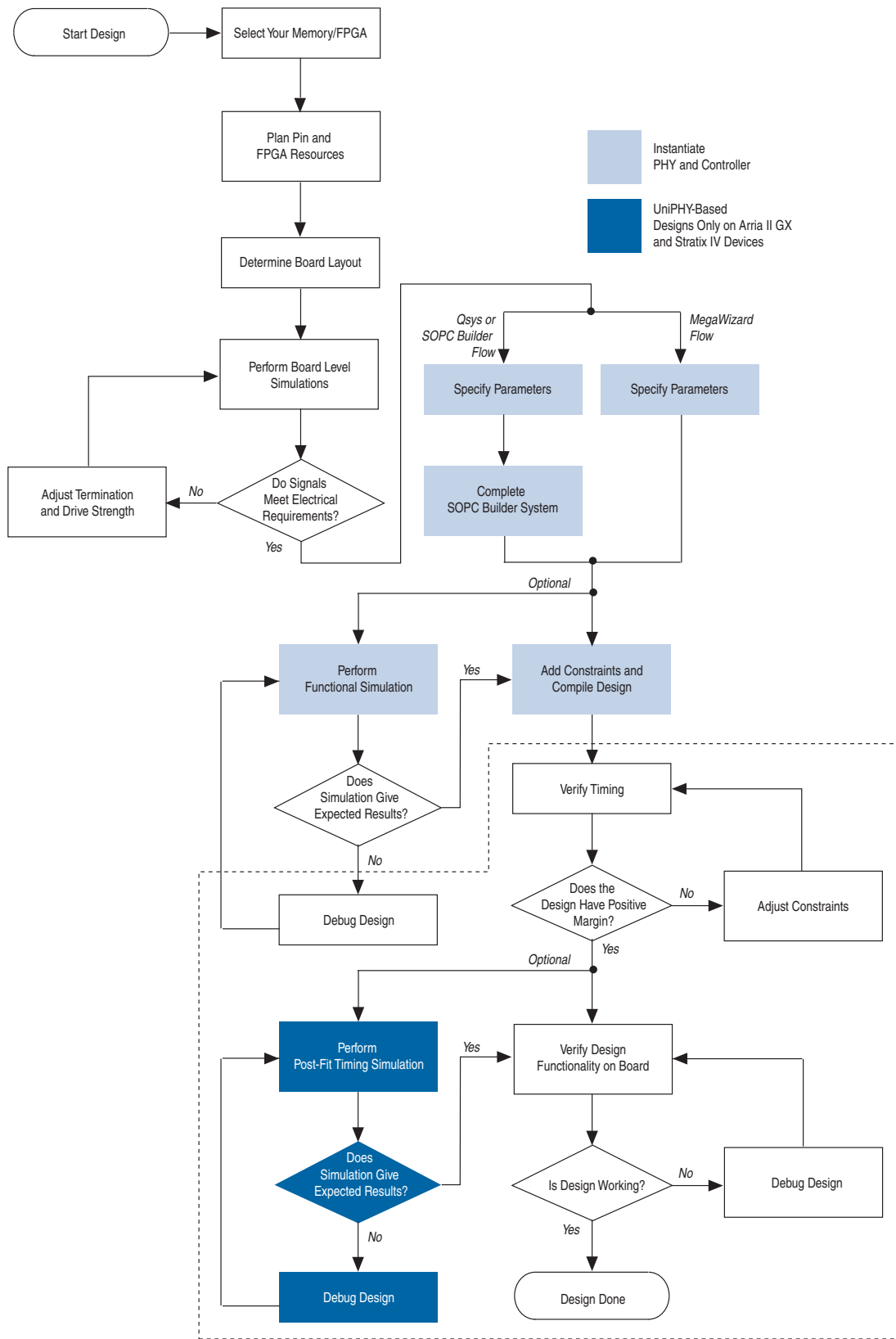


This chapter describes the Altera-recommended design flow for successfully implementing external memory interfaces in Altera® devices. Altera recommends that you create an example top-level file with the desired pin outs and all interface IP instantiated, which enables the Quartus® II software to validate your design and resource allocation before PCB and schematic sign off. Use the “Design Checklist” on page 2–5, to verify whether you have performed all the recommended steps in creating a working and robust external memory interface.


Figure 2–1 shows the design flow to provide the fastest out-of-the-box experience with external memory interfaces in Altera devices. This topic directs you where to find information on how to perform each step of the recommended design flow. The flow assumes that you are using Altera IP to implement the external memory interface.

Figure 2-1. External Memory Interfaces Design Flowchart




## Select Your Memory

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

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

## Select Your FPGA

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

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


## Planning Pin and FPGA Resources

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

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

## Determine Board Layout

To improve the signal integrity, you have to consider the termination scheme that you use, the drive strength setting on the FPGA, and the loading seen by the driver. You must understand the tradeoffs between the different types of termination schemes and the effects of output drive strengths and loading, to choose the best possible settings for your designs.


