External Memory Interface Handbook Volume 1: Altera Memory Solution Overview and Design Flow

Last updated for Quartus Prime Design Suite: 15.1
EMI_GS
2015.11.02

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

Subscribe
Send Feedback

Contents

Introduction to Altera Memory Solution 1

2015.11.02

EMI_GS Subscribe Send Feedback

The following topics provide an overview of Altera's External Memory Interface solutions.

Altera provides the fastest, most efficient, and lowest latency memory interface IP cores. Altera's external memory interface IP is designed to easily interface with today's higher speed memory devices.

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 Prime software. The Quartus Prime software also provides external memory toolkits that help you test the implementation of the IP in the FPGA device.

Refer to the External Memory Interface Spec Estimator page for the maximum speeds supported by Altera FPGAs.

Related Information

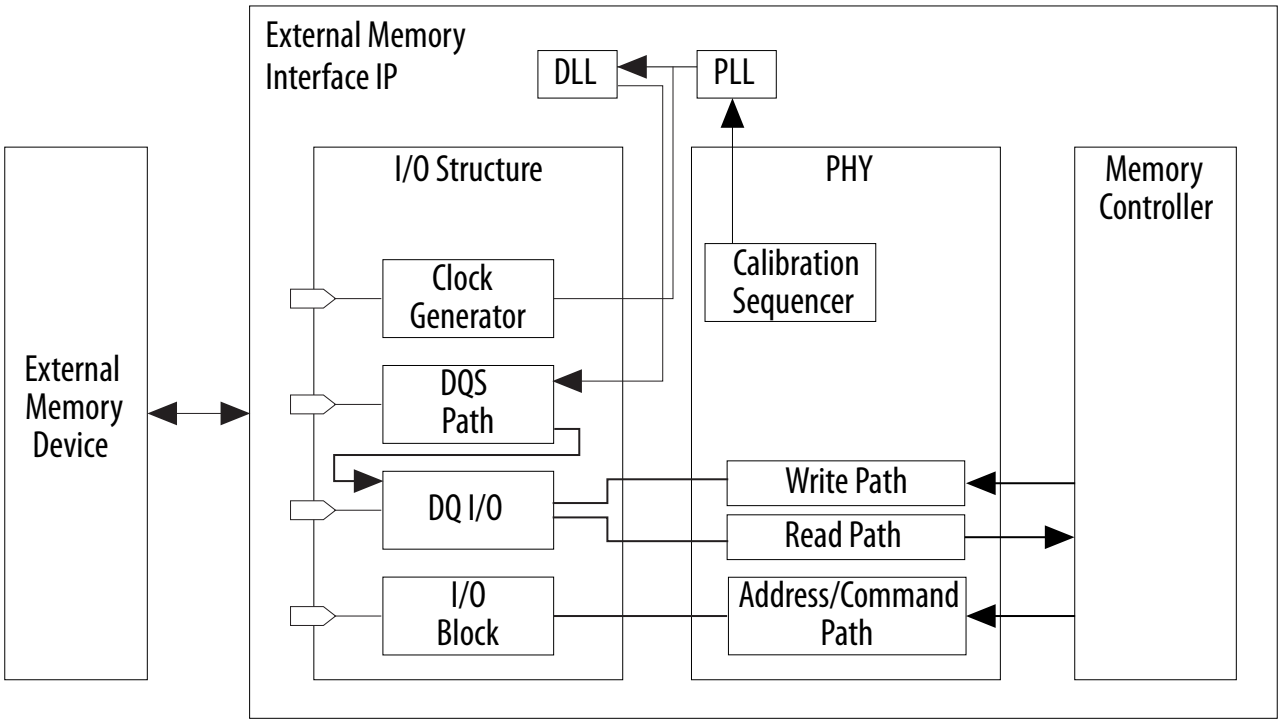
- **External Memory Interface Spec Estimator**
- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.
- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.

Memory Solutions

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

- Physical layer interface (PHY) which builds the data path and manages timing transfers between the FPGA and the memory device.
- Memory controller which implements all the memory commands and protocol-level requirements.
- Multi-port front end (MPFE) which allows multiple components inside the FPGA device to share a common memory interface. The MPFE is available in Arria V and Cyclone V devices.

