WWW.MWFTR.COM

Atom-FPGA Communication using Ubuntu 13.10 on Development Machine

by
Emmanuel Ademuwagun

EECE 494: Computer Bus and SoC Interfacing
Department of Electrical and Computer Engineering
Howard University

Instructor: Dr. Charles Kim

Presentation

- Quartus 12.1 Installation
- USB-Blaster Configuration
- Task Execution
- Configuring FPGA using Qsys
- Loading SRAM Object File (.sof) on Board
- Setting Up TFTP
- PCIe Driver
- User Application
- Assignment Execution

Quartus 12.1 Installation (Ubuntu)

Select a previous version of Quartus II: 12.1

Download Manager | Individual Files | DVD .iso File

Download Option 2: Individual Files

Download large, standalone installation files to manually install complete software and device support. Note: If you download and install only the QuartusÂ® II software, subsequently you must also download and install the device support . You do not have to install all the device support files. You can install only the device support files that you need.

Quartus II Web Edition	Platform	File Name	Size
Quartus II Web Edition	Windows	12.1_quartus_free_windows.exe MD5: cbafa092a9cd264904694ca8b9403e11	3.8 GB
Quartus II Web Edition	Linux	12.1_quartus_free_linux.tar.gz MD5: a206a7585d23d433c26ada2a916b24d1	4.6 GB

What the MD5 sum value is and what it is for

Other Individual Download Files:

- Quartus II Subscription Edition
- ModelSim-Altera
- ModelSim-Altera Starter
- DSP Builder
- Programming Software

System Requirements

Operating System Requirements

Disk space: A full installation of the Altera Complete Design Suite v12.1 requires approximately 10 GB of available disk space on the drive or partition where you are installing the software, and approximately 30 MB of available space on the drive that contains your TEMP directory (Windows only).

Quartus 12.1 Installation (Ubuntu)

Extract folder from compressed download (.tar.gz) in chosen directory

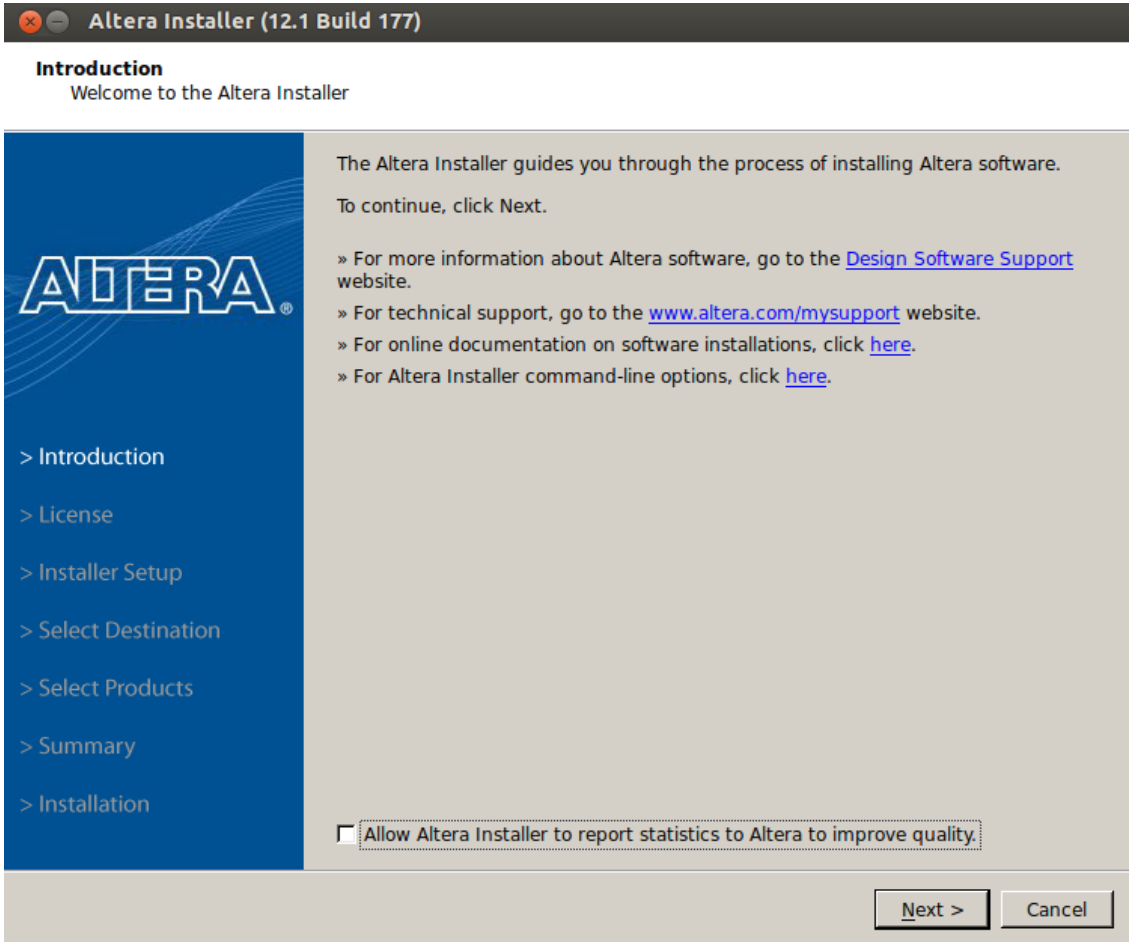
```
emmanuel@emmanuel-desktop:~$
emmanuel@emmanuel-desktop:~$ mkdir fpga/
emmanuel@emmanuel-desktop:~$ cd fpga/
emmanuel@emmanuel-desktop:~/fpga$ tar -zxvf ~/Downloads/12.1_quartus_free_linux.
tar.gz
12.1_177_quartus_free_linux/
12.1_177_quartus_free_linux/dsp_lic.txt
12.1_177_quartus_free_linux/setup
12.1_177_quartus_free_linux/nios2_lic.txt
12.1_177_quartus_free_linux/license.txt
12.1_177_quartus_free_linux/devices/
12.1_177_quartus_free_linux/devices/web/
12.1_177_quartus_free_linux/devices/web/max7000ae.qda
12.1_177_quartus_free_linux/devices/web/arriaii.qdalog
12.1_177_quartus_free_linux/devices/web/maxii.qdalog
12.1_177_quartus_free_linux/devices/web/max7000ae.qdalog
```

