ECSE-4770 Computer Hardware Design: 74163 Quartus II Tutorial

Prepared By: Lawrence Fan

Introduction:

The purpose of this tutorial is to demonstrate how to use the 74163 4-bit counter in Quartus II simulations. In Quartus II, 7400-series logic is included in the default schematic symbol libraries under **others > maxplus2**. In this tutorial, you will be building a simple state machine designed to count up from zero to nine in binary and decimal. On the next clock cycle, the state machine will reset back to zero. It is assumed that you have completed and understood the material shown in the first CHD Quartus tutorial. This tutorial was prepared using the 13.0sp1 version of Quartus II Web Edition. Finally, here is an overview of the logic ICs that we will be using:

1. 7400: 2-Input NAND Gate
2. 7404: Inverter
3. 74154: 4-to-16 Binary Line Decoder
4. 74163: 4-Bit Counter

Tutorial:

1. Begin by creating a new project in Quartus. Remember that the Computer Hardware Design labs use the Altera DE2 board, which uses the **Cyclone II EP2C35F672C6** FPGA IC. Name the project “74163_Demo” and save it in a convenient location.
2. Create a new Block Diagram/Schematic File. Go to **File > Save As** and save the schematic file. Quartus should automatically rename the file from “Block1.bdf” to “74163_Demo.bdf”.
3. Select **Project > Set as Top-Level Entity** while “74163_Demo.bdf” is open to set it as the top-level entity.
4. Double-click on the schematic to open the symbol selection tool. In the “Libraries” menu, navigate to others > maxplus2 > 74163. Select this block and press “OK”. Place one 74163 onto your schematic diagram.

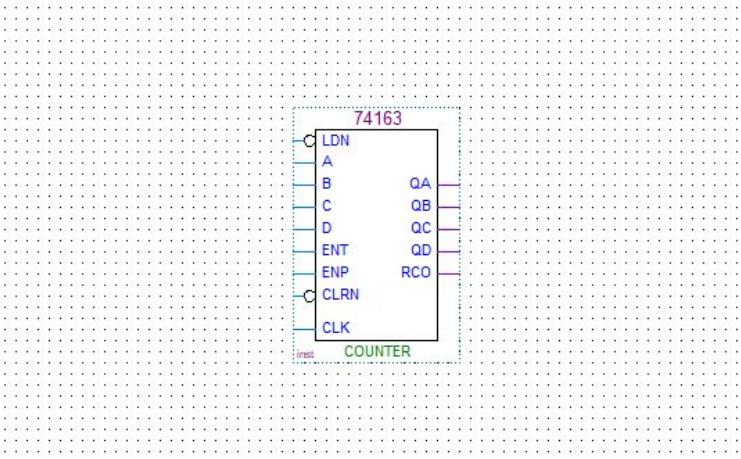

Figure 1: 74163 Counter

5. Some things to note about the 74163:
    a. The LDN line is an active-low input that causes the 74163 to load a 4-bit value from the lines A, B, C, D into the internal memory of the counter.
    b. A, B, C, D are the active-high input lines that the counter loads from.
    c. ENT and ENP are active-high enable lines.
    d. CLRN is the active-low clear line to zero the counter's value. We will not be using it in this tutorial, opting to load a zero value instead. The load value can be easily changed to start the counter at a non-zero value.
    e. CLK is a rising-edge triggered input that causes the counter to increment its internal value by one.
    f. QA, QB, QC, QD are active-high outputs representing the binary value of the counter's internal state.
    g. RCO is an active-high output that is useful for chaining multiple 74163s into a 4n-bit counter.
6. In the symbol selection tool, go to **primitives > other > gnd** to insert a ground. Place the ground near the input side of the 74163. Connect A, B, C, and D to the ground with wires using the **Orthogonal Node Tool**.

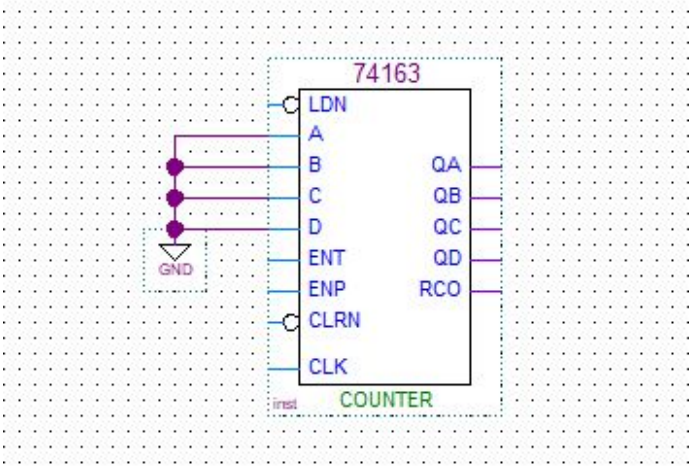

Figure 2: Load input lines pulled down to ground

7. In the symbol selection tool, insert a VCC symbol. It is located in the same folder as the GND symbol. Connect the ENT, ENP, and CLRN lines to the VCC symbol to tie them to the high rail of this circuit.

