

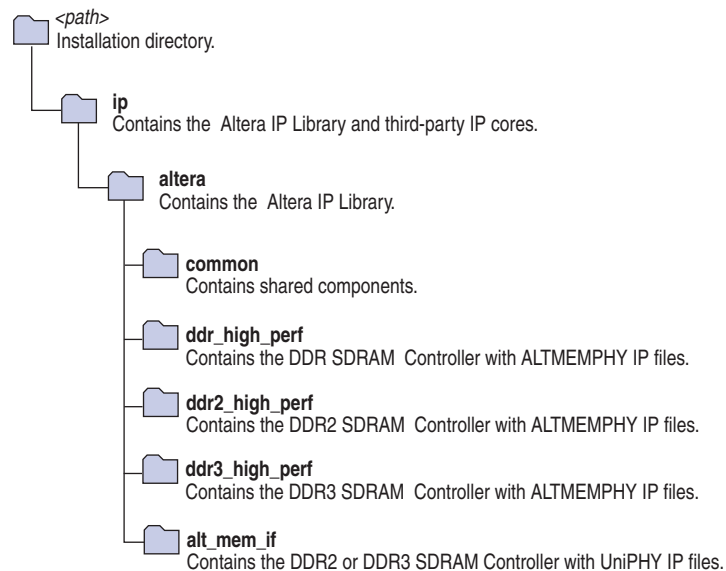
This chapter describes the general overview of the Altera® IP core design flow to help you quickly get started with any Altera IP core. The Altera IP Library is installed as part of the Quartus® II installation process. You can select and parameterize any Altera IP core from the library. Altera provides an integrated parameter editor that allows you to customize IP cores to support a wide variety of applications. The parameter editor guides you through the setting of parameter values and selection of optional ports. The following section describes the general design flow and use of Altera IP cores.

Installation and Licensing

The Altera IP Library is distributed with the Quartus II software and downloadable from the Altera website (www.altera.com).

Figure 8–1 shows the directory structure after you install the memory controller with the memory IP, where *<path>* is the installation directory. The default installation directory on Windows is *c:\altera\<version>*; on Linux it is */opt/altera<version>*.

Figure 8–1. Directory Structure



You can evaluate an IP core in simulation and in hardware until you are satisfied with its functionality and performance. Some IP cores require that you purchase a license for the IP core when you want to take your design to production. After you purchase a license for an Altera IP core, you can request a license file from the [Altera Licensing](#) page of the Altera website and install the license on your computer. For additional information, refer to [Altera Software Installation and Licensing](#).

Free Evaluation

Altera's OpenCore Plus evaluation feature is only applicable to the DDR, DDR2 and DDR3 SDRAM HPC. With the OpenCore Plus evaluation feature, you can perform the following actions:

- Simulate the behavior of a megafunction (Altera MegaCore® function or AMPPSM megafunction) within your system.
- Verify the functionality of your design, as well as evaluate its size and speed quickly and easily.
- Generate time-limited device programming files for designs that include MegaCore functions.
- Program a device and verify your design in hardware.

You need to purchase a license for the megafunction only when you are completely satisfied with its functionality and performance, and want to take your design to production.

OpenCore Plus Time-Out Behavior

OpenCore Plus hardware evaluation can support the following two modes of operation:

- Untethered—the design runs for a limited time
- Tethered—requires a connection between your board and the host computer. If tethered mode is supported by all megafunctions in a design, the device can operate for a longer time or indefinitely

All megafunctions in a device time out simultaneously when the most restrictive evaluation time is reached. If there is more than one megafunction in a design, a specific megafunction's time-out behavior may be masked by the time-out behavior of the other megafunctions.



For MegaCore functions, the untethered time-out is 1 hour; the tethered time-out value is indefinite.

Your design stops working after the hardware evaluation time expires and the `local_ready` output goes low.

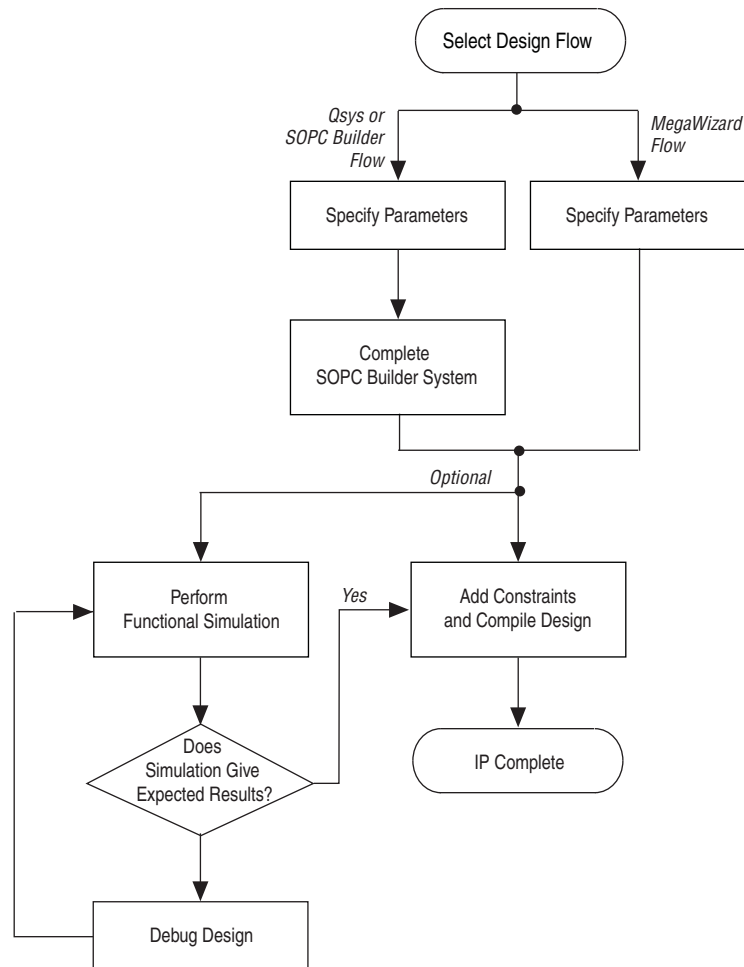
Design Flow

You can implement the memory controllers with ALTMEMPHY IP or UniPHY IP using any of the following flows:

- MegaWizard™ Plug-In Manager flow
- SOPC Builder flow
- Qsys Flow

Figure 8–2 shows the stages for creating a system in the Quartus II software using the available flows.

Figure 8–2. Design Flows (1)



Note to Figure 8–2:

(1) Altera IP cores may or may not support the Qsys and SOPC Builder design flows.

The MegaWizard Plug-In Manager flow offers the following advantages:

- Allows you to parameterize an IP core variant and instantiate into an existing design
- For some IP cores, this flow generates a complete example design and testbench

The SOPC Builder flow offers the following advantages:

- Generates simulation environment
- Allows you to integrate Altera-provided custom components
- Uses Avalon® memory-mapped (Avalon-MM) interfaces

The Qsys flow offers the following additional advantages over SOPC Builder:

- Provides visualization of hierarchical designs
- Allows greater performance through interconnect elements and pipelining
- Provides closer integration with the Quartus II software


MegaWizard Plug-In Manager Flow


The MegaWizard Plug-In Manager flow allows you to customize the memory controller with ALTMEMPHY or UniPHY IP, and manually integrate the function into your design.

Specifying Parameters


To specify parameters using the MegaWizard Plug-In Manager flow, perform the following steps:

1. Create a Quartus II project using the **New Project Wizard** available from the File menu.
2. In the Quartus II software, launch the **MegaWizard Plug-in Manager** from the Tools menu, and follow the prompts in the MegaWizard Plug-In Manager interface to create or edit a custom IP core variation.
3. Select a memory controller with the memory IP in the **Installed Plug-Ins** list in the **External Memory** folder.
4. Specify the parameters on all pages in the **Parameter Settings** tab.


 For detailed explanation of the parameters, refer to “[Parameterizing Memory Controllers with ALTMEMPHY IP](#)” on page 8–38 and “[Parameterizing Memory Controllers with UniPHY IP](#)” on page 8–57.

 Some IP cores provide preset parameters for specific applications. If you wish to use preset parameters, click the arrow to expand the **Presets** list, select the desired preset, and then click **Apply**. To modify preset settings, in a text editor modify the `<installation directory>/ip/altera/alt_mem_if/alt_mem_if_interfaces/alt_mem_if_<memory_protocol>_emif/alt_mem_if_<memory_protocol>_mem_model.qprs`.


5. If the IP core provides a simulation model, specify appropriate options in the wizard to generate a simulation model.

 Altera IP supports a variety of simulation models, including simulation-specific IP functional simulation models and encrypted RTL models, and plain text RTL models. These are all cycle-accurate models. The models allow for fast functional simulation of your IP core instance using industry-standard VHDL or Verilog HDL simulators. For some cores, only the plain text RTL model is generated, and you can simulate that model.

 For more information about functional simulation models for Altera IP cores, refer to *Simulating Altera Designs* chapter in volume 3 of the *Quartus II Handbook*.


 Use the simulation models only for simulation and not for synthesis or any other purposes. Using these models for synthesis creates a nonfunctional design.

6. This step applies to memory controllers with ALTMEMPHY IP. If the parameter editor includes **EDA** and **Summary** tabs, follow these steps:
 - a. Some third-party synthesis tools can use a netlist that contains the structure of an IP core but no detailed logic to optimize timing and performance of the design containing it. To use this feature if your synthesis tool and IP core support it, turn on **Generate netlist**.


 When targeting a VHDL simulation model, the MegaWizard Plug-In Manager still generates the `<variation_name>_alt_mem_phy.v` for the Quartus II synthesis. Do not use this file for simulation. Use the `<variation_name>.vho` for simulation instead.

The ALTMEMPHY megafunction only supports functional simulation. You cannot perform timing or gate-level simulation when using the ALTMEMPHY megafunction.

- b. On the **Summary** tab, if available, select the files you want to generate. A gray checkmark indicates a file that is automatically generated. All other files are optional.

 If file selection is supported for your IP core, after you generate the core, a generation report (`<variation name>.html`) appears in your project directory. This file contains information about the generated files.

7. Click the **Finish** button, the parameter editor generates the top-level HDL code for your IP core, and a simulation directory which includes files for simulation.

 The **Finish** button may be unavailable until all parameterization errors listed in the messages window are corrected.

8. Click **Yes** if you are prompted to add the `.qip` to the current Quartus II project. You can also turn on **Automatically add Quartus II IP Files to all projects**.

9. This step applies to memory controllers with ALTMEMPHY IP. If you are using the UniPHY IP, for the high-performance controller (HPC or HPC II), set the `<variation_name>_example_top.v` or `.vhd` to be the project top-level design file.
 - a. On the File menu, click **Open**.
 - b. Browse to `<variation_name>_example_top` and click **Open**.
 - c. On the Project menu, click **Set as Top-Level Entity**.

You can now integrate your custom IP core instance in your design, simulate, and compile. While integrating your IP core instance into your design, you must make appropriate pin assignments. You can create a virtual pin to avoid making specific pin assignments for top-level signals while you are simulating and not ready to map the design to hardware.

For some IP cores, the generation process also creates complete example designs. An example design for hardware testing is located in the `<variation_name>_example_design/example_project/` directory. An example design for RTL simulation is located in the `<variation_name>_example_design/simulation/` directory.



For information about the Quartus II software, including virtual pins and the MegaWizard Plug-In Manager, refer to [Quartus II Help](#).

Constraining the Design

After you have generated the memory IP MegaCore function, you may need to set timing constraints and perform timing analysis using the Quartus II TimeQuest Timing Analyzer. When you generate the MegaCore function, the MegaWizard Plug-In Manager also generates a Synopsys Design Constraint File (`.sdc`), `<variation_name>.sdc`, and a pin assignment script, `<variation_name>_pin_assignments.tcl`. Both the `.sdc` and the `<variation_name>_pin_assignments.tcl` scripts support multiple instances. These scripts iterate through all instances of the core and apply the same constraints to all of them. You can derive the timing constraints from the external device data sheet and tolerances from the board layout.

For more information about timing constraints and analysis, refer to the [Analyzing Timing of Memory IP](#) chapter.

Add Pins and DQ Group Assignments


The `<variation_name>_pin_assignments.tcl` script, sets up the I/O standards and the input/output termination for the memory IP. This script also helps to relate the DQ pin groups together for the Quartus II Fitter to place them correctly.


The pin assignment script does not create a PLL reference clock for the design. You must create a clock for the design and provide pin assignments for the signals of both the example driver and testbench that the MegaCore variation generates.

Run the `<variation_name>_pin_assignments.tcl` script to add the input and output termination, I/O standards, and DQ group assignments to the example design. To run the pin assignment script, follow these steps:

1. On the Processing menu, point to **Start**, and click **Start Analysis and Synthesis**.

2. On the Tools menu click **Tcl Scripts**.
3. Specify the **pin_assignments.tcl** and click **Run**.

 If the PLL input reference clock pin does not have the same I/O standard as the memory interface I/Os, a no-fit might occur because incompatible I/O standards cannot be placed in the same I/O bank.

 If you are upgrading your memory IP from an earlier Quartus II version, follow these steps:


- For UniPHY IP, rerun the **pin_assignments.tcl** script in the later Quartus II revision.
- For ALTMEMPHY IP, delete all the memory non-location I/O assignments and rerun the **pin_assignments.tcl** script.

Compiling the Design

After constraining your design, compile your design in the Quartus II software to generate timing reports to verify whether timing has been met.

To compile the design, on the Processing menu, click **Start Compilation**.

After you have compiled the top-level file, you can perform RTL simulation or program your targeted Altera device to verify the top-level file in hardware.

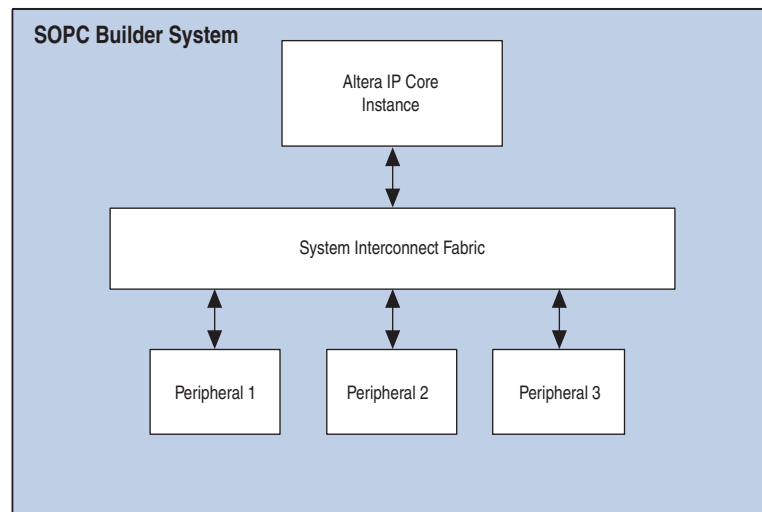
 For more information about simulating the memory IP, refer to the *Simulating Memory IP* chapter.

SOPC Builder Flow

You can use SOPC Builder to build a system that includes your customized IP core. You can easily add other components and quickly create an SOPC Builder system. SOPC Builder automatically generates HDL files that include all of the specified components and interconnections. SOPC Builder defines default connections, which you can modify. The HDL files are ready to be compiled by the Quartus II software to produce output files for programming an Altera device.

Figure 8-3 shows a block diagram of an example SOPC Builder system.

Figure 8-3. SOPC Builder System



For more information about system interconnect fabric, refer to the *System Interconnect Fabric for Memory-Mapped Interfaces* and *System Interconnect Fabric for Streaming Interfaces* chapters in the *SOPC Builder User Guide* and to the *Avalon Interface Specifications*.


For more information about SOPC Builder and the Quartus II software, refer to the *SOPC Builder Features* and *Building Systems with SOPC Builder* sections in the *SOPC Builder User Guide* and to Quartus II Help.


Specifying Parameters

To specify IP core parameters in the SOPC Builder flow, follow these steps:


1. Create a new Quartus II project using the **New Project Wizard** available from the File menu.
2. On the Tools menu, click **SOPC Builder**.
3. For a new system, specify the system name and language.
4. On the **System Contents** tab, double-click the name of your IP core to add it to your system. The relevant parameter editor appears.

5. Specify the required parameters in the parameter editor. For detailed explanations of these parameters, refer to “Parameterizing Memory Controllers with ALTMEMPHY IP” on page 8-38 and “Parameterizing Memory Controllers with UniPHY IP” on page 8-57.

 Some IP cores provide preset parameters for specific applications. If you wish to use preset parameters, click the arrow to expand the **Presets** list, select the desired preset, and then click **Apply**. To modify preset settings, in a text editor modify the `<installation_directory>/ip/altera/alt_mem_if/alt_mem_if_interfaces/alt_mem_if_<memory_protocol>_emif/alt_mem_if_<memory_protocol>_mem_model.qprs`.

 You must also turn on **Generate SOPC Builder compatible resets** on the **Controller Settings** tab when parameterizing those cores.

6. Click **Finish** to complete the IP core instance and add it to the system.

 The **Finish** button may be unavailable until all parameterization errors listed in the messages window are corrected.

Completing the SOPC Builder System

To complete the SOPC Builder system, follow these steps:

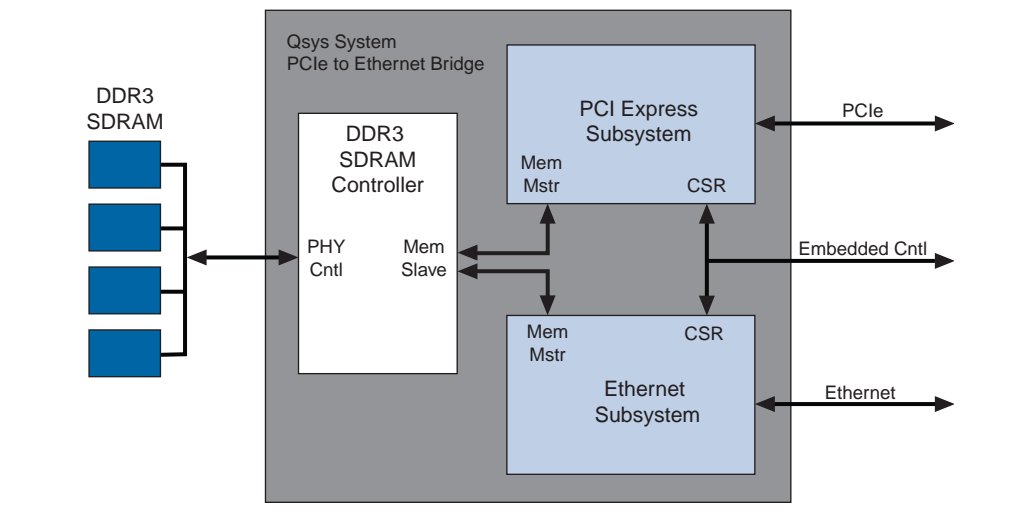
1. Add and parameterize any additional components. Some IP cores include a complete SOPC Builder system design example.
2. Use the Connection panel on the **System Contents** tab to connect the components.
3. By default, clock names are not displayed. To display clock names in the **Module Name** column and the clocks in the **Clock** column in the **System Contents** tab, click **Filters** to display the **Filters** dialog box. In the **Filter** list, click **All**.
4. Click **Generate** to generate the system. SOPC Builder generates the system and produces the `<system name>.qip` that contains the assignments and information required to process the IP core or system in the Quartus II Compiler.
5. In the Quartus II software, click **Add/Remove Files in Project** and add the `.qip` to the project.
6. Compile your design in the Quartus II software.



Qsys System Integration Tool Design Flow

You can use the Qsys system integration tool to build a system that includes your customized IP core. You easily can add other components and quickly create a Qsys system. Qsys automatically generates HDL files that include all of the specified components and interconnections. In Qsys, you specify the connections you want. The HDL files are ready to be compiled by the Quartus II software to produce output files for programming an Altera device. Qsys generates Verilog HDL simulation models for the IP cores that comprise your system.

Figure 8-4 shows a high level block diagram of an example Qsys system.

Figure 8-4. Example Qsys System





-  For more information about the Qsys system interconnect, refer to the *Qsys Interconnect* chapter in volume 1 of the *Quartus II Handbook* and to the *Avalon Interface Specifications*.
-  For more information about the Qsys tool and the Quartus II software, refer to the *System Design with Qsys* section in volume 1 of the *Quartus II Handbook* and to Quartus II Help.


