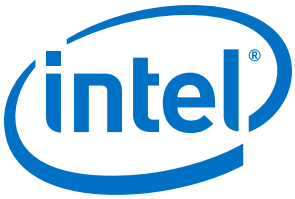


External Memory Interfaces Intel® Agilex™ FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.3**

UG-20219 | 2019.10.18
Latest document on the web: **PDF** | **HTML**

Subscribe
Send Feedback

Contents

1. About the External Memory Interfaces Intel® Agilex™ FPGA IP

1.1. Release Information

IP versions are the same as the Intel® Quartus® Prime Design Suite software versions up to v19.1. From Intel Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme.

The IP versioning scheme (X.Y.Z) number changes from one software version to another. A change in:

- X indicates a major revision of the IP. If you update your Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Item	Description
IP Version	1.2.0
Intel Quartus Prime	19.3
Release Date	2019.09.30

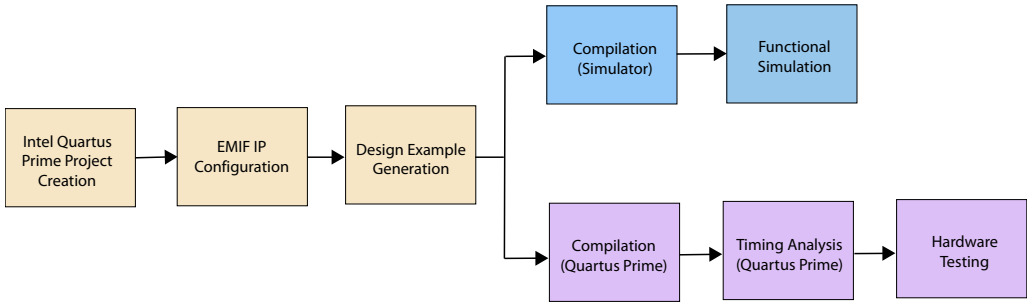
2. Design Example Quick Start Guide for External Memory Interfaces Intel Agilex™ FPGA IP

An automated design example flow is available for Intel Agilex™ external memory interfaces.

The **Generate Example Designs** button on the **Example Designs** tab allows you to specify and generate the synthesis and/or simulation design example file sets which you can use to validate your EMIF IP.

You can generate a design example that matches the Intel FPGA development kit, or for any EMIF IP that you generate. You can use the design example to assist your evaluation, or as a starting point for your own system.

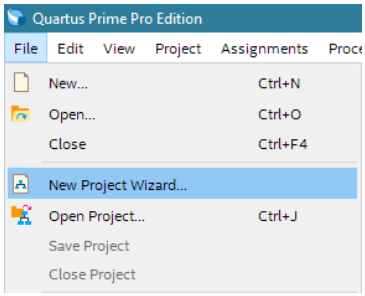
Figure 1. General Design Example Workflows



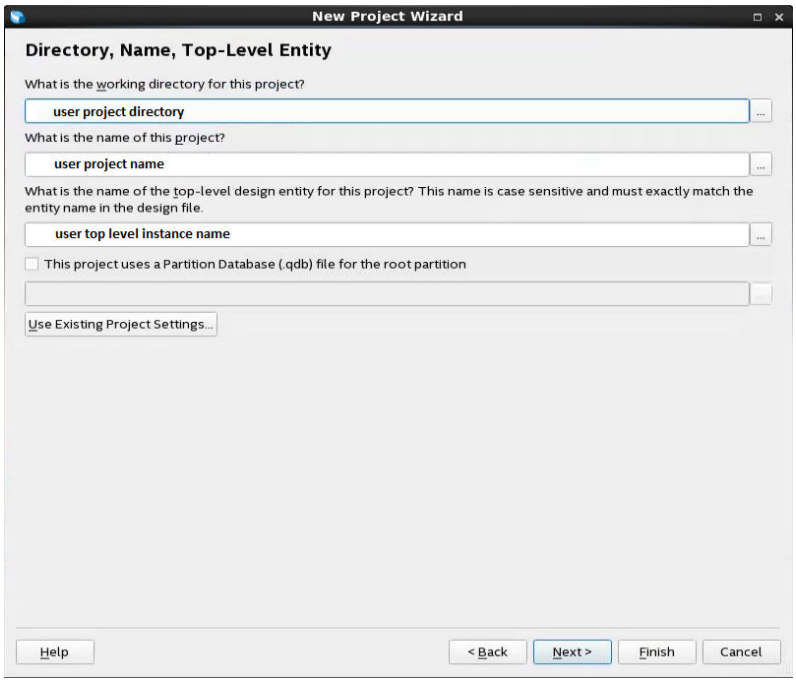
2.1. Creating an EMIF Project

For the Intel Quartus Prime software version 17.1 and later, you must create an Intel Quartus Prime project before generating the EMIF IP and design example.

