**Lab #6**

**ECE 4304 Spring 2021**

**Professor Aly**

**California State Polytechnic University, Pomona**

**Ahiezer Lopez ahiezerlopez@cpp.edu Bronco ID:012715521**

**Joe Si**

**joesi@cpp.edu**

**Bronco ID: 012636091**

**Sander Zuckerman sazuckerman@cpp.edu Bronco ID: 012644671**

**3/23/2021**

**Objective:**

Students will design a generic x bit input barrel shifter. The number of control bits will be log2(x) and this will determine the number of bits will be rotating. For board implementation purposes, the input will be 8 bits with the control bits being 3 bits. There will be 2 seven segments representing the input and 2 seven segments representing the output of the barrel shifter. Meanwhile, there will be 1 seven segment representing the control bits and there will be a switch dedicated to the direction of the rotation.

**Materials:**

* FPGA (Nexys A7-100T)
* Vivado Software
* Computer

**Contributions:**

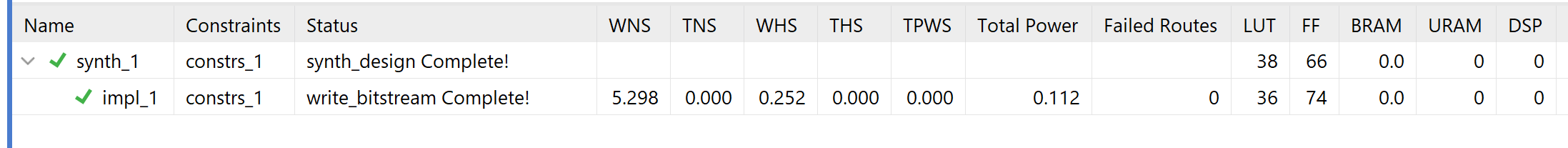
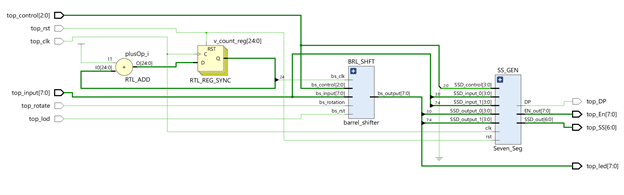
For this lab everyone had their own part in it. Using the Ahiezer’s state machine for multiplying in last week’s lab, Joe was able to design the barrel shifter. Sander made some minor adjustments with the barrel shifter while also connecting all the modules together. Lastly, Ahiezer was responsible for creating the textio of the barrel shifter along with helping connect all the modules together.

**Design Process:**

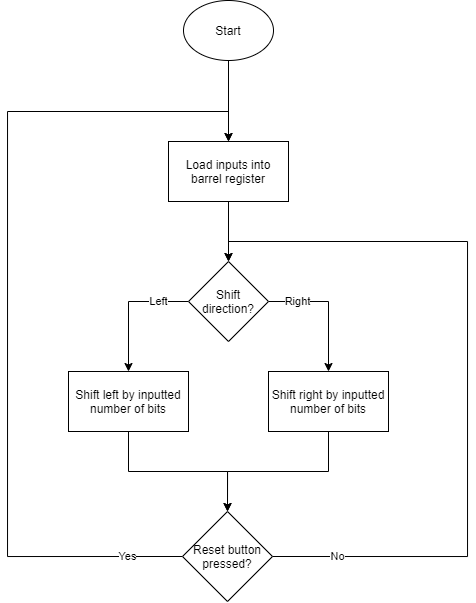
Using the flowchart from below, we implemented our idea on how to approach this lab. We believed it would be best to use a state machine just like with the multiplication from last week's lab. Through the state machine we can rotate it once to the left or right through concatenation. Then we would continue to rotate left or right depending on the number of bits we need to rotate and it would then go onto the next state where the output is assigned to the final output. This was our concept for the barrel shifter and it worked. There were some minor tweaks we did but overall our idea worked perfectly. After connecting everything we had some issues with displaying the right output. This issue was resolved after finding an error in some of the connections between the modules.

**Design:**

Using some code from previous lab experiments such as the seven segment code and the multiplication code. We were able to complete this week’s lab. We used the multiplication code as our template to design the barrel shifter. The output of the barrel shifter was then displayed on the seven segment as well as the input and the control bits. Lastly, the output is also displayed on the leds of the board to demonstrate that it is rotating properly.

