Task 1:

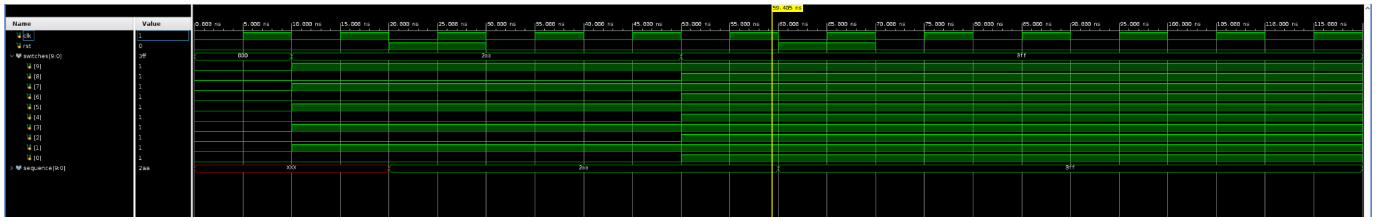

Figure 1: Selector Waveform

Figure 1 shows the waveform from the selector's test bench. To verify our design, the sequence value should be stored when rst is pressed. Looking at the sequence output, it initially doesn't have a value as indicated by xxx. When rst is pushed, however, we can see that the sequence is stored based on the switch input to 2AA in hexadecimal. We then change the input of switches to all 1 and hit rst. That then changes the output to 3FF as the new output.

Task 2:

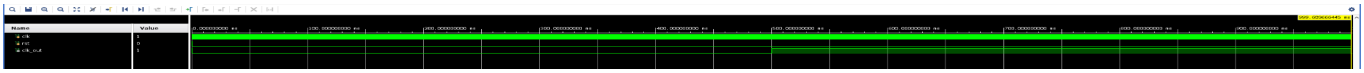

Figure 2: Clock Divider Waveform (1s Timescale)

Figure 3: Clock Divider Waveform (1ns Timescale)

The clock divider waveform is shown in Figures 2 and 3. Figure 2 shows the waveform using an ms timescale, and Figure 3 shows it using an ns timescale. To find out if we successfully convert the clock input from 100 Mhz to 1 hz we have to find the period of the input and output. Figure 2 shows the period of the output clock on the bottom and we can see that the period is 1s according to the waveform. Figure 3 shows the input clock on the top with a period of 10 ns which translates to 100 Mhz confirming that our clock divider is working properly.

Converter:

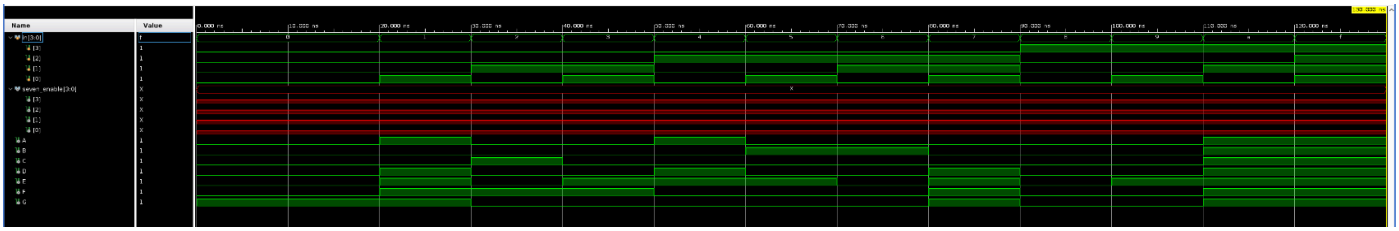

Figure 4: Converter Waveform

In Figure 4, the converter is derived from the top module code and converts inputs from the switches to a 7-segment display. In our code, the output of 1 means that the display is turned off and 0 is on. For example, 1111 input at 120 ns is too high as it's outside of the 0-9 decimal range of our module, and as such all the outputs are as a result 1 which means the display turns off thus confirming our code is working. Another example is at 20 ns where the input is 0001 which translates to a 1001111 output which means that B and C are on which translates to 1 displaying on the output display on the FPGA.

State Machine Diagram:

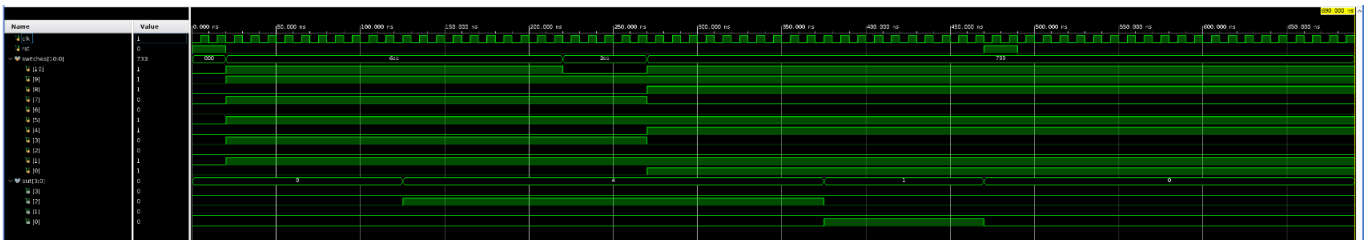

Figure 5: State Machine Diagram

In Figure 5, the state diagram machine is shown above. The clk input is 100 MHz but with our clock divider module, its clock speed is in actuality 1 Hz. The rst input results in the output starting back from 0. The rst button also begins the state diagram. Finally, the switch input is a state diagram that reads the amount of '101's that are active in our inputs. Switch 10 is the pause switch and runs only when it's at 1. To verify our state diagram, we must observe if the output correctly identifies how many '101's. Our input is 1010101010 which corresponds to to 4 '101's. At 125 ns we can observe the output is 0100 in binary which translates to 4 in decimal this verifies our input. The rst is pushed at 475 ns and resets the output.

Top Module:

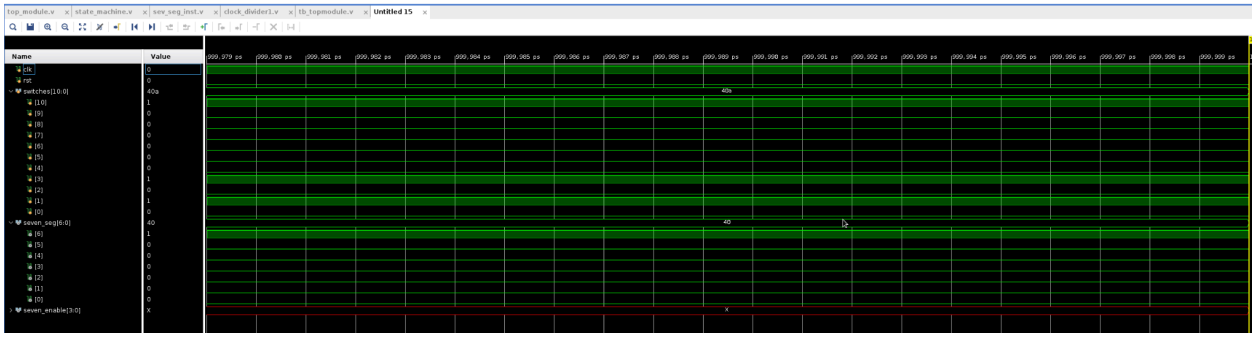

Figure 6: Top Module Waveform

In Figure 6, we have the top module waveform where the inputs are rst, clk, and the 10 switches and the outputs are the 7 segments. The output should be a 7-segment display showing our results as 1 as our input is 0000000101 which corresponds to 1 ‘101’ in our switches.