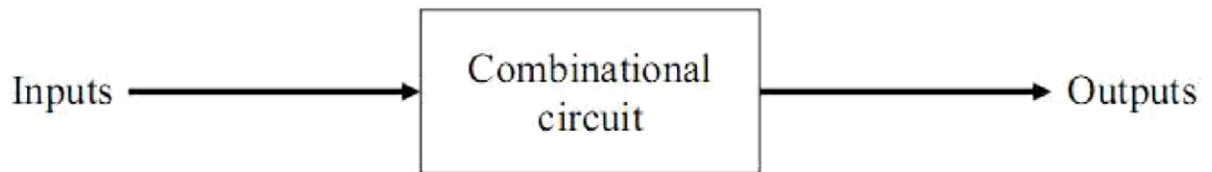


# LAB8

## 組合電路 vs. 序向電路

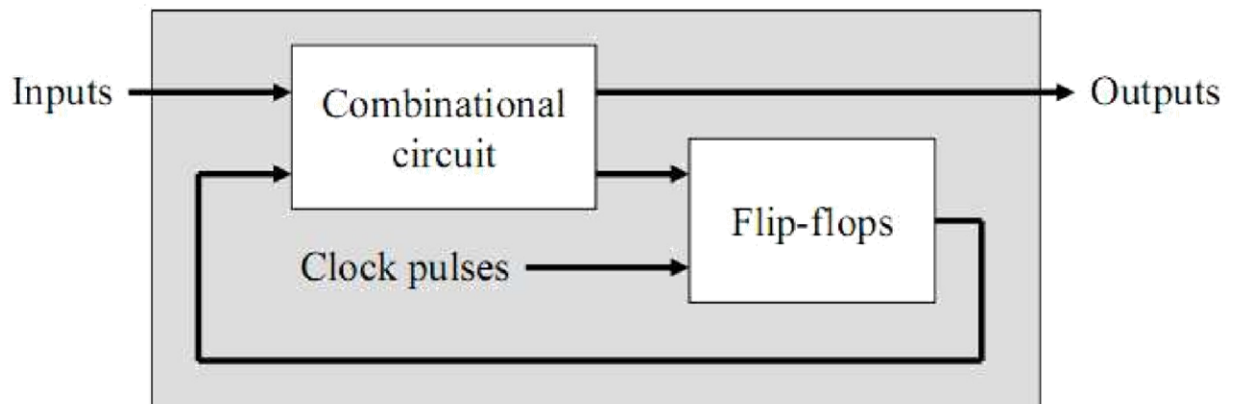
組合電路： $O \leftarrow I$

(例如：加法器)

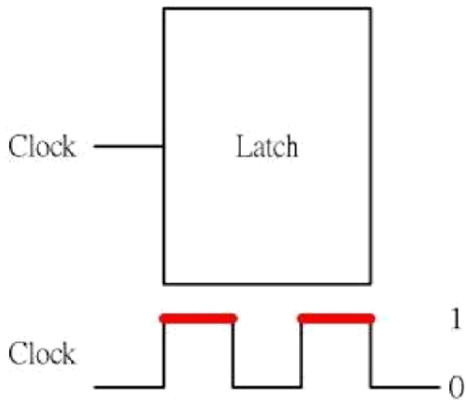


序向電路： $O \leftarrow I \times S$

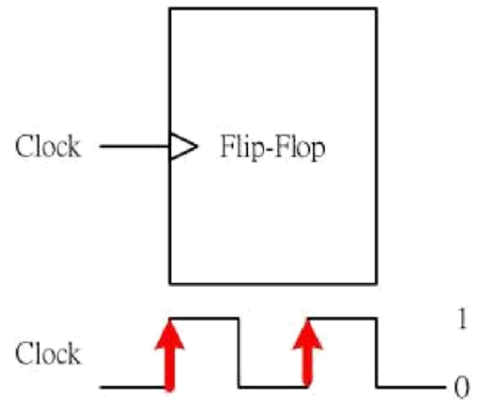
(例如：計數器)



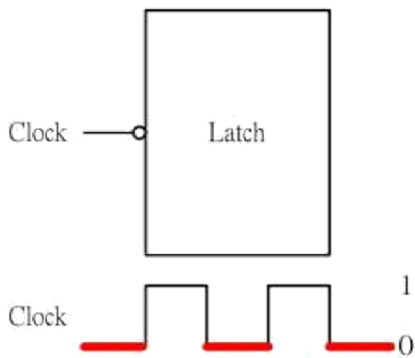
正層觸發



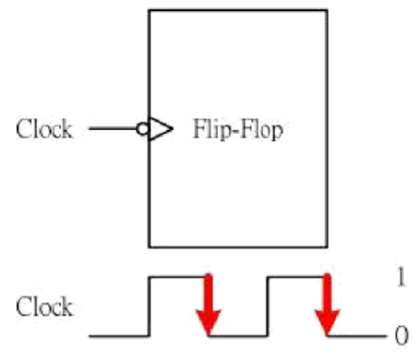
正緣觸發



負層觸發



負緣觸發



# Lab 8-1

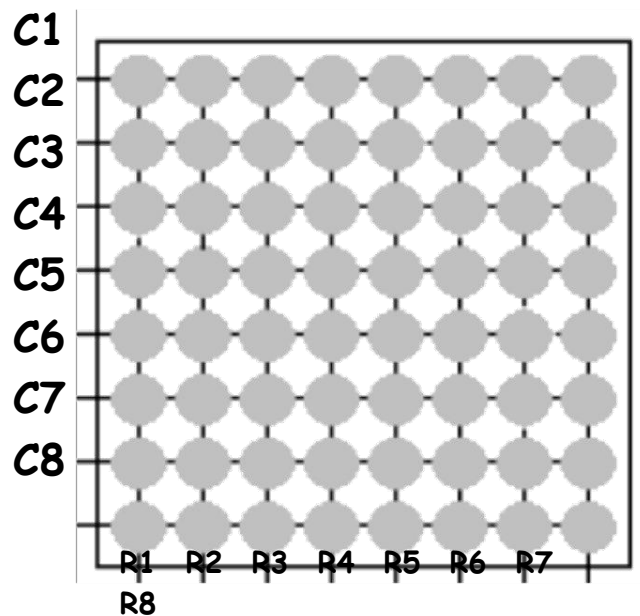
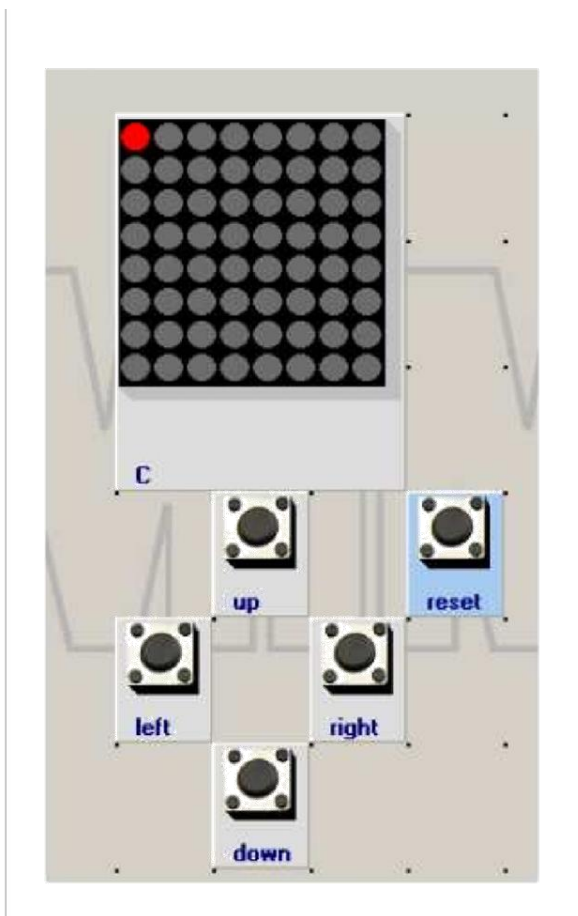
## 8X8 LED 矩陣與 De-bounce 電路

題目：使 8x8 LED 只有一個 LED 燈亮，按下 reset 後，亮點一開始在最左上上的 LED 燈，使用者可以按上下左右的 push button 來操控這個亮點的位置。  
(註：電路要按下 reset 後才開始運作。)

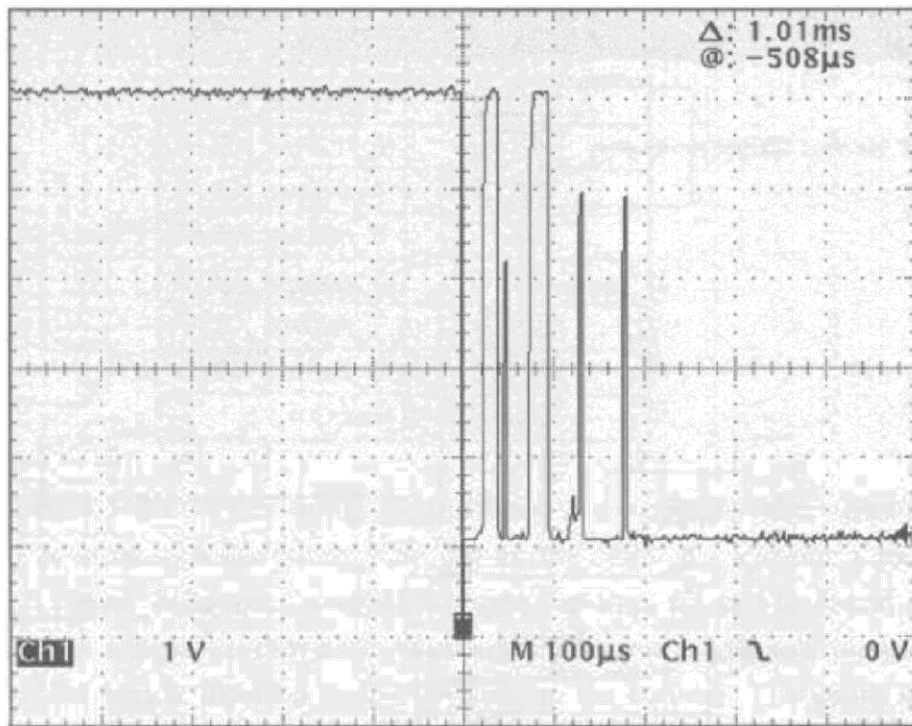
包含：

lab8\_1.v

oneshot.v



# De-bounce



實體 push button 在高低電位切換時的 bounce 現象

- 上圖為實際的 push button 的輸出在高低電位切換時會有震盪的現象，因為這個現象所以電路會不只讀到一個正(負)源觸發，故需要有 de-bounce 電路，如下的 oneshot.v

# oneshot.v

```

module oneshot (clk, din, dout);

    input clk;
    input din;

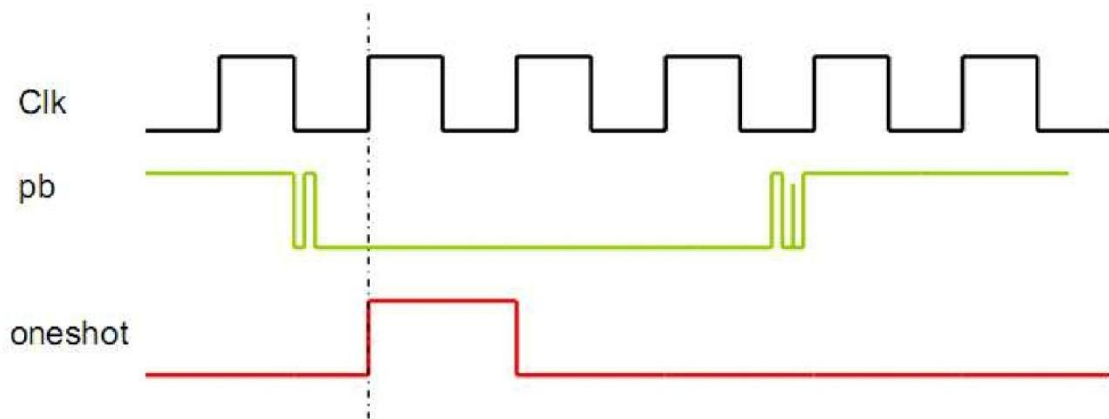
    output dout;

    reg[1:0] ss;

    always@(posedge clk)
    begin
        case (ss)
            2'b00:
                if (din) ss <= 2'b01;
            2'b01:
                ss <= 2'b10;
            2'b10:
                if (~din)
                    ss <= 2'b00;
            default:
                ss <= 2'b00;
        endcase
    end

    assign dout = ss[0];
endmodule

```



# lab8\_1.v

```
module lab8_1 (clk,reset,up,down,left,right,R,C);
input clk;
input reset;

input up,down,left,right;
output [7:0] R,C;

reg [7:0]R,C;
reg [7:0]R_t,C_t;
wire up_oneshot,down_oneshot,left_oneshot,right_oneshot;

//De-bounce logic
oneshot u_oneshot1(clk,up,up_oneshot);
oneshot u_oneshot2(clk,down,down_oneshot);
oneshot u_oneshot3(clk,left,left_oneshot);
oneshot u_oneshot4(clk,right,right_oneshot);

//8X8 LED row control
always@(posedge clk)
begin
    if(reset)
    begin
        R<=8'b00000001;
    end
    else
    begin
        R<=R_t;
    end
end
end
```

```
always@(*)
begin
    if(down_one-shot)
    begin
        R_t[7:1]=R[6:0];
        R_t[0]=R[7];
    end
    else if(up_one-shot)
    begin
        R_t[6:0]=R[7:1];
        R_t[7]=R[0];
    end
    else
    begin
        R_t=R;
    end
end

///請同學完成 column control logic 程式碼
/*
..
..
*/
///
endmodule
```



# Lab 8-2

## 8X8 LED 矩陣進階操作

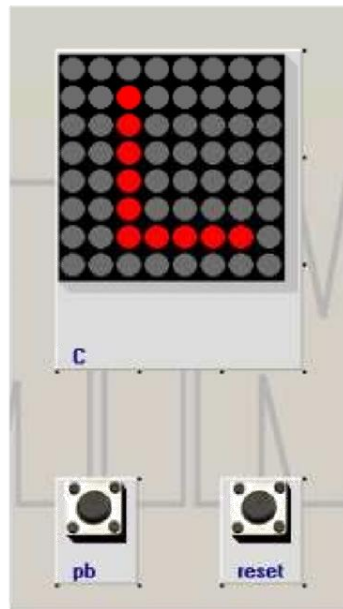
題目：使用 8x8 LED 依序顯示文字'L' 'O' 'G' 'I' 'C'。按下 reset 後一開始顯示'L'，每按一次 push button 就顯示下一個文字，到'C'再按下 push button 則回到'L'繼續循環。(註: 電路需按下 reset 後才開始運作。)

包含：

lab8\_2.v

clk\_div.v

oneshot.v



- Hint : 若顯示的結果不正確，請試著修改 LOGIC 的 control logic

# clk\_div.v

```
module clk_div(clk,clk_1000HZ) ;

input clk;
output clk_1000HZ;

reg [31:0]count_1000HZ;
reg clk_1000HZ;

always@(posedge clk)
begin
    if(count_1000HZ<32'd48000)
        count_1000HZ<=count_1000HZ+32'd1;
    else
        count_1000HZ<=32'd1;
end
always@(posedge clk)
begin
    if(count_1000HZ<=20'd24000)
        clk_1000HZ<=1'd1;
    else
        clk_1000HZ<=1'd0;
end
endmodule
```

# lab8\_2.v

```
module lab8_2 (clk,reset,pb,R,C);

    input clk;
    input reset;
    input pb;
    output [7:0]R;
    output [7:0]C;

    reg [7:0] R1,R2,R3,R4,R5;
    reg [7:0] R;
    reg [7:0] C;
    reg [2:0] counter;
    reg [2:0] display;

    wire pb_oneshot;
    wire reset_oneshot;
    wire clk_1000HZ;

    //de-bounce logic
    oneshot u_oneshot1(clk,pb,pb_oneshot);
    oneshot u_oneshot2(clk,reset,reset_oneshot);

    //clk_div logic
    clk_div u_clkdiv(clk,clk_1000HZ);

    always@(posedge clk_1000HZ or posedge reset_oneshot)
    begin
        if(reset_oneshot)
            counter<=3'd0;
        else
            counter<=counter +3'd1;
    end
end
```

```
//L
always@ (counter)
begin
    case (counter)
        3'd0: R1=8'b00000000;
        3'd1: R1=8'b00100000;
        3'd2: R1=8'b00100000;
        3'd3: R1=8'b00100000;
        3'd4: R1=8'b00100000;
        3'd5: R1=8'b00111100;
        3'd6: R1=8'b00000000;
        3'd7: R1=8'b00000000;

    endcase
end

//O
always@ (counter)
begin
    case (counter)
        3'd0: R2=8'b00000000;
        3'd1: R2=8'b01111100;
        3'd2: R2=8'b01000100;
        3'd3: R2=8'b01000100;
        3'd4: R2=8'b01000100;
        3'd5: R2=8'b01000100;
        3'd6: R2=8'b01111100;
        3'd7: R2=8'b00000000;

    endcase
end
//請同學完成 G I C 的 control logic
//G

//I

//C
```

### 挑戰題

題目：以 LAB8\_2 為基礎，讓 8x8 LED 不在由使用者按 push button 才切換顯示文字，而是隨時間慢慢依序顯示 LOGIC 五個英文字母並循環。  
(請利用或修改 clk\_div.v 來完成本題)

### 結報問題

1. 請說出若再 LAB8\_1 和 LAB8\_2 中無法使用 De-bounce 電路，則會造成電路出現什麼結果？
2. 你覺得 8x8 LED 還能有什麼實際或有趣的應用？請舉一個例子，並簡述如何用 verilog 來實現這個電路。