Navigate through directory and launch installer

```
emmanuel@emmanuel-desktop:~/fpga$ cd 12.1_177_quartus_free_linux/altera_installer/bin/
emmanuel@emmanuel-desktop:~/fpga/12.1_177_quartus_free_linux/altera_installer/bin$ sudo ./altera_installer_gui --gui
```

Quartus 12.1 Installation (Ubuntu)

Complete Installation using a Graphical User Interface

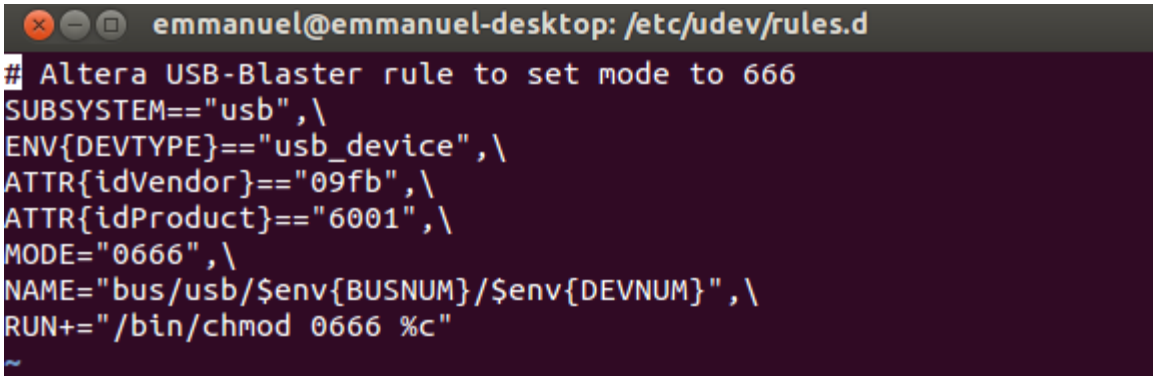


USB-Blaster Configuration (Ubuntu)

Create a new rules file for USB Blaster

```
emmanuel@emmanuel-desktop:~$ cd /etc/udev/rules.d/
emmanuel@emmanuel-desktop:/etc/udev/rules.d$ vi 51-usbblaster.rules
```

Copy the content below in the file



Feed the new rule into the OS by reloading all rules

```
emmanuel@emmanuel-desktop:/etc/udev/rules.d$ sudo udevadm control --reload-rules
```

USB-Blaster Configuration (Ubuntu)

Test Configuration

```
emmanuel@emmanuel-desktop:~/Downloads$ lsusb
Bus 002 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:8008 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 004: ID 045e:07b2 Microsoft Corp.
Bus 003 Device 003: ID 09fb:6001 Altera Blaster
Bus 003 Device 002: ID 045e:075d Microsoft Corp. LifeCam Cinema
Bus 003 Device 005: ID 045e:0745 Microsoft Corp. Nano Transceiver v1.0 for Bluetooth
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
emmanuel@emmanuel-desktop:~/Downloads$ jtagconfig
1) USB-Blaster [3-5]
  028040DD   EP4CGX150
```

ALTERA.

Task - Description

- Use 16 switches on the DE2i-150 to control 4 hexadecimal switches on the board.
- 4 switches → 1 Hexadecimal Value (0 – F)
- Download Source Files from:
  - http://rijndael.ece.vt.edu/de2i150/designs/hellopci.tgz

Task - Sources

Extract Folder and View Contents

```
emmanuel@emmanuel-desktop:~/Downloads$ tar -xvf hellopci.tgz
hellopci/
hellopci/quick/
hellopci/quick/pcihello.sof
hellopci/quick/app
hellopci/quick/altera_driver.ko
hellopci/source/
hellopci/source/driver/
hellopci/source/driver/altera_driver.c
hellopci/source/driver/Makefile
hellopci/source/app/
hellopci/source/app/Makefile
hellopci/source/app/app.c
hellopci/source/fpga/
hellopci/source/fpga/pcihello.qar
emmanuel@emmanuel-desktop:~/Downloads$ ls hellopci/
quick  source
emmanuel@emmanuel-desktop:~/Downloads$ ls hellopci/quick/
altera_driver.ko  app  pcihello.sof
emmanuel@emmanuel-desktop:~/Downloads$ ls hellopci/source/
app  driver  fpga
emmanuel@emmanuel-desktop:~/Downloads$ ls hellopci/source/app
app.c  Makefile
emmanuel@emmanuel-desktop:~/Downloads$ ls hellopci/source/driver
altera_driver.c  Makefile
emmanuel@emmanuel-desktop:~/Downloads$ ls hellopci/source/fpga
pcihello.qar
emmanuel@emmanuel-desktop:~/Downloads$
```

Configuring FPGA using Qsys

For convenience, add bin directory to your PATH variable in order to run your commands from anywhere in the terminal

