



External Memory Interface Handbook

Volume 1: Altera Memory Solution Overview and Design Flow



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

EMI_GS-1.0

Document last updated for Altera Complete Design Suite version: 11.1
Document publication date: November 2011

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. 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. 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.



Chapter Revision Dates v

Section I. Altera Memory Solution Overview and Design Flow

Chapter 1. Introduction to Altera Memory Solution

Soft and Hard Memory IP 1-1
 Memory Solutions 1-2
 Low Latency 1-4
 Efficiency 1-5
 Document Revision History 1-6

Chapter 2. Recommended Design Flow

Select Your Memory 2-3
 Select Your FPGA 2-3
 Planning Pin and FPGA Resources 2-3
 Determine Board Layout 2-3
 Implementing and Parameterizing Memory IP 2-3
 Simulating Memory IP 2-4
 Analyzing Timing of Memory IP 2-4
 Perform Post-Fit Timing Simulation 2-4
 Debugging Memory IP 2-4
 Design Checklist 2-5
 Document Revision History 2-8

The chapters in this document, External Memory Interface Handbook, Volume 1: Altera Memory Solution Overview and Design Flow, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Introduction to Altera Memory Solution

Revised: *November 2011*

Part Number: *EMI_GS_001*

Chapter 2. Recommended Design Flow

Revised: *November 2011*

Part Number: *EMI_GS_002*

This section provides an overview of Altera memory solutions and the recommended memory IP design flow.

This section includes the following chapters:

- [Chapter 1, Introduction to Altera Memory Solution](#)
- [Chapter 2, Recommended Design Flow](#)



For information about the revision history for chapters in this section, refer to “Document Revision History” in each individual chapter.

This chapter describes the memory solutions that Altera provides.

Altera provides the fastest, most efficient, and lowest latency memory controllers. The controllers are designed to allow you to easily interface with today's higher speed memories.

Altera supports a wide variety of memory interfaces suitable for applications ranging from routers and switches to video cameras. You can easily implement Altera's intellectual property (IP) using the memory MegaCore functions through the Quartus II software. The Quartus II software also provides an external memory toolkit that helps you test the implementation of the IP in the FPGA device.

-  Refer to the [External Memory Interface Spec Estimator](#) page for the maximum speed that supported by Altera FPGAs.

Soft and Hard Memory IP

Altera's latest devices, the 28-nm FPGAs provide two types of memory solutions: soft memory IP and hard memory IP. Arria V and Cyclone V devices offer both soft and hard memory IP, while Stratix V devices offer only soft memory IP.

The soft memory IP gives you the flexibility to design your own interfaces to meet your system requirements and still benefit from the industry leading performance. The hard memory IP is designed to give you a complete out-of-the-box experience when designing a memory controller.

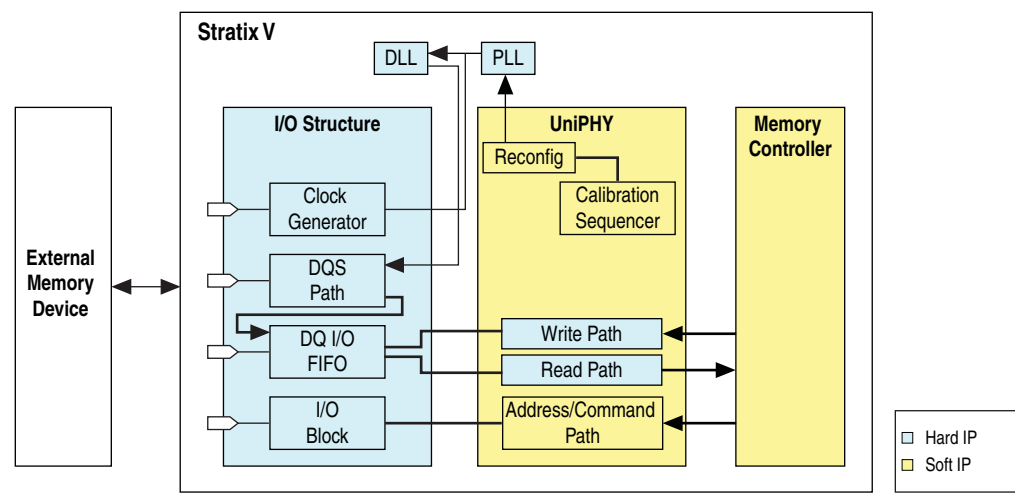
Table 1-1 lists the features of the soft and hard memory IP.