Figure 1-1: Memory Interface Architecture



Altera's FPGAs provide two types of memory solutions, depending on device family: soft memory IP and hard 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.

The following table lists 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
• Includes hardened PHY with soft controller. • Allows maximum flexibility in choosing location, size, and configuration of the memory interface. • Can optionally be used in PHY-only mode to integrate with a custom user-designed controller.	• Includes hardened PHY, hardened controller, and hardened MPFE. • Supports maximum performance and lowest latency. • May have a fixed location on a device and/or a fixed pinout for address and command signals. • Simplifies the overall integration of a memory interface and provides an out-of-the-box experience for every designer.

Altera provides modular memory solutions that allow you to customize your memory interface design to a variety of configurations:

- PHY with your own controller
- PHY with Altera controller
- PHY with Altera controller and a multiport front end. (MPFE is a configurable block available for hard interfaces in Arria V and Cyclone V devices.)

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

Related Information

- **Volume 3: Reference Material**
- **ALTDLL and ALTDQ_DQS Megafunctions User Guide**
- **ALTDQ_DQS2 Megafunction User Guide**
- **Altera PHYLite for Memory Megafunction User Guide**
- **Functional Description: Arria 10 EMIF IP**
- **Functional Description: MAX 10 EMIF IP**
- **External Memory Interface Spec Estimator**
- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.
- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.

Protocol Support Matrix

The following table lists the device family and IP architecture support for each memory protocol in the current release of the Altera Complete Design Suite.

Figure 1-2: Protocol Support Matrix (1) (2) (3) (4) (5)

Protocol	MAX 10	Arria 10	Stratix V / Arria V GZ	Arria V GX, GT, SX, ST	Cyclone V	Stratix IV	Stratix III	Arria II GZ	Arria II GX	Clock Rate	Hard / Soft PHY	Burst length	Sequencer	Controller
DDR4	–	•	–	–	–	–	–	–	–	Quarter	Hard	8	Hard Nios	Hard
DDR3	–	•	–	–	–	–	–	–	–	Quarter	Hard	8	Hard Nios	Hard
	–	•	–	–	–	–	–	–	–	Half	Hard	8	Hard Nios	Hard
	–	–	–	U	U	–	–	–	–	Full	Hard	8	Nios II	HPC II
	U	–	U	U	U	U	U	U	A	Half	Soft	8	Nios II	HPC II
	–	–	U	U	–	–	–	–	–	Quarter	Soft	8	Nios II	HPC II
DDR2	–	–	–	U	U	–	–	–	–	Full	Hard	4,8	Nios II	HPC II
	–	–	U	–	–	U	U	U	A	Full	Soft	4,8	Nios II	HPC II
	U	–	U	U	U	U	U	U	A	Half	Soft	4,8	Nios II	HPC II
LPDDR3	–	•	–	–	–	–	–	–	–	Quarter Half	Hard	8	Hard Nios	Hard
LPDDR2	U	–	–	U	U	–	–	–	–	Half	Soft	4,8,16	Nios II	HPC II
	–	–	–	–	U	–	–	–	–	Full	Hard	4,8,16	Nios II	HPC II
RLDRAM 3	–	–	U	–	–	–	–	–	–	Half	Soft	2,4,8	Nios II	–
	–	–	U	–	–	–	–	–	–	Quarter	Soft	2,4,8	Nios II	–
	–	•	–	–	–	–	–	–	–	Quarter	Hard	2,4,8	Hard Nios	3RD Party
RLDRAM II	–	–	U	–	–	U	U	U	–	Full	Soft	2,4,8	RTL	RLDRAM II
	–	–	U	U	–	U	U	U	–	Half	Soft	4,8	Nios II	RLDRAM II
	–	–	U	–	–	U	U	U	–	Half	Soft	4,8	RTL	RLDRAM II
QDR II/II+	–	–	U	–	–	U	U	U	U	Full	Soft	2,4	RTL	QDR II/II+
	–	–	U	U	–	U	U	U	–	Half	Soft	4	Nios II	QDR II/II+
	–	–	U	–	–	U	U	U	U	Half	Soft	4	RTL	QDR II/II+
	–	•	–	–	–	–	–	–	–	Full	Hard	2,4	Hard Nios	QDR II/II+
	–	•	–	–	–	–	–	–	–	Half	Hard	4	Hard Nios	QDR II/II+
QDR II+ Xtreme	–	•	–	–	–	–	–	–	–	Half	Hard	4	Hard Nios	QDR II/II+
QDR IV	–	•	–	–	–	–	–	–	–	Quarter	Hard	2,4,8	Hard Nios	QDR IV

