LogiCORE IP AXI EMC (v1.01a)

DS762 June 22, 2011 Product Specification

Introduction

The AXI EMC (Advanced Microcontroller Bus Architecture (AMBA®) Advanced extensible Interface (AXI) External memory controller) provides the 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 interface with the AXI 4 Interface.

Features

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

- Supports AXI 4 specification for AXI interface
- Full AXI Slave interface supports 32- Bit Address bus and 32/64-bit data bus
- Supports 32-Bit configurable AXI4 Lite control interface to access internal registers
- Supports Burst transfers of 1-256 beats for INCR burst type and 2, 4, 8, 16 beats for WRAP burst type
- Supports AXI narrow transfers, unaligned transfer type of transactions
- Supports multiple (up to 4) external memory banks
- Supports independent memory configuration of each memory bank
- Supports memory data widths of 64-bit, 32-bit, 16-bit and 8-bit for each of the memory banks
- Supports Synchronous / Asynchronous SRAMs, Linear and Page Mode NOR Flash, and PSRAM/Cellular RAM memory devices
- Supports configurable Byte parity check for each of the memory banks for Synchronous / Asynchronous SRAMs
- Supports memory configuration, Timing parameters, Data with for each memory bank independently
- Supports memory configuration, Timing parameters, Data with for each memory bank independently
- Supports configurable registers for PSRAM mode of operations.

LogiCORE IP Facts Table				
Core Specifics				
Supported Device Family[1]	Artix-7, Virtex-7, Kintex-7, Virtex-6, Spartan-6			
Supported User Interfaces	AXI4-Lite, AXI4			
Resources				
	Resources			Frequency
	LUTs	FFs	Block RAMs	Max Freq
	See Table 22 through Table 26		N/A	See Table 22 through Table 26
Provided with Core				
Documentation	Product Specification			
Design Files	VHDL			
Example Design	Not Provided			
Test Bench	Not Provided			
Constraints File	UCF (user constraints file)			
Simulation Model	NA			
Tested Design Tools				
Design Entry Tools	XPS 13.2			
Simulation	Mentor Graphic ModelSim [2]			
Synthesis Tools	ISE 13.2			
Support				
Provided by Xilinx, Inc.				

1. For a complete list of supported derivative devices, please see the IDS Embedded Edition Derivative Device Support.
2. For the supported versions of the tools, see the ISE Design Suite 13: Release Notes Guide.

Functional Description

The AXI EMC device is comprised of several modules. They are:

- AXI4 Native Interface module
- Select Parameters module
- Mem State Machine module
- Mem Steer module
- Address Counter Mux module
- Counters module
- I/O Registers module
- IPIC Interface module

The architectural block diagram of the AXI EMC core is shown in Figure 1. The device modules are described in the subsequent sections.

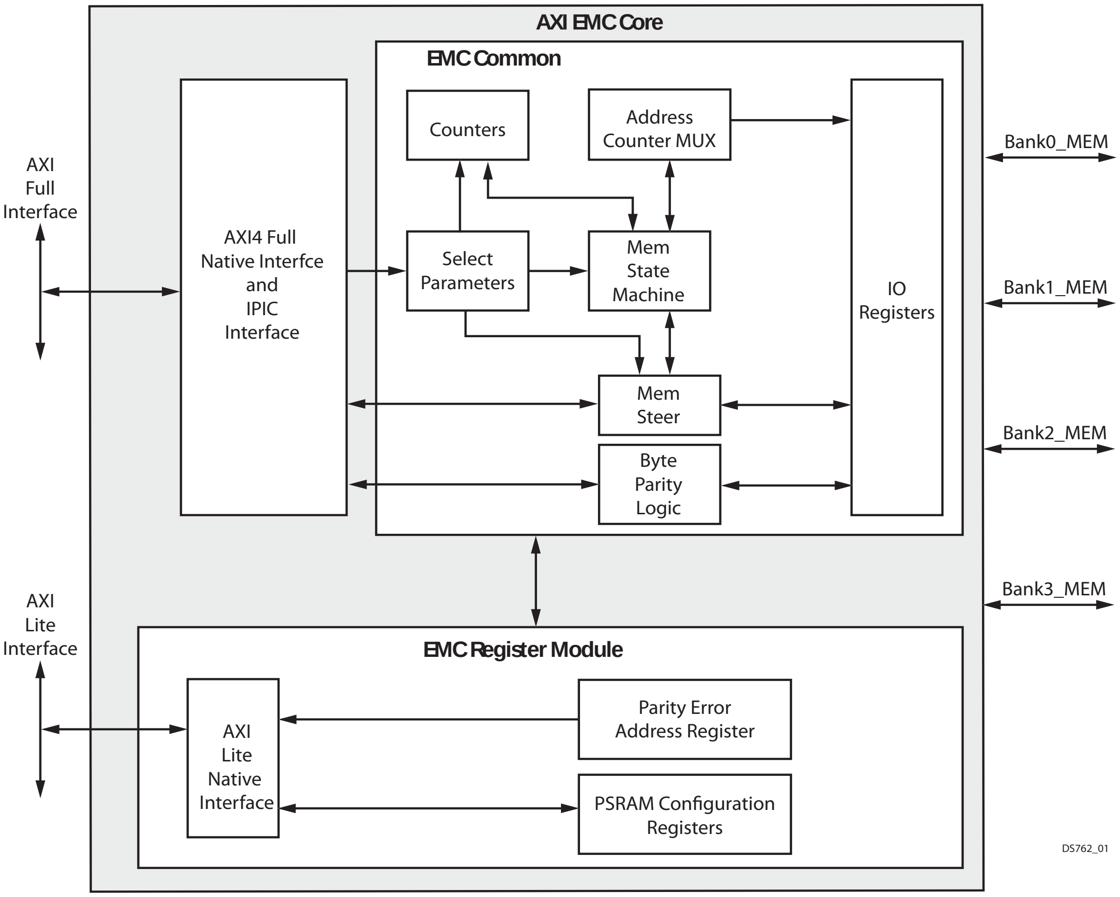


Figure 1: **Top Level Block Diagram of the AXI EMC Core**

Native AXI4 Full Interface Module

The native AXI4 full interface module provides the interface to the Full AXI memory mapped interface and implements AXI4 protocol logic. AXI4 Native Interface Module is a bi-directional interface between a user IP core and the Full AXI interface standard. To simplify the process of attaching an AXI EMC core to the AXI4, the core make use of a portable bus interface native logic which includes address decoding and address generation and control signal generation, which manages the bus interface signals, interface protocols, and other interfaces. This interface will be used to access external memories

Native AXI Lite Interface Module

The native AXI4 lite interface module provides the interface to the AXI4 lite interface and implements AXI protocol logic. This interface will be used to access internal registers. This interface is enabled only when C_S_AXI_EN_REG is 1. This is not a separate module as such, but it 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 AXI Interface Module. This module contains data width matching logic.

Address Counter Mux

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 which is directed to the Mem State Machine.

Mem State Machine

The state diagram of Mem State Machine is shown in Figure 2

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

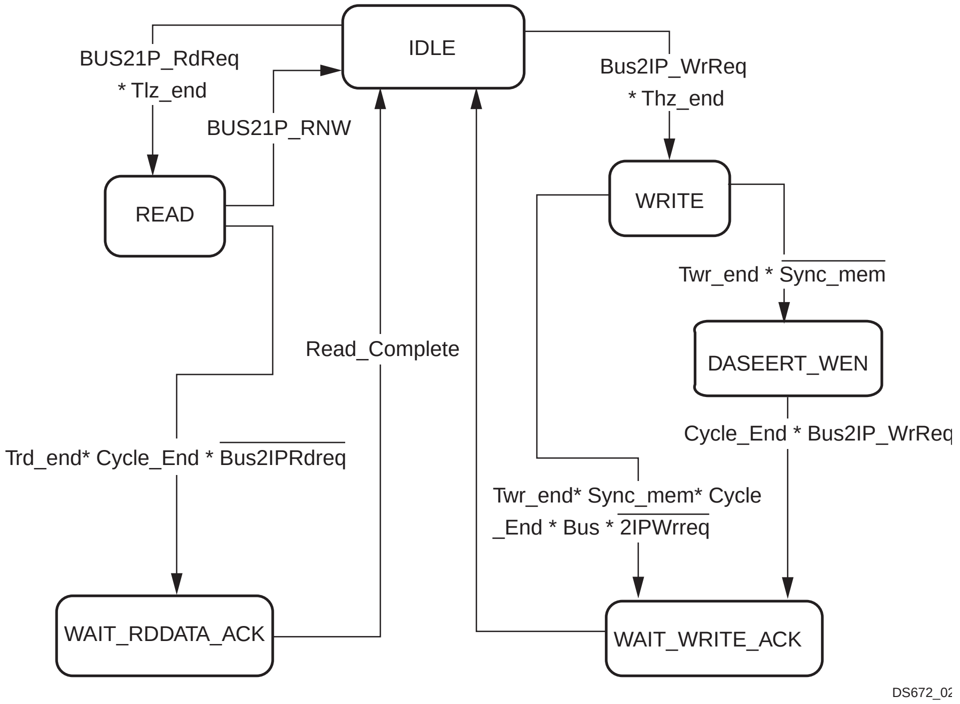


Figure 2: **Mem State Machine**

Byte parity Logic

The Byte parity logic generates and calculates the parity logic, if C_USE_PARITY_x parameter is enabled, for any memory bank. The number of such blocks instantiated depends on C_NUM_BANKS_MEM. There will be a parity bit 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 the memory. For a memory read, the parity bit will be calculated on a read byte, then compared with the parity bit from memory. When the parity error occurs, the AXI logic responds with an AXI OKAY/SLVERROR response. The error address will be 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 configuration register (PSRAM_CONFG_REG_x). The PERR_ADDR_REG_x registers contain the address for the parity error that occurred. The number of PSRAM_CONFG_REG_x and PSRAM_CONFG_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

This is a separate module which is at 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 output 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 3, and can be used at lower clock frequency to provide adequate setup and hold times to synchronous memories.

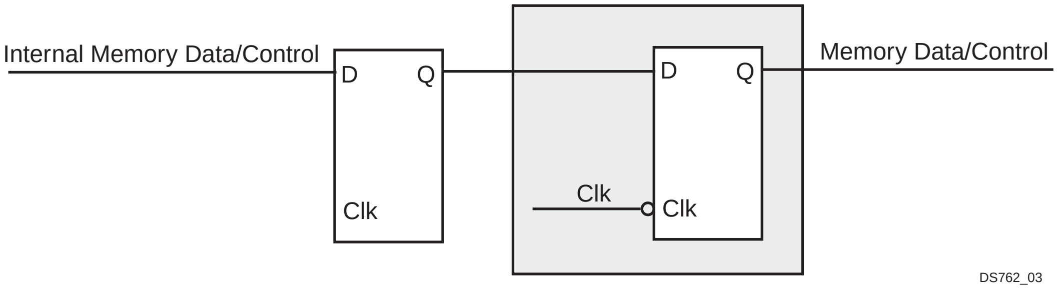


Figure 3: **Output Registers When C_INCLUDE_NEGEDGE_IOREGS = 1**

IPIC Interface

The IPIC Interface module connects the EMC core to the AXI Interface Module.

I/O Signals

The I/O signals are listed and described in Table 1.

Table 1: **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 gives 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
P10	S_AXI_REG_WREADY	AXI	O	1	Write ready. This signal indicates that the slave can accept the write data

Table 1: **I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
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 will remain 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.
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.

