University of Florida   EEL 3701   Dr. Eric M. Schwartz
Department of Electrical & Computer Engineering   Revision 0   26-Aug-19
Page 1/9

Quartus Tutorial with Basic Graphical Gate Entry and Simulation
(Last verified for Quartus Prime Lite Edition 18.1)

Tips
- Create a folder where you will store all of your Quartus projects.
- Do **NOT** use spaces and special characters in Quartus file/project names.
- Accept defaults when saving files.
- Use **labels** instead of crossing wires.

Example Problem

Given the logic equation $Y = A*/B + /C$, implement this equation using a two input AND gate, a two input OR gate and two inverters under the Quartus environment. Upon completion of the schematic entry portion of the example, simulate the circuit and print out copies of the circuit & simulation results. **Draw a Logic Table and Voltage Table (in counting order) for the inputs & output** and you will later compare it with the simulation results. Assume A, B, C, and Y are active high signals so your voltage table will look identical to your truth table but instead of 0's there will be L's and instead of 1's there will be H's.

I. Creating a Project in Quartus

A. New Project Design Creation
1. Setup a **local** 'lab1_ex' directory (or folder) on your PC to hold your design & simulation files.
2. Launch the Altera Quartus software under Windows.
3. Open the New Project Wizard by selecting the New Project Wizard icon or by selecting "File" and "New Project Wizard ...". Select "Next" (feel free to check the "Don't show me this introduction again" box).
4. Under "What is the working directory for this project", use the "…" button to browse and select the directory you created in step number 1.
5. Under "What is the name of this project", name the project lab1_ex. (This may already be done for you. This should also make the top-level design entity name lab1_ex. If not, again type in lab1_ex.) Select "Next", then "Next" (for "Empty project"), and "Next" (for "Add files"). Note that dashes should **NOT** be used in file names. The simulator has a problem with dashes.
6. Change the "Family" under "Device family" to **MAX 10 (DA/DF/DC/SA/SC)**. In the "Name filter" enter **10M02SCU169C8G** or under "Available devices" select this device. Select "Next". The MAX 10 is an FPGA.
   - Previously, we used the following: Change the "Family" under "Device family" to **MAX V**. Under "Available devices" select **5M570ZT100C5**. Select "Next".
7. In "EDA Tool Setting", for "Simulation" chose "ModelSim-Altera" and for "Format(s)" chose "VHDL".
8. Select "Next" and then "Finish".

B. Configuring Quartus Window
1. By default, the Project Navigator (left, top), Tasks (left bottom), Messages (bottom), and IP Catalog (right) should all be on the screen with a big area in the middle for your design. If not, select "View" and "Utility Windows" to create each of these windows.

2. Select "Tools" and "Customize." Make sure "File", "Standard" and "Applications" are selected. "Feedback" can also be optionally selected.

II. Designing

A. Creating a BDF

Figure 0: Button for new bdf

1. If you have not already done so, create a new bdf by selecting the bdf button (see Figure 0) or by selecting the pull-down button "File" and "New | Device Design Files | Block Diagram/Schematic File" and then press "OK".
2. Remove the grid dots by selecting "View" and "Show Guidelines".
3. Select "File" and "Save As". Save your bdf as "lab1_ex" in your lab1_ex" project directory. If you used the directory (folder name) that I suggested, then this should be the default name. The file will be given the bdf extension; bdf stands for "block design file" and contains schematics, symbols or block diagrams.
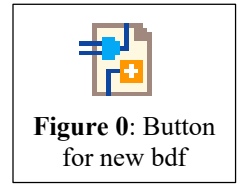
B. Adding Text
1. Select the "A" in the toolbar of your bdf window.
2. Select a point near the top left in the window with the left mouse key. Type "**Lab # Part #**", where # is replaced by that week's lab number and the lab part number respectively and then hit the "Enter" key.
3. Type your "**Name:**" followed by your first and last name, and then hit the "Enter" key.
4. Type your "**Class: ####**" and "**Section: ####**" and then hit the "Enter" key.
5. Type "**PI Name:**", followed by your PI's first and last name. Hit the "Enter" key.
6. Type "**Description:**" followed by an appropriate description. In this case type the following equation, "**Y = A /B + /C,**" and then hit the "Enter" key. See Figure 1 for a sample of header for this course.

Lab 1 Part 1
Name: Al E. Gator
Class:12345
Section: 12AB
PI Name: Alli Gator
Description: Y =A /B + /C

Figure 1: Sample bdf heading

7. Hit the "Esc" (escape) key to end text additions.
8. If your text is getting clipped off/cut off, this may be due to a resolution issue. This following link has instructions on resolving this issues: Quartus Display Issues. If there is still a problem with parts being hidden in part of the screen, just detach the design (bdf) file by selecting the "Window" tab and then "Detach Window".

