LogiCORE IP AXI EMC v1.03b

Product Guide

PG100 December 18, 2012

Table of Contents

SECTION I: SUMMARY

IP Facts

Chapter 1: Overview

Chapter 2: Product Specification

Chapter 3: Designing with the Core

SECTION II: VIVADO DESIGN SUITE

Chapter 4: Customizing and Generating the Core

Chapter 5: Constraining the Core

SECTION III: ISE DESIGN SUITE

Chapter 6: Customizing and Generating the Core

Chapter 7: Constraining the Core

SECTION IV: APPENDICES

Appendix 9: Debugging

Appendix 10: Additional Resources

SECTION I: SUMMARY

IP Facts

Introduction

The AXI External Memory Controller (EMC) IP core provides a control interface for external synchronous, asynchronous SRAM, Flash and PSRAM/Cellular RAM memory devices through the AXI interface. This soft IP core is designed to support the AXI4 interface.

Features

The AXI EMC is a soft IP core designed for Xilinx FPGAs and provides the following features:

- Supports the AXI4 specification for AXI interfaces
- AXI4 slave interface supports 32-bit address bus and 32/64-bit data bus
- 32-bit configurable AXI4-Lite control interface to access internal registers
- Burst transfers of 1-256 beats for INCR burst type and 2, 4, 8, 16 beats for WRAP burst type
- AXI4 narrow transfers, unaligned transfer type of transactions
- Multiple (up to four) external memory banks
- Independent memory configuration of each memory bank
- Memory data widths of 64-bit, 32-bit, 16-bit and 8-bit for each of the memory banks
- Synchronous/Asynchronous SRAMs, Linear Page and Burst Mode NOR Flash, and PSRAM/Cellular RAM memory devices
- Configurable byte parity check for each of the memory banks for Synchronous / Asynchronous SRAMs
- Memory configuration, timing parameters, data width for each memory bank independently
- Configurable registers for PSRAM and Linear Flash in burst mode

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	Zynq™-7000[2], Artix-7, Virtex-7, Kintex-7, Virtex-6, Spartan-6
Supported User Interfaces	AXI4-Lite, AXI4
Resources	
	See Table 2-2 through Table 2-6
Provided with Core	
Documentation	Product Specification
Design Files	VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	ISE: UCF Vivado: XDC
Simulation Model	N/A
Supported S/W Driver[4]	Standalone
Tested Design Flows	
Design Entry	XPS 14.4 Vivado Design Suite v2012.4[4]
Simulation[5]	Mentor Graphic ModelSim
Synthesis	ISE 14.4 Vivado Synthesis
Support	
Provided by Xilinx, Inc.	

1. For a complete list of supported derivative devices, see the IDS Embedded Edition Derivative Device Support.
2. Support for Zynq-7000 devices is not available in Vivado 2012.2.
3. Supports only 7 series devices.
4. Standalone driver information can be found in the EDK or SDK installation directory.
   See xilinx_drivers.htm <install_directory>/doc/usenglish.
   Linux OS and driver support information is available from http://wiki.xilinx.com.
5. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Chapter 1

Overview

The AXI EMC device is comprised of several modules:

- AXI4 Native Interface Module
- AXI4-Lite Native Interface Module
- Select Parameters Module
- Mem Steer Module
- Address Counter Mux Module
- Mem State Machine Module
- Byte parity Logic
- EMC Register Module
- I/O Registers Module

The architectural block diagram of the AXI EMC core is shown in Figure 1-1. The device module descriptions follow.

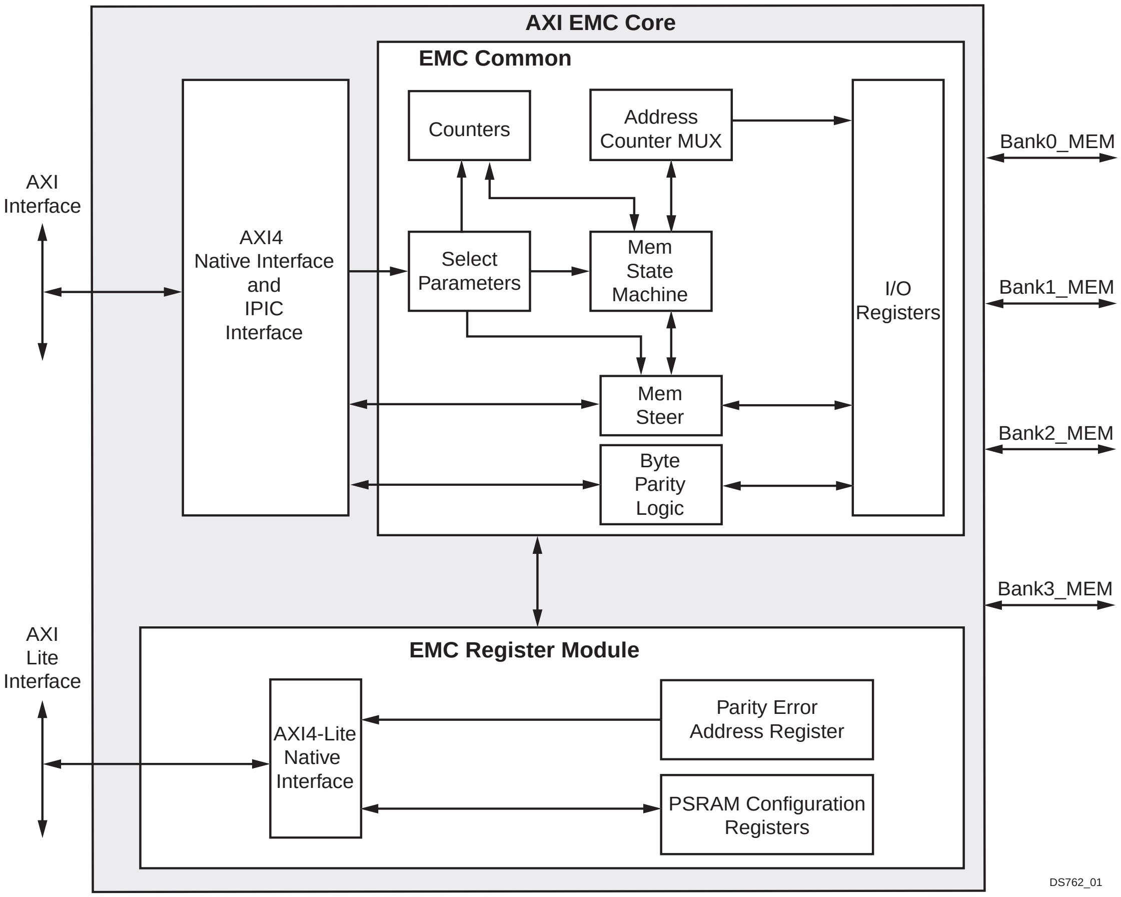


Figure 1-1: **Top Level Block Diagram of the AXI EMC Core**

AXI4 Native Interface Module

The AXI4 Native Interface Module provides the interface to the AXI memory mapped interface and implements AXI4 protocol logic. The AXI4 Native Interface Module is a bidirectional interface between an IP core and the full AXI interface standard. To simplify the process of attaching an AXI EMC core to the AXI4 interface, the core makes use of a portable bus interface native logic that includes address decoding, address generation and control signal generation. The control signal generation manages the bus interface signals, interface protocols and other interfaces. This interface is used to access external memories. The AXI4 Native Interface also provides he IPIC interface signals.

AXI4-Lite Native Interface Module

The Native AXI4-Lite interface module provides the interface to the AXI4-Lite interface and implements AXI protocol logic. This interface is used to access internal registers. This interface is enabled only when C_S_AXI_EN_REG is 1. This is not a separate module, but is embedded in the top level module of the AXI EMC core.

Select Parameters Module

The Select Parameters module provides the necessary pipeline delays or timing delays based on the type of memory. It also indicates the type of memory that is connected to the core.

Mem Steer Module

The Mem Steer module contains the logic to provide the steering of read data, write data, and memory control signals. It generates the acknowledge signals for the AXI4 Interface Module. This module contains data width matching logic.

Address Counter Mux Module

The Address Counter Mux module provides the address count to the Mem Steer module, as well as the address suffix to generate the memory address. In addition, it manages the cycle end logic that is directed to the Mem State Machine.

Mem State Machine Module

The Mem State Machine controls read and write transactions for the memory., handles all single, burst and page mode read (flash) conditions, and provides the necessary control signals to the Counters, Mem Steer, and Address Counter Mux modules.

Byte parity Logic

The Byte Parity Logic block generates and calculates the parity logic for any memory bank if the C_USE_PARITY_x parameter is enabled. The number of such blocks that are instantiated depends on C_NUM_BANKS_MEM. A parity bit is attached to a byte of data written or read from memory. For a memory write to this block, generate a parity bit (even/odd parity depending on C_PARITY_TYPE_x), and write a parity bit for each byte into memory. For a memory read, the parity bit is calculated on a read byte and then compared with the parity bit from memory. When the parity error occurs, the AXI4 logic responds with an AXI OKAY/SLVERROR response. The error address is updated in the Parity Error Address register.

EMC Register Module

There are two register sets: the parity error address register (PERR_ADDR_REG_x) and the PSRAM/Linear Flash configuration register (PSRAM_FLASH_CONFIG_REG_x). The PERR_ADDR_REG_x registers contain the address for the parity error that occurred. The number of PSRAM_FLASH_CONFIG_REG_x and PERR_ADDR_REG_x depends on C_NUM_BANKS_MEM and C_MEM_TYPE. This is not a separate module, but it is embedded in the top level file of the AXI EMC core.

I/O Registers Module

The I/O Registers module is a separate module that is at the memory interface. Registers are used on all signals to and from the memory bank to provide consistent timing on the memory interface. The I/O Registers module present in the design depends on the setting of the C_INCLUDE_NEGEDGE_IOREGS. All signals outputs to the memory bank are registered on the rising edge of the system clock. If C_INCLUDE_NEGEDGE_IOREGS = 1, the signals are registered again on the falling edge of the system clock, as shown in Figure 1-2, and can be used at lower clock frequency to provide adequate setup and hold times to synchronous memories.

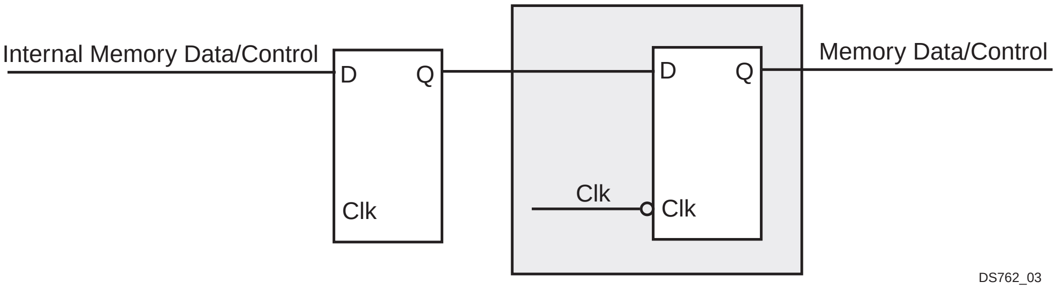


Figure 1-2: **Output Registers When C_INCLUDE_NEGEDGE_IOREGS = 1**

Licensing and Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx Vivado Design Suite and ISE Design Suite Embedded Edition tools under the terms of the Xilinx End User License.

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Chapter 2

Product Specification

This chapter contains details about the performance and ports of the core.

Performance

To measure the system performance (F_{MAX}) of the AXI EMC core, it was added as the Device Under Test (DUT) to a Virtex-6 or Spartan-6 FPGA system as shown in Figure 2-1. Because the AXI EMC core is used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When this core is combined with other designs in the system, the utilization of FPGA resources and timing of the design can vary from the results reported here.

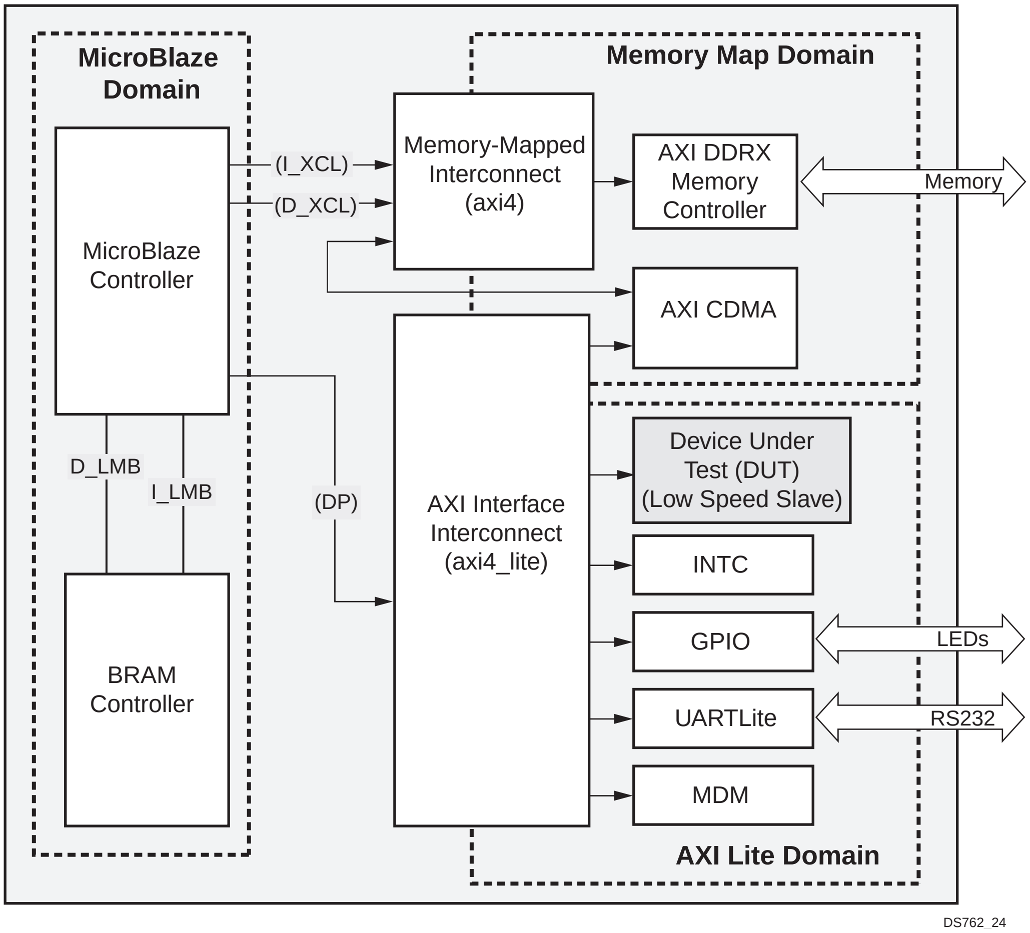


Figure 2-1: **Virtex-6 and Spartan-6 FPGA Systems with the AXI EMC as the DUT**

The target FPGA was then filled with logic to drive the LUT and block RAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target F_{MAX} numbers are shown in Table 2-1.

Maximum Frequencies

The AXI EMC core is used with other design modules and the utilization and timing numbers reported in this section are estimates only. When the core is combined with other designs in the system, the utilization of FPGA resources and timing of the design varies from the results reported here.

The target FPGA was filled with logic to drive the Lookup Table (LUT) and block RAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target F_{MAX} numbers are shown in Table 2-1.

Table 2-1: **AXI EMC System Performance**

Target FPGA	F_{MAX} (MHz)
Artix-7	110
Kintex-7	180
Virtex-7	180
Spartan-6	110
Virtex-6	180

Resource Utilization

Because the AXI EMC core is used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When the AXI EMC core is combined with other designs in the system, the utilization of FPGA resources and timing of the AXI EMC design can vary from the results reported here.

The AXI EMC resource utilization for various parameter combinations measured with Artix™-7 as the target device are detailed in Table 2-2.

Note: Performance and resource usage can be estimated for Zynq-7000 devices based on Table 2-3.

Table 2-2: **Performance and Resource Utilization on Artix-7 and Zynq-7000 (Artix-7 based Fabric)**

Parameter Values (other parameters at default value)									Device Resources			Perfor mance
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
1	1	0	0	32	32	0	1	1	159	344	310	150
2	1	0	0	32	16	1	1	1	162	329	330	150
3	1	0	0	32	8	1	1	1	160	324	333	150
4	1	0	1	32	32	0	0	0	169	326	353	150
5	1	0	1	32	16	1	0	0	187	333	389	150

Table 2-2: **Performance and Resource Utilization on Artix-7 and Zynq-7000 (Artix-7 based Fabric)**

Parameter Values (other parameters at default value)									Device Resources			Performance
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_ INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
6	1	0	1	32	8	1	0	0	170	332	383	150
7	1	1	0	32	32	0	1	1	235	502	354	150

The AXI EMC resource utilization for various parameter combinations measured with Kintex™-7 as the target device are detailed in Table 2-3.

Note: Performance and resource usage can be estimated for Zynq-7000 devices based on Table 2-3.

Table 2-3: **Performance and Resource Utilization Kintex-7 and Zynq-7000 (Kintex-7 based Fabric)**

Parameter Values (other parameters at default value)									Device Resources			Performance
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_ INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
1	1	0	0	32	32	0	1	1	197	344	310	200
2	1	0	0	32	16	1	1	1	211	329	327	200
3	1	0	0	32	8	1	1	1	221	324	337	200
4	1	0	1	32	32	0	0	0	225	326	372	200

Table 2-3: **Performance and Resource Utilization Kintex-7 and Zynq-7000 (Kintex-7 based Fabric)**

Parameter Values (other parameters at default value)									Device Resources			Performance
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_ INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
5	1	0	1	32	16	1	0	0	243	333	385	200
6	1	0	1	32	8	1	0	0	239	332	390	200
7	1	1	0	32	32	0	1	1	292	502	357	200

The AXI EMC resource utilization for various parameter combinations measured with Virtex®-7 as the target device are detailed in Table 2-4.

Table 2-4: **Performance and Resource Utilization Benchmarks on the Virtex-7 (XC7V450TFFG784-3)**

Parameter Values (other parameters at default value)									Device Resources			Performance
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_ INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
1	1	0	0	32	32	0	1	1	166	344	316	200
2	1	0	0	32	16	1	1	1	155	329	326	200
3	1	0	0	32	8	1	1	1	158	324	333	200
4	1	0	1	32	32	0	0	0	187	326	361	200

Table 2-4: **Performance and Resource Utilization Benchmarks on the Virtex-7 (XC7V450TFFG784-3)**

Parameter Values (other parameters at default value)									Device Resources			Performance
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_ INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
5	1	0	1	32	16	1	0	0	191	333	389	200
6	1	0	1	32	8	1	0	0	178	332	396	200
7	1	1	0	32	32	0	1	1	222	502	360	200

The XPS EMC resource utilization for various parameter combinations measured with the Virtex-6 FPGA as the target device with randomly selected case are detailed in Table 2-5.

Table 2-5: **Performance and Resource Utilization Benchmarks on the Virtex-6 FPGA (XC6VCX130T-FF784)**

Parameter Values (other parameters at default value)									Device Resources			Performance
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_ INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
1	1	0	0	32	32	0	1	1	199	521	335	200
2	1	0	0	16	16	1	1	1	189	447	326	200
3	1	0	0	8	8	1	1	1	171	416	359	200
4	1	0	1	32	32	0	1	1	147	508	426	200

Table 2-5: **Performance and Resource Utilization Benchmarks on the Virtex-6 FPGA (XC6VCX130T-FF784)** ***(Cont'd)***

Parameter Values (other parameters at default value)									Device Resources			Performance
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_ INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
5	1	0	1	16	16	1	1	1	193	458	452	200
6	1	0	1	8	8	1	1	1	196	430	438	200
7	1	1	0	32	32	0	1	1	448	633	382	200

Notes:

1. The parameter C_S_AXI_MEM_ADDR_WIDTH = 32 is constant for these results.

The AXI EMC resource utilization for various parameter combinations measured with Spartan®-6 as the target device are detailed in Table 2-6.

