CEE-445 Embedded System Design

FPGA Lab#1:
Introduction to Intel Quartus Prime Software and MAX 10 FPGA with DE10-Lite board

Getting Started with Quartus Prime Software for FPGA Design:
review for the traffic light controller example

Lab Goal

This lab is the beginning of the lab experiments to create a Bluetooth connected wheeled robot using an Intel FPGA, Android app, bluetooth module, and servo motors. The wheeled robot will be controlled with an Android app for navigation. In this lab, we will first introduce the DE10 board and get familiar with the feature rich Intel MAX 10 FPGA. Quartus Software will be the programming platform for the Intel FPGA. We will start with a simple design with a traffic light controller. At the end of this lab, you will have a traffic light controller running on the DE10 FPGA board.

Introduction to Quartus software:

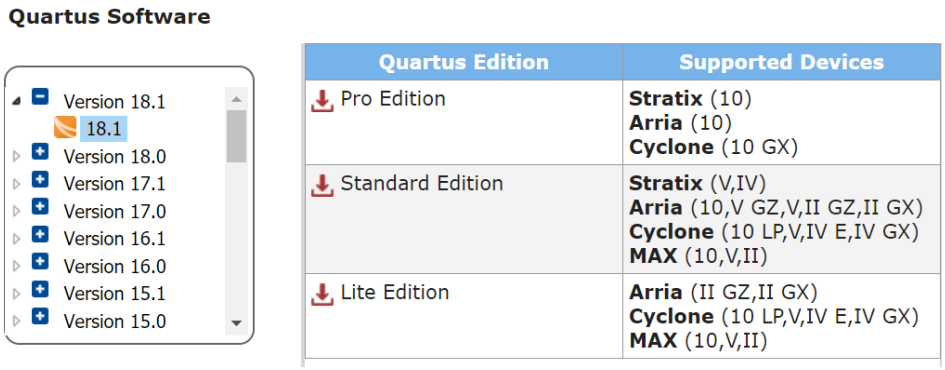
Quartus by Altera is a Programming Logic Device (PLD) Design Software that is suitable for high-density Field- Programmable Gate Array (FPGA) designs, low-cost FPGA designs, and Complex Programmable Logic Devices CPLD designs. As most commercial CAD tools are continuously being improved and updated, Quartus software has gone through a number of releases. The version known as Quartus Lite V18.1 is used in this tutorial.

The tutorial is organized as follows. First section gives pointers to Intel's website from where this software can be downloaded and instructions to install this software. Second section describes a step-by-step approach for designing a traffic light controller using MAX10:10M50DAF484C6GES FPGA on the DE10 board.

Download the most current Quartus II software:

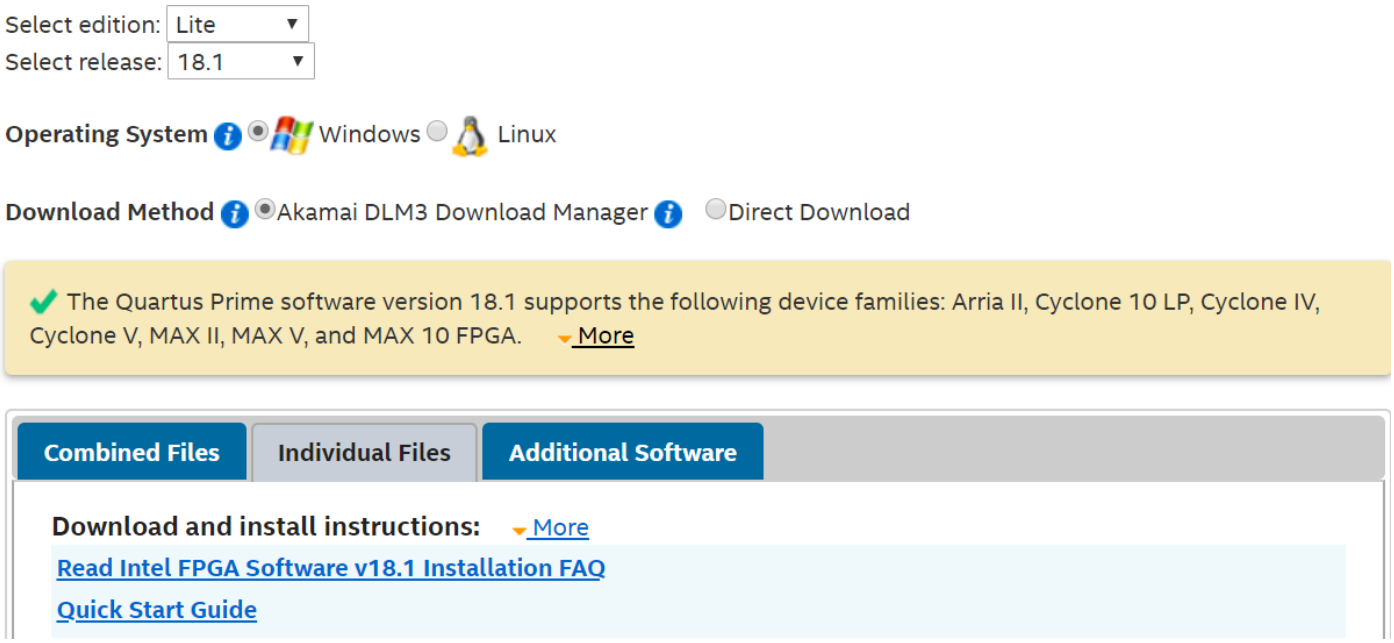
You will be asked to create an account with Intel. You will also be asked to enter some personal information to complete the registration process before you can download the Altera software.

