Lab 4
Introduction to Quartus II

Name______________________________ Class __________ Date ________________________________

In this tutorial we will learn how to use Quartus II to create a simple logic circuit and simulate the circuit operation with a set of input and output waveforms. The logic circuit is described by the Boolean expression:

$$Y = AB + BC + AC$$

Figure 1shows the design for the tutorial example. The output of this circuit is HIGH if any two inputs are HIGH or if all three inputs are HIGH.

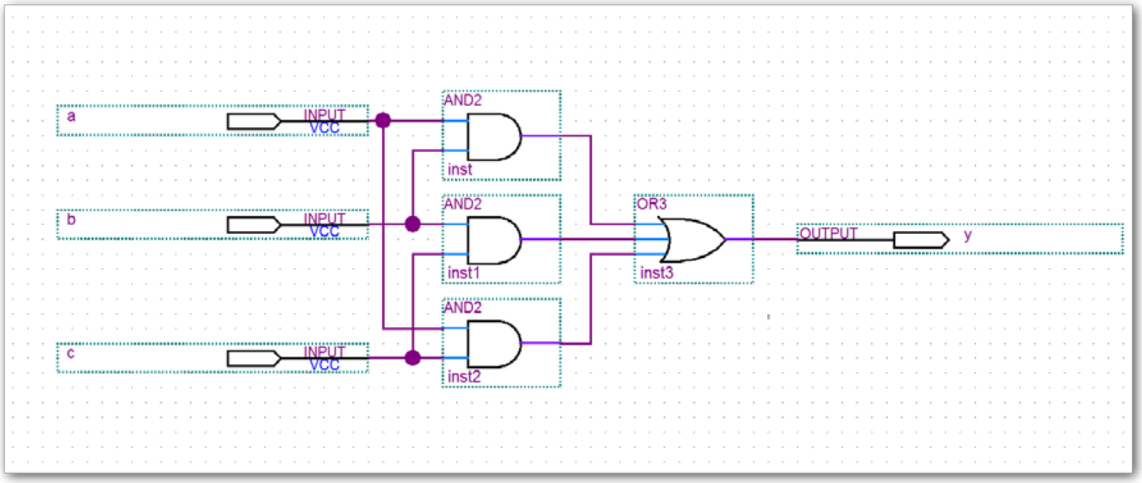


Figure 1 Example Design (Two or Three Inputs HIGH Make Output HIGH)

Make a Project Folder

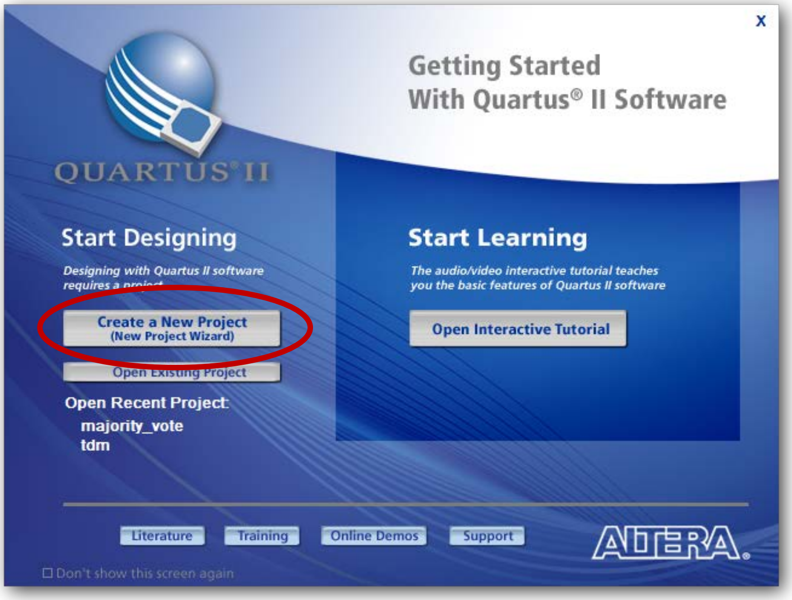
In all the Quartus II Labs, you should use a USB memory stick to save your work. Every Quartus II project must be in its own folder. Examples in the labs assume that save your files in a folder on your network drive, **G:\**.

Make a folder for the tutorial project called
G:\Digital_Logic\qdesigns\01_Quartus_Tutorial.

Lab 4
Introduction to Quartus II

Make a New Project

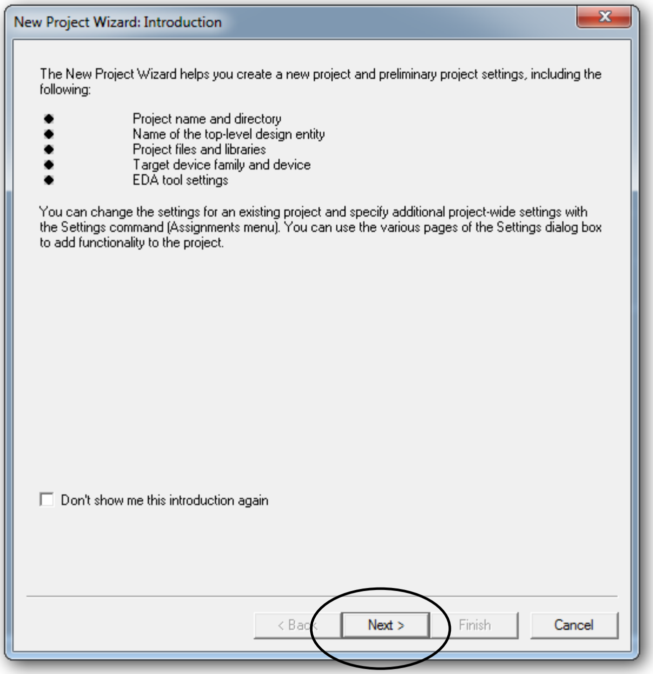
Start the Quartus II software. The dialog box in Figure 2 opens. Select **Create a New Project**. (Another way to start a new project is to select **New Project Wizard** from the Quartus II **File** menu.)



Designing with Quartus II software requires a project

The audio/video interactive tutorial teaches you the basic features of Quartus II software

Figure 2 Quartus II Opening Dialog



New Project Wizard: Introduction

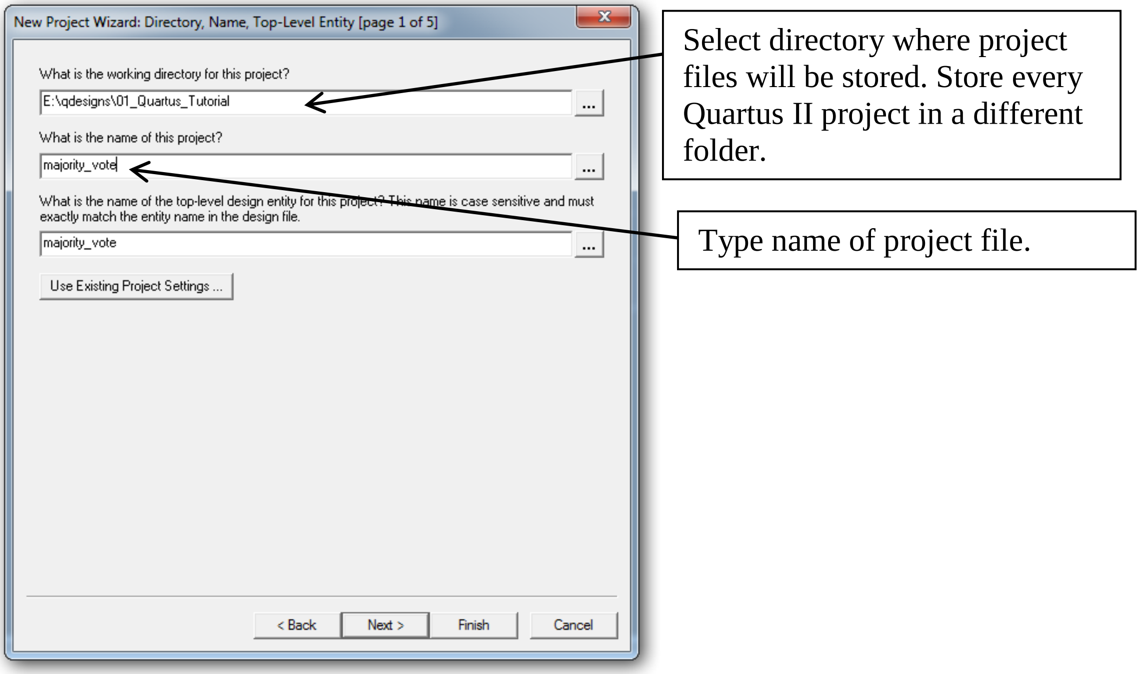
The New Project Wizard helps you create a new project and preliminary project settings, including the following:

- Project name and directory
- Name of the top-level design entity
- Project files and libraries
- Target device family and device
- EDA tool settings

You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project.

□ Don't show me this introduction again

Figure 3 New Project Introduction (Click Next)



New Project Wizard: Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?
E:\qdesigns\01_Quartus_Tutorial

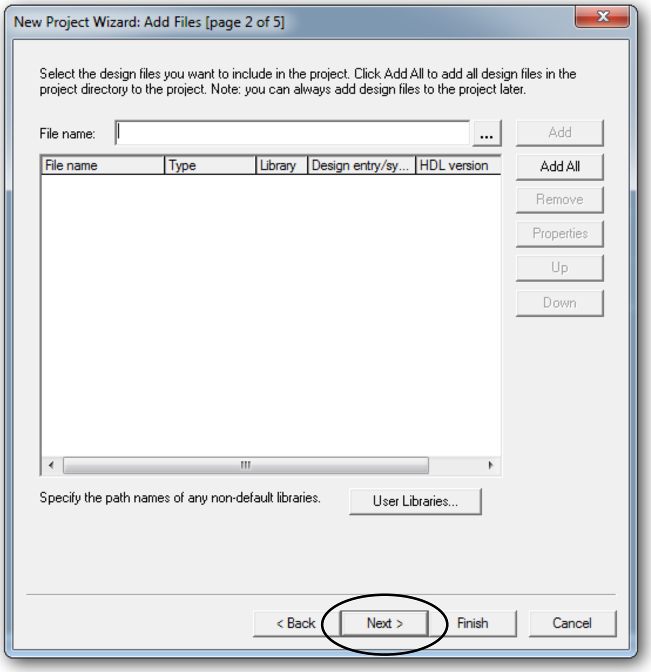
What is the name of this project?
majority_vote

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.
majority_vote

Select directory where project files will be stored. Store every Quartus II project in a different folder.

Type name of project file.

Figure 4 Working Directory and Project Name



New Project Wizard: Add Files [page 2 of 5]

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project. Note: you can always add design files to the project later.

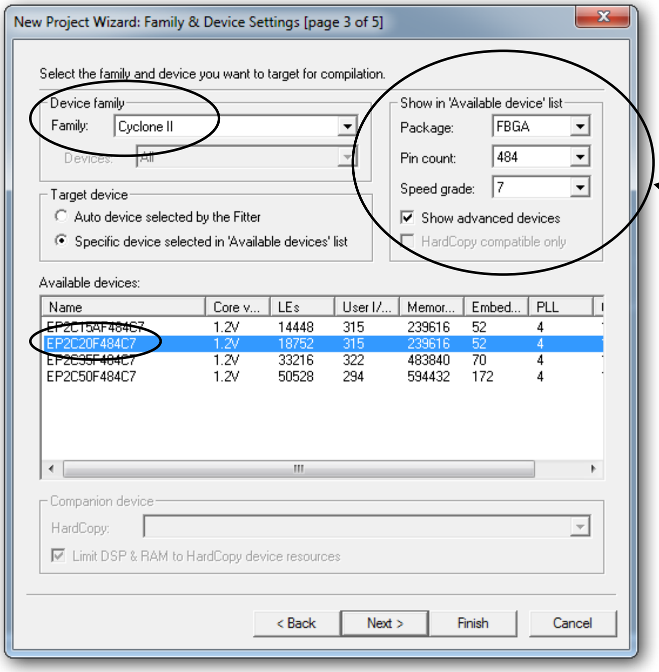
File name:

File name	Type	Library	Design entry/sy...	HDL version

Specify the path names of any non-default libraries.

