第五次實驗報告

題目:5-1~5-4

姓名:羅名志

學號:0813228

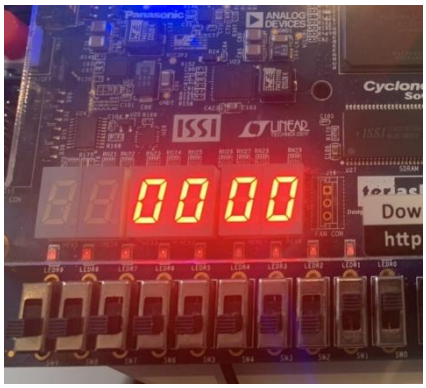
繳交日期:2022/4/4

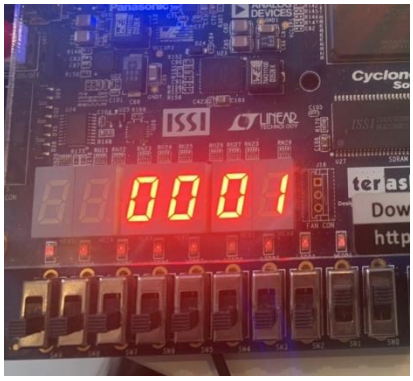
一、5-1

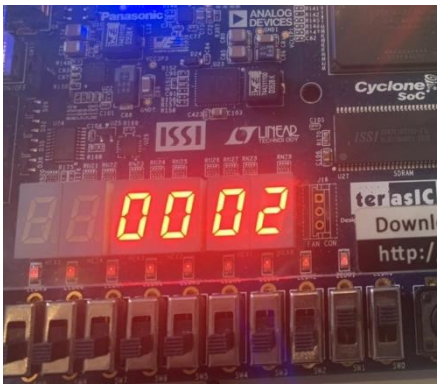
(1)實驗程式碼:

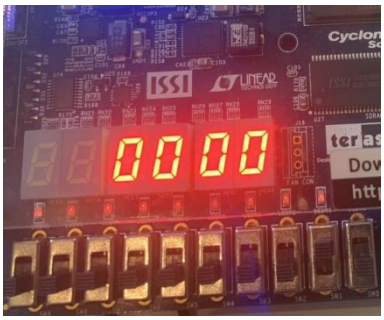
```
module labfive1(KEY,SW,HEX3,HEX2,HEX1,HEX0);
   input [0:0]KEY;
   input [1:0]SW;
   wire [15:0]Q;
   wire [27:0]w;
   output [6:0]HEX3,HEX2,HEX1,HEX0;

   t_ff t1(SW[1],KEY[0],SW[0],Q[0]);
   t_ff t2((SW[1] & Q[0]),KEY[0],SW[0],Q[1]);
   t_ff t3((SW[1] & Q[1] & Q[0]),KEY[0],SW[0],Q[2]);
   t_ff t4((SW[1] & Q[1] & Q[0] & Q[2]),KEY[0],SW[0],Q[3]);
   t_ff t5((SW[1] & Q[1] & Q[0] & Q[2] & Q[3]),KEY[0],SW[0],Q[4]);
   t_ff t6((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4]),KEY[0],SW[0],Q[5]);
   t_ff t7((SW[1] & Q[1] & Q[0] & Q[2] & Q[3] & Q[4] & Q[5]),KEY[0],SW[0],Q[6]);
   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]);
   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]);
   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]);
   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]);
   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]);
   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]);
   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]);
   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]);
   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]


   hexto7segment a(Q[3:0],HEX0);
   hexto7segment b(Q[7:4],HEX1);
  hexto7segment c(Q[11:8],HEX2);
  hexto7segment d(Q[15:12],HEX3);


endmodule


module t_ff(t, clk, reset, q);

   input t;
   input clk;
   input reset;
   wire w;
   output reg q;



always@(posedge clk)
begin

    if (reset==0)
        q=0;
    else begin

   if(t==0)
     q=q;
   else
        q=~q;

end
end
 endmodule

module hexto7segment (
    input [3:0] iDIG,
    output reg [6:0] oSEG
     );

 always@(iDIG) begin
    case(iDIG)
      4'h1: oSEG = 7'b1111001;
    4'h2: oSEG = 7'b0100100;  // ---t----
      4'h3: oSEG = 7'b0110000;  // lt    rt
      4'h4: oSEG = 7'b0011001;  // |      |
      4'h5: oSEG = 7'b0010010;  // ---m----
      4'h6: oSEG = 7'b0000010;  // |      |
      4'h7: oSEG = 7'b1111000;  // lb    rb
      4'h8: oSEG = 7'b0000000;  // |      |
      4'h9: oSEG = 7'b0011000;  // ---b----
      4'ha: oSEG = 7'b0001000;
      4'hb: oSEG = 7'b0000011;
      4'hc: oSEG = 7'b1000110;
      4'hd: oSEG = 7'b0100001;
      4'he: oSEG = 7'b0000110;
      4'hf: oSEG = 7'b0001110;
      4'h0: oSEG = 7'b1000000;
    endcase
  end

 endmodule
```

