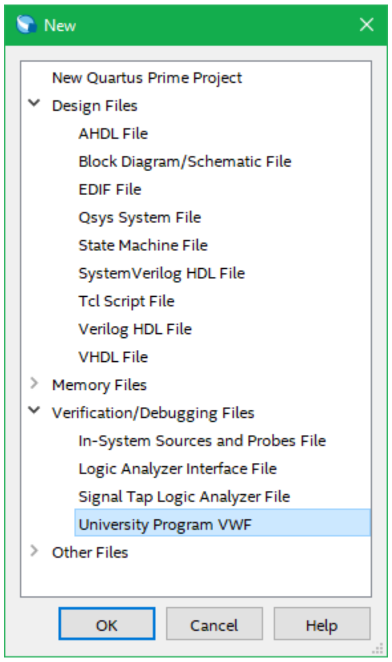
Quartus Tutorial – Simulations

Quick Recap

We last left off the Quartus tutorial having just finished the BDF file. We will use the BDF file for the simulation tutorial, but the same exact steps will work for VHDL.

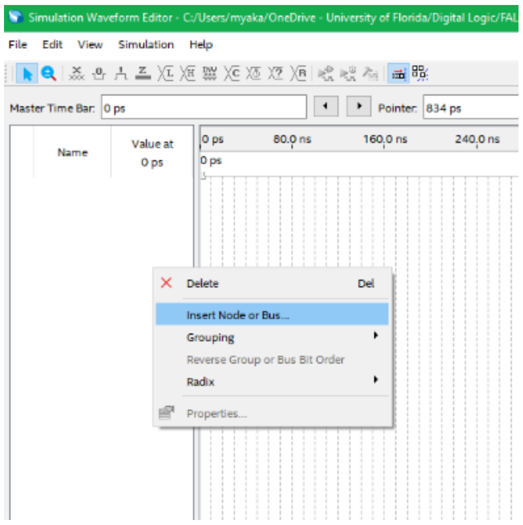
Step 1 – Create the Waveforms file

Click on **File > New**, then under the **Verification/Debugging Files**, click **University Program VWF**



Step 2 – Setting the grid size and end time

Click on **Edit > Grid Size…** and enter **25ns**. Click **OK**.
Click on **Edit > Set End Time… and enter 1.2us**. Click **OK**.

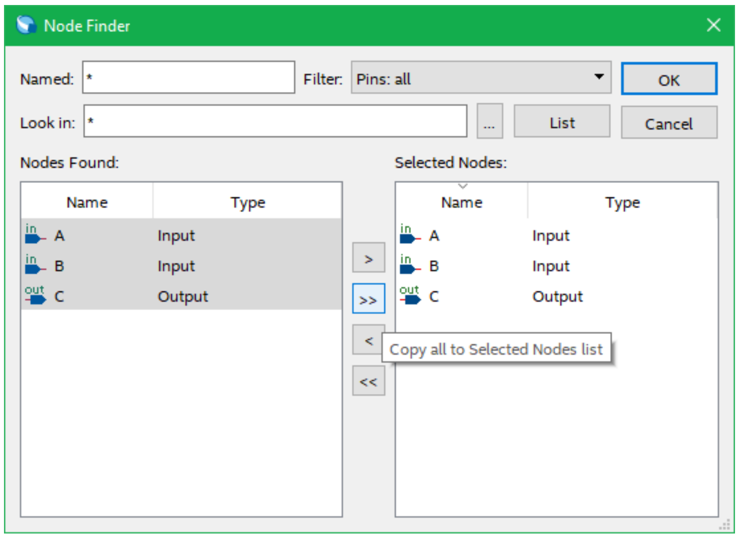


Step 3 – Adding your signals

This step is only necessary when you create a new file or change the inputs/outputs of your circuit

Right click in the empty area on the left, then click **Insert Node or Bus…**.

Then, click on **Node Finder…**. In the window that appears, click on **List**. Then, click on the button with two arrows on it, **>>**. Then click **OK**.