Figure 5 Add Files (nothing to add yet; click Next)

Lab 4
Introduction to Quartus II



New Project Wizard: Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family

Family: Cyclone II

Devices: All

Target device

○ Auto device selected by the Fitter

● Specific device selected in 'Available devices' list

Show in 'Available device' list

Package: FBGA

Pin count: 484

Speed grade: 7

☑ Show advanced devices

□ HardCopy compatible only

Available devices:

Name	Core v...	LEs	User I/...	Memor...	Embed...	PLL
EP2C15AF484C7	1.2V	14448	315	239616	52	4
EP2C20F484C7	1.2V	18752	315	239616	52	4
EP2C35F484C7	1.2V	33216	322	483840	70	4
EP2C50F484C7	1.2V	50528	294	594432	172	4

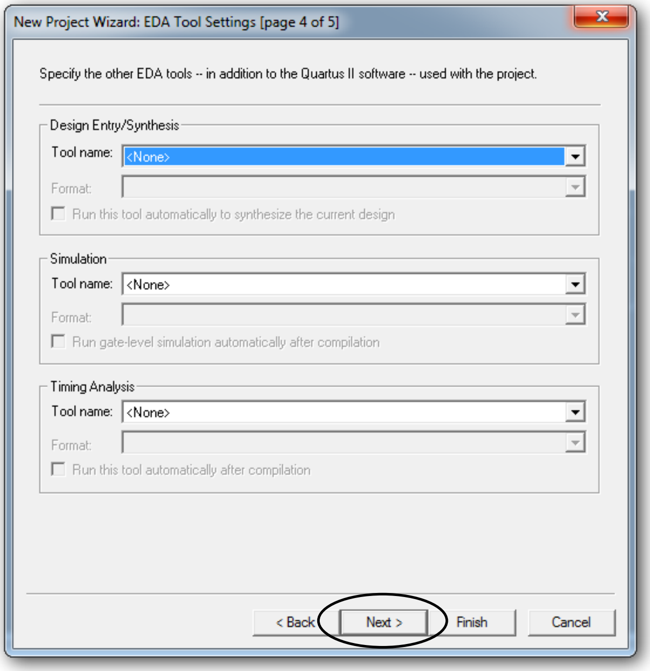
Companion device

HardCopy:

☑ Limit DSP & RAM to HardCopy device resources

Limit the choice of available devices by selecting these filter values.

Figure 6 Select Device Family (Cyclone II) and Device (EP2C20F484C7)



New Project Wizard: EDA Tool Settings [page 4 of 5]

Specify the other EDA tools -- in addition to the Quartus II software -- used with the project.

Design Entry/Synthesis

Tool name: <None>

Format:

□ Run this tool automatically to synthesize the current design

Simulation

Tool name: <None>

Format:

□ Run gate-level simulation automatically after compilation

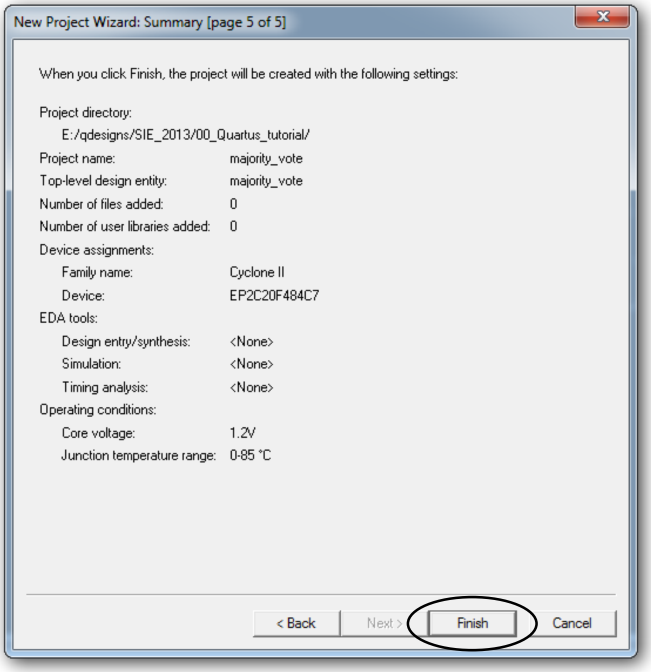
Timing Analysis

Tool name: <None>

Format:

□ Run this tool automatically after compilation

Figure 7 EDA Tool Settings (Click Next)



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

Project directory:	
E:/qdesigns/SIE_2013/00_Quartus_tutorial/	
Project name:	majority_vote
Top-level design entity:	majority_vote
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	Cyclone II
Device:	EP2C20F484C7
EDA tools:	
Design entry/synthesis:	<None>
Simulation:	<None>
Timing analysis:	<None>
Operating conditions:	
Core voltage:	1.2V
Junction temperature range:	0-85 °C

Figure 8 New Project Wizard Summary (click Finish)

Make a Block Diagram File (bdf)

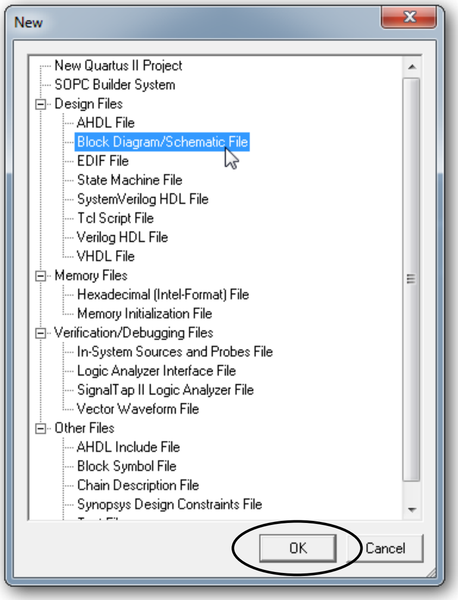
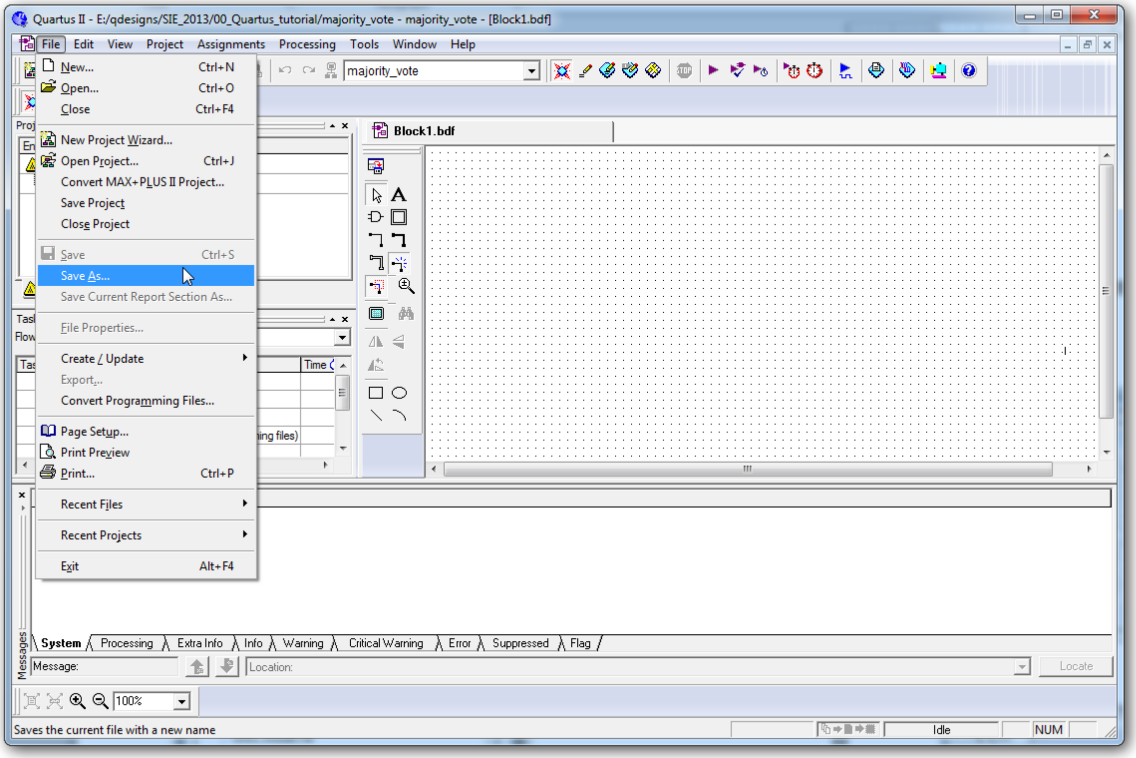


Figure 9 File Menu; New File Dialog Box (Block Diagram File)

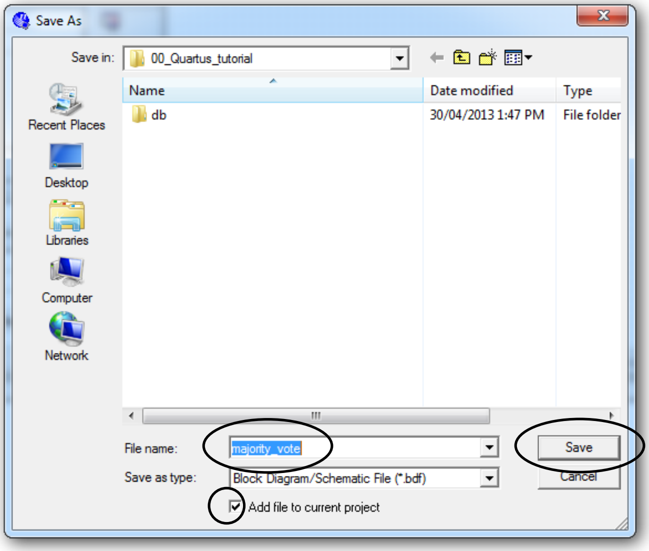


Quartus II - E:/qdesigns/SIE_2013/00_Quartus_tutorial/majority_vote - majority_vote - [Block1.bdf]

Saves the current file with a new name

Figure 10 Save Block Diagram File

Save the new Block Diagram file as shown in Figure 10 and Figure 11. In Figure 11, the automatic file name is the same as the project name you typed in Figure 4. Use this name. Also make sure that the box labeled **Add file to current project** is checked.



Save As

Save in: 00_Quartus_tutorial

Name	Date modified	Type
db	30/04/2013 1:47 PM	File folder

File name: majority_vote

Save as type: Block Diagram/Schematic File (*.bdf)

☑ Add file to current project

Figure 11 Dialog Box for "Save As " Function

Add Logic Symbols to the Block Diagram File

To make the diagram in Figure 1, we must add logic symbols to the block diagram file and connect them. Double-click in the Block Diagram File. Figure 12 will open.