Table 2-6: **Performance and Resource Utilization Benchmarks Spartan-6 FPGA (XC6SLX75T-FGG484-3)**

Parameter Values (other parameters at default value)										Device Resources		
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_ INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
1	1	0	0	32	32	0	1	1	204	522	337	133
2	1	0	0	16	16	1	1	1	154	448	369	133
3	1	0	0	8	8	1	1	1	153	418	370	133

Table 2-6: **Performance and Resource Utilization Benchmarks Spartan-6 FPGA (XC6SLX75T-FGG484-3)**

Parameter Values (other parameters at default value)										Device Resources		
Sr. Number	C_NUM_BANKS_MEM	C_S_AXI_EN_REG	C_MEMx_TYPE	C_S_AXI_MEM_DATA_WIDTH	C_MEMx_WIDTH	C_INCLUDE_DATAWIDTH_MATCHING_x	C_PARITY_TYPE_x	C_SYNCH_PIPEDELAY_x	Slices	SliceFlip Flops	LUTs	F_{MAX} (MHz)
4	1	0	1	32	32	0	1	1	175	509	465	133
5	1	0	1	16	16	1	1	1	156	459	464	133
6	1	0	1	8	8	1	1	1	170	431	460	133
7	1	1	0	32	32	0	1	1	197	634	441	133

Notes:

1. The parameter C_S_AXI_MEM_ADDR_WIDTH = 32 is constant for these results.

IP Utilization and Performance

Table 2-7: **Variables For StrataFlash (x16 mode) Example**

Memory	Data Width	Bus Utilization	Settings	Additional Information	Latency Number
Sync SRAM	32	99.22%	Pipeline=2	Continuos write, 260 clock cycles after the first write data valid	AWVALID to WVALID - 0 clock cycles WLAST to BVALID - 1 clock cycle
Sync SRAM	32	100%	Pipeline=2	Continuos read, 260 clock cycles after the first write data valid	ARVALID to RVALID 11 clock cycles
Sync SRAM	8	99.22%	Pipeline=2	Continuos write, 260 clock cycles after the first write data valid	AWVALID to WVALID - 0 clock cycles WLAST to BVALID - 2 clock cycle
Sync SRAM	8	100%	Pipeline=2	Continuos read, 260 clock cycles after the first write data valid	ARVALID to RVALID 12 clock cycles
ASync SRAM	32	33.33%	Pipeline=2	Continuos write, 768 clock cycles after the first write data valid	AWVALID to WVALID - 0 clock cycles WLAST to BVALID - 8 clock cycle

Table 2-7: **Variables For StrataFlash (x16 mode) Example *(Cont'd)***

Memory	Data Width	Bus Utilization	Settings	Additional Information	Latency Number
ASync SRAM	32	33.33%	Pipeline=2	Continuos read, 768 clock cycles after the first write data valid	ARVALID to RVALID 12 clock cycles
ASync SRAM	8	16.8%	Pipeline=2	Continuos write,1536 clock cycles after the first write data valid	AWVALID to WVALID - 0 clock cycles WLAST to BVALID - 6 clock cycle
ASync SRAM	8	16.8%	Pipeline=2	Continuos read, 1536 clock cycles after the first write data valid	ARVALID to RVALID 18 clock cycles

Port Descriptions

The I/O signals are listed and described in Table 2-8.

Table 2-8: **I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
		AXI Global System Signals			
P1	S_AXI_ACLK[1]	AXI	I	-	AXI clock
P2	S_AXI_ARESETN	AXI	I	-	AXI reset; active Low
P3	RdClk[2]	System	I	-	Read clock to capture the data from memory
		AXI4-LITE Interface Signals[3]			
		AXI4 Write Address Channel Signals			
P4	S_AXI_REG_AWADDR [C_S_AXI_REG_ADDR_WIDTH-1:0]	AXI	I	-	AXI write address: The write address bus provides the address of the write transaction
P5	S_AXI_REG_AWVALID	AXI	I	-	Write address valid: This signal indicates that valid write address and control information are available
P6	S_AXI_REG_AWREADY	AXI	O	1	Write address ready: This signal indicates that the slave is ready to accept an address and associated control signals
		AXI4Lite Write Channel Signals			
P7	S_AXI_REG_WDATA [C_S_AXI_REG_DATA_WIDTH - 1: 0]	AXI	I	-	Write data
P8	S_AXI_REG_WSTB [C_S_REG_AXI_DATA_WIDTH/8-1:0]	AXI	I	-	Write strobes: This signal indicates which byte lanes to update in memory
P9	S_AXI_REG_WVALID	AXI	I	-	Write valid. This signal indicates that valid write data and strobes are available

Table 2-8: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P10	S_AXI_REG_WREADY	AXI	O	1	Write ready. This signal indicates that the slave can accept the write data
AXI4 Write Interface Response Channel Signals					
P11	S_AXI_REG_BRESP[1:0]	AXI	O	0x0	Write response: This signal indicates the status of the write transaction 00 - OKAY 10 - SLVERR
P12	S_AXI_REG_BVALID	AXI	O	0x0	Write response valid: This signal indicates that a valid write response is available
P13	S_AXI_REG_BREADY	AXI	I	-	Response ready: This signal indicates that the master can accept the response information
AXI4-Lite Read Address Channel Signals					
P14	S_AXI_REG_ARADDR [C_S_AXI_REG_ADDR_WIDTH-1:0]	AXI	I	-	Read address: The read address bus gives the address of a read transaction
P15	S_AXI_REG_ARVALID	AXI	I	-	Read address valid: This signal indicates, when high, that the read address and control information is valid and remains stable until the address acknowledgement signal, ARREDY, is high.
P16	S_AXI_REG_ARREADY	AXI	O	0x1	Read address ready: This signal indicates that the slave is ready to accept an address and associated control signals.
AXI4-Lite Read Data Channel Signals					
P17	S_AXI_REG_RDATA [C_S_AXI_REG_DATA_WIDTH -1:0]	AXI	O	0x0	Read data
P18	S_AXI_REG_RRESP[1:0]	AXI	O	0x0	Read response: This signal indicates the status of the read transfer. 00 - OKAY 10 - SLVERR
P19	S_AXI_REG_RVALID	AXI	O	0x0	Read valid: This signal indicates that the required read data is available and the read transfer can complete
P20	S_AXI_REG_RREADY	AXI	I	-	Read ready: This signal indicates that the master can accept the read data and response information
AXI4 Full Write Data Channel Signals					
P21	S_AXI_MEM_AWID [C_S_MEM_AXI_ID_WIDTH-1:0]	AXI	I	-	Write address ID: This signal is the identification tag for the write address group of signals.
P22	S_AXI_MEM_AWADDR [C_S_AXI_MEM_ADDR_WIDTH-1:0]	AXI	I	-	AXI Write address: The write address bus gives the address of the first transfer in a write burst transaction.

Table 2-8: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P23	S_AXI_MEM_AWLEN[7:0]	AXI	I	-	Burst length: This signal gives the exact number of transfers in a burst. 00000000 - 11111111 indicates burst length 1 - 256.
P24	S_AXI_MEM_AWSIZE[2:0]	AXI	I	-	Burst size: This signal indicates the size of each transfer in the burst. 000 - 1 byte 001 - 2 byte (Half word) 010 - 4 byte (word) others - NA (up to 128 bytes)
P25	S_AXI_MEM_AWBURST[1:0]	AXI	I	-	Burst type: This signal coupled with the size information, details how the address for each transfer within the burst is calculated. 00 - FIXED 01 - INCR 10 - WRAP 11 - Reserved
P26	S_AXI_MEM_AWLOCK	AXI	I	-	Lock type: This signal provides additional information about the atomic characteristics of the transfer. This signal is not used in the design.
P27	S_AXI_MEM_AWCACHE[4:0]	AXI	I	-	Cache type: This signal indicates the bufferable, cacheable, write-through, write-back and allocate attributes of the transaction Bit-0: Bufferable (B) Bit-1: Cacheable (C) Bit-2: Read Allocate (RA) Bit-3: Write Allocate (WA) The combination where C=0 and WA/RA=1 are reserved. This signal is not used in the design.
P28	S_AXI_MEM_AWPROT[2:0]	AXI	I	-	Protection type: This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access. Bit-0: 0=Normal access, 1=Privileged access Bit-1: 0=Secure access, 1=non-secure access Bit- 2: 0=data access; 1=instruction access This signal is not used in the design.
P29	S_AXI_MEM_AWVALID	AXI	I	-	Write address valid: This signal indicates that valid write address and control information are available.
P30	S_AXI_MEM_AWREADY	AXI	O	0	Write address ready: This signal indicates that the slave is ready to accept an address and associated control signals.
AXI4 Full Interface Write Channel Signals					

Table 2-8: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P31	S_AXI_MEM_WDATA [C_S_AXI_MEM_DATA_WIDTH-1:0]	AXI	I	-	Write data bus.
P32	S_AXI_MEM_WSTB [(C_S_AXI_MEM_DATA_WIDTH/ 8)-1:0]	AXI	I	-	Write strobes: This signal indicates the byte lanes in S_AXI_WDATA are valid.
P33	S_AXI_MEM_WLAST	AXI	I	-	Write last: This signal indicates the last transfer in a write burst.
P34	S_AXI_MEM_WVALID	AXI	I	-	Write valid: This signal indicates that valid write data and strobes are available.
P35	S_AXI_MEM_WREADY	AXI	O	0	Write ready: This signal indicates that the slave can accept the write data.
AXI4 Full Interface Write Response Channel Signals					
P36	S_AXI_MEM_BID [C_S_AXI_MEM_ID_WIDTH-1:0]	AXI	O	0	Write response ID: This signal is the identification tag of the write response. The BID value must match the AWID value of the write transaction to which the slave is responding.
P37	S_AXI_MEM_BRESP[1:0]	AXI	O	0	Write response: This signal indicates the status of the write transaction. 00 - OKAY 01 - EXOKAY - NA 10 - SLVERR - NA 11 - DECERR - NA
P38	S_AXI_MEM_BVALID	AXI	O	0	Write response valid: This signal indicates that a valid write response is available.
P39	S_AXI_MEM_BREADY	AXI	I	-	Response ready: This signal indicates that the master can accept the response information.

Table 2-8: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
AXI4 Full Interface Read Address Channel Signals					
P40	S_AXI_MEM_ARID [C_S_AXI_MEM_ID_WIDTH-1:0]	AXI	I	-	Read address ID: This signal is the identification tag for the read address group of signals.
P41	S_AXI_MEM_ARADDR [C_S_AXI_MEM_ADDR_WIDTH -1:0]	AXI	I	-	Read address: The read address bus gives the initial address of a read burst transaction.
P42	S_AXI_MEM_ARLEN[7:0]	AXI	I	-	Burst length: This signal gives the exact number of transfers in a burst. 00000000 - 11111111 indicates Burst Length 1 - 256.
P43	S_AXI_MEM_ARSIZE[2:0]	AXI	I	-	Burst size: This signal indicates the size of each transfer in the burst.
P44	S_AXI_MEM_ARBURST[1:0]	AXI	I	-	Burst type: The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated. 00 - FIXED 01 - INCR 10 - WRAP 11 - Reserved
P45	S_AXI_MEM_ARLOCK	AXI	I	-	Lock type: This signal provides additional information about the atomic characteristics of the transfer. This signal is not used in the design.
P46	S_AXI_MEM_ARCACHE[4:0]	AXI	I	-	Cache type: This signal provides additional information about the cacheable characteristics of the transfer. Bit-0: Bufferable (B) Bit-1: Cacheable (C) Bit-2: Read Allocate (RA) Bit-3: Write Allocate (WA) The combination where C=0 and WA/RA=1 are reserved. This signal is not used in the design.
P47	S_AXI_MEM_ARPROT[2:0]	AXI	I	-	Protection type: This signal provides protection unit information for the transaction. This signal is not used in the design.
P48	S_AXI_MEM_ARVALID	AXI	I	-	Read address valid: This signal indicates, when high, that the read address and control information is valid and remains stable until the address acknowledgement signal, ARREDY, is high.
P49	S_AXI_MEM_ARREADY	AXI	O	0	Read address ready: This signal indicates that the slave is ready to accept an address and associated control signals.

Table 2-8: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
AXI4 Full Interface Read Data Channel Signals					
P50	S_AXI_MEM_RID [C_S_MEM_AXI_ID_WIDTH-1:0]	AXI	O	0	Read ID tag: This signal is the ID tag of the read data group of signals. The RID value is generated by the slave and must match the ARID value of the read transaction to which it is responding.
P51	S_AXI_MEM_RDATA [C_S_AXI_MEM_DATA_WIDTH -1:0]	AXI	O	0	Read data bus.
P52	S_AXI_MEM_RRESP[1:0]	AXI	O	0	Read response: This signal indicates the status of the read transfer.
P53	S_AXI_MEM_RLAST	AXI	O	0	Read last: This signal indicates the last transfer in a read burst.
P54	S_AXI_MEM_RVALID	AXI	O	0	Read valid: This signal indicates that the required read data is available and the read transfer can complete.
P55	S_AXI_MEM_RREADY	AXI	I	-	Read ready: This signal indicates that the master can accept the read data and response information.
EXternal Memory Interface Signal					
P56	MEM_DQ_I [C_MAX_MEM_WIDTH - 1:0]	External memory	I	-	Memory input data bus
P57	MEM_DQ_O [C_MAX_MEM_WIDTH - 1:0]	External memory	O	0	Memory output data bus
P58	MEM_DQ_T [C_MAX_MEM_WIDTH - 1:0]	External memory	O	0	Memory output 3-state signal
P59	MEM_DQ_PARITY_I [C_MAX_MEM_WIDTH/8 - 1:0]	External memory	I	-	Memory parity input data bits
P60	MEM_DQ_PARITY_O [C_MAX_MEM_WIDTH/8 - 1:0]	External memory	O	0	Memory parity output data bits
P61	MEM_DQ_PARITY_T [C_MAX_MEM_WIDTH/8 - 1:0]	External memory	O	0	Memory parity 3-state signals
P62	MEM_A [C_S_AXI_MEM_ADDR_WIDTH - 1:0]	External memory	O	0	Memory address bus
P63	MEM_RPN	External memory	O	1	Memory reset/power down
P64	MEM_CEN [C_NUM_BANKS_MEM - 1:0]	External memory	O	1	Memory chip enables(4); active Low
P65	MEM_OEN [C_NUM_BANKS_MEM - 1:0]	External memory	O	1	Memory output enable
P66	MEM_WEN	External memory	O	1	Memory write enable

Table 2-8: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P67	MEM_QWEN [(C_MAX_MEM_WIDTH/8) - 1:0]	External memory	O	1	Memory qualified write enables
P68	MEM_BEN [(C_MAX_MEM_WIDTH/8) - 1:0]	External memory	O	0	Memory byte enables
P69	MEM_CE[C_NUM_BANKS_MEM - 1:0]	External memory	O	0	Memory chip enables[4]; active High
P70	MEM_ADV_LDN	External memory	O	1	Memory advance burst address/load new address
P71	MEM_LBON	External memory	O	1	Memory linear/interleaved burst order
P72	MEM_CKEN	External memory	O	0	Memory clock enable
P73	MEM_RNW	External memory	O	1	Memory read not write
P74	MEM_CRE	External memory	O	0	Command sequence configuration of RSRAM
P75	Mem_WAIT	External memory	I	-	Input signal from Numonyx Flash device; This signal is used only when C_MEMx_TYPE is set to 5

Notes:

1. The same clock and reset signals should be used for both the register and memory interfaces.
2. This clock is used to capture the data from memory. Connection to this port is required. |n general, this should be connected to the system/bus clock. It can be connected to other clock nets, for example, the phase shift clock or feedback clock.
3. The AXI4-Lite interface is only available when C_S_AXI_EN_REG = 1.
4. Most asynchronous memory devices only use MEM_CEN. Most synchronous memory devices use both MEM_CEN and MEM_CE. See the device data sheet for the correct connection of these signals.

Inferred Parameters

In addition to the parameters listed in Table 2-9, there are also parameters that are inferred for each AXI interface in the EDK tools. Through the design, these EDK-inferred parameters control the behavior of the AXI interconnect. For a complete list of the interconnect settings related to the AXI interface, see [Ref 4].

Table 2-9: **Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameter					
G1	Target FPGA family	C_FAMILY	artix7. kintex7, virtex7, zynq7000, virtex6. spartan6	virtex6	string
AXI4-Lite Interface Parameters[1]					

Table 2-9: **Design Parameters *(Cont'd)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G2	Include AXI4-LITE interface	C_S_AXI_EN_REG	0-1	0	integer
G4	AXI address bus width	C_S_AXI_REG_ADDR_WIDTH	32	32	integer
G5	AXI data bus width	C_S_AXI_REG_DATA_WIDTH	32	32	integer
AXI4 Full Interface Parameters					
G6	AXI address bus width	C_S_AXI_MEM_ADDR_WIDTH	32	32	integer
G7	AXI data bus width	C_S_AXI_MEM_DATA_WIDTH	32,64	32	integer
G8	AXI Identification tag width	C_S_AXI_MEM_ID_WIDTH	1 - 16	4	integer
EMC Parameters					
G9	Number of memory banks	C_NUM_BANKS_MEM	1 - 4	1	integer
G10	Width of memory bank x data bus	C_MEMx_WIDTH [2]	8, 16, 32, 64	32	integer
G11	Type of memory	C_MEMx_Type[2]	0 = Sync SRAM 1 = Async SRAM 2 = Linear Flash 3 = Page Mode Flash 4 = PSRAM 5 = Numonyx Flash	0	integer
G12	Burst mode for Linear Flash	C_LINEAR_FLASH_SYNC_BURST	1=User can configure Numonyx flash for burst mode through register interface 0 = Burst mode cannot be configured even though the memory bank type is 5	0	Integer
G13	Execute multiple memory access cycles to match memory bank x data width to AXIdata width	C_INCLUDE_DATAWIDTH_MATCHING_x[2][3]	0 = Do not include data width matching 1 = Include data width matching	0	integer
G14	Parity Check	C_PARITY_TYPE_x[2][4]	0 = No parity 1 = Odd Parity 2= Even Parity	0	integer
G15	Pipeline delay (in clock cycles) of memory bank x	C_SYNCH_PIPEDELAY_x[2][5]	1 = Flow-Through model 2 = Pipeline Model	2	integer

Table 2-9: **Design Parameters *(Cont'd)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G16	Input/output data and control signals using the falling edge of the clock	C_INCLUDE_NEGEDGE_ IOREGS(7)(8)	0 = Do not include negative edge I/O registers (data and control signals are input/output on the rising edge of the clock) 1 = Include negative edge I/O registers (data and control signals are input/output on the falling edge of the clock)	0	integer
Memory Bank Timing Parameters					
G17	Read cycle chip enable low to data valid duration of memory bank x	C_TCEDV_PS_MEM_x(2)(10)(11)	Integer number of picoseconds	15000(12)	integer
G18	Read cycle address valid to data valid duration of memory bank x	C_TAVDV_PS_MEM_x(2)(10)(13)	Integer number of picoseconds	15000(12)	integer
G19	Read cycle chip enable high to data bus high impedance duration of memory bank x	C_THZCE_PS_MEM_x(2)(14)(15)	Integer number of picoseconds	7000(12)	integer
G20	Read cycle output enable high to data bus high impedance duration of memory bank x	C_THZOE_PS_MEM_x(2)(14)(16)	Integer number of picoseconds	7000(12)	integer
G21	Page access time of memory bank x in Page Mode Flash mode	C_TPACC_PS_FLASH_x(2)(17)	Integer number of picoseconds	25000(12)(17)	integer
G22	Write cycle time of memory bank x	C_TWC_PS_MEM_x(2)(18)(19)	Integer number of picoseconds	15000(12)	integer
G23	Write enable minimum pulse width duration of memory bank x	C_TWP_PS_MEM_x(2)(18)(20)	Integer number of picoseconds	12000(12)	integer
G24	Write cycle write enable high to data bus low impedance duration of memory bank x	C_TLZWE_PS_MEM_x(2)(21)(22)	Integer number of picoseconds	35000(12)	integer
G25	Write cycle phase time period duration of memory bank x	C_WPH_PS_MEM_x	Integer number of picoseconds	12000(12)	integer

