Tips

- Create a folder where you will store all of your Quartus projects.
- Do **NOT** use spaces and special characters (-, +, *, /, %, etc) in Quartus file/project names. Special characters are also not allowed in the path to the files. So, for example, c:/users/pat/Quartus Projects/… is **NOT allowed** since there is a blank space between Quartus and Projects. Also, avoid common digital-related names like AND2, NOR3, block, reg, input, output, in, out, or any HDL (VHDL or Verilog) keywords (all are known to cause Quartus failure). Using these names may cause either compilation or simulation errors.
- Accept defaults when saving files.
- Pin and wire labels must always start with a letter (not a number or an underscore). Numbers and underscores are allowed elsewhere in the name.
- If you change the device, you must recompile before programming the PLD.
- If you change pin numbers, you must recompile before programming the PLD.
- I recommend that you do **NOT** run Quartus across any cloud (e.g., OneDrive) or network drive. This has caused problems in the past.
- Use **labels** instead of crossing wires.

Driver Installation

If you have not yet installed the driver necessary to program your DE10-Lite, attach the DE1-Lite to your computer and complete the driver installation as specified below. This should work for both Windows 11 and Windows 10.

a) Connect the USB-Blaster (DE10-Lite) to your PC.
b) Open Device Manager.
c) Locate `USB-Blaster` under `Other devices`.
d) Right-click on `USB-Blaster` and select `Update Driver`.
e) Choose `Browse my computer for drivers`.
f) Select the `Browse...` button in the new window.
g) Navigate to your Quartus installation directory:
   - This is typically found at: `C:\intelFPGA_lite\22.1std\quartus\drivers`
     - If you can't find this directory, try clicking "`This PC`" then you should be able to find the C: drive or the `intelFPGA_lite` folder.
   - Adjust the path according to your specific Quartus version (e.g., it may not be `22.1std` for you).
   - Note: Stop at the `drivers` folder, i.e., do not go deeper by opening a folder within the `drivers` folder.
h) Confirm the file path and click `Next`.
i) If prompted by Windows Security:
   - Check the box for `Always trust software from "Altera Corporation"`.
   - Click `Install`.

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. Start by creating truth and voltage tables for this circuit. 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. Upon completion of the schematic entry portion of the example, simulate the circuit and verify that your simulation matches your voltage table. Finally, turn in copies of the circuit & simulation results.

I. Creating a Project in Quartus

A. New Project Design Creation

1. Set up a folder named 'hw1_tutorial' on your PC to hold your design & simulation files. Ensure that the folder is not inside a cloud storage program like OneDrive. **Verify there are no spaces in the path name of the folder.**
2. Launch the Quartus Prime software.
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 hw1_tutorial. (This may already be done for you. This should also make the top-level design entity name hw1_tutorial. If not, again type in hw1_tutorial.) Select "Next", then "Next" (for "Empty project"), and "Next" (for "Add files").
6. Click the "Board" tab towards the top of the screen, change the family to "MAX 10", and select "MAX 10 DE10 – Lite". **De-select** the checkbox labeled "Create top-level design file" towards the bottom of the window. Select "Next".
7. In "EDA Tool Setting", for "Simulation" choose the simulator you installed in the Quartus Installation Tutorial, either ModelSim or QuestaSim. Under the "Format(s)" dropdown in the "Simulation" row, choose "VHDL". Select "Next".
8. Select "Finish".

B. Configuring Quartus Window

1. By default, the Project Navigator (left, top), Tasks (left, middle), 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

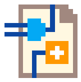


Figure 0: Button for new BDF

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 "hw1_tutorial" in your "hw1_tutorial" 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 a graphical schematic of a circuit design. **In some instances, you cannot save the BDF until you have added something. If this happens, simply proceed, and save the BDF immediately after completion of the next section.**

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 and type the assignment name. For labs, you would type "**Lab # Part #**", where # is replaced by the lab number and the lab part number respectively. For this assignment, you can just type "**Homework 1 Tutorial**". Hit the "Enter" key.
3. Next, type your "**Name:**" followed by your first and last name, and then hit the "Enter" key.

```
Lab #, Part #
Name: Al E. Gator
Class #: #####
PI Name:
Description: Y = A*/B + /C
```

Figure 1: Sample bdf heading

4. Type your "**Class:** #####"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.
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 right click inside the schematic view and select "Insert" then "Symbol"). The "Symbol" dialog box will appear. This window lists the available Altera libraries.
2. Select the > icon to expand the "c:/intelfpga_lite/22.1std/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.

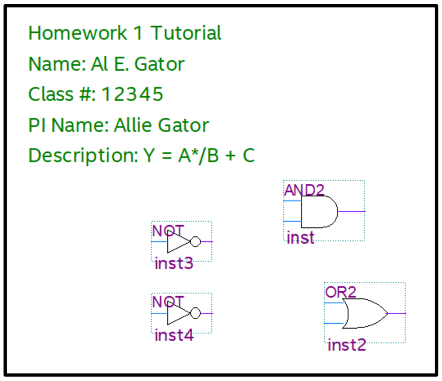

Homework 1 Tutorial
Name: Al E. Gator
Class #: 12345
PI Name: Allie Gator
Description: Y = A*/B + C

Figure 2: BDF with logic symbols

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 gate symbols 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.