Table 1: **I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
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 (upto 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					
P31	S_AXI_MEM_WDATA [C_S_AXI_MEM_DATA_WIDTH-1:0]	AXI	I	-	Write data bus.

Table 1: **I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
P32	S_AXI_MEM_WSTB [(C_S_AXI_MEM_DATA_WIDTH/8)-1:0]	AXI	I	-	Write strobes: This signal indicates which byte lanes in S_AXI_WDATA are/is 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.
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

Table 1: **I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
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 will remain 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.
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

Table 1: **I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
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
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

Notes:

1. Same clock and reset signals should be used for both Register and Memory interface.
2. This clock is used to capture the data from memory. Connection to this port is must otherwise design will fail. Generally this should be connected to system/bus clock. User can connected this to other clock nets e.g. phase shift clock or feedback clock.
3. The AXI4 Lite interface will be available only when the C_S_AXI_EN_REG = 1.
4. Most asynchronous memory devices will only use MEM_CEN. Most synchronous memory devices will use both MEM_CEN and MEM_CE. Refer to the device data sheet for correct connection of these signals.

Design Parameters

To allow the user to create a core design that is uniquely tailored for the user's system, certain features are parameterizable in the design. This allows the user to have a design that utilizes 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 as shown in Table 2.

Inferred Parameters

In addition to the parameters listed in Table 2, 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 DS768, *AXI Interconnect IP Data Sheet*.

Table 2: **Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameter					
G1	Target FPGA family	C_FAMILY	spartan-6, virtex-6	virtex6	string
AXI4 Lite Interface Parameters(1)					
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)	000 = Sync SRAM 001 = Async SRAM 010 = Linear Flash 011 = Page Mode Flash 100 = PSRAM	000	std_logic_vector
G12	Execute multiple memory access cycles to match memory bank x data width to AXIdata width	C_INCLUDE_DATAWIDTH_MATCHING_x(2)(3)	0 = Don't include data width matching 1 = Include data width matching	0	integer
G13	Parity Check	C_PARITY_TYPE_x(2)(4)	00 = No parity 01 = Odd Parity 10 = Even Parity	00	std_logic_vector

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

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G14	Pipeline delay (in clock cycles) of memory bank x	C_SYNCH_PIPEDELAY_x(2)(5)	1 = Flow-Through model 2 = Pipeline Model	2	integer
G15	Input/output data and control signals using the falling edge of the clock	C_INCLUDE_NEGEDGE_ IOREGS(7)(8)	0 = Don't include negative edge IO registers (data and control signals are input/output on the rising edge of the clock) 1 = Include negative edge IO registers (data and control signals are input/output on the falling edge of the clock)	0	integer
Memory Bank Timing Parameters					
G16	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
G17	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
G18	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
G19	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
G20	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
G21	Write cycle time of memory bank x	C_TWC_PS_MEM_x(2)(18)(19)	Integer number of picoseconds	15000(12)	integer
G22	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
G23	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	0(12)	integer

Table 2: **Design Parameters *(Cont’d)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G24	Write cycle phase time period duration of memory bank x	C_WPH_PS_MEM_x	Integer number of picoseconds	12000[12]	integer
Auto Calculated Parameter[23]					
G25	Maximum data width of the memory devices in all banks	C_MAX_MEM_WIDTH[2]	8, 16, 32, 64	32	integer
Address Space Parameters					
G26	Base address of memory bank x	C_S_AXI_MEMx_BASEADDR[2]	Valid address range[24]	None[24]	std_logic_vector
G27	High address of memory bank x	C_S_AXI_MEMx_HIGHADDR[2]	Valid address range[24]	None[24]	std_logic_vector
G28	Base address of Register space x	C_S_AXI_REG_BASEADDR[2]	Valid address range[24]	None[24]	std_logic_vector
G29	High address of Register bank x	C_S_AXI_REG_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

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

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type

Notes:

1. Is 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 = 000 or 001
5. Is Valid only when C_MEMx_Type = 000
6. Is Valid only when C_MEMx_Type not equal to 000
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 = 000
9. C_SYNCH_PIPEDELAY_x is used when C_MEMx_Type = 000
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 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. Refer to 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 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 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 in the respective memory device data sheets.
17. Page access time, C_TPACC_PS_FLASH_x is equivalent to t_{PACC} of 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 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 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 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 the user must set this parameter to the maximum value of the C_MEMx_WIDTH generics.
24. No default value will be specified for C_MEMx_BASEADDR and C_MEMx_HIGHADDR to insure that the actual value is set, i.e. if the value is not set, a compiler error will be generated. The range specified by C_MEMx_BASEADDR and C_MEMx_HIGHADDR must be a power of 2

Allowable Parameter Combinations

The AXI4 Lite Native interface is included in the design only when C_S_AXI_EN_REG=1. The registers will be valid only for PSRAM and SRAM memories (C_MEMx_Type = 000,001,100). C_S_AXI_EN_REG should be used for the above mentioned memories only.

If C_MEMx_Type = 000, 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.

Parameter - Port Dependencies

The dependencies between the AXI EMC core design parameters and I/O signals are described in Table 3. In addition, when certain features are parameterized out of the design, the related logic will no longer be a part of the design. The unused input signals and related output signals are set to a specified value.

Table 3: **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 Address width of the ports
G5	C_S_AXI_REG_DATA_WIDTH	P7, P8, P17	-	Defines data width of the ports
G6	C_S_AXI_MEM_ADDR_WIDTH	P22, P41, P62	-	Defines Address width of the ports
G7	C_S_AXI_MEM_DATA_WIDTH	P31, P32, P51	-	Defines data width of the ports
G8	C_S_AXI_ID_WIDTH	P21, P40, P36, P50		Defines the ID width
G20	C_TPACC_PS_FLASH_x	-	G11	used when C_MEMx_Type not set to 011
G24	C_MAX_MEM_WIDTH	P56, P57, P58, P59, P60, P61, P67, P68		Defines the maximum memory width in the given configuration setting
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

Table 3: **Parameter-Port Dependencies *(Cont'd)***

Generic or Port	Name	Affects	Depends	Relationship Description
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
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]	-	G24	Port width depends on the generic C_MAX_MEM_WIDTH
P57	MEM_DQ_O[C_MAX_MEM_WIDTH - 1:0]	-	G24	Port width depends on the generic C_MAX_MEM_WIDTH
P58	MEM_DQ_T[C_MAX_MEM_WIDTH - 1:0]	-	G24	Port width depends on the generic C_MAX_MEM_WIDTH
P59	MEM_DQ_PARITY_I[C_MAX_MEM_WIDTH/8 - 1:0]	-	G24	Port width depends on the generic C_MAX_MEM_WIDTH
P60	MEM_DQ_PARITY_O[C_MAX_MEM_WIDTH/8 - 1:0]	-	G24	Port width depends on the generic C_MAX_MEM_WIDTH
P61	MEM_DQ_PARITY_T[C_MAX_MEM_WIDTH/8 - 1:0]	-	G24	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]	-	G24	Port width depends on the generic C_MAX_MEM_WIDTH
P68	MEM_BEN[(C_MAX_MEM_WIDTH/8) - 1:0]	-	G24	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

How the Timing Parameters Are Calculated Internally?

The timing parameters for the core will be in effect in the design only when the targeted memory type is either of Async SRAM or Linear Flash or Page Mode Flash or PSRAM type. While doing the internal signal assertion of de-assertion, come parameters are combined and the referred for final calculation. The internal timing relation is calculated as below -

- 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

Read Cycle End to Data Bus High Impedance is calculated as (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 is calculated as
((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)
is calculated as (C_TPACC_PS_FLASH_x/C_BUS_CLOCK_PERIOD_PS).

Write Cycle to Data Store is calculated as
((max2(1,max2(C_TWC_PS_MEM_x, C_TWP_PS_MEM_x))-1)/C_BUS_CLOCK_PERIOD_PS).

Write cycle de-active period is calculated as (C_TWPH_PS_MEM_x/C_BUS_CLOCK_PERIOD_PS).

Write Cycle End Data Hold Time is calculated
as ((max2(1,C_TLZWE_PS_MEM_0)-1)/C_BUS_CLOCK_PERIOD_PS).

Register Descriptions

There are four internal registers in the AXI EMC design as shown in Table 4. 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 4.

Table 4: **Registers**

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)

Table 4: **Registers**

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
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_CONFIG_REG_0	Write/ Read	0x24	Bank-0 PSRAM memory Configuration Register[4]
C_S_AXI_REG_BASEADDR + 0x14[1][2]	PSRAM_CONFIG_REG_1	Write/ Read	0x24	Bank-1 PSRAM memory Configuration Register[4]
C_S_AXI_REG_BASEADDR + 0x18[1][2]	PSRAM_CONFIG_REG_2	Write/ Read	0x24	Bank-2PSRAM memory Configuration Register[4]
C_S_AXI_REG_BASEADDR + 0x1C[1][2]	PSRAM_CONFIG_REG_3	Write/ Read	0x24	Bank-3PSRAM memory Configuration Register[4]

Notes:

1. Register block will be included when C_S_AXI_EN_REG = 1
2. Number of memory banks depends on C_NUM_BANKS_MEM value
3. Parity check registers valid when C_MEMx_Type=000/001
4. PSRAM configuration registers valid only when C_MEMx_Type=100

PARITY ERROR ADDRESS Register (PARITY_ERR_ADDR_REG_x)

AXI EMC Parity Error Address registers are read only registers which gives the AXI address on which the parity error occurred. This registers are valid only when C_S_AXI_EN_REG = 1 and C_MEMx_Type=000/001. The number of such registers depends on 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 on which error occurred.

The Parity Error Address Register is shown in Figure 4 and described in Table 5

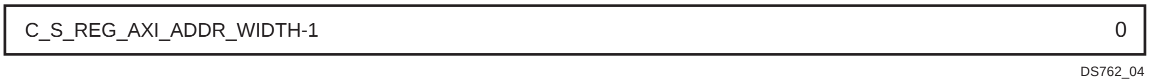


Figure 4: **AXI EMC Parity Error address Register**

Table 5: **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 Configuration Register (PSRAM_CONFIG_REG_X)

The PSRAM configuration registers are write and read registers which are used for configuration of the controller for the PSRAM memories. These registers are valid only when C_S_AXI_EN_REG = 1 and C_MEMx_Type=100. The number of such registers depends on the C_NUM_BANKS_MEM parameter and their corresponding C_MEMx_Type parameters.

The PSRAM Memory for EMC controller works only with Active Low wait polarity. Burst configurations of BL of the PSRAM BCR registers should be configured based on the AXI cacheline configurations.

There can be four PSRAM_CONFIG_REG registers. The PSRAM configuration register is shown in Figure 5 and described in Table 6.

