INTEL DEVCLOUD FOR FPGAS RTL DEVELOPMENT WITH USB BLASTER CONNECTIVITY QUICKSTART GUIDE

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

CONTENTS

INTRODUCTION

Summary

Welcome to the FPGA DevCloud. This is a QuickStart guide that will demonstrate basic project setup and execution in Quartus using the Intel hosted cloud service known as the DevCloud.

At the end of this guide, the user will be able to run and simulate a project in the Quartus Prime Software both **locally** to a development board connected to the user's PC and **remotely** through SSH connection to the DevCloud server connected FPGA. ModelSim, University Waveform, and the In-System Memory Content editor tool will be used to demonstrate successful remote access to the DevCloud servers and provide insight into Quartus FPGA development flows. Flows using the GUI and command line will be demonstrated. The programming image will be downloaded to a development kit connected to the devcloud, and optionally to a local DE10-Lite board should you have one of these kits in your possession.

Assumptions

Assumptions on cloud access: This user guide assumes you have cloud access, a ssh client (MobaXterm or Linux terminal), and a basic understanding of the UNIX operating system, and can use an editor such as vi or emacs. This guide assumes you know what Quartus development tools are and that you can successfully connect to the DevCloud to open the Quartus environment on the X2Go Client application. Expertise on the Quartus toolset is not needed to follow these instructions.

LAB 1: LAUNCHING QUARTUS PRIME SOFTWARE

Summary

This is a short lab that completes the basic project setup using both the Quartus Prime GUI and Tcl Console. At the end of this lab, you will be able to start a new project in the DevCloud using Quartus Prime Software.

Lab Instruction

1.0: Editing and Sourcing Quartus Prime Lite/Pro Software

- ☐ Make sure that you are properly connected to the DevCloud server and have the X2Go windowing system open as shown below. Please refer to the *Public Devcloud Access Instructions* that were sent to you when you first registered for an account on how to connect to the DevCloud.

- ☐ Open a terminal window in the X2Go client by right-clicking on the desktop and selecting "Open Terminal Here."

- ☐ In the terminal, edit the quartus_setup.sh file to ensure that **Quartus Lite** edition is running.

*Note: Launch **Quartus Prime Lite** if the board you are connecting to contains a MAX or Cyclone class device. Otherwise, launch **Quartus Prime Pro** if the board is Arria or Stratix.*

```
#!/bin/bash

# source this file from .profile in Ubuntu home directory
# or.... source this file from .bash_profile in CentOS home directory
# typically this will be added to profile file:  source ~/tools.sh (assuming you
 put this file in your home directory where .profile or .bash_profile are locate
d

# Set to LITE if using Quartus Lite tools: Cyclone and MAX FPGAs. Set to PRO if
using Quartus Pro tools: Arria 10 and Stratix 10
QUARTUS_EDITION="LITE"

# Set OPENCL or HLS variables to true. Set one or the other as issues might aris
e if you set both to TRUE. These only work with the PRO version.
#OPENCL="TRUE"
HLS="TRUE"

# location where all development tools will get installed
TOOLS_ROOT=/glob/development-tools/versions
if [ "$QUARTUS_EDITION" = "LITE" ]; then
   ACDS_ROOT="$TOOLS_ROOT/intelFPGA_lite/18.1"
   MODELSIM_ROOT="$ACDS_ROOT/modelsim_ase/linuxaloem"
else
"quartus_setup.sh" 41L, 1608C                                 9,0-1         Top
```

Figure 1: Correct settings for the quartus_setup.sh file

☐ After editing the quartus_setup.sh file, source the quartus_setup.sh file and run Quartus. Enter the following in the terminal by sequence:

```
source ~/quartus_setup.sh
quartus &
```

☐ Ensure that the main window of the Quartus environment is titled: **Quartus Prime Lite Edition**. If it says **"Quartus Prime Pro Edition,"** you failed to reconfigure the quartus_setup.sh file.

