Introduction

DDR3 SDRAM is the latest generation of DDR SDRAM technology, with improved power, higher data bandwidth, and enhanced signal quality by offering multiple on-die termination (ODT) selection and output driver impedance control. The high performance DDR3 SDRAM can be used in a broad range of applications, such as PCs, embedded processor systems, image processing, storage, communications, and networking.

Although DDR2 SDRAM is currently the more popular SDRAM, designers looking to save system power and increase system performance should consider using DDR3 SDRAM. DDR3 SDRAM offers lower power by using 1.5 V for the supply and I/O voltage compared to the 1.8-V supply and I/O voltage used by DDR2 SDRAM. DDR3 SDRAM also has better performance compared to DDR2 SDRAM by increasing the data rate per pin and the number of banks.

Stratix® III devices support DDR3 SDRAM interfacing with dedicated DQS circuitry, write and read-leveling circuitry. Table 1 displays the maximum clock frequency for DDR3 SDRAM in Stratix III devices.

Table 1. DDR3 SDRAM Maximum Clock Frequency Supported in Stratix III Devices Notes (1), (2), (3)

<table>
<thead>
<tr>
<th>DDR Memory Type</th>
<th>I/O Standard</th>
<th>-2 Speed Grade (MHz)</th>
<th>-3 Speed Grade (MHz)</th>
<th>-4 Speed Grade (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR3 SDRAM</td>
<td>SSTL_15</td>
<td>Top/Bottom I/Os</td>
<td>Left/Right I/Os (4)</td>
<td>Top/Bottom I/Os</td>
</tr>
<tr>
<td></td>
<td></td>
<td>400</td>
<td>300</td>
<td>333</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Top/Bottom I/Os</td>
<td>Left/Right I/Os (4)</td>
<td>Top/Bottom I/Os</td>
</tr>
<tr>
<td></td>
<td></td>
<td>300</td>
<td>333</td>
<td>TBD (5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TBD (5)</td>
</tr>
</tbody>
</table>

Notes to Table 1

(1) Numbers are preliminary until characterization is final.
(2) Performance is based on 1.1-V core voltage.
(3) This applies for interfaces with both modules and components.
(4) Left/right I/O banks have lower maximum performance than the top/bottom I/O banks due to the left/right I/Os having LVDS I/O support.
(5) Support will be evaluated after characterization.

This application note describes the FPGA design flow to implement an external memory interface using Stratix III devices, and provides design guidelines. Use this application note in conjunction with the following Stratix III literature:
This application note demonstrates how to use the FPGA design flow to generate an example design featuring a DDR3 SDRAM memory interface using the data path provided with Altera’s® ALTMEMPHY megafuntion.

FPGA Design Flow

Altera recommends the design guidelines described in this section as best practices for successful memory interface implementation in Stratix III devices. These guidelines provide the fastest out-of-the-box experience with external memory interfaces in Stratix III devices. Figure 1 illustrates the user design flow required for Stratix III memory interfaces with a detailed discussion of each step in the following sections.
Figure 1. Design Flow for Implementing External Memory Interfaces in Stratix III Devices

1. Select Device
2. Create a Quartus II Project
3. Instantiate PHY and Controller
4. Add Constraints
5. Compile Design
6. Perform Gate-Level Simulation (Optional)
8. Determine Board Design Constraints
9. Perform Board Level Simulations
10. Is Eye Diagram Verified? Yes → Verify FPGA Functionality, No → Adjust Termination, Drive Strength, etc.
11. Verify FPGA Functionality

13. Perform RTL/Functional Simulation (Optional)
Step 1: Select Device

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

\[ \text{Bandwidth} = \text{Data width (bits)} \times \text{data rate transfer (1/sec)} \times \text{efficiency} \]

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

After determining the bandwidth of the memory interface, determine which memory and FPGA to use.

Refer to the Selecting the Right High-Speed Memory Technology for Your System white paper for information on selecting the different memory types.

In addition, Altera’s FPGA devices support various data widths for different memory interfaces. The memory interface support between density and package combinations is different, therefore you must determine which FPGA device density and package combination is best suited for your application.

Refer to the External Memory Interfaces chapter of the FPGA Device Handbook for information on the FPGA 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, 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 using the ALTMEMPHY megafunction for all Stratix III memory interface designs for data path/PHY. This megafunction features a license-free physical interface that can be used with the
standard Altera controller or your own custom controller. Using the Altera physical interface provides you with a proven solution for memory interfaces using Stratix III devices.

