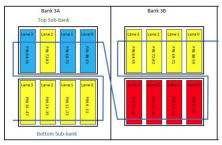
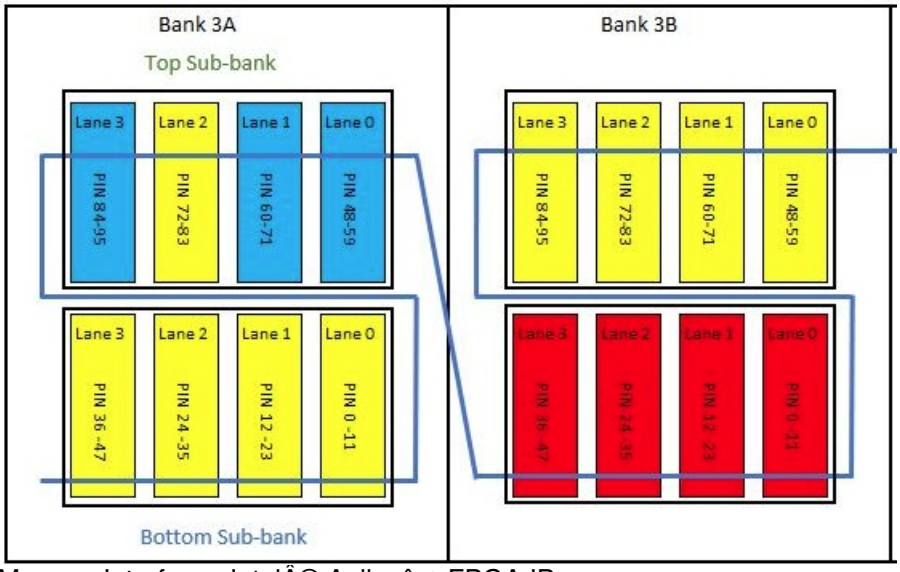


UG-20219 External Memory Interfaces Intel Agilex FPGA IP Design Example User Guide

Contents

UG-20219 External Memory Interfaces Intel Agilex FPGA IP Design Example



About the External Memory Interfaces IntelÃ® Agilexâ„¢ FPGA IP

Release Information

IP versions are the same as the IntelÂ® QuartusÂ® 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:

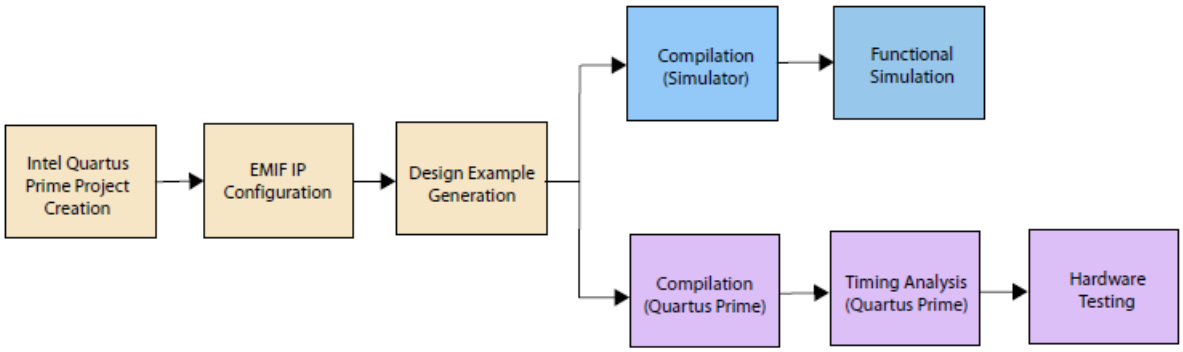
- X indicates a major revision of the IP. If you update your Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Item	Description
IP Version	2.4.2
Intel Quartus Prime	21.2
Release Date	2021.06.21

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 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.