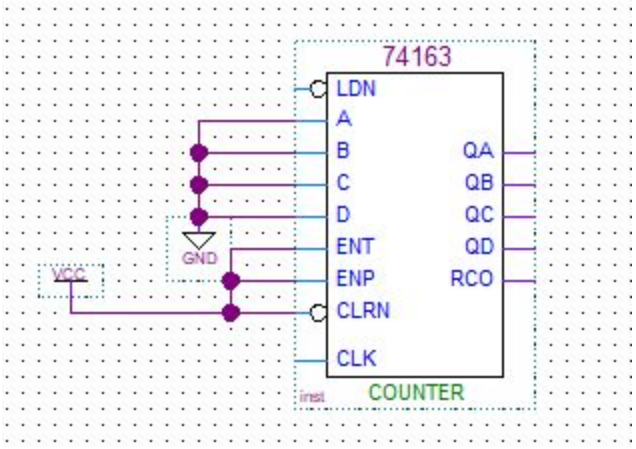

Figure 3: ENT, ENP, CLRN pulled high to VCC

8. In the symbol selection tool, navigate to **primitives > pin > input** to select the input pin symbol. Place an input near the CLK line of the 74163 and wire it to CLK. The pin will initially have a default name. Double-click on the name to select it and type in “CLK_IN”. Press the “**Enter**” key to rename the pin.

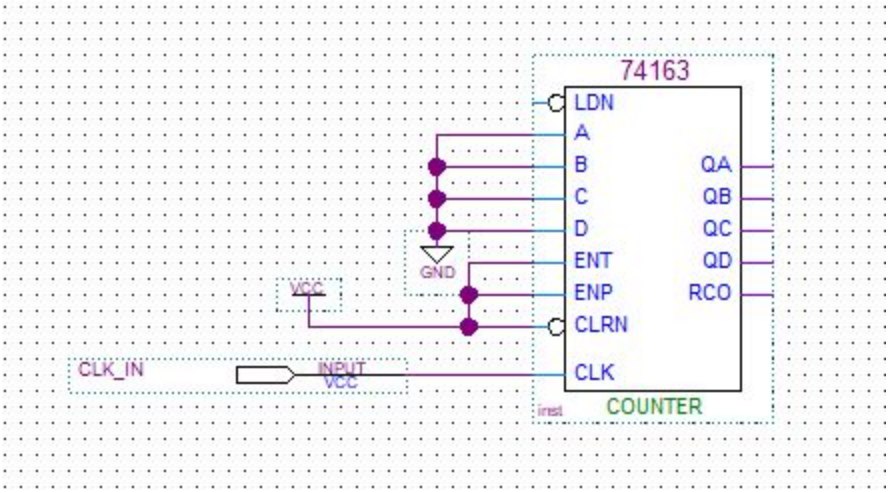

Figure 4: CLK_IN input pin attached to CLK input of 74163

9. We want our state machine to reset after it reaches State 9. Therefore, we should implement some logic to detect this state, which occurs at DCBA = 1001. The simplest way to do this is by using a 2-input AND gate connected to QD and QA. Because the load input is active-low, we need to invert the gate output, thereby making it a NAND gate. The 2-input NAND gate is the first 7400-series IC, and can be found at **others > maxplus2 > 7400**. Place a 7400 above the 74163 and connect its inputs to QD and QA. Connect its output to LDN. You will notice that the 7400, when first placed, is in an inconvenient orientation. Select the gate and press the “**Flip Horizontal**” button on the

toolbar above the schematic view.

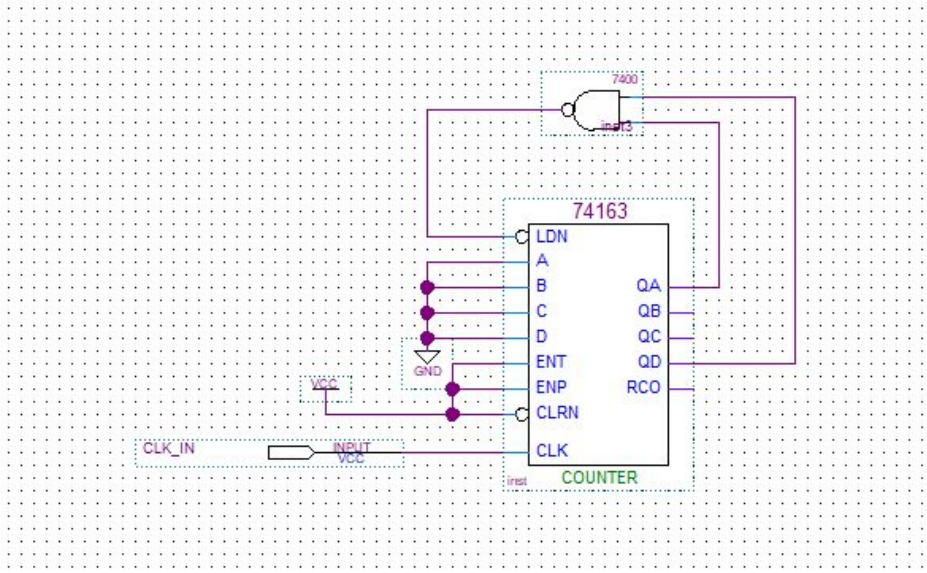

Figure 5: 2-input NAND gate connected to 74163

10. Now, we will insert the 4-to-16 decoder. Open the symbol selector and navigate to **others > maxplus2 > 74154**. Place a 74154 below and to the right of the 74163.

