External Memory Interface Handbook Volume 5

Section I. Implementing Custom Memory Interface PHY

101 Innovation Drive
San Jose, CA 95134
www.altera.com

EMI_CUSTOM-1.1

Subscribe

Contents

Additional Information

1. Creating a Custom PHY

This chapter describes the FPGA design flow to implement a memory interface datapath (PHY) using Stratix® III and Stratix IV devices and the ALTDLL and ALTDQ_DQS megafunctions. You can use this method as an alternative to using the available external memory interface IP. You can then connect this PHY with a custom memory controller to interface with other memory components.

Altera recommends using the available external memory IPs instead of using the ALTDLL and ALTDQ_DQS megafunctions whenever possible.

The ALTDLL megafunction implements the dedicated DQS circuitry, while the ALTDQ_DQS megafunction implements the read and write PHY required for the interface. For more information about these megafunctions, refer to the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

You must constrain the timing of the PHY when using ALTDLL and ALTDQ_DQS megafunctions. This chapter does not cover details of the timing constraints; however, it includes information about timing analysis and board-level constraints that demonstrate and validate the interface.

The design flow steps in this chapter focus on the "Instantiate PHY and Controller" step of the *Recommended Design Flow* chapter of the *External Memory Handbook*. You can adapt the steps for a DDR2 SDRAM interface to create other types of memory interfaces.

After selecting the appropriate device and memory type, create a project in the Quartus II software that targets the device and memory type. When instantiating the datapath for DDR and DDR2 SDRAM interfaces in Stratix III and Stratix IV devices, Altera recommends using the ALTDLL and ALTDQ_DQS megafunctions for the datapath of custom external memory interfaces that are not supported by the ALTMEMPHY or UniPHY IP.

You use the following megafunctions to create a complete memory interface PHY:

- ALTDLL megafunction
- ALTDQ_DQS megafunction
- ALTPLL megafunction
- ALTIOBUF megafunction
- ALTOCT megafunction

The following sections describe the work flow when you instantiate PHY via ALTDLL and ALTDQ_DQS megafunctions and your custom-designed controller in a Quartus II project:

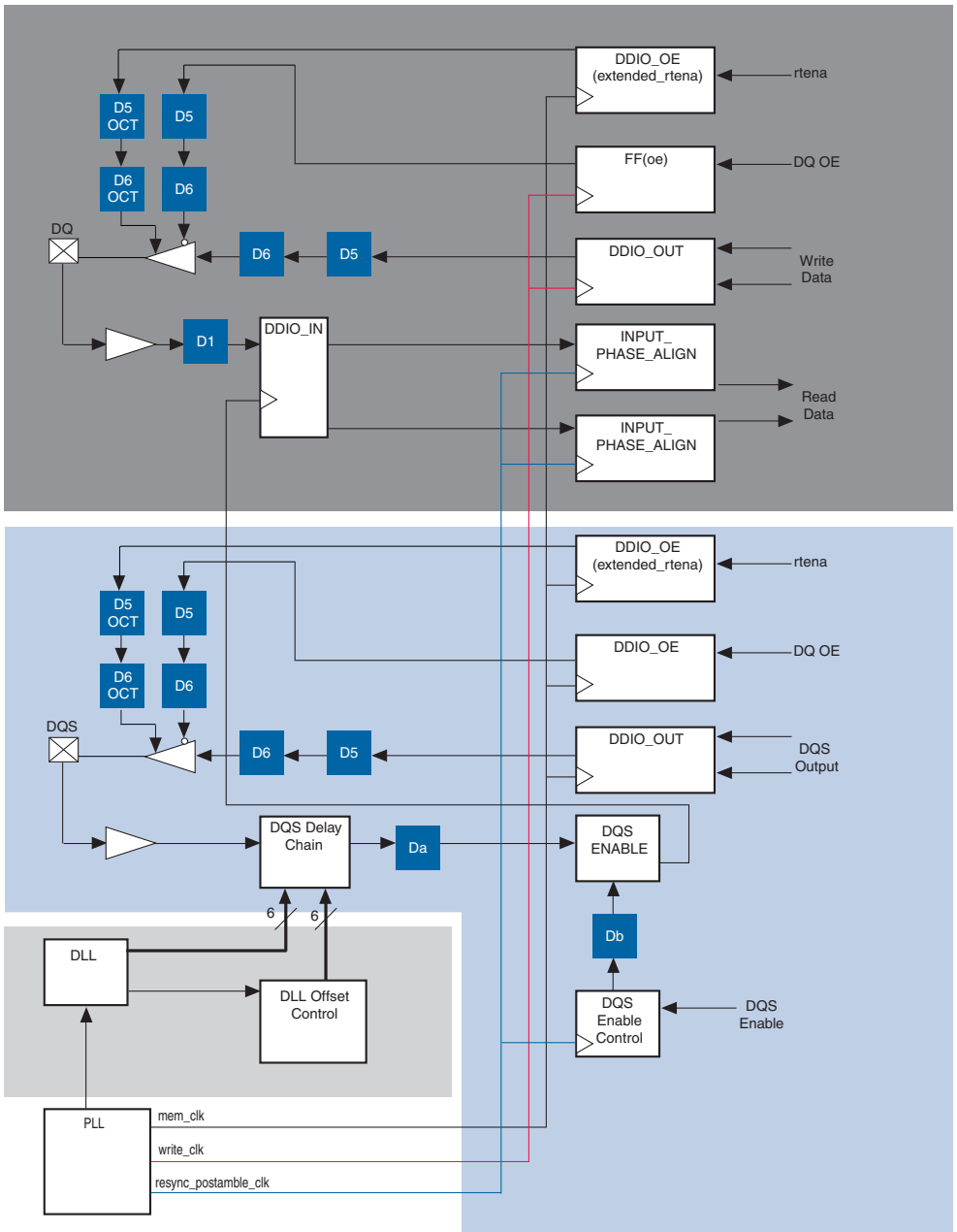
Instantiate the ALTPLL Megafunction

The ALTPLL instance provides the necessary clocking scheme for the custom PHY. Clocking schemes for the custom external memory interface are important. You must understand what clocks are required for a full-rate or half-rate interface. Both interfaces require different clocking schemes.

Clocking Scheme Requirement for Full-Rate Interface

Figure 1–1 on page 1–2 shows ta sample full-rate interface.

Figure 1–1. Example Full-Rate Interface



The PLL instance should generate the following five clock outputs for the full-rate interface (refer to Figure 1–1):

- DLL Input clock
- Resynchronize postamble clock (`resync_postamble_clk`)
- Write Clock (`write_clk`)
- Memory clock (`mem_clk`)
- Address/command clock

The following sections descibe each of the five clock outputs for the full-rate interface.

DLL Input Clock

This clock toggles the DLL block. The DLL clock is equivalent to the full rate of the interface and has a 0° phase shift. You must connect the DLL block directly to a single dedicated PLL clock output.

Resynchronize Postamble Clock (resync_postamble_clk)

The `resync_postamble_clk` clock is an optional clock.

You can use this clock for two purposes:

- Clock the DQS enable control block. This block is used to disable the DQS input strobe when the strobe goes to Hi-Z (after a DDR read postamble). This is part of the DQS input path.
- Clock the INPUT_PHASE_ALIGN block. This block phase shifts the input signal and is primarily used to match the arrival delay of the DQS to the latest arrival delay of a DQS from the DDR or DDR2 DIMM. It is also used to resynchronize the data read from the DDIO_IN block. This is part of the DQ input path.

The `resync_postamble_clk` clock should be the full rate of the interface and have dynamic phase. You can set the dynamic phase capability for the PLL clock outputs with the ALTPLL megafunction. The correct `resync_postamble_clk` clock phase is crucial to correct data transfer. The ALTDQ_DQS megafunction cannot determine the phase that the `resync_postamble_clk` clock should have. Therefore, you must solve this with round-trip delay analysis or creating a custom data training circuitry to write and read back a training pattern to and from the memory device, and then dynamically adjust the resynchronization clock phase of the PLLs to find a good working phase.

Write Clock (write_clk)

The `write_clk` clock clocks the DDIO_OUT block that is part of the DQ output path. The clock is used for writing data to the DQ pins and also to clock the FF (oe) block (refer to Figure 1–1 on page 1–2), which is part of the DQ OE path. The FF (oe) block acts as the output enable for the DQ output path.

The `write_clk` clock should be the full rate of the interface and have a –90° phase shift. This ensures the write DQ data is center-aligned with respect to the DQS write strobe.

Memory Clock (mem_clk)

The mem_clk clocks the following blocks:

- DDIO_OUT block that is part of the DQS output path and
- DDIO_OUT blocks that generate the CK and CK# signals.
- DDIO_OE block that is part of the DQS OE path
- DDIO_OE (extended_rtena) block that is part of the dynamic termination for the DQS pin
- DDIO_OE (extended_rtena) block that is part of the dynamic termination for the DQ pin

The mem_clk should be the full rate of the interface and typically have 0° phase shift.

Address/Command Clock

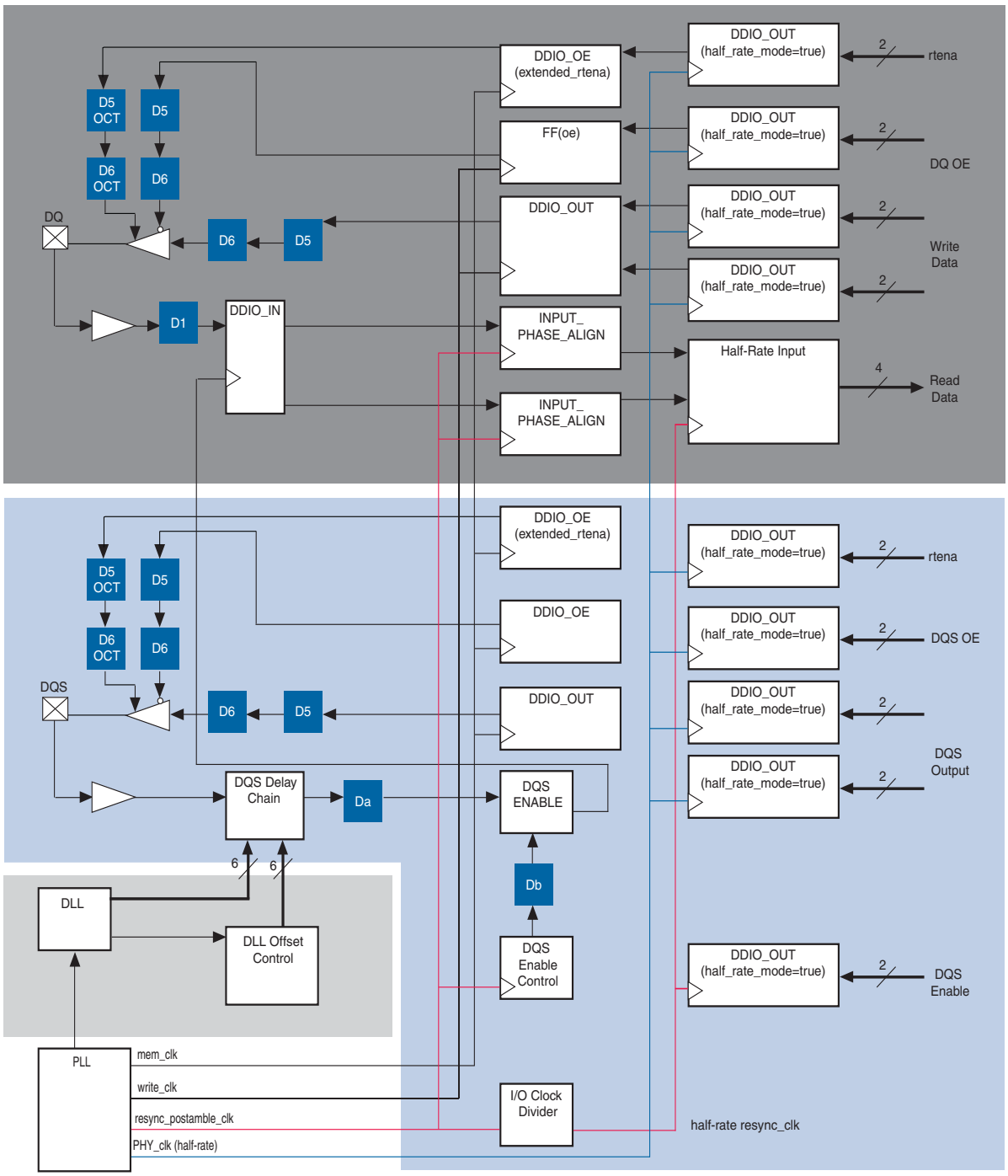
The address/command clock is used to clock the DDIO_OUT blocks that are used for the address and command pins. Although not shown in Figure 1–1 on page 1–2, it is discussed further in "Create Controller Logic for Address and Command Pins" on page 1–21.

