

Avnet Zynq Mini Module Plus PCIe Endpoint Design



**Version 1.0
December 2016**

1 Introduction

The objective of this reference design is to show how to use the CORE Generator PCIe Wizard to design with the Zynq-7000 PCIe Endpoint Block, add user interface to the PCIe Endpoint Block backend interface, and also to show how to verify the operation of a Zynq-7000 PCIe Endpoint design in a system.

2 Reference Design Requirements

This reference design will require the following software and hardware setups.

2.1 Software

The software requirements for this reference design are:

- Xilinx Vivado 2016.4
- PCITree and PCIECV application software

2.2 Hardware

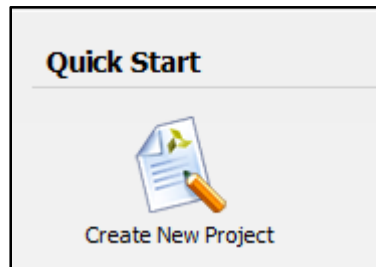
The hardware setup for this reference design is:

- Computer with 4 GB RAM and 1 GB virtual memory (recommended)
- Avnet Zynq Mini Module Plus
- Avnet Mini Module Plus Baseboard 2
- Power supply Module
- 12V power supply
- USB A-mini-B cable

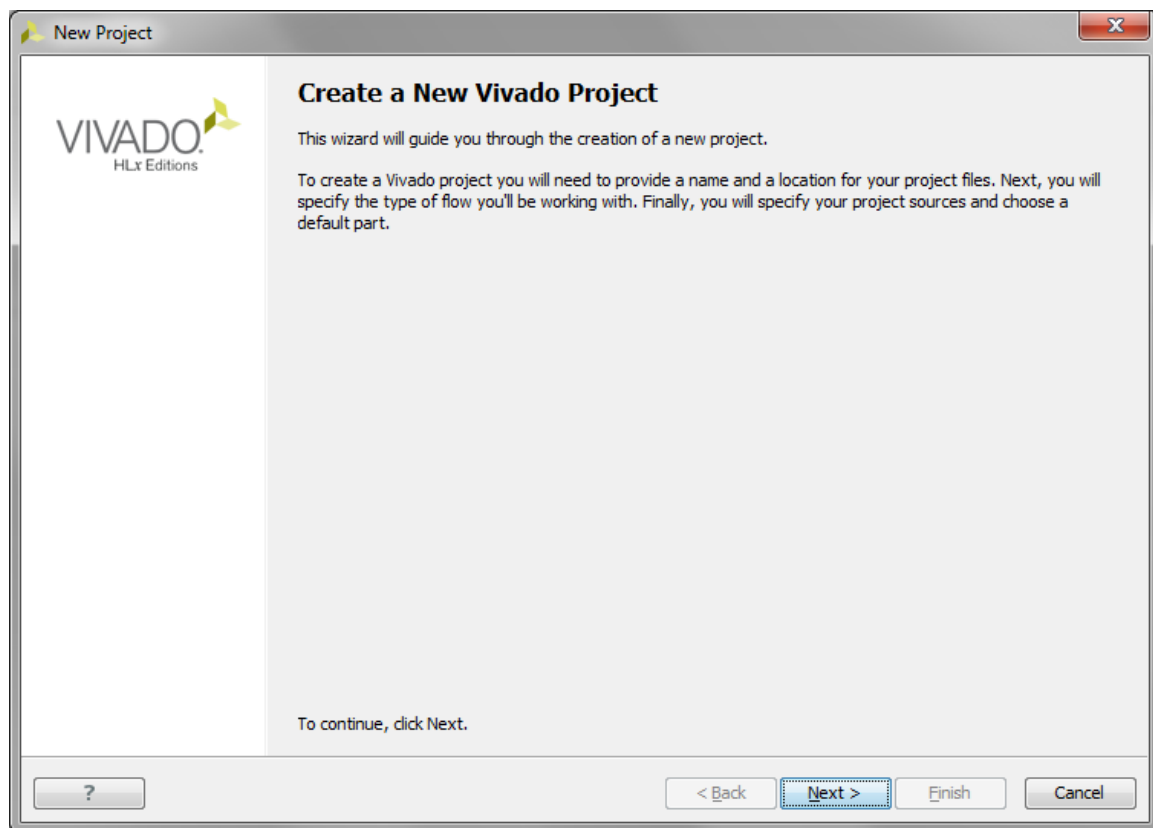
3 Using the Xilinx Core Generator PCIe Wizard

This reference design is developed to assist the users in using the Xilinx Vivado PCIe Wizard to design with the Zynq-7000 PCIe Endpoint Block. The Vivado tool outputs a reference design called Programmed Input/Output (PIO) that can be used to verify the operation of the Zynq-7000 PCIe Endpoint Block in a system.

1. Start Vivado 2016.4 via **Start > All Programs > Xilinx Design Tools > Vivado 2016.4 > Vivado 2016.4**.
2. Click on the **Create New Project** as shown in the following figure.

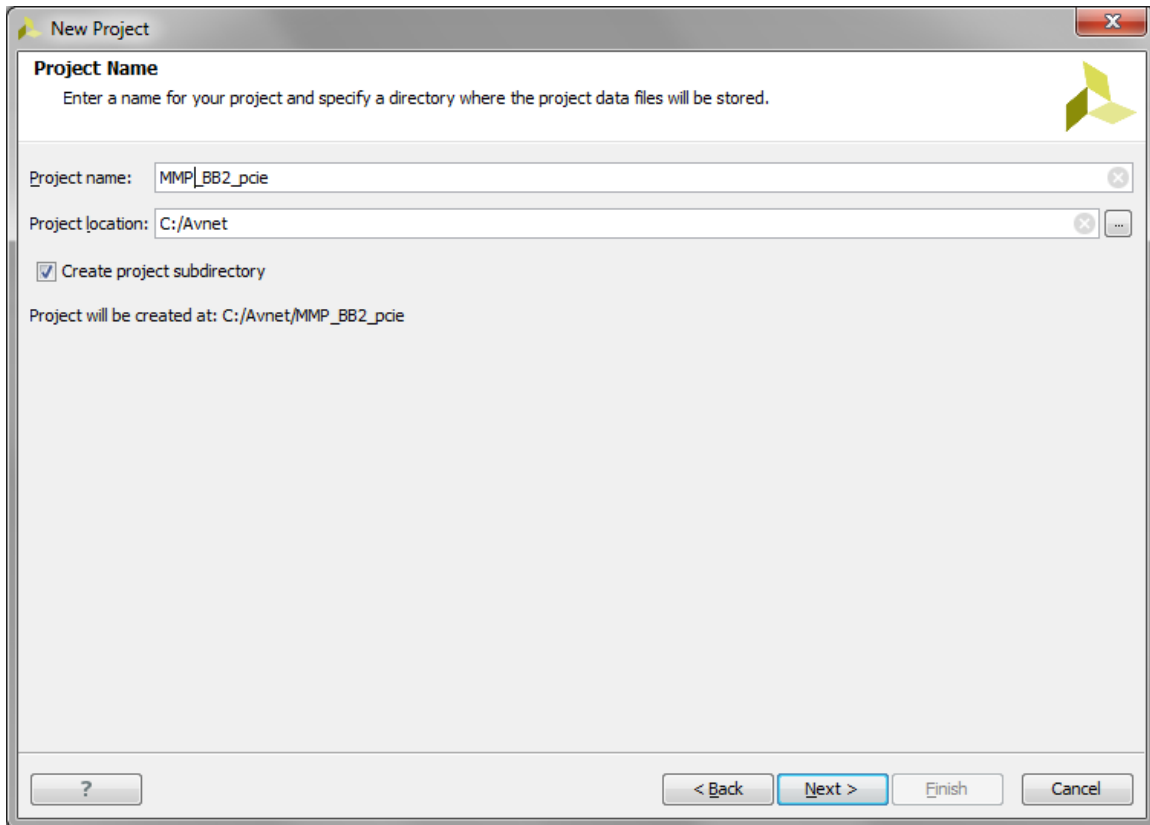


3. In the following figure, click **Next** to continue.



4. Specify **MMP_BB2_pcie** as project name and make sure the **Create project subdirectory** box is checked as shown in the following figure. Click **Next** to continue

Note: Project can be stored on any hard drive such as the D drive in this example, but keep your path as short as possible.