(2)實驗結果:



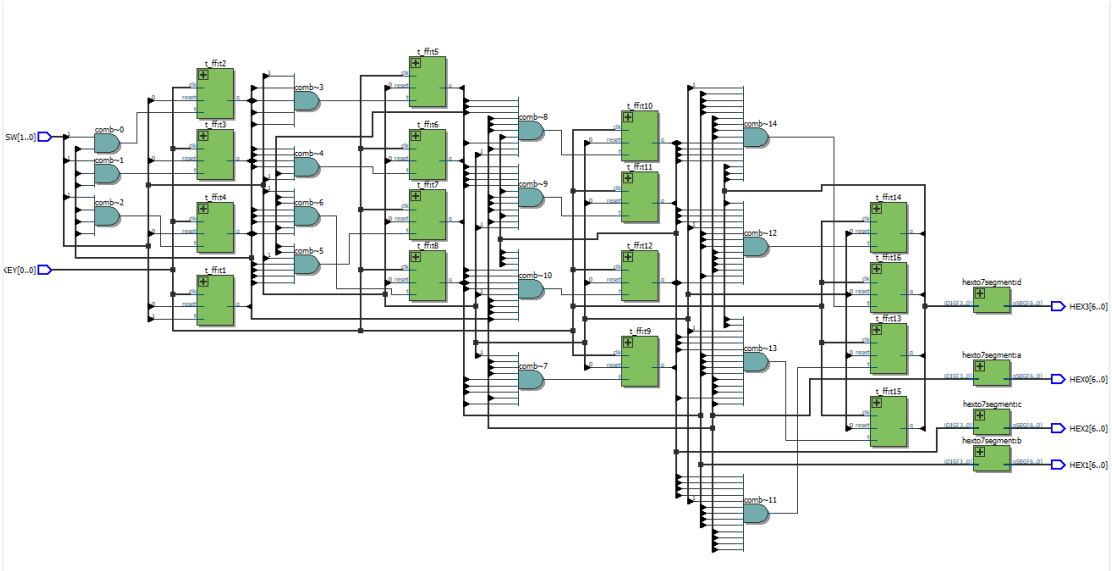






//中下:enable=0; 右下:reset=0;

(3)RTL 布局:



(4)問題與討論:

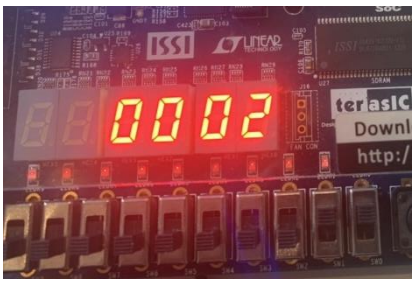
一開始遇到超多問題，像是 pin 的 TFF 保留字不能用，還有轉 HEX 時一直出錯，導致這次實驗做很久最後沒做完，不過更加深了 T flip-flop 的運用，同時也了解到 enable 跟 reset 怎麼控制 output，算是學到非常多。

