Quartus Tutorial with Basic Graphical Gate Entry and Simulation

(Last verified for Quartus II Web Edition v9.1)

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 for the inputs & output and compare it with the simulation results.

I. Circuit Design Tutorial (using the Schematic Editor)

A. ***New Project Design Creation***

1. Setup a 'lab1_ex' directory on your PC to hold your design & simulation files.
2. Launch the Altera Quartus software under Windows.
3. Select the pull-down button "File" and "New Project Wizard ...". Select "Next". Now use the "…" button to browse and select the directory you created in step number 1. 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 "Finish".
4. Select the pull-down button "File" and "New | Device Design Files | Block Diagram/Schematic File". Press OK. This should open a palette where you will design your circuit. This palette is designated "Block1.bdf".
5. Select the pull-down button "file" and save the graphic design file "lab1_ex" in your 'lab1_ex' project directory created in step #1 (by selecting "Save As"). The file will be given the bdf extension; bdf stands for "block design file" and contains schematics, symbols or block diagrams.

Special Note: You can remove the grid dots (or add them back) by selecting the pull-down button "View" and then select "Show Guidelines."

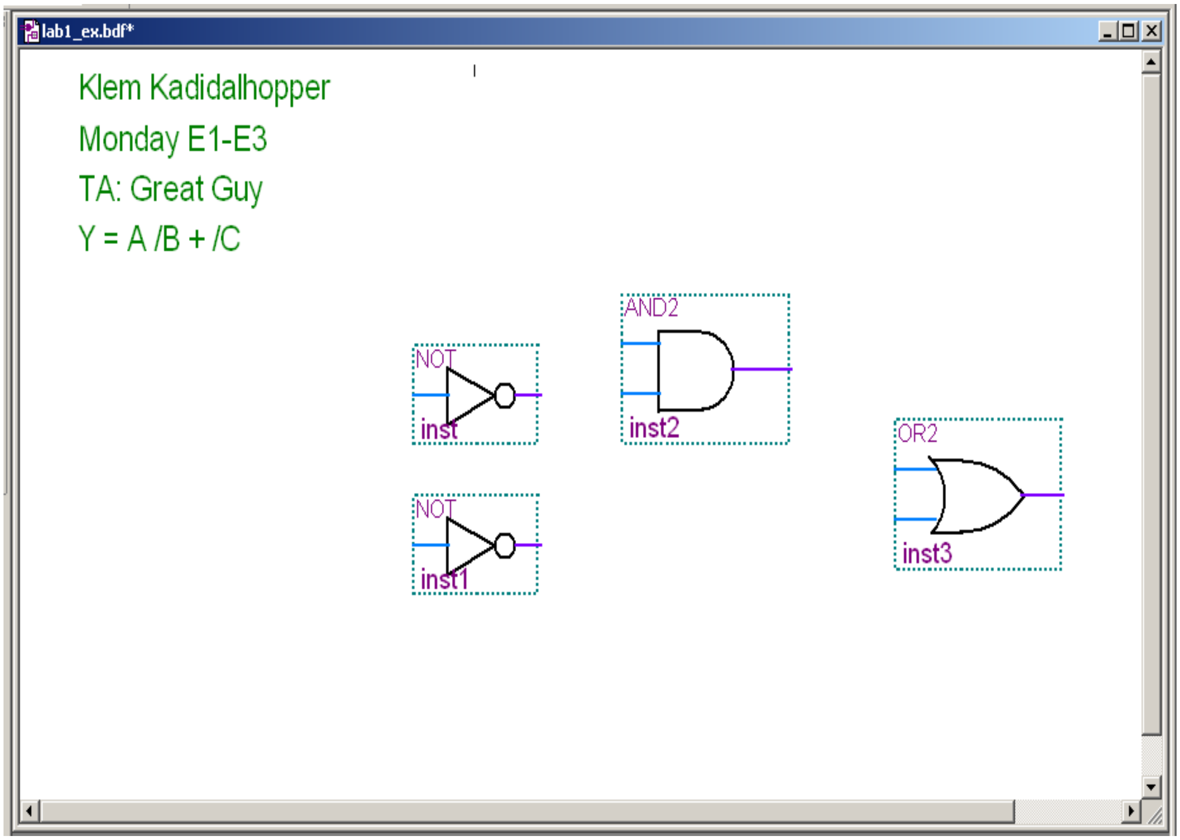
B. ***Adding Text***

1. Select the "A" below the arrow to the left of your Block Diagram/Schematic File window (also known as the palette).
2. Select a point near the top left in the window with the left mouse key. Type your name and then hit the "Enter" key.
3. Type your lab day and periods and then hit the "Enter" key.
4. Type **TA: Name**, where **Name** is replaced by your TA's first and last name. Hit the "Enter" key.
5. Type the following equation, **Y = A * /B + /C** , and then hit the "Enter" key.
6. Hit the "Esc" (escape) key to end text additions.

C. ***Component Selection Process and Moving Components***

1. With your mouse pointing inside the palette, double-click (with the left mouse key) [or click the right mouse key and select "Insert | Symbol"]. The "Symbol" dialog box will appear. This window lists the available Altera libraries.
2. Select the + icon to expand the "/altera/80sp1/quartus1/libraries" folder.
3. Select the + icon to expand the "primitives" folder and then expand the "logic" folder.
4. In the logic folder, select the "and2" component by double clicking on it (or by selecting it with a single click, then selecting "OK"). (If you wanted to add multiple 2-input AND gates, you could select the "Repeat-insert mode" box.)
5. Click the pointer at the desired location in the Block Diagram/Schematic Editor window to insert the AND symbol into the design file.
6. Repeat steps 1-5 to enter an OR (or2) gate and two Level-Shifter (not) gates.

