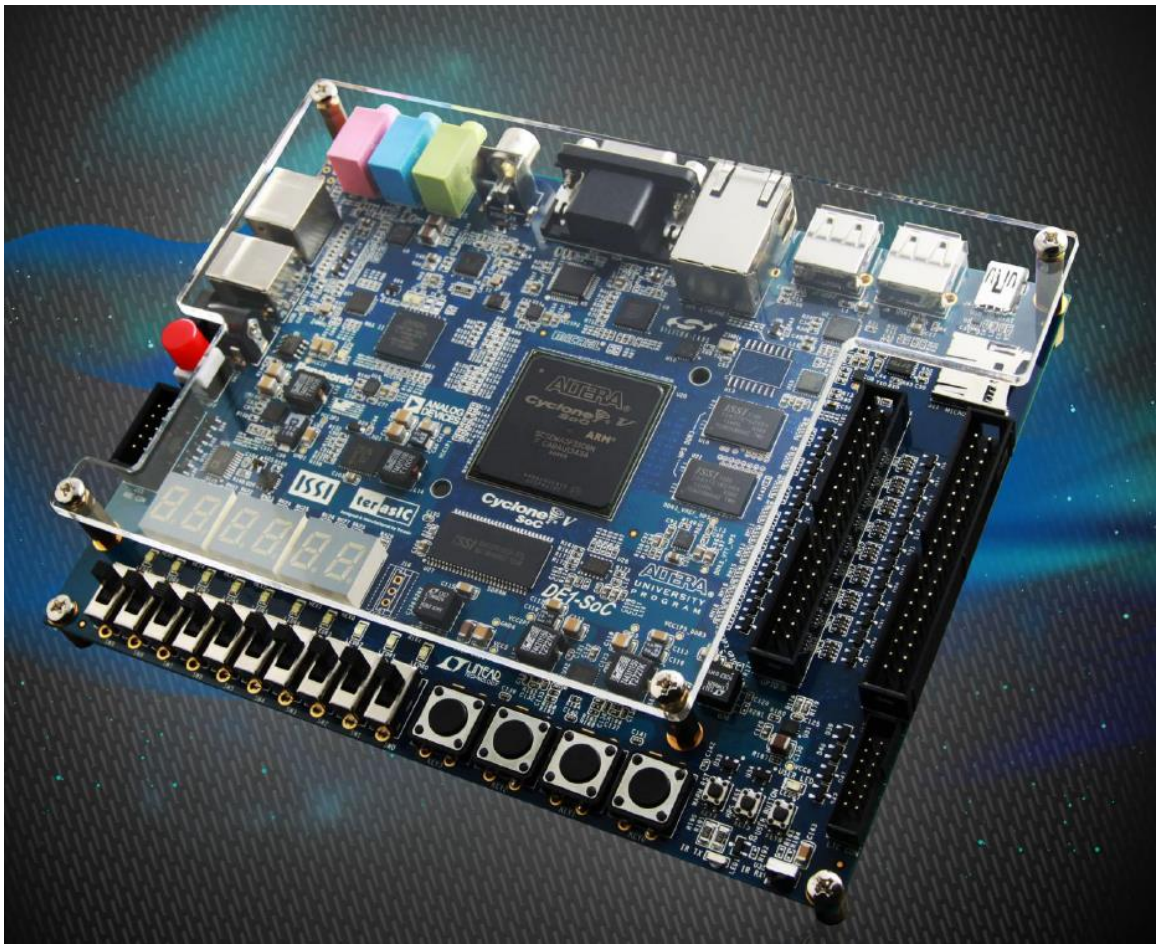



# Computer Design Lab Manual- 3710

COMPREHENSIVE TUTORIAL FOR SETTING UP DE1-SOC HW/SW PLATFORM

Instructor – Priyank Kalla
Document Author - Vikas Rao

# Objective –

This tutorial provides comprehensive information that will help you understand how to create a FPGA design and run it on your DE1-SoC development board. The following sections provide a quick overview of the design flow, explain what you need to get started, and describe what you will learn.

The standard FPGA design flow starts with design entry using schematics or a hardware description language (HDL), such as Verilog HDL or VHDL. In this step, you can create a digital circuit that is implemented inside the FPGA. The flow then proceeds through compilation, simulation, programming, and verification in the FPGA hardware.

This tutorial will not make you an expert, but at the end, you will understand basic concepts about Quartus Lite projects, such as entering a design using a schematic editor and HDL, compiling your design, and downloading it into the FPGA on your DE1-SoC development board.

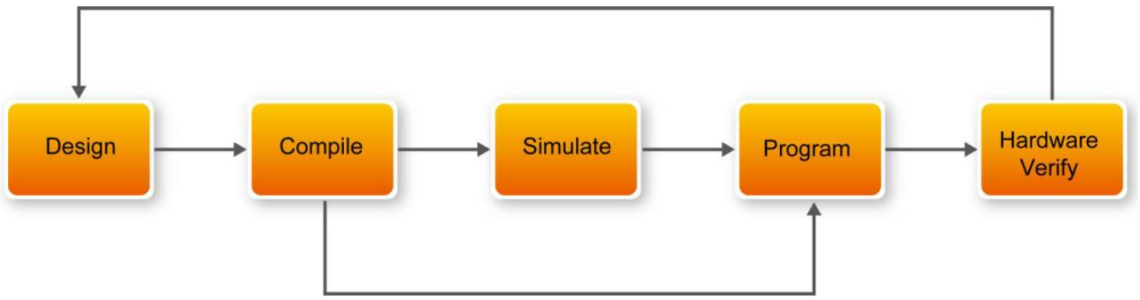


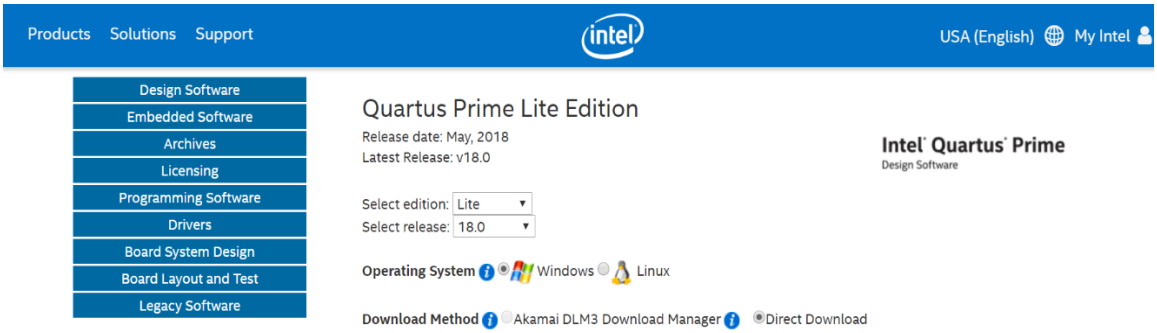

**Figure 1-1 Design Flow**

# Index -

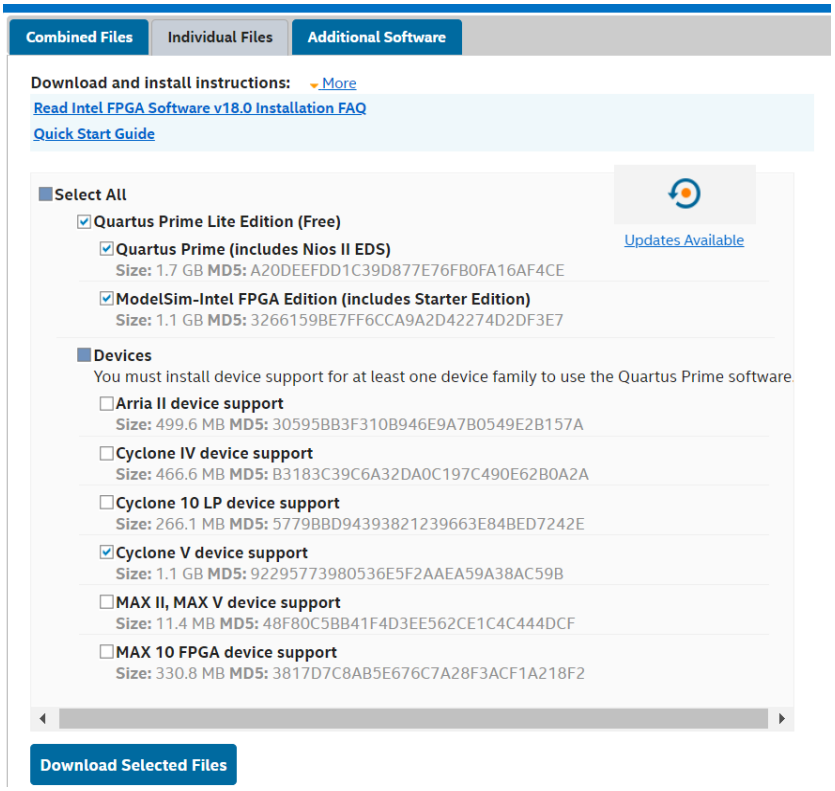

- Section 1 – Download and install Altera Quartus Lite Software on Windows/Linux
- Section 2 – Setup a new project in Quartus Lite Software
- Section 3 – Implement a function using Verilog HDL
- Section 4 - Simulate the Verilog circuit using ModelSim + Verilog test bench
- Section 5 – Pin assignment and Constraint generation
- Section 6 - Synthesize, Implement, Generate, and Program for DE1-SoC board

# Section 1 - Download and install Altera Quartus Lite Software

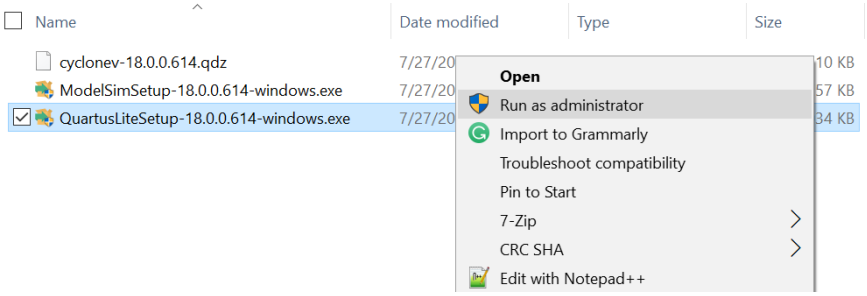
1. Browse to the hyperlink provided below and make sure that Lite edition and 18.0 release are selected. Also, check the appropriate OS box(Windows/Linux). Use direct download method and make sure to download only the individual files marked in the next bullet. Create a basic account on Intel using UnID if prompted.
   http://dl.altera.com/18.0/?edition=lite



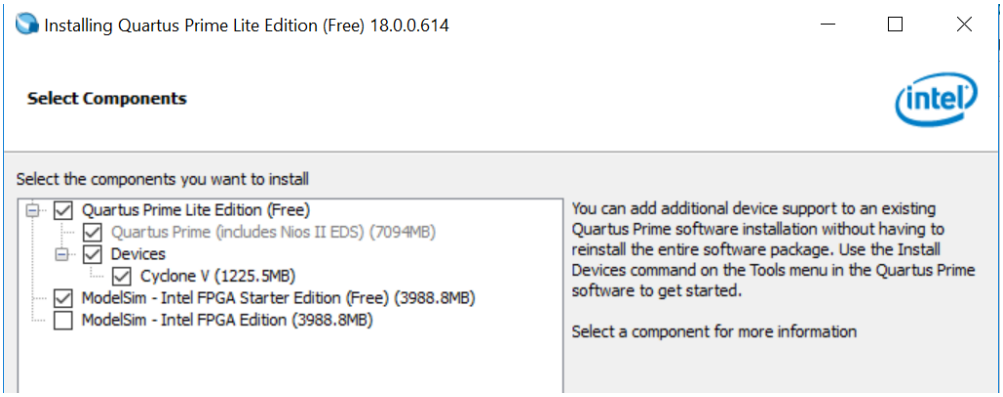

2. Navigate to the Individual files tab and select the Quartus prime lite edition, ModelSim FPGA edition along with cyclone V device support and download the selected files.



