Introduction to Graphical Gate Entry and Simulation in Quartus II

(A.W. Bojanczyk and D. Long)

Quartus II from Altera will be used in the ENGRD2300 digital lab in all design projects. Please note that this document may not have been updated to reflect recent changes to the Quartus II software; some of the details and screen shots may be slightly different.

1 Downloading, licensing and installing student version of Quartus II

Quartus II is available in the CIT lab in 318 Phillips Hall for your use outside of lab. You may use it there or download your own copy for use on your own PC. You may skip this section if you want to use Quartus II in the CIT lab.

The free version of the Quartus II software, the Quartus II Web Edition, can be downloaded to your own PC running Windows Vista, XP or 2000. (Sorry, there is no MAC version.) You can use this version of the Quartus II design software to work on your prelabs outside of lab and then bring your completed designs to lab for your lab sessions. Please note that this version may differ slightly from the version on the lab computers, but this is generally not a problem.

The Quartus II Web Edition can be downloaded from the Altera web site from the following URL:

http://www.altera.com/products/software/quartus-ii/web-edition/qts-we-index.html

Once you are there, do the following to install the software:

1) Click on 'Download Software Web Edition – Free'.
2) Click on the 'Download' button for **Quartus® II Web Edition Software for Windows.**
3) On the Altera.com Account Sign-In page, click on 'Get One-Time Access', fill in the information requested information, and click on 'Establish One-Time Access'.
4) Follow the instructions in the Download Manager to download your file. This is a rather large file, so it can take a while to download.
5) After Quartus II has downloaded, run the installer and install the program. You should now be able to use the software

2 Introduction

We will illustrate the basic graphical entry and simulation in Quartus by designing a 4-bit adder. The design will be hierarchical starting from a simple XOR gate going to a Half Adder, Full Adder, and finally to 4-bit Adder

3 Design of an XOR gate using the Schematic Editor

3.1 Making a Project

1. On your PC, create a 'Lab1' directory and in this directory, create a subdirectory called 'myxor'.
2. Launch the Altera Quartus II software.
3. Go to File | New Project Wizard… and click 'Next'. This will start the new project wizard.
4. The directory, the project name and the top-level project name should be entered on the next screen. Enter 'myxor' as the name of the project. The name of the top-level design entity should be automatically filled in with the name 'myxor'. If not type 'myxor' in this box, too. Click 'Next'.
5. Type 'myxor.bdf' in the File name box. Click 'Add' and then click 'Finish'. You now have a project which can eventually be compiled, simulated and loaded into a device.

3.2 New Design Creation

1. Under the pull-down menu File select New and next Block Diagram/Schematic File. Click OK.

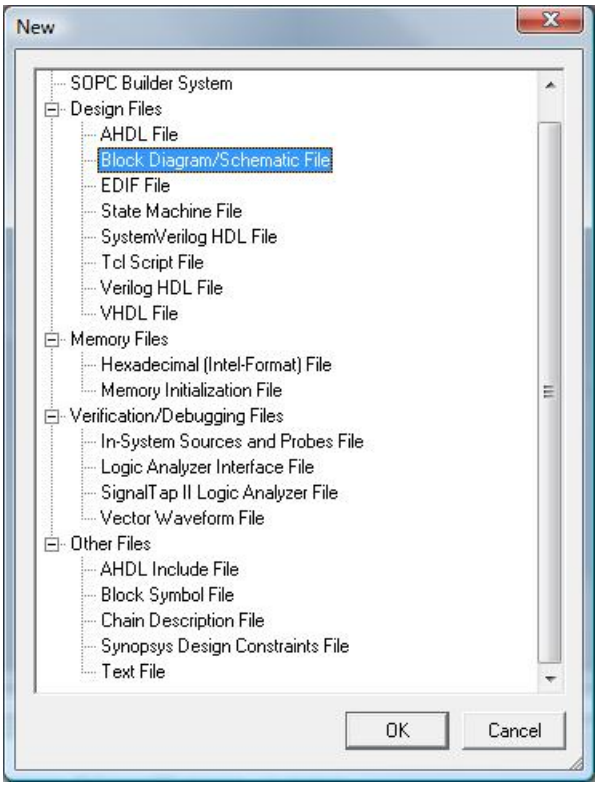

Figure 1

This will open a palette or circuit board that is designated “Block1.bdf”.

2. Select the pull-down menu File and save the graphic design file as “myxor.bdf” in your ‘myxor’ project directory created in Step 1. The .bdf extension is added automatically and stands for ‘block design file’.

3.3 Adding Text

1. On the palette to the left of your Block Diagram/Schematic File window, select the “A” below the arrow.
2. With the left mouse key select a point near the bottom left in the window. Type your name and Netid, hit the ‘Enter’ key, type ‘Monday/Tuesday/Wednesday Section’ (select your section), hit the ‘Enter’ key and type an appropriate title, such as ‘Lab1, 8-bit Adder’.
3. Hitting the ‘Esc’ (escape) key or clicking on the Arrow on the palette will end the text mode.

3.4 Component Selection Process and Moving Components

1. There are two ways of inserting library symbols to Block Diagram windows:
   (a) With your mouse pointing inside the window, double-click with the left mouse key, or
   (b) Click the right mouse key and select **Insert/Symbol.** The ‘Symbol’ dialog box will appear. This window lists the available Altera libraries (and user created library, if any).
2. Select the + icon to expand the “c:/altera/80sp1/quartus/libraries” folder. (This folder may have a different name if you are using a different version of Quartus II.)
3. Select the + icon to expand the “primitives” folder and then expand the “logic” folder.
4. In the logic folder,
   (a) Select the “and2” component and double click on it, or
   (b) Select the “and2” component, single click on it and then click “OK”, or
   (c) Type “and2” in the ‘Name’ box and then click “OK”.
5. Click the pointer at the desired location in the Block Diagram/Schematic Editor window to insert the ‘and2’ symbol into the design file.
6. If you need to add multiple copies of the desired gate, you could
   (a) Select the ‘Repeat-insert mode’ box, and follow instructions in point 5, or
   (b) In the Block Diagram/Schematic Editor, highlight the desired symbol, copy (ctrl-c) and paste it (ctrl-v).
7. Repeat Steps 1-6 to enter (the total of) two ‘and2’, two ‘not’ gates, and a single ‘or2’ gate.
8. Repeat Steps 1-6 to enter (the total of) two ‘input’ and a single ‘output’ terminal. These can be found in the “c:/altera/80sp1/quartus/libraries/primitives/pin” library.
9. Rearrange gates so the gate placement is the one you want to have. You can move a component by selecting it with your mouse, holding down the left button and moving it to another location on the palette. For example, the window could look like follows:

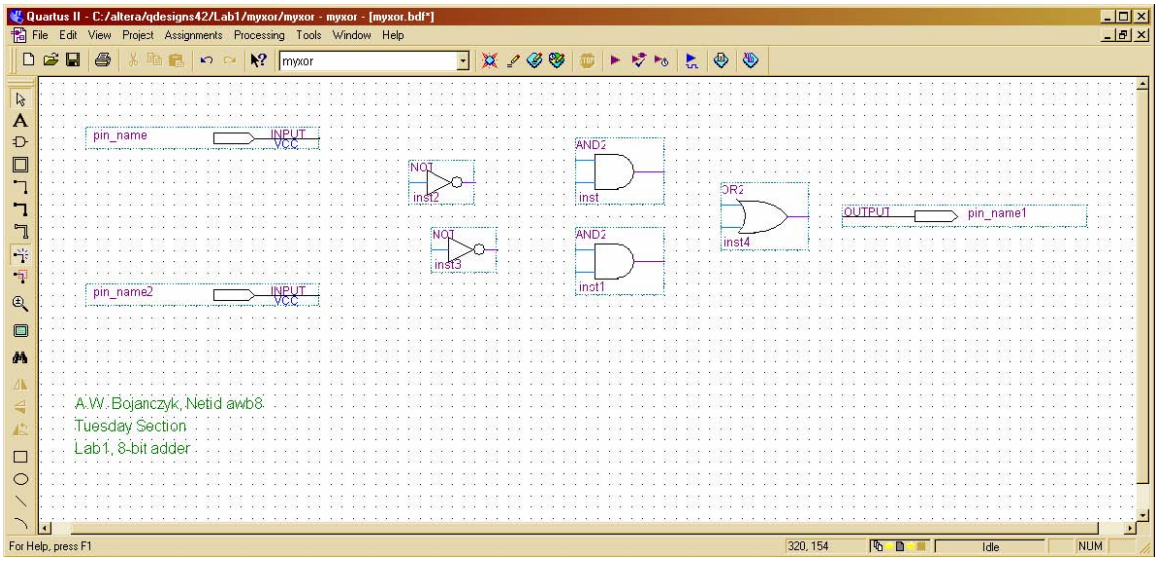

Figure 2

10. You can use the 'Zoom Tool' to get a closer look at any part of your design. To use the 'Zoom Tool', select the magnifying glass on the palette to the left of your block Block Diagram/Schematic Editor window. Select a point in the window. The image will get larger each time you push the left mouse button. The image will get smaller each time you push the right mouse button.
11. To exit the 'Zoom Tool', either hit the "Esc" (escape) key on the keyboard, or select the arrow symbol to return to selection mode.

3.5 Naming Terminals

1. Select the 'Text Tool' (depicted as the capital letter 'A' to the left of the .bdf window). Left click on the top input terminal. The Pin Properties window will open. Type Type 'a' for the Pin Name: and click OK.
2. Repeat Step 1 for the other input terminal (typing 'b'), and next for the output terminal (typing 'x').

3.6 Adding/Deleting Wires

1. You are now ready "wire up" your circuit. First save your design. It is a good idea to save your design often, just in case something bad happens. Next select the arrow tool on the left of the .bdf window.
2. Place your pointer on the output of the top 'not' symbol and hold the left mouse button down. You should see a cross-hairs or "+" appear at the output.
3. Drag your pointer to the bottom input of the top 'and2' gate. Every time you release the mouse key, the line (wire) ends. If your wire did not reach the input of the 'and2' gate, you can add to the wire by putting your mouse over an end of the wire and again selecting it with your left mouse button and dragging your mouse to another position.
4. To delete a wire or a portion of a wire, simply click on it (it should change color to indicate selection) and press the delete key.

5. If wires are connected to the component as you are moving it, the wires will stretch and stay connected to the component.
6. Add the rest of the wires needed to connect the logic diagram as shown below.

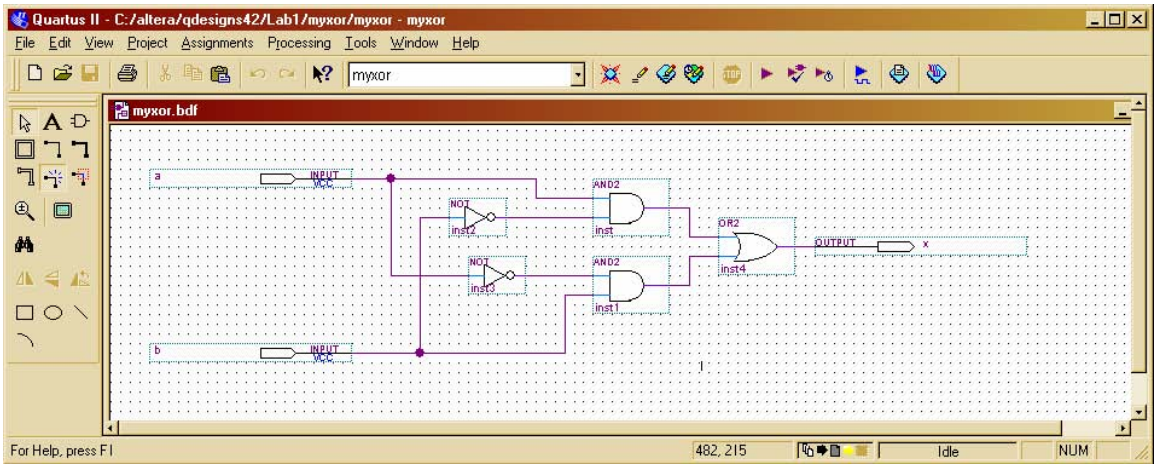

Figure 3

3.7 Naming Wires

The Quartus II software allows you to give wires explicit names. This is often a good design practice. However, you should be aware that the Quartus II software will assume that terminals and wires with the same names are *connected*. This can lead to inadvertent errors in your design if you are not careful!

The process for naming wires is as follows:

1. Add small input and output lines where the inputs and the output are to be placed.
2. Select the wire near the where you would like to connect signal ‘a’. The wire should change colors. Type ‘a’. An ‘a’ should appear near the selected point.
3. Do the same to place ‘b’ and ‘x’ at the appropriate points.
4. The ‘a’ label will connect the input labeled ‘a’ to this wire. Similar connections are made by the labels on the other inputs and output. The window might look something like the figure below.

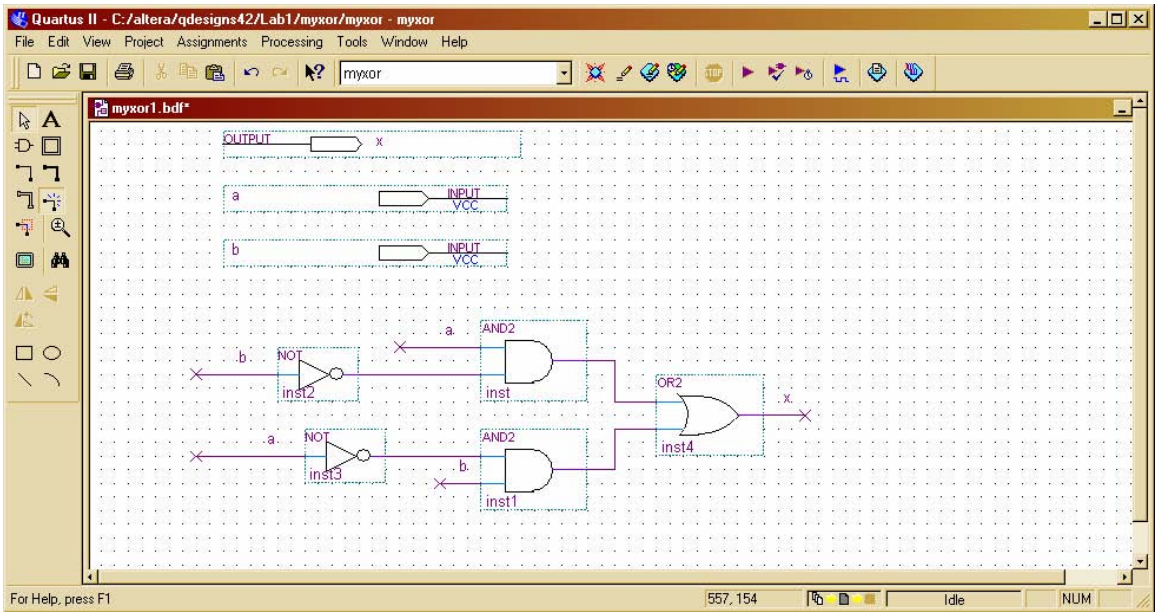

Figure 4

WARNING: It is very easy in Quartus II to make inadvertent connections by naming two *different* things with the *same* name. Be especially careful in naming wires and avoid names that end with digits (e.g., a0, b1, c17,…) as these names *may* also be used by Quartus II as individual components of busses (this is primarily a problem with older versions of the software).

Also, while it possible to create an entire schematic with no wires (by naming each connection), this does not necessarily create a *good* schematic. A good schematic helps the viewer to understand the circuit visually; making *explicit* connections (using wires) helps the viewer understand things much more quickly. For example, the first XOR schematic we showed you is probably easier to understand than the second.

Now you are ready to compile and simulate your circuit.

3.8 Compiling

Later in the semester we will be using Altera DE2 development boards which include an Altera Cyclone II device. Thus it is a good idea to get in the habit of compiling your designs with the target device that we will be using in lab.

