Design Guidelines for Implementing External Memory Interfaces in Stratix II and Stratix II GX Devices

July 2007, v1.1 Application Note 449

Introduction

Stratix® II offers support for double data rate (DDR) memories, such as DDR2/DDR SDRAM, QDRII+/QDRII SRAM, and RLDRAM II interfaces. Applications with these memory interfaces include PCs, embedded processing, image processing, storage, communications, and networking.

Designing applications with memory interfaces can be challenging considering the different requirements that the memory standards impose on the controller. Moreover, changing a portion of the interface can affect many other things in the application such that designers have to pay close attention to each step of the design process. Knowing the proper design flow is crucial in reducing design and debug time to get the application up and running.

Table 1 summarizes the maximum clock rate Stratix II and Stratix II GX devices can support with external memory devices. Depending on the target clock rate performance, you can use either the DLL- or PLL-based implementation for your memory interface. While both solutions use pre-determined DQS and DQ pins listed in the pin tables, the DLL-based solution also uses dedicated circuitry to shift the DQS read signal inside the IOE to capture the DQ read data. The PLL-based solution allows the Stratix II side I/O banks to interface with memory devices. However, this solution ignores the DQS read signal and uses a PLL output to capture the DQ read data. Therefore, the PLL-based solution has a lower maximum supported clock rate, because the advantage of the tight skew between DQS and DQ signals from the memory devices is lost when the DQS is not being used to capture data.

Table 1. Stratix II and Stratix II GX Maximum Clock Rate Support for External Memory Interfaces (Part 1 of 2) *Note (1), (2)*

Memory Standards	-3 Speed Grade (MHz)		-4 Speed Grade (MHz)		-5 Speed Grade (MHz)	
	DLL-Based	PLL-Based	DLL-Based	PLL-Based	DLL-Based	PLL-Based
DDR2 SDRAM *(3)*	333	200	267	167	233	167
DDR SDRAM *(3)*	200	150	200	133	200	100
RLDRAM II	300	200	250 *(4)*	175	200	175
QDRII+ SRAM	300	–	–	–	–	–

Table 1. Stratix II and Stratix II GX Maximum Clock Rate Support for External Memory Interfaces (Part 2 of 2) *Note (1), (2)*

Memory Standards	-3 Speed Grade (MHz)		-4 Speed Grade (MHz)		-5 Speed Grade (MHz)	
	DLL-Based	PLL-Based	DLL-Based	PLL-Based	DLL-Based	PLL-Based
QDRII SRAM	300	200	250	167	250	167

Notes to Table 1:

(1) Memory interface timing specifications are dependent on the memory, board, physical interface, and core logic. Refer to each memory interface application note for more details on how each specification was generated.

(2) The respective Altera® MegaCore® function and the EP2S60F1020C3 timing information featured in the Quartus® II software version 6.1 was used to define these clock rates.

(3) This applies for interfaces with both modules and components.

(4) You must underclock a 300-MHz RLDRAM II device to achieve this clock rate.

Refer to:

- AN 325: Interfacing RLDRAM II with Stratix II, Stratix, and Stratix GX Devices
- AN 326: Interfacing QDRII SRAM with Stratix II, Stratix, and Stratix GX Devices
- AN 327: Interfacing DDR SDRAM with Stratix II Devices
- AN 328: Interfacing DDR2 SDRAM with Stratix II Devices

for more information about the respective memory interface implementation.

This application note describes the typical memory interface design flow for Stratix II devices and provides literature links offered by Altera that are pertinent to the specific point you are at in the design cycle.

For the HardCopy II memory interface design flow, refer to *AN 413: Using Legacy Integrated Static Data Path and Controller Megafunction with HardCopy II Structured ASICs*.

Stratix II Memory Interface Design Flow

Altera recommends the design guidelines described in this section as best practices for successful memory interface implementation in Stratix II devices. These guidelines provide the fastest out-of-the-box experience with external memory interfaces in Stratix II devices. A detailed discussion of each step presented in Figure 1 appears in the following sections.

