



Section I. About This Handbook



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

Document Version: 2.0
Document Date: July 2010

EMI_INTRO_ABOUT-2.0

Copyright © 2010 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



About This Section

Revision History	v
------------------------	---

Chapter 1. How to Use this Handbook

Introduction to Altera External Memory Interfaces	1-1
Device, Pin, and Board Layout Guidelines	1-1
Implementing Altera Memory Interface IPs	1-1
Simulation, Timing Analysis, and Debugging	1-2
Implementing a Custom PHY	1-2
Design Flow Tutorials	1-2

Chapter 2. Recommended Design Flow

Select a Device	2-3
Determine Board Layout	2-3
Perform Board-Level Simulations	2-3
Device-Side Termination	2-4
Memory-Side Termination	2-4
Adjust Termination and Drive Strength	2-4
Instantiate PHY and Controller	2-4
Verify Timing	2-5
Adjust Constraints	2-5
Perform Timing Simulation	2-5
Verify Design Functionality	2-5
Design Checklist	2-6

Chapter 3. Glossary

Additional Information

How to Contact Altera	Info-1
Typographic Conventions	Info-1

Revision History

The following table shows the revision history for this section.

Date	Version	Changes Made
July 2010	2.0	Updated for 10.0 release.
January 2010	1.1	<ul style="list-style-type: none"> ■ Improved description for <i>Implementing Altera Memory Interface IP</i> chapter. ■ Added timing simulation to flow chart and to design checklist.
November 2009	1.0	First published.

The *External Memory Interface Handbook* contains information that you require to implement an external memory interface. The handbook focuses on the Altera® solution for DDR, DDR2, DDR3 SDRAM; QDR II and QDR II+ SRAM; and RLDRAM II interfaces. The handbook is organized with a typical design flow in mind, into the following six volumes:

- “Introduction to Altera External Memory Interfaces”
- “Device, Pin, and Board Layout Guidelines”
- “Implementing Altera Memory Interface IPs”
- “Simulation, Timing Analysis, and Debugging”
- “Implementing a Custom PHY”
- “Design Flow Tutorials”

Introduction to Altera External Memory Interfaces

This volume includes this *How to Use this Handbook* chapter, recommended design flow, and a glossary. In addition, this volume includes basic information for various memory standards.

The *Specifications* section lists a complete scorecard of Altera device family support for various memory standards.

Device, Pin, and Board Layout Guidelines

This volume describes the initial steps of selecting the correct Altera device with the right resources and number of user I/O available for that interface.

The second section of the volume describes how you can select the correct termination and drive strength based on your board simulation for DDR, DDR2, and DDR3 SDRAM interfaces. It offers board results correlated with board simulation and the Altera recommended settings based on this correlation.

Implementing Altera Memory Interface IPs

This volume covers the following Altera memory IP products:

- DDR and DDR2 SDRAM High Performance Controllers
- DDR and DDR2 SDRAM High Performance Controllers II
- DDR3 SDRAM High Performance Controllers
- QDR II and QDR II+ SRAM Controllers with UniPHY
- RLDRAM II Controllers with UniPHY

Each IP has two modules: the PHY and the memory controller. The DDR, DDR2, and DDR3 SDRAM high-performance controllers use the Altera ALTMEMPHY megafunction for the PHY, which you can use standalone from the controller. The QDR II, QDR II+, and RLDRAM II IP use the Altera UniPHY PHY.

The functional description of each of these modules is described separately so you do not need to read the memory controller functional description if you are creating your own custom memory controller.

The volume also contains information on how to implement the IP, including a description of the parameterization GUI, the required constraints, and latency information. The last chapter of this volume includes timing diagrams showing the memory operations that may help you debug your system or help you create a custom memory controller.

Simulation, Timing Analysis, and Debugging

When you have implemented your external memory interface, you can use this volume for information on how to perform functional simulation, how to analyze timing, and how to debug your design.