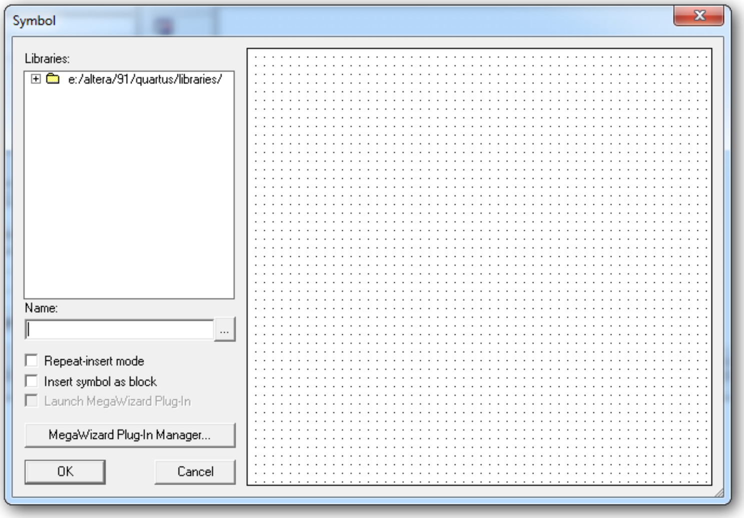
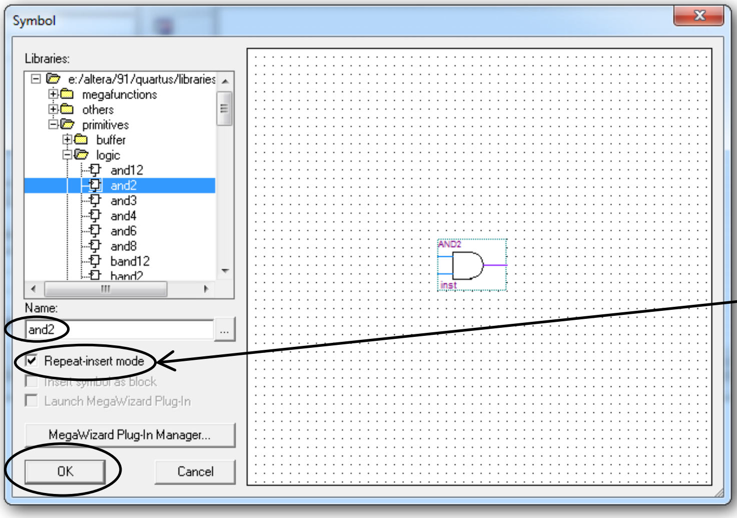

Symbol
Libraries:
e:/altera/91/quartus/libraries/
Name:
...
□ Repeat-insert mode
□ Insert symbol as block
□ Launch MegaWizard Plug-In
MegaWizard Plug-In Manager...
OK
Cancel

Figure 12 "Symbol" Dialog Box

The example circuit requires three 2-input AND gates. Type **and2** in the **Name** box and check the **Repeat-insert mode** box. Click **OK** and place three copies of the AND gate in the block diagram file. Press the **ESC** key when you have placed all three AND gates.



Check this box to insert multiple copies of the selected symbol.

Figure 13 "AND2" Symbol Selected

The component name for each component is shown at the top of the component symbol. For the example circuit we need the following components.

Quantity	Component Name	Component Function
3	and2	2-input AND gate
1	or3	3-input OR gate
3	input	Input pin
1	output	Output pin

Place these components as shown in Figure 14.

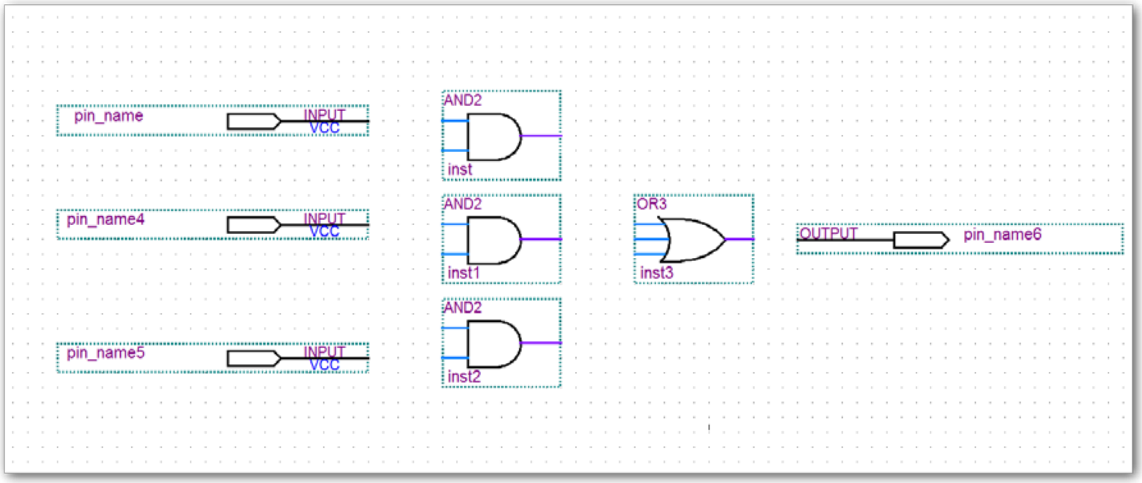


Figure 14 All Symbols for Example Circuit

Change the input and output pin names so that they are the same as in Figure 16. To change a pin name, double-click on the pin and retype the name.

The next four diagrams show how to connect the components. Figure 15 shows how the smart cursor becomes a right-angle line tool when you hover the mouse over a connection point on a component. To connect to another component, click and drag until you see a square over the second connection, as shown in Figure 16.