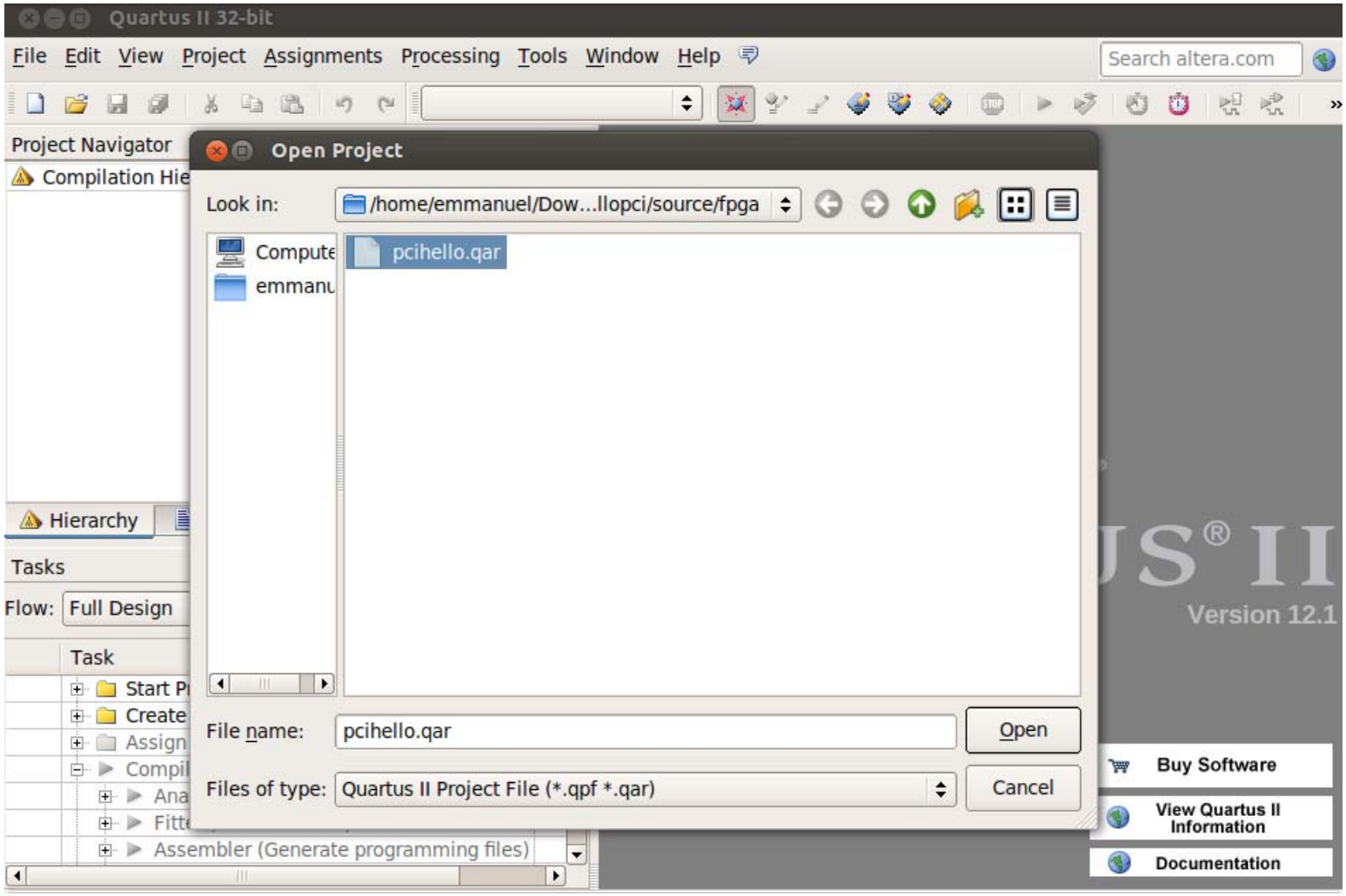
```
emmanuel@emmanuel-desktop:~/Downloads$ export PATH=$PATH:/home/emmanuel/altera/12.1/quartus/bin
```

Launch Quartus

```
emmanuel@emmanuel-desktop:~$
emmanuel@emmanuel-desktop:~$ quartus
```

