AN 861: Intel® Stratix® 10 H-Tile PCI Express* Link Hardware Validation
1. Intel® Stratix® 10 H-Tile PCIe* Link Hardware Validation

1.1. Overview

The Intel® Stratix® 10 Hard IP for PCI Express* IP core includes a programmed I/O (PIO) design example to help you understand the usage for this IP. The PIO design example transfers data from the host memory to the local memory on a target device, which in this case is an Intel FPGA. This design example is appropriate for low-bandwidth applications. In the PCIe* link hardware validation process, this design example serves as an Endpoint (EP) that interacts with the host, which is the Root Port. The same design example can be used in your hardware to validate if a PCIe link is working as expected. The Intel Stratix 10 GX FPGA Development Kit with H-Tile device is used in the hardware validation process. This validation is performed on both the Linux* and Windows* platforms.

1.2. Hardware Validation System Block Diagram

Figure 1. Hardware Validation System Simplified Block Diagram

The figure above shows the simplified block diagram of the hardware validation system. The Root Port in this system is the host PC, which interfaces with the PIO design example to perform memory reads and memory writes on an on-chip memory without the use of DMA.

The three main components of the design example are:

1. Device-Under-Test (DUT): This refers to the PCIe Hard IP (HIP), which you can configure.
2. PIO Application: This takes in the Avalon®-ST data and converts it to the Avalon-MM format before sending it to the slave.
3. The slave in this case is an on-chip memory with a size that matches the DUT’s BAR size.

For the host (or Root Port), a driver and application software are provided for the validation process. After the driver is installed, the hardware will be initialized and ready for transactions. Data is then packetized, and can be received by the DUT. In the validation process, the application software performs Memory Writes to the on-chip memory.
chip memory followed by Memory Reads from the on-chip memory. It then compares the written data with the read data. A matching dataset indicates a successful validation.

1.3. Hardware and Software Requirements

The following sections describe the hardware and software used to perform the hardware validation.

1.3.1. Hardware Requirements

- Intel Stratix 10 GX Development Kit (1SG280HU2F50E2VG)
- Hewlett Packard Z620 workstation with CentOS* 6.0
- Hewlett Packard Z620 workstation with Windows* 7

Figure 2. Intel Stratix 10 GX FPGA Development Kit

1.3.2. Software Requirements

- Intel Quartus® Prime Pro Edition version 18.0
- CentOS 6.0 with Linux Kernel version 2.6.32-696
- Windows 7
- Linux driver generated together with the example design
- Intel-developed PCIe Windows Demo Driver
1.4. Quick Start Guide

1.4.1. Hardware Setup

For the hardware setup to run the reference design, follow these steps:

1. Connect the Intel Stratix 10 GX FPGA Development Kit into the x16 PCIe slot of the motherboard.
2. Connect the USB cable to the Micro USB Blaster connector on the development kit.
3. Connect the power adapter (packaged together with the development board) to the power supply jack.
4. Connect the Auxiliary power from the motherboard to the development kit.
5. Power on the workstation.
6. Turn on the power for the Intel Stratix 10 GX FPGA Development Kit. The hardware system is now ready for programming.

Related Information
Intel Stratix 10 GX FPGA Development Kit

1.4.2. Running the Reference Design

1.4.2.1. Generating the Design

1. In the Intel Quartus Prime Pro Edition software, create a new project (go to File, and choose New Project Wizard).
2. Specify the Directory, Name, and Top-Level Entity.
3. For Project Type, accept the default value, Empty project. Click Next.
4. For Add Files, click **Next**.
5. For Family, Device & Board Settings, select Intel Stratix 10 for Family. Then select the Target Device for your design from the list of Available Devices.
6. Click **Finish**.
7. In the IP Catalog, locate and add the **Avalon-ST Intel Stratix 10 Hard IP for PCI Express**.
8. In the New IP Variant dialogue box, specify a name for your IP.
9. On the IP Settings tabs, specify the parameters for your IP variation.
10. On the Example Designs tab, make the following selections:
   a. **For Available Example Designs**, select **PIO**.
   b. **For Example Design Files**, turn on the **Simulation** and **Synthesis** options.
   c. If you have selected a x16 configuration, for **Select Simulation Root Complex BFM**, choose the appropriate Bus Functional Model (BFM):
      - **Intel FPGA BFM**: for all configurations up to Gen3 x8. This BFM supports x16 configurations by downtraining to x8.
      - **Third-party BFM**: for x16 configurations if you want to simulate all 16 lanes. Refer to **AN-811: Using the Avery BFM for PCI Express Gen3x16 Simulation on Intel Stratix 10 Devices** for information about simulating with the Avery BFM.
   d. **For Generated HDL Format**, only Verilog is available in the current release.
   e. **For Target Development Kit**, select the appropriate option. Note: If you select **None**, the generated design example targets the device specified. If you intend to test the design in hardware, make the appropriate pin assignments in the .qsf file.
11. Select **Generate Example Design** to create a design example that you can simulate and download to hardware. If you select one of the Intel Stratix 10 development boards, the device on that board overwrites the device previously selected in the Intel Quartus Prime project if the devices are different. When the prompt asks you to specify the directory for your example design, accept the default directory, / pcie_s10_hip_ast_0_example_design.
12. Close the IP Parameter Editor followed by the project.

1.4.2.2. Compiling the Design Example and Programming the Device

1. Navigate to `<project_dir>/pcie_s10_hip_ast_0_example_design/` and open `pcie_example_design.qpf`.
2. On the Processing menu, select **Start Compilation**.
3. After compilation, program the targeted device with the Programmer.
4. After successfully programming the device, reboot the workstation.