1. Assign the design to a particular device family by selecting Assignments | Device. The device we will be using starting from Lab 4 is from the Cyclone II family.
2. Select the 'Specific device selected…' and then choose EP2C35F672C6 from the list of Available devices.
3. In the Processing menu select 'Compiler Tool'. You should see the following:

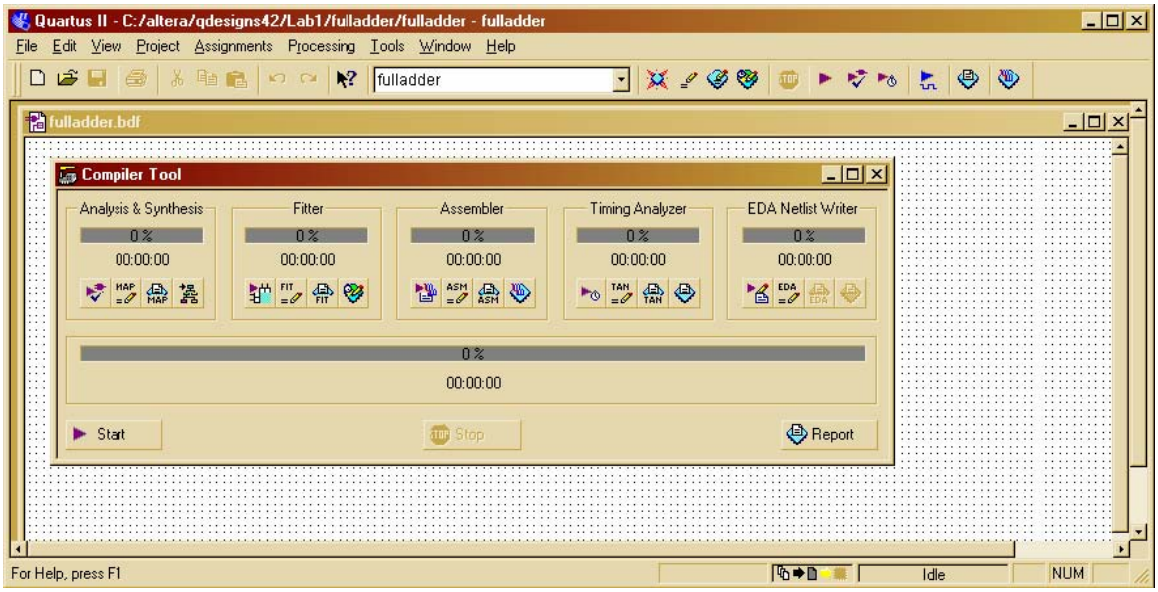

Figure 5

4. Click the 'Start' button. After some time you should see a message from the compiler that says "Full compilation was successful."

   NOTE: The compilation process consists of several sub-processes that we will breakdown and describe at a future date. For now, we just want you to understand the main function of the compiler is to translate the circuit into a set of equations that are fitted or placed into a programmable logic device.

5. Close the 'Compiler Tool' window.

3.9 Creating a Symbol

In the next part of this tutorial your XOR gate will be used to create a Half Adder. Instead of copying and pasting the entire XOR circuit it is convenient to create a symbol for the XOR gate.

1. Under the File menu select 'Create/Update' and click on 'Create Symbol Files for Current File'. Quartus II should respond with 'Created Block Symbol File myxor'. You will be able to use this symbol as an XOR gate in the later part of this tutorial.

3.10 Creating a Waveform for use with the Simulator

Once the circuit has been fitted into a particular programmable logic device, we now can run the simulator to check the circuit functionality. We will stimulate the circuit with a vector of inputs to check whether the output agrees with values in the voltage (truth) table for the XOR gate.

1. In the File menu select New. In the 'Verification/Debugging' group (click the + icon to expand if necessary) select 'Vector Waveform File' and click OK. You should see something like the following:

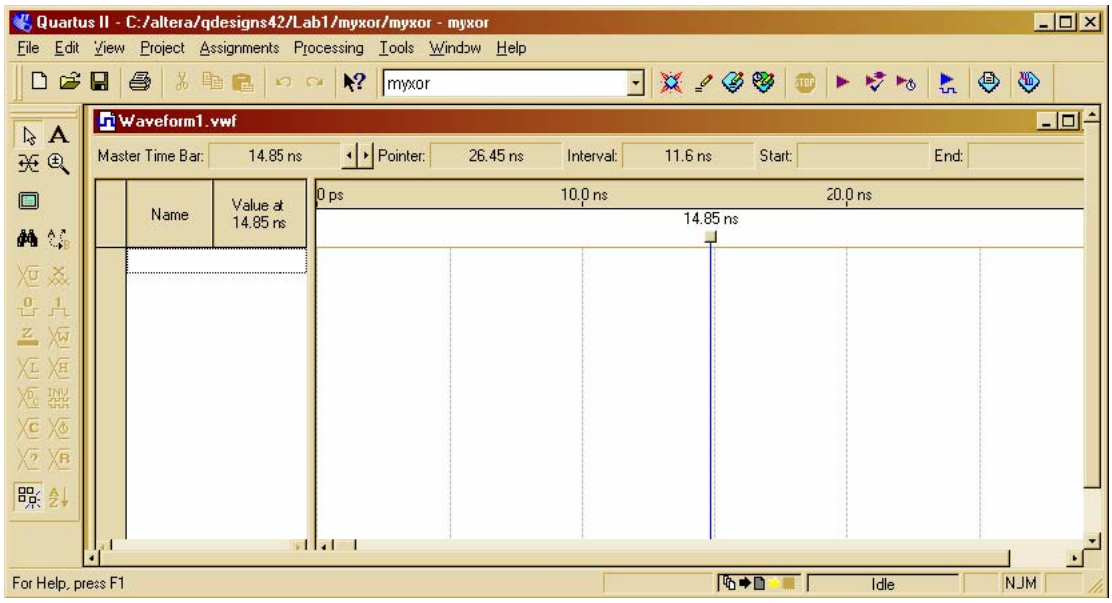

Figure 6

Save this file in your project directory (this should be saved automatically with the name "myxor.vwf". If not type "myxor.vwf" in the Name box).