New Project

Project Name
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: MMP_BB2_pcie

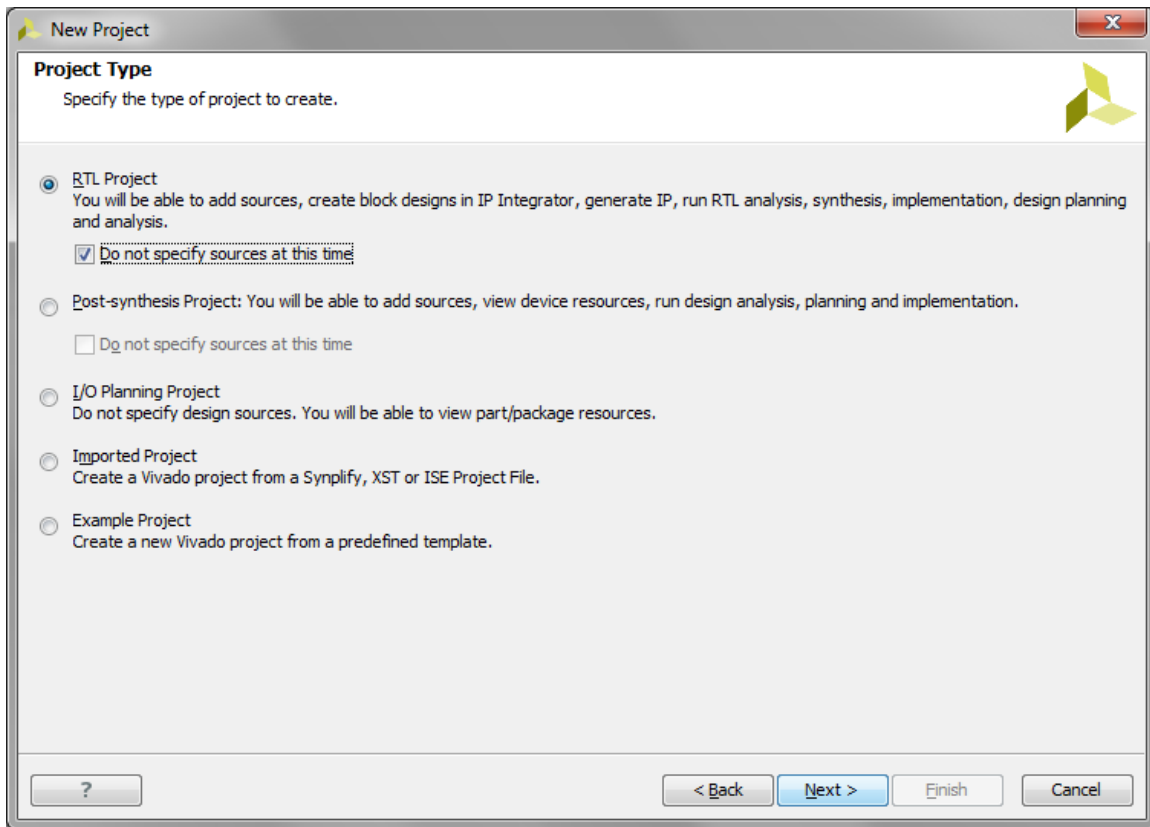
Project location: C:/Avnet

☒ Create project subdirectory

Project will be created at: C:/Avnet/MMP_BB2_pcie

? < Back Next > Finish Cancel

5. Specify the project type as shown in the following figure and click **Next** to continue.



6. If you are using the Zynq MMP XC7Z045 version, please set the device information as shown in the following figure, click on the **xc7z045ffg900-1** device to highlight it, and click **Next** to continue.

New Project

Default Part
Choose a default Xilinx part or board for your project. This can be changed later.

Select: **Parts** Boards

Filter

Product category: All Speed grade: -1
Family: Zynq-7000 Temp grade: All Remaining
Package: ffg900

Reset All Filters

Search:

Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	GTXE2 Transceivers	Gb Transceivers	Avg IOB
xc7z035ffg900-1	900	500	900	343800	0	16	16	362
xc7z045ffg900-1	900	545	900	437200	0	16	16	362
xc7z100ffg900-1	900	755	2020	554800	0	16	16	362

Next >

7. If you are using the Zynq MMP XC7Z100 version, please set the device information as shown in the following figure, click on the **xc7z100ffg900-2** device to highlight it, and click **Next** to continue.

New Project

Default Part
Choose a default Xilinx part or board for your project. This can be changed later.

Select: **Parts** ☐ Boards ☐

Filter

Product category: All Speed grade: -1
Family: Zynq-7000 Temp grade: All Remaining
Package: ffg900

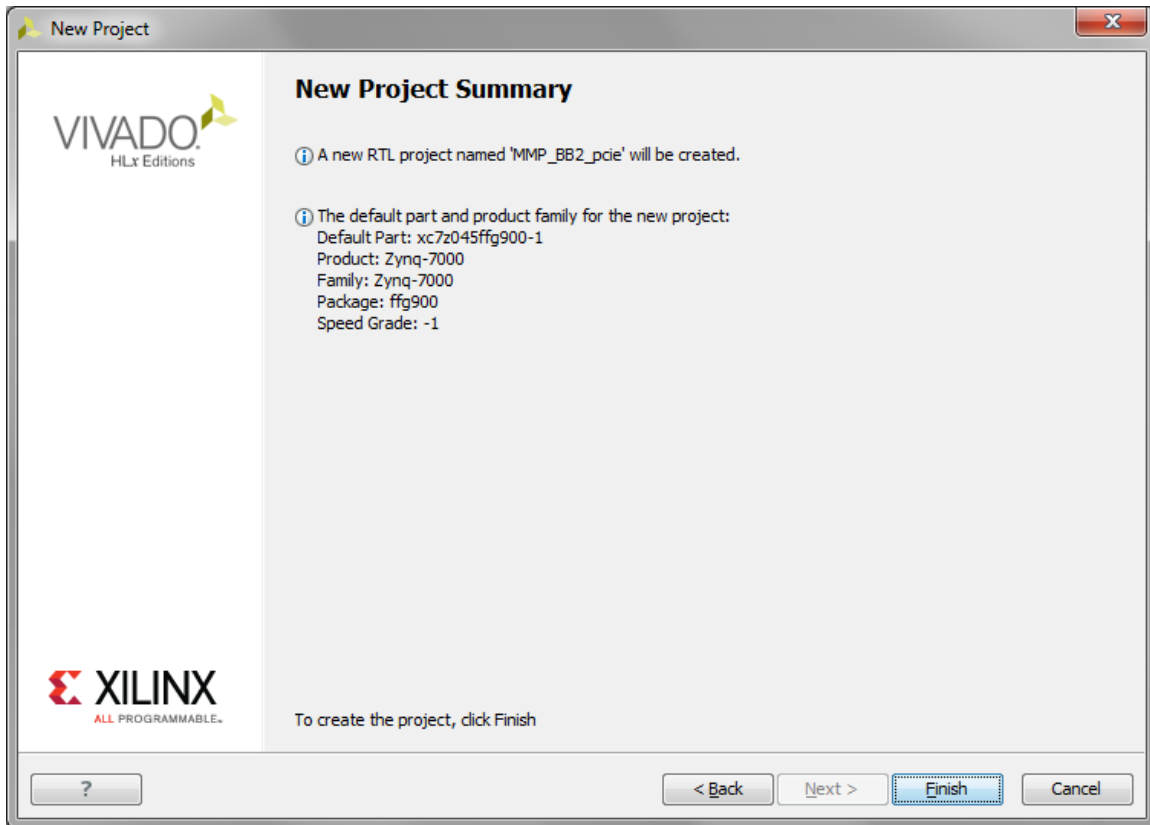
Reset All Filters

Search:

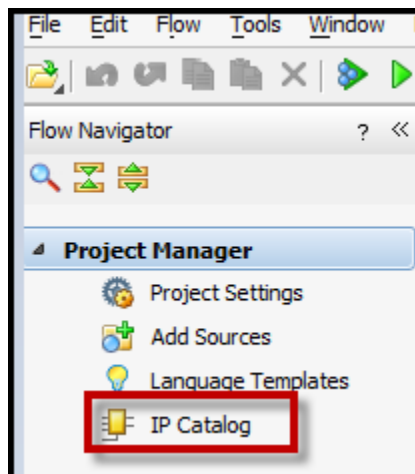
Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	GTXE2 Transceivers	Gb Transceivers	Average IOB
xc7z035ffg900-1	900	500	900	343800	0	16	16	362
xc7z045ffg900-1	900	545	900	437200	0	16	16	362
xc7z100ffg900-1	900	755	2020	554800	0	16	16	362

? < Back Next > Finish Cancel

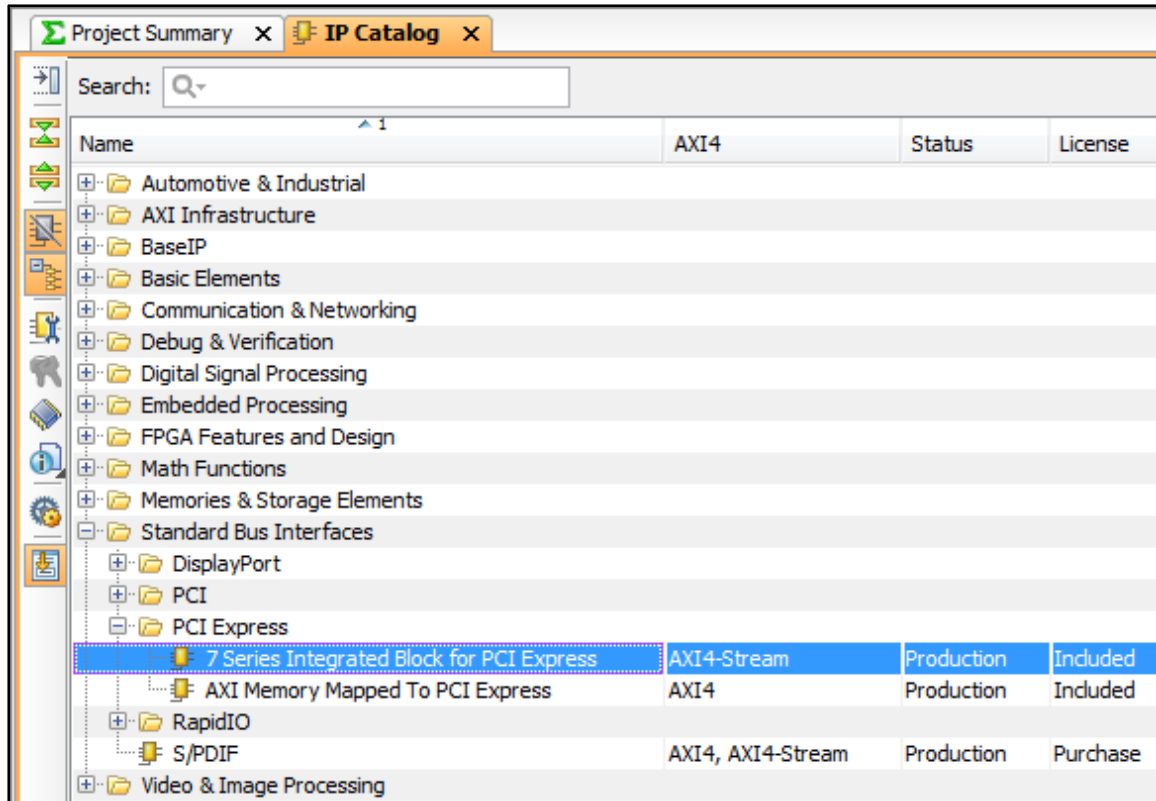
8. In the following figure, click Finish to continue.