Specify Parameters

To specify parameters for your IP core using the Qsys flow, follow these steps:


1. Create a new Quartus II project using the **New Project Wizard** available from the File menu.
2. On the Tools menu, click **Qsys**.
3. In the **Component Library** window, double-click the name of your IP core to add it to your system. The relevant parameter editor appears.

-  Specify the required parameters in all tabs in the Qsys tool. For detailed explanations of these parameters, refer to “*Parameterizing Memory Controllers with ALTMEMPHY IP*” on page 8-38 and “*Parameterizing Memory Controllers with UniPHY IP*” on page 8-57.

-  If your design includes external memory interface IP cores, you must turn on **Generate power-of-2 bus widths for SOPC Builder** on the **Controller Settings** tab when parameterizing those cores.

 Some IP cores provide preset parameters for specific applications. If you wish to use preset parameters, click the arrow to expand the **Presets** list, select the desired preset, and then click **Apply**. To modify preset settings, in a text editor modify the `<installation_directory>/ip/altera/alt_mem_if/alt_mem_if/alt_mem_if_interfaces<memory_protocol>_emif/alt_mem_if_<memory_protocol>_mem_model.qprs`.


4. Click **Finish** to complete the IP core instance and add it to the system.

 The **Finish** button may be unavailable until all parameterization errors listed in the messages window are corrected.

Complete the Qsys System

To complete the Qsys system, follow these steps:

1. Add and parameterize any additional components.
2. Connect the components using the Connection panel on the **System Contents** tab.
3. In the **Export** column, enter the name of any connections that should be a top-level Qsys system port.
4. If you intend to simulate your Qsys system, on the **Generation** tab, select either **Create testbench Qsys system to Standard, BFM for standard Avalon interfaces** to create a testbench with bus functional models (BFMs) attached to all exported interfaces or **Simple, BFM for clocks and resets** to create a testbench with BFMs driving only clocks and reset interfaces.
5. To generate a simulation model for the testbench Qsys system at the same time, set **Create testbench simulation model** to **Verilog** or **VHDL**. Set this option to **None** to view or modify the generated testbench system before generating its simulation model.
6. If your system is not part of a Quartus II project and you want to generate synthesis register transfer language (RTL) or high-level hardware description language (HDL) files, turn on **Create HDL design files for synthesis**.
7. Click **Generate** to generate the system. Qsys generates the system and produces the `<system name>.qip` that contains the assignments and information required to process the IP core or system in the Quartus II Compiler.
8. In the Quartus II software, click **Add/Remove Files in Project** and add the `.qip` to the project.
9. Compile your project in the Quartus II software.

 To ensure that the **memory** and **oct** interfaces are exported to the top-level RTL file, be careful not to accidentally rename or delete either of these interfaces in the **Export** column of the **System Contents** tab.

Qsys and SOPC Builder Interfaces

Table 8-1 and Table 8-2 list the DDR2 and DDR3 SDRAM with UniPHY signals available for each interface in Qsys and SOPC Builder and provide a description and guidance on how to connect those interfaces.

Table 8-1. DDR2 SDRAM Controller with UniPHY Interfaces (Part 1 of 5)

Signals in Interface	Interface Type	Description/How to Connect
pll_ref_clk interface		
pll_ref_clk	Clock input	PLL reference clock input.
global_reset interface		
global_reset_n	Reset input	Asynchronous global reset for PLL and all logic in PHY.
soft_reset interface		
soft_reset_n	Reset input	Asynchronous reset input. Resets the PHY, but not the PLL that the PHY uses.
afi_reset interface		
afi_reset_n	Reset output (PLL master/no sharing)	When the interface is in PLL master or no sharing modes, this interface is an asynchronous reset output of the AFI interface. The controller asserts this interface when the PLL loses lock or the PHY is reset.
afi_reset_in interface		
afi_reset_n	Reset input (PLL slave)	When the interface is in PLL slave mode, this interface is a reset input that you must connect to the <i>afi_reset</i> output of an identically configured memory interface in PLL master mode.
afi_clk interface		
afi_clk	Clock output (PLL master/no sharing)	This AFI interface clock can be a full-rate or half-rate memory clock frequency based on the memory interface parameterization. When the interface is in PLL master or no sharing modes, this interface is a clock output.
afi_clk_in interface		
afi_clk	Clock input (PLL slave)	This AFI interface clock can be a full-rate or half-rate memory clock frequency based on the memory interface parameterization. When the interface is in PLL slave mode, you must connect this <i>afi_clk</i> input to the <i>afi_clk</i> output of an identically configured memory interface in PLL master mode.

Table 8-1. DDR2 SDRAM Controller with UniPHY Interfaces (Part 2 of 5)

Signals in Interface	Interface Type	Description/How to Connect
afi_half_clk interface		
afi_half_clk	Clock output (PLL master/no sharing)	The AFI half clock that is half the frequency of afi_clk. When the interface is in PLL master or no sharing modes, this interface is a clock output.
afi_half_clk_in interface		
afi_half_clk	Clock input (PLL slave)	The AFI half clock that is half the frequency of afi_clk. When the interface is in PLL slave mode, this is a clock input that you must connect to the afi_half_clk output of an identically configured memory interface in PLL master mode.
memory interface		
mem_a	Conduit	Interface signals between the PHY and the memory device.
mem_ba		
mem_ck		
mem_ck_n		
mem_cke		
mem_cs_n		
mem_dm		
mem_ras_n		
mem_cas_n		
mem_we_n		
mem_dq		
mem_dqs		
mem_dqs_n		
mem_odt		
mem_ac_parity		
mem_err_out_n		
mem_parity_error_n		

Table 8-1. DDR2 SDRAM Controller with UniPHY Interfaces (Part 3 of 5)

Signals in Interface	Interface Type	Description/How to Connect
avl interface		
avl_ready	Avalon-MM Slave	Avalon-MM interface signals between the memory interface and user logic.
avl_burst_begin		
avl_addr		
avl_rdata_valid		
avl_rdata		
avl_wdata		
avl_be		
avl_read_req		
avl_write_req		
avl_size		
status interface		
local_init_done	Conduit	Memory interface status signals.
local_cal_success		
local_cal_fail		
oct interface		
rup (Stratix® III/IV, Arria® II GZ)	Conduit	OCT reference resistor pins for rup / rdn or rzqin .
rdn (Stratix III/IV, Arria II GZ)		
rzq (Stratix V)		
local_powerdown interface		
local_powerdn_ack	Conduit	This powerdown interface for the controller is enabled only when you turn on Enable Auto Powerdown .
pll_sharing interface		
pll_mem_clk	Conduit	Interface signals for PLL sharing, to connect PLL masters to PLL slaves. This interface is enabled only when you set PLL sharing mode to master or slave.
pll_write_clk		
pll_addr_cmd_clk		
pll_locked		
pll_avl_clk		
pll_config_clk		
pll_hr_clk		
pll_p2c_read_clk		
pll_c2p_write_clk		
pll_dr_clk		
dll_sharing interface		
dll_delayctrl	Conduit	DLL sharing interface for connecting DLL masters to DLL slaves. This interface is enabled only when you set DLL sharing mode to master or slave.

Table 8-1. DDR2 SDRAM Controller with UniPHY Interfaces (Part 4 of 5)

Signals in Interface	Interface Type	Description/How to Connect
oct_sharing interface		
seriesterminationcontrol	Conduit	OCT sharing interface for connecting OCT masters to OCT slaves. This interface is enabled only when you set OCT sharing mode to master or slave.
parallelerminationcontrol		
hcx_dll_reconfig interface		
dll_offset_ctrl_addnsub	Conduit	This DLL reconfiguration interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to enable DLL reconfiguration.
dll_offset_ctrl_offset		
dll_offset_ctrl_addnsub ⁽¹⁾		
dll_offset_ctrl_offset ⁽¹⁾		
dll_offset_ctrl_offsetctrlout ⁽¹⁾		
dll_offset_ctrl_b_offsetctrlout ⁽¹⁾		
hcx_pll_reconfig interface		
configupdate	Conduit	This PLL reconfiguration interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to enable PLL reconfiguration.
phasecounterselect		
phasestep		
phaseupdown		
scanclk		
scanckena		
scandata		
phasedone		
scandataout		
scandone		
hcx_rom_reconfig interface		
hc_rom_config_clock	Conduit	This ROM loader interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to control the loading of the sequencer ROM.
hc_rom_conig_datain		
hc_rom_config_rom_data_ready		
hc_rom_config_init		
hc_rom_config_init_busy		
hc_rom_config_rom_rden		
hc_rom_config_rom_address		
autoprecharge_req interface		
local_autopch_req	Conduit	Precharge interface for connection to a custom control block. This interface is enabled only when you turn on Auto-precharge Control .
user_refresh interface		
local_refresh_req	Conduit	User refresh interface for connection to a custom control block. This interface is enabled only when you turn on User Auto-Refresh Control .
local_refresh_chip		
local_refresh_ack		

Table 8-1. DDR2 SDRAM Controller with UniPHY Interfaces (Part 5 of 5)

Signals in Interface	Interface Type	Description/How to Connect
self_refresh interface		
local_self_rfsh_req	Conduit	Self refresh interface for connection to a custom control block. This interface is enabled only when you turn on Self-refresh Control .
local_self_rfsh_chip		
local_self_rfsh_ack		
ecc_interrupt interface		
ecc_interrupt	Conduit	ECC interrupt signal for connection to a custom control block. This interface is enabled only when you turn on Error Detection and Correction Logic .
csr interface		
csr_write_req	Avalon-MM Slave	Configuration and status register signals for the memory interface, for connection to an Avalon_MM master. This interface is enabled only when you turn on Configuration and Status Register .
csr_read_req		
csr_waitrequest		
csr_addr		
csr_be		
csr_wdata		
csr_rdata		
csr_rdata_valid		
Note to Table 8-1: (1) Signals available only in DLL master mode.		

Table 8-2. DDR3 SDRAM Controller with UniPHY Interfaces (Part 1 of 6)

Signals in Interface	Interface Type	Description/How to Connect
pll_ref_clk interface		
pll_ref_clk	Clock input	PLL reference clock input.
global_reset interface		
global_reset_n	Reset input	Asynchronous global reset for PLL and all logic in PHY.
soft_reset interface		
soft_reset_n	Reset input	Asynchronous reset input. Resets the PHY, but not the PLL that the PHY uses.
afi_reset interface		
afi_reset_n	Reset output (PLL master/no sharing)	When the interface is in PLL master or no sharing modes, this interface is an asynchronous reset output of the AFI interface. This interface is asserted when the PLL loses lock or the PHY is reset.

Table 8-2. DDR3 SDRAM Controller with UniPHY Interfaces (Part 2 of 6)

Signals in Interface	Interface Type	Description/How to Connect
afi_reset_in interface		
afi_reset_n	Reset input (PLL slave)	When the interface is in PLL slave mode, this interface is a reset input that you must connect to the <code>afi_reset</code> output of an identically configured memory interface in PLL master mode.
afi_clk interface		
afi_clk	Clock output (PLL master/no sharing)	This AFI interface clock can be full-rate or half-rate memory clock frequency based on the memory interface parameterization. When the interface is in PLL master or no sharing modes, this interface is a clock output.
afi_clk_in interface		
afi_clk	Clock input (PLL slave)	This AFI interface clock can be full-rate or half-rate memory clock frequency based on the memory interface parameterization. When the interface is in PLL slave mode, this is a clock input that you must connect to the <code>afi_clk</code> output of an identically configured memory interface in PLL master mode.
afi_half_clk interface		
afi_half_clk	Clock output (PLL master/no sharing)	The AFI half clock that is half the frequency of <code>afi_clk</code> . When the interface is in PLL master or no sharing modes, this interface is a clock output.
afi_half_clk_in interface		
afi_half_clk	Clock input (PLL slave)	The AFI half clock that is half the frequency of the <code>afi_clk</code> . When the interface is in PLL slave mode, you must connect this <code>afi_half_clk</code> input to the <code>afi_half_clk</code> output of an identically configured memory interface in PLL master mode.

Table 8-2. DDR3 SDRAM Controller with UniPHY Interfaces (Part 3 of 6)

Signals in Interface	Interface Type	Description/How to Connect
memory interface		
mem_a	Conduit	Interface signals between the PHY and the memory device.
mem_ba		
mem_ck		
mem_ck_n		
mem_cke		
mem_cs_n		
mem_dm		
mem_ras_n		
mem_cas_n		
mem_we_n		
mem_dq		
mem_dqs		
mem_dqs_n		
mem_odt		
mem_reset_n		
mem_ac_parity		
mem_err_out_n		
mem_parity_error_n		
avl interface		
avl_ready	Avalon-MM Slave	Avalon-MM interface signals between the memory interface and user logic.
avl_burst_begin		
avl_addr		
avl_rdata_valid		
avl_rdata		
avl_wdata		
avl_be		
avl_read_req		
avl_write_req		
avl_size		
status interface		
local_init_done	Conduit	Memory interface status signals.
local_cal_success		
local_cal_fail		
oct interface		
rup (Stratix III/IV, Arria II GZ)	Conduit	OCT reference resistor pins for rup/rdn or rzqin.
rdn (Stratix III/IV, Arria II GZ)		
rzq (Stratix V)		

Table 8-2. DDR3 SDRAM Controller with UniPHY Interfaces (Part 4 of 6)

Signals in Interface	Interface Type	Description/How to Connect
local_powerdown interface		
local_powerdn_ack	Conduit	This powerdown interface for the controller is enabled only when you turn on Enable Auto Power Down .
pll_sharing interface		
pll_mem_clk	Conduit	Interface signals for PLL sharing, to connect PLL masters to PLL slaves. This interface is enabled only when you set PLL sharing mode to master or slave.
pll_write_clk		
pll_addr_cmd_clk		
pll_locked		
pll_avl_clk		
pll_config_clk		
pll_hr_clk		
pll_p2c_read_clk		
pll_c2p_write_clk		
pll_dr_clk		
dll_sharing interface		
dll_delayctrl	Conduit	DLL sharing interface for connecting DLL masters to DLL slaves. This interface is enabled only when you set DLL sharing mode to master or slave.
oct_sharing interface		
seriesterminationcontrol	Conduit	OCT sharing interface for connecting OCT masters to OCT slaves. This interface is inabled only when you set OCT sharing mode to master or slave.
parallelerminationcontrol		
hcx_dll_reconfig interface		
dll_offset_ctrl_addnsub	Conduit	This DLL reconfiguration interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to enable DLL reconfiguration.
dll_offset_ctrl_offset		
dll_offset_ctrl_addnsub ⁽¹⁾		
dll_offset_ctrl_offset ⁽¹⁾		
dll_offset_ctrl_offsetctrlout ⁽¹⁾		
dll_offset_ctrl_b_offsetctrlout ⁽¹⁾		

Table 8-2. DDR3 SDRAM Controller with UniPHY Interfaces (Part 5 of 6)

Signals in Interface	Interface Type	Description/How to Connect
hcx_pll_reconfig interface		
configupdate	Conduit	This PLL reconfiguration interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to enable PLL reconfiguration.
phasecounterselect		
phasestep		
phaseupdown		
scanclk		
scanclkena		
scandata		
phasedone		
scandataout		
scandone		
hcx_rom_reconfig interface		
hc_rom_config_clock	Conduit	This ROM loader interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to control loading of the sequencer ROM.
hc_rom_conig_datain		
hc_rom_config_rom_data_ready		
hc_rom_config_init		
hc_rom_config_init_busy		
hc_rom_config_rom_rden		
hc_rom_config_rom_address		
autoprecharge_req interface		
local_autopch_req	Conduit	Precharge interface for connection to a custom control block. This interface is enabled only when you turn on Auto-precharge Control .
user_refresh interface		
local_refresh_req	Conduit	User refresh interface for connection to a custom control block. This interface is enabled only when you turn on User Auto-Refresh Control .
local_refresh_chip		
local_refresh_ack		
self_refresh interface		
local_self_rfsh_req	Conduit	Self refresh interface for connection to a custom control block. This interface is enabled only when you turn on Self-refresh Control .
local_self_rfsh_chip		
local_self_rfsh_ack		
ecc_interrupt interface		
ecc_interrupt	Conduit	ECC interrupt signal for connection to a custom control block. This interface is enabled only when you turn on Error Detection and Correction Logic .

Table 8–2. DDR3 SDRAM Controller with UniPHY Interfaces (Part 6 of 6)

Signals in Interface	Interface Type	Description/How to Connect
csr interface		
csr_write_req	Avalon-MM Slave	Configuration and status register signals for the memory interface, for connection to an Avalon_MM master. This interface is enabled only when you turn on Configuration and Status Register .
csr_read_req		
csr_waitrequest		
csr_addr		
csr_be		
csr_wdata		
csr_rdata		
csr_rdata_valid		
Note to Table 8–2 (1) Signals available only in DLL master mode.		

Table 8–3 lists the QDR II and QDR II+ SRAM signals available for each interface in Qsys and SOPC Builder and provides a description and guidance on how to connect those interfaces.

Table 8–3. QDR II and QDR II+ SRAM Controller with UniPHY Interfaces (Part 1 of 5) (Part 1 of 5)

Signals in Interface	Interface Type	Description/How to Connect
pll_ref_clk interface		
pll_ref_clk	Clock input	PLL reference clock input.
global_reset interface		
global_reset_n	Reset input	Asynchronous global reset for PLL and all logic in PHY.
soft_reset interface		
soft_reset_n	Reset input	Asynchronous reset input. Resets the PHY, but not the PLL that the PHY uses.
afi_reset interface		
afi_reset_n	Reset output (PLL master/no sharing)	When the interface is in PLL master or no sharing modes, this interface is an asynchronous reset output of the AFI interface. This interface is asserted when the PLL loses lock or the PHY is reset.
afi_reset_in interface		
afi_reset_n	Reset input (PLL slave)	When the interface is in PLL slave mode, this interface is a reset input that you must connect to the <i>afi_reset</i> output of an identically configured memory interface in PLL master mode.

