Indian Institute of Technology Bombay
Department of Electrical Engineering
Digital Circuits Lab (EE-214)
End Sem Exam(SET_3), Date: Oct 24, 2021

Timing: 9:00 AM to 12:30 PM   Autumn 2021   Max mark: 30

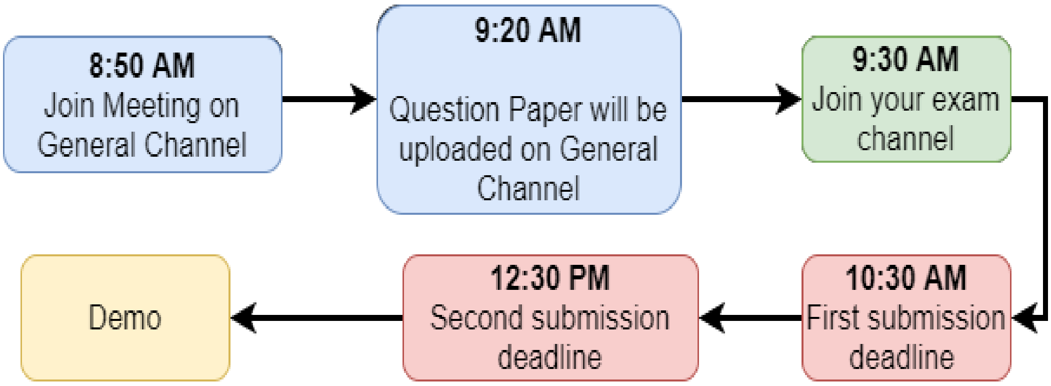

8:50 AM
Join Meeting on General Channel
9:20 AM
Question Paper will be uploaded on General Channel
9:30 AM
Join your exam channel
10:30 AM
First submission deadline
12:30 PM
Second submission deadline
Demo

Figure 1: Exam Flowchart

1 Instructions

1. Use **Behavioural style of modelling**.
2. ***NO TRACEFILE AND SCAN-CHAIN is REQUIRED. HOWEVER KRYPTON BOARD is REQUIRED***
3. You are allowed to use any components of your own VHDL descriptions in the experiments/homework problems so far.
4. ***NOTE: In case any hardware issue or for verification purpose, use the testbench provided. Remember to simulate it lower timer value(in μs) and RTL simulation only . Otherwise it will take 5 hrs to simulate :P***

2 Problem Statement

(a) In this part you have to design a very simple state machine. It has 4 states and each state has it's fixed output.

Table 1: State Table

reset	Input	Present_State	Next_State	Output
1	xxx	any state	RST	00
0	001	any state	timer1	01
0	010	any state	timer2	10
0	100	any state	timer3	11
0	xxx	any state	same state	output of that state

NOTE: x means don't care.
Here is the list of states and their corresponding output.

- RST and output is "00"
- timer1 and output is "01"
- timer2 and output is "10"
- timer3 and output is "11"

State machine has two clocks (50MHz and 1Hz), one reset input slide switch and 3 inputs through slide switches. At the beginning the machine will be in the reset (RST) state.

Here is the top level entity description.

```
entity timer_controller is
port(   inp_switch:in std_logic_vector(2 downto 0);
        reset,clock_50, clock_1:in std_logic;
        out_LED: out std_logic_vector(3 downto 0));
end timer_controller;
```

- inp_switch is 3 bit, and will be given by 3 slide switches.
- reset will be given by slide switch.
- out_LED is on board LED.

clock_50 is the 50 MHz on-board clock(Pin 89), clock_1 is 1 Hz on-board clock(Pin 18). ***NOTE: Pin numbers for switches and LEDs are given in the Page 6.***

(b) In this part you have to design a new module called timer_ckt as described below.

