OPB External Memory Controller (EMC)

DS421 March 11, 2004 **Product Specification**

Introduction

This specification defines the architecture and interface requirements for the EMC. This module supports data transfers between the On-Chip Peripheral Bus (OPB) and external synchronous and asynchronous memory devices.

Example synchronous devices for use with this controller are the synchronous Integrated Device Technology, Inc. IDT71V546 SRAM with ZBT™ Feature. Example asynchronous devices include the IDT71V416S SRAM and Intel 28F128J3A StrataFlash Memory.

The Xilinx EMC design allows the customer to tailor the EMC to suit their application by setting certain parameters to enable/disable features.

Features

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

- Parameterized for up to a total of eight memory (Synchronous/Asynchronous) banks
  - Separate base addresses and address range for each bank of memory
- Separate Control Register for each bank of memory to control memory mode
- OPB V2.0 bus interface with byte-enable support
- Memory width is independent of OPB bus width (memory width must be less than or equal to OPB bus width)
  - Supports memory widths of 32 bits, 16 bits, or 8 bits
  - Memory width can vary by bank
- Parameterizable memory data-width/bus data-width matching
  - Multiple memory cycles will be performed when the memory width is less than the OPB bus width to provide full utilization of the OPB bus
  - Data-width matching can be enabled separately for each memory bank
- Configurable wait states for read, write, read in page, read recovery before write, and write recovery before read

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex-II Pro™, Virtex-II™, Virtex™, Virtex-E™, Spartan™-II	
Version of Core	opb_emc	v1.10b
Resources Used		
	Min	Max
Slices	168	310
LUTs	157	254
FFs	215	445
Block RAMs	0	0
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	5.1i or later	
Verification	N/A	
Simulation	ModelSim SE/EE 5.6e or later	
Synthesis	XST	
Support		
Support provided by Xilinx, Inc.		

- Optional faster access for in-page read accesses (page size 8 bytes)
  - Optional faster access for in-page read accesses (page size 8 bytes)

EMC Background

The EMC module receives instructions from the OPB to read and write to external memory devices.

The OPB Memory Controller provides an interface between the OPBand one to eight external banks of memory components. The controller supports OPB data bus widths of 8 to 32 bits, and memory subsystem widths of 8 to 32 bits. This controller supports the OPB V2.0 byte enable architecture. Any access size up to the width of the OPB data bus is permitted. When the width of the memory is less than the width of the OPB, multiple memory cycles are performed to transfer the data width of the bus if data-width matching has been enabled for that memory bank.

The controller provides basic read/write control signals and the ability to configure the access times for read, read-in-page, write, and recovery times when switching from read to write or write to read. The controller can be configured to support page-mode reads which can be up to six times faster than non-page reads. The in-page detection logic is automatically configured out of the controller if page mode is not required.

When the OPB Memory Controller is set for flash memory control it is organized much like an SRAM interface. This controller assumes that the Flash programming circuitry is built into the Flash components and that the command interface to the Flash is handled in software. Note that the Flash algorithm for programming Flash locations requires that a write command is written to the location before the actual data is written to the location, therefore, data-width matching should not be enabled for banks of Flash memory.

The ZBT control does not support sleep mode, burst mode, parity checking, or parity generating.

EMC Parameters

To allow you to obtain an EMC that is uniquely tailored for your system, certain features can be parameterized in the EMC design. This allows you to configure a design that only utilizes the resources required by your system, and operates with the best possible performance.

The following characteristics of the External Memory Controller are parameterizable and are described in Table 1:

- Number of separate memory banks
- Memory type, synchronous or asynchronous, per memory bank
- Data width of each memory bank
- Pipeline delay of each memory bank (synchronous memories)
- Read and write access times for each memory bank (asynchronous memories)
- Enabling of data-width matching per memory bank
- Base address for the External Memory Controller registers

Table 1: **EMC Parameters**

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Number of Memory Banks	C_NUM_BANKS_MEM	1 - 8	2	integer
Output/Input data and control signals using the falling edge of the clock(11)	C_INCLUDE_NEGEDGE_IOREGS	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
OPB Clock Period	C_OPB_CLK_PERIOD_PS	Integer number of picoseconds	10000	integer
Control Register Bank Base Address	C_BASEADDR	Valid Address Range(3)	None(4)	std_logic_vector
Control Register Bank High Address	C_HIGHADDR	Address range must be a power of 2 and > 0x01F(3)	None(4)	std_logic_vector
Memory Bank x Base Address (x = 0 to 7)	C_MEMx(1,2)_BASEADDR	Valid Address Range(3)	None(4)	std_logic_vector
Memory Bank x High Address (x = 0 to 7)	C_MEMx(1,2)_HIGHADDR	Address range must be a power of 2 and ≤ OPB Address Space(3)	None(4)	std_logic_vector
OPB Data Bus Width	C_OPB_DWIDTH	32	32	integer
OPB Address Bus Width	C_OPB_AWIDTH	8 - 32	32	integer
Width of Memory Bank x Data Bus	C_MEMx_WIDTH(1)	≤ C_OPB_DWIDTH	32	integer
Execute multiple memory access cycles to match width of Memory Bank x data bus to OPBdata bus	C_INCLUDE_DATAWIDTH_MATCHING_x(1,9)	0 = don't include data-width matching 1 = include data-width matching	1	integer
Device ID number (unique for each device)	C_DEV_BLK_ID	0 to 255	1	integer
Enable/Disable Module ID Register	C_DEV_MIR_ENABLE	0 = don't include Module ID Register 1 = include Module ID Register	1	integer

Table 1: **EMC Parameters *(Continued)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Memory type	C_SYNCH_MEM_x(1)	0 = memory type is asynchronous 1 = memory type is synchronous	0	integer
Pipeline delay (only used if C_SYNCH_MEM_x = 1)	C_SYNCH_PIPE_DELAY_x(1)	1 to 2 clocks	2	integer
Address access time for reads when EMCCRx[0] = 0 or EMCCRx[1] = 1 and access is out of page(6)	C_READ_ADDR_TO_OUT_SLOW_PS_x(1)	Integer number of picoseconds	0(7)	integer
Address valid to end of write when EMCCRx[0] = 0(5,6)	C_WRITE_ADDR_TO_OUT_SLOW_PS_x(1)	Integer number of picoseconds	0(7)	integer
Minimum time that write enable goes low (follows address being driven out). Applies to all writes.(5,6)	C_WRITE_MIN_PULSE_WIDTH_PS_ x(1)	Integer number of picoseconds	0(7)	integer
Address access time for reads when EMCCRx[0] = 1 or EMCCRx[1] = 1 and access is in page(6)	C_READ_ADDR_TO_OUT_FAST_PS_x(1)	Integer number of picoseconds	0(7)	integer
Address valid to end of write when EMCCRx[0] = 1(5,6)	C_WRITE_ADDR_TO_OUT_FAST_PS_x(1)	Integer number of picoseconds	0(7)	integer
Delay inserted before Write Enable goes low if previous access was Read. (6)	C_READ_RECOVERY_BEFORE_WR_PS_x(1,10)	Integer number of picoseconds	0(7)	integer

Table 1: **EMC Parameters *(Continued)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Delay inserted before Output Enable goes low if previous access was Write. (6)	C_WRITE_RECOVERY_BEFORE_RD_PS_x(1,10)	Integer number of picoseconds	0(7)	integer
Maximum width of the memory devices in all of the banks - width of memory bus out to the devices **AUTOCALCULATED** (8)	C_MAX_MEM_WIDTH	8,16,32	32	

Notes:

1. x = values for memory banks 0 to 7
2. This design can accommodate up to 8 banks of memory. The address range generics are designated as C_MEM0_BASEADDR, C_MEM1_BASEADDR, C_MEM0_HIGHADDR, C_MEM1_HIGHADDR, etc.
3. Address range specified by C_BASEADDR and C_HIGHADDR must be a power of 2 and >=0x0144 when C_DEV_MIR_ENABLE=1 or >= 0x01F when C_DEV_MIR_ENABLE=0. C_MEMx_BASEADDR and C_MEMx_HIGHADDR must be a power of 2 and less than or equal to the OPB address space.
4. No default value will be specified for C_BASEADDR and C_HIGHADDR AND C_MEMx_BASEADDR, 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. These generics must be a power of 2 and encompass the memory size for C_MEMx_BASEADDR, C_MEMx_HIGHADDR.
5. Write enable low time is the maximum of C_WR_ADDR_TO_OUT_FAST/SLOW_PS and C_WR_MIN_PULSE_WIDTH.
6. As specified by the memory device data sheet.
7. 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.
8. This parameter is automatically calculated when using CoreGen, otherwise the user must set this parameter to the maximum value of the C_MEMx_WIDTH generics.
9. Data-width matching should be set to 0 for memory banks containing Flash memory devices due to the algorithm required for writing Flash memory.
10. These parameters only have meaning for Flash memory devices and should be set to 0 for all other memory types.
11. This parameter should only be set to 1 under the following conditions:
    - Tfpga_outdelay + Tsetup_memory + Tboard_route_delay < Clock_period/2
    - and
    - Tmemory_outdelay + Tfpga_setup + Tboard_route_delay < Clock_period/2

    See section "IO Registers" on page 18 for more information on using this parameter.

Allowable Parameter Combinations

The EMC supports up to 8 banks of Synchronous and/or Asynchronous memory. Each bank of memory has it's own independent base address and address range. The address range of a bank of memory is restricted to be a power of 2.

If the desired address range is represented by 2^n, then the *n* least significant bits of the base address must be 0. 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.

If C_DEV_MIR_ENABLE = 1, then the minimum address range specified by C_BASEADDR and C_HIGHADDR = 0x144. If C_DEV_MIR_ENABLE = 0, then the minimum address range specified by C_BASEADDR and C_HIGHADDR=0x01F.

If C_SYNCH_MEMORY=1, then C_SYNCH_PIPE_DELAY 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_SYNCH_MEMORY=0, C_SYNCH_PIPE_DELAY 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_DATAWIDTH_MATCHING must be set to 0 for memory banks containing Flash devices. Each data write to a particular Flash memory location must be preceded by a write command and is therefore not compatible with the implementation of data-width matching in this controller.

C_INCLUDE_NEGEDGE_IOREGS provides no benefit when interfacing to asynchronous memories. Therefore, if there are no synchronous memories in the system, this parameter should be set to 0.

EMC I/O Signals

The I/O signals for the EMC are listed in Table 2.

Table 2: **EMC I/O Signals**

Signal Name	Interface	I/O	Description
OPB_Clk	OPB	I	OPB Clock
OPB_Rst	OPB	I	OPB Reset
OPB_ABus(0:C_OPB_AWIDTH-1)	OPB	I	OPB Address Bus
OPB_BE(0:C_OPB_DWIDTH/8-1)	OPB	I	OPB Byte Enables
OPB_DBus(0:C_OPB_DWIDTH-1)	OPB	I	OPB Data Bus
OPB_RNW	OPB	I	OPB Read, Not Write
OPB_select	OPB	I	OPB Select
OPB_seqAddr	OPB	I	OPB Sequential Address
Sln_DBus(0:C_OPB_DWIDTH-1)	OPB	O	Memory Controller Data Bus
Sln_errAck	OPB	O	Memory Controller Error Acknowledge
Sln_retry	OPB	O	Memory Controller Retry
Sln_toutSup	OPB	O	Memory Controller Timeout Suppress
Sln_xferAck	OPB	O	Memory Controller Transfer Acknowledge
Mem_DQ_I(0:C_MAX_MEM_WIDTH-1)	IP Core	I	Memory Input Data Bus
Mem_DQ_O(0:C_MAX_MEM_WIDTH-1)	IP Core	O	Memory Output Data Bus
Mem_DQ_T(0:C_MAX_MEM_WIDTH-1	IP Core	O	Memory Output 3-state Signal
Mem_A(0:C_OPB_AWDITH-1)	IP Core	O	Memory Address Bus
Mem_RPN	IP Core	O	Memory Reset/Power Down
Mem_CEN(0:C_NUM_BANKS_MEM-1)	IP Core	O	Memory Chip Enables (Active Low)(1)
Mem_OEN(0:C_NUM_BANKS_MEM-1)	IP Core	O	Memory Output Enable
Mem_WEN	IP Core	O	Memory Write Enable
Mem_QWEN(0:(C_MEM_WIDTH/8) -1)	IP Core	O	Memory Qualified Write Enables
Mem_BEN(0:(C_MEM_WIDTH/8) -1)	IP Core	O	Memory Byte Enables
Mem_CE(0:C_NUM_BANKS_MEM-1)	IP core	O	Memory Chip Enables (Active High)(1)
Mem_ADV_LDN	IP core	O	Memory Advance Burst Address/Load New Address

Table 2: **EMC I/O Signals *(Continued)***

Signal Name	Interface	I/O	Description
Mem_LBON	IP core	O	Memory Linear/Interleaved Burst Order
Mem_CKEN	IP core	O	Memory Clock Enable
Mem_RNW	IP core	O	Memory Read/Write Signal

Notes:
1. 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.

Parameter-Port Dependencies

The width of many of the EMC signals depends on the number of memories in the system and the width of the various data and address busses. In addition, when certain features are parameterized away, the related input signals are unconnected and the related output signals are set to a constant values. The dependencies between the EMC design parameters and I/O signals are shown inTable 3.

Table 3: **Parameter-Port Dependencies**

Name	Affects	Depends	Relationship Description
C_OPB_DWIDTH	OPB_BE OPB_DBus Sln_DBus		Number of Byte Enables, width of the OPB Data Bus, and the width of the Slave Data Bus all vary based on the PLB data width
C_OPB_AWDITH	OPB_ABUS Mem_A		Width of the OPB Address Bus and the width of the Memory Address Bus all vary based on the OPBaddress width
C_MAX_MEM_WIDTH	Mem_DQ_I Mem_DQ_O Mem_BEN Mem_QWEN		Width of the memory interface and control busses vary based on the maximum data width of the memory banks
C_NUM_BANKS_MEM	Mem_CEN Mem_CE Mem_OEN		Status signal per Memory Bank Chip Enable (low) per Memory Bank Chip Enable (high) per Memory Bank Output Enable per Memory Bank
I/O Signals			
OPB_ABus		C_OPB_AWIDTH	Width varies with the width of the OPB Address Bus
OPB_BE		C_OPBDWIDTH	Width varies with the width of the OPB Data Bus
Mem_DQ_I		C_MEM_WIDTH	Width varies with the width of the memory specified
Mem_DQ_O		C_MEM_WIDTH	Width varies with the width of the memory specified
Mem_A		C_OPB_AWDITH	Width varies with the width of the OPB Address Bus
Mem_CEN		C_NUM_BANKS_MEM	Width varies with the width of the number of memory banks specified
Mem_OEN		C_NUM_BANKS_MEM	Width varies with the width of the number of memory banks specified

Table 3: **Parameter-Port Dependencies *(Continued)***

Name	Affects	Depends	Relationship Description
Mem_QWEN		C_MEM_WIDTH	Width varies with the width of the memory specified
Mem_BEN		C_MEM_WIDTH	Width varies with the width of the memory specified
Mem_CE		C_NUM_BANKS_MEM	Width varies with the width of the number of memory banks specified

EMC Address Map and Register Descriptions

EMC Address Map

The EMC contains addressable control registers for each memory bank which sets timing for write operations. The base address for these registers is set in the parameter C_BASEADDR. Table 4 shows all of the EMC control registers and their addresses when the EMC has been parameterized to support all 8 memory banks. Control registers are only present in the design for memory banks that are used.

