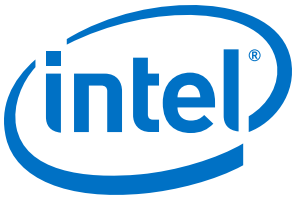


Intel® FPGA P-Tile Avalon® Streaming (Avalon-ST) IP for PCI Express* Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.4**

IP Version: **1.1.0**

Subscribe

Send Feedback

UG-20234 | 2019.12.16

Latest document on the web: **PDF** | **HTML**

Contents

1. Design Example Description....3
1.1. Functional Description for the Programmed Input/Output (PIO) Design Example.... 3
1.2. Functional Description for the Single Root I/O Virtualization (SR-IOV) Design Example....4
1.3. Serial Data Signals 5
1.4. Registers....6

2. Quick Start Guide....7
2.1. Directory Structure....8
2.2. Generating the Design Example....9
2.3. Simulating the Design Example....10
2.4. Compiling the Design Example....11
2.5. Installing the Linux Kernel Driver.... 12
2.6. Running the Design Example Application....13

A. Document Revision History for the Intel P-Tile Avalon-ST Hard IP for PCIe Design Example User Guide....15
A.1. Intel P-Tile Avalon Streaming (Avalon-ST) IP for PCIe Design Example User Guide Revision History....15

1. Design Example Description

1.1. Functional Description for the Programmed Input/Output (PIO) Design Example

The PIO design example performs memory transfers from a host processor to a target device. In this example, the host processor requests single-dword MemRd and MemWr TLPs.

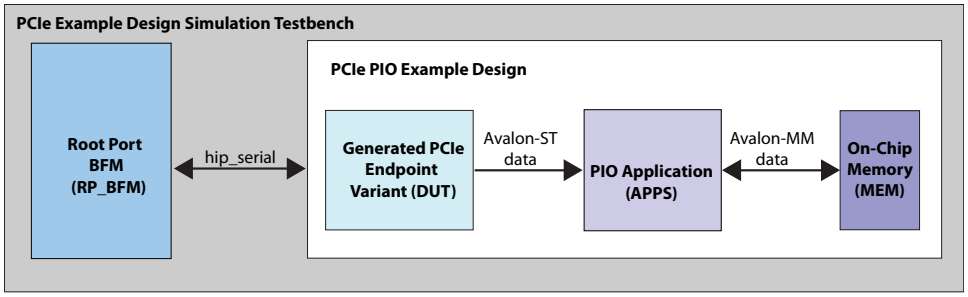
The PIO design example automatically creates the files necessary to simulate and compile in the Intel® Quartus® Prime software. The design example covers a wide range of parameters. However, it does not cover all possible parameterizations of the P-Tile Hard IP for PCIe.

This design example includes the following components:

- The generated P-Tile Avalon-ST Hard IP Endpoint variant (DUT) with the parameters you specified. This component drives TLP data received to the PIO application.
- The PIO Application (APPS) component, which performs the necessary translation between the PCI Express TLPs and simple Avalon-MM writes and reads to the on-chip memory.
- An on-chip memory (MEM) component.

The simulation testbench instantiates the PIO design example and a Root Port BFM to interface with the target Endpoint.

Figure 1. Block Diagram for the Platform Designer PIO Design Example Simulation Testbench



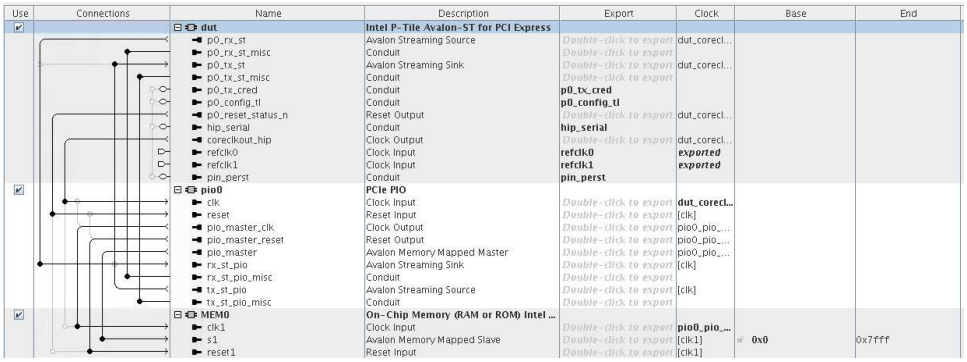
The test program writes to and reads back data from the same location in the on-chip memory. It compares the data read to the expected result. The test reports, "Simulation stopped due to successful completion" if no errors occur.