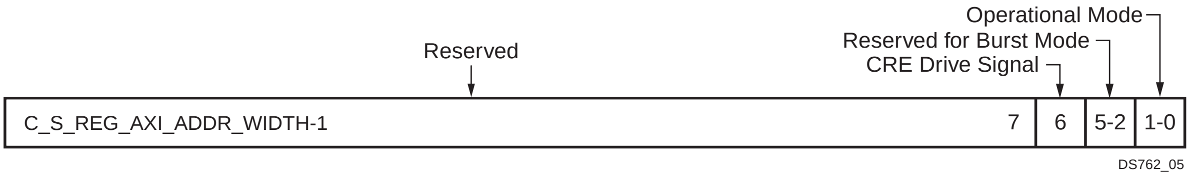


Figure 5: **PSRAM Configuration Register**

Table 6: **PSRAM Configuration Register(PARITY_ERR_ADDR_REG_x)**

Bits	Name	Core Access	Reset Value	Description
31:7	Reserved	Write/Read	0x0	reserved for future usage
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 usage 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= Async Mode 0x01= Page Mode 0x10= Reserved for Future Usage (Burst Mode)

Notes:

1. Various Latency codes can be configured. Refer PSRAM memory data sheet for reference

Address Map Description

As shown in the Table 7, the AXI EMC device supports up to four banks of external memory. The number of available banks actually used is determined by the value of the C_NUM_BANKS_MEM parameter. This parameter may 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 in terms of 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 will be 0 and n least significant bits of the high address will be 1. For example, a memory bank with an addressable range of 16M (2^{24}) bytes could have a base address of 0xFF000000 and a high address of 0xFFFFFFFF. A memory bank with an addressable range of 64K (2^{16}) bytes 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 memory bank is shown in Table 7.

Table 7: **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:

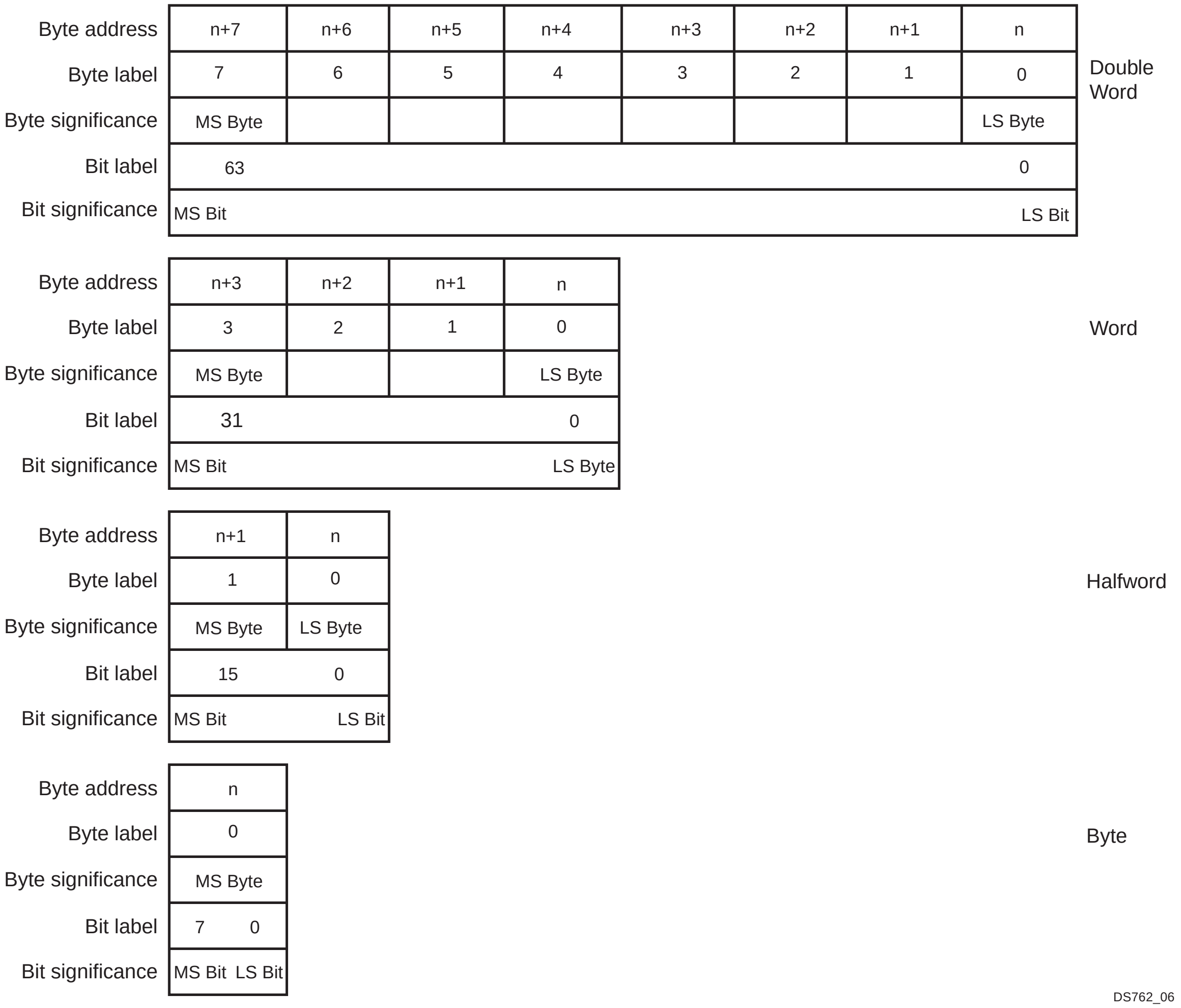
1. Number of memory banks depends on C_NUM_BANKS_MEM value
2. Register block will be included when C_S_AXI_EN_REG = 1

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 and supported. The bit and byte labeling for the big-endian data types is shown in Figure 6.



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 6: **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 the synchronous memory.

To synchronize the synchronous memory clock to the internal FPGA clock, the FPGA system design should include a DCM external to the AXI EMC core that uses the synchronous memory clock input as the feedback clock as shown in Figure 7. This means that the synchronous clock output from the FPGA must be routed back to the FPGA on a clock pin.

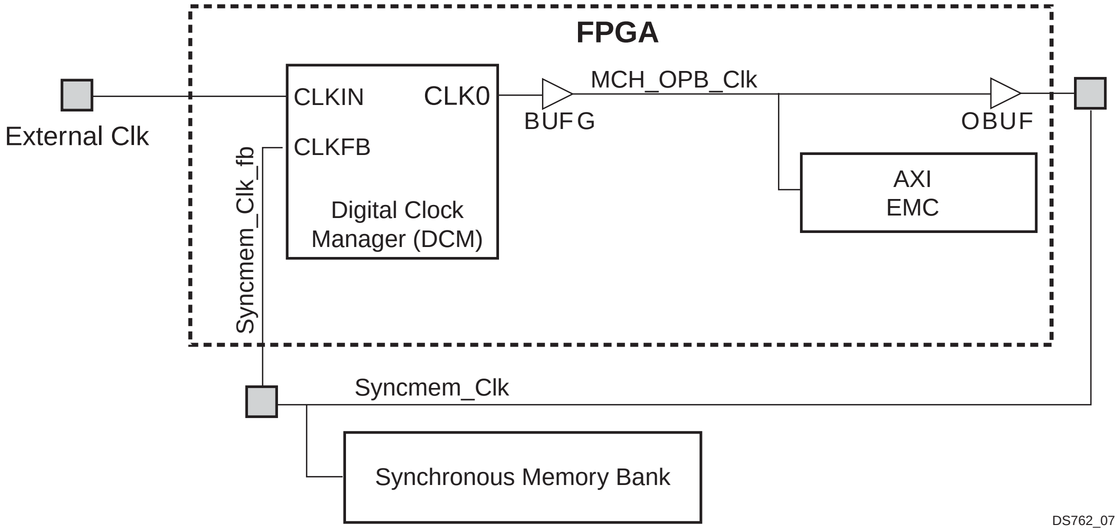


Figure 7: **Synchronous Memory Bank Clocked by FPGA Output With Feedback**

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 8 should be included in the FPGA external to the AXI EMC core.

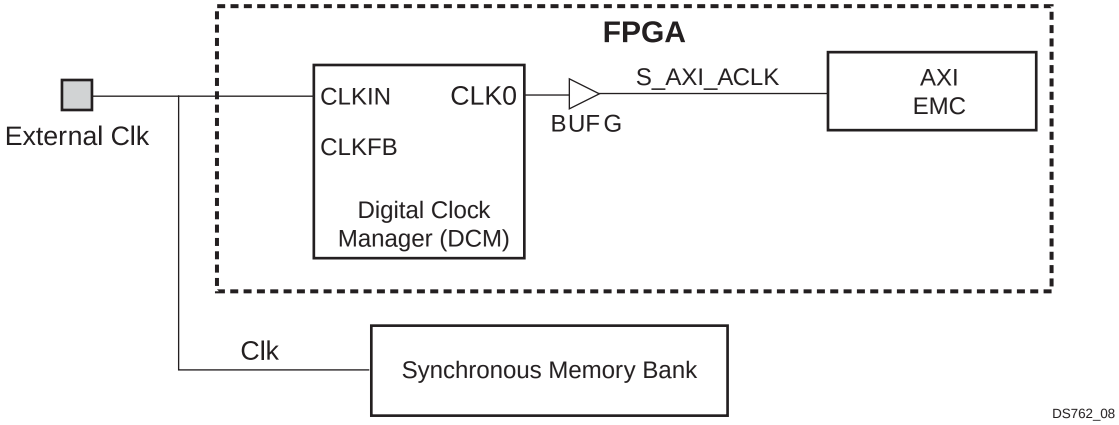


Figure 8: **Synchronous Memory Bank Clocked by External Clock**

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), D(31) is the MSB), and memory devices are little-endian D(31:0) with D(31) as the MSB.

Care must be taken when connecting the chip enable signals. Most asynchronous memory devices will only use MEM_CEN, while most synchronous memory devices will 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.

Exercise caution when making the connections to the external memory devices to avoid incorrect data and address connections. The following tables show the correct mapping of memory controller pins to memory device pins.

Table 8 shows variables used in defining memory subsystem and Table 9 shows interconnection of AXI EMC signals to memory interface signals.

Table 8: **Variables Used In Defining Memory Subsystem**

Variable	Allowed Range	Definition
BN	0 to 3	Memory bank number
DN	0 to 63	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	8 to 64	Width in bits of memory subsystem
DW	1 to 63	Width in bits of data bus for memory device
MAW	1 to 32	Width in bits of address bus for memory device
AU	1 to 63	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)
HAW	1 to 32	Width in bits of AXI address bus

Notes:

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

Connecting to SRAM

Table 9: **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(HAW-AS-1:HAW-MAW-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 10 shows variables for a simple SRAM example.

Table 10: **Variables For Simple SRAM 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; 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

Table 10: **Variables For Simple SRAM Example *(Cont'd)***

Variable	Value	Definition
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 (e.g. AXI)

Table 11 shows connection to 32-bit memory using 2 IDT71V416S (256K X 16-Bit) parts.

Table 11: **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)	$\overline{\text{BHE}}$:$\overline{\text{BLE}}$
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)	$\overline{\text{BHE}}$:$\overline{\text{BLE}}$

Connecting to Byte Parity Memory:

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

Table 12 shows variables for simple SRAM example.

Table 12: **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 (e.g. AXI)

Table 13 shows connection to 32-bit memory using 2 IDT71V416S (256K X 16-Bit) parts.

Table 13: **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 function properly. Table 14 shows an example of how to connect the big-endian AXI EMC bus to the little-endian StrataFlash parts.

The proper connection ordering is also indicated in a more general form in Table 9. StrataFlash parts have a x8 mode and a 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 mode, since A0 always selects a byte.

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

Supports byte read, but not byte write. The smallest data type that can be written is 16-bit data.

Table 14 shows variables for StrataFlash (x16 mode) example.

Table 14: **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 (e.g. AXI)

Table 15 shows connection to 32-bit memory using two StrataFlash parts.

Table 15: **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.

Supports byte reads and writes.

