PCIe PIO Reference Design and Tutorial

25 April 2017
Version 2016_4

Overview

The PCIe PIO Reference Design for PicoZed 7015 / 7030 SOMs from Avnet Electronics Marketing provides engineers with guidance in how to use the Xilinx PCIe endpoint IP core in PIO mode (Gen1 x 1). A device driver and GUI app are provided to interact with the PCIe hardware. The software can control onboard LEDs and monitor button status. The software has been targeted to Win7x64 platform. The device driver is created by Windows Driver Kit (WDK) and user app is created by Visual Studio 2013 Express (C#). This document describes how to use Avnet Electronic Marketing's new flow utilizing GIT for an expedited experience.

Figure 1 – PicoZed Logo

Objectives

This tutorial is a guide for how to:

- Retrieve the design files from the public Avnet git repository
- Build the reference design
- Execute the reference design on hardware

Reference Design Overview

Utilizing the Vivado built in cores, this document will help provide a method for producing a PCIe based design as quickly as possible in order to allow an analysis of the transceivers for your hardware helping to expedite your design.

© 2017 Avnet. All rights reserved. All trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Avnet is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Avnet makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Avnet expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Experiment Setup

This tutorial makes use of Xilinx Vivado Design Suite in graphical (GUI) mode in order to create a project, then using secondary software (provided), demonstrate connectivity between a Windows 7x64 PC and a Zynq-7000 SoC based PicoZed SOM.

Software

The software required to build, and execute the reference design is:

- Windws-7 64-bit
- Xilinx Vivado Design Suite 2016.4

Optional Software

The software required to customize the clock configuration is:

- None

Hardware

The hardware required to build, and execute the reference design is:

- PC with minimum amount of additional RAM available for Xilinx tools as specified at www.xilinx.com/design-tools/vivado/memory.htm for either the XC7Z015 or XC7Z030 device
    - 4GB required but 8GB recommended
- Avnet PicoZed 7015 or 7030 SOM
- Avnet PicoZed FMC2 Carrier Card (AES-PZCC-FMC-V2-G), although a compatible FMCv1 card can be used, and design files are provided, the steps to use this card will not be covered
- JTAG Programming Cable (Xilinx Platform Cable, Digilent HS1, HS2, or HS3 cables)
    - If you don’t already have a JTAG Cable, Avnet recommends the Digilent HS3 Cable
    - http://www.em.avnet.com/jtaghs3

Optional Hardware

The optional hardware setup for this reference design is:

- None

Setting up the Board

1. Snap your PicoZed 7015 or 7030 into your FMCv2 carrier card.
    a. **If you are using a PicoZed 7030 SOM, ensure the bank voltage jumpers are set to 1.8V! Leave the voltage selection open, or short pins 1-2**

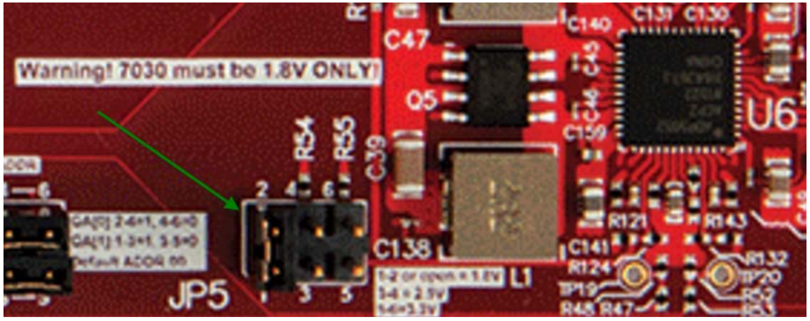
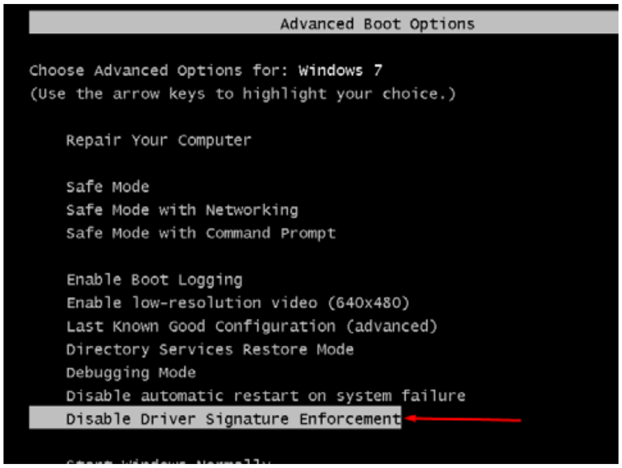


Figure 2 - Jumper Setting for 1.8V Selection

2. Ensure the Power Switch is set to the **ON** position.
3. Configure the boot jumpers for JTAG, or if you choose to create your own SDCARD Bin file, configure for SDCARD as we will still be able to JTAG once per power up.
4. With your PC turned **OFF**, plug your FMCv2 carrier card into an available PCIe slot.
    a. It is recommended to use the included PCIe mechanical bracket to mechanically support the card as well as lock it in place to reduce any issues with the card coming loose.
5. If your design will be a large design, it is recommended to use the included AT – 6 pin power adapter cable. This will ensure proper current can flow into the PicoZed FMCv2 carrier card.
    a. **NOTE THE 6 PIN POWER INPUT IS NOT a PC AT STANDARD! Plugging in a PC power cable WITHOUT the included ADAPTER will DAMAGE your PICOZED and potentially your PC!**
    b. The adapter is not necessary for this design. The design will NOT use enough power to warrant the use of this adapter and the card can be powered through the PCIe slot

