Document prepared by Philip Jacob &
Steve Nicholas –Fall 2009

Rensselaer Polytechnic Institute
Computer Hardware Design – ECSE 4770

Quartus Tutorial

1. Introduction:

This tutorial session has the goal of familiarizing you with the tools necessary to complete the labs based on FPGA boards. First we will cover the basics of the software. Then you will enter a simple design the part will be simulated. Finally, there will be a brief description of downloading the design to the FPGA.

2. Quartus program basics:

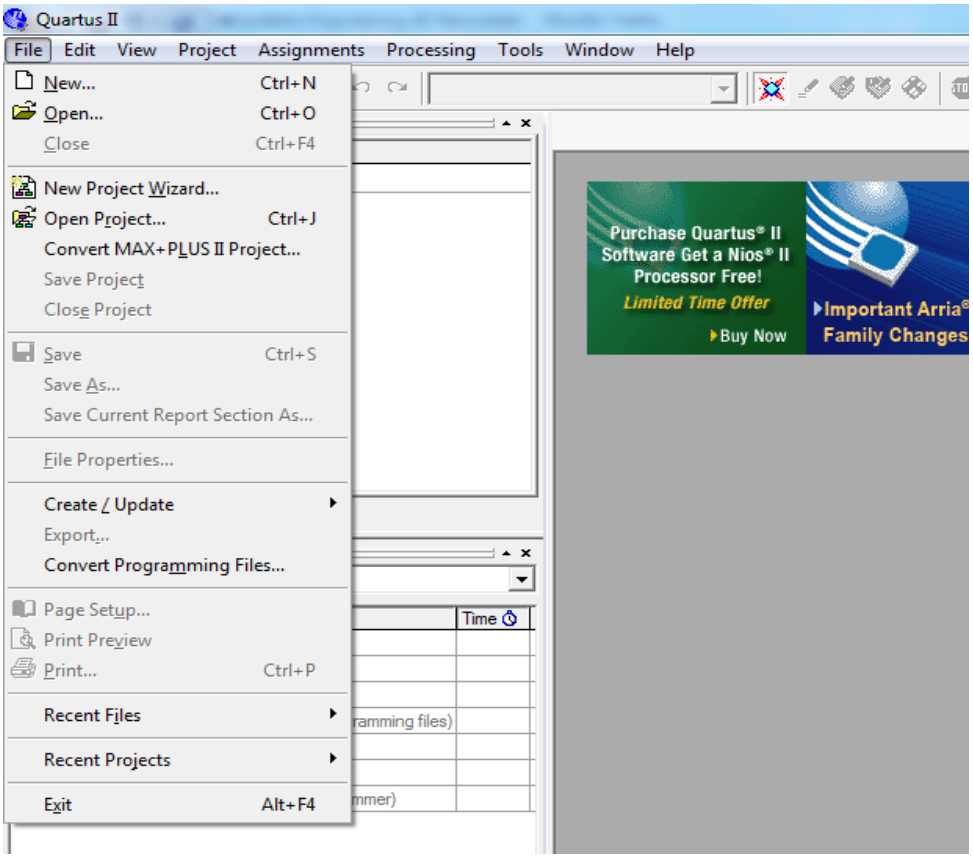
2.1 Download: The Quartus tool is provided by Altera for their FPGA boards. It is available for download from the following website.
https://www.altera.com/support/software/download/altera_design/quartus_we/dnl-quartus_we.jsp
Current version is V9.0 service pack 2 (1.31GB). Download this and install on your laptop. This version doesn't require a license. The lab workstations have a fall back version of 7.2 which is known to work in the previous academic year. In case you have problems you can get this version from the archives section of the same site.
https://www.altera.com/support/software/download/altera_design/quartus_archive/dnl-quartus_archive.jsp

2.2 Project creation:

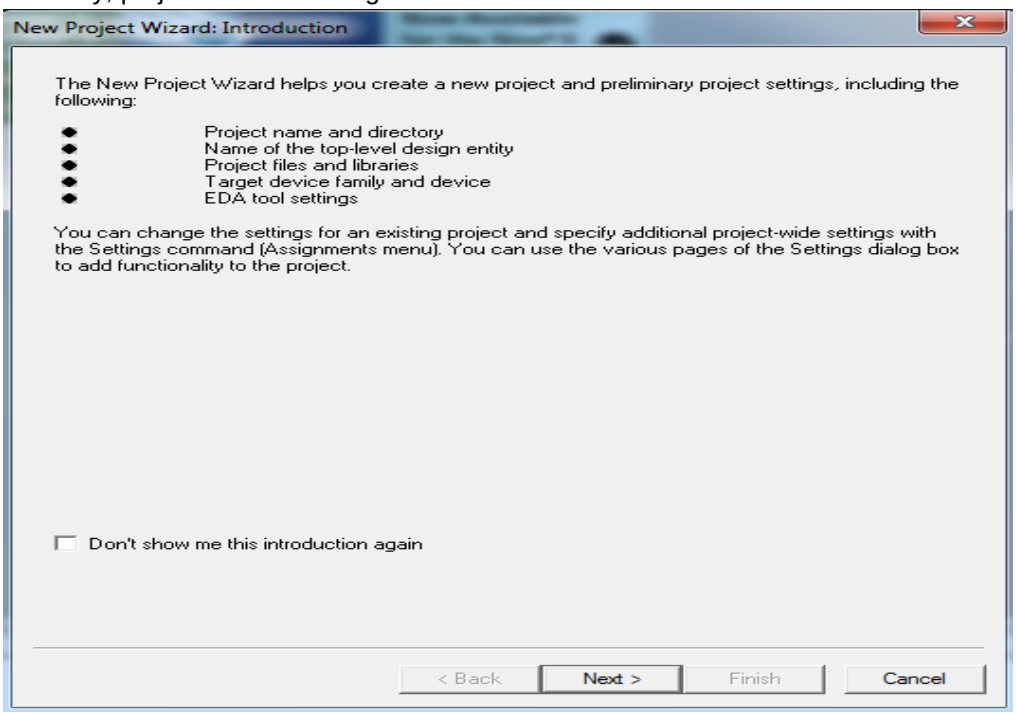
Start the Quartus software from your program files.

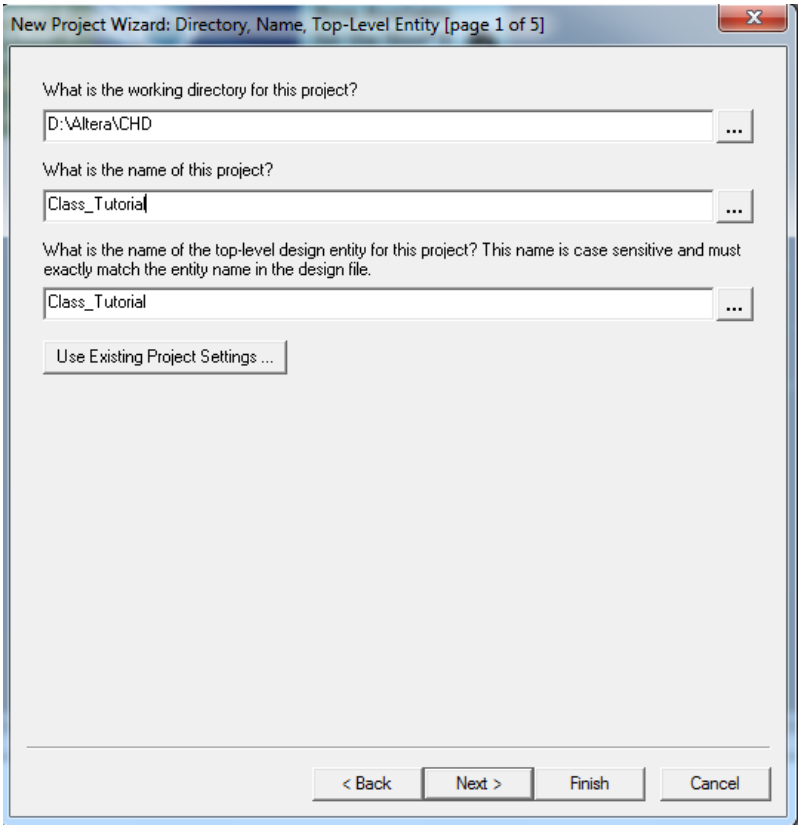
All programs→ Altera → QuartusII 9.0 web edition→QuartusII 9.0sp2 web edition (blue logo )
The quartus program starts up with the following screen with the options to create a new project or open an existing project.

Alternatively, you could open a new project or existing project by going to the file menu.