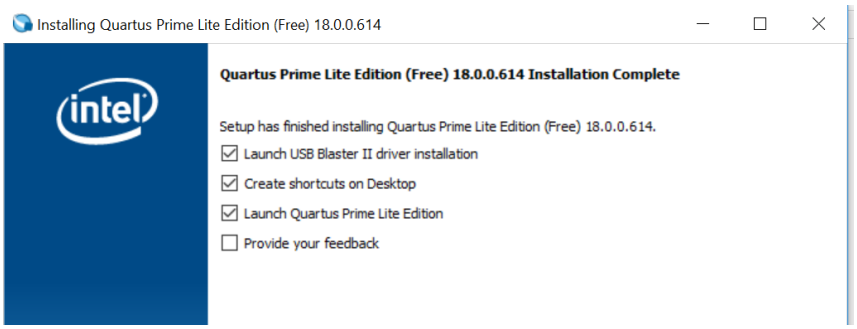
3. Select the QuartusLite setup file and Run as administrator for windows installation. For Linux, modify the *.run file with executable permissions(chmod +x *.run) and execute the *.run on terminal as super user(sudo \.*.run).




4. Press Yes and click Next in the QuartusLite setup installation prompt. Accept the agreement and click Next again. Select the appropriate installation folder or retain the default directory. Make sure the following boxes are marked(modelsim and cyclone V device support) for installation. And Press Next and let the installation complete



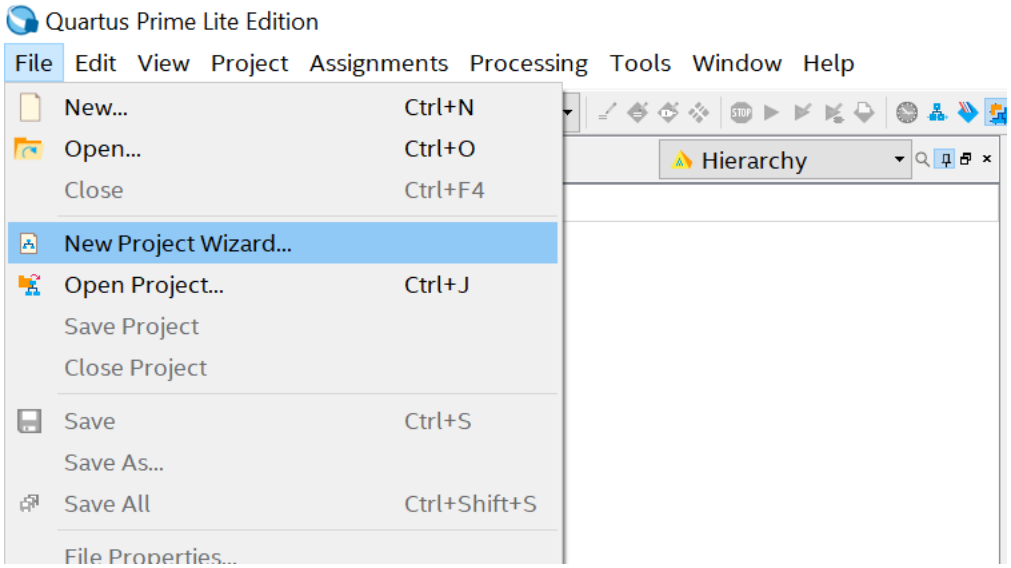

5. The setup should also install ModelSim FPGA edition along with QuartusLite Software. After the software installation, the setup will prompt for installation of USB Blaster 2 device driver which is required for FPGA programming. Click Next and finish the installation.



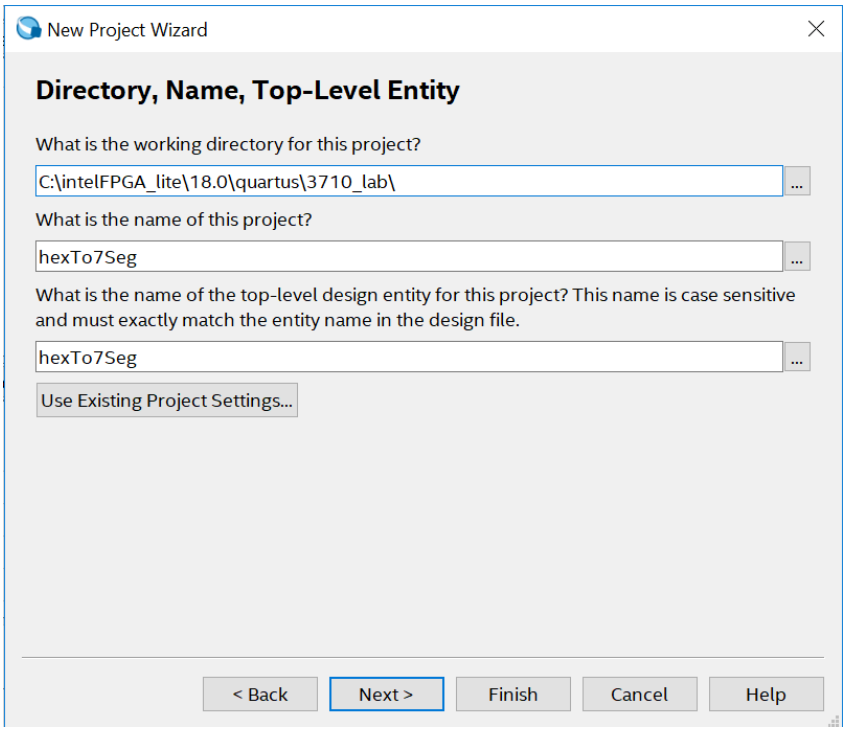
# Section 2- Setup a new project in Quartus Lite Software

You begin this section by creating a new Quartus II project. A project is a set of files that maintain information about your FPGA design. The Quartus II Settings File (.qsf) and Quartus II Project File (.qpf) files are the primary files in a Quartus II project. To compile a design or make pin assignments, you must first create a project.

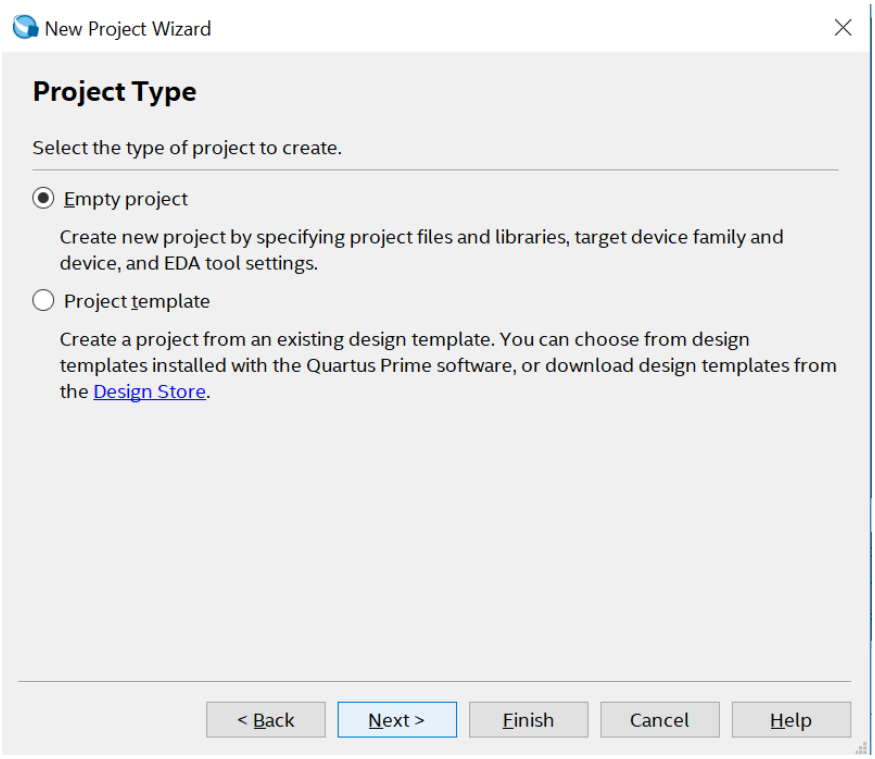
1. Launch the Quartus II software, select File > New Project Wizard. The Introduction page opens



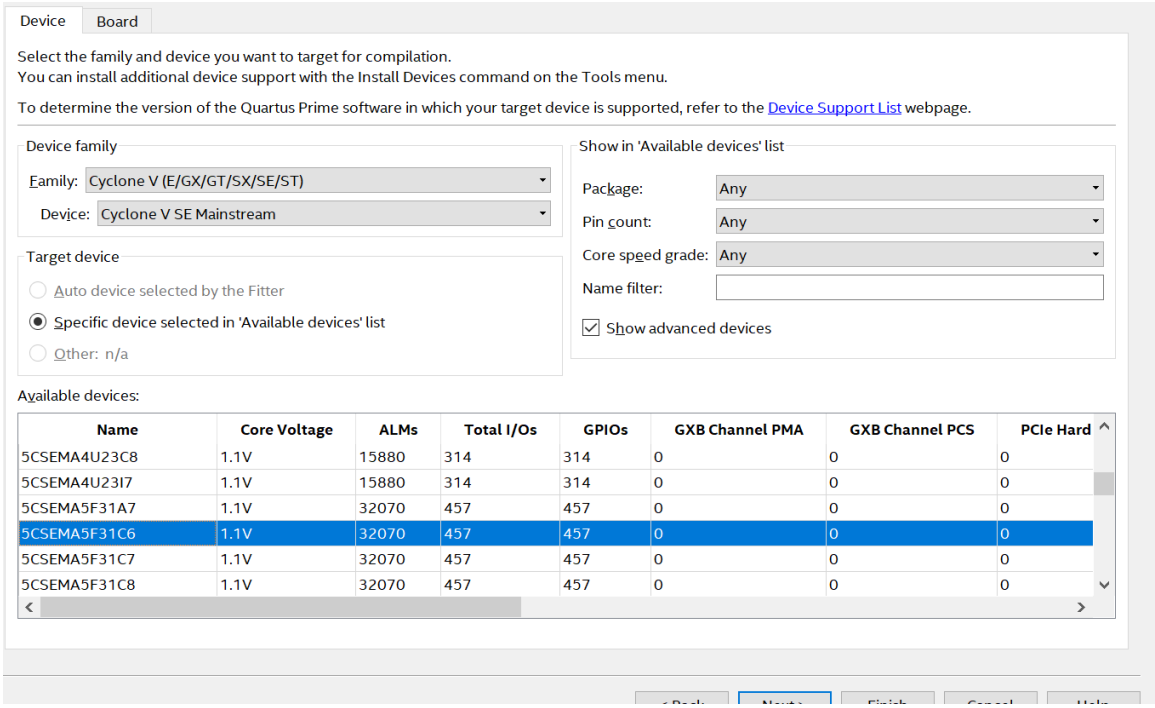

2. Enter the following information about your project as shown in the next snapshot:
    a. What is the working directory for this project? Enter a directory in which you will store your Quartus II project files for this design.
    b. For example, C:\intelFPGA_lite\18.0\quartus\3710_lab\
    c. File names, project names, and directories in the Quartus II software cannot contain spaces.
    d. What is the name of this project? Type hexTo7Seg.
    e. What is the name of the top-level design entity for this project? Type hexTo7Seg.
    f. Create a directory if it isn't present.



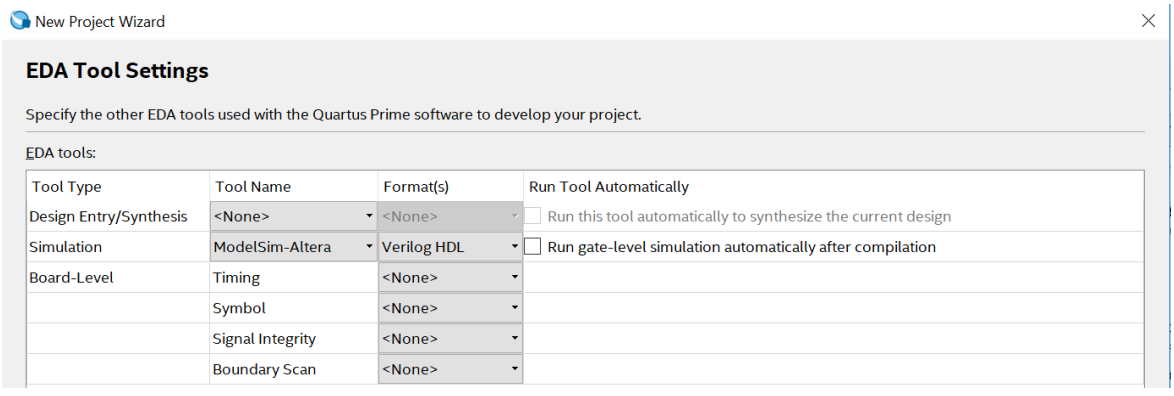

3. Select the Empty project template and click Next:



4. Click Next twice and navigate to the Family, Device and Board Settings. Select the specifics on the respective targets as mentioned below. You can add the required Verilog files later or edit the files in the Quartus editor once the project has been setup.