Table 2-9: **Design Parameters *(Cont'd)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G26	Write Recovery Period for Flash Memories	C_WR_REC_TIME_MEM_x	Integer number of picoseconds[25]	60000	integer
Auto Calculated Parameter[23]					
G27	Maximum data width of the memory devices in all banks	C_MAX_MEM_WIDTH[2]	8, 16, 32, 64	32	integer
Address Space Parameters					
G28	Base address of memory bank x	C_S_AXI_MEMx_BASEADDR[2]	Valid address range[24]	None[24]	std_logic_vector
G29	High address of memory bank x	C_S_AXI_MEMx_HIGHADDR[2]	Valid address range[24]	None[24]	std_logic_vector
Clock Period Parameter					
G30	AXI clock period	C_AXI_CLK_PERIOD_PS	Integer number of picoseconds	10000	integer
G31	Linear Flash clock period	C_LFLASH_PERIOD_PS	Integer number of picoseconds	20000	Integer

Notes:

1. Valid only when C_S_AXI_EN_REG=1.
2. x = values for memory banks 0 to 3.
3. Always set this parameter to 1 when C_MEMx_WIDTH not equal to C_S_AXI_DATA_WIDTH.
4. C_PARITY_TYPE_x is valid only when C_MEMx_Type = 0 or 1.
5. Valid only when C_MEMx_Type = 000.
6. Valid only when C_MEMx_Type not equal to 0.
7. This parameter should only be set to 1 under the following conditions:
    - Tfpga_output_buffer_delay + Tmemory_setup + Tboard_route_delay < Clock_period/2
    - Tmemory_output_bufferdelay + Tfpga_setup + Tboard_route_delay < Clock_period/2
    - Tfpga_output_buffer_delay is the delay between the flop output and the output pin
    - Tmemory_setup is the memory input setup time requirement
    - Tboard_route_delay is time delay between FPGA output pin to Memory input pin and vice versa
    - Tfpga_setup time is the input setup time requirement of input pins of FPGA.
8. C_INCLUDE_NEGEDGE_IOREGS used when C_MEMx_Type = 0.
9. C_SYNCH_PIPEDELAY_x is used when C_MEMx_Type = 0.
10. Read cycle time is the maximum of C_TCEDV_PS_MEM_x and C_TAVDV_PS_MEM_x, and C_TCEDV_PS_MEM_x or C_TAVDV_PS_MEM_x should be >= tRC timing parameter of the given asynchronous memory.

11. Chip enable low to data valid, C_TCEDV_PS_MEM_x, is equivalent to t_{ACE} for asynchronous SRAM and t_{ELQV} for flash memory in the respective memory device data sheets.
12. A value must be set for this parameter if the memory type in this bank is asynchronous. See the memory device data sheet for the correct value.
13. Address valid to data valid, C_TAVDV_PS_MEM_x, is equivalent to t_{AA} for asynchronous SRAM and t_{AVQV} for flash memory in the respective memory device data sheets.
14. Read cycle recovery to write is the maximum of C_THZCE_PS_MEM_x and C_THZOE_PS_MEM_x.
15. Chip enable high to data bus high impedance, C_THZCE_PS_MEM_x, is equivalent to t_{HZCE} for asynchronous SRAM and t_{EHQZ} for flash memory in the respective memory device data sheets.
16. Output enable high to data bus high impedance, C_THZOE_PS_MEM_x, is equivalent to t_{HZOE} for asynchronous SRAM and t_{GHQZ} for flash memory in the respective memory device data sheets.
17. Page access time, C_TPACC_PS_FLASH_x is equivalent to t_{PACC} in the Page Mode Flash device data sheet and must be assigned only if C_MEMx_Type = 011.
18. Write enable low time is the maximum of C_TWC_PS_MEM_x and C_TWP_PS_MEM_x.
19. Write cycle time, C_TWC_PS_MEM_x, is equivalent to t_{WC} for asynchronous SRAM and t_{CW} for flash memory in the respective memory device data sheets.
20. Write cycle minimum pulse width, C_TWP_PS_MEM_x is equivalent to t_{WP} for asynchronous SRAM and t_{PWE} for flash memory in the respective memory device data sheets.
21. Write enable high to data bus low impedance, C_TLZWE_PS_MEM_x, is equivalent to t_{LZWE} for asynchronous SRAM and t_{WHGL} for flash memory in the respective memory device data sheets.
22. C_TLZWE_PS_MEM_x is the parameter set to meet write recovery to read time requirements.
23. This parameter is automatically calculated when using EDK, otherwise set this parameter to the maximum value of the C_MEMx_WIDTH generics.
24. No default value is specified for C_MEMx_BASEADDR and C_MEMx_HIGHADDR to ensure that the actual value is set, that is, if the value is not set, a compiler error is generated. The range specified by C_MEMx_BASEADDR and C_MEMx_HIGHADDR must be a power of 2.
25. The write recovery time period is applicable only for flash memories. A maximum of 320 ns time period is allowed for this parameter.
26. Period of C_LFLASH_PERIOD_PS must be integer multiple of C_AXI_CLK_PERIOD_PS.

Allowable Parameter Combinations

The AXI4-Lite Native Interface is included in the design when C_S_AXI_EN_REG=1. The registers are valid only for PSRAM, Linear Flash (in burst mode) and SRAM memories (C_MEMx_Type = 0,4,5) and therefore C_S_AXI_EN_REG should be used for the these memories only.

If C_MEMx_Type = 0, then C_SYNCH_PIPEDELAY_x specifies the pipeline delay of that synchronous memory type. All other timing parameters for that memory bank can remain at the default value of 0. If C_MEMx_Type is not equal to 0, then C_SYNCH_PIPEDELAY_x is unused. All other timing parameters for that memory bank must be set to the value specified in the memory device data sheet.

C_INCLUDE_NEGEDGE_IOREGS provides no benefit when interfacing to asynchronous memories and is valid if C_MEMx_Type is not equal to 0. Therefore, if there are no synchronous memories in the system, this parameter should be set to 0.

All memory timing parameters are valid only for asynchronous memories, for example, when C_MEMx_Type is not equal to 0 and in some modes of PSRAM memories.

Internal Calculation of the Timing Parameters

The timing parameters for the core are in effect in the design only when the targeted memory type is Asynchronous SRAM, Linear Flash, Page Mode Flash or PSRAM. When used for internal counter signal assertion and de-assertion, some parameters are combined and referred for final calculation. This calculation reflects in the behavior of external signals activation and de-activation. Table 2-10 lists the calculations for the timing parameters.

Table 2-10: **Timing Parameter Calculation**

Timing Parameters	Calculation
Read Cycle End to Data Bus High Impedance	(max2(1,max2(C_THZCE_PS_MEM_x, C_THZOE_PS_MEM_x))-1)/C_BUS_CLOCK_PERIOD_PS)
Read Chip Enable To Data Active From The Memory	((max2(1,max2(C_TCEDV_PS_MEM_x, C_TAVDV_PS_MEM_x))-1)/C_BUS_CLOCK_PERIOD_PS)
Page Access Time of Memory Bank (for Page Mode Flash)	(C_TPACC_PS_FLASH_x/C_BUS_CLOCK_PERIOD_PS)
Write Cycle to Data Store	((max2(1,max2(C_TWC_PS_MEM_x, C_TWP_PS_MEM_x))-1)/C_BUS_CLOCK_PERIOD_PS)
Write Cycle De-Active Period	(C_TWPH_PS_MEM_x/C_BUS_CLOCK_PERIOD_PS)
Write Cycle End Data Hold Time	((max2(1,C_TLZWE_PS_MEM_0)-1)/C_BUS_CLOCK_PERIOD_PS)
Write Recovery Time for Flash Memories	(C_WR_REC_TIME_MEM_0/C_BUS_CLOCK_PERIOD_PS)

Internally, for timing calculations, the parameters are used as follows:

- C_TCEDV_PS_MEM_x - Read cycle chip enable low to data valid duration of memory bank x
- C_TAVDV_PS_MEM_x - Read cycle address valid to data valid duration of memory bank x
- C_TRD_TPACC_x - Page access time of memory bank x in Page Mode Flash mode
- C_THZCE_PS_MEM_x - Read cycle chip enable low to data valid duration of memory bank x
- C_THZOE_PS_MEM_x - Enable high to data bus high impedance duration of memory bank x
- C_TWC_PS_MEM_x - Write cycle time of memory bank x
- C_TWP_PS_MEM_x - Write enable minimum pulse width duration of memory bank x
- C_TWPH_PS_MEM_x - Write phase cycle time for memory bank x
- C_TLZWE_PS_MEM_x - Write cycle write enable high to data bus low impedance duration of memory bank x

- C_TPACC_PS_FLASH_x - Page access time of memory bank x in Page Mode Flash memory
- C_WR_REC_TIME_MEM_x - Write recovery time period for flash memories
- C_LFLASH_PERIOD_PS - Period of the Linear Flash clock operated in synchronous burst mode

These timing parameters are used internally by the design to calculate the number of clock cycles for each of the operations.

Register Space

There are four internal registers in the AXI EMC design (Table 2-11). The memory map of the AXI EMC registers is determined by setting the C_S_AXI_REG_BASEADDR parameter. The internal registers of the AXI EMC are at a fixed offset from the base address and are byte accessible. The AXI EMC internal registers and their offset are listed in Table 2-11.

Table 2-11: **Internal Registers and Offsets**

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
C_S_AXI_REG_BASEADDR + 0x00[1]	PARITY_ERR_ADDR_REG_0	Read	0x0	Bank-0 parity error address Register[3]
C_S_AXI_REG_BASEADDR + 0x04[1][2]	PARITY_ERR_ADDR_REG_1	Read	0x0	Bank-1 parity error address Register[3]
C_S_AXI_REG_BASEADDR + 0x08[1][2]	PARITY_ERR_ADDR_REG_2	Read	0x0	Bank-2 parity error address Register[3]
C_S_AXI_REG_BASEADDR + 0x0C[1][2]	PARITY_ERR_ADDR_REG_3	Read	0x0	Bank-3 parity error address Register[3]
C_S_AXI_REG_BASEADDR + 0x10[1]	PSRAM_FLASH_CONFIG_REG_0	Write/ Read	0x24	Bank-0 PSRAM/FLASH Configuration Register[4]
C_S_AXI_REG_BASEADDR + 0x14[1][2]	PSRAM_FLASH_CONFIG_REG_1	Write/ Read	0x24	Bank-1 PSRAM/FLASH Configuration Register[4]
C_S_AXI_REG_BASEADDR + 0x18[1][2]	PSRAM_FLASH_CONFIG_REG_2	Write/ Read	0x24	Bank-2PSRAM/FLASH Configuration Register[4]
C_S_AXI_REG_BASEADDR + 0x1C[1][2]	PSRAM_FLASH_CONFIG_REG_3	Write/ Read	0x24	Bank-3PSRAM/FLASH Configuration Register[4]

Notes:

1. The register block is included when C_S_AXI_EN_REG = 1
2. The number of memory banks depends on C_NUM_BANKS_MEM value
3. Parity check registers valid when C_MEMx_Type=0/1
4. PSRAM configuration registers valid only when C_MEMx_Type=5
5. FLASH configuration registers valid only when C_MEMx_Type=4 and C_LINEAR_FLASH_SYNC_BURST = 1

PARITY ERROR ADDRESS Register (PARITY_ERR_ADDR_REG_x)

The AXI EMC Parity Error Address registers are read-only registers that provide the AXI address on which the parity error occurred. These registers are valid only when C_S_AXI_EN_REG = 1 and C_MEMx_Type=0/1. The number of such registers depends on the C_NUM_BANKS_MEM parameter.

There can be four PARITY_ERR_ADDR_REG registers. Whenever a parity error occurs for a AXI read operation, the EMC_COMMON module updates this register with the address at which the error occurred. The Parity Error Address Register is shown in Figure 2-2 and described in Table 2-12.

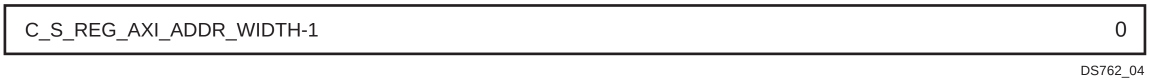


Figure 2-2: **AXI EMC Parity Error address Register**

Table 2-12: **Parity Error Address Register Description**

Bits	Name	Core Access	Reset Value	Description
C_S_AXI_REG_ADDR_WIDTH-1:0	PARITY_ERR_ADDR_REG_x	Read	0x0	Parity error register width

PSRAM/FLASH Configuration Register (PSRAM_FLASH_CONFIG_REG_X)

The PSRAM/FLASH configuration registers are write and read registers that are used for configuration of the controller for the PSRAM or Flash memories. These registers are used to configure PSRAM memory when C_S_AXI_EN_REG = 1 and C_MEMx_Type = 4; Flash memories in burst mode when C_S_AXI_EN_REG = 1, C_MEMx_Type = 5 and C_LINEAR_FLASH_SYNC_BURST = 1. The number of such registers depends on the C_NUM_BANKS_MEM parameter and their corresponding C_MEMx_Type parameters.

Note: All banks must be connected as either PSRAM or FLASH type memories; combination of these memories is not possible.

The PSRAM for the EMC controller works only with active Low wait polarity. Burst configurations of the PSRAM Control registers should be configured based on the AXI cache line configurations.

Linear Flash Burst mode for the EMC controller works with active High wait polarity, WAIT deassertion with valid data, linear burst, rising edge, no wrap and 4, 8, 16 word burst length configurations.

Burst Mode Configuration of Flash Memories

The EMC is designed to support the configuration of Numonyx Flash (28F00AP30TF) in synchronous burst mode. The process to configure the EMC to control the Flash in synchronous burst mode is described as follows:

- The C_LINEAR_FLASH_SYNC_BURST parameter must be set to 1 to enable synchronous Linear Flash Burst mode.
- The frequency of operation of the EMC must always be integer multiple (greater than or equal to 2) of the Flash operating frequency.
- The C_S_AXI_EN_REG parameter must be set to 1 to enable the register interface.
- Use bits [31:30] of the PSRAM_FLASH_CONFIG_REG_X register to enable the EMC to access Flash data in synchronous burst mode.
    - Set bit 31 to one: EMC registers address (Aligned/Un-aligned) of any transaction through the AXI4 interface and output to FLASH address port.
    - Set bit 31 to zero: the EMC output aligned address to the flash address port.
    - Set bit 30 to one: the EMC operates in synchronous burst read mode
    - Set bit 30 to zero: the EMC operates in asynchronous mode

The following steps are used to configure Numonyx flash only. For other third party Flash memories, the user must configure the EMC to operate in synchronous burst mode.

Numonyx Flash (28F00AP30TF) Configuration in Synchronous Burst Mode

IMPORTANT: *Read the Numonyx Flash (28F00AP30TF) data sheet carefully to understand the mechanism for write and read process from the memory.*

The EMC always outputs aligned addresses to the memories but accessing its configuration control register requires an unaligned address output. An Unaligned address from the EMC can be provided by setting the 31^{st} bit of the PSRAM_FLASH_CONFIG_REG_X control register. This control register can be accessed using the AXI4-Lite interface of the EMC.

- Write to PSRAM_FLASH_CONFIG_REG_X through the AXI4-Lite interface:
    - Set bit 31 to one
    - Set bit 30 to one
- An AXI4 full transaction of two word writes must be initiated with first data as 0x60 and second data as 0x03 (the address of the transaction A[15:0] must be equal to read configuration register[15:0])
- Using the AXI4-Lite interface, bit 31 of the control register must be set to 0 to pass the actual address from the EMC to the Flash memory.

Note: Linear Flash Burst mode is validated on the KC705 board.

There can be four PSRAM_FLASH_CONFIG_REG registers. The PSRAM/Linear Flash Burst mode configuration register is shown in Figure 2-3 and described in Table 2-13.

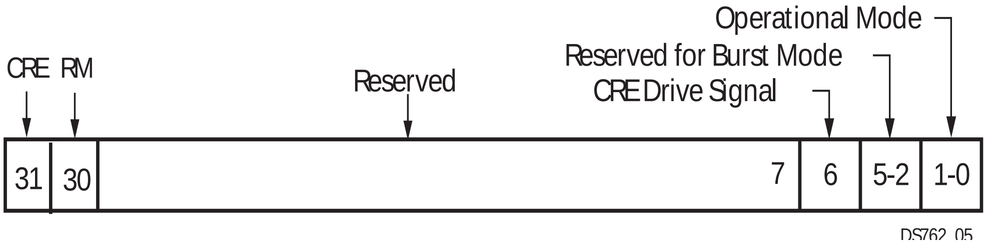


Figure 2-3: **PSRAM / Linear Flash Configuration Register**

Table 2-13: **PSRAM / Linear Flash Configuration Register (PSRAM_FLASH_CONFIG_REG_X)**

Bits	Name	Core Access	Reset Value	Description
31	Control register enable (CRE)	Write/ Read	0x0	When set, next memory writes copies PSRAM_FLASH_CONFIG_REG_x[30:15] to the address lines[16:1] of flash
30	Read Mode (RM)	Write/ Read	0x0	• 0 = Asynchronous Page Mode • 1 = Synchronous Burst Mode
29:7	Reserved	N/A	0	Reserved. If read, may return undefined value. If written to, these bits will not be updated.
6	CRE drive	Write/ Read	0	• 0x0= Remove Drive Command for CRE • 0x1= Drive Command for CRE
5:2	Reserved	Write/ Read	0x9	Reserved for future use of powerdown and burst mode schemes of cellular RAM
1:0	Operational Mode	Write/ Read	0x0	This bits describe the mode in which PSRAM is configured • 0x00= Asynchronous Mode • 0x01= Page Mode • 0x10= Reserved for future use (Burst Mode)

Notes:

1. Various latency codes can be configured. See the PSRAM data sheet for reference.

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the core.

General Design Guidelines

Address Map Description

As shown in Table 3-1, the AXI EMC device supports up to four banks of external memory. The number of available banks actually used is determined by the C_NUM_BANKS_MEM parameter. This parameter can take values between 1 and 4, inclusive. The banks that are used, if any, are banks 0 to C_NUM_BANKS_MEM - 1. Each bank of memory has its own independent base address and high address range. The address range of a bank of memory is restricted to have a size (in bytes) that is a power of 2 and to be address-aligned to the same power of 2. This means that for an address-range of size is 2^n, the *n* least significant bits of the base address is 0 and *n* least significant bits of the high address are 1. For example, a memory bank with an addressable range of 16 MB (2^{24}) could have a base address of 0xFF000000 and a high address of 0xFFFFFFFF. A memory bank with an addressable range of 64 kB (2^{16}) could have a base address of 0xABCD0000 and a high address of 0xABCDFFFF. The AXI EMC core transactions must fall between the bank x base address C_MEMx_BASEADDR and high address C_MEMx_HIGHADDR. The addresses for the memory banks are shown in Table 3-1.

