XAPP709 (v2.0) October 27, 2006

# DDR SDRAM Controller Using Virtex-4 FPGA Devices

Author: Rich Chiu


## Summary

This application note describes a DDR SDRAM controller implemented in a Virtex™-4 XC4VLX25 FF668 -10C device. This implementation uses direct clocking for data capture and an automatic calibration circuit to adjust delay on the data lines.

DDR SDRAM devices are low-cost, high-density storage resources that are widely available from many memory vendors. This reference design has been developed using both SDRAM components and DIMMs.


## DDR SDRAM Description

The DDR SDRAM specification details are available from JEDEC organization, part of the Electronic Industries Alliance (EIA), at http://www.jedec.org/. The DDR SDRAM specifications are published in the JEDEC document, under the reference JESD79E.

DDR SDRAM devices are the silicon memory resource most frequently used in systems today, with applications ranging from consumer products to video systems. DDR SDRAM device frequencies range up to 200 MHz or DDR400. DRAM devices are available in component or module configurations.

### DDR Controller Commands

Table 1 presents the commands issued by the controller. These commands are passed to the memory using the following control signals:

- Row Address Select ($\overline{RAS}$)
- Column Address Select ($\overline{CAS}$)
- Write Enable ($\overline{WE}$)
- Clock Enable (CKE) (always held High after device configuration)
- Chip Select ($\overline{CS}$) (always held Low during device operation)

*Table 1:* **DDR SDRAM Commands**

| Signal No. | Function | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ |
|---|---|---|---|---|
| 1 | Load Mode Register | L | L | L |
| 2 | Auto Refresh | L | L | H |
| 3 | Precharge(1) | L | H | L |
| 4 | Select Bank Activate Row | L | H | H |
| 5 | Write Command | H | L | L |
| 6 | Read Command | H | L | H |

*Table 1:* **DDR SDRAM Commands *(Continued)***

| Signal No. | Function | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ |
|---|---|---|---|---|
| 7 | No Operation (NOP) | H | H | H |

**Notes:**

1. Address signal A10 is held High during PRECHARGE ALL BANKS and is held Low during single bank precharge.

## Command Functions

### Mode Register

The Mode register is used to define the specific mode of DDR SDRAM operation, including the selection of burst length, burst type, CAS latency, and operating mode. Figure 1 shows the Mode register features that this controller uses.