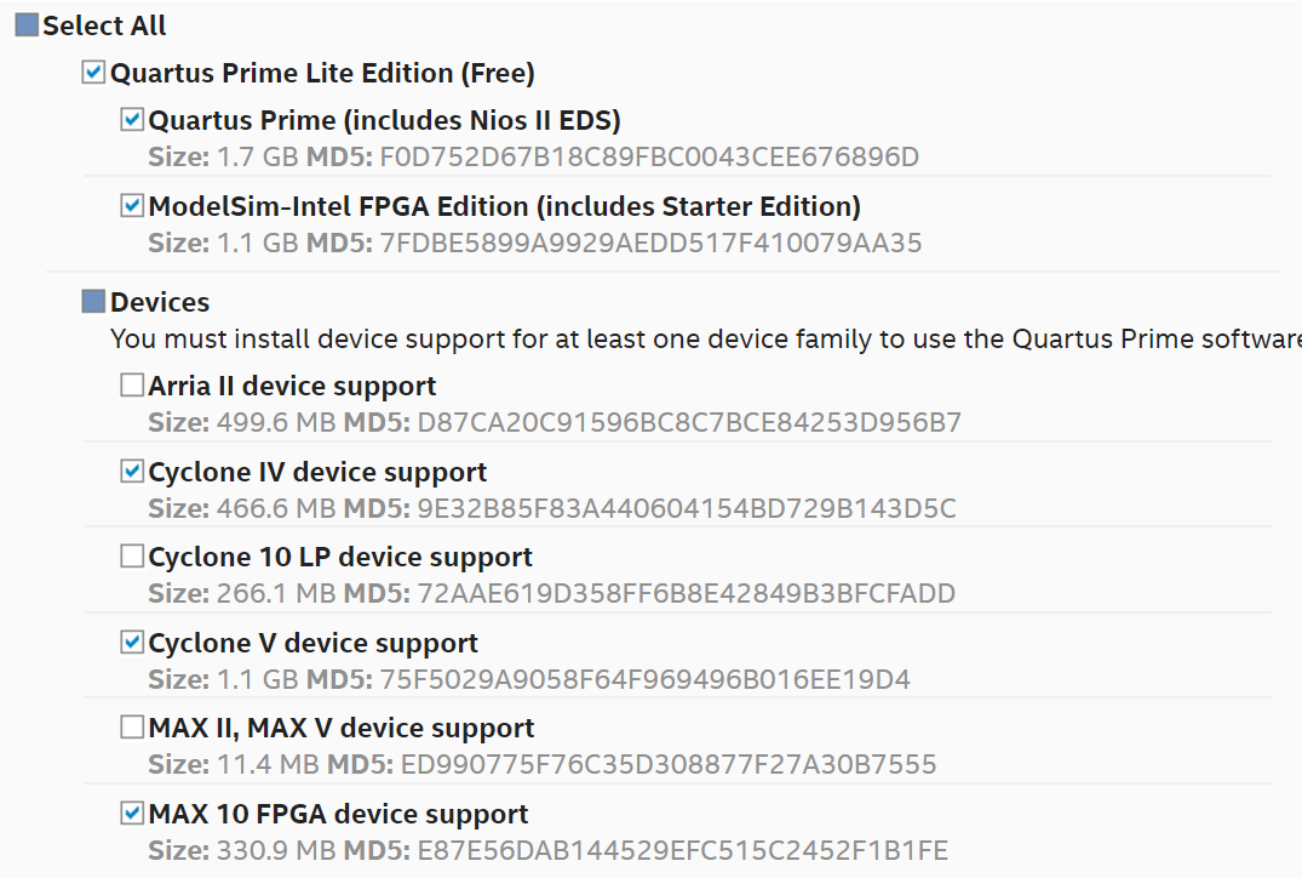
Quartus Prime Lite Edition (v18.1)



Install the latest version of the software:

- Go to the following URL: https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html
- Select the Quartus Lite Edition or a new version and include the following packages: modelsim-Altera Starter edition, and devices (MAX 10) support.