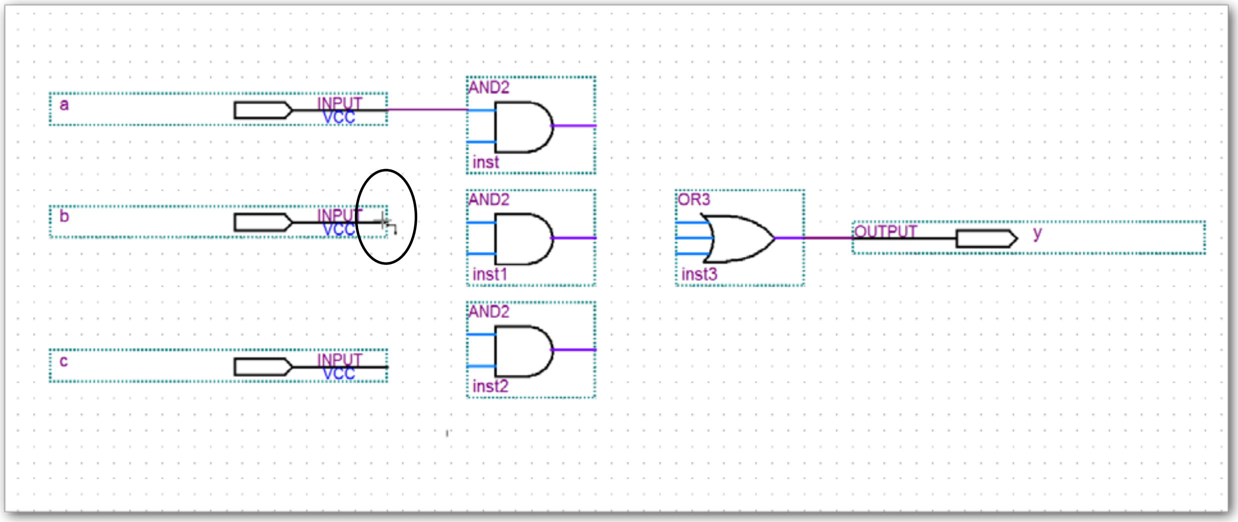


Figure 15 Smart cursor becomes a right-angle line tool

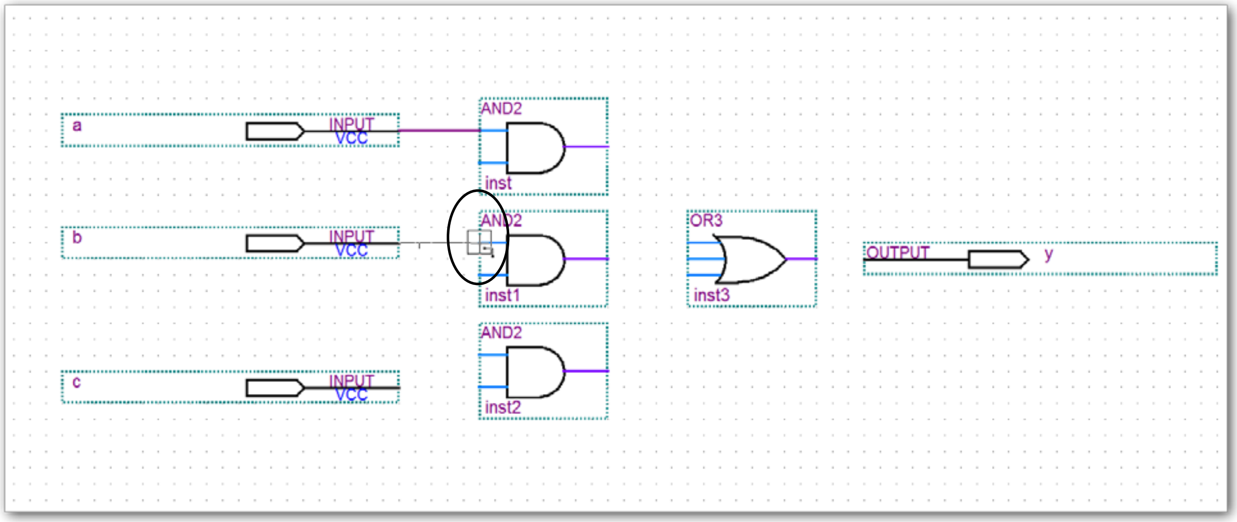


Figure 16 Cursor shows a square when a connection is made

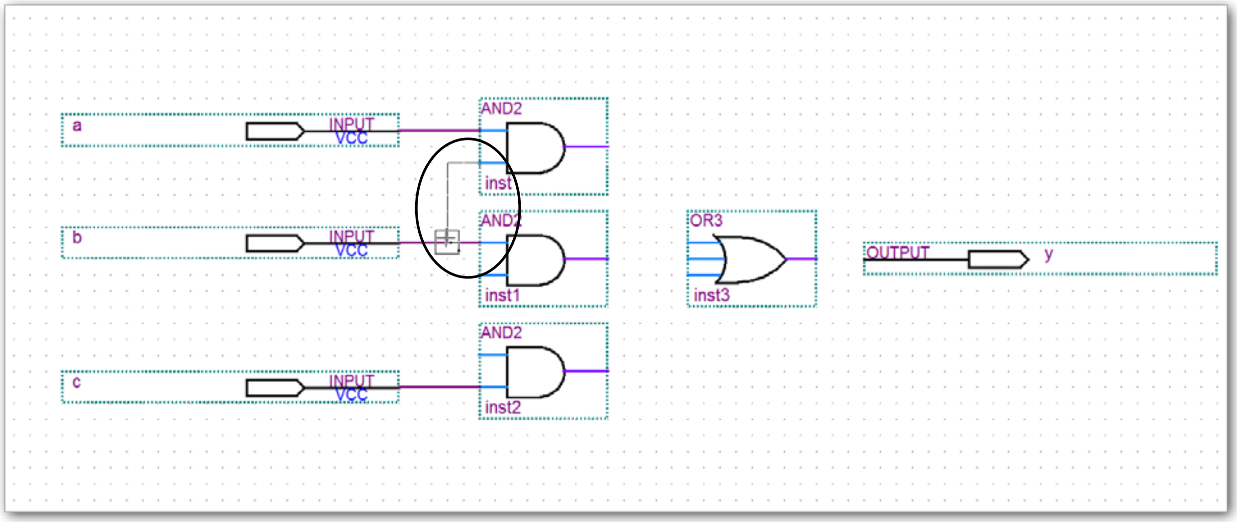


Figure 17 Line tool can draw one right-angle bend

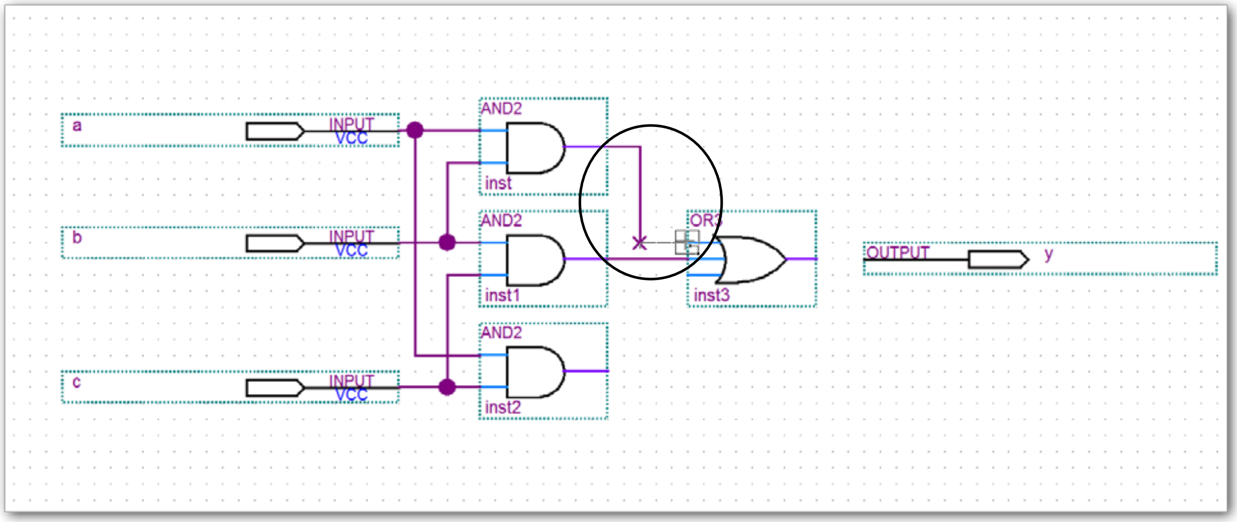
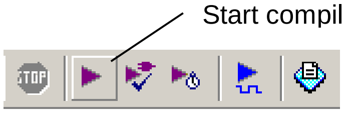


Figure 18 Line with two right angles must be drawn with two separate lines

Lab 4
Introduction to Quartus II

Compiling and Simulating a Design in Quartus II

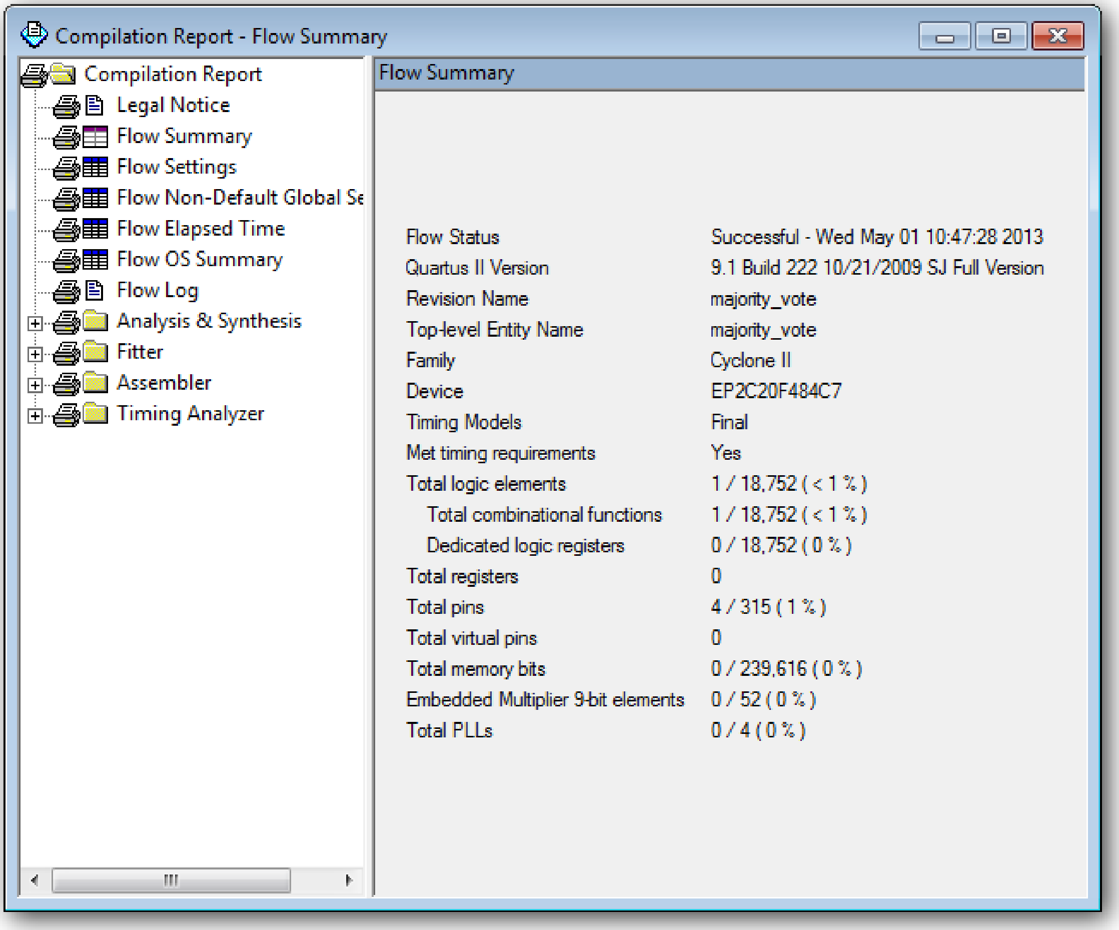
Before a design can be simulated, it must be **compiled**, that is, converted from the design format to a binary file. To start the compilation process, click the **Start Compilation** button on the Quartus II toolbar, as shown in Figure 19. Compilation can also be started by selecting **Start Compilation** from the Quartus II **Processing** menu or by using the keyboard shortcut **Ctrl+L**.



Start compilation.

Figure 19 Start Compilation (Toolbar Button)

When compilation is complete, Quartus II displays a **Compilation Report**, shown in Figure 20.



Flow Status	Successful - Wed May 01 10:47:28 2013
Quartus II Version	9.1 Build 222 10/21/2009 SJ Full Version
Revision Name	majority_vote
Top-level Entity Name	majority_vote
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	1 / 18,752 (< 1 %)
Total combinational functions	1 / 18,752 (< 1 %)
Dedicated logic registers	0 / 18,752 (0 %)
Total registers	0
Total pins	4 / 315 (1 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

Figure 20 Compilation Report Window

Simulation

A standard part of the FPGA design cycle is to create a **simulation** of a design before programming it into an FPGA. A simulation is a timing diagram which is generated by specifying a set of input waveforms to the design under test. The simulation software examines the design equations and input logic waveforms and calculates the response of the digital circuit, which is displayed as a set of output waveforms. The simulation allows us to determine if the design is working as planned by observing the response of the design to a defined input.

To create a good simulation we must ask, "How will we know that the circuit is working?", which leads to two further questions:

- What input values will fully test all functions of the design?
- What outputs should we see for each input condition?

The circuit must generate a HIGH output when a majority of inputs is HIGH. Since we have three inputs, this means two or more inputs must be HIGH. The most best way to test the circuit is to apply all possible input combinations to the circuit in an ascending binary sequence, in other words, to make its truth table.

Simulation
• Apply all possible input values to the circuit in an ascending binary sequence.
• For any input combination having two or more HIGH inputs (011, 101, 110, 111), the output must be HIGH.
• For any other input combination (000, 001, 010, 100), the output must be LOW.

A simulation is based on a **Vector Waveform File (vwf)**, which specifies simulation input values in the form of graphical waveforms. Output names are also listed, but their values are left for the simulator to generate. To create a new **vwf**, select **New** from the **File** menu or click the appropriate toolbar button. From the box in Figure 21, click the **Other Files** tab and select **Vector Waveform File**.

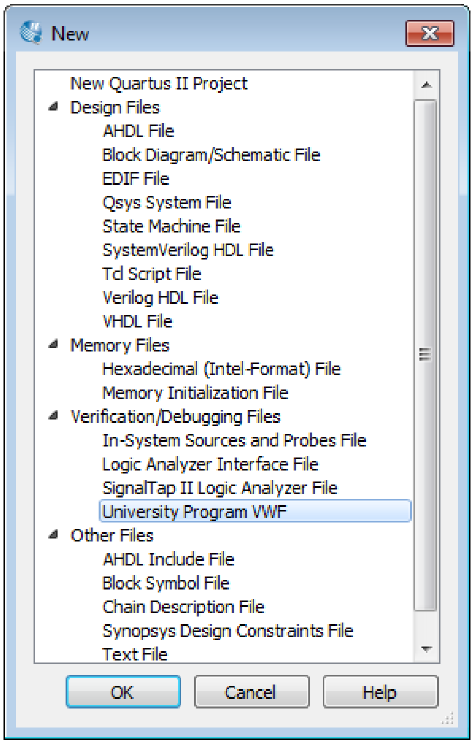


Figure 21 New File Dialog (Vector Waveform File)

The waveform editor window will appear, as shown in Figure 22.

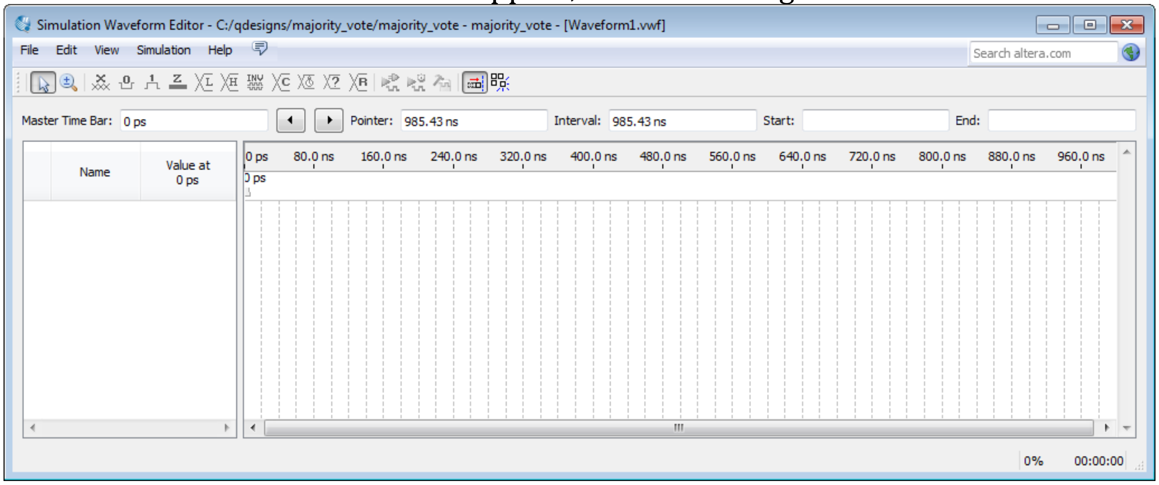
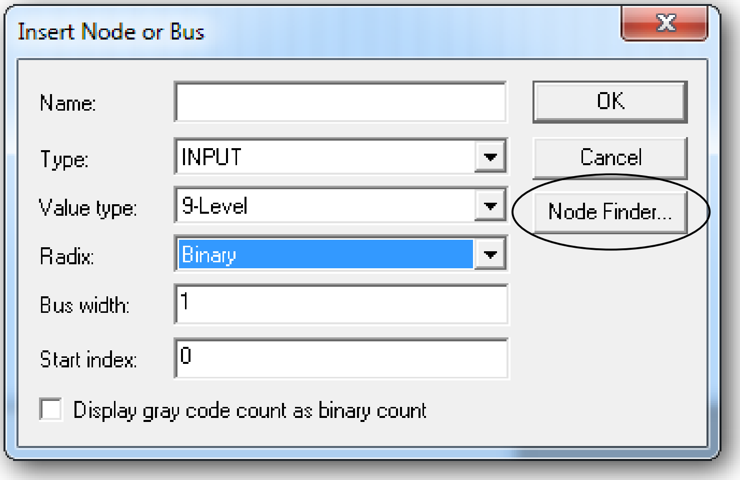


Figure 22 Default Waveform Editor Window

Add Nodes to the Simulation

A node in a simulation is an input or output signal. We can use the **Node Finder** utility to enter input and output nodes into our simulation.

