XAPP549 (v1.2) April 30, 2007

DDR2 SDRAM Memory Interface for Virtex-II Pro FPGAs

Author: Maria George

Summary

This application note describes a DDR2 SDRAM memory interface for Virtex™-II Pro FPGAs.

Architecture

This DDR2 SDRAM memory interface has a 72-bit data width. The data bus must be placed on either the left or right side banks in the FPGA; the data bus can not be split. If sufficient pins are not available in the FPGA, the data width must be reduced. Address and control signals can use either the same side as the data bus, or any other bank.

Like DDR SDRAM memory, DDR2 SDRAM memory is source-synchronous and has a double-data-rate interface. It performs data transfers on both edges of a clock cycle.

The advancements of DDR2 SDRAM memories are largely the result of changes in DRAM architecture and signaling, as well as additions to the mode register for lower power and improved command and data bandwidth.

The SSTL_18 Class II I/O standard is used for address, control, and data.

Interface Model

The DDR2 SDRAM memory interface is layered to simplify the design and make the design modular. Figure 1 shows the layered memory interface. The three layers consist of an application layer, an implementation layer, and a physical layer.

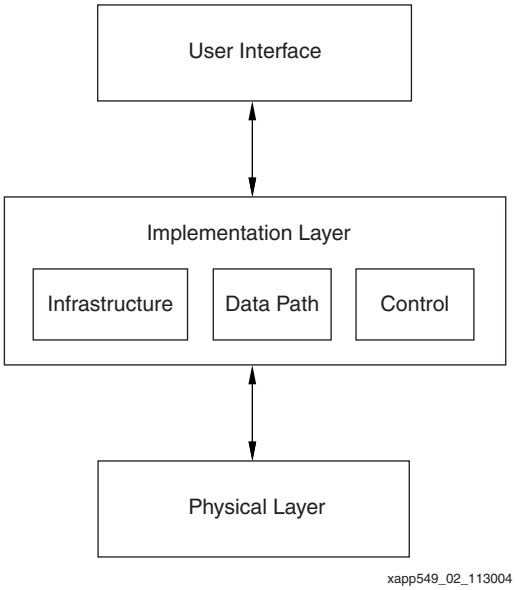

Figure 1: **Interface Layering Model**

DDR2_V2P Controller Modules

Figure 2 is a block diagram of the DDR2_V2P memory interface. The four blocks shown in this figure are the major blocks of the DDR2 SDRAM controller design. The function of each block is explained in the following sections.

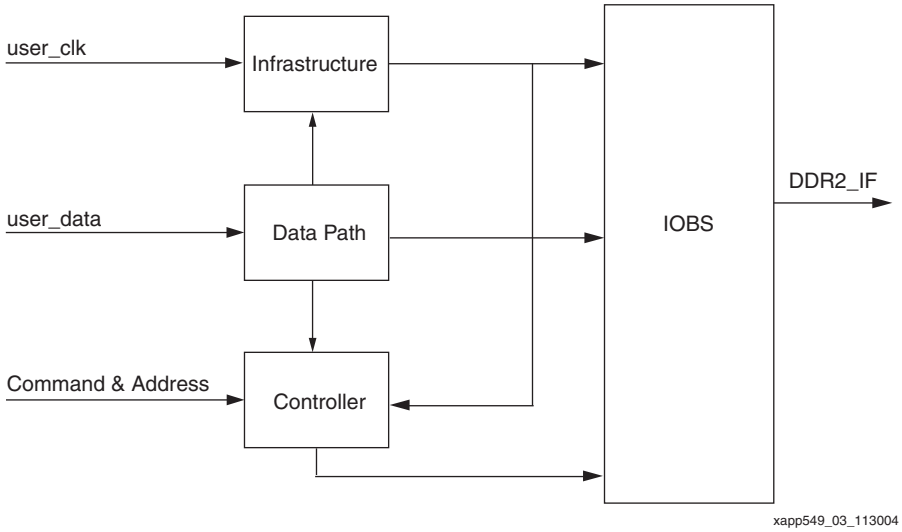

Figure 2: **DDR2_V2P Memory Interface Modules**

Controller

The controller's design supports a burst length of four, and CAS latencies of three and four. The design implements the write latency feature of DDR2 SDRAM memory. The controller initializes the EMR(2) and EMR(3) registers during DDR2 SDRAM initialization.

