

## Introduction

Over the years, as applications have become more demanding, systems have increasingly resorted to external memory as a way to boost performance while reducing cost. Single data rate (SDR) memories gave way to double data rate (DDR) memories as system designers continually sought solutions to boost system performance without increasing system complexity and cost.

DDR2 SDRAM is the next generation of DDR SDRAM technology, with improvements that include lower power consumption, higher data bandwidth, enhanced signal quality, and on-die termination schemes. DDR2 SDRAM brings higher memory performance to a broad range of applications, such as PCs, embedded processor systems, image processing, storage, communications, and networking.

DDR SDRAM is currently the popular memory of choice. Designers looking to save system power are moving toward using DDR2 SDRAM, which uses a lower 1.8-V I/O voltage compared to the DDR SDRAM I/O voltage of 2.5 V. Table 1 shows the DDR and DDR2 SDRAM interface support in Cyclone™ II devices.

<i>Table 1. DDR &amp; DDR2 SDRAM Support in Cyclone II Devices</i> <i>Notes (1), (2)</i>				
DDR Memory Type	I/O Standard	Maximum Clock Rate		
		-6 Speed Grade Commercial Wire-Bond	-7 Speed Grade Commercial Wire-Bond	-8 Speed Grade Commercial/Industrial Wire-Bond
DDR SDRAM (3)	SSTL-2	167 MHz	150 MHz	133 MHz (4)
DDR2 SDRAM (3)	SSTL-18	167 MHz	150 MHz	133 MHz (4)

**Notes to Table 1:**

- (1) These specifications are for SSTL-2 and SSTL-18 class 1 I/O standards. Altera recommends using these I/O standards for interfacing Cyclone II devices with DDR and DDR2 SDRAM memory.
- (2) This analysis is based on the EP2C70F896. Ensure you perform a timing analysis for your chosen FPGA.
- (3) Specifications are based on the use of clock delay control circuitry.
- (4) The Q240 packages are only available in the -8 commercial speed grade, and the maximum operating frequency is 125 MHz.

Cyclone II devices can interface with DDR and DDR2 SDRAM in device or module configurations up to 167 MHz/333 Mbps. This application note describes the DDR and DDR2 SDRAM interfacing in Cyclone II devices and provides detailed timing analysis.



Use this application note together with the *External Memory Interfaces* chapter of the *Cyclone II Device Handbook*.

## DDR & DDR2 SDRAM Overview

This section provides information on DDR and DDR2 SDRAM.

### DDR SDRAM

DDR SDRAM is a 2n prefetch architecture with two data transfers per clock cycle. In the 2n prefetch architecture, two data words are fetched from the memory array during a single read command. DDR SDRAM uses a strobe signal (DQS) that is associated with a group of data pins (DQ) for read and write operations. Both the DQS and DQ ports are bidirectional. Address ports are shared for write and read operations.

Write and read operations are sent in bursts, and DDR SDRAM supports burst lengths of two, four, and eight. This means that you need to provide 2, 4, or 8 groups of data for each write transaction, and you receive two, four, or eight of data for each read transaction. The interval between the time the read command is clocked into the memory and the time the data is presented at the memory pins is called the column address strobe (CAS) latency. DDR SDRAM supports CAS latencies of 2, 2.5, and 3, depending on the operating frequency. Both the burst length and CAS latency are set in the DDR SDRAM mode register.

DDR SDRAM specifies the use of the SSTL-2 I/O standard. Each DDR SDRAM device is divided into four banks, and each bank has a fixed number of rows and columns and can hold between 64 Mb to 1 Gb of data. Only one row per bank can be accessed at one time. The `ACTIVE` command opens a row and the `PRECHARGE` command closes a row.

### DDR2 SDRAM

DDR2 SDRAM is the second generation of the DDR SDRAM memory standard. DDR2 SDRAM is a 4n prefetch architecture with two data transfers per clock cycle. In the 4n prefetch architecture, four data words are fetched from the memory array during a single read command. DDR2 SDRAM uses a DQS that is associated with a group DQs for read and write operations. Both the DQS and DQ ports are bidirectional. Address ports are shared for write and read operations.

Although DDR2 SDRAM devices can use the optional differential strobes (DQS and DQS#), Cyclone II devices do not support this mode. Cyclone II devices only use the DQS signal to read from and write to the DDR2 SDRAM device.

Write and read operations are sent in bursts, and DDR2 SDRAM supports burst lengths of 4 and 8. This means that you need to provide 4 or 8 clusters of data for each write transaction, and you receive 4 or 8 clusters of data for each read transaction. The interval between the time the read command is clocked into the memory and the time the data is presented at the memory pins is called the column address strobe (CAS) latency. DDR2 SDRAM supports CAS latencies of 3 and 4, depending on the operating frequency. DDR2 SDRAM does not support half-clock latencies. Both the burst length and CAS latency are set in the DDR2 SDRAM mode register.

Besides CAS latency, DDR2 SDRAM offers posted CAS additive latencies of 0, 1, 2, 3, and 4. During the read operation, instead of getting the data CAS latency after the read command is issued, you can send the read command earlier with a particular additive latency value. The additive latency setting specifies how many clock cycles earlier you should send the read command.

The memory device holds the read command for the number of cycles specified by the additive latency. The read latency is the sum of the additive latency and the CAS latency. During the write operation, instead of sending the data one clock cycle after the write command is issued, you can send the write command earlier with a particular additive latency value. The additive latency setting specifies how many clock cycles earlier you send the write command. The memory device holds the write command for the number of cycles specified by the additive latency. The write latency is one clock cycle less than the read latency.

DDR2 SDRAM specifies the use of the SSTL-18 I/O standard and can hold between 64 Mb to 4 Gb of data. DDR2 SDRAM devices with capacities up to 512 Mb are divided into four banks, and devices with capacities between 1 and 4 Gb are divided into eight banks. Only one row per bank can be accessed at one time for devices with four banks. Only four banks can be accessed at one time for devices with eight banks. The ACTIVE command opens a row and the PRECHARGE command closes a row.

DDR2 SDRAM uses a delay-locked loop (DLL) inside the memory device to edge-align the DQ and DQS signals with respect to CK. The DLL in the memory devices is turned on for normal operation and is turned off for debugging purposes. All timing analyses done in this document assume that the DLL in the memory devices is on.

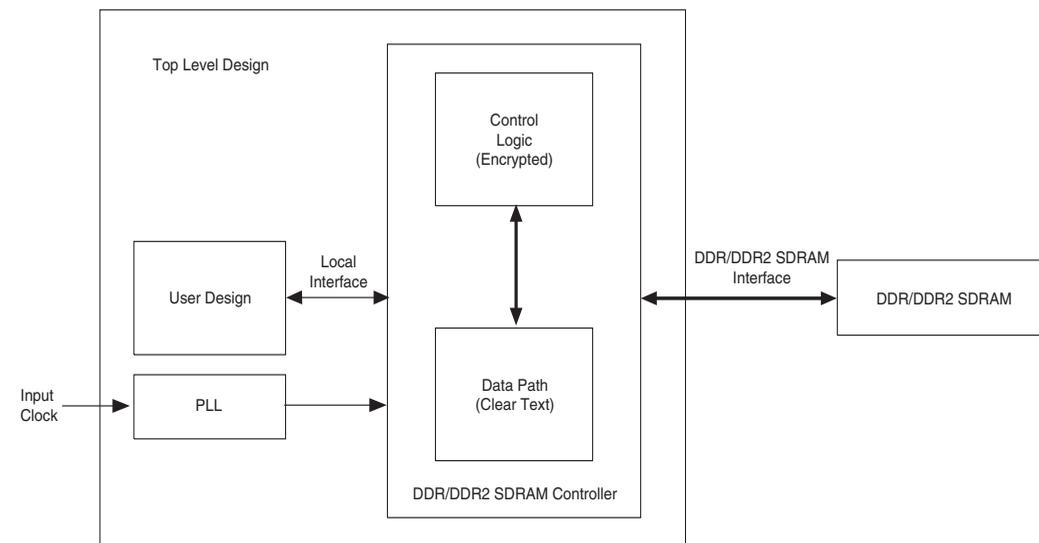
DDR2 SDRAM devices also have adjustable data-output drive strength, so Altera recommends that you use the highest drive strength the memory device can support for maximum performance. DDR2 SDRAM devices also offer parallel on-die termination, which has an effective resistance of either 75 or 150  $\Omega$ .

### Differences Between DDR & DDR2 SDRAM

DDR2 SDRAM offers some key improvements over DDR SDRAM. DDR2 SDRAM has on-die termination to improve signal integrity and timing margin. DDR2 SDRAM has a feature called posted CAS additive latency, which improves the bus efficiency over the DDR SDRAM. DDR2 SDRAM consumes less power since it uses the SSTL-18 I/O standard instead of the SSTL-2 I/O standard that DDR SDRAM uses.

## Interface Description

The following section provides a detailed description of the interface signals between the FPGA and the DDR/DDR2 SDRAM devices, how the FPGA pins should be configured to meet the DDR/DDR2 SDRAM electrical and timing requirements, and lists the number of DQS/DQ pins available in the FPGA. In addition, this section also describes the architecture of the interface between the FPGA and the DDR/DDR2 SDRAM memory. Understanding how complicated the interface can be, Altera offers a complete solution that creates the memory controller within minutes. The Altera® DDR/DDR2 SDRAM Controller MegaCore® allows you to use the Altera recommended data path whether or not you are using the Altera controller logic. When you use this recommended data path, you benefit from the knowledge that this is a proven working system and the DDR/DDR2 SDRAM Controller MegaCore constrains your interface pins and data path logic for optimal operation. [Figure 1](#) shows the block diagram for the FPGA to DDR/DDR2 SDRAM interface.

**Figure 1. DDR/DDR2 SDRAM Controller System Level Block Diagram**

## Interface Signals

Table 2 shows the DDR SDRAM interface pins and how to connect them to Cyclone II devices.

<b>Pins</b>	<b>Description</b>	<b>Cyclone II Pin Utilization</b>
DQ	Bidirectional read/write data	DQ
DQS	Bidirectional read/write data strobe	DQS
CK	Memory clock	User I/O pin
CK#	Memory clock	User I/O pin
DM	Optional write data mask, edge-aligned to DQ during write	DM
All other	Addresses and commands	User I/O pin

Table 3 shows the DDR2 SDRAM interface pins and how to connect them to Cyclone II devices.

<b>Pins</b>	<b>Description</b>	<b>Cyclone II Pin Utilization</b>
DQ	Bidirectional read/write data	DQ
DQS	Bidirectional read/write data strobe	DQS
DQS# (1)	Optional bidirectional differential read/write data strobe	N/A
CK	Memory clock	User I/O pin
CK#	Memory clock	User I/O pin
DM	Optional write data mask, edge-aligned to DQ during write	DM
All other	Addresses and commands	User I/O pin

**Note to Table 3:**

- (1) The DQS# signal in DDR2 SDRAM devices is optional. Cyclone II devices do not use DQS# pins when interfacing with DDR2 SDRAM.

This section provides a description of the clock, control, address, and data signals on DDR and DDR2 SDRAM devices.

### Strobes & Data Signals

Both DQ and DQS signals are bidirectional (the same signals are used for both read and write). The DQS# pins in DDR2 SDRAM are not used in the Cyclone II DDR2 SDRAM interface. A group of DQ pins is associated with one DQS pin.