Notes to Table:

1. **U**= Supported by UniPHY-based IP.
2. **A** = Supported by ALTMEMPHY-based IP. Refer to the *External Memory Interface Handbook* for the Quartus II software version 12.1 or earlier for information about ALTMEMPHY-based IP.
3. **—** = Not supported.
4. **.** = Supported in Arria 10 device.
5. The RTL-based sequencer is not available for QDR II or RLDRAM II interfaces targeting Arria V devices.

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

For more information on the Arria 10 External Memory Interface IP, see *Functional Description—Arria 10 EMIF IP*.

For more information on the MAX 10 External Memory Interface IP, see *Functional Description—MAX 10 EMIF IP*.

For more information on the Arria 10 PHYLite IP, see the *PHYLite IP Megafunction User Guide*.

Related Information

- **External Memory Interface Spec Estimator**
- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.
- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.

Arria 10 EMIF Future Protocol Support

The following table lists planned future memory protocol support for Arria 10 EMIF IP.

Protocol	Current Support	Future Support
DDR4	• Hard PHY and Hard Controller • Hard PHY only	Yes (LRDIMM, RDIMM, x4 DQ/DQS)
DDR3	• Hard PHY and Hard Controller • Hard PHY only	Yes (LRDIMM, RDIMM, x4 DQ/DQS)
DDR2	No	Yes (via Altera PHYLite for Memory)
LPDDR3	Yes	Yes
LPDDR2	No	Yes (via Altera PHYLite for Memory)
QDR II / II+ / QDR II+ Xtreme	Hard PHY and Soft Controller	Yes
RLDRAM 3	Hard PHY only	Yes
RLDRAM II	No	Yes (via Altera PHYLite for Memory)

Related Information

- **External Memory Interface Spec Estimator**
- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.

- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.

Document Revision History

Date	Version	Changes
November 2015	2015.11.02	• Changed instances of *Quartus II* to *Quartus Prime*. • Changed Arria 10 EMIF current support for LPDDR3 to yes. • Added LPDDR3 to product support matrix.
May 2015	2015.05.04	Maintenance release.
December 2014	2014.12.15	• Added QDR IV and MAX 10 support to the *Protocol Support Matrix*.
August 2014	2014.08.15	• Added information for Quartus II software versions 14.0 and 14.0 Arria 10 Edition to *Altera Memory Types, PHY, and Controllers in the Quartus II Software* table. • Added QDR II, QDR II+, and QDR II+ Xtreme support for Arria 10 to the *Protocol Support Matrix*. • Updated DDR3, DDR4, QDR II+ / QDR II+ Extreme, and RLDRAM 3 support in the *Arria 10 EMIF Future Protocol Support* table.
December 2013	2013.12.16	• Added Arria 10 and DDR4 information to *Protocol Support Matrix* and *Memory Solutions*. • Combined *Soft and Hard Memory IP* and *Memory Solutions* sections. • Removed HardCopy III/IV from *Protocol Support Matrix*. • Added note to *Protocol Support Matrix* that RTL-based sequencer is not available for QDR II or RLDRAM II interfaces targeting Arria V devices
November 2012	2.0	• Added Arria V GZ information. • Added RLDRAM III information to *Protocol Support Matrix* and *Memory Solutions*.
June 2012	1.2	Change to Table 1–3.
June 2012	1.1	• Added *Protocol Support Matrix*. • Added Feedback icon.
November 2011	1.0	Initial release.

