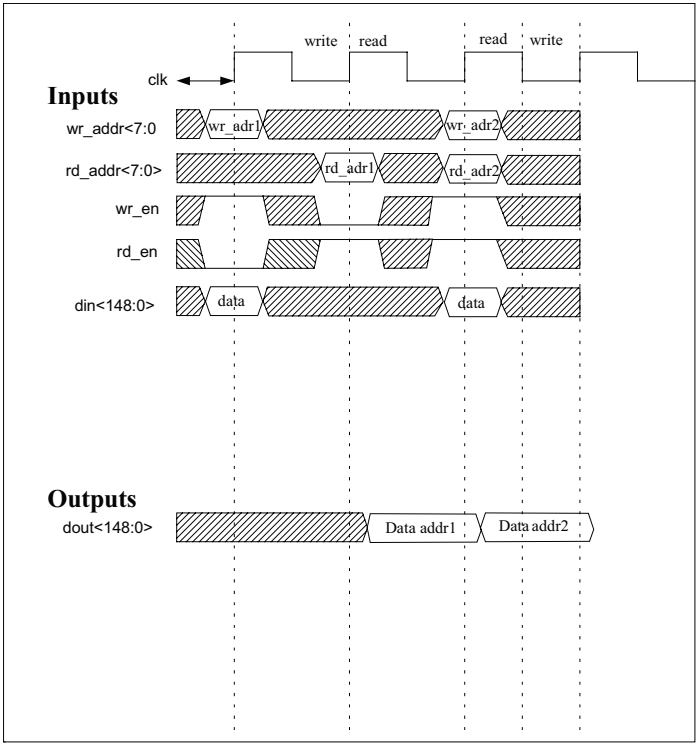


FIGURE 23-4 Internal Timing, Read

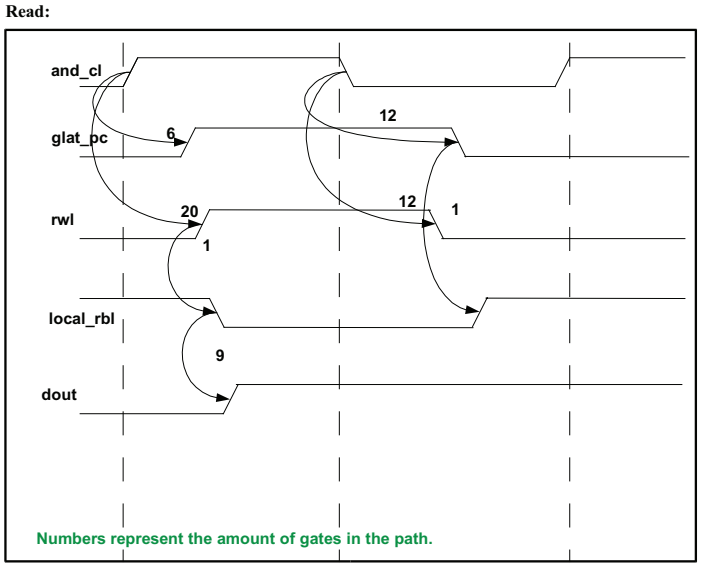
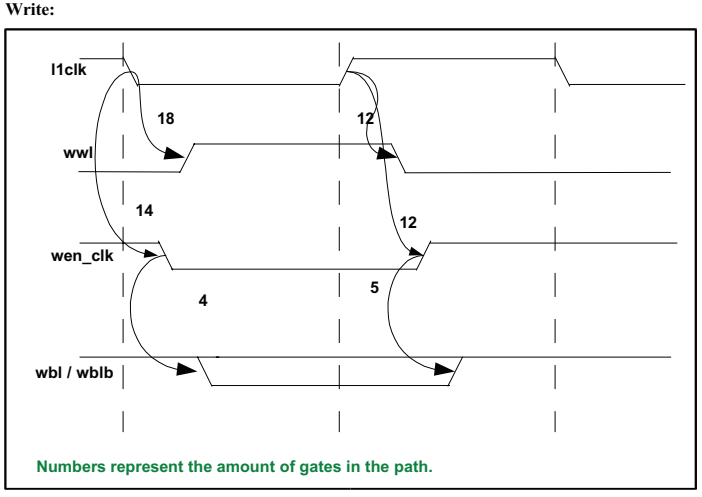


FIGURE 23-5 Internal Timing, Write



CHAPTER 24

Data Management Unit (512x60)Register File

This chapter describes the following topics:

24.1 Functional Description

- Simple register file (RF) functionality:1 read, 1 write
- All inputs are flopped. Outputs are latched within the block to hold read data, there is combinational logic outside the block and then it is flopped.
- It is a 512x60 1r1w Register File.
- The arrays use a 8T bitcell. All transistors in the bitcell are HVt
- Single ended reads. Full swing for read bit lines
- Full-swing complementary write bit lines
- There is a 0in monitor in RTL which will give out xx's if read and write to the same location occurs in 1 cycle. Physical implementation allows read and write to the same location without any problems.
- Frequency of operation 375Mhz.
- Both read and write have separate enables.
- All built-in self test (BIST) external to array. The block supports both BIST and Macrotest.
- 4:1 Column Mux is used.

24.1.1 Read Operation

- Read operation in a single cycle, and occurs in phase 1
- Reads from entries specified and flopped by rd_adr[] and rd_en, then drives to dout[] . The data from dout will have some combinational logic outside the block and flopped after this logic.

24.1.2 Write Operation

- Write operation in a single cycle, and occurs in phase 2
- Writes from entries specified and flopped by wr_adr[], wr_en and din[] . The inputs have to setup to the rising edge of the clock.

24.2 Block Diagrams

FIGURE 24-1 DMU (512x60) Block Diagram

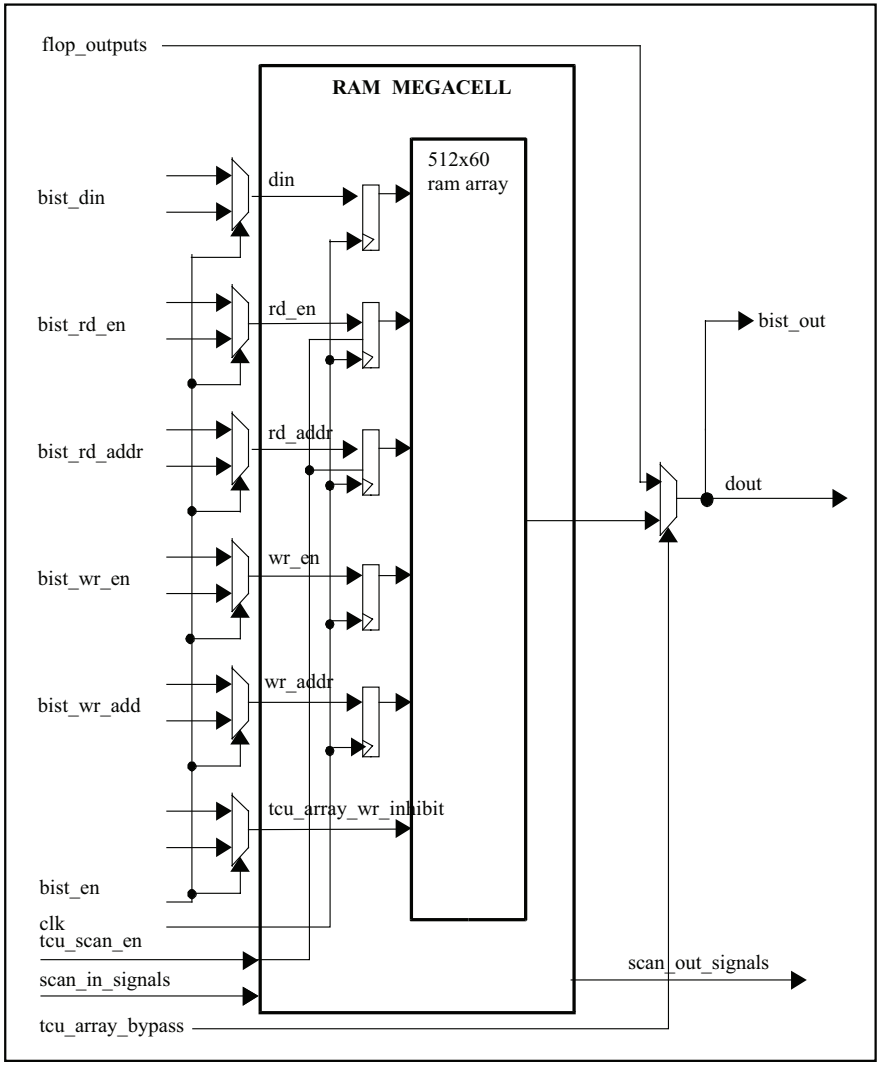
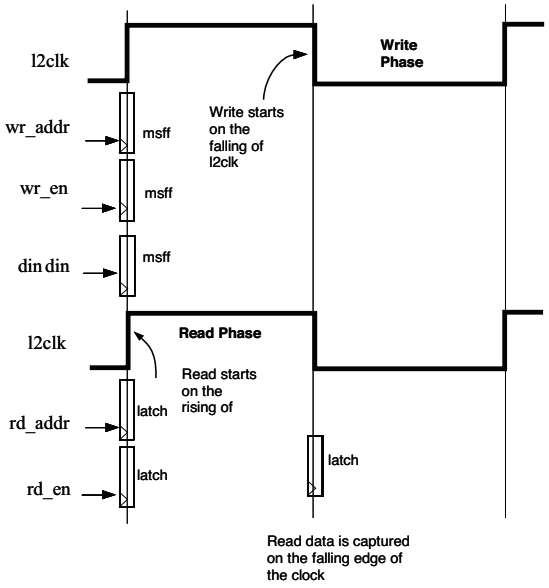