Table 8-3. QDR II and QDR II+ SRAM Controller with UniPHY Interfaces (Part 2 of 5) (Part 2 of 5)

Signals in Interface	Interface Type	Description/How to Connect
afi_clk interface		
afi_clk	Clock output (PLL master/no sharing)	<p>This AFI interface clock can be full-rate or half-rate memory clock frequency based on the memory interface parameterization.</p> <p>When the interface is in PLL master or no sharing modes, this interface is a clock output.</p>
afi_clk_in interface		
afi_clk	Clock input (PLL slave)	<p>This AFI interface clock can be full-rate or half-rate memory clock frequency based on the memory interface parameterization.</p> <p>When the interface is in PLL slave mode, this is a clock input that you must connect to the <code>afi_clk</code> output of an identically configured memory interface in PLL master mode.</p>
afi_half_clk interface		
afi_half_clk	Clock output (PLL master/no sharing)	<p>The AFI half clock that is half the frequency of <code>afi_clk</code>.</p> <p>When the interface is in PLL master or no sharing modes, this interface is a clock output.</p>
afi_half_clk_in interface		
afi_half_clk	Clock input (PLL slave)	<p>The AFI half clock that is half the frequency of <code>afi_clk</code>.</p> <p>When the interface is in PLL slave mode, you must connect this <code>afi_half_clk</code> input to the <code>afi_half_clk</code> output of an identically configured memory interface in PLL master mode.</p>

Table 8-3. QDR II and QDR II+ SRAM Controller with UniPHY Interfaces (Part 3 of 5) (Part 3 of 5)

Signals in Interface	Interface Type	Description/How to Connect
memory interface		
mem_a	Conduit	Interface signals between the PHY and the memory device.
mem_cqn		
mem_bws_n		
mem_cq		
mem_d		
mem_k		
mem_k_n		
mem_q		
mem_wps_n		
mem_rps_n		
mem_doff_n		
avl_r interface		
avl_r_read_req	Avalon-MM Slave	Avalon-MM interface between memory interface and user logic for read requests.
avl_r_ready		
avl_r_addr		
avl_r_size		
avl_r_rdata_valid		
avl_r_rdata		
avl_w interface		
avl_w_write_req	Avalon-MM Slave	Avalon-MM interface between memory interface and user logic for write requests.
avl_w_ready		
avl_w_addr		
avl_w_size		
avl_w_wdata		
avl_w_be		
status interface		
local_init_done	Conduit	Memory interface status signals.
local_cal_success		
local_cal_fail		
oct interface		
rup (Stratix III/IV, Arria II GZ, Arria II GX)	Conduit	OCT reference resistor pins for rup/rdn or rzqin.
rdn (Stratix III/IV, Arria II GZ, Arria II GX)		
rzq (Stratix V)		

Table 8-3. QDR II and QDR II+ SRAM Controller with UniPHY Interfaces (Part 4 of 5) (Part 4 of 5)

Signals in Interface	Interface Type	Description/How to Connect
pll_sharing interface		
pll_mem_clk	Conduit	Interface signals for PLL sharing, to connect PLL masters to PLL slaves. This interface is enabled only when you set PLL sharing mode to master or slave.
pll_write_clk		
pll_addr_cmd_clk		
pll_locked		
pll_avl_clk		
pll_config_clk		
pll_hr_clk		
pll_p2c_read_clk		
pll_c2p_write_clk		
pll_dr_clk		
dll_sharing interface		
dll_delayctrl	Conduit	DLL sharing interface for connecting DLL masters to DLL slaves. This interface is enabled only when you set DLL sharing mode to master or slave.
oct_sharing interface		
seriesterminationcontrol (Stratix III/IV/V, Arria II GZ)	Conduit	OCT sharing interface for connecting OCT masters to OCT slaves. This interface is enabled only when you set OCT sharing mode to master or slave.
parallelerminationcontrol (Stratix III/IV/V, Arria II GZ)		
terminationcontrol (Arria II GX)		
hcx_dll_reconfig		
dll_offset_ctrl_addnsub	Conduit	This DLL reconfiguration interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to enable DLL reconfiguration.
dll_offset_ctrl_offset		
dll_offset_ctrl_addnsub ⁽¹⁾		
dll_offset_ctrl_offset ⁽¹⁾		
dll_offset_ctrl_offsetctrlout ⁽¹⁾		
dll_offset_ctrl_b_offsetctrlout ⁽¹⁾		
hcx_pll_reconfig		
configupdate	Conduit	This PLL reconfiguration interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to enable PLL reconfiguration.
phasecounterselect		
phasestep		
phaseupdown		
scanclk		
scanckena		
scandata		
phasedone		
scandataout		
scandone		

Table 8-3. QDR II and QDR II+ SRAM Controller with UniPHY Interfaces (Part 5 of 5) (Part 5 of 5)

Signals in Interface	Interface Type	Description/How to Connect
hcx_rom_reconfig		
hc_rom_config_clock	Conduit	This ROM loader interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to control loading of the sequencer ROM.
hc_rom_config_datain		
hc_rom_config_rom_data_ready		
hc_rom_config_init		
hc_rom_config_init_busy		
hc_rom_config_rom_rden		
hc_rom_config_rom_address		
Note to Table 8-3 (1) Signals available only in DLL master mode.		

Table 8-4 lists the RLDRAM II signals available for each interface in Qsys and SOPC Builder and provides a description and guidance on how to connect those interfaces.

Table 8-4. RLDRAM II Controller with UniPHY Interfaces (Part 1 of 5)

Interface Name	Interface Type	Description
pll_ref_clk interface		
pll_ref_clk	Clock input.	PLL reference clock input.
global_reset interface		
global_reset_n	Reset input	Asynchronous global reset for PLL and all logic in PHY.
soft_reset interface		
soft_reset_n	Reset input	Asynchronous reset input. Resets the PHY, but not the PLL that the PHY uses.
afi_reset interface		
afi_reset_n	Reset output (PLL master/no sharing)	When the interface is in PLL master or no sharing modes, this interface is an asynchronous reset output of the AFI interface. This interface is asserted when the PLL loses lock or the PHY is reset.
afi_reset_in interface		
afi_reset_n	Reset input (PLL slave)	When the interface is in PLL slave mode, this interface is a reset input that you must connect to the <i>afi_reset</i> output of an identically configured memory interface in PLL master mode.

Table 8-4. RLDRAM II Controller with UniPHY Interfaces (Part 2 of 5)

Interface Name	Interface Type	Description
afi_clk interface		
afi_clk	Clock output (PLL master/no sharing)	This AFI interface clock can be full-rate or half-rate memory clock frequency based on the memory interface parameterization. When the interface is in PLL master or no sharing modes, this interface is a clock output.
afi_clk_in interface		
afi_clk	Clock input (PLL slave)	This AFI interface clock can be full-rate or half-rate memory clock frequency based on the memory interface parameterization. When the interface is in PLL slave mode, you must connect this <code>afi_clk</code> input to the <code>afi_clk</code> output of an identically configured memory interface in PLL master mode.
afi_half_clk interface		
afi_half_clk	Clock output (PLL master/no sharing)	The AFI half clock that is half the frequency of the <code>afi_clk</code> . When the interface is in PLL master or no sharing modes, this interface is a clock output.
afi_half_clk_in interface		
afi_half_clk	Clock input (PLL slave)	The AFI half clock that is half the frequency of the <code>afi_clk</code> . When the interface is in PLL slave mode, you must connect this <code>afi_half_clk</code> input to the <code>afi_half_clk</code> output of an identically configured memory interface in PLL master mode.

Table 8-4. RLDRAM II Controller with UniPHY Interfaces (Part 3 of 5)

Interface Name	Interface Type	Description
memory interface		
mem_a	Conduit	Interface signals between the PHY and the memory device.
mem_ba		
mem_ck		
mem_ck_n		
mem_cs_n		
mem_dk		
mem_dk_n		
mem_dm		
mem_dq		
mem_qk		
mem_qk_n		
mem_ref_n		
mem_we_n		
avl interface		
avl_size	Avalom-MM Slave	Avalon-MM interface between memory interface and user logic.
avl_wdata		
avl_rdata_valid		
avl_rdata		
avl_ready		
avl_write_req		
avl_read_req		
avl_addr		
status interface		
local_init_done	Conduit	Memory interface status signals.
local_cal_success		
local_cal_fail		
oct interface		
rup (Stratix III/IV, Arria II GZ)	Conduit	OCT reference resistor pins for rup/rdn or rzqin.
rqn (Stratix III/IV, Arria II GZ)		
rzq (Stratix V)		

Table 8-4. RLDRAM II Controller with UniPHY Interfaces (Part 4 of 5)

Interface Name	Interface Type	Description
pll_sharing interface		
pll_mem_clk	Conduit	Interface signals for PLL sharing, to connect PLL masters to PLL slaves. This interface is enabled only when you set PLL sharing mode to master or slave.
pll_write_clk		
pll_addr_cmd_clk		
pll_locked		
pll_avl_clk		
pll_config_clk		
pll_hr_clk		
pll_p2c_read_clk		
pll_c2p_write_clk		
pll_dr_clk		
dll_sharing interface		
dll_delayctrl	Conduit	DLL sharing interface for connecting DLL masters to DLL slaves. This interface is enabled only when you set DLL sharing mode to master or slave.
oct_sharing interface		
seriesterminationcontrol	Conduit	OCT sharing interface for connecting OCT masters to OCT slaves. This interface is enabled only when you set OCT sharing mode to master or slave.
parallelerminationcontrol		
hcx_dll_reconfig interface		
dll_offset_ctrl_addnsub	Conduit	This DLL reconfiguration interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to enable DLL reconfiguration.
dll_offset_ctrl_offset		
dll_offset_ctrl_addnsub ⁽¹⁾		
dll_offset_ctrl_offset ⁽¹⁾		
dll_offset_ctrl_offsetctrlout ⁽¹⁾		
dll_offset_ctrl_b_offsetctrlout ⁽¹⁾		
hcx_pll_reconfig interface		
configupdate	Conduit	This PLL reconfiguration interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to enable PLL reconfiguration.
phasecounterselect		
phasestep		
phaseupdown		
scanclk		
scanckena		
scandata		
phasedone		
scandataout		
scandone		

Table 8-4. RLDRAM II Controller with UniPHY Interfaces (Part 5 of 5)

Interface Name	Interface Type	Description
hcx_rom_reconfig interface		
hc_rom_config_clock	Conduit	This ROM loader interface is enabled only when you turn on HardCopy Compatibility Mode . You can connect this interface to user-created custom logic to control loading of the sequencer ROM.
hc_rom_config_datain		
hc_rom_config_rom_data_ready		
hc_rom_config_init		
hc_rom_config_init_busy		
hc_rom_config_rom_rden		
hc_rom_config_rom_adress		
parity_error_interrupt interface		
parity_error	Conduit	Parity error interrupt conduit for connection to custom control block. This interface is enabled only if you turn on Enable Error Detection Parity .
user_refresh interface		
ref_req	Conduit	User refresh interface for connection to custom control block. This interface is enabled only if you turn on Enable User Refresh .
ref_ba		
ref_ack		
reserved interface		
reserved	Conduit	Reserved interface required for certain pin configurations when you select the Nios® II-based sequencer.
<p>Note to Table 8-4</p> <p>(1) Signals available only in DLL master mode.</p>		

Generated Files

When you complete the IP generation flow, there are generated files created in your project directory. The directory structure created varies somewhat, depending on the tool used to parameterize and generate the IP.



The PLL parameters are statically defined in the `<variation_name>_parameters.tcl` at generation time. To ensure timing constraints and timing reports are correct, when you edit the PLL parameters, apply those changes to the PLL parameters in this file.

The following sections list the generated files for the ALTMEMPHY and UniPHY IP.

Generated Files for Memory Controllers with the ALTMEMPHY IP

Table 8–5 lists the ALTMEMPHY generated directory and key files using the MegaWizard Plug-In Manager.

Table 8–5. ALTMEMPHY Generated Files (Part 1 of 2)

File Name	Description
alt_mem_phy_defines.v	Contains constants used in the interface. This file is always in Verilog HDL regardless of the language you chose in the MegaWizard Plug-In Manager.
<variation_name>.ppf	Pin planner file for your ALTMEMPHY variation.
<variation_name>.qip	Quartus II IP file for your ALTMEMPHY variation, containing the files associated with this megafunction.
<variation_name>.v/.vhd	Top-level file of your ALTMEMPHY variation, generated based on the language you chose in the MegaWizard Plug-In Manager.
<variation_name>.vho	Contains functional simulation model for VHDL only.
<variation_name>_alt_mem_phy_seq_wrapper.vo/.vho	A wrapper file, for simulation only, that calls the sequencer file, created based on the language you chose in the MegaWizard Plug-In Manager.
<variation_name>.html	Lists the top-level files created and ports used in the megafunction.
<variation_name>_alt_mem_phy_seq_wrapper.v/.vhd	A wrapper file, for compilation only, that calls the sequencer file, created based on the language you chose in the MegaWizard Plug-In Manager.
<variation_name>_alt_mem_phy_seq.vhd	Contains the sequencer used during calibration. This file is always in VHDL language regardless of the language you chose in the MegaWizard Plug-In Manager.
<variation_name>_alt_mem_phy.v	Contains all modules of the ALTMEMPHY variation except for the sequencer. This file is always in Verilog HDL language regardless of the language you chose in the MegaWizard Plug-In Manager. The <variation_name>_alt_mem_phy_seq.vhd includes the DDR3 SDRAM sequencer.
<variation_name>_alt_mem_phy_pll_<device>.ppf	This XML file describes the MegaCore pin attributes to the Quartus II Pin Planner.
<variation_name>_alt_mem_phy_pll.v/.vhd	The PLL megafunction file for your ALTMEMPHY variation, generated based on the language you chose in the MegaWizard Plug-In Manager.
<variation_name>_alt_mem_phy_delay.vhd	Includes a delay module for simulation. This file is only generated if you choose VHDL as the language of your MegaWizard Plug-In Manager output files.
<variation_name>_alt_mem_phy_dq_dqs.vhd or .v	Generated file that contains DQ/DQS I/O atoms interconnects and instance. Only generated when targeting Arria II GX devices.

Table 8–5. ALTMEMPHY Generated Files (Part 2 of 2)

File Name	Description
<code><variation_name>_alt_mem_phy_dq_dqs_clearbox.txt</code>	Specification file that generates the <code><variation_name>_alt_mem_phy_dq_dqs</code> file using the clearbox flow. Only generated when targeting Arria II GX devices.
<code><variation_name>_alt_mem_phy_pll.qip</code>	Quartus II IP file for the PLL that your ALTMEMPHY variation uses that contains the files associated with this megafunction.
<code><variation_name>_alt_mem_phy_pll_bb.v/.cmp</code>	Black box file for the PLL used in your ALTMEMPHY variation. Typically unused.
<code><variation_name>_alt_mem_phy_reconfig.qip</code>	Quartus II IP file for the PLL reconfiguration block. Only generated when targeting Arria GX, HardCopy® II, Stratix II, and Stratix II GX devices.
<code><variation_name>_alt_mem_phy_reconfig.v/.vhd</code>	PLL reconfiguration block module. Only generated when targeting Arria GX, HardCopy II, Stratix II, and Stratix II GX devices.
<code><variation_name>_alt_mem_phy_reconfig_bb.v/.cmp</code>	Black box file for the PLL reconfiguration block. Only generated when targeting Arria GX, HardCopy II, Stratix II, and Stratix II GX devices.
<code><variation_name>_bb.v/.cmp</code>	Black box file for your ALTMEMPHY variation, depending whether you are using Verilog HDL or VHDL language.
<code><variation_name>_ddr_pins.tcl</code>	Contains procedures used in the <code><variation_name>_ddr_timing.sdc</code> and <code><variation_name>_report_timing.tcl</code> files.
<code><variation_name>_pin_assignments.tcl</code>	Contains I/O standard, drive strength, output enable grouping, DQ/DQS grouping, and termination assignments for your ALTMEMPHY variation. If your top-level design pin names do not match the default pin names or a prefixed version, edit the assignments in this file.
<code><variation_name>_ddr_timing.sdc</code>	Contains timing constraints for your ALTMEMPHY variation.
<code><variation_name>_report_timing.tcl</code>	Script that reports timing for your ALTMEMPHY variation during compilation.

Table 8–6 lists the modules that are instantiated in the `<variation_name>_alt_mem_phy.v/.vhd` file. A particular ALTMEMPHY variation may or may not use any of the modules, depending on the memory standard that you specify.

Table 8–6. Modules in `<variation_name>_alt_mem_phy.v` File (Part 1 of 2)

Module Name	Usage	Description
<code><variation_name>_alt_mem_phy_addr_cmd</code>	All ALTMEMPHY variations	Generates the address and command structures.
<code><variation_name>_alt_mem_phy_clk_reset</code>	All ALTMEMPHY variations	Instantiates PLL, DLL, and reset logic.

Table 8-6. Modules in <variation_name>_alt_mem_phy.v File (Part 2 of 2)

Module Name	Usage	Description
<variation_name>_alt_mem_phy_dp_io	All ALTMEMPHY variations	Generates the DQ, DQS, DM, and QVLD I/O pins.
<variation_name>_alt_mem_phy_mimic	DDR2/DDR SDRAM ALTMEMPHY variation	Creates the VT tracking mechanism for DDR and DDR2 SDRAM PHY IPs.
<variation_name>_alt_mem_phy_oct_delay	DDR2/DDR SDRAM ALTMEMPHY variation when dynamic OCT is enabled.	Generates the proper delay and duration for the OCT signals.
<variation_name>_alt_mem_phy_postamble	DDR2/DDR SDRAM ALTMEMPHY variations	Generates the postamble enable and disable scheme for DDR and DDR2 SDRAM PHY IPs.
<variation_name>_alt_mem_phy_read_dp	All ALTMEMPHY variations (unused for Stratix III or Stratix IV devices)	Takes read data from the I/O through a read path FIFO buffer, to transition from the resynchronization clock to the PHY clock.
<variation_name>_alt_mem_phy_read_dp_group	DDR2/DDR SDRAM ALTMEMPHY variations (Stratix III and Stratix IV devices only)	A per DQS group version of <variation_name>_alt_mem_phy_read_dp.
<variation_name>_alt_mem_phy_readata_valid	DDR2/DDR SDRAM ALTMEMPHY variations	Generates read data valid signal to sequencer and controller.
<variation_name>_alt_mem_phy_seq_wrapper	All ALTMEMPHY variations	Generates sequencer for DDR and DDR2 SDRAM.
<variation_name>_alt_mem_phy_write_dp	All ALTMEMPHY variations	Generates the demultiplexing of data from half-rate to full-rate DDR data.
<variation_name>_alt_mem_phy_write_dp_fr	DDR2/DDR SDRAM ALTMEMPHY variations	A full-rate version of <variation_name>_alt_mem_phy_write_dp.

Table 8-7 lists the additional files generated by the HPC II that may be in your project directory.

Table 8-7. Controller-Generated Files (Part 1 of 2)

Filename	Description
alt_mem_ddrx_addr_cmd.v	Decodes internal protocol-related signals into memory address and command signals.
alt_mem_ddrx_addr_cmd_wrap.v	A wrapper that instantiates the alt_mem_ddrx_addr_cmd.v file.
alt_mem_ddrx_ddr2_odt_gen.v	Generates the on-die termination (ODT) control signal for DDR2 memory interfaces.
alt_mem_ddrx_ddr3_odt_gen.v	Generates the ODT control signal for DDR3 memory interfaces.
alt_mem_ddrx_odt_gen.v	Wrapper that instantiates alt_mem_ddrx_ddr2_odt_gen.v and alt_mem_ddrx_ddr3_odt_gen.v. This file also controls the ODT addressing scheme.
alt_mem_ddrx_rdwr_data_tmng.v	Decodes internal data burst related signals to memory data signals.
alt_mem_ddrx_arbiter.v	Contains logic that determines which command to execute based on certain schemes.
alt_mem_ddrx_burst_gen.v	Converts internal DRAM-aware commands to AFI signals.
alt_mem_ddrx_cmd_gen.v	Converts user requests to DRAM-aware commands.

Table 8–7. Controller-Generated Files (Part 2 of 2)

Filename	Description
<code>alt_mem_ddrx_csr.v</code>	Contains configuration registers.
<code>alt_mem_ddrx_buffer.v</code>	Contains buffer for local data.
<code>alt_mem_ddrx_buffer_manager.v</code>	Manages the allocation of buffers.
<code>alt_mem_ddrx_burst_tracking.v</code>	Tracks data received per local burst command.
<code>alt_mem_ddrx_dataid_manager.v</code>	Manages the IDs associated with data stored in buffer.
<code>alt_mem_ddrx_fifo.v</code>	Contains the FIFO buffer to store local data to create a link; is also used in <code>rdata_path</code> to store the read address and error address.
<code>alt_mem_ddrx_list.v</code>	Tracks the DRAM commands associated with the data stored internally.
<code>alt_mem_ddrx_rdata_path.v</code>	Contains read data path logic.
<code>alt_mem_ddrx_wdata_path.v</code>	Contains write data path logic.
<code>alt_mem_ddrx_define.v</code>	Defines common parameters used in the RTL files.
<code>alt_mem_ddrx_ecc_decoder.v</code>	Instantiates appropriate width of the ECC decoder logic.
<code>alt_mem_ddrx_ecc_decoder_32_syn.v</code>	Contains synthesizable 32-bit version of the ECC decoder.
<code>alt_mem_ddrx_ecc_decoder_64_syn.v</code>	Contains synthesizable 64-bit version of the ECC decoder.
<code>alt_mem_ddrx_ecc_encoder.v</code>	Instantiates appropriate width of the ECC encoder logic.
<code>alt_mem_ddrx_ecc_encoder_32_syn.v</code>	Contains synthesizable 32-bit version of the ECC decoder.
<code>alt_mem_ddrx_ecc_encoder_64_syn.v</code>	Contains synthesizable 64-bit version of the ECC decoder.
<code>alt_mem_ddrx_ecc_encoder_decoder_wrapper.v</code>	Wrapper that instantiates all ECC logic.
<code>alt_mem_ddrx_input_if.v</code>	Contains local input interface logic.
<code>alt_mem_ddrx_mm_st_converter.v</code>	Contains supporting logic for Avalon-MM interface.
<code>alt_mem_ddrx_rank_timer.v</code>	Contains a timer associated with rank timing.
<code>alt_mem_ddrx_sideband.v</code>	Contains supporting logic for user-controlled refresh and precharge signals.
<code>alt_mem_ddrx_tbp.v</code>	Contains command queue and associated logic for reordering features.
<code>alt_mem_ddrx_timing_param.v</code>	Contains timer logic associated with non-rank timing.
<code>alt_mem_ddrx_controller_st_top.v</code>	Wrapper that instantiates all submodules and configuration registers.
<code>alt_mem_ddrx_controller_top.v</code>	Wrapper that contains memory controller with Avalon-MM interface.
<code>alt_mem_ddrx_controller.v</code>	Wrapper that instantiates all submodules.

