ECE275: Sequential Logic Circuits
Lab 1: Install Quartus and Program FPGA

Pascal Francis-Mezger

August 27, 2021

Contents

1 Lab Overview

The goal of this lab will be to install the Intel Quartus software, and then utilize it to program the provided Altera DE0 development FPGAs with an example Verilog program.

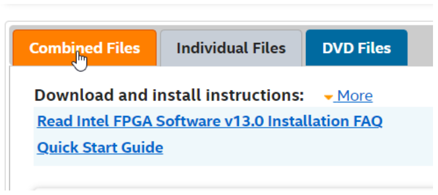
The installation can be somewhat finicky, so make sure to perform all of the steps and get the correct software version.

While there is a Linux version of Quartus, it has not been tested for this class, so use that at your own risk. There is no native Mac OS support.

2 Installing Quartus

We will be utilizing the Quartus II 13.0SP1 software for this class. This version has been tested to be compatible with DE0 FPGA boards. You can download it from here: `https://fpgasoftware.intel.com/13.0sp1/?edition=web`

You will need to select the combined tab to download the combined software so you will not have to download device support for the FPGA separately. Make sure you are downloading the 13.0sp1 Web Edition. You will need to create a free account to download the software.

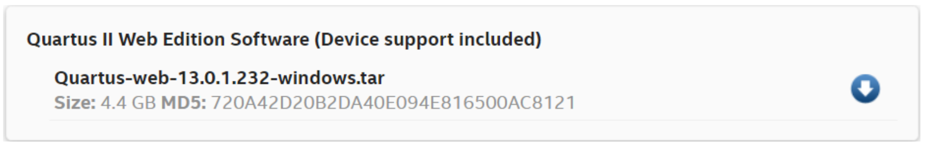


Download and install instructions:

Read Intel FPGA Software v13.0 Installation FAQ

Quick Start Guide

Figure 1: Select the Combined Installation



Quartus II Web Edition Software (Device support included)

Quartus-web-13.0.1.232-windows.tar

Size: 4.4 GB **MD5:** 720A42D20B2DA40E094E816500AC8121

Figure 2: Quartus Version to Download

Important Note

The Intel website is very finicky, and may try to change the version you have selected after you login, or if you click around on the page. Double check your selection before downloading.

Once the download is finished, install the Quartus II software, using the default options on each page. Take note of your installation path, you will need it for the next step.

3 Setup the USB Blaster Driver

Instructions from `http://www.terasic.com.tw/wiki/Altera_USB_Blaster_Driver_Installation_Instructions`

For Windows 10:

1. Use the provided USB cable to connect the DE0 board to your computer
2. Open your control panel, and then go to devices and printers
3. Under Unspecified, USB Blaster should be listed. Right click on this and then select Properties
4. Select the Hardware tab and select Properties
5. A new window should pop up with the General tab already selected. Select Change Settings.
6. Again a new window should pop up with the General tab already selected. Select Update.
7. Select Browse my computer for driver software.
8. Find <Path to Quartus II installation>\quartus\drivers\ Check notes below for additional path information
9. Select OK. Make sure the proper path was selected then select Next.
10. If the Windows security window pops up check the Always trust software from "Altera Corporation"box and select Install.

- Note 1: Your Altera file is located at the location you selected when you first installed Quartus. The location listed in this document is the default location) C:\altera\13.0sp1

- Note 2: Stop at the drivers folder, i.e., do NOT go deeper by opening a folder within the drivers folder)

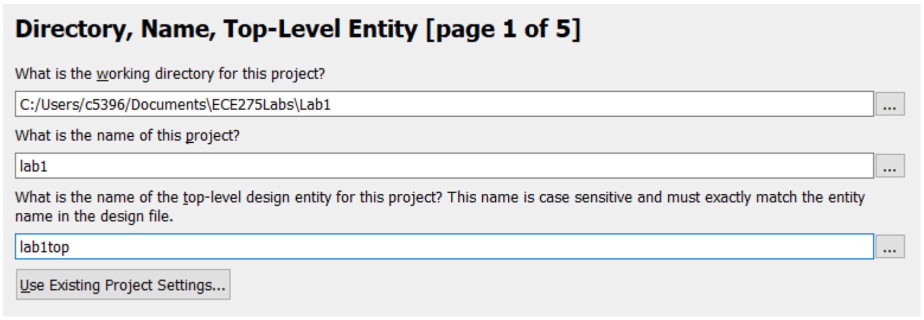
4 Creating a New Project with the Quartus Software

The Quartus software should be in your start menu programs list in a folder named Altera 13.0.1.232 Web Edition. Run the Quartus II 13.0sp1 software in this folder.