Setting up the Avnet provided driver

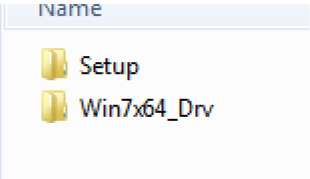
1. At this point, boot your PC.
2. During BOOT, just prior to the Windows 7x64 “Starting Windows” screen, press F8.
    a. Note, you might need to press F8 a few times
3. In the selections, choose the option for disable the driver signature enforcement. This is because device driver provided here is for test purpose and not digitally signed by Microsoft. Windows 7 will not load unsigned driver



```
                    Advanced Boot Options

Choose Advanced Options for: Windows 7
(Use the arrow keys to highlight your choice.)

    Repair Your Computer

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable low-resolution video (640x480)
    Last Known Good Configuration (advanced)
    Directory Services Restore Mode
    Debugging Mode
    Disable automatic restart on system failure
    Disable Driver Signature Enforcement

    Start Windows Normally
```

Figure 3 - Disable Driver Signature

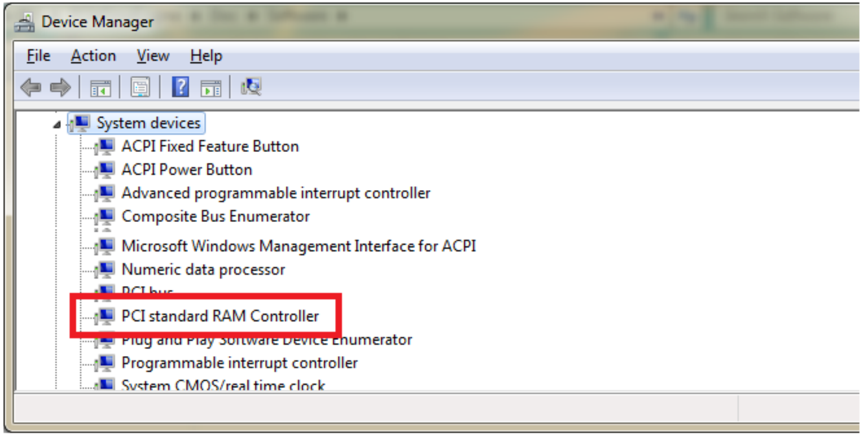
4. Unzip the PicoZed PIO Zip file to your file system.
5. Note that in the \Software folder, you will find two folders. Setup and Win7x64_Drv.



Setup
Win7x64_Drv

Figure 4 - Software Folder

6. After booting, open Device Manger and expand system device
7. Browse through the list and find the device PCI standard RAM Controller



- System devices
  - ACPI Fixed Feature Button
  - ACPI Power Button
  - Advanced programmable interrupt controller
  - Composite Bus Enumerator
  - Microsoft Windows Management Interface for ACPI
  - Numeric data processor
  - PCI bus
  - PCI standard RAM Controller
  - Plug and Play Software Device Enumerator
  - Programmable interrupt controller
  - System CMOS/real time clock

Figure 5 - PCI Standard RAM Controller

8. Double click on PCI standard RAM Controller
9. Select tab Details and Property Hardware Ids

10. Make sure **VEN_10EE&DEV_7011** could be found in the Value field. This indicates that our hardware is recognized by Windows Device Manager
    a. If the values don't match, please search through the Device Manager to locate the device or you may need to check the hardware
11. Press Cancel
12. Right click on PCI standard RAM Controller and select Update Driver Software…
13. Select Locate and install driver software manually
14. The prebuilt driver is included in the design package. It is located at PicoZedPCIePIO_20170324\Software\Win7x64_Drv
15. If the driver is installed successfully, you will see Sample Device->Sample Driver for the PCIe PIO Adapter under Device Manger

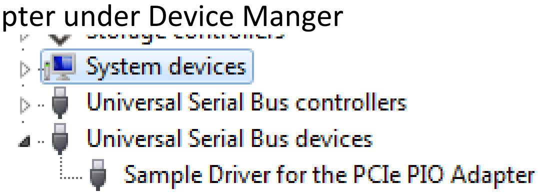

System devices
Universal Serial Bus controllers
Universal Serial Bus devices
Sample Driver for the PCIe PIO Adapter

Figure 6 - Sample Driver Example

16. Every time you boot your PC, you will need to complete steps 1-3, otherwise Windows will enforce signature verification. Missing this step should NOT uninstall the driver, only prevent it from being loaded. If you miss this step, simply reboot your PC and complete steps 1-3.
17. At this point, your hardware is setup and ready to execute your design!