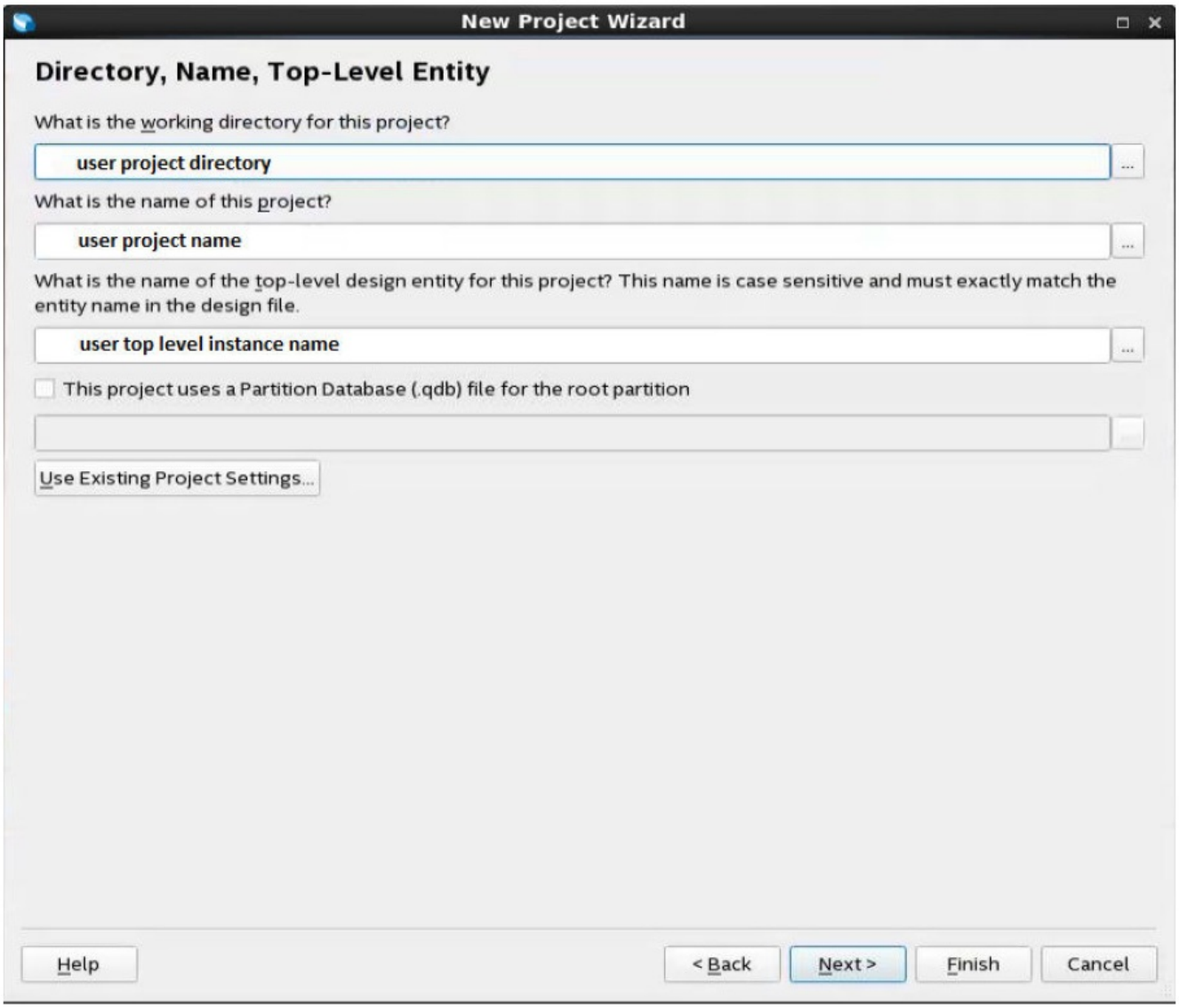
General Design Example Workflows



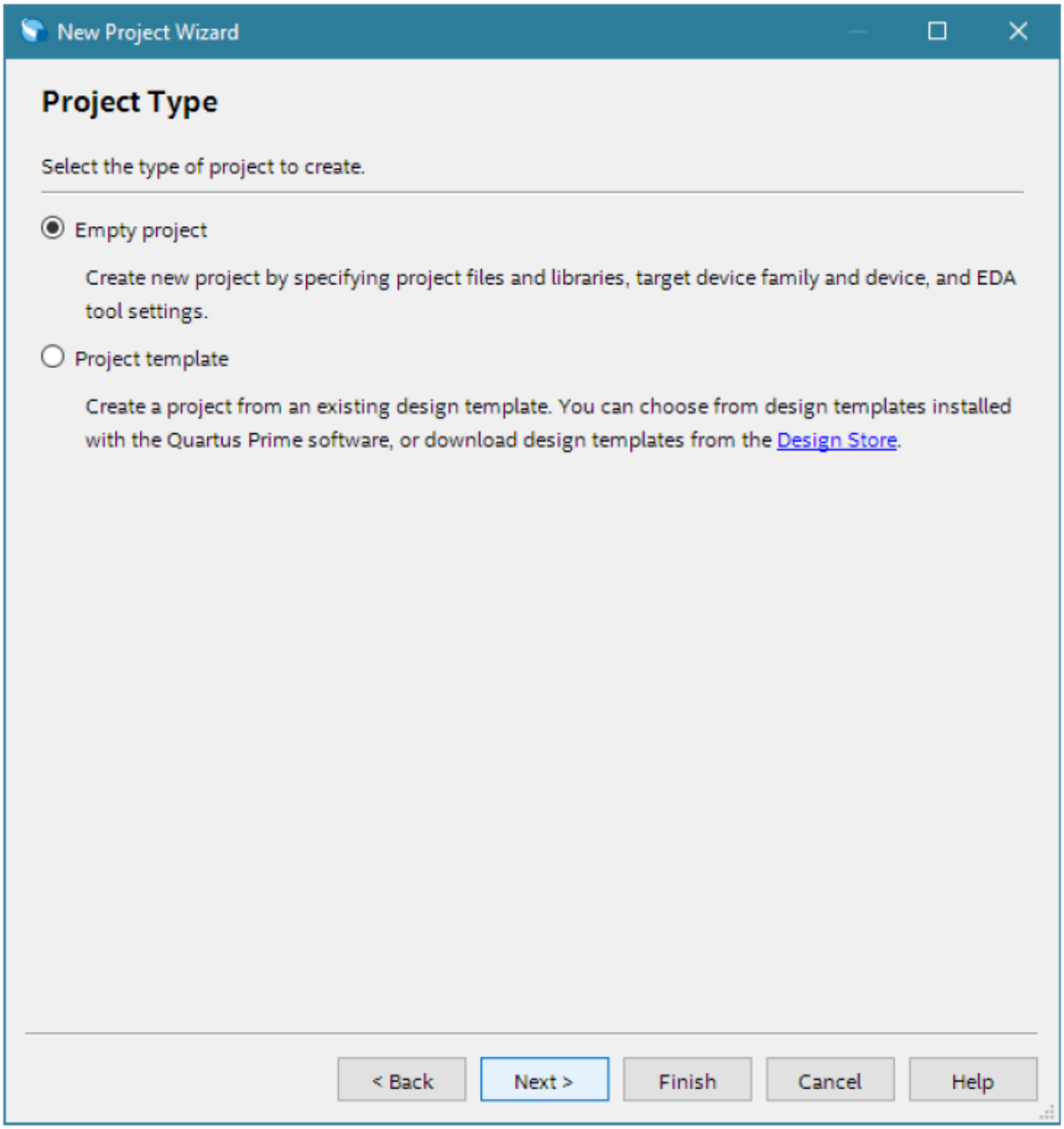
Creating an EMIF Project

For he 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.

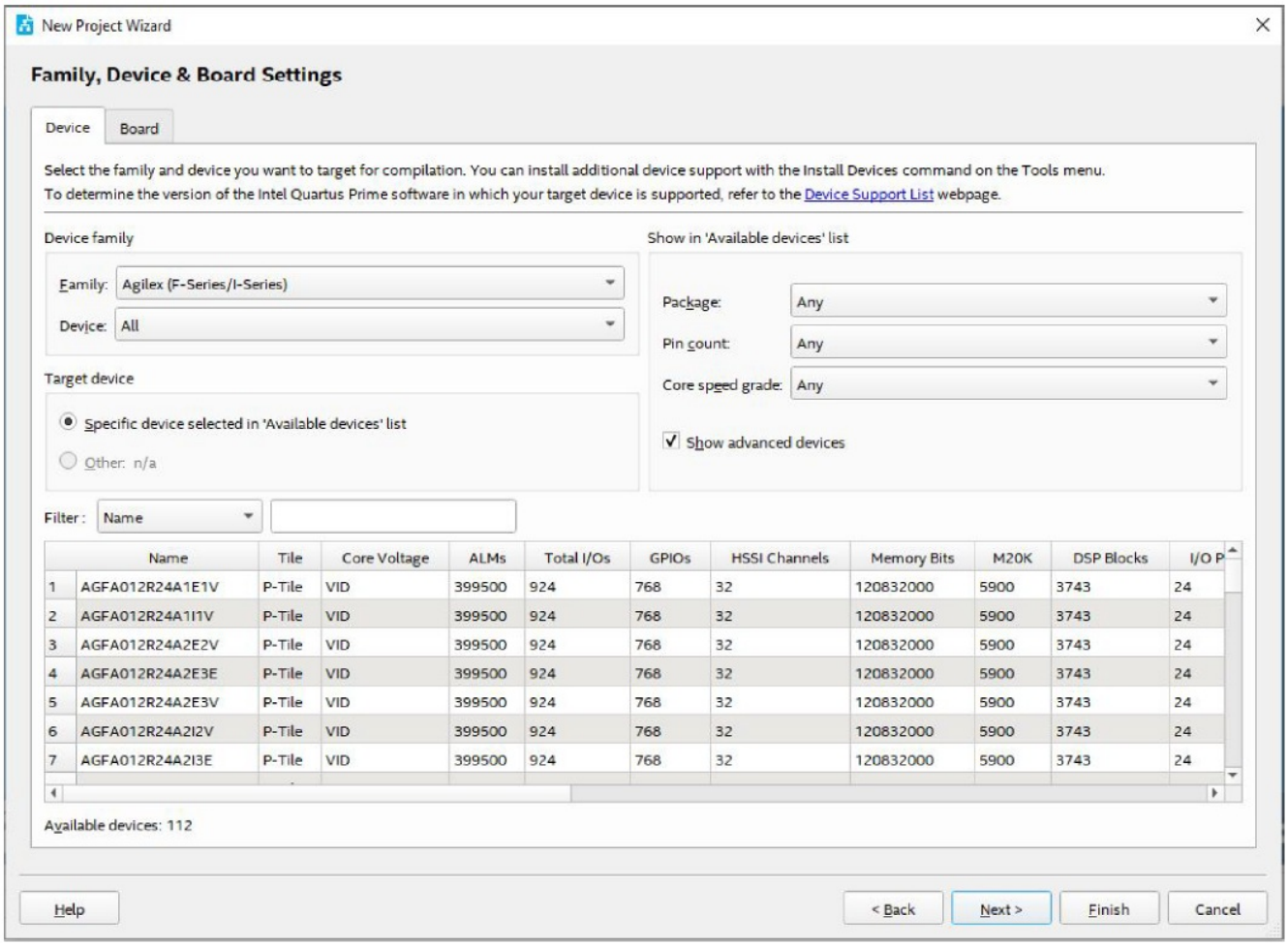
1. Launch the Intel Quartus Prime software and select File ➤ New Project Wizard. Click Next. Design Example Quick Start Guide for External Memory Interfaces Intel Agilex™ FPGA IP
2. Specify a directory (<user project directory>), a name for the Intel Quartus Prime project (<user project name>), and a 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.



