ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH

THIẾT KẾ VI MẠCH

BÁO CÁO LAB 1

GVHD: Huỳnh Phúc Nghị
SV: Ngô Quang Hải
SV: Nguyễn Khoa Nam
SV: Trịnh Cao Thắng
SV: Võ Đăng Thi
SV: Nguyễn Anh Tuấn

TP. HỒ CHÍ MINH, THÁNG 3/2023

Mục lục

1 File Design

```
module bound_flasher(clock, reset, flick, led);

input clock, reset, flick;

output [15:0] led;

//input ports data type
wire clock, reset, flick;

//output ports data type
reg [15:0] led;

//internal parameters
parameter INIT= 3'd0,
        S1=3'd1,
        S2=3'd2,
        S3=3'd3,
        S4=3'd4,
        S5=3'd5,
        S6=3'd6;

//internal variable
reg [2:0] state;
reg [3:0] counter;  //range 0->15 for 16 led
//code start here
always@(posedge clock or negedge reset)
begin
    if (reset == 1'b0) begin
        state <= INIT;
        led <= 16'b0000000000000000;
        counter <= 4'b0000;
    end
    else begin
        case(state)
            INIT: if(flick == 1'b1) begin
                    state <= S1;
                    end
            S1: begin
                if(counter == 4'd5) begin
                    state <= S2;
                    led[counter] <= 1'b1;
                    end
                else begin
                    led[counter] <= 1'b1;
                    counter <= counter + 1;
                end
                end
```

```
48              S2 : begin
49                  if(counter == 4'd0) begin
50                      state <= S3;
51                      led[counter] <= 1'b0;
52                      end
53                  else begin
54                      led[counter] <= 1'b0;
55                      counter <= counter - 1;
56                  end
57                  end
58              S3: begin
59                  if(counter == 4'd5 && flick == 1'b1) begin
60                      led[counter] <= 1'b1;
61                      state <= S2;
62                      end
63                  else if(counter == 4'd10 && flick == 1'b1) begin
64                      led[counter] <= 1'b1;
65                      state <= S2;
66                      end
67                  else if(counter == 4'd10) begin
68                      led[counter] <= 1'b1;
69                      state <= S4;
70                      end
71                  else begin
72                      led[counter] <= 1'b1;
73                      counter = counter + 1;
74                  end
75                  end
76              S4 : begin
77                  if(counter == 4'd5) begin
78                      led[counter] <= 1'b0;
79                      state <= S5;
80                      end
81                  else begin
82                      led[counter] <= 1'b0;
83                      counter <= counter - 1;
84                  end
85                  end
86              S5: begin
87                  if(counter == 4'd10 && flick == 1'b1) begin
88                      led[counter] <= 1'b1;
89                      state <= S4;
90                      end
91                  else if(counter == 4'd15) begin
92                      led[counter] <= 1'b1;
93                      state <= S6;
94                      end
95                  else begin
96                      led[counter] <= 1'b1;
```

```
                    counter = counter + 1;
                end
                end
            S6: begin
                if(counter == 4'd0) begin
                    led[counter] <= 1'b0;
                    state <= INIT;
                    end
                else begin
                    led[counter] <= 1'b0;
                    counter <= counter - 1;
                end
                end
            default: state <= INIT;
        endcase
    end
end

endmodule
```

Program 1: File Design

2 File Testbench

```
module bound_flasher_tb;

  //Test input signal
  reg clock, reset, flick;

  //Test output signal
  wire [15:0] led;

  bound_flasher dut(
  .clock(clock),
  .reset(reset),
  .flick(flick),
  .led(led)
  );

  always #5 clock=~clock;

  initial begin
  clock = 0;
  reset = 0;
  flick = 0;

  #10;
  reset = 1;
```