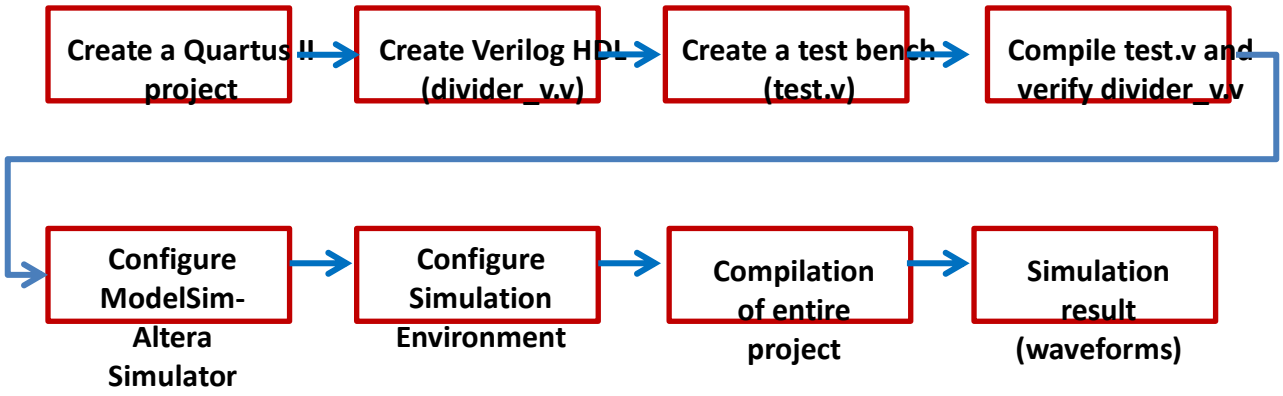




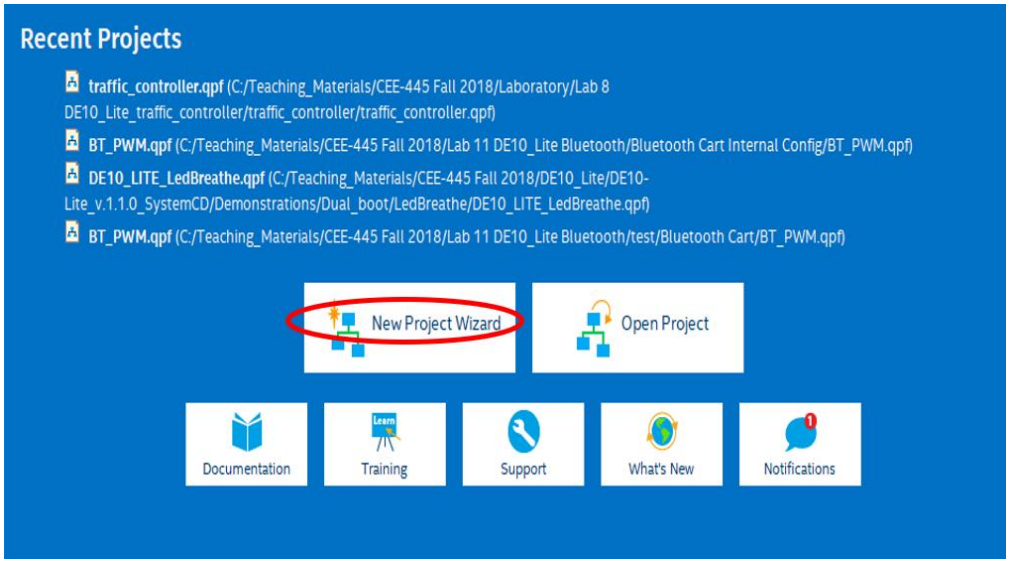
The example shows to download Quartus Lite Edition:

- There are three options for downloading the software. If you want to download the complete package, which requires several GB disk spaces, you must select "Combined Files". However, for the purpose of this course, there is no need to install all files. Therefore, you can select the "Individual Files" option and download the following files only, which takes less disk space in total:
- Note: Remember to download all the files into a same directory.
    - Quartus II Software (includes Nios II EDS)
    - ModelSim-Altera Edition (includes Starter Edition)
    - EPIC device support
    - Quartus II Help
- After downloading the required files, you are now ready to install the software. To do so, run the **QuartusSetup-18.1 (it can be a different setup file name)** and follow the installation process.
- Once the installation is complete, the software will be launched automatically.