Figure 1. Design Flow for Implementing External Memory Interfaces in Stratix III Devices

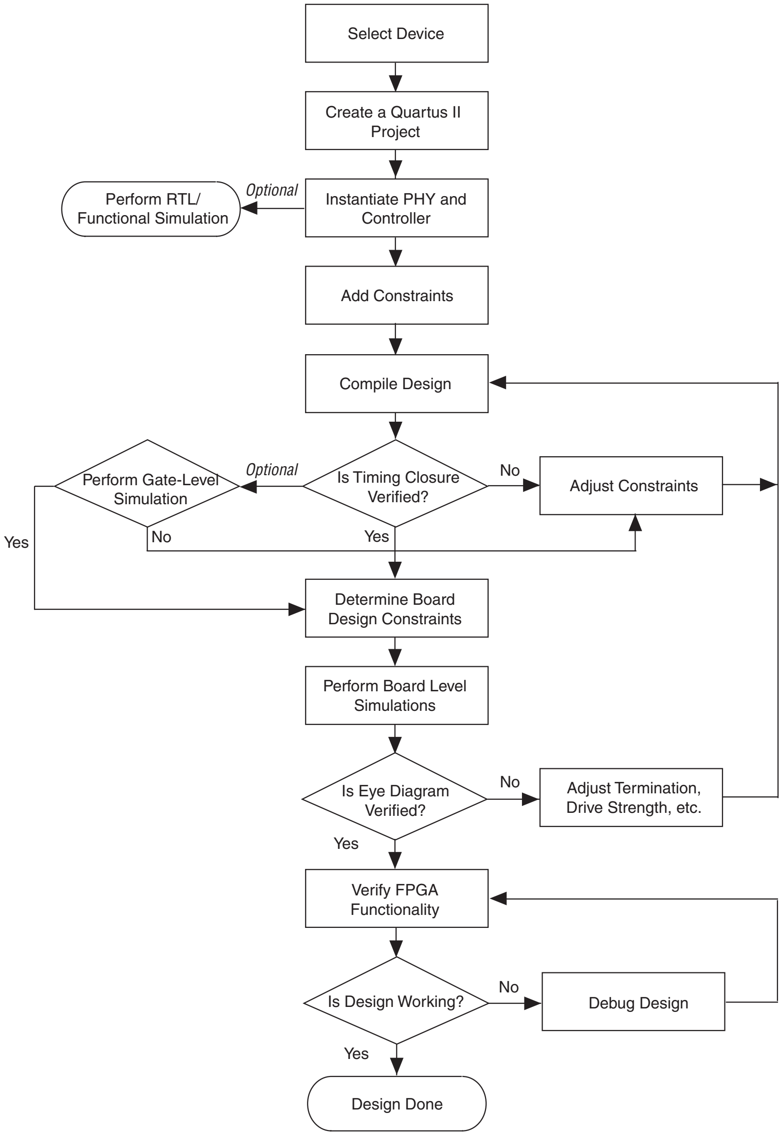
Select Device
Create a Quartus II Project
Instantiate PHY and Controller
Optional
Perform RTL/ Functional Simulation
Add Constraints
Compile Design
Is Timing Closure Verified?
No
Adjust Constraints
Optional
Perform Gate-Level Simulation
Yes
No
Yes
Determine Board Design Constraints
Perform Board Level Simulations
Is Eye Diagram Verified?
No
Adjust Termination, Drive Strength, etc.
Yes
Verify FPGA Functionality
Is Design Working?
No
Debug Design
Yes
Design Done

Table 2 shows a summary of collateral reference for each design step in Figure 1.

Table 2. Corresponding Collateral with the Memory Interface Design Flow (Part 1 of 3) Note (1)