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.
2. Place your pointer on the output of one of the Level Shifters. You should see a crosshair or "+" appear at the output. You can then click and drag to create a wire coming out of that output.

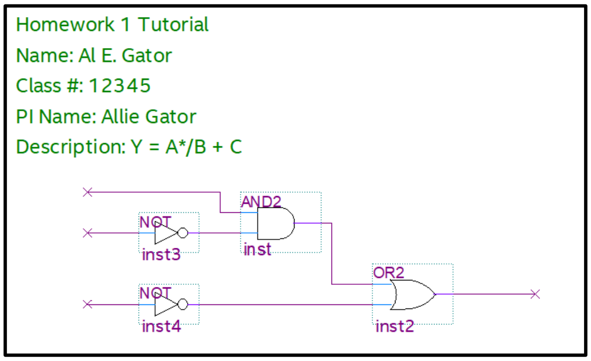

Homework 1 Tutorial
Name: Al E. Gator
Class #: 12345
PI Name: Allie Gator
Description: Y = A*/B + C

Figure 3: BDF with wires

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.

E. Adding Input & Output Ports

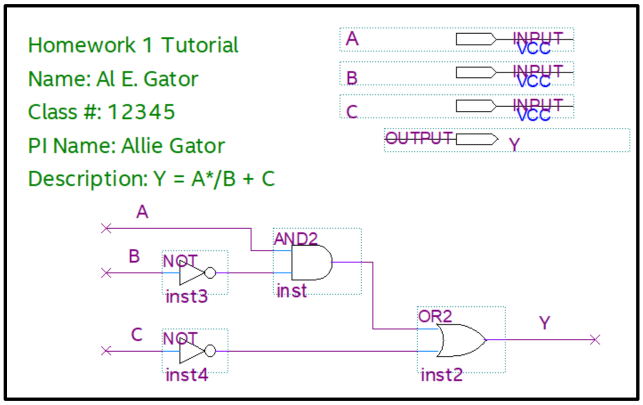

Homework 1 Tutorial
Name: Al E. Gator
Class #: 12345
PI Name: Allie Gator
Description: Y = A*/B + C

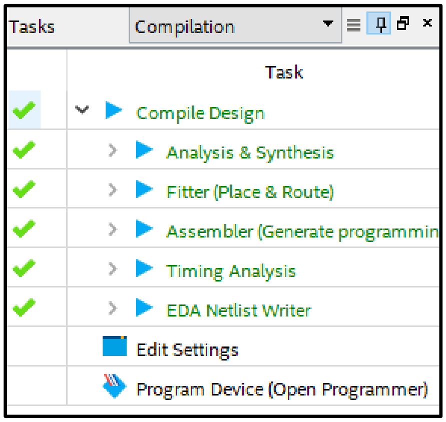
Figure 4: BDF with inputs and outputs

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 | input" (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.
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.

F. Full Compilation (for programming your PLD)

0. For "Functional Compilation," 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 if it has unsaved changes. 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.
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.



Tasks	Compilation
	Task
✓	▶ Compile Design
✓	▶ Analysis & Synthesis
✓	▶ Fitter (Place & Route)
✓	▶ Assembler (Generate programmin
✓	▶ Timing Analysis
✓	▶ EDA Netlist Writer
	Edit Settings
	Program Device (Open Programmer)

Figure 5: Completed Full Compilation

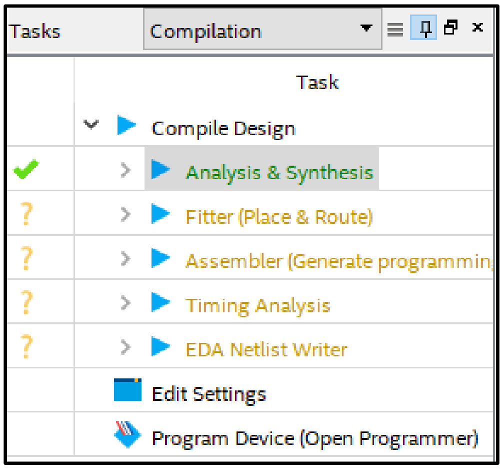
Note: Several common errors involve having floating (unconnected) inputs and outputs or short circuits (possible locations where two inputs or two outputs are connected). Another error is when the top-level entity is undefined; if this occurs choose “Files” from the dropdown menu at the top of the Project Navigator utility window, right click your bdf (in this case hw1_tutorial.bdf), and select “Set as 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., 08A and 08B. Each AND gate on a single 74’08 can be labelled 08-1, 08-2, 08-3, and 08-4.

G. Functional Compilation (for simulation only)

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 8 seconds, whereas the full compilation took 26 seconds.

1. To functionally compile your design, double click the “Analysis & Synthesis” in the Compilation window shown in Figure 6 or select the symbol . 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.



Tasks	Compilation
	Task
	▶ Compile Design
✓	▶ Analysis & Synthesis
?	▶ Fitter (Place & Route)
?	▶ Assembler (Generate programmin
?	▶ Timing Analysis
?	▶ EDA Netlist Writer
	Edit Settings
	Program Device (Open Programmer)

Figure 6: Completed Functional Compile

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 right click and select “Insert Node or Bus”. 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” 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.

