

EECS 2070 02 Digital Design Labs 2019

Lab 4

學號：104021215 姓名：熊磊

1. 實作過程

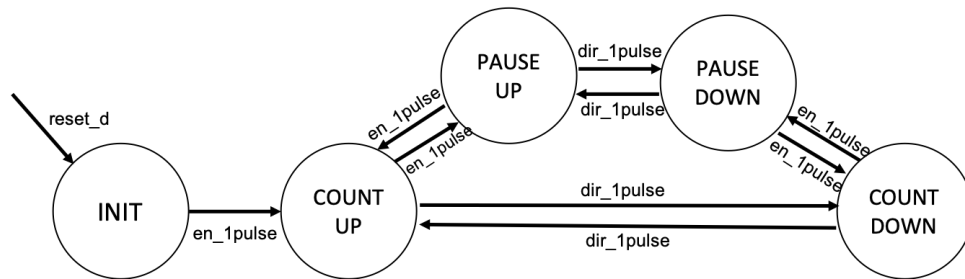
[Lab4_1]

- 第一個 Lab 是要做用 SW[15:0]來做 Binary input，並把結果顯示在 7-segment 顯示器上。要注意的是因為要讓顯示器更新頻率夠快，所以在 clock_divider，要用 2^{13} 來除頻。
- 另外，因為這次要用到 button，而 button 因為結構上的設計，在按下去之後需要一段時間來達到穩定，因此為了要讓 reset 按下去的時候，可以一直維持在 1 的 value，所以在 reset 的地方，要用 debounce 來讓訊號變穩定。
- Display 的顯示方法就是一次控制一個 Digit，因此在每個 clk_d 的 posedge 時，就會把 value 送到顯示器上，並在下一個 clock 跳到下一個 digit。因為我們的頻率夠快，所以看起來四個 digit 都是同時在控制的。

```
31 wire clk_d, reset_d;
32 reg [3:0] value, DIGIT;
33 wire [3:0] BCD0, BCD1, BCD2, BCD3;
34
35 clock_divider #(.n(13)) c1(.clk(clk), .clk_div(clk_d));
36 debounce dreset(.pb_debounced(reset_d), .pb(reset), .clk(clk_d));
37
38 assign BCD0 = (reset_d == 1) ? 4'b0 : (SW[3:0] > 4'd9) ? 4'd9 : SW[3:0];
39 assign BCD1 = (reset_d == 1) ? 4'b0 : (SW[7:4] > 4'd9) ? 4'd9 : SW[7:4];
40 assign BCD2 = (reset_d == 1) ? 4'b0 : (SW[11:8] > 4'd9) ? 4'd9 : SW[11:8];
41 assign BCD3 = (reset_d == 1) ? 4'b0 : (SW[15:12] > 4'd9) ? 4'd9 : SW[15:12];
42 assign DISPLAY = (value == 4'd0) ? 7'b0000001 :
43                  (value == 4'd1) ? 7'b1001111 :
44                  (value == 4'd2) ? 7'b0010010 :
45                  (value == 4'd3) ? 7'b0000110 :
46                  (value == 4'd4) ? 7'b1001100 :
47                  (value == 4'd5) ? 7'b0100100 :
48                  (value == 4'd6) ? 7'b0100000 :
49                  (value == 4'd7) ? 7'b0001111 :
50                  (value == 4'd8) ? 7'b0000000 :
51                  (value == 4'd9) ? 7'b0000100 : 7'b0000001;
52
53 always @(posedge clk_d) begin
54     case (DIGIT)
55     4'b1110: begin
56         value = BCD1;
57         DIGIT = 4'b1101;
58     end
59     4'b1101: begin
60         value = BCD2;
61         DIGIT = 4'b1011;
62     end
63     4'b1011: begin
64         value = BCD3;
65         DIGIT = 4'b0111;
66     end
67     4'b0111: begin
68         value = BCD0;
69         DIGIT = 4'b1110;
70     end
71     default: begin
72         value = BCD0;
73         DIGIT = 4'b1110;
74     end
75     endcase
76 end
```

[Lab4_2]

- 第二個 lab 是要做 BCD counter，然後夠過一些 button 控制往上數或往下數、暫停、重設，並可以做 record 的功能。因此透過 FSM 來做會比較方便，而我的 state 設計如下幾個。



透過在哪個 state，來傳送 direction 和 enable 的訊號

先設計 1 digit BCD counter 的 module

```

93  always@(*)begin
94      case(state)
95          `INIT: begin
96              en_counter1 = 0;
97              dir_now1 = 1;
98          end
99          `COUNTUP: begin
100             en_counter1 = 1;
101             dir_now1 = 1;
102          end
103          `COUNTDOWN: begin
104             en_counter1 = 1;
105             dir_now1 = 0;
106          end
107          `PAUSEUP: begin
108             en_counter1 = 0;
109             dir_now1 = 1;
110          end
111          `PAUSEDOWN: begin
112             en_counter1 = 0;
113             dir_now1 = 0;
114          end
115          default: begin
116             en_counter1 = 0;
117             dir_now1 = 1;
118          end
119      endcase
120  end
  