The address/command clock should be the full rate of the interface and have a 180° phase shift. The 180° value is dependent on the address/command/control signals being perfectly matched to the clock signals in time delay on the PCB. Typically, this is not the case, but it is ideal.

Clocking Scheme Requirement for Half-Rate Interface

Figure 1–2 shows a sample half-rate interface.

Figure 1–2. Example Half-Rate Interface



The PLL instance for a half-rate interface generates the following six clock outputs (refer to Figure 1–2):

- DLL input clock

- Resynchronize postamble clock (resync_postamble_clk)
- Write clock (write_clk)
- Memory clock (mem_clk)
- PHY Clock (PHY_clk)
- Address/command clock

The following sections describe the details of the six clock outputs for the half-rate interface.

DLL Input Clock

The DLL clock is used to toggle the DLL block. The DLL clock should be equivalent to the full rate of the interface and have a 0° phase shift. You must connect the DLL block directly to a single dedicated PLL clock output.

Resynchronize Postamble Clock (resync_postamble_clk)

The resync_postamble_clk clock is an optional clock. You can use this clock for three purposes:

- Clock the DQS enable control block. The DQS enable control block is used to disable the DQS input strobe after the strobe goes to Z (after a DDR read postamble). This is part of the DQS input path.
- Clock the INPUT_PHASE_ALIGN block. This block phase shifts the input signal and is primarily used to match the arrival delay of the DQS to the latest arrival delay of a DQS from the DDR or DDR2 DIMM. It is also used to resynchronize the data read from the DDIO_IN block.
- Feed the I/O clock divider block that is a part of the DQS input path. The I/O clock divider divides the clock by two (half-rate resync_clk) and uses it to clock the DDIO_OUT (half_rate_mode=TRUE) block that is part of the DQS input path. The DQS input path controls the DQS enable signal and also clocks the half-rate input block that is part of the DQ input path.

The resync_postamble_clk clock should be the full rate of the interface and have dynamic phase. You can set the dynamic phase capability for the PLL clock outputs with the ALTPLL megafunction. The correct resync_postamble_clk clock phase is crucial to correct data transfer. The ALTDQ_DQS megafunction cannot determine the phase that the resync_postamble_clk should have. Therefore, you must solve this with RTD analysis or creating a custom data training circuitry to write and read back a training pattern to and from the memory device and then dynamically adjust the resynchronization clock phase of the PLLs to find a good working phase.

Write Clock (write_clk)

The write_clk clocks the DDIO_OUT block that is a part of the DQ output path. The DDIO_OUT block is used for writing data to the DQ pins and to clock the FF (oe) block that is a part of the DQ OE path. The FF (oe) block (refer to Figure 1–2) acts as the output enable for the DQ output path.

The write_clk should be the full rate of the interface and have about a –90° phase shift. This requirement ensures the write DQ data is center-aligned with respect to the DQS write strobe.

Memory Clock (mem_clk)

The `mem_clk` clocks the following blocks:

- DDIO_OUT block that is part of the DQS output path

  The memory clock is also used to clock the DDIO_OUT blocks that generate the `CK` and `CK#` signals.
- DDIO_OE block that is part of the DQS OE path
- DDIO_OE (`extended_rtena`) block that is part of the dynamic termination for the DQS pin
- DDIO_OE (`extended_rtena`) block that is part of the dynamic termination for the DQ pin

The memory clock should be the full rate of the interface and have about a 0° phase shift.

PHY Clock (PHY_clk)

The `PHY_clk` clocks the following blocks:

- Two DDIO_OUT (`half_rate_mode=TRUE`) blocks that are part of the DQS output path
- DDIO_OUT (`half_rate_mode=TRUE`) block that is part of the DQS OE path
- DDIO_OUT (`half_rate_mode=TRUE`) block that is part of the dynamic termination for the DQS pin
- Two DDIO_OUT (`half_rate_mode=TRUE`) blocks that are part of the DQ output path
- DDIO_OUT (`half_rate_mode=TRUE`) block that is part of the DQ OE path
- DDIO_OUT (`half_rate_mode=TRUE`) block that is part of the dynamic termination for the DQ pin

The `PHY_CLK` should be the half rate of the interface and have a 0° phase shift.

Address and Command Clock

The address and command clock clocks the DDIO_OUT blocks that are used for the address and commend pins. Although not shown in Figure 1–2 on page 1–5, it is discussed further in "Create Controller Logic for Address and Command Pins" on page 1–21.

The address and command clock should be the full rate of the interface and have a phase shift between 180° and 359°. The value is dependent on the address/command/control signals being perfectly matched to the clock signals in time delay on the PCB. Typically, this is not the case, but it is ideal.

For more information about using PLLs or other blocks, refer to the *ALTPLL Megafunction User Guide.*

After you decide on your clocking scheme as discussed in this section, you can proceed to the next step.

Instantiate ALTDLL Megafunctions

Using the MegaWizard Plus-In Manager, instatiate an ALTDLL megafunction that corresponds with the memory interface frequency. You must set the input frequency, delay chain length, and delay buffer mode for the DLL in the MegaWizard interface. Specify a DLL frequency mode that corresponds with the middle frequency range of the DLL frequency mode. This DLL frequency mode determines the delay buffer mode and delay chain length of interface. You can also specify a jitter reduction for the DLL offset control blocks.

Refer to the "DLL and DQS Logic Block Specifications" section of the the *Stratix III Device Datasheet* or the *Stratix IV Device Datasheet*.

For more information about the settings in this section, refer to the “MegaWizard Plug-In Manager Page Descriptions for the ALTDLL Megafunction” section of the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

Instantiate ALTDQ_DQS Megafunctions

After configuring the DLL settings, you must set the dedicated circuitry settings for external memory interfaces with the ALTDQ_DQS megafunction.

A typical custom external memory interface consists of the following:

- Zero or one DQS I/O pin (read or write strobe/clock) with an optional DQSn I/O pin (differential strobe/clock), or a CQ and CQn I/O pair (complementary clock)
- One or more DQ I/O pins (read or write data)
- One DM/D pin (output-only data mask, write data, or both)

The ALTDQ_DQS megafunction allows you to instantiate a group of DQ pins (ranging from ×4 to ×36) along with their corresponding DQS block, or DQSn I/O (optional) block, or both (optional). The ALTDQ_DQS megafunction enables you to generate the necessary DQ/DQS circuitry to use with external memory interfaces. You can configure the DQS and DQSn I/O as input-only, output-only, or bidirectional pins. All `bidir_dq` I/O pins are identically configured. Similarly, all `output_dq` (output-only DQ/DM I/O pins) and `input_dq` I/O pins (input-only DQ pins) are identically configured. Hence, if delay chains or half-data rate blocks are used in a path, the configuration applies to all paths of the same type. The paths can be all-input paths, all-output paths, or bidirectional paths in the ALTDQ_DQS variation. The ALTDQ_DQS megafunction generates only one DQS I/O pin per instantiation. If you need a 72-bit data interface composed of ×9 data groups, you must generate a ×9 data group with the megafunction and then instantiate it eight times.

The following steps describe the general flow when using the ALTDQ_DQS megafunction:

1. Configure the general settings for the ALTDQ_DQS instances, which includes the number of input, output, and bidirectional DQ, DQS delay chain stages, DQS input frequency, use half-rate components, and use dynamic OCT path.
2. Configure the DQS input path, which includes using input delay chain (D1), using DQS delay chain, enabling DQS busout delay chain, enabling DQS enable block, enabling DQS enable control block, and enabling DQS enable block delay chain.

3. If you enable the DQS delay chain block to true, the **DQS Delay Chain Settings** dialog box appears. You must configure the DQS delay chain settings.
4. If you enable the DQS enable control to true, the **DQS Enable Control Settings** dialog box appears. You must configure the DQS enable control settings.
5. Configure the DQS output/OE path, which includes enabling DQS output delay chain 1 (D5), enabling DQS output delay chain 2 (D6), configuring DQS output registers, enabling DQS OE delay chain 1 (D5), enabling DQS OE delay chain 2 (D6), and configuring the DQS OE registers.
6. Configure the DQ input path, which includes configuring the DQ input registers, selecting the clock source for the DQ input register, enabling the DQ input phase alignment block, use the DQ half-rate `dataoutbypass` port, and enabling the DQ input delay chain (D1).
7. If you enable the DQ input phase alignment block to true, the **DQ Input Phase Alignment Block Settings** dialog box appears. You must configure the DQ input phase alignment block settings.
8. Configure the DQ output/OE path, which includes enabling DQ output delay chain 1 (D5), enabling DQ output delay chain 2 (D6), configuring the DQ output registers, enabling DQ OE delay chain 1 (D5), enabling DQ OE delay chain 2 (D6), and configuring the DQ OE registers.
9. Configure the half-rate components, which includes configuring the IO clock divider block, source clock, creating the `io_clock_divider_masterin` input port for chaining multiple IO_CLOCK_DIVIDER blocks, enabling the `io_clock_divider_clkout` output port for clocking core registers, enabling the `io_clock_divider_slaveout` output port for chaining multiple IO_CLOCK_DIVIDER blocks, and inverting the phase through this block.
10. Configure the Dynamic OCT components, which includes enabling OCT delay chain 1(D5 OCT), enabling OCT delay chain 2(D6 OCT), and configuring OCT register mode.
11. Configure the DQS/DQSn IO,which includes configuring the DQS/DQSn pair as a differential pair or complementary pair.
12. Configure the reset ports for the ALTDQ_DQS instance. The configuration enables the asynchronous and synchronous reset ports to all the registers in the DQS and DQ datapath.

Table 1–1 shows two examples of the ALTDQ_DQS settings.

Table 1–1. ALTDQ_DQS Interface Example Settings (Part 1 of 2)

Settings	Full-Rate Interface	Half-Rate Interface
Device	Stratix III, speed grade C3	Stratix III, speed grade C3
DQS/DQSn	1 bidirectional differential DQS/DQS	1 bidirectional differential DQS/DQS
DQ Pins	8 bidirectional DQ pins	8 bidirectional DQ pins
DM Pins	1 output DM pin	1 output DM pin
DQS Frequency	155 MHz	450 MHz
Using dynamic termination path	Yes	Yes
Using half-rate blocks	No	Yes

Table 1–1. ALTDQ_DQS Interface Example Settings (Part 2 of 2)

Settings	Full-Rate Interface	Half-Rate Interface
DQS delay chain phase shift	90°	90°
delay_chain_length for DLL	12	8
DLL has offset control blocks enabled	Yes	Yes
Data rate of interface	310 Mbps	900 Mbps

MegaWizard Plug-In Manager Options for ALTDQ_DQS Megafunction for Full-Rate and Half-Rate Settings

This section provides descriptions of the options available on the individual pages of the ALTDQ_DQS MegaWizard™ Plug-In Manager for both full-rate and half-rate interfaces, as shown in Figure 1–1 on page 1–2 and Figure 1–2 on page 1–5.

On page 3 of the ALTDQ_DQS MegaWizard Plug-In Manager, you can select options on the **Parameter Settings** page as shown in Table 1–2.