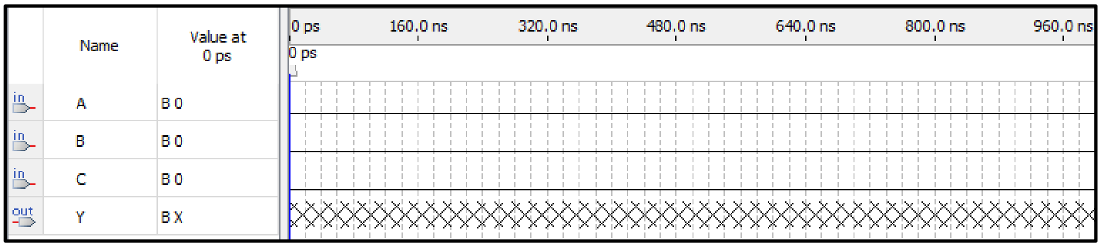


Figure 7: VWF with inputs and outputs

3. **In some cases, simulation may fail with the error "Modelsim Executable Not Found", if this happens, you need to locate your Modelsim installation (mine was C:/intelFPGA_lite/22.1/modelsim_ase/win32aloem) and paste it into "Options | General | EDA Tool Options | ModelSim." Once you have done this, try running the simulation again.**

C. Changing Grid Size and End Time

The default time scale shown above the simulation waveforms 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.

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. Your simulation should look like Figure 8.

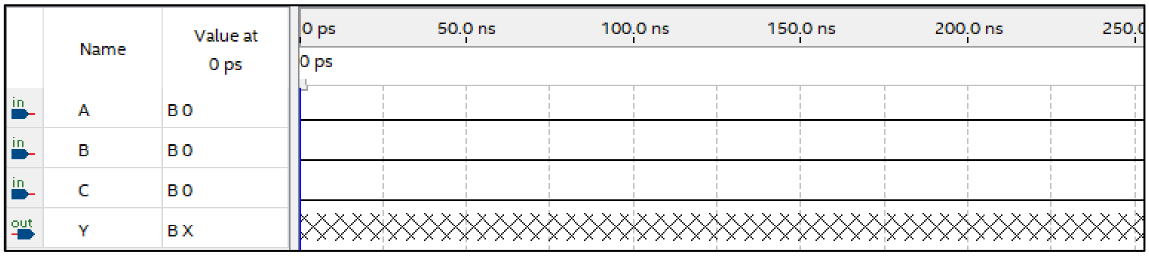


Figure 8: VWF window with 25 ns grid size

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

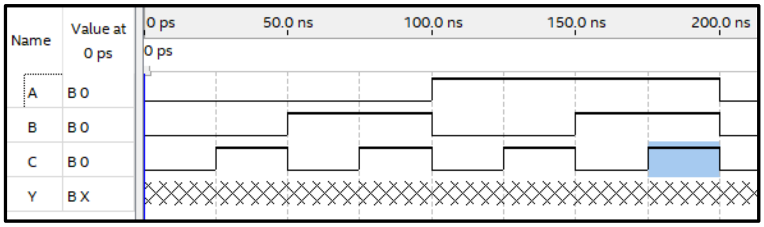


Figure 9: VWF with manual inputs

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 9.
2. Using your mouse (make sure the pointer is selected), click and drag your mouse cursor across 100 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.

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.

Special Note: Quartus runs voltage simulations. Your simulation will thus 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. **Note**: There are better ways to input data then to enter each of the values you want by changing default inputs. See Section F.

E. Functional and Timing Simulation

0. 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 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.
1. Zoom in on the window. You should see something similar to Figure 10. This file is not saved, so if you need a copy of it, use something like the "Snipping Tool" in Windows.
2. 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 since Quartus runs voltage simulations.**
   - 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 program like Snipping Tool or Paint to annotate the simulation. Use arrows and text to describe what is occurring and demonstrate that you received the proper results. Quartus does have a JPEG exporter (use File | Export), but if you use it, be sure to crop the resulting image before inserting it in your lab document.

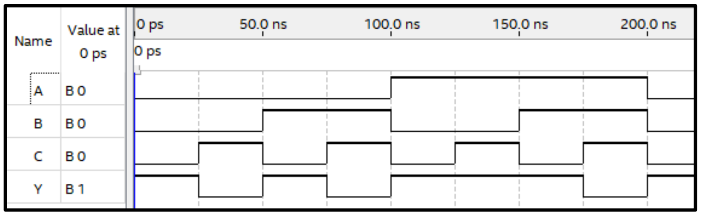


Figure 10: Output of Functional Simulation

3. **Unfortunately, Quartus Prime Lite does not support timing simulations for the MAX 10 devices that we will be using this semester. Because of this, you can only perform functional simulations for the MAX 10.** But Quartus Prime Lite **does** support timing simulations for **MAX V** parts. Therefore, you can and should perform the timing simulation below using any MAX V as described below.
   a. Close the simulation window.
   b. Go to "Assignment" and then" Device" and then select "MAX V" in "Device family | Family:" Now select the first item in the list, for example, "5M2210ZF256I5". Then select "OK".
   c. Re-compile the bdf design file, using a full compilation (not a functional compilation) with the ▶ button.
   d. In the Simulation Waveform Editor, select "Simulation" and then "Simulation Settings". Select "Restore Defaults" and then "Save".
   e. 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.

f. Zoom in on the window. You should see something like Figure 11.

