Open PXIe System Documentation

# Building the System Controller FPGA Firmware

## Requirements

* Vivado 2015.4
* System\_controller\_7015\_2015.4

## Building

1. Open Vivado 2015.4
2. In the Tcl console at the bottom, change your directory to the location of the system\_controller\_7015\_2015.4 project.
3. Run command *source mk\_prj.tcl* This starts the block building process
4. Once the block building is complete, click File > Export > Export Hardware
5. run *generate bitstream.* The result may output a statement saying that timing is not met. Ignore this for the time being.
6. After this process, the bitstream can be found in *proj/proj.runs/impl\_1/zynq\_rc\_x4g1\_wrapper.bit*

## Notes

* This step can be skipped if you have the pre-generated bitstream

# Building the System Controller (SC) Linux OS

## Requirements

* ­Ubuntu (recommend 14.04 for compatibility reasons)
* Petalinux 2015.4
* Xpcie\_test (testing user space application)
* Xpcie (driver)
* system\_controller\_7015\_2015.4 Vivado Project, with generated bitstream

## Building

1. Start command line terminal
2. Source Petalinux settings using command: *source /opt/petalinux-v2015.4-final/settings.sh* This needs to be done every new instance of the terminal
3. Generate a new Petalinux project using command: *petalinux-create –type project –template zynq –name system\_controller\_os*
4. This creates a new folder in your current directory called *system\_controller\_os.* Change your current directory to the newly created folder
5. Copy the generated bitstream *zynq\_rc\_x4g1\_wrapper.bit* from Vivado project into *./system\_controller\_os/*
6. To incorporate the FPGA firmware, run command: *petalinux-config –get\_hw\_description=. -p .*
7. This will launch a configuration menu, press escape twice to exit the menu
8. To add *xpcie\_test­* application, run command: *petalinux-create –type apps –name xpcie\_test --enable*
9. Copy and paste *xpcie\_test.c* and *Makefile* into *./system\_controller\_os/components/apps/xpcie\_test* Make sure you replace the old files
10. To add the *xpcie* driver, run command: *petalinux-create –type modules –name xpcie –enable*
11. Copy and paste *Makefile, xpcie.h, xpcie\_cdma.c, xpcie\_core.c xpcie\_pci.c* Make sure you replace the old files
12. Make sure your current directory is *./system\_controller\_os*
13. Run command *petalinux-config -c rootfs*
14. Go to Filesystem Packages > console/utils > pciutils
15. Enable libpci3, pciutils, pciutils-uds
16. Hit escape a couple of times until it asks you to save your configuration. Save your configuration.
17. Run command *petalinux-config -c kernel*
18. Go to Bus support
19. Enable PCI Support, MSI, PCI debugging, PCI resource re-allocation detection, PCI express port bus spport
20. Again, hit escape until it asks you to save your configuration, save your configuration
21. Run *petalinux-build*
22. Make sure your current directory is *./system\_controller\_os*
23. Run *petalinux-package –boot –format BIN –project . –fsbl images/linux/zynq\_fsbl.elf –fpga images/linux/zynq\_rc\_x4g1\_wrapper.bit –u-boot –force*
24. This generates the BOOT.bin and image.ub required for the SD card. Copy these from *images/linux* and store it into an SD card.

# Building Peripheral Module (PM) FPGA Firmware

## Requirements

* Te0712\_pcie\_barebones\_2016.4\_7\_11\_17
* Vivado 2016.4

## Building

1. Open Vivado 2016.4
2. In the Tcl console at the bottom, change your directory to the location of the te0712\_pcie\_barebones\_2016.4\_7\_11\_17.
3. Run command *source mk\_prj.tcl* This starts the block building process
4. run *generate bitstream*.
5. After this process, the bitstream can be found in *proj/proj.runs/impl\_1/system\_wrapper.bit*

# Memory Reads/Writes Test between SC and PM

## Requirements

* SD card with BOOT.bin and image.ub
* PXIe-1062Q chassis
* System Controller in system controller slot of chassis
* Peripheral Module in peripheral module slot of chassis
* JTAG platform USB cable
* Vivado 2016.4
* Putty

## Process

1. Connect JTAG cable to JTAG connector on Peripheral Module
2. Start Vivado 2016.4 and open te0712\_pcie\_barebones\_2016.4\_7\_11\_17
3. Connect USB cable to UART port on System Controller (directly below SD card)
4. Click open hardware manager
5. Turn on PXIe chassis
6. Start Putty, select serial with serial line configured to the right COM port (check device manager) and a speed of 115200. Click open
7. Putty should be displaying the boot messages
8. While booting, go back to Vivado and click program device
9. Once the Peripheral Module is programmed, restart the System Controller with push button P84 (closest to the blue LED)
10. After the boot process, login with username *root* and password *root*
11. Run lspci and check that the Peripheral Module is there
12. Run command *insmod /lib/modules/4.0.0-xilinx/extra/xpcie.ko*
13. Run command *xpcie\_test*
14. Enter device name */dev/xpcie\_9* (Might want to check */dev* folder for an xpcie device if this doesn’t work)
15. Type ‘d’ to run demo memory test, consists of 100 Byte words being written and read 64 times
16. Type ‘m’ to run manual memory address. At the moment, it is capable of having 100000 integers