ASSIGNMENT – 6

1) Introduction:

In this lab we have extended the lab work done in lab-3 i.e. we have started by taking the 4- digit display code as building block and then used it to generate a stopwatch with a precision as less as 10^{th} part of a second. For that we have created a time reference basically by using our "clock" and "push buttons" for the three functionalities (start/continue, pause, reset).

2) Implementation design:

For forming a stopwatch using the 4 digit display we have taken the same vectors "refresh_timer" of 20 bits initialized to all 0's and "LED_activation" which is basically nothing but the 18^{th} and 19^{th} bit of the refresh timer. Also we have taken a new counter named "clock_timer" which is nothing but an integer which will get updated only in case of the stopwatch is running and will count only till 10000000 because that is the $1/10^{th}$ part of second in hardware architecture.

The 2 bit values of refresh_timer vector can be any of "00", "01", "10", "11" and thus are the possible values of "LED_activation". And these values are used for selection from the 4 different anodes basically working as a multiplexer.

We have also 4 4-bit vectors named one_tenth_sec, one_sec, ten_secs, minute which are just used for which one of the 4 digits to display at a particular time.

LED_activation	Anode_Activate	which_led
00	0111	minute
01	1011	Ten_secs
10	1101	One_sec
11	1110	One_tenth_sec

Use of buttons and the stopwatch functionalities:

We have taken an enable input from a flip/flop or latch which is set to '1' when button_start is pressed or button_pressed = '1' and is 0 when button_paused

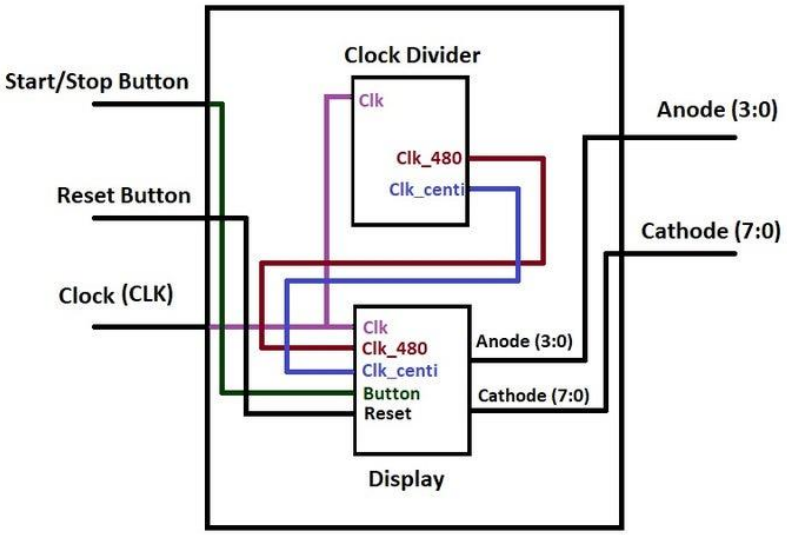
is pressed or button_paused = ‘1’. And reset input comes from the button button_reset.

And our 4 counters are just 4 different N modulo counters and whenever any of those reaches N they are set back to 0.

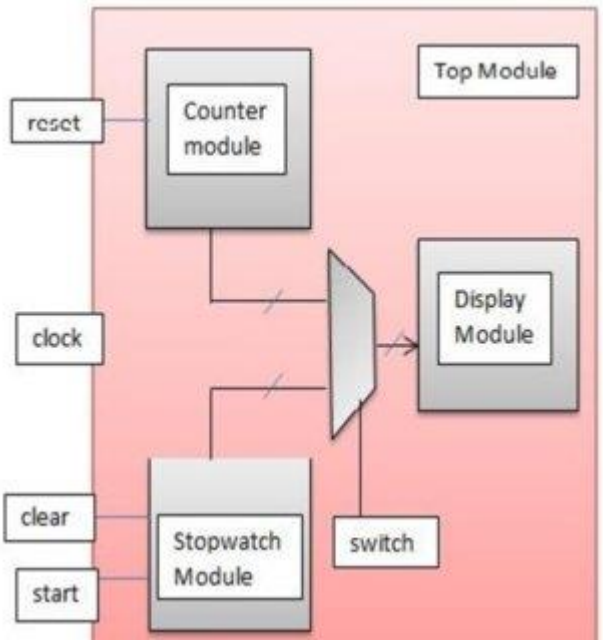
Which_led	Value Of N (modulo counter)
minute	10
Ten_secs	6
One_sec	10
One_tenth_sec	10

