EE 1315 DIGITAL LOGIC LAB

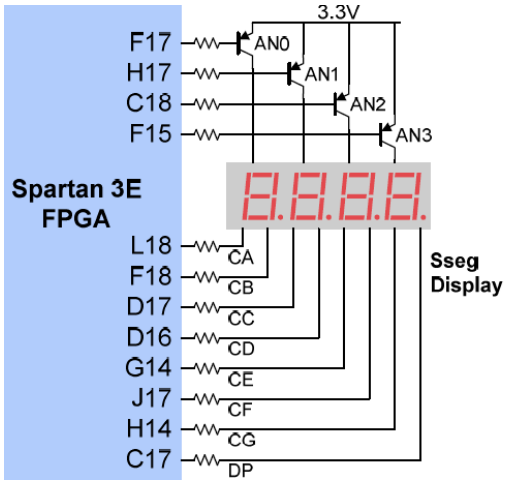
EE Dept, UMD

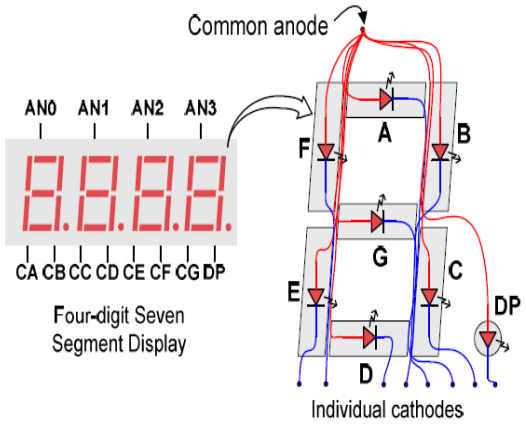
EXPERIMENT # 11: Electric Stop Watch

The purpose of this experiment is to learn how to create a schematic module in Xilinx, learn about the usages of counters, and to interface with the seven segment display on the Nexys board. In this lab, you will emulate a sort of an electric timer that counts numbers 0, 1, 2, ..., 9999, and then when you press a switch, the number stops creating the effect of a timer. For counters, you will use four BCD counters provided in the Xilinx ISE (the CD4CE symbols).

Part 1: Seven Segment Decoder

To display the numbers on the seven segment displays present on the Nexys board, the binary numbers output by the counters will need to be converted to a seven segment code prior to display on the seven segment modules. The picture below shows the configuration of the seven segment display on the Nexys board.





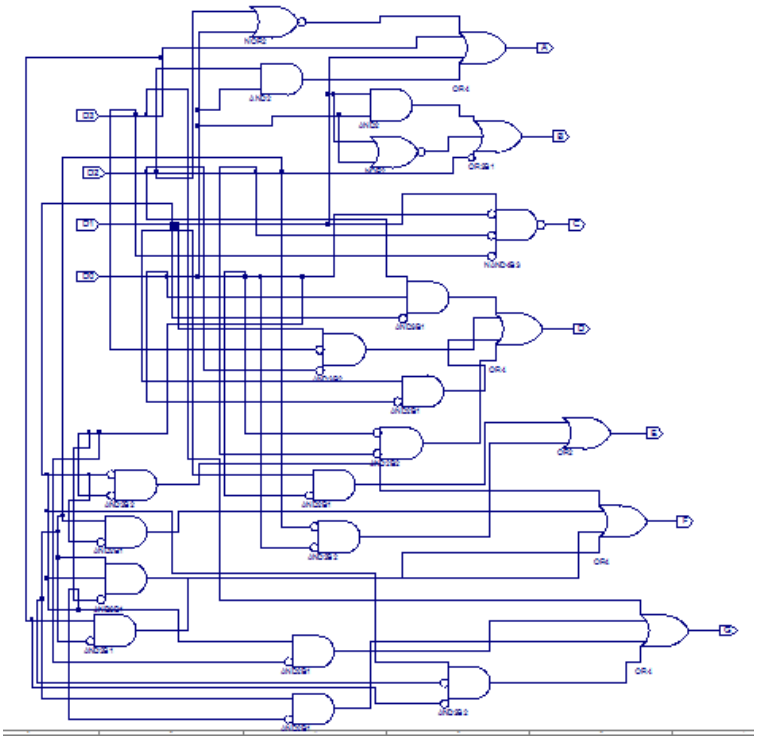
As you can see the binary number 0 needs to light the LEDs A,B,C,D,E, and F while not lighting LED G(lighting G would make it an 8). This can be done by purely combinatorial logic. This will be the first task of this lab. The logic table is provided below to help you in your design. Please note that since the counters only count 0-9, the four digit binary numbers above 9 are all don't cares(X).