9. Click on the **IP Catalog** in the Vivado GUI as shown in the following figure. The **IP Catalog** will open.



10. In the **IP Catalog** view, double-click on the **7 Series Integrated Block for PCI Express** as shown in the following figure.



11. Under the **Basic** tab, please specify various fields as shown in the following figure.

The image shows a software configuration window for a component named "pcie_7x_0". The "Basic" tab is selected, showing various configuration options for a PCI Express Endpoint device. The settings are as follows:

- Component Name:** pcie_7x_0
- Mode:** Basic
- Device Port Type:** PCI Express Endpoint device
- Xilinx Development Board:** None
- PCIe Block Location:** X0Y0
- Silicon Revision:** GES and Production
- Number of Lanes:** Lane Width is set to X4.
- Maximum Link Speed:** 5.0 GT/s is selected (2.5 GT/s is unselected).
- AXI Interface Frequency:** Frequency (MHz) is set to 250.
- AXI Interface Width:** 64 bit is selected (128 bit is unselected).
- Reference Clock Frequency (MHz):** 100 MHz
- Tandem Configuration:** None is selected (Tandem PROM (Refer PG054) is unselected).
- PIPE Mode Simulations:** None is selected (Enable Pipe Simulation and Enable External PIPE Interface are unselected).
- Enable External STARTUP primitive:** Unchecked checkbox.
- Enable External GT Channel DRP:** Unchecked checkbox.
- Additional Transceiver Control and Status Ports:** Unchecked checkbox.

12. Click on the **IDs** tab and configure values as shown in the following figure.

Component Name:

Basic **IDs** BARs Core Capabilities Interrupts

ID Initial Values

Vendor ID	<input type="text" value="10EE"/>	<input type="button" value="x"/>	Range: 0000..FFFF
Device ID	<input type="text" value="7024"/>	<input type="button" value="x"/>	Range: 0000..FFFF
Revision ID	<input type="text" value="00"/>	<input type="button" value="x"/>	Range: 00..FF
Subsystem Vendor ID	<input type="text" value="10EE"/>	<input type="button" value="x"/>	Range: 0000..FFFF
Subsystem ID	<input type="text" value="0007"/>	<input type="button" value="x"/>	Range: 0000..FFFF

Class Code

☐ Use Class Code Lookup Assistant

Base Class Menu	<input type="text" value="Simple communication controllers"/>		
Base Class	<input type="text" value="05"/>	<input type="button" value="x"/>	Range: 00..FF
Sub Class Interface Menu	<input type="text" value="Generic XT compatible serial controller"/>		
Sub-Class	<input type="text" value="80"/>	<input type="button" value="x"/>	Range: 00..FF
Interface	<input type="text" value="00"/>	<input type="button" value="x"/>	Range: 00..FF
Class Code (Hex):	<input type="text" value="058000"/>		

Cardbus CIS Pointer Range: 00000000..FFFFFFFF

13. Under the **BARs** tab, set the BAR0 address space to 2 MB as shown in the following figure.

Component Name:

Basic | IDs | **BARs** | Core Capabilities | Interrupts

Base Address Registers (BARs) serve two purposes. Initially, they serve as a mechanism for the device to request blocks of address space in the address map. After the BIOS or OS determines what addresses to assign to the device, the Base Address Registers are programmed with addresses and sizes to perform address decoding.

Bar0 Enabled	Bar1 Enabled	Bar2 Enabled	Bar3 Enabled	Bar4 Enabled	Bar5 Enabled	Expansion Rom Enabled
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Type: <input type="text" value="Memory"/> <input type="checkbox"/> 64 bit <input type="checkbox"/> Prefetchable	Type: <input type="text" value="N/A"/> <input type="checkbox"/> 64 bit <input type="checkbox"/> Prefetchable	Type: <input type="text" value="N/A"/> <input type="checkbox"/> 64 bit <input type="checkbox"/> Prefetchable	Type: <input type="text" value="N/A"/> <input type="checkbox"/> 64 bit <input type="checkbox"/> Prefetchable	Type: <input type="text" value="N/A"/> <input type="checkbox"/> 64 bit <input type="checkbox"/> Prefetchable	Type: <input type="text" value="N/A"/> <input type="checkbox"/> Prefetchable	Size: <input type="text" value="2"/> <input type="text" value="Kilobytes"/>
Size Unit: <input type="text" value="Megaby..."/> Size Value: <input type="text" value="2"/>	Size Unit: <input type="text" value="Kilobytes"/> Size Value: <input type="text" value="2"/>	Size Unit: <input type="text" value="Kilobytes"/> Size Value: <input type="text" value="2"/>	Size Unit: <input type="text" value="Kilobytes"/> Size Value: <input type="text" value="2"/>	Size Unit: <input type="text" value="Kilobytes"/> Size Value: <input type="text" value="2"/>	Size Unit: <input type="text" value="Kilobytes"/> Size Value: <input type="text" value="2"/>	
Value (Hex): <input type="text" value="FFE00000"/>	Value (Hex): <input type="text" value="00000000"/>	Value (Hex): <input type="text" value="00000000"/>	Value (Hex): <input type="text" value="00000000"/>	Value (Hex): <input type="text" value="00000000"/>	Value (Hex): <input type="text" value="00000000"/>	Value (Hex): <input type="text" value="00000000"/>

14. Click on the **Core Capabilities** tab and leave all default values as shown in the following figure

Component Name:

Basic IDs BARs **Core Capabilities** Interrupts

Capabilities Register

Capability Version (Hex):

Device Port / Type:

☐ Slot Implemented

Capabilities Register (Hex):

Device Capabilities Register

Max Payload Size:

Device Capabilities Register (Hex):

BRAM Configuration Options

☒ Buffering Optimized for Bus Mastering Applications ☐ Finite Completions

Performance Level	Transmit TLPs Buffered	Receiver Buffer Size (bytes)	Posted Header/Data Credits	Non-post
High	15	8192	32/181	12/24
	30	16384	32/437	12/24

☐ Disable Completion Timeout

Completion Timeout:

Supported Ranges

Range A: 50µs to 10ms

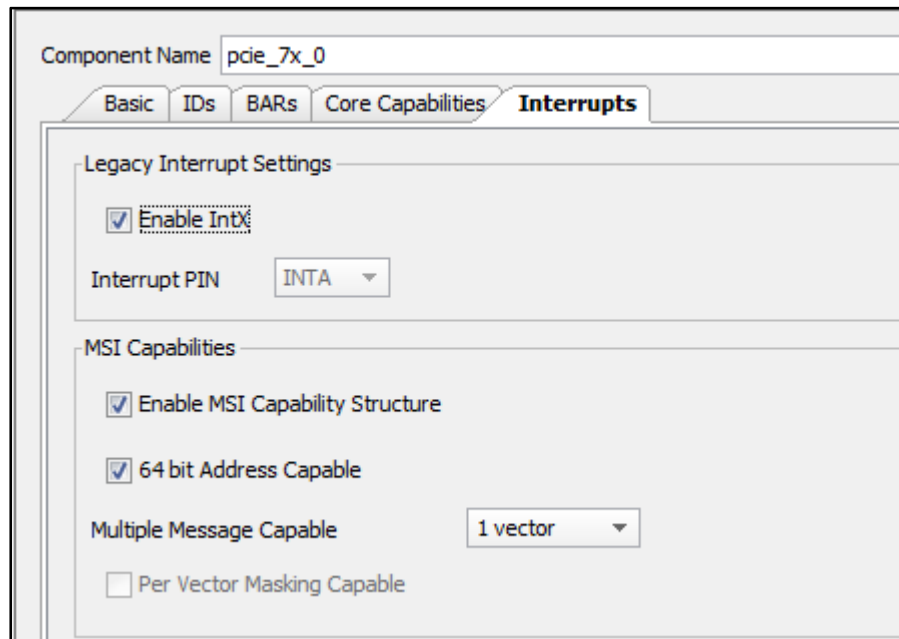
Range B: 10ms to 250ms

Range C: 250ms to 4s

Range D: 4s to 64s

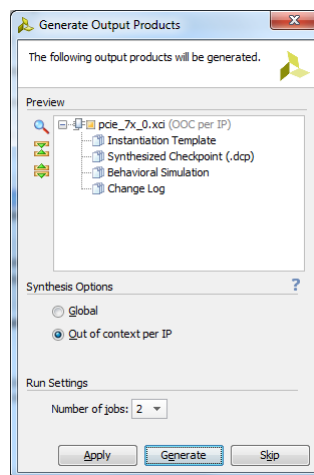
Device Capabilities 2 Register (Hex):

15. Click on the **Interrupts** tab and leave all default values as shown in the following figure

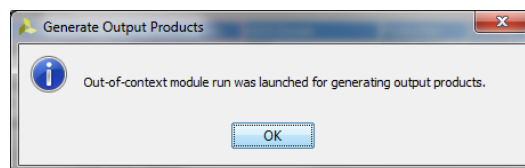


16. Click **OK** to continue. In the confirmation dialog box, also click **OK**.

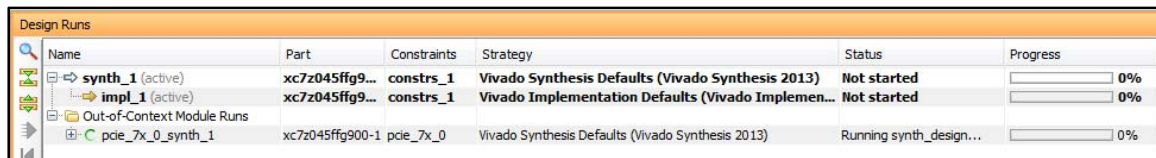
17. In the following dialog box, click **Generate** to continue.



18. When the following dialog appears, click **OK** to continue.

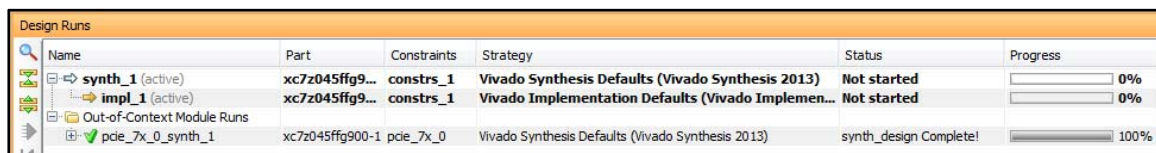


19. The Vivado Core Generator will begin generating the PCIe core. You should see **Running Synth_design** with 0% completion as shown in the following figure.