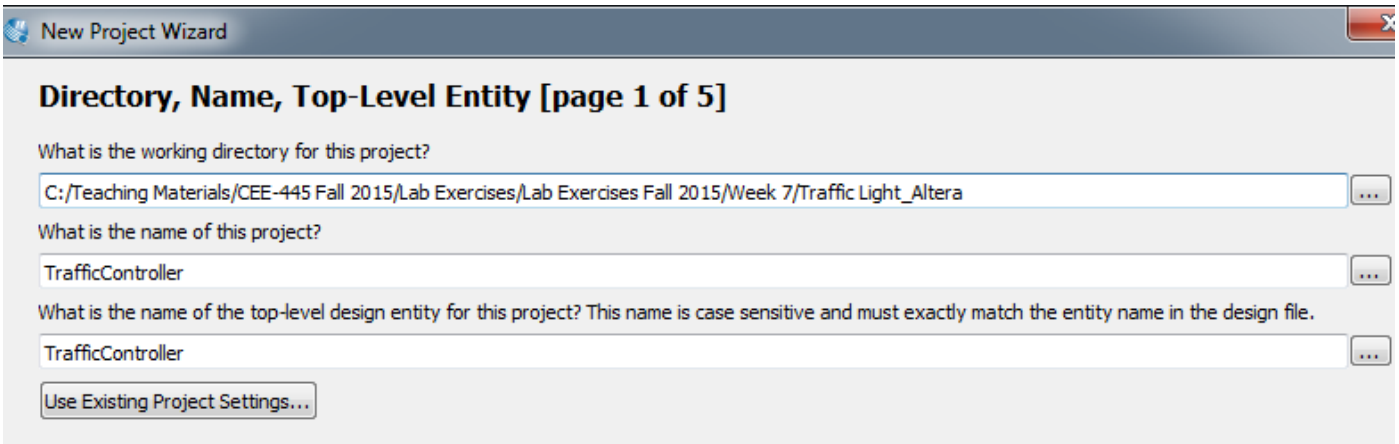
Traffic Light Controller Design:



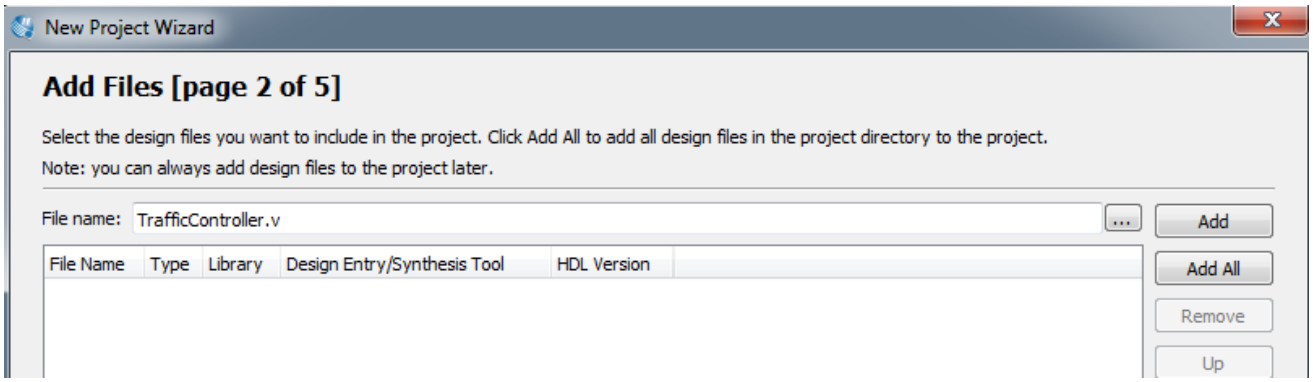
1. Click on the "Create a new project" button to build a new project.



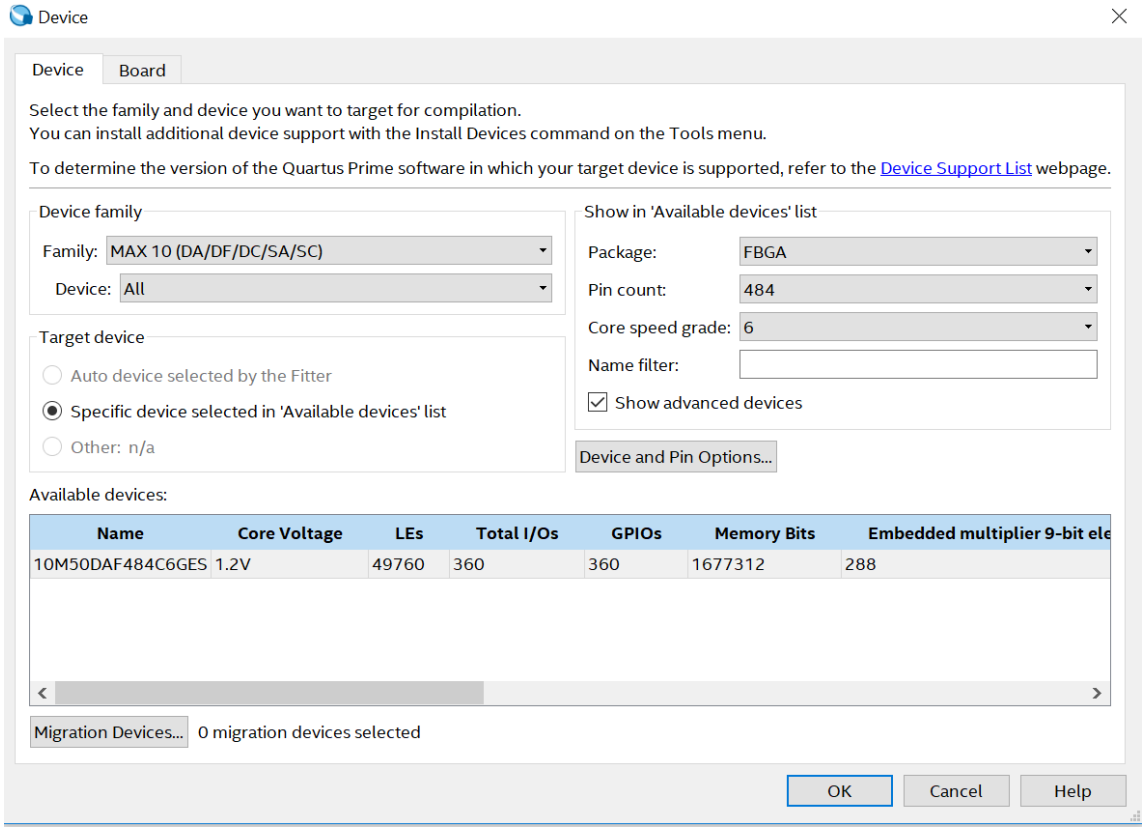
2. Select Next on a new project wizard, select a working directory and the name of this project.