5. Select the simulation Tool as ModelSim-Altera(note – this is not just ModelSim but the one with Altera) and format as Verilog HDL. Retain the other Tool selections to default.



6. Review the new project summary page and make sure all the specifics that you entered are reflected in it.

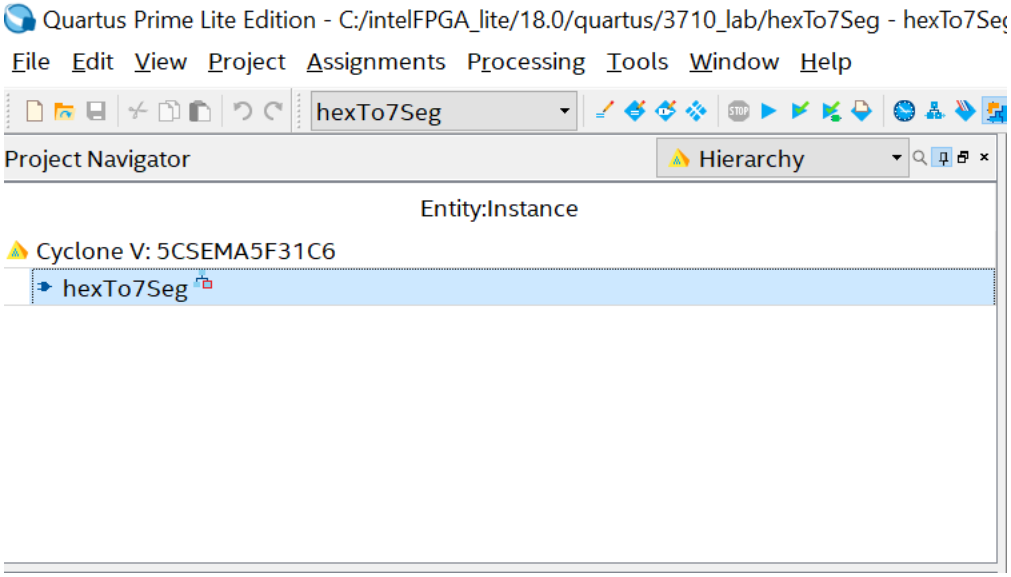
New Project Wizard

**Summary**

When you click Finish, the project will be created with the following settings:

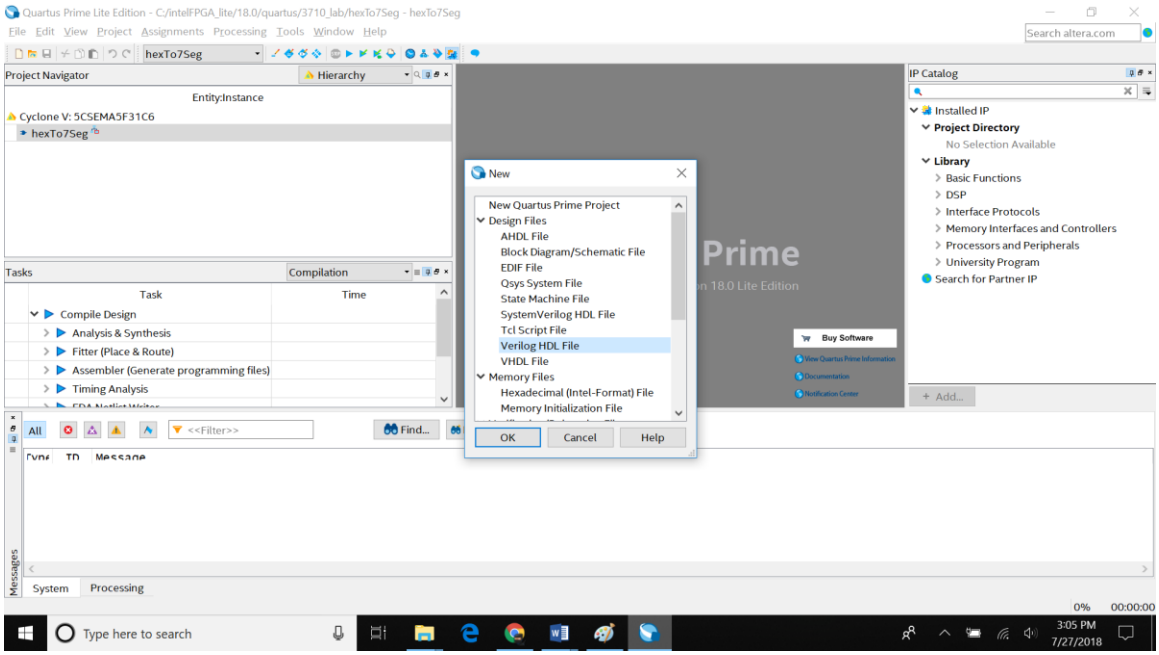
| | |
|---|---|
| Project directory: | C:\intelFPGA_lite\18.0\quartus\3710_lab\ |
| Project name: | hexTo7Seg |
| Top-level design entity: | hexTo7Seg |
| Number of files added: | 2 |
| Number of user libraries added: | 0 |
| Device assignments: | |
| Design template: | n/a |
| Family name: | Cyclone V (E/GX/GT/SX/SE/ST) |
| Device: | 5CSEMA5F31C6 |
| Board: | n/a |
| EDA tools: | |
| Design entry/synthesis: | <None> (<None>) |
| Simulation: | ModelSim-Altera (Verilog HDL) |
| Timing analysis: | () |
| Operating conditions: | |
| Core voltage: | 1.1V |
| Junction temperature range: | 0-85 °C |

7. Quartus Lite Project navigator will now reflect the instance as your top level design entity under the device name.

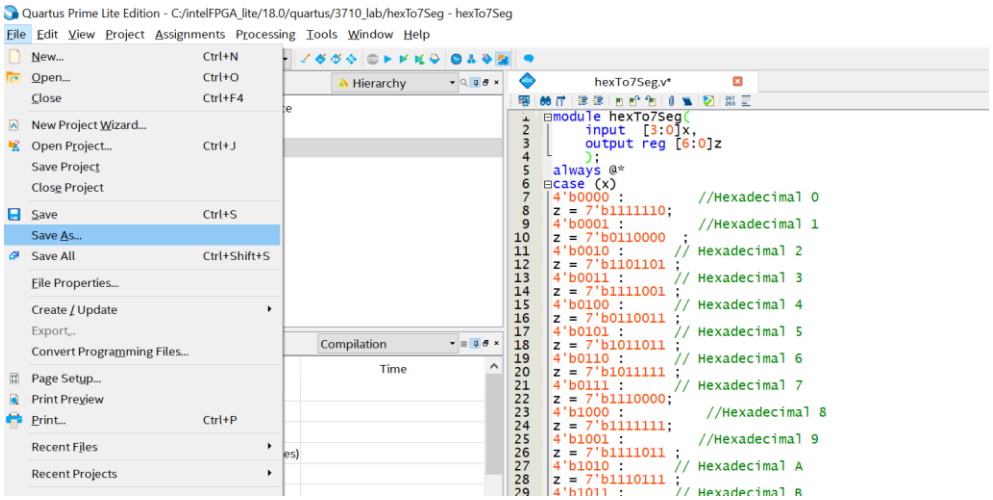


# Section 3 – Implement a function using Verilog HDL

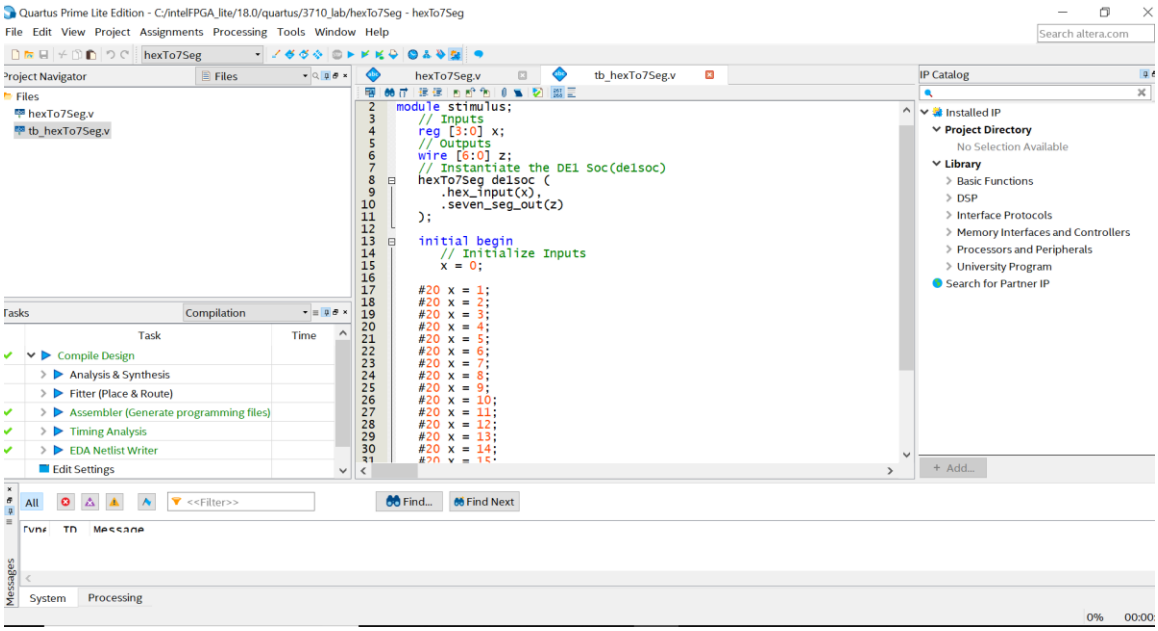
1. Once you have created and setup your project, the next step is to describe your design using a hardware description language.
   Go to File-->New→Select Verilog HDL File.




2. Once a new file is created, you will be able to enter your code into it. When you save it, you have to have the module name and file name match up. Remember that one of the file names must match the name of the project. Double check and make sure that the Save as type is Verilog HDL Files.

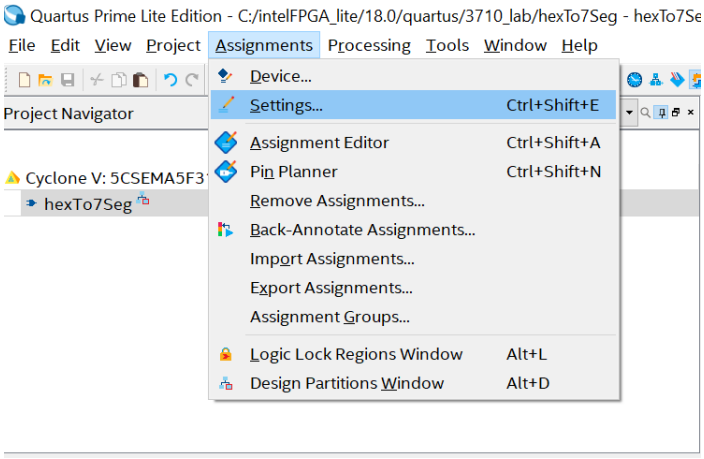


3. Every project in hardware needs a testbench to generate all necessary inputs and read outputs to ensure they are correct. This is very similar to writing test cases in software programming. The standard practice of naming a testbench is to add a "tb_" in front of the name of the module you are testing. In this case, we are testing hex to seven segment display, so the name of the new Verilog HDL file is saved as "tb_hexTo7Seg". A testbench is just another standard Verilog HDL File.