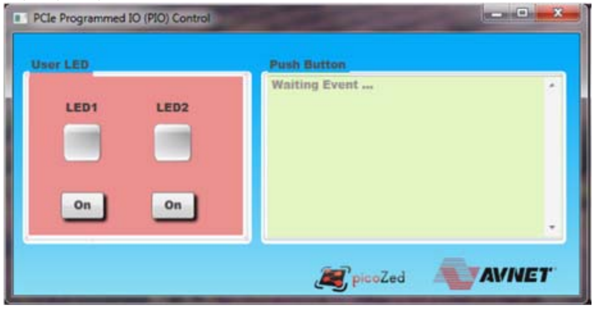
Setting up the Avnet provided software

1. Make sure .Net Framework is installed
2. It is suggested to disable User Account Control (UAC) during the install
3. Double click on setup.exe which is located at \Software\Setup
4. Follow the setup wizard to complete the installation

Using the Avnet provided GUI software

1. Make sure driver signature enforcement is disabled at system boot up
    a. Refer to steps 1-3 under the previous section "Setting up the Avnet provided driver"
2. Select All programs -> Avnet -> PCIe Programmed IO (PIO) Control to launch the app

3. The below GUI will pop up



Waiting Event ...

Figure 7 - PCIe PIO Control GUI

4. In the GUI, press On/Off button under User LED field to turn on/off LED. Please check the LEDS located near JPS1 on the card edge

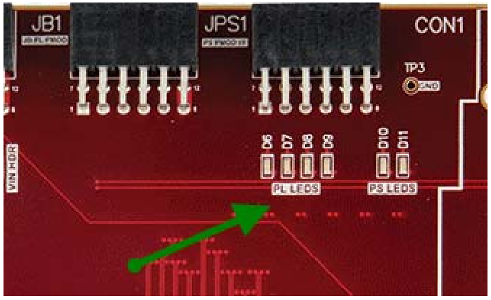


Figure 8 - LED Bank

5. The GUI is also ready to receive events from the board.
6. The two buttons called out in the XDC will be the North and South buttons, as also indicated by SW1 and SW5.
7. On the hardware, press either switch, you will see messages in the event window, similar to:
    a. Event PB1 down
    b. Event PB1 up
    c. Event PB2 down
    d. Event PB2 up

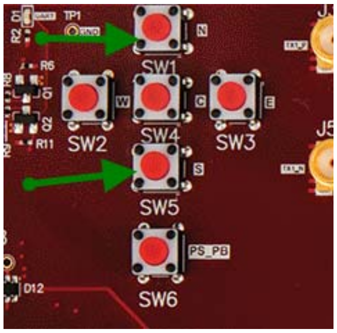


Figure 9 - Joystick Layout for Buttons

Reusable Components

The tutorial does not use any components that are readily available for reuse, although all the IP that is used is provided as part of the Vivado Suite install. If you would like access to the driver or GUI software, please contact your local Silica or Avnet FAE.

Experiment 1: Retrieve the design files

In this section, the design files for the reference design will be retrieved from the Avnet git repository.