C. Component Selection Process and Moving Components
1. With your mouse pointing inside the bdf, double-click (or click the right mouse key and select "Insert" then

University of Florida | EEL 3701 | Dr. Eric M. Schwartz
Department of Electrical & Computer Engineering | Revision 0 | 26-Aug-19
Page 2/9 **Quartus Tutorial with Basic Graphical Gate Entry and Simulation**
(Last verified for Quartus Prime Lite Edition 18.1)

"Symbol"). The "Symbol" dialog box will appear. This window lists the available Altera libraries.

2. Select the > icon to expand the "c:/intelfpga_lite/18.1/quartus/libraries/" folder.
3. Select the > icon to expand the "primitives" folder and then expand the "logic" folder.
4. Select the "and2" component by double clicking on it (or by selecting it with a single click, then selecting "OK").
5. Hit the left mouse key when the pointer is at the desired location in the bdf to insert the AND symbol into the design file.
6. Double-click in the bdf window. Insert an "OR2" gate by typing **or2** into the box under "Name:". Hit the keyboard "Enter" key or press "OK".
7. Place the OR symbol into the bdf.
8. Double-click in the bdf window. This time type **not** into the box (to insert a Level Shifter) and click on the box next to "Repeat-insert mode".
9. Click the pointer at the desired location in the bdf to insert the first NOT symbol into the design file. Now click on another desired location in the bdf to insert a second NOT symbol.
10. Hit the "Esc" (escape) key to end Repeat-insert mode.
11. Select the magnifying glass in the toolbar of your bdf window. Select a point in the window with the left mouse button. Notice that the image gets larger with the center of the enlargement at the point you selected. Now select a point in the window with the right mouse button. Notice that the image gets smaller with the center of the enlargement at the point you selected.
12. Hit the "Esc" (escape) key to end magnifying options or select the pointer symbol to return to selection mode.
13. Rearrange your devices in approximately the placement you would like for the logic diagram you are trying to construct. You can move a component by selecting it with your mouse, and either holding down the left mouse button and moving it to another location on your bdf or using your up/down/left/right arrow keys. The window should look similar to Figure 2.
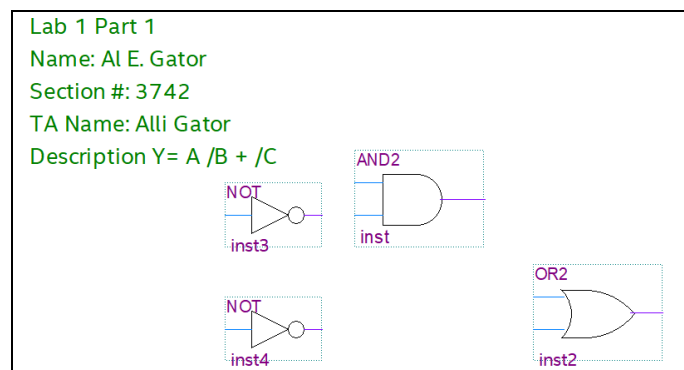

Figure 2: Bdf with logic symbols

D. Adding/Deleting Wires

1. Save your design. You are now ready "wire up" your circuit. It is a good idea to save your design often, just in

case something bad happens (like a Windows crash or a power outage).

2. Place your pointer on the output of one of the Level-Shifters and hold the left mouse button down. You should see a cross-hairs or "+" appear at the output.
3. Drag your pointer to the input of the AND gate. Every time you release the mouse key, the line (wire) ends. If your wire did not reach the AND 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 right click on the wire and select "Delete" or left click on the wire (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 drag and stay connected to the component. This is referred to as "rubber banding" and is a feature of all major schematic entry design packages.
6. Add the rest of the wires needed to connect the logic diagram. Add small input lines where the three inputs will be placed and an output line where the output will be placed. Your bdf should look similar to Figure 3.
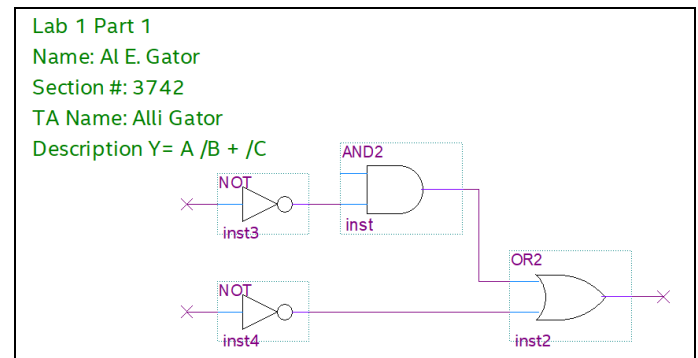