The controller accepts user commands, decodes these user commands, and generates read, write, and refresh commands to the DDR2 SDRAM memory. The controller also generates signals for other modules. Detailed design and timing analyses of the controller module are provided in later sections of this document.

Data Path

The data path module is responsible for transmitting data to and receiving data from the memories. Major functions include:

- Writing data to the memory
- Reading data from the memory
- Transferring the read data from the memory clock domain to the FPGA clock domain

For a description of write data transmission and read data capture techniques, see XAPP688, "Creating High-Speed Memory Interfaces With Virtex-II and Virtex-II Pro FPGAs," and XAPP678c, "Data Capture Technique Using CLB Flip-Flops" (available only under NDA).

The write data and strobe are clocked out of the FPGA. The strobe is center-aligned with respect to the data. For DDR2 SDRAM memories, the strobe must be differential and non-free running. The design supports differential and non-differential data strobes. To meet the requirements specified above, the write data is clocked out using a clock that is shifted 90° and 270° from the primary clock going to the memory. The differential data strobes are generated out of primary clocks going to the memory.

Memory read data is edge-aligned with a source-synchronous clock. The DDR2 SDRAM clock is a non-free running strobe. The data is received using the non-free running strobe and transferred to the FPGA clock domain. The input side of the data uses resources similar to the input side of the strobe. This ensures matched delays on data and strobe signals until the strobe is delayed in the strobe delay circuit.

Infrastructure

The Infrastructure module generates the FPGA clocks and reset signals. A Digital Clock Manager (DCM) is used to generate the clock and its inverted version. A delay calibration circuit is also implemented in this module.

The delay calibration circuit is used to select the number of delay elements used to delay the strobe lines with respect to read data. The delay calibration circuit calculates the delay of a circuit that is identical in all respects to the strobe delay circuit. All aspects of the delay are considered for calibration, including all the component and route delays. The calibration circuit selects the number of delay elements for any given time. After the calibration is done, it asserts the select lines for the delay circuit. Refer to XAPP688 and XAPP678c for details about delay calibration.

IOBS

All FPGA input and output signals are implemented in the IOBS module. All address and control signals are registered going into and coming out from the IOBS module.

User Interface Signals

Table 1 shows user interface signal descriptions; all signal directions are with respect to the DDR2_V2P controller.

Table 1: **User Interface Signals**

Signal Name	Direction	Description
dip1	Input	Clock enable signal for DDR2 SDRAM (active low). This is common for multicontrollers.
dip3	Input	Enable signal for DDR2 SDRAM memory chip select. This is common for multicontrollers.
rst_dqs_div_in	Input	This active Low signal enables the dqs_div flop during DDR2 SDRAM memory read. Users do not need to input this signal. The rst_dqs_div_out signal from the controller is connected to this input.
reset_in	Input	System reset
user_input_data[(2n-1):0]	Input	Write Data for DDR2 SDRAM, where 'n' is the width of the memory interface
user_input_address[addwidth:0]	Input	DDR2 SDRAM row and column address
user_bank_address[bankaddwidth:0]	Input	DDR2 SDRAM bank address
user_config_reg1[14:0]	Input	DDR2 SDRAM configuration data register1
user_config_reg2[12:0]	Input	DDR2 SDRAM configuration data register2
user_command_reg[3:0]	Input	User command register for DDR2 SDRAM controller
burst_done	Input	Burst data transfer done signal
rst_dqs_div_out	Output	This signal is externally connected to rst_dqs_div_in. This active Low signal enables the dqs_div flop. The High state of the signal keeps the dqs_div flop in clear condition.
user_output_data[(2n-1):0]	Output	Read data from DDR2 SDRAM
user_data_valid	Output	This active low signal indicates that read data from DDR2 SDRAM memory is valid.
user_cmd_ack	Output	Acknowledge signal for user_command
init_val	Output	Indicates DDR2 SDRAM is initialized
ar_done	Output	Indicates auto-refresh command has executed.