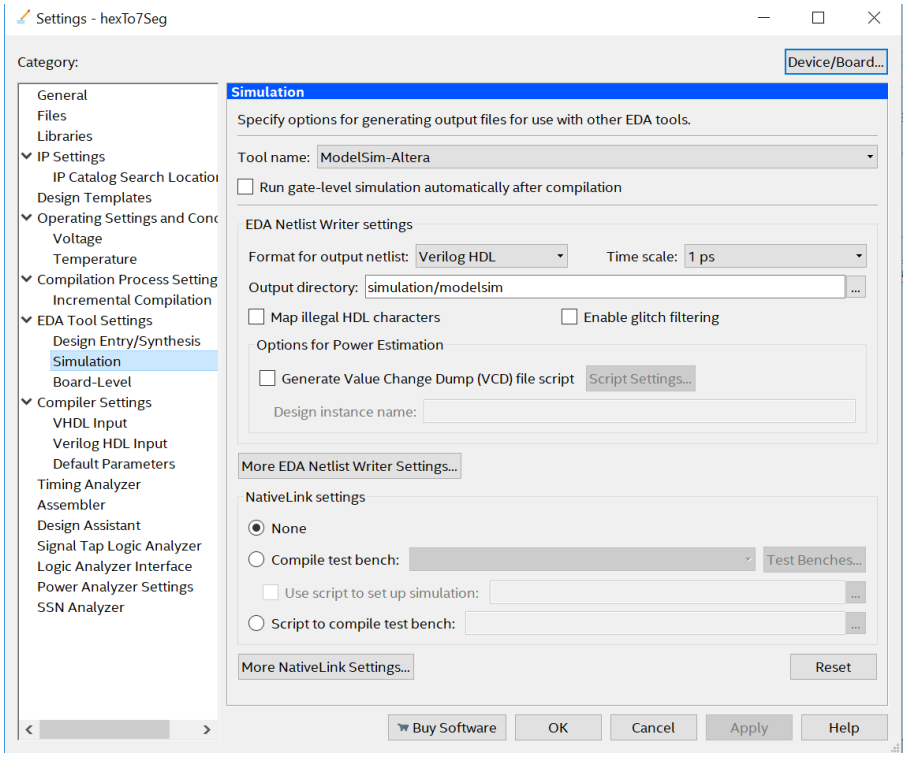


# Section 4 - Simulate the Verilog circuit using ModelSim + Verilog test bench

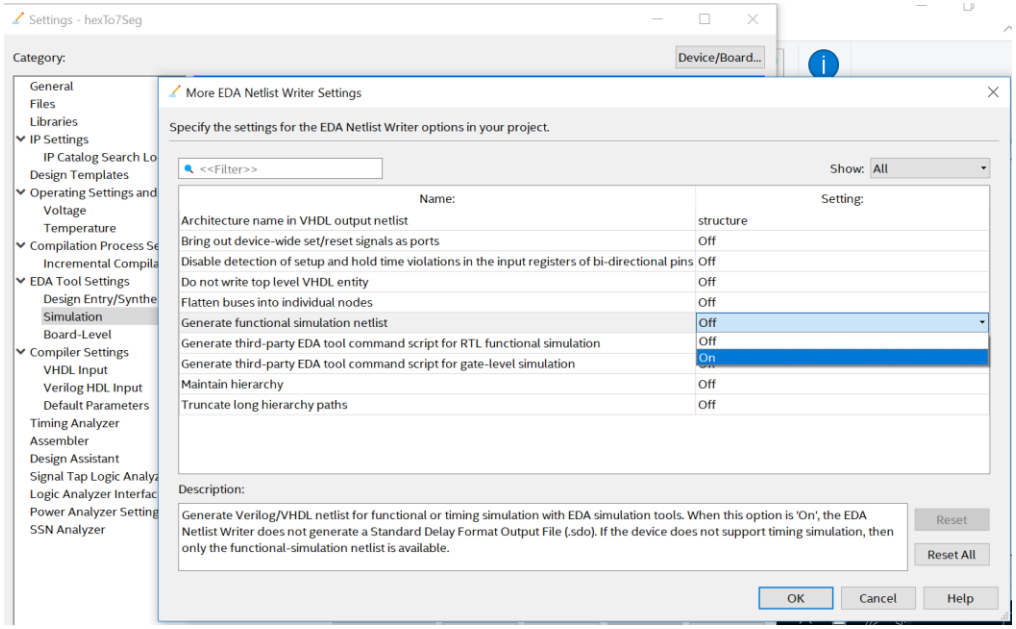
1. To setup the simulation Tool path for ModelSim-Altera, navigate to Assignments→ Settings and modify the simulation parameters as shown below.



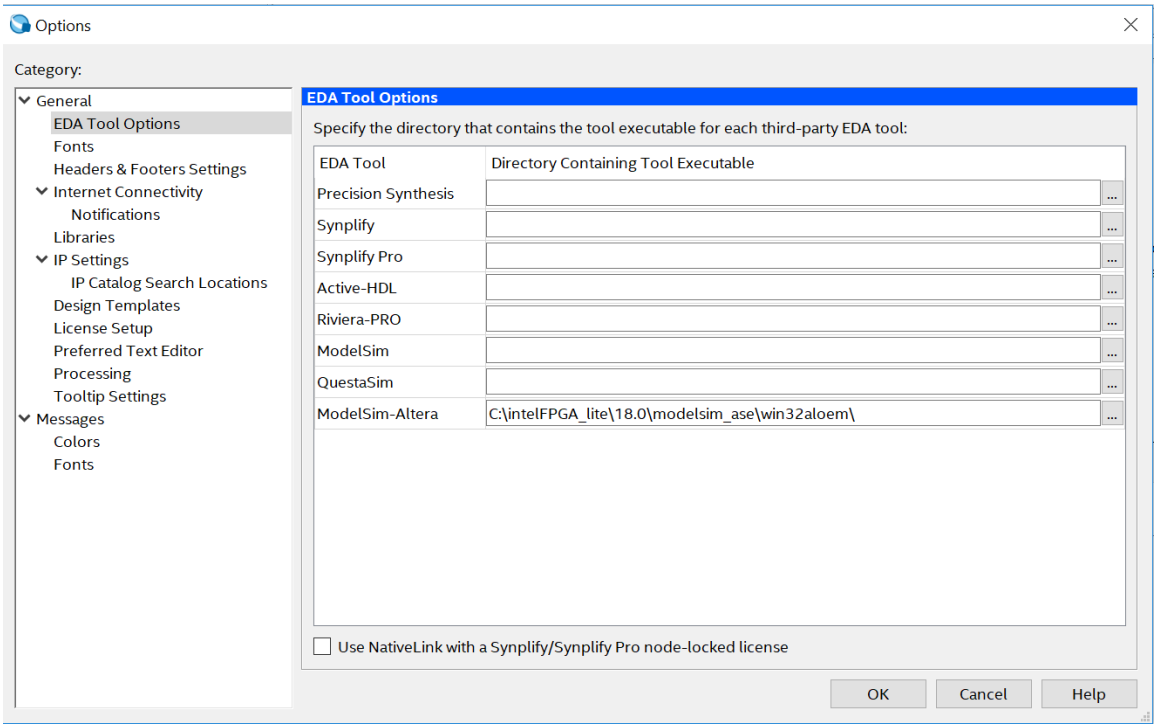

2. Under EDA Tool settings, check on Simulation option and it should show the Tool name as ModelSim-Altera.



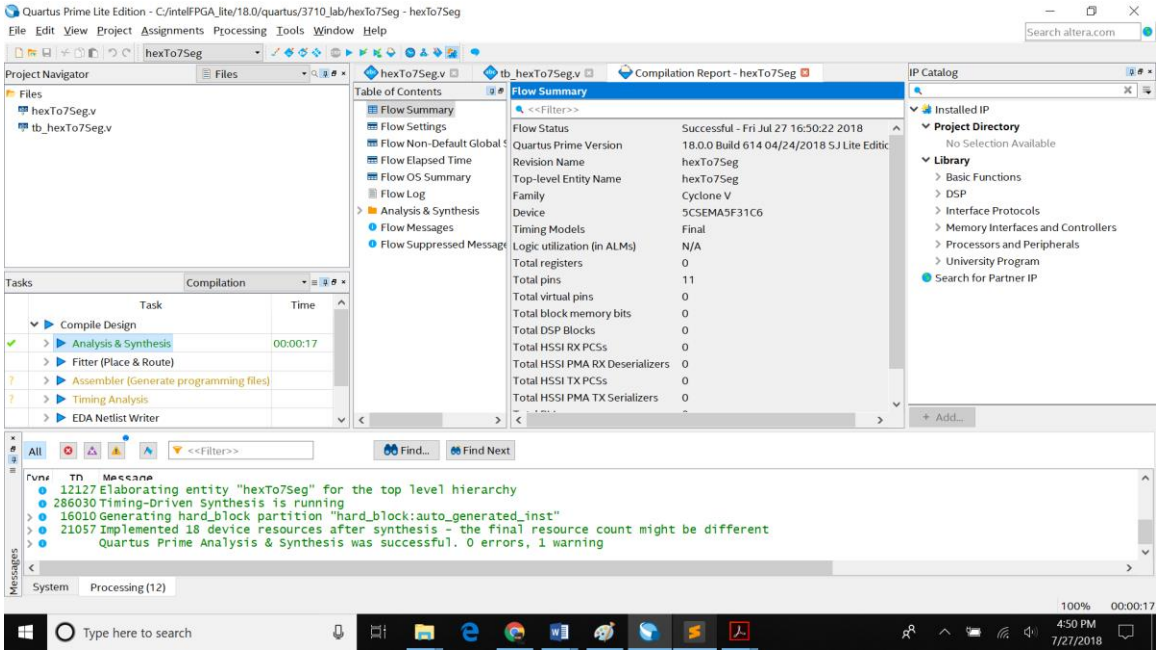
3. Navigate to More EDA Netlist Writer settings, and set the generate functional simulation netlist option to ON and click OK and Apply.



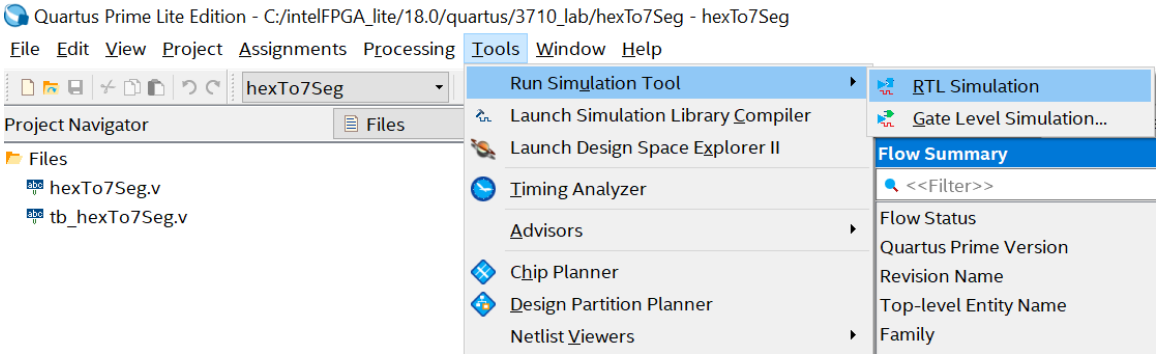

4. To setup the simulation executable path, click on Tools> Options> EDA Tool Options and and set the Modelsim-Altera path to your installation directory.