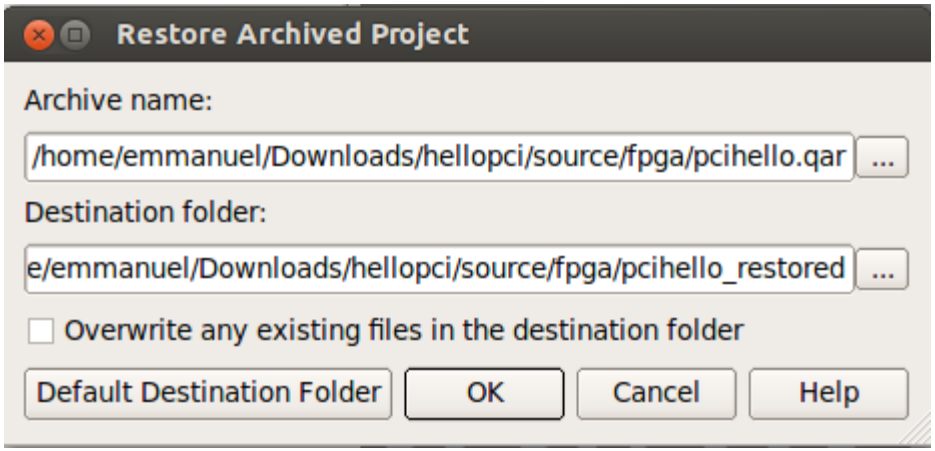
Configuring FPGA using Qsys

Open Quarts II Archive File (.qar) from FPGA source folder



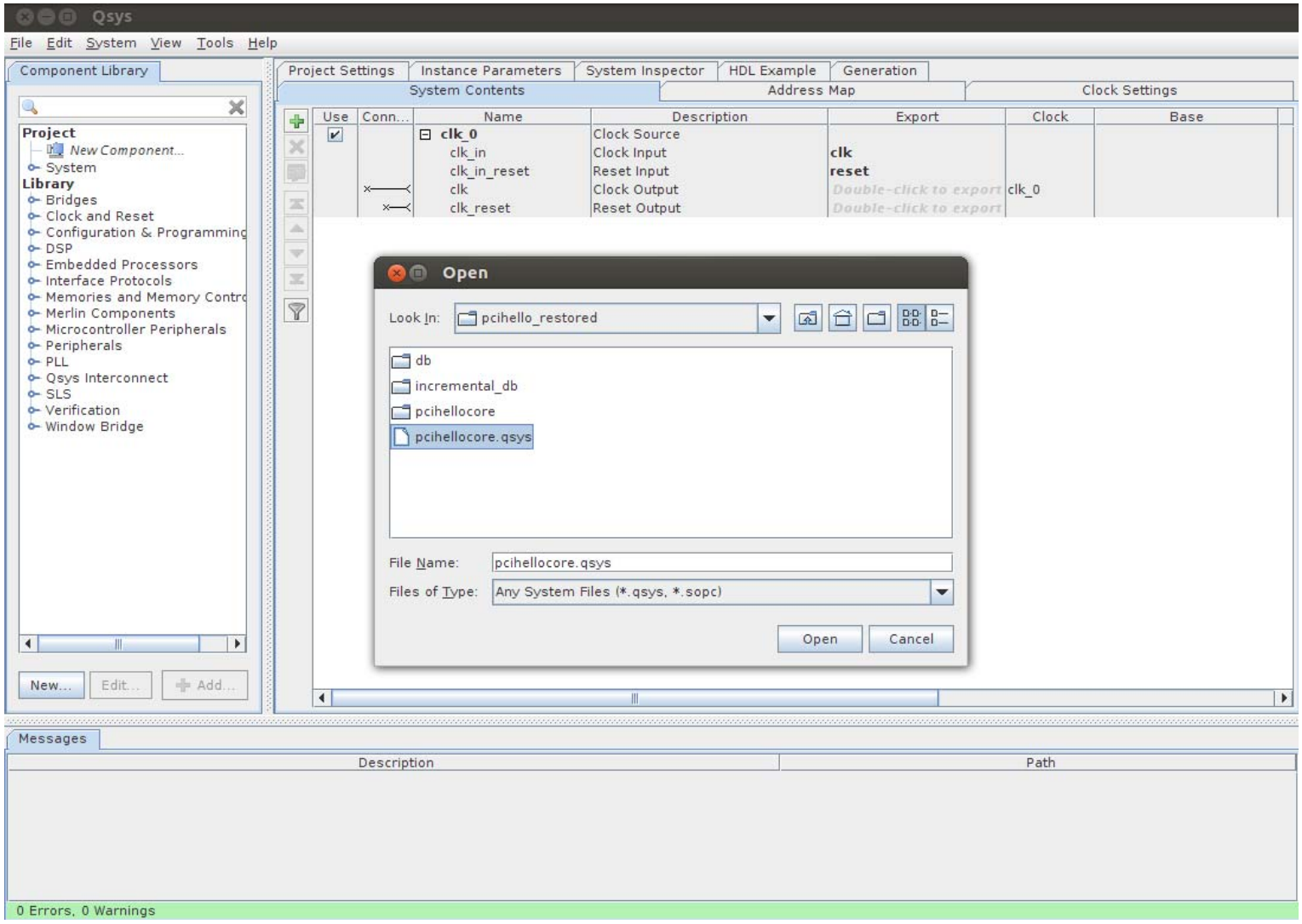
Configuring FPGA using Qsys

Click Okay for Quartus to restore project files using the archive file (.qar)