Notes:

1. All signal directions are with respect to the DDR2_V2P controller.

Signal Descriptions

user_input_data[(2n-1):0]

This is the write data to DDR2 SDRAM from the user interface. The data is valid on a DDR2 SDRAM write command, where n is the width of the DDR2 SDRAM memory. The DDR2 SDRAM controller converts single data rate to double data rate on the physical layer side.

user_input_address[addwidth:0]

This is the sum of row and column address for DDR2 SDRAM writes and reads. Depending on address width variable selection, user_input_address is divided into row and column address bits.

user_bank_address[bankaddwidth:0]

Bank address for DDR2 SDRAM. There is a variable through which the bank address is selectable.

user_config_reg1[14:0]

Configuration data for DDR2 SDRAM memory initialization. The contents of this register are loaded into the mode register during a Load Mode command. The format for user_config_reg1 is as follows:

14	13 – 11	10	9 – 7	6 – 4	3	2 – 0
PD	WR	TM	Res	Cas_latency	BT	Burst_length

Burst_length[2:0]

The controller supports only a burst length of four.

BT

This bit selects the burst type. The controller supports only sequential bursts. This bit is always set to zero in the controller.

Cas_latency [6:4]

Bits 6:4 select the cas latency. The DDR2 SDRAM controller supports a cas latency of 3 and 4.

Res [9:7]

Bits 9:7 are reserved for future implementation.

TM

This bit is loaded into the TM bit of the Load Mode Register.

WR [13:11]

These three bits are written to WR (write recovery) bits of the Load Mode register.

PD

This bit is written to PD (Power Down Mode) bit of the Load Mode register.

Refer to the Micron DDR2 SDRAM data sheets for details on the Load Mode register.

user_config_reg2[12:0]

DDR2 SDRAM configuration data for the Extended Mode Register. The format of user_config_reg2 is as follows:

12	11 – 10	9 – 7	6 – 4	3 – 2	1	0
OUT	RDQS	DQS	OCD	Posted CAS	RTT	ODS	Res

Refer to the Micron DDR2 SDRAM data sheets for details on the Extended Mode register.

user_command_reg[3:0]

This is the user command register. Various commands are passed to the DDR2_V2P module through this register. Table 2 lists supported commands.

Table 2: **User Commands**

user_command_reg[3:0]	User Command Description
0000	NOP
0010	Memory (DDR2 SDRAM) initialization
0011	Auto-refresh
0100	Write
0101	Load Mode (Only Load mode)
0110	Read
Others	Reserved

burst_done

Users should enable this signal, for two clock periods, at the end of the data transfer. The DDR2 SDRAM controller supports write burst or read burst for a single row. Users must terminate read or write commands on a column boundary and reinitialize for the next row of transactions. The controller terminates a write burst or read burst by issuing a pre-charge command to DDR2 SDRAM memory.

user_output_data[(2n-1):0]

This is the read data from DDR2 SDRAM memory. The DDR2 SDRAM controller converts DDR SDRAM data from DDR2 SDRAM memory to SDR data. As the DDR SDRAM data is converted to SDR data, the width of this bus is 2n, where n is data width of DDR2 SDRAM memory.

user_data_valid

The user_output_data[(2n-1):0] signal is valid on assertion of this signal.

user_cmd_ack

This is the acknowledgement signal for a user read or write command. It is asserted by the DDR2 SDRAM controller during a read or write to DDR2 SDRAM. No new command should be given to the controller until this signal is deasserted.

init_val

The DDR2 SDRAM controller asserts this signal after completing DDR2 SDRAM initialization.

ar_done

The DDR2 SDRAM controller asserts this signal for one clock cycle after the auto-refresh command is given to DDR2 SDRAM.

Table 3 shows memory interface signals.

Table 3: **Memory Interface Signals**

Signal Name	Direction	Description
ddr_dq[(datawidth –1):0]	Input	Bidirectional DDR2 SDRAM memory data
ddr_dqs[(dqswidth-1):0]	Input	Bidirectional DDR2 SDRAM memory data strobe (positive) signals. The number of strobe signals depends on the data width and strobe to data ratio.
ddr_dqs_n[(dqswidth-1):0]	Input	Bidirectional DDR2 SDRAM memory data strobe (negative) signals. These are valid, when the differential strobes option is selected.
ddr_cke	Output	Clock enable signal for DDR2 SDRAM memory
ddr_csb	Output	Active low memory chip select signal
ddr_rasb	Output	Active low memory row address strobe
ddr_casb	Output	Actie low memory column address strobe
ddr_web	Output	Active low memory write enable signal
ddr_dm	Output	Memory data mask signal for write data
ddr_ba	Output	Memory bank address
ddr_address	Output	Memory address (both row and column address)
ddr2_clk*	Output	Memory differential clock signals
ddr_odt	Output	Memory on-die termination signal