Implementing a Custom PHY

This volume describes the steps to create a custom PHY and offers examples on some custom PHYs that are already available for some interfaces.

Design Flow Tutorials

This volume offers step-by-step tutorials in creating a memory interface for specific Altera development boards for debugging and testing. In addition, this volume also discusses special information such as how to implement external memory interfaces using SOPC builder or how to implement multiple memory interfaces. The design flow tutorials follow the flow in [“Recommended Design Flow”](#) on page 2-1.

This chapter describes the Altera-recommended design flow for successfully implementing external memory interfaces in Altera devices. Altera recommends that you create an example top-level file with the desired pin outs and all interface IP instantiated, which enables the Quartus® II software to validate your design and resource allocation before PCB and schematic sign off. Use the “[Design Checklist](#)” on [page 2-6](#), to verify whether you have performed all the recommended steps in creating a working and robust external memory interface.

[Figure 2-1](#) shows the design flow to provide the fastest out-of-the-box experience with external memory interfaces in Altera devices. This topic directs you where to find information on how to perform each step of the recommended design flow. The flow assumes that you are using Altera IP to implement the external memory interface.


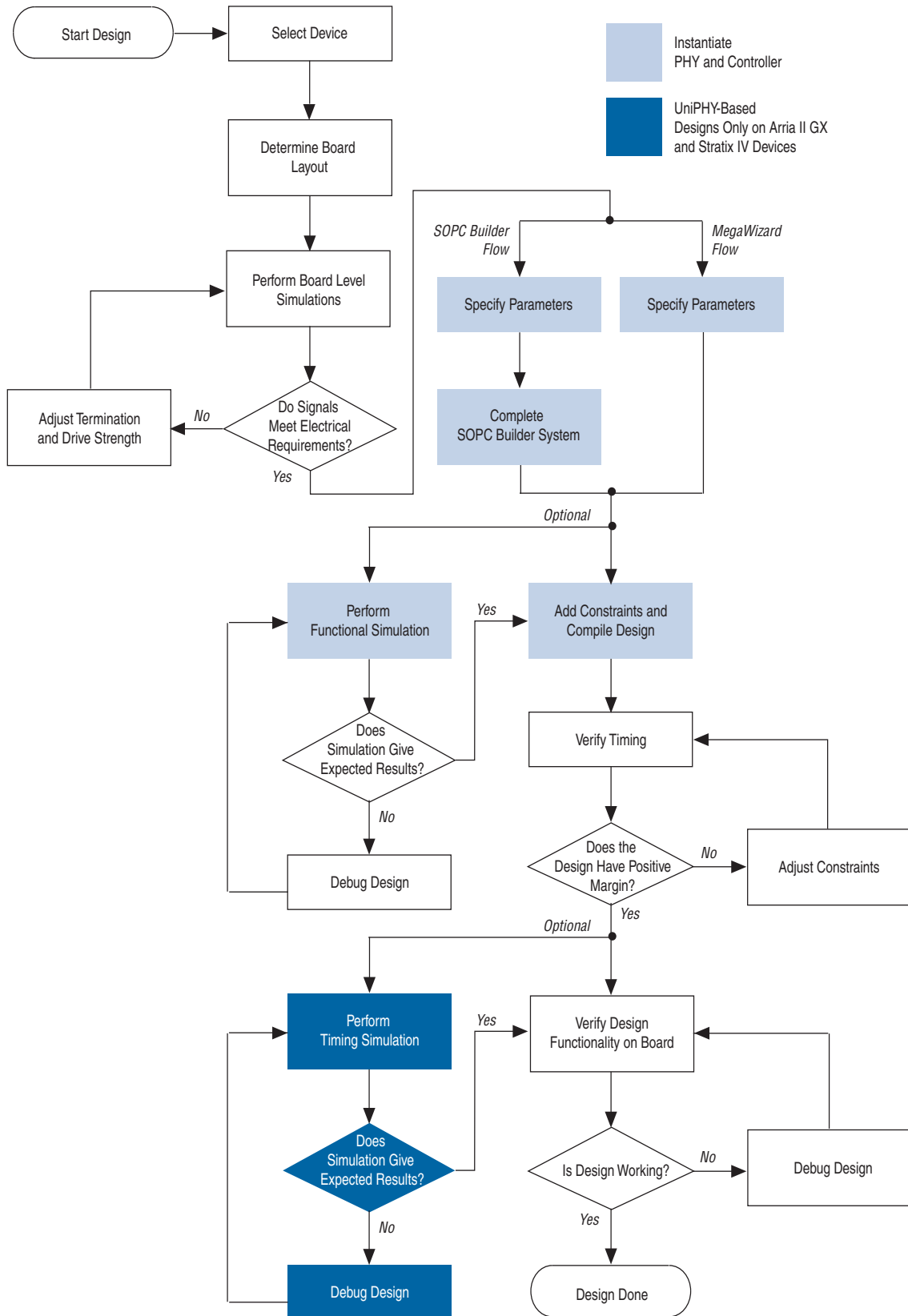
 For design examples that follow the recommended design flow in this chapter, refer to *Volume 6: Design Flow Tutorials* of the *External Memory Interface Handbook*.