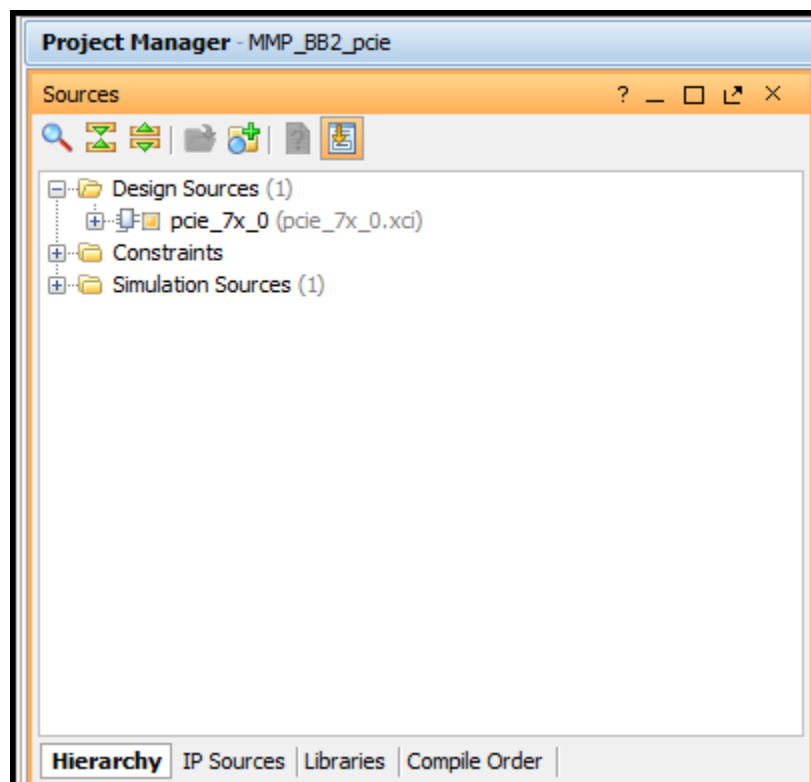
Name	Part	Constraints	Strategy	Status	Progress
synth_1 (active)	xc7z045ffg9...	constrs_1	Vivado Synthesis Defaults (Vivado Synthesis 2013)	Not started	0%
impl_1 (active)	xc7z045ffg9...	constrs_1	Vivado Implementation Defaults (Vivado Implemen...	Not started	0%
Out-of-Context Module Runs					
pcie_7x_0_synth_1	xc7z045ffg900-1	pcie_7x_0	Vivado Synthesis Defaults (Vivado Synthesis 2013)	Running synth_design...	0%

20. Once the PCIe core is synthesized, you should see the **synth_design Complete** with 100% completion as shown in the following figure.

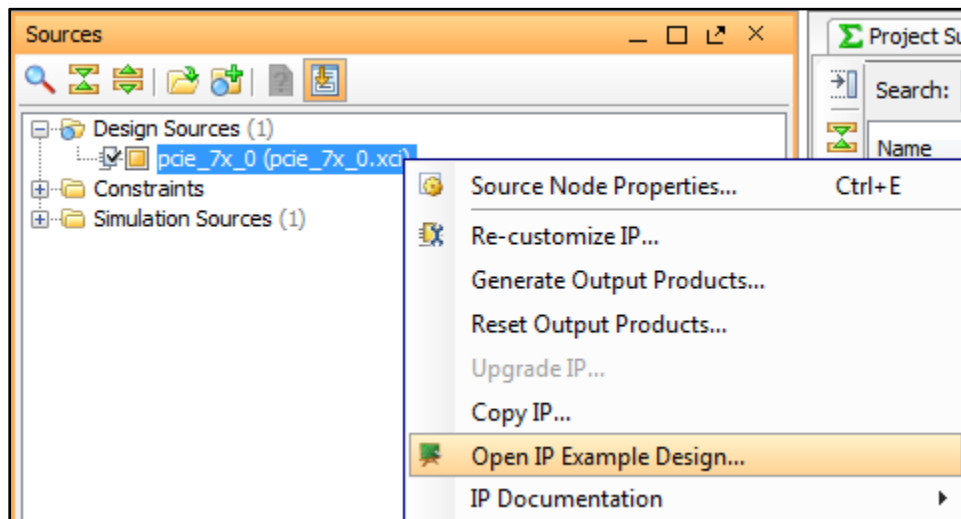


Name	Part	Constraints	Strategy	Status	Progress
synth_1 (active)	xc7z045ffg9...	constrs_1	Vivado Synthesis Defaults (Vivado Synthesis 2013)	Not started	0%
impl_1 (active)	xc7z045ffg9...	constrs_1	Vivado Implementation Defaults (Vivado Implemen...	Not started	0%
Out-of-Context Module Runs					
pcie_7x_0_synth_1	xc7z045ffg900-1	pcie_7x_0	Vivado Synthesis Defaults (Vivado Synthesis 2013)	synth_design Complete!	100%

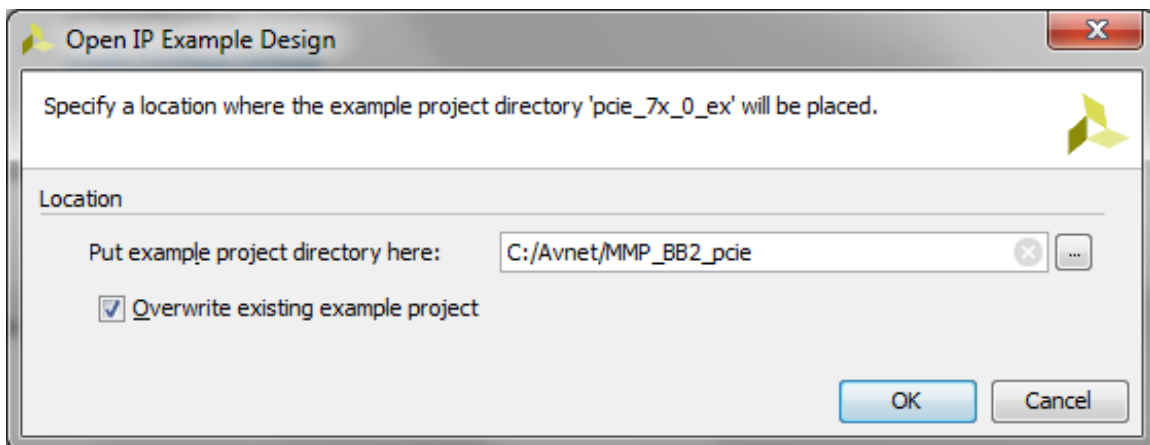
21. At this time, the **Sources** window should look as shown in the following figure.



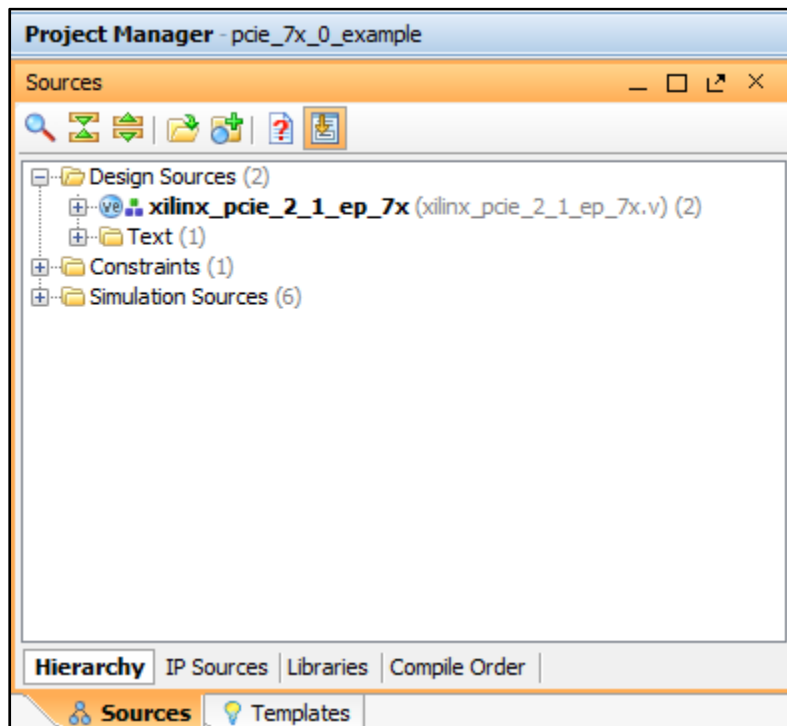
22. Right-click on the **pcie_7x_0.xci** and select **Open IP Example Design** as shown in the following figure. This action will start a new Vivado project to implement a PCIe example design using the PCIe core generated in the previous Vivado project.



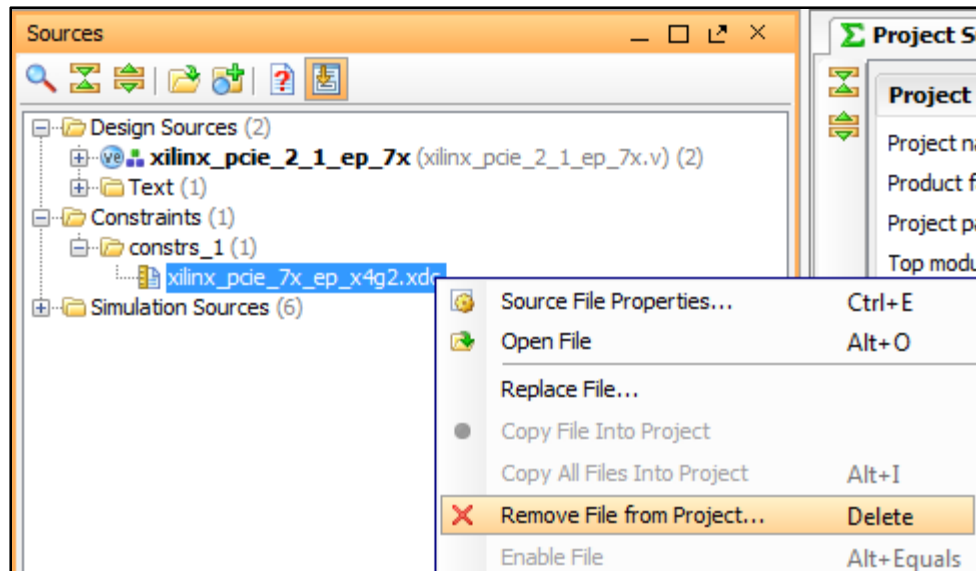
23. When the following dialog box appears, set the location as shown below and click **OK** to continue.