1. Navigate to the following web site : https://github.com/Avnet/hdl

2. Click the **branch:master** button

3. Specify the following search criteria : ibert

4. Click the **Tags** tab

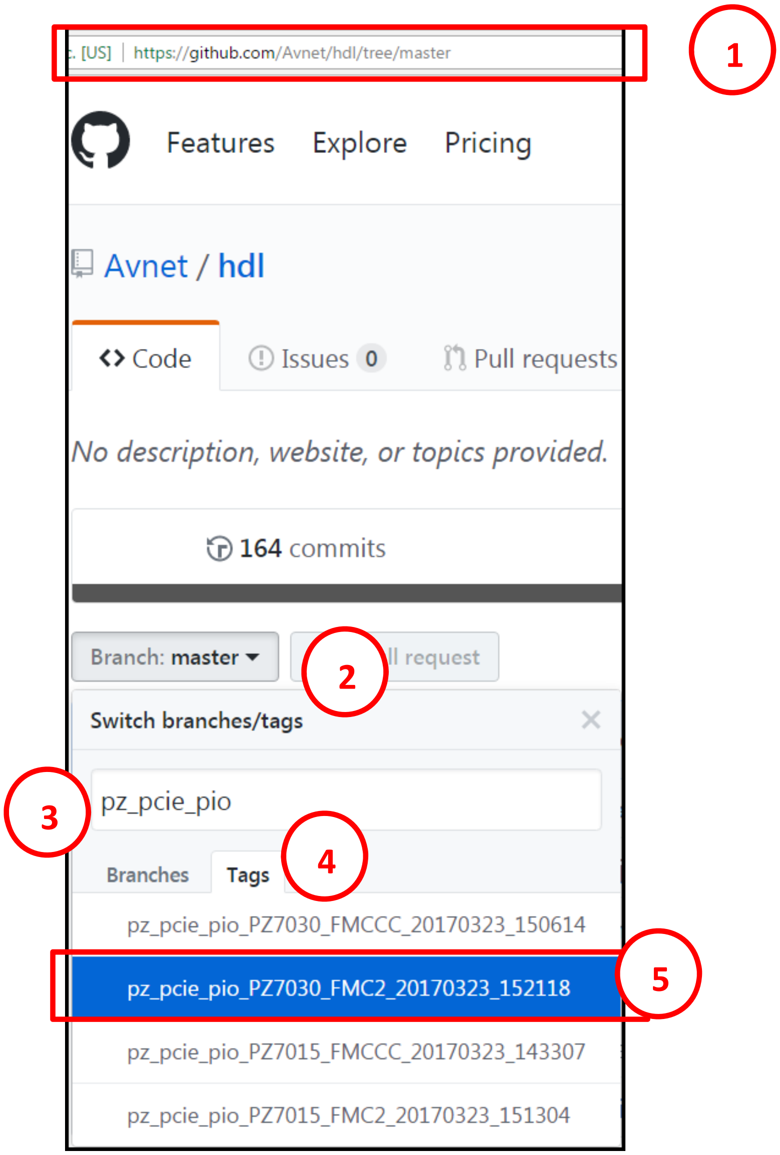

1
[US] | https://github.com/Avnet/hdl/tree/master
Features
Explore
Pricing
Avnet / hdl
Code
Issues 0
Pull requests
No description, website, or topics provided.
164 commits
Branch: master
ll request
2
Switch branches/tags
3
pz_pcie_pio
Branches
Tags
4
pz_pcie_pio_PZ7030_FMCCC_20170323_150614
pz_pcie_pio_PZ7030_FMC2_20170323_152118
5
pz_pcie_pio_PZ7015_FMCCC_20170323_143307
pz_pcie_pio_PZ7015_FMC2_20170323_151304

Figure 10 – Avnet GitHub repository – Retrieving specific version with tag

5. Select the **pz_pcie_pio_PZ7030_FMC2_20170323_152118** tag

   This will retrieve a known working version of the design files for the IBERT reference design for both PicoZed SOMs using either FMC Carrier Card target.

6. Click the Clone or download button, then the Download ZIP button

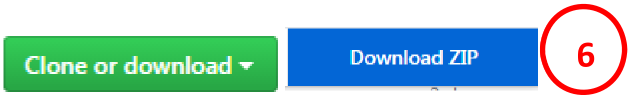


Figure 11 – Avnet GitHub repository – Download ZIP

7. Create an "Avnet" directory in your root C:\ drive

8. Save **hdl-pz_pcie_pio_PZ7030_FMC2_20170323_152118.zip** file to the **C:\Avnet** directory, and extract the contents of the zip file in this directory

9. Rename the "hdl- pz_pcie_pio_PZ7030_FMC2_20170323_152118" directory to "hdl"

   You should see the following directory structure

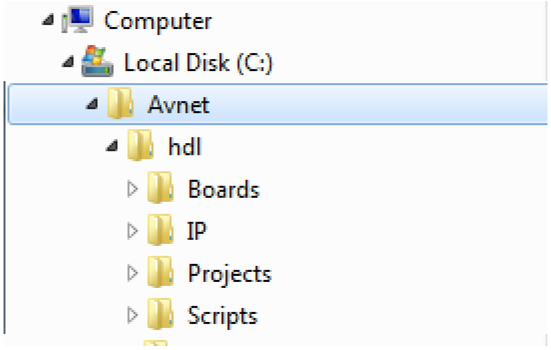


Figure 12 – Extracted C:\Avnet\hdl directory structure

NOTE: The exact directory name is not critical, but it must remain short on Windows machines, due to the directory length limitation of Windows

The **C:\Avnet\hdl** repository contains the following sub-directories:

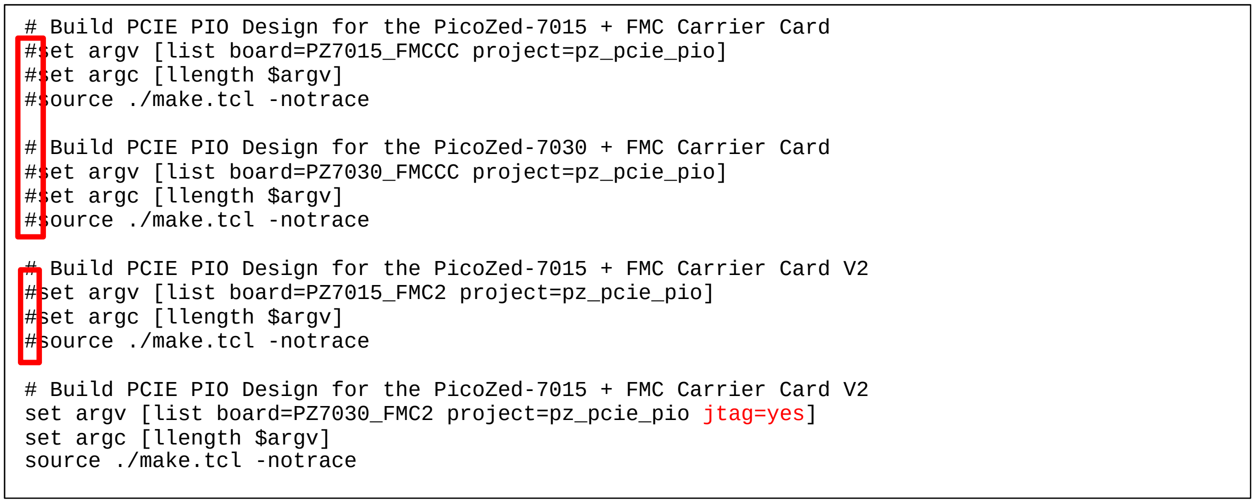
Directory	Content Description
C:\Avnet\hdl\Boards	contains board related files
C:\Avnet\hdl\IP	contains the IP cores used by ref designs
C:\Avnet\hdl\Projects	contains project related files
C:\Avnet\hdl\Scripts	contains scripts used to automatically build the designs

