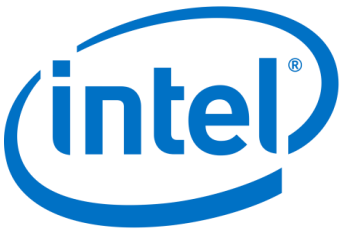

EMBEDDED NIOS DEVCLOUD QUICKSTART

To learn more, visit http://fpgauniversity.intel.com

Contents

1 Introduction

This lab teaches you how to create an embedded system implanted in programmable logic using the Intel Nios II processor, sometimes referred to as a “soft” processor. The Nios II can be synthesized on any Intel FPGA device and has a built-in programmable logic fabric that can be easily modified to suit an applications’ requirements. Intel SoC FPGA devices contain a processor built from standard cells that cannot be changed without redesigning the chip and are therefore called a textithard processor system. The Nios II processor is supported by a rich set of peripherals and intellectual property (IP) blocks built that can be configured and connected to the processor using the Platform Designer tool within the Intel Quartus Prime software suit. Intel also distributes the Nios II Software Build Tools (SBT) within the Quartus download for use with Eclipse during software development.

This lab is organized to run on several Intel FPGA development kits through the Intel Devcloud on any computer that is set up to run the Intel Devcloud environment.

2 Assumptions

This lab assumes the following:

- Prior FPGA knowledge
- Intel Devcloud registration and SSH key set up
- MobaXterm installed and set up
- X2GO installed and set up
- WinSCP installed and set up
- Local USB Blaster and devkit set up

If any of the above assumptions are incorrect, please refer to the relevant set up guides.

3 Requirements

3.1 Hardware Requirements

This lab requires a local DE-10 Lite Development Kit connected to your computer. The development kit will be programmed locally on your computer through the Devcloud environment.

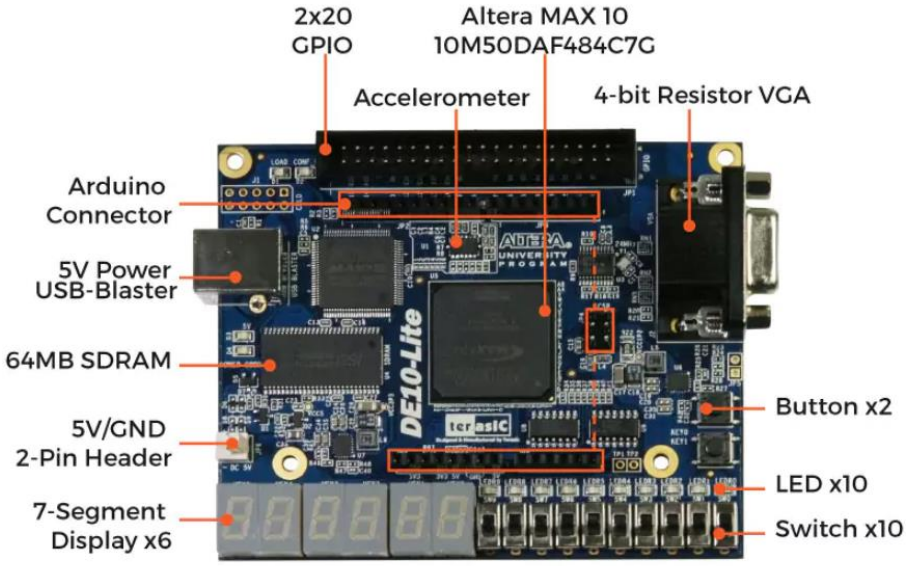

Figure 1: DE-10 Lite Development Kit

3.2 Software Requirements

To install Quartus Prime Lite, follow these instructions inside the Devcloud environment:

- Open up the terminal and run the command:

```
cp /glob/development-tools/versions/intelFPGA_lite/quartus_setup.sh ~
```

- You have to change the Quartus version to “LITE’ so run this command to open the file

```
vi ~/quartus_setup.sh
```

- In this text file, press the key “i” to edit the document. Use the arrow keys to maneuver to the top and change **QUARTUS_EDITION** from “PRO” to “LITE” and hit esc. After that press CTRL + ZZ to save the document and exit.
- After this you have to run it so input this command:

```
source ~/quartus_setup.sh
```

4 Walkthrough

Open up MobaXterm and login to the Devcloud. Start up the X2GO tunnel and open up the visual environment.

4.1 File Setup

- ☐ Unzip the zip folder called **NiosDevcloud.zip**. You will be left with a folder called **NiosDevcloud** which contains the following files: **NiosDevcloud.qar** and **software.zip**.
- ☐ Startup WinSCP. Login to the Devcloud. You should see a screen like below.