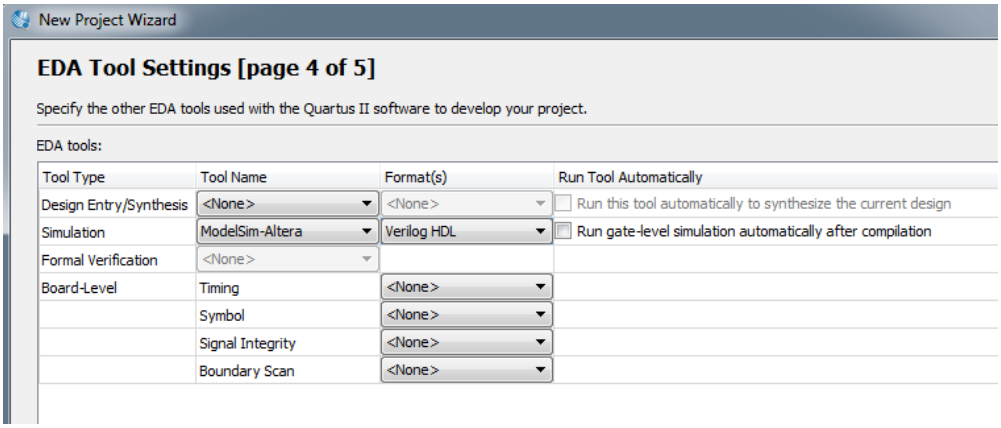
3. Download the traffic controller. v file from D2L and add it to your project.