Figure 2-1. External Memory Interfaces Design Flowchart



Select a Device

- For more information on selecting a device, refer to the *Device and Pin Planning* section in volume 2 of the *External Memory Interface Handbook*.

Determine Board Layout

Altera recommends prelayout SI simulations (line simulations) should take place before board layout and that you use these parameters and rules during the initial design development cycle. Advanced I/O timing and board trace models now directly impact device timing closure.

In addition, the termination scheme that you use, the drive strength setting on the FPGA, and the loading seen by the driver can directly affect the signal integrity. You must understand the tradeoffs between the different types of termination schemes and the effects of output drive strengths and loading, to choose the best possible settings for your designs.

- For more information, refer to the *Board Layout Guidelines* section in volume 2 of the *External Memory Interface Handbook*.

Perform Board-Level Simulations

To determine the correct board constraints, perform board-level simulations to see if the settings provide the optimal signal quality. With many variables that can affect the signal integrity of the memory interface, simulating the memory interface provides an initial indication of how well the memory interface performs. There are various EDA simulation tools available to perform board-level simulations. The simulations should be performed on the data, data strobe, control, command, and address signals. If the memory interface does not have good signal integrity, adjust the settings, such as drive strength setting, termination scheme or termination values to improve the signal integrity (realize that changing these settings affects the timing and it may be necessary to go back to the timing closure if these change).

- For detailed information about understanding the different effects on signal integrity design, refer to the *Board Layout Guidelines* section in volume 2 of the *External Memory Interface Handbook*.

Enter topology information from your board-level simulations into the Quartus II board trace model information. The typical information required includes, but is not limited to, the following values:

- Near and far trace lengths
- Near and far trace distributed inductance
- Near and far trace distributed capacitance
- Near end deration capacitor values (if fitted)
- Far end capacitive (IC) load
- Far end termination values

Device-Side Termination

Many Altera devices support both series and parallel OCT resistors to improve signal integrity. OCT eliminates the need for external termination resistors on the FPGA side, which simplifies board design and reduces overall board cost. You can dynamically switch between the series and parallel OCT resistor depending on whether the FPGA devices are performing a write or a read operation. The OCT features offer user-mode calibration to compensate for any variation in VT during normal operation to ensure that the OCT values remain constant. The parallel and series OCT features are available in either 25 or 50 Ω settings.

Memory-Side Termination

The DDR2, DDR3 SDRAM, and QDR II SRAM have a dynamic parallel ODT feature that you can turn on when the FPGA is writing to the memory and turn off when the FPGA is reading from the memory. To further improve signal integrity, DDR2 SDRAM supports output drive strength control so that the driver can better match the transmission line. DDR3 SDRAM devices additionally support calibrated output impedances.



