

第五次實驗報告

題目:5-1~5-4

姓名:羅名志

學號:0813228

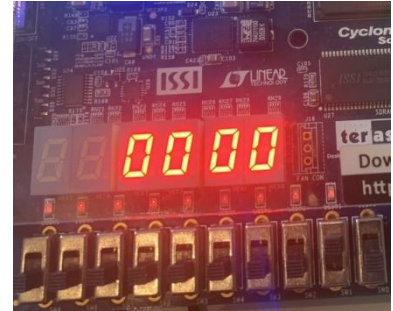
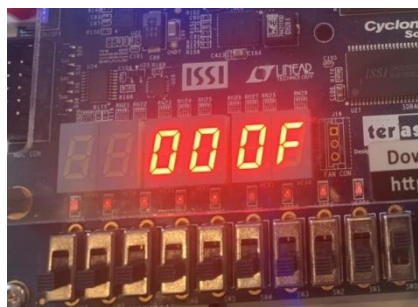
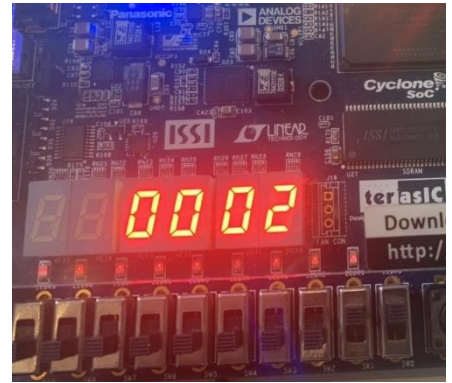
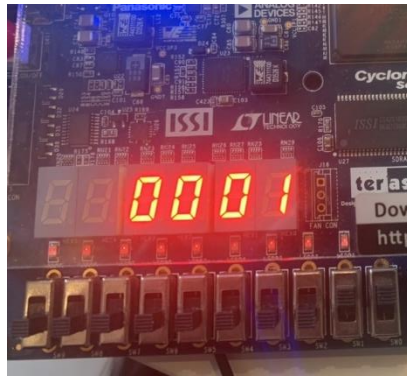
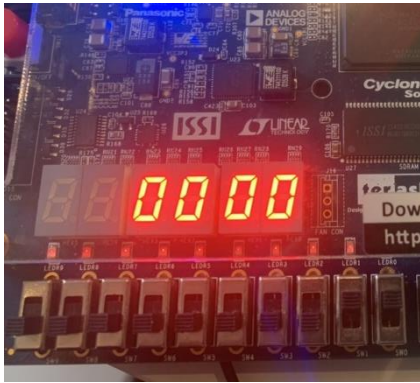
繳交日期:2022/4/4

一、5-1

(1)實驗程式碼:

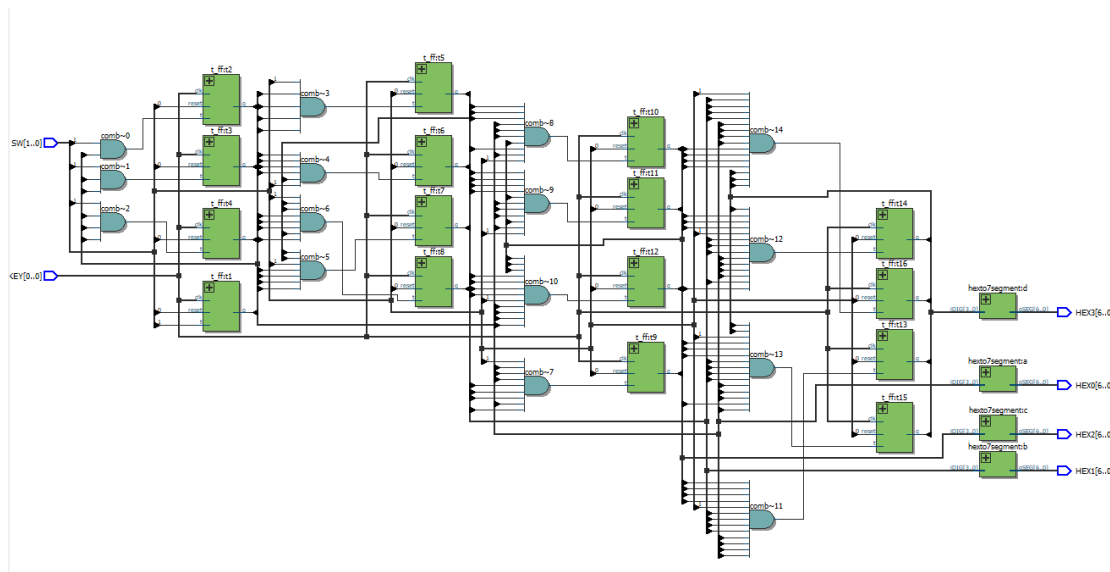
```
1 module labfive1(KEY,SW,HEX3,HEX2,HEX1,HEX0);
2   input [0:0]KEY;
3   input [1:0]SW;
4   wire [15:0]Q;
5   wire [27:0]w;
6   output [6:0]HEX3,HEX2,HEX1,HEX0;
7
8   t_ff t1(SW[1],KEY[0],SW[0],Q[0]);
9   t_ff t2((SW[1] & Q[0]),KEY[0],SW[0],Q[1]);
10  t_ff t3((SW[1] & Q[1] & Q[0]),KEY[0],SW[0],Q[2]);
11  t_ff t4((SW[1] & Q[1] & Q[0] & Q[2]),KEY[0],SW[0],Q[3]);
12  t_ff t5((SW[1] & Q[1] & Q[0] & Q[2] & Q[3]),KEY[0],SW[0],Q[4]);
13  t_ff t6((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4]),KEY[0],SW[0],Q[5]);
14  t_ff t7((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5]),KEY[0],SW[0],Q[6]);
15  t_ff t8((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5] & Q[6]),KEY[0],SW[0],Q[7]);
16  t_ff t9((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5] & Q[6] & Q[7]),KEY[0],SW[0],Q[8]);
17  t_ff t10((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5] & Q[6] & Q[7] & Q[8]),KEY[0],SW[0],Q[9]);
18  t_ff t11((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5] & Q[6] & Q[7] & Q[8] & Q[9]),KEY[0],SW[0],Q[10]);
19  t_ff t12((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5] & Q[6] & Q[7] & Q[8] & Q[9] & Q[10]),KEY[0],SW[0],Q[11]);
20  t_ff t13((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5] & Q[6] & Q[7] & Q[8] & Q[9] & Q[10] & Q[11]),KEY[0],SW[0],Q[12]);
21  t_ff t14((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5] & Q[6] & Q[7] & Q[8] & Q[9] & Q[10] & Q[11] & Q[12]),KEY[0],SW[0],Q[13]);
22  t_ff t15((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5] & Q[6] & Q[7] & Q[8] & Q[9] & Q[10] & Q[11] & Q[12] & Q[13]),KEY[0],SW[0],Q[14]);
23  t_ff t16((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5] & Q[6] & Q[7] & Q[8] & Q[9] & Q[10] & Q[11] & Q[12] & Q[13] & Q[14]),KEY[0],SW[0],Q[15]);
24
25
26  hexto7segment a(Q[3:0],HEX0);
27  hexto7segment b(Q[7:4],HEX1);
28  hexto7segment c(Q[11:8],HEX2);
29  hexto7segment d(Q[15:12],HEX3);
30
31
32 endmodule
33
34
35 module t_ff(t, clk, reset, q);
36
37   input t;
38   input clk;
39   input reset;
40   wire w;
41   output reg q;
42
43
44
45   always@(posedge clk)
46   begin
47
48     if (reset==0)
49       q=0;
50   else begin
51
52     if(t==0)
53       q=q;
54     else
55       q=~q;
56
57   end
58 end
59 endmodule
60
61 module hexto7segment (
62   input [3:0] iDIG,
63   output reg [6:0] oSEG
64 );
65
66 always@(iDIG) begin
67   case(iDIG)
68     4'h1: oSEG = 7'b1111001;
69     4'h2: oSEG = 7'b0100100; // ---t---
70     4'h3: oSEG = 7'b0110000; // lt   rt
71     4'h4: oSEG = 7'b0011001; // |   |
72     4'h5: oSEG = 7'b0010010; // ---m---
73     4'h6: oSEG = 7'b0000010; // |   |
74     4'h7: oSEG = 7'b1111000; // lb   rb
75     4'h8: oSEG = 7'b0000000; // |   |
76     4'h9: oSEG = 7'b0011000; // ---b---
77     4'ha: oSEG = 7'b0001000;
78     4'hb: oSEG = 7'b0000011;
79     4'hc: oSEG = 7'b1000110;
80     4'hd: oSEG = 7'b0100001;
81     4'he: oSEG = 7'b0000110;
82     4'hf: oSEG = 7'b0001110;
83     4'h0: oSEG = 7'b1000000;
84   endcase
85 end
86
87 endmodule
```