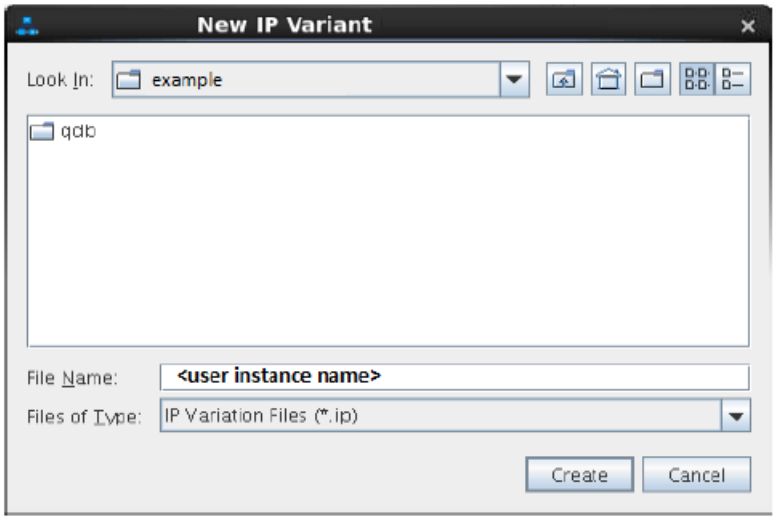
7. Click Finish.

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.

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.

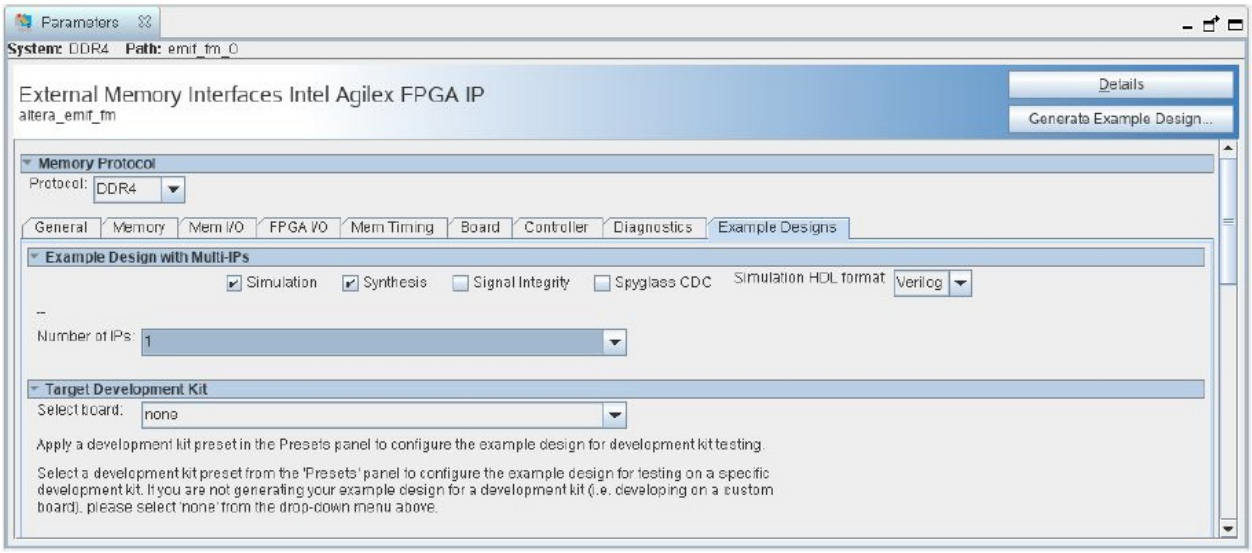
FPGA I/O	• For initial project investigations, you may use the default settings on the **FPGA I/O** tab. • For advanced design validation, you should perform board simulation with associated IBIS models to select appropriate I/O standards.
Mem Timing	• For initial project investigations, you may use 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.

Generating the Synthesizable EMIF Design Example

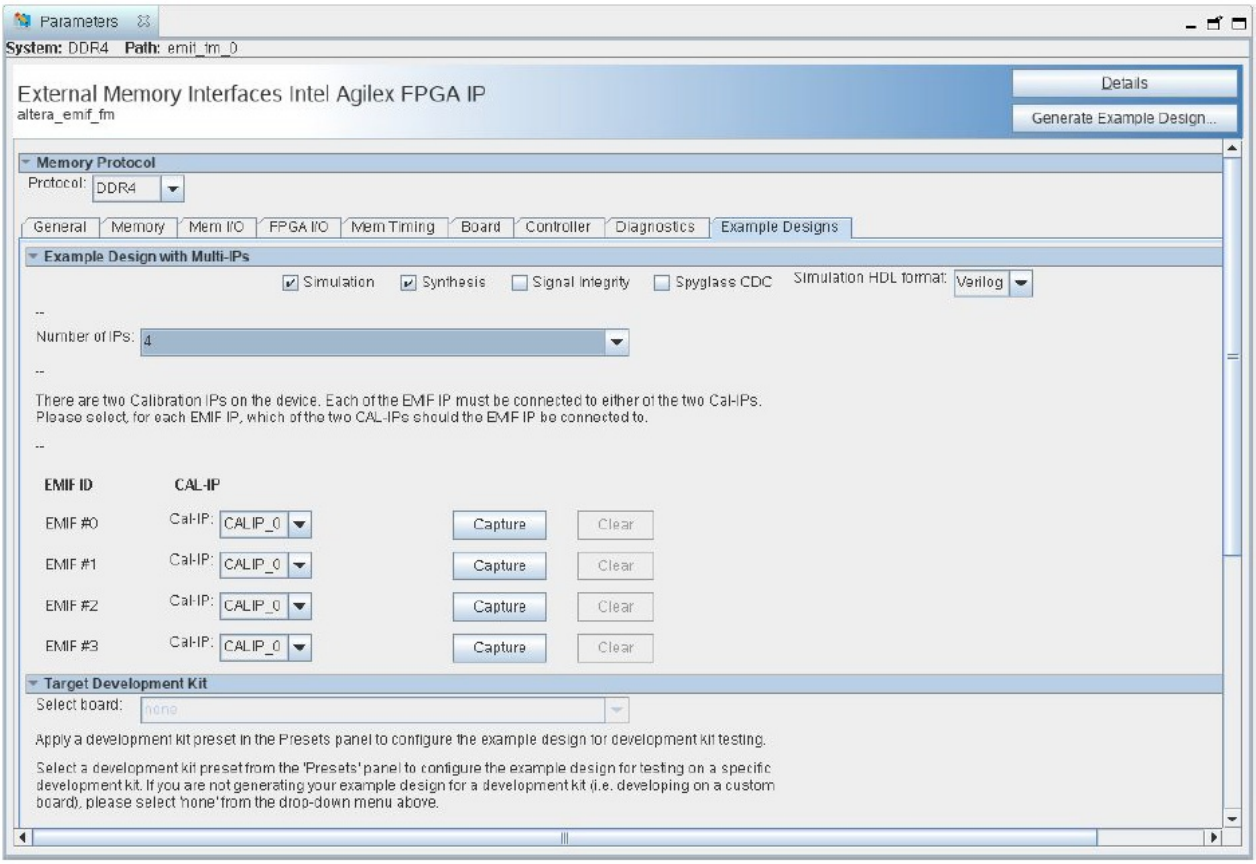
For the Intel Agilex development kit, it is sufficient to leave most of the Intel Agilex EMIF IP settings at their default values. To generate the synthesizable design example, follow these steps:

1. On the Example Designs tab, ensure that the Synthesis box is checked.
    - If you are implementing single interface example design, configure the EMIF IP and click File➤ Save to save the current setting into the user IP variation file (<user instance name>.ip).