For the IBERT reference design, the following content is of interest:

Directory	Content Description
C:\Avnet\hdl\Projects\pz_pcie_pio	Folder containing files for IBERT reference design
C:\Avnet\hdl\Scripts\make_pz_pcie_pio.tcl	TCL script to launch the build of the IBERT reference design

By default, the script will build the design for all four combinations of the PicoZed 7015 and 7030 SOM, combined with either a FMC Carrier Card or the FMC V2 Carrier Card.

10. Edit the **make_pz_pcie_pio.tcl** script to only build for your PicoZed SOM, as well as enable JTAG to allow the script to take the build all the way to configuring your board set.
    As an example, if you have a PicoZed 7030 SOM, with FMC2 Carrier Card comment out the build for both PicoZed 7015 SOMs as well as the , PicoZed 7030 with FMC Carrier Card, as shown below. Optionally add “jtag=yes” if you wish for the script to attempt to JTAG your board for you.



```
# Build PCIE PIO Design for the PicoZed-7015 + FMC Carrier Card
#set argv [list board=PZ7015_FMCCC project=pz_pcie_pio]
#set argc [llength $argv]
#source ./make.tcl -notrace

# Build PCIE PIO Design for the PicoZed-7030 + FMC Carrier Card
#set argv [list board=PZ7030_FMCCC project=pz_pcie_pio]
#set argc [llength $argv]
#source ./make.tcl -notrace

# Build PCIE PIO Design for the PicoZed-7015 + FMC Carrier Card V2
#set argv [list board=PZ7015_FMC2 project=pz_pcie_pio]
#set argc [llength $argv]
#source ./make.tcl -notrace

# Build PCIE PIO Design for the PicoZed-7015 + FMC Carrier Card V2
set argv [list board=PZ7030_FMC2 project=pz_pcie_pio jtag=yes]
set argc [llength $argv]
source ./make.tcl -notrace
```

Figure 13 – Editing the make script to build only for PicoZed 7015 SOM, with FMC2 Carrier Card

Experiment 2: Build the reference design

In this section, the Vivado project will be created and built with TCL scripts, implementing the PCIe PIO Design.

1. From the Start menu, open the "Vivado 2016.4" GUI

2. In the TCL Console, Change to the **C:\Avnet\hdl\Scripts** directory

```
cd c:/Avnet/hdl/Scripts/
```

Figure 14 – Vivado 2015.4 TCL Console – Changing to C:/Avnet/hdl/Scripts directory

3. Launch the build with the "**source ./make_pz_pcie_pio.tcl**" TCL command

```
source ./make_pz_pcie_pio.tcl
# set argv [list board=PZ7015_FMCCC project=pz_pcie_pio]
# set argc [llength $argv]
# source ./make.tcl -notrace

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-                                                     -*
*-        Welcome to the Avnet Project Builder         -*
*-                                                     -*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

+------------------+------------------------------------+
| Setting          |     Configuration                  |
+------------------+------------------------------------+
| Board            |     PZ7015_FMCCC                   |
+------------------+------------------------------------+
| Project          |     pz_pcie_pio                    |
+------------------+------------------------------------+



Version of Vivado acceptable, continuing...
```

Figure 15 – Vivado 2016.4 TCL Console – Launching the build

The build will perform the following steps:

- Create and build the hardware design with Vivado 2016.4,
  including the Xilinx Coregen PCIe IP wrapped with the PIO HDL wrapper

```
Your Build Took
seconds [377]

or a total of:

days [0]
hrs  [0]
min  [6]
sec  [17]

to complete

*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*
*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*
*_                                                     _*
*_            Finished Running Script                  _*
*_                                                     _*
*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*
*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*
```

Figure 16 – Vivado 2016.4 TCL Console – Scripted Build Complete

NOTE: If you see the Critical Warning:

CRITICAL WARNING: [IP_Flow 19-4965] IP pcie_7x_0 was packaged with board value 'em.avnet.com:picozed_7015_fmc2:part0:1.1'. Current project's board value is unset. Please update the project settings to match the packaged IP.

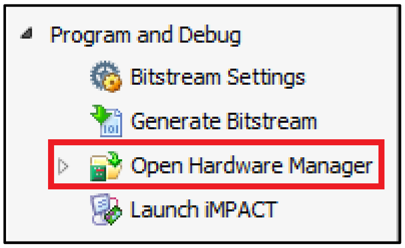
This is safe to ignore in this case, as the script will preset the entire design's configuration. This Warning stems from the IP, which was originally generated with a board preset definition and used an AXI to PCIe capability. The portability of the Xilinx IP allows for us to easily integrate this IP into the PIO design without effort – however the tool is performing correct in warning the user that it is possible we have a mismatch. In this case it is intentional.