Figure 3: Bdf with wires

E. Adding Input & Output Ports

1. In the same manner that you placed a gate onto the bdf, add three input pins from the "Symbol" libraries. Input pins can be found under "primitives | pin | inputs" (or just type **input** just as you typed **not** previously in the "Name:" box). I suggest placing these inputs together, above your logic diagram and just to the right of your name as shown in Figure 4.
2. Double click on the first input pin name (on the left of the input port symbol) and change it to 'A'. Repeat these two steps to create input ports 'B' & 'C'.
3. In the same manner and in the same library that you found the input pins, add an output pin from the "Symbol" library. I put this output pin under and to the right of the input pins. Change the pin name to 'Y' on the output port, as shown in Figure 4.
4. Now select the top wire near the left most point where you would like to connect signal A. The wire should change colors. Type "A." An "A" should appear near the point you selected.

University of Florida   EEL 3701   Dr. Eric M. Schwartz
Department of Electrical & Computer Engineering   Revision **0**   26-Aug-19
Page 3/9   **Quartus Tutorial with Basic Graphical Gate Entry and Simulation**
(Last verified for Quartus Prime Lite Edition 18.1)

5. Do the same to place "B", "C" and "Y" at the appropriate points.
6. 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 bdf should look similar to Figure 4.
7. Save your design. You are now ready to proceed to compile and simulate the circuit.
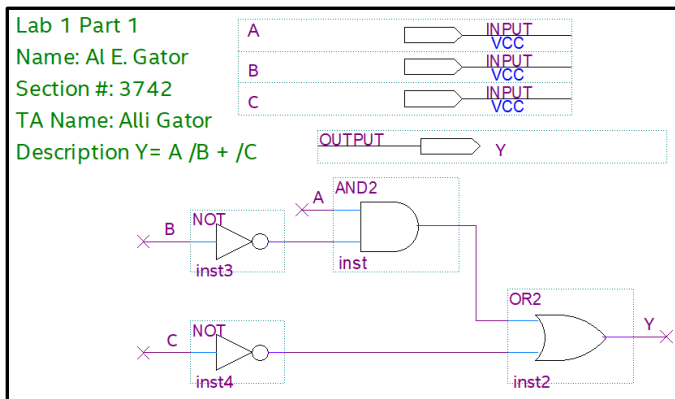
Figure 4: Bdf with inputs and outputs

F. Compiling
0. For "Functional Compiling," see the next section.
1. To compile your design, click on the blue isosceles triangle button (▷),or double click on "Compile Design" in the task utility window, or select "Processing" and then "Start Compilation". You will be asked to save the bdf file because has been modified. Select "Yes." After the file is compiled (which depending on your computer, may take 30 seconds to two minutes), your task utility window should be similar to the image in Figure 5.
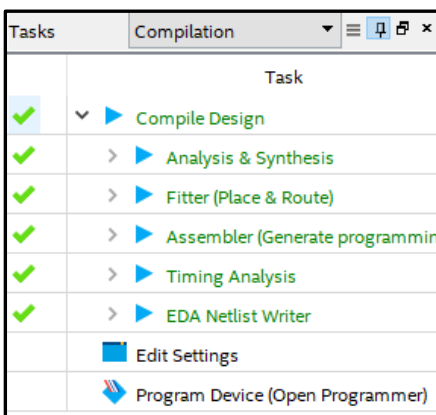
Figure 5: Compile Complete

2. Notice that the Messages Utility Bar on the bottom of the Quartus window says "Quartus Prime Full Compilation was successful. 0 errors, 13 warnings". You may have a different number of warnings. You can ignore most warnings. If you did something wrong, Quartus will not compile and will give you an error.

Note: Several common errors involve having floating (unconnected) inputs and outputs or short circuits (possible locations where two inputs or two outputs are connected). Another error is when the top-level entity is undefined; if this occurs click on "Files" on the Project Navigator utility window, right click your bdf (in this case lab1_ex.bdf), and select "Settings", and then select that file for "Top-level entity."

Note: When using Quartus schematic entry (bdf) files as your circuit diagram for constructing circuits on your breadboard, always label the parts and pin numbers of the chip. If there are multiples of the same part needed, e.g., if you need five 2-input AND gates when the 74'08 only has four per chip, then label the two 74'08's differently, i.e., 08$_A$ and 08$_B$. Each AND gate on a single 74'08 can be labelled 08-1, 08-2, 08-3, and 08-4.

