LAB 1 – SIMULATION

Lớp L04 - Nhóm 03

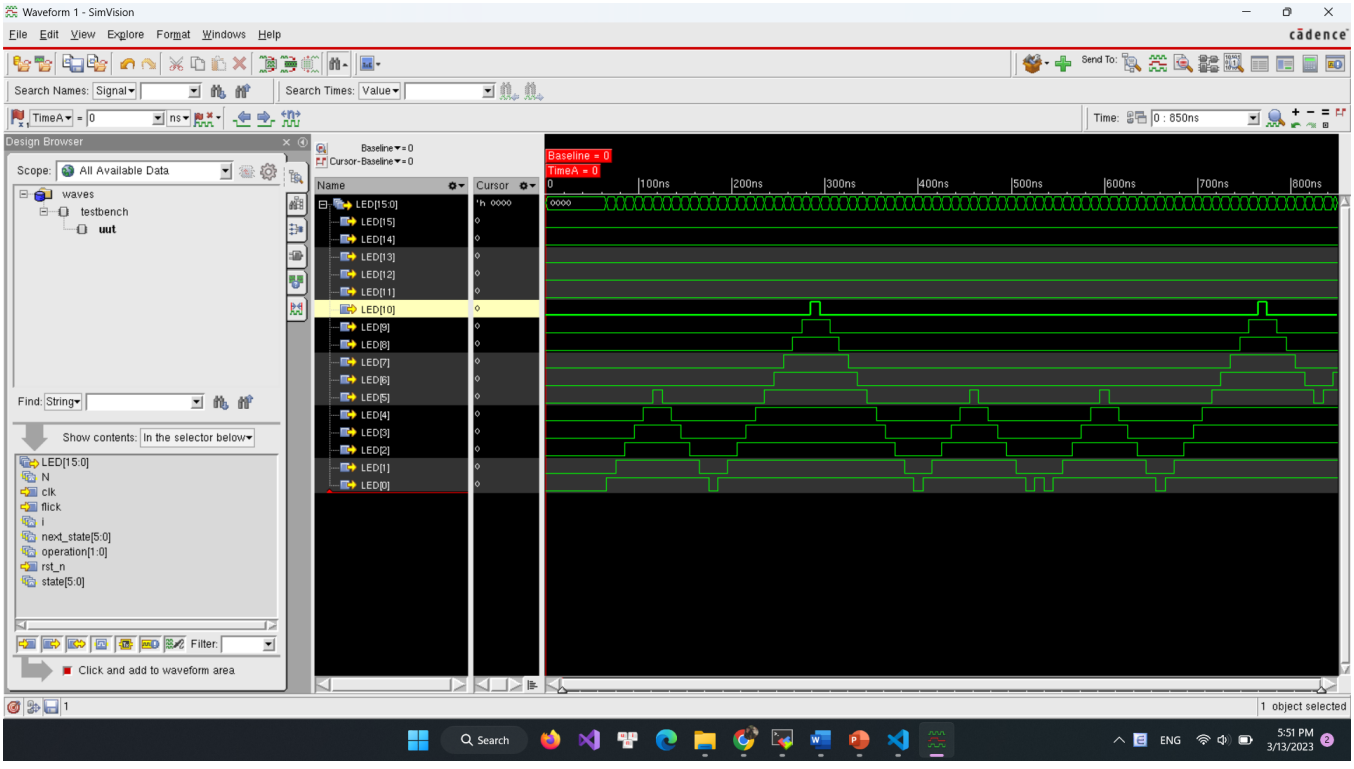
Họ và tên SV	MSSV
Nguyễn Trọng Vinh	2015070
Ngô Nhật Thiên	2014567
Phạm Đình Văn	2015025
Trương Mạnh Khôi	2013536
Chung Thịnh	2012103

Within the testbench file BoundFlasher_tb.v, code for generating test signal pattern is divided into 3 seperate section, with each corresponding to a specific test of the BoundFlasher module.

1. Normal operation without flick signal interruption
2. Flick active at the reset point of STATE_UP_1_10, testing both condition of 5 lights and 10 lights
3. Flick active at the reset point at state STATE_UP_6_16 and state STATE_DOWN_9_5, testing both condition of 5 lights and 10

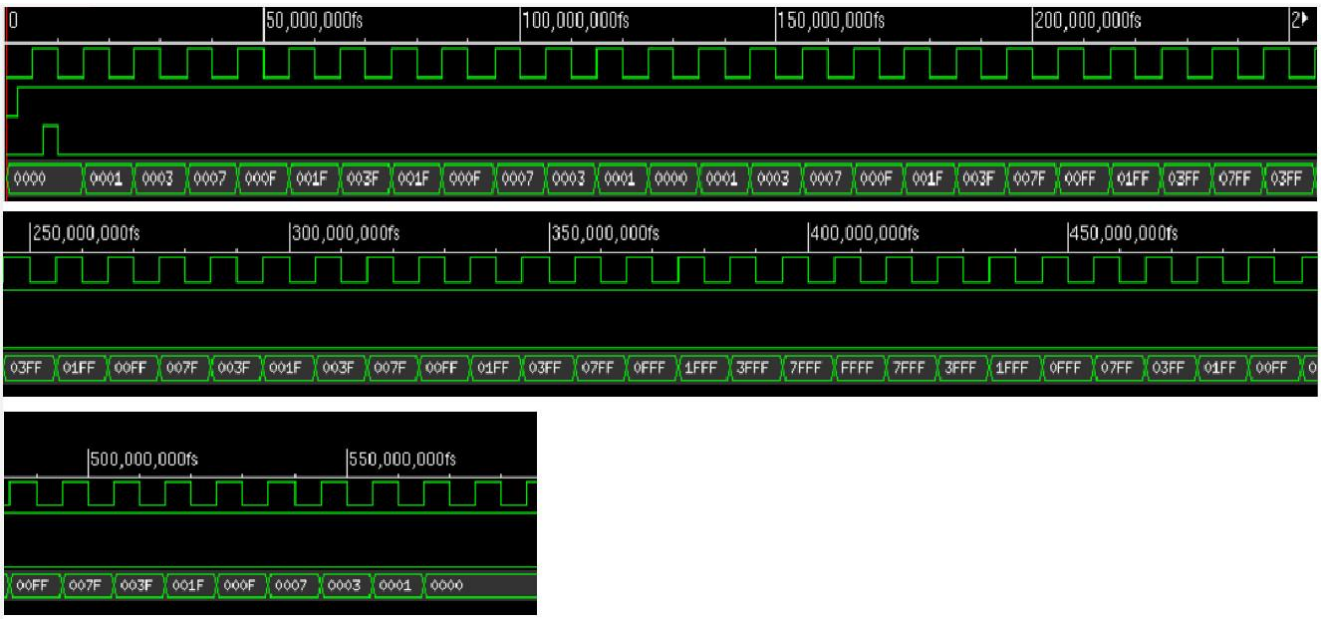
I. Answer for question:

1. Simulation WaveForm on Xcelium:



*) WaveForm Detail:

*)Testcase Normal Flow:



2. Code Testbench :

a. File Testbench:

```
module
boundFlasher_tb;

                    parameter HALF_CYCLE = 5;
                    parameter CYCLE = HALF_CYCLE * 2;

                    reg flick, clk, rst;
                    wire [15:0] led;

                    boundFlasher dut(.flick(flick), .clk(clk), .rst(rst), .LEDs(led));

                    //generate clock
                    always begin
                      clk = 1'b0;
                      #HALF_CYCLE clk = 1'b1;
                      #HALF_CYCLE;
                    end

                    initial begin
                      rst = 0;          // second 0
                      flick = 0;               // second 0
                      #2 rst = 1;       // second 2


                          //INITIAL
                          //STATE_1: 60
                          //STATE_2: 60
                          //STATE_3: 110
                          //STATE_4: 60
                          //STATE_5: 110
                          //FINAL: 160

                      // Test case :
                      #5 flick = 1;     // second 7           (Testcase 1) Normal flow
                      #3 flick = 0;     // second 10

                      #580 flick = 1; // second 590   (Testcase 6) Flick signal to repeat
                  the process
                      #3 flick = 0;     // second 593

                          #173 flick = 1; // second 766 (Testcase 2) Flick signal at L5 in
                  state 3
                          #3 flick = 0;
```

```
        #139 flick = 1; //second 908 (Testcase 8) Flick signal at
between L5 and L10 in state 3
        #3 flick = 0;

        #25 flick = 1; // second 936 (Testcase 3) Flick signal at L10 in
state 3
        #3 flick = 0;

        #288 flick = 1; //second 1227 (Testcase 4) Flick signal at L5 in
state 5
    #3 flick = 0;

        #70 flick = 1; //second 1300 (Testcase 5) Flick signal at L10 in
state 5
        #3 flick = 0;

        #343 flick = 1; // second 1646     (Testcase 6) Flick signal to
repeat the process
        #3 flick = 0;

        #100 flick = 1; // second 1749 (Testcase 7) Flick signal at any
time slot (not kickback point in turning off leds state)
        #3 flick = 0;

        #100 rst = 0; // second 1852 (Testcase 9) Reset signal at any
time slot
        #3 rst = 1;



        #3 flick = 1; //second 1858 (Testcase 6) Flick signal to repeat
the process
        #3 flick = 0;

        #55 flick = 1; //second 1916 (Testcase 10) Flick signal at
initial state
        #3 flick = 0;

        #167 rst = 0; flick = 1; // second 2086 (Testcase 11) Reset
signal with flick signal at kickback point in turning off leds state
        #3 flick = 0;
        #3 rst = 1;
```

```
    ////////////////////////////
    #(CYCLE*45) $finish;
  end

  // To see waveform in online EDA playground only
//  initial begin
//    $dumpfile("debug.vcd");
//    $dumpvars;
//  end
  /////


initial begin
    $recordfile("waves");
    $recordvars("depth=0", boundFlasher_tb);
end


endmodule
```

***) Explain:**

+) **On Normal Flow**, display leds according to the normal patten without reset flick signal (flick = 1 in the entrire run).

+) **Flick signal at L5 in state 2,** When the kickback point L5 is off at state 2 and if flick signal triggers, system goes back to the previous state (state 1), turning on leds to L15.

+) **Flick signal at L5 in state 4**, When the kickback point L5 is off at state 2 and if flick signal triggers, system goes back to the previous state (state 3), turning on leds to L10.

+) **Flick signal at L0 in state 4,** When the kickback point L0 is off at state 2 and if flick signal triggers, system goes back to the previous state (state 3), turning on leds to L10.

+) **Flick signal to repeat the process**, after a complete flow, flick signal to check the process can be repeated from state 1 not state 6.

+) **Flick signal at any time slot (not kickback point in turning off leds state)**, If flick signal takes effect in those states then the code is wrong, else then everything is OK.

+) **Flick signal at between L5 and L0 in state 4, Between the kickback point L5 and L0,** trigger flick signal to test if there is any state transition, if yes, then the code is wrong, else then the code working normally.

+) **Reset signal at any time slot, whenever reset signal is 0**, the system reset immediately to the Initial State, all leds are off.

+) **Flick signal at final state**, The kickback point inactive at final state

+) **Reset signal with flick signal at kickback point in turning off leds state,** Both reset signal and flick signal trigger at the same time, system must go back to Initial state.