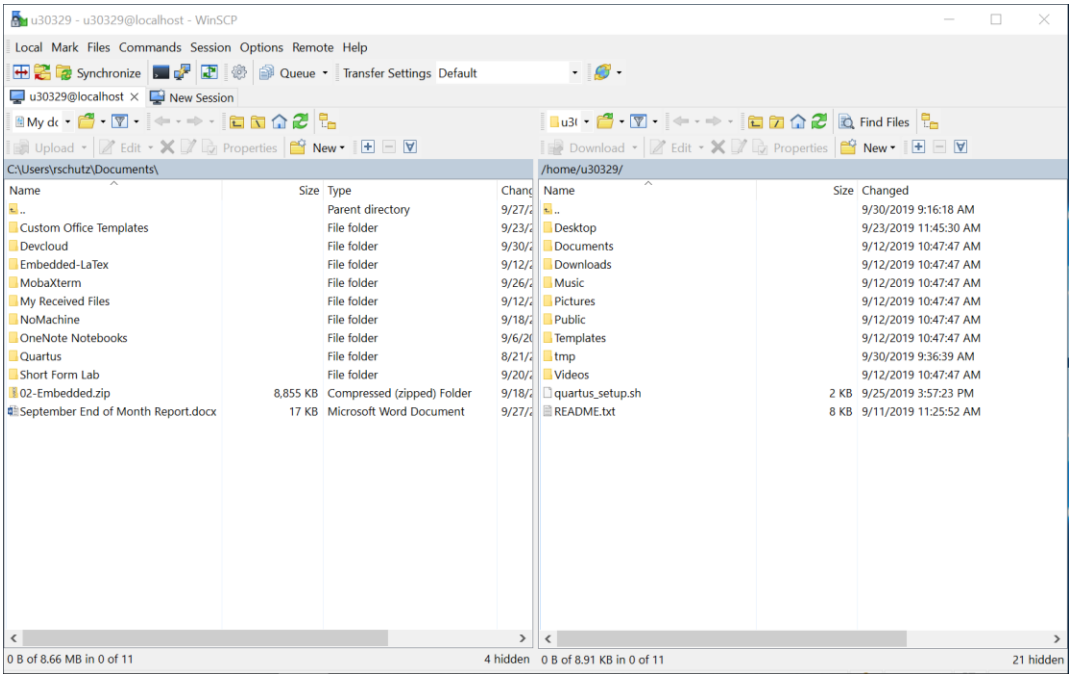

Figure 2: WinSCP Startup Window

- ☐ Go to the desktop folder in WinSCP and drag and drop the **NiosDevcloud** folder into it.
- ☐ You should see the following window. Press OK.

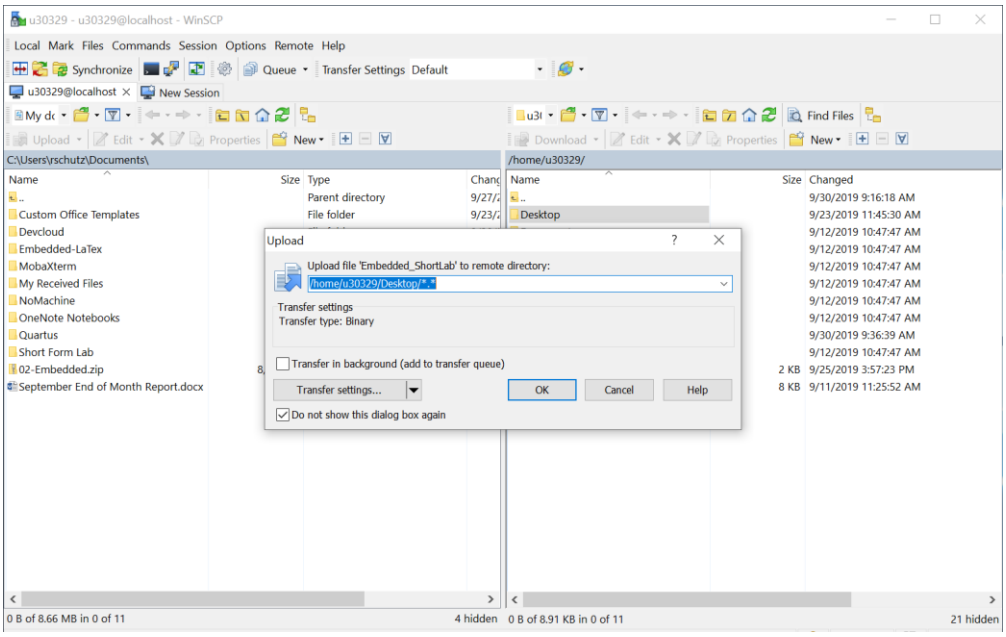

Figure 3: WinSCP File Upload

☐ Now Open X2GO Window which has already been connected to. Inside X2GO you should see the folder **NiosDevcloud**.

