EEL 3701C – Digital Logic & Computer System

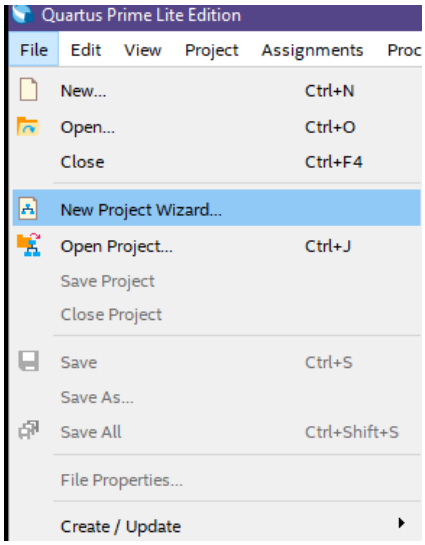
Getting Started With Quartus

This document assumes that Quartus Prime is installed on your computer. Quartus Prime is a Computer-aided design (CAD) software from Intel that includes everything needed to design for Intel FPGAs, SoCs, and complex programmable logic device (CPLD) such as the MAX10 development Board used in this class.

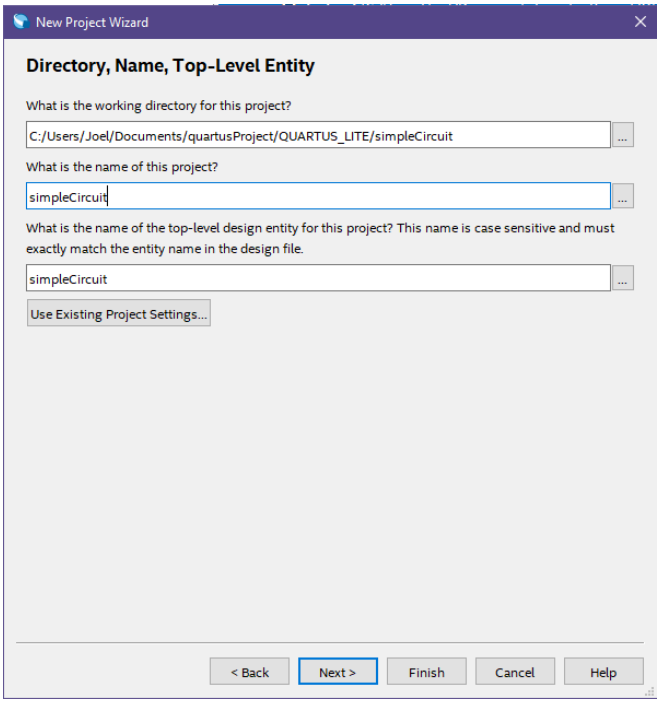
The tutorial below walks you through the specification of your circuit in Quartus, the synthesis and the configuration and test on the boards.

Creating the Quartus project

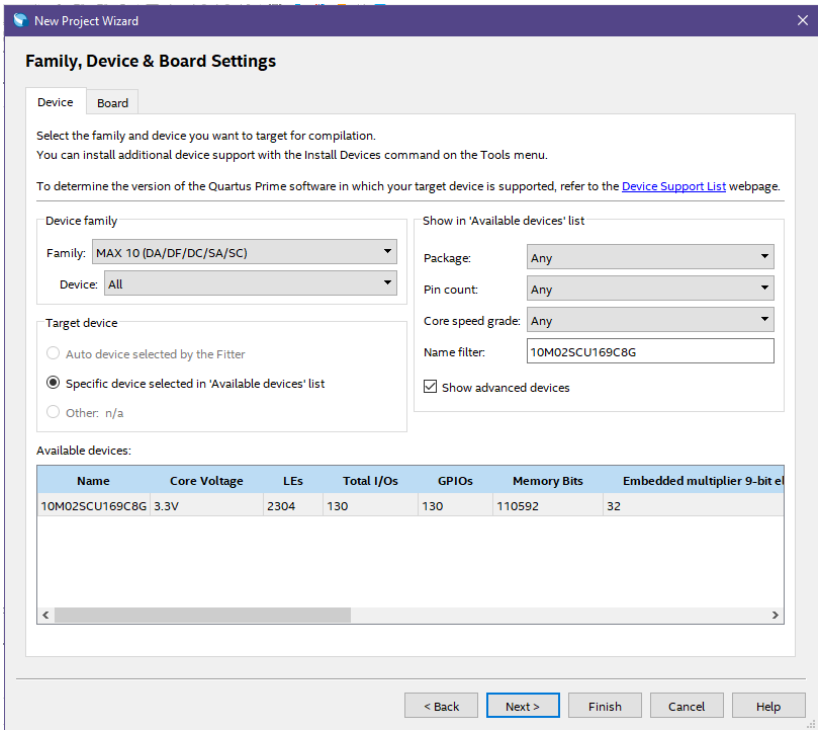
- Start Quartus by double-clicking on its icon on the desktop.
- Start a new project by clicking: **File-> New Project Wizard**.



