Boise State University
Electrical and Computer Engineering Department

EE230L: Digital Systems Laboratory
Spring 2011

Final Project:
Variable Sequence Detector and Display Module

Team #: 10
Team Members**:**
Sandy Ferguson
James Kress

Report Due: 5/10/11

Lab Objective:

The objective of the final was to create a module that would detect when a parent remote is calling on it, and display data on the seven-segment display.

Equipment:

Prototyping Kit

- FPGA

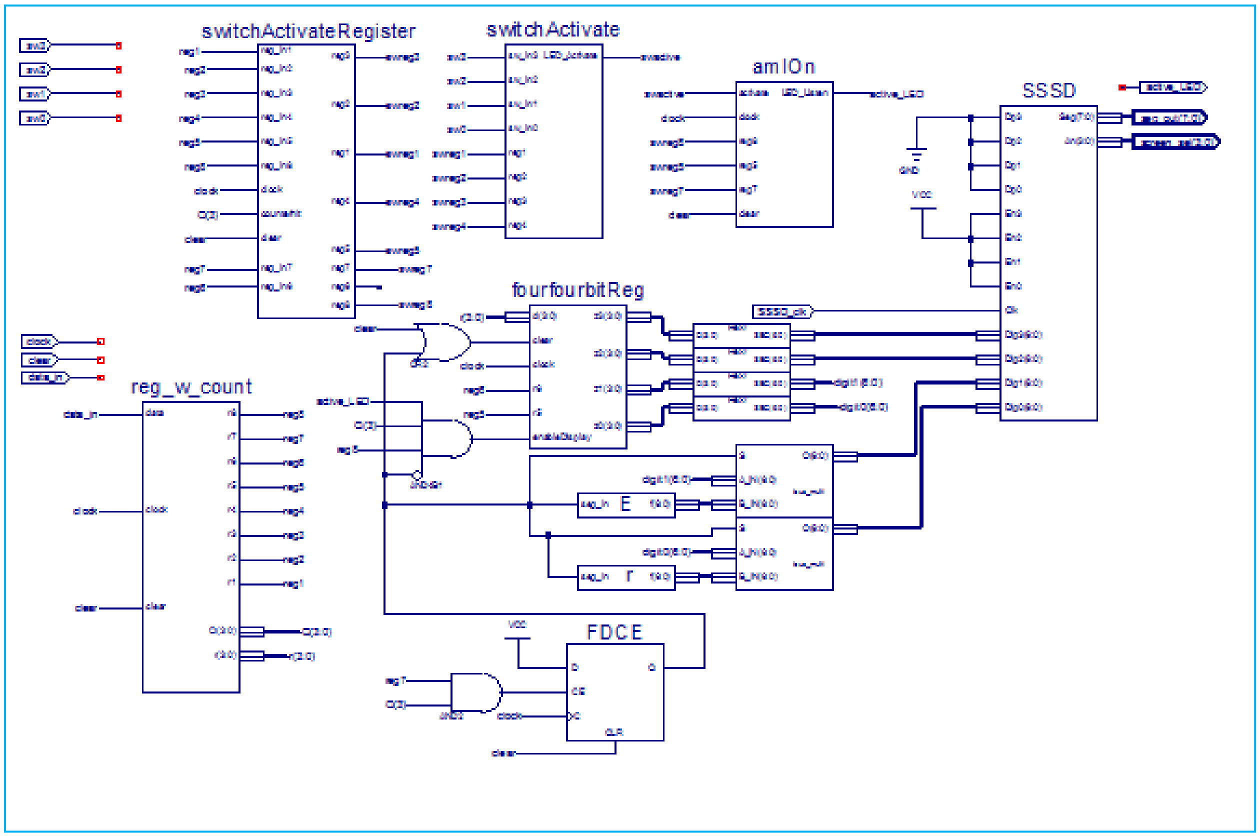
Files

- final.sch
- SSSD.vhd
- Hex7.sch
- amIOn.sch
- E.sch
- r.sch
- fourbitReg.sch
- finalCounter.sch
- eightBitReg.sch
- bus_multi.sch
- fourfourbitReg.sch
- reg_w_count.sch
- switchActivate.sch
- switchActivateRegister.sch

Design Description/Decisions

The design of this project was done is multiple stages. First, a register with a counter was designed, tested, and implemented. Next, a new register was created for use with the switchAcitvate.sch module. What this does is checks to see if the module is being addressed by checking the switch and incoming values. Next, an amIOn.sch module was created. This module activates when the board is being addressed by the Base Station, and checks to see if the incoming bits tell it to turn the module on or off. Next, a module with four four-bit registers was created to store the data output values for display on the Hex7 modules. Lastly, an error checker was implemented to check for bad data input values. The overall schematic (See Figure 1: Final Schematic), has a few logic circuits on the top level, but this is used just to decide on when to clear and activate the error modules.