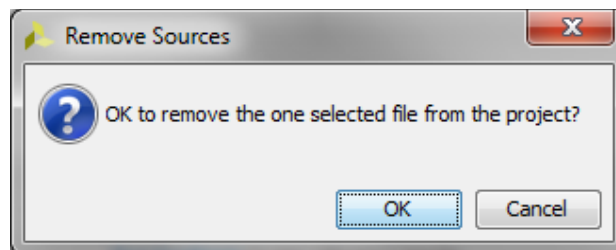
24. The PCIe example project will open in a new session of the Vivado tool and you should see the following in the **Sources** window of the PCIe example design.



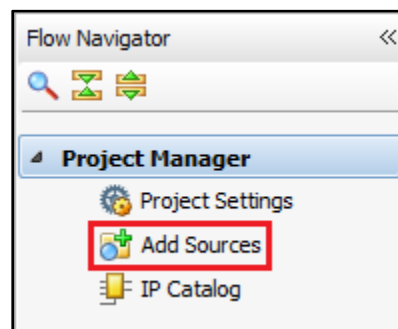
25. Right-click on the **xilinx_pcie_7x_ep_x4g2.xdc** file and select **Remove File from Project** as shown in the following figure. We will be adding an XDC file for the Avnet Zynq MMP module to this project in the next step.



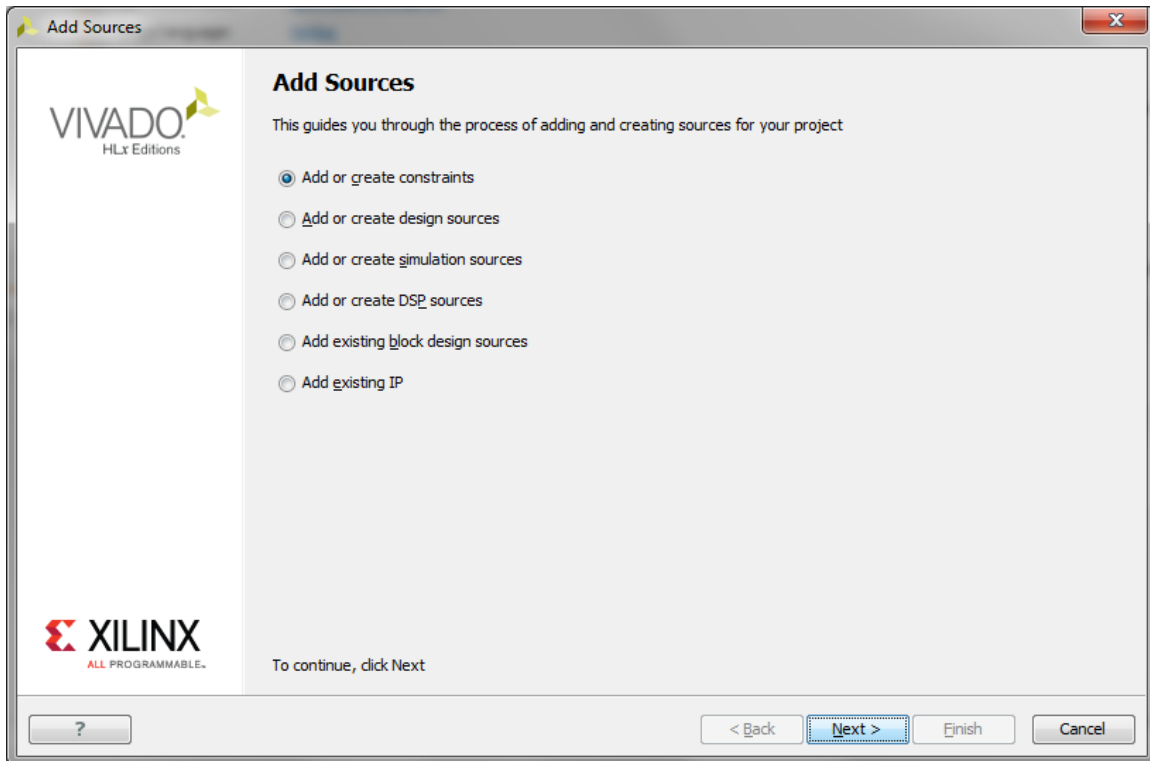
26. When the following dialog box appears, click **OK** to continue.



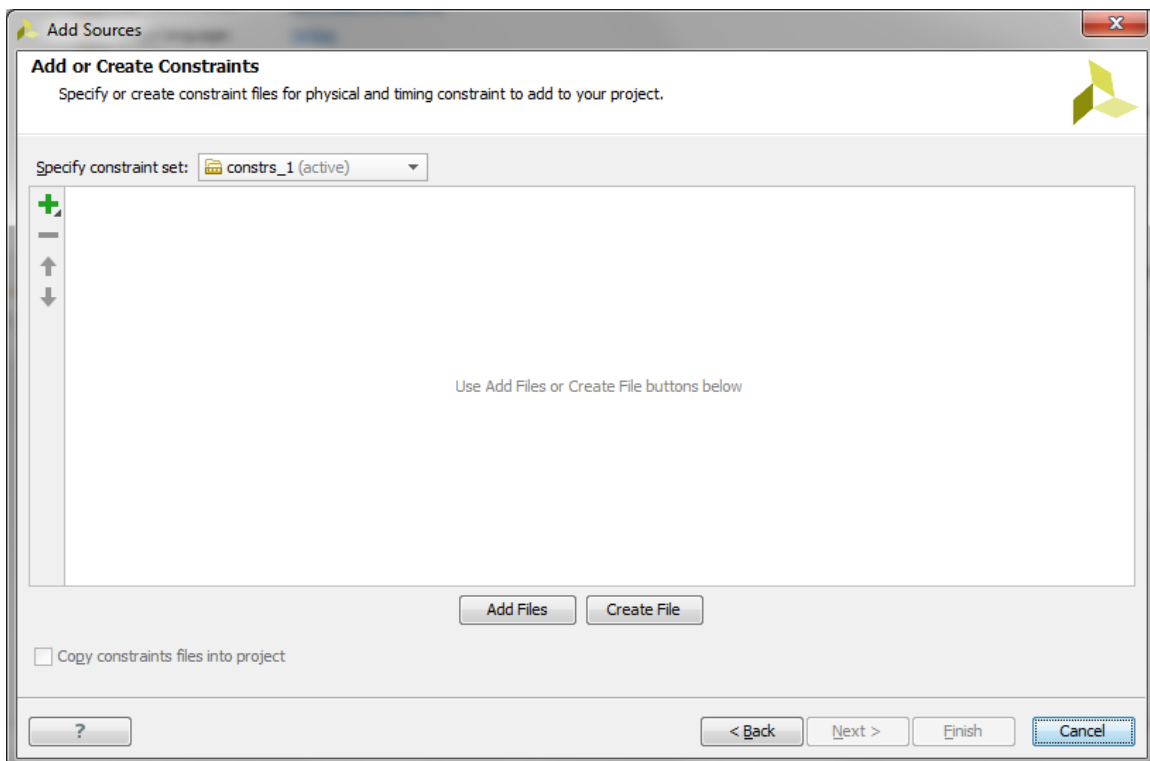
27. Click on Add Sources in the Vivado GUI as shown below.



28. Click on **Add or Create Constraints** and click **Next** to continue.



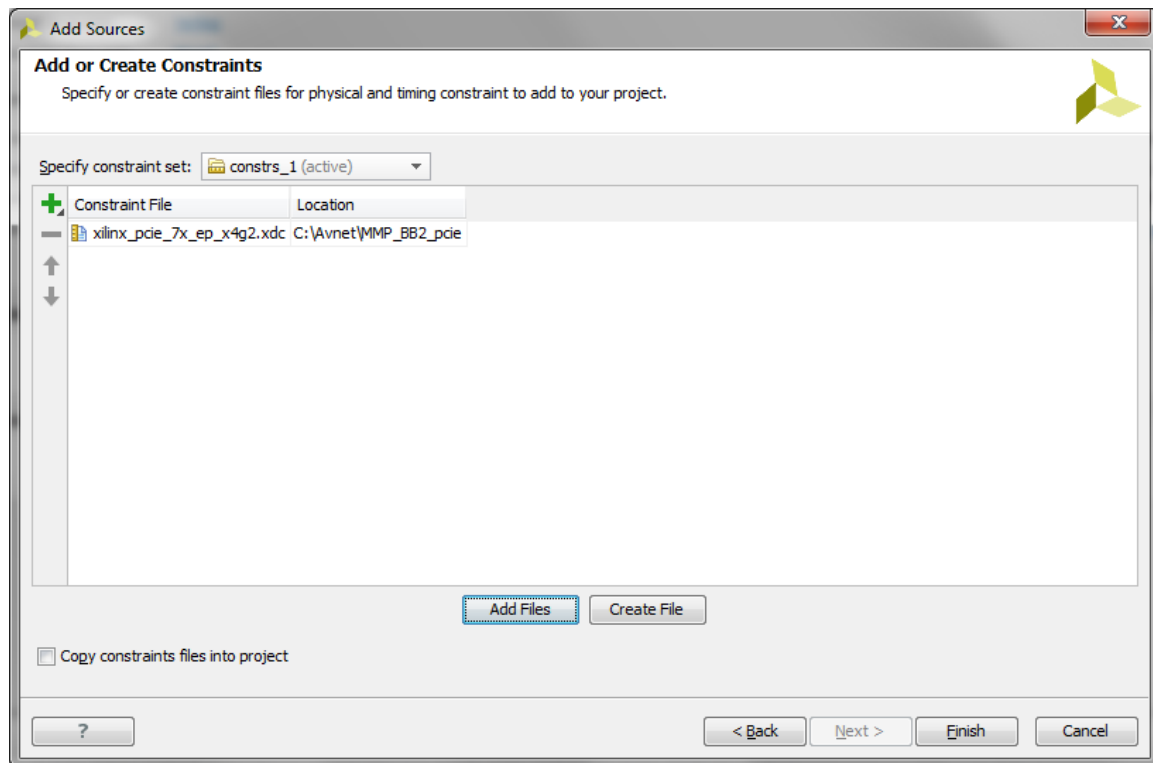
29. Click on **Add Files** to continue.