**Circuit Diagram:**

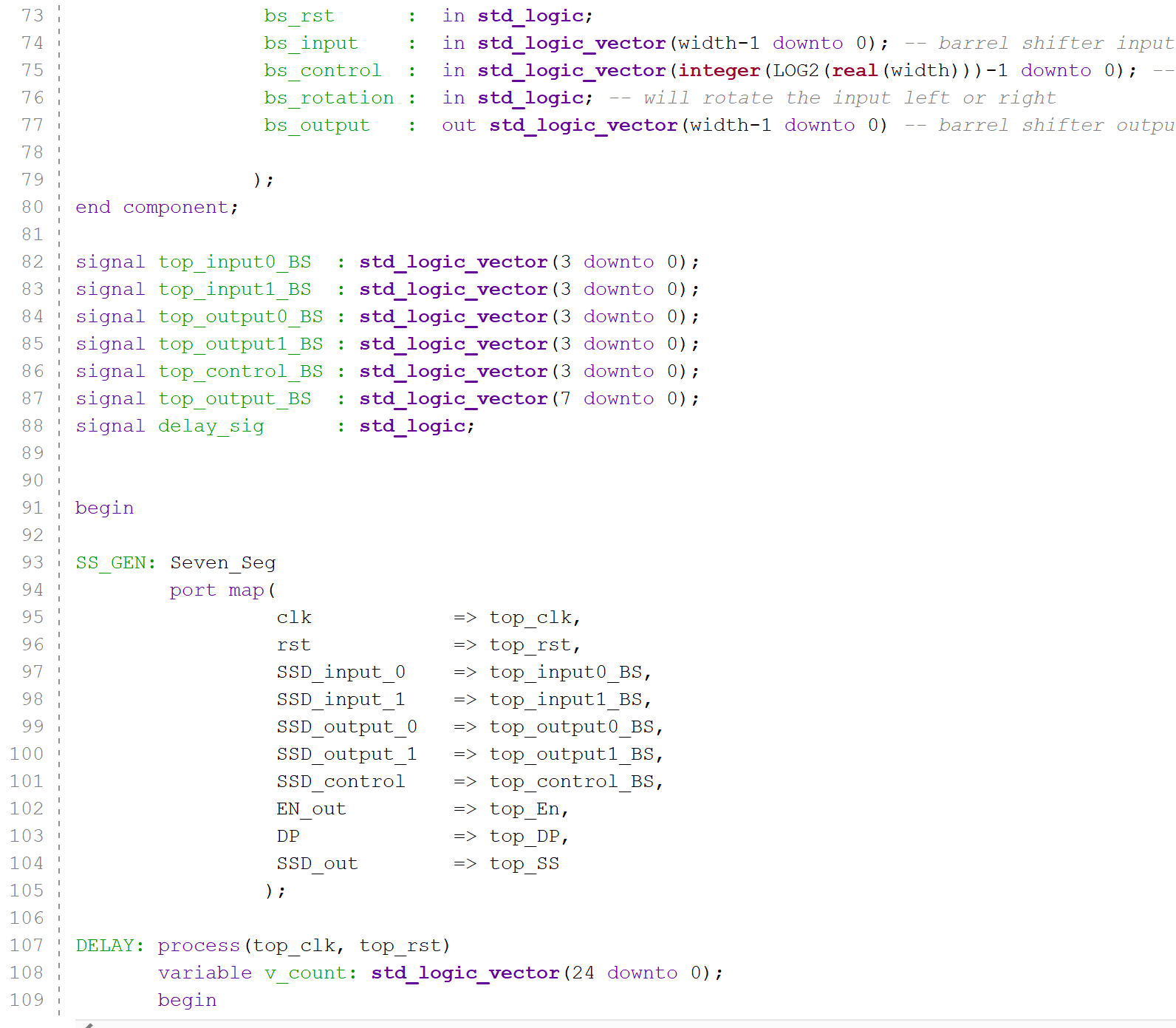
**Implementation:  
 Logic Flow Diagram**