Figure 1: Final Schematic



Register

The design of the register and counter were very simple. The overall schematic (see Figure 2: register with Counter), contains the symbols for both the register and the counter. The counter (See Figure 3: Counter) we used was a library counter that had the ability to be preset to a '1' upon activation. This was instrumental in keeping the counts accurate and reading in the right amounts of data for each cycle. As it stands, it runs from one to eight, resets to one, and does the same thing again. This enables the register (See Figure 4: 8-Bit Register) to read and latch one bit of data each clock cycle, and then be cleared for the next round of data input on that same eighth clock cycle.

Figure 2: Register with Counter

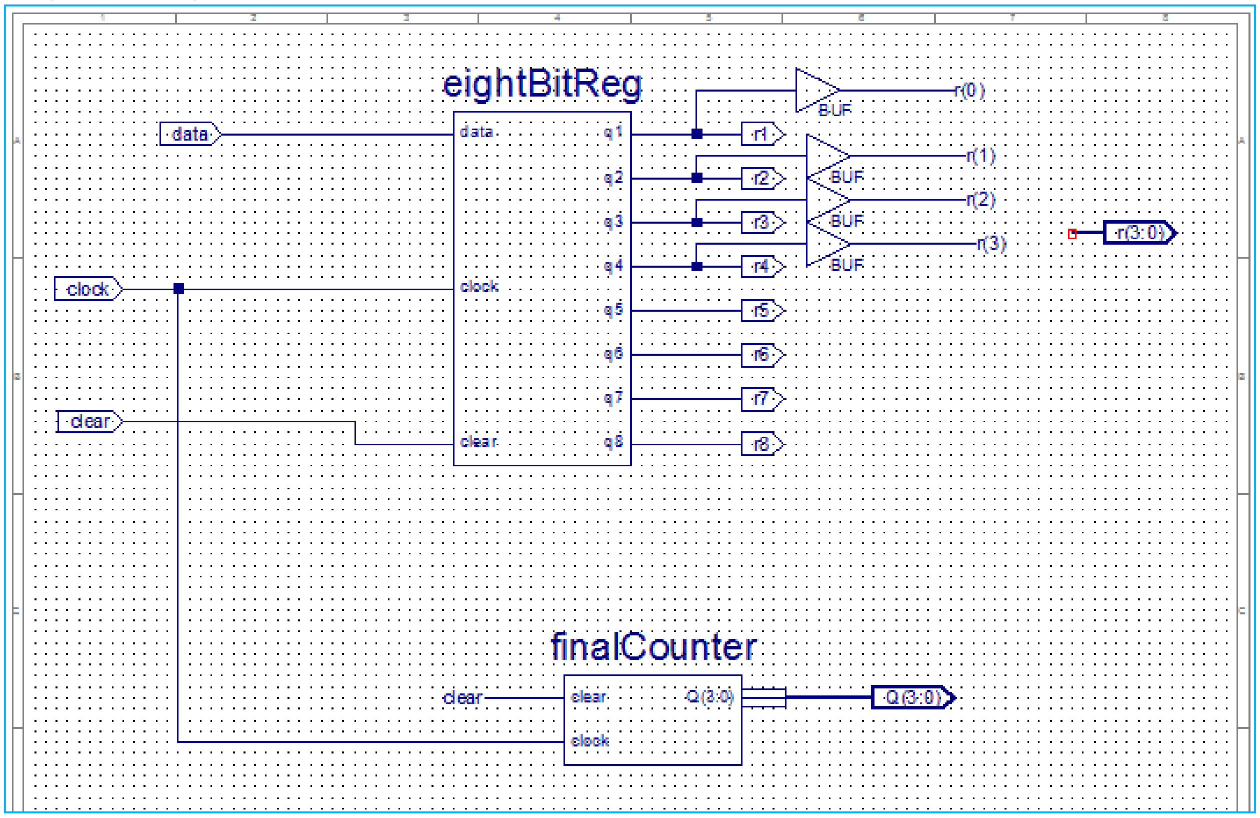


Figure 3: Counter

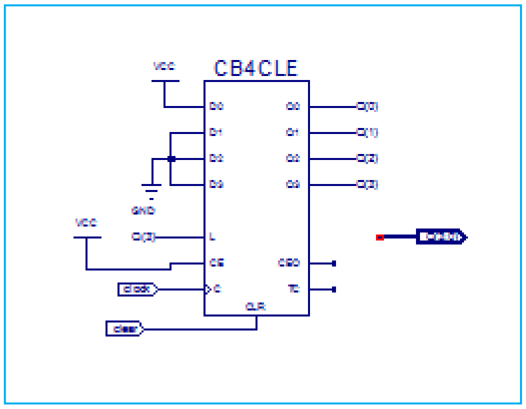
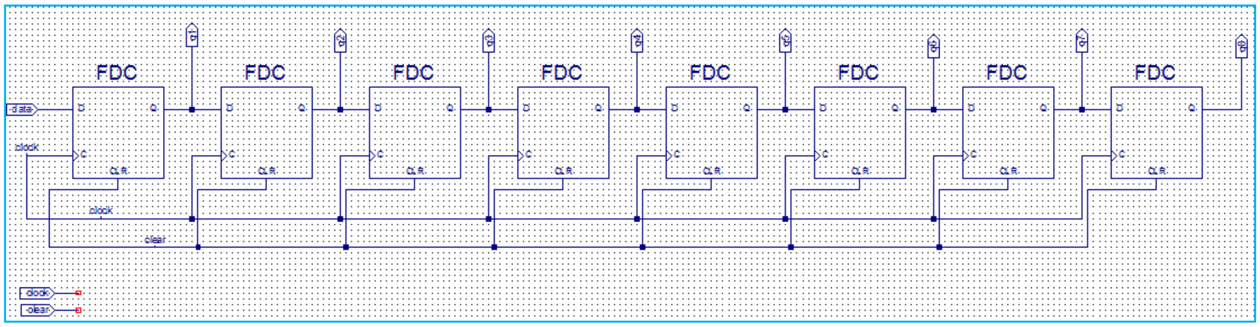
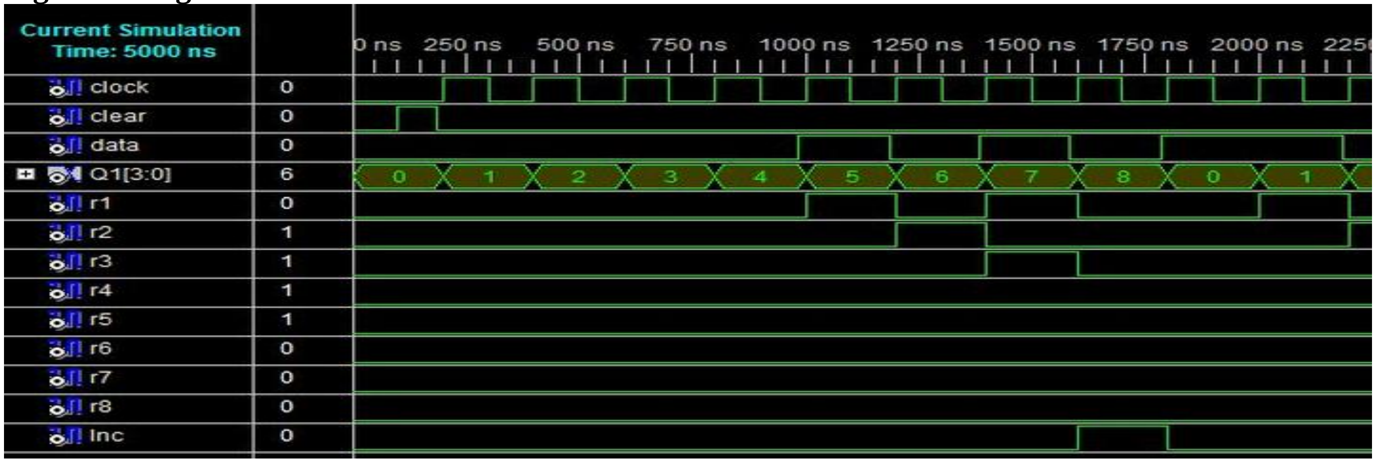


Figure 4: 8 bit Register



These designs worked great in combination with each other (See Figure 5: Register with Counter Test Bench). As is shown, the clock counts form one to eight, and latches one bit of data into the register on each clock cycle.

Figure 5: Register with Counter Test Bench



Switch Activation and Recognition

To enable the circuit to save and use the input data, a new register was created to store the values for the next eight clock cycles (See Figure 6: Switch Activation Register). This register is used by the Switch Activate Module (See Figure 7: Switch Activate Schematic). This module takes in the values from the switch register and decides if it is being addressed based on the inputs from the four switches, and the four register values that are addressed to the switches. If four ones are put into the registers as data, the modules will listen to the rest of the data in the register because this is the case in which all modules should listen. In all other cases, it looks for the case in which the inputs match the switches positions.