Design Steps	Variation of Steps	Reference
Select device	None	● *Selecting the Right High-Speed Memory Technology for Your System* White Paper ● *External Memory Interfaces* chapter of the *Stratix II* or *Stratix II GX Device Handbook*
Create a Quartus II project	None	● *Tutorial* in the Quartus II Software
Instantiate PHY and controller	Controllers with ALTMEMPHY megafunction	● *DDR and DDR2 SDRAM High Performance Controller User Guide* ● *ALTMEMPHY Megafunction User Guide*
	Legacy memory controllers	● *DDR and DDR2 SDRAM Controller Compiler User Guide* ● *QDRII SRAM Controller User Guide* ● *RLDRAM II Controller User Guide* ● *AN 328: Interfacing DDR2 SDRAM with Stratix II Devices* ● *AN 327: Interfacing DDR SDRAM with Stratix II Devices* ● *AN 326: Interfacing QDRII SRAM with Stratix II, Stratix, and Stratix GX Devices* ● *AN 325: Interfacing RLDRAM II with Stratix II, Stratix, and Stratix GX Devices* ● *AN 392: Multiple DDR and DDR2 SDRAM* ● *AN 398: Using DDR/DDR2 SDRAM with SOPC Builder*
Perform RTL/functional simulation	None	● *AN 380: Test DDR or DDR2 SDRAM Interfaces on Hardware Using the Example Driver*

Table 2. Corresponding Collateral with the Memory Interface Design Flow (Part 2 of 3) Note (1)		
Design Steps	**Variation of Steps**	**Reference**
Add constraints	Controllers with ALTMEMPHY megafunction	● *DDR and DDR2 SDRAM High Performance Controller User Guide* ● *ALTMEMPHY Megafunction User Guide*
	Legacy memory controllers	● *DDR and DDR2 SDRAM Controller Compiler User Guide* ● *QDRII SRAM Controller User Guide* ● *RLDRAM II Controller User Guide* ● *DDR Timing Wizard (DTW) User Guide* ● *AN 408: DDR2 Memory Interface Termination, Drive Strength and Loading* ● *Loading Design Guidelines*
Compile design	None	● *Tutorial in the Quartus II Software*
Verify timing closure	Controllers with ALTMEMPHY megafunction	● *DDR and DDR2 SDRAM High Performance Controller User Guide* ● *ALTMEMPHY Megafunction User Guide*
	Legacy controllers	● *DDR Timing Wizard (DTW) User Guide*
Perform gate-level simulation		● *Verification* section of the *Quartus II Software Handbook, Volume 3* ● *DDR and DDR2 SDRAM Controller Compiler User Guide* ● *QDRII SRAM Controller User Guide* ● *RLDRAM II Controller User Guide*
Determine board design constraints		● *AN 408: DDR2 Memory Interface Termination, Drive Strength and Loading* ● *AN 444: Dual DIMM DDR2 SDRAM Memory Interface Design Guidelines* ● *AN 315: Guidelines for Designing High-Speed FPGA PCBs* ● *AN 224: High-Speed Board Layout Guidelines* ● *AN 75: High-Speed Board Designs*
Perform board level simulations		● *AN 408: DDR2 Memory Interface Termination, Drive Strength & Loading*

Table 2. Corresponding Collateral with the Memory Interface Design Flow (Part 3 of 3) Note (1)

Design Steps	Variation of Steps	Reference
Verify FPGA functionality		● *Chapter 13: Design Debugging Using the SignalTap II Embedded Logic Analyzer* of the *Quartus II Software Handbook, Volume 3* ● *AN 380: Test DDR or DDR2 SDRAM Interfaces on Hardware Using the Example Driver*

Note to Table 2:

(1) Go to Altera’s External Memory Solutions Center at **http://www.altera.com/technology/memory/mem-index.jsp** for the most up-to-date collateral for Stratix II devices.

Step 1: Select Device

Prior to the start of designing any memory interface, determine the required bandwidth of the memory interface. Bandwidth can be expressed as:

Bandwidth = Data width (bits) × data rate transfer (1/sec) × efficiency

The efficiency is the percentage of time the data bus is transferring data. It is dependent on the type of memory. For example, in a memory interface with separate read and write ports, the efficiency is 100% when there is an equal amount of reads and writes on these memory interfaces.

After calculating the bandwidth of the memory interface, determine which memory and FPGA to use. Altera has a memory selection white paper, which highlights the differences between the memory types. Refer to the *Selecting the Right High-Speed Memory Technology for Your System* White Paper for information about selecting the different memory types.

Altera's FPGA device families support various data widths and maximum performance for different memory interfaces. The memory interface support between density and package combinations differs, so you must determine which FPGA device density and package combination best suits your application.