- This module has the input from state machine output and it consists of 3 timers.
- Each timer is basically a counter which will count for a certain amount of time(see Table 2) depending upon state machine output.
- Each timer will have a corresponding LED, which will blink till the timer count is over or forcefully terminated by reset or state machine.
- Another LED will glow steadily if the machine output is "00".
- If the machine output is "00" all the timers will be initialized to their initial value.
- If any of the timers finishes it's counting you have to manually reset the timers using reset switch to use again.

Table 2: Timer and LED specs. Table

State machine output From part 1 FSM	Timer Duration	Timer Duration for (RTL and Gate-level Sim)	LED Specification
00	don't care	don't care	LED 3 Will glow steadily
01	2 sec	$2\mu s$	LED 0 will blink at 1 Hz for 2 sec then off
10	7 sec	$7\mu s$	LED 1 will blink at 1 Hz for 7 sec then off
11	10 sec	$10\mu s$	LED 2 will blink at 1 Hz for 10 sec then off

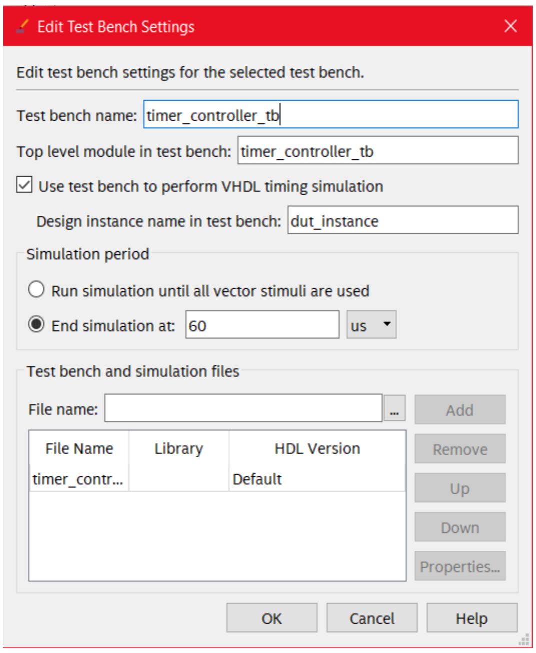
3 Instructions for Solution Upload and Demo

1. (a) Upload the hand-written design on Moodle hand-written submission link. Handwritten design should comprise of **state diagram, state table, block diagram, justifications(whichever is required)**.
   Submit it in a **single zip file (RollNumber_handwd.zip)**. ***(5 Marks)***

   (b) Describe the FSM in VHDL. Find the State Table in State Machine Viewer in Quartus. ***(10 Marks)***

   (c) Describe the timer circuit in VHDL and integrate it with the state machine. The final Design(VHDL Design files, qpf file) should be uploaded in Moodle (VHDL design submission link) by 12:30 PM. Submit it in a **single zip file (RollNumber_vhdl.zip)**. Show the RTL, gate-level simulation, hardware demo to your TA after submission. ***(15 Marks)***

NOTE: In case any hardware issue or for verification purpose, use the test-bench provided. Remember to simulate it lower timer value (in μs) and RTL simulation only . Otherwise it will take 5 hrs to simulate :P
Note: Use the given test-bench for RTL and gate-level simulation and of course given timer value in Table 2



Edit Test Bench Settings

Edit test bench settings for the selected test bench.

Test bench name: timer_controller_tb

Top level module in test bench: timer_controller_tb

☑ Use test bench to perform VHDL timing simulation

Design instance name in test bench: dut_instance

Simulation period

○ Run simulation until all vector stimuli are used

● End simulation at: 60 us

Test bench and simulation files

File name:

File Name	Library	HDL Version
timer_contr...		Default

Figure 2: Test Bench Setting

Use the above testbench settings(**End simulation at:60 us**)