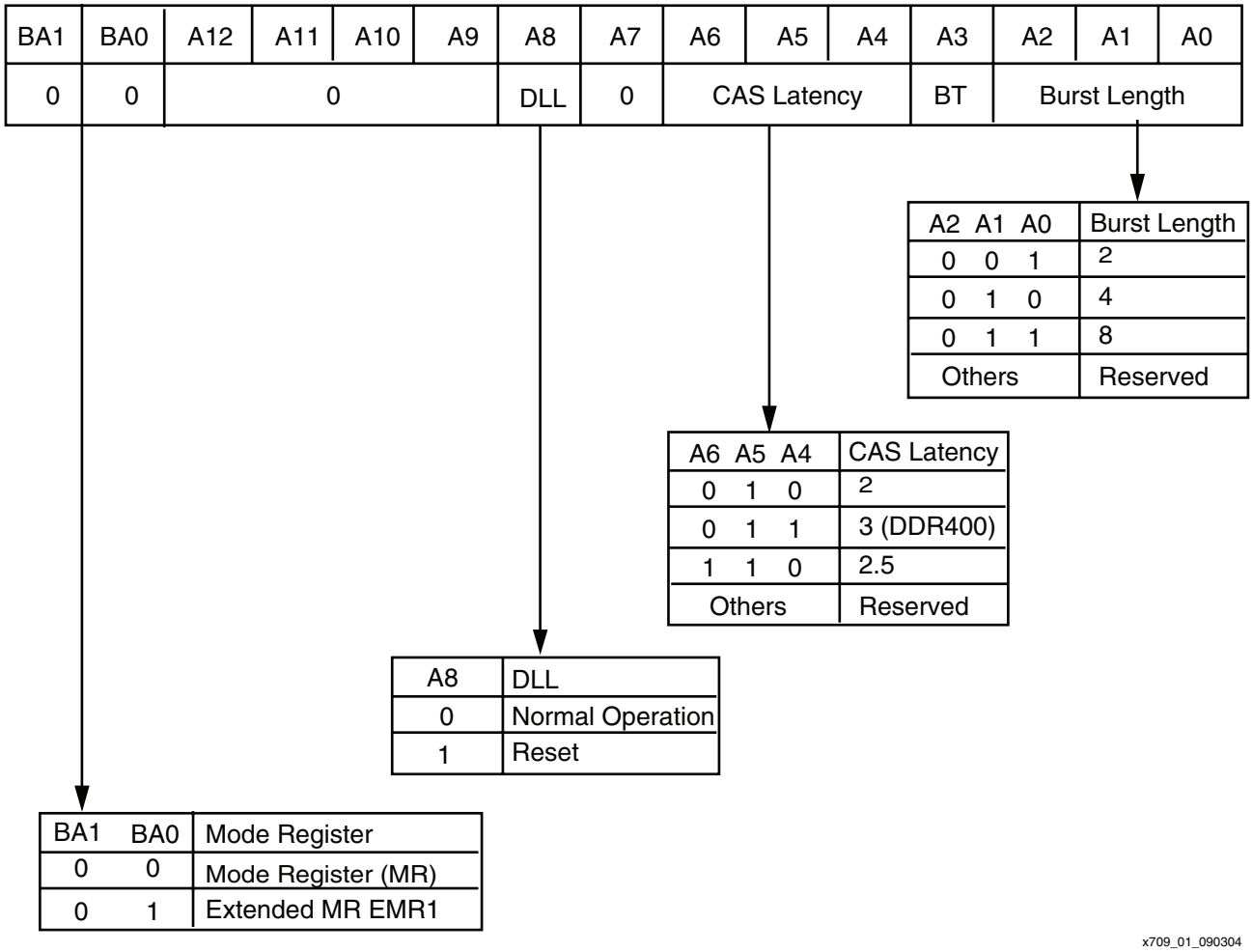


*Figure 1:* **Mode Register Definition for DDR400**

Bank Addresses BA1 and BA0 select the Mode registers. Figure 1 shows the Bank Address bits configuration.

### Extended Mode Register

The Extended Mode register controls functions beyond those controlled by the Mode register. These additional functions are DLL enable/disable and output drive strength for DDR SDRAM interfaces shown in Figure 2.

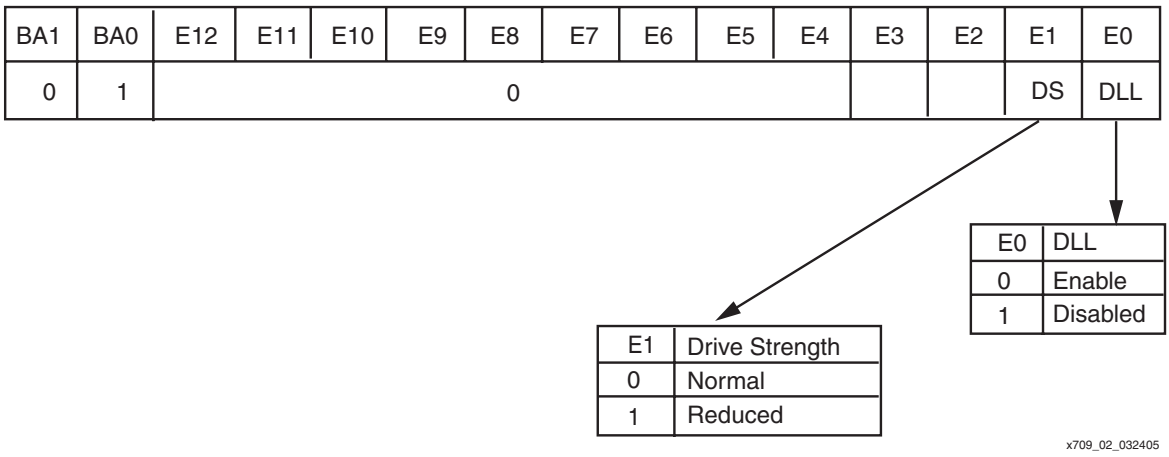


*Figure 2:* **Extended Mode Register for DDR400**

### Initialization Sequence

The Initialization Sequence used in the controller state machine follows the DDR SDRAM specifications. The configuration sequence is split in two steps: the first step is handled by the hardware at power up; the second step is handled by the FPGA memory controller design. Figure 3 shows the sequence of commands issued for initialization.