(2)實驗結果:



//中下:enable=0; 右下:reset=0;

(3)RTL 布局:



(4)問題與討論:

一開始遇到超多問題，像是 pin 的 TFF 保留字不能用，還有轉 HEX 時一直出錯，導致這次實驗做很久最後沒做完，不過更加深了 T flip-flop 的運用，同時也了解到 enable 跟 reset 怎麼控制 output，算是學到非常多。

二、5-2

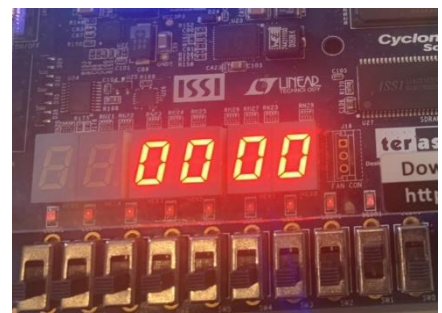
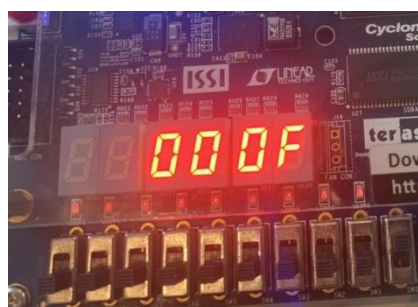
(1)實驗程式碼:

```

1  module labfive2(KEY,SW,HEX3,HEX2,HEX1,HEX0);
2      input [0:0]KEY;
3      input [1:0]SW;
4      reg [15:0]Q;
5      output [6:0]HEX3,HEX2,HEX1,HEX0;
6
7      always@(posedge KEY[0])
8      if(~SW[0]) Q=0;
9      else begin
10         if(SW[1])
11             Q<=Q+1;
12         end
13         hexto7segment a(Q[3:0],HEX0);
14         hexto7segment b(Q[7:4],HEX1);
15         hexto7segment c(Q[11:8],HEX2);
16         hexto7segment d(Q[15:12],HEX3);
17
18     endmodule
19
20
21 module hexto7segment (
22     input [3:0] iDIG,
23     output reg [6:0] oSEG
24 );
25
26 always@(iDIG) begin
27     case(iDIG)
28         4'h1: oSEG = 7'b1111001;
29         4'h2: oSEG = 7'b0100100; // ---t---
30         4'h3: oSEG = 7'b0110000; // lt   rt
31         4'h4: oSEG = 7'b0011001; // |   |
32         4'h5: oSEG = 7'b0010010; // ---m---
33         4'h6: oSEG = 7'b0000010; // |   |
34         4'h7: oSEG = 7'b1111000; // lb   rb
35
36         4'h8: oSEG = 7'b0000000; // |
37         4'h9: oSEG = 7'b0011000; // ---
38         4'ha: oSEG = 7'b0001000;
39         4'hb: oSEG = 7'b0000011;
40         4'hc: oSEG = 7'b1000110;
41         4'hd: oSEG = 7'b0100001;
42         4'he: oSEG = 7'b0000110;
43         4'hf: oSEG = 7'b0001110;
44         4'h0: oSEG = 7'b1000000;
45     endcase
46 end
47 endmodule

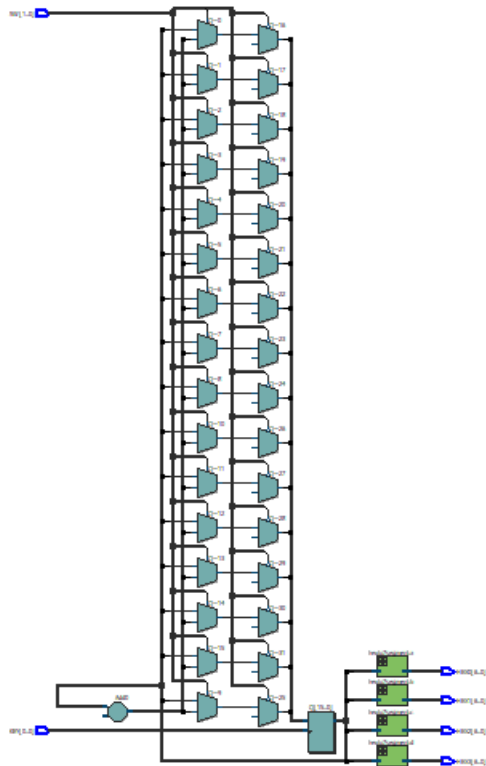
```