After the software is started, you will need to create a new project. You can do this by selecting "Create a New Project" in the initial pop up window, or by going to the file menu at the top left, going to new, and then selecting the new project option.

In the project setup wizard, you will need to set up the project name hierarchy, and select the correct FPGA model.

The first window should ask for the project working directory, project name, and top-level name. Try to use underscores instead of spaces in the project name and path, and avoid use of any special characters. The project save path will be a folder that contains multiple files that relate to the project, so it is a good idea to create a unique folder in a larger folder structure for each lab section. For example if a lab had multiple parts with unique code, the path could be <Path to Your Documents Folder>\ECE275Labs\Lab1\Lab1Part1 for the first part, and then <Path to Your Documents Folder>\ECE275Labs\Lab1\Lab1Part2 for the second part. The next box asks for the name of the project. Give it a short relevant name. The last box asks for the top level, generally an easy name to use is the project name with top added at the end (ex. Project: lab2part1, Top Level: lab2part1top). This makes it easy to keep track of your top level module name (you will use the top level name similar to how main is used for the main function in C).



Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

C:/Users/c5396/Documents\ECE275Labs\Lab1

What is the name of this project?

lab1

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

lab1top

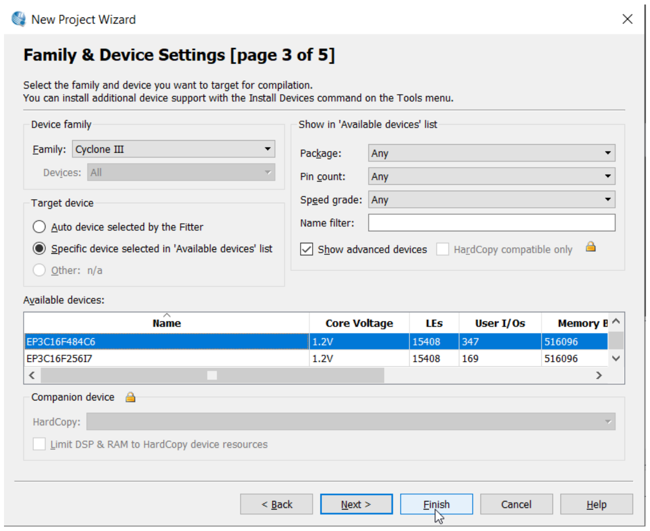
Figure 3: Set up the project naming hierarchy

The next step will ask you to add files, you can just hit next here.

Next you will need to select the correct FPGA model. You can use the family dropdown to select Cyclone III. Then, in the available devices window, find the model name:

```
FPGA Model Name:  EP3C16F484C6
```

After selecting the correct model name, you can hit Finish on the New Project Wizard.



New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

Device family

Family: Cyclone III

Devices: All

Target device

( ) Auto device selected by the Fitter
(●) Specific device selected in 'Available devices' list
( ) Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Speed grade: Any

Name filter:

[✓] Show advanced devices [ ] HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory B
EP3C16F484C6	1.2V	15408	347	516096
EP3C16F256I7	1.2V	15408	169	516096

Companion device

HardCopy:

[ ] Limit DSP & RAM to HardCopy device resources

Figure 4: Select the correct FPGA model, EP3C16F484C6

4.1 Create a New Verilog File and Copy the Code

In future labs, you will be writing your own code in Verilog, but for today's lab it is being provided for you.

To create a new Verilog file, go to the file menu in the top left, go to new, and then in the Window that pops up, select Verilog HDL File

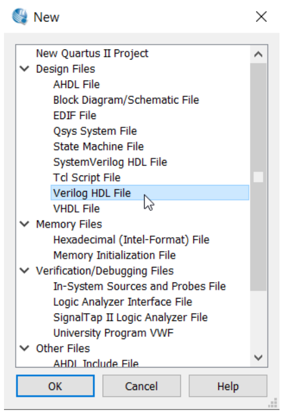


Figure 5: Create a new Verilog HDL File

This will create a new blank file, where you can copy and paste the following code. You will need to change the module name at the top to the name you chose for your top level in the wizard.