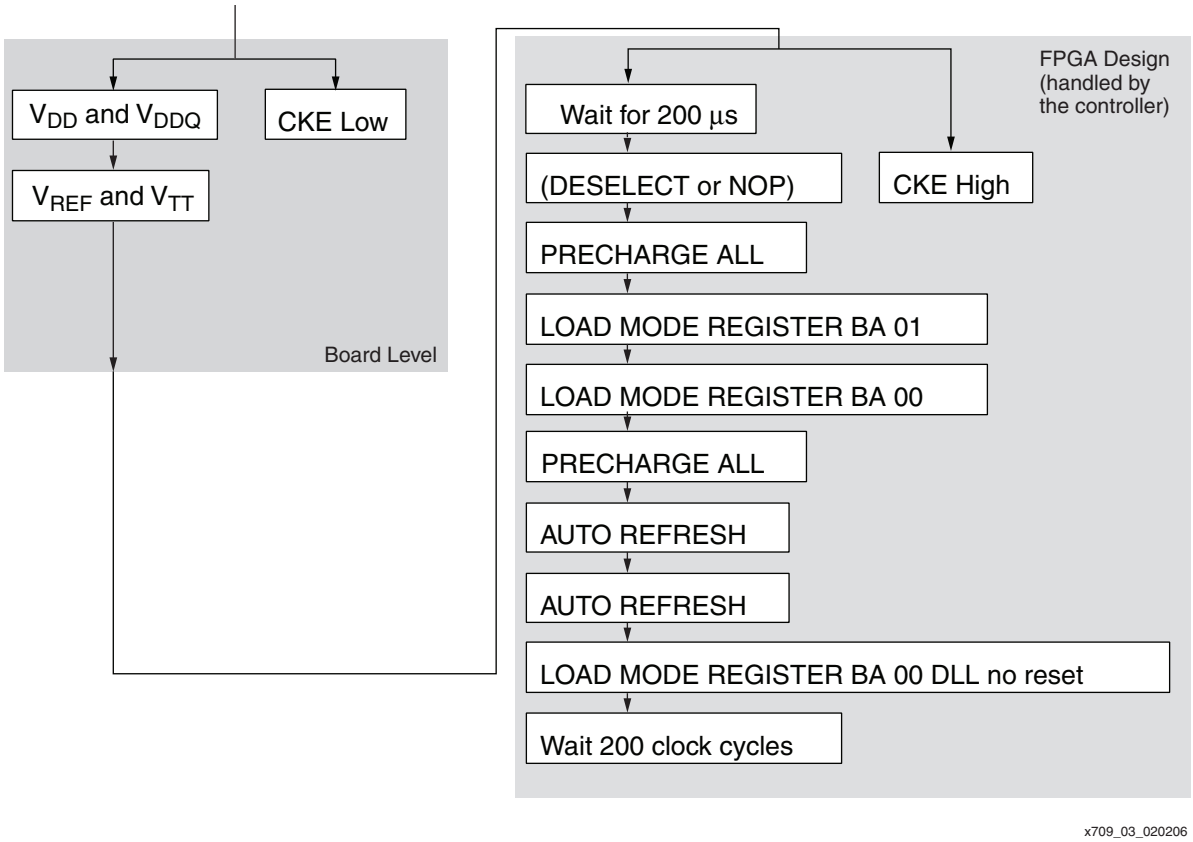


*Figure 3:* **Initialization Sequence for DDR SDRAM with Virtex-4 Device**

After the initialization sequence is complete, the controller issues dummy Read commands to the DDR SDRAM memory device. This process allows the datapath module to select the right number of taps in the Virtex-4 input delay block. The datapath module uses the Data Strobe (DQS) issued by the memory during this dummy Read command period to determine the relationship between the incoming DQS to the internal system clock (CLK0). After the datapath

module has determined the right number of delay taps required, a Tap_select_done signal is issued to the controller. The controller then goes into IDLE state.

### Calibration of Read Enables

After DQS calibration, the controller issues a command to write a predefined data pattern to the memory. After the write operation is complete, data is read back and compared with the written data. A module counts the number of clock periods that it took after issuing the read command to receive the valid data in this module. This number is used to set up the correct internal delay on the Read Enable signals for normal read operation.

### PRECHARGE Command

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

### AUTO REFRESH Command

DDR devices are required to be refreshed once every 7.8 µs. The circuit to request auto refresh counters is built inside the controller. The controller uses the DCM's CLKDV output for the refresh counter. This output provides the low frequency clock required for the auto refresh counter. To save on the BUFG that is used by the CLKDV output of the DCM, designers can use the high frequency CLK0 output of the DCM or the CLK/4 output of the DCM (used by the IDELAY Circuit) to clock the refresh counter. If the clock to the auto refresh circuitry is changed, the max_ref_count in the mem_interface_top_parameters_0.v file needs to be changed accordingly.

The auto_ref signal flags the need for a pending AUTO REFRESH command. This signal is held High until the controller issues an AUTO REFRESH command. The controller completes the transactions in a current open bank before issuing the AUTO REFRESH command.

### ACTIVE Command

The ACTIVE command activates a row in a bank, allowing any READ or WRITE commands to be issued to a bank in the memory array. After a row has been opened, READ or WRITE commands can be issued to that row, subject to the $t_{RCD}$ specification. When the controller detects an incoming address that refers to a row in a bank other than the currently opened row, the controller issues an address conflict signal. A PRECHARGE command is also issued by the controller to deactivate the open row. The controller also issues another ACTIVE command to the new row.

### READ Command

The READ command is used to initiate a burst read access to an active row. The value on BA0 and BA1 selects the bank address. The address inputs provided on A0 – Ai select the starting column location. After the read burst is over, the row is still available for subsequent access until it is precharged.

Figure 4 shows the case of a READ command with an additive latency of zero. Hence, the Read latency in this case is the same as the CAS latency, which is 3 in the DDR400 specification.