G. Functional Compiling
To functionally compile your design takes **significantly less time** than the full compilation described above. When I just tried it for this circuit, it took 13 seconds, whereas the full compilation took 41 seconds.
1. To functionally compile your design, double click the "Analysis & Synthesis" in the Compilation window shown in Figure 6. Figure 6 shows the result of the Functional Compiling.
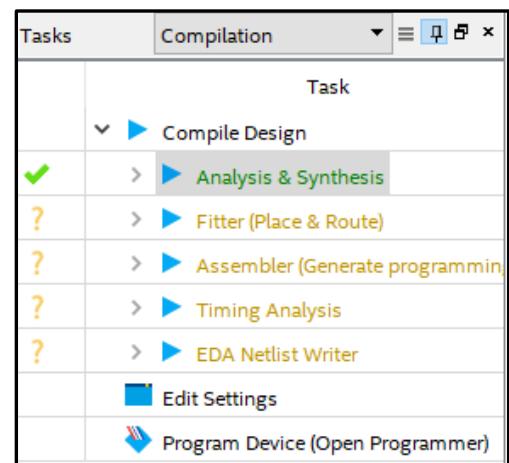2. Note that if perform a Functional Compiling, when you simulate, you can **ONLY** run a Functional Simulation.

Figure 6: Functional Compile Complete

III. Simulating

A. Creating a VWF (Vector Waveform)
1. Select "File" and "New". Under "Verification/Debugging Files" select "University Program VWF".
2. Save this file under the suggested (default) name, "Waveform.vwf" by selecting "File | Save As".

B. Adding signals
1. In the Waveform.vwf window, in the left side of the window (under "Name") double click with the left mouse button or click the right mouse button. The "Insert Node or Bus" window will appear. Select "Node Finder". Under

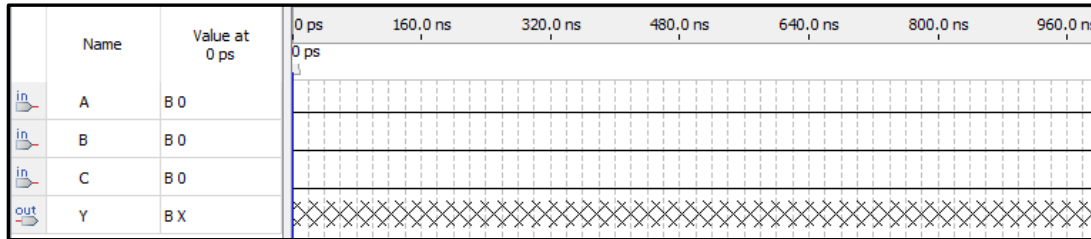(Last verified for Quartus Prime Lite Edition 18.1)

Figure 7: Vwf with inputs and outputs

"Filter" select "Pins: all" then select "List." Hit the >> button to copy all the nodes (inputs and outputs) to the "Selected Nodes" list on the right. Select "OK" and then "OK".

2. You should now see the inputs and outputs in the vector waveform file window. Save this file. The window should look like Figure 7.

Special Note: The time scale is shown above is in increments of 10 ns (ns = nano seconds). This is too small for our parts. Our parts have a propagation delay of between 10 and 20 ns, i.e., the output of the gates does not change until approximately 10-20 ns after the inputs change.

C. Changing Grid Size and End Time
1. Change the default grid size to 25 ns (or more) by the following. Go to the "Edit" menu and select "Grid Size". Then change the Time period to 25.0 ns.
2. Go to the "Edit" menu and select "Set End Time". Change the default time from 1.0 us (1 μs = 1000 ns) to 1.2 us (1200 ns). **NOTE:** When you increase the end time, Quartus simulation will pick signal values to fill up the total time. Be sure to review your inputs to verify that the simulation is doing what you want. **NOTE:** To avoid possible simulation issues, **ALWAYS INCREASE END TIME**. You may give it a new end time once you simulate at least once, but you should always increase the end time initially.

D. Manually Changing VWF
1. Use the magnifying glass to zoom in on the waveform window until you can see 50 ns intervals as seen in the image of Figure 8.
2. Using your mouse (make sure the pointer is selected), click and drag your mouse cursor across 50 ns to 200 ns on input A to select the area. Now select the "1" button on the toolbar (or select "Edit", "Value", "Forcing High (1)") to set this signal to High during this time window.
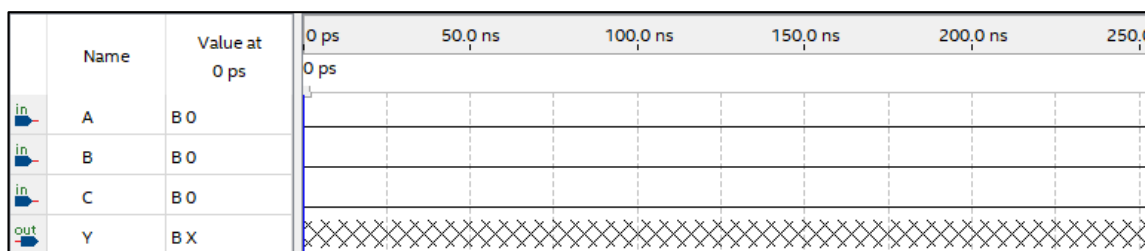
3. Using the same methods in the previous step, set the time period of 50 ns to 100 ns to Low by using the "0" button.

