

Laboratory 9

軟硬體共同實驗 (一): 正反器實作



Department of Electrical Engineering
National Cheng Kung University

實驗目的

- 瞭解組合電路、序向電路、觸發(Trigger)與除頻的概念
- 學習Latch與Flip-flop的邏輯電路
- 實作正反器電路並且利用FPGA驗證

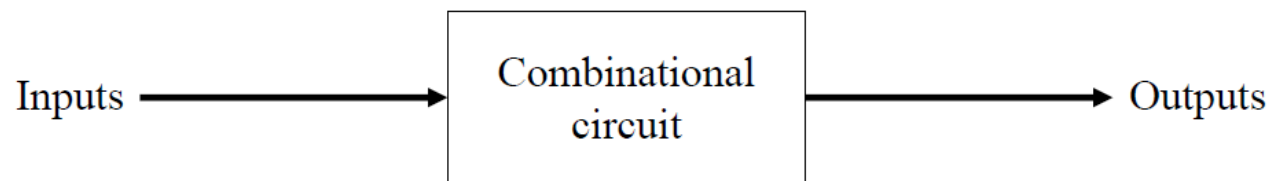
使用器材

- 桌上型電腦
- Xilinx FPGA 板

組合電路 v.s. 序向電路

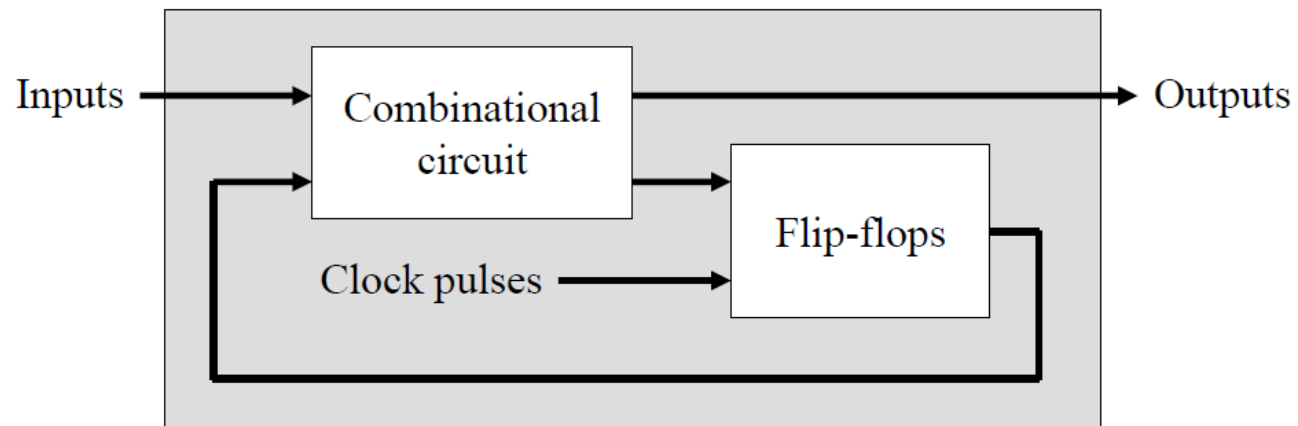
組合電路： $O \leftarrow I$

(例如：加法器)



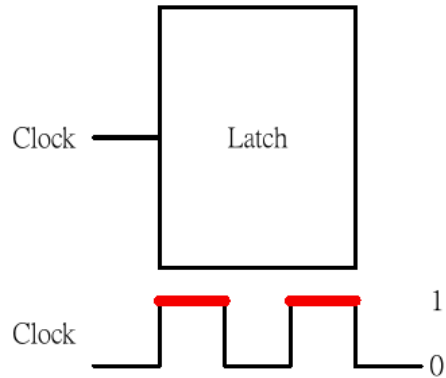
序向電路： $O \leftarrow I \times S$

(例如：計數器)