The P-Tile Avalon®-ST design example supports the following configurations:

- Gen4 x16 Endpoint
- Gen3 x16 Endpoint
- Gen4 x8 Endpoint
- Gen3 x8 Endpoint

Figure 2. Platform Designer System Contents for P-Tile Avalon-ST PCI Express PIO Design Example

The Platform Designer generates this design for up to Gen4 x16 variants.



1.2. Functional Description for the Single Root I/O Virtualization (SR-IOV) Design Example

The SR-IOV design example performs memory transfers from a host processor to a target device. It supports up to two PFs and 32 VFs per PF.

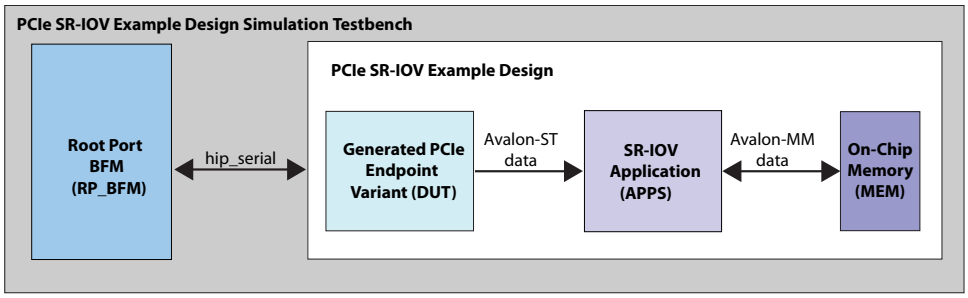
The SR-IOV design example automatically creates the files necessary to simulate and compile in the Intel Quartus Prime software. You can download the compiled design to an Intel Stratix® 10 DX Development Kit.

This design example includes the following components:

- The generated P-Tile Avalon Streaming (Avalon-ST) IP Endpoint variant (DUT) with the parameters you specified. This component drives the received TLP data to the SR-IOV application.
- The SR-IOV Application (APPS) component, which performs the necessary translation between the PCI Express TLPs and simple Avalon-ST writes and reads to the on-chip memory.
- An on-chip memory (MEM) component.

The simulation testbench instantiates the SR-IOV design example and a Root Port BFM to interface with the target Endpoint.

Figure 3. Block Diagram for the Platform Designer SR-IOV Design Example Simulation Testbench

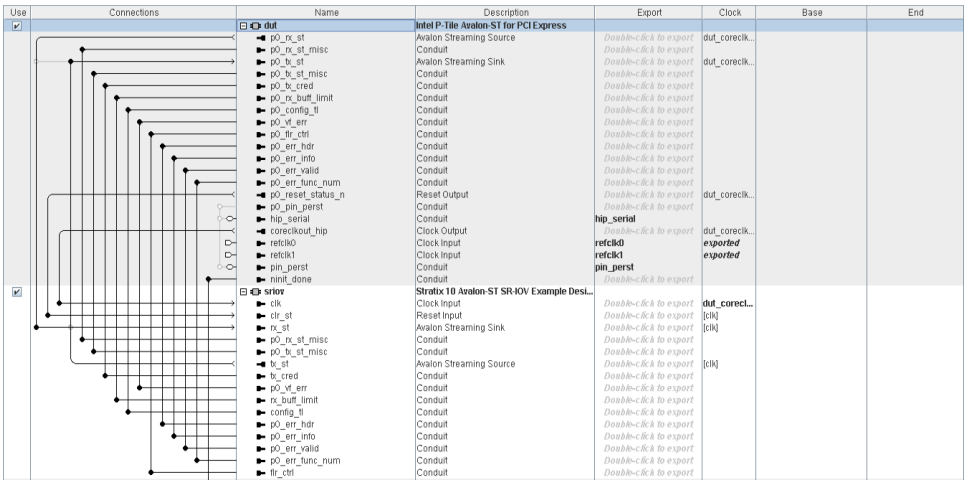


The test program writes to and reads back data from the same location in the on-chip memory across 2 PFs and 32 VFs per PF. It compares the data read to the expected result. The test reports, "Simulation stopped due to successful completion" if no errors occur.