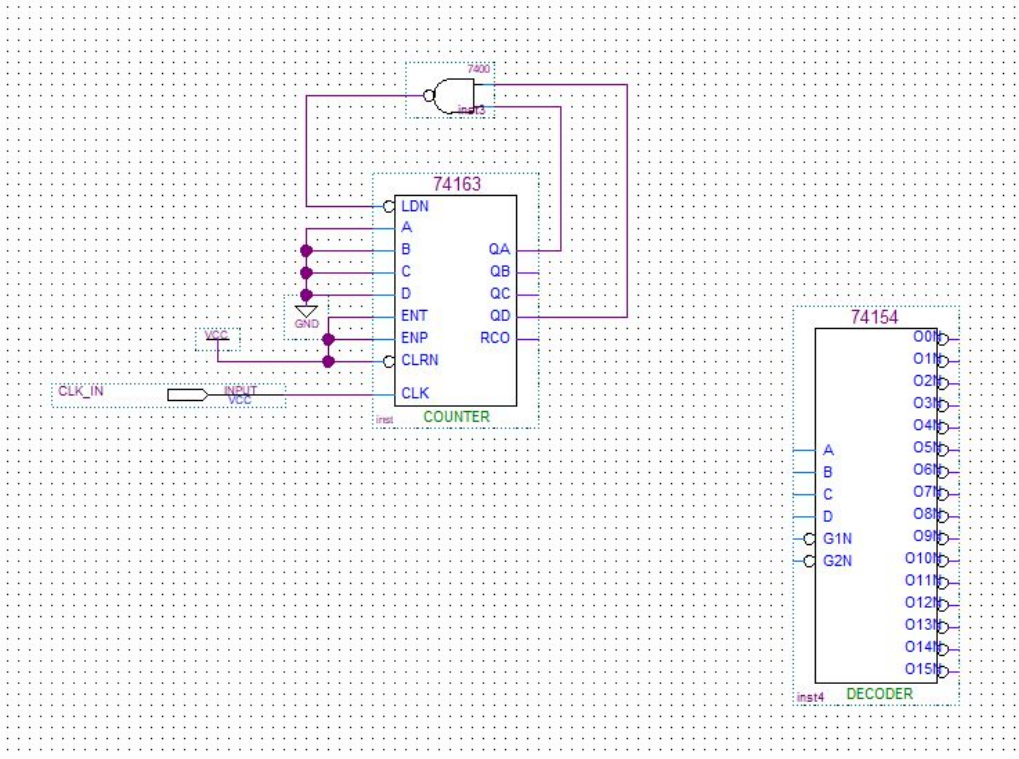

Figure 6: State machine and decoder

11. Some things to note about the 74154:
    a. The four binary inputs are A, B, C, D.
    b. G1N and G2N are both active-low enable lines.

c. O0-O16 are the sixteen active-low decimal output lines. These will be enabled one at a time to represent the decimal value of the binary input value.

12. Place another GND symbol close to G1N and G2N of the decoder. Connect G1N and G2N to the ground.

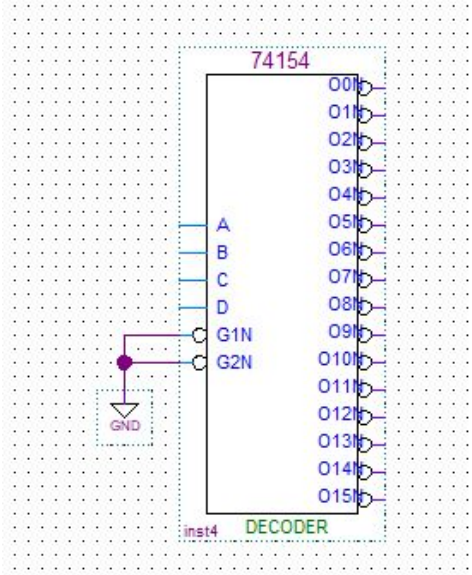

Figure 7: 4-to-16 decoder with enable lines tied low.

13. We can simplify and clean up our circuit by using busses to represent multi-bit or parallel signals. In this case, we should connect the 4-bit DCBA line from the 74163 to the 74154 using a bus. In the toolbar, select the **Orthogonal Bus Tool**. It is directly adjacent to the wire tool that we have previously been using.