Figure 6: Switch Activation Register

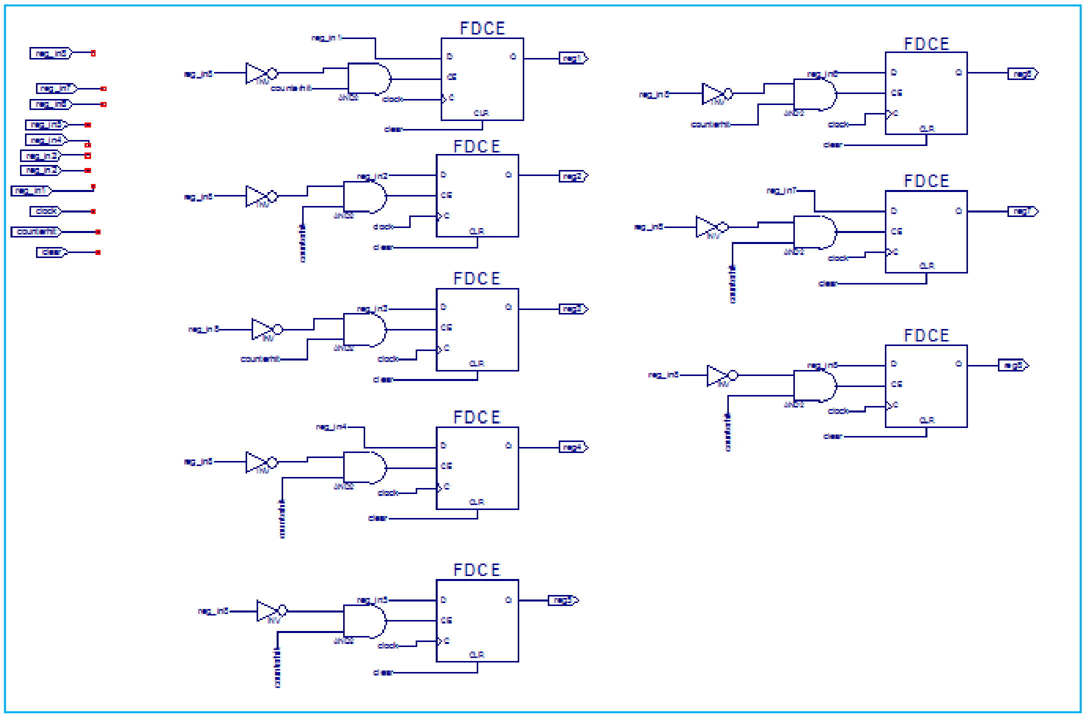
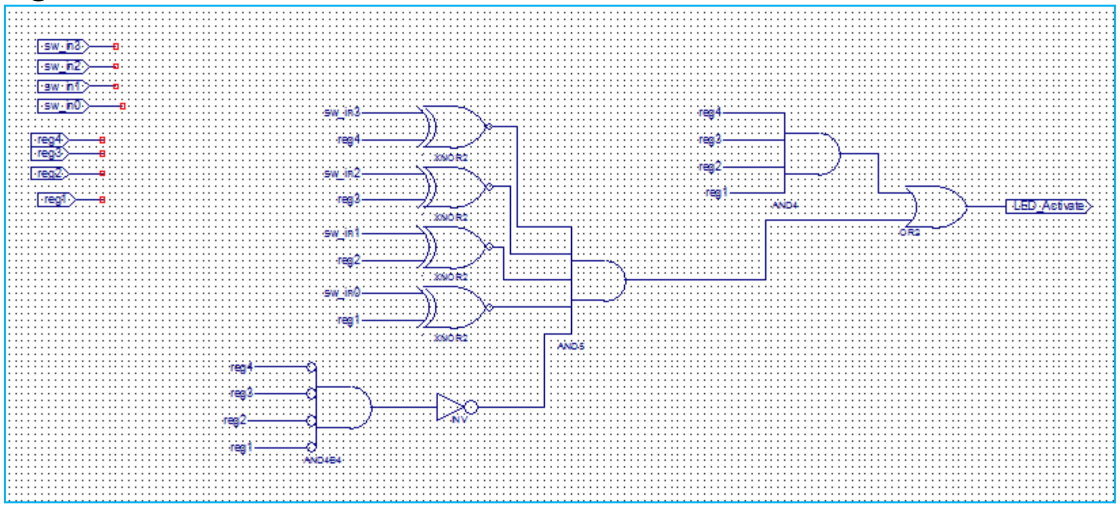