Table 1-1. Features of the Soft and Hard Memory IP

Soft Memory IP	Hard Memory IP
<ul style="list-style-type: none"> ■ Consists of a DDR2 or DDR3 SDRAM high-performance memory controller with UniPHY IP. ■ Has hardened read and write data paths to ensure your design meets timing at the highest speeds. The data paths include I/O, phase-locked loops (PLLs), delay-locked loop (DLL), and read and write FIFO buffers. ■ Allows you to choose the location to place the memory controller and the ability to size the memory controller based on the system requirements, especially in the Stratix V devices. 	<ul style="list-style-type: none"> ■ Consists of a DDR2 or DDR3 SDRAM high-performance memory controller with a hard UniPHY IP, and a multiport front-end block. ■ Has a fixed location on the die and a fixed maximum width; $\times 32$ for Arria V devices and $\times 16$ for Cyclone V devices. ■ Runs at full rate to allow decreased latency and to minimize the required bus width of signals going into the core of the device. ■ Simplifies the overall memory design in Arria V and Cyclone V devices, and provides a truly out-of-the-box experience for every designer.

Figure 1-1 shows the hardened data paths in the soft memory IP of a Stratix V device.

Figure 1-1. Hardened Data Paths in the Soft Memory IP



Memory Solutions

Altera FPGAs achieve optimal memory interface performance with external memory IP. The IP provides the following components:

- Physical layer interface (PHY) which handles the timing on the data path itself.
- Memory controller block which implements all the memory commands and addresses.
- Multiport front-end (MPFE) block which allows multiple processes inside the FPGA device to share a common bank of memory. The MPFE block is a new feature in Arria V and Cyclone V devices.

These blocks are critical to the design and the use of the memory interface block.

Altera provides modular memory solutions that allow you to customize your memory interface design to any of the following configurations:

- PHY with your own controller
- PHY with Altera controller
- PHY with Altera controller and the MPFE block

You can also build a custom PHY, a custom controller, or both, as desired.

Table 1–2 shows the recommended memory types and controllers that Altera offers with the PHY IP.

Table 1–2. Altera Memory Types, PHY, and Controllers in the Quartus II Software (Part 1 of 2)


Quartus II Version	Memory	PHY IP	Controller IP
11.1	DDR/DDR2/DDR3	ALTMEMPHY (AFI) ⁽¹⁾	HPC II
	DDR2/DDR3	UniPHY	HPC II
	QDR II/QDR II+	UniPHY	QDR/RLD II controller
	RLDRAM II	UniPHY	QDR/RLD II controller
	Other	ALTDQ_DQS ⁽²⁾	Custom
	Other	ALTDQ_DQS2 ⁽³⁾	Custom
11.0	DDR/DDR2/DDR3	ALTMEMPHY (AFI)	HPC II
	DDR2/DDR3	UniPHY	HPC II
	QDR II/QDR II+	UniPHY	QDR/RLD II controller
	RLDRAM II	UniPHY	QDR/RLD II controller
	Other	ALTDQ_DQS ⁽²⁾	Custom
	Other	ALTDQ_DQS2 ⁽³⁾	Custom
10.1	DDR/DDR2/DDR3	ALTMEMPHY (AFI)	HPC HPC II
	DDR2/DDR3	UniPHY Nios-based Sequencer	HPC II
	QDR II/QDR II+	UniPHY RTL Sequencer	QDR/RLD II controller
	RLDRAM II	UniPHY RTL Sequencer	QDR/RLD II controller
	Other	ALTDQ_DQS ⁽²⁾	Custom
	Other	ALTDQ_DQS2 ⁽³⁾	Custom
10.0	DDR/DDR2/DDR3	ALTMEMPHY (AFI)	HPC HPC II
	DDR2/DDR3	UniPHY Nios-based Sequencer	HPC II
	QDR II/QDR II+	UniPHY RTL Sequencer	QDR/RLD II controller
	RLDRAM II	UniPHY RTL Sequencer	QDR/RLD II controller
	Other	ALTDQ_DQS ⁽²⁾	Custom
	Other	ALTDQ_DQS2 ⁽³⁾	Custom

