

## Lab 9-1 按鈕器進階操作

題目：設計一個 4 bits 計數器，當使用者按下 VeriLite PCB 版上的實體 Push-Button 一次才向上計數一次。按下 reset 則將計數器歸零。將結果顯示在 PCB 版的 LED 燈 D0~D3 上。

1. 請在” `c:\logiclab\<你自己的學號>` ” 的路徑下新增一資料夾 `lab9_1`
2. 開啟Quartus II，設定專案包含下列Verilog 檔：
  - `lab9_1.v`
3. 完成設計後請download 至VeriLite 版的EP1C6Q240C8 Device 來驗證結果。

LED		
D0	PIN_122	連接PCB上LED之腳位
D1	PIN_123	
D2	PIN_124	
D3	PIN_125	
D4	PIN_126	
D5	PIN_127	
D6	PIN_128	
D7	PIN_131	

Button		
BT0	PIN_132	連接PCB上BT0之腳位
BT1	PIN_133	連接PCB上BT1之腳位

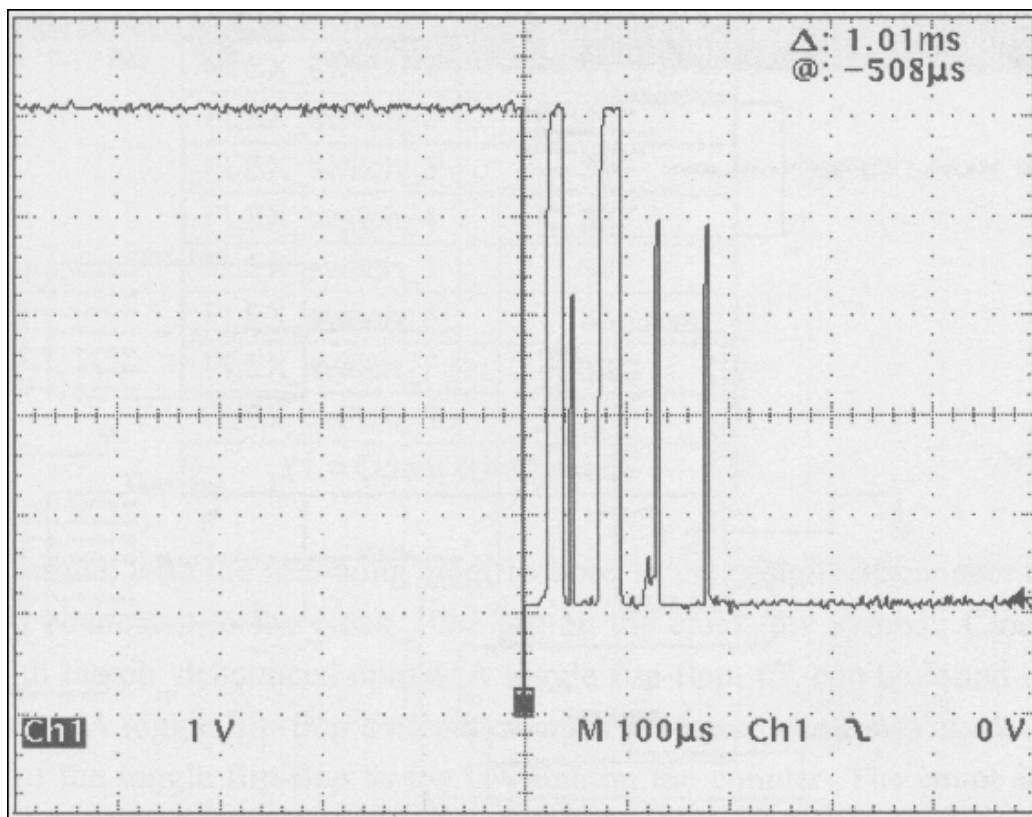
### Lab9\_1.v

```
module lab9_1(reset_n, pb_n, led);
    input  reset_n;
    input  pb_n;
    output [3:0] led;
    reg [3:0] counter;
    wire      reset, pb;
        assign reset = ~reset_n;
        assign pb = ~pb_n;
// 4 bits counter
/* PS: PCB 板上的push button 為共陰極 */
    always@(posedge reset or posedge pb)
    begin
        if(reset)
            counter <= 4'd0;
        else
            counter <= counter + 4'd1;
    end
// led 為共陽極
    assign led = ~counter;
endmodule
```

