111-1　535313

進階可程式邏輯系統設計與應用

Advanced Programmable Logic System Design and Application

實驗編號 : LAB 02

實驗名稱 : Time-of-day clock

結報完成日期 : 2022.10.17

姓名 : 王語

系級 : 機械四

學號 : 0811127

一、 實驗目的

本實驗藉由 D latch 接觸時序電路，並藉由多種方法描述計數器，了解不同構造下計數器性質的差異，也嘗試了 Quartus IP Library 的使用，而在最後的 Time-of-day clock 實驗中也練習到多個正緣觸發的使用，並且複習 BCD 碼的進位與換算。

二、 Verilog 程式碼

LAB02 > Lab2_0811127.v

```v
// 0811127  王語
// LAB 2

//top module
module Lab2_0811127 (   input [9:0] SW ,
                        input CLOCK_50 ,
                        output [6:0] HEX0 , HEX1 , HEX2 , HEX3 , HEX4 , HEX5 );

reg [3:0] sec_ten , sec_one , hr_ten , hr_one , min_one , min_ten ; //at t0 = 0000
reg SW8_temp0 , SW8_temp1 ;
integer count = 0;

ssd sec_onedigits ( .Din(sec_one),
                    .Dout(HEX0));
ssd sec_tendigits ( .Din(sec_ten),
                    .Dout(HEX1));
ssd min_onedigits ( .Din(min_one),
                    .Dout(HEX2));
ssd min_tendigits ( .Din(min_ten),
                    .Dout(HEX3));
ssd hr_onedigits (  .Din(hr_one),
                    .Dout(HEX4));
ssd hr_tendigits (  .Din(hr_ten),
                    .Dout(HEX5));

```

```v
// The usage among of resource of ">" is greater than ">="
always @(posedge CLOCK_50) begin
    SW8_temp0 <= SW8_temp1 ;
    SW8_temp1 <= SW[8] ;

    if (SW[8]&&(SW8_temp0 != SW[8])) begin
        if (SW[9]) begin
            hr_one <= SW[3:0];
            hr_ten <= SW[7:4];
        end
        else begin
            min_one <= SW[3:0];
            min_ten <= SW[7:4];
        end
    end

```

```
41
42      if (count == 50000000) begin
43          count <= 0;
44
45          sec_one <= sec_one + 4'b0001 ;
46          if (sec_one >= 4'd9 ) begin
47              sec_one <= 4'b0000 ;
48              sec_ten <= sec_ten + 4'b0001;
49
50              if (sec_ten >= 4'd5 ) begin
51                  sec_ten <= 4'b0000 ;
52                  min_one <= min_one + 4'b0001;
53
54                  if (min_one >= 4'd9 ) begin
55                      min_one <= 4'b0000 ;
56                      min_ten <= min_ten + 4'b0001;
57
58                      if (min_ten >= 4'd5 ) begin
59                          min_ten <= 4'b0000 ;
60                          hr_one <= hr_one + 4'b0001;
61
62                          if (hr_one >= 4'd9 ) begin
63                              hr_one <= 4'b0000 ;
64                              hr_ten <= hr_ten + 4'b0001 ;
65
66                          end
67
68                          if ( (hr_one >= 4'd3) && (hr_ten >= 4'd1) ) begin
69                                  hr_one <= 4'b0000 ;
70                                  hr_ten <= 4'b0000 ;
71                              end
72                      end
73                  end
74              end
75          end
76      end
77
78      else begin
79          count <= count+1;
80      end
81  end
82
83  endmodule
```

```
84
85  // sub module
86  //seven segment display
87  module ssd (    input [3:0] Din,
88                  output [6:0] Dout );
89  assign Dout[0] =    ((!Din[3])&(!Din[2])&(!Din[1])&( Din[0]))|
90                      ((!Din[3])&( Din[2])&(!Din[1])&(!Din[0]))|
91                      (( Din[3])&( Din[2])&(!Din[1])&( Din[0]))|
92                      (( Din[3])&(!Din[2])&( Din[1])&( Din[0]));
93
94  assign Dout[1] =    ((!Din[3])&( Din[2])&(!Din[1])&( Din[0]))|
95                      (          ( Din[2])&( Din[1])&(!Din[0]))|
96                      (( Din[3])&          ( Din[1])&( Din[0]))|
97                      (( Din[3])&( Din[2])&          (!Din[0]));
98
99  assign Dout[2] =    ((!Din[3])&(!Din[2])&( Din[1])&(!Din[0]))|
100                     (( Din[3])&( Din[2])&( Din[1])          )|
101                     (( Din[3])&( Din[2])&          (!Din[0]));
102
103
104 assign Dout[3] =    ((!Din[3])&(!Din[2])&(!Din[1])&( Din[0]))|
105                     ((!Din[3])&( Din[2])&(!Din[1])&(!Din[0]))|
106                     (( Din[3])&(!Din[2])&( Din[1])&(!Din[0]))|
107                     (          ( Din[2])&( Din[1])&( Din[0]));
108
109 assign Dout[4] =    ((!Din[3])&                  &( Din[0]))|
110                     (          (!Din[2])&(!Din[1])&( Din[0]))|
111                     ((!Din[3])&( Din[2])&(!Din[1])          );
112
113
114 assign Dout[5] =    (( Din[3])&( Din[2])&(!Din[1])&( Din[0]))|
115                     ((!Din[3])&(!Din[2])&          ( Din[0]))|
116                     ((!Din[3])&(!Din[2])&( Din[1])          )|
117                     ((!Din[3])&          ( Din[1])&( Din[0]));
118
119 assign Dout[6] =    ((!Din[3])&( Din[2])&( Din[1])&( Din[0]))|
120                     (( Din[3])&( Din[2])&(!Din[1])&(!Din[0]))|
121                     ((!Din[3])&(!Din[2])&(!Din[1])          );
122
123 endmodule
124
```

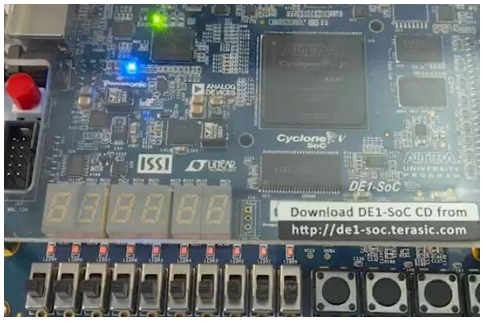
下面這些沒用到

```
124
125 // module bin2bcd( input [13:0] bin ,
126 //                 output reg [15:0] bcd );
127
128 // integer i;
129
130 // always @(bin) begin
131 //     bcd=0;
132 //     for (i=0;i<14;i=i+1) begin                  //Iterate once for each bit in input number
133 //     if (bcd[3:0] >= 5) bcd[3:0] = bcd[3:0] + 3;      //If any BCD digit is >= 5, add three
134 //  if (bcd[7:4] >= 5) bcd[7:4] = bcd[7:4] + 3;
135 //  if (bcd[11:8] >= 5) bcd[11:8] = bcd[11:8] + 3;
136 //  if (bcd[15:12] >= 5) bcd[15:12] = bcd[15:12] + 3;
137 //  bcd = {bcd[14:0],bin[13-i]};               //Shift one bit, and shift in proper bit from input
138 //     end
139 // end
140 // endmodule
```

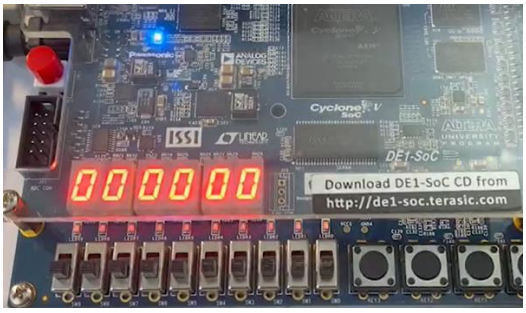
三、　實驗結果

由於這次的實驗結果是動態的，因此附上影片連結:
https://pse.is/4jnny7

燒錄前(亮綠燈)

