High Bandwidth Memory (HBM2) Interface Intel® FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: 19.2

IP Version: 19.2.0
1. About the High Bandwidth Memory (HBM2) Interface
Intel® 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.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Version</td>
<td>19.2.0</td>
</tr>
<tr>
<td>Intel Quartus Prime Version</td>
<td>19.2</td>
</tr>
<tr>
<td>Release Date</td>
<td>2019.07.01</td>
</tr>
</tbody>
</table>
2. High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example Quick Start Guide

An automated design example flow is available for the High Bandwidth Memory (HBM2) Interface Intel FPGA IP.

You can use the Example Designs tab and the Generate Example Designs button in the IP Parameter Editor Pro window to specify and generate synthesis and simulation example design file sets with which you can validate your HBM2 IP.

The generated design example reflects the parameterization that you set in the IP Parameter Editor Pro window. You can generate a design example to specifically match an Intel FPGA development kit for your evaluation. Or you can generate a design example to match your own actual system requirements, as a starting point for creating your own system.

Figure 1. General Design Example Flow

2.1. Creating an Intel Quartus Prime Project for Your HBM2 System

To create a new IP variation, you must have an open Intel Quartus Prime project with a valid device part. If you do not already have an appropriate Intel Quartus Prime project, follow these steps to create an Intel Quartus Prime project targeting a device that supports High Bandwidth Memory (HBM2) Interface Intel FPGA IP.

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

2. Specify a directory name (<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. In the **Device Family** section of the dialog box, select Intel Stratix® 10 for the **Family** and Stratix 10 MX for the **Device**. Alternatively, if you already know the specific part number of the device that you want to use, type that device part number in the **Name filter** field.

   **Note:** The generated HBM2 IP targets the specific Intel Stratix 10 MX device that you specify. Because different Intel Stratix 10 MX part numbers may identify devices with differing numbers of HBM2 devices and differing HBM2 device locations, as well as differing stack count, height, and density properties, it is important that you specify the correct Intel Stratix 10 MX device part number before proceeding to the next step.

5. Select the appropriate device under **Available Devices**.
6. Click **Finish**. The system creates the \(<\text{user project directory}>\).qpf and \(<\text{user project directory}>\).qsf files within the \(<\text{user project directory}>\).

**Figure 2.** Generated Intel Quartus Prime Project and Settings File

- \(<\text{user project directory}>\) - Your Intel Quartus Prime project directory
- \(<\text{user instance name}>\).qpf and \(<\text{user instance name}>\).qsf - Generated Intel Quartus Prime project and settings files

### 2.2. Configuring the High Bandwidth Memory (HBM2) Interface Intel FPGA IP

The following steps illustrate the IP Catalog (standalone) configuration flow. The Platform Designer (standalone) flow is generally similar.

1. Verify that the **IP Catalog** window is visible. If the **IP Catalog** window is not visible, select **View ➤ Utility Windows ➤ IP Catalog**.

2. In the **IP Catalog** window, select **Installed IP ➤ Library ➤ Memory Interfaces and Controllers ➤ High Bandwidth Memory (HBM2) Interface Intel FPGA IP**.

3. In the **New IP Variant** window, type a name \(<\text{user instance name}>\) for your HBM2 IP in the **File Name** field. Click **Create**.
4. In the **IP Parameter Editor Pro** window, configure the parameters on each tab to reflect your Intel FPGA development kit requirement or your actual HBM2 interface and system requirement. For information on individual parameters, refer to the Parameterizing the High Bandwidth Memory (HBM2) Interface Intel FPGA IP section in the High Bandwidth Memory (HBM2) Interface Intel FPGA IP User Guide.

**Related Information**

Parameterizing the High Bandwidth Memory (HBM2) Interface Intel FPGA IP
The following table provides high-level guidance for parameterizing each of the tabs in the HBM2 IP parameter editor.

For detailed guidance on individual parameters, refer to the Parameter Description section in the High Bandwidth Memory (HBM2) Interface Intel FPGA IP User Guide.