7. Select the magnifying glass to the left of your Block Diagram/Schematic Editor 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.
8. Hit the "Esc" (escape) key to end magnifying options or select the arrow symbol to return to selection mode.
9. 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, holding down the left button and moving it to another location on the palette. The window should look something like this.

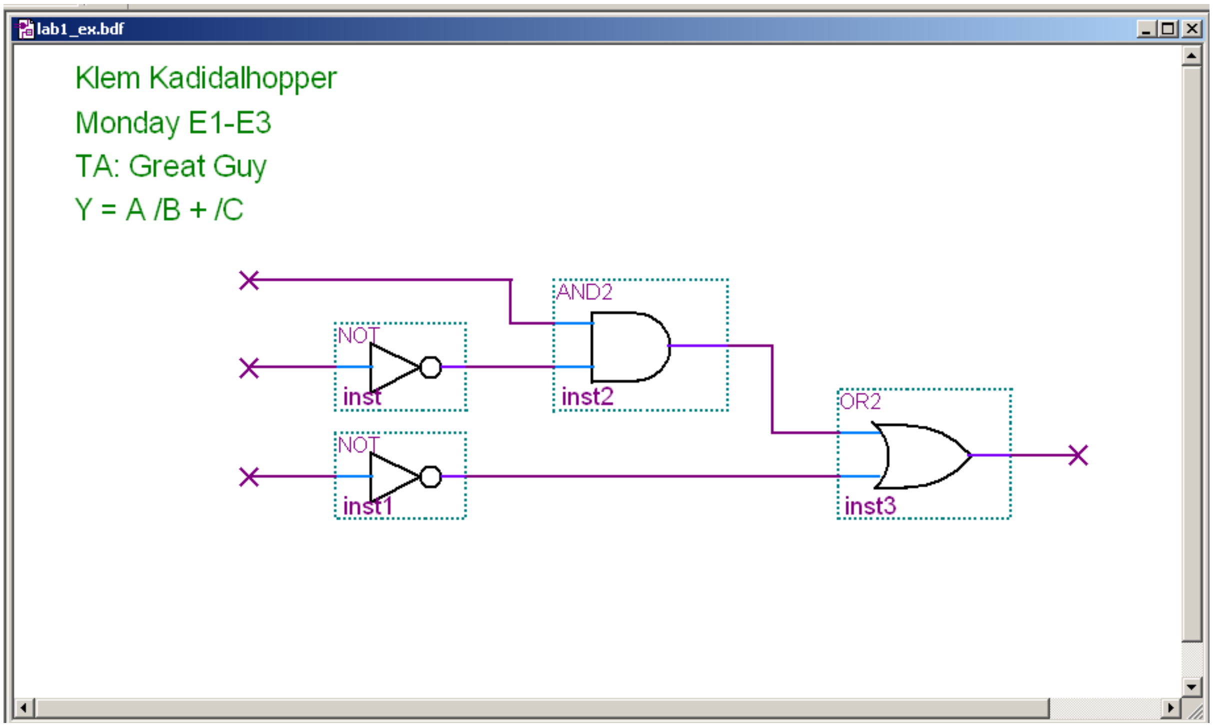


Klem Kadidalhopper
Monday E1-E3
TA: Great Guy
Y = A /B + /C

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, a power outage or a nasty roommate).
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 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 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. The window should look something like this.