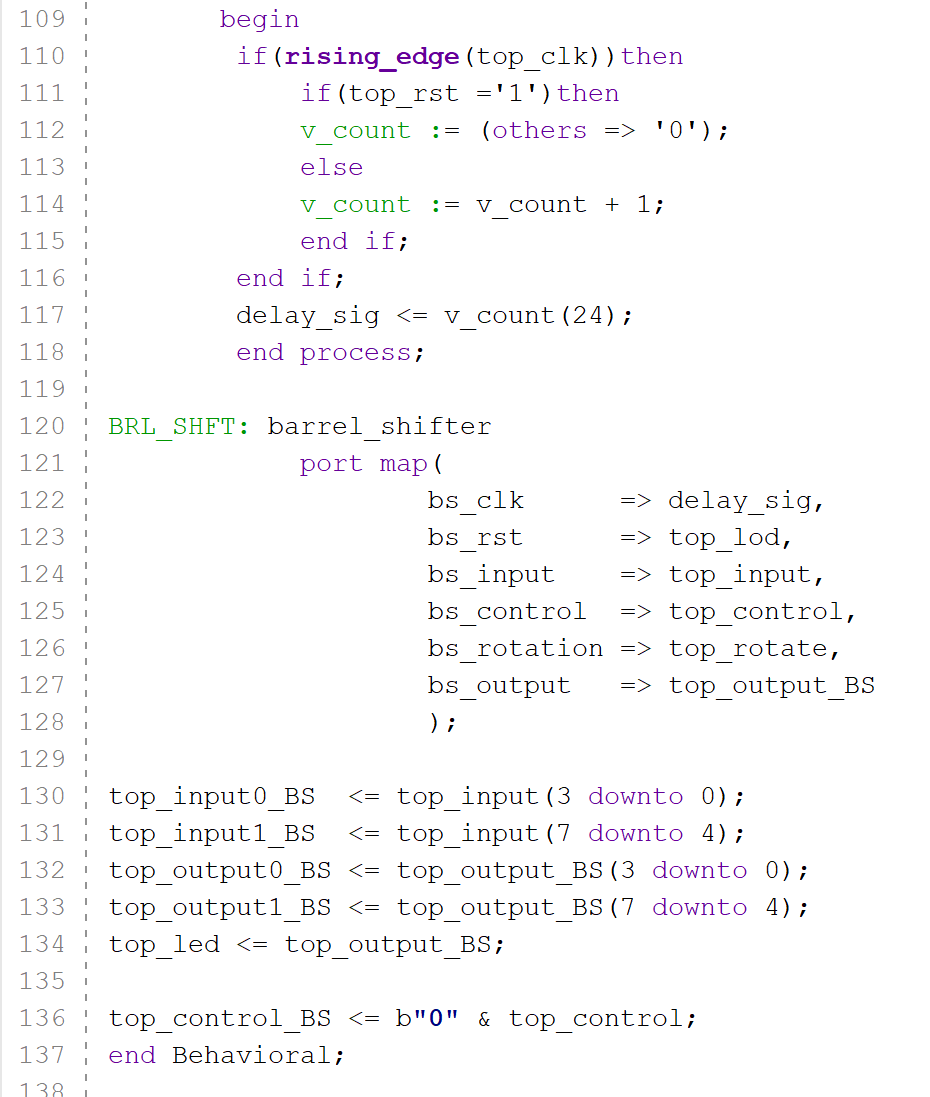
****

**Design:**

***Top Module***