Generated Files for Memory Controllers with the UniPHY IP

Table 8–8 lists the generated directory structure and key files created with the MegaWizard Plug-In Manager, SOPC Builder, and Qsys.

Table 8–8. Generated Directory Structure and Key Files (Part 1 of 4)

	Directory	File Name	Description
MegaWizard Plug-In Manager			
Synthesis Files	<working_dir>/	<variation_name>.qip	Quartus II IP file which refers to all generated files in the synthesis fileset. Include this file in your Quartus II project.
	<working_dir>/	<variation_name>.v or <variation_name>.vhd	Top-level wrapper synthesis files. .v is IEEE Encrypted Verilog. .vhd is generated VHDL.
	<working_dir>/<variation_name>/	<variation_name>_0002.v	UniPHY top-level wrapper.
	<working_dir>/<variation_name>/	*.v, *.sv, *.tcl, *.sdc, *.ppf	RTL and constraints files for synthesis.
	<working_dir>/<variation_name>/	<variation_name>_p0_pin_assignments.tcl	Pin constraints script to be run after synthesis.
Simulation Files	<working_dir>/<variation_name>_sim /	<variation_name>.v	Top-level wrapper simulation files for both Verilog and VHDL.
	<working_dir>/<variation_name>_sim /<subcomponent_module>/	*.v, *.sv, *.vhd, *.vho, *.hex, *.mif	RTL and constraints files for simulation. .v and .sv files are IEEE Encrypted Verilog. .vhd and .vho are generated VHDL.

Table 8–8. Generated Directory Structure and Key Files (Part 2 of 4)

Directory		File Name	Description
MegaWizard Plug-In Manager—Example Design Fileset			
Synthesis Files	<variation_name>_example_design/example_project/	<variation_name>_example.qip	Quartus II IP file that refers to all generated files in the synthesizable project.
	<variation_name>_example_design/example_project/	<variation_name>_example.qpf	Quartus II project for synthesis flow.
	<variation_name>_example_design/example_project/	<variation_name>_example.qsf	Quartus II project for synthesis flow.
	<variation_name>_example_design/example_project/ <variation_name>_example/	<variation_name>_example.v	Top-level wrapper.
	<variation_name>_example_design/example_project/ <variation_name>_example/ submodules/	*.v, *.sv, *.tcl, *.sdc, *.ppf	RTL and constraints files.
	<variation_name>_example_design/example_project/ <variation_name>_example/ submodules/	<variation_name>_example_if0_p0_pin_assignments.tcl	Pin constraints script to be run after synthesis. _if0 and _p0 are instance names. For more information, refer to Table 8–9 on page 8–37 .

Table 8–8. Generated Directory Structure and Key Files (Part 3 of 4)

	Directory	File Name	Description
Simulation Files	<variation_name>_example_design/simulation/	generate_sim_verilog_example_design.tcl	Run this file to generate the Verilog simulation example design.
	<variation_name>_example_design/simulation/	generate_sim_vhdl_example_design.tcl	Run this file to generate the VHDL simulation example design.
	<variation_name>_example_design/simulation/	README.txt	A text file with instructions about how to generate and run the simulation example design.
	<variation_name>_example_design/simulation/verilog/mentor	run.do	ModelSim script to simulate the generated Verilog example design.
	<variation_name>_example_design/simulation/vhdl/mentor	run.do	ModelSim script to simulate the generated VHDL example design.
	<variation_name>_example_design/simulation/verilog/ <variation_name>_sim/	<variation_name>_example_sim.v	Top-level wrapper (Testbench) for Verilog.
	<variation_name>_example_design/simulation/vhdl/ <variation_name>_sim/	<variation_name>_example_sim.vhd	Top-level wrapper (Testbench) for VHDL.
	<variation_name>_example_design/simulation/ <variation_name>_sim/verilog/ submodules/	*.v, *.sv, *.hex, *.mif	RTL and ROM data for Verilog.
	<variation_name>_example_design/simulation/ <variation_name>_sim/vhdl/ submodules/	*.vhd, *.vho, *.hex, *.mif	RTL and ROM data for VHDL.
SOPC Builder			
	<working_dir>/	<system_name>.qip	Quartus II IP file that refers to all the generated files in the SOPC Builder project.
	<working_dir>/	<system_name>.v	System top-level RTL.
	<working_dir>/	<module_name>.v	Module wrapper RTL.
	<working_dir>/<module_name>/	*.v, *.sv, *.tcl, *.sdc, *.ppf	Subdirectory of TL and constraints for each system module.
Qsys			
	<working_dir>/<system_name>/synthesis/	<system_name>.qip	Quartus II IP file that refers to all the generated files in the synthesis filesset.
	<working_dir>/<system_name>/synthesis/	<system_name>.v	System top-level RTL for synthesis.

Table 8–8. Generated Directory Structure and Key Files (Part 4 of 4)

Directory	File Name	Description
<working_dir>/<system_name>/simulation/	<system_name>.v or <variation_name>.vhd	System top-level RTL for simulation. .v file is IEEE Encrypted Verilog. .vhd file is generated VHDL.
<working_dir>/<system_name>/synthesis/ submodules/	*.v, *.sv, *.tcl, *.sdc, *.ppf	RTL and constraints files for synthesis.
<working_dir>/<system_name>/simulation/ submodules/	*.v, *.sv, *.hex, *.mif	RTL and ROM data for simulation.

Table 8–9 lists the prefixes or instance names of submodule files within the memory interface IP. These instances are concatenated to form unique synthesis and simulation filenames.

Table 8–9. Prefixes of Submodule Files

Prefixes	Description
_c0	Specifies the controller.
_d0	Specifies the driver or traffic generator.
_dll0	Specifies the DLL.
_e0	Specifies the example design.
_if0	Specifies the memory Interface.
_m0	Specifies the AFI mux.
_oct0	Specifies the OCT.
_p0	Specifies the PHY.
_pll0	Specifies the PLL.
_s0	Specifies the sequencer.
_t0	Specifies the traffic generator status checker.

Parameterizing Memory Controllers with ALTMEMPHY IP

This section describes the parameters you can set for the DDR, DDR2, and DDR3 SDRAM memory controllers with the ALTMEMPHY IP.

The **Parameter Settings** page in the ALTMEMPHY parameter editor allows you to parameterize the following settings:

- Memory Settings
- PHY Settings
- Board Settings

The text window at the bottom of the MegaWizard Plug-In Manager displays information about the memory interface, warnings, and errors if you are trying to create something that is not supported. The **Finish** button is disabled until you correct all the errors indicated in this window.

The following sections describe the four tabs of the **Parameter Settings** page in more detail.

Memory Settings

Use this tab to apply the memory parameters from your memory manufacturer's data sheet.


[Table 8–10](#) describes the **General Settings** available on the **Memory Settings** page of the ALTMEMPHY parameter editor.

Table 8–10. General Settings (Part 1 of 2)

Parameter Name	Description
Device family	Targets device family (for example, Arria II GX). The device family selected here must match the device family selected on page 2a of the parameter editor. For more information about selecting a device family, refer to the "Device Family Selection" section in the <i>Selecting your FPGA Device</i> chapter of the <i>External Memory Interface Handbook</i> .
Speed grade	Selects a particular speed grade of the device (for example, 2, 3, or 4 for the Arria II GX device family).
PLL reference clock frequency	Determines the clock frequency of the external input clock to the PLL. Ensure that you use three decimal points if the frequency is not a round number (for example, 166.667 MHz or 100 MHz) to avoid a functional simulation or a PLL locking problem.
Memory clock frequency	Determines the memory interface clock frequency. If you are operating a memory device below its maximum achievable frequency, ensure that you enter the actual frequency of operation rather than the maximum frequency achievable by the memory device. Also, ensure that you use three decimal points if the frequency is not a round number (for example, 333.333 MHz or 400 MHz) to avoid a functional simulation or a PLL locking issue.
Controller data rate	Selects the data rate for the memory controller. Sets the frequency of the controller to equal to either the memory interface frequency (full-rate) or half of the memory interface frequency (half-rate). The full-rate option is not available for DDR3 SDRAM devices.
Enable half rate bridge	This option is only available for HPC II full-rate controller. Turn on to keep the controller in the memory full clock domain while allowing the local side to run at half the memory clock speed, so that latency can be reduced.

Table 8-10. General Settings (Part 2 of 2)

Parameter Name	Description
Local interface clock frequency	Value that depends on the memory clock frequency and controller data rate.
Local interface width	Value that depends on the memory clock frequency and controller data rate.

 When targeting a HardCopy device migration with performance improvement, the ALTMEMPHY IP should target the mid speed grade to ensure that the PLL and the PHY sequencer settings match. The compilation of the design can be executed in the faster speed grade.

Show in ‘Memory Preset’ List

Table 8-11 describes the options available to filter the **Memory Presets** that are displayed. This set of options is where you indicate whether you are creating a datapath for DDR3 SDRAM.


Table 8-11. Show in ‘Memory Presets’ List

Parameter Name	Description
Memory type	You can filter the type of memory to display, for example, DDR3 SDRAM.
Memory vendor	You can filter the memory types by vendor. JEDEC is also one of the options, allowing you to choose the JEDEC specifications. If your chosen vendor is not listed, you can choose JEDEC for the DDR3 SDRAM interfaces. Then, pick a device that has similar specifications to your chosen device and check the values of each parameter. Make sure you change the each parameter value to match your device specifications.
Memory format	You can filter the type of memory by format, for example, discrete devices or DIMM packages.
Maximum frequency	You can filter the type of memory by the maximum operating frequency.

Memory Presets

Pick a device in the **Memory Presets** list that is closest or the same as the actual memory device that you are using. Then, click the **Modify Parameters** button to parameterize the following settings in the **Preset Editor** dialog box:

- Memory attributes—These are the settings that determine your system's number of DQ, DQ strobe (DQS), address, and memory clock pins.
- Memory initialization options—These settings are stored in the memory mode registers as part of the initialization process.
- Memory timing parameters—These are the parameters that create and time-constrain the PHY.

 Even though the device you are using is listed in **Memory Presets**, ensure that the settings in the **Preset Editor** dialog box are accurate, as some parameters may have been updated in the memory device datasheets.

You can change the parameters with a white background to reflect your system. You can also change the parameters with a gray background so the device parameters match the device you are using. These parameters in gray background are characteristics of the chosen memory device and changing them creates a new custom memory preset. If you click **Save As** (at the bottom left of the page) and save the new settings in the `<quartus_install_dir>\quartus\common\ip\altera\altmemphy\lib\` directory, you can use this new memory preset in other Quartus II projects created in the same version of the software.

When you click **Save**, the new memory preset appears at the bottom of the **Memory Presets** list in the **Memory Settings** tab.



If you save the new settings in a directory other than the default directory, click **Load Preset** in the **Memory Settings** tab to load the settings into the **Memory Presets** list.

The **Advanced** option shows the percentage of memory specification that is calibrated by the FPGA. The percentage values are estimated by Altera based on the process variation.

Preset Editor Settings for DDR and DDR2 SDRAM

Table 8-12 through Table 8-14 describe the DDR2 SDRAM parameters available for memory attributes, initialization options, and timing parameters. DDR SDRAM has the same parameters, but their value ranges are different than DDR2 SDRAM.

Table 8-12. DDR2 SDRAM Attributes Settings (Part 1 of 2)

Parameter Name	Range ⁽¹⁾	Units	Description
Output clock pairs from FPGA	1–6	pairs	Defines the number of differential clock pairs driven from the FPGA to the memory. More clock pairs reduce the loading of each output when interfacing with multiple devices. Memory clock pins use the signal splitter feature in Arria II GX, Stratix III, and Stratix IV devices for differential signaling.
Total Memory chip selects	1, 2, 4, or 8	bits	Sets the number of chip selects in your memory interface. The number of chip selects defines the depth of your memory. You are limited to the range shown as the local side binary encodes the chip select address. You can set this value to the next higher number if the range does not meet your specifications. However, the highest address space of the ALTMEMPHY megafunction is not mapped to any of the actual memory address. The ALTMEMPHY megafunction works with multiple chip selects and calibrates against all chip select, <code>mem_cs_n</code> signals.
Memory interface DQ width	4–288	bits	Defines the total number of DQ pins on the memory interface. If you are interfacing with multiple devices, multiply the number of devices with the number of DQ pins per device. Even though the GUI allows you to choose 288-bit DQ width, the interface data width is limited by the number of pins on the device. For best performance, have the whole interface on one side of the device.

Table 8–12. DDR2 SDRAM Attributes Settings (Part 2 of 2)

Parameter Name	Range ⁽¹⁾	Units	Description
Memory vendor	JEDEC, Micron, Qimonda, Samsung, Hynix, Elpida, Nanya, other	—	Lists the name of the memory vendor for all supported memory standards.
Memory format	Discrete Device, Unbuffered DIMM, Registered DIMM	—	Specifies whether you are interfacing with devices or modules. SODIMM is supported under unbuffered or registered DIMMs.
Maximum memory frequency	See the memory device datasheet	MHz	Sets the maximum frequency supported by the memory.
Column address width	9–11	bits	Defines the number of column address bits for your interface.
Row address width	13–16	bits	Defines the number of row address bits for your interface.
Bank address width	2 or 3	bits	Defines the number of bank address bits for your interface.
Chip selects per DIMM	1 or 2	bits	Defines the number of chip selects on each DIMM in your interface.
DQ bits per DQS bit	4 or 8	bits	Defines the number of data (DQ) bits for each data strobe (DQS) pin.
Precharge address bit	8 or 10	bits	Selects the bit of the address bus to use as the precharge address bit.
Drive DM pins from FPGA	Yes or No	—	Specifies whether you are using DM pins for write operation. Altera devices do not support DM pins in ×4 mode.
Maximum memory frequency for CAS latency 3.0	80–533	MHz	Specifies the frequency limits from the memory data sheet per given CAS latency. The ALTMEMPHY parameter editor generates a warning if the operating frequency with your chosen CAS latency exceeds this number.
Maximum memory frequency for CAS latency 4.0			
Maximum memory frequency for CAS latency 5.0			
Maximum memory frequency for CAS latency 6.0			

Note to Table 8–12:

(1) The range values depend on the actual memory device used.

Table 8-13. DDR2 SDRAM Initialization Options

Parameter Name	Range	Units	Description
Memory burst length	4 or 8	beats	Sets the number of words read or written per transaction. Memory burst length of four equates to local burst length of one in half-rate designs and to local burst length of two in full-rate designs.
Memory burst ordering	Sequential or Interleaved	—	Controls the order in which data is transferred between memory and the FPGA during a read transaction. For more information, refer to the memory device datasheet.
Enable the DLL in the memory devices	Yes or No	—	Enables the DLL in the memory device when set to Yes . You must always enable the DLL in the memory device as Altera does not guarantee any ALTMEMPHY operation when the DLL is turned off. All timings from the memory devices are invalid when the DLL is turned off.
Memory drive strength setting	Normal or Reduced	—	Controls the drive strength of the memory device's output buffers. Reduced drive strength is not supported on all memory devices. The default option is normal.
Memory ODT setting	Disabled, 50, 75, 150	W	Sets the memory ODT value. Not available in DDR SDRAM interfaces.
Memory CAS latency setting	3, 4, 5, 6	cycles	Sets the delay in clock cycles from the read command to the first output data from the memory.

Table 8-14. DDR2 SDRAM Timing Parameter Settings ⁽¹⁾ (Part 1 of 3)

Parameter Name	Range	Units	Description
t_{INIT}	0.001–1000	μ s	Minimum memory initialization time. After reset, the controller does not issue any commands to the memory during this period.
t_{MRD}	2–39	ns	Minimum load mode register command period. The controller waits for this period of time after issuing a load mode register command before issuing any other commands. t_{MRD} is specified in ns in the DDR2 SDRAM high-performance controller and in terms of t_{CK} cycles in Micron's device datasheet. Convert t_{MRD} to ns by multiplying the number of cycles specified in the datasheet times t_{CK} , where t_{CK} is the memory operation frequency and not the memory device's t_{CK} .
t_{RAS}	8–200	ns	Minimum active to precharge time. The controller waits for this period of time after issuing an active command before issuing a precharge command to the same bank.
t_{RCD}	4–65	ns	Minimum active to read-write time. The controller does not issue read or write commands to a bank during this period of time after issuing an active command.
t_{RP}	4–65	ns	Minimum precharge command period. The controller does not access the bank for this period of time after issuing a precharge command.
t_{REFI}	1–65534	μ s	Maximum interval between refresh commands. The controller performs regular refresh at this interval unless user-controlled refresh is turned on.

Table 8-14. DDR2 SDRAM Timing Parameter Settings ⁽¹⁾ (Part 2 of 3)

Parameter Name	Range	Units	Description
t_{RFC}	14–1651	ns	Minimum autorefresh command period. The length of time the controller waits before doing anything else after issuing an auto-refresh command.
t_{WR}	4–65	ns	Minimum write recovery time. The controller waits for this period of time after the end of a write transaction before issuing a precharge command.
t_{WTR}	1–3	t_{CK}	Minimum write-to-read command delay. The controller waits for this period of time after the end of a write command before issuing a subsequent read command to the same bank. This timing parameter is specified in clock cycles and the value is rounded off to the next integer.
t_{AC}	300–750	ps	DQ output access time from CK/CK# signals.
t_{DQSCK}	100–750	ps	DQS output access time from CK/CK# signals.
t_{DQSQ}	100–500	ps	The maximum DQS to DQ skew; DQS to last DQ valid, per group, per access.
t_{DQSS}	0–0.3	t_{CK}	Positive DQS latching edge to associated clock edge.
t_{DS}	10–600	ps	DQ and DM input setup time relative to DQS, which has a derated value depending on the slew rate of the DQS (for both DDR and DDR2 SDRAM interfaces) and whether DQS is single-ended or differential (for DDR2 SDRAM interfaces). Ensure that you are using the correct number and that the value entered is referenced to $V_{REF}(dc)$, not $V_{IH}(ac)$ min or $V_{IL}(ac)$ max. Refer to “Derating Memory Setup and Hold Timing” on page 8–50 for more information about how to derate this specification.
t_{DH}	10–600	ps	DQ and DM input hold time relative to DQS, which has a derated value depending on the slew rate of the DQS (for both DDR and DDR2 SDRAM interfaces) and whether DQS is single-ended or differential (for DDR2 SDRAM interfaces). Ensure that you are using the correct number and that the value entered is referenced to $V_{REF}(dc)$, not $V_{IH}(dc)$ min or $V_{IL}(dc)$ max. Refer to “Derating Memory Setup and Hold Timing” on page 8–50 for more information about how to derate this specification.
t_{DSH}	0.1–0.5	t_{CK}	DQS falling edge hold time from CK.
t_{DSS}	0.1–0.5	t_{CK}	DQS falling edge to CK setup.
t_{IH}	100–1000	ps	Address and control input hold time, which has a derated value depending on the slew rate of the CK and CK# clocks and the address and command signals. Ensure that you are using the correct number and that the value entered is referenced to $V_{REF}(dc)$, not $V_{IH}(dc)$ min or $V_{IL}(dc)$ max. Refer to “Derating Memory Setup and Hold Timing” on page 8–50 for more information about how to derate this specification.

Table 8-14. DDR2 SDRAM Timing Parameter Settings ⁽¹⁾ (Part 3 of 3)

Parameter Name	Range	Units	Description
t_{IS}	100–1000	ps	Address and control input setup time, which has a derated value depending on the slew rate of the CK and CK# clocks and the address and command signals. Ensure that you are using the correct number and that the value entered is referenced to $V_{REF}(dc)$, not $V_{IH}(ac)$ min or $V_{IL}(ac)$ max. Refer to “Derating Memory Setup and Hold Timing” on page 8-50 for more information about how to derate this specification.
t_{QHS}	100–700	ps	The maximum data hold skew factor.
t_{RRD}	2.06–64	ns	The activate command to activate time, per device, RAS to RAS delay timing parameter.
t_{FAW}	7.69–256	ns	The four-activate window time, per device.
t_{RTP}	2.06–64	ns	Read to precharge time.