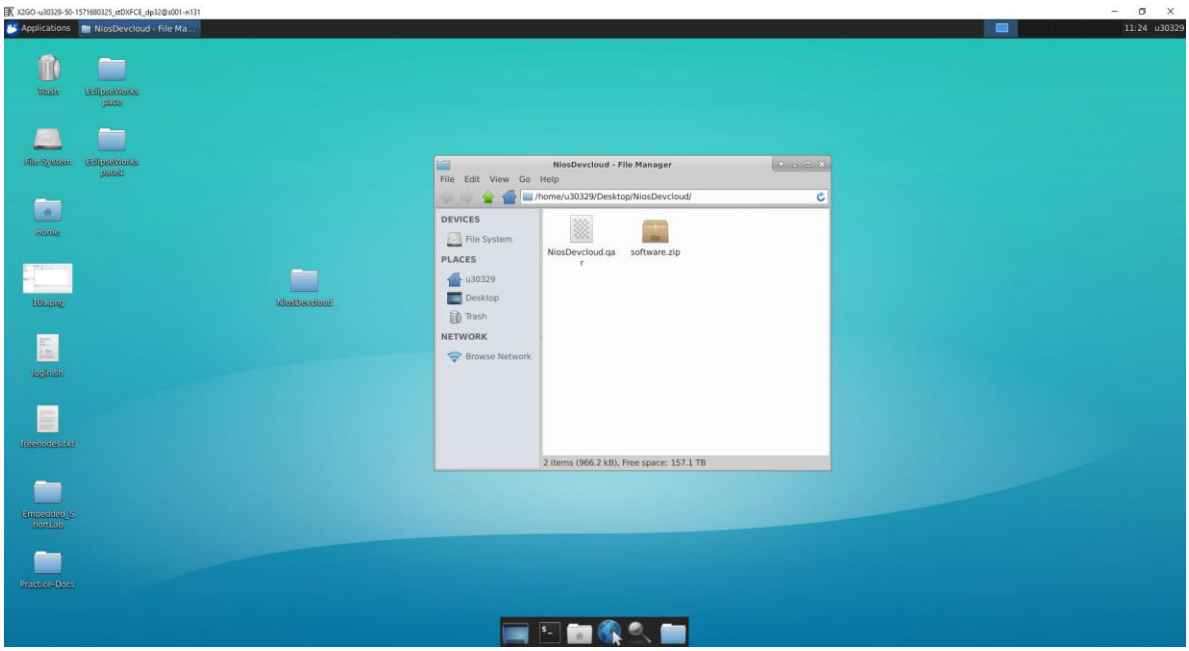

Figure 4: X2GO Desktop

☐ Right click on the desktop and click **Open Terminal Here**.

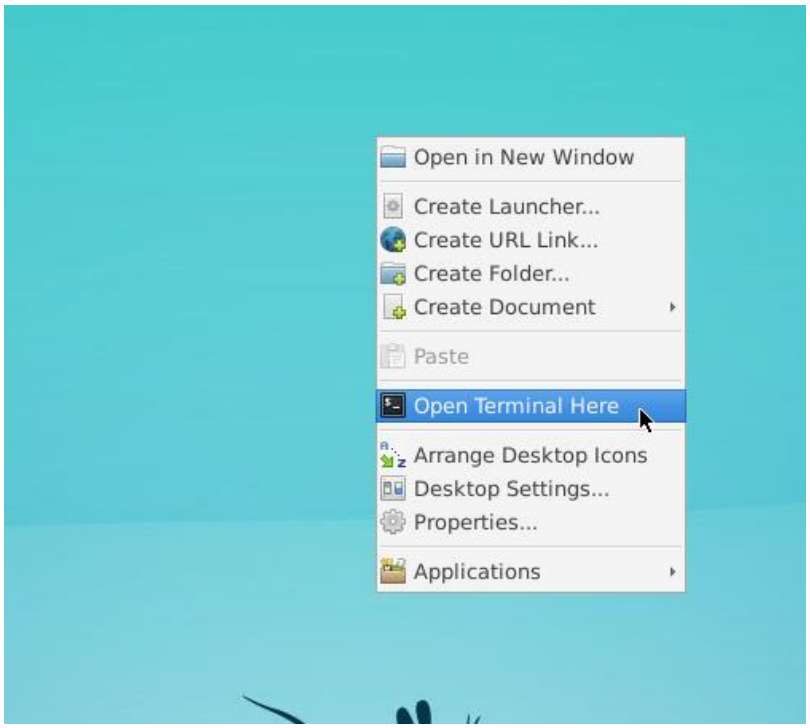

Figure 5: Terminal Open

☐ Open up Quartus by typing: **quartus &** and hitting enter.