To start the Node Finder, place the mouse cursor in the **Name** column of the Waveform Editor and double-click. A dialog box labeled **Insert Node or Bus** will appear, as shown in Figure 23. Select the **Radix** as **Binary** and click **Node Finder** to start the Node Finder utility, shown in **Error! Reference source not found.**.



Insert Node or Bus

Name:

Type: INPUT

Value type: 9-Level

Radix: Binary

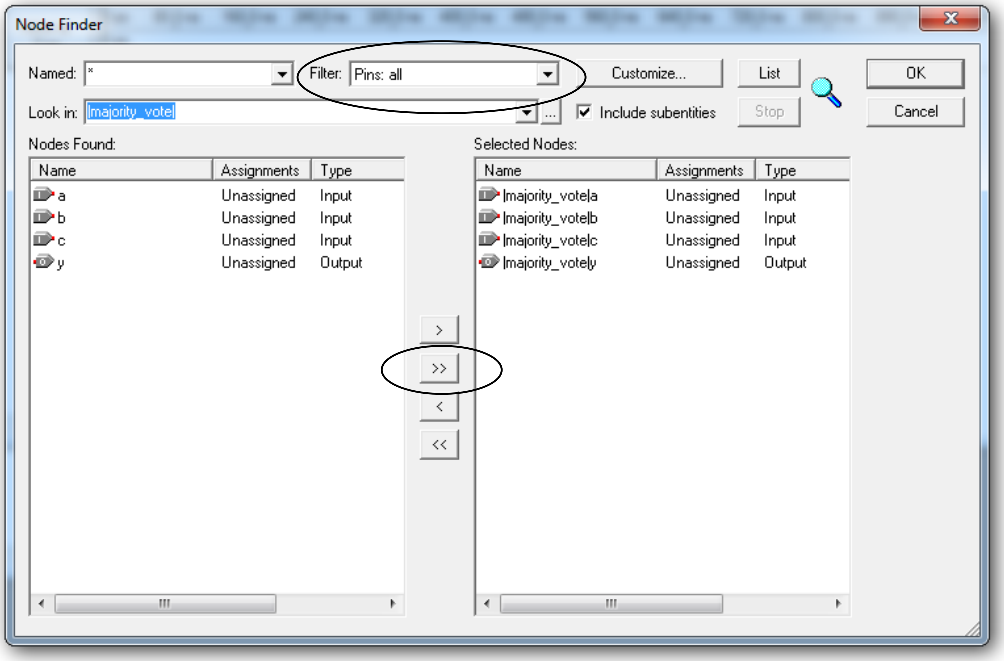
Bus width: 1

Start index: 0

□ Display gray code count as binary count

Figure 23 Insert Node or Bus Dialog

In the Node Finder dialog box, select **Pins: all** from the drop-down menu labeled **Filter**. Click **List** to see a list of all input and output nodes in the design. Click the right-going double-arrow button (shown at the left) to transfer all the pin names from the left side of the Node Finder window (shown in Figure 24 as **Nodes Found**) to the right side (**Selected Nodes**). Click **OK** to close the **Node Finder**. Click **OK** to close the **Insert Node or Bus** dialog. The selected nodes will appear in the Waveform Editor, as shown in Figure 25.



Named: *
Filter: Pins: all
Look in: |majority_vote|
☑ Include subentities

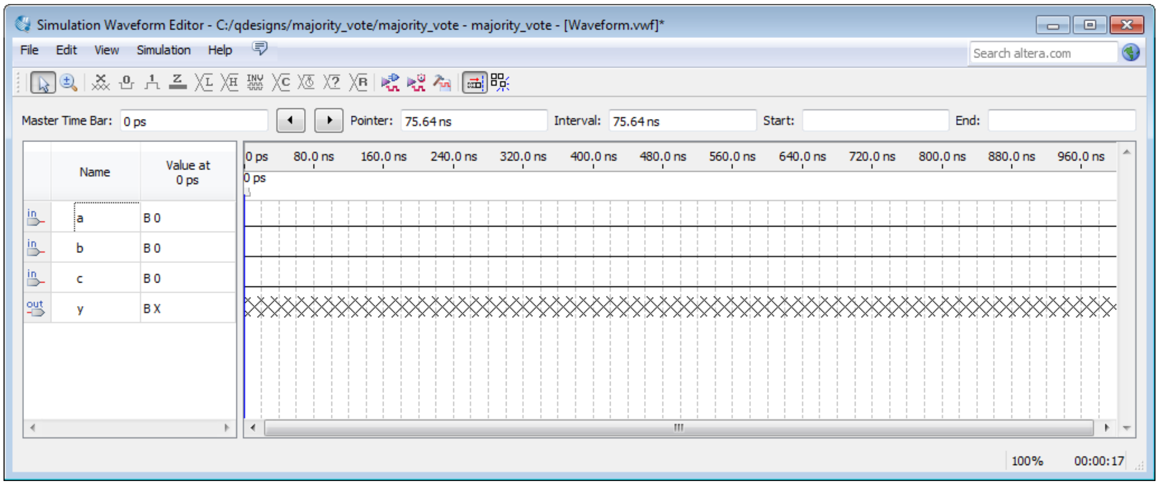
Nodes Found:

Name	Assignments	Type
a	Unassigned	Input
b	Unassigned	Input
c	Unassigned	Input
y	Unassigned	Output

Selected Nodes:

Name	Assignments	Type
|majority_vote|a	Unassigned	Input
|majority_vote|b	Unassigned	Input
|majority_vote|c	Unassigned	Input
|majority_vote|y	Unassigned	Output

Figure 24 Node Finder Window showing All Pin Names



Simulation Waveform Editor - C:/qdesigns/majority_vote/majority_vote - majority_vote - [Waveform.vwf]*

Master Time Bar: 0 ps Pointer: 75.64 ns Interval: 75.64 ns Start: End:

	Name	Value at 0 ps
in	a	B 0
in	b	B 0
in	c	B 0
out	y	B X

Figure 25 Waveform Editor showing Input and Output Nodes at Default Levels

Creating an input group

We could enter waveforms on **a**, **b**, and **c** individually to create an increasing 3-bit binary sequence, but it is easier and more accurate to group these waveforms together and apply the sequence to the whole group. Figure 26 shows a highlighted group of waveforms in the Quartus II Waveform Editor.

To highlight a group, press the CNTL key and click on waveform **a**, then **b**, then **c**.

Press CNTL and click **a** to **c** to highlight the waveform group.

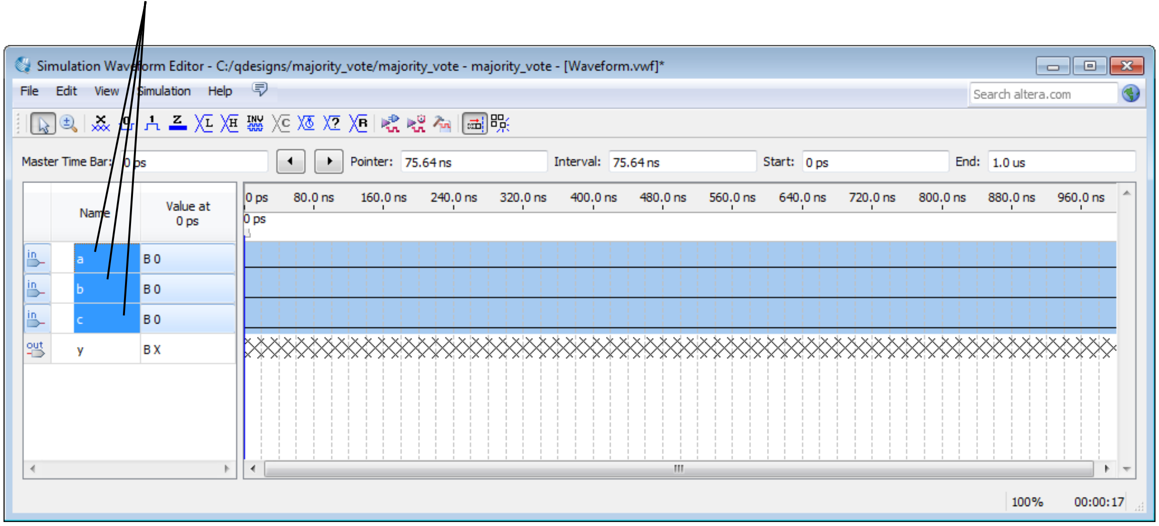
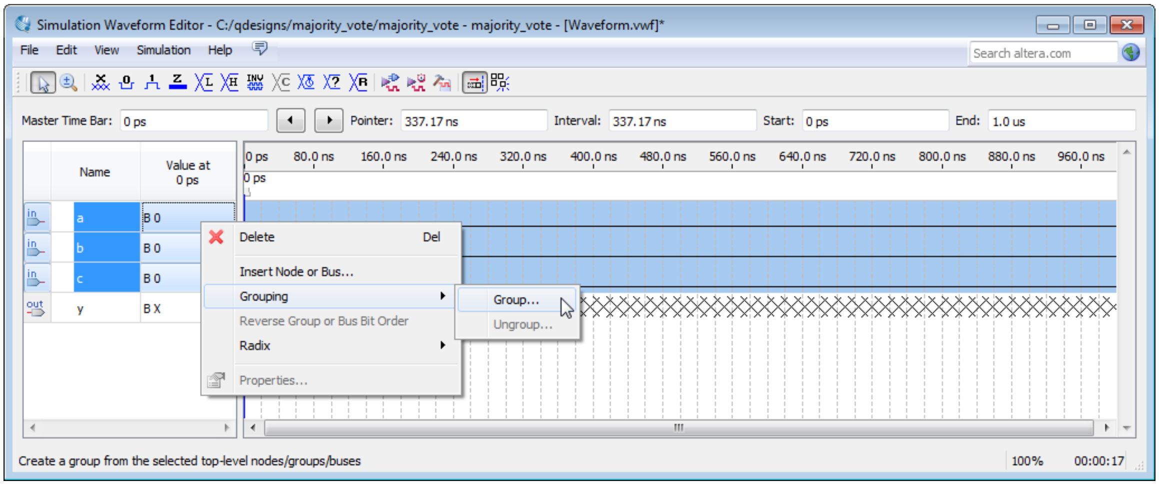


Figure 26 Highlighting a Group of Waveforms

Right-click on the highlighted group to get the pop-up menu shown in Figure 27. Select **Group** from the menu.



Simulation Waveform Editor - C:/qdesigns/majority_vote/majority_vote - majority_vote - [Waveform.vwf]*

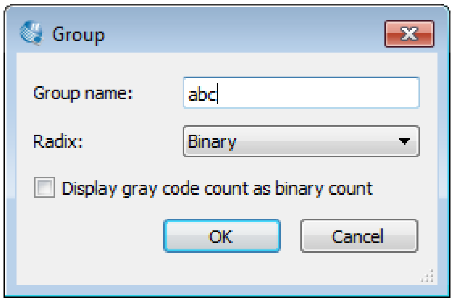
Master Time Bar: 0 ps | Pointer: 337.17 ns | Interval: 337.17 ns | Start: 0 ps | End: 1.0 us

Name	Value at 0 ps
a	B 0
b	B 0
c	B 0
y	B X

Create a group from the selected top-level nodes/groups/buses

Figure 27 Pop-up Menu showing Group Function

In the **Group** dialog box (Figure 28), type **abc** in the **Group name** box and select **Binary** for the **Radix** box. (The group name is a user choice; we use the name **abc** because it is descriptive of the group function and easy to type.) Click **OK** to accept the choices and close the box. The waveforms will appear as shown in Figure 29.