Note: For the Intel Stratix 10 MX development kit, you may leave most of the High Bandwidth Memory (HBM2) Interface Intel FPGA IP settings at their default values.

Table 2. Tab Parameterization Guidelines

<table>
<thead>
<tr>
<th>Parameter Editor Tab</th>
<th>Guideline</th>
</tr>
</thead>
</table>
| General              | Correctly enter the following parameters to reflect your Intel FPGA development kit requirement or your HBM2 interface and system requirement:  
• The Speed grade for the device. The displayed speed grade should match the selected device in Intel Quartus Prime Project for User.  
• The desired HBM location.  
• The desired number of HBM channels. This determines the number of Traffic Generator pairs.  
• The desired HBM2 memory clock frequency.  
• The PLL reference clock frequency. This reference clock is for the High Bandwidth Memory (HBM2) Interface Intel FPGA IP subsystem and should match the PLL reference clock frequency that you provide to the hbm_0_example_design_pll_ref_clk_clk pin.  
• The Reference clock frequency for example design core clock PLL. This reference clock is for the core clock PLL instantiated in the design example file as an example external clock that feeds the ext_core_clock port of the High Bandwidth Memory (HBM2) Interface Intel FPGA IP. The value should match the PLL reference clock frequency that you supply to the core_clk_iopll_ref_clk_clk pin.  
• The desired Core clock frequency. The value determines the output clock of the core clock PLL instantiated in the design example file as an example external clock that feeds the ext_core_clock port of the High Bandwidth Memory (HBM2) Interface Intel FPGA IP. The clock eventually clocks the Traffic Generator and any other components clocked by the wmc_clk_0_clk clock of the High Bandwidth Memory (HBM2) Interface Intel FPGA IP. |
| Controller #         | Set the parameters to reflect your actual HBM2 interface and system requirement for the controller. |
| Diagnostic           | For initial project investigations, you may use the default settings on the Diagnostic tab.  
For hardware testing using the synthesizable design example, check the Enable In-System-Sources-and-Probes checkbox to allow you to easily control and monitor the High Bandwidth Memory (HBM2) Interface Intel FPGA IP example design system through the Intel Quartus Prime software.  
For efficiency testing on both synthesis and simulation designs, check the Use Efficiency Pattern and Enable Efficiency Test Mode checkboxes. Keep both the read count and write count the same to ensure that the validity check passes. Select the Data Sequence (Random/Sequential) option for testing, and check Enable data check for efficiency measurement for data validity check.  
For simulation, if you want the simulation to report the efficiency number for each HBM channel that you have enabled, check the Enable Efficiency Monitor checkbox. (For hardware testing using the synthesizable example design project, you should not enable the Efficiency Monitor, as this feature reduces the frequency at which the interface will close timing in the core clock domain.) |

continued...
Parameter Editor Tab | Guideline
--- | ---
You can also use the other parameters on the Diagnostics tab to assist you in evaluating, verifying and debugging the High Bandwidth Memory (HBM2) Interface Intel FPGA IP.

Example Designs | To get the correct design example file sets, ensure that you check either the Simulation or Synthesis checkbox, or both, in the Example Design Files section. The generated design example is a complete HBM2 system consisting of the High Bandwidth Memory (HBM2) Interface Intel FPGA IP and a driver that generates random traffic to validate the memory interface.

2.4. Generating the Synthesizable High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example

For the Intel Stratix 10 MX development kit, you may leave most of the High Bandwidth Memory (HBM2) Interface Intel FPGA IP settings at their default values. To generate the synthesizable design example, follow these additional steps.

1. On the Diagnostics tab, select Example Design ➤ Enable In-System-Sources-and-Probes to provide access to the available debugging features integrated in the design example.

2. On the Diagnostics tab, select Traffic Generator ➤ Use efficiency pattern to provide high-efficiency concurrent traffic pattern features integrated in the design example. You can configure the different number of read/write transactions, select between sequential or random traffic addressing, and enable or disable the data-matching check. This option also enables the Read command round trip latency feature. Latency is measured from the start of read command valid by the driver, until the response is received.

