**Lab #3**

**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**

**2/24/2021**

**Objective:**

Students will create a BCD converter, and a Seven Segment Decoder. Using these two modules the students must have 2 displays on with 4 bits each connected to it. The numbers that must be displayed are zero through nine. Students are required to have the number reset after 9 there for the letter a, or 10 will be zero.

**Materials:**

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

**Contributions:**

The lab was completed with each individual’s help. Sander was able to look back on 3300 verilog labs and recreate the BCD converter. Ahiezer was able to also translate the seven segment verilog code to VHDL and created the top module. Finally Joe was able to help with the implementation of the code and any small bugs the code may have had.

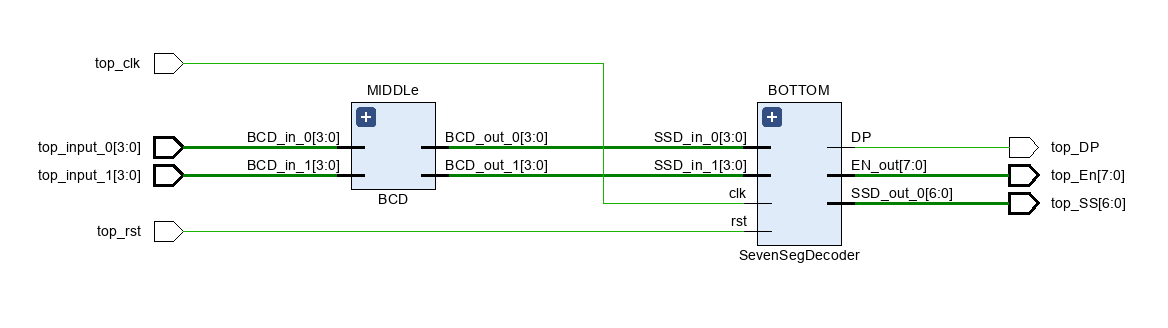
**Design Process:**

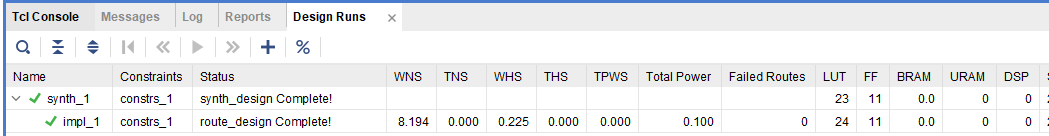
Using the flowchart below, we were able to implement our version of the BCD converter and the Seven Segment Decoder. Initially it was difficult for us to convert the code, however with persistence it was easily possible. There were some issues with the seven segment decoder as some of the numbers were flipped, however there was a quick fix as the problem was due to flipped bits. Another problem was based upon weird flickering of the seven segments, but that was fixed by changing the amount of bits the counter has. We were able to use 0.1 mW for power and create 23 LUT and 11 FF for the synthesis with the implementation being 1 more LUT but same FF.

**Design:**

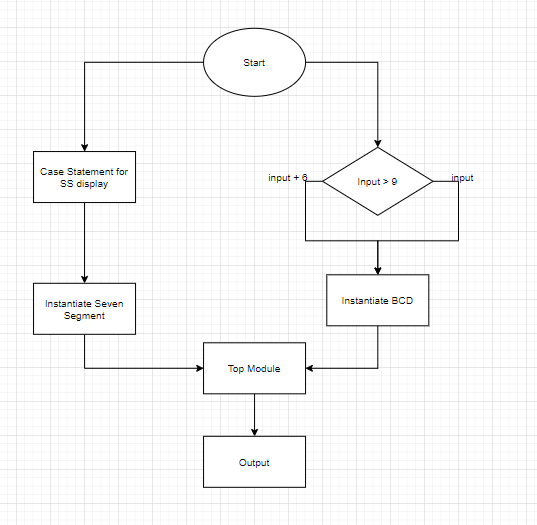
Using previous labs from 3300, we were able to translate the BCD converter, and seven segment decoder modules we created a circuit with 8 switch inputs 4 bits for each seven segment being used, one clk input, one reset, 8 bit output for the enable and a 7 bit output for the seven segment LEDs. The decoder was created with a counter and multiple case statements. The BCD decoder was a basic “when” else statement to create the right number and to reset the number for digits 10 and above.