Experiment 3: Execute the reference design on hardware

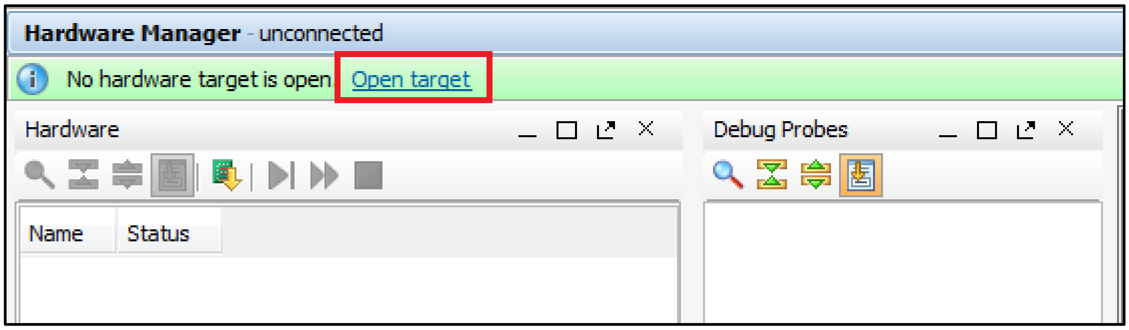
If you did not opt to have the script JTAG your SOM, it is now time to perform the JTAG and begin testing using the PCIe test GUI. Please refer to the Avnet “Tech Tip - Transceiver Tools 102: We have an IBERT bit stream, now what?” at 3:06 seconds in. Here we walk you through performing a JTAG of an IBERT design to the PL of a PicoZed 7015 SOM. This same process is used for each configuration of the SOM as well as this design. If you would rather have a printed document, please see the next section for a step-by-step procedure for programming your PCIe PIO design.

Running the PCIe PIO Design on the PicoZed Carrier Card V2

1. After opening your design, Click on the **Open Hardware Manager** in the Vivado GUI as shown in the following figure.

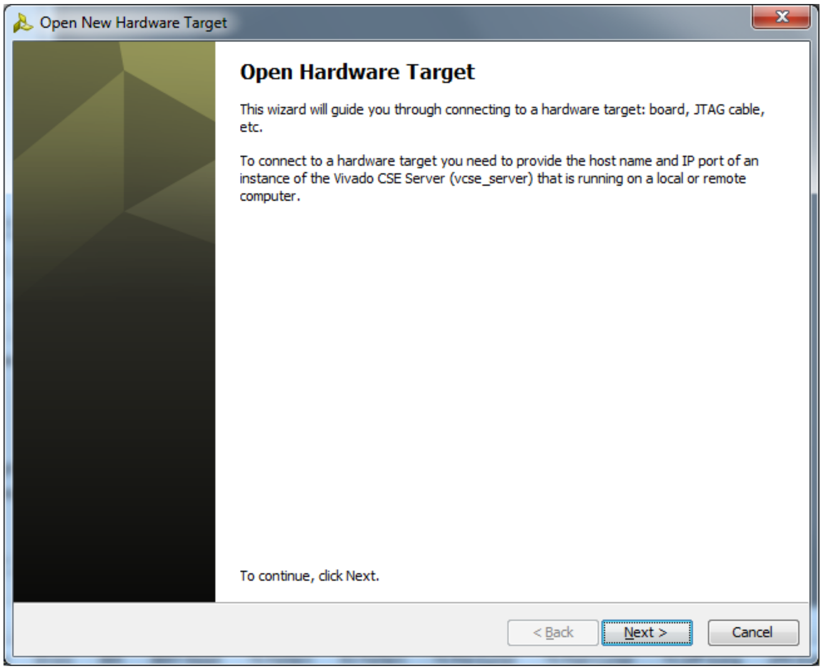


2. Click on the **Open target** and then select **Open New Target** in the Vivado GUI as shown in the following figure.


Hardware Manager - unconnected

No hardware target is open Open target

3. Click **Next** to continue.



Open New Hardware Target

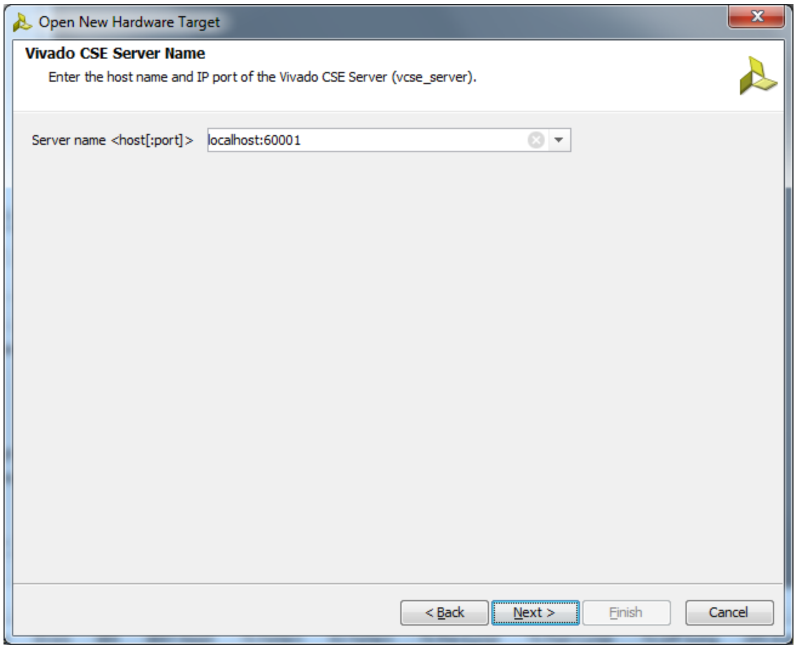
Open Hardware Target

This wizard will guide you through connecting to a hardware target: board, JTAG cable, etc.

