

數位邏輯設計實驗結報

- 實驗班別：☐資工(星期三)
- 實驗名稱：(09) 編碼與解碼器
- 實驗日期：2022 年 05 月 29 日
- 組別編號：20
- 成員名單：

(學號)	<u>410921202</u>	(姓名)	<u>林芷萱</u>
	<u>410921203</u>		<u>林芯卉</u>

一、實驗目的

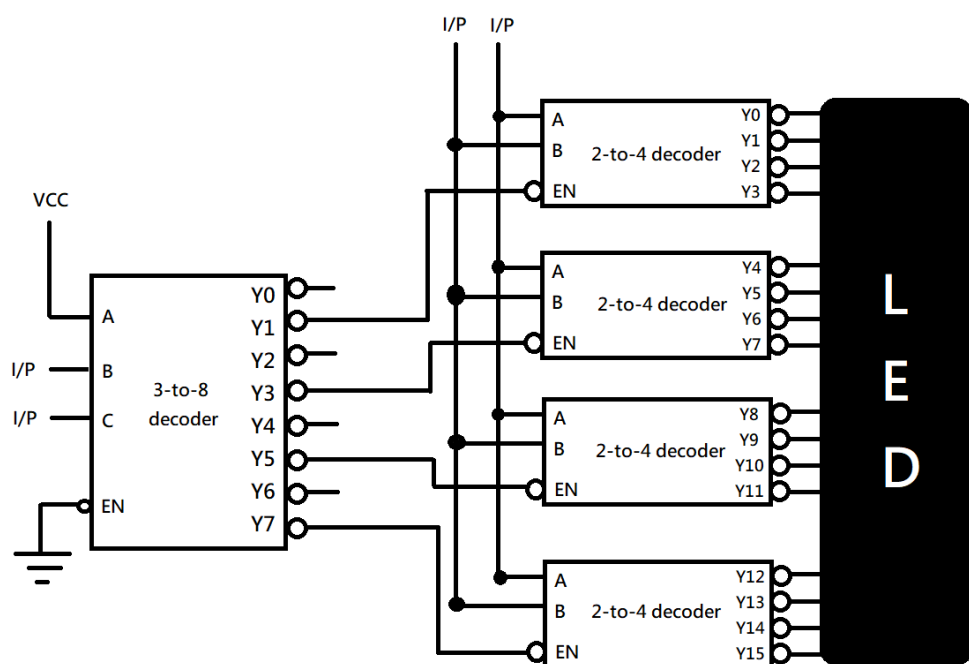
以 Verilog HDL 硬體描述語言，在 DE2-115 實驗平台上實作「實驗三」編碼器與解碼器電路。

二、實驗題目

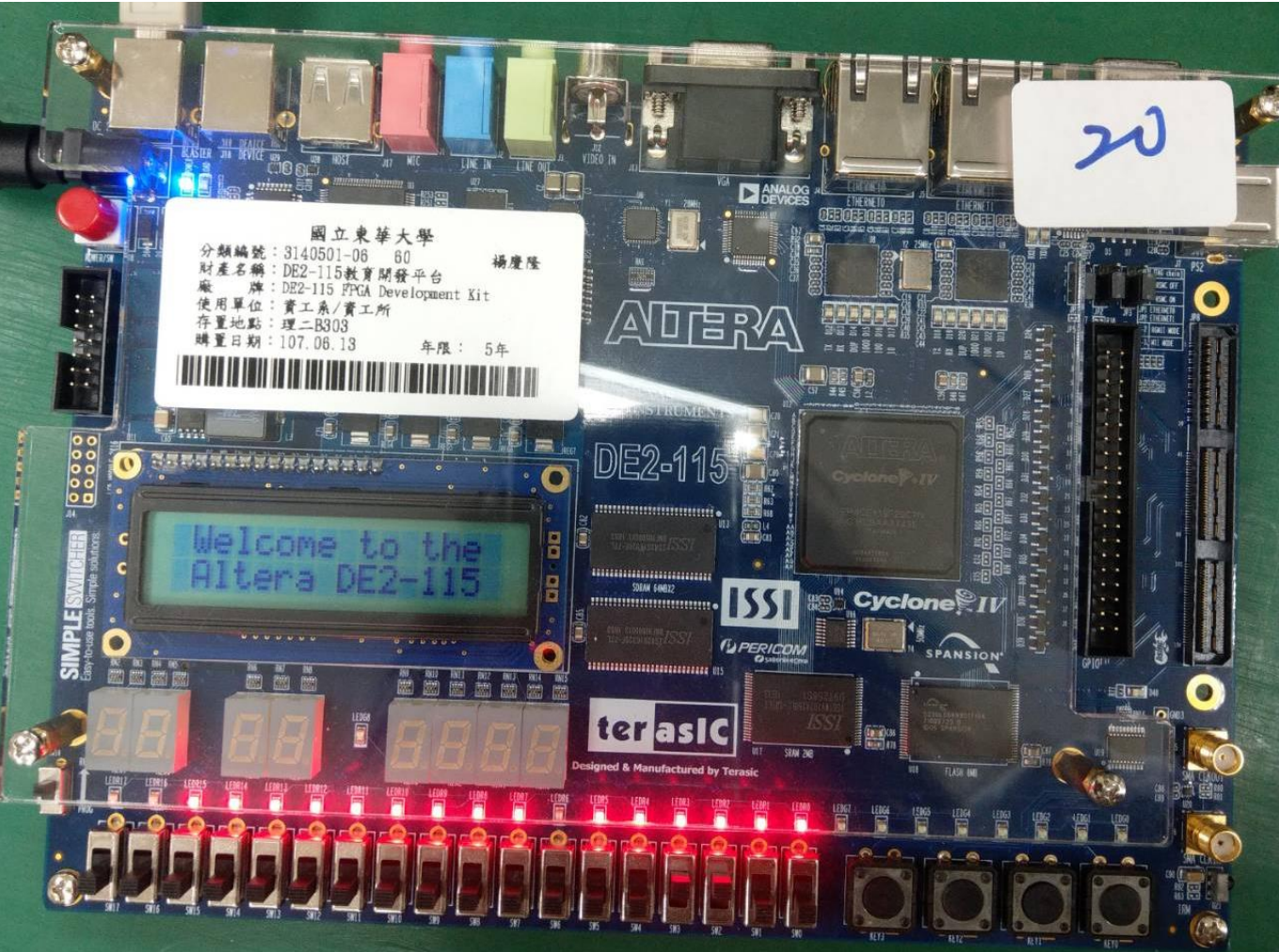
請使用 Verilog 和 DE2-115 實驗 4-to-16 解碼器。在程式中，您要先完成兩個模組：一個 2-to-4 解碼器、和一個 3-to-8 解碼器。然後，使用一個 3-to-8 解碼器和四個 2-to-4 解碼器來實現一個 4-to-16 解碼器。請使用 16 個紅色 LED 作為其輸出，並使用 4 個滑動開關作為其輸入。提示：使用一個 3-to-8 解碼器作為一個 2-to-4 解碼器。

因為我們這組需要將 3-to-8 解碼器的輸入端 A 作為 1，故輸出部分皆至 2-to-4 解碼器的角位應分別為 001(1)、011(3)、101(5)、111(7)。

三、實驗電路圖



四、實驗結果



五、問題與討論

1. 請列出本次實驗的完整程式碼。

```
1 module dec3to8(input [2:0]in, input en, output reg [7:0]y);
2     always @(in)
3     begin
4         if(~en)
5         begin
6             case(in)
7                 3'b000: y=8'b11111110; // 0
8                 3'b001: y=8'b11111101; // 1
9                 3'b010: y=8'b11111011; // 2
10                3'b011: y=8'b11110111; // 3
11                3'b100: y=8'b11101111; // 4
12                3'b101: y=8'b11011111; // 5
13                3'b110: y=8'b10111111; // 6
14                3'b111: y=8'b01111111; // 7
15            endcase
16        end
17    else
18        y = 8'b11111111;
19    end
20 endmodule
21
22 module dec2to4(input [1:0]in, input en, output reg[3:0] y);
23     always @(in)
24     begin
25         if(~en)
26         begin
27             case(in)
28                 2'b00: y=4'b1110; // 0
29                 2'b01: y=4'b1101; // 1
30                 2'b10: y=4'b1011; // 2
31                 2'b11: y=4'b0111; // 3
32            endcase
33        end
34    else
35        y = 4'b1111;
36    end
37 endmodule
38
39 module test(input [3:0]in, input en, output [15:0]y); //4to16
40     wire [2:0]re;
41     wire [7:0]control;
42     assign re[2]=in[3];
43     assign re[1]=in[2];
44     assign re[0]=1;
45     wire [1:0]re2;
46     assign re2[1]=in[1];
47     assign re2[0]=in[0];
48     dec3to8 a0(re, 0, control);
49     dec2to4 a1(re2, control[1], y[3:0]);
50     dec2to4 a2(re2, control[3], y[7:4]);
51     dec2to4 a3(re2, control[5], y[11:8]);
52     dec2to4 a4(re2, control[7], y[15:12]);
53
54 endmodule
```

2. 編解碼電路若不利用 case 撰寫，還有什麼寫法？請列出程式碼。

```
1 module dec3to8(input [2:0]in, input en, output reg [7:0]y);
2     always @(in)
3     begin
4         if(~en)
5             begin
6                 if(in=3'b000) y=8'b11111110; // 0
7                 if(in=3'b001) y=8'b11111101; // 1
8                 if(in=3'b010) y=8'b11111011; // 2
9                 if(in=3'b011) y=8'b11110111; // 3
10                if(in=3'b100) y=8'b11101111; // 4
11                if(in=3'b101) y=8'b11011111; // 5
12                if(in=3'b110) y=8'b10111111; // 6
13                if(in=3'b111) y=8'b01111111; // 7
14            end
15        else
16            begin
17                y = 8'b11111111;
18            end
19    end
20 endmodule
21
22 module dec2to4(input [1:0]in, input en, output reg[3:0] y);
23     always @(in)
24     begin
25         if(~en)
26             begin
27                 if(in=2'b00) y=4'b1110; // 0
28                 if(in=2'b01) y=4'b1101; // 1
29                 if(in=2'b10) y=4'b1011; // 2
30                 if(in=2'b11) y=4'b0111; // 3
31            end
32        else
33            begin
34                y = 4'b1111;
35            end
36    end
37 endmodule
38
39 module test(input [3:0]in, input en, output [15:0]y); //4to16
40     wire [2:0]re;
41     wire [7:0]control;
42     assign re[2]=in[3];
43     assign re[1]=in[2];
44     assign re[0]=1;
45     wire [1:0]re2;
46     assign re2[1]=in[1];
47     assign re2[0]=in[0];
48     dec3to8 a0(re, 0, control);
49     dec2to4 a1(re2, control[1], y[3:0]);
50     dec2to4 a2(re2, control[3], y[7:4]);
51     dec2to4 a3(re2, control[5], y[11:8]);
52     dec2to4 a4(re2, control[7], y[15:12]);
53
54 endmodule
```

3. 試設計一個 4-to-2 的編碼器和一個 2-to-4 解碼器，並列出程式碼。

```
1 // 4-to2 encoder
2 module enc4to2(input [3:0]in, output reg [1:0]y);
3     always @(in)
4     begin
5         case(in)
6             4'b1110: y=2'b00; // 0
7             4'b1101: y=2'b01; // 1
8             4'b1011: y=2'b10; // 2
9             4'b0111: y=2'b11; // 3
10            default: y=2'b00; // 4
11        endcase
12    end
13 endmodule

14
15 // 2-to-4 decoder
16 module dec2to4(input [1:0]in, output reg[3:0] y);
17     always @(in)
18     begin
19         case(in)
20             2'b00: y=4'b1110; // 0
21             2'b01: y=4'b1101; // 1
22             2'b10: y=4'b1011; // 2
23             2'b11: y=4'b0111; // 3
24            default: y = 4'b1111;
25        endcase
26    end
27 endmodule
```