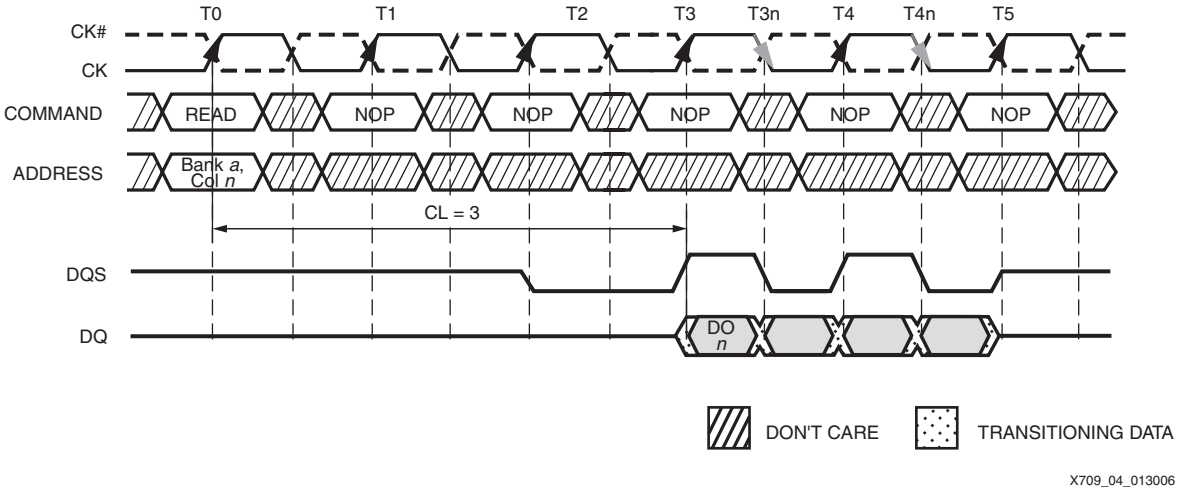


*Figure 4:* **DDR SDRAM Read Access Waveforms**

### WRITE Command

The WRITE command is used to initiate a burst access to an active row. The value on BA0 and BA1 selects the bank address, while the value on address inputs A0 – Ai selects the starting column location in the active row. The value of Write Latency is equal to one clock cycle.

Figure 5 shows a Write burst with a Write latency of 1. The time between the WRITE command and the first rising edge of the DQS signal is determined by the Write latency.

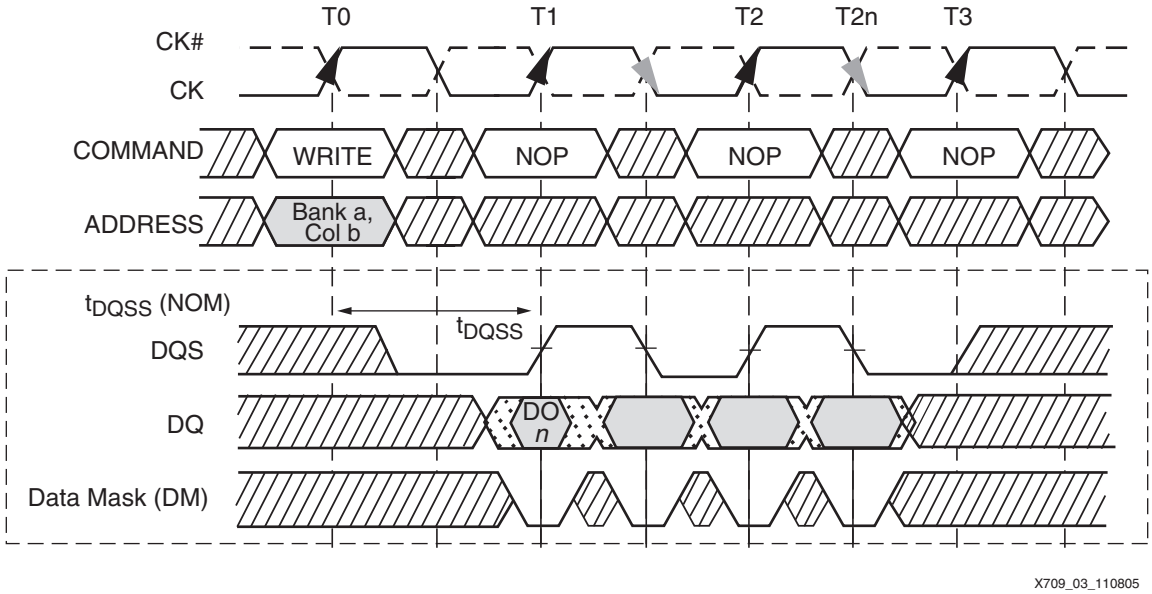


*Figure 5:* **DDR SDRAM WRITE Command Waveform**

## Timing Analysis

The Virtex-4 DDR reference design leverages the unique I/O and clocking features of the device to maximize the performance and timing margins.

The physical layer on which the reference design is based uses the Direct Clocking method. It is described in XAPP701: http://www.xilinx.com/bvdocs/appnotes/xapp701.pdf.

This section presents an example of timing analysis for the address/control paths, the Write datapath, and the Read (or capture) datapath.

### Address/ Control Paths

The address and control signals are synchronized on CLK180 to ensure sufficient set-up and hold margin for the memory device in respect to the clock CLK0.

Table 2 shows the address and control signal timing analysis at 165 MHz for the DDR interface. The designer must factor in additional board-specific conditions, such as differences in loading between the clock and address/control nets for an implementation in a Virtex-4, -10 speed grade device.

*Table 2:* **Address and Control Signal Timing Analysis at 165 MHz**

| Parameter | Value (ps) | Leading Edge Uncertainties | Trailing Edge Uncertainties | Description |
|---|---|---|---|---|
| $T_{CLOCK}$ | 6061 | | | Clock period. |
| $T_{SETUP}$ | 600 | 600 | 0 | - |
| $T_{HOLD}$ | 600 | - | 600 | - |
| $T_{PACKAGE\_SKEW}$ | ±20 | 20 | 20 | Package skew. |
| $T_{DCD\_MEMORY\_DLL}$ | ±150 | 150 | 150 | Falling edge of CLK0 (i.e., rising edge of CLK180) used to clock out rising edge of DDR CK. Rising edge of CLK0 used to clock out address/control. |
| $T_{JITTER}$ | ±0 | 0 | 0 | Same DCM output used to generate CK and address/control. |
| $T_{CLOCK\_TREE\_SKEW}$ | ±100 | 100 | 100 | Small value used. Assumes CK/CK# outputs placed close to control/address. |
| $T_{CLKOUT\_PHASE}$ | ±140 | 140 | 140 | Phase offset between any DCM output parameter value, see DS302:*Virtex-4 Data Sheet.* |
| $T_{PCB\_LAYOUT\_SKEW}$ | ±400 | 400 | 400 | Skew between address/control lines and associated CK/CK# traces on the board. |
| Total Uncertainties | | 1410 | 1410 | - |
| Timing Margin | | 1620 | 1620 | - |

### Write Datapath

The Write datapath is synchronized to CLK90. However, the Write data words are transmitted as DDR values and, therefore, must have adequate set-up and hold margins with respect to both the rising edge and falling edges of CLK180.