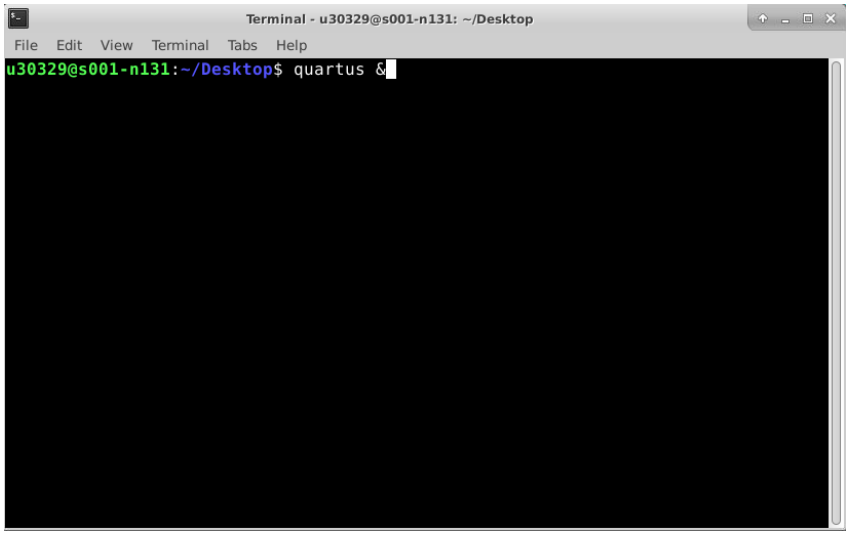

Figure 6: Opening Quartus GUI

☐ Go to **File -> Open Project** and click on the **NiosDevcloud.qar** file.

NO SPACES MUST BE PRESENT IN FOLDER NAME OR FILE NAME

- ☐ Select a destination folder where you want your project to be restored, by clicking on **...** near the destination folder. Make sure the folder it is placed in contains **no spaces**.
- ☐ Select **OK**, and Quartus will begin unpacking all its files.
  You will be able to navigate around the main Quartus window. We will start modifying our system by using Platform Designer.

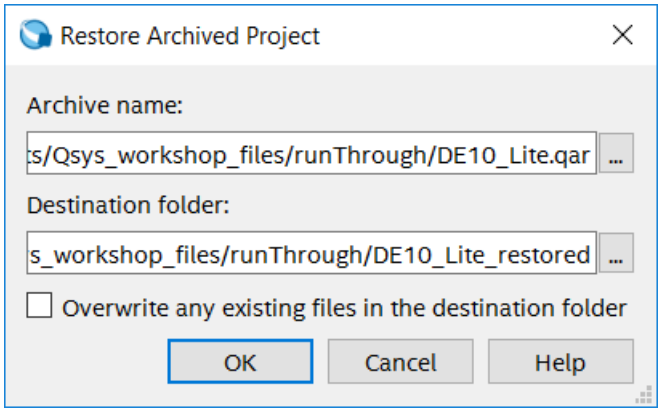

Figure 7:Selecting Archive Name and Destination Folder for the .qar file

4.2 Hardware Design

Building the Platform Designer system is a highly efficient way of designing systems with or without a processor.

☐ Launch Platform Designer tool from Quartus: **Tools -> Platform Designer**. A pop-up should open asking if you would like to open a system file. If it does not, click **File -> Open**. Open the file **nios_setup_v2.qsys**. The entire design of the system was completed before-hand.

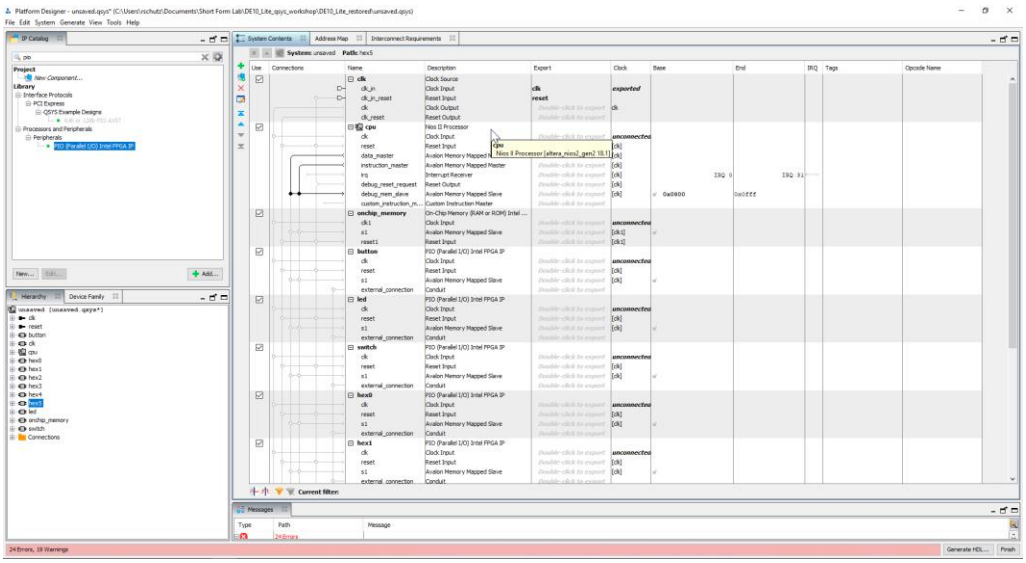

Figure 8: Platform Designer Main Panel

☐ Save your Platform Designer system by using **File -> Save** The information is saved in a .qsys file. (Note: By saving, you still have not generated the files that you need for Quartus compilation or with the Eclipse SBT).

☐ Click on the button **Generate HDL**. Click **Generate** on the panel that appears.