Table 16 shows variables for StrataFlash (x8 mode) example.

Table 16: **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 (e.g. AXI)

Table 17 shows connection to 32-bit memory using four StrataFlash parts.

Table 17: **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}
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. A x8 configuration, DQ(15:8) are not used and should be treated according to manufacturer's data sheet.

Connecting to Spansion PageModeFlash S29GL032N

Table 18 shows variables for Spansion Page Mode Flash(x16 mode) example.

Table 18: **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 (e.g. AXI)

Table 19: **Connection To 16-bit Memory Using PageModeFlash 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 20: **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 (e.g. AXI)

Table 21: **Connection To 16-bit Memory Using PageModeFlash 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 : 0)	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

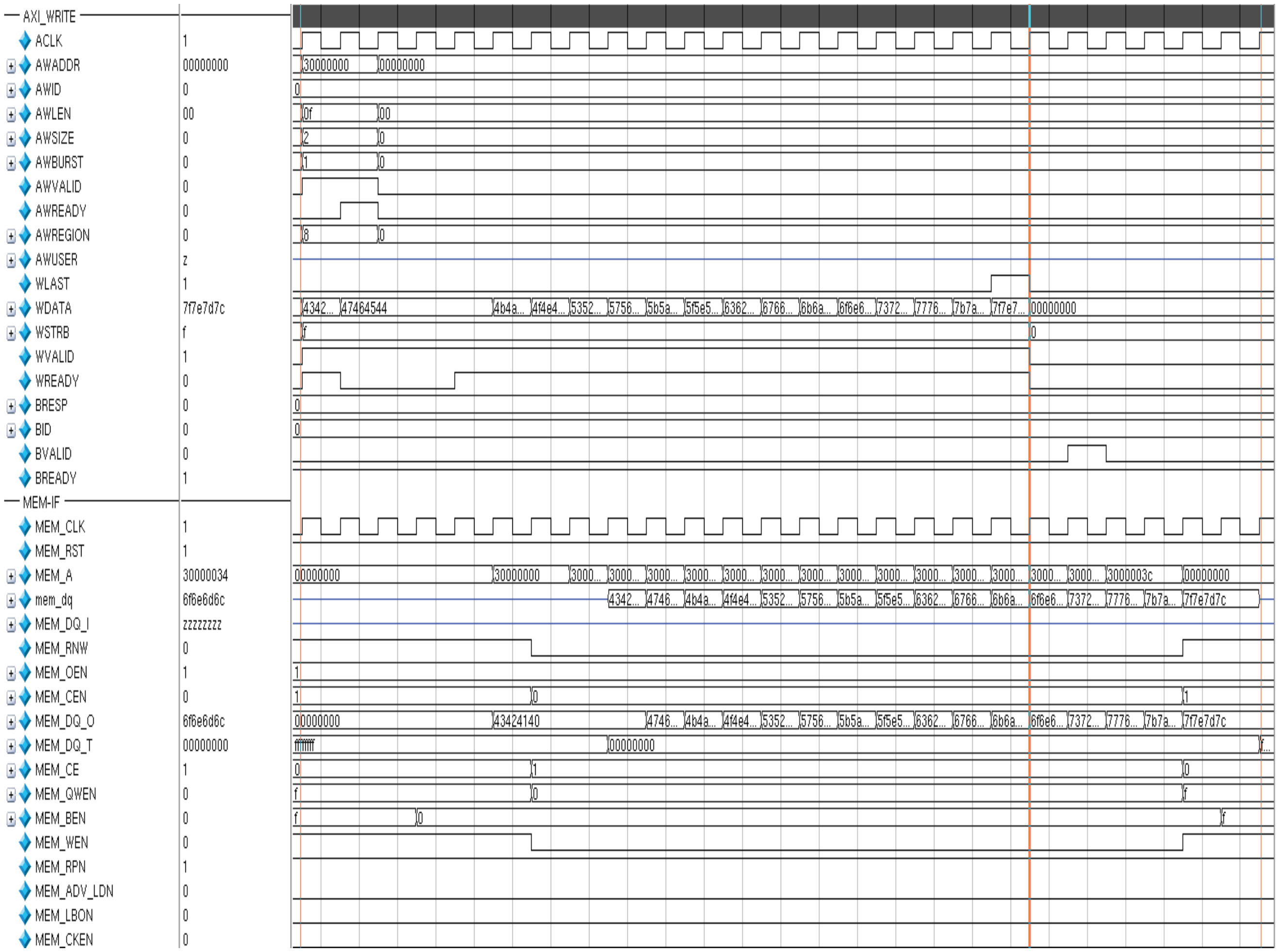
Software Considerations

Absolute memory addresses are frequently required for specifying command sequences to flash devices. Due to the memory address shift (AS), any absolute addresses which much 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

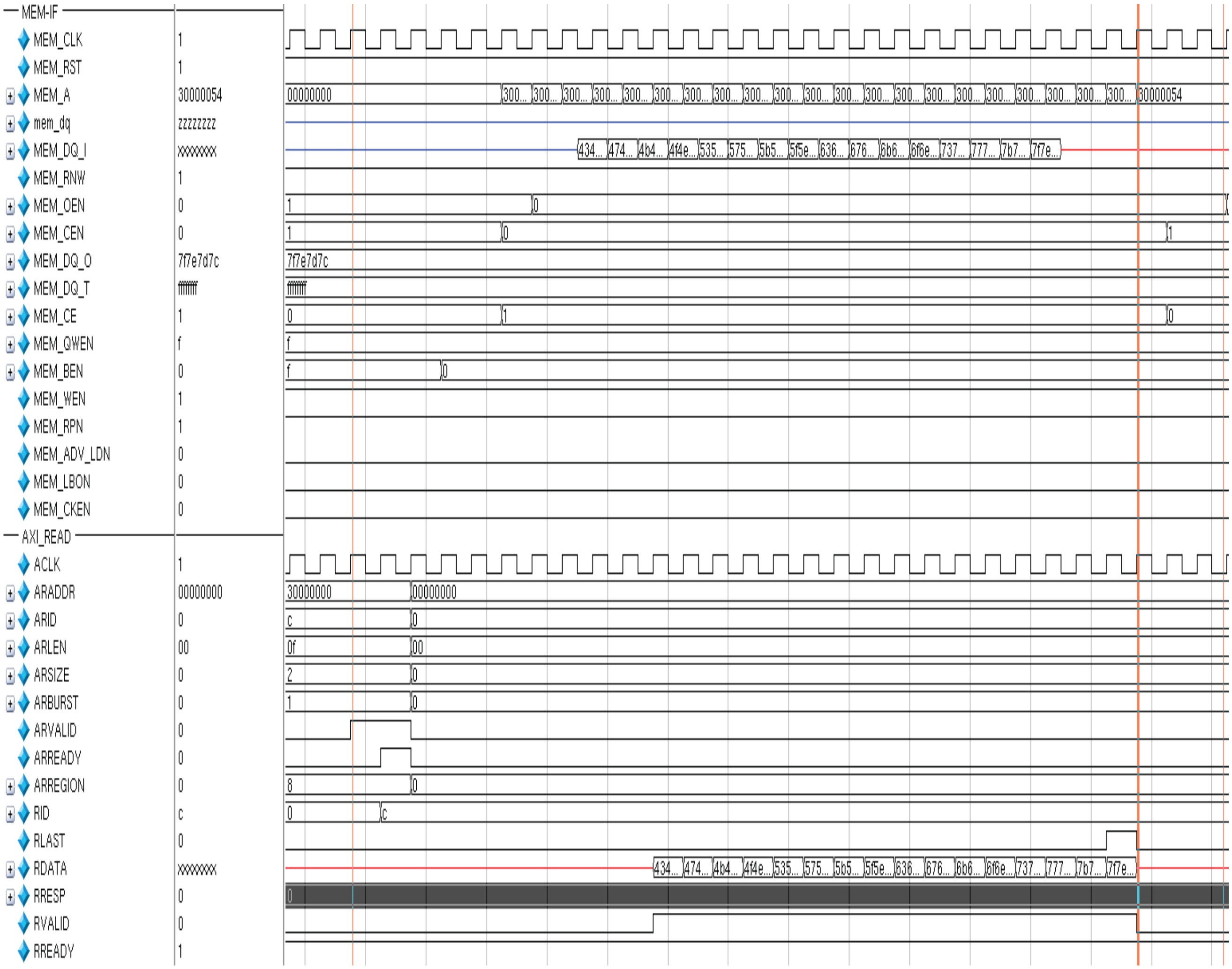
Figure 9 shows the timing waveform for 32 bit Write for synch SRAM Memory.



DS762_09

Figure 9: **32-Bit Write Synch SRAM with Pipeline Delay 2**

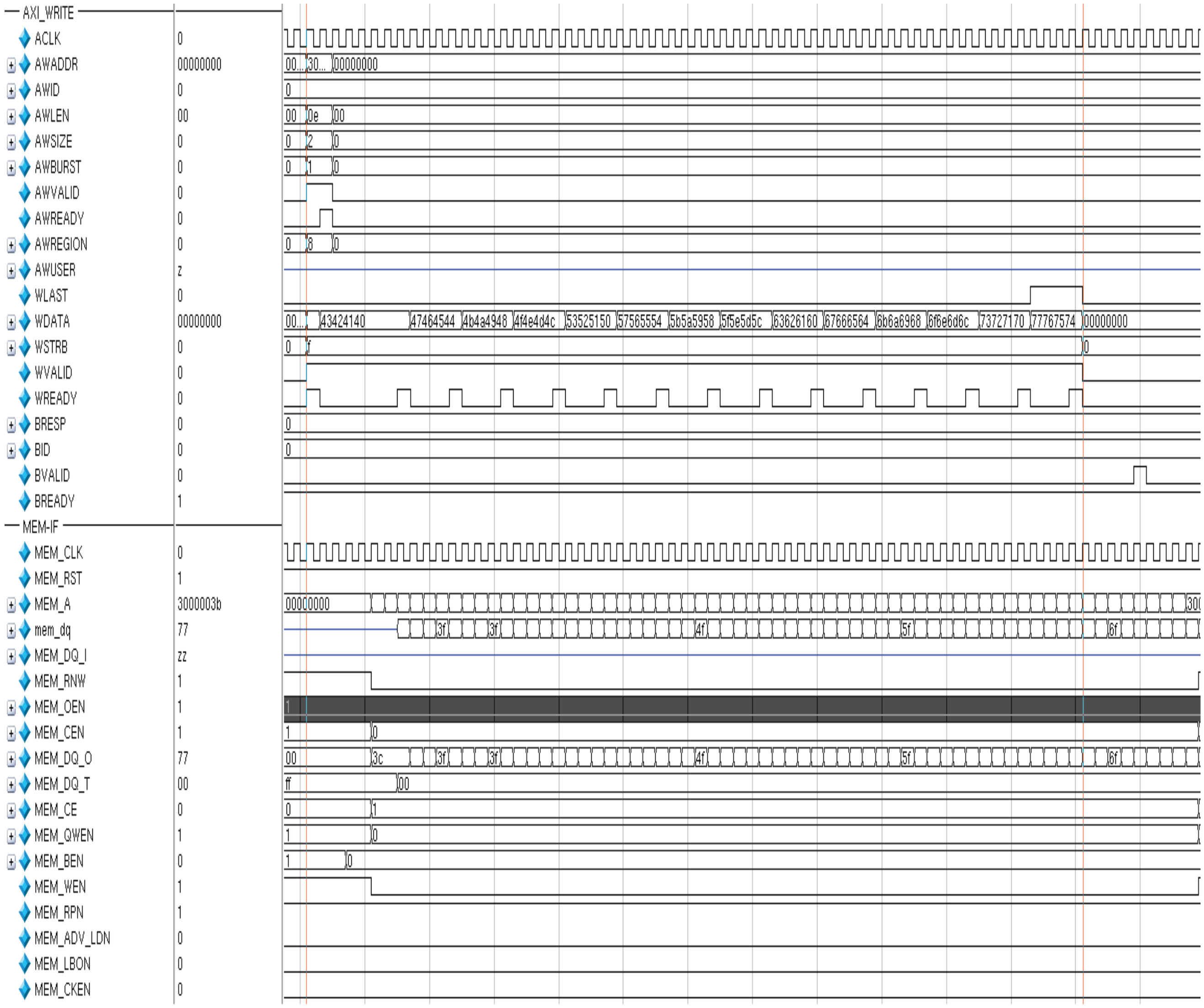
Figure 10 shows the timing waveform for 32 bit Read for a synch SRAM Memory.