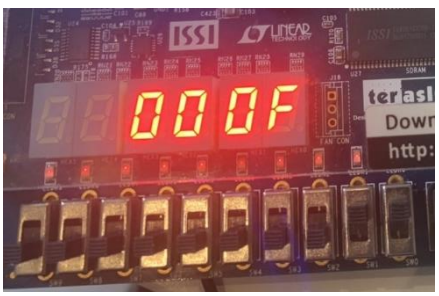
二、5-2

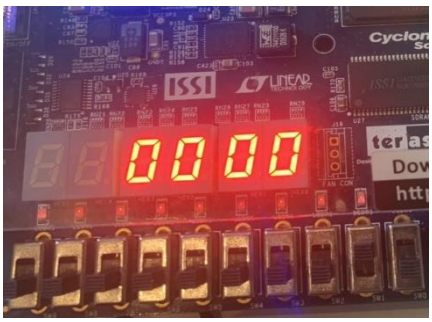
(1)實驗程式碼:

```
module labfive2(KEY,SW,HEX3,HEX2,HEX1,HEX0);
   input [0:0]KEY;
   input [1:0]SW;
   reg [15:0]Q;
   output [6:0]HEX3,HEX2,HEX1,HEX0;

   always@(posedge KEY[0])
   if(~SW[0]) Q=0;
   else begin
   if(SW[1])
      Q<=Q+1;
   end
   hexto7segment a(Q[3:0],HEX0);
   hexto7segment b(Q[7:4],HEX1);
   hexto7segment c(Q[11:8],HEX2);
   hexto7segment d(Q[15:12],HEX3);


endmodule

module hexto7segment (
   input [3:0] iDIG,
   output reg [6:0] oSEG
    );

 always@(iDIG) begin
   case(iDIG)
     4'h1: oSEG = 7'b1111001;
    4'h2: oSEG = 7'b0100100;  // ---t----
     4'h3: oSEG = 7'b0110000;  // lt    rt
     4'h4: oSEG = 7'b0011001;  // |      |
     4'h5: oSEG = 7'b0010010;  // ---m----
     4'h6: oSEG = 7'b0000010;  // |      |
     4'h7: oSEG = 7'b1111000;  // lb    rb
     4'h8: oSEG = 7'b0000000;  // |
     4'h9: oSEG = 7'b0011000;  // ---
     4'ha: oSEG = 7'b0001000;
     4'hb: oSEG = 7'b0000011;
     4'hc: oSEG = 7'b1000110;
     4'hd: oSEG = 7'b0100001;
     4'he: oSEG = 7'b0000110;
     4'hf: oSEG = 7'b0001110;
     4'h0: oSEG = 7'b1000000;
   endcase
 end

endmodule
```

(2)實驗結果:

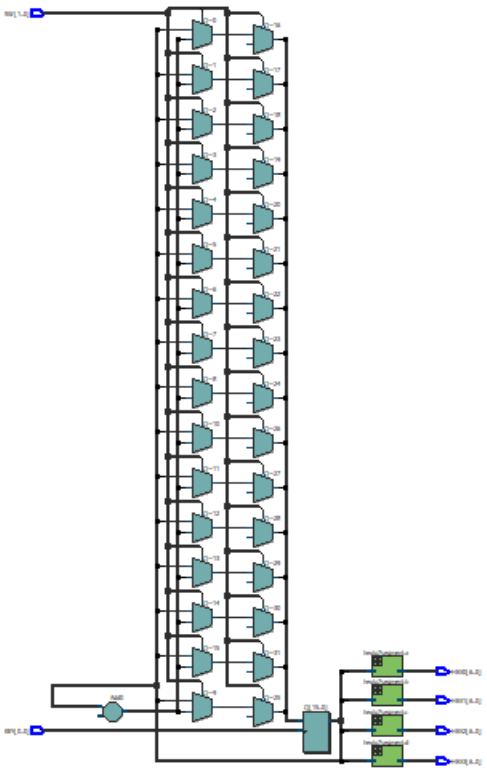






//中下:enable=0; 右下:reset=0;