To connect to a hardware target you need to provide the host name and IP port of an instance of the Vivado CSE Server (vcse_server) that is running on a local or remote computer.

To continue, click Next.

4. Click **Next** to continue.



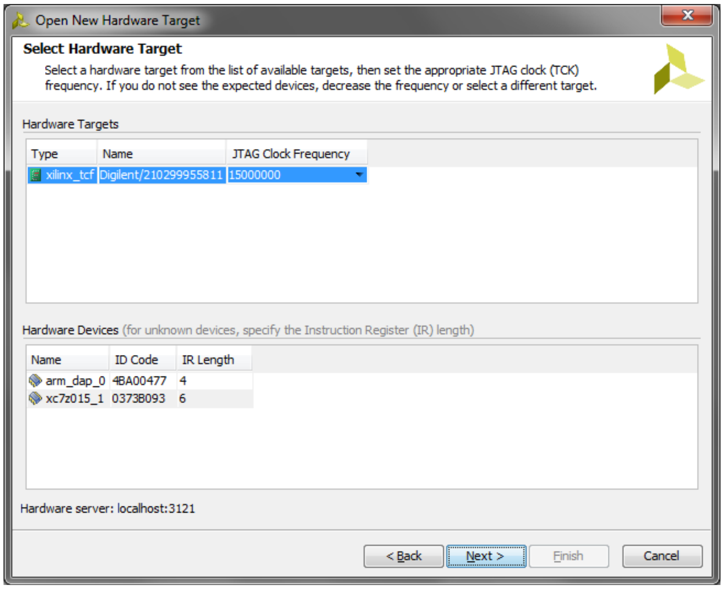
Open New Hardware Target

Vivado CSE Server Name

Enter the host name and IP port of the Vivado CSE Server (vcse_server).

Server name <host[:port]> localhost:60001

5. Click **Next** to continue. Note, for the xc7030 based SOM, you will see 7030 in the below window, under Hardware Devices.



Select Hardware Target

Select a hardware target from the list of available targets, then set the appropriate JTAG clock (TCK) frequency. If you do not see the expected devices, decrease the frequency or select a different target.

Hardware Targets

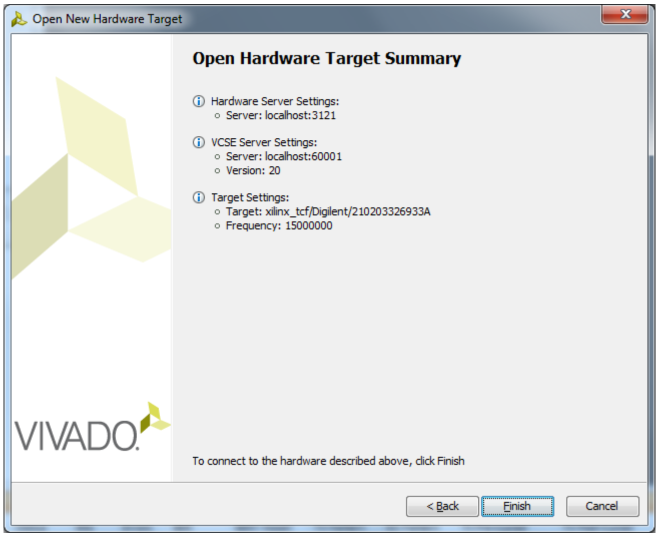
Type	Name	JTAG Clock Frequency
xilinx_tcf	Digilent/210299955811	15000000

Hardware Devices (for unknown devices, specify the Instruction Register (IR) length)

Name	ID Code	IR Length
arm_dap_0	4BA00477	4
xc7z015_1	0373B093	6

Hardware server: localhost:3121

6. Click **Finish** to continue.



Open Hardware Target Summary

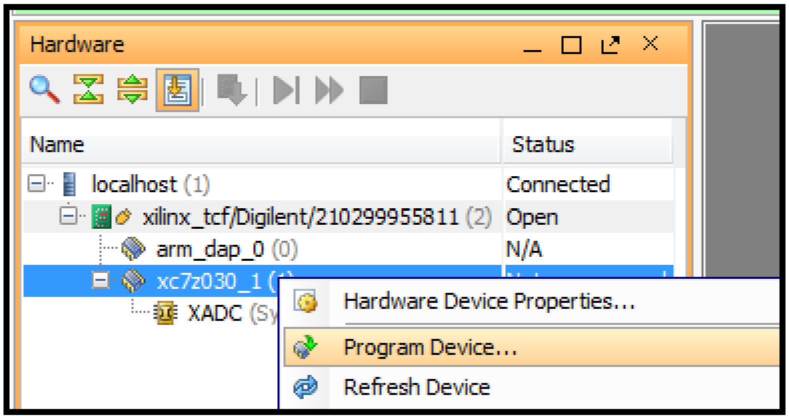
Hardware Server Settings:
- Server: localhost:3121