-  For more information about guidelines to determine your board layout for the different memory controllers, refer to the following chapters in the *External Memory Interface Handbook*:

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

## Implementing and Parameterizing Memory IP


After selecting the appropriate device and memory type, create a project in the Quartus II software that targets the device and memory type.

When implementing and parameterizing external memory interfaces, Altera recommends that you use Altera memory interface IP, which includes a PHY that you can use with the Altera high-performance controller or with your own custom controller.

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


## Simulating Memory IP

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

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


## Analyzing Timing of Memory IP

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

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

## Perform Post-Fit Timing Simulation

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

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

## Debugging Memory IP

You need to perform system level verification to correlate the system against your design targets, using the Altera SignalTap<sup>®</sup> II logic analyzer.

-  For more information about using the SignalTap II analyzer, refer to the *Debugging Memory IP* chapter in the *External Memory Interface Handbook*.

## Design Checklist

This topic contains a design checklist that you can use when implementing external memory interfaces in Altera devices.

Done	Action	Reference
	<b>Select Your Memory</b>	
1.	<input type="checkbox"/> Select the memory interface frequency of operation and bus width.	■ <a href="#">Selecting Your Memory</a> chapter in the <i>External Memory Interface Handbook</i> .
	<b>Select Your FPGA</b>	
2.	<input type="checkbox"/> Select the FPGA device density and package combination that you want to target.	■ <a href="#">Selecting Your FPGA</a> chapter in the <i>External Memory Interface Handbook</i> .
	<b>Plan Pin and FPGA Resources</b>	
3.	<input type="checkbox"/> Ensure that the target FPGA device supports the desired clock rate and memory bus width. Also the FPGA must have sufficient I/O pins for the DQ/DQS read and write groups.	For detailed device resource information, refer to the relevant device handbook chapter on external memory interface support.
	<b>Determine Board Layout</b>	
4.	<input type="checkbox"/> Select the termination scheme and drive strength settings for all the memory interface signals on the memory side and the FPGA side.	■ <a href="#">DDR2 and DDR3 SDRAM Board Design Guidelines</a> chapter in the <i>External Memory Interface Handbook</i> .
5.	<input type="checkbox"/> Ensure you apply appropriate termination and drive strength settings on all the memory interface signals, and verify using board level simulations.	■ <a href="#">Dual-DIMM DDR2 and DDR3 SDRAM Board Design Guidelines</a> chapter in the <i>External Memory Interface Handbook</i> .
6.	<input type="checkbox"/> Use board level simulations to pick the optimal setting for best signal integrity. On the memory side, Altera recommends the use of external parallel termination on input signals to the memory (write data, address, command, and clock signals).	■ <a href="#">RLDRAM II Board Design Guidelines</a> chapter in the <i>External Memory Interface Handbook</i> .
7.	<input type="checkbox"/> Perform board level simulations, to ensure electrical and timing margins for your memory interface	■ <a href="#">QDR II SRAM Board Design Guidelines</a> chapter in the <i>External Memory Interface Handbook</i> .
8.	<input type="checkbox"/> Ensure you have a sufficient eye opening using simulations. Use the latest FPGA and memory IBIS models, board trace characteristics, drive strength, and termination settings in your simulation.  Any timing uncertainties at the board level that you calculate using simulations must be used to adjust the input timing constraints to ensure the accuracy of Quartus II timing margin reports. For example crosstalk, ISI, and slew rate deration.	
	<b>Parameterize and Implement the Memory IP</b>	
9.	<input type="checkbox"/> Parameterize and instantiate the Altera external memory IP for your target memory interface.	■ <a href="#">Implementing and Parameterizing Memory IP</a> chapter in the <i>External Memory Interface Handbook</i>