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.

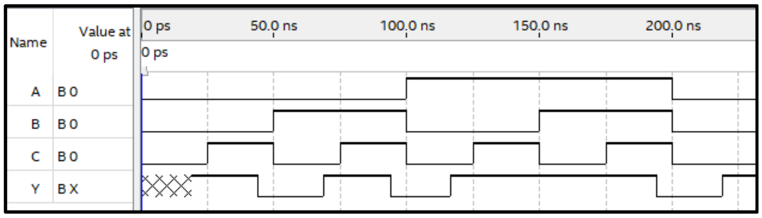


Figure 11: Output of Timing Simulation

7. **Note: This step did NOT seem to work for me. But you will not generally move between devices.** Now change the device back to our MAX 10 DE10 – Lite. (Select "Assignment" and then" Device" and then select "Board" tab towards the top of the screen, change the family to "MAX 10", and select "MAX 10 DE10 – Lite".) Redo the functional compilation. In the Simulation Waveform Editor, select "Simulation" and then "Simulation Settings". Select "Restore Defaults" and then "Save". Now redo the functional simulation.

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.
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.

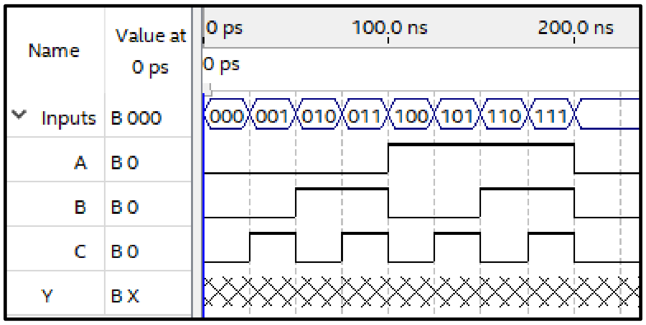


Figure 12: Grouping on vwf

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. Pin Assignments & Programming

A. Assigning pins using Pin Planner

1. To make sure you do **NOT** accidentally assign pins to the wrong location, review the document DE10-Lite_Pins.pdf to see all of the available pins and their intended purposes. Here is an excerpt from that document, with the corresponding DE-10 Lite GPIO Header pins boxed in red on the left. These pins would work well for interfacing with an external breadboard (when using appropriate connectors).

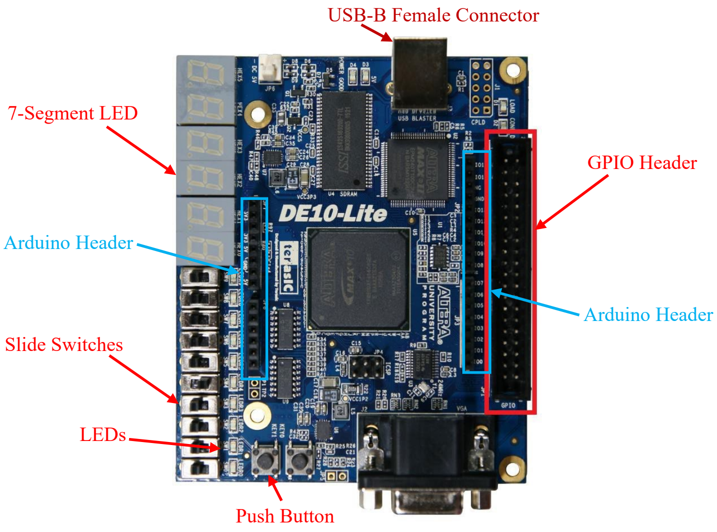
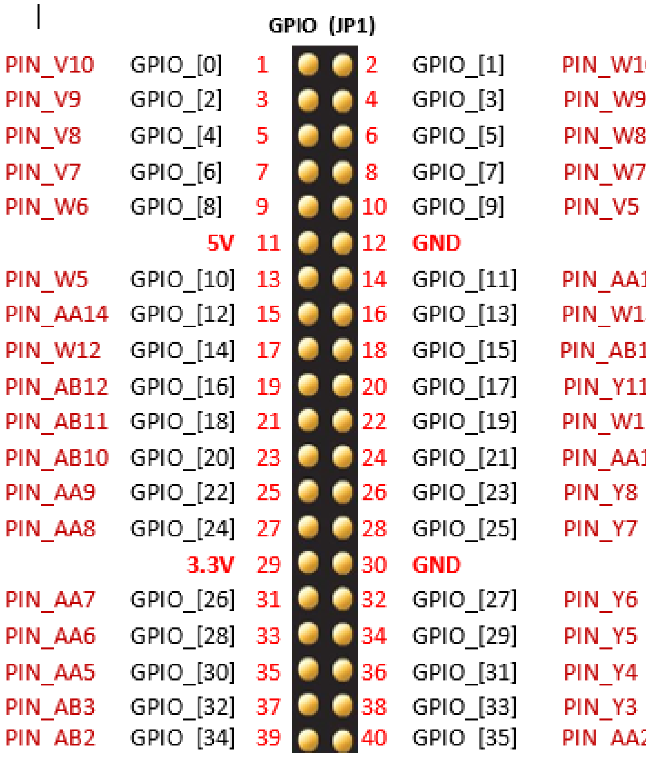