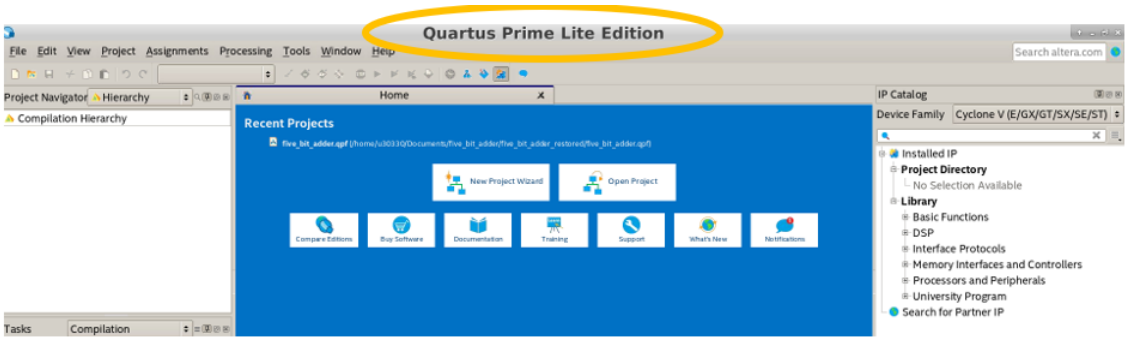

Figure 2: Quartus Prime Lite Edition main window

LAB 2: NEW QUARTUS PRIME DESIGN FROM DEVCLOUD SERVER

Lab Instruction

After Quartus is opened, we need to transfer two Quartus Archive Project Files from our local downloads folder to the DevCloud.

2.0: Downloading .qar Files from University Workshop Page

- ☐ Download the two following .qar files onto your local desktop downloads folder from the Intel DevCloud University WikiPage:
  •five_bit_adder.qar
  •ram_adder.qar
- ☐ Create a source destination folder in your Documents folder using the X2Go terminal named: *quickstart_project*

2.1: Copying Local .qar Files from PC to DevCloud Terminal

To copy local files to your login node (your PC terminal), you cannot be on the head node. In other words, use a local terminal on your PC. You can copy local files to your login node like this:

```
scp /path/to/local/file colfax-intel:/user#/path/to/remote/directory
```

- ☐ Copy the file from your local desktop to the DevCloud server using the following commands and destination path. Revise the source pathway accordingly.

```
scp /home/scabanda/MyDocuments/rtl_quickstart_files/five_bit_adder.qar
colfax-intel:/home/u30330/Documents/quickstart_project

scp /home/scabanda/MyDocuments/rtl_quickstart_files/ram_adder.qar
colfax-intel:/home/u30330/Documents/quickstart_project
```

Ignore the “X11 forwarding request failed on channel 0” message. Look in your destination folder in X2Go to determine if the transfer is complete. Alternatively, use Section 9.1: WinSCP instructions show in the DevCloud Installation Instructions.

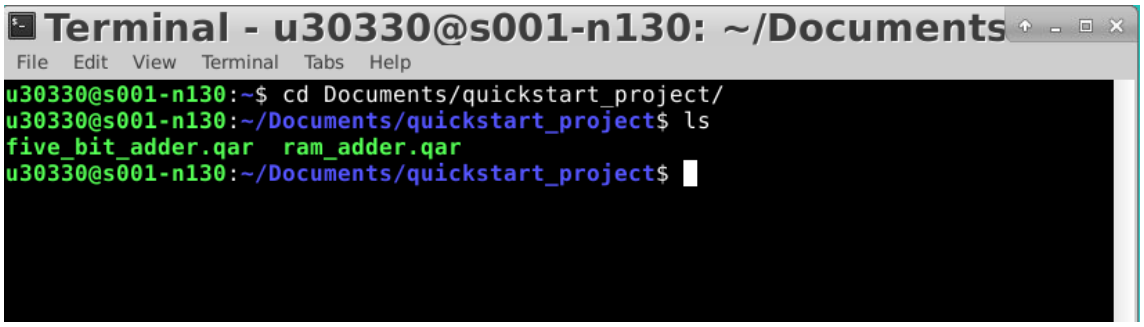

Figure 3: Successful Transfer of File in Project directory in X2Go terminal

2.2: Unarchiving .qar files in Quartus Prime GUI

Now we need to unarchive the .qar files that were just transferred to the DevCloud server.

☐ Under the **File** tab, click **Open Project** or press **Ctrl+J** to open a project.

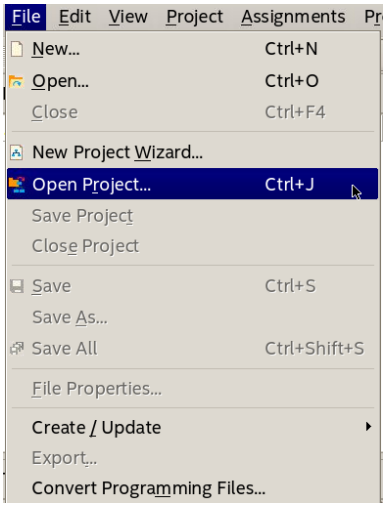

Figure 4: Open Project menu

☐ Locate the **five_bit_adder.qar** file that you transferred into the quickstart_project destination folder. Refer to the pathway that you entered in your local command terminal.

☐ Select the **five_bit_adder.qar** file, and click **Open**.