For more information on available settings of the ODT, the output drive strength features, and the timing requirements for driving the ODT pin, refer to your DDR2 or DDR3 SDRAM datasheet.

Adjust Termination and Drive Strength

Although the recommended terminations are based on the simulations and experimental results, you must perform simulations, either using I/O buffer information specification (IBIS) or HSPICE models, to determine the quality of signal integrity on your designs.



Any changes made to the board should also be made in the board trace model in the Quartus II software.



For information on Altera-recommended terminations for memory interfaces, refer to the *Board Layout Guidelines* section in volume 2 of the *External Memory Interface Handbook*.



Instantiate PHY and Controller

After selecting the appropriate device and memory type, create a project in the Quartus II software that targets the device and memory type.


When implementing external memory interfaces, Altera recommends that you use Altera memory interface IP, which includes a PHY that you can use with the Altera high-performance controller or with your own custom controller.

Instantiating the PHY and controller includes the following steps:

- Specify parameters
- Perform functional simulation
- Add constraints and compile design

-  For more information about specifying parameters, adding constraints, and compiling, refer to the *DDR and DDR2 SDRAM High-Performance Controller and ALTMEMPHY IP User Guide* section and the *DDR3 SDRAM High-Performance Controller and ALTMEMPHY IP User Guide* section in volume 3 of the *External Memory Interface Handbook*.
-  For more information about simulation, refer to the *Simulation* section in volume 4 of the *External Memory Interface Handbook*.


Verify Timing

-  For more information about verifying timing, refer to the *Timing Analysis* section in volume 4 of the *External Memory Interface Handbook*.

Adjust Constraints


In the timing report of the design, you can see the worst case setup and hold margin for the different paths in the design. If the setup and hold margin are unbalanced, achieve a balanced setup and hold margin by adjusting the phase setting of the clocks associated with 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 -30° . The system clock controls 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, so 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.

-  For more information on adjusting constraints, refer to the *Timing Analysis* section in volume 4 of the *External Memory Interface Handbook*.


Perform Timing Simulation

This step is optional, but recommended to ensure that the IP is working properly. This step only applies to UniPHY-based interfaces (on Arria® II GX and Stratix® IV devices only), as ALTMEMPHY-based interfaces do not support timing simulation.

-  For more information about simulating, refer to the *Simulation* section in volume 4 of the *External Memory Interface Handbook*.

Verify Design Functionality

Perform system level verification to correlate the system against your design targets, using the Altera SignalTap® II logic analyzer.

-  For more information about using the SignalTap II analyzer, refer to the *Debugging* section in volume 4 of the *External Memory Interface Handbook*.

Design Checklist

This topic contains a design checklist that you can use when implementing external memory interfaces in Altera devices.

Done

Select Device

1. Select the memory interface frequency of operation and bus width.
For information about selecting memory, refer to the *Memory Standard Overview* section in volume 1 of the *External Memory Interface Handbook*.
2. Select the FPGA device density and package combination that you want to target.
For information about selecting an Altera device, refer to the *Device and Pin Planning* section in volume 2 of the *External Memory Interface Handbook*.
3. Ensure that the target FPGA device supports the desired clock rate and memory bus width. Also the FPGA must have sufficient I/O pins for the DQ/DQS read and write groups.
For detailed device resource information, refer to the relevant device handbook chapter on external memory interface support.
For information about supported clock rates for external memory interfaces, refer to the *External Memory Interface System Specifications* section in volume 1 of the *External Memory Interface Handbook*.

Determine Board Layout

4. Select the termination scheme and drive strength settings for all the memory interface signals on the memory side and the FPGA side.
5. Ensure you apply appropriate termination and drive strength settings on all the memory interface signals, and verify using board level simulations.
6. Use board level simulations to pick the optimal setting for best signal integrity. On the memory side, Altera recommends the use of external parallel termination on input signals to the memory (write data, address, command, and clock signals).
For information, refer to the *Board Layout Guidelines* section in volume 2 of the *External Memory Interface Handbook*.

