CPE 301: Altera FPGA Tutorial for 5CSEMA5F31C6N

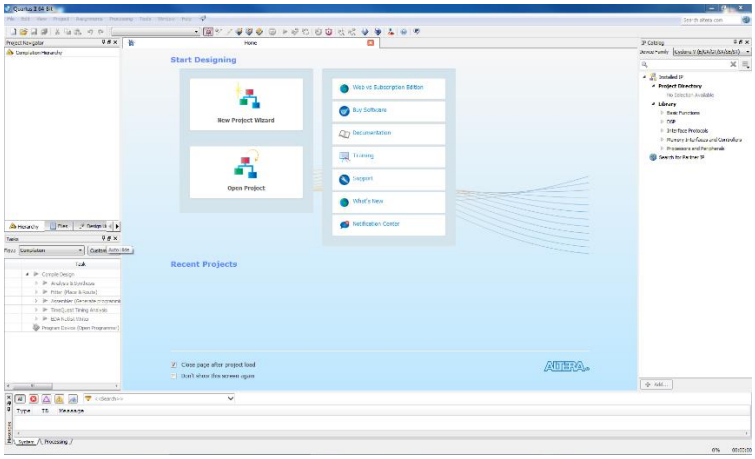
Spring 2019

This tutorial will guide you through the following steps:

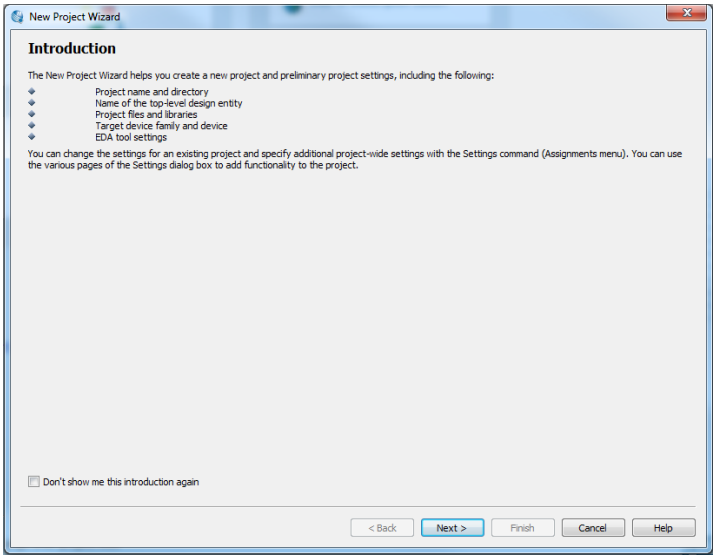
1. Creating a new project
2. Adding Source Code
3. Assigning Package Pins to Signals
4. Uploading the Code to the FPGA

Creating a New Project

1. Open Quartus II from the Start Menu / Desktop



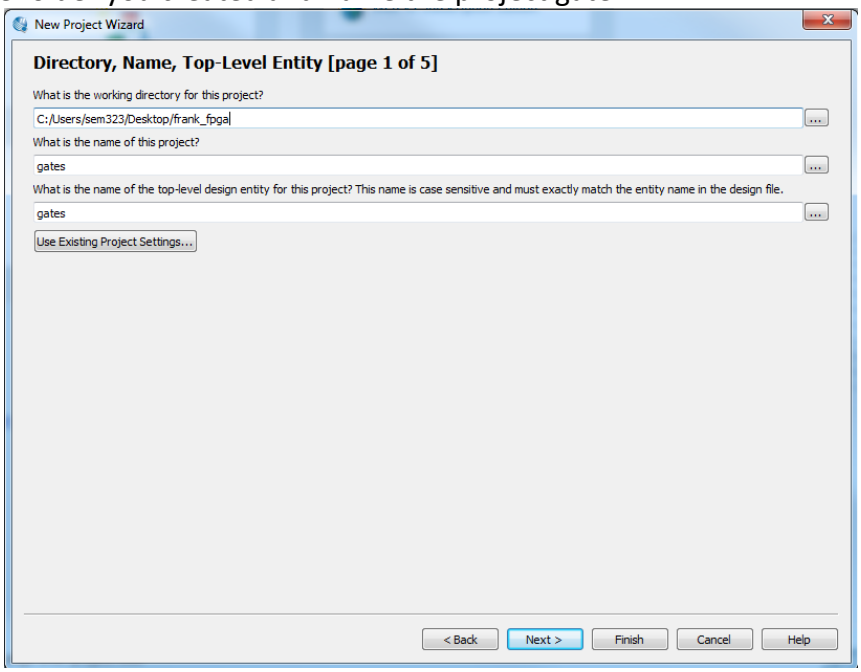
2. File -> New Project Wizard



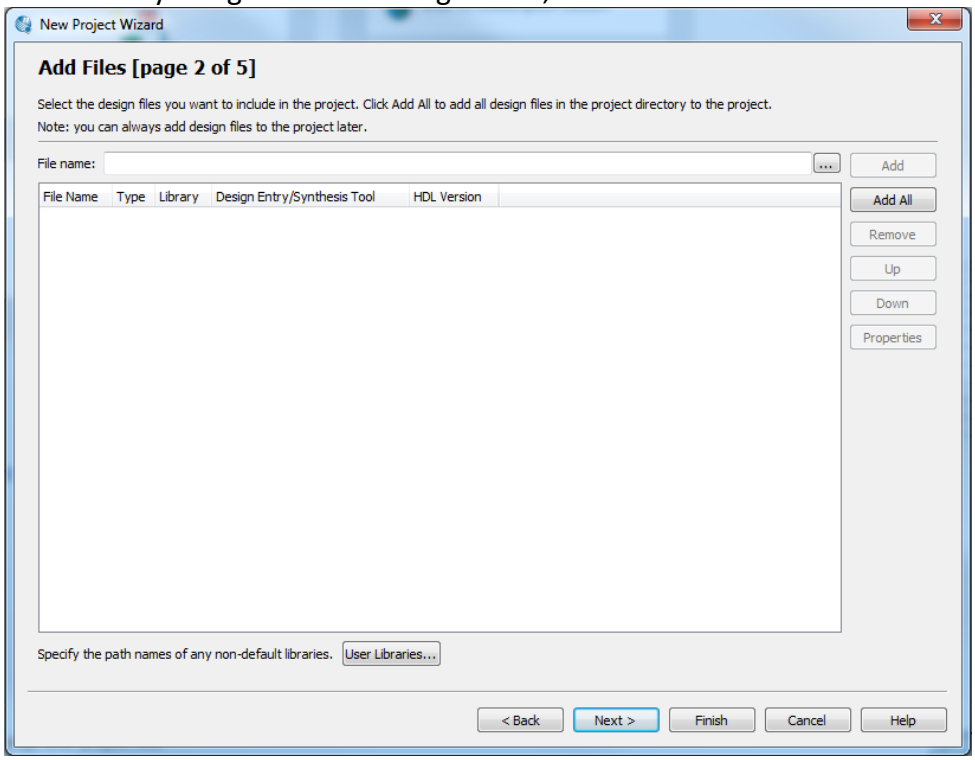
3. Create your own project folder on the Desktop.

a. Name it something unique.
b. Make sure it is on the local desktop, not on your cloud drive.
c. Explicitly place in on the C: drive : (C:\Users\<MY_USERNAME>\Desktop)