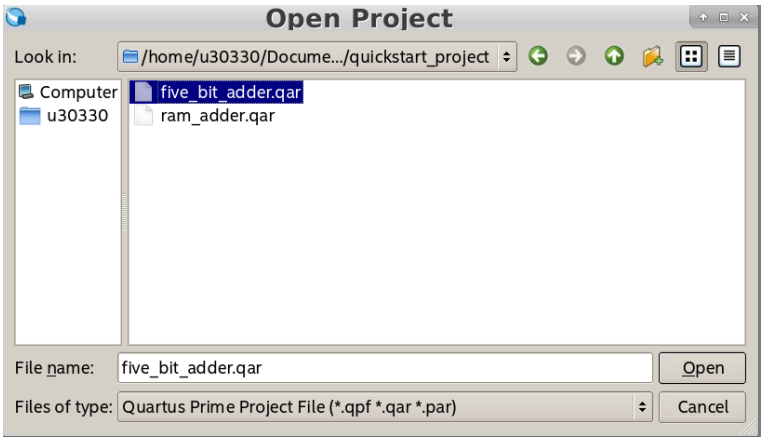

Figure 5: Open Project five_bit_adder.qar file window

The window illustrated in Figure 10 will pop up.

- ☐ Click **OK** and all the source and design files will be saved in the destination folder named five_bit_adder_restored.

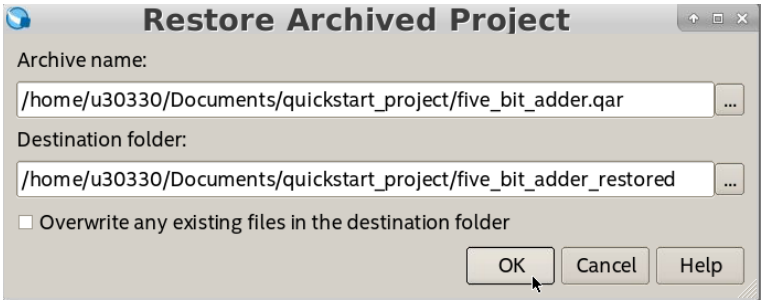

Figure 6: Restore Archived Project for five_bit_adder_restored window

2.3: Unarchiving .qar Files in Command Line

This section describes how to unarchive .qar files using the Quartus Prime Shell.

- ☐ Change your directory to the quickstart_project folder containing five_bit_adder.qar and ram_adder.qar.
- ☐ From a terminal that is logged in to the DevCloud, type and enter the following to restore a project archive in the Quartus Prime Shell:

```
quartus_sh --restore [<options>] <.qar file name>
```

- ☐ Run the following command in the terminal to unarchive ram_adder.qar:

```
quartus_sh --restore -output ram_adder_restored ram_adder.qar
```

The Quartus Prime Shell should state that the job was successfully completed. All source and design files will be saved in the destination folder named ram_adder_restored. For additional information on Quartus Prime Command-Line and Tcl API Help, enter the following into the terminal:

```
quartus_sh --qhelp
```

2.4: Installing the USB Blaster to Download a Design to a Local FPGA

- ☐ To download your completed FPGA design from the DevCloud into a Local PC attached development kit, start by connecting the USB Blaster cable between your PC USB port and the USB Blaster port on your development kit. If you are not using the DE10-Lite, you may have to plug the kit into power using a wall adapter. Upon plugging in your device, you should see flashing LEDs and 7-segment displays counting in hexadecimal, or other lit up LEDs and 7-segments depending on previous projects that have been downloaded to the local development kit.
  NOTE: The lights and switches controlled on the DevCloud connected server kit cannot be controlled unless system console or another form of instrumentation is used.

- ☐ To use the USB Blaster to program your local device, you need to install the USB Blaster driver. To begin, open your Device Manager by hitting the Windows Key  and typing **Device Manager**. Click the appropriate tile labeled Device Manager that appears.

- ☐ Navigate to the **Other Devices** section of the Device Manager and expand the section below.

- ☐ Right click the USB Blaster device and select **Update Driver Software**.

- ☐ Choose to browse your computer for driver software and navigate to the path shown below in Figure 8.

- ☐ Once you have the proper file path selected, click on **Next** and the driver for the USB Blaster should be installed.