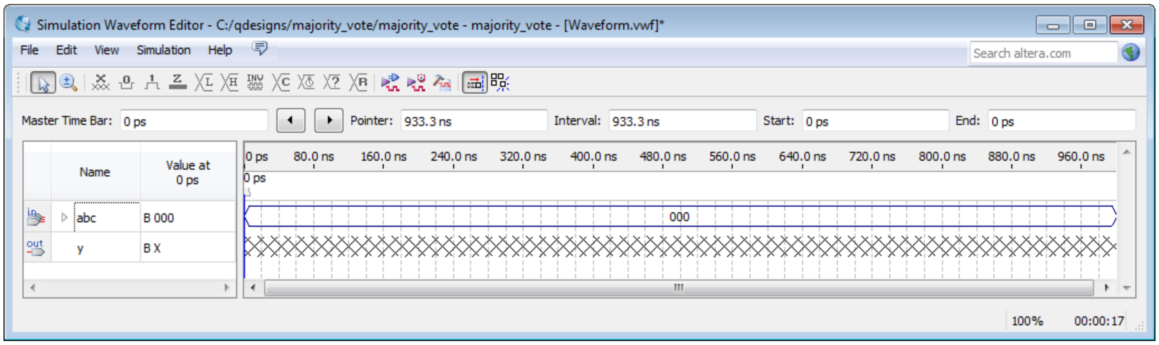
Group

Group name: abc

Radix: Binary

☐ Display gray code count as binary count

Figure 28 Group Dialog Box



Simulation Waveform Editor - C:/qdesigns/majority_vote/majority_vote - majority_vote - [Waveform.vwf]*

Master Time Bar: 0 ps | Pointer: 933.3 ns | Interval: 933.3 ns | Start: 0 ps | End: 0 ps

Name	Value at 0 ps
abc	B 000
y	B X

Figure 29 Waveform Editor Showing Input Group

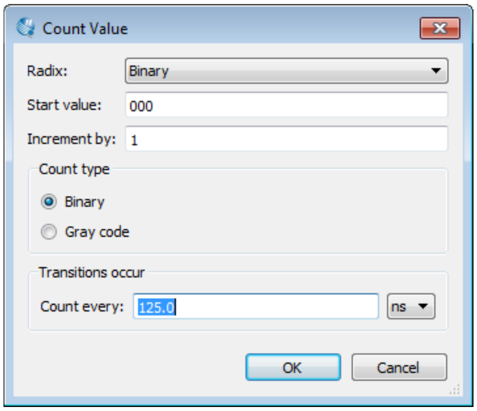
Add a Binary Count to the Input Group

The easiest way to go through all possible combinations of inputs is to add a binary count sequence to the input group **abc**. The simulation has a default end time of 1 μs (or 1000 ns). A 3-bit sequence has eight possible states. To fill the whole simulation with one cycle of the 3-bit sequence (from 000 to 111), we should divide up the 1000 ns into eight slices: we count every 1000 ns/8 = 125 ns.

Click on the **abc** waveform to highlight it, then click the **Count Value** button from the Waveform Editor toolbar, as shown at the left, or select **Value, Count Value** from either the **Edit** menu or from the pop-up menu that appears when you highlight **abc** and right-click.

Fill in the **Counting** tab of the **Count Value** dialog box, shown in Figure 30, as follows:

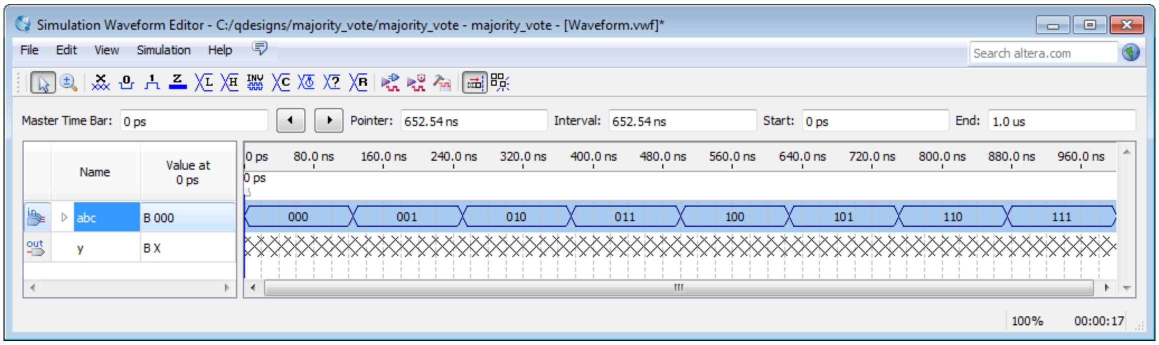
Radix: Binary
Start Value: 000
Increment by: 1
Count type: Binary
Count every: 125 ns



Radix: Binary
Start value: 000
Increment by: 1
Count type
● Binary
○ Gray code
Transitions occur
Count every: 125.0 ns

Figure 30 Count Value Dialog

Click **OK** to close the box. The binary count appears on input waveform **inputs**, as shown in Figure 31.



Simulation Waveform Editor - C:/qdesigns/majority_vote/majority_vote - majority_vote - [Waveform.vwf]*

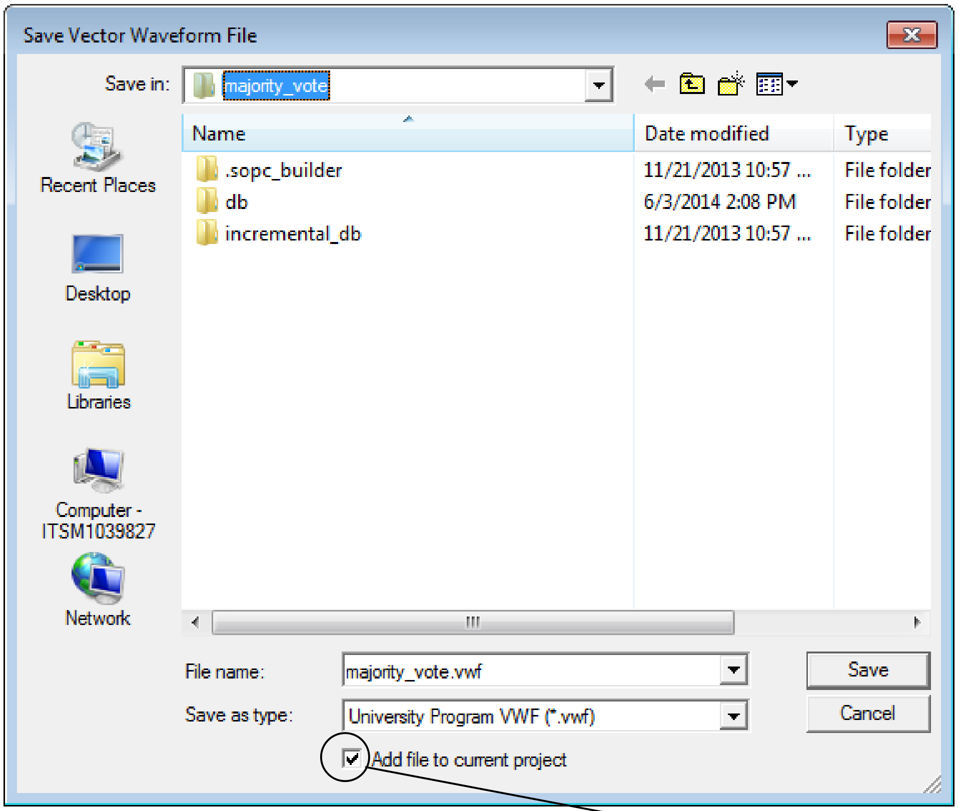
Master Time Bar: 0 ps Pointer: 652.54 ns Interval: 652.54 ns Start: 0 ps End: 1.0 us

Name	Value at 0 ps
abc	B 000
y	B X

Figure 31 Waveform Editor with Binary Input Count

Run Simulation and Save Vector Waveform File

Save the Vector Waveform File as **majority_vote.vwf** in your working folder. The box labeled **Add file to current project** should be checked.



Save Vector Waveform File

Save in: majority_vote

Name	Date modified	Type
.sopc_builder	11/21/2013 10:57 ...	File folder
db	6/3/2014 2:08 PM	File folder
incremental_db	11/21/2013 10:57 ...	File folder

File name: majority_vote.vwf

Save as type: University Program VWF (*.vwf)

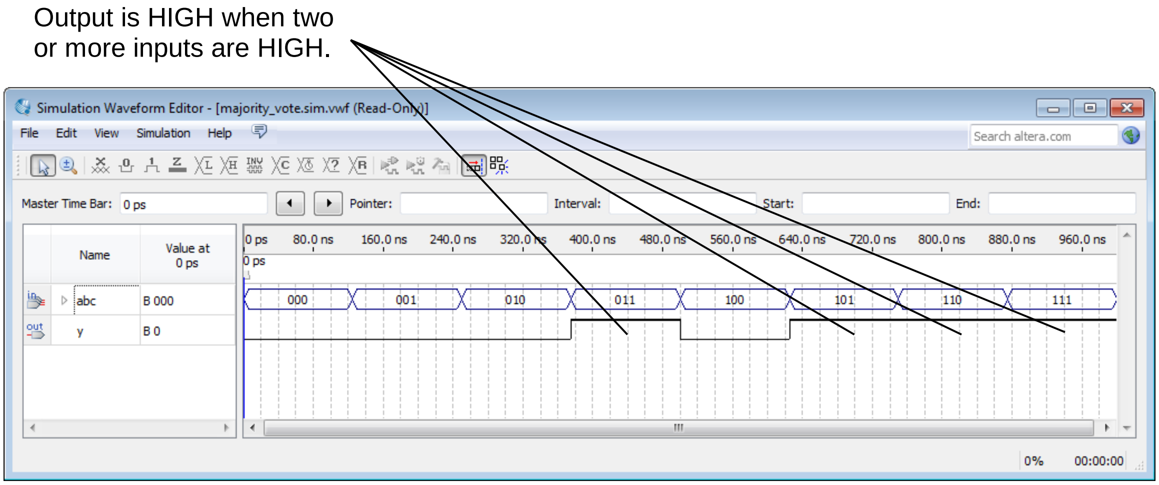
☑ Add file to current project

Figure 32 Saving a Vector Waveform File

This box should be checked.

Start the simulation by clicking the either of the toolbar buttons shown at the left or by selecting an item from the **Simulation** menu. The first icon on the left selects a **Functional Simulation**, which only accounts for the Boolean function of the circuit. The second icon selects a **Timing Simulation**, which accounts for the Boolean function of the circuit and physical timing delays that occur inside the circuit. For now, use the Functional Simulation.

After the simulation runs, the simulation report waveforms should appear, as shown in Figure 33. (Result: Output is HIGH when two or more inputs are HIGH.)


Output is HIGH when two or more inputs are HIGH.

Simulation Waveform Editor - [majority_vote.sim.vwf (Read-Only)]
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer:
Interval:
Start:
End:
Name
Value at 0 ps
0 ps 80.0 ns 160.0 ns 240.0 ns 320.0 ns 400.0 ns 480.0 ns 560.0 ns 640.0 ns 720.0 ns 800.0 ns 880.0 ns 960.0 ns
0 ps
abc B 000
y B 0
000 001 010 011 100 101 110 111
0%
00:00:00

Figure 33 Simulation Report Waveforms

Show the simulation result to your instructor.

Instructor's Initials______

Transferring a Design to a Target FPGA

After we have completed the compilation and simulation steps and are satisfied that our design is free of conceptual errors, we can convert the project to a physical design within our target FPGA. This involves three steps.