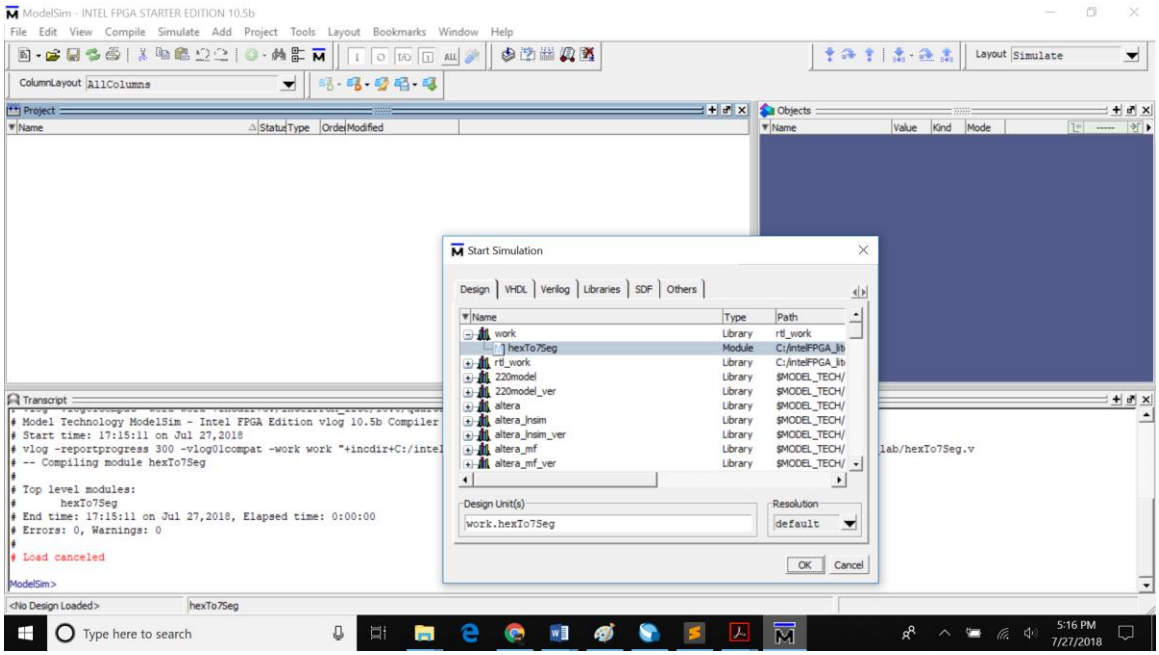
5. After you have finished coding up your modules, double click on Analysis & Synthesis under the Tasks pane. If you do not see Analysis & Synthesis, double check that Flow is set to Compilation. This will compile and synthesize your program(s). If there are no errors, you will see a pop-up saying the Analysis & Compilation was successful. If not, it will tell you your errors in the messages pane at the bottom of the screen. It is good habit to just review your warnings (if any) to ensure you have no latches or other design hazards.



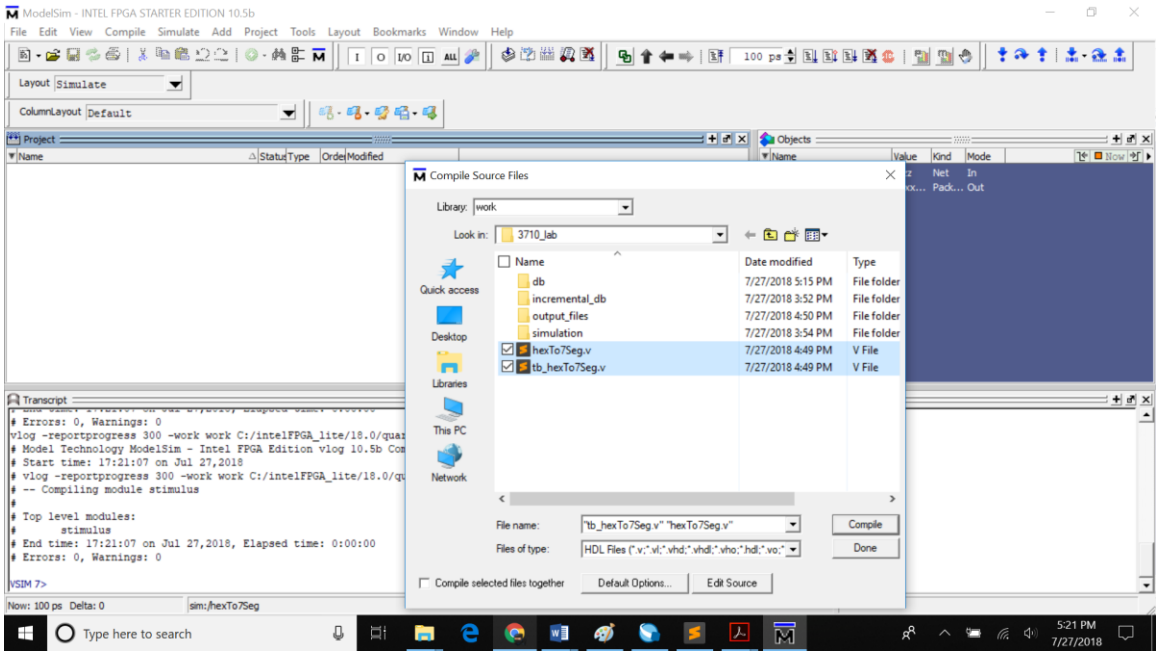

6. Once the Analysis & Synthesis is successful, you can do a RTL Simulation. Go to Tools-->Run Simulation Tool-->RTL Simulation (this will launch ModelSim).



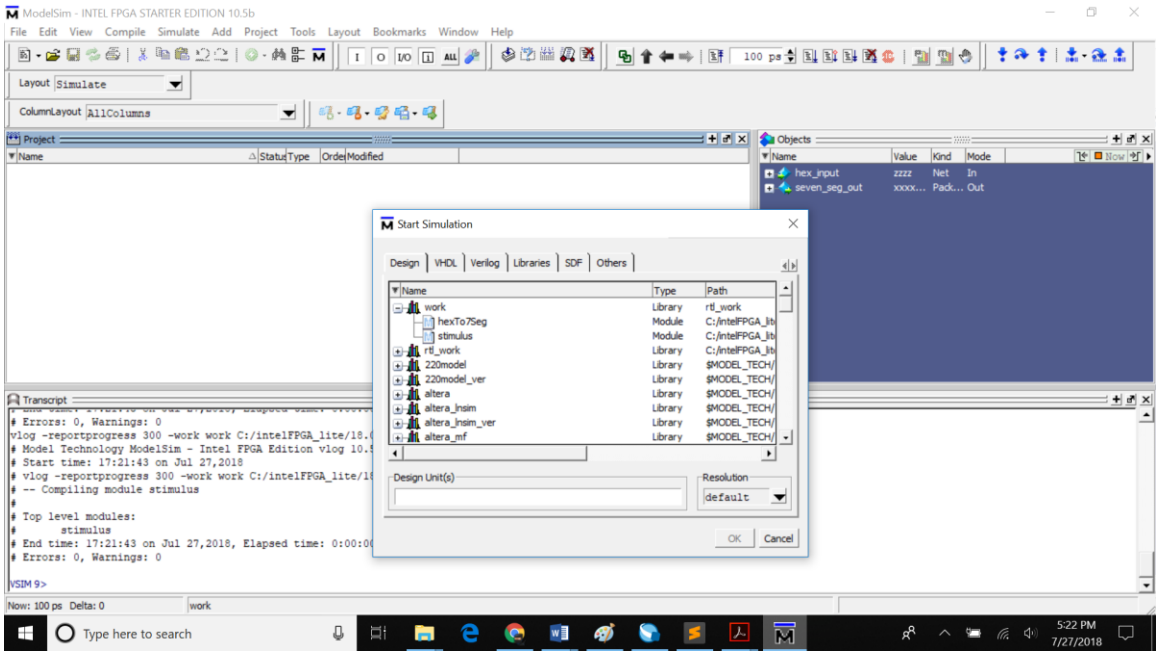
7. The Library you will be working in is called **work**. Do not bother creating another library as it will cause complications when you try to simulate your program again after closing ModelSim. Expanding the work library will show at least one of the files in your project.



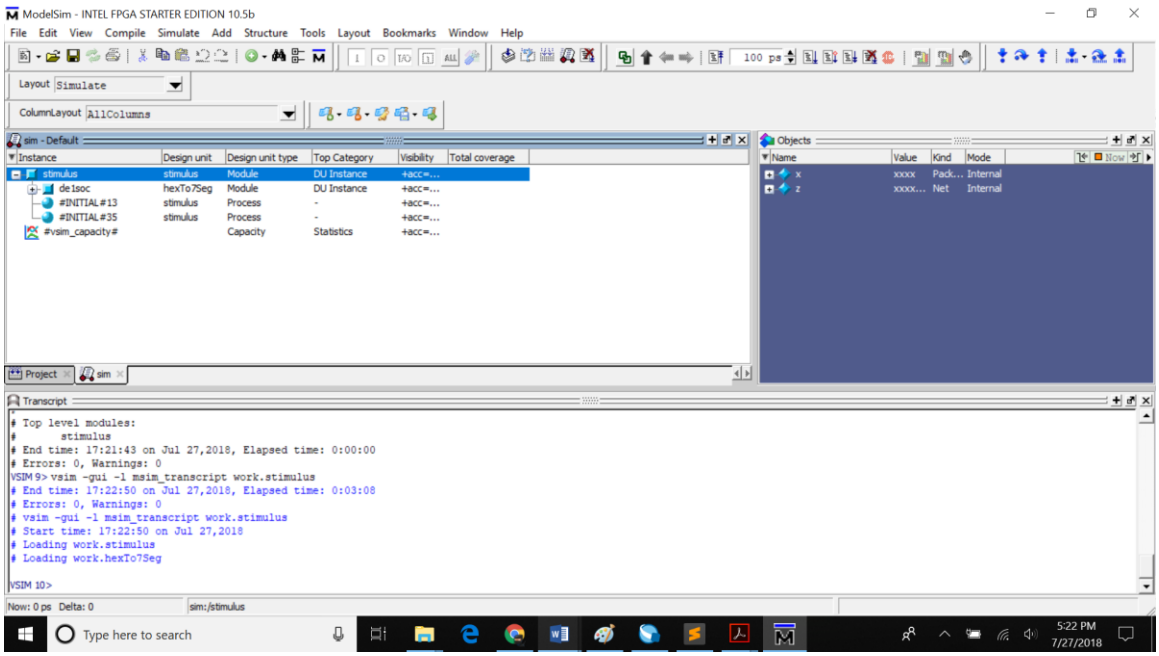

8. Go to Compile-->Compile. Ensure that the Library is work and navigate to your project directory. Select all Verilog HDL Files that pertain to your project. This includes the testbench. Hit Compile and then Done.



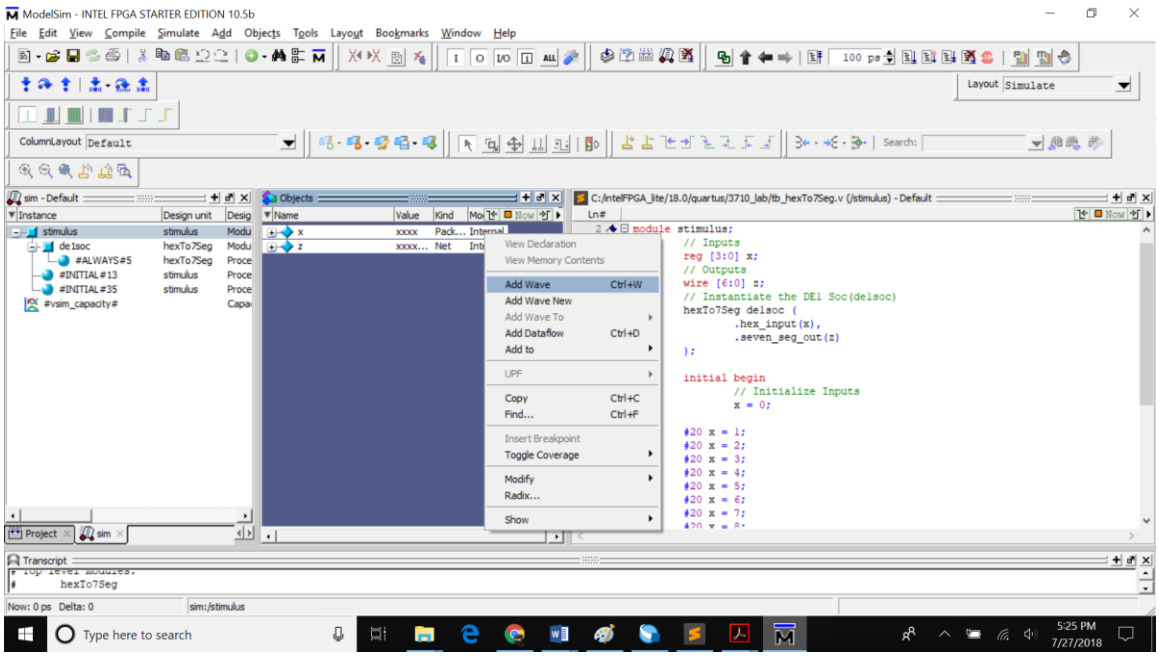
9. If you look at the bottom in the Transcript pane, you will see that it did compile and there were no errors. The work library should have all the files you just compiled. If not, repeat the previous step. Since the testbench does the signal generations and testing, double click on your testbench file.