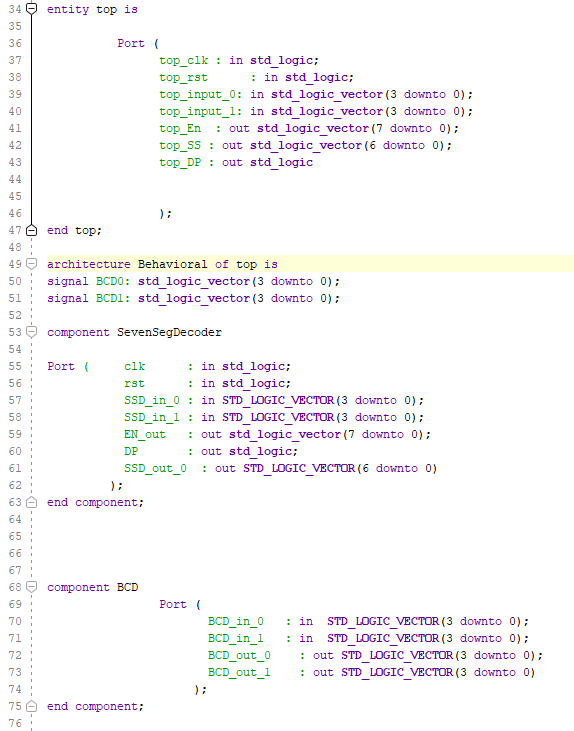
**Circuit Diagram:**

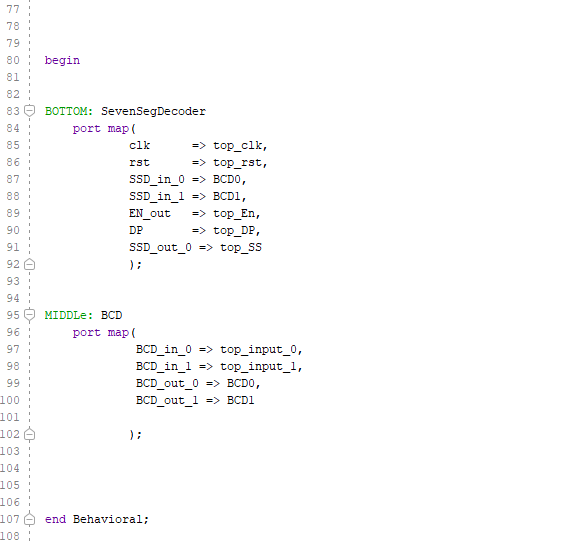
****

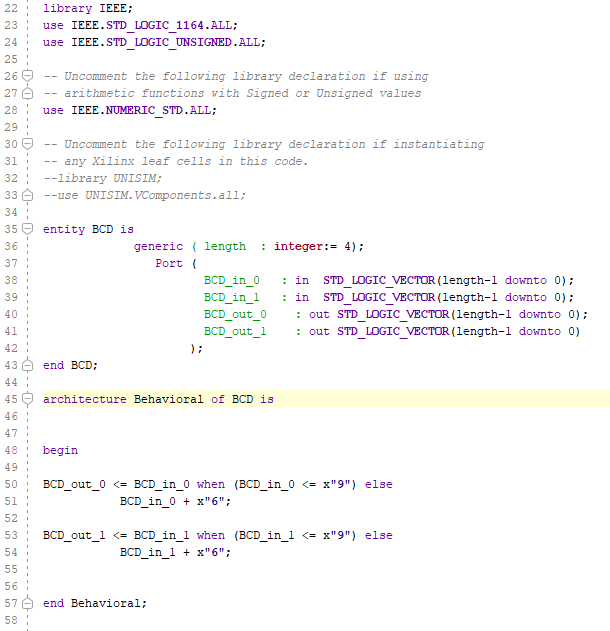