3. If it is not already checked, check the Synthesis checkbox under Example Design Files on the Example Designs tab.
Note: The system enables both the **Simulation** and **Synthesis** checkboxes by default.

4. Click **File ➤ Save**. The system creates and saves the current settings in the user IP variation file (`<user_instance_name>.ip`), within the `<user_project_directory>` that you specified.

5. Click **Generate Example Design** in the top-right corner of the window.

6. Specify a directory for the generated design example (`<design_example_directory>`) and click **OK**. The system creates the following design example file sets within the specified directory:
7. To exit the IP Parameter Editor Pro window, click File ➤ Exit.

2.5. Generating the Synthesizable High Bandwidth Memory (HBM2) Interface Intel FPGA IP for High Efficiency

To generate the synthesizable design example with high efficiency in the HBM2 interface and measure the traffic efficiency using the In-System Sources and Probes (ISSP) Editor, follow these steps.

1. On the Controller tab, deselect the Controller configuration ➤ Enable Reorder Buffer option. The Reorder Buffer rearranges the read data in the order of the issued requests.
2. On the Diagnostics tab, select Traffic Generator ➤ Force traffic generator to issue different AXI Read/Write IDs. Also, select Traffic Generator ➤ Use efficiency pattern.

3. Set both the Read Count and Write Count values to at least 2500, and select the Sequential option for data traffic Sequence. (Alternatively, if you want to measure the efficiency for traffic with random addresses, set the Sequence option to Random.)

4. On the Diagnostics tab, select Performance ➤ Enable Efficiency Monitor. Also, select Performance ➤ Enable Efficiency Test Mode. If it is not already checked, check the Enable In-System-Sources-and-Probes checkbox in the Example Design group box.

5. Click Generate Example Design at the top-right corner of the parameter editor to generate the synthesizable design example.
6. Apply the specific pinout for the development kit and compile the design.
7. After the design example is programmed on the circuit board, you can obtain efficiency performance results using the ISSP editor.
8. Monitor and ensure transactions have completed as indicated by a 1 on the EFFT probe in the In-System Sources and Probes (ISSP) Editor.

The EFFR 32-bit probe signals can display the value of valid write and read transactions and the total number of active transactions.

The EFFW is a 3-bit input set source input signal which enables output on the EFFR 32-bit probe signals. The following table shows input values for EFFW and the corresponding EFFR counter output values:

<table>
<thead>
<tr>
<th>EFFW Source Value</th>
<th>Description of EFFR output (32-bits wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3'b 000</td>
<td>Number of valid write data transactions.</td>
</tr>
<tr>
<td>3'b 001</td>
<td>Number of valid read data transactions.</td>
</tr>
<tr>
<td>3'b 010</td>
<td>Total active transaction count.</td>
</tr>
<tr>
<td>3'b 100</td>
<td>Minimum read latency.</td>
</tr>
<tr>
<td>3'b 101</td>
<td>Maximum read latency.</td>
</tr>
</tbody>
</table>

9. You can calculate efficiency using the following formula, based on the counter values:

\[
\text{Efficiency} = \left( \frac{\text{Number of Write Data transactions} + \text{Number of Read Data transactions accepted by HBM2 controller}}{\text{total active transaction count}} \times \left( \frac{\text{core clock frequency} \times 2}{\text{HBM2 interface frequency}} \right) \right) \times 100.
\]
For example, given the following values:

- Write data transactions: 5000
- Read data transactions: 5000
- Total active transactions: 10358
- Core clock frequency: 250 MHz
- HBM2 interface frequency: 600 MHz

Efficiency = \[
\frac{(5000+5000)}{(10358)} \times (250) \times 2 \div 650 \times 100 = 81.3\%
\]

10. The traffic generator runs with the efficiency pattern for one test loop by default. To reinitiate the test, reset the HBM2 design example using the ISSP editor. You can use the RSTN signal to reset the HBM2 design example by writing a 0 and then a 1.