In  $\times 8$  and  $\times 16$  DDR SDRAM devices, one DQS pin is associated with eight DQ pins. Cyclone II devices support both  $\times 8$  and  $\times 16$  DDR SDRAM. Use the DQS pins and their associated DQ pins listed in the Cyclone II pin tables when interfacing with DDR and DDR2 SDRAM from Cyclone II I/O banks.

Table 4 shows the number of DQS/DQ groups supported in Cyclone II devices.

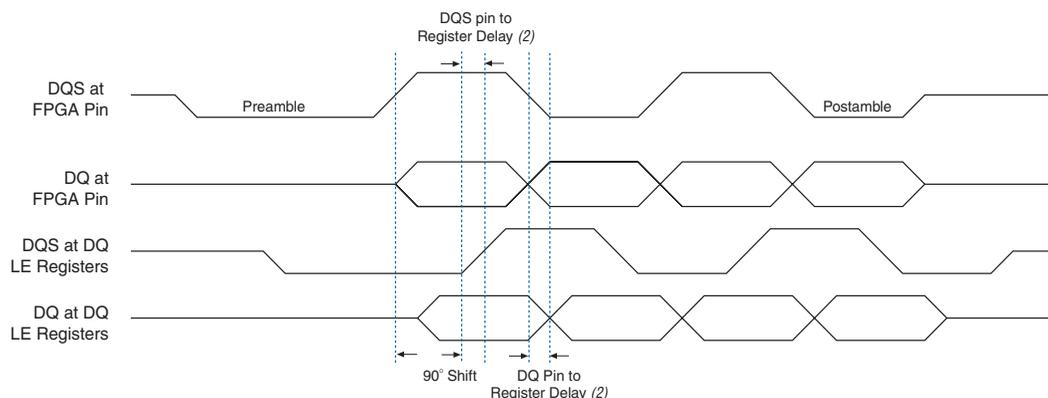
Device	Package	Number of ×8 Groups	Number of ×16 Groups
EP2C5	144-pin TQFP (1)	3	0
	208-pin PQFP	7 (2)	3
	256-pin FineLine BGA	8 (2)	4
EP2C8	144-pin TQFP (1)	3	0
	208-pin PQFP	7 (2)	3
	256-pin FineLine BGA	8 (2)	4
EP2C20	240-pin PQFP	8	4
	256-pin FineLine BGA	8	4
	484-pin FineLine BGA	16 (3)	8 (4)
EP2C35	484-pin FineLine BGA	16 (3)	8 (4)
	672-pin FineLine BGA	20 (3)	8 (4)
EP2C50	484-pin FineLine BGA	16 (3)	8 (4)
	672-pin FineLine BGA	20 (3)	8 (4)
EP2C70	672-pin FineLine BGA	20 (3)	8 (4)
	896-pin FineLine BGA	20 (3)	8 (4)

**Notes to Table 4:**

- (1) EP2C5 and EP2C8 devices in the 144-pin TQFP package do not have any DQ pin groups in I/O bank 1.
- (2) Because of available clock resources, only a total of 6 DQ/DQS groups can be implemented.
- (3) Because of available clock resources, only a total of 14 DQ/DQS groups can be implemented.
- (4) Because of available clock resources, only a total of 7 DQ/DQS groups can be implemented.

During a read from the memory, the DQS's are edge-aligned with the DQ. During a write to the memory, the Cyclone II device transmits the DQS signals center-aligned relative to the DQ signals. Figures 2 and 3 illustrate the DQ and DQS relationships during a DDR and DDR2 SDRAM read and write. The memory controller on the device center-aligns the DQS signal during a write and shifts the DQS signal during a read so that the DQ and DQS signals are center-aligned at the capture register. The Cyclone II device uses a phase-locked loop (PLL) to center-align the DQS signal with respect to the DQ signals during writes and uses dedicated DQS programmable delay-chain circuitry to shift the incoming DQS signal during reads.

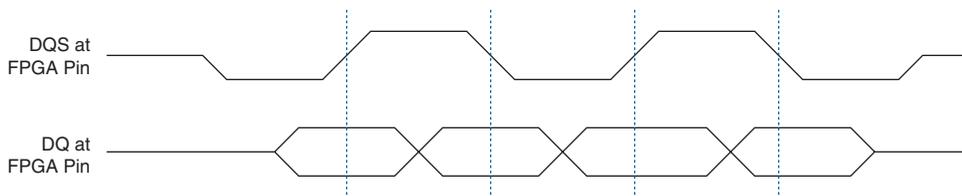
**Figure 2. DQ & DQS Relationship During DDR & DDR2 SDRAM Read in Burst of 4 Mode** *Note (1)*



**Notes to Figure 2:**

- (1) DDR SDRAM supports burst lengths of 2.
- (2) The DQS and DQ pins to the register delay are not the same.

**Figure 3. DQ & DQS Relationship During DDR & DDR2 SDRAM Write in Burst of 4 Mode** *Note (1)*



**Note to Figure 3:**

- (1) DDR SDRAM supports burst lengths of 2.

The setup ( $t_{DS}$ ) and hold ( $t_{DH}$ ) times for the DQ and DM pins at the memory during a write are relative to the edges of DQS write signals and not the CK and CK# clocks. The memory setup and hold times are equal ( $t_{DS} = t_{DH}$ ) and are typically 0.45 ns for a 167-MHz DDR SDRAM device. Unlike the DDR SDRAM devices, the DDR2 SDRAM memory setup and hold times are not necessarily equal depending on the input data slew rate and the usage of the differential DQS# signal. Cyclone II devices do not use the differential DQS# signal from the DDR2 SDRAM memory devices for clocking, so  $t_{DS}$  is 0.445 ns and  $t_{DH}$  is 0.385 ns for 167-MHz DDR2 SDRAM devices assuming 1 V/ns input slew rate for both DQS and DQ. Refer to the DDR2 SDRAM data sheet for information on the memory's setup to hold time.

The DDR and DDR2 SDRAM  $t_{DQSS}$  timing is the amount of time between when the memory detects the write command to the first DQS transition. The DQS signal is normally generated on the positive edge of system clock to meet the  $t_{DQSS}$  requirement. The DQ and DM signals are clocked using a  $-90^\circ$  shifted clock from the system clock. The edges of the DQS signal are centered on the DQ and DM signals when they arrive at the DDR SDRAM device.

To minimize the skew between the arrival times of these signals, the DQS, DQ, and DM board trace lengths should be similar.

## Clock Signals

DDR and DDR2 SDRAM devices use the CK and CK# signals to clock commands and addresses into the memory. The memory also uses these clock signals to generate the DQS signal during a read via a DLL inside the memory. The skew between CK or CK# and the SDRAM-generated DQS signal is specified as  $t_{DQSCK}$  in the DDR and DDR2 SDRAM data sheet.

The DDR and DDR2 SDRAM  $t_{DQSS}$  write requirement states that on writes, the positive edge of the DQS signal must be within  $\pm 25\%$  ( $\pm 90^\circ$ ) of the positive edge of the DDR and DDR2 SDRAM clock input. Therefore, you should use the logic element (LE) registers in the FPGA to generate the CK and CK# signals. This helps match the CK and CK# signals with the DQ signal and reduces any process, voltage, temperature variations, and skew between CK or CK# and DQ signals.

## DM Signals

DDR and DDR2 SDRAM use the DM pins during the write operation. Driving the DM pin low indicates that the write is valid. Driving the DM pin high results in the memory masking the DQ signals. You can use any of the I/O pins in the same bank as the DQS/DQ pins to generate the DM signal.

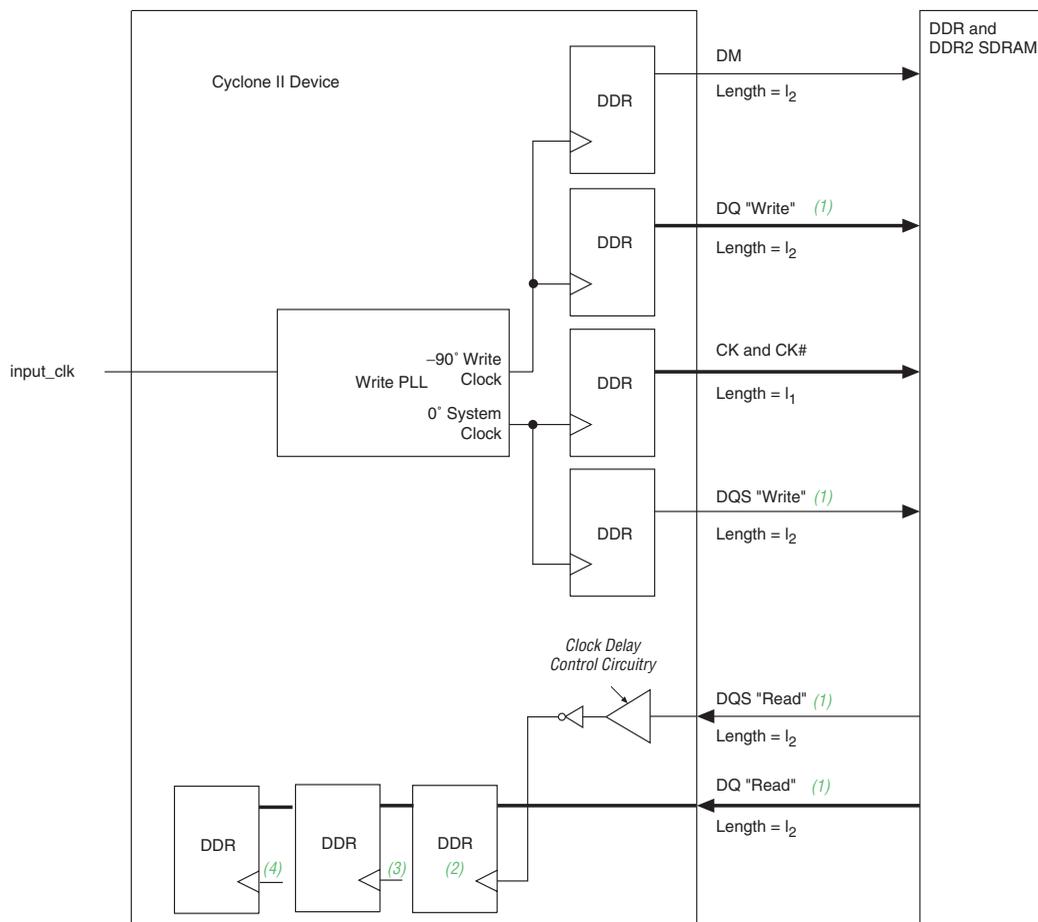
The timing requirements for DM signals at the DDR and DDR2 SDRAM are identical to those of the DQ output signals. Similarly, the DM signals are clocked out by the  $-90^\circ$  shifted clock.

### Commands & Addresses Signals

Commands and addresses in DDR and DDR2 SDRAM devices are clocked into the memory using the CK and CK# signals at a single data rate using only the positive clock edge. DDR and DDR2 SDRAM devices have 12 to 14 address pins, depending on the device capacity. The address pins are multiplexed, so two clock cycles are required to send the row, column, and bank addresses. The CS, RAS, CAS, and WE pins are DDR and DDR2 SDRAM command pins.