Initializing DDR2 SDRAM Memory

Before issuing the memory read and write commands, the DDR2 SDRAM memory must be initialized using the memory initialization command. The data to be written in the Mode Register and in the Extended Mode Register is placed on user_config_reg1[14:0] and user_config_reg2[12:0] until DDR2 SDRAM initialization is completed. Once the DDR2 SDRAM is initialized, the init_val signal is asserted by the DDR2 SDRAM controller. Figure 3 shows a timing diagram of the memory initialization command.

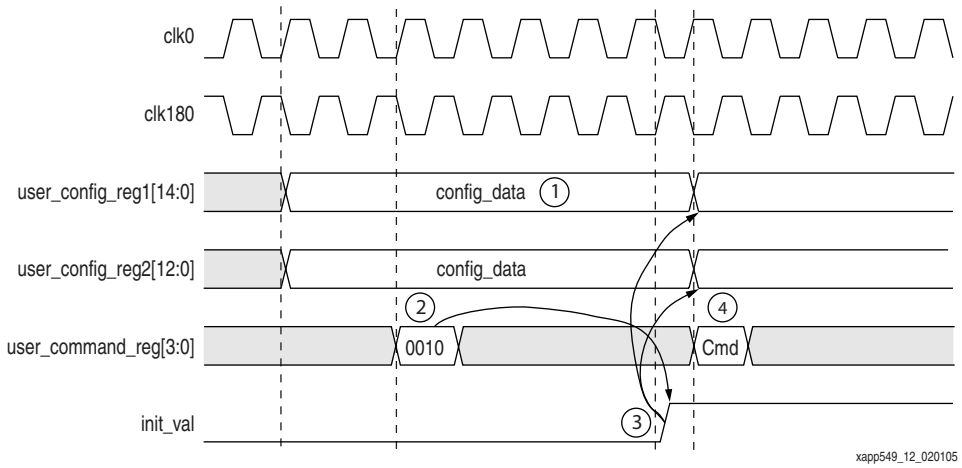

Figure 3: **DDR2 SDRAM Memory Initialization**

1. The user places valid configuration data on user_config_reg1[14:0] and user_config_reg2[12:0] two clocks prior to placing the initialization command (2) on user_command_reg[3:0].
2. The user places the initialization command (cmd = 2) on user_command_reg[3:0] for one clock cycle. This starts the initialization sequence. The user command is asserted on a rising edge of clk0.
3. The controller indicates that the configuration is complete by asserting the init_val signal. The init_val signal is asserted on a rising edge of clk180.
4. After init_val is asserted, the user can pass the next command at any time.

DDR2 SDRAM Memory Write

Figure 4 shows a DDR2 SDRAM memory write timing diagram for a burst length of four with two successive bursts. Memory write is preceded by a write command to the DDR2 SDRAM controller. The write command and write address are asserted on a rising edge of clk0. The first write address should be asserted along with the write command. In response to the write command, the DDR2 SDRAM controller acknowledges with a user_cmd_ack signal on a rising edge of clk180. Users should wait for this signal before proceeding to the next step.

Two and a half clock cycles after user_cmd_ack is asserted, the next memory burst address is placed on user_input_address[addwidth:0] lines. The user_input_address is asserted on a rising edge of clk0. Any subsequent write addresses are asserted on an alternate positive edge of clk0. The data to be written into memory is asserted with clk90. The first user_input_data is provided on a positive edge of clk90 after user_cmd_ack is asserted. On every rising edge of clk90, subsequent data is asserted after user_cmd_ack is asserted. The user data width is twice that of the memory data width. The controller converts it into double data rate before it is passed to memory.