Decimal	D3=MSB	D2	D1	D0=LSB	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	1	1	0	0
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

Using the above table, find equations for A,B,C,D,E,F, and G as functions of D3,D2,D1,and D0. You may use any **logic gates** that are available in the Xilinx ISE library (some functions may have up to 4 min-terms so a 3 or 4 input gate may be useful). Implement these functions in a schematic named s_seg_dec(or some other descriptive name) and then test the function of your circuit on the Nexys board using the following Scalar Port assignments.

Scalar Port	D3	D2	D1	D0	A	B	C	D	E	F	G
Site	K17	K18	H18	G18	F4	P15	E17	K14	K15	J15	J14

If the outputs are correct, this circuit is ready to decode a single BCD (binary coded Decimal) number. A schematic of a working circuit is shown below. Although the schematic is given, it may be difficult (or impossible) to use it to create your circuit. Finding the correct functions for each output is recommended (and will prove easier). Note the use of gates with inverting inputs to lessen the number of gates placed on the schematic.

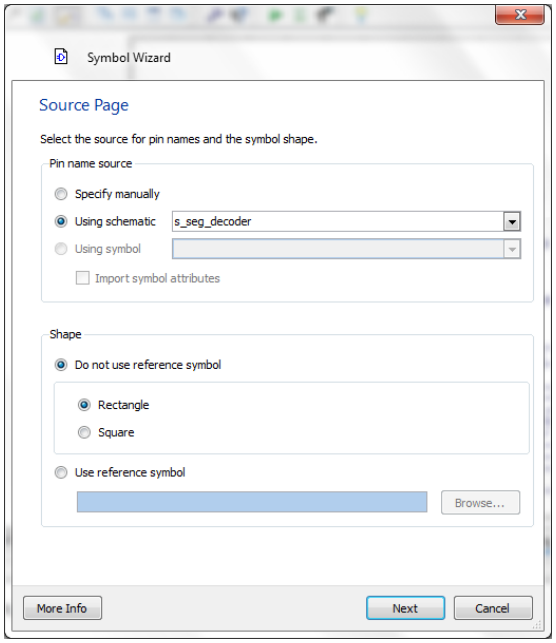


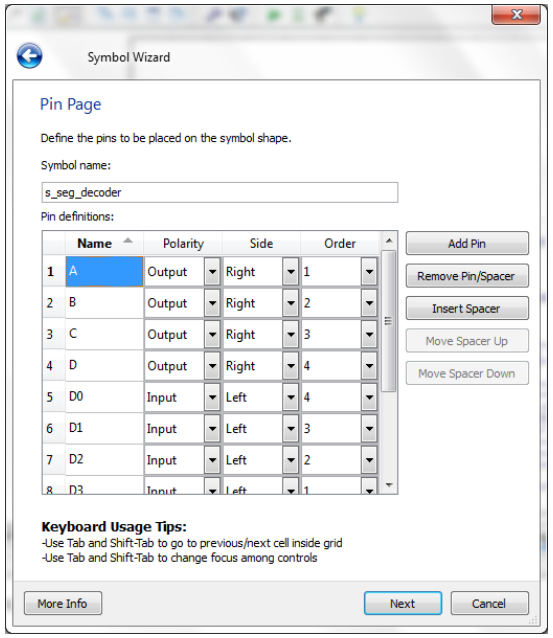
BCD to 7-segment decoder circuit

Now that you have developed and tested a circuit that will decode a single 4-bit BCD number you can display one of the numbers from your counter circuit to the seven segment display. To display four BCD numbers, you will need to create four of these circuits. Creating three more of these diagrams would be very tedious, but luckily Xilinx ISE allows you to make this circuit into a schematic module that can be dragged and dropped onto your schematic. We will go through these steps now (BE SURE THE CIRCUIT WORKS CORRECTLY BEFORE CONTINUING OR ALL SUBSEQUENT CIRCUITS WILL MALFUNCTION).