## Lab 9-2 按鈕器 De-bounce 電路

題目：在 lab9-1 我們發現當按下 push button 時，計數器會不只計數一次，這是由於 push button 在切換高低電位時會有 bounce 現象，故 lab9-2 加入了 de-bounce 電路 oneshot.v 來消除 bounce 的影響。

1. 請在”`c:\logiclab\<你自己的學號>`”的路徑下新增一資料夾 `lab9_2`
2. 開啟Quartus II，設定專案包含下列Verilog 檔：
  - `lab9_2.v`
  - `oneshot.v`
3. 完成設計後請download 至VeriLite 版的EP1C6Q240C8 Device 來驗證結果。



實體push button 在高低電位切換時的bounce 現象

註1：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)緣觸發，故使用實體push button時會有 de-bounce 電路，lab9-2 的oneshot.v 即為一個de-bounce 電路。 VeriInstrument 虛擬儀器的push button 不會有bounce 現象。

註2：VeriLite PCB 板上的push button 為共陰極，即按下後輸出為0，放開輸出為1。 VeriInstrument 的虛擬儀器push button 則為共陽極，按下輸出為1，放開輸出為0。

## oneshot.v

```
module oneshot(clock, din, dout);

    input  clock;
    input  din;

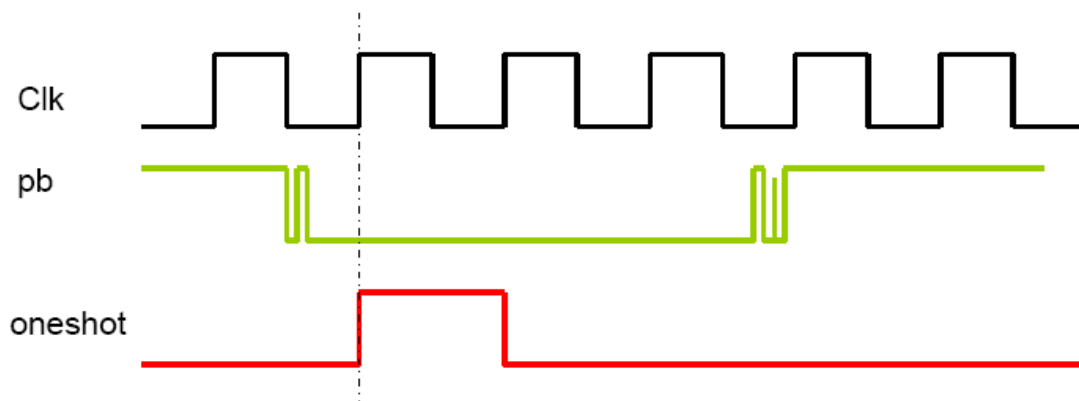
    output dout;

    reg [1:0] ss;

    always @(posedge clock)
    begin
        case(ss)
            2'b00:
                if(~din) //若push button 為共陰極用~din，共陽用din
                    ss<=2'b01;
            2'b01:
                ss<=2'b10;
            2'b10:
                if(din) //若push button 為共陰極用din，共陽用~din
                    ss<=2'b00;
            default:
                ss<=2'b00;
        endcase
    end

    assign dout=ss[0];

endmodule
```



