



XAPP723 (v1.4) October 17, 2007

## DDR2 Controller (267 MHz and Above) Using Virtex-4 Devices

Author: Karthi Palanisamy

### Summary

DDR2 SDRAM devices offer new features that go beyond the DDR SDRAM specification and enable the DDR2 device to operate at data rates of 666 Mb/s. High data rates require higher performance from the controller and the I/Os in the FPGA. To achieve the desired bandwidth, it is essential for the controller to operate synchronously with the operating speed of the memory.

### Introduction

This application note describes a DDR2 controller implementation in a Virtex™-4 device interfacing to a Micron DDR2 SDRAM device. For performance levels of 267 MHz and above, use the controller design outlined in this application note with the read data capture technique explained in application note [XAPP721](#), *High-Performance DDR2 SDRAM Interface Data Capture Using ISERDES and OSERDES*.

This application note provides a brief overview of DDR2 SDRAM device features, followed by a detailed explanation of the controller operation when interfacing to high-speed DDR2 memories. It also explains the backend user interface to the controller.

### DDR2 SDRAM Overview

DDR2 SDRAM devices are the next generation devices in the DDR SDRAM family. DDR2 SDRAM devices use the SSTL 1.8V I/O standard. The following section explains the features available in the DDR2 SDRAM devices and the key differences between DDR SDRAM and DDR2 SDRAM devices.

DDR2 SDRAM devices use a DDR architecture to achieve high-speed operation. The memory operates using a differential clock provided by the controller. Commands are registered at every positive edge of the clock. A bidirectional data strobe (DQS) is transmitted along with the data for use in data capture at the receiver. DQS is a strobe transmitted by the DDR2 SDRAM device during Reads and by the controller during Writes. DQS is edge aligned with data for Reads and center aligned with data for Writes.

Read and write accesses to the DDR2 SDRAM device are burst oriented. Accesses begin with the registration of an Active command, which is then followed by a Read or Write command. The address bits registered with the Active command are used to select the bank and row to be accessed. The address bits registered with the Read or Write command are used to select the bank and the starting column location for the burst access.

The DDR2 controller reference design includes a user backend interface to generate the Write address, Write data, and Read addresses. This information is stored in three backend FIFOs for address and data synchronization between the backend and controller modules. Based on the availability of addresses in the address FIFO, the controller issues the correct commands to the memory, taking into account the timing requirements of the memory. The implementation details of the logic blocks are explained in the following sections.

### DDR2 SDRAM Commands Issued by the Controller

[Table 1](#) lists and describes the commands issued by the controller. The commands are detected by the memory using the following control signals: Row Address Select ( $\overline{RAS}$ ),

© 2005–2007 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. PowerPC is a trademark of IBM Inc. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Column Address Select ( $\overline{\text{CAS}}$ ), and Write Enable ( $\overline{\text{WE}}$ ) signals. Clock Enable (CKE) is held High after device configuration, and Chip select ( $\overline{\text{CS}}$ ) is held Low throughout device operation. The [Mode Register Definition](#) section describes the DDR2 command functions supported in the controller.

Table 1: DDR2 Commands

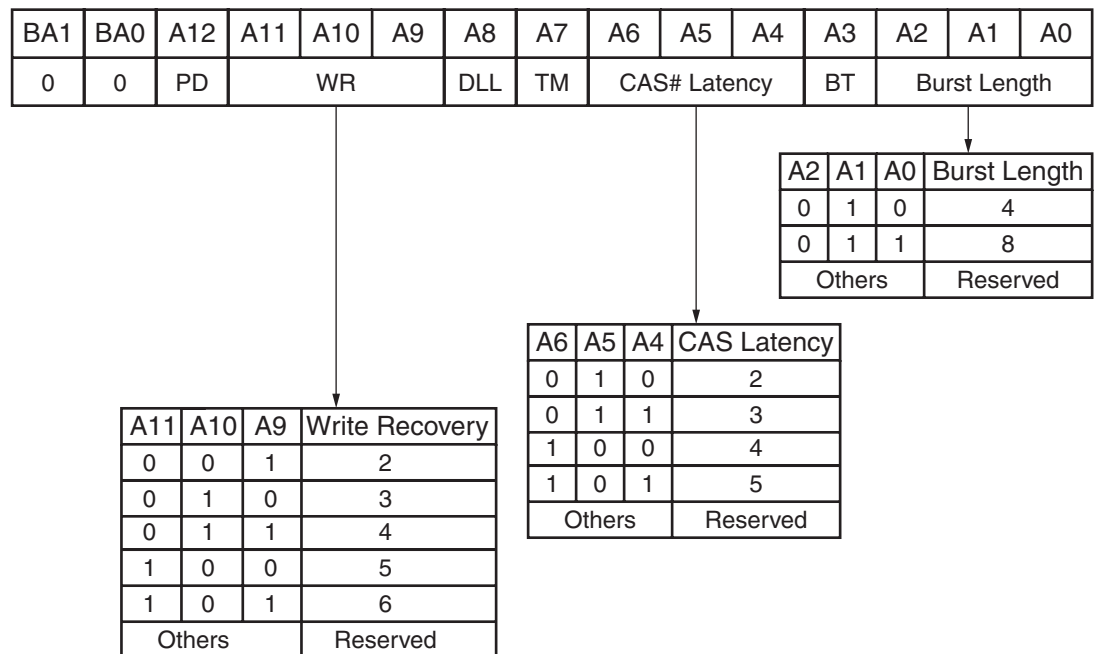
Step	Function	$\overline{\text{RAS}}$	$\overline{\text{CAS}}$	$\overline{\text{WE}}$
1	Load Mode	L	L	L
2	Auto Refresh	L	L	H
3	Precharge (1)	L	H	L
4	Bank Activate	L	H	H
5	Write	H	L	L
6	Read	H	L	H
7	No Operation/IDLE	H	H	H

**Notes:**

- Address signal A10 is held High during *Precharge All Banks* and is held Low during single bank precharge.