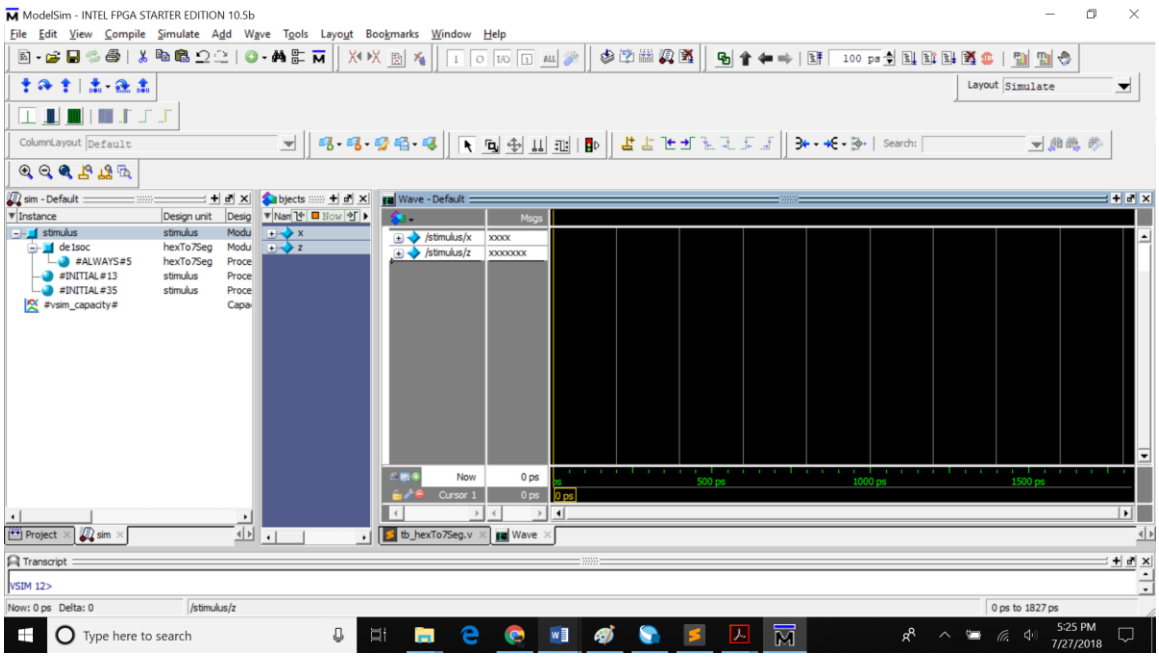

10. Click Simulate> Start simulation . Under work select the module to be simulated and click ok.



11. The next step is to add signals to the wave and show the wave if it is not already present. On the left, in the sim pane, right click on the testbench file which should be the top most file. Go to Add->To Wave->All signals in region.

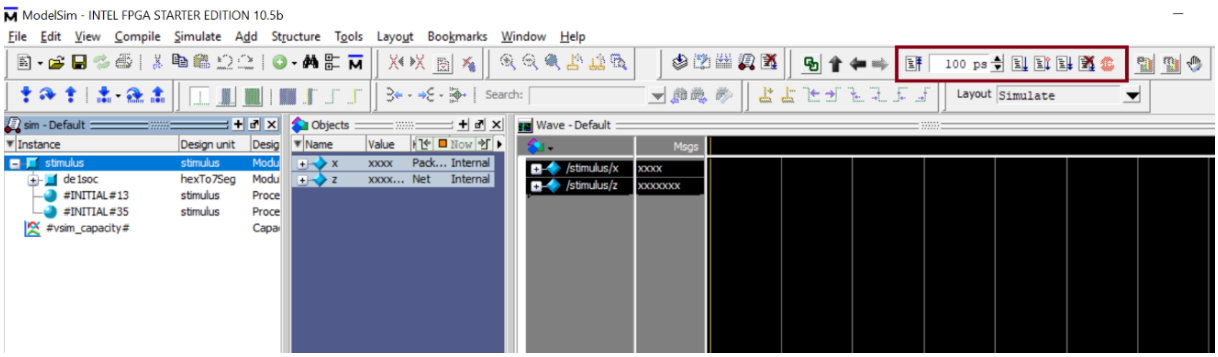



12. In the Wave pane, you will see all the signals declared and used in the testbench.

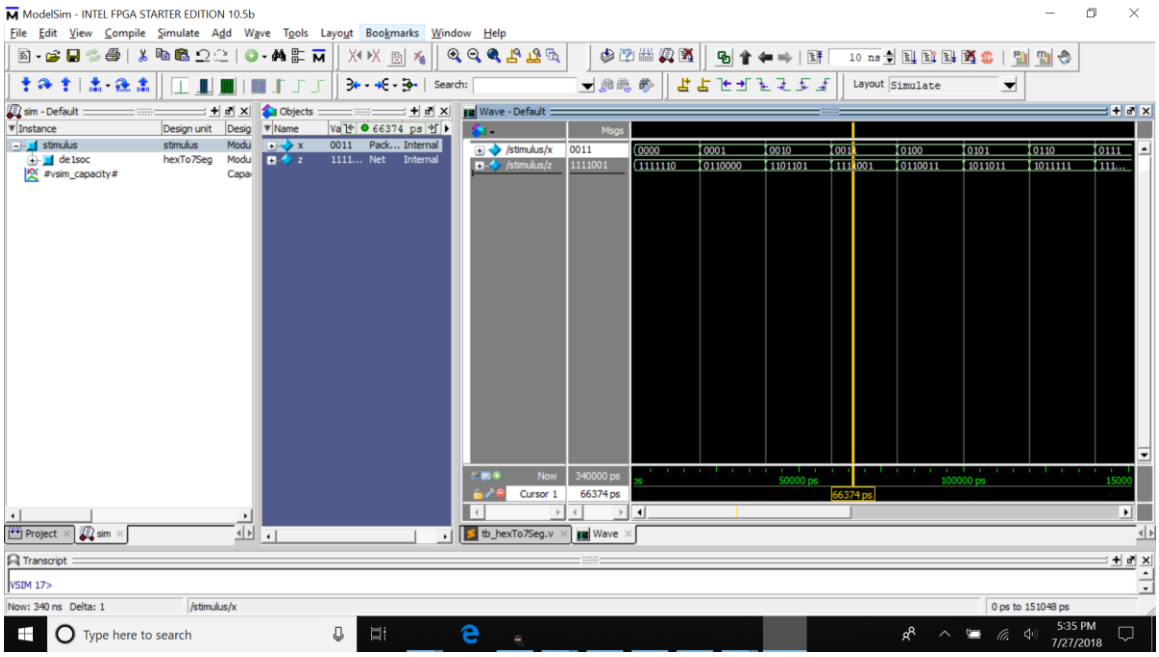


Note - When dealing with signals that are many bits, it is easier to see its value as an unsigned integer/Hex rather than binary. To make this conversion, right click on the signal you want, go to Radix and choose the format you want.

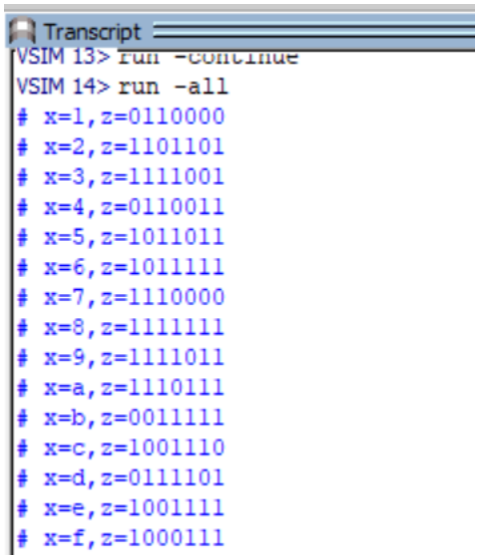
13. You are now ready to simulate your program. The icons boxed in the below screenshot are used to run the testbench. The first icon is Restart which will reset the simulation as if you never ran it. This is helpful to rerun the simulation without recompiling everything. The Run Length allows you to enter a specific amount of time you want the program to run for. It defaults to pico-seconds, but nano-seconds is the best time to use. The icon Run right after the Run Length is to run your program for the amount of time specified in the Run Length. If you set Run Length to be 10 ns, each time you press Run, the program will continue for 10 ns. Continue Run will run the program until it terminates. The same is true for Run -All. All the programs in this class will terminate in less than one second. If you find yourself waiting for longer than a few seconds until the program terminates, hit the Stop button and recheck your logic. you will see the following screen once your program terminates. It shows you where the program terminated. To go back to the Wave, click on the Wave tab.



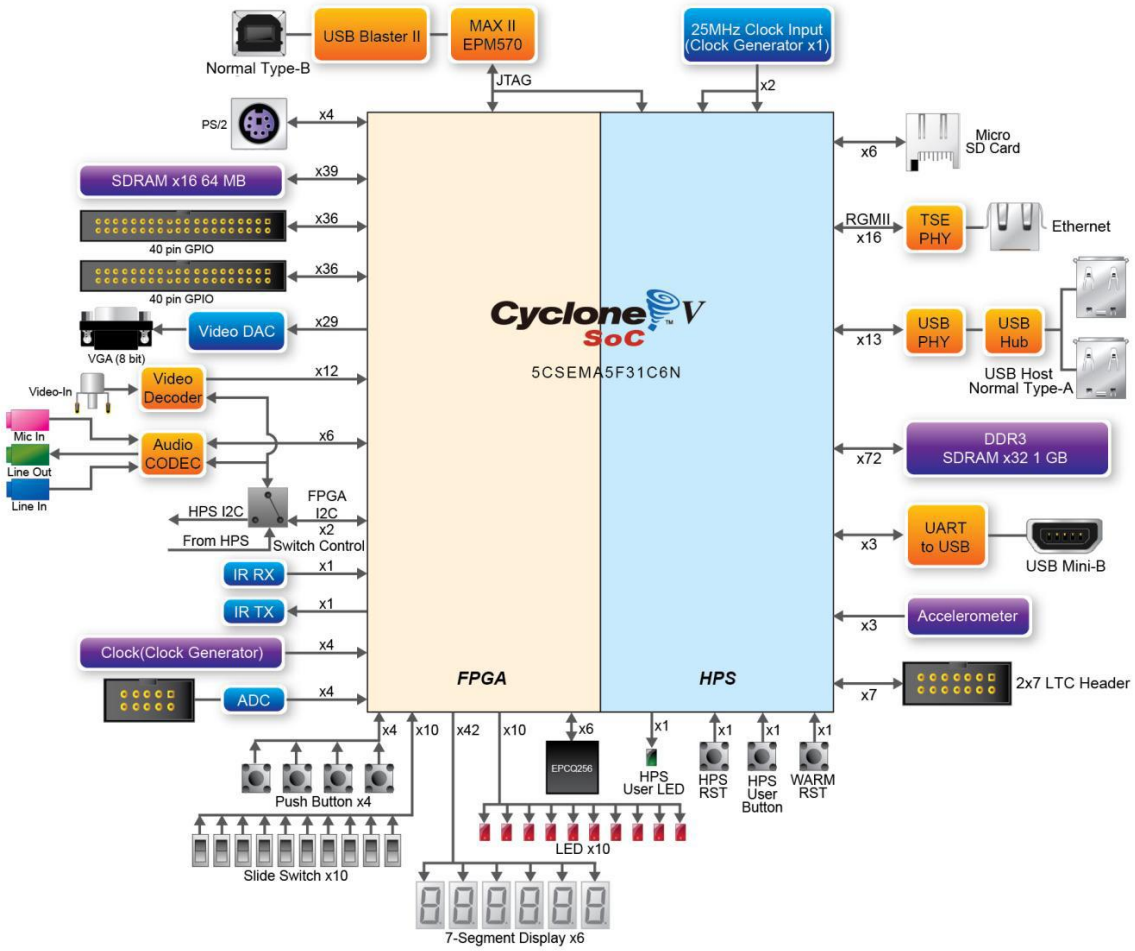
14. Once you click Run, you should see something like this on your Wave with values propagated and reflected on all the intended signals in the wave.

