數位系統技術

7-Segment Snake

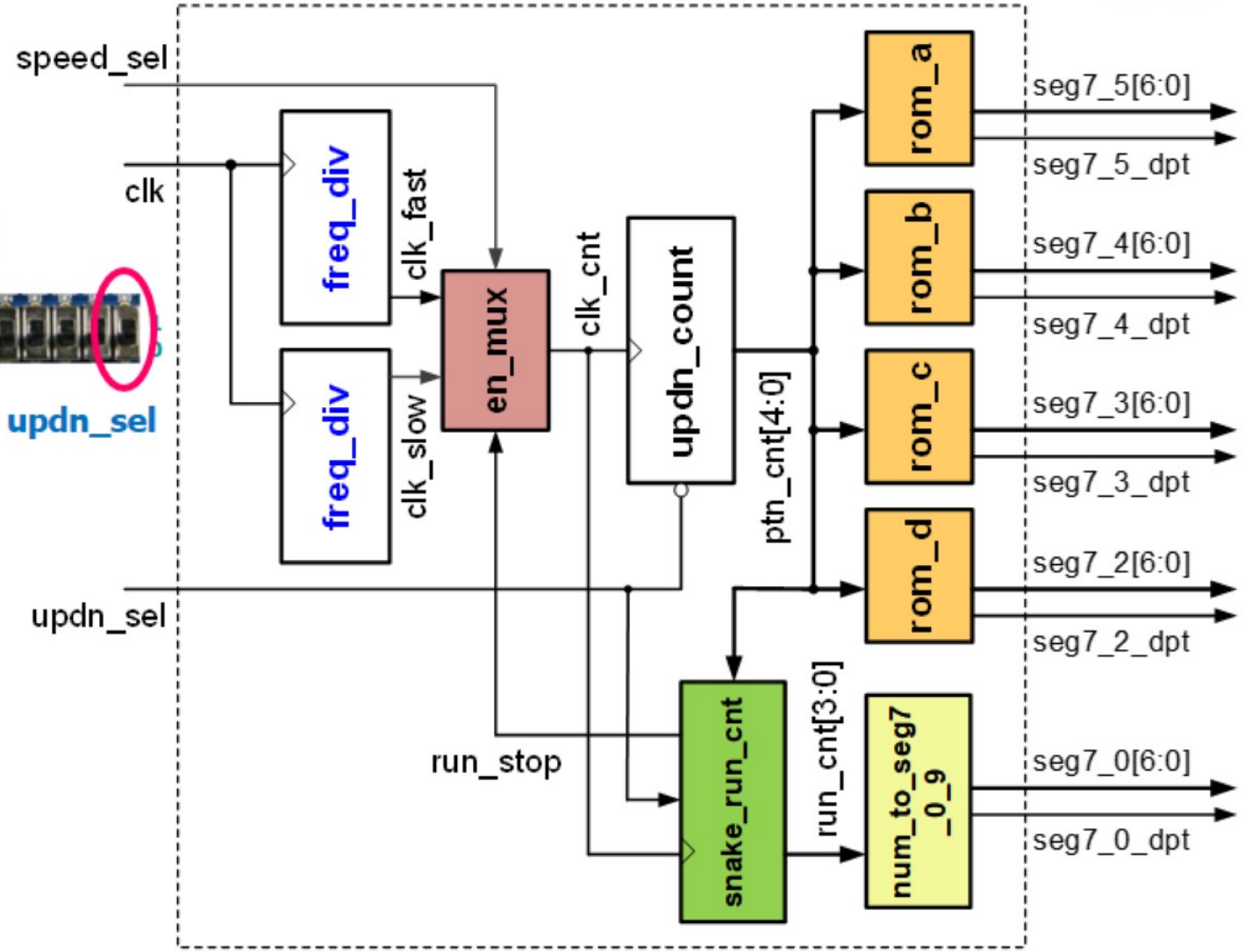
Ren-Der Chen (陳仁德)
Department of Computer Science and
Information Engineering
National Changhua University of Education
E-mail: rdchen@cc.ncue.edu.tw
Spring, 2025

實驗內容

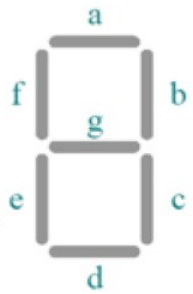
- 利用4個七段顯示器設計類似貪食蛇動畫。
- 外部之2個輸入可分別控制動畫之快慢及正反轉。
- 動畫循環之次數可顯示於另1個七段顯示器之上，而當次數達9次時，電路停止動作。
- 將所設計之電路燒錄至FPGA晶片中，並利用實驗板週邊進行電路功能驗證。

- Create New Project - snake_top

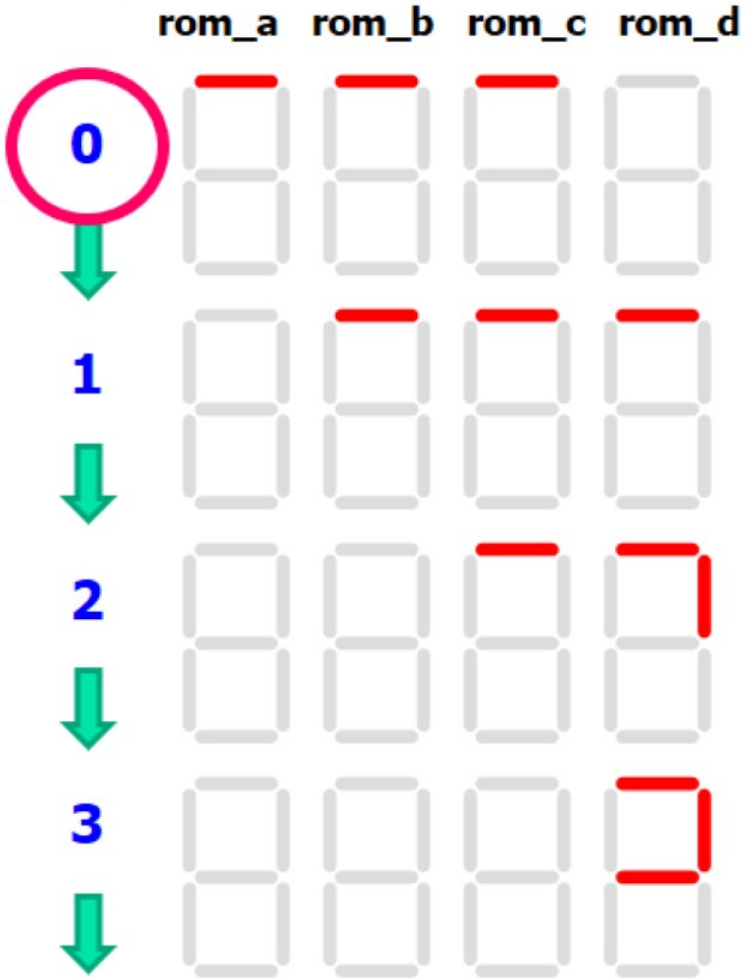
snake_top 電路方塊圖

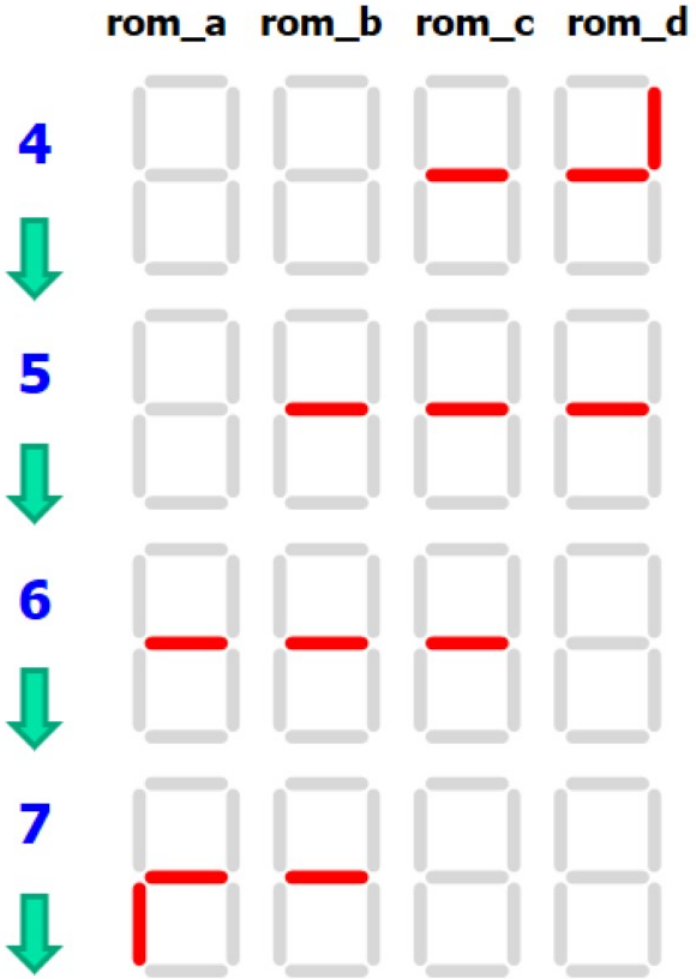


Snake設計原理 (1/3)

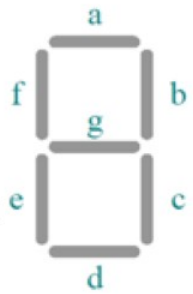


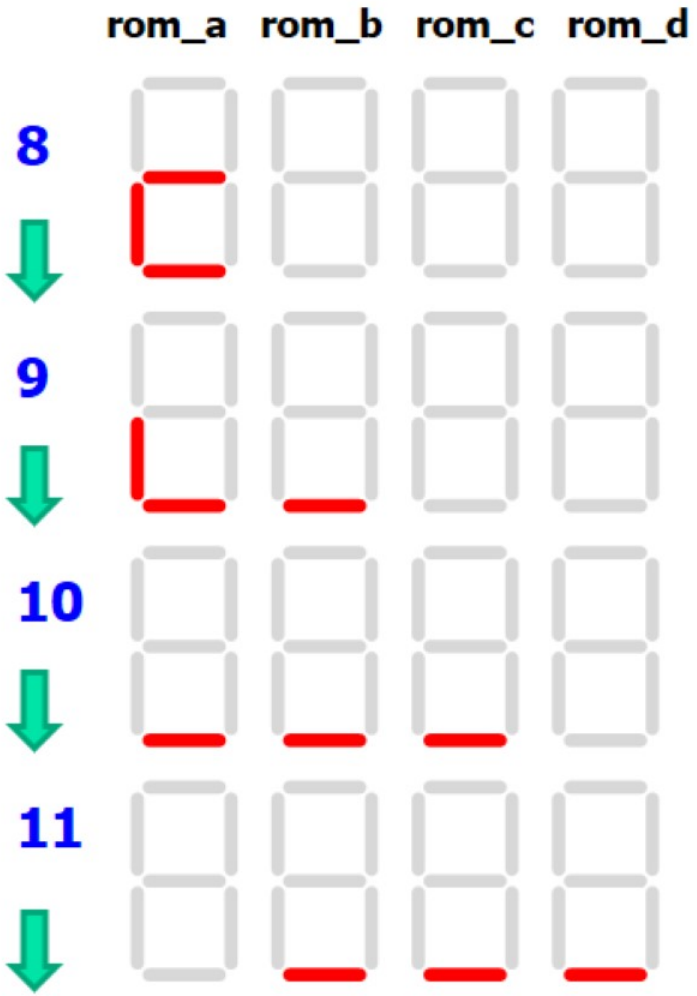
循環開始

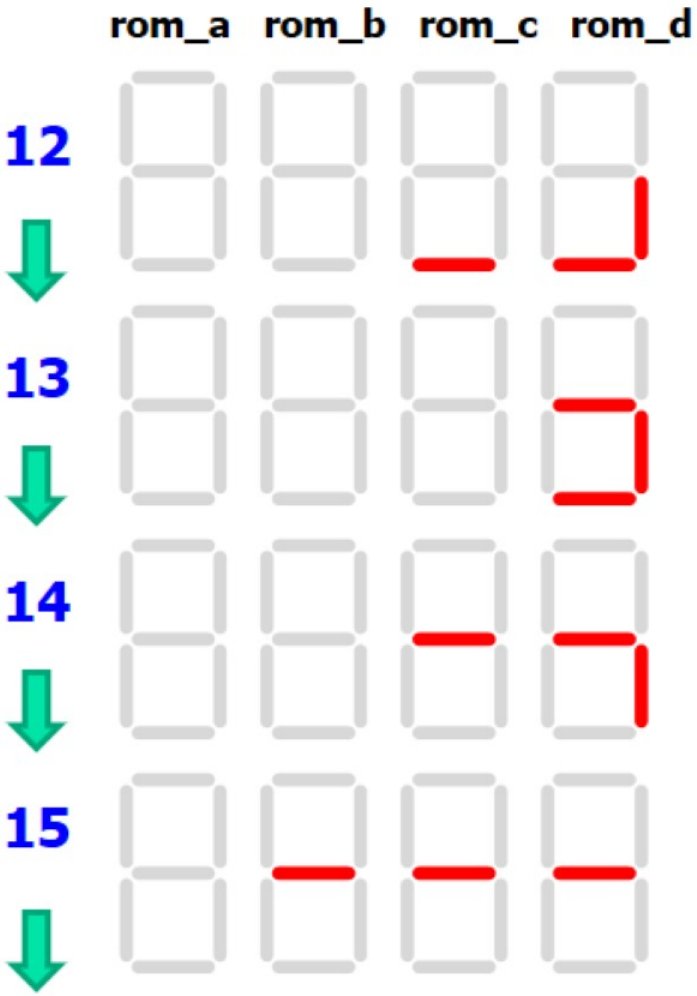




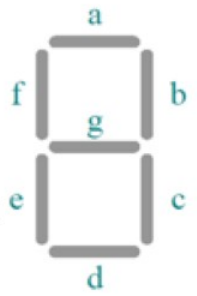
Snake設計原理 (2/3)

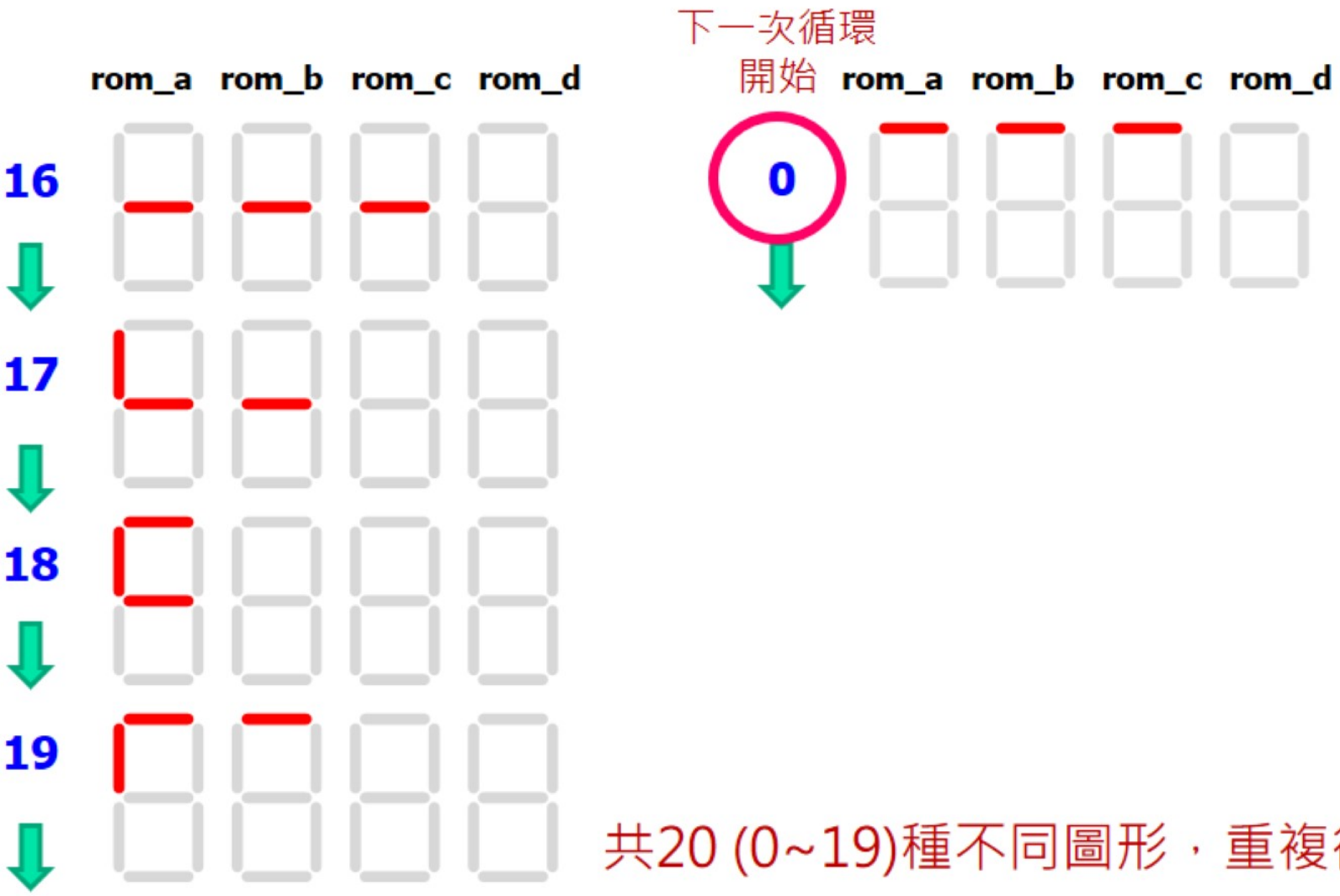






Snake設計原理 (3/3)

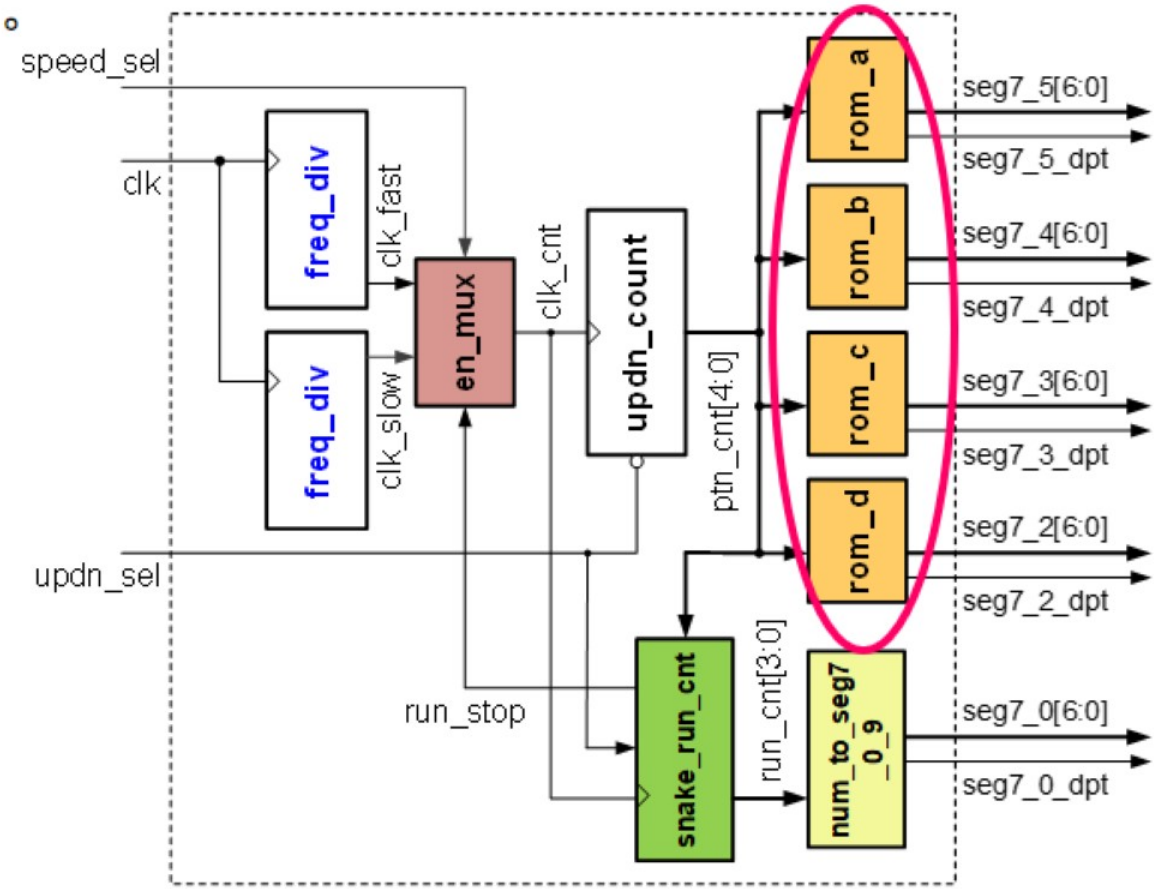




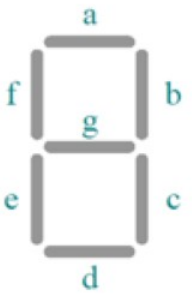
共20 (0~19)種不同圖形，重複循環

Seg7 Patterns

- 針對4個seg7，分別設計其動畫之變化圖案，存放於4個檔案(rom_a.v, rom_b.v, rom_c.v, and rom_d.v)，並加入至project中。

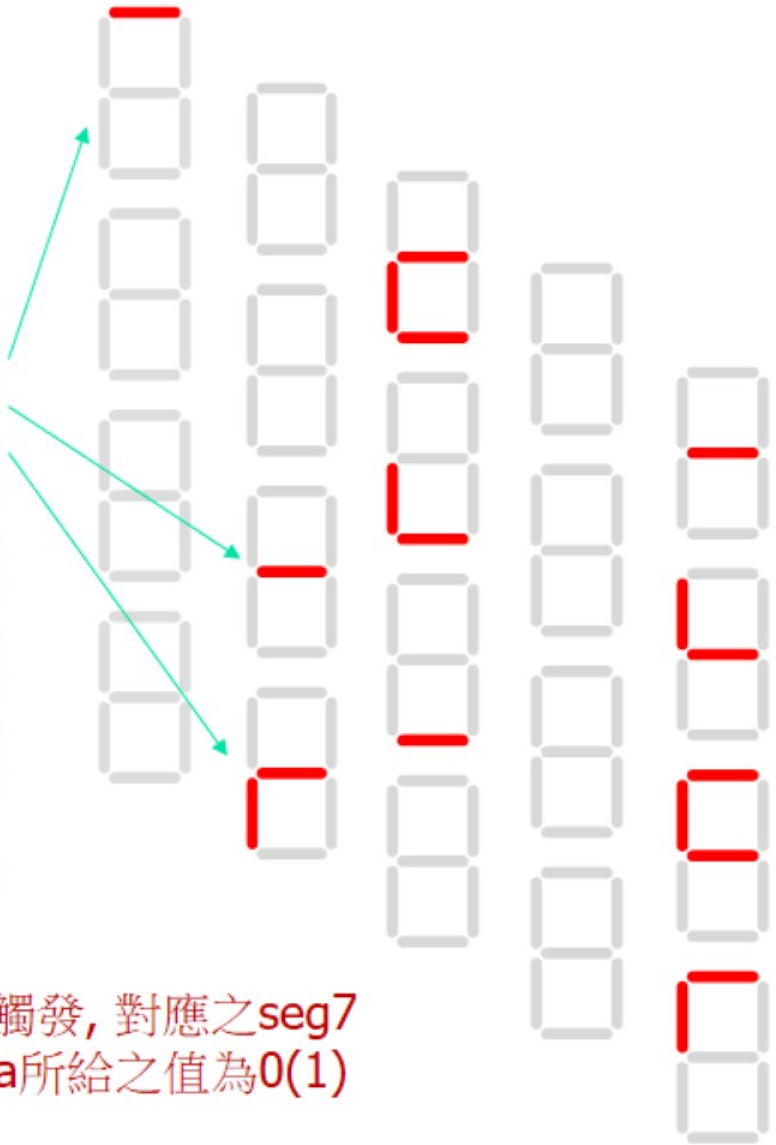


Design 1_1: rom_a



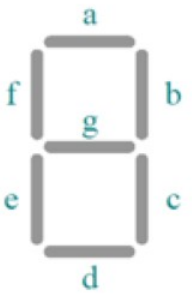

```
module rom_a(addr, data, dpt);
    input        [4:0] addr;
    output  reg [6:0] data;
    output  dpt;

    always @(addr) begin
        case(addr)          // gfedcba
            5'd0:  data = 7'b1111110;
            5'd6:  data = 7'b0111111;
            5'd7:  data = 7'b0101111;
            5'd8:  data = 7'b0100111;
            5'd9:  data = 7'b1100111;
            5'd10: data = 7'b1110111;
            5'd16: data = 7'b0111111;
            5'd17: data = 7'b0011111;
            5'd18: data = 7'b0011110;
            5'd19: data = 7'b1011110;

            default: data=7'b1111111;
        endcase
    end // always

    assign dpt=1'b1;
endmodule
```

// 共10種
狀況有亮燈

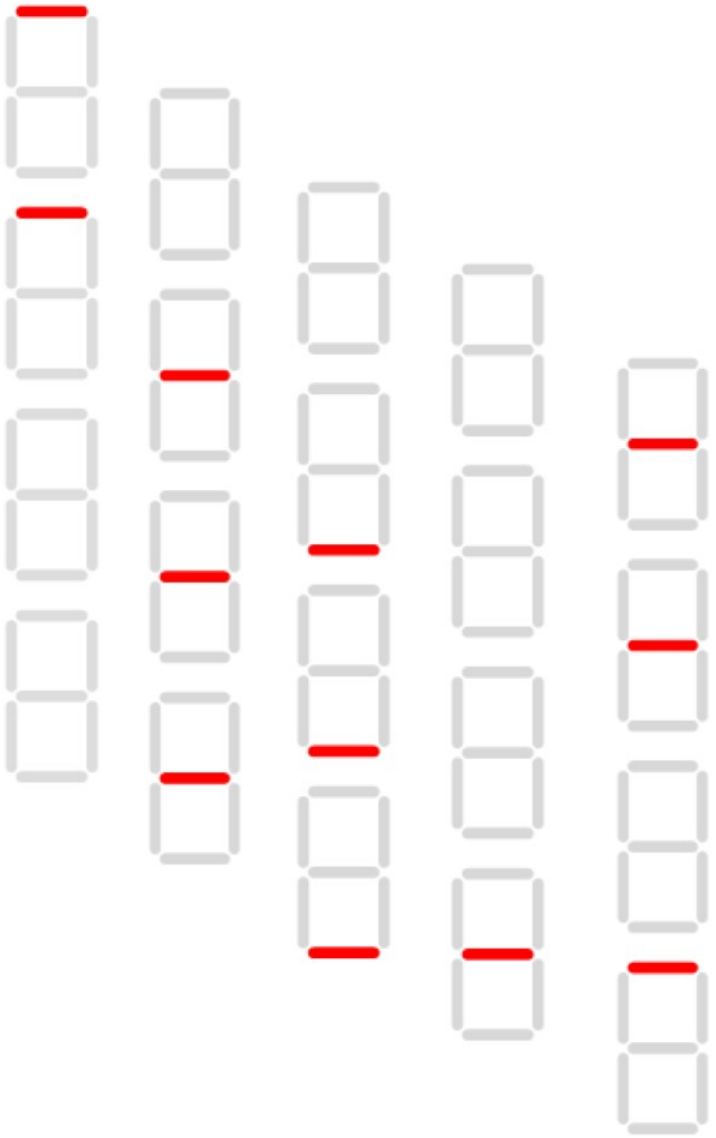


// 由於seg7為0觸發, 對應之seg7
若要亮(暗), data所給之值為0(1)

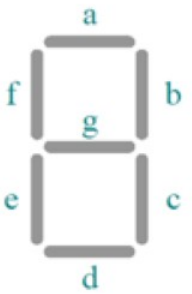
Design 1_2: rom_b



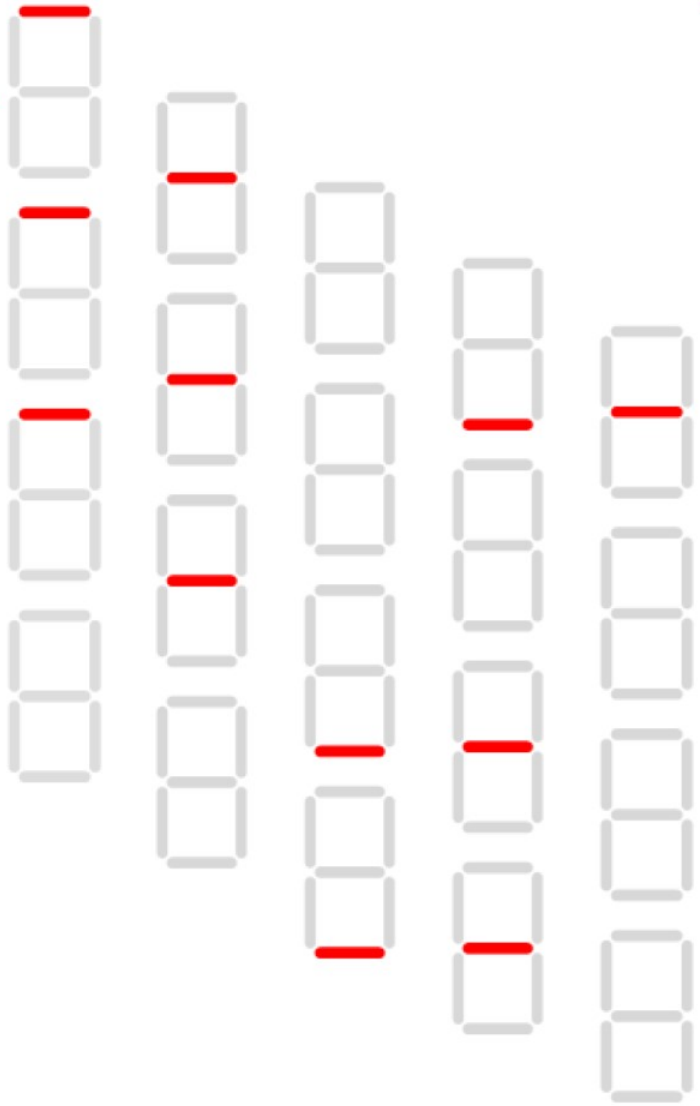

```
module rom_b(addr, data, dpt);
    input        [4:0] addr;
    output  reg [6:0] data;
    output  dpt;

    always @(addr) begin
        case(addr)          // gfedcba
            5'd0:   data = 7'b1111110;
            default: data = 7'b1111111;
        endcase
    end // always

    assign dpt=1'b1;
endmodule
```



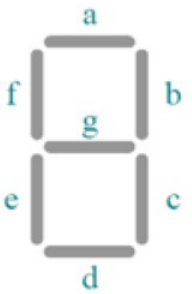
Design 1_3: rom_c



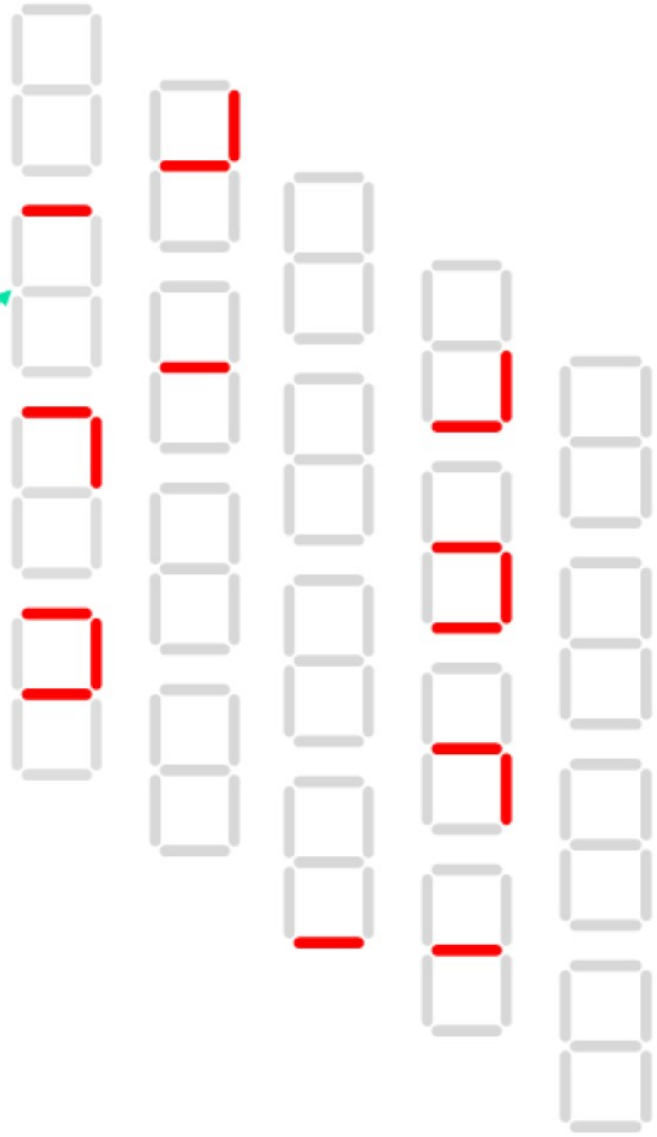

```
module rom_c(addr, data, dpt);
    input       [4:0] addr;
    output  reg [6:0] data;
    output  dpt;

    always @(addr) begin
        case(addr)          // gfedcba
            5'd0:  data = 7'b1111110;

            default: data = 7'b1111111;
        endcase
    end // always

    assign dpt=1'b1;
endmodule
```



Design 1_4: rom_d

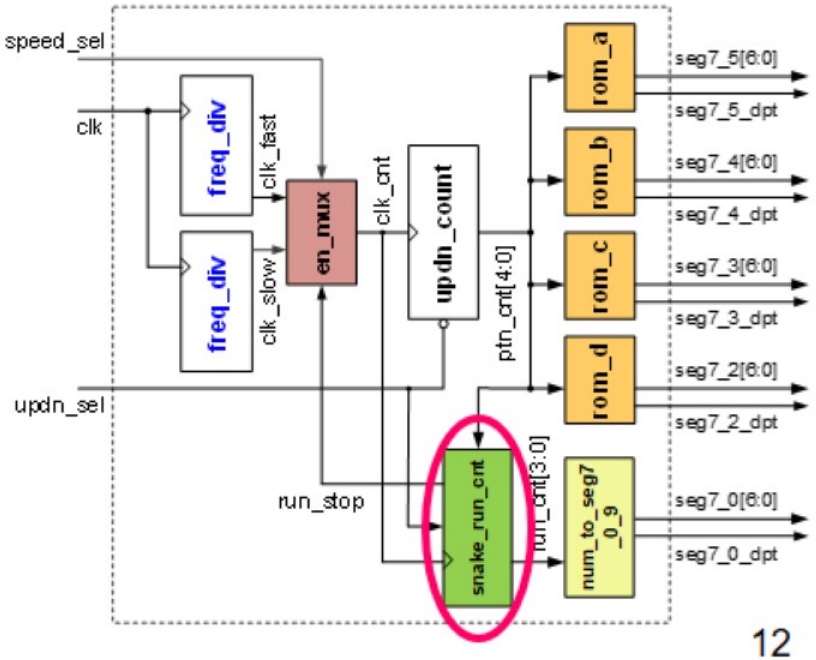



```
module rom_d(addr, data, dpt);
    input   [4:0] addr;
    output  reg [6:0] data;
    output  dpt;

    always @(addr) begin
        case(addr)          // gfedcba
            5'd1:  data = 7'b1111110;

            default: data = 7'b1111111;
        endcase
    end // always

    assign dpt=1'b1;
endmodule
```



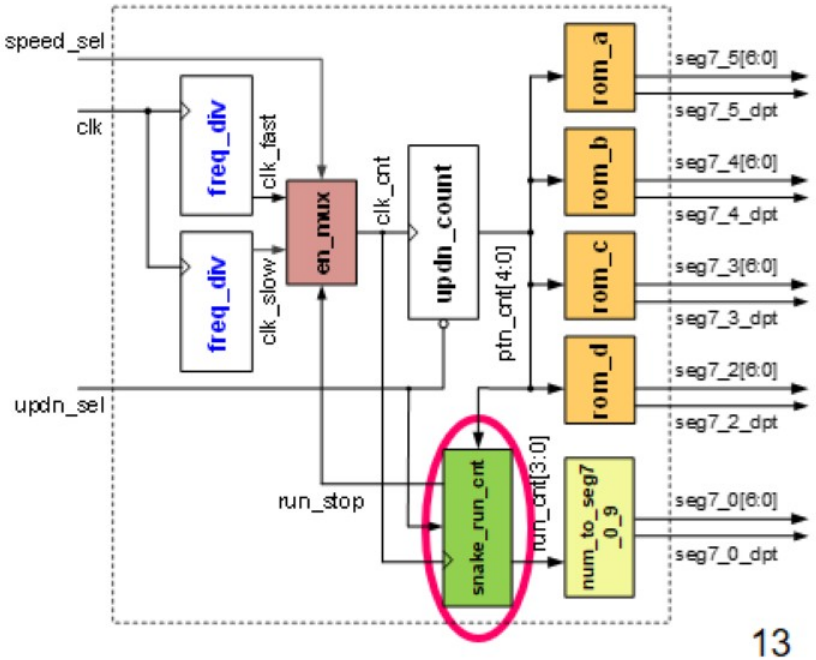
Design 2: snake_run_cnt (1/4)

- 設計snake_run_cnt.v，紀錄目前是第幾次動畫循環於信號run_cnt[3:0]，及目前是否停止電路執行於信號run_stop中。
- 依據目前是正轉(updn=1)還是反轉(updn=0)，及目前產生至第幾個seg7變化圖案(ptn_cnt[4:0])，即可決定目前是第幾次之動畫循環(run_cnt[3:0])。



Design 2: snake_run_cnt (2/4)

- 由於seg7只能顯示0~9之數字，當第9次循環之動畫結束時，電路即停止動作(run_stop=1)。
- 動畫圖案計數值ptn_cnt[4:0]由0開始數，若是正轉(updn=1)，則當數到(0, 1, 2, …, 18, **19**), 0, …之19時表示正轉已經過一個循環。
- 若是反轉(updn=0)，當數到(0, 19, 18, …, 2, **1**), 0, …之1時表示反轉已經過一個循環。



Design 2: snake_run_cnt (3/4)

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

    input   clk, rst, updn;
    input
    // number of pattern counts from 0-to-19 or 19-to-0

    output
    // number of run counts from 0-to-9
    output  reg run_stop;

            run_cnt_tmp;

    always @(posedge clk or posedge rst) begin
        if (rst) begin
            run_cnt = 4'd0;
            run_stop = 1'b0;
            run_cnt_tmp = 4'd1;
        end
```

使用此module時，透過#(20)參數傳遞，將CNT_LENGTH設定為20

Design 2: snake_run_cnt (4/4)

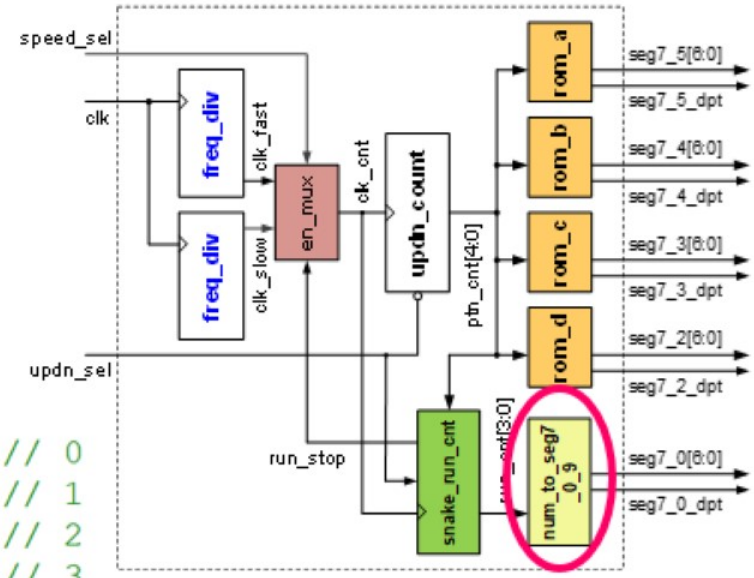
```
        else begin
                // count up
            if (( updn ==        && ptn_cnt ==                                ) ||
                // count down
                ( updn ==        && ptn_cnt ==        ))
                run_cnt_tmp = run_cnt_tmp +

            if (run_cnt_tmp <= 4'd9)
                run_cnt =
            else
                run_stop =
        end
    end
endmodule
```

預防當run_cnt=9時，再遞增1後，run_cnt之值會變成10，無法顯示於seg7，因此將遞增作用於run_cnt_tmp，若run_cnt_tmp<=9，再將run_cnt之值設定為run_cnt_tmp。

Design 3: num_to_seg7_0_9

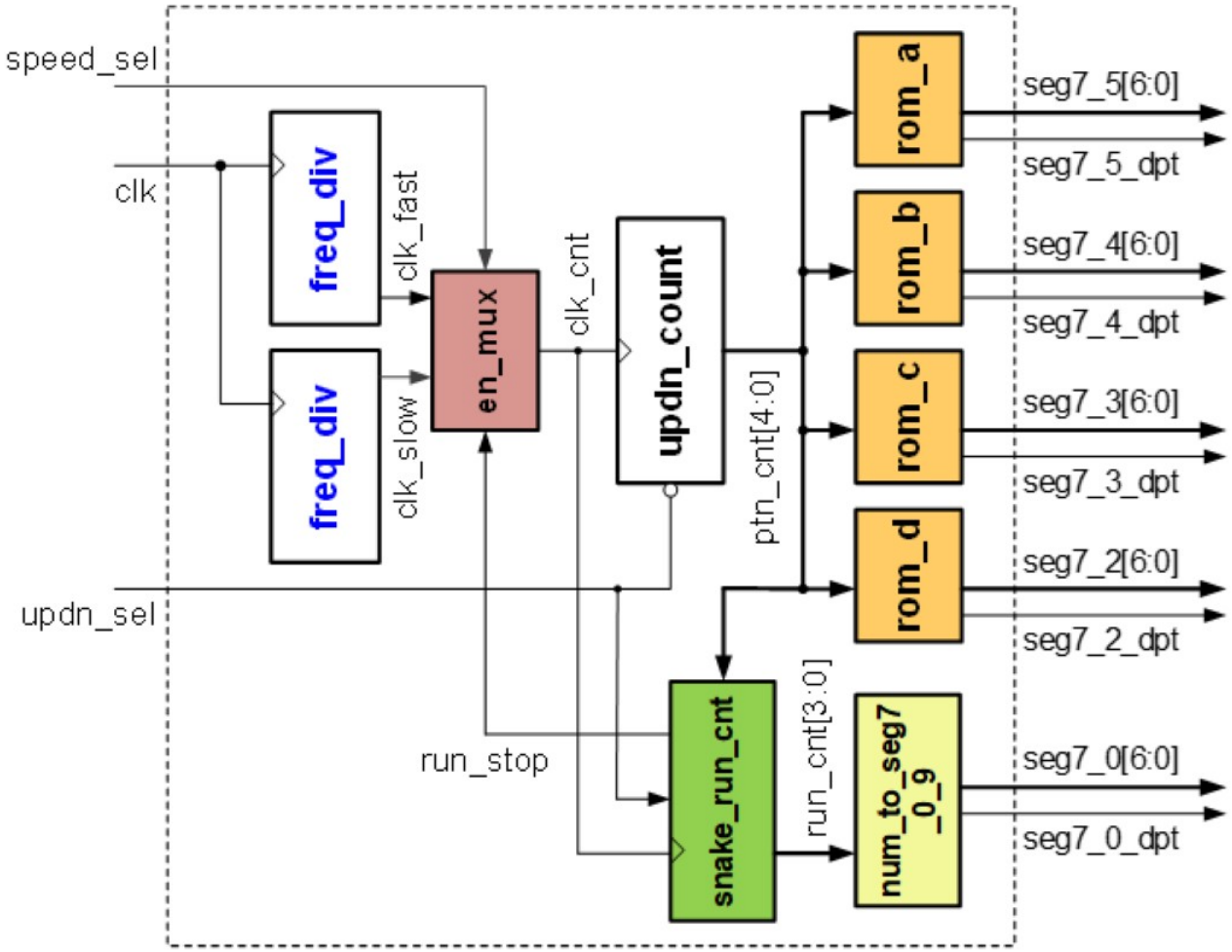
- 修改lab04之num_to_seg7 (其顯示之數字為0~7及-1~-8)，成為新的module，以顯示0~9。

```
module num_to_seg7_0_9(num, seg7, dpt);
    input       [3:0] num;
    output      [6:0] seg7;
    output      dpt;
    reg         [6:0] seg7;
    reg         dpt;

    always @(num)
        case(num)                   //  gfedcba, dpt
            4'b0000:{seg7, dpt} = {7'b1000000, 1'b1}; // 0
            4'b0001:{seg7, dpt} = {7'b1111001, 1'b1}; // 1
            4'b0010:{seg7, dpt} = {7'b0100100, 1'b1}; // 2
            4'b0011:{seg7, dpt} = {7'b0110000, 1'b1}; // 3
            4'b0100:{seg7, dpt} = {7'b0011001, 1'b1}; // 4
            4'b0101:{seg7, dpt} = {7'b0010010, 1'b1}; // 5
            4'b0110:{seg7, dpt} = {7'b0000010, 1'b1}; // 6
            4'b0111:{seg7, dpt} = {7'b1111000, 1'b1}; // 7
            default:{seg7, dpt} = {7'b1111111, 1'b1};
        endcase
endmodule
```



Design 4: snake_top (1/4)

- 設計snake_top.v並加入至project中，整合所有電路模組。

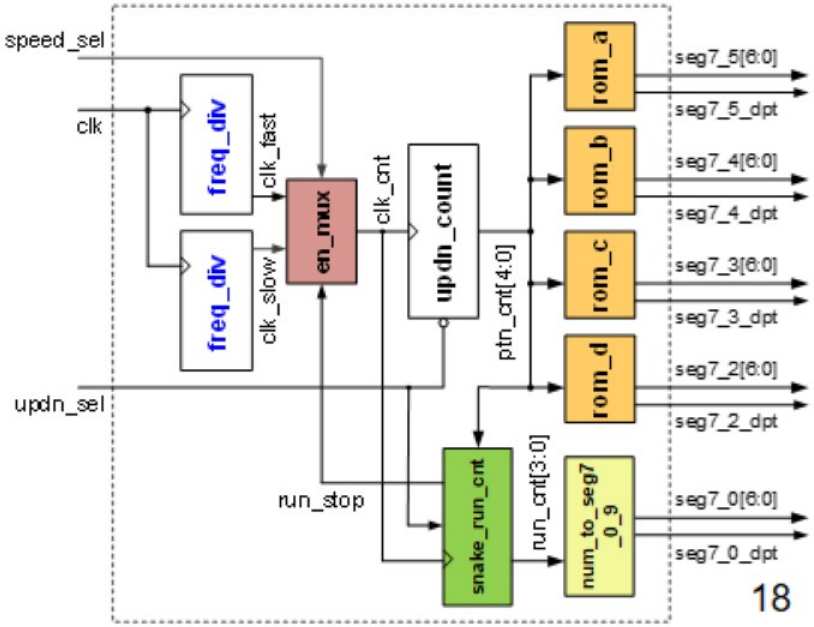


Design 4: snake_top (2/4)

```
module snake_top(clk, rst, speed_sel, updn_sel,
seg7_5, seg7_5_dpt, seg7_4, seg7_4_dpt,
seg7_3, seg7_3_dpt, seg7_2, seg7_2_dpt,
seg7_0, seg7_0_dpt);
```

```
input
output
output
```

```
wire
```

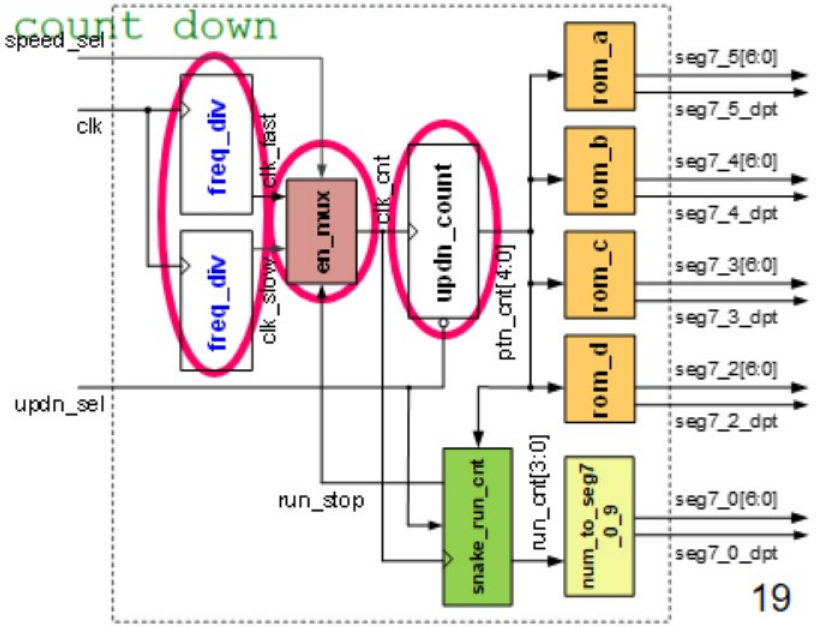


Design 4: snake_top (3/4)

```
freq_div
freq_div

wire    run_stop;
assign
// run_stop=1 => clk_cnt=0
// run_stop=0 => (speed_sel=1 => fast, speed_sel=0 => slow

wire    [4:0] ptn_cnt;
updn_count
// updn_sel=0 => count up, updn_sel=1 => count down
```

rst必須先反相: ~rst

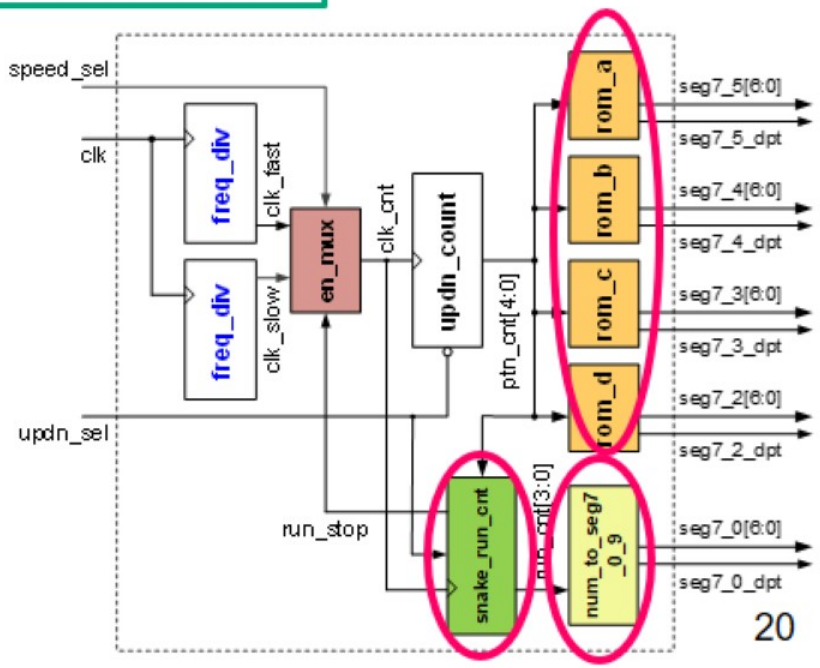
由於按鍵關係，向下撥為正轉(updn_sel=0)，
向上撥為反轉(updn_sel=1)，因此updn_sel
傳入updn_count時，必須先反相: ~updn_sel



Design 4: snake_top (4/4)

```
    rom_a
    rom_b
    rom_c
    rom_d

    wire     [3:0] run_cnt;
    snake_run_cnt

    num_to_seg7_0_9
endmodule
```

rst及updn_sel必須先
反相: ~rst, ~updn_sel



Pin Assignments

speed_sel

updn_sel

0: slow
1: fast

0: count up
1: count down

seg7_5 ~ seg7_2

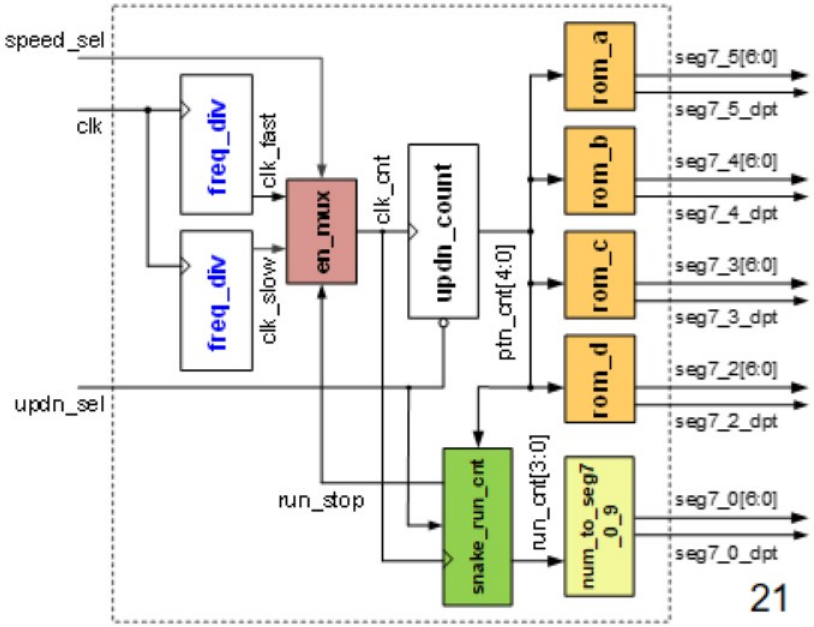
seg7_0

clk

Clocks: 50 MHZ
CLK1: P11
CLK2: N14

rst

KEY0: B8 **Push: 0**
KEY1: A7 **Not push: 1**



File List

- Verilog files
  - freq_div.v
  - updn_count.v
  - rom_a.v
  - rom_b.v
  - rom_c.v
  - rom_d.v
  - snake_run_cnt.v
  - num_to_seg7_0_9.v
  - snake_top.v

實驗結果驗收

- 請老師或助教驗收snake_top 電路於實驗板之快慢及正反轉貪食蛇動畫行為