Perform Board Level Simulations

7. Perform board level simulations, to ensure electrical and timing margins for your memory interface
8. Ensure you have a sufficient eye opening using simulations. Use the latest FPGA and memory IBIS models, board trace characteristics, drive strength, and termination settings in your simulation.
Any timing uncertainties at the board level that you calculate using simulations must be used to adjust the input timing constraints to ensure the accuracy of Quartus II timing margin reports. For example crosstalk, ISI, and slew rate deration.
For information, refer to the *Board Layout Guidelines* section in volume 2 of the *External Memory Interface Handbook*.

Instantiate PHY and Controller

Done

9. Parameterize and instantiate the Altera external memory IP for your target memory interface.
- You have the following three choices in implementing a memory interface in the Quartus II software:
- Using the Altera memory controller and PHY.
 - For information about how to implement a specific memory interface, refer to *Volume 3: Implementing Altera Memory Interface IP* of the *External Memory Interface Handbook*.
 - The *Parameter Settings* chapter of each section describes the IP supported features page by page.
 - Using the Altera PHY with your own custom controller.
 - For information about how to implement a specific memory interface, refer to *Volume 3: Implementing Altera Memory Interface IP* of the *External Memory Interface Handbook*.
 - The *Parameter Settings* chapter of each section describes the IP supported features page by page. The *Functional Description* chapter describes the workings of the PHY and how you can connect a custom controller to the PHY.
 - Implement a custom PHY and custom controller.

For information about creating custom IP, refer to *Volume 5: Implementing CustomMemory Interface PHY* of the *External Memory Interface Handbook*.
10. Ensure that you perform the following actions:
- Pick the correct memory interface data rates, width, and configurations.
 - For DDR, DDR2, and DDR3 SDRAM interfaces, ensure that you derate the tIS, tIH, tDS, and tDH parameters, as necessary.
 - Include the board skew parameter for your board.
11. Connect the PHY's local signals to your driver logic and the PHY's memory interface signals to top-level pins.
- Ensure that the local interface signals of the PHY are appropriately connected to your own logic. If the ALTMEMPHY megafunction is compiled without these local interface connections, you may encounter compilation problems, when the number of signals exceeds the pins available on your target device.
- For more information about the example top-level file, refer to the *Functional Description* chapter for the relevant memory controller.
- You may also use the example top-level file as an example on how to connect your own custom controller to the Altera memory PHY.

Perform Functional Simulation

12. Simulate your design using the RTL functional model.
- Use the IP functional simulation model with your own driver logic, testbench, and a memory model, to ensure correct read and write transactions to the memory.
- You may need to prepare the memory functional model by setting the speed grade and device bus mode.
- For more information about simulation, refer to the *Simulation* section in volume 4 of the *External Memory Interface Handbook*.

Add Constraints

Done

13. Add timing constraints. The wizard-generated **.sdc** file adds timing constraints to the interface. However, you may need to adjust these settings to best fit your memory interface configuration.
14. Add pin settings and DQ group assignments. The wizard-generated **.tcl** file includes I/O standard and pin loading constraints to your design.
15. Ensure that generic pin names used in the constraint scripts are modified to match your top-level pin names. The loading on memory interface pins is dependent on your board topology (memory components).
16. Add pin location assignments. However, you need to assign the pin location assignments manually using the Pin Planner.
17. Ensure that the example top-level file or your top-level logic is set as top-level entity.
18. Adjust optimization techniques, to ensure the remaining unconstrained paths are routed with the highest speed and efficiency:
- a. On the Assignments menu click **Settings**.
 - b. Select **Analysis & Synthesis Settings**.
 - c. Select **Speed** under **Optimization Technique**.
 - d. Expand **Fitter Settings**.
 - e. Turn on **Optimize Hold Timing** and select **All Paths**.
 - f. Turn on **Optimize Fast Corner Timing**.
 - g. Select **Standard Fit** under **Fitter Effort**.
19. Provide board trace delay model. For accurate I/O timing analysis, you specify the board trace and loading information in the Quartus II software. This information should be derived and refined during your board development process of prelayout (line) simulation and finally post-layout (board) simulation. Provide the board trace information for the output and bidirectional pins through the board trace model in the Quartus II software.