(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

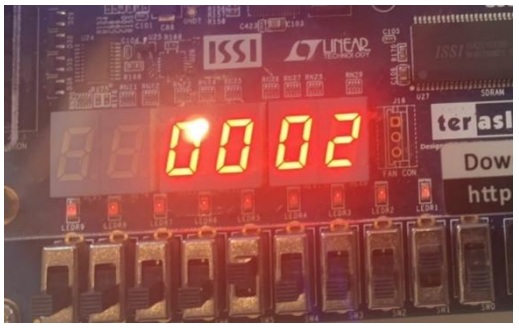
(1)實驗程式碼:

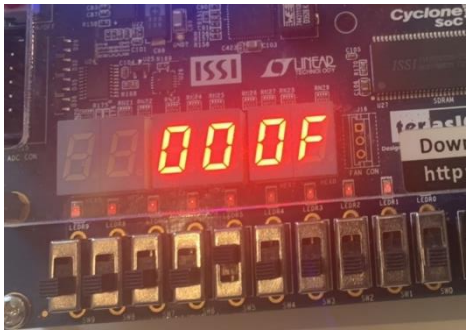
```
module labfivee5(KEY,SW,HEX3,HEX2,HEX1,HEX0)
   input [0:0]KEY;
   input [1:0]SW;
   wire [15:0]Q;
   output [6:0]HEX3,HEX2,HEX1,HEX0;

   ha(SW[1],KEY[0],SW[0],Q);

   hexto7segment e(Q[3:0],HEX0);
   hexto7segment f(Q[7:4],HEX1);
   hexto7segment g(Q[11:8],HEX2);
   hexto7segment h(Q[15:12],HEX3);

endmodule

module hexto7segment (
   input [3:0] iDIG,
   output reg [6:0] oSEG
    );

 always@(iDIG) begin
   case(iDIG)
     4'h1: oSEG = 7'b1111001;
   4'h2: oSEG = 7'b0100100;  // ---t----
     4'h3: oSEG = 7'b0110000;  // lt    rt
     4'h4: oSEG = 7'b0011001;  // |     |
     4'h5: oSEG = 7'b0010010;  // ---m----
     4'h6: oSEG = 7'b0000010;  // |     |
     4'h7: oSEG = 7'b1111000;  // lb    rb
     4'h8: oSEG = 7'b0000000;  // |     |
     4'h9: oSEG = 7'b0011000;  // ---b----
     4'ha: oSEG = 7'b0001000;
     4'hb: oSEG = 7'b0000011;
     4'hc: oSEG = 7'b1000110;
     4'hd: oSEG = 7'b0100001;
     4'he: oSEG = 7'b0000110;
     4'hf: oSEG = 7'b0001110;
     4'h0: oSEG = 7'b1000000;
   endcase
 end

endmodule
```

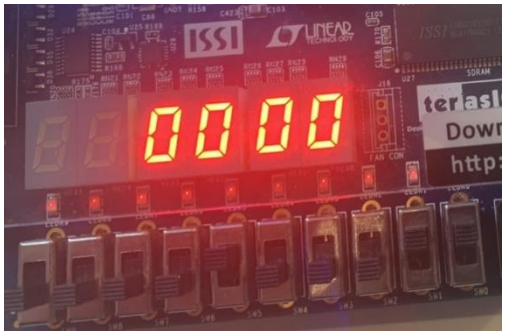
```
`timescale 1 ps / 1 ps
// synopsys translate_on
module ha (
   clk_en,
   clock,
   sclr,
   q);

   input   clk_en;
   input   clock;
   input   sclr;
   output   [15:0]  q;

   wire [15:0] sub_wire0;
   wire [15:0] q = sub_wire0[15:0];

   lpm_counter LPM_COUNTER_component (
            .clk_en (clk_en),
            .clock (clock),
            .sclr (sclr),
            .q (sub_wire0),
            .aclr (1'b0),
            .aload (1'b0),
            .aset (1'b0),
            .cin (1'b1),
            .cnt_en (1'b1),
            .cout (),
            .data ({16{1'b0}}),
            .eq (),
            .sload (1'b0),
            .sset (1'b0),
            .updown (1'b1));
   defparam
      LPM_COUNTER_component.lpm_direction = "UP",
    LPM_COUNTER_component.lpm_port_updown = "PORT_UNUSED",
    LPM_COUNTER_component.lpm_type = "LPM_COUNTER",
    LPM_COUNTER_component.lpm_width = 16;


endmodule
```

//右:Megawizard_counter

(2)實驗結果:

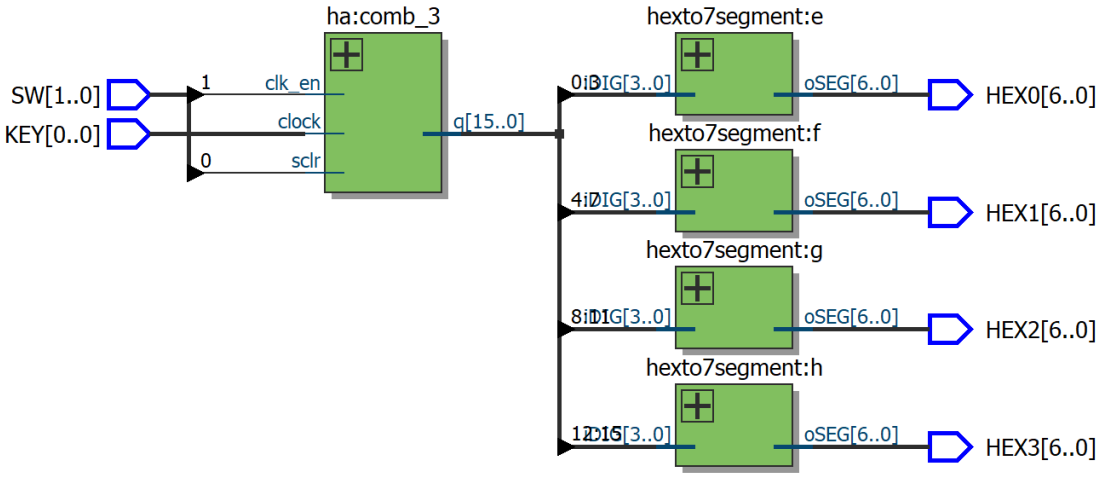






//中下:enable=0; 右下:reset=1;

(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:

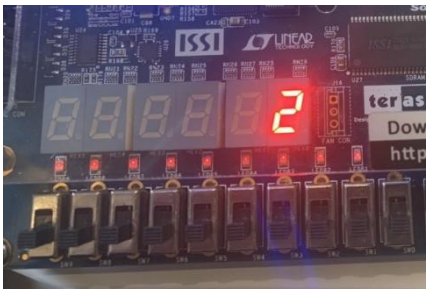
```
module labfive4(CLOCK_50,HEX0);
   input CLOCK_50;
   wire [3:0]Q;
   output [6:0]HEX0;
   reg [3:0]a;
   reg [30:0]count;
   reg temp;

   always@(posedge CLOCK_50) begin

   if(count==50000000) begin
      count=0;
      temp=1;
      if(Q==4'h9)
         a<=1;
      else
         a<=0;
      end
   else begin
      count=count+1;
      temp=0;
   end
   end

   test3(temp,a,Q);


   hexto7segment e(Q[3:0],HEX0);

endmodule

module hexto7segment (
   input [3:0] iDIG,
   output reg [6:0] oSEG
    );

 always@(iDIG) begin
   case(iDIG)
     4'h1: oSEG = 7'b1111001;
    4'h2: oSEG = 7'b0100100;  /
     4'h3: oSEG = 7'b0110000;
     4'h4: oSEG = 7'b0011001;
     4'h5: oSEG = 7'b0010010;
     4'h6: oSEG = 7'b0000010;
     4'h7: oSEG = 7'b1111000;
     4'h8: oSEG = 7'b0000000;
     4'h9: oSEG = 7'b0011000;
     4'ha: oSEG = 7'b0001000;
     4'hb: oSEG = 7'b0000011;
     4'hc: oSEG = 7'b1000110;
     4'hd: oSEG = 7'b0100001;
     4'he: oSEG = 7'b0000110;
     4'hf: oSEG = 7'b0001110;
     4'h0: oSEG = 7'b1000000;
   endcase
 end

endmodule
```