Table 4: **EMC Control Registers**

Register Name	OPB Address	Access
MEM0 Control Register (EMCCR0)	C_BASEADDR + 0x00	Read/Write
MEM1 Control Register (EMCCR1)	C_BASEADDR + 0x04	Read/Write
MEM2 Control Register (EMCCR2)	C_BASEADDR + 0x08	Read/Write
MEM3 Control Register (EMCCR3)	C_BASEADDR + 0x0C	Read/Write
MEM4 Control Register (EMCCR4)	C_BASEADDR + 0x10	Read/Write
MEM5 Control Register (EMCCR5)	C_BASEADDR + 0x14	Read/Write
MEM6 Control Register (EMCCR6)	C_BASEADDR + 0x18	Read/Write
MEM7 Control Register (EMCCR7)	C_BASEADDR + 0x1C	Read/Write

The EMC can be parameterized to contain a Module Identification Register (MIR). This register has an independent address located at a calculated offset of 0x140. Therefore, if C_DEV_MIR_ENABLE is set equal to "1" then the MIR is located atC_BASEADDR + 0x00000140.

Table 5: **EMC Module Identification Register**

Register Name	OPB Address	Access
EMC Module Identification Register (MIR)	C_BASEADDR + 0x140	Read

The EMC supports up to 8 banks memory. Each bank of memory has it's own independent base address and address range. The address range of a bank of memory is restricted to be a power of 2. If the desired address range is represented by 2^n, then the *n* least significant bits of the base address must be 0. 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 addresses for each bank of memory are shown in Table 6.

Table 6: **EMC Memory Banks**

Memory	Base Address	High Address	Access
Bank 0	C_MEM0_BASEADDR	C_MEM0_HIGHADDR	Read/Write
Bank 1	C_MEM1_BASEADDR	C_MEM1_HIGHADDR	Read/Write
Bank 2	C_MEM2_BASEADDR	C_MEM2_HIGHADDR	Read/Write
Bank 3	C_MEM3_BASEADDR	C_MEM3_HIGHADDR	Read/Write
Bank 4	C_MEM4_BASEADDR	C_MEM4_HIGHADDR	Read/Write
Bank 5	C_MEM5_BASEADDR	C_MEM5_HIGHADDR	Read/Write
Bank 6	C_MEM6_BASEADDR	C_MEM6_HIGHADDR	Read/Write
Bank 7	C_MEM7_BASEADDR	C_MEM7_HIGHADDR	Read/Write

EMC Control Register (EMCCR)

The EMC Control Register is shown in Figure 1. Table 7 shows the Control Register bit definitions.

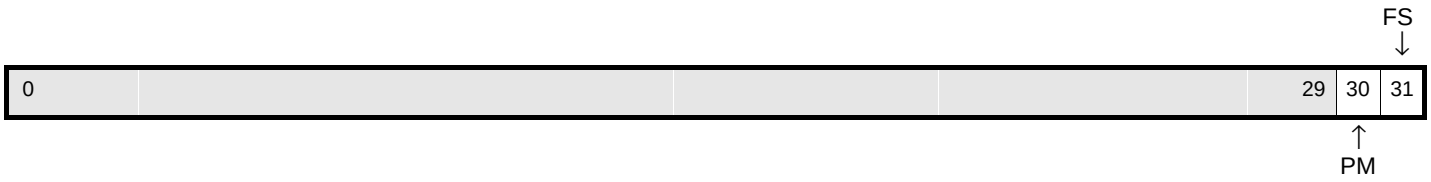


Figure 1: **EMC Control Register**

Table 7: **EMC Control Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 29	Reserved			
30	PM	Read/Write	'0'(1)	**Page Mode Enable.** Determines whether or not in-page detection logic is created with a corresponding decrease in read access time for in-page reads. • '0' - Page Mode is Disabled • '1' - Page Mode is Enabled
31	FS	Read/Write	'0'(1)	**Fast/Slow Mode Enable.** Determines the number of Wait States required based on the input timing parameters. • '0' - Slow Access Time • '1' - Fast Access Time

The functionality of the EMC Control Register bits is described in Table 8. Note that the bits in this register really only have meaning for FLASH memory devices.

Table 8: **EMC Control Register Bit Functionality**

Read/Write	PM Enable	FS Enable	Function
Read	0	0	Slow access
	0	1	Fast access
	1	X	Fast in page, slow not in page
Write	X	0	Slow access
	X	1	Fast access

Module Identification Register (MIR)

The MIR is actually located within the OPB IPIF. This register is read-only and reflects the current IPIF version as well as the Block ID specified by the C_DEV_BLK_ID parameter. Figure 2 shows the MIR and Table 9 details the MIR bit definitions. Note that the only field that is variable is the Block ID field.

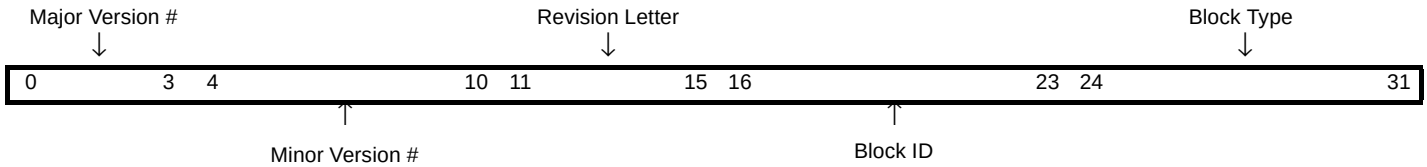


Figure 2: **Module Identification Register**

Table 9: **Module Identification Register (MIR) Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 3	Major Version #	Read	0010	OPBIPIF Major Version Number = 2
4 - 10	Minor Version #	Read	0000000000	OPBIPIF Minor Version Number = 0
11-15	Revision Letter	Read	00000	OPBIPIF Revision Letter = 00000(a)
16 - 23	Block ID	Read	C_DEV_BLK_ID	Block Identification Number. The value read is an echo of the input parameter C_DEV_BLK_ID
24 - 31	Block Type	Read	00000001	OPBIPIF Block Type = 1

Memory Data Types and Organization

Memory can be accessed through the EMC as one of three types: byte (8 bits), halfword (2 bytes), or word (4 bytes). From the point of view of the OPB, data is organized as big-endian. The bit and byte labeling for the big-endian data types is shown below in Figure 3.

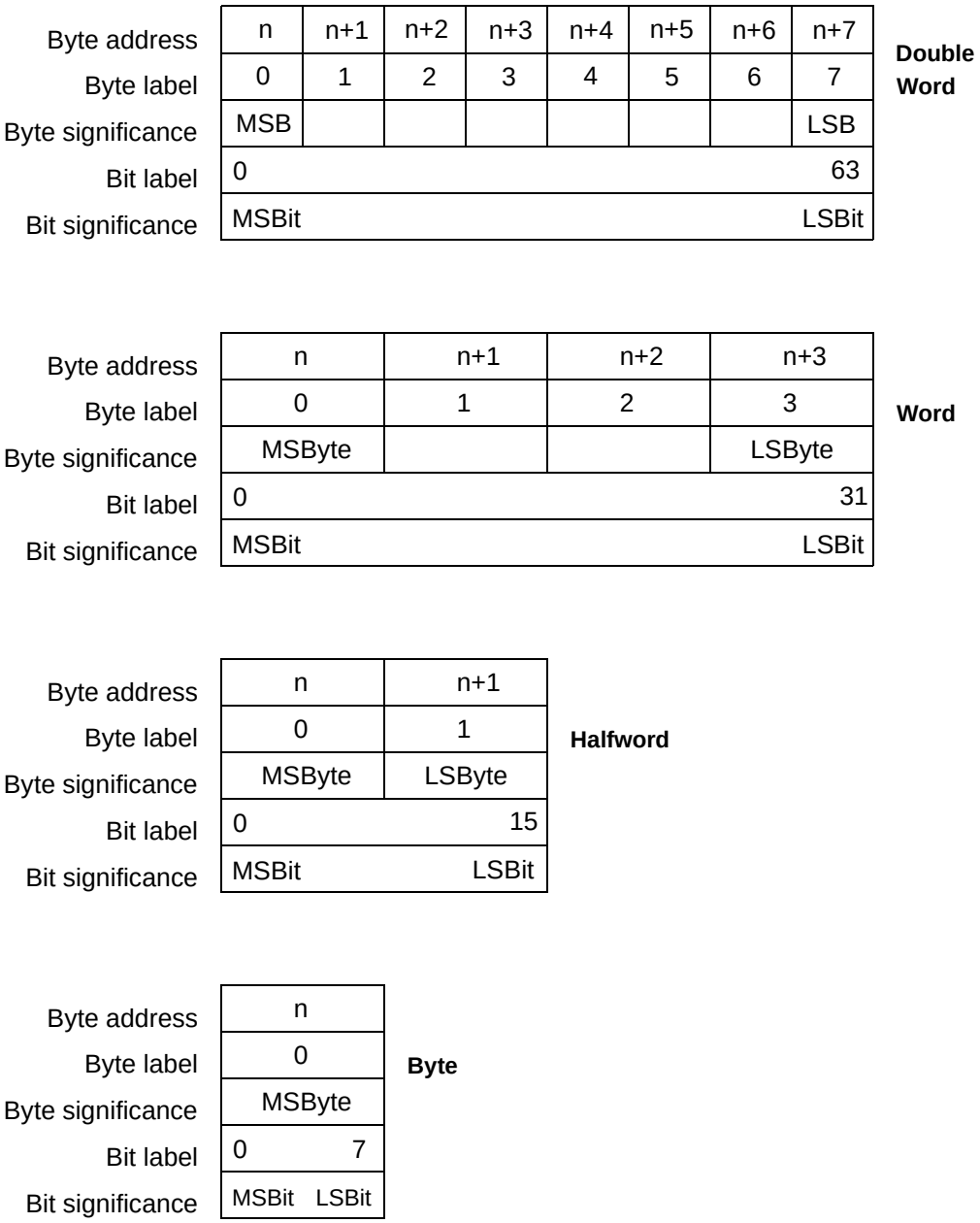
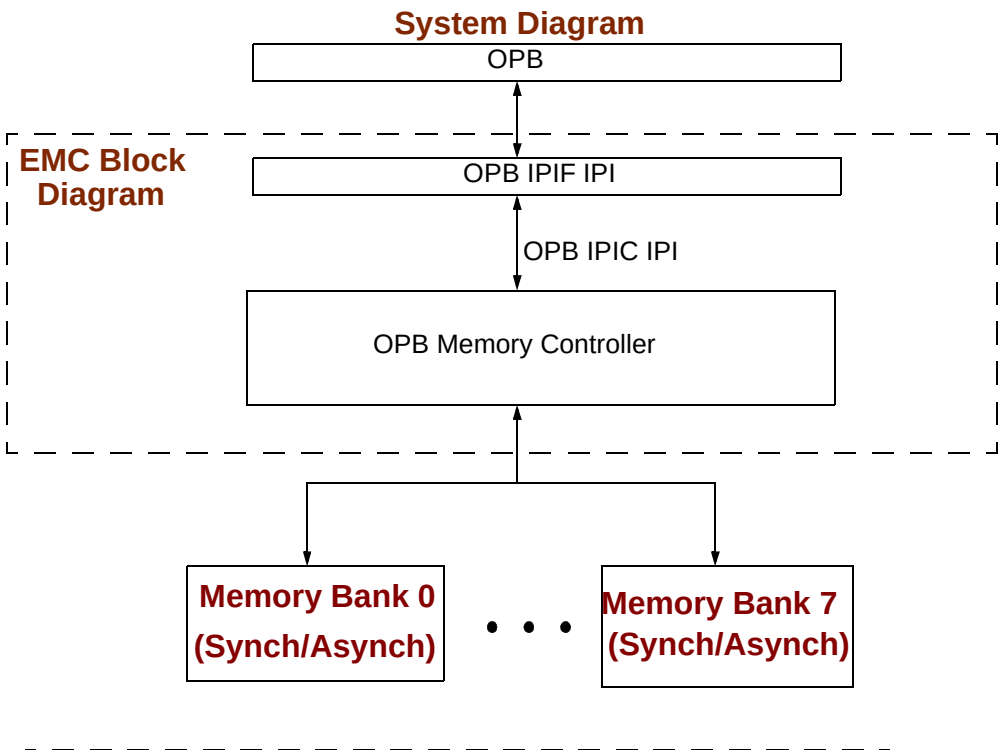


Figure 3: **Big-Endian Data Types**

EMC Block Diagram

The top-level block diagram for the EMC is shown in Figure 4. The detailed block diagram of the EMC is found in Figure 5.