Schematic diagram of a stopwatch with 1 digit for minute, 2 digits for second and 1 digit for tenth of a second

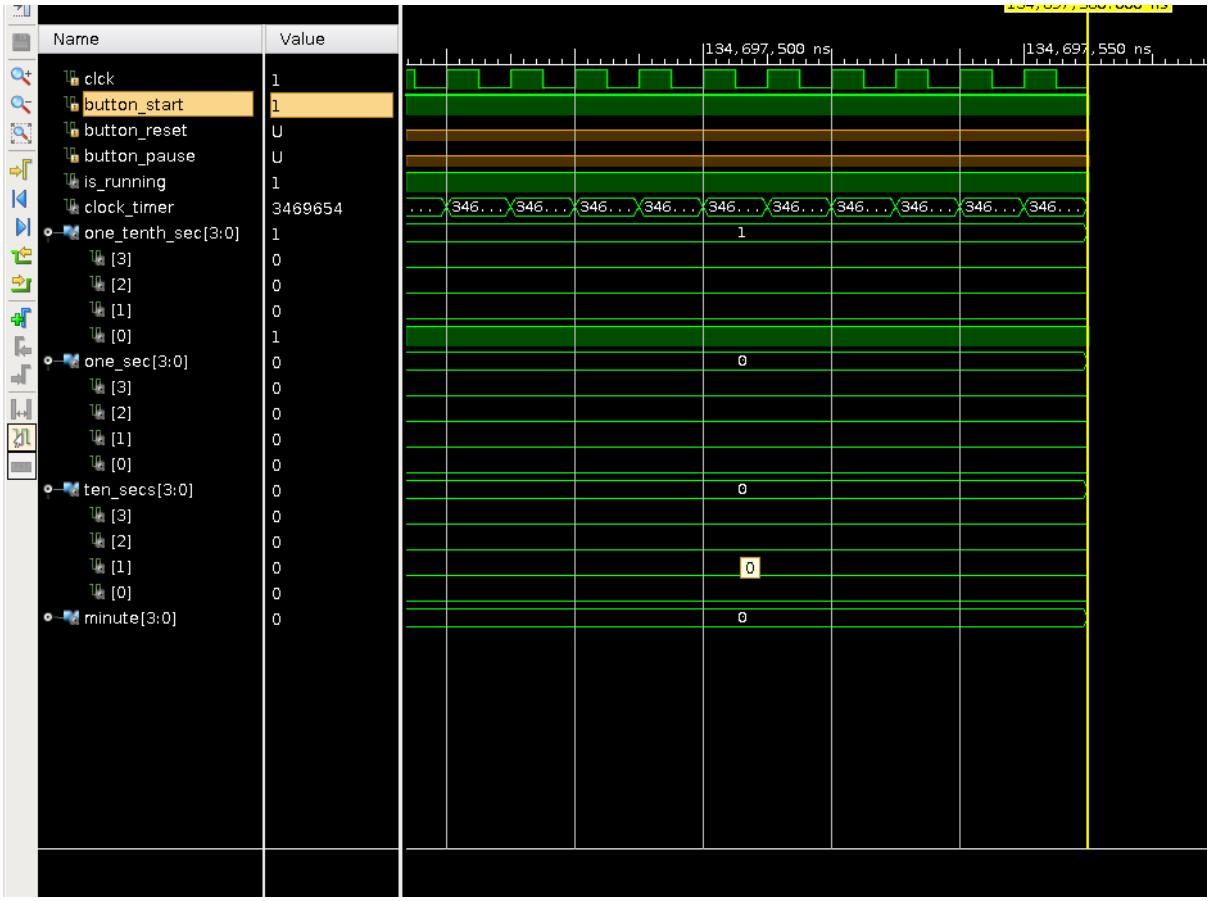
(The diagram is basically not a fully correct visualization of our stopwatch (different frequencies)