	Done	Action	Reference
10.	<input type="checkbox"/>	Ensure that you perform the following actions: <ul style="list-style-type: none"> <li>■ Pick the correct memory interface data rates, width, and configurations.</li> <li>■ For DDR, DDR2, and DDR3 SDRAM interfaces, ensure that you derate the tIS, tIH, tDS, and tDH parameters, as necessary.</li> <li>■ Include the board skew parameters for your board.</li> </ul>	
11.	<input type="checkbox"/>	Connect the PHY's local signals to your driver logic and the PHY's memory interface signals to top-level pins. Ensure that the local interface signals of the PHY are appropriately connected to your own logic. If the ALTMEMPHY IP is compiled without these local interface connections, you may encounter compilation problems, when the number of signals exceeds the pins available on your target device. You may also use the example top-level file as an example on how to connect your own custom controller to the Altera memory PHY.	<ul style="list-style-type: none"> <li>■ <i>Functional Description: HPC II</i> chapter in the <i>External Memory Interface Handbook</i>.</li> <li>■ <i>Functional Description: QDR II and QDR II+ SRAM Controller</i> chapter in the <i>External Memory Interface Handbook</i>.</li> <li>■ <i>Functional Description: RLDRAM II Controller</i> chapter in the <i>External Memory Interface Handbook</i>.</li> </ul>
12.	<input type="checkbox"/>	<p style="text-align: center;"><b>Perform Functional Simulation</b></p> Simulate your design using the RTL functional model. Use the IP functional simulation model with your own driver logic, testbench, and a memory model, to ensure correct read and write transactions to the memory. You may need to prepare the memory functional model by setting the speed grade and device bus mode.	<ul style="list-style-type: none"> <li>■ <i>Simulating Memory IP</i> chapter in the <i>External Memory Interface Handbook</i></li> </ul>
13.	<input type="checkbox"/>	<p style="text-align: center;"><b>Add Constraints</b></p> Add timing constraints. The wizard-generated <b>.sdc</b> file adds timing constraints to the interface. However, you may need to adjust these settings to best fit your memory interface configuration.	
14.	<input type="checkbox"/>	Add pin settings and DQ group assignments. The wizard-generated <b>.tcl</b> file includes I/O standard and pin loading constraints to your design.	
15.	<input type="checkbox"/>	Ensure that generic pin names used in the constraint scripts are modified to match your top-level pin names. The loading on memory interface pins is dependent on your board topology (memory components).	
16.	<input type="checkbox"/>	Add pin location assignments. However, you need to assign the pin location assignments manually using the Pin Planner.	
17.	<input type="checkbox"/>	Ensure that the example top-level file or your top-level logic is set as top-level entity.	

	Done	Action	Reference
18.	<input type="checkbox"/>	Adjust optimization techniques, to ensure the remaining unconstrained paths are routed with the highest speed and efficiency: <ol style="list-style-type: none"> <li>On the Assignments menu click <b>Settings</b>.</li> <li>Select <b>Analysis &amp; Synthesis Settings</b>.</li> <li>Select <b>Speed</b> under <b>Optimization Technique</b>.</li> <li>Expand <b>Fitter Settings</b>.</li> <li>Turn on <b>Optimize Hold Timing</b> and select <b>All Paths</b>.</li> <li>Turn on <b>Optimize Fast Corner Timing</b>.</li> <li>Select <b>Standard Fit</b> under <b>Fitter Effort</b>.</li> </ol>	
19.	<input type="checkbox"/>	Provide board trace delay model. For accurate I/O timing analysis, you specify the board trace and loading information in the Quartus II software. This information should be derived and refined during your board development process of prelayout (line) simulation and finally post-layout (board) simulation. Provide the board trace information for the output and bidirectional pins through the board trace model in the Quartus II software.	
<b>Compile Design and Verify Timing</b>			
20.	<input type="checkbox"/>	Compile your design and verify timing closure using all available models.	
21.	<input type="checkbox"/>	Run the wizard-generated <code>&lt;variation_name&gt;_report_timing.tcl</code> file, to generate a custom timing report for each of your IP instances. Run this process across all device timing models (slow 0°C, slow 85°C, fast 0°C).	
22.	<input type="checkbox"/>	If there are timing violations, adjust your constraints to optimize timing	
23.	<input type="checkbox"/>	As required, adjust PLL clock phase shift settings or appropriate timing and location assignments margins for the various timing paths within the IP.	<ul style="list-style-type: none"> <li>■ <a href="#">Analyzing Timing of Memory IP</a> chapter in the <i>External Memory Interface Handbook</i></li> </ul>
<b>Perform Post-Fit Timing Simulation</b>			
24.	<input type="checkbox"/>	Perform post-fit timing simulation to ensure that all the memory transactions meet the timing specifications with the vendor's memory model.	<ul style="list-style-type: none"> <li>■ <a href="#">Simulating Memory IP</a> chapter in the <i>External Memory Interface Handbook</i>.</li> </ul>
<b>Verify Design Functionality</b>			
25.	<input type="checkbox"/>	Verify the functionality of your memory interface in the system	<ul style="list-style-type: none"> <li>■ <a href="#">Debugging Memory IP</a> chapter in the <i>External Memory Interface Handbook</i></li> </ul>

## Document Revision History

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

**Table 2-1. Document Revision History**

<b>Date</b>	<b>Version</b>	<b>Changes</b>
November 2011	2.1	Updated the design flow and the design checklist.
July 2010	2.0	Updated for 10.0 release.
January 2010	1.1	<ul style="list-style-type: none"><li>■ Improved description for <i>Implementing Altera Memory Interface IP</i> chapter.</li><li>■ Added timing simulation to flow chart and to design checklist.</li></ul>
November 2009	1.0	First published.