Accordingly, the timing analysis for the Write datapath, shown in Table 3, incorporates the maximum duty cycle distortion of the memory clocks. This analysis is also for a Virtex-4 device, -10 speed grade.

*Table 3:* **Write Datapath Timing Analysis at 165 MHz**

| Parameter | Value (ps) | Leading Edge Uncertainties | Trailing Edge Uncertainties | Description |
|---|---|---|---|---|
| $T_{CLOCK}$ | 6061 | - | - | Clock period. |
| $T_{DCD}$ | ±303 | 303 | 303 | Duty cycle distortion of memory clock (5% of clock period). |
| $T_{CLOCK\_PHASE}$ | 2727 | - | - | Data period is half the clock period with 10% duty cycle distortion subtracted from it. |
| $T_{SETUP}$ | 400 | 400 | 0 | Set-up time from memory data sheet. |
| $T_{HOLD}$ | 400 | 0 | 400 | Hold time from memory data sheet. |
| $T_{PACKAGE\_SKEW}$ | ±20 | 20 | 20 | Package skew. |
| $T_{JITTER}$ | ±50 | 50 | 50 | Same DCM used to generate CLK0/CLK180 and CLK90. Only the difference in jitter between CLK0, CLK90 outputs should be considered. |
| $T_{CLOCK\_SKEW\_FPGA}$ | ±100 | 100 | 100 | Small value considered for skew on global clock line because detection of DQS and associated DQ are placed close to each other. |
| $T_{CLKOUT\_PHASE}$ | ±140 | 140 | 140 | Phase offset between any DCM output parameter value, see the *Virtex-4 Data Sheet*. |
| $T_{PCB\_LAYOUT\_SKEW}$ | ±50 | 50 | 50 | Skew between data lines and associated strobe on the board. |
| Total Uncertainties | | 760 | 760 | - |
| Timing Margin | | 603 | 603 | |

### Read Datapath

The Read datapath values are captured directly into the FPGA clock domain using the previously described direct clocking techniques. Thus, the data capture timing analysis must be performed with respect to CLK0 and consideration must be given to the IDELAY tap delay resolution. In addition, the potential skew between the DQS strobes and the bits of data on the bus must also be taken into account. Table 4 presents the timing analysis for Read data capture, again assuming a Virtex-4 device, -10 speed grade. Refer to *XAPP701* for a discussion of the delay parameters used in the read timing analysis.

*Table 4:* **Read Datapath Timing Analysis at 165 MHz**

| Uncertainty Parameters | Value (ps) | Description |
|---|---|---|
| $T_{CLOCK}$ | 6061 | Clock period. |
| $T_{CLKOUT\_DUTY\_CYCLE\_DLL}$ | 150 | DCM output duty cycle distortion is subtracted from clock phase (equal to half clock period) to determine $T_{DATA\_PERIOD}$. |
| $T_{DATA\_PERIOD}$ | 2880 | Data period is half the clock period with duty cycle distortion subtracted from it. |
| $T_{AC}$ | 1400 | Memory data output access time specified. |
| $T_{PACKAGE\_SKEW}$ | 20 | A small value for package skew is considered because PCB trace lengths are adjusted to compensate for this skew. |
| $T_{SAMP}$ | 550 | This parameter is defined in the Virtex-4 source synchronous data sheet and includes a number of sources of uncertainty. |
| $T_{IDELAYPAT\_JIT}$ | 732 | At 165 MHz, the total number of taps in the worst case is 3/4 x clock_period = 61 taps.<br>61 x 12 = 732 ps of pattern jitter. |
| $T_{CLOCK\_TREE\_SKEW}$ - Maximum | 100 | Small value considered for skew on global clock line because DQS and associated DQ are placed close to each other. |
| $T_{PCB\_LAYOUT\_SKEW}$ | 50 | Skew between data lines and associated strobe on the board. |
| Uncertainties | 2852 | |
| Window | 28 | |

# DDR SDRAM Interface Implementation

This section presents the characteristics of the DDR SDRAM controller and interface, the interface block diagram (Figure 6), and the controller state machine (Figure 7).