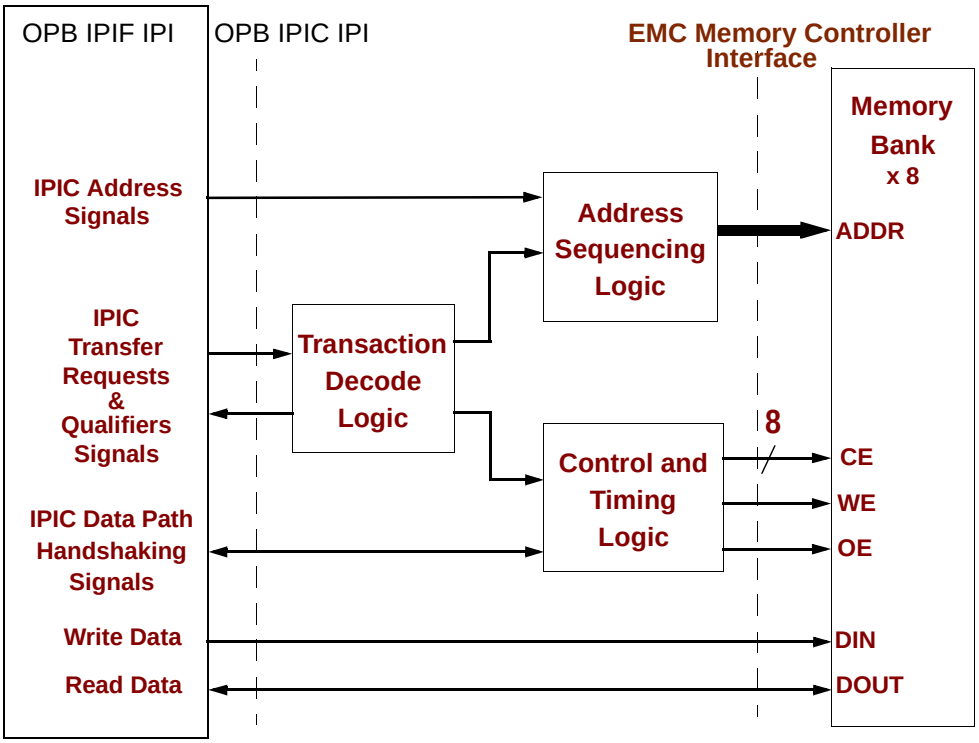


Figure 4: **EMC Top-level Block Diagram**

Figure 5 depicts the Memory Control Block Diagram implemented in the EMC.

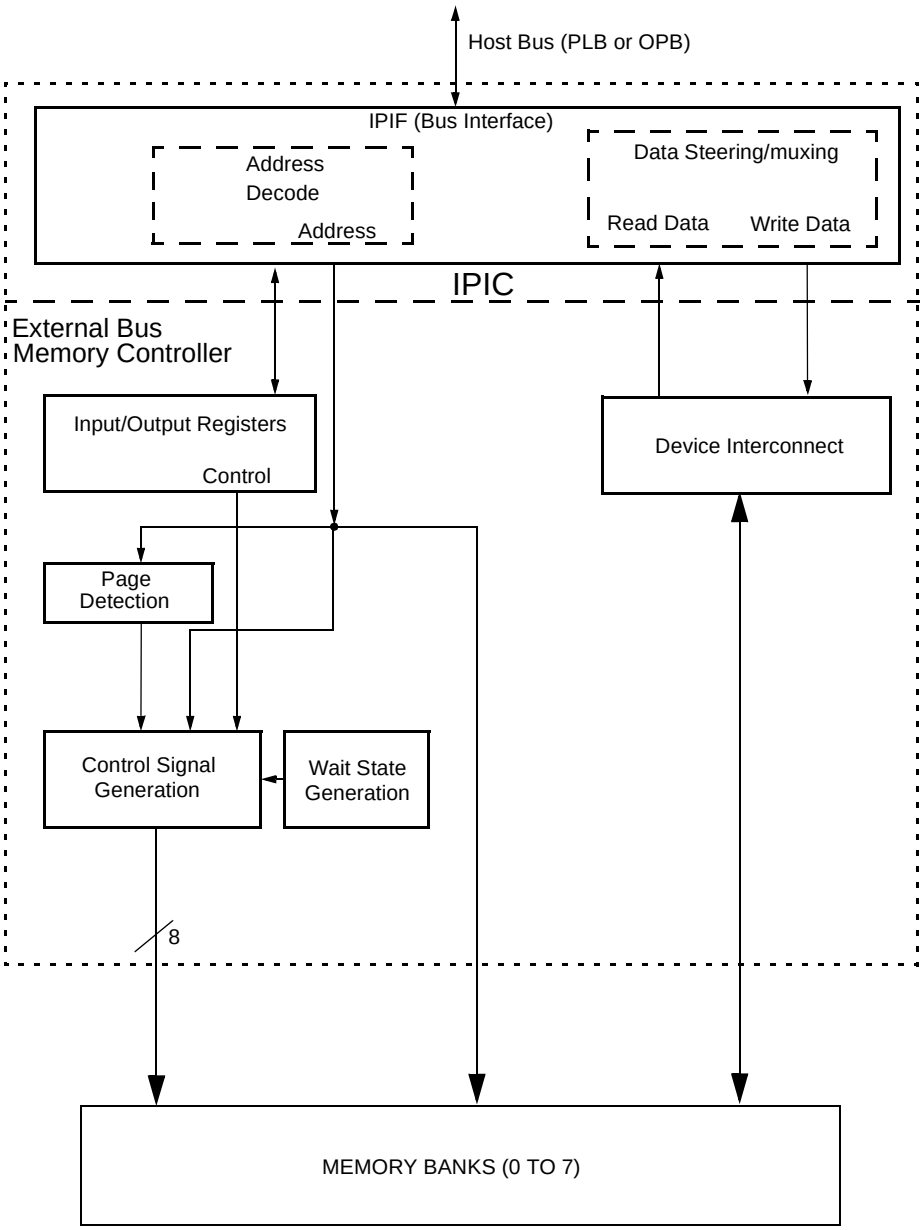


Figure 5: **EMC Memory Control Block Diagram**

Memory Controller Operation

Memory Controller State Machine

The memory controller state machine is a single state machine but is depicted in two separate diagrams, one for asynchronous memories and one for synchronous memories, to improve readability. Figure 6 depicts the memory controller state machine when the addressed memory bank is an asynchronous memory type.

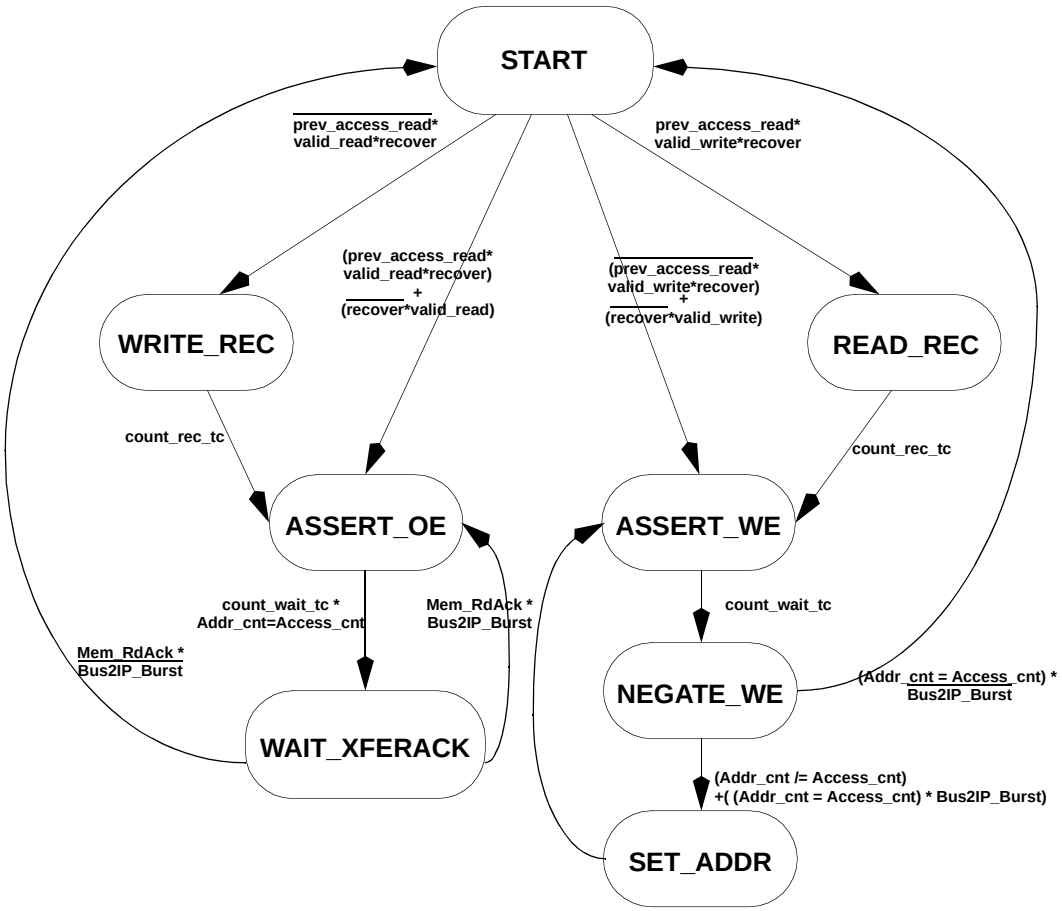


Figure 6: **EMC Memory Control State Machine for Asynchronous Memories**

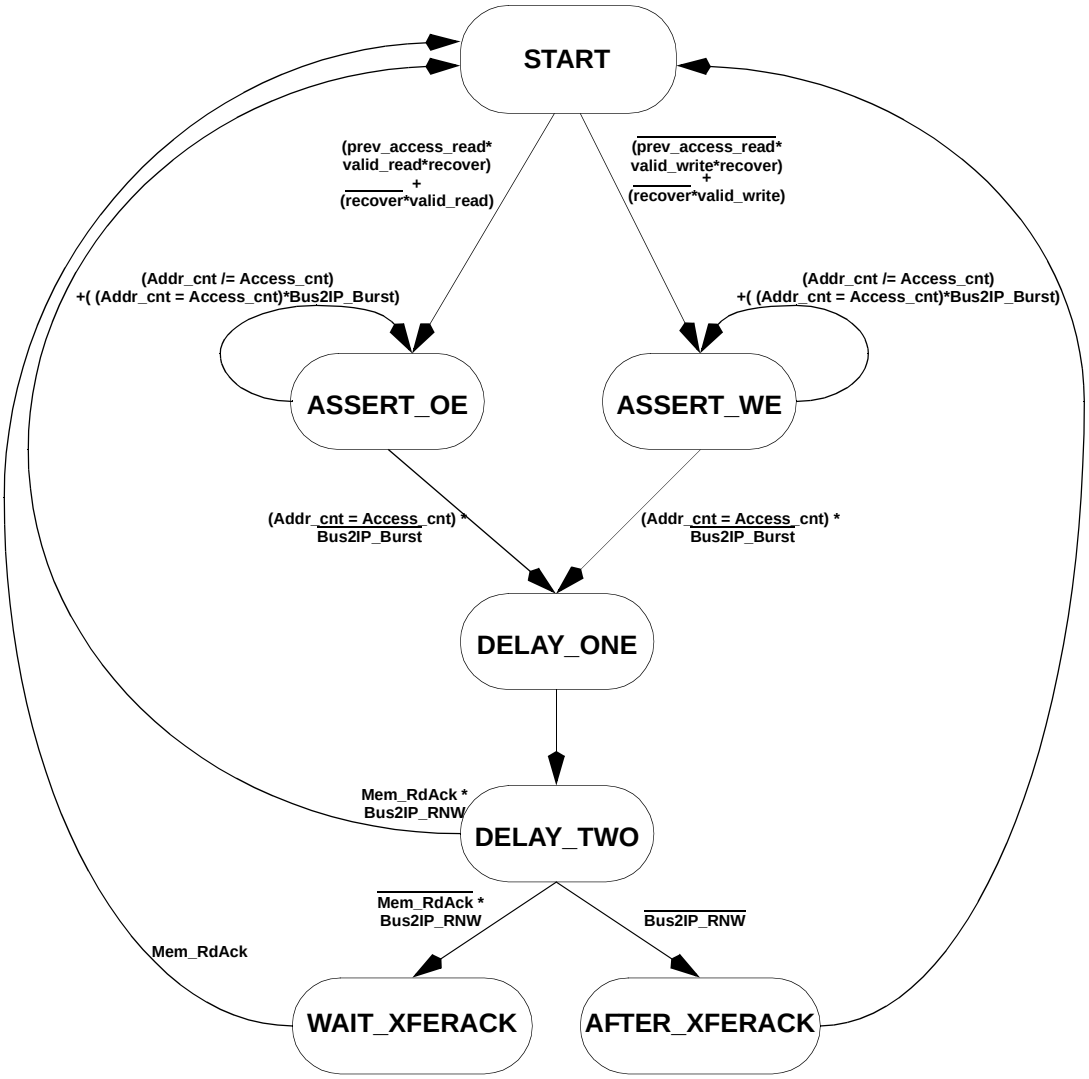


Figure 7: **EMC Memory Control State Machine for Synchronous Memories with 2 clock Pipeline Delay**

Data-Width Matching

Data-width matching is the term used to describe the EMC operation of performing multiple memory accesses to match the width of the memory bank's data bus to the width of the OPBPLB data bus. When C_INCLUDE_DATAWIDTH_MATCHING=1 for a particular memory bank and the width of the memory bank is less than the width of the OPB data bus, multiple accesses are made to the memory bank so that the OPB data bus is fully utilized. An address counter within the EMC is used to provide the correct memory addresses. The address counter is controlled by the state machine and is incremented as each transaction completes.

For example, if the memory bank being addressed is 8 bits wide, then 4 writes to or reads from that memory will be performed in order that the full 32 bits of the OPB data bus are used. For a write cycle, the first byte of the OPBdata bus (bits 0 -7) is written to the memory on bits 0-7 of the memory data bus, then the second byte (bits 8-15) on bits 0-7 of the memory bus, and so on. The write data multiplexing is shown in Figure 8 for a 32-bit processor data bus to illustrate the basic concept. This can easily be extended for a 64-bit processor data bus, but is not shown in this document.