4. Choose the folder you created and name the project gate2.

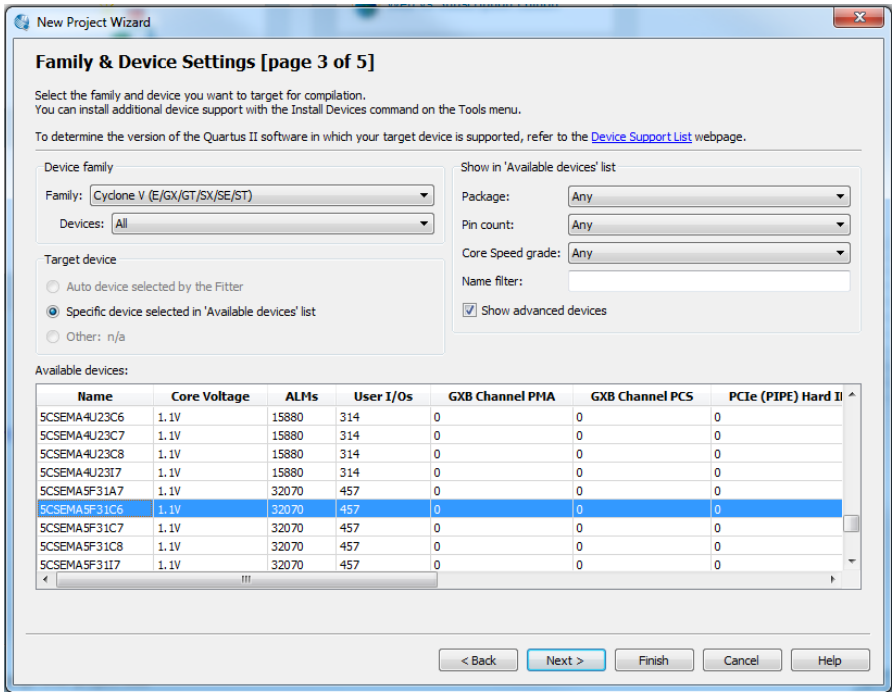


5. Click Next
6. We don't have any design files to add right now, so click Next

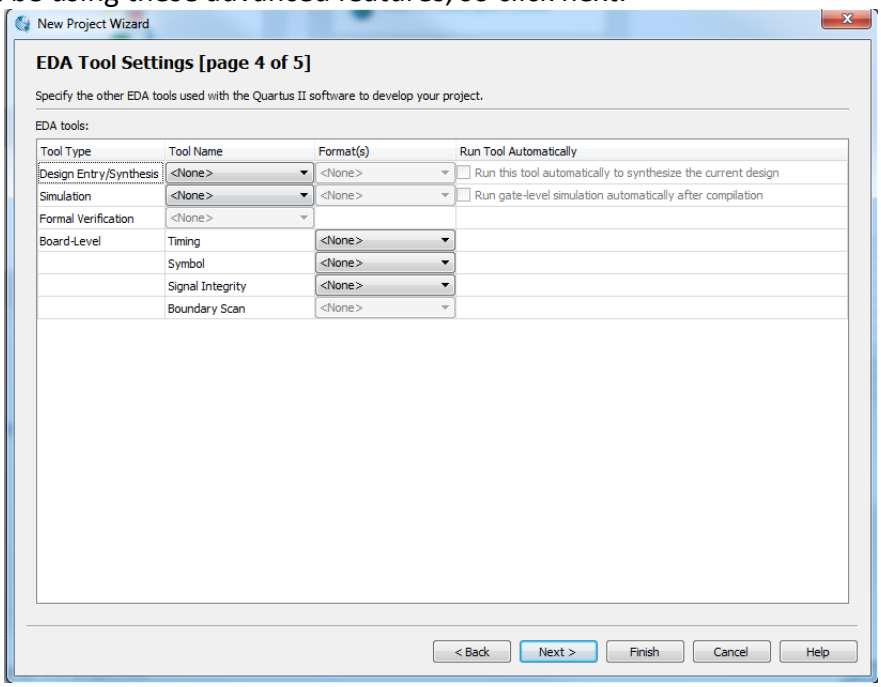


7. Select the correct device:
   a. Family: Cyclone V

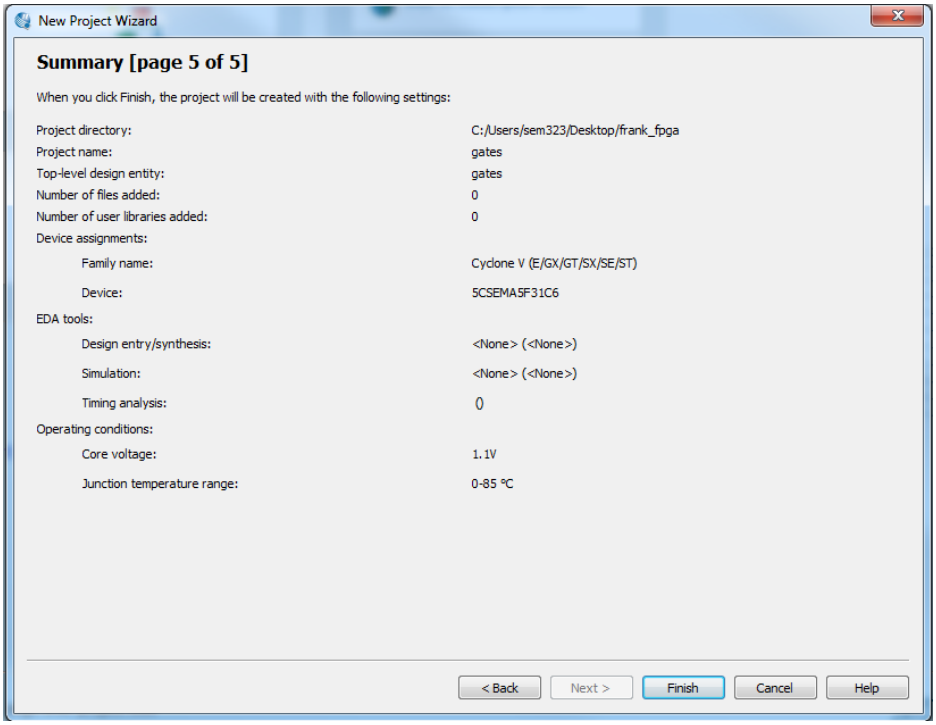
b. Device: 5CSEMA5F31C6N



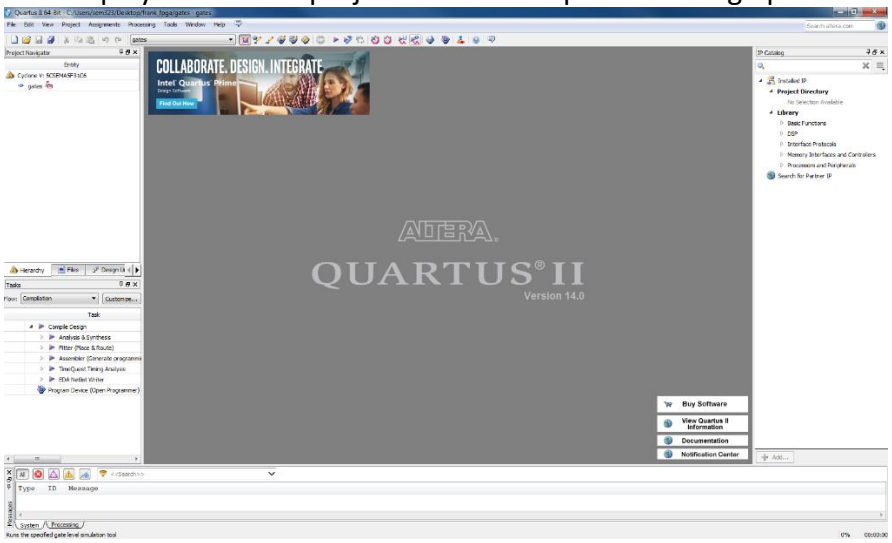
8. Click Next
9. We won't be using these advanced features, so click next.



10. Review the project settings, and click finish.

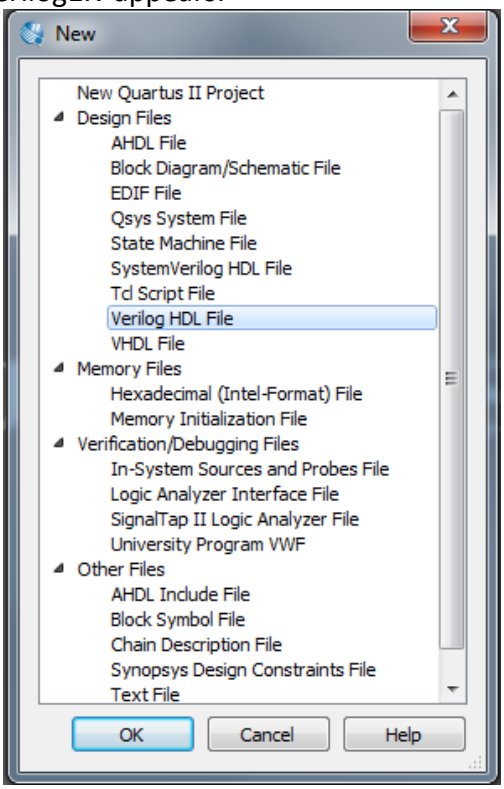


This is the general display of the new project. We've completed setting up our new project.