Table 3-1: **Registers and Memory Bank**

Memory	Base Address	High Address	Access
Bank 0	C_S_AXI_MEM0_BASEADDR	C_S_AXI_MEM0_HIGHADDR	Read/Write
Bank 1[1]	C_S_AXI_MEM1_BASEADDR	C_S_AXI_MEM1_HIGHADDR	Read/Write
Bank 2[1]	C_S_AXI_MEM2_BASEADDR	C_S_AXI_MEM2_HIGHADDR	Read/Write
Bank 3[1]	C_S_AXI_MEM3_BASEADDR	C_S_AXI_MEM3_HIGHADDR	Read/Write
Registers[2]	C_S_AXI_REG_BASEADDR	C_S_AXI_REG_HIGHADDR	Read/Write

Notes:

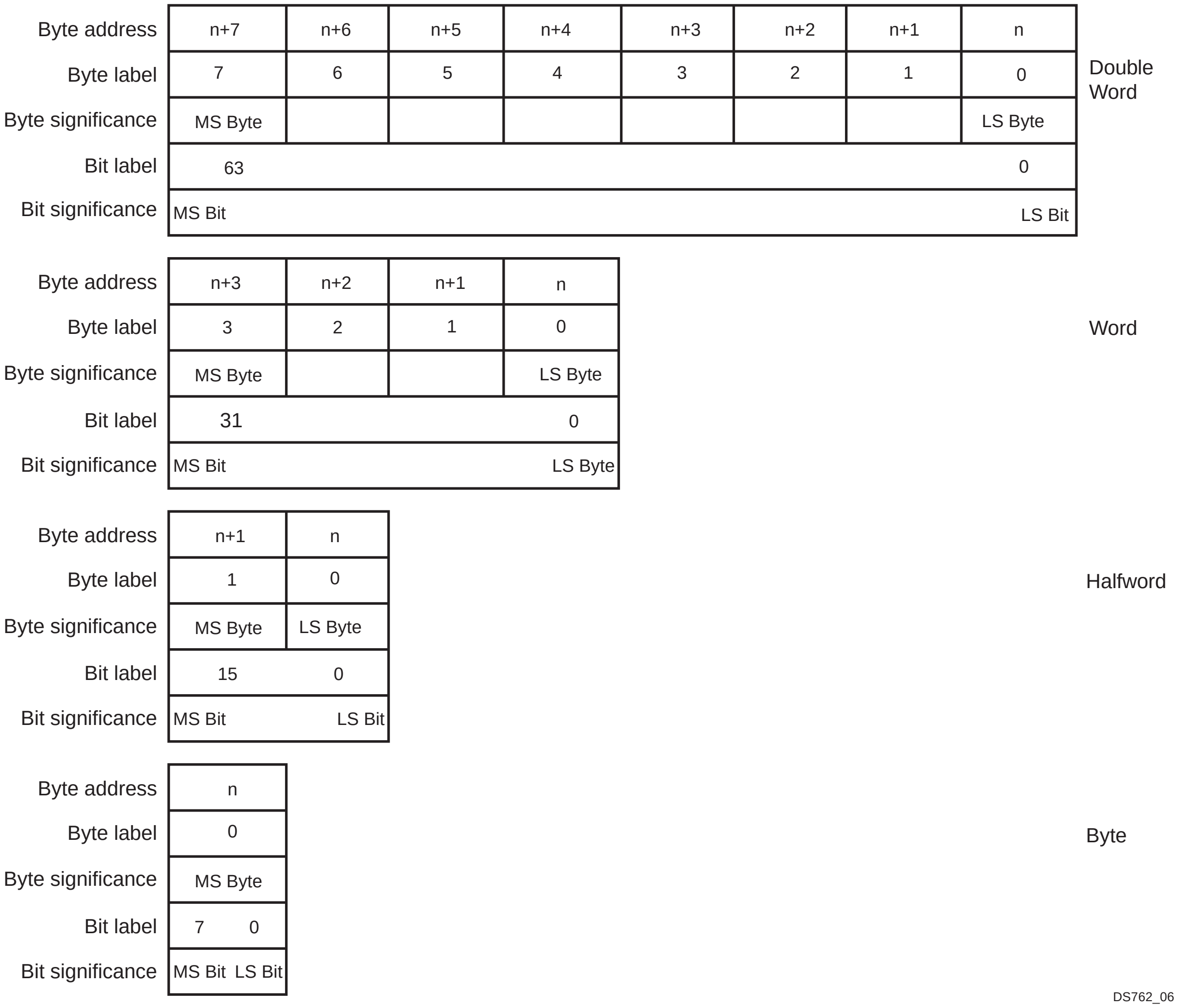
1. The number of memory banks depends on C_NUM_BANKS_MEM value
2. Register block is included when C_S_AXI_EN_REG = 1

Memory Data Types and Organization

Memory can be accessed through the AXI EMC core as one of four types:

- Byte (8-bit)
- Halfword (16-bit)
- Word (32-bit)
- Doubleword (64-bit)

Data to and from the AXI interface is organized as little-endian. The bit and byte labeling for the big-endian data types is shown in Figure 3-1.



Double Word

Byte address	n+7	n+6	n+5	n+4	n+3	n+2	n+1	n
Byte label	7	6	5	4	3	2	1	0
Byte significance	MS Byte							LS Byte
Bit label	63							0
Bit significance	MS Bit							LS Bit

Word

Byte address	n+3	n+2	n+1	n
Byte label	3	2	1	0
Byte significance	MS Byte			LS Byte
Bit label	31			0
Bit significance	MS Bit			LS Byte

Halfword

Byte address	n+1	n
Byte label	1	0
Byte significance	MS Byte	LS Byte
Bit label	15	0
Bit significance	MS Bit	LS Bit

Byte

Byte address	n
Byte label	0
Byte significance	MS Byte
Bit label	7 0
Bit significance	MS Bit LS Bit

DS762_06

Figure 3-1: **Memory Data Types**

Connecting to Memory

Clocking Synchronous Memory

The AXI EMC core does not provide a clock output to any synchronous memory. The AXI clock should be routed through an output buffer to provide the clock to synchronous memory. To synchronize the synchronous memory clock to the internal FPGA clock, the FPGA system design should include a Digital Clock Manager (DCM), external to the AXI EMC core, that uses the synchronous memory clock input as the feedback clock as shown in Figure 3-2. This means that the synchronous clock output from the FPGA must be routed back to the FPGA on a clock pin.

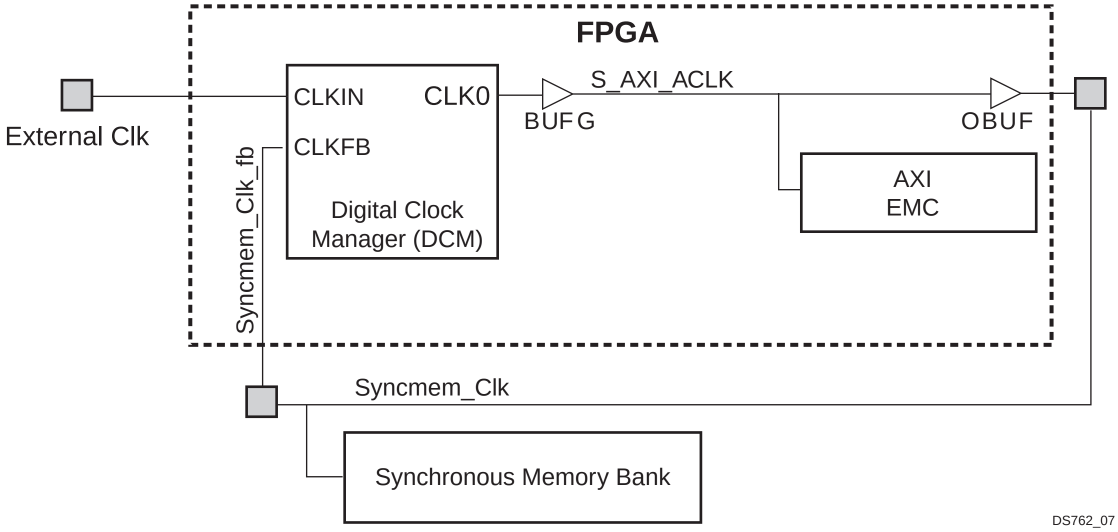


Figure 3-2: **Synchronous Memory Bank Clocked by FPGA Output With Feedback**

If the synchronous memory is clocked by the same external clock as the FPGA, or if the clock feedback is not available, the DCM shown in Figure 3-3 should be included in the FPGA external to the AXI EMC core.

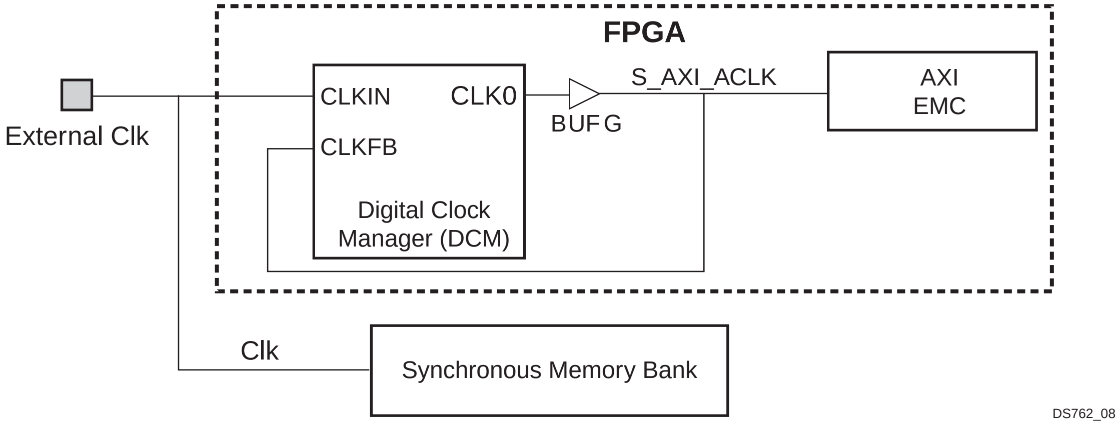


Figure 3-3: **Synchronous Memory Bank Clocked by External Clock**

Address Bus, Data Bus and Control Signal Connections

The three primary considerations for connecting the controller to memory devices are the width of the AXI data bus, the width of the memory subsystem and the number of memory devices used. The data and address signals at the memory controller are labeled with little-endian bit labeling (for example, D(31:0) where D(31) is the MSB), and memory devices are also little-endian.

Note: Most asynchronous memory devices only use MEM_CEN, while most synchronous memory devices use both MEM_CEN and MEM_CE. MEM_CEN is a function of the address decode while MEM_CE is a function of the state machine logic.

Avoid incorrect data and address connections to the external memory devices by using Tables 3-3, 3-5, 3-7, 3-9, 3-11, 3-13 and 3-15 which show the correct mapping of memory controller pins to memory device pins.

Table 3-2 shows variables used in defining memory subsystem and Table 3-3 shows interconnection of AXI EMC signals to memory interface signals.

Table 3-2: **Variables Used in Defining Memory Subsystem**

Variable	Allowed Range	Definition
BN	3 down to 0	Memory bank number
DN	3 down to 0	Memory device number within a bank. The memory device attached to the most significant bit in the memory subsystem is 0 to 3.
MW	8 to 64	Width in bits of memory subsystem (8, 16, 32, and 64 are allowed values)
DW	63 down to 1	Width in bits of data bus for memory device
MAW	32 down to 1	Width in bits of address bus for memory device
AU	63 down to 1	Width in bits of smallest addressable data word on the memory device
AS	X[1]	Address shift for address bus = $\log_2((MW*AU/DW)/8)$

Table 3-2: **Variables Used in Defining Memory Subsystem *(Cont'd)***

Variable	Allowed Range	Definition
HAW	32 down to 1	Width in bits of AXI address bus

Notes:

1. The value of X depends on variables MW, AU and DW.

Connecting to SRAM

Table 3-3: **Core To Memory Interconnect**

Description	AXI EMC Signals (MSB:LSB)	Memory Device Signals (MSB:LSB)
Data bus	MEM_DQ(((DN+1)*DW)-1:DN*DW)	D(DW-1 : 0)
Address bus	MEM_A(MAW-AS-1:AS)	A(MAW-1 : 0)
Chip enable (active Low)	MEM_CEN(BN)	CEN
Output enable (active Low)	MEM_OEN	OEN
Write enable (active Low)	MEM_WEN	WEN (for devices that have byte enables)
Qualified write enable (active Low)	MEM_QWEN(DN*DW/8)	WEN (for devices that do not have byte enables)
Byte enable (active Low)	MEM_BEN((((DN+1)*DW/8)-1):(DN*DW/8))	BEN(DW/8-1 : 0)

Example 1: Connection to 32-bit memory using two IDT71V416S SRAM parts.

Table 3-4 shows variables for a simple SRAM example.

Table 3-4: **Variables For Simple SRAM Example**

Variable	Value	Definition
BN	0	Memory bank number
DN	1 down to 0	Memory device number within a bank. The memory device attached to the most significant bit in the memory subsystem is 0; device numbers increase toward the least significant bit.
MW	32	Width in bits of memory subsystem
DW	16	Width in bits of data bus for memory device
MAW	18	Width in bits of address bus for memory device
AU	16	Width in bits of smallest addressable data word on the memory device
AS	2	Address shift for address bus = $\log_2$((MW*AU/DW)/8)
HAW	32	Width in bits of host address bus (for example, AXI)

Table 3-5 shows connection to 32-bit memory using two IDT71V416S (256K X 16-Bit) parts.

Table 3-5: **Connection to 32-bit Memory Using Two IDT71V416S Parts**

DN	Description	AXI EMC Signals (MSB:LSB)	Memory Device Signals (MSB:LSB)
0	Data bus	MEM_DQ(15 : 0)	I/O(15 : 0)
	Address bus	MEM_A(19 : 2)	A(17 : 0)
	Chip enable (active Low)	MEM_CEN(0)	CS
	Output enable (active Low)	MEM_OEN	OE
	Write enable (active Low)	MEM_WEN	WE
	Byte enable (active Low)	MEM_BEN(1 : 0)	$\overline{BHE}$:$\overline{BLE}$
1	Data bus	MEM_DQ(31 : 16)	I/O(15 : 0)
	Address bus	MEM_A(19 : 2)	A(17 : 0)
	Chip enable (active Low)	MEM_CEN(0)	CS
	Output enable (active Low)	MEM_OEN	OE
	Write enable (active Low)	MEM_WEN	WE
	Byte enable (active Low)	MEM_BEN(3 : 2)	$\overline{BHE}$:$\overline{BLE}$

Connecting to Byte Parity Memory

Connection to 32-bit memory using two IS61LVPS25636A Asynchronous SRAM parts

Table 3-6 shows the variables for a simple SRAM example.

Table 3-6: **Variables for Simple SRAM Example**

Variable	Value	Definition
BN	0	Memory bank number
DN	0	Memory device number within a bank. The memory device attached to the most significant bit in the memory subsystem is 0; device numbers increase toward the least significant bit.
MW	32	Width in bits of memory subsystem
DW	32	Width in bits of data bus for memory device
MAW	18	Width in bits of address bus for memory device
AU	32	Width in bits of smallest addressable data word on the memory device
AS	2	Address shift for address bus = $\log_2$((MW*AU/DW)/8)
HAW	32	Width in bits of host address bus (for example, AXI)

Table 3-7 shows the connection to 32-bit memory using two IDT71V416S (256K X 16-Bit) parts.

Table 3-7: **Connection to 32-bit Memory Using Two IDT71V416S Parts**

DN	Description	AXI EMC Signals (MSB:LSB)	Memory Device Signals (MSB:LSB)
0	Data bus	MEM_DQ(31 : 0)	I/O(35 : 0)
	Address bus	MEM_A(19 : 2)	A(17 : 0)
	Chip enable (active Low)	MEM_CEN(0)	CS
	Output enable (active Low)	MEM_OEN	OE
	Write enable (active Low)	MEM_WEN	WE
	Byte enable (active Low)	MEM_BEN(3 : 0)	BWAb,BWBb,BWCb,BWDb
	Parity Bits	MEM_DQ_PARITY(3 : 0)	I/O(35 : 32)

Connecting to Intel StrataFlash

StrataFlash parts contain an identifier register, a status register, and a command interface. Therefore, the bit label ordering for these parts is critical to enable correct operation. Table 3-8 shows an example of how to connect the big-endian AXI EMC bus to the little-endian StrataFlash parts. The correct connection ordering is also indicated in a more general form in Table 3-3. StrataFlash parts have an x8 mode and an x16 mode, selectable with the BYTE# input pin. To calculate the proper address shift, the minimum addressable word is 8 bits for both x8 and x16 modes, because A0 always selects a byte.

Example 2: Connection to 32-bit Memory Using two StrataFlash Parts in x16 Mode.

This configuration supports byte read, but not byte write. The smallest data type that can be written is 16-bit data. Table 3-8 shows the variables for StrataFlash (x16 mode) example.

Table 3-8: **Variables For StrataFlash (x16 mode) Example**

Variable	Value	Definition
BN	0	Memory bank number
DN	0 to 1	Memory device number within a bank. The memory device attached to the most significant bit in the memory subsystem is 0; the device numbers increase toward the least significant bit.
MW	32	Width in bits of memory subsystem
DW	16	Width in bits of data bus for memory device
MAW	24	Width in bits of address bus for memory device
AU	8	Width in bits of smallest addressable data word on the memory device
AS	1	Address shift for address bus = $\log_2((MW*AU/DW)/8)$
HAW	32	Width in bits of host address bus (for example, AXI)

Table 3-9 shows the connection to 32-bit memory using two StrataFlash parts.

Table 3-9: **Connection to 32-bit Memory Using Two StrataFlash Parts**

DN	Description	AXI EMC Signals (MSB:LSB)	StrataFlash Signals (MSB:LSB)
0	Data bus	MEM_DQ(15 : 0)	DQ(15 : 0)
	Address bus	MEM_A(24 : 1)	A(23 : 0)
	Chip enable (active Low)	GND,GND,MEM_CEN(0)	CE(2 : 0)
	Output enable (active Low)	MEM_OEN	OE#
	Write enable (active Low)	MEM_QWEN(0)	WE#
	Reset/Power down (active Low)	MEM_RPN	RP#
	Byte mode select (active Low)	N/A - tie to GND	BYTE#
	Program enable (active High)	N/A - tie to VCC	V_{PEN}
1	Data bus	MEM_DQ(31 : 16)	DQ(15 : 0)
	Address bus	MEM_A(24 : 1)	A(23 : 0)
	Chip enable (active Low)	GND, GND, MEM_CEN(0)	CE(2 : 0)
	Output enable (active Low)	MEM_OEN	OE#
	Write enable (active Low)	MEM_QWEN(2)	WE#
	Reset/Power down (active Low)	MEM_RPN	RP#
	Byte mode select (active Low)	N/A - tie to GND	BYTE#
	Program enable (active High)	N/A - tie to VCC	V_{PEN}

Example 3: Connection to 32-bit Memory Using Four StrataFlash Parts in x8 Mode

This configuration supports byte reads and writes. Table 3-10 shows the variables for the StrataFlash (x8 mode) example.

Table 3-10: **Variables for StrataFlash (x8 mode) Example**

Variable	Value	Definition
BN	0	Memory bank number
DN	0 to 3	Memory device number within a bank. The memory device attached to the most significant bit in the memory subsystem is 0; the device numbers increase toward the least significant bit.
MW	32	Width in bits of memory subsystem
DW	8	Width in bits of data bus for memory device
MAW	24	Width in bits of address bus for memory device
AU	8	Width in bits of smallest addressable data word on the memory device
AS	2	Address shift for address bus = $\log_2$((MW*AU/DW)/8)
HAW	32	Width in bits of host address bus (for example, AXI)

Table 3-11 shows the connection to 32-bit memory using four StrataFlash parts.