4. Select Device: MAX 10, and then 10M50DAF484C6GES

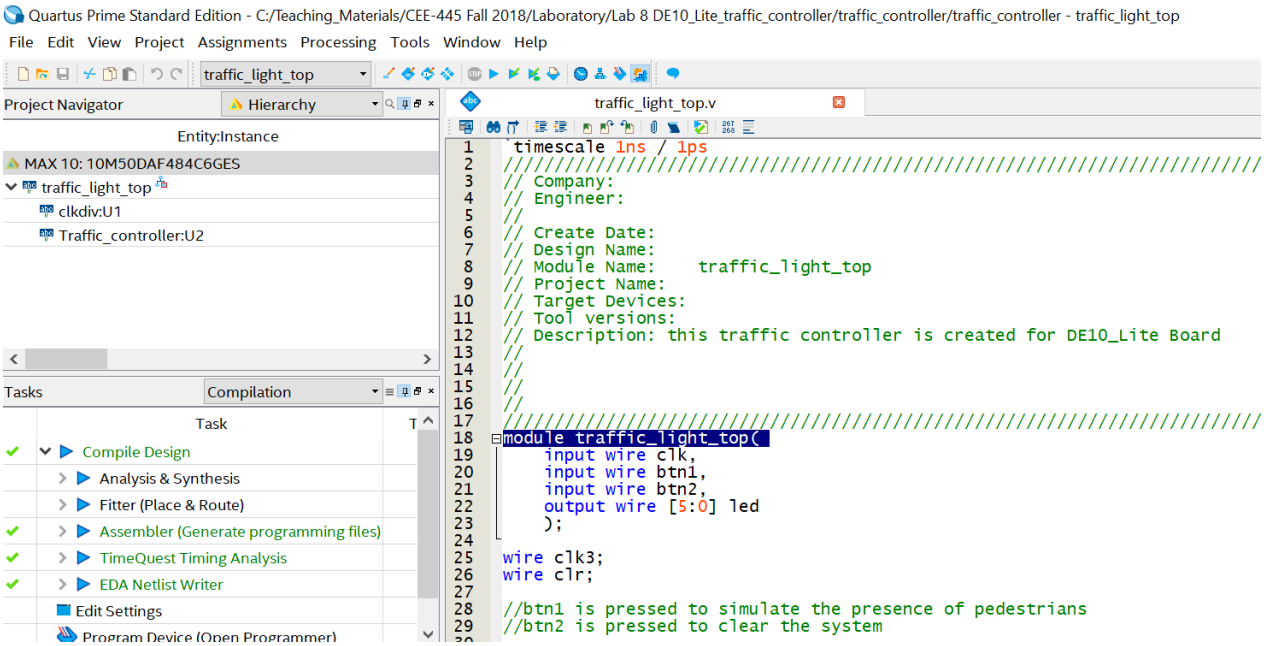


5. Select EDA tool

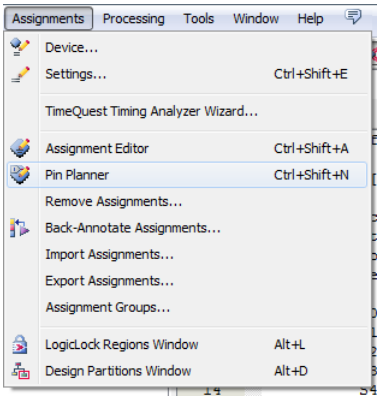


6. Click Finish.

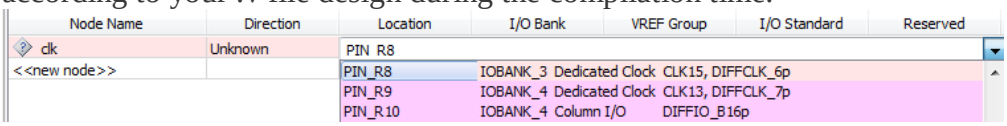
7. This project should look like the image as follows. If you can't find your Verilog source file for the Traffic Light Controller from your CEE-325 class last year, ask the instructor for help.



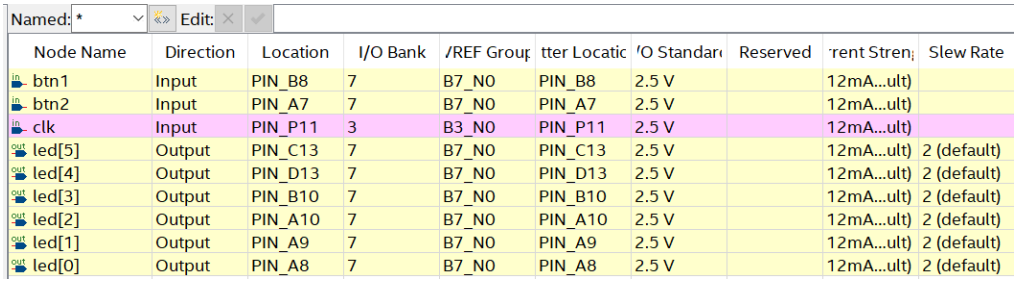
8. Select the Pin Planner to assign pins for the project. Select Assignments tab → Pin Planner as shown.



9. Add those pins as below. Type the input (clk) and outputs led[7] to led[0] to the Pin Planner as shown. Leave the Direction column unknown because the direction (input or output) is configured according to your .v file design during the compilation time.



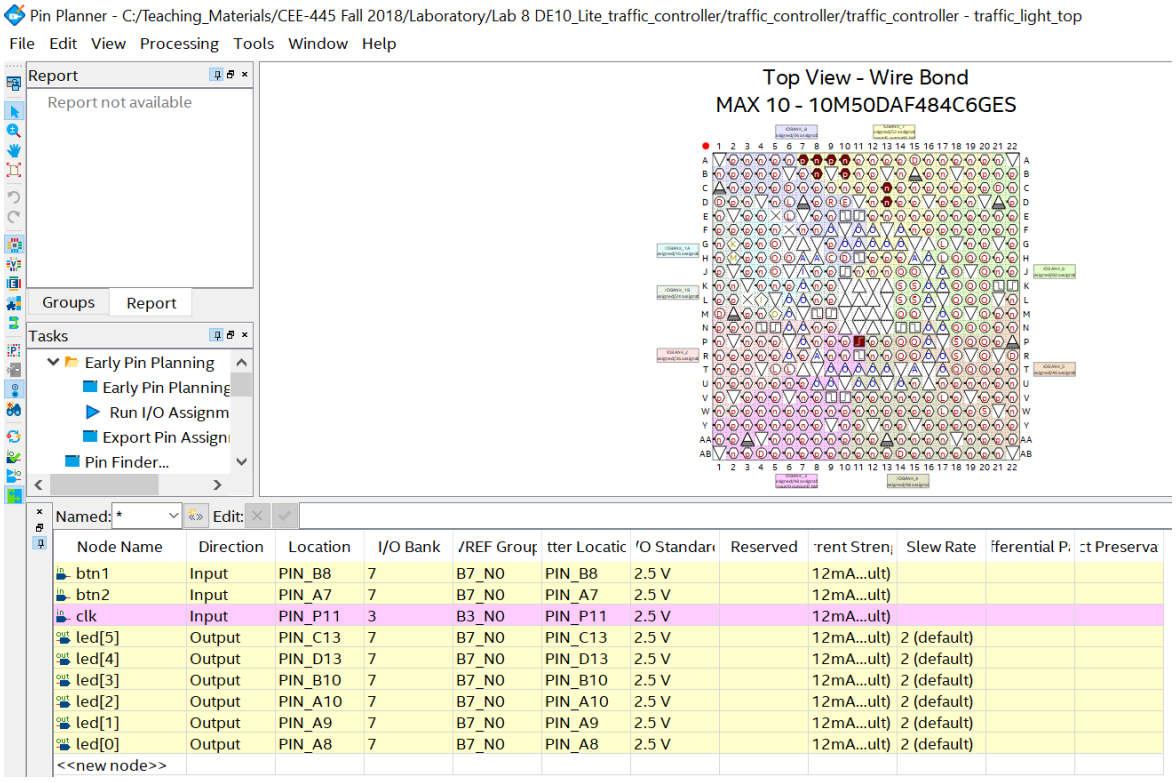
Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved
clk	Unknown	PIN_R8				
<<new node>>		PIN_R8	IOBANK_3	Dedicated Clock	CLK15, DIFFCLK_6p	
		PIN_R9	IOBANK_4	Dedicated Clock	CLK13, DIFFCLK_7p	
		PIN_R10	IOBANK_4	Column I/O	DIFFIO_B16p	