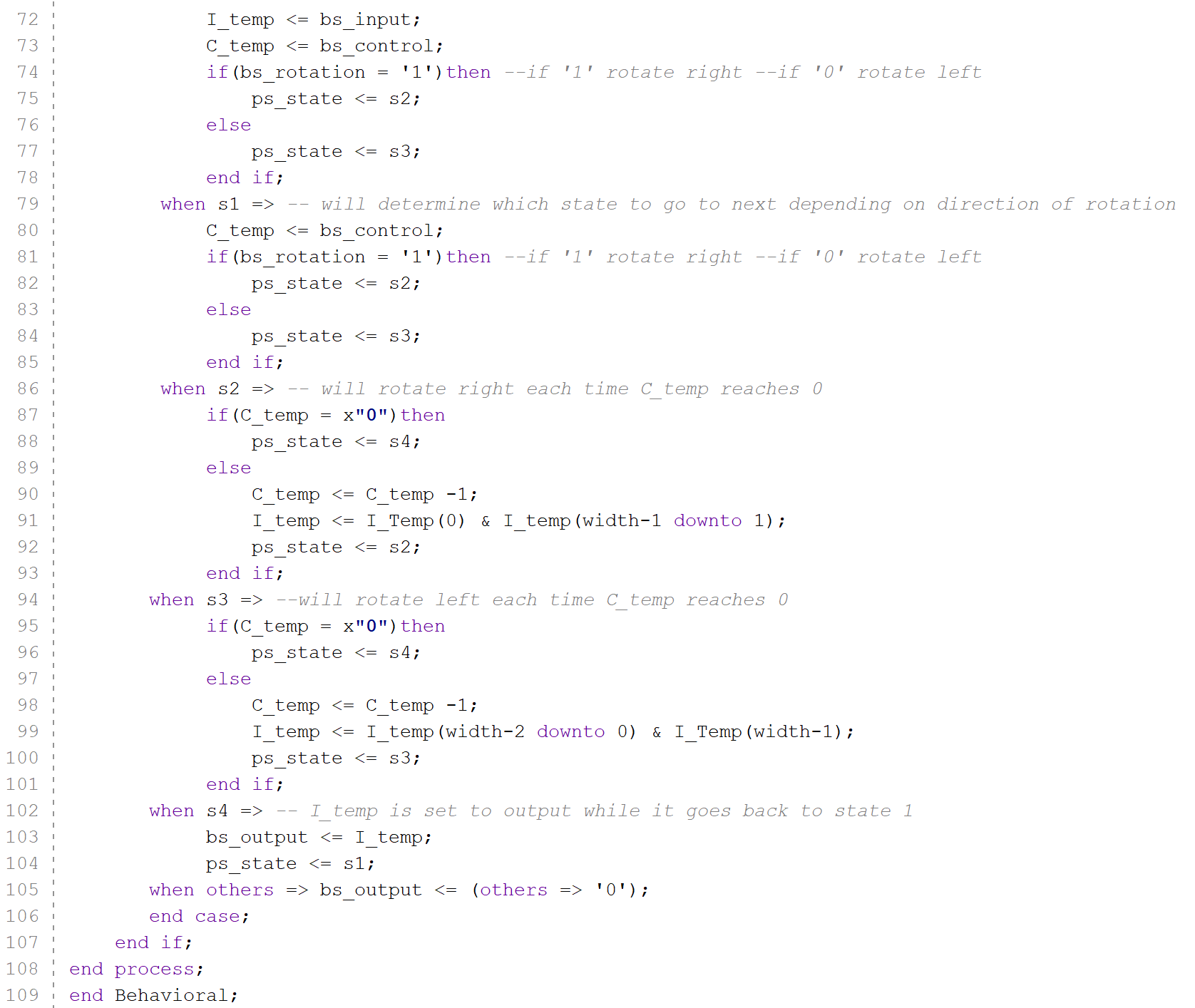




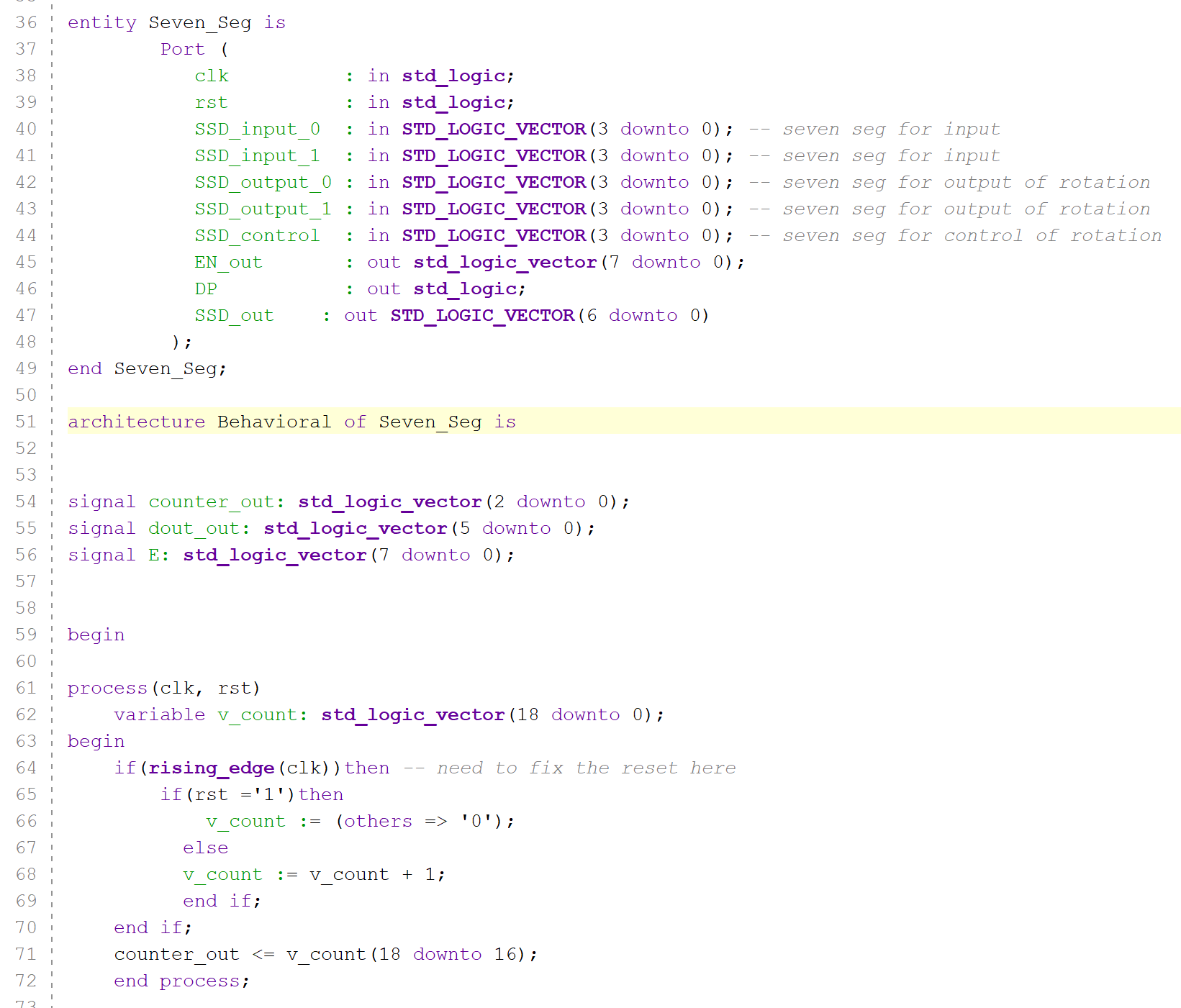


***Barrel Shifter