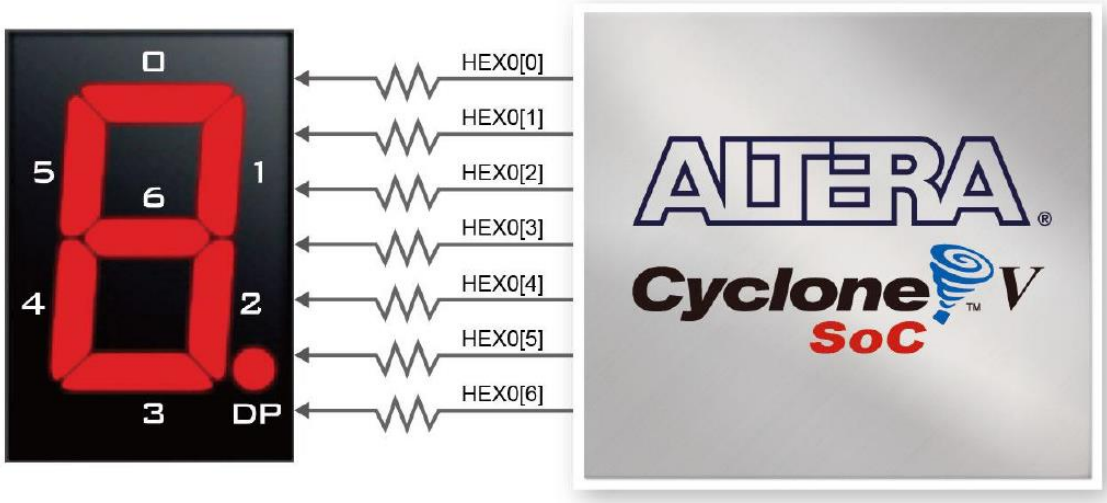



15. If you expand or scroll through the Transcript pane, you will see the output of any $display statements you have in your code.



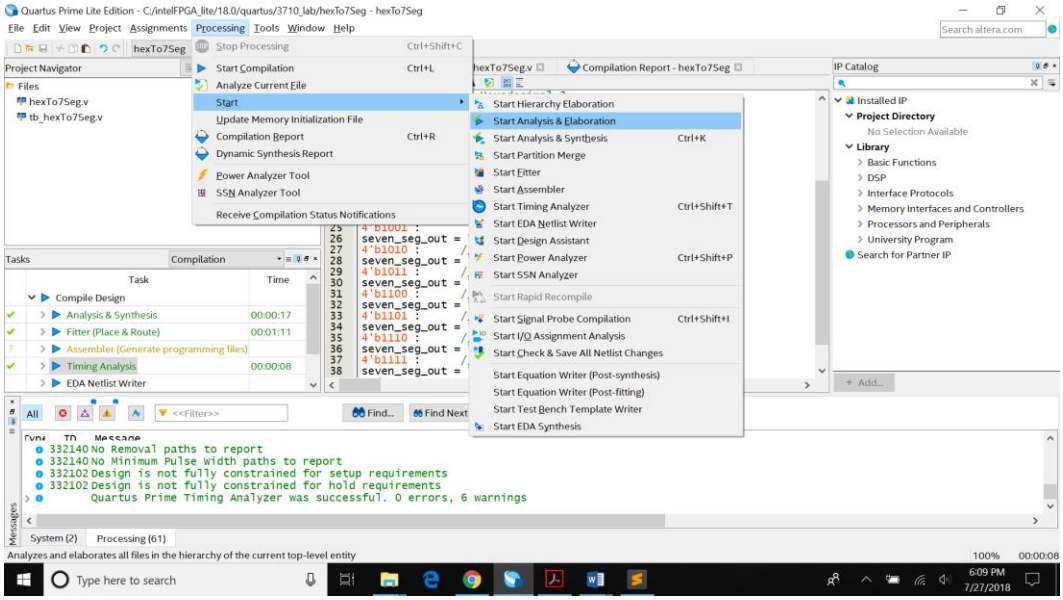
# Section 5 – Pin assignment and Constraint generation



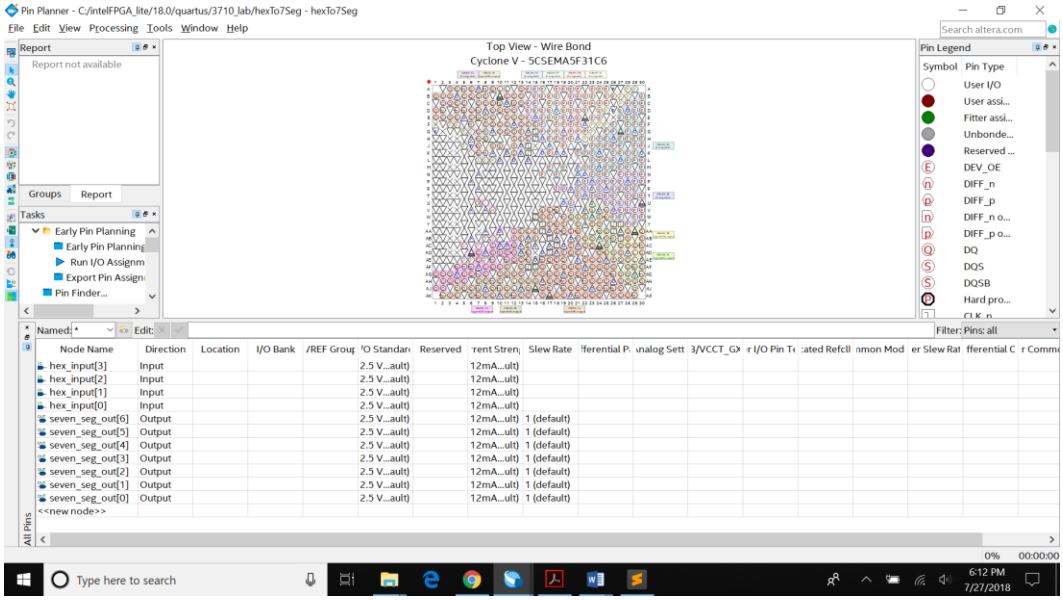



In this section, you will make pin assignments for all the signals from the design which interact with outer world. Before making pin assignments, perform the following steps:

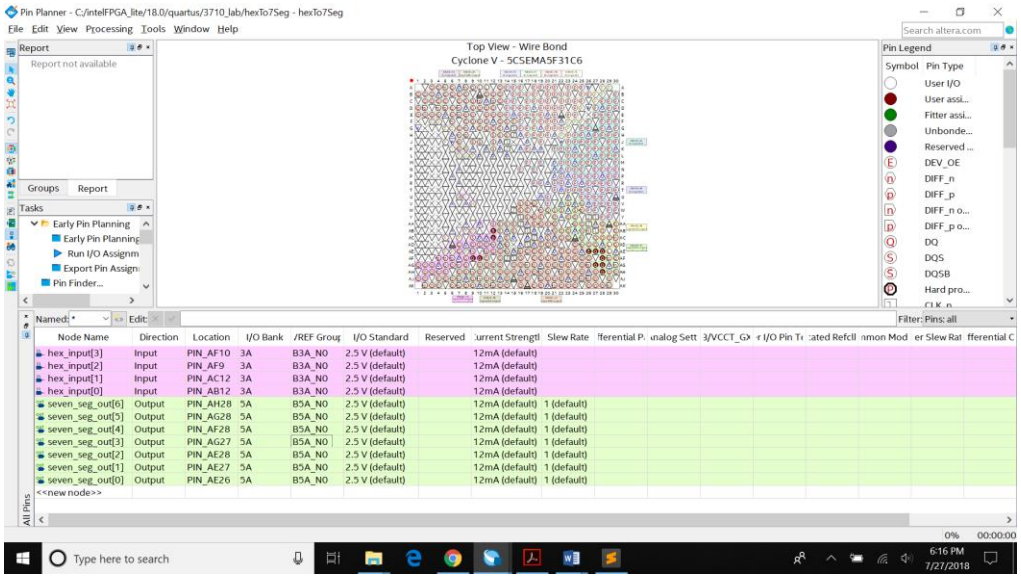
1. Choose Processing > Start > Start Analysis & Elaboration in preparation for assigning pin locations. Click OK in the message window that appears after analysis and elaboration completes.




2. To make pin assignments that correlate to the hex_input[3:0] input pins and seven_seg_out[6:0] output pins, perform the following steps: Choose Assignments > Pin Planner, which opens the Pin Planner, a spreadsheet-like table of specific pin assignments. The Pin Planner shows the designs pins.



3. In the Location column next to each of the node names, add the coordinates(pin numbers) as shown from the tables below for the actual values to use with your DE1-SoC board(check manual for any particular pins you are looking for based on your design usage). Double-click in the Location column for any of the pins to open a drop-down list and type the location shown in the table alternatively, you can select the pin from a drop-down list. For example, if you type AF10 and press the Enter key, the Quartus II software fills in the full PIN_AF10 location name for you. The software also keeps track of corresponding FPGA data such as the I/O bank and VREF Group. Each bank has a distinct color, which corresponds to the top-view wire bond drawing in the upper right window.




| *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 |

| *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 |

4. The design.qsf file should reflect these pin assignments once you have set them on pin planner.

```
set_location_assignment PIN_AF9 -to hex_input[2]

set_location_assignment PIN_AC12 -to hex_input[1]

set_location_assignment PIN_AB12 -to hex_input[0]

set_location_assignment PIN_AF10 -to hex_input[3]

set_location_assignment PIN_AE26 -to seven_seg_out[0]

set_location_assignment PIN_AE27 -to seven_seg_out[1]

set_location_assignment PIN_AE28 -to seven_seg_out[2]

set_location_assignment PIN_AG27 -to seven_seg_out[3]

set_location_assignment PIN_AF28 -to seven_seg_out[4]

set_location_assignment PIN_AG28 -to seven_seg_out[5]

set_location_assignment PIN_AH28 -to seven_seg_out[6]
```

5. Timing settings are critically important for a successful design. For this tutorial you will create a basic Synopsys Design Constraints File (.sdc) that the Quartus II TimeQuest Timing Analyzer uses during design compilation. For more complex designs and requirements, you will need to create your .sdc files by considering the timing requirements more carefully.

   To create an SDC, perform the following steps:

   a. Open the TimeQuest Timing Analyzer by choosing Tools > TimeQuest Timing Analyzer.
   b. Choose File > New SDC file. The SDC editor opens in the quartus software with a file extension as .sdc.
   c. Type in your timing constraints(clock/pll) into the file and save it as top-level file(hexTo7Seg.sdc in this case). Since the current design doesn't have a clock, we will ignore the constraints. Refer the DE1-Soc manual for creating a PLL/clock as it is explained in detail.
   d. Naming the SDC with the same name as the top-level file except for the .sdc extension causes the Quartus II software to using this timing analysis file automatically by default. If you used another name, you would need to add the SDC to the assignments file list.