1. We must **assign pin numbers** to each of the pin names previously assigned.
2. We must **recompile the file** so as to get the programming information to correspond to the new pin assignments.
3. We must **use the Quartus II programming tool** to transfer the design from our PC to the target device.

Note: *In order to assign pins and program the FPGA, your project must have the correct device assignment. Check this by opening the* ***Device*** *dialog box from the Quartus II* ***Assignments*** *menu. For the Altera DE1 board, the device family is Cyclone II and the device number is EP2C20F484C7.*

Assigning Pins

The target FPGA (EP2C20F484C7) has 484 pins, 315 of which can be programmed by the user to act as an input or output. Because any of these 315 pins can be used for any user function, we must specify a mapping between the input/output functions and the FPGA pins assigned to these functions.

We can assign pins either from a utility called the **Pin Planner** or from the Quartus II Assignment Editor. The procedure for using the Pin Planner is outlined below.

We will use the following pin assignments for the majority vote example:

Pin Name	Input/Output Device	Pin Number
a	SW[2]	M22
b	SW[1]	L21
c	SW[0]	L22
y	LEDG0	U22

Using the Quartus II Pin Planner

From the **Assignments** menu, select **Pin Planner**, as shown in Figure 34. The **Pin Planner** screen, shown in Full Screen mode in Figure 35, will appear.

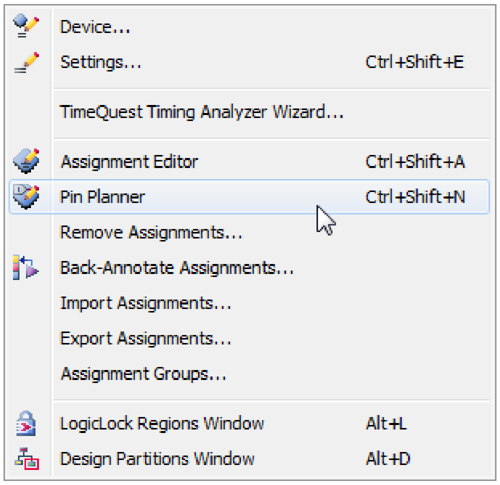
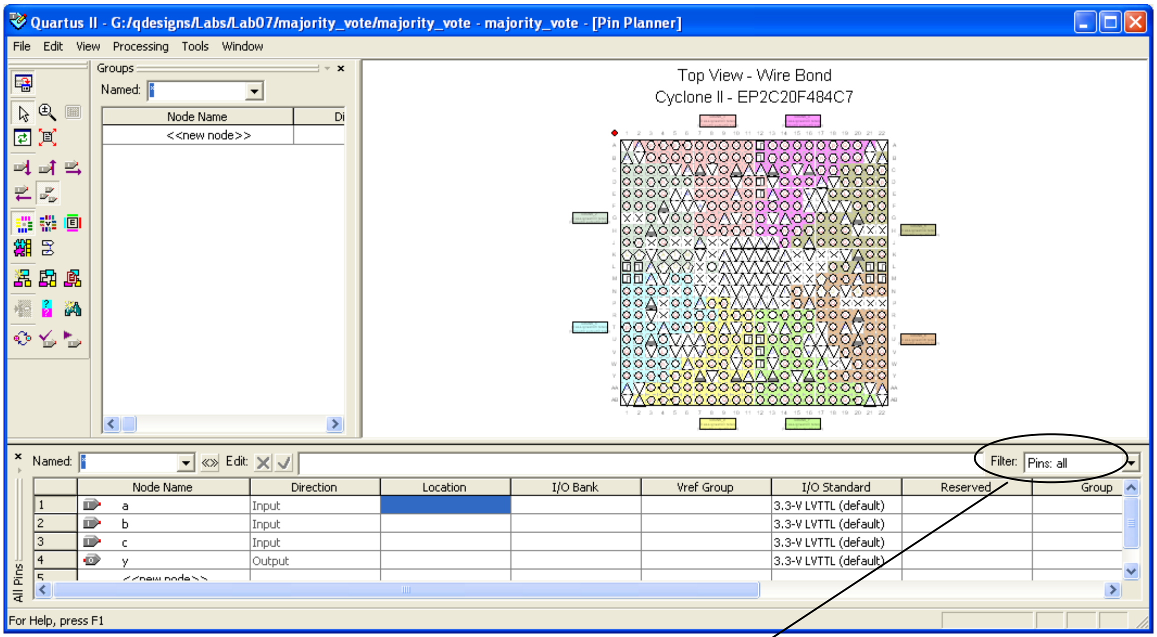

Device...
Settings... Ctrl+Shift+E
TimeQuest Timing Analyzer Wizard...
Assignment Editor Ctrl+Shift+A
Pin Planner Ctrl+Shift+N
Remove Assignments...
Back-Annotate Assignments...
Import Assignments...
Export Assignments...
Assignment Groups...
LogicLock Regions Window Alt+L
Design Partitions Window Alt+D

Figure 34 Assignments Menu (Pins)



Quartus II - G:/qdesigns/Labs/Lab07/majority_vote/majority_vote - majority_vote - [Pin Planner]

Filter: Pins: all

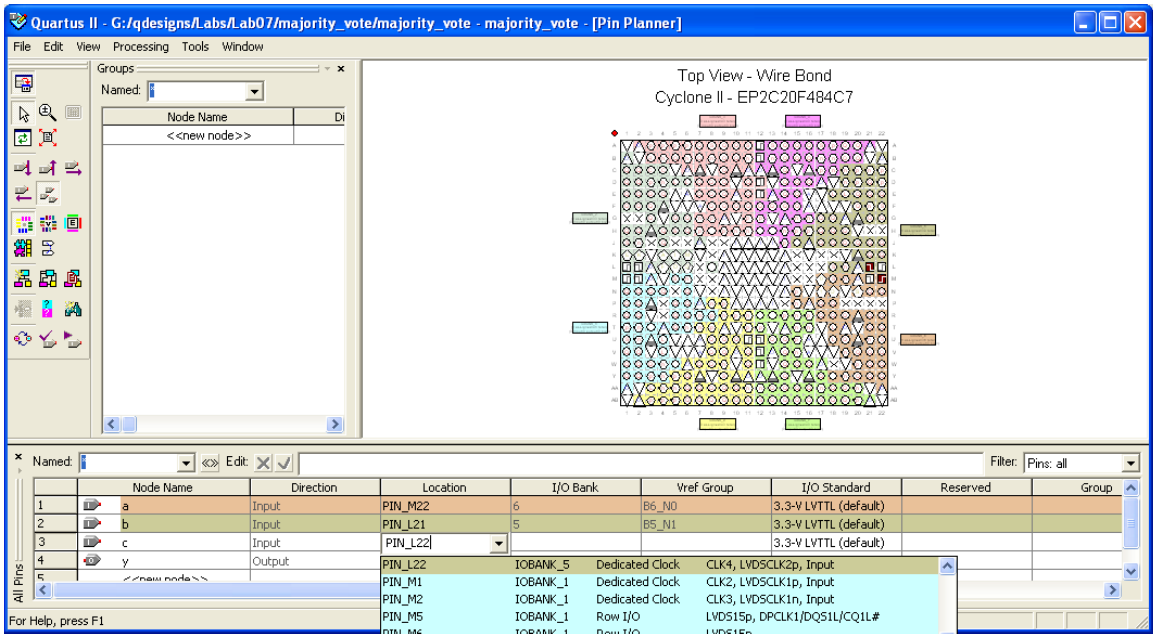
	Node Name	Direction	Location	I/O Bank	Vref Group	I/O Standard	Reserved	Group
1	a	Input				3.3-V LVTTL (default)		
2	b	Input				3.3-V LVTTL (default)		
3	c	Input				3.3-V LVTTL (default)		
4	y	Output				3.3-V LVTTL (default)		
5	<<new node>>							

Figure 35 Pin Planner

Set **Filter** to **Pins: all**.

If the **Filter** box is set to **Pins: all**, then all inputs and outputs are listed under the **Node Name** column. Pins **TCK**, **TDI**, **TDO**, and **TMS** are standard pins used by the device programming interface. They do not need to be assigned; Quartus II does this automatically.

To assign a pin number, double-click in the **Location** column beside the pin name you want to assign and select a pin number from the drop-down box that appears. Figure 36 shows this process for assignment of pin L22 to input **c**.



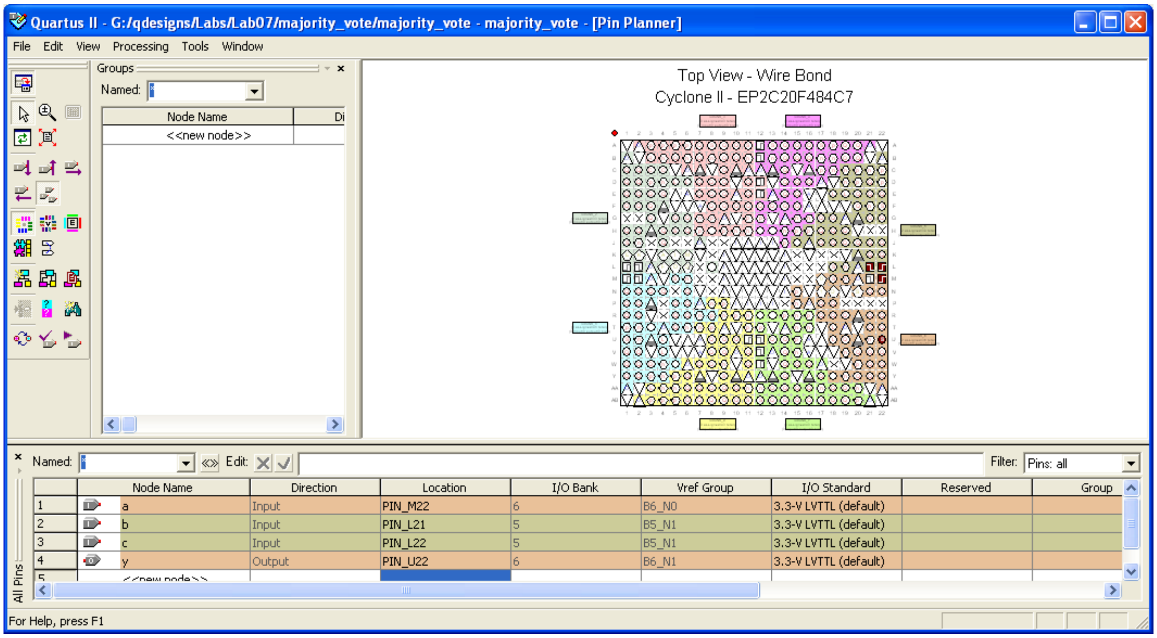
Quartus II - G:/qdesigns/Labs/Lab07/majority_vote/majority_vote - majority_vote - [Pin Planner]

	Node Name	Direction	Location	I/O Bank	Vref Group	I/O Standard	Reserved	Group
1	a	Input	PIN_M22	6	B6_N0	3.3-V LVTTL (default)		
2	b	Input	PIN_L21	5	B5_N1	3.3-V LVTTL (default)		
3	c	Input	PIN_L22			3.3-V LVTTL (default)		
4	y	Output						
5	<<new node>>							

PIN_L22	IOBANK_5	Dedicated Clock	CLK4, LVDSCLK2p, Input
PIN_M1	IOBANK_1	Dedicated Clock	CLK2, LVDSCLK1p, Input
PIN_M2	IOBANK_1	Dedicated Clock	CLK3, LVDSCLK1n, Input
PIN_M5	IOBANK_1	Row I/O	LVDS15p, DPCLK1/DQS1L/CQ1L#

Figure 36 Assigning a Pin Number

Once the pins have been assigned, they will appear in the Pin Planner, as shown in Figure 37, and the Block Diagram File, as shown in Figure 38. ***In order for the pin assignments to "stick", you must compile the project again.***