Table 1–2. Parameter Settings (Part 1 of 2) ***(Note 1)***

Option *(2)*	Full-Rate Interface	Half-Rate Interface
Number of bidirectional DQ	**8**. There are 8 DQ pins for the interface.	**8**. There are 8 DQ pins for the interface.
Number of input DQ	**0**. This option is not used for the interface.	**0**. This option is not used for the interface.
Number of output DQ	**1**. This is used for the 1 DM output pin.	**1**. This is used for the 1 DM output pin.
Number of stages in dqs_delay_chain	**3**. Center the DQS strobe signal with the DQ data (DQS signal must be 90° phase shifted). This setting depends on the intended phase shifted (90°) and the DLL delay chain length (12).	**2**. Center the DQS strobe signal with the DQ data (DQS signal must be 90° phase shifted). This setting depends on the intended phase shifted (90°) and the DLL delay chain length (8).
DQS Input Frequency	Enter **155 MHz**. This value must match the input frequency for the DLL instance (via ALTDLL).	Enter **450 MHz**. This value must match the input frequency for the DLL instance (via ALTDLL).
Use half rate components	Turn off this option. This option is not applicable for the full-rate interface.	Turn on this option. Turning on this option enables the following 11 half-rate components.

Table 1–2. Parameter Settings (Part 2 of 2) *(Note 1)*

Option *(2)*	Full-Rate Interface	Half-Rate Interface
Use dynamic OCT path *(3)*	Turn on this option. Turning on this option enables the following two dynamic OCT components.	Turn on this option. Turning on this option enables the following four dynamic OCT components.

Notes to Table 1–2:

(1) For more information about these options, refer to the "ALTDQ_DQS High-Level Configuration Settings" and "MegaWizard Plug-In Manager Page Descriptions for the ALTDQ_DQS Megafunction" sections in the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

(2) You can use differential or complementary mode instead of single-ended mode for the DQS IO.

(3) With dynamic OCT, you can enable the parallel termination (R_t) during reads from the DDR and DDR2 external memory, and disable the R_t during writes from the DDR and DDR2 external memory. This significantly reduces power consumption for external memory interfaces.

Table 1–3 shows the settings needed to achieve particular phase shifts for the DQS delay chain block mentioned in Table 1–2:

Setting the DQS delay chain phase is one of the most important parts of the ALTDQ_DQS megafunction. It determines the necessary phase shift in the DQS delay chain that clocks the DDIO_IN block in the DQ input path. This requires a phase shift of 90° to center align the DQS strobe with the DQ data. Consider the following scenarios:

- Scenario One

  If you are using a Stratix III device with C3 speed grade, refer to the *Stratix III Device Handbook* for the appropriate settings.

  Because the DQS input frequency for the first scenario is 155 MHz, the device speed grade is C3, the DLL delay chain length is 12, delay buffer mode is low, and the intended phase shift is 90°. For these parameters, set the **Stages of the DQS Delay Chain** option to 3.

- Scenario Two

  For a particular DLL delay chain length, you might not get your intended phase setting.

  For example, set the DLL delay chain length to 10 and the DQS input frequency to 190 MHz. Your intended phase setting is 90°, but the closest available is a 72° phase shift, which is DQS delay stages setting of 2. Instantiate the DLL offset control blocks (ALTDLL) to introduce the necessary offset to the DQS delay chain to achieve the intended 90° phase shift.

Page 4 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQS IN Advanced Options** page. Table 1–3 shows how to configure the DQS input path of the ALTDQ_DQS instance.

Table 1–3. DQS IN Advanced Options Page (Part 1 of 3)

Option *(1)*	Full-Rate Interface	Half-Rate Interface
Enable DQS Input Path *(2)*	Turn on this option. Turning on this option gives you the access to the five blocks in the DQS input path as shown in Figure 1–1 on page 1–2	Turn on this option. Turning on this option gives you the access to the six blocks in the DQS input path as shown in Figure 1–2 on page 1–5.
Delay chain usage: (Enable dynamic delay chain)	Turn off this option. This option is not applicable for this interface.	Turn off this option. This option is not applicable for this interface.

Table 1–3. DQS IN Advanced Options Page (Part 2 of 3)

Option *(1)*	Full-Rate Interface	Half-Rate Interface
Delay chain usage: (Enable dqs_delay_chain)	Turn on this option. Turning on this option enables the DQS delay chain block.	Turn off this option. Turning off this option enables the DQS delay chain block.
Advanced Delay Chain Options (DQS Delay Chain Phase Setting Options)	Turn off the **Set dynamically using configuration registers** option.	Turn off the **Set dynamically using configuration registers** option.
Advanced Delay Chain Options (DQS Delay Chain Phase Setting Options)	Select the **DLL** option. This allows DLL to control the delay in the DQS delay chain block.	Select the **DLL** option. This allows DLL to control the delay in the DQS delay chain block.
Advanced Delay Chain Options (DQS Delay Buffer Mode)	Select the **Low** option. This must be set to low because this must match the delay buffer mode of the DLL which is used for full-rate interface.	Select the **High** option. This must be set to high because this must match the delay buffer mode of the DLL which is used for half rate interface.
Advanced Delay Chain Options (DQS Phase Shift) *(3)*	Enter **9,000** for 90°. The phase shift for the interface.	Enter **9,000** for 90°. The phase shift for the interface.
Advanced Delay Chain Options (Enable DQS offset control)	Turn on this option. This allows the offset settings (6-bit output) from the DLL offset control block to control the DQS delay chain block in.	Turn on this option. This allows the offset settings (6- bit output) from the DLL offset control block to control the DQS delay chain block.
Advanced Delay Chain Options: (Enable DQS delay chain latches)	Turn off this option. This option is not applicable for this interface.	Turn off this option. This option is not applicable for this interface.
Enable DQS busout delay chain *(2)*	Turn on this option. Turning on this option enables the Da block.	Turn on this option. Turning on this option enables the Da block**.**
Enable DQS enable block *(4)*	Turn on this option. This is used to enable or disable the DQS signal.	Turn on this option. This is used to enable or disable the DQS signal.
Enable DQS enable control block	Turn on this option. This is used to control the DQS enable block.	Turn on this option. This is to control the DQS enable block.
Advanced Enable Control Options (DQS Enable Control Phase Setting)	Select the **Set statically to '0'** option. The delay here is set to the static value of zero.	Select the **Set statically to '0'** option. The delay here is set to the static value of zero.
Advanced Enable Control Options (DQS Enable Control Invert Phase)	Select the **Never** option.	Select the **Never** option.

Table 1–3. DQS IN Advanced Options Page (Part 3 of 3)

Option *(1)*	Full-Rate Interface	Half-Rate Interface
Enable DQS enable block delay chain *(2)*	Turn on this option.	Turn on this option.

Notes to Table 1–3:

(1) For more information about these options, refer to the "Configuring the DQS Input Path" and "MegaWizard Plug-In Manager Page Descriptions for the ALTDQ_DQS Megafunction" sections in the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

(2) Da represents the DQSBUSOUT_DELAY_CHAIN block and Db represents the DQSENABLE_DELAY_CHAIN block.

(3) This setting is meant for timing analysis. This informs the TimeQuest timing analyzer to analyze the delay through the DQS delay chain block as the corresponding phase shift.

(4) The DQS enable represents the AND-gate control on the DQS input used to enable the DQS input strobe after the strobe goes to Z (after a DDR read postamble).

Page 5 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQS OUT/OE Advanced Options** page. Table 1–4 shows how to configure the DQS OUTPUT and DQS OE path of the ALTDQ_DQS instance.

Table 1–4. DQS OUT/OE Advanced Options Page

Option *(1)*	Full-Rate Interface	Half-Rate Interface
Enable DQS output path *(2)*	Turn on this option. Turning on this option gives you the access to the three blocks in the DQS output path.	Turn on this option. Turning on this option gives you the access to the three blocks in the DQS output path.
DQS Output Path Options (Enable DQS output delay chain1) *(2)*	Turn on this option. Turning on this option enables the D5 block.	Turn on this option. Turning on this option enables the D5 block.
DQS Output Path Options: (Enable DQS output delay chain2) *(2)*	Turn on this option. Turning on this option enables the D6 block.	Turn on this option. Turning on this option enables the D6 block.
DQS Output Path Options: (DQS output register mode)	Select the **DDIO** option. This sets the DDIO_OUT block that is part of the DQS output path in to a DDIO_OUT.	Select the **DDIO** option. This sets the DDIO_OUT block that is part of the DQS output path in to a DDIO_OUT.
DQS Output Enable Options (Enable DQS output enable) *(2)*	Turn on this option. Turning on this option gives you the access to the three blocks in the DQS OE path.	Turn on this option. Turning on this option gives you the access to the three blocks in the DQS OE path.
DQS Output Enable Options (Enable DQS output enable delay chain1) *(2)*	Turn on this option. Turning on this option enables the D5 block. This is part of the DQS OE path.	Turn on this option. Turning on this option enables the D5 block. This is part of the DQS OE Path.
DQS Output Enable Options (Enable DQS output enable delay chain2) *(2)*	Turn on this option. Turning on this option enables the D6 block. This is part of the DQS OE path.	Turn on this option. Turning on this option enables the D6 block. This is part of the DQS OE path.
DQS Output Enable Options (DQS output enable register mode)	Select the **DDIO** option. This sets the DDIO_OUT block this is part of the DQS OE path to a DDIO_OUT.	Select the **DDIO** option. This sets the DDIO_OUT block that is part of the DQS OE path to a DDIO_OUT.

Notes to Table 1–4:

(1) For more information about these options, refer to the "Configuration the DQS Output Path, Configuration the DQS OE Path" and "MegaWizard Plug-In Manager Page Description for the ALTDQ_DQS Megafunction" sections in the ALTDLL and ALTDQ_DQS Megafunction User Guide.

(2) D5 block is dynamic output delay chain1 and D6 block is dynamic output delay chain 2. These delay chains are configured dynamically during FPGA run-time to deskew the DQS pins, providing improved timing margins. For more information about delay chains, refer to the "Delay Chains For External Memory Interfaces" section in the ALTDLL and ALTDQ_DQS Megafunctions User Guide.

Page 6 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQ IN Advanced Options** page. You can select options based on Table 1–5. This page configures the DQ input path of the ALTDQ_DQS instance.

Table 1–5. DQ IN Advanced Options Page (Part 1 of 2)

Option *(1)*	Full-Rate Interface	Half-Rate Interface
Enable DQS Input Path *(2)*, *(4)*, *(5)*	Turn on this option. Turning on this option gives you the access to the four blocks in the DQ input path.	Turn on this option. Turning on this option gives you the access to the five blocks in the DQ input path .
DQ Input Register Options (DQ input register mode)	Select the **DDIO** option. This sets the DDIO_IN block that is part of the DQ input path.	Select the **DDIO** option. This sets DDIO_IN block that is part of the DQ input path.
DQ Input Register Options (DQ input register clock source) *(3)*	Select the **'dqs_bus_out' port** option and turn off the **'Connect DDIO clkn to DQS_BUS from complementary DQSn'** option. This is used for the DDIO_IN block.	Select the **'dqs_bus_out' port** option and turn off the **'Connect DDIO clkn to DQS_BUS from complementary DQSn'**option. This is used for the DDIO_IN.
DQ Input Register Options Use DQ input phase alignment *(4)*	Turn on this option. Turning on this option enables two INPUT_PHASE_ALIGN blocks that are part of the DQ input path. This is part of the DQ input path.	Turn on this option. Turning on this option enables two INPUT_PHASE_ALIGN blocks that are part of the DQ input path. This is part of the DQ Input Path.
Advanced DQ IPA options (DQ Input Phase Alignment Phase Setting)	Select the **Set statically to '0'** option. The delay is set to the static value of zero. This is used for the two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.	Select the **Set statically to '0'** option. The delay is set to the static value of zero. This is meant for the two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.
Advanced DQ IPA options (Add DQ Input Phase Alignment Input Cycle Delay)	Select the **Never** option. This option is only used for the two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.	Select the **Never** option. This option is only used for the two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.
Advanced DQ IPA options (Invert DQ Input Phase Alignment Phase)	Select the **Never** option. This option is only used for the two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.	Select the **Never** option. This option is only used for the two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.
Advanced DQ IPA options (Register DQ input phase alignment bypass output)	Turn off this option. This option is only used for two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.	Turn off this option. This option is only used for two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.
Advanced DQ IPA options (Register DQ input phase alignment add phase transfer)	Turn off this option. This option is only used for the two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.	Turn off this option. This option is only used for the two INPUT_PHASE_ALIGN blocks that are part of the DQ input path.
Use DQ half rate 'dataoutbypass' port *(5)*	Turn off this option. This option is only used for the half-rate input block that is part of the DQ input path.	Turn off this option. This option is only used for the half-rate input block that is part of the DQ input path.