2. Under the Edit menu select 'Insert Node or Bus'. You should see

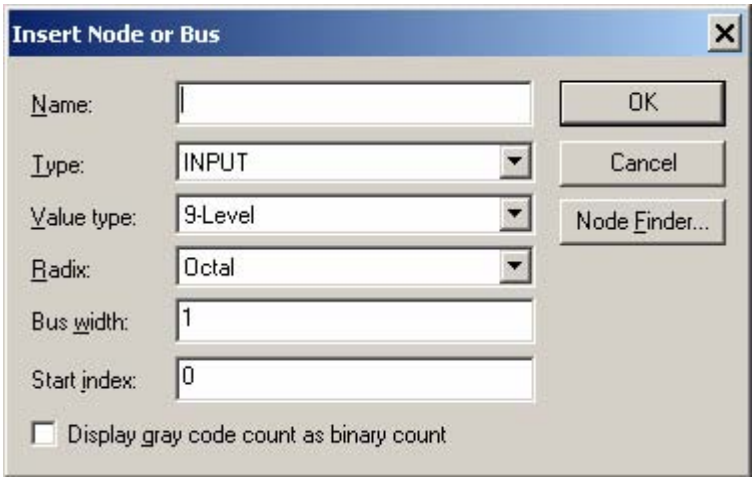

Figure 7

3. Select 'Node Finder'. The next window to pop up is

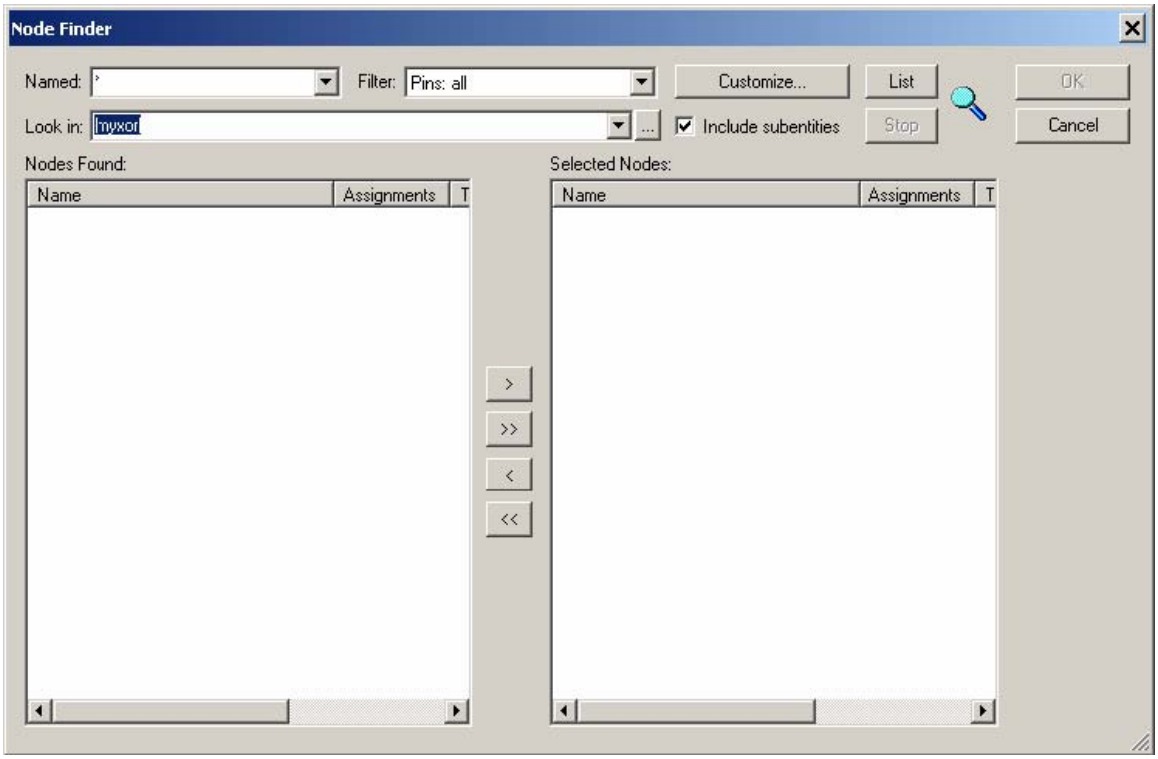

Figure 8

4. Under ‘Filter’ box select ‘Pins:all’ and click the box ‘List’. On the left side of the new window you will see all terminals associated with the current design. To select specific nodes highlight them and click the ‘>’symbol. To select all nodes, which we want to do here, click on the ‘>>’ symbol. You should see the following