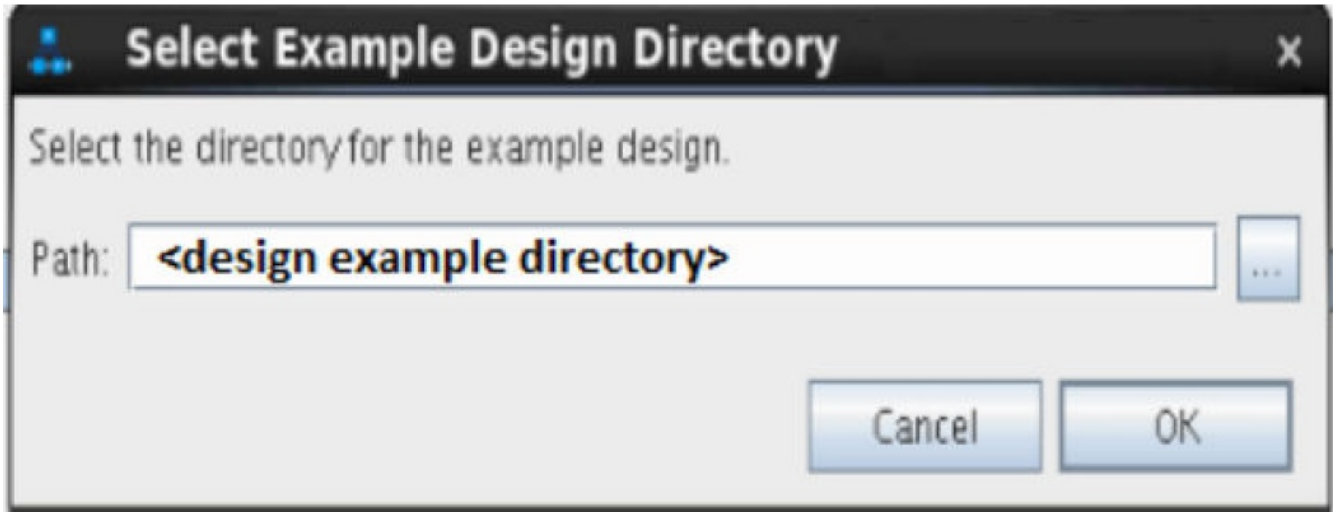
- If you are implementing an example design with multiple interfaces, specify Number of IPs to the desired number of interfaces. You can see the total number of EMIF ID same as the selected Number of IPs. Follow these steps to configure each interface:

- Select the Cal-IP to specify the connection of the interface to the Calibration IP.
- Configure the EMIF IP accordingly in all the Parameter Editor Tab.
- Return to Example Design tab and click Capture on the desired EMIF ID.
- Repeat step a to c for all EMIF ID.
- You may click the Clear button to remove the captured parameters and repeat step a to c to make changes to the EMIF IP.
- Click File➤ Save to save the current setting into the user IP variation file (<user instance name>.ip).



2. Click Generate Example Design in the upper-right corner of the window.

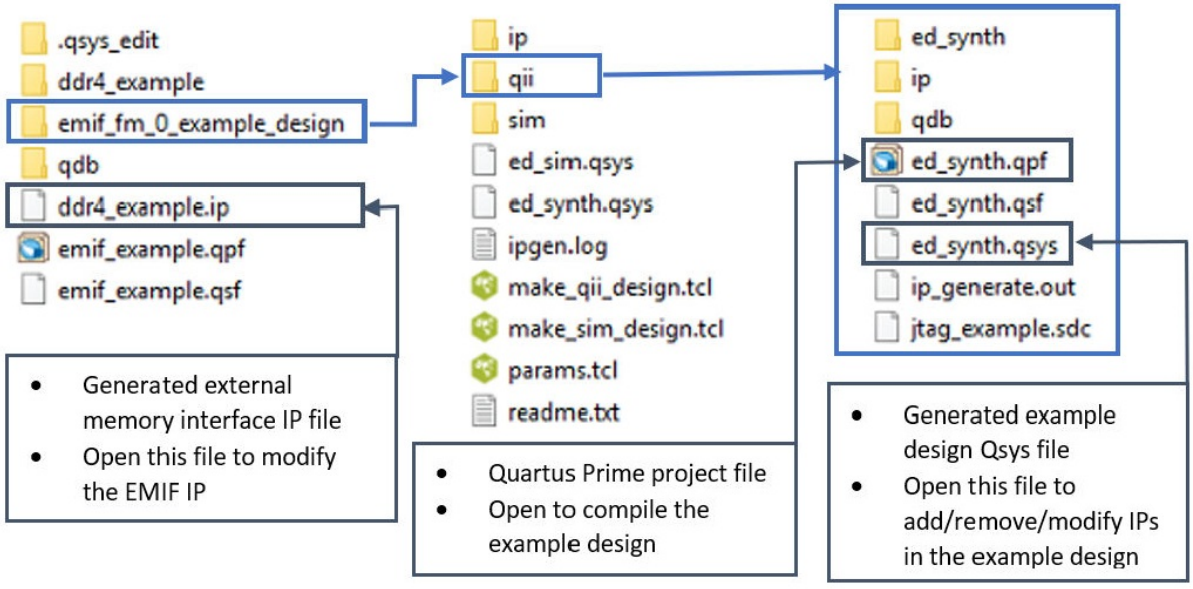
3. Specify a directory for the EMIF design example and click OK. Successful generation of the EMIF design example creates the following fileset under a qii directory.



4. 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.
5. 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 4. Generated Synthesizable Design Example File Structure



For information on constructing a system with two or more external memory interfaces, refer to Creating a Design Example with Multiple EMIF Interfaces, in the External Memory Interfaces Intel Agilex FPGA IP User Guide. For information on debugging multiple interfaces, refer to Enabling the EMIF Toolkit in an Existing Design, in the External Memory Interfaces Intel Agilex FPGA IP User Guide.

Note: If you don't select the Simulation or Synthesis checkbox, the destination directory contains only Platform Designer design files, which are not compilable by the Intel Quartus Prime software directly, but which you can view or edit in 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.tclscript 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.