Quartus II - G:/qdesigns/Labs/Lab07/majority_vote/majority_vote - majority_vote - [Pin Planner]

	Node Name	Direction	Location	I/O Bank	Vref Group	I/O Standard	Reserved	Group
1	a	Input	PIN_M22	6	B6_N0	3.3-V LVTTL (default)		
2	b	Input	PIN_L21	5	B5_N1	3.3-V LVTTL (default)		
3	c	Input	PIN_L22	5	B5_N1	3.3-V LVTTL (default)		
4	y	Output	PIN_U22	6	B6_N1	3.3-V LVTTL (default)		
5	<<new node>>							

Figure 37 All Design Input and Output Pins Assigned

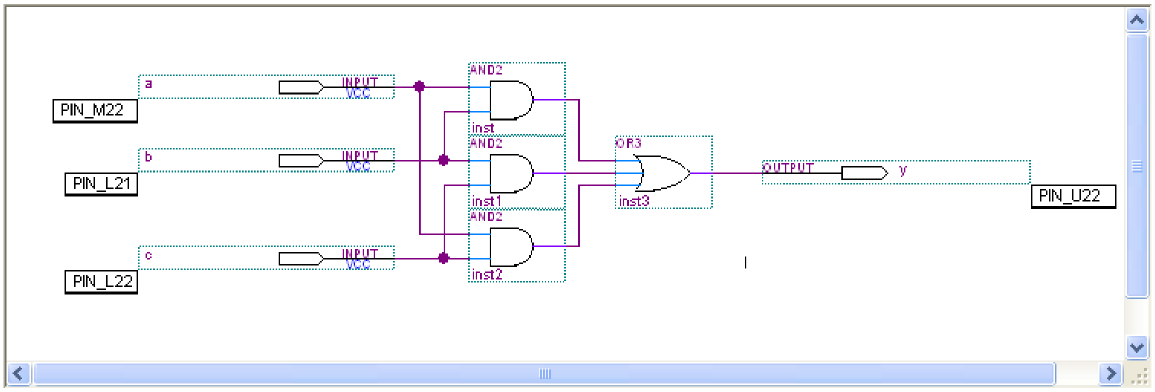


Figure 38 Block Diagram File Showing Pin Numbers

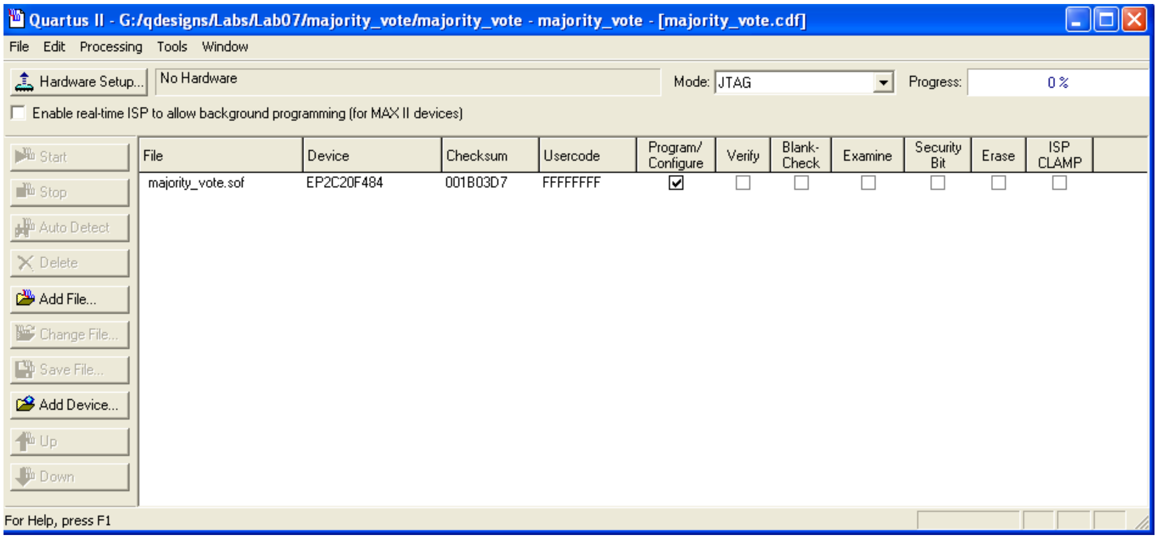
Programming FPGAs on the Altera DE1 circuit board

The FPGA on the Altera DE1 circuit board is programmed via the programming software in Quartus II and a USB cable called the **USB-Blaster**.

Quartus II Programmer

To program a device on the Altera DE1 board, connect the USB-Blaster cable from the USB connector near the power jack on the DE1 board to a USB port on the PC running Quartus II.

Start the Quartus II Programmer, either with the toolbar button shown at the left or by selecting **Programmer** from the **Tools** menu. The programmer dialog box (Figure 39) will open, showing the programming file for the top-level file of the open project.



No Hardware

Mode: JTAG

Progress: 0 %

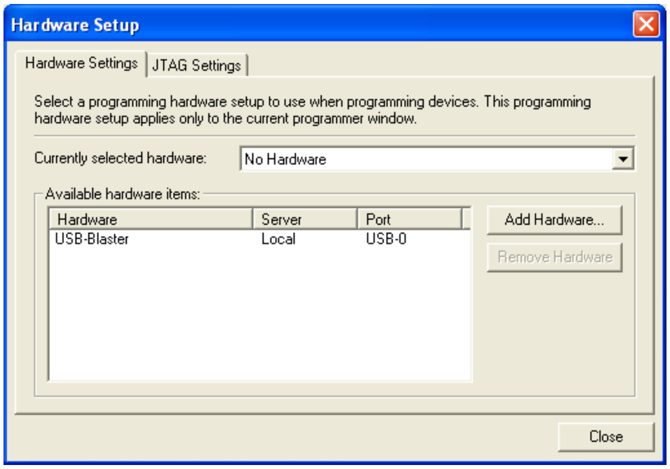
□ Enable real-time ISP to allow background programming (for MAX II devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
majority_vote.sof	EP2C20F484	001B03D7	FFFFFFFF	☑	□	□	□	□	□	□

Figure 39 Programmer Dialog

Note: *If you have not programmed an FPGA with your particular version of Quartus II, you will need to set up the programming hardware before proceeding. If you have at any time programmed an FPGA with your PC running this version of Quartus II, you do not need to perform the following hardware setup steps. An exception is for the PCs in the RRC labs, which reset to a default state upon boot-up and may need to be configured each time the machine is started.*

Click the **Hardware Setup** button at the top left corner of the Programmer dialog box. The **Hardware Setup** dialog box, shown in Figure 40, will open.



Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

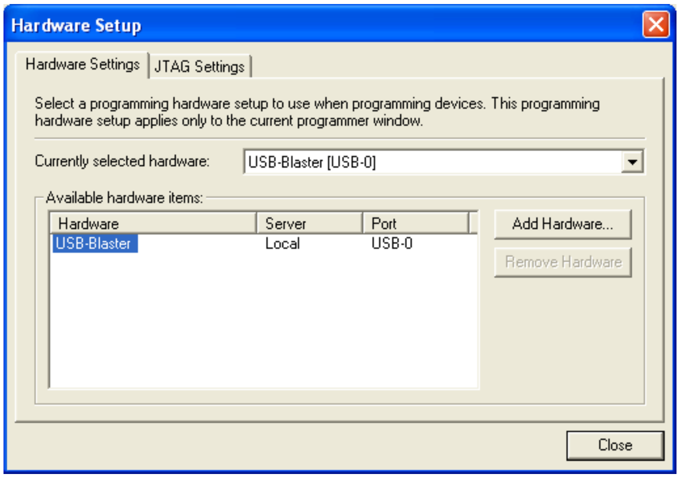
Currently selected hardware: No Hardware

Available hardware items:

Hardware	Server	Port
USB-Blaster	Local	USB-0

Figure 40 Hardware Setup Dialog

In the **Hardware Setup** dialog, highlight **USB-Blaster** in the **Available hardware items** box by clicking the item, then double-click the item. The result will be as shown in Figure 41. Click **Close** to return to the Programmer dialog.



Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

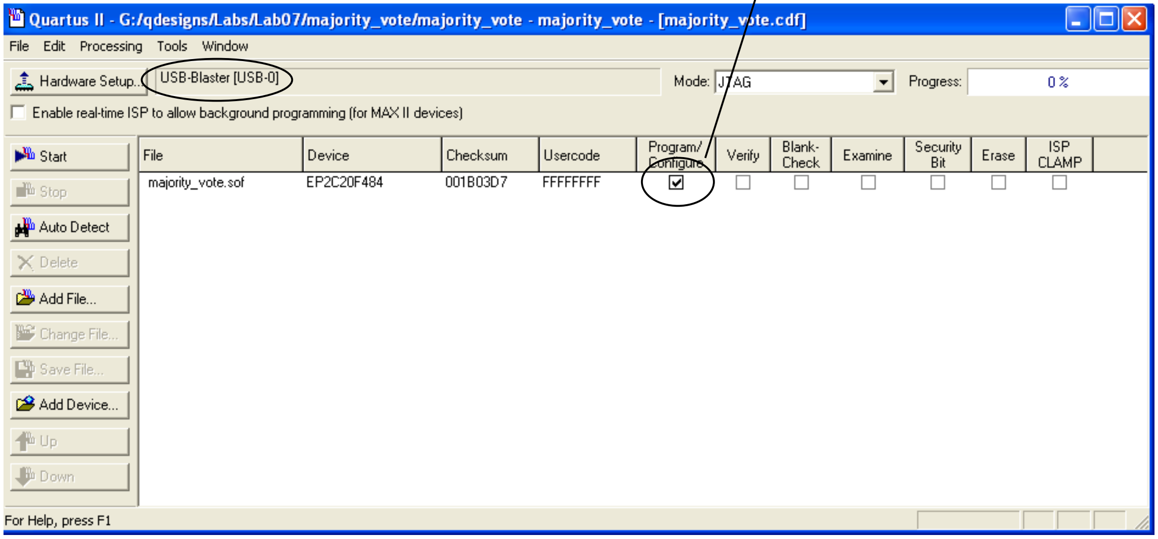
Currently selected hardware: USB-Blaster [USB-0]

Available hardware items:

Hardware	Server	Port
USB-Blaster	Local	USB-0

Figure 41 Hardware Setup Dialog Showing Selected Hardware

Figure 42 shows the programmer dialog box, now with the programming hardware selected. In order to program the FPGA, highlight the required programming file by clicking it, then select the checkbox for **Program/Configure**.



Quartus II - G:/qdesigns/Labs/Lab07/majority_vote/majority_vote - majority_vote - [majority_vote.cdf]

Hardware Setup... USB-Blaster [USB-0]   Mode: JTAG   Progress: 0 %

□ Enable real-time ISP to allow background programming (for MAX II devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
majority_vote.sof	EP2C20F484	001B03D7	FFFFFFFF	☑	□	□	□	□	□	□

For Help, press F1

Figure 42 Programmer Dialog with Programming Hardware and File Selected

Start programming the FPGA by clicking the **Start** button. The window might disappear during programming, but can be brought to the front again to check the programming status.

Once programming is complete, the majority vote circuit can be tested by checking all binary combinations of the switches assigned to inputs A, B, and C and observing the state of the LED assigned to output Y. The table of pin assignments and their corresponding input or output devices are shown again on this page for your reference. Fill in the following truth table for the design example:

A	B	C	Y

Instructor's Initials____

Lab 4
Introduction to Quartus II

Pin Name	Input/Output Device	Pin Number
a	SW[2] – Switch 2	M22
b	SW[1] – Switch 1	L21
c	SW[0] – Switch 0	L22
y	LEDG0 – Green LED 0	U22