Related Information

- **External Memory Interface Spec Estimator**

- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.
- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.

Recommended Design Flow

2

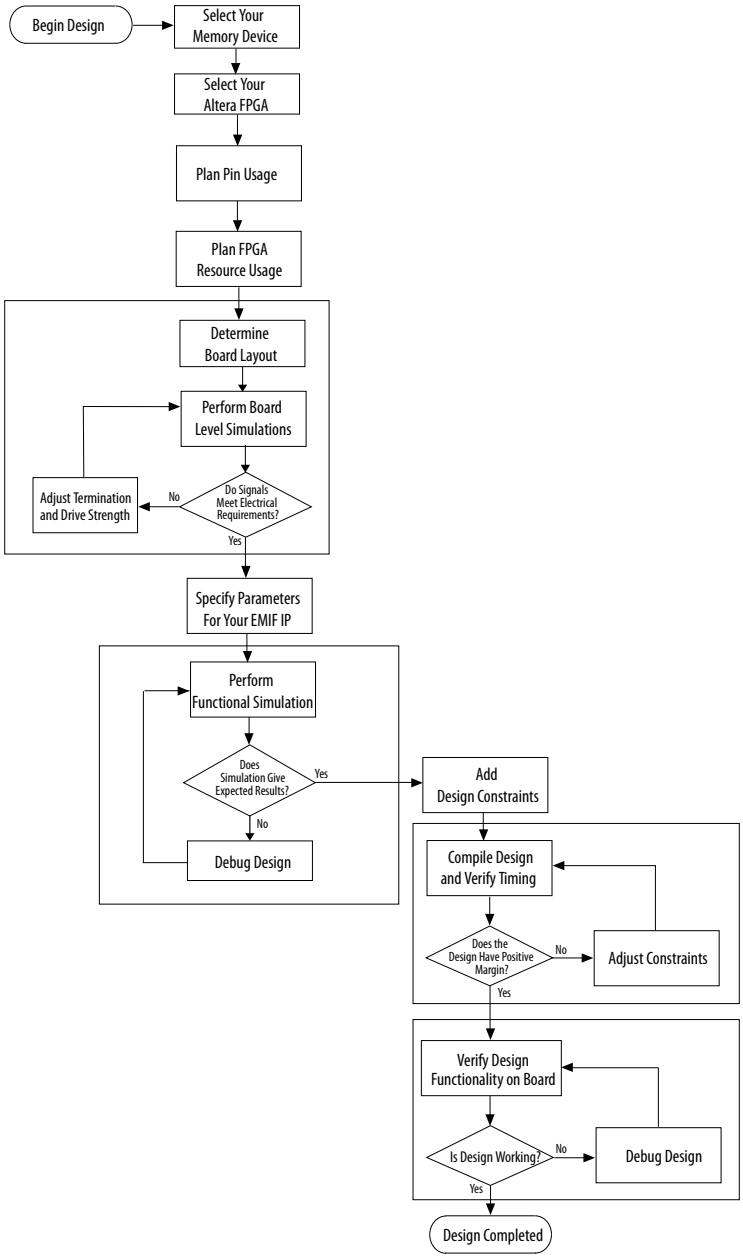
2015.11.02

EMI_GS

Altera recommends that you create an example top-level file with the desired pin outs and all interface IP instantiated, which enables the Quartus® Prime software to validate your design and resource allocation before PCB and schematic sign off.

The following figure shows the design flow to provide the fastest out-of-the-box experience with external memory interfaces in Altera devices. This design flow assumes that you are using Altera IP to implement the external memory interface.

Figure 2-1: External Memory Interfaces Design Flowchart



Refer to *Getting Started with External Memory Interfaces* for guidance in performing the recommended steps in creating a working and robust external memory interface.

Related Information

- **Getting Started With External Memory Interfaces** on page 2-3
- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.

- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.

Getting Started With External Memory Interfaces

To create your external memory interface, you must complete several high-level tasks. This topic outlines the major tasks in the design flow, and provides links to detailed procedures for each task.

Refer to this section for a big-picture view of the overall design process, and for links to related information for each task.