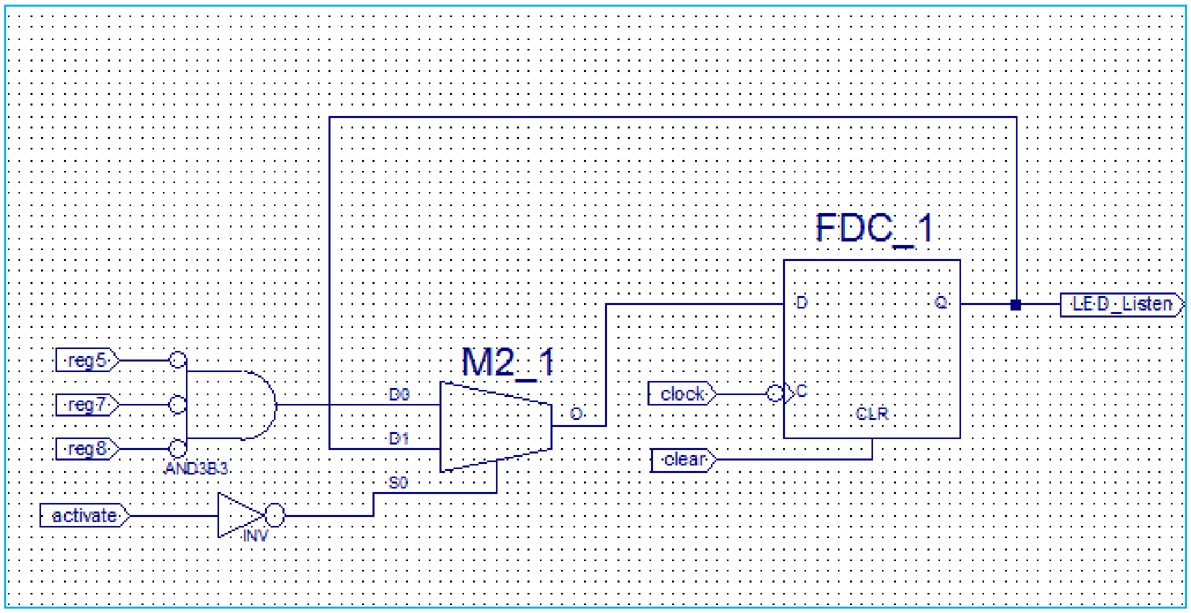
Figure 7: Switch Active Schematic



Remote Activation

The next module that was created, was one to take in the value of whether or not the module is being talked to, and to change the listening state of the board based on whether or not the input values from the registers are telling it to turn on or off (See Figure 8: AmIOn schematic). If it is told to turn on, it will latch the on value until a clear or a turn off signal is sent.