Table 1–5. DQ IN Advanced Options Page (Part 2 of 2)

Option *(1)*	Full-Rate Interface	Half-Rate Interface
Use DQ input delay chain *(2)*	Turn on this option. Turning on this option enables the D1 block. This is part of the DQ input path.	Turn on this option. Turning on this option enables the D1 block. This is part of the DQ input path.

Notes to Table 1–5:

(1) For more information about these options, refer to the "Configuring the DQ Input Path" and "MegaWizard Plug-In Manager Page Descriptions for the ALTDQ_DQS Megafunction" sections in the *ALTDLL and ALTDQ_DQ Megafunctions User Guide*.

(2) D1 block is dynamic input delay chain 1.This delay chain can be dynamically configured during FPGA run-time to deskew the DQ pins, hence providing improved timing margins. For more information about delay chains, refer to "Delay Chains For External Memory Interfaces" section in the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

(3) You have a choice whether to clock the DQ input registers from the DQS input path (via `dqs_bus_out` port) or from the FPGA core (via `dq_input_reg_clk`).

(4) The INPUT_PHASE_ALIGN blocks represent the circuitry required to phase shift the input signal. This is primarily used to match the arrival delay of the DQS to the latest arrival delay of a DQS from the DIMM. The phase settings for the INPUT_PHASE_ALIGN blocks must match the I/O clock divider block.

(5) The half-rate input block represents the circuitry required to transfer the input signal to a half-rate clock. One of the outputs can also be switched to a direct input from the input buffer. This block is use only in half-rate interfaces.

Page 7 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQ OUT/OE Advanced Options** page. You can select options based on Table 1–6. This configures the DQS OUTPUT and DQS OE path of the ALTDQ_DQS instance.

Table 1–6. DQ OUT/OE Advanced Option Page

Option *(1)*	Full-Rate Interface	Half-Rate Interface
DQ Output Path Options (Enable DQ output delay chain1) *(2)*	Turn on this option. Turning on this option enables the D5 block. This is part of the DQ output path.	Turn on this option. Turning on this option enables the D5 block. This is part of the DQ output path.
DQ Output Path Options (Enable DQ output delay chain2) *(2)*	Turn on this option. Turning on this option enables the D6 block. This is part of the DQ output path.	Turn on this option. Turning on this option enables the D6 block . This is part of the DQ output path.
DQ Output Path Options (DQS output register mode).	Select the **DDIO** option. This sets the DDIO_OUT block that is part of the DQ output path.	Select the **DDIO** option. This sets DDIO_OUT block that is part of the DQ output path.
DQ Output Enable Options (Enable DQ output enable)	Turn on this option. Turning on this option gives you the access to the three blocks in the DQ OE path.	Turn on this option. Turning on this option gives you the access to the three blocks in the DQ OE path.
DQ Output Enable Options (Enable DQS output enable delay chain1) *(2)*	Turn on this option. Turning on this option enables D5 block. This is part of the DQ OE path.	Turn on this option. Turning on this option enables D5 block. This is part of the DQ OE path.
DQ Output Enable Options (Enable DQ output enable delay chain2) *(2)*	Turn on this option. Turning on this option enables the D6 block in. This is part of the DQ OE path.	Turn on this option. Turning on this option enables the D6 block. This is part of the DQ OE path.
DQ Output Enable Options (DQ output enable register mode).	Select the **FF** option. This sets the FF (oe) block that is part of the DQ OE path.	Select the **FF** option. This sets FF (oe) block that is part of the DQ OE path.

Notes to Table 1–6:

(1) For more information about these options, refer to the "Configuring the DQS Output Path", "Configuring the DQ OE Path", and "MegaWizard Plug-In Manager Page Descriptions for the ALTDQ_DQS Megafunction" sections in the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

(2) D5 block is a dynamic output delay chain 1 and D6 block is dynamic output delay chain 2. These delay chains can be dynamically configured during FPGA run-time to deskew the DQS pins, hence providing improved timing margins. For more information about delay chains, refer to the "Delay Chains For External Memory Interfaces" section in the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

Page 8 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **Half-Rate Advanced Options** page. You can select options based on Table 1–7. This configures the DQ input path of the ALTDQ_DQS instance.

Table 1–7. Half-Rate Advanced Options Page

Option *(1)*	Full-Rate Interface	Half-Rate Interface
IO Clock Divider Source *(2)*, *(3)*	This option is not applicable for full-rate interface.	Select the **'dqs_bus_out' port** option. This uses the clock source from the DQS input path, divides it by half, and then clocks the half-rate blocks, which are: ■ DDIO_OUT (`half_rate_mode=TRUE`) block that is part of the DQS input path, which controls the DQS enable signal ■ Half-rate input block that is part of the DQ input path. This is used for the I/O clock divider block.
Create 'io_clock_divider_masterin' input port *(2)*, *(4)*	No. Not applicable for full-rate interface.	Turn off this option. For this half-rate interface, there is only one ALTDQ_DQS instance. Therefore, I/O clock divider blocks are automatically chained in the instance depending on the amount of DQ pins. This is used for the I/O clock divider block .
Create 'io_clock_divider_clkout' output port *(2)*, *(5)*		Turn on this option. This output should clock the core registers that are getting the half-rate DQ data from the half-rate registers.
Create 'io_clock_divider_slaveout' output port *(2)*		Turn off this option. For this half-rate interface, there is only one ALTDQ_DQS instance. Therefore, I/O clock divider blocks are automatically chained in the instance depending on the amount of DQ pins. This is used for the I/O clock divider block.
IO Clock Divider Invert Phase *(2)*		Select the **Never** option. This is used for the IO_Clock_Divider block.

Notes to Table 1–7:

(1) For more information about these options, refer to the "Configuring the DQS Input Path", "I/O Clock Divider Primitive", and "MegaWizard Plug-In Manager Page Descriptions for the ALTDQ_DQS Megafunction" sections in the ALTDLL and ALTDQ_DQS Megafunctions User Guide.

(2) The I/O_Clock_Divider block represents a div-by-2 clock divider for transferring data to the core at one half the speed of the I/O input and output clock. Each divider can feed up to six pins (a x4 DQS group) in the device. To feed wider DQS groups, multiple clock dividers must be chained together by feeding the `slaveout` output of one divider to the `masterin` input of the neighboring pins' divider. The phase settings for the INPUT_PHASE_ALIGN blocks must match the I/O clock divider block.

(3) This is used to clock the IO_CLOCK_DIVIDER sub-block, which is used during ahalf-rate operation.

(4) If enabled, then the `masterin` input is used to synchronize this divider with another IO_CLOCK_DIVIDER sub-block. If disabled, then this divider operates independently (this mode is meant for the master divider for a group of dividers).

(5) Enables the clock output signal divided by 2. It can be connected to the clock input of a half-rate input block or it can feed the FPGA core.

Page 9 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **OCT Advanced Options** page. You can select options based on Table 1–8. This configures the OCT registers and delay chain settings of the ALTDQ_DQS instance.

Table 1–8. OCT Advanced Options Page

Option *(1)*	Full-Rate Interface	Half-Rate Interface
Dynamic OCT Options (Enable OCT delay chain1) *(2)*	Turn on this option. Turning on this option enables two D5 OCT blocks. This is part of the DQ and DQS dynamic termination path.	Turn on this option. Turning on this option enables two D5 OCT blocks. This is part of the DQ OE Path. This is part of the DQ and DQS dynamic termination path.
Dynamic OCT Options (Enable OCT delay chain2) *(2)*	Turn on this option. Turning on this option enables two D6 OCT blocks. This is part of the DQ and DQS dynamic termination path.	Turn on this option. Turning on this option enables two D6 OCT blocks. This is part of the DQ OE Path. This is part of the DQ and DQS dynamic termination path.
DQ Output Path Options (OCT register mode)	Select the **DDIO** option. This sets two blocks: ■ DDIO_OE (`extended_rtena`) block that is part of the dynamic termination for the DQS pin ■ DDIO_OE (`extended_rtena`) block that is part of the dynamic termination for the DQ pin.	Select the **DDIO** option. This sets four blocks: ■ DDIO_OE (`extended_rtena`) block that is part of the dynamic termination for the DQS pin ■ DDIO_OE (`extended_rtena`) block that is part of the dynamic termination for the DQ pin ■ DDIO_OUT (`half_rate_mode=TRUE`) block that is part of the dynamic termination for the DQS pin ■ DDIO_OUT (`half_rate_mode=TRUE`) block that is part of the dynamic termination for the DQ pin.

Notes to Table 1–8:

(1) For more information about these options, refer to the "Configuring the DQ/DQS OCT Path" and "MegaWizard Plug-In Manager Page Descriptions for the ALTDQ_DQS Megafunction" sections in the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

(2) The D5 OCT block is dynamic output delay chain 1 and the D6 OCT block is dynamic output delay chain 2. These delay chains are used together with the D5 block and D6 block. With Dynamic OCT, you can enable R_t during reads from the DDR and DDR2 external memory and disable R_t during writes from the DDR and DDR2 external memory. This has a benefit of significantly reducing power consumption for external memory interfaces. Hence the necessary usage of the D5 OCT block and D6 OCT block to synchronize during reads and writes from both DQ and DQS pins, and improve overall timing margins. For more information about delay chains, refer to the "Delay Chains For External Memory Interfaces" section in the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

Page 10 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQS/DQSn IO Advanced Options** page. You can select options based on Table 1–9.

Table 1–9. DQS/DQSn IO Advanced Options Page

Options *(1)*	Full-Rate Interface	Half-Rate Interface
Use DQSn I/O	Turn on this option.	Turn on this option.
DQS and DQSn IO Configuration mode *(2)*	Select **Differential Pair**. This is required for this particular scenario.	Select **Differential Pair**. This is required for this particular scenario

Notes to Table 1–9:

(1) For more information about these options, refer to the "Configuring the DQSn I/O Pin Path" and "MegaWizard Plug-In Manager Page Descriptions for the ALTDQ_DQS Megafunction" sections in the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

(2) For the "Differential pair" configuration, the DQSn I/O is configured in a differential pair along with the DQS IO. This means that the OE and OCT paths are configured for the DQSn I/O, whereas the input and output paths are shared with the DQS IO.

Page 11 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **Reset Port Advanced Options** page. You can select options based on Table 1–10.

Table 1–10. Reset Ports Advanced Options Page

Option	Full-Rate Interface	Half-Rate Interface
Create 'dqs_areset' input port	Turn on this option. Turning on this option enables asynchronous reset port for all registers in the DQS datapath.	Turn on this option. Turning on this option enables asynchronous reset port for all registers in the DQS datapath.
Create 'dqs_sreset' input port	Turn on this option. Turning on this option enables synchronous reset port for all registers in the DQS datapath.	Turn on this option. Turning on this option enables synchronous reset port for all registers in the DQS datapath.
Create 'input_dq_areset' input port	Turn off this option. This option is not applicable in this scenario because the DQ datapath is bidirectional.	Turn off this option. This option is not applicable in this scenario because the DQ datapath is bidirectional.
Create 'input_dq_sreset' input port	Turn off this option. This option is not applicable in this scenario because the DQ datapath is bidirectional.	Turn off this option. This option is not applicable in this scenario because the DQ datapath is bidirectional.
Create 'output_dq_areset' input port	Turn on this option. Turning on this option enables asynchronous reset port for all registers in the DQ datapath. This is meant for the DM pin.	Turn on this option. Turning on this option enables asynchronous reset port for all registers in the DQ datapath. This is meant for the DM pin.
Create 'output_dq_sreset' input port	Turn on this option. Turning on this option enables synchronous reset port for all registers in the DQ datapath. This is meant for the DM pin.	Turn on this option. Turning on this option enables synchronous reset port for all registers in the DQ datapath. This is meant for the DM pin.
Create 'bidir_dq_areset' input port	Turn on this option. Turning on this option enables asynchronous reset port for all registers in the DQ datapath.	Turn on this option. Turning on this option enables asynchronous reset port for all registers in the DQ datapath.
Create 'bidir_dq_sreset' input port	Turn on this option. Turning on this option enables synchronous reset port for all registers in the DQ datapath.	Turn on this option. Turning on this option enables synchronous reset port for all registers in the DQ datapath.