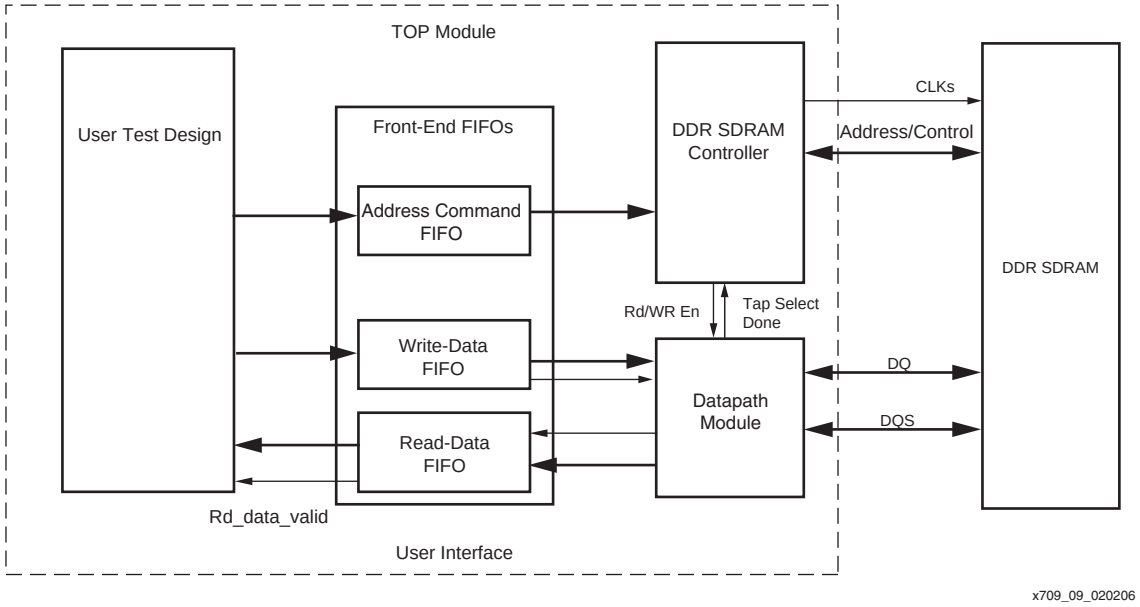


*Figure 6:* **Controller Design Block Diagram**

## Hardware Testbench

The hardware testbench provides address and data patterns to test all the design aspects of the DDR SDRAM controller. The user 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 device. The address locations are pre-stored in a block RAM and used here as a Read Only Memory (ROM). The address values stored have been selected to test accesses to different rows and banks in the DDR SDRAM device. The data pattern generator includes a state machine that issues data patterns. The backend state machine serves as a user design and issues the Write or Read Enable signals to determine which FIFO needs to be accessed by the data generator module.

## User Interface

The backend user interface is comprised of three FIFOs, namely the Address Command FIFO, the Write Data FIFO, and the Read Data FIFO. The first two FIFOs are accessed by the user backend modules, while the Read Data FIFO is accessed by the datapath module to store the captured Read data.

## DDR SDRAM Controller Interface

Figure 7 presents the state machine of the DDR SDRAM command generation state machine.