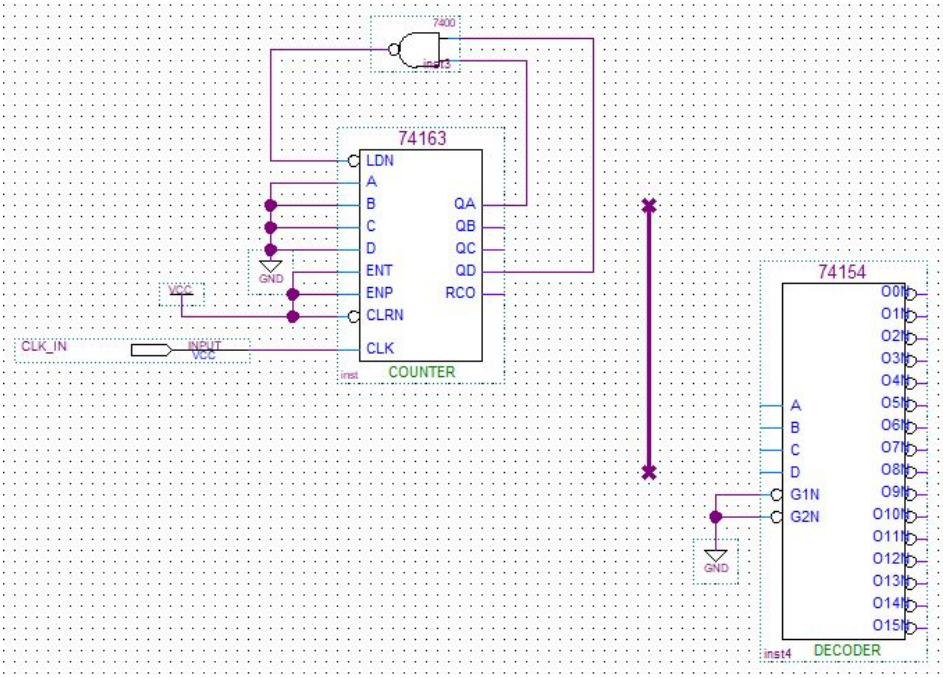

Figure 8: Multi-line bus between the counter and the decoder

14. Use the **Orthogonal Node Tool** to connect QA, QB, QC, and QD on the counter, as well as A, B, C, and D on the decoder, to the bus.

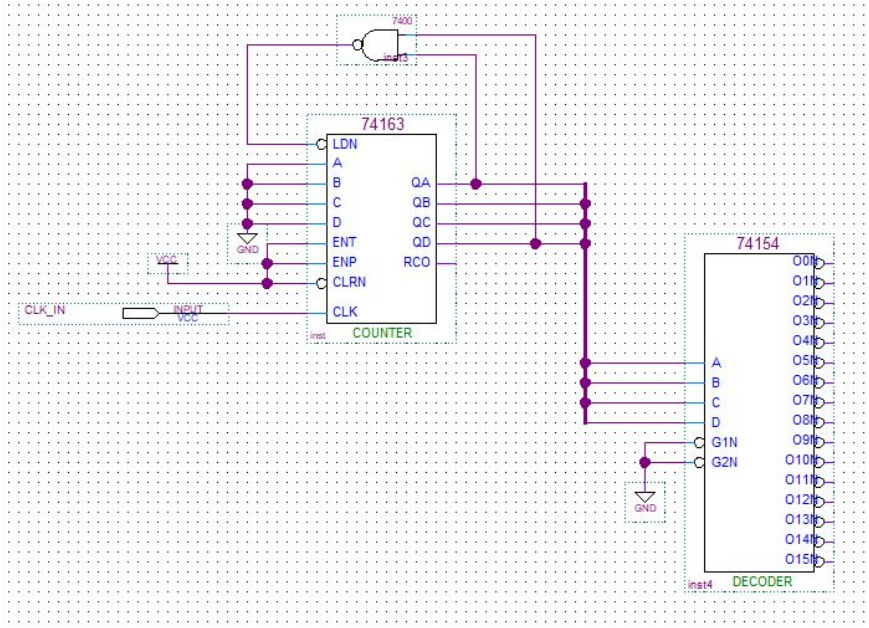

Figure 9: Multi-line bus connecting counter to decoder

15. In order to tell Quartus which wires should connect together, we need to name them. Name the line between QA and A “SM_A”, and repeat likewise for the rest of the lines. Name a wire by selecting it and right-clicking to bring up a menu. Select “**Properties**” to bring up the **Node Properties** menu. Enter a name into the “**Name**” text box and press “**OK**”.