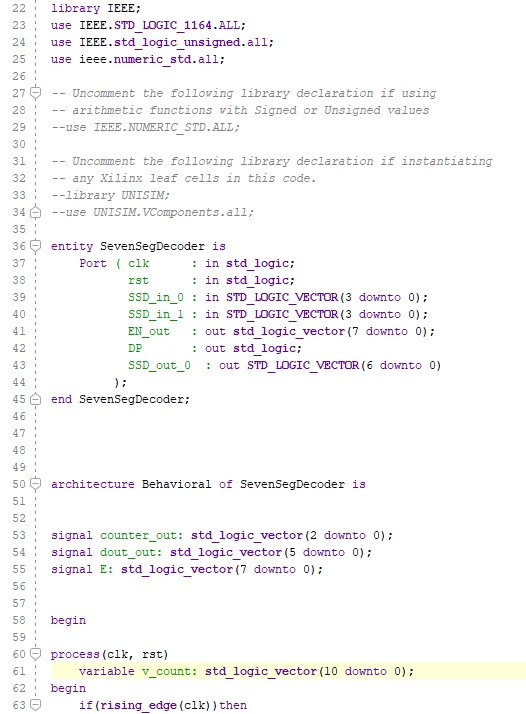
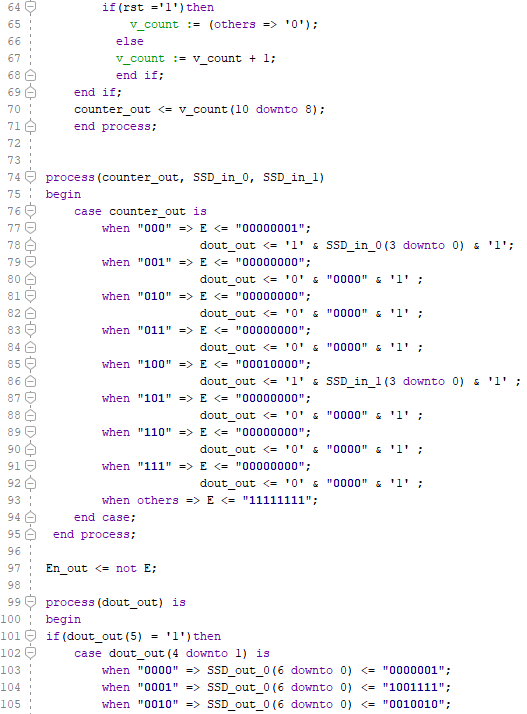
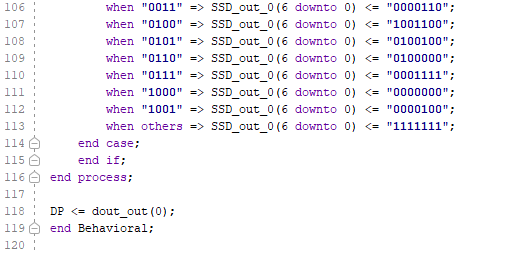
**Implementation:  
 Logic Flow Diagram**