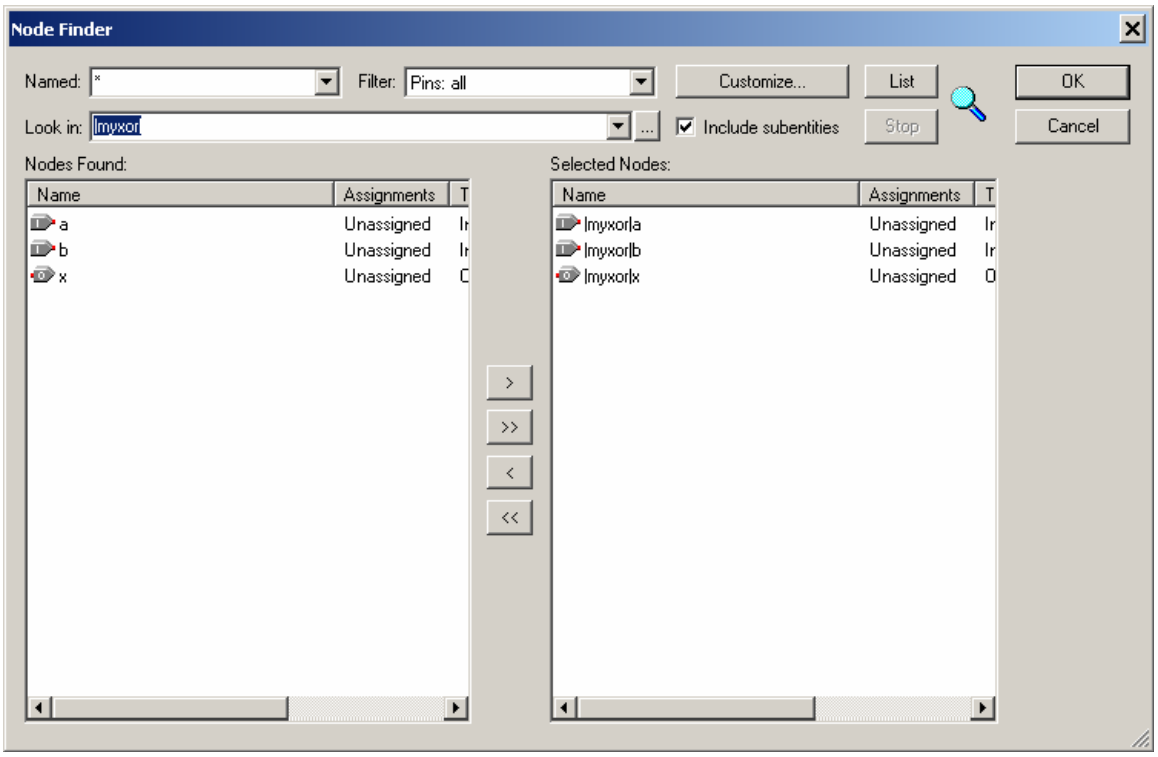

Figure 9

5. After clicking ‘OK’ twice the waveform window will appear.

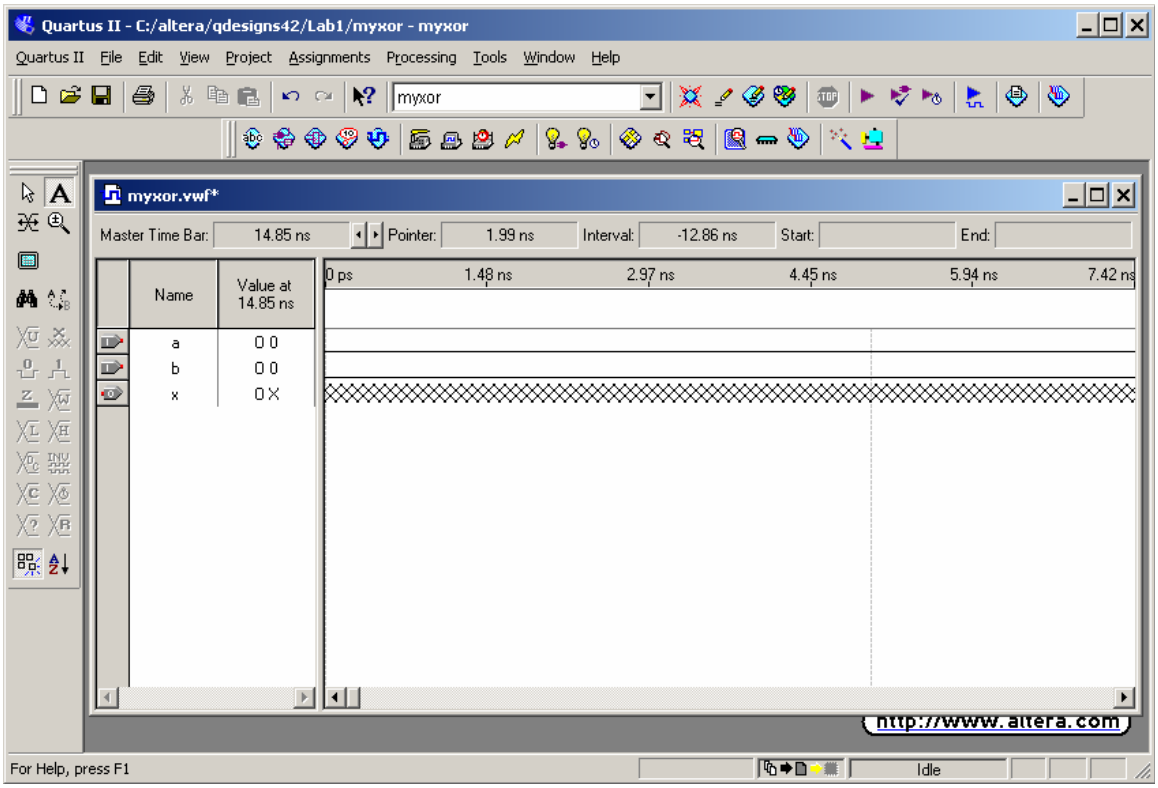

Figure 10

6. Next, we need to get the right time scale. Under the Edit menu select ‘Grid Size’ and set the ‘Time Period’ to 25ns. Again, under the Edit menu select ‘End Time’ and set the ‘Time’ to 100ns. (There are 4 entries in the truth table for the XOR gate. 100/25 = 4 so the simulation will test all possible combination of input values).
7. The waveform window may start with a different time scale. Use the ‘Zoom Tool’ to see the entire extent of the simulation time (100ns).
8. Now we can associate values with the input terminals ‘a’ and ‘b’. Select the ‘Selection Tool’ (the arrow) from the left toolbar. Using your mouse, drag it across the first 25 ns segment corresponding to the ‘a’ terminal. Now press the left toolbar button that has a “0” in it. Now try pressing the left toolbar button that has a “1” in it. You will notice that the waveform in the segment goes down when a ‘0’ is clicked, and goes up when a ‘1’ is pressed. This is how you set input segments to particular values. Set it so you have the following values assigned to inputs

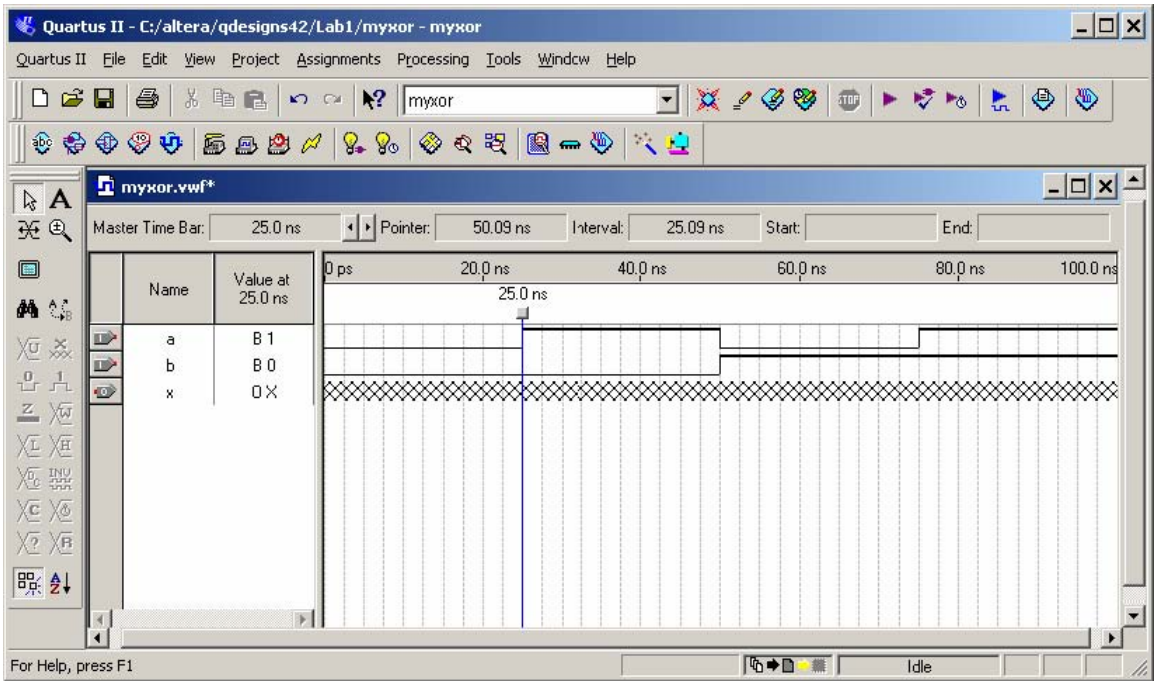

Figure 11

The cross hatching, ‘XXXXXX’, in the output terminal ‘x’ signifies that the output is undefined (we have not started simulation yet).

9. Save the file to make the changes permanent.
10. Under the Processing menu select ‘Simulator Tool’. A new window will pop up.

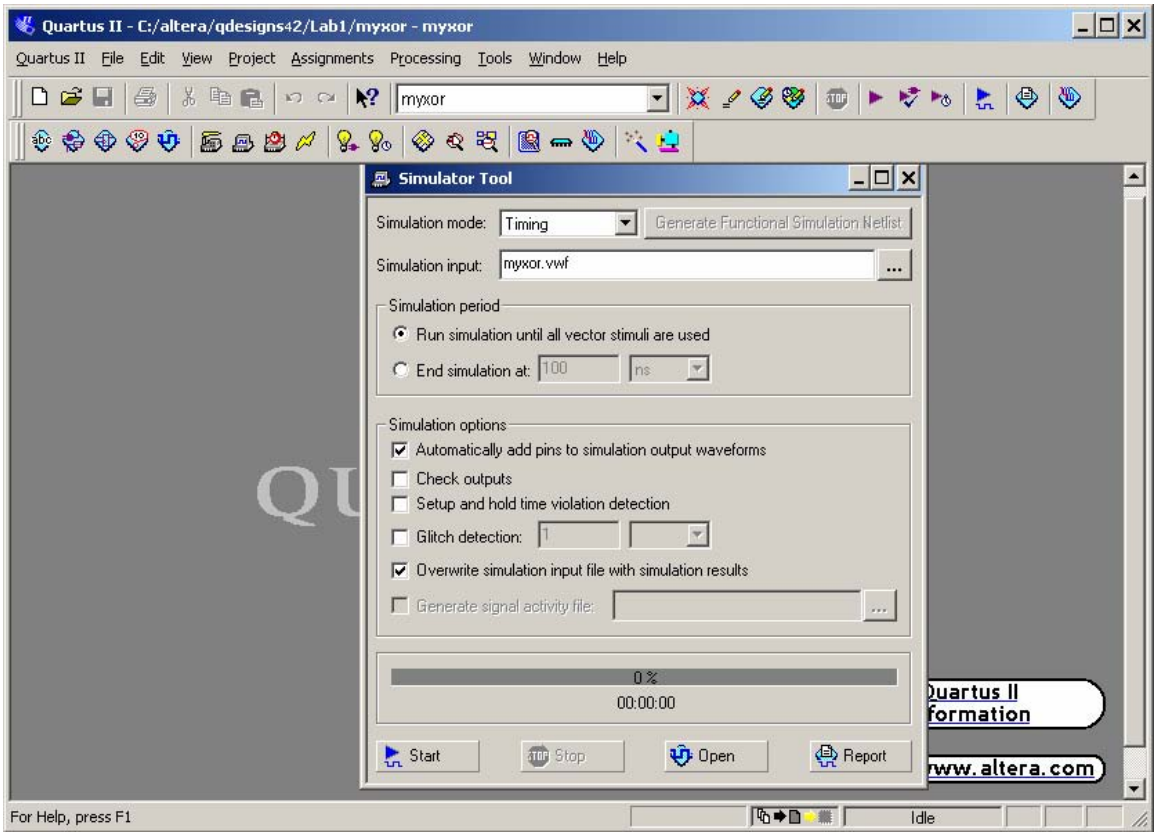

Figure 12

11. Press the ‘Start’ button. After a while the simulator should respond with the message “Simulator was successful”. Click ‘OK’ and then in the Simulator Tool window press the ‘Report’ button. The results of the simulation should show up in another window.