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

The DDR3 SDRAM High Performance Controller is not available in Quartus II version 6.1, but will be available in a future version of Quartus II.

Refer to the ALTMEMPHY Megafunction User Guide for detailed information on instantiating the PHY.

**Step 4: Perform RTL/Functional Simulation (Optional)**

When instantiating the ALTMEMPHY megafunction by 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 a cycle-accurate HDL model file produced by the Quartus II software. When instantiating the memory interface using Altera’s High Performance Controller, it generates an example design and a testbench, in addition to the ALTMEMPHY megafunction simulation model. The models work with Altera-supported VHDL and Verilog HDL simulators.

Simulation for the ALTMEMPHY megafunction and High Performance Controller is not supported in Quartus II version 6.1, but will be supported in a future version of Quartus II.

**Step 5: Add Constraints**

The next step in the design process is to add all key constraints related to the external memory interface, including timing, pin locations, I/O standards, and pin loading assignments. The ALTMEMPHY megafunction only supports timing analysis using the TimeQuest Timing Analyzer with Synopsys Design Constraints (SDC) assignments. These constraints are derived from the parameters you entered for the ALTMEMPHY megafunction or the High-Performance Controller, based
on the DDR3 SDRAM data sheet and tolerances from the board layout. The ALTMEMPHY megafunction uses TimeQuest timing constraints along with the timing driven fitter to achieve timing closure.

After instantiating the ALTMEMPHY megafunction, the ALTMEMPHY MegaWizard generates the following files that you need in order to properly constrain the design.

- `<variation_name>_phy_ddr_timing.sdc` to set timing constraints
- `<variation_name>_pin_assignments.tcl` to add I/O standard setting assignments
- `<variation_name>_phy_assign_dq_groups.tcl` to add the DQ group assignments to relate the DQ and DQS pin groups together for the Quartus II fitter to place them correctly

These script files are based on the design name used when instantiating the ALTMEMPHY megafunction. If you plan to use your own top-level design, you must edit the scripts to match your custom top-level design.

Refer to the ALTMEMPHY Megafunction User Guide for detailed information on creating, generating, and setting the constraints for the design.

To determine which drive strength and termination to use, refer to AN408: DDR2 Memory Interface Termination and Drive Strength Loading.

**Step 6: Compile Design and Verify Timing Closure**

After constraining the design, compile the design in Quartus II. During the generation of the ALTMEMPHY megafunction or the High Performance SDRAM Memory Controller, the MegaWizard generates a report timing script called `<variation_name>_phy_report_timing.tcl`. After compiling the design in Quartus II, run this script and it produces the timing report for different paths, such as write data, read data and address/command, and core (entire interface) timing path in the design.

Refer to AN438: Constraining and Analyzing Timing for External Memory Interfaces for detailed information on timing analysis and reporting using the ALTMEMPHY megafunction.

**Step 7: 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, in the case of the address/command margin, the
address/command outputs are clocked by an address/command clock. This clock can be different with respect to the system clock, which is 0°. The system clock is used to clock the clock outputs going to the memory. If the report timing script indicates that using the default phase setting for the address/command clock results in more hold than setup, adjust the address/command clock to be less negative than the default phase setting with respect to the system clock so that there will be less hold margin. Similarly, you adjust the address/command clock to be more negative than the default phase setting with respect to system clock if there is more setup margin.

Refer to the *ALTMEMPHY Megafunction User Guide* for detailed information on the clocks used in the ALTMEMPHY megafuction.

**Step 8: Determine Board Design Constraints**

After closing the timing for the design, evaluate the trade-offs posed by various board design choices. Different factors contribute to signal integrity affecting the overall timing margin for both 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. Evaluate 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.

**Step 9: Perform Board Level Simulations**

To determine the correct board constraints, run board level simulations to see if the settings provide the optimum 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 will perform. There are various electronic design automation (EDA) simulation tools available to perform board level simulations. Perform simulations 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 your timing and you may have to go back to your timing closure if these change).

**Step 10: Verify FPGA Functionality**

You can perform system level verification to correlate the system against your design targets. Use Altera’s SignalTap II to help in this effort.
Refer to Chapter 13: Design Debugging Using the SignalTap II Embedded Logic Analyzer of the Quartus II Software Handbook Volume 3 for detailed information on using SignalTap® II.

400MHz DDR3 SDRAM Example Design Flow

In the following sections, use the design flow described in the preceding sections to design a 8-bit wide 400-MHz/800-Mbps DDR3 SDRAM memory interface. The example design will also provide some recommended settings, including termination scheme and drive strength setting in order to simplify your design.