Figure 13: Pin numbers for the female headers on the DE10-Lite (left); Location of the female headers on the DE10-Lite (right)

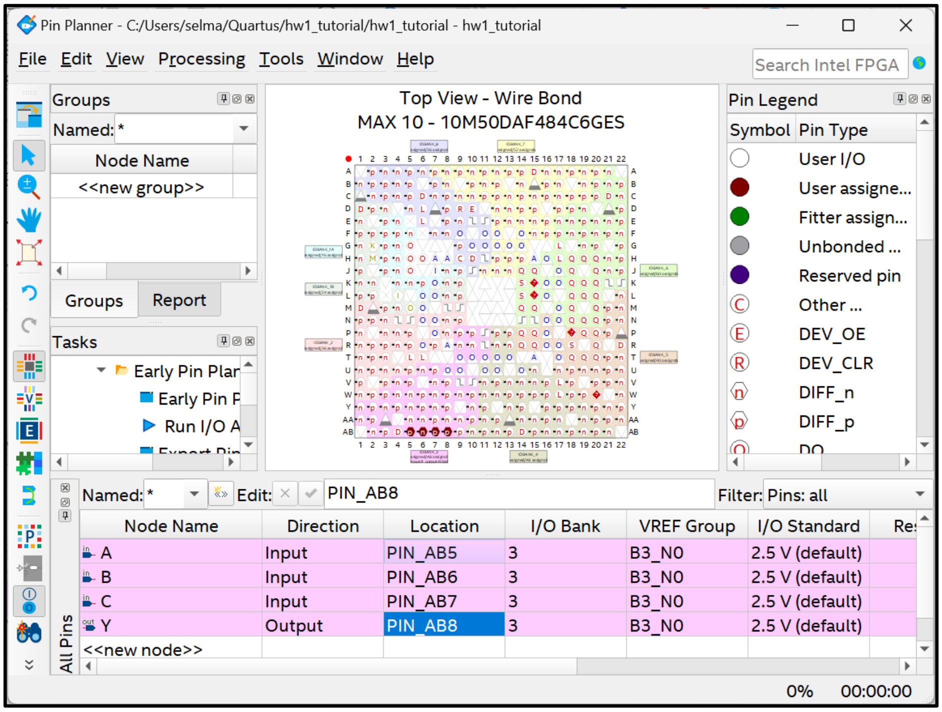


GPIO (JP1)

PIN_V10	GPIO_[0]	1	2	GPIO_[1]	PIN_W1
PIN_V9	GPIO_[2]	3	4	GPIO_[3]	PIN_W9
PIN_V8	GPIO_[4]	5	6	GPIO_[5]	PIN_W8
PIN_V7	GPIO_[6]	7	8	GPIO_[7]	PIN_W7
PIN_W6	GPIO_[8]	9	10	GPIO_[9]	PIN_V5
	5V	11	12	GND	
PIN_W5	GPIO_[10]	13	14	GPIO_[11]	PIN_AA1
PIN_AA14	GPIO_[12]	15	16	GPIO_[13]	PIN_W1
PIN_W12	GPIO_[14]	17	18	GPIO_[15]	PIN_AB1
PIN_AB12	GPIO_[16]	19	20	GPIO_[17]	PIN_Y11
PIN_AB11	GPIO_[18]	21	22	GPIO_[19]	PIN_W1
PIN_AB10	GPIO_[20]	23	24	GPIO_[21]	PIN_AA1
PIN_AA9	GPIO_[22]	25	26	GPIO_[23]	PIN_Y8
PIN_AA8	GPIO_[24]	27	28	GPIO_[25]	PIN_Y7
	3.3V	29	30	GND	
PIN_AA7	GPIO_[26]	31	32	GPIO_[27]	PIN_Y6
PIN_AA6	GPIO_[28]	33	34	GPIO_[29]	PIN_Y5
PIN_AA5	GPIO_[30]	35	36	GPIO_[31]	PIN_Y4
PIN_AB3	GPIO_[32]	37	38	GPIO_[33]	PIN_Y3
PIN AB2	GPIO [34]	39	40	GPIO [35]	PIN AA2

Table 1: List of DE10-Lite internal pin numbers associated with each female header.

To program a device, you need to specify device pins for top level inputs and outputs. Select "Assignments" and "Pin Planner". A window similar to Figure 14 will appear. Under "Location", type in a pin number for A, B, C, and Y. For example, you could choose pins AA5, AA6, AA7, and AA8. These pins are all accessible for breadboarding using the female headers on your DE10-Lite. Press Enter after you type each pin name to confirm the pin assignments. Exit out of Pin Planner.



