OpenSPARC™ T1 Processor Megacell Specification

Sun Microsystems, Inc.
www.sun.com

Part No. 819-5016-10
March 2006, Revision A

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

Please
Recycle

Contents

Figures

Tables

Preface

This document provides information regarding the megacells used in the OpenSPARC™ T1 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 T1 design.

Chapter 2 describes the dual-port content-addressable memory (CAM) block in the OpenSPARC T1 processor.

Chapter 3 describes the 9-Kbyte 4-way set associative data cache data (DCD) array.

Chapter 4 describes the directory content-addressable memory (DCM), a subblock located in the sctag cluster.

Chapter 5 describes the electronic fuse array (EFA), composed of an array of 64x32 poly fuses and address decoders.

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 (IDCT).

Chapter 9 describes the integer register file (IRF), which is a 32-entry x 72-bit structure, replicated four times for each thread.

Chapter 10 describes Level 2 cache data, an inclusive, 3-Mbyte cache composed of four symmetrical banks.

Chapter 11 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 12 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 13 describes the translation lookaside buffer (TLB), commonly used as an instruction lookaside buffer (I-TLB) or a data lookaside buffer (D-TLB) for instruction fetch and for load and store units.

Chapter 14 describes the register file 16 x 128 (RF16x128d), a 16-entry, 128-bit-wide register file with two ports.

Chapter 15 describes the register file 16 x 160 (RF16x160), a 16-entry, 160-bit-wide register file with two ports.

Chapter 16 describes the register file 16 x 32 (RF16x32), a 16-entry, 32-bit-wide register file with two ports.

Chapter 17 describes the register file 32 x 108 (RF32x108), a 32-entry, 108-bit-wide register file with two ports.

Chapter 18 describes the register file 32 x 152 (RF32x152b), a 32-entry, 152-bit-wide register file with two ports.

Chapter 19 describes the register file 32 x 80 (RF32x80), a 32-entry, 80-bit-wide register file with two ports.

Chapter 20 describes register files 16x65 and 16x81, which are small register files that have 16 entries and are 65 bits and 81 bits wide, respectively.

Glossary is a list of words and phrases and their definitions.

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
OpenSPARC T1 instruction set	*UltraSPARC Architecture 2005 Specification*	950-4895-03	PDF	Online
OpenSPARC T1 processor internal registers	*UltraSPARC T1 Supplement to the UltraSPARC Architecture 2005*	819-3404-02	PDF	Online
OpenSPARC T1 processor J-Bus and SSI interfaces	*OpenSPARC T1 Processor External Interface Specification*	818-5014-10	PDF	Download
OpenSPARC T1 signal pin list	*OpenSPARC T1 Processor Datasheet*	819-5015-10	PDF	Download
Running simulations and synthesis on the OpenSPARC T1 processor	*OpenSPARC T1 Processor Design and Verification User's Guide*	819-5019-10	PDF	Download

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 T1 Processor Megacell Specification, part number 819-5016-10

CHAPTER 1

Megacells in the OpenSPARC T1 Processor

This chapter provides a summary of the megacells in the OpenSPARC T1 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

The following table lists the megacells in the OpenSPARC T1 design.

TABLE 1-1 Megacells in the OpenSPARC T1 Processor

Megacell Name	Chapter No.	Description
bw_r_cm_16x40 and bw_r_cm_16x40b	2	Dual-port CAM
bw_r_dcd	3	Data cache (D$) data array
bw_r_dcm	4	Directory CAM
bw_r_efa	5	e-Fuse array
bw_r_frf	6	Floating-point register file
bw_r_icd	7	Instruction cache (I$) data
bw_r_idct	8	Instruction and data cache tag
bw_r_irf	9	Integer unit register file
bw_r_l2d	10	Level 2 cache (L2$) data

TABLE 1-1 Megacells in the OpenSPARC T1 Processor

Megacell Name	Chapter No.	Description
bw_r_l2t	11	Level 2 cache (L2$) tag
bw_r_scm	12	Store buffer CAM
bw_r_tlb	13	Translation lookaside buffer
bw_r_rf16x128d	14	Register file 16 x 128
bw_r_rf16x160	15	Register file 16 x 160
bw_r_rf16x32	16	Register file 16 x 32
bw_r_rf32x108	17	Register file 32 x 108
bw_r_rf32x152b	18	Register file 32 x 152
bw_r_rf32x80	19	Register file 32 x 80
bw_rf_16x65 and bw_rf_16x81	20	Register files 16 x 65 and 16 x 81

CHAPTER 2

Dual-Port CAM

This chapter describes the following topics:

2.1 Functional Description of the CAMs

The dual-port content-addressable memory (CAM) has two types of dual-port CAMs – bw_r_cm16x40 and bw_r_cm16x40b. The CAM is organized as follows:

- Two CAMs are located in the secondary cache tag (sctag), one of three L2-cache banks.
- Each CAM cell has a lookup and separate read/write ports.
- Array size is 16-entry x 40-bit.
- Both CAMs have the same functional operation and structure. The only difference is the timing in the look-up operation.
  - bw_r_cm16x40 has a phase 1 look-up operation.
  - bw_r_cm16x40b has a phase 2 look-up operation.
- Read/write address inputs are fully decoded signals.
- Write is inhibited when rst_tri_en is enabled.
- During a write cycle, all 40 bits can be written in a selected entry. Write address signals are fully decoded signals. Write operation starts on the rising edge of the clock. Write access can occur on consecutive cycles without restrictions in both bw_r_cm16x40 and bw_r_cm16x40b.

  When the write_en signal is low, memory content is not affected in either bw_r_cm16x40 or bw_r_cm16x40b.

- During a read cycle, all 40 bits of a selected entry can be read out. Read address signals are also fully decoded signals. Read operation starts on the rising edge of the clock. Read access can occur on consecutive cycles without restrictions in both bw_r_cm16x40 and bw_r_cm16x40b.

  When the read_en input signal is low, data on the readout port will hold the previous read data in both bw_r_cm16x40 and bw_r_cm16x40b.

- There are two kinds of look-up operations in each CAM:
  - Partial lookup – match for bit[17:8]
  - Full lookup – match for bit[39:8]

  When the signal lookup_en is low, all the data on the match output ports goes to low. Also, look-up operations can occur on consecutive cycles without restrictions in both bw_r_cm16x40 and bw_r_cm16x40b.

  In bw_r_cm16x40, because read, write, and look-up ports are separated, read, write, and look-up operations can occur at the same cycle. If a read entry is the same as a write entry or a look-up entry is the same as write entry, a read or a look-up result will be ignored (don't care condition).

  In bw_r_cm16x40b, because read, write, and look-up ports are also separated, read, write, and look-up operations can occur at the same cycle. If a read entry is the same as a write entry, a read result will be ignored (don't care condition). However, unlike bw_r_cm16x40, if a look-up entry is the same as a write entry, a look-up result reflects the new written data because a lookup starts on the falling edge of clock whereas a write starts on the rising edge of the clock.

The modes of operation for dual-port CAMs are shown in the following table.

TABLE 2-1 Modes of Operation (16x40 & 16x40b)

read_en	write_en	lookup_en	Description
0	0	0	No operation
0	0	1	Look-up cycle
0	1	0	Write cycle
0	1	1	Write and look-up cycle (16x40) 1. Look-up Entry == Write Entry, match data is "don't care". 2. Look-up Entry =! Write Entry, match data is valid.
0	1	1	Write and look-up cycle (16x40b) 1. Look-up Entry == Write Entry, match data is valid. 2. Look-up Entry =! Write Entry, match data is valid.
1	0	0	Read cycle (16x40 and 16x40b)

TABLE 2-1 Modes of Operation (16x40 & 16x40b) *(Continued)*

read_en	write_en	lookup_en	Description
1	0	1	Read and look-up cycle (16x40 and 16x40b)
1	1	0	Read and write cycle (16x40 and 16x40b) 1. Write Entry == Read Entry, read data is “don’t care”. 2. Write Entry =! Read Entry, read data is valid.
1	1	1	Read, write, and look-up cycle 1. Read Entry == Write Entry, read data is “don’t care” (16x40 and 16x40b). 2. Look-up Entry == Write Entry, match data is "don’t care” (16x40). 3. Read Entry =! Write Entry, read data is valid (16x40 and 16x40b). 4. Look-up Entry =! Write Entry, match data is valid (16x40 and 16x40b). 5. Look-up Entry == Write Entry, match data is valid (16x40b).

2.2 Block Diagrams

This section provides the logic symbol diagram and the functional block diagrams for the dual-port CAMs.

2.2.1 Logic Symbol

Following is the logic symbol diagram for cm16x40 and cm16x40b.

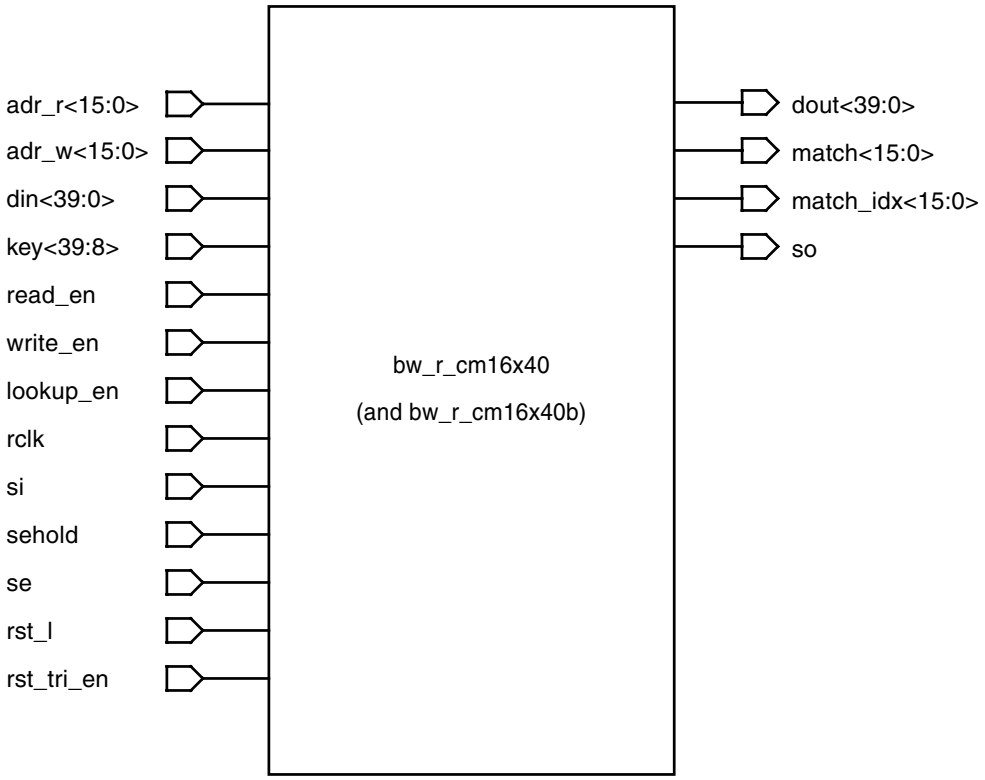

FIGURE 2-1 Logic Symbol Diagram for cm16x40 and cm16x40b

2.2.2 Functional Block Diagrams

Following is a functional block diagram of the bw_r_cm16x40 array.

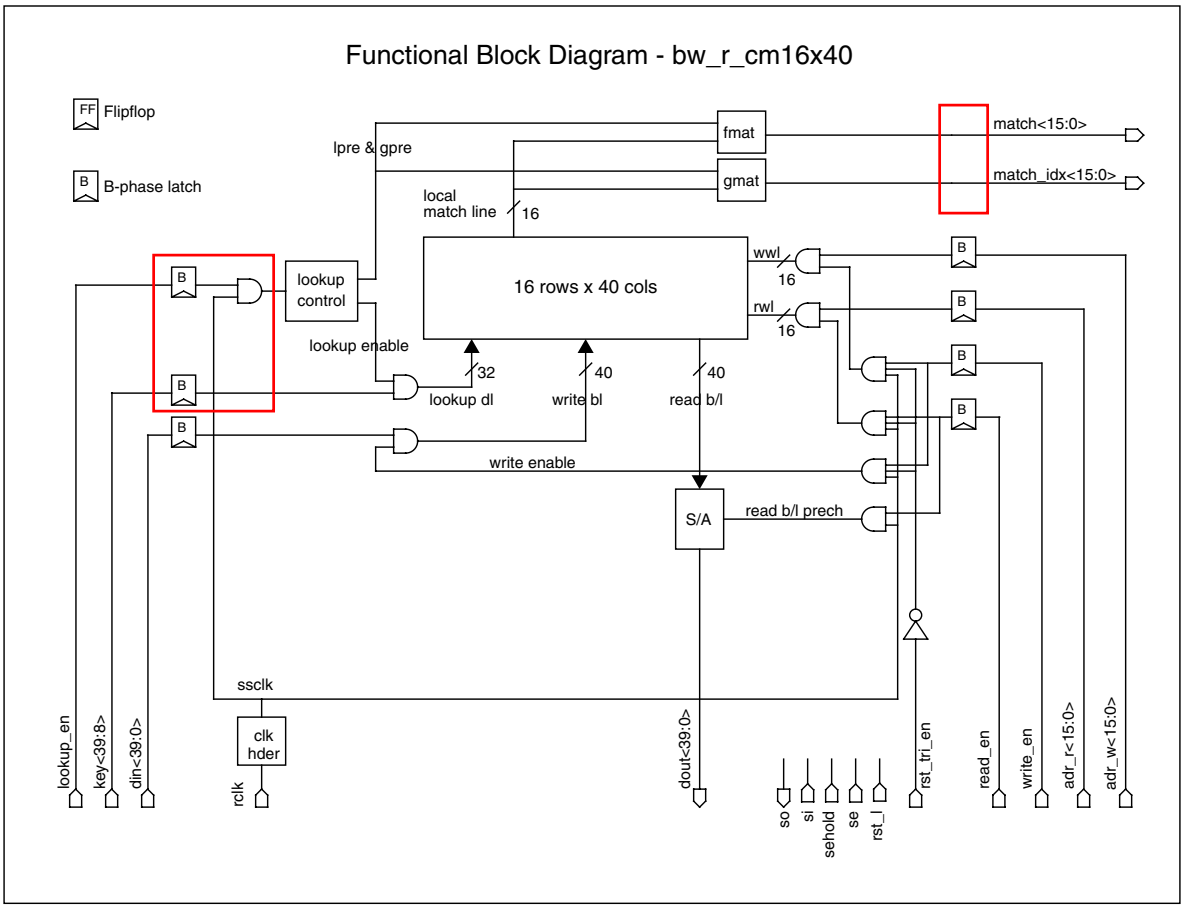

FIGURE 2-2 Functional Block Diagram of the bw_r_cm16x40 Array

Following is a functional block diagram of the bw_r_cm16x40b array.

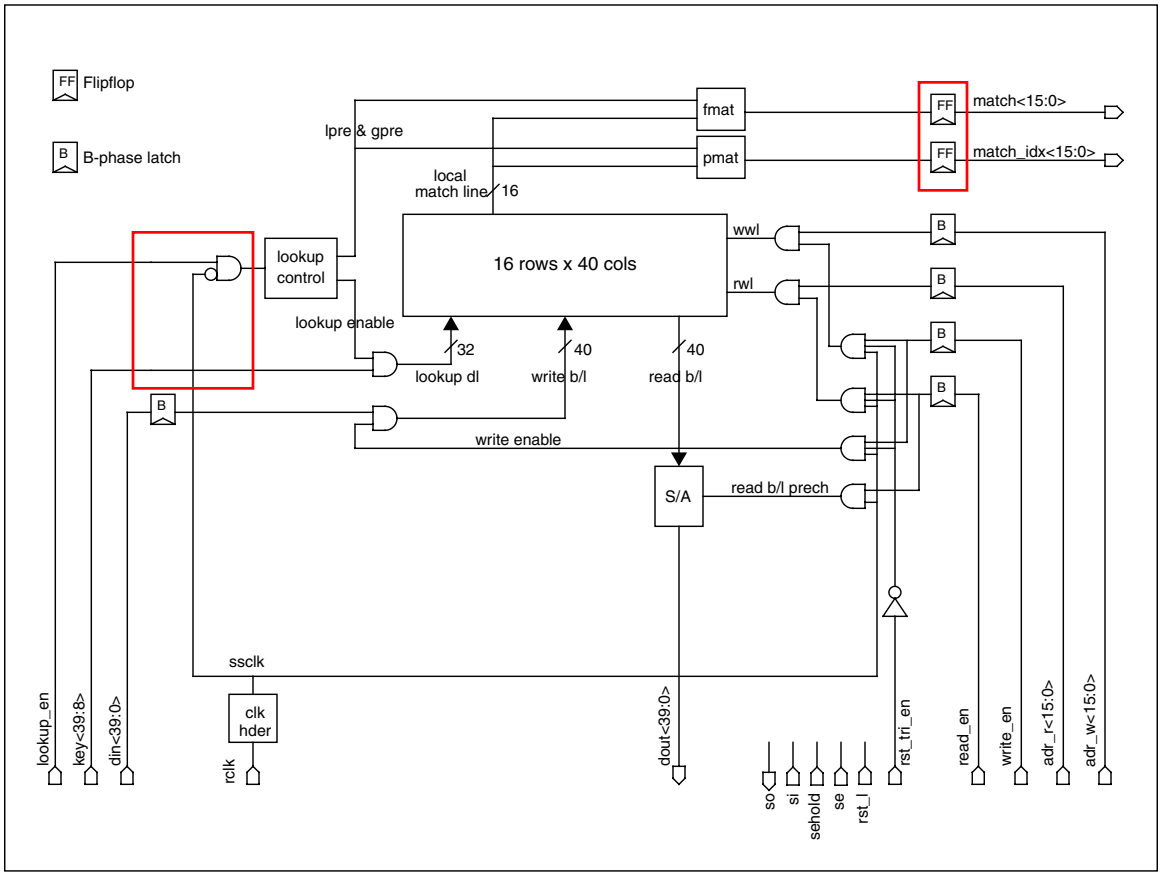

FIGURE 2-3 Functional Block Diagram of the bw_r_cm16x40b Array

2.3 I/O List

This section provides information regarding the bw_r_cm16x40 and bw_r_cm16x40b array I/O signals.

2.3.1 bw_r_cm16x40 Signal List

Following is the signal list for the bw_r_cm16x40 array.

TABLE 2-2 I/O Signal List (16x40)

Signal Name	Type	Source/Destinatrion	Description
adr_r<15:0>	Input	Multiple	Read W/L address
adr_w<15:0>	Input	Multiple	Write W/L address
din<39:0>	Input	Multiple	Write data input
key<39:8>	Input	Multiple	Look-up address input
lookup_en	Input	Multiple	Look-up enable signal
rclk*	Input	Clock grid	Clock
read_en	Input	Multiple	Read-enable signal
write_en	Input	Multiple	Write-enable signal
rst_l*	Input	-	Asynchronous reset signal
rst_tri_en*	Input	-	Memory protection at scan mode
se*	Input	-	Scan-enable signal
sehold*	Input	-	Scan hold signal
si	Input	-	Scan input
so	Output	-	Scan output
dout<39:0>*	Output	Multiple	Read data output
match<15:0>*	Output	Multiple	Full match output
match_idx<15:0>*	Output	Multiple	Partial match output

2.3.2 bw_r_cm16x40b Signal List

Following is the signal list for the bw_r_cm16x40b array.

TABLE 2-3 I/O Signal List (16x40b)

Signal Name	Type	Source/Destinatrion	Description
adr_r<15:0>	Input	Multiple	Read W/L address
adr_w<15:0>	Input	Multiple	Write W/L address
din<39:0>	Input	Multiple	Write data input
key<39:8>*	Input	Multiple	Look-up address input
lookup_en*	Input	Multiple	Look-up enable signal
rclk*	Input	Clock grid	Clock
read_en	Input	Multiple	Read-enable signal
write_en	Input	Multiple	Write-enable signal
rst_l*	Input	-	Asynchronous reset signal
rst_tri_en*	Input	-	Memory protection at scan mode
se*	Input	-	Scan-enable signal
sehold*	Input	-	Scan hold signal
si	Input	-	Scan input
so	Output	-	Scan output
dout<39:0>*	Output	Multiple	Read data output
match<15:0>	Output	Multiple	Full match output
match_idx<15:0>	Output	Multiple	Partial match output

2.4 Timing Diagrams

This section provides timing diagrams of the dual-port CAMs.

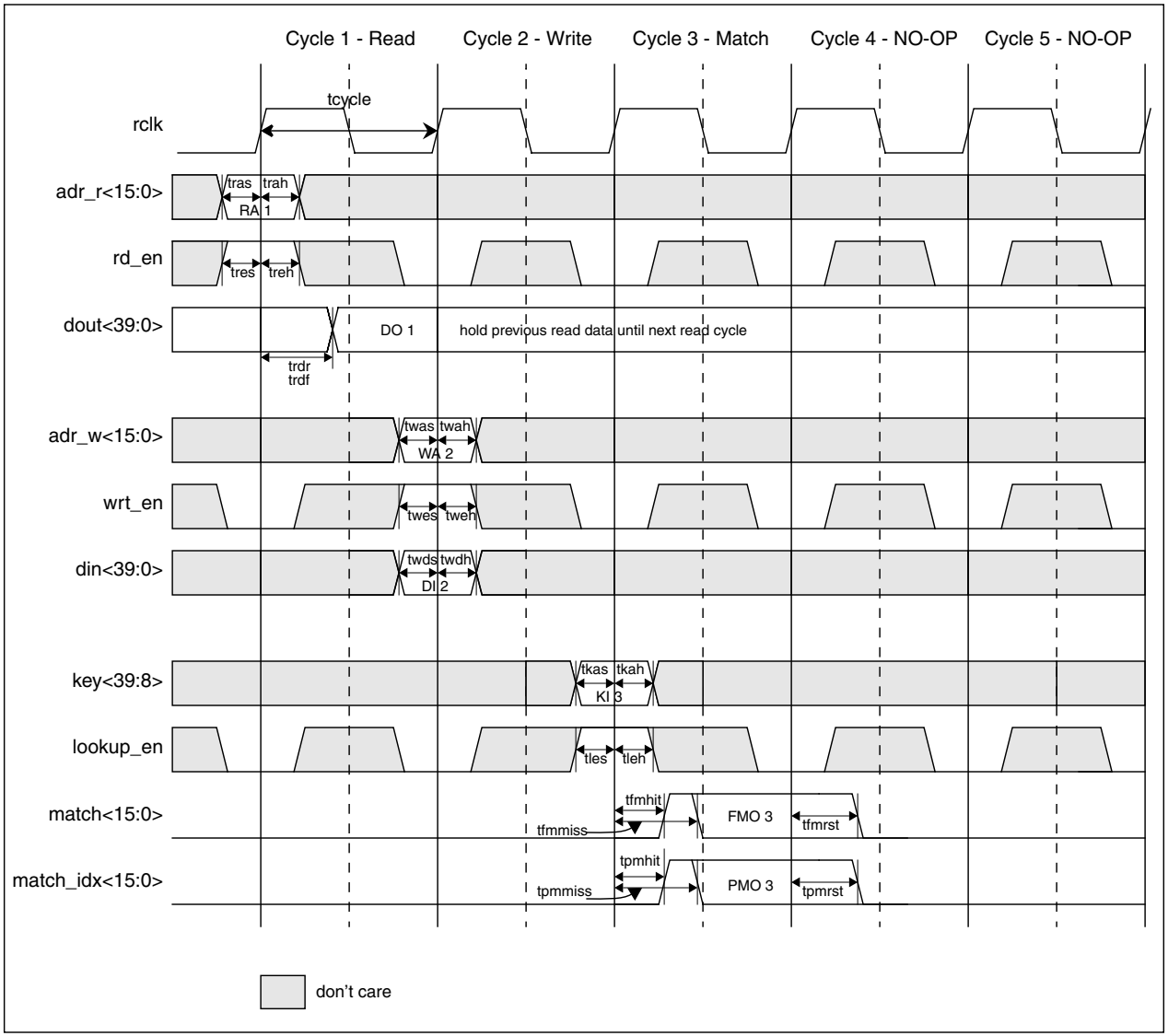

FIGURE 2-4 Timing Diagram (16x40)

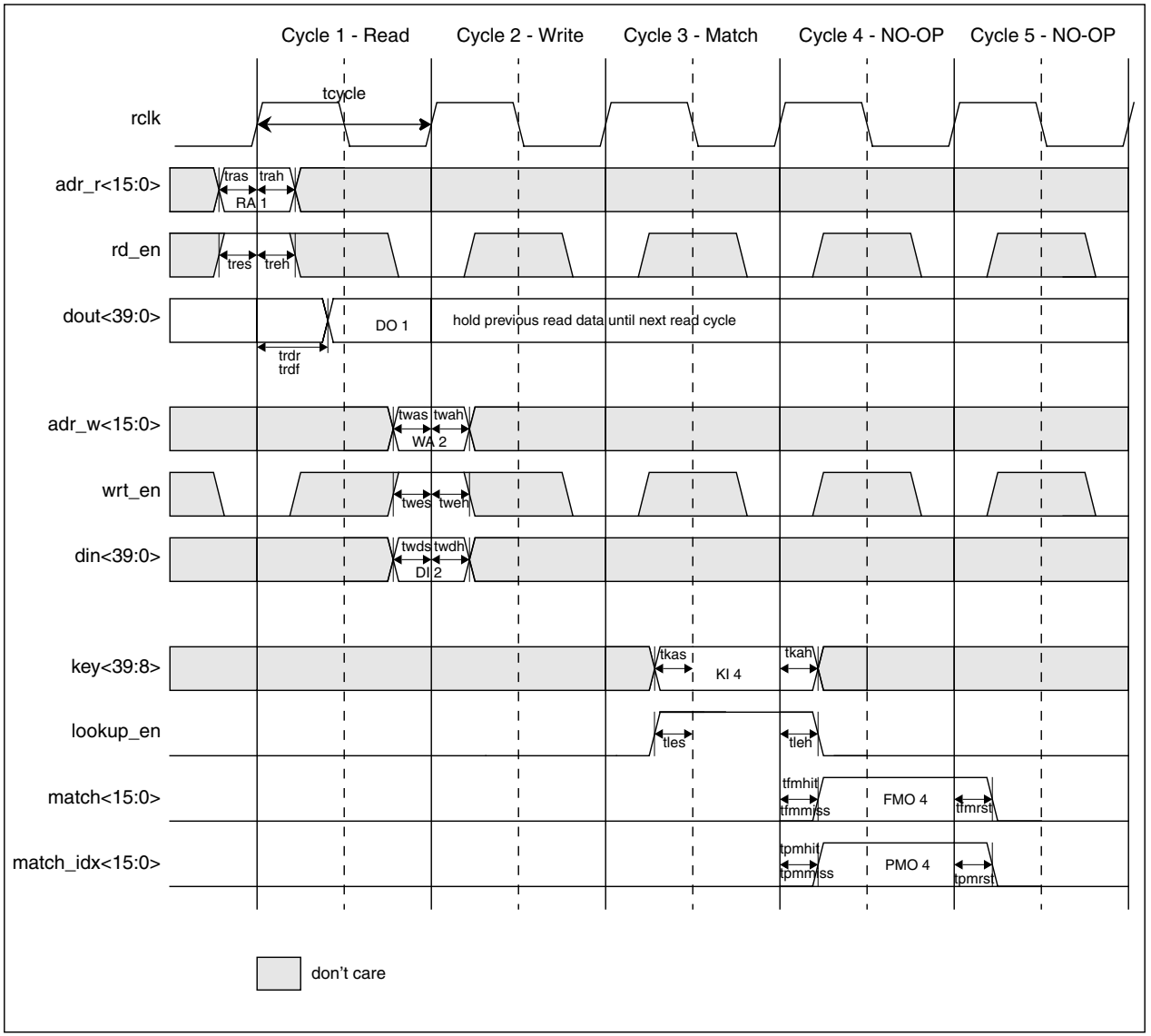

FIGURE 2-5 Timing Diagram (16x40b)

CHAPTER 3

Data Cache Data Array

This chapter describes the following topics:

3.1 Functional Description of the DCD

This section provides a functional description of the data cache data (DCD) array.

3.1.1 DCD Organization

The OpenSPARC T1 processor uses a 9-Kbyte 4-way set associative data cache. The DCD is organized as follows:

- Each way contains 128 entries of 144 bits (16 bytes, where a byte is eight data bits and one parity bit).
- There are eight copies of the data cache, one in each instance of the OpenSPARC T1 cluster.
- DCD is a single-ported static random access memory (SRAM) configured as two banks of 128 entries x 288 bits.
- Each cache line consists of four words. Each word consists of four 9-bit bytes (eight bits of data plus one parity bit).

- Each cache line is contained in one bank and each bank stores two doublewords and four parity bits per word for each of two ways, which gives a total of 288 bits per bank.

3.1.2 DCD Functions

The functions of the data cache array are the following:

- Read operation has single-cycle latency and single-cycle throughput.

  During a read access, the upper seven bits of the 8-bit input address are decoded to select one of 128 entries. The least-significant bit (LSB) is then used to select either the upper or lower doubleword to be read out. All four ways are read simultaneously. Finally, the read way inputs select a doubleword from one of the four ways to be sent to the output bus.

  The most-significant bits (MSBs) of each byte in each way are also output directly from the cache on the same cycle that the read occurs.

  The output bus is not floating even if the read way select inputs are all low.

  Reading across the cache line in the same fetch cycle is not supported.
- Parity calculate and check is provided inside the block. The final parity check result for the selected way is output from the block.
- 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).

3.2 Block Diagrams

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

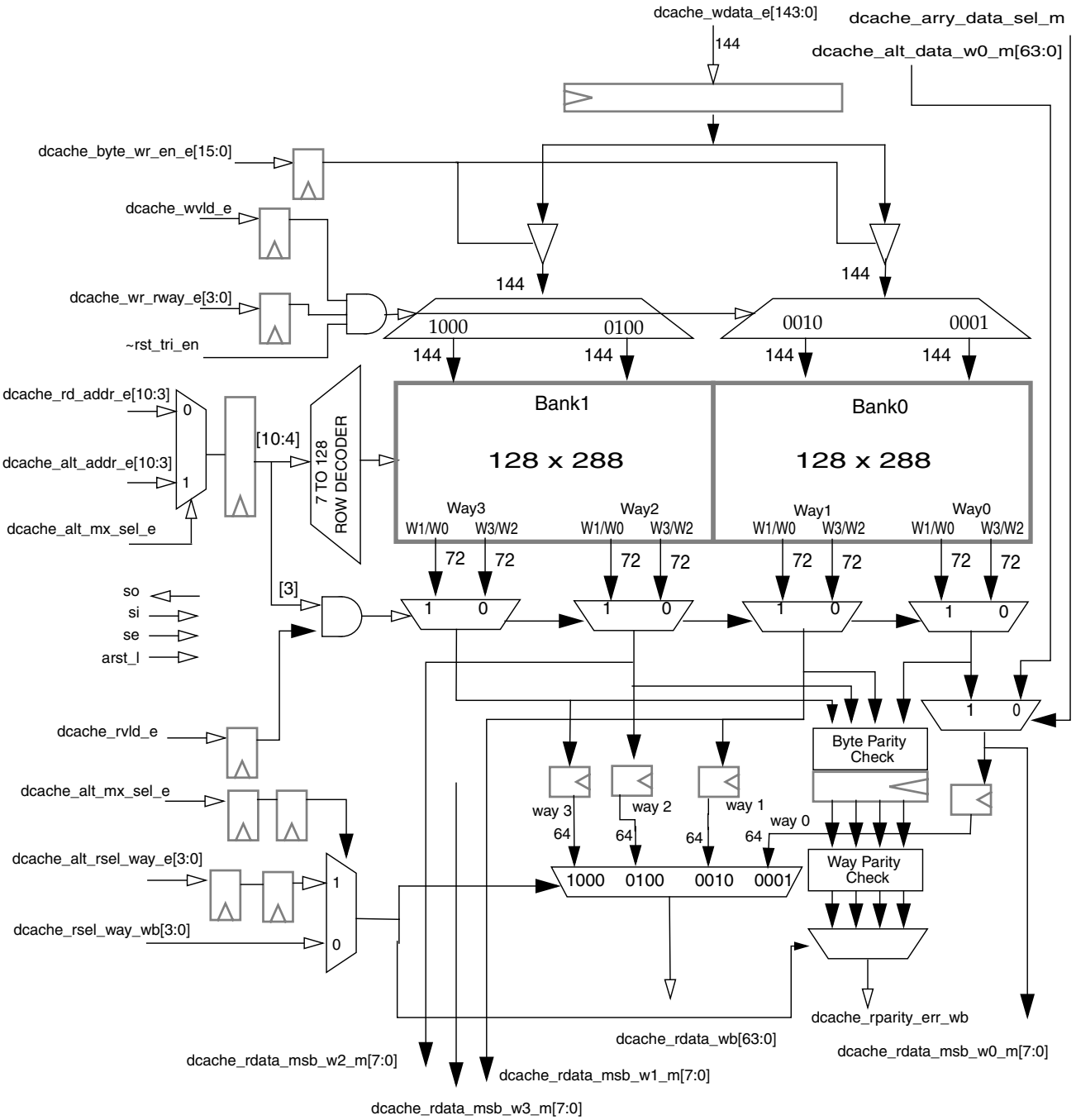

FIGURE 3-1 Functional Block Diagram of the Data Cache Data Array

3.3 DCD Logic Symbol

Following is the data cache data array logic symbol.

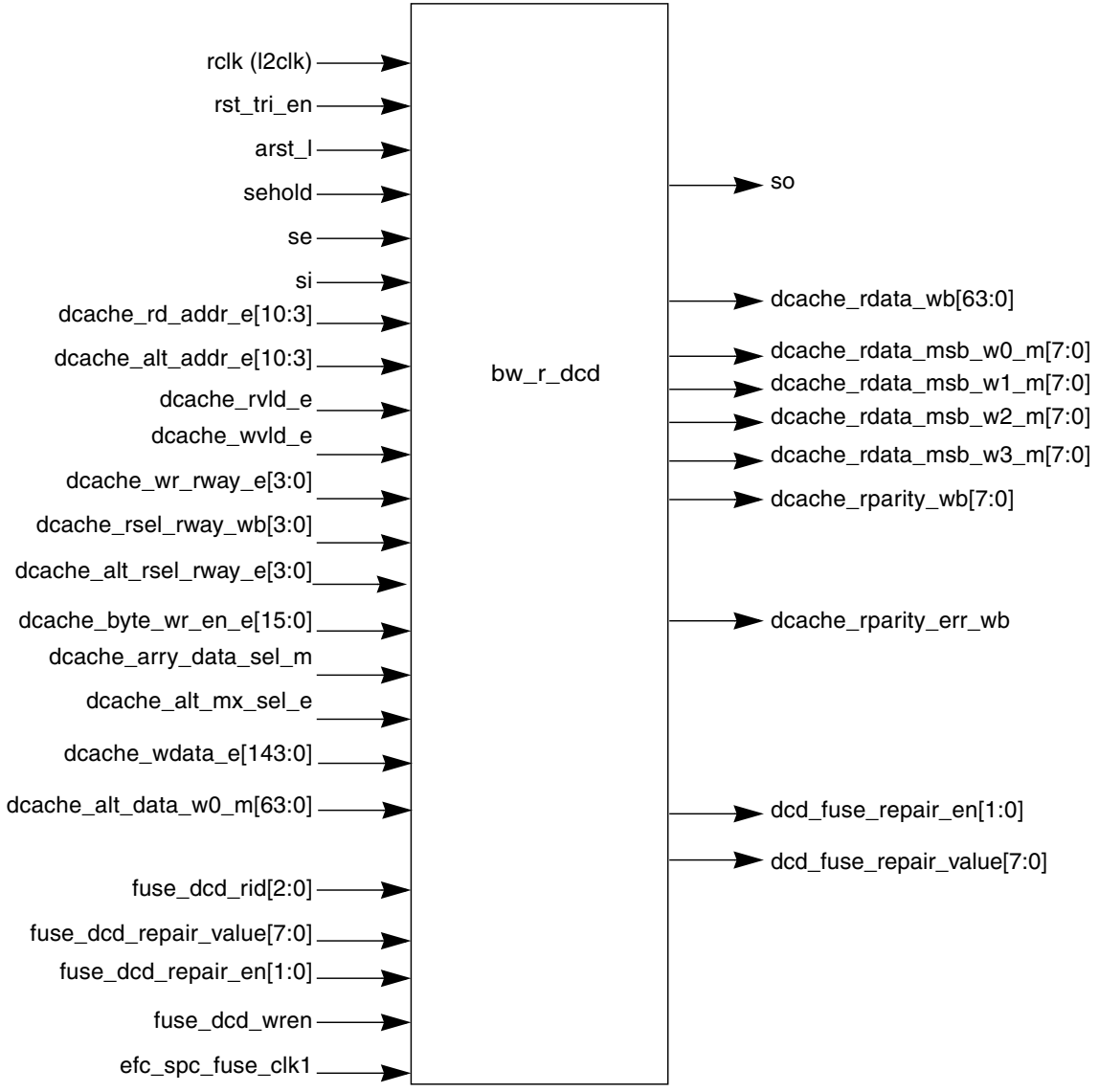

FIGURE 3-2 DCD Logic Symbol

3.4 I/O List

The following table describes the data cache data array input/output signals.

TABLE 3-1 DCD I/O List

Signal Name	Type*	Source / Destination	Description
si	Input	TBD	Scan input
se, se_hold *	Input	global	Scan enable and scan hold
rst_tri_en *	Input	global	Disable writes during scan
rclk	Input	global	L2 clock input to block
arst_l *	Input	global	Asynchronous reset for redundancy control logic
dcache_wvld_e	Input	dctl	Cache write enable
dcache_byte_wr_en_e[15:0]	Input	dctl	Byte write select
dcache_wr_rway_e[3:0]	Input	qdp2	Way select for write data
dcache_wdata_e[143:0]	Input	qdp2	128-bit data + 16-bit parity input
dcache_rvld_e	Input	dctl	Cache read enable
dcache_rd_addr_e[10:3]	Input	exu	Address input
dcache_rsel_way_wb[3:0] *	Input	dtlb	Way select for read data
dcache_rdata_wb[63:0]	Output	dcdp	Output doubleword
dcache_alt_addr_e[10:3]	Input	dctl	Address input for BIST
dcache_alt_mx_sel_e*	Input	dctl	Normal/BIST mode selector
dcache_alt_rsel_way_e[3:0]	Input	qdp2	Alternate way select for read
dcache_alt_data_w0_m[63:0]*	Input	qdp1	Alternate data for Way 0
dcache_arry_data_sel_m*	Input	dctl	Select alternate data for Way 0
dcache_rdata_msb_w0_m[7:0]*	Output	dcdp	Raw MSB bits from Way 0 read
dcache_rdata_msb_w1_m[7:0]*	Output	dcdp	Raw MSB bits from Way 1 read
dcache_rdata_msb_w2_m[7:0]*	Output	dcdp	Raw MSB bits from Way 2 read
dcache_rdata_msb_w3_m[7:0]*	Output	dcdp	Raw MSB bits from Way 3 read
dcache_rparity_wb[7:0]	Output	dcdp	Parity bits from selected way (test mode only)
dcache_rparity_err_wb	Output	dctl	Parity check result

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

Signal Name	Type*	Source / Destination	Description
so	Output	TBD	Scan output
fuse_dcd_repair_value[7:0]	Input	TBD	Fuse repair value
fuse_dcd_repair_en[1:0]	Input	TBD	Fuse repair enable
fuse_dcd_rid[2:0]*	Input	TBD	Fuse repair register address
fuse_dcd_wren*	Input	TBD	Fuse write enable
dcd_fuse_repair_value[7:0]	Output	TBD	Fuse repair value
dcd_fuse_repair_en[1:0]	Output	TBD	Fuse enables
efc_spc_fuse_clk1	Input	Global	Fuse clock

* Non-registered I/O signals are marked with an asterisk (*).