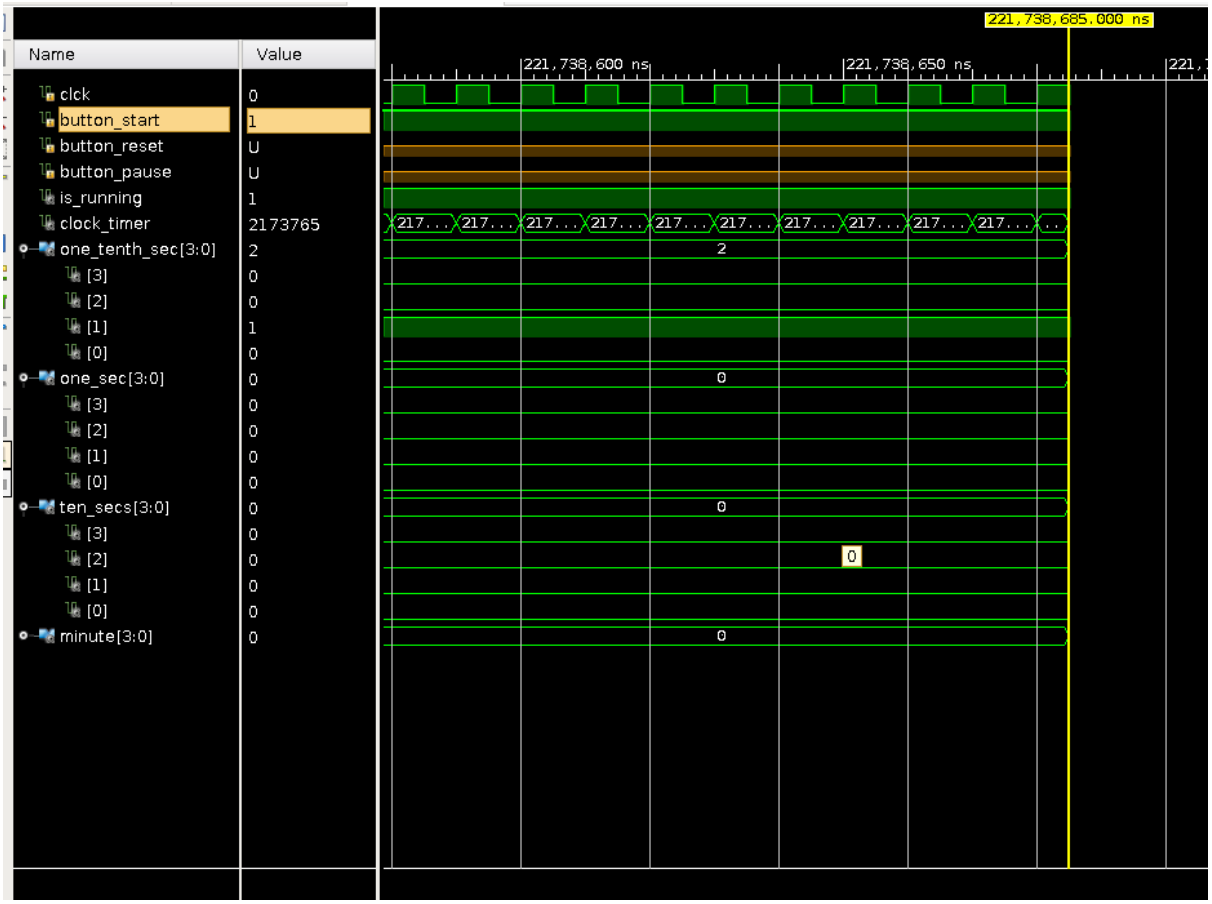


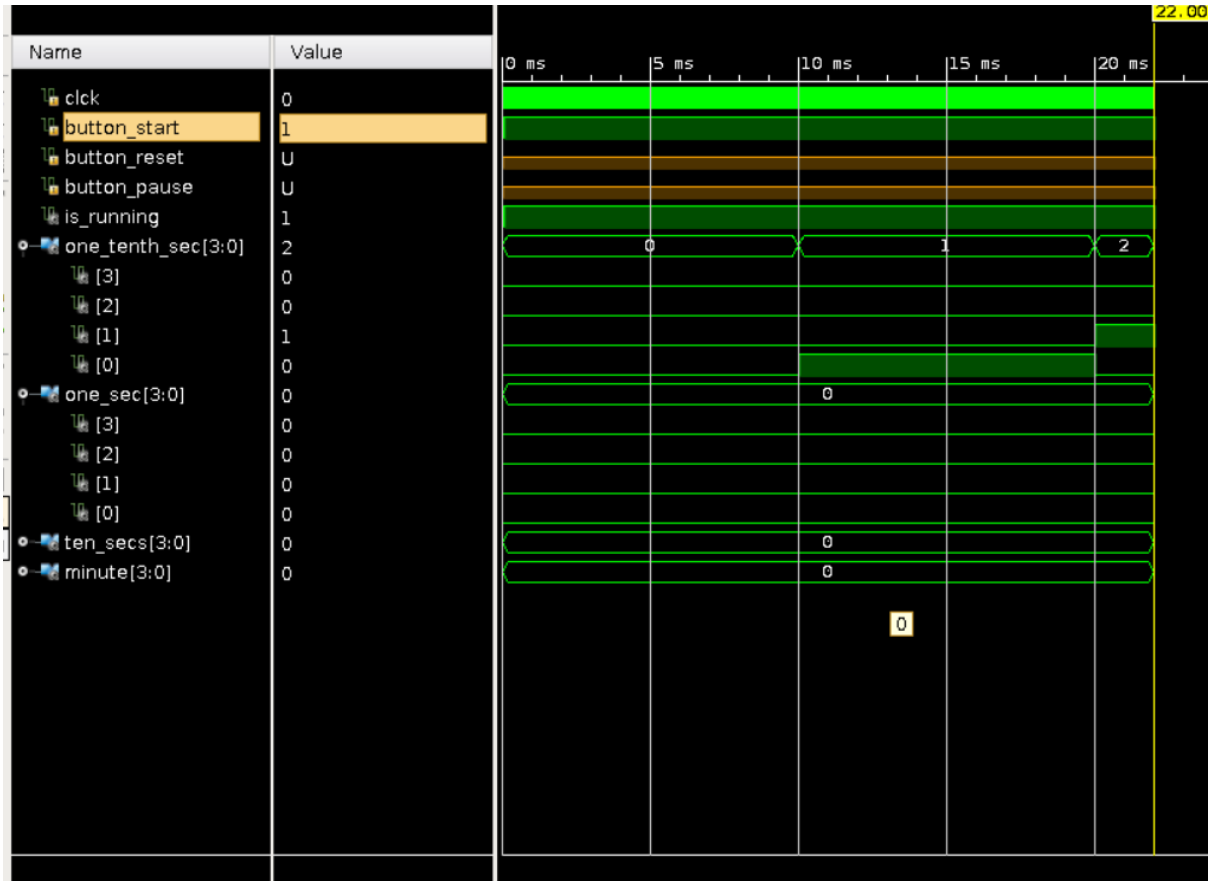
The below figure shows the modules of the stopwatch that we have used which are basically counter, stopwatch and display module.