Symbol	Pin Type
○	User I/O
●	User assigne...
●	Fitter assign...
●	Unbonded ...
●	Reserved pin
Ⓒ	Other ...
Ⓔ	DEV_OE
Ⓡ	DEV_CLR
ⓝ	DIFF_n
ⓟ	DIFF_p
ⓠ	DQ

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Re:
A	Input	PIN_AB5	3	B3_N0	2.5 V (default)	
B	Input	PIN_AB6	3	B3_N0	2.5 V (default)	
C	Input	PIN_AB7	3	B3_N0	2.5 V (default)	
Y	Output	PIN_AB8	3	B3_N0	2.5 V (default)	
<<new node>>						

Figure 14: Pin Planner

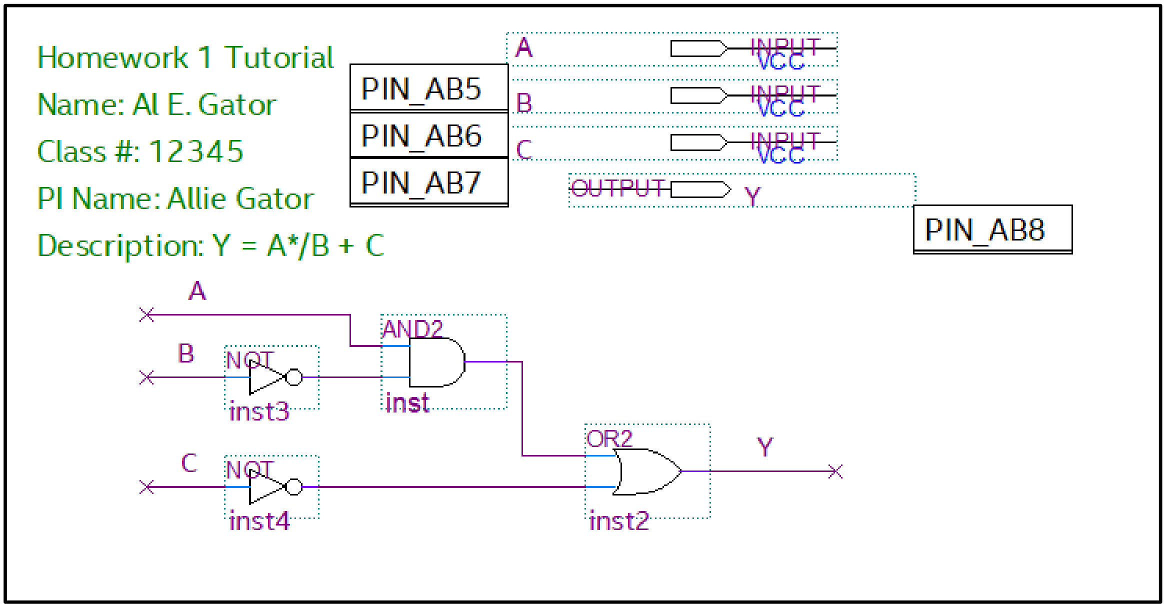

Homework 1 Tutorial
Name: Al E. Gator
Class #: 12345
PI Name: Allie Gator
Description: Y = A*/B + C

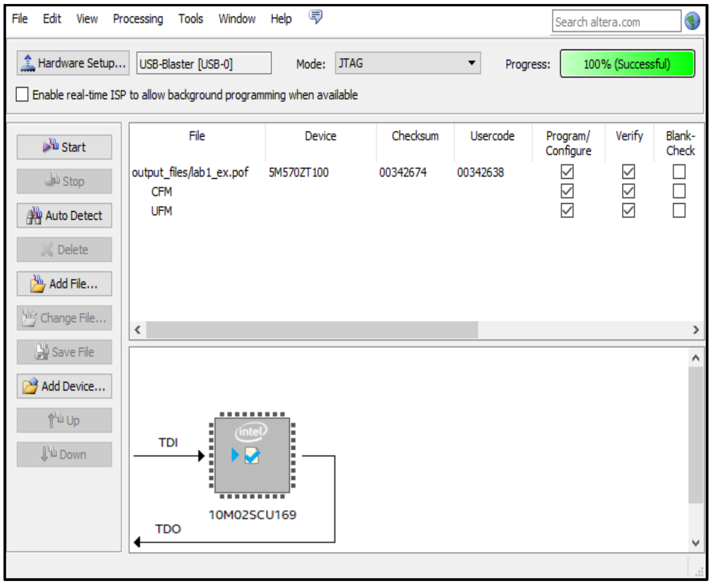
Figure 15: BDF with pin assignments. (Note that the pins show here have replaced AA with AB because of a prior set of pins being utilized.)

The result is that your bdf will now change to look something like Figure 15. Note that you can move the pin numbers around in the bdf.

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 for **EVERY** design that will be downloaded to your PLD. In order for the above to take effect, you must recompile the design and then program the PLD.



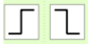
USB-Blaster [USB-0]   Mode: JTAG   Progress: 100% (Successful)

☐ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check
output_files/lab1_ex.pof	5M570ZT100	00342674	00342638	☑	☑	☐
CFM				☑	☑	☐
UFM				☑	☑	☐

Figure 16: Programmer

D. Clock pins

1. When a clock signal is necessary, always choose pins shown by a rising or falling edge in the "Pin Planner," shown as : . Note that the only available clock pin on the GPIO Header is PIN_V10 (GPIO_[0]) and therefore this pin should be utilized for any external clock source.