This completes the settings for both full-rate and half-rate scenarios for the ALTDQ_DQS instance.

Instaniate the ALTOCT Megafunction

Use of the OCT scheme in Stratix III and Stratix IV devices eliminates the need for external series or parallel termination resistors and simplifies the design of a PCB. Stratix III and Stratix IV devices support calibrated on-chip series, parallel, and dynamic termination in all I/O banks for single-ended I/O standards. OCT calibration allows you to establish an optimal termination value that compensates for impedance change due to temperature and voltage fluctuation. You can calibrate Stratix III and Stratix IV devices with user-controlled signals during device operation, or by default during device configuration. For calibrated termination, use the ALTOCT megafunction to utilize the dedicated calibration blocks for termination available in the FPGA together with the dedicated OCT in the I/O buffers with the ALTIOBUF megafunction.

For custom external memory interfaces, there are significant advantages to using both dynamic OCT and calibrated termination:

- OCT calibration allows you to establish an optimal termination value that compensates for impedance change due to temperature and voltage fluctuation.
- Simplifies the design of a PCB.
- Significant power consumption reduction for external memory interfaces. With dynamic OCT you can enable R_t during reads from the DDR and DDR2 external memory and disable R_t during writes from the DDR and DDR2 external memory and when the interface is idle.

You can use the ALTDLL and ALTDQ_DQS custom external memory interfaces to achieve these advantages.

For more information about using the dynamic calibration blocks for termination, refer to *Dynamic Calibrated On-Chip Termination (ALTOCT) Megafunction User Guide*.

For more information about using the dynamic OCT, refer to *I/O Buffer (ALTIOBUF) Megafunction User Guide*.

For more information about implementing calibrated dynamic OCT, refer to *AN 465: Implementing OCT Calibration in Stratix III Devices*.

Create Controller Logic for the Read and Write Datapaths

Consider the following details when designing the controller for these signals:

- General (applies to both full-rate and half-rate interfaces):
  - The DQ signals are edge-aligned with the DQS signal during a read from the memory and are center-aligned with the DQS signal during a write to the memory.
  - The memory controller shifts the DQ signals by –90° during a write operation to center-align the DQ and DQS signals. Center-align the PLL to the DQS signal with respect to the DQ signals during writes.
  - The memory controller delays the DQS signal during a read, to ensure that the DQ and DQS signals are center-align at the capture register. Use dedicated DQS phase-shift circuitry (ALTDLL and ALTDQ_DQS) to shift the incoming DQS signal during reads.
  - The DQS signal is generated on the positive edge of the system clock to meet the t_{DQSS} requirement. The setup (t_{DS}) and hold times (t_{DH}) of the memory device for the write DQ and DM pins are relative to the edges of DQS write signals and not the CK or CK# clock.
  - DQ and DM signals use a clock shifted –90° from the system clock to ensure that the DQS edges are centered on the DQ or DM signals when they arrive at the DDR2 SDRAM.
  - DDR2 SDRAM uses the DM pins during a write operation. Driving the DM pins low shows that the write is valid. The memory masks the DQ signals if the DM pins are driven high. The timing requirements of the DM signal at the DDR2 SDRAM input are identical to those for DQ data. The Stratix III DDR registers, clocked by the –90° shifted clock, create the DM signals. While you can use any of the I/O pins in the same bank as the associated DQS and DQ pins to generate the DM signal, Altera recommends using the spare DQ pin in the same DQS group as the respective data to minimize skew.
  - Some DDR2 SDRAM DIMMs support error correction coding (ECC) to detect and automatically correct errors in data transmission. The 72-bit DDR2 SDRAM modules contain eight ECC pins in addition to 64-data pins. Connect the eight DDR2 SDRAM device ECC pins to a single DQS or DQ group.
  - The DQS, DQ, and DM board trace lengths must be tightly matched (20 ps).
  - The DQS is bidirectional. The DQSn pins in DDR2 SDRAM devices are optional but recommended for DDR2 SDRAM designs operating at more than 333 MHz.
  - The DQ pins are also bidirectional. Regardless of interface width, DDR SDRAM always operates in ×8 mode DQS groups. DDR2 SDRAM interfaces can operate in either ×4 or ×8 mode DQS groups, which is dependant on your chosen memory device or DIMM, and is also not related to the actual interface width. The ×4 and ×8 configurations use one pair of bidirectional data strobe signals, DQS and DQSn, to capture input data.
  - Two pairs of data strobes, UDQS and UDQS# (upper byte) and LDQS and LDQS# (lower byte), are required by the ×16 configuration devices. A group of DQ pins must remain associated with its respective DQS and DQSn pins.

- Full-rate interface (refer to Figure 1–1 on page 1–2):
  - DQS, DQSn, DM, and DQ signals from FPGA core must be connected to the ALTDQ_DQS instance. The ALTDQ_DQS instance interfaces to the external memory (DDR and DDR2) for these pins.
  - Read data from the DQ input path should go through a dual-clock FIFO to transfer the data from the INPUT_PHASE_ALIGN blocks (`resync_postamble_clk` clock domain) to the FPGA core clock domain.
  - Controller has to ensure that the write data from core and DQS output path strobe are aligned to ensure that when writing data to external memory, the DQS strobe is center-aligned with the output DQ data to the external memory.
- Half-rate interface (refer to Figure 1–2 on page 1–5):
  - DQS, DQSn, DM, and DQ signals from FPGA core must be connected to the ALTDQ_DQS instance. The ALTDQ_DQS instance interfaces to the external memory (DDR and DDR2) for these pins.
  - Read data from the DQ input path should go through a dual-clock FIFO to transfer the data from the half-rate input block (half-rate `resync_clk` clock domain) to the FPGA core clock domain.
  - Controller has to make sure the write data from core and DQS output path strobe must be aligned to ensure that when writing data to external memory, the DQS strobe is center-aligned with the output DQ data to the external memory.

For more information about the timing requirements for the DQS and DQSn strobe signal, DM signal, and DQ data during reading and writing operations, refer to the respective external memory device datasheet.

Create Controller Logic for Address and Command Pins

Because this is a custom external memory interface for DDR and DDR2 interface, you must design your own custom controller to control the address and command pins from the FPGA core.

Consider the following details when designing the controller for the `CK` and `CK#` pins:

- DDR2 SDRAM components use `CK` and `CK#` signals to clock the address and command signals into the memory. Furthermore, the memory uses these clock signals to generate the DQS signal during a read through the DLL inside the memory. The DDR2 SDRAM data sheet specifies the following timings:
  - t_{DQSCK} is the skew between the `CK` or `CK#` signals and the DDR3 SDRAM-generated DQS signal
  - t_{DSH} is the DQS falling edge from `CK` rising edge hold time
  - t_{DSS} is the DQS falling edge from `CK` rising edge setup time
  - t_{DQSS} is the positive DQS latching edge to `CK` rising edge
  - t_{DQSCK} is the DQS output access time from `CK`

- The DDR2 SDRAM has a write requirement (t_{DQSS}) that states the positive edge of the DQS signal on writes must be ± 25% (± 90°) of the positive edge of the DDR2 SDRAM clock input. Therefore, the memory controller should generate the CK and CK# signals using the DDR registers in the IOE to match with the DQS signal and reduce any variations across process, voltage, and temperature. The positive edge of the DDR2 SDRAM clock, CK, is aligned with the DQS write to satisfy t_{DQSS}.

Consider the following details when designing the controller for the A (address), BA (bank address), CKE, CS#, RAS#, CAS#, WE#, and ODT pins:

1. Address and command signals in DDR2 SDRAM components are clocked into the memory device with the CK or CK# signal. These pins operate at a single data rate (SDR) using only one clock edge. The number of address pins depends on the DDR2 SDRAM device capacity. The address pins are multiplexed, so two clock cycles are required to send the row, column, and bank address. The CS, RAS, CAS, WE, CKE, and ODT pins are DDR2 SDRAM command and control pins. The DDR2 SDRAM address and command inputs do not have a symmetrical setup and hold time requirement with respect to the DDR2 SDRAM clocks, CK, and CK#.
2. In Stratix III and Stratix IV devices, the address and command clock should be one of the PLL dedicated clock outputs whose phase can be adjusted to meet the setup and hold requirements of the memory clock. The address and command clock is also typically half rate, although a full rate implementation can also be created. The command and address pins use the DDIO output circuitry to launch commands from either the rising or falling edges of the clock.
3. The chip selects (CS_N), clock enable (CKE), and ODT pins are only enabled for one memory clock cycle and can be launched from either the rising or falling edge of the address and command clock signal. The address and other command pins are enabled for two memory clock cycles and can also be launched from either the rising or falling edge of the address and command clock signal.

For more information about the timing requirements for the address and command pins during reading and writing operations, refer to the respective external memory device datasheet.

Create Controller Logic for the OCT Calibration Block

If calibrated series, parallel, or dynamic termination is used for the I/O in your design, your design requires a calibration block. This block requires a pair of RUP and RDN pins located in a bank that shares the same V_{CCIO} voltage as your memory interface. This calibration block is not required to be in the same bank or side of the device as the IOEs it is serving. To use these capabilities in the FPGA, you must use the ALTOCT megafunction.

Consider the following details when designing the memory controller to be used with the ALTOCT megafunction:

- The RUP and RDN input ports of the ALTOCT instance must be connected to the respective RUP and RDN pins in the FPGA. Note that the RUP and RDN pins are dual-purpose pins and there are multiple pairs of RUP and RDN pins for each bank.

- The memory controller controls the `calibration_request`, `s2pload`, and `calibration_wait` input ports of this ALTOCT instance, to ensure that the termination is calibrated based on the PVT (process, voltage and temperature) variation requirements of the custom external memory interface.
- The `calibration_busy` and `calibration_shift_busy` output ports of this ALTOCT instance must be used by the memory controller to determine when the calibration process of the termination is complete. This is important, because then only normal operations like reading to and writing from and to external memory can be performed. During termination calibration, the pins interfacing to the external memory must not be used for any operations.
- The `seriesterminationcontrol` and `parallelterminationcontrol` output ports of this ALTOCT instance must be connected to the `ser_term_ctrl` and `par_term_ctrl` input ports of the ALTIOBUF instance, which interfaces with the external memory pins. This is to transfer the calibrated termination settings to the I/O buffers.

For more information about using these ports in the ALTOCT megafunction, refer to *Dynamic Calibrated On-Chip Termination (ALTOCT) Megafunction User Guide*.

For more information about implementing calibrated dynamic OCT in your designs, refer to *AN 465: Implementing OCT Calibration in Stratix III Devices*.

Instantiate ALTIOBUF Megafunctions

All of the interface pins must be connected to I/O buffers via the ALTIOBUF megafunction. You must use this megafunction to enable differential capabilities for the I/O buffer which is connected to a differential DQS pin or to enable dynamic OCT capabilities for the respective interface pins.