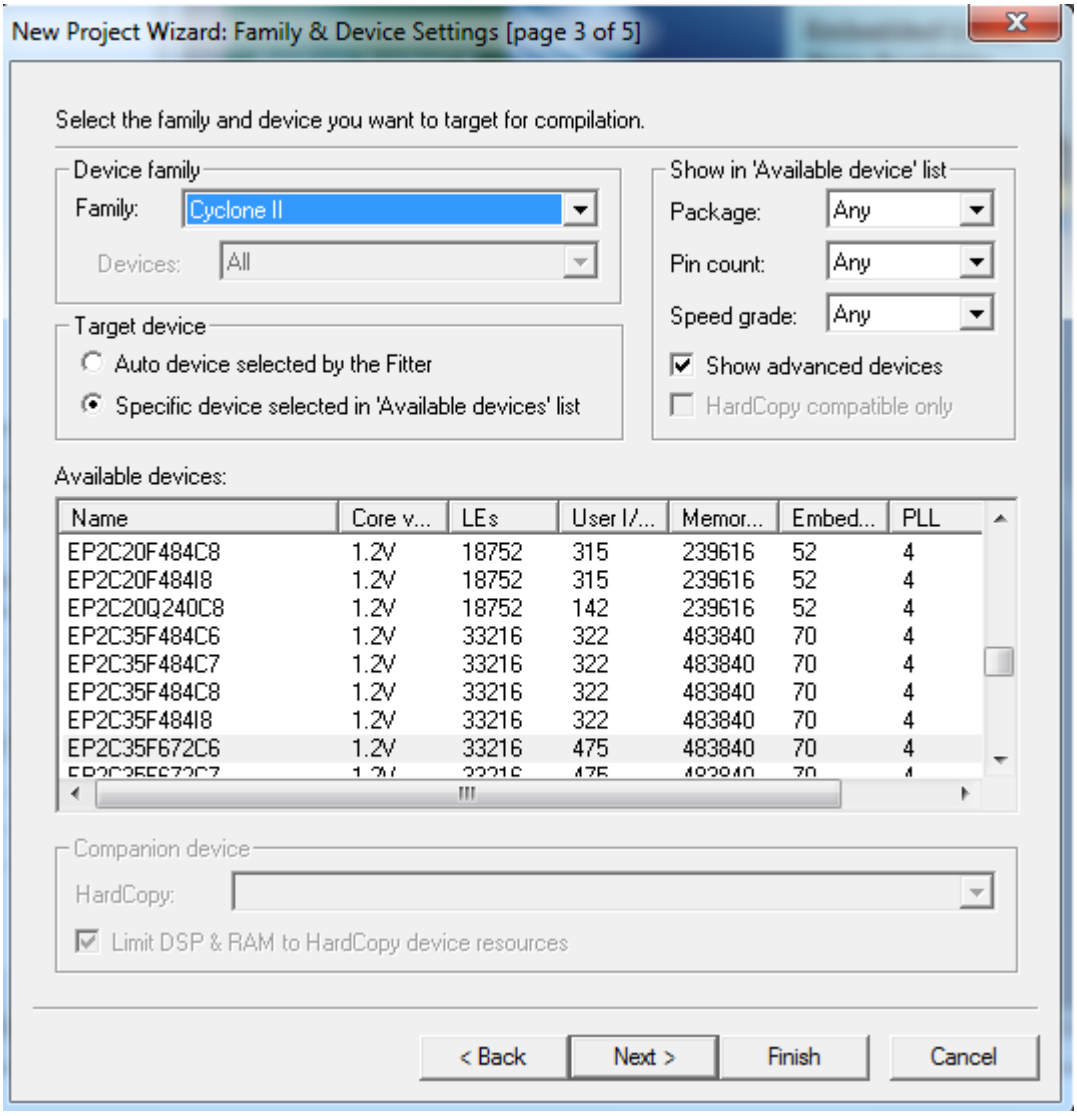
Create a new project using one of the above options. You get the following screen to set the project directory, project name and target FPGA device. Click next.





In the above screen, the working directory was chosen by user preference. The default location would have been in the installation directory. Each version upgrades of Quartus creates a separate installation

folder. Hence, it is advisable to have a separate folder for quartus projects. This choice is left to the user. Enter the project name and click Next to get the next screen. Skip the next screen on adding files and proceed to the screen on setting family and device settings.



Altera has a long list of FPGA chips including Flex10k, Max7000, Stratix, Cyclone (I, II, III).

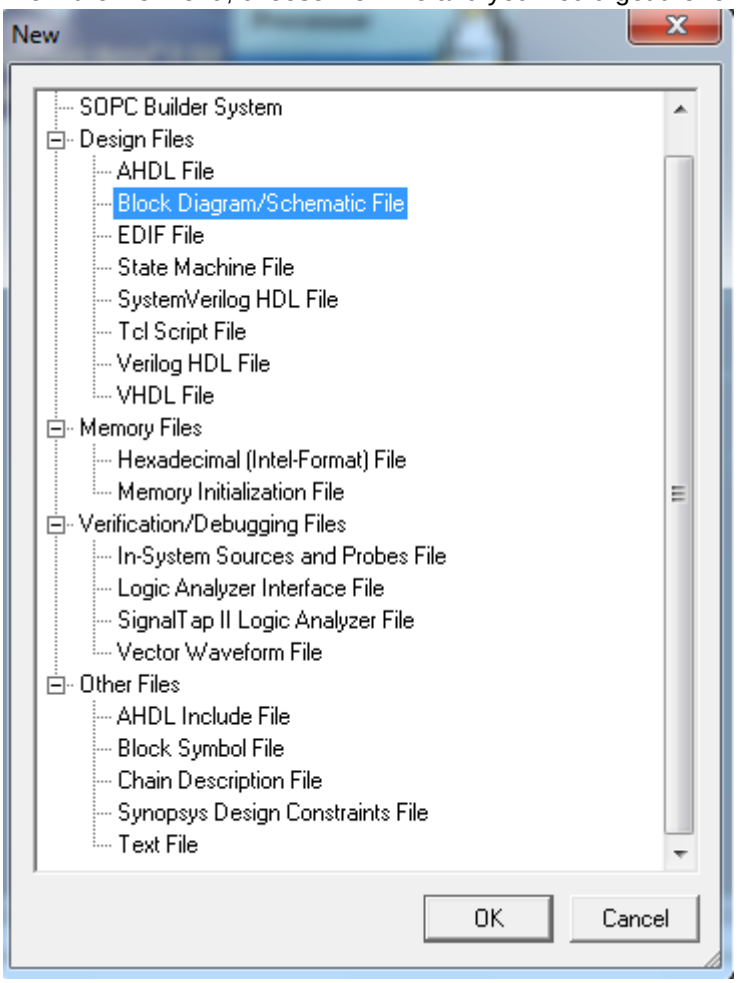
Our lab supports two boards for the course namely UP2 and DE2. UP2 board which has been used in the previous academic years has the Flex10k and Max7000 series chips. The DE2 board is the newer version and has the Cyclone II chip on it. We also have boards for Cyclone III. Depending on the family of FPGA, the number of logic components that can be programmed into the chip varies. For the tutorial, **choose Cyclone II** in the Family drop box. In the Available devices box, choose **EP2C35F672C6**. You can identify the right chip and number by looking at the Altera board's FPGA chip. If a different board is chosen, choose accordingly the chip used on that board.

Skip the next screen on EDA tools. This is provided if you wish to use external EDA tools from synopsis or cadence. Altera/ Quartus provides a built in tool for compiling Verilog, VHDL, AHDL and a timing analyser. This is sufficient for our course requirements.

Click finish on the last screen.

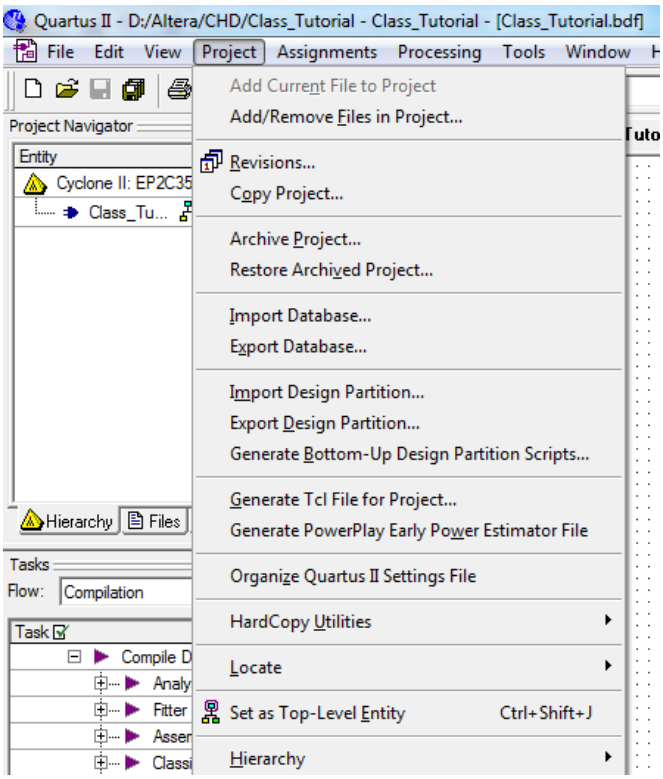
2.3 Design Entry:

From the file menu, choose New file and you would get the following screen.



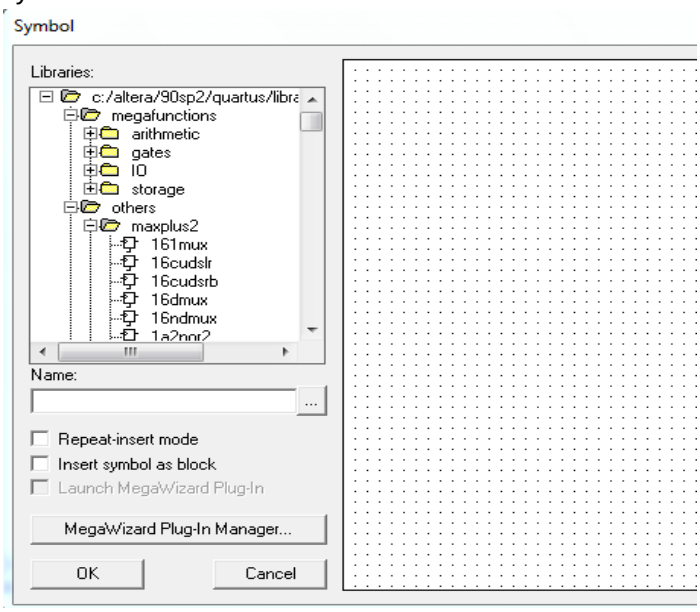
There are several file types listed in the above screen. Block Diagram/ Schematic file entry is chosen for graphical entry of the schematic and is the preferred form of logic diagram entry. Verilog and VHDL file is also supported to input the circuit description in those languages. We also have a file for Memory initialization file. This file is used to store a certain pattern of hexa-decimal digits. This is useful in the advanced computer hardware design lab. The other file of interest is the Vector Waveform File. This file is used to setup the testing waveforms for input and to observe the output after the simulation.

Choose Block Diagram/Schematic entry to proceed with the project design. The default name would be Block#.bdf. Save this file to the file name of your choice. The first file of the project is generally saved to the default project name. Set this particular file as your top level entity in your design as shown in the following screen – **Project → Set As Top level Entity**.