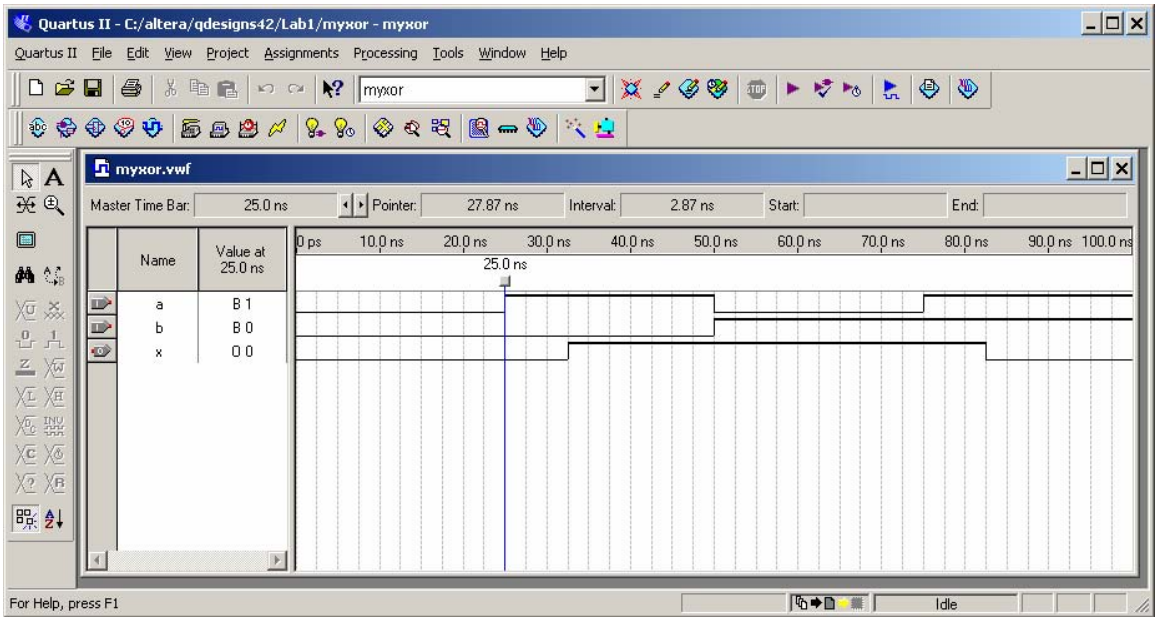

Figure 13

12. The values of the output terminal ‘x’ are defined now and are voltage low (Boolean ‘0’) when the waveform is low, and voltage high (Boolean ‘1’) when the waveform is high. If the output values agree with the output column in the truth table your design is (logically) correct.

Note: You will notice that the output ‘x’ does not change immediately after the input ‘a’ and/or ‘b’ changes. In our Cyclone II device, there is about a 5ns to 7.5ns propagation delay between the time an input changes and time the output changes.

4 Design of a Half Adder using the Schematic Editor

A Half Adder has 2 inputs ‘a’ and ‘b’ and two outputs ‘s’ and ‘c_out’ which are given by the following logic equations

$$s = a \text{ XOR } b$$
$$c\_out = a \text{ AND } b$$

In this design, you will use the myxor gate that you designed earlier. It can be helpful to derive a truth table for the half adder. Fill in the missing values for ‘s’ and ‘c_out’ according to the equations above.

a	b	s	c_out
0	0		
0	1		
1	0		
1	1		

Table 1

4.1 Creating a New Project

1. Under the File menu select New Project Wizard. Click next. The following window will appear.

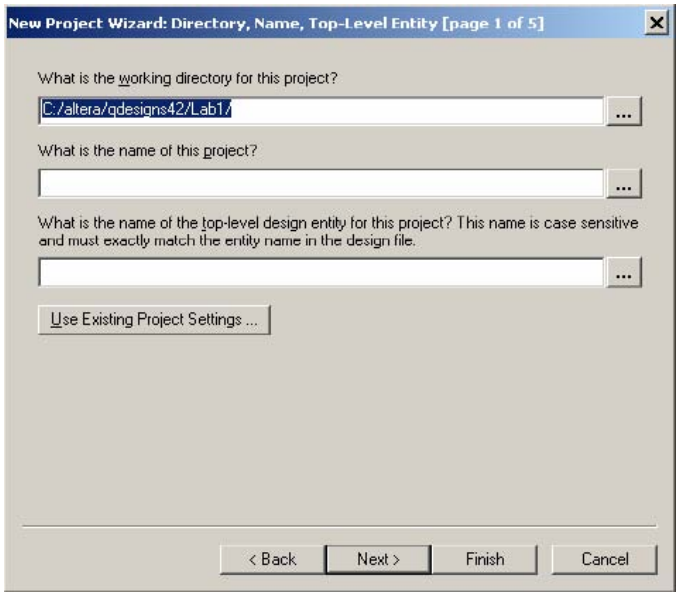

Figure 14

2. In the box 'What is the name of this project' type "halfadder". The text "halfadder" will automatically appear in the next box. Click 'Next' and answer 'No' to the question that will appear next. This will bring up a 'New Project' window.

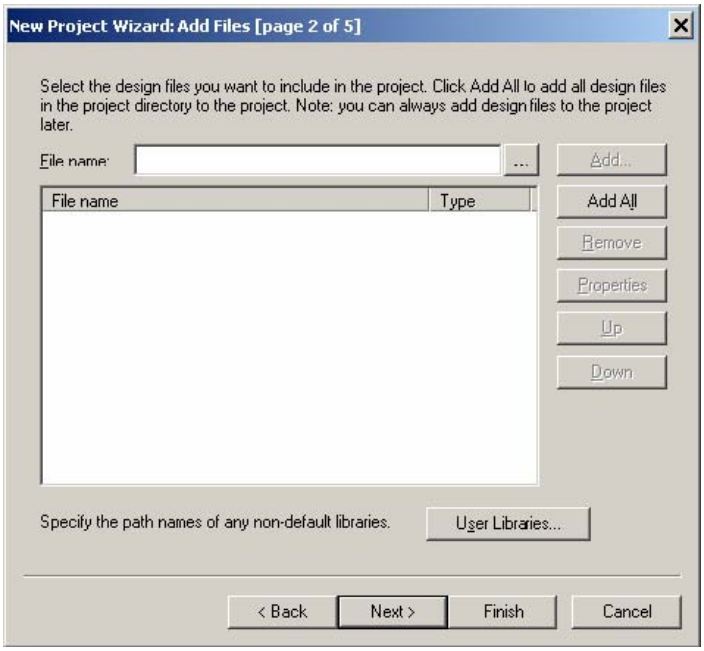

Figure 15

3. We need to add the design file myxor to this new project. In the File name box type the complete path to the directory that contains the myxor design file (this often can also be achieved by clicking the box '…' on the right of the File name box). In the list of files that will appear highlight 'myxor' and then click Add.

4.2 Creating a Half Adder Schematics

1. Create a new Block Design file named “halfadder.bdf” (described in steps 3.1.1 to 3.1.4.)
2. Create the following schematic (follow the directions given in sections 3.3 to 3.5.).

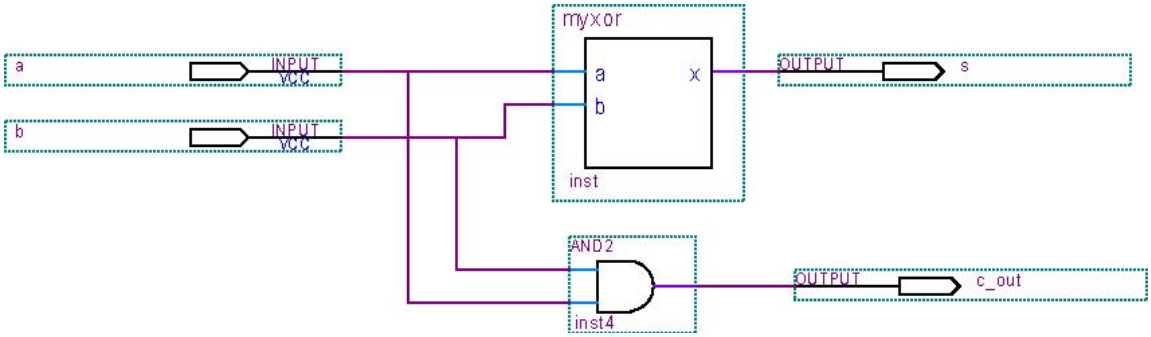

Figure 16

3. Create a symbol for this circuit (follow the directions given in section 3.9).

4.3 Compilation and Simulation

1. Compile your circuit (as described in section 3.8).
2. Run a simulation, setting up the inputs “a” and “b” as in 3.10, and set the outputs to undefined (symbol ‘XXX’ on the left menu bar).
3. Make sure that the results of the simulation agree with the truth table of the half adder.