Named: * Edit:

Node Name	Direction	Location	I/O Bank	/REF Group	tter Locatic	/O Standar	Reserved	rent Stren	Slew Rate
btn1	Input	PIN_B8	7	B7_N0	PIN_B8	2.5 V		12mA...ult)	
btn2	Input	PIN_A7	7	B7_N0	PIN_A7	2.5 V		12mA...ult)	
clk	Input	PIN_P11	3	B3_N0	PIN_P11	2.5 V		12mA...ult)	
led[5]	Output	PIN_C13	7	B7_N0	PIN_C13	2.5 V		12mA...ult)	2 (default)
led[4]	Output	PIN_D13	7	B7_N0	PIN_D13	2.5 V		12mA...ult)	2 (default)
led[3]	Output	PIN_B10	7	B7_N0	PIN_B10	2.5 V		12mA...ult)	2 (default)
led[2]	Output	PIN_A10	7	B7_N0	PIN_A10	2.5 V		12mA...ult)	2 (default)
led[1]	Output	PIN_A9	7	B7_N0	PIN_A9	2.5 V		12mA...ult)	2 (default)
led[0]	Output	PIN_A8	7	B7_N0	PIN_A8	2.5 V		12mA...ult)	2 (default)

10. After compilation is run and complete, the input and output for each pin will show up according to your .v design file.

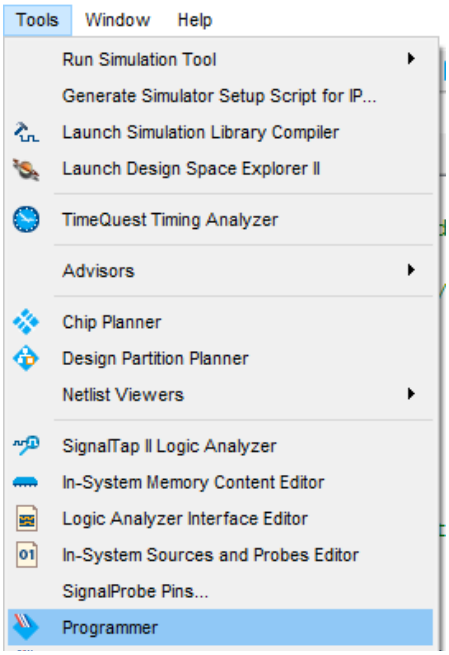


Node Name	Direction	Location	I/O Bank	/REF Group	tter Locatic	/O Standar	Reserved	rent Stren	Slew Rate	fferential P	ct Preserva
btn1	Input	PIN_B8	7	B7_N0	PIN_B8	2.5 V		12mA...ult)			
btn2	Input	PIN_A7	7	B7_N0	PIN_A7	2.5 V		12mA...ult)			
clk	Input	PIN_P11	3	B3_N0	PIN_P11	2.5 V		12mA...ult)			
led[5]	Output	PIN_C13	7	B7_N0	PIN_C13	2.5 V		12mA...ult)	2 (default)		
led[4]	Output	PIN_D13	7	B7_N0	PIN_D13	2.5 V		12mA...ult)	2 (default)		
led[3]	Output	PIN_B10	7	B7_N0	PIN_B10	2.5 V		12mA...ult)	2 (default)		
led[2]	Output	PIN_A10	7	B7_N0	PIN_A10	2.5 V		12mA...ult)	2 (default)		
led[1]	Output	PIN_A9	7	B7_N0	PIN_A9	2.5 V		12mA...ult)	2 (default)		
led[0]	Output	PIN_A8	7	B7_N0	PIN_A8	2.5 V		12mA...ult)	2 (default)		
<<new node>>											