3.5 Functional Modes

This section describes cache modes of operation and signal usage in different modes.

TABLE 3-2 Cache Modes of Operation

rvld	wvld	rdata[63:0]	alt_mx	alt_rsel_way	Cache Operation
1	1	X	X	X	Not allowed. Memory contents are affected. Read data is not guaranteed.
0	1	0	0	X	Write cycle.
1	0	valid	0	X	Normal read cycle. Perform read according to input control signals.
1	0	valid	1	valid	Select alt. address and alt. read way select.
0	0	0	X	X	No operation.

TABLE 3-3 Signal Usage in Different Modes

Signal	Read	Write	NO-OP	Scan
WL	Toggle	Toggle	Toggle	Toggle
RCS	Toggle	Off	Off	Toggle if rvld=1
WCS	Off	Toggle	Off	Off
SAE	Toggle	Off	Off	Toggle if rvld=1
SAEQ	Toggle	On	On	Toggle if rvld=1
STPEQ	Toggle	Toggle	Toggle	Toggle
STPWREQ	Off	Toggle	Off	Toggle if wvld=1
WAMP	Toggle	Toggle	Toggle	Toggle

3.6 Timing Diagram

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

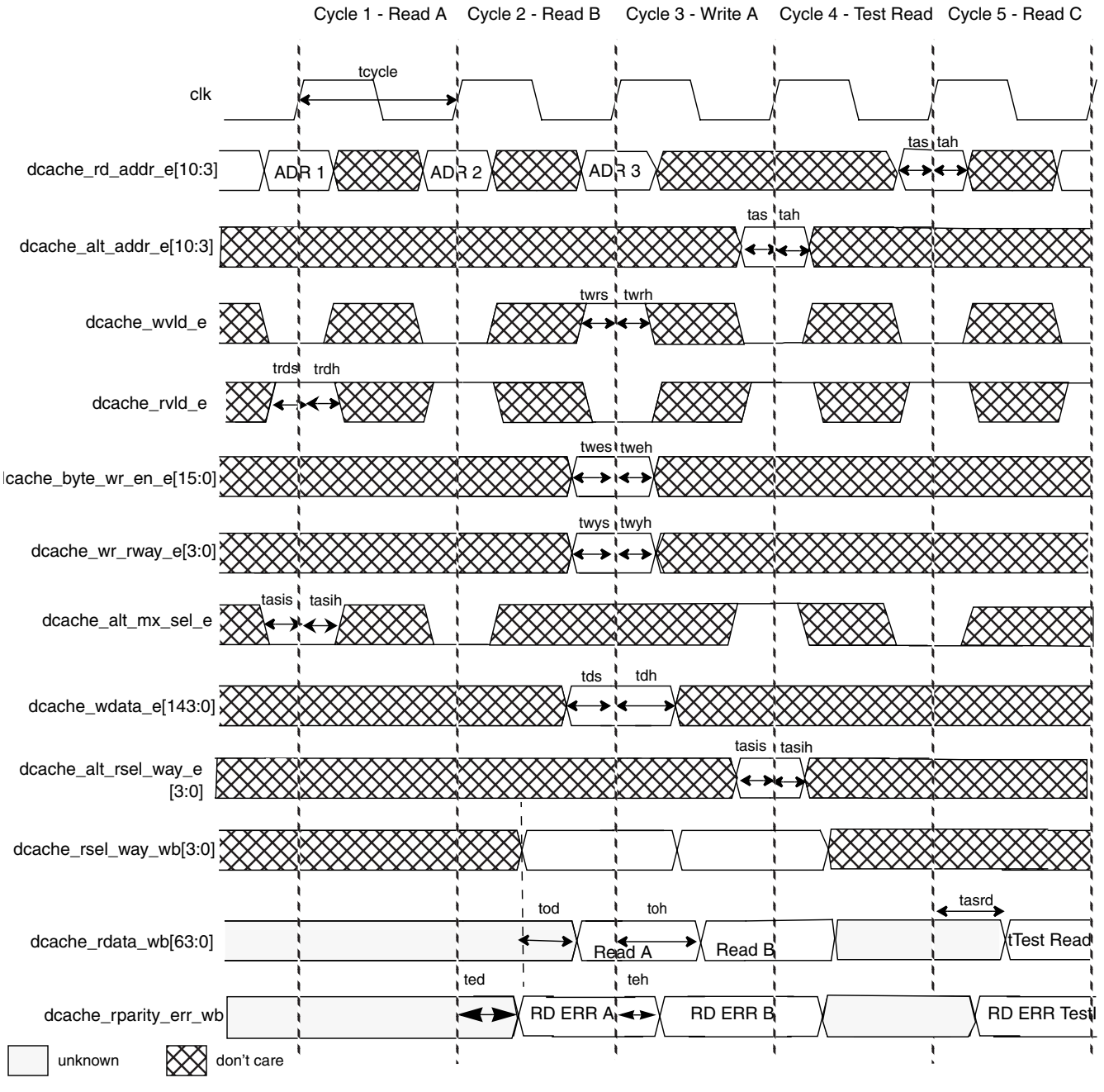

FIGURE 3-3 Read/Write Timing of Data Cache Array

CHAPTER 4

Directory CAM

This chapter describes the following topics:

4.1 Functional Description of the DCM

The directory content-addressable memory (DCM) is a subblock located in the sctag cluster. The DCM has the following characteristics:

- The directory CAM is instantiated and replicated as follows:
  - Replicated four times to create the instruction directory CAM (ICDIR) cluster (which contains a copy of the I$ Tag contents).
  - Replicated four times to form the data directory CAM (DCDIR) cluster (which contains a copy of the D$ Tag contents).
- Each directory CAM forms one row of the ICDIR or DCDIR and consists of four individual adjacent CAM panels and subarrays.

  Each panel consists of a CAM array which contains 64 entries of 32 bits (30 key bits+ 1 valid bit + 1 parity bit). Each panel has only 32 logical entries, but to avoid instantiating eight panels with 32 entries each, the CAM design has been collapsed to have four panels with 64 entries each. An external bit (write data bit 2) is used to select one hit output for every two entries.

  Only one out of four panels will be read or written in a given cycle and each panel has individual read-enable (rd_en) and write-enable (wr_en) signals to initiate the respective operations. These enables are guaranteed to be one-hot before reaching the DCM.

  Only one out of four panels can be accessed during lookup. Each panel has an individual cam_en signal to guarantee exclusivity.

- A read and a lookup can occur simultaneously (even on the same panel). However, the read/write and lookup/write combinations are not permissible. Permissible combinations are shown in TABLE 4-1.

  A read or a lookup can coincide with a warm reset, but the write operation is mutually exclusive with a warm reset (architecturally guaranteed). Otherwise, there will be a write/reset contention in the valid bits which will cause DC current, electromigration issues, and uncertain data to be stored there.

  A coincident read/reset warm operation will result in the CAM being read and latched first before the valid bit entries are reset.

  Upon an asynchronous reset, all valid bits are immediately forced to 0. The read and look-up hit values are not guaranteed because the asynchronous reset can occur any time in the cycle.

TABLE 4-1 Permissible Combinations of CAM Operations in Any CAM Panel Within Either the Instruction Directory CAM or the Data Directory CAM

rd_en (Read Enable for a Panel)	wr_en (Write Enable for a Panel)	cam_en (Lookup Enable for a Panel)	rst_warm	Outcome
0	0	0	0	No operation. Read and look-up outputs retain previous value.
0	0	0	1	Valid bits are reset in phase 2. Read and look-up outputs only change in the next cycle read and look-up operations.
0	0	1	0	Lookup occurs.
0	0	1	1	Lookup occurs in phase 1. Valid bit is reset in phase 2. row_hit outputs still reflect look-up value qualified with the old valid bit.
0	1	0	0	Write operation completes in phase 1. Bitlines recover in phase 2.
0	1	0	1	Not allowed.
0	1	1	0	Not allowed.
0	1	1	1	Not allowed.
1	0	0	0	Read occurs.
1	0	0	1	Read initiates in phase 1 and completes before valid bit is reset in phase 2.

TABLE 4-1 Permissible Combinations of CAM Operations in Any CAM Panel Within Either the Instruction Directory CAM or the Data Directory CAM *(Continued)*

rd_en (Read Enable for a Panel)	wr_en (Write Enable for a Panel)	cam_en (Lookup Enable for a Panel)	rst_warm	Outcome
1	0	1	0	Both read and look-up operations initiate in phase 1 and outputs latch in phase 2.
1	0	1	1	Read and lookup occur in phase 1 and outputs are latched in phase 2 before valid bit is reset.
1	1	0	0	Not allowed.
1	1	0	1	Not allowed.
1	1	1	0	Not allowed.
1	1	1	1	Not allowed.

- When the chip is in scan or debug mode, rst_tri_en is asserted globally and will disable the outputs of the DCM flops storing the read enable, write enable, and lookup enable. This sequence will prevent any operation from occurring during scan or debug which can potentially change the state of the CAM.
- During a read cycle, all 32 bits will be read out of one panel, but the results are flopped in external logic located in the channel space between instantiated DCM rows.
- During a write cycle, all 32 bits of a selected panel will be written. The write data in bus also doubles as the look-up key bus.
- A look-up operation compares the upper 30 key bits against the contents of every entry in the CAM panel while bit 2 muxes between the even and odd entries, and the valid bit is read out to qualify the compared result. If the CAM result is a hit and the invalid mask input is high, the valid bit memory location is updated before the beginning of the next cycle. The parity bit does not participate during lookup.
- All control signal inputs are flopped locally within the DCM, but the write and read datapath I/O signals are externally flopped in the channels between DCM rows (because they are shared between two sets of DCM rows). Ultimately, this implies there is clock skew between the external write/lookup data flops and the internal DCM dynamic write/lookup data drivers which must be accounted for by internally delaying the qualifying write/lookup driver clocks to guarantee data setup time.
- The lookup is a timing-critical single-cycle operation whose ouputs are flopped in a control block which is outside not only the directory CAM, but the entire ICDIR/DCIDR cluster stack.

- The look-up key is 33 bits wide with bit 2 used to select between even and odd rows during the CAM look-up operation (bits 0 and 1 are the valid and parity bits which serve other functions). Wr_data[2] is not stored within the array but is used as a mux select between look-up entries from even rows and odd rows. Hence, the look-up key has one more bit than the write data in bus, even though that bit is not actually stored in the CAM entry.
- Write is a single-cycle (phase A) operation starting at the panel input flops and ending with the writing of the CAM entry. The write data also doubles as the look-up key during the look-up operation.
- The read operation also occurs within a single cycle, but its data outputs are flopped by external logic which share I/O flops between two instantiated DCM rows.
- Lookup is a single-cycle (phase A) operation whose outputs are flopped externally, but should there be a hit and the corresponding mask bit set high, then the valid bit for that entry will be reset in phase B. The valid bit can also be independently reset with an asynchronous reset occurrence (such as during power-up) or if the logic signal rst_sync is asserted.

4.2 I/O List

This section describes the directory content-addressable memory (DCM) I/O signals.

TABLE 4-2 DCM I/O Signal List

Signal Name	Type	Source / Destination	Description
row_hit [31:0]	Output	dirvecdp	32-bit look-up hit bus.
rd_data0 [31:0]	Output	dir_in	32-bit read data from Panel 0.
rd_data1[31:0]	Output	dir_in	32-bit read data from Panel 1.
rd_data2 [31:0]	Output	dir_in	32-bit read data from Panel 2.
rd_data3 [31:0]	Output	dir_in	32-bit read data from Panel 3.
cam_en[3:0]	Input flop	dir_ctl	One-hot CAM look-up operation enables Panels 3, 2, 1, and 0.
inv_mask0 [7:0]	Input flop	dir_ctl	Each mask serves eight of 64 entries in Panel 0 and will force the valid bit to reset for those entries which have a CAM hit.
inv_mask1 [7:0]	Input flop	dir_ctl	Each mask serves eight of 64 entries in Panel 1 and will force the valid bit to reset for those entries which have a CAM hit.
inv_mask2 [7:0]	Input flop	dir_ctl	Each mask serves eight of 64 entries in Panel 2 and will force the valid bit to reset for those entries which have a CAM hit.

TABLE 4-2 DCM I/O Signal List *(Continued)*

Signal Name	Type	Source / Destination	Description
inv_mask3 [7:0]	Input flop	dir_ctl	Each mask serves eight of 64 entries in Panel 3 and will force the valid bit to reset for those entries which have a CAM hit.
rclk	Input	rclk grid	l2clk from fullchip clock grid.
rd_en[3:0]	Input flop	dir_ctl	One-hot read enables for Panels 3 ,2, 1, and 0.
rw_addr0[5:0]	Input flop	dir_ctl	Address bits to decode one of 64 row entries during read/write operations for Panel 0.
rw_addr1[5:0]	Input flop	dir_ctl	Address bits to decode one of 64 row entries during read/write operations for Panel 1.
rw_addr2[5:0]	Input flop	dir_ctl	Address bits to decode one of 64 row entries during read/write operations for Panel 2.
rw_addr3[5:0]	Input flop	dir_ctl	Address bits to decode one of 64 row entries during read/write operations for Panel 3.
wr_en[3:0]	Input flop	dir_ctl	One-hot enables write operation within Panels 3, 2, 1, and 0.
wr_data0 [32:0]	Input	dir_in	Doubles as the look-up key for the CAM compare operation and write input data for Panel 0. The flops for these inputs exist in the sctag_dir_in block external to the DCM.
wr_data1 [32:0]	Input	dir_in	Doubles as the look-up key for the CAM compare operation and the write input data for Panel 1. The flops for these inputs exist in the sctag_dir_in block external to the DCM.
wr_data2 [32:0]	Input	dir_in	Doubles as the look-up key for the CAM compare operation and the write input data for Panel 2. The flops for these inputs exist in the sctag_dir_in block external to the DCM.
wr_data3 [32:0]	Input	dir_ctl	Doubles as the look-up key for the CAM compare operation and the write input data for Panel 3. The flops for these inputs exist in the sctag_dir_in block external to the DCM.
rst_l_0	Input	dir_ctl	Asynchronous (cold) global reset forces the valid bit of every entry in Panels 0 and 1 to 0.
rst_l_1	Input	dir_ctl	Asynchronous (cold) global reset forces the valid bit of every entry in Panels 2 and 3 to 0.
rst_warm_0	Input flop	dir_ctl	Synchronous logical signal associated with warm global reset which forces the valid bit of every entry in Panels 0 and 1 to 0 in phase 2 only if clk is on.
rst_warm_1	Input flop	dir_ctl	Synchronous logical signal associated with warm global reset which forces the valid bit of every entry in Panels 2 and 3 to 0 in phase 2 only if clk is on.

TABLE 4-2 DCM I/O Signal List *(Continued)*

Signal Name	Type	Source / Destination	Description
rst_tri_en_0	Input	dir_ctl	Asynchronous memory write-disable signal for Panels 0 and 1 that gates off the output of the wr_en flop as well as the outputs of the flops associated with the look-up operation (mask, cam_en, rst_warm) to prevent them from erroneously invalidating the valid bits during scan shift.
rst_tri_en_1	Input	dir_ctl	Asynchronous memory write-disable signal for Panels 2 and 3 that gates off the output of the wr_en flop as well as the outputs of the flops associated with the look-up operation (mask, cam_en, rst_warm) to prevent them from erroneously invalidating the valid bits during scan shift.
si_0	Input		Scan chain input for shared Panels 0 and 1.
si_1	Input		Scan chain input for shared Panels 2 and 3.
se_0	Input		Scan chain enable for shared Panels 0 and 1.
se_1	Input		Scan chain enable for shared Panels 2 and 3.
sehold_0	Input		Scan chain flop output feedack hold for shared Panels 0 and 1.
sehod_1	Input		Scan chain flop output feedack hold for shared Panels 2 and 3.
so_0	Output		Scan chain output for shared Panels 0 and 1.
so_1	Output		Scan chain output for shared Panels 2 and 3.

The following figures show DCM read, write, and look-up panel timing diagrams.

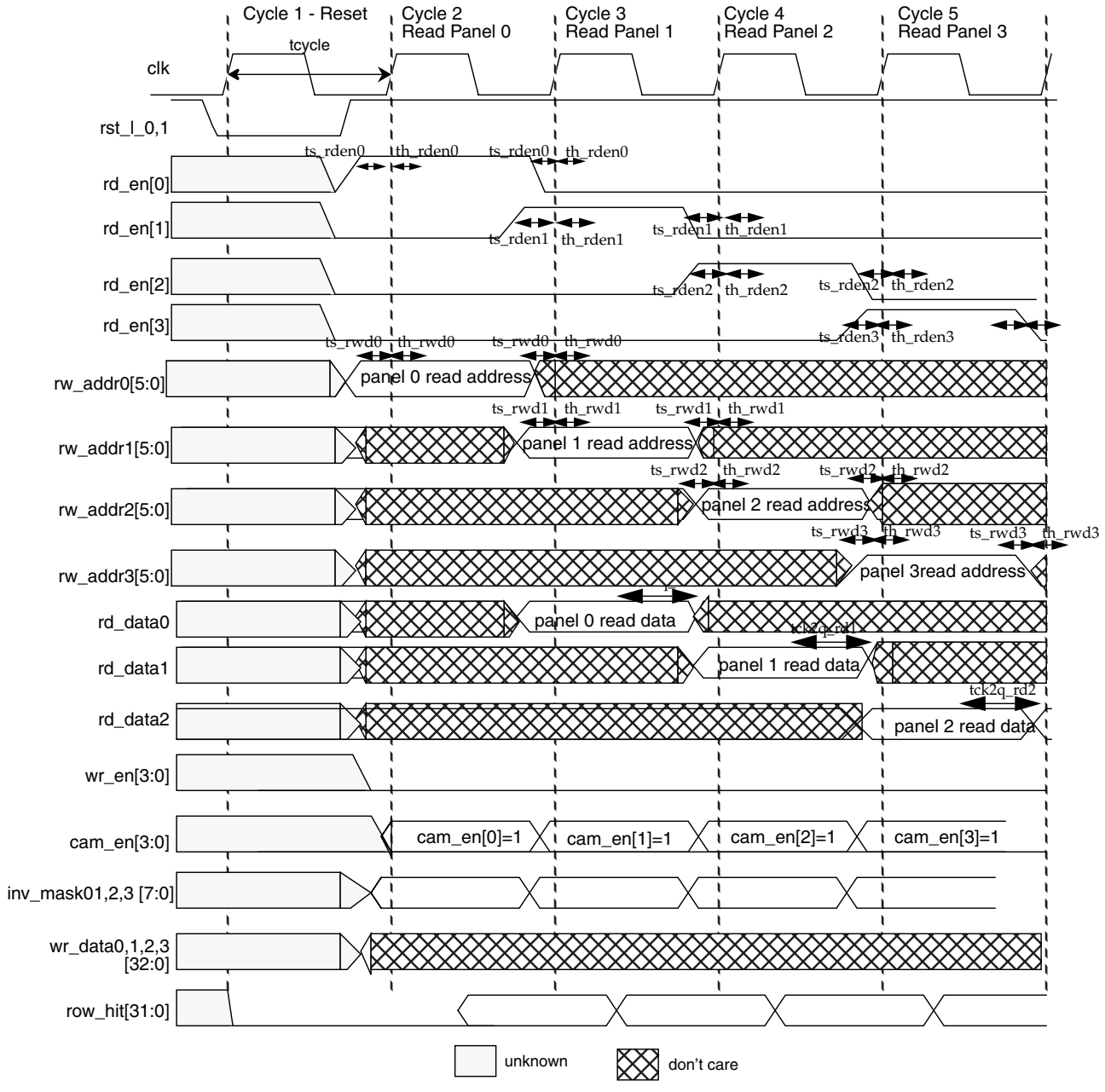

FIGURE 4-1 Read Timing Diagram of DCM Row Panel (Only 1 of 4 Panels Per Read Cycle)

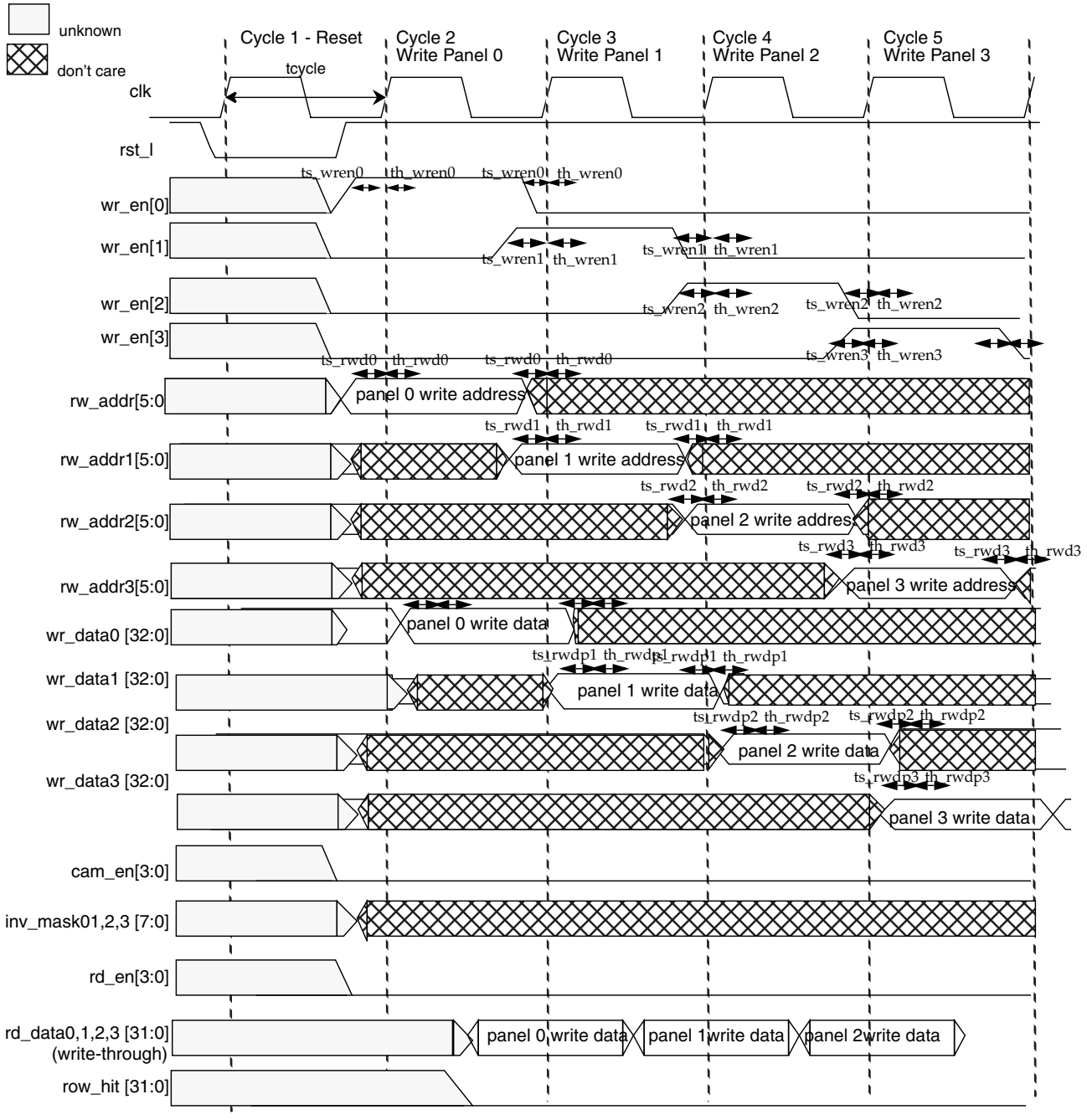

FIGURE 4-2 Write Timing Diagram of Row Panel (only 1 of 4 written per Write cycle)

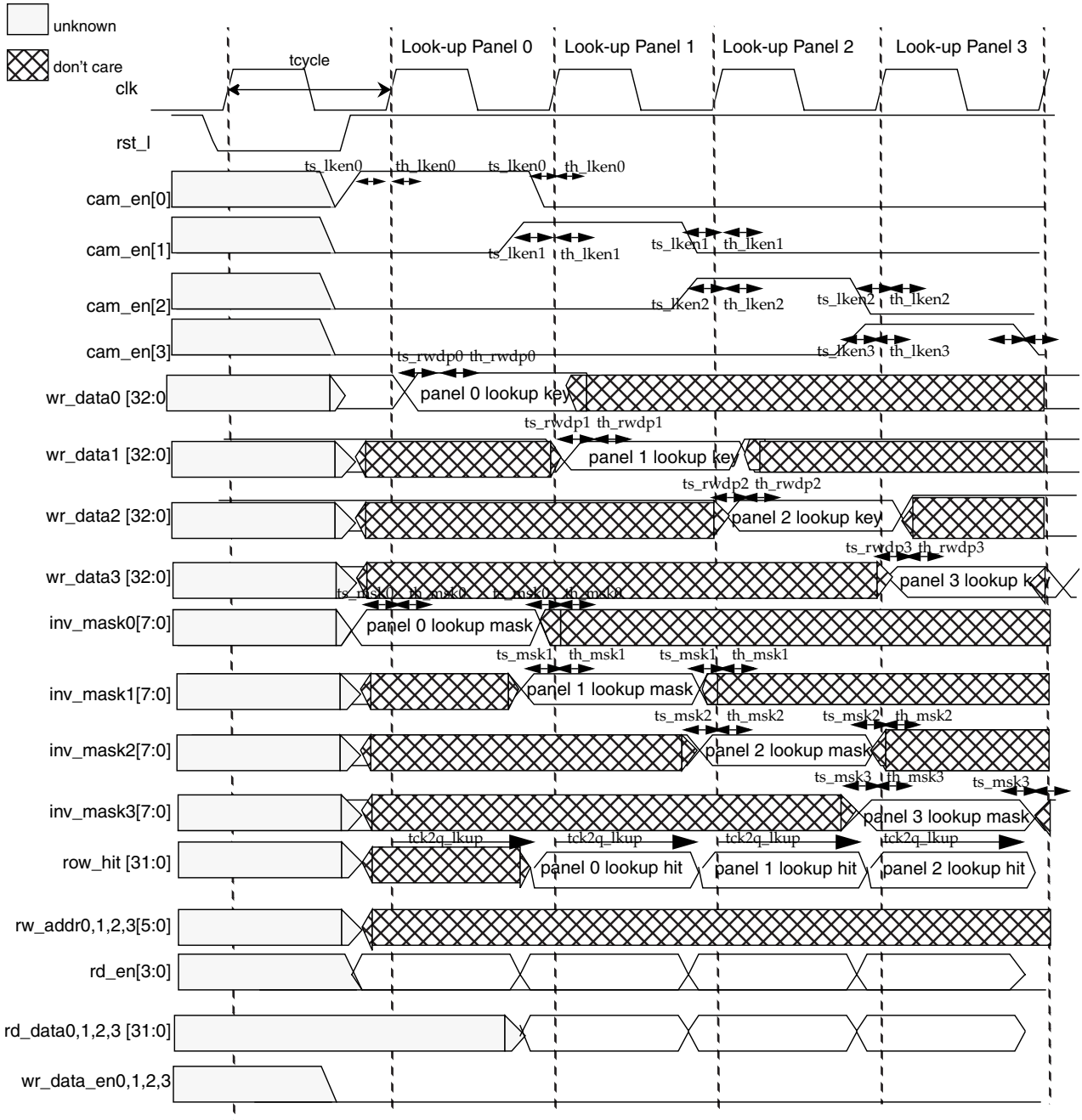

FIGURE 4-3 Panel Look-up Operation Timing Sequence

CHAPTER 5

e-Fuse Array

This chapter describes the following topics:

5.1 Functional Description of the EFA

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).

5.2 EFA Interfaces

Fuse controller interfaces are shown in the following figure.

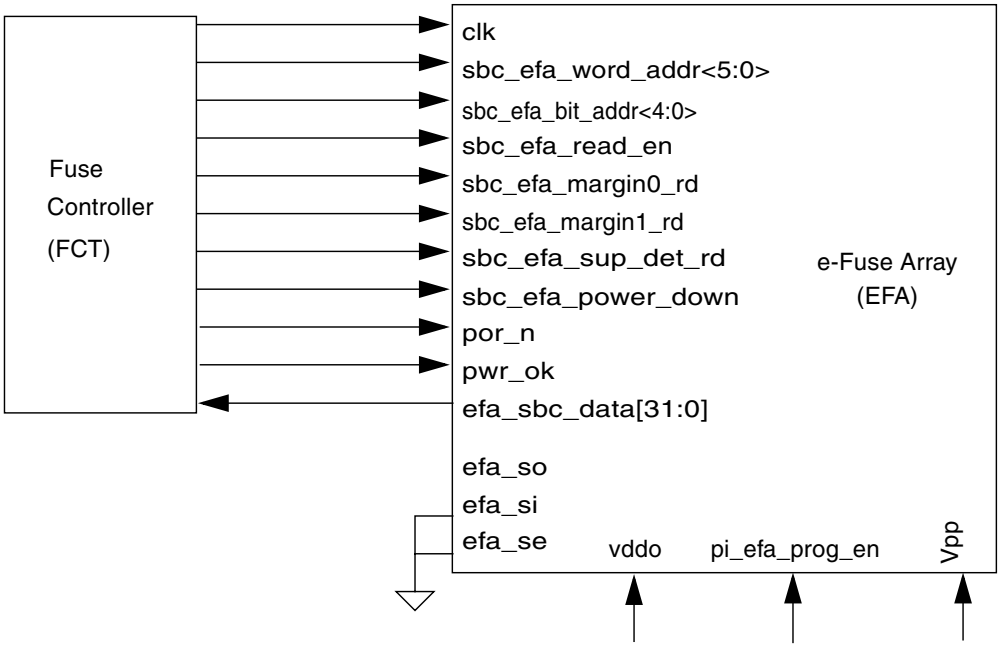

FIGURE 5-1 EFA Interfaces

5.3 I/O List

The following table describes electronic fuse array I/O signals.

TABLE 5-1 EFA I/O Signal List

Signal Name	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	
sbc_efa_margin1_rd	Input	
efa_sbc_data<31:0>	Output	Data out from EFA.
sbc_efa_power_down	Input	
sbc_efa_sup_det_rd	Input	
por_n	Input	Soft reset (NO-OP).
pwr_ok	Input	Hard reset. Clears all DataOut flops to ground.
clk	Input	Clock (l2clk).
efa_se	Input	Scan enable.
efa_si	Input	Scan in.
efa_so	Output	Scan out.
vpp	Input	Programming voltage.
vddo	Input	I/O supply voltage for power detection.

CHAPTER 6

Floating-Point Register File

This chapter describes the following topics:

6.1 Functional Description of the FRF

The 8-Kbit floating-point register file (FRF) contains 128 entries of 64-bit doublewords. The FRF has the following characteristics:

- Each doubleword has 14 error correction code (ECC) bits associated with it, making an additional 1.75 Kbits of ECC stored in the array. The total size of the array is 9.75 Kbits. Bits [38:32] are the ECC bits for the lower word (data[31:0]). Bits [77:71] are the ECC bits for the upper word (data[70:39]).
- Single-ported SRAM with a logical array size of 128 entry x 78 bits.
- During a write cycle, the two write-enable bits decide if either or both of the two words (upper and lower) of the doubleword must be written into.
- During a read cycle, all 64 data bits and 14 ECC bits are read and sent to the output bus.
- Single-cycle read and write, both in phase 1.
- Data input and read data are flopped.
- 78-bit write data inputs are provided for a single-cycle write.
- A read cycle can immediately follow a write cycle, and vice versa.
- Read/write contention does not allow either read or write to be accomplished.
- Write is inhibited when rst_tri_en is enabled.

The following table describes floating-point register file modes of operation.

TABLE 6-1 FRF Modes of Operation

ren	wen	rdata[77:0]	Memory Operation
0	1	0	Write cycle. Performs write according to word-enable masks.
1	0	read data	Read cycle.
1	1	0	Neither read nor write occurs. No operation.
0	0	0	No operation.

The following table describes floating-point register file word-enable bit masks for writes.

TABLE 6-2 Word-Enable Masks for Writes

Word Enable Bit	Input Data Mask
ctl_frf_wen[0]	dp_frf_data[38:0]
ctl_frf_wen[1]	dp_frf_data[77:39]

6.2 Block Diagrams

This section provides a floating-point front-end unit (FFU) top-level logic block diagram, an FRF logic symbol diagram, and a functional block diagram.

6.2.1 FFU Top-Level Logic Block Diagram

The top-level floating-point front-end unit is presented in FIGURE 6-1.

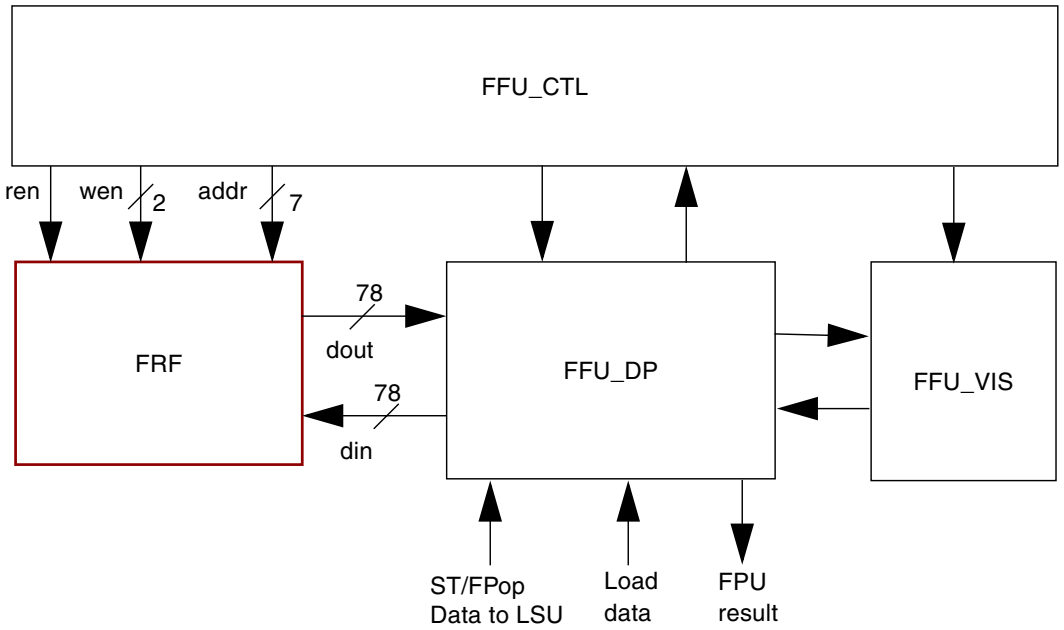

FIGURE 6-1 Top-Level Floating-Point Front-end Unit

6.2.2 FRF Logic Symbol

The floating-point front-end unit logic symbol is shown in the following figure.

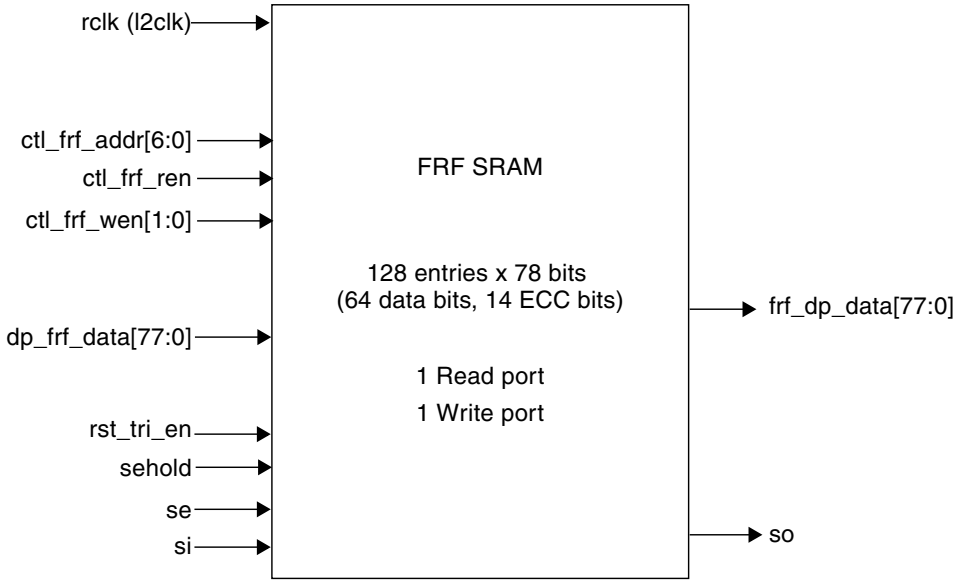

FIGURE 6-2 Floating-Point Register File Logic Symbol

6.2.3 Functional Block Diagram of FRF Array

The floating-point front-end unit functional block diagram is shown in the following figure.

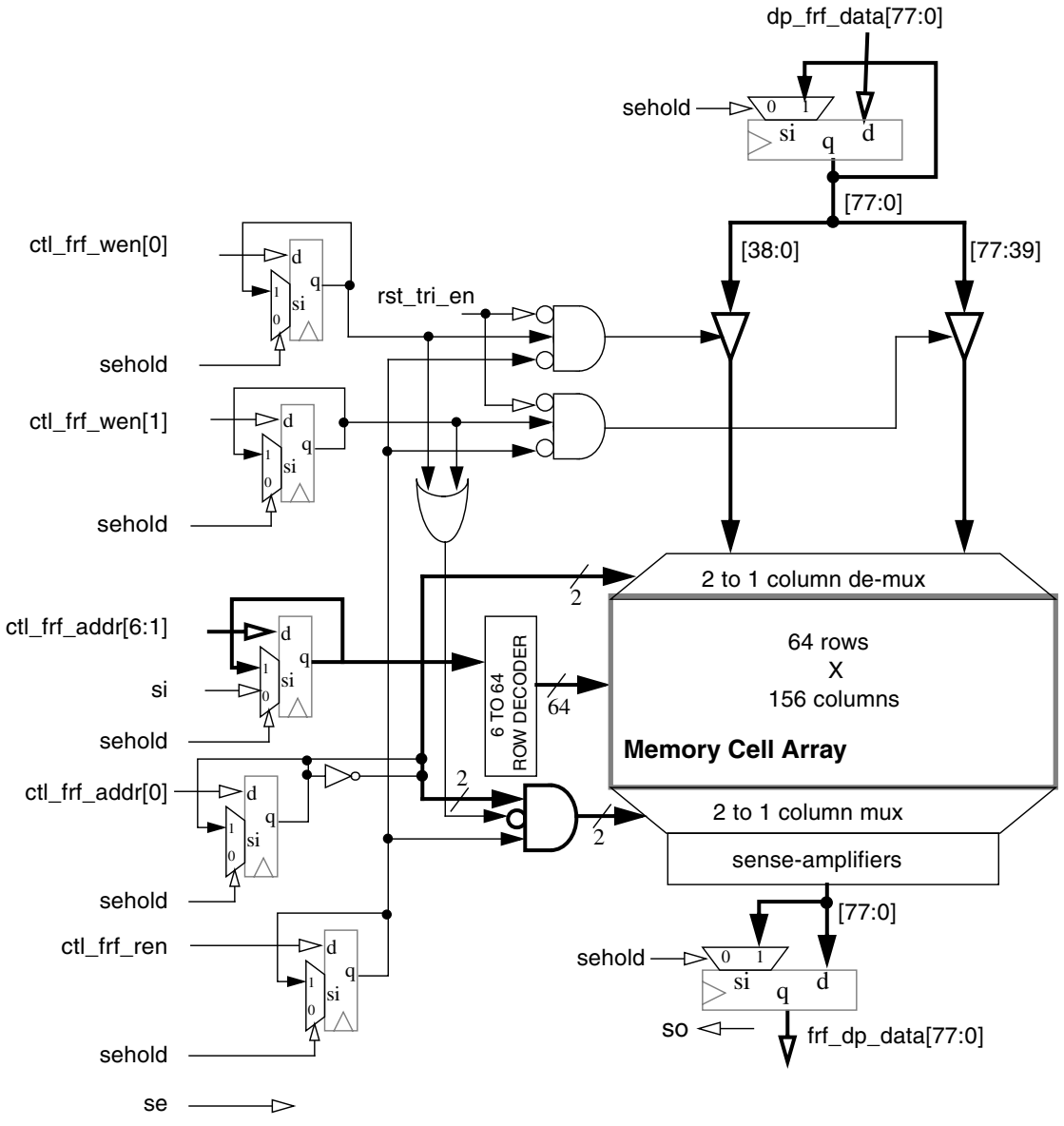

FIGURE 6-3 Functional Block Diagram of FRF Array

6.3 I/O List

The following table describes the floating-point front-end unit signals.