5 Design of a Full Adder using the Schematic Editor

A Full Adder (1-bit adder) can be built from 2 Half Adders and an XOR gate.

1. Create a new project called ‘fulladder’ (described in section 4.1.1 to 4.1.4). Make sure that the new project includes earlier designs ‘myxor’ and ‘halfadder’.
2. Create the following schematic (follow the directions given in sections 3.3 to 3.5).

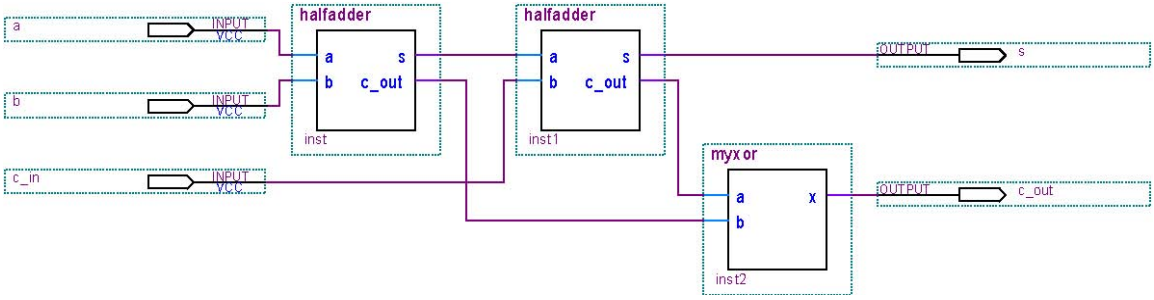

Figure 17

Fill in the truth table below for the full adder defined by the schematic above.

a	b	c_in	s	c_out
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table 2

Create a symbol for this circuit; call it ‘fulladder’.

4. Compile your circuit (as in section 3.8). Simulate for all binary values of ‘a’, ‘b’, and ‘c_in’. Make sure that the results of the simulation agree with the truth table of the full adder.

6 Design of a 4-bit Adder using the Schematic Editor

A 4-bit adder can be built from 4 Full Adders.

6.1 Component Selection and Naming

1. Create a new project by repeating steps 4.1.1 to 4.1.4. Call the new project ‘fourbitadder’. Make sure that the new project includes earlier designs ‘myxor’, ‘halfadder’ and ‘fulladder’.
2. Repeat steps 3.C-3.D to arrive at something similar to the following,

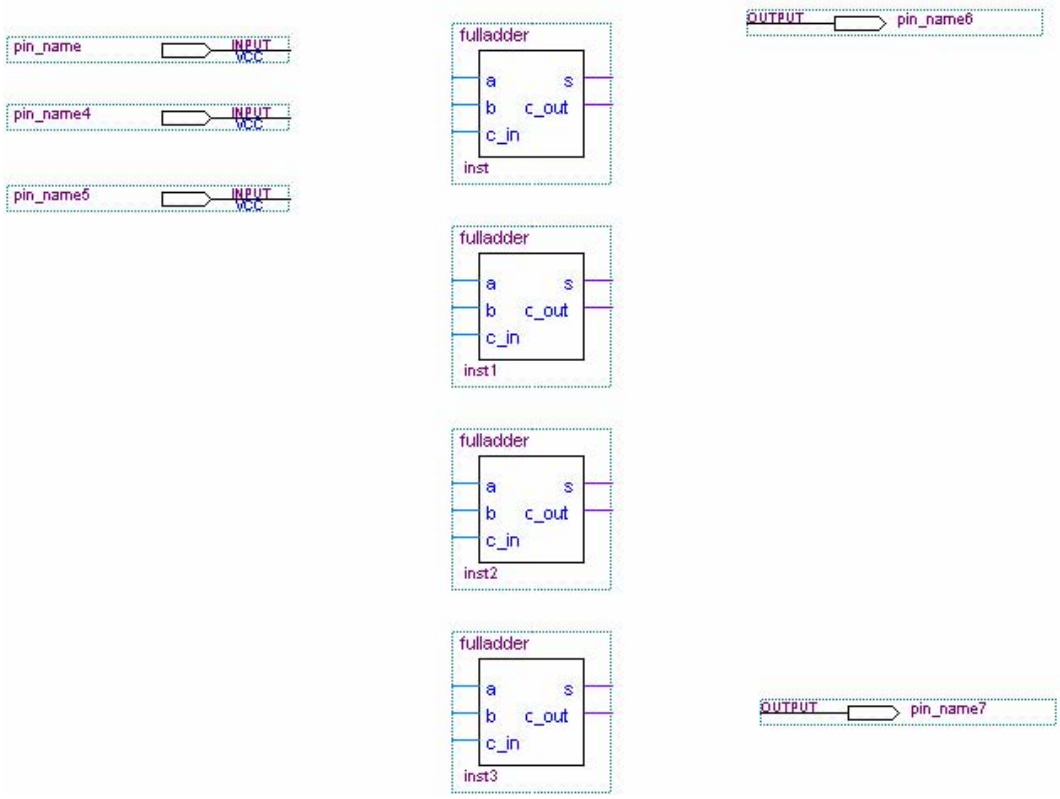

Figure 18

Note that we now have only 3 input terminals instead of expected 13. This is because in Quartus II it is possible to group inputs and represent them by a single input terminal.

3. Select the Text Tool and click on the top input terminal. As its name type a[3..0].
4. Go to the second input terminal and type b[3..0].
5. Name the third input as c_in (it is a single bit input terminal).
6. Go to the top output terminal and name it s[3..0].
7. Name the bottom output terminal as c_out.

6.2 Buses and Naming

1. The grouped terminals need to be connected to buses (groups of single wires). Select the Bus Tool (it is indicated by a solid thick orthogonal line to the left of the Block Graphic window). Place the crosshair at the end of the input terminal a[3..0] and draw a vertical line to the bottom of the window. Repeat with the terminal b[3..0] and then with the output terminal s[3..0].
2. Using the Selection Arrow draw (a single bit) line connecting the input terminal ‘a’ of the top fulladder with the bus a[3..0]. Click on the wire and type a[0].
3. Repeat Step 3 for the remaining terminals ‘a’ and naming the wires as a[1], a[2] and a[3].
4. Repeat Steps 2 and 3 for the terminals ‘b’ and ‘s’.

5. Using the Selection Arrow draw a (single bit) line connecting the input terminal c_in with the c_in input of the top fulladder.
6. Using the Selection Arrow draw a (single bit) line connecting the output terminal c_out with the c_out terminal of the bottom fulladder.
7. Draw a single bit line joining the c_out terminal of the top fulladder with the c_in input terminal of the fulladder immediately below the top fulladder.
8. Repeat Step 7 to propagate c_out to c_in. After completing Steps 1-8 your schematic should look similar to the one below.