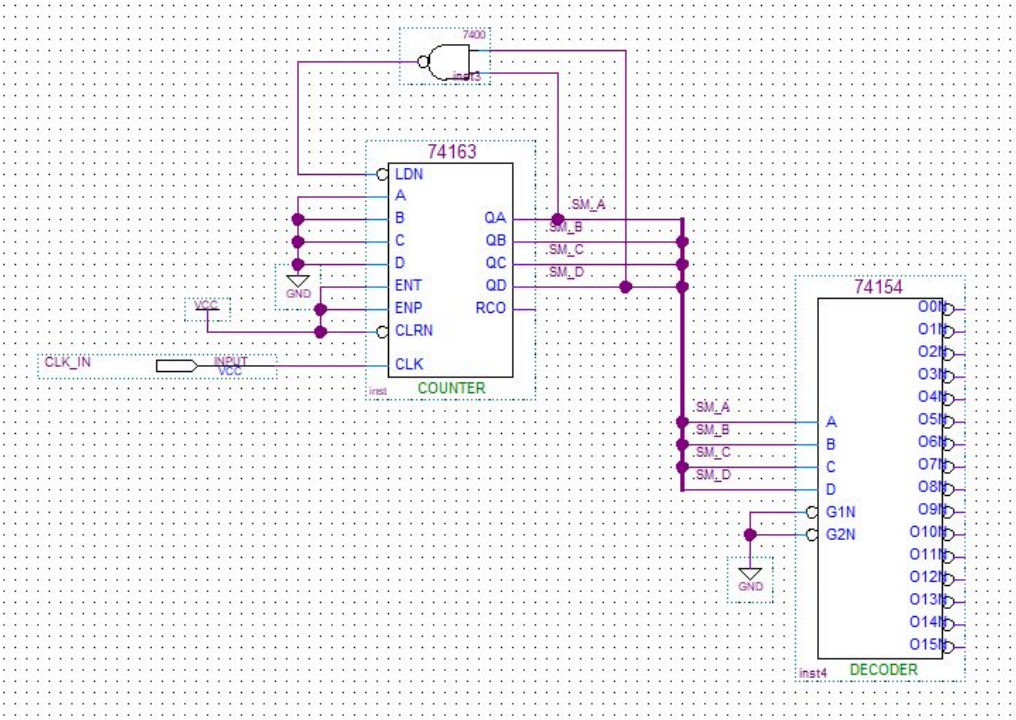

Figure 10: All input lines and output lines connected to bus have been named

16. The 74154 has active-low outputs, so we have to use inverters to get an active-high output. Select the 7404 inverter at **others > maxplus2 > 7404** and place ten inverters at the first ten outputs of the 74154. Connect them to O0-O9 of the 74154. There is a bug where the first inverter you place may not be named correctly. If it has the name “inst” instead of “instx” where x is a number, place it somewhere else and then place the rest of the inverters. Go back and delete the improperly-named inverter after everything else is wired.

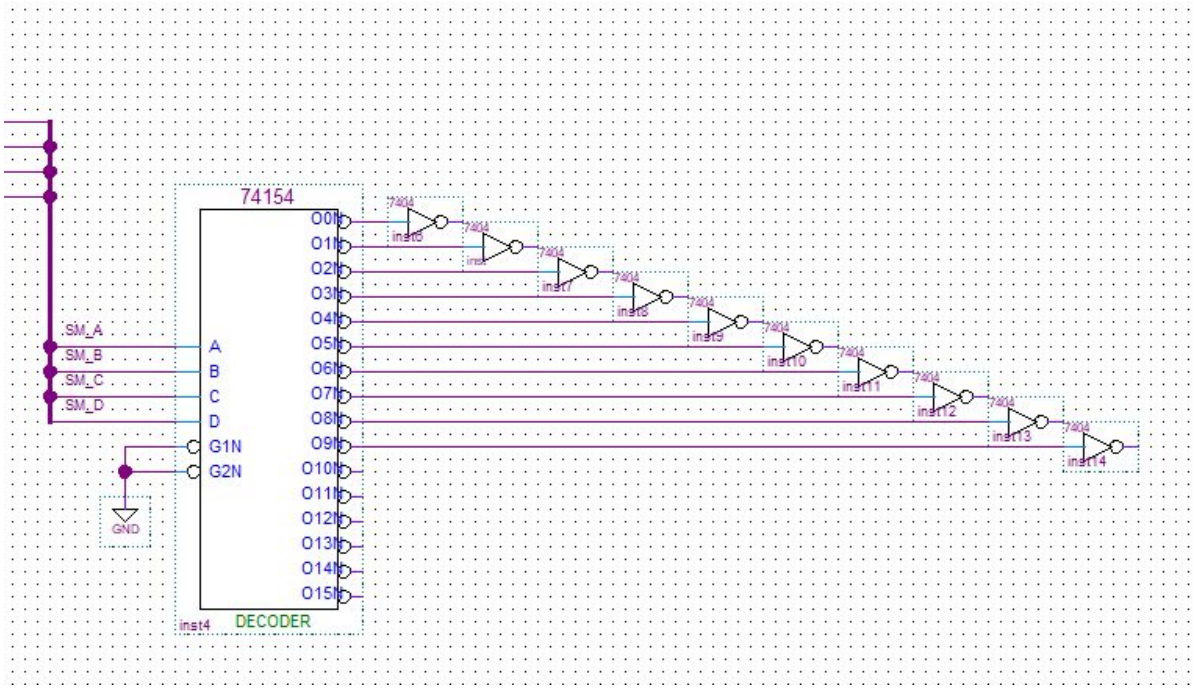

Figure 11: Ten 7404 inverters placed at the active-low outputs of the 74154 decoder. Note the inverter name bug at the output of O1N.