Table 3-11: **Connection to 32-bit Memory Using Four StrataFlash Parts**

DN	Description	AXI EMC Signals (MSB:LSB)	StrataFlash Signals (MSB:LSB)
0	Data bus	MEM_DQ(7 : 0)	DQ(7:0)[1]
	Address bus	MEM_A(23 : 2)	A(21:0)
	Chip enable (active Low)	GND, GND, MEM_CEN(0)	CE(2:0)
	Output enable (active Low)	MEM_OEN	OE#
	Write enable (active Low)	MEM_QWEN(0)	WE#
	Reset/Power down (active Low)	MEM_RPN	RP#
	Byte mode select (active Low)	N/A - tie to GND	BYTE#
	Program enable (active High)	N/A - tie to VCC	V_{PEN}
1	Data bus	MEM_DQ(15 : 8)	DQ(7:0) ()
	Address bus	MEM_A(23 : 2)	A(21:0)
	Chip enable (active Low)	GND,GND, MEM_CEN(0)	CE(2:0)
	Output enable (active Low)	MEM_OEN	OE#
	Write enable (active Low)	MEM_QWEN(1)	WE#
	Reset/Power down (active Low)	MEM_RPN	RP#
	Byte mode select (active Low)	N/A - tie to GND	BYTE#
	Program enable (active High)	N/A - tie to VCC	V_{PEN}
2	Data bus	MEM_DQ(23 : 16)	DQ(7:0)[1]
	Address bus	MEM_A(23 : 2)	A(21:0)
	Chip enable (active Low)	GND, GND, MEM_CEN(0)	CE(2:0)
	Output enable (active Low)	MEM_OEN	OE#
	Write enable (active Low)	MEM_QWEN(2)	WE#
	Reset/Power down (active Low)	MEM_RPN	RP#
	Byte mode select (active Low)	N/A - tie to GND	BYTE#
	Program enable (active High)	N/A - tie to VCC	V_{PEN}

Table 3-11: **Connection to 32-bit Memory Using Four StrataFlash Parts *(Cont'd)***

DN	Description	AXI EMC Signals (MSB:LSB)	StrataFlash Signals (MSB:LSB)
3	Data bus	MEM_DQ(31 : 24)	DQ(7:0)[1]
	Address bus	MEM_A(23 : 2)	A(21:0)
	Chip enable (active Low)	GND, GND, MEM_CEN(0)	CE(2:0)
	Output enable (active Low)	MEM_OEN	OE#
	Write enable (active Low)	MEM_QWEN(3)	WE#
	Reset/Power down (active Low)	MEM_RPN	RP#
	Byte mode select (active Low)	N/A - tie to GND	BYTE#
	Program enable (active High)	N/A - tie to VCC	V_{PEN}

Notes:

1. In an x8 configuration, DQ(15:8) are not used and should be treated according to the manufacturer's data sheet.

Connecting to Spansion Page Mode Flash S29GL032N

Table 3-12 shows the variables for the Spansion Page Mode Flash(x16 mode) example.

Table 3-12: **Variables for Page Mode Flash (x16 mode) Example**

Variable	Value	Definition
BN	0	Memory bank number
DN	0	Memory device number within a bank. The memory device attached to the most significant bit in the memory subsystem is 0.
MW	32	Width in bits of memory subsystem
DW	16	Width in bits of data bus for memory device
MAW	21	Width in bits of address bus for memory device
AU	16	Width in bits of smallest addressable data word on the memory device
AS	1	Address shift for address bus = $\log_2((MW*AU/DW)/8)$
HAW	32	Width in bits of host address bus (for example, AXI)

Table 3-13: **Connection to 16-bit Memory Using Page Mode Flash Parts**

DN	Description	AXI EMC Signals (MSB:LSB)	PagemModeFlash Signals (MSB:LSB)
0	Data bus	MEM_DQ(15 : 0)	DQ(15 : 0)
	Address bus	MEM_A(22 : 1)	A(21 : 0)
	Chip enable (active Low)	MEM_CEN(0)	CE
	Output enable (active Low)	MEM_OEN	OE#
	Write enable (active Low)	MEM_QWEN(0)	WE#
	Reset/Power down (active Low)	MEM_RPN	RP#
	Byte mode select (active Low)	N/A - tie to VCC	BYTE#
	Program enable (active High)	N/A - tie to VCC	WP#

Connecting to PSRAM

Table 3-14: **Variables PSRAM (x16 mode) Example**

Variable	Value	Definition
BN	0	Memory bank number
DN	0	Memory device number within a bank. The memory device attached to the most significant bit in the memory subsystem is 0.
MW	32	Width in bits of memory subsystem
DW	16	Width in bits of data bus for memory device
MAW	23	Width in bits of address bus for memory device
AU	16	Width in bits of smallest addressable data word on the memory device
AS	1	Address shift for address bus = $\log_2((MW*AU/DW)/8)$
HAW	32	Width in bits of host address bus (for example, AXI)

Table 3-15: **Connection to 16-bit Memory Using PSRAM Parts**

DN	Description	AXI EMC Signals (MSB:LSB)	PageModeFlash Signals (MSB:LSB)
0	Data bus	MEM_DQ(15 : 0)	DQ(15 : 0)
	Address bus	MEM_A(22 : 1)	A(21 : 0)
	Chip enable (active Low)	MEM_CEN(0)	CE
	Output enable (active Low)	MEM_OEN	OE#
	Write enable (active Low)	MEM_QWEN(0)	WE#
	Reset/Power down (active Low)	MEM_RPN	RP#
	Byte Enable (active Low)	Mem_BEN(1:0)	UB#, LB#
	Control Register Enable (active High)	Mem_CRE	CRE

Software Considerations

Absolute memory addresses are frequently required for specifying command sequences to flash memory devices. Due to the memory address shift (AS), any absolute addresses that must appear on the memory device address bus must also have the source AXI address left-shifted to compensate for the AS.

Timing Diagrams

Synch SRAM Timing

Figure 3-4 shows the timing waveform for 32 bit write for synchronous SRAM.

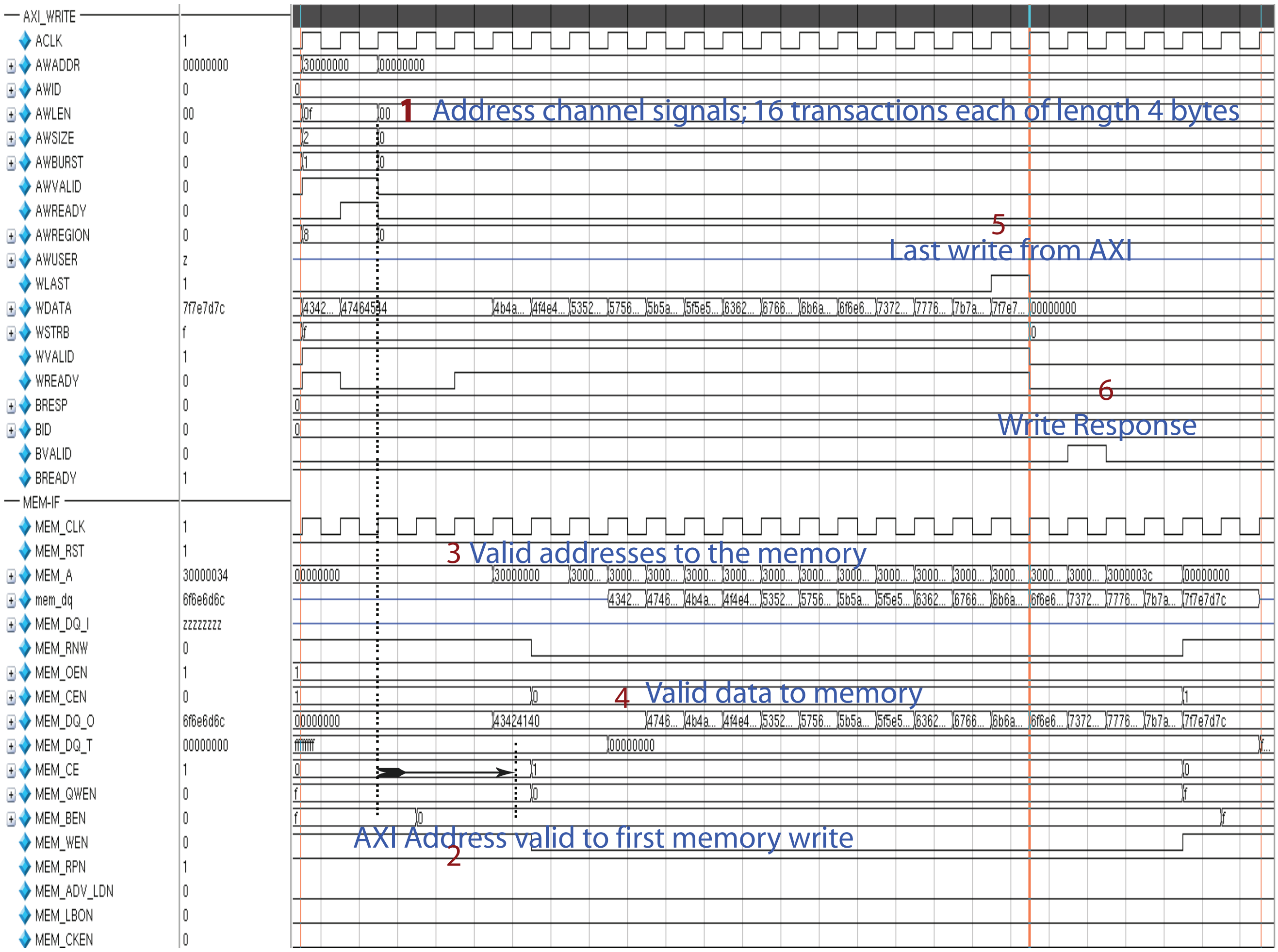
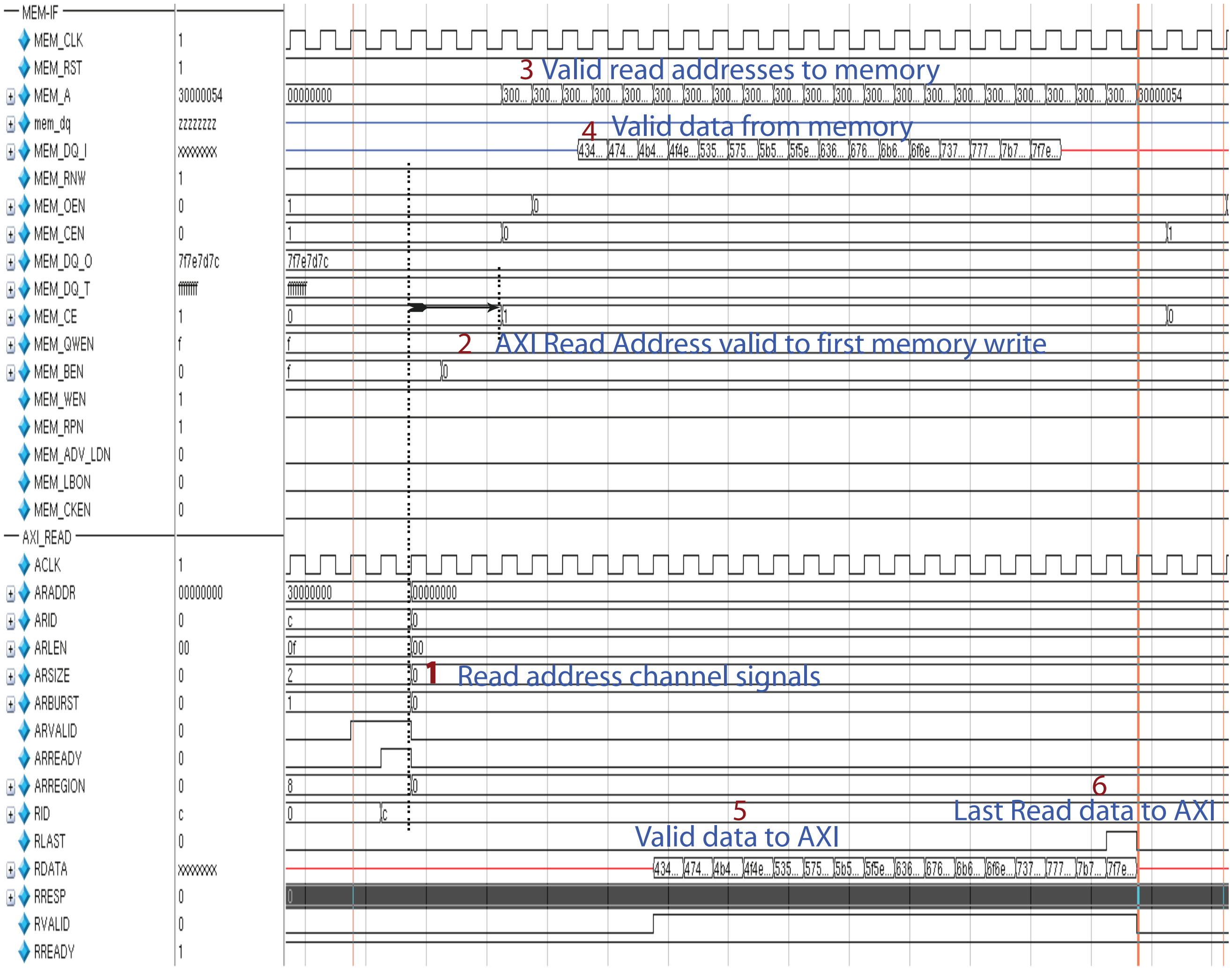


Figure 3-4: **32-Bit Write Synch SRAM with Pipeline Delay 2**

Figure 3-5 shows the timing waveform for 32 bit read for a synchronous SRAM.



DS762_10

Figure 3-5: **32 Bit Read Synch SRAM With Pipeline Delay 2**

Figure 3-6 shows the timing waveform for 8 bit write for synchronous SRAM.

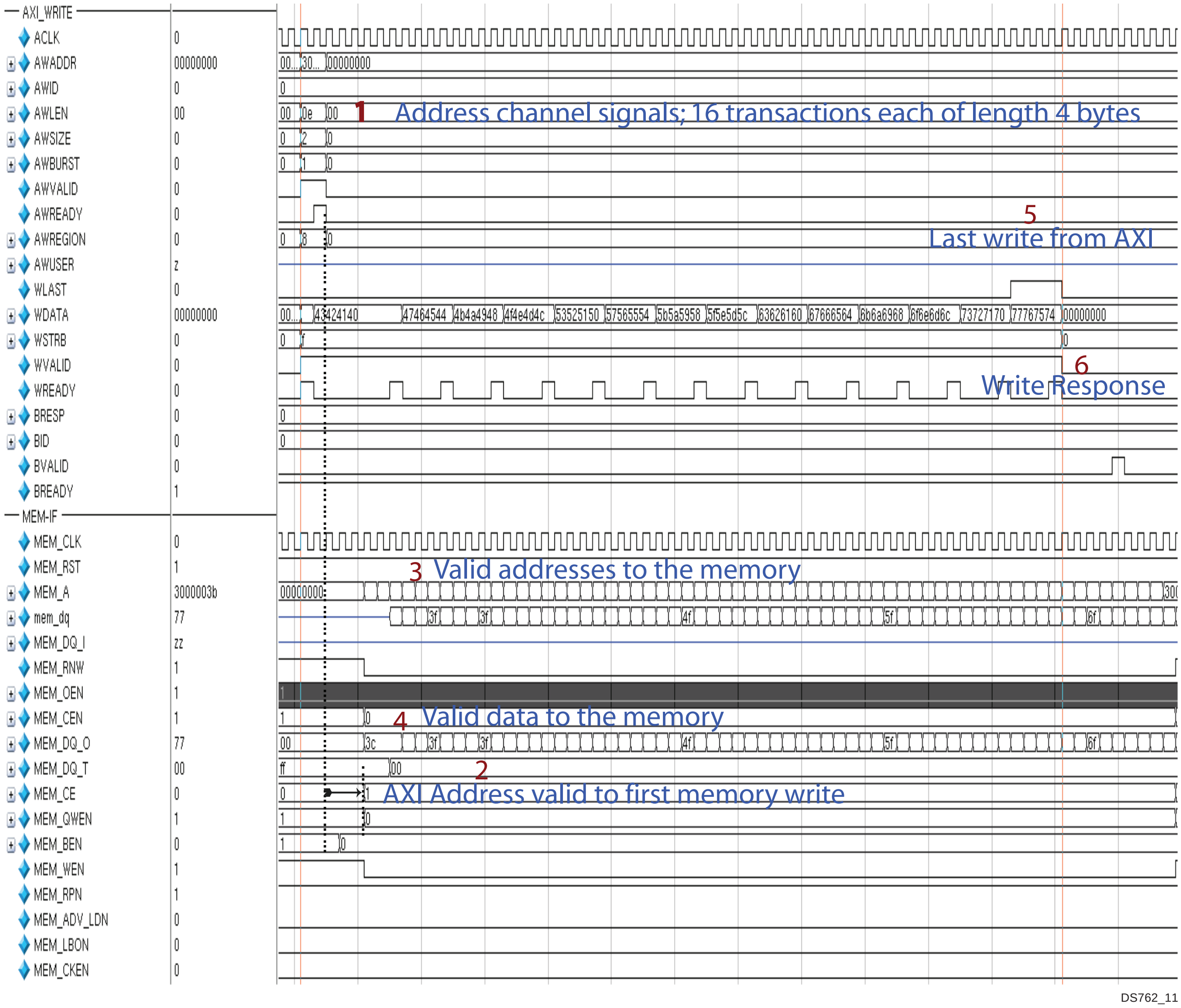


Figure 3-6: **8 Bit Write Synch SRAM with Pipeline Delay 2**

Figure 3-7 shows the timing waveform for 8 bit read for synchronous SRAM.