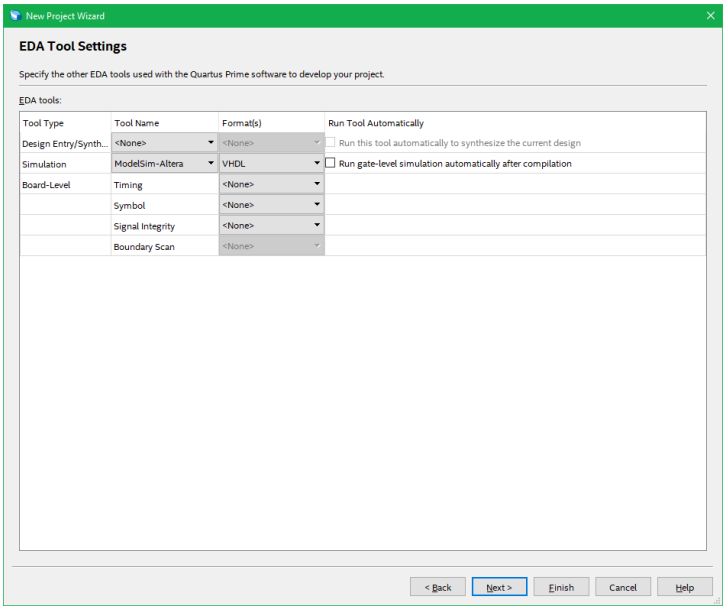
- Click “Next” until reaching the “**Directory, Name, Top-Level Entity**” page. Choose a directory for your project. You will most likely want to make a new folder for it. Call that folder “simpleCircuit” and select it.



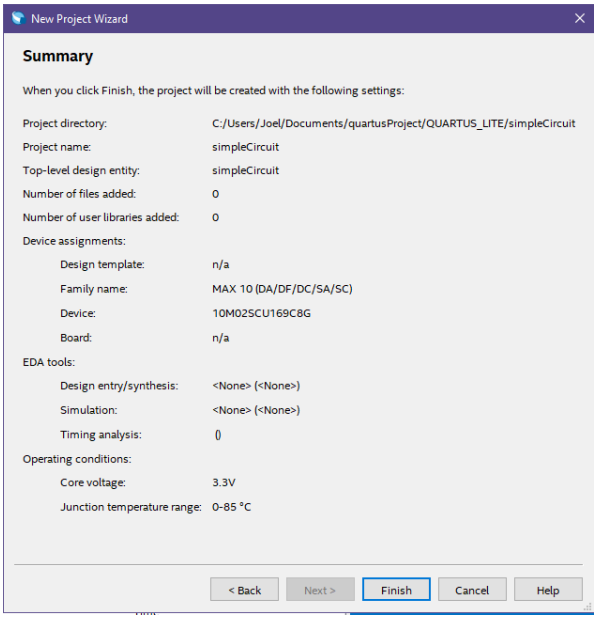
- Click “Next” until you get to the “**Family, Device & Board Settings**” page: select **MAX 10** for the **Family** and enter **10M02SCU169C8G** in the **Name Filter**. Then click “Next”.



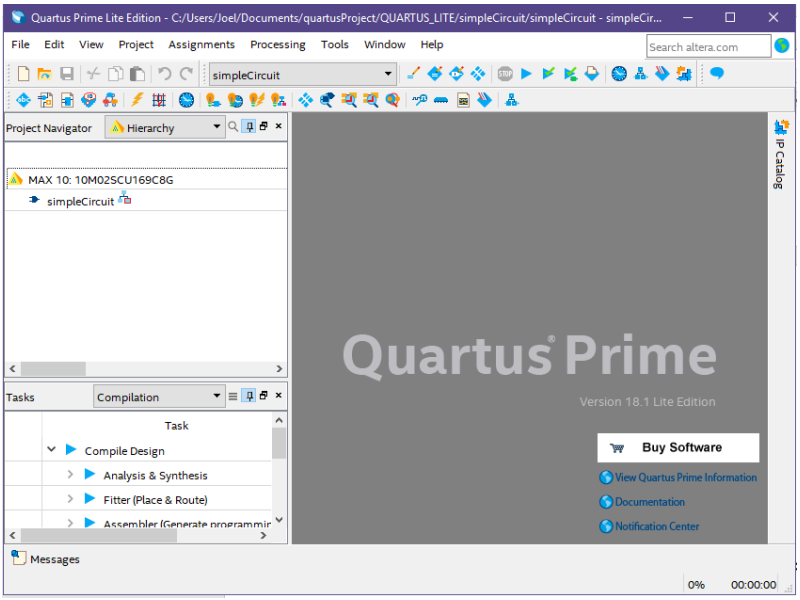
- On the EDA Tool Settings Page, on the row for **Simulation**, select **ModelSim-Altera** and **VHDL** for the two dropdown boxes.



