



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:

[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 Quartus 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
```

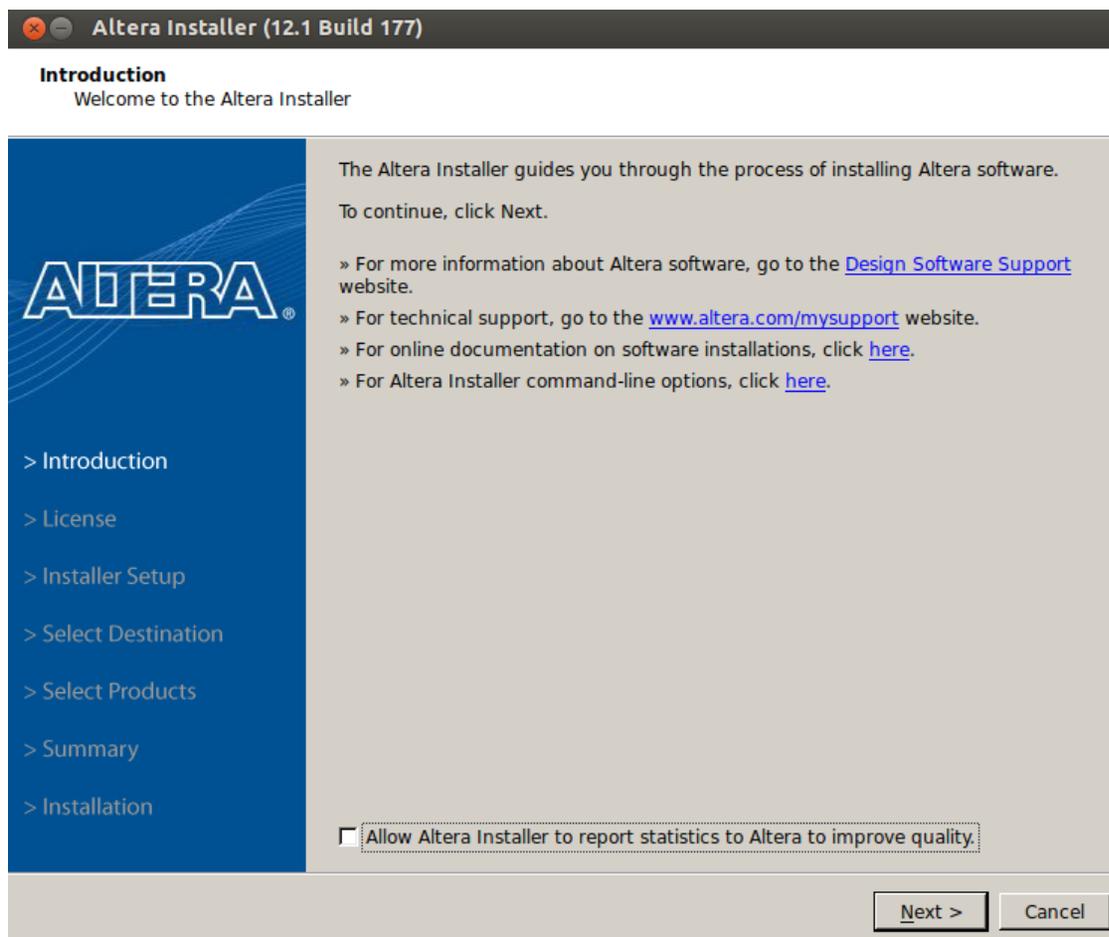
Navigate through directory and launch installer

```
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

```
emmanuel@emmanuel-desktop:/etc/udev/rules.d  
# Altera USB-Blaster rule to set mode to 666  
SUBSYSTEM=="usb",\  
ENV{DEVTYPE}=="usb_device",\  
ATTR{idVendor}=="09fb",\  
ATTR{idProduct}=="6001",\  
MODE="0666",\  
NAME="bus/usb/$env{BUSNUM}/$env{DEVNUM}",\  
RUN+="/bin/chmod 0666 %c"
```

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
```



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
```