The SR-IOV design example supports the following configurations:

- Gen4 x16 Endpoint
- Gen3 x16 Endpoint

Figure 4. Platform Designer System Contents for P-Tile Avalon-ST with SR-IOV for PCI Express Design Example



1.3. Serial Data Signals

This differential, serial interface is the physical link between a Root Port and an Endpoint.

The P-Tile PCIe IP Core supports 4, 8, or 16 lanes. Each lane includes a TX and RX differential pair. Data is striped across all available lanes.

Table 1. Serial Data Interface Signals

In the following table <*n*> is the number of lanes.

Signal	Direction	Description
tx_(p/n)_out[<*n*>-*1*:0]	Output	Transmit output. These signals are the serial outputs of lanes <*n*>-*1*–0.
rx_(p/n)_in[<*n*>-*1*:0]	Input	Receive input. These signals are the serial inputs of lanes <*n*>-*1*–0.

Refer to *Pin-out Files for Intel Devices* for pin-out tables for all Intel devices in **.pdf**, **.txt**, and **.xls** formats.

Related Information

Pin-out Files for Intel Devices

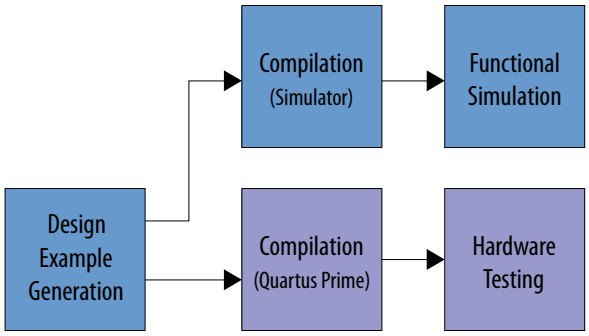
1.4. Registers

There are no control registers for the PIO design example. The *PCI Express Base Specification 4.0* defines a comprehensive set of configuration, control, and status registers to control and debug the design example.

2. Quick Start Guide

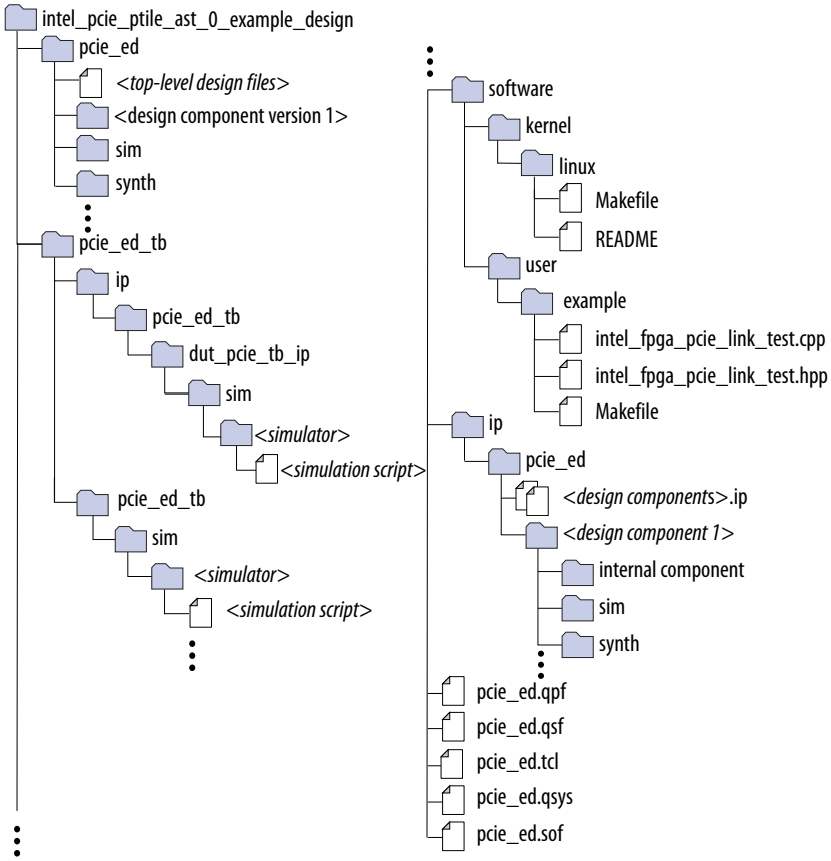
Using Intel Quartus Prime software, you can generate a programmed I/O (PIO) design example for the Intel FPGA P-Tile Avalon-ST Hard IP for PCI Express* IP core. The generated design example reflects the parameters that you specify. The PIO example transfers data from a host processor to a target device. It is appropriate for low-bandwidth applications. This design example automatically creates the files necessary to simulate and compile in the Intel Quartus Prime software. You can download the compiled design to your FPGA Development Board. To download to custom hardware, update the Intel Quartus Prime Settings File (`.qsf`) with the correct pin assignments .