Consider the following details when designing the memory controller to be used with the ALTIOBUF megafunction:

1. The `ser_term_ctrl` and `par_term_ctrl` input ports of this ALTIOBUF instance must be connected to the respective `seriesterminationcontrol` and `parallelterminationcontrol` output ports of the ALTOCT instance, which interfaces with the external memory pins. This is to transfer the calibrated termination settings to the I/O buffers.
2. The `dyn_term_ctrl` input ports of this ALTIOBUF instance must be connected. This port enables R_t during reads from the DDR and DDR2 external memory, and disables R_t during writes from the DDR and DDR2 external memory. This port must be connected to the output of the dynamic OCT DQ/DQS path in the ALTDQ_DQS instance. The input of this path in the ALTDQ_DQS instance should be connected to the memory controller, where it can determine when a read or write operation is done.

For more information about using these ports, refer to *I/O Buffer Megafunction (ALTIOBUF) User Guide*.

For more information about implementing calibrated dynamic OCT, refer to *AN 465: Implementing OCT Calibration in Stratix III Devices*.

Connect all Instances Used in Custom External Memory Interface

When all instances of ALTDLL, ALTPLL, ALTDQ_DQS, ALTIOBUF, ALTOCT, and the custom memory controller are completed, you must connect them in the Quartus II software.

Add Constraints

The next step in the design flow is to add the timing, location, and physical constraints related to the external memory interface. These constraints include:

- Timing constraints
- Pin locations, DQ group assignments and I/O standards
- Pin loading, termination, and drive strength assignments

Unlike the ALTMEMPHY solution that automatically generates TCL scripts for timing constraints, I/O standard settings, and DQ group assignments, the ALTDLL and ALTDQ_DQS solution requires you to manually create these constraints via the Assignment Editor in the Quartus II software, and then source these assignments to a TCL script for reusability.

Timing Constraints

The ALTDLL and ALTDQ_DQS external memory solution only supports timing analysis using the TimeQuest timing analyzer with Synopsys Design Constraints (**.sdc**) assignments. These constraints are derived from the DDR2 and DDR SDRAM data sheet and tolerances from the board layout. The ALTDLL and ALTDQ_DQS external memory solution uses TimeQuest timing constraints and the timing driven fitter to achieve timing closure.

Consider the following details when performing timing analysis external on the memory interface:

- Read timing margins for DLL-based implementation
- Write data timing analysis
- Round-trip delay calculation
- DQS postamble timing

Because external memory interfaces are essentially source-synchronous interfaces, these are timing margins that you must verify:

- Address and command setup, and hold margin
- Half-rate address and command setup, and hold margin
- FPGA Core setup and hold margin
- FPGA Core reset and removal setup and hold margin
- Write setup and hold margin
- Read capture setup and hold margin

These are explained in detail in *AN 433: Constraining and Analyzing Source-Synchronous Interfaces*.

For more information about creating timing constraints in SDC format for the TimeQuest timing analyzer, refer to the *TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

Pin Locations, DQ Group Assignments, and I/O Standard

Pin locations assignments and DQ group assignments depend on your external memory interface pins. This is specific to FPGA device used. These assignments must be done in the Assignment Editor.

For the proper assignment values for pin location assignments and DQ group assignments, refer to the *External Memory Interfaces in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook* and the *External Memory Interfaces in Stratix IV Devices* chapter in volume 1 of the *Stratix IV Device Handbook*.

I/O standard assignments depend on your external memory that interfaced to the FPGA. This is specific to the external memory used. For example, DDR interfaces use SSTL I/O Standards. These assignments must be done in the Assignment Editor.

For the proper assignment value for I/O standard assignments, refer to the *Stratix III Device I/O Features* chapter in volume 1 of the *Stratix III Device Handbook* and the *I/O Features in Stratix IV Devices* chapter in volume 1 of the *Stratix IV Device Handbook*.

Pin Loading, Termination, and Drive Strength Assignments

These assignments depend on your external memory that interfaced to the FPGA. You must execute these assignments through the Assignment Editor.

For the proper assignment values, refer to the *Board Design Guidelines* section in Volume 2 of the *Eternal Memory Interface Handbook*.

Plan Resources

This section describes planning resources.

Table 1–11 shows the pin placements that Altera recommends.

Table 1–11. Stratix III DDR and DDR2 SDRAM Pin Placement Recommendations

Signal	Pin or FPGA	Pin on Memory Devices
Data (`mem_dq`)	DQ	DQ
Data mask (`mem_dm`)	DQ *(1)*	DM
Data strobe (`mem_dqs`)	DQS or DQS	DQS or DQS#
Memory clock (`mem_clk`)	DQ, or DQ, or DQSn	CK or CK#
Address	Any user I/O *(2)*	A or BA
Command	Any user I/O *(2)*	CS#, RAS#, CAS#, WE#, CKE, or ODT

Notes to Table 1–11:

(1) The DM pins must be in the DQ group.

(2) Ensure that address and command pins are placed on the same side of the device as the memory clock pins. If OCT is used, ensure that the RUP and RDN pins are assigned correctly.

The ALTDLL and ALTDQ_DQS external memory solution do not generate pin assignments for non-memory signals such as clock sources or pin location assignments for your design. Launch the Pin Planner to make these assignments to your design.

Advanced IO Timing

As part of I/O planning, especially with high-speed designs, you must take board-level signal integrity and timing into account. When adding an FPGA device with high-speed interfaces to a board design, the quality of the signal at the far end of the board route, and the propagation delay in getting there, are vital for proper system operation.

Perform RTL or Functional Simulation

After instantiating the SDRAM high-performance controller, it generates a design example and driver for testing the memory interface.

When using the ALTDLL and ALTDQ_DQS external memory solution, unlike the ALTMEMPHY solution, it does not use the SDRAM high-performance controller to generate an example design and driver for testing the memory interface. If you use this solution, you must create your own driver circuitry to test the custom-made controller logic with the datapath created via the ALTDLL, ALTDQ_DQS, ALTOCT, ALTPLL, and ALTIOBUF megafunctions. This implies that you have to manually create your own testbench to verify the custom external memory interface.

To perform functional simulation with the ALTDLL and ALTDQ_DQS external memory solution, you must connect the simulation model of the driver, the simulation model of your design (which includes customized controller logic and ALTDLL and ALTDQ_DQS datapath), and the simulation model of their external memory in their functional simulation environment.

Compile Design and Verify Timing

After constraining your design, compile your design in the Quartus II software. Because the ALTDLL and ALTDQ_DQS external memory solution does not automatically generate timing reports to verify whether timing has been met, you must use SDC commands to manually compile the design.

After compiling your design in the Quartus II software, run the verifying timing script to produce the timing report for different paths, such as write data, read data, address and command, and core (entire interface) timing paths in your design.

The custom verifying timing script should report about margins on the following paths:

- Address and command setup and hold margin
- Half-rate address and command setup and hold margin
- Core setup and hold margin
- Core reset and removal setup and hold margin
- Write setup and hold margin

- Read capture setup and hold margin

For more information about timing analysis and reporting using the ALTDLL and ALTDQ_DQS external memory solution, refer to *AN 438: Constraining and Analyzing Timing for External Memory Interfaces*.

Adjust Constraints

The timing report of your designs show the worst case setup and hold margin for the different paths in your design. If the setup and hold margin are unbalanced, achieve a balanced setup and hold margin by adjusting the phase setting of the clocks that clock these paths.

For example, for the address and command margin, the address and command outputs are clocked by an address and command clock that can be different with respect to the system clock, which is 0º. The system clock times the clock outputs going to the memory. If the report timing script indicates that using the default phase setting for the address and command clock results in more hold time than setup time, adjust the address and command clock to be less negative than the default phase setting with respect to the system clock to ensure that there is less hold margin. Similarly, adjust the address and command clock to be more negative than the default phase setting with respect to the system clock if there is more setup margin.

2. Implementing a Custom DDR2 SDRAM Interface

This chapter describes how to implement an 8-bit wide, 150-MHz, 300-Mbps DDR2 SDRAM full-rate interface using the ALTDLL and ALTDQ_DQS megafunctions.

This design example also provides some recommended settings, such as termination scheme and drive strength settings, to simplify your design. Although the design example is specifically for the DDR2 SDRAM interface, the design flow for a DDR SDRAM interface is the same.

At the end of this chapter, you create an example design similar to the one in www.altera.com/later/later, that targets the Stratix III FPGA Development Kit, which includes a component module (MT47H32M8BP-3:B). You can adapt this design to target any other board.

Software Requirements

This chapter assumes that you have experience with the Quartus II software. In addition, ensure that you have the Quartus II software version 9.0 or later installed.

Create a Quartus II Project and Specify a Target Device

Use the New Project wizard (**File > New Project Wizard**) to create a project in the Quartus II software that targets the EP3SL150F1152-C2ES device. This example design targets the EP3SL150F1152-C2 device targeting an 8-bit wide DDR2 SDRAM interface at 150 MHz. The design uses an 8-bit wide 256-MB Micron MT47H32M8BP-3:B 333-MHz DDR2 SDRAM component.

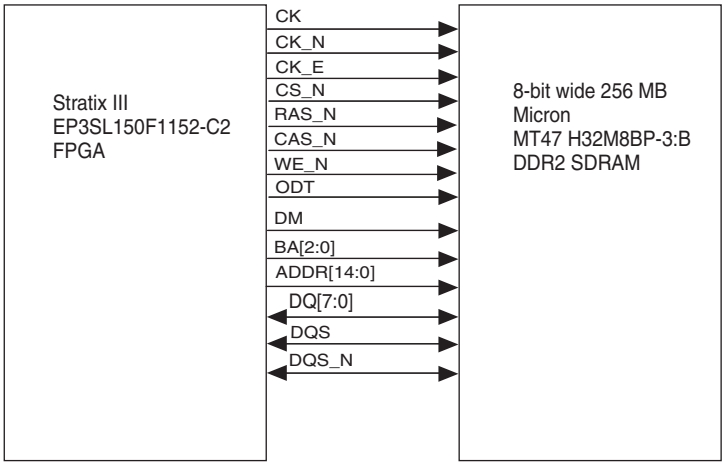
For detailed step-by-step instructions on how to create a Quartus II project, refer to the Tutorial in the Quartus II software. On the Help menu in the Quartus II window, click **Tutorial**.

Instantiate the PHY and Controller

This section describes the steps required to instantiate a custom PHY, implemented using the ALTDLL and ATLTDQ_DQS megafunctions, as well as a custom designed controller. This design example is in Verilog HDL and consists of one top-level module or instance that connects 19 sub-modules or instances for a complete customized memory controller for DDR2. All these modules or instances are defined in the following sections.

Figure 2–1 shows the example interface between the Stratix III EP3SL150F1152-C2 device and an 8-bit wide 256-MB Micron MT47H32M8BP-3:B 333-MHz DDR2 SDRAM component.

Figure 2–1. Interface Between a Stratix III Device and a DDR2 SDRAM Component



The CK, CK_N, CK_E, CS_N, RAS_N, CAS_N, WE_N, ODT, DM, BA[2:0], and ADDR[14:0] signals are output-only signals. DQ[7:0], DQS and DQS_N are bidirectional signals.

Instantiate ALTPLL Megafunctions

This design example uses the ALTPLL megafunction to implement two PLLs that clock the entire full-rate interface. Use the MegaWizard Plug-In Manager (Tools menu) to instantiate two variations of the ALTPLL megafunction with the following parameters:

- pll_1:
  - in_clock = 50 MHz
  - mode = no compensation
  - c0 = 150 MHz, 0° phase shift. Used to clock the dll_1 instance. This PLL is dedicated to clock only the DLL
- pll_2:
  - in_clock = 50 MHz
  - mode = normal
- c0 = 150 MHz, 0° phase shift. This is the mem_clk. It is used to clock the registers in the altdq_dqs_1 instance (the DQS output registers, the DQS OE registers, the DQS dynamic OCT registers, and DQ dynamic OCT registers), in the mem_clock_generate instance (the registers for generating CK and CK_N signals), the control_init_ddr instance (state machine for initialization of the DDR2), the control_write_ddr instance (state machine for writing data to the DDR2), the control_read_ddr instance (state machine for reading data from the DDR2), and

the control_driver_ddr instance (state machine for driving the control_init_ddr instance, control_write_ddr instance, and control_read_ddr instance). It is also used to clock the registers in the addr_cmd_generate and addr_cmd_generate_oe instances which are used to generate the ADDR[14:0], BA[2:0], CK_E, CS_N, RAS_N, CAS_N, WE_N, and ODT signals.

