數位系統技術

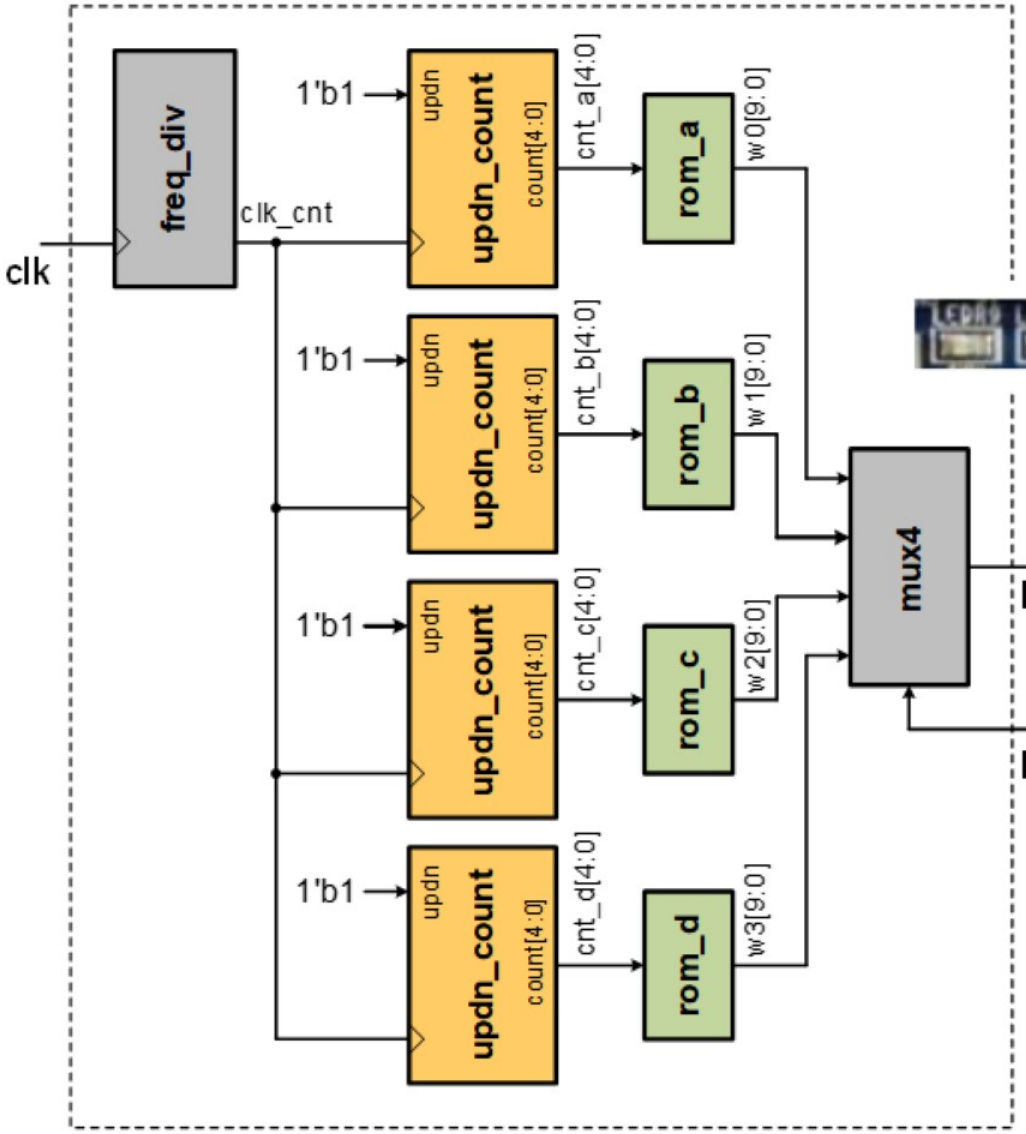
LED Animation

Ren-Der Chen (陳仁德)
Department of Computer Science and
Information Engineering
National Changhua University of Education
E-mail: rdchen@cc.ncue.edu.tw
Spring, 2025

實驗內容

- 設計一個可於10個led上，呈現四種不同動態燈號變化之電路。
- 設計具上數/下數功能之計數器(up/down counter)，及除頻器(frequency divider)電路。
- 將所設計之電路燒錄至FPGA晶片中，並利用實驗板週邊進行電路功能驗證。

led_top 電路方塊圖



led_b[9:0] 決定10個led之亮暗情形

ptn_sel[1:0] 選擇四種led動畫之其中一種

Design 1: updn_count (1/3)

- Create New Project - led_top
- 設計一個具有上數/下數功能之計數器電路updn_count.v，並加入至project中。
- 計數器之輸出信號為5 bits，計數值最小為0，最大為31。
- 採用parameter方式設計，可自行設定計數個數X，計數範圍為 0 ~ X-1。

Design 1: updn_count (2/3)

```
module updn_count(clk, rst, updn, count);
    parameter CNT_LENGTH = 8;
    // default count length = 8, from 0 to 7

    input   clk, rst, updn;
    output              count;
    // max count length = 32, from 0 to 31
            // Default # of bits = 5
    always @(posedge clk or posedge rst) begin
        if (rst)
            count = 5'd0;
        else begin
            if (updn == 1'b1) begin // count up
                if (count ==                    )
                    count = 5'd0;
                else
                    count =
            end // if
```

// updn=1, count up: 0, 1, ..., 7, 0, ...

// updn=0, count down: 0, 7, ..., 1, 0, ...

// 0, 1, 2, ..., 6, 7, 0, 1, ...

// 0, 1, 2, ..., 6, 7, 0, 1, ...