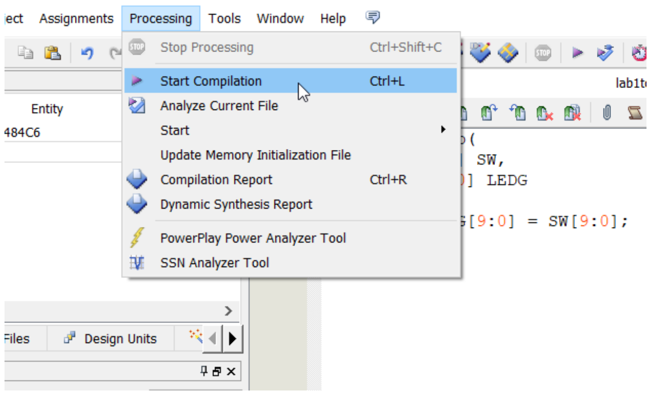
```
module lab1top(
    input [9:0] SW,
    output [9:0] LEDG
);
    assign LEDG[9:0] = SW[9:0];
endmodule
```

After pasting the code, hit control+s to save the file. You can leave the file name as your top-level name.

4.2 Run Compilation

The next step is to run compilation. You will have to run this before downloading to the FPGA anytime you make any code updates, or pin assignment changes. The compilation creates the files that are downloaded to the FPGA from your project files.

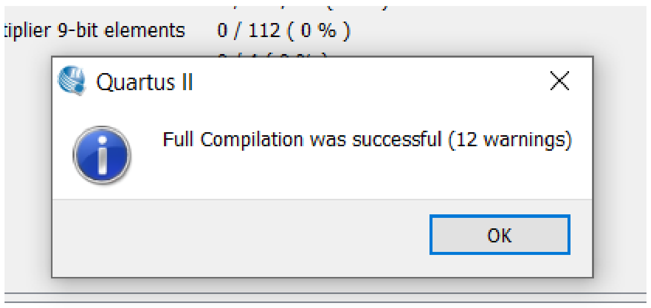
You can run compilation by hitting the purple play icon in the center top of the toolbar, or by going to the processing menu at the top, and selecting Start Compilation.



```
 SW,
] LEDG
[9:0] = SW[9:0];
```

Figure 6: Start Project Compilation

When the compilation finished, a box should pop up indicating there were 12 warnings. This is ok, as long as you do not have any errors. Errors are higher level than warnings, and would indicate you did not copy something correctly with the code, or you forgot to change the module name to your top-level name.



tiplier 9-bit elements 0 / 112 (0 %)

Quartus II

Full Compilation was successful (12 warnings)

Figure 7: Compilation was successful, even with warnings

5 Set Pin Assignments

The last step before downloading to the FPGA is to set the pin assignments. This essentially references pins on the FPGA to their real world device connection. The pins are laid out in a grid on the FPGA, so are referenced with names such as AA12. This step will associate the switches and LED's that are utilized in the Verilog program to pins they are physically connected to on the FPGA.

The easiest way to set pin assignments is to use their unique names as the variable names in the verilog code, and then use pre-made qsf files to create the references. The .qsf file for the board in this class will provided to you. For this lab, you only need a few lines from the qsf file, and those are provided below:

```
set_location_assignment PIN_B1 -to LEDG[9]
set_location_assignment PIN_B2 -to LEDG[8]
set_location_assignment PIN_C2 -to LEDG[7]
set_location_assignment PIN_C1 -to LEDG[6]
set_location_assignment PIN_E1 -to LEDG[5]
set_location_assignment PIN_F2 -to LEDG[4]
set_location_assignment PIN_H1 -to LEDG[3]
set_location_assignment PIN_J3 -to LEDG[2]
set_location_assignment PIN_J2 -to LEDG[1]
set_location_assignment PIN_J1 -to LEDG[0]
set_location_assignment PIN_D2 -to SW[9]
set_location_assignment PIN_E4 -to SW[8]
set_location_assignment PIN_E3 -to SW[7]
set_location_assignment PIN_H7 -to SW[6]
set_location_assignment PIN_J7 -to SW[5]
set_location_assignment PIN_G5 -to SW[4]
set_location_assignment PIN_G4 -to SW[3]
set_location_assignment PIN_H6 -to SW[2]
set_location_assignment PIN_H5 -to SW[1]
set_location_assignment PIN_J6 -to SW[0]
```

You will need to copy and paste these lines into the Quartus Console for your project, and then hit enter to run them. The Quartus console can be found below your verilog file in the Quartus window. If it is not showing, you may need to go to the view file menu, go to Utility Windows, and check if the Tcl Console is activated.