17. Navigate to **primitives > pin > output** in the symbol selection window to select an output pin. Place ten output pins near the inverters. Wire each inverter output to its corresponding output pin. Name each output pin. In this case, I named them “ZERO” through “NINE”.

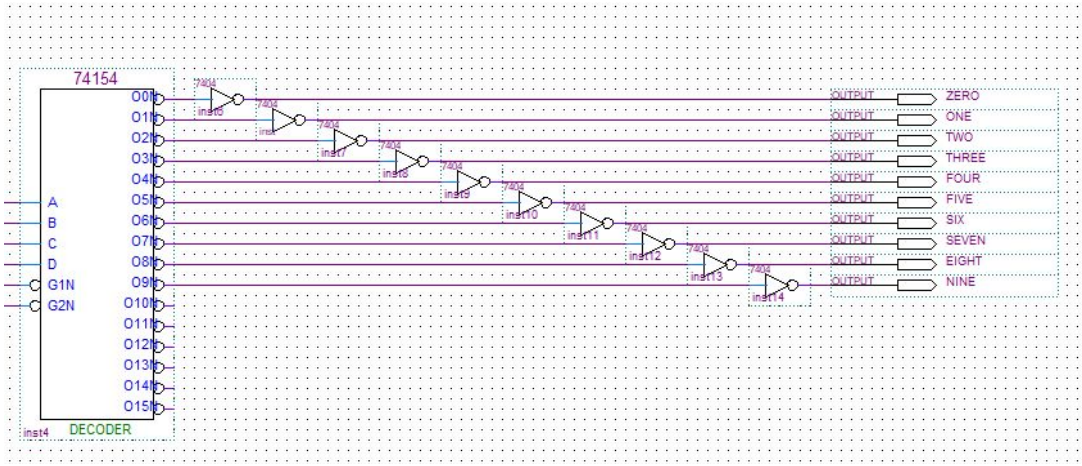

Figure 12: Ten named output pins connected to the inverted outputs of the 74154

18. Please check your schematic against this one:

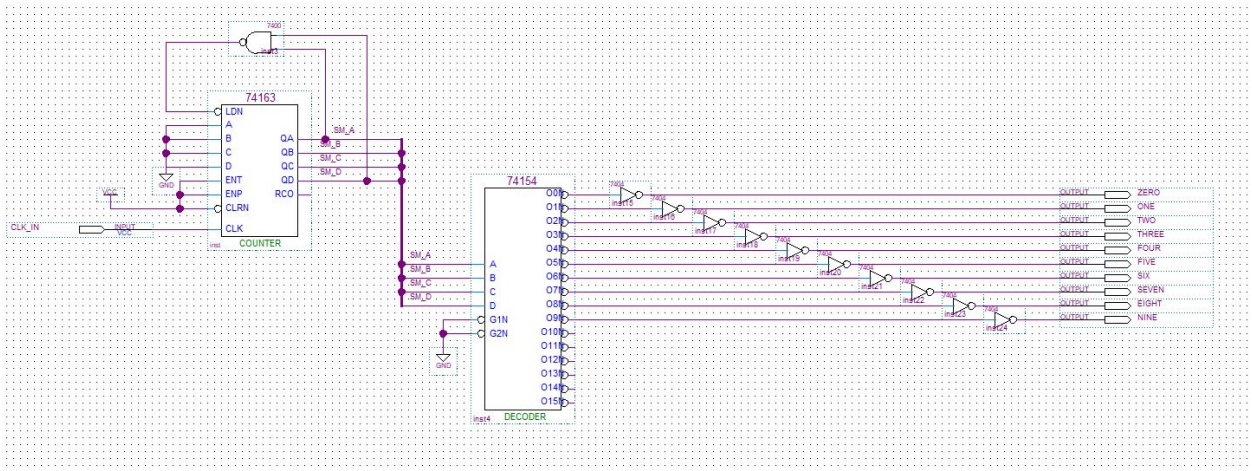

Figure 13: Final Schematic

19. Go to **Processing > Start Compilation** and compile your schematic.

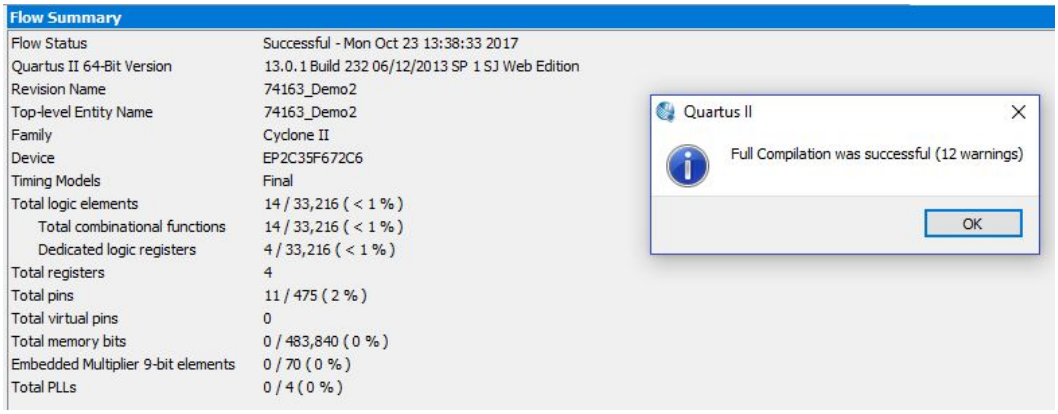

Figure 14: Output of successful compilation

20. Go to **File > New** and create a new **University Program VWF**. This option is under "**Verification/Debugging Files**". Right-click on the bar in the left side of the screen to bring up a menu. Select "**Insert Node or Bus. . .**". In the window that opens, select "**Node Finder. . .