

# **Design Entry, Simulation, and Emulation using the Intel FPGA Quartus Prime® and ModelSim® CAD Software and the Terasic DE2-115 Rapid Prototyping Platform**

B. Earl Wells and Dennis Hite

## **Introduction**

The Fundamentals of Digital Design class (CPE 322) you will use the Intel FPGA Quartus Prime® computer aided design software to create a schematic diagram or a Verilog representation of logic designs that you have created. You will also often use the integrated Mentor Graphics' Intel FPGA ModelSim® Starter Edition simulator package to aid you in understanding and verifying the functionality of your design. In the co-requisite Advanced Digital Design Lab (CPE 324) you will emulate your design by downloading it into a Field Programmable Gate Array, FPGA, that is present on the Terasic DE2-115 Rapid Prototyping Platform.

## **What is the purpose of this Intel FPGA Quartus Prime®/ModelSim® CAD Software?**

The Intel FPGA Quartus Prime® CAD development software is a fully integrated programmable logic design environment. This easy-to-use tool supports a large array of Intel FPGA® programmable device families and works in both the Microsoft Windows® PC and Linux environments. The tool supports design capture and has the Intel FPGA ModelSim Starter Edition® as a plug in. The ModelSim® software tool supports digital logic simulation and timing analysis.

## **How can I get access to the Software?**

There are two main options for getting accessing the Intel FPGA Quartus Prime/ModelSim® software. One is to use the software that is already installed in the Advanced Logic Design Laboratory in Room 226 of the Engineering building. The software in this laboratory is the full commercial version which is final version that supports the Cyclone IV E FPGA family that is present on the DE2-115 Rapid Prototyping platform in the lab.

There is also a web version of Quartus Prime that you can download and run on your own PC. You can download it from the Intel FPGA web site at

<http://dl.altera.com>

Then select your download method making sure that the Quartus Prime, ModelSim-Intel FPGA Editions, and Devices Cyclone IV E are checked which should be included on the defaults. The Web (Lite) Edition software contains the same basic functionality as the commercial version but it is limited in terms of the devices that are supported and the size of your designs. This option is mentioned only for your convenience and UAH is not liable for any activities concerning the use of the software that you download.

## Getting Started

There are many Quartus Prime® file types, three of the most common types include:

**Block Description Files (bdf)** = schematic or block diagram

**Block Symbols File (bsf)** = symbol or schematic component

**Hardware Description File (tdf, v, vhd)** = AHDL, Verilog, or VHDL, -- alternative higher level ways of capturing your design.

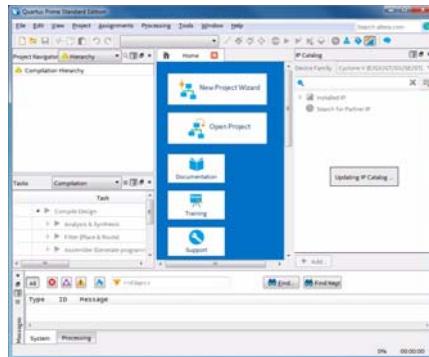
To complete the simulation exercises in this course, you will be concerned mainly with creating Block Description Files (bdf) and Verilog Hardware Description Files (v). Both type of files can appear in the same design. A bdf file is where you will place your logic schematic that represents your design. A Verilog file is where you place your textual description of your design that is written in the Verilog hardware description language. A given design can contain a hierarchy of block description and Verilog files.

After you have created your design you will need to compile it. At this point the design can be simulated to determine its functionality. This can be done using the same computer that you entered it via a schematic/block diagram and or hardware description language. Simulation requires that you drive the inputs of your design in some predetermined and meaningful manner at specified points in time. This is called providing an input *stimulus* to the simulated design. It also requires that you select and view the resulting output that you want to observe. The simulation program will produce an output that should mimic the behavior of the actual design. This is then compared with the expected output. Simulation outputs can be viewed as a timing diagram or in a textual manner. The stimulus can be provided by creating a testbench element of your design that provides the inputs and outputs to your design (this is often expressed using a hardware description language such as Verilog) or through stimulus generation commands that are part of the ModelSim® Simulation Environment.

# I. Setting up the work environment

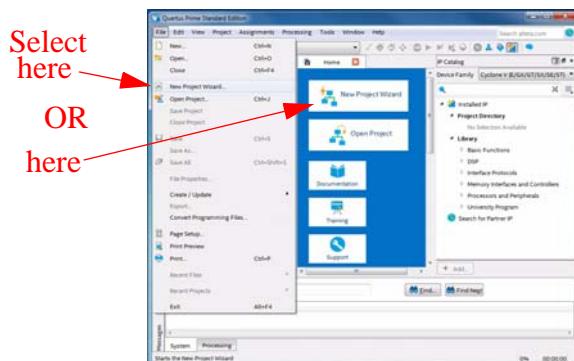
This section describes the steps necessary to utilize the Quartus Prime/ModelSim® environment within the UAH ECE Department's EB 226 Advanced Logic Design lab. Individual student implementations should work in a similar manner. *Throughout this tutorial it is assumed that Windows® maps your flash drive to the device labeled f: If this is not the case you should substitute the actual drive letter for the f: labeling presented here.*

- 1 Before entering the Quartus Prime® CAD software place your personal USB flash drive in the computers USB port. This is where you are to store your design files for the labs in this course. It is suggested that you use a USB flash thumb drive that has a capacity of 512 Mbytes or greater. Make sure that you are careful with this drive and that you exit the Quartus Prime® software and properly stop the drive before you remove it from the PC or you may have loss of data. It is the student's responsibility to properly backup his/her flash drive. Do not put your personal files on the disk drive of the laboratory PCs.
- 2 To open the Quartus Prime® CAD tool package, first double click with the left mouse button on the Intel FPGA Quartus Prime® icon  on your desktop. Note that the exact version number of Quartus Prime may not match the one shown here but this tutorial should still be valid for your version of the software. You will then see a welcome screen that is similar to that in Figure 1.1.



**Figure 1.1**

- 3 Next left-click on the *New Project Wizard* which is under the *File* Menu (or click on it on the splash screen in the middle of the window).

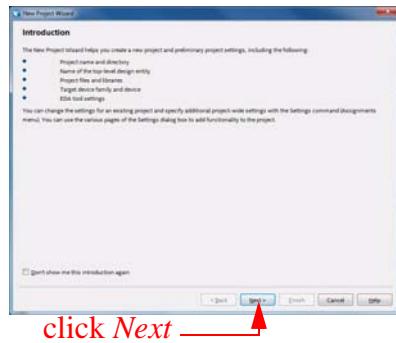


Note: The *Project* is the virtual container that points to all of the important files that are part of your design.

The name of your top-level design file should match your project name.

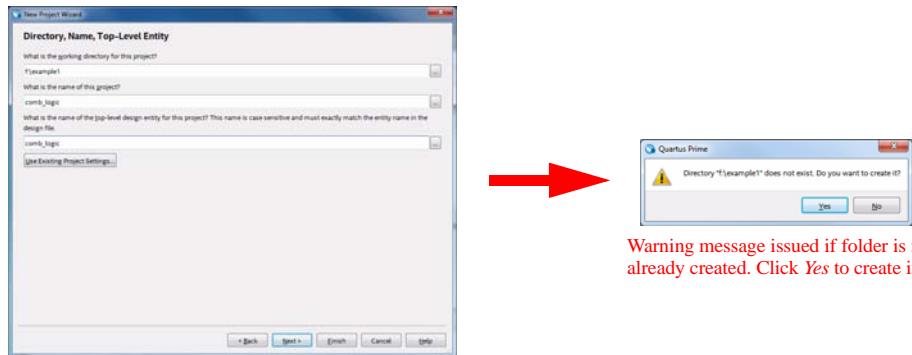
**Figure 1.2**

- 4 An information window similar to the one shown in Figure 1.3 will appear, left-click on *Next* to continue.



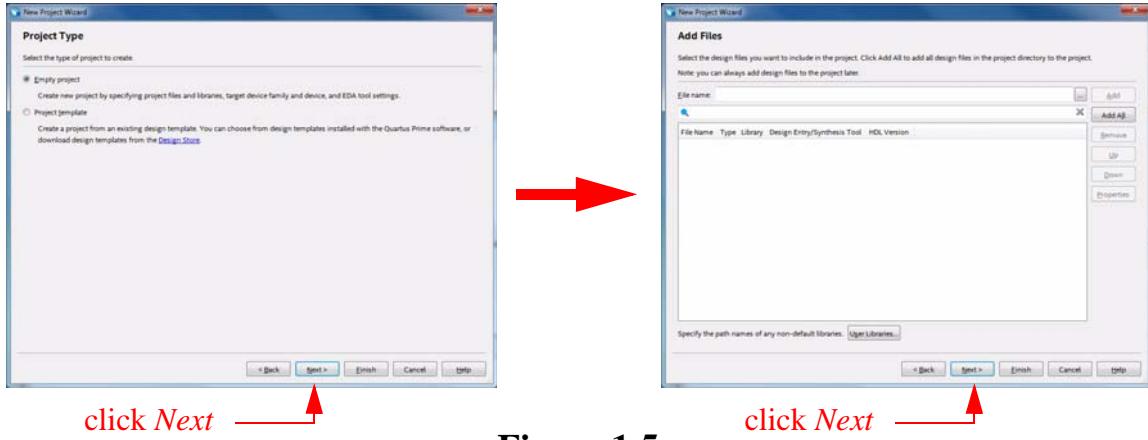
**Figure 1.3**

- 5 A dialog box similar to the one shown in Figure 1.4 will then appear. In the first field of it enter the name of the working directory such as **f:\example1** as shown in the figure. Then enter the name of the project. The project contains all the pertinent information about the design. In this case enter the name **comb\_logic** in the following field. Notice that as you enter this name it also is being entered in the bottom field where will be used to indicate your top-level design file. The Quartus Prime tool supports hierarchical design methodologies that allows the user to use many files to represent the entire design. In this example we will only use a single top-level design file which will have the same name as the project. This is the default so simply left-click on *Next* after you have entered the name of the project in the second field