Note to Table 8-14:

- (1) Refer to the memory device data sheet for the parameter range. Some of the parameters are listed in a clock cycle (t_{CK}) unit. If the MegaWizard Plug-In Manager requires you to enter the value in a time unit (ps or ns), convert the number by multiplying it with the clock period of your interface (and not the maximum clock period listed in the memory data sheet).

Preset Editor Settings for DDR3 SDRAM

Table 8-15 through Table 8-17 describe the DDR3 SDRAM parameters available for memory attributes, initialization options, and timing parameters.

Table 8-15. DDR3 SDRAM Attributes Settings (Part 1 of 2)

Parameter Name	Range ⁽¹⁾	Units	Description
Output clock pairs from FPGA	1-6	pairs	Defines the number of differential clock pairs driven from the FPGA to the memory. Memory clock pins use the signal splitter feature in Arria II GX devices for differential signaling. The ALTMEMPHY parameter editor displays an error on the bottom of the window if you choose more than one for DDR3 SDRAM interfaces.
Total Memory chip selects	1, 2, 4, or 8	bits	Sets the number of chip selects in your memory interface. The number of chip selects defines the depth of your memory. You are limited to the range shown as the local side binary encodes the chip select address.
Memory interface DQ width	4-288	bits	Defines the total number of DQ pins on the memory interface. If you are interfacing with multiple devices, multiply the number of devices with the number of DQ pins per device. Even though the GUI allows you to choose 288-bit DQ width, DDR3 SDRAM variations are only supported up to 80-bit width due to restrictions in the board layout which affects timing at higher data width. Furthermore, the interface data width is limited by the number of pins on the device. For best performance, have the whole interface on one side of the device.
Mirror addressing	—	—	On multiple rank DDR3 SDRAM DIMMs address signals are routed differently to each rank; referred to in the JEDEC specification as address mirroring. Enter ranks with mirrored addresses in this field. There is one bit per chip select. For example, for four chip selects, enter 1011 to mirror the address on chip select #3, #1, and #0.
Memory vendor	Elpida, JEDEC, Micron, Samsung, Hynix, Nanya, other	—	Lists the name of the memory vendor for all supported memory standards.
Memory format	Discrete Device	—	Arria II GX devices only support DDR3 SDRAM components without leveling, for example, Discrete Device memory format.
Maximum memory frequency	See the memory device datasheet	MHz	Sets the maximum frequency supported by the memory.
Column address width	10-12	bits	Defines the number of column address bits for your interface.
Row address width	12-16	bits	Defines the number of row address bits for your interface. If your DDR3 SDRAM device's row address bus is 12-bit wide, set the row address width to 13 and set the 13 th bit to logic-level low (or leave the 13 th bit unconnected to the memory device) in the top-level file.

Table 8-15. DDR3 SDRAM Attributes Settings (Part 2 of 2)

Parameter Name	Range ⁽¹⁾	Units	Description
Bank address width	3	bits	Defines the number of bank address bits for your interface.
Chip selects per device	1 or 2	bits	Defines the number of chip selects on each device in your interface. Currently, calibration is done with all ranks but you can only perform timing analysis with one.
DQ bits per DQS bit	4 or 8	bits	Defines the number of data (DQ) bits for each data strobe (DQS) pin.
Drive DM pins from FPGA	Yes or No	—	Specifies whether you are using DM pins for write operation. Altera devices do not support DM pins with ×4 mode.
Maximum memory frequency for CAS latency 5.0	80–700	MHz	Specifies the frequency limits from the memory data sheet per given CAS latency. The ALTMEMPHY MegaWizard Plug-In Manager generates a warning if the operating frequency with your chosen CAS latency exceeds this number. The lowest frequency supported by DDR3 SDRAM devices is 300 MHz.
Maximum memory frequency for CAS latency 6.0			
Maximum memory frequency for CAS latency 7.0			
Maximum memory frequency for CAS latency 8.0			
Maximum memory frequency for CAS latency 9.0			
Maximum memory frequency for CAS latency 10.0			

Note to Table 8-15:

(1) The range values depend on the actual memory device used.

Table 8-16. DDR3 SDRAM Initialization Options (Part 1 of 3)

Parameter Name	Range	Units	Description
Memory burst length	4, 8, on-the-fly	beats	Sets the number of words read or written per transaction.
Memory burst ordering	Sequential or Interleaved	—	Controls the order in which data is transferred between memory and the FPGA during a read transaction. For more information, refer to the memory device datasheet.
DLL precharge power down	Fast exit or Slow exit	—	Sets the mode register setting to disable (Slow exit) or enable (Fast exit) the memory DLL when CKE is disabled.

Table 8-16. DDR3 SDRAM Initialization Options (Part 2 of 3)

Parameter Name	Range	Units	Description
Enable the DLL in the memory devices	Yes or No	—	Enables the DLL in the memory device when set to Yes . You must always enable the DLL in the memory device as Altera does not guarantee any ALTMEMPHY operation when the DLL is turned off. All timings from the memory devices are invalid when the DLL is turned off.
ODT R_{tt} nominal value	ODT disable, RZQ/4, RZQ/2, RZQ/6	W	RZQ in DDR3 SDRAM interfaces are set to 240 Ω . Sets the on-die termination (ODT) value to either 60 Ω (RZQ/4), 120 Ω (RZQ/2), or 40 Ω (RZQ/6). Set this to ODT disable if you are not planning to use ODT. For a single-ranked DIMM, set this to RZQ/4 .
Dynamic ODT (R_{tt_WR}) value	Dynamic ODT off, RZQ/4, RZQ/2	W	RZQ in DDR3 SDRAM interfaces are set to 240 Ω . Sets the memory ODT value during write operations to 60 Ω (RZQ/4) or 120 Ω (RZQ/2). As ALTMEMPHY only supports single rank DIMMs, you do not need this option (set to Dynamic ODT off).
Output driver impedance	RZQ/6 (Reserved) or RZQ/7	W	RZQ in DDR3 SDRAM interfaces are set to 240 Ω . Sets the output driver impedance from the memory device. Some devices may not have RZQ/6 available as an option. Be sure to check the memory device datasheet before choosing this option.
Memory CAS latency setting	5.0, 6.0, 7.0, 8.0, 9.0, 10.0	cycles	Sets the delay in clock cycles from the read command to the first output data from the memory.
Memory additive CAS latency setting	Disable, CL - 1, CL - 2	cycles	Allows you to add extra latency in addition to the CAS latency setting.
Memory write CAS latency setting (CWL)	5.0, 6.0, 7.0, 8.0	cycles	Sets the delay in clock cycles from the write command to the first expected data to the memory.
Memory partial array self refresh	Full array {BA[2:0]=000,001, 010,011}, Quarter array {BA[2:0]=000,001}, Eighth array {BA[2:0]=000}, Three Quarters array {BA[2:0]=010,011, 100,101,110,111}, Half array {BA[2:0]=100,101, 110,111}, Quarter array {BA[2:0]=110, 111}, Eighth array {BA[2:0]=111}	—	Determine whether you want to self-refresh only certain arrays instead of the full array. According to the DDR3 SDRAM specification, data located in the array beyond the specified address range are lost if self refresh is entered when you use this. This option is not supported by the DDR3 SDRAM Controller with ALTMEMPHY IP, so set to Full Array if you are using the Altera controller.

Table 8-16. DDR3 SDRAM Initialization Options (Part 3 of 3)

Parameter Name	Range	Units	Description
Memory auto self refresh method	Manual SR reference (SRT) or ASR enable (Optional)	—	Sets the auto self-refresh method for the memory device. The DDR3 SDRAM Controller with ALTMEMPHY IP currently does not support the ASR option that you need for extended temperature memory self-refresh.
Memory self refresh range	Normal or Extended	—	Determines the temperature range for self refresh. You need to also use the optional auto self refresh option when using this option. The Altera controller currently does not support the extended temperature self-refresh operation.

Table 8-17. DDR3 SDRAM Timing Parameter Settings (Part 1 of 3) (1)

Parameter Name	Range	Units	Description
Time to hold memory reset before beginning calibration	0–1000000	μs	Minimum time to hold the reset after a power cycle before issuing the MRS commands during the DDR3 SDRAM device initialization process.
t_{INIT}	0.001–1000	μs	Minimum memory initialization time. After reset, the controller does not issue any commands to the memory during this period.
t_{MRD}	2–39	ns	Minimum load mode register command period. The controller waits for this period of time after issuing a load mode register command before issuing any other commands. t_{MRD} is specified in ns in the DDR3 SDRAM high-performance controller and in terms of t_{CK} cycles in Micron's device datasheet. Convert t_{MRD} to ns by multiplying the number of cycles specified in the datasheet times t_{CK} , where t_{CK} is the memory operation frequency and not the memory device's t_{CK} .
t_{RAS}	8–200	ns	Minimum active to precharge time. The controller waits for this period of time after issuing an active command before issuing a precharge command to the same bank.
t_{RCD}	4–65	ns	Minimum active to read-write time. The controller does not issue read or write commands to a bank during this period of time after issuing an active command.
t_{RP}	4–65	ns	Minimum precharge command period. The controller does not access the bank for this period of time after issuing a precharge command.
t_{REFI}	1–65534	μs	Maximum interval between refresh commands. The controller performs regular refresh at this interval unless user-controlled refresh is turned on.
t_{RFC}	14–1651	ns	Minimum autorefresh command period. The length of time the controller waits before doing anything else after issuing an auto-refresh command.
t_{WR}	4–65	ns	Minimum write recovery time. The controller waits for this period of time after the end of a write transaction before issuing a precharge command.

Table 8-17. DDR3 SDRAM Timing Parameter Settings (Part 2 of 3) (1)

Parameter Name	Range	Units	Description
t_{WTR}	1–6	t_{CK}	Minimum write-to-read command delay. The controller waits for this period of time after the end of a write command before issuing a subsequent read command to the same bank. This timing parameter is specified in clock cycles and the value is rounded off to the next integer.
t_{AC}	0–750	ps	DQ output access time.
t_{DQACK}	50–750	ps	DQS output access time from CK/CK# signals.
t_{DQSQ}	50–500	ps	The maximum DQS to DQ skew; DQS to last DQ valid, per group, per access.
t_{DQSS}	0–0.3	t_{CK}	Positive DQS latching edge to associated clock edge.
t_{DH}	10–600	ps	DQ and DM input hold time relative to DQS, which has a derated value depending on the slew rate of the differential DQS and DQ/DM signals. Ensure that you are using the correct number and that the value entered is referenced to $V_{REF}(dc)$, not $V_{IH}(dc)$ min or $V_{IL}(dc)$ max. Refer to “Derating Memory Setup and Hold Timing” on page 8–50 for more information about how to derate this specification.
t_{DS}	10–600	ps	DQ and DM input setup time relative to DQS, which has a derated value depending on the slew rate of the differential DQS signals and DQ/DM signals. Ensure that you are using the correct number and that the value entered is referenced to $V_{REF}(dc)$, not $V_{IH}(ac)$ min or $V_{IL}(ac)$ max. Refer to “Derating Memory Setup and Hold Timing” on page 8–50 for more information about how to derate this specification.
t_{DSH}	0.1–0.5	t_{CK}	DQS falling edge hold time from CK.
t_{DSS}	0.1–0.5	t_{CK}	DQS falling edge to CK setup.
t_{IH}	50–1000	ps	Address and control input hold time, which has a derated value depending on the slew rate of the CK and CK# clocks and the address and command signals. Ensure that you are using the correct number and that the value entered is referenced to $V_{REF}(dc)$, not $V_{IH}(dc)$ min or $V_{IL}(dc)$ max. Refer to “Derating Memory Setup and Hold Timing” on page 8–50 for more information about how to derate this specification.
t_{IS}	65–1000	ps	Address and control input setup time, which has a derated value depending on the slew rate of the CK and CK# clocks and the address and command signals. Ensure that you are using the correct number and that the value entered is referenced to $V_{REF}(dc)$, not $V_{IH}(ac)$ min or $V_{IL}(ac)$ max. Refer to “Derating Memory Setup and Hold Timing” on page 8–50 for more information about how to derate this specification.
t_{QHS}	0–700	ps	The maximum data hold skew factor.
t_{QH}	0.1–0.6	t_{CK}	DQ output hold time.

Table 8-17. DDR3 SDRAM Timing Parameter Settings (Part 3 of 3) (1)

Parameter Name	Range	Units	Description
t_{RRD}	2.06–64	ns	The activate to activate time, per device, RAS to RAS delay timing parameter.
t_{FAW}	7.69–256	ns	The four-activate window time, per device.
t_{RTP}	2.06–64	ns	Read to precharge time.

Note to Table 8-17:

- (1) Refer to the memory device data sheet for the parameter range. Some of the parameters are listed in a clock cycle (t_{CK}) unit. If the MegaWizard Plug-In Manager requires you to enter the value in a time unit (ps or ns), convert the number by multiplying it with the clock period of your interface (and not the maximum clock period listed in the memory data sheet).

Derating Memory Setup and Hold Timing

Because the base setup and hold time specifications from the memory device datasheet assume input slew rates that may not be true for Altera devices, derate and update the following memory device specifications in the **Preset Editor** dialog box:

- t_{DS}
- t_{DH}
- t_{IH}
- t_{IS}



For Arria II GX and Stratix IV devices (excluding DDR SDRAM), you need not derate using the **Preset Editor**. You only need to enter the parameters referenced to V_{REF} and the deration is done automatically when you enter the slew rate information on the **Board Settings** tab.


After derating the values, you then need to normalize the derated value because Altera input and output timing specifications are referenced to V_{REF} . However, JEDEC base setup time specifications are referenced to V_{IH}/V_{IL} AC levels; JEDEC base hold time specifications are referenced to V_{IH}/V_{IL} DC levels.

When the memory device setup and hold time numbers are derated and normalized to V_{REF} update these values in the **Preset Editor** dialog box to ensure that your timing constraints are correct.

Example 8-1. Derating DDR2 SDRAM

For example, according to JEDEC, 400-MHz DDR2 SDRAM has the following specifications, assuming 1V/ns DQ slew rate rising signal and 2V/ns differential slew rate:

- Base $t_{DS} = 50$
- Base $t_{DH} = 125$
- $V_{IH}(ac) = V_{REF} + 0.2 V$
- $V_{IH}(dc) = V_{REF} + 0.125V$
- $V_{IL}(ac) = V_{REF} - 0.2 V$
- $V_{IL}(dc) = V_{REF} - 0.125 V$

 JEDEC lists two different sets of base and derating numbers for t_{DS} and t_{DH} specifications, whether you are using single-ended or differential DQS signaling, for any DDR2 SDRAM components with a maximum frequency up to 267 MHz. In addition, the $V_{IL}(ac)$ and $V_{IH}(ac)$ values may also be different for those devices.

The V_{REF} referenced setup and hold signals for a rising edge are:

$$t_{DS}(V_{REF}) = \text{Base } t_{DS} + \text{delta } t_{DS} + (V_{IH}(ac) - V_{REF})/\text{slew_rate} = 50 + 0 + 200 = 250 \text{ ps}$$

$$t_{DH}(V_{REF}) = \text{Base } t_{DH} + \text{delta } t_{DH} + (V_{IH}(dc) - V_{REF})/\text{slew_rate} = 125 + 0 + 67.5 = 192.5 \text{ ps}$$

If the output slew rate of the write data is different from 1V/ns, you have to first derate the t_{DS} and t_{DH} values, then translate these AC/DC level specs to V_{REF} specification.

For a 2V/ns DQ slew rate rising signal and 2V/ns DQS-DQSn slew rate:

$$t_{DS}(V_{REF}) = \text{Base } t_{DS} + \text{delta } t_{DS} + (V_{IH}(ac) - V_{REF})/\text{slew_rate} = 25 + 100 + 100 = 225 \text{ ps}$$

$$t_{DH}(V_{REF}) = \text{Base } t_{DH} + \text{delta } t_{DH} + (V_{IH}(dc) - V_{REF})/\text{slew_rate} = 100 + 45 + 62.5 = 207.5 \text{ ps}$$

For a 0.5V/ns DQ slew rate rising signal and 1V/ns DQS-DQSn slew rate:

$$t_{DS}(V_{REF}) = \text{Base } t_{DS} + \text{delta } t_{DS} + (V_{IH}(ac) - V_{REF})/\text{slew_rate} = 25 + 0 + 400 = 425 \text{ ps}$$

$$t_{DH}(V_{REF}) = \text{Base } t_{DH} + \text{delta } t_{DH} + (V_{IH}(dc) - V_{REF})/\text{slew_rate} = 100 - 65 + 250 = 285 \text{ ps}$$

A similar approach can be taken to address/command slew rate derating. For t_{IS}/t_{IH} the slew rate used in the derating equations is the address/command slew rate; for t_{DS}/t_{DH} the DQ slew rate is used.

Example 8-2. Derating DDR3 SDRAM

For example, according to JEDEC, 533-MHz DDR3 SDRAM has the following specifications, assuming 1V/ns DQ slew rate rising signal and 2V/ns DQS-DQSn slew rate:

- Base $t_{DS} = 25$
- Base $t_{DH} = 100$
- $V_{IH}(ac) = V_{REF} + 0.175 \text{ V}$
- $V_{IH}(dc) = V_{REF} + 0.100 \text{ V}$
- $V_{IL}(ac) = V_{REF} - 0.175 \text{ V}$
- $V_{IL}(dc) = V_{REF} - 0.100 \text{ V}$

The V_{REF} referenced setup and hold signals for a rising edge are:

$$t_{DS}(V_{REF}) = \text{Base } t_{DS} + \text{delta } t_{DS} + (V_{IH}(ac) - V_{REF})/\text{slew_rate} = 25 + 0 + 175 = 200 \text{ ps}$$

$$t_{DH}(V_{REF}) = \text{Base } t_{DH} + \text{delta } t_{DH} + (V_{IH}(dc) - V_{REF})/\text{slew_rate} = 100 + 0 + 100 = 200 \text{ ps}$$

If the output slew rate of the write data is different from 1V/ns, you have to first derate the t_{DS} and t_{DH} values, then translate these AC/DC level specs to V_{REF} specification.

For a 2V/ns DQ slew rate rising signal and 2V/ns DQS-DQSn slew rate:

$$t_{DS}(V_{REF}) = \text{Base } t_{DS} + \text{delta } t_{DS} + (V_{IH(ac)} - V_{REF})/\text{slew_rate} = 25 + 88 + 87.5 = 200.5 \text{ ps}$$

$$t_{DH}(V_{REF}) = \text{Base } t_{DH} + \text{delta } t_{DH} + (V_{IH(dc)} - V_{REF})/\text{slew_rate} = 100 + 50 + 50 = 200 \text{ ps}$$

For a 0.5V/ns DQ slew rate rising signal and 1V/ns DQS-DQSn slew rate:

$$t_{DS}(V_{REF}) = \text{Base } t_{DS} + \text{delta } t_{DS} + (V_{IH(ac)} - V_{REF})/\text{slew_rate} = 25 + 5 + 350 = 380 \text{ ps}$$

$$t_{DH}(V_{REF}) = \text{Base } t_{DH} + \text{delta } t_{DH} + (V_{IH(dc)} - V_{REF})/\text{slew_rate} = 100 + 10 + 200 = 310 \text{ ps}$$

PHY Settings

Click **Next** or the **PHY Settings** tab to set the options described in [Table 8-18](#). The options are available if they apply to the target Altera device.

Table 8-18. ALTMEMPHY PHY Settings (Part 1 of 3)

Parameter Name	Applicable Device Families		Description
	DDR/DDR2 SDRAM	DDR3 SDRAM	
Use dedicated PLL outputs to drive memory clocks	HardCopy II and Stratix II (prototyping for HardCopy II)	Not supported	Turn on to use dedicated PLL outputs to generate the external memory clocks, which is required for HardCopy II ASICs and their Stratix II FPGA prototypes. When turned off, the DDIO output registers generate the clock outputs. When you use the DDIO output registers for the memory clock, both the memory clock and the DQS signals are well aligned and easily meets the t_{DQSS} specification. However, when the dedicated clock outputs are for the memory clock, the memory clock and the DQS signals are not aligned properly and requires a positive phase offset from the PLL to align the signals together.
Dedicated memory clock phase	HardCopy II and Stratix II (prototyping for HardCopy II)	Not supported	The required phase shift to align the CK/CK# signals with DQS/DQS# signals when using dedicated PLL outputs to drive memory clocks.
Use differential DQS	Arria II GX, Stratix III, and Stratix IV	Not supported	Enable this feature for better signal integrity. Recommended for operation at 333 MHz or higher. An option for DDR2 SDRAM only, as DDR SDRAM does not support differential DQSS.