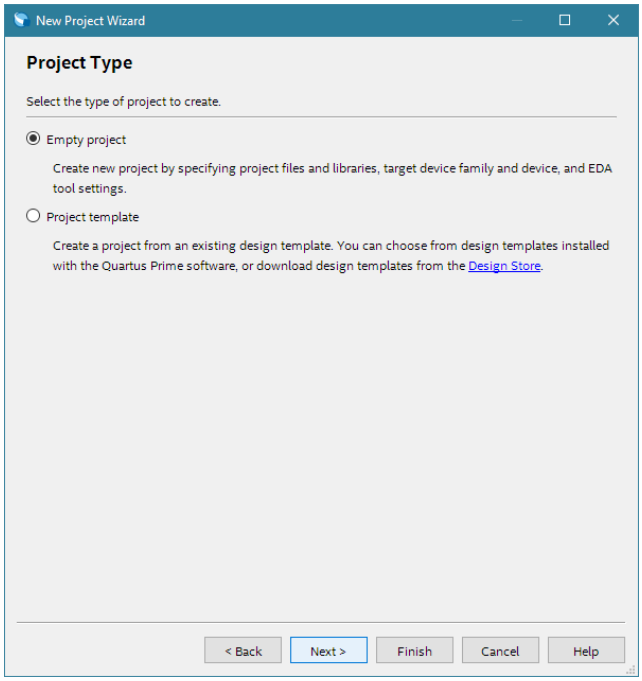
1. Launch the Intel Quartus Prime software and select **File ➤ New Project Wizard**. Click **Next**.



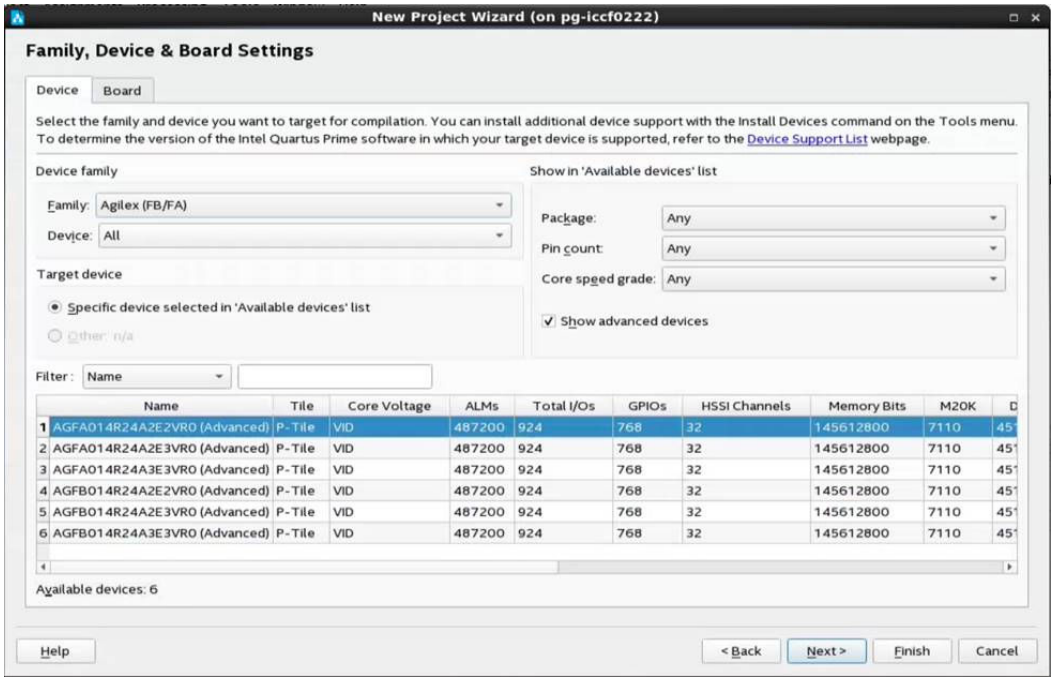
2. Specify a directory (<*user project directory*>), a name for the Intel Quartus Prime project (<*user project name*>), and a top-level design entity name (<*user top-level instance name*>) that you want to create. Click **Next**.



3. Verify that **Empty Project** is selected. Click **Next** two times.



4. Under **Family**, select Intel Agilex.
5. Under **Name filter**, type the device part number.
6. Under **Available devices**, select the appropriate device.