Node Finder

Named: *

Filter: Pins: all

Look in: *

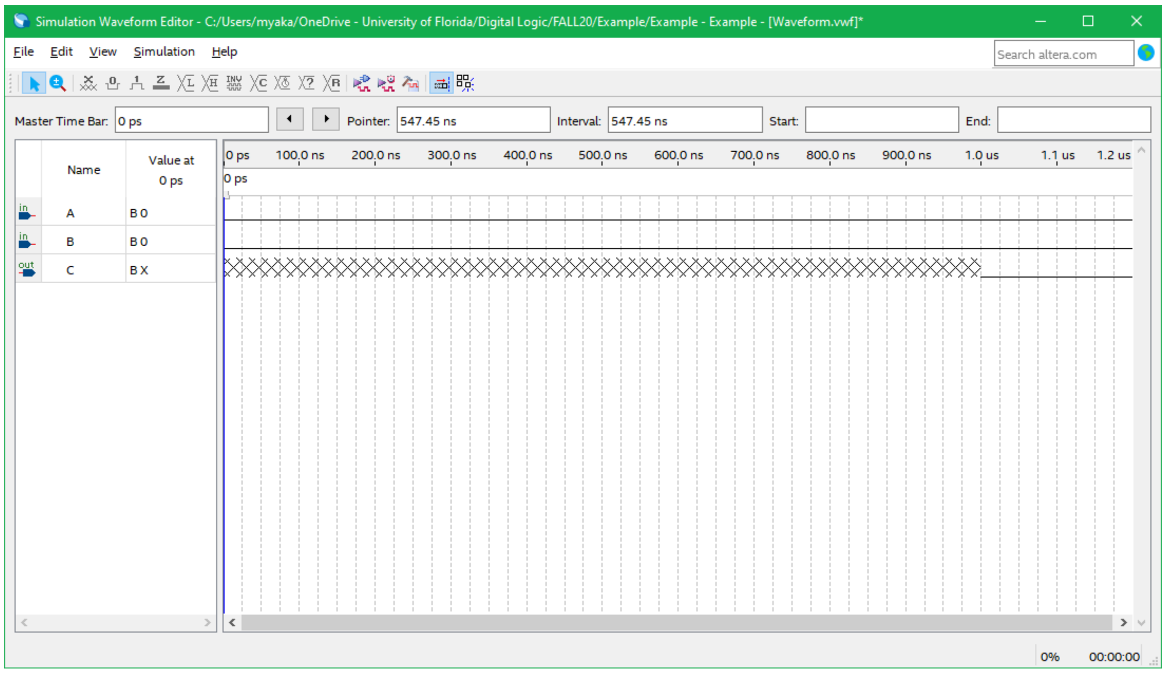
Nodes Found:

Name	Type
A	Input
B	Input
C	Output

Selected Nodes:

Name	Type
A	Input
B	Input
C	Output

You should now see a screen that looks like this:


Simulation Waveform Editor - C:/Users/myaka/OneDrive - University of Florida/Digital Logic/FALL20/Example/Example - Example - [Waveform.vwf]*

Master Time Bar: 0 ps

Pointer: 547.45 ns

Interval: 547.45 ns

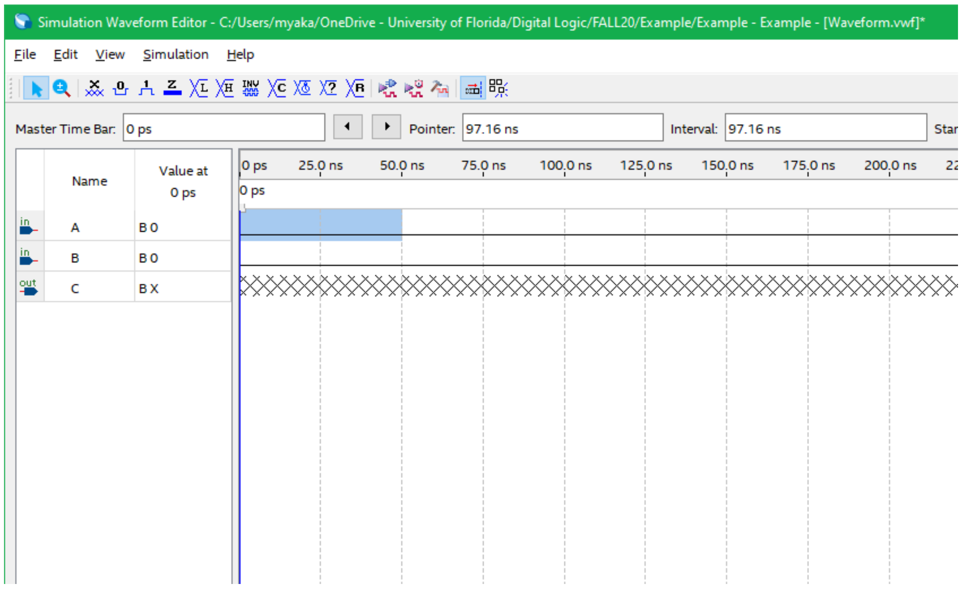
Start:

End:

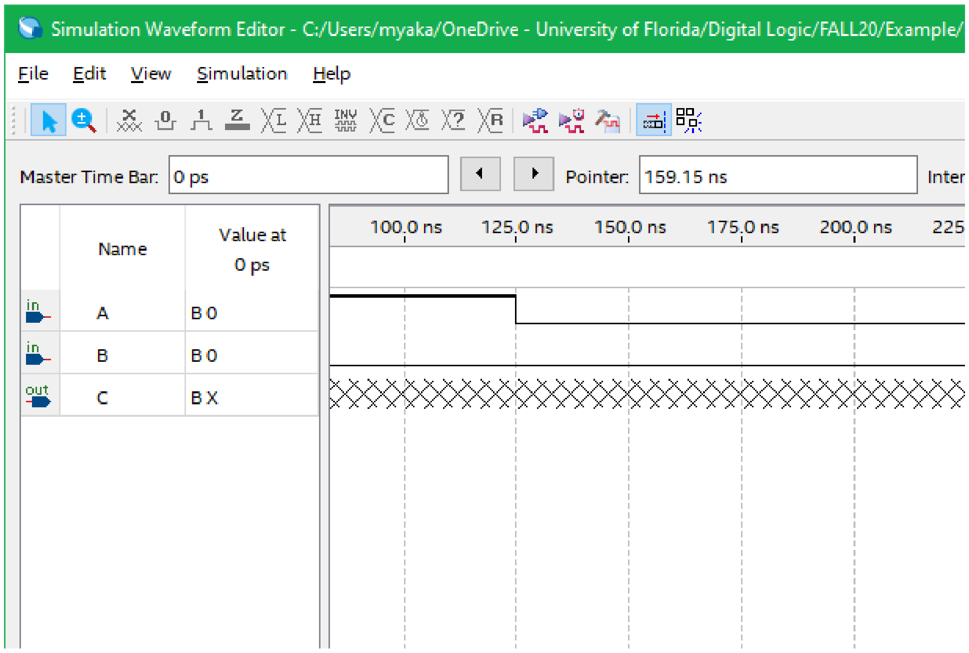
Name	Value at 0 ps
A	B 0
B	B 0
C	B X

Step 4 – Setting values

Zoom in so you can see each division clearly using the magnifier glass or **Ctrl+Scroll**. Then, dragging in a downward-rightward motion, select two divisions of signal A, such as in the picture:

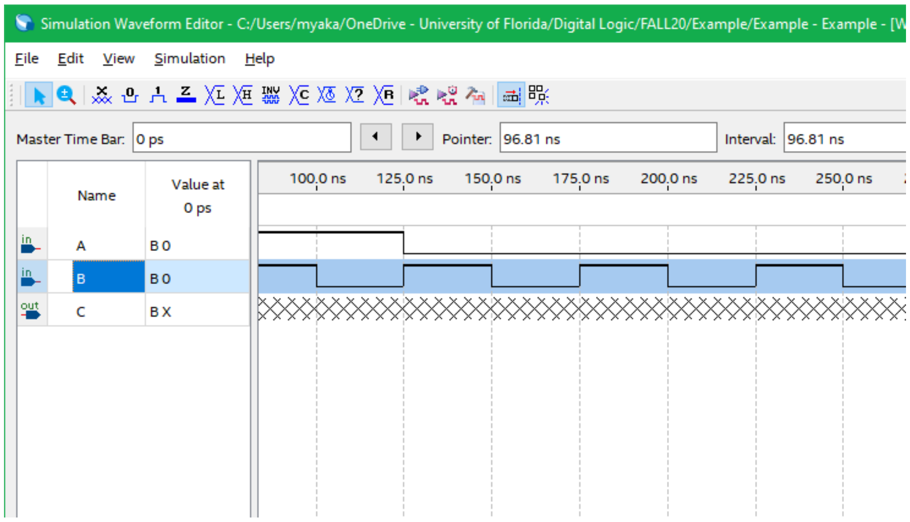

Simulation Waveform Editor - C:/Users/myaka/OneDrive - University of Florida/Digital Logic/FALL20/Example/Example - Example - [Waveform.vwf]*
File Edit View Simulation Help
Master Time Bar: 0 ps
Pointer: 97.16 ns
Interval: 97.16 ns
Name
Value at 0 ps
A B 0
B B 0
C B X
0 ps
25.0 ns
50.0 ns
75.0 ns
100.0 ns
125.0 ns
150.0 ns
175.0 ns
200.0 ns

Click on the **1** button in the toolbar on the top. You should see the two segments go high. Click on the **0** button, and you should see the segments go back low. Click on the **1** button again and you should see this:

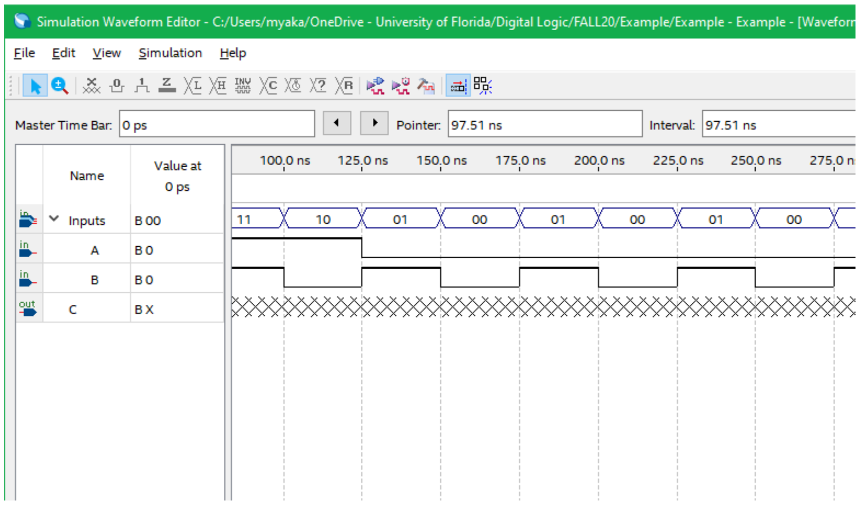

Simulation Waveform Editor - C:/Users/myaka/OneDrive - University of Florida/Digital Logic/FALL20/Example/
File Edit View Simulation Help
Master Time Bar: 0 ps
Pointer: 159.15 ns
Name
Value at 0 ps
A B 0
B B 0
C B X
100.0 ns
125.0 ns
150.0 ns
175.0 ns
200.0 ns

Next, click on the square for **B** on the left to select the whole row. Then, click on the **C** button on the **toolbar**. Then, enter **25ns** on the bottom and click **OK**. You just used the "Counter" mode to make a clock signal.

The counter mode will count a binary number up until its maximum value, and then reset it back to 0. In this case, because B is only 1 bit, it will continuously count between 0 and 1 – a clock!