(2)實驗結果:



//中下:enable=0; 右下:reset=0;

(3)RTL 布局:



(4)問題與討論:

延續上題的題目，利用 `always` 的做法，搭配 `if / else`，將 `reset` 跟 `enable` 考慮進去，此題較輕鬆簡單。

三、5-3

(1)實驗程式碼:

```

1 module labfivee5(KEY,SW,HEX3,HEX2,HEX1,HEX0)
2   input [0:0]KEY;
3   input [1:0]SW;
4   wire [15:0]Q;
5   output [6:0]HEX3,HEX2,HEX1,HEX0;
6
7   ha(SW[1],KEY[0],SW[0],Q);
8
9   hexto7segment e(Q[3:0],HEX0);
10  hexto7segment f(Q[7:4],HEX1);
11  hexto7segment g(Q[11:8],HEX2);
12  hexto7segment h(Q[15:12],HEX3);
13
14 endmodule
15
16 module hexto7segment (
17   input [3:0] iDIG,
18   output reg [6:0] oSEG
19 );
20
21 always@(iDIG) begin
22   case(iDIG)
23     4'h1: oSEG = 7'b1111001; // ---t---
24     4'h2: oSEG = 7'b0100100; // lt rt
25     4'h3: oSEG = 7'b0110000; // | |
26     4'h4: oSEG = 7'b0011001; // | |
27     4'h5: oSEG = 7'b0010010; // ---m---
28     4'h6: oSEG = 7'b0000010; // | |
29     4'h7: oSEG = 7'b1111000; // lb rb
30     4'h8: oSEG = 7'b0000000; // | |
31     4'h9: oSEG = 7'b0011000; // ---b---
32     4'ha: oSEG = 7'b0001000;
33     4'hb: oSEG = 7'b0000011;
34     4'hc: oSEG = 7'b1000110;
35     4'hd: oSEG = 7'b0100001;
36     4'he: oSEG = 7'b0000110;
37     4'hf: oSEG = 7'b0001110;
38     4'h0: oSEG = 7'b1000000;
39   endcase
40 end
41
42 endmodule