The DDR and DDR2 SDRAM address and command inputs both require the same setup and hold times with respect to the DDR and DDR2 SDRAM clocks. The Cyclone II device address and command signals change at the same time as the DQS write signal because they are both generated from the system clock. The positive edge of the DDR and DDR2 SDRAM clocks, CK, is aligned with DQS to satisfy  $t_{DQSS}$ . If the command and address outputs are generated on the clock's positive edge, they may not meet the setup and hold time requirements ([Figure 9 on page 29](#)). Therefore, you should use the negative edge of the system clock for the commands and addresses to the DDR and DDR2 SDRAM. You can use any of the FPGA's I/O pins for the commands and addresses.

Figure 4. DDR &amp; DDR2 SDRAM with Clock Delay Control Circuitry

**Notes to Figure 4:**

- (1) DQ and DQS signals are bidirectional. One DQS signal is associated with a group of DQ signals.
- (2) Although there are three LE registers for capturing the read data, this figure only shows one register.
- (3) The clock to the resynchronization register is either from the system clock, write clock, and extra clock output from the write PLL, or from the read PLL.
- (4) The clock to this register can either be the system clock or another clock output of the write PLL. If the design needs another write PLL clock output, another register is needed to transfer the data back to the system clock domain.

## Interface Architecture

In the Cyclone II device, the read-side implementation for DDR/DDR2 SDRAM interface architecture in the Cyclone II device uses the clock delay control circuitry to phase shift the DQS signals and center-align the strobe signal with the read data (DQ).

The write-side implementation where a write-side PLL outputs two clocks that generates the write data and center-aligned the write clock using dedicated double data rate input/output (DDIO) circuits. This implementation results in matched propagation delays for clock and data signals from the FPGA to the DDR2 SDRAM, minimizing skew.

## Data Path Architecture Using Clock Delay Control Circuitry

The DDR/DDR2 SDRAM interface implementation using clock delay control circuitry uses the following:

- A write-side PLL to generate CK and CK# system clocks and clock out address, command, strobe, and data signals.
- A read-side clock delay control circuitry to register read data from the memory using DQS signals.

The clock delay control circuitry is available on each DQS pin and it shifts the DQS signal to center-align the signal with the DQ signal at the LE registers, ensuring the data gets latched at the LE registers. The shifted DQS signals drive the global clock network, which in turn clocks the DQ signals on the internal LE registers. The DQS signal is inverted before going to the DQ LE clock ports, as described in the *External Memory Interfaces* chapter of the *Cyclone II Device Family Handbook*.

Figure 4 shows a summary of how Cyclone II devices generate the DQ, DQS, CK, and CK# signals. The write PLL generates the system clock and the  $-90^\circ$  shifted clock (write clock). The write PLL's input clock can either be the same or a different frequency as the DDR or DDR2 SDRAM frequency of operation. The system clock and write clock have the same frequency as the DQS frequency. The write clock is  $-90^\circ$  shifted from the system clock.



See “Round Trip Delay” on page 32 for more information on how the PVT variation in the clock delay control circuitry affects the design.

## Altera Memory Controller IP

Altera Corporation has a DDR/DDR2 SDRAM Controller MegaCore function that allows you to instantiate a simplified interface to industry-standard DDR/DDR2 SDRAM memory. The DDR/DDR2 SDRAM Controller initializes the memory devices, manages SDRAM banks, and keeps devices refreshed at appropriate intervals. The MegaCore function translates read and write requests from the local interface into all the necessary SDRAM command signals. The DDR/DDR2 SDRAM Controller contains encrypted control logic as well as an open source data path that you can use in your design without a license. Download this MegaCore function whether you plan to use the Altera DDR/DDR2 SDRAM controller or not to get the open source data path, open source DQS postamble logic, placement constraints, and timing margin analysis.

The MegaCore function is accessible through the MegaWizard® Plug-In Manager. When you parameterize your custom DDR/DDR2 SDRAM interface, the IP Toolbench automatically decides the best phase-shift and FPGA settings to give you the best margin for your DDR/DDR2 SDRAM interface. It then generates an example instance that instantiates a PLL, an example driver, and your DDR/DDR2 SDRAM Controller custom variation as shown in [Figure 1 on page 5](#).

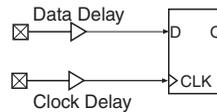
## Interface Timing Analysis

When designing an external memory interface for your FPGA, you have to analyze timing margins for several paths. All memory interfaces require analysis of the read capture, write capture, and address/command timing paths. Additionally, some interfaces might require analysis of the resynchronization timing paths and other memory-specific paths (such as postamble timing). This application note describes Altera's recommended timing methodology using read and write capture timing paths as examples. You should use this methodology for analyzing timing for all applicable timing paths (including address/command, resynchronization, postamble, etc.). To ensure successful operation, the Altera DDR/DDR2 SDRAM Controller MegaCore performs timing analysis on the read capture, postamble paths, and resynchronization. While these analyses account for all FPGA related timing effects, you should design in adequate margin to account for board-level effects. This section analyzes the read and write capture timing margins for a DDR and DDR2 SDRAM interface with a Cyclone II device. The timing analysis methodology is illustrated using the EP2C70F896C6 FPGA interfacing with a Micron MT9HTF3272AY-40E DIMM. However, you should use the same methodology to analyze timing for your preferred FPGA and memory device.

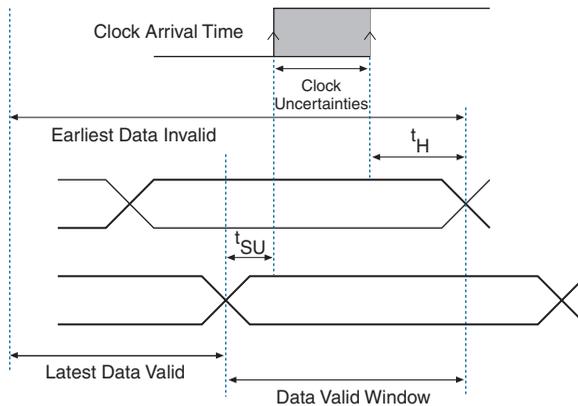
## Methodology Overview

Timing paths are analyzed by considering the data and clock arrival times at the destination register. In Figures 5 and 6, the setup margin is defined as the time between “earliest clock arrival time” and “latest valid data arrival time” at the register ports. Similarly, hold margin is defined as the time between “earliest invalid data arrival time” and the “latest clock arrival time” at the register ports. These arrival times are calculated based on propagation delay information with respect to a common reference point (such as a DQS edge or system clock edge).

**Figure 5. Simplified Block Diagram for Timing Analysis**



**Figure 6. Data Valid Window Timing Waveform**



### *FPGA Timing Information*

Since you expect your design to work under all conditions, timing margins should be evaluated at all process, voltage, and temperature (PVT) conditions. To facilitate this, Altera provides two device timing models in the Quartus® II software: slow corner model and fast corner model.

- The slow corner model provides timing delays between two nodes within the FPGA with slow silicon, high temperature, and low voltage. In other words, the model provides the slowest possible delay for that timing path on any device for that particular speed grade.
- The fast corner model provides timing delays between two nodes within the FPGA with fast silicon, low temperature, and high voltage. In other words, the model provides the fastest possible delay for that timing path on any device. Note that while almost all FPGA timing delays and uncertainties are modeled in the Quartus II software, a limited number of uncertainties that cannot be modeled are published in the FPGA handbook for use in margin calculations. Some examples include clock jitter on PLL outputs. These timing uncertainties or adder terms, when used in conjunction with Quartus II software reported timing data, provide the most accurate device timing information. The following analysis details the use of these timing adder terms.

### **Read Timing Margins**

During read operations, the DDR/DDR2 SDRAM provides a clock strobe (DQS) that is edge-aligned with the data bus (DQ). The memory controller (in the FPGA) is required to shift the clock edge to the center of the data valid window and capture the DQ input data. [Figure 7](#) illustrates the timing relationship between the DQS and DQ signals during a read operation. [Figure 7](#) shows a more detailed picture of the Cyclone II device read data path for x8 mode. The DQS signal goes to the clock delay control circuit and is shifted to be center aligned with the incoming data. The shifted DQS signal is then routed to the global clock bus. The DQS global clock bus signal is then inverted before it clocks the DQ at the LE registers. The outputs from the LE input registers then go to the resynchronization registers. The `resynch_clk` signal clocks the resynchronization register. The `resynch_clk` can come from the system clock, the write clock, or the write PLL clock.



Armed with these memory timing parameters, we can calculate the data valid window at the memory to be equal to  $t_{HP} - t_{QHS} - t_{DQSQ} = 1900$  ps. Assuming the board trace length variations amongst all DQ and DQS traces are not more than  $\pm 20$  ps, the data valid window present at the FPGA input pins is 1860 ps.

### **FPGA Timing Parameters**

FPGA timing parameters are obtained from two sources: the Quartus II timing analyzer and the *Cyclone II Device Family Data Sheet*. While the former provides all clock and data propagation delays, the data sheet specifies all clock uncertainties and skew adder terms.

The timing analysis methodology outlined earlier suggests we calculate the earliest and latest arrival times for clock and data. Let us begin with the clock (DQS).

The Cyclone II features dedicated clock delay control circuitry for the DQS pins of the device that has 63 delay settings that center-aligns the DQS edge with respect the DQ input signals.

Analyze timing with a 0 delay setting on the DQS strobe, knowing that the phase shift can always be adjusted at the end of this timing analysis for balanced setup and hold margins on the read capture register.

The clock delay control circuitry uses static delay chains to provide the delay shift. This means you must account for phase-shift error on the DQS signal.

After encountering the phase-shift circuitry, the DQS travels on a dedicated local clock bus to the DQ capture registers. The fanout of this local clock bus could range from  $\times 4$  to  $\times 18$ . While the Quartus II software provides clock propagation delays to each of these DQ register clock ports, unmodeled uncertainties are accounted for with the  $t_{DQS\_SKEW\_ADDER}$  adder term listed in the data sheet. For the  $\times 8$  mode used by this Micron DDR2 SDRAM device, the skew adder is 77.5 ps.

To obtain Quartus II timing data for the target device, you should instantiate and compile the DDR2 SDRAM Controller MegaCore. If you are using your own controller logic, you should instantiate the clear-text DDR2 data path instead to obtain timing delays. For the read interface, the Quartus II software reports individual setup and hold times for each DQ pin, and selecting the "List Paths" option in the timing report would provide data and clock propagation delays for that DQ pin. Select the worst-case setup and hold DQ registers and extract the minimum and maximum propagation delays.

For example, a “List Paths” on the setup time for DQ [0] shows the propagation delays of 1.945 ns on the DQ pin to register path, and 3.177 ns on the DQS clock pin to register path (Figure 8).