**Note:** For additional information on changing the test duration and traffic pattern for the Intel Stratix 10 HBM2 traffic generator, refer to the *Intel Stratix 10 High Bandwidth Memory (HBM2) Traffic Generator* video, available here: https://www.youtube.com/watch?v=XLfw12VCm0U.

### 2.6. Generating the High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example for Simulation

For the Intel Stratix 10 MX development kit, you may leave most of the High Bandwidth Memory (HBM2) Interface Intel FPGA IP settings at their default values. To generate the design example for simulation, follow these additional steps.

1. On the Diagnostics tab, click Performance ➤ Enable Efficiency Monitor to gather and report statistics on the efficiency of the interface during simulation.

**Note:** If you enable In-System-Sources-and-Probes, it is instantiated only if synthesized by the Intel Quartus Prime software. For simulation, the corresponding signal is tied to either ground or Vcc.
2. On the Diagnostics tab, click Traffic Generator ➤ Use efficiency pattern to provide high-efficiency concurrent traffic pattern features which are integrated in the design example. You can configure the different number of read/write transactions, select between sequential or random traffic addressing, and enable or disable the data-matching check. This option also enables the Read command round trip latency feature. Latency is measured from the start of read command valid by the driver, until the response is received.

3. Ensure that the Simulation checkbox is checked under Example Design Files on the Example Designs tab.

Note: By default, both the Simulation and Synthesis checkboxes are checked.

4. Click File ➤ Save. The system creates and saves the current settings in the user IP variation file (<user instance name>.ip), within the <user project directory> that you specified.
5. Click **Generate Example Design** in the top-right corner of the window.

6. Specify a directory for the generated design example (**<design example directory>**) and click **OK**. The system creates the following design example file sets within the specified directory:
2.7. Regenerating the High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example After Modification

If you modify the IP variation files or Platform Designer files in the post-generation synthesis folder, you can let the Intel Quartus Prime software regenerate the design example files through the integrated IP regeneration process during the Intel Quartus Prime compilation. Alternatively, you can use the design example’s pregeneration files to view, modify, generate, or regenerate the design example files. The design
example's pregeneration scripts are generated when you click the **Generate Example Design** button in the IP Parameter Editor Pro window. The pregeneration scripts are generated regardless of whether you select the **Simulation** or **Synthesis** checkbox.

1. To generate or regenerate the synthesis or simulation file sets, run one of the following scripts:
   - To create a project for compilation, run the `quartus_sh -t make_qii_design.tcl` script in the destination directory.
   - To create a project for simulation, run the `quartus_sh -t make_sim_design.tcl` script in the destination directory.
3. High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example Description

When you parameterize the High Bandwidth Memory (HBM2) Interface Intel FPGA IP, you can have the system generate simulation and synthesis file sets, by selecting Simulation or Synthesis under Example Design Files on the Example Designs tab.

The system creates a complete file set for simulation or for synthesis, in accordance with your selection.

3.1. High Bandwidth Memory (HBM2) Interface Intel FPGA IP Synthesis Design Example

The synthesis design example contains the following major blocks.

- An instance of the High Bandwidth Memory (HBM2) Interface Intel FPGA IP that manages the read, write, and other operations to the HBM2 device.
- Two independent traffic generators for every HBM channel enabled (one traffic generator for each HBM Pseudo-channel). The traffic generator is a synthesizable AXI-4 type 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 I/O PLL to emulate the external core clock that feeds the ext_core_clock port of the HBM2 IP. The clock eventually clocks the Traffic Generator and any other components that are clocked by the wmc_clk_0_clk clock of the HBM2 IP.

Figure 3. Synthesis Design Example

```
 Synthesis Design Example

 External Core Clock I/O PLL

 Traffic Generator CH0 PC0

 Traffic Generator CH0 PC1

 High Bandwidth Memory (HBM2) Interface Intel FPGA IP

 High Bandwidth Memory

 ext_core_clk

 AXI-4

 AXI-4

 UIB
```
If you enable more than one HBM channel, the synthesis design example includes an additional pair of traffic generators based on how many HBM channels are enabled. Because each HBM channel and HBM pseudo-channel are independent, each of the additional traffic generators are also independent from one another, although connected to a single High Bandwidth Memory (HBM2) Interface Intel FPGA IP. The following figure shows an example with HBM channels enabled.