Refer to the *External Memory Interfaces* chapter of the*Stratix II* or *Stratix II GX Device Handbook* for information about the Stratix II device density and package support for the different memory types.

Step 2: Create a Quartus II Project

After selecting the appropriate FPGA device and memory type, you can create a project in the Quartus II software targeting the FPGA device and memory type.

Refer to the *Tutorial* in the Quartus II software for step-by-step instructions on creating a Quartus II project.

Step 3: Instantiate PHY and Controller

When instantiating the data path for your memory interface, Altera recommends that you use the data path from the Altera memory controller MegaCore functions and megafunctions listed in Table 3 with their corresponding user guides.

Table 3. Altera Memory Controller MegaCore Functions and Megafunctions

MegaCore Functions	User Guide	Link to the User Guide
DDR and DDR2 SDRAM High Performance Controller	*DDR and DDR2 SDRAM High-Performance Controller User Guide*	http://www.altera.com/literature/ug/ug_ddr_ddr2_sdram_hp.pdf
ALTMEMPHY Megafunction *(1)*	*ALTMEMPHY Megafunction User Guide*	http://www.altera.com/literature/ug/ug_altmemphy.pdf
DDR and DDR2 SDRAM Controller Compiler	*DDR and DDR2 SDRAM Controller Compiler User Guide*	http://www.altera.com/literature/ug/ug_ddr_sdram.pdf
QDRII SRAM Controller	*QDRII SRAM Controller MegaCore Function User Guide*	http://www.altera.com/literature/ug/ug_qdrii_sram.pdf
RLDRAM II Controller	*RLDRAM II Controller MegaCore Function User Guide*	http://www.altera.com/literature/ug/ug_rldram_ii.pdf

Note to Table 3:

(1) The ALTMEMPHY megafunction is the data path used in the DDR and DDR2 SDRAM High Performance Controller. This is unlike the other controllers where the data path is not offered as a separate entity as a megafunction.

With the exception of the DDR and DDR2 SDRAM High Performance Controller, the legacy MegaCore functions listed in Table 3 include an integrated clear-text data path and encrypted controller in each MegaCore function. The DDR and DDR2 SDRAM High Performance Controller uses the ALTMEMPHY megafunction to set the data path of the memory controller. The legacy MegaCore functions also support both DLL-based and PLL-based implementations, while the DDR and DDR2 SDRAM High Performance Controller only supports DLL-based implementations.

Altera provides two different memory controllers for DDR and DDR2 SDRAM interfaces, depending on the clock rate of operation: the DDR and DDR2 SDRAM High Performance Controller with the ALTMEMPHY data path and the legacy DDR and DDR2 SDRAM Controller Compiler. Table 4 lists the differences between the legacy controller compiler and the new controller.

Table 4. Comparing the Legacy Controller Compiler and the New Controller

Legacy DDR and DDR2 SDRAM Controller Compiler	DDR and DDR2 SDRAM High Performance Controller with the ALTMEMPHY Data Path
Up to 267 MHz in Stratix II Stratix III support for migration design only	Up to 333 MHz in Stratix II Up to 400 MHz in Stratix III
Integrated controller and data path	Separate datapath (available in the ALTMEMPHY megafunction) and controller
Static phase shift selection for resynchronization	Auto-calibrated resynchronization clock
Require board trace length measurements	Does not require board trace length information
1-PLL or 2-PLL mode	1-PLL solution
DLL-based or PLL-based mode	DLL-based mode only
Full-rate controller	Half-rate controller *(1)*
TAN and TimeQuest support	TimeQuest support only
DDR2/DDR SDRAM, RLDRAM II, QDRII+/QDRII SRAM	DDR3/DDR2/DDR SDRAM for Quartus II version 6.1 RLDRAM II and QDRII+/QDR SRAM in future versions of Quartus II

Note to Table 4:

(1) Full-rate controller support will be added in a future version of Quartus II software.

☞ Since the high-performance controller uses a FIFO for resynchronization, the legacy controller incurs less latency than the high-performance controller. Furthermore, the half-rate high-performance controller uses more device resources than the legacy controller.