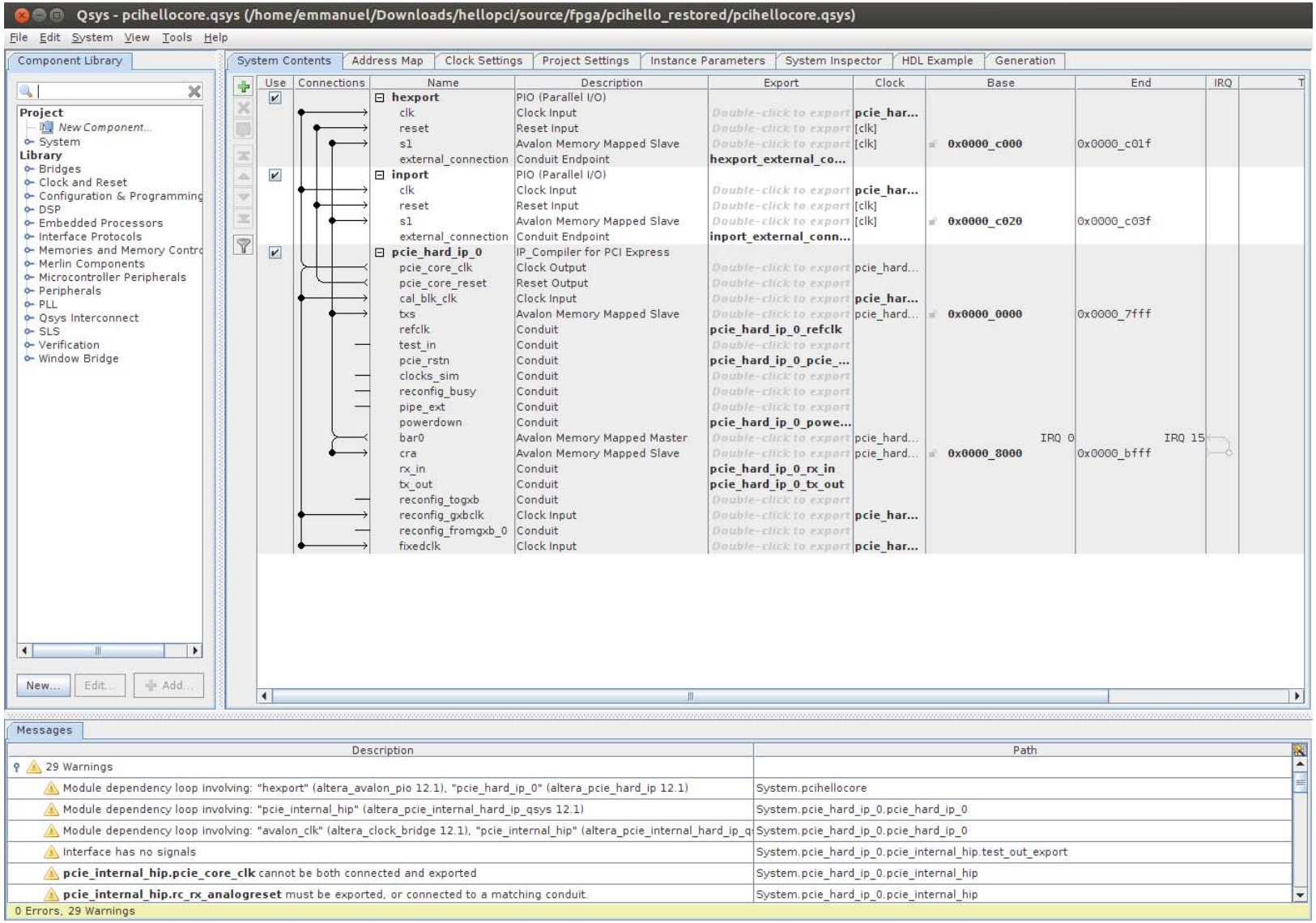


Configuring FPGA using Qsys

Launch Qsys: **Tools > Qsys**

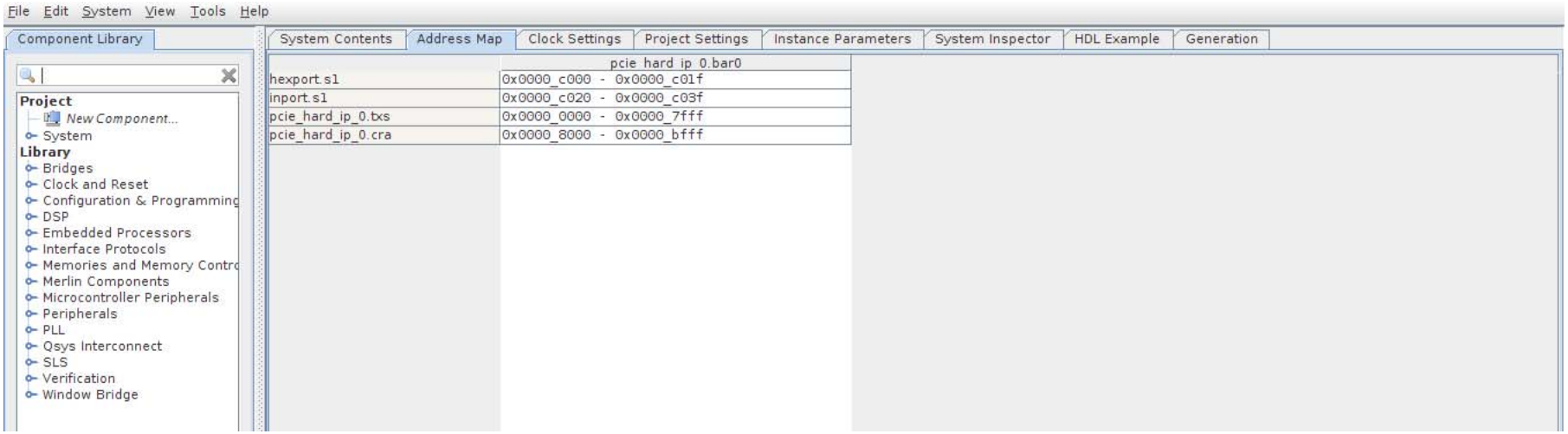


Configuring FPGA using Qsys



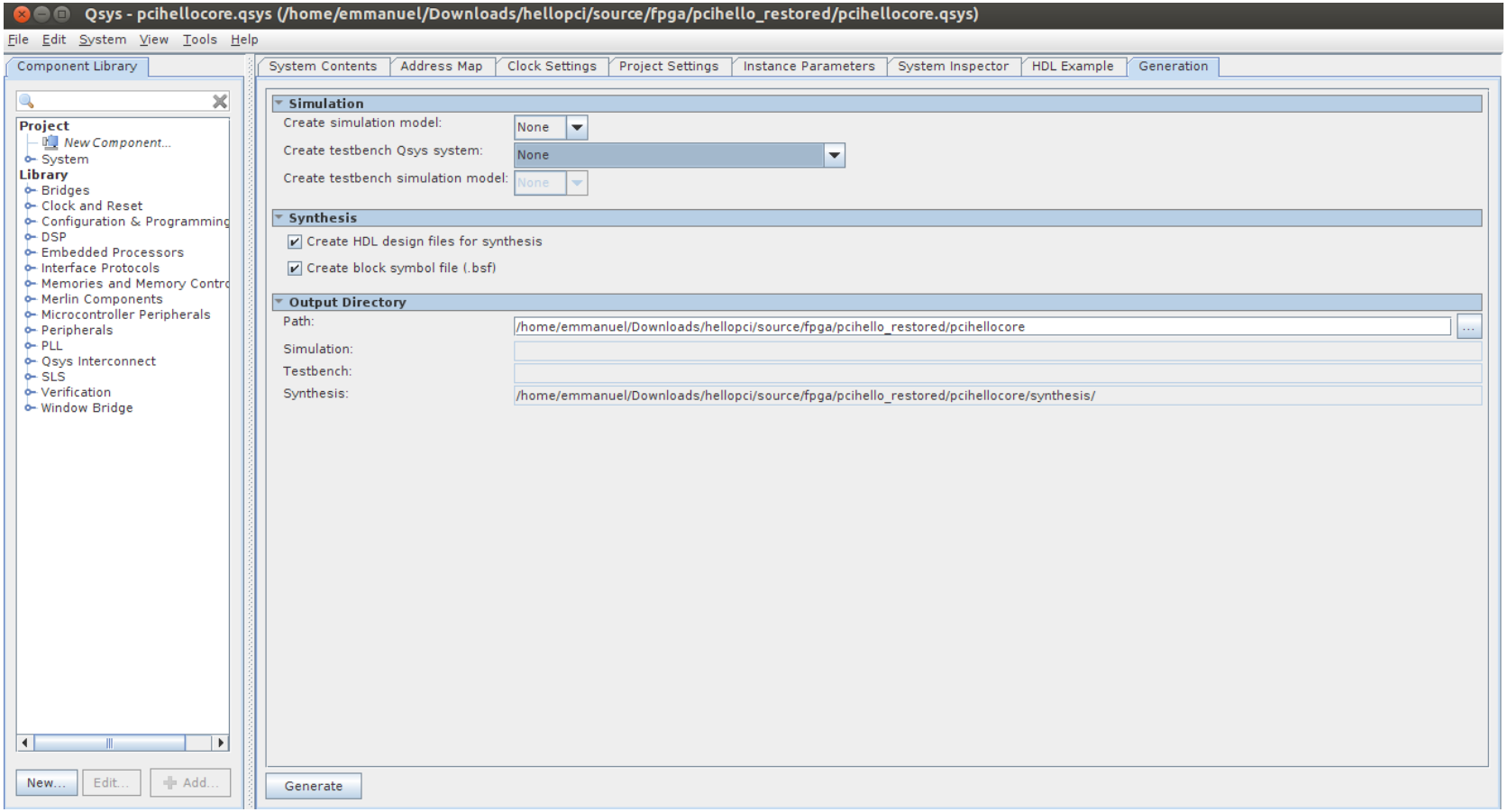
Configuring FPGA using Qsys

View Memory Map. Used in PCIe Driver for Atom