VCSE Server Settings:
- Server: localhost:60001
- Version: 20

Target Settings:
- Target: xilinx_tcf/Digilent/210203326933A
- Frequency: 15000000

To connect to the hardware described above, click Finish

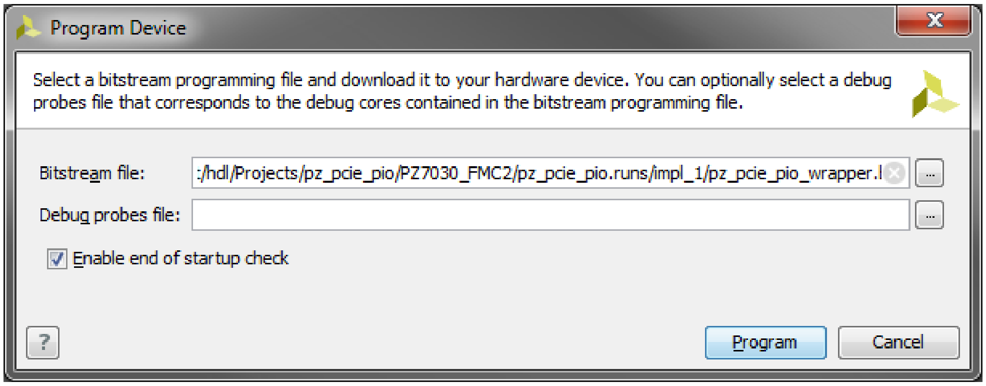
7. Right-click on the **XC7Z015_1 or XC7Z030_1** device and select Program Device as shown in the following figure.



Hardware

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210299955811 (2)	Open
arm_dap_0 (0)	N/A
xc7z030_1 (	
XADC (Sy	

8. When the following dialog box appears, click **Program** to continue.



Program Device

Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

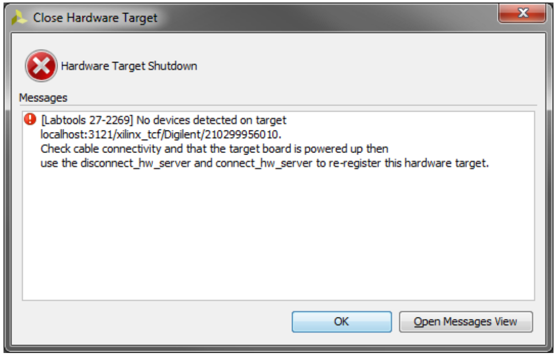
Bitstream file: :/hdl/Projects/pz_pcie_pio/PZ7030_FMC2/pz_pcie_pio.runs/impl_1/pz_pcie_pio_wrapper.l

Debug probes file:

☑ Enable end of startup check

Note: Previous versions of Vivado always had a Debug probes file listed. This is not true for Vivado 2016.4

9. If you do not have your SOM set for JTAG, you might see the following message. When the following dialog box appears, click **OK** to continue.



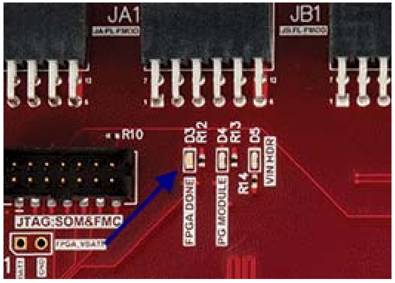
Close Hardware Target

Hardware Target Shutdown

Messages

```
[Labtools 27-2269] No devices detected on target
localhost:3121/xilinx_tcf/Digilent/210299956010.
Check cable connectivity and that the target board is powered up then
use the disconnect_hw_server and connect_hw_server to re-register this hardware target.
```

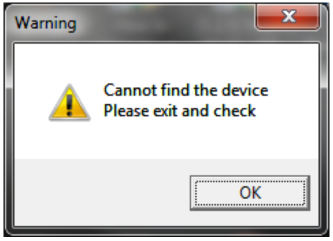
10. This above message will appear as the design will disable JTAG. You can validate your programming was successful by checking the BLUE FPGA DONE LED is lit



11. Reboot your PC following the Steps 1-3 of the **Setting up the Avnet provided driver** section. This reboot is needed to allow the SOM to register with the motherboard's PCIe memory space.

12. You can load the PCIE GUI and test the various buttons and features as described in **Using the Avnet provided GUI software** section.

13. If you see the following figure, you missed the F8 key and will need to reboot your PC, pressing the F8 key as shown in steps 1-3 under the **Setting up the Avnet provided driver** section. You should not need to repeat the JTAG procedure as the SOM will retain the programming through a reboot.



Cannot find the device
Please exit and check

14. From here follow the steps in **Using the Avnet provided GUI software** section. You should be able to see various LEDS and button presses force communications back and forth between the SOM and the PC.

Revision History

Date	Version	Revision
24 Mar 2017	2016.4	Initial release for PicoZed targets using GIT flow