Adding new source files

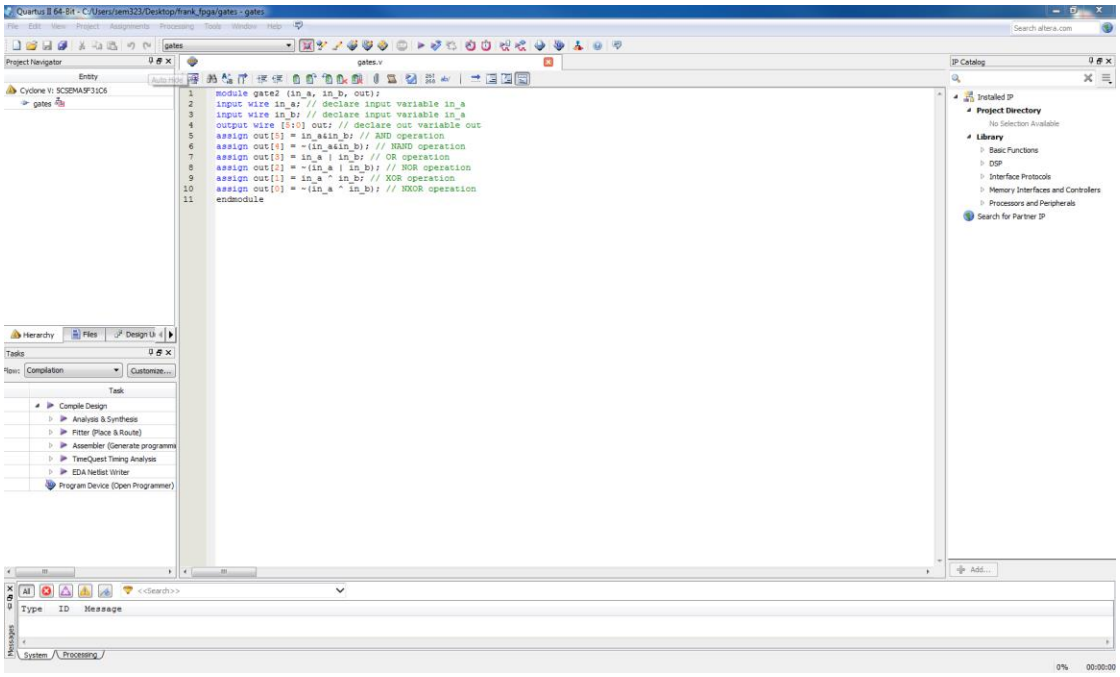
11. Let's add a new source code file to the project: File -> New -> Verilog HDL File
A blank file with the name Verilog1.v appears.



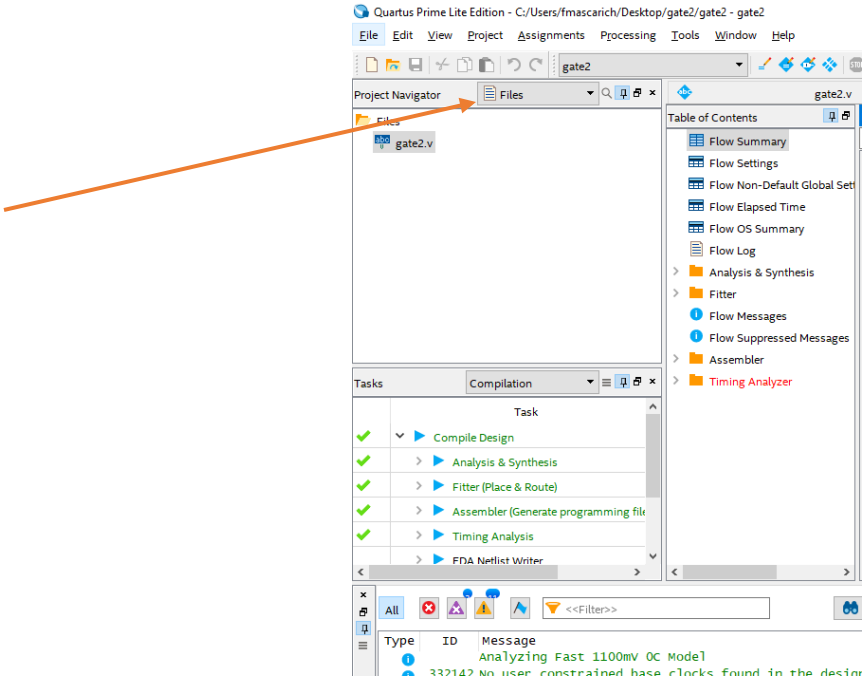
12. Now save it as gate2.v: File -> Save as
13. Copy this example code into the source file:

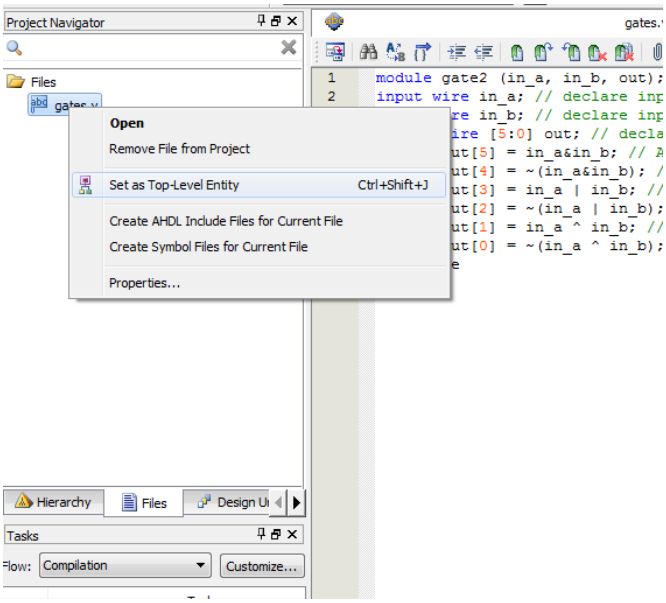
```
module gate2 (in_a, in_b, out);
input wire in_a; // declare input variable in_a
input wire in_b; // declare input variable in_a
output wire [5:0] out; // declare out variable out
assign out[5] = in_a&in_b; // AND operation
assign out[4] = ~(in_a&in_b); // NAND operation
assign out[3] = in_a | in_b; // OR operation
assign out[2] = ~(in_a | in_b); // NOR operation
assign out[1] = in_a ^ in_b; // XOR operation
assign out[0] = ~(in_a ^ in_b); // NXOR operation
endmodule
```

Your screen should then look like this:

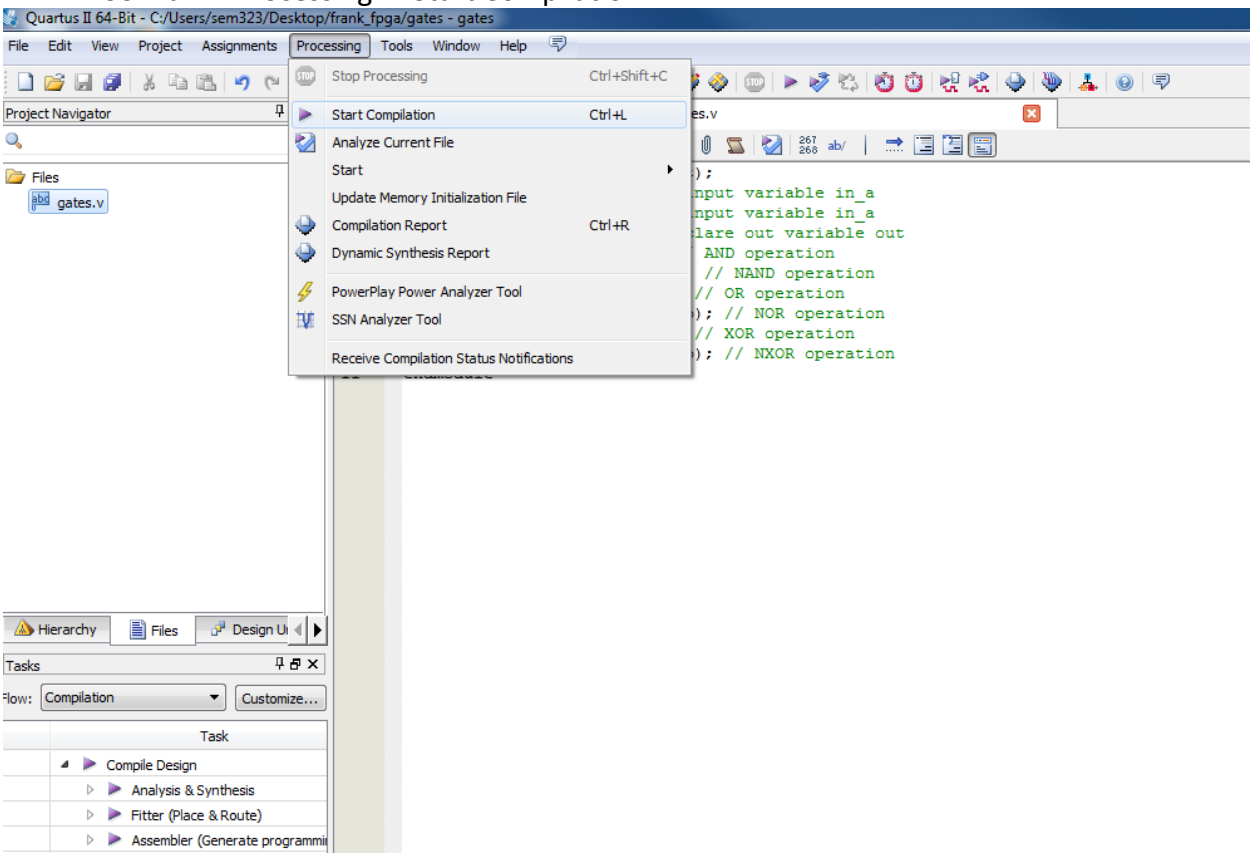


14. Right-click the source file in the file hierarchy and set the file as the top level entity. (You may need to select “Files” in the project navigator)

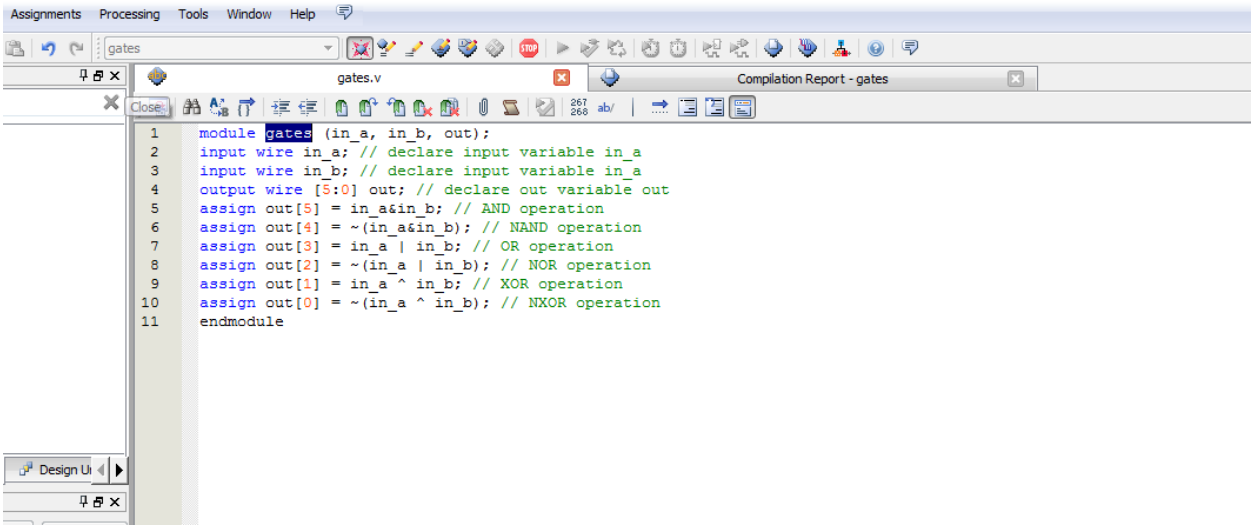




15. Double-check that you selected the correct device:
    Tool bar ->Assignments ->Devices..: You should see Cyclone V, 5CSEMAF31C6.
    You can also go to the Boards menu and select the DE1-SoC at the bottom of the list.
16. Start compiling the code:
    Tool Bar -> Processing -> Start Compilation



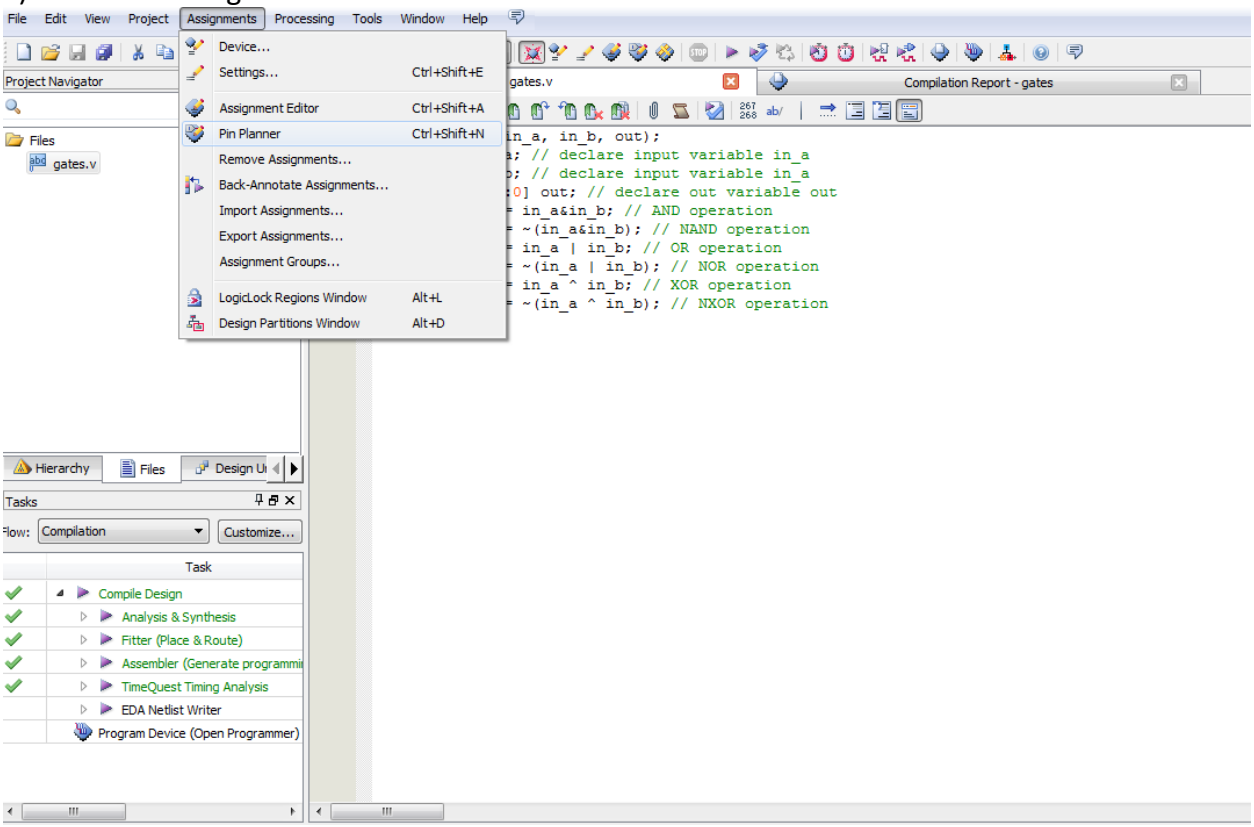
Make sure the module name matches the file name:



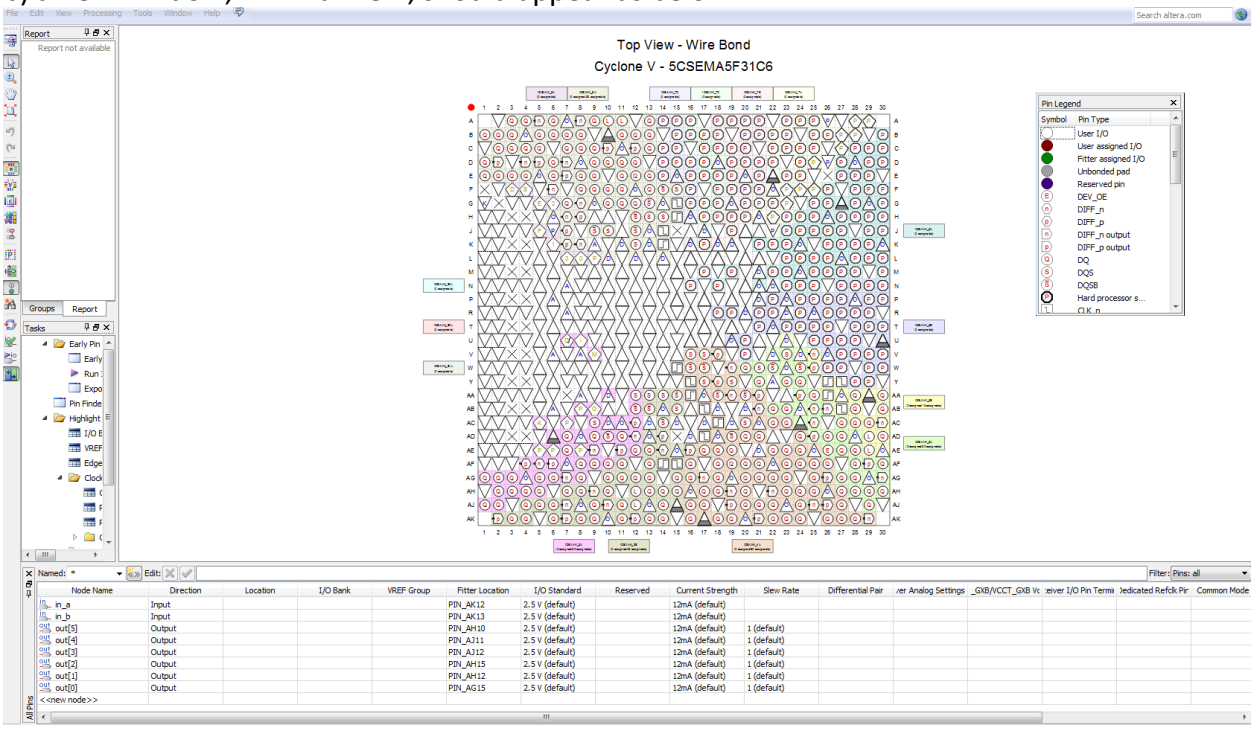
Assigning Pins to Signals

17. Assign the pins to the signals.

a) Tool bar ->Assignments -> Pins



b) a new window,"Pin Planner", should appear as below:



Note: if you don't see the bottom window, go to View ->All Pins List from the Pin Planner window.

Here are the pin assignments you'll need as provided by Quartus:

LED Pins:

Signal Name	FPGA Pin No.	Description	I/O Standard
LEDR[0]	PIN_V16	LED [0]	3.3V
LEDR[1]	PIN_W16	LED [1]	3.3V
LEDR[2]	PIN_V17	LED [2]	3.3V
LEDR[3]	PIN_V18	LED [3]	3.3V
LEDR[4]	PIN_W17	LED [4]	3.3V
LEDR[5]	PIN_W19	LED [5]	3.3V
LEDR[6]	PIN_Y19	LED [6]	3.3V
LEDR[7]	PIN_W20	LED [7]	3.3V
LEDR[8]	PIN_W21	LED [8]	3.3V
LEDR[9]	PIN_Y21	LED [9]	3.3V

Slide Switch Pins:

Signal Name	FPGA Pin No.	Description	I/O Standard
SW[0]	PIN_AB12	Slide Switch[0]	3.3V
SW[1]	PIN_AC12	Slide Switch[1]	3.3V
SW[2]	PIN_AF9	Slide Switch[2]	3.3V
SW[3]	PIN_AF10	Slide Switch[3]	3.3V
SW[4]	PIN_AD11	Slide Switch[4]	3.3V
SW[5]	PIN_AD12	Slide Switch[5]	3.3V
SW[6]	PIN_AE11	Slide Switch[6]	3.3V
SW[7]	PIN_AC9	Slide Switch[7]	3.3V
SW[8]	PIN_AD10	Slide Switch[8]	3.3V
SW[9]	PIN_AE12	Slide Switch[9]	3.3V

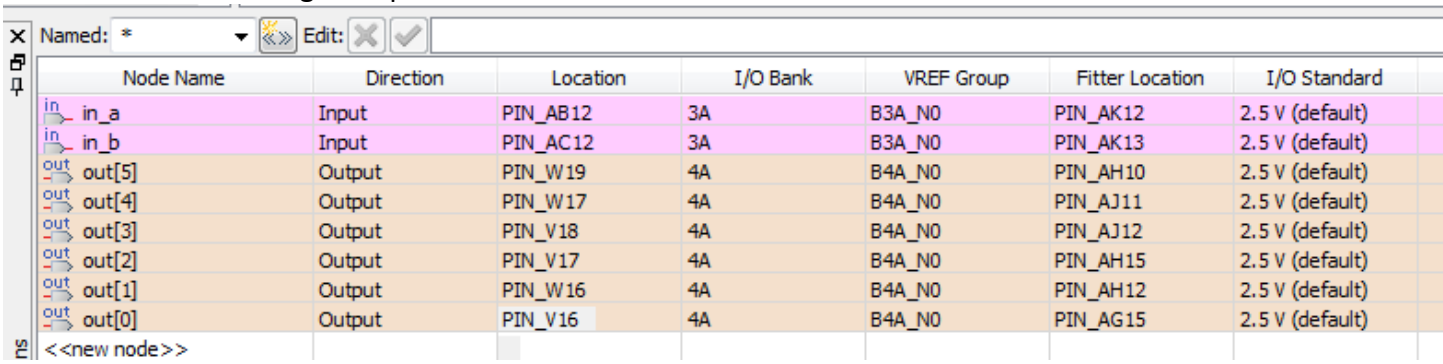
Push Button Pins:

Signal Name	FPGA Pin No.	Description	I/O Standard
KEY[0]	PIN_AA14	Push-button[0]	3.3V
KEY[1]	PIN_AA15	Push-button[1]	3.3V
KEY[2]	PIN_W15	Push-button[2]	3.3V
KEY[3]	PIN_Y16	Push-button[3]	3.3V