Figure 5. Development Steps for the Design Example



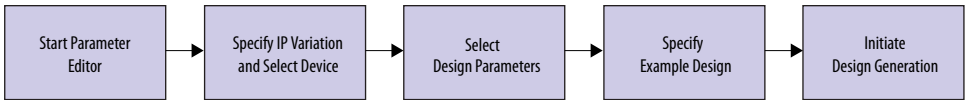
2.1. Directory Structure

Figure 6. Directory Structure for the Generated Design Example



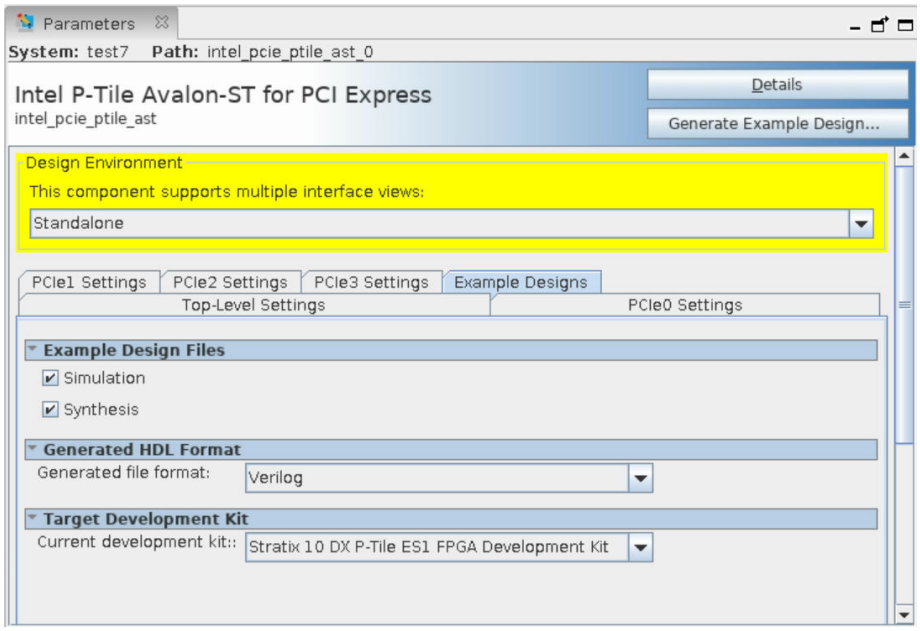
2.2. Generating the Design Example

Figure 7. Procedure



1. In the Intel Quartus Prime Pro Edition software, create a new project (**File ➤ 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** under **Family**, select **Intel Agilex™** or **Intel Stratix 10**.
6. If you selected **Intel Stratix 10** in the last step, select **Stratix 10 DX** in the **Device** pull-down menu.
7. Select the **Target Device** for your design.
8. Click **Finish**.
9. In the IP Catalog locate and add the **Intel P-Tile Avalon-ST Hard IP for PCI Express**.
10. In the **New IP Variant** dialog box, specify a name for your IP. Click **Create**.
11. On the **Top-Level Settings** and **PCIe* Settings** tabs, specify the parameters for your IP variation. If you are using the SR-IOV design example, do the following steps to enable SR-IOV:
    a. On the **PCIe0 Device** tab under the **PCIe0 PCI Express / PCI Capabilities** tab, check the box **Enable multiple physical functions**.
    b. On the **PCIe0 Multifunction and SR-IOV System Settings** tab, check the box **Enable SR-IOV support** and specify the number of PFs and VFs.
    c. On the **PCIe0 MSI-X** tab under the **PCIe0 PCI Express / PCI Capabilities** tab, enable the MSI-X feature as required.
    d. On the **PCIe0 Base Address Registers** tab, enable BAR0 for both PF and VF.
12. On the **Example Designs** tab, make the following selections:
    a. For **Example Design Files**, turn on the **Simulation** and **Synthesis** options. If you do not need these simulation or synthesis files, leaving the corresponding option(s) turned off significantly reduces the example design generation time.
    b. For **Generated HDL Format**, only Verilog is available in the current release.
13. Select **Generate Example Design** to create a design example that you can simulate and download to hardware. If you select one of the P-Tile 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, you can accept the default directory, `./intel_pcie_ptile_ast_0_example_design`, or choose another directory.