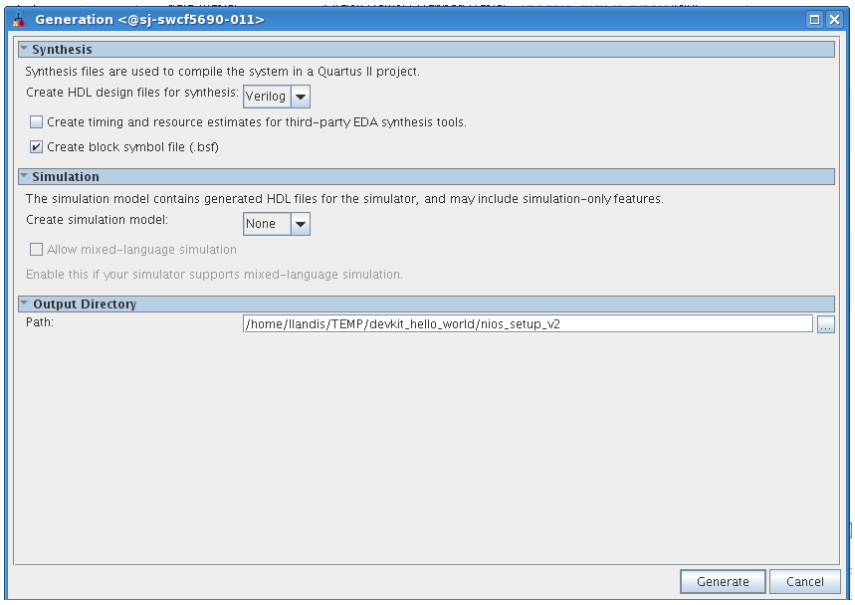

Figure 9: HDL Generation Panel

☐ When the file generation is complete, click **Finish** to exit the Platform Designer window.

☐ In the Quartus main window, go to **Project -> Add/Remove Files.**

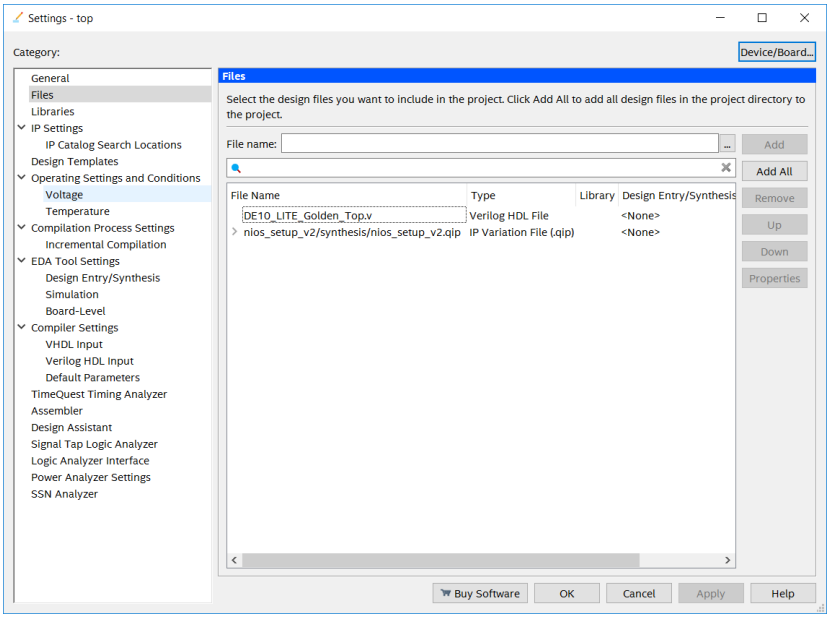

Figure 10: Quartus Add/Remove Files Pane

☐ Add the **nios_setup_v2.qip** file. (You can also just add the nios_setup_v2.qsys file). Press the "…" button to open the file dialog box.

- ☐ The **nios_setup_v2.qip** file should be found under **nios_setup_v2 -> Synthesis** directory in your project.
- ☐ You will need to change the filter to display **All files** if you cannot see it.

☐ Next, you will compile the design. Click on the play button as shown below.

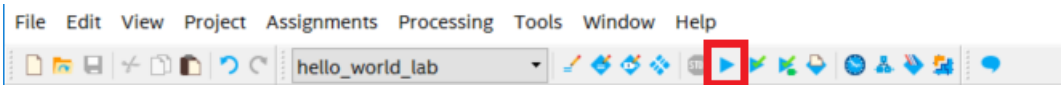

Figure 11: Compilation Button on Quartus Toolbar

☐ Launch the Programmer by clicking **Tools -> Programmer**

☐ Next, you need to download what is called a **.sof** file or SRAM object file. This is the programming image file that gets downloaded in the FPGA. The default location is <working_directory>/output_files.

☐ Right click on the first row <none> under File and click on **Change File**. Navigate to the output_files directory and select **top.sof**.

☐ Click **Open**.

☐ In the first row under **Program/Configure** click in the check box as shown.