Step 1: Select Device

For this example design, choose the EP3SL150F1152-C2, which supports 8-bit wide DDR3 SDRAM at 400 MHz. For the memory device, choose an 8-bit wide, 32-MB DDR3 SDRAM device.

Step 2: Create a Quartus II Project

After choosing a device, create a project in the Quartus II software targeting the EP3SL150F1152-C2 device as shown in Figure 2.
Figure 2. Creating Quartus II Project Targeting EP3SL150F1152-C2

For detailed step-by-step instructions on how to create a Quartus II project, refer to the Tutorial in the Quartus II software. To do this click the Help menu in Quartus II. Select Tutorial.

Step 3: Instantiate PHY and Controller

After creating a Quartus II Project, instantiate the ALTMEMPHY megafunction. For this example design, instantiate the ALTMEMPHY megafunction using the MegaWizard Plug-In Manager. Find the ALTMEMPHY megafunction in the I/O section of the MegaWizard Plug-In Manager as shown in Figure 3. For this example, choose `ddr3_phy` to be the name of the ALTMEMPHY megafunction.
After the ALTMEMPHY megafuction is invoked, configure the ALTMEMPHY megafuction to interface with a DDR3 SDRAM memory interface. Figure 4 shows the configuration of the ALTMEMPHY megafuction for a 400-MHz, 8-bit wide DDR3 SDRAM memory interface.
In the ALTMEMPHY megafuction **Memory Settings** panel, set the Speed grade of the Stratix III device used in the design to 2. In the **Memory Settings** panel, select from a list of memory presets. By choosing **DDR3 SDRAM** for the Memory type parameter as shown in Figure 4, reduce the total number of memory presets to only DDR3 SDRAM. For this example, select the available memory preset for an 8-bit wide, 32-MB preliminary DDR3 SDRAM device. If required, it is possible to create memory presets by clicking **Modify parameters**, which brings up the **Preset Editor** dialog box where modifications may be made to the memory presets as shown in Figure 5.
After selecting the desired memory type, set up the physical layer (PHY) setting of the ALTMEMPHY megafunction.

The DDR3 SDRAM has a write requirement ($t_{DQSS}$) that states the positive edge of the DQS signal on writes must be within $\pm 25\%$ ($\pm 90^\circ$) of the positive edge of the DDR3 SDRAM clock input. In the **PHY Settings** panel, the default setting uses the DDR registers in the I/O Element (IOE) to generate the CK and CK# signals.

Although there is an option to use dedicated PLL outputs to generate CK and CK#, do not select this option. If this option is selected, the ALTMEMPHY megafunction gives an error.
Altera recommends using the default setting to generate CK and CK# signals.

In the **PHY Settings** panel, the board skew parameter for the board was added in the Board Timing Parameters section. This timing parameter is the board trace variation between the CK, CK#, address, command, and control signals. The default value is set to 50 ps. The ALTMEMPHY megafunction supports automatic calibration of the DQ and DQS signals for DDR3 SDRAM, so the board skew is automatically calibrated out during system calibration.

**Figure 6** displays the dialog box used to set CK/CK# generation and board timing parameters.
For detailed step-by-step instructions on configuring the ALTMEMPHY megafuntion, refer to the *ALTMEMPHY Megafunction User Guide*.

Generate the ALTMEMPHY after completing its configuration.
Step 4: Perform RTL/ Functional Simulation (Optional)

During the instantiation of the ALTMEMPHY megafunction, there is an option to generate a simulation model of the ALTMEMPHY megafunction so you can perform functional simulation on your design. Simulation of the DDR3 SDRAM memory interface with the ALTMEMPHY megafunction is not supported in Quartus II version 6.1.

Step 5: Add Constraints

Instantiating the ALTMEMPHY megafunction generates constraints files for the design. The timing constraint file, `ddr3_phy_ddr_timing.sdc`, constrains the clock and input/output delay on the ALTMEMPHY megafunction. To add the timing constraints, go to the Assignments menu and click the Settings option. In the Settings dialog box, under Timing Analysis Settings, select the TimeQuest Timing Analyzer option. Select the SDC file and click Add, as shown in Figure 7. You can use other SDC files together with the generated SDC file. Make sure that you add all the SDC files in order for the TimeQuest Timing Analyzer to read from them.
Next, the pin assignment script, `ddr3_phy_pin_assignments.tcl`, sets up the I/O standards for the DDR3 SDRAM memory interface. The DQ group assignment script, `ddr3_phy_assign_dq_groups.tcl`, relates the DQ and DQS pin groups together for the Quartus II fitter to place them correctly. Run the pin and DQ group assignment scripts to add the I/O standards and DQ group assignments to the example design. To do this, go to the Tools menu and click **Tcl Scripts**. The **Tcl Scripts** dialog box appears as shown in Figure 8.
Another way to run these scripts is by sourcing the scripts in the Quartus II Tcl Console panel as shown in Figure 9.