For more information, refer to the *Add Constraints* chapter for the relevant memory standard in [volume 3](#) of the *External Memory Interface Handbook* or refer to [Volume 6: Design Flow Tutorials](#).

Compile Design and Verify Timing

20. Compile your design and verify timing closure using all available models.
21. Run the wizard-generated `<variation_name>_report_timing.tcl` file, to generate a custom timing report for each of your IP instances. Run this process across all device timing models (slow 0° C, slow 85° C, fast 0° C).
22. If there are timing violations, adjust your constraints to optimize timing
23. As required, adjust PLL clock phase shift settings or appropriate timing and location assignments margins for the various timing paths within the IP.

For information, refer to the *Timing Analysis* section in volume 4 of the *External Memory Interface Handbook*.

Perform Timing Simulation

Done

24. Perform gate-level or timing simulation to ensure that all the memory transactions meet the timing specifications with the vendor's memory model. Timing simulation is only supported with UniPHY-based memory interfaces.
For more information about simulation, refer to the *Simulation* section in volume 4 of the *External Memory Interface Handbook*.

Verify Design Functionality

25. Verify the functionality of your memory interface in the system
For more information, refer to *Volume 6: Design Flow Tutorials* in the *External Memory Interface Handbook*.

This chapter lists the definitions of the terms that the Altera external memory interfaces solutions use.

Table 3–1 shows a glossary of terms.

Table 3–1. Glossary of Terms (Part 1 of 3)

Term	Description
×36 emulation	A QDR II or QDR II+ SRAM implementation to support ×36 QDR II and QDR II+ SRAM devices with Stratix III or Stratix IV devices that do not support ×36 DQS groups. Two ×18 DQS groups are combined to emulate the ×36 group. The CQ and CQn signals have to be split on the board to go to the DQS and CQn pins of the two ×18 DQS groups.
Advanced I/O timing	Advanced I/O timing allows the TimeQuest timing analyzer to produce enhanced timing reports based on the board layout information included in the design. Use advanced I/O timing with the board trace model.
Altera PHY interface (AFI)	The bus that connects calibrated PHYs with the Altera memory controller IP.
ALTMEMPHY megafunction	The Altera PHY IP that the high-performance controllers use. The latest generation PHY is the UniPHY IP.
Board trace model	The board trace model in the Quartus II software includes the board termination, trace length, and impedance in the project so that TimeQuest can produce enhanced timing reports with the included board information. You must have the advance I/O timing enabled when using he board trace model.
Calibration	The process of setting up the initial relationship between clocks. For example, the resynchronization window to provide the greatest timing margin in DDR, DDR2, and DDR3 SDRAM interfaces in Altera devices. The initial calibration is done only once at system reset after device initialization is complete.
Cascaded PLLs	A scheme whereby the clock for the PLL in the memory interface IP comes from another PLL. For more information about cascaded PLLs, refer to the <i>Device and Pin Planning</i> section in volume 2 of the <i>External Memory Interface Handbook</i> .
Cascaded	A signal topology, in which a signal is routed from one component to the next in series, but is buffered by each component.
Column address strobe (CAS)	A signal sent from a memory controller to a DRAM circuit to indicate the column address lines are valid.
Complementary clocks	A clocking scheme where only the rising edges of a clock and its inverted clock clock double data rate data into a device, for example on QDR II and QDR II+ SRAM interfaces.
Daisy-chain	A routing topology in which a single trace is routed from one component to the next, and so on, in series to the last component.
DDR	Double data-rate transfer, where data is latched and sent at both rising and falling edges of the (accompanying) clock. Operates at the full-rate clock frequency (twice the width of SDR data).
DDR3 SDRAM with leveling	A standard DDR3 SDRAM topology that requires the use of the ALTMEMPHY megafunction with read and write leveling. DDR3 SDRAM with levelling applies to any DDR3 DIMM interfaces.
DDR3 SDRAM without leveling	A non standard topology with synchronous DDR2-like balanced address, command, and clock layout.