E. Programming

2. Do **NOT** program your DE10-Lite while the board is connected to anything (other than power and ground) on your breadboard, i.e., remove it from your circuit (switches, LEDs, etc.) before programming or reprogramming.
3. **After completing the next part of this document (part F)**, as long as you **remove power** when wiring up a new circuit, you **can ignore** item 1 above and you will be in no danger of damaging your DE10-Lite's PLD.
4. Using your USB-A cable, connect your board to your computer. A green light should turn on.
5. On Quartus, select "Tools" and "Programmer".
6. Select "Hardware Setup" and then select (double-click on) USB Blaster. If it does not show up, make sure your PLD is plugged into your computer correctly. Then press "Close". If you are on a Mac using a virtual machine, make sure your PLD's USB port 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 hw1_tutorial.sof. Press "Open".
    a. The difference between the SOF and POF is that a POF is "persistent" and will remain programmed even after power is disconnected. SOF files are temporary and your DE-10 will lose its program once it is unplugged from power. **It is recommended that all debugging is done with SOF files since disconnecting power will return you to a safe state. SOF files are also much faster to program than POF files.**
9. Click the boxes under Program/Configure and Verify.
10. Press "Start". Progress should read "100% Successful." If under Progress it says "Failed", make sure that you have the correct Device and Board selected for your project (i.e. that you are not still compiling for MAX V like you did for the timing simulation).
11. To avoid having to repeat steps 6-7 every time you program, you can save this configuration. Press "File" and "Save As". Name your file "hw1_tutorial".

F. Putting your DE10-Lite into a SAFE STATE

If you do the following (programming your DE10-Lite with a **pof**), your DE10-Lite will be safe from damage when you connect it to a breadboard with a circuit already constructed. This is **IF**, in the future, you only program your DE10-Lite with **sof**.

Download the **safe_3701.qar** archive file (available on our website). Un-archive the file. Perform a full compilation. When you open the Quartus programmer, the **sof** file is automatically selected to be programmed. Click on the **sof** file to highlight it. Then press "**Delete**" available in the left panel of buttons. Then click "**Add file**" and navigate to the **output_files** folder within your project. Choose the **pof** file. Now check the boxes for the **pof** under **Program/Configure** and **Verify**. Now you can program your board and put it in a ***SAFE STATE*** for future use in 3701!

G. 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.

H. 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 "Lab6a.qar."

B. Un-archiving (restoring) your project into a qar file

1. To restore the project, open the archive file. Specify a "Destination folder:", for example, t c:/3701/LabX, where X is the Lab number.
2. I have found that when un-archiving (restoring) an archived project, I need to do the following to correct folder path information.
   a. In Quartus, open the vwf file. This will open the Simulator Waveform Editor.
   b. 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.
3. Note the "Restore Defaults" is a common correction that is sometimes necessary to get a simulation to work properly.

VI. Miscellaneous

A. Quartus Wire

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, as shown in Figure 17.

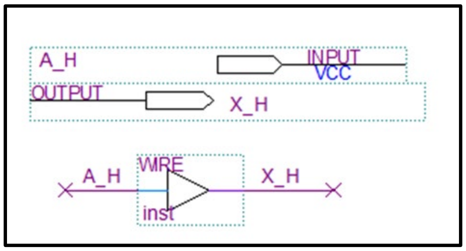
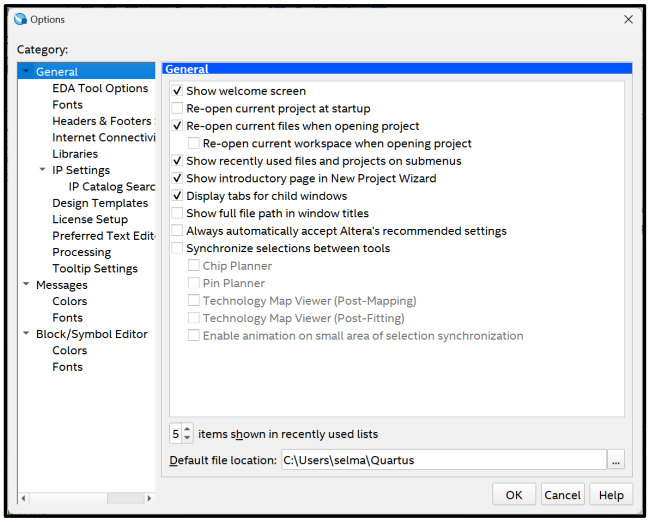


Figure 17: Wire component

B. Setting up a Default Directory

1. Open Quartus.
2. On the top row of commands, select Tools, then select Options. A new window will appear.
3. Navigate to the 'General' category.
4. As shown in Figure 18, above the "OK" button near the bottom of the window, you will see an option for "Default File location:". Click on the ellipsis, then find the directory that you would like to use as your default when creating a new project using the Project Wizard and click "Select Folder".
5. Once this default file path has been selected, select "OK".
6. Finally, close the Quartus application to save this change. When you reopen Quartus, the default directory (folder) will be active.



General

☑ Show welcome screen
☐ Re-open current project at startup
☑ Re-open current files when opening project
☐ Re-open current workspace when opening project
☑ Show recently used files and projects on submenus
☑ Show introductory page in New Project Wizard
☑ Display tabs for child windows
☐ Show full file path in window titles
☐ Always automatically accept Altera's recommended settings
☐ Synchronize selections between tools
☐ Chip Planner
☐ Pin Planner
☐ Technology Map Viewer (Post-Mapping)
☐ Technology Map Viewer (Post-Fitting)
☐ Enable animation on small area of selection synchronization

5 items shown in recently used lists

Default file location: C:\Users\selma\Quartus ...