Commands displayed in Figure 9 include:
Design Guidelines for Implementing DDR3 SDRAM Interfaces in Stratix III Devices

- source ddr3_phyassign_dq_groups.tcl
- source ddr3_phy_pin_assignments.tcl
- export_assignments

**Step 6: Compile Design and Verify Timing Closure**

Before compiling the design, set the top level entity of the project to the design generated by the ALTMEMPHY megafunction. In this case, the design is called `ddr3_phy.v`. After setting the design as the top level entity, set up the Quartus II software to ensure the remaining unconstrained paths are routed with the highest speed and efficiency. To do this, click the Assignments tab in the Quartus II software and select Settings. Click Analysis and Synthesis Settings to check the Optimization Technique in Quartus II. The default setting for the Optimization Technique is Balanced as shown in Figure 10.

![Figure 10. Recommended Setting on the Optimization Technique in Quartus II](image)
Next, click Fitter Settings. In the Fitter Settings panel, check that the Fitter effort is set to the default setting Auto Fit as shown in Figure 11.

**Figure 11. Recommended Setting on the Fitter Effort in Quartus II**

Now compile the design.

After successfully compiling the design in the Quartus II software, run the timing reporting script generated by the ALTMEMPHY megafunction during the megafunction instantiation called `ddr3_phy_report_timing.tcl`, which will produce the timing report for the design.

Run the report timing script, `ddr3_phy_report_timing.tcl`, either in the Quartus II Tcl Console, or in the TimeQuest Timing Analyzer window. Running the report timing script reports the following margins on the following paths:
Design Guidelines for Implementing DDR3 SDRAM Interfaces in Stratix III Devices

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

The report timing script does not perform timing analysis on the write/read leveling circuitry datapath of the DDR3 SDRAM as the timing of these datpaths is guaranteed correct by design. Figure 12 shows the timing margins reported in the Quartus II software after running the report timing script in the Quartus II Tcl Console.

Figure 12. Timing Margin Reported in Quartus II

To run the report timing script in the TimeQuest Timing Analyzer window, open the window in the Quartus II software. Then complete the following tasks:

- Create timing netlist
- Read SDC File
- Update timing netlist
Perform these tasks by double-clicking **Update Timing Netlist** in the left pane, which will automatically run **Create Timing Netlist** and **Read SDC**. After a task is executed, it becomes green. When all tasks have been completed, run the report timing script by going to the Script menu and clicking **Run Tcl Script**. Figure 13 shows the timing margin reported in the TimeQuest Timing Analyzer after running the report timing script, which is the same as Quartus II results shown in Figure 12.

Figure 13. Timing Margin Reported in TimeQuest Timing Analyzer

For more information on the **TimeQuest Timing Analyzer** window, refer to the **TimeQuest Timing Analyzer User Guide**.

Refer to **AN438: Constraining and Analyzing Timing for External Memory Interfaces** for detailed information on timing analysis and reporting using the ALTMEMPHY megafunction.
Step 7: Adjust Constraints

The timing margin report shows that there is positive setup and hold margin in both the address/command, read and write datapath. However, the hold time on the address/command datapath is quite small. Adjusting the clock that is feeding the address/command output registers can improve the hold margin on the address/command datapath. To find out which clock is clocking the address/command registers, click on the address/command report in the Report section in the TimeQuest Timing Analyzer window and select the path that indicates the minimum hold time as shown in Figure 14.

This report indicates that clk6 of the PLL is the clock that is clocking the address/command registers. Go to the PLL megafunction and change the phase setting of clk6. For this design, the initial phase setting of clk6 is set to 315° resulting in the address/command being launched too early and resulting in a small hold time. The hold margin reported is 14 ps, so delay clk6 by increasing the phase setting. For this example, clk6 is
200 MHz. To increase the hold margin, delay c1k6 to 330°. 15° delay in c1k6 results in an increase in the hold margin by 208 ps which results in a final hold margin of 222 ps.

