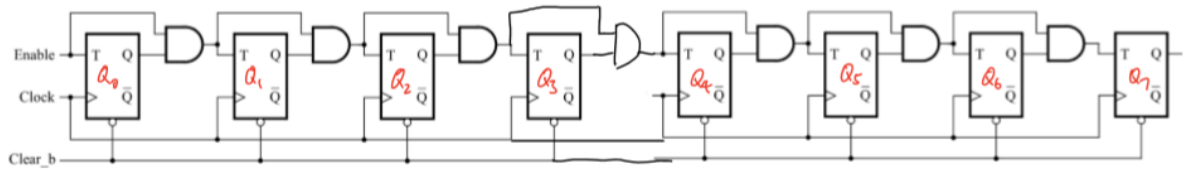


Part I:

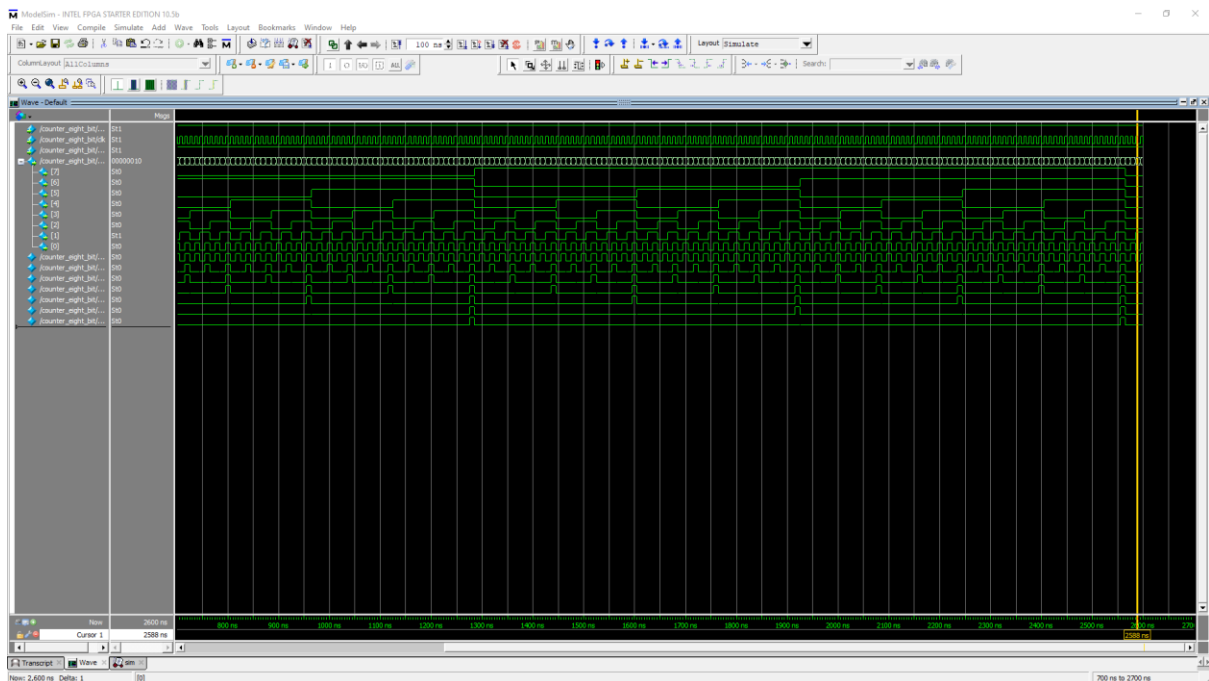
Q1&2:



Schematic for 8-bit counter.

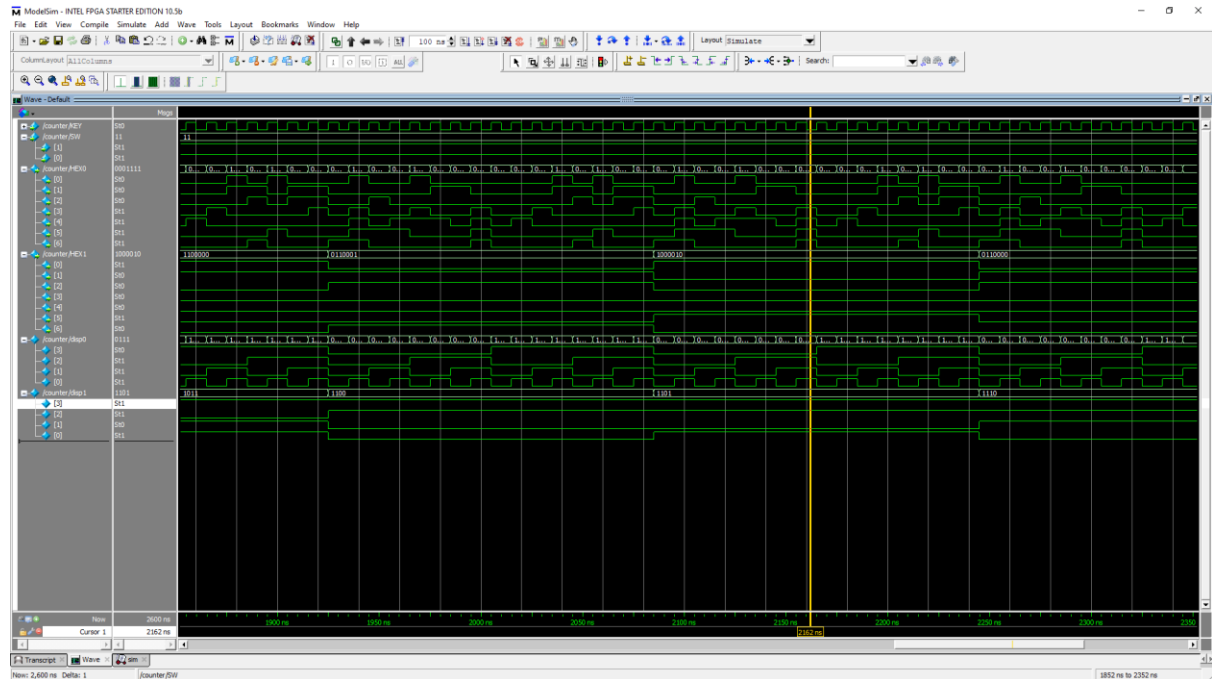
3. Verilog code included in final counter.v file. 8bit counter is defined as counter\_eight\_bit and T flip flop is tflipflop.