Klem Kadidalhopper

Monday E1-E3

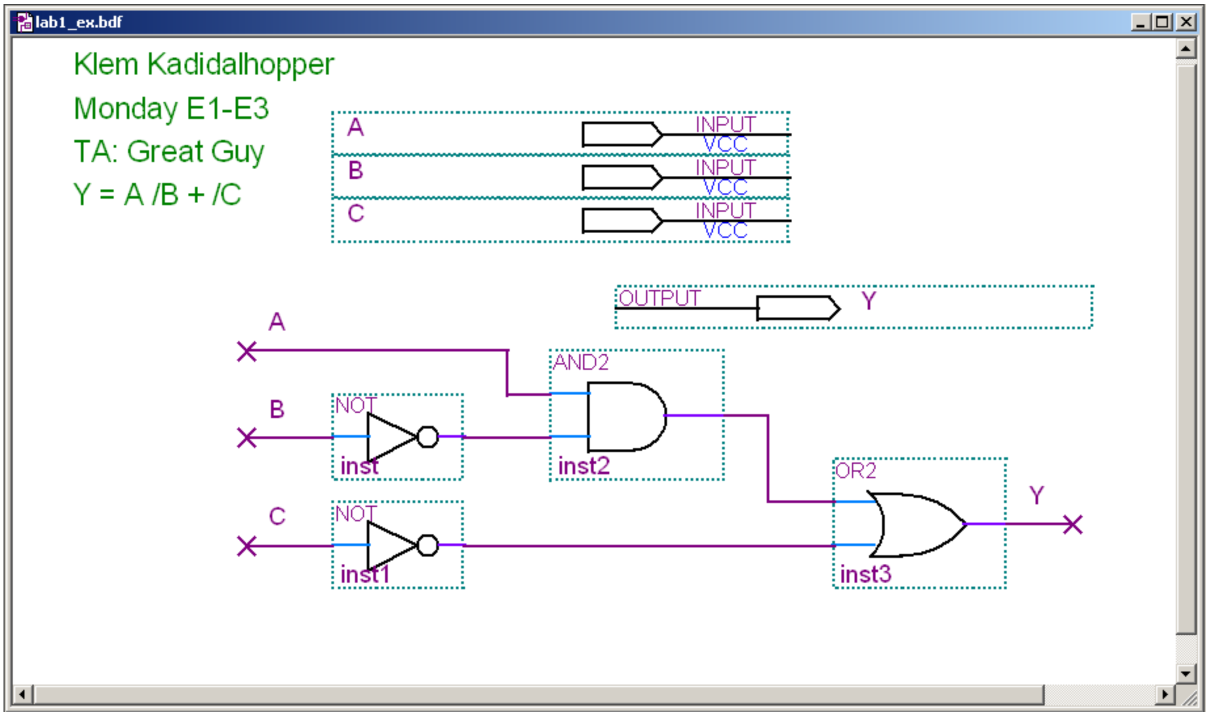
TA: Great Guy

Y = A /B + /C

E. Adding Input & Output Ports

1. In the same manner that you placed a gate onto the palette, add three input pins from the "Symbol" libraries. Input pins can be found under "primitives | pin | inputs". I would place these inputs together, above your logic diagram and just to the right of your name.
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.
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.
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 window should look something like the below figure.
7. Save your design. You are now ready to proceed to simulation of the circuit.
8. To print, go to "File | Print…". If you want to change what appears on the printout or how it appears, go to "File | Page Setup…" change print settings. Before printing, you can view what the print will look like by selecting "File | Print Preview".

Note: You could connect the input and output pins directly to your circuit, but I find keeping the pins separate from the circuit diagram has advantages that will be discussed later.



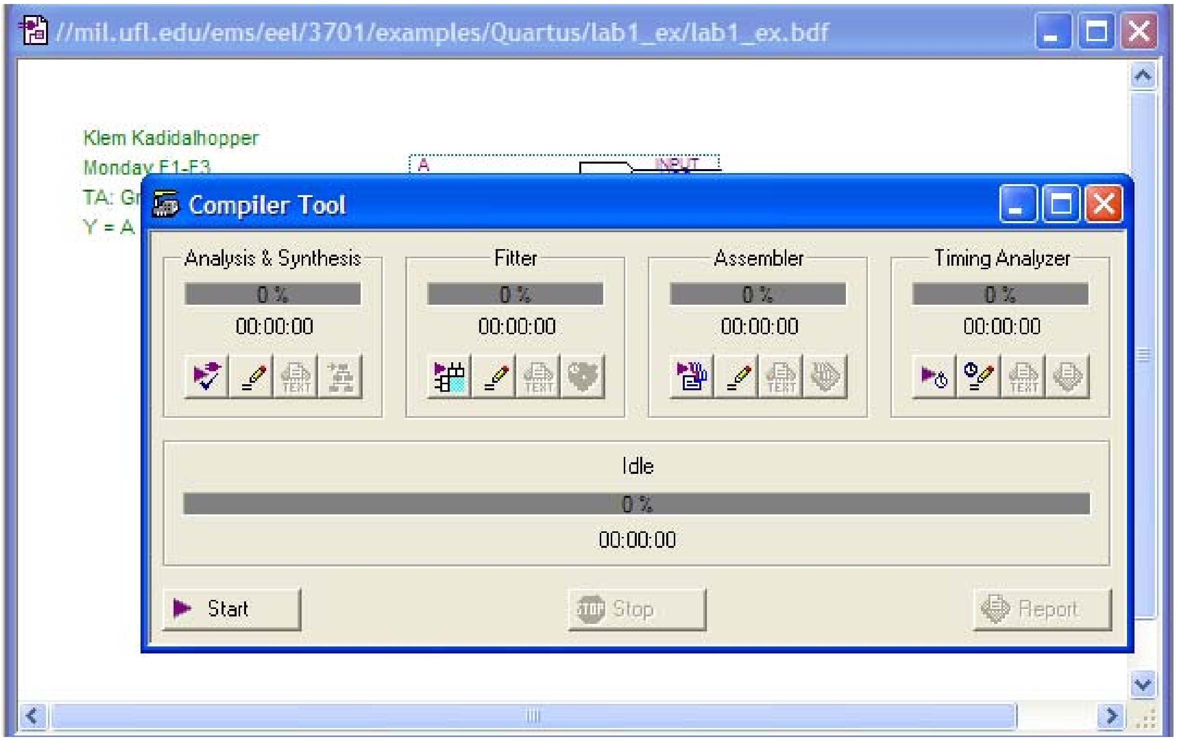
II. Circuit Compilation and Simulation

A. Fitting the Design to a Component and Compiling

1. Before we can simulate the design, we first must run the fitter software to place the design into a programmable component.
   a. Select the "Tools" pull-down menu (or right click on the upper toolbar). Select "Customize | Toolbars." Put a check mark by the "Applications" box (unless one is already there) and then select "Close."
   b. Place your mouse over the icon on the top bar that has a small smokestack.

   When you put your mouse over the small smokestack, on the bottom bar of the window you should see an icon help message saying "Opens the Compiler Tool window or brings it to the foreground."
2. Press the smokestack icon (compiler) and the following compiler window should open up. (An alternative to this is to select "Processing | Compiler Tool".) See the below figure for an example of what your screen might look like at this point.
3. Select "Assignments | Device" from the pull-down menu. This will open a new screen. Select MAX7000S from the "Family" pull-down list.
4. Select the "Specific device selected…" and then choose EPM7064SLC44-10, which is the device we will be using starting with lab 4 or 5. Select "OK."
5. Press the "Start" button in the compiler window. If the complete compilation process is successful, you should see a message back from the compiler that ends with says "Info: Quartus II Full Compilation was successful. 0 errors, 0 warnings." You can ignore most warnings, like "Warning: Timing Analysis does not support the analysis of latches as synchronous elements for the currently selected device family."