```

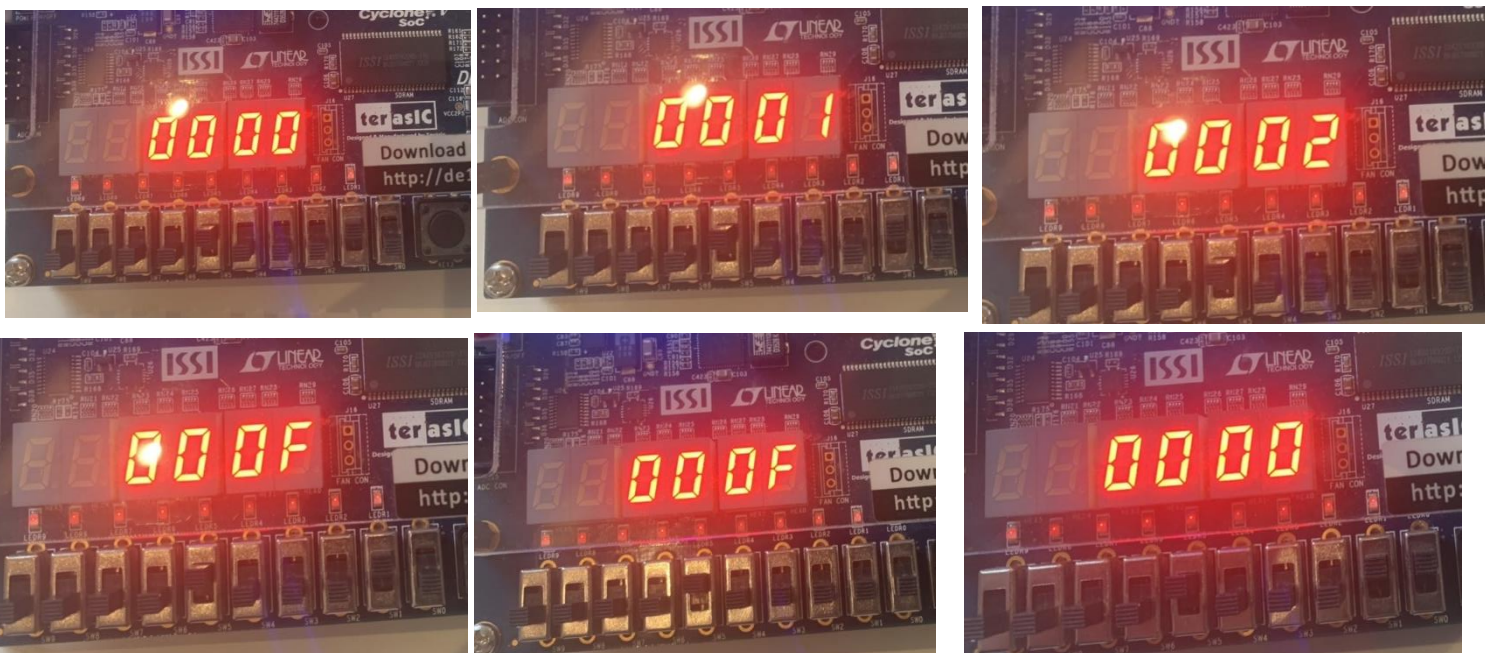
```

37 `timescale 1 ps / 1 ps
38 // synopsys translate_on
39 module ha (
40   clk_en,
41   clock,
42   sclr,
43   q);
44
45   input clk_en;
46   input clock;
47   input sclr;
48   output [15:0] q;
49
50   wire [15:0] sub_wire0;
51   wire [15:0] q = sub_wire0[15:0];
52
53   lpm_counter LPM_COUNTER_component (
54     .clk_en (clk_en),
55     .clock (clock),
56     .sclr (sclr),
57     .q (sub_wire0),
58     .aclr (1'b0),
59     .aload (1'b0),
60     .aset (1'b0),
61     .cin (1'b1),
62     .cnt_en (1'b1),
63     .cout (),
64     .data ({16{1'b0}}),
65     .eq (),
66     .sload (1'b0),
67     .sset (1'b0),
68     .updown (1'b1));
69
70   defparam
71     LPM_COUNTER_component.lpm_direction = "UP",
72     LPM_COUNTER_component.lpm_port_updown = "PORT_UNUSED",
73     LPM_COUNTER_component.lpm_type = "LPM_COUNTER",
74     LPM_COUNTER_component.lpm_width = 16;
75
76 endmodule