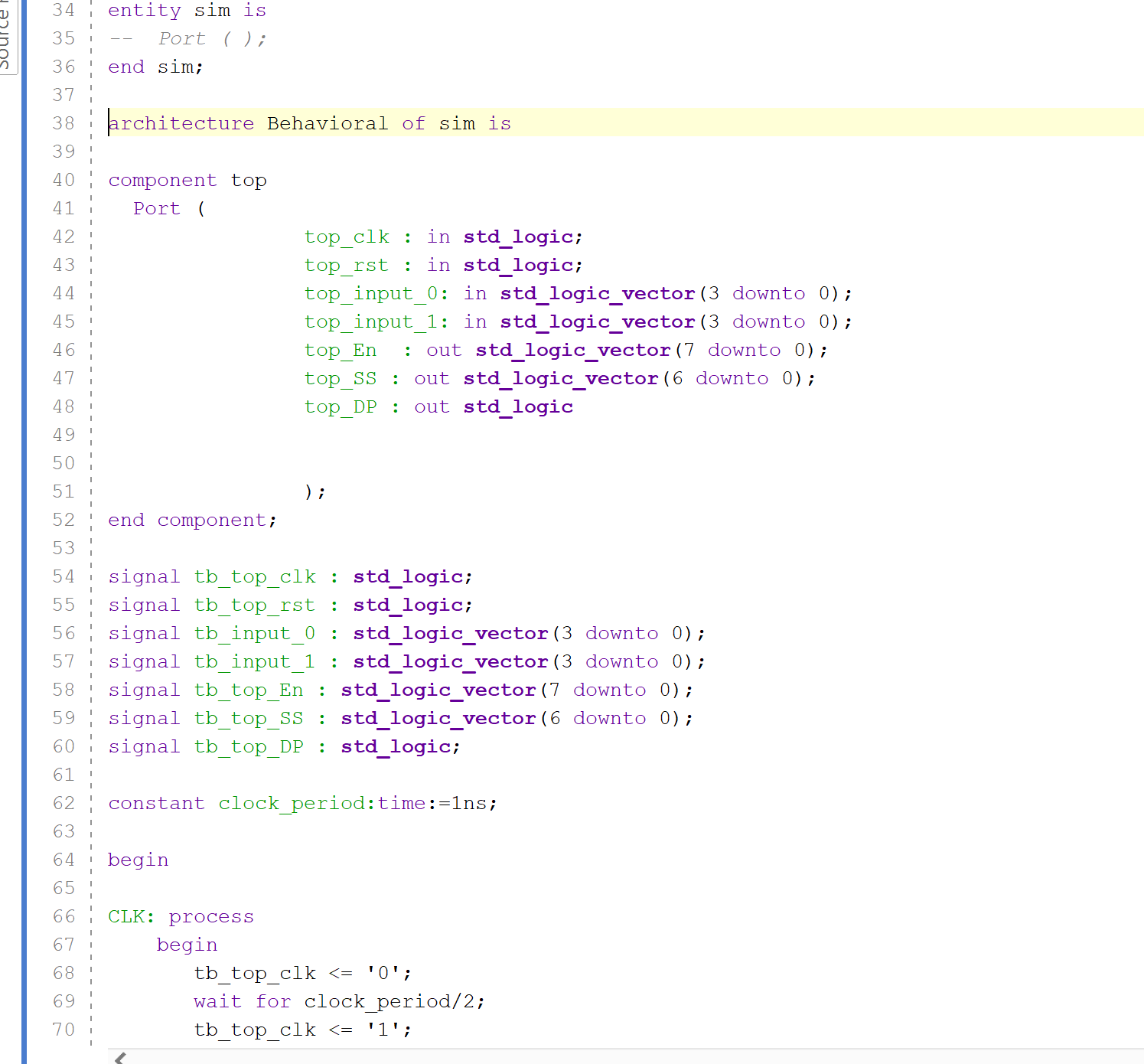
**Design:**  
 ***Top Module:***

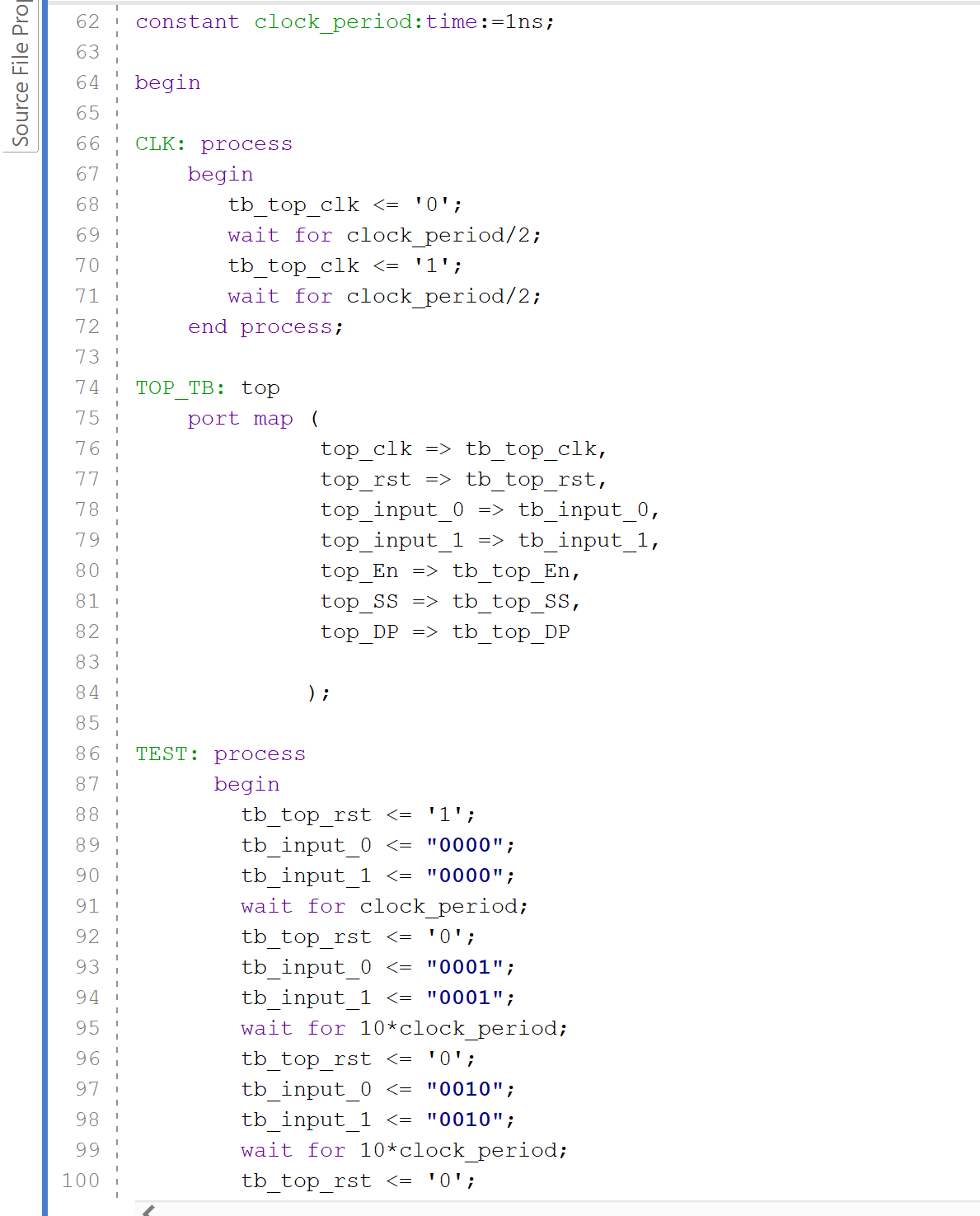
****

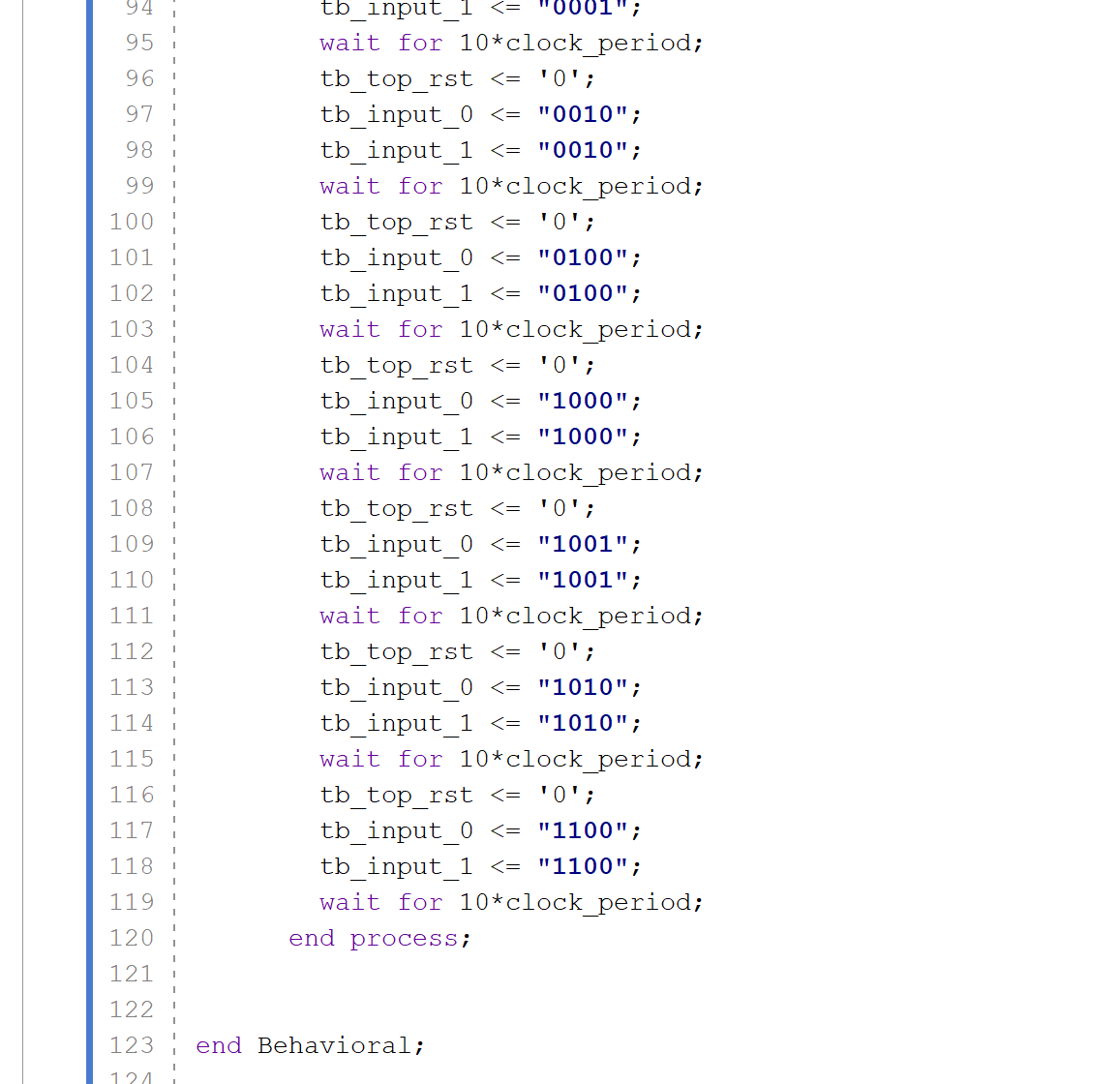
***BCD Module:*  
**

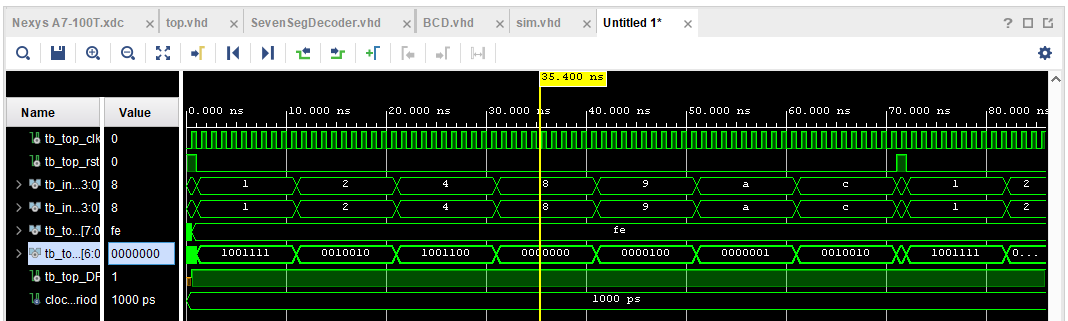
***SevenSegment Module:*  
  
  
**

**Test Bench:**

****

****

****



**Analysis:**

In this lab, we created a BCD converter by scanning the binary input and conditionally outputting the binary-encoded decimal output. In this case, this was achieved by adding 6 to the output if the encoded input was greater than or equal to 10 in decimal. We also created a 7-segment decoder, which takes in a 4-bit input and outputs 7 bits. For this board, the 7 segments have a common anode, meaning logically the 7 segments are “active low”. This means they are activated when a low logic (ground) is outputted, and deactivated when a high logic is outputted.