For a burst length of four, four pieces of data are given to the DDR2 SDRAM controller with each user address. To terminate the write burst, burst_done is asserted on the rising edge of clk0 for two clocks. The burst_done signal is asserted two clocks after the last memory address. The user command is deasserted two clocks after burst_done is deasserted. Any further commands to the DDR2 SDRAM controller should be given after user_cmd_ack is deasserted. The DDR2 SDRAM controller does the precharge automatically after each read or write command.

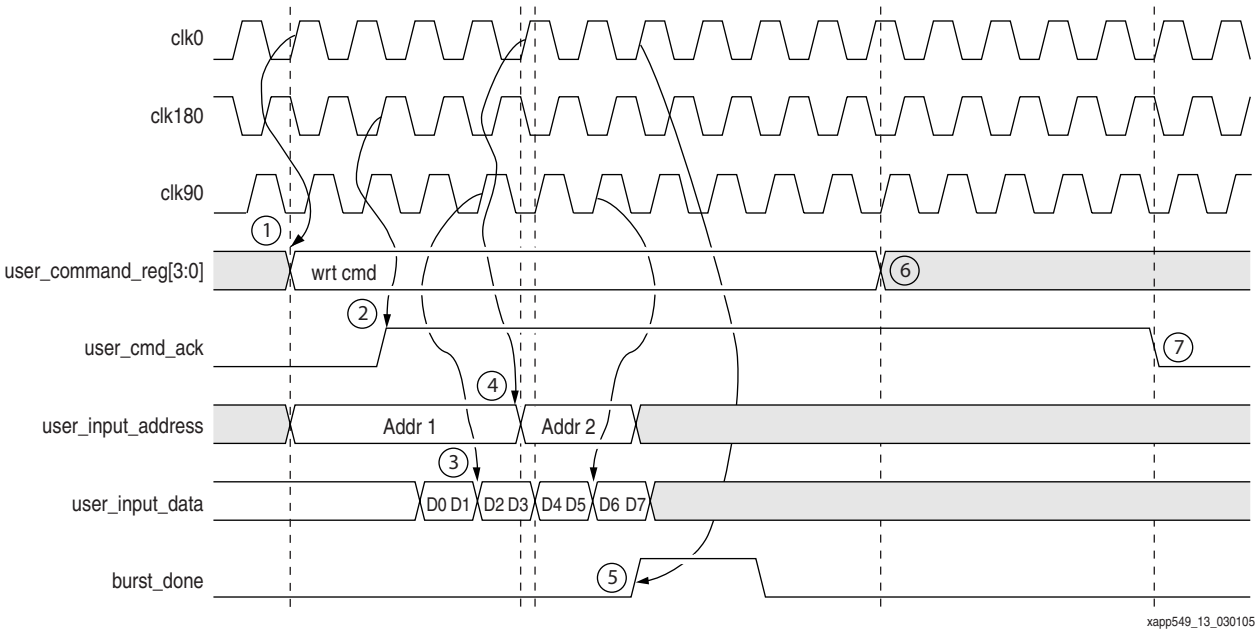

Figure 4: **DDR2 SDRAM Memory Write Burst, Burst Length of Four, Two Bursts**

1. A memory write is initiated by issuing a write command to the DDR2 SDRAM controller. The write command must be asserted on a rising edge of clk0.
2. The DDR2 SDRAM controller acknowledges the write command by asserting the user_cmd_ack signal on a rising edge of clk180.
3. The first user_input_address should be placed along with the command. The input data is asserted with the clk90 signal after the user_cmd_ack signal is asserted.
4. Two and half clocks after the user_cmd_ack signal assertion, the next memory address is placed on user_input_address [21:0]. The user_input_address signal is asserted on a rising edge of the clk0. All subsequent addresses are asserted on alternate positive edges of clk0.

5. To terminate the write burst, burst_done is asserted after two clocks of the last user_input_address. The burst_done signal is asserted for two clock cycles.
6. The user command is deasserted two cycles after burst_done is deasserted.
7. The controller deasserts the user_cmd_ack signal any time after the user command is deasserted.
8. The next command should be be given only after user_cmd_ack is deasserted. Back to back write operations are supported only within the same bank and row.

DDR2 SDRAM Memory Read

Figure 5 shows a memory read timing diagram for two successive bursts with a burst length of four. A memory read is initiated by sending a read command to the DDR2 SDRAM controller.