If you now go to the **Project → Add/Remove Files**, you will encounter a screen that we had encountered during project set up. And you will notice the current file listed in the box.

In the schematic window, you will notice a graphical menu on the left. Hovering your mouse over these options will list Selection tool, Text tool, Symbol tool, Block tool, Orthogonal node tool, Orthogonal bus tool and some options to rotate/ flip your symbols. To start, **double click in the schematic window** to get the following screen listing the various libraries/ symbols listed. Expand the selection to see various symbol lists.



Altera provides prebuilt library entities for mega functions (custom built to your preferences), maxplus libraries that include the 74 series chips commonly used in the lab protoboards, as well as some primitives. The name box allows you to search or a particular symbol (such as input, and2, output, nor2 etc.). The Mega wizard plug-in manager gives you more options to build custom designs. This will be shown later.

Select the logic symbol you wish to enter and press OK to insert it into the schematic window. Place them anywhere in the schematic. They can be moved around by clicking on them and dragging to the new position. For the tutorial, we will have a simple logic with NAND, AND and an OR gate.

Choose **primitives → Logic → nand2**; similarly for nor2 and and2.

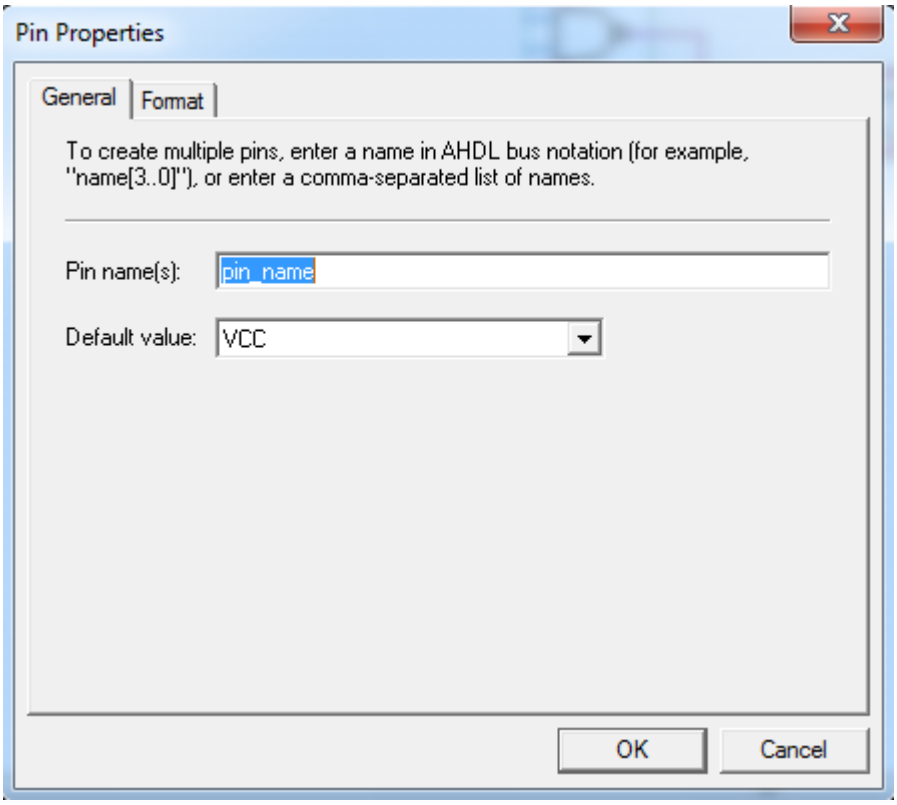
Insert the symbols for Input and output by following the same procedure. These symbols are also listed in the primitives section (or enter input or output in the name box).

You may also enter the actual chip (7400 for nand and 7402 for nor) from the maxplus library which is available in the "others" →maxplus2 → 7400; similarly for 7402. Do check datasheets available on websites such as alldatasheet.com for these chip numbers and pin numbers for more details.

If you wish to delete any incorrect entry, select the symbol and press delete on your keyboard.

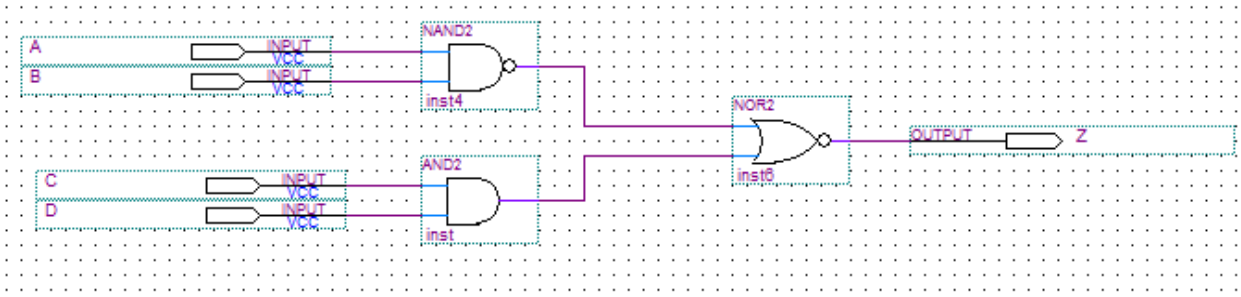
Connect the blocks using the orthogonal node tool (thin wire) available on the left menu. This allows drawing vertical/horizontal lines. You will need to click and hold as you draw. Ensure that there are no gaps to ensure proper connection between the two terminals. Moving one of the symbols should automatically stretch the connected wire if there are no breaks in the wires.

Rename the input and output pins by double clicking them. You will get the following pop up screen.



Enter the corresponding name for the pin name.

Create the following schematic.



The above logic can be reused as a new design entity in a different schematic. To do this, you would need to create a symbol file for the given schematic.

Choose File → Create / Update → Create Symbol File for current file. You can further edit the symbol file by opening it and making changes to the symbol file.

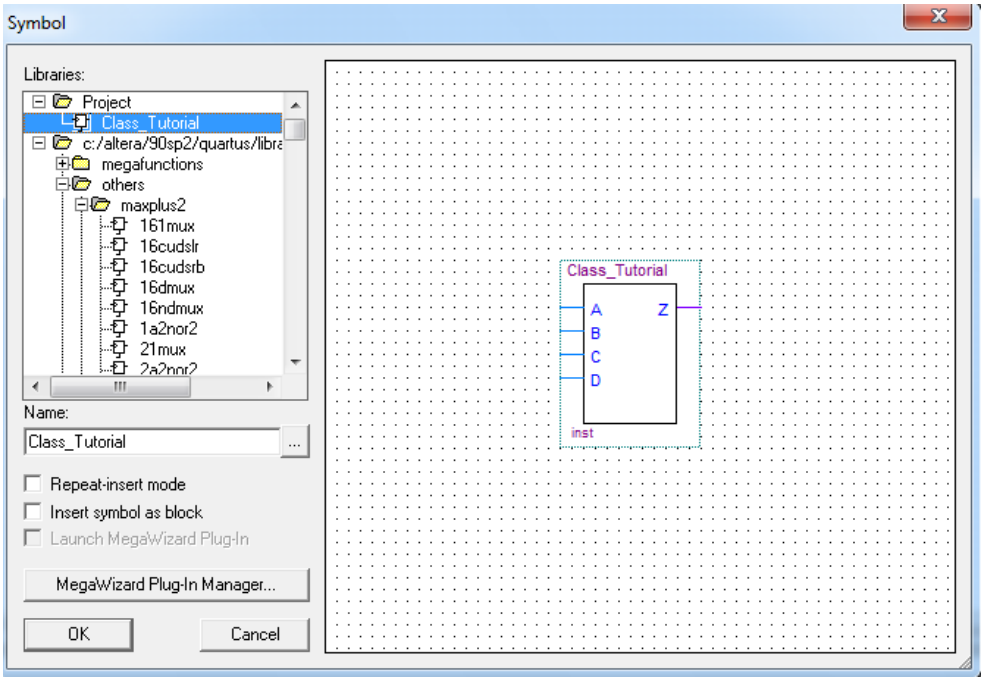
A new file is created with the default name as the same existing file but the extension is .bsf (block symbol file). Alternatively, you can manually create a new symbol file from the **File→ new file → Other Files→ Block symbol file** and Enter the input and output pins. This is not recommended for the class as you may miss pins that could cause compilation errors later.

To use this schematic in a different schematic, open a new block schematic file.

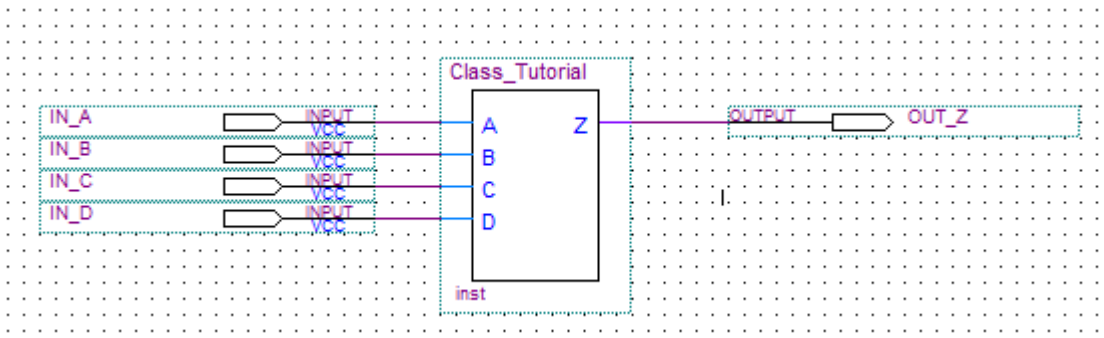
File → New → Block Schematic file

Since we are instantiating the other schematic symbol in this new file, we will need to set the new file as the top level entity after it has been saved to a file name. With the new file window selected, choose **Project→Set as top-level entity**.