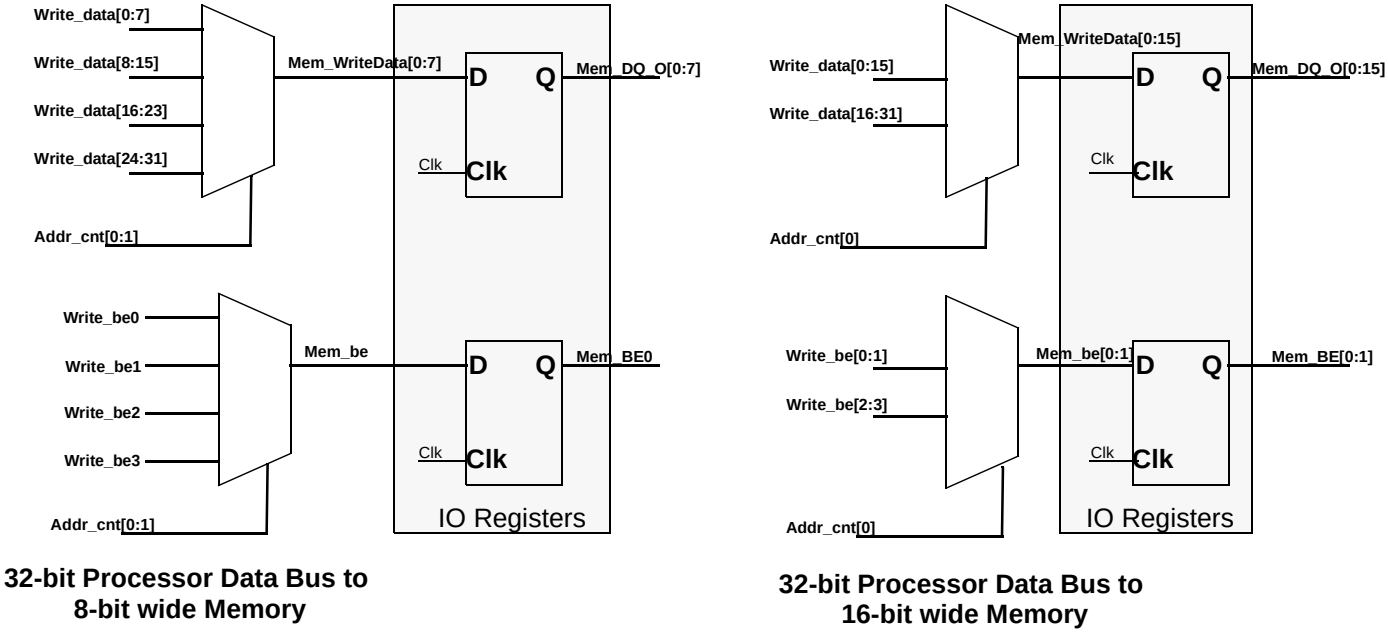


Figure 8: **Data-width matching for Write operations (32-bit Processor Data bus)**

For a read cycle, the data read on bits 0-7 of the memory bus is stored as bits 0-7 of the OPBdata bus for the first byte, then the data read on bits 0-7 of the memory bus is stored as bits 8-15 OPBdata bus for the second byte and so on until all bytes have been read. Then all bytes are presented to the OPBdata bus and the read acknowledge is generated. This is shown in Figure 9 for a 32-bit wide processor data bus to illustrate the basic concept. This can easily be extended for a 64-bit processor data bus, but is not shown in this document. The signal Mem_width_bytes specifies the data-width of the memory bank currently being accessed.

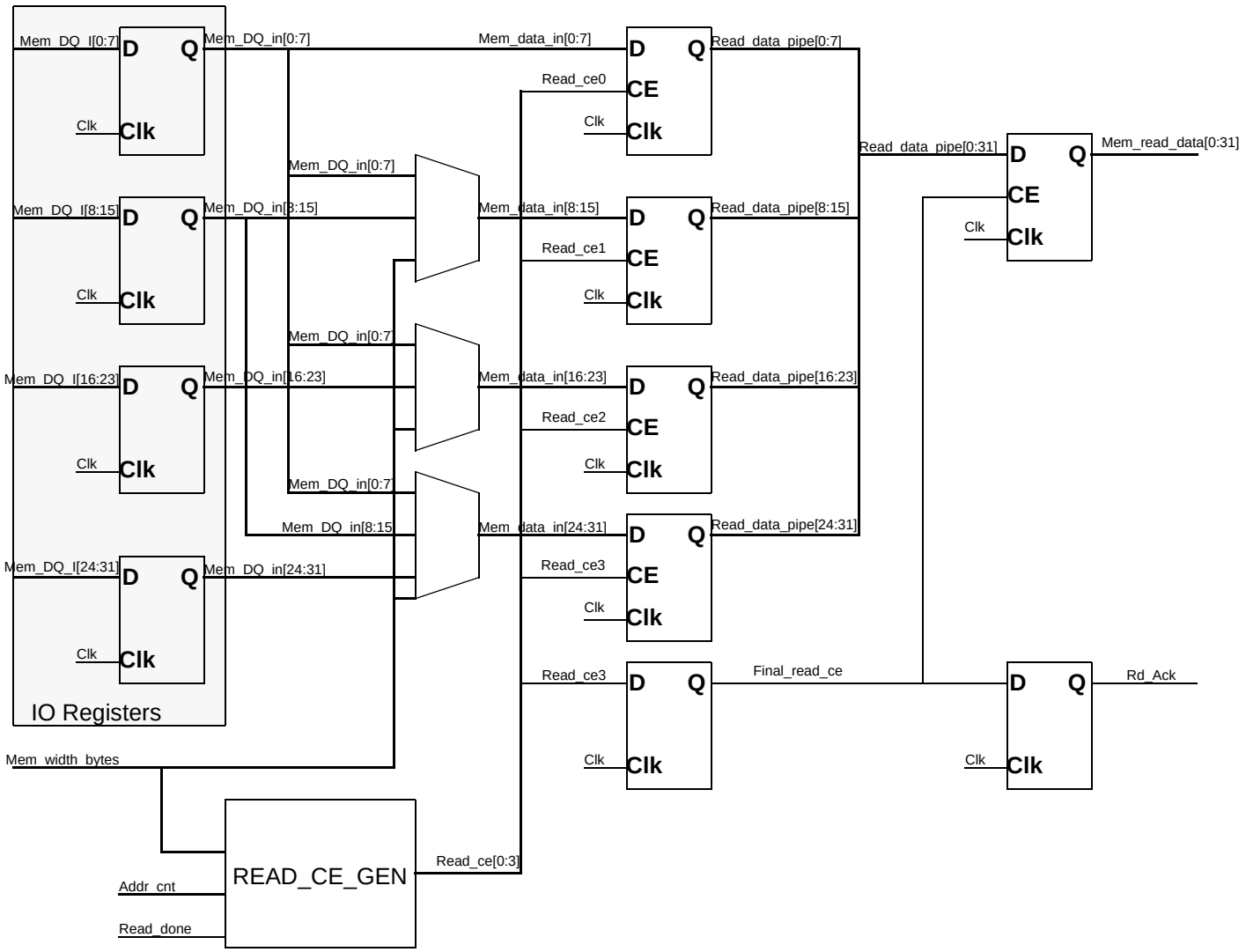


Figure 9: **Data-width matching for Read operations (32-bit data bus)**

The READ_CE signals are generated according to Table 10 for a 32-bit processor data bus to illustrate the basic logic. This can easily be extended for a 64-bit processor data bus, but is not shown in this document. :

Table 10: **READ_CE Generation (32-bit processor data bus)**

Mem_width_bytes	Addr_cnt	Read_ce(0)	Read_ce(1)	Read_ce(2)	Read_ce(3)
8	00	Read_done	0	0	0
	01	0	Read_done	0	0
	10	0	0	Read_done	0
	11	0	0	0	Read_done
16	0	Read_done	Read_done	0	0
	1	0	0	Read_done	Read_done
32	0	Read_done	Read_done	Read_done	Read_done

Since data-width matching writes to memories by performing multiple write cycles to the memory, each time writing a portion of the OPBdata, data-width matching does not work properly when writing to FLASH memories of widths smaller than the OPBdata bus. This is due to the fact that a write to memory location must be preceded by a write of the WRITE command to that address. In the case of data-width matching, multiple WRITE commands would be written to multiple addresses, followed by the writing of the actual data which violates the algorithm of writing to FLASH memory locations. Therefore, C_INCLUDE_DATAWIDTH_MATCHING should be set to 0 for memory banks containing FLASH devices.

The data bus to/from the memory banks uses big-endian bit labeling (i.e. bit 0 is MSB) and is sized according to the C_MAX_MEM_WIDTH parameter. Memories that have smaller widths should connect to this bus starting at bit 0 (MSB - big endian bit labeling). For example, if 3 memory banks are present of sizes 8, 16, and 32 bits, the 8 bit wide memory bank should connect to bits 0-7 of the memory data bus, the 16 bit wide memory bank should connect to bits 0 - 15 of the memory data bus, and the 32 bit wide memory should connect to bits 0 - 31 of the memory data bus.

IO Registers

Registers are used on all signals to and from the memory banks to provide consistent timing on the memory interface. The IO registers present in the design depend upon the setting of the C_INCLUDE_NEGEDGE_IOREGS. All signals output to the memory banks 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 10 and can be used at lower clock frequencies to provide setup and hold to synchronous memories. This parameter can be used for asynchronous memories as well or if there is a mixture of synchronous and asynchronous memories in the system. However, if there are only asynchronous memories in the system, setting this parameter to 1 provides no value, and is therefore recommended to be set to 0.

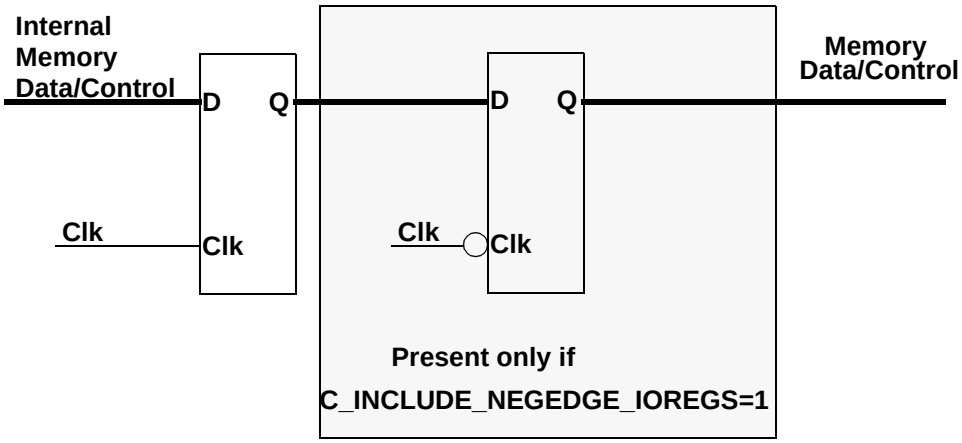


Figure 10: **EMC Output Registers**

If C_INCLUDE_NEGEDGE_IOREGS = 1, the data input from the memories is clocked into the FPGA on the falling edge of the clock. This can be used at lower clock frequencies to obtain setup and hold from synchronous memories in the system. If data width matching is included for any of the memory banks, the data is again registered on the rising edge of the clock before going into the data width matching multiplexors. These registers are not included if data-width matching is not needed for any of the memory banks. This is shown in Figure 11 and can be used at lower clock frequencies to provide setup and hold from synchronous memories. This parameter can be used for asynchronous memories as well or if there is a mixture of synchronous and asynchronous memories in the system. However, if there are only asynchronous memories in the system, setting this parameter to 1 provides no value, and is therefore recommended to be set to 0.

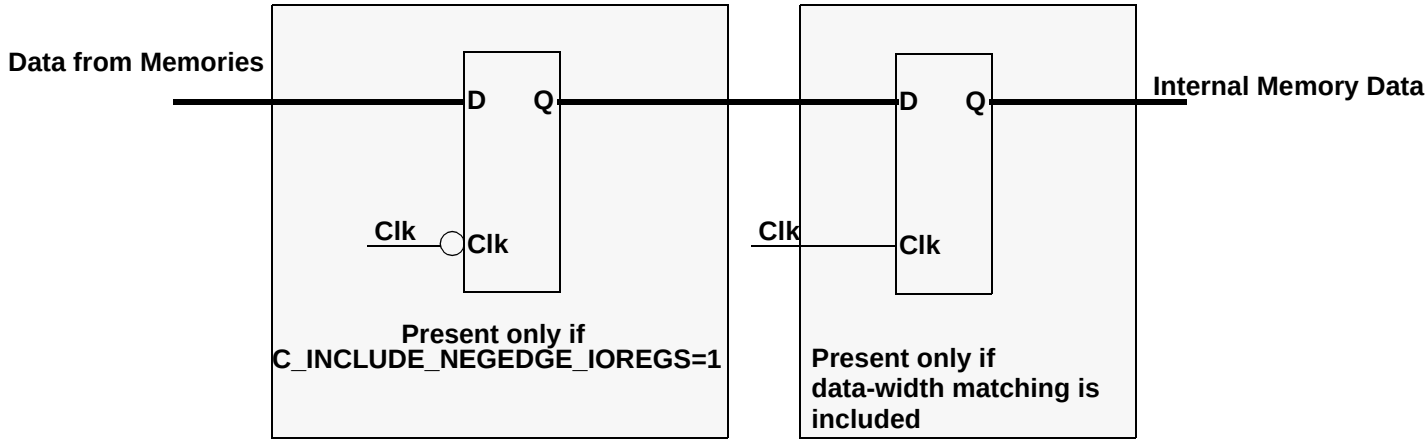


Figure 11: **EMC Input Registers**

NOTE: C_INCLUDE_NEGEDGE_IOREGS should only be set to 1 if the output delay from the FPGA plus the setup requirement of the synchronous memories in the system plus the board route delay is less than half the clock period AND the output delay from the synchronous memories in the system plus the FPGA setup requirement plus the board route delay is less than half the clock period.

- Tfpga_outdelay + Tsetup_memory + Tboard_route_delay < Clock_period/2
- AND
- Tmemory_outdelay + Tfpga_setup + Tboard_route_delay < Clock_period/2

OPB Timing Diagrams

The following timing diagrams show various OPB bus transactions and the resulting memory accesses. Timing diagrams are not shown for all memory widths, but just a sampling to provide an example of how various memory widths are supported.