### Mode Register Definition

The Mode register is used to define the specific mode of operation of the DDR2 SDRAM. This includes the selection of burst length, burst type, CAS latency, and operating mode. [Figure 1](#) shows the Mode register features used by this controller.



x723\_01\_091505

Figure 1: Mode Register

Bank Addresses BA1 and BA0 select the Mode registers. Table 2 lists the Bank Address bit configuration.

Table 2: Bank Address Bit Configuration

BA1	BA0	Mode Register
0	0	Mode Register (MR)
0	1	EMR1
1	0	EMR2
1	1	EMR3

### Extended Mode Register Definition

The extended Mode register (Table 3) controls additional functions beyond those controlled by the Mode register. These additional functions are DLL enable/disable, output drive strength, On-Die Termination (ODT), Posted CAS Additive Latency (AL), off-chip driver impedance calibration (OCD),  $\overline{DQS}$  enable/disable,  $RDQS/\overline{RDQS}$  enable/disable, and OUTPUT disable/enable. Off-chip Driver Calibration (OCD) is not used in this reference design.

Table 3: Extended Mode Register

BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	Out	$\overline{RDQS}$	$\overline{DQS}$	OCD Program			$R_{TT}$	Posted $\overline{CAS}$			$R_{TT}$	ODS	DLL

#### Extended Mode Register 2 (EMR2)

Bank Addresses are set to 10 (BA1 is set High, and BA0 is set Low). The address bits are all set to Low.

#### Extended Mode Register 3 (EMR3)

Bank Address bits are set to 11 (BA1 and BA0 are set High). Address bits are all set Low, as in EMR2.

### Initialization Sequence

The initialization sequence used in the controller state machine follows the DDR2 SDRAM specifications. The voltage requirements of the memory need to be met by the interface. The following is the sequence of commands issued for initialization:

1. After stable power and clock, a NOP or Deselect command is applied for 200  $\mu$ s.
2. CKE is asserted.
3. Precharge All command after 400 ns.
4. EMR (2) command. BA0 is held Low, and BA1 is held High.
5. EMR (3) command. BA0 and BA1 are both held High.
6. EMR command to enable the memory DLL. BA1 and A0 are held Low, and BA0 is held High.
7. Mode Register Set command for DLL reset. To lock the DLL, 200 clock cycles are required.
8. Precharge All command.
9. Two Auto Refresh commands.
10. Mode Register Set command with Low to A8, to initialize device operation.
11. EMR command to enable OCD default by setting bits E7, E8, and E9 to 1.
12. EMR command to enable OCD exit by setting bits E7, E8, and E9 to 0.

After the initialization sequence is complete, the controller issues a dummy write followed by dummy reads to the DDR2 SDRAM memory for the datapath module to select the right number of taps in the Virtex-4 input delay block. The datapath module determines the right number of delay taps required and then asserts the `dp_dly_slct_done` signal to the controller. The controller then moves into the IDLE state.

**Precharge Command**

The Precharge command is used to deactivate the open row in a particular bank. The bank is available for a subsequent row activation a specified time ( $t_{RP}$ ) after the Precharge command is issued. Input A10 determines whether one or all banks are to be precharged.

**Auto Refresh Command**

DDR2 devices need to be refreshed every 7.8  $\mu$ s. The circuit to flag the Auto Refresh commands is built into the controller. The controller uses a system clock divided by 16 output as the refresh counter. When asserted, the `auto_ref` signal flags the need for Auto Refresh commands. The `auto_ref` signal is held High 7.8  $\mu$ s after the previous Auto Refresh command. The controller then issues the Auto Refresh command after it has completed its current burst. Auto Refresh commands are given the highest priority in the design of this controller.