2.5: Connecting a Local PC USB Blaster through the DevCloud

- ☐ On your PC, launch the Quartus Programmer. Search **"Programmer"** in the File Explorer:

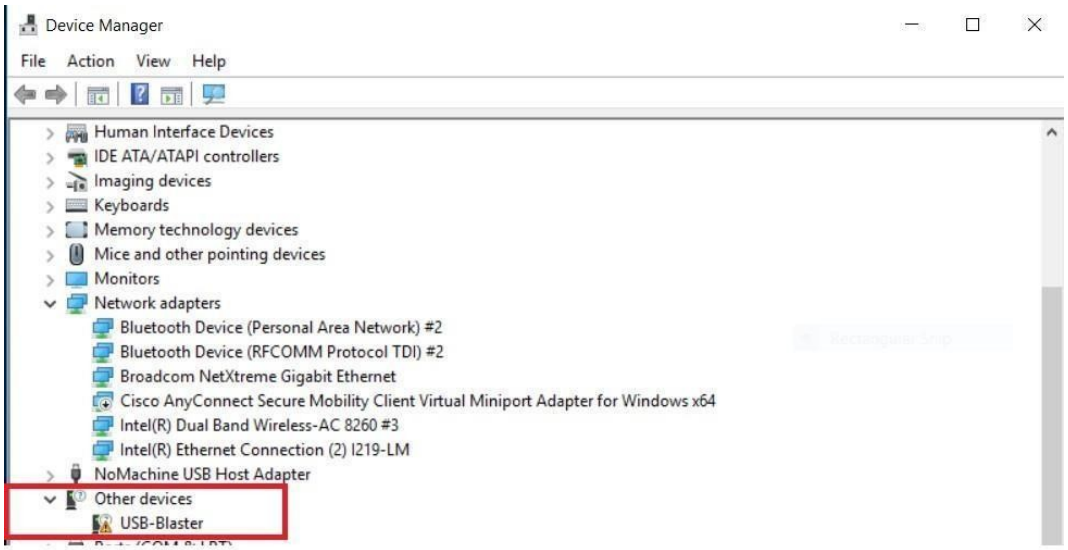

Figure 7: Device Manager with uninstalled USB Blaster driver

If you don't have the Programmer on your PC, download it from this link: http://fpgasoftware.intel.com/18.1/?edition=lite&download_manager=dlm3&platform=windows

☐ Select **Additional Software** and download the **Quartus Prime Programmer** and **Tools**.

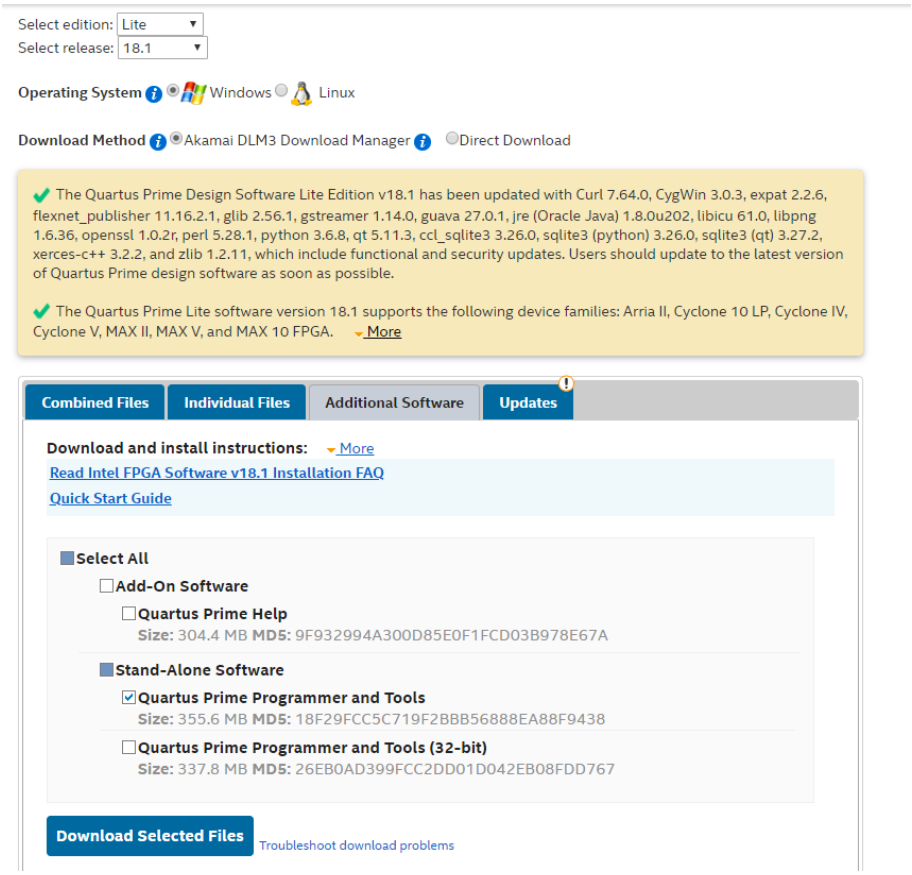

Figure 10: Downloading Quartus Prime Programmer and Tools Package

☐ Follow the login prompts, download, and install the Programmer.