**Figure 1.4**

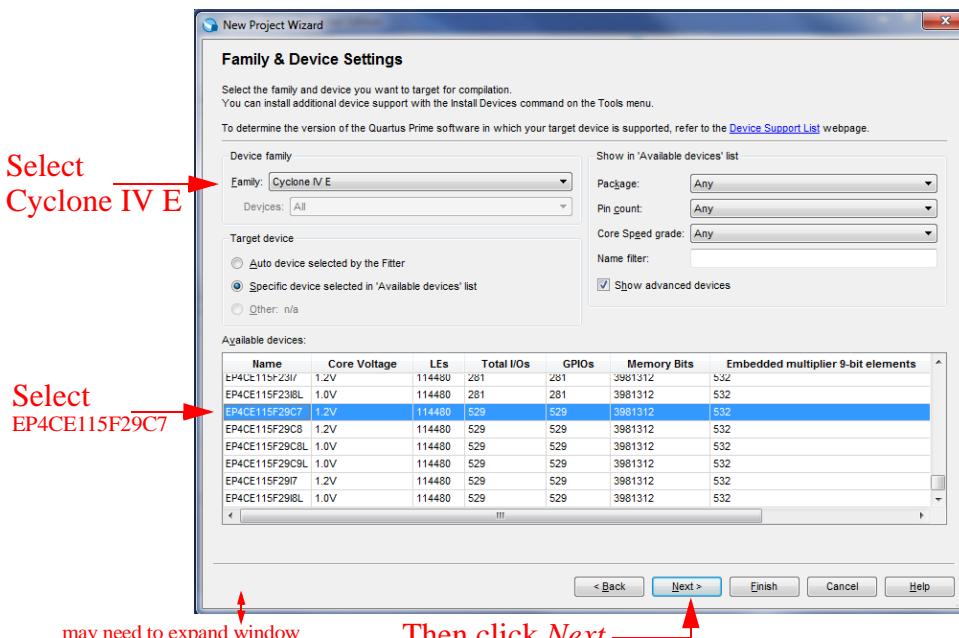
- 6 If the working folder (**f:\example1** in this example) is not present then a warning message window will appear asking you if you want to create the file. In this case simply left-click on the *Yes* button to continue. The next two windows that are spawned by the New Project Wizard are the *Project Type Window* and the *Add Files Window* which are both shown in Figure 1.5. The *Project Type Window* gives you the option to simplify the design entry process by utilizing an existing template and the *New Project Wizard* window gives you the opportunity to add more files to the project.



**Figure 1.5**

In this tutorial it is assumed that you desire to create your design from scratch and that you have not yet created the schematic file (you have no files to add). In both cases you should simply left-click on *Next* button to continue to the next screen.

- 7 The *New Project Wizard* will then display the *Family & Device Setting* window which will allow you to select a programmable logic device to configure with your design (i.e. load your design into). You are to select the Cyclone IV E family which is under the **Device family:** heading. Then select the EP4CE115F29C7 device from the list that is shown under the **Available devices:** heading as shown in Figure 1.6.

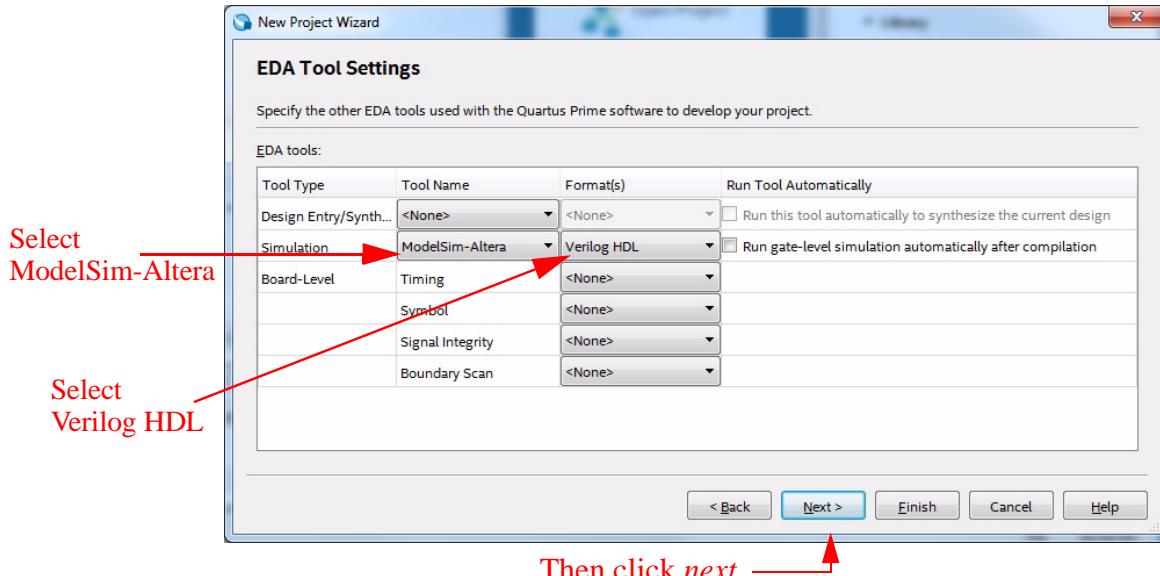


**Figure 1.6**

After the *Family*, and *Target Device* is selected you should left-click on the *Next* button which should advance you to the *EDA Tool Settings* window as shown in Figure 1.7.

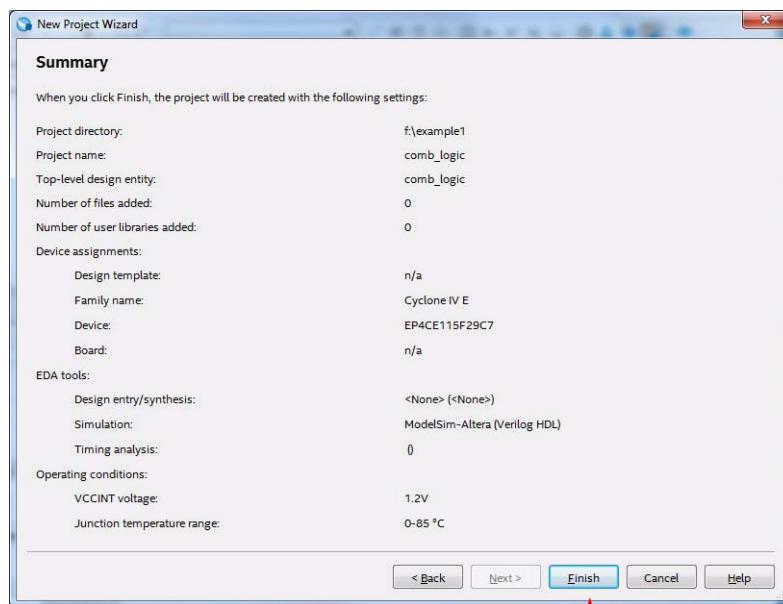
- 8 The *EDA Tool Settings* window is where you can specify the configuration of the “plug-in” type CAD tools you would like to use as part of your design flow. In this case we would like to employ the customized version of ModelSim® that incorporates the Cyclone IV E Intel FPGA library and you would like to specify that this tool utilize Verilog as its simulation environment. This requires that you select the **ModelSim-Altera** option on the *Simulation*

row as the *Tool Name* and on this same row then select the **Verilog HDL** option as shown in Figure 1.7.



**Figure 1.7**

- After this is done left-click on the *Next* button to bring up the *Summary* window as shown in Figure 1.8 which is the final *New Project Wizard* window.



**Figure 1.8**

This highlights the major options which you chose previously for this design including the project directory, project name, FPGA device number, and the EDA tools setup. If this is correct left-click on the *Finish* button, otherwise click on the *Back* button to make the necessary corrections.

## II. Design Entryx

### Part A: Creating a Schematic or Block Diagram Representation of your Design

#### Entering the Block Editor

- 1 After you have exited the *New Project Wizard*, you have the option to enter your design as a logic schematic diagram. Before you do this it is advised that you dismiss the *IP Catalog* sub-window which is on the right size of the screen. It is likely that you will need more space and you will not be using any IP components in this tutorial. To do this click on the upper right  symbol in the subwindow as shown in Figure 2.1.

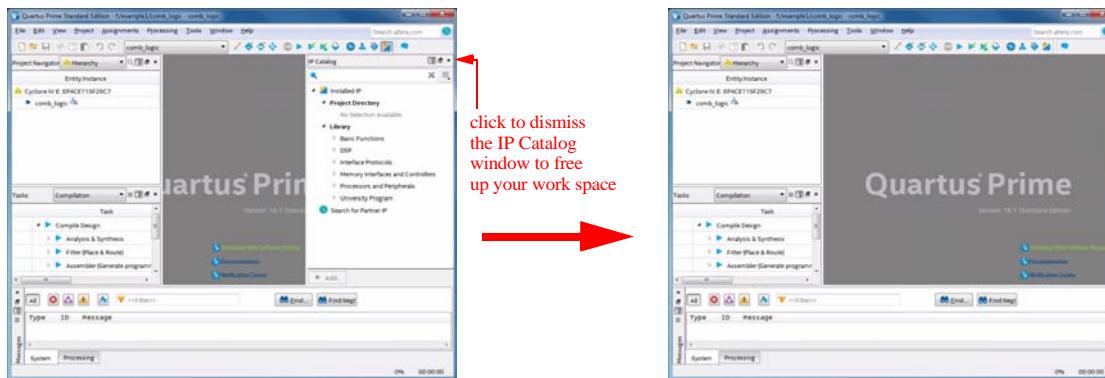


Figure 2.1

- 2 Then to create the schematic select *New* from the *File* menu (or click on the upper left  icon on the top tool bar of the Quartus Prime window). This will bring up the *File Type* box that is shown in Figure 2.2. Then choose the *Block Diagram/Schematic File* option and press the *OK* button.