In the new file, double click to insert a symbol. In the menu, under Project, you will see the previously created symbol file listed. Select this and insert it in the schematic.



You may set a new set of input /output pins in the new schematic.



We have completed schematic entry and symbol creation.

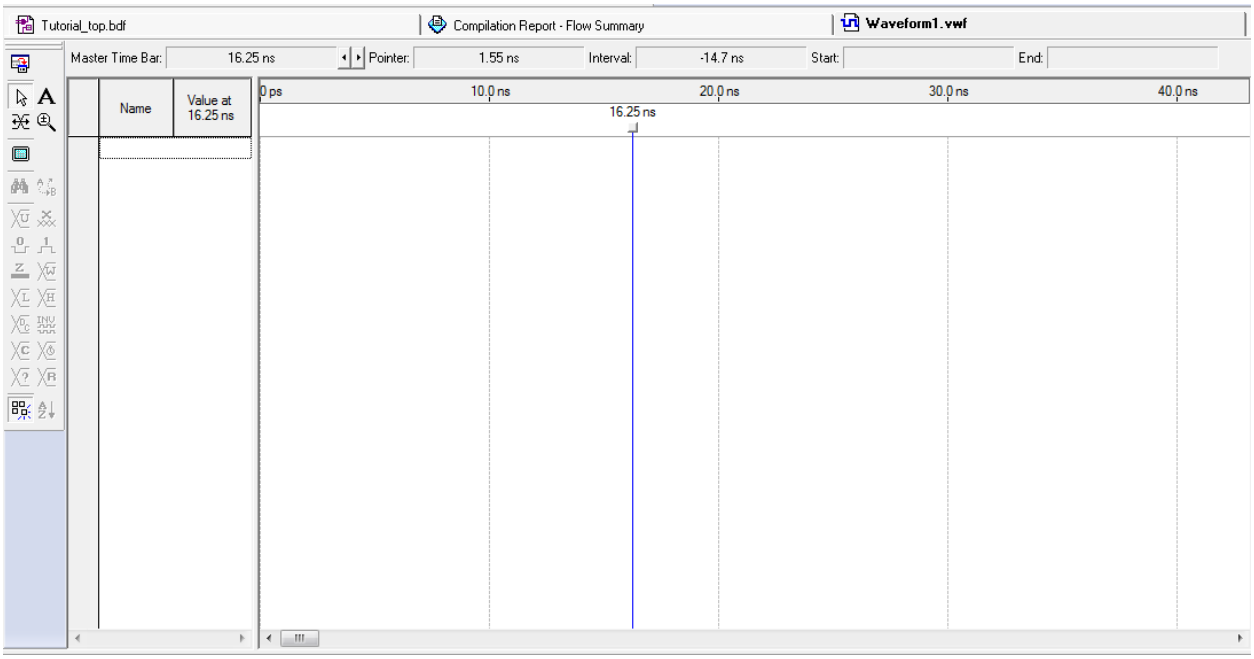
3. Compilation and Simulation:

To compile the project, click on the **Processing→Start Compilation**. This would carry out a series of steps from Analysis and Synthesis, Fitter (Place & Route), Assembler and Timing Analysis. Alternatively, you could do this one step at a time by going to Processing → start → and choose each step in sequence one at a time.

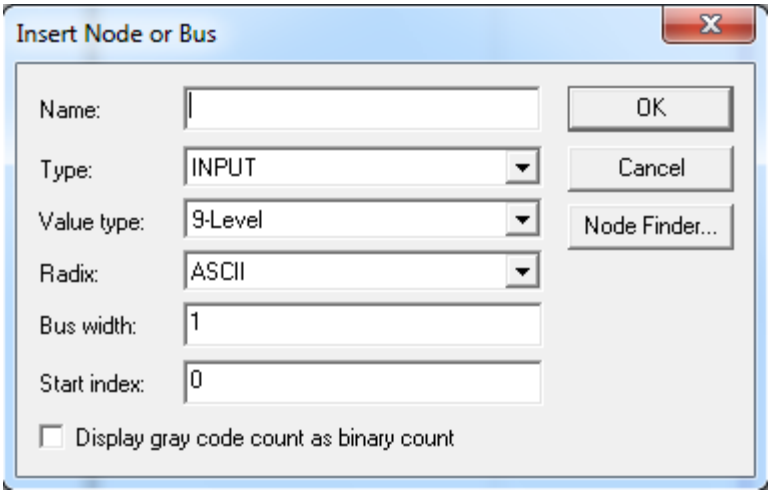
The window below shows the compilation messages. If there are any messages in red, they need to be fixed.

To carry out simulations, we need to setup the simulation input and output vectors. **Create a new file (Vector Waveform File).**

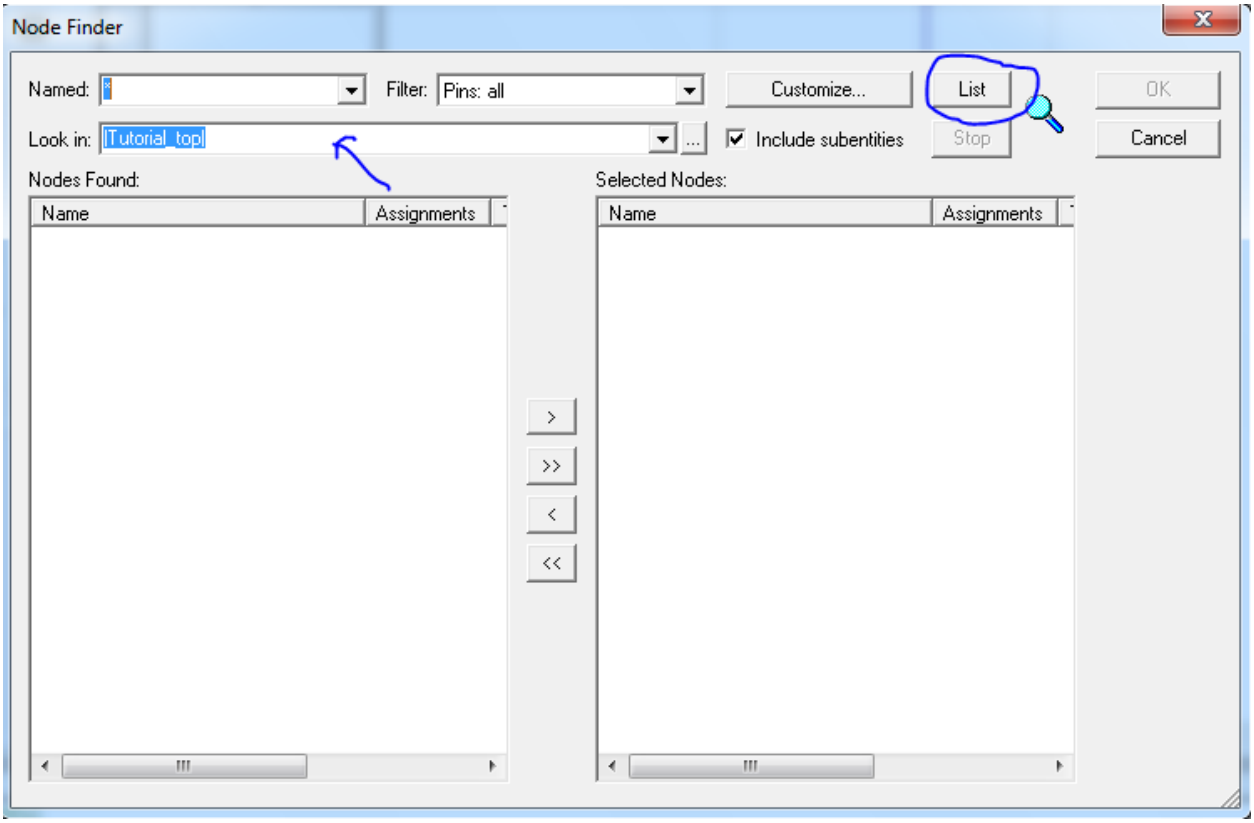
You should be getting a blank waveform file as shown in the following screen capture.



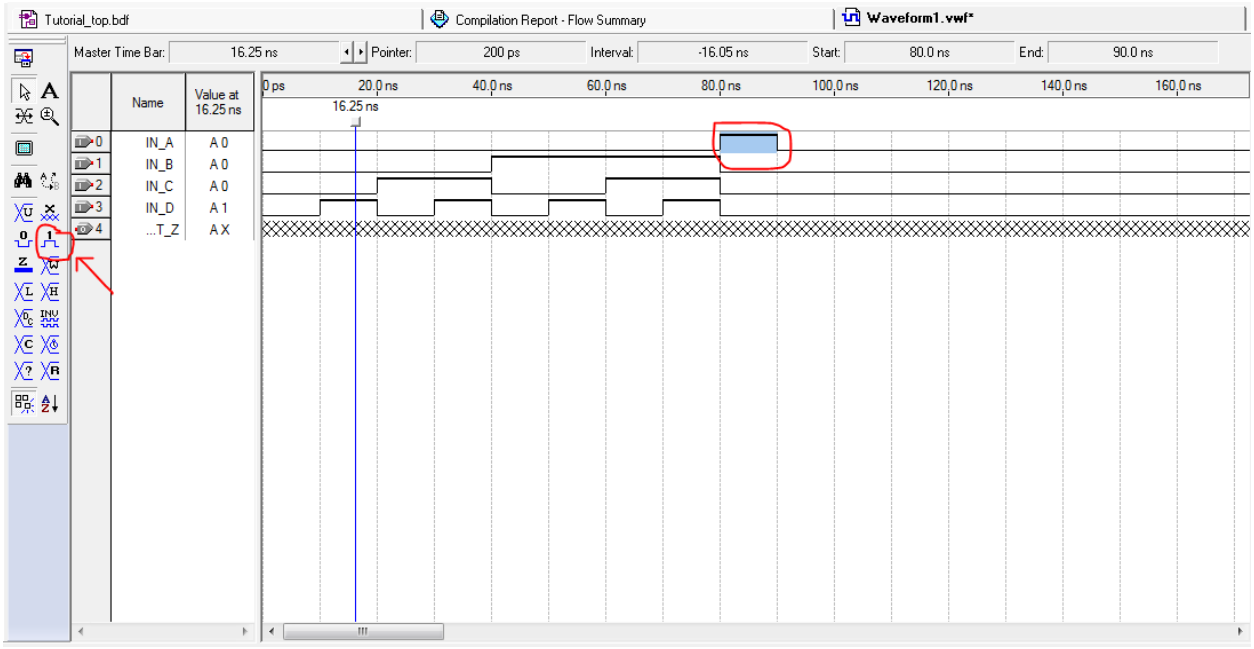
The left partition (Name) would be where the nodes are added. The right partition is where the input and output waveforms are observed. To select the input and output pins that are being observed for simulations, **right click in the Name pane → Insert → Node or bus.** This should give you the following screen