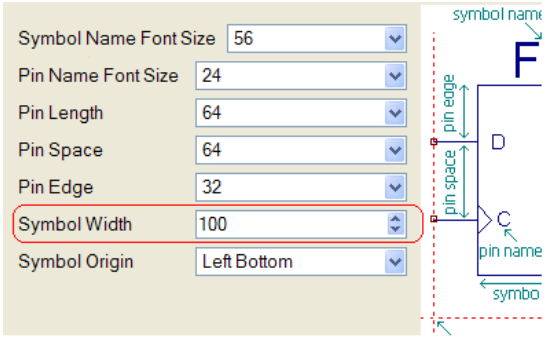
Steps to create decoder symbol

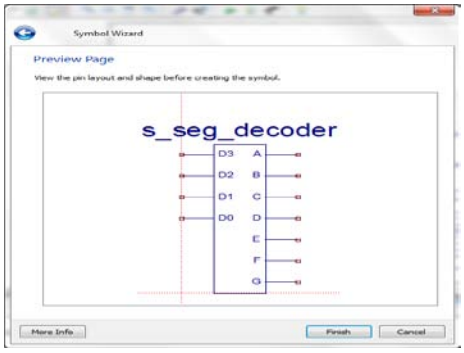
While viewing your schematic select **Tools->Symbol Wizard** from the drop down menu at the top of the screen. Then select **Using Schematic** and **Rectangle** as seen below. Click **Next**. The **Pin Page** will then appear. Make sure all the outputs (A – G) are set to **Output** for polarity and **Right** for side. Also be sure that the inputs (D3-D0) are set to **Input** for polarity and **Left** for side. This can be seen below. Once the parameters are set, click **Next**.





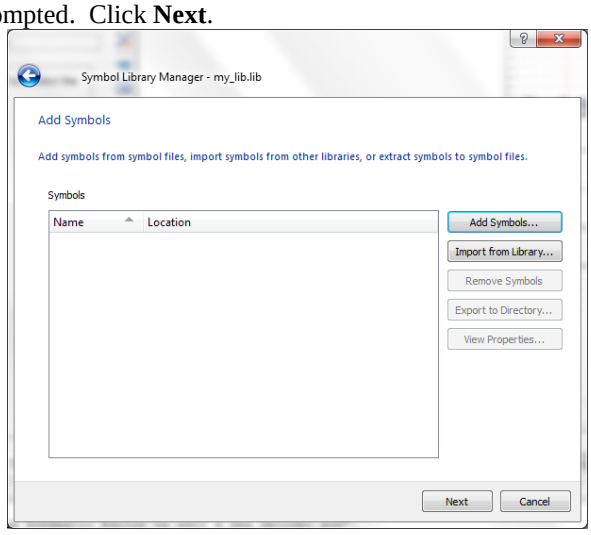
In the next window that appears, set the symbol width to 100 and click **Next**. You will then see a screen that gives a preview of your newly made symbol. Click **Finish**.

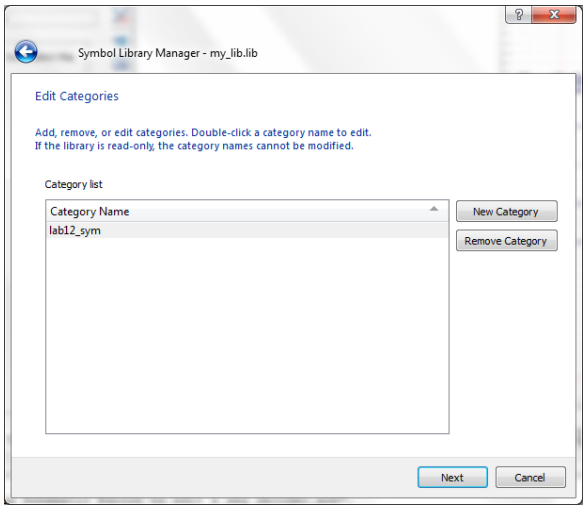




Next, select **Tools->Symbol Library Manager**. In the window that appears, click the **Browse** button on the **Library File Name**. Browse to your project folder location (somewhere on your flash drive named Lab12) and type in "mylib.lib" as the library name. Click on the second browse button and again choose your Lab12 folder.

In the next window (seen below) choose **Add Symbols**. A window will appear; select **s_seg_decoder.sym** and click **Open**. In the following window (Also seen below right) choose **New Category** and type "lab12_sym" when prompted. Click **Next**.