DS762_10

Figure 10: **32 Bit Read Synch SRAM With Pipeline Delay 2**

Figure 11 shows the timing waveform for 8 bit Write for synch SRAM Memory.



DS762_11

Figure 11: **8 Bit Write Synch SRAM With Pipeline Delay 2**

Figure 12 shows the timing waveform for 8 bit Read for synch SRAM Memory.

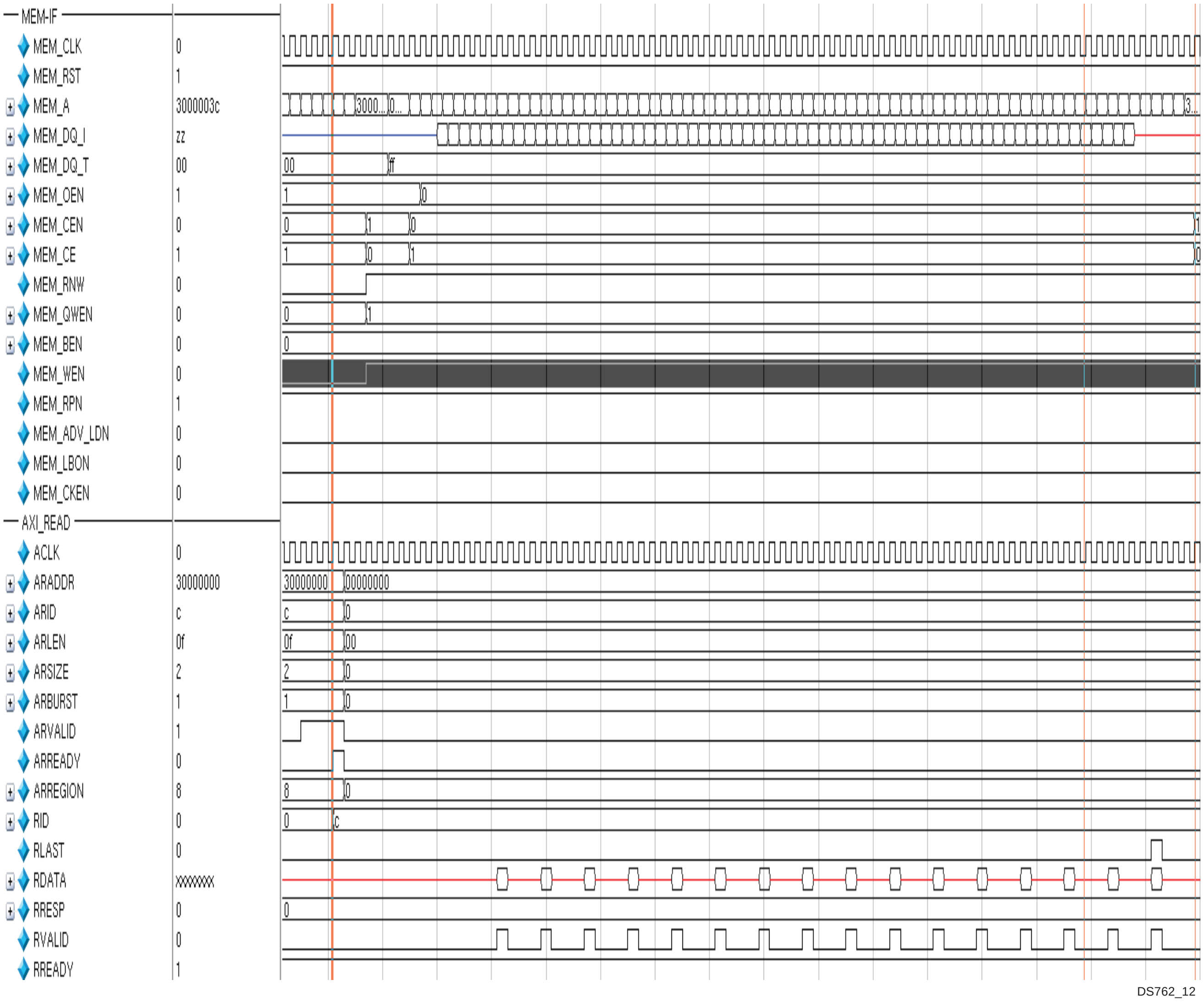
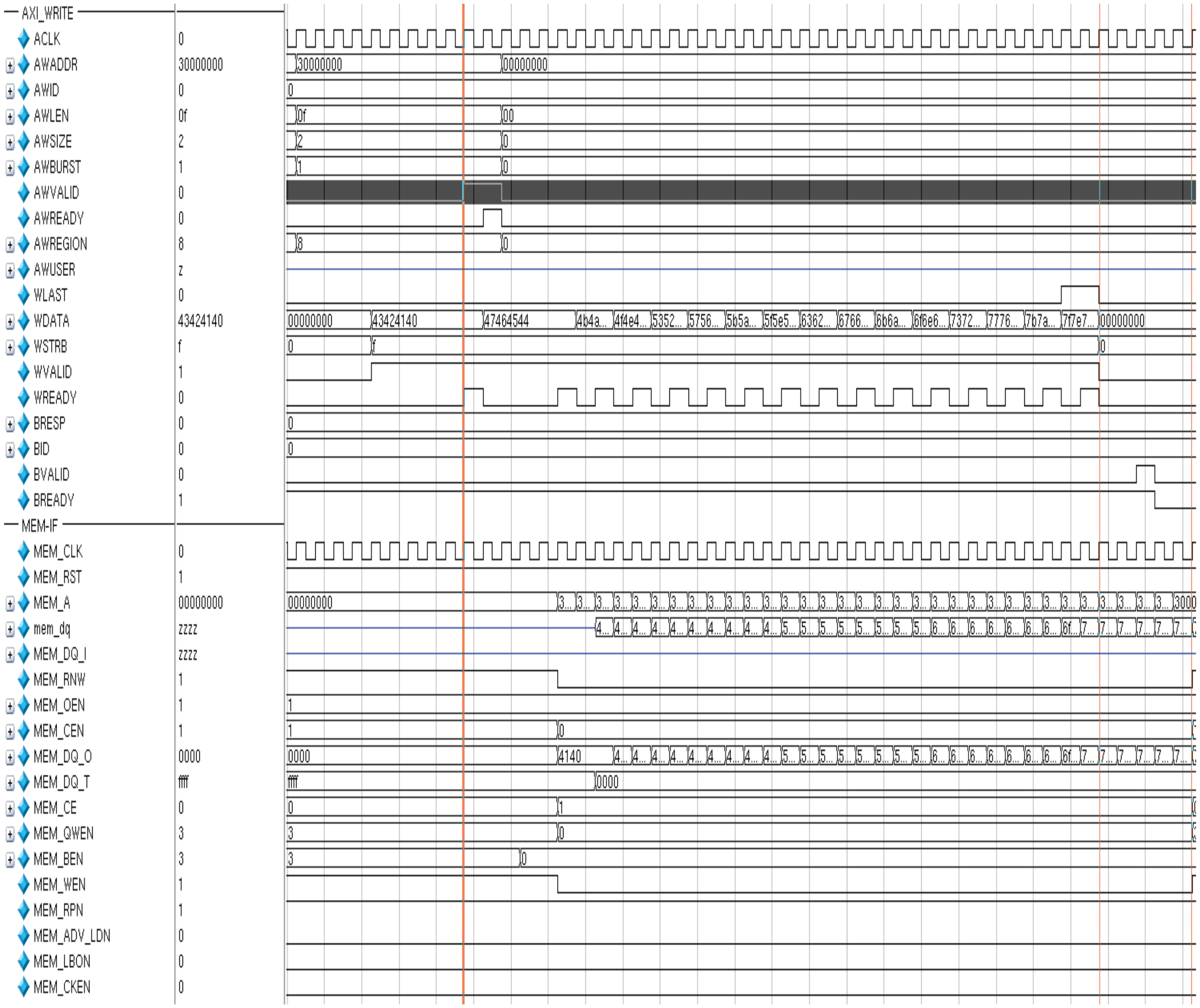


Figure 12: **8 Bit Read Synch SRAM With Pipeline Delay 2**

Figure 13 shows the timing waveform for 16 bit Write for synch SRAM Memory.



DS762_13

Figure 13: **16 Bit Write Synch SRAM With Pipeline Delay 1**

Figure 14 shows the timing waveform for 16 bit Read synch SRAM Memory.