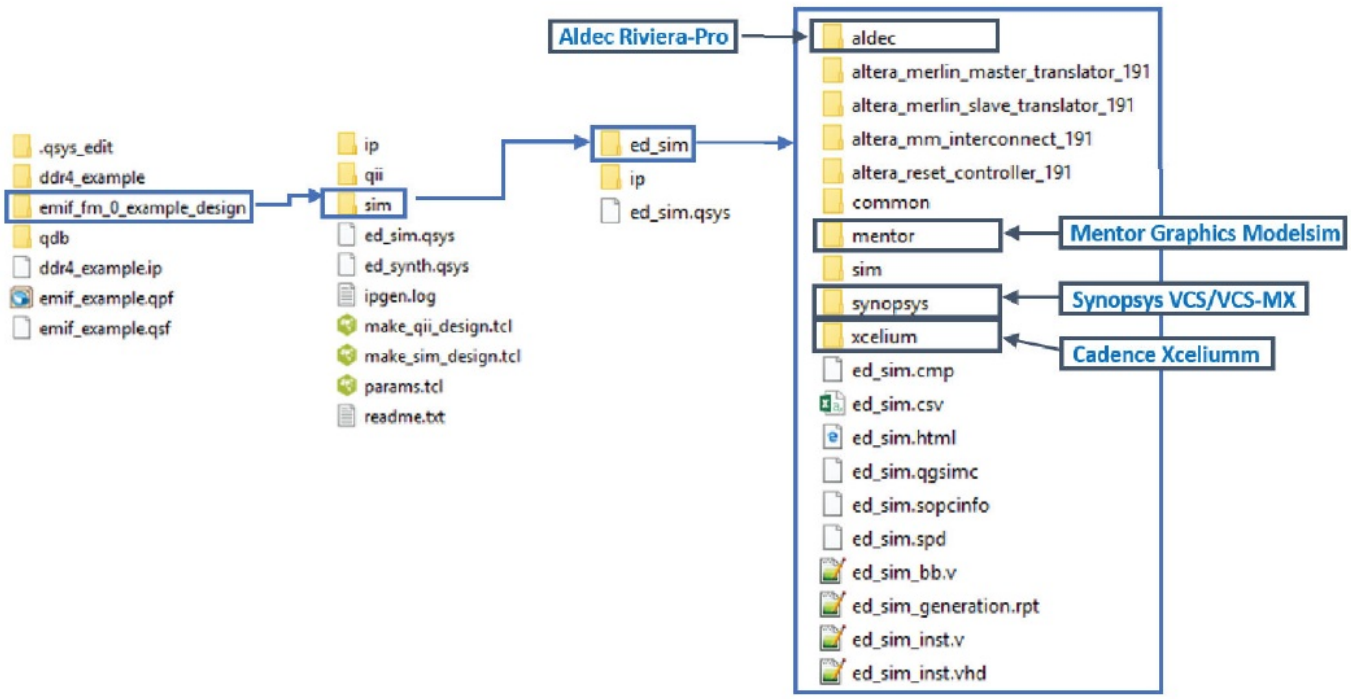
Note: If you have generated a design example and then make changes to it in the parameter editor, you must regenerate the design example to see your changes implemented. The newly generated design example does not overwrite the existing design example files.

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 Pro window. The system prompts, Recent changes have not been generated. Generate now? Click No to continue with the next flow.

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 is planned in future releases.

Note: If you don't select the Simulation or Synthesis checkbox, the destination directory contains only Platform Designer design files, which are not compilable by the Intel Quartus Prime software directly, but which you can view or edit in 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.

Note: If you have generated a design example and then make changes to it in the parameter editor, you must regenerate the design example to see your changes implemented. The newly generated design example does not overwrite the existing design example files.

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 par allel.	Nios initialization and calibration code execute sequentially.
Interfaces assert cal_done signal simultaneously in si mulation.	Fitter operations determine the order of calibration, an d 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.

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.

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.

Note: For more detailed pin information, refer to the Intel Agilex FPGA EMIF IP Pin and Resource Planning section in the protocol-specific chapter for your external memory protocol, in the External Memory Interfaces Intel Agilex FPGA IP User Guide.

- Ensure that the pins for a given external memory interface reside within the same I/O row.
- 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 EMIF Architecture: I/O Bank topic in the External Memory Interfaces Intel Agilex FPGA IP User Guide.
- All address and command and associated pins must reside within a single subbank.
- Address and command and data pins can share a sub-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 contain 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 a n I/O sub-bank.

Note: For more detailed pin information, refer to the Intel Agilex FPGA EMIF IP Pin and Resource Planning section in the protocol-specific chapter for your external memory protocol, in the External Memory Interfaces Intel Agilex FPGA IP User Guide.

- Ensure that the pins for a given external memory interface reside within the same I/O row.
- 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 EMIF Architecture: I/O Bank topic in the External Memory Interfaces Intel Agilex FPGA IP User Guide.
- All address and command and associated pins must reside within a single subbank.
- Address and command and data pins can share a sub-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 contain data pins.

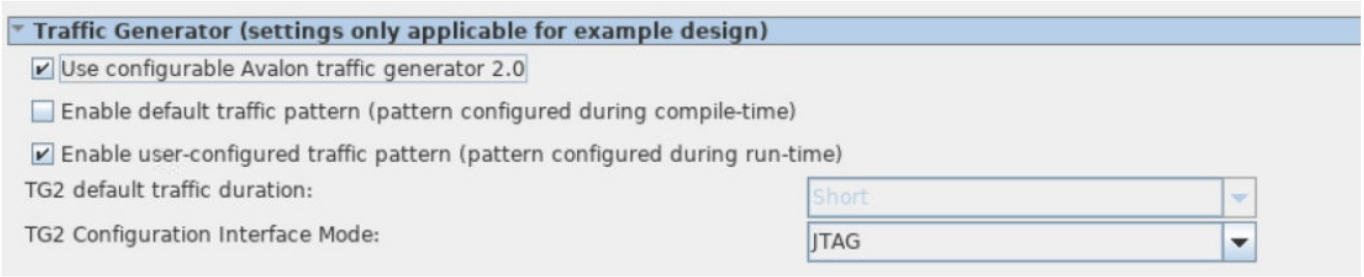
Generating a Design Example with the TG Configuration Option

The generated EMIF design example includes a traffic generator block (TG). By default, the design example uses a simple TG block (altera_tg_avl) which can only be reset in order to relaunch a hard-coded traffic pattern. If necessary, you may choose to enable a configurable traffic generator (TG2) instead. In the configurable traffic generator (TG2) (altera_tg_avl_2), you can configure the traffic pattern in real time through control registers—meaning that you do not have to recompile the design to change or relaunch the traffic pattern. This traffic generator provides fine control over the type of traffic that it sends on the EMIF control interface. Additionally, it provides status registers that contain detailed failure information.

Enabling the Traffic Generator in a Design Example

You can enable the configurable traffic generator from the Diagnostics tab in the EMIF parameter editor. To enable the configurable traffic generator, turn on Use configurable Avalon traffic generator 2.0 on the Diagnostics tab.

Figure 6.



- You may choose to disable the default traffic pattern stage or the userconfigured traffic stage, but you must

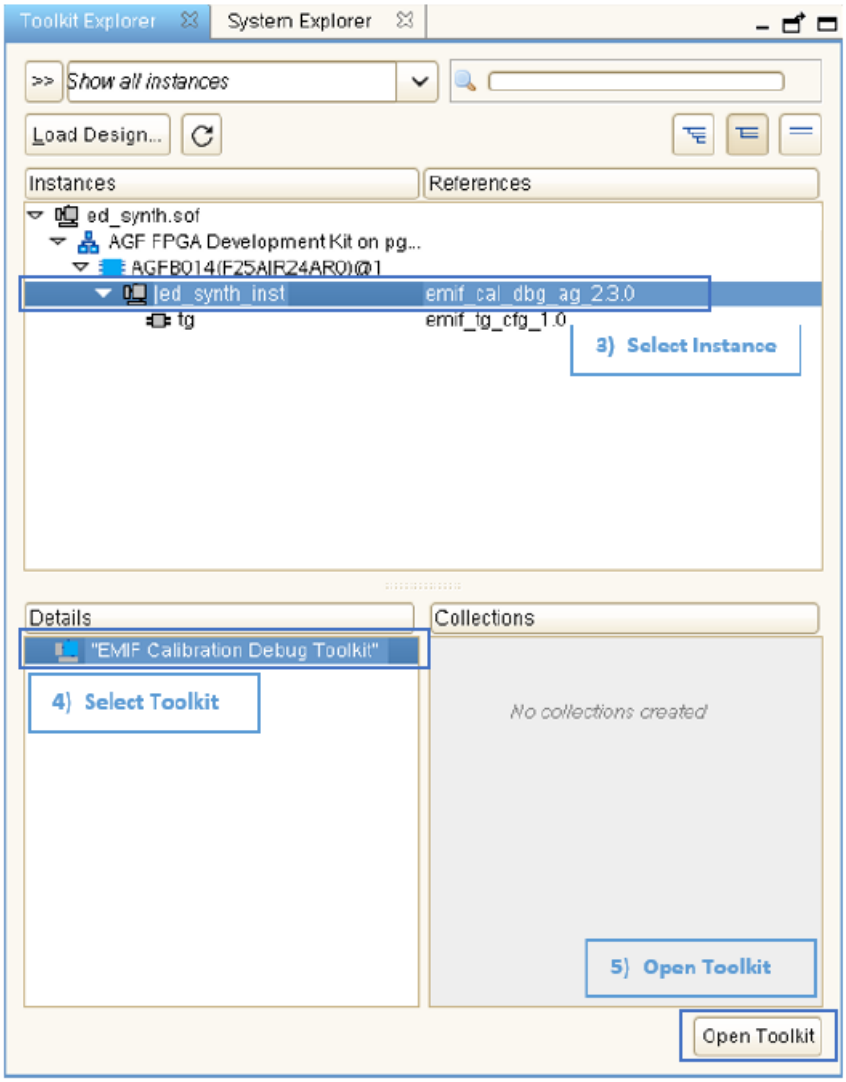
have at least one stage enabled. For information on these stages, refer to Default Traffic Pattern and User-configured Traffic Pattern in the External Memory Interfaces Intel Agilex FPGA IP User Guide.