**" and switch to the Node Finder window. Click the "**List**" button to import all I/O nodes from the schematic and press the double chevron button "**>>**" to import all nodes. Press "**OK**" on the Node Finder and "**OK**" on the "**Insert Node or Bus**" to finish.

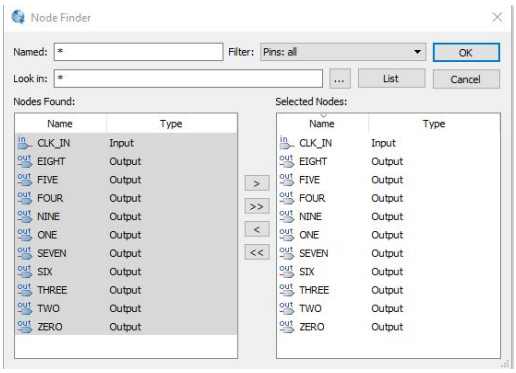

Figure 15: Importing all I/O nodes

21. The box on the left side of the window will have the nodes out of order. Click and drag the signals up and down to rearrange them. The solid line next to CLK_IN represents that it is a controllable input line. The hatched lines next to the other signals represents that they are uncalculated outputs.

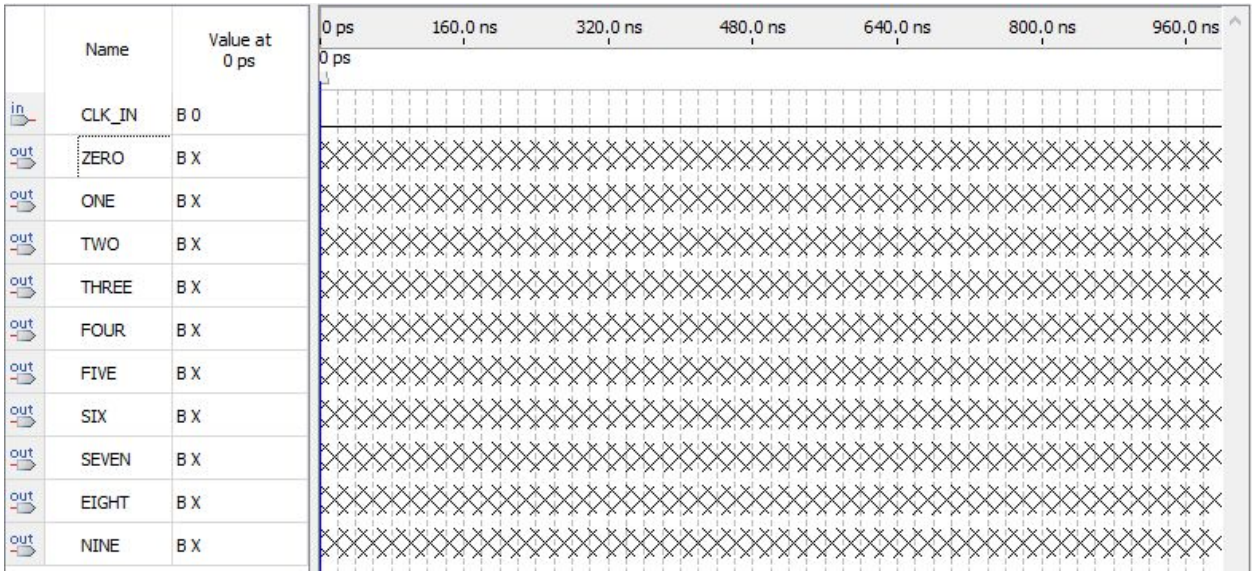

Figure 16: Timing Diagram

22. You can set input values by clicking and dragging to select a piece of a line and then pressing the ‘1’ or ‘0’ buttons to set the value of that section of the line. Make a simple square wave with ten pulses.