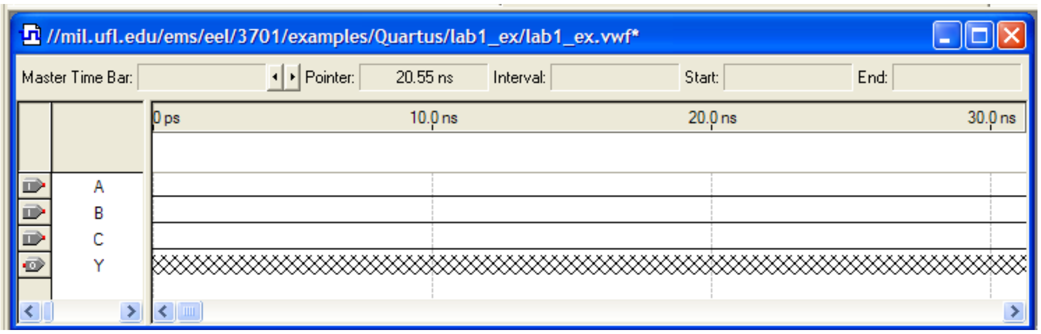
6. To view information about the compilation, select "Processing | Compilation Report" (or Ctrl-R). You can also select the icon with a blue diamond with a paper on it.



Special 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.

B. Creating a Waveform for use with the Simulator

1. Once the circuit has been fitted into a particular programmable logic device, you can run the simulator to check the circuit functionality. Place your mouse over the icon that looks like a blue diamond with a rectangular waveform on it. You should see an icon message on the bottom of the Quartus window that says "Creates a new file in the Waveform Editor". Select this icon. (You could also go to "File | New | Other Files | Vector Waveform File" to open up the same waveform window.)
2. Now you will stimulate the circuit with a vector of inputs so that you can verify that the outputs match those in your **voltage** table (**not** your **truth** table). (Note: The comparison of your simulation should be with your **voltage** table and not your **truth** table.)
3. Save this file: "File | Save". The file should be saved under the name lab1_ex with the extension .vwf for "Vector Waveform File..
4. In the lab1_ex.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 "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. Select OK again.
5. You should now see the inputs and outputs in the vector waveform file window. Save this file. The window should look something like below.

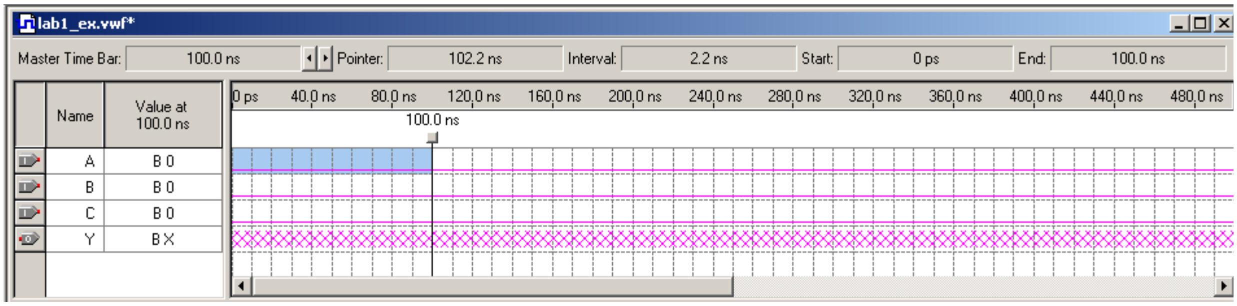


//mil.ufl.edu/ems/eel/3701/examples/Quartus/lab1_ex/lab1_ex.vwf*

Master Time Bar: | Pointer: 20.55 ns | Interval: | Start: | End:

Special Note: The time scale is shown above is in increments of 10ns (ns = nano seconds). This is too small for our parts. Our parts have a propagation delay of 10ns, i.e., the output does not change until 10ns after the inputs change. If you use the magnifying glass to change the time scale, you will find that the total time for the simulation is probably only 100ns. We need to increase this to about 1000ns=1us (us = µs = micro second) for this problem.

6. Change the default grid size (which is 10ns by default) to 25ns by the following. Go to the "Edit" menu and select "Grid Size". Then change the Time period to 25.0ns.
7. Go to the "Edit" menu and select "End Time". Then change the "Time" to 1000ns (or 1us). The "us" in Quartus represents "µs" or 10^{-6} seconds. Now use the magnifying glass to zoom out on the waveform window.
8. Using your mouse, click and drag your mouse cursor across the 100 ns segment to select the area that is shown as a darkened area in the below screen snapshot. You should see a darkened area as result of this operation. Now press the left toolbar button that has a "1" in it. Now try pressing the left toolbar button that has a "0" in it. This is how you set the highlighted input segment to a particular value. Set the time segment from 50 ns to 100 ns of C to '1' and point & click your mouse to a new area on the window to make the change permanent.