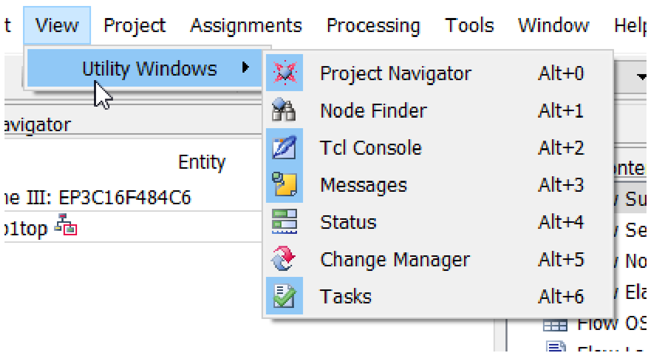
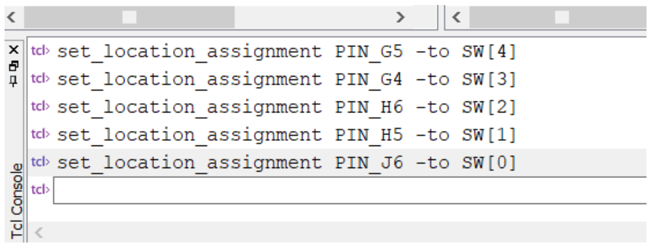


Figure 8: Check if the Tcl Console is Activated



```
set_location_assignment PIN_G5 -to SW[4]
set_location_assignment PIN_G4 -to SW[3]
set_location_assignment PIN_H6 -to SW[2]
set_location_assignment PIN_H5 -to SW[1]
set_location_assignment PIN_J6 -to SW[0]
```

Figure 9: Tcl Console After Running Pin Assignments

You can check if the pin assignments were correctly run by opening the pin planner. This can be found in the assignments menu at the top. Your pins should match the pin layout shown in the picture.

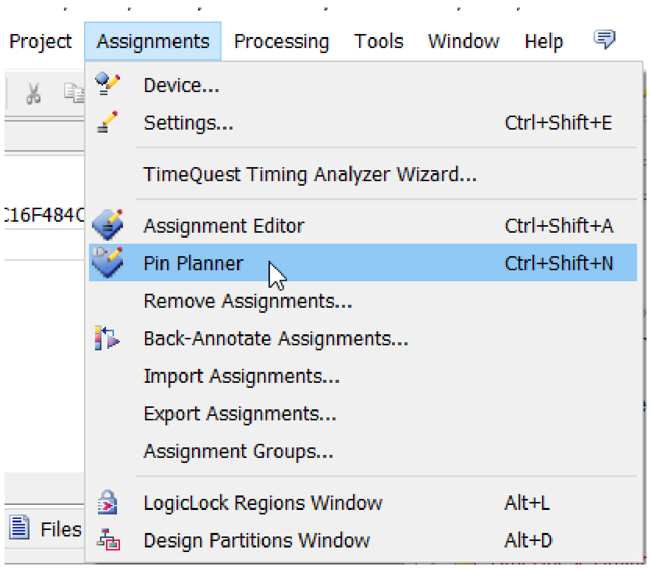
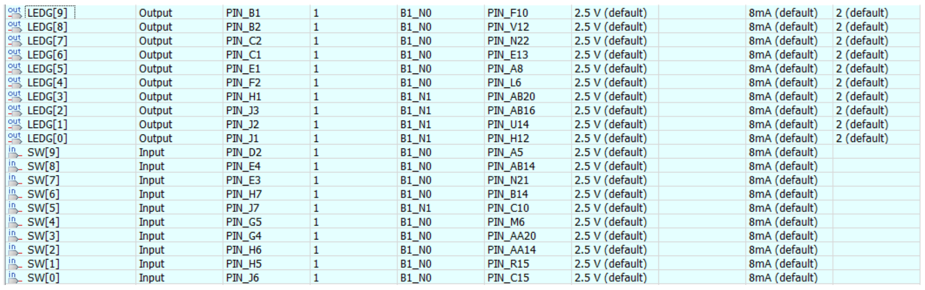