Table 1-2. Altera Memory Types, PHY, and Controllers in the Quartus II Software (Part 2 of 2)

Quartus II Version	Memory	PHY IP	Controller IP
9.1	DDR/DDR2/DDR3	ALTMEMPHY (AFI)	HPC HPC II
	QDR II/QDR II+	UniPHY	QDR II controller
	RLDRAM II	UniPHY	RLDRAM II controller
	Other	ALTDQ_DQS ⁽²⁾	Custom

Note to Table 1-2:

- (1) AFI = Altera PHY interface
- (2) Applicable for Arria II, Stratix III, and Stratix IV devices.
- (3) Applicable only for Arria V and Stratix V devices.

 For more information about the controllers with the UniPHY or the ALTMEMPHY IP, refer to the *Functional Descriptions* section in **Volume 3** of the *External Memory Interface Handbook*.

For more information about the ALTDQ_DQS megafunction, refer to the *ALTDLL and ALTDQ_DQS Megafunctions User Guide*.

For more information about the ALTDQ_DQS2 megafunction, refer to the *ALTDQ_DQS2 Megafunction User Guide*.

For more information and design example about custom PHY, refer to the [Design Example - Stratix III ALTDQ DQS DDR2 SDRAM](#) page.

Low Latency

Altera generally offers low latency solutions that are drastically better than Altera's competitors. Altera's 28-nm FPGA devices have a balanced clocked network in the periphery to reduce switching noise. The hardened read data FIFO buffer guarantees timing and makes it easier for the fitter to place the controller. Together with the latest UniPHY IP, these design changes provide drastic reduction in latency.

[Table 1-3](#) shows latency comparison for Altera and its closest competition.

Table 1-3. Latency Comparison for Quarter-Rate DDR3 SDRAM Controllers

Latency Type	Latency (Memory Clock Cycles)		Advantage
	Competitor ⁽¹⁾	Altera	
Write Command	46	29	Altera
Read Command	46	29	Altera
Read Data	31	11	Altera

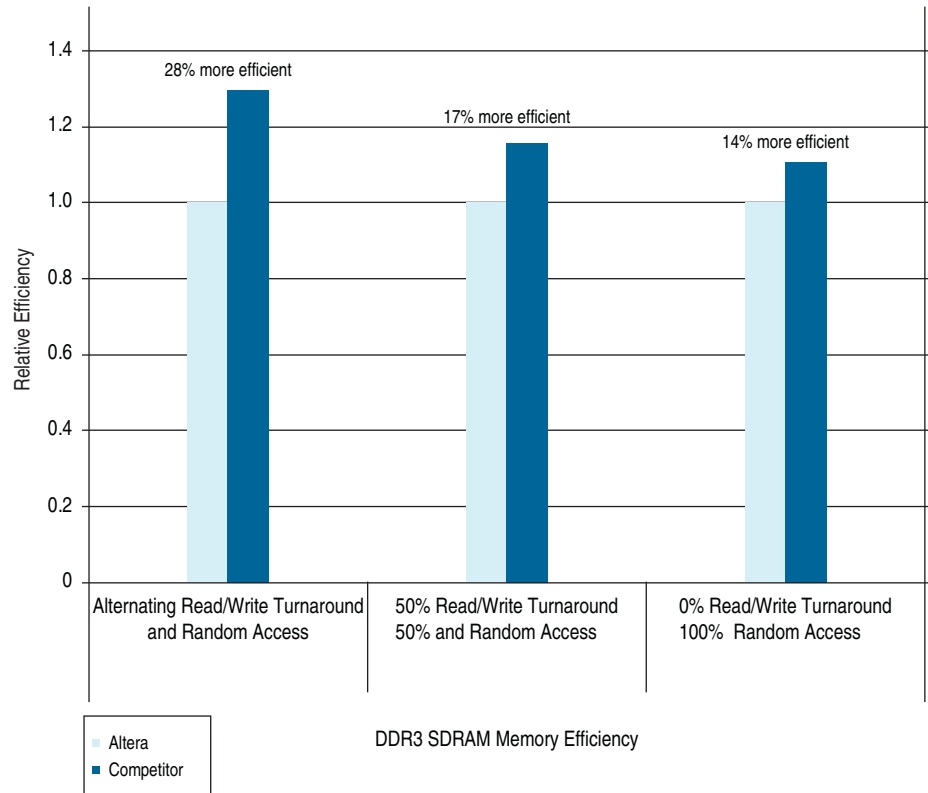
Note to Table 1-3:

- (1) Does not include AXI latency.

Efficiency

Altera memory controllers are also highly efficient. Figure 1-2 shows the memory efficiency of a DDR3 SDRAM memory controller with UniPHY IP.

Figure 1-2. Memory Efficiency of DDR3 SDRAM Memory Controllers with UniPHY



Document Revision History

Table 1–4 shows the revision history for this document.

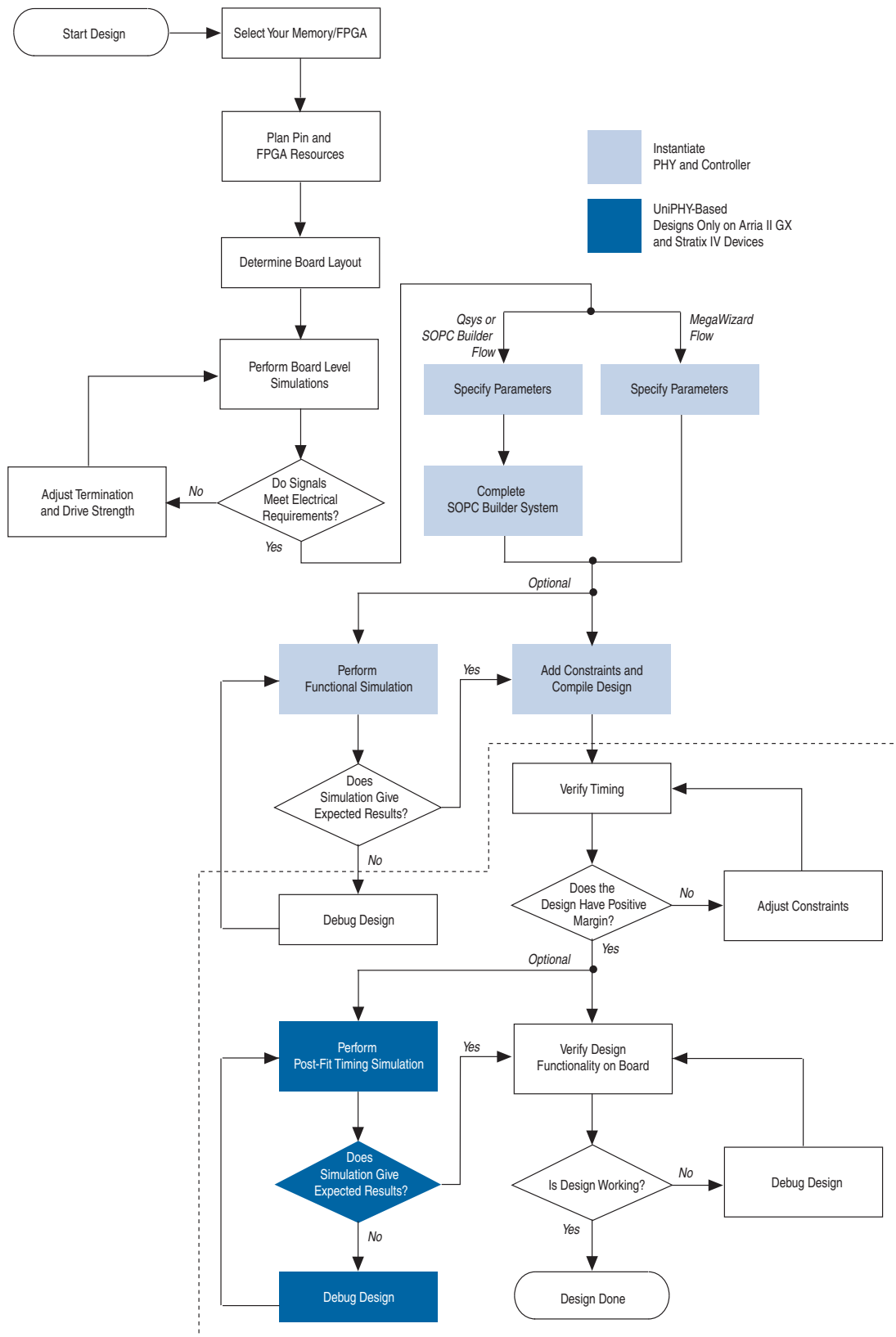
Table 1–4. Document Revision History

Date	Version	Changes
November 2011	1.0	Initial release.

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 2–5, 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.

Figure 2-1. External Memory Interfaces Design Flowchart




Select Your Memory

When you select an external memory device, you have to consider factors like bandwidth, data storage, latency and power consumption.

-  For more information about selecting your memory device, refer to the *Selecting Your Memory* chapter in the *External Memory Interface Handbook*.

Select Your FPGA

Different Altera FPGAs support different memory types and configurations. Depending on the requirements of your design, you need to determine the appropriate FPGA.

-  For more information about selecting your device, refer to the *Selecting Your FPGA* chapter in the *External Memory Interface Handbook*.


Planning Pin and FPGA Resources

Before determining the board layout, you need to determine the usage of FPGA pins, phase-locked loop (PLL), delay-locked loops (DLLs), and other resources.

-  For more information about planning pins and resources, refer to the *Planning Pin and FPGA Resources* chapter in the *External Memory Interface Handbook*.

Determine Board Layout