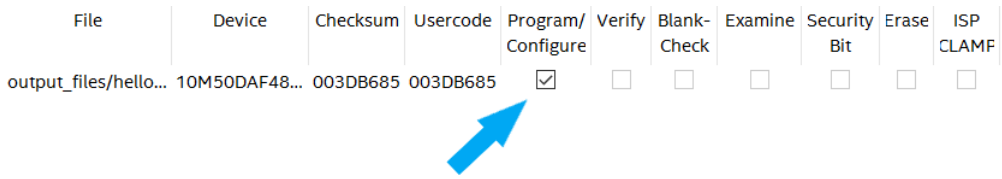

Figure 12: Program/Configure Checkbox

☐ Click on **Hardware Setup**, located in the top left corner of the programmer window. In the currently selected hardware section, click on the drop-down menu and select the **USB Blaster**. It may say something like localhost:13090[USB-0].

☐ Click **Start**, located on the left of the programmer window. When programming is complete, the progress meter should read 100% (Successful).

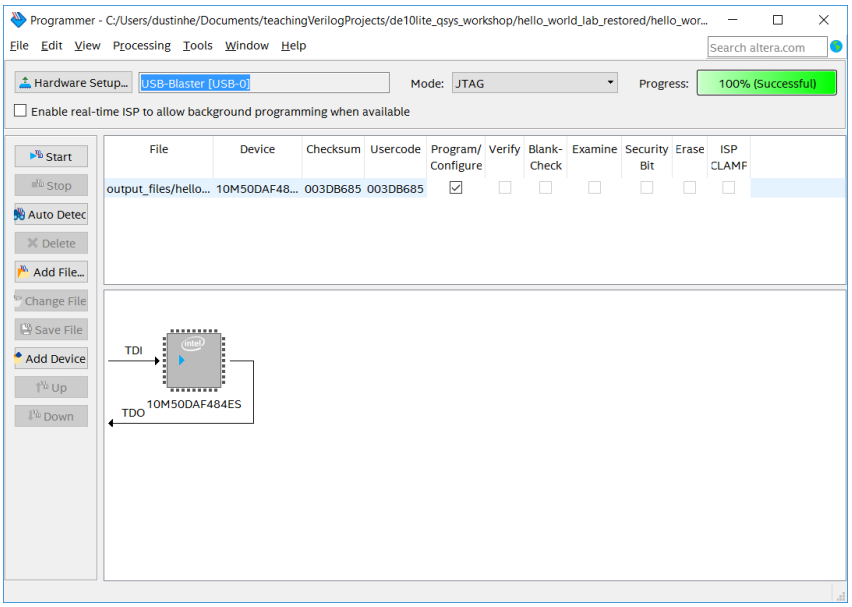

Figure 13: Programmer Progress Successful

Your FPGA hardware design is now complete!

4.3 Software Design

The NIOS Software Build Tools for Eclipse are included as part of Quartus. These tools will help manage creation of the application software and Board Support Package (BSP).

☐ Launch **Tools -> NIOS II Software Build Tools** for Eclipse. You can use the default location that Eclipse picks for you.

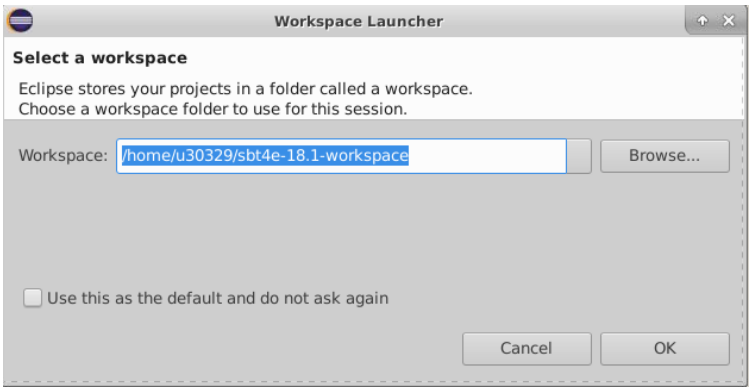

Figure 14: Initial Workspace Setup

☐ Click **OK** in the Workspace Launcher. Next, the Eclipse SBT will launch.