After modifying the c1k6 phase setting, recompile the design for the new PLL setting to take effect and run the report timing script again. Figure 15 shows the timing margin reported in the Quartus II software after adjusting the phase setting of c1k6.

The timing report indicates that the final margin reported is as expected.

**Step 8: Determine Board Design Constraints**

The Stratix III devices support both series and parallel on-chip termination (OCT) resistors to improve signal integrity. Another benefit of using the Stratix III OCT features is eliminating 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 Stratix III
devices are performing a write or a read operation. The OCT features offer user-mode calibration to compensate for any variation in voltage and temperature during normal operation to ensure that the OCT values remain constant. The parallel and series OCT features on the Stratix III devices are available in either 25 Ω or 50 Ω settings.

Refer to the Selectable I/O Standards in Stratix III chapter of the FPGA Device Handbook for information on the OCT features.

On the DDR3 SDRAM, there is a dynamic parallel on-die termination feature that can be turned on when the FPGA is writing to the DDR3 SDRAM memory and turned off when the FPGA is reading from the DDR3 SDRAM memory. To further improve signal integrity, DDR3 SDRAM supports output driver impedance control so the driver impedance will match the transmission line. The ODT and output driver impedance features are programmable and are controlled by an external resistor, RZQ where the recommended value is 240 Ω with ±1% tolerance. For the ODT feature, there are 2 settings for parallel on-die termination (RTT_WR and RTT_NOM). RTT_NOM is used when the ODT signal going to the memory is asserted and there is not a write operation occurring to that chip; RTT_WR is used when the ODT signal going to the memory is asserted and a write is occurring on that specific chip.

The output driver impedance is used during reads and is achieved by de-asserting the ODT pin on the memory device.

Refer to the DDR3 SDRAM datasheet for additional information on available settings of the ODT and the output driver impedance features, and the timing requirements for driving the ODT pin.

Figure 16 illustrates the write operation to the DDR3 SDRAM memory with the ODT feature turned on, and using the 50-Ω series OCT feature of the Stratix III FPGA device. In this setup, the transmitter (FPGA) is properly terminated with matching impedance to the transmission line thus eliminating any ringing or reflection. The receiver (DDR3 SDRAM memory) is also properly terminated when the dynamic ODT setting is at 60 Ω (RTT_WR = RZQ/4).
**Figure 16. Write Operation to DDR3 SDRAM Memory Using Parallel ODT of the DDR3 SDRAM Memory and 50-Ω serial OCT of the Stratix III FPGA Device**

![Diagram]

**Figure 17 illustrates the read operation from the DDR3 SDRAM memory using the parallel OCT feature of the Stratix III device together with the output drive impedance control feature of the DDR3 SDRAM memory. In this setup, the driver’s (DDR3 SDRAM memory) output impedance is set to 40 Ω; \( RTT_{WR} = RZQ/6 \) which combines with the on DIMM series resistor to match the transmission line resulting in optimal signal transmission to the receiver. On the receiver (FPGA) side, it is properly terminated with 50 Ω which matches the impedance of the transmission line thus eliminating any ringing or reflection.**

**Figure 17. Read Operation From DDR3 SDRAM Memory Using the Output Driver Impedance Control Feature of the DDR3 SDRAM Memory and the Parallel OCT of the Stratix III FPGA Device**

![Diagram]
Finally, the loading seen by the FPGA during writes to the memory is different between a system using dual-inline memory modules (DIMMs) versus a system using components. The additional loading from the DIMM connector can reduce the edge rates of the signals arriving at the memory thus affecting available timing margin.

Even though the OCT feature in Stratix III is available, the ALTMEMPHY megafunction and High Performance Controller do not support the usage of the OCT feature in Quartus II version 6.1.


**Conclusion**

Stratix III devices have dedicated circuitry to interface with DDR3 SDRAM at speeds up to 400 MHz with comfortable and consistent margins. The advanced clocking features available in Stratix III devices allow for a high performance, versatile interface to DDR3 SDRAM. For applications requiring lower power consumption and the greater memory bandwidth offered by DDR3 SDRAM, Altera offers a complete memory solution for Stratix III FPGA.

**References**

DDR3 SDRAM Specification, Revision 0.0, January 2007, Samsung Electronics

**Document Revision History**

<table>
<thead>
<tr>
<th>Date and Document Version</th>
<th>Changes Made</th>
<th>Summary of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 2007 v1.0</td>
<td>Initial Release</td>
<td></td>
</tr>
</tbody>
</table>