```

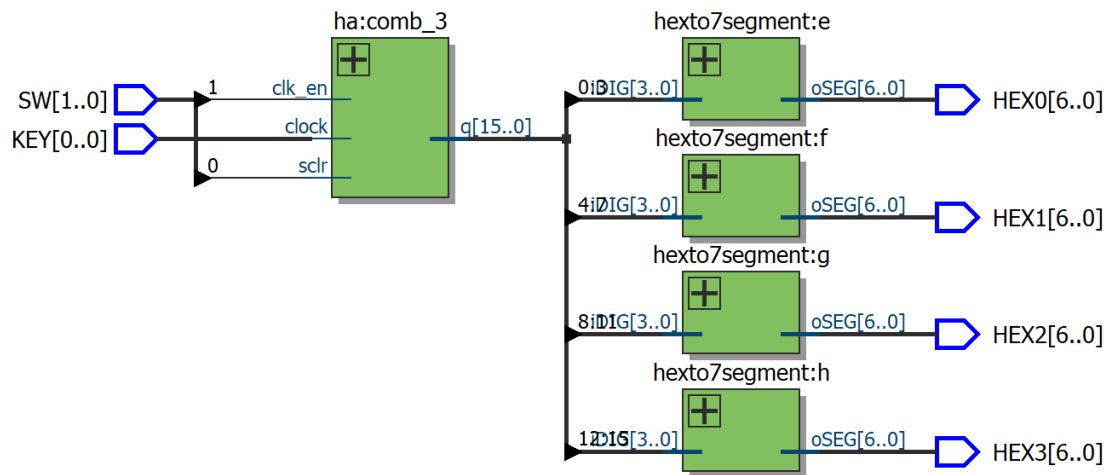
//右:Megawizard_counter

(2)實驗結果:



//中下:enable=0; 右下:reset=1;

(3)RTL 布局:



(4)問題與討論:

第一次使用 mega wizard，利用內建的 counter，設定好 reset 跟 enable 的參數，即可輕鬆解決前幾 part 的內容。

四、5-4

(1)實驗程式碼:

```

1 module labfive4(CLOCK_50,HEX0);
2     input CLOCK_50;
3     wire [3:0]Q;
4     output [6:0]HEX0;
5     reg [3:0]a;
6     reg [30:0]count;
7     reg temp;
8
9     always@(posedge CLOCK_50) begin
10
11         if(count==5000000) begin
12             count=0;
13             temp=1;
14             if(Q==4'h9)
15                 a<=1;
16             else
17                 a<=0;
18             end
19         else begin
20             count=count+1;
21             temp=0;
22         end
23     end
24
25     test3(temp,a,Q);
26
27     hexto7segment e(Q[3:0],HEX0);
28
29 endmodule
30
31 module hexto7segment (
32     input [3:0] iDIG,
33     output reg [6:0] oSEG
34 );
35
36
37 always@(iDIG) begin
38     case(iDIG)
39         4'h1: oSEG = 7'b11111001;
40         4'h2: oSEG = 7'b0100100; /.
41         4'h3: oSEG = 7'b0110000;
42         4'h4: oSEG = 7'b0011001;
43         4'h5: oSEG = 7'b0010010;
44         4'h6: oSEG = 7'b0000010;
45         4'h7: oSEG = 7'b1111000;
46         4'h8: oSEG = 7'b0000000;
47         4'h9: oSEG = 7'b0011000;
48         4'ha: oSEG = 7'b0001000;
49         4'hb: oSEG = 7'b0000011;
50         4'hc: oSEG = 7'b1000110;
51         4'hd: oSEG = 7'b0100001;
52         4'he: oSEG = 7'b0000110;
53         4'hf: oSEG = 7'b0001110;
54         4'h0: oSEG = 7'b1000000;
55     endcase
56 end
57
58 endmodule