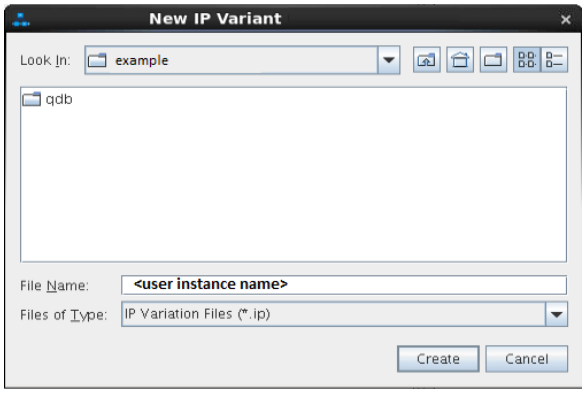
7. Click **Finish**.

2.2. Generating and Configuring the EMIF IP

The following steps illustrate how to generate and configure the EMIF IP. This walkthrough creates a DDR4 interface, but the steps are similar for other protocols. (These steps follow the **IP Catalog** (standalone) flow; if you choose to use the **Platform Designer** (system) flow instead, the steps are similar.)

1. In the **IP Catalog window**, select **External Memory Interfaces Intel Agilex FPGA IP**. (If the **IP Catalog window** is not visible, select **View ➤ IP Catalog**.)

2. In the **IP Parameter Editor**, provide an entity name for the EMIF IP (the name that you provide here becomes the file name for the IP) and specify a directory. Click **Create**.



3. The parameter editor has multiple tabs where you must configure parameters to reflect your EMIF implementation.

2.2.1. Intel Agilex EMIF Parameter Editor Guidelines

This topic provides high-level guidance for parameterizing the tabs in the Intel Agilex EMIF IP parameter editor.

Table 1. EMIF Parameter Editor Guidelines

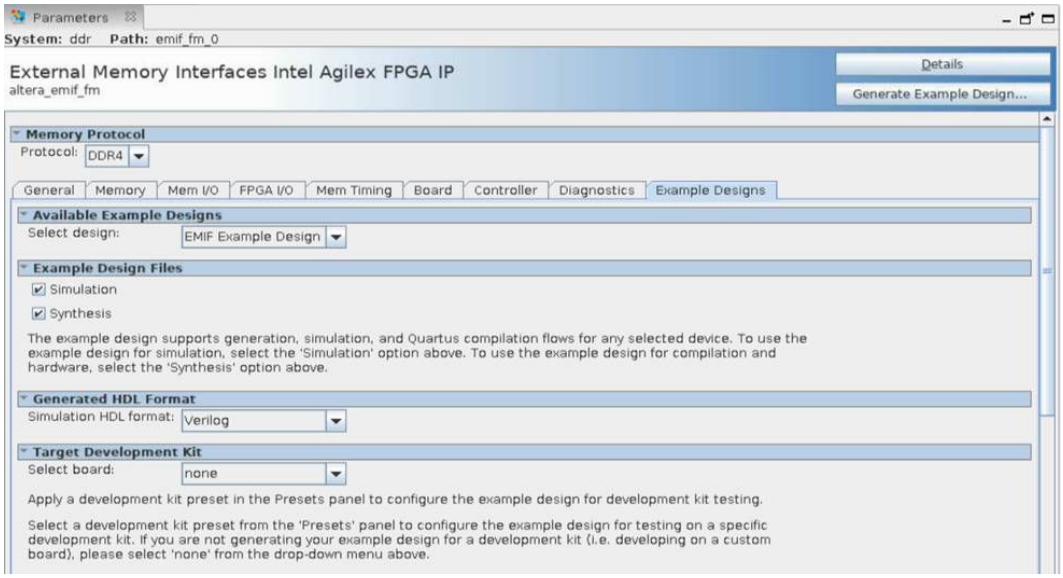
Parameter Editor Tab	Guidelines
General	Ensure that the following parameters are entered correctly: • The speed grade for the device. • The memory clock frequency. • The PLL reference clock frequency.
Memory	• Refer to the data sheet for your memory device to enter the parameters on the **Memory** tab. • You should also enter a specific location for the ALERT# pin. (Applies to DDR4 memory protocol only.)
Mem I/O	• For initial project investigations, you may use the default settings on the **Mem I/O** tab. • For advanced design validation, you should perform board simulation to derive optimal termination settings.
FPGA I/O	• For initial project investigations, you may use the default settings on the **FPGA I/O** tab. • For advanced design validation, you should perform board simulation with associated IBIS models to select appropriate I/O standards.
Mem Timing	• For initial project investigations, you may use the default settings on the **Mem Timing** tab. • For advanced design validation, you should enter parameters according to your memory device's data sheet.
Controller	Set the controller parameters according to the desired configuration and behavior for your memory controller.
Diagnostics	You can use the parameters on the **Diagnostics** tab to assist in testing and debugging your memory interface.
Example Designs	The **Example Designs** tab lets you generate design examples for synthesis and for simulation. The generated design example is a complete EMIF system consisting of the EMIF IP and a driver that generates random traffic to validate the memory interface.