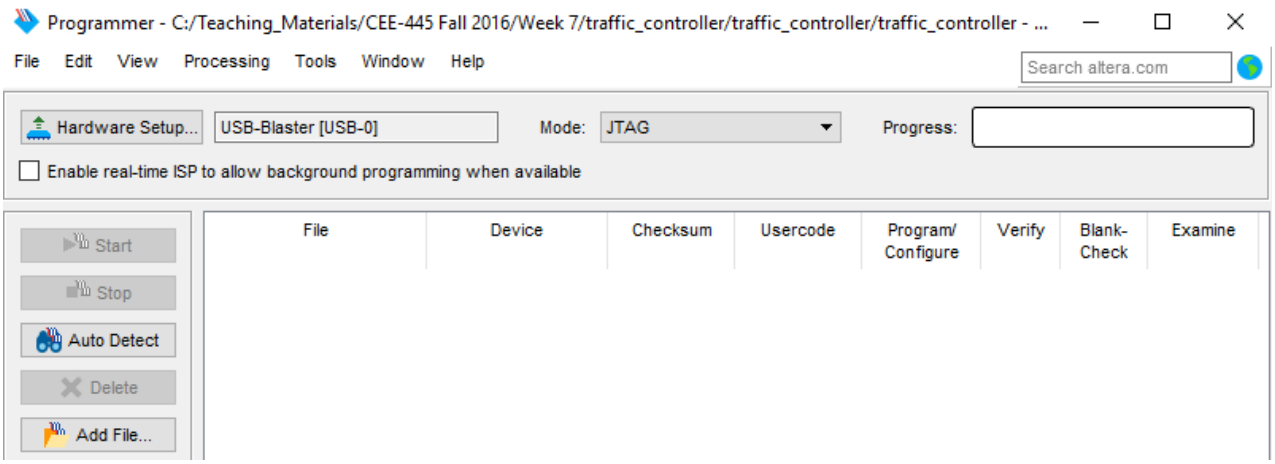
11. The USB-Blaster is used to download the bit stream file from the Quartus Prime to the FPGA board. The instruction on how to install the USB-blaster driver to your laptop can be found in a separate lab document (USB-Blaster driver installation guide).

12. Download the output file (.sof for Altera's FPGA) and verify the operation of the traffic light controller on the FPGA board.

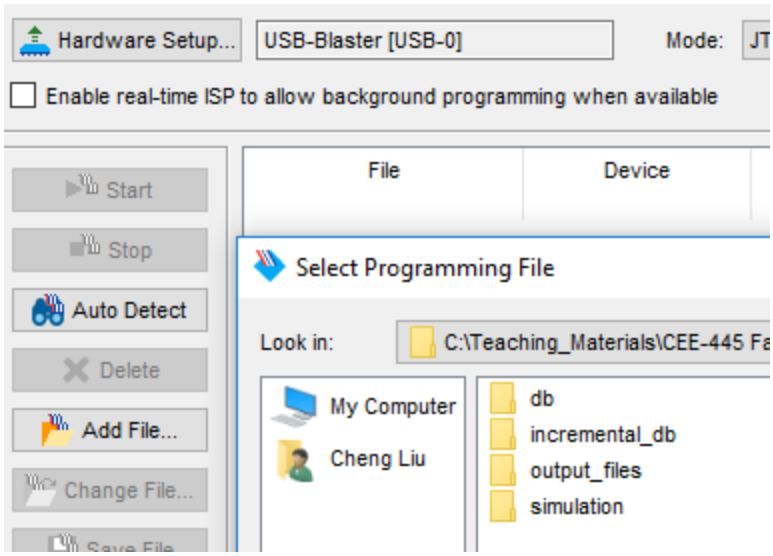
13. Select Tools → Programmer.



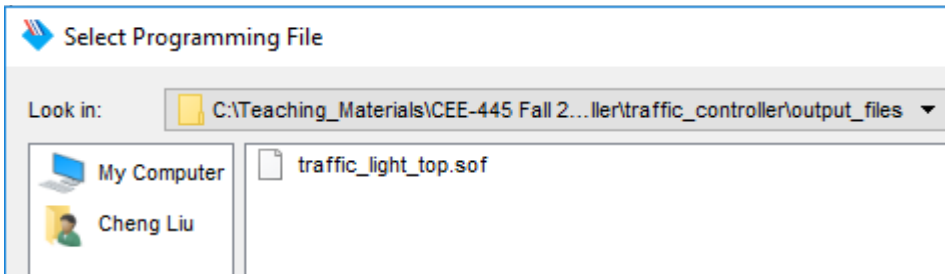
14. Check if you have the USB-blaster driver installed by selecting the hardware setup as shown below. If you are able to select USB-Blaster [USB-0], you had the driver installed, then proceed to the next step. If the USB Blaster driver is not installed, follow the USB-Blaster driver installation guide on D2L.



15. Next, select Add file and then double click on output_files folder



16. Select the .sof file to download to the Nano board.



17. The traffic light controller should be running on the board.

References:

Building Small Autonomous Robotic Vehicles Using an FPGA Platform: https://peer.asee.org/teaching-digital-designs-by-building-small-autonomous-robotic-vehicles-using-an-fpga-platform