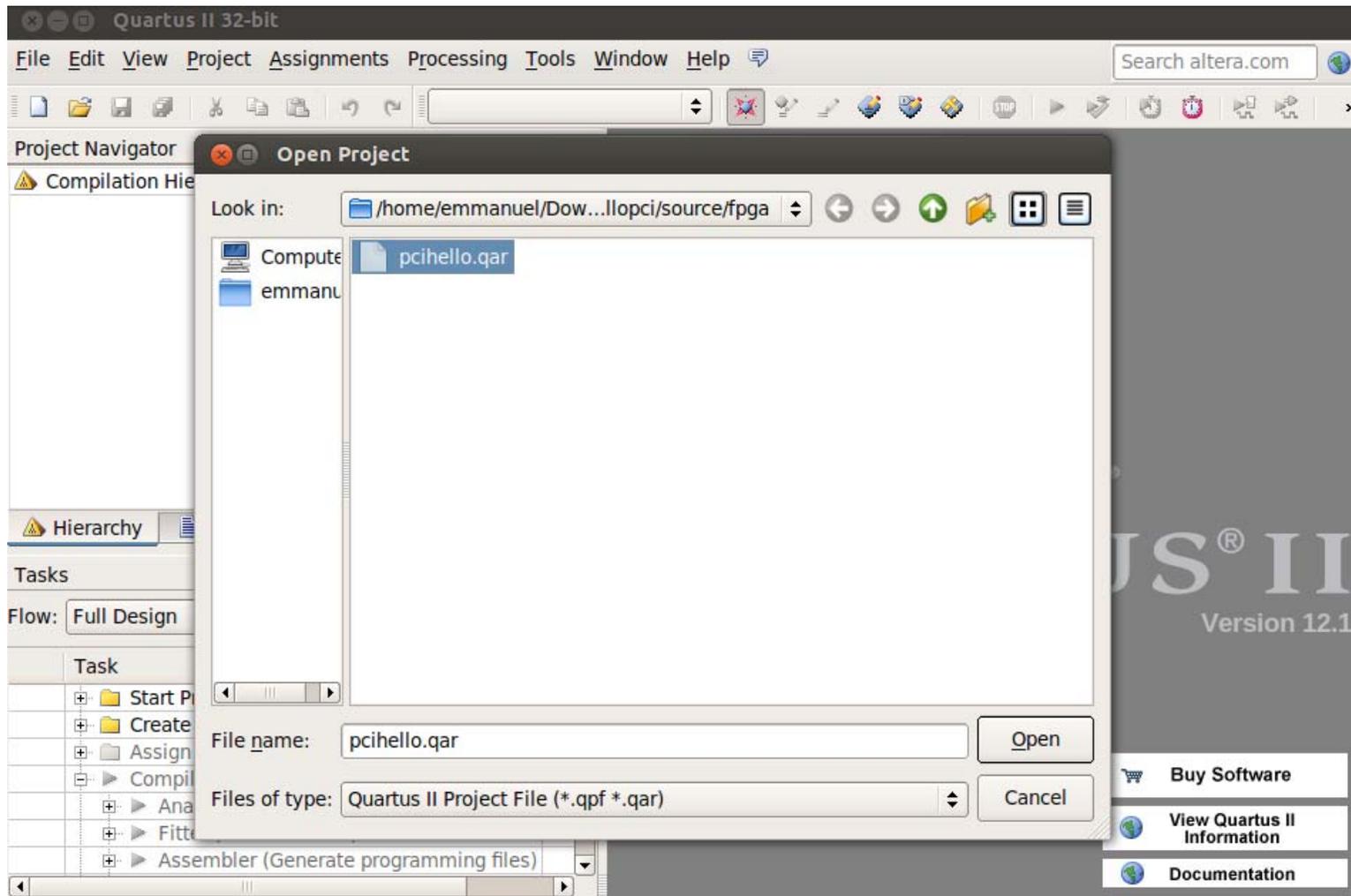
Launch Quartus

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



Configuring FPGA using Qsys

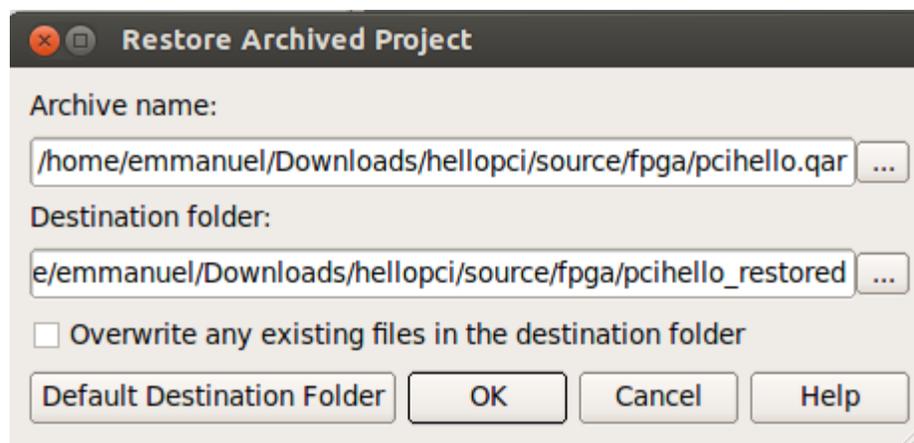
Open Quartus II Archive File (.qar) from FPGA source folder





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

The screenshot shows the Qsys software interface. The main window displays the 'System Contents' table, which lists components and their connections. An 'Open' dialog box is overlaid on the main window, showing the file 'pchellocore.qsys' selected in the 'Look In' directory 'pchello_restored'.

Use	Conn...	Name	Description	Export	Clock	Base
<input checked="" type="checkbox"/>		clk_0	Clock Source			
		clk_in	Clock Input			
		clk_in_reset	Reset Input			
	x--<	clk	Clock Output	clk	clk_0	
	x-->	clk_reset	Reset Output	reset		

Open dialog box details:

- Look In: pchello_restored
- Files of Type: Any System Files (*.qsys, *.sopc)
- File Name: pchellocore.qsys
- Files listed: db, incremental_db, pchellocore, pchellocore.qsys



Configuring FPGA using Qsys

The screenshot displays the Qsys software interface for configuring an FPGA. The main window shows the 'System Contents' tab, which lists various components and their connections. The 'Component Library' on the left shows the project structure, including 'System' and 'Library' components. The 'Messages' panel at the bottom shows 29 warnings and 0 errors.

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>	→	hexport	PIO (Parallel I/O)	Double-click to export	pcie_har...			
	→	clk	Clock Input	Double-click to export	[clk]			
	→	reset	Reset Input	Double-click to export	[clk]			
	→	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_c000	0x0000_c01f	