Module:***

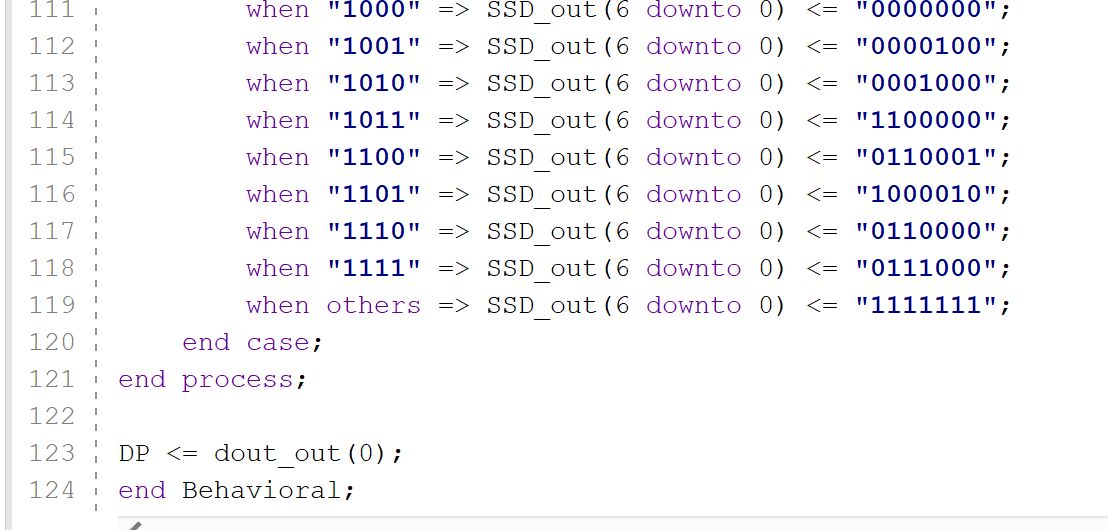




***Seven-Segment Display Module:***