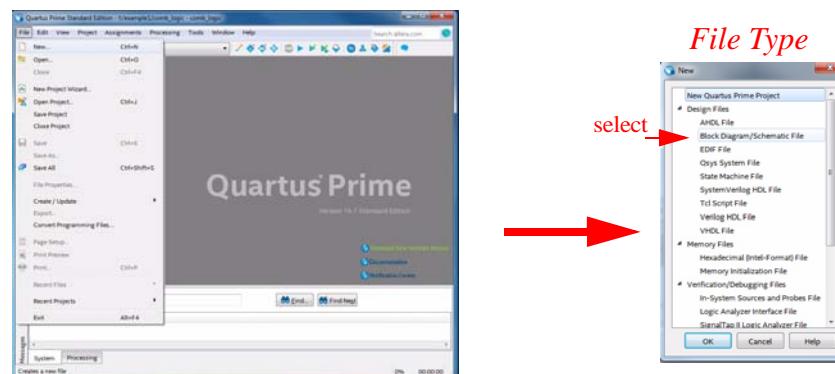
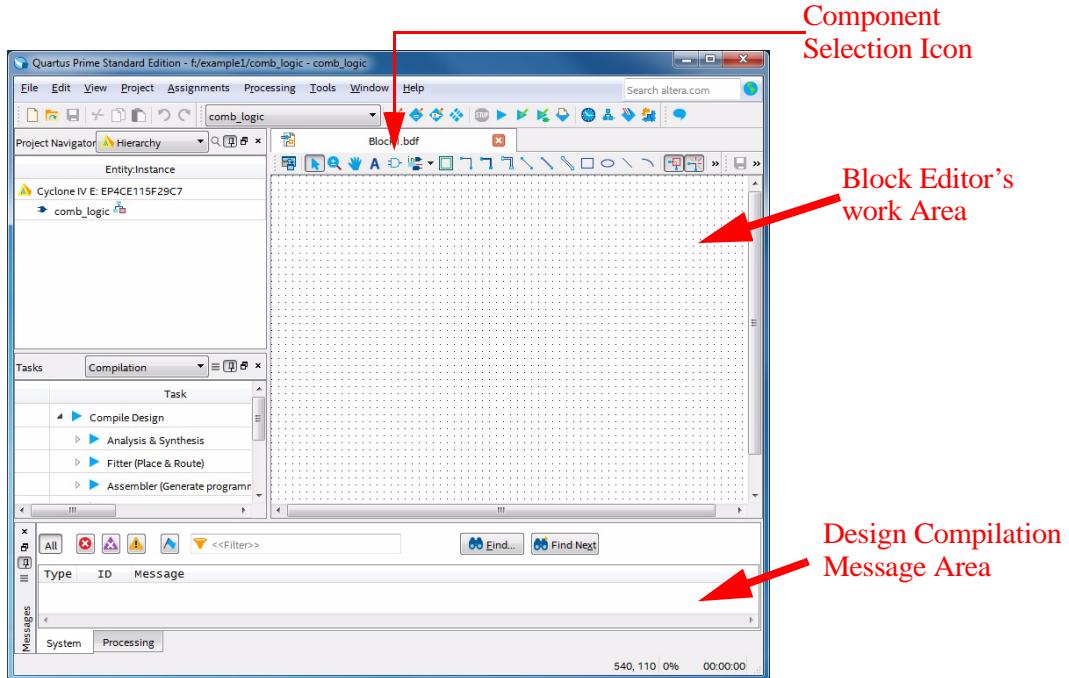


Figure 2.2

- 3 This will bring up the *Block Editor* which is shown in Figure 2.3. On the right hand side of the block editor is the location of the graphics window work area where you will create your schematic. The name of this graphics window is tentatively set by the software as **Block1.bdf** but this should be renamed to the name of your project when you save your schematic the first time. Double-click with the left mouse button somewhere in the *Block Editor*'s work area or left-click on the  icon to select a given component to enter into your design (you can also select the *Insert Symbol* option from the *Edit* Menu).



**Figure 2.3**

## Entering Components

- 1 Before you can enter a component, you will need to choose it from the appropriate component library of symbols. In the previous step you brought up the *Symbol Window* which has a list of such library components in the left hand side of the window. To reveal the components expand upon the *c:/intelFPGA/.../quartus/libraries* entry by left-clicking on the ▷ symbol next to it. This will reveal the **megafunctions**, **other**, and **primitives** component library symbol elements where:

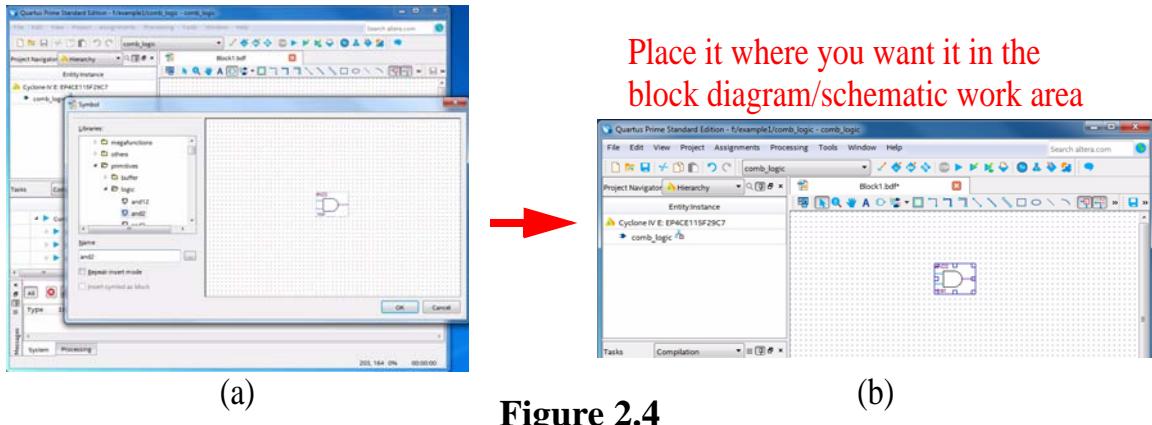
**megafunctions:** (larger customizable parameterizable logic components such as multipliers, adders, decoders, multiplexers, counters, and special i/o components)

**others:** (7400 equivalents, SSI, legacy MSI components)

**primitives:** (gates, flip-flops, I/O pins, VCC, GND symbols, etc.)

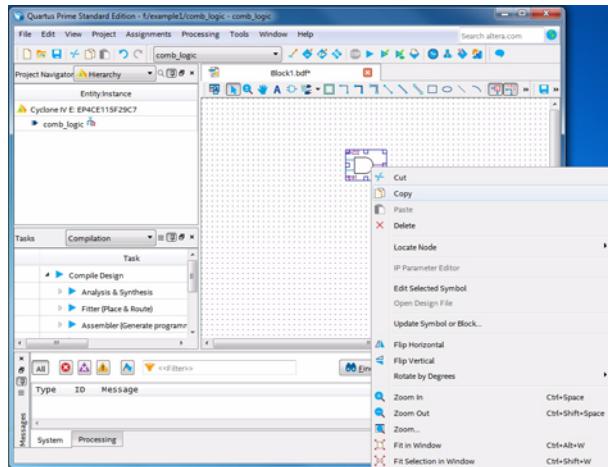
To enter the **comb\_logic** example choose a 2-input AND gate from the **primitives** library. You can do this by expanding the **primitives** library to reveal the **logic** sub-library as shown in Figure 2.4a (i.e. select *c:/intelFPGA/.../quartus/libraries/primitives/logic*). Once the *and2* component is selected left-click the *OK* button and then left-click the place in the *Block Editor* graphics window work area where you would like to put the component as shown in Figure 2.4b.

Select the *and2* library component



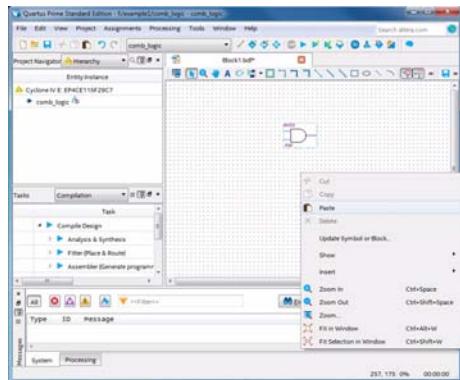
**Figure 2.4**

- 2 In the **comb\_logic** example this design is a simple two-level logic schematic composed of three 2-input AND gates that all feed into a single 3-input OR gate. Once you have selected the first AND gate you can cut-and-paste a copy in the normal windows manner. First, using your mouse, right-click on the *and2* gate that you just added then scroll down and then right-click on **Copy** as shown in Figure 2.5.

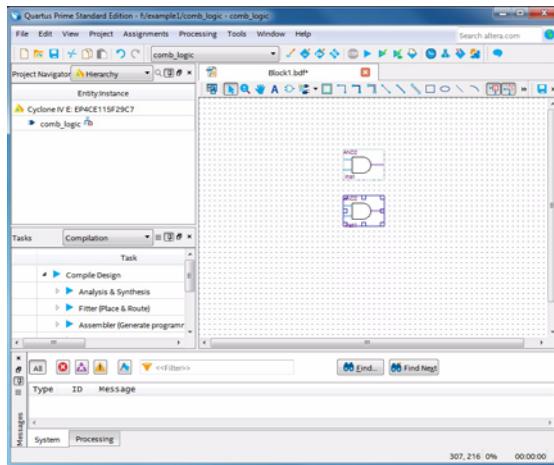


**Figure 2.5**

- 3 Then choose a position below the original *and2* gate, right-click, then scroll down and left-click on **Paste** as shown in Figures 2.6-2.7 (*ctrl+c* and *ctrl+v* also work).