Special Note: Your simulation will match your **voltage** table (**not** your **truth** table). When you compare the outputs, you should verify it with the outputs of your **voltage** table and not your **truth** table. When you see a 0, it is **LOW**. When you see a 1, it is **HIGH**. Please do not get this confused.

4. Manually manipulate your signals to match the image in Figure 9. Save this simulation design.
5. The inputs have now been defined and "count" or increment through the binary numbers 000 to 111 (ABC where A is the most significant bit and C is the least significant bit). We can now run the design simulation at this point. Note: The Y output is comprised of 'XXX' in the waveform editor to show that the output is presently undefined.

Note: There are better ways to input data then to enter each of the values you want by changing default inputs. See Section F.
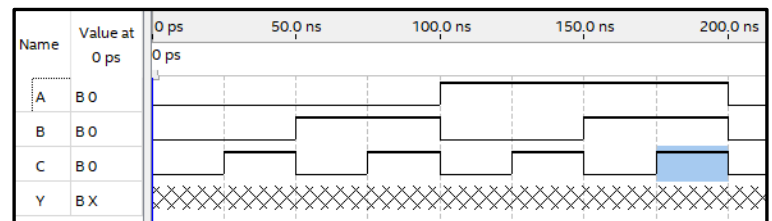

Figure 9: Vwf with manual inputs

E. Functional and Timing Simulation
0. **Unfortunately, Quartus Prime Lite 18.1 does not support timing simulations for the MAX 10 devices. This may change in the future.**
1. Select "Simulation" in waveform window and "Run Functional Simulation". Quartus will ask you to save the file first. Do it! You'll notice a Simulation Flow Progress

Figure 8: Vwf window with 25 ns grid size

University of Florida   EEL 3701   Dr. Eric M. Schwartz
Department of Electrical & Computer Engineering   Revision 0   26-Aug-19
Page 5/9   **Quartus Tutorial with Basic Graphical Gate Entry and Simulation**
(Last verified for Quartus Prime Lite Edition 18.1)

window pop up then a new simulation window will open. It will look similar to your vwf except that it is read only. You cannot modify signals on this window.

2. Zoom in on the window. You should see something similar to Figure 10.
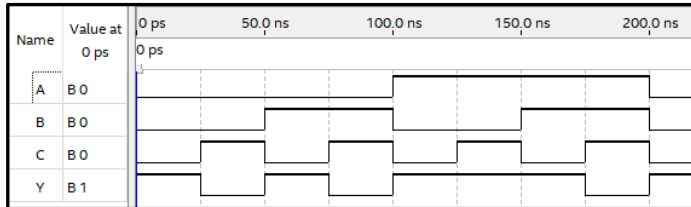
Figure 10: Output of Functional Simulation

3. If you haven't created a logic and voltage table for the equation you entered under Quartus, do so now and compare these results with those obtained from simulation. Because the signals are assumed to be active high, the truth and voltage tables will look identical (with 0 and 1 replaced with L and H, respectively). Your simulation results from Quartus should match your voltage table.
   - For active-low signals, I suggest that you name your signals with a suffix of **_L**. For active-high signals, I suggest no suffix (or you could use **_H**). For example, if X is active-low and Z was active-high, use signal names **X_L** and **Z**.
   - When submitting screenshots for lab documents, take a screenshot of this window with the **simulated** output. Use a tool like snip tool or paint to annotate the simulation. Use arrows and text to describe what is occurring and demonstrate that you received the proper results.

4. **Unfortunately, Quartus Prime Lite 18.1 does not support timing simulations for the MAX 10 devices. This may change in the future.**

5. Close the simulation window. Re-compile the bdf design file, using a full compilation (not a functional compilation). Then select "Simulation" and "Run Timing Simulation". You'll again notice a Simulation Flow Progress window pop up and a new simulation window will pop up.

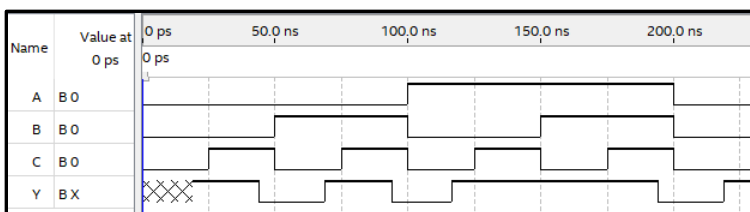6. Zoom in on the window. You should see something similar to Figure 11.