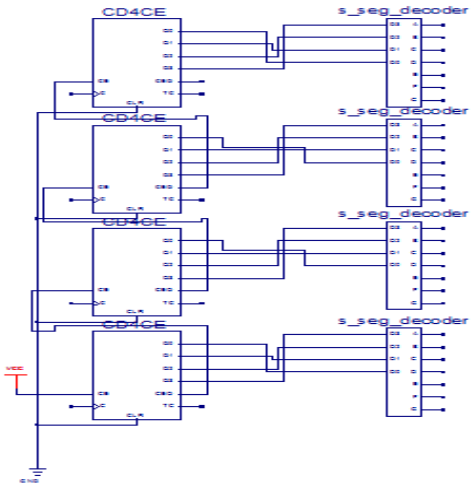


In the next window, click on **lab12_sym** in the right hand column and then click **Add All** then click **Next**. Then click **Check All**, **Next,** and **Finish**. Your decoder is now ready for you to place on schematics! Place one on a new schematic and test the module's functionality.

(This is a good place to end week one)

Part 2: Creating a Stop Watch

Create a new schematic by navigating to the **Design** window and **right-clicking** on the chip name. Choose **Add New Source** and add a new schematic source to your project with the name of counter. After adding the new schematic, navigate to the **Design** window and **right click** on the newly created counter module. Select **Set As Top Module** and click **Yes**. Now the stopwatch will be created using the CD4CE counters supplied in the Xilinx ISE library. These counters will count up from 0 to 9 in binary and then reset. To make a four digit counter, **right click** on the symbol after you have added it to the schematic, choose **Object Properties** and click **Symbol Info**. The data sheet provided explains how to cascade counters to create a base ten four digit counter. After placing the four counters, s_seg_secoders, and making connections your schematic should look similar to that seen below. Notice that the CE input of the LSB is connected to VCC (always on) and the rest are connected to the CEO output of the module below it. This is called cascading counters.

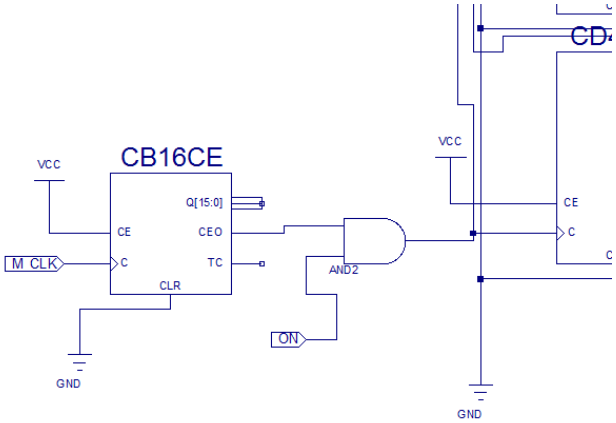


There are still some connections that will need to be made. The BCD counters need a clock input, a state machine to control the anode output will need to be created, and the outputs to the seven segment display need to be multiplexed. First the clocking circuit will be created.

Clocking Circuit

This circuit will be the clock signal for the DBC counters as well as for the state machine that controls the anodes to the seven segment display. The internal oscillator on the Nexys board runs at 50 MHz. This is far too fast for us to see the changes in our stopwatch take place. Instead we will use a clock that operates at approximately 1 KHz to run our circuit. This value will allow us to have a four digit stopwatch that will count milliseconds. This value is also a good speed for refreshing the seven segment display which will be done by our state machine in the next section.

To change a 50 MHz clock to a 1KHz clock, a counter circuit will be used. Much like the cascaded BCD counters, the CEO output of a counter will be used to clock ALL of the BCD counters. To obtain the exact value of 1KHz, a counter that overflows after 50,000 clock cycles would be needed. If a binary counter with 16 bits is used, the overflow will occur after 65,536 clock cycles. This will give a clock that will count at a frequency of 763 Hz, which will suffice for this lab exercise. Place a CB16CE symbol on your schematic. Connect the CEO to an AND gate and the output of the AND gate to the clock inputs of your BCD counters. The other input of the AND gate should be an I/O port called START (this will control the start/stop of your clock). Also connect an input port to the clock input of the CB16CE and name it M_CLK (this will be the 50 MHz internal oscillator), connect the CLR input to ground and connect VCC to the CE input. Other connections can be left unconnected. This can be seen below.