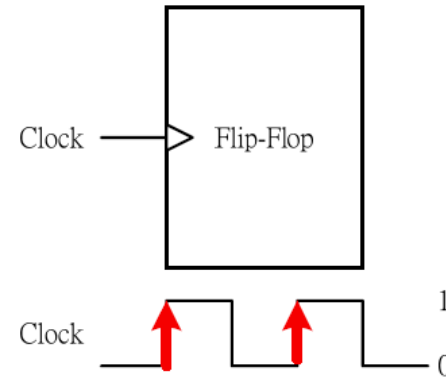


觸發 Trigger

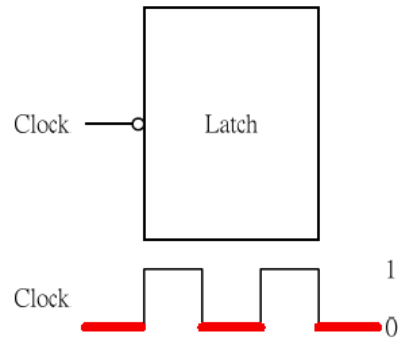
正層觸發



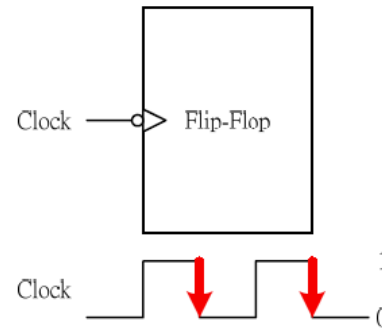
正緣觸發



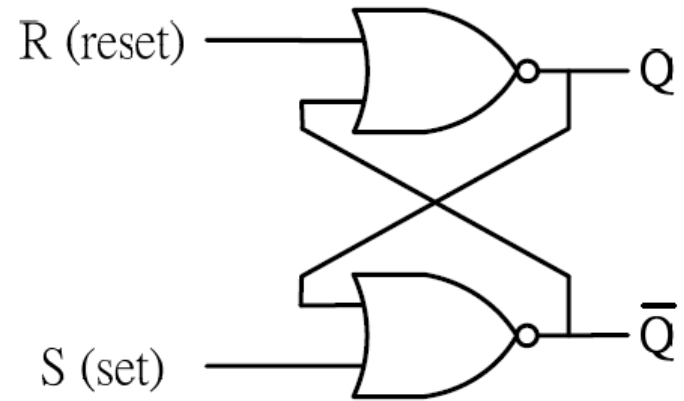
負層觸發



負緣觸發

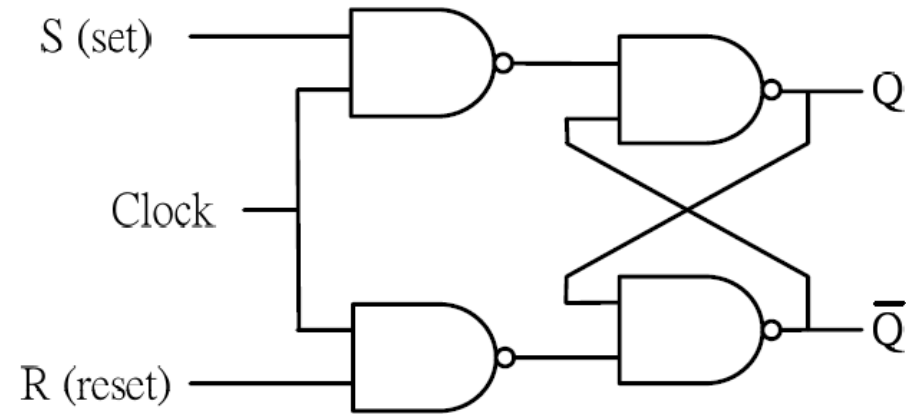


SR Latch



S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

SR Latch with Control Input

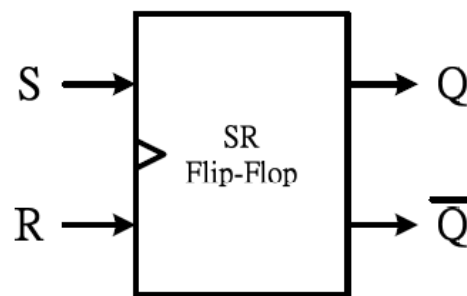


Clock	S	R	Q_{n+1}
0	x	x	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	?

四種正反器 (Flip-flop) (1/2)

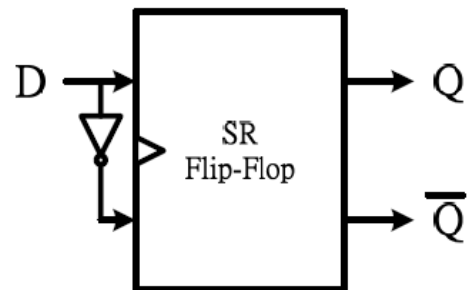
(1) SR正反器

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?



(2) D正反器

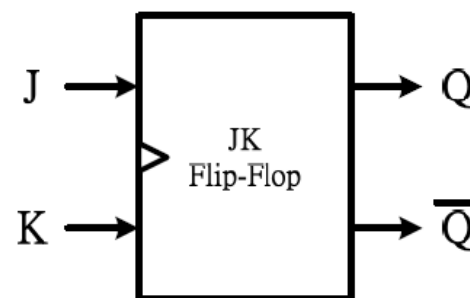
D	S	R	Q_{n+1}
0	0	1	0
1	1	0	1



四種正反器 (Flip-flop) (2/2)