For detailed information on individual parameters, refer to the appropriate chapter for your memory protocol in the *External Memory Interfaces Intel Agilex FPGA IP User Guide*.

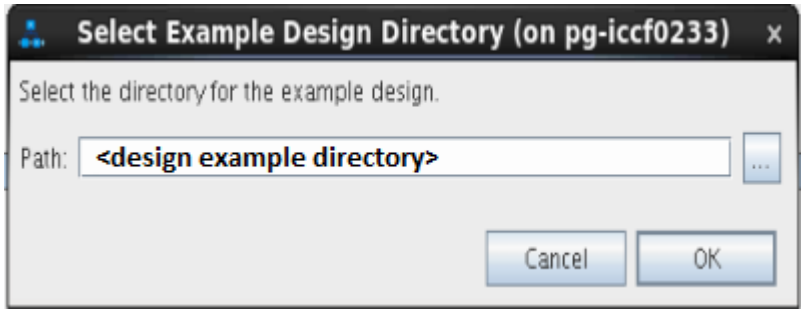
2.3. Generating the Synthesizable EMIF Design Example

For the Intel Agilex development kit, it is sufficient to leave most of the Intel Agilex EMIF IP settings at their default values. To generate the synthesizable design example, follow these steps:

1. On the **Example Designs** tab, ensure that the **Synthesis** box is checked.
2. Configure the EMIF IP and click **File ➤ Save** to save the current setting into the user IP vatiation file (<*user instance name*>.ip).
3. Click **Generate Example Design** in the upper-right corner of the window.



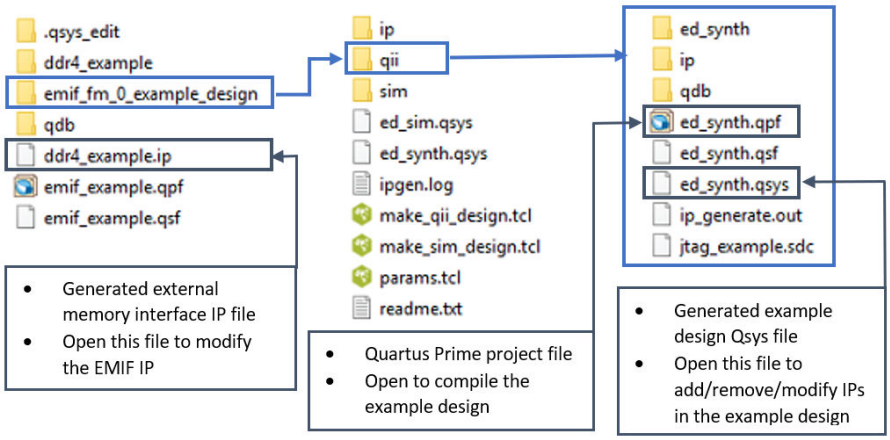
4. Specify a directory for the EMIF design example and click **OK**. Successful generation of the EMIF design example creates the following fileset under a `qii` directory.



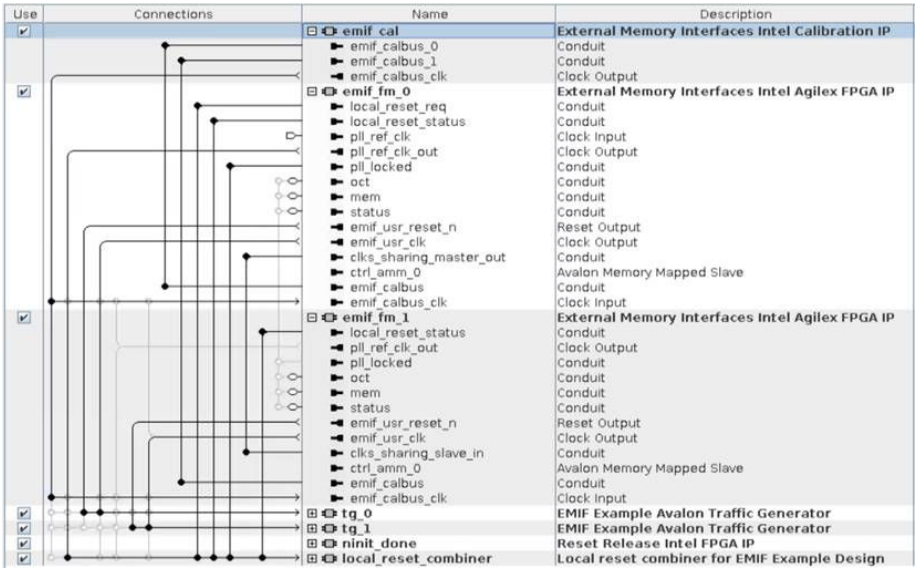
5. Click **File ➤ Exit** to exit the **IP Parameter Editor Pro** window. The system prompts, *Recent changes have not been generated. Generate now?* Click **No** to continue with the next flow.
6. To open the example design, click **File ➤ Open Project**, and navigate to the `<project_directory>/<design_example_name>/qii/ed_synth.qpf` and click **Open**.