TABLE 6-3 FRF Array I/O Signal List

Signal Name*	TYPE	Source	Destination	Description
frf_dp_data[77:0]	Output	frf	ffu_dp	64-bit doubleword + 14-bit ECC output
so	Output	frf		Scan output
si	Input		frf	Scan input
se *	Input		frf	Scan enable
sehold *	Input		frf	Macro-test hold signal
rclk	Input		frf	L2 clock input
rst_tri_en *	Input		frf	Write-disable signal
ctl_frf_addr[6:0]	Input	ffu_ctl	frf	Address input
ctl_frf_wen[1:0]	Input	ffu_ctl	frf	Word select for write
ctl_frf_ren	Input	ffu_ctl	frf	Read enable
dp_frf_data[77:0]	Input	ffu_dp	frf	64-bit data + 14-bit ECC input

* Non-registered I/O is marked with an asterisk (*).

6.4 Timing Diagram

The following figure provides read/write I/O timing information.

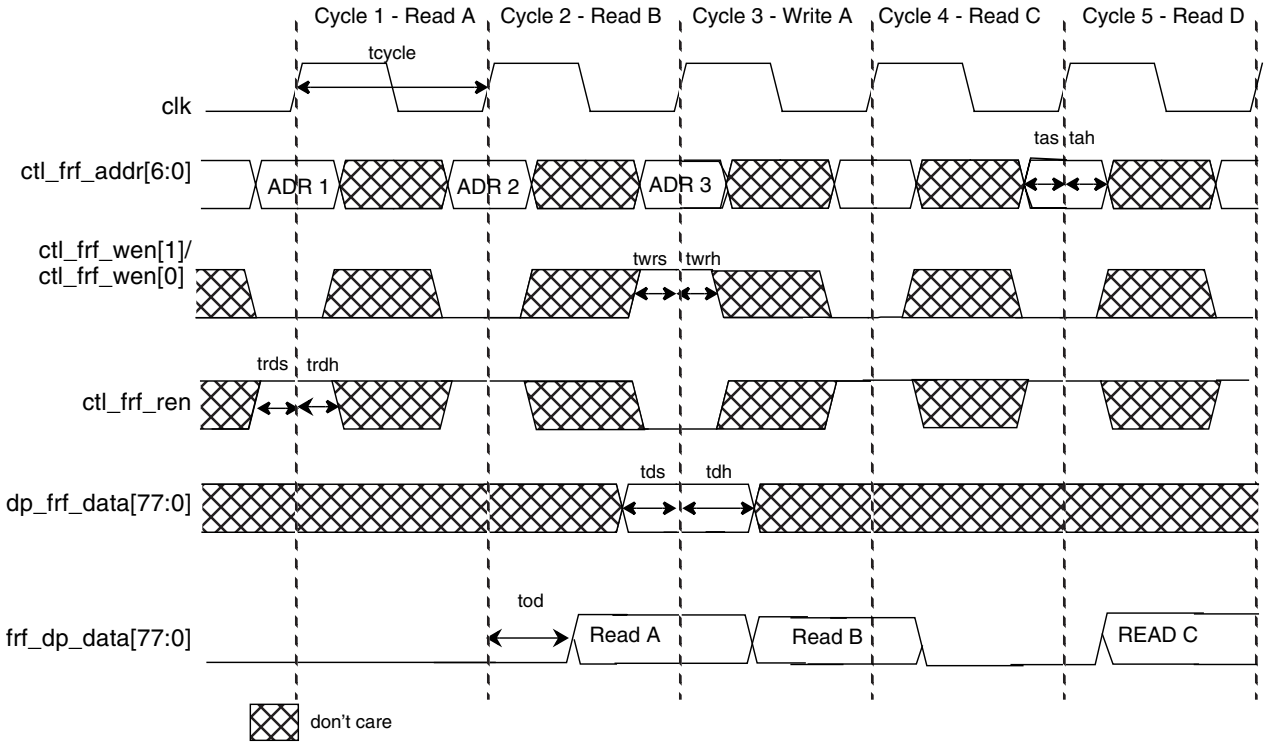

FIGURE 6-4 Read/Write Timing Diagram of FRF Array

CHAPTER 7

Instruction Cache Data

This chapter describes the following topics:

7.1 Functional Description of the ICD

The instruction cache data (ICD) is a 16-Kbyte, 32-byte line size, 4-way set associative instruction cache. The instruction cache data has the following characteristics:

- The instruction cache array is a single-ported SRAM with the configuration of 4 ways x 128 entries x 272 bits (including 16 parity bits).
- Total size of the instruction array is 17 Kbytes, with 1-Kbyte parity bits.
- Each cache line is contained in one bank. Each bank stores eight 32-bit instructions and two parity bits per instruction to make a total of 272 bits.
- During a read access, the upper seven bits of the 10-bit address are decoded to select one of 128 entries. The lower three bits are used to select one instruction out of the eight within a cache line as shown in TABLE 7-1.

  All four banks are read simultaneously.
- 136-bit write data inputs are provided for a single-cycle write into all four banks. Write way selects only one of four ways in a cycle to write.
- During a write cycle, four word-enable signals are provided for each of the four instructions in big-endian.

worden[0] -> wr_data[135:102] .. worden[3] -> wr_data[33:0]

- A read cycle can immediately follow a write cycle.
- Write is inhibited using the rst_tri_en signal during a scan operation.
- All the inputs are buffered and inverted at the boundary to reduce wire caps.
- All the inputs except rst_tri_en, se, and fuse-related signals are latched at each array.
- All the input/output latches and flops, except redundancy registers, are scannable elements.
- ICD supports macro test.

TABLE 7-1 Word Selection From an Instruction Array Cache Line (Read Access)

fdp_icd_index_bf[4:2]	fetdata[33:0]	topdata[33:0]
000	I0	I1
001	I1	I1
010	I2	I3
011	I3	I3
100	I4	I5
101	I5	I5
110	I6	I7
111	I7	I7

Note – topdata always reads out instructions 1, 3, 5, and 7 regardless of the value of fdp_icd_index_bf[2].

7.2 Block Diagram

A functional block diagram of the ICD array is shown in the following figure.

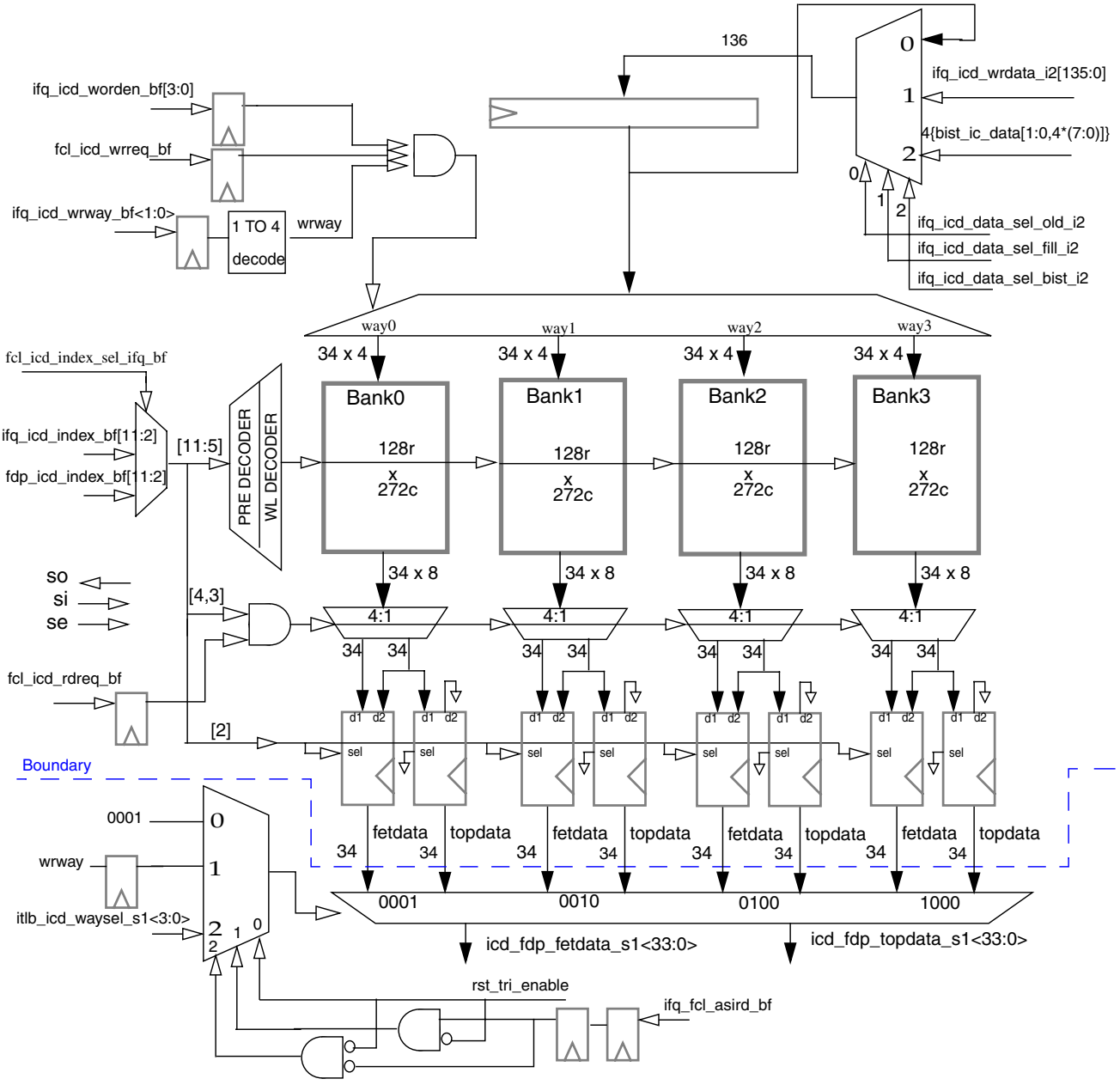

FIGURE 7-1 Functional Diagram of the ICD Array

7.3 ICD Logic Symbol

The instruction cache data logic symbol is shown in the following figure.

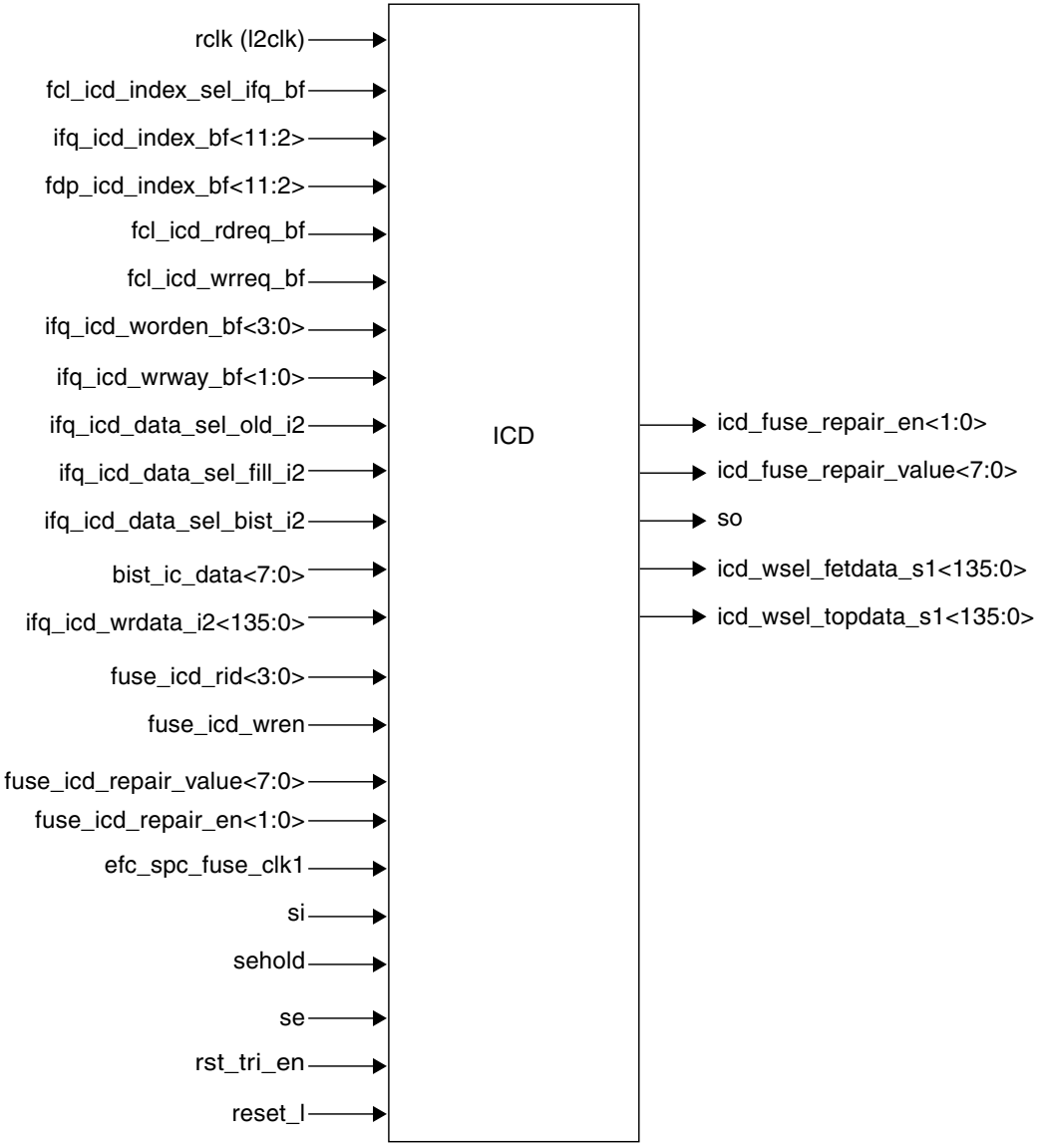

FIGURE 7-2 ICD Logic Symbol With Signals

7.4 I/O List

The following table describes the ICD I/O signals.

TABLE 7-2 Functional ICD Input/Output Signal List

Signal Name	Type	Source/Destination	Description
rclk	Input	CLOCK GRID	Clock input to header
fdp_icd_index_bf[11:2]	Input	FDP	Address input for instruction fetch
ifq_icd_index_bf[11:2]	Input	IFQDP	Alternate input address
fcl_icd_index_sel_ifq_bf*	Input	FCL	Input address selector
ifq_icd_wrway_bf[1:0]	Input	IFQDP	Way select for write data
ifq_icd_worden_bf[3:0]	Input	IFQCTL	Write word select
ifq_icd_wrdata_i2[135:0]	Input	IFQDP	128-bit data + 8-bit parity input
fcl_icd_rdreq_bf	Input	FCL	Cache read enable
fcl_icd_wrreq_bf	Input	FCL	Cache write enable
bist_ic_data[7:0]	Input	MBIST	BIST write data
ifq_icd_data_sel_bist_i2*	Input	IFQCTL	BIST write data selector
ifq_icd_data_sel_fill_i2*	Input	IFQCTL	Normal input write data selector
ifq_icd_data_sel_old_i2*	Input	IFQCTL	Write data from previous cycle selector
icd_fdp_topdata_s1[33:0] [W3:W0]	Output	WSELDP	Output instruction (I1,I3,I5,I7)
icd_fdp_fetdata_s1[33:0] [W3:W0]	Output	WSELDP	Output instruction (I1 to I7)
so	Output	TBD	Scan output
si	Input	TBD	Scan input
se*	Input	TEST_STUB	Scan enable
sehold*	Input	TEST_STUB	Scan hold input
rst_tri_en*	Input	TEST_STUB	Gates the write operation during scan
reset_l*	Input	TEST_STUB	Reset for redundancy registers
efc_spc_fuse_clk1	Input	CTU	
fuse_icd_rid[3:0]*	Input	REDHDR	Redundancy register ID. [3,2] determines banks, [1,0] determines row/column redundancy

TABLE 7-2 Functional ICD Input/Output Signal List *(Continued)*

Signal Name	Type	Source/Destination	Description
fuse_icd_wren*	Input	REDHDR	Redundancy register write enable
fuse_icd_repair_en[1:0]	Input	REDHDR	Enable bits to turn redundancy
fuse_icd_repair_value[7:0]	Input	REDHDR	Data to be stored in redundancy registers
icd_fuse_repair_en[1:0]	Output	REDHDR	Read data out from repair-enable registers
icd_fuse_repair_value[7:0]	Output	REDHDR	Read data out from redundancy registers

*Signals marked with an asterisk (*) are not latched or flopped inside ICD.

7.5 Functional Modes

The following tables describe the ICD functional modes of cache operation, write way select controls, and write word enables.

TABLE 7-3 ICD Functional Modes

rdreq	wrreq	rst_tri_enable	Cache Operation
0	0	X	No operation
1	0	X	Read cycle
0	1	H	Write cycle with data masking during scan test
0	1	L	Write cycle
1	1	L	Illegal operation
1	1	H	Read cycle with write masking

TABLE 7-4 Write Way Select Controls

ifq_icd_wrway_bf[1:0]	Way Select
00	0
01	1
10	2
11	3

TABLE 7-5 Write Word Enable

Word-Enable Bit	Input Data Enable
ifq_icd_worden_bf[0]	ifq_icd_wrdata_i2[135:102]
ifq_icd_worden_bf[1]	ifq_icd_wrdata_i2[101:68]
ifq_icd_worden_bf[2]	ifq_icd_wrdata_i2[67:34]
ifq_icd_worden_bf[3]	ifq_icd_wrdata_i2[33:0]

7.6 Timing Diagram

The following figure shows ICD I/O read/write timing.

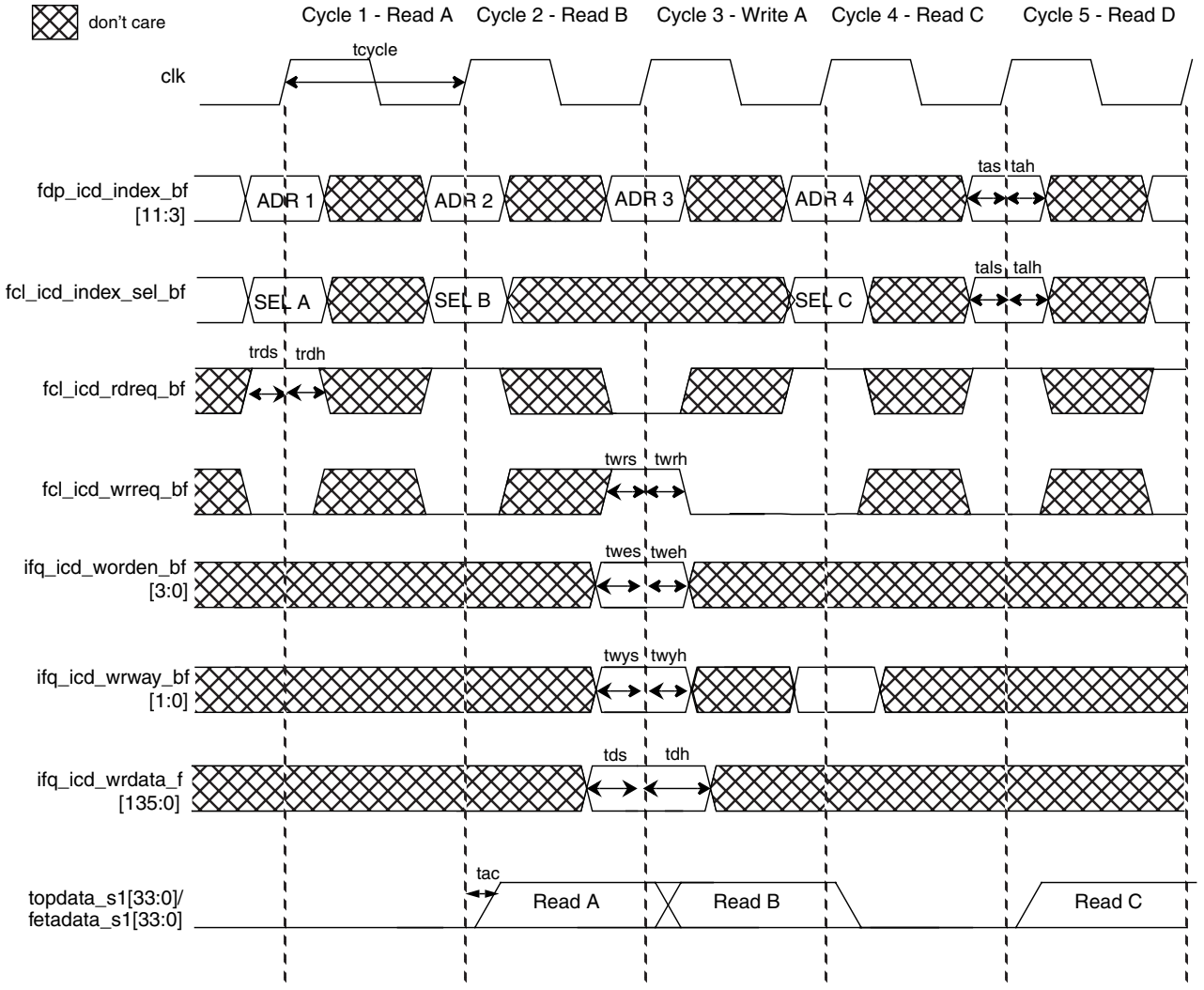

FIGURE 7-3 I/O Read/Write Timing Diagram of ICD

CHAPTER 8

Instruction and Data Cache Tag

This chapter describes the following topics:

8.1 Architectural Description

This section provides an architectural description of the instruction and data cache tag (IDCT), including a functional description and the memory behavior during read/write collision.

8.1.1 Functional Description of the IDCT

The OpenSPARC T1 processor uses a 128-entry, 4-way set-associative instruction and data cache. The actual tag field for instruction and data cache is 29 bits wide. Physically IDCT has 33 bits. This architecture consolidates IDCT and modular arithmetic memory (MAMEM) into a single design.The instruction and data cache tag has the following characteristics:

- IDCT contains tag bits for all four ways.
- There is one parity bit for 29 tag bits.
- Total array size is 16.5 Kbits (including all four ways).
- Address and control inputs are available in the stage before read/write, which is referred to as "_x" (rtl).
- Write data is available in the same stage as the write to the SRAM, which is referred to as "_y" (rtl).

- Write is triggered on the falling clock edge and is accomplished within the same cycle (self timed).
- Read data is also read out and available within a single-cycle "_y".
- Single-cycle read is initiated on the rising clock edge.
- During a read access, data is read from all four ways and is available to a 33*4 bit output data bus.
- Input data bus is 33 bits wide for each way. Write is only performed on the selected ways.
- During a write cycle, bitline equalization is deasserted for the selected column on the falling clock edge. This is done to prevent bitlines from discharging too much and ensures that the write margin is not compromised.
- Write data is set up to the falling clock edge and is neither latched nor flopped.
- Sense amplifier-enable is derived from the clock rising edge. Sense amplifier-enable delay control is provided through control pins adj[1:0]. One fast setting and two slow settings are provided. Operating frequency may need to be lowered for slower settings. See TABLE 8-2.
- Because the write to the SRAM is self timed, write self-time margin control is provided through control pins adj[3:2]. One fast setting and two slow settings are provided as shown in TABLE 8-3. Operating frequency must be lowered for slower settings.
- Read data is latched (not flopped) at the output and is held until just before the next valid read access.
- Memory contents and data on the output latch are preserved after recovering from a warm reset.
- At least one clock cycle is needed before reset is deasserted so that the SRAM is ready for normal operations. A setup time with regard to the positive clock edge should be satisfied to ensure that no memory corruption occurs while recovering from a warm reset.

8.1.2 Memory Behavior During Read/Write Collision

Because read is triggered by the rising clock edge, during a read/write collision a read would occur followed by a write which is triggered by the falling clock edge. But the write margin is compromised because one of the bitlines may have discharged in the wrong direction.

There is a crow-bar current path between the write driver and sense amplifier input precharge transistors through read and write column select. But this should not be an issue because the read column select is deasserted a bit earlier than the sense amplifier input precharge is asserted (16 ps at ttlh).

Bitline equalization (bleql/r<1:0>) does not assert during a read/write collision because the circuit bw_r_idct_bleq_en uses a flopped rdreq signal for internal control. However, the write recovery signal (wrrecl/r_l<1:0>) asserts to precharge and equalizes the bitlines.

During a read-write collision cycle the bitline precharge for the unselected column (determined by index<0>) remains deasserted until the clock low phase of the next Read (RD) or Write (WR) cycle. The bitlines remain in the floating state until the next RD or WR cycle. This means that the next read (after a collision case) would not guarantee any valid data at the output of IDCT. Also, if the next cycle (after a collision case) is a write cycle2, the data written during this cycle will also not be guaranteed. After this one RD or WR cycle, memory can resume normal operation.

TABLE 8-1 IDCT Modes of Operation

rdreq_x	wrreq_x	reset_l	rdtag_w[n]_y<32:0>	Memory Operation
1	1	1	Unknown	Read/write contention. Read data is guaranteed, but write is not guaranteed.
0	1	1	Last read data	Write cycle.
1	0	1	Read data	Read cycle.
X	X	0	Last read data	Reset.

Minimum “tcycle” calculation is based on 0 ps slack at 10% margin for frequency dependent paths inside IDCT.

8.2 Block Diagrams

Following are block diagrams of the instruction and data cache tag for the load and store unit (LSU), instruction fetch unit (IFU), and stream processing unit (SPU). Figures of the IDCT logic symbol and the IDCT array are also found in this section.

8.2.1 Top-Level Logic Block Diagrams

The following figure shows the LSU for the instruction and data cache tag.

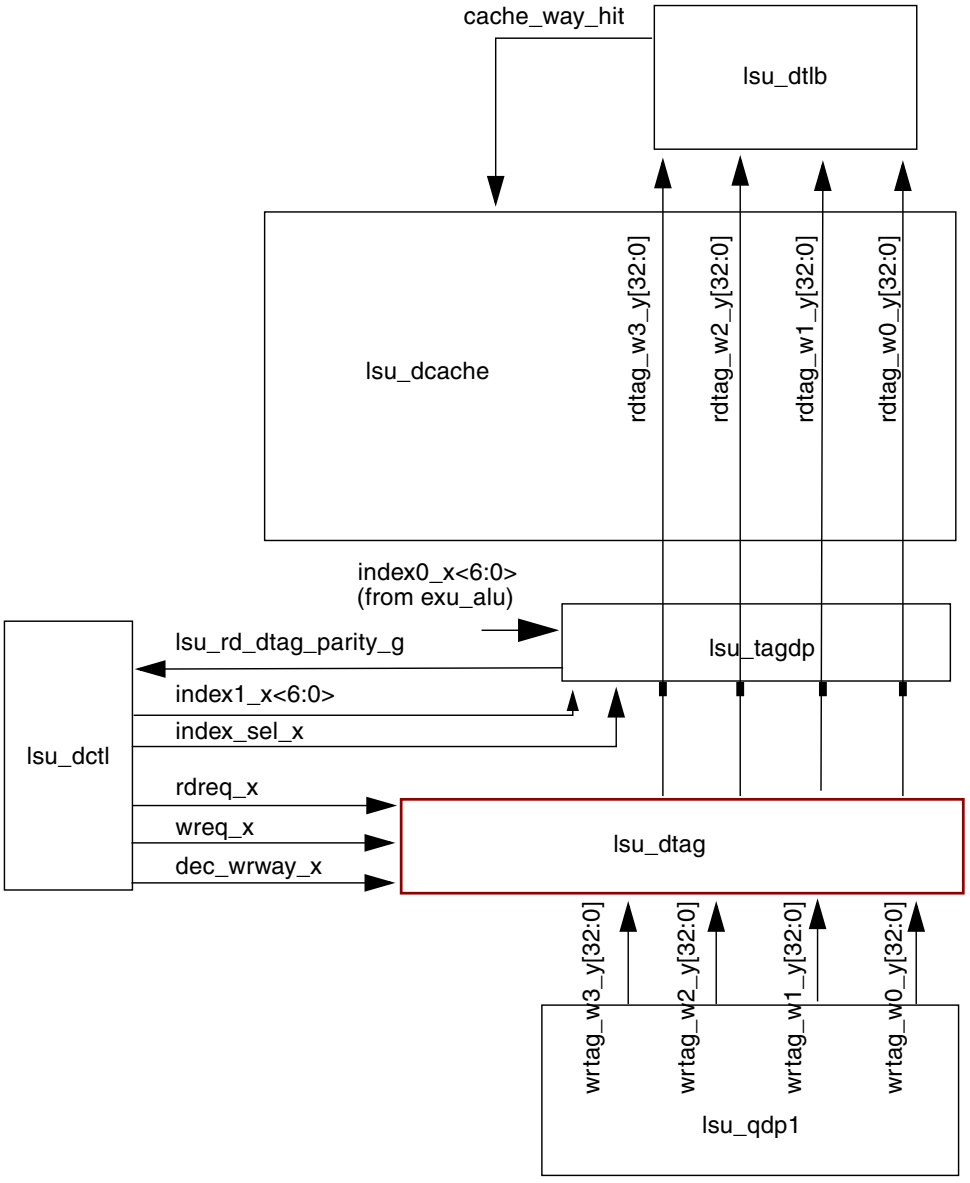

FIGURE 8-1 Load and Store Unit (LSU)

The following figure shows the IFU for the instruction and data cache tag.

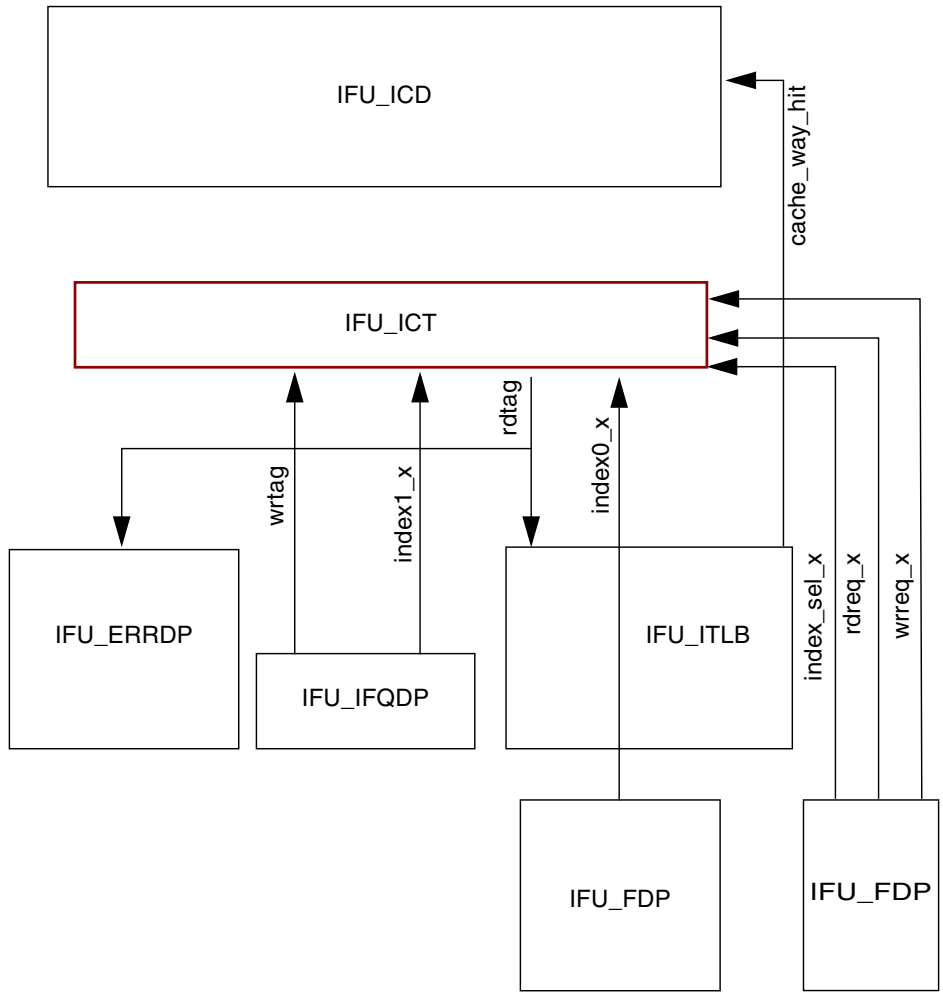

FIGURE 8-2 Instruction Fetch Unit (IFU)

The following figure shows the SPU for the instruction and data cache tag

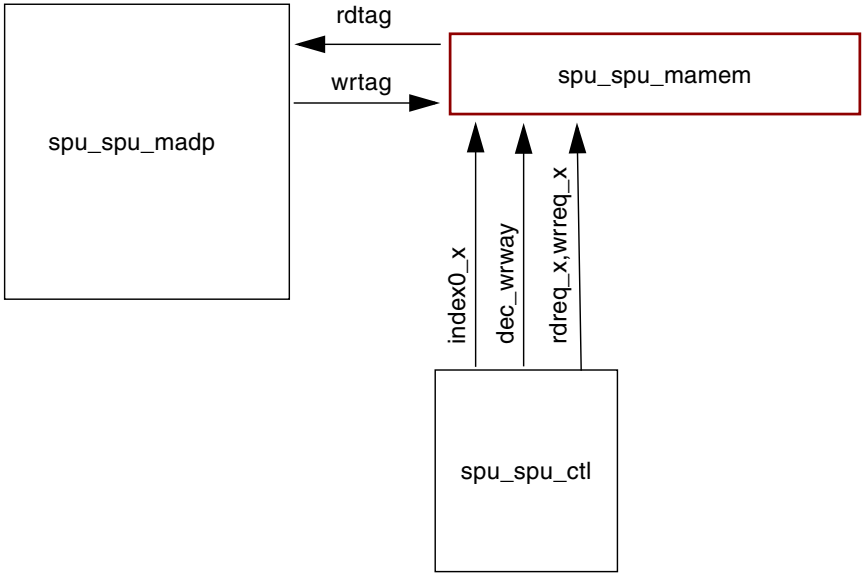

FIGURE 8-3 Stream Processing Unit (SPU)

8.2.2 IDCT Logic Symbol

The instruction and data cache tag logic symbol is shown in the following figure.

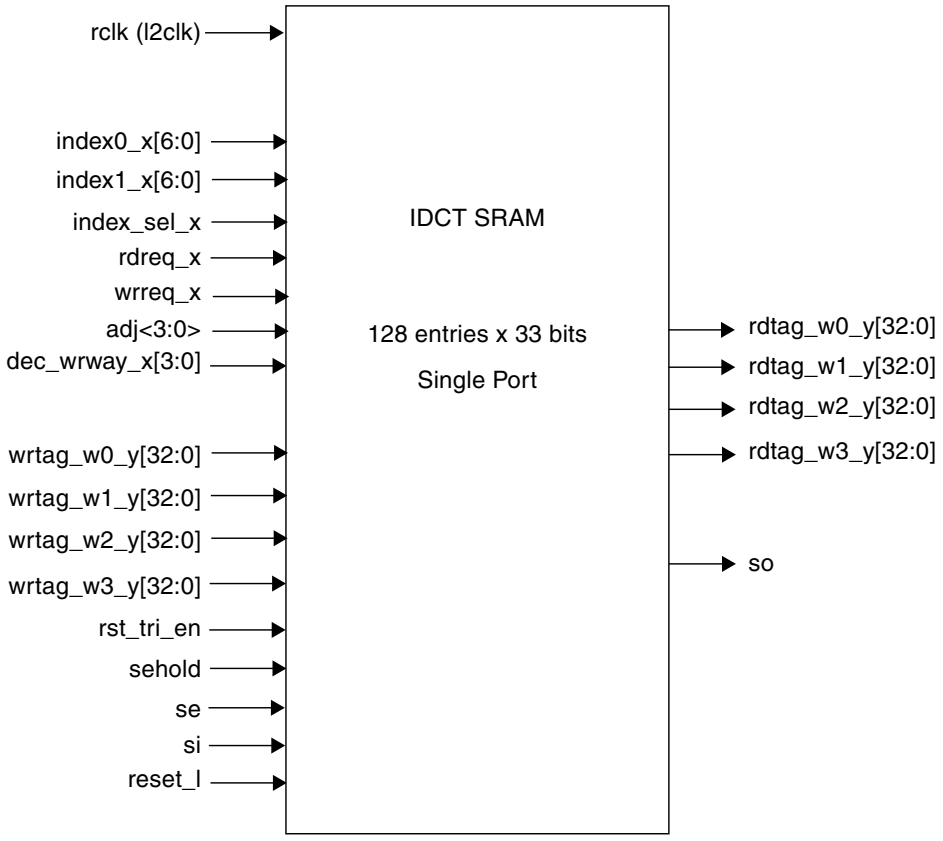

FIGURE 8-4 IDCT Logic Symbol

8.2.3 Functional Block Diagram of IDCT Array

A functional block diagram of the IDCT array is shown in the following figure.