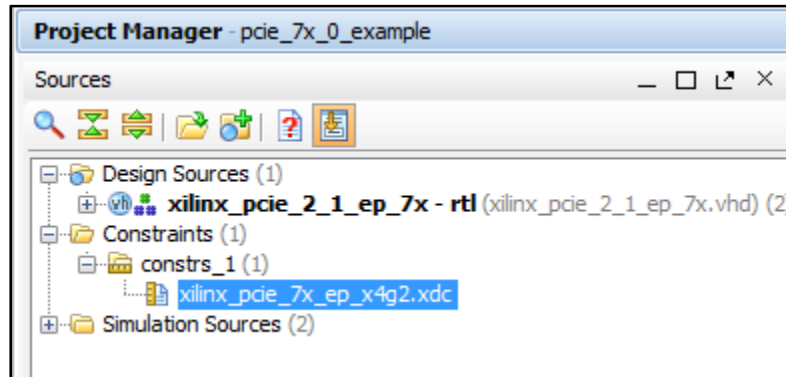
30. Browse to the root directory of the Avnet Zynq MMP PCIe reference design zip file and double-click on the **xilinx_pcie_7x_ep_x4g2.xdc** file.

Name	Date modified	Type
make_mcs_file	11/12/2013 8:20 AM	File folder
ready_to_download	11/12/2013 8:19 AM	File folder
zynq_mmp_pcie_design	10/21/2013 12:30 ...	File folder
xilinx_pcie_7x_ep_x4g2	10/21/2013 12:33 ...	XDC File

31. The **Add or Create Constraints** dialog box should look similar to the one shown below. Make sure the **Copy constraints files into project** box is checked. Click **Finish** to continue

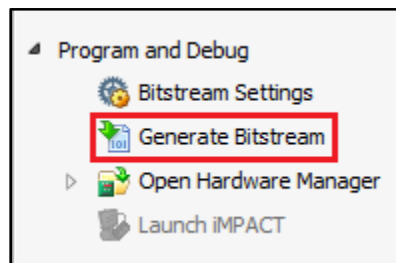


32. The Avnet Zynq MMP module XDC file will be added to the project as shown in the following figure.

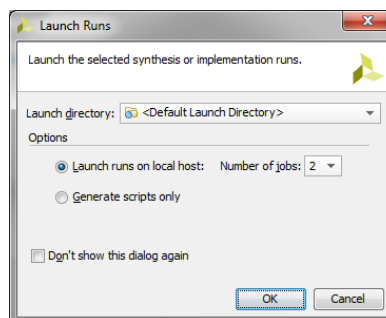


Note: The PCIe example design Vivado project is saved in the **MMP_BB2_pcie\pcie_7x_0_ex** folder in case you need to reopen the project.

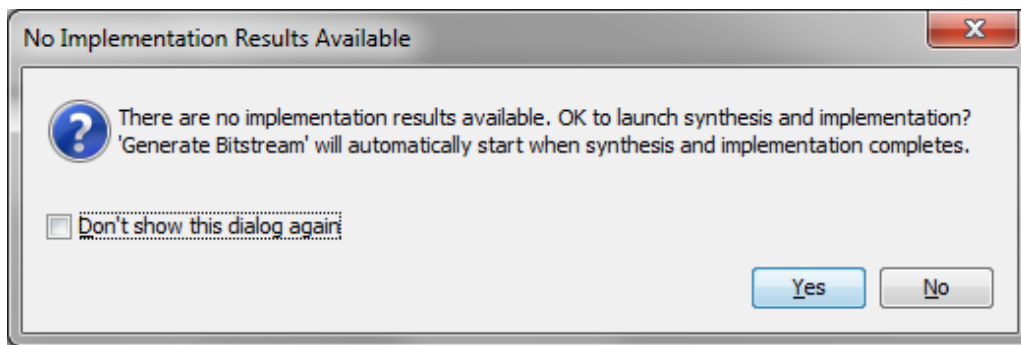
33. Click on **Generate Bitstream** to implement the design and generate a bitstream.



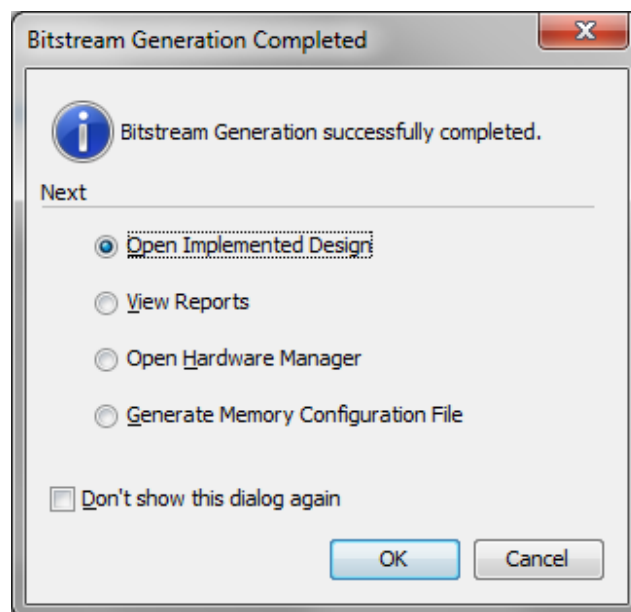
34. When the following dialog box appears, click **OK** to continue.



35. When the following dialog box appears, click **Yes** to continue.



36. At the completion of the bitstream generation, when the following dialog box appears, click **Cancel** to continue.



4 Generating an MCS file for the PCIe Example Design

The following section describes how to generate an MCS file from the bit file generated for the PCIe example design.

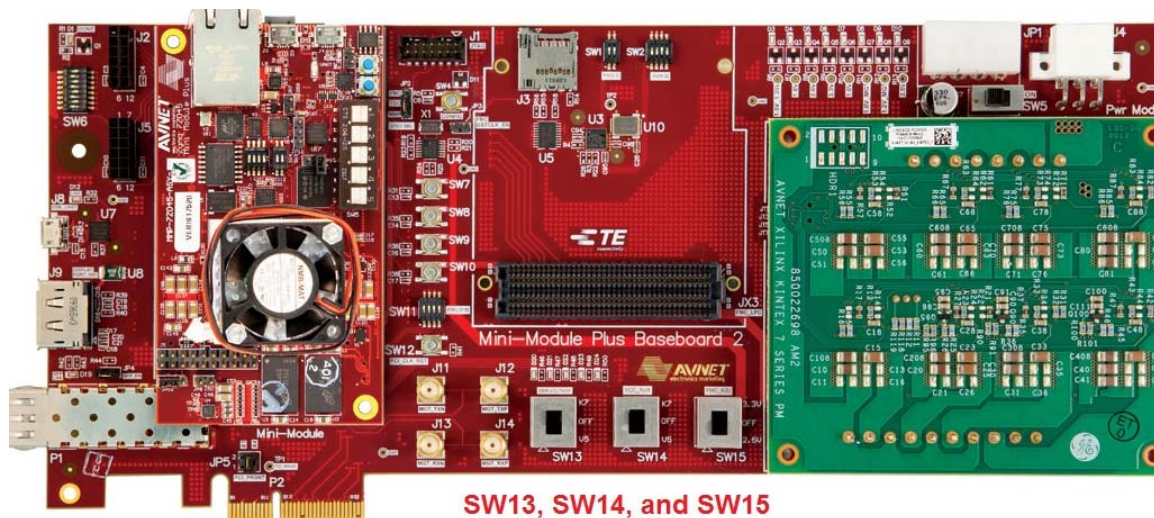
1. Copy the **xilinx_pcie_2_1_ep_7x.bit** file from the **MMP_BB2_pcie\pcie_7x_0_ex\pcie_7x_0_ex.runs\impl_1** folder of your PCIe design to the **make_mcs_file** folder of the reference design.
2. Please go to the **make_mcs_file** folder of the reference design and double-click on **make_mcs_file.bat** file to generate the PCIe example design MCS file.

Note: The bit and MCS files for the Avnet PCIe example design are located in the **ready_to_download** folder of the reference design, should you have any issues with the bit and MCS files generated for your PCIe example design.

5 Setting up the System

Please perform the following steps to setup the Zynq Mini Module Plus kit.

1. Install a jumper on the MMP baseboard JP2 pins 2-3.
2. Install a jumper on the MMP baseboard JP4.
3. Install a jumper on the MMP JP11 pins 2-3.
4. Install a jumper on the MMP JP10 pins 2-3.
5. Slide the SW13 switch to position 3 (marked as K7) on the MMP baseboard (Rev B board only).
6. Slide the SW14 switch to position 3 (marked as K7) on the MMP baseboard (Rev B board only).
7. Slide the SW15 switch to position 1 (all the way down) on the MMP baseboard.
8. connection).
9. Plug the Zynq MMP onto the baseboard via JX1/JX2 connectors.
10. Install the power module onto the MMP baseboard via J6, J7, and J15 connectors.
11. Set the **SW11** to **ON**, **OFF**, **OFF**, and **OFF** positions on the MMP baseboard (F_Sel2, F_Sel1, F_Sel0 = 100) to generate a 100 MHz clock for PCIe.
12. Connect the USB A-mini-B cable to the U1 USB port (on the MMP baseboard, under the Zynq Mini Module Plus) and the USB port of the PC (this would be your JTAG connection).
13. Set the boot mode switch (SW5, positions 1-5) on the MMP to 00000 (Cascaded JTAG mode).
14. Connect 12V power supply to J4.
15. Slide the SW5 power switch to the **ON** position on the MMP baseboard.