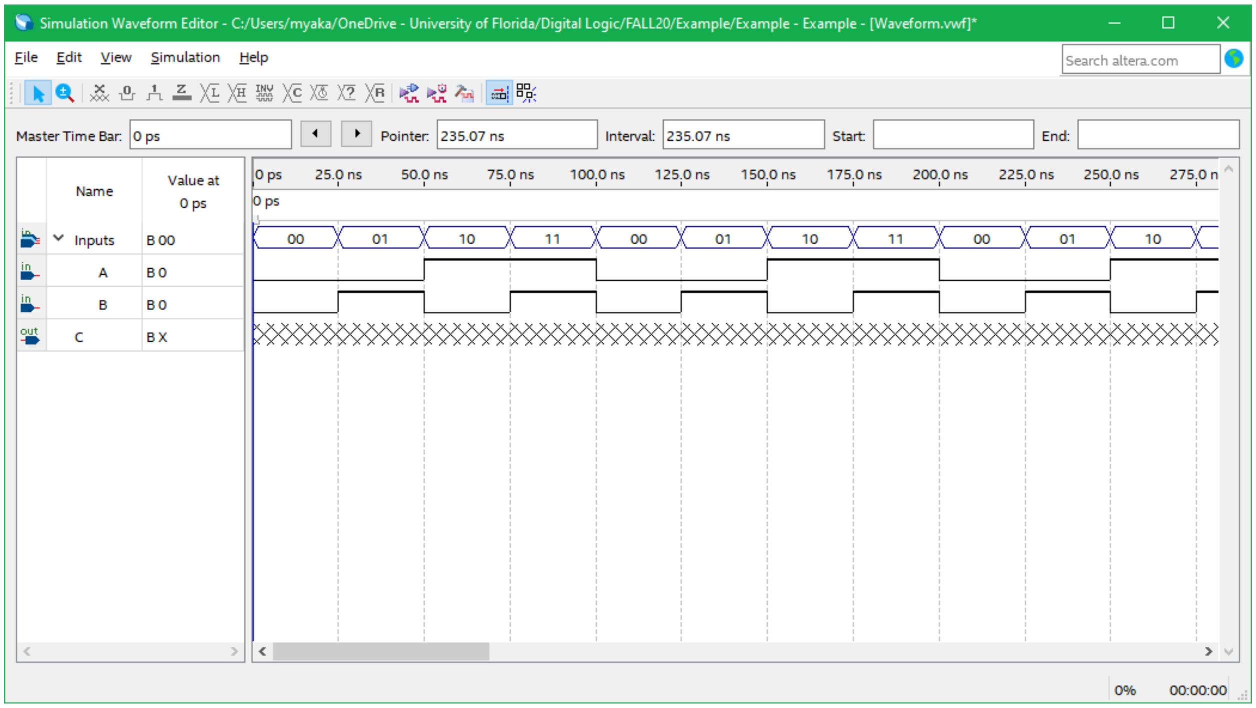

Simulation Waveform Editor - C:/Users/myaka/OneDrive - University of Florida/Digital Logic/FALL20/Example/Example - Example - [W
File Edit View Simulation Help
Master Time Bar: 0 ps
Pointer: 96.81 ns
Interval: 96.81 ns
Name
Value at 0 ps
100,0 ns 125,0 ns 150,0 ns 175,0 ns 200,0 ns 225,0 ns 250,0 ns
A B 0
B B 0
C B X

The counter mode can also be applied to "Groups" of inputs. **Hold Shift** and click on both **A** and **B** to select both at the same time. When both A and B are blue, **right click** on either of them, then click on **Grouping > Group….** Enter "Inputs" for the group name and keep the radix as binary. Expand the new group on the left and you should see the following:


Simulation Waveform Editor - C:/Users/myaka/OneDrive - University of Florida/Digital Logic/FALL20/Example/Example - Example - [Wavefor
File Edit View Simulation Help
Master Time Bar: 0 ps
Pointer: 97.51 ns
Interval: 97.51 ns
Name
Value at 0 ps
100,0 ns 125,0 ns 150,0 ns 175,0 ns 200,0 ns 225,0 ns 250,0 ns 275,0 n
Inputs B 00
A B 0
B B 0
C B X
11 10 01 00 01 00 01 00

As you can see, the Inputs group is now treated as a single binary number, with A as the most significant bit (MSB), and B as the least significant bit (LSB). For future reference, if you need to switch the order, right click the group, and select Reverse Group or Bus Bit Order. However, we will not be doing that for this tutorial.

Finally, click on “Inputs” on the left to select the whole group, then click on the counter mode button at the top (the **C** button). Enter **25ns** and click **OK**. You should now see the Inputs group counting up from 0 to 3 in a continuous loop.