Click “Next” to go to the final page. Then ,click finish.



After project creation, you should get a display like this:



We can now specify our circuit. **We will first do it in VHDL, then with the Quartus schematic editor.**

The first step (design entry) consists of creating the VHDL/schematic top file in which we will edit the circuit. Then, we will compile (synthesize) the design. Next, we will assign the pins (pin assignment) corresponding to the mapping inputs and outputs of our circuit. Finally, we will compile the design again to get the updated input/output pins, and then program it to our MXDB.

I) Designing your first circuit using VHDL

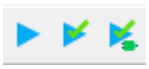
a. Design Entry

- Create a VHDL file: click **File-> New-> VHDL File**.
- Below is a simple snippet for you to implement.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

-- Define the entity and its inputs+outputs
entity simpleCircuit is
   PORT (
      A, B: in std_logic;
      C: out std_logic -- Notice the lack of semicolon on the last item in PORT
   );
end entity simpleCircuit;

-- Define the architecture of simpleCircuit - ie, what the interior of the entity looks like.
architecture arch_simpleCircuit of simpleCircuit is
begin

   C <= (not A and B) or (A and not B); -- Assign C to be /AB + A/B.

end architecture arch_simpleCircuit;
```

b. Synthesis

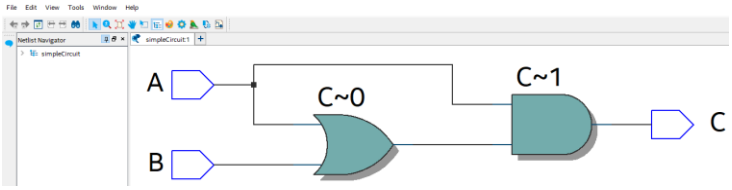
- Save (use the name “simpleCircuit.vhd”) and compile the design by clicking: **Processing-> Start -> Start Analysis and Synthesis**. Alternatively, you can do **Ctrl+K**, or click on the third play button at the top. Any of these will do a “**fast compile**.” The fast compile allows us to do pin assignments without spending time in the routing and fitting steps (which can take up to a minute to do). A “**full compile**” (the first play button) is much slower but is required in the final step to program it to the MXDB.