**Figure 8. Information  $t_{SU}$  for Register Example**

```
Info: tsu for register "example.<...>:dq_io[ddio_bidir_c0].auto_generated_cell_h[0]"
(data pin = "ddr2_dq[0]", clock pin = "ddr2_dqs[0]") is -1.268 ns
Info: + Longest pin to register delay is 1.945 ns
Info: 1: + IC(0.000 ns) + CELL(0.000 ns) = 0.000 ns; Loc. = PIN_E11; Fanout = 1; PIN Node = 'ddr2_dq[0]'
Info: 2: + IC(0.000 ns) + CELL(0.635 ns) = 0.635 ns; Loc. = IOC_X24_Y51_N3; Fanout = 2; COMB Node = 'example.<...>:auto_generated[ddio_bidira[0]]-COMBOUT'
Info: 3: + IC(0.981 ns) + CELL(0.245 ns) = 1.861 ns; Loc. = LCCOMB_X25_Y50_N4; Fanout = 1; COMB Node = 'example.<...>:auto_generated_cell_h[0]-feeder'
Info: 4: + IC(0.000 ns) + CELL(0.084 ns) = 1.945 ns; Loc. = LCFF_X25_Y50_N5; Fanout = 1; REG Node = 'example.<...>:auto_generated_cell_h[0]'
Info: Total cell delay = 0.964 ns (49.56 %)
Info: Total interconnect delay = 0.981 ns (50.44 %)
Info: + Micro setup delay of destination is -0.036 ns
Info: - Shortest clock path from clock "ddr2_dqs[0]" to destination register is 3.177 ns
Info: 1: + IC(0.000 ns) + CELL(0.000 ns) = 0.000 ns; Loc. = PIN_A9; Fanout = 1; CLK Node = 'ddr2_dqs[0]'
Info: 2: + IC(0.000 ns) + CELL(0.665 ns) = 0.665 ns; Loc. = IOC_X24_Y51_N1; Fanout = 1; COMB Node = 'example.<...>:auto_generated[combout[0]]'
Info: 3: + IC(0.081 ns) + CELL(0.633 ns) = 1.379 ns; Loc. = CLKDELAYCTRL_G9; Fanout = 1; COMB Node = 'example.<...>:g_ddr_io[delayed_dqs[0]]'
Info: 4: + IC(0.000 ns) + CELL(0.000 ns) = 1.379 ns; Loc. = CLKCTRL_G9; Fanout = 25; COMB Node = 'example.<...>:g_ddr_io[wire_dqs_clkctrl_outclk[0]]'
Info: 5: + IC(1.261 ns) + CELL(0.537 ns) = 3.177 ns; Loc. = LCFF_X25_Y50_N5; Fanout = 1; REG Node = 'example.<...>:auto_generated_cell_h[0]'
Info: Total cell delay = 1.895 ns (57.76 %)
Info: Total interconnect delay = 1.342 ns (42.24 %)
```

Using this approach, minimum and maximum propagation delays on the clock and data path are extracted and present in Table 5. This timing extraction is done twice, once with each device model (fast corner and slow corner). The difference between minimum and maximum delays is very small because of the matched routing paths within the die and package.

**Table 5. FPGA Timing Delays Note (1)**

	Fast Corner (ns)	Slow Corner (ns) (-6 Speed Grade)
Data delay (minimum)	1.090	1.902
Data delay (maximum)	1.164	1.998
Clock delay (minimum)	2.015	3.089
Clock delay (maximum)	2.204	3.342
Micro setup (2)	-0.032	-0.036
Micro hold (2)	0.152	0.266

**Notes to Table 5:**

- (1) These delays are reported in the <core\_instance\_name>\_extraction\_data.txt file located in your project directory. Data delay is the propagation delay from each DQ pin to the input DDR register and is reported as dq\_capture. Clock delay is the propagation delay to the DDR input registers from the corresponding DQS pin, and is calculated as dqs\_clkctrl + clkctrl\_capture.
- (2) The micro setup and micro hold times are specified in the DC & Switching Characteristics chapter, Volume 1, of the Cyclone II Device Handbook.

### Setup & Hold Margins Calculations

After obtaining all relevant timing information from the memory, FPGA, and board we are ready to calculate the setup and hold margins at the DQ capture register during read operations.

$$\begin{aligned}
 \text{Earliest clock arrival time} &= \text{Minimum clock delay within FPGA} - \text{DQS} \\
 &\quad \text{uncertainties} \\
 &= \text{Clock delay (minimum)} - t_{\text{DQS\_SKEW\_ADDER}} \\
 &= 3089 - 77.5 \\
 &= 3012 \text{ ps (with slow timing model)}
 \end{aligned}$$

$$\begin{aligned}
 \text{Latest data valid time} &= \text{Memory DQS-to-DQ valid} + \text{maximum data} \\
 &\quad \text{delay in FPGA} \\
 &= t_{\text{DQSQ}} + \text{data delay (maximum)} \\
 &= 350 + 1998 \\
 &= 2348 \text{ ps (with slow timing model)}
 \end{aligned}$$

$$\begin{aligned}
 \text{Setup margin} &= \text{Earliest clock arrival} - \text{latest data valid} - \\
 &\quad \text{micro setup} - \text{board uncertainty} \\
 &= t_{\text{EARLY\_CLOCK}} - t_{\text{LATE\_DATA\_VALID}} - ut_{\text{SU}} - t_{\text{EXT}} \\
 &= 3012 - 2348 - (-36) - 20 \\
 &= 680 \text{ ps (with slow turning model)}
 \end{aligned}$$

Repeating these calculations with the fast corner timing model, we derive 436 ps of setup margin.

$$\begin{aligned}
 \text{Latest clock arrival time} &= \text{Maximum clock delay within FPGA} + \text{DQS} \\
 &\quad \text{uncertainties} \\
 &= \text{Clock delay (maximum)} + t_{\text{DQS\_SKEW\_ADDER}} \\
 &= 3342 + 77.5 \\
 &= 3420 \text{ ps}
 \end{aligned}$$

$$\begin{aligned}
 \text{Earliest data invalid time} &= \text{Memory DQS-to-DQ invalid} + \text{minimum} \\
 &\quad \text{data delay in FPGA} \\
 &= (t_{\text{HP}} - t_{\text{QHS}}) + \text{data delay (minimum)} \\
 &= (2730 - 450) + 1902 \\
 &= 4182 \text{ ps (with slow turning model)}
 \end{aligned}$$

$$\begin{aligned}
 \text{Hold margin} &= \text{Latest clock arrival time} - \text{earliest data} \\
 &\quad \text{invalid time} - \text{micro hold} - \text{board uncertainty} \\
 &= t_{\text{EARLY\_DATA\_INVALID}} - t_{\text{LATE\_CLOCK}} - ut_{\text{H}} - t_{\text{EXT}} \\
 &= 4182 - 3420 - 266 - 20 \\
 &= 476 \text{ ps}
 \end{aligned}$$

Repeating these calculations with the fast corner timing model, we derive 917 ps of hold margin.

Table 6 shows the read timing analysis for the DDR2 SDRAM memory interface.

<b>Table 6. Read Timing Analysis for DDR2 SDRAM Interface in EP2C70F896C6 (Part 1 of 2)</b>				
	<b>Parameter</b>	<b>Fast Corner Model (ns)</b>	<b>Slow Corner Model (ns)</b>	<b>Description</b>
Memory specifications (1)	$t_{HP}$	2.730	2.730	Half period as specified by the memory data sheet (including memory clock duty cycle distortion)
	$t_{DQSQ}$	0.350	0.350	Skew between DQS and DQ from the memory
	$t_{QHS}$	0.450	0.450	Data hold skew factor as specified by the memory data sheet
FPGA specifications (2)	$t_{DQS\_SKEW\_ADDER}$	0.078	0.078	Clock delay skew adder for x8
	Minimum Clock Delay (3), (4)	2.015	3.089	Minimum DQS pin to LE register delay from the Quartus II software (with 0° static-based phase shift)
	Maximum Clock Delay (3), (4)	2.204	3.342	Maximum DQS pin to LE register delay from the Quartus II software (with 0° static-based phase shift)
	Minimum Data Delay (3), (4)	1.090	1.902	Minimum DQ pin to LE register delay from the Quartus II software
	Maximum Data Delay (3), (4)	1.164	1.998	Maximum DQ pin to LE register delay from the Quartus II software
	$\mu t_{SU}$	-0.032	-0.036	Intrinsic setup time of the LE register (rounded up)
	$\mu t_{H}$	0.152	0.266	Intrinsic hold time of the LE register (rounded up)
Board specifications	$t_{EXT}$	0.020	0.020	Board trace variations on the DQ and DQS lines

**Table 6. Read Timing Analysis for DDR2 SDRAM Interface in EP2C70F896C6 (Part 2 of 2)**

	Parameter	Fast Corner Model (ns)	Slow Corner Model (ns)	Description
Timing calculations	$t_{\text{EARLY\_CLOCK}}$	1.938	3.012	Earliest possible clock edge after DQS phase-shift circuitry and uncertainties (minimum clock delay – $t_{\text{DQS\_SKEW\_ADDER}}$ )
	$t_{\text{LATE\_CLOCK}}$	2.282	3.420	Latest possible clock edge after DQS phase-shift circuitry and uncertainties (maximum clock delay + $t_{\text{DQS\_SKEW\_ADDER}}$ )
	$t_{\text{EARLY\_DATA\_INVALID}}$	3.370	4.182	Time for earliest data to become invalid for sampling at FPGA flop ( $t_{\text{HP}} - t_{\text{QHS}} +$ minimum data delay)
	$t_{\text{LATE\_DATA\_VALID}}$	1.514	2.348	Time for latest data to become valid for sampling at FPGA flop ( $t_{\text{DQSQ}}$ + maximum data delay)
Results	Read setup timing margin (5)	0.436	0.680	$t_{\text{EARLY\_CLOCK}} - t_{\text{LATE\_DATA\_VALID}} - \mu t_{\text{SU}} - t_{\text{EXT}}$
	Read hold timing margin (5)	0.917	0.476	$t_{\text{EARLY\_DATA\_INVALID}} - t_{\text{LATE\_CLOCK}} - \mu t_{\text{H}} - t_{\text{EXT}}$
	Total margin	1.352	1.156	Setup + hold time margin

**Notes to Table 6:**

- (1) The memory numbers used here are from the Micron 256MB  $\times 72$  DIMM, MT9HTF3272A data sheet.
- (2) This analysis is performed with FPGA timing parameters for Cyclone II EP2C70F896. You should use this template to analyze timing for your preferred Cyclone II density-package combination.
- (3) These numbers are from the Quartus II software, version 6.0 using the Altera IP MegaCore function 3.4.0 for 72-bit DDR2 SDRAM interface.
- (4) The  $\times 72$  bit interface is configured using the top and bottom banks. The left bottom corner DQS (DQS[1]B) is used for this implementation.
- (5) DQS phase shift is user-defined and adjustable if you need to balance the setup and hold time margin.

Similarly, Table 7 shows the read timing analysis for the DDR SDRAM memory interface.