OPB Accesses to 8-bit SRAM

OPB Single Write followed by Single Read to 8-bit SRAM

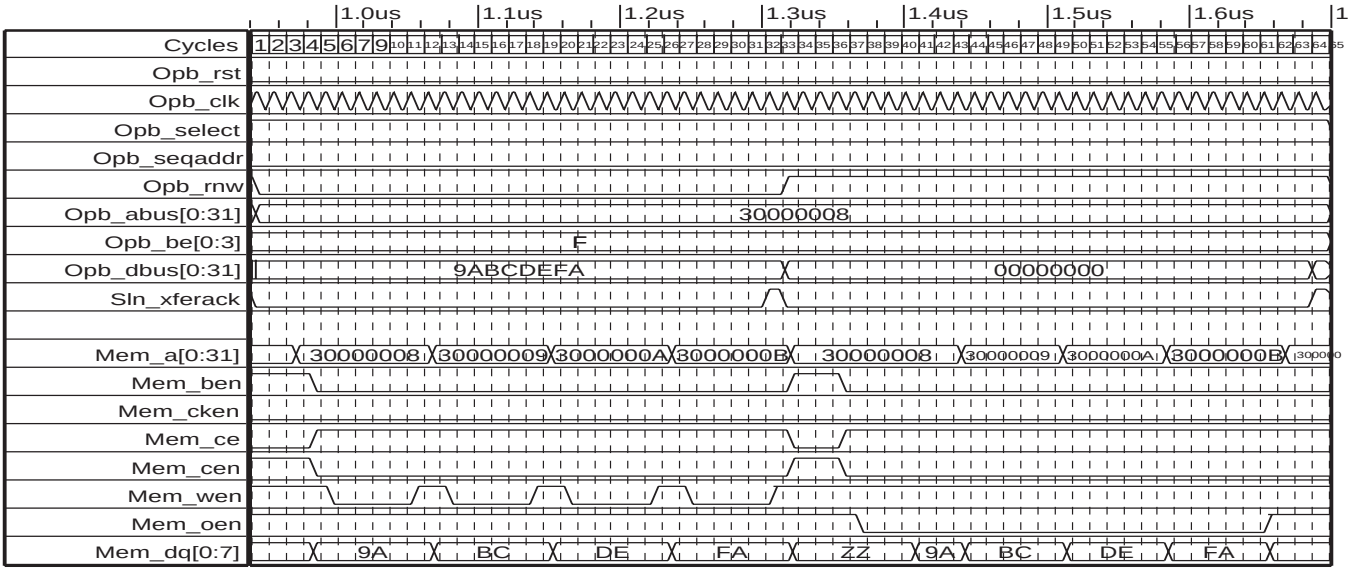


Figure 12: **OPB Single Write/Single Read to 8-bit SRAM**

OPB Accesses to 16-bit SRAM

OPB Single Write followed by Single Read to 16-bit SRAM

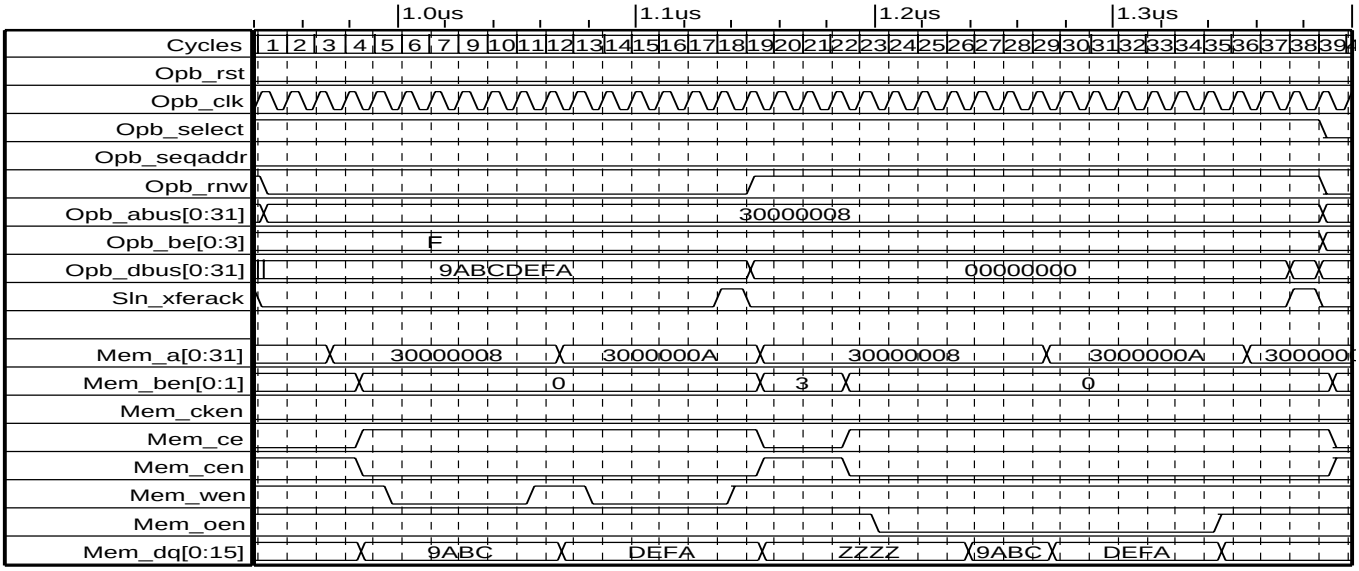


Figure 13: **OPB Single Write/Single Read to 16-bit SRAM**

OPB Accesses to 32-bit SRAM

OPB Single Write followed by Single Read to 32-bit SRAM

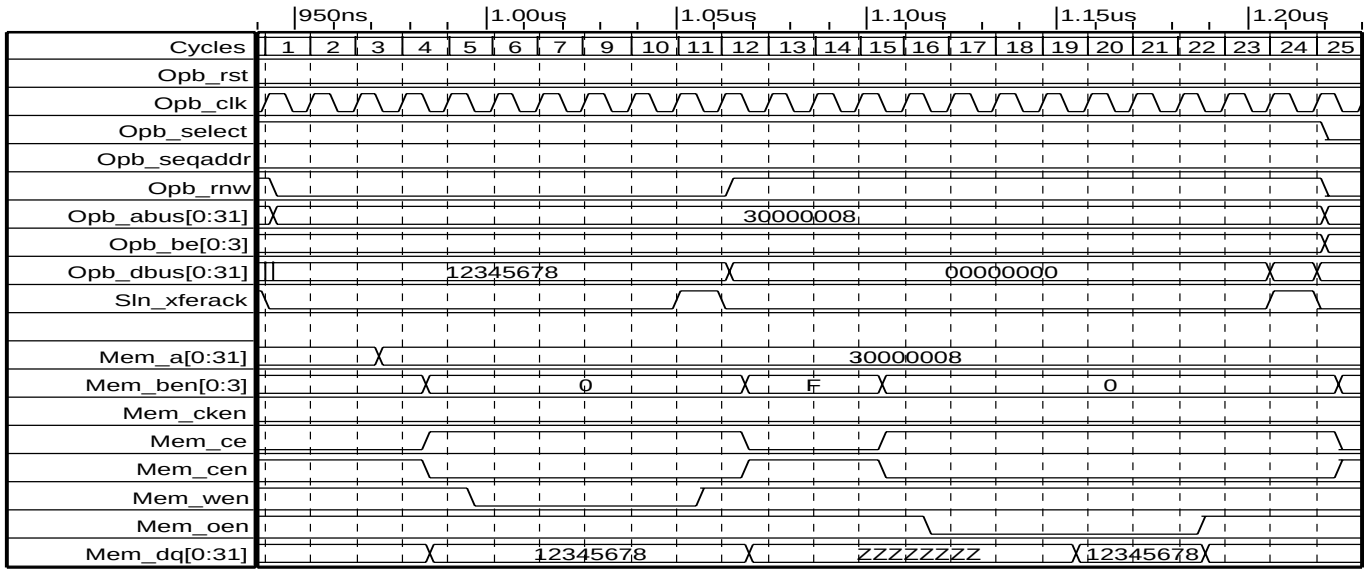


Figure 14: **OPB Single Write/Single Read to 32-bit SRAM**

OPB Accesses to 32-bit Pipeline ZBT

OPB Single Write followed by Single Read to 32-bit Pipeline ZBT

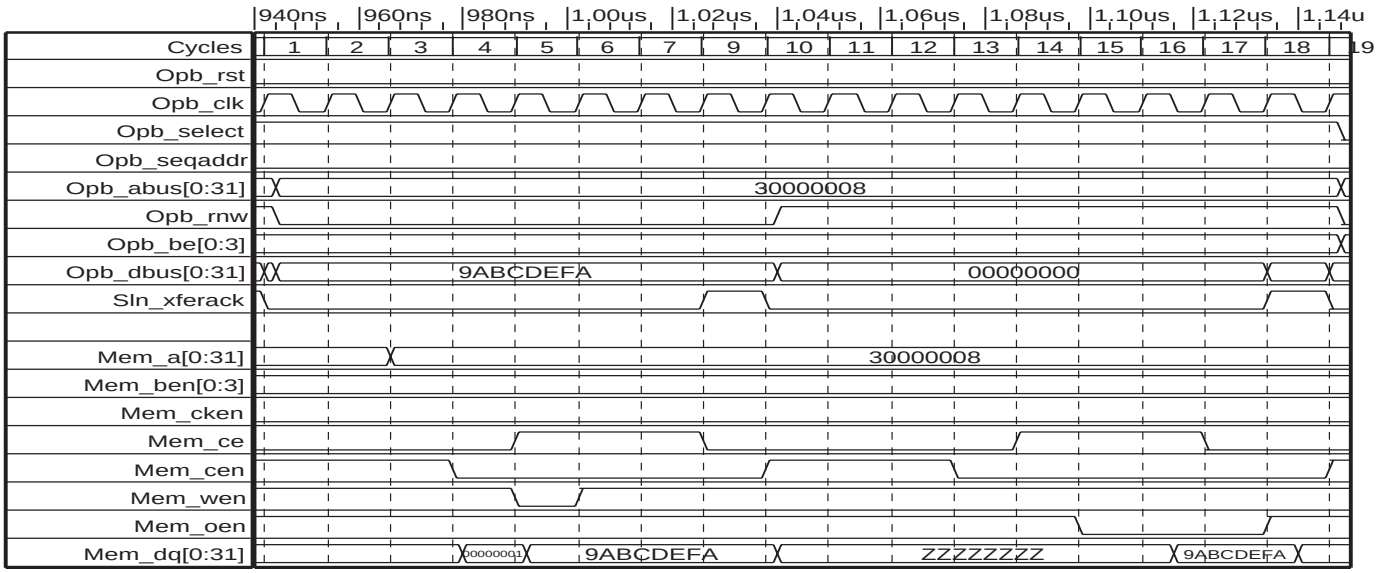


Figure 15: **OPB Single Write/Single Read to 32-bit Pipeline ZBT**

OPB Accesses to 32-bit Flowthrough ZBT

OPB Single Write followed by Single Read to 32-bit Flowthrough ZBT

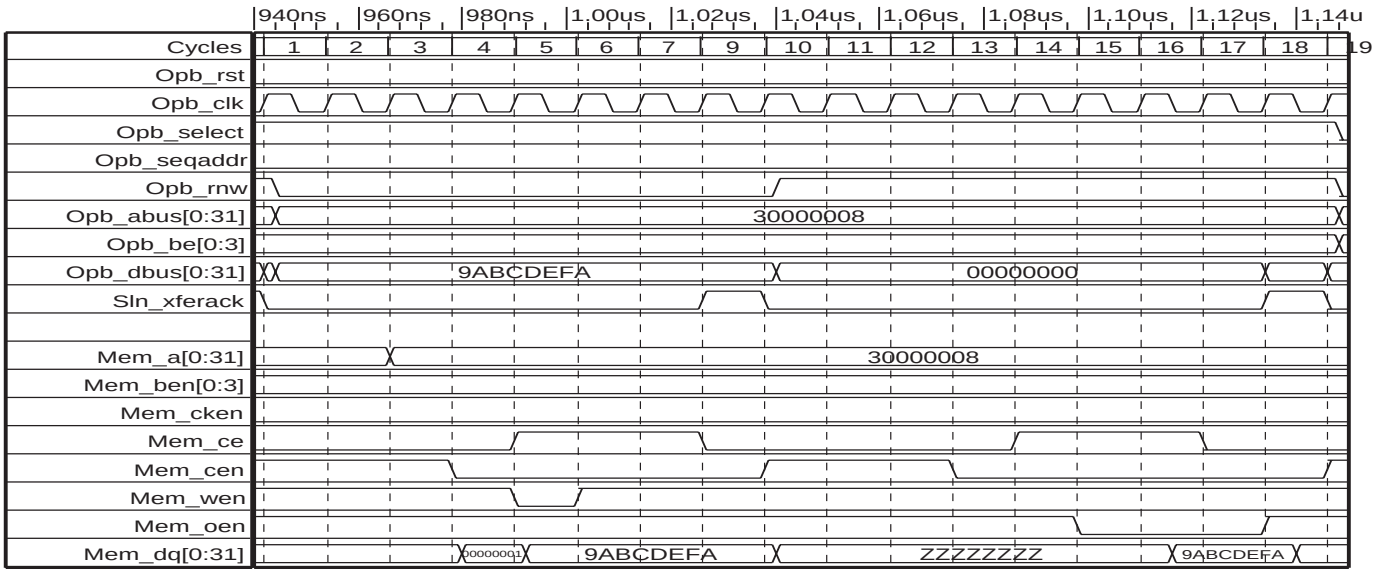


Figure 16: **OPB Single Write/Single Read to 32-bit Flowthrough ZBT**

Basic Timing for Memory

The Memory Controller is designed to connect to a variety of memory subsystem configurations. For detailed descriptions regarding the timing and protocol of the IDT 71V416S SRAM, IDT 71V546 SRAM with ZBT™ Features and the Intel 28F128J3 StrataFlash please refer to their respective data sheets.

However, basic read and write timing diagrams are listed below. Figure 17 and Figure 18 illustrate the basic read and write functions for the SRAM. Table 11 defines the symbols used in the figures for the SRAM.