**Active Command**

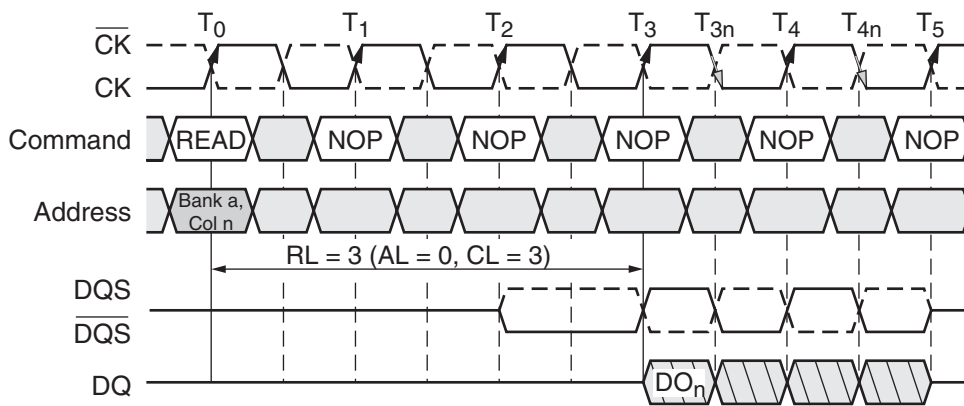
Before any Read or Write commands can be issued to a bank within the DDR2 SDRAM memory, a row in the bank must be activated using an Active command. After a row is opened, Read or Write commands can be issued to the row subject to the  $t_{RCD}$  specification. DDR2 SDRAM devices also support a new feature called posted CAS additive latencies. This feature allows a Read or Write command to be issued prior to the  $t_{RCD}$  specification by delaying the actual registration of the Read or Write command to the internal device using additive latency clock cycles.

When the controller detects a conflict, it issues a Precharge command to deactivate the open row and then issues another Active command to the new row. A conflict occurs when an incoming address refers to a row in a bank other than the currently opened row.

**Read Command**

The Read command is used to initiate a burst read access to an active row. The values on BA0 and BA1 select the bank address, and the address inputs provided on  $A_0 - A_i$  select the starting column location. After the read burst is over, the row is still available for subsequent access until it is precharged.

Figure 2 shows an example of a Read command with an additive latency of zero. Hence, in this example, the Read latency is three, the same as the CAS latency.



x723\_02\_091505

Figure 2: Read Command Example

### Write Command

The Write command is used to initiate a burst access to an active row. The value on BA0 and BA1 select the bank address while the value on address inputs A<sub>0</sub> - A<sub>i</sub> select the starting column location in the active row. DDR2 SDRAMs use a write latency equal to read latency minus one clock cycle.

$$\text{Write Latency} = \text{Read Latency} - 1 = (\text{Additive Latency} + \text{CAS Latency}) - 1$$

Figure 3 shows the case of a Write burst with a Write latency of 2. The time between the Write command and the first rising edge of the DQS signal is determined by the WL.

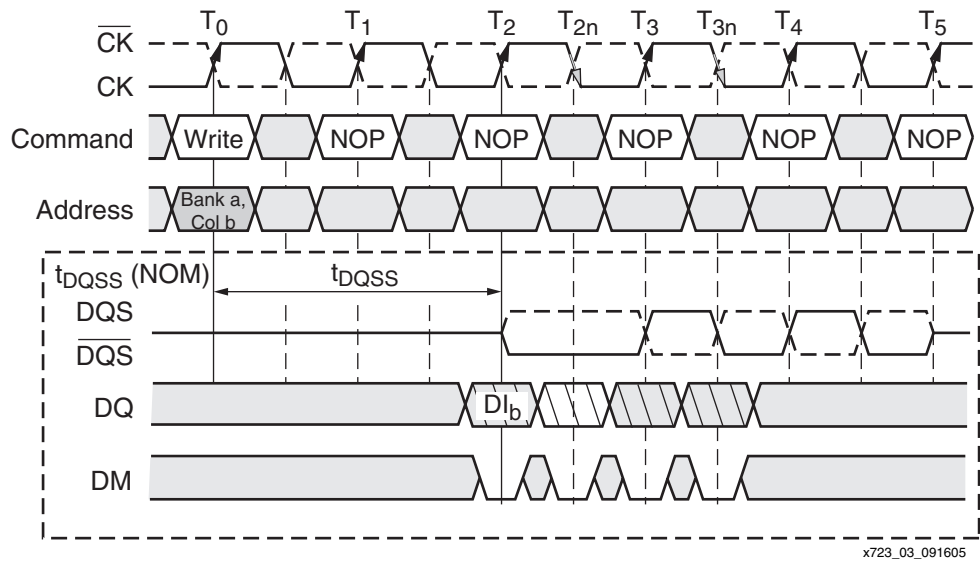


Figure 3: Write Command Example

## DDR2 SDRAM Interface Design

The user interface to the DDR2 controller (Figure 4) and the datapath are clocked at half the frequency of the interface, resulting in improved design margin at frequencies above 267 MHz. The operation of the controller at half the frequency does not affect the throughput or latency. DDR2 SDRAM devices support a minimum burst size of 4, only requiring a command every other clock. The possible burst sizes are as follows:

- For a burst of 4, the controller issues a command every half-frequency clock cycle
- For a burst of 8, the controller issues a command every other half-frequency clock cycle

All the FIFOs in the user interface are asynchronous FIFOs, allowing the user's backend to operate at any frequency. The I/Os toggle at the target frequency.