The read command flow is similar to the write command. A read command and the first burst read address is asserted on the rising edge of clk0. The first user read address should be asserted along with the read command. The DDR2 SDRAM controller asserts the user_cmd_ack signal in response to the read command on the rising edge of clk180. After two and half clock cycles of user_cmd_ack, the next memory burst read address is placed on user_input_address[addwidth:0]. The user_input_address signal is asserted on the rising edge of clk0. Subsequent read addresses are asserted on alternate positive edges of clk0.

The data read from the DDR2 SDRAM memory is available on user_output_data, which is asserted with clk90. The data on user_output_data is valid only when user_data_valid signal is asserted. As the DDR SDRAM data is converted to SDR data, the width of this bus is 2n, where n is the data width of the DDR2 SDRAM memory. For a read burst length of four, the DDR2 SDRAM controller outputs only two data with each user address, each of 2n width of DDR2 SDRAM memory. To terminate the read burst, a burst_done signal is asserted for two clock cycles on the rising edge of clk0. The burst_done signal is asserted after the last memory address. Any further commands to the DDR2 SDRAM controller should be given after user_cmd_ack is deasserted.

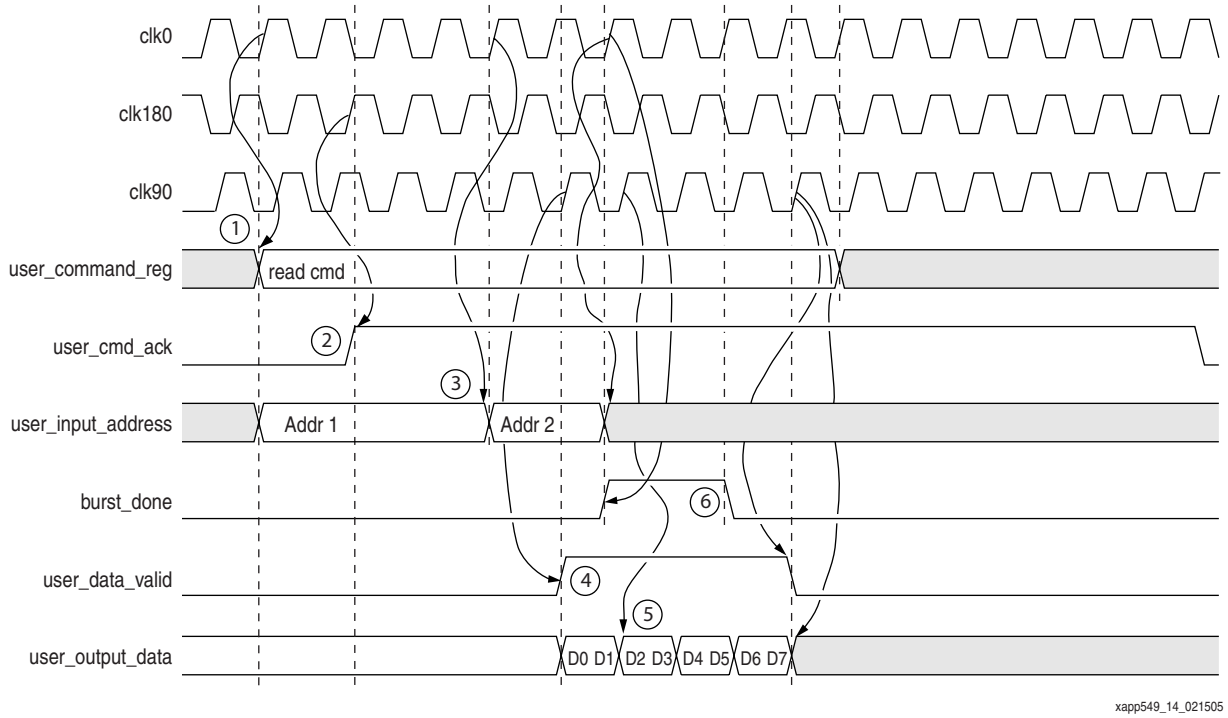

Figure 5: **DDR2 SDRAM Memory Read Burst Length of Four**

The read command flow is similar to the write command flow:

1. A memory read is initiated by issuing a read command to the DDR2 SDRAM controller. The read command is accepted on a rising edge of the clk0.
2. The first read address should be placed along with the read command. In response to the read command, the DDR2 SDRAM controller asserts the user_cmd_ack signal on a rising edge of clk180.
3. Two and half clocks after user_cmd_ack, the next memory read address is placed on user_input_address [addwidth:0]. The user_input_address signal is then accepted on the rising edge of clk0. All subsequent memory read addresses are asserted on alternate positive edges of clk0.
4. The data on user_output_data is valid only when the user_data_valid signal is asserted.
5. The data read from the DDR2 SDRAM memory is available on user_output_data, which is asserted with clk90. Since the DDR SDRAM data is converted to SDR data, the width of this bus is 2n, where n is the data width of the DDR2 SDRAM memories. For a read burst length of four, the DDR2 SDRAM controller outputs only two data words with each user address.
6. To terminate the read burst, burst_done is asserted for two clocks on the rising edge of clk0. The burst_done signal is asserted after two clocks of the last memory address.
7. The user command is deasserted two cycles after burst_done is deasserted.
8. The controller deasserts the user_cmd_ack signal any time after the user command is deasserted.
9. Any further commands to the DDR2 SDRAM controller should be given after user_cmd_ack is deasserted. Back to back read operations are supported only within the same bank and row. Around 20 clock cycles pass between the time a read command is asserted on the user interface and the time data becomes available on the user interface.

DDR2 SDRAM Memory Auto_Refresh

The DDR2 SDRAM controller does not support memory refresh on its own and must periodically be provided with an auto_refresh command. The auto_refresh command is asserted with clk0 for one clock period. The ar_done signal is asserted by the DDR2 SDRAM controller upon completion of the auto_refresh command. The ar_done signal is asserted with clk180.

The next command can be given any time after ar_done is asserted. Figure 6 shows a timing diagram for the auto_refresh command.

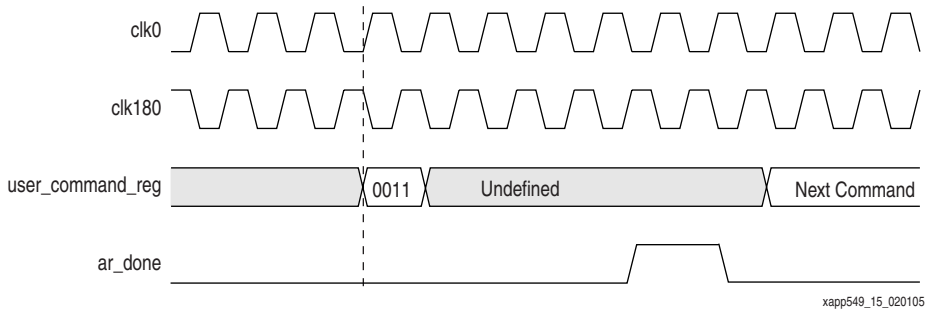

Figure 6: **DDR2 SDRAM Auto Refresh Timing Diagram**

Physical Layer and Delay Calibration

The physical layer for DDR2 SDRAM is similar to the DDR SDRAM physical layer described in XAPP678 and XAPP688. The delay calibration technique described in those application notes is also used in the DDR2 SDRAM interface.

Timing Calculations

Write Timing

Table 4: **Write Data**

Parameter	Value (ps)	Leading Edge Uncertainities	Trailing Edge Uncertainities	Meaning
Tclock	5000			Clock period
Tclock_phase	2500			Clock phase
Tdcd	250			Duty cycle distortion of clock to memory
Tdata_period	2250			Total data period, Tclock_phase-Tdcd
Tclock_skew	50	50	50	Minimal skew, since the right/left sides are used and the bits are close together
Tpackage_skew	65	65	65	Skew due to package pins and board layout (This can be reduced further with tighter layout.)
Tsetup	350	350	0	Setup and hold
Thold	350	0	350	
Tphase_offset_error	140	140	140	Offset error between different clocks from the same DCM
Tjitter	0	0	0	The same DCM is used to generate the clock and data; hence, they jitter together.
Total uncertainties	955	605	605	Worst case for leading and trailing can never happen simultaneously.
Window	1040	605	1645	Total worst case window is 1040ps.