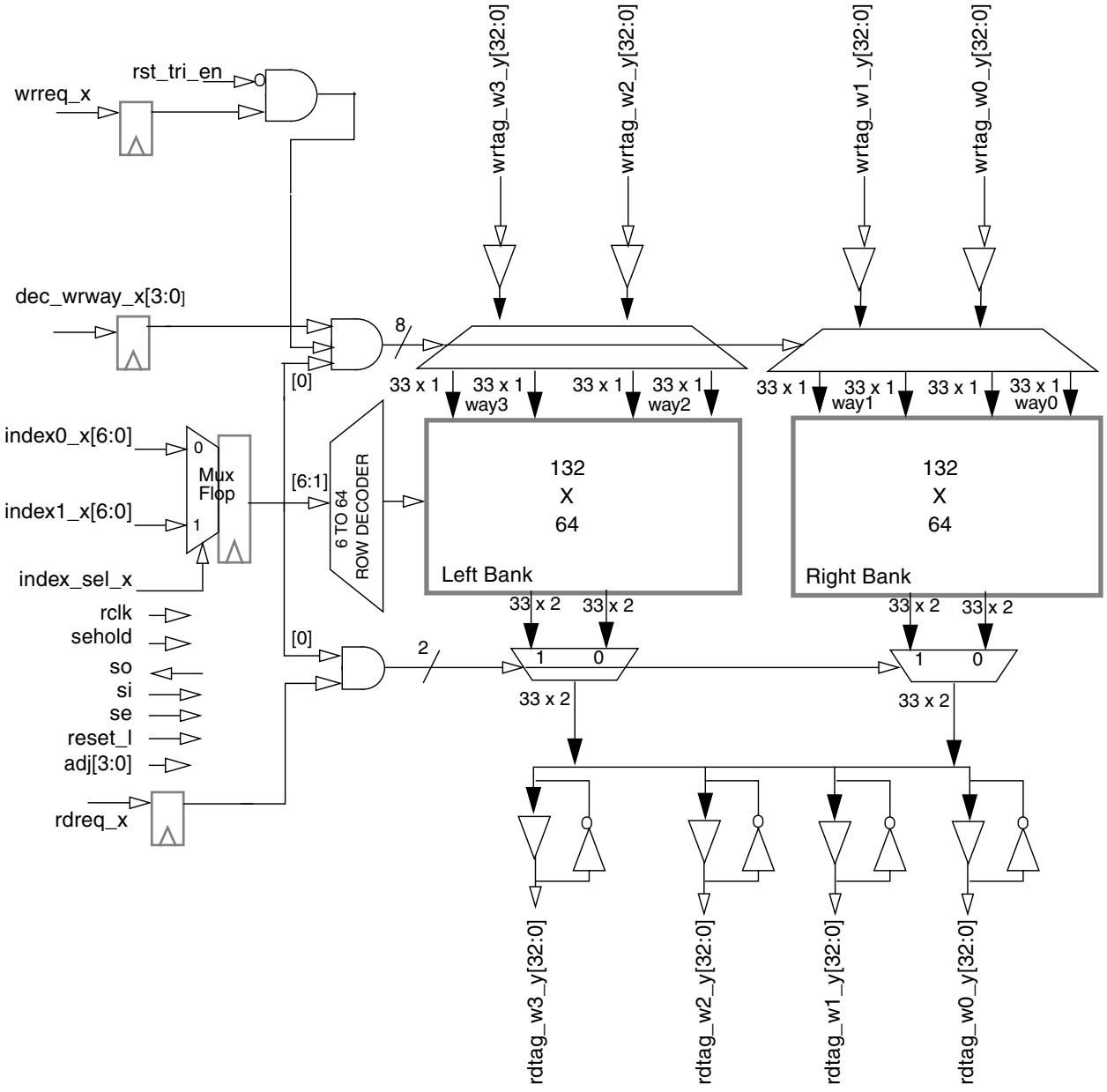

FIGURE 8-5 Functional Block Diagram of IDCT Array

8.3 I/O List

The following table describes the IDCT LSU I/O signals.

TABLE 8-2 IDCT Array I/O Signal List (LSU)

Signal Name*	Tyoe	Source	Destination	Description
rdtag_w0_y[32:0]*	Output		lsu_tagdp/tlb	29-bit read data from way 0 + 1-bit parity
rdtag_w1_y[32:0]*	Output		lsu_tagdp/tlb	29-bit read data from way 1 + 1-bit parity
rdtag_w2_y[32:0]*	Output		lsu_tagdp/tlb	29-bit read data from way 2 + 1-bit parity
rdtag_w3_y[32:0]*	Output		lsu_tagdp/tlb	29-bit read data from way 3 + 1-bit parity
so	Output		lsu_excpctl	Scan output
si	Input	bw_r_rf16x32		Scan input
se*	Input			Scan enable
sehold*	Input			Macro-test hold signal
rclk	Input			L2 clock input
reset_l*	Input	lsu_ctl		Reset signal
index0_x[6:0]	Input	exu		Address input
index1_x[6:0]	Input	lsu_dctl		Address input
index_sel_x*	Input	lsu_dctl		Address input mux selector
dec_wrway_x[3:0]	Input	lsu_dctl		Decoded write way selector
wrtag_w0_y[32:0]*	Input	lsu_qdp1		29-bit data + 1-bit parity input for way 0
wrtag_w1_y[32:0]*	Input	lsu_qdp1		29-bit data + 1-bit parity input for way 1
wrtag_w2_y[32:0]*	Input	lsu_qdp1		29-bit data + 1-bit parity input for way 2
wrtag_w3_y[32:0]*	Input	lsu_qdp1		29-bit data + 1-bit parity input for way 3
adj[3:0]*	Input	lsu_dctldp		Read/write self-time margin control
rst_tri_en*	Input			Write disable signal
rdreq_x	Input	lsu_dctl		Read enable
wrreq_x	Input	lsu_dctl		Write enable

* Non-registered I/O is marked with an asterisk (*).

lsu_dtag (IDCT) uses only 30 bits of rdtag_wn_y and wrtag_wn_y.

The following table describes the IDCT IFU I/O signals.

TABLE 8-3 IDCT Array I/O Signal List (IFU)

Signal Name*	Type	Source	Destination	Description
rdtag_w0_y[32:0]*	Output		ifu_errdp/tlb	29-bit read data from way 0 + 1-bit parity
rdtag_w1_y[32:0]*	Output		ifu_errdp/tlb	29-bit read data from way 1 + 1-bit parity
rdtag_w2_y[32:0]*	Output		ifu_errdp/tlb	29-bit read data from way 2 + 1-bit parity
rdtag_w3_y[32:0]*	Output		ifu_errdp/tlb	29-bit read data from way 3 + 1-bit parity
so	Output		bw_r_rf16x32	Scan output
si	Input	bw_r_tlb		Scan input
se*	Input	sparc_ifu_dec		Scan enable
sehold*	Input	sparc_ifu_fcl		Macro-test hold signal
rclk	Input			L2 clock input
reset_l*	Input	sparc_ifu_swl		Reset signal
index0_x[6:0]	Input	ifu_fdp		Address input
index1_x[6:0]	Input	ifu_ifqdp		Address input
index_sel_x*	Input	ifu_fcl		Address input mux selector
dec_wrway_x[3:0]	Input	ifu_invctl		Decoded write way selector
wrtag_w0_y[32:0]*	Input	ifu_ifqdp		29-bit data + 1-bit parity input for way 0
wrtag_w1_y[32:0]*	Input	ifu_ifqdp		29-bit data + 1-bit parity input for way 1
wrtag_w2_y[32:0]*	Input	ifu_ifqdp		29-bit data + 1-bit parity input for way 2
wrtag_w3_y[32:0]*	Input	ifu_ifqdp		29-bit data + 1-bit parity input for way 3
adj[3:0]*	Input	lsu_dctldp		Read/write self time margin control
rst_tri_en*	Input			Write disable signal
rdreq_x	Input	ifu_fcl		Read enable
wrreq_x	Input	ifu_fcl		Write enable

* Non-registered I/O is marked an asterisk (*).

ifu_ict (IDCT) uses only 30 bits of rdtag_wn_y and wrtag_wn_y.

The following table describes the IDCT SPU I/O signals.

TABLE 8-4 IDCT Array I/O Signal List (SPU)

Signal Name*	Type	Source	Destination	Description
rdtag_w0_y[32:0]*	Output		spu_madp	32-bit read data from way 0
rdtag_w1_y[32:0]*	Output		spu_madp	32-bit read data from way 1
rdtag_w2_y[32:0]*	Output		spu_madp	32-bit read data from way 2
rdtag_w3_y[32:0]*	Output		spu_madp	32-bit read data from way 3
so	Output		sparc_ffu	Scan output
si	Input	tlu		Scan input
se*	Input	test_stub		Scan enable
sehold*	Input	test_stub		Macro-test hold signal
rclk	Input			L2 clock input
reset_l*	Input			Reset signal
index0_x[6:0]	Input	spu_ctl		Address input
index1_x[6:0]	Input	unused		Address input
index_sel_x*	Input	gnd		Address input mux selector
dec_wrway_x[3:0]	Input	spu_ctl		Decoded write way selector
wrtag_w0_y[32:0]*	Input	spu_madp		32-bit data input for way 0
wrtag_w1_y[32:0]*	Input	spu_madp		32-bit data for way 1
wrtag_w2_y[32:0]*	Input	spu_madp		32-bit data for way 2
wrtag_w3_y[32:0]*	Input	spu_madp		32-bit data for way 3
adj[3:0]*	Input			Read/write self-time margin control
rst_tri_en*	Input	test_stub		Write disable signal
rdreq_x	Input	spu_ctl		Read enable
wrreq_x	Input	spu_ctl		Write enable

* Non-registered I/O is marked with an asterisk (*).

spu_spu_mamem uses full 33 bits of rdtag_wn_y and wrtag_wn_y.

8.4 Timing Diagram

Following is a read/write timing diagram for the instruction and data cache tag.

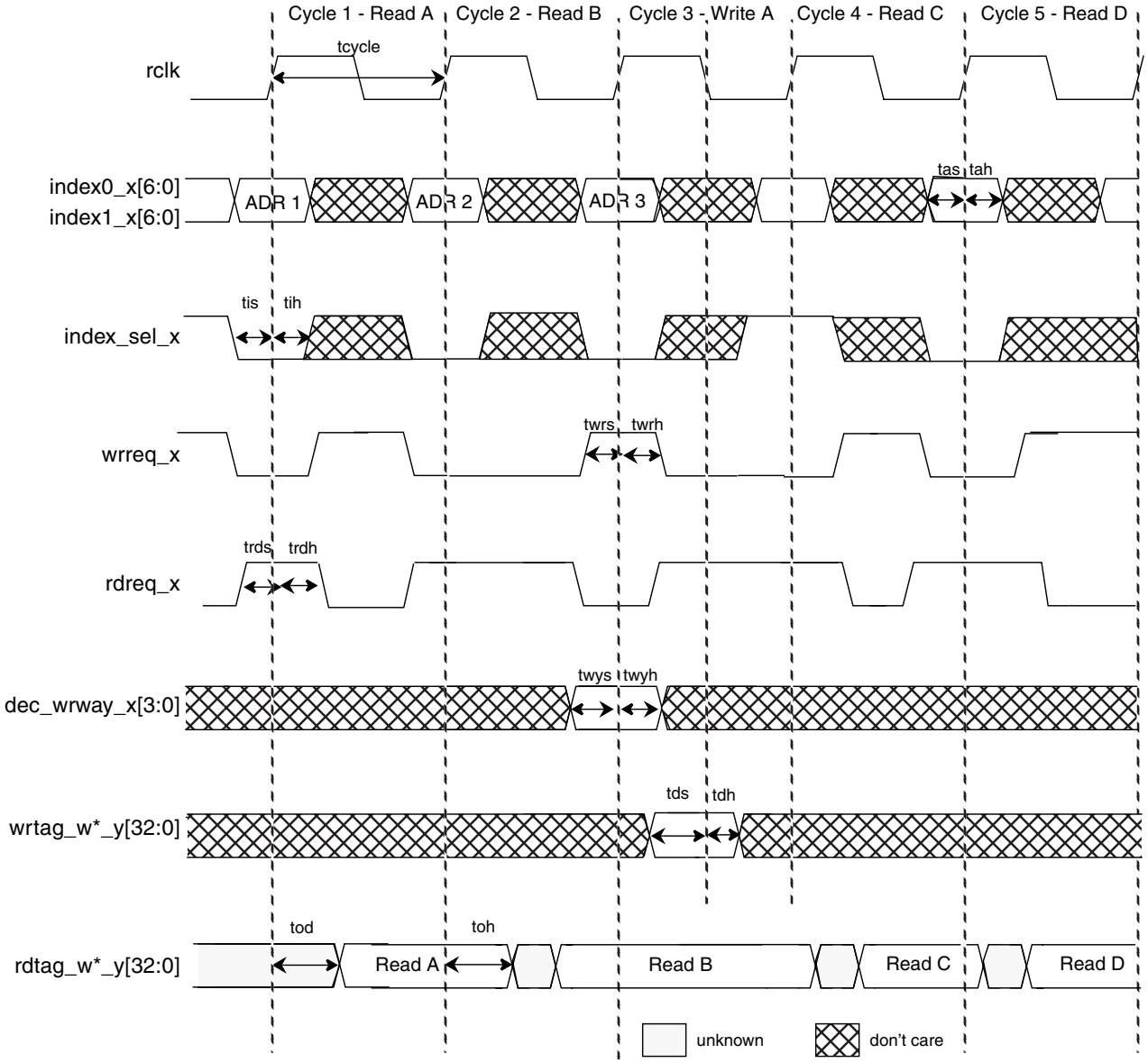

FIGURE 8-6 Read/Write Timing Diagram of IDCT Array

CHAPTER 9

Integer Register File

This chapter describes the following topics:

9.1 Functional Description of the IRF

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 9-1. 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.

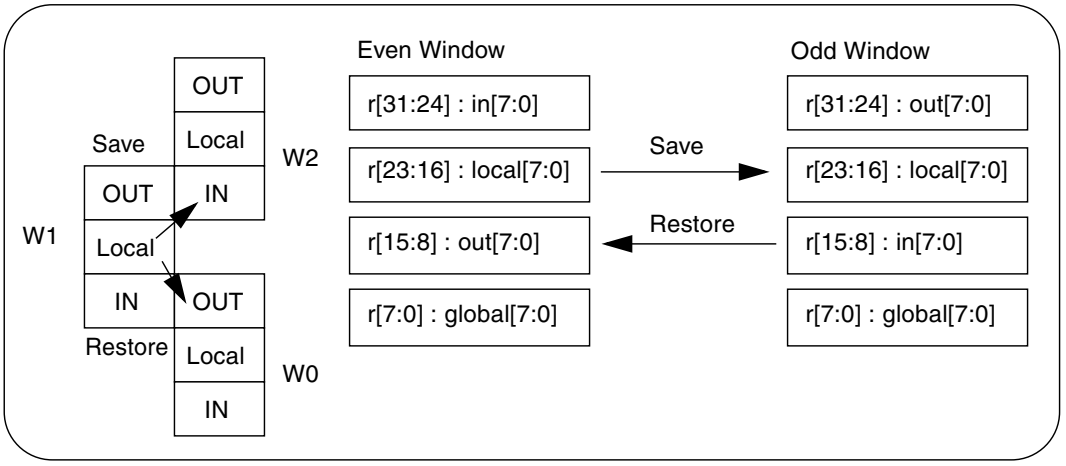

FIGURE 9-1 Register File Window Structure

9.1.1 Functional Summary

Integer register file functions include the following:

- 32-entry x 72-bit structure (16x144 physically), replicated four times for each thread.
- Three read ports and two write ports.
- All three reads occur to the same thread. Read is single ended.
- Writes may be to different threads. Write is differential.
- 32 entries are split into 16 I/O registers – eight local registers and eight global registers.
- The register file supports eight windows per thread.
- All the threads work independently. Save to one thread and write or read to another thread can occur. Similarly, restore to one thread and read to another thread can occur at the same time.

- The same thread cannot have multiple operations at the same time to the same address. For example, read and write cannot occur at the same time to the same thread.

9.2 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 9-1 IRF I/O Signal List

Signal Name	Type	Flop/ Latch	Description
ifu_exu_tid_s2 [1:0]	Input	Flop	S stage thread ID for read.
ifu_exu_rs1_s [4:0]	Input	Flop	Read port 1 address.
ifu_exu_rs2_s [4:0]	Input	Flop	Read port 2 address.
ifu_exu_rs3_s [4:0]	Input	Flop	Read port 3 address.
ifu_exu_ren1_s	Input	Flop	Read enable for port 1.
ifu_exu_ren2_s	Input	Flop	Read enable for port 2.
ifu_exu_ren3_s	Input	Flop	Read enable for port 3.
ecl_irf_wen_w	Input	Latch	Write enable for write port 1.
ecl_irf_wen_w2	Input	Latch	Write enable for write port 2.
ecl_irf_rd_m [4:0]	Input	Flop	Write port 1 address.
ecl_irf_rd_g [4:0]	Input	Flop	Write port 2 address.
byp_irf_rd_data_w [71:0]	Input	Latch	Write data for port 1.
byp_irf_rd_data_w2 [71:0]	Input	Latch	Write data for port 2.
ecl_irf_tid_m [1:0]	Input	Flop	W stage thread ID for write port 1.

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

Signal Name	Type	Flop/ Latch	Description
ecl_irf_tid_g [1:0]	Input	Flop	W2 stage thread ID for write port 2.
rml_irf_old_lo_cwp_e [2:0]	Input	Flop	Current window pointer for locals and odds.
rml_irf_new_lo_cwp_e[2:0]	Input	Flop	Target window pointer for locals and odds.
rml_irf_old_e_cwp_e [2:1]	Input	Flop	Current window pointer for evens.
rml_irf_new_e_cwp_e [2:1]	Input	Flop	Target window pointer for evens.
rml_irf_swap_even_e	Input	Flop	Swap control for even registers.
rml_irf_swap_odd_e	Input	Flop	Swap control for odd registers.
rml_irf_swap_local_e	Input	Flop	Swap control for local registers.
rml_irf_kill_restore_w	Input	Flop	Cancellation of RESTORE (SWAP) transition. Does not affect globals.
rml_irf_cwpswap_tid_e [1:0]	Input	Flop	Thread ID for swap of local, evens and odds.
rml_irf_old_agp [1:0]	Input	Flop	Alternate global pointer for the set of globals in thread.
rml_irf_new_agp [1:0]	Input	Flop	Alternate global pointer for the set of globals in thread.
rml_irf_swap_global	Input	Flop	Swap control for global registers.
rml_irf_global_tid [1:0]	Input	Flop	Thread ID for global registers.
irf_byp_rs1_data_d_l [71:0]	Output	-	Register file outputs for read port 1.
irf_byp_rs2_data_d_l [71:0]	Output	-	Register file outputs for read port 2.
irf_byp_rs3_data_d_l [71:0]	Output	-	Register file outputs for read port 3.
irf_byp_rs3h_data_d_l [31:0]	Output	-	Register file outputs for read port 4.
rclk	Ground	-	Clock signal to this register file.
si	Input	-	Scan in.
so	Output	-	Scan out.
se	Input	-	Scan enable.
reset_l	Ground	-	Global reset.
sehold	Input	-	Holds scan data for one more cycle when se is on.
rst_tri_en	Ground	-	Reset to block write/swap during scan.

9.3 Block Diagrams

The following figures show the logic symbol and top-level block diagram of the IRF.

The OpenSPARC T1 processor has eight SPARC® cores. IRF is part of each SPARC core.

9.3.1 Logic Symbol of the SPARC IRF

The integer register file logic symbol is shown in the following figure.

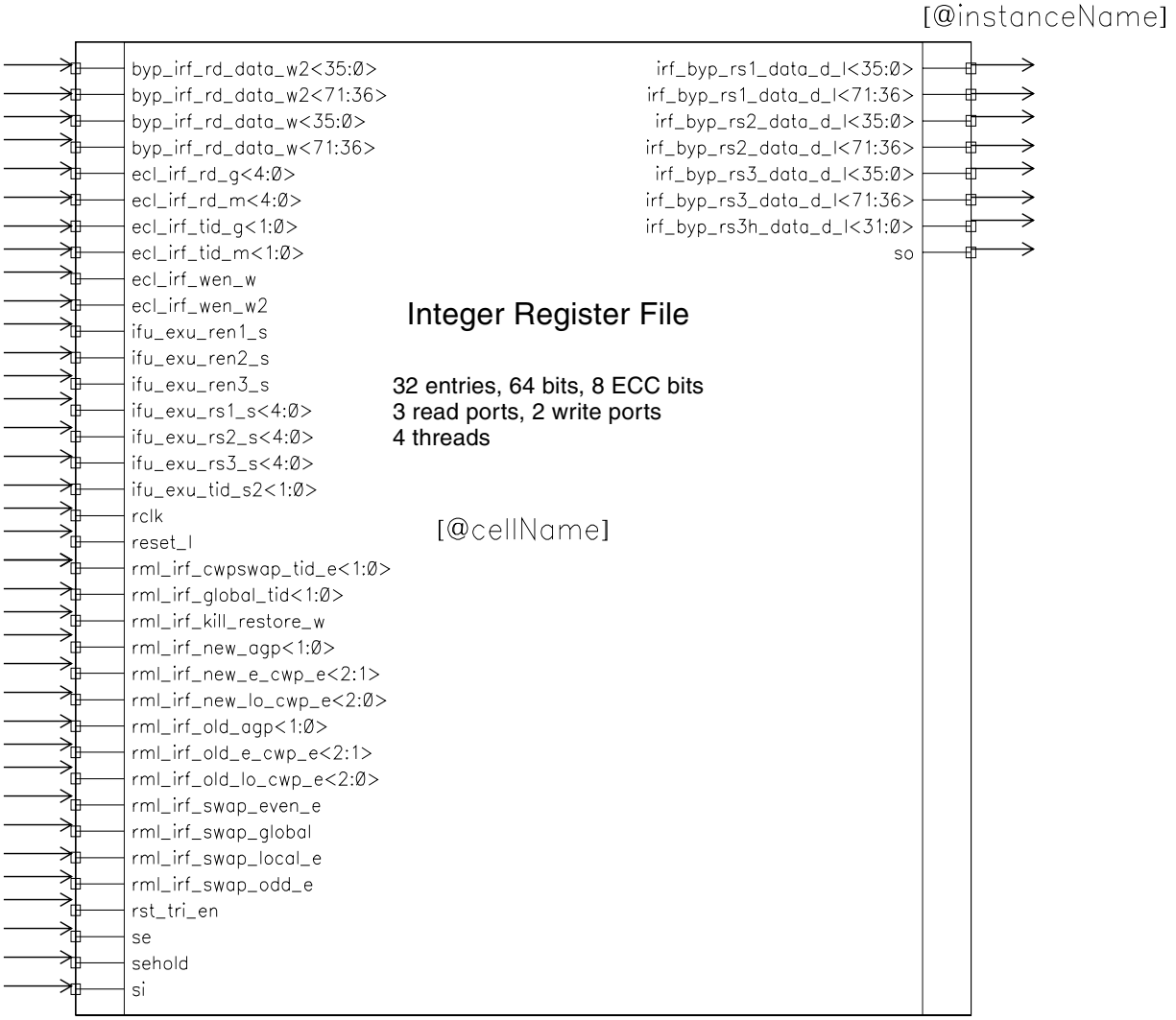

FIGURE 9-2 Logic Symbol for the SPARC Processor Integer Register File (IRF)

9.3.2 SPARC IRF Top-Level Block Diagram

A top-level block diagram of the integer register file is shown in the following figure.

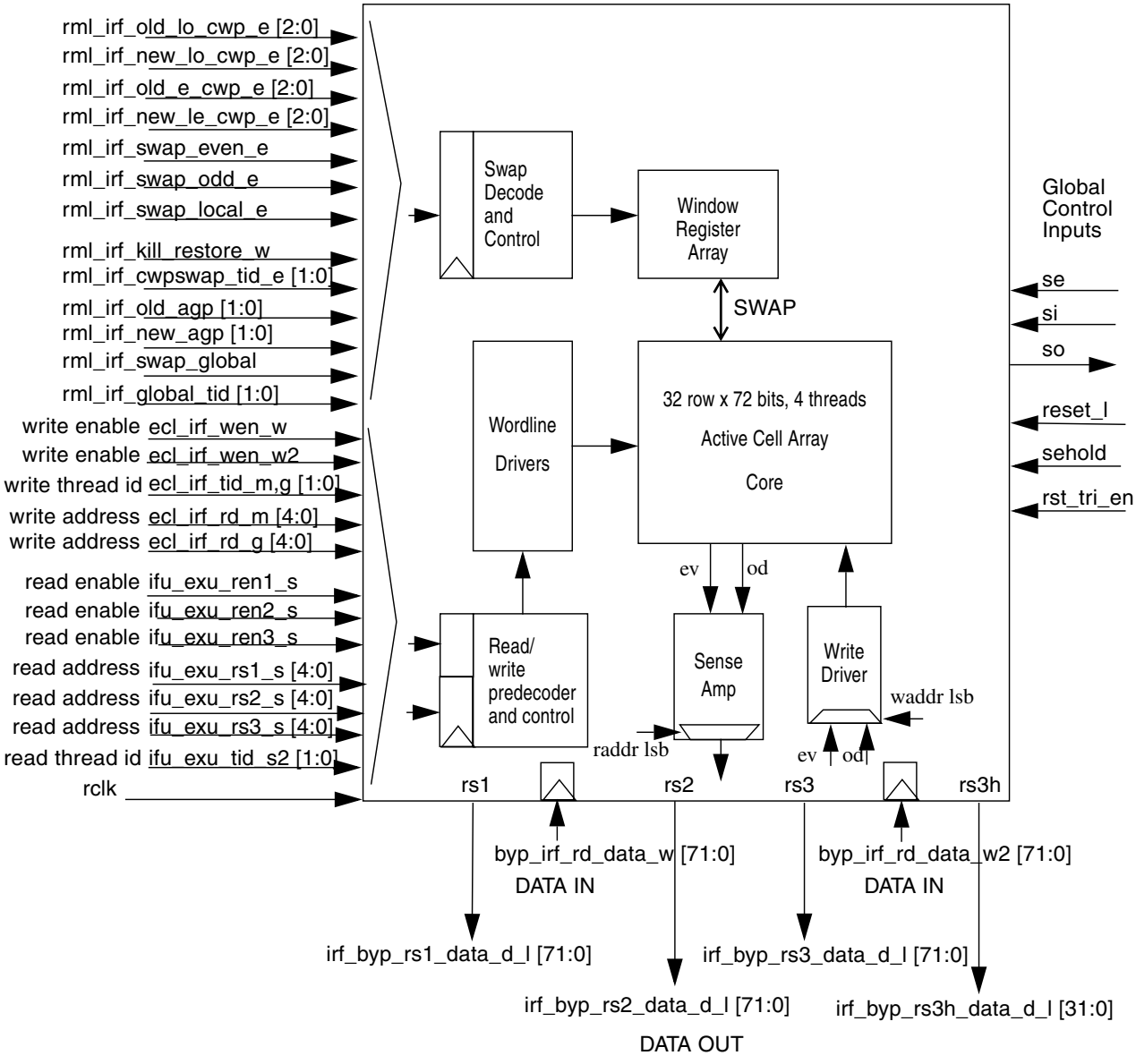

FIGURE 9-3 SPARC IRF Top-Level Block Diagram

9.4 Timing Diagram

FIGURE 9-4 shows the timing assumption of the IRF.

- 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.

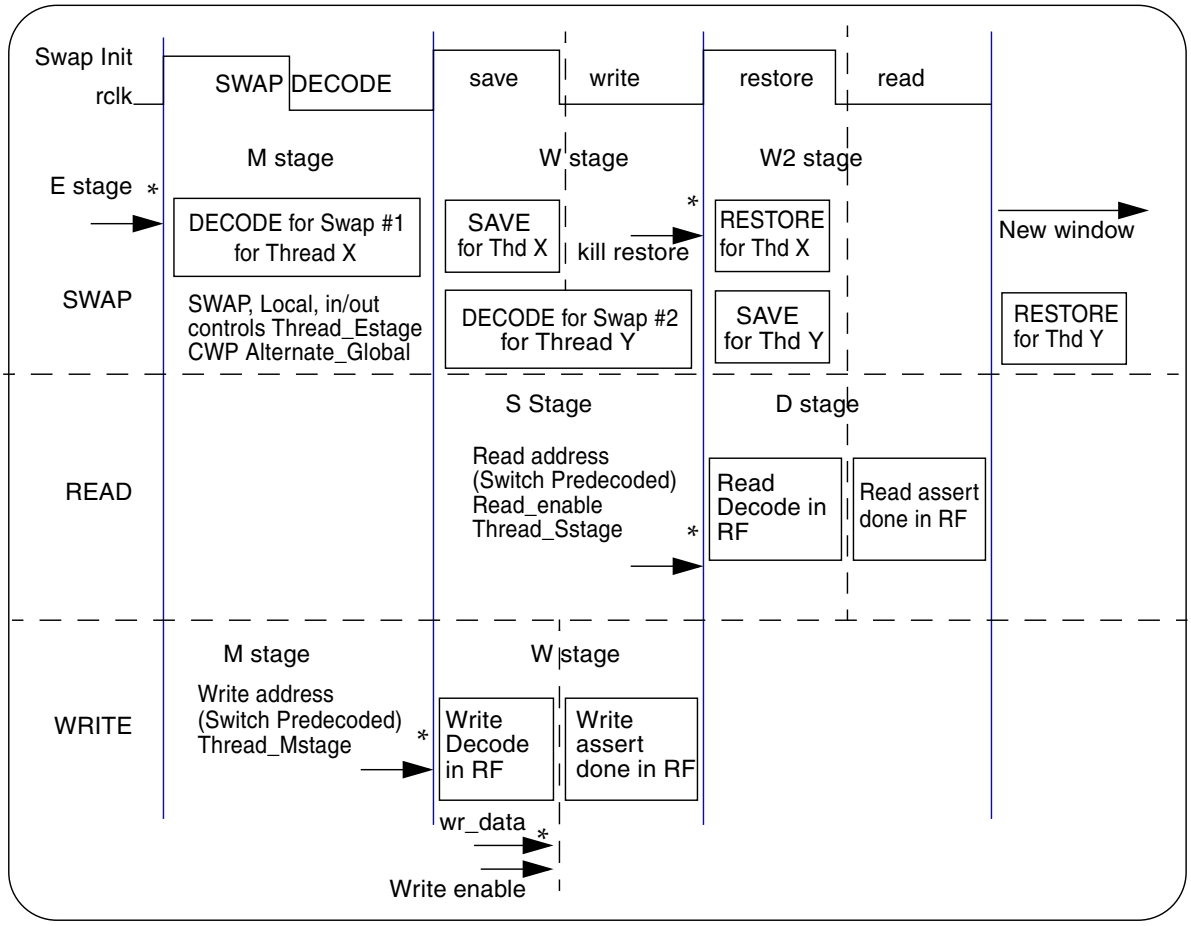

FIGURE 9-4 IRF Timing Diagram

CHAPTER 10

L2-cache Data Array

This chapter describes the following topics:

10.1 Functional Description

The OpenSPARC T1 processor Level 2 cache (L2-cache) data array is inclusive, 3 Mbytes in size, and composed of four symmetrical banks. The L2-cache has the following characteristics:

- Each L2 bank contains the following:
  - 1K entries, each entry is 12-way, set-associative, 64-byte cache line
  - L2 tag, L2 VUAD (valid, used, allocated, and dirty) bits, L2 data, input queue, output queue, directory, miss buffer, fill buffer, writeback buffer, and snoop response buffer
- More than one bank can be accessed at the same time.
- Each L2 data array bank is a single-ported structure that supports the following operations:
  - 16-byte and 64-byte read
  - 4-byte, 8-byte, and 64-byte write with a combination of any word enable
- Each L2 bank is further divided into four logical subbanks. Only one of the logical subbanks is turned on for 16-byte access.
- L2-cache data array accesses take two cycles.

- All access can be pipelined except back-to-back accesses to the same logical subbank. Throughput is single cycle for different subbank accesses.
- The 64-byte accesses must be preceded and followed by an access bubble.
- Each 32-bit word is protected by seven bits of ECC.
- Physically, the logical subbank is partitioned into three 64-Kbyte blocks (bw_r_l2d). Therefore, there are 12 instances of 64-Kbyte blocks in one bank.

10.2 Block Diagrams

The following figures show an overview of L2-cache simplified addressing and data and a functional block diagram of one bank (scdata) of the L2 data array.

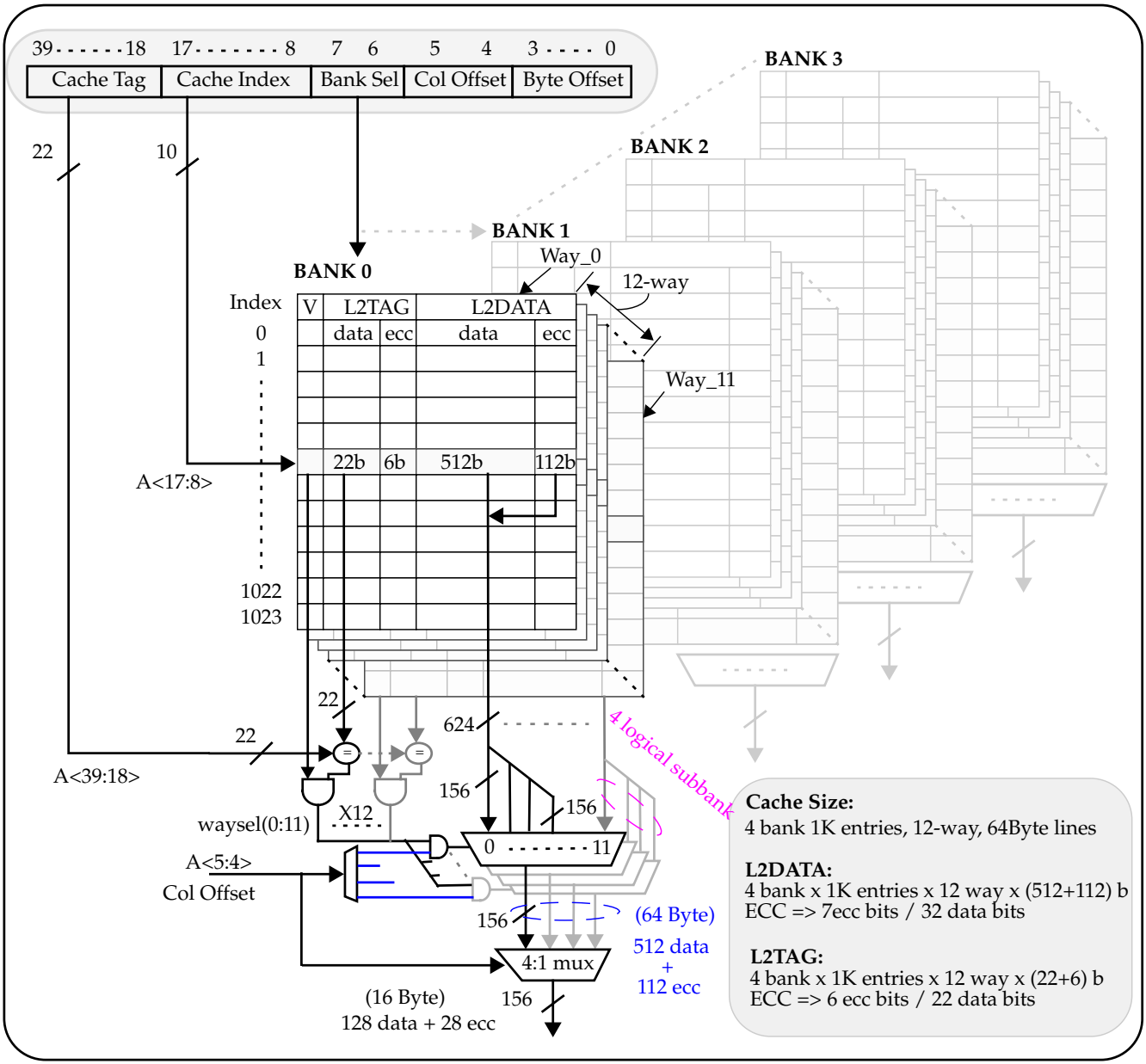

FIGURE 10-1 L2-cache Simplified Addressing and Data Overview

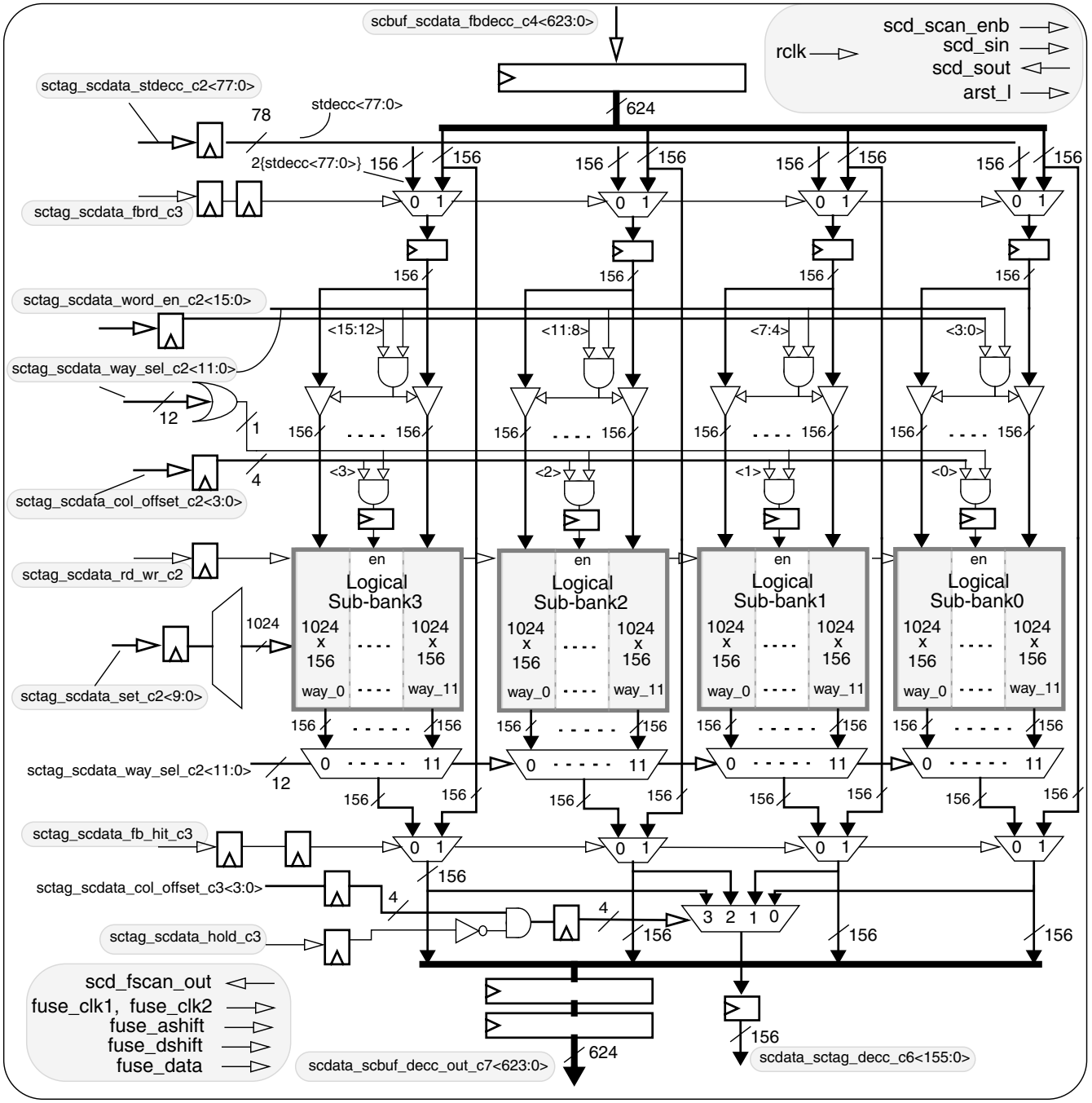

FIGURE 10-2 Functional Block Diagram of One Bank of L2 Data Array (scdata)

10.3 I/O List

The following table describes the scdata signals.