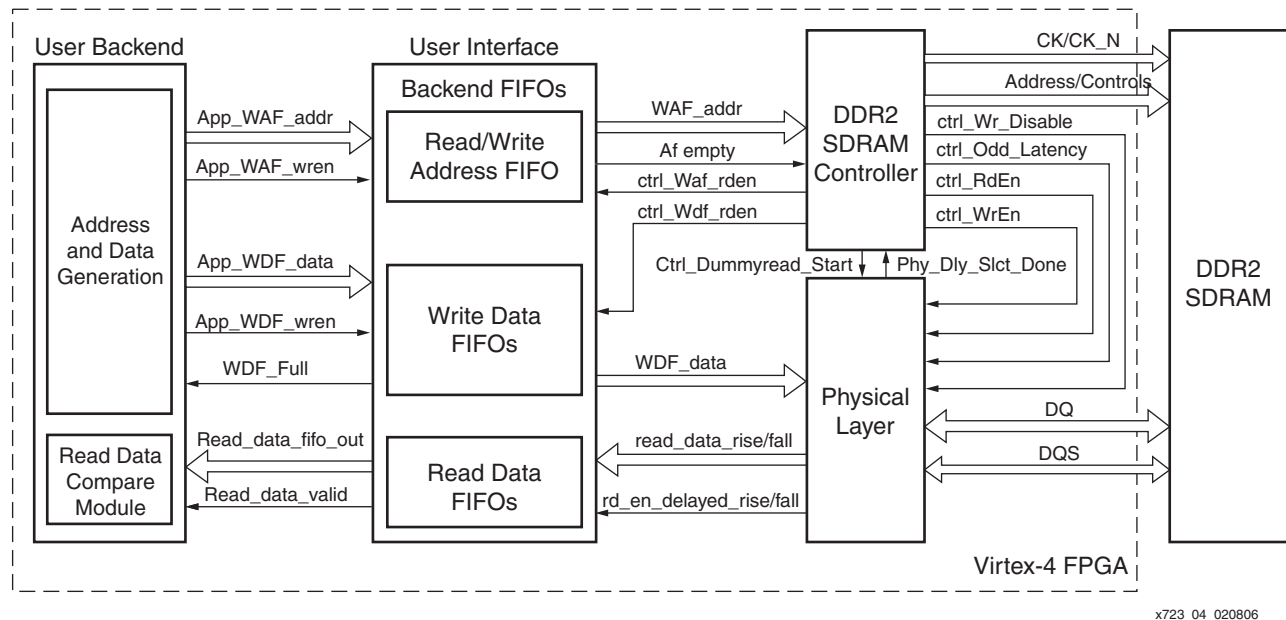


Figure 4: DDR2 Complete Interface Block Diagram

### User Backend

The backend is designed to provide address and data patterns to test all the design aspects of a DDR2 controller. The backend includes the following blocks: backend state machine, read data comparator, and a data generator module. The data generation module generates the various address and data patterns that are written to the memory. The address locations are pre-stored in a block RAM, being used here as a ROM. The address values stored have been selected to test accesses to different rows and banks in the DDR2 SDRAM device. The data pattern generator includes a state machine issuing patterns of data. The backend state machine emulates a user backend. This state machine issues the write or read enable signals to determine the specific FIFO that will be accessed by the data generator module.

### User Interface

The backend user interface has three FIFOs:

- Address FIFO
- Write Data FIFO
- Read Data FIFO

The first two FIFOs are accessed by the user backend modules. The Read Data FIFO is accessed by the Datapath module to store the captured Read data.

## User-to-Controller Interface

Table 4 lists the signals between the user interface and the controller.

Table 4: Signals Between User Interface and Controller

Port Name	Port Width	Port Description	Notes
Af_addr	36	Output of the Address FIFO in the user interface. Mapping of these address bits: <ul style="list-style-type: none"> <li>• Memory Address (CS, Bank, Row, Column) - [31:0]</li> <li>• Reserved - [35]</li> <li>• Dynamic Command Request - [34:32]</li> </ul>	Monitor FIFO-full status flag to write address into the address FIFO.
Af_empty	1	The user interface Address FIFO empty status flag output. The controller processes the address on the output of the FIFO when this signal is deasserted.	FIFO16 Empty Flag.
ctrl_Waf_RdEn	1	Read Enable input to address FIFO in the user interface.	This signal is asserted for one clock cycle when the controller state is Write, Read, Load Mode Register, Precharge All, Auto Refresh, or Active resulting from dynamic command requests.
ctrl_Wdf_RdEn	1	Read Enable input to Write Data FIFO in the user interface.	The controller asserts this signal for one clock cycle after the first write state. This signal is asserted for two clock cycles for a burst length of 8.  Sufficient data must be available in Write Data FIFO associated with a write address for the required burst length before issuing a write command.  For example, for a 64-bit data bus and a burst length of 4, the user should input two 128-bit data words in the Write Data FIFO for every write address before issuing the write command.

Table 5 lists the memory address (Af\_addr), which includes the column address, row address, bank address, and chip-select width for deep memory interfaces.