7 Segment Pins:

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX0[0]	PIN_AE26	Seven Segment Digit 0[0]	3.3V
HEX0[1]	PIN_AE27	Seven Segment Digit 0[1]	3.3V
HEX0[2]	PIN_AE28	Seven Segment Digit 0[2]	3.3V
HEX0[3]	PIN_AG27	Seven Segment Digit 0[3]	3.3V
HEX0[4]	PIN_AF28	Seven Segment Digit 0[4]	3.3V
HEX0[5]	PIN_AG28	Seven Segment Digit 0[5]	3.3V
HEX0[6]	PIN_AH28	Seven Segment Digit 0[6]	3.3V
HEX1[0]	PIN_AJ29	Seven Segment Digit 1[0]	3.3V
HEX1[1]	PIN_AH29	Seven Segment Digit 1[1]	3.3V
HEX1[2]	PIN_AH30	Seven Segment Digit 1[2]	3.3V
HEX1[3]	PIN_AG30	Seven Segment Digit 1[3]	3.3V
HEX1[4]	PIN_AF29	Seven Segment Digit 1[4]	3.3V
HEX1[5]	PIN_AF30	Seven Segment Digit 1[5]	3.3V
HEX1[6]	PIN_AD27	Seven Segment Digit 1[6]	3.3V
HEX2[0]	PIN_AB23	Seven Segment Digit 2[0]	3.3V
HEX2[1]	PIN_AE29	Seven Segment Digit 2[1]	3.3V
HEX2[2]	PIN_AD29	Seven Segment Digit 2[2]	3.3V
HEX2[3]	PIN_AC28	Seven Segment Digit 2[3]	3.3V
HEX2[4]	PIN_AD30	Seven Segment Digit 2[4]	3.3V
HEX2[5]	PIN_AC29	Seven Segment Digit 2[5]	3.3V
HEX2[6]	PIN_AC30	Seven Segment Digit 2[6]	3.3V
HEX3[0]	PIN_AD26	Seven Segment Digit 3[0]	3.3V
HEX3[1]	PIN_AC27	Seven Segment Digit 3[1]	3.3V
HEX3[2]	PIN_AD25	Seven Segment Digit 3[2]	3.3V
HEX3[3]	PIN_AC25	Seven Segment Digit 3[3]	3.3V
HEX3[4]	PIN_AB28	Seven Segment Digit 3[4]	3.3V
HEX3[5]	PIN_AB25	Seven Segment Digit 3[5]	3.3V
HEX3[6]	PIN_AB22	Seven Segment Digit 3[6]	3.3V
HEX4[0]	PIN_AA24	Seven Segment Digit 4[0]	3.3V
HEX4[1]	PIN_Y23	Seven Segment Digit 4[1]	3.3V
HEX4[2]	PIN_Y24	Seven Segment Digit 4[2]	3.3V
HEX4[3]	PIN_W22	Seven Segment Digit 4[3]	3.3V
HEX4[4]	PIN_W24	Seven Segment Digit 4[4]	3.3V
HEX4[5]	PIN_V23	Seven Segment Digit 4[5]	3.3V
HEX4[6]	PIN_W25	Seven Segment Digit 4[6]	3.3V
HEX5[0]	PIN_V25	Seven Segment Digit 5[0]	3.3V
HEX5[1]	PIN_AA28	Seven Segment Digit 5[1]	3.3V
HEX5[2]	PIN_Y27	Seven Segment Digit 5[2]	3.3V
HEX5[3]	PIN_AB27	Seven Segment Digit 5[3]	3.3V
HEX5[4]	PIN_AB26	Seven Segment Digit 5[4]	3.3V
HEX5[5]	PIN_AA26	Seven Segment Digit 5[5]	3.3V
HEX5[6]	PIN_AA25	Seven Segment Digit 5[6]	3.3V

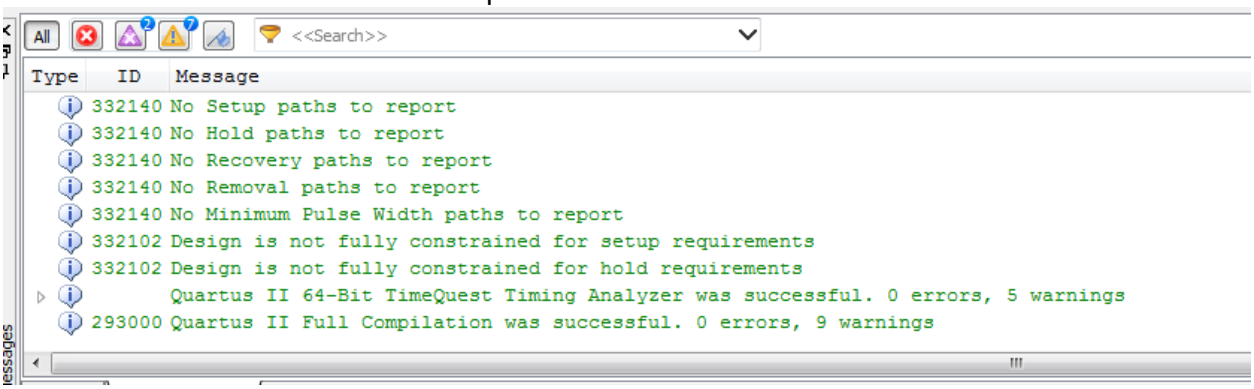
18. Now we will assign the pins as follows:



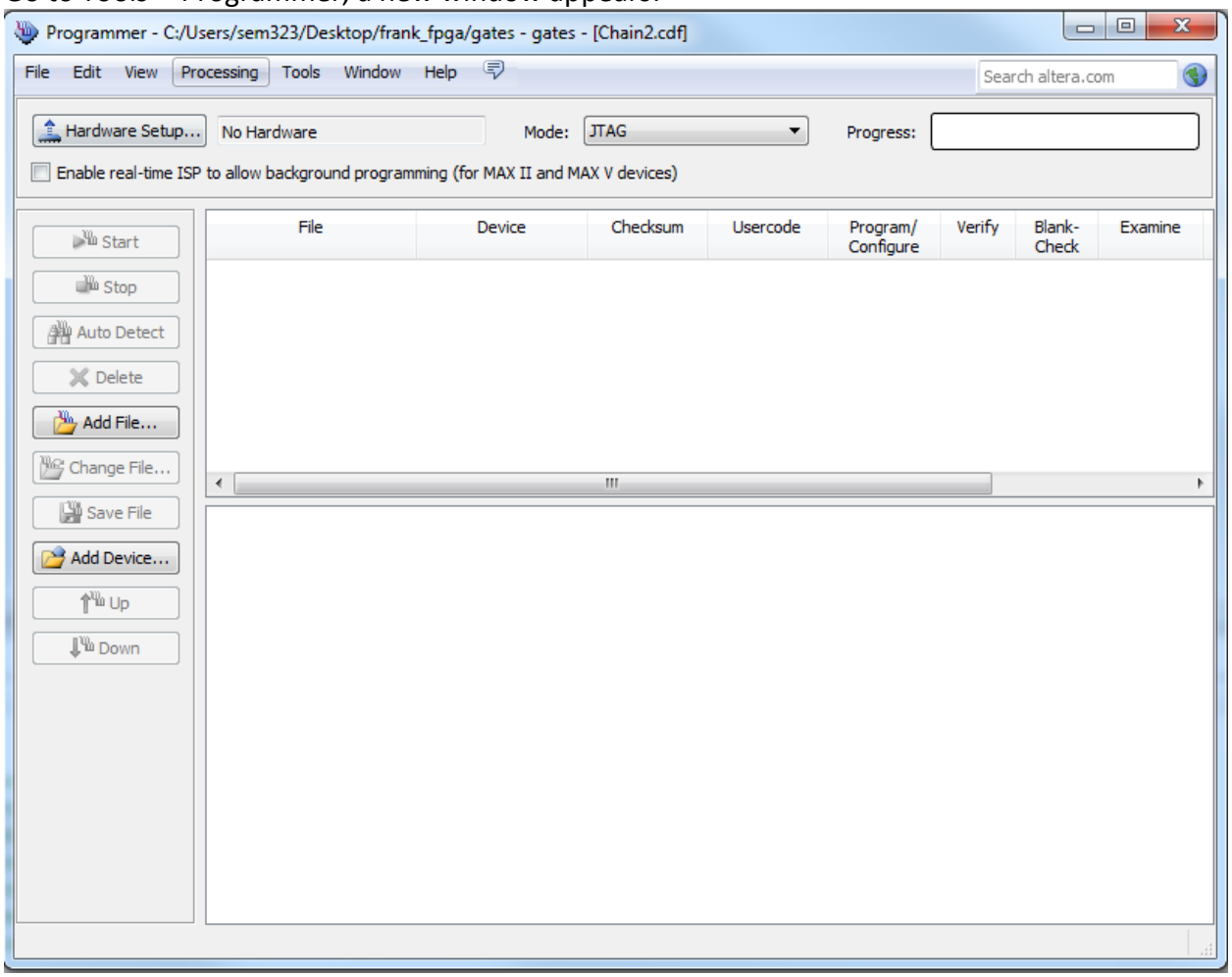
19. Close the Pin Planner window.