Figure 8: Directory containing USB Blaster drivers

☐ For Intel Employees within the Firewall, in the File Explorer Search window, search "Programmer", and select **Run as administrator**. For other users, you can open the Programmer (Quartus Prime 18.1) normally.

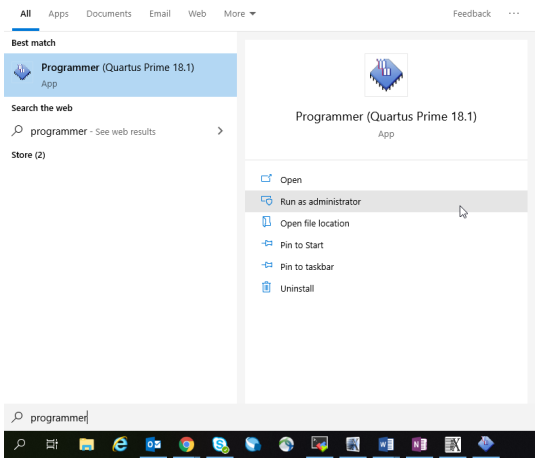

Figure 11: Running Programmer as administrator in Windows

☐ Select *Yes* if a yellow window will pop-up asking if you to allow app changes from an unknown publisher.

The Programmer window should then pop-up.

☐ Left click on **Hardware Setup...** and then select the **JTAG Settings** tab.

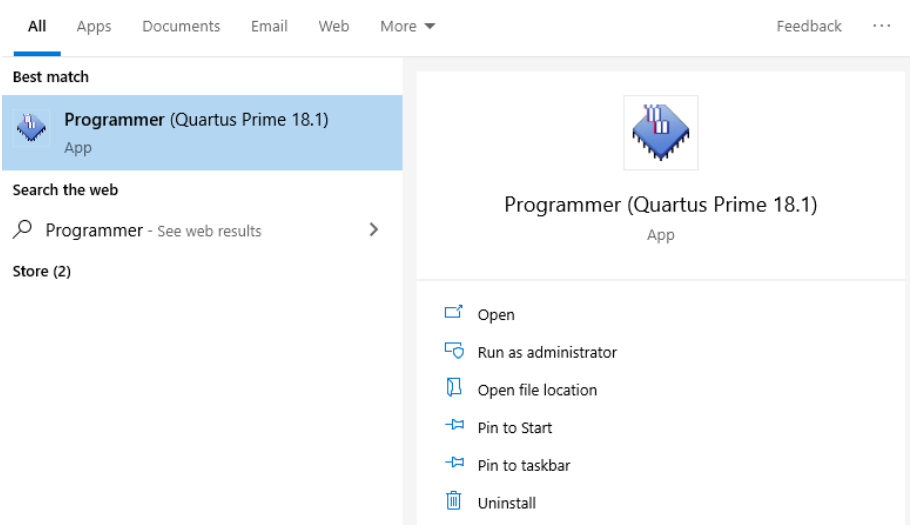

Figure 9: Windows navigation to Programmer (Quartus Prime 18.1)