Figure 10: Open the Pin Planner



LEDG[9]	Output	PIN_B1	1	B1_N0	PIN_F10	2.5 V (default)		8mA (default)	2 (default)
LEDG[8]	Output	PIN_B2	1	B1_N0	PIN_V12	2.5 V (default)		8mA (default)	2 (default)
LEDG[7]	Output	PIN_C2	1	B1_N0	PIN_N22	2.5 V (default)		8mA (default)	2 (default)
LEDG[6]	Output	PIN_C1	1	B1_N0	PIN_E13	2.5 V (default)		8mA (default)	2 (default)
LEDG[5]	Output	PIN_E1	1	B1_N0	PIN_A8	2.5 V (default)		8mA (default)	2 (default)
LEDG[4]	Output	PIN_F2	1	B1_N0	PIN_L6	2.5 V (default)		8mA (default)	2 (default)
LEDG[3]	Output	PIN_H1	1	B1_N1	PIN_AB20	2.5 V (default)		8mA (default)	2 (default)
LEDG[2]	Output	PIN_J3	1	B1_N1	PIN_AB16	2.5 V (default)		8mA (default)	2 (default)
LEDG[1]	Output	PIN_J2	1	B1_N1	PIN_U14	2.5 V (default)		8mA (default)	2 (default)
LEDG[0]	Output	PIN_J1	1	B1_N1	PIN_H12	2.5 V (default)		8mA (default)	2 (default)
SW[9]	Input	PIN_D2	1	B1_N0	PIN_A5	2.5 V (default)		8mA (default)	
SW[8]	Input	PIN_E4	1	B1_N0	PIN_AB14	2.5 V (default)		8mA (default)	
SW[7]	Input	PIN_E3	1	B1_N0	PIN_N21	2.5 V (default)		8mA (default)	
SW[6]	Input	PIN_H7	1	B1_N0	PIN_B14	2.5 V (default)		8mA (default)	
SW[5]	Input	PIN_J7	1	B1_N1	PIN_C10	2.5 V (default)		8mA (default)	
SW[4]	Input	PIN_G5	1	B1_N0	PIN_M6	2.5 V (default)		8mA (default)	
SW[3]	Input	PIN_G4	1	B1_N0	PIN_AA20	2.5 V (default)		8mA (default)	
SW[2]	Input	PIN_H6	1	B1_N0	PIN_AA14	2.5 V (default)		8mA (default)	
SW[1]	Input	PIN_H5	1	B1_N0	PIN_R15	2.5 V (default)		8mA (default)	
SW[0]	Input	PIN_J6	1	B1_N0	PIN_C15	2.5 V (default)		8mA (default)	

Figure 11: Pin layout in Quartus Pin Planner

6 Rerun Compilation and Program the FPGA

Make sure to first rerun the compilation using the purple play icon or start compilation in the processing menu. After that is finished, go to the tools menu, and select the programmer option.

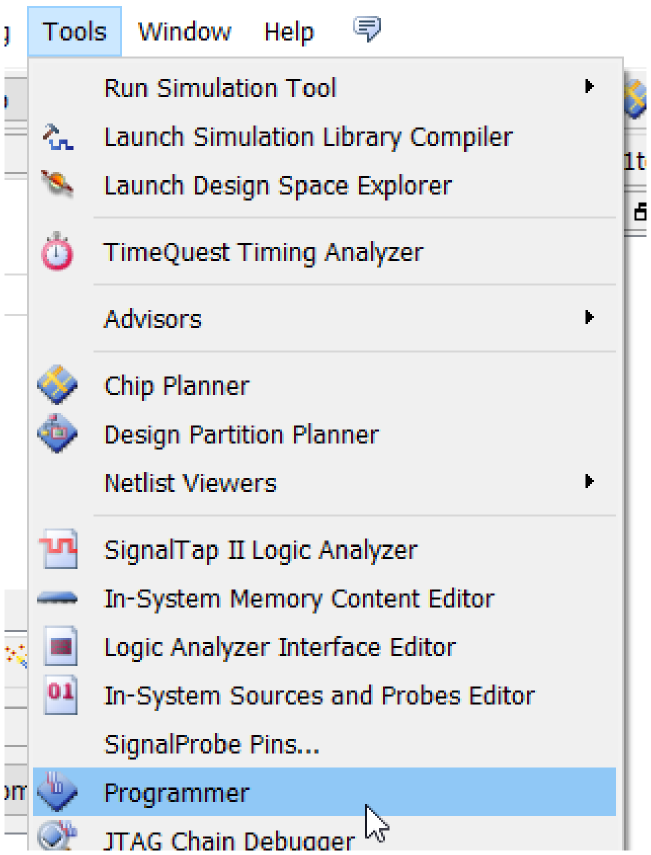
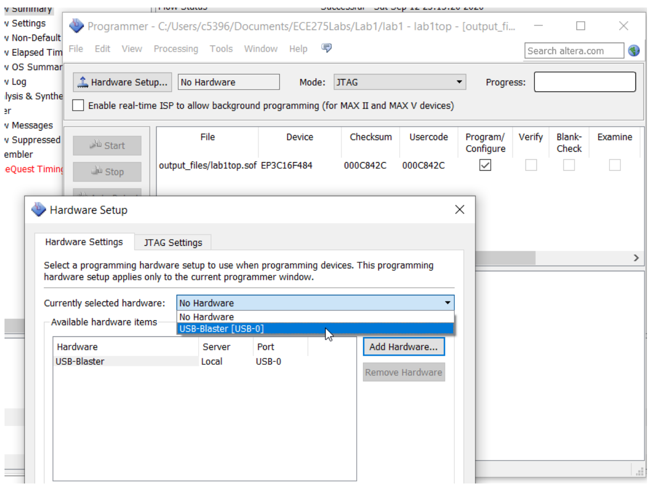