Uploading Code to the Board

20. Compile the code again by going to Processing ->Start Compilation. The Message window should look like this when compilation is successful.



21. Go to Tools ->Programmer, a new window appears:

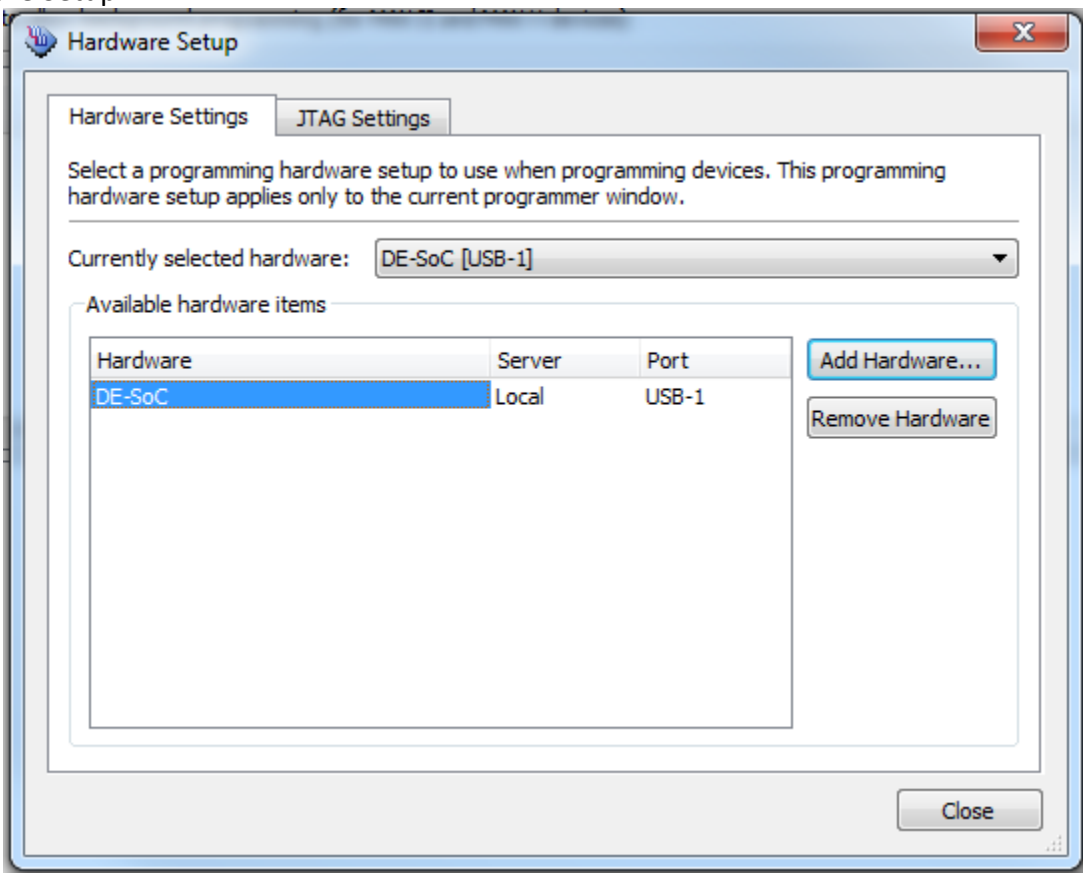


22. Check if the Hardware Setup section is USB-Blaster. If not, install the USB driver as follow:

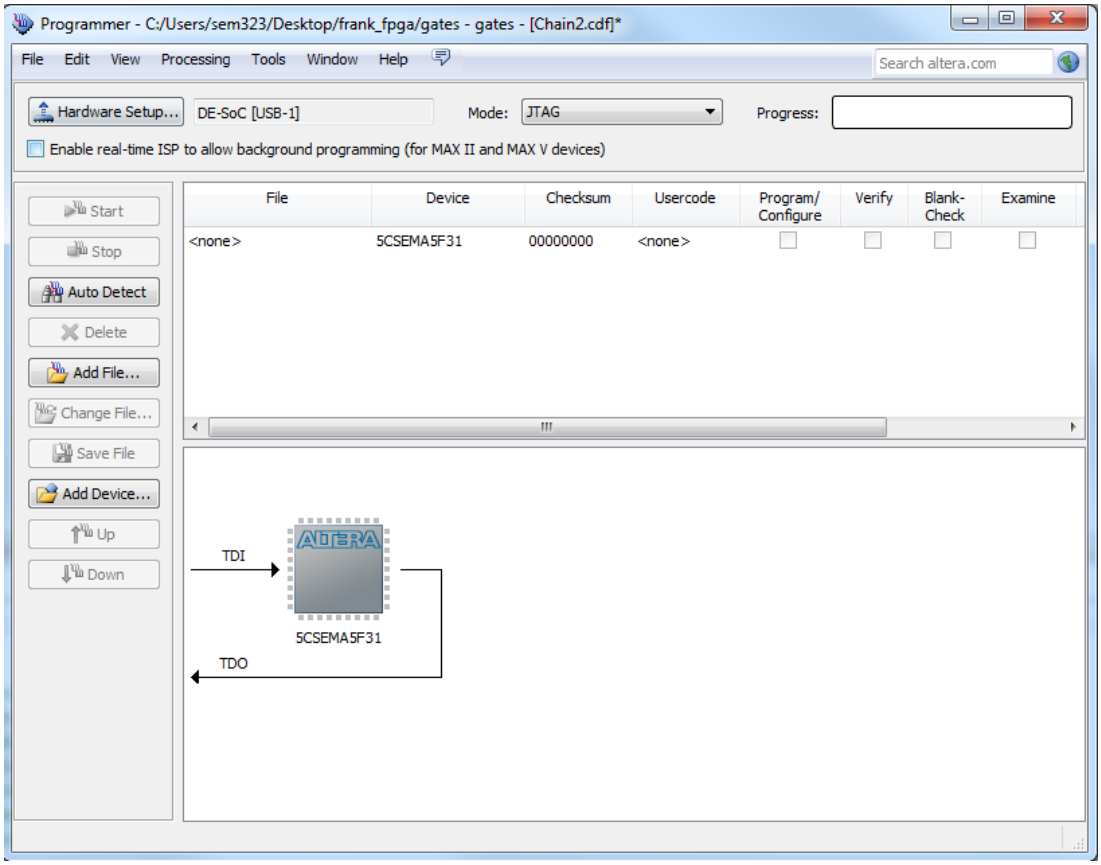
a. Go to device manager, select Other Devices and select the USB-Blaster -> Update Driver Software -> Browse my computer for driver software.

b. Point to the folder: C:\altera\91sp2\quartus\drivers\usb-blaster where C:\altera is the