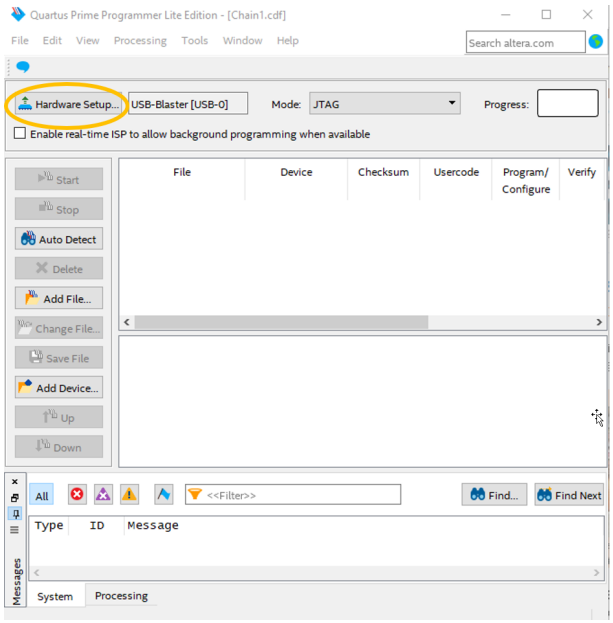

Figure 12: Hardware setup window in Quartus Prime Programmer

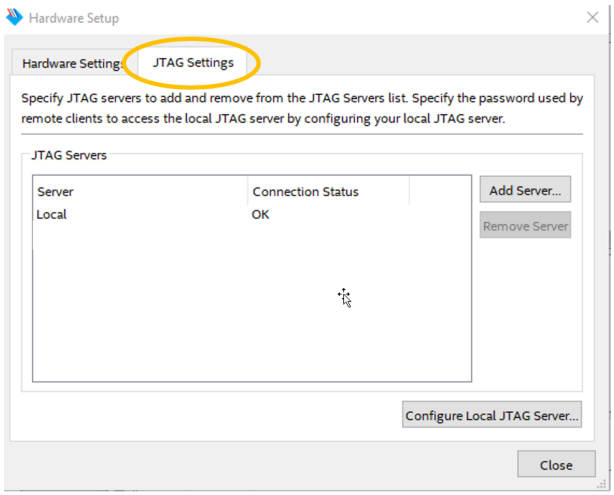

Figure 13: Configuring JTAG Settings

☐ Click on **Configure Local JTAG Server...**.

☐ **Enable remote clients to connect to the local JTAG** server and **enter a password** in the prompt box and **remember this password**. It will be used to connect later.

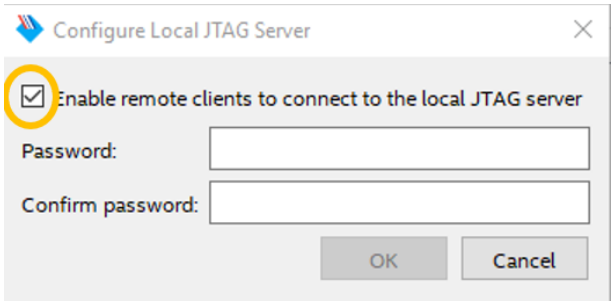

Figure 14: Configuring JTAG Settings

☐ On your local PC terminal, type in the following command to tunnel from the DevCloud to your local USB:

*Note: The last parameter **s001-n138** points to node **138**.*
For server consistency, you need to adjust this number to the node number you are currently using to connect to the DevCloud.

```
    ssh -tR 13090:localhost:1309 colfax-intel ''ssh -t -R 13090:localhost:13090
s001-n138''
```

Ignore the following messages:

```
      stty: standard input: Inappropriate ioctl for device
      X11 forwarding request failed on channel 0
```

☐ On the X2Go app and Quartus Prime Lite window, launch the programmer by selecting **Tools** → **Programmer**.

☐ Left click on **Hardware Setup**, select the **JTAG Settings** tab, and **Add Server**.

☐ Enter in the following information:
Server name: **localhost:13090**
Server password: (password you set up for your PC local JTAG server)

☐ Select **OK**, and you should see the localhost on the list of JTAG Servers.