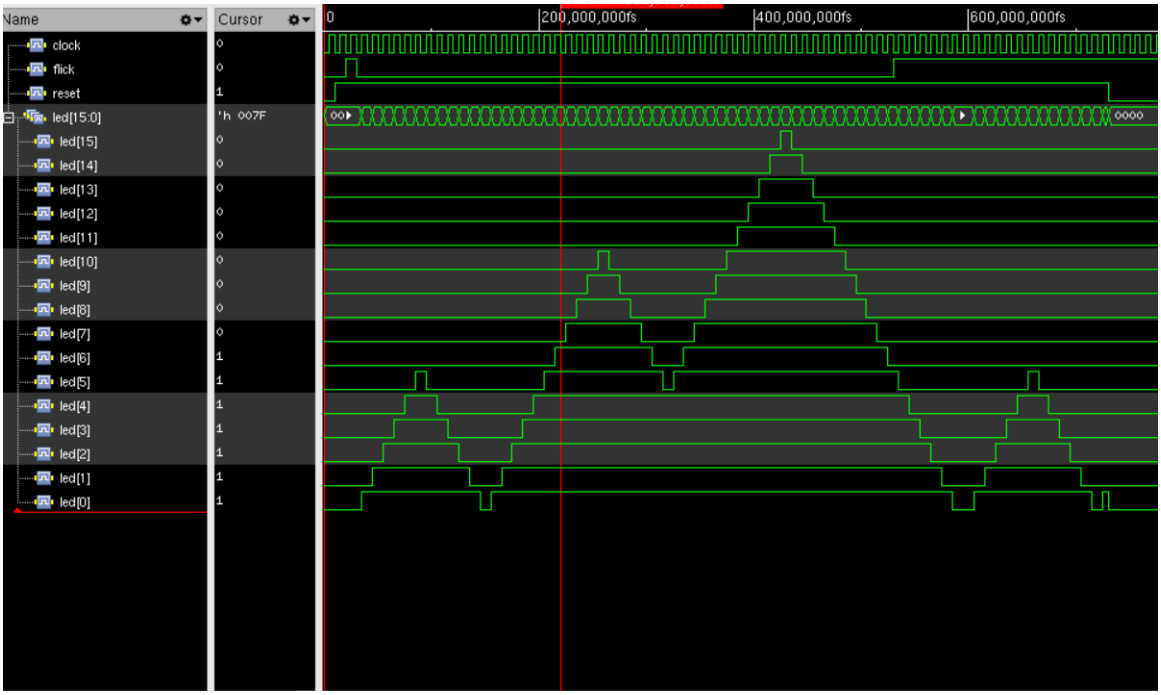
```
25
26  //Test case 1: Normal
27  #10
28  flick = 1;
29  #10
30  flick = 0;
31  #500;
32  flick = 1;
33  #200;
34  reset = 0;
35
36  // Test case 2: Start and flick=1 at kickback point
37   #50
38   reset = 1;
39   #10;
40   flick = 1; //flick at kickback point 5
41   #200;
42   flick = 0;
43   #150;
44   flick = 1; //flick at kickback point 10
45   #200;
46   flick = 0;
47   #50;
48   flick = 1;
49   #200;
50   reset=0;
51   #50;
52
53   // Test case 3: Start Led and assert reset
54   reset = 1;
55   #10
56   flick = 1;
57   #300;
58   reset = 0;
59   #50;
60   reset = 1;
61   #10;
62   flick = 1;
63   #10;
64   flick = 0;
65   #250;
66   reset = 0;
67   #50
68   reset = 1;
69   #10
70   flick =1 ;
71   #50
72
73
```

```
74   $finish;
75  end
76
77  initial begin
78   $recordfile ("waves");
79   $recordvars ("depth=0", bound_flasher_tb);
80  end
81
82 endmodule
```

Program 2: File Testbench

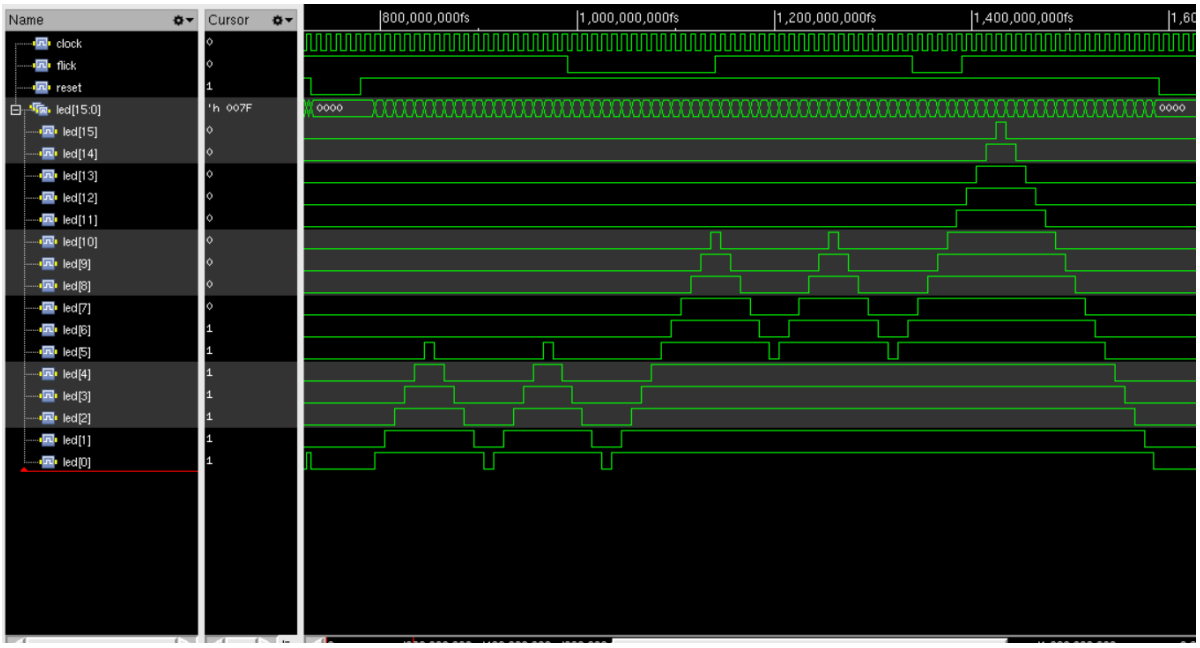
3 Kết quả mô phỏng

Trường hợp 1: không có reset và flick=1 tại điểm kickback point.



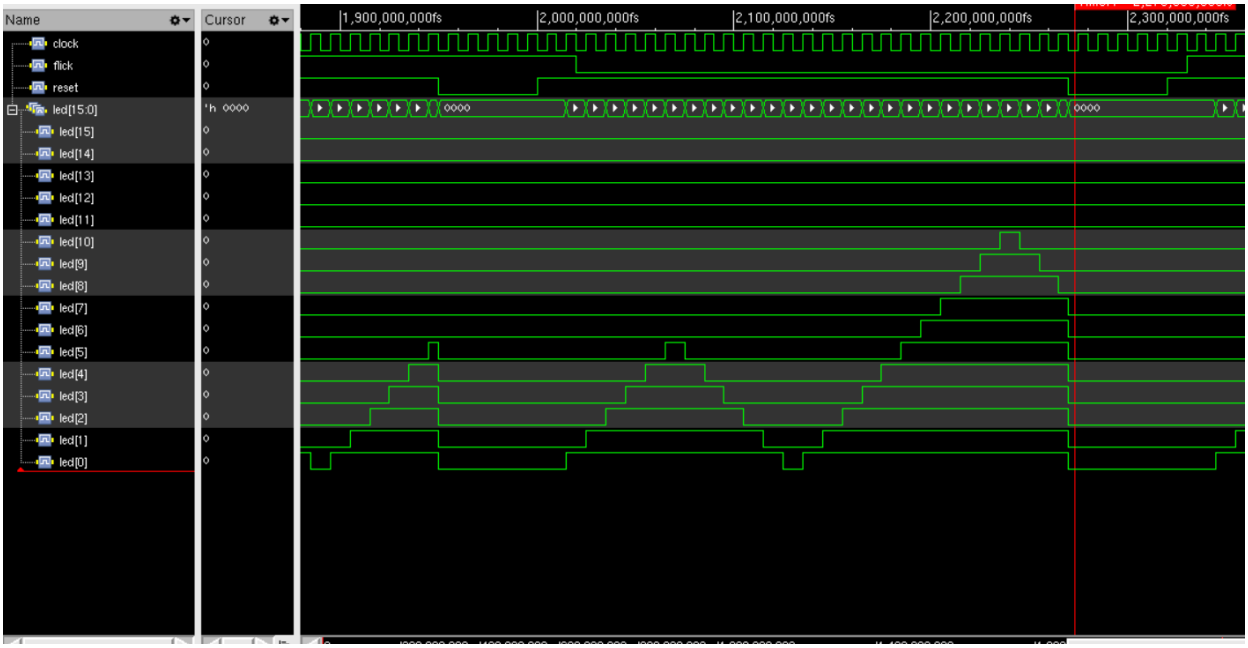
Hình 1: *Trường hợp 1*

Trường hợp 2: flick=1 tại các điểm kickback point.



Hình 2: *Trường hợp 2*

Trường hợp 3: Reset khi đèn đang bật



Hình 3: *Trường hợp 3*

4 Link drive

Link drive file design và testbench:
https://drive.google.com/drive/folders/1CHiy98PfJLoF4MOPfGvTGzc1ADz0cpvq?usp=sharing