1.4.3. Hardware Validation

Hardware validation can be performed on either the Linux or Windows platforms.

1.4.3.1. Hardware Validation on the Linux Platform

1.4.3.1.1. Linux Kernel Driver

A Linux kernel driver is generated together with the design example at `./software/user/example` under the design example generation directory. You can use this driver to perform the following tests:

- A PCIe link test that performs 100 writes and reads
- Memory space DWORD reads and writes
- Configuration space DWORD reads and writes
In addition, you can use the driver to change the value of the following parameters:

- The BAR
- The bus, device and function (BDF) numbers to specify the selected device

### 1.4.3.1.2. Installing the Linux Kernel Driver

To install the kernel driver, do the following:

1. Navigate to `./software/kernel/Linux` under the design example generation directory.
2. Change the permissions on the install, load, and unload files with the following command:
   ```bash
   $ chmod 777 install load unload
   ```
3. Install the driver with the following command:
   ```bash
   $ sudo ./install
   ```
4. Verify the driver installation with the following command:
   ```bash
   $ lsmod | grep intel_fpga_pcie_drv
   ```
   An example of a returned result is "intel_fpga_pcie_drv 17792 0".
5. Verify the PCIe configuration space information with the following command:
   ```bash
   $ lspci -vv
   ```
   The following figure shows an example of PCIe configuration space information.

---

Figure 5. An example of PCIe configuration space information

---

6. Verify that Linux recognizes the PCIe design example with the following command:
   ```bash
   $ lspci -d 1172:000 -v | grep intel_fpga_pcie_drv
   ```
   The expected result returned should be "Kernel driver in use: intel_fpga_pcie_drv" as shown in the following figure.
1. Intel® Stratix® 10 H-Tile PCIe® Link Hardware Validation

Figure 6. Verify the Linux kernel driver in use

```bash
[root@GAE linux]# lspci -d 1172:0000 -v | grep intel_fpga_pcie_drv
Kernel driver in use: intel_fpga_pcie_drv
```

7. Verify the BDF of the Intel Stratix 10 GX Development Kit with the following command:

```bash
$ lspci -d 1172:0000
```

The figure below shows an example of the BDF results.

Figure 7. BDF information of the Intel Stratix 10 GX Development Kit

---

1.4.3.1.3. Running the Design Example Application

1. Navigate to `.\software\user\example` under the design example directory.
2. Compile the design example application with the following command:

```bash
$make
```
3. Run the test with the following command:

```bash
$ ./intel_fpga_pcie_link_test
```

By default, you can run the Intel FPGA IP PCIe link test in manual or automatic mode. In this application note, the Intel FPGA IP PCIe link test is run in automatic mode. The following figure shows an example of passing results with 0 write or read error.

Figure 8. Passing results for automatic mode with 0 write or read error

---

1.4.3.2. Hardware Validation on the Windows Platform

1.4.3.2.1. Installing the Windows driver

This section describes the steps to set up the Windows driver for hardware validation on the Windows platform.
1. Download the driver from the following alterawiki link:
2. Unzip the Altera_PCIEInterop_Test.zip.
3. Open the Windows Device Manager and scan for hardware changes.
4. Select the Intel FPGA listed as an unknown PCI* device and point to the appropriate 32- or 64-bit driver (`altera_pcie_win_driver.intf`) in the Windows_driver directory.
5. After the driver loads successfully, a new device named Altera PCI API Device should appear in the Windows Device Manager.
6. You can determine the BDF number for the Altera PCI API Device listed in the Windows Device Manager by doing the following:
   a. Expand the tab Altera PCI API Driver under the devices.
   b. Right click on Altera PCI API Device and select Properties.
   c. Take note of the bus, device and function numbers for the device. Figure 10 shows an example of the bus, device and function numbers for the new PCIe device.

**Figure 9.** An example of BDF numbers for a new PCIe device in Device Manager
1.4.3.2.2. Running the Windows demo driver with design example

The software application to test the PCI Express design example on the Intel Stratix 10 FPGA Development Kit is available on both 32- and 64-bit Windows platforms. This program performs the following tasks:

1. Print the configuration space, lane rate and lane width.
2. Write 0x00000000 to the specified BAR at offset 0x00000000 to initialize the memory and read it back.
3. Write 0xABCD1234 at offset 0x00000000 of the specified BAR, then read it back and compare.

If successful, the test program displays the message "PASSED".

Follow these steps to perform the PCIe link test with the Windows demo driver with the design example in hardware:

1. In the Interop_software folder, run the Alt_Test.exe.
2. When prompted, enter the bus, device and function numbers for your Altera PCI API Driver as shown in the following figure.
3. Enter the BAR number (0 - 5) that you specified during the parameterization of the Avalon-ST Intel Stratix 10 Hard IP for PCI Express IP core.

4. The Interop software application will proceed with the PCIe link test.

5. If the test is successful, you will see the message "PASSED" as shown in the figure below.

Note: The bus, device and function numbers for your hardware setup may be different.
Figure 11. "PASSED" message to indicate a successful link test

Note: There is a known issue with this Windows demo driver. The lane rate and link width are incorrectly displayed as Lane Rate = 0 and Link Width = 00. This issue will be fixed in the future release of the Windows demo driver.

1.5. Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018.06.29</td>
<td>18.0.1</td>
<td>Initial revision for Intel Stratix 10 H-Tile.</td>
</tr>
</tbody>
</table>