Simulation Waveform Editor - C:/Users/myaka/OneDrive - University of Florida/Digital Logic/FALL20/Example/Example - Example - [Waveform.vwf]*
File Edit View Simulation Help
Search altera.com
Master Time Bar: 0 ps
Pointer: 235.07 ns
Interval: 235.07 ns
Start:
End:
Name
Value at 0 ps
0 ps 25.0 ns 50.0 ns 75.0 ns 100.0 ns 125.0 ns 150.0 ns 175.0 ns 200.0 ns 225.0 ns 250.0 ns 275.0 n
Inputs B 00
A B 0
B B 0
C B X
00 01 10 11 00 01 10 11 00 01 10
0% 00:00:00

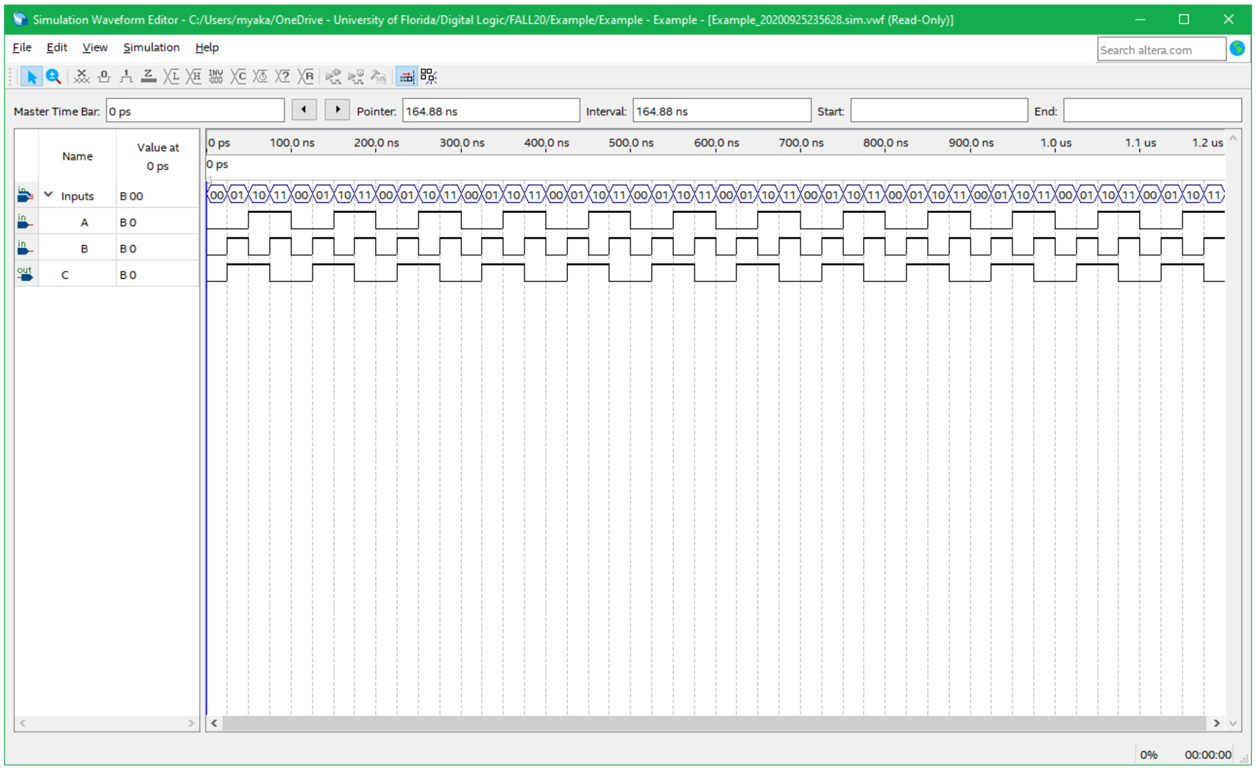
Step 4 – Save and simulate

Now, do **File > Save**, and save it in the root directory of your project (where your BDF and VHDL files are located). Keep the default name.

Common Error! Quartus does not like it when you change the name of the simulation file. But sometimes you must if you plan on doing multiple different types of simulations in the same project. If you save it as a different name, you must click on **Simulation > Simulation Settings**, then click on the **VHDL** selector at the top. Finally, click on **Restore Defaults** and then **Save**.