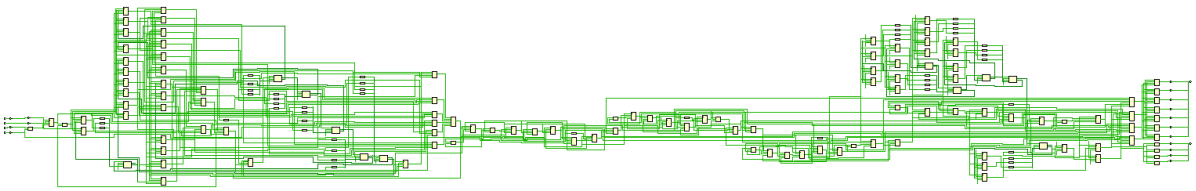
3) Simulation waveform for our stopwatch:




Name
Value
clck
button_start
button_reset
button_pause
is_running
clock_timer
one_tenth_sec[3:0]
[3]
[2]
[1]
[0]
one_sec[3:0]
[3]
[2]
[1]
[0]
ten_secs[3:0]
[3]
[2]
[1]
[0]
minute[3:0]
0
1
U
U
1
2173765
2
0
0
1
0
0
0
0
0
0
0
0
0
0
0
0
0
221,738,600 ns
221,738,650 ns
221,738,685.000 ns
217...
2
0
0
0

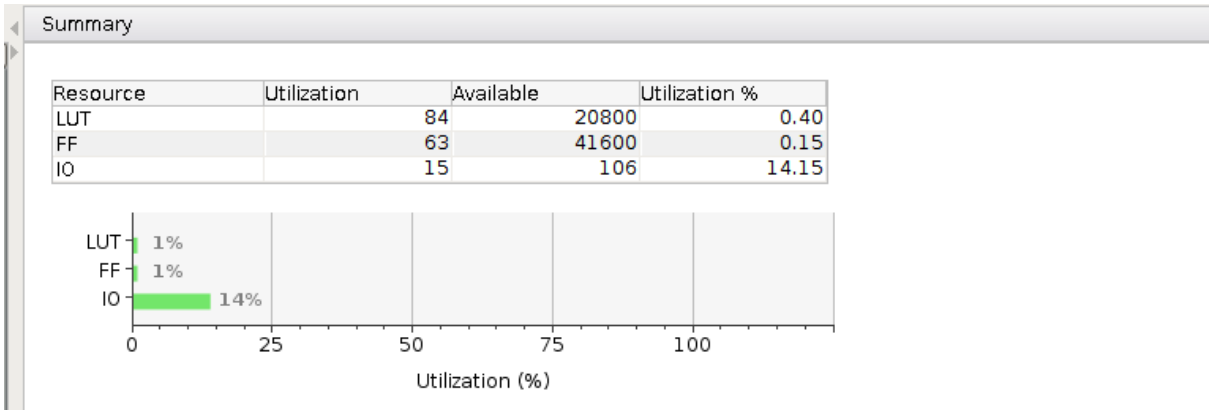

Name
Value
clck
button_start
button_reset
button_pause
is_running
one_tenth_sec[3:0]
[3]
[2]
[1]
[0]
one_sec[3:0]
[3]
[2]
[1]
[0]
ten_secs[3:0]
minute[3:0]
0
1
U
U
1
2
0
0
1
0
0
0
0
0
0
0
0
0
0 ms
5 ms
10 ms
15 ms
20 ms
22.00
0
1
2
0
0
0
0

4) Digital circuit for our stopwatch



5) Resource Utilization:
   a) LUT Memory = 0
   b) LUT logic = 84
   c) DSP = 0
   d) Flip Flops = 63
   e) BRAM = 0

6) Some other relevant details for resource utilization:
   a) Summary:



   b) Primitives:

Primitives

Ref Name	Used	Functional Category
FDRE	63	Flop & Latch
LUT1	48	LUT
LUT4	23	LUT
OBUF	11	IO
CARRY4	11	CarryLogic
LUT2	10	LUT
LUT5	8	LUT
LUT6	5	LUT
LUT3	4	LUT
IBUF	4	IO
BUFG	1	Clock

c) Hierarchy:

Hierarchy

Name	Slice LUTs (20800)	Slice Registers (41600)	Bonded IOB (106)	BUFGCTRL (32)
StopWatch	84	63	15	1