Click on Node finder to get to the next screen shot



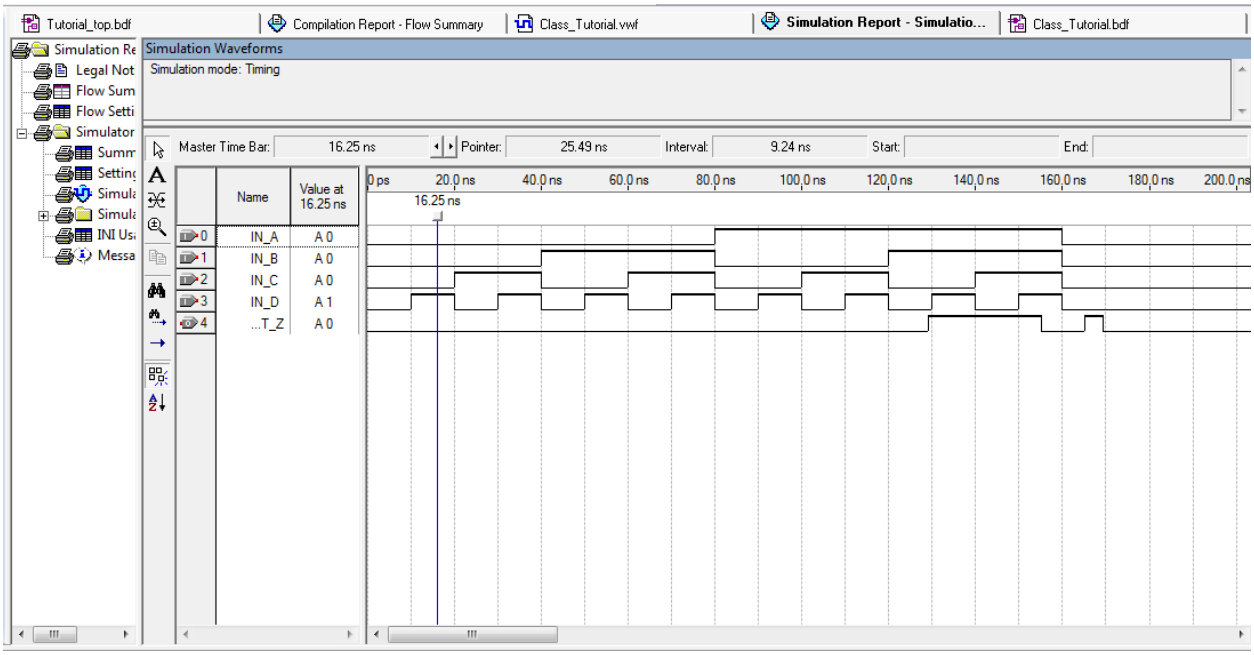
The “Look in” box identifies the schematic file from which the node is listed. Click List. The left pane identifies the nodes found. Select the required nodes and send them to the right pane by clicking the right arrow. Click ok to return to the insert node/bus window. Click ok to return to the vector waveform screen.



The waveform pattern for the input initially set to default zero. You can change the pattern to the desired value of high or low by drawing a rectangle for each segment you wish to change with the mouse and choosing the high or low option.

Save the vector waveform file.

Click simulate icon shown in the figure below to get the output results of the simulation

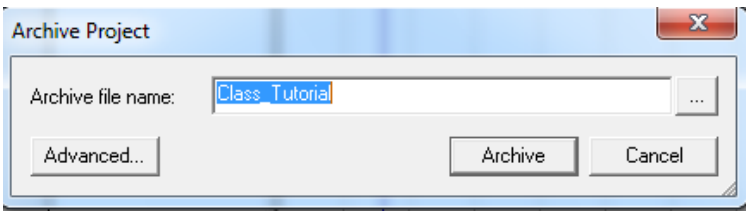


Take a moment to validate the results for each set of inputs. The output should correspond to the logic implemented in the schematic.

Save your project (**File → Save Project**)

Quartus provides a means to archive the files into a small quartus archival which is convenient for transfer to a flashdrive or uploading to your email. You can restore the files from this archival on a different machine or simply keep these archives as a backup for your designs. To do this, first check whether all the required files are present in the project (**Project → Add/Remove files** in Project).

To archive: click **Project→Archive** Project resulting in the following pop up window.



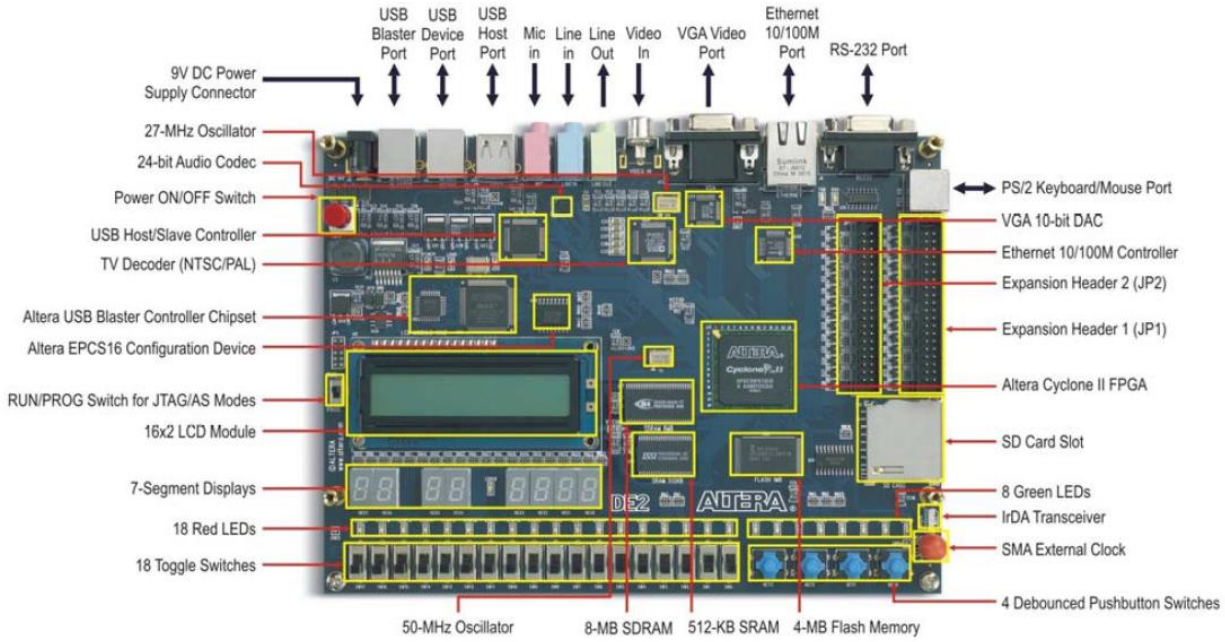
All the relevant files for the project will be automatically included and archived. You can locate the archived file as your project name.qar (Quartus II Archive File) in the directory where the project was initially created. To restore, open Quartus and click Project → Restore Archived file. Browse to locate the .qar file and follow the onscreen instructions regarding the directory where the project needs to be restored.

4. Transferring design to FPGA board

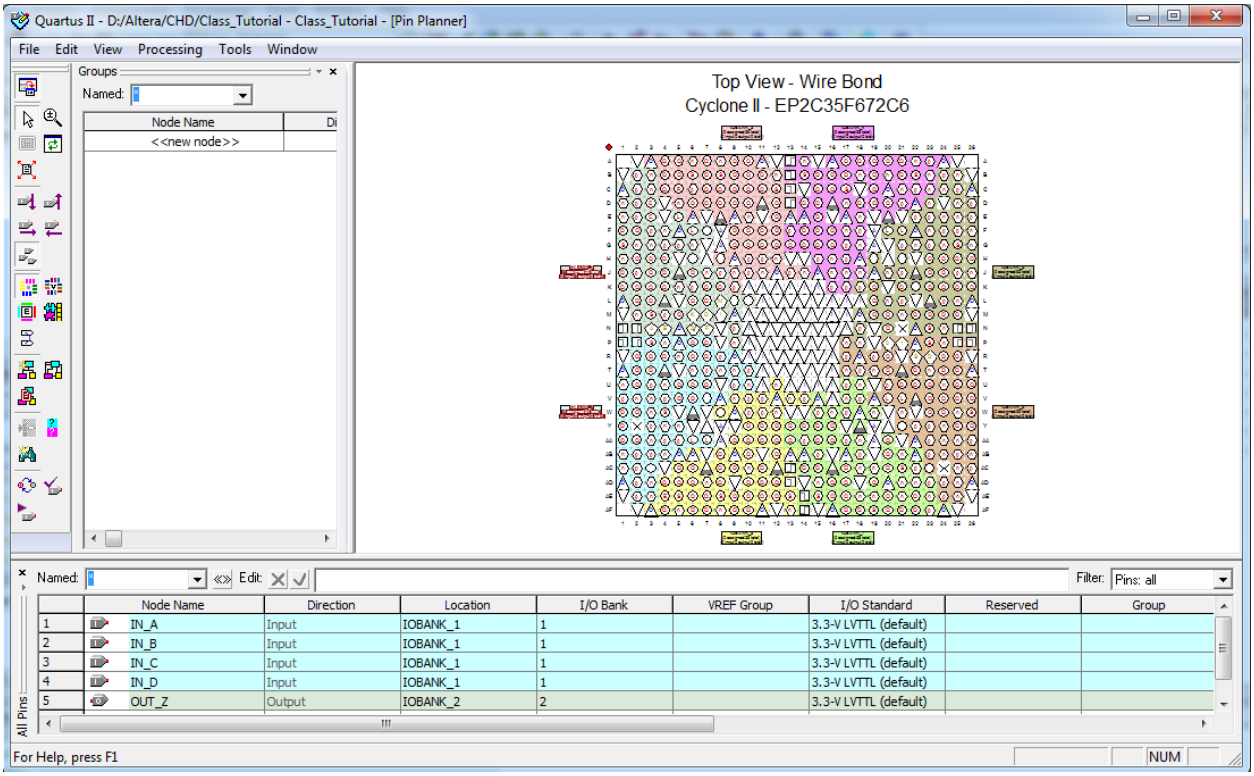
4.1 Initial setup for DE2 board:

The board details can be found on the DE2 user manual found in the following link.

ftp://ftp.altera.com/up/pub/Webdocs/DE2_UserManual.pdf



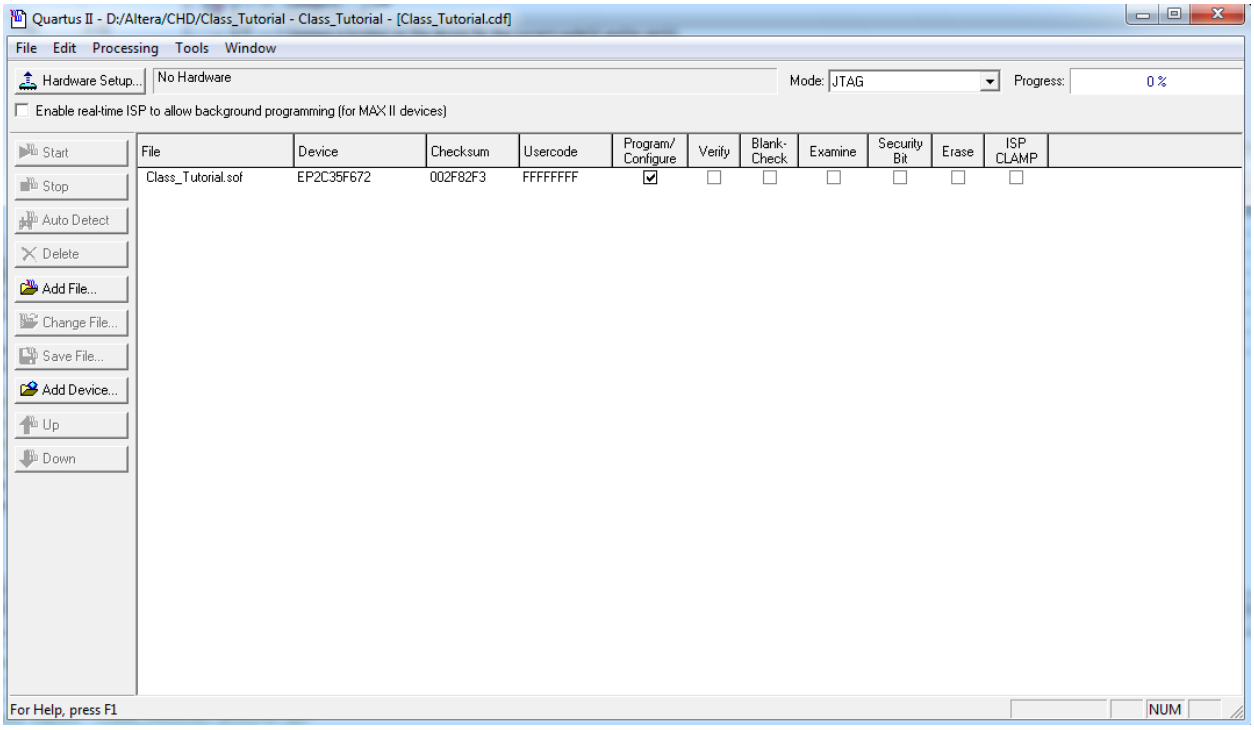
Before transferring the compiled binary to the FPGA board, you need to assign the pin number/ names to the input /output pins in the schematic. Go to Assignments – pins to get the following screen shot.



Do the pin assignments for the given input / output pins to the corresponding IObanks on the FPGA chip. You have the freedom to choose any IOBank and pin number. Later, we will show how to assign the pins to a particular switch or led.

Now that the assignments have been done, compile the project again. Click **Tools → Programmer** to open a new window which would allow us to transfer the binary to the FPGA.

Ensure that your computer is connected to the FPGA board. In the case of DE2 boards (Cyclone II), you can connect directly through the USB port on your computer. For the older boards with MAX/FLEX chips, you would need to connect through the printer port. Installation of a byte blaster program is also required for the printer port method. These have been set up on the lab computers for the older boards.



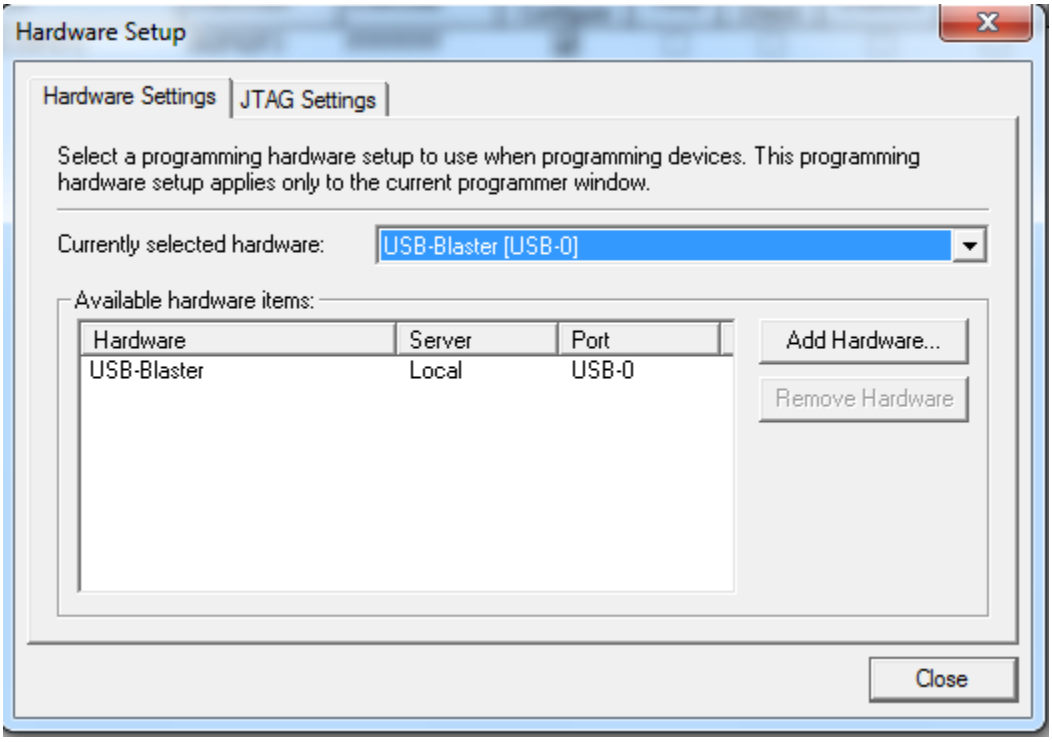
When the hardware is connected, the start button is enabled. Click it to transfer the sof file to the FPGA. Your logic circuit is now programmed into the FPGA. Connecting external wires to the corresponding pins on the IOBanks to the breadboards, one can simulate the logic with external signal waveforms.

4.2 Altera USB blaster driver:

This should be already come with Quartus installation. Check the drivers directory in the quartus installation folder. C:\altera\90sp2\quartus\drivers

1. Plug the USB-Blaster download cable into the PC. The **Found New Hardware** dialog box appears.
2. Select **Locate and install driver software (recommended)**.
3. Select **Don't search online**.
4. When you are prompted to **Insert the disc that came with your USB-Blaster**, select **I don't have the disc. Show me other options**.
5. Select **Browse my computer for driver software (advanced)** when you see the **Windows couldn't find driver software for your device**.
6. Click **Browse...** and browse to the <*Path to Quartus II installation*>**\drivers\usb-blaster** directory. Click **OK**. {C:\altera\90sp2\quartus\drivers\x32}
7. Select the **Include subfolders** option and click **Next**.
8. If you are prompted **Windows can't verify the publisher of this driver software**, select **Install this driver software anyway** in the **Window Security** dialog box.
9. The installation begins.
10. When **the software for this device has been successfully installed** appears, click **Close**.

In the previous programmer window, you may need to set up the Altera hardware setup (left top corner). Change the currently selected hardware to USB-Blaster and close the window.



4.3 Quartus II Version 4.2 and later

1. Start the Quartus® II software.
2. Choose Programmer from the Tools menu. The Programmer window will open.
3. Click the Hardware Setup... button to open the Hardware Setup window.
    a. The selected programming hardware is identified as Currently Selected Hardware.
    b. Programming hardware that is already set up appears in the Available hardware items window.
4. Click the Add Hardware button to open the Add Hardware window if the programming hardware you would like to use is not listed in the Available hardware items window.
    a. Select the appropriate programming cable or programming hardware from the Hardware Type list.
    b. Select the appropriate port and baud rate if necessary.
    c. Click OK.
5. Select the programming hardware you would like to use by choosing it in the Available hardware items list.
6. Click Close.
7. Your programming hardware has been set up.

4.4 Configuration to use Switches and Leds:

Rename the input/output pins to SW# (#: 0 to 17), LEDG# (0 to 7) for green LEDs and LEDR#(0 to 17) for red LEDs.