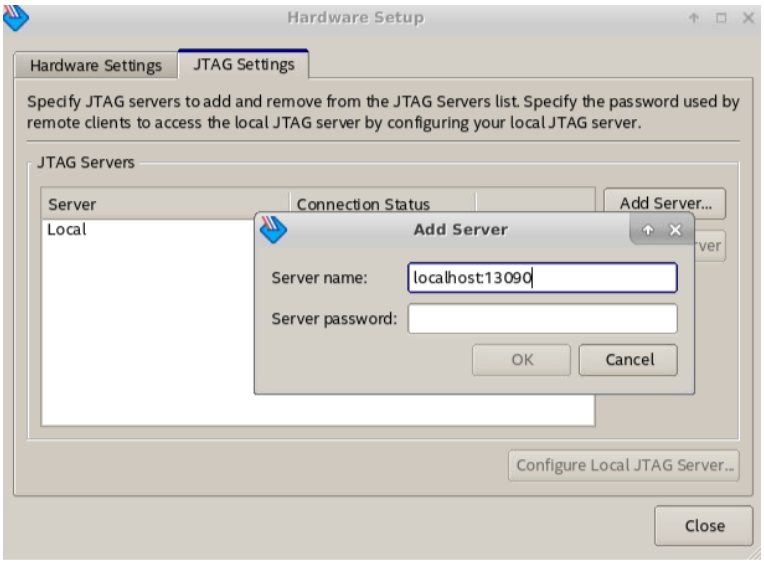

Figure 15: Adding Server for USB Tunneling

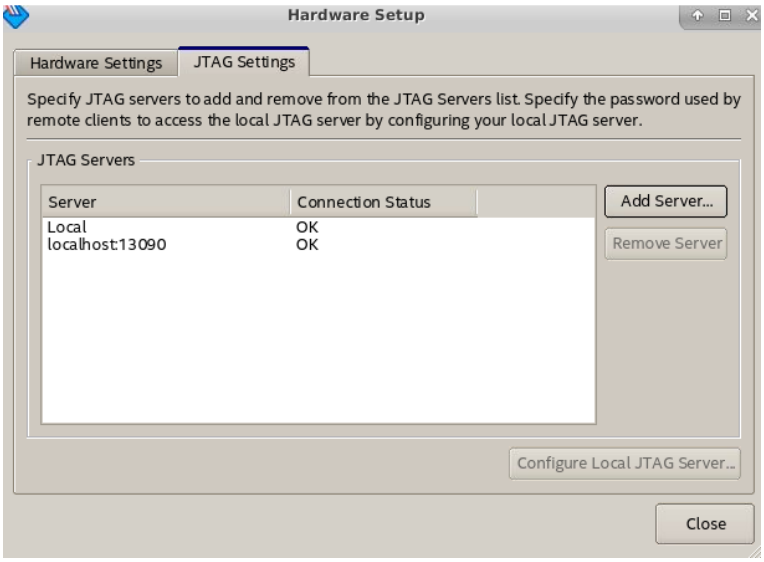

Figure 16: List of JTAG servers

☐ Click on the **Hardware settings** tab, double click on the **localhost:13090**, and that should now be your selected USB blaster download connection.

Make sure localhost:13090 shows up as your currently selected hardware and that the connection status is OK.

2.6: Programming a Design into a Local PC Connected FPGA

- ☐ Select the programming file to be downloaded to the FPGA.
- ☐ Click on **Add File**, **output files** folder, and select the **five_bit_adder.sof** file.
- ☐ Click **OK** and select **Start**. The progress bar should show 100% (Successful) and turn green. If it fails the first time, click **Start** a second time.

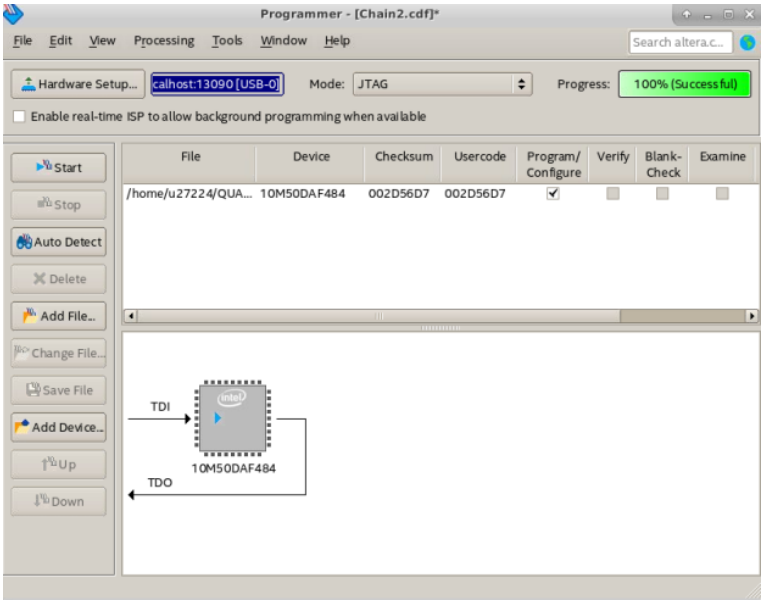

Figure 17: Programming a Design into a Local PC Connected FPGA

2.7: Testing the Design on the Local PC Connected FPGA

The picture below illustrates how the hex displays and switches interact as a function of a five-bit adder.