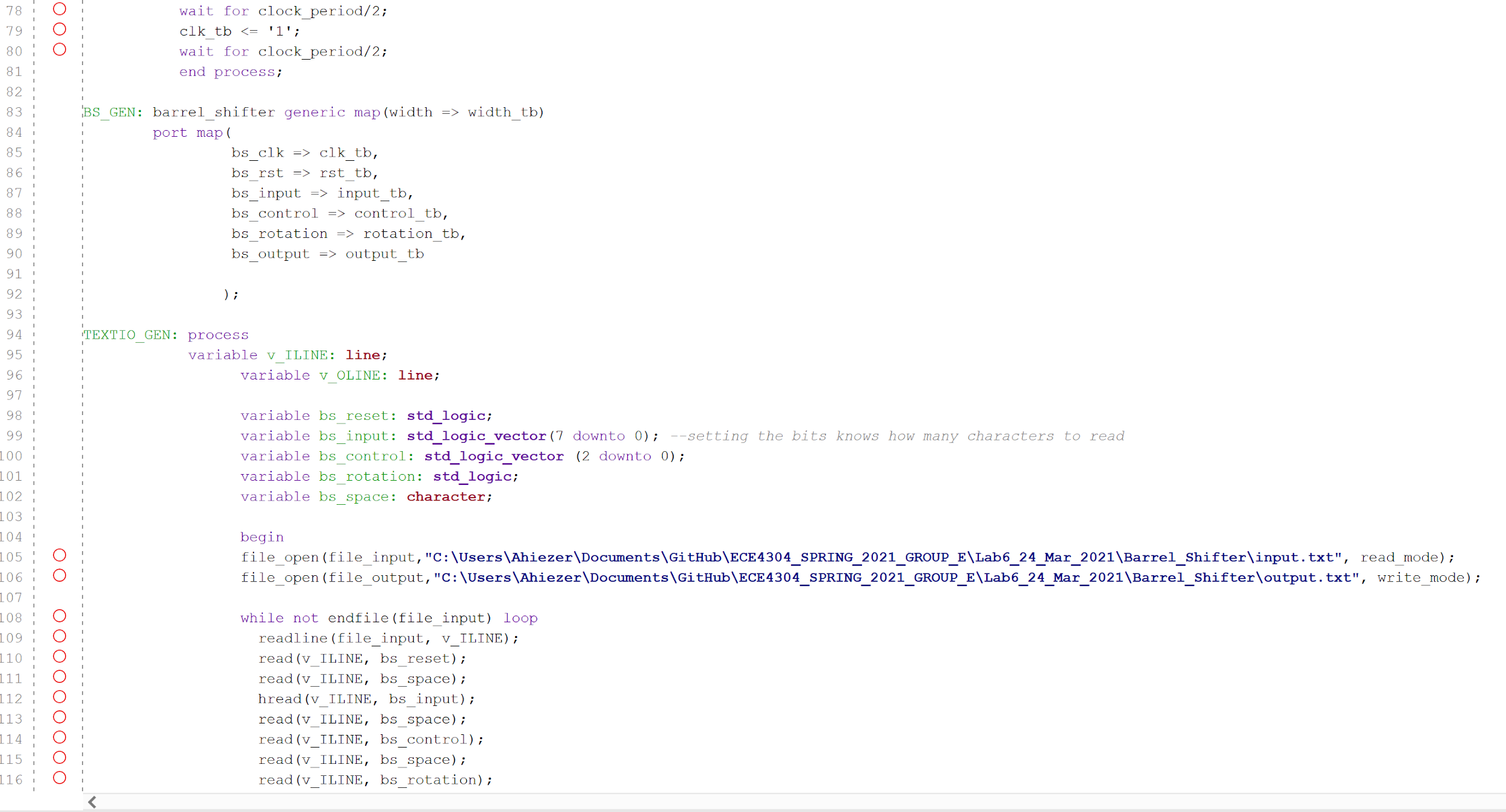
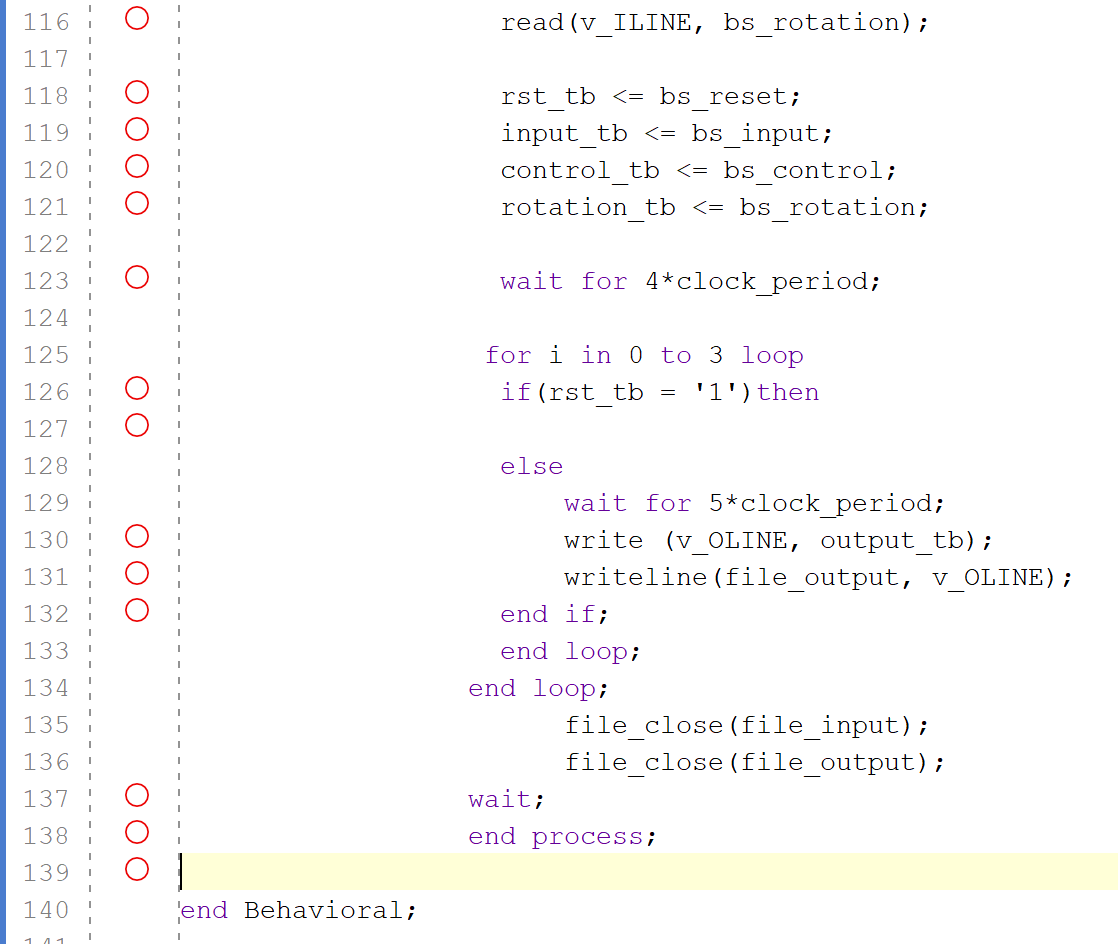