TABLE 10-1 scdata I/O Signal List

Signal Name	Type*	Source / Destination	Description
cmp_gclk[1:0]	Input		Input clock.
sctag_scdata_col_offset_c2[3:0]	Input	sctag	Select 16-byte quarter line, decoded.
sctag_scdata_way_sel_c2[11:0]	Input	sctag	Way select inputs, decoded.
sctag_scdata_rd_wr_c2	Input	sctag	Read or write operation select, low during write.
sctag_scdata_set_c2[9:0]	Input	sctag	Set index inputs.
sctag_scdata_word_en_c2[15:0]	Input	sctag	Enables for word writes, decoded, qualified externally by write so they are stable during read.
sctag_scdata_fbrd_c3	Input	sctag	MUX select for fb_data or stdata write, 1 = fb_data.
sctag_scdata_fb_hit_c3	Input	sctag	MUX select for fb_data or L2 data out, 1=fb_data.
scbuf_scdata_fbdecc_c4[623:0]	Input	scbuf	Fb_data/ecc to be MUXed with L2 output data for reads, and with store input data for writes.
sctag_scdata_stdecc_c2[77:0]	Input	sctag	Data/ecc for store operations (writes).
arst_l	Input		Power-on reset and asynchronous reset.
si	Input		Scan chain data input.
grst_l	Input		Global reset.
cluster_cken	Input		Cluster clock enable.
efc_scdata_fuse_clk1	Input	efa ctrl	1st non-overlapping clk to shift-in data to redundancy register.
efc_scdata_fuse_clk2	Input	efa ctrl	2nd non-overlapping clk to shift-in data to redundancy register.
efc_scdata_fuse_ashift	Input	efa ctrl	e-Fuse address shift control.
efc_scdata_fuse_dshift	Input	efa ctrl	e-Fuse data shift control.
efc_scdata_fuse_data	Input	efa ctrl	e-Fuse data.
scdata_sctag_decc_c6[155:0]	Output	sctag	L2 output data/ecc for loads (reads).

TABLE 10-1 scdata I/O Signal List *(Continued)*

Signal Name	Type*	Source / Destination	Description
scdata_scbuf_decc_out_c7[623:0]	Output	scbuf	L2 output data/ecc for evictions (reads).
scdata_efc_fuse_data	Output		e-Fuse data output.
so	Output		scdata scan out.

* Non-registered I/O is marked with an asterisk (*).

10.4 scdata Functional Table

The following table provides information on the scdata functions.

TABLE 10-2 scdata Functional Table

way_sel	col_offset	rd_wr	fbrd	fb_hit	word_en	Operation/ Function	Description
All 0's	X	X	X	X	X	NO-OP	Nothing happens. No way selected.
X	All 0's	X	X	X	X	NO-OP	Nothing happens. No bank selected.
1 of 12 = 1	1 of 4 = 1	1	X	0	X	Load	Read from L2 data to sctag (156b).
1 of 12 = 1	All 1's	1	X	0	X	Evict	Read from L2 data to wbb or rdma (624b).
X	X	X	X	1	X	Cache bypass "Fill Op"	Fill buffer data sent instead of wbb data (624b).
X	1 of 4 = 1	X	X	1	X	Load data bypass	Fill buffer data sent to sctag (156b).
1 of 12 = 1	1 of 4 = 1	0	0	X	(1 or 2) of16 =1	Store	Write into L2 from sctag (#b from word_en's).
1 of 12 = 1	4 of 4 = 1	0	1	X	All 1's	Fill	Write into L2 from fbb.
>1 of 12 = 1						Illegal	Way_sel's must be mutually exclusive.

10.5 scdata Timing Diagrams

Following are timing diagrams for load, evict, store, fill, load data bypass, and cache bypass operations.

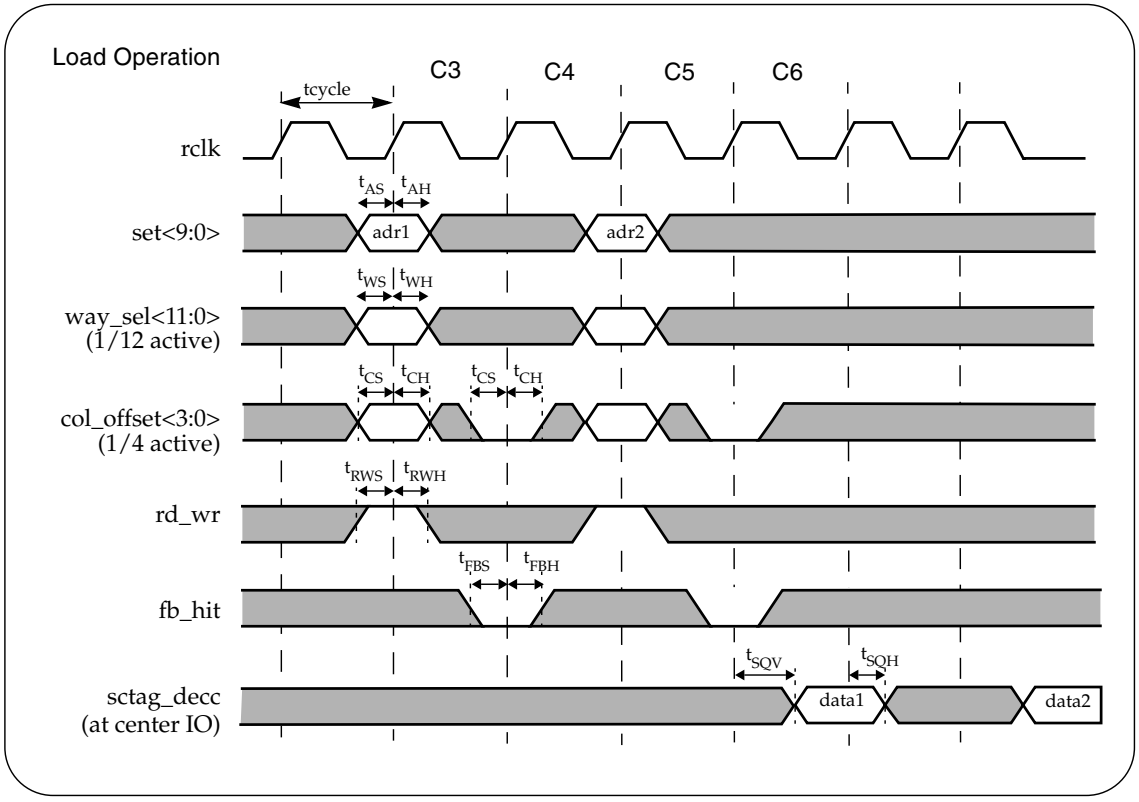

FIGURE 10-3 scdata Load Operation

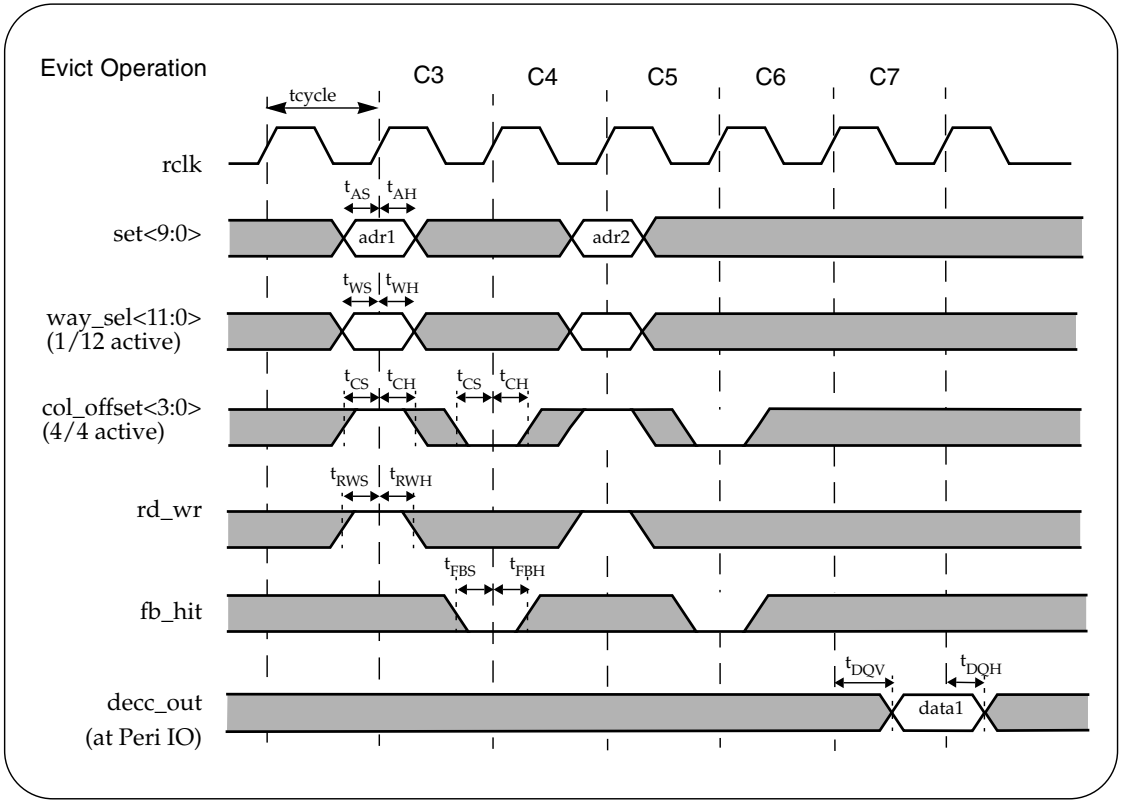

FIGURE 10-4 scdata Evict Operation

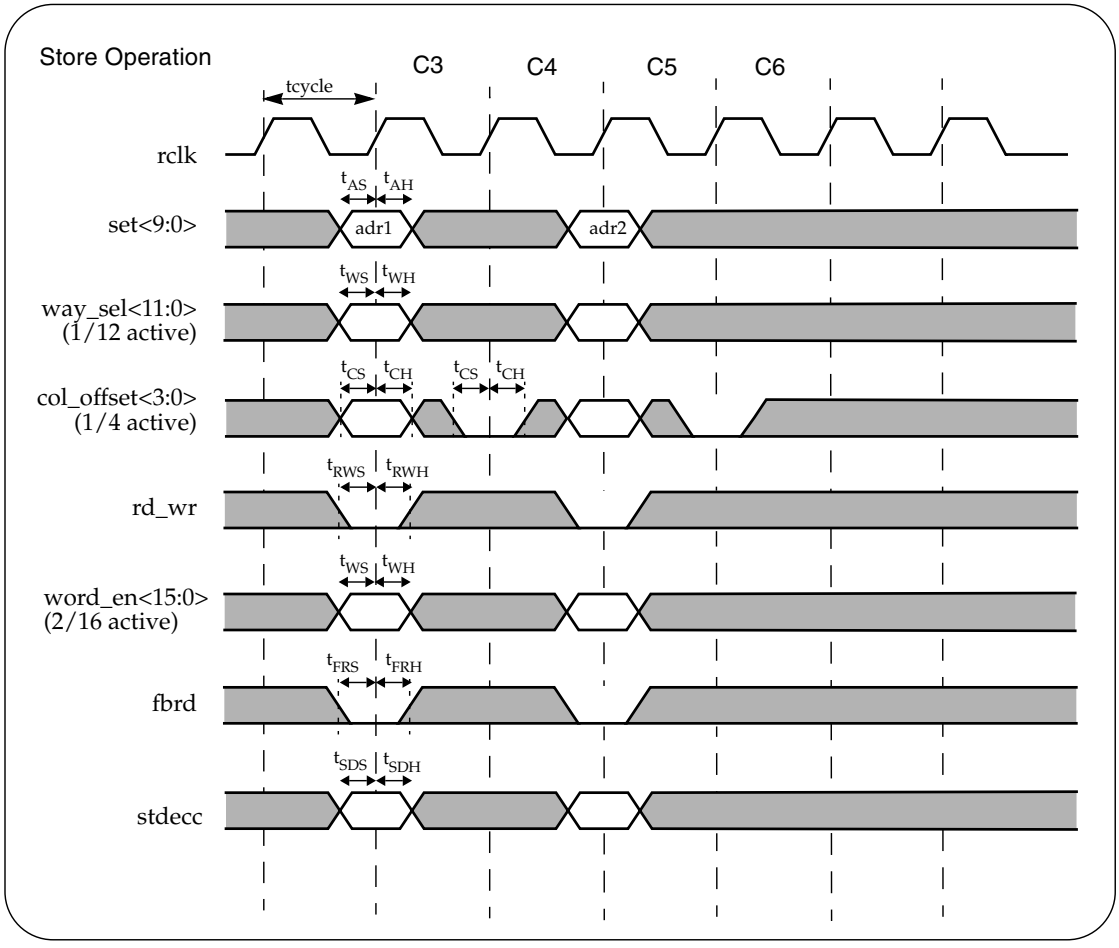

FIGURE 10-5 scdata Store Operation

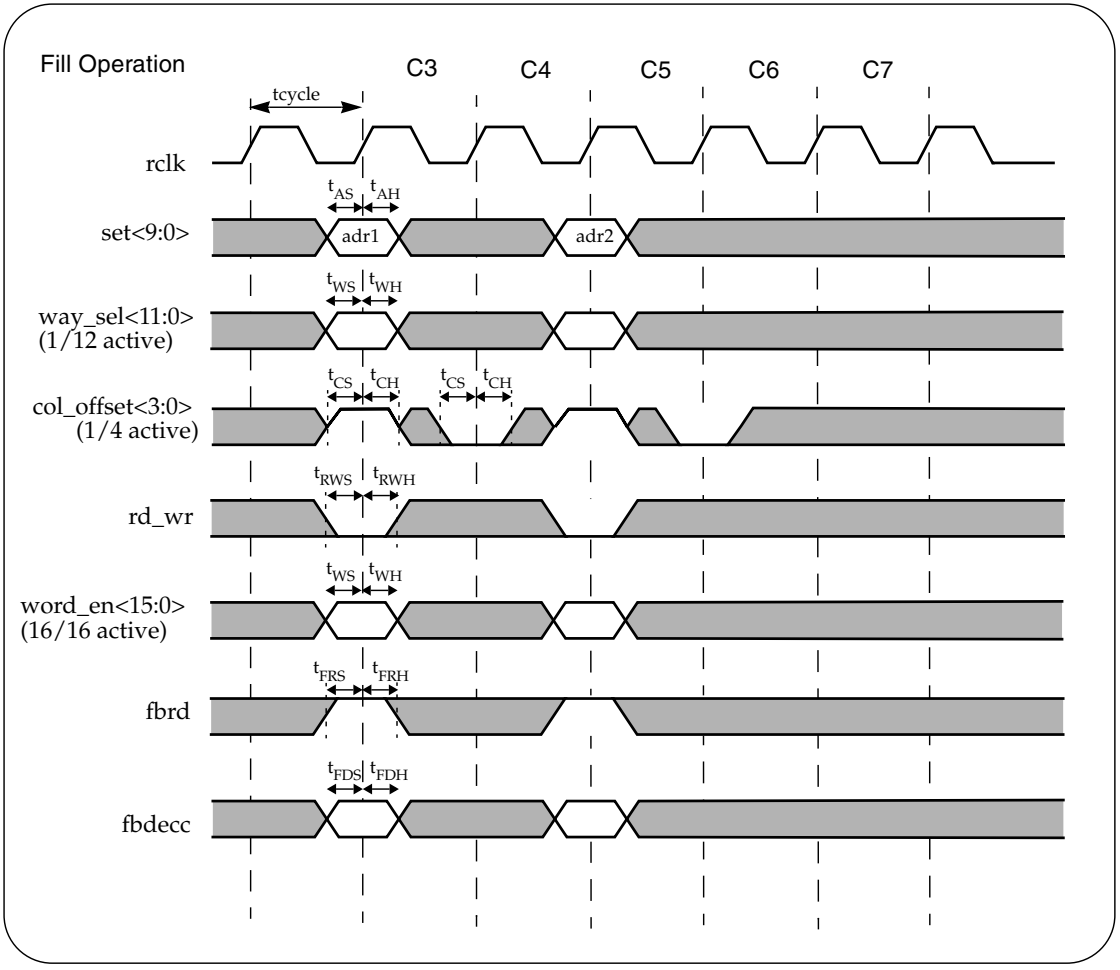

FIGURE 10-6 scdata Fill Operation

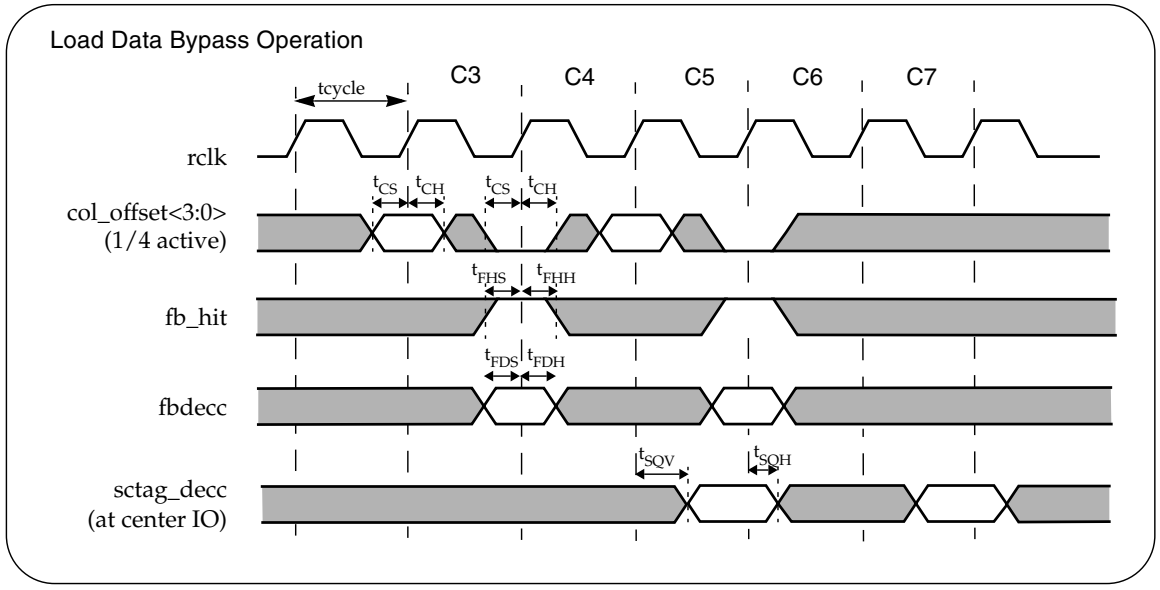

FIGURE 10-7 scdata Load Data Bypass Operation

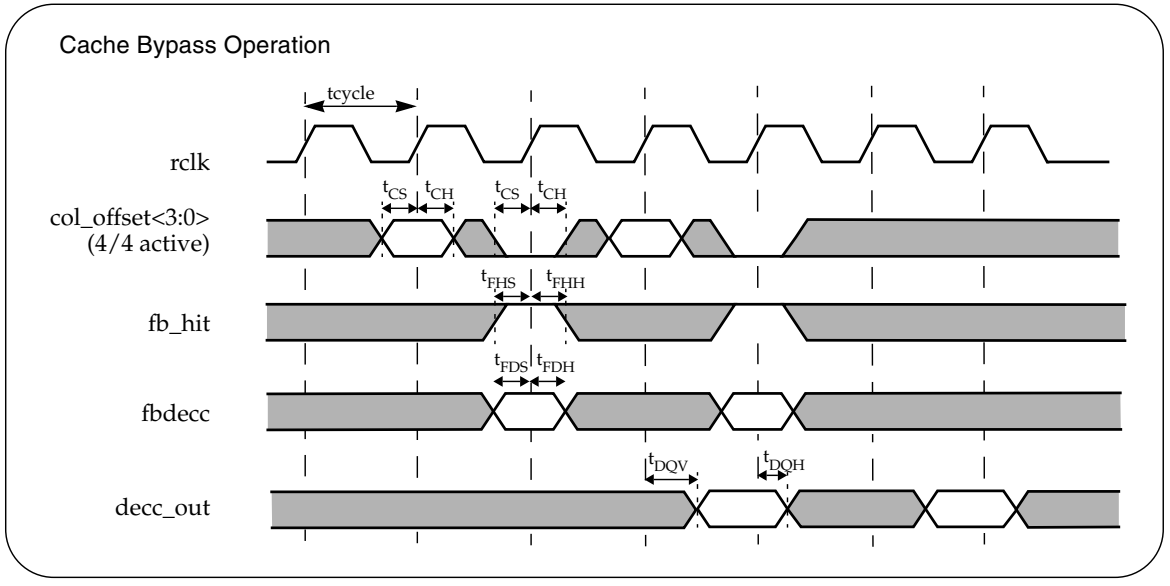

FIGURE 10-8 scdata Cache Bypass Operation

10.6 bw_r_l2d I/O List

The following table describes the bw_r_l2d I/O signals.

TABLE 10-3 bw_r_l2d I/O List

Signal Name	Type*	Source / Dest.	Description
rclk	Input*		Input clock.
col_offset	Input	Center I/O	Subbank enable.
way_sel[11:0]	Input	Center I/O	Way select inputs, decoded.
wr_en	Input	Center I/O	Write/read operation select, low during read.
set[9:0]	Input	Center I/O	Set index inputs.
word_en[3:0]	Input	Center I/O	Enables for word writes, decoded, qualified externally by write so they are stable during read.
decc_in[155:0]	Input*	Center I/O	Write data in.
decc_read_in[155:0]	Input*	bw_r_l2d	Connected to previous l2d's decc_out.
scbuf_scdata_fbdecc_top[155:0]	Input*	scbuf	scbuf data to subbank 3 and 1, to be muxed with store input data for write.
scbuf_scdata_fbdecc_bot[155:0]	Input*	scbuf	scbuf data to subbank 2 and 0, to be muxed with store input data for write.
scdata_scbuf_decc_top[155:0]	Input*	Center I/O	Read data out from subbank 3 and 1, to be routed to scbuf.
scdata_scbuf_decc_bot[155:0]	Input*	Center I/O	Read data out from subbank 3 and 1, to be routed to scbuf.
mem_write_disable	Input*	Center I/O	To gate col_offset output signal during start-up/power-up and during scan.
arst_l	Input*	Center I/O	Power-on reset/asynchronous reset.
si	Input*	Center I/O	Scan chain data input.
se	Input*	Center I/O	Scan enable.
sehold	Input*	Center I/O	Test signal to support macro test.
efc_scdata_fuse_clk1	Input*	Center I/O	1st non-overlapping clk to shift-in/out data to and from the redundancy register.
efc_scdata_fuse_clk2	Input*	Center I/O	2nd non-overlapping clk to shift-in/out data to and from the redundancy register.

TABLE 10-3 bw_r_l2d I/O List *(Continued)*

Signal Name	Type*	Source / Dest.	Description
fuse_l2d_data_in	Input*	Center I/O	Address data in for bad row/column, shifted in serially.
fuse_l2d_rid[2:0]	Input*	Center I/O	Address of one of the six redundancy registers set in the 32-Kbyte block.
fuse_l2d_rden	Input*	Center I/O	Read out the content of the redundancy registers serially.
fuse_l2d_wren[5:0]	Input*	Center I/O	When asserted, the fuse_data-in will be shifted to the redundant register set selected by rid[2:0].
fuse_read_data_in	Input*	Center I/O	Second input to the redundancy register, connected to the output of the prior redundancy register in the fuse read out chain.
word_en_buf[3:0]	Output	bw_r_l2d	Buffered word_en[3:0].
way_sel_buf[11:0]	Output	bw_r_l2d	Buffered way_sel[11:0].
set_buf[9:0]	Output	bw_r_l2d	Buffered set[9:0].
col_offset_buf	Output	bw_r_l2d	Buffered col_offset.
wr_en_buf	Output	bw_r_l2d	Buffered wr_en.
decc_in_buf [155:0]	Output	bw_r_l2d	Buffered decc_in[155:0].
scbuf_scdata_fbdecc_top_buf [155:0]	Output		Buffered scbuf_scdata_fbdecc_top [155:0].
scbuf_scdata_fbdecc_bot_buf [155:0]	Output		Buffered scbuf_scdata_fbdecc_bot[155:0].
scdata_scbuf_decc_top_buf [155:0]	Output		Buffered scdata_scbuf_decc_top[155:0].
scdata_scbuf_decc_bot_buf [155:0]	Output		Buffered scdata_scbuf_decc_bot.
decc_out[155:0]	Output		Read data out.
fuse_l2d_rid_buf[2:0]	Output		Buffered fuse_l2d_rid[2:0].
fuse_l2d_data_in_buf	Output		Buffered fuse_l2d_data_in.
fuse_l2d_rden_buf	Output		Buffered fuse_l2d_rden.
fuse_clk1_buf	Output		Buffered fuse_clk1.
fuse_clk2_buf	Output		Buffered fuse_clk2.
fuse_l2d_wren_buf[5:0]	Output		Buffered fuse_l2d_wren[5:0].
l2d_fuse_data_out	Output		Redundancy register output.

TABLE 10-3 bw_r_l2d I/O List *(Continued)*

Signal Name	Type*	Source / Dest.	Description
arst_l_buf	Output		Buffered arst_l.
se_buf	Output		Buffered se.
sehold_buf	Output		Buffered sehold.
mem_write_disable_buf	Output		Buffered mem_write_disable.
so	Output		Scan data out.

* Non-registered I/O is marked with an asterisk (*).

CHAPTER 11

L2-cache Tag Array

This chapter describes the following topics:

11.1 Functional Description of the L2 Tag

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

11.1.1 L2 Tag Array Organization

The OpenSPARC T1 processor has 172 Kbytes of 12-way set-associative Level 2 cache tag array. This array is split into four banks of 43 Kbytes each. The L2-cache tag array has the following characteristics:

- The L2-cache tag array is a single-ported SRAM and has a logical organization of 12 ways x 1024 entries x 28 bits.
- Each 28-bit tag entry is split into six error correction code (ECC) bits and 22 tag data bits.
- An L2 Tag bank is built using 12 arrays. Each array has 128 rows and 224 columns and stores 512 tags for two different ways. The 1024 entries for each way are split across two arrays.

11.1.2 L2 Tag Functions

During a read access, index[8:0] are used to select one of 512 entries for each way. Index[9] is used as an array select to select between the outputs of two arrays to generate the final output for a given way.

- In a read, data is read from all 12 ways. Only six arrays must be activated during each read operation.
- Data output from each way is compared against a 27-bit-wide lkuptag data to generate a one-bit way_select signal for each way.
- lkuptag[27:1] signal is not flopped. Bits [27:6] are timed as having a 0 ps setup time with respect to the negative edge of the clock as they go to the dynamic XNOR. Bits [5:1] are fed into static XORs, therefore setup time for those bits is with respect to the positive edge of the clock.

During a write cycle, 28-bit data is written to the way selected by the way[11:0] signal. The entry written is selected by the index[9:0] signal.

- Wordline decoder for each array is gated with read-enable, write-enable, and array-enable (index [9]) signals to deactivate unselected arrays or during idle cycles.
- Single-cycle back-to-back read and write operations are allowed.
- All inputs except lkuptag [27:1], rst_tri_en, se, and all fuse-related signals are latched at each array boundary.
- All outputs are flopped at each array boundary.
- All input/output latches and flops are scannable elements.
- Write is inhibited using the rst_tri_en signal during scan operation.
- L2 Tag array supports macro test.

11.1.3 Functional Modes

The following table lists the L2 Tag functional modes.

TABLE 11-1 L2 Tag Functional Modes

Inputs			Outputs		
rd_en	**wr_en**	**way [11:0]**	**way_sel**	**tag_way [27:0]**	**Description**
0	1	0	0	0	Data not written. No way selected.
0	1	Valid	0	0	Data written to selected way.
1	0	x	Valid	Valid	Data read out from all ways.
1	1	x	x	x	Invalid operation.

11.2 Block Diagrams

The following figures provide an L2 Tag overview and a functional diagram of the L2 Tag array.

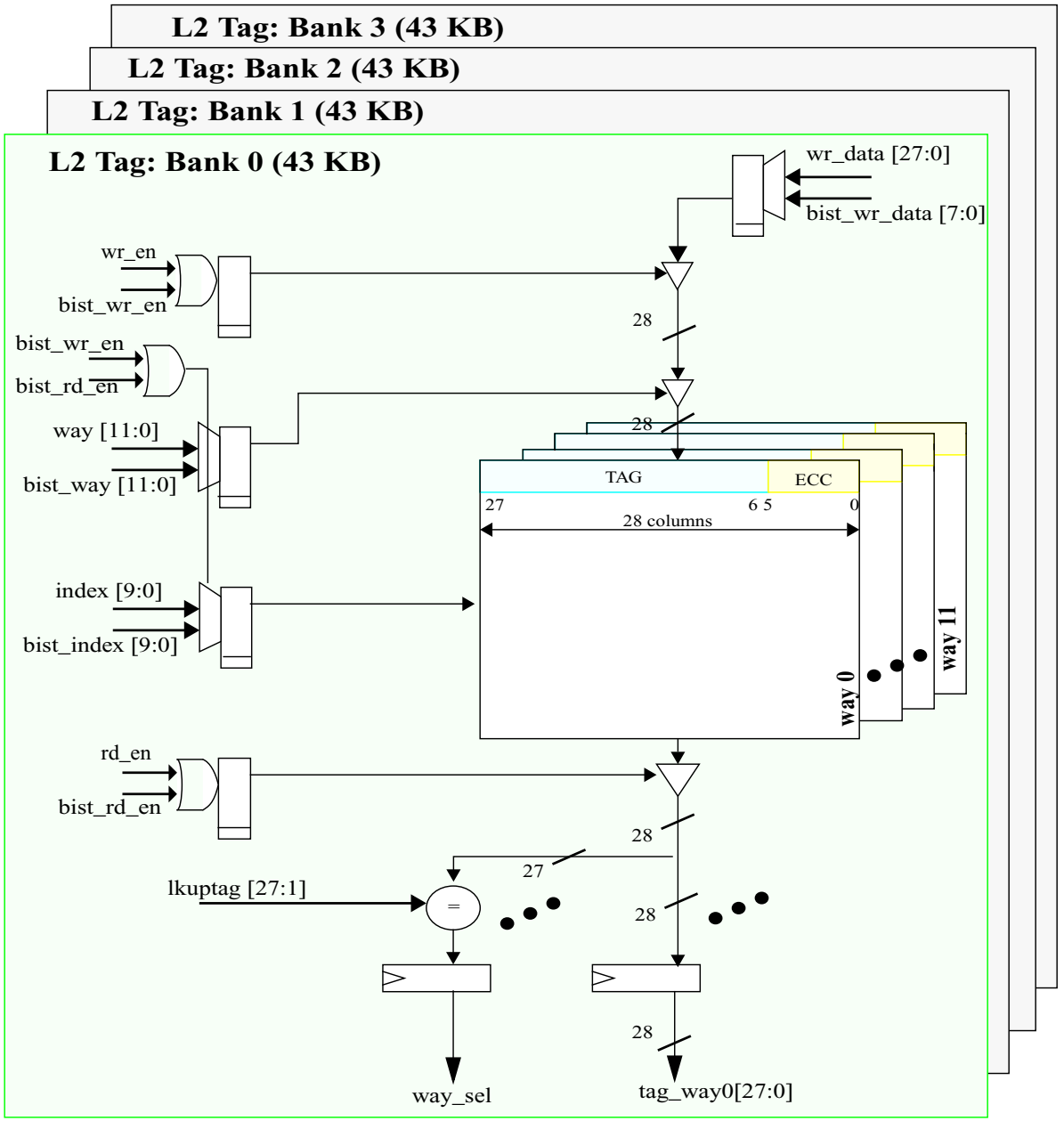

FIGURE 11-1 L2 Tag Overview

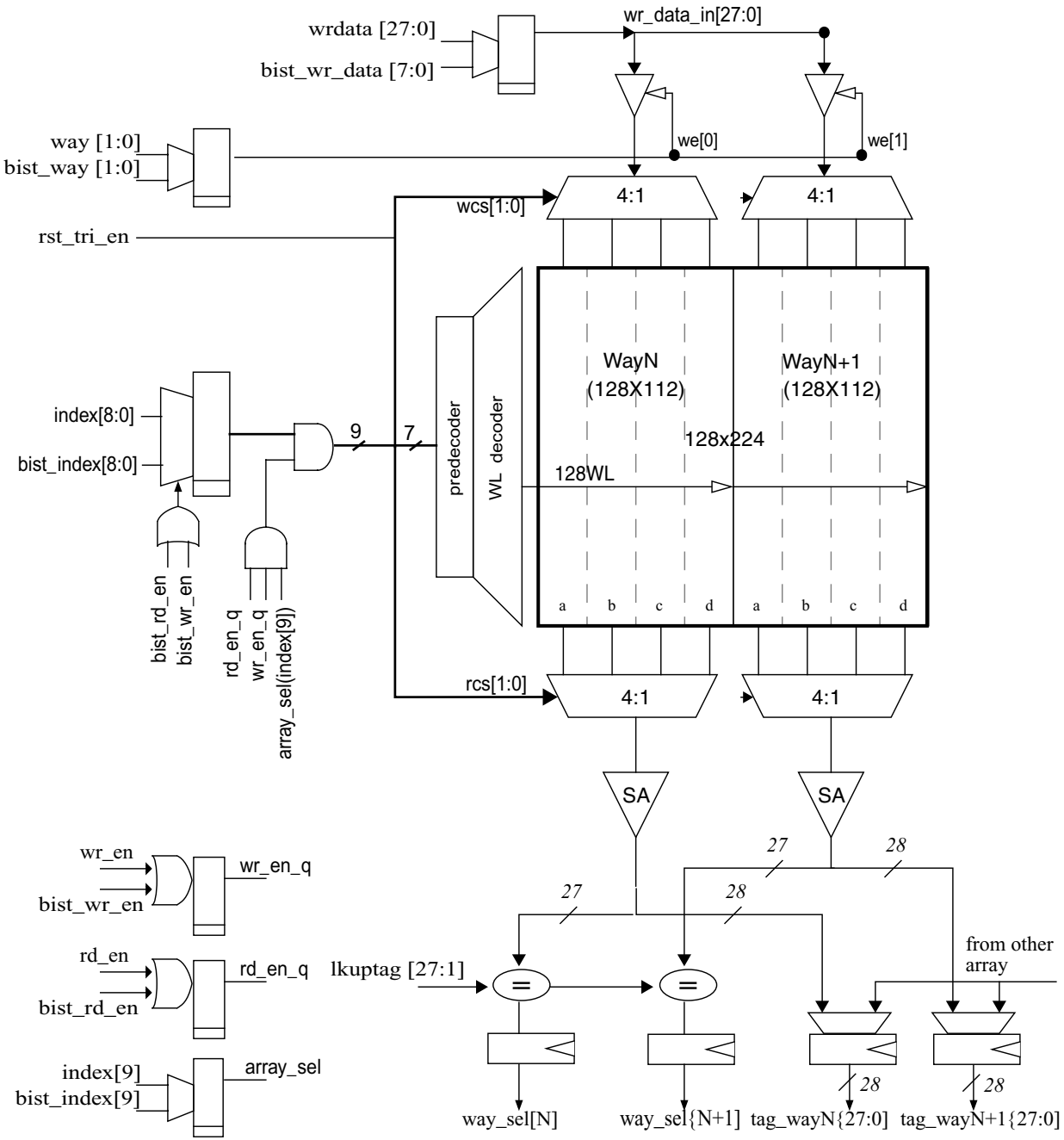

FIGURE 11-2 Functional Diagram of the L2 Tag Array

11.2.1 L2 Tag Logic Symbol

The L2 Tag logic symbol is shown in the following figure.

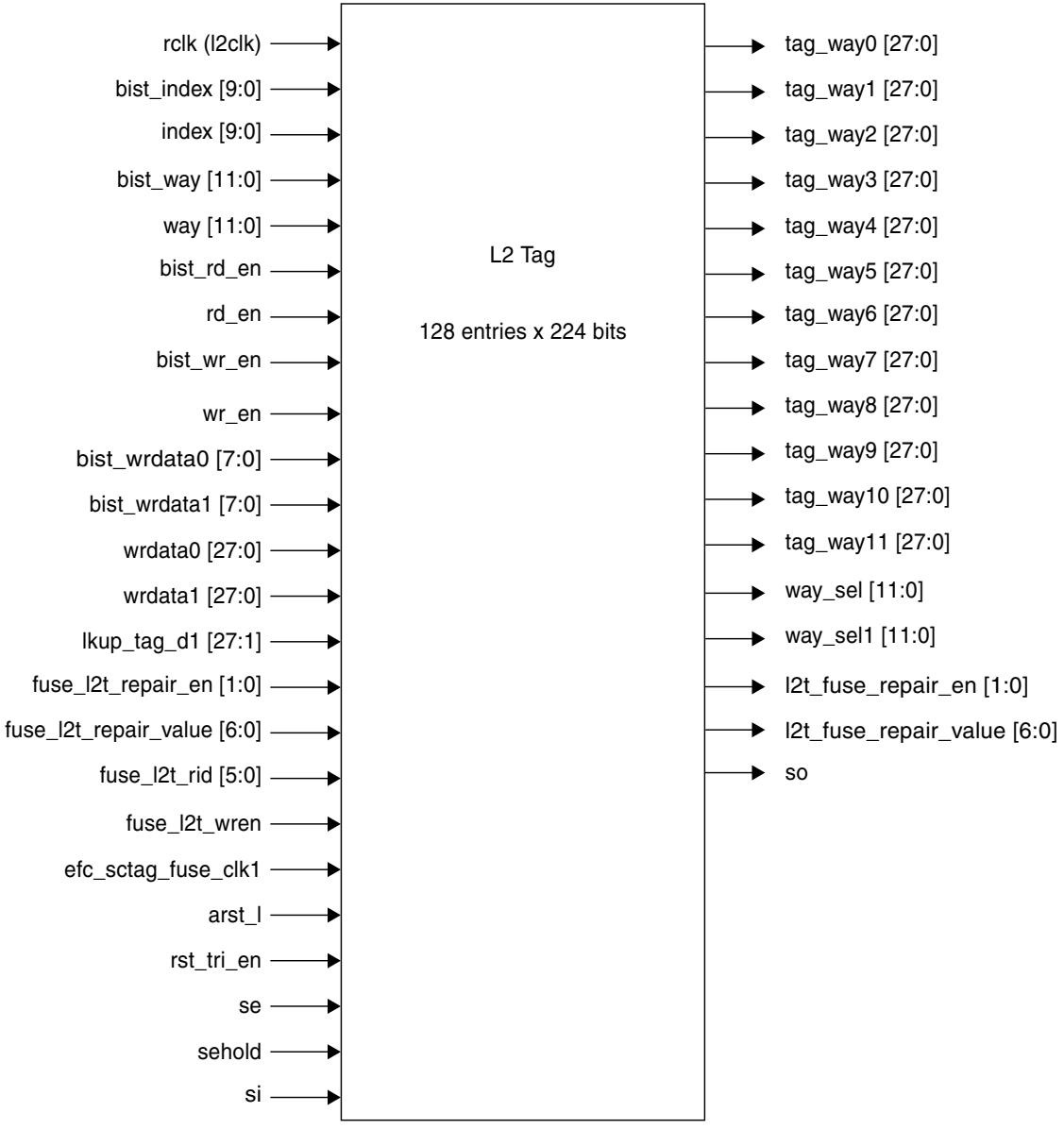

FIGURE 11-3 L2 Tag Symbol With Signals

11.3 I/O List

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

TABLE 11-2 L2 Tag Input /Output Signals

Signal Name	Type	Source/ Destination	Description
rclk	Input	clock grid	Input clock.
index [9:0]	Input	sctag_arbaddrdp	Address input for normal read and write.
bist_index [9:0]	Input	sctag_mbist	Address input used during BIST operation.
rd_en	Input	sctag_arbctl	Read-enable control input for normal read operation.
bist_rd_en	Input	sctag_mbist	Read-enable control input used during BIST operation.
way [11:0]	Input	sctag_arbctl	Way select used during a fill/tag write.
bist_way [11:0]	Input	sctag_mbist	Way select used during BIST operation.
wr_en	Input	sctag_arbctl	Write-enable control input.
bist_wr_en	Input	sctag_mbist	Write-enable control input used during BIST operation.
wrdata0 [27:0]	Input	sctag_arbaddrdp	Tag data to be written into left side arrays.
bist_wrdata0 [7:0]	Input	sctag_mbist	Tag write data into left side arrays during BIST.
wrdata1 [27:0]	Input	sctag_arbaddrdp	Duplicate of wrdata0 for right side arrays.
bist_wrdata1 [7:0]	Input	sctag_mbist	Duplicate of bist_wrdata0 for right side arrays.
lkup_tag_d1 [27:1]*	Input	sctag_tagdp	Compare data input to the comparator. (This signal is not latched inside L2 Tag.)
way_sel [11:0]	Output	sctag_tagctl	Tag compare output for each way.
way_sel1 [11:0]	Output	sctag_tagctl	Tag compare output for each way.
tag_way11 [27:0] ... tag_way0 [27:0]	Output	sctag_tagldp	Tag data output for all the 12 ways.
fuse_l2t_wren*	Input	fuse_header	Redundancy register write enable.
fuse_l2t_rid [5:0]*	Input	fuse_header	Redundancy register ID. [5:2] determines sub-bank. [1:0] determines row/column redundancy.