Now, click on **Simulation > Run Simulation** or the first play button at the top

A new window will pop up with the result of the simulation.



Simulation Waveform Editor - C:/Users/myaka/OneDrive - University of Florida/Digital Logic/FALL20/Example/Example - Example - [Example_20200925235628.sim.vwf (Read-Only)]

Master Time Bar: 0 ps | Pointer: 164.88 ns | Interval: 164.88 ns | Start: | End:

Name	Value at 0 ps
Inputs	B 00
A	B 0
B	B 0
C	B 0

Here, you can see that A and B accurately represent an **XOR** gate, as in it follows the truth table of:

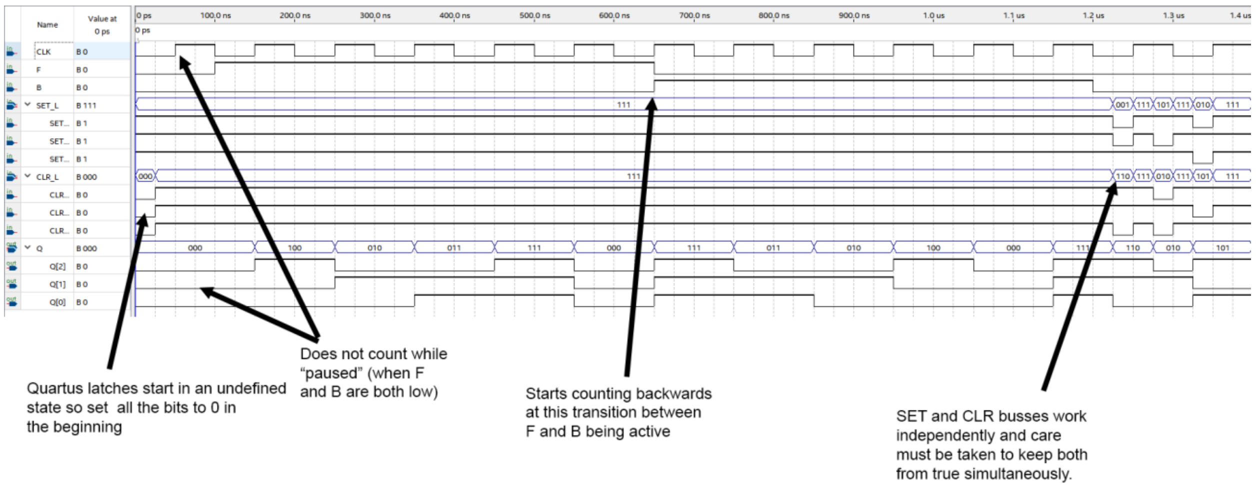
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Table 1 – XOR Truth Table

Annotations

All simulations you screenshot in this class must be annotated. This not only helps us grade your labs, but it more importantly helps you understand what you're doing and what you've made.

Here is an example of a good annotation:


Name
Value at 0 ps
CLK
F
B
SET_L
CLR_L
Q
Q[2]
Q[1]
Q[0]
Quartus latches start in an undefined state so set all the bits to 0 in the beginning
Does not count while "paused" (when F and B are both low)
Starts counting backwards at this transition between F and B being active
SET and CLR busses work independently and care must be taken to keep both from true simultaneously.

Final Notes

Whenever you change your circuit, you must *at minimum* perform a **fast compile** (Ctrl+K or the third play button) of your circuit to see the changes take effect in your simulation. If you change any of the inputs or outputs in your circuit, you should repeat **Step 3** of this tutorial again.

Quartus simulations are a good way to test your circuits, but they do not match reality perfectly. For example, they do not consider propagation delays or gate delays, as Quartus does not support timing simulations for the Intel MAX10 FPGA.

Additionally, latches and flip flops in Quartus will start in an "undefined/unknown" state if you use connect either the PRECLEAR or PRESET. In simulations that use gates or latches, you should always reset them at the very beginning of your simulation. You can see an example of this in the annotation above.

Conclusion

Simulations are an extremely important part of the circuit design process. You should always simulate before you flash your FPGA with your own designs to ensure that everything works as intended. This will be something you will be doing very often in this class, so make sure you know how to do it quickly and efficiently!