Figure 11: Output of Timing Simulation

6. When comparing the results of the logic table with that of the simulator, it should be apparent that they match but that there is a small delay between when the inputs change and when the output changes to the expected value. For example, if we look at the time segment from 25 to 50 ns, we see that the inputs C change at 25 ns, but that Y does

not change immediately. This delay is called the propagation delay of the device. This slight delay is due to the physical gate delay of the gates in the programmable logic device (PLD) required to implement the circuit. In other words, every gate in your circuit has a chunk of PLD hardware that is associated with it and an associated physical delay.

F. Grouping Signals and Using Count Value and Clock Value

1. You can group inputs by selecting several and then click the right mouse button, select "Grouping" and then select "Group." Group A, B, and C and name them "Inputs". Select the radix (base you want to use, i.e., binary, hexadecimal, or octal). In this case you should stick to the default: binary. Press "Ok".

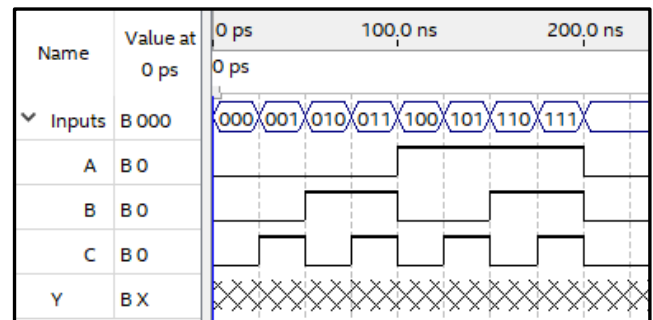2. Click on the arrow next to Inputs to expand your group. You should see something similar to Figure 12.

Figure 12: Grouping on vwf

3. Click "Inputs" and press the "0" button to make all the inputs low.

4. Click "Inputs" and press the C button in the toolbar (or Select "Edit", "Value", "Count Value …"). This will allow you to count up from a start value (in this case from 000 to 111). Leave the default "Start value" at 000 and the default "Increment by" at 1. At "Count occur", change it to 50 ns. Press Ok.

5. Zoom out on the window. Notice that "Inputs" automatically counts for the entire simulation. Press Save.

6. Perform a functional and timing simulation.

7. You can also use Clock value to generate a clock signal. The button looks like a stopwatch. You will use this later during this semester. You can also change the period, offset, and duty cycle.

IV. Programming

A. Using the Out of the Box PLD Programmer
View the instructions on how to use the Out of the Box PLD Programmer. This programmer replaces the alternative part that costs anywhere from $50 to $300 (from Terasic or Altera, respectively). Programming with this PLD programmer is slower than with the Terasic and Altera devices, and it does limit the types of devices that it can program.

The OOTB PLD Programmer instructions has a link to a required FTDI driver for USB on Windows computers (or those setup to have Windows virtual machines). A link is also

provided for those who want to try to run this with other operating systems (including Linux). Download, unzip, and then execute the file to install the driver. See our website and either the Software/Docs page or the Labs page for the driver.

In addition, a DLL file is provided on our website and either the Software/Docs page or the Labs page. As the instructions tell you, this file must be copied and placed in the appropriate folder on your computer.

B. Assigning pins using Pin Planner

1. To make sure you do **NOT** accidently assign pins to the wrong location, use only the pins shown Figures 13 and 14. Do **NOT** use the clock pins for general purpose inputs or outputs, e.g., C2P (i.e., CLK2P, MAX 10 pin G9) and C3P (i.e., CLK3P, MAX10 pin F13); use these pins only for clock inputs.

Pin as Chip #	Available FPGA Pins	Available FPGA Pins	Pin as Chip #
1	C2P (CLK2P) G9	C2P (CLK2P) G9	1
2	F12	C3P (CLK3P) F13	40
3	E3	D11	39
4	E4	E3	3
5	F4	E4	4
6	G4	F4	5
7	J7	F12	2
8	H3	G4	6
9	H2	G13	29
10	K5	H2	9
11	J2	H3	8
12	K2	H8	31
13	L2	H13	28
14	J5	J2	11
15	L5	J5	14
16	L4	J6	17
17	J6	J7	7
18	K6	J8	32
19	GND	J13	27
20	GND	K2	12
21	3.3 V	K5	10
22	3.3 V	K6	18
23	N10	K7	30
24	N12	K8	33
25	M13	K12	34
26	L13	L2	13
27	J13	L4	16
28	H13	L5	15
29	G13	L13	26
30	K7	M9	35
31	H8	M10	36
32	J8	M11	37
33	K8	M12	38
34	K12	M13	25
35	M9	N10	23
36	M10	N12	24
37	M11	GND	19
38	M12	GND	20
39	D11	3.3 V	22
40	C3P (CLK3P) F13	3.3 V	21

Figure 13: PLD PCB top layer with available FPGA pins

Figure 14: Available FPGA pins

University of Florida   EEL 3701   Dr. Eric M. Schwartz
Department of Electrical & Computer Engineering   Revision 0   26-Aug-19
Page 7/9   **Quartus Tutorial with Basic Graphical Gate Entry and Simulation**
(Last verified for Quartus Prime Lite Edition 18.1)

2. To program a device you need to give you device pins. Select "Assignments" and "Pin Planner". Under location, type in a pin number for A, B, C, and Y (in this case type in E3, E4, F4, and G4, respectively). Exit out of pin planner.