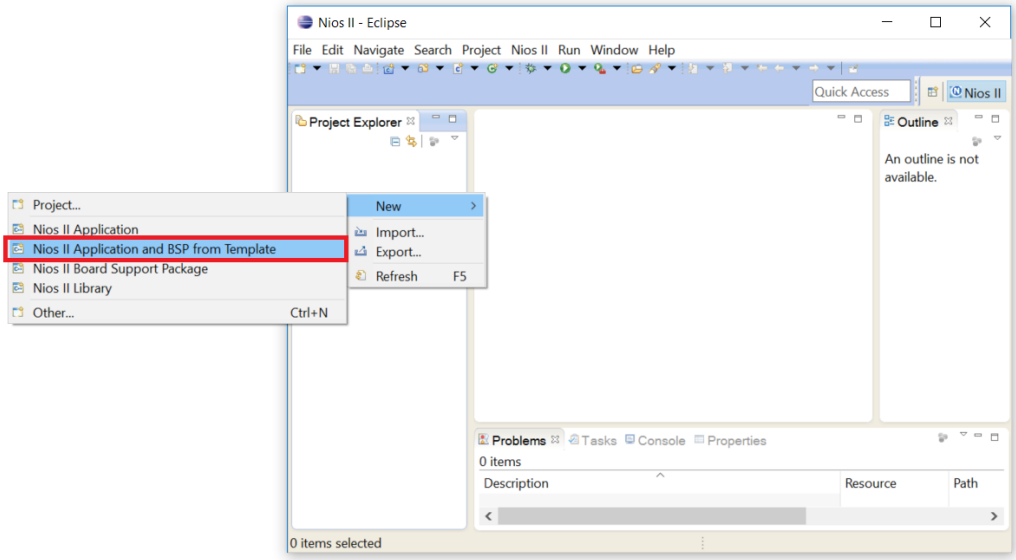

Figure 15: Creating the Initial Project in the Eclipse SBT

☐ Right click in the area called Project Explorer and select **New -> Nios II Application and BSP from Template.**

The BSP is the "Board Support Package" that contains the drivers for things like translating *printf* C commands to the appropriate instructions to write to the terminal.

Next you will see a panel that requests information to setup your design.

- ☐ Navigate to your working directory and click on the **.sopcinfo** file. The **.sopcinfo** file informs Eclipse on what your Platform Designer system contains.
- ☐ Click **OK**.

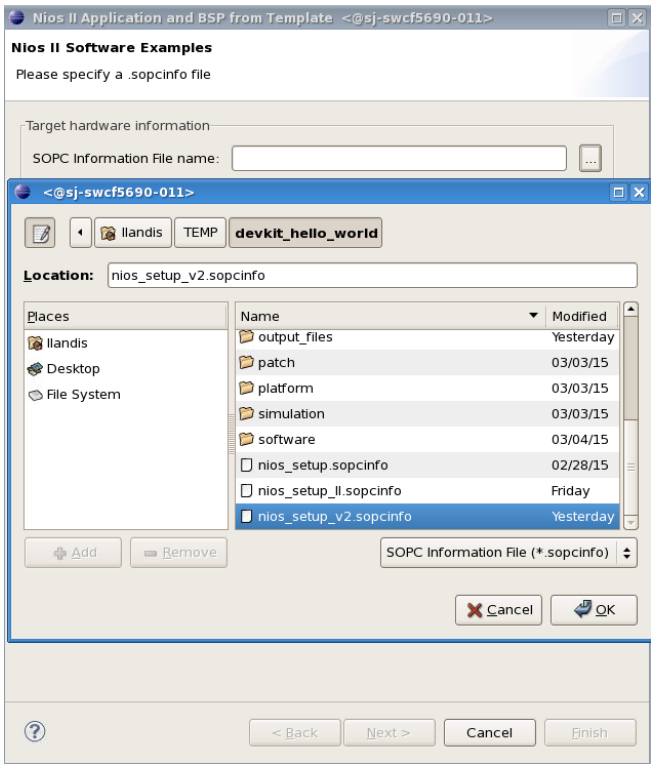

Figure 16: Navigating to the .sopcinfo File

- ☐ Fill in the Project name, call it **hello_world_sw.**
- ☐ Next you will be asked to pick a template design. Select the **Hello World Small** application template. This template writes “Hello from Nios II” to the screen.
    - Make sure to pick Hello World Small and not Hello World or you will not have enough memory in your FPGA design to store the program executable.
- ☐ Click **Finish**.

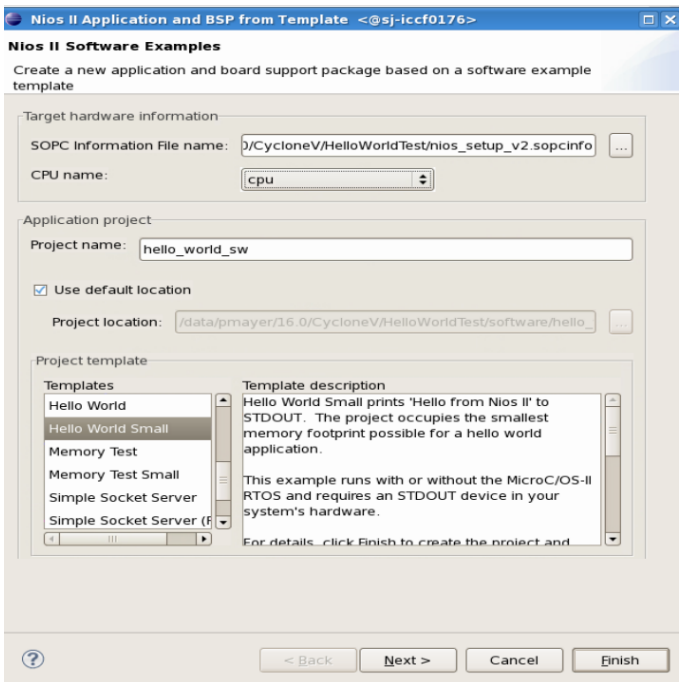

Figure 17: Completing the Nios II Software Examples Setup Screen

☐ Right click on the project **hello_world_sw** and then go to **Run as -> Nios II Hardware.**