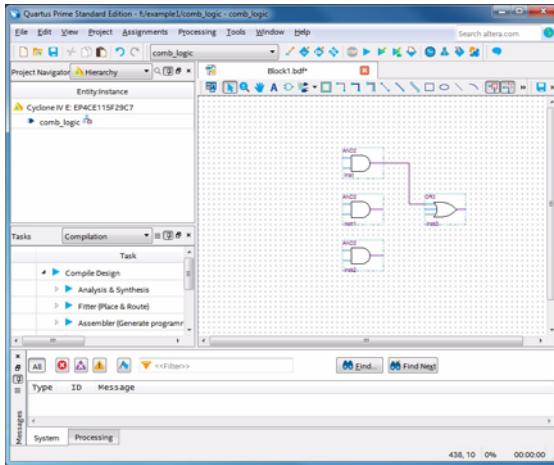


**Figure 2.6**



**Figure 2.7**

- 4 Now repeat the procedure to add a third *and2* gate. You can move components by selecting the component as you hold down the left mouse button and drag-and-drop the symbol to where you want it to go.
- 5 Now add a 3-input OR gate (*or3*) to the right of the second (middle) *and2* gate. Use the same procedure you used to select and add your first *and2* gate.
- 6 Next, wire the outputs of all three *and2* gates to the inputs of the *or3* gate. To add the first wire, point to the output pin of the first *and2* gate, click the left mouse button and hold it down while dragging the wire (line) toward the top input of the *or3* gate. Once the wire is touching the *or3* input pin release the mouse button. Make sure that there are no visible gaps in the wire. Figure 2.8 shows how your design should look after wiring the first *and2* gate to the *or3* gate.

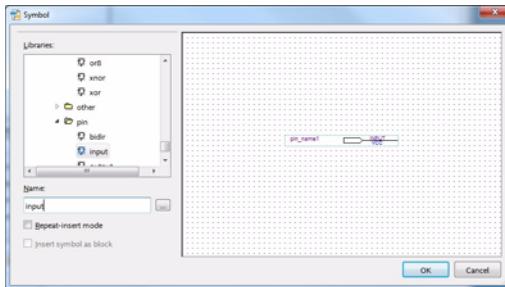


**Figure 2.8**

- 7 Repeat the procedure and connect the remaining outputs of the two remaining *and2* to the unattached *or3* inputs. Note that components and wires may be relocated as needed by dragging and dropping them with the mouse.

### Defining Input and Outputs

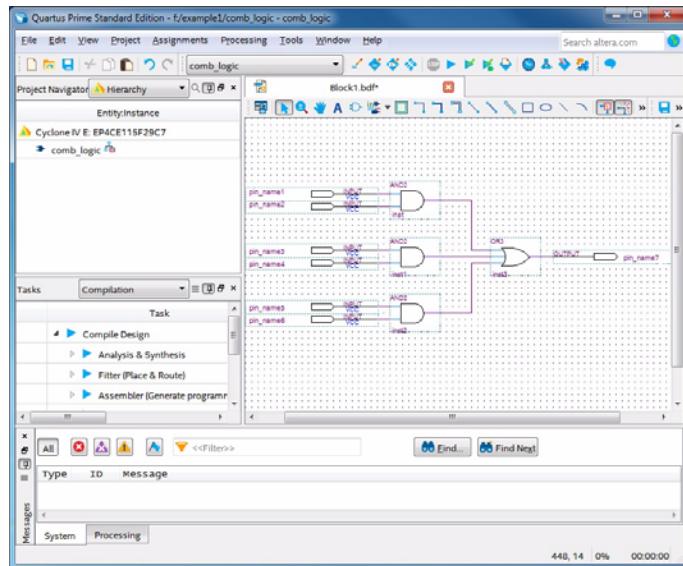
- 1 Your logic design will have one or more input and outputs that define its interface with the outside world. In Quartus Prime®, these I/O lines must be wired-from or wired-into special components that are named *input*, *output*, or *bidir*. It is also desirable to give these I/O components descriptive/user defined names. To accomplish this first double-click using your left mouse button on an empty space within the graphics window work area.
- 2 Next, select the special *input* component from the Intel FPGA component library. You can do this by expanding the **primitives** library to reveal the **pin** sub-library (i.e. select `c:\intelFPGA\...\quartus\libraries\primitives\pin`). Then select the *input* component and left-click on the *OK* button as shown in Figure 2.9.



**Figure 2.9**

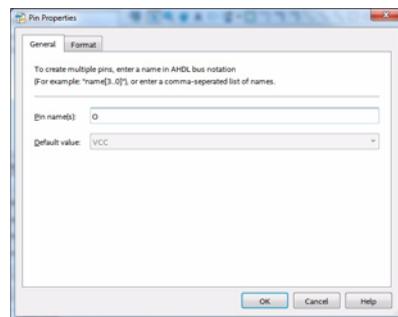
- 3 Then left-click on the place in the graphics window work area of the *Block Editor* where you would like to place the *input* component.
- 4 Make five copies of the *input* component and connect all six of these components to the three *and2* components using wires in the manner described previously.
- 5 Next select an *output* component from the same **pin** sub-library (i.e. select `c:\intelFPGA\...\quartus\libraries\primitives\pin`)
- 6 Connect the *output* component to the output of the *or3* gate.

When you are finished all of your inputs and outputs should be wired as shown in Figure 2.10.



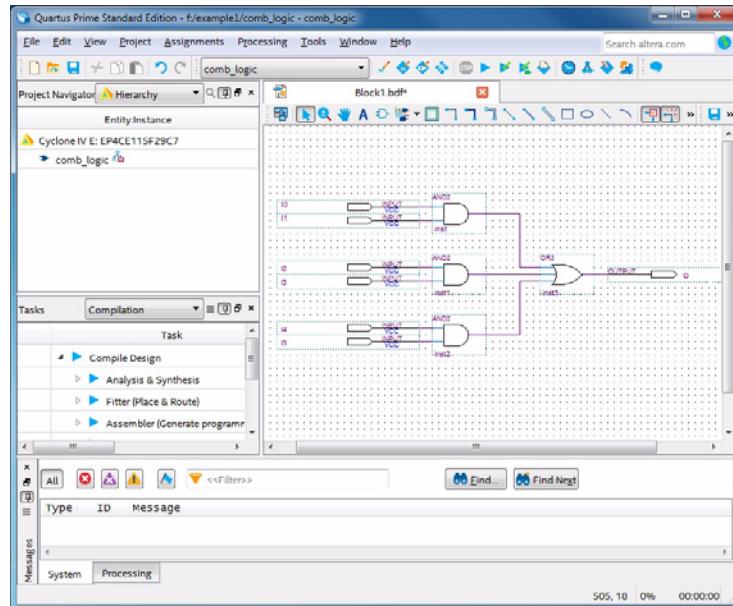
**Figure 2.10**

- 7 The next step is to give user-defined names to your inputs and output. For this example, you are to name your inputs **I0**, **I1**, **I2**, **I3**, **I4**, and **I5**. To do this double-click on the “pin\_name” text of each input symbol and then type the appropriate new name. In the same manner name the *output* component **O**. If you miss the *pin\_name* area of the *input* or *output* components but click on the component itself a *Pin Properties* dialog box will appear and the name can be entered in the *Pin name(s)*: field as shown in Figure 2.11. In this case, just click on the *OK* button after entering the name.



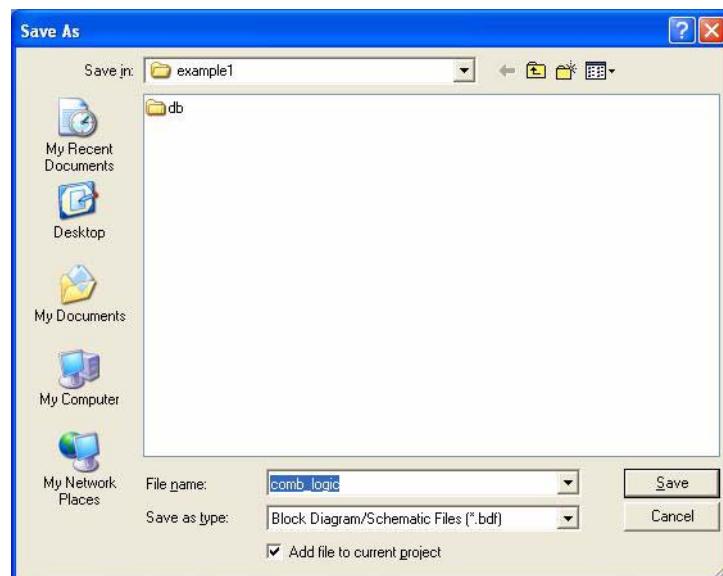
**Figure 2.11**

The final schematic is shown in Figure 2.12.



**Figure 2.12**

- 8 Now you need to save the schematic (bdf) file and include it in your project. To accomplish this select *Save* from the *File* menu or left-click on the icon on the tool bar. This will bring up the *Save As* window as shown in Figure 2.13. Notice that the default file name is the same as the project name (**comb\_logic**) and the *Add file to current project* option is automatically selected. This is what you desire so simply accept the defaults and left-click on the *Save* button.



**Figure 2.13**

Congratulations, you have now completed your first schematic block diagram.

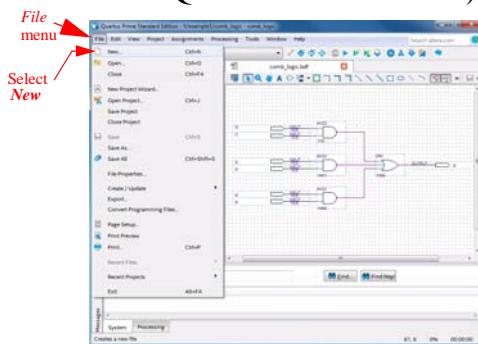
## Creating a Design Hierarchy in your Schematic Representation

The schematic representation of Figure 2.12 can be considered to be a “flat” design because it utilizes only low-level logic primitives which are interconnected together on a single sheet to express the desired functionality. While this is an acceptable way to express very simple designs, for most situations creating a flat design would be too complicated, and error prone and even if successful would result in a representation that would be impossible to understand or maintain. One solution to this complexity management problem is to employ functional decomposition techniques that partitions the design along a multi-level design hierarchy. At the top level there can be a basic block diagram of the system where each block represents a high-level function. As one transverses to the next level by traversing “into” one of the blocks then one will view a schematic of the subsystem that is represented by that block. As one transverses down the hierarchy one would expect to eventually find low-level schematics with all components for that module being described at a primitive component level (AND OR, NOR, etc.).