The High-Level Tasks

1. **Selecting Your External Memory Device** on page 2-4
   Different memory types excel in different areas. As a first step in planning your external memory interface, you must determine the memory type that best meets the requirements of your system.
2. **Selecting Your FPGA** on page 2-4
   Different Altera FPGA devices support different memory types; not all Altera devices support all memory protocols and configurations. Before you start your design, you must select an Altera device, which supports the memory standard and configurations you plan to use.
3. **Planning Your Pin Requirements** on page 2-5
   Before you can specify parameters for your external memory interface, you must determine the pin requirements.
4. **Planning Your FPGA Resources** on page 2-6
   Before you can specify parameters for your external memory interface, you must determine the FPGA resource requirements.
5. **Determining Your Board Layout** on page 2-6
   Before you can specify parameters for your external memory interface, you must determine the necessary board-related settings for your IP.
6. **Specifying Parameters for Your External Memory Interface** on page 2-6
   After you have determined all the necessary requirements, you can parameterize your external memory interface.
7. **Performing Functional Simulation** on page 2-7
   Simulate your design to determine correct operation, timing closure, and overall latency.
8. **Adding Design Constraints** on page 2-7
   Design constraints establish the timing characteristics of your IP and the physical locations of I/O and routing resources.
9. **Compiling Your Design and Verifying Timing** on page 2-8
   When you compile your design, the TimeQuest Timing Analyzer generates timing reports for your design.
10. **Verifying and Debugging External Memory Interface Operation** on page 2-8
    Operational problems can generally be attributed to one of the following: resource and planning problems, interface configuration problems, functional problems, signal integrity problems, or timing problems.

Selecting Your External Memory Device

Different memory types excel in different areas. As a first step in planning your external memory interface, you must determine the memory type that best meets the requirements of your system.

1. Determine your requirements for the following:
   - bandwidth
   - speed
   - data capacity
   - latency
   - power consumption
2. Compare your requirements to the specifications for available memory protocols to find the memory device appropriate for your application.

Related Information

Selecting Your Memory

Selecting Your FPGA

Different Altera FPGA devices support different memory types; not all Altera devices support all memory protocols and configurations. Before you start your design, you must select an Altera device, which supports the memory standard and configurations you plan to use.

1. Determine the I/O interface that best suits your design requirements.
2. Determine whether your design requires read or write levelling circuitry.

   Some Altera FPGAs support read and write levelling, to apply or remove skew from an interface on a DQS group basis.
3. Determine whether your design requires dynamic calibrated on-chip termination (OCT).

   Some Altera FPGAs provide dynamic OCT, allowing a specified series termination to be enabled during writes and parallel termination to be enabled during reads. Dynamic OCT can simplify your PCB termination schemes.
4. Consult the Altera Product Selector to find the Altera FPGA that provides the combination of features that your design requires.
5. Refer to the Ordering Information section of the appropriate device handbook, to determine the correct ordering code for the device that you require. Consider the following characteristics in determining the correct ordering code:

- Speed grade: Affects performance, timing closure, and power consumption. The device with the smallest speed grade number is the fastest device.
- Operating temperature: Altera FPGAs are divided into the following temperature categories:
  - Commercial grade—Used for all device families. Operating temperature ranges from 0 degreec C to 85 degrees C.
  - Industrial grade—Used for all device families. Operating temperature ranges from -40 degreec C to 100 degrees C.
  - Military grade—Used for Stratix IV device families. Operating temperature ranges from -55 degreec C to 125 degrees C.
  - Automotive grade—Used for Cyclone V device families. Operating temperature ranges from -40 degreec C to 125 degrees C.
- Package size: Refers to the physical size of the FPGA device, and corresponds to the number of pins. For example, the package size for the smallest Stratix IV device is 29 mm x 29 mm, categorized under the F780 package option, where F780 refers to a device with 780 pins.
- Device density: Refers to the number of logic elements, such as PLLs and memory blocks. Devices with higher density contain more logic elements in less area.
- I/O pin counts: The number of I/O pins required on an FPGA depends on the memory standard, the number of memory interfaces, and the memory data width.