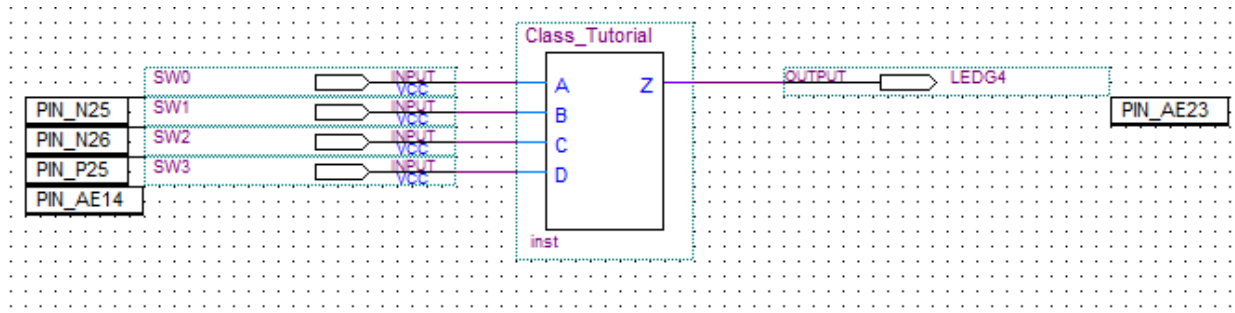
In the pin assignment window (Assignments → Pins)

Rename the nodes and pin numbers to the following entries

	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Group
1	LEDG4	Output	PIN_AE23	7	B7_N0	3.3-V LVTTL (default)		
2	SW0	Input	PIN_N25	5	B5_N1	3.3-V LVTTL (default)		
3	SW1	Input	PIN_N26	5	B5_N1	3.3-V LVTTL (default)		
4	SW2	Input	PIN_P25	6	B6_N0	3.3-V LVTTL (default)		
5	SW3	Input	PIN_AE14	7	B7_N1	3.3-V LVTTL (default)		

The values for Switch location and Led location can be found from the DE2 manual- Table 4.1 (Page 28) and Table 4.3 (Page 4.3).

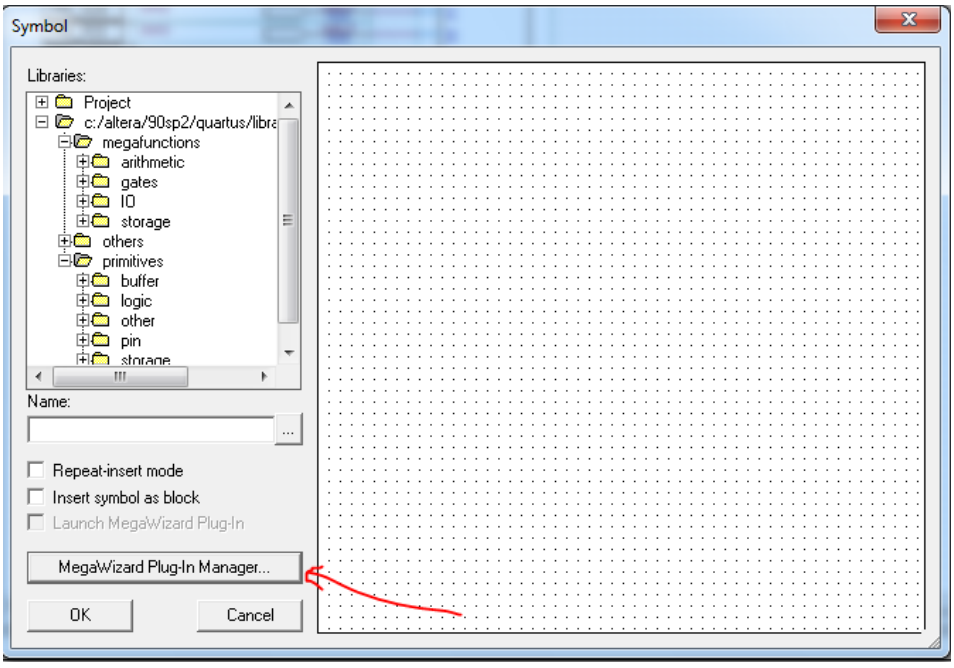
Compile again and download program to test your setup. Do remember to rename the pin names to the corresponding LED/SW names in the schematic as well as simulation window too.



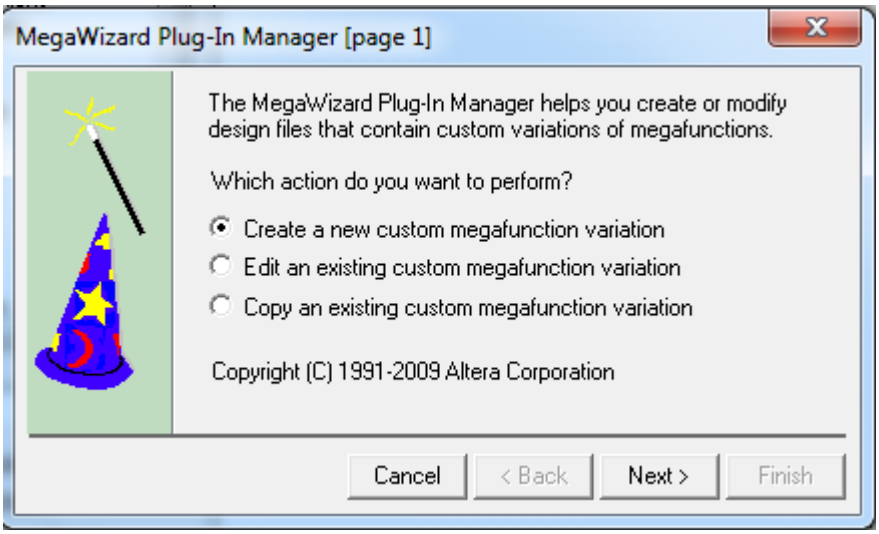
Other pins frequently used are the 7 segment displays (Table 4.4) and the expansion headers (Table 4.7).

5. Additional details regarding using Mega functions:

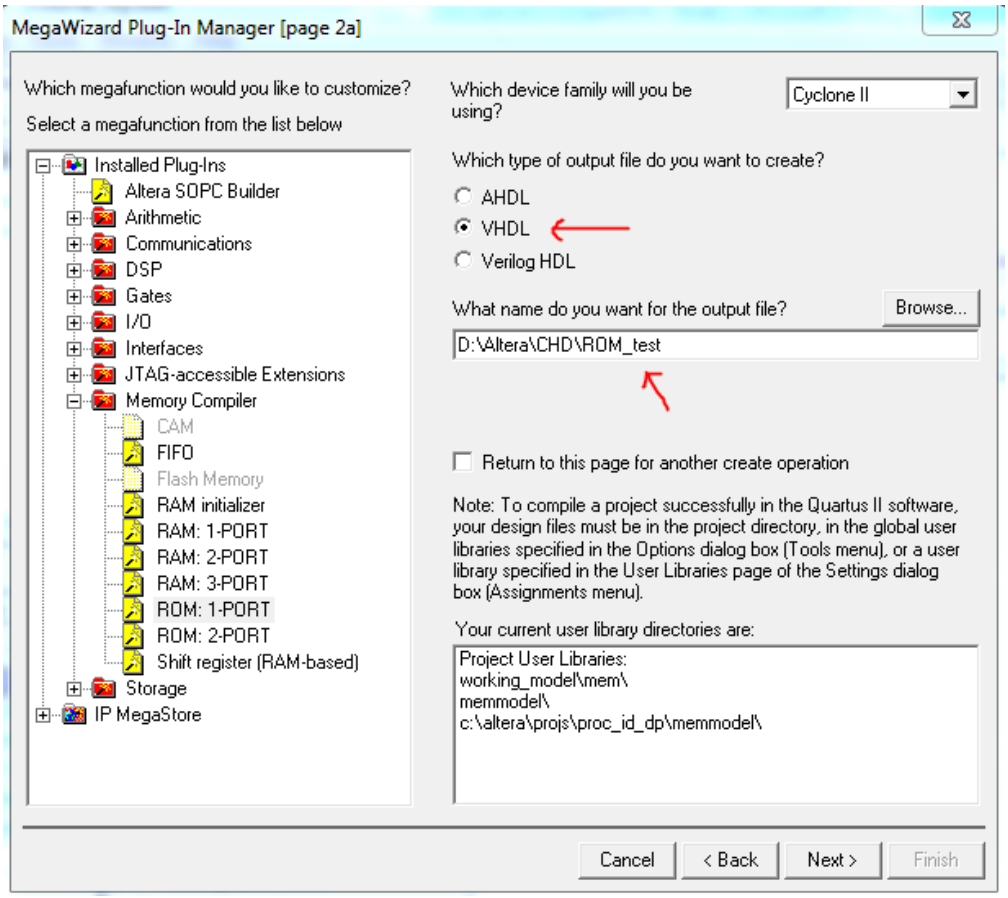
When handling larger designs, this feature of quartus will be quite useful. It provides customized library of bigger functional blocks such as adder, Flipflop array, memory etc. Open the symbol insert window and choose Mega Wizard Plugin manager as shown in the following screen.



