HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

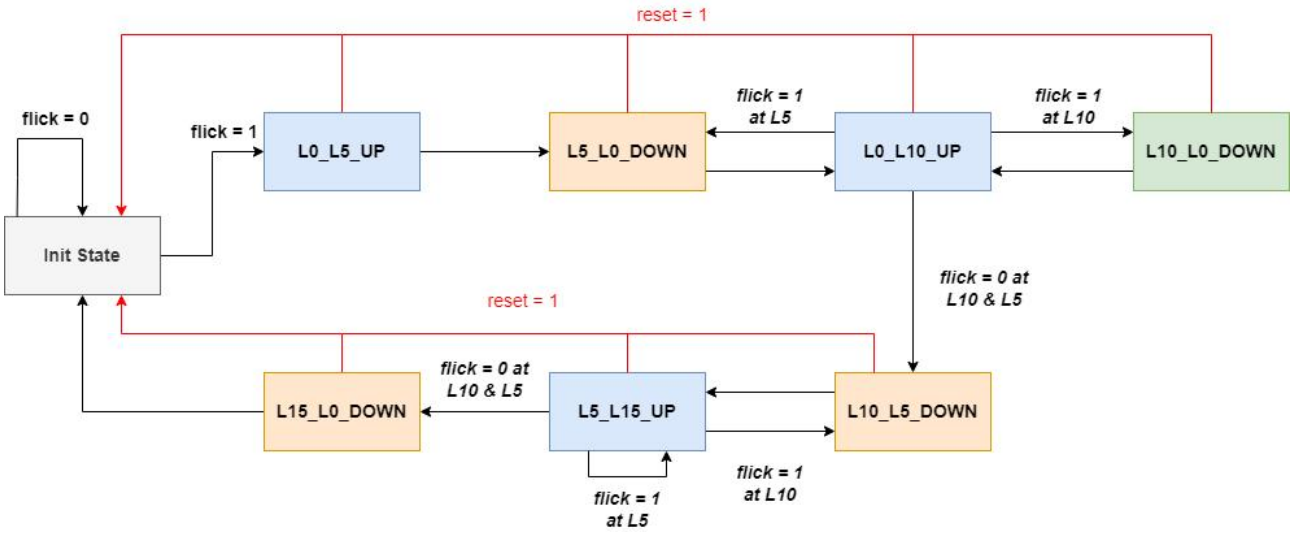
LSI LOGIC DESIGN

LAB 1 - SIMULATION

Giảng viên hướng dẫn: Huỳnh Phúc Nghị
Sinh viên: Trần Minh Trí - 1910637
Nguyễn Lê Gia Hinh - 2011213
Mai Lê Cường - 2012764
Lê Thanh Dương - 2012883
Lê Thanh Tiến - 1912196

- 03/2023 -

I) FINITE STATE MACHINE:

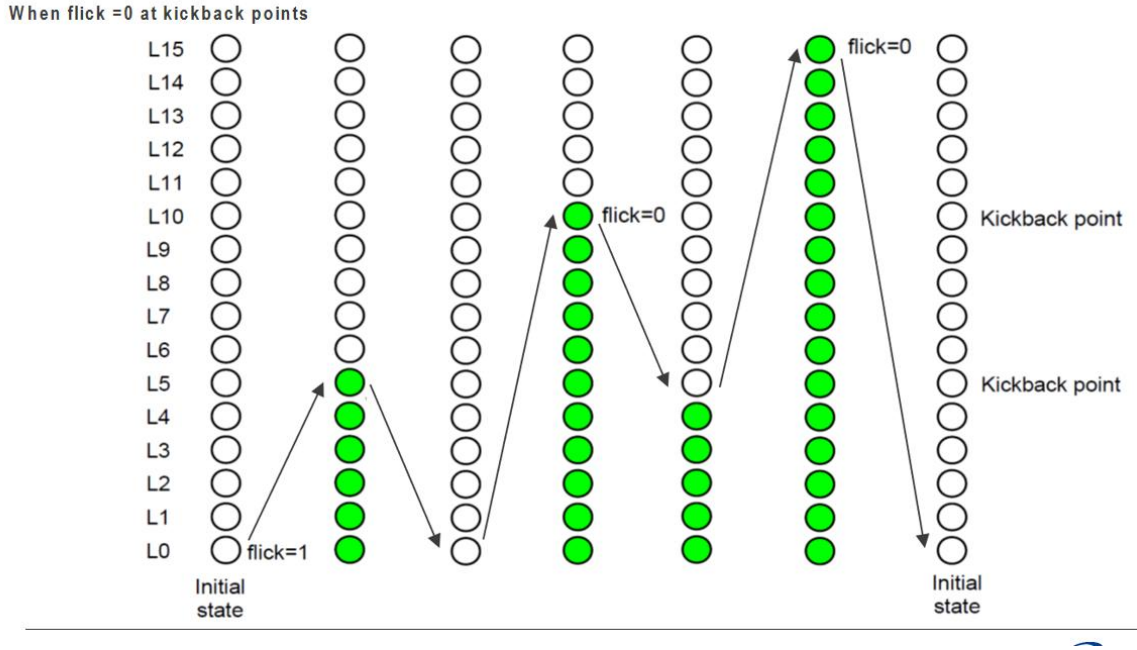


!!! Lưu ý: Ta có thêm 1 trường hợp phụ là L5_OFF. Khi ta ở trạng thái L5_L15_up và khi đèn L5 sáng, nếu flick = 1 thì state_next là L5_OFF và đèn L5 sẽ được tắt sau đó ta sẽ tiếp tục trạng thái L5_L15_up

STT	STATE	DESCRIPTION
1	Init State	All lamps are OFF
2	L0_L5_UP	The lamps are turned ON gradually from LED[0] to LED[5]. After finish, the LED[0] to LED[5] are ON.
3	L5_L0_DOWN	The LEDs are turned OFF gradually from LED[5] to LED[0]. After finish, all LEDs are OFF.
4	L0_L10_UP	The LEDs are turned ON gradually from LED[0] to LED[10]. After finish, the LED[0] to LED[10] are ON.
5	L10_L0_DOWN	The LEDs are turned OFF gradually from LED[10] to LED[0]. After finish, all LEDs are off.
6	L15_L0_DOWN	The LEDs are turned OFF gradually from LED[15] to LED[0]. After finish, all LEDs are off.
7	L5_L15_UP	The LEDs are turned ON gradually from LED[5] to LED[10]. After finish, all the LEDs are ON.
8	L10_L5_DOWN	The LEDs are turned OFF gradually from LED[10] to LED[5]. After finish, the LED[0] to LED[4] are ON.
9	L5_OFF	Đã được giải thích ở trên.

II) GIẢI THÍCH CÁCH HIỆN THỰC:

Specification



- Quan sát yêu cầu đề, ta thấy có thể dễ dàng giải được yêu cầu đề bằng cách:

+ Ta coi từng đèn như 1 bit. Ta sẽ tạo **1 reg vector 16 bit (outut reg [15:0] led_out)** tượng trưng cho 16 đèn led. Và sau đó ta sẽ tạo các STATE của FSM bằng cách dùng **parameter. Đồng thời ta khởi tạo giá trị ban đầu cho vector của chúng ta trong khối initial(phục vụ cho việc mô phỏng).**

```
    output reg [15:0] led_out;
    reg [15:0] tmpLed; // dung de luu trang thai cua den doi xung clock roi moi gan vo led_out
    reg [2:0] state_curr;
    reg [2:0] state_next;
            // Ta khai bao cac state co the xay ra
     parameter INIT_MAP = 16'b0;
    parameter STATE_INIT = 3'b000;
    parameter L0_L5_up = 3'b001;
    parameter L5_L0_down = 3'b010;
    parameter L0_L10_up = 3'b011;
    parameter L10_L5_down = 3'b100;
    parameter L5_L15_up = 3'b101;
    parameter L15_L0_down = 3'b110;
   parameter L5_OFF = 3'b111;
 initial begin
            led_out = INIT_MAP;
            tmpLed = INIT_MAP;
            state_curr = STATE_INIT;
 end
```