3.

The result is that your bdf will now change to look something like Figure 16. Note that you can move the pin numbers around in the bdf.

Figure 15: Pin Planner

Figure 16: Bdf with pin assignments

University of Florida   EEL 3701   Dr. Eric M. Schwartz

Department of Electrical & Computer Engineering   Revision 0   26-Aug-19
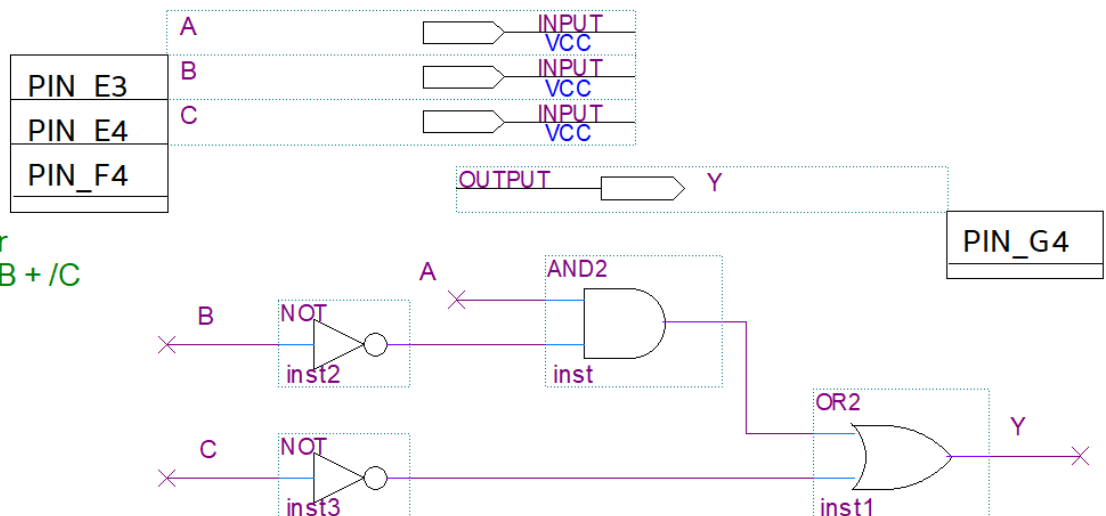
Page 8/9   **Quartus Tutorial with Basic Graphical Gate Entry and Simulation**
(Last verified for Quartus Prime Lite Edition 18.1)

4. See the ***OOTB 7-segment display*** document (on our website) for information about the MAX 10 pins used for the two 7-segment displays. Figure 17 shows the correpsonding MAX 10 pin assignment for the two 7-segment displays.

7-Seg Segs	MAX 10 Pins	7-Seg Segs	MAX 10 Pins
U7-A	D9	U8-A	H10
U7-B	C9	U8-B	F8
U7-C	G12	U8-C	E8
U7-D	L10	U8-D	J12
U7-E	K10	U8-E	H9
U7-F	J9	U8-F	J10
U7-G	E6	U8-G	L11
U7-DP	E12	U8-DP	E9

**Figure 17:** 7-segment display inputs with corresponding ***MAX 10*** pins.

C. Tri-stating unused pins
1. In the default configuration, Quartus will program unused pins as outputs. The functions of these pins can be destroyed if they are connected as outputs. To prevent this from occurring, unused pins should be set as tri-stated.
2. In the "Assignment" menu select "Device"
3. Select "Device and Pin Options…"
4. Select the "Unused pins" tab and then select "As input tri-stated" Press "Ok". Press "Ok".

This process is necessary (and should be done) for **EVERY** design that will be downloaded to your PLD (CPLD or FPGA). In order for the above to take effect, you must recompile the design and then program the PLD.