Tools
Window
Help
Run Simulation Tool
Launch Simulation Library Compiler
Launch Design Space Explorer
TimeQuest Timing Analyzer
Advisors
Chip Planner
Design Partition Planner
Netlist Viewers
SignalTap II Logic Analyzer
In-System Memory Content Editor
Logic Analyzer Interface Editor
In-System Sources and Probes Editor
SignalProbe Pins...
Programmer
JTAG Chain Debugger

Figure 12: Open the Quartus Programmer

In the programmer window, the is an option for hardware setup towards the top left. Click on it to open the hardware setup window. Here you will have to select the USB Blaster option. If that is not available, make sure you properly completed the step of installing the USB-Blaster driver, and that the FPGA is on and connected to your computer through USB. Leave the JTAG programming mode selected in the main programmer window.


Programmer - C:/Users/c5396/Documents/ECE275Labs/Lab1/lab1 - lab1top - [output_fi...

Hardware Setup... No Hardware Mode: JTAG Progress:

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
output_files/lab1top.sof	EP3C16F484	000C842C	000C842C	✓	☐	☐	☐

Hardware Setup

Hardware Settings | JTAG Settings

Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: No Hardware
- No Hardware
- USB-Blaster [USB-0]

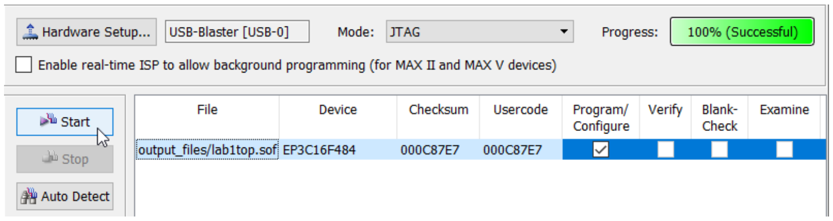
Available hardware items

Hardware	Server	Port
USB-Blaster	Local	USB-0

Add Hardware... Remove Hardware

Figure 13: Select the USB-Blaster

At this point, you should be able to program your FPGA by selecting the output files .sof file in the programmer window, and then hit Start to program the FPGA. There is a confusing switch on the FPGA labeled RUN and PROG next to the 7-Segment LEDs. This must be in the RUN position to program the board when using JTAG. If the programmer says programming failed, check to see if that is in the RUN position.


Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress: 100% (Successful)

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

Start Stop Auto Detect

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
output_files/lab1top.sof	EP3C16F484	000C87E7	000C87E7	✓	☐	☐	☐

Figure 14: Program the FPGA

If your programmer window does not show any file in the middle to select, you will have to add your compiled project file. You can do this by selecting Add File, and then in the window the pops up double click on the output files folder, and then double clicking on your project's .sof file inside.

7 Completing the Lab

After programming the FPGA, you should see that if you change the position of any of switches on the FPGA (SW0-SW9) it should illuminate or turn off the corresponding green LED. Please review the Verilog code so you can understand why this is accomplished. After this is working correctly, the lab instructor or TA will ask you the three questions in the questions section at the end of the lab. The answers to these questions are in the lab writeup, be prepared to answer them.

8 Questions

8.1 Question 1

When do you have to run or rerun compilation for a Quartus project?

8.2 Question 2

After copying commands from a .qsf file to assign FPGA pins to the correct real world devices, where would you navigate in Quartus to check if the pins were assigned properly?

8.3 Question 3

What position should the RUN/PROG switch be in on the DE0 boards when programming with JTAG?