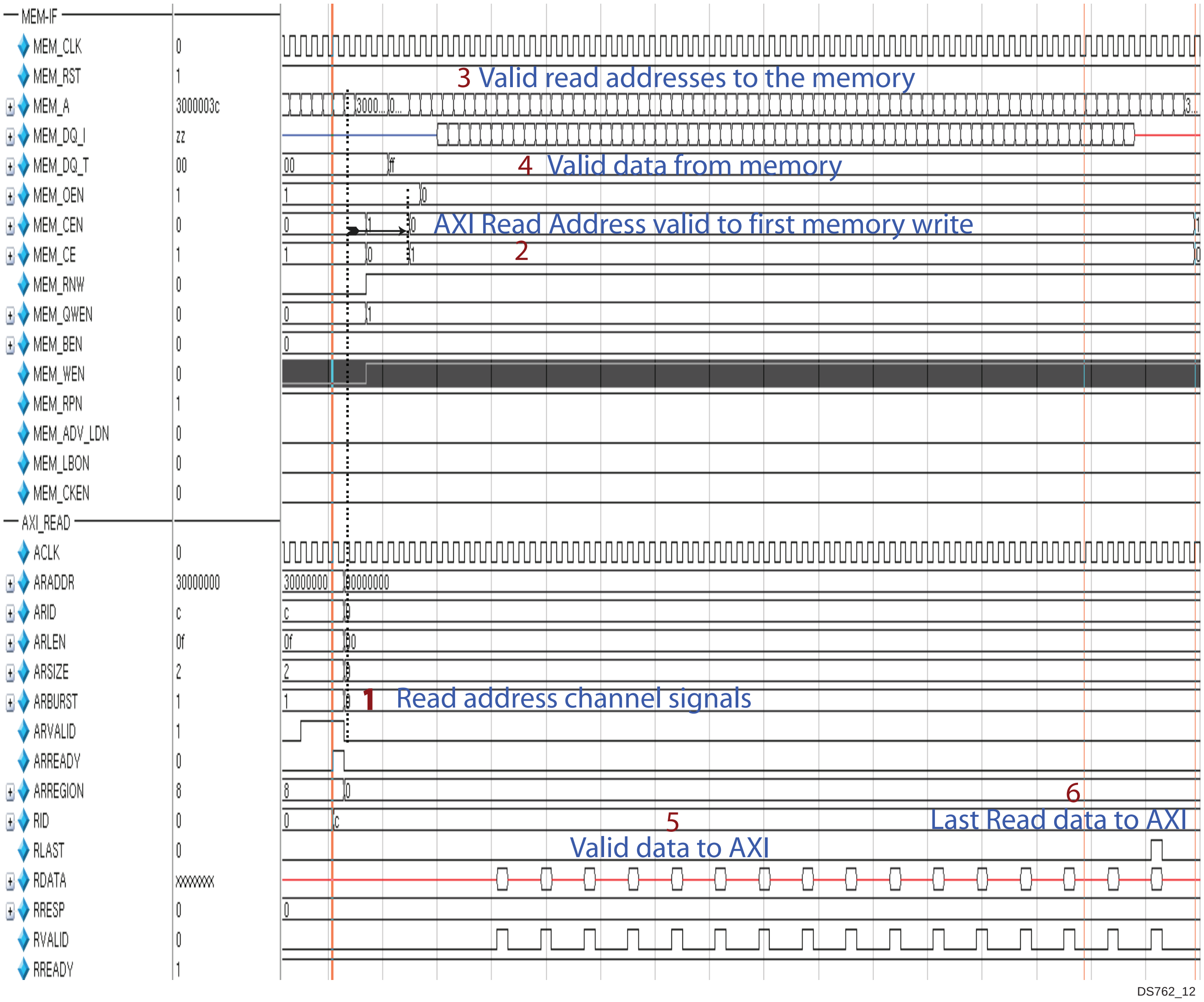


Figure 3-7: **8 Bit Read Synch SRAM with Pipeline Delay 2**

Figure 3-8 shows the timing waveform for 16 bit write for synchronous SRAM.

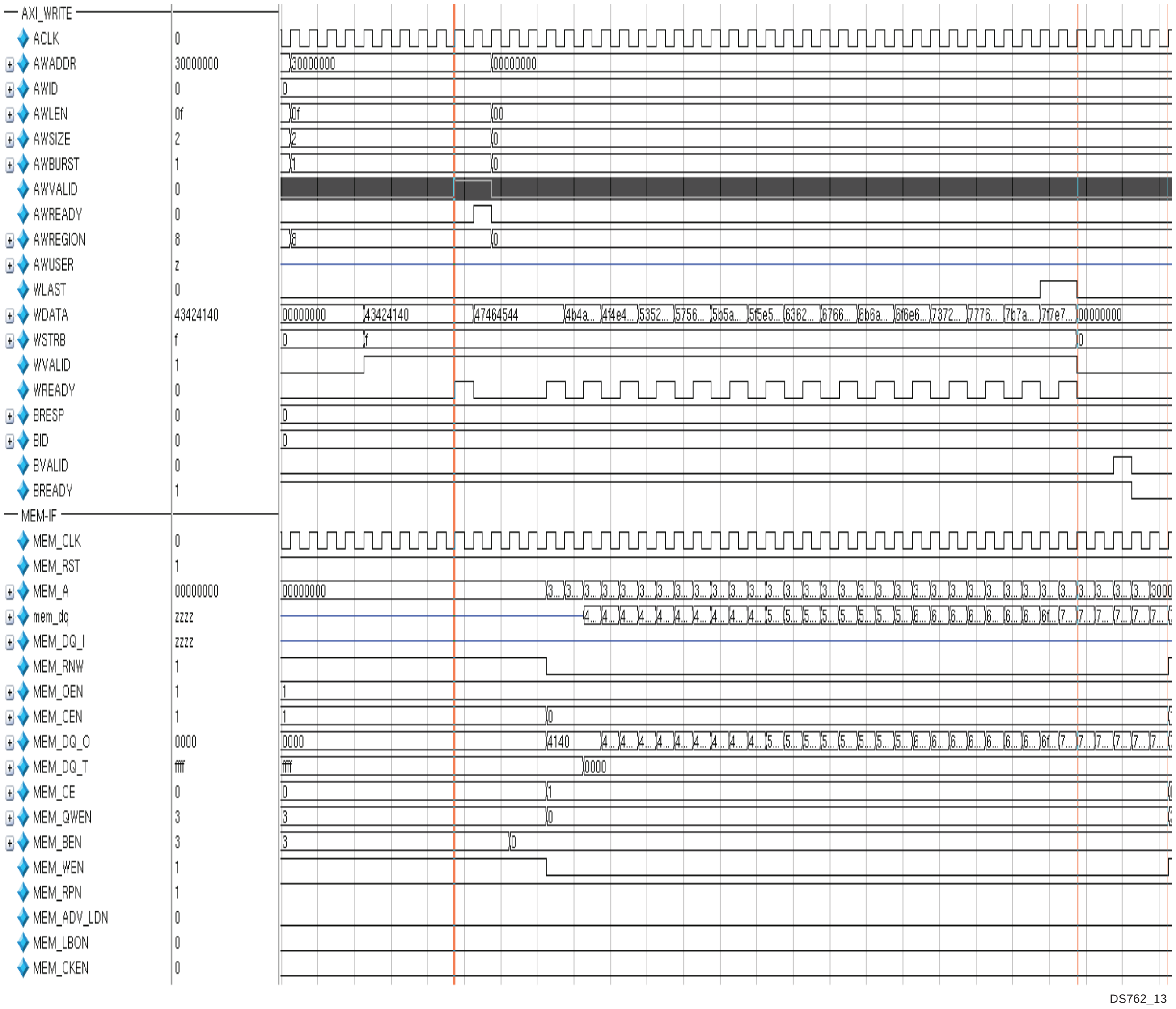


Figure 3-8: **16 Bit Write Synch SRAM With Pipeline Delay 1**

Figure 3-9 shows the timing waveform for 16 bit read for synchronous SRAM.

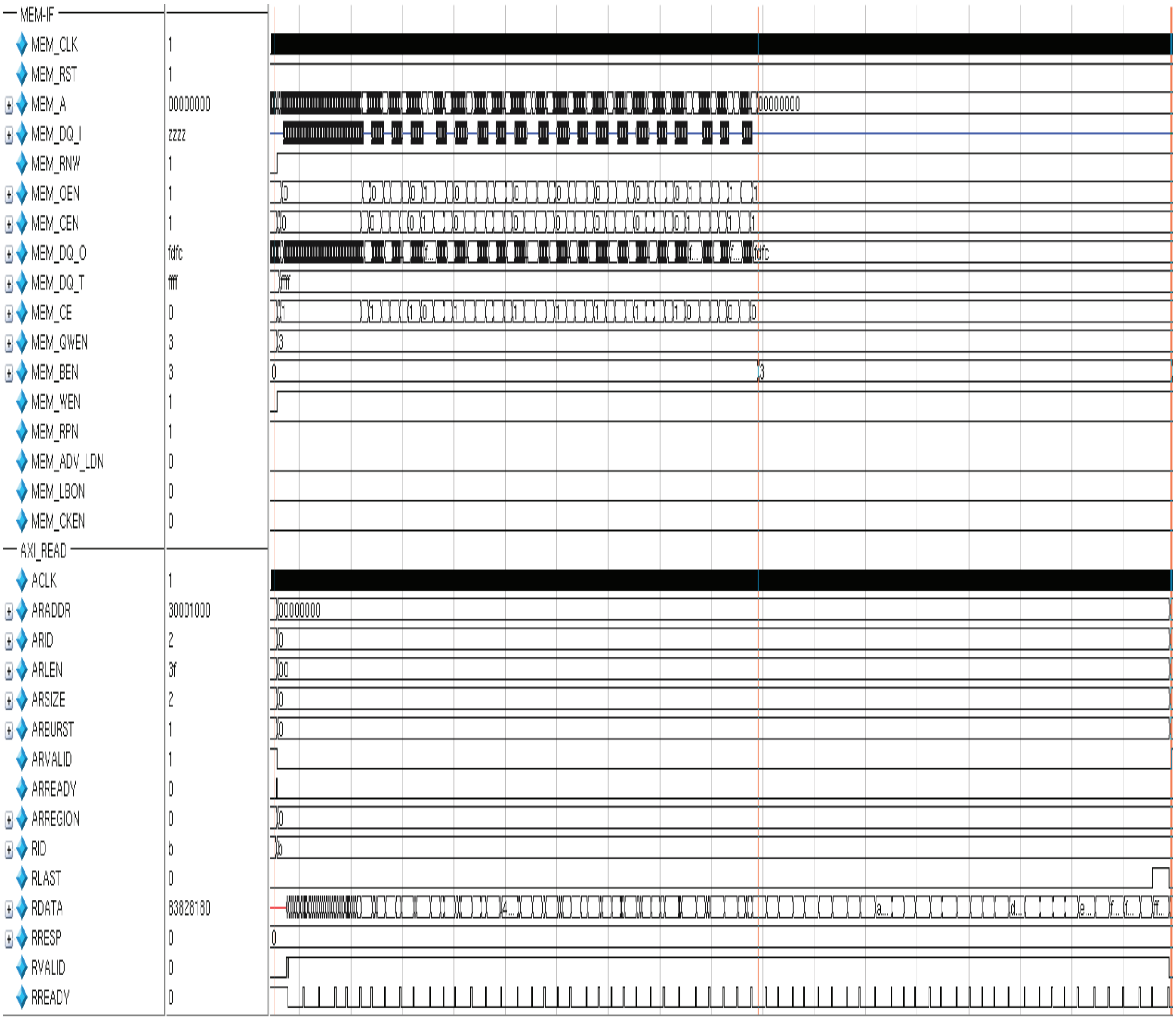
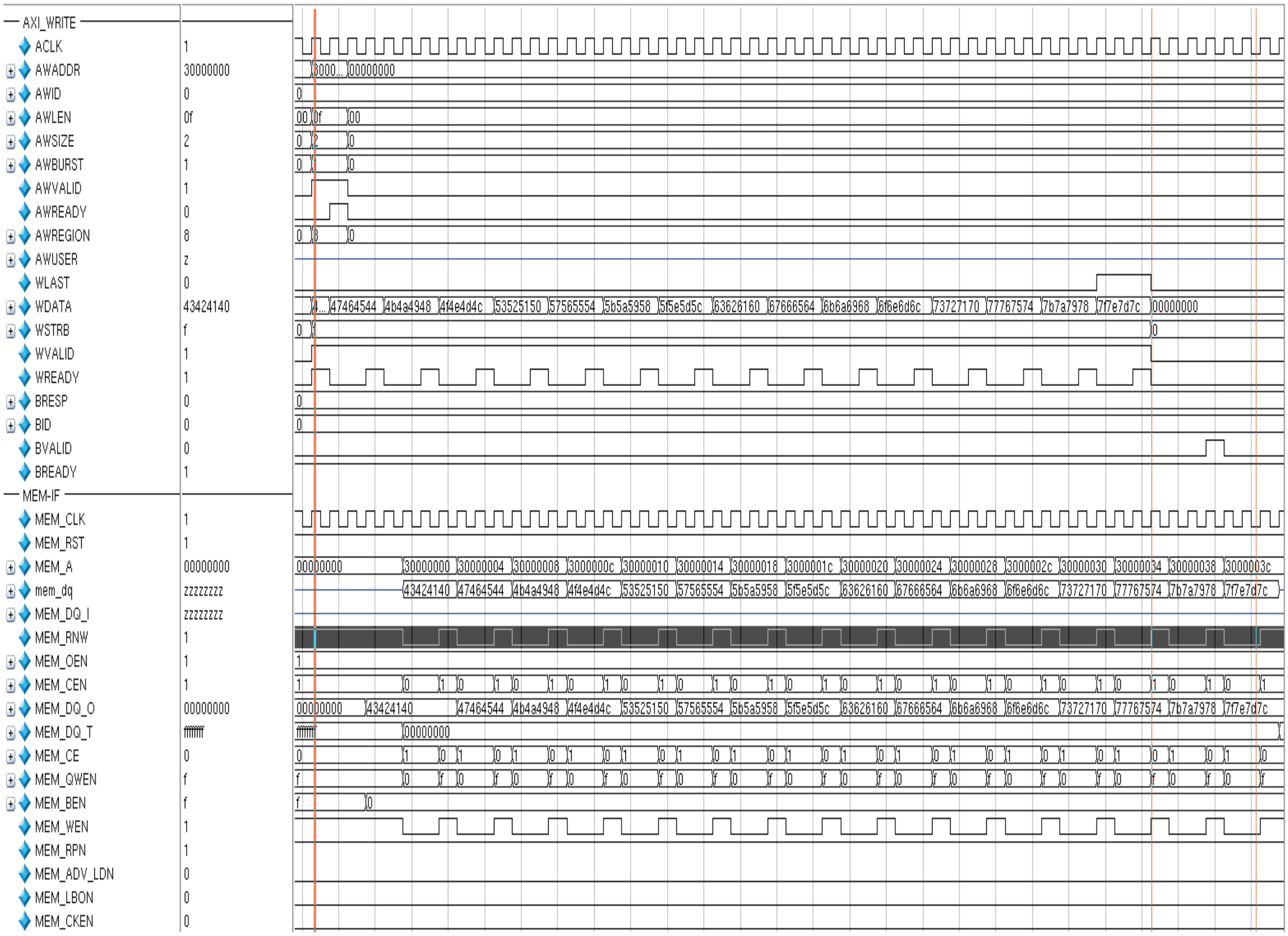


Figure 3-9: **16 Bit Read Synch SRAM for S_AXI_MEM_ARVALID Throttle**

Asynchronous SRAM Timing

Figure 3-10 shows the timing waveform for 32 bit write for asynchronous SRAM.



DS762_15

Figure 3-10: **32 Bit Write Asynchronous SRAM**

Figure 3-11 shows the timing waveform for 32 bit read for asynchronous SRAM.

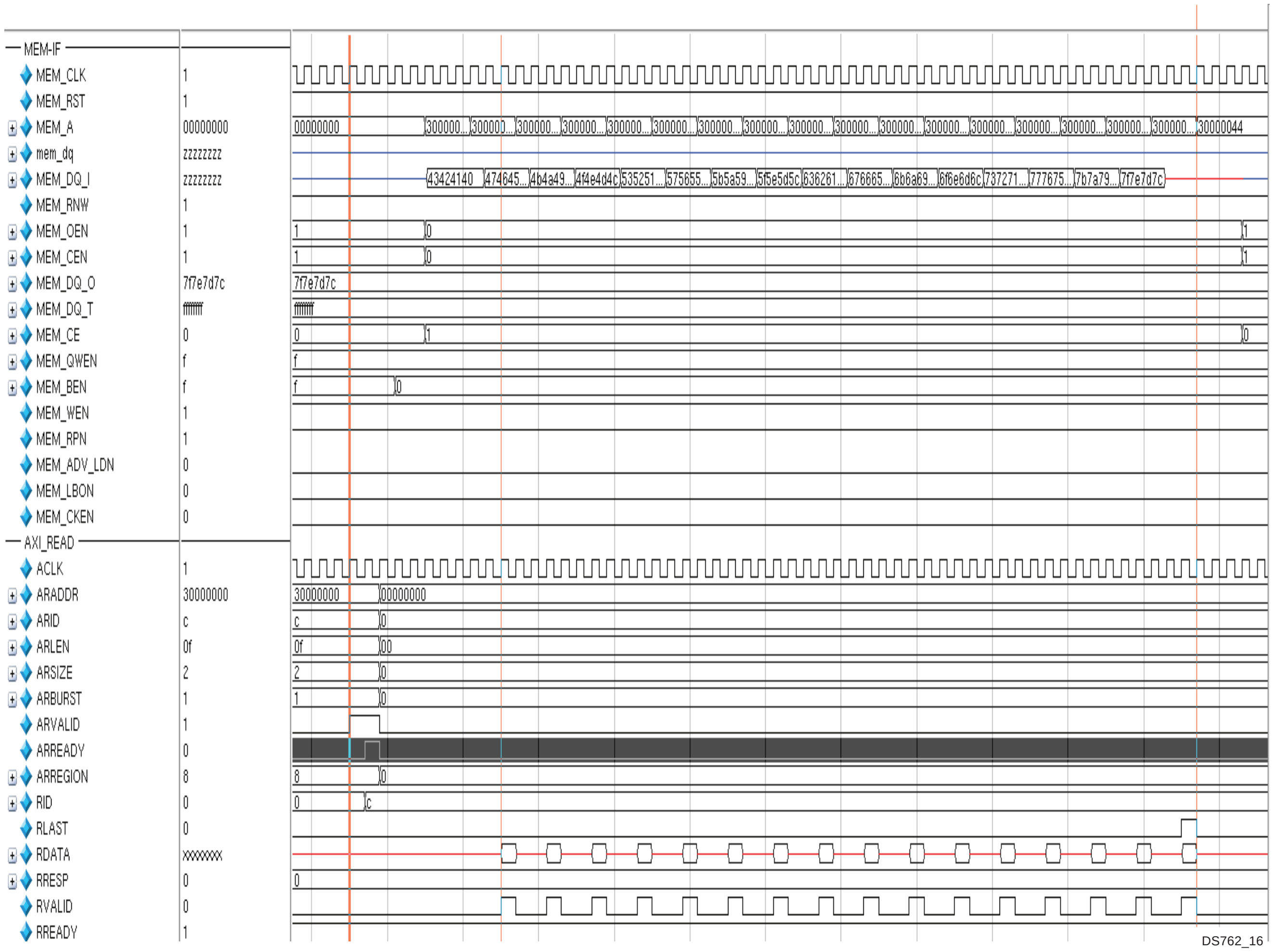
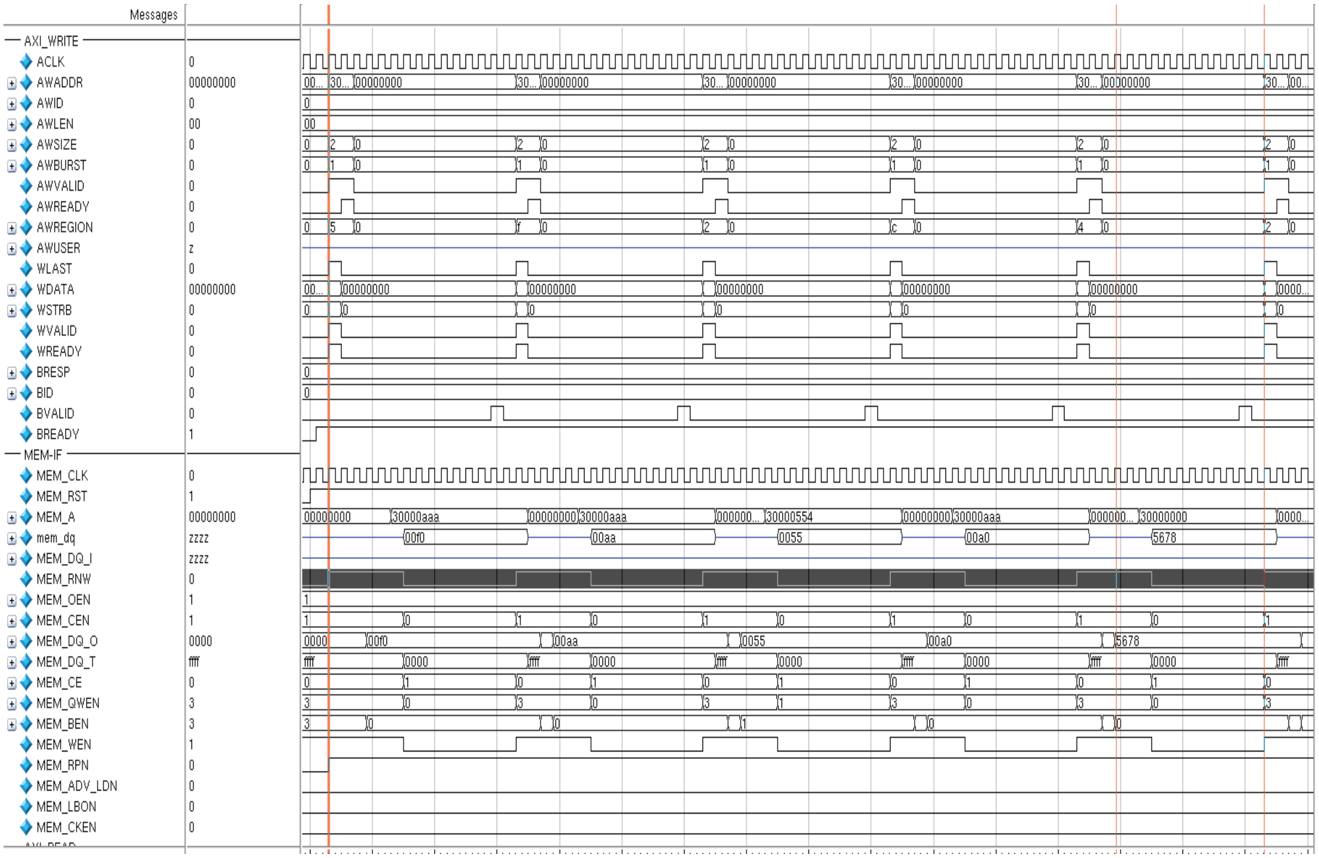