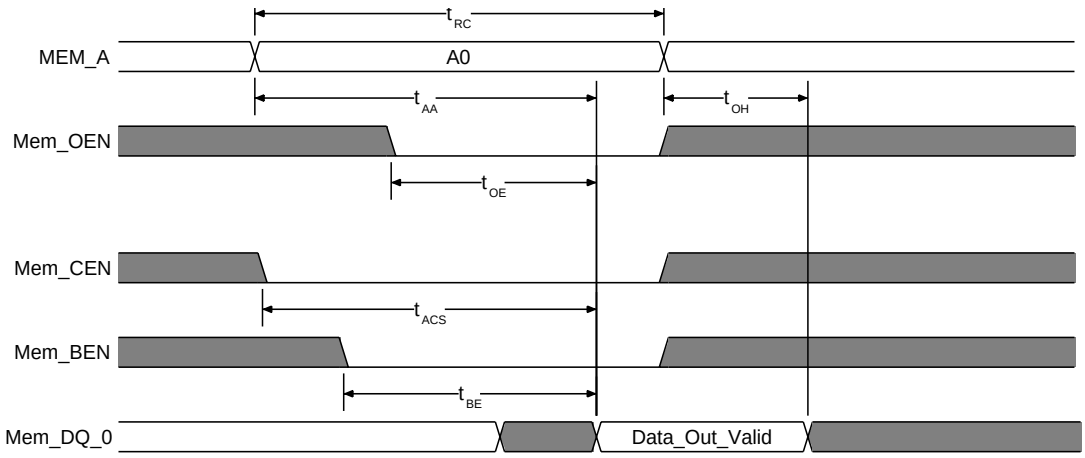


Figure 17: **Timing Waveform for SRAM Read Cycle**

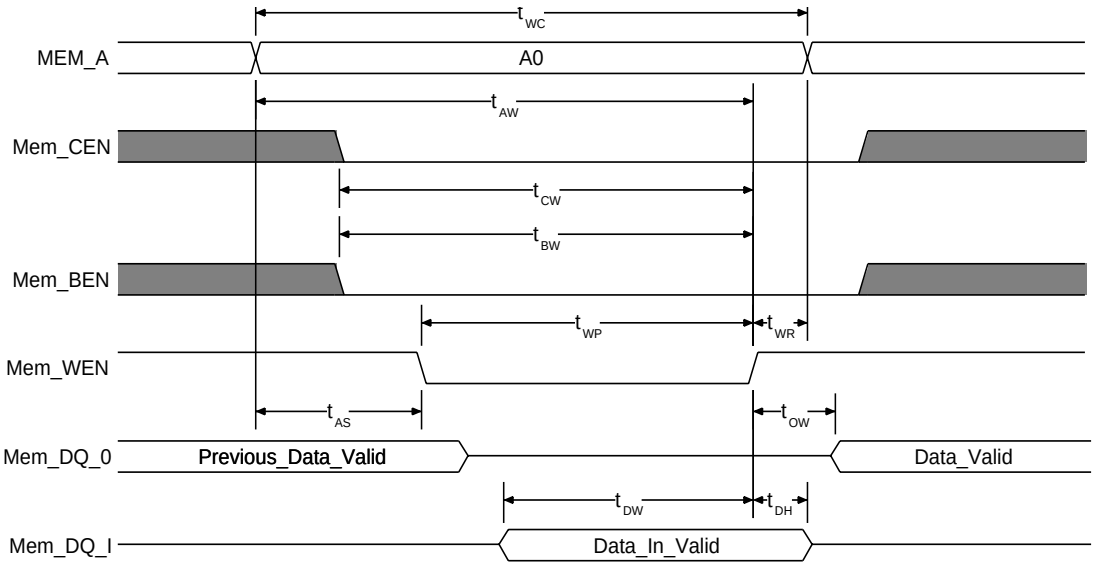


Figure 18: **Timing Waveform for SRAM Write Cycle**

Table 11: **SRAM Parameter Description**

Symbol	Parameter(1)
READ CYCLE	
t_{RC}	Read Cycle Time
t_{AA}	Address Access Time
t_{ACS}	Chip Select Access Time
t_{OE}	Output Enable Low to Output Valid
t_{OH}	Output Hold from Address Change
t_{BE}	Byte Enable Low to Output Valid
WRITE CYCLE	
t_{WC}	Write Cycle Time
t_{AW}	Address Valid to End of Write
t_{CW}	Chip Select Low to End of Write
t_{BW}	Byte Enable Low to End of Write
t_{AS}	Address Set-up Time
t_{WR}	Address Hold from End of Write
t_{WP}	Write Pulse Width
t_{DW}	Data Valid to End of Write
t_{DH}	Data Hold Time
t_{OW}	Write Enable High to Output Low-Z

Notes:
1. Refer to IDT71V416S Data Sheet for specific timing parameters.
2. WEN is HIGH for Read Cycle
3. Address must be valid prior to or coincident with the later CEN, BEN transition LOW; otherwise t_{AA} is the limiting parameter
4. Write Cycle Timing is WEN controlled.

Figure 19 and Figure 20 illustrate the basic read and write functions for the Flash. Table 12 defines the symbols used in the figures for the Flash.

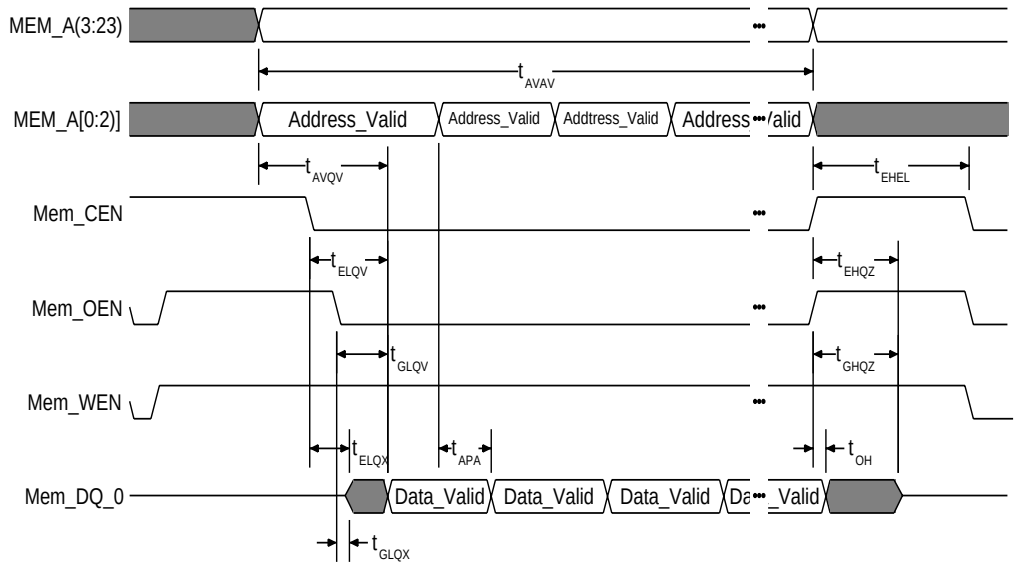


Figure 19: **Waveform for both page-mode and standard word/byte read operation**

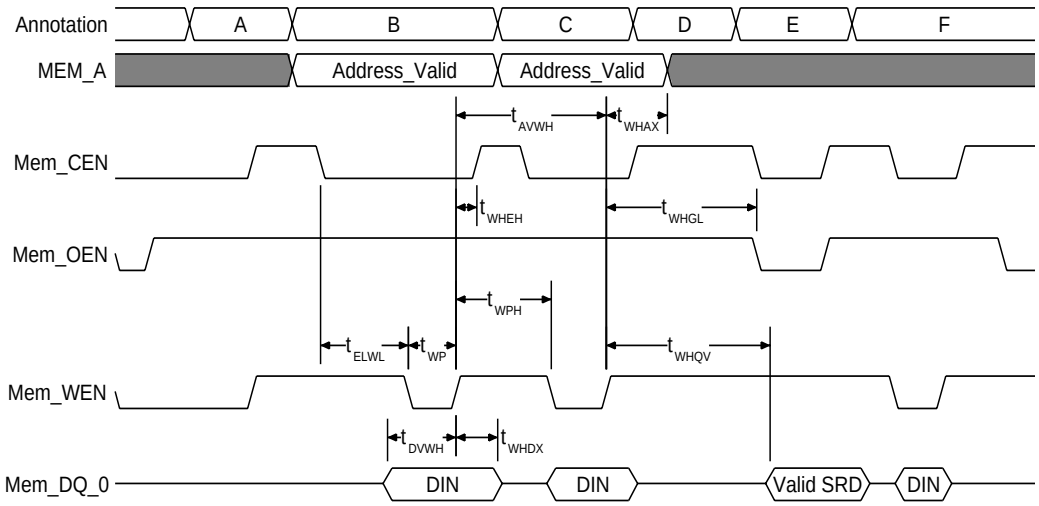


Figure 20: **Waveform for write operations**

Table 12: **StrataFlash Parameter Description**

Symbol	Parameter(1)
READ ONLY	
t_{AVAV}	Read/Write Cycle Time
t_{AVQV}	Address to Output Delay
t_{ELQV}	CEN to Output Delay

Table 12: **StrataFlash Parameter Description *(Continued)***

Symbol	Parameter[1]
t_{GLQV}	OEN to Non-Array Output Delay
t_{ELQX}	CEN to Output Low-Z
t_{GLQX}	OEN to Output Low-Z
t_{EHQZ}	CEN High to Output in High-Z
t_{GHQZ}	OEN High to Output in High-Z
t_{OH}	Output Hold from Address, CEN, or OEN Change, Whichever occurs first
t_{EHEL}	CEN High to CEN Low
t_{APA}	Page Address Access Time
Write Operations	
A	Power-up and standby
B	Write block erase, write buffer, or program set-up
C	Write block erase or write buffer confirm, or valid address and data
D	Automated erase delay
E	Read status register or query data
F	Write read array command
t_{ELWL}	CEN(WEN) Low to WEN(CEN) Going Low
t_{WP}	Write Pulse Width
t_{DVWH}	Data Setup to WEN(CEN) Going High
t_{AVWH}	Address Setup to WEN(CEN) Going High
t_{WHEH}	CEN(WEN) Hold from WEN(CEN) High
t_{WHDX}	Data Hold from WEN(CEN) High
t_{WHAX}	Address Hold from WEN(CEN) High
t_{WPH}	Write Pulse High
t_{WHGL}	Write Recovery before Read

Notes:
1. Refer to Intel 28F128J3A Data Sheet for specific timing parameters.

Figure 21 and Figure 22 illustrate the basic read and write functions for synchronous memories previously stated in this specification. Figure 23 illustrates the basic Chip Enabled operation. Table 13 defines the symbols used in the figures for the synchronous memories referenced in this specification.