- The compiler should return a report like the one below. If you get any red colors here, you can read the errors you made in the Messages panel on the bottom. You may have to expand it out by dragging your mouse to read it in full view.

	Task	Time
	Compile Design	
✔	Analysis & Synthesis	00:00:15
	Fitter (Place & Route)	
	Assembler (Generate programming files)	
	Timing Analysis	
	EDA Netlist Writer	
	Edit Settings	
	Program Device (Open Programmer)	

- Optionally, you can visualize the schematic of your circuit (this can be useful to verify the functionality). Use **Tools-> Netlist Viewers->RTL Viewer**:

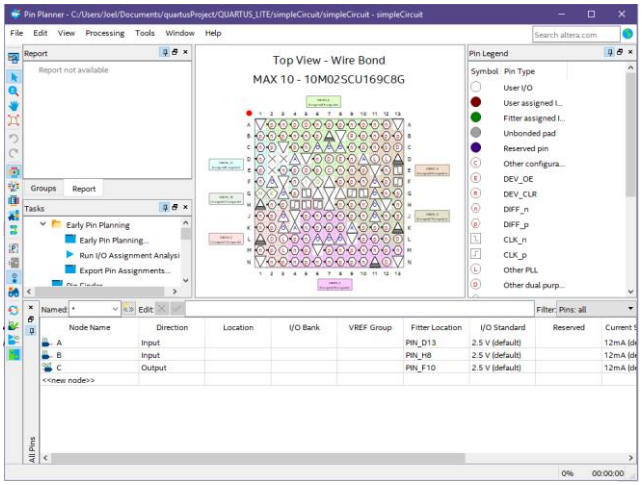


Observe that the resulting circuit corresponds to the VHDL specification. We can now proceed with the pin assignment.

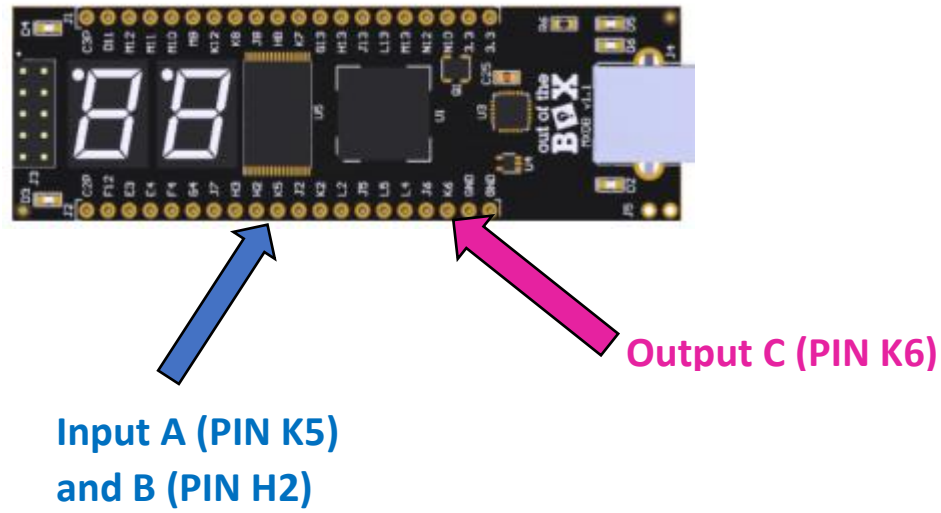
c. Pin Assignment

The purpose of the pin assignment is to connect the input and output of your design to interfaces (pins) on the board.

- Click on **Assignments->Pin Planner**. You should get something like:



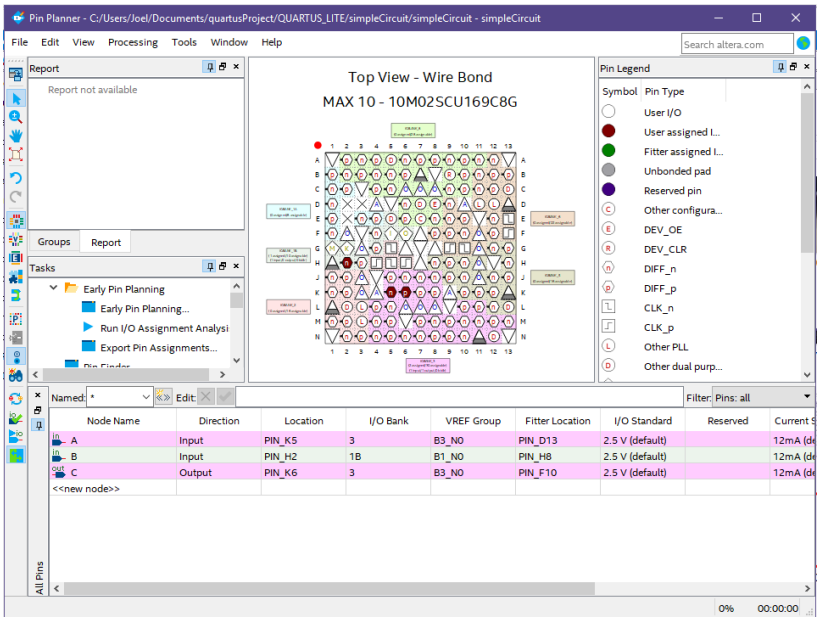
Here we must specify the **Locations** on the MAX10 chip where we want to assign port A, B, and C from our design. Remember that A and B are the inputs, and C is the output. To visually test our circuit on the MAX 10 board, we will assign A and B to 2 pull-up switches, and C to an LED. The following image shows how the FPGA has different general-purpose input and output (GPIO) pins. We can use any of them. Here, we consider putting the input A and B in K5 and H2 pin and output at K6.