Figure 3-11: **32 Bit Read Asynchronous SRAM**

Flash Timing

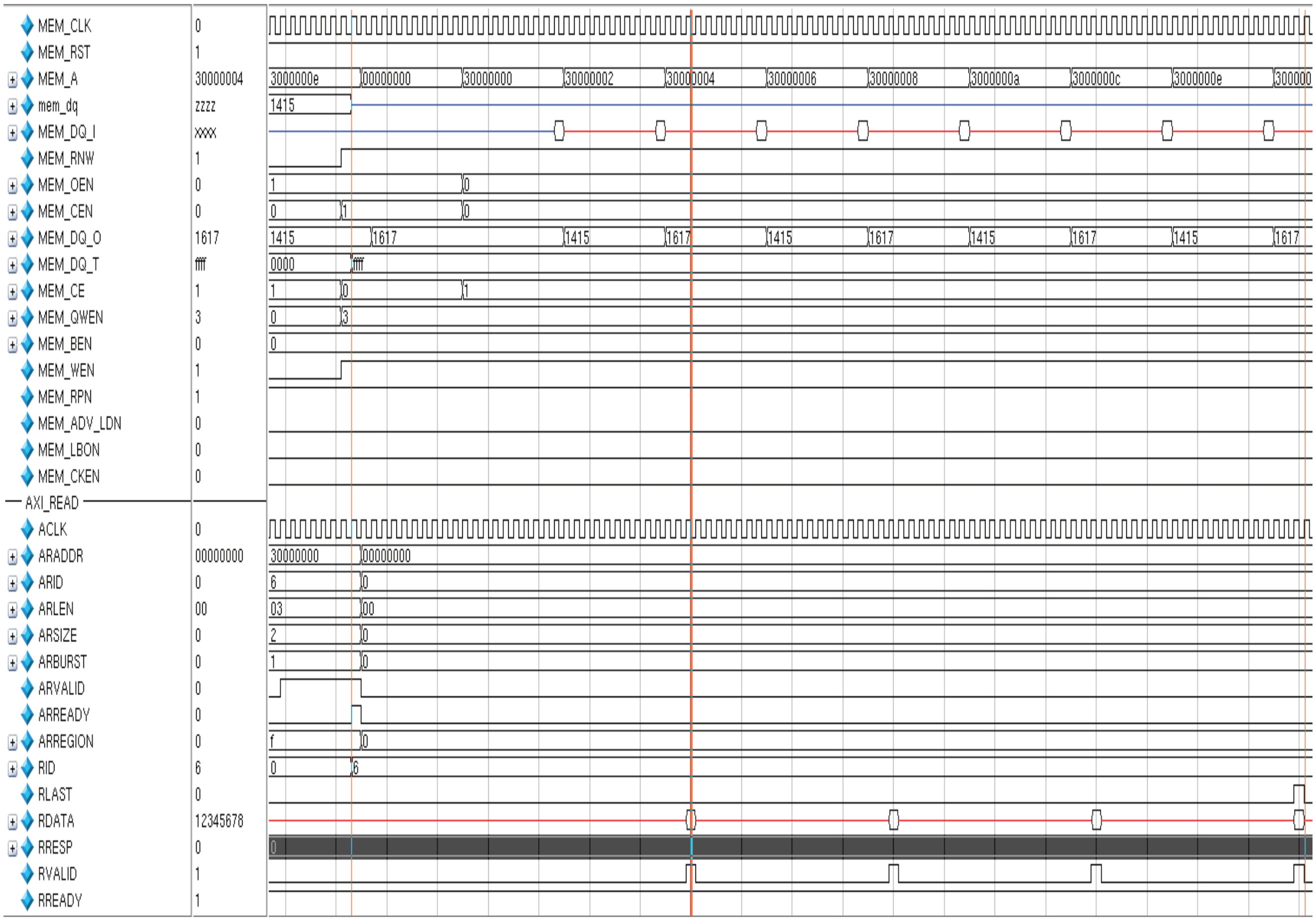
Figure 3-12 shows the timing waveform for 16 bit write for flash memory.



DS762_17

Figure 3-12: **16 Bit Write Flash**

Figure 3-13 shows the timing waveform for 16 bit read for flash memory.

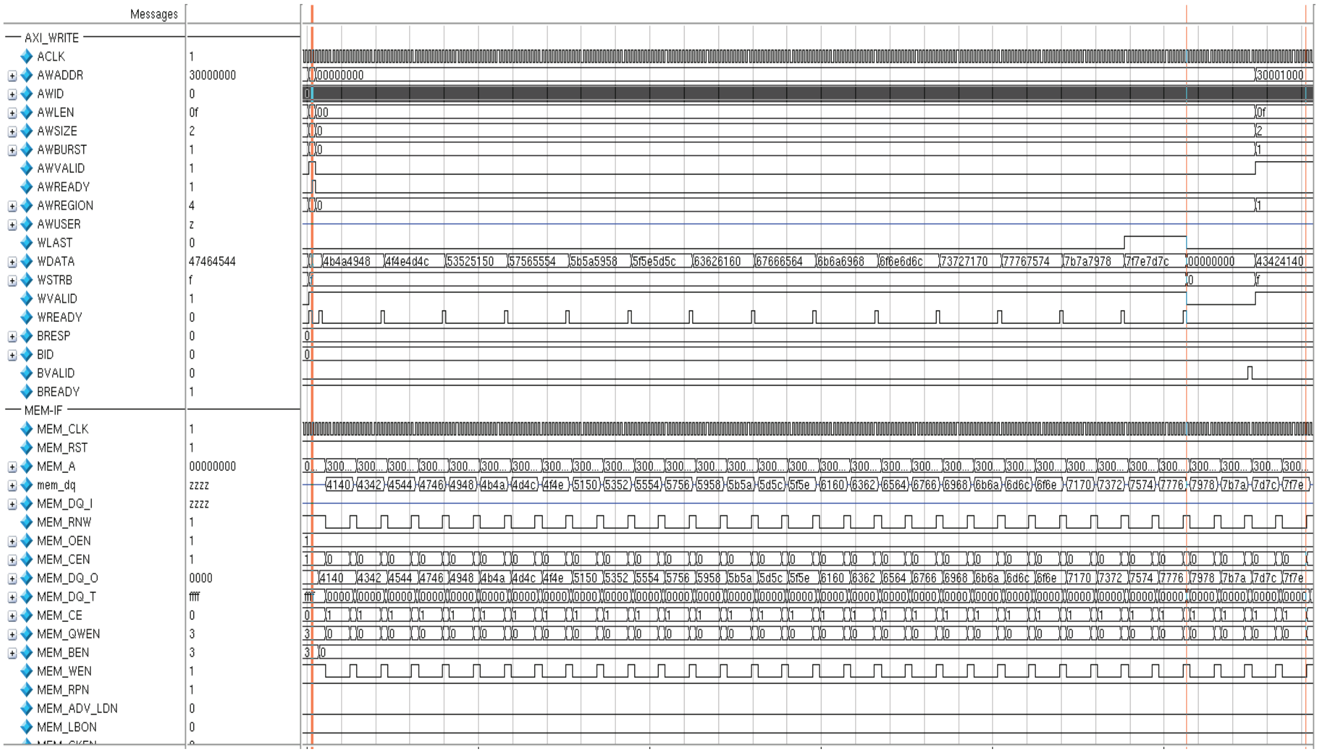


DS762_18

Figure 3-13: **16 Bit Read Flash**

PSRAM/Cellular RAM Timing

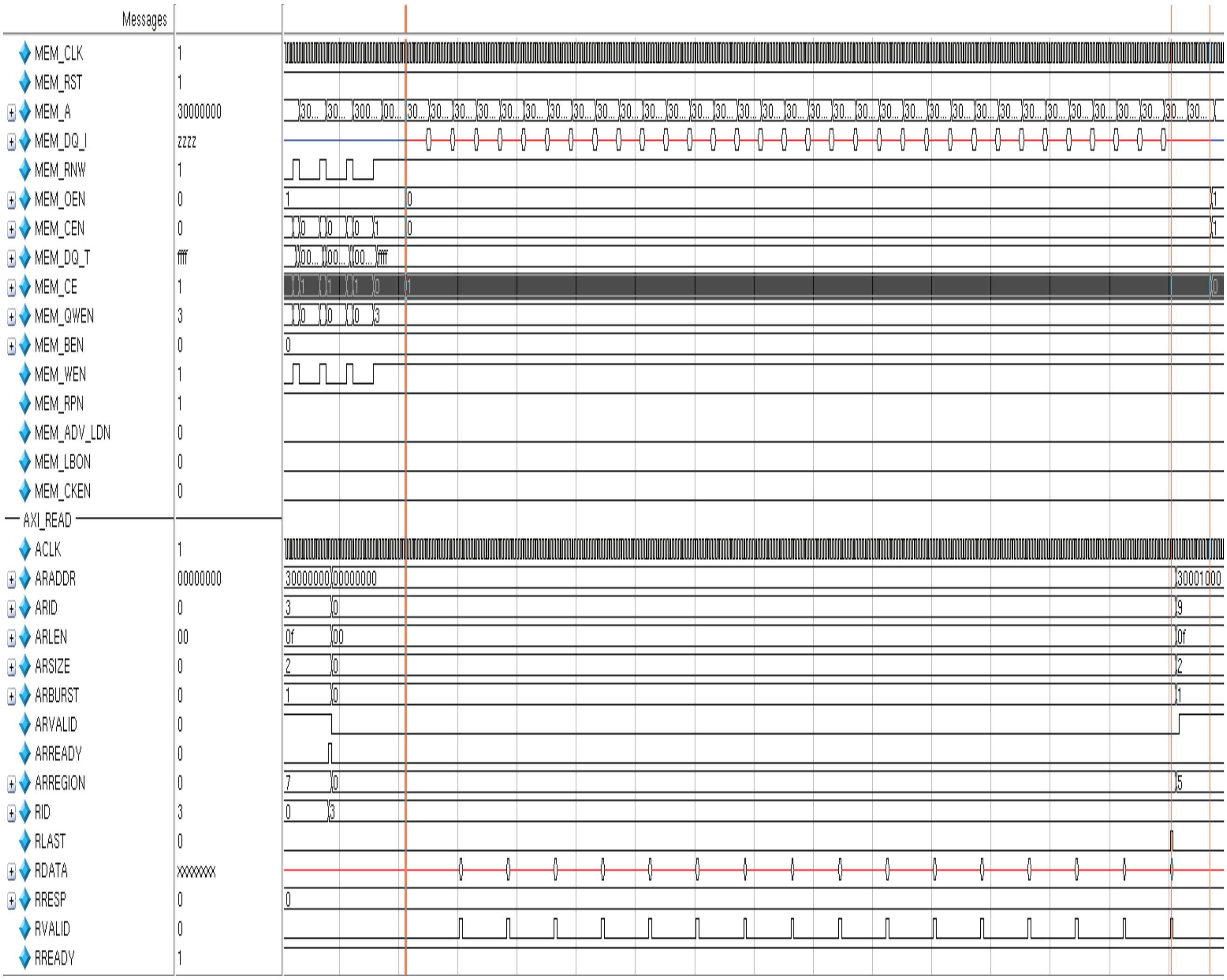
Figure 3-14 shows the timing waveform for 16 bit write for Cellular RAM memory.



DS762_19

Figure 3-14: **16 Bit Write Cellular RAM**

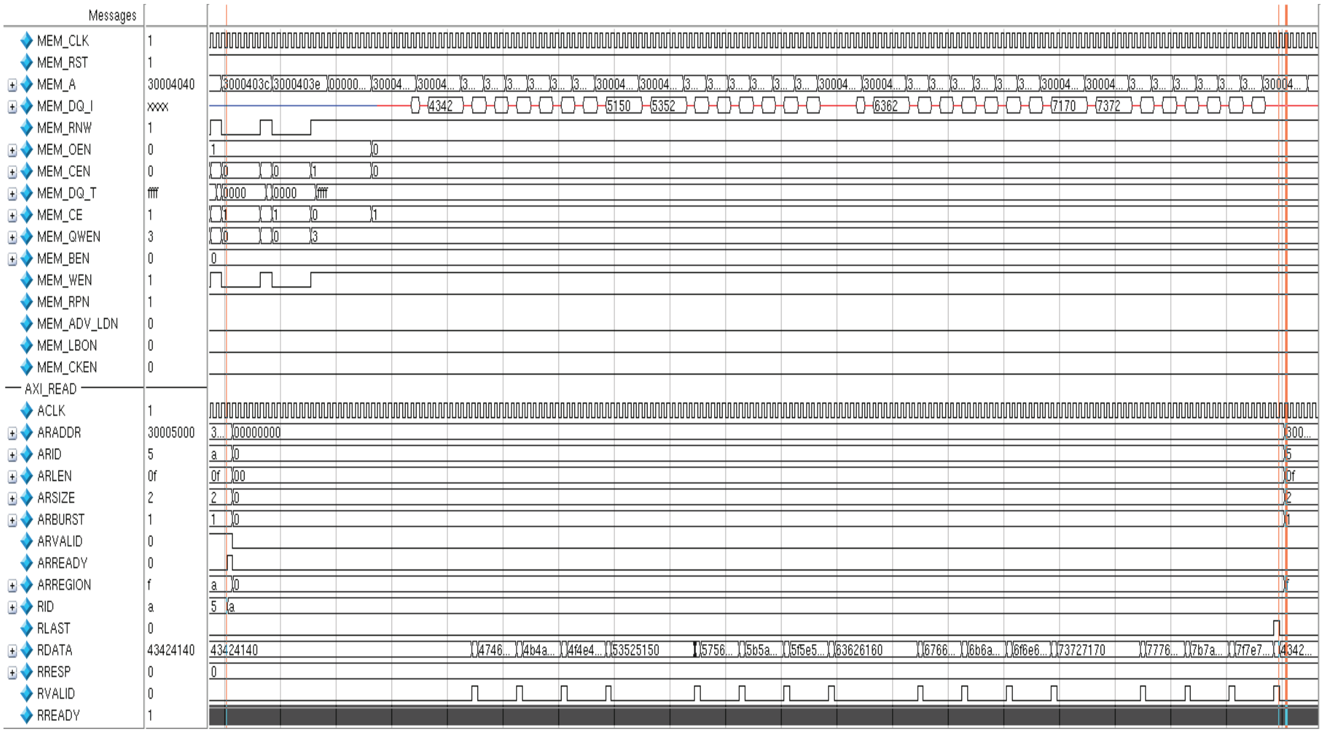
Figure 3-15 shows the timing waveform for 16 bit read for Cellular RAM memory in asynchronous mode.



DS762_20

Figure 3-15: **16 Bit Read Cellular RAM**

Figure 3-16 shows the timing waveform for 16 bit write for Cellular RAM memory in Page Mode Flash.



DS762_21

Figure 3-16: **16 Bit Page Write Cellular RAM**

SECTION II: VIVADO DESIGN SUITE

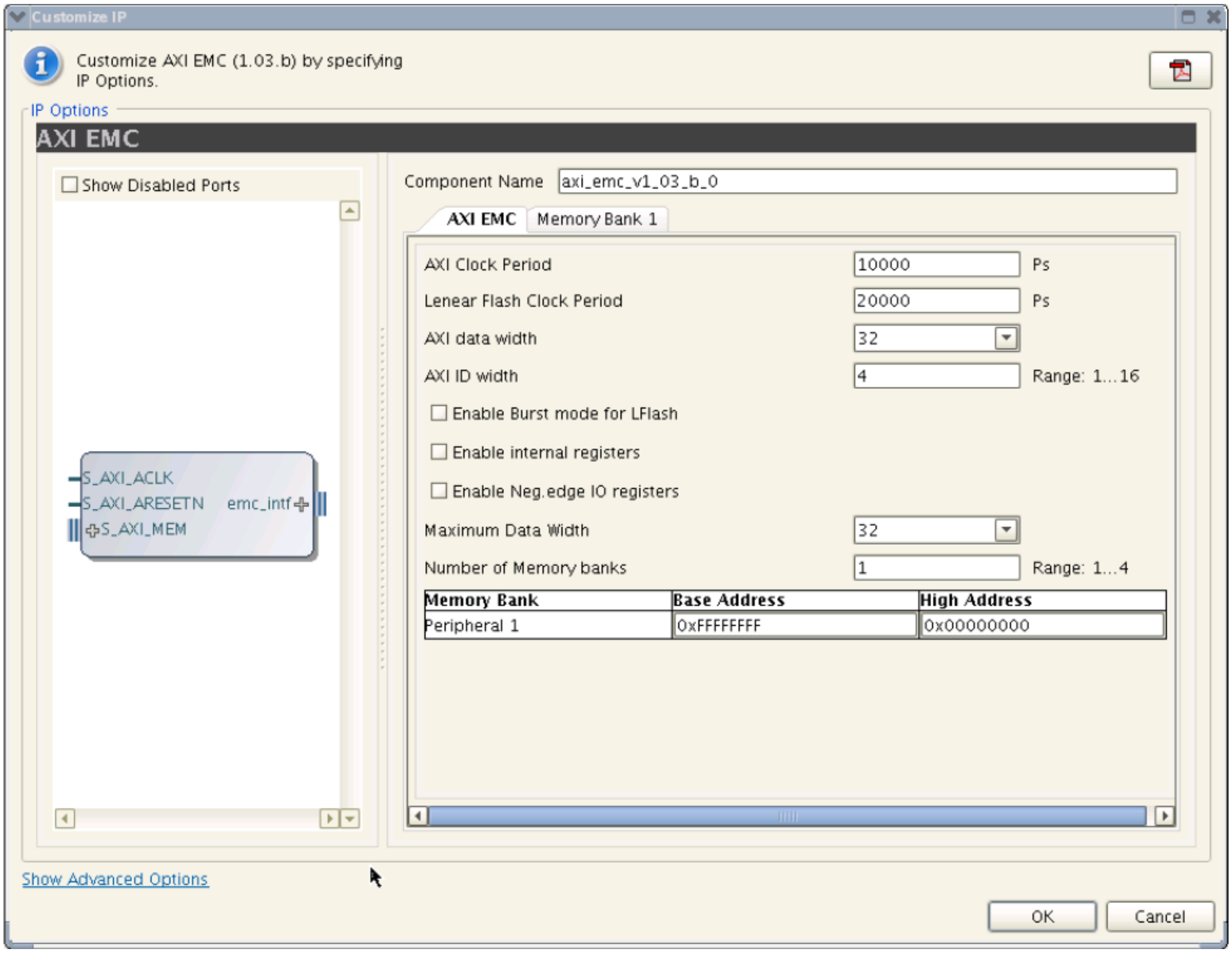
Chapter 4

Customizing and Generating the Core

This chapter includes information about using Xilinx tools to customize and generate the core in the Vivado™ Design Suite environment.