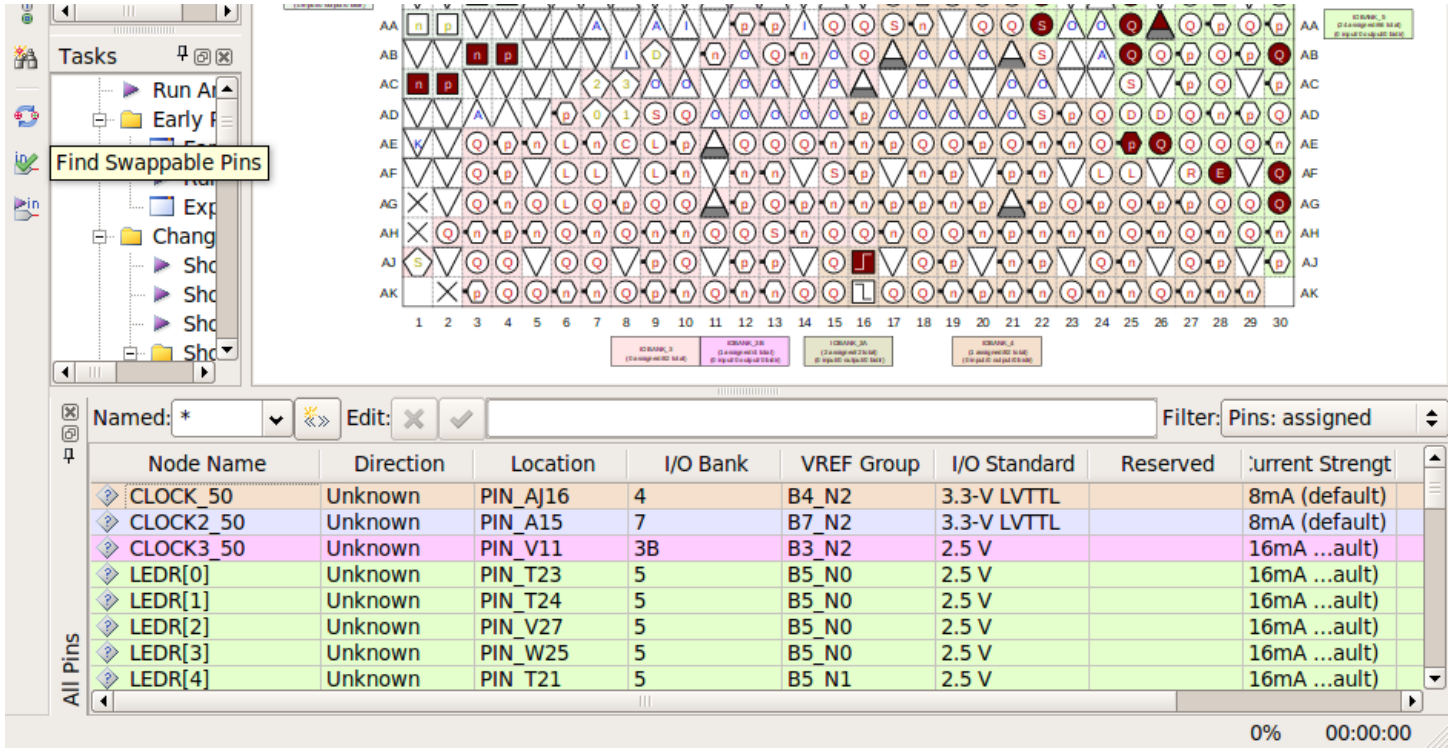
Configuring FPGA using Qsys

Generate Verilog Code for Compilation and close Qsys



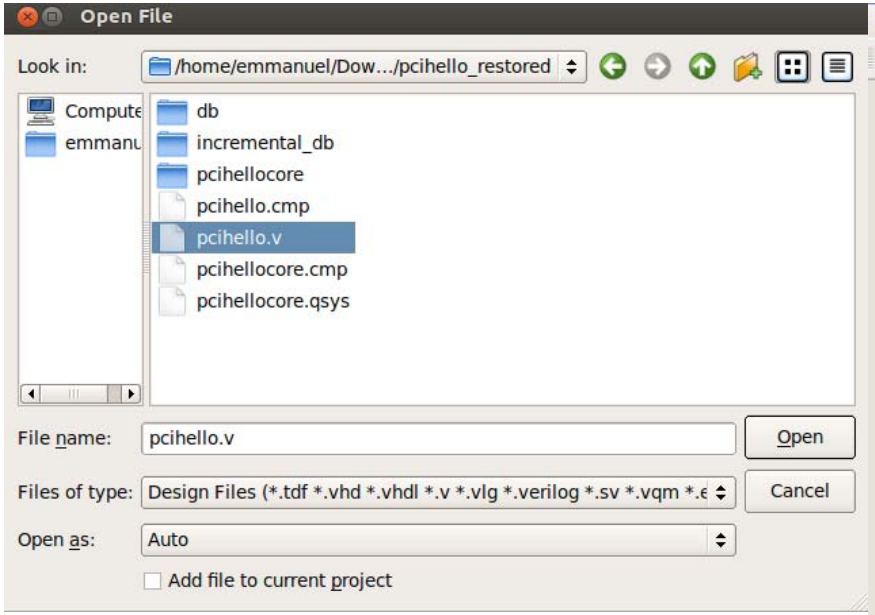
Configuring FPGA using Qsys

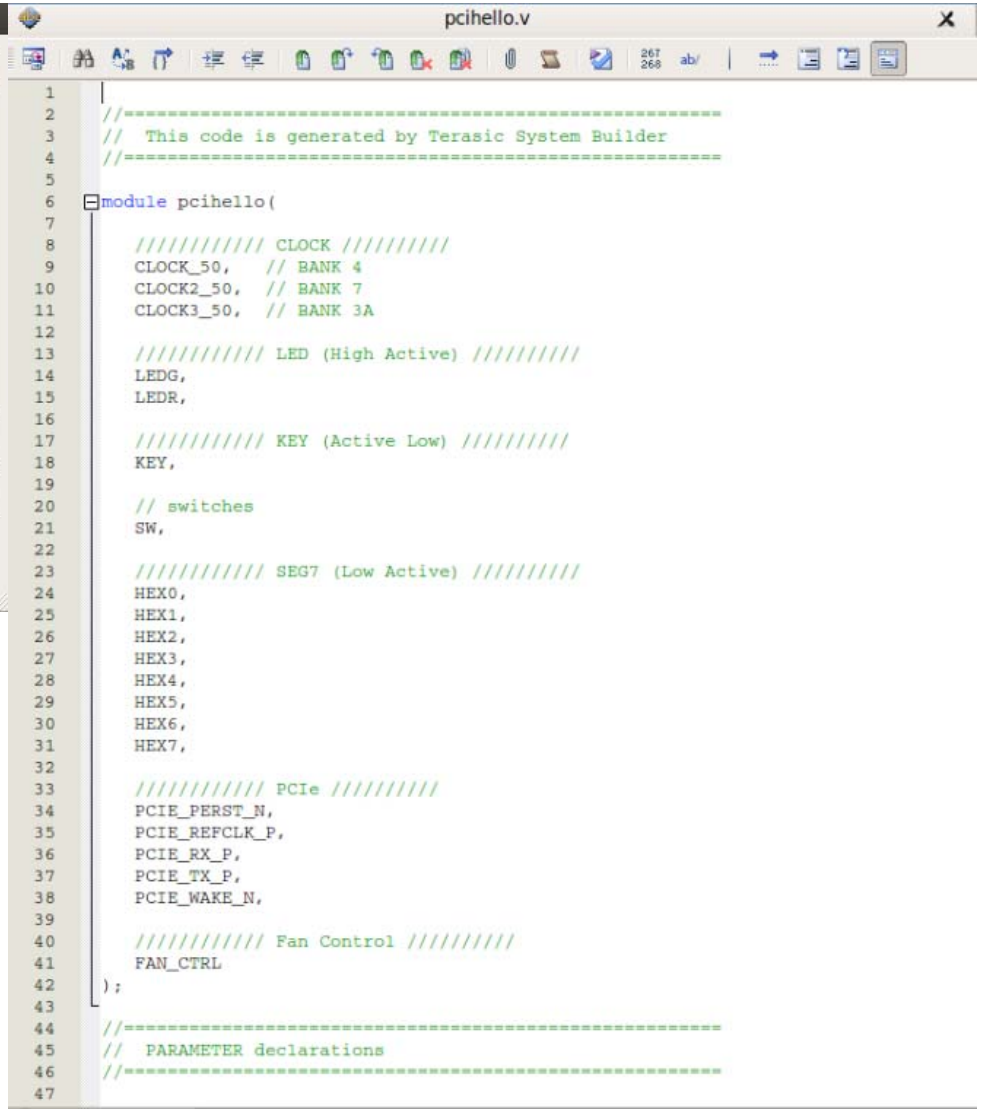
Auto-generated Pin Assignments. No worries



Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strengt
CLOCK_50	Unknown	PIN_AJ16	4	B4_N2	3.3-V LVTTL		8mA (default)
CLOCK2_50	Unknown	PIN_A15	7	B7_N2	3.3-V LVTTL		8mA (default)
CLOCK3_50	Unknown	PIN_V11	3B	B3_N2	2.5 V		16mA ...ault)
LEDR[0]	Unknown	PIN_T23	5	B5_N0	2.5 V		16mA ...ault)
LEDR[1]	Unknown	PIN_T24	5	B5_N0	2.5 V		16mA ...ault)
LEDR[2]	Unknown	PIN_V27	5	B5_N0	2.5 V		16mA ...ault)
LEDR[3]	Unknown	PIN_W25	5	B5_N0	2.5 V		16mA ...ault)
LEDR[4]	Unknown	PIN_T21	5	B5_N1	2.5 V		16mA ...ault)

Configuring FPGA using Qsys





View Verilog
Generated Code and
Start Compilation

Loading SRAM Object File (.sof) on Board

```
emmanuel@emmanuel-desktop:~/Downloads/hellopci/quick$ quartus_pgm -c USB-Blaster -m jtag -o "P;pcihello.sof"
Info: *******************************************************************
Info: Running Quartus II 32-bit Programmer
    Info: Version 12.1 Build 177 11/07/2012 SJ Web Edition
    Info: Copyright (C) 1991-2012 Altera Corporation. All rights reserved.
    Info: Your use of Altera Corporation's design tools, logic functions
    Info: and other software and tools, and its AMPP partner logic
    Info: functions, and any output files from any of the foregoing
    Info: (including device programming or simulation files), and any
    Info: associated documentation or information are expressly subject
    Info: to the terms and conditions of the Altera Program License
    Info: Subscription Agreement, Altera MegaCore Function License
    Info: Agreement, or other applicable license agreement, including,
    Info: without limitation, that your use is for the sole purpose of
    Info: programming logic devices manufactured by Altera and sold by
    Info: Altera or its authorized distributors.  Please refer to the
    Info: applicable agreement for further details.
    Info: Processing started: Thu Mar 20 10:49:21 2014
Info: Command: quartus_pgm -c USB-Blaster -m jtag -o P;pcihello.sof
Info (213045): Using programming cable "USB-Blaster [3-5]"
Info (213011): Using programming file pcihello.sof with checksum 0x009B7B29 for device EP4CGX150DF31@1
Info (209060): Started Programmer operation at Thu Mar 20 10:49:22 2014
Info (209016): Configuring device index 1
Info (209017): Device 1 contains JTAG ID code 0x028040DD
Info (209007): Configuration succeeded -- 1 device(s) configured
Info (209011): Successfully performed operation(s)
Info (209061): Ended Programmer operation at Thu Mar 20 10:49:30 2014
Info: Quartus II 32-bit Programmer was successful. 0 errors, 0 warnings
    Info: Peak virtual memory: 145 megabytes
    Info: Processing ended: Thu Mar 20 10:49:30 2014
    Info: Elapsed time: 00:00:09
    Info: Total CPU time (on all processors): 00:00:01
emmanuel@emmanuel-desktop:~/Downloads/hellopci/quick$
```

Setting Up TFTP

Install Packages

```
emmanuel@emmanuel-desktop:~$ sudo apt-get install xinetd tftp tftpd
[sudo] password for emmanuel:
Reading package lists... Done
Building dependency tree
Reading state information... Done
xinetd is already the newest version.
tftp is already the newest version.
tftpd is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 220 not upgraded.
emmanuel@emmanuel-desktop:~$
```

Add TFTP entry

```
emmanuel@emmanuel-desktop:~$ vi /etc/xinetd.d/tftp
emmanuel@emmanuel-desktop:~$
```

```
service tftp
{
protocol        = udp
port            = 69
socket_type     = dgram
wait            = yes
user            = nobody
server          = /usr/sbin/in.tftpd
server_args     = /home/emmanuel/tftp
disable         = no
}
```

PCIe Driver - Sources

Makefile

```
ifeq ($(KERNELRELEASE),)

# Assume the source tree is where the running kernel was built
# You should set KERNELDIR in the environment if it's elsewhere
        KERNELDIR ?= /lib/modules/$(shell uname -r)/build
# The current directory is passed to sub-makes as argument
        PWD := $(shell pwd)

modules:
        $(MAKE) -C $(KERNELDIR) M=$(PWD) modules
        cp altera_driver.ko /home/emmanuel/tftp

modules_install:
        $(MAKE) -C $(KERNELDIR) M=$(PWD) modules_install

clean:
        rm -rf *.o *~ core .depend .*.cmd *.ko *.mod.c .tmp_versions modules.order Module.symvers

.PHONY: modules modules_install clean

else
# called from kernel build system: just declare what our modules are
        obj-m := altera_driver.o
endif
```