Note: Please make sure SW13, SW14, and SW15 switches are set as stated above prior to powering the board or you could damage the Zynq device. This is for Rev B boards only as the SW13 and SW14 have been removed on the later revision of the board.

6 Programming the Zynq MMP QSPI Flash

Please go to the **ready_to_download** folder of the reference design and double-click on the **program_qspi.bat** file to program the QSPI device on the Zynq MMP with the PCIe MCS file.

7 Verifying the Operation of the Zynq-7000 PCIe Endpoint in a System

This section is designed to verify the operation of the Zynq-7000 PCIe Endpoint Block in a system. The Avnet Zynq MMP baseboard will be installed in a PCI Express system and the PCITree application will be used to scan the PCI system, read configuration space and identify the Avnet Zynq MMP baseboard installed in the system.

PCIECV is the PCI SIG configuration space test that can also be used to perform functional test. This reference design will use the PCIECV to verify the functionality of the Zynq-7000 PCIe Endpoint block on the Avnet Zynq MMP baseboard.

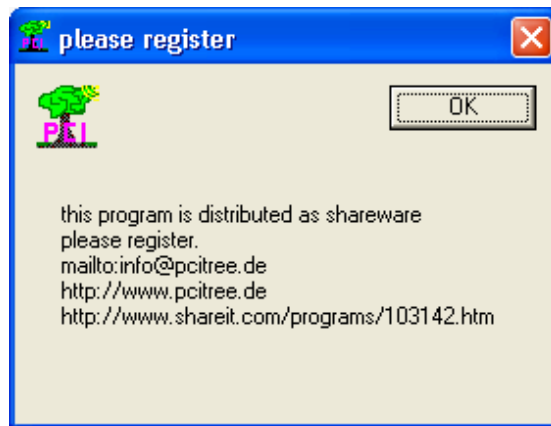
Please refer to the ReadMe file in the **\PCITree_and_PCIECV_Software** folder of the reference design zip file for instructions on how to install the **PCITree** and **PCIECV** tools.

- Slide the SW5 power switch to the **OFF** position on the MMP baseboard.
- Disconnect the 12V power supply from J4 connector on the baseboard.
- Disconnect the USB cable from U1 on the baseboard (your JTAG connection).
- Set the boot mode switch (SW5, positions 1-5) on the MMP to 00010 (QSPI boot mode).
- Slide the SW5 power switch to the **ON** position on the MMP baseboard.
- Make sure your test PC is turned OFF.
- Plug the Avnet MMP baseboard into the test PC PCIe x4 slot.
- Connect a test PC hard drive ATX power cable to the JP1 ATX power connector on the MMP baseboard.
- Turn **ON** the test PC and let it boot to completion.

7.1 Running the PCITree Test

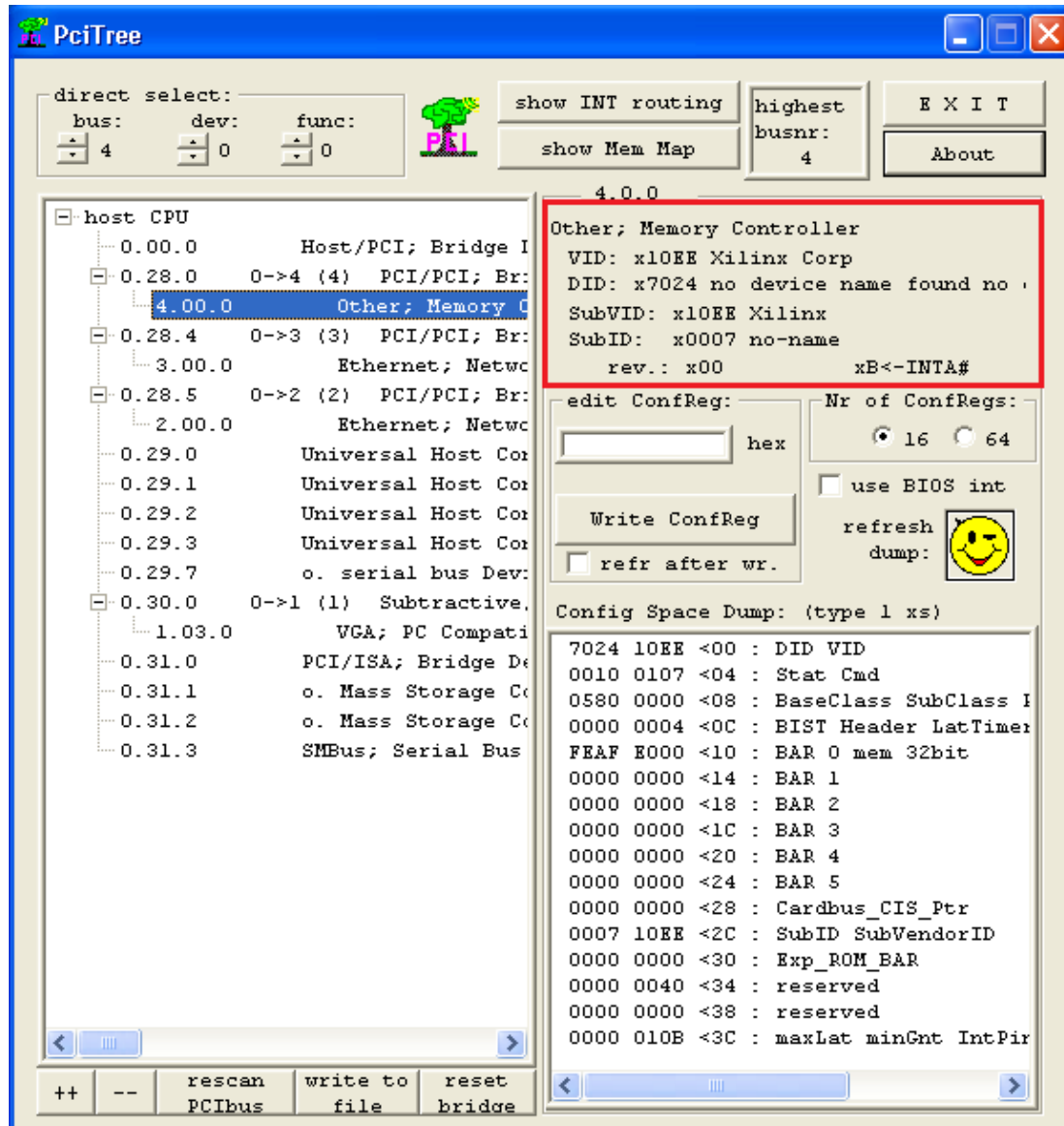
PCITree is a graphical Windows tool to look at all the hardware devices of the PCI bus in a system. The devices are displayed in a tree like view. Information about the devices and its vendors is obtained from a separate database. PCITree enables read and write access to the configuration registers of each device and even each device's memory given by the Base Address Register (BAR). PCITree can be downloaded from the <http://pcitree.de/> site (this software is already included in this reference design zip file).

1. Start the **PCITree** application by double-clicking on the PCITree in the **PCITree_and_PCIECV_Software\pcitree** folder of the reference design. When the following dialog box appears, click **OK** to continue.



Note: In the following sections, the screenshots may vary from what you will see on your own test PC as each PC has different set of cards installed.

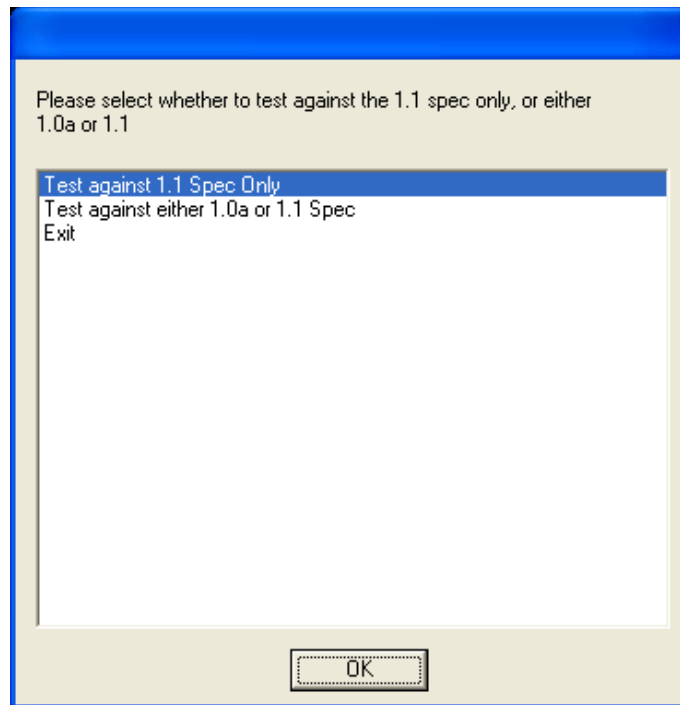
-
2. The main window of PCItree shows the structure of the PCI bus on the left side. Configuration registers of a selected device are displayed on the lower right side. Above the configuration space dump is detailed information on the selected configuration register. There is a field for editing a double-word (DW) configuration register (**edit ConfReg**) and a **Write ConfReg** button. After a write operation, you must click on the **refresh dump** button (smiley face icon) to refresh the display window.
 - a) The Zynq MMP baseboard will show up as **Other; Memory Controller** device in the system as shown in the following figure. Click on the **Co-Other; Memory Controller** device to highlight it, the device information will be displayed on the right-hand side as shown in the following figure. Notice the device **VID**, **DID**, **SubVID**, and **SubID** match the values you used to configure the PCIe Endpoint Block in CORE Generator.