For all new designs, Altera recommends the use of the ALTMEMPHY data path, available as a megafunction or in the DDR and DDR2 SDRAM High Performance Controller, especially if you are planning to migrate to a Stratix III device or if you are running at a frequency higher than 200 MHz. You can use the legacy controller with the 2-PLL mode when running at or above 200 MHz (up to 267 MHz) only if you cannot use a half-rate controller or only if you must use the Classic Timing Analyzer, instead of the TimeQuest Timing Analyzer.

There are two ways to instantiate the ALTMEMPHY megafunction: using the ALTMEMPHY MegaWizard® Plug-In Manager or using Altera's DDR and DDR2 SDRAM High Performance Controller, which already includes the ALTMEMPHY megafunction. Even if you plan to use your own controller, Altera recommends that you first create a design using Altera's DDR and DDR2 SDRAM High Performance Controller and then replace the Altera controller with your own controller. In this way, you get an example design which you can simulate and verify.

You can use either the MegaWizard Plug-In Manager or the SOPC Builder to invoke the IP Tool Bench, where you can instantiate and customize your memory controller. Use SOPC Builder if you are going to use the external memory as a peripheral of a NIOS soft processor.

Refer to *AN 398: Using DDR/DDR2 SDRAM with SOPC Builder* for more information about instantiating a legacy memory controller with the SOPC Builder.

CAUTION The DDR and DDR2 SDRAM High Performance Controller in Quartus II version 6.1 does not support SOPC Builder, but this support will be added in a future version of the Quartus II software.

As stated earlier, Altera recommends the use of the ALTMEMPHY megafunction or the DDR and DDR2 SDRAM High Performance Controller. You can use the legacy memory controllers if one of the following exceptions applies:

- You are not planning to migrate to a Stratix III device
- You are not interfacing with DDR/DDR2 SDRAM devices above 200 MHz
- You are unable to use TimeQuest Timing Analyzer
- You are unable to use half-rate controllers or logic in the design

If you plan to have multiple legacy memory controllers in your design, refer to *AN 392: Multiple DDR and DDR2 SDRAM Controllers on One Device*.

The following sections cover design flow steps for both the legacy controllers and controllers with the ALTMEMPHY data path.

☞ Stratix II only supports RLDRAM II, QDRII+/QDRII SRAM with the legacy controllers.

Step 4: Perform RTL/Functional Simulation

Simulating the design to verify functionality is crucial in ensuring that the application performs as designed in your system.

Controllers with ALTMEMPHY Data Path

When using the ALTMEMPHY MegaWizard Plug-In Manager, there is an option to generate a simulation model of the design in either Verilog HDL or VHDL. This IP functional simulation model is also a cycle-accurate HDL model file produced by the Quartus II software. When instantiating the memory interface using Altera's DDR and DDR2 SDRAM High Performance Controller, it generates an example design and a testbench in addition to the ALTMEMPHY megafunction simulation model.

Legacy Controllers

Similarly, when you instantiate the memory controller using the Altera legacy memory controller MegaCore functions, you can select a cycle-accurate Verilog HDL or VHDL functional simulation model to generate a test bench for the example design.

The test bench file for both implementations is saved in the *<project_directory>*/**testbench** folder.

The models work with Altera-supported VHDL and Verilog HDL simulators and include a generic memory device model. Download the actual memory device model that you are using from the vendor website.

For more information simulating the controller, refer to *Appendix C. Perform Functional Simulation* section of *AN 380: Test DDR or DDR2 SDRAM Interfaces on Hardware Using the Example Driver* or the respective memory controller user guides.

Step 5: Add Constraints

The next step in the design process is to add all timing, location, and physical constraints related to the external memory interface. This includes timing, pin locations, I/O standards, and pin loading assignments.

To determine which drive strength and termination to use, refer to *AN 408: DDR2 Memory Interface Termination, Drive Strength and Loading*.