Furthermore, as all 7-segment displays do, each display is connected to an enable bit. Between each 8 displays, there are only 7 segments for all of them, meaning to display different numbers on different displays, the board must quickly switch between them, displaying the correct number on the correct display for a fraction of a second before switching to another display. Due to the physiology of the human eye, this switching (which would appear as flickering) is not visible, a phenomena called the “persistence of vision”. Achieving this fast switching speed is done by utilizing the board’s clock, which is 100 MHz. It is first divided down to an appropriate speed to accommodate the number of displays, and is used as an input for a MUX selector.

As the waveform above shows, all the logic shown above successfully works. Two individual inputs are considered in a BCD converter, which is directly connected to a 7-segment decoder, which itself is connected to the displays on the board.

**Conclusion:**

In this lab, we successfully created two independent 7-segment displays which each take four switches as an input. The two main problems to consider, namely the 7-segment switching and BCD conversion, were both successfully solved. The display switching was solved by utilizing the system’s clock and dividing down / switching as needed to display each inputted number appropriately for a fraction of a second. The BCD conversion was simply achieved by conditionally outputting a number in the range of 0-9, looping around if the input is greater than 9. This was all connected to the switches and displays on the board, and works entirely as intended. This lab allowed us to begin using various on-board hardware, in this case, the 7-segment displays, onboard switches, and the system clock, all of which were successfully utilized as needed.