```

```

180  module OneDigitBCDCounter(clk, reset, en, dir, BCD, cout);
181      input clk, reset, en, dir;
182      output [3:0]BCD;
183      output reg cout;
184      reg [3:0]BCD, outputs;
185
186      always @* begin
187          if(en==0) begin
188              outputs <= BCD;
189              cout <= 0;
190          end
191          else if(en==1 && dir==1) begin
192              if(BCD==4'b0000 && BCD<=4'b1000) begin
193                  {cout, outputs} = BCD + 4'b0001;
194              end
195              else if(BCD==4'b1001) begin
196                  outputs <= 4'b0000;
197                  cout <= 1;
198              end
199              else begin
200                  outputs <= 4'b0000;
201                  cout <= 0;
202              end
203          end
204          else if(en==1 && dir==0) begin
205              if(BCD==4'b0001 && BCD<=4'b1001) begin
206                  outputs <= BCD - 4'b0001;
207                  cout <= 0;
208              end
209              else if(BCD==4'b0000) begin
210                  outputs <= 4'b1001;
211                  cout <= 1;
212              end
213              else begin
214                  outputs <= 4'b0000;
215                  cout <= 0;
216              end
217          end
218          else begin
219              outputs <= 4'bxxxx;
220              cout <= 1'bx;
221          end
222      end
223
224      always @(negedge clk, posedge reset)
225      begin
226          if(reset==1) BCD <= 4'b0000;
227          else BCD <= outputs;
228      end
229  endmodule
  
```

- 接著把兩個 digit 合起來變成 2 digits BCD counter。

```

169  module TwoDigitBCDCounter(clk, reset, en, dir, BCD0, BCD1, cout);
170      input clk, reset, en, dir;
171      output [3:0] BCD0, BCD1;
172      output cout;
173
174      wire carry0;
175
176      OneDigitBCDCounter c0(.clk(clk), .reset(reset), .en(en), .dir(dir), .BCD(BCD0), .cout(cout));
177      OneDigitBCDCounter c1(.clk(clk), .reset(reset), .en(carry0), .dir(dir), .BCD(BCD1), .cout(cout));
178  endmodule
  
```

- 紀錄 counter 的部分，就是先用 record_1pulse 來讓連續的訊後，只產生一個 cycle 的訊號，在這個 cycle 用 posedge 來把 BCD0, BCD1 的訊號送到 BCD2, BCD3。

```

132     always@(posedge record_1pulse, posedge reset_d) begin
133         if(reset_d == 1) begin
134             BCD2 = 4'b0;
135             BCD3 = 4'b0;
136         end
137         else begin
138             BCD2 = BCD0;
139             BCD3 = BCD1;
140         end
141     end

```

- 在 count 的過程中，如果到往上數到 max(99)，或往下數到 min(00)，這時要把 max 和 min 的 LED 亮起來。

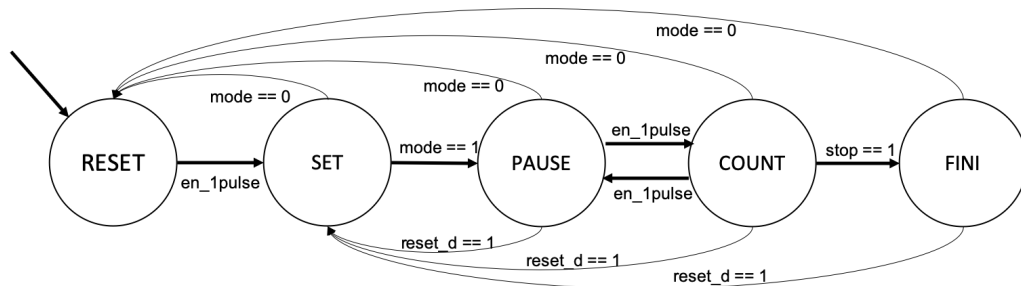
```

69     assign max = (state != `INIT && BCD0==9 && BCD1==9 && dir_now == 1) ? 1 : 0;
70     assign min = (state != `INIT && BCD0==0 && BCD1==0 && dir_now == 0) ? 1 : 0;

```

[Lab4_3]

- 這個 Lab 是要做一個倒數計時器，有分兩個 mode，一個是 setting，一個是 counting。



- 設定時間的部分，我是利用 min_plus 和 sec_plus 按下去的時候，當作 Time counter 的 enable 訊號。

```

69     assign en_min = (out1 == 1) ? out1 : (state == `SET) ? min_plus_1pulse : 0;
70     assign en_sec = (state == `SET) ? sec_plus_1pulse : en_counter1;

```

- 而 min 和 sec 的增加和減少，是透過 state 來控制，如果是在 setting mode，就往上加，如果是其他 mode 就往下減。

```

67     assign min_dir = (state == `SET) ? (out1 == 1) ? 1 : min_plus_1pulse : 0;
68     assign sec_dir = (state == `SET) ? sec_plus_1pulse : 0;

```

2. 學到的東西與遇到的困難

這次的 Lab 還蠻有趣的，因為可以做出一個計時器。可惜最後還是沒做出完整一秒的 counter，但是大概的想法是用一個 count 來數到 10^8 ，不過最後跑起來有 bug，所以最後沒有成功 demo。

3. 想對老師或助教說的話

謝謝助教的幫忙☺