	<b>Parameter</b>	<b>Fast Corner Model (ns)</b>	<b>Slow Corner Model (ns)</b>	<b>Description</b>
Memory specifications (1)	$t_{HP}$	2.700	2.700	Half period as specified by the memory data sheet (including memory clock duty cycle distortion)
	$t_{DQSQ}$	0.450	0.450	Skew between DQS and DQ from the memory
	$t_{QHS}$	0.600	0.600	Data hold skew factor as specified by the memory data sheet
FPGA specifications (2)	$t_{DQS\_SKEW\_ADDER}$	0.078	0.078	Clock delay skew adder for x8
	Minimum Clock Delay (3), (4)	2.113	3.245	Minimum DQS pin to LE register delay from the Quartus II software (with 0° static-based phase shift)
	Maximum Clock Delay (3), (4)	2.302	3.498	Maximum DQS pin to LE register delay from the Quartus II software (with 0° static-based phase shift)
	Minimum Data Delay (3), (4)	1.043	1.864	Minimum DQ pin to LE register delay from the Quartus II software
	Maximum Data Delay (3), (4)	1.117	1.961	Maximum DQ pin to LE register delay from the Quartus II software
	$\mu t_{SU}$	-0.032	-0.036	Intrinsic setup time of the LE register (rounded up).
	$\mu t_{H}$	0.152	0.266	Intrinsic hold time of the LE register (rounded up).
Board specifications	$t_{EXT}$	0.020	0.020	Board trace variations on the DQ and DQS lines

**Table 7. Read Timing Analysis for DDR SDRAM Interface in EP2C70F896C6 (Part 2 of 2)**

	Parameter	Fast Corner Model (ns)	Slow Corner Model (ns)	Description
Timing calculations	$t_{\text{EARLY\_CLOCK}}$	2.036	3.168	Earliest possible clock edge after DQS phase-shift circuitry and uncertainties (minimum clock delay – $t_{\text{DQS\_SKEW\_ADDER}}$ )
	$t_{\text{LATE\_CLOCK}}$	2.380	3.576	Latest possible clock edge after DQS phase-shift circuitry and uncertainties (maximum clock delay + $t_{\text{DQS\_SKEW\_ADDER}}$ )
	$t_{\text{EARLY\_DATA\_INVALID}}$	3.143	3.964	Time for earliest data to become invalid for sampling at FPGA flop ( $t_{\text{HP}} - t_{\text{QHS}} +$ minimum data delay)
	$t_{\text{LATE\_DATA\_VALID}}$	1.567	2.411	Time for latest data to become valid for sampling at FPGA flop ( $t_{\text{DQSQ}} +$ maximum data delay)
Results	Read setup timing margin (5)	0.481	0.773	$t_{\text{EARLY\_CLOCK}} - t_{\text{LATE\_DATA\_VALID}} - \mu t_{\text{SU}} - t_{\text{EXT}}$
	Read hold timing margin (5)	0.591	0.102	$t_{\text{EARLY\_DATA\_INVALID}} - t_{\text{LATE\_CLOCK}} - \mu t_{\text{H}} - t_{\text{EXT}}$
	Total margin	1.072	0.875	Setup + hold time margin

**Notes to Table 7:**

- (1) The memory numbers used here are from the Micron 256MB  $\times 72$  DIMM, data sheet.
- (2) This analysis is performed with FPGA timing parameters for Cyclone II EP2C70F896. You should use this template to analyze timing for your preferred Cyclone II density-package combination.
- (3) These numbers are from the Quartus II software, version 6.0 using the Altera IP MegaCore function 3.4.0 for 72-bit DDR2 SDRAM interface.
- (4) The  $\times 72$  bit interface is configured using the top and bottom banks. The left bottom corner DQS (DQS[1]B) is used for this implementation.
- (5) DQS phase shift is user-defined and adjustable if you need to balance the setup and hold time margin.

## Write Timing Margins

Timing margin analysis for write data and address and control signals are very similar. This section analyzes timing for the write data signals. You can use the same approach to repeat this for the address and control signals.

For write operations, the DDR2 memory requires the clock strobe (DQS) to be center-aligned with the data bus (DQ). This is implemented in Cyclone II using the PLL phase-shift feature. Two output clocks are created from the PLL, with a relative  $90^\circ$  phase offset. The leading clock edge clocks out the DQ write data output pins to the memory, while the lagging clock edge generates the DQS clock strobe and CK/CK# memory

output clocks. Figure 3 on page 8 illustrates the timing relationship between the DQS and DQ inputs required by the memory during a write operation.

Figure 4 on page 11 shows that the write side uses a PLL to generate the clocks listed in Table 8.

<b>Clock</b>	<b>Description</b>
System clock	This is used for the memory controller and to generate the DQS write and CK/CK# signals.
Write clock ( $-90^\circ$ shifted from system clock)	This is used in the data path to generate the DQ write signals.

### **Memory Timing Parameters**

When writing to a memory, the FPGA needs to ensure that setup and hold times are met. These specifications ( $t_{DS}$  &  $t_{DH}$ ) are obtained from the data sheet (445 ps and 385 ps for the 167 MHz DDR2 example). Additionally, the FPGA needs to provide a memory clock (CK/CK#) that meets the clock high/low time specifications. Finally, the skew between the DQS output strobe and CK output clock cannot exceed limits set by the memory. The last parameter doesn't affect timing margins, but it must be met for successful memory operation.

### **FPGA Timing Parameters**

The timing paths within the FPGA for the DQ and DQS outputs to memory are matched by design. Dedicated clock networks drive double-data rate IO structures to generate DQ & DQS. This results in minimal skew between these outputs. Lets study these skew parameters: phase-shift error, clock skew, and package skew.

The two clock networks used are driven by the same PLL, however with a  $90^\circ$  relative phase shift. The  $0^\circ$  clock generates DQS, while a negative  $90^\circ$  clock generates DQ. Typical PLL uncertainties such as jitter and compensation error, affect both clock networks equally. Hence, these timing parameters do not affect write timing margins. However, because the clock generating DQ is phase shifted, the PLL phase-shift uncertainty ( $t_{PLL\_PSERR} = \pm 60$  ps, listed in the *DC & Switching Characteristics* chapter in volume 1 of the *Cyclone II Device Handbook*) affects DQ arrival times at the memory pins.

The Quartus II software models intra-clock skew, such as skew between nodes driven by the same dedicated clock network. However, skew between two such clock networks is not modeled and specified as an adder term. You should add this skew component to the propagation delays extracted from the Quartus II software.

For our 72-bit DDR2 interface that spans four I/O banks in the top and bottom of the device, the clock skew adder between two clock networks is specified as  $\pm 138$  ps ( $t_{\text{CLOCK\_SKEW\_ADDER}}$ ). We account for this uncertainty while calculating DQS arrival times at the memory pins.

The final skew component is package skew. As noted earlier, the Quartus II software models package trace delay for each pin on the device. Extracted propagation delays reflect any skew between output signals to the memory.

Table 9 shows the write timing analysis for the DDR2 SDRAM memory interface.

	Parameter	Fast Corner Model (ns)	Slow Corner Model (ns)	Description
Memory specifications (1)	$t_{\text{DS}}$	0.445	0.445	Memory data setup requirement
	$t_{\text{DH}}$	0.385	0.385	Memory data hold requirement
FPGA specifications (2)	$t_{\text{CK}/2}$	3.000	3.000	Ideal half clock period
	$t_{\text{DCD}}$	0.180	0.180	FPGA output clock duty cycle distortion
	$t_{\text{PLL\_JITTER}}$	0.000	0.000	Does not affect margin as the same PLL generates both write clocks ( $0^\circ$ and $90^\circ$ )
	$t_{\text{PLL\_PSERR}}$	0.060	0.060	PLL phase-shift error (On $-90^\circ$ data output)
	$t_{\text{CLOCK\_SKEW\_ADDER}}$	0.138	0.138	Clock skew between two dedicated clock networks feeding IO banks across the FPGA
	Minimum Clock $t_{\text{CO}}$ (3), (4)	2.529	4.732	Minimum DQS $t_{\text{CO}}$ from the Quartus II software
	Maximum Clock $t_{\text{CO}}$ (3), (4)	2.760	5.229	Maximum DQS $t_{\text{CO}}$ from the Quartus II software
	Minimum Data $t_{\text{CO}}$ (3), (4)	1.008	3.210	Minimum DQ $t_{\text{CO}}$ from the Quartus II software
Maximum Data $t_{\text{CO}}$ (3), (4)	1.250	3.715	Maximum DQ $t_{\text{CO}}$ from the Quartus II software	

**Table 9. Write Timing Analysis for DDR2 SDRAM Interface in EP2C70F896C6 (Part 2 of 2)**

	Parameter	Fast Corner Model (ns)	Slow Corner Model (ns)	Description
Board specifications	$t_{EXT}$	0.020	0.020	Board trace variations on the DQ and DQS lines
Timing calculations	$t_{EARLY\_CLOCK}$	2.391	4.594	Earliest possible clock edge seen by memory device (minimum clock delay – $t_{PLL\_JITTER} - t_{CLOCK\_SKEW\_ADDER}$ )
	$t_{LATE\_CLOCK}$	2.898	5.367	Latest possible clock edge seen by memory device (maximum clock delay + $t_{PLL\_JITTER} + t_{CLOCK\_SKEW\_ADDER}$ )
	$t_{EARLY\_DATA\_INVALID}$	3.768	5.970	Time for earliest data to become invalid for sampling at the memory input pins ( $t_{HP} - t_{DCD} +$ minimum data delay - $t_{PLL\_PSERR}$ )
	$t_{LATE\_DATA\_VALID}$	1.310	3.775	Time for latest data to become valid for sampling at the memory input pins (maximum data delay + $t_{PLL\_PSERR}$ )
Results	Write setup timing margin	0.616	0.354	$t_{EARLY\_CLOCK} - t_{LATE\_DATA\_VALID} - t_{DS} - t_{EXT}$
	Write hold timing margin	0.465	0.198	$t_{EARLY\_DATA\_INVALID} - t_{LATE\_CLOCK} - t_{DH} - t_{EXT}$
	Total margin	1.081	0.552	Setup + hold time margin

**Notes to Table 9:**

- (1) The memory numbers used here are from the Micron 256MB  $\times 72$  DIMM, MT9HTF3272A data sheet.
- (2) This analysis is performed with FPGA timing parameters for Cyclone II EP2C70F896. You should use this template to analyze timing for your preferred Cyclone II density-package combination.
- (3) These numbers are from the Quartus II software, version 6.0 using the Altera IP MegaCore function 3.4.0 for 72-bit DDR2 SDRAM interface.
- (4) The  $\times 72$  bit interface is configured using the top and bottom banks. The left bottom corner DQS (DQS[1]B) is used for this implementation.

Similarly, Table 10 shows the write timing analysis for the DDR SDRAM memory interface.