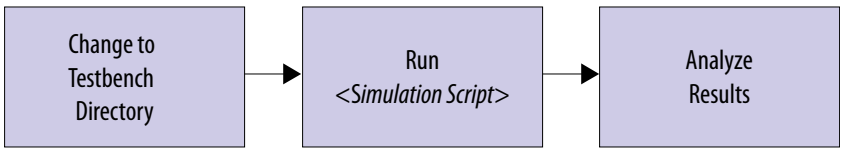
Figure 8. Example Designs Tab



14. Click **Finish**. You may save your `.ip` file when prompted, but it is not required to be able to use the example design.
15. Open the example design project.
16. Compile the example design project to generate the `.sof` file for the complete example design. This file is what you download to a board to perform hardware verification.
17. Close your example design project.

2.3. Simulating the Design Example

Figure 9. Procedure



1. Change to the testbench simulation directory, `pcie_ed_tb`.
2. Run the simulation script for the simulator of your choice. Refer to the table below.
3. Analyze the results.

Table 2. Steps to Run Simulation

Simulator	Working Directory	Instructions
VCS*	<example_design>/pcie_ed_tb/pcie_ed_tb/sim/synopsys/vcs	1. Type `sh vcs_setup.sh USER_DEFINED_COMPILE_OPTIONS="" USER_DEFINED_ELAB_OPTIONS="-xlrm\ uniq_prior_final" USER_DEFINED_SIM_OPTIONS=""` *Note:* The command above is a single-line command. 2. A successful simulation ends with the following message, "Simulation stopped due to successful completion!" *Note:* To run a simulation in interactive mode, use this command: `sh vcs_setup.sh USER_DEFINED_COMPILE_OPTIONS="-debug_access+all -debug_pp" USER_DEFINED_ELAB_OPTIONS="-xlrm\ uniq_prior_final" USER_DEFINED_SIM_OPTIONS="-gui"`

This testbench simulates up to a Gen4 x16 variant.

The simulation reports, "Simulation stopped due to successful completion" if no errors occur.

2.4. Compiling the Design Example

1. Navigate to `<project_dir>/intel_pcie_ptile_ast_0_example_design/` and open `pcie_ed.qpf`.
2. If you are using the Intel Stratix 10 DX development kit, add the following assignments in the `.qsf` file of your project for the VID feature:
   - set_global_assignment -name USE_CONF_DONE SDM_IO16
   - set_global_assignment -name VID_OPERATION_MODE "PMBUS MASTER"
   - set_global_assignment -name USE_PWRMGT_SCL SDM_IO0
   - set_global_assignment -name USE_PWRMGT_SDA SDM_IO12
   - set_global_assignment -name PWRMGT_BUS_SPEED_MODE "100 KHZ"
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE_TYPE OTHER
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE0_ADDRESS 60
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE1_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE2_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE3_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE4_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE5_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE6_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE7_ADDRESS 00
   - set_global_assignment -name PWRMGT_VOLTAGE_OUTPUT_FORMAT "DIRECT FORMAT"
   - set_global_assignment -name PWRMGT_DIRECT_FORMAT_COEFFICIENT_M 1

- set_global_assignment -name PWRMGT_DIRECT_FORMAT_COEFFICIENT_R 0
- set_global_assignment -name PWRMGT_TRANSLATED_VOLTAGE_VALUE_UNIT MILLIVOLTS
- set_global_assignment -name PWRMGT_PAGE_COMMAND_ENABLE ON

3. If you are using the Intel Agilex development kit, add the following assignments in the `.qsf` file of your project for the VID feature:
   - set_global_assignment -name USE_CONF_DONE SDM_IO16
   - set_global_assignment -name USE_INIT_DONE SDM_IO0
   - set_global_assignment -name USE_CVP_CONFDONE SDM_IO15
   - set_global_assignment -name VID_OPERATION_MODE "PMBUS MASTER"
   - set_global_assignment -name USE_PWRMGT_SCL SDM_IO14
   - set_global_assignment -name USE_PWRMGT_SDA SDM_IO11
   - set_global_assignment -name PWRMGT_BUS_SPEED_MODE "100 KHZ"
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE_TYPE OTHER
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE0_ADDRESS 47
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE1_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE2_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE3_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE4_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE5_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE6_ADDRESS 00
   - set_global_assignment -name PWRMGT_SLAVE_DEVICE7_ADDRESS 00
   - set_global_assignment -name PWRMGT_TRANSLATED_VOLTAGE_VALUE_UNIT VOLTS
   - set_global_assignment -name PWRMGT_PAGE_COMMAND_ENABLE ON
4. On the Processing menu, select **Start Compilation**.