To improve the signal integrity, you have to consider the termination scheme that you use, the drive strength setting on the FPGA, and the loading seen by the driver. 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 about guidelines to determine your board layout for the different memory controllers, refer to the following chapters in the *External Memory Interface Handbook*:

- *DDR2 and DDR3 SDRAM Board Design Guidelines*
- *Dual-DIMM DDR2 and DDR3 SDRAM Board Design Guidelines*
- *RLDRAM II Board Design Guidelines*
- *QDR II SRAM Board Design Guidelines*

Implementing and Parameterizing Memory IP


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 and parameterizing 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.

-  For more information about specifying parameters, refer to the *Implementing and Parameterizing Memory IP* chapter in the *External Memory Interface Handbook*.


Simulating Memory IP

After implementing and parameterizing the memory IP, you need to perform functional simulation.

-  For more information about simulation, refer to the *Simulating Memory IP* chapter in the *External Memory Interface Handbook*.


Analyzing Timing of Memory IP

To ensure your external memory interface meets the various timing requirements, you need to analyze the timing paths, adjust constraints, and verify timing.

-  For more information about analyzing timing, adjust constraints, and verify timing, refer to the *Analyzing Timing of Memory IP* chapter in the *External Memory Interface Handbook*.

Perform Post-Fit Timing Simulation

This step is optional. It ensures that the IP is working properly.

-  For more information about simulating, refer to the *Simulating Memory IP* chapter in the *External Memory Interface Handbook*.