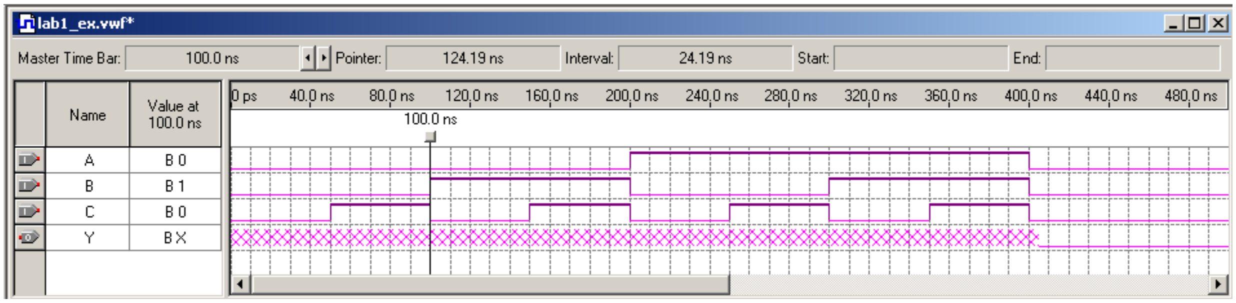


lab1_ex.vwf*

Master Time Bar: 100.0 ns | Pointer: 102.2 ns | Interval: 2.2 ns | Start: 0 ps | End: 100.0 ns

Name	Value at 100.0 ns
A	B 0
B	B 0
C	B 0
Y	B X

9. Use the magnifying glass again to increase the segments view to about 500 ns and modify the inputs so that they look like the patterns shown below. Save the file.



lab1_ex.vwf*

Master Time Bar: 100.0 ns | Pointer: 124.19 ns | Interval: 24.19 ns | Start: | End:

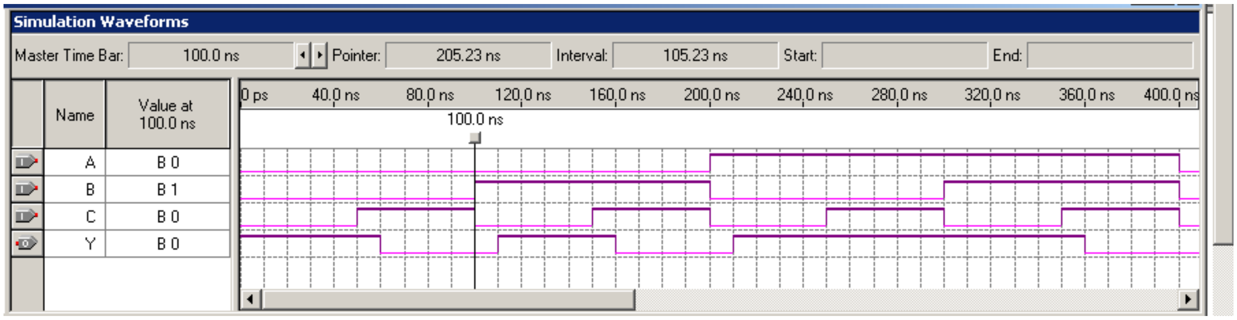
Name	Value at 100.0 ns
A	B 0
B	B 1
C	B 0
Y	B X

10. 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. The reason that it is undefined is that although the inputs have been defined, the simulation has not been run yet.

Special Note: The above pattern looks like the values that you would have in a truth table (but is actually represents a voltage table since the 0 and 1 in the Quartus simulations represent L and H, respectively). You can generate this pattern more easily by grouping the inputs and then giving the group a count input sequence. **This technique will be described in class.**

C. Running the Simulator and Analyzing the Results

1. Select “Processing | Simulator Tool”. Alternatively, you can look for a computer with a waveform in it; this symbol is usually next to the smokestack for the “Compiler Tool”. Make sure that the Simulation mode is set to “Timing”. Check “Overwrite simulation input file with simulation results”. (I also suggest that you check “Glich detection” and use time of 1.0ns.) Select “Start”. When the simulation is complete, you will get a message “Simulator was successful.” Select “OK” and then select “Open” on the “Simulator Tool” window. Select “Yes” when prompted about “… overwriting portions of the file …”.
2. Resize the “Simulation Waveforms” as desired. It should look something like this.



Simulation Waveforms

Master Time Bar: 100.0 ns Pointer: 205.23 ns Interval: 105.23 ns Start: End:

Name	Value at 100.0 ns
A	B 0
B	B 1
C	B 0
Y	B 0

3. To print it, use “File | Print...” or Ctrl-P to print. To print a selected portion of the simulation, select “Options” and then specify both a “From” and “To” time. See part *E* of this section for more information.
4. If you haven’t created a logic table for the equation you entered under Quartus, do so now and compare these results with those obtained from simulation. The table that actually is needed is a **voltage** table, but since all the signals were assumed active-high for this example, they will look identical (with 0 and 1 replaced with L and H, respectively). Your simulation results from Quartus should match your logic table.
5. 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 40 to 80 ns, we see that the inputs C change at 50 ns, but that Y does not change until 60 ns. This 10 ns 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. With this new knowledge, what do you think is the worst delay path in your circuit?

D. Printing the Simulation Results

To print a waveform file, select “File | Print.” If you want to print only a portion of the waveform, select “File | Print | Options” and select the “Time range” as desired.

A poor alternative to the above is to capture a portion of the screen and to then paste this captured portion into a word processing or drawing application. To copy a simulation output (or any window on the screen), make it as big as possible and then (while it is the active [selected]) window), hit “Alt-Print Scrn” (i.e., hold down the “Alt” and while still holding it, press the “Print Scrn” key). Then paste the captured window into your favorite word-processing or drawing program. In your chosen program you can crop and enlarge you figure as desired.