Tip: For additional, device-specific, information, refer to the External Memory Interface chapter in the device handbook for your Altera device.

Related Information

- **Selecting Your FPGA Device**
- **Altera Product Selector**
- **Altera Device Handbooks**

Planning Your Pin Requirements

Before you can specify parameters for your external memory interface, you must determine the pin requirements. You should use the Quartus Prime software for final pin fitting; however, you can estimate whether you have enough pins for your memory interface.

1. Determine how many read data pins are associated per read data strobe or clock pair.
2. Check the device density and packaging information for your FPGA to determine whether you can implement your interface in one I/O bank, or on one side of the device, or on two adjacent sides.
3. Calculate the number of other memory interface pins needed, including any other clocks (write clock or memory system clock), address, command, RUP, RDN, RZQ, and any other pins to be connected to the memory components. Ensure you have enough pins to implement the interface in one I/O bank or one side or on two adjacent sides.
4. Apply the General Pin-Out Guidelines, and observe any device- or protocol-specific guidelines or exceptions applicable to your design situation.

Related Information

- **Planning Pin and FPGA Resources**
- **External Memory Interface Spec Estimator**

Planning Your FPGA Resources

Before you can specify parameters for your external memory interface, you must determine the FPGA resource requirements. The FPGA resources required by your design depend on many factors, including the memory interface frequency, timing requirements, and the IP that your design uses.

1. Determine the PLLs and clock networks that your design requires.
2. If multiple PLLs are required for multiple controllers that cannot be shared, ensure that enough PLL resources are available within each quadrant to support your interface number requirements.
3. Determine whether cascading of PLLs is appropriate for your design.
4. Determine the appropriate DLL usage for your design. If multiple external memory interfaces must share DLL resources, ensure that the frequency and mode requirements are compatible.
5. Determine the registers, memory blocks, OCT blocks, and other FPGA resources required by your design.

Related Information

- **Planning Pin and FPGA Resources**
- **External Memory Interface Spec Estimator**

Determining Your Board Layout

Before you can specify parameters for your external memory interface, you must determine the necessary board-related settings for your IP.

1. Review the recommended board design guidelines for your external memory interface protocol.
2. Select the termination scheme and drive strength settings for all the memory interface signals connected between the FPGA and the external memory device.
3. Perform board-level simulations to determine the optimal settings for best signal integrity, appropriate timing margins, and sufficient eye opening.
   - Successful board-level simulation is often an iterative process, experimenting with different combinations of drive strength, terminations, IP board parameters, and timing results.
   - Ensure that your simulation applies the latest FPGA and memory device IBIS models, board trace characteristics, drive strength, and termination settings.
   - You might identify board-level timing uncertainties such as crosstalk, ISI, or slew rate deration during simulation. If you identify such timing uncertainties, adjust the Board Settings in the IP Catalog with the slew rate deration, ISI/crosstalk, and board skews to ensure the accuracy of the TimeQuest timing margins report.

Related Information

- **DDR2, DDR3, and DDR4 SDRAM Board Design Guidelines**
- **Dual-DIMM DDR2 and DDR3 SDRAM Board Design Guidelines**
- **LPDDR2 SDRAM Board Design Guidelines**
- **QDR II and QDR IV SRAM Board Design Guidelines**
- **RLDRAM II and RLDRAM 3 Board Design Guidelines**

Specifying Parameters for Your External Memory Interface

After you have determined all the necessary requirements, you can parameterize your external memory interface.

1. In the parameter editor, set the parameters for the external memory IP for your target memory interface.
   - Refer to *Specifying IP Core Parameters and Options* for information about using the IP Catalog and parameter editor.
   - Refer to *Implementing and Parameterizing Memory IP* for detailed information about parameterizing external memory interface IP.
2. Specify the correct parameters for each of the following:
   - Memory interface data rate, width, and configuration.
   - Necessary deratings for tIS, tIH, tDH, and tDS parameters, as appropriate.
   - Board skew parameters based on actual board simulation.
3. Connect the local signals from the PHY and controller to your driver logic, and the memory interface signals from the PHY to the top-level pins.
   - It is important that you connect the local interface signals from the PHY or controller correctly to your own logic. If you do not connect these local interface signals, you might encounter problems with insufficient pins when you compile your design.
   - Logic that is not connected may be optimized away during compilation, resulting in problems later.
   - If you want to use your own custom memory controller with the Altera PHY, you can refer to the example top-level file as an example for connecting your controller.