<b>Table 10. Write Timing Analysis for DDR SDRAM Interface in EP2C70F896C6 (Part 1 of 2)</b>				
	<b>Parameter</b>	<b>Fast Corner Model (ns)</b>	<b>Slow Corner Model (ns)</b>	<b>Description</b>
Memory specifications (1)	$t_{DS}$	0.450	0.450	Memory data setup requirement
	$t_{DH}$	0.450	0.450	Memory data hold requirement
FPGA specifications (2)	$t_{CK/2}$	3.000	3.000	Ideal half clock period
	$t_{DCD}$	0.150	0.150	FPGA output clock duty cycle distortion
	$t_{PLL\_JITTER}$	0.000	0.000	Does not affect margin as the same PLL generates both write clocks (0x and 90x)
	$t_{PLL\_PSERR}$	0.060	0.060	PLL phase-shift error (On -90x data output)
	$t_{CLOCK\_SKEW\_ADDER}$	0.138	0.138	Clock skew between two dedicated clock networks feeding IO banks across the FPGA
	Minimum Clock $t_{CO}$ (3), (4)	2.322	4.129	Minimum DQS $t_{CO}$ from the Quartus II software
	Maximum Clock $t_{CO}$ (3), (4)	2.553	4.626	Maximum DQS $t_{CO}$ from the Quartus II software
	Minimum Data $t_{CO}$ (3), (4)	0.801	2.606	Minimum DQ $t_{CO}$ from the Quartus II software
Maximum Data $t_{CO}$ (3), (4)	1.043	3.112	Maximum DQ $t_{CO}$ from the Quartus II software	
Board specifications	$t_{EXT}$	0.020	0.020	Board trace variations on the DQ and DQS lines
Timing calculations	$t_{EARLY\_CLOCK}$	2.184	3.991	Earliest possible clock edge seen by memory device (minimum clock delay – $t_{PLL\_JITTER}$ – $t_{CLOCK\_SKEW\_ADDER}$ )
	$t_{LATE\_CLOCK}$	2.691	4.764	Latest possible clock edge seen by memory device (maximum clock delay + $t_{PLL\_JITTER}$ + $t_{CLOCK\_SKEW\_ADDER}$ )
	$t_{EARLY\_DATA\_INVALID}$	3.441	5.246	Time for earliest data to become invalid for sampling at the memory input pins ( $t_{HP} - t_{DCD}$ + minimum data delay - $t_{PLL\_PSERR}$ )
	$t_{LATE\_DATA\_VALID}$	1.103	3.172	Time for latest data to become valid for sampling at the memory input pins (maximum data delay + $t_{PLL\_PSERR}$ )

**Table 10. Write Timing Analysis for DDR SDRAM Interface in EP2C70F896C6 (Part 2 of 2)**

	Parameter	Fast Corner Model (ns)	Slow Corner Model (ns)	Description
Results	Write setup timing margin	0.611	0.349	$t_{\text{EARLY\_CLOCK}} - t_{\text{LATE\_DATA\_VALID}} - t_{\text{DS}} - t_{\text{EXT}}$
	Write hold timing margin	0.280	0.012	$t_{\text{EARLY\_DATA\_INVALID}} - t_{\text{LATE\_CLOCK}} - t_{\text{DH}} - t_{\text{EXT}}$
	Total margin	0.891	0.361	Setup + hold time margin

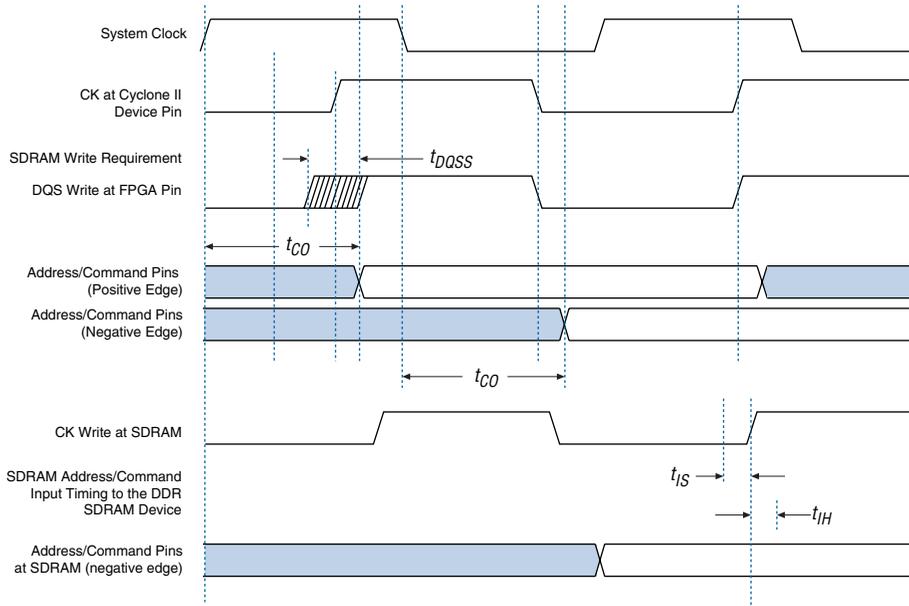
**Notes to Table 10:**

- (1) The memory numbers used here for come from Micron 256MB x72 DIMM, MT9VDDT3272A data sheet.
- (2) This analysis is performed with FPGA timing parameters for Cyclone II EP2C70F896. You should use this template to analyze timing for your preferred Cyclone II density-package combination.
- (3) These numbers are from the Quartus II software, version 6.0 using the Altera IP MegaCore function 3.4.0 for 72-bit DDR2 SDRAM interface.
- (4) The x72 bit interface is configured using the top and bottom banks. The left bottom corner DQS (DQS[1]) is used for this implementation.

## Command & Address Timing

Command and address signals are generated from the system clock (or another clock) in single data rate. The command and address signals must meet the setup and hold time requirement with respect to the rising edge of the CK signal at the DDR2/DDR SDRAM device. The FPGA also generates the CK signal from the system clock directly. Depending on the location of the registers for the commands and addresses, you may need to use a different system clock edge or add a phase shift on the system clock to make sure that these signals meet the setup and hold time requirement at the DDR2/DDR SDRAM device. This section outlines the commands and addresses timing considerations.

In this example, the command and address signals are edge-aligned with the CK signal (if there are no variations in the package or board trace length of the system for the different signals). This means that the address and command signals cannot meet the setup time requirement of the DDR2/DDR SDRAM device. In order to meet the setup and hold time requirement, you have to use the negative edge of the system clocks to generate the command and address signals. [Figure 9](#) shows the command and address timing and how the system clock edge affects how the signals meet the DDR2/DDR SDRAM  $t_{\text{DQS}}$ ,  $t_{\text{IS}}$ , and  $t_{\text{IH}}$  requirements.

**Figure 9. Address & Command Timing Notes (1), (2)**

**Notes to Figure 9:**

- (1) The address and command timing shown in Figure 9 is applicable for both read and write.
- (2) If the board trace lengths for the DQS, CK, address, and command pins are the same, the signal relationships at the Cyclone II device pins are maintained at the DDR and DDR2 SDRAM pins.

You can perform timing analysis for command and address signals similar to the write data timing analysis to find the optimal phase shift to generate the command and address signals. The only difference between the write data timing analysis and the command and address timing analysis is that the command and address timing signals are single data rate whereas the data signals are double data rate.

The default clock for the command and address signals in the DDR2/DDR MegaCore function is the falling edge of the system clock. The Quartus II software reports the  $t_{CO}$ , but does not automatically verify that source-synchronous outputs meet the setup/hold requirements at the destination. You must subtract a 1/2 clock cycle from the  $t_{CO}$  reported. Table 11 shows the  $t_{CO}$  reported by the Quartus II software and the actual  $t_{CO}$  if you relate this  $t_{CO}$  with the CK/CK#  $t_{CO}$ .

**Table 11. Reported & Adjusted Command/Address  $t_{CO}$  Note (1)**

	Fast Timing Model (ns)	Slow Timing Model (ns) (-6 Speed Grade)
Quartus II software reported minimum command/address $t_{CO}$	1.520	3.042
Quartus II software reported maximum command/address $t_{CO}$	1.586	3.108
Adjusted minimum command/address $t_{CO}$	-1.480	0.042
Adjusted maximum command/address $t_{CO}$	-1.414	0.108

**Note to Table 11:**

- (1) The Quartus II software does not automatically verify that source-synchronous outputs meet the setup/hold requirements at the destination. This is a system timing analysis, so you need to find the actual  $t_{CO}$  compared to the  $t_{CO}$  for the CK/CK# signals. The adjusted  $t_{CO}$  numbers show the 1/2 clock cycle reported from the Quartus II software reported  $t_{CO}$  to account for the falling edge signal.

Table 12 shows an example of the command and address timing analysis for an EP2C70F896C6 interfacing with 167-MHz DDR2 SDRAM DIMM.

**Table 12. Command & Address Write Timing Analysis for DDR2 SDRAM Interface in EP2C70F896C6 (Part 1 of 3)**

	Parameter	Fast Corner Model (ns)	Slow Corner Model (ns)	Description
Memory specifications (1)	$t_{DS}$	0.600	0.600	Memory data setup requirement
	$t_{DH}$	0.600	0.600	Memory data hold requirement

**Table 12. Command & Address Write Timing Analysis for DDR2 SDRAM Interface in EP2C70F896C6 (Part 2 of 3)**

	Parameter	Fast Corner Model (ns)	Slow Corner Model (ns)	Description
FPGA specifications (2)	$t_{CK}$	6.000	6.000	Ideal clock period
	$t_{DCD}$	0.000	0.000	Does not affect margin as command and address are single data rate operations
	$t_{PLL\_JITTER}$	0.000	0.000	Does not affect margin as the same PLL generates both write clocks (0° and 90°)
	$t_{PLL\_PSERR}$	0.060	0.060	PLL phase-shift error (On -90° data output)
	$t_{CLOCK\_SKEW\_ADDER}$	0.118	0.118	Clock skew between two dedicated clock networks feeding one IO bank on the same side of the FPGA
	Minimum Clock $t_{CO}$ (3)	2.150	4.298	Minimum clock $t_{CO}$ from the Quartus II software
	Maximum Clock $t_{CO}$ (3)	2.325	4.570	Maximum clock $t_{CO}$ from the Quartus II software
	Minimum Cmd/Add $t_{CO}$ (4)	-1.480	0.042	Minimum Cmd/Add $t_{CO}$ from the Quartus II software
Maximum Cmd/Add $t_{CO}$ (4)	-1.414	0.108	Maximum Cmd/Add $t_{CO}$ from the Quartus II software	
Board specifications	$t_{EXT}$	0.020	0.020	Board trace variations on the DQ and DQS lines
Timing calculations	$t_{EARLY\_CLOCK}$	2.032	4.108	Earliest possible clock edge seen by memory device (minimum clock delay – $t_{PLL\_JITTER} - t_{CLOCK\_SKEW\_ADDER}$ )
	$t_{LATE\_CLOCK}$	2.443	4.688	Latest possible clock edge seen by memory device (maximum clock delay + $t_{PLL\_JITTER} + t_{CLOCK\_SKEW\_ADDER}$ )
	$t_{EARLY\_DATA\_INVALID}$	4.460	5.952	Time for earliest data to become invalid for sampling at the memory input pins ( $t_{CK} - t_{DCD} +$ minimum cmd/add delay - $t_{PLL\_PSERR}$ )
	$t_{LATE\_DATA\_VALID}$	-1.354	0.198	Time for latest data to become valid for sampling at the memory input pins (maximum cmd/add delay + $t_{PLL\_PSERR}$ )

**Table 12. Command & Address Write Timing Analysis for DDR2 SDRAM Interface in EP2C70F896C6 (Part 3 of 3)**

	Parameter	Fast Corner Model (ns)	Slow Corner Model (ns)	Description
Results	Read setup timing margin	2.766	3.362	$t_{\text{EARLY\_CLOCK}} - t_{\text{LATE\_DATA\_VALID}} - t_{\text{DS}} - t_{\text{EXT}}$
	Read hold timing margin	1.397	0.644	$t_{\text{EARLY\_DATA\_INVALID}} - t_{\text{LATE\_CLOCK}} - t_{\text{DH}} - t_{\text{EXT}}$
	Total margin	4.163	4.006	Setup + hold time margin

**Notes to Table 12:**

- (1) The memory numbers used here for come from Micron 256MB × 72 DIMM, MT9HTF3272A data sheet.
- (2) This analysis is performed with FPGA timing parameters for Cyclone II EP2C70F896. You should use this template to analyze timing for your preferred Cyclone II density-package combination.
- (3) These numbers are from the Quartus II software, version 6.0 using the Altera IP MegaCore function 3.4.0 for the 72-bit DDR2 SDRAM interface.
- (4) Command and address signals are generated on the falling edge of the system clock in this example. This value is adjusted from the Quartus II software reported  $t_{\text{CO}}$  because the Quartus II software reports the  $t_{\text{CO}}$  based on the rising edge of the input clock regardless of how the signal is generated.