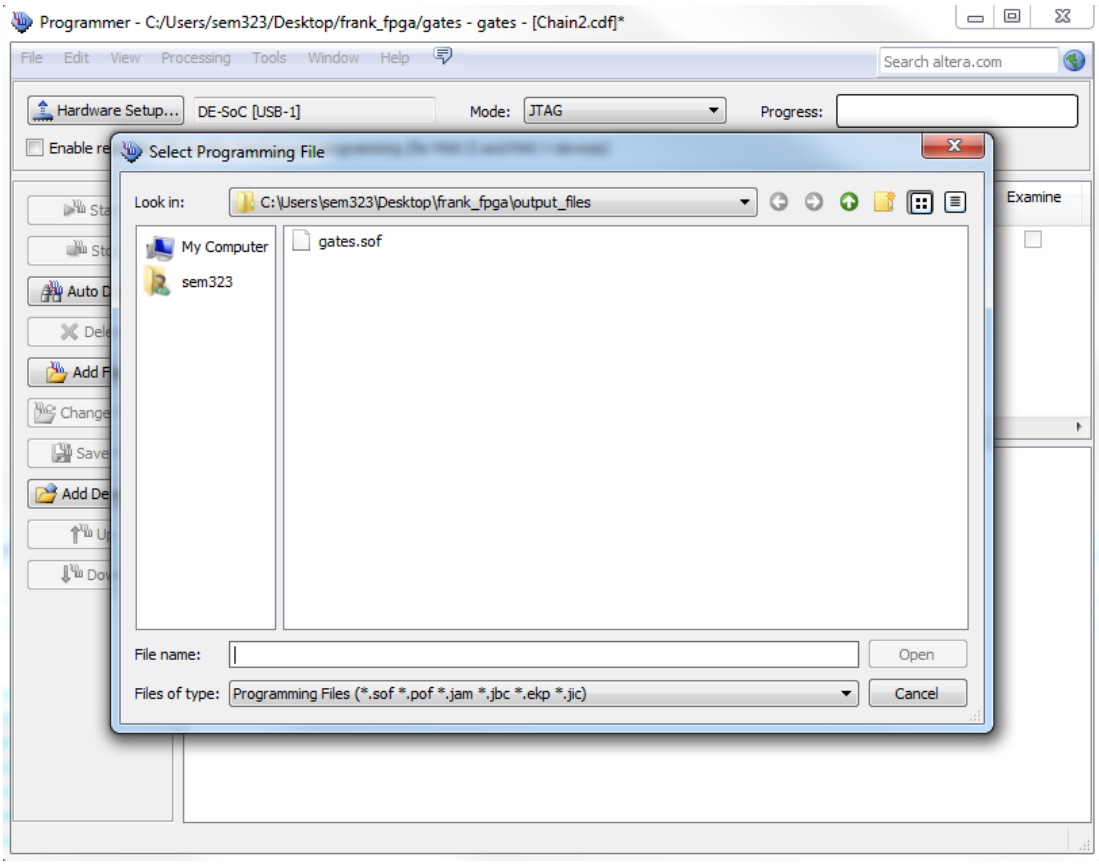
installation folder of Quartus.
c. Click Next -> Done.
d. If the driver is successfully installed, you should be able to select USB-Blaster in Hardware Setup.



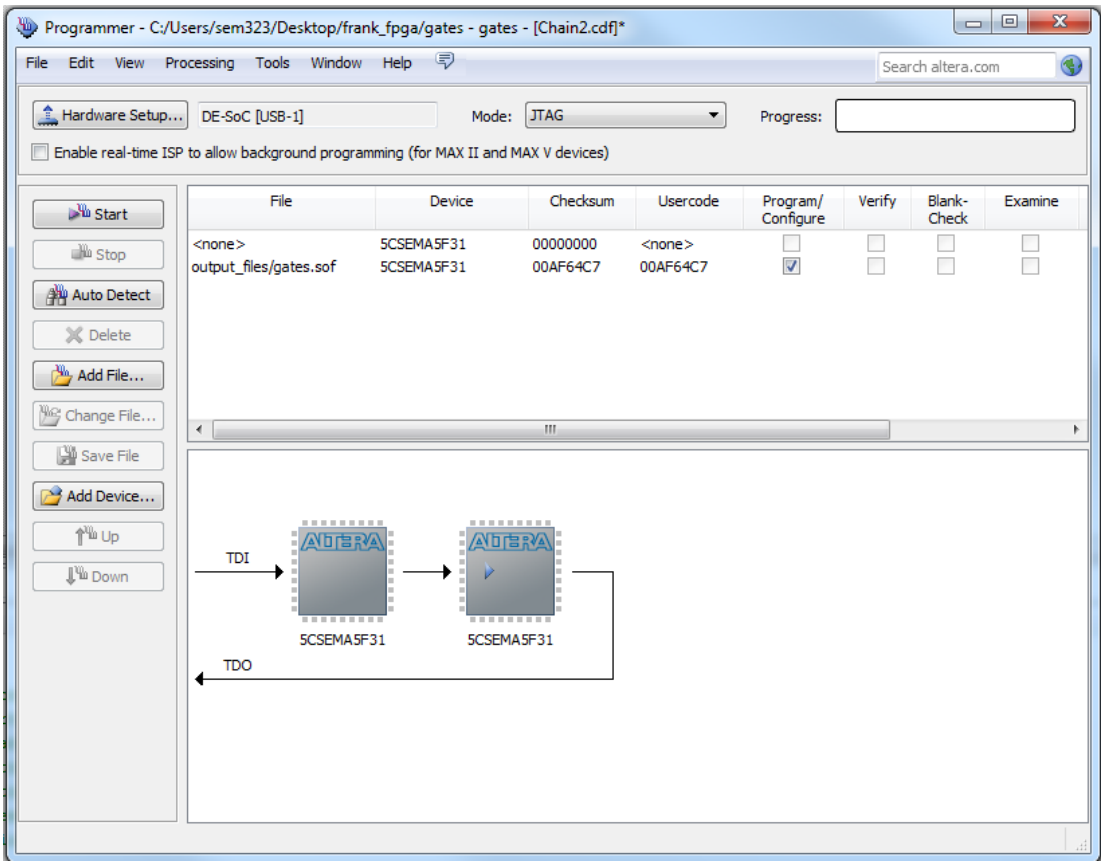
23. Select the Correct Device



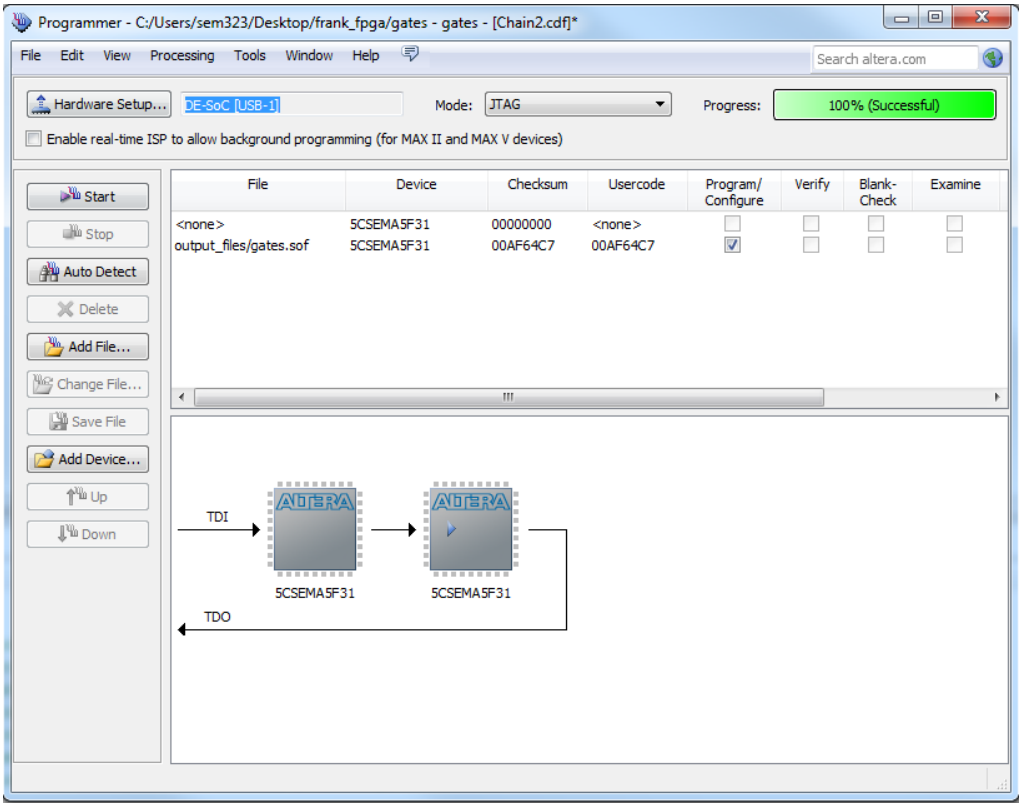
24. Add the programming file, by clicking “Add File”



25. Add the “gates.sof” file



26. Remove any file listings which may have been added by default, they will not have 5CSMA5F31 in the Device column.
27. Click Start



We have programmed the FPGA, verify that the code we programmed is working as expected.