Table 5: Af\_addr Memory Address

Address	Description
Column Address	[col_ap_width – 1:0]
Row Address	[col_ap_width + row_address – 1:col_ap_width]
Bank Address	[col_ap_width + row_address + bank_address – 1:col_ap_width + row_address]
Chip Select	[col_ap_width + row_address + bank_address + chip_address – 1:col_ap_width + row_address + bank_address]

The address space spanned by Af\_addr is discontinuous. Specifically, bit Af\_addr [10] in the user interface address bus is ignored by the controller logic. In the case in which the memory controller is interfacing to DDR2 devices with only nine column bits, Af\_addr[9] is also ignored. The column address width parameter col\_ap\_width includes an autoprecharge bit (A10) and a column address parameter. The column address parameter indicates the number of column address bits of the selected memory component. Column definitions are as follows:

- For a 9-bit column address, col\_ap\_width is defined as 11. The lower-order nine bits carry the column address, bit A9 is not used, and bit A10 is tied Low during normal reads and writes. As a result, the Auto Precharge function is not supported. The col\_ap\_width parameter is used internally for prepending the A10 bit during the Precharge command.
- For a 10-bit column address, col\_ap\_width is defined as 11.
- For an 11-bit column address, col\_ap\_width is defined as 12.

## Dynamic Command Request

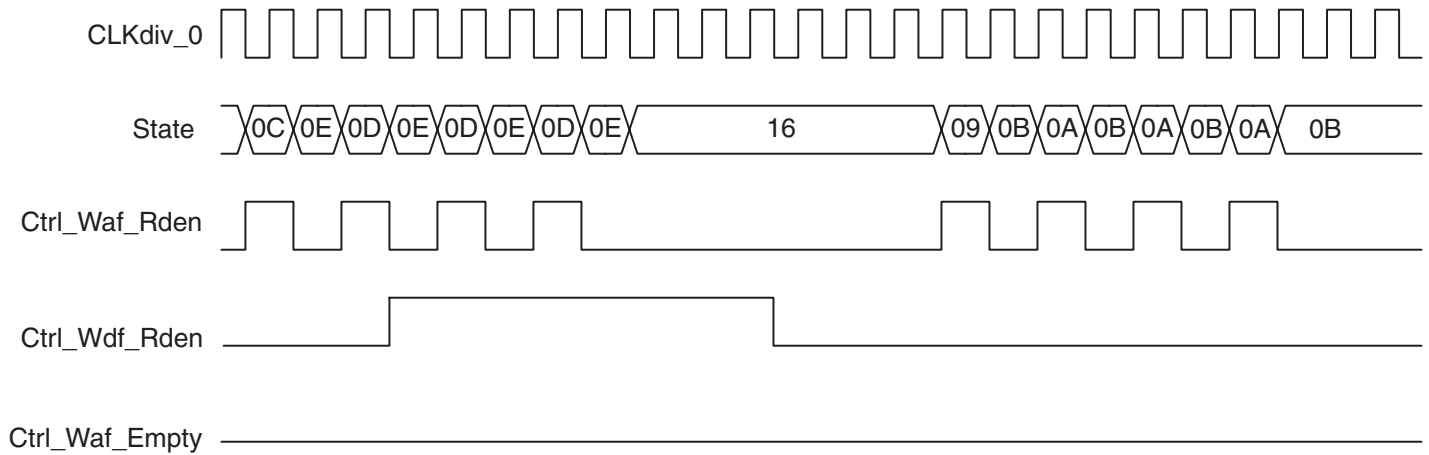
Table 6 lists the optional commands. These commands are not required for normal operation of the controller. The user has the option to request these commands when required by an application.

Table 6: Optional Commands

Command	Description
000	Load Mode Register
001	Auto Refresh
010	Precharge All
011	Active
100	Write
101	Read
110	NOP
111	NOP



Figure 5 describes four consecutive Writes followed by four consecutive Reads with a burst length of 8.



X723\_05\_091905

Figure 5: Consecutive Reads Followed by Consecutive Writes with Burst Length of 8

Table 7 lists the state signal values for Figure 5.

Table 7: Values for State Signals in Figure 5

State	Description
0C	First Write
0E	Write Wait
0D	Burst Write
16	Write Read
09	First Write
0B	Read Wait
0A	Burst Read