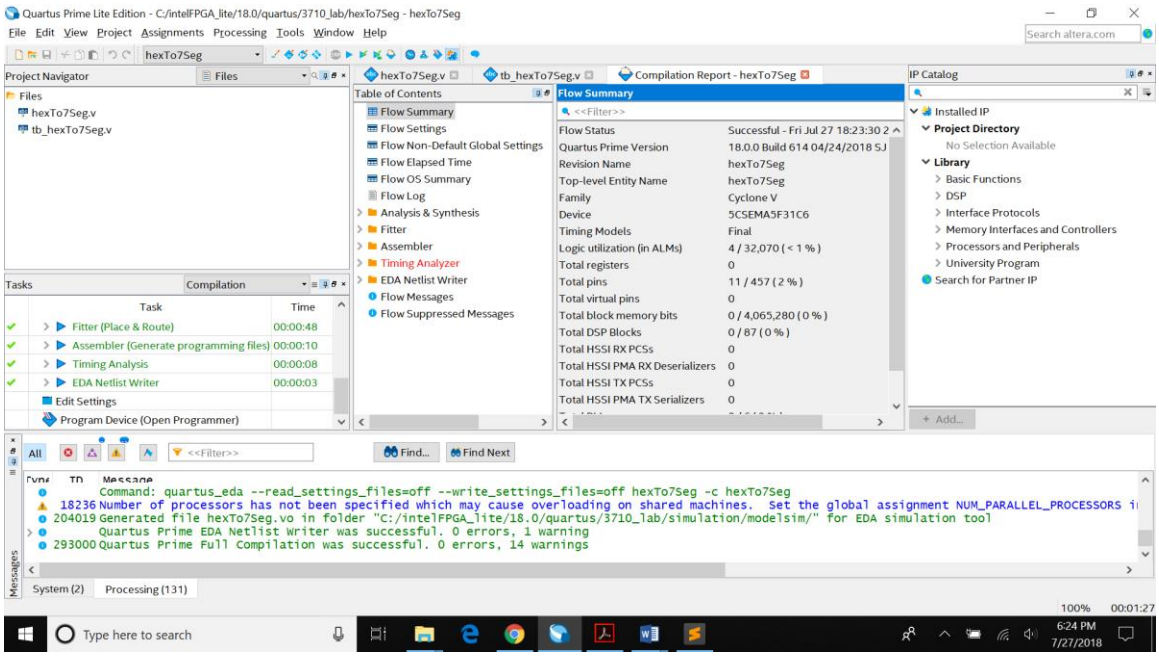
# Section 6 – Synthesize, Implement, Generate, and Program for DE1-Soc board

After creating your design you must compile it to convert the design into a bitstream that can be downloaded into the FPGA. The most important output of compilation is an SRAM Object File (.sof), which you use to program the device. The software also generates other report files that provide information about your code as it compiles. If you want to store .SOF in memory device (such as flash or EEPROMs), you must first convert the SOF to a file type specifically for the targeted memory device.

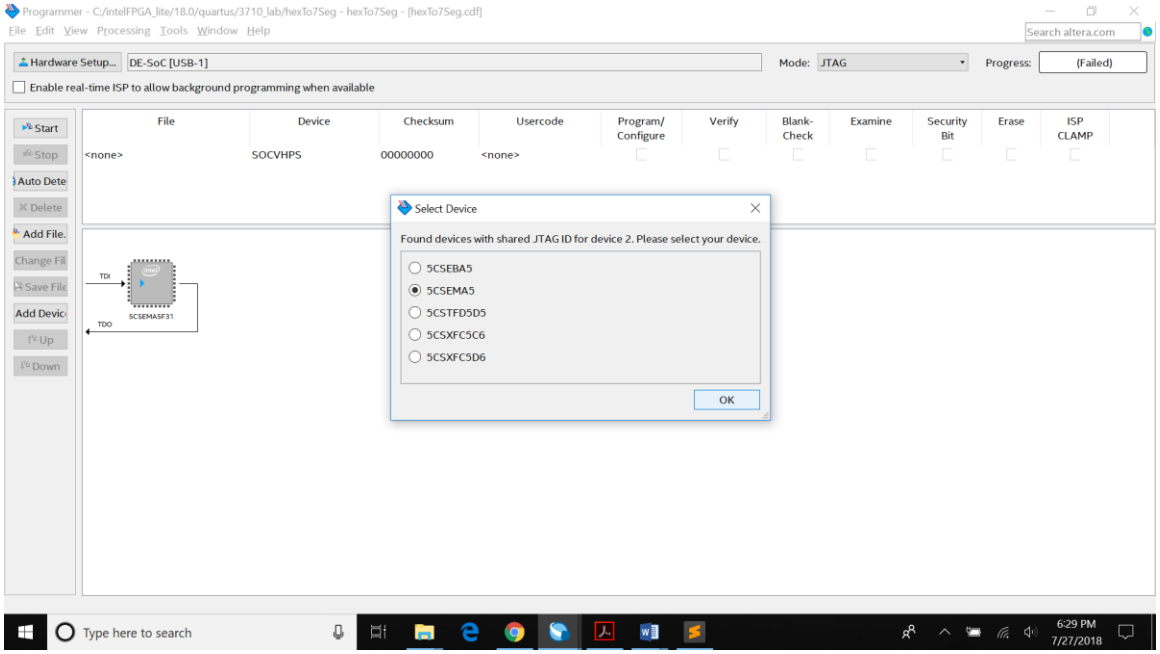
Now that you have created a complete Quartus II project and entered all assignments, you can compile the design.

1. In the Processing menu, choose Start Compilation or click the Play button on the toolbar. When compilation is complete, the Quartus II software displays a message. Click OK to close the message box.

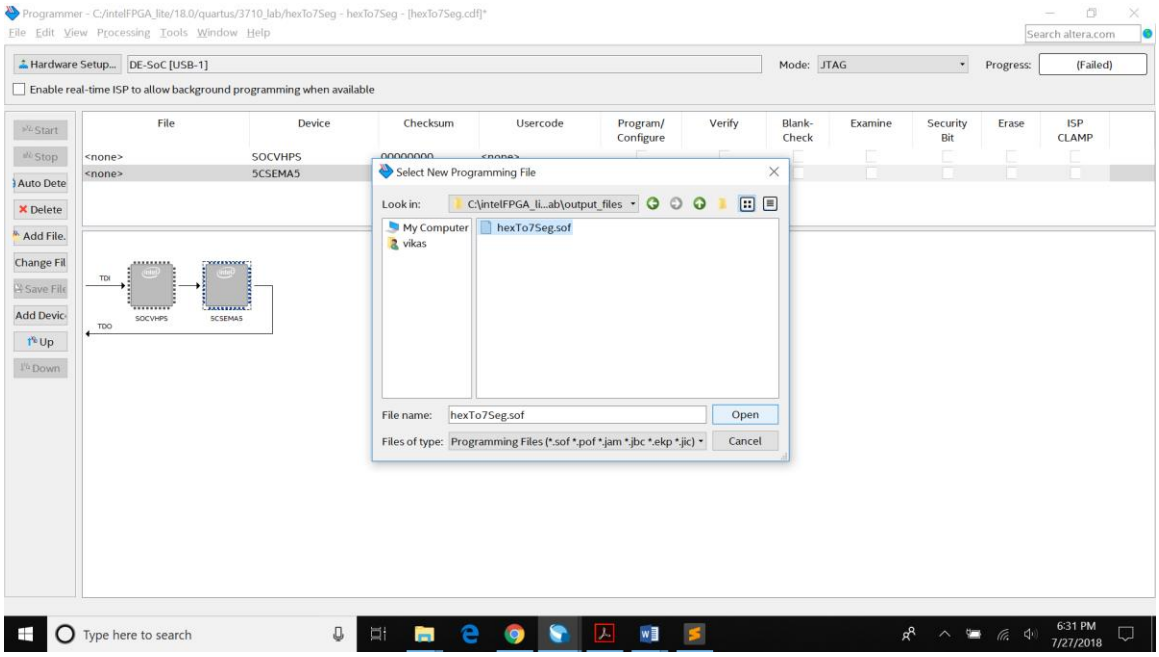
The Quartus II Messages window displays many messages during compilation. It should not display any critical warnings; it may display a few warnings that indicate that the device timing information is preliminary or that some parameters on the I/O pins used for the LEDs were not set. The software provides the compilation results in the Compilation Report tab as shown.



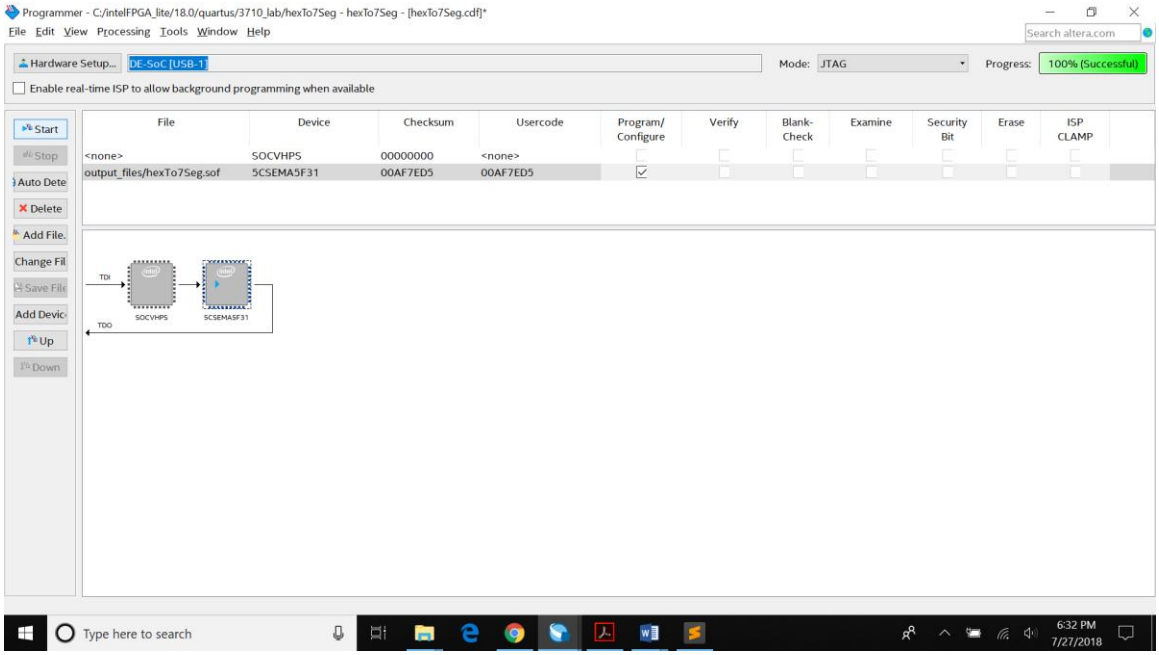
2. After compiling and verifying your design you are ready to program the FPGA on the development board. You download the SOF you just created into the FPGA using the USB-BlasterII circuitry on the board. Set up your hardware for programming using the following steps:
    a. Connect the power supply cable to your board and to a power outlet.
    b. For the DE1-SoC board, connect the USB-BlasterII (included in your development kit) to pin J13 and the USB cable to the USB-BlasterII. Connect the other end of the USB cable to the host computer. (Refer to the getting started user guide for detailed instructions on how to connect the cables in case you are getting confused)
    c. Turn the DE1-SoC board on using the on/off switch.
3. Program the FPGA using the following steps.
    a. Choose Tools > Programmer. Once the Programmer window opens, click Hardware Setup. If it is not already turned on, turn on the DE1-SoC [USB-1] option under currently selected hardware.
    b. Make sure the Mode is set to JTAG.
    c. Click Auto Detect to detect all the devices on the JTAG chain. Select the device as 5CSEMA5 which reflects the device ID for DE1-SoC cyclone 5 board. Click OK after the selection.



4. Both HPS and FPGA will be listed on the programmer window. Select FPGA device and click Change File to program the FPGA with the relevant .sof file(hexTo7Seg.sof in this case).




5. click Start to program the .sof file into FPGA and make sure the progress bar reflects success and not failure.



6. When you verify the design in hardware, observe the runtime behavior of the FPGA hardware and ensure that it is functioning appropriately.

"Congratulations, you have created, compiled, and programmed your first FPGA design! The compiled SRAM Object File (.sof) is loaded onto the FPGA on the development board and the design should be running."

**References and miscellaneous links –**

DE1-SoC for 3710 overview: http://www.terasic.com.tw/SocDE1

Altera Quartus Lite software: http://dl.altera.com/18.0/?edition=lite

User manual/datasheets/demonstrations/sample codes: Click the link below http://download.terasic.com/downloads/cd-rom/de1-soc/ and download the latest system build (DE1-SoC_v.5.1.1_HWrevF_SystemCD.zip) with Quartus 16.0(>) support.

For enabling inbuilt Linux Subsytem in windows please follow the instructions mentioned in the article, it works like a charm: https://www.windowscentral.com/how-install-bash-shell-command-line-windows-10