TABLE 11-2 L2 Tag Input /Output Signals *(Continued)*

Signal Name	Type	Source/ Destination	Description
fuse_l2t_repair_value [6:0]	Input	fuse_header	Data to be stored in redundancy registers.
fuse_l2t_repair_en [1:0]	Input	fuse_header	Enable bits to turn redundancy on.
l2t_fuse_repair_value [6:0]	Output	fuse_header	Read data out from redundancy registers.
l2t_fuse_repair_en [1:0]	Output	fuse_header	Read data out from repair enable registers.
efc_sctag_fuse_clk1	Input	fuse_header	e-Fuse clock.
rst_tri_en*	Input	teststub	Gates the write operation during scan.
arst_l*	Input	TBD	Asynchronous reset for redundancy registers.
sehold*	Input	TBD	Scan hold input used during macrotest.
si	Input	TBD	Scan input.
so	Output	TBD	Scan output.
se*	Input	TBD	Scan enable.

*Non-registered I/O are marked with an asterisk (*).

11.4 Timing Diagrams

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

11.4.1 Read Timing Diagram

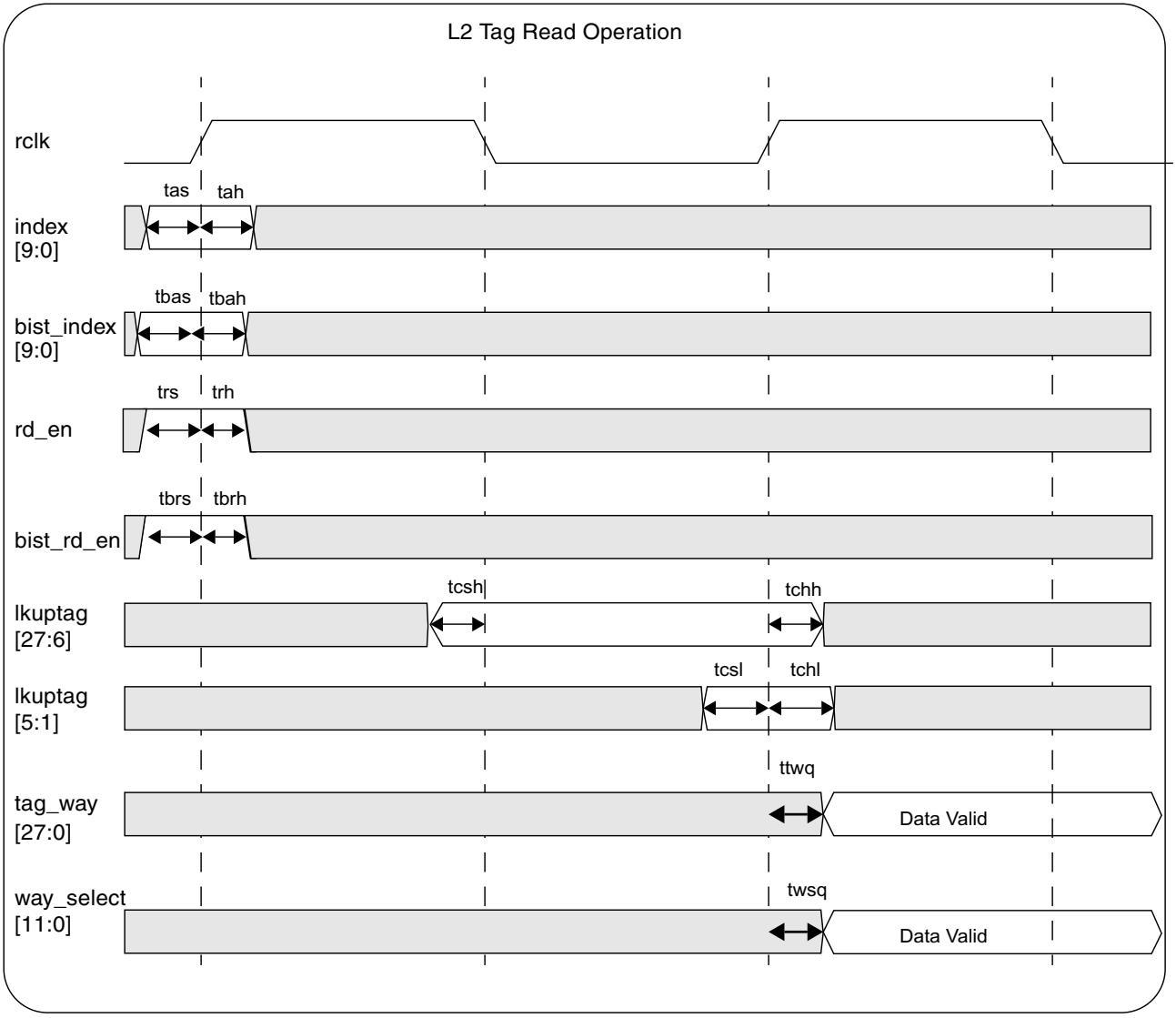

FIGURE 11-4 I/O Read Timing Diagram of L2 Tag

11.4.2 Write Timing Diagram

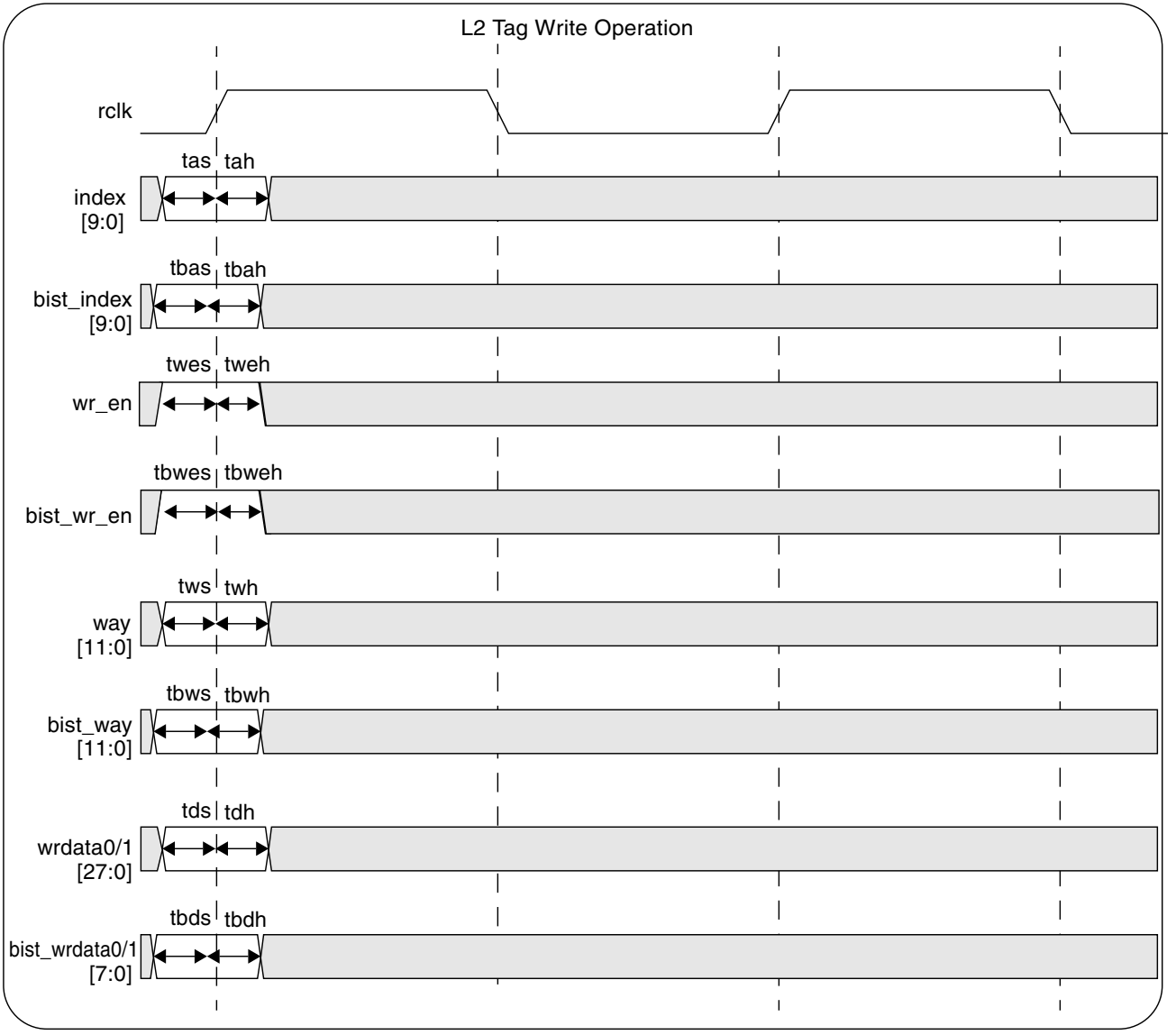

FIGURE 11-5 I/O Write Timing Diagram of L2 Tag

CHAPTER 12

Store Buffer CAM

This chapter describes the following topics:

12.1 Functional Overview of the SCM

The store buffer content-addressable memory (SCM) is one of two arrays associated with the store buffer of the load and store unit (LSU). The other array is the data random access memory (RAM), which is implemented as bw_rf_32x80. The SCM has the following characteristics:

- The store buffer CAM has a single read/write port and CAM/lookup port.
- Of the three possible operations, the read and write are mutually exclusive with respect to each other, but either can coincide with a lookup.

The SCM is organized into two banks.

- The first bank is 32 entries by 38 bits and performs a standard lookup.
- The second bank is 32 entries by eight bits and is used to qualify the look-up results of the first bank.

Each bank is logically divided into four groups containing eight entries (one per thread) with only one of the groups being active at a time. Physically, all the entries are compared with the input vector simultaneously. The results are separated with a 4:1 mux.

- A CAM hit is defined as the incoming CAM key vector matching one of eight entries within the selected group.
- A multi-hit occurs when more than one entry has a match with the input key.

- Full and partial outputs are also provided to reflect portions of a match. They are generated by qualifying the look-up results of the 32x38 bank with the corresponding outputs from the 32x8 bank during a lookup.
- All the outputs from the look-up operation (hit, multi-hit, full, partial) can be explicitly masked off by an 8-bit input mask vector.
- Both the CAM input key and the write data can be written into the CAM, but only the CAM data is used in the look-up operation.

12.1.1 Read Operation

The predecoder and wordline decoder are triggered off the rising edge of rclk. The sense amplifier is also triggered off the rising edge for the clock.

- On the rising edge of the rclk, the sense amplifier is active and the read data will be read out. The memory goes into recovery state when the sense amplifier is turned off. The bitlines are precharged and the wordline gets deasserted until the next cycle.
- Data is guaranteed to be valid after the quoted access time and is held static until the next read access.
- Read access can occur on consecutive clock cycles without restrictions.
- There is no column decoder when all the bits are read out.

12.1.2 Write Operation

The write operation starts on the rising edge of rclk, which triggers the predecoder and wordline decoder and turns on the bitline pass gate enabling the data input through to the bitlines.

- Row address, data input, and write-enable signals all must set up before the rising edge of rclk. There is no column decoder when all the cells in the row are written at the same time.
- Data is written into the memory within half a clock cycle.
- On the falling edge of rclk, write recovery initiates, the bitlines are precharged, and the memory contents are made inaccessible until the next write.
- Clock qualified with write enable and read enable is used to fire the wordline and control the I/O pass gate.

12.1.3 Look-up Operation

The look-up operation starts on the rising edge of rclk. Clock qualified with "stb_cam_vld" (clk_l) is used to activate the latch to let the key input data latch into the look-up circuits and make the comparison. The result of the look-up comparison also must be qualified with clk_l at the end.

12.1.4 Logical Size

The bw_r_scm macro has 32 entry/rows and 45 bits. Each SPARC core instantiates one SCM block. There are eight cores in a chip.

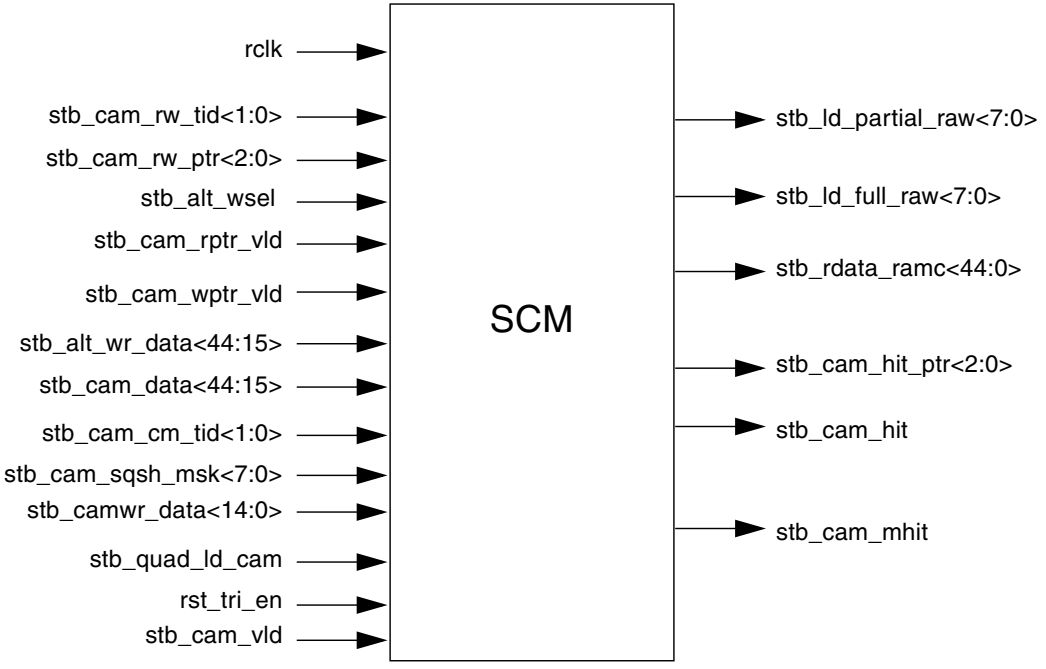

FIGURE 12-1 SCM Logic Symbol

12.1.5 Operation Modes

Operation modes for SCM are the following:

- The rst_tri_en signal gates write operation. This signal gets asserted in scan mode. When it asserts (rst_tri_en="1"), wordlines are still asserted but clk_w is not toggled, so the bitlines are disconnected from the write data driver.
- Read is disabled when stb_cam_rptr_vld is low.
  - Wordlines are still asserted if write enable signal stb_cam_wptr_vld is 1.
  - Sense amp is off.
  - Data on the readout port is the last latched read value if write does not occur. If there is a write, data on the readout port is the value of the write data input. This is the write-through feature.
- Write is disabled when stb_cam_wptr_vld is low.
  - Memory content is not affected regardless of the value of the address and data input.
  - Wordlines are still asserted but clk_w does not toggle, so the bitlines are disconnected from the write data driver.
- Lookup is disabled when stb_cam_vld is low.
  - Memory content is not affected regardless of the value of the address and data input.
  - Key data is disconnected from look-up data drivers.

TABLE 12-1 Memory Behavior in Functional Mode

#	rst_tri_en	we	re	lk	ml	wl	bl	sa	Operation
1	0	0	0	0	Prechg	Reset 0	Prechg	Off	No operation
2	0	1	0	0	Prechg	Assert 1	Toggle	On	Write
3	0	0	0	1	Toggle	Assert 1	Prechg	Off	Look up
4	0	0	1	0	Prechg	Assert 1	Toggle	On	Read
5	0	0	1	1	Toggle	Assert 1	Toggle	On	Read and lookup
6	1	x	1	0	Prechg	Assert 1	Toggle	On	Read
7	1	x	1	1	Toggle	Assert 1	Toggle	On	Read and lookup

Note – During scan mode, SCM still functions as normal mode.

12.2 I/O List

The following table describes the SCM I/O signals.

TABLE 12-2 SCM I/O Signal List

Pin	Type	Flop/ Latch	Description
rclk	Input	None	Clock input
stb_cam_wptr_vld	Input	Latch	Write pointer valid or write enable
stb_cam_rptr_vld	Input	Latch	Read pointer valid or read enable
rst_tri_en	Input	None	Disable write enable in test mode
stb_cam_rw_tid[1:0]	Input	Latch	Thread ID for write/read
stb_cam_rw_ptr<2:0>	Input	Latch	Read/write pointer
stb_alt_wsel	Input	Latch	Data input mux selection
stb_cam_data<44:15>	Input	Latch	Write data or CAM key input
stb_alt_wr_data<44:15>	Input	Latch	Write data or CAM key input
stb_camwr_data<14:0>	Input	Latch	Write/CAM key data input
stb_cam_vld	Input	Latch	CAM valid
stb_cam_cm_tid<1:0>	Input	Latch	Thread ID for CAM operation
stb_cam_sqsh_msk<7:0>	Input	None	Mask for squashing CAM results
stb_quad_ld_cam	Input	None	Quad ID for CAM operation
stb_rdata_ramc<44:0>	Output	None	Read data output
stb_ld_full_raw<7:0>	Output	None	Load with full raw
stb_ld_partial_raw<7:0>	Output	None	Load with partial raw
stb_cam_hit_ptr<2:0>	Output	None	CAM hit pointer
stb_cam_hit	Output	None	Any hit in SCM
stb_cam_mhit	Output	None	Multiple hits in SCM

CHAPTER 13

Translation Lookaside Buffer

This chapter describes the following topics:

13.1 Functional Description of the TLB

The translation lookaside buffer (TLB) is commonly used as an instruction lookaside buffer (I-TLB) or a data lookaside buffer (D-TLB) for instruction fetch and load and store units. It consists of 64-entry fully associative CAM, 64-entry data memory, four comparators, and control logic to support various modes of operations.

- Tag entry is 59 bits wide which consists of a parity bit; a valid bit; a locked bit; a used bit; 35-bit virtual address; three select bits supporting 256-Mbyte, 4-Mbyte, 64-Kbyte, and 8-Kbyte page addresses; 3-bit partition ID (PID); 13-bit context; and a real bit identifying if the entry is real or non-real.
- Data entry is 43 bits wide including three select bits (12, 16, and 23) specifying the supported page sizes. These three select bits are active low signals.
- Tag can be operated in four modes – CAM, demaps, read, and write. The write operation supports index write and replacement write. Data RAM supports three modes – index and replacement write, index read, and CAM-addressed read. These operations are mutually exclusive.
- Physical address, either from a CAM-addressed read or a bypass, is compared against four sets of input physical tags to determine if there is a cache hit in TLB. In bypass mode, the TLB content is unused.
- All the operations are done in one cycle, except the calculation of index for replacement write which is a two-cycle operation.

- The replacement algorithm will only replace an entry which is not used. If all entries are used, then TLB will attempt to clear all unlocked used entries. If this still does not yield a replacement entry, then entry[63] is chosen by default, even if this entry is locked.
- The used bit reset operation is blocked from other normal operations of the TLB.
- Every write in the TLB is preceded by an auto demap to ensure that there is no duplicate entry in the tag. The auto demap will generate internal hit signals which are used to invalidate duplicate entries when the write cycle starts.
- TLB has two resets – arst_l (active low) is an asynchronous power-on reset (POR) reset and rst_soft_l (active low) is a synchronous software reset.
- All inputs are flopped or latched except tlb_wr_tte_tag, tlb_wr_tte_data, rst_soft_l, and the following global signals – arst_l, adj[7:0], se, hold, and rst_tri_en.
- All outputs are flopped or latched except tlb_cam_hit and tlb_pgnum_crit[39:10].
- All normal operations are inhibited when rst_tri_en is enabled.
- I/O slice including sense amplifier (SA) design are similar to directory content-addressable memory (DCM) and store buffer content-addressable memory (SCM) design.

TABLE 13-1 Translation Lookaside Buffer Features

Feature	Value
Size	Tag: 64X59 Data: 64X43
Word lines	Tag/Data: 64+1 modeled WL+2 dummy WL
Columns	Tag: 59+1 modeled BL+2 dummy+1 unused Data: 43+1 modeled BL+2 dummy+1 unused
Number of ports	One

13.2 Block Diagrams

This section provides diagrams of translation lookaside buffer top-level logic blocks, a TLB functional diagram, and tag read/write control clocks.

13.2.1 Top-Level Logic Block Diagram

The following figure shows the three main TLB blocks – tag, data, and comparator.

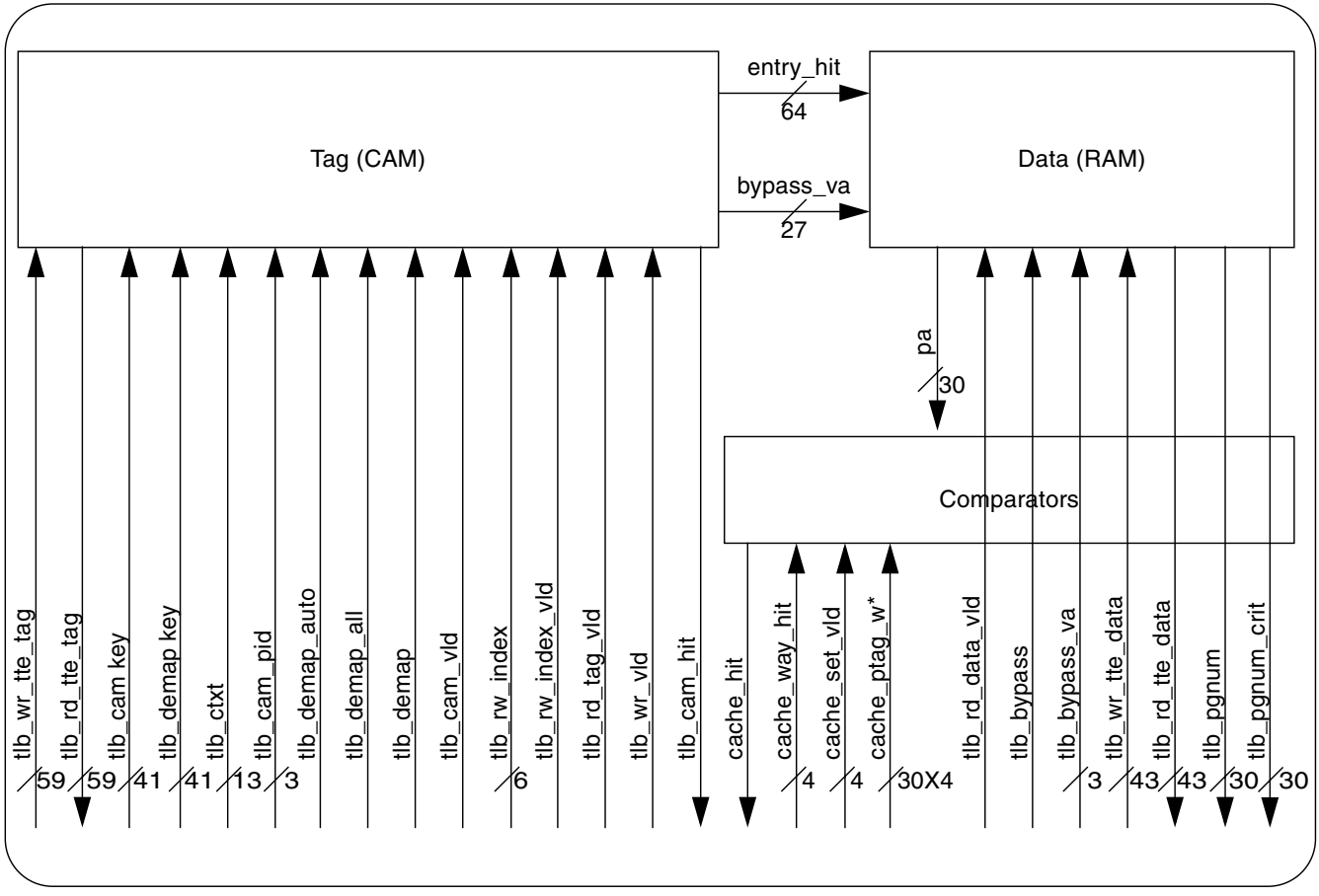

FIGURE 13-1 TLB Top-Level Block Diagram

13.2.2 Functional Block Diagram of the TLB

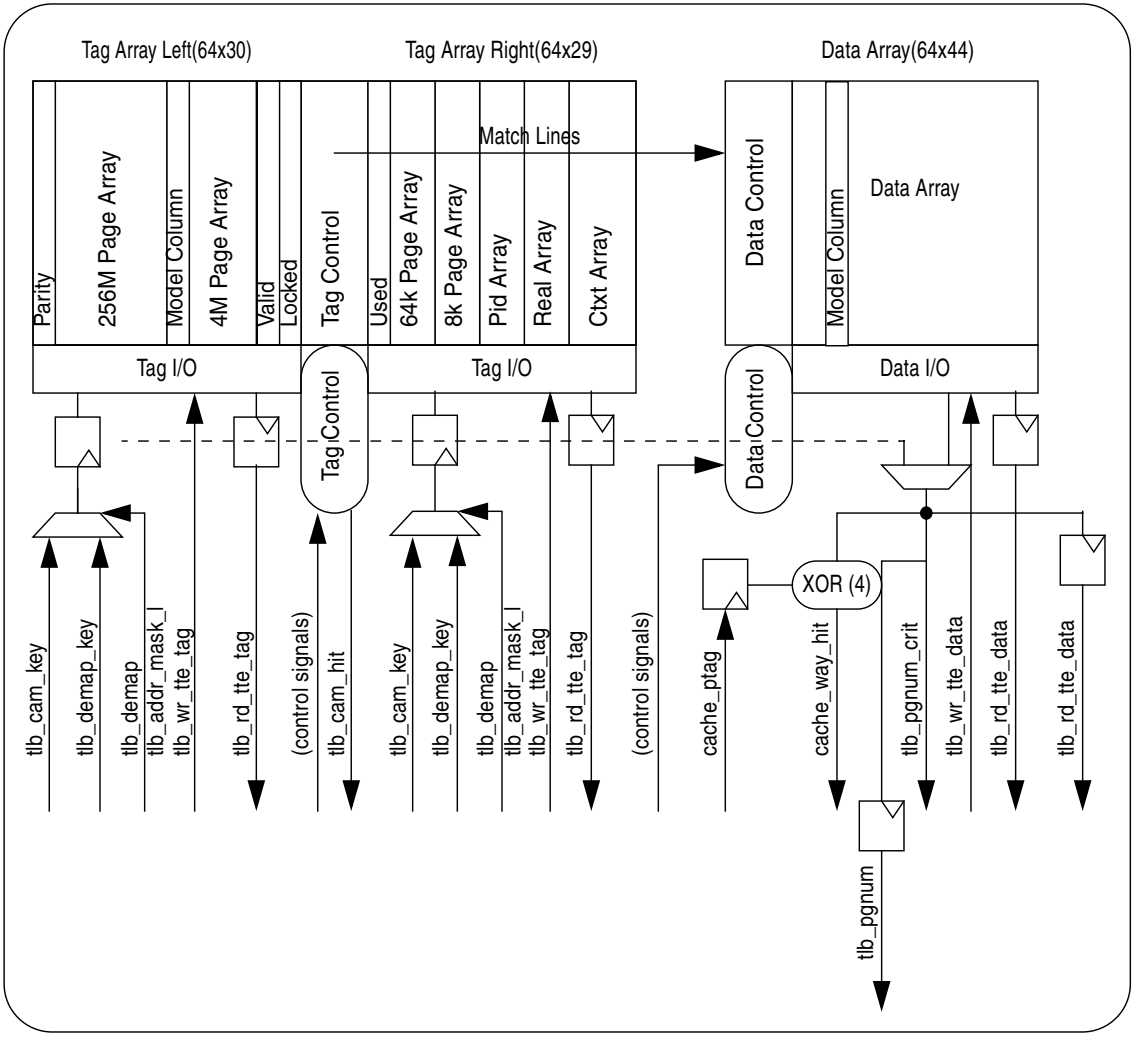

FIGURE 13-2 TLB Functional Block Diagram

13.3 I/O List

The following table describes the TLB I/O signals.

TABLE 13-2 TLB I/O Signal List

Pin	Type	Driver	Description	Flop
rclk	Input		RCLK Input.	
adj[7:0]	Input		Margin control pins.	
arst_l	Input		Global reset pin. Resets valid bits and deactivates TLB.	
hold	Input		If 1, hold the states of input flops.	
se	Input		Scan shift enable.	
si	Input		Scan input.	Yes
cache_ptag_w0[29:0]	Input		Way 0 comparator direct inputs.	Yes
cache_ptag_w1[29:0]	Input		Way 1 comparator direct inputs.	Yes
cache_ptag_w2[29:0]	Input		Way 2 comparator direct inputs.	Yes
cache_ptag_w3[29:0]	Input		Way 3 comparator direct inputs.	Yes
cache_set_vld[3:0]	Input		Enables four ways.	Yes
rst_soft_l	Input		Synchronous software reset.	
tlb_addr_mask_l	Input		If 0, masks the cam key (for 32b/64b).	Yes
tlb_bypass	Input		Bypass TLB translation.	Yes
tlb_bypass_va[12:10]	Input		Bypass other bits from CAM data.	Yes
tlb_cam_key[40:0]	Input		Keys for CAM operation.	Yes
tlb_cam_pid[2:0]	Input		Partition ID.	Yes
tlb_cam_vld	Input		CAM valid. If 1, activates CAM operation.	Yes
tlb_ctxt[12:0]	Input		Context keys.	Yes
tlb_demap	Input		Demap valid. If 1, activates demap operation.	Yes
tlb_demap_all	Input		Demap all. Invalidates all entries which are not locked.	Yes
tlb_demap_auto	Input		Auto demap.	Yes
tlb_demap_key[40:0]	Input		Keys for demap operation.	Yes

TABLE 13-2 TLB I/O Signal List *(Continued)*

Pin	Type	Driver	Description	Flop
tlb_rd_data_vld	Input		Data read enable.	Yes
tlb_rd_tag_vld	Input		Tag read enable.	Yes
tlb_rw_index[5:0]	Input		Index address, address 64 rows.	Yes
tlb_rw_index_vld	Input		Enable index address for index write.	Yes
tlb_wr_tte_data[42:0]	Input		Data write inputs.	No
tlb_wr_tte_tag[58:0]	Input		Tag write inputs.	No
tlb_wr_vld	Input		TLB write enable.	Yes
rst_tri_en	Input		Reset enable.	
so	Output	8X	Scan output.	Yes
cache_hit	Output	12X	Sum of CACHE_WAY_HIT[3:0]'s.	Yes
cache_way_hit[3:0]	Output	24X	Tag comparison results.	Yes
tlb_cam_hit	Output	24X	CAM hit.	No
tlb_pgnum[39:10]	Output	8X	Bypass or translated page number.	Yes
tlb_pgnum_crit[39:10]	Output	12X	Bypass or translated page number (not flopped).	No
tlb_rd_tte_data[42:0]	Output	24X	Data read outputs.	Yes
tlb_rd_tte_tag[58:0]	Output	8X	Tag read outputs.	Yes

CHAPTER 14

Register File 16 x 128

This chapter describes the following topics:

14.1 Functional Description of the Block

The register file 16 x 128 (RF16x128d) is a 16-entry, 128-bit-wide register file with two ports. Port A is read only. Port B is write only. The block has the following characteristics:

- No sharing or data dependency on the control mechanism between the two ports.
- Single-cycle read and write, both triggered in clock high phase.
- Supports all combinations of consecutive reads and writes.
- If both read and write access are performed simultaneously (both on phase 1) on the same entry, the newly written data is likely to be read out but not guaranteed.
- The high read current memory cell is connected to a full-swing bitline with a single-ended sense amplifier.
- Differential write mechanism is used.
- Both read and write address is fully decoded.
- Write is inhibited when rst_tri_en is enabled.
- Read is gated by rst_tri_en and se.
- All inputs are latched except for reset_l.
- Latches are transparent during the low phase of clock and remain closed during the high phase.

- Output flops are located outside the block.

TABLE 14-1 Memory Behavior Under Combinations of read_en, write_en, reset/pwr_dn

#	rst_l	se	hold	rstri	we	re	wwl	wbl	rwl	rbl	sa	dout
1	1	0	0	0	0	0	Reset 0	Prechg	Reset 0	Prechg	Off	Last read
2	1	0	0	0	1	0	Assert 1	Toggle	Reset 0	Prechg	Off	Last read
3	1	0	0	0	0	1	Reset 0	Prechg	Assert 1	Toggle	On	New read
4	1	0	0	0	1	1	Assert 1	Toggle	Assert 1	Toggle	On	New read/ collision
5	1	1	0	1	x	x	Reset 0	Toggle	Reset 0	Prechg	Toggle	Read 1
6	1	0	1	0	1	0	Toggle	Toggle	Toggle	Toggle	Toggle	Write access
7	1	0	1	0	0	1	Toggle	Toggle	Toggle	Toggle	Toggle	Read access/ capture
8	0	x	x	x	x	x	Reset 0	Prechg	Reset 0	Prechg	Off	Read 0

TABLE 14-2 Blocks Instantiating bw_r_rf16x128d

Block Name	Location	Logical Org (entry x word x way)	Physical Org (row x column)	Instance Count	Special Features
l2mbdata	scbuf	16x100x1	16x128	4	Fully decoded address inputs

14.2 Block Diagrams

The following figures provide a logic symbol diagram and a functional block diagram.

14.2.1 Top-Level Location

The top-level location instantiating bw_r_rf16x128d is shown in the following table.

TABLE 14-3 Top-Level Location Instantiating bw_r_rf16x128d

Location	Instance Count in This Location	Instance Count in Chip
sctag	1	4

14.2.2 RF16x128d Logic Symbol

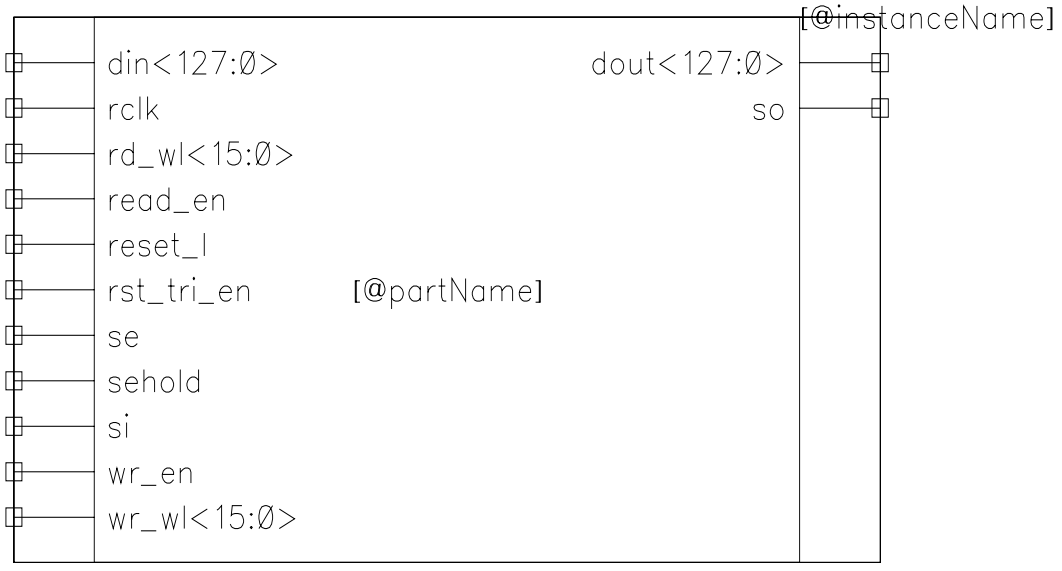

FIGURE 14-1 RF16x128d Logic Symbol

14.2.3 Functional Block Diagram of RF16x128d Array

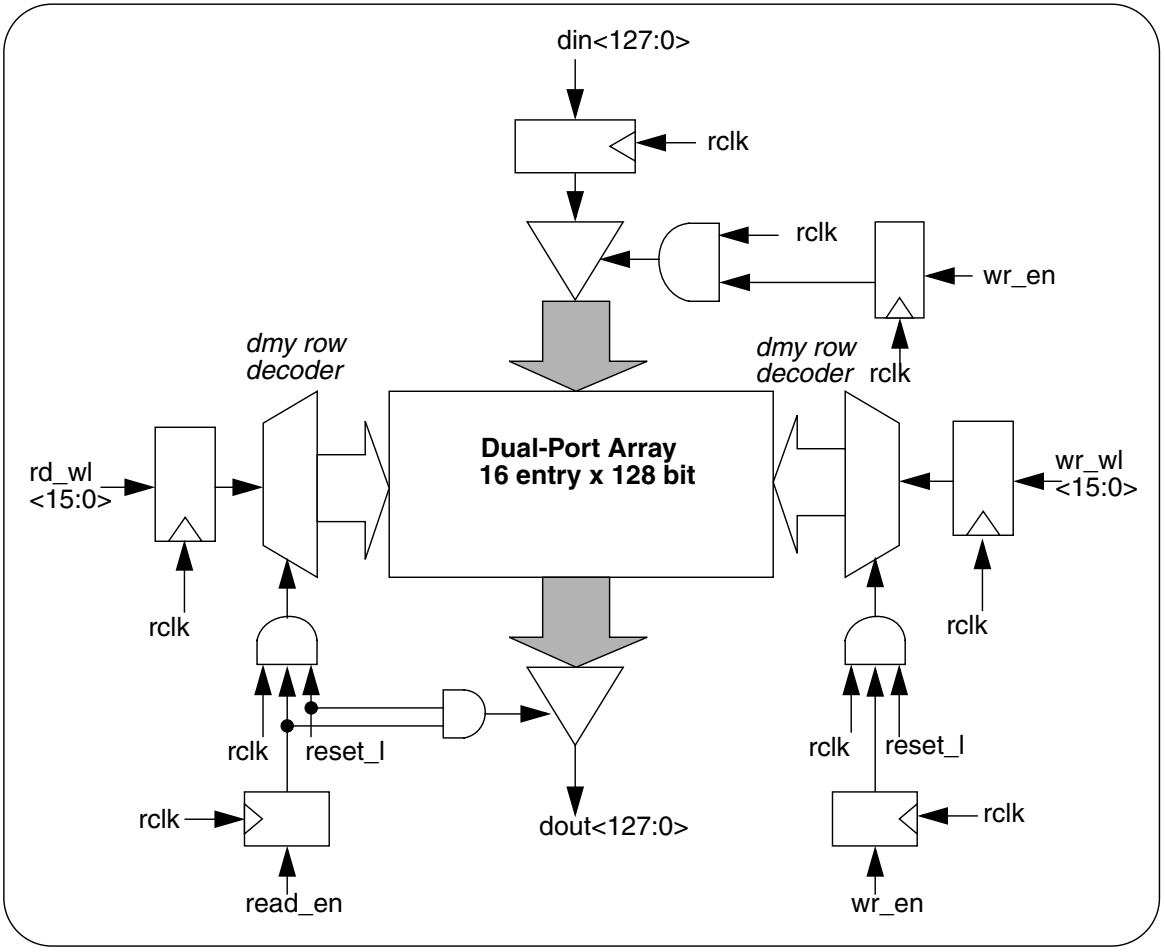

FIGURE 14-2 Functional Block Diagram of bw_r_rf16x128d

14.3 I/O List

The following table lists the dual-port register file I/O signals.