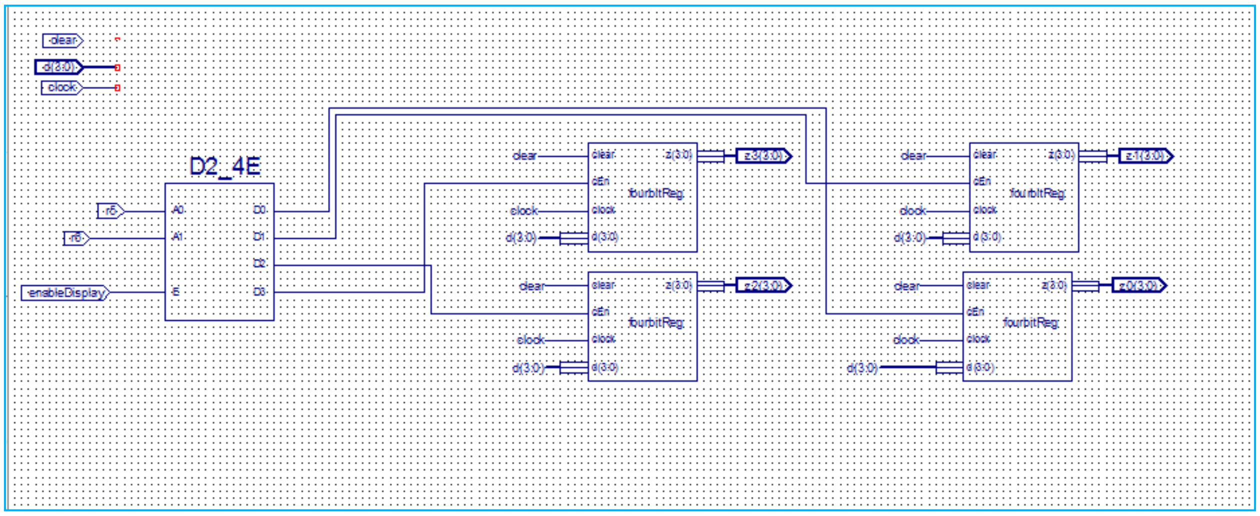
Figure 8: AmIOn Schematic



Display Registers

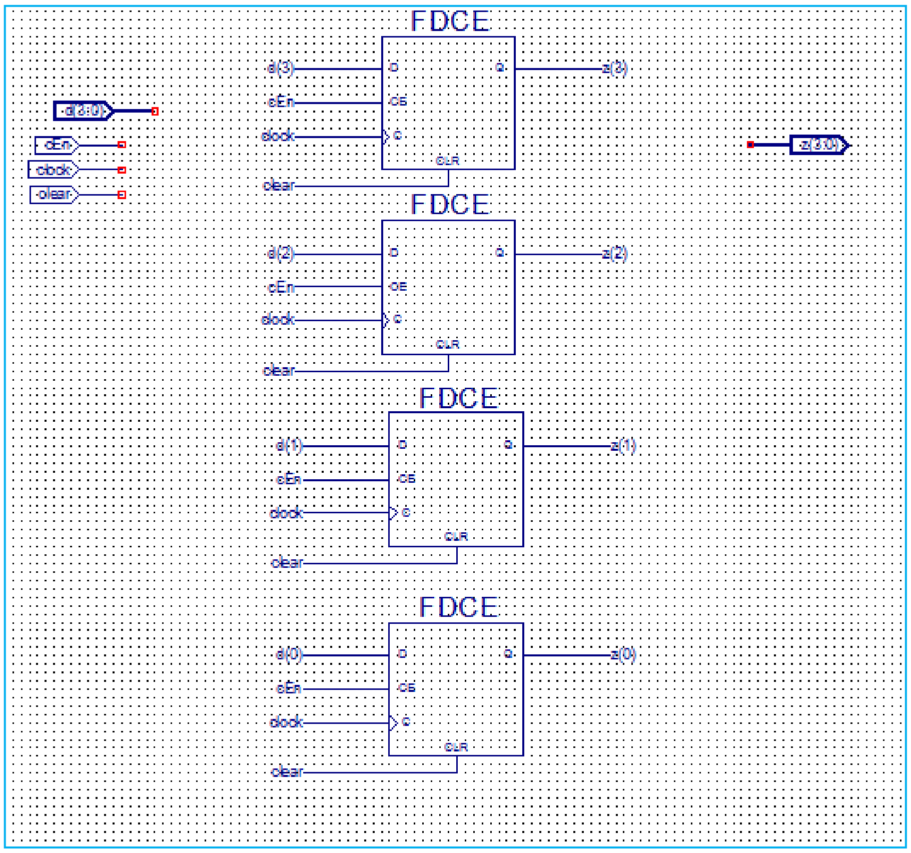
The next thing that was created was the four four-bit register file (See figure 9: 4 four bit Register). This schematic contains four, four-bit registers. What it does is store the display values that are input into the main register when the board is activated. This allows the display values to continue to operate until a clear signal is sent, an error is detected, or a new value is sent to replace the old value.

Figure 9: 4 four bit Register



This (Figure 10: 4-Bit Register) is the four-bit register that is referenced four times in the above schematic. It allows for all four values to be latched simultaneously.