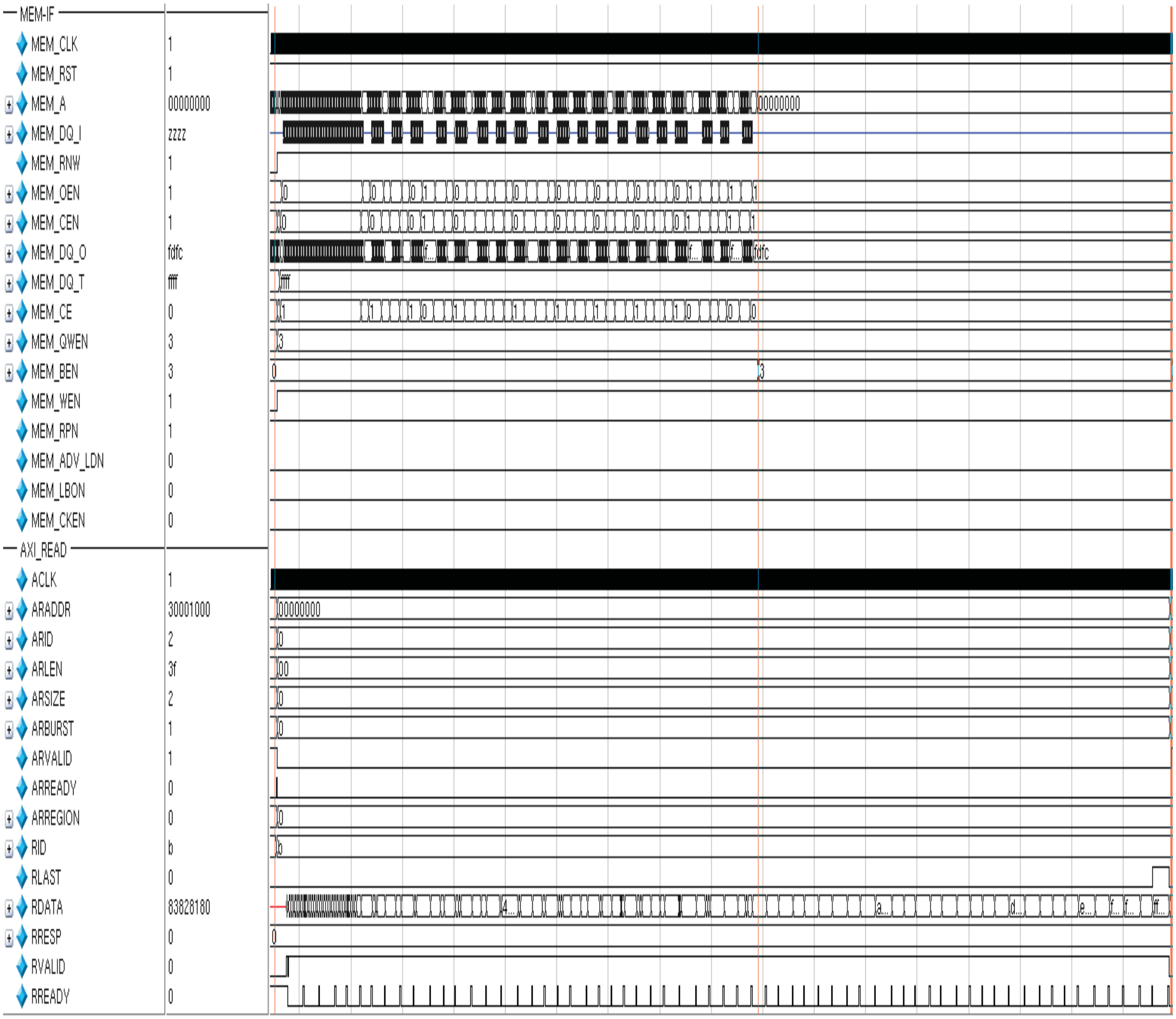


Figure 14: **16 Bit Read Synch SRAM for S_AXI_MEM_ARVALID Throttle**

Asynch SRAM Timing

Figure 15 shows the timing waveform for 32 bit Write for Asynch SRAM Memory.

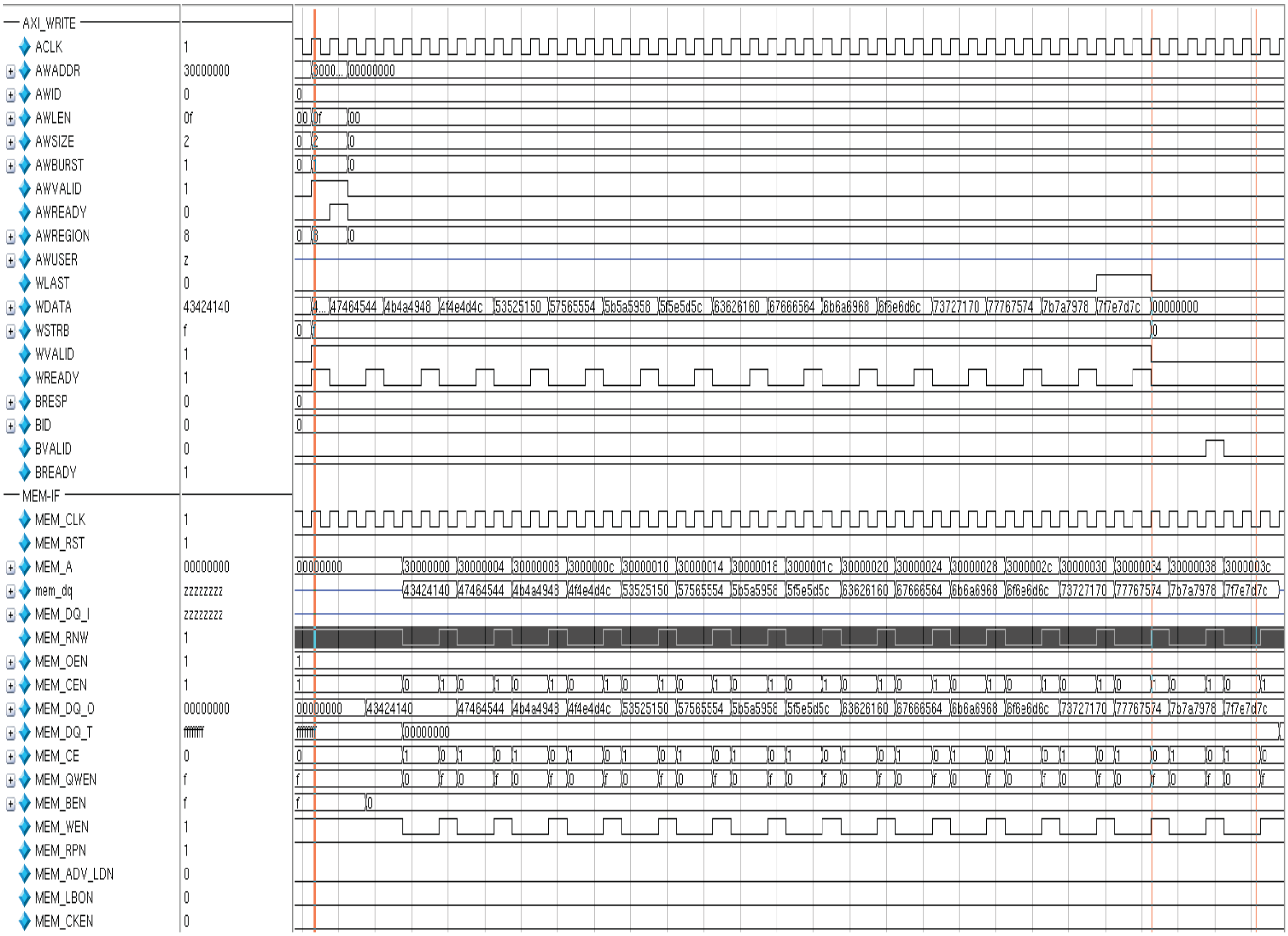


Figure 15: **32 Bit Write Asynch SRAM**

Figure 16 shows the timing waveform for 32 bit Read for Asynch SRAM Memory.

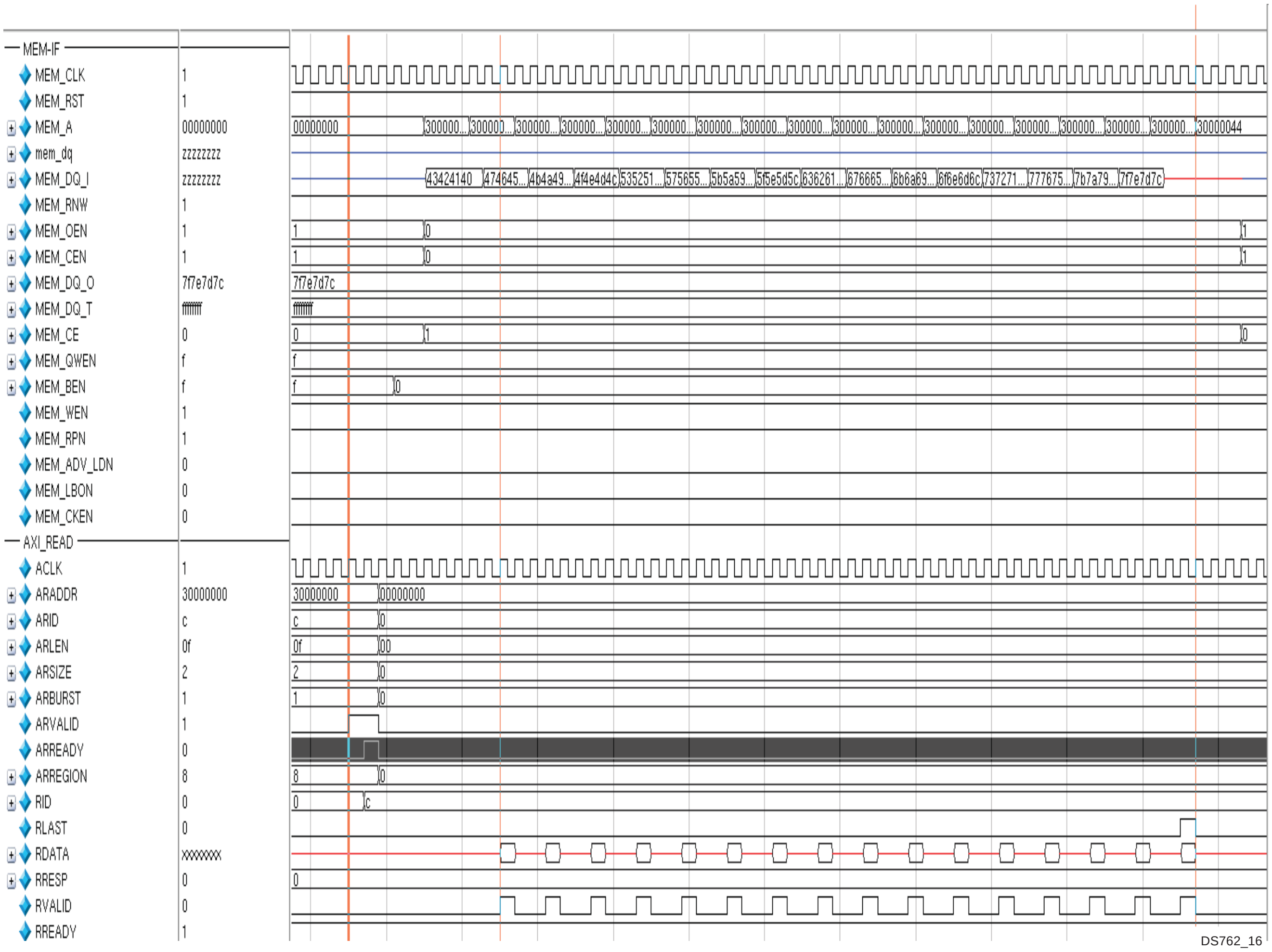
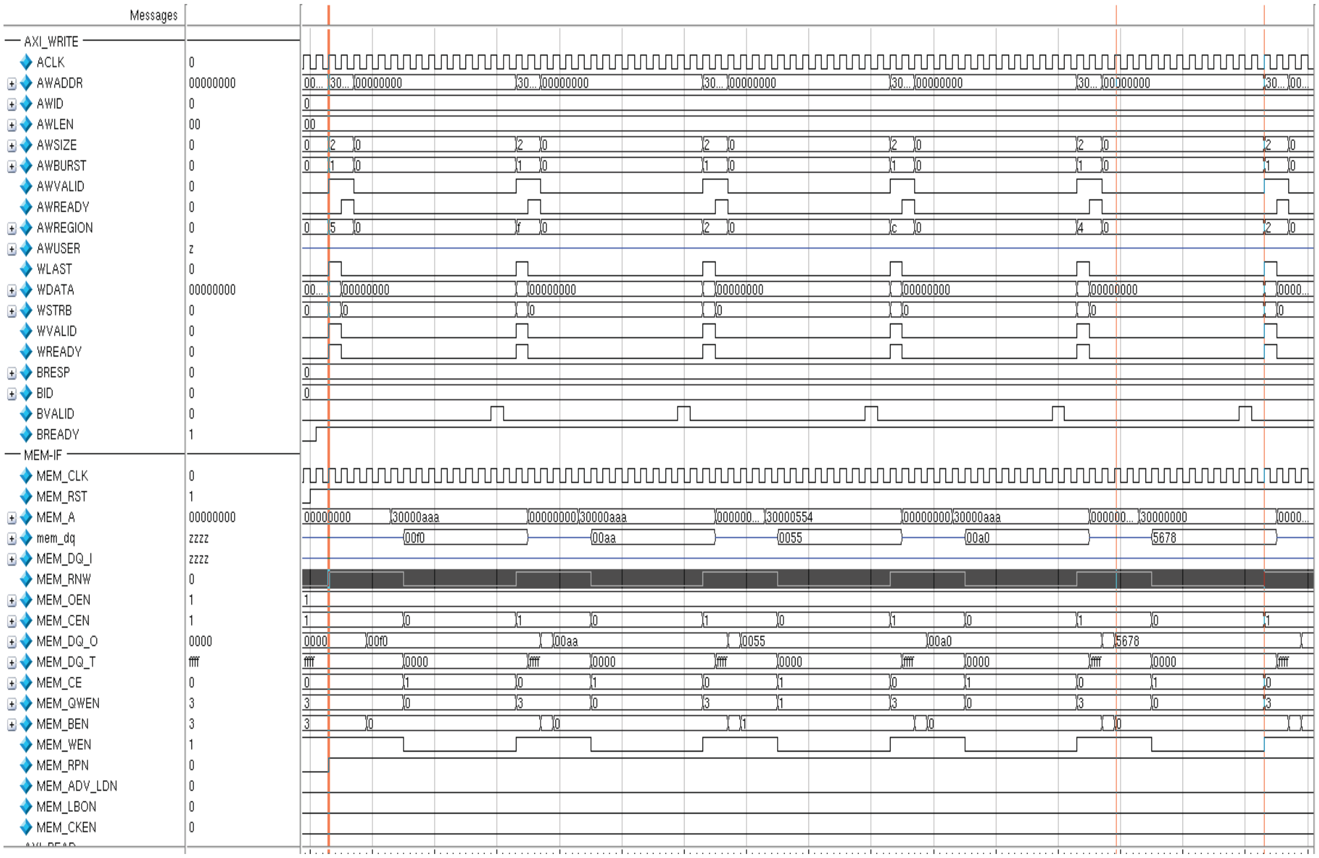