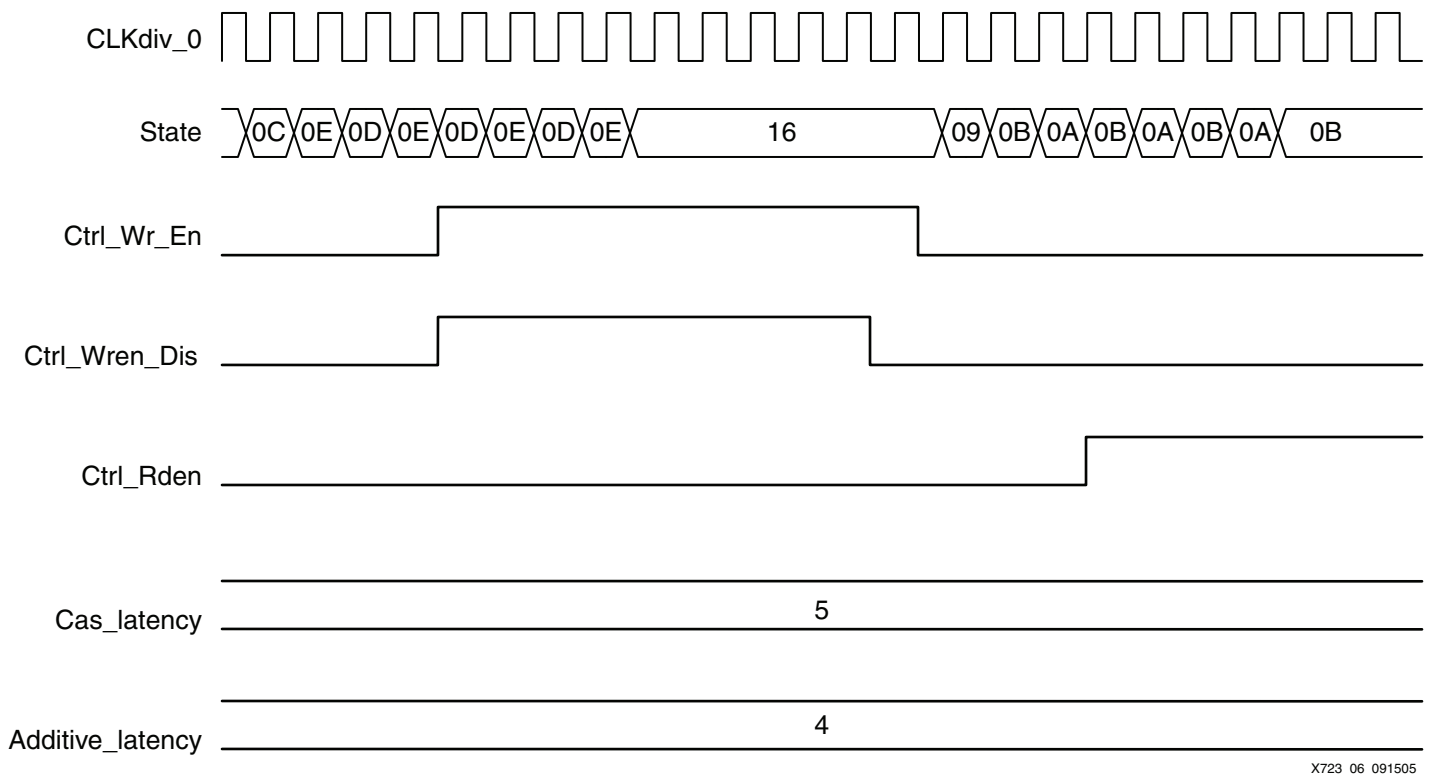
## Controller to Physical Layer Interface

Table 8 lists the signals between the controller and the physical layer.

Table 8: Signals Between the Controller and Physical Layer

Signal Name	Signal Width	Signal Description	Notes
ctrl_WrEn	1	Output from the controller to the write datapath. Write DQS and DQ generation begins when this signal is asserted.	Asserted for two controller clock cycles for a burst length of 4 and three controller clock cycles for a burst length of 8. Asserted one controller clock cycle earlier than the WRITE command for CAS latency values of 4 and 5.
ctrl_wr_disable	1	Output from the controller to the write datapath. Write DQS and DQ generation ends when this signal is deasserted.	Asserted for one controller clock cycle for a burst length of 4 and two controller clock cycles for a burst length of 8. Asserted one controller clock cycle earlier than the WRITE command for CAS latency values of 4 and 5.
ctrl_Odd_Latency	1	Output from the controller to write datapath. Asserted when the selected CAS latency is an odd number. Required for generation of write DQS and DQ after the correct write latency (Write latency = CAS latency – 1).	
ctrl_Dummyread_Start	1	Output from the controller to the read datapath. When this signal is asserted, the strobe and data calibration begin.	This signal must be asserted when valid read data is available on the data bus. This signal is deasserted when the dp_dly_slct_done signal is asserted.
dp_dly_slct_done	1	Output from the read datapath to the controller indicating the strobe and data calibration are complete.	This signal is asserted when the data and strobe have been calibrated. Normal operation begins after this signal is asserted.
ctrl_RdEn	1	Output from the controller to the read datapath for a read-enable signal.	This signal is asserted for one controller clock cycle for a burst length of 4 and two controller clock cycles for a burst length of 8. The CAS latency and additive latency values determine the timing relationship of this signal with the read state.

Figure 6 describes the timing waveform for control signals from the controller to the physical layer.