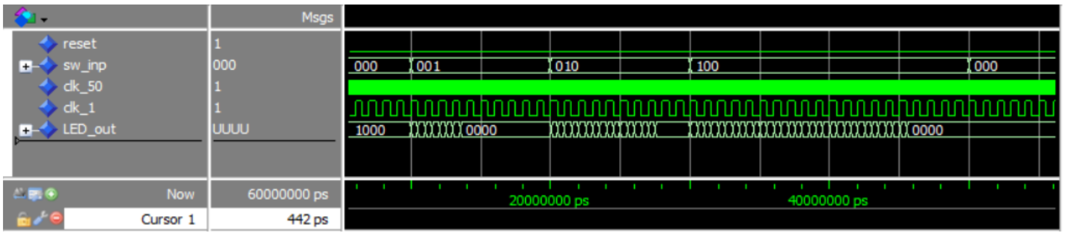


Figure 3: Expected Output

4 Pin Planning

Strictly follow the pin planning

Table 3: Pin Planning Table

Name In Your Design (entity timer_controller)	**Name on Krypton Board**	Pin Number
out_LED(3) (After reset it will glow steadily) (MSB of out_LED in the entity)	LED1	58
out_LED(2) (timer3)	LED6	51
out_LED(3) (timer2)	LED7	50
out_LED(0) (timer1) (LSB)	LED8	49
Reset switch	S1	48
inp_switch(0)	S2	45
inp_switch(1)	S3	44
inp_switch(2)	S4	43

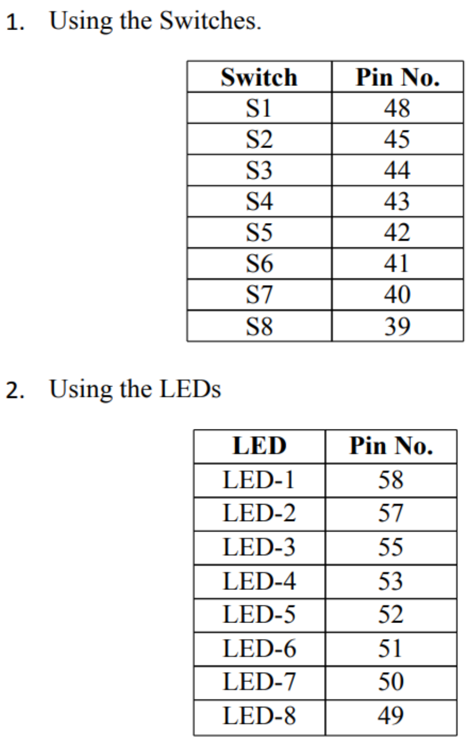
5 Hints

Hints for timer circuit (use port map to instantiate it in the timer_controller)

```
entity timer_ckt is
    port (  clock_1, clock_50, reset : in std_logic;
            LED : out std_logic_vector(3 downto 0);
            timer_inp : in std_logic_vector(1 downto 0));
            -- timer_inp is State Machine output
end entity timer_ckt;
-- Define architecture body
process(clock_50)
        variable timer1 : integer range 0 to 500E6 := 1;
        -- 500E6 means 500 x 10 ^ 6
        -- the above variable will count the number
        -- of clock pulses till it reaches required
        -- number of seconds
        -- Use similar 2 other variables for timer2, timer3
        begin
            if.................. -- Fill your code here
            -- You may have to use multiple nested if condition here
            -- e.g. clock_50'event, reset, condition for increment etc.
            -- reset will initialize the timer1, timer2 and timer3 with
            -- the value 1
                timer1 := timer1 + 1;
            -- this is the syntax to increment timer variable
            -- remember : you have to assign LED here also
            end if;
end process;
------------------------------------------------------
```

Best wishes

6 Appendix



1. Using the Switches.

Switch	Pin No.
S1	48
S2	45
S3	44
S4	43
S5	42
S6	41
S7	40
S8	39

2. Using the LEDs

LED	Pin No.
LED-1	58
LED-2	57
LED-3	55
LED-4	53
LED-5	52
LED-6	51
LED-7	50
LED-8	49

Figure 4: Pin numbers