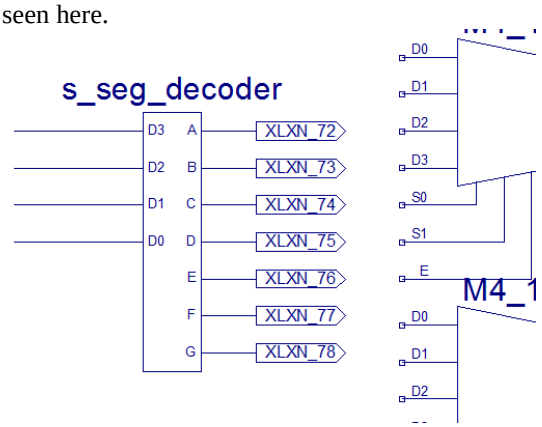


Multiplexing the Seven Segment Display

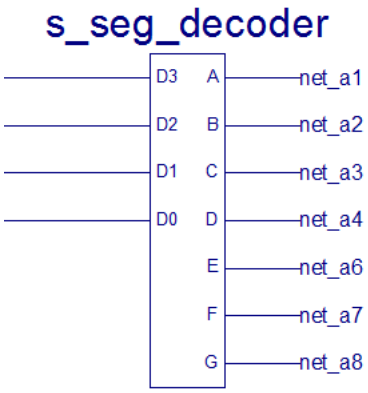
Now that there is a clock with the desired frequency available to you, the state machine that controls the display of the numbers will be developed. The seven segment display can only show one digit at a time. For this reason, a single number is shown for a brief period of time and then the next number is shown and so on. This happens very quickly, and to the human eye it appears that all the digits are being shown at the same time. To control this, a state machine will need to be developed to output the correct digit and the correct code to the anodes of the seven segment display.

First the outputs of the seven segment decoders need to be multiplexed. This can be done with seven 4 to 1 MUX (mux4_1e in the Xilinx library). Each of the mux will control a single segment of the display. Each output of the mux will have an output port. Place the seven mux. When connecting from output ports on created modules, Xilinx ISE requires a net name for each connection. To do this, follow these steps.

1) Add I/O ports to the outputs as seen here.

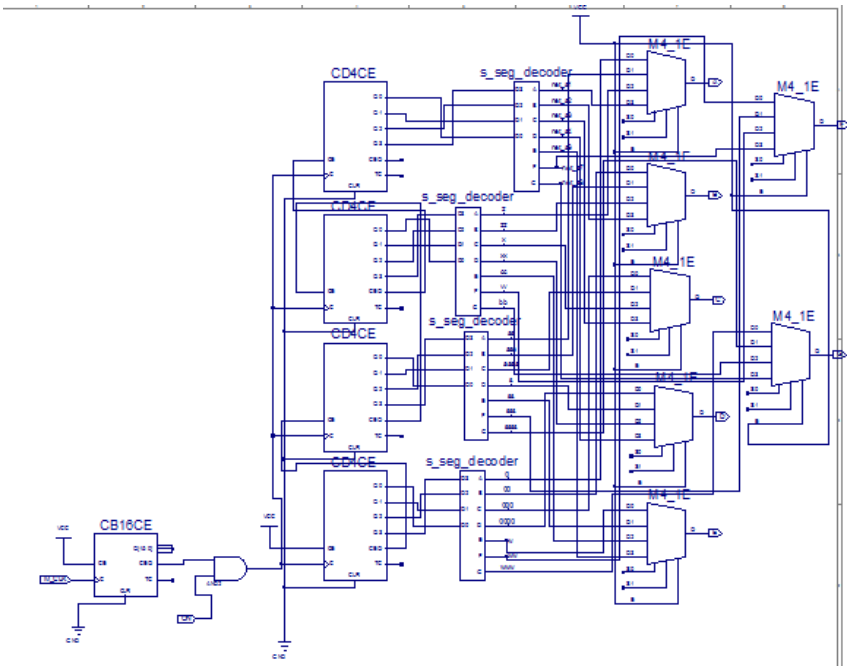


2) Rename the I/O port to something like Net_a1. Then delete ONLY the I/O port. The result is shown below:



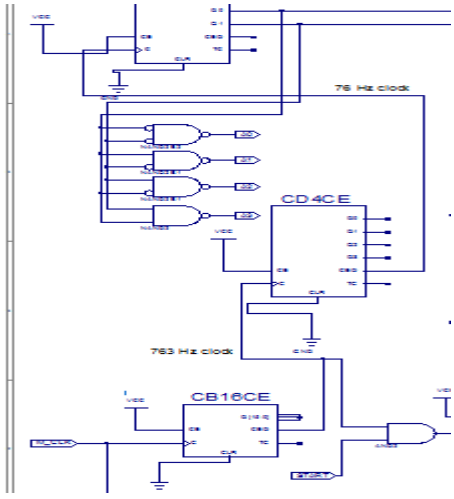
3) Connections can now be made using the **Add Wire** tool. If a red square appears on your connections, it indicates there it is not properly connected. **Warning:** If you move your s_seg_decoder symbol, the named net will **NOT** move with it, causing the connection to be broken.