Figure 16: **32 Bit Read Asynch SRAM**

Flash Timing

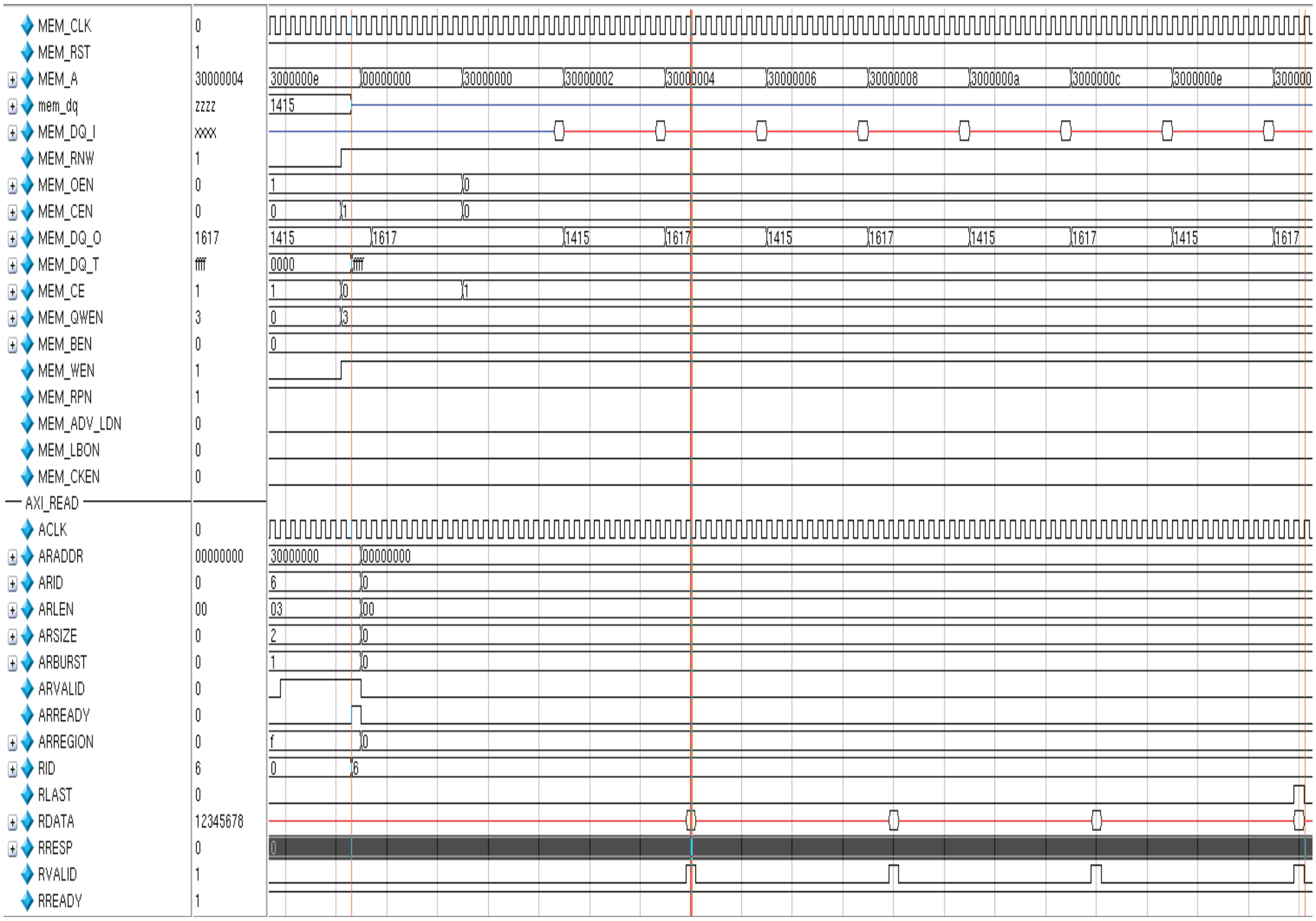
Figure 17 shows the timing waveform for 16 bit Write for Flash Memory.



DS762_17

Figure 17: **16 Bit Write Flash**

Figure 18 shows the timing waveform for 16 bit Read for Flash Memory.