E. Functional Compilation and Simulation

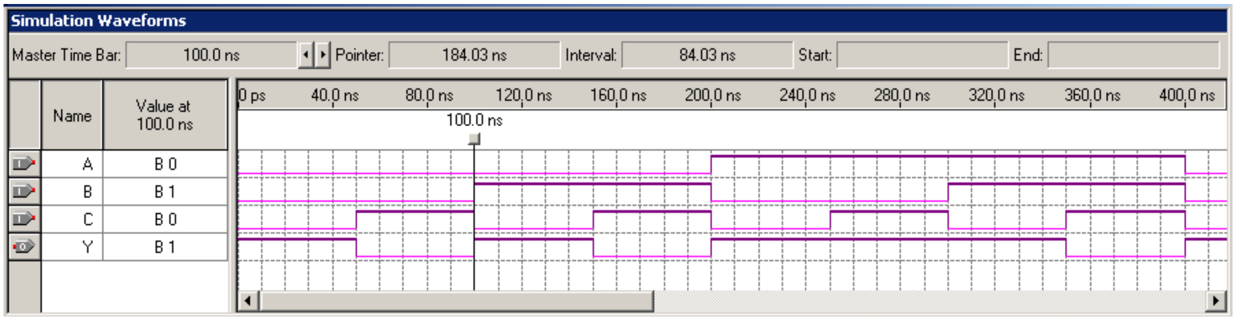
When designing something big, it is a good idea to keep a lot of data available for simulation. Then, when you know that your design simulates correctly, you can remove as many outputs as needed to get the design to fit into your particular device. Extra signals can be output and used in your simulation. I call these extra outputs **debug outputs**.

A **functional** compilation and simulation has (effectively) no limit on the number of inputs, outputs and internal elements available. Below, I discuss how to functionally compile and simulate your design in Quartus.

First design your parts as usual (either with the graphics editor or in VHDL). There is no need to assign a device to the design.

- To compile **functionally**, pull up the “Compiler Tool.” This tool can be found under “Processing | Compiler Tool.” Select the left-most button under “Analysis and Synthesis.” (This button has an triangle point right, an AND gate, and a check mark.) This will compile your design with out trying to fit it into any particular part. When your design compiles without error, you are ready to functionally simulate.
- Create a waveform file as you have done previously.
- Open the “Simulator Tool” under “Processing | Simulator Tool.” Under “Simulation Mode” select “Functional.” Then select the button labeled “Generate Functional Simulation Netlist.” Check the box labeled “Overwrite simulation input file with simulation results.” Select the “Start” button. Your functional simulation will now be completed.

The figure below shows the functional simulation of the previous example. It should be noted that the functional simulation will **not** show propagation delays. Compare this simulation output to the timing simulation output done previously (with propagation delays).



Simulation Waveforms

Master Time Bar: 100.0 ns Pointer: 184.03 ns Interval: 84.03 ns Start: End:

Name	Value at 100.0 ns
A	B 0
B	B 1
C	B 0
Y	B 1

When you are happy with your simulation results, you can remove the debug outputs you don't really need. Then go back to the compiler tool to re-compile your design for your PLD. Before compiling, make sure to assign the appropriate device for your design. Now go back to the compiler tool and select the "Start" button to do a complete compilation (which will include the "Fitter, Assembler and Timing Analyzer"). Then, in the "Simulator Tool" under "Simulation Mode, select Timing." Then select the "Start" button. Your timing simulation will now be completed. The timing simulation will show propagation.

III. Pin Assignments (when you use Quartus to program a device)

There are two ways to have a device's pins assigned to Inputs & Outputs in your Quartus design:

1. Let Quartus do it automatically and then look them up.
2. Assign them yourself in the design.

These two methods are now described.

Quartus Automatic Pin Assignment

If you don't tell Quartus what pins to use on your particular device, it will automatically choose them for you. This is actually very good because the software will try to fit your design using all the available resources (gates) in the device. This is usually what is usually done when first creating a design for a particular device. You simply compile the design and pins are automatically selected. The question now becomes how do we look up the pins that Quartus has chosen for I/O in your design:

1. You need to open a window called the "Pin-Out File". This is done as follows.
2. On the "Processing" menu select "Compilation Report" and then select "Fitter | Pin-Out File". (There is also a "Pin Planner" icon shown her with a pencil on a chip and an "I". If you hold your mouse over this "Pin Planner", you will see its name show up right under the icon. You can also find the "Pin Planner" under "Assignments | Pin Planner". In the Pin Planner, select this puzzle pieces icon to "Show Fitter Placements.") If you save the pin assignments, you will be able to see the pin assignments in the lower part of the window. To save the pin assignments, select "Assignments | Back-Annotate Assignments…"

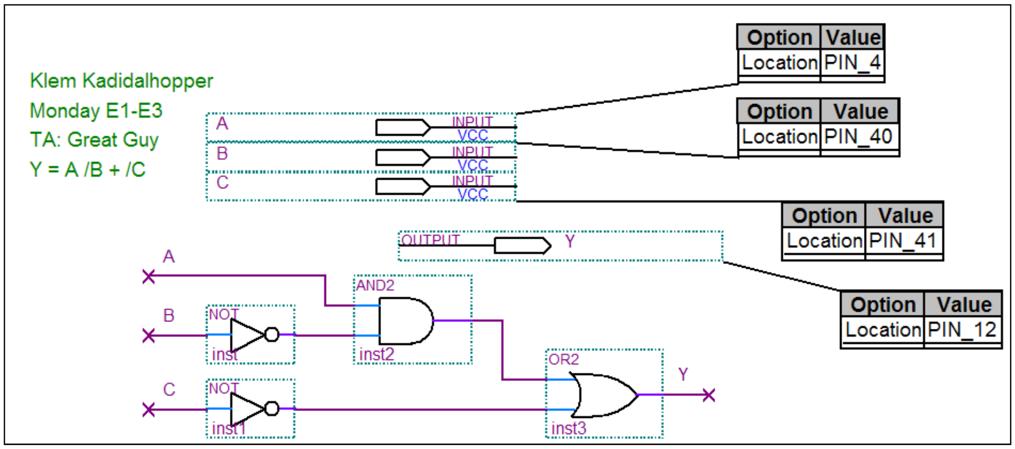
   Pin Planner Icon

   Show Fitter Placements Icon
3. Alternatively, you can select "Assignments | Timing Closure Floorplan". When you hold your mouse over the input signals A, B, C and output Y, you should see it defined as an I/O pin one the chip with a pin number designation.