+ Đối với các TH đèn sáng dần, ta sẽ tiến hành dịch trái bit của **vector reg led_out**, sau đó ta tiến hành **bitwise or với 16'b1.** Tương tự, với TH đèn tắt dần, ta sẽ dịch phải từng bit của **led_out.**

```
always@(posedge clock)
    case(state_curr)
     STATE_INIT: tmpLed =  INIT_MAP;
        L0_L5_up, L5_L15_up, L0_L10_up: tmpLed = (led_out << 1)|16'b1;
     L10_L5_down, L15_L0_down, L5_L0_down,L5_OFF: tmpLed = led_out >> 1;
     default: tmpLed = INIT_MAP;
     endcase
```

+ Reg tmpLed sẽ là nơi lưu giá trị mới của led_out trước khi đợi xung clock lên để gán vào led_out

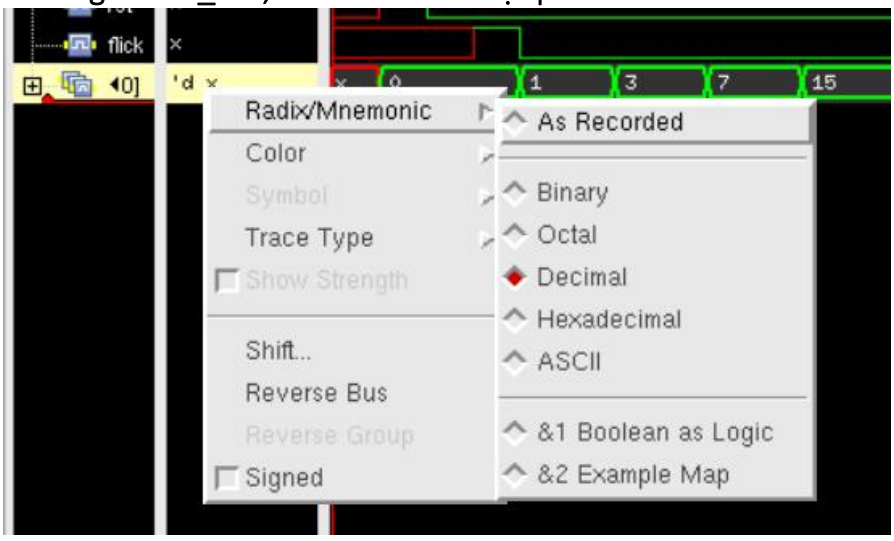
```
always @(posedge reset or posedge clock)
    begin
        if(reset) led_out <= INIT_MAP;
        else led_out <= tmpLed;
    end
```

+ Công việc còn lại của ta chỉ còn là hiện thực FSM(lưu ý 2 state L0_10_up và L5_15_up)

III) KIỂM THỬ

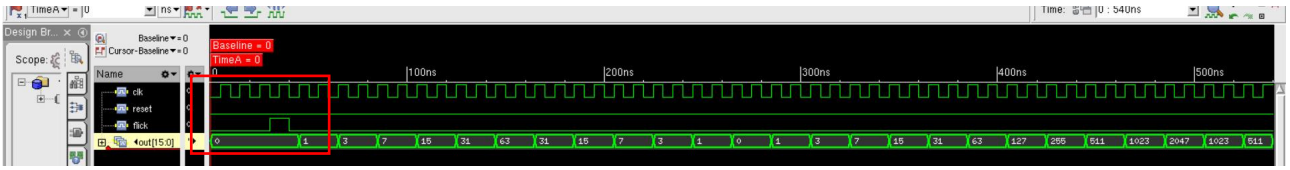
!!! Để dễ quan sát ta sẽ tiến hành đổi các số nhị phân về thập phân như sau:

+ Chọn vào Signal led_out, sau đó click chuột phải và làm như hình:



a) TH1: Flick không được active ở trạng thái đầu, đợi 1 khoảng thời gian sau đó Flick = 1.

-> Kết quả cần đạt được: ban đầu khi chưa có flick đèn sẽ không hoạt động và giữ ở trạng thái Initial. Sau đó, khi flick = 1 thì sẽ bắt đầu hoạt động theo Normal Flow của đề.