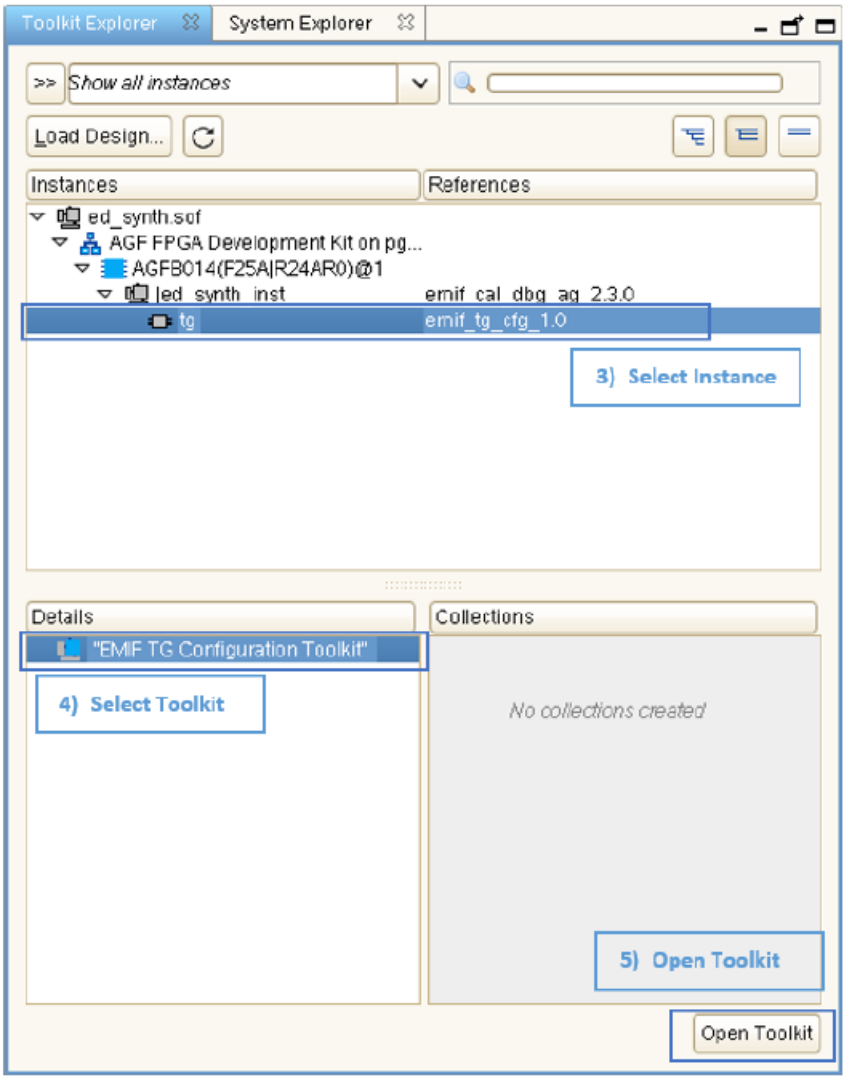
- The TG2 test duration parameter applies only to the default traffic pattern. You may choose a test duration of short, medium, or infinite.
- ou may choose either of two values for the TG2 Configuration Interface Mode parameter:
  - **JTAG:** Allows use of a GUI in the system console. For more information, refer to Traffic Generator Configuration Interface in the External Memory Interfaces Intel Agilex FPGA IP User Guide.
  - **Export:** Allows use of custom RTL logic to control the traffic pattern.

Using the Design Example with the EMIF Debug Toolkit

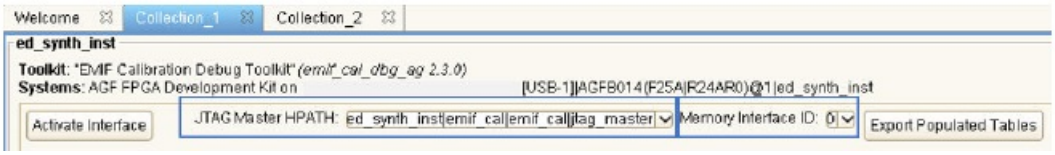
Before launching the EMIF Debug Toolkit, ensure that you have configured your device with a programming file that has the EMIF Debug Toolkit enabled. To launch the EMIF Debug Toolkit, follow these steps:

1. In the Intel Quartus Prime software, open the System Console by selecting Tools ➤ System Debugging Tools ➤ System Console.
2. [Skip this step if your project is already open in the Intel Quartus Prime software.] In the System Console, load the SRAM object file (.sof) with which you programmed the board (as described in Prerequisites for Using the EMIF Debug Toolkit, in the External Memory Interfaces Intel Agilex FPGA IP User Guide).
3. Select instances to debug.
4. Select EMIF Calibration Debug Toolkit for EMIF calibration debugging, as described in Generating a Design Example with the Calibration Debug Option. Alternatively, select EMIF TG Configuration Toolkit for traffic generator debugging, as described in Generating a Design Example with the TG Configuration Option.
5. Click Open Toolkit to open the main view of the EMIF Debug Toolkit.


Toolkit Explorer
System Explorer
Show all instances
Load Design...
Instances
References
ed_synth.sof
AGF FPGA Development Kit on pg...
AGFB014(F25AIR24AR0)@1
ed_synth_inst
emif_cal_dbg_ag_23.0
tg
emif_tg_cfg_1.0
3) Select Instance
Details
Collections
"EMIF Calibration Debug Toolkit"
4) Select Toolkit
No collections created
5) Open Toolkit
Open Toolkit