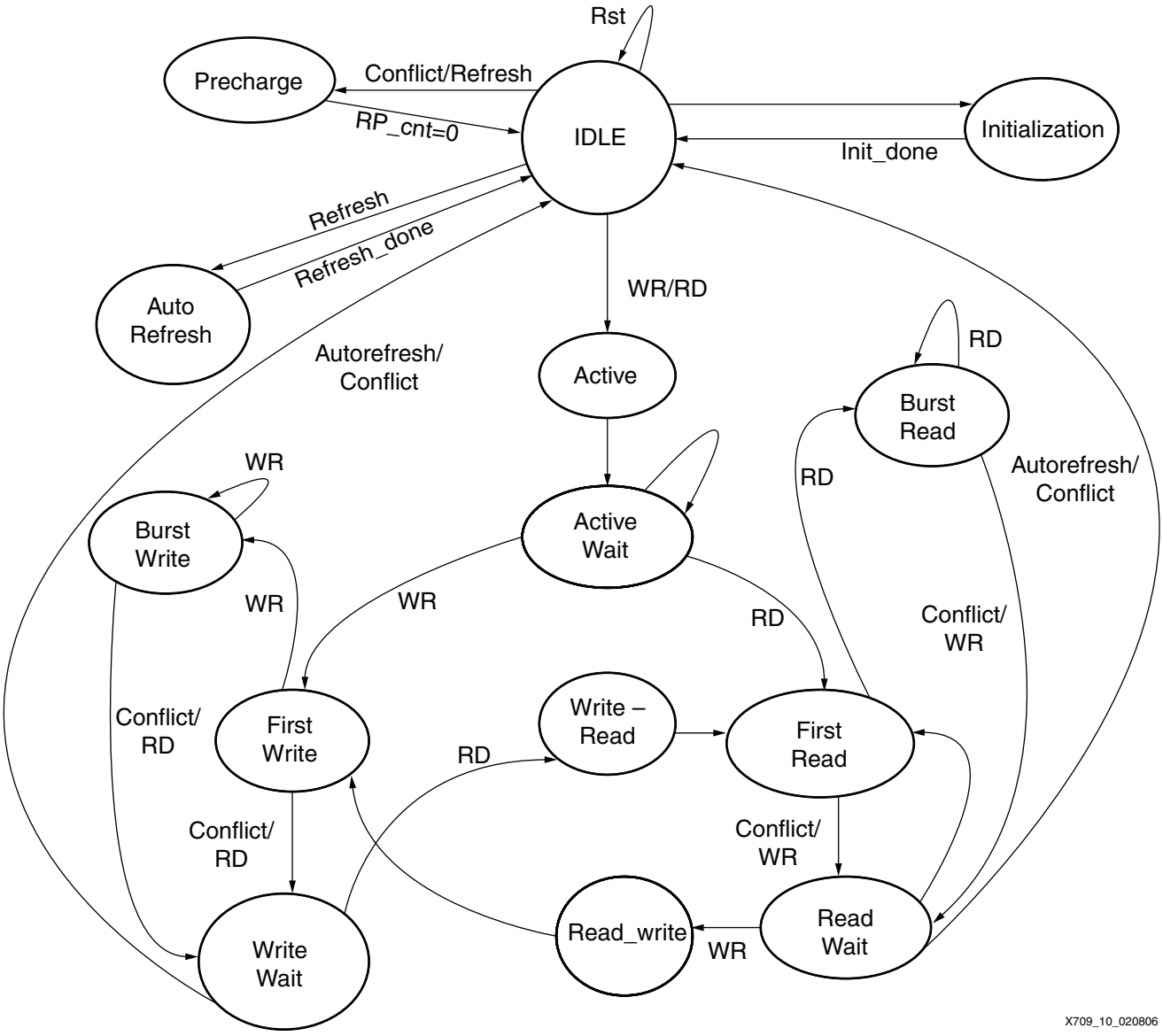


*Figure 7:* **DDR SDRAM Controller State Machine**

These steps are followed before the controller issues the commands to the memory:

1. The controller issues a read-enable signal to the read/write address FIFO.
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. In the write state, if the controller detects 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 detects a Write command from the command logic block, the controller waits for the read_to_write time before issuing the Write command.

4. A dynamic command request from the backend user application for a Precharge, Auto Refresh, Active, or Load Mode register results in the controller issuing a Precharge command.
5. The commands are pipelined to synchronize with the address signals before being issued to the DDR memory.

## Reference Design Specifications

The reference design for the DDR SDRAM memory controller using the Direct Clocking Data Capture technique is integrated with the Memory Interface Generator (MIG) tool. This tool has been integrated with 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

The design has been hardware characterized at frequencies higher than 200 MHz. Table 5 provides the reference design specifications.

*Table 5:* **Reference Design Specifications**

| Parameter | Specifications/Details |
|---|---|
| Frequency of operation | 165 MHz |
| Virtex-4 device speed grade | -10 |
| Device utilization for component design with test bench and ChipScope™ | 1269 Slices |
| Number of slices for DIMM interface only | 2846 Slices |
| Number of slices for component interface only | 944 Slices |
| Supported burst modes | 2, 4, and 8 |
| Supported CAS latency | 2, 2.5, and 3 |
| HDL language | Verilog, VHDL |
| Bus width | 8–144 bits |
| Device used for verification for components | Micron MT46V32M16 |
| Device used for verification for DIMMs | Micron MT18VDDF6472AG-40BG4 |

### Design Files

Table 6 lists the reference design files:

*Table 6:* **Design File List With Descriptions for the Verilog Version**

| Module Name | Description of Each Module |
|---|---|
| top.v | Top-level module for the DDR controller and Physical Layer. |
| data_path.v | Top level for the physical layer. Instantiates the following modules: tap_ctrl, data_tap_inc, idelay_ctrl, idelay_rd_en, v4_dqs_iob, v4_dq_iob, and rd_data_fifo. |
| data_tap_inc.v | Implements the tap selection controller for data bits associated with a strobe. |
| idelay_ctrl.v | Instantiates the IDELAYCTRL primitive required when the IDELAY primitive is used in the design. |

*Table 6:* **Design File List With Descriptions for the Verilog Version *(Continued)***

| Module Name | Description of Each Module |
|---|---|
| tap_ctrl.v | This module detects two transitions of a DQS signal and determines the tap delay required for the associated data bits in order to center them with respect to internal FPGA clock, CLK. |
| v4_dm_iob.v | Instantiates the IDELAY primitive and IOB flip-flops for the bidirectional data. |
| v4_dq_iob.v | Instantiates the IDELAY primitive and IOB flip-flops for the bidirectional data. |
| v4_dqs_iob.v | Instantiates the IDELAY primitive and IOB flip-flops for the bidirectional strobe. |
| ddr_controller.v | • Provides the read enable signals to the Write Address, Write Data, and the Read Address FIFOs.<br>• Includes the controller state machine. Supplies the right command signals to the DDR device. AUTO REFRESH commands are generated by the controller taking the Auto Refresh Command Interval into account.<br>• Provides the address signals to the DDR device. |
| test_bench.v | Synthesizable test bench for the memory interface. |
| backend_rom.v | Stores the data and address to be written and accessed in the memory array. |
| cmp_rd_data.v | Generates the error signal in case of bit errors. It compares the read data with expected data value. |
| user_interface.v | Instantiates the FIFO16 primitive for read data. One FIFO for rising-edge data and the other for falling-edge data. |
| backend_fifo.v | Instantiates the backend FIFOs for the DDR SDRAM interface. This includes the Write Address and Data FIFOs, and the Read Address and Data FIFOs. The FIFOs are implemented using the Virtex-4 FIFO16 primitives or FIFO made from LUT RAM instances. |
| Rd_data_fifo.v | Instantiates the FIFO16 primitive for read data. One FIFO for rising-edge data and the other for falling-edge data. |
| RAM_D.v | Instantiates LUT RAMS to build the asynchronous FIFOs. It is configured to clock data on the rising edge of the clock. |
| parameter.v | Values to be used for the DDR SDRAM reference design at 200 MHz. |
| Infrastructure.v | The Virtex-4 DCM primitive is instantiated. Used to generate the FPGA clk_0, clk_90, and the system reset synchronous with the respective clocks. |
| Infrastructure_iobs.v | Instantiates the ODDR and the OBUFDS primitives for memory clocks. |
| Pattern compare.v | Compares the read data from the memory with a fixed pattern to calculate the read enable delay from the point where the command is issued. |
| Main.v | Instantiates the top.v and the test_bench.v modules. |