Table 8-18. ALTMEMPHY PHY Settings (Part 2 of 3)

Parameter Name	Applicable Device Families		Description
	DDR/DDR2 SDRAM	DDR3 SDRAM	
Enable external access to reconfigure PLL prior to calibration	HardCopy II, Stratix II, Stratix III, and Stratix IV (prototyping for HardCopy II)	HardCopy II	<p>When enabling this option for HardCopy II, Stratix II, Stratix III, and Stratix IV devices, the inputs to the ALTPLL_RECONFIG megafunction are brought to the top level for debugging purposes.</p> <p>This option allows you to reconfigure the PLL before calibration to adjust, if necessary, the phase of the memory clock (<code>mem_clk_2x</code>) before the start of the calibration of the resynchronization clock on the read side. The calibration of the resynchronization clock on the read side depends on the phase of the memory clock on the write side.</p>
Instantiate DLL externally	All supported device families, except for Cyclone® III devices	All supported device families	<p>Use this option with Stratix III, Stratix IV, HardCopy III, or HardCopy IV devices, if you want to apply a non-standard phase shift to the DQS capture clock. The ALTMEMPHY DLL offsetting I/O can then be connected to the external DLL and the Offset Control Block.</p> <p>As Cyclone III devices do not have DLLs, this feature is not supported.</p>
Enable dynamic parallel on-chip termination	Stratix III and Stratix IV	Not supported	<p>This option provides I/O impedance matching and termination capabilities. The ALTMEMPHY megafunction enables parallel termination during reads and series termination during writes with this option checked. Only applicable for DDR and DDR2 SDRAM interfaces where DQ and DQS are bidirectional. Using the dynamic termination requires that you use the OCT calibration block, which may impose a restriction on your DQS/DQ pin placements depending on your R_{UP}/R_{DN} pin locations.</p> <p>Although DDR SDRAM does not support ODT, dynamic OCT is still supported in Altera FPGAs.</p> <p>For more information, refer to the <i>External Memory Interfaces in Stratix III Devices</i> chapter in volume 1 of the <i>Stratix III Device Handbook</i> or the <i>External Memory Interfaces in Stratix IV Devices</i> chapter in volume 1 of the <i>Stratix IV Device Handbook</i>.</p>
Clock phase	Arria II GX, Arria GX, Cyclone III, HardCopy II, Stratix II, and Stratix II GX	Arria II GX	<p>Adjusting the address and command phase can improve the address and command setup and hold margins at the memory device to compensate for the propagation delays that vary with different loadings. You have a choice of 0°, 90°, 180°, and 270°, based on the rising and falling edge of the <code>phy_clk</code> and <code>write_clk</code> signals. In Stratix IV and Stratix III devices, the clock phase is set to dedicated.</p>

Table 8–18. ALTMEMPHY PHY Settings (Part 3 of 3)

Parameter Name	Applicable Device Families		Description
	DDR/DDR2 SDRAM	DDR3 SDRAM	
Dedicated clock phase	Stratix III and Stratix IV	Not supported	When you use a dedicated PLL output for address and command, you can choose any legal PLL phase shift to improve setup and hold for the address and command signals. You can set this value to between 180° and 359°, the default is 240°. However, generally PHY timing requires a value of greater than 240° for half-rate designs and 270° for full-rate designs.
Board skew	All supported device families except Arria II GX and Stratix IV devices	Not supported	Maximum skew across any two memory interface signals for the whole interface from the FPGA to the memory (either a discrete memory device or a DIMM). This parameter includes all types of signals (data, strobe, clock, address, and command signals). You need to input the worst-case skew, whether it is within a DQS/DQ group, or across all groups, or across the address and command and clocks signals. This parameter generates the timing constraints in the <code>.sdc</code> .
Autocalibration simulation options	All supported device families		Choose between Full Calibration (long simulation time), Quick Calibration , or Skip Calibration . For more information, refer to the “Simulation Options” section in the <i>Simulating Memory IP</i> chapter.

Board Settings

Click **Next** or the **Board Settings** tab to set the options described in Table 8–19. The board settings parameters are set to model the board level effects in the timing analysis. The options are available if you choose Arria II GX or Stratix IV device for your interface. Otherwise, the options are disabled. The options are also disabled for all devices using DDR SDRAM.

Table 8–19. ALTMEMPHY Board Settings (Part 1 of 2)

Parameter Name	Units	Description
Number of slots/discrete devices	—	Sets the single-rank or multi-rank configuration.
CK/CK# slew rate (differential)	V/ns	Sets the differential slew rate for the CK and CK# signals.
Addr/command slew rate	V/ns	Sets the slew rate for the address and command signals.
DQ/DQS# slew rate (differential)	V/ns	Sets the differential slew rate for the DQ and DQS# signals.
DQ slew rate	V/ns	Sets the slew rate for the DQ signals.
Addr/command eye reduction (setup)	ns	Sets the reduction in the eye diagram on the setup side due to the ISI on the address and command signals.
Addr/command eye reduction (hold)	ns	Sets the reduction in the eye diagram on the hold side due to the ISI on the address and command signals.
DQ eye reduction	ns	Sets the total reduction in the eye diagram on the setup side due to the ISI on the DQ signals.
Delta DQS arrival time	ns	Sets the increase of variation on the range of arrival times of DQS due to ISI.

Table 8–19. ALTMEMPHY Board Settings (Part 2 of 2)

Parameter Name	Units	Description
Max skew between DIMMs/devices	ns	Sets the largest skew or propagation delay on the DQ signals between ranks, especially true for DIMMs in different slots. This value affects the Resynchronization margin for the DDR2 interfaces in multi-rank configurations for both DIMMs and devices.
Max skew within DQS group	ns	Sets the largest skew between the DQ pins in a DQS group. This value affects the Read Capture and Write margins for the DDR2 interfaces in all configurations (single- or multi-rank, DIMM or device).
Max skew between DQS groups	ns	Sets the largest skew between DQS signals in different DQS groups. This value affects the Resynchronization margin for the DDR2 interfaces in both single- or multi-rank configurations.
Addr/command to CK skew	ns	Sets the skew or propagation delay between the CK signal and the address and command signals. The positive values represent the address and command signals that are longer than the CK signals, and the negative values represent the address and command signals that are shorter than the CK signals. This skew is used by the Quartus II software to optimize the delay of the address/command signals to have appropriate setup and hold margins for the DDR2 interfaces.

Controller Settings


 This section describes parameters for the High Performance Controller II (HPC II) with advanced features introduced in version 11.0 for designs generated in version 11.0. Designs created in earlier versions and regenerated in version 11.0 do not inherit the new advanced features; for information on parameters for HPC II without the version 11.0 advanced features, refer to the *External Memory Interface Handbook* for Quartus II version 10.1, available in the [Literature: External Memory Interfaces](#) page of the Altera website.

Table 8–20 lists the options provided in the **Controller Settings** tab.


Table 8–20. Controller Settings (Part 1 of 3)

Parameter	Description
Controller architecture	Specifies the controller architecture.
Enable self-refresh controls	Turn on to enable the controller to allow you to have control on when to place the external memory device in self-refresh mode, refer to the “User-Controlled Self-Refresh” section in the <i>Functional Description—HPC II Controller</i> chapter of the <i>External Memory Interface Handbook</i> .
Enable power down controls	Turn on to enable the controller to allow you to have control on when to place the external memory device in power-down mode.
Enable auto power down	Turn on to enable the controller to automatically place the external memory device in power-down mode after a specified number of idle controller clock cycles is observed in the controller. You can specify the number of idle cycles after which the controller powers down the memory in the Auto Power Down Cycles field, refer to the “Automatic Power-Down with Programmable Time-Out” section in the <i>Functional Description—HPC II Controller</i> chapter of the <i>External Memory Interface Handbook</i> .

Table 8–20. Controller Settings (Part 2 of 3)

Parameter	Description
Auto power down cycles	Determines the desired number of idle controller clock cycles before the controller places the external memory device in a power-down mode. The legal range is 1 to 65,535. The auto power-down mode is disabled if you set the value to 0 clock cycles.
Enable user auto-refresh controls	Turn on to enable the controller to allow you to issue a single refresh.
Enable auto-precharge control	Turn on to enable the auto-precharge control on the controller top level. Asserting the auto-precharge control signal while requesting a read or write burst allows you to specify whether or not the controller should close (auto-precharge) the current opened page at the end of the read or write burst.
Enable reordering	Turn on to allow the controller to perform command and data reordering to achieve the highest efficiency.
Starvation limit for each command	Specifies the number of commands that can be served before a waiting command is served. The legal range is from 1 to 63.
Local-to-memory address mapping	Allows you to control the mapping between the address bits on the Avalon interface and the chip, row, bank, and column bits on the memory interface. If your application issues bursts that are greater than the column size of the memory device, choose the Chip-Row-Bank-Column option. This option allows the controller to use its look-ahead bank management feature to hide the effect of changing the currently open row when the burst reaches the end of the column. On the other hand, if your application has several masters that each use separate areas of memory, choose the Chip-Bank-Row-Column option. This option allows you to use the top address bits to allocate a physical bank in the memory to each master. The physical bank allocation avoids different masters accessing the same bank which is likely to cause inefficiency, as the controller must then open and close rows in the same bank.
Command queue look-ahead depth	Specifies a command queue look-ahead depth value to control the number of read or write requests the look-ahead bank management logic examines.
Local maximum burst count	Specifies a burst count to configure the maximum Avalon burst count that the controller slave port accepts.
Reduce controller latency by	Specifies, in controller clock cycles, a value by which to reduce the controller latency. The default value is 0 but you have the option to choose 1 to enhance the latency performance of your design at the expense of timing closure.
Enable configuration and status register interface	Turn on to enable run-time configuration and status retrieval of the memory controller. Enabling this option adds an additional Avalon-MM slave port to the memory controller top level that allows run-time reconfiguration and status retrieving for memory timing parameters, memory address size and mode register settings, and controller features. If the Error Detection and Correction Logic option is enabled, the same slave port also allows you to control and retrieve the status of this logic. For more information, refer to the “Configuration and Status Register (CSR) Interface” section in the <i>Functional Description—HPC II Controller</i> chapter of the <i>External Memory Interface Handbook</i>
Enable error detection and correction logic	Turn on to enable error correction coding (ECC) for single-bit error correction and double-bit error detection.
Enable auto error correction	Turn on to allow the controller to perform auto correction when the ECC logic detects a single-bit error. Alternatively, you can turn off this option and schedule the error correction at a desired time for better system efficiency.

Table 8–20. Controller Settings (Part 3 of 3)

Parameter	Description
Multiple controller clock sharing	<p>This option is only available in SOPC Builder Flow. Turn on to allow one controller to use the Avalon clock from another controller in the system that has a compatible PLL. This option allows you to create SOPC Builder systems that have two or more memory controllers that are synchronous to your master logic.</p> <p> This option is not for use with Cyclone III or Cyclone IV family devices.</p>
Local interface protocol	<p>Specifies the local side interface between the user logic and the memory controller. The Avalon-MM interface allows you to easily connect to other Avalon-MM peripherals. The HPC II architecture supports only the Avalon-MM interface.</p>

Parameterizing Memory Controllers with UniPHY IP

This section describes the parameters you can set for the DDR2, DDR3 SDRAM, QDR II, QDR II+ SRAM, and RLDRAM II memory controllers with the UniPHY IP.

The **Parameter Settings** page in the UniPHY parameter editor allows you to parameterize the following settings:

- PHY Settings
- Memory Parameters
- Memory Timing
- Board Settings
- Controller Settings
- Diagnostics

The text window at the bottom of the MegaWizard Plug-In Manager displays information about the memory interface, warnings, and errors if you are trying to create something that is not supported. The **Finish** button is disabled until you correct all the errors indicated in this window.

The following sections describe the tabs of the **Parameter Settings** page in more detail.

PHY Settings

Table 8–21 lists the PHY parameters.

Table 8–21. Clock Parameters

Parameter	Description
General Settings	
Speed Grade	Specifies the speed grade of the targeted FPGA device that affects the generated timing constraints and timing reporting.
Generate PHY only	Turn on this option to generate the UniPHY core without a memory controller. When you turn on this option, the AFI interface is exported so that you can easily connect your own memory controller.

Table 8–21. Clock Parameters

Parameter	Description
Clocks	
Memory clock frequency	The frequency of the clock that drives the memory device. Use up to 4 decimal places of precision. To obtain the maximum supported frequency for your target memory configuration, refer to the External Memory Interface Spec Estimator page on the Altera website.
Achieved memory clock frequency	The actual frequency the PLL generates to drive the external memory interface (memory clock).
PLL reference clock frequency	The frequency of the input clock that feeds the PLL. Use up to 4 decimal places of precision.
Rate on Avalon-MM interface	The width of data bus on the Avalon-MM interface. Full results in a width of 2× the memory data width. Half results in a width of 4× the memory data width. Quarter results in a width of 8× the memory data width and is only supported for DDR3 SDRAM using Stratix V devices. Use Quarter for memory frequency 533 MHz and above. To determine the Avalon-MM interface rate selection for other memories, refer to the local interface clock rate for your target device in the External Memory Interface Spec Estimator page on the Altera website.
Achieved local clock frequency	The actual frequency the PLL generates to drive the local interface for the memory controller (AFI clock).

Table 8–21. Clock Parameters

Parameter	Description
Advanced PHY Settings	
Advanced clock phase control	<p>Enables access to clock phases. Default value should suffice for most DIMMs and board layouts, but can be modified if necessary to compensate for larger address and command versus clock skews.</p> <p>This option is available for DDR3 SDRAM only.</p>
Additional address and command clock phase	<p>Allows you to increase or decrease the amount of phase shift on the address and command clock. The base phase shift center aligns the address and command clock at the memory device, which may not be the optimal setting under all circumstances. Increasing or decreasing the amount of phase shift can improve timing. The default value is 0 degrees.</p> <p>To achieve the optimum setting, adjust the value based on the address and command timing analysis results.</p> <p>This option is not available for Stratix V devices.</p>
Additional phase for core-to-periphery transfer	<p>Allows you to phase shift the latching clock of the core-to-periphery transfers. By delaying the latch clock, a positive phase shift value improves setup timing for transfers between registers in the core and the half-rate DDIO_OUT blocks in the periphery, respectively. Adjust this setting according to the core timing analysis.</p> <p>This option is available for Stratix V devices only.</p>
Additional phase for periphery-to-core transfer	<p>Allows you to phase shift the latching clock of the periphery-to-core transfers. By advancing the latch clock, a negative phase shift value improves setup timing for transfers between read-fifo in the periphery and the core. Adjust this setting according to the core timing analysis.</p> <p>This option is available for Stratix V devices only.</p>
Additional CK/CK# phase	<p>Allows you to increase or decrease the amount of phase shift on the CK/CK# clock. The base phase shift center aligns the address and command clock at the memory device, which may not be the optimal setting under all circumstances. Increasing or decreasing the amount of phase shift can improve timing. Increasing or decreasing the phase shift on CK/CK# also impacts the read, write, and leveling transfers, which increasing or decreasing the phase shift on the address and command clocks does not.</p> <p>To achieve the optimum setting, adjust the value based on the address and command timing analysis results. Ensure that the read, write, and write leveling timings are met after adjusting the clock phase. Adjust this value when there is a core timing failure after adjusting Additional address and command clock phase.</p> <p>This option is available for DDR3 SDRAM only. However, this option is not available for Stratix V devices.</p>
Enable read DQS tracking	<p>Improves timing margins by continuously compensating temperature variations. When you turn on this option, you will observe increased design agree and the refresh command times are longer due to tracking accesses. Altera recommends that you turn on this option for design running at 533 MHz and above.</p> <p>This option is available for DDR3 SDRAM only.</p>

Table 8–21. Clock Parameters

Parameter	Description
Supply voltage	The supply voltage and sub-family type of memory. This option is available for DDR3 SDRAM only. DDR3L is currently supported only on Stratix V.
I/O standard	The I/O standard voltage. Set the I/O standard according to your design's memory standard.
PLL sharing mode	<p>When you select No sharing, the parameter editor instantiates a PLL block without exporting the PLL signals. When you select Master, the parameter editor instantiates a PLL block and exports the signals. When you select Slave, the parameter editor exposes a PLL interface and you must connect an external PLL master to drive the PLL slave interface signals.</p> <p>Select No sharing if you are not sharing PLLs, otherwise select Master or Slave.</p> <p>For more information about resource sharing, refer to “The DLL and PLL Sharing Interface” section in the <i>Functional Description—UniPHY</i> chapter of the <i>External Memory Interface Handbook</i>.</p> <p>You must modify the timing script file to reflect the resource sharing during timing analysis. For more information, refer to the UniPHY tutorials on the List of designs using Altera External Memory IP page of the Altera Wiki website.</p>
DLL sharing mode	<p>When you select No sharing, the parameter editor instantiates a DLL block without exporting the DLL signals. When you select Master, the parameter editor instantiates a DLL block and exports the signals. When you select Slave, the parameter editor exposes a DLL interface and you must connect an external DLL master to drive the DLL slave signals.</p> <p>Select No sharing if you are not sharing DLLs, otherwise select Master or Slave.</p> <p>For more information about resource sharing, refer to “The DLL and PLL Sharing Interface” section in the <i>Functional Description—UniPHY</i> chapter of the <i>External Memory Interface Handbook</i>.</p>
OCT sharing mode	<p>When you select No sharing, the parameter editor instantiates an OCT block without exporting the OCT signals. When you select Master, the parameter editor instantiates an OCT block and exports the signals. When you select Slave, the parameter editor exposes an OCT interface and you must connect an external OCT control block to drive the OCT slave signals.</p> <p>Select No sharing if you are not sharing OCT blocks, otherwise select Master or Slave.</p> <p>For more information about resource sharing, refer to “The OCT Sharing Interface” section in the <i>Functional Description—UniPHY</i> chapter of the <i>External Memory Interface Handbook</i>.</p>

Table 8–21. Clock Parameters

Parameter	Description
HardCopy compatibility	<p>Enables all required HardCopy compatibility options for the generated IP core. For some parameterizations, a pipeline stage is added to the write datapath to help the more challenging timing closure for designs using HardCopy devices; the pipeline stage does not affect the overall read and write latency.</p> <p>Turn on this option if you are migrating your design to a HardCopy device. For more information, refer to the <i>HardCopy Design Migration Guidelines</i> chapter.</p>
Reconfigurable PLL location	<p>When you set the PLL used in the UniPHY memory interface to be reconfigurable at run time, you must specify the location of the PLL. This assignment generates a PLL that can only be placed in the given sides.</p> <p>This option is enabled when you turn on HardCopy compatibility. In HardCopy designs, you must specify the PLL location according to the location of the interface.</p>
Sequencer optimization	<p>Select Performance to enable the Nios II-based sequencer, or Area to enable the RTL-based sequencer.</p> <p>Altera recommends that you enable the Nios-based sequencer for memory clock frequencies greater than 400 MHz and enable the RTL-based sequencer if you want to reduce resource utilization.</p> <p>This option is available for QDR II and QDR II+ SRAM, and RLDRAM II only.</p>

Memory Parameters

Use this tab to apply the memory parameters from your memory manufacturer’s data sheet.

DDR2 and DDR3 SDRAM

Table 8–22 lists the memory parameters for DDR2 and DDR3 SDRAM.

Table 8–22. Memory Parameters (Part 1 of 3)

Parameter	Description
Memory vendor	The vendor of the memory device. Select the memory vendor according to the memory vendor you use. For memory vendors that are not listed in the setting, select JEDEC with the nearest memory parameters and edit the parameter values according to the values of the memory vendor that you use. However, if you select a configuration from the list of memory presets, the default memory vendor for that preset setting is automatically selected.
Memory format	The format of the memory device. Select Discrete if you are using just the memory device. Select Unbuffered or Registered for DIMM format. Use the DIMM format to turn on levelling circuitry for DDR3 SDRAM.
Memory device speed grade	The maximum frequency at which the memory device can run.
Total interface width	The total number of DQ pins of the memory device. Limited to 144 bits for DDR2 and DDR3 SDRAM (with or without leveling).
DQ/DQS group size	The number of DQ bits per DQS group.