**Figure 4.** Synthesis Design Example with Three HBM Channels Enabled

<table>
<thead>
<tr>
<th>Signal Group</th>
<th>Signal Name</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLL Ref clk Inputs</td>
<td><code>core_clk_iopll_ref_clk_clk</code></td>
<td>Input</td>
<td>1</td>
<td>LVDS differential reference clock used by the I/O PLL to generate the fabric core clock. The design example automatically instantiates the I/O PLL that generates the core clock.</td>
</tr>
<tr>
<td></td>
<td><code>hbm_0_example_design_pll_ref_clk_clk</code></td>
<td>Input</td>
<td>1</td>
<td>LVDS differential reference clock used by the UIB PLL. The design example automatically instantiates the UIB</td>
</tr>
</tbody>
</table>

*continued...*
### Signal Group

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resets</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>core_clk_iopll_reset_reset</td>
<td>Input</td>
<td>1</td>
<td>PLL that generates the clock for the UIB subsystem.</td>
</tr>
<tr>
<td>hbm_0_example_design_wmcrst_n_in_reset_n</td>
<td>Input</td>
<td>1</td>
<td>Reset input for the core clock I/O PLL. The reset polarity is active high.</td>
</tr>
<tr>
<td>hbm_only_reset_in_reset</td>
<td>Input</td>
<td>1</td>
<td>General core logic reset; active low.</td>
</tr>
<tr>
<td><strong>Boundary Scan Signals</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m2u_bridge_cattrip</td>
<td>Input</td>
<td>1</td>
<td>HBM2 boundary signals that are not driven by the traffic generator. These</td>
</tr>
<tr>
<td>m2u_bridge_temp</td>
<td>Input</td>
<td>3</td>
<td>signals must be exposed at the design example top level to enable successful</td>
</tr>
<tr>
<td>m2u_bridge_wso</td>
<td>Input</td>
<td>8</td>
<td>compilation. These signals should not be actively driven. The Intel Quartus Prime</td>
</tr>
<tr>
<td>m2u_bridge_reset_n</td>
<td>Output</td>
<td>1</td>
<td>software places these signals on pins that are connected to the HBM2 memory.</td>
</tr>
<tr>
<td>m2u_bridge_wrst_n</td>
<td>Output</td>
<td>1</td>
<td>Do not add any location assignments to these pins, as doing so causes</td>
</tr>
<tr>
<td>m2u_bridge_wrck</td>
<td>Output</td>
<td>1</td>
<td>compilation errors.</td>
</tr>
<tr>
<td>m2u_bridge_shiftwr</td>
<td>Output</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>m2u_bridge_capturewr</td>
<td>Output</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>m2u_bridge_updatewr</td>
<td>Output</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>m2u_bridge_selectwir</td>
<td>Output</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>m2u_bridge_wsi</td>
<td>Output</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td><strong>Traffic Generator Status Signals</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tgx_0_status Traffic_gen_pass</td>
<td>Output</td>
<td>1</td>
<td>Traffic generator status signals (Pass, Fail and Timeout) for Pseudo Channel 0, per Channel.</td>
</tr>
<tr>
<td>tgx_0_status Traffic_gen_fail</td>
<td>Output</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>tgx_0_status Traffic_gen_timeout</td>
<td>Output</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

*continued...*
### Signal Groups

<table>
<thead>
<tr>
<th>Signal Group</th>
<th>Signal Name</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tgx_1_status_traffic_gen_pass</td>
<td>Output</td>
<td>1</td>
<td></td>
<td>Traffic generator status signals (Pass, Fail and Timeout) for Pseudo Channel 1, per Channel.</td>
</tr>
<tr>
<td>tgx_1_status_traffic_gen_fail</td>
<td>Output</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tgx_1_status_traffic_gen_timeout</td>
<td>Output</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Using the Design Example in a Project with Multiple HBM2 Interfaces

Do not instantiate an HBM2 design example in a project multiple times, because a Fitter error may occur.