## Round Trip Delay

Figure 10 shows the timing analysis and illustration of the round trip delay. The round trip delay is the delay from the FPGA clock to the DDR or DDR2 SDRAM and back to the FPGA (input to register B). The analysis is required to reliably transfer data from the register A, which is in the DQS clock domain, to register B, which is in the system clock domain.



To sample the Q output of register A into register B, you need the time relationship between register B's clock input and the D input, which depends on the phase relationship between DQS and the clock signal and involves the following steps:

1. Calculate the system's round trip delay.
2. Select a resynchronization phase of the system clock or other available clock that reliably samples the Q output of register A, based on the calculated safe resynchronization window. See [Figure 11](#).
3. Apply the correct clock edge for your resynchronization logic in your memory controller.

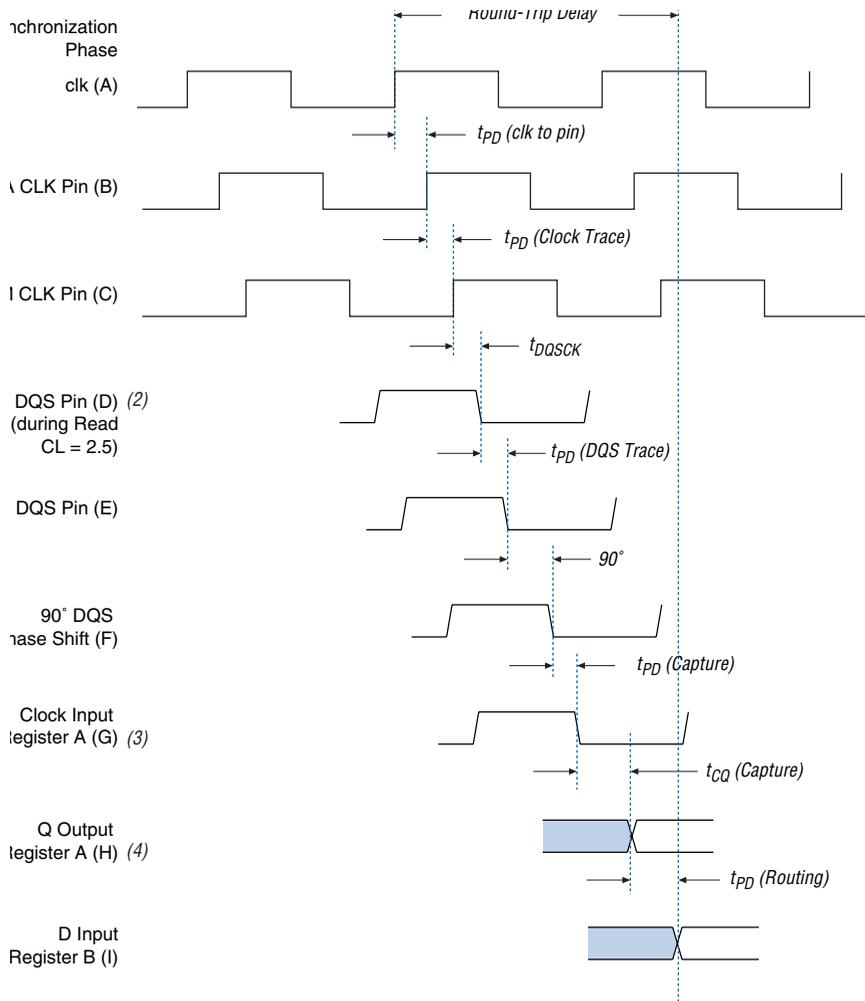
You can use the `clk` or `clk_shifted` signals as the register B clock input. You can invert `clk` and `clk_shifted` if needed. To determine the timing of data at the D input of register B relative to `clk`, consider the following timing-path dependencies, which are in chronological order:

- The DDR/DDR2 SDRAM clock input arrives (a delayed version of `clk`)
- DQS strobe from the DDR/DDR2 SDRAM arrives at the clock input of register A
- Data arrives at the Q output of register A
- Data arrives at the D input of register B

There are three main parts to this path:

- Clock delays between the FPGA global clock net and the DDR or DDR2 SDRAM clock input
- DQS strobe delays between the DDR or DDR2 SDRAM clock input and DQS's arrival at the FPGA capture registers
- Read data delays between the output of register A and the input of register B

[Figure 10](#) shows the individual delays between points (A) and (I). The sum of all these delays is the round trip delay. [Figure 11](#) shows the timing relationship of the signals for the delays between points (A) to (I) for a DDR SDRAM of CAS latency of 2.5.

**Figure 11. RTD Calculation for DDR SDRAM** Notes (1) through (4)

**Notes to Figure 11:**

- (1) The letters in parenthesis refer to the letters in Figure 10.
- (2) The DQS strobe edge can be anywhere within  $\pm t_{DQSK}$  of the DDR2/DDR SDRAM clock pin edge. Figure 10 assumes the DQS strobe occurs  $t_{DQSK}$  time after the clock for the maximum round-trip delay calculation and occurs  $t_{DQSK}$  time before the clock for minimum round-trip delay calculation.
- (3) The delays in the DQS path from the FPGA pin to the capture register are matched to the delays for the DQ path with the exception of the DQS delay chain.
- (4) Although data is initially sampled at a capture register on the positive edge of DQS,  $Q_H$  and  $Q_L$  are only available on the negative edge in SDR at the Q outputs of the DDR capture logic.

To determine the point at which the data can be reliably resynchronized, calculate the minimum and maximum round-trip delay. You can then determine what resynchronization logic to use for your system. Make sure to take PVT variations into account.

Delay (A) to (B) is the clock-to-output time to generate the clock signals to the DDR or DDR2 SDRAM device.

Delay (B) to (C) is the trace delay for the clock. If there are multiple devices in the system, use the one furthest away from the FPGA for the maximum calculation and the one closest to the FPGA for the minimum calculation.

Delay (C) to (D) is the relationship between the clock and the DQS strobe timing during reads. The  $t_{DQSCk}$  time in DDR and DDR2 SDRAM specifications is nominally 0, but varies by  $\pm 0.75$  ns for DDR SDRAM and  $\pm 0.5$  ns for DDR2 SDRAM, depending on the device speed grade. The DQS output strobe is guaranteed to be within  $\pm t_{DQSCk}$  of the clock input. Use  $t_{DQSCk}(\text{maximum})$ , typically  $+0.75$  ns for DDR SDRAM and  $+0.5$  ns for DDR2 SDRAM, to calculate the maximum round-trip delay and use  $t_{DQSCk}(\text{minimum})$ , typically  $-0.75$  ns for DDR SDRAM and  $-0.5$  ns for DDR2 SDRAM, to calculate the minimum delay.

Delay (D) to (E) is the trace delay for DQS, which typically matches the trace delay for the DQ signals in the same byte group. To calculate the maximum round-trip delay, use the byte group with the longest trace lengths. Use the byte group with the shortest trace lengths to calculate the minimum round-trip delay. Similarly, if there are multiple devices in the system, use the one furthest from the FPGA for the maximum calculation and the one closest to the FPGA for the minimum. Trace lengths between different byte groups do not have to be tightly matched, but a difference between the longest and shortest trace lengths decreases the safe resynchronization window, the window size within which the data can be reliably resynchronized.

To calculate the maximum round trip delay, use the longest delay for the whole interface and use the shortest delay for the whole interface for the minimum round trip delay.

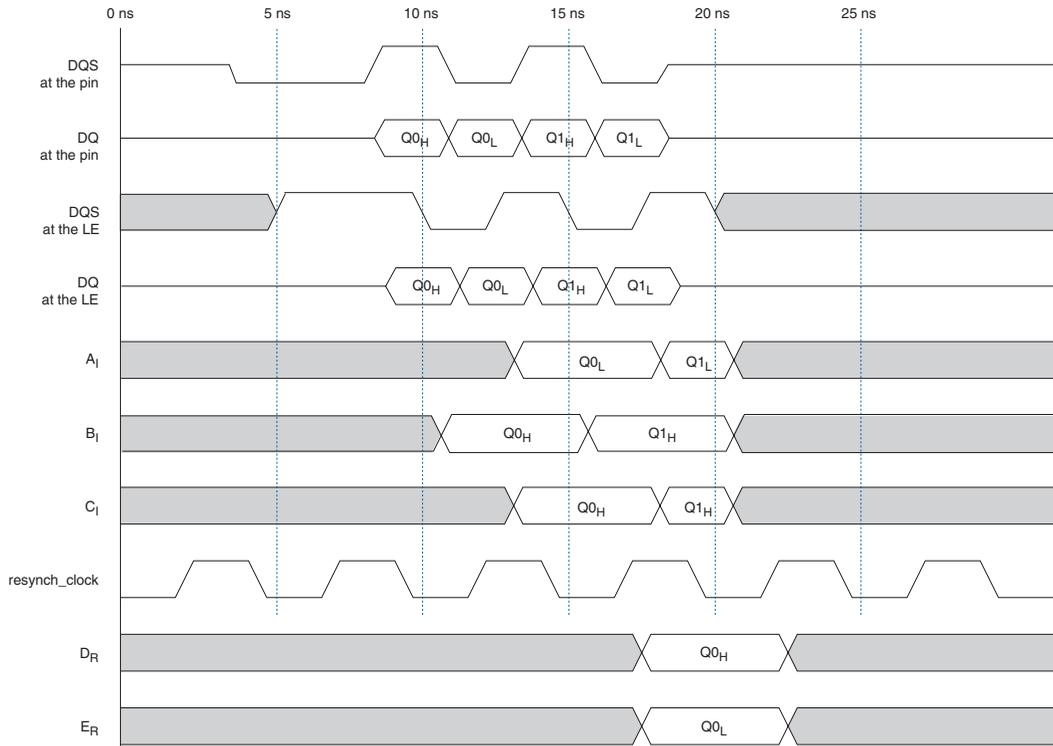
PLL jitter, clock duty cycle, and the half cycle used to align  $Q_H$  and  $Q_L$  also affect the round-trip delay. You must add each of these delays to the maximum round-trip delay and subtract them from the minimum round-trip delay.

## DQS Postamble

The DQ and DQS pins use the SSTL-2 I/O standard for DDR SDRAM and the SSTL-18 I/O standard for DDR2 SDRAM. When either the Cyclone II or the DDR SDRAM device does not drive the DQ and DQS pins, the signals go to a high-impedance state. Since a pull-up resistor terminates both DQ and DQS to  $V_{TT}$  (1.25 V for SSTL-2 and 0.9 V for SSTL-18), the effective voltage on the high-impedance line is 1.25 or 0.9 V, respectively. According to the JEDEC JESD 8-9 specification for the SSTL-2 I/O standard and JEDEC JESD8-15A specification for the SSTL-18 I/O standard, this is an indeterminate logic level and the input buffer can interpret this as either a logic high or logic low. If there is any noise on the DQS line, the input buffer may interpret that noise as actual strobe edges. Therefore, when the DQS signal gets tri-stated after a read postamble, you should disable the input LE registers so that erroneous data does not get latched in and all the data from the memory gets resynchronized properly.