```

```

37 `timescale 1 ps / 1 ps
38 // synopsys translate_on
39 module test3 (
40     clock,
41     sclr,
42     q);
43
44     input clock;
45     input sclr;
46     output [3:0] q;
47
48     wire [3:0] sub_wire0;
49     wire [3:0] q = sub_wire0[3:0];
50
51     lpm_counter LPM_COUNTER_component (
52         .clock (clock),
53         .sclr (sclr),
54         .q (sub_wire0),
55         .aclr (1'b0),
56         .aload (1'b0),
57         .aset (1'b0),
58         .cin (1'b1),
59         .clk_en (1'b1),
60         .cnt_en (1'b1),
61         .cout (),
62         .data ({4{1'b0}}),
63         .eq (),
64         .load (1'b0),
65         .sset (1'b0),
66         .updown (1'b1));
67
68     defparam
69         LPM_COUNTER_component.lpm_direction = "UP",
70         LPM_COUNTER_component.lpm_port_updown = "PORT_UNUSED",
71         LPM_COUNTER_component.lpm_type = "LPM_COUNTER",
72         LPM_COUNTER_component.lpm_width = 4;
73
74 endmodule

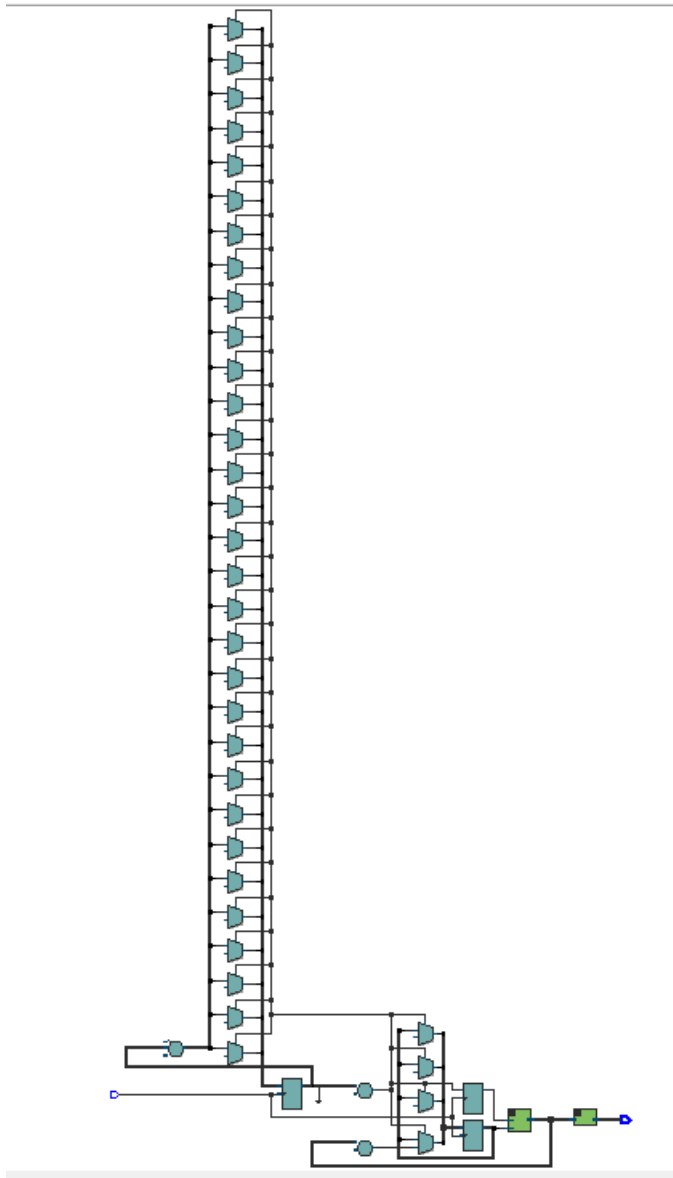
```

//右:Megawizard_counter

(2)實驗結果:



(3)RTL 布局:



(4)問題與討論:

一開始根本不知道 50MHZ 的 clock signal 怎麼變成一秒，最後發現只要設定一參數從 0 加到 50M 就可以判斷一秒的間隔，然後再使用設定的變數抓 clock 的 posedge 瞬間，搭配 mega wizard 的 counter 完成此題，算是淺顯易懂的一題。寫完後發現身邊同學寫的都比我有效率許多，看來我真的要好好努力，不然期中退選一定有我。