   Note: For information on compiling and programming the design example, refer to Compiling and Programming the Intel Agilex EMIF Design Example.

Figure 2. Generated Synthesizable Design Example File Structure



7. If you are implementing an example design with multiple interfaces, you must share the External Memory Interface Intel Calibration IP with each interface and set the correct number of calibration interfaces that are connected to the calibration IP. Place all IP files for the design into a single design directory, so that the Platform Designer can locate the IP files.



Note: If you don't select the **Simulation** or **Synthesis** checkbox, the destination directory will contain Platform Designer design files, which are not compilable by the Intel Quartus Prime software directly, but can be viewed or edited under the Platform Designer. In this situation you can run the following commands to generate synthesis and simulation file sets.

- To create a compilable project, you must run the

  ```
  quartus_sh -t make_qii_design.tcl
  ```

  script in the destination directory.
- To create a simulation project, you must run the

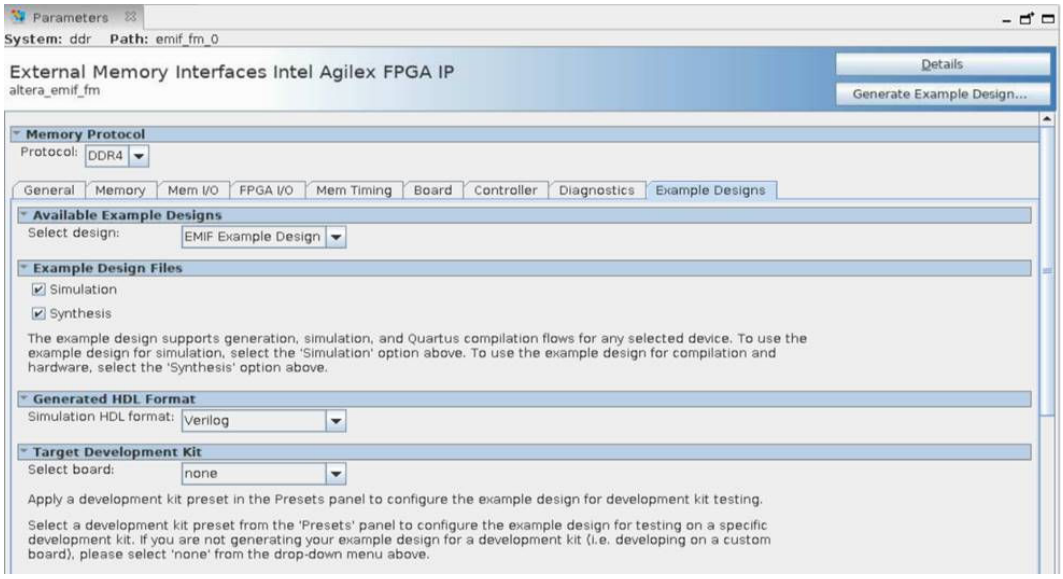
  ```
  quartus_sh -t make_sim_design.tcl
  ```

  script in the destination directory.

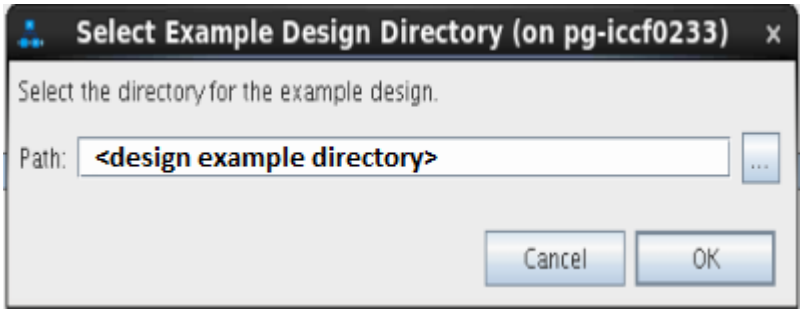
2.4. Generating the EMIF Design Example for Simulation

For the Intel Agilex development kit, it is sufficient to leave most of the Intel Agilex EMIF IP settings at their default values. To generate the design example for simulation, follow these steps:

1. On the **Example Designs** tab, ensure that the **Simulation** box is checked. Also choose the required **Simulation HDL format**, either Verilog or VHDL.
2. Configure the EMIF IP and click **File ➤ Save** to save the current setting into the user IP variation file (<*user instance name*>.ip).
3. Click **Generate Example Design** in the upper-right corner of the window.