- c1 = 150 MHz, –90° phase shift. This is the write_clk. It is used to clock the registers in the altdq_dqs_1 instance (the DQ output registers and the DQ OE registers).

Instantiate the ALTDLL Megafunction

After deciding on the clocking scheme, you use the ALTDLL megafunction to instantiate a DLL of the correct operational frequency (DQS signal frequency) of the custom external memory. This frequency setting depends on the bandwidth you want.

Use the MegaWizard Plug-In Manager to instantiate a variations of the ALTPLL megafunction with the following parameters:

- dll_1:
  - turn on jitter reduction = No
  - delay chain length = 12
  - delay buffer mode = low
  - DQS input frequency = 150 MHz
  - Instantiate DLL offset control A = No
  - Instantiate DLL offset control B = No
  - create ‘dll_aload’ port = No
  - create ‘dll_dqsupdate’ port = Yes

Instantiate ALTDQ_DQS Megafunction

Next, you initialize the settings for the dedicated circuitry for external memory interfaces using the ALTDQ_DQS megafunction.

This custom DDR2 external memory interface consists of the following:

- One pair of differential DQS/DQSn I/O pin (read or write strobe/clock)
- Eight DQ I/O pins (read or write data)
- One DM pin (output-only data mask)

The following section describes how to instantiate a variation of the ALTDQ_DQS megafunction with the correct parameters.

Specify altdq_dqs_1 Parameters

This section provides descriptions of the options that must be set for the altdq_dqs_1 instance on the individual pages of the ALTDQ_DQS MegaWizard Plug-In Manager for full-rate mode.

Page 3 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **Parameter Settings** page. To configure the general settings for the ALTDQ_DQS instance, specify the options shown in Table 2–1.

Table 2–1. Parameter Settings

Option	Full-Rate Interface
Number of bidirectional DQ	**8**. There are 8 bidirectional DQ pins
Number of input DQ	**0**. Not used.
Number of output DQ	**1**. There is 1 output only DM pin.
Number of stages in dqs_delay_chain	**3**. This enables the DQS strobe signal to be centered with the DQ data (DQS signal must be +90° phase shifted)
DQS input frequency	Enter **150 MHz**
Use half-rate components	Turn off this option.
Use dynamic OCT path	Turn off this option.

Page 4 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQS IN Advanced Options** page. To configure the DQS input path of the ALTDQ_DQS instance, set the options shown in Table 2–2.

Table 2–2. DQS IN Advanced Options Page

Option	Full-Rate Interface
Enable DQS Input Path	Turn on this option.
Delay chain usage:	Select the **Enable dqs_delay_chain** option.
Advanced Delay Chain Options (DQS Delay Chain Phase Setting Options)	Turn off the **Select dynamically using configuration registers** option.
Advanced Delay Chain Options (DQS Delay Chain Phase Setting Options)	Select the **DLL** option.
Advanced Delay Chain Options (DQS Delay Buffer Mode)	Select the **Low** option.
Advanced Delay Chain Options (DQS Phase Shift)	Enter **9,000**
Advanced Delay Chain Options (Enable DQS offset control)	Turn off this option.
Advanced Delay Chain Options Enable DQS delay chain latches	Turn on this option.
Enable DQS busout delay chain	Turn on this option.
Enable DQS enable block	Turn on this option.
Enable DQS enable control block	Turn on this option.
Advanced Enable Control Options (DQS Enable Control Phase Setting)	Select the **Set statically to '0'** option.
Advanced enable control options: (DQS Enable Control Invert Phase)	Select the **Never** option.
Enable DQS enable block delay chain	Turn on this option.

Page 5 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQS OUT/OE Advanced Options** page. To configure the DQS OUTPUT and DQS OE path of the ALTDQ_DQS instance, select the options shown in Table 2–3.

Table 2–3. DQS OUT/OE Advanced Options Page

Option	Full-Rate Interface
Enable DQS output path	Turn on this option.
DQS Output Path Options (Enable DQS output delay chain1)	Turn on this option.
DQS Output Path Options (Enable DQS output delay chain2)	Turn on this option.
DQS Output Path Options (DQS output register mode)	Select the **DDIO** option.
DQS Output Enable Options (Enable DQS output enable)	Turn on this option.
DQS Output Enable Options (Enable DQS output enable delay chain1)	Turn on this option.
DQS Output Enable Options (Enable DQS output enable delay chain2)	Turn on this option.
DQS Output Enable Options (DQS output enable register mode)	Select the **FF** option.

Page 6 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQ IN Advanced Options** page. To configure the DQ input path of the ALTDQ_DQS instance, select the options shown in Table 2–4.

Table 2–4. DQ IN Advanced Options Page

Option	Full-Rate interface
DQ Input Register Options (DQ input register mode)	Select DDIO option.
DQ Input Register Options (DQ input register clock source)	Select the **'dqs_bus_out' port** option and disable the **'Connect DDIO clkn to DQS_BUS from complementary DQSn'** option.
DQ Input Register Options (Use DQ input phase alignment)	Turn on this option.
Advanced DQ IPA options: (DQ Input Phase Alignment Phase Setting)	Select the **Set statically to '0'** option.
Advanced DQ IPA options: (Add DQ Input Phase Alignment Input Cycle Delay)	Select the **Never** option.
Advanced DQ IPA options: (Invert DQ Input Phase Alignment Phase)	Select the **Never** option.
Advanced DQ IPA options: (Register DQ input phase alignment bypass output)	Turn on this option.
Advanced DQ IPA options: (Register DQ input phase alignment add phase transfer)	Turn off this option.
DQ Input Register Options (Use DQ half rate 'dataoutbypass' port)	Turn off this option.
Use DQ input delay chain	Turn on this option.

Page 7 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQ OUT/OE Advanced Options** page. To configure the DQ OUTPUT and DQ OE path of the ALTDQ_DQS instance, select the options shown in Table 2–5.

Table 2–5. DQ OUT/OE Advanced Options Page

Option	Full-Rate Interface
DQ Output Path Options (Enable DQ output delay chain1)	Turn on this option.
DQ Output Path Options (Enable DQ output delay chain2)	Turn on this option.
DQ Output Path Options (DQ output register mode)	Select the **DDIO** option.
DQ Output Enable Options (Enable DQ output enable)	Turn on this option.
DQ Output Enable Options (Enable DQ output enable delay chain1)	Turn on this option.
DQ Output Enable Options (Enable DQ output enable delay chain2)	Turn on this option.
DQ Output Enable Options (DQ output enable register mode)	Select the **FF** option.

Page 8 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **Half-rate Advanced Options** page. To configure the half-rate settings of the ALTDQ_DQS instance, select the options shown in Table 2–6.

Table 2–6. Half-Rate Advanced Options Page

Option	Full-Rate Interface
IO Clock Divider Source	Turn off this option. This option is not applicable for full-rate interface.
Create 'io_clock_divider_masterin' input port	Turn off this option. This option is not applicable for full-rate interface.
Create 'io_clock_divider_clkout' output port	Turn off this option. This option is not applicable for full-rate interface.
Create 'io_clock_divider_slaveout' output port	Turn off this option. This option is not applicable for full-rate interface.
IO Clock Divider Invert Phase	Select the **Never** option. Not applicable for full-rate interface

Page 9 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **OCT Path Advanced Options** page. To configure the OCT registers and delay chain settings of the ALTDQ_DQS instance, select the options shown in Table 2–7.

Table 2–7. OCT Path Advanced Options Page

Option	Full-Rate Interface
Dynamic OCT Options (Enable OCT delay chain 1)	Turn off this option. This option is not applicable for full-rate interface.
Dynamic OCT Options (Enable OCT delay chain 2)	Turn off this option. This option is not applicable for full-rate interface.
DQ Output Path Options (OCT register mode)	Turn off this option. This option is not applicable for full-rate interface.

Page 10 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **DQS/DQSn IO Advanced Options** page. Select options shown in Table 2–8.

Table 2–8. DQS/DQSn IO Advanced Options Page

Option	Full-Rate Interface
Use DQSn I/O	Turn on this option.
DQS and DQSn IO Configuration mode	Select the **Differential Pair** option. This is required for this particular scenario.

Page 11 of the ALTDQ_DQS MegaWizard Plug-In Manager is the **Reset Ports Advanced Options** page. Select options shown in Table 2–9.

Table 2–9. Reset Ports Advanced Options Page

Option	Full-Rate Interface
Create 'dqs_areset' input port	Turn on this option.
Create 'dqs_sreset' input port	Turn off this option.
Create 'input_dq_areset' input port	Turn off this option.
Create 'input_dq_sreset' input port	Turn off this option.
Create 'output_dq_areset' input port	Turn on this option.
Create 'output_dq_sreset' input port	Turn off this option.
Create 'bidir_dq_areset' input port	Turn on this option.
Create 'bidir_dq_sreset' input port	Turn off this option.

This completes the parameter settings for the `altdq_dqs_1` instance for this design example.

Design Customized Memory Controller Datapath Logic

Because this is a custom external memory interface for DDR2 interface, you must design the necessary datapath to control the `CK`, `CK_N`, `ADDR[14:0]`, `BA[2:0]`, `CK_E`, `CS_N`, `RAS_N`, `CAS_N`, `WE_N`, and `ODT` signals. Altera provides a design example that you can use to create your own logic.

In the Altera-provided design example, the datapath of the `CK` and `CK_N` signals are controlled by the `mem_clock_generate` instance. This instance consists of two DDIO_OUT blocks. For the `CK` signal, the inputs of the DDIO_OUT block are each tied to V_{CC} and GND. For the `CK_N` signal, the inputs of the DDIO_OUT block are each tied to GND and V_{CC} to reflect the inverse of the `CK` signal.

The `addr_cmd_generate` instance controls the datapath of the `ADDR[14:0]`, `BA[2:0]`, `CK_E`, `CS_N`, `RAS_N`, `CAS_N`, `WE_N`, and `ODT` signals. This `addr_cmd_generate` instance consists of 24 DDIO_OUT blocks to individually represent the 24 signals. For these signals, there are 24 DFF blocks to control their respective OE (output enable) signals. The OE signals are controlled by the `addr_cmd_generate_oe` instance.The inputs of these 48 blocks are fed accordingly with data, depending on the three state machines that act as the control path of the customized memory controller. The three state machines are as follows:

- `control_init_ddr` instance

- control_write_ddr instance
- control_driver_ddr instance

Multiplxer Instances

There are two multiplexer instances: cmd_addr_mux_1 and dqs_dqsn_dq_dm_mux_1.

cmd_addr_mux_1

The cmd_addr_mux_1 instance is a 144-bit to 72-bit multiplexer with a 1-bit select signal. The cmd_addr_mux_1 instance multiplexes the CK_E, CK_N, RAS_N, CAS_N, WE_N, BA_OE, BA_DATA, ADDR_OE, ADDR_DATA and ODT signals from the following two control path state machines of the customized memory controller:

- control_init_ddr instance
- control_write_ddr instance

The output of the multiplexer is sent to the data path of the command and address instances, addr_cmd_generate and addr_cmd_generate_oe. For this multiplexer, the control_driver_ddr instance controls the 1-bit select.

dqs_dqsn_dq_dmr_mux_1

The dqs_dqsn_dq_dmr_mux_1 instance is a 66-bit to 33-bit multiplexer with a 1-bit select signal. The dqs_dqsn_dq_dmr_mux_1 instance multiplexes the dm_oe, dm_data, dq_oe, dq_data, dqs_oe, dqs_data, dqsn_oe, and dqsn_data signals from the following two control path state machines of the customized memory controller:

- control_init_ddr instance
- control_write_ddr instance

The output of the multiplexer is sent to the data path of the DQS, DQSN, DQ, and DM (ALTDQ_DQS instance), which is the altdq_dqs_1 instance. The 1-bit select for this multiplexer is controlled by the control_driver_ddr instance.