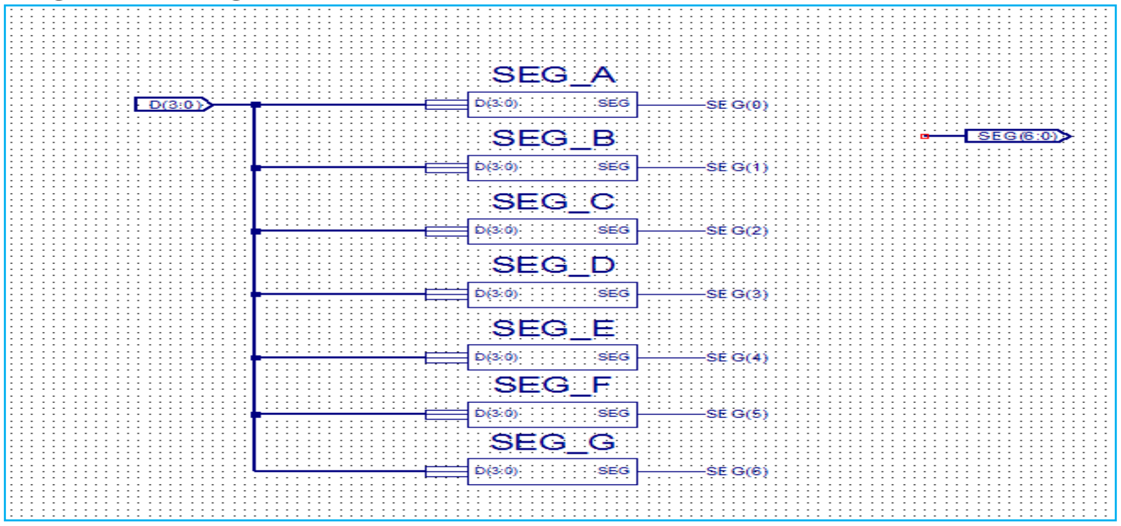
Figure 10: 4-Bit Register



Display and Error Modules

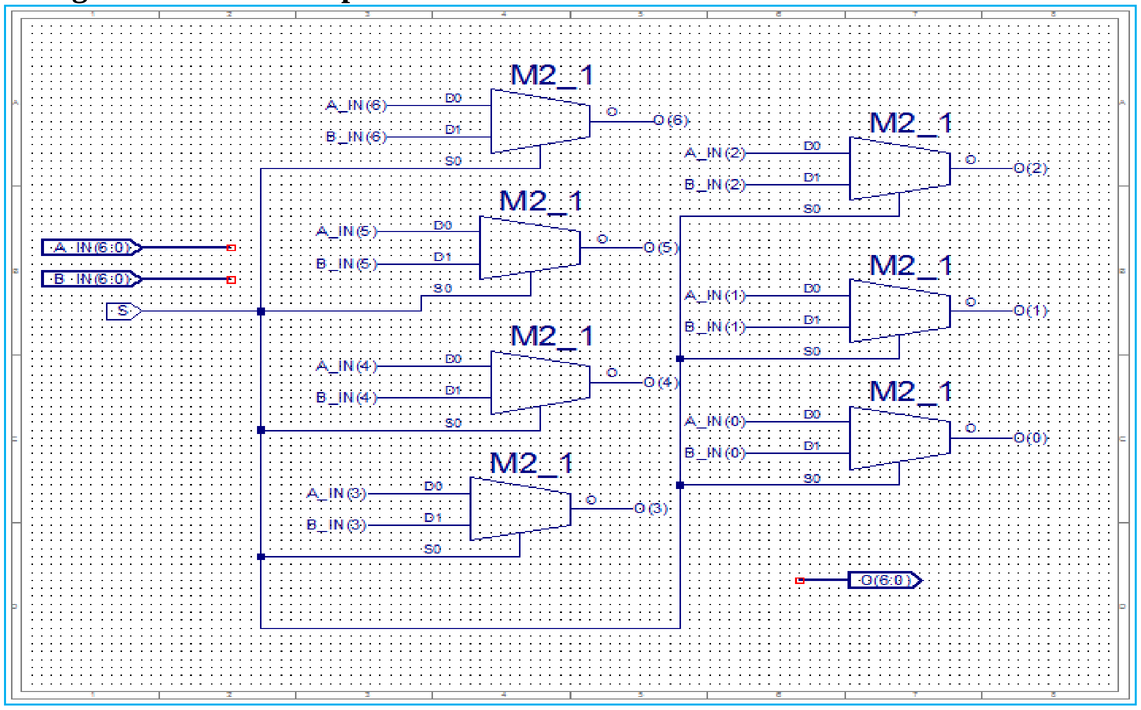
To display all of the values from the registers, we used four 7-Segment display modules (See Figure 11: 7Seg Schematic). These were first described in Lab5, so we will leave that detail out here. Once a register is activated and latched, which is done by using a series of combinational logic and a decoder, the value is displayed on the referenced seven-segment display. This value will stay until it is overwritten by a new value, a clear is sent, or an error is detected.

Figure 11: 7 Seg Schematic



In the event that an error is detected, which is detected by seeing if a one is being latched in the 7th bit position on the eighth clock cycle, the buss Multiplexer then becomes activated and activates the error signal. The Buss Multiplexer was described in Lab 6, but in essence it is a series of muxes that choose between two set of four-bit inputs. When this happens, the Buss Multiplexer (See Figure 12: Buss Multiplexer), switches inputs into the last two seven segment displays, and clears the other two. On that switch, it displays the letters 'Er' on the board, and ignores all further input until a clear is sent.

Figure 12: Buss Multiplexer



The schematics for the ‘E’ (See Figure 14: Letter E Schematic) and the ‘r’ (See Figure 13: Letter r Schematic) symbol were also described in Lab 6, but are displayed below. Upon the error circuit being initialized, these two schematics are activated and the circuit ignores input until a clear.