Debugging Memory IP

You need to 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 Memory IP* chapter in 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	Action	Reference
	Select Your Memory	
1.	<input type="checkbox"/> Select the memory interface frequency of operation and bus width.	■ Selecting Your Memory chapter in the <i>External Memory Interface Handbook</i> .
	Select Your FPGA	
2.	<input type="checkbox"/> Select the FPGA device density and package combination that you want to target.	■ Selecting Your FPGA chapter in the <i>External Memory Interface Handbook</i> .
	Plan Pin and FPGA Resources	
3.	<input type="checkbox"/> 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.
	Determine Board Layout	
4.	<input type="checkbox"/> Select the termination scheme and drive strength settings for all the memory interface signals on the memory side and the FPGA side.	■ DDR2 and DDR3 SDRAM Board Design Guidelines chapter in the <i>External Memory Interface Handbook</i> .
5.	<input type="checkbox"/> Ensure you apply appropriate termination and drive strength settings on all the memory interface signals, and verify using board level simulations.	■ Dual-DIMM DDR2 and DDR3 SDRAM Board Design Guidelines chapter in the <i>External Memory Interface Handbook</i> .
6.	<input type="checkbox"/> 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).	■ RLDRAM II Board Design Guidelines chapter in the <i>External Memory Interface Handbook</i> .
7.	<input type="checkbox"/> Perform board level simulations, to ensure electrical and timing margins for your memory interface	■ QDR II SRAM Board Design Guidelines chapter in the <i>External Memory Interface Handbook</i> .
8.	<input type="checkbox"/> 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.	
	Parameterize and Implement the Memory IP	
9.	<input type="checkbox"/> Parameterize and instantiate the Altera external memory IP for your target memory interface.	■ Implementing and Parameterizing Memory IP chapter in the <i>External Memory Interface Handbook</i>

	Done	Action	Reference
10.	<input type="checkbox"/>	Ensure that you perform the following actions: <ul style="list-style-type: none"> ■ 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 parameters for your board. 	
11.	<input type="checkbox"/>	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 IP 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. 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.	<ul style="list-style-type: none"> ■ <i>Functional Description: HPC II</i> chapter in the <i>External Memory Interface Handbook</i>. ■ <i>Functional Description: QDR II and QDR II+ SRAM Controller</i> chapter in the <i>External Memory Interface Handbook</i>. ■ <i>Functional Description: RLDRAM II Controller</i> chapter in the <i>External Memory Interface Handbook</i>.
		Perform Functional Simulation	
12.	<input type="checkbox"/>	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.	<ul style="list-style-type: none"> ■ <i>Simulating Memory IP</i> chapter in the <i>External Memory Interface Handbook</i>
		Add Constraints	
13.	<input type="checkbox"/>	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.	<input type="checkbox"/>	Add pin settings and DQ group assignments. The wizard-generated .tcl file includes I/O standard and pin loading constraints to your design.	
15.	<input type="checkbox"/>	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.	<input type="checkbox"/>	Add pin location assignments. However, you need to assign the pin location assignments manually using the Pin Planner.	
17.	<input type="checkbox"/>	Ensure that the example top-level file or your top-level logic is set as top-level entity.	

	Done	Action	Reference
18.	<input type="checkbox"/>	Adjust optimization techniques, to ensure the remaining unconstrained paths are routed with the highest speed and efficiency: <ol style="list-style-type: none"> On the Assignments menu click Settings. Select Analysis & Synthesis Settings. Select Speed under Optimization Technique. Expand Fitter Settings. Turn on Optimize Hold Timing and select All Paths. Turn on Optimize Fast Corner Timing. Select Standard Fit under Fitter Effort. 	
19.	<input type="checkbox"/>	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.	
Compile Design and Verify Timing			
20.	<input type="checkbox"/>	Compile your design and verify timing closure using all available models.	
21.	<input type="checkbox"/>	Run the wizard-generated <code><variation_name>_report_timing.tcl</code> 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.	<input type="checkbox"/>	If there are timing violations, adjust your constraints to optimize timing	
23.	<input type="checkbox"/>	As required, adjust PLL clock phase shift settings or appropriate timing and location assignments margins for the various timing paths within the IP.	<ul style="list-style-type: none"> ■ Analyzing Timing of Memory IP chapter in the <i>External Memory Interface Handbook</i>
Perform Post-Fit Timing Simulation			
24.	<input type="checkbox"/>	Perform post-fit timing simulation to ensure that all the memory transactions meet the timing specifications with the vendor's memory model.	<ul style="list-style-type: none"> ■ Simulating Memory IP chapter in the <i>External Memory Interface Handbook</i>.
Verify Design Functionality			
25.	<input type="checkbox"/>	Verify the functionality of your memory interface in the system	<ul style="list-style-type: none"> ■ Debugging Memory IP chapter in the <i>External Memory Interface Handbook</i>

Document Revision History

Table 2-1 shows the revision history for this document.

Table 2-1. Document Revision History

Date	Version	Changes
November 2011	2.1	Updated the design flow and the design checklist.
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.