Related Information

- **Implementing and Parameterizing Memory IP**
- **Functional Description—HPC II Controller**
- **Functional Description–Hard Memory Interface**
- **Functional Description—QDR II Controller**
- **Functional Description—QDR IV Controller**
- **Functional Description—RLDRAM II Controller**
- **Functional Description—RLDRAM 3 PHY-Only IP**
- **Functional Description—Arria 10 EMIF**
- **Functional Description—MAX 10 EMIF**

Performing Functional Simulation

Simulate your design to determine correct operation, timing closure, and overall latency.

1. Simulate your design using the RTL functional model.
2. 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.
3. You may need to prepare the memory functional model by setting the speed grade and device bus mode.

Related Information

Simulating Memory IP

Adding Design Constraints

Design constraints establish the timing characteristics of your IP and the physical locations of I/O and routing resources.

1. Add timing constraints.
2. Add pin assignments.
3. Add pin location assignments.
4. Ensure that the example top-level file or your top-level logic is set as top-level entity.
5. Adjust optimization techniques, to ensure the remaining unconstrained paths are routed with the highest speed and efficiency, as follows:
   a. In the Quartus Prime software, click **Assignments** > **Settings**.
   b. In the **Settings** dialog box, select the **Compiler Settings** category.
   c. In the **Compiler Settings** dialog box, click **Advanced Settings (Synthesis)** and set the **Optimization Technique** value to **Speed**.
   d. In the **Compiler Settings** dialog box, click **Advanced Settings (Fitter)** and set **Optimize hold timing** to **All Paths**. Turn on **Optimize multi-corner timing**. Set **Fitter Effort** to **Standard Fit**.

Related Information

Analyzing Timing of Memory IP

Compiling Your Design and Verifying Timing

When you compile your design, the TimeQuest Timing Analyzer generates timing reports for your design.

1. Compile your design by clicking **Processing** > **Start Compilation**.
   Memory timing scripts run automatically as part of `Report DDR`.
2. Verify timing closure using all available models, and evaluate the timing reports generated by the TimeQuest Timing Analyzer.
   As required, adjust the constraints described in **Adding Design Constraints** to resolve timing or location issues.
3. Iteratively recompile your IP and evaluate the timing results as necessary to achieve the required timing margins.

Related Information

- **Analyzing Timing of Memory IP**
- **Implementing and Parameterizing Memory IP**

Verifying and Debugging External Memory Interface Operation

Operational problems can generally be attributed to one of the following: resource and planning problems, interface configuration problems, functional problems, signal integrity problems, or timing problems.

- Refer to *Debugging Memory IP* and the *External Memory Interface Debug Toolkit* for information on resolving operational problems.

Related Information

- **Debugging Memory IP**
- **External Memory Interface Debug Toolkit**

Document Revision History

Date	Version	Changes
November 2015	2015.11.02	Changed instances of *Quartus II* to *Quartus Prime*.
May 2015	2015.05.04	Maintenance release.
December 2014	2014.12.15	• Revised the *External Memory Interfaces Design Flowchart*. • Removed the *Design Checklist* and added *Getting Started With External Memory Interfaces*, and associated subtopics.
August 2014	2014.08.15	Removed MegaWizard Plug-In Manager flow and added IP Catalog Flow to *External Memory Interfaces Design Flowchart*.
December 2013	2013.12.16	• Removed references to ALTMEMPHY. • Removed references to SOPC Builder. • Removed ALTMEMPHY-related step from design checklist.
June 2012	2013.12.02	• Removed overlapping information. • Added Feedback icon.
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	• Improved description for Implementing Altera Memory Interface IP chapter. • Added timing simulation to flow chart and to design checklist.
November 2009	1.0	Initial release.