Figure 13: Letter r Schematic

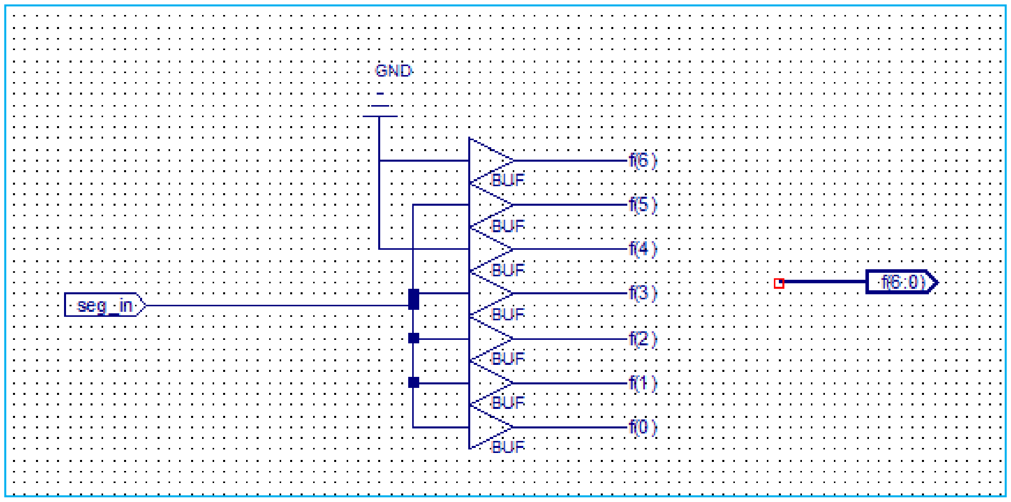
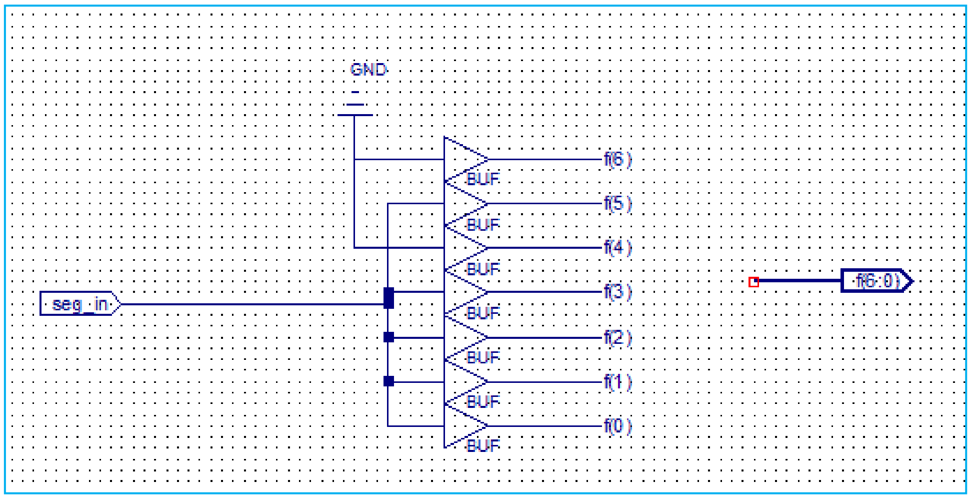


Figure 5: Letter E Schematic



Design Testing Details:

Testing this design was the main problem for the whole project. As each module was created, we would run a test bench and see if the results that were produced were as expected. However, even with this, problems would crop up when the modules were all combined on the top level schematic. In the end, this mandated that we break up the register for the activation switch, the activation switch, and the amIOn module. When all of these were broken up and tested separately, they all worked, and upon combination, they worked tighter efficiently.

The main register and counter were two problems in and of themselves. It took almost until the rest of the project was done, before we had a counter that worked, as we needed. In the end, it took us using a library counter that enables the presetting of the start value. This enabled the circuit to run from a count of one to eight, and reset on eight.

Once the counter was working, the register was able to pick up on all of the input values and transmit them effectively to the other modules in the design.

Lastly, we ran an overall test bench of everything together, and this was very helpful in seeing how the modules were talking to each other. One thing that we found was that one of the modules was not registering correctly because of a lag in the transmission of data. We ended up having to make of our flip-flops register on the falling edge of the clock in order to counter the problem.

After that, the modules all worked together very well, and the check of was completed without difficulty.

Results/Conclusion

Overall, this project started off to be quite challenging. This was because of the scope of the project, and all of the modules that were required. However, once it was broken down, it was much easier to see what needed to be accomplished where, and the project began to go smoothly. The only hiccup was the counter, which took quite a while to fix. However, once it was working, we were able to finish the project with no more problems.