Controllers with ALTMEMPHY Data Path

The top level of the DDR and DDR2 High Performance Controller design is fully timing-constrained using Synopsys design constraints (SDC) assignments. These timing assignments are a function of the parameters you enter in the ALTMEMPHY MegaWizard Plug-In or the High Performance Controller MegaWizard Plug-In, and are derived from the memory datasheet and tolerances from your board layout. The ALTMEMPHY megafunction uses accurate TimeQuest timing constraints and the timing driven fitter to achieve timing closure. After you instantiate the ALTMEMPHY megafunction, the MegaWizard Plug-In Manager generates the following script files that you must use in order to properly constrain the design.

- <*variation_name*>**_phy_ddr_timing.sdc**
- <*variation_name*>**_pin_assignments.tcl**

These script files are based on the design name used when instantiating the ALTMEMPHY megafunction or the DDR and DDR2 SDRAM High Performance Controller MegaCore function. If you plan to use your own top level design, you will need to edit the scripts to match your custom top level design.

For more information on ALTMEMPHY or the high performance controller, refer to the *ALTMEMPHY Megafunction User Guide* or the *DDR and DDR2 SDRAM High Performance Controller User Guide.*

Legacy Controllers

When using the legacy memory controller MegaCore functions, select the DQS and DQ pin location from the **IP Tool Bench Constraints** dialog box. When you click **Generate**, the IP Tool Bench creates the pin location, I/O standard, and loading assignments based on the memory device, FPGA device, and board information in a tcl file called **add_constraints_for_**<*variation_name*>**.tcl**. In addition, the IP Tool Bench also generates a tcl file called **auto_add_**<*ddr*|*qdrii*|*rldramii*>**_constraints.tcl** that contains all the constraints for multiple memory controllers of the same type in the project. For example, if the project has two DDR2 SDRAM memory controllers, constraints for both controllers can be sourced using the **auto_add_ddr_constraints.tcl** file.

To add timing constraints for the legacy controllers, run the **DDR Timing Wizard** (DTW).

For more information about how to use the DTW, refer to the *DDR Timing Wizard (DTW) User Guide.*

For more information on the **add_constraints_for_**<*variation_name*>**.tcl** script, refer to the respective legacy memory controller user guides.

Step 6: Compile Design and Verify Timing Closure

Once the constraints are made, compile the design to ensure that it meets timing requirements.

Controllers with ALTMEMPHY Data Path

During the generation of the ALTMEMPHY megafunction or the High Performance SDRAM Memory Controller, the MegaWizard Plug-In Manager generates a report timing script called <*variation_name*>**_report_timing.tcl**, which you can run after compiling the design to produce the timing report for different paths, such as write data, read data and command/address, and controller timing paths in the design.

For more information on ALTMEMPHY or the high performance controller, refer to the *ALTMEMPHY Megafunction User Guide* or the *DDR and DDR2 SDRAM High Performance Controller User Guide.*

Legacy Controllers

Memory interfaces design constrained using DTW use the **dtw_timing_analysis.tcl**, that is included with the DTW User Guide. This allows the Quartus II software to report the margin for different timing paths in the interface and to recommend the ideal phase shift settings for PLL-based read capture, resynchronization, and DLL-based postamble clocks. After compiling the design, run the **dtw_timing_analysis.tcl** script to verify timing closure. You can change the clock cycles and phase shifts of the design to the script-recommended results if timing is not met.

For more information on how to use the script and DTW, refer to the *DDR Timing Wizard (DTW) User Guide.*

When using DTW, the **verify_timing.tcl** script generated by IP Tool Bench for legacy controllers is ignored. The DTW and **dtw_timing_analysis.tcl** provide a more complete timing analysis and report compared with the IP Tool Bench **verify_timing.tcl** script.

Step 7: Perform Gate Level Simulation (Optional)

This optional step allows you to ensure that your system meets the proper timing requirements needed by each module of the design in a graphical waveform.

Refer to the *Verification* section of the *Quartus II Software Handbook, Volume 3* or the respective memory controller User Guides for more information on simulating your controller.

Step 8: 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 and you wish to achieve balanced setup and hold margin, adjust the phase setting of the clocks that are used to clock these paths. For example, if the report timing script (either from <*variation_name*>**_report_timing.tcl** for controllers with the ALTMEMPHY data path, or from **dtw_timing_analysis.tcl** for legacy controllers) indicates that the current address/command clock phase shift results in more hold time than setup time, you can add more phase shift to the address/command clock with respect to the system clock so that there will be less hold margin. Similarly, you subtract the address/command clock phase shift with respect to the system clock if there is more setup margin.

