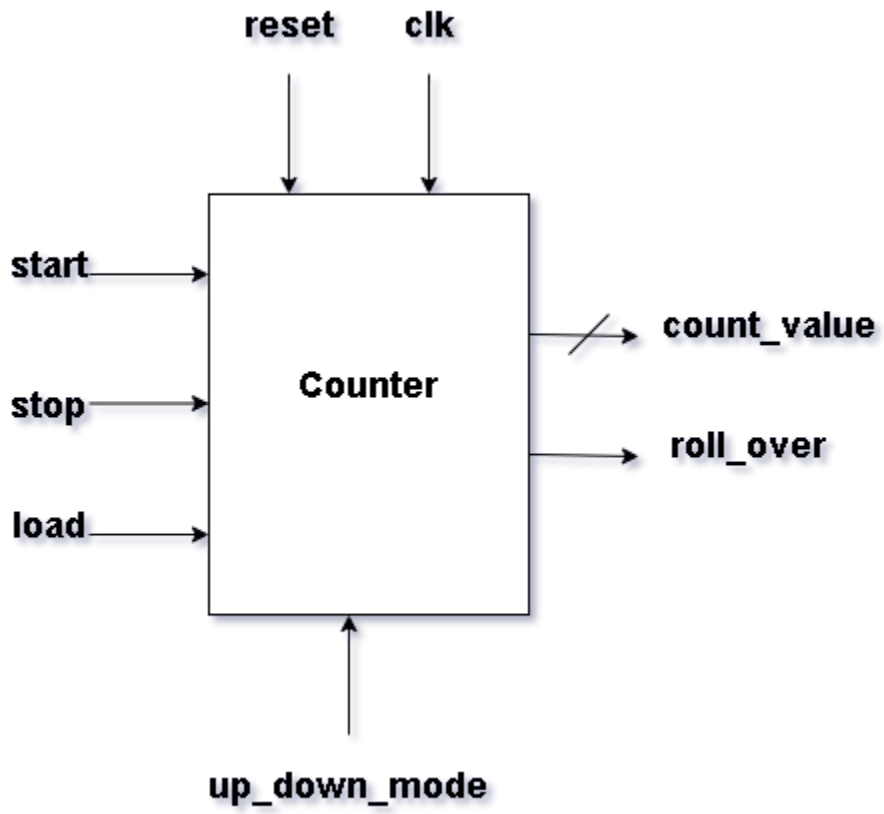
Challenge 1 : Counter design

Problem Statement:

Design a Synchronous decimal counter using Verilog



Counter Design and Implementation has following features 🙂

1) On application of 'reset' the count_value must be
   (a) 00)d - in up mode
   (b) 99)d - in down mode

2) On application of 'load' the count_value must be
   (a) 90)d - in up mode
   (b) 10)d - in down mode

3) On application of 'start' the count_value must be
   (a) increments - in up mode
   (b) decrements - in down mode

4) On application of 'stop' the count_value halts.
   On application of 'start' the counting operation resumes from where it left.

5) The counting must happen for every 1s.

Hint given 🤔

[The clock is 50MHz available from FPGA.
Frequency = 1/Time]

6) On every rollover the buzzer must get active to indicate the roll_over

Hint given 🤔

[Keep the roll_over signal active for 1a to lear it's voice]

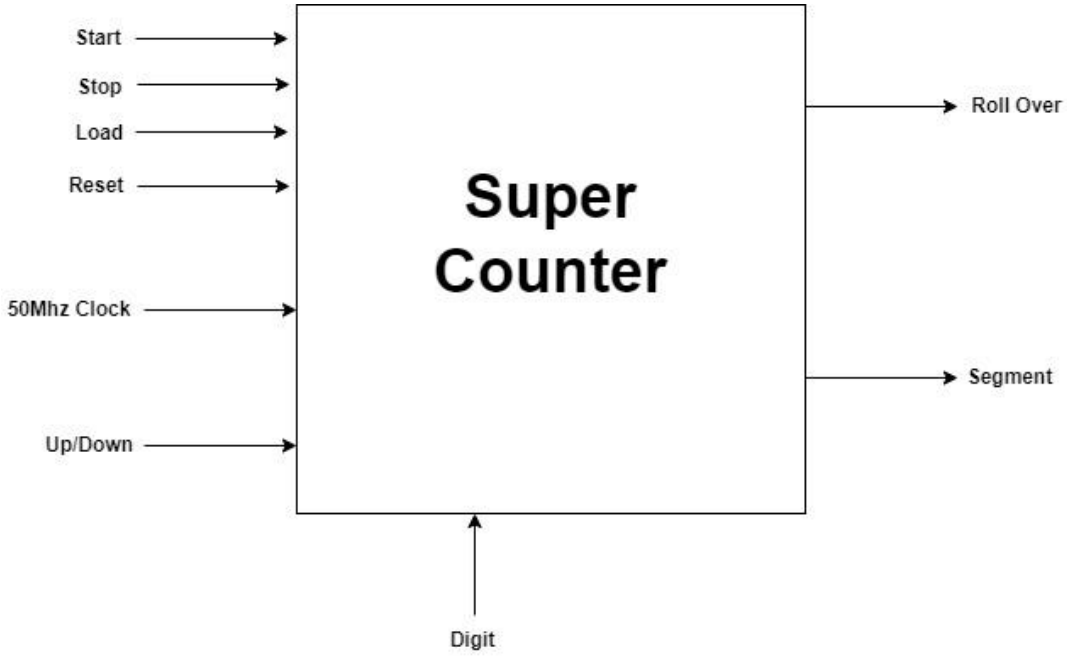
99)d to 00)d in up mode
00)d to 99)d in down mode