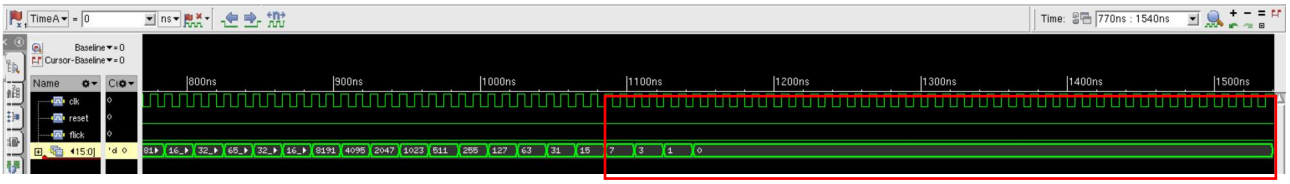


-> Ta quan sát thấy đạt yêu cầu.

b) TH2: Sau khi kết thúc state L15_L0_down, quay về trạng thái Initial State. Flick chỉ Active 1 lần ở Initial State đầu tiên.

-> Kết quả cần đạt được: khi tới Initial State lần hai, phải đợi Flick = 1, nếu flick cứ bằng 0 thì đèn không hoạt động(tắt hết các led).

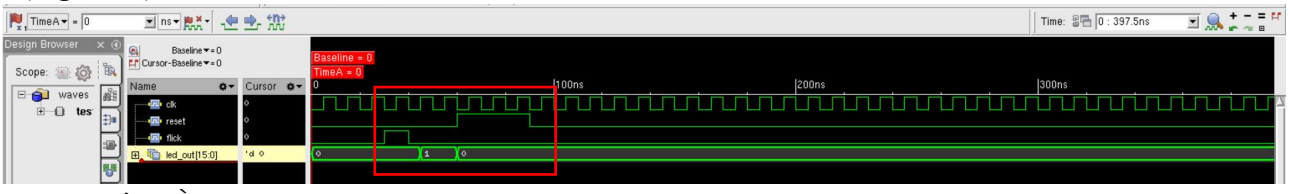
```
clk = 0;
 reset = 0;
 flick = 0;
 #30 flick = 1;
 #10 flick = 0;
 #1500 $finish;
```



c) TH4: Kiểm tra reset

```
clk = 0;
 reset = 0;
 flick = 0;
 #30 flick = 1;
 #10 flick = 0;
 #20 reset = 1;
 #30 reset = 0;
 #1500 $finish;
```

-> Kết quả cần đạt được: khi có reset, cần quay trở lại Initial State và đợi flick Active để hoạt động trở lại:



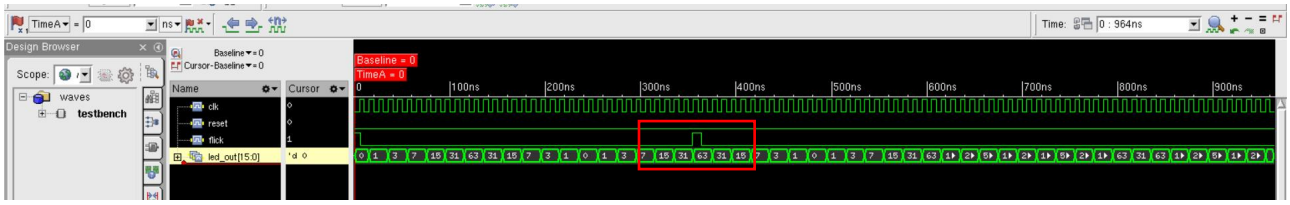
- > Đạt yêu cầu.

d) TH5: Flow with flick is ACTIVE in the kickback point L5 at L0_L10_UP

-> Kết quả cần đạt được: khi đèn L5 sáng (lúc này led_out đang có giá trị thập phân là 63 - 6'b111111)