We can now assign the pins to the 3 ports by single-clicking in the Location column of each pin, typing in the name of the pin, then pressing enter. For example, if you want to select pin D11, you simply type in **D11** and press enter.

Select K5 for A, H2 for B, and K6 for C.

You should see a screen like this after you have typed in the pin names:



We have one more step to do before we program the board. We need to set the MXDB to configure unused pins into tri-state. Click on **Assignments -> Device -> Device and Pin Options …** and you will come to a new window. On the left, click on **Unused Pins**. In the dropdown box that says **Reserve All Unused Pins**, select **"As Input Tri-Stated"**. You only have to do this tri-state step once per project.

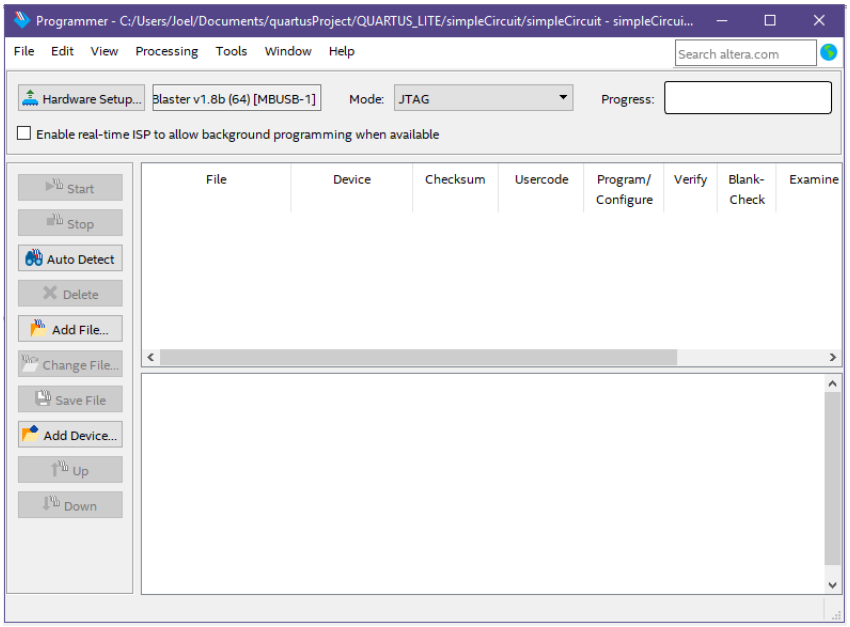
d. Synthesis (Compilation)

- Close Pin Planner and re-compile the design with a full compile (**Processing->Start Compilation** or the first play button)

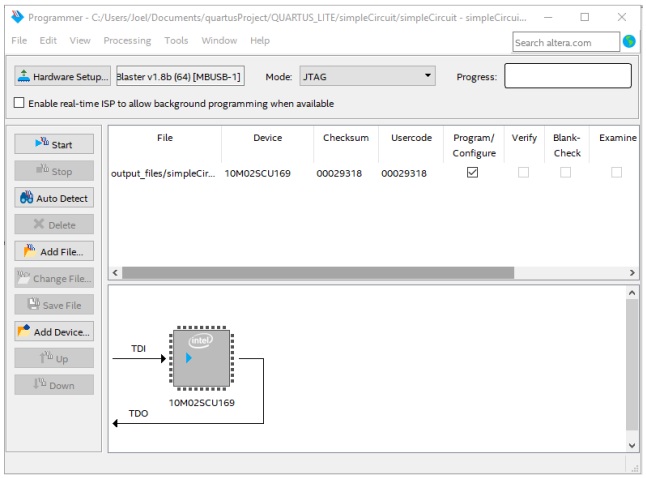
- Make sure that do not get any errors during the compilation

e. Configuration (Programing)

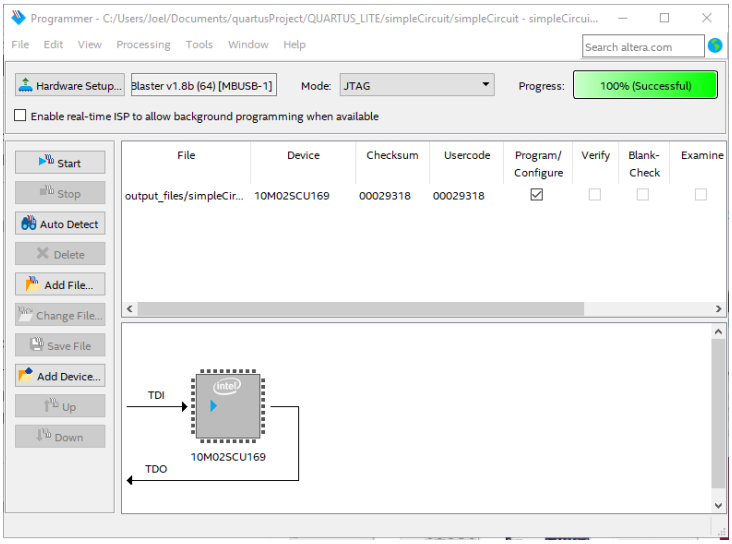
- Connect the power and JTAG USB cables to the board and to your **computer**. The power LED on the MAX10 board should now turn green. Unplug the MXDB from your breadboard during this step.
- In the Task panel (bottom-left corner of Quartus GUI), double-click on "**Program Device**"
- Make sure that you have **MBFTDI-Blaster…** in the "**Hardware Setup**" box. If you don't, click on "**Hardware Setup**", select the entry in "**Available hardware items**" that has **"Blaster"** in it. If the "**Available hardware items**" list is empty, go back to installing the driver and restart Quartus. Click close.



- Click on "**Add File…**", open the "**output_files**" folder, and double-click on "**simpleCircuit.sof**". You should get a display like this:
    - Note: SOF files are volatile meaning they are erased once the MXDB is powered down.
    - If you would like the program to persist on power down, select the **.pof** file instead. This will take around 30 seconds to compile instead of a single second.



- Simply click on "**Start**" to program the board. When you get the green light marked with "**100% (Successful)**", the design is downloaded into the MAX10.
    - If programming fails, make sure of the following:
        - The MXDB is plugged in with the green light on
        - The USB Blaster (the programmer) is plugged in with the green light on
        - The provided ribbon cable is plugged into both the MXDB and the Blaster. The ribbon cable should be going *away* from the boards in both cases (see the picture below for an example).
        - The blaster is plugged directly into the computer and not into a USB hub.

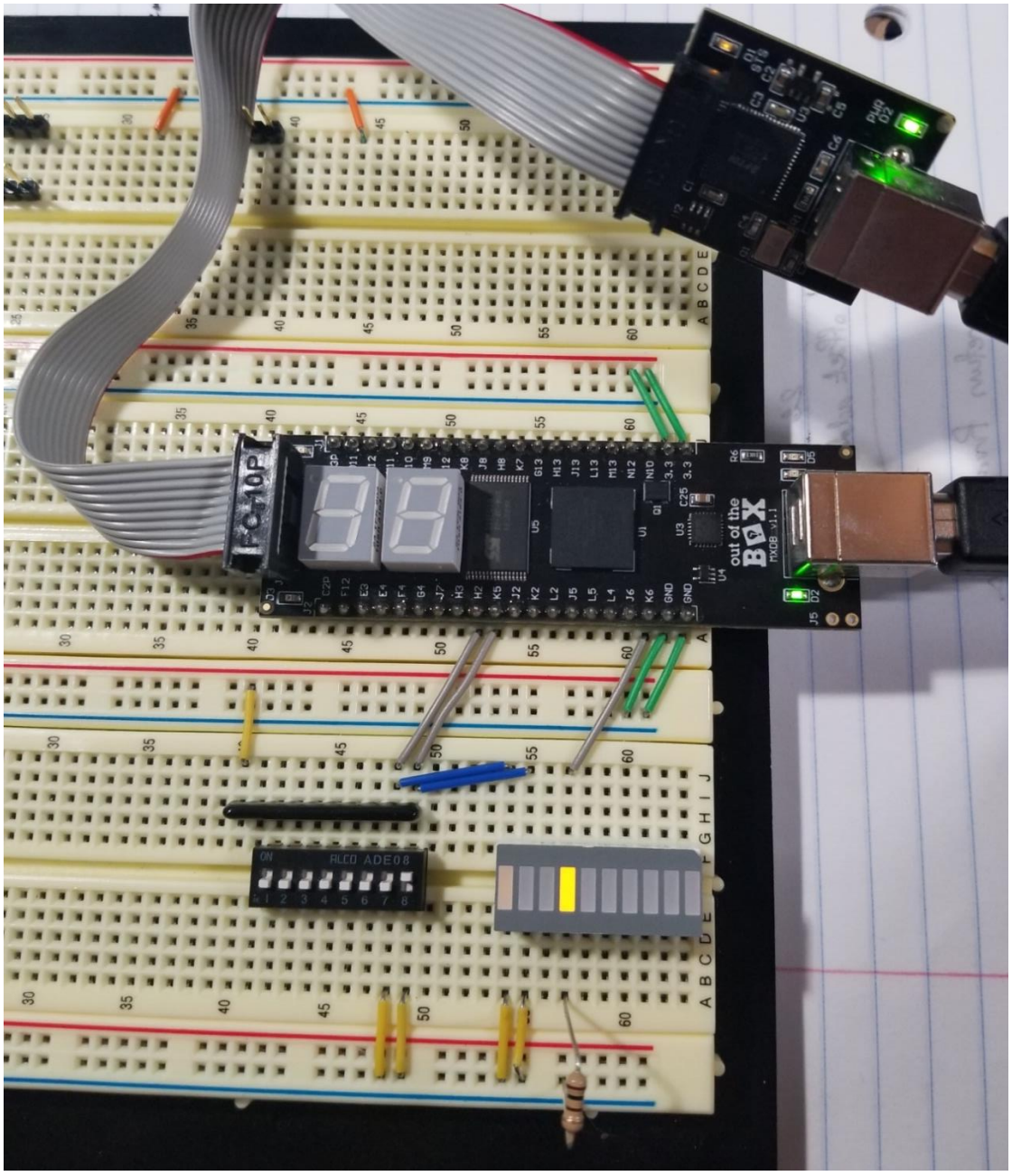


- Now your FPGA is programmed!

We will now verify the using pull-up switches and an LED. We will connect input A and B with two **pull-up switches** and output C with an LED **with a resistor!** We will also connect the inputs to the LED bank (the LEDs for A and B do not need a resistor on the cathode because the pull-up configuration already restricts the current).

Input A and B are connected to 7 and 8 on the switch bank. In addition, they are connected to the first and second LEDs. Output C is connected to the fourth LED.

The output LED will only turn on if A and B are different (ie., one is switched to ON and the other is switched to OFF). Also take note on how the input LED for A is illuminated even though A is switched to OFF. This is due to the pull-up configuration.



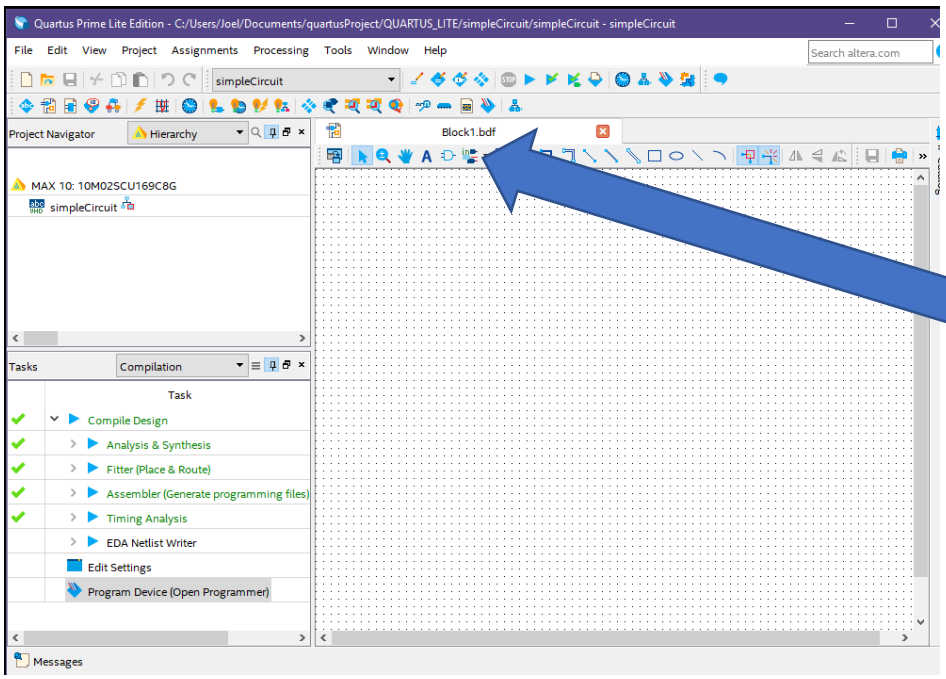