## Lab9\_2.v

```
module lab9_2(clk, reset, pb, led);

    input          clk;
    input          reset;
    input          pb;
    output [3:0]   led;

    reg [3:0]      counter;
    wire           pb_oneshot;
    wire           reset_oneshot;

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

    // 4 bits counter
    /* PS: PCB 板上的push button 為共陰極 */

    always@(posedge reset_oneshot or posedge pb_oneshot)
    begin
        if(reset_oneshot)
            counter <= 4'd0;
        else
            counter <= counter + 4'd1;
    end

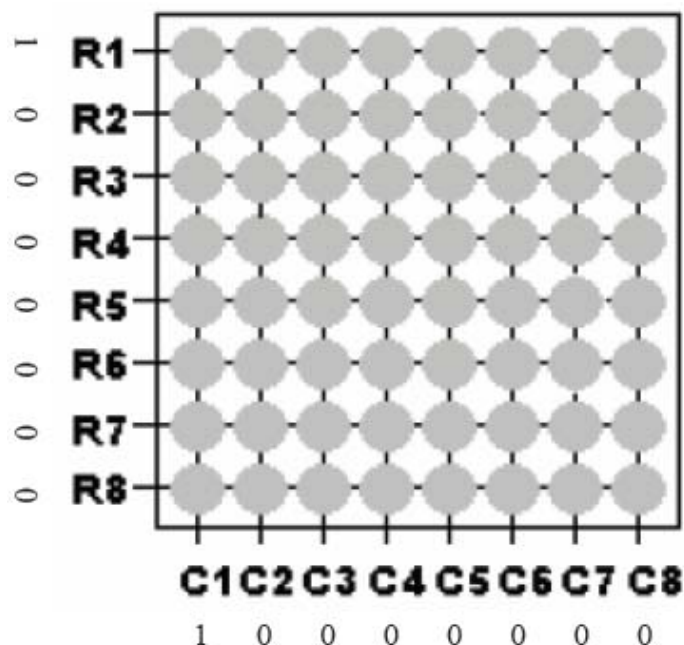
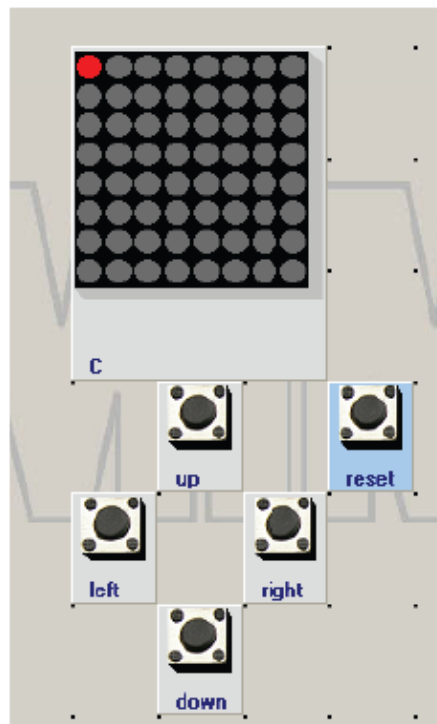
    // led 為共陽極
    assign led = ~counter;

endmodule
```

## Lab 9-3 8x8 LED 矩陣之基本操作

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。)

1. 請在”`c:\logiclab\<你自己的學號>`”的路徑下新增一資料夾 `lab9_3`
2. 開啟Quartus II，設定專案包含下列Verilog 檔：
  - `lab9_3.v`
  - `oneshot.v`
3. 完成設計後請download 至VeriLite 版的EP1C6Q240C8 Device 來驗證結果。



## *oneshot.v* (適用於虛擬 *push button*)

```
module oneshot(clock, din, dout);
    input  clock;
    input  din;
    output dout;

    reg [1:0] ss;

    always @(posedge clock)
    begin
        case(ss)
            2'b00:
                if(din) //若push button 為共陰極用~din，共陽用din
                    ss<=2'b01;
            2'b01:
                ss<=2'b10;
            2'b10:
                if(~din) //若push button 為共陰極用din，共陽用~din
                    ss<=2'b00;
            default:
                ss<=2'b00;
        endcase
    end

    assign dout=ss[0];

endmodule
```