This completes the steps for the customized memory controller datapath logic to generate the CK,CK_N, ADDR[14:0], BA[2:0], CK_E, CS_N, RAS_N, CAS_N, WE_N, and ODT signals.

Design Customized Memory Controller Control Path Logic

Because this is a custom external memory interface for the DDR and DDR2 interface, you must design control path logic to control the DQS, DQS_N, DQ, DM, A (address), BA (bank address), CK, CK#, CKE, CS#, RAS#, CAS#, WE#, and ODT signals from the FPGA core.

The design example contains three state machines to control these signals to enable proper operation of the external memory interface. These state machines are as follows:

- control_init_ddr instance

  The control_init_ddr instance initializes the DDR2 component for the proper interface operation following the timing requirements as specified in the specific Micron DDR2 datasheet. Because the ALTMEMPHY megafunction does not support a burst length of 8, the customized memory controller in this design example initialized the DDR2 for this mode of operation.

- control_write_ddr instance

  The control_write_ddr instance writes a set of 8 data to the memory array in the DDR2 component following the timing requirements as specified in the specific Micron DDR2 datasheet.

- control_driver_ddr instance

  The control_driver_ddr instance coordinates the two state machines (control_init_ddr and control_write_ddr instances) following the timing requirements as specified in the specific Micron DDR2 datasheet to enable proper operation of this design example. This includes sequentially enabling the following instances:

  - The control_init_ddr instance state machine to initialize the DDR2 component
  - The control_write_ddr instance state machine to write the data to the memory array in the DDR2 component

  The state machine also controls the select signals of the two multiplexers (cmd_addr_mux_1 and dqs_dqsn_dq_dm_mux_1) depending on which of the two control path state machine is currently active.

Instantiate ALTIOBUF Megafunctions for Pins

For an external memory interface, the DQ, DQS, DQSn, DM, ADDR[14:0], BA[2:0], CK, CK_N, CK_E, CS_N, RAS_N, CAS_N, WE_N, and ODT pins must be connected to I/O buffers via the ALTIOBUF megafunction. There are 14 interface signals for this design example that need I/O buffers to be interfaced with the FPGA pins. These I/O buffers are contained in the dqs_io_buffer, ck_io_buffer, dq_io_buffer, dm_io_buffer, addr_io_buffer, ba_io_buffer, and cmd_io_buffer instances. There are 35 I/O buffers used in this instance.

Table 2–10 shows the requirements that must be set for these signals in the I/O buffer instances.

Table 2–10. ddr2_io_buffer Instance Signal Requirements (Part 1 of 2)

Signal	Instance	Requirement
DQS and DQS_N	dqs_io_buffer	1-bit bidirectional I/O buffer with differential capabilities enabled
CK and CK_N	ck_io_buffer instance	2-bit output I/O buffer with differential capabilities enabled
DQ[7:0]	dq_io_buffer	8-bit bidirectional I/O bufferI/O buffer
DM	dm_io_buffer	1-bit output I/O buffer
ADDR[14:0]	addr_io_buffer	15-bit output I/O buffer
BA[2:0]	ba_io_buffer	3-bit output I/O buffer

Table 2–10. ddr2_io_buffer Instance Signal Requirements (Part 2 of 2)

Signal	Instance	Requirement
CK_E	cmd_io_buffer	1-bit output I/O buffer
CS_N	cmd_io_buffer	1-bit output I/O buffer
RAS_N	cmd_io_buffer	1-bit output I/O buffer
CAS_N	cmd_io_buffer	1-bit output I/O buffer
WE_N	cmd_io_buffer	1-bit output I/O buffer
ODT	cmd_io_buffer	1-bit output I/O buffer

Connect All the Instances That are Used in the Custom External Memory Interface

This connection is highlighted with the top-level instance of this design example that top_custom_ddr2_controller instances. It connects all instances discussed in the previous section.

Add Constraints to Design Example

After instantiating the necessary instances to create a customized DDR2 memory controller from the "Instantiate PHY (via ALTDLL and ALTDQ_DQS) and (Custom-Designed) Controller in a Quartus II Project" stage, you must generate the constraints files for the design example. Apply these constraints to the design before compilation.

☞ You must manually specify constraints because this design example does not use the ALTMEMPHY megafunction or the DDR2 SDRAM high-performance controller.

Add Timing Constraints

When you instantiate the customized DDR2 memory controller design, it does not automatically generate a timing constraint file (SDC file). You must manually create your own SDC file to constrain the timing on this design. This design example comes with a timing constraints file, **top_custom_ddr2_controller_phy_ddr_timing.sdc**.

The timing constraint file constrains the clocks on the customized DDR2 memory controller design.

To add timing constraints, perform the following steps:

1. On the Assignments menu, click **Settings**.
2. In the **Category** list, expand **Timing Analysis Settings** and select **TimeQuest Timing Analyzer**.
3. Select the **top_custom_ddr2_controller_phy_ddr_timing.sdc** file and click **Add**.
4. Click **OK**.

Set Top-Level Entity

Before compiling the design, set the top-level entity of the project. The design example top-level file is **top_custom_ddr2_controller.v**, which connects 13 other sub-modules or instances for a complete customized memory controller for DDR2.

To set the top-level file, perform the following steps:

1. Open the top-level entity file, **top_custom_ddr2_controller.v**.
2. On the Project menu, click **Set as Top-Level Entity**.

Set Optimization Technique

To ensure the remaining unconstrained paths are routed with the highest speed and efficiency, set the optimization technique to **Speed**. To set the optimization technique, perform the following steps:

1. On the Assignments menu, click **Settings**.
2. Select **Analysis & Synthesis Settings**.
3. Under **Optimization Technique,** select **Speed**.
4. Click **OK**.

Set Fitter Effort

To set the Fitter effort to Standard Fit, perform the following steps:

1. On the Assignments menu, click **Settings**.
2. Expand **Fitter Settings**.
3. Turn on **Optimize Hold Timing** and select **All Paths**.
4. Turn on **Optimize Fast Corner Timing**.
5. Under **Fitter Effort,** select **Standard Fit**.
6. Click **OK**.

Enter Pin Location Assignments

To enter the pin location assignments, perform the following steps:

1. On the Processing menu, point to **Start**, and click **Start Analysis and Synthesis**.
2. Assign all your pins, so the Quartus II software fits your design correctly and gives correct timing analysis. To assign pin locations for the Stratix III development kit, run the **top_custom_ddr2_controller_PinLocations.tcl** file, which is provided with the design example or manually assign pin locations with the Pin Planner using the Stratix III FPGA Development Kit at the Altera website.

If you are at the design exploration phase of your design cycle and do not have any PCB defined pin locations, you should still manually define an initial set of pin constraints, which can become more specific during your development process.

To manually assign pin locations, perform the following steps:

1. Open Pin Planner. On the Assignments menu, click **Pin Planner**.

2. Assign DQ and DQS pins.
   a. To select the device DQS pin groups that the design uses, assign each `DQS` pin in your design to the required DQS pin in the Pin Planner. The Quartus II Fitter then automatically places the respective DQ signals onto suitable DQ pins within each group. To see the DQS groups in Pin Planner, right click, select **Show DQ/DQS Pins**, and click **In ×8/×9 Mode**. Pin Planner shows each DQS group in a different color and with a different legend: S = DQS pin, Sbar = DQSn pin, and Q = DQ pin.

   
   Most DDR2 SDRAM devices operate in ×8/×9 mode. However, some DDR2 SDRAM devices operate in ×4 mode. Refer to your specific memory device datasheet.

   b. Select the DQ mode to match the DQ group width (number of DQ pins/number of DQS pins) of your memory device. DQ mode is not related to the memory interface width.

   
   The DQ group order and DQ pin order in each group is not important. However, you must place DQ pins in the same group as their respective strobe pin.

3. Place DM pins in their respective DQ group.
4. Place address and control command pins on any spare I/O pins ideally in the same bank or side of the device as the `CK` and `CK_N` pins.
5. Ensure you place `CK` and `CK_N` pins on differential I/O pairs for the CK/CK# pin pair. To identify differential I/O pairs, right-click in Pin Planner and select **Show Differential Pin Pair Connections**. Pin pairs show a red line between each pin pair.

   
   You must place `CK` and `CK_N` on a `DIFFIO_RX` pin pair, if your design uses differential DQS signaling.

6. Place the `clock_source` pin on a dedicated PLL clock input pin with a direct connection to this design example's PLL and DLL pair—usually on the same side of the device as your memory interface. This recommendation reduces PLL jitter, saves a global clock resource, and eases timing and fitter effort.
7. Place the `global_reset` pin (like any high fan-out signal) on a dedicated clock pin.


For more information about how to use the Quartus II Pin Planner, refer to the *I/O Management* chapter in volume 2 of the *Quartus II Handbook*.

Board Trace Delay Models

For accurate I/O timing analysis, you must specify the board trace and loading information. This information should be derived and refined during your PCB development process of pre-layout (line) simulation and finally post-layout (board) simulation.

Perform RTL or Functional Simulation (Optional)

This section describes RTL and functional simulation. To set up simulation option, perform the following steps:

1. Unzip the **top_custom_ddr2_controller_msim.zip** file to any directory on your PC. Obtain and copy the vendors memory Verilog HDL simulation model to the directory where the previous zip file was uncompressed. This can be retrieved from the Micron website. Get the **ddr2.v**, **ddr2_mcp.v**, and **ddr2_parameters.vh** memory model files from the Micron website and save them in the directory of this design example.
2. Open the memory model file (**ddr2.v**) in a text editor and add the following `define` statements to the top of the file:

   ```
   'define sg3
   'define x8
   'define MAX_MEM
   ```

   The three `define` statements prepare the DDR2 SDRAM interface model.

   The first statement specifies the memory device speed grade as –3.

   The second statement specifies the memory device width per DQS.

   The third statement says to allocate memory for every address supported by the DDR2 model.
3. Open the testbench (**tb_top_custom_ddr2_controller.v**) from the directory in a text editor, instantiate the downloaded memory model, and connect its signals to the rest of the design.
4. Start the ModelSim® software.
   a. On the File menu, click **Change Directory**.
   b. Select the folder in which you unzipped the files.
   c. On the Tools menu, click **Execute Macro**.
   d. Select the **top_custom_ddr2_controller_msim.do** file and click **Open**. This is a script file for the ModelSim-Altera software to automate the necessary settings for the simulation.
   e. Verify the results shown in the Wave window.

Compile Design and Verify Timing for Design Example

To compile the design, on the Processing menu, click **Start Compilation**.

After successfully compiling the design, run the TimeQuest timing analyzer to verify the timing based on the SDC file.For more information about the TimeQuest Timing Analyzer window, refer to the *Quartus II TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

For more information about timing analysis, refer to the *Timing Analysis* section of the *External Memory Interfaces Handbook*.

Adjust Constraints for this Design Example

If the timing margin report shows negative hold time on the address and command datapath, adjusting the clock that is regulating the address and command output registers can improve the hold margin on the address and command datapath.

Verifying Design on a Board

You have the option to verify the customized DDR2 memory controller design example. You can implement the necessary debug logic to observe the read and write activity of the external memory interface during FPGA run-time.

For more information about using the SignalTap II logic analyzer, refer to *Design Debugging Using the SignalTap II Embedded Logic Analyzer* chapter in volume 3 of the *Quartus II Handbook*

This chapter provides additional information about the document and Altera.

Document Revision History

The following table shows the revision history for this document.

Date	Version	Changes
December 2010	1.1	Maintenance release.
April 2010	1.0	Initial release.

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note to Table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *Stratix IV Design Guidelines*.
italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, *<file name>* and *<project name>*.**pof** file.

Visual Cue	Meaning
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, "Typographic Conventions."
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix `n` denotes an active-low signal. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
↵	An angled arrow instructs you to press the Enter key.
1., 2., 3., and a., b., c., and so on	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	A question mark directs you to a software help system with related information.
	The feet direct you to another document or website with related information.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
	A warning calls attention to a condition or possible situation that can cause you injury.
	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.