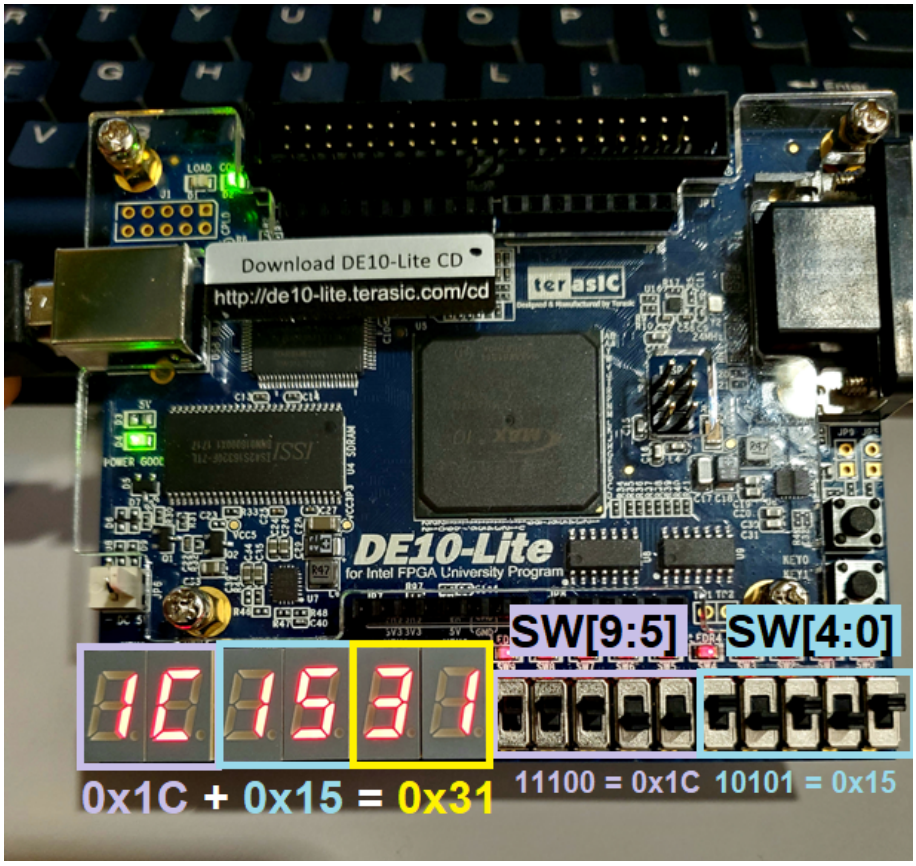

Figure 18: Testing the Five-bit Adder Design on a Local PC Connected FPGA

.

2.8: Programming a Design into the DevCloud Hosted Server FPGA

We can also upload our design to the Local PC connected FPGA, however, we will not be able to interact with it like we did with the Local FPGA board.
To upload the design to the DevCloud Hosted Server FPGA:

- ☐ Click **Hardware Setup...** and then double click the **Local USB-Blaster** at the top of the available hardware list.
  Close the Hardware Setup window.
- ☐ Click on **Add File**, **output files** folder, and select the **five_bit_adder.sof** file.
- ☐ Click **OK** and select **Start**. The progress bar should show 100% (Successful) and turn green. If it fails the first time, click **Start** a second time.

See section 3.2 to learn how to interact with the DevCloud Hosted FPGA remotely using the In-System Memory content editor.

LAB 3: QUARTUS PRIME SIMULATIONS ON DEVCLOUD SERVER FPGA

Summary

This section will step you through the process of simple simulations by synthesizing and compiling pre-designed ModelSim and University Waveform. At the end of this lab, you will also be able to test the functionality of a design downloaded to the DevCloud Hosted FPGA board using a tool known as the In-System Memory Content editor.

3.0: Running ModelSim on the DevCloud

This section will walk you through how to run a ModelSim Simulation for the five-bit adder module.

- ☐ Re-open the Quartus Prime environment and open the **testbench.v** file.
- ☐ You can do this by changing the Project Navigator drop-down menu from **Hierarchy** to **Files** and double clicking on the **testbench.v** file.

Now we need to compile the project.

3.1: Compiling a Testbench in the Quartus Prime Environment

- ☐ Set the testbench.v file as Top-Level Entity. Right click on the **testbench.v** file in the Project Navigator window and select **Set as Top-Level Entity**.
- ☐ Click on the **Analysis & Elaboration** icon  in the Quartus Toolbar. Allow the system to fully compile.

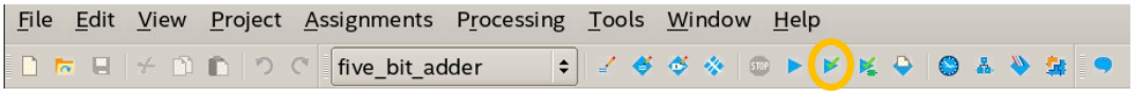

Figure 19: Analysis & Elaboration icon in the Quartus Toolbar

Launch the vsim.wlf file generated from Analysis and Elaboration.

- ☐ Inside the five_bit_adder_restored project folder, enter the following in the terminal:

```
vsim
```

*Note: Some nodes, such as 130, did not install ModelSim correctly. Try using nodes **137-139** if an error is thrown and the vim GUI does not open.*
Please refer to the Public Devcloud Access Instructions that were sent to you when you first registered for an account on how to connect to different nodes in the DevCloud.
The ModelSim window named vsim should pop-up.

☐ In the lower transcript panel, enter the following:

```
do five_bit_adder_run_msim_rtl_verilog.do
vsim work.testbench
do setup.do
```

You should be able to see the following:

Figure 20: ModelSim Waveforms Simulation Results

Congrats! You have finished this RTL DevCloud QuickStart guide.

APPENDIX

List of Figures

List of Tables

Revision History

DATE	NAME	DESCRIPTION
09/26/2019	S. Cabanday	Initial Release
10/09/2019	S. Cabanday	Re-formatting and information reducation

Table 1: Revision Control History