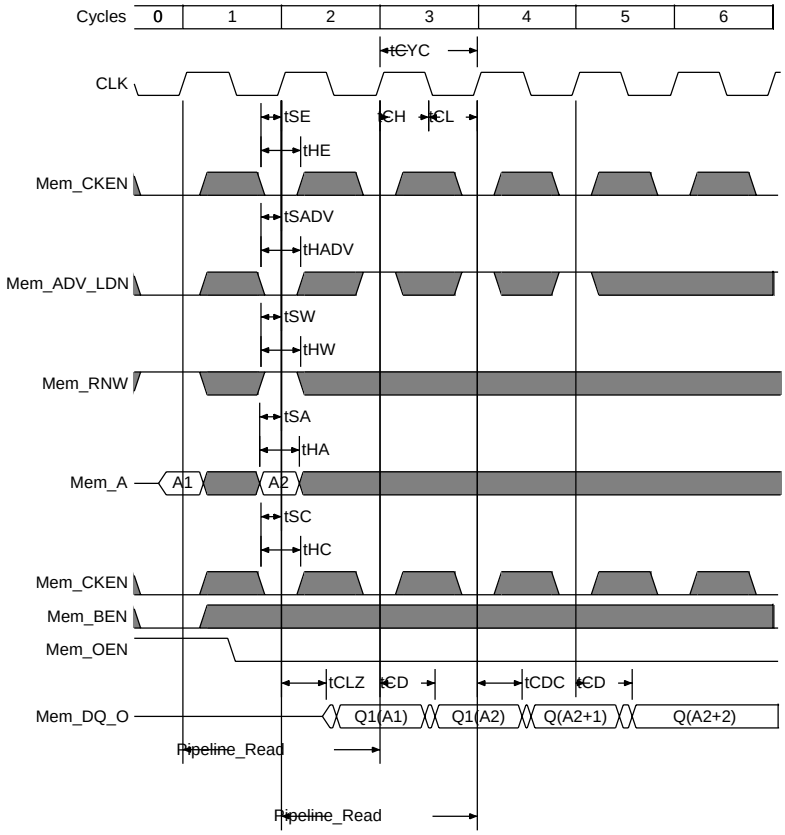


Figure 21: **Timing Waveform for Synchronous Read Cycle**

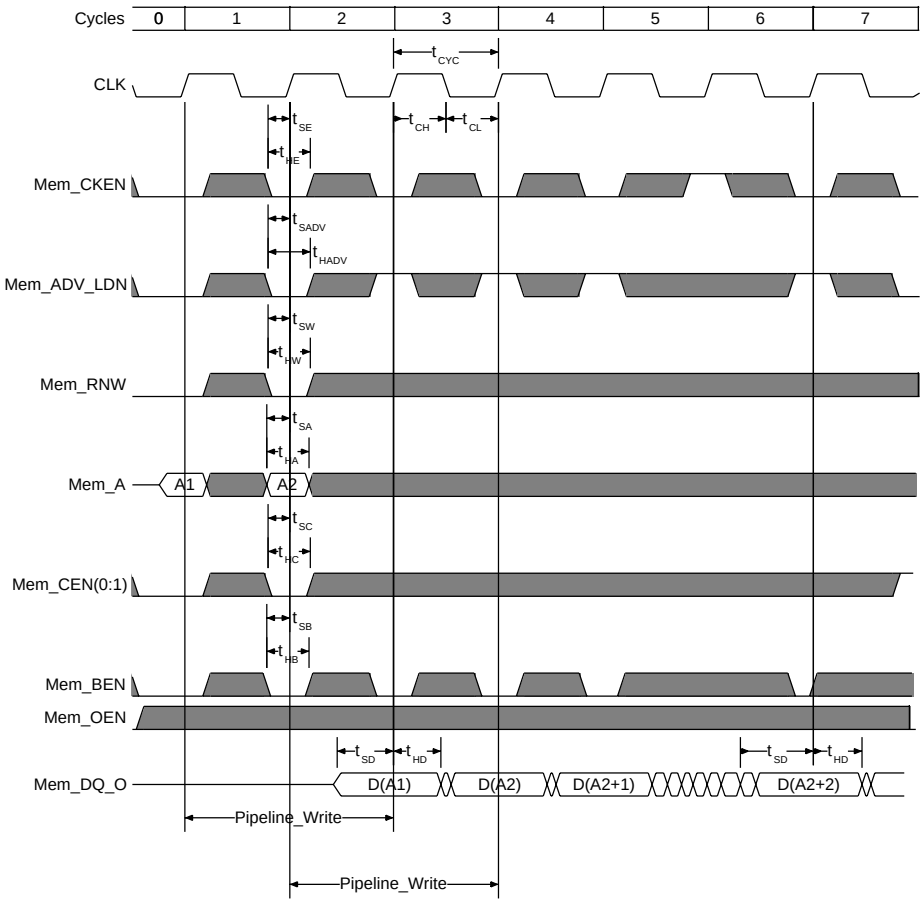


Figure 22: **Timing Diagram for Synchronous Write Cycle**

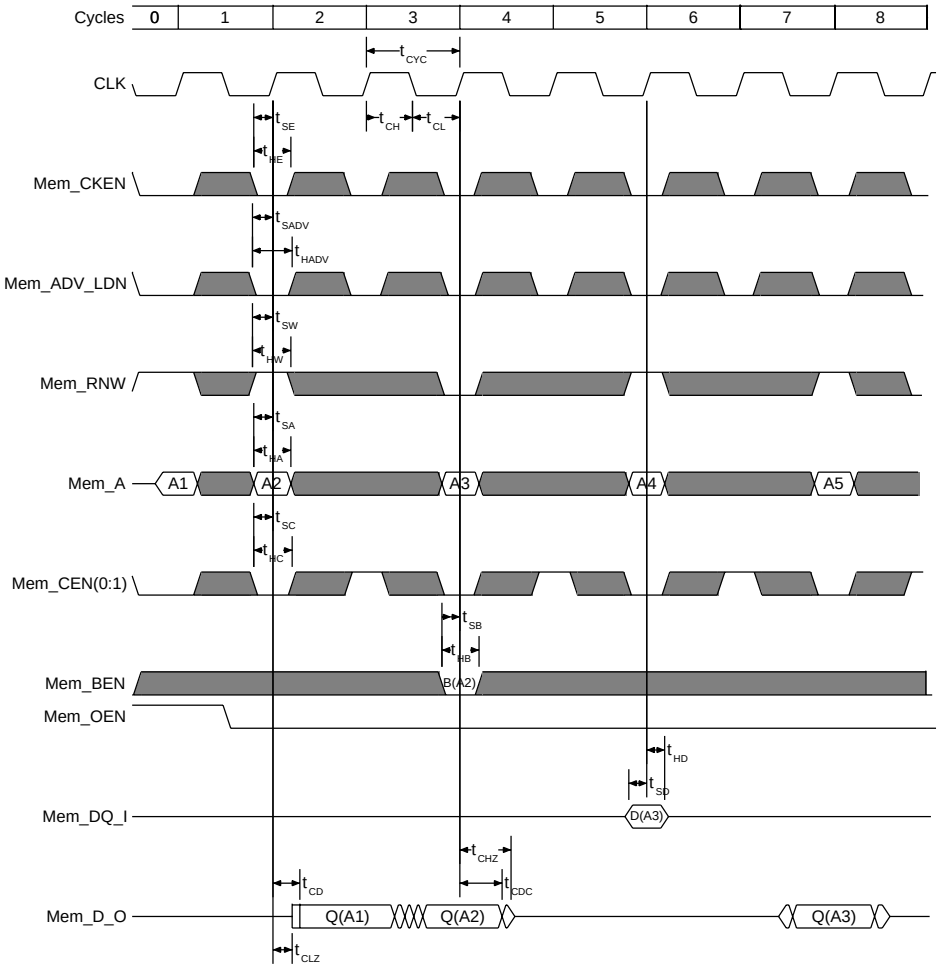


Figure 23: **Timing Waveform for Synchronous Chip Enabled Operation**

Table 13: **Synchronous Parameter Description**

Symbol	Parameter(1)
Clock Parameters	
t_{CYC}	Clock Cycle Time
t_F	Clock Frequency = $1/t_{CYC}$
t_{CH}	Clock High Pulse Width
t_{CL}	Clock Low Pulse Width
Output Parameters	
t_{CD}	Clock High to Valid Data
t_{CDC}	Clock High to Data Change
t_{CLZ}	Clock High to Output Active
t_{CHZ}	Clock High to Data High-Z
t_{OE}	Output Enable Access
t_{OLZ}	Output Enable Low to Data Active
t_{OHZ}	Output Enable High to Data High-Z

Table 13: **Synchronous Parameter Description *(Continued)***

Symbol	Parameter[1]
Setup Parameters	
t_{SE}	Clock Enable Setup
t_{SA}	Address Setup
t_{SD}	Data in Setup
t_{SW}	Read/Write Setup
t_{SADV}	Advance/Load Setup
t_{SC}	Chip Enable/Select Setup
t_{SB}	Byte Write Enable Setup
Hold Parameters	
t_{HE}	Clock Enable Hold
t_{HA}	Address Hold
t_{HD}	Data in Hold
t_{HW}	Read/Write Hold
t_{HADV}	Advance/Load Hold
t_{HC}	Chip Enable/Select Hold
t_{HB}	Byte Write Enable Hold

Notes:
1. Refer to Integrated Device Technology IDT 71V546 Data Sheet for specific timing parameters.

Connecting to Memory

Clocking Synchronous Memories

The EMC does not provide a clock output to any synchronous memories. The OPB clock should be routed through an output buffer to provide the clock to the synchronous memories.

To synchronize the synchronous memory clock to the internal FPGA clock, the FPGA system design should include a DCM external to the EMC core that uses the synchronous memory clock input as the feedback clock as shown in Figure 24. This means that the synchronous clock output from the FPGA must be routed back to the FPGA on a clock pin with a connection to a DCM clock feedback input.

.

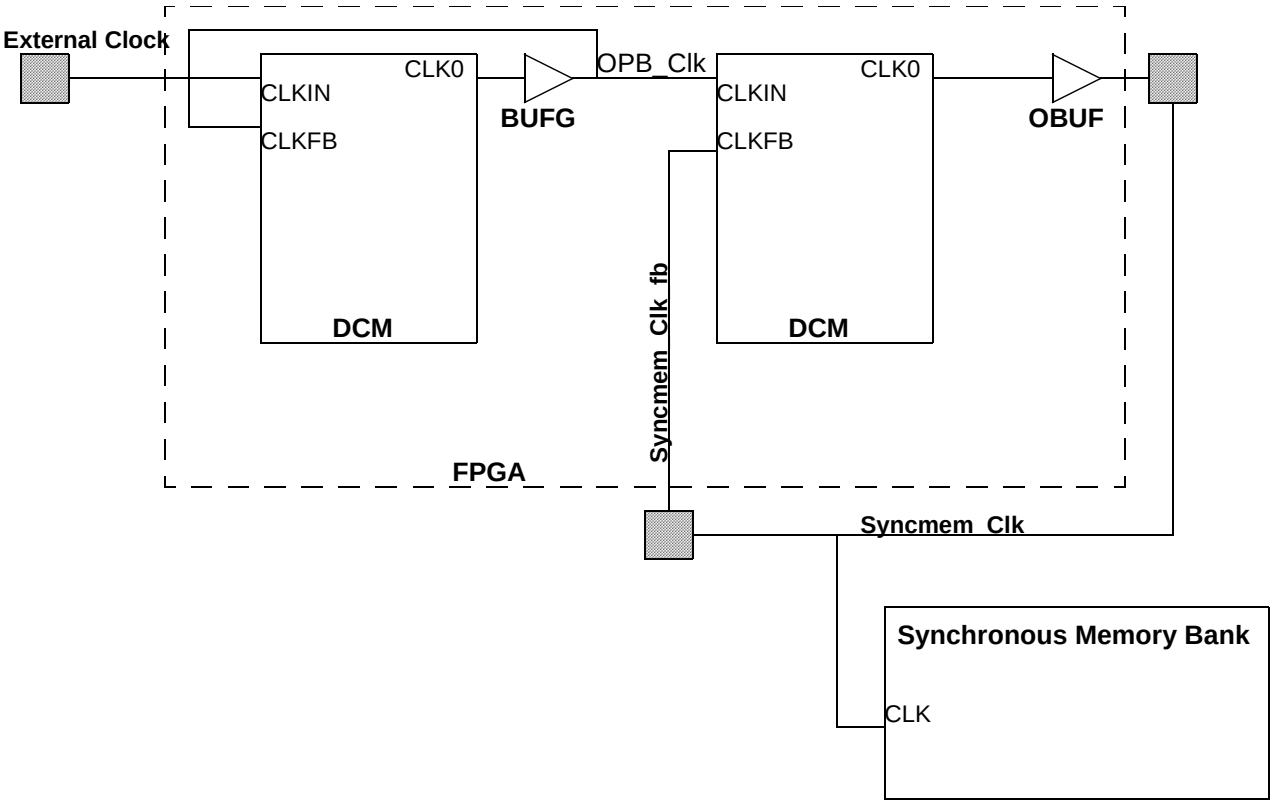


Figure 24: **Synchronous Memory Bank clocked by FPGA Output with feedback**

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 25 (or something similar) or Figure 26 should be included in the FPGA external to the EMC core.

NOTE: If DLLs are used, the designer must reference XAPP132 v2.4, "Using the Virtex Delay-Locked Loop" for the correct DLL implementation

.

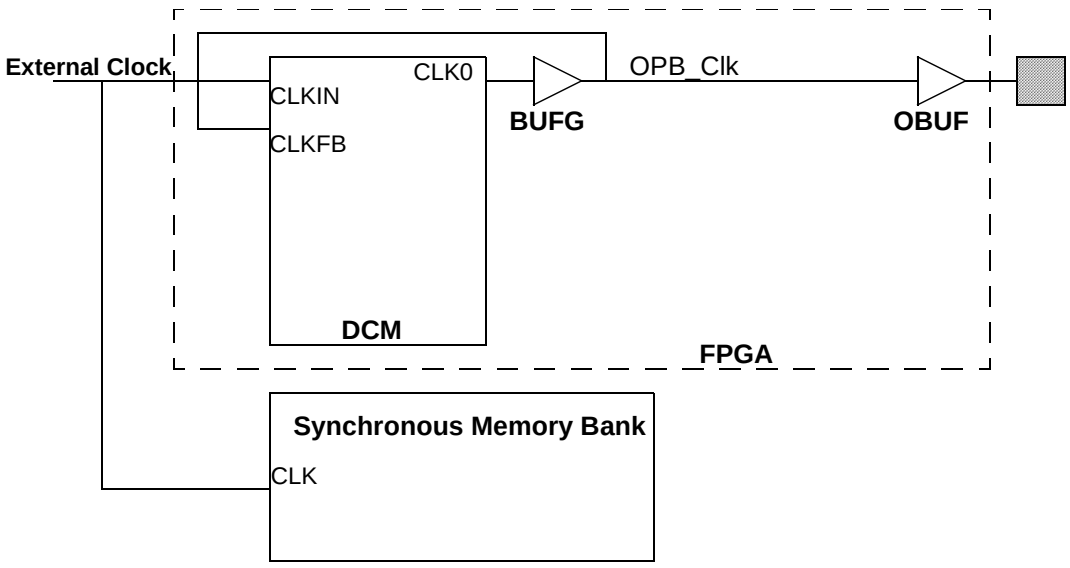


Figure 25: **Synchronous Memory clocked by external clock**

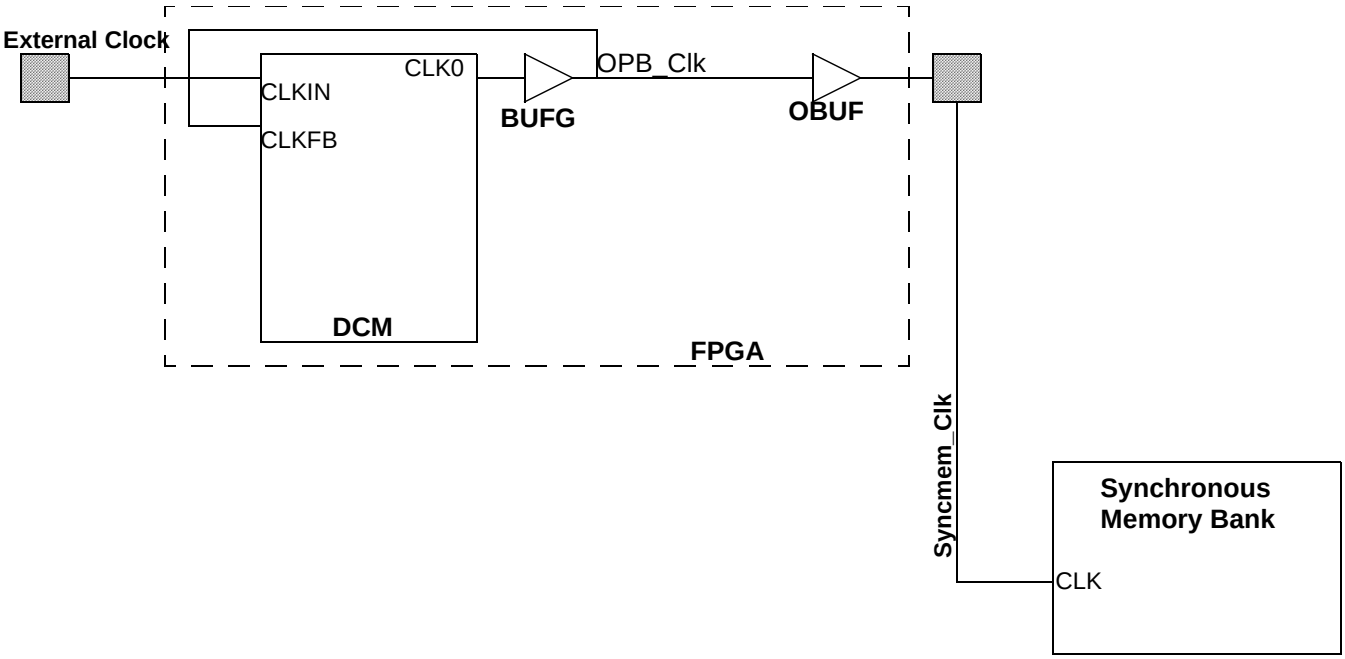


Figure 26: **Synchronous Memory clocked by FPGA Output - no feedback available**

Address Bus, Data Bus, and Control Signal Connections

The three primary considerations for connecting the controller to memory devices are the width of the OPB data bus, the width of the memory subsystem, and the number of memory devices used.

The width of the memory subsystem is simply the maximum width of data that can be read from or written to the memory subsystem. The memory width must be less than or equal to the OPB data bus width.

The data and address signals at the memory controller are labeled with big-endian bit labeling (for example, D(0:31), D(0) is the MSB), whereas most memory devices are either endian agnostic (they can be connected either way) or 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.

Caution must be exercised with 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 14: **Variables used in defining memory subsystem**

Variable	Allowed Range	Definition
BN	0 to 7	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 32	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 32	Width in bits of smallest addressable data word on the memory device
AS	≥ 0	Address shift for address bus = $\log_2$(MW*AU/DW/8)
HAW	1 to 32	Width of OPB address bus in bits

Table 15: **Memory controller to memory interconnect**

Description	EMC Signal (MSB:LSB)	Memory Device Signal (MSB:LSB)
Data bus	Mem_DQ(DN*DW:(DN+1)*DW-1)	D(DW-1:0)
Address bus	Mem_A(HAW-MAW-AS:HAW-AS-1)	A(MAW-1:0)
Chip Enable, low-true	MEM_CEN(BN)	CEN
Output Enable, low-true	MEM_OEN	OEN
Write Enable, low-true	MEM_WEN	WEN (for devices that have byte enables or do not require byte enables)
Byte-Enable-Qualified Write Enable, low-true	MEM_QWEN(INT(DN*DW/8))	WEN (for devices that require byte enables and do not have them)
Byte Enable, low-true	MEM_BEN(INT(DN*DW/8): INT((DN+1)*DW/8-1))	BEN(DW/8-1:0)

Example Memory Connections

Example 1

Example 1: Connection to 32-bit memory using 2 IDT71V416S SRAM parts.