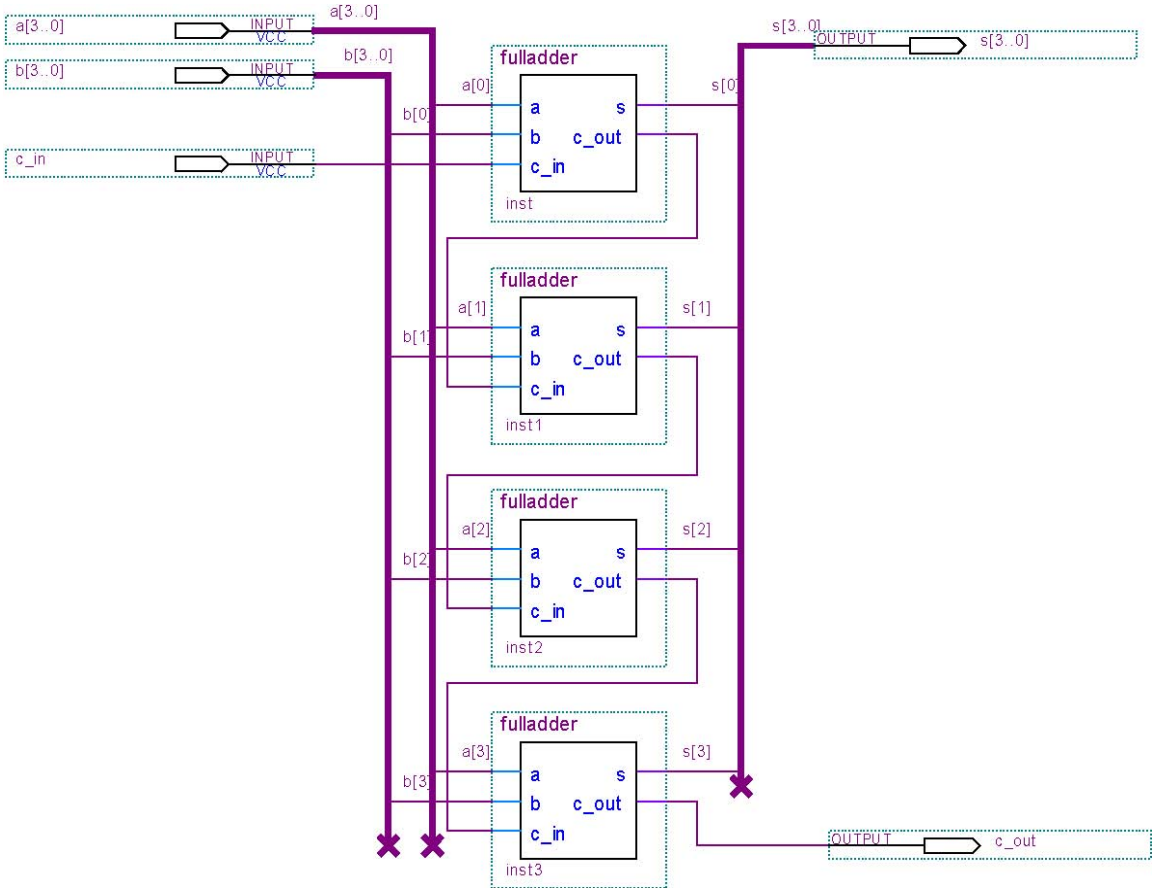

Figure 19

9. Save the file fourbitadder.
10. Compile your circuit (see section 3.8).

6.3 Simulation

The simulation part for the fourbitadder is analogous to the simulations you did earlier except that Quartus II allows you to assign numerical values (unsigned integers) to terminals a[3..0] and b[3..0]. Also the simulation results on the output terminal s[3..0] can be represented as numerical values.

1. Create a new waveform file and open the node finder (see sections 3.10.1 to 3.10.3).
2. Next, select only node a, b, s, c_in and c_out (see section 3.10.4). Note that nodes a[3]-to-a[0] (as well as b[3]-to-b[0] and s[3]-to-s[0]) could be selected individually. However, at this point we are not interested in the individual values of bits.

3. Set the End Time to 2us and the Grid Size to 100ns.
4. In the waveform window click on the node ‘a’ (an entire line will be highlighted). Now click on the ‘Count value’ icon to the left of the window (it has a C’ in it). This will bring up the Count Value window.

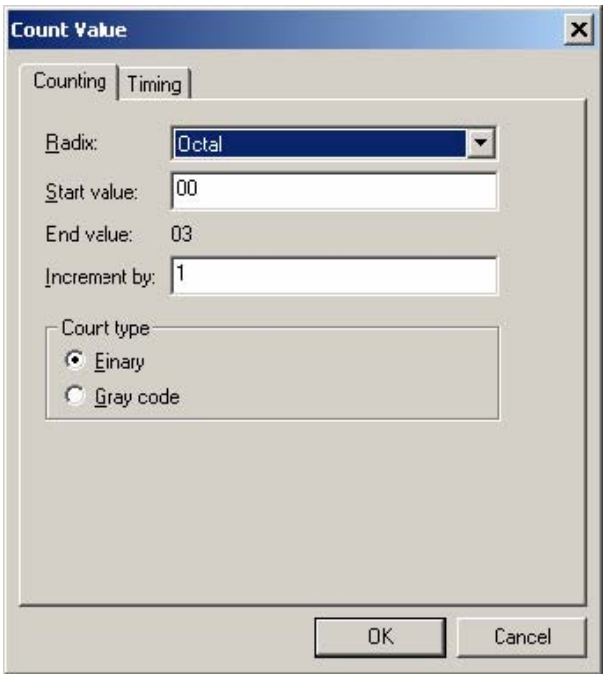

Figure 20

5. In the box Radix select Hexadecimal and click OK.
6. Repeat Steps 4-5 for the node ‘b’.
7. Right click on ‘s’ and select Properties. Set the Radix to Hexadecimal and then click OK. Then set ‘s’ to undefined (symbol ‘XXX’ on the left menu bar).
8. In the waveform window click on the node c_in and then on the ‘Count value’ icon ‘C’. This brings up the Count Value window again. In the Radix box select ‘Binary’ and click OK.
9. Save the waveform file. It should look like the one below.

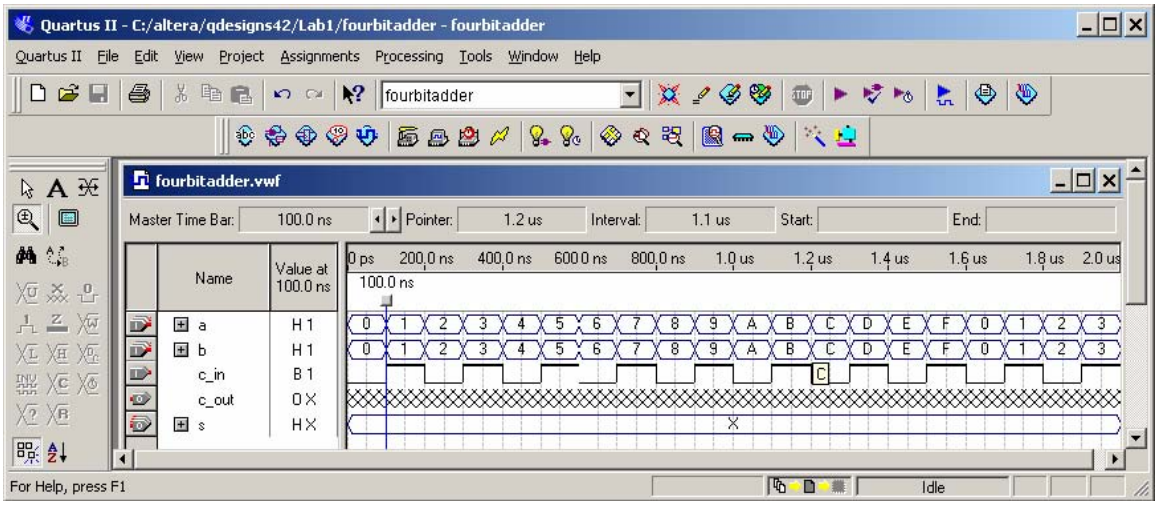

Figure 21

10. Start your simulation (as described in sections 3.10.10 to 3.10.12).
11. You can test your circuit on different inputs by changing the input waveforms (as you did in section 3.10.8). Highlight the portion of the waveform that you want to assign a new value to. Next click on the Count Value icon. In the Count Value window that shows up, in the box Start Value type a desired value, say '7'. Change the waveform in other time segments. Repeat simulation and check whether the output waveform represents correct results of the summation.

7 Error reporting

Please report any errors in this tutorial to dll26@cornell.edu.