Special Note: Make sure that you assign the device appropriately before compiling your design.

Manually configuring a Pin Assignment (or altering the Quartus Pin Assignments)

Back annotation allows you to store the pin associations of a design. Future compilations will try not to change these pin numbers. To execute back-annotation select "Assignments | Back-Annotate Assignments…" Select "Pins & device assignments" and then select "OK". Pin numbers will now appear on the bdf design. Once you have back annotated your design, if you want to change any of the pins, you can select the "Assignment Editor" from the "Assignments" menu (or use Ctrl-Shift-A or the button that looks like a pencil writing on a blue chip, shown here). Now you can just re-type the pin number you want. You can alternately move pins around on the Floorplan View. Again, if any changes are made, you must re-compile your design before downloading your design to your board. The figure below shows the default pin assignments from the example we have been working on.



Option	Value
Location	PIN_4

Option	Value
Location	PIN_40

Option	Value
Location	PIN_41

Option	Value
Location	PIN_12

To show pin locations on the Block Diagram/Schematic file, hit the right mouse key you're your cursor in the Block Diagram/Schematic window. Select "Show | Show Location Assignments". The pin numbers will now appear (slightly different from above).

Compile your design and verify the signal pins in the Floorplan View or in the Pin-Out File. They should correspond to the pins you have set (providing you chose valid I/O pins for the particular device that you are using).

If you want to use the pins that you have just defined and make them fixed as you edit your design, than under "Assignments" menu select "Back-Annotate Assignments…" Select "Pins & device assignments" and then select "OK". You must re-compile your design for the back annotation to take effect.

Protecting your Device (Tri-stating unused pins)

In the default configuration, Quartus will program unused pins as outputs. These functions of these pins can be destroyed if they are connected as inputs. To prevent this from occurring, unused pins should be set as inputs and tri-stated. To tri-state unused inputs, execute the following steps.

1. In the "Assignment" menu select "Device"
2. Select "Device and Pin Options"
3. Select the "Unused pins" tab and then select "As input tri-stated"

IV. Some More Settings I Use

In the simulator tool, I suggest that you select "Overwrite simulation input file with simulation results." This will save time each time you simulate. Quartus will remember this setting.

Since the propagation delay of most of the devices we now use is 10ns or less, the test vectors used in simulation should change much less often than 10ns.. To facilitate construction of test vectors with the waveform editor, I suggest that you do the following. On the "Tools" menu select "Options" and then select "Waveform Editor." Change the "Default Grid Period" to something at least twice as large as the propagation delay of the chip. I usually use "25ns" or "30ns."

A tool bar that I find helpful is available under the "Tools" menu. Select "Tools", then "Customize," then "Toolbars." Check "Applications" (and leave the "Standard Quartus II" selected).

V. Programming your Device

If you are running Windows in 64-bit mode, you **MUST** boot your computer as following both when installing the USB-Blaster driver and when using Quartus to program your PLD.

- During boot up of the computer, continuously press F8 (but do **NOT** hold it down).
- Scroll down to "Disable Driver Signature Enforcement," then hit Enter.

You must set up the driver before programming your PLD (CPLD or FPGA).

For Windows XP, go to the below site for instructions on how to install the drivers for the USB-Blaster programming device.

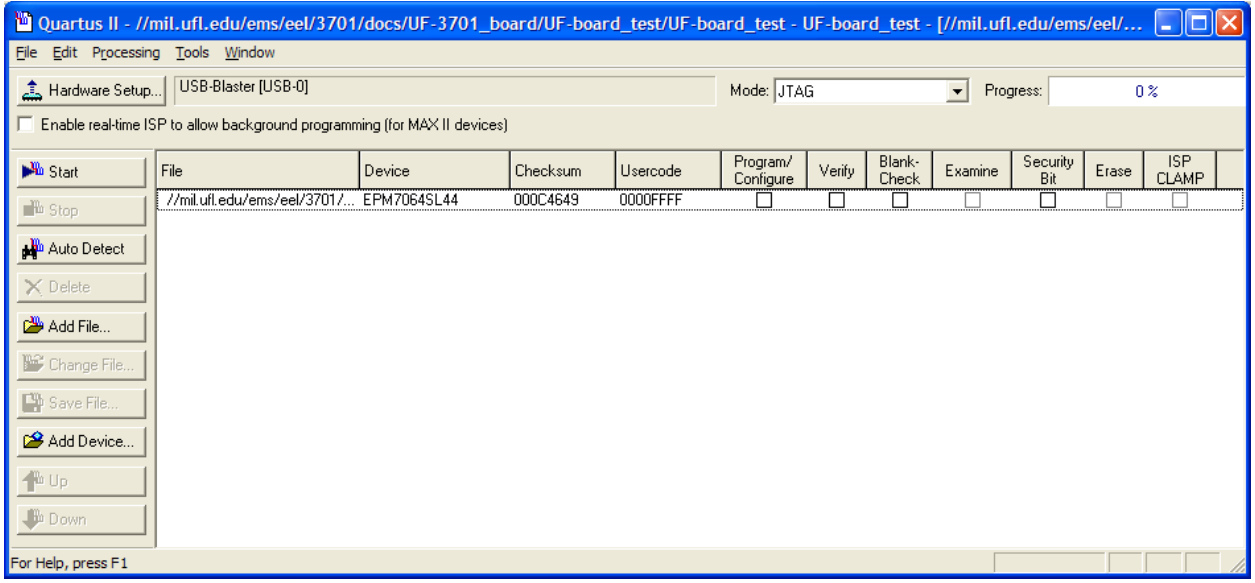
http://www.altera.com/support/software/drivers/usb-blaster/dri-usb-blaster-xp.html

For Windows Vista, go to go to the below site for instructions on how to install the drivers for the USB-Blaster programming device.

http://www.altera.com/support/software/drivers/usb-blaster/dri-usb-blaster-vista.html

After successfully compiling your design (and installing the Blaster driver), you can program your PLD. Be sure that you have noted the pin assignments or changed them as desired (as discussed in Part III of this document).