II) Designing your circuit using schematic editor

Implementing the circuit with the Schematic Editor instead of VHDL follows the exact same design steps with the only difference that we do not create a VHDL file but the schematic of the circuit.

You do not need to create a new project or close Quartus. The schematic will use the same pin assignments.

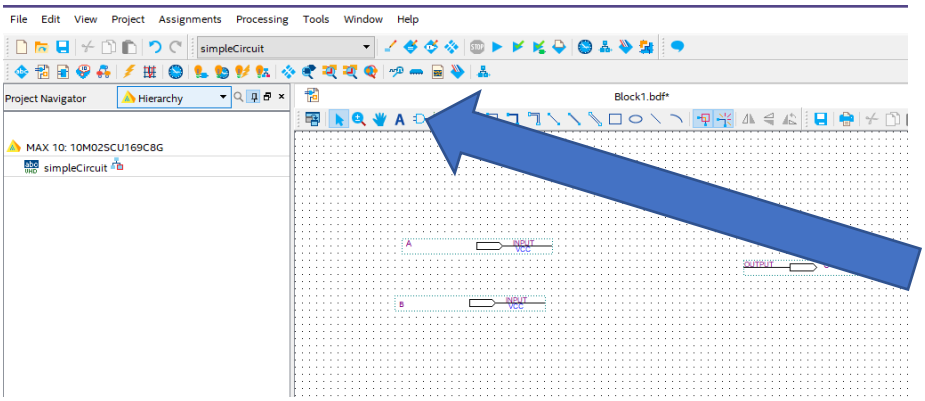
Follow the steps:

- On Quartus, double-click on **File->New->Block Diagram/Schematic File**. You should get a display similar to this:



Draw the inputs (port A and port B) and the output (port C) of our circuit by clicking here ^
Alternatively, you can double click any empty area and type in “input” or “output”.

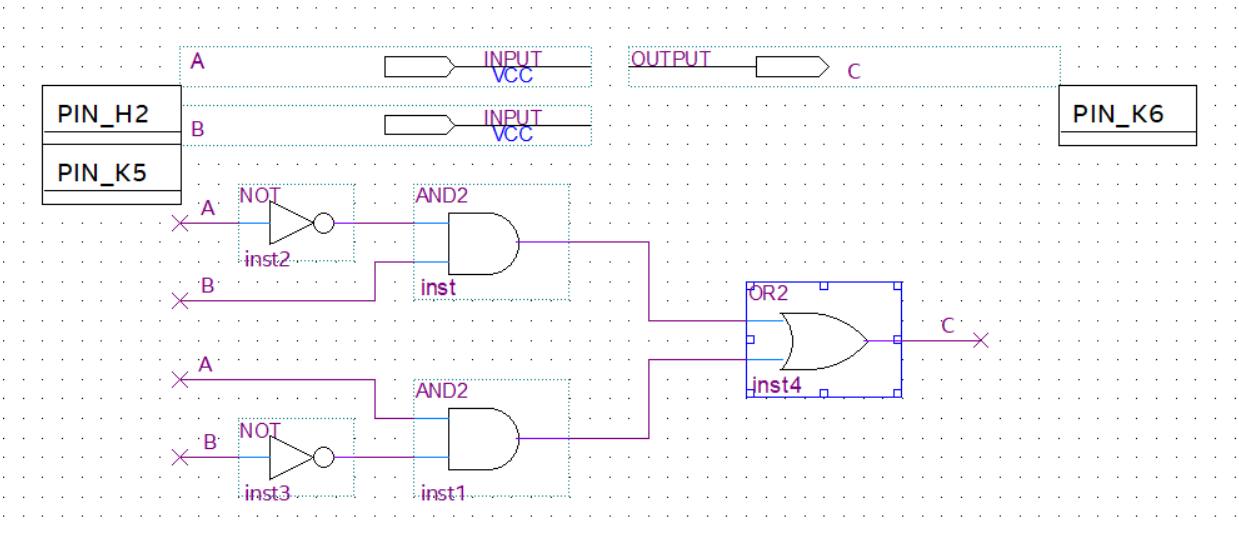
You will get the following figure.



Click here for the next step

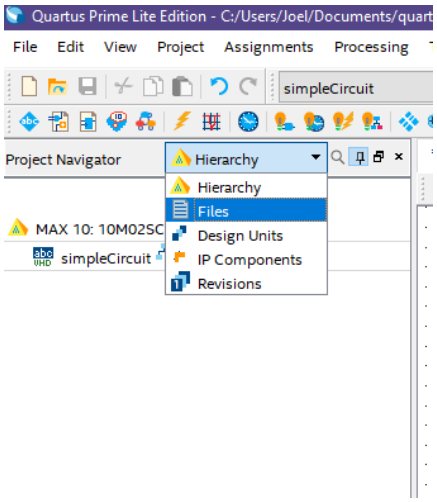
- Now add the AND and OR gates by clicking where it is pointed in the previous image. Alternatively, you can double click in any empty area.
  In the new window, type in “and2” and press Enter. Place this down in the schematic. Repeat this for another AND gate, two NOT gates (named “not”), and an OR gate (named “or2”).

- Connect the different components of the circuit simply by left-clicking the end of a wire and dragging to any position where you want the connection to be.
  - Note: In the below picture, the “PIN_XX” notes may only show up after you compile.
  - Click on the unattached wires and begin typing to label it. Anything with the same name gets connected invisibly.
  - A common compilation error can happen if Quartus does not automatically change the name of some of the instances, causing the instance name to occur twice. If this happens, double click on the offending instance name (such as “inst4” in the bottom right of the image) and rename it to something else.