Figure 18: Changing the default folder (directory).

VI. Installing Another Family of Devices.

If you realize that you need to install a new family of devices after already installing Quartus, follow these instructions.

1. Go to the ***Quartus Installation Instructions***. Follow the instructions to download **ONLY** the device family that you want, e.g., "Intel Cyclone V Device Support."
2. Go to the Windows Start Menu (by selecting the Window icon shown here). Click "All apps"
   a. Select "Intel FPGA ..." and then "Device Installer (Quartus ...)" as shown in Figure 19.
   b. Then follow the directions!

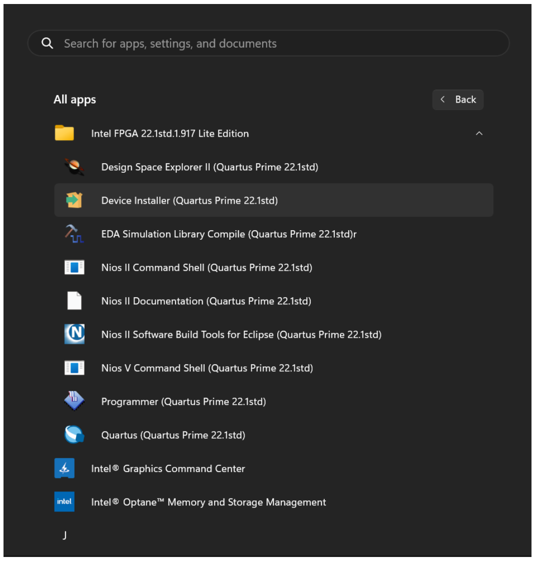
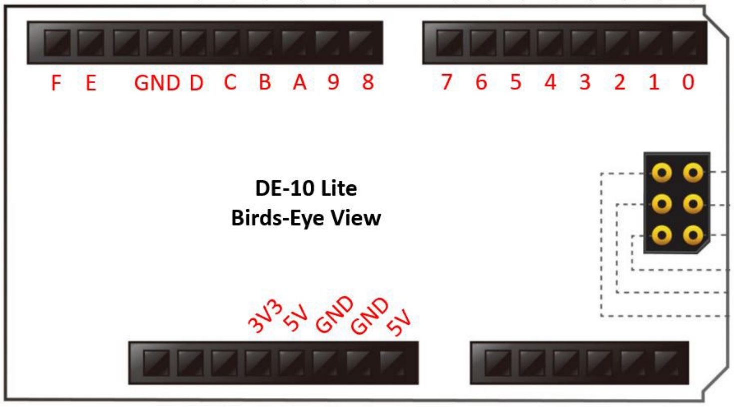

Search for apps, settings, and documents
All apps
Back
Intel FPGA 22.1std.1.917 Lite Edition
Design Space Explorer II (Quartus Prime 22.1std)
Device Installer (Quartus Prime 22.1std)
EDA Simulation Library Compile (Quartus Prime 22.1std)r
Nios II Command Shell (Quartus Prime 22.1std)
Nios II Documentation (Quartus Prime 22.1std)
Nios II Software Build Tools for Eclipse (Quartus Prime 22.1std)
Nios V Command Shell (Quartus Prime 22.1std)
Programmer (Quartus Prime 22.1std)
Quartus (Quartus Prime 22.1std)
Intel® Graphics Command Center
Intel® Optane™ Memory and Storage Management
J

Figure 19: Finding the Quartus device installer in the Windows Start Menu.

ARDUINO HEADERS

- Useful for connecting male-male jumper wires to an external breadboard. (Not used in 3701.)



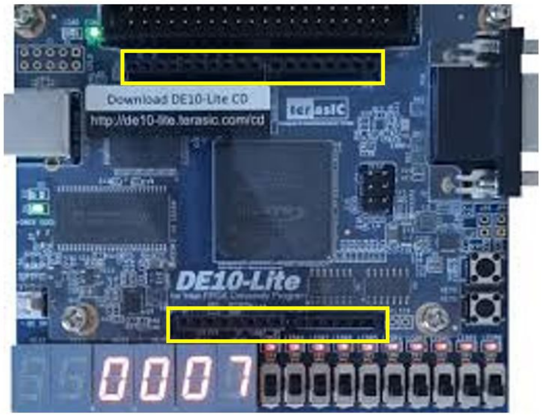


Figure 20: Pin numbers for the female headers on the DE10-Lite (left); Location of the female headers on the DE10-Lite (right).

Pin	FPGA Pin #
0	PIN_AB5
1	PIN_AB6
2	PIN_AB7
3	PIN_AB8
4	PIN_AB9
5	PIN_Y10
6	PIN_AA11
7	PIN_AA12
8	PIN_AB17
9	PIN_AA17
A	PIN_AB19
B	PIN_AA19
C	PIN_Y19
D	PIN_AB20
E	PIN_AB21
F	PIN_AA20

Table 2: List of DE10-Lite internal pin numbers associated with each female header.