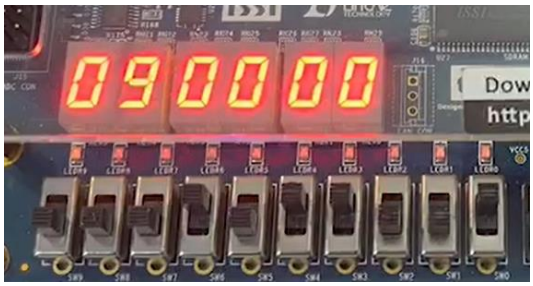


剛燒錄完成(起始狀態)



00:24:59 >>　00:25:00

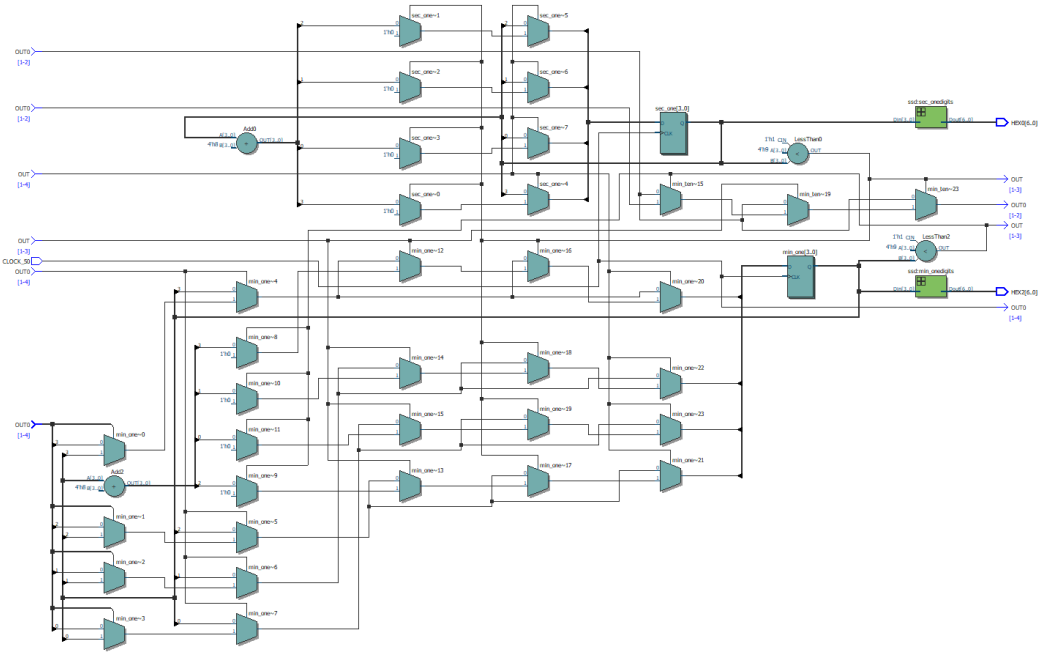
08:59:59>>　09:00:00



24:59:59>>　00:00:00

四、 RTL

PDF 檔: https://pse.is/4jnny7



五、 問題與討論

1. 整個實驗困惑最久的地方，就是如何設計 SW8 的正緣觸發，使得 SW8 變為 1 的時候能夠設定時和分的數值，因為這項功能比須和 always @(posedge clk)包在一起，否則會出現重複賦值的情況，但是又沒有 if (posedge SW[8])這種語法，於是我宣告另外兩個變數 SW8_temp0 和 SW8_temp1，利用 Flip-Flop 延遲一個 CLK，即可以 SW[8]&&(SW[8]!=SW8_temp0)完成正緣觸發判斷。

2. 另外就是在 Always 裡面不能接 submodule，這次實驗犯了這個毛病，導致鬼打牆，一直抓不出 bug。

3. 最後是 wire 和 reg 的一些規則，像是 assign 等號左邊一定要是 wire，在透過網路資源複習語法後就解決這些問題了。

六、 心得

因為沒有仔細把題目看清楚，導致繞了不少路，一直以為要用 switch 以二進制的方式輸入數值，花了很多時間設計 binary to BDC 的電路，結果最後才發現用不到。但實驗過程中獲得相當豐富的成果，也讓我成就感滿滿，雖然因為確診沒有如期 DEMO，但最後還是順利完成實驗，也謝謝助教細心教學與批閱。