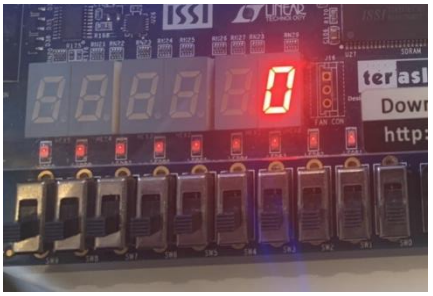
```
`timescale 1 ps / 1 ps
// synopsys translate_on
module test3 (
   clock,
   sclr,
   q);

   input   clock;
   input   sclr;
   output   [3:0]  q;

   wire [3:0] sub_wire0;
   wire [3:0] q = sub_wire0[3:0];

   lpm_counter LPM_COUNTER_component (
            .clock (clock),
            .sclr (sclr),
            .q (sub_wire0),
            .aclr (1'b0),
            .aload (1'b0),
            .aset (1'b0),
            .cin (1'b1),
            .clk_en (1'b1),
            .cnt_en (1'b1),
            .cout (),
            .data ({4{1'b0}}),
            .eq (),
            .sload (1'b0),
            .sset (1'b0),
            .updown (1'b1));
   defparam
      LPM_COUNTER_component.lpm_direction = "UP",
      LPM_COUNTER_component.lpm_port_updown = "PORT_UNUSED",
      LPM_COUNTER_component.lpm_type = "LPM_COUNTER",
      LPM_COUNTER_component.lpm_width = 4;


endmodule
```

//右:Megawizard_counter

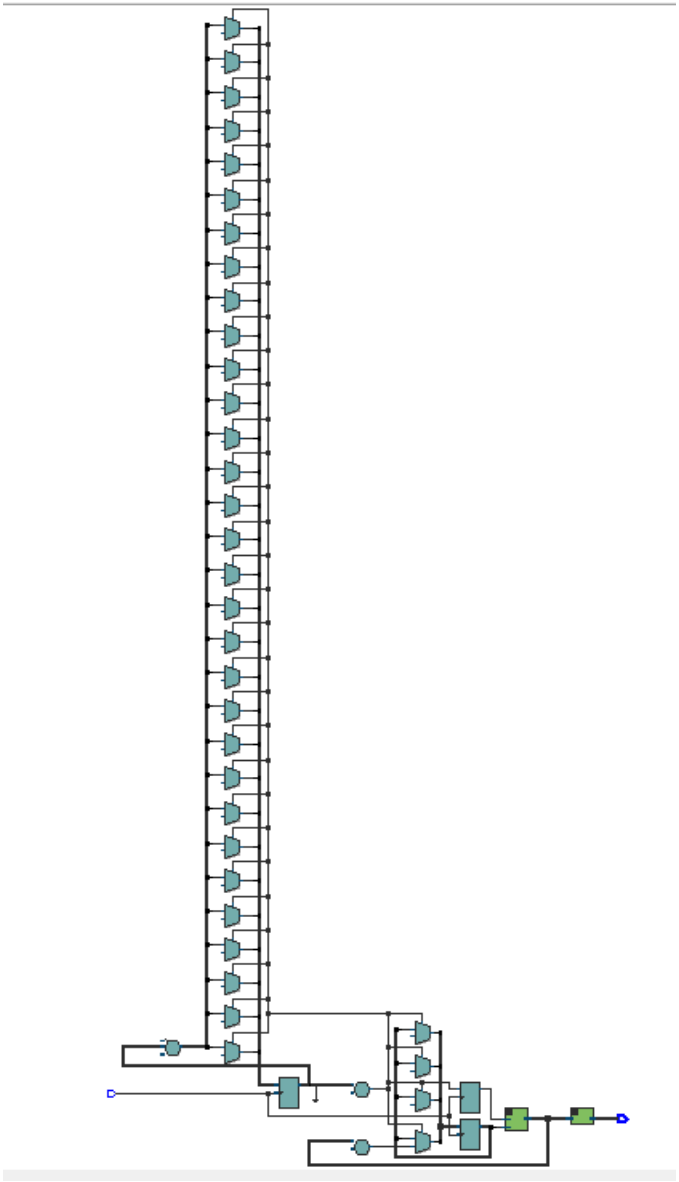
(2)實驗結果:





(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。