FIGURE 24-2 Internal Operational Diagram



24.3 I/O List

TABLE 24-1 DMU I/O List

Signal Name	Width	I/O	Description
rd_addr	[8:0]	IN/latched	Read Address
wr_addr	[8:0]	IN/flopped	Write Address
rd_en	1 bit	IN/latched	Read Enable
wr_en	1 bit	IN/flopped	Write Enable
din	[59:0]	IN/flopped	Data in
dout	[59:0]	OUT/latched	Data out
		Clock and Test Related I/Os	
clk		IN/No	L2 clock input
scan_in		IN/No	Scan input
tcu_scan_en		IN/No	Test controller input for scan enable

TABLE 24-1 DMU I/O List *(Continued)*

Signal Name	Width	I/O	Description
tcu_se_scancoller_in		IN/No	Test controller input
tcu_pce_ov		IN/No	Test controller input
Pce		IN/No	Test controller input
tcu_adk		IN/No	Test controller scan-in clock
tcu_bdk		IN/No	Test controller scan-out clock
scan_out		OUT/No	Scan output

24.4 Functional Operation

TABLE 24-2 Modes of Operation

Enable values		
Read	**Write**	**Memory Operation**
1	1	RTL does not allow this to the same address
0	1	Write cycle at phase 2
1	0	Read cycle at phase 1
0	0	No Operation

0in monitor in RTL ; if read address = write address in the same cycle output will be x.
When tcu_array_wr_inhibit goes high writes and reads are both disabled to the block.