Table 3-1. Glossary of Terms (Part 2 of 3)

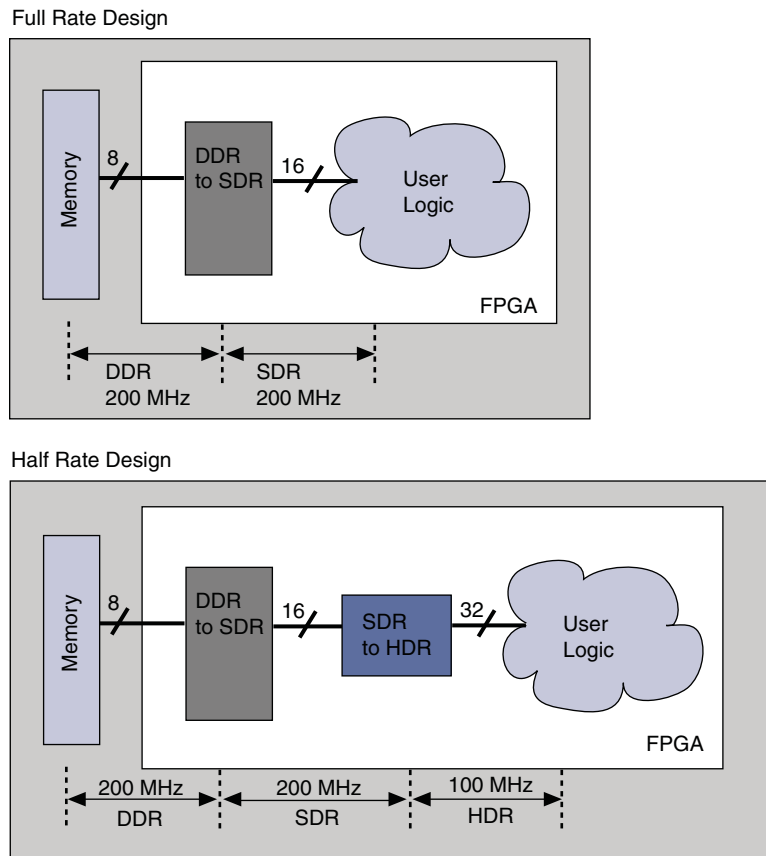
Term	Description
Deskew	Aligning the data signals with respect to the clock signal to compensate for the skew between the data and clock signals, by using the delay chains in the IOEs.
Fly-by topology	When the termination is placed after the component at the end of the line.
Full-rate clock	Clock with a frequency that is equal to the frequency of the memory interface clock.
Full-rate controller	A memory controller implemented as a full-rate design.
Full-rate design	A variation of the memory interfaces where the clock frequency of the controller and user-interface logic is the same as the memory interface clock (refer to Figure 3-1).
Half data rate (HDR)	Data that changes on one edge of the half-rate clock (twice the width of DDR data and four times the width of SDR data).
Half-rate clock	Clock with a frequency that is half the frequency of the memory interface clock.
Half-rate controller	A memory controller implemented as a half-rate design.
Half-rate design	A variation of the memory interface where the clock frequency of the controller and user-interface logic is half of the memory interface clock frequency (refer to Figure 3-1).
High-performance controller	Altera memory controller that uses the ALTMEMPHY megafunction for the datapath. Altera offers the high-performance controller for DDR, DDR2, and DDR3 SDRAM interfaces.
High-performance controller II	Latest version of the Altera memory controller for DDR, DDR2, and DDR3 interfaces. Several new features, including command look-ahead, which improves interface efficiency.
Hybrid	Obsolete term for wraparound I/Os.
Legacy controller	The legacy integrated static datapath and controller MegaCore® functions with no support for calibration and tracking. For more information about legacy integrated static datapath and controller MegaCore functions, refer to the DDR and DDR2 SDRAM Controller Compiler User Guide , QDR II SRAM Controller MegaCore Function User Guide , and RLDRAM II Controller MegaCore Function User Guide .
Memory pessimism removal (MPR)	Memory chip calibration—when the PHY calibrates some portion of the JEDEC variation.
Multiple chip select and multiple rank	Multiple chip select is the general term, whereas multiple rank is reserved for DIMMs. A rank refers to a group of DRAM components. Each rank has an individual CS signal. When there are <i>N</i> such ranks in a system, it is referred to as multiple rank.
Non-AFI	The legacy interface standard between the controller and PHY. Not recommended for new designs.
On-chip termination (OCT)	An FPGA device feature that eliminates the need for external resistors for termination. The ALTMEMPHY megafunction supports dynamic OCT for DDR2 and DDR3 SDRAM variations for Altera devices and static OCT for QDR II and QDR II+ SRAM and RLDRAM II variations.
On-die termination (ODT)	A memory vendor device feature equivalent to Altera's OCT.
Planar	Topology of some DIMM raw cards.
Quad data rate (QDR)	Two data writes and two data reads per memory clock cycle.
RLDRAM II	A DDR memory standard that has reduced latency and simpler bank management compared to the DDR, DDR2, or DDR3 SDRAM memory standard.
RLDRAM II CIO	A variant of the RLDRAM II devices that uses common I/O pins for read and write data pins.
RLDRAM II SIO	A variant of the RLDRAM II devices that uses separate I/O pins for read and write data pins.
Row address strobe (RAS)	A signal sent from a memory controller to a DRAM circuit to indicate the row address lines are valid.
R _{UP} and R _{DN} pins	Reference pins for the OCT block.
Sequencer	The logic block that performs calibration and tracking operations.