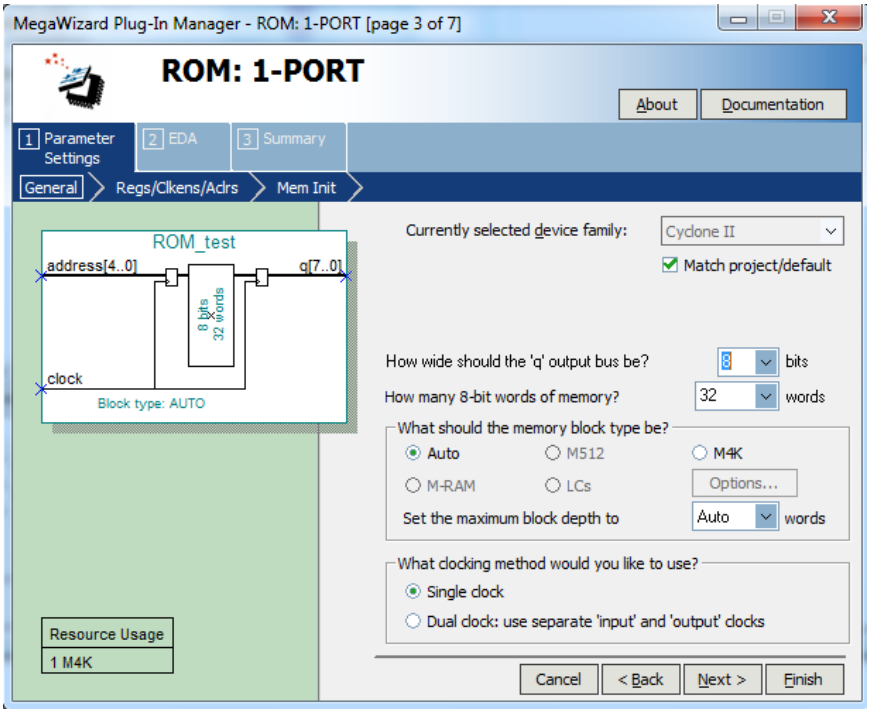
You get a new pop up window where you can create a new custom design or edit an existing design. Choose New and click Next.

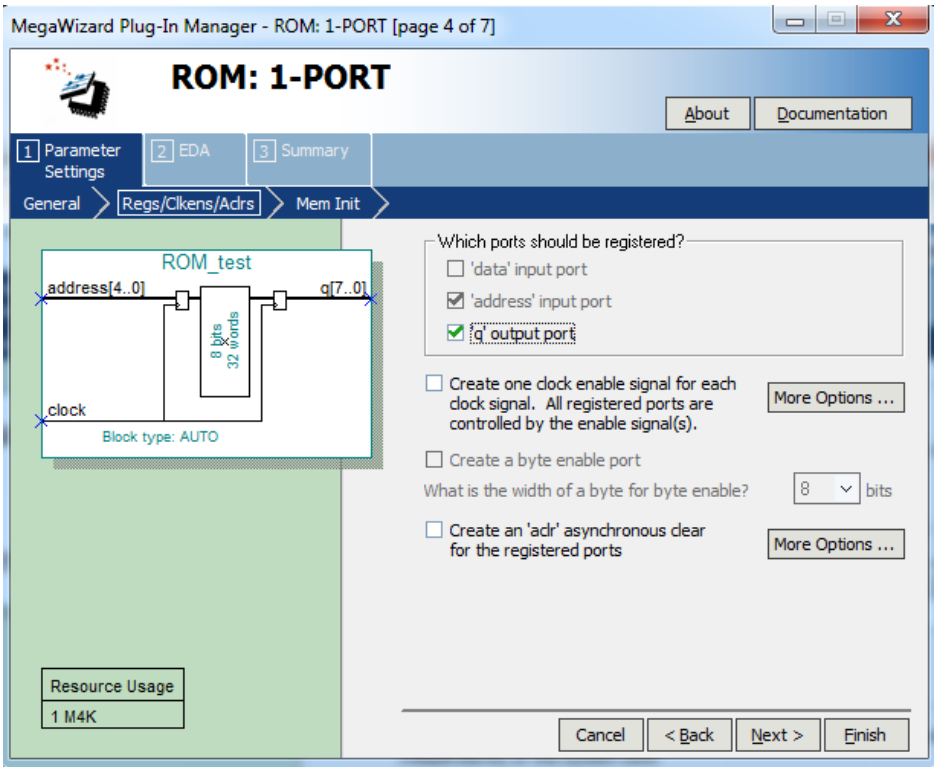


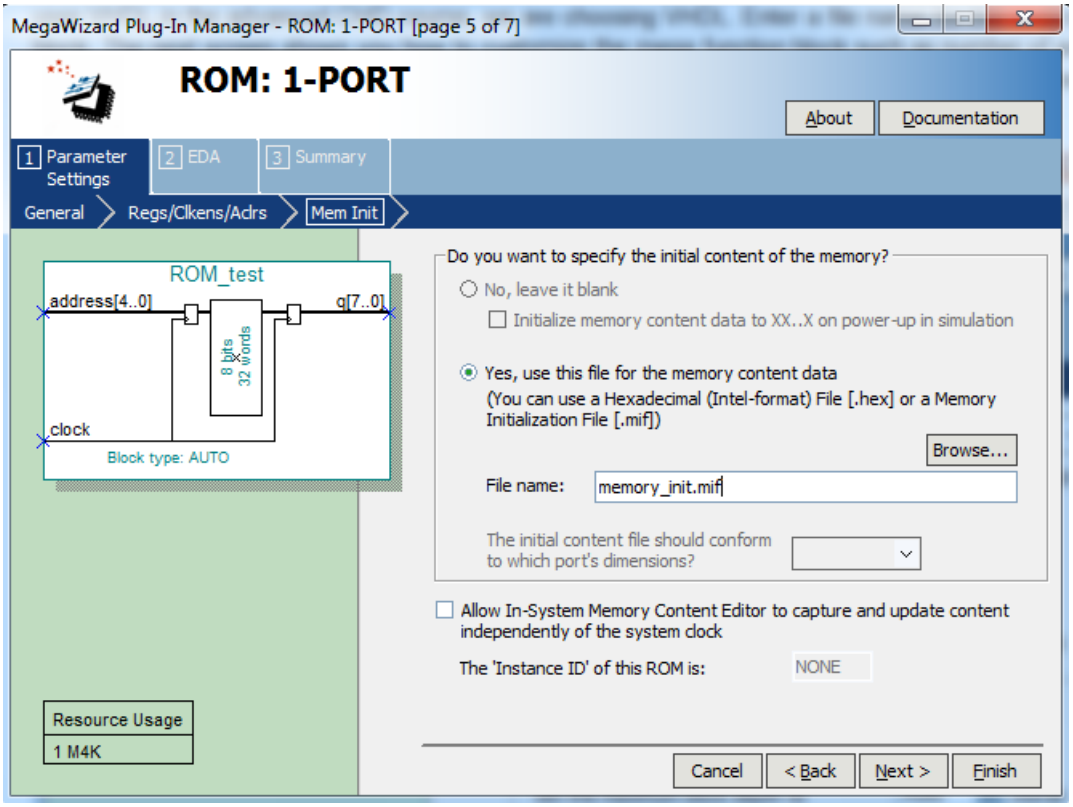
You now get a new screen which has many library functions in the left pane. Choose the type of function you wish to build. For the tutorial example, we are showing how to build a Read Only Memory custom block. In the right pane, you have the option to generate a verilog or vhdl or ahdl file.



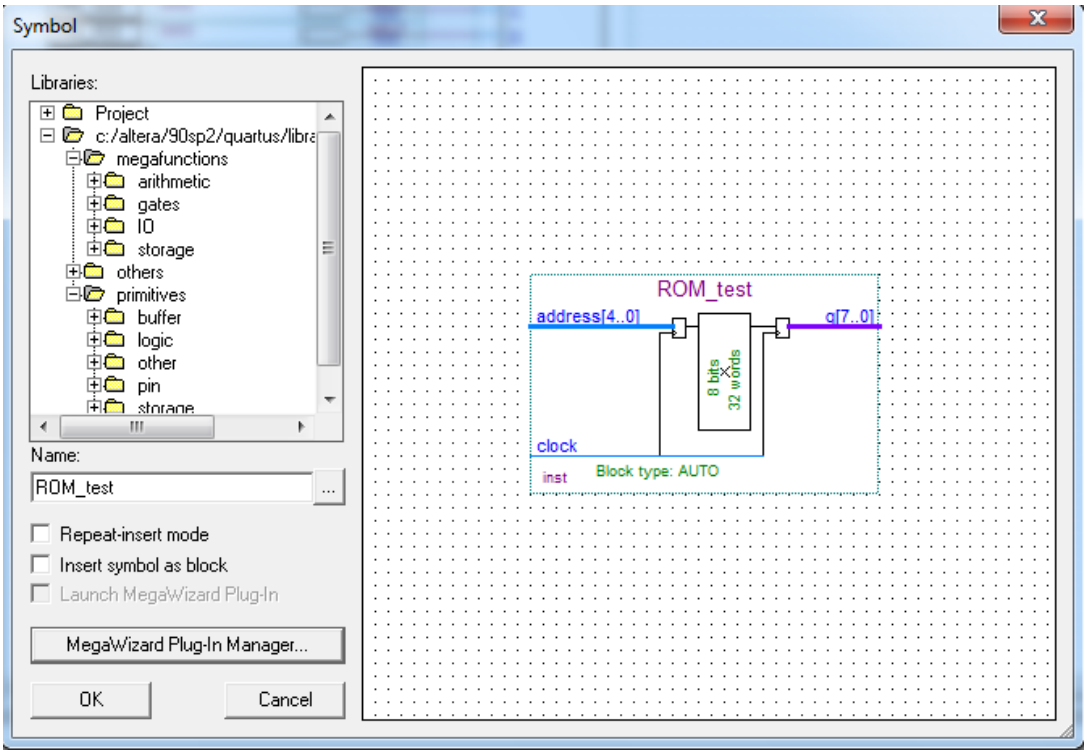
It would be advisable to stay consistent with a particular language throughout the project. Since the class uses VHDL in the advanced CHD course, we are choosing VHDL. Enter a file name for the vhdl custom block. The next screen shows you how to customize the mega function block such as number of address bits, clock enabled, bus width, size of memory. Click Next and proceed through the following screens till the end.







Click Next to the end and finish.



You can now insert this new symbol and proceed with your design.

Summary: This tutorial covered all the introductory aspects of the Quartus Altera tool. Students should be in a position to implement new design and execute them on the FPGA boards.