4.



Simulation for the eight-bit counter, it counts correctly and resets at the end when number overflows.

5.



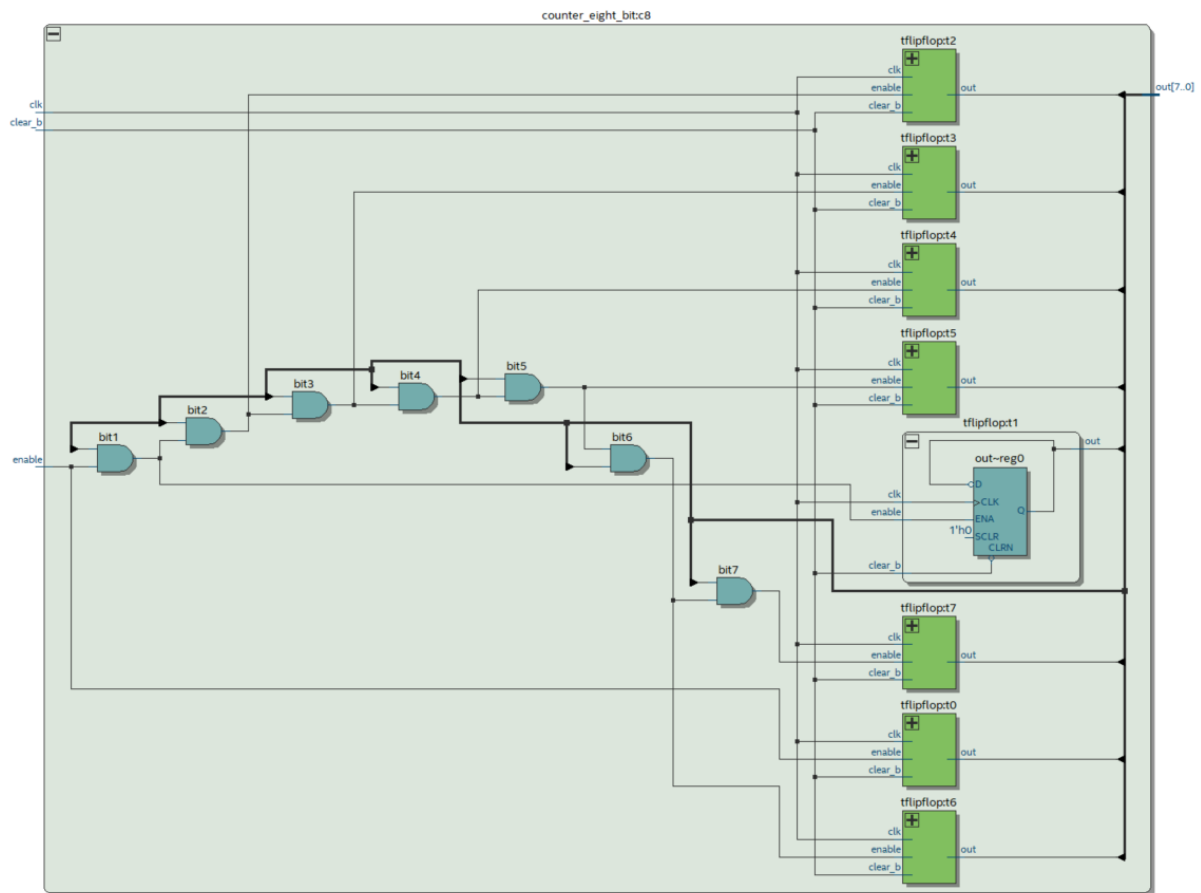
Connected inputs properly, hex displays 0 and 1 now show output of counter with KEY[0] as clk. Verilog code submitted on Quercus.