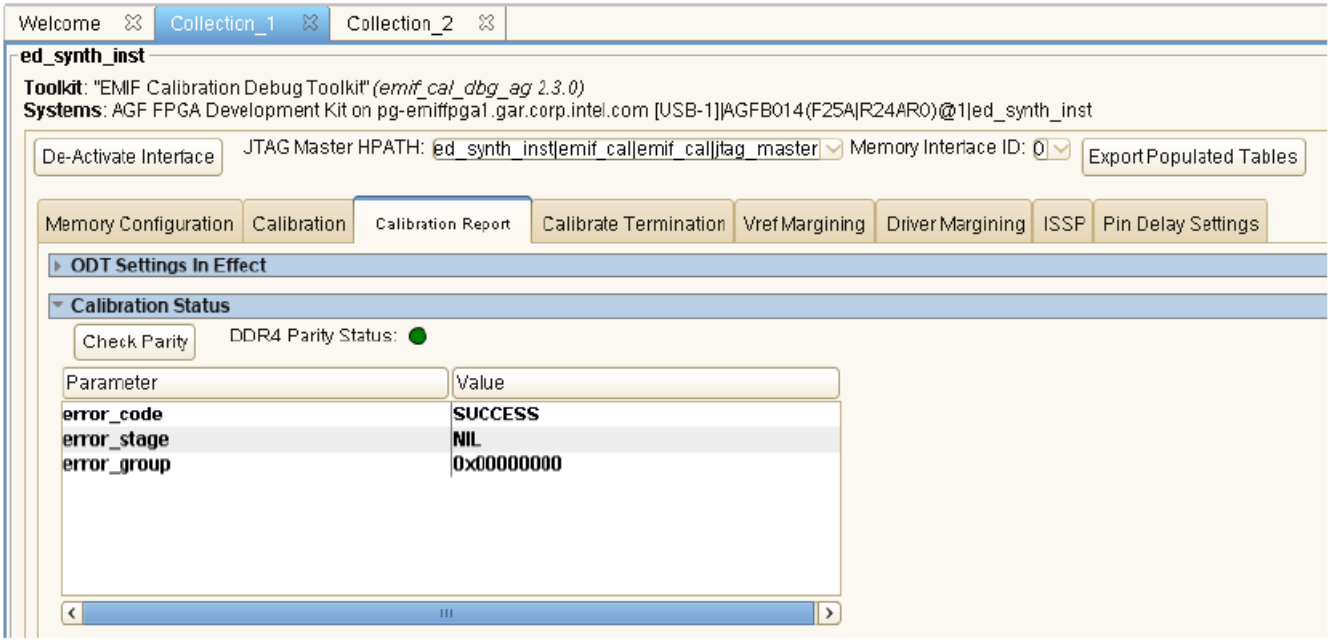
6. If there are multiple EMIF instances in the programmed design, select the column (path to JTAG master) and memory interface ID of the EMIF instance for which to activate the toolkit.

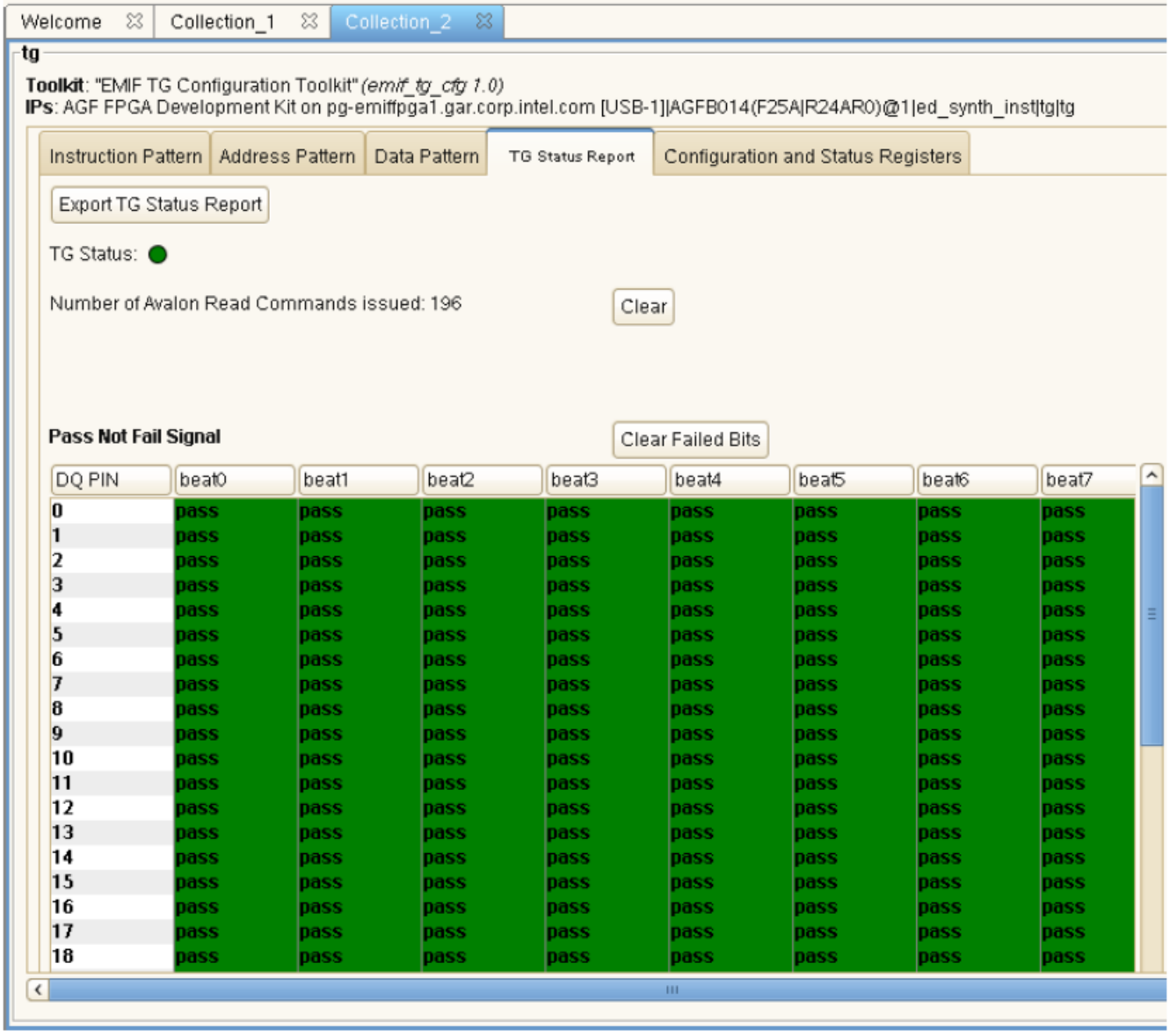


7. Click Activate Interface to allow the toolkit to read the interface parameters and calibration status.

8. You must debug one interface at a time; therefore, to connect to another interface in the design, you must first deactivate the current interface.

The following are examples of reports from the EMIF Calibration Debug Toolkit and the EMIF TG Configuration Toolkit:, respectively.





Note: For details on calibration debugging, refer to Debugging with the External Memory Interface Debug Toolkit, in the External Memory Interfaces Intel Agilex FPGA IP User Guide.

Note: For details on traffic generator debugging, refer to Traffic Generator Configuration User Interface, in the External Memory Interfaces Intel Agilex FPGA IP User Guide.

Design Example Description for External Memory Interfaces Intel Agilex FPGA IP

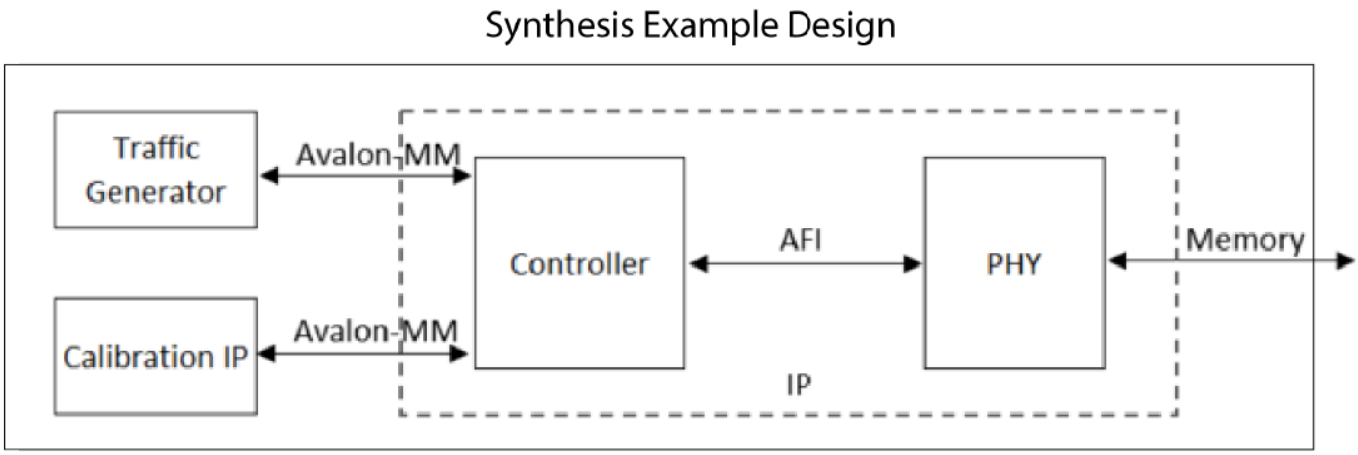
When you parameterize and generate your EMIF IP, you can specify that the system create directories for simulation and synthesis file sets, and generate the file sets automatically. If you select Simulation or Synthesis under Example Design Files on the Example Designs tab, the system creates a complete simulation file set or a complete synthesis file set, in accordance with your selection.