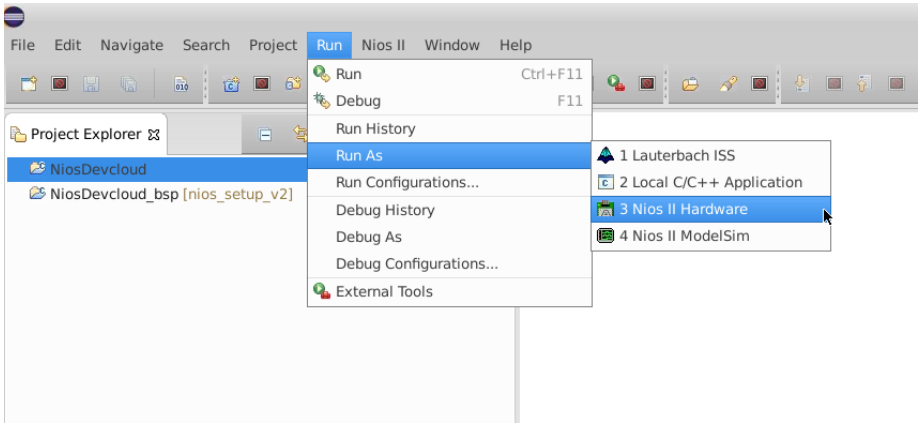

Figure 18: Run NIOS II Hardware

☐ A window will appear as shown below. Click on the **Target Connection** tab.
- The connection should indicate that Eclipse has connected to the USB-blaster.
- If the connection is not identified, you can click **Refresh Connections**
- You might need to stretch the window wider to see the Refresh Connections button.

☐ Click **Run.** If the run button is grayed out but your device shows up under the connections window, you may need to select **Ignore mismatched system ID** and **Ignore mismatched system timestamp.**

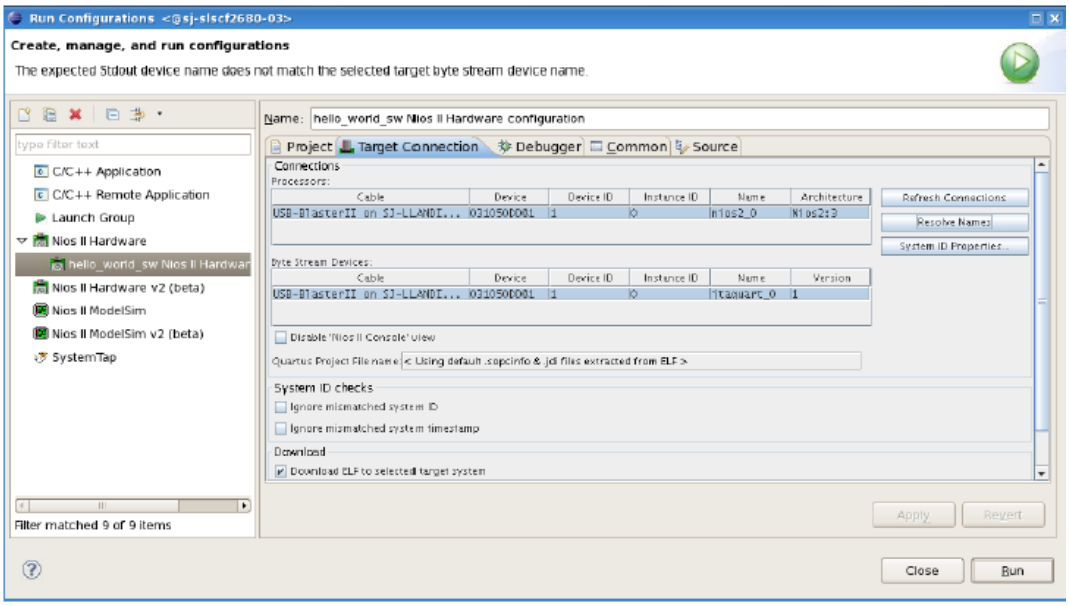

Figure 19: Eclipse SBT Tools after Connection is made to the USB-Blaster

☐ Now you have hardware and software downloaded into your board. You should observe “Hello from Nios II!” printed on the Nios II Console tab.

- If you do not get this printed in the console tab, please refer to the troubleshooting document.

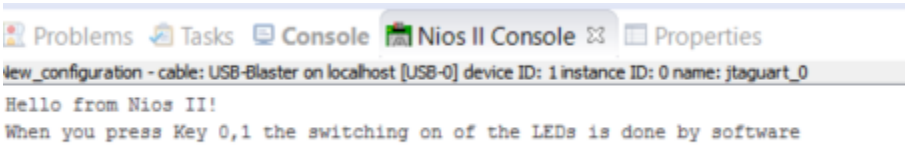

Figure 20: "Hello from Nios II!" on Nios II Console Tab

5 Table of Figures

6 Document Revision History

List the revision history for the application note.

Name	Date	Changes
Rony Schutz	10/28/2019	Initial Release of Quickstart Guide