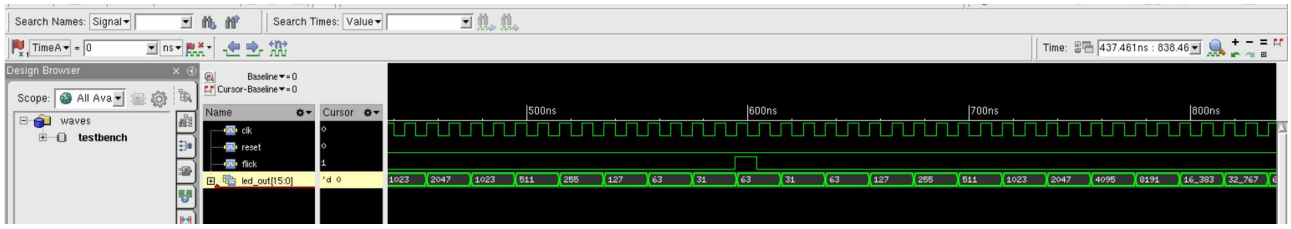
```
clk = 0;
 reset = 0;
 flick = 1;
```

```
#6 flick = 0;
#328 flick = 1;
#10 flick = 0;
#600 $finish;
```



e) TH6: Flow with flick is ACTIVE in the kickback point L5 at L5_L15_UP

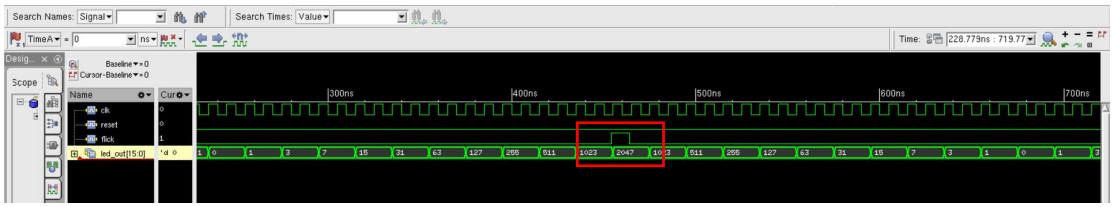
```
clk = 0;
 reset = 0;
 flick = 1;
 #6 flick = 0;
 #588 flick = 1;
 #10 flick = 0;
 #1000 $finish;
```



-> Ta thấy khi đèn L5 sáng lên(state L5_L15_up) và flick = 1 thì đèn L5 sẽ tắt(led_out = 31 - 5'b11111) sau đó tiếp tục trạng thái L5_L15_up (led_out = 63 - 6'b111111)

f) TH7: Flow with flick is ACTIVE in the kickback point L10 at L0_L10_UP.

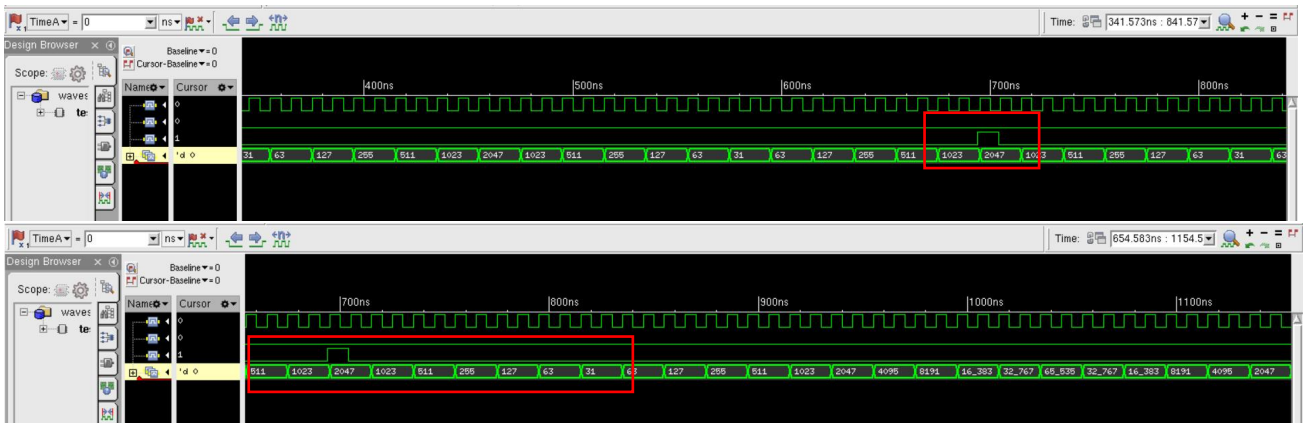
```
clk = 0;
 reset = 0;
 flick = 1;
 #6 flick = 0;
 #448 flick = 1;
 #10 flick = 0;
 #1500 $finish;
```