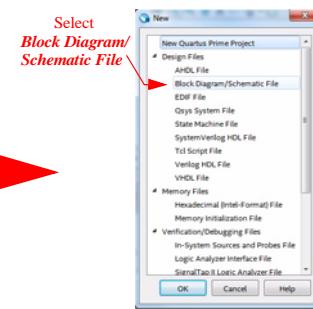
It should be noted that within this paradigm commonly used logic configurations are often encapsulated into modules that are called components. Each time this logic configuration is needed then a copy of that component can be placed in the design (instantiated) at the specified point in the hierarchy from a library in the same manner as the primitives. Components can come in prepackaged libraries or the user can add them to the current project library. Common components include multiplexers, decoders, encoders, latches, buffers, ROMs, RAMs, ALUs, and even CPUs.

To demonstrate how to create a hierarchical design let us now assume that we would like to use our entire previous design as a component building block. While the simplicity of this function does not lend itself to widespread use the same steps should be followed when creating other more useful component modules or blocks in a multi-level hierarchy.

- 1 Enter your existing project and open up a new bdf file. To do this left-click on *New* which is under the *File Menu* as shown in Figure 12.14 (or click on the upper left  icon on the upper tool bar of the Quartus Prime window).



**Figure 2.14**

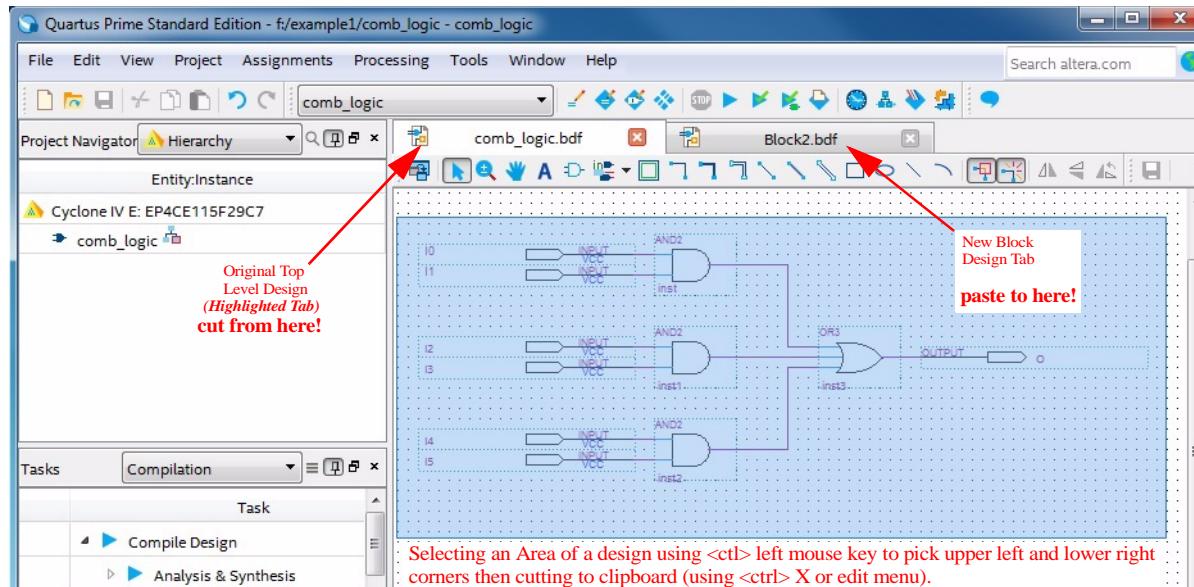


**Figure 2.15**

The *New* window should appear as shown in figure 2.15. Choose the *Block Diagram/Schematic File* option and press the *Ok* button. A new block editor tab should appear as shown in Figure 2.16. It is tentatively given a default name (**Block2.bdf**).

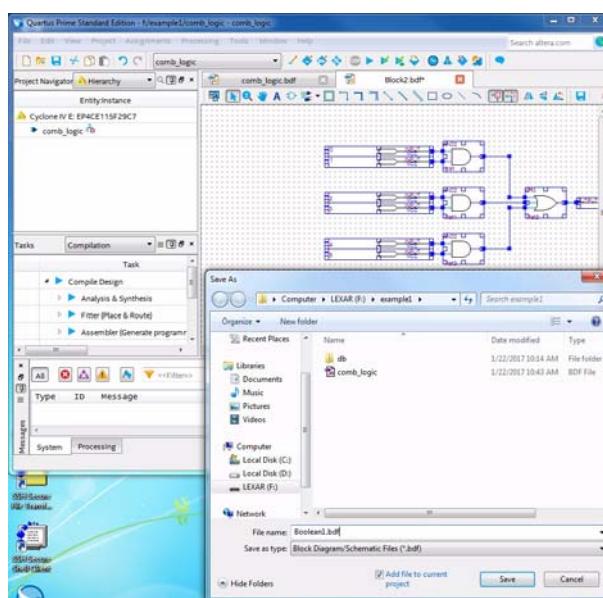
- 2 The next step is to enter the logic associated with the new block you are creating. In general you will create a schematic in a similar way that you did for your flat designs, entering components and I/O pins and interconnecting them using the Quartus Prime methods described previously. The major difference is that your I/O pins will become component pins, not general I/O pins with the component pins matching the names that you have given these pins.

- 3 For this example, to place all the functionality of your original design into this block without reentering it from scratch you can simply click on the **comb\_logic.bdf** tab. Then cut your original top level design into the clipboard. After which you can select the new block tab (**Block2.bdf**) and then paste the design into the work area of the new block. One way to do this is to click on the upper left-hand corner of your design that is under the **comb\_logic.bdf** tab with your left mouse button while holding down the **<ctrl>** key and bring the mouse down to the right hand corner after which you release the mouse button. All elements should be highlighted. Then you can cut the item and place it in your clipboard. Do this by pressing **<ctrl> X** or through the edit menu. Then go to the new block (**Block2.bdf**) tab and paste your design into this block.



**Figure 2.16**

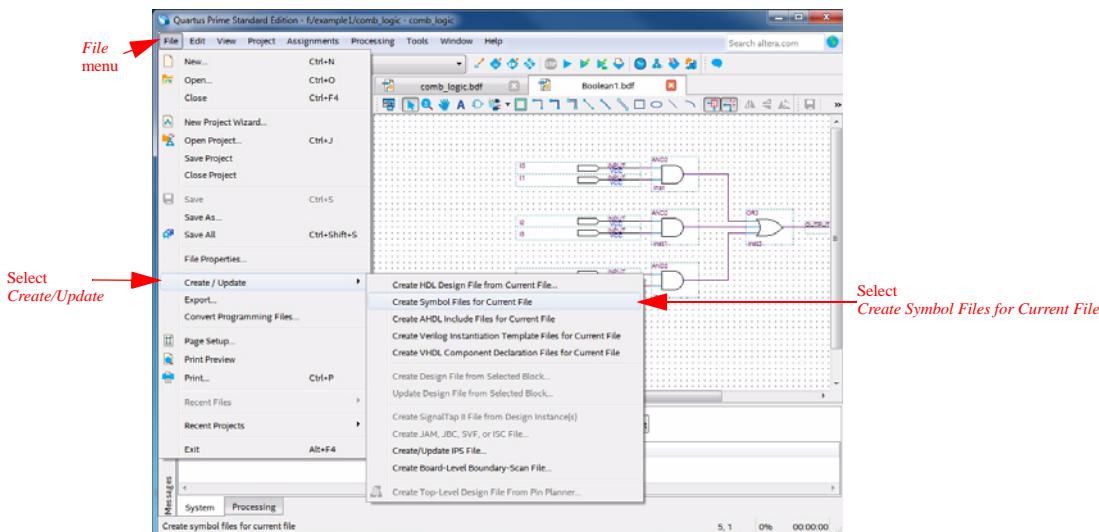
- 4 The next step is to save this new block file and include it in your project. To accomplish this select *Save* from the *File* menu or left-click on the  icon on the tool bar. This will bring up the *Save As* window as shown in Figure 2.17.



**Figure 2.17**

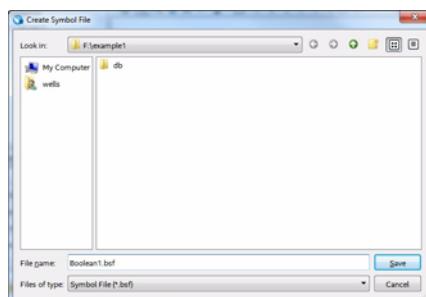
Initially the default name, **Block2.bdf**, should be highlighted. Change it to something more meaningful such as **Boolean1.bdf** as shown in the figure. This will also become the name of the component module. Make sure that the *Add file to current project* box is checked and click the *Save* button. The block tab's name should now appear as **Boolean1.bdf** and the asterisk should have disappeared indicating that it has been saved. The module has now been saved and added to the project but before it can be used as a component a symbol will have to be created.

- 5 The next step is to create a component symbol for this block. To do this first make sure that the block tab (**Boolean1.bdf** in the example's case) is selected. Then chose the *Create Symbol Files for Current File* option that is under the *Create/Update* submenu of the *File* menu as shown in Figure 2.18.