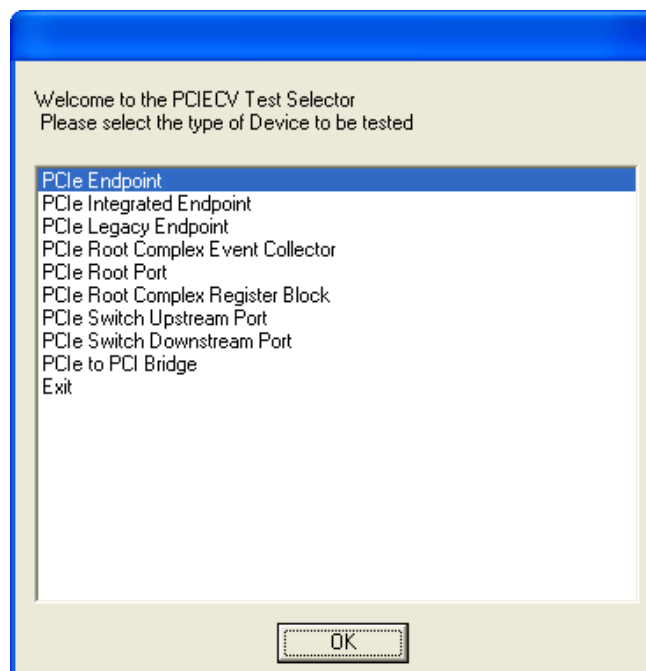
7.2 Running the PCIe Configuration Verification (PCIECV) Test

PCIECV is the PCI SIG configuration space test. SIG members can download it free from the SIG web site www.pcisig.com. This is a comprehensive test that tests the configuration space of any device and has functional tests as well. We are currently running the v1.1 of this test. Note the driver does not need to be loaded to run this test.

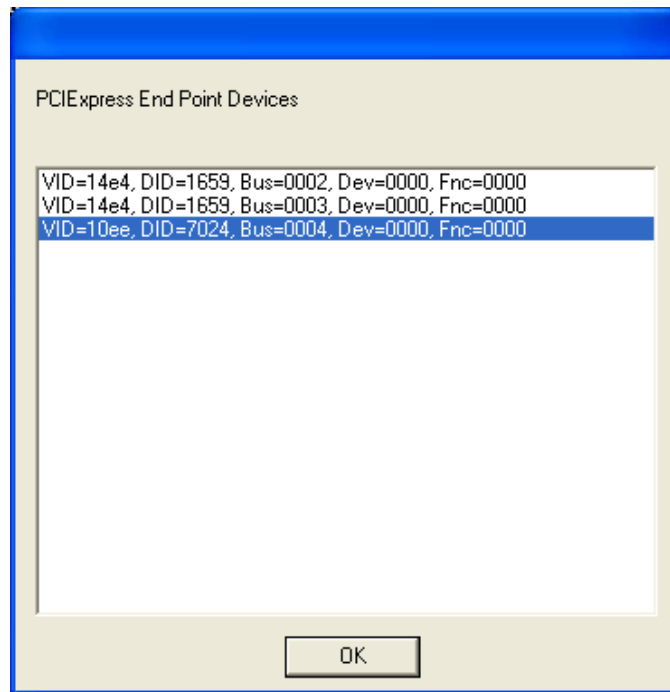
1. Start the **PCIECV** application by double clicking on the **PCIECVApp**. The following dialog box will appear.
 - a) Click on **Test against 1.1 Spec Only**.
 - b) Click **OK** to continue.



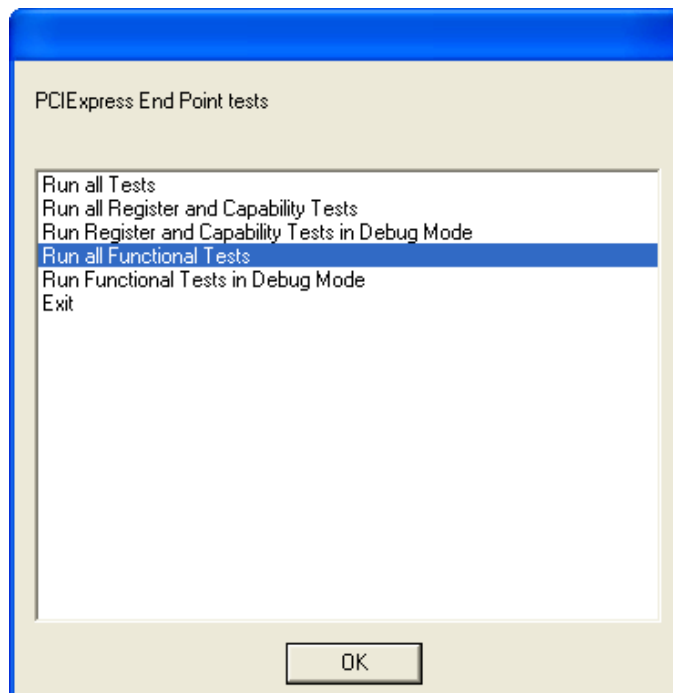
2. Click on **PCIe Endpoint**.
 - a) Click **OK** to continue.



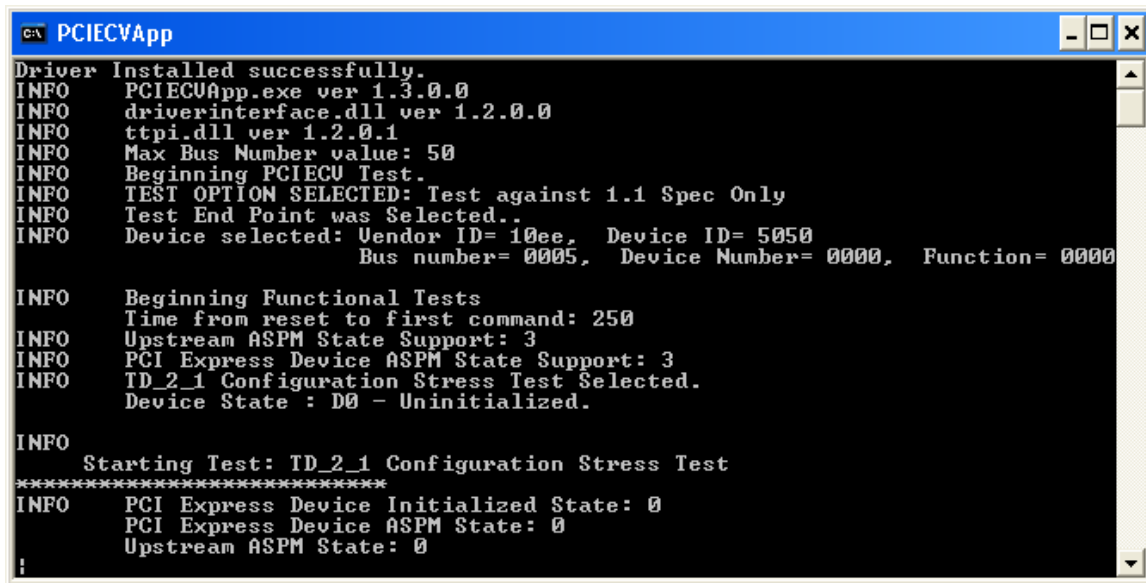
3. Click on **VID=10ee,DID=7024,Bus=0004,Dev=0000,Fnc=0000**.
 - a) Click **OK** to continue.



4. Click on **Run all Functional Tests**.
 - a) Click **OK** to continue.



The test will start and a screen similar to the following will appear in the PCIECVApp window.

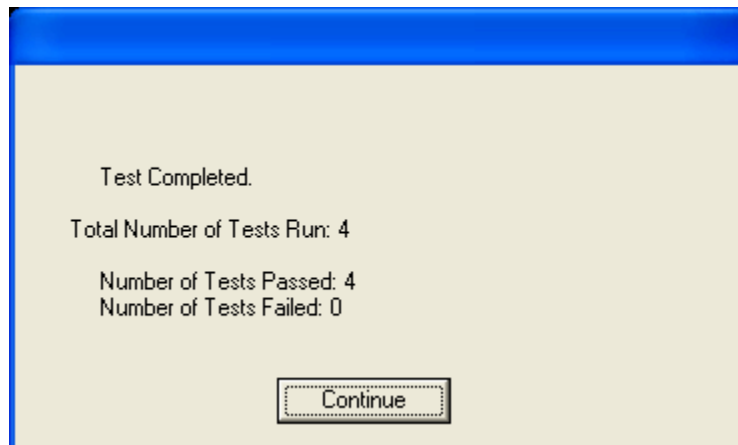


```
Driver Installed successfully.
INFO    PCIECVApp.exe ver 1.3.0.0
INFO    driverinterface.dll ver 1.2.0.0
INFO    ttpi.dll ver 1.2.0.1
INFO    Max Bus Number value: 50
INFO    Beginning PCIECU Test.
INFO    TEST OPTION SELECTED: Test against 1.1 Spec Only
INFO    Test End Point was Selected..
INFO    Device selected: Vendor ID= 10ee, Device ID= 5050
            Bus number= 0005, Device Number= 0000, Function= 0000

INFO    Beginning Functional Tests
INFO    Time from reset to first command: 250
INFO    Upstream ASPM State Support: 3
INFO    PCI Express Device ASPM State Support: 3
INFO    TD_2_1 Configuration Stress Test Selected.
INFO    Device State : D0 - Uninitialized.

INFO    Starting Test: TD_2_1 Configuration Stress Test
*****
INFO    PCI Express Device Initialized State: 0
INFO    PCI Express Device ASPM State: 0
INFO    Upstream ASPM State: 0
:
```

At the end of the test, the test results will be displayed as shown in the following figure.



5. At the completion of the test,
 - a) Click on **Continue**.
 - b) In the next window click on **OK** to continue.
 - c) In the next window click on **Exit** and then click on **OK** to continue.
 - d) In the next window click on **Exit** and then click on **OK** to exit from the PCIECVApp application.