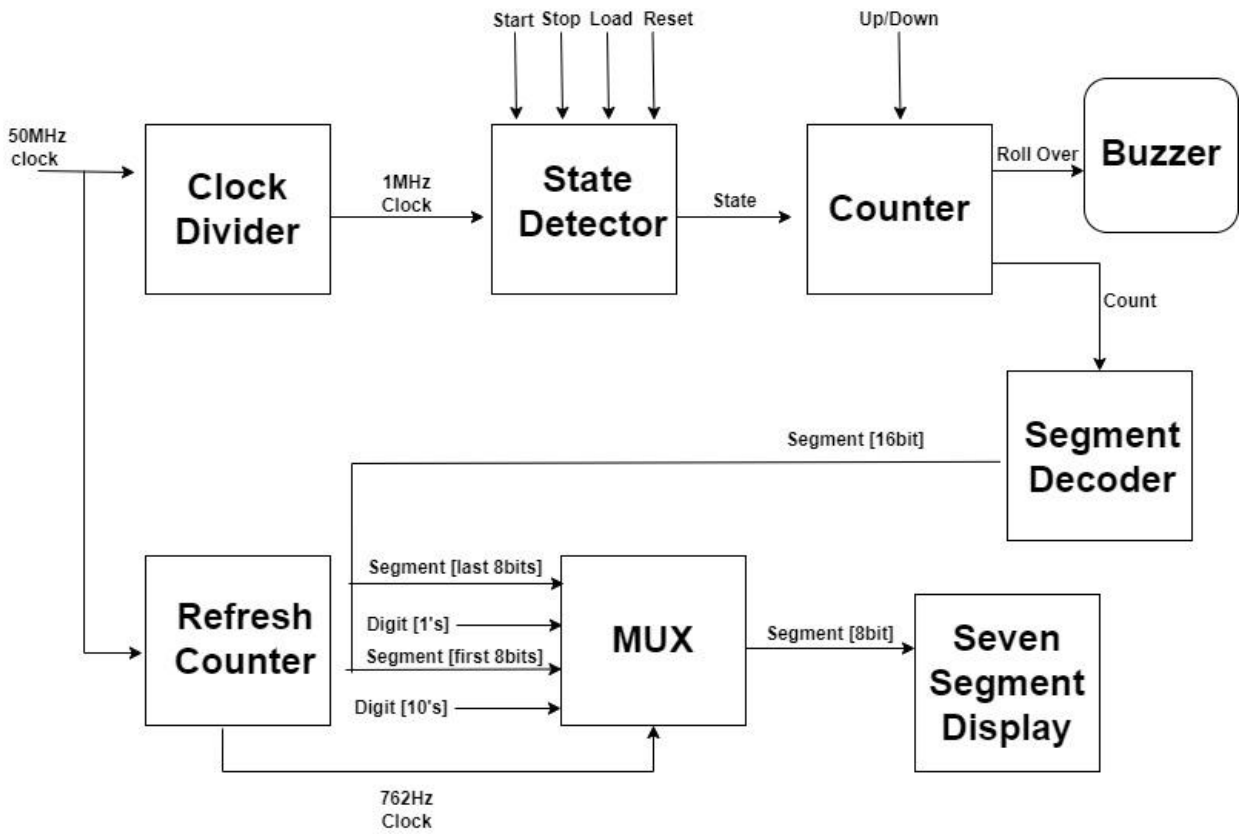
7) Use testbench to verify basic functionality

Hint given 🤔

Don't include feature 5) & 6) while using testbench to verify because 1s in simulation will take hours. So use them during the implementation]

Approach:

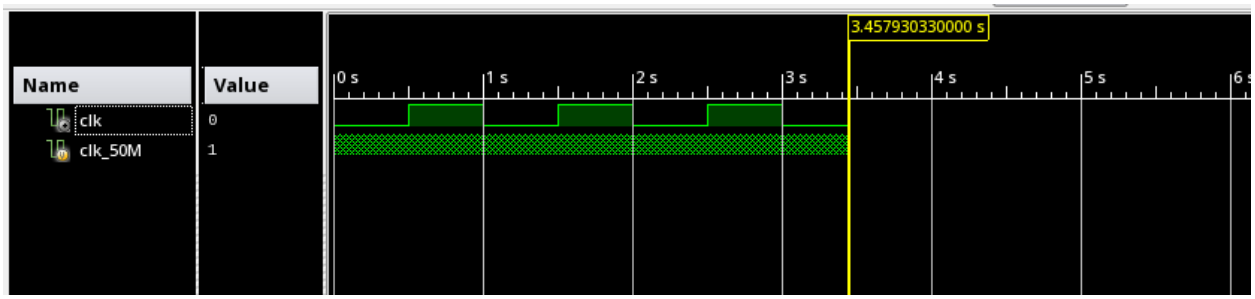

Start
Stop
Load
Reset
50Mhz Clock
Up/Down
Super Counter
Roll Over
Segment
Digit



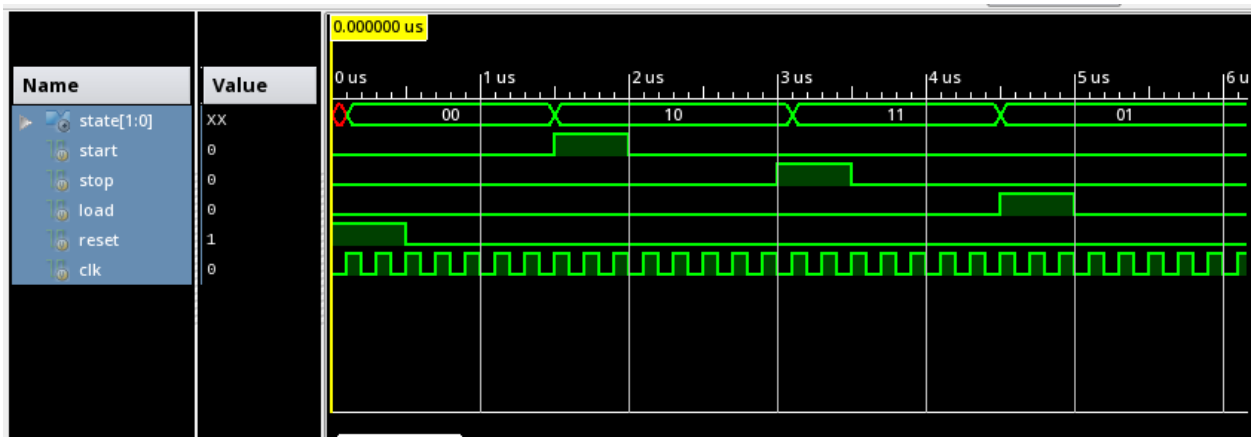
Clock divider: For the counting to happen at every 1s, the 50MHz clock is divided to get 1Hz clock to get time period 1s.
State detector: Input to state detector is 1Hz clock, start, stop, load, reset. For every case a state is selected.
Counter: Selected state is given to the counter which has either up/down count mode using switch. There are two outputs from the counter, rollover and count.
Buzzer: When the rollover is high, buzzer is turned on for 1s (i.e 99-0 for upcount mode and 0-99 for downcount mode).
Segment Decoder: Count output from counter is given to segment decoder and for every count value a 16 bit seven segment code is assigned.
Refresh Counter: To make the segment values visible to eyes, and for displaying separate values in seven segment refresh counter is used.
MUX: When the refresh clock is LOW, the count in one's place is displayed by using the digit for one's place and when the refresh clock is HIGH the count in tens place is displayed in a seven segment display.
Basically MUX takes 16 bit input from segment decoder and displays 8 bit output based on select line which is refresh clock.

Simulation Results:

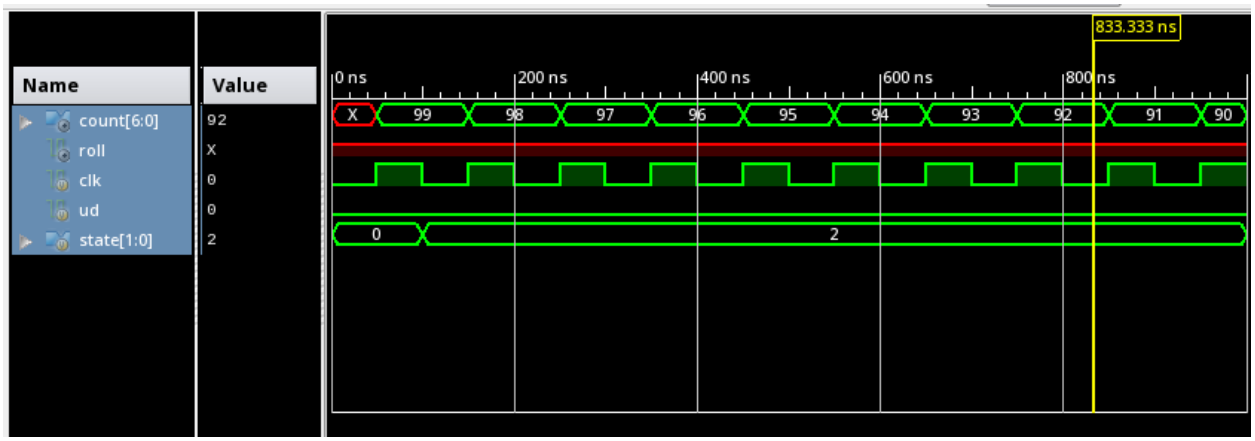
Clock divider



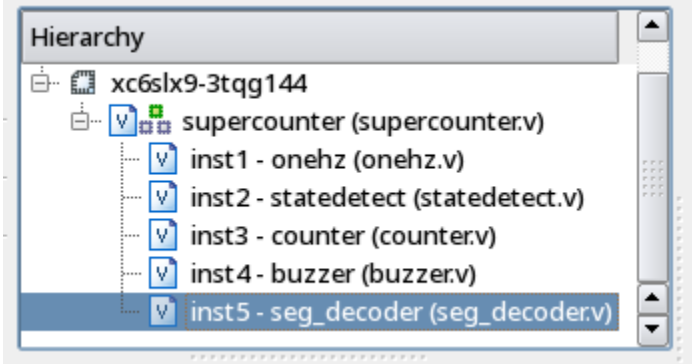
State detector :

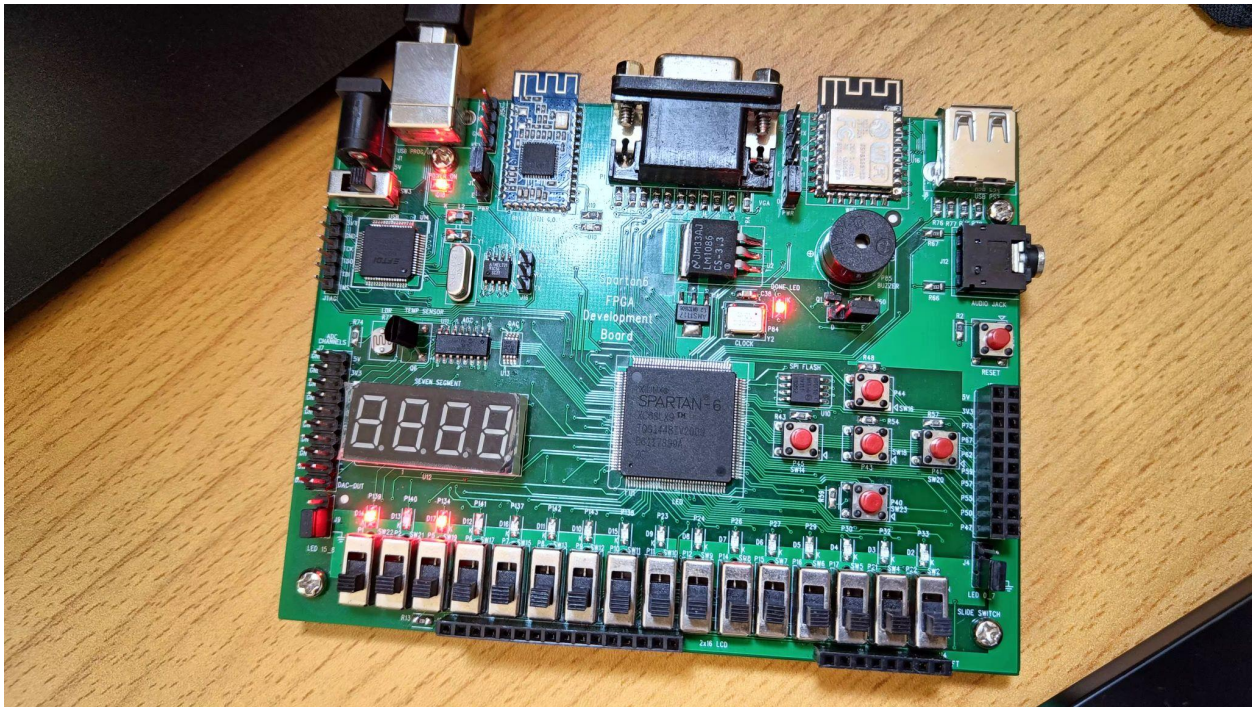


Counter:



FPGA implementation:





Errors:

- Buzzer doesn't turn on.
- Seven segment display does not turn on, instead LED turns on(they have the same pin numbers).