When using the legacy data path, the **dtw_timing_analysis.tcl** script recommends the ideal clock cycles and phase shifts for the resynchronization and postamble clocks, whenever applicable. The **report_timing.tcl** for controllers with the ALTMEMPHY data path only reports the margin of the interface.

Refer to the *DDR Timing Wizard (DTW) User Guide* for more information on the **dtw_timing_analysis.tcl** script.

Step 9: Determine Board Design Constraints

To determine the correct board constraints, run 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 electronic design automation (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

Changing the settings for the memory interface affects the timing. If you change them, it may be necessary to go back to the timing closure.

Once you close the timing for the design, evaluate the trade-offs posed by various board design choices. Different factors contribute to signal integrity and affect the overall timing margin for the memory and the FPGA. Some factors to consider that can affect the signal integrity include the termination scheme used, the drive strength setting on the FPGA, and the loading seen by the driver. Learn the trade-offs between the different types of termination schemes, the effects of output drive strengths, and loading, so that you can swiftly navigate through the multiple combinations and choose the best possible settings for your designs.

Refer to *AN 408: DDR2 Memory Interface Termination, Drive Strength and Loading Design Guidelines* for detailed information about understanding the different effects on signal integrity design.

For systems with two DIMMs connected in parallel, refer to *AN 444: Dual DIMM DDR2 SDRAM Memory Interface Design Guidelines*.

Altera offers the following application notes regarding high speed board design information:

- *AN 315: Guidelines for Designing High-Speed FPGA PCBs*
- *AN 224: High-Speed Board Layout Guidelines*
- *AN 75: High-Speed Board Designs*

For more information on Stratix II Signal Integrity, go to: **http://www.altera.com/technology/signal/devices/stratix2/sgl-st2.html**

Step 10: Perform Board Level Simulations

To ensure you determine the correct board constraints, run board level simulations with your preferred board simulator software 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 you with an initial indication of how well the memory interface performs. The simulations should be performed on the data, data strobe, command, and address signals. If the memory interface does not have good signal integrity, adjust the settings, such as the drive strength setting, termination scheme, or termination values to improve the signal integrity (realize that changing these settings affects your timing and you may have to go back to your timing closure if these change). There are various electronic design automation (EDA) simulation tools available to perform board level simulations.

Refer to *AN 408: DDR2 Memory Interface Termination, Drive Strength and Loading Design Guidelines* for detailed information about understanding the different effects on signal integrity design.

Step 11: Verify FPGA Functionality

Perform system level verification to correlate the system against your design targets. You can use Altera's SignalTap® II Embedded Logic Analyzer to help in this effort.

Refer to *Chapter 13: Design Debugging Using the SignalTap II Embedded Logic Analyzer* of the *Quartus II Software Handbook* in *Volume 3* for detailed information about using SignalTap II.

If you would like step-by-step instructions for how to test the example design created by the legacy memory controller on the board using the SignalTap II Logic Analyzer, refer to *AN 380: Test DDR or DDR2 SDRAM Interfaces on Hardware Using the Example Driver.*

Conclusion

Memory interfaces are useful in many different applications. Stratix II offers a complete solution, from hardware, to software and documentation that may all be used to help build a robust, high performance memory interface.

Stratix II device features allow memory interfaces to run up to 333 MHz (666 Mbps) with Altera’s newest memory controller that implements the ALTMEMPHY data path.

With a straightforward design flow, empowered with collateral and design examples, customers can easily implement their memory interface with confidence that the system runs as designed.

Document Revision History

Table 5 shows the revision history for this document.

Table 5. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
July 2007 v1.1	Change title of referenced *AN 413, Using Legacy Integrated Static Data Path and Controller Megafunction with HardCopy II Structured ASICs*, from title, Implementing External Memory Interfaces in Hardcopy II Devices. Added links to all reference documentation. Changed revision number and date.	—
February 2007 v1.0	Initial Release	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Literature Services:
literature@altera.com

I.S. EN ISO 9001