Table 8–22. Memory Parameters (Part 2 of 3)

Parameter	Description	
Number of DQS groups	The number of DQS groups is calculated automatically from the Total interface width and the DQ/DQS group size parameters.	
Number of chip selects	The number of chip-selects the IP core uses for the current device configuration. Specify the total number of chip-selects according to the number of DIMM slots and the number of rank for each slot. For example, select 4 chip-selects if there are two DIMM slots with two ranks in each DIMM slot.	
Number of clocks	The width of the clock bus on the memory interface.	
Row address width	The width of the row address on the memory interface.	
Column address width	The width of the column address on the memory interface.	
Bank-address width	The width of the bank address bus on the memory interface.	
Enable DM pins	Specifies whether the DM pins of the memory device are driven by the FPGA. You can turn off this option to avoid overusing FPGA device pins when using x4 mode memory devices. When you are using using x4 mode memory devices, turn off this option for DDR3 SDRAM. You must turn on this option if you are using Avalon byte enable.	
DQS# Enable (DDR2)	Turn on differential DQS signaling to improve signal integrity and system performance. This option is available for DDR2 SDRAM only.	
Memory Initialization Options—DDR2		
Address and command parity	Enables address/command parity checking.	
Mode Register 0	Burst length	Specifies the burst length.
	READ burst type	Determines whether the controller performs accesses within a given burst in sequential or interleaved order.
	DLL precharge power down	Determines whether the DLL in the memory device is in slow exit mode or in fast exit mode during precharge power down.
	Memory CAS latency setting	Determines the number of clock cycles between the READ command and the availability of the first bit of output data at the memory device. Set this parameter according to the target memory speed grade.
Mode Register 1	Output drive strength setting	Determines the output driver impedance setting at the memory device. To obtain the optimum signal integrity performance, select the optimum setting based on the board simulation results.
	Memory additive CAS latency setting	Determines the posted CAS additive latency of the memory device. Enable this feature to improve command and bus efficiency, and increase system bandwidth. For more information, refer to the <i>Optimizing the Controller</i> chapter.
	Memory on-die termination (ODT) setting	Determines the on-die termination resistance at the memory device. To obtain the optimum signal integrity performance, select the optimum setting based on the board simulation results.
Mode Register 2	SRT Enable	Determines the selfrefresh temperature (SRT). Select 1x refresh rate for normal temperature (0-85C) or select 2x refresh rate for high-temperature (>85C).
Memory Initialization Options—DDR3		

Table 8-22. Memory Parameters (Part 3 of 3)

Parameter		Description
Mirror Addressing: 1 per chip select		Specifies the mirror addressing. Enter ranks with mirrored addresses in this field. For example, for four chip selects, enter 1101 to mirror the address on chip select #3, #2, and #0.
Address and command parity		Enables address/command parity checking to detect errors in data transmission.
Mode Register 0	READ burst type	Specifies whether accesses within a given burst are in sequential or interleaved order.
	DLL precharge power down	Specifies whether the DLL in the memory device is off or on during precharge power-down.
	Memory CAS latency setting	The number of clock cycles between the read command and the availability of the first bit of output data at the memory device. Set this parameter according to the target memory speed grade.
Mode Register 1	Output drive strength setting	The output driver impedance setting at the memory device. To obtain the optimum signal integrity performance, select the optimum setting based on the board simulation results.
	Memory additive CAS latency setting	The posted CAS additive latency of the memory device. Enable this feature to improve command and bus efficiency, and increase system bandwidth. For more information, refer to the <i>Optimizing the Controller</i> chapter.
	ODT Rtt nominal value	The on-die termination resistance at the memory device. To obtain the optimum signal integrity performance, select the optimum setting based on the board simulation results.
Mode Register 2	Auto selfrefresh method	Disable or enable auto selfrefresh.
	Selfrefresh temperature	Specifies the selfrefresh temperature as Normal or Extended .
	Memory write CAS latency setting	The number of clock cycles from the releasing of the internal write to the latching of the first data in, at the memory device.
	Dynamic ODT (Rtt_WR) value	The mode of the dynamic ODT feature of the memory device. To obtain the optimum signal integrity performance, select the optimum setting based on the board simulation results.

QDR II and QDR II+ SRAM

Table 8-23 describes the memory parameters for QDR II and QDR II+ SRAM.

Table 8-23. Memory Parameters (Part 1 of 2)

Parameter	Description
Address width	The width of the address bus on the memory device.
Data width	The width of the data bus on the memory device.
Data-mask width	The width of the data-mask on the memory device,
CQ width	The width of the CQ (read strobe) bus on the memory device.
K width	The width of the K (write strobe) bus on the memory device.
Burst length	The burst length supported by the memory device.

Table 8–23. Memory Parameters (Part 2 of 2)

Parameter	Description
Topology	
x36 emulated mode	Emulates a larger memory-width interface using smaller memory-width interfaces on the FPGA. Turn on this option when the target FPGA do not support x36 DQ/DQS group. This option allows two x18 DQ/DQS groups to emulate 1 x36 read data group.
Emulated write groups	Number of write groups to use to form the x36 memory interface on the FPGA. Select 2 to use 2 x18 DQ/DQS group to form x36 write data group. Select 4 to use 4 x9 DQ/DQS group to form x36 write data group.
Device width	Specifies the number of memory devices used for width expansion.

RLDRAM II

Table 8–23 describes the memory parameters for RLDRAM II.

Table 8–24. Memory Parameters

Parameter	Description
Address width	The width of the address bus on the memory device.
Data width	The width of the data bus on the memory device.
Bank-address width	The width of the bank-address bus on the memory device.
Data-mask width	The width of the data-mask on the memory device,
QK width	The width of the QK (read strobe) bus on the memory device. Select 1 when data width is set to 9. Select 2 when data width is set to 18 or 36.
DK width	The width of the DK (write strobe) bus on the memory device. Select 1 when data width is set to 9 or 18. Select 2 when data width is set to 36.
Burst length	The burst length supported by the memory device.
Memory mode register configuration	Configuration bits that set the memory mode.
Topology	
Device width	Specifies the number of memory devices used for width expansion.

Memory Timing

Use this tab to apply the memory timings from your memory manufacturer's data sheet. Table 8-25 shows the memory timing parameters.

Table 8-25. Memory Timing Parameters (Part 1 of 2)

Parameter	Description
DDR2/DDR3 SDRAM	
tIS (base)	Address and control setup to CK clock rise.
tIH (base)	Address and control hold after CK clock rise.
tDS (base)	Data setup to clock (DQS) rise.
tDH (base)	Data hold after clock (DQS) rise.
tDQSQ	DQS, DQS# to DQ skew, per access.
tQHS (DDR2)	DQ output hold time from DQS, DQS# (absolute time value)
tQH (DDR3)	DQ output hold time from DQS, DQS# (percentage of tCK).
tDQSK	DQS output access time from CK/CK#.
tDQSS	First latching edge of DQS to associated clock edge (percentage of tCK).
tQSH (DDR3) tDQSH (DDR2)	DQS Differential High Pulse Width (percentage of tCK). Specifies the minimum high time of the DQS signal received by the memory.
tDSH	DQS falling edge hold time from CK (percentage of tCK).
tDSS	DQS falling edge to CK setup time (percentage of tCK).
tINIT	Memory initialization time at power-up.
tMRD	Load mode register command period.
tRAS	Active to precharge time.
tRCD	Active to read or write time.
tRP	Precharge command period.
tREFI	Refresh command interval.
tRFC	Auto-refresh command interval.
tWR	Write recovery time.
tWTR	Write to read period.
tFAW	Four active window time.
tRRD	RAS to RAS delay time.
tRTP	Read to precharge time.
QDR II and QDR II+	
tWL (cycles)	The write latency.
tRL (cycles)	The read latency.
tSA	The address and control setup to K clock rise.
tHA	The address and control hold after K clock rise.
tSD	The data setup to clock (K/K#) rise.
tHD	The data hold after clock (K/K#) rise.
tCQD	Echo clock high to data valid.
tCQDOH	Echo clock high to data invalid.

Table 8-25. Memory Timing Parameters (Part 2 of 2)

Parameter	Description
Internal jitter	The QDRII/II+ internal jitter.
TCQHCQnH	The CQ clock rise to CQn clock rise (rising edge to rising edge).
TKHKnH	The K clock rise to Kn clock rise (rising edge to rising edge).
RLDRAM II	
Maximum memory clock frequency	The maximum frequency at which the memory device can run.
Refresh interval	The refresh interval.
tCKH (%)	The input clock (CK/CK#) high expressed as a percentage of the full clock period.
tQKH (%)	The read clock (QK/QK#) high expressed as a percentage of tCKH.
tAS	Address and control setup to CK clock rise.
tAH	Address and control hold after CK clock rise.
tDS	Data setup to clock (CK/CK#) rise.
tDH	Data hold after clock (CK/CK#) rise.
tQKQ_max	QK clock edge to DQ data edge (in same group).
tQKQ_min	QK clock edge to DQ data edge (in same group).
tCKDK_max	Clock to input data clock (max).
tCKDK_min	Clock to input data clock (min).

Board Settings

Use the **Board Settings** tab to model the board-level effects in the timing analysis. The **Board Settings** tab allows you to set the following settings:

- Setup and hold derating (Available for DDR2/DDR3 SDRAM and RLDRAM II only)
- Intersymbol interference
- Board skews





For accurate timing results, you must enter board settings parameters that are correct for your PCB.

The IP core supports single and multiple chip-select configurations. Altera has determined the effects on the output signalling of these configurations for certain Altera boards, and has stored the effects on the output slew rate and the intersymbol interference (ISI) within the wizard.



These stored values are representative of specific Altera boards. You must change the values to account for the board-level effects for your board. You can use HyperLynx or similar simulators to obtain values that are representative of your board.

-  For more information about how to include your board simulation results in the Quartus II software and how to assign pins using pin planners, refer to the design flow tutorials and design examples on the [List of designs using Altera External Memory IP](#) page of the Altera Wiki website
-  For information about timing deration methodology, refer to the “Timing Deration Methodology for Multiple Chip Select DDR2 and DDR3 SDRAM Designs” section in the [Analyzing Timing of Memory IP](#) chapter.

Setup and Hold Derating

The slew rate of the output signals affects the setup and hold times of the memory device. You can specify the slew rate of the output signals to see their effect on the setup and hold times of both the address and command signals and the DQ signals, or specify the setup and hold times directly.


-  You should enter information derived during your PCB development process of prelayout (line) and postlayout (board) simulation.

Table 8–26 lists the setup and hold derating parameters.

Table 8–26. Setup and Hold Derating Parameters (Part 1 of 3)

Parameter	Description
DDR2/DDR3 SDRAM	
Derating method	Derating method. The default settings are based on Altera internal board simulation data. To obtain accurate timing analysis according to the condition of your board, Altera recommends that you perform board simulation and enter the slew rate in the Quartus II software to calculate the derated setup and hold time automatically or enter the derated setup and hold time directly. For more information, refer to the “Timing Deration Methodology for Multiple Chip Select DDR2 and DDR3 SDRAM Designs” section in the Analyzing Timing of Memory IP chapter.
CK/CK# slew rate (differential)	CK/CK# slew rate (differential).
Address/Command slew rate	Address and command slew rate.
DQS/DQS# slew rate (Differential)	DQS and DQS# slew rate (differential).
DQ slew rate	DQ slew rate.
tIS	Address/command setup time to CK.
tIH	Address/command hold time from CK.
tDS	Data setup time to DQS.
tDH	Data hold time from DQS.

Table 8-26. Setup and Hold Derating Parameters (Part 2 of 3)

Parameter	Description
RLDRAM II	
tAS Vref to CK/CK# Crossing	For a given address/command and CK/CK# slew rate, the memory device data sheet provides a corresponding "tAS Vref to CK/CK# Crossing" value that can be used to determine the derated address/command setup time.
tAS VIH MIN to CK/CK# Crossing	For a given address/command and CK/CK# slew rate, the memory device data sheet provides a corresponding "tAS VIH MIN to CK/CK# Crossing" value that can be used to determine the derated address/command setup time.
tAH CK/CK# Crossing to Vref	For a given address/command and CK/CK# slew rate, the memory device data sheet provides a corresponding "tAH CK/CK# Crossing to Vref" value that can be used to determine the derated address/command hold time.
tAH CK/CK# Crossing to VIH MIN	For a given address/command and CK/CK# slew rate, the memory device data sheet provides a corresponding "tAH CK/CK# Crossing to VIH MIN" value that can be used to determine the derated address/command hold time.
tDS Vref to CK/CK# Crossing	For a given data and DK/DK# slew rate, the memory device data sheet provides a corresponding "tDS Vref to CK/CK# Crossing" value that can be used to determine the derated data setup time.
tDS VIH MIN to CK/CK# Crossing	For a given data and DK/DK# slew rate, the memory device data sheet provides a corresponding "tDS VIH MIN to CK/CK# Crossing" value that can be used to determine the derated data setup time.
tDH CK/CK# Crossing to Vref	For a given data and DK/DK# slew rate, the memory device data sheet provides a corresponding "tDH CK/CK# Crossing to Vref" value that can be used to determine the derated data hold time.
tDH CK/CK# Crossing to VIH MIN	For a given data and DK/DK# slew rate, the memory device data sheet provides a corresponding "tDH CK/CK# Crossing to VIH MIN" value that can be used to determine the derated data hold time.
Derated tAS	The derated address/command setup time is calculated automatically from the "tAS", the "tAS Vref to CK/CK# Crossing", and the "tAS VIH MIN to CK/CK# Crossing" parameters.
Derated tAH	The derated address/command hold time is calculated automatically from the "tAH", the "tAH CK/CK# Crossing to Vref", and the "tAH CK/CK# Crossing to VIH MIN" parameters.

Table 8-26. Setup and Hold Derating Parameters (Part 3 of 3)

Parameter	Description
Derated tDS	The derated data setup time is calculated automatically from the "tDS", the "tDS Vref to CK/CK# Crossing", and the "tDS VIH MIN to CK/CK# Crossing" parameters.
Derated tDH	The derated data hold time is calculated automatically from the "tDH", the "tDH CK/CK# Crossing to Vref", and the "tDH CK/CK# Crossing to VIH MIN" parameters.

Intersymbol Inteference

Intersymbol interference is the distortion of a signal in which one symbol interferes with subsequent symbols. Typically, when going from a single chip-select configuration to a multiple chip-select configuration there is an increase in intersymbol interference because there are multiple stubs causing reflections.

Table 8-27 lists the intersymbol interference parameters.

Table 8-27. ISI Parameters (Part 1 of 2)

Parameter	Description
Derating method	Choose between default Altera settings (with specific Altera boards) or manually enter board simulation numbers obtained for your specific board. This option is supported in DDR2/DDR3 SDRAM only.
Address and command eye reduction (setup)	The reduction in the eye diagram on the setup side (or left side of the eye) due to ISI on the address and command signals compared to a case when there is no ISI. (For single rank designs, ISI can be zero; in multirank designs, ISI is necessary for accurate timing analysis.) For more information about how to measure the ISI value for the address and command signals, refer to the "Measuring Eye Reduction for Address/Command, DQ, and DQS Setup and Hold Time" section in the <i>Analyzing Timing of Memory IP</i> chapter.
Address and command eye reduction (hold)	The reduction in the eye diagram on the hold side (or right side of the eye) due to ISI on the address and command signals compared to a case when there is no ISI. For more information about how to measure the ISI value for the address and command signals, refer to "Measuring Eye Reduction for Address/Command, DQ, and DQS Setup and Hold Time" section in the <i>Analyzing Timing of Memory IP</i> chapter.

Table 8-27. ISI Parameters (Part 2 of 2)

Parameter	Description
DQ/ D eye reduction	The total reduction in the eye diagram due to ISI on DQ signals compared to a case when there is no ISI. Altera assumes that the ISI reduces the eye width symmetrically on the left and right side of the eye. For more information about how to measure the ISI value for the address and command signals, refer to “Measuring Eye Reduction for Address/Command, DQ, and DQS Setup and Hold Time” section in the <i>Analyzing Timing of Memory IP</i> chapter.
Delta DQS/Delta K/ Delta DK arrival time	The increase in variation on the range of arrival times of DQS compared to a case when there is no ISI. Altera assumes that the ISI causes DQS to further vary symmetrically to the left and to the right. For more information about how to measure the ISI value for the address and command signals, refer to “Measuring Eye Reduction for Address/Command, DQ, and DQS Setup and Hold Time” section in the <i>Analyzing Timing of Memory IP</i> chapter.

Board Skews

PCB traces can have skews between them that can reduce timing margins. Furthermore, skews between different chip selects can further reduce the timing margin in multiple chip-select topologies. The **Board Skews** section of the parameter editor allows you to enter parameters to compensate for these variations. Very large board trace skews should be specified in your board trace model.

Table 8-28 lists the board skew parameters.

Table 8-28. Board Skew Parameters (Part 1 of 6)

Parameter	Description
DDR2/DDR3 SDRAM	
Maximum CK delay to DIMM/device	The delay of the longest CK trace from the FPGA to the memory device, whether on a DIMM or the same PCB as the FPGA is expressed by the following equation: $\max_n(CK_n PathDelay)$ where n is the number of memory clocks. For example, the maximum CK delay for two pairs of memory clocks is expressed by the following equation: $\max_2(CK_1 PathDelay, CK_2 PathDelay)$
Maximum DQS delay to DIMM/device	The delay of the longest DQS trace from the FPGA to the memory device, whether on a DIMM or the same PCB as the FPGA is expressed by the following equation: $\max_n(DQS_n PathDelay)$ where n is the number of memory clocks. For example, the maximum DQS delay for two DQS is expressed by the following equation: $\max_2(DQS_1 PathDelay, DQS_2 PathDelay)$

Table 8–28. Board Skew Parameters (Part 2 of 6)

Parameter	Description
Minimum delay difference between CK and DQS	<p>The minimum skew (or largest negative skew) between the CK signal and any DQS signal when arriving at the same DIMM over all DIMMs is expressed by the following equation:</p> $\min_{n,m}(CK_n PathDelay - DQS_m PathDelay)$ <p>where n is the number of memory clocks and m is the number of DQS. For example, the minimum delay difference between CK and DQS for two pairs of memory clocks and four DQS signals (two DQS signals for each clock) is expressed by the following equation:</p> $\min_{2,2}\{(Ck_1 Delay - DQS_1 Delay), (Ck_1 Delay - DQS_2 Delay), (Ck_2 Delay - DQS_3 Delay), (Ck_2 Delay - DQS_4 Delay)\}$ <p>This parameter value affects the write leveling margin for DDR3 interfaces with leveling in multirank configurations.</p>
Maximum delay difference between CK and DQS	<p>The maximum skew (or largest positive skew) between the CK signal and any DQS signal when arriving at the same DIMM over all DIMMs is expressed by the following equation:</p> $\max_{n,m}(CK_n PathDelay - DQS_m PathDelay)$ <p>where n is the number of memory clocks and m is the number of DQS. For example, the maximum delay difference between CK and DQS for two pairs of memory clocks and four DQS signals (two DQS signals for each clock) is expressed by the following equation:</p> $\max_{2,2}\{(Ck_1 Delay - DQS_1 Delay), (Ck_1 Delay - DQS_2 Delay), (Ck_2 Delay - DQS_3 Delay), (Ck_2 Delay - DQS_4 Delay)\}$ <p>This value affects the write Leveling margin for DDR3 interfaces with leveling in multi-rank configurations.</p>
Maximum skew within DQS group	<p>The largest skew between DQ and DM signals in a DQS group. This value affects the read capture and write margins for DDR2 and DDR3 SDRAM interfaces in all configurations (single or multiple chip-select, DIMM or component).</p>
Maximum skew between DQS groups	<p>The largest skew between DQS signals in different DQS groups. This value affects the resynchronization margin in memory interfaces without leveling such as DDR2 SDRAM and discrete-device DDR3 SDRAM in both single- or multiple chip-select configurations.</p>
Average delay difference between DQ and DQS	<p>The average delay difference between each DQ signal and the DQS signal, calculated by averaging the longest and smallest DQ signal delay values minus the delay of DQS. The average delay difference between DQ and DQS is expressed by the following equation:</p> $\sum_{n=1}^n \left[\left(\frac{Longest\ DQ\ Path\ Delay\ in\ DQS_n\ group + Shortest\ DQ\ Path\ Delay\ in\ DQS_n\ group}{2} \right) - DQS_n PathDelay \right]$ <p>where n is the number of DQS groups.</p>
Maximum skew within address and command bus	<p>The largest skew between the address and command signals.</p>