Connect the all of the A outputs of the s_seg_decoder to a single mux. That mux should have an I/O port added named A. Be sure to put the Most Significant Digit (MSD) on the D3 connection of the mux and the LSD on the D0 connection. Make sure to properly connect the enable pin. Leave the selector bits unattached for now. The schematic should look similar to that shown below.

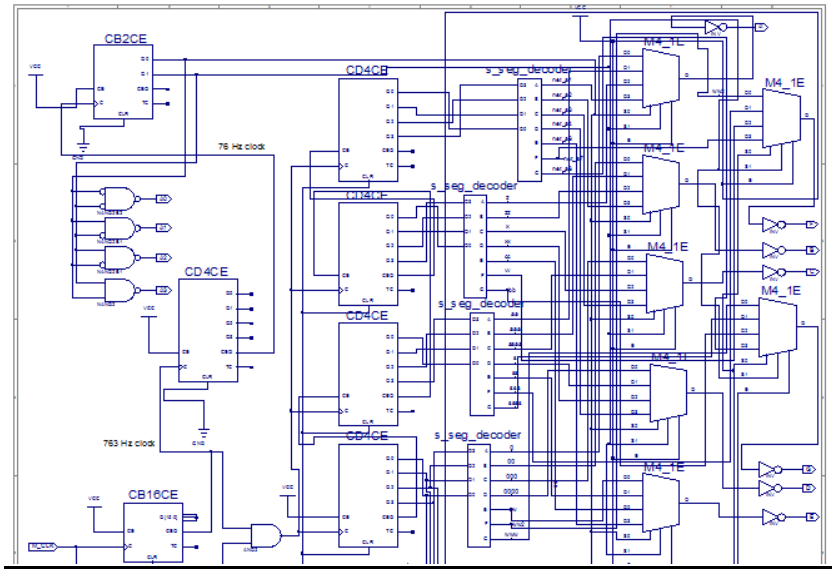


State Machine Control

Now that all the supporting logic has been completed, a state machine must be made to control the circuit. This will be a simple counter that will control two aspects of the circuit. First it will send the proper four bit code to the cathodes of the seven segment display. Secondly, it will select the proper selection bits for the seven mux in the circuit. A state machine that counts 00..01..10..11..00..01….. will accomplish this task, but a counter could also be used to simplify the design. Connect the 16 bit binary counter CEO output to the clock input of your state machine(or counter) and connect CE and CLR inputs as needed. **Make sure that the clock input to the state machine is taken from BEFORE the AND gate with the START switch!** The 763 Hz signal is too fast for this, so this must be further divided to clock your state machine at a proper frequency. 60 Hz is a good frequency to refresh the seven segment display, so dividing the 763 Hz clock by 10 would result in a good refresh rate. By running the input to the AND gate through another CD4CE counter and properly connecting it, the result will be a 76 Hz clock signal to your state machine. This can be seen below.



The selector bits for the mux can be directly connected to the state machine (counter), but some logic will be needed for the outputs to the anodes of the circuit. When a number is being displayed, the corresponding anode must be a logic '0' and the rest must be logic '1's. For example if state '00' is the LSD of your counter, the mux should have selection bits of '00' and the anodes should be '1110' (MSD to LSD). Next state would require the anodes to be '1101' (again MSD to LSD). This logic should not be too difficult and can be completed with a single gate for each anode. Make output ports A0, A1, A2, and A3 for the anodes of the seven segment display. When you are finished, your schematic may look something similar to this.



Finally use the following table to set your Scalar Ports, generate your programming file and test your circuit. Show your TA after it is completed.

Scalar Port	START	A0	A1	A2	A3	A	B	C	D	E	F	G	M_CLK
Site	G18	F17	H17	C18	F15	L18	F18	D17	D16	G14	J17	H14	B8

--- Congratulations!!! There will be no more labs in this semester. You have successfully completed the last ECE 1315 Lab, which would be one of the milestones in learning the digital world. Hope you have enjoyed the design experiences of digital logic circuits, and wishing you the best and continuous successes in your future. --- Prof. Taek Kwon

Experiment #11 Results

(Print this sheet and bring to the lab session)

Your Name: _____

Witnessed by Instructor or TA: _____

Date: _____

Check off the lab by showing the final circuit to the TA.