24.5 Timing Diagrams

FIGURE 24-3 I/O Timing Diagram

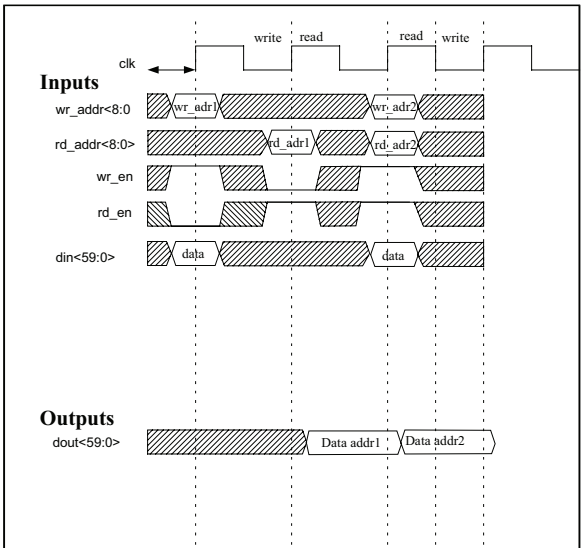


FIGURE 24-4 Internal Timing, Read

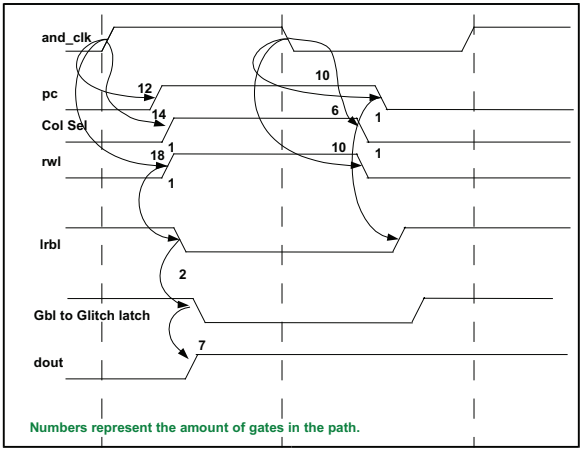
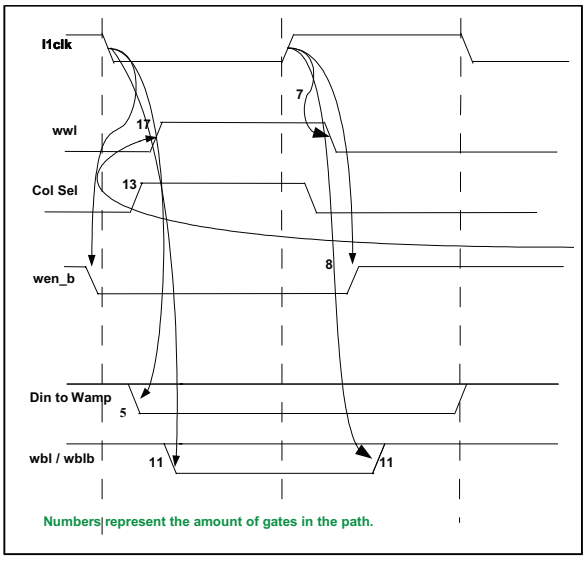


FIGURE 24-5 Internal Timing, Write



CHAPTER 25

e-Fuse Array

This chapter describes the following topics:

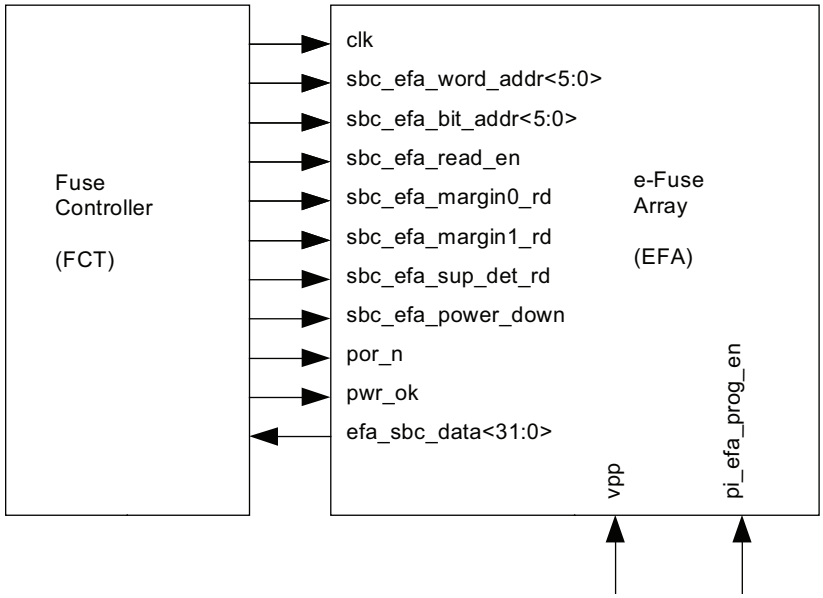
25.1 Functional Description

The electronic fuse array (EFA) consists of an array of 64x32 poly fuses and address decoders. EFA and the fuse controller (FCT) reside in the e-Fuse cluster (EFC).

25.2 EFA Interfaces

Fuse controller interfaces are shown in the following figure.

FIGURE 25-1 EFA Block Diagram



25.3 I/O List

TABLE 25-1 EFA I/O Signal List

signal name	Width	Type	Description
sbc_efa_word_addr	[5:0]	Input	Encoded row address
sbc_efa_bit_addr	[4:0]	Input	Encoded bit address
sbc_efa_read_en		Input	Read request signal
pi_efa_prog_en		Input	Program enable from I/O
sbc_efa_margin0_rd		Input	Sensing mode control
sbc_efa_margin1_rd		Input	Sensing mode control
sbc_efa_power_down		Input	Save power during no access
sbc_efa_sup_det_rd		Input	Power supplies detection read enable
por_n		Input	Soft reset (NO-OP)
pwr_ok		Input	Hard reset. Clears all DataOut flops to ground
clk		Input	Clock (ioclk)
vpp		Input	Programming voltage
efa_sbc_data	[31:0]	Output	Data out from EFA

Glossary

AOK Global address OK. State maintained by each requestor port to decide if there is room for address data at the targets for a transaction. The transaction is not driven on the bus if there is no room at the destination.