6.

a) Logic utilization is 13/32070 ( < 1% ), and 8 registers are in use. The size of the circuit is much smaller than the size of the FPGA.

b) 621.89 Mhz

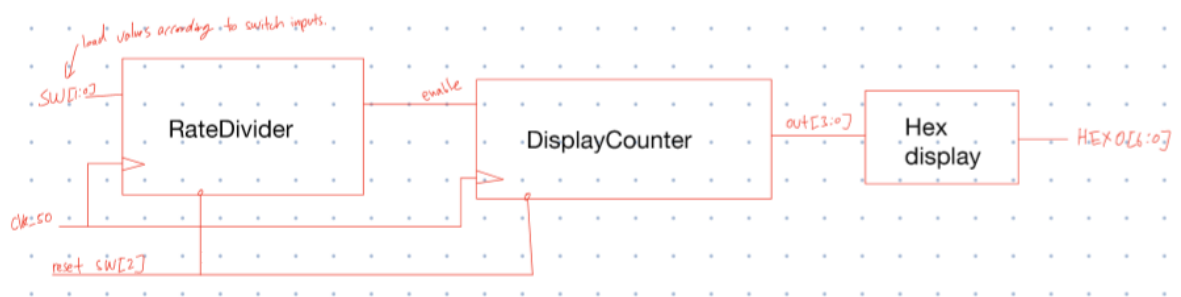
7.



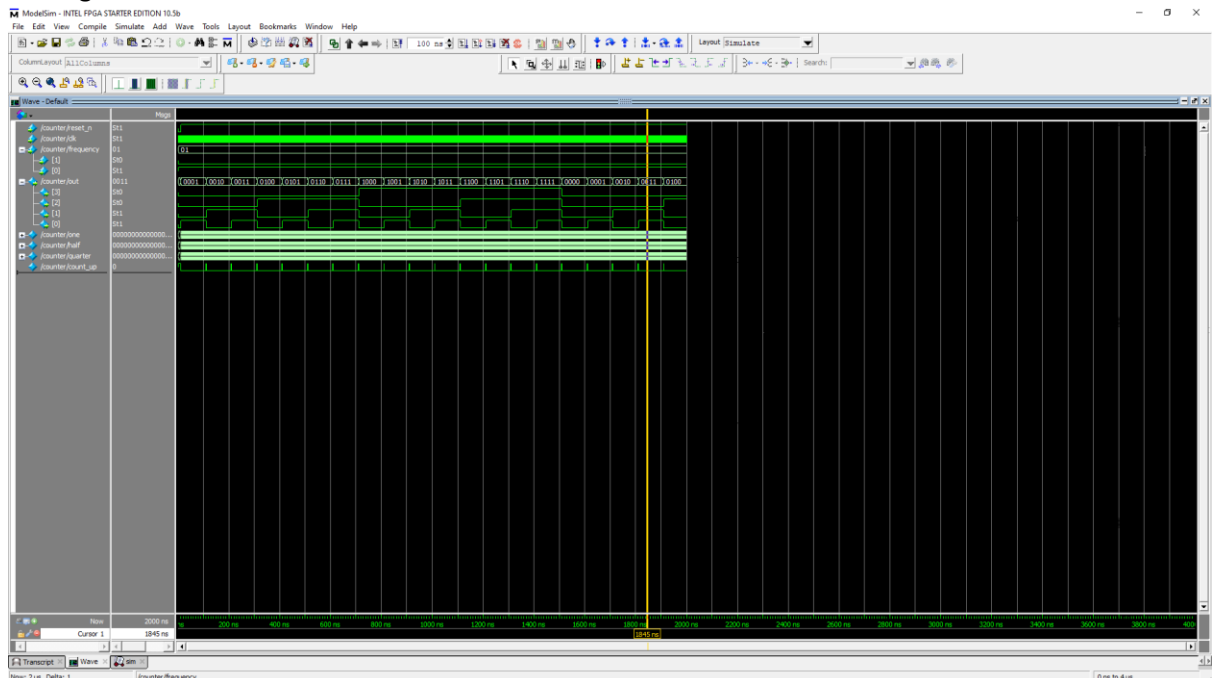
The circuitry is roughly the same, except that there are 8 output lanes and 8 flipflops instead of 4.

## Part II:

- The check for maximum value is not necessary as when it reaches the maximum value, the next 'tick' will cause it to overflow and go back to 0.
- The if ( $q == 4 \text{ 'b1111}$ ) should be changed to if ( $q == 4 \text{ 'b1001}$ ) for the counter to count from 0-9.
- The period of the clock is extremely short, so it is likely that the LED's do not have time to completely switch off, and hence consistently red segments will be displayed. i.e. the hex displays will consistently show '8'.
- You would need 26 bits.



- Schematic for counter circuit. With SW[1:0] as inputs and SW[2] as reset signal.
- Verilog code submitted on Quercus.