Figure 12 shows a read operation example when the DQS postamble could be a problem. Waveform  $A_1$  shows the output of register  $A_1$ . Waveform  $B_1$  shows the output of the LE register  $B_1$ . The output of register  $B_1$  goes into register  $C_1$  whose output is shown in waveform  $C_1$ . Waveforms  $D_R$  and  $E_R$  illustrate the output signals after the resynchronization registers.

**Figure 12. Read Example with a DQS Postamble Issue**



The first falling edge of the DQS signal at the LE register occurs at 10 ns. At this point, data  $Q0_H$  is clocked in by register B<sub>1</sub> (waveform B<sub>1</sub>). At 12.5 ns, data  $Q0_L$  is clocked in by the active high register A<sub>1</sub> (waveform A<sub>1</sub>) and data  $Q0_H$  passes through the register C<sub>1</sub> (waveform C<sub>1</sub>). In this example, the positive edge of the `resynch_clock` signal occurs at 16.5 ns, where both  $Q0_H$  and  $Q0_L$  are sampled by the LE's resynchronization registers. Similarly, data  $Q1_H$  is clocked in by register B<sub>1</sub> at 15 ns, while data  $Q1_L$  is clocked in by register A<sub>1</sub> and data  $Q1_H$  passes through register C<sub>1</sub> at 17.5 ns. At 20 ns, assume that noise on the DQS line causes a valid clock edge at the LE registers and changes the values of waveforms A<sub>1</sub>, B<sub>1</sub>, and C<sub>1</sub>. The next rising edge of the `resynch_clock` signal does not occur until 21.5 ns, but data  $Q1_L$  and  $Q1_H$  are not valid anymore at the output of register A<sub>1</sub> and register C<sub>1</sub>, so the resynchronization registers do not sample  $Q1_L$  and  $Q1_H$  and may sample the wrong data instead.

Cyclone II devices have non-dedicated logic that can be configured to prevent a false edge trigger at the end of the DQS postamble. Each Cyclone II DQS signal is connected to postamble logic that consists of a D flip flop (see Figure 13). This register is clocked by the shifted DQS signal. Its input is connected to ground. The controller needs to include extra logic to tell the reset signal to release the preset signal on the falling DQS edge at the start of the postamble. This disables any glitches that happen right after the postamble. This postamble logic is automatically implemented by the Altera® MegaCore® DDR2 SDRAM Controller in the LE register as part of the open-source data path.

**Figure 13. Cyclone II DQS Postamble Circuitry Connection**

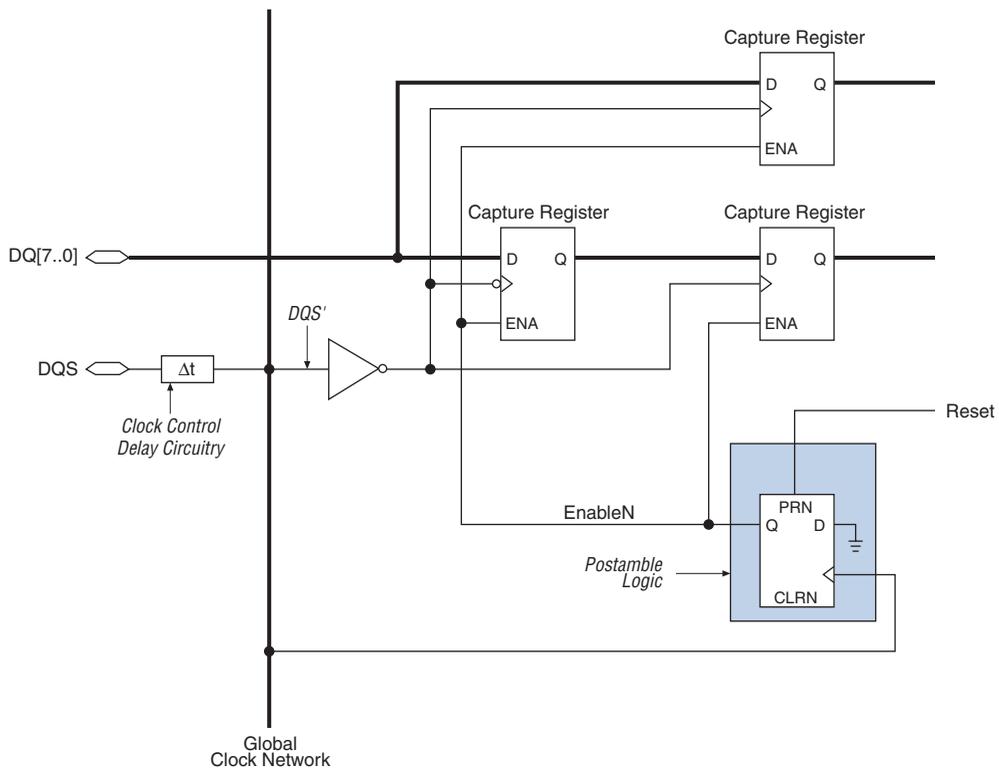


Figure 14 shows the timing waveform for Figure 13. Figure 15 shows the read timing waveform when the Cyclone II DQS postamble logic is used. When the postamble logic detects the falling DQS edge at the start of postamble, it sends out a signal to disable the capture registers to prevent any accidental latching.

**Figure 14. Cyclone II DQS Postamble Circuitry Control Timing Waveform**

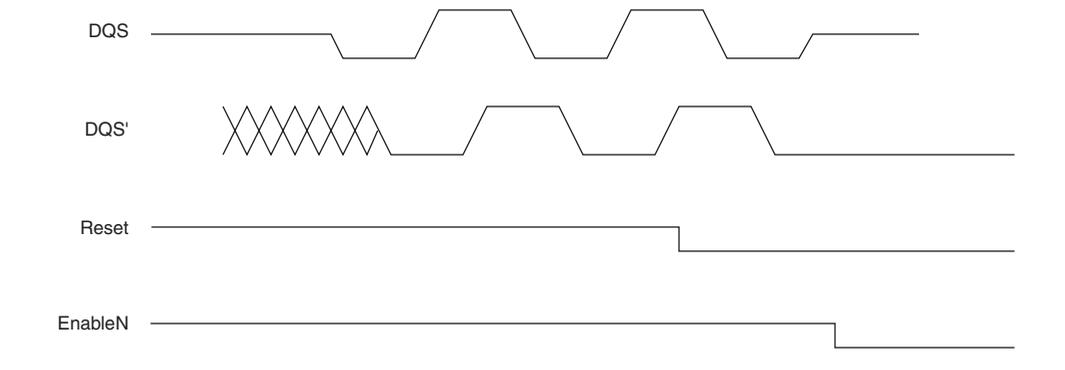
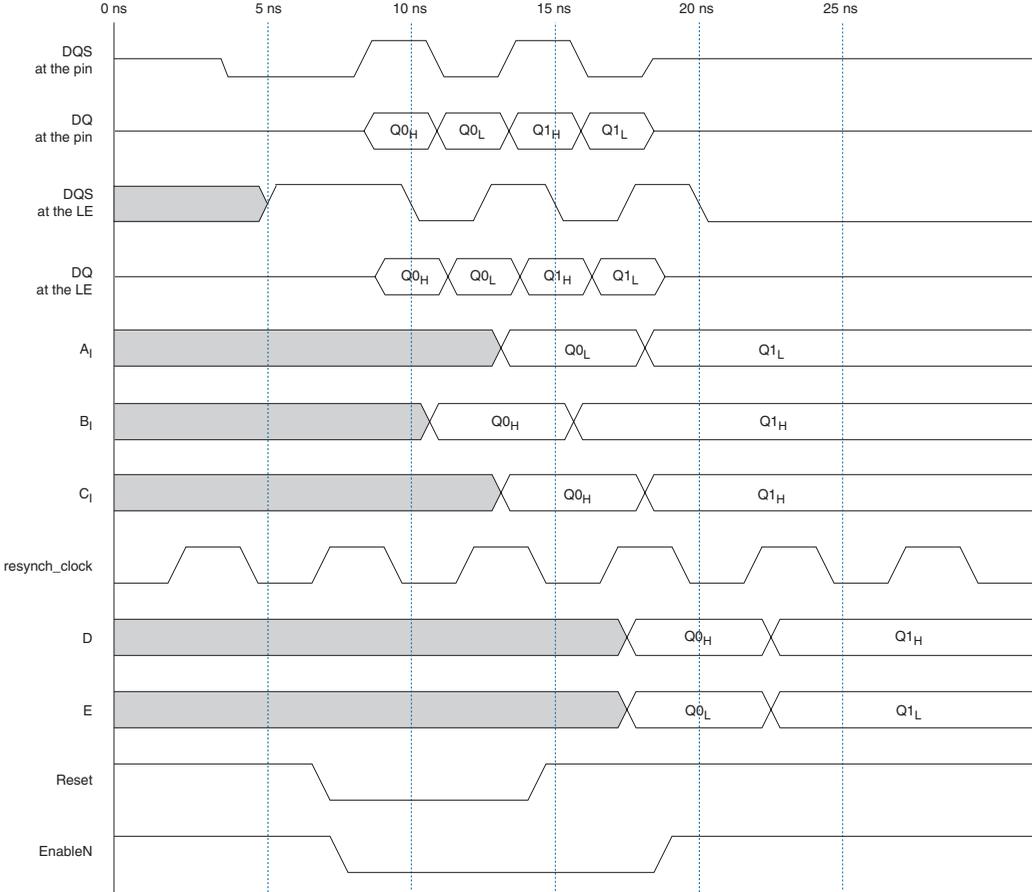


Figure 15. Cyclone II DQS Postamble Circuitry Read Timing Waveform



### Conclusion

DDR and DDR2 SDRAM devices are widely used in FPGA designs, and the DDR technology is the most popular DRAM architecture. Cyclone II devices have dedicated circuitry to interface with DDR and DDR2 SDRAM at speeds up to 167 MHz with comfortable and consistent margins. Additionally, this allows system designers to enhance their Cyclone II system performance through the use of commercial off-the-shelf PC memory, reducing cost. The Cyclone II device's DDR and DDR2 interface allows designers to use these devices in applications that require fast data transmission, simplified the system design, and improved performance.

## References

JEDEC Standard Publication JESD79C, DDR SDRAM Specification, JEDEC Solid State Technology Association.

JEDEC Standard Publication JESD79-2, DDR2 SDRAM Specification, JEDEC Solid State Technology Association.

JEDEC Standard Publication JESD8-9B, Stub Series Termination Logic for 2.5V (SSTL-2), JEDEC Solid State Technology Association.

JEDEC Standard Publication JESD8-15A, Stub Series Termination Logic for 1.8V (SSTL-18), JEDEC Solid State Technology Association.

MT9VDDT3272A, 256 MB: DDR SDRAM Unbuffered DIMM Data Sheet, Micron Technology, Inc.

MT9HTE3272A, 256 MB: DDR2 SDRAM Unbuffered DIMM Data Sheet, Micron Technology, Inc.



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
www.altera.com  
Applications Hotline: (800) 800-EPLD  
Literature Services: literature@altera.com



Copyright © 2006 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