GUI

Figure 4-1 shows the Vivado GUI.



Customize AXI EMC (1.03.b) by specifying IP Options.

□ Show Disabled Ports

Component Name axi_emc_v1_03_b_0

AXI Clock Period	10000	Ps
Lenear Flash Clock Period	20000	Ps
AXI data width	32	
AXI ID width	4	Range: 1...16

□ Enable Burst mode for LFlash

□ Enable internal registers

□ Enable Neg.edge IO registers

Maximum Data Width	32	
Number of Memory banks	1	Range: 1...4

Memory Bank	Base Address	High Address
Peripheral 1	0xFFFFFFFF	0x00000000

Figure 4-1: **AXI EMC Vivado GUI**

The AXI EMC GUI includes the following options:

- **AXI Clock Period**
- **Linear Flash Clock Period**
- **AXI Data Width**: Choose the correct AXI data width.
- **Enable Burst mode for LFlash**
- **Enable Internal Registers**: Choose this option if any of the memories refer to the internal core registers.
- **Enable Neg Reg Edge IO registers**: Choose this option if any of the memory requires a -ve edge-triggered register.
- **Maximum Data Width**
- **No. of Memory Banks**: User should choose the number of memory banks required in the application. With each increase in this number, a new tab is generated for that particular memory bank configuration. Use each tab to configure the required memory. Asynchronous memory needs the correct timing parameters.
- **Peripheral Memory Address**: Ensure this address is in the address range of the core.

IMPORTANT: *Check each memory bank tab to choose the proper memory, data width and pipeline delay/parity (in case of SRAM). If the memory data width is less than the AXI interface data width of the core, select the data width match option.*

For all asynchronous and flash memories, it is required to have the proper timing parameters provided.

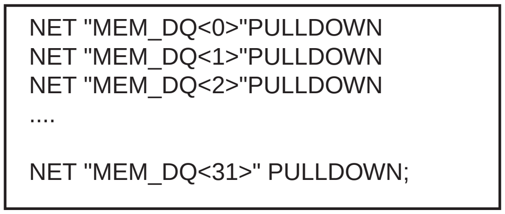
Chapter 5

Constraining the Core

This chapter contains information about constraining the core in the Vivado™ Design Suite environment.

Required Constraints

If external pull-ups/pull-downs are not available on the MEM_DQ signals, then these pins should be specified to use pull-up or pull-down resistors. An example is shown in Figure 5-1.



```
NET "MEM_DQ<0>"PULLDOWN
NET "MEM_DQ<1>"PULLDOWN
NET "MEM_DQ<2>"PULLDOWN
....

NET "MEM_DQ<31>" PULLDOWN;
```

DS762_23

Figure 5-1: **Pin Constraints**

Device, Package, and Speed Grade Selections

See IP Facts for details about device support.

Clock Frequencies

There are no clock frequency constraints for this core.

Clock Management

There are no clock management constraints for this core.

Clock Placement

There are no clock placement constraints for this core.

Banking

There are no banking constraints for this core.

Transceiver Placement

There are no transceiver placement constraints for this core.

I/O Standard and Placement

There are no I/O constraints for this core.

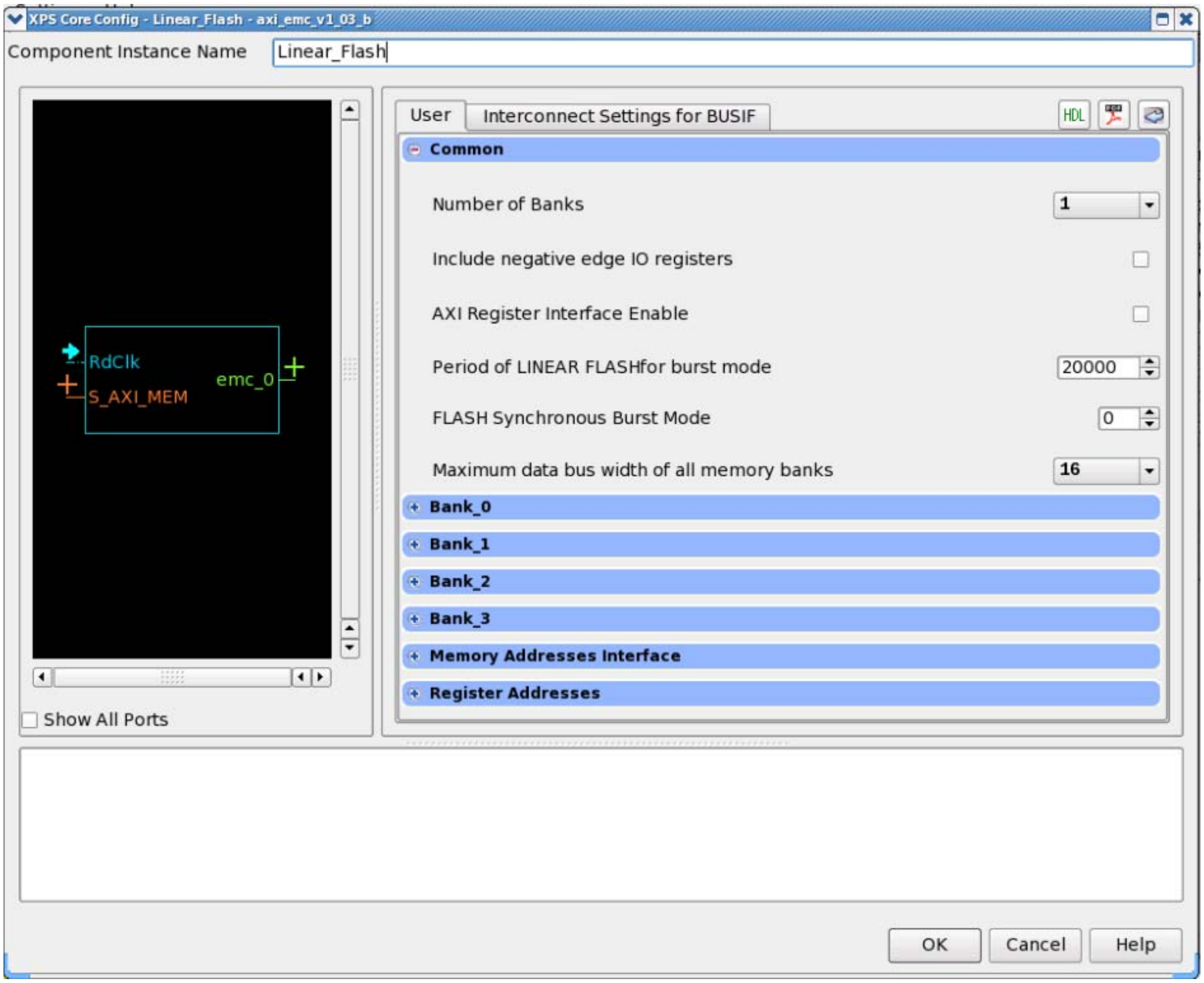
SECTION III: ISE DESIGN SUITE

Customizing and Generating the Core

This chapter includes information about using Xilinx tools to customize and generate the core in the ISE® Design Suite environment.

GUI

Figure 6-1 shows the XPS GUI for the AXI EMC core.



XPS Core Config - Linear_Flash - axi_emc_v1_03_b

Component Instance Name: Linear_Flash

User | Interconnect Settings for BUSIF

Common

- Number of Banks: 1
- Include negative edge IO registers: □
- AXI Register Interface Enable: □
- Period of LINEAR FLASHfor burst mode: 20000
- FLASH Synchronous Burst Mode: 0
- Maximum data bus width of all memory banks: 16

Bank_0

Bank_1

Bank_2

Bank_3

Memory Addresses Interface

Register Addresses

□ Show All Ports

Figure 6-1: **AXI EMC XPS GUI**

The AXI EMC GUI includes the following options:

- **AXI Clock Period**
- **Linear Flash Clock Period**
- **AXI Data Width**: Choose the correct AXI data width.
- **Enable Burst mode for LFlash**
- **Enable Internal Registers**: Choose this option if any of the memories refer to the internal core registers.
- **Enable Neg Reg Edge IO registers**: Choose this option if any of the memory requires a -ve edge-triggered register.
- **Maximum Data Width**
- **No. of Memory Banks**: User should choose the number of memory banks required in the application. With each increase in this number, a new tab is generated for that particular memory bank configuration. Use each tab to configure the required memory. Asynchronous memory needs the correct timing parameters.
- **Peripheral Memory Address**: Ensure this address is in the address range of the core.

IMPORTANT: *Check each memory bank tab to choose the proper memory, data width and pipeline delay/parity (in case of SRAM). If the memory data width is less than the AXI interface data width of the core, select the data width match option.*

For all asynchronous and flash memories, it is required to have the proper timing parameters provided.

Parameter Values in the XCO File

To create a core design that is uniquely tailored for a particular system, certain features are parameterizable in the design. This allows a design that uses only the resources required by the system and runs at the best possible performance. The features that are parameterizable in the AXI EMC core are shown in Table 2-9.

Parameter - Port Dependencies

The dependencies between the AXI EMC core design parameters and I/O signals are described in Table 6-1. In addition, when certain features are parameterized out of the design, the related logic is no longer a part of the design. The unused input signals and related output signals are set to a specified value.

Table 6-1: **Parameter-Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G2	C_S_AXI_EN_REG	P4-P20	-	Valid only when C_S_AXI_EN_REG=1
G4	C_S_AXI_REG_ADDR_WIDTH	P4, P14	-	Defines the address width of the ports
G5	C_S_AXI_REG_DATA_WIDTH	P7, P8, P17	-	Defines the data width of the ports
G6	C_S_AXI_MEM_ADDR_WIDTH	P22, P41, P62	-	Defines the address width of the ports
G7	C_S_AXI_MEM_DATA_WIDTH	P31, P32, P51	-	Defines the data width of the ports
G8	C_S_AXI_ID_WIDTH	P21, P40, P36, P50		Defines the ID width
G12	C_LINEAR_FLASH_SYNC_BURST	P75		Used when Linear Flash Burst mode is required
G20	C_TPACC_PS_FLASH_x	-	G11	Used when C_MEMx_Type not set to 011
G25	C_MAX_MEM_WIDTH	P56, P57, P58, P59, P60, P61, P67, P68		Defines the maximum memory width in the given configuration setting
G33	C_LFLASH_PERIOD_PS	P75		Used when Linear Flash Burst mode is required
I/O Signals				
P4-P20	AXI4-LITE Native Interface Ports	-	G2	Valid only when C_S_AXI_EN_REG=1
P4	S_AXI_REG_AWADDR[C_S_REG_AXI_ADDR_WIDTH-1:0]	-	G4	Port width depends on the generic C_S_AXI_REG_ADDR_WIDTH
P7	S_AXI_REG_WDATA[C_S_REG_AXI_DATA_WIDTH - 1: 0]	-	G5	-Port width depends on the generic C_S_AXI_REG_DATA_WIDTH
P8	S_AXI_REG_WSTB[C_S_AXI_REG_DATA_WIDTH/8-1:0]	-	G5	Port width depends on the generic C_S_AXI_REG_DATA_WIDTH
P14	S_AXI_REG_ARADDR[C_S_REG_AXI_ADDR_WIDTH -1:0]	-	G4	Port width depends on the generic C_S_AXI_REG_ADDR_WIDTH
P17	S_AXI_REG_RDATA[C_S_REG_AXI_DATA_WIDTH -1:0]	-	G5	Port width depends on the generic C_S_AXI_REG_DATA_WIDTH
P21	S_AXI_MEM_AWID[C_S_MEM_AXI_ID_WIDTH-1:0]	-	G8	Port width depends on the generic C_S_AXI_MEM_ID_WIDTH
P22	S_AXI_MEM_AWADDR[C_S_MEM_AXI_ADDR_WIDTH-1:0]	-	G6	Port width depends on the generic C_S_AXI_MEM_ADDR_WIDTH
P31	S_AXI_MEM_WDATA[C_S_MEM_AXI_DATA_WIDTH-1:0]	-	G7	Port width depends on the generic C_S_AXI_MEM_DATA_WIDTH
P32	S_AXI_MEM_WSTB[(C_S_MEM_AXI_DATA_WIDTH/8)-1:0]	-	G7	Port width depends on the generic C_S_AXI_MEM_DATA_WIDTH

Table 6-1: **Parameter-Port Dependencies *(Cont'd)***

Generic or Port	Name	Affects	Depends	Relationship Description
P36	S_AXI_MEM_BID[C_S_MEM_AXI_ID_WIDTH-1:0]	-	G6	Port width depends on the generic C_S_AXI_MEM_ID_WIDTH
P40	S_AXI_MEM_ARID[C_S_MEM_AXI_ID_WIDTH-1:0]	-	G8	Port width depends on the generic C_S_AXI_MEM_ID_WIDTH
P41	S_AXI_MEM_ARADDR[C_S_AXI_MEM_ADDR_WIDTH -1:0]	-	G6	Port width depends on the generic C_S_AXI_MEM_ADDR_WIDTH
P50	S_AXI_MEM_RID[C_S_MEM_AXI_ID_WIDTH-1:0]	-	G8	Port width depends on the generic C_S_AXI_MEM_ID_WIDTH
P51	S_AXI_MEM_RDATA[C_S_MEM_AXI_DATA_WIDTH -1:0]	-	G7	Port width depends on the generic C_S_AXI_MEM_DATA_WIDTH
P56	MEM_DQ_I[C_MAX_MEM_WIDTH - 1:0]	-	G25	Port width depends on the generic C_MAX_MEM_WIDTH
P57	MEM_DQ_O[C_MAX_MEM_WIDTH - 1:0]	-	G25	Port width depends on the generic C_MAX_MEM_WIDTH
P58	MEM_DQ_T[C_MAX_MEM_WIDTH - 1:0]	-	G25	Port width depends on the generic C_MAX_MEM_WIDTH
P59	MEM_DQ_PARITY_I[C_MAX_MEM_WIDTH/8 - 1:0]	-	G25	Port width depends on the generic C_MAX_MEM_WIDTH
P60	MEM_DQ_PARITY_O[C_MAX_MEM_WIDTH/8 - 1:0]	-	G25	Port width depends on the generic C_MAX_MEM_WIDTH
P61	MEM_DQ_PARITY_T[C_MAX_MEM_WIDTH/8 - 1:0]	-	G25	Port width depends on the generic C_MAX_MEM_WIDTH
P62	MEM_A[C_S_AXI_MEM_ADDR_WIDTH - 1:0]	-	G6	Port width depends on the generic C_S_AXI_MEM_ADDR_WIDTH
P64	MEM_CEN[C_NUM_BANKS_MEM - 1:0]	-	G9	Port width depends on the generic C_NUM_BANKS_MEM
P65	MEM_OEN[C_NUM_BANKS_MEM - 1:0]	-	G9	Port width depends on the generic C_NUM_BANKS_MEM
P67	MEM_QWEN[(C_MAX_MEM_WIDTH/8) - 1:0]	-	G25	Port width depends on the generic C_MAX_MEM_WIDTH
P68	MEM_BEN[(C_MAX_MEM_WIDTH/8) - 1:0]	-	G25	Port width depends on the generic C_MAX_MEM_WIDTH
P69	MEM_CE[C_NUM_BANKS_MEM - 1:0]	-	G9	Port width depends on the generic C_NUM_BANKS_MEM

Chapter 7

Constraining the Core

This chapter contains information about constraining the core in the ISE® Design Suite environment.

Required Constraints

There are no required constraints for this core.

Device, Package, and Speed Grade Selections

See IP Facts for details about device support.

Clock Frequencies

There are no clock frequency constraints for this core.

Clock Management

There are no clock management constraints for this core.

Clock Placement

There are no clock placement constraints for this core.

Banking

There are no banking constraints for this core.

Transceiver Placement

There are no transceiver placement constraints for this core.

I/O Standard and Placement

There are no I/O constraints for this core.

SECTION IV: APPENDICES

Appendix 9

Debugging

This appendix includes details about resources available on the Xilinx Support website and debugging tools. The following topics are included in this appendix:

- Finding Help on Xilinx.com
- Interface Debug

Finding Help on Xilinx.com

To help in the design and debug process when using the AXI EMC, the Xilinx Support web page (www.xilinx.com/support) contains key resources such as product documentation, release notes, answer records, information about known issues, and links for opening a Technical Support WebCase.

Documentation

This product guide is the main document associated with the AXI EMC. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page (www.xilinx.com/support) or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Design Tools tab on the Downloads page (www.xilinx.com/download). For more information about this tool and the features available, open the online help after installation.

Release Notes

Known issues for all cores, including the AXI EMC are described in the IP Release Notes Guide (XTP025).

Known Issues

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product.

Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core are listed below, and can also be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use proper keywords such as

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Contacting Technical Support

Xilinx provides premier technical support for customers encountering issues that require additional assistance.

To contact Xilinx Technical Support:

1. Navigate to www.xilinx.com/support.
2. Open a WebCase by selecting the WebCase link located under Support Quick Links.

When opening a WebCase, include:

- Target FPGA including package and speed grade.
- All applicable Xilinx Design Tools and simulator software versions.
- Additional files based on the specific issue might also be required. See the relevant sections in this debug guide for guidelines about which file(s) to include with the WebCase.

Interface Debug

AXI4-Lite Interfaces

Read from a register that does not have all 0s as a default to verify that the interface is functional. Output S_AXI_REG_ARREADY asserts when the read address is valid, and output S_AXI_REG_RVALID asserts when the read data/response is valid. If the interface is unresponsive, ensure that the following conditions are met:

- The S_AXI_ACLK and ACLK inputs are connected and toggling.
- The interface is not being held in reset, and S_AXI_ARESET is an active-Low reset.

- The interface is enabled, and `s_axi_aclken` is active-High (if used).
- The main core clocks are toggling and that the enables are also asserted.
- If the simulation has been run, verify in simulation and/or a ChipScope debugging tool capture that the waveform is correct for accessing the AXI4-Lite interface.

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

www.xilinx.com/support.

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm.

References

These documents provide supplemental material useful with this product guide:

1. *ARM® AMBA® AXI Protocol v2.0 Specification* (ARM IHI 0022C)
2. *AXI Slave Burst Data Sheet* (DS769)
3. *AXI Lite IPIF Data Sheet* (DS765)
4. *AXI Interconnect IP Data Sheet* (DS768)
5. *7 Series FPGAs Overview* (DS180)
6. *Virtex-6 Family Overview* (DS150)
7. *Spartan-6 Family Overview* (DS160)
8. Vivado™ Design Suite user documentation

Technical Support

Xilinx provides technical support at www.xilinx.com/support for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

See the Embedded Edition Derivative Device Support web page (www.xilinx.com/ise/embedded/ddsupport.htm) for a complete list of supported derivative devices for this core.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/18/12	1.0	Initial Xilinx release as a product guide. Replaces *LogiCORE IP AXI EMC Data Sheet*, DS762.

Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at http://www.xilinx.com/warranty.htm; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: http://www.xilinx.com/warranty.htm#critapps.