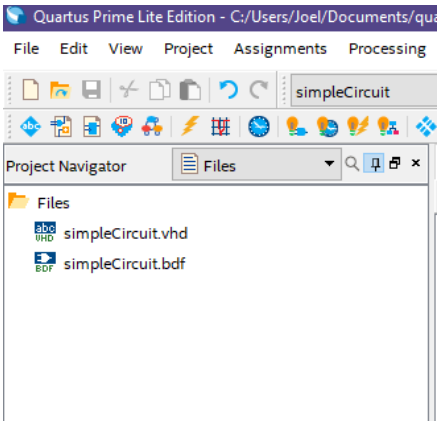


- Save the circuit under the name “**simpleCircuit.bdf**”

- Since we have want to compile a different file, we need to tell Quartus to change the top-level entity.
- In the Project Navigator, change from **Hierarchy** to **Files**.

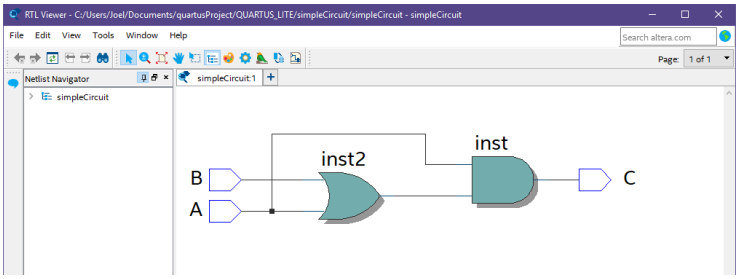


You will get the display:



- Right-click on "**simpleCircuit.bdf**" and select "**Set as Top-Level Entity**".
- To avoid conflicting names, we will remove "**simpleCircuit.vhd**" from the project: this will not delete the file but just remove it from the project. Right click on "simpleCircuit.vhd" and select "**Remove File from Project**".
    - In the future, if you choose different names for each file, you will not have to do this step.
- Now you can start compiling the new circuit defined as a schematic using the same approach seen previously: **Processing-> Start Compilation**, or the first play button at the top. We will not do a fast compile this time because we have already made the pin assignments.

- Once the design completes compiling (of course if you did not see any red errors), you can double check the correctness circuit in the RTL viewer as we did previously: **Tools-> Netlist Viewers-> RTL Viewer**.



The remaining steps consists of downloading the design into the MAX10 and visually verifying the functionality using switches and LEDs. Follow the exact same steps as we did with VHDL.

Go through all the steps once again and try to understand the whole design process, with VHDL and with the Schematic Editor. Throughout the semester, we will be using the same flow to design and test our circuits. Understand the design flow, master it and develop as much automatism as possible.