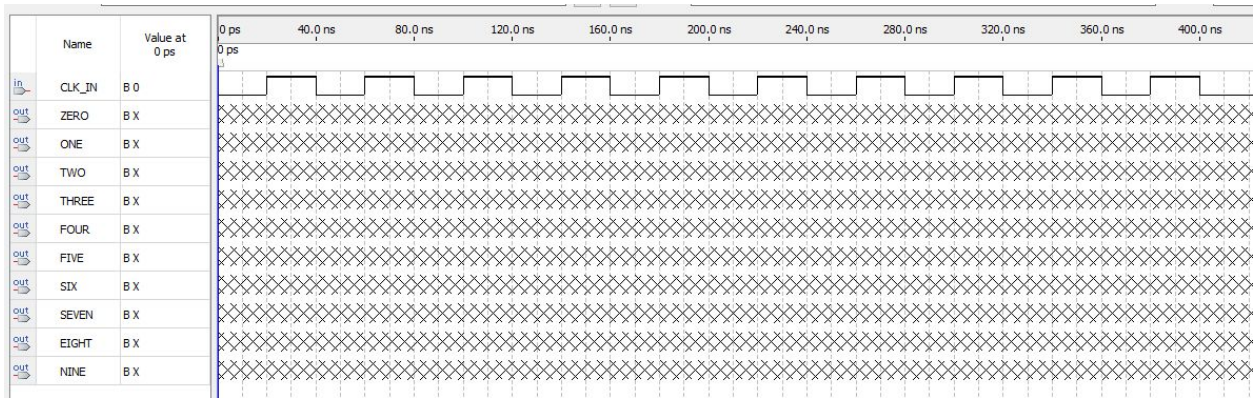

Figure 17: Timing Diagram with input set

23. Go to **Simulation > Options** and set the simulator to “**Quartus II Simulator**”. This will have to be done every time this waveform file is opened. The program will open a warning dialog box. Press “**OK**” to ignore it. Then, go to **Simulation > Run Functional Simulation** in order to simulate the circuit behavior. The program will ask you to save the waveform file. Then, the simulation results will open up in a separate window.

24. Check your results against the sample results:

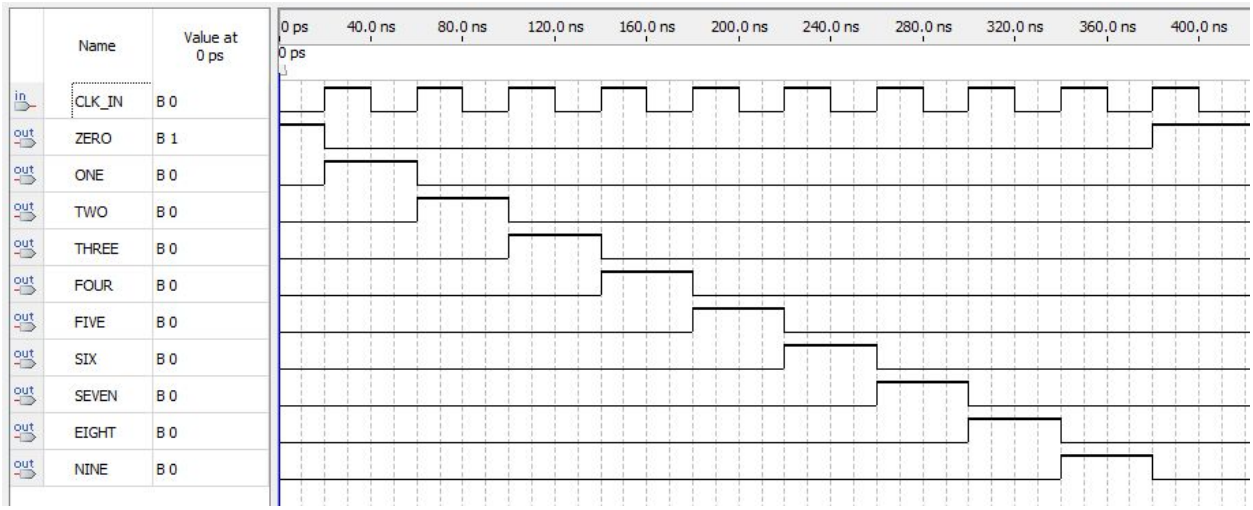

Figure 18: Computed Timing Diagram

Note that the circuit output is active-high and the input is rising-edge triggered as expected. The circuit starts in State 0, and on each rising edge of the input clock, it advances to the next state, until it reaches State 9. At that point, on the next rising edge, the state machine resets to State 0.

Miscellaneous Quartus II Tips:

a. Connecting pins using wires in Quartus II schematics requires superhuman precision. You will often make mistakes that lead to unconnected pins or extra wiring. When this happens, press **CTRL+Z** to undo the wire and then try again. When you finish wiring, carefully look over your schematic for the small ‘x’ symbol that represents an unconnected pin or wire.
b. Double-check that your selected device is the **EP2C35F672C6** in the **Cyclone II** family. If you select a different FPGA, Quartus will still compile and it may load to the DE2 board. However, it will not work.
c. During simulations, make sure that the “**Quartus II Simulator**” is selected. The program will reset this to “ModelSim” every time you close and reopen the waveform file.
d. If you are using the MalwareBytes Antivirus software on your computer, add “C:\altera” to the antivirus exclusion list. Otherwise, the AV’s heuristic detection engine may flag Quartus as malware and corrupt your installation.