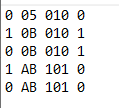
**Test Bench:**



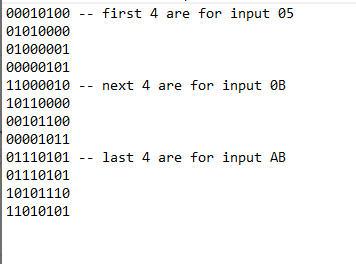


**We did only 4 rotations.**

**Input Text:**



**Output Text:**



**Analysis:**

Surprisingly, we were able to create the barrel shifter with ease. We can only say that the main problem would be that there were some bits off while first creating the block of code. But besides that it was able to work with ease. What we did was create a moore machine where one state would initialize the values, one would load the control bits once again to create a continuous loop. The next two were to either shift the bits left or shift the bits right depending on the rotation bit we created. Lastly was after the loop shifted it to the amount we needed, it would output the correct shifted value.

**Conclusion:**

In this lab, we were successfully able to create a barrel shifter circuit. Using previous code from the last lab, we were able to port over the seven segment code and the bcd code to correctly show on the FPGA board. With a similar format of the multiplication code, we were able to copy the same concept we used to create this barrel shifter and with a few changes, the code was working perfectly. One issue may be that we output the values in hex, however we did not believe that there would be much of an issue.