## Lab9\_3.v

```
module lab9_3(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 logic
    always@(posedge clk)
    begin
        if(reset)
            R <= 8'b00000001;
        else
            R <= R_t;
    end

    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
            R_t <= R;
    end

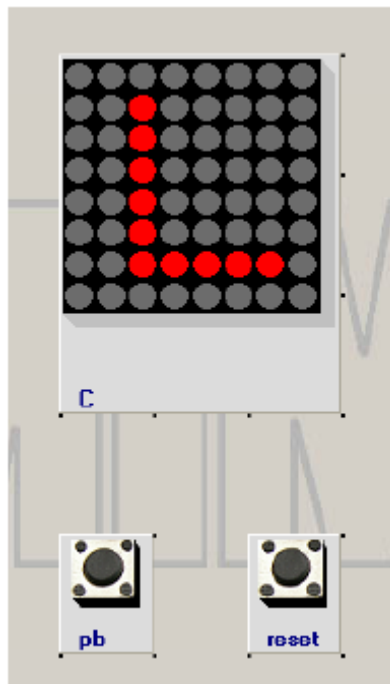
end

// 8x8 LED column control logic
/*
.
.
.
請同學完成column control logic 的程式碼
.
.
*/
endmodule
```

## Lab 9-4 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。)

1. 請在”`c:\logiclab\<你自己的學號>`”的路徑下新增一資料夾 `lab9_4`
2. 開啟Quartus II，設定專案包含下列Verilog 檔：
  - `lab9_4.v`
  - `oneshot.v`
3. 完成設計後請download 至VeriLite 版的EP1C6Q240C8 Device 來驗證結果。





## Lab9\_4.v

```
module lab9_4(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; // 'L' 'O' 'G' 'I' 'C'
    reg [7:0]      R;
    reg [7:0]      C;
    reg [2:0]      counter;
    reg [2:0]      display;
    wire           pb_oneshot;
    wire           reset_oneshot;

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

    always @(posedge clk or posedge reset_oneshot)
        if (reset_oneshot)
            counter <= 3'd0;
        else
            counter <= counter + 3'd1;

    // 8x8 LED row control logic

    // 'L'
    always @(counter)
    begin
        case(counter)
            3'd0:
                R1 = 8'b00000000;
            3'd1:
                R1 = 8'b01000000;
            3'd2:
                R1 = 8'b01000000;
            3'd3:
                R1 = 8'b01000000;
            3'd4:
                R1 = 8'b01000000;
            3'd5:
                R1 = 8'b01111110;
            3'd6:
                R1 = 8'b00000000;
            3'd7:
                R1 = 8'b00000000;
        endcase
    end

end
```

```

//O'
always @(counter)
begin
    case(counter)
    3'd0:
        R2 = 8'b00000000;
    3'd1:
        R2 = 8'b00111100;
    3'd2:
        R2 = 8'b01000010;
    3'd3:
        R2 = 8'b01000010;
    3'd4:
        R2 = 8'b01000010;
    3'd5:
        R2 = 8'b01000010;
    3'd6:
        R2 = 8'b00111100;
    3'd7:
        R2 = 8'b00000000;
    endcase
end

//G'
always @(counter)
begin
    .
    ./*請同學完成'G', 'T', 'C'的row control logic*/
    .
end

//T'

//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;
end

```

```
        default: R = R1;
        endcase
    end

//8x8 LED coulumn control logic
always@(posedge clk)
begin
    if(rst)
        C <= 8'b10000000;
    else
    begin
        C[6:0] <= C[7:1];
        C[7] <= C[0];
    end
end

endmodule
```

## 實習報告

### ■ 實驗原理及結果

### ■ 問題

為oneshot.v 建立一個專案，並建立一個.vwf 檔編輯oneshot 輸入波形，輸入波形請模擬使用者按下push button 再放開的情況，結報請附上輸出波形圖，並說明reg [1:0] ss 的值在波型圖上不同時間點的變化為何。

### ■ 問題心得