X723\_06\_091505

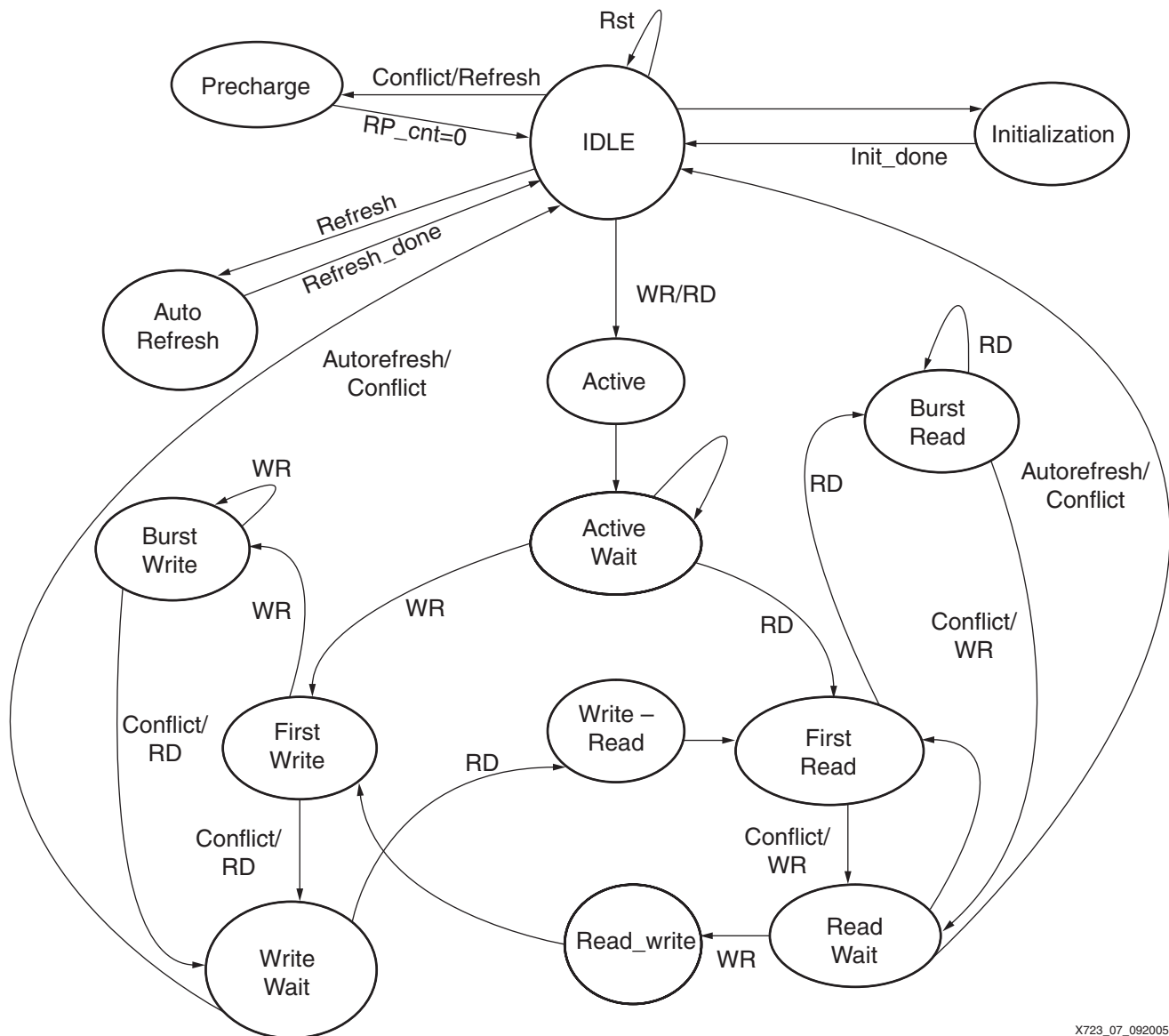
Figure 6: Timing Waveform for Control Signals from the Controller to the Physical Layer

## Controller Implementation

The controller is clocked at the half the frequency of the interfaces. Therefore, the address, bank address, and command signals ( $\overline{\text{RAS}}$ ,  $\overline{\text{CAS}}$ , and  $\overline{\text{WE}}$ ) are asserted for two clock cycles of the fast memory interface clocks. The control signals ( $\overline{\text{CS}}$ ,  $\text{CKE}$ , and  $\text{ODT}$ ) are DDR of the half frequency clocks, ensuring that the control signals are asserted for just one clock cycle of the fast memory interface clock.

The controller state machine issues the commands in the correct sequencing order while determining the timing requirements of the memory.

The following sections explain in detail the various stages of the controller state machine.



X723\_07\_092005

Figure 7: DDR2 Controller State Machine

Figure 7 shows the DDR2 controller state machine. Before the controller issues the commands to the memory:

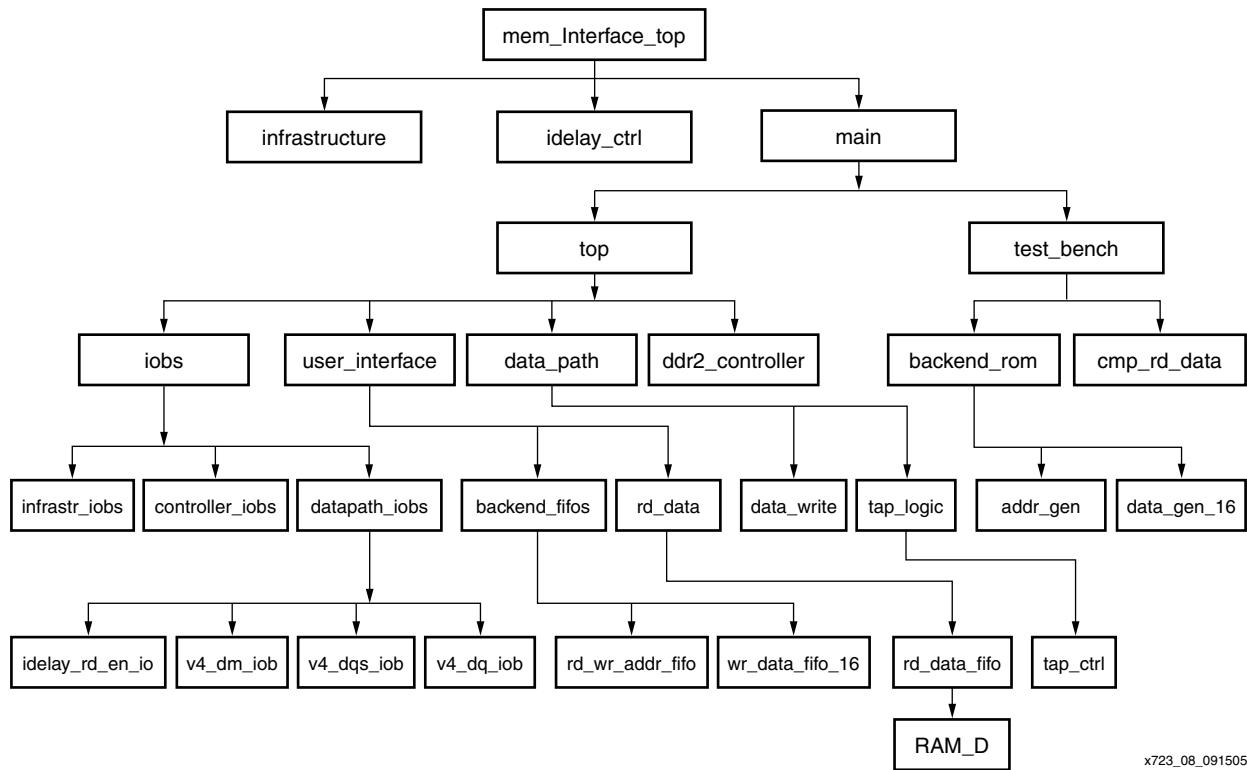
1. The address FIFO is in first-word-fall-through mode (FWFT). In this mode, the first address written into the FIFO appears at the FIFO output. The controller decodes the address.
2. The controller activates a row in the corresponding bank if all banks have been precharged, or it compares the bank and row addresses to the already open row and bank address. If there is a conflict, the controller precharges the open bank and then issues an Active command before moving to the Read/Write states.
3. After arriving in the Write state, if the controller gets a Read command, the controller waits for the write\_to\_read time before issuing the Read command. Similarly, in the Read state, when the controller sees a Write command from the command logic block, the controller waits for the read\_to\_write time before issuing the Write command. In the read or write state, the controller also asserts the read enable to the address FIFO to get the next address.
4. The commands are pipelined to synchronize with the Address signals before being issued to the DDR2 memory.

## Reference Design

Figure 8 shows the design hierarchy, beginning with a top-level module called `mem_interface_top`.

The reference design for the DDR2 SDRAM interface is integrated with the MIG tool. This tool has been integrated with the Xilinx CORE Generator™ software. For the latest version of the design, download the IP Update on the Xilinx website at:

[http://www.xilinx.com/xlnx/xil\\_sw\\_updates\\_home.jsp](http://www.xilinx.com/xlnx/xil_sw_updates_home.jsp).



x723\_08\_091505

Figure 8: Reference Design Hierarchy

## Reference Design Summary

[Table 9](#) lists the maximum frequency by speed grade for a 72-bit interface.

*Table 9: Maximum Frequency for a 72-bit Interface*

Speed Grade	Maximum Frequency by Speed Grade (MHz)
-10	230
-11	267
-12	300

[Table 10](#) lists the reference design summary for a 72-bit interface.

*Table 10: Reference Design Summary for a 72-bit Interface*

Parameters for Design Details	Design Details / Notes
Device Utilization	6714 slices. Includes the controller, synthesizable testbench, the user interface, and the physical layer.
	6 BUFPGs. Includes one BUFPG for the 200 MHz reference clock for the IDELAY block.
	9 BUFIOs. Equals the number of strobes in the interface.
	1 DCM.
	1 PMCD.
	72 ISERDES. Equals the number of data bits in the interface.
	99 OSERDES. Equals the sum of the data bits, strobes, and data mask bits.

## Conclusion

The DDR2 controller described in this application note, along with the data capture method described in application note [XAPP721](#), *High-Performance DDR2 SDRAM Interface Data Capture Using ISERDES and OSERDES*, provide a solution for high-performance memory interfaces.

This design provides a high margin because the logic in the FPGA fabric (excluding the calibration logic) is clocked at half the frequency of the interface, eliminating critical paths. This design was verified in hardware.

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/15/05	1.0	Initial Xilinx release.
12/16/05	1.1	Revised <a href="#">Table 8</a> and <a href="#">Table 10</a> .
02/02/06	1.2	Revised <a href="#">Figure 4</a> .
02/08/06	1.3	Revised <a href="#">Figure 4</a> .

Date	Version	Revision
10/17/07	1.4	<ul style="list-style-type: none"><li>• Revised "Introduction."</li><li>• Revised Table 4.</li><li>• Added a note after Table 5.</li><li>• Retitled old section "Design Hierarchy" to "Reference Design" and changed text.</li><li>• Retitled old section "Reference Design Utilization" to "Reference Design Summary."</li><li>• Added Table 9.</li><li>• Retitled old Table 10 from "Resource Utilization for a 64-Bit Interface" to "Reference Design Summary for a 72-bit Interface." Revised text in Table 10.</li><li>• Revised "Conclusion."</li></ul>