1. Start the programmer by selecting "Tools | Programmer".
2. Select "Hardware Setup" and then select (double-click on) USB-Blaster (or ByteBlaster if you are using a parallel port ByteBlaster). Then select "Close". You should see the "USB-Blaster [USB-0]" appear next to Hardware Setup as shown below


Quartus II - //mil.ufl.edu/ems/eel/3701/docs/UF-3701_board/UF-board_test/UF-board_test - UF-board_test - [//mil.ufl.edu/ems/eel/...

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

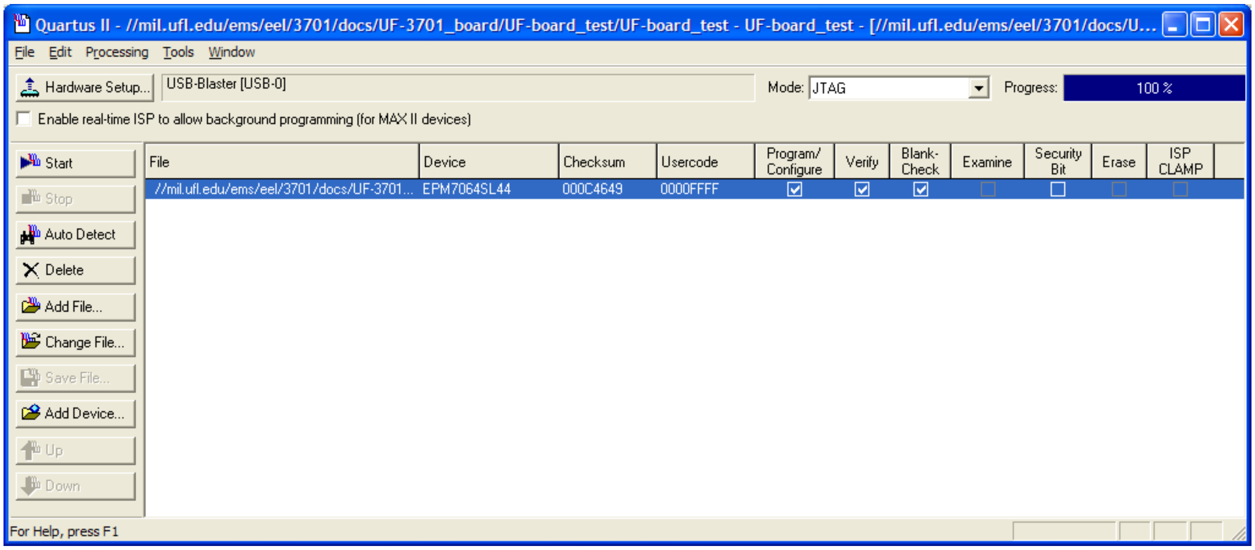
□ Enable real-time ISP to allow background programming (for MAX II devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
//mil.ufl.edu/ems/eel/3701/...	EPM7064SL44	000C4649	0000FFFF	□	□	□	□	□	□	□

For Help, press F1

3. Select "Add File" to program the appropriate "pof" file (if it is not already listed)..
4. Make sure that the "Mode" shows "JTAG".
5. Select "Program/Configure". A check mark will appear in the box. If you can also select "Verify", and "Blank Check", then do so.

6. Select "Start" to program your device. See the message window for information on the programming. You should see a window like below.


Quartus II - //mil.ufl.edu/ems/eel/3701/docs/UF-3701_board/UF-board_test/UF-board_test - UF-board_test - [//mil.ufl.edu/ems/eel/3701/docs/U...
File Edit Processing Tools Window
Start
Stop
Auto Detect
Delete
Add File...
Change File...
Save File...
Add Device...
Up
Down
For Help, press F1

Hardware Setup... USB-Blaster [USB-0]
Mode: JTAG
Progress: 100 %
□ 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
//mil.ufl.edu/ems/eel/3701/docs/UF-3701...	EPM7064SL44	000C4649	0000FFFF	☑	☑	☑	□	□	□	□

VI. Using SignalTap II Logic Analyzer

For information on using the SignalTap II Logic Analyzer in Quartus, please see the document *Tutorial for SignalTap II Logic Analyzer* on the class website.

VII. Technology Map Viewer and State Machine Viewer

To see a block diagram of the parts that Quartus actually uses to create your design (a visual representation of the equations files), execute the following steps.

1. In the "Tools" menu select "Netlist Viewers"
2. Select either "Technology Map Viewer" or "Technology Map Viewer (Post-Mapping)"

Another interesting block diagram can be found with the "State Machine Viewer" under the "Tools | Netlist Viewers".

It should be noted that Quartus will separate every 50 nodes onto another page (upper right hand corner, "Page x of y"), unless you select to Tools | Options | Netlist Viewers and change "Nodes per page" to a higher number.