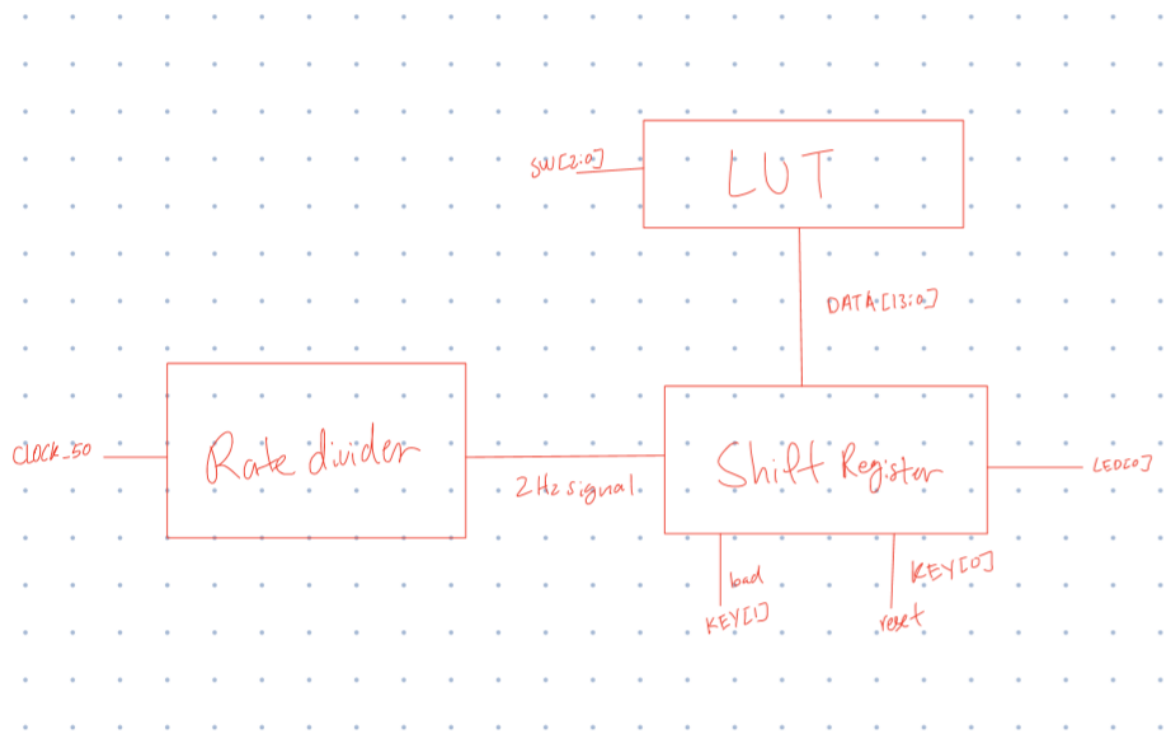


- Simulation of rate divider with a counter set to enable every 50 clock cycles.

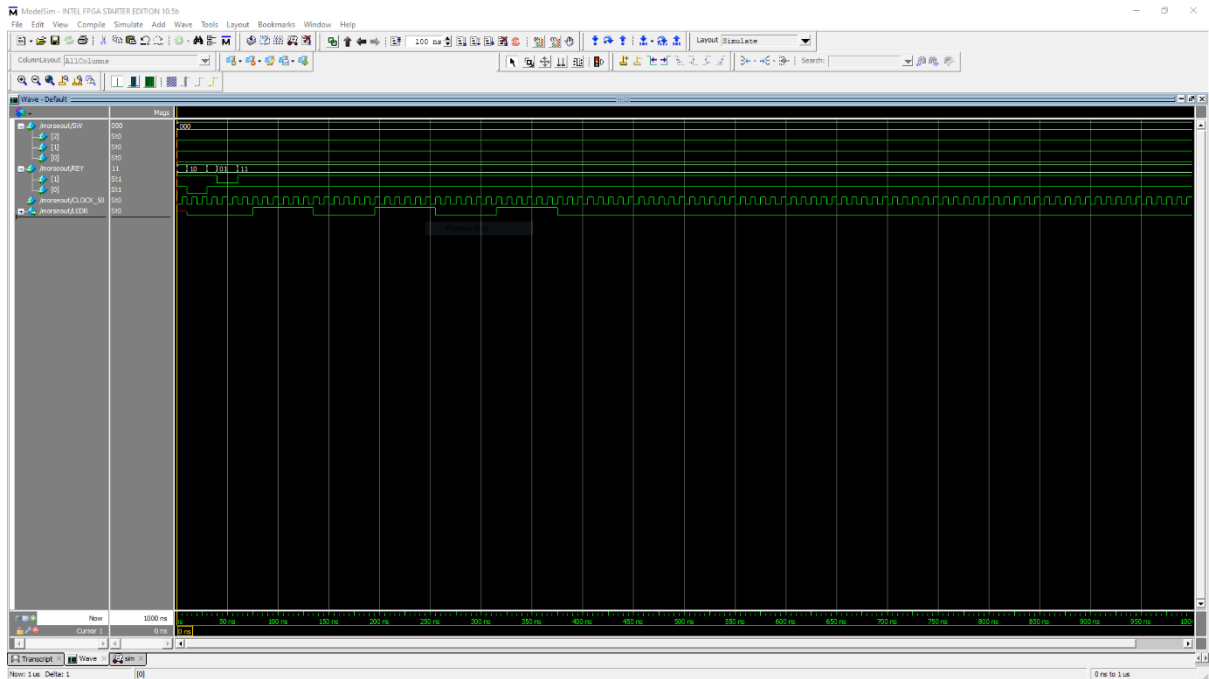
Part III:

1. S:10101000000000  
T:11100000000000  
U:10101110000000  
V:10101011100000  
W:10111011100000  
X:11101010111000  
Y:11101011101110  
Z:11101110101000

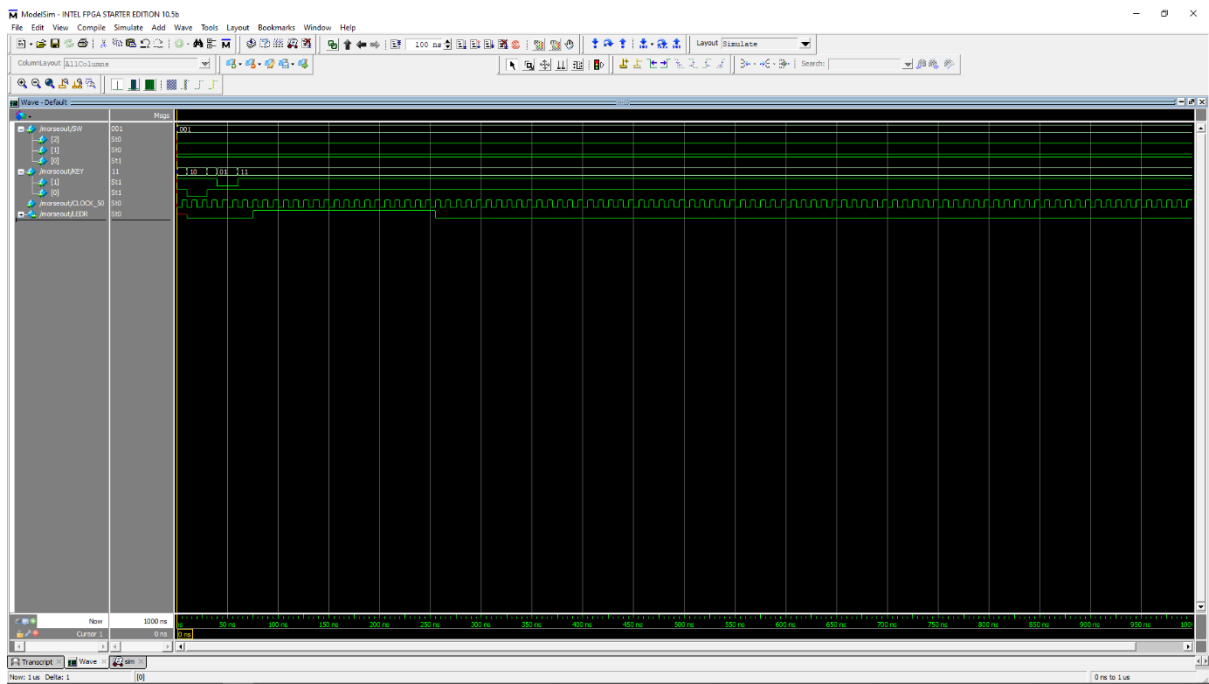
We can use 14 bits to represent each of the letters.



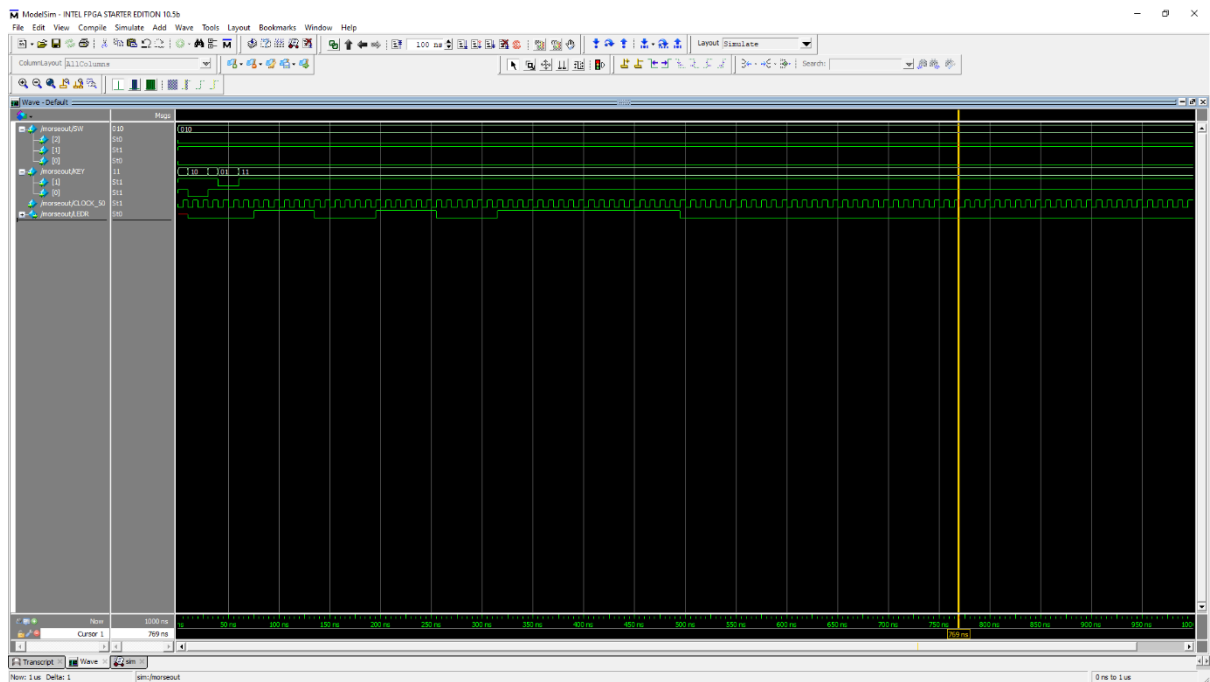
2. Schematic for morse encoder.
3. Verilog code for morse encoder submitted on Quercus.
4. All screenshots of 8 possible outputs for letters S, T, U, V, W, X, Y, Z included below:



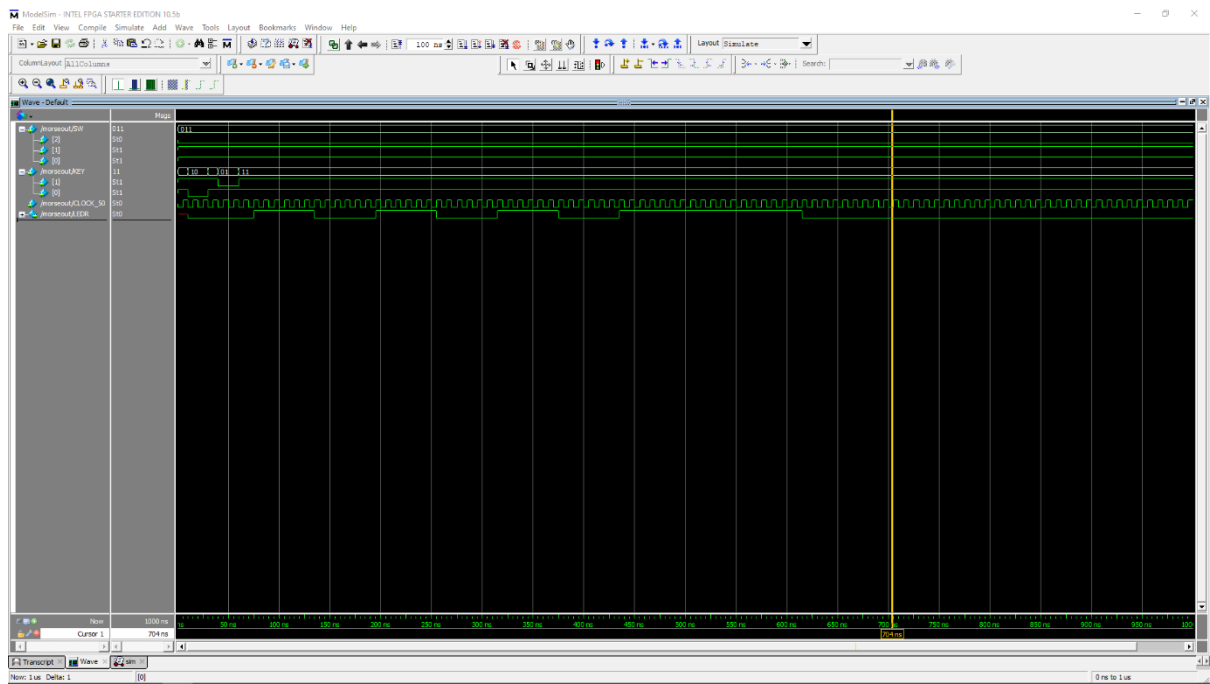
Output S



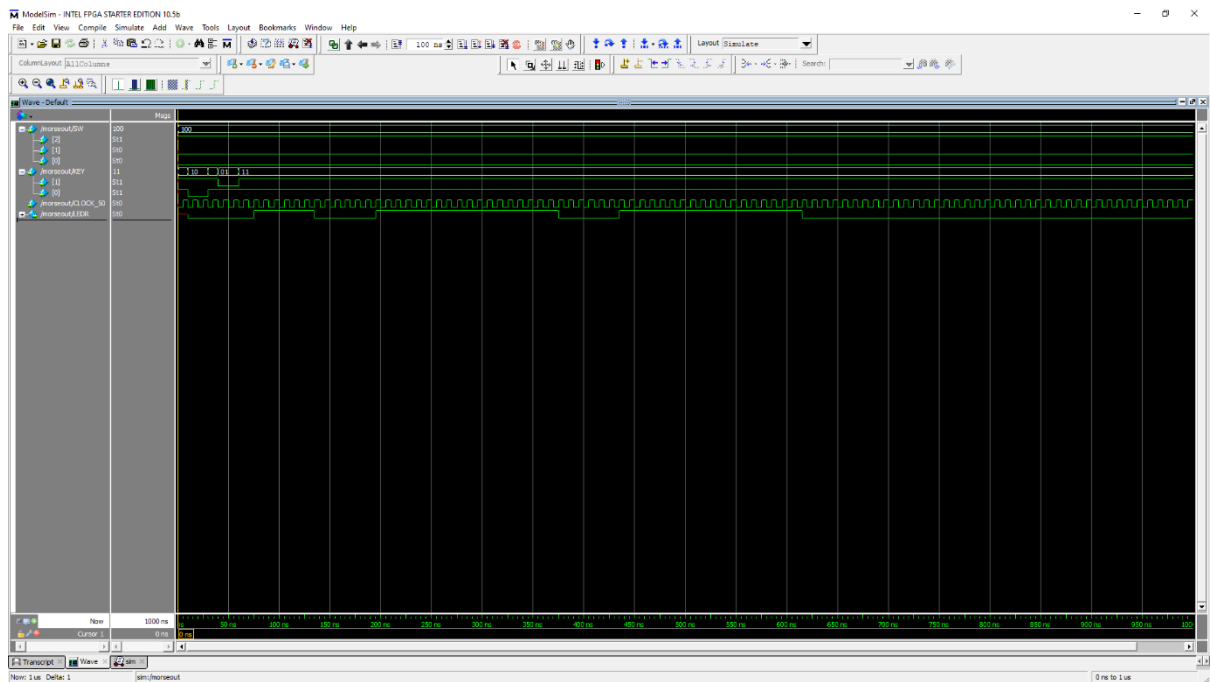