TABLE 14-4 Dual-Port RF16x128d Register File I/O Signal List

Signal Name	Type	Flop/ Latch	Description
din[127:0]	Input	Latch	Data input
wr_en	Input	Latch	Global write enable
wr_wl[15:0]	Input	Latch	Write address
read_en	Input	Latch	Read enable
rd_wl[15:0]	Input	Latch	Read address
rclk	Input	None	L2 clock
reset_l	Input	Latch	Reset active low
se	Input	None	Scan enable
rst_tri_en	Input	None	Write disable
sehold	Input	None	Scan hold
si	Input	Flop	Scan in
dout[127:0]	Output	None	Data out
so	Output	None	Scan out

14.4 Timing Diagram

Following is an I/O read/write timing diagram for the register file 16 x 128.

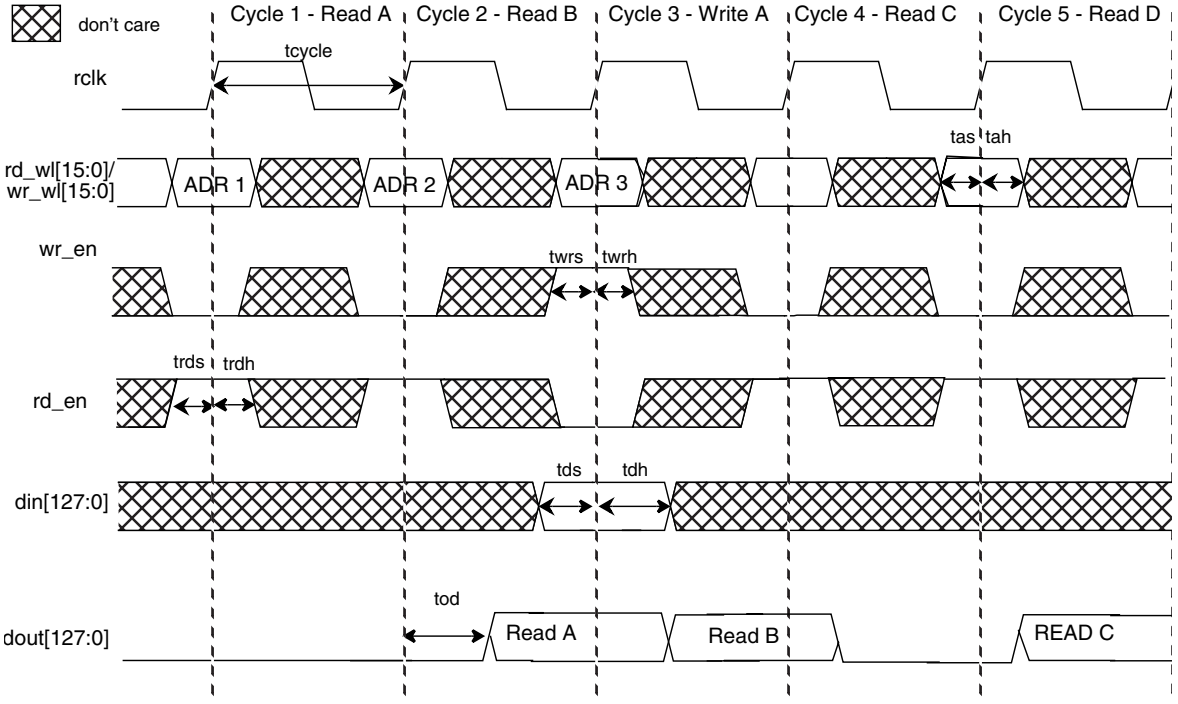

FIGURE 14-3 Read/Write Timing Diagram of RF16x128d Array

CHAPTER 15

Register File 16 x 160

This chapter describes the following topics:

15.1 Functional Description of the Block

The register file 16 x 160 (RF16x160) is a 16-entry, 160-bit-wide register file with two ports. Port A is read only. Port B is write only. The block has the following characteristics:

- No sharing or data dependency on the control mechanism between the two ports.
- Single-cycle read and write are both triggered in clock-high phase.
- Supports all combinations of consecutive reads and writes.
- If both read and write access are performed simultaneously (both on phase 1) on the same entry, the newly written data is likely to be read out, but not guaranteed.
- The high-read current memory cell is connected to a full-swing bitline with a single-ended sense amplifier.
- Write and read have two different clock pin names. Write and read may occur in different clock domains.
- Differential write mechanism used.
- The 4-bit word write-enable feature performs interleaved 40-bit wide writes.
- The 20-bit byte write-enable feature performs 8-bit wide writes.
- Write is inhibited when rst_tri_en is enabled.
- Read is *not* gated by any scan-related signals.

- All inputs are latched except for reset_l.
- Latches are transparent during the low phase of the clock and remain closed during the high phase.
- Output flops are located outside the block.

TABLE 15-1 Memory Behavior Under Combinations of read_en, write_en, reset/pwr_dn

#	rst_l	se	hd	rstri	we	re	wwl	wbl	rwl	rbl	sa	dout
1	1	0	0	0	0	0	Reset 0	Prechg	Reset 0	Prechg	Off	Last read
2	1	0	0	0	1	0	Assert 1	Toggle	Reset 0	Prechg	Off	Last read
3	1	0	0	0	0	1	Reset 0	Prechg	Assert 1	Toggle	On	New read
4	1	0	0	0	1	1	Assert 1	Toggle	Assert 1	Toggle	On	New read/ collision
5	1	1	0	1	x	x	Reset 0	Toggle	Toggle	Toggle	Toggle	Dmy read
6	1	0	1	0	1	0	Toggle	Toggle	Toggle	Toggle	Toggle	Wr access
7	1	0	1	0	0	1	Toggle	Toggle	Toggle	Toggle	Toggle	Wd access/ capture
8	0	x	x	x	x	x	Reset 0	Prechg	Reset 0	Prechg	Off	Read 0

TABLE 15-2 Top-Level Location Instantiating bw_r_rf16x160

Location	Instance Count in This Location	Instance Count in Chip
jbi	10	10
iobdg	2	2
sparc	1	8
sctag	2	8
fpu	1	1
scbuf	12	48

15.2 Block Diagrams

The following figures provide a logic symbol diagram and a functional block diagram for the register file 16 x 160.

15.2.1 RF16x160 Logic Symbol

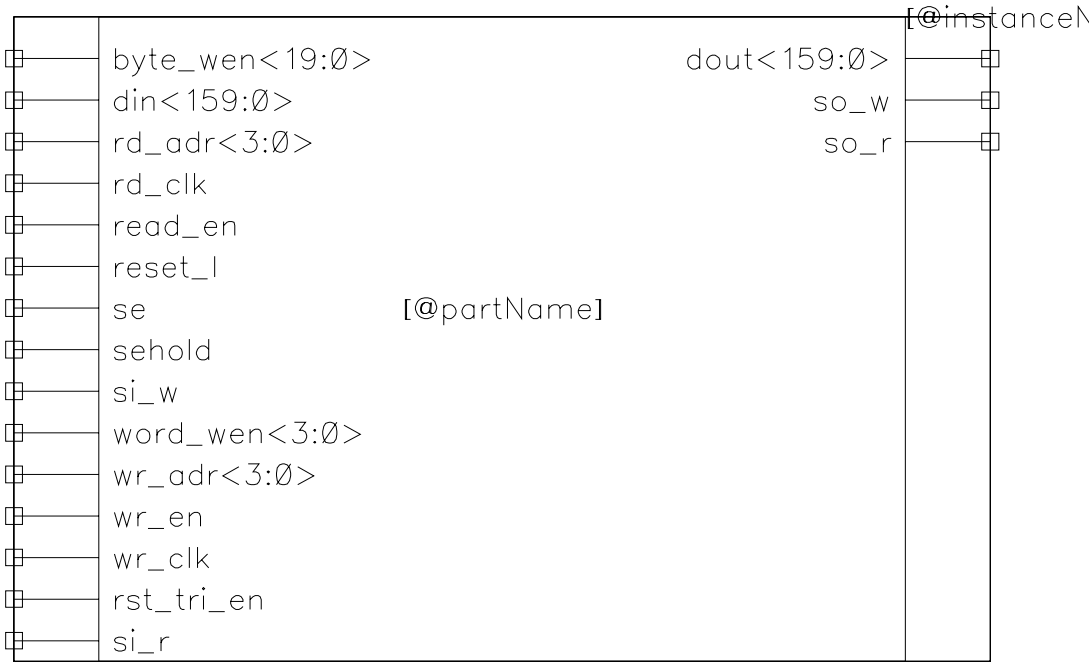

FIGURE 15-1 RF16x160 Logic Symbol

15.2.2 Functional Block Diagram of RF16x160 Array

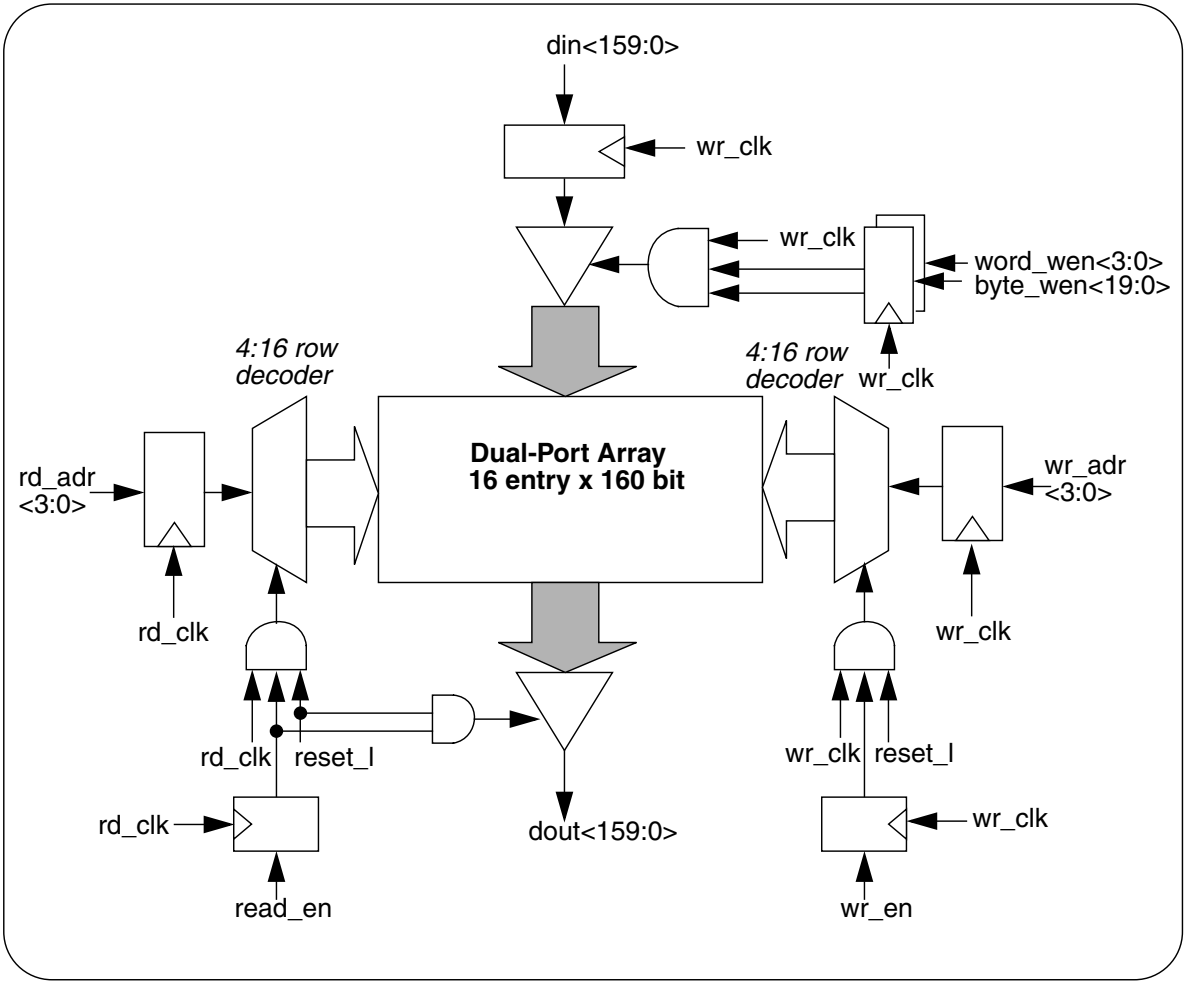

FIGURE 15-2 Functional Block Diagram of RF16x160 Array

15.3 I/O List

The following table lists the dual-port register file I/O signals.

TABLE 15-3 Dual-Port RF16x160 Register File I/O Signal List

Signal Name	Type	Flop/ Latch	Description
din[159:0]	Input	Latch	Data input
byte_wen[19:0]	Input	Latch	Byte write enable
word_wen[3:0]	Input	Latch	Word write enable
wr_en	Input	Latch	Global write enable
wr_adr[3:0]	Input	Latch	Write address
read_en	Input	Latch	Read enable
rd_adr[3:0]	Input	Latch	Read address
wr_clk	Input	None	Write clock
rd_clk	Input	None	Read clock
reset_l	Input	None	Reset active low; primary input not latched
se	Input	Latch	Scan enable; latch with free running clock
rst_tri_en	Input	Latch	Write disable; latch with free running clock
sehold	Input	Latch	Scan hold; latch with free running clock
si_w	Input	Flop	Write path scan in
si_r	Input	Flop	Read path scan in
dout[159:0]	Output	None	Data out
so_w	Output	None	Write path scan out
so_r	Output	None	Read path scan out

15.4 Timing Diagram

Following is a read/write timing diagram for the register file 16 x 160.

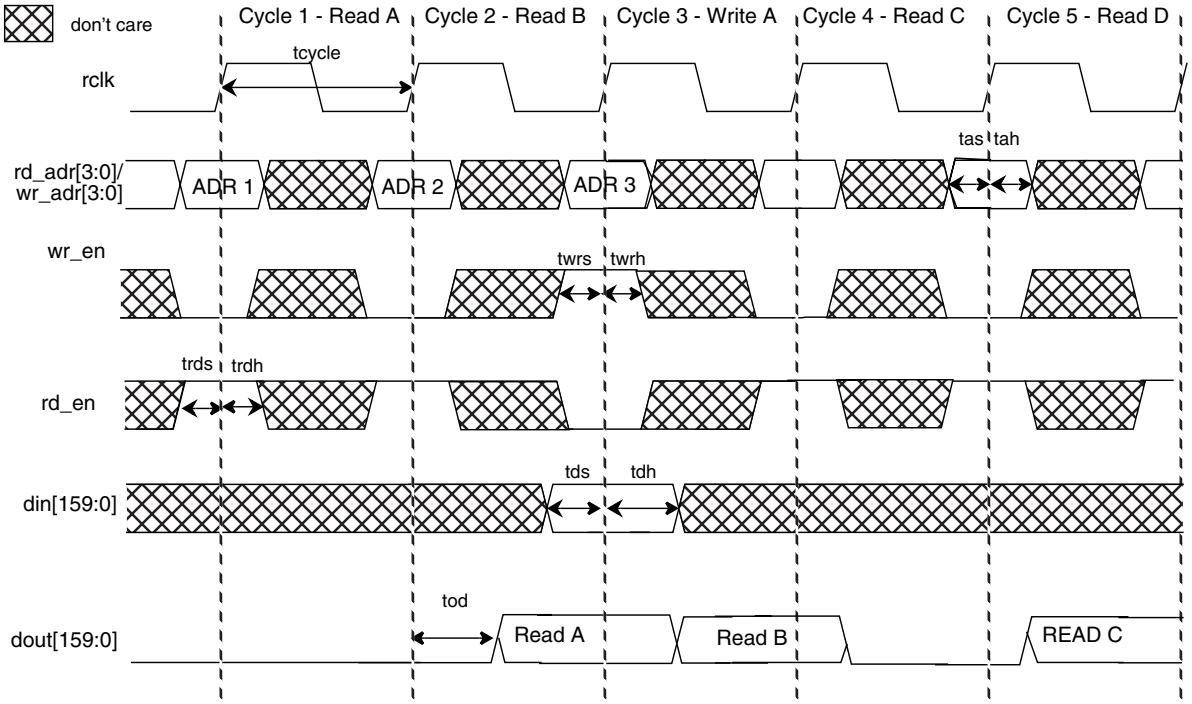

FIGURE 15-3 Read/Write Timing Diagram of RF16x160 Array

CHAPTER 16

Register File 16 x 32

This chapter describes the following topics:

16.1 Functional Description of the Block

The register file 16 x 32 (RF16x32) is a 16-entry, 32-bit-wide register file with two ports. Port A is read only. Port B is write only. The block has the following characteristics:

- No sharing or data dependency on the control mechanism between the two ports.
- Single-cycle read and write are both triggered in clock high phase.
- Supports all combinations of consecutive reads and writes.
- If both read and write access are performed simultaneously (both on phase 1) on the same entry, the newly written data is likely to be read out, but not guaranteed.
- The high read current memory cell is connected to a full-swing bitline with a single-ended sense amplifier.
- Differential write mechanism used.
- There is only one l2clk “rclk” for both read and write clock.
- A read address 2:1 mux is implemented.
- There is one bit of write address for write column 1:2 decoding and three bits of read address for read column 8:1 mux.
- The 16-bit write-enable feature performs 2-bit wide writes.
- Output is cleared to 0 when reset_l is active.

- Output is cleared to 0 when rd_en is not active.
- Write is inhibited when rst_tri_en is enabled.
- Read is *not* gated by any scan-related signals.
- All inputs are latched, except for reset_l.
- Latches are transparent during the low phase of clock and remain closed during the high phase.
- Output flops are located outside the block.
- During write/read contention, the functions listed in TABLE 16-1: are needed.

TABLE 16-1 Write/Read Contention Functions

Mem Old Content	New din	dout
0	0	0
0	1	x
1	0	0
1	1	1

Note – From the above table, even with rd/wr contention, always have write through for all cases.

TABLE 16-2 Memory Behavior Under Combinations of read_en, write_en, and reset/pwr_dn

#	rst_l	se	hd	rstri	we	re	wwl	wbl	rwl	rbl	sa	dout
1	1	0	0	0	0	0	Reset 0	Prechg	Reset 0	Prechg	Off	Read 0
2	1	0	0	0	1	0	Assert 1	Toggle	Reset 0	Prechg	Off	Read 0
3	1	0	0	0	0	1	Reset 0	Prechg	Assert 1	Toggle	On	New read
4	1	0	0	0	1	1	Assert 1	Toggle	Assert 1	Toggle	On	New read/ collision
5	1	1	0	1	x	x	Reset 0	Toggle	Toggle	Toggle	Toggle	Dmy read
6	1	0	1	0	1	0	Toggle	Toggle	Toggle	Toggle	Toggle	Write access
7	1	0	1	0	0	1	Toggle	Toggle	Toggle	Toggle	Toggle	Readd access/ capture
8	0	x	x	x	x	x	Reset 0	Prechg	Reset 0	Prechg	Off	Read 0

TABLE 16-3 Top-Level Location Instantiating bw_r_rf16x32

Location	Instance Count in This Location	Instance Count in Chip
sparc	2	16

16.2 Block Diagrams

The following figures provide a logic symbol diagram and a functional block diagram for the register file 16 x 32.

16.2.1 RF16x32 Logic Symbol

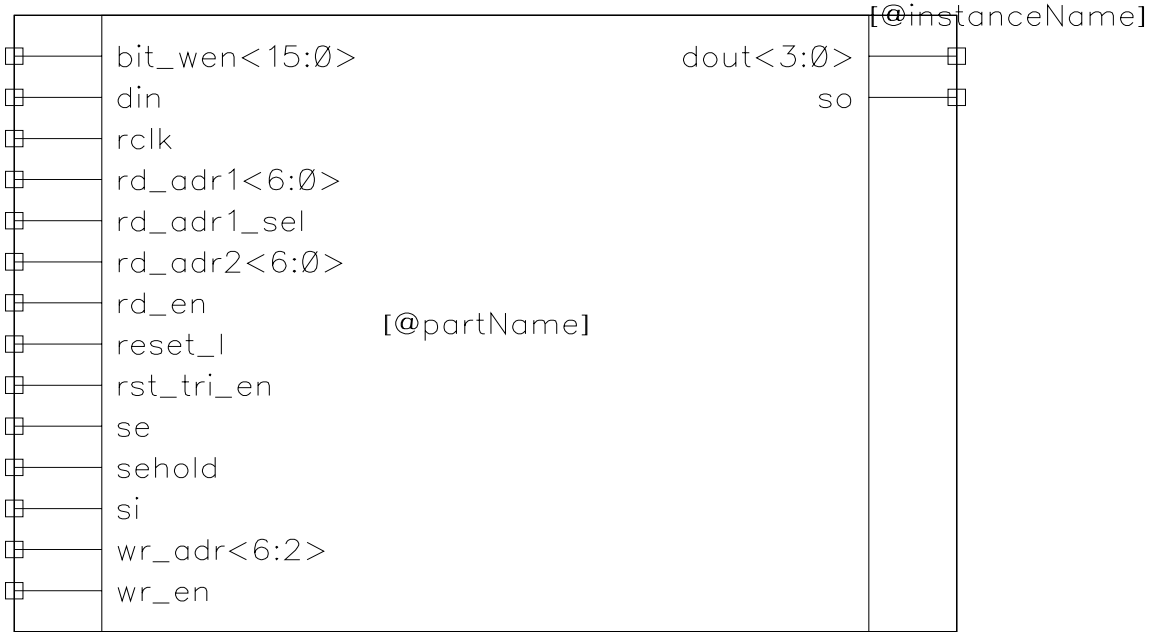

FIGURE 16-1 RF16x32 Logic Symbol

16.2.2 Functional Block Diagram of RF16x32 Array

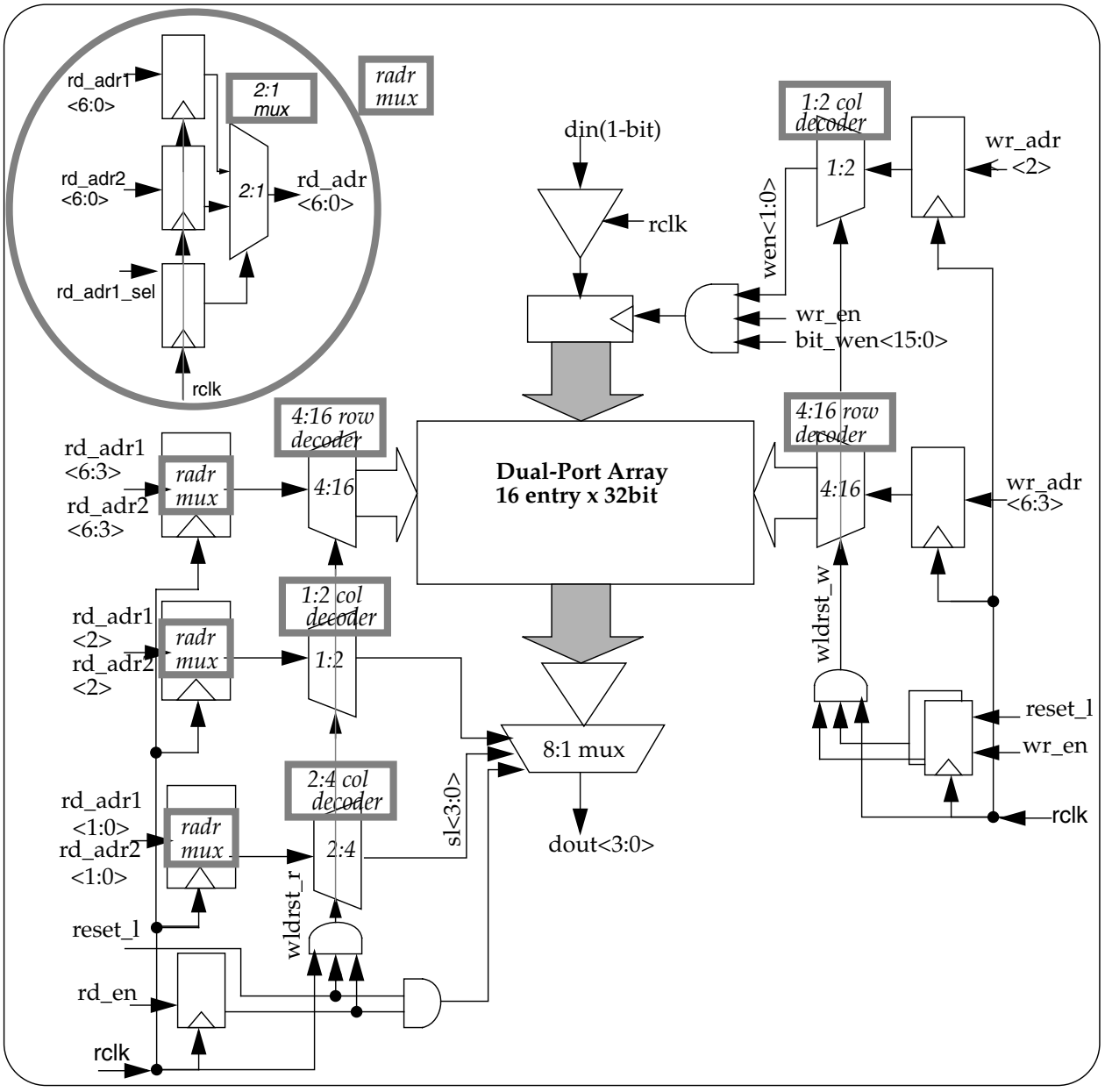

FIGURE 16-2 Functional Block Diagram of RF16x32 Array

16.3 I/O List

The following table lists the dual-port register file I/O signals.

TABLE 16-4 Dual-Port RF16x32 Register File I/O Signal List

Signal Name	Type	Flop/ Latch	Description
din	Input	Latch	Data input
bit_wen[15:0]	Input	Latch	Bit write enable
wr_en	Input	Latch	Global write enable
wr_adr[6:2]	Input	Latch	Write address
rd_en	Input	Latch	Read enable
rd_adr1[6:3]	Input	Latch	Read address 1
rd_adr2[6:3]	Input	Latch	Read address 2
rd_adr1[2:0]	Input	Flop	Read address 1
rd_adr2[2:0]	Input	Flop	Read address 2
rd_adr1_sel	Input	Latch for rdadr<6:3>/ flop for rdadr<2:0>	Read address 1 select
rclk	Input	None	L2 clock
reset_l	Input	Latch	Reset active low
se	Input	None	Scan enable
rst_tri_en	Input	None	Write disable
sehold	Input	None	Scan hold
si	Input	Flop	Scan in
dout[3:0]	Output	None	Data out
so	Output	None	Scan out

16.4 Timing Diagram

The following figure shows an I/O read/write timing diagram.

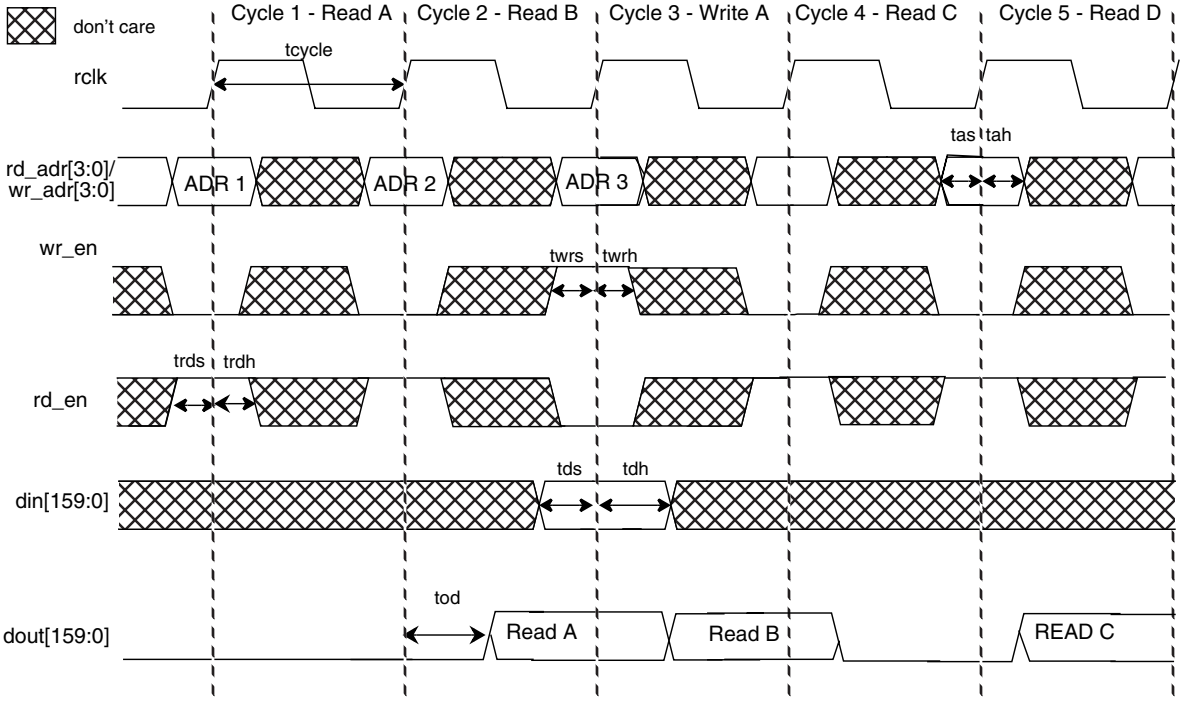

FIGURE 16-3 Read/Write Timing Diagram of RF16x32 Array

CHAPTER 17

Register File 32 x 108

This chapter describes the following topics:

17.1 Functional Description of the Block

The register file 32 x 108 (RF32x108) is a 32-entry, 108-bit-wide register file with two ports. Port A is read only. Port B is write only. The block has the following characteristics:

- No sharing or data dependency on the control mechanism between the two ports.
- Single-cycle read and write, both triggered in clock high phase.
- Supports all combinations of consecutive reads and writes.
- If both read and write access are performed simultaneously (both on phase 1) on the same entry, the newly written data is likely to be read out, but not guaranteed.
- The high read current memory cell is connected to a full-swing bitline with a single-ended sense amplifier.
- Differential write mechanism used.
- The 4-bit word write-enable feature performs interleaved 27-bit wide writes.
- Write is inhibited when rst_tri_en is enabled.
- Read is *not* gated by any scan-related signals.
- All inputs are latched, except for reset_l.
- Latches are transparent during the low phase of clock and remain closed during the high phase.

- Output flops are located outside the block.

TABLE 17-1 Memory Behavior Under Combinations of read_en, write_en, reset/pwr_dn

#	rst_l	se	hold	rstri	we	re	wwl	wbl	rwl	rbl	sa	dout
1	1	0	0	0	0	0	Reset 0	Prechg	Reset 0	Prechg	Off	Last read
2	1	0	0	0	1	0	Assert 1	Toggle	Reset 0	Prechg	Off	Last read
3	1	0	0	0	0	1	Reset 0	Prechg	Assert 1	Toggle	On	New read
4	1	0	0	0	1	1	Assert 1	Toggle	Assert 1	Toggle	On	New read/ collision
5	1	1	0	1	x	x	Reset 0	Toggle	Toggle	Toggle	Toggle	Nmy read
6	1	0	1	0	1	0	Toggle	Toggle	Toggle	Toggle	Toggle	Nr access
7	1	0	1	0	0	1	Toggle	Toggle	Toggle	Toggle	Toggle	Rd access/ capture
8	0	x	x	x	x	x	Reset 0	Prechg	Reset 0	Prechg	Off	Read 0

TABLE 17-2 Top-Level Location Instantiating bw_r_rf32x108

Location	Instance Count in This Location	Instance Count in Chip
sctag	16	64

17.2 Block Diagrams

The following figures provide a logic symbol diagram and a functional block diagram for the register file RF32x108.

17.2.1 RF32x108 Logic Symbol

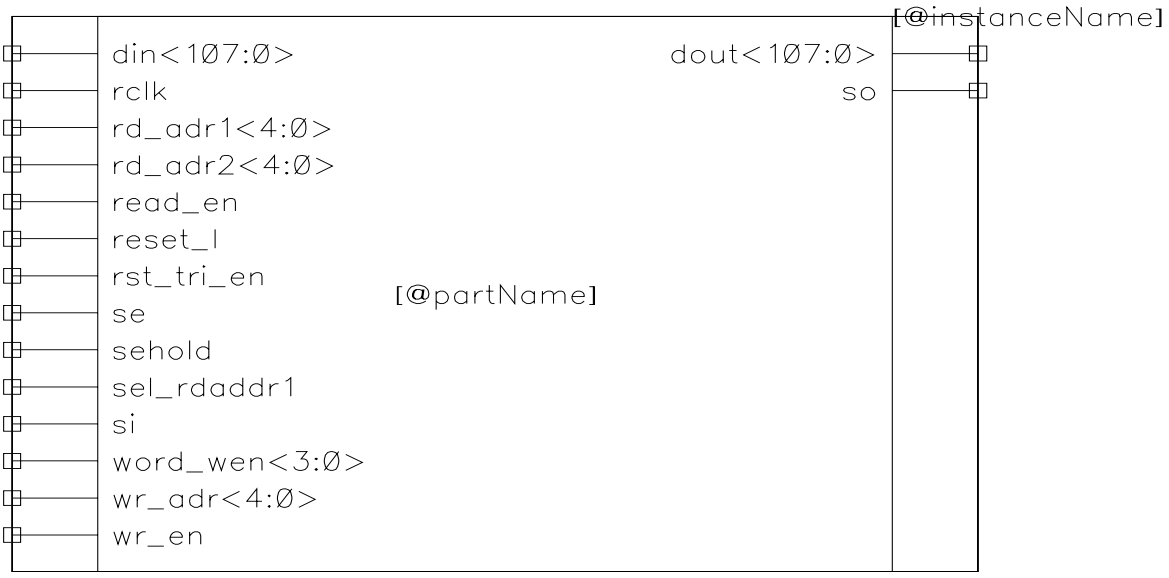

FIGURE 17-1 RF32x108 Logic Symbol

17.2.2 Functional Block Diagram of RF32x108 Array

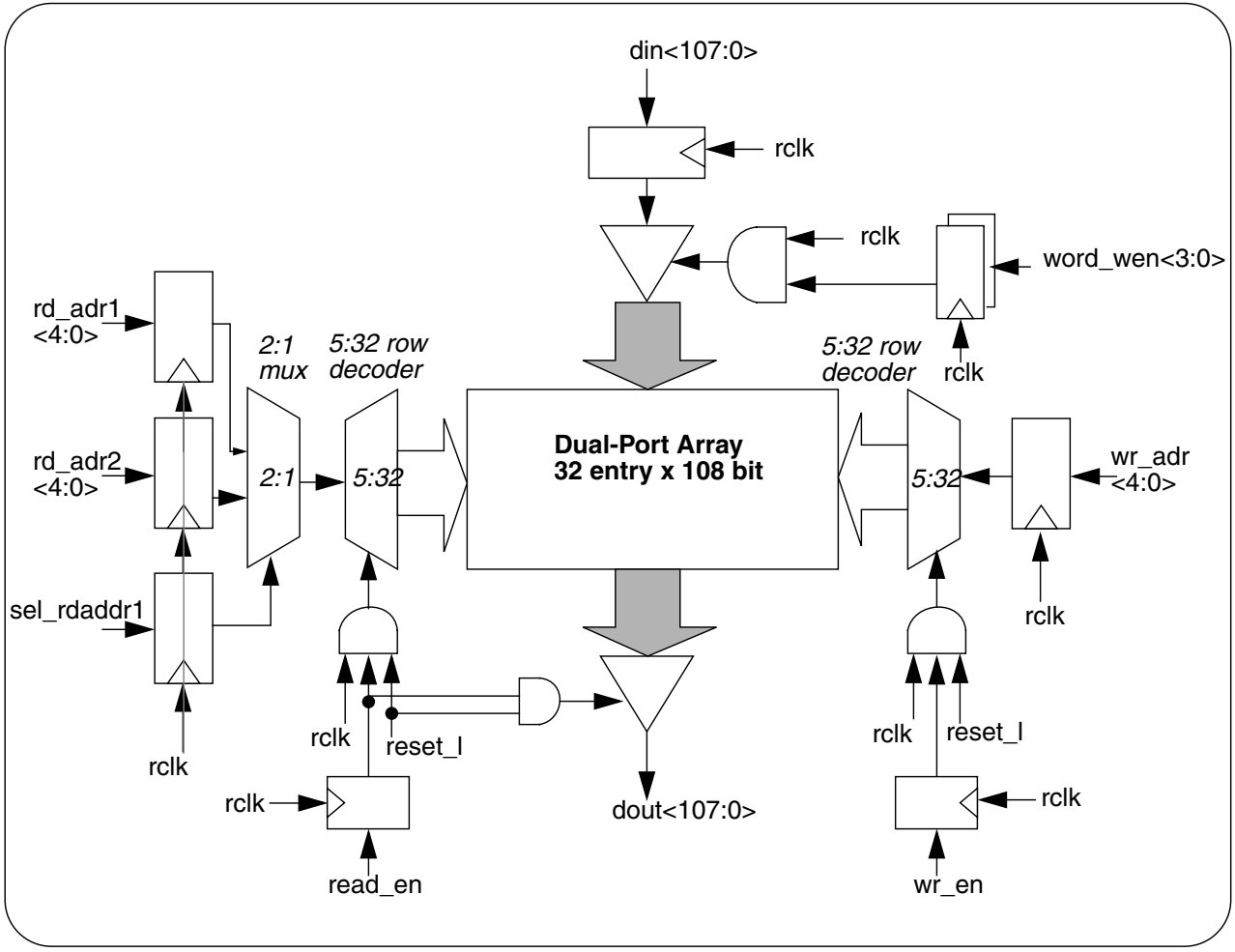

FIGURE 17-2 Functional Block Diagram of RF32x108 Array

17.3 I/O List

The following table lists the dual-port register file I/O signals.

TABLE 17-3 Dual-Port RF32x108 Register File I/O Signal List

Signal Name	TYPE	Flop/ Latch	Description
din[107:0]	Input	Latch	Data input
word_wen[3:0]	Input	Latch	Word write enable
wr_en	Input	Latch	Global write enable
wr_adr[4:0]	Input	Latch	Write address
read_en	Input	Latch	Read enable
rd_adr1[4:0]	Input	Latch	Read address 1
rd_adr2[4:0]	Input	Latch	Read address 2
sel_rdaddr1	Input	Latch	Read address select
rclk	Input	None	L2 clock
reset_l	Input	Latch	Reset active low
se	Input	None	Scan enable
rst_tri_en	Input	None	Write disable
sehold	Input	None	Scan hold
si	Input	Flop	Scan in
dout[107:0]	Ouput	None	Data out
so	Output	None	Scan out

17.4 Timing Diagram

Following is an I/O read/write timing diagram for the RF32x108 array.