ASI address space identifier

ASIC application-specific integrated circuit

ASR ancillary state register

atomic A load and store pair with the guarantee that no other memory transaction will alter the state of the memory between the load and the store.

BIST built-in self test

collision Read and write to the same address. Collision is not enabled for asynchronous read/write.

CAM content-addressable memory

CE correctable error

CMP chip-level multiprocessor or chip-level multiprocessing

CSR control status register

Ctags The cache tags of the coherent cache in a master J-Bus port. The Ctags maintain the five MOESI cache states and participate in the J-Bus cache coherence protocol.

CTU clock and test unit

data block 64 bytes on the 128-bit data bus. Four quadwords are transferred, one quadword per clock cycle. The byte order is big-endian. In WriteMerge (WRM) transactions, valid bytes are identified with a 64-bit bytemask.

DCD data cache data array

DCDR data directory content-addressable memory

DCM directory content-addressable memory

DDR2 SDRAM double data rate-synchronous dynamic random-access memory

DFT design for testability

DIMM dual inline memory module

dirty victim A dirty cache block which is victimized (displaced) by a cache miss.

distributed address blocking Content-addressable memory function which delays matching snoops against outstanding reads.

DMA direct memory access

DRAM dynamic random-access memory

DTL dynamic termination logic

D-TLB data lookaside buffer

ECC error correction code

EFA electronic fuse array

EFC e-Fuse cluster

e-Fuse electronic fuse

FCT fuse controller

FPU floating-point unit

FRF floating-point register file

HSTL high-speed transistor logic

ICD instruction cache data array

ICDIR instruction directory content-addressable memory

IDCT instruction and data cache tag

IFU instruction fetch unit

invalidate Nullify a cache state.

IOB I/O bridge

IRF integer register file

ISI intersymbol interference

I-TLB instruction translation lookaside buffer

JBI J-Bus Interface

latency The amount of time (often measured in clock cycles) between a request to read memory and when the target data in memory is actually on the output bus.

livelock The condition when one port constantly and solely gets the bus after AOK off/on transitions and blocks out forever.

LRU-cache Least recently used cache. Used to avoid time-stamp guessing.

LSU	load and store unit
L2 Tag	Level 2 cache tag array
MAMEM	modular arithmetic memory
megacell	Custom blocks instantiated in the top-level clusters of the OpenSPARC T2 processor.
MMU	Memory Management Unit
mondo vector	A large collection of encoded interrupts (64x8).
PA	physical address, as in PA{35:32}.
PCI	Peripheral Component Interconnect
PECL	pseudo emitter-coupled logic
PIO	programmable input/output
PID	partition ID
PLL	phase-locked loop
POR	power-on reset
quadword	16 bytes. The byte order is big-endian. In noncached read/write transactions, valid bytes within the quadword are identified with a 16-bit bytemask.
RAS	reliability, availability, serviceability
RTL	register transfer level
SA	sense amplifier
scbuf	secondary cache buffer. One of three L2-cache banks.
scdata	secondary cache data. One of three L2-cache banks.
SCM	store buffer content-addressable memory
sctag	secondary cache tag. One of three L2-cache banks.
SMP	symmetric multiprocessor
snooping	The act of looking up the Ctags to determine the state of a cache block.
SRAM	static random access memory
SSI	Serial System Interface
SSO	simultaneous switching output
TAP	test access port

TLB translation lookaside buffer. A cache within an MMU that contains recent partial translations. TLBs speed up closely following translations by often eliminating the need to reread Translation Table Entries from memory.

TSB translation storage buffer. A table of the address translations that is maintained by software in system memory and that serves as a cache of the address translations.

UE uncorrectable error

Vref voltage reference

VUAD Valid, Used, Allocated, and Dirty bits.

writeback A dirty victimized data block that must be written back to memory. The victimized block is evicted from the cache due to displacement miss and is put in a writeback buffer. The block is written to memory with a writeback transaction from the J-Bus master interface.

XIR Externally initiated reset. A non-masked debug interrupt that is a special non-flow-controlled transaction that is never constrained by AOK or DOK or any interrupt flow control. All ports should respond to XIR at any time. The change transaction is similar. The XIR transaction is not supported by the OpenSPARC T2 processor.