*Table 6:* **Design File List With Descriptions for the Verilog Version *(Continued)***

| Module Name | Description of Each Module |
|---|---|
| Mem_interface_top.v | Wrapper module that connects the memory interface and testbench designs, i.e., the clock, reset, and memory device. |
| Addr_gen.v | Instantiates the block RAM in which the user address values and commands are stored at reset. |
| Data_gen.v | Generates the data pattern for the write and the read data. |
| Data_write.v | Generates the control signals for the data iobs. |
| Tap_logic.v | Instantiates the tap_ctrl.v and the data_tap_inc.v. |
| Iobs.v | Instantiates the data_path_iobs.v, controller_iobs.v, and the infrastructure_iobs.v. |
| Data_path_iobs.v | Instantiates the v4_dq_iob, v4_dm_iob and v4_dqs_iob modules. |
| Controller_iobs.v | Instantiates the OBUF primitive for memory control signals. |
| Wr_data_fifo.v | Instantiates the FIFO16 primitive, which stores the user data and the mask information. |

## Design Hierarchy

Figure 8 summarizes the hierarchy in the reference design in tree view:

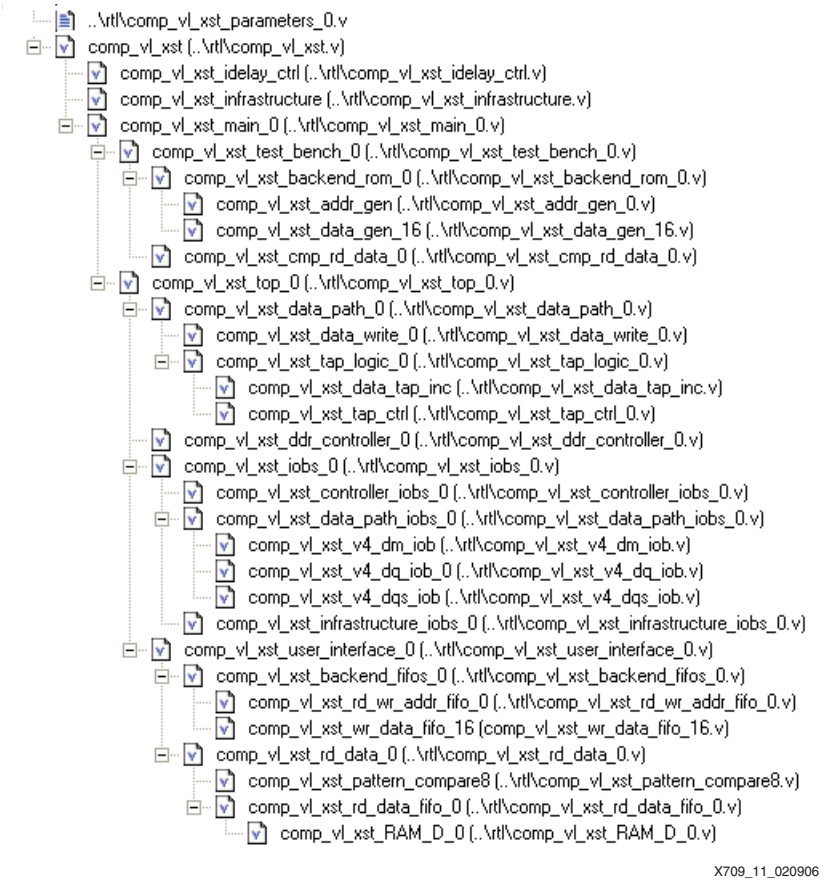


*Figure 8:* **Design Hierarchy**

## Revision History

The following table shows the revision history for this document.

| Date | Version | Revision |
|---|---|---|
| 09/10/04 | 1.0 | Initial Xilinx release. |
| 11/16/04 | 1.1 | Revised "ACTIVE Command" and "WRITE Command" sections. Added a link to the reference design. Completed Table 6. |
| 04/01/05 | 1.2 | Updated Figure 2, Figure 5, reference design with the results of hardware testing on ML461 platform, "Design Hierarchy", Table 5, and Table 6. Added "Timing Analysis". |
| 08/27/05 | 1.3 | Clarified Table 2 and Table 3. Updated reference design with ChipScope files. |
| 11/18/05 | 1.4 | Updated reference design link. See "Reference Design Specifications." Updated Read Data Timing Analysis (see Table 4) and Figure 8. |
| 03/27/06 | 1.5 | Updated Figure 3, replaced Figure 4, updated Figure 6, Figure 7, and Figure 8, added the "Calibration of Read Enables" section, updated the "WRITE Command" section, updated Table 5 and Table 6. Reference design updated in MIG 1.5. |
| 10/27/06 | 2.0 | Updated Table 2, Table 3, and Table 4. Deleted Figure 6, Figure 7, and Figure 8. |