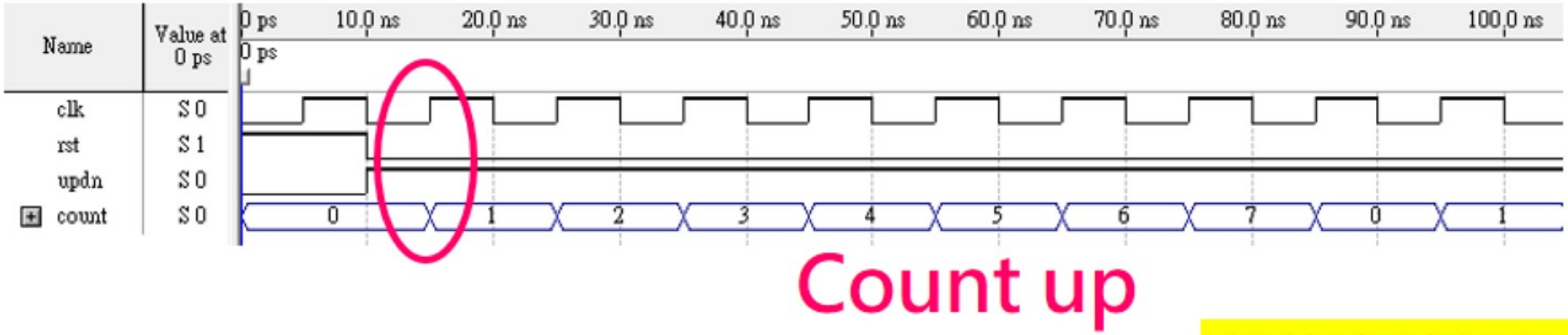
Design 1: updn_count (3/3)

```
            else begin // count down
                if (count == 5'd0)
                    count = [          ];
                else
                    count = [      ]
                end // else
            end // else
        end // always
    endmodule
```

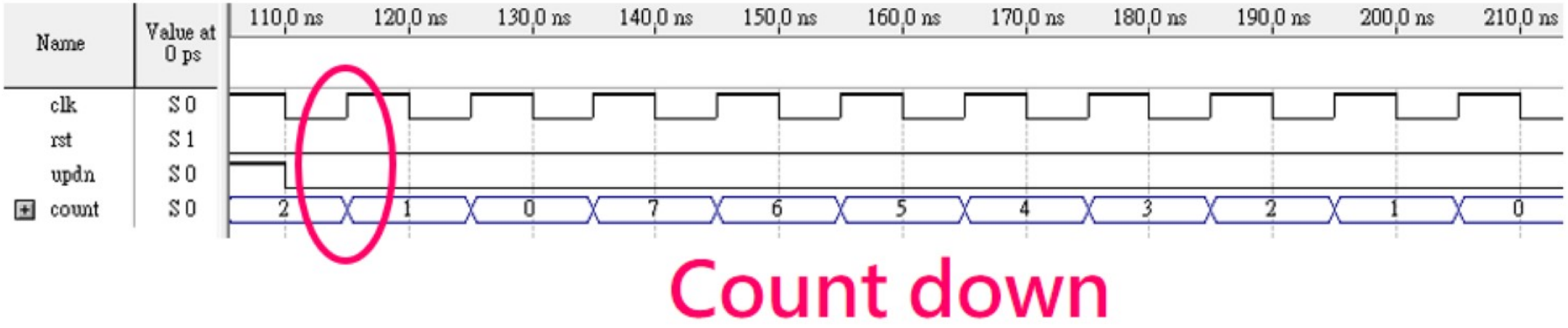
// 0, 7, 6, ..., 1, 0, 7, 6, ...

// 0, 7, 6, ..., 1, 0, 7, 6, ...

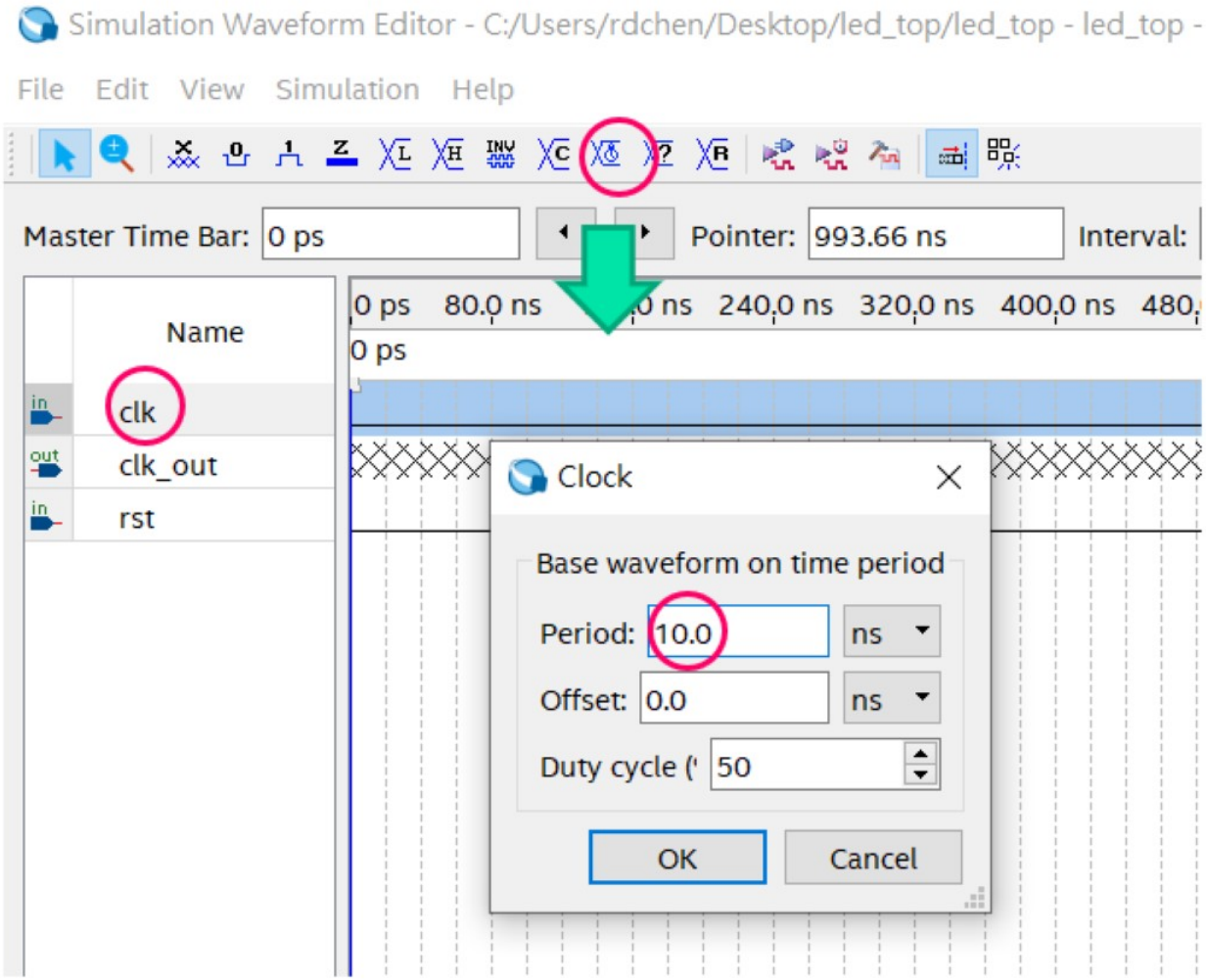
Functional Simulation for “updn_count”



記錄實驗結果

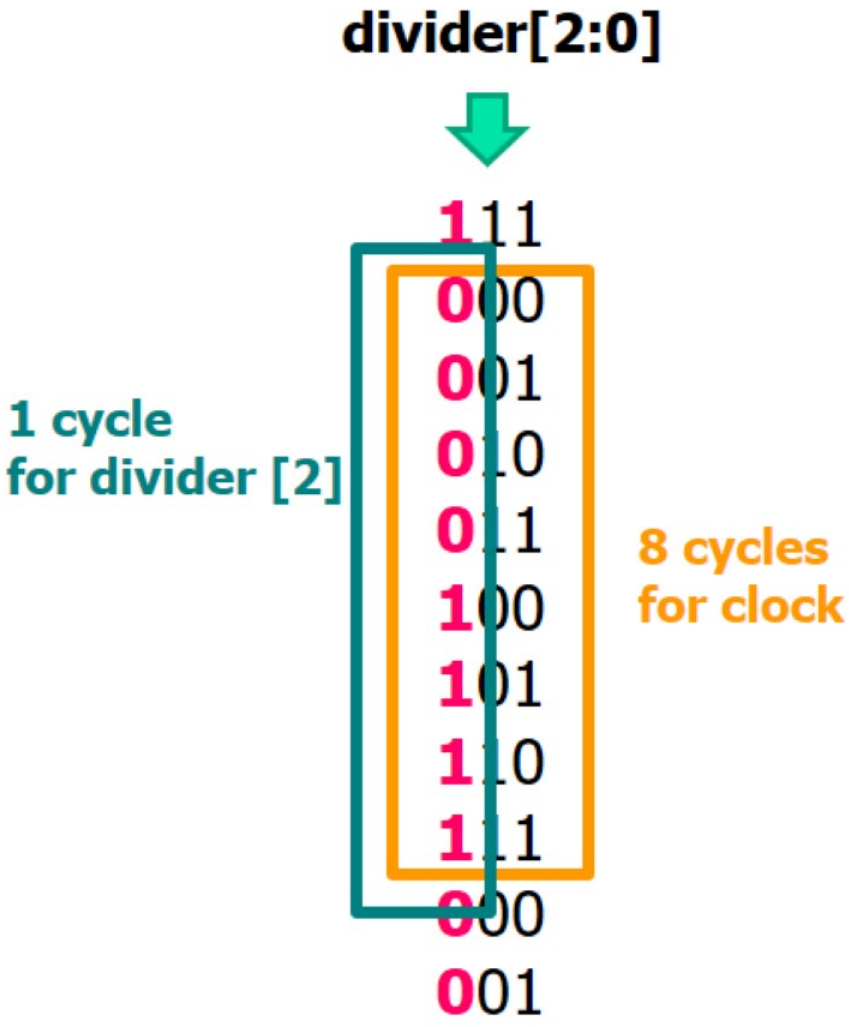


Set Clock Period



Design 2: freq_div (1/2)

- 設計一個除頻器電路 freq_div.v，並加入至 project中。
- 實驗板提供之clock頻率為 50MHz，利用除頻器電路可將該頻率除以2^n，適合於檢視週邊信號變化。



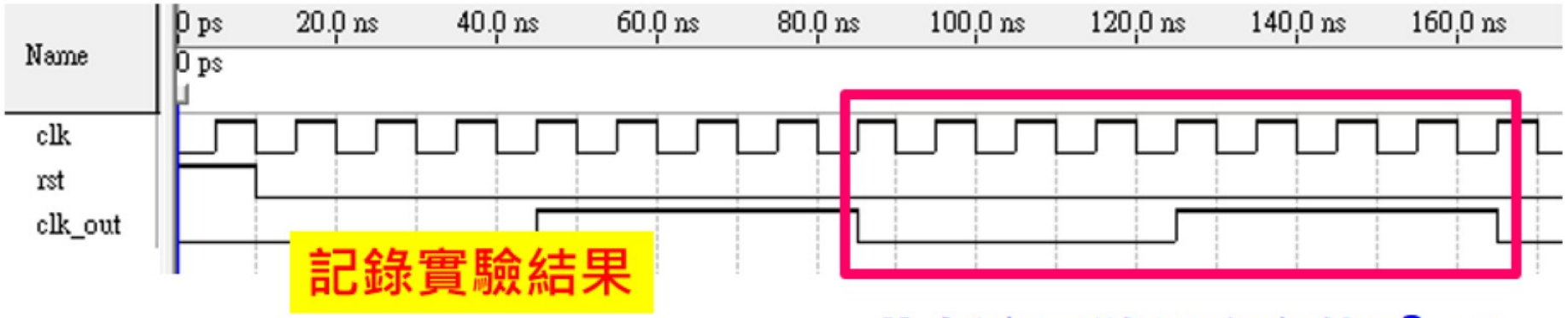
Design 2: freq_div (2/2)

```
module freq_div(clk, rst, clk_out);
    parameter   EXP = 3;
    // divided by default power(2, 3)

    input   clk, rst;
    output  clk_out;

    reg             divider;

    assign clk_out =

    always @(posedge clk or posedge rst) begin
        if (rst)
            divider =
        else
            divider =
    end // always
endmodule
```

clk為輸入時脈, 計數值為divider[2:0], 若將clk_out=divider[2]輸出, 其頻率就變成clk的1/8

每一次的clk觸發, 計數值divider[2:0]即遞增1

Functional Simulation for “freq_div”



clk頻率已除以內定值$2^3=8$

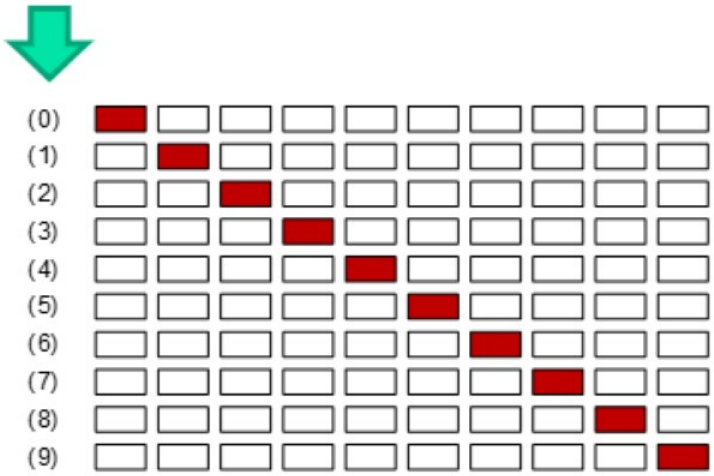
LED Patterns

- 設計四種led patterns，分別存放於四個檔案(rom_a.v, rom_b.v, rom_c.v, and rom_d.v) ，並加入至project中。

Design 3: rom_a

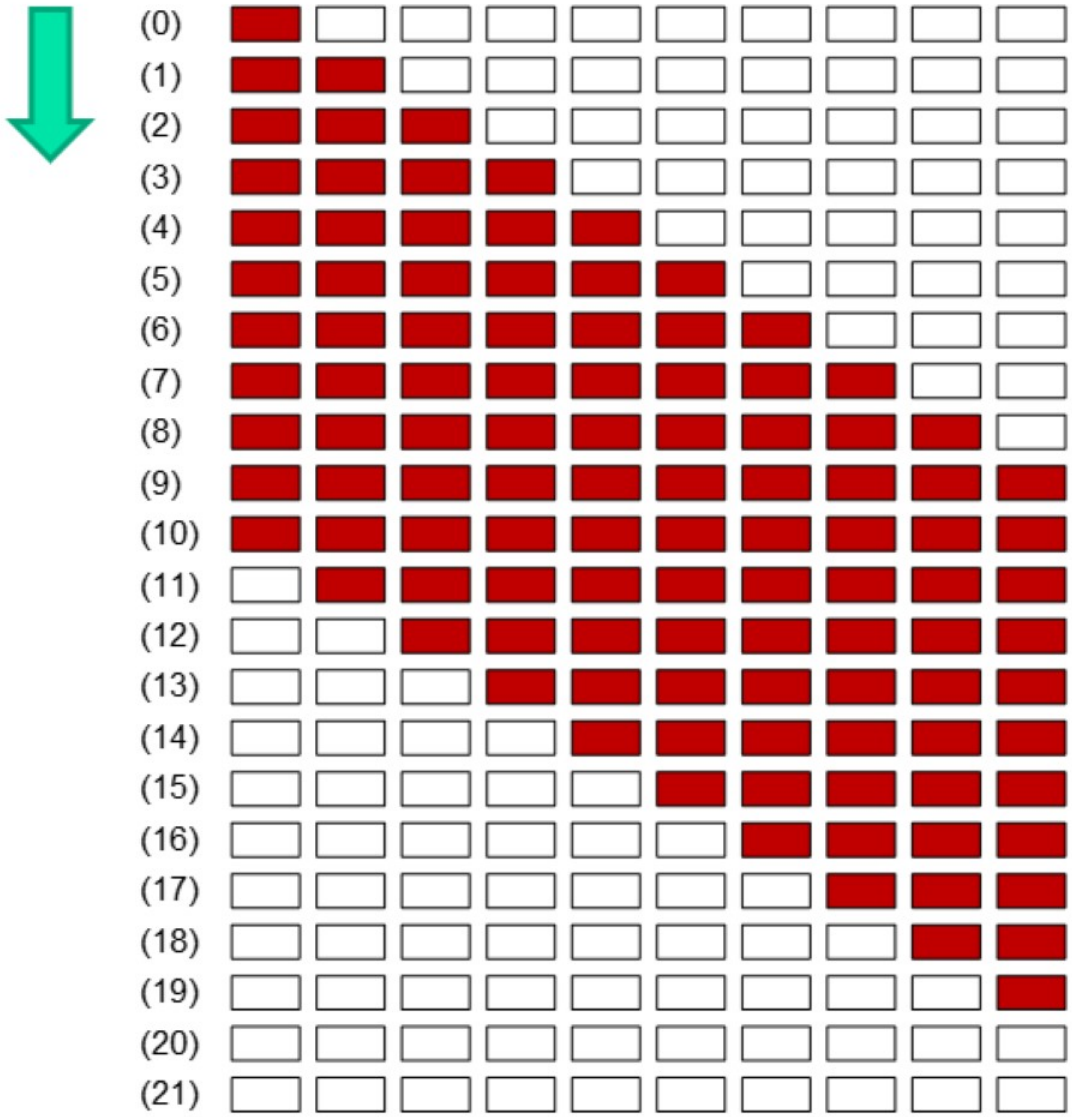
```
module rom_a(addr, data);
    input   [4:0] addr;
    output  reg [9:0] data;

    always @(addr) begin
        case(addr)
            5'd0:   data = 10'b1000000000;
            5'd1:   data = 10'b0100000000;
            5'd2:   data = 10'b0010000000;
            5'd3:   data = 10'b0001000000;
            5'd4:   data = 10'b0000100000;
            5'd5:   data = 10'b0000010000;
            5'd6:   data = 10'b0000001000;
            5'd7:   data = 10'b0000000100;
            5'd8:   data = 10'b0000000010;
            5'd9:   data = 10'b0000000001;

            default: data = 10'b0000000000;
        endcase
    end // always
endmodule
```

data[9:0]對應到10個 led,
1: on, 0: off



共10種圖案循環

Design 4: rom_b (1/2)



共22種圖案循環

Design 4: rom_b (2/2)

```
module rom_b(addr, data);
    input   [4:0] addr;
    output  reg [9:0] data;

    always @(addr) begin
        case(addr)
            default: data = 10'b0000000000;
        endcase
    end // always
endmodule
```

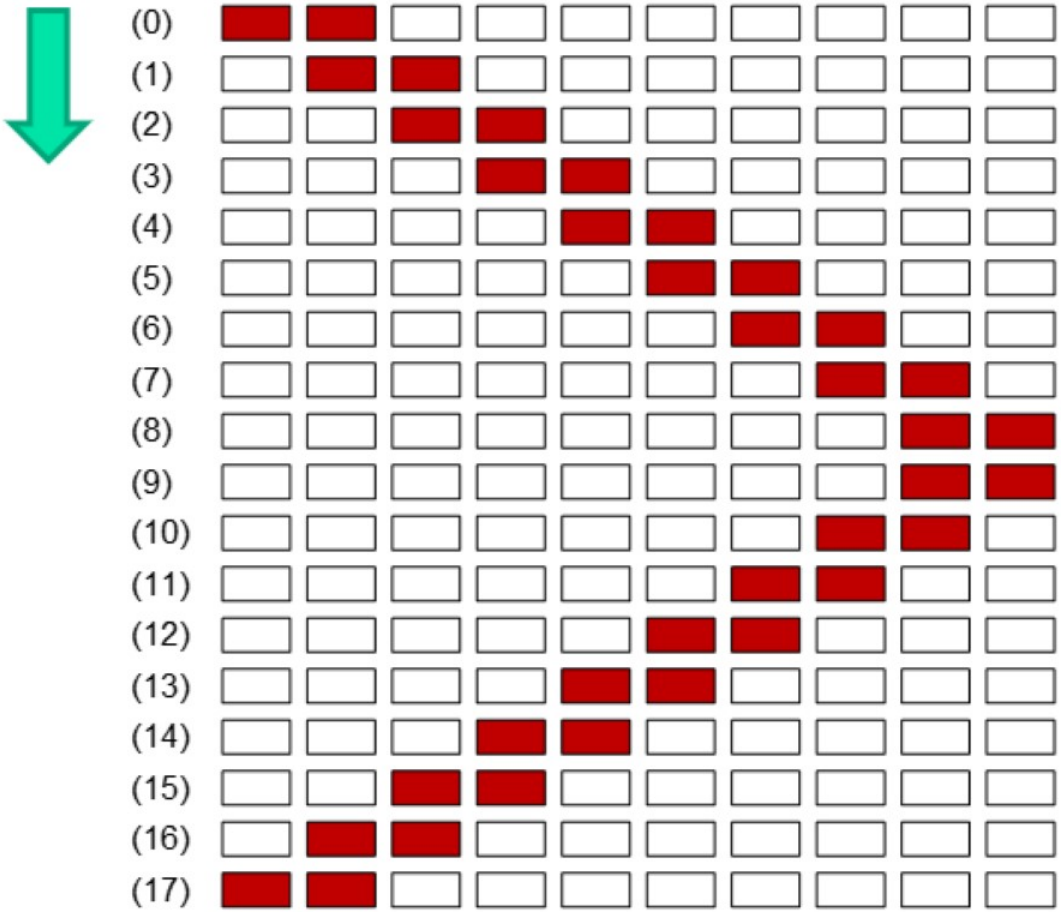
Design 5: rom_c (1/2)

共18種圖案循環

Design 5: rom_c (2/2)

```
module rom_c(addr, data);
    input   [4:0] addr;
    output  reg [9:0] data;

    always @(addr) begin
        case(addr)
            default: data = 10'b0000000000;
        endcase
    end // always
endmodule
```

Design 6: rom_d (1/2)



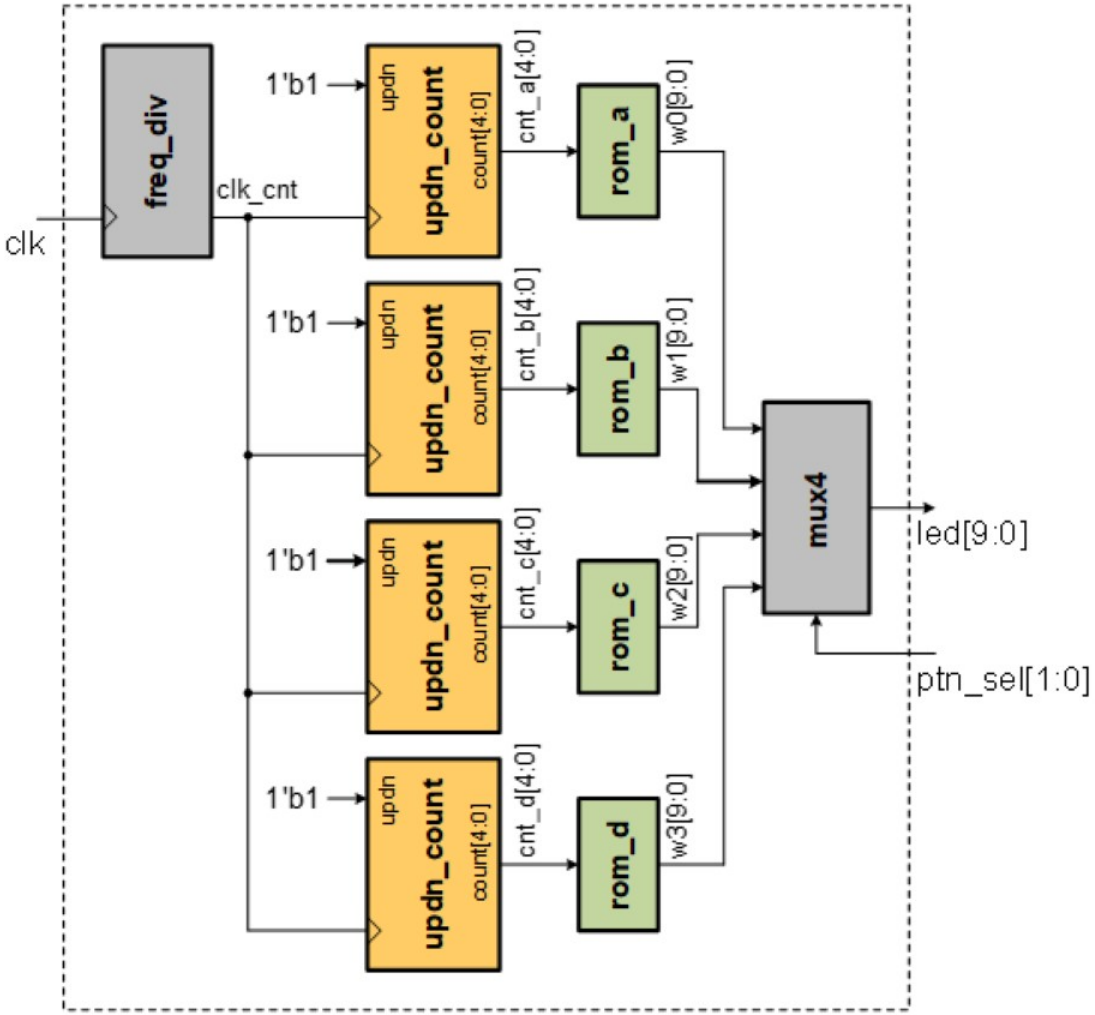
共18種圖案循環

Design 6: rom_d (2/2)

```
module rom_d(addr, data);
    input   [4:0] addr;
    output  reg [9:0] data;

    always @(addr) begin
        case(addr)
            default: data = 10'b0000000000;
        endcase
    end // always
endmodule
```

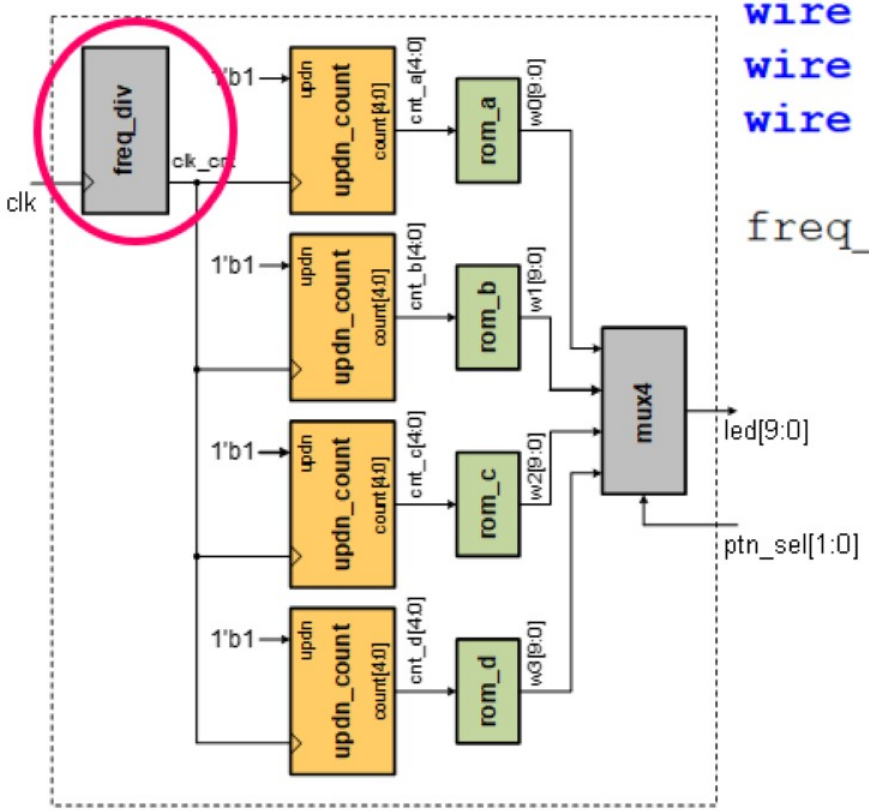
Design 7: led_top (1/4)

- 設計led_top.v並加入至project中，整合所有電路模組。



Design 7: led_top (2/4)

```
module led_top(clk, rst, ptn_sel, led);
    input   clk, rst;
    input   [1:0]  ptn_sel;
    output  [9:0] led;

    wire    clk_cnt;
    wire    [ ]  cnt_a, cnt_b, cnt_c, cnt_d;
    wire    w0, w1, w2, w3;

    freq_div [ ] f1(clk, [ ] clk_cnt);
```



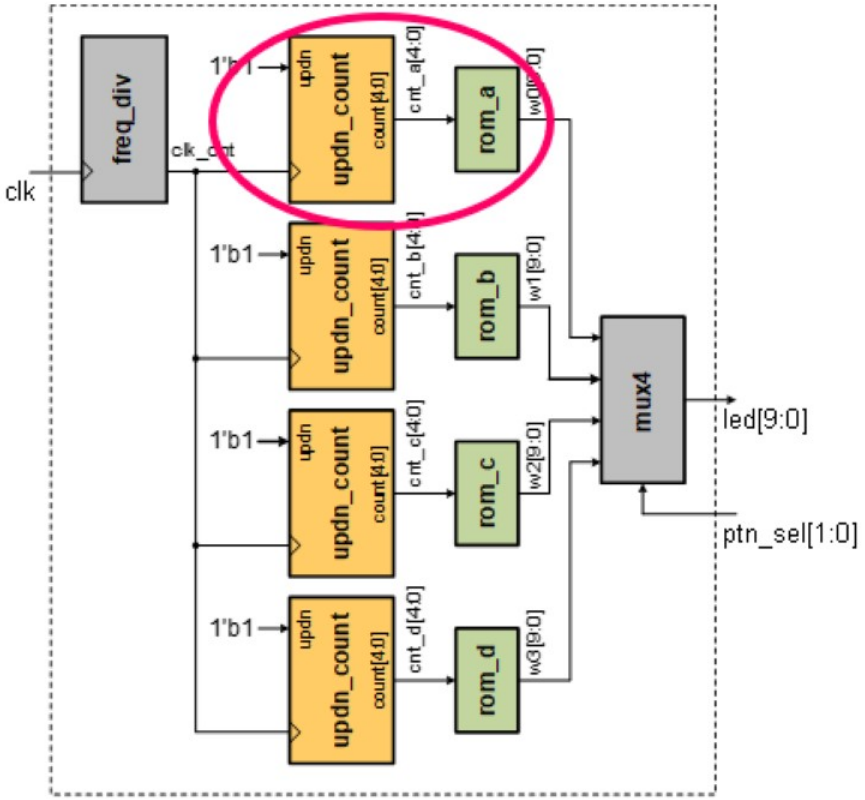
1. freq_div之傳入參數#(24), 為配合動畫執行速度所設定之值，除頻大小為power(2, 24)。
2. 實驗板上接rst信號之按鍵為0觸發, 按下代表0, 因此接至該按鍵之rst必須先反相再進入電路。

Design 7: led_top (3/4)

// led pattern之圖案數量　// updn固定為1, 因為只需要上數

```
updn_count #(10) cnt0(clk_cnt, ~rst, 1'b1, cnt_a);
updn_count
updn_count
updn_count
```

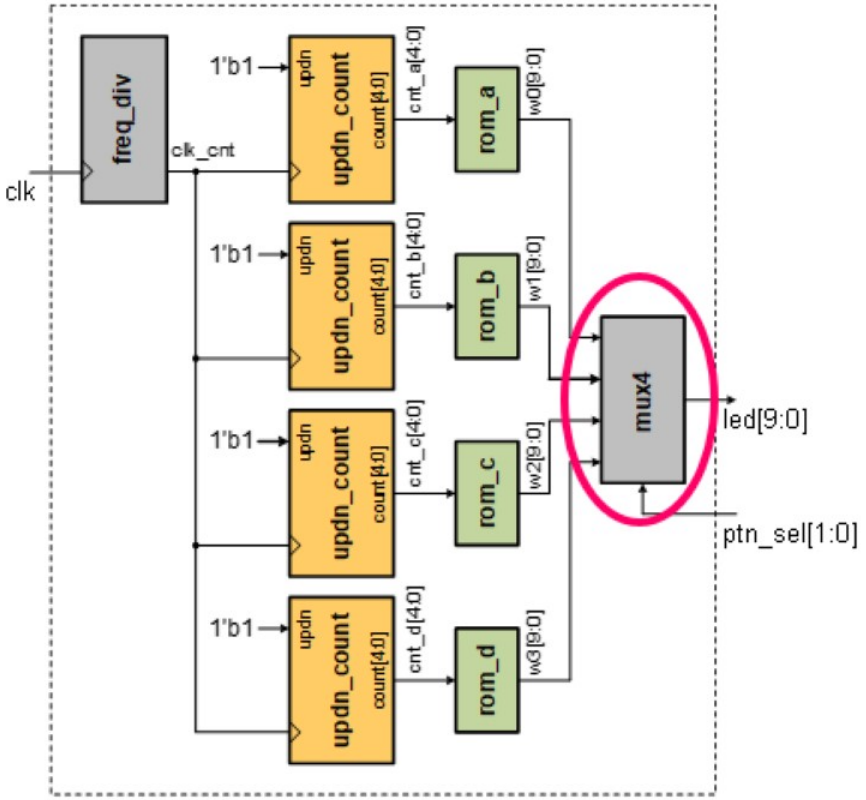
```
rom_a r0(cnt_a, w0);
rom_b r1
rom_c r2
rom_d r3
```



Design 7: led_top (4/4)

```
// mux 4x1
assign led =
```

ptn_sel=00, 選擇rom_a
ptn_sel=01, 選擇rom_b
ptn_sel=10, 選擇rom_c
ptn_sel=11, 選擇rom_d



Pin Assignments

ptn_sel[1:0]

led[9:0]

clk

Clocks: 50 MHZ

CLK1: P11

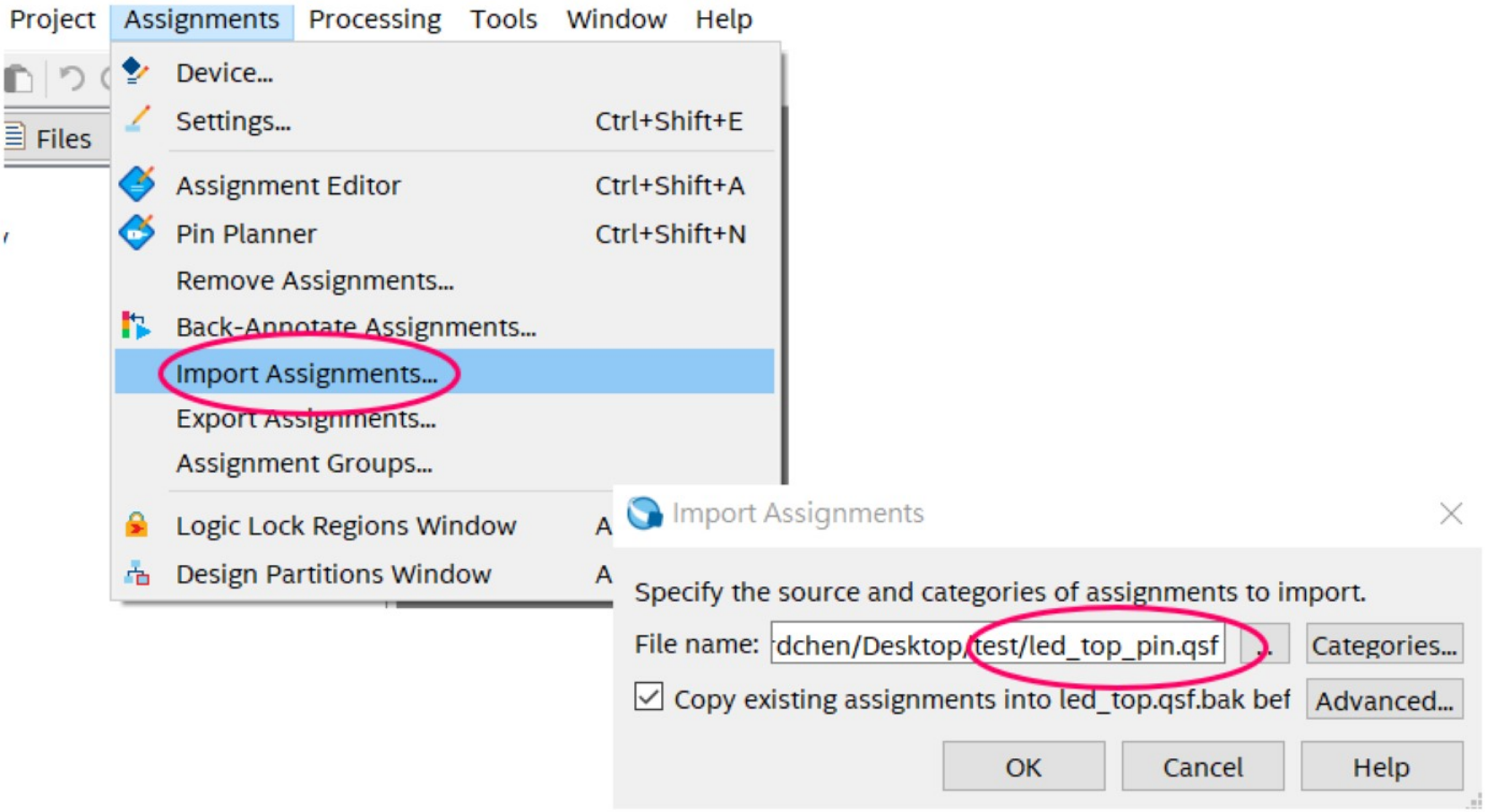
CLK2: N14

rst

KEY0: B8 Push: 0

KEY1: A7 Not push: 1

Import Pin Assignments



led_top_pin.qsf

```
set_location_assignment PIN_P11 -to clk
set_location_assignment PIN_B8 -to rst
set_location_assignment PIN_C11 -to ptn_sel[1]
set_location_assignment PIN_C10 -to ptn_sel[0]
set_location_assignment PIN_B11 -to led[9]
set_location_assignment PIN_A11 -to led[8]
set_location_assignment PIN_D14 -to led[7]
set_location_assignment PIN_E14 -to led[6]
set_location_assignment PIN_C13 -to led[5]
set_location_assignment PIN_D13 -to led[4]
set_location_assignment PIN_B10 -to led[3]
set_location_assignment PIN_A10 -to led[2]
set_location_assignment PIN_A9 -to led[1]
set_location_assignment PIN_A8 -to led[0]
```

File List

- Verilog files
  - updn_count.v
  - freq_div.v
  - rom_a.v
  - rom_b.v
  - rom_c.v
  - rom_d.v
  - led_top.v
- Waveform files
  - updn_count.vwf
  - freq_div.vwf

實驗結果驗收

- 請老師或助教驗收
  - updn_count.v 之波形
  - freq_div.v 之波形
  - led_top 電路於實驗板之四種led動畫行為