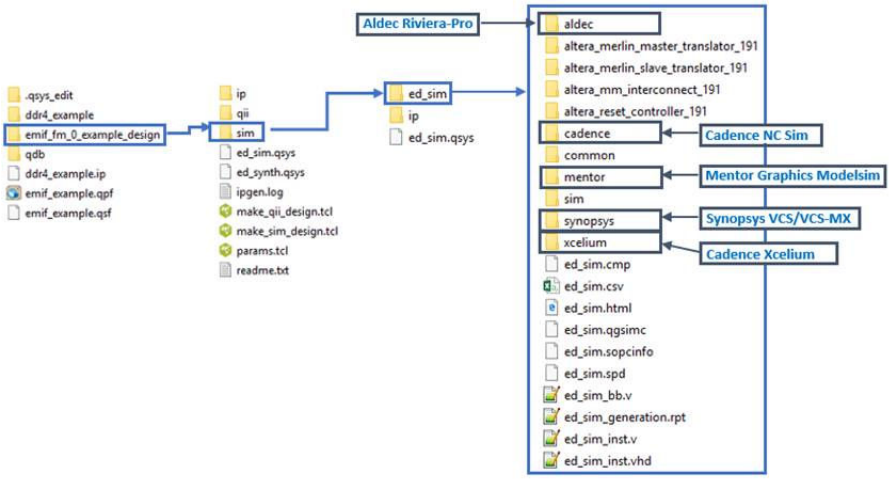


4. Specify a directory for the EMIF design example and click **OK**. Successful generation of the EMIF design example creates multiple file sets for various supported simulators, under a `sim/ed_sim` directory.



5. Click **File ➤ Exit** to exit the IP parameter editor to continue with the next flow.

Figure 3. Generated Simulation Design Example File Structure



Note: The External Memory Interfaces Intel Agilex FPGA IP currently supports only the VCS, ModelSim/QuestaSim, and Xcelium simulators. Additional simulator support will be available in future releases.

Note: If you don't select the **Simulation** or **Synthesis** checkbox, the destination directory will contain Platform Designer design files, which are not compilable by the Intel Quartus Prime software directly, but can be viewed or edited under the Platform Designer. In this situation you can run the following commands to generate synthesis and simulation file sets.

- To create a compilable project, you must run the

  ```
  quartus_sh -t make_qii_design.tcl
  ```

  script in the destination directory.
- To create a simulation project, you must run the

  ```
  quartus_sh -t make_sim_design.tcl
  ```

  script in the destination directory.

2.5. Simulation Versus Hardware Implementation

For external memory interface simulation, you can select either skip calibration or full calibration on the **Diagnostics** tab during IP generation.

EMIF Simulation Models

This table compares the characteristics of the skip calibration and full calibration models.

Table 2. EMIF Simulation Models: Skip Calibration versus Full Calibration

Skip Calibration	Full Calibration
System-level simulation focusing on user logic.	Memory interface simulation focusing on calibration.
Details of calibration are not captured.	Captures all stages of calibration.
Has ability to store and retrieve data.	Includes leveling, per-bit deskew, etc.
Represents accurate efficiency.	
Does not consider board skew.	

RTL Simulation Versus Hardware Implementation

This table highlights key differences between EMIF simulation and hardware implementation.

Table 3. EMIF RTL Simulation Versus Hardware Implementation

RTL Simulation	Hardware Implementation
Nios® initialization and calibration code execute in parallel.	Nios initialization and calibration code execute sequentially.
Interfaces assert `cal_done` signal signal simultaneously in simulation.	Fitter operations determine the order of calibration, and interfaces do not assert `cal_done` simultaneously.

You should run RTL simulations based on traffic patterns for your design's application. Note that RTL simulation does not model PCB trace delays which may cause a discrepancy in latency between RTL simulation and hardware implementation.

2.6. Simulating External Memory Interface IP With ModelSim

This procedure shows how to simulate the EMIF design example.

1. Launch the Mentor Graphics* ModelSim software and select **File ➤ Change Directory**. Navigate to the `sim/ed_sim/mentor` directory within the generated design example folder.
2. Verify that the **Transcript** window is displayed at the bottom of the screen. If the **Transcript** window is not visible, display it by clicking **View ➤ Transcript**.
3. In the **Transcript** window, run **source msim_setup.tcl**.
4. After **source msim_setup.tcl** finishes running, run **ld_debug** in the **Transcript** window.
5. After **ld_debug** finishes running, verify that the **Objects** window is displayed. If the **Objects** window is not visible, display it by clicking **View ➤ Objects**.