Output T



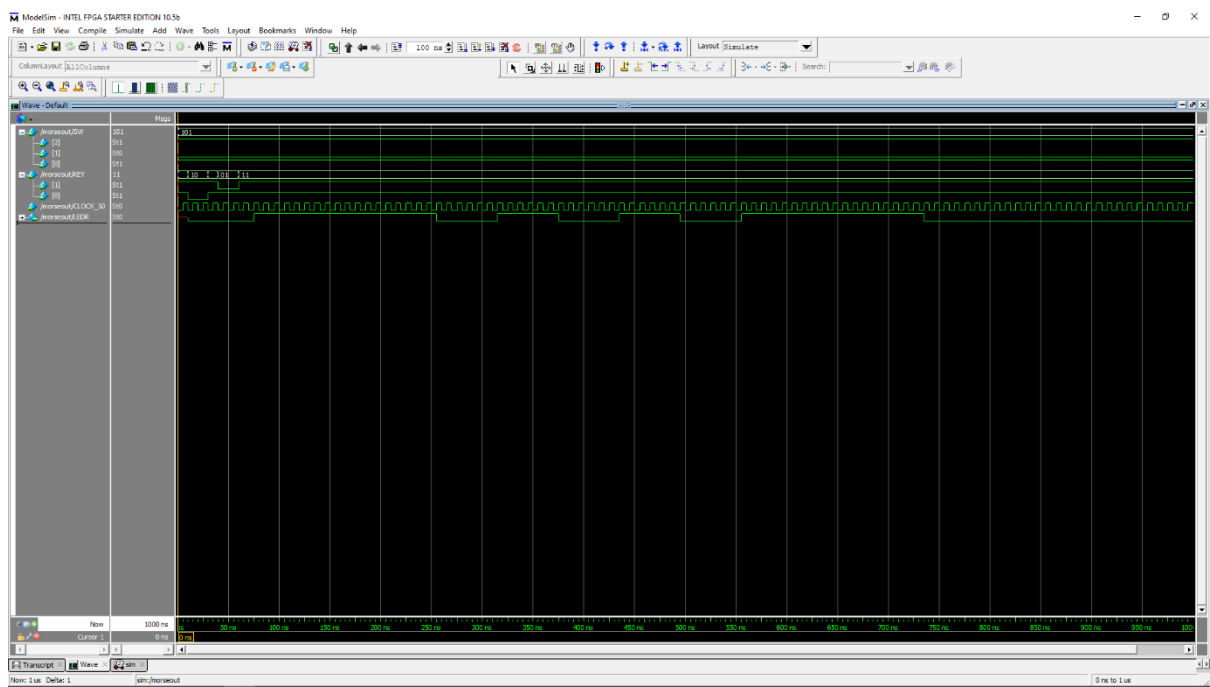
Output U



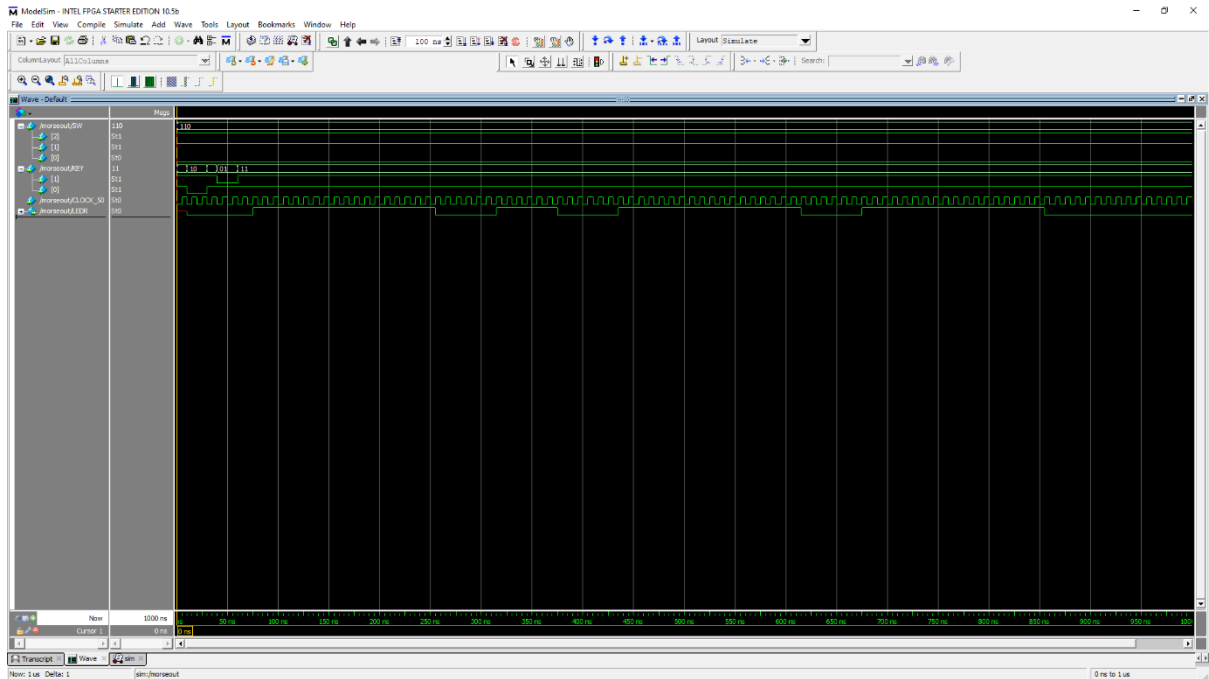