D. Programming
1. Make sure the driver is installed. Go to this website if you have not installed the driver.
2. Unless your breadboard is completely empty of wires and components, remove your PLD board from your breadboard. This will prevent possible shorts.
3. Using your USB-B cable, connect your board to your computer. A green light should turn on.
4. Connect your USB-Blaster to your computer and to your PLD board.
5. On Quartus, select "Tools" and "Programmer".
6. Select "Hardware Setup" and then select (double-click on) USB-Blaster. (See Figure 17.) If it does not show up make sure your USB-Baster is plugged into your computer and your driver is installed. Then press "Close". If you are on a mac using a virtual machine, make sure your USB-blaster is connected to your virtual machine. Look in your virtual machine settings for something relating to USB Connection Preferences.
7. Make sure your mode is "JTAG".
8. Select "Add File" and go to the "output_files" directory. Select lab1_ex.pof. Press "Open".
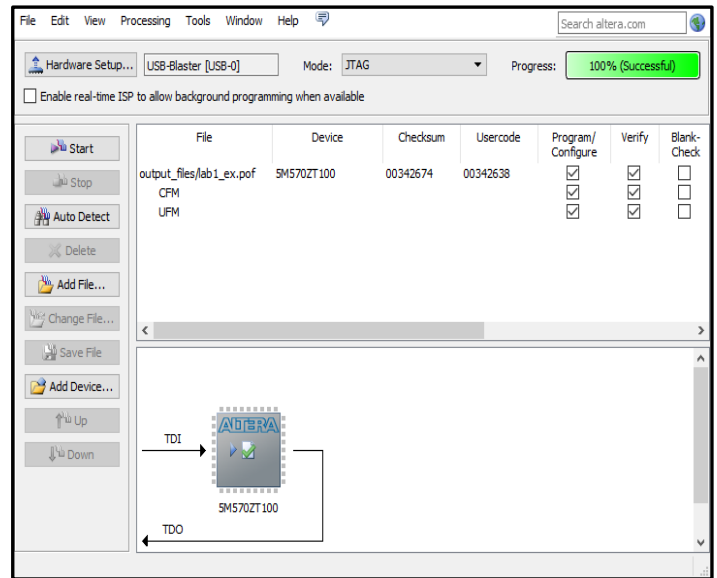
Figure 17: Programmer

9. Click the boxes under Program/Configure and Verify.
10. Press "Start". Progress should read "100% Successful." The 2 orange lights should now turn on. If under Progress it says "Failed", make sure your USB-Blaster is not plugged in backwards and make sure it is plugged into all the pins. If that is not the issue, check your soldering.
11. Remove the USB-Blaster and USB-B cable. Insert your PLD into your breadboard. Do not plug in power until you wire it up.
12. To avoid having to repeat step 8 every time you program, you can save this configuration. Press "File" and "Save As". Name your file "lab1_ex".

Note: When you program your device with a pof file and then remove power from the PLD, the design on the PLD is remembered, i.e., it is non-volatile and will be restored when power is restored. If you program your device with a sof file, when power is removed, the design on the PLD will be forgotten.

E. Exporting Pin Assignments
1. Sometimes when you make multiple projects, you want to use the same pin assignments so you don't need to rewire the board. As long as you give the input and output pins the same names, you can easily do this by exporting the pin assignments from one project as a ".qsf" file and importing it to another project.
2. To export assignments, select "Assignments" and "Export Assignments". You may change the location you wish to save your qsf but do not change the name.
3. To import assignments, select "Assignments" and "Import Assignments. Click the "…" and locate the ".qsf" file. Press "Open". Press "Ok". If you have any additional pins that was not in your original file, go to pin planner and give those pins location. If there are pins that are not in the new project, they will be ignored. Recompile.

University of Florida
Department of Electrical & Computer Engineering
Page 9/9

EEL 3701
Revision **0**

Dr. Eric M. Schwartz
26-Aug-19

F. Deleting Pin Assignments

To delete Pin Assignments, select "Assignments" and "Remove Assignments". Check "Pin, Location & Routing Assignments" and Press "OK". Recompile.

V. Archiving Your Project

A. Archiving your project into a qar file

1. Archiving a project will save the relevant project files into a single compressed file. In the "Project" menu, select "Archive Project..."
2. Change the archive name, if necessary, and select "Archive." The archive file will have the file name extension qar, e.g., the file name might be "Lab6**A**.qar."
3. I have found that when un-archiving (restoring) an archived project, I need to do the following to correct folder path information.
   a. Specify a "Destination folder:", for example, t c:/3701/LabX, where X is the Lab number.
   b. In Quartus, open the vwf file. This will open the MultiSim Simulator Waveform Editor.
   c. Select Simulation, then Simulation Settings, then the Restore Defaults button on the bottom of this screen, and then select Save. This will fix the path information for the destination you used for the project files.

VI. Miscellaneous

A "wire" component in Quartus will allow you to connect an input to directly to an output. You can think of this as a Level Shifter with no bubbles, i.e., it does **not** change the activation level.
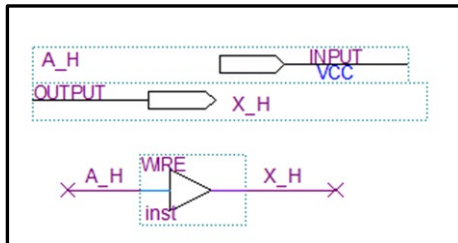

Figure 18: Wire component