6. In the **Objects** window, select the signals that you want to simulate by right-clicking and selecting **Add Wave**.
7. After you finish selecting the signals for simulation, execute **run -all** in the **Transcript** window. The simulation runs until it is completed.
8. If the simulation is not visible, click **View ➤ Wave**.

2.7. Pin Placement for Intel Agilex EMIF IP

This topic provides guidelines for pin placement.

Overview

Intel Agilex FPGAs have the following structure:

- Each device contains up to 8 I/O banks.
- Each I/O bank contains 2 sub-I/O banks.
- Each sub-I/O bank contains 4 lanes.
- Each lane contains 12 general-purpose I/O (GPIO) pins.

General Pin Guidelines

The following are general pin guidelines:

- Ensure that the pins for a given external memory interface reside within a single I/O column.
- Interfaces that span multiple banks must meet the following requirements:
  - The banks must be adjacent to one another. For information on adjacent banks, refer to the *External Memory Interfaces Intel Agilex FPGA IP User Guide*.
- Unused pins can be used as general-purpose I/O pins.
- All address and command and associated pins must reside within a single bank.
- Address and command and data pins can share a bank under the following conditions:
  - Address and command and data pins cannot share an I/O lane.
  - Only an unused I/O lane in the address and command bank can be used for data pins.

Table 4. General Pin Constraints

Signal Type	Constraint
Data Strobe	All signals belonging to a DQ group must reside in the same I/O lane.
Data	Related DQ pins must reside in the same I/O lane. For protocols that do not support bidirectional data lines, read signals should be grouped separately from write signals.
Address and Command	Address and Command pins must reside in predefined locations within an I/O sub-bank.

Adjacent Banks

For banks to be considered adjacent, they must reside in the same I/O row. To determine if banks are adjacent, refer to the *Modular I/O banks Location and Pin Counts in Intel Agilex Devices* section located in the Intel Agilex General Purpose I/O and LVDS SERDES User Guide.

When referring to tables in the *Intel Agilex General Purpose I/O and LVDS SERDES User Guide*, it is safe to assume that all banks shown are adjacent, unless a ' - ' symbol is present; a ' - ' symbol indicates that the bank is not bonded out for the package.

Pin Assignments

To determine locations for all EMIF I/O pins you should refer to the pin table for your device. When referring to the pin table, the bank numbers, I/O bank indices, and pin names are provided. You can find the pin indices for address and command pins in the *Intel Agilex EMIF Pin Table* at the following location: https://www.intel.com/content/www/us/en/programmable/support/literature/lit-dp.html.

You can perform pin assignments in a variety of ways. The recommended approach is to manually constrain some interface signals and let the Intel Quartus Prime Fitter handle the rest. This method consists of consulting the pin tables to find legal positions for some of the interface pins and assigning them through the `.qsf` file that is generated with the EMIF design example. For this method of I/O placement, you must constrain the following signals:

- CK0
- One DQS pin per group
- PLL reference clock
- RZQ

Based on the above constraints, the Intel Quartus Prime Fitter will rotate pins within each lane as necessary.

2.8. Compiling and Programming the Intel Agilex EMIF Design Example

After you have made the necessary pin assignments in the `.qsf` file, you can compile the design example in the Intel Quartus Prime software.

1. Navigate to the Intel Quartus Prime folder containing the design example directory.
2. Open the Intel Quartus Prime project file, (`.qpf`).
3. To begin compilation, click **Processing ➤ Start Compilation**. The successful completion of compilation generates a `.qsf` file, which enables the design to run on hardware.
4. To program your device with the compiled design, open the programmer by clicking **Tools ➤ Programmer**.
5. In the programmer, click **Auto Detect** to detect supported devices.

6. Select the Intel Agilex device and then select **Change File**.
7. Navigate to the generated `ed_synth.sof` file and select **Open**.
8. Click **Start** to begin programming the Intel Agilex device. When the device is successfully programmed, the progress bar at the top-right of the window should indicate **100% (Successful)**.

3. Design Example Description for External Memory Interfaces Intel Agilex FPGA IP

When you parameterize and generate your EMIF IP, you can specify that the system create directories for simulation and synthesis file sets, and generate the file sets automatically.