To create a project with two HBM2 interfaces, create two instances of the HBM2 IP, one with HBM location = Top and one with HBM2 location = bottom. Generate the design example projects for each IP and then instantiate them in the multiple HBM2 interface project.

### Related Information

Intel Stratix 10 Device Datasheet

### 3.2. High Bandwidth Memory (HBM2) Interface Intel FPGA IP Simulation Design Example

The simulation design example contains the following major blocks.

- The simulation example includes all the major blocks that exist in the synthesis design example, including a traffic generator pair, and an instance of the High Bandwidth Memory (HBM2) Interface Intel FPGA IP and external core clock I/O PLL. These blocks default to abstract simulation models where appropriate for rapid simulation. The design example may also include an Efficiency Monitor block for every HBM channel that you have enabled. The Efficiency Monitor block reports the efficiency number for the particular HBM channel at the end of simulation.
- An HBM2 memory model, which acts as a generic model that conforms to the HBM2 protocol specifications. Frequently, HBM2 vendors provide simulation models for their specific HBM2 components that you can download from their websites.
- A simulation checker, which monitors the status signals from the HBM2 IP and the traffic generator, to signal an overall pass or fail condition.
- The clock source and reset source instances which are the Bus Functional Model (BFM) that generates the reference clock and reset signal.
Simulation Design Example

Note: Unlike other styles of memory interface where the testbench normally wraps the synthesis example design top-level file, for the High Bandwidth Memory (HBM2) Interface Intel FPGA IP, the Traffic Generator and other non-High Bandwidth Memory (HBM2) Interface Intel FPGA IP components are instantiated in the top-level testbench file.

3.3. High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example Interface tab

You can use the Example Designs tab in the parameter editor to parameterize and generate your design examples.
4. High Bandwidth Memory (HBM2) Interface Intel 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.

<table>
<thead>
<tr>
<th>IP Core Version</th>
<th>User Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.1</td>
<td>High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example User Guide</td>
</tr>
<tr>
<td>18.1.1</td>
<td>High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example User Guide</td>
</tr>
</tbody>
</table>
# 5. Document Revision History for the High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example User Guide

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Intel Quartus Prime Version</th>
<th>IP Version</th>
<th>Changes</th>
</tr>
</thead>
</table>
| 2019.08.30       | 19.2                        | 19.2.0     | • Added About the High Bandwidth Memory (HBM2) Interface Intel FPGA IP.  
                   |                             |            | • Added Generating the Synthesizable High Bandwidth Memory (HBM2) Interface Intel FPGA IP for High Efficiency topic.  
                   |                             |            | • Added text to 2 of the Generating the Synthesizable High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example topic.  
                   |                             |            | • Added text to 2 of the Generating the High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example for Simulation topic.  
                   |                             |            | • Updated the figure in 4 of Configuring the High Bandwidth Memory (HBM2) Interface Intel FPGA IP.  
                   |                             |            | • Updated the figures in steps 1 and 3 of Generating the Synthesizable High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example.  
                   |                             |            | • Updated the figures in steps 1 and 3 of Generating the High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example for Simulation. |
| 2019.05.03       | 19.1                        | 19.1       | • Modified the Diagnostic description in Table 1, Tab Parameter Guidelines in the IP Parameter Editor Pro Guidelines for High Bandwidth Memory (HBM2) Interface Intel FPGA IP topic.  
                   |                             |            | • Updated the Diagnostics tab figure and added a step for using the Traffic GeneratorUse efficiency pattern feature in the Generating the Synthesizable High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example topic.  
                   |                             |            | • Updated the Diagnostics tab figure and added a step for using the Traffic GeneratorUse efficiency pattern feature in the Generating the High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example for Simulation topic.  
                   |                             |            | • Added High Bandwidth Memory (HBM2) Interface Intel FPGA IP Design Example User Guide Archives chapter. |