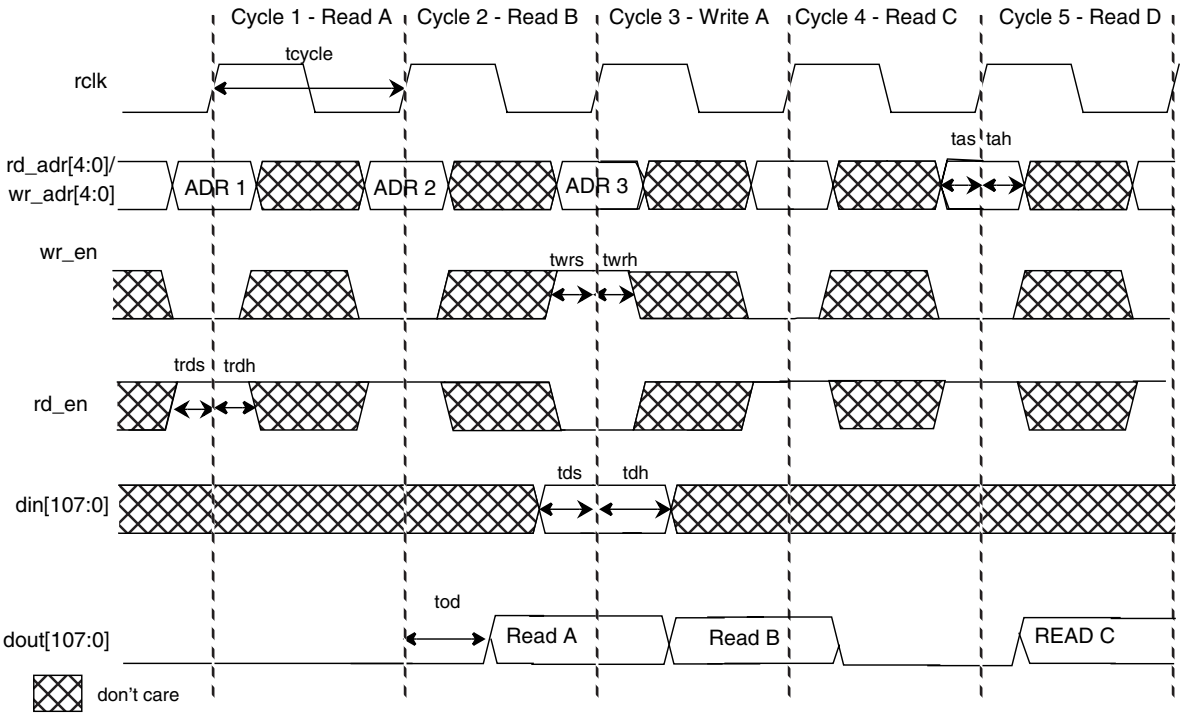

FIGURE 17-3 Read/Write Timing Diagram of RF32x108 Array

CHAPTER 18

Register File 32 x 152

This chapter describes the following topics:

18.1 Functional Description of the Block

The register file 32 x 152 (RF32x152b) is a 32-entry, 152-bit-wide register file with two ports. Port A is read only. Port B is write only. The block has the following characteristics:

- No sharing or data dependency on the control mechanism between the two ports.
- Single-cycle read and write, both triggered in clock high phase.
- Supports all combinations of consecutive reads and writes.
- Supports simultaneous read and write operations in a cycle.
- "Write through" occurs during read/write contention, although this is not a requirement.
- The high read current memory cell is connected to a full-swing bitline with a single-ended sense amplifier.
- Differential write mechanism is used.
- Write is inhibited when rst_tri_en is enabled.
- Read is gated by se.
- All inputs are latched, except se, sehold, rst_tri_en, reset_l.
- Latches are transparent during the low phase of the clock and remain closed during the high phase.

- Output flops are located outside the block.

TABLE 18-1 Functional Modes of Operation (se=0, sehold=0)

#	reset_l	rst_tri_en	we	re	wwl	wbls	rwl	rbl	sa	Operation
1	1	X	X	0	X	X	Reset 0	Prechg	off	Last read
2	1	X	X	1	X	X	Assert 1	Toggle	on	New read
3	1	X	0	X	Reset 0	Toggle	X	X	X	No write
5	1	1	X	X	Reset 0	Toggle	X	X	X	No write
6	1	0	1	X	Assert 1	Toggle	X	X	X	New write
7	1	0	1	1	Assert 1	Toggle	Assert 1	Toggle	on	New read/new write*
8	0	X	X	X	Reset 0	Prechg	Reset 0	Prechg	off	Read 0

* In this case, if the read address and write address are the same, write through occurs.

TABLE 18-2 Blocks Instantiating bw_r_rf32x152b

Block Name	Location	Logical Org (entry x word x way)	Physical Org (row x column)	Instance Count	Instance Count in Chip
dfq	lsu	32x152x1	32x152b	1	8

18.2 Block Diagrams

The following figures provide a logic symbol diagram and a functional block diagram for the register file 32 x 152.

18.2.1 RF32x152b Logic Symbol

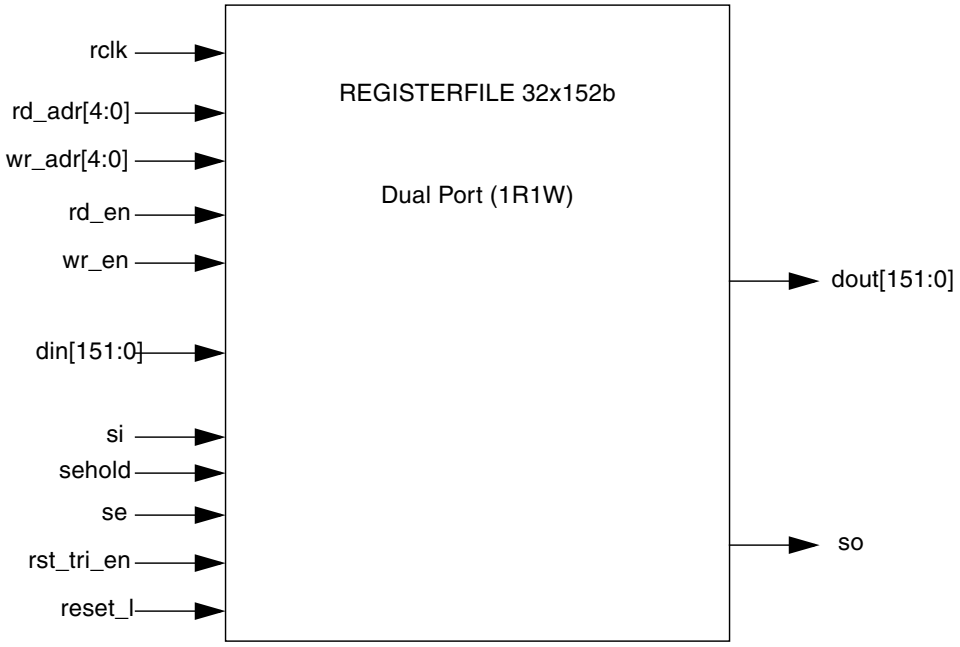

FIGURE 18-1 RF32x152b Logic Symbol

18.2.2 Functional Block Diagram of RF32x152b Array

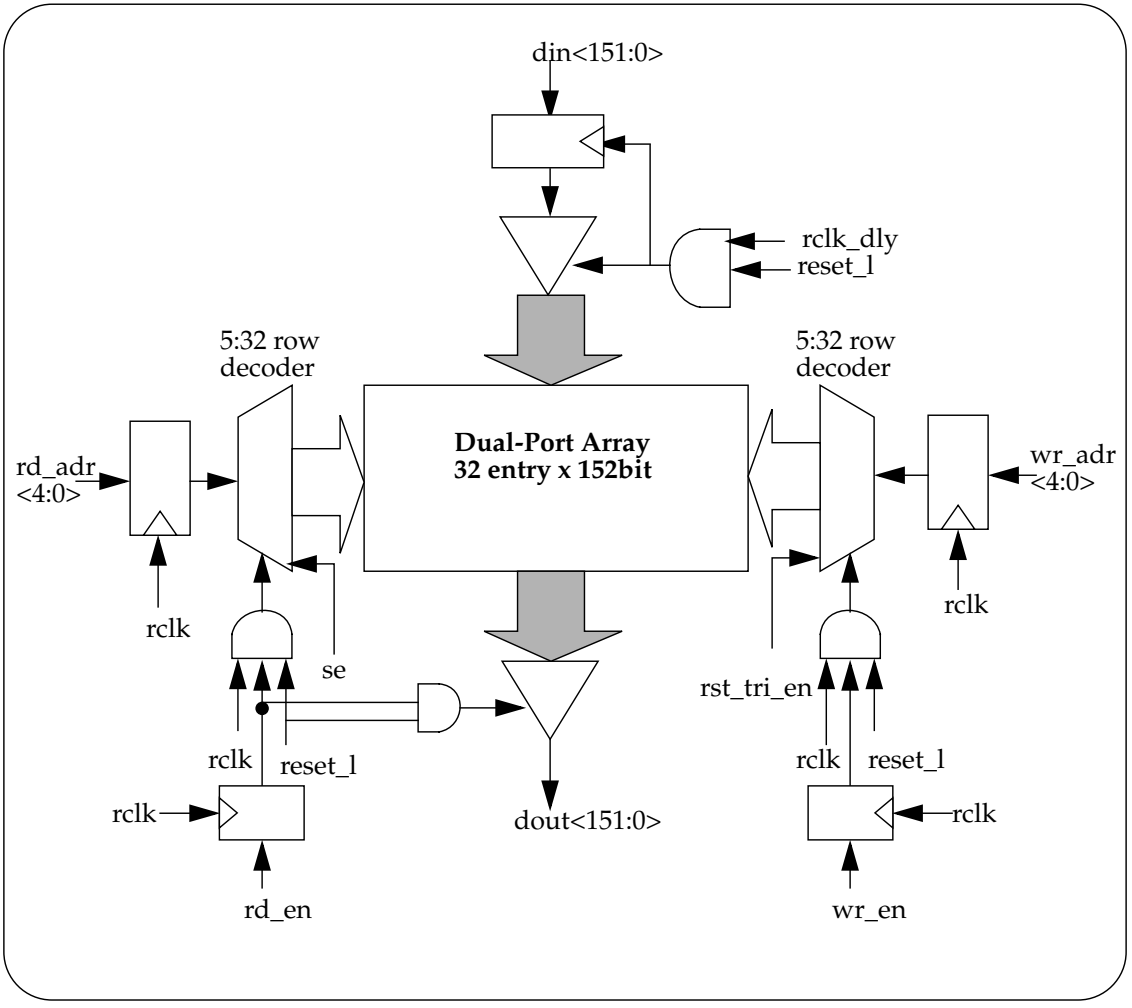

FIGURE 18-2 Functional Block Diagram of RF32x152b Array

18.3 I/O List

The following table lists the dual-port register file I/O signals.

TABLE 18-3 RF32x152b I/O Signal List Descriptions

Signal Name	TYPE	Flop/ Latch	Description
din[151:0]	Input	Latch	Data input
wr_en	Input	Latch	Global write enable
wr_adr[4:0]	Input	Latch	Write address
rd_en	Input	Latch	Read enable
rd_adr[4:0]	Input	Latch	Read address
rclk	Input	None	L2 clock
reset_l	Input	None	Reset active low
se	Input	None	Scan enable
rst_tri_en	Input	None	Write disable
sehold	Input	None	Scan hold
si	Input	Flop	Scan in
dout[151:0]	Output	None	Sata out
so	Output	None	Scan out

18.4 Timing Diagram

Following is an I/O read/write timing diagram for the RF32x152b register file.

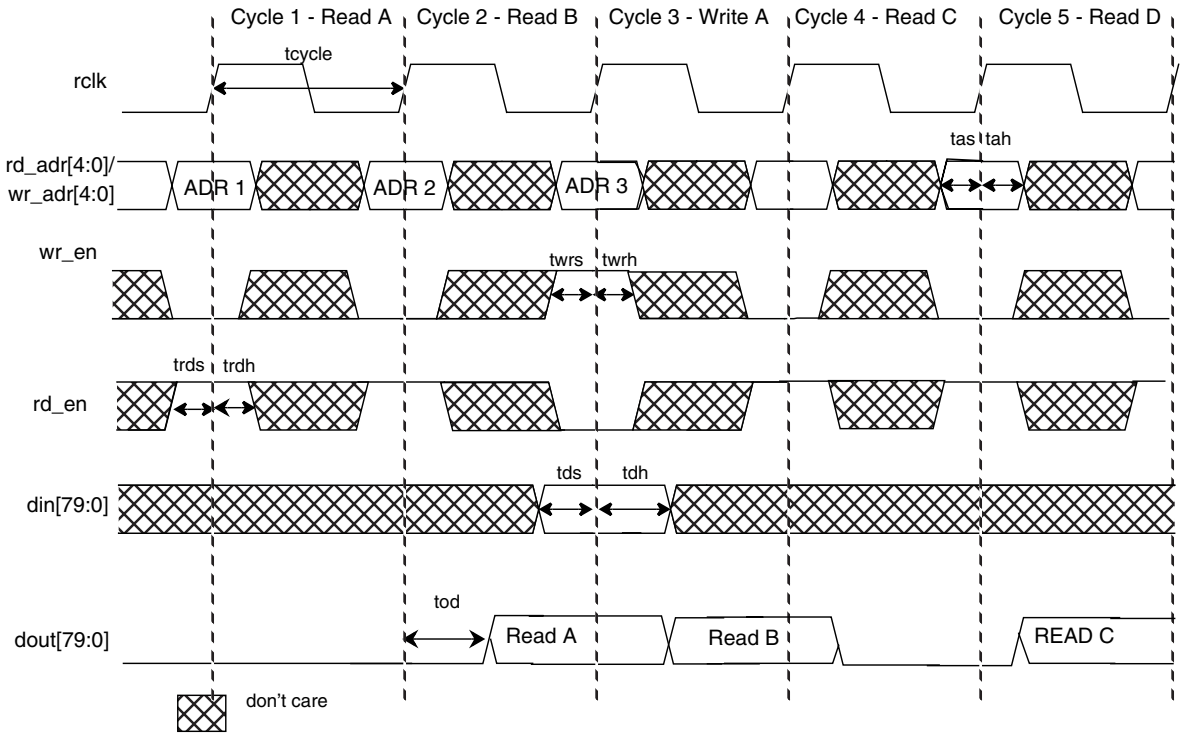

FIGURE 18-3 Read/Write Timing Diagram of RF32x152b Array

CHAPTER 19

Register File 32 x 80

This chapter describes the following topics:

19.1 Functional Description of the Block

The register file 32 x 80 (RF32x80) is a 32-entry, 80-bit-wide register file with two ports. Port A is read only. Port B is write only.

- No sharing or data dependency on the control mechanism between the two ports.
- Single-cycle read and write, both triggered in clock high phase.
- Supports all combinations of consecutive reads and writes.
- Supports simultaneous read and write operations in a cycle.
- "Write through" occurs during read/write contention, although this is not a requirement.
- The high read current memory cell is connected to a full-swing bitline with a single-ended sense amplifier.
- Differential write mechanism is used.
- The 20-bit nibble write-enable feature performs 4-bit-wide writes.
- Write is inhibited when rst_tri_en is enabled.
- Read is gated by se.
- All inputs are latched, except se, sehold, rst_tri_en, reset_l.
- Latches are transparent during the low phase of clock and remain closed during the high phase.

- Output flops are located outside the block.

TABLE 19-1 Functional Modes of Operation (se=0, sehold=0)

#	reset_l	rst_tri_en	nib_wr_en	we	re	wwl	wbls	rwl	rbl	sa	Operation
1	1	X	X	X	0	X	X	Reset 0	Prechg	Off	Last read
2	1	X	X	X	1	X	X	Assert 1	Toggle	On	New read
3	1	X	X	0	X	Reset 0	X	X	X	X	No write
4	1	X	0	X	X	X	Prechg	X	X	X	No write
5	1	1	X	X	X	Reset 0	X	X	X	X	No write
6	1	0	1	1	X	Assert 1	Toggle	X	X	X	New write
7	1	0	1	1	1	Assert 1	Toggle	Assert 1	Toggle	On	New read/new write*
8	0	X	X	X	X	Reset 0	Prechg	Reset 0	Prechg	Off	Read 0

* In this case, if the read address and write address are the same, write through occurs

TABLE 19-2 Blocks Instantiating bw_r_rf32x80

Block Name	Location	Logical Org (entry x word x way)	Physical Org (row x column)	Instance Count	Instance Count in the Chip	Special Features
scpd	tlu	8 x 64 x 4	32 x 80	1	8	nibble_wr_en
tsa0	tlu	26 x 79	32 x 80	1	8	nibble_wr_en
tsa1	tlu	26 x 64	32 x 80	1	8	nibble_wr_en
stb_data	lsu	32 x 80 x 1	32 x 80	1	8	None

19.2 Block Diagrams

The following figures provide a logic symbol diagram and a functional block diagram for the RF32x80 register file.

19.2.1 RF32x80 Logic Symbol

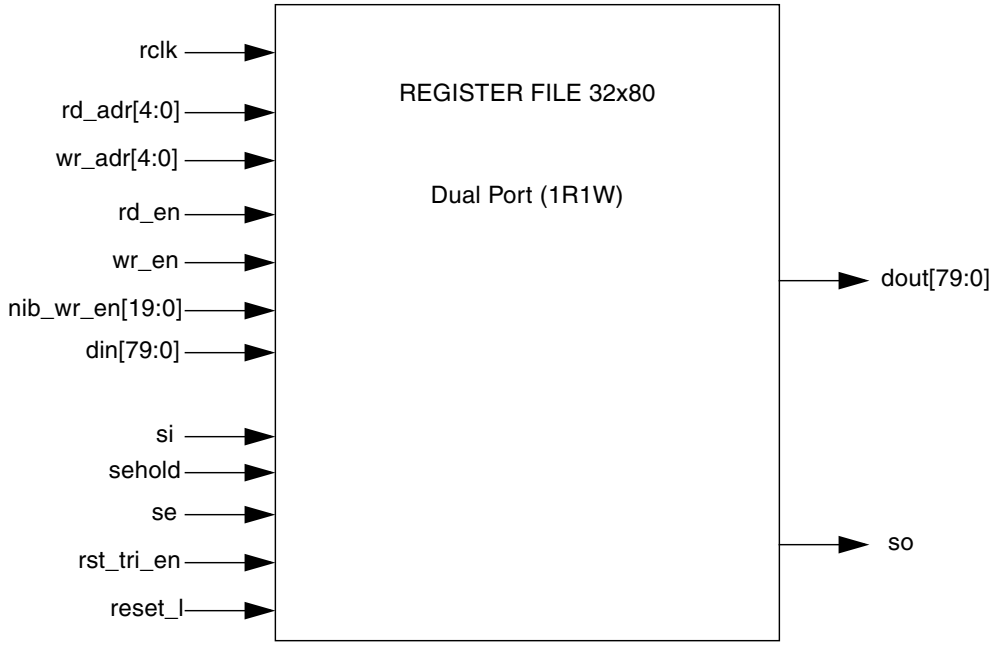

FIGURE 19-1 RF32x80 Logic Symbol

19.2.2 Functional Block Diagram of RF32x80 Array

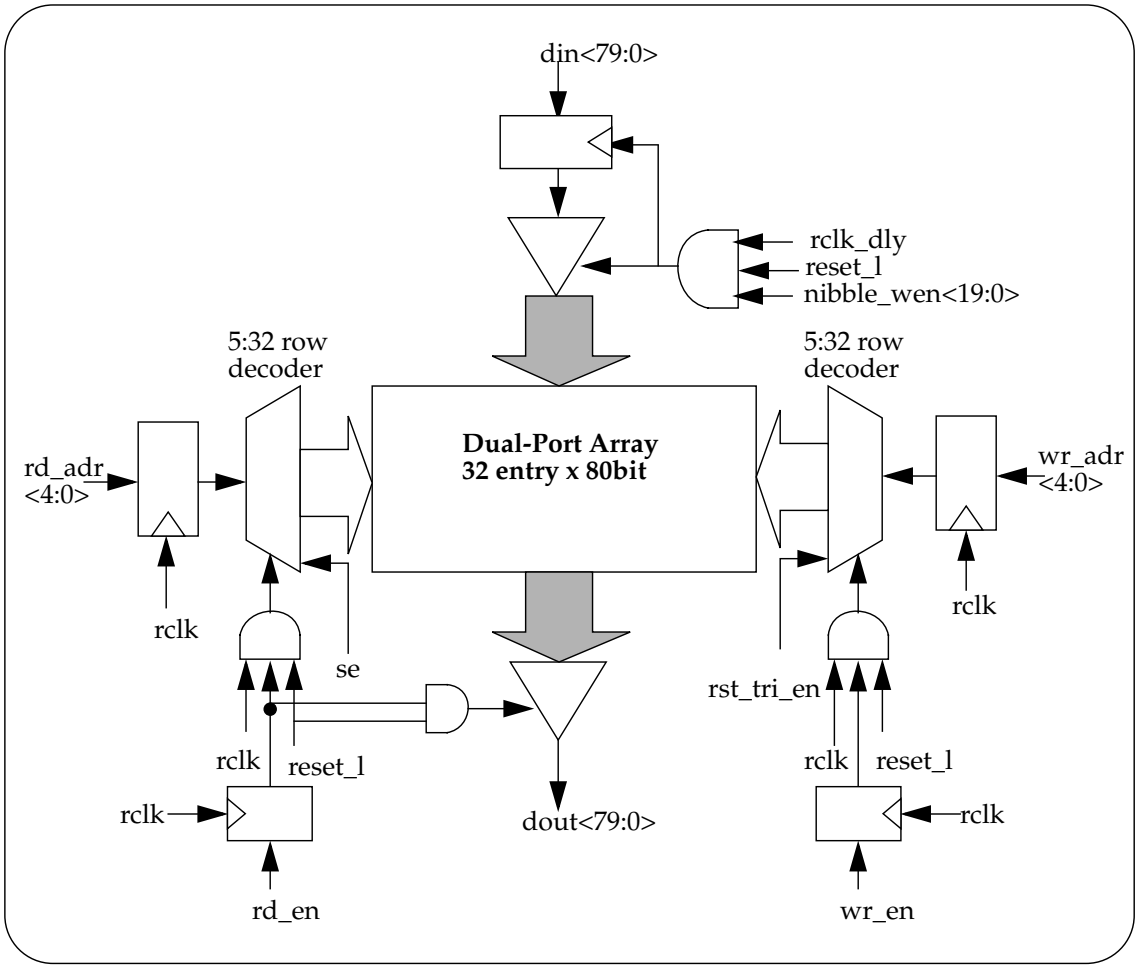

FIGURE 19-2 Functional Block Diagram of RF32x80 Array

19.3 I/O List

The following table lists the dual-port register file I/O signals.

TABLE 19-3 RF32x80 I/O Signal List

Signal Name	TYPE	Flop/ Latch	Description
din[79:0]	Input	Latch	Data input
nibble_wr_en[19:0]	Input	Latch	Nibble write enable
wr_en	Input	Latch	Global write enable
wr_adr[4:0]	Input	Latch	Write address
rd_en	Input	Latch	Read enable
rd_adr[4:0]	Input	Latch	Read address
rclk	Input	None	L2 clock
reset_l	Input	None	Reset active low
se	Input	None	Scan enable
rst_tri_en	Input	None	Write disable
sehold	Input	None	Scan hold
si	Input	Flop	Scan in
dout[79:0]	Output	None	Data out
so	Output	None	Scan out

19.4 Timing Diagram

Following is an I/O read/write timing diagram for the RF32x80 register file.

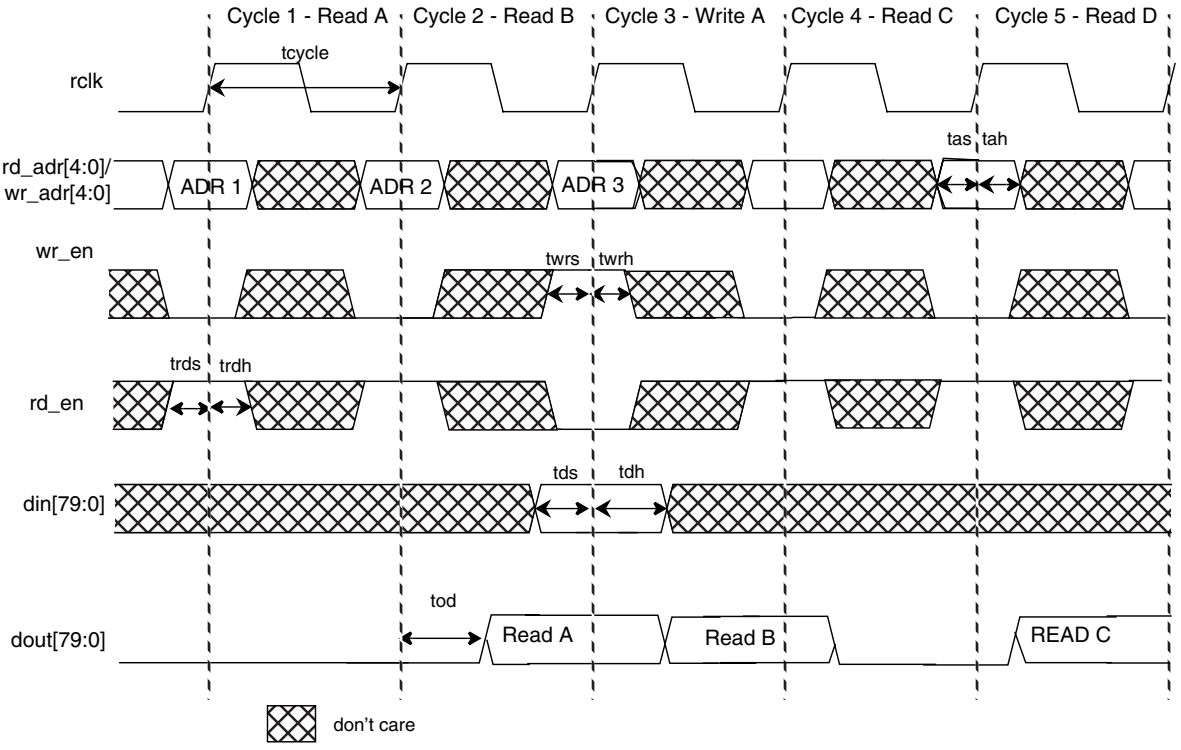

FIGURE 19-3 Read/Write Timing Diagram of RF32x80 Array

CHAPTER 20

Register Files 16x65 and 16x81

This chapter describes the following topics:

20.1 Functional Description

The register files bw_rf_16x81 and bw_rf_16x65 are small register files which have 16 entries and are 65 bits or 81 bits wide, respectively. The register files have the following characteristics:

- Dual-port cells are used to construct the array.
- Read is single ended. Write is differential.
- Read operation is on phase1 (clock high).
- Write is on phase2 (clock low) of the clock cycle.
- The register file can operate at multiple frequencies. For example, read can operate at 200 MHz and write can operate at 1.4 GHz simultaneously.
- Collision (read and write to the same address) is not enabled for asynchronous read/write.

20.1.1 Uses of the Register Files

The following table identifies the uses of register files 16x65 and 16x81 in the I/O bridge (IOB), dynamic random-access memory (DRAM) and the J-Bus Interface (JBI).

TABLE 20-1 Uses of 16x65 and 16x81 Register Files

Cluster	Memory	Register File	Instance Count	Read/Write Frequency
IOB	cpu_buf	16x65	2	200 MHz/1.4 GHz
IOB	mondo_data	16x65	4	1.4 GHz/1.4 GHz
IOB	l2_dbg_buf	16x65	2	200 MHz/1.4 GHz
DRAM	data_write_Q	16x65	4	200 MHz/1.4 GHz
JBI	rhq_buf	16x65	1	1.4 GHz/200 MHz
JBI	dbg_buf	16x65	4	200 MHz/200 MHz
JBI	prtq_buf	16x65	1	1.4 GHz/200 MHz
JBI	prrq_buf	16x81	1	200 MHz/200 MHz
JBI	prtq_buf	16x81	1	200 MHz/200 MHz

20.1.2 Read and Write Operation

A register file has dual ports (one read, one write), so the ports can be accessed simultaneously. Read and write can occur in a single cycle – read is on phase1 and write is on phase2.

20.1.2.1 Read Access

Register file read is on phase1. To access the register file, read address and read enable must meet setup time before the rising edge of read clock. Data out is available after clock rising and the value is held until the next rising edge of the read clock. When the read clock is low, read wordline turns off, internal read bitlines are precharged to vdd, and output latches are turned off to block internal precharged values from propagating to the output pin.

20.1.2.2 Write Access

Register file write is on phase2. To write to the register file, write address, write enable, and data input must meet setup time before the rising edge of write clock (phase1). But write does not start until write clock goes low (phase2). When write clock goes high, write is done and write bitlines (bl, blb) are precharged back to vdd.

20.1.2.3 Read and Write Collision

The read and write clock determines collision status as follows:

- If read and write clock are running at the same frequency, collision is enabled.
- If read and write clock are running at different frequencies, collision is not enabled.

20.1.3 Functional Truth Table

TABLE 20-2 Functional Truth Table

csn_rd	csn_wr	Description
0	0	Read is on phase1. Write is on phase2.
0	1	Read is on phase 1, no write.
1	0	Write on phase2, no read.
1	1	No write, no read.

20.2 Block Diagram

FIGURE 20-1 provides a functional block diagram of the bw_rf_16x65 register file.

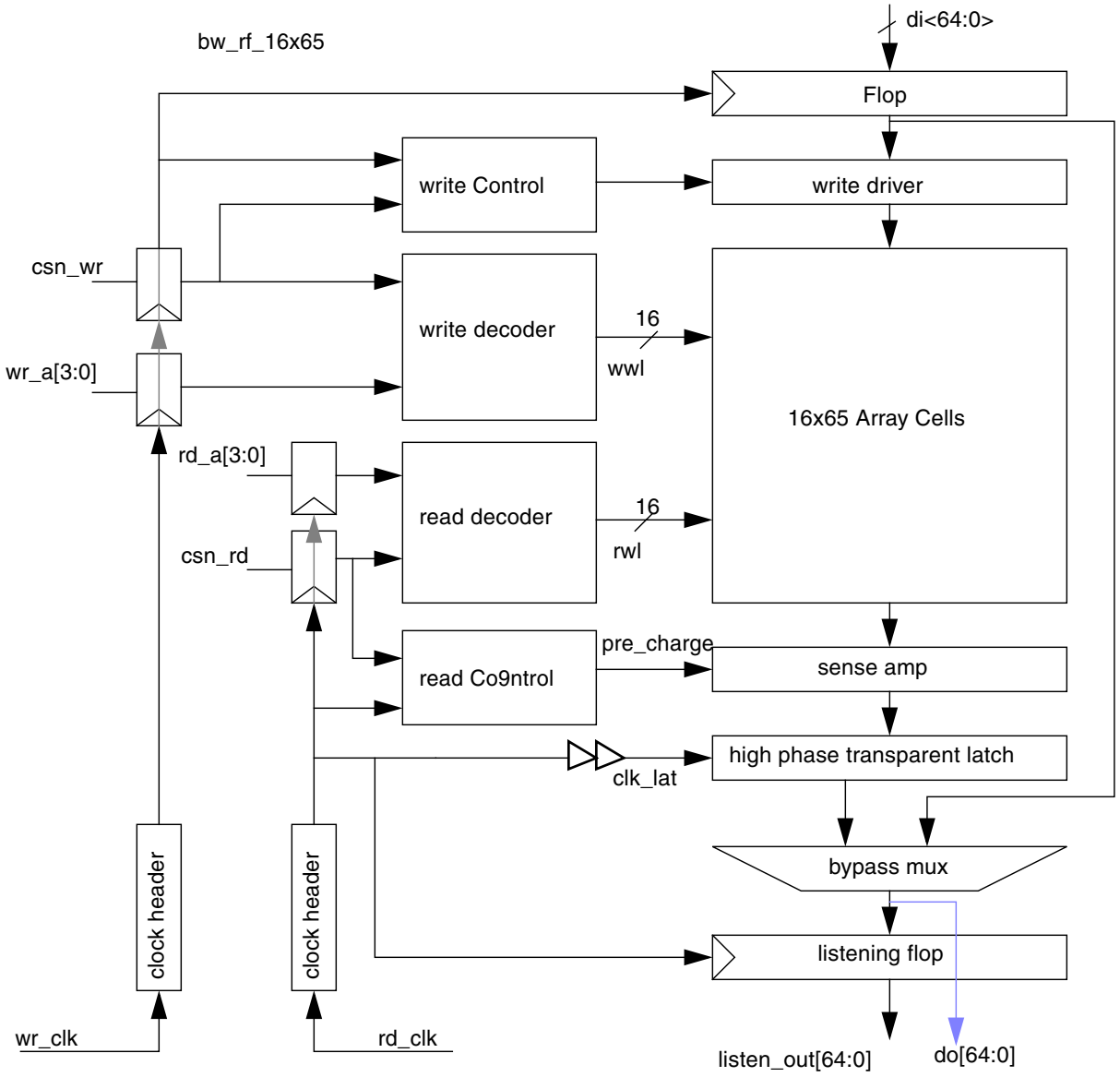

FIGURE 20-1 Functional Block Diagram of bw_rf_16x65

20.3 I/O Lists

The following table describes the bw_rf_16x81 I/O signals.

TABLE 20-3 bw_rf_16x81 I/O Signal List

Signal Name	Description	Type	Flop	Clock
csn_w	Active low write enable.	Input	Yes	wr_clk
csn_rd	Active low read enable.	Input	Yes	rd_clk
wr_a[3:0]	Write address.	Input	Yes	wr_clk
rd_a[3:0]	Read address.	Input	Yes	rd_clk
scan_en	Scan enable.	Input	No	rd_clk, wr_clk
testmux_sel	High indicates bypass array output.	Input	No	
hold	Hold is on as changing from scan to functional to test array.	Input	No	
di[80:0]	Data inputs.	Input	Yes	wr_clk
margin[4:0]	margin[1:0] controls rising edge of rwl . margin[3:2] controls rising edge of wwl. margin[4] turn on to provide extra keeper on rbl.	Input	No	
do[80:0]	Data outputs.	Output	No	
listen_out[80:0]	Outputs of listening flops.	Output	Yes	
rd_clk	Read clock.	Input	No	
wr_clk	Write clock.	Input	No	
si	Scan in.	Input	Yes	smclk_w
so	Scan out.	Output	No	rd_clk

The following table describes the bw_rf_16x65 I/O signals.

TABLE 20-4 bw_rf_16x65 I/O Signal List

Signal Name	Description	Type	Flop	Clock
csn_wr	Active low write enable.	Input	Yes	wr_clk
csn_rd	Active low read enable.	Input	Yes	rd_clk
wr_a[3:0]	Write address.	Input	Yes	wr_clk

TABLE 20-4 bw_rf_16x65 I/O Signal List *(Continued)*

Signal Name	Description	Type	Flop	Clock
rd_a[3:0]	Read address.	Input	Yes	rd_clk
scan_en	Scan enable.	Input	No	rd_clk, wr_clk
testmux_sel	High indicates bypass array output.	Input	No	
hold	Hold is on as changing from scan to functional to test array.	Input	No	
di[64:0]	Data inputs.	Input	Flop	wr_clk
margin[4:0]	margin[1:0] controls rising edge of rwl. margin[3:2] controls rising edge of wwl. margin[4] turn on to provide extra keeper on rbl.	Input	No	
do[64:0]	Data outputs.	Output	No	
listen_out[64:0]	Outputs of listening flops.	Output	Yes	
rd_clk	Read clock.	Input	No	
wr_clk	Write clock.	Input	No	
si	Scan in.	Input	Yes	smclk_w
so	Scan out.	Output	No	rd_clk

20.4 Timing Diagrams

The following figures provide read and write timing diagrams for the register files.

20.4.1 Read Timing Diagram

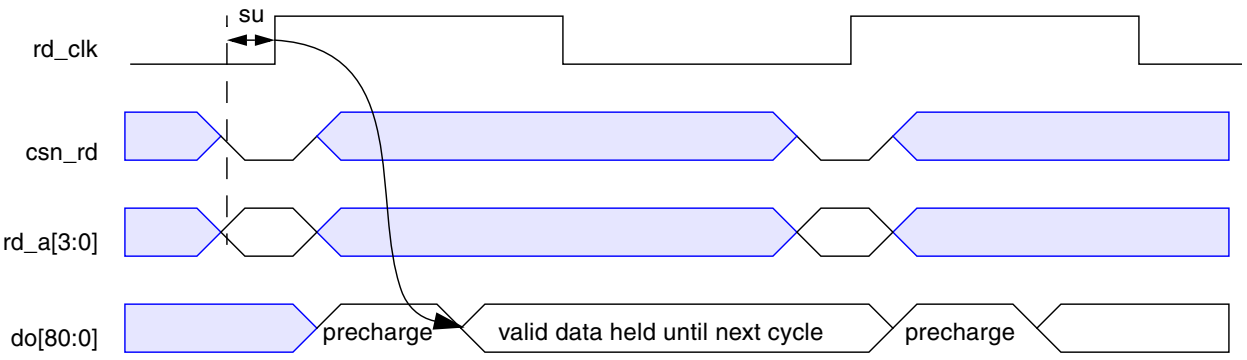

FIGURE 20-2 Read Timing Diagram

20.4.2 Write Timing Diagram

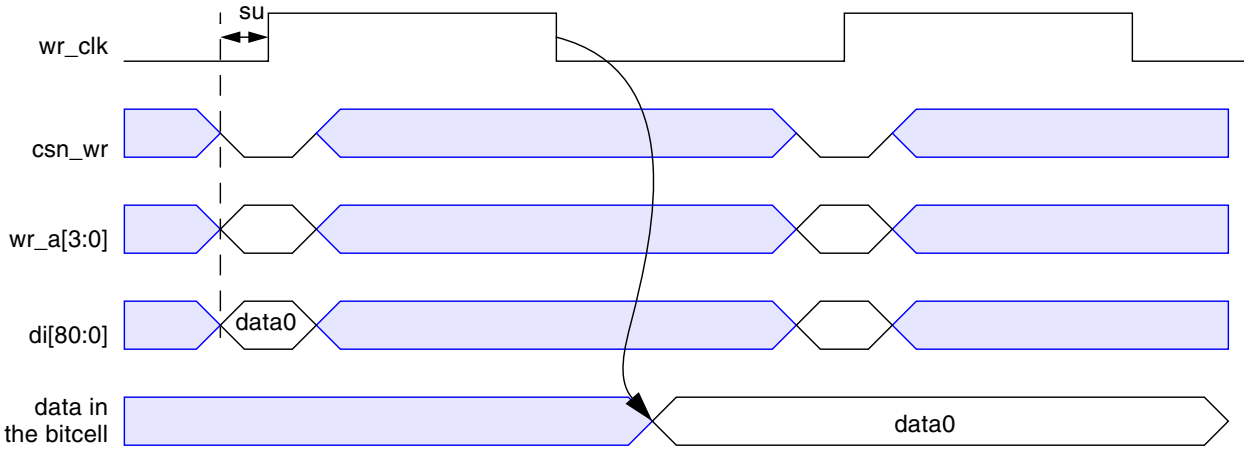

FIGURE 20-3 Write Timing Diagram

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.

big-endian An addressing convention. Within a multiple-byte integer, the byte with the smallest address is the most significant. A byte's significance decreases as its address increases.

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

CREG control register

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

DOK Port and address-range specific data OK. State maintained by each requestor port to decide if there is room for write data at the targets for a transaction. The transaction is not driven on the bus if there is no room at the destination.

DRAM dynamic random-access memory

DTL dynamic termination logic

D-TLB data lookaside buffer

ECC error correction code

EFA electronic fuse array

EFC The 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

little-endian An addressing convention. Within a multiple-byte integer, the byte with the smallest address is the least significant. A byte's significance increases as its address increases.

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 T1 processor.

MMU Memory Management Unit

mondo vector A large collection of encoded interrupts (64x8).

ODT on-die termination

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.

SPU stream processing unit

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 T1 processor.

Index

F

I

J

L

M

R

S

T