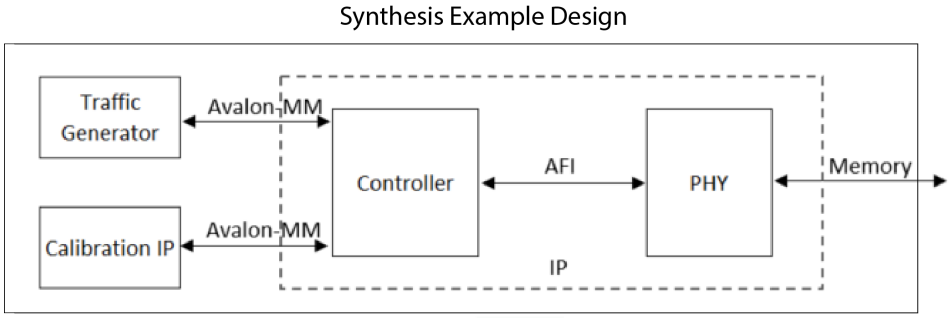
If you select **Simulation** or **Synthesis** under **Example Design Files** on the **Example Designs** tab, the system creates a complete simulation file set or a complete synthesis file set, in accordance with your selection.

3.1. Synthesis Design Example

The synthesis design example contains the major blocks shown in the figure below.

- A traffic generator, which is a synthesizable Avalon®-MM example driver that implements a pseudo-random pattern of reads and writes to a parameterized number of addresses. The traffic generator also monitors the data read from the memory to ensure it matches the written data and asserts a failure otherwise.
- An instance of the memory interface, which includes:
  - A memory controller that moderates between the Avalon-MM interface and the AFI interface.
  - The PHY, which serves as an interface between the memory controller and external memory devices to perform read and write operations.

Figure 4. Synthesis Design Example



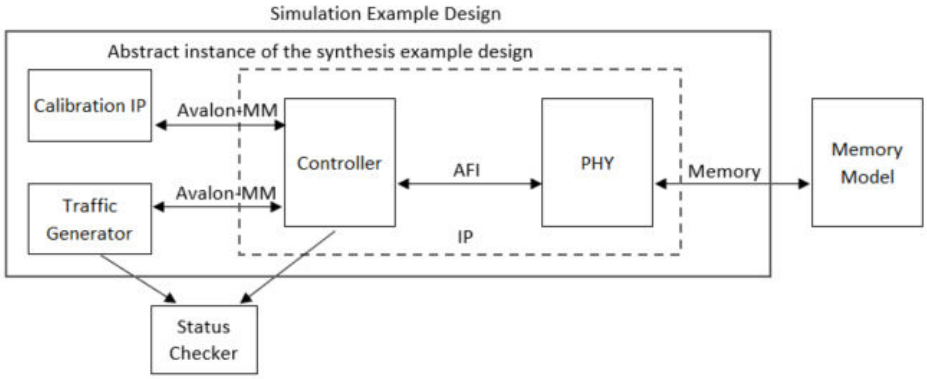
Note: If one or more of the **PLL Sharing Mode**, **DLL Sharing Mode**, or **OCT Sharing Mode** parameters are set to any value other than **No Sharing**, the synthesis design example will contain two traffic generator/memory interface instances. The two traffic generator/memory interface instances are related only by shared PLL/DLL/OCT connections as defined by the parameter settings. The traffic generator/memory interface instances demonstrate how you can make such connections in your own designs.

3.2. Simulation Design Example

The simulation design example contains the major blocks shown in the following figure.

- An instance of the synthesis design example. As described in the previous section, the synthesis design example contains a traffic generator, calibration component, and an instance of the memory interface. These blocks default to abstract simulation models where appropriate for rapid simulation.
- A memory model, which acts as a generic model that adheres to the memory protocol specifications. Frequently, memory vendors provide simulation models for their specific memory components that you can download from their websites.
- A status checker, which monitors the status signals from the external memory interface IP and the traffic generator, to signal an overall pass or fail condition.

Figure 7. Simulation Design Example



3.3. Example Designs Interface Tab

The parameter editor includes an **Example Designs** tab which allows you to parameterize and generate your design examples.

4. Document Revision History for External Memory Interfaces Intel Agilex FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.10.18	19.3		• In the *Creating an EMIF Project* topic, updated the image with point 6. • In the *Generating and Configuring the EMIF IP* topic, updated the figure with step 1. • In the table in the *Intel Agilex EMIF Parameter Editor Guidelines* topic, changed the description for the *Board* tab. • In the *Generating the Synthesizable EMIF Design Example* and *Generating the EMIF Design Example for Simulation* topics, updated the image in step 3 of each topic. • In the *Generating the EMIF Design Example for Simulation* topic, updated the *Generated Simulation Design Example File Structure* figure and modified the note following the figure. • In the *Generating the Synthesizable EMIF Design Example* topic, added a step and a figure for multiple interfaces.
2019.07.31	19.2	1.2.0	• Added *About the External Memory Interfaces Intel Agilex FPGA IP* chapter and Release Information. • Updated dates and version numbers. • Minor enhancement to the *Synthesis Design Example* figure in the *Synthesis Design Example* topic.
2019.04.02	19.1		• Initial release.