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 7. Synthesis Design Example



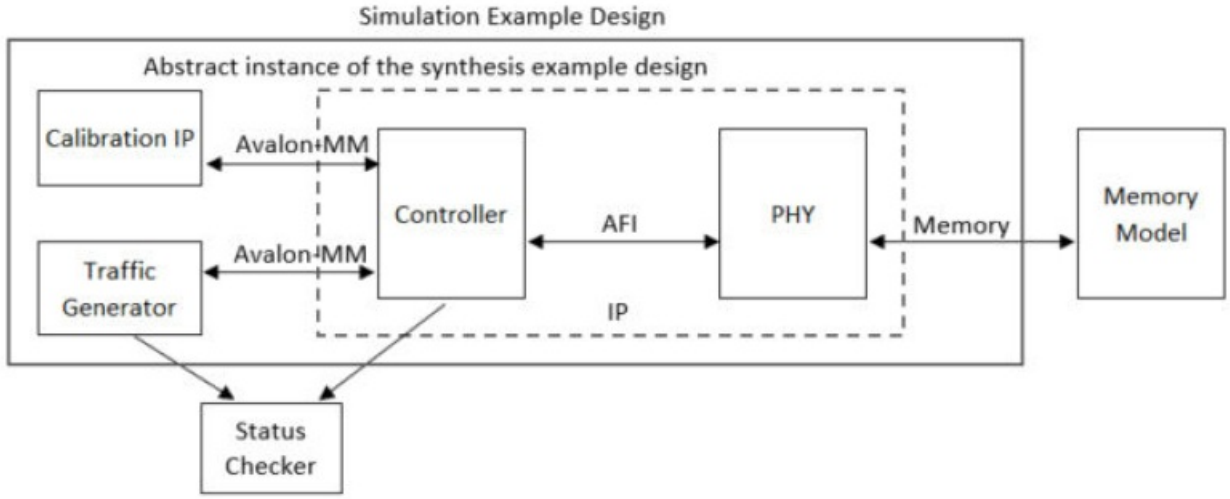
Note: If one or more of the PLL Sharing Mode, DLL Sharing Mode, or OCT Sharing Mode parameters are set to any value other than No Sharing, the synthesis design example will contain two traffic generator/memory interface instances. The two traffic generator/memory interface instances are related only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

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 10. Simulation Design Example



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

External Memory Interfaces Intel Agilex FPGA IP Design Example User Guide Archives

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, IPs have a new IP versioning scheme. If an IP core version is not listed, the user guide for the previous IP core version applies.

IP Core Version	User Guide
2.4.0	External Memory Interfaces Intel Agilex FPGA IP Design Example User Guide Archives
2.3.0	External Memory Interfaces Intel Agilex FPGA IP Design Example User Guide Archives
2.3.0	External Memory Interfaces Intel Agilex FPGA IP Design Example User Guide Archives
2.1.0	External Memory Interfaces Intel Agilex FPGA IP Design Example User Guide Archives
19.3	External Memory Interfaces Intel Agilex FPGA IP Design Example User Guide Archives

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

Document Version	Intel Quartus Prime Version	IP Version	Changes

2021.06.21	21.2	2.4.2	In the *Design Example Quick Start* chapter: • Added a note to the *Compiling and Programming the Intel Agilex EMIF Design Example* topic. • Modified the title of the *Generating a Design Example with the Calibration Debug Option* topic. • Added the *Generating a Design Example with the TG Configuration Option* and *Enabling the Traffic Generator in a Design Example* topics. • Modified steps 2, 3, and 4, updated several figures, and added a note, in the *Using the Design Example with the EMIF Debug Toolkit* topic.
2021.03.29	21.1	2.4.0	In the *Design Example Quick Start* chapter: • Added a note to the *Generating the Synthesizable EMIF Design Example* and *Generating the EMIF Design Example for Simulation* topics. • Updated the File Structure diagram in the *Generating the EMIF Design Example for Simulation* topic.
2020.12.14	20.4	2.3.0	In the *Design Example Quick Start* chapter, made the following changes: • Updated the *Generating the Synthesizable EMIF Design Example* topic to include multi-EMIF designs. • Updated the figure for step 3, in the *Generating the EMIF Design Example for Simulation* topic.

2020.10.05	20.3	2.3.0	In the *Design Example Quick Start Guide* chapter, made the following changes: • In *Creating an EMIF Project*, updated the image in step 6. • In *Generating the Synthesizable EMIF Design Example*, updated the figure in step 3. • In *Generating the EMIF Design Example for Simulation*, updated the figure in step 3. • In *Simulation Versus Hardware Implementation*, corrected a minor typo in the second table. • In *Using the Design Example with the EMIF Debug Toolkit*, modified step 6, added steps 7 and 8.
continued...			

Document Version	Intel Quartus Prime Version	IP Version	Changes

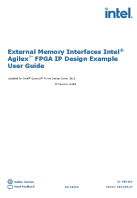
2020.04.13	20.1	2.1.0	• In the *About* chapter, modified the table in t he *Release Information* topic. • In the *Design Example Quick Start Guide* chapter: — Modified step 7 and the associated image, in the *Generating the Synthesizable EMIF Desi gn Example* topic. — Modified the *Generating the Design Examp le with the Debug Option* topic. — Modified the *Using the Design Example wit h the EMIF Debug Toolkit* topic.
2019.12.16	19.4	2.0.0	• In the *Design Example Quick Start* chapter: — Updated the illustration in step 6 of the *Creating an EMIF Project* topic. — Updated the illustration in step 4 of the *Ge nerating the Synthesizable EMIF Design Exam ple* topic. — Updated the illustration in step 4 of the *Ge nerating the EMIF Design Example for Simulati on* topic. — Modified step 5 in the *Generating the EMIF Design Example for Simulation* topic. — Modified the *General Pin Guidelines* and *A djacent Banks* sections of the *Pin Placement fo r Intel Agilex EMIF IP* topic.

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.

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

Documents / Resources

	intel UG-20219 External Memory Interfaces Intel Agilex FPGA IP Design Example [pdf] User Guide UG-20219 External Memory Interfaces Intel Agilex FPGA IP Design Example, UG-20219, External Memory Interfaces Intel Agilex FPGA IP Design Example, Interfaces Intel Agilex FPGA IP Design Example, Agilex FPGA IP Design Example

References

- Intel | Data Center Solutions, IoT, and PC Innovation
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- 1. About the External Memory Interfaces Intel® Agilex™ FPGA IP
- Pin-Out Files for Intel Field Programmable Gate Arrays (FPGAs) | Intel
- Intel ISO 9001:2015 Registrations