Read Timing

Table 5: **Read Data**

Parameter	Value (ps)	Leading Edge Uncertainties	Trailing Edge Uncertainties	Meaning
Tclock	5000			Clock period
Tphase	2500			Clock phase
Tmem_dcd	250			Duty cycle distortion from memory DLL
Tdata_period	2250			Total data period, Tphase - Tmem_dcd
Tdqsq	300	300	0	Strobe to data distortion
Tpackage_skew	65	65	65	This parameter depends on the exact package. Since the 8 data bits are close together, skew is less than this.
Tsetup	210	210	0	Setup time from Virtex-II Pro data sheet for XC2VP -6 part
Thold	-40	0	-40	Hold time from V2 Pro data sheet
Tjitter	100	0	0	Data and strobe jitter together, since they are generated off of the same clock
Tlocal_clock_line	25	25	25	Observed skew is lower than this value since loading is light and all bits are close together

Table 5: **Read Data *(Continued)***

Parameter	Value (ps)	Leading Edge Uncertainties	Trailing Edge Uncertainties	Meaning
Tpcb_layout_skew	50	50	50	Skew between data lines on the board
Tqhs	400	0	400	Hold skew factor for DQ
Uncertainties	1140	650	500	
Window	1100	650	1750	Worst case window of 1100 ps

Notes:

1. Reference for Tdqsq and Tqhs are from Micron data sheet for MT47H64M4FT-37E, Rev C, 05/04 EN.
2. Reference for Virtex-II Pro timing is XC2VP20FF1152 -7, Speeds file version 1.85

Address and Command Timing

Table 6: **Address and Command Data**

Parameter	Value (ps)	Leading Edge Uncertainties	Trailing Edge Uncertainties	Meaning
Tclock	5000			Clock period
Tclock_skew	50	50	50	Minimal skew, since right/left sides are used and the bits are close together
Tpackage_skew	65	65	65	Using same bank reduces the package skew
Tsetup	500	500	0	Setup time from memory data sheet
Thold	500	0	500	Hold time from memory data sheet
Tphase_offset_error	140	140	140	Offset between different phases of the clock
Tduty_cycle_distortion	0	0	0	Duty cycle distortion does not apply
Tjitter	0	0	0	Since the clock and address are generated using the same clock, the same jitter exists in both; hence, it does not need to be included.
Total uncertainties		755	755	
Command window	3490	755	4245	Worst case window of 3490 ps

Performance and Characterization

A 72-bit design was characterized using a Virtex-II Pro ML367 board. The results are shown in Table 7.

Table 7: **Characterization Results**

Part	Temperature	Voltage	Max Frequency
XC2VP20FF1152 -7	Room	nominal	250 MHz
		-5%	240 MHz
	0°C		
		-5%	240 MHz
	70°C		
		-5%	235 MHz

Table 8 gives the maximum supported frequency per speed grade.

Table 8: **Supported Performance per Speed Grade**

Speed Grade	Max Frequency (MHz)
-6	200
-7	230

Reference Design

The reference design for the DDR2 SDRAM memory controller is integrated with the Memory Interface Generator (mig007) tool. For the latest version of the design, download the mig007 tool on the Xilinx website at:
http://www.xilinx.com/products/design_resources/mem_corner/index.htm

References

Xilinx Application Notes:

- XAPP678c: “Data Capture Technique Using CLB Flip-Flops” (available under click license)
- XAPP688: “Creating High-Speed Memory Interfaces With Virtex-II and Virtex-II Pro FPGAs”
- XAPP688c: “Creating High-Speed Memory Interfaces With Virtex-II and Virtex-II Pro FPGAs” (available under click-through license)

Xilinx Reference Designs:

- http://www.xilinx.com/memory

Micron Data Sheet MT47H16M16FG-37E, available online at:
http://download.micron.com/pdf/datasheets/dram/ddr2/256MbDDR2.pdf

Conclusion

It is possible to implement a high-performance DDR2 SDRAM memory interface using Virtex-II Pro FPGAs.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/10/04	1.0	Initial Xilinx release.
03/01/05	1.1	Revised Figure 3 thru Figure 6 and associated text.
04/30/07	1.2	• Added section “Reference Design” pointing the user to the mig007 tool. • Added Table 8. • Corrected URL for Micron data sheet.