Output V



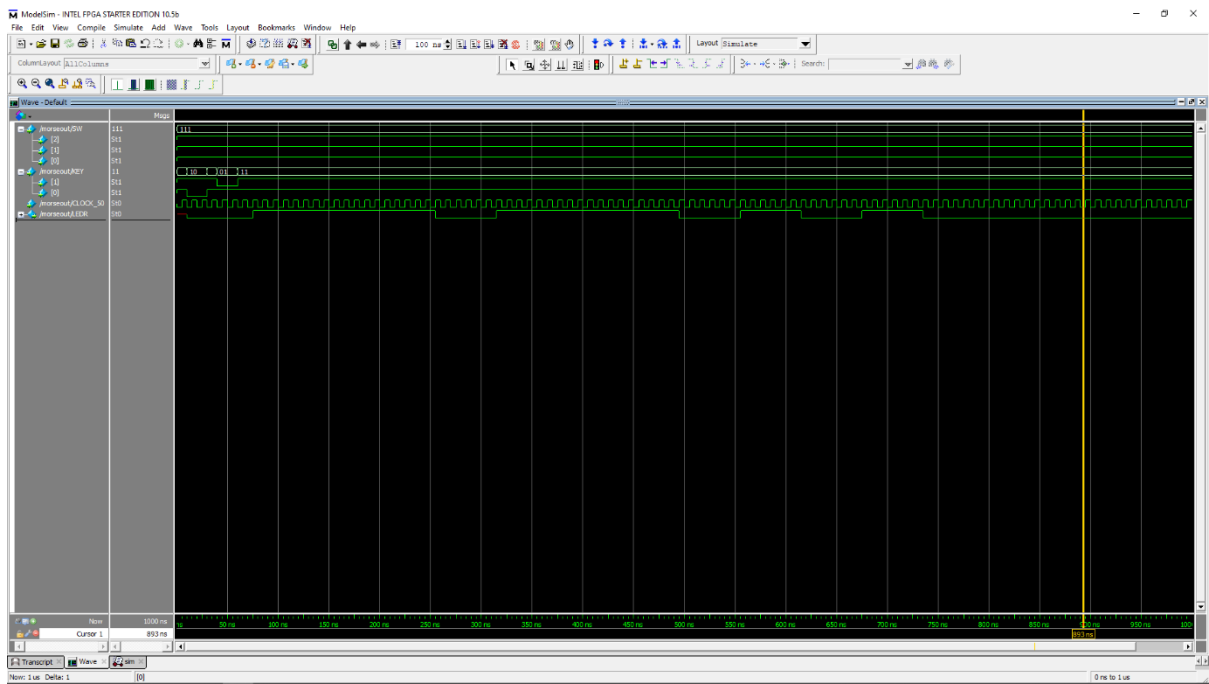
Output W







Output Y



Output Z