

# **FPGA Lab-04**

---

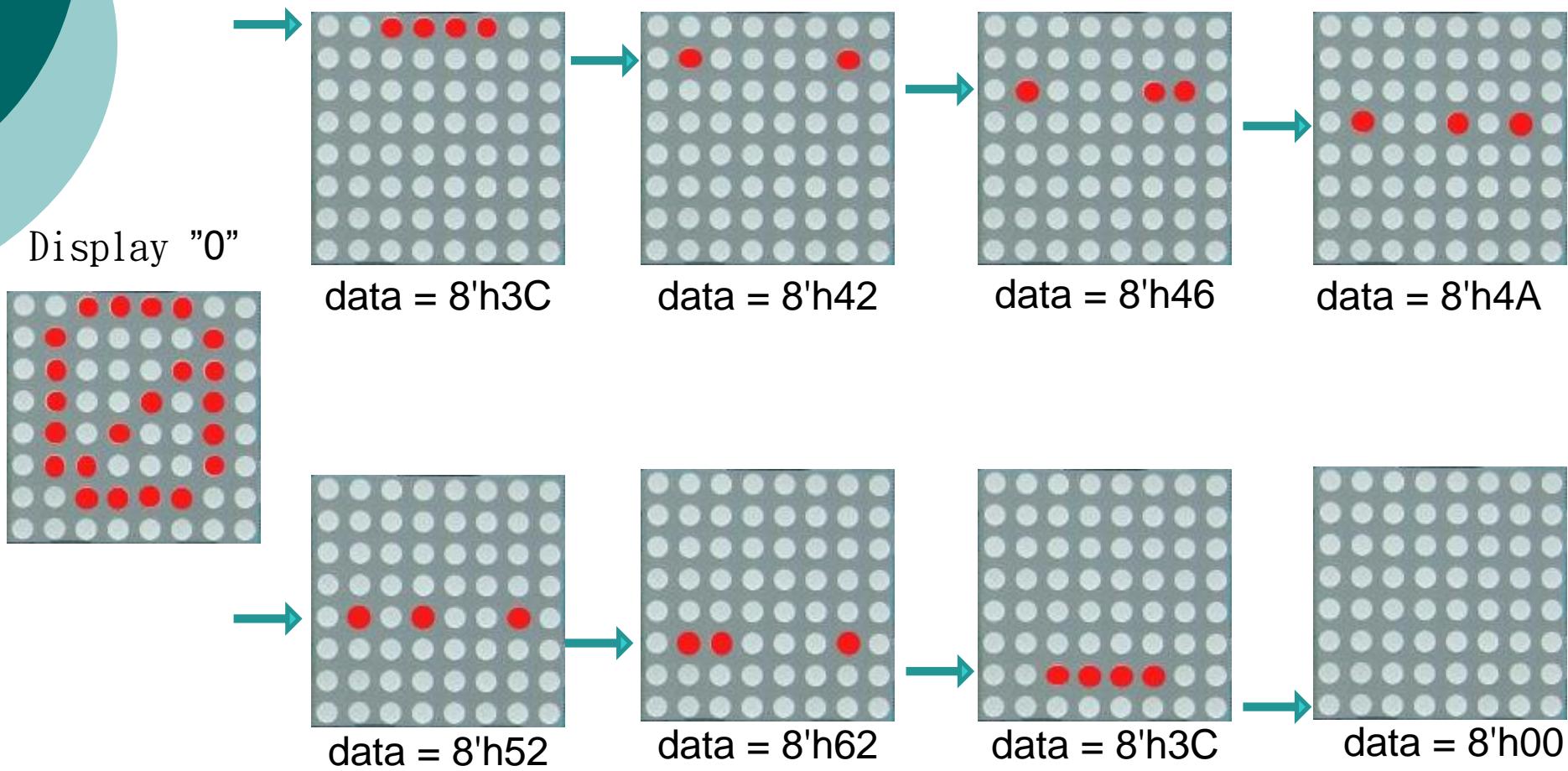
**8\*8 LED MATRIX APPLICATION**

# Lab

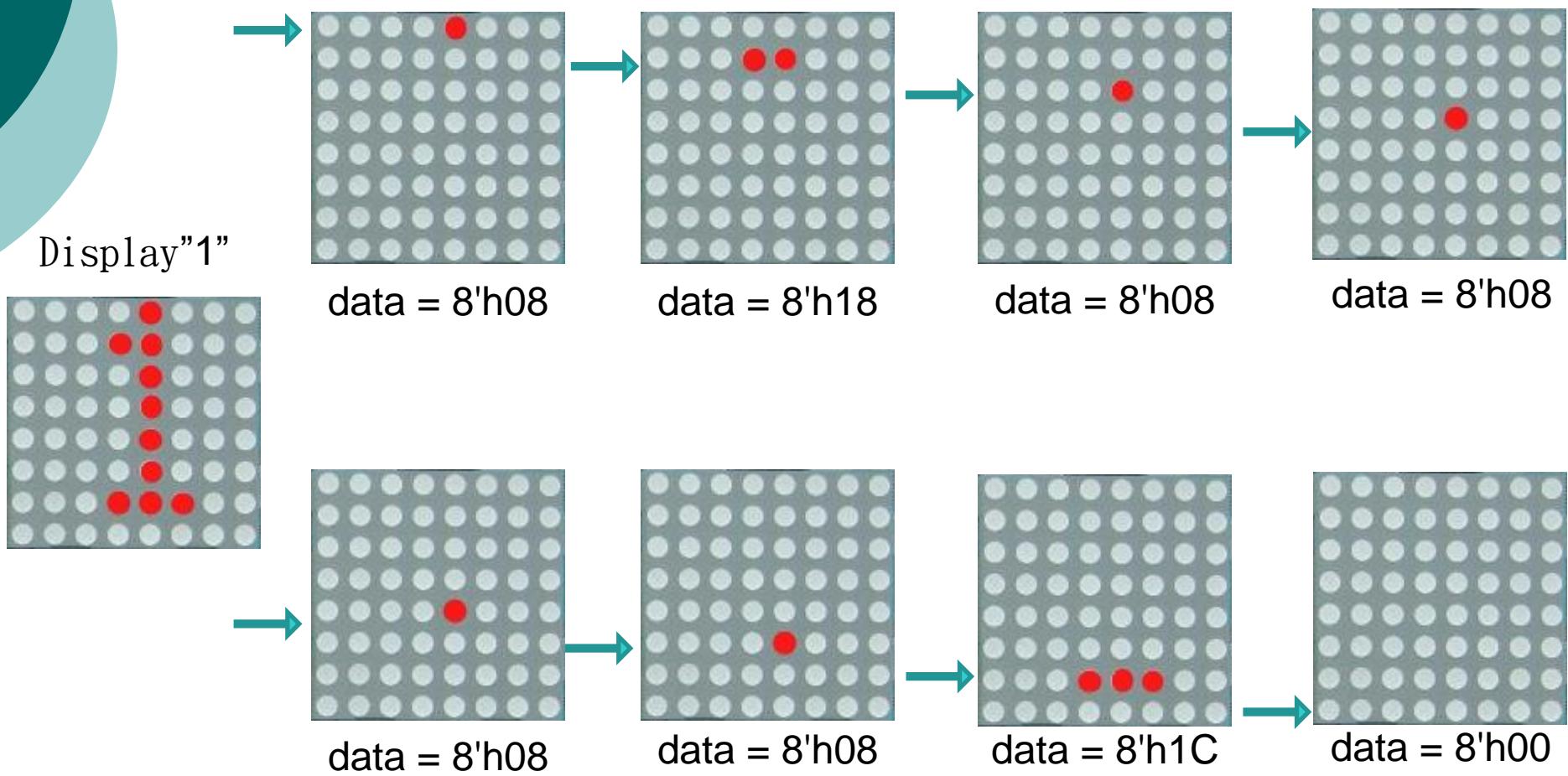
---

- Lab 1: Display 0 to 9 on the LED matrix, choose any colour you prefer
  
- Lab 2: The LED matrix moves from the bottom to the top to display the numbers 0-9 in sequence, and the color of the display can be selected as yellow, green and red.

# LED Matrix Display Principle(1)

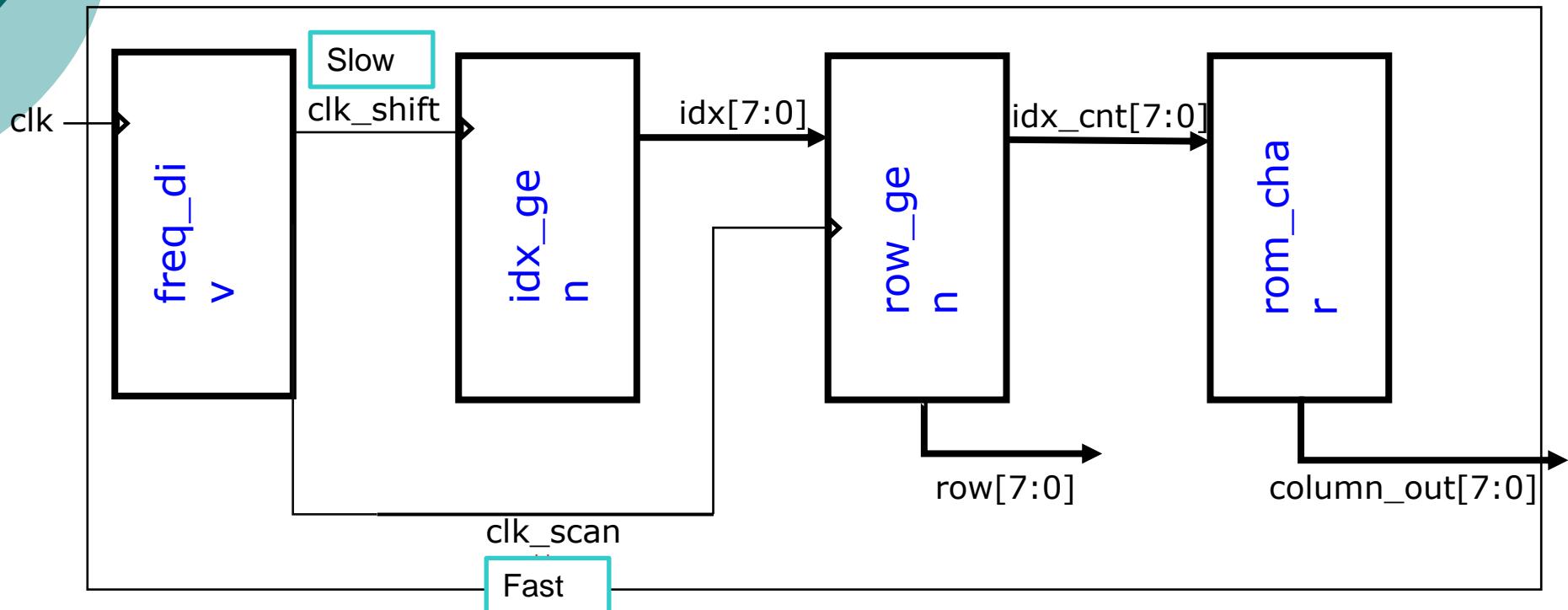


# LED Matrix Display Principle (2)



# Lab 1: Display the numbers 0-9 sequentially on the LED matrix

## ○ Programming architecture :



# Program Architecture

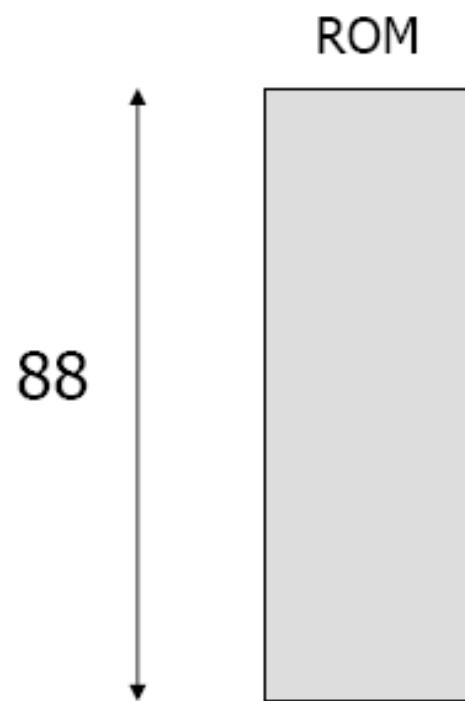
---

- rom\_char.v
- idx\_gen.v
- row\_gen.v
- freq\_div.v (Please copy from previous code)
- color\_matrix\_top.v

## rom\_char.v

---

- Save the number of characters to be displayed



# rom\_char.v

rom\_char : (餵address進去，生出data)

○○○○○○○○	addr=7'd0	data=8'h00
○○○○○○○○	addr=7'd1	data=8'h00
○○○○○○○○	addr=7'd2	data=8'h00
○○○○○○○○	addr=7'd3	data=8'h00
○○○○○○○○	addr=7'd4	data=8'h00
○○○○○○○○	addr=7'd5	data=8'h00
○○○○○○○○	addr=7'd6	data=8'h00
○○○○○○○○	addr=7'd7	data=8'h00

○○●●●●○○	addr=7'd8	data=8'h3C
●●○○○○●○	addr=7'd9	data=8'h42
●●○○○○●○	addr=7'd10	data=8'h46
●●○○○○●○	addr=7'd11	data=8'h4A
●●○○○○●○	addr=7'd12	data=8'h52
●●○○○○●○	addr=7'd13	data=8'h62
●●○○○○●○	addr=7'd14	data=8'h3C
○○○○○○○○	addr=7'd15	data=8'h00

○○●●○○○○	addr=7'd16	data=8'h08
○○●●○○○○	addr=7'd17	data=8'h18
○○●●○○○○	addr=7'd18	data=8'h08
○○●●○○○○	addr=7'd19	data=8'h08
○○●●○○○○	addr=7'd20	data=8'h08
○○●●○○○○	addr=7'd21	data=8'h08
○○●●○○○○	addr=7'd22	data=8'h1C
○○○○○○○○	addr=7'd23	data=8'h00

...

○○●●●●○○	addr=7'd80	data=8'h
●●○○○○●○	addr=7'd81	data=8'h
●●○○○○●○	addr=7'd82	data=8'h
●●○○○○●○	addr=7'd83	data=8'h
●●○○○○●○	addr=7'd84	data=8'h
●●○○○○●○	addr=7'd85	data=8'h
●●○○○○●○	addr=7'd86	data=8'h
○○○○○○○○	addr=7'd87	data=8'h

# 程式碼 rom\_char.v

---

- module rom\_char(addr, data);
- input[6:0]addr;
- output[7:0]data;
- reg[7:0]data;
- always@(addr) begin
- case(addr)
- 7'd0: data = 8'h00; 7'd1: data = 8'h00; // Blank
- 7'd2: data = 8'h00; 7'd3: data = 8'h00;
- 7'd4: data = 8'h00; 7'd5: data = 8'h00;
- 7'd6: data = 8'h00; 7'd7: data = 8'h00;

(8'h00 can be written as 8'b0000\_0000)

- 
- 7'd8: data = 8'h3C; 7'd9: data = 8'h42; // 0
  - 7'd10: data = 8'h46; 7'd11: data = 8'h4A;
  - 7'd12: data = 8'h52; 7'd13: data = 8'h62;
  - 7'd14: data = 8'h3C; 7'd15: data = 8'h00;
  - 7'd16: data = 8'h08; 7'd17: data = 8'h18;// 1
  - 7'd18: data = 8'h08; 7'd19: data = 8'h08;
  - 7'd20: data = 8'h08; 7'd21: data = 8'h08;
  - 7'd22: data = 8'h1C; 7'd23: data = 8'h00;
  - 7'd24: data = 8'h3C; 7'd25: data = 8'h42;// 2
  - 7'd26: data = 8'h42; 7'd27: data = 8'h04;
  - 7'd28: data = 8'h08; 7'd29: data = 8'h10;
  - 7'd30: data = 8'h7E; 7'd31: data = 8'h00;

- 
- 7'd32: data = 8'h3C;
  - 7'd34: data = 8'h02;
  - 7'd36: data = 8'h02;
  - 7'd38: data = 8'h3C;
  - 7'd40: data = 8'h1C;
  - 7'd42: data = 8'h44;
  - 7'd44: data = 8'h44;
  - 7'd46: data = 8'h04;
  - 7'd48: data = 8'h7E;
  - 7'd50: data = 8'h40;
  - 7'd52: data = 8'h02;
  - 7'd54: data = 8'h3C;
  - Please complete the data for 6,7,8,9
  - endcase
  - end
  - endmodule
- 7'd33: data = 8'h42;// 3
  - 7'd35: data = 8'h3C;
  - 7'd37: data = 8'h42;
  - 7'd39: data = 8'h00;
  - 7'd41: data = 8'h24;// 4
  - 7'd43: data = 8'h44;
  - 7'd45: data = 8'h7E;
  - 7'd47: data = 8'h00;
  - 7'd49: data = 8'h40;//5
  - 7'd51: data = 8'h7C;
  - 7'd53: data = 8'h42;
  - 7'd55: data = 8'h00;

## idx\_gen.v

---

- Generates the starting position of the block to be displayed.
- Follows the clock sequence

(Use a slower clk to generate idx)

Output the first row of each 8\*8 array image :

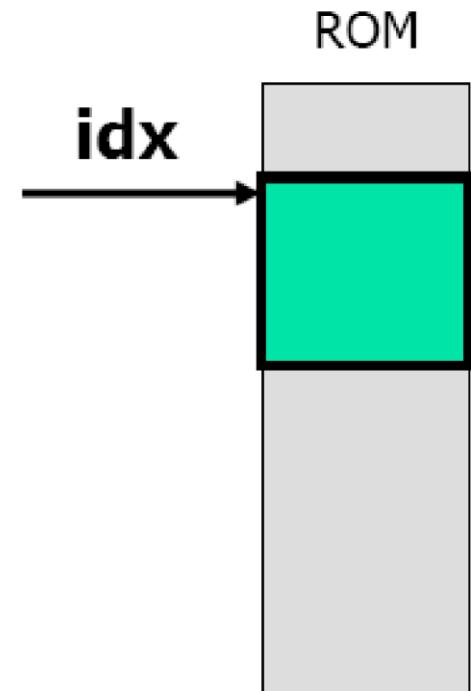
idx=7'd0

idx=7'd8

idx=7'd16

...

idx=7'd80 (The first row of the last 8x8 array.)



# Code for idx\_gen.v

---

```
○ module idx_gen(clk, rst, idx);
○ input clk, rst;
○ output[6:0] idx;
○ reg[6:0]idx;
○ always@(posedge clk or posedge rst) begin
○   if(rst)
○     idx= 7'd0;
○   else if(idx==7'd80)
○     idx= 7'd0;
○   else
○     idx=idx+7'd08;
○ end
○ endmodule
```

# row\_gen.v

- 產生所要顯示區塊的每一個位置
- 隨著時脈(較快)依序循環

row\_gen : (餵idx進去，生出idx\_cnt，這個idx\_cnt就是要丟到rom\_char去當address位址)  
(以比較快的clk輪流將每一排LED致能，致能就是enable，讓每一排LED輪流亮)

當idx\_cnt=7'd0時：

row=8'b1000\_0000 (致能第一排LED，意即第一排LED燈都打開)

cnt=3'd0 (0~7的計數器)

idx\_cnt=7'd0+3'd0=7'd0 (第一排LED亮相對應位址7'd0的row燈號)

row=8'b0100\_0000 (致能第二排LED)

cnt=3'd1

idx\_cnt=7'd0+3'd1=7'd1 (第二排LED亮相對應位址7'd1的row燈號)

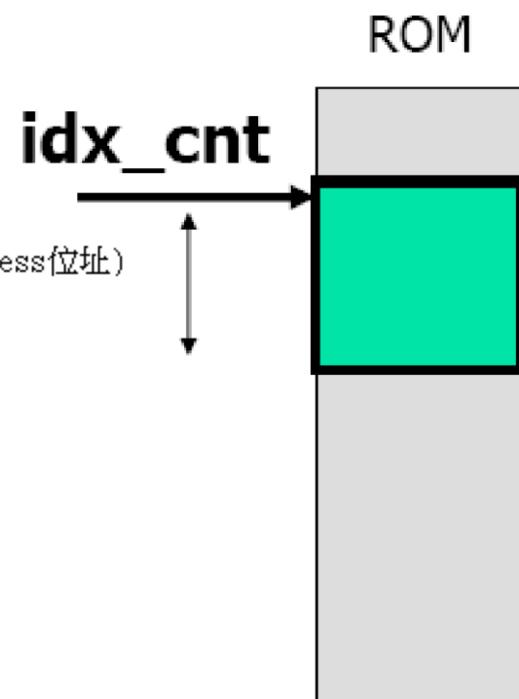
...

row=8'b0000\_0001 (致能第八排LED)

cnt=3'd7

idx\_cnt=7'd0+3'd7=7'd7 (第八排LED亮相對應位址7'd7的row燈號)

→產生出第一個8\*8矩陣圖



# 程式碼 row\_gen.v

---

```
○ module row_gen(clk, rst, idx, row, idx_cnt);
○   input clk, rst;
○   input[6:0]idx;
○   output[7:0] row;
○   output[6:0]idx_cnt;
○   reg[7:0] row;
○   reg[6:0]idx_cnt;
○   reg[2:0]cnt;
○   always@(posedge clk or posedge rst) begin
○     if(rst) begin
○       row <= 8'b0000_0001;
○       cnt <= 3'd0;
○       idx_cnt <= 7'd0;
○     end
○     else begin
○       row <= ~yourcode~      ;//(Turn on the LED for each row)
○       cnt <= ~yourcode~    ;//(count from 0 to 7)
○       idx_cnt <= ~yourcode~;//(count from 0 to 7)
○     end
○   end
○ endmodule
```

# Main module color\_matrix\_top.v

---

- module color\_matrix\_top.v(clk, rst, row, sel, column\_green, column\_red);
- input clk, rst;
- input[1:0] sel; //select the green or red LED to lightup  
//pin AA15 ,AA14
- output[7:0] row, column\_green, column\_red;  
//row:pin T22 ,R21 ,C6 ,B6 ,B5 ,A5 ,B7 ,A7  
//column\_green/red compare the Lab1 pin
- wire clk\_shift, clk\_scan;
- wire[6:0] idx, idx\_cnt;
- wire[7:0] column\_out;

- 
- assign column\_green= (sel== 2'b01 || sel== 2'b11)? column\_out: 8'b0;
  - assign column\_red= (sel== 2'b10 || sel== 2'b11)? column\_out: 8'b0;
  - freq\_div#(22) M1 (your code);
  - freq\_div#(12) M2 (your code);
  - idx\_gen M3 (your code);
  - row\_gen M4 (your code);
  - rom\_char M5 (your code);
  - endmodule

Lab 2: The LED matrix moves from the bottom to the top to display the numbers 0-9 in sequence

---

- Only change the idx\_gen.v
- Those who have done so can think about how to make the numbers 9-0 move from **top to bottom.**(Bonus Question)

idx\_gen：(由下往上)

idx=7'd0  
idx=7'd1  
idx=7'd2  
...  
idx=7'd80

代表什麼呢？

產生出的idx一樣會被餵給row\_gen

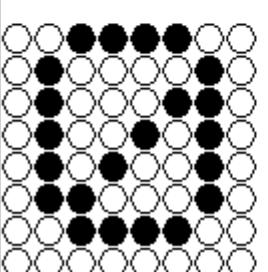
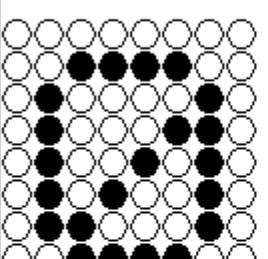
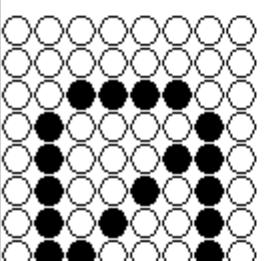
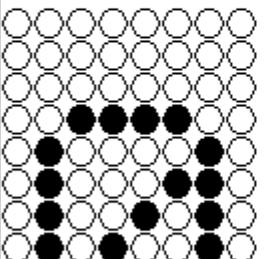
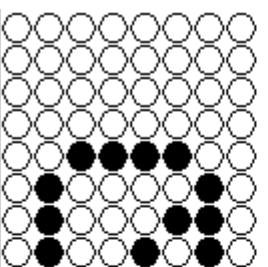
row\_gen產生出的idx\_cnt會產生出以下的圖形，看起來就像是數字由下往上移動：

○○○○○○○○      addr=7'd0      data=8'h00  
○○○○○○○○      addr=7'd1      data=8'h00  
○○○○○○○○      addr=7'd2      data=8'h00  
○○○○○○○○      addr=7'd3      data=8'h00  
○○○○○○○○      addr=7'd4      data=8'h00  
○○○○○○○○      addr=7'd5      data=8'h00  
○○○○○○○○      addr=7'd6      data=8'h00  
○○○○○○○○      addr=7'd7      data=8'h00

○○○○○○○○      addr=7'd1      data=8'h00  
○○○○○○○○      addr=7'd2      data=8'h00  
○○○○○○○○      addr=7'd3      data=8'h00  
○○○○○○○○      addr=7'd4      data=8'h00  
○○○○○○○○      addr=7'd5      data=8'h00  
○○○○○○○○      addr=7'd6      data=8'h00  
○○○○○○○○      addr=7'd7      data=8'h00  
○○○○○○○○      addr=7'd8      data=8'h3C

○○○○○○○○      addr=7'd2      data=8'h00  
○○○○○○○○      addr=7'd3      data=8'h00  
○○○○○○○○      addr=7'd4      data=8'h00  
○○○○○○○○      addr=7'd5      data=8'h00  
○○○○○○○○      addr=7'd6      data=8'h00  
○○○○○○○○      addr=7'd7      data=8'h00  
○○○○○○○○      addr=7'd8      data=8'h3C  
○○○○○○○○      addr=7'd9      data=8'h42

○○○○○○○○      addr=7'd3      data=8'h00  
○○○○○○○○      addr=7'd4      data=8'h00  
○○○○○○○○      addr=7'd5      data=8'h00  
○○○○○○○○      addr=7'd6      data=8'h00  
○○○○○○○○      addr=7'd7      data=8'h00  
○○○○○○○○      addr=7'd8      data=8'h3C  
○○○○○○○○      addr=7'd9      data=8'h42  
○○○○○○○○      addr=7'd10      data=8'h46



addr=7'd4      data=8'h00  
addr=7'd5      data=8'h00  
addr=7'd6      data=8'h00  
addr=7'd7      data=8'h00  
addr=7'd8      data=8'h3C  
addr=7'd9      data=8'h42  
addr=7'd10      data=8'h46  
addr=7'd11      data=8'h4A

addr=7'd5      data=8'h00  
addr=7'd6      data=8'h00  
addr=7'd7      data=8'h00  
addr=7'd8      data=8'h3C  
addr=7'd9      data=8'h42  
addr=7'd10      data=8'h46  
addr=7'd11      data=8'h4A  
addr=7'd12      data=8'h52

addr=7'd6      data=8'h00  
addr=7'd7      data=8'h00  
addr=7'd8      data=8'h3C  
addr=7'd9      data=8'h42  
addr=7'd10      data=8'h46  
addr=7'd11      data=8'h4A  
addr=7'd12      data=8'h52  
addr=7'd13      data=8'h62

addr=7'd7      data=8'h00  
addr=7'd8      data=8'h3C  
addr=7'd9      data=8'h42  
addr=7'd10      data=8'h46  
addr=7'd11      data=8'h4A  
addr=7'd12      data=8'h52  
addr=7'd13      data=8'h62  
addr=7'd14      data=8'h3C

addr=7'd8      data=8'h3C  
addr=7'd9      data=8'h42  
addr=7'd10      data=8'h46  
addr=7'd11      data=8'h4A  
addr=7'd12      data=8'h52  
addr=7'd13      data=8'h62  
addr=7'd14      data=8'h3C  
addr=7'd15      data=8'h00

Move the LED matrix from bottom to top to display the numbers 0-9 in sequence. You can choose either colour to display