Table 16: **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
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 of host address bus (e.g. OPB or PLB) in bits

Table 17: **Connection to 32-bit memory using 2 IDT71V416S parts**

DN	Description	EMC Signal (MSB:LSB)	Memory Device Signal (MSB:LSB)
0	Data bus	Mem_DQ(0:15)	I/O(15:0)
	Address bus	Mem_A(12:29)	A(17:0)
	Chip Enable, low-true	MEM_CEN(0)	CS
	Output Enable, low-true	MEM_OEN	OE
	Write Enable, low-true	MEM_WEN	WE
	Byte Enable, low-true	MEM_BEN(0:1)	$\overline{BHE}$:$\overline{BLE}$
1	Data bus	Mem_DQ(16:31)	I/O(15:0)
	Address bus	Mem_A(12:29)	A(17:0)
	Chip Enable, low-true	MEM_CEN(0)	CS
	Output Enable, low-true	MEM_OEN	OE
	Write Enable, low-true	MEM_WEN	WE
	Byte Enable, low-true	MEM_BEN(2:3)	$\overline{BHE}$:$\overline{BLE}$

Example 2

Example 2: Connection to 2 banks of 64-bit memory using 4 IDT71V416S SRAM parts per bank.

Table 18: **Variables for two banks of SRAM**

Variable	Value	Definition
BN	0 to 1	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**; device numbers increase toward the least significant bit.
MW	64	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	3	Address shift for address bus = $\log_2$(MW*AU/DW/8)
HAW	32	Width of host address bus (e.g. OPB or PLB) in bits

Table 19: **Connection to 64-bit memory using 8 IDT71V416S parts**

BN	DN	Description	EMC Signal (MSB:LSB)	Memory Device Signal (MSB:LSB)
0	0	Data bus	Mem_DQ(0:15)	I/O(15:0)
		Address bus	Mem_A(11:28)	A(17:0)
		Chip Enable, low-true	MEM_CEN(0)	CS
		Output Enable, low-true	MEM_OEN	OE
		Write Enable, low-true	MEM_WEN	WE
		Byte Enable, low-true	MEM_BEN(0:1)	$\overline{BHE}$:$\overline{BLE}$
	1	Data bus	Mem_DQ(16:31)	I/O(15:0)
		Address bus	Mem_A(11:28)	A(17:0)
		Chip Enable, low-true	MEM_CEN(0)	CS
		Output Enable, low-true	MEM_OEN	OE
		Write Enable, low-true	MEM_WEN	WE
		Byte Enable, low-true	MEM_BEN(2:3)	$\overline{BHE}$:$\overline{BLE}$
	2	Data bus	Mem_DQ(32:47)	I/O(15:0)
		Address bus	Mem_A(11:28)	A(17:0)
		Chip Enable, low-true	MEM_CEN(0)	CS
		Output Enable, low-true	MEM_OEN	OE
		Write Enable, low-true	MEM_WEN	WE
		Byte Enable, low-true	MEM_BEN(4:5)	$\overline{BHE}$:$\overline{BLE}$
	3	Data bus	Mem_DQ(48:63)	I/O(15:0)
		Address bus	Mem_A(11:28)	A(17:0)
		Chip Enable, low-true	MEM_CEN(0)	CS
		Output Enable, low-true	MEM_OEN	OE
		Write Enable, low-true	MEM_WEN	WE
		Byte Enable, low-true	MEM_BEN(6:7)	$\overline{BHE}$:$\overline{BLE}$

Table 19: **Connection to 64-bit memory using 8 IDT71V416S parts *(Continued)***

BN	DN	Description	EMC Signal (MSB:LSB)	Memory Device Signal (MSB:LSB)
1	0	Data bus	Mem_DQ(0:15)	I/O(15:0)
		Address bus	Mem_A(11:28)	A(17:0)
		Chip Enable, low-true	MEM_CEN(1)	CS
		Output Enable, low-true	MEM_OEN	OE
		Write Enable, low-true	MEM_WEN	WE
		Byte Enable, low-true	MEM_BEN(0:1)	$\overline{BHE}$:$\overline{BLE}$
	1	Data bus	Mem_DQ(16:31)	I/O(15:0)
		Address bus	Mem_A(11:28)	A(17:0)
		Chip Enable, low-true	MEM_CEN(1)	CS
		Output Enable, low-true	MEM_OEN	OE
		Write Enable, low-true	MEM_WEN	WE
		Byte Enable, low-true	MEM_BEN(2:3)	$\overline{BHE}$:$\overline{BLE}$
	2	Data bus	Mem_DQ(32:47)	I/O(15:0)
		Address bus	Mem_A(11:28)	A(17:0)
		Chip Enable, low-true	MEM_CEN(1)	CS
		Output Enable, low-true	MEM_OEN	OE
		Write Enable, low-true	MEM_WEN	WE
		Byte Enable, low-true	MEM_BEN(4:5)	$\overline{BHE}$:$\overline{BLE}$
	3	Data bus	Mem_DQ(48:63)	I/O(15:0)
		Address bus	Mem_A(11:28)	A(17:0)
		Chip Enable, low-true	MEM_CEN(1)	CS
		Output Enable, low-true	MEM_OEN	OE
		Write Enable, low-true	MEM_WEN	WE
		Byte Enable, low-true	MEM_BEN(6:7)	$\overline{BHE}$:$\overline{BLE}$

Connecting to Intel StrataFlash

StrataFlash parts contain an identifier register, a status register, and a command interface, so the bit label ordering for these parts is critical to their proper functioning. The tables below show examples of how to connect the big-endian EMC buses to the little-endian StrataFlash parts.

The proper connection ordering is also indicated in a more general form in Table 15. 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 3

Example 3: Connection to 32-bit memory using 2 StrataFlash parts in x16 mode (supports byte read, but no byte write; smallest data type that can be written is 16-bit data).

Table 20: **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**; 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 of host address bus (e.g. OPB or PLB) in bits

Table 21: **Connection to 32-bit memory using 2 StrataFlash parts**

DN	Description	EMC Signal (MSB:LSB)	StrataFlash Signal (MSB:LSB)
0	Data bus	Mem_DQ(0:15)	DQ(15:0)
	Address bus	Mem_A(7:30)	A(23:0)
	Chip Enable, low-true	GND,GND,MEM_CEN(0)	CE(2:0)
	Output Enable, low-true	MEM_OEN	OE#
	Write Enable, low-true	MEM_QWEN(0)	WE#
	Reset/Power down, low-true	MEM_RPN	RP#
	Byte mode select, low-true	N/A - tie to VCC	BYTE#
	Program enable, high-true	N/A - tie to VCC	V_{PEN}
1	Data bus	Mem_DQ(16:31)	DQ(15:0)
	Address bus	Mem_A(7:30)	A(23:0)
	Chip Enable, low-true	GND,GND,MEM_CEN(0)	CE(2:0)
	Output Enable, low-true	MEM_OEN	OE#
	Write Enable, low-true	MEM_QWEN(2)	WE#
	Reset/Power down, low-true	MEM_RPN	RP#
	Byte mode select, low-true	N/A - tie to VCC	BYTE#
	Program enable, high-true	N/A - tie to VCC	V_{PEN}

Example 4

Example 4: Connection to 32-bit memory using 4 StrataFlash parts in x8 mode (supports byte reads and writes).

Table 22: **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**; 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 of host address bus (e.g. OPB or PLB) in bits

Table 23: **Connection to 32-bit memory using 4 StrataFlash parts**

DN	Description	EMC Signal (MSB:LSB)	StrataFlash Signal (MSB:LSB)
0	Data bus	Mem_DQ(0:7)	DQ(7:0)(1)
	Address bus	Mem_A(8:29)	A(23:0)
	Chip Enable, low-true	GND,GND,MEM_CEN(0)	CE(2:0)
	Output Enable, low-true	MEM_OEN	OE#
	Write Enable, low-true	MEM_QWEN(0)	WE#
	Reset/Power down, low-true	MEM_RPN	RP#
	Byte mode select, low-true	N/A - tie to GND	BYTE#
	Program enable, high-true	N/A - tie to VCC	V_{PEN}
1	Data bus	Mem_DQ(8:15)	DQ(7:0)(1)
	Address bus	Mem_A(8:29)	A(23:0)
	Chip Enable, low-true	GND,GND,MEM_CEN(0)	CE(2:0)
	Output Enable, low-true	MEM_OEN	OE#
	Write Enable, low-true	MEM_QWEN(1)	WE#
	Reset/Power down, low-true	MEM_RPN	RP#
	Byte mode select, low-true	N/A - tie to GND	BYTE#
	Program enable, high-true	N/A - tie to VCC	V_{PEN}

Table 23: **Connection to 32-bit memory using 4 StrataFlash parts *(Continued)***

DN	Description	EMC Signal (MSB:LSB)	StrataFlash Signal (MSB:LSB)
2	Data bus	Mem_DQ(16:23)	DQ(7:0)(1)
	Address bus	Mem_A(8:29)	A(23:0)
	Chip Enable, low-true	GND,GND,MEM_CEN(0)	CE(2:0)
	Output Enable, low-true	MEM_OEN	OE#
	Write Enable, low-true	MEM_QWEN(2)	WE#
	Reset/Power down, low-true	MEM_RPN	RP#
	Byte mode select, low-true	N/A - tie to GND	BYTE#
	Program enable, high-true	N/A - tie to VCC	V_{PEN}
3	Data bus	Mem_DQ(24:31)	DQ(7:0)(1)
	Address bus	Mem_A(8:29)	A(23:0)
	Chip Enable, low-true	GND,GND,MEM_CEN(0)	CE(2:0)
	Output Enable, low-true	MEM_OEN	OE#
	Write Enable, low-true	MEM_QWEN(3)	WE#
	Reset/Power down, low-true	MEM_RPN	RP#
	Byte mode select, low-true	N/A - tie to GND	BYTE#
	Program enable, high-true	N/A - tie to VCC	V_{PEN}

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

Design Constraints

Timing Constraints

A timing constraint should be placed on the system clock, setting the frequency to meet the bus timing requirements. An example is shown in Figure 27.

```
NET "OPB_Clk" TNM_NET = "OPB_Clk";
TIMESPEC "TS_OPB_Clk" = PERIOD "OPB_Clk" 7 ns HIGH 50 %;
```

Figure 27: **EMC Timing 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 28.

```
NET "MEM_DQ<0>" PULLDOWN;
NET "MEM_DQ<1>" PULLDOWN;
. . . . .
NET "MEM_DQ<31>" PULLDOWN;
```

Figure 28: **EMC Pin Constraints**

Design Implementation

Target Technology

The intended target technology is a Virtex-II Pro FPGA.

Device Utilization and Performance Benchmarks

This section will be updated when the design has been completed. It will contain the resources and timing for various values of the parameters.

Since the EMC is a module that will be used with other design pieces in the FPGA, the utilization and timing numbers reported in this section are just estimates. As the EMC is combined with other pieces of the FPGA design, the utilization of FPGA resources and timing of the EMC design will vary from the results reported here.

The EMC benchmarks are shown in Table 24 for a Virtex-II Pro -7 FPGA.

Table 24: **EMC FPGA Performance and Resource Utilization Benchmarks (Virtex-II Pro -7)**

Parameter Values					Device Resources			f_{MAX}
C_NUM_BANKS_MEM	C_MEM_WIDTH	C_SYNCH_MEM_x	C_INCLUDE_DATAWIDTH_MATCHING	C_INCLUDE_NEGEDGE_IOREGS	Slices	Slice Flip-Flops	4-input LUTs	
1	8	0	0	0	193	243	199	145
1	8	0	1	0	230	318	212	147
1	16	0	0	0	196	262	193	147
1	16	0	1	0	238	330	231	165
1	32	0	0	0	210	296	172	152
1	32	0	1	0	254	363	260	162
1	8	1	0	0	171	231	176	146
1	8	1	1	0	226	351	176	175
1	16	1	0	0	175	263	177	156
1	16	1	1	0	240	366	203	145
1	32	1	0	0	207	330	177	174
1	32	1	1	0	251	397	224	147
1	8	0	0	1	224	299	201	150
1	8	0	1	1	263	375	215	145
1	32	0	1	1	314	471	264	143
1	16	1	0	1	215	334	176	146
1	16	1	1	1	280	439	204	145
1	32	1	0	1	267	437	174	149
2	32	0,0	0,0	0	220	301	181	148
4	32	0,0,0,0	0,0,0,0	0	228	309	185	146
2	32	1,1	0,0	0	210	335	180	156

Table 24: **EMC FPGA Performance and Resource Utilization Benchmarks (Virtex-II Pro -7) *(Continued)***

4	32	1,1,1,1	0,0,0,0	0	218	343	184	172
4	8,16,32,32	0,1,0,1	1,1,0,0	0	250	356	263	171

Notes:
1. These benchmark designs contain only the EMC without any additional logic. Benchmark numbers approach the performance ceiling rather than representing performance under typical user conditions.

Reference Documents

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

- IDT IDT71V416S,IDT71V416L 3.3V CMOS Static Ram 4 Meg Datasheet
- IDT IDT71V546 128K x 36. 3.3V Synchronous SRAM with ZBT^{TM} Feature Datasheet
- Intel 3V StrataFlash Memory, 28F128J3A (x8/x16) Datasheet

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/22/02	1.0	Initial Xilinx release.
06/04/02	1.1	Update for EDK 1.0
07/29/02	1.2	Add XCO parameters for System Generator
12/30/02	1.3	Updated document for OPB EMC core version 1.10a which added data-width matching
01/08/03	1.4	Update for EDK SP3
03/20/03	1.5	Updated document for OPB EMC core version 1.10b which provided a parameter to allow user to choose negative edge or positive edge IO registers.
07/14/03	1.6	Update to new template
07/28/03	1.6.1	Change DS number because of duplications
10/13/03	1.7	Update Table 24 benchmarks.
12/15/03	1.7.1	Remove unused signal.
03/11/04	1.8	Added timing diagrams