Table 8-28. Board Skew Parameters (Part 3 of 6)

Parameter	Description
Average delay difference between address and command and CK	<p>A value equal to the average of the longest and smallest address and command signal delay values, minus the delay of the CK signal. The value can be positive or negative. Positive values represent address and command signals that are longer than CK signals; negative values represent address and command signals that are shorter than CK signals. The average delay difference between address and command and CK is expressed by the following equation:</p> $\sum_{n=1}^n \left[\left(\frac{\text{Longest AC Path Delay} + \text{Shortest AC Path Delay}}{2} \right) - \text{CK}_n \text{ Path Delay} \right]$ <p>where n is the number of memory clocks.</p> <p>The Quartus II software uses this skew to optimize the delay of the address and command signals to have appropriate setup and hold margins for DDR2 and DDR3 SDRAM interfaces. You should derive this value through board simulation.</p>
QDR II and QDR II+	
Maximum delay difference between devices	<p>The maximum delay difference of data signals between devices is expressed by the following equation:</p> $\text{Abs} \left[\left(\frac{\text{Longest device 1 delay} - \text{Shortest device 2 delay}}{2} \right) - \left(\frac{\text{Longest device 2 delay} - \text{Shortest device 1 delay}}{2} \right) \right]$ <p>For example, in a two-device configuration there is greater propagation delay for data signals going to and returning from the furthest device relative to the nearest device. This parameter is applicable for depth expansion. Set the value to 0 for non-depth expansion design.</p>
Maximum skew within write data group (ie, K group)	<p>The maximum skew between D and BWS signals referenced by a common K signal.</p>
Maximum skew within read data group (ie, CQ group)	<p>The maximum skew between Q signals referenced by a common CQ signal.</p>
Maximum skew between CQ groups	<p>The maximum skew between CQ signals of different read data groups.</p>
Maximum skew within address/command bus	<p>The maximum skew between the address/command signals.</p>

Table 8–28. Board Skew Parameters (Part 4 of 6)

Parameter	Description
<p>Average delay difference between address/command and K</p>	<p>A value equal to the average of the longest and smallest address/command signal delay values, minus the delay of the K signal. The value can be positive or negative.</p> <p>The average delay difference between the address and command and K is expressed by the following equation:</p> $\frac{\sum_{n=1}^n \left[\left(\frac{\text{Longest AC Path Delay} + \text{Shortest AC Path Delay}}{2} \right) - K_n \text{PathDelay} \right]}{n}$ <p>where n is the number of K clocks.</p>
<p>Average delay difference between write data signals and K</p>	<p>A value equal to the average of the longest and smallest write data signal delay values, minus the delay of the K signal. Write data signals include the D and BWS signals. The value can be positive or negative.</p> <p>The average delay difference between D and K is expressed by the following equation:</p> $\frac{\sum_{n=1}^n \left[\left(\frac{\text{Longest D Path Delay in } K_n \text{ group} + \text{Shortest D Path Delay in } K_n \text{ group}}{2} \right) - K_n \text{PathDelay} \right]}{n}$ <p>where n is the number of DQS groups.</p>
<p>Average delay difference between read data signals and CQ</p>	<p>A value equal to the average of the longest and smallest read data signal delay values, minus the delay of the CQ signal. The value can be positive or negative.</p> <p>The average delay difference between Q and CQ is expressed by the following equation:</p> $\frac{\sum_{n=1}^n \left[\left(\frac{\text{Longest Q Path Delay in } CQ_n \text{ group} + \text{Shortest Q Path Delay in } CQ_n \text{ group}}{2} \right) - CQ_n \text{PathDelay} \right]}{n}$ <p>where n is the number of CQ groups.</p>
<p>RLDRAM II</p>	
<p>Maximum CK delay to device</p>	<p>The delay of the longest CK trace from the FPGA to any device/DIMM is expressed by the following equation:</p> $\max_n(CK_n \text{PathDelay})$ <p>where n is the number of memory clocks. For example, the maximum CK delay for two pairs of memory clocks is expressed by the following equation:</p> $\max_2(CK_1 \text{PathDelay}, CK_2 \text{PathDelay})$
<p>Maximum DK delay to device</p>	<p>The delay of the longest DK trace from the FPGA to any device/DIMM is expressed by the following equation:</p> $\max_n(DK_n \text{PathDelay})$ <p>where n is the number of DK. For example, the maximum DK delay for two DK is expressed by the following equation:</p> $\max_2(DK_1 \text{PathDelay}, DK_2 \text{PathDelay})$

Table 8-28. Board Skew Parameters (Part 5 of 6)

Parameter	Description
Minimum delay difference between CK and DK	<p>The minimum delay difference between the CK signal and any DK signal when arriving at the memory device(s). The value is equal to the minimum delay of the CK signal minus the maximum delay of the DK signal. The value can be positive or negative.</p> <p>The minimum delay difference between CK and DK is expressed by the following equations:</p> $\min_{n,m}(CK_n PathDelay - DK_m PathDelay)$ <p>where n is the number of memory clocks and m is the number of DK. For example, the minimum delay difference between CK and DK for two pairs of memory clocks and four DK signals (two DK signals for each clock) is expressed by the following equation:</p> $\min_{2,2} \{(Ck_1 Delay - DK_1 Delay), (Ck_1 Delay - DK_2 Delay), (Ck_2 Delay - DK_3 Delay), (Ck_2 Delay - DK_4 Delay)\}$
Maximum delay difference between CK and DK	<p>The maximum delay difference between the CK signal and any DK signal when arriving at the memory device(s). The value is equal to the maximum delay of the CK signal minus the minimum delay of the DK signal. The value can be positive or negative.</p> <p>The maximum delay difference between CK and DK is expressed by the following equations:</p> $\max_{n,m}(CK_n PathDelay - DK_m PathDelay)$ <p>where n is the number of memory clocks and m is the number of DK. For example, the maximum delay difference between CK and DK for two pairs of memory clocks and four DK signals (two DK signals for each clock) is expressed by the following equation:</p> $\max_{2,2} \{(Ck_1 Delay - DK_1 Delay), (Ck_1 Delay - DK_2 Delay), (Ck_2 Delay - DK_3 Delay), (Ck_2 Delay - DK_4 Delay)\}$
Maximum delay difference between devices	<p>The maximum delay difference of data signals between devices is expressed by the following equation:</p> $Abs \left[\left(\frac{Longest\ device\ 1\ delay - Shortest\ device\ 1\ delay}{2} \right) - \left(\frac{Longest\ device\ 2\ delay - Shortest\ device\ 2\ delay}{2} \right) \right]$ <p>For example, in a two-device configuration there is greater propagation delay for data signals going to and returning from the furthest device relative to the nearest device. This parameter is applicable for depth expansion. Set the value to 0 for non-depth expansion design.</p>
Maximum skew within QK group	The maximum skew between the DQ signals referenced by a common QK signal.
Maximum skew between QK groups	The maximum skew between QK signals of different data groups.
Maximum skew within address/command bus	The maximum skew between the address/command signals.

Table 8–28. Board Skew Parameters (Part 6 of 6)

Parameter	Description
<p>Average delay difference between address/command and CK</p>	<p>A value equal to the average of the longest and smallest address/command signal delay values, minus the delay of the CK signal. The value can be positive or negative.</p> <p>The average delay difference between the address and command and CK is expressed by the following equation:</p> $\frac{\sum_{n=1}^n \left[\left(\frac{\text{Longest AC Path Delay} + \text{Shortest AC Path Delay}}{2} \right) - \text{CK}_n \text{ Path Delay} \right]}{n}$ <p>where n is the number of memory clocks.</p>
<p>Average delay difference between write data signals and DK</p>	<p>A value equal to the average of the longest and smallest write data signal delay values, minus the delay of the DK signal. Write data signals include the DQ and DM signals. The value can be positive or negative.</p> <p>The average delay difference between DQ and DK is expressed by the following equation:</p> $\frac{\sum_{n=1}^n \left[\left(\frac{\text{Longest DQ Path Delay in DK}_n \text{ group} + \text{Shortest DQ Path Delay in DK}_n \text{ group}}{2} \right) - \text{DK}_n \text{ Path Delay} \right]}{n}$ <p>where n is the number of DK groups.</p>
<p>Average delay difference between read data signals and QK</p>	<p>A value equal to the average of the longest and smallest read data signal delay values, minus the delay of the QK signal. The value can be positive or negative.</p> <p>The average delay difference between DQ and QK is expressed by the following equation:</p> $\frac{\sum_{n=1}^n \left[\left(\frac{\text{Longest DQ Path Delay in QK}_n \text{ group} + \text{Shortest DQ Path Delay in QK}_n \text{ group}}{2} \right) - \text{QK}_n \text{ Path Delay} \right]}{n}$ <p>where n is the number of QK groups.</p>

Controller Settings

Use this tab to apply the controller settings suitable for your design.


 This section describes parameters for the High Performance Controller II (HPC II) with advanced features introduced in version 11.0 for designs generated in version 11.0. Designs created in earlier versions and regenerated in version 11.0 do not inherit the new advanced features; for information on parameters for HPC II without the version 11.0 advanced features, refer to the External Memory Interface Handbook for Quartus II version 10.1, available on the [Literature: External Memory Interfaces](#) page of the Altera website.

Table 8-29 lists the controller settings.

Table 8-29. Controller Settings

Parameter	Description	
DDR2/DDR3 SDRAM		
Avalon Interface	Generate power-of-2 bus widths for SOPC Builder	Rounds down the Avalon-MM side data bus to the nearest power of 2. You must enable this option for both Qsys and SOPC Builder systems.
	Generate SOPC Builder compatible resets	You must enable this option if the IP core is to be used in an SOPC Builder system. When turned on, the reset inputs become associated with the PLL reference clock and the paths must be cut. This option must be enabled for SOPC Builder, but is not required when using the MegaWizard Plug-in Manager or Qsys.
	Maximum Avalon-MM burst length	Specifies the maximum burst length on the Avalon-MM bus. Affects the <code>AVL_SIZE_WIDTH</code> parameter.
	Enable Avalon-MM byte-enable signal	When you turn on this option, the controller adds the byte enable signal (<code>avl_be</code>) for the Avalon-MM bus to control the data mask (<code>mem_dm</code>) pins going to the memory interface. You must also turn on Enable DM pins if you are turning on this option. When you turn off this option, the byte enable signal (<code>avl_be</code>) is not enabled for the Avalon-MM bus, and by default all bytes are enabled. However, if you turn on Enable DM pins with this option turned off, all write words are written.
	Avalon interface address width	The address width on the Avalon-MM interface.
	Avalon interface data width	The data width on the Avalon-MM interface.
Low Power Mode	Enable self-refresh controls	Enables the self-refresh signals on the controller top-level design. These controls allow you to control when the memory is placed into self-refresh mode.
	Enable auto-power down	Allows the controller to automatically place the memory into power-down mode after a specified number of idle cycles. Specifies the number of idle cycles after which the controller powers down the memory in the auto-power down cycles parameter.
	Auto power-down cycles	The number of idle controller clock cycles after which the controller automatically powers down the memory. The legal range is from 1 to 65,535 controller clock cycles.

Table 8–29. Controller Settings

Parameter		Description
Efficiency	Enable user auto-refresh controls	Enables the user auto-refresh control signals on the controller top level. These controller signals allow you to control when the controller issues memory autorefresh commands.
	Enable auto-precharge control	Enables the autoprecharge control on the controller top level. Asserting the autoprecharge control signal while requesting a read or write burst allows you to specify whether the controller should close (autoprecharge) the currently open page at the end of the read or write burst.
	Local-to-memory address mapping	Allows you to control the mapping between the address bits on the Avalon-MM interface and the chip, row, bank, and column bits on the memory. Select Chip-Row-Bank-Col to improve efficiency with sequential traffic . Select Chip-Bank-Row-Col to improve efficiency with random traffic. Select Row-Chip-Bank-Col to improve efficiency with multiple chip select and sequential traffic.
	Command queue look-ahead depth	Selects a look-ahead depth value to control how many read or writes requests the look-ahead bank management logic examines. Larger numbers are likely to increase the efficiency of the bank management, but at the cost of higher resource usage. Smaller values may be less efficient, but also use fewer resources. The valid range is from 1 to 16.
	Enable reordering	Allows the controller to perform command and data reordering that reduces bus turnaround time and row/bank switching time to improve controller efficiency.
	Starvation limit for each command	Specifies the number of commands that can be served before a waiting command is served. The valid range is from 1 to 63.
Configuration, Status, and Error Handling	Enable Configuration and Status Register Interface	Enables run-time configuration and status interface for the memory controller. This option adds an additional Avalon-MM slave port to the memory controller top level, which you can use to change or read out the memory timing parameters, memory address sizes, mode register settings and controller status. If Error Detection and Correction Logic is enabled, the same slave port also allows you to control and retrieve the status of this logic.
	CSR port host interface	Specifies the type of connection to the CSR port. The port can be exported, internally connected to a JTAG Avalon Master, or both. Select Internal (JTAG) to export the CSR port. Select Avalon-MM Slave to connect the CSR port to a JTAG Avalon Master. Select Shared to export and connect the CSR port to a JTAG Avalon Master.
	Enable error detection and correction logic	Enables ECC for single-bit error correction and double-bit error detection. Your memory interface must be a multiple of 40 or 72 bits wide to use ECC.
	Enable auto error correction	Allows the controller to perform auto correction when a single-bit error is detected by the ECC logic.
Advanced Controller Features	Enable half rate bridge	Turn on this option to enable half rate bridge block.
	Enable hard memory controller	Turn on this option to enable hard memory controller.

Table 8–29. Controller Settings

Parameter		Description
Multiple Port Front End	Export bonding port	Turn on this option to export bonding interface for wider avalon data width with two controllers. Bonding ports are exported to the top level.
	Number of ports	Specifies the number of Avalon-MM Slave ports to be exported. The number of ports depends on the width and the type of port you selected. There are four 64-bit read FIFOs and four 64-bit write FIFOs in the multi-port front-end (MPFE) component. For example, If you select 256 bits width and bidirectional slave port, all the FIFOs are fully utilized, therefore you can only select one port.
	Width	Specifies the local data width for each Avalon-MM Slave port. The width depends on the type of slave port and also the number of ports selected. This is due to the limitation of the FIFO counts in the MPFE. There are four 64-bit read FIFOs and four 64-bit write FIFOs in the MPFE. For example, if you select one bidirectional slave port, you can select up to 256 bits to utilize all the read and write FIFOs.
	Priority	Specifies the absolute priority for each Avalon-MM Slave port. Any transaction from the port with higher priority number will be served before transactions from the port with lower priority number.
	Weight	Specifies the relative priority for each Avalon-MM Slave port. When there are two or more ports having the same absolute priority, the transaction from the port with higher (bigger number) relative weight will be served first. You can set the weight from a range of 0 to 32.
	Type	Specifies the type of Avalon MM slave port to either a bidirectional port, read only port or write only port.
QDR II/QDR II+ SRAM and RLDRAM II		
Generate power-of-2 data bus widths for SOPC Builder		This option must be enabled if this core is to be used in an SOPC Builder system. When turned on, the Avalon-MM side data bus width is rounded down to the nearest power of 2.
Generate SOPC Builder compatible resets		This option must be enabled if this core is to be used in an SOPC Builder system.
Maximum Avalon-MM burst length		Specifies the maximum burst length on the Avalon-MM bus.
Enable Avalon-MM byte-enable signal		When you turn on this option, the controller adds a byte-enable signal (<code>avl_be_w</code>) for the Avalon-MM bus, in which controls the <code>bws_n</code> signal on the memory side to mask bytes during write operations. When you turn off this option, the <code>avl_be_w</code> signal is not available and the controller will always drive the memory <code>bws_n</code> signal so as to not mask any bytes during write operations.
Avalon interface address width		Specifies the address width on the Avalon-MM interface.
Avalon interface data width		Specifies the data width on the Avalon-MM interface.
Reduce controller latency by		Specifies the number of clock cycles by which to reduce controller latency. Lower controller latency results in lower resource usage and f_{MAX} while higher latency results in higher resource usage and f_{MAX} .
Enable user refresh		Enables user-controlled refresh. Refresh signals will have priority over read/write requests. This option is available for RLDRAM II only.
Enable error detection parity		Enables per-byte parity protection. This option is available for RLDRAM II only

Diagnostics

The **Diagnostics** tab allows you to set parameters for certain diagnostic functions.

Table 8–30 describes parameters for simulation.

Table 8–30. Simulation Options

Parameter	Description
Simulation Options	
Auto-calibration mode	<p>Specifies whether you want to improve simulation performance by reducing calibration. There is no change to the generated RTL. The following autocalibration modes are available:</p> <ul style="list-style-type: none"> ■ Skip calibration—provides the fastest simulation. It loads the settings calculated from the memory configuration and enters user mode. ■ Quick calibration—calibrates (without centering) one bit per group before entering user mode. ■ Full calibration—calibrates the same as in hardware, and includes all phases, delay sweeps, and centering on every data bit. You can use timing annotated memory models. Be aware that full calibration can take hours or days to complete. <p>To perform proper PHY simulation, select Quick calibration or Full calibration. For more information, refer to the “Simulation Options” section in the <i>Simulating Memory IP</i> chapter.</p> <p>For QDR II, QDR II+ SRAM, and RLDRAM II, the Nios II-based sequencer must be selected to enable the auto calibration modes selection.</p>
Skip memory initialization delays	When you turn on this option, required delays between specific memory initialization commands are skipped to speed up simulation.
Enable verbose memory model output	Turn on this option to display more detailed information about each memory access during simulation.
Enable support for Nios II ModelSim® flow in Eclipse	<p>Initializes the memory interface for use with the Run as Nios II ModelSim flow with Eclipse.</p> <p>This option is not available for QDR II and QDR II+ SRAM.</p>
Debug Options	
Debug level	Specifies the debug level of the memory interface.
Efficiency Monitor and Protocol Checker Settings	
Enable the Efficiency Monitor and Protocol Checker on the Controller Avalon Interface	<p>Enables efficiency monitor and protocol checker block on the controller Avalon interface.</p> <p>This option is not available for QDR II and QDR II+ SRAM.</p>

Document Revision History

Table 8–31 lists the revision history for this document.

Table 8–31. Document Revision History

Date	Version	Changes
November 2011	4.0	<ul style="list-style-type: none"> ■ Updated Installation and Licensing section. ■ Combined Qsys and SOPC Builder Interfaces sections. ■ Combined parameter settings for DDR, DDR2, DDR3 SDRAM, QDRII SRAM, and RLDRAM II for both ALTMEMPHY and UniPHY IP. ■ Added parameter usage details to Parameterizing Memory Controllers with UniPHY IP section. ■ Moved “Functional Description” section for DDR, DDR2, DDR3 SDRAM, QDRII SRAM, and RLDRAM II to volume 3 of the <i>External Memory Interface Handbook</i>.
June 2011	3.0	<ul style="list-style-type: none"> ■ Removed references to High-Performance Controller. ■ Updated High-Performance Controller II information. ■ Removed HardCopy III, HardCopy IV E, HardCopy IV GX, Stratix III, and Stratix IV support. ■ Updated Generated Files lists. ■ Added Qsys and SOPC Builder Interfaces section.
December 2010	2.1	<p>Updated the following items for 10.1:</p> <ul style="list-style-type: none"> ■ Updated Design Flows and Generated Files information. ■ Updated Parameterizing Memory Controllers with UniPHY IP chapter.
July 2010	2.0	<ul style="list-style-type: none"> ■ Added information for new GUI parameters: Controller latency, Enable reduced bank tracking for area optimization, and Number of banks to track. ■ Removed information about IP Advisor. This feature is removed from the DDR/DDR2 SDRAM IP support for version 10.0.
February 2010	1.3	Corrected typos.
February 2010	1.2	<ul style="list-style-type: none"> ■ Full support for Stratix IV devices. ■ Added timing diagrams for initialization and calibration stages for HPC.
November 2009	1.1	Minor corrections.
November 2009	1.0	First published.