2.5. Installing the Linux Kernel Driver

Before you can test the design example in hardware, you must install the Linux kernel driver. You can use this driver to perform the following tests:

- A PCIe link test that performs 100 writes and reads
- Memory space DWORD(1) reads and writes
- Configuration Space DWORD reads and writes

(1) Throughout this user guide, the terms word, DWORD and QWORD have the same meaning that they have in the PCI Express Base Specification. A word is 16 bits, a DWORD is 32 bits, and a QWORD is 64 bits.

In addition, you can use the driver to change the value of the following parameters:

- The BAR being used
- The selects device by specifying the bus, device and function (BDF) numbers for the required device

Complete the following steps to install the kernel driver:

1. Navigate to `./software/kernel/linux` under the example design generation directory.
2. Change the permissions on the `install`, `load`, and `unload` files:

   ```
   $ chmod 777 install load unload
   ```
3. Install the driver:

   ```
   $ sudo ./install
   ```
4. Verify the driver installation:

   ```
   $ lsmod | grep intel_fpga_pcie_drv
   ```

   Expected result:

   ```
   intel_fpga_pcie_drv 17792 0
   ```
5. Verify that Linux recognizes the PCIe design example:

   ```
   $ lspci -d 1172:000 -v | grep intel_fpga_pcie_drv
   ```

   Note: If you have changed the Vendor ID, substitute the new Vendor ID for Intel's Vendor ID in this command.

   Expected result:

   ```
   Kernel driver in use: intel_fpga_pcie_drv
   ```

2.6. Running the Design Example Application

1. Navigate to `./software/user/example` under the design example directory.
2. Compile the design example application:

   ```
   $ make
   ```
3. Run the test:

   ```
   $ sudo ./intel_fpga_pcie_link_test
   ```

   You can run the Intel FPGA IP PCIe link test in manual or automatic mode.

   — In automatic mode, the application automatically selects the device. The test selects the Intel PCIe device with the lowest BDF by matching the Vendor ID. The test also selects the lowest available BAR.

   — In manual mode, the test queries you for the bus, device, and function number and BAR.

   For the Intel Stratix 10 DX or Intel Agilex Development Kit, you can determine the BDF by typing the following command:

```
$ lspci -d 1172
```

4. Here are sample transcripts for automatic and manual modes:

```
Intel FPGA PCIe Link Test - Automatic Mode
Version 2.0
0: Automatically select a device
1: Manually select a device
**************************************************
>0
Opened a handle to BAR 0 of a device with BDF 0x100
**************************************************
0: Link test - 100 writes and reads
1: Write memory space
2: Read memory space
3: Write configuration space
4: Read configuration space
5: Change BAR
6: Change device
7: Enable SR-IOV
8: Do a link test for every enabled virtual function
   belonging to the current device
9: Perform DMA
10: Quit program
**************************************************
> 0
Doing 100 writes and 100 reads . .
Number of write errors:     0
Number of read errors:      0
Number of DWORD mismatches: 0
```

```
Intel FPGA PCIe Link Test - Manual Mode
Version 1.0
0: Automatically select a device
1: Manually select a device
**************************************************
> 1
Enter bus number:
> 1
Enter device number:
> 0
Enter function number:
> 0
BDF is 0x100
Enter BAR number (-1 for none):
> 4
Opened a handle to BAR 4 of a device with BDF 0x100
```

Related Information

PCIe Link Inspector Overview

Use the PCIe Link Inspector to monitor the link at the Physical, Data Link and Transaction Layers.

A. Document Revision History for the Intel P-Tile Avalon-ST Hard IP for PCIe Design Example User Guide

A.1. Intel P-Tile Avalon Streaming (Avalon-ST) IP for PCIe Design Example User Guide Revision History

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.12.16	19.4	1.1.0	Added SR-IOV design example description.
2019.11.13	19.3	1.0.0	Added Gen4 x8 Endpoint and Gen3 x8 Endpoint to the list of supported configurations.
2019.05.03	19.1.1	1.0.0	Initial release.