-> Ta thấy khi ở state L0_L10_up, khi đèn L10 sáng và flick = 1 (led_out = 2047 - 11'b11111111111) thì sau đó sẽ tắt dần về 0 (min của trạng thái trước)

g) TH8: Flow with flick is ACTIVE in the kickback point L10 at L5_L15_UP

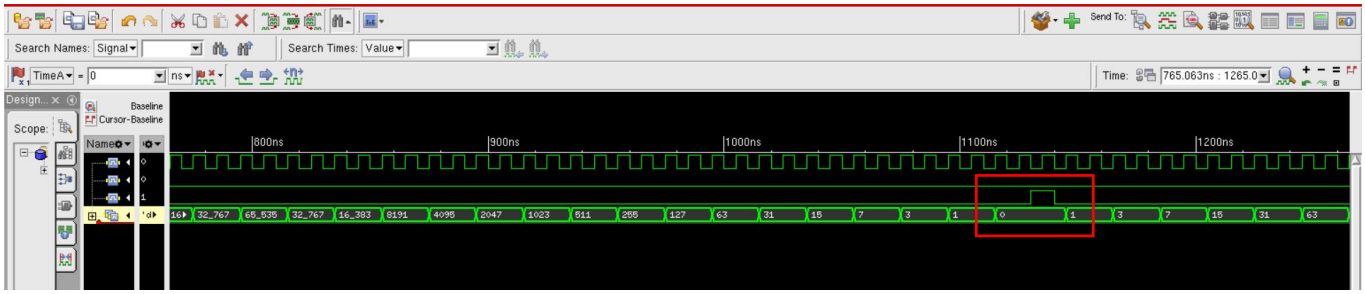
```
clk = 0;
reset = 0;
flick = 1;
#6 flick = 0;
#688 flick = 1;
#10 flick = 0;
#1500 $finish;
```



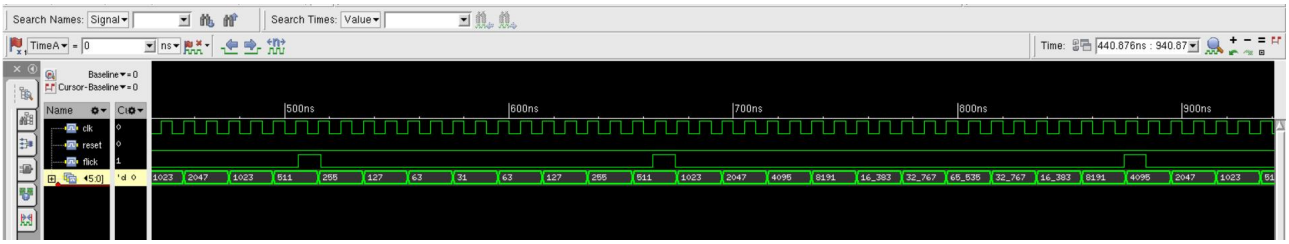
-> Ta quan sát thấy ngay tại L10 của trạng thái L5_L15_up khi flick = 1 thì sẽ bắt đầu tắt dần về min của trạng thái trước tức là tắt dần về L5(led_out = 31 - 5'b11111). Sau đó tiếp tục quay lại trạng thái L5_L15_up

h) TH9: L15_L0_down kết thúc, quay lại Initial State, sau đó đợi Flick = 1 để tiếp tục vận hành;

```
clk = 0;
reset = 0;
flick = 1;
#6 flick = 0;
#1124 flick = 1;
#10 flick = 0;
#1500 $finish;
```



i) TH10: Ta chọn random 3 điểm để flick = 1 mà không phải kickback point



- Ta thấy nó không ảnh hưởng đến Normal Flow -> đạt yêu cầu

IV) Schematic Tracer: