Laboratory 10

軟硬體共同實驗 (二): 按鈕器之進階控制實作

Department of Electrical Engineering
National Cheng Kung University

實驗目的

- 學習 8x8 LED 矩陣使用
- 利用HDL 實現 De-bounce 電路

使用器材

- 桌上型電腦
- Xilinx FPGA 板

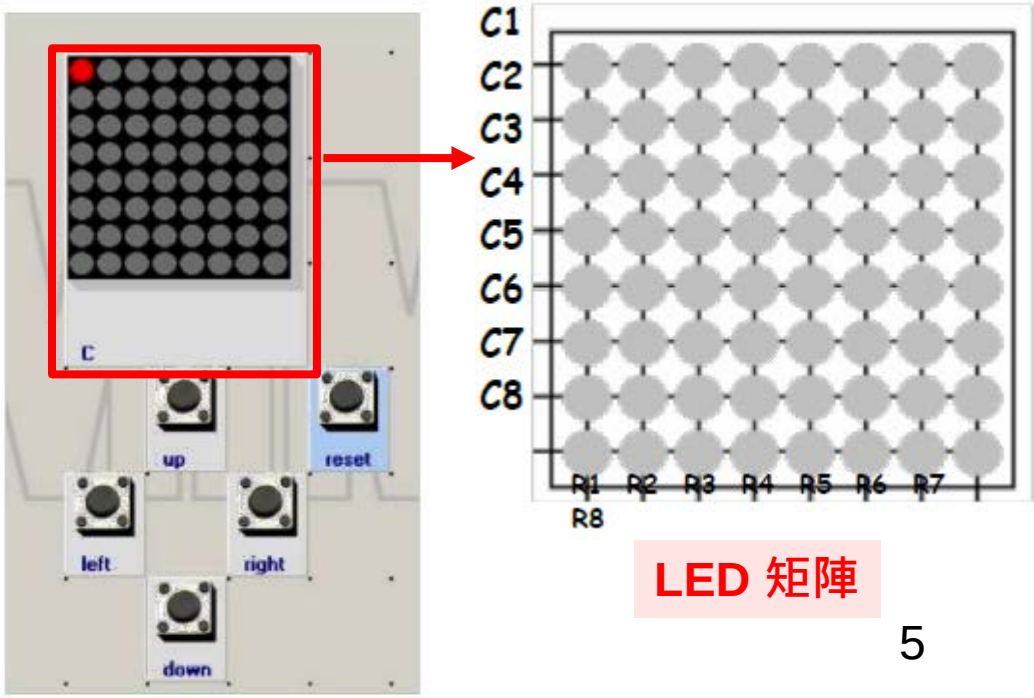
實作題

本次實作分為兩個基本實作、一個挑戰實作與一個結報問題

實作題(一) 8X8 LED 矩陣與 De-bounce 電路 (1/4)

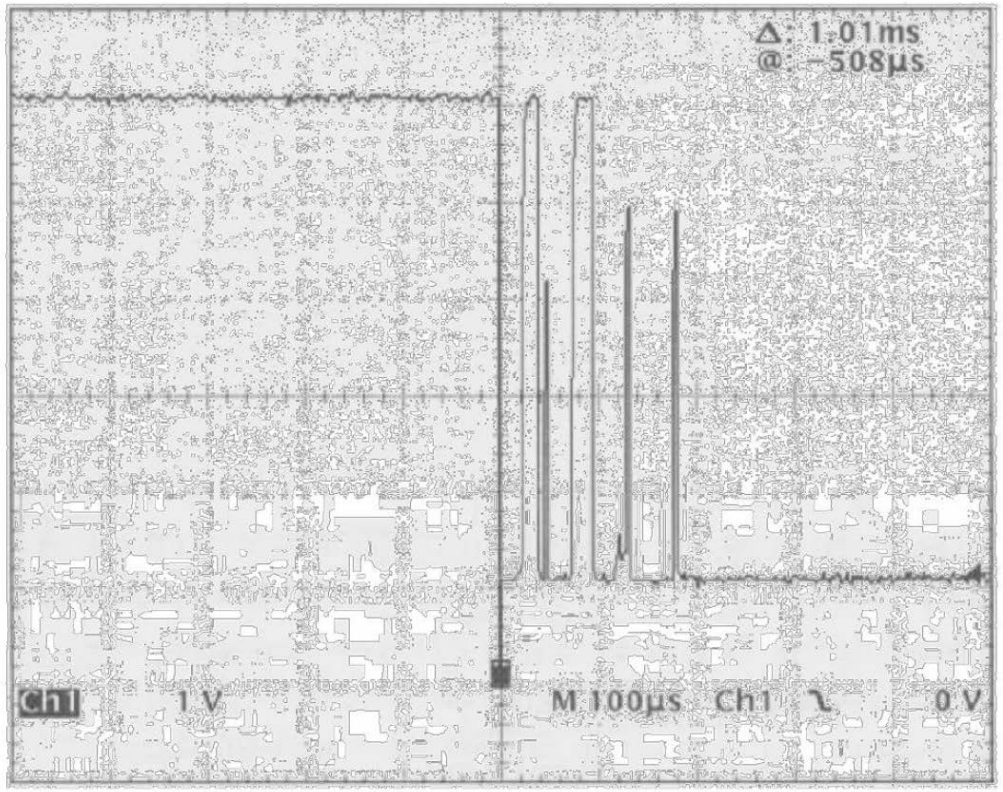
➢請實作8X8LED矩陣應用與De-bounce電路

- 8x8 LED 固定只有一個燈亮。按下 reset 後，亮點一開始在最左上的 LED 燈，使用者可以按左右 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)
- 需要完成下列*.v檔
  - oneshot.v
  - lab10_1.v



實作題(一) 8X8 LED 矩陣與 De-bounce 電路 (2/4)

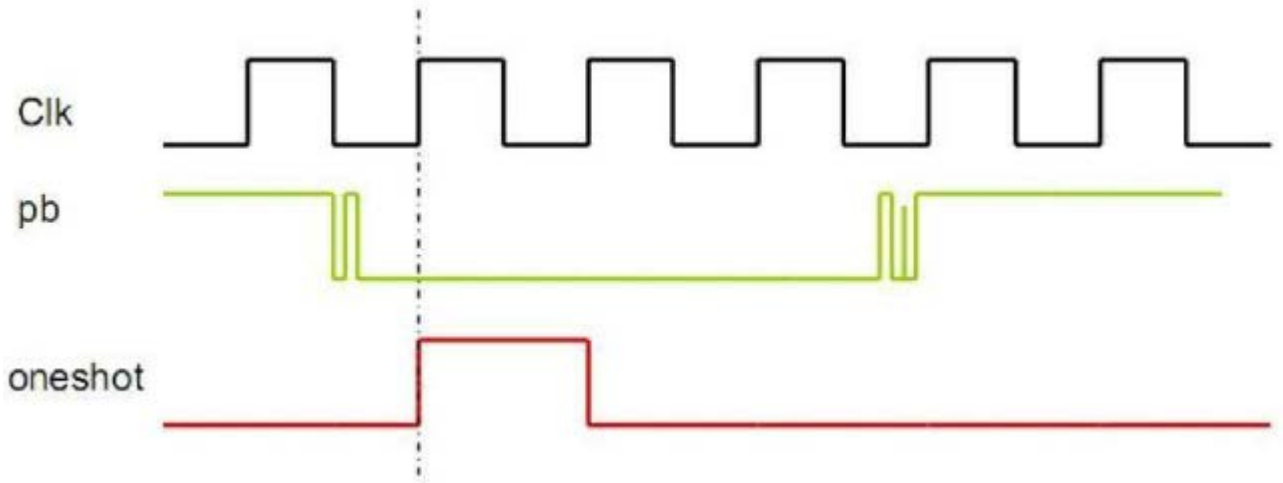
- 下圖為實際push botton的輸出在高低電位切換時會有震盪的現象，因為這個現象所以電路會不只讀到一個正(負)緣觸發，故需要有de-bounce電路(**oneshot.v**)來排除震盪。



實體 push button 在高低電位切換時的 bounce 現象

實作題(一) 8X8 LED 矩陣與 De-bounce 電路 (3/4)

```
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
```



oneshot.v

實作題(一) 8X8 LED 矩陣與 De-bounce 電路 (4/4)

```
module lab10_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
```

序向電路，
用於更新狀態
(<=)

```
always@(*)
begin
    if(down_oneshot)
    begin
        R_t[7:1]=R[6:0];
        R_t[0]=R[7];
    end
    else if(up_oneshot)
    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
```

組合電路，
用於產生下個狀態
(=)

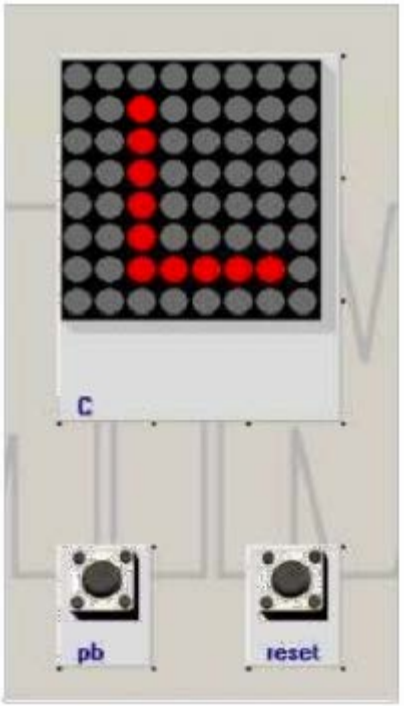
lab10_1.v

實作題(二) 8X8 LED 矩陣進階操作 (1/3)

➢請利用8X8LED矩陣與除頻器實作一個簡易幻燈片顯示系統

- 使 8x8 LED矩陣依序顯示文字”L”、”O”、”G”、”I”、”C”。按下 reset 後一開始顯示”L”，每一次 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: 電路要按下 reset 後才開始運作)
- 需要完成下列*.v檔
  - oneshot.v
  - clk_div.v
  - lab10_2.v

※Hint : 若顯示的結果不正確，請試著修改 L O G I C 的control logic



實作題(二) 8X8 LED 矩陣進階操作 (2/3)

```
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
```

clk_div.v

實作題(二) 8X8 LED 矩陣進階操作 (3/3)

```
module lab10_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
```

counter控制要顯示的行數，注意要跟後面掃描的部分同步，否則會有位移

```
//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
```

```
always@(posedge pb_oneshot or posedge reset_oneshot)
begin
    if(reset_oneshot)
        display<=3'd0;
    else if(display==3'd4)
        display<=3'd0;
    else
        display<=display+3'd1;
end

always@(*)
begin
    case(display)
    3'd0: R=R1;
    3'd1: R=R2;
    3'd2: R=R3;
    3'd3: R=R4;
    3'd4: R=R5;
    default: R=R1;
    endcase

end

always@(posedge clk_1000HZ or posedge reset_oneshot)
begin
    if(reset_oneshot)
        C<=8'b10000000;
    else
    begin
        C[7:1]<=C[6:0];
        C[0]<=C[7];
    end
end

endmodule
```

在這邊選擇要顯示的畫面(多工器)

8x8Matrix會根據除頻器的頻率來掃描與顯示畫面(一次只顯示一行)

lab10_2.v

挑戰題

- 請以 LAB10_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 LAB10_1 和 LAB10_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矩陣 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。