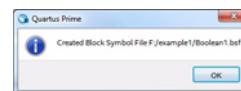


**Figure 2.18**

This will bring up the *Create Symbol File* dialog box as shown in Figure 2.19. By default the symbol name will be the same as the module file name (without the .bdf extension). Click the *Save* button to continue.



**Figure 2.19**



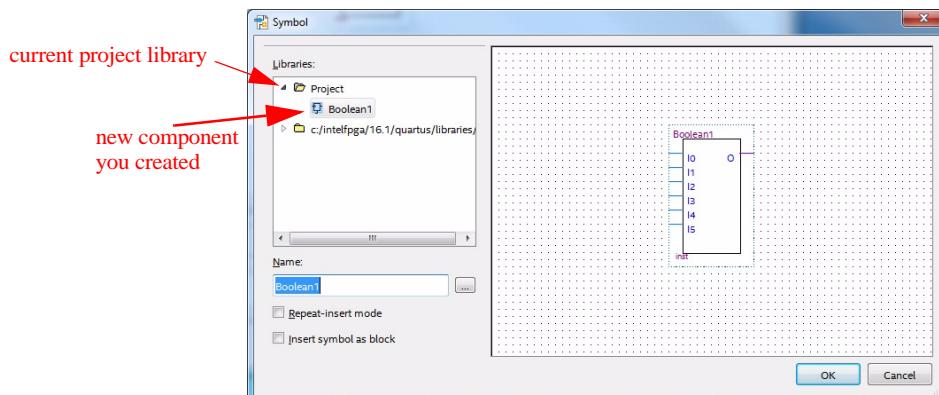
**Figure 2.20**

If there are no errors then Quartus Prime will generate a status window as shown in Figure 2.20 that indicates that the Block Symbol file was successfully created. Click *Ok* to continue.

- 6 The next step is to actually use the component that you have created in the design. This is sometimes called instantiating the component. In general to do this requires that you go to the place within your design hierarchy that you need the component and then simply select

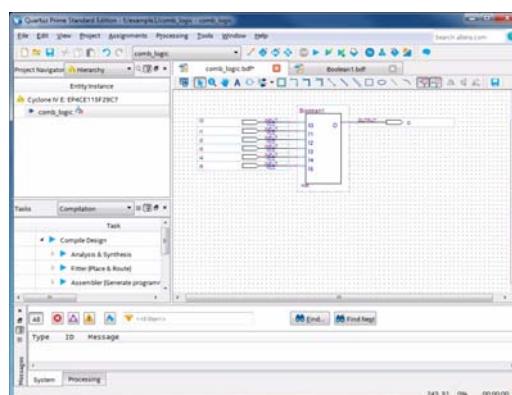
the component from your project library that is associated with your design. The component will be interconnected to others in the normal manner.

In this example we would like to replace our original design at the top level with the new component that we have just created. To do this go back to the **comb\_logic.bdf** tab. It now has an asterisk because we have modified the original top level design by cutting the circuitry. Double click in the white space or left-click on the  icon in the normal manner to reveal the new component you have created (you can also select the *Insert Symbol* option from the *Edit Menu*). You should now see a *Project* folder above the other folders in the library area as shown in Figure 2.21.



**Figure 2.21**

Expand this folder by clicking the  symbol. You should see the new component **Boolean1**. Click on this and click on the *OK* button. You can now place this in your top level schematic as you would any other component. Notice the pin names on the symbol are the same as your pin names on the lower level of the design. To create an equivalent two level hierarchical design that is equivalent in functionality to your original flat one simply connect up the inputs and the output to I/O connector components (input and output) and name them in the same way as you did previously. Figure 2.22 illustrates the resulting implementation.



**Figure 2.22**

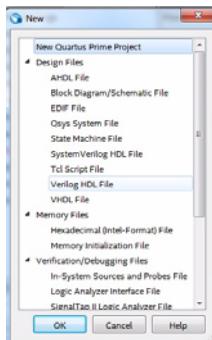
After you have saved this file you can transcend down the hierarchy by simply double clicking with your left mouse button on the individual components. In this case, double clicking on the **Boolean1** component will cause you to descend down to its gate level representation that you entered previously making active its tab in the bdf editor. This is true of any design component regardless of whether it is user entered or part of a preexisting library.

## **Part B: Creating a Verilog HDL Representation of your Design**

Often you may desire to create a design that is written entirely in the Verilog hardware description language to take advantage of this language's portability, flexibility, and ability to model hardware at multiple levels of abstraction. To use this method of design capture follow the following steps.

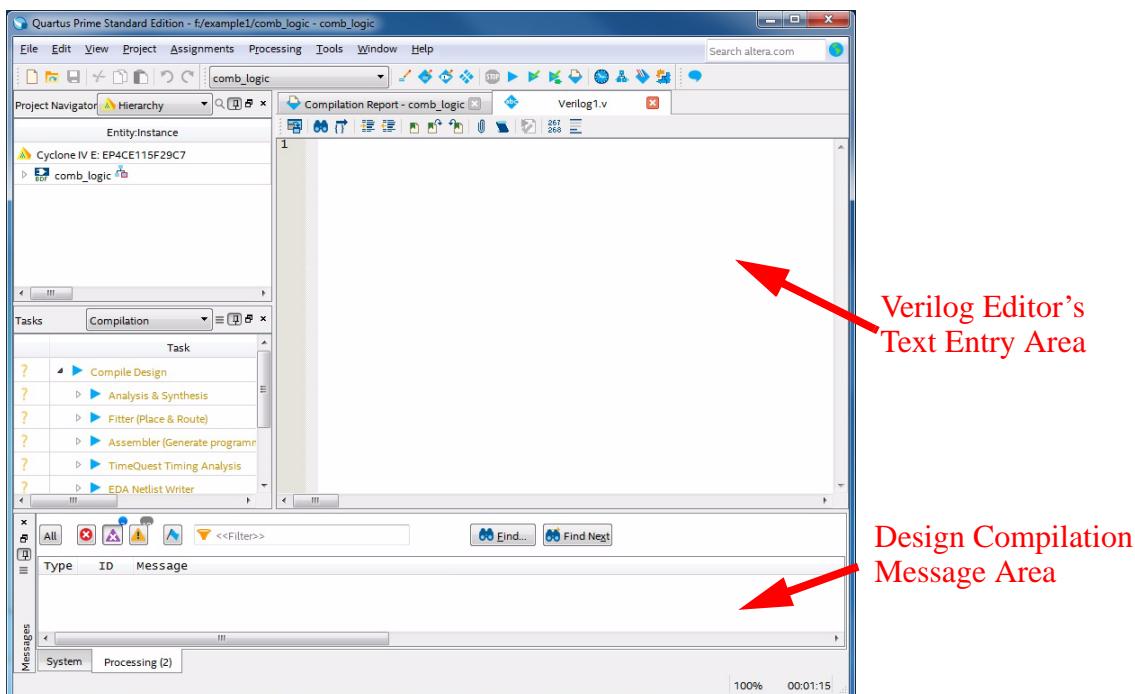
### **Entering the Verilog Model Editor**

- 1 First create your project using the *New Project Wizard* as described in Part 1 of this tutorial.
- 2 You will then use a special text editor to allow you to enter your design as a Verilog HDL module. To do this, first select *New* from the *File* menu. This will bring up the file type box that is shown in Figure 2.23. Then highlight the *Verilog HDL File* option and press the *OK* button.



**Figure 2.23**

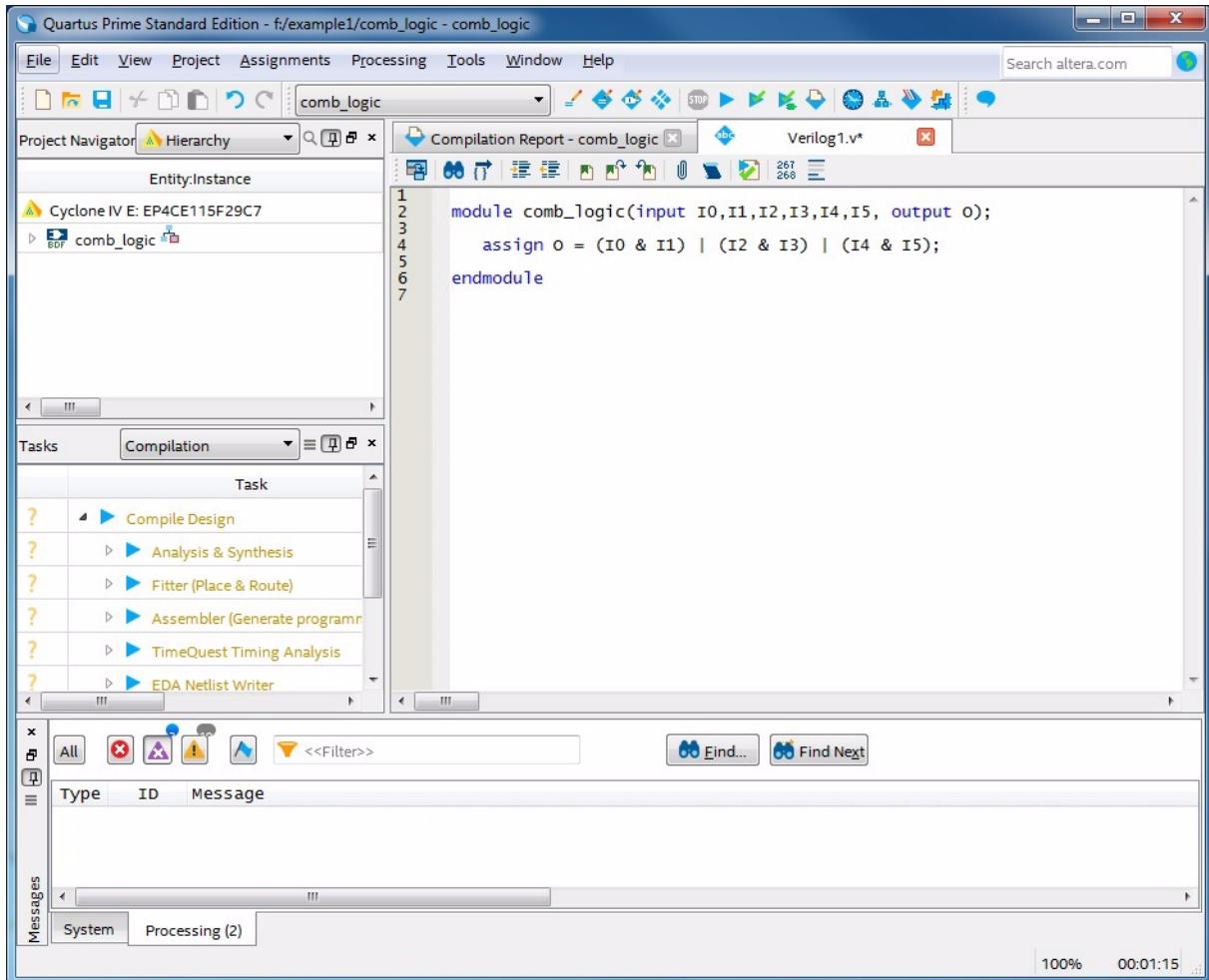
- 3 This will bring up the *Verilog Text Editor* as shown in Figure 2.24.