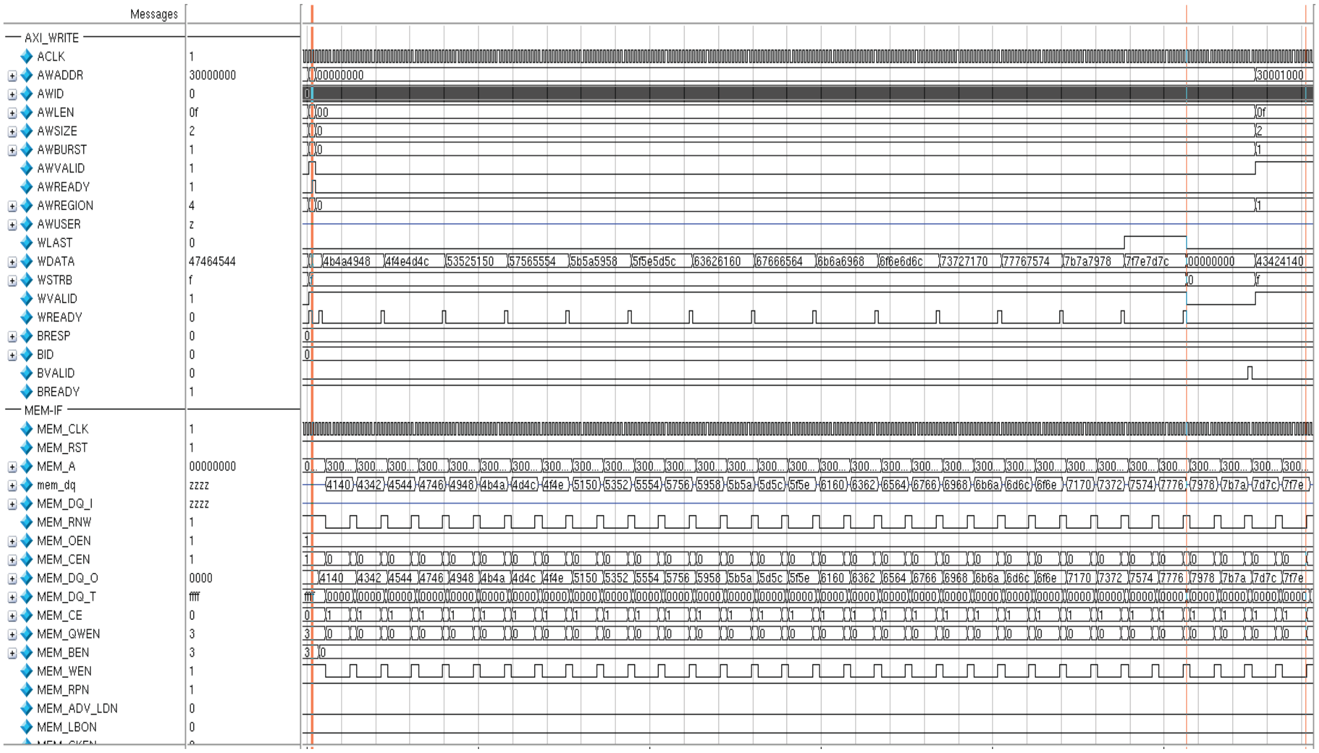


DS762_18

Figure 18: **16 Bit Read Flash**

PSRAM/CellularRAM Timing

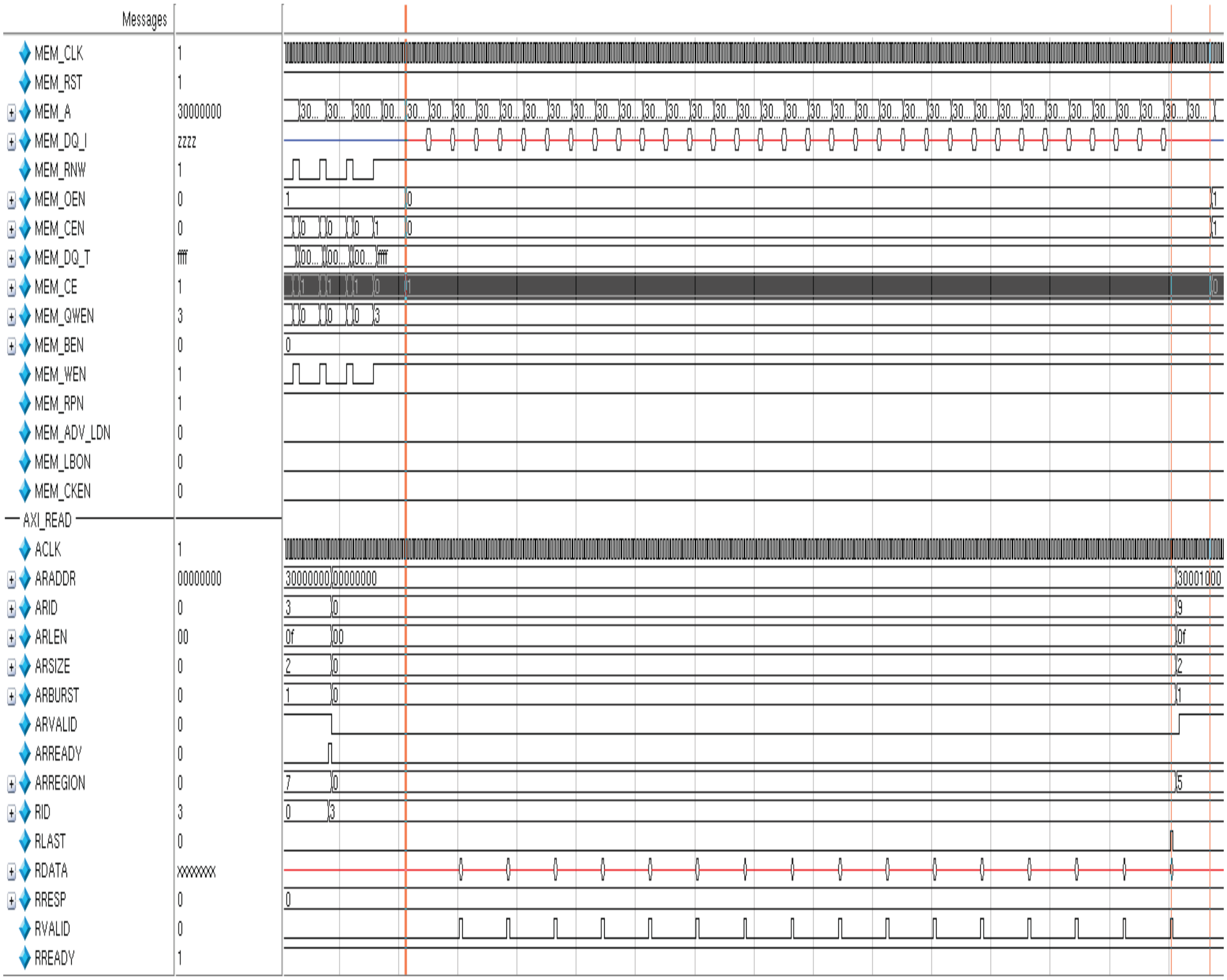
Figure 19 shows the timing waveform for 16 bit Write for Cellular RAM Memory.



DS762_19

Figure 19: **16 Bit Write Cellular RAM**

Figure 20 shows the timing waveform for 16 bit Read for Cellular RAM Memory in Async Mode.



DS762_20

Figure 20: **16 Bit Read Cellular RAM**

Figure 21 shows the timing waveform for 16 bit Write for Cellular RAM Memory in Page Mode.

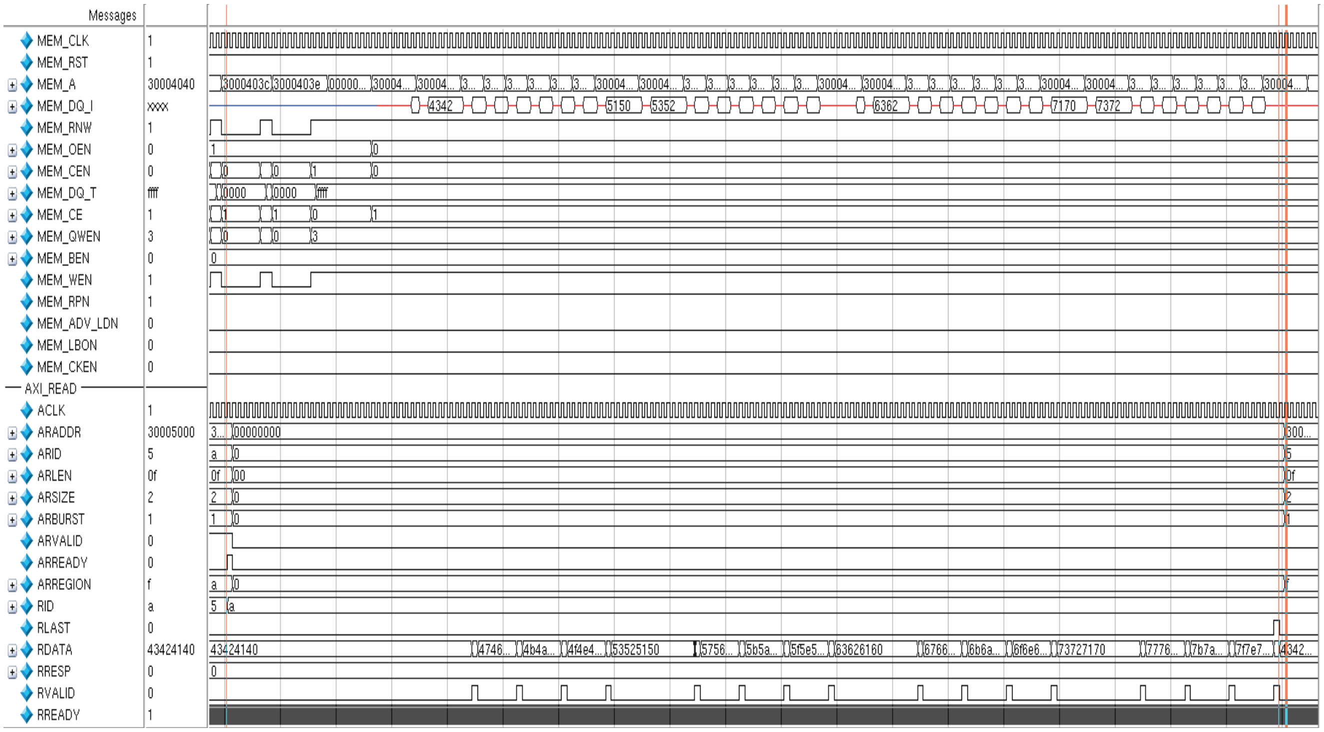
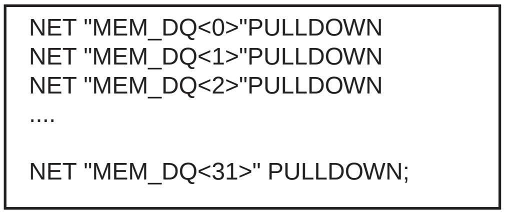


Figure 21: **16 Bit Page Read Cellular RAM**

Design Constraints

Pin Constraints

If external pullups/pulldowns are not available on the MEM_DQ signals, then these pins should be specified to use pullup or pulldown resistors. An example is shown in Figure 22.


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

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

Figure 22: **Pin Constraints**

Design Implementation

Target Technology

The target technology is an FPGA listed in the Supported Device Family field of the LogiCORE IP Facts Table.

Device Utilization and Performance Benchmarks

Core Performance

Because the AXI EMC core will be 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 will 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 22.

Table 22: **Performance and Resource Utilization Benchmarks Artix-7 FPGA (XC7A175TDIE-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	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
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 23.

Table 23: **Performance and Resource Utilization Benchmarks Kintex-7 FPGA (XC7K325T-FFG676-3)**

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	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
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 24.

Table 24: **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
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 25.

Table 25: **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
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 above cases

The AXI EMC resource utilization for various parameter combinations measured with Spartan®-6 as the target device are detailed in Table 26.

Table 26: **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
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 above cases

IP Utilization and Performance

Table 27: **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
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

System 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 23.

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 will vary from the results reported here.

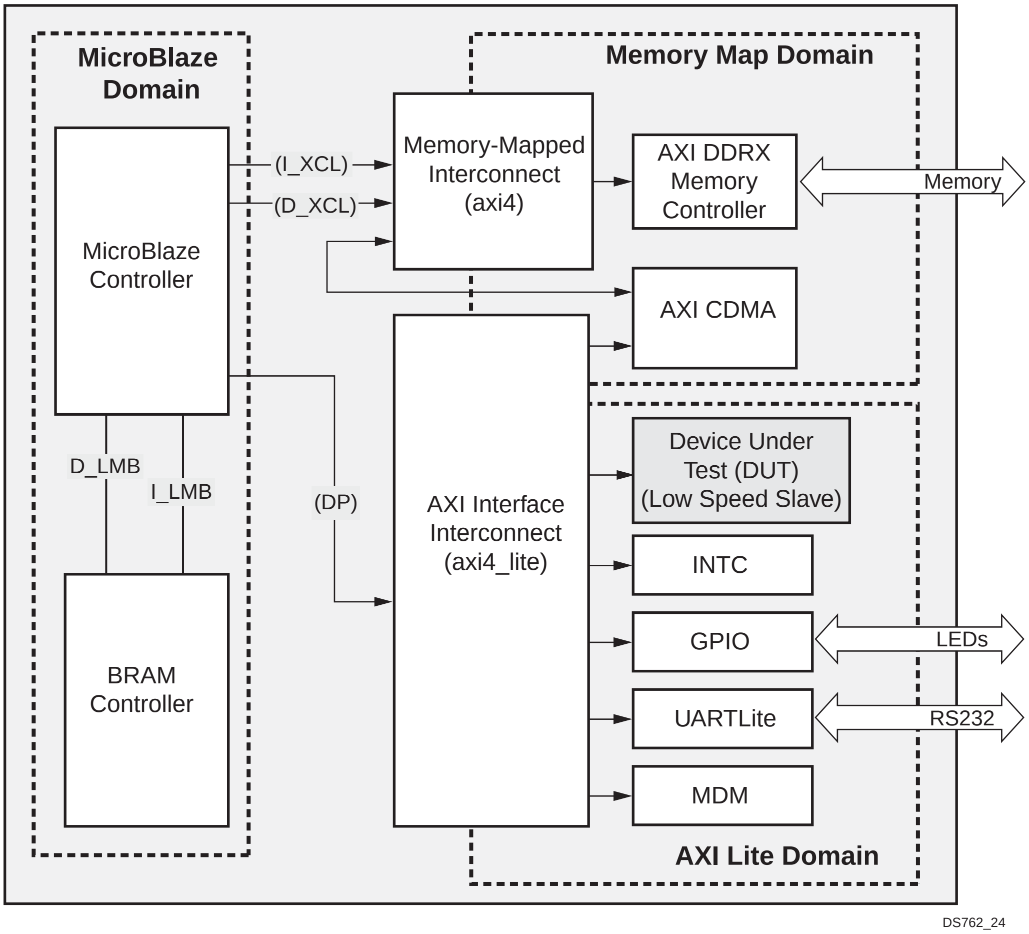


Figure 23: **Virtex-6 and Spartan-6 FPGA Systems with the AXI EMC as the DUT**

The target FPGA was then filled with logic to drive the LUT and BRAM 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 28.

Table 28: **FPGA System Performance**

Target FPGA	Target F_{MAX} (MHz)
xc6vcx130t-1	180
xc6slx45-2	110

The target F_{MAX} is influenced by the exact system and is provided for guidance. It is not a guaranteed value across all systems.

Reference Documents

The following documents contain reference information important to understanding the AXI EMC design:

1. *ARM AMBA® AXI Protocol v2.0 Specification* (ARM IHI 0022C)
2. DS769, *AXI Slave Burst Data Sheet*
3. DS765, *AXI Lite IPIF Data Sheet*
4. DS768, *AXI Interconnect IP Data Sheet*
5. DS180, *7 Series FPGAs Overview*

Support

Xilinx provides technical support for this LogiCORE 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*.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx ISE Design Suite Embedded Edition software under the terms of the Xilinx End User License. The core is generated using the Xilinx ISE Embedded Edition software (EDK).

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 modules and software, please contact your local Xilinx sales representative.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
9/21/10	1.0	Xilinx Initial Release.
3/1/11	1.1	Updated to v1.01a for the 13.1 release.
6/22/11	2.0	Updated for Xilinx tools v13.2.

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.