Table 3-1. Glossary of Terms (Part 3 of 3)

Term	Description
Signal splitter	The ability of the Stratix III and Stratix IV DDIO output to feed both the positive and negative legs of the differential I/O pins.
SDR	Data that changes on one edge of the full-rate clock.
T topology	A tree-type topology with balanced routing as used on DDR2 SDRAM DIMMs for address and command signals.
Tracking	Performed as a background process during device operation, to track voltage and temperature (VT) variations to maintain the data valid window that was achieved at calibration. Only applies to DDR, DDR2, and DDR3 SDRAM interfaces.
UniPHY	The latest generation Altera PHY IP.
Wraparound interface	Previously referred to as hybrid memory interface. A memory interface where the read or write or bidirectional datapath is split across the top or bottom and left or right of the device. However, a read datapath on one edge and a write datapath on an adjacent edge is not classed as a wraparound interface.
ZQ calibration	The DDR commands that calibrate the DDR3 SDRAM component ODT values.

Figure 3-1 shows the differences in the datapath width and frequency at which data is handled between full-rate and half-rate controllers.

Figure 3-1. Full-Rate and Half-Rate Controller Description



How to Contact Altera

For the most up-to-date information about Altera® products, see the following table.

Contact <i>(Note 1)</i>	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General) (Software Licensing)	Email	nacomp@altera.com
	Email	authorization@altera.com







Note:

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

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates 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, dialog box options, software utility names, and other GUI labels. For example, \qdesigns directory, d: drive, and chiptrip.gdf file.
<i>Italic Type with Initial Capital Letters</i>	Indicates document titles. For example, <i>AN 519: Stratix IV Design Guidelines</i> .
<i>Italic type</i>	Indicates variables. For example, $n + 1$. Variable names are enclosed in angle brackets (< >). For example, <file name> and <project name>.pdf file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, 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. Active-low signals are denoted by suffix n. 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).

Visual Cue	Meaning
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 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 angled arrow instructs you to press Enter.
	The feet direct you to more information about a particular topic.