PCIe Driver - Sources

Some Important Functions in Driver Code

```
static int __init altera_driver_init(void) {
   int t = register_chrdev(MAJOR_NUMBER, "de2i150_altera", &file_opts);
   t = t | pci_register_driver(&pci_driver);

   if(t<0)
      printk(KERN_ALERT "altera_driver: error: cannot register char or pci.\n");
   else
     printk(KERN_ALERT "altera_driver: char+pci drivers registered.\n");

   return t;
}
```

```
static int pci_probe(struct pci_dev *dev, const struct pci_device_id *id) {
  int vendor;
  int retval;
  unsigned long resource;

  retval = pci_enable_device(dev);

  if (pci_get_revision(dev) != 0x01) {
    printk(KERN_ALERT "altera_driver: cannot find pci device\n");
    return -ENODEV;
  }

  pci_read_config_dword(dev, 0, &vendor);
  printk(KERN_ALERT "altera_driver: Found Vendor id: %x\n", vendor);

  resource = pci_resource_start(dev, 0);
  printk(KERN_ALERT "altera_driver: Resource start at bar 0: %lx\n", resource);

  hexport = ioremap_nocache(resource + 0XC000, 0x20);
  inport  = ioremap_nocache(resource + 0XC020, 0x20);

  return 0;
}
```

PCI Driver - Build

Yocto Version

```
root@genericx86:~# uname -r
3.10.11-yocto-standard
root@genericx86:~#
```

Build Kernel for .ko file

```
emmanuel@emmanuel-desktop:~/Downloads/hellopci/source/driver$ source /home/emmanuel/poky/1.5/environment-setup-core2-poky-linux

emmanuel@emmanuel-desktop:~/Downloads/hellopci/source/driver$ KERNELDIR=/home/emmanuel/genericx86-64-dora-10.0.0/build/tmp/work/g
enericx86-poky-linux/linux-yocto/3.10.11+gitAUTOINC+363bd856c8_702040ac7c-r0/linux-genericx86-standard-build/ make
make -C /home/emmanuel/genericx86-64-dora-10.0.0/build/tmp/work/genericx86-poky-linux/linux-yocto/3.10.11+gitAUTOINC+363bd856c8_7
02040ac7c-r0/linux-genericx86-standard-build/ M=/home/emmanuel/Downloads/hellopci/source/driver modules
make[1]: Entering directory `/home/emmanuel/genericx86-64-dora-10.0.0/build/tmp/work/genericx86-poky-linux/linux-yocto/3.10.11+gi
tAUTOINC+363bd856c8_702040ac7c-r0/linux-genericx86-standard-build'
  CC [M]  /home/emmanuel/Downloads/hellopci/source/driver/altera_driver.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/emmanuel/Downloads/hellopci/source/driver/altera_driver.mod.o
  LD [M]  /home/emmanuel/Downloads/hellopci/source/driver/altera_driver.ko
make[1]: Leaving directory `/home/emmanuel/genericx86-64-dora-10.0.0/build/tmp/work/genericx86-poky-linux/linux-yocto/3.10.11+git
AUTOINC+363bd856c8_702040ac7c-r0/linux-genericx86-standard-build'
cp altera_driver.ko /home/emmanuel/tftp
emmanuel@emmanuel-desktop:~/Downloads/hellopci/source/driver$
```

If you have v3.0.32, you do not have to do this step. You can do use the kernel object in the **quick** folder

User Application - Sources

Makefile

```
app: app.c
	$(CC) app.c -o app
	cp app /home/emmanuel/tftp

clean:
	rm -rf *.o *~ core app
```

App Source Code

```
#include <unistd.h>
#include <fcntl.h>

unsigned char hexdigit[] = {0x3F, 0x06, 0x5B, 0x4F,
                            0x66, 0x6D, 0x7D, 0x07,
                            0x7F, 0x6F, 0x77, 0x7C,
                            0x39, 0x5E, 0x79, 0x71};

int main() {
  int i, j, k;

  int dev = open("/dev/de2i150_altera", O_RDWR);

  for (i=0; i>-1; i++) {
    read(dev, &j, 4);
    k = hexdigit[j & 0xF]
      | (hexdigit[(j >>  4) & 0xF] << 8)
      | (hexdigit[(j >>  8) & 0xF] << 16)
      | (hexdigit[(j >> 12) & 0xF] << 24);
    k = ~k;
    write(dev, &k, 4);
  }

  close(dev);
  return 0;
}
```

User Application - Build

```
emmanuel@emmanuel-desktop:~/Downloads/hellopci/source/app$ KERNELDIR=/home/emmanuel/genericx86-64-dora-10.0.0/build/tmp/work/gene
ricx86-poky-linux/linux-yocto/3.10.11+gitAUTOINC+363bd856c8_702040ac7c-r0/linux-genericx86-standard-build/ make
i586-poky-linux-gcc  -m32 -march=core2 -msse3 -mtune=generic -mfpmath=sse --sysroot=/home/emmanuel/poky/1.5/sysroots/core2-poky-l
inux app.c -o app
cp app /home/emmanuel/tftp
emmanuel@emmanuel-desktop:~/Downloads/hellopci/source/app$
```

Again, if you have Yocto v3.0.32, you do not have to do this step. You can do use the **app** in the **quick** folder

Task Execution

Copy PCIe Driver and Application Code via TFTP

```
root@genericx86:~# ls
root@genericx86:~#
root@genericx86:~#
root@genericx86:~# tftp -g -r altera_driver.ko 192.168.2.101
root@genericx86:~# tftp -g -r app 192.168.2.101
root@genericx86:~#
root@genericx86:~#
root@genericx86:~# ls
altera_driver.ko  app
root@genericx86:~#
```

Task Execution

Insert Module into Yocto Kernel and run app

```
root@genericx86:~#
root@genericx86:~# insmod altera_driver.ko
root@genericx86:~# mknod /dev/de2i150_altera c 91 1
mknod: '/dev/de2i150_altera': File exists
root@genericx86:~# ./app
```

Kernel Debug Messages

```
[   36.899466] altera_driver: Found Vendor id: 41172
[   36.899743] altera_driver: Resource start at bar 0: 80200000
[   36.900144] altera_driver: char+pci drivers registered.
[   73.872062] altera_driver: opened 1 time(s)
[  137.822454] altera_driver: device closed.
root@genericx86:~#
```

Thank you for your time!