**Figure 2.24**

On the upper right hand side of this editor is the location of the text entry area where you will create your Verilog HDL model. The name of this text window is tentatively set by the software as *Verilog1.v* but this will automatically be renamed to the name of your project when you save your design for the first time. To begin entering your model simply left-click with

your mouse into the *Verilog Editor*'s work area. For the **comb\_logic** example a *continuous assignment* style Verilog representation is shown in Figure 2.25 below. This is equivalent in functionality to the schematic capture version of this design that was presented in Figure 2.12 in the previous section.

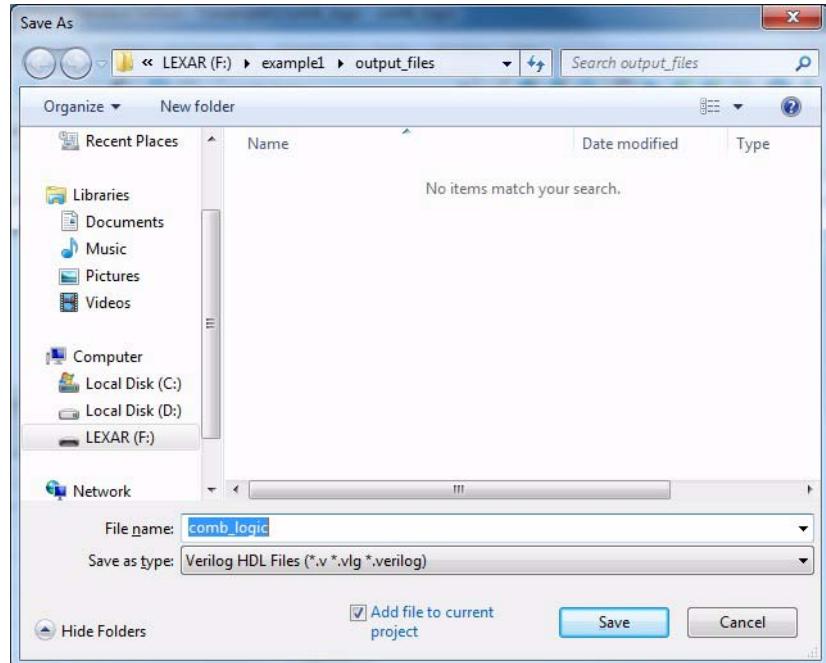


**Figure 2.25**

Note that the keywords in Verilog are highlighted in blue. User defined names are shown in black. If this is the top level of your model, the signals that are declared as module **input**, **output** or inout will automatically form the inputs and outputs of your design of your targeted FPGA representation. There is no need to name it further. This will be the simulation signal name and the emulation pin name that you will use during pin assignment during the configuration. ***In the past it has been a requirement that the top-level module name of the Verilog model be the same as your Quartus Prime project name.*** This requirement has been relaxed but it is probably good design practice to keep this original convention.

- 4 The next step is to save the Verilog model as a v file and include it in your project. To accomplish this select *Save* from the *File* menu or left-click on the icon on the tool bar. This will bring up the *Save As* window as shown in Figure 2.26. Notice that for the Verilog version of the **comb\_logic** example the default file name is the same as the project name (**comb\_logic**)

and the *Add file to current project* option is automatically selected. This is what you desire so simply accept the defaults and left-click on the *Save* button.



**Figure 2.26**

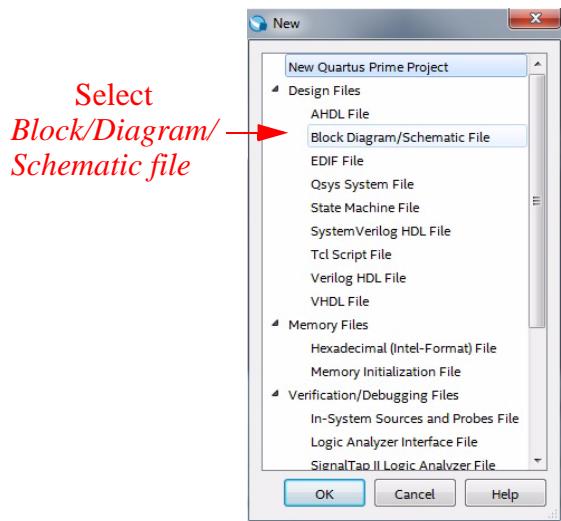
Congratulations, you have now completed your first Verilog design. You are now ready to compile the project. You may have to re-enter this step if there are errors or warnings or if you discover errors in your design during the simulation, timing analysis, or emulation phases of the design process

## **Part C: Creating a Hybrid Schematic Capture/Verilog HDL Design**

It is common practice to incorporate both schematic capture and HDL modules within the same design. Such a design by its very nature is hierachal. While it is possible to have Verilog HDL as your top module and then to instantiate bdf derived components within Verilog itself the more common approach is to create a block diagram representation at the top level and to then instantiate within this framework modules that are represented in Verilog HDL. This section highlights how one would proceed with the later approach -- creating a hybrid schematic capture/Verilog HDL representation where the top level of the design is a schematic. In many ways this is done in the same manner as was discussed in the subsection entitled “Creating a Design Hierarchy in you Schematic Representation” of Part A but with Verilog HDL techniques used to create new modules instead of Schematic Capture techniques. The steps to follow are outlined below:

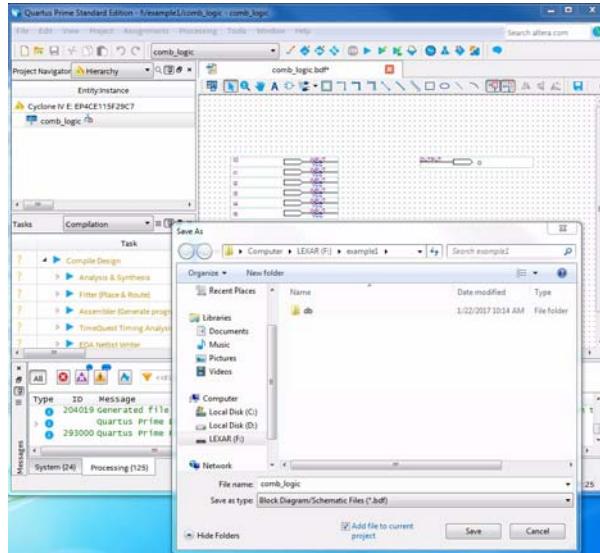
### **Entering the Verilog Model Editor**

- 1 First create your project using the *New Project Wizard* as described in Part 1 of this tutorial.
- 2 Enter your design as a schematic diagram. To do this, first select *New* from the *File* menu (or click on the upper left  icon on the top tool bar of the Quartus Prime window). This will bring up the file type box that is shown in Figure 2.27. Then choose the *Block Diagram/Schematic File* option and press the *OK* button.



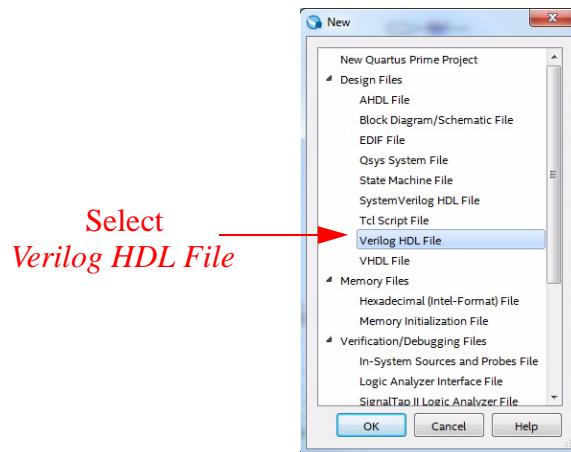
**Figure 2.27**

- 3 This step involves saving the top level schematic design file. This design file will eventually contain the highest level components and modules that will make up your design but many of these modules and components may not be present the first time you enter your design. You must first create these modules and add their symbols to the project library. At this point in time you can go ahead as save the file which has the same name as the project that you created in Part 1 of this tutorial. Quartus Prime will not allow you to save a blank version of the file so you can enter some of the necessary components such as the *input*, *output*, or *bidir* connectors. After this is done select *Save* from the *File* menu or left-click on the  icon on the tool bar. This will bring up the *Save As* window as shown in Figure 2.28. Notice that the default file name is the same as the project name (for this base example, **comb\_logic.bdf**) and the *Add file to current project* option is automatically selected. This is what you desire so simply accept the defaults and left-click on the *Save* button.