	→	external_connection	Conduit Endpoint	hexport_external_co...				
<input checked="" type="checkbox"/>	→	inport	PIO (Parallel I/O)	Double-click to export	pcie_har...			
	→	clk	Clock Input	Double-click to export	[clk]			
	→	reset	Reset Input	Double-click to export	[clk]			
	→	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_c020	0x0000_c03f	
	→	external_connection	Conduit Endpoint	inport_external_conn...				
<input checked="" type="checkbox"/>	→	pcie_hard_ip_0	IP_Compiler for PCI Express	Double-click to export	pcie_hard...			
	→	pcie_core_clk	Clock Output	Double-click to export	pcie_hard...			
	→	pcie_core_reset	Reset Output	Double-click to export	pcie_hard...			
	→	cal_blk_clk	Clock Input	Double-click to export	pcie_hard...			
	→	bxs	Avalon Memory Mapped Slave	Double-click to export	pcie_hard...	0x0000_0000	0x0000_7fff	
	→	refclk	Conduit	pcie_hard_ip_0_refclk				
	→	test_in	Conduit	Double-click to export				
	→	pcie_rstn	Conduit	pcie_hard_ip_0_pcie_...				
	→	clocks_sim	Conduit	Double-click to export				
	→	reconfig_busy	Conduit	Double-click to export				
	→	pipe_ext	Conduit	Double-click to export				
	→	powerdown	Conduit	pcie_hard_ip_0_powe...				
	→	bar0	Avalon Memory Mapped Master	Double-click to export	pcie_hard...			
	→	cra	Avalon Memory Mapped Slave	Double-click to export	pcie_hard...	0x0000_8000	IRQ 0	IRQ 15
	→	rx_in	Conduit	pcie_hard_ip_0_rx_in			0x0000_bfff	
	→	tx_out	Conduit	pcie_hard_ip_0_tx_out				
	→	reconfig_togxb	Conduit	Double-click to export				
	→	reconfig_gxbclk	Clock Input	Double-click to export	pcie_har...			
	→	reconfig_fromgxb_0	Conduit	Double-click to export				
	→	fixedclk	Clock Input	Double-click to export	pcie_har...			

Messages

Description	Path
29 Warnings	
Module dependency loop involving: "hexport" (altera_avalon_pio 12.1), "pcie_hard_ip_0" (altera_pcie_hard_ip 12.1)	System.pcihellocore
Module dependency loop involving: "pcie_internal_hip" (altera_pcie_internal_hard_ip_qsys 12.1)	System.pcie_hard_ip_0.pcie_hard_ip_0
Module dependency loop involving: "avalon_clk" (altera_clock_bridge 12.1), "pcie_internal_hip" (altera_pcie_internal_hard_ip_qsys 12.1)	System.pcie_hard_ip_0.pcie_hard_ip_0
Interface has no signals	System.pcie_hard_ip_0.pcie_internal_hip.test_out_export
pcie_internal_hip.pcie_core_clk cannot be both connected and exported	System.pcie_hard_ip_0.pcie_internal_hip
pcie_internal_hip.rc_rx_analogreset must be exported, or connected to a matching conduit.	System.pcie_hard_ip_0.pcie_internal_hip

0 Errors, 29 Warnings



Configuring FPGA using Qsys

View Memory Map. Used in PCIe Driver for Atom

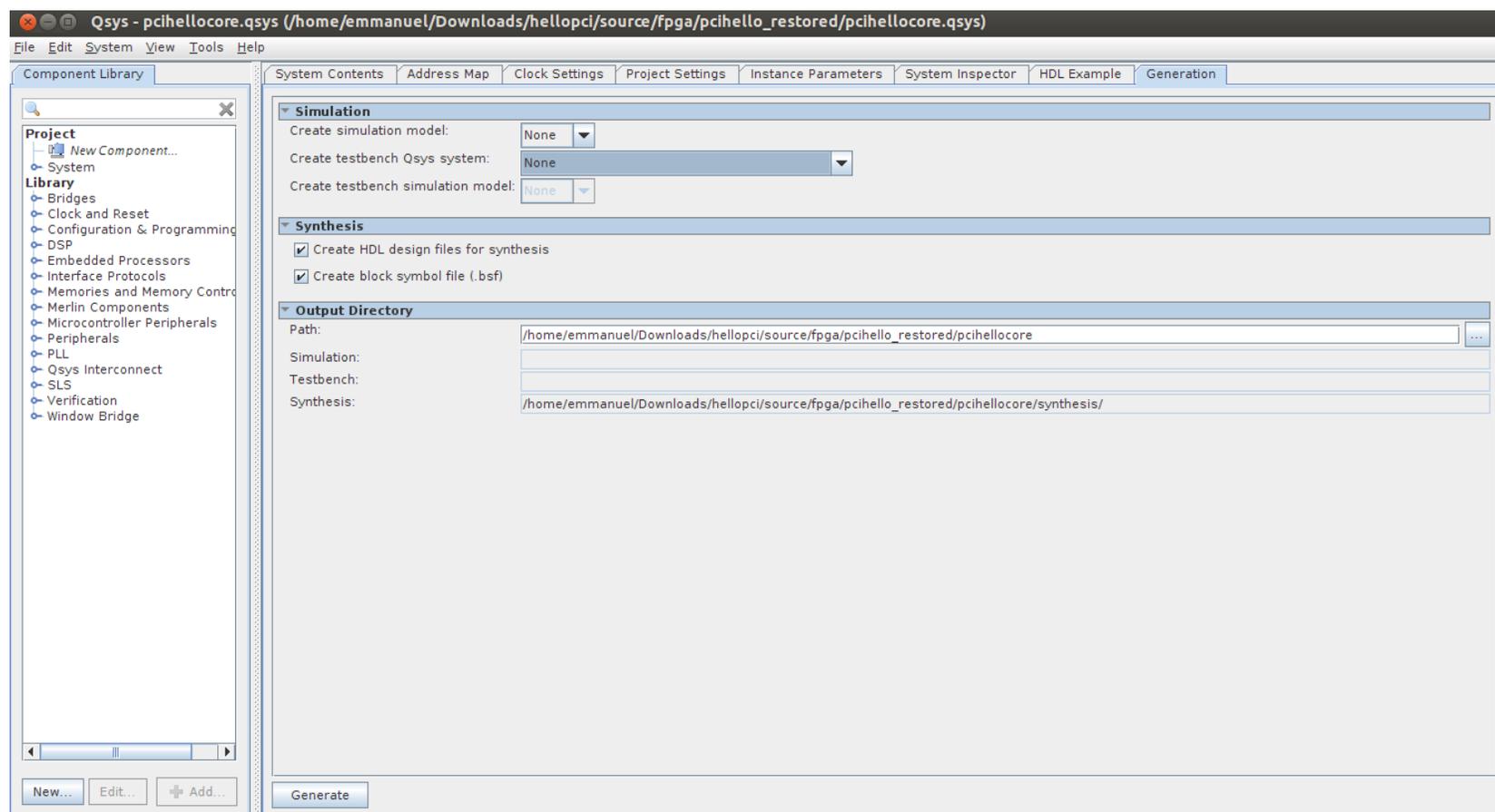
The screenshot shows the Qsys software interface with the 'Address Map' tab selected. The memory map table is as follows:

Component	Start Address	End Address
hexport s1	0x0000_c000	0x0000_c01f
inport s1	0x0000_c020	0x0000_c03f
pcie_hard_ip_0.txs	0x0000_0000	0x0000_7fff
pcie_hard_ip_0.cra	0x0000_8000	0x0000_bfff



Configuring FPGA using Qsys

Generate Verilog Code for Compilation and close Qsys





Configuring FPGA using Qsys

The screenshot displays the Qsys software interface. On the left, an 'Open File' dialog box is open, showing a file browser with the path `/home/emmanuel/Dow.../pcihello_restored`. The file `pcihello.v` is selected. The 'File name' field contains `pcihello.v`, and the 'Files of type' dropdown is set to 'Design Files (*.tdf *.vhd *.vhdl *.v *.vlg *.verilog *.sv *.vqm *.e)'. The 'Open as' dropdown is set to 'Auto'. Below the dialog, there is a checkbox for 'Add file to current project'.

On the right, the Verilog code editor shows the content of `pcihello.v`. The code is as follows:

```
1 //-----  
2 // This code is generated by Terasic System Builder  
3 //-----  
4  
5  
6 module pchello(  
7  
8     /////////////// CLOCK ///////////////  
9     CLOCK_50, // BANK 4  
10    CLOCK2_50, // BANK 7  
11    CLOCK3_50, // BANK 3A  
12  
13    /////////////// LED (High Active) ///////////////  
14    LEDG,  
15    LEDR,  
16  
17    /////////////// KEY (Active Low) ///////////////  
18    KEY,  
19  
20    // switches  
21    SW,  
22  
23    /////////////// SEG7 (Low Active) ///////////////  
24    HEX0,  
25    HEX1,  
26    HEX2,  
27    HEX3,  
28    HEX4,  
29    HEX5,  
30    HEX6,  
31    HEX7,  
32  
33    /////////////// PCIe ///////////////  
34    PCIE_PERST_N,  
35    PCIE_REFCLK_P,  
36    PCIE_RX_P,  
37    PCIE_TX_P,  
38    PCIE_WAKE_N,  
39  
40    /////////////// Fan Control ///////////////  
41    FAN_CTRL  
42 );  
43  
44 //-----  
45 // PARAMETER declarations  
46 //-----  
47
```

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

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

App Source Code

Makefile

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

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

```
#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/genericx86-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-linux 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!