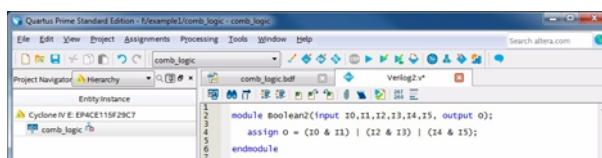
**Figure 2.28**

- 4 The next step involves entering a new Verilog HDL or schematic capture representation of each specific building block module. In the case of a Verilog HDL module select *New* from the *File* menu. This will bring up the file type box that is shown in Figure 2.29. Then highlight the *Verilog HDL File* option and press the *OK* button.



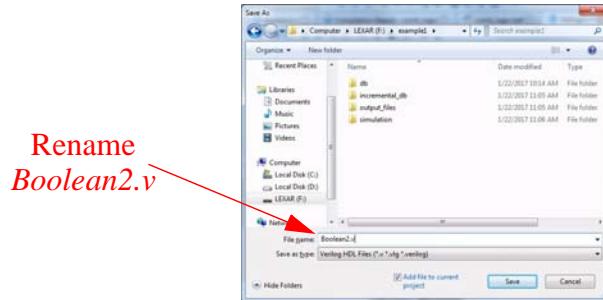
**Figure 2.29**

Then under the tab that contains the Verilog text editor enter you design giving the top-level module a unique name. For the example design we will give the module the name Boolean2. In all other aspects this Verilog file is the same as in the Verilog only representation of Part B as shown in Figure 2.30.



**Figure 2.30**

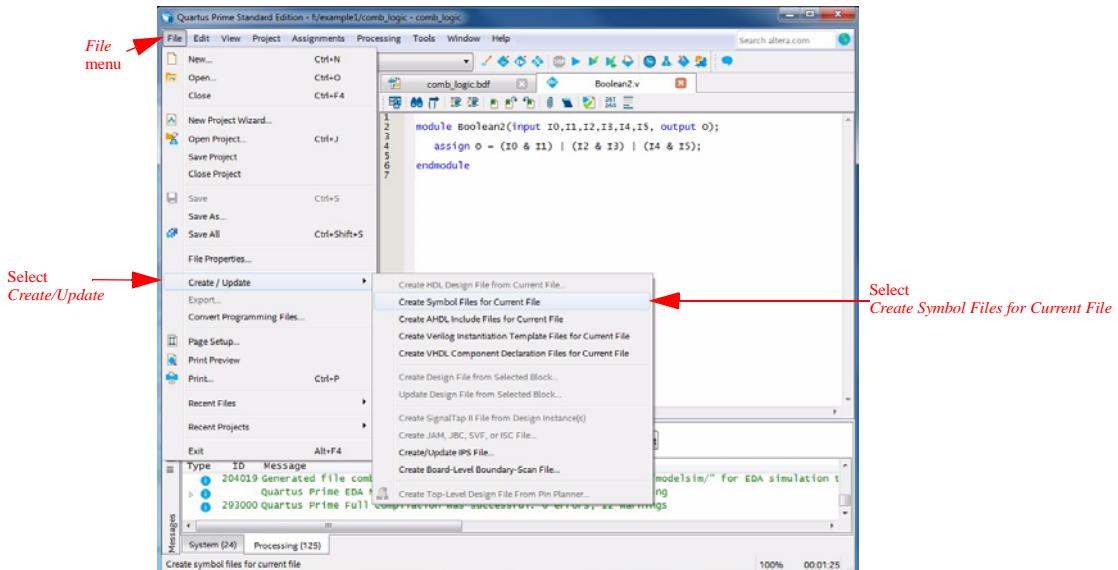
- 5 The next step is to save this new Verilog HDL file and include it in your project. To accomplish this select *Save* from the *File* menu or left-click on the  icon on the tool bar. This will bring up the *Save As* window as shown in Figure 2.31.



**Figure 2.31**

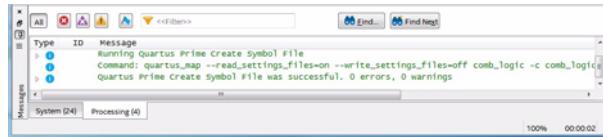
Initially the default name, **Verilog1.v**, should be highlighted. Change it to something more meaningful such as **Boolean2.v** as shown in the figure. This will also become the name of the component module. Make sure that the *Add file to current project* box is checked and click *Save*. The modules tab's name should now appear as **Boolean2.v** and the asterisk should have disappeared indicating that it has been saved. The module has now been saved and added to the project but before it can be used as a component a symbol will have to be created.

- 6 The next step is to create a component symbol for this module. To do this first make sure that the module tab (**Boolean2.v** in this case) is selected. Then chose the *Create Symbol Files for Current File* option that is under the *Create/Update* submenu of the *File* menu as shown in Figure 2.32.



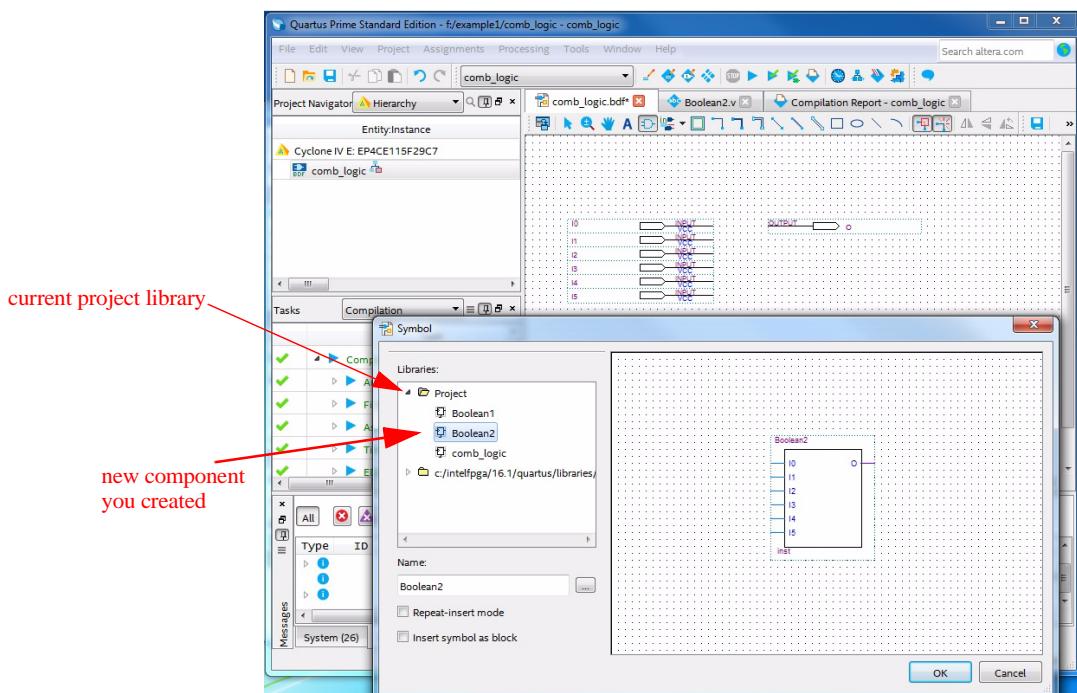
**Figure 2.32**

If there are no errors then Quartus Prime will generate a message at the bottom of the Message window as shown in Figure 2.33 that indicates that the symbol file was successfully created.



**Figure 2.33**

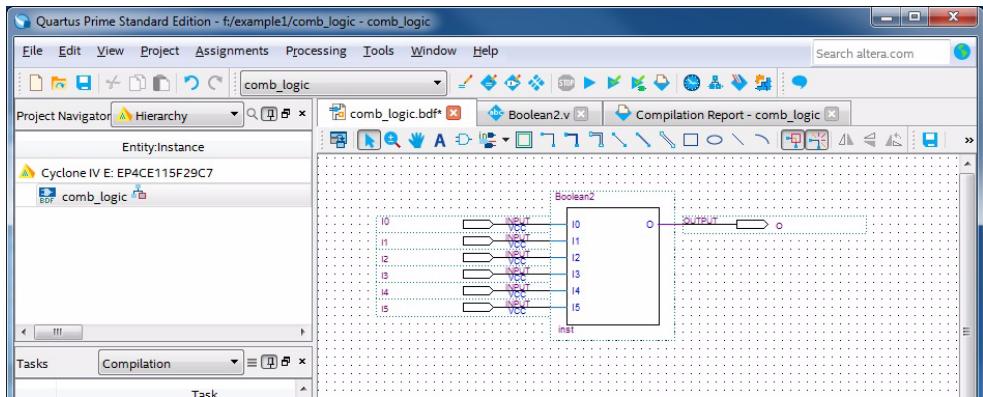
- 7 The next step is to actually use the component that you have created in the design. This is sometimes called instantiating the component. In general to do this requires that you go to the place within your block diagram/schematic hierarchy that you need the component and then simply select the component from your project library that is associated with your design. The component will be interconnected to others in the normal manner.
- In this example we would like to use this component within our top level schematic. To do this click on the **comb\_logic.bdf** tab. Then double click in the white space in the design window or left-click on the icon in the normal manner to reveal the new component you have created (you can also select the *Insert Symbol* option from the *Edit Menu*). You should now see a *Project* folder above the other folders in the library area as shown in Figure 2.34.



**Figure 2.34**

Expand this folder by clicking the symbol. You should see the new component **Boolean1**. Click on this and click on the *OK* button. You can now place this in your top level schematic as you would any other component. Notice the pin names on the symbol are the same as the wire names of your inputs and output in your Verilog HDL module.

To create an equivalent two level hierarchical design that has the same functionality as the schematic or Verilog VHDL only implementation you only need to connect up the inputs and the output to I/O connector components (input and output) and name them in the same way as you did previously. Figure 2.35 illustrates the resulting implementation.



**Figure 2.35**

After you have saved this file you can transcend down the hierarchy by simply double clicking with your left mouse button on the individual components. In this case, double clicking on the **Boolean2** component will cause you to descend down to its Verilog HDL representation that you entered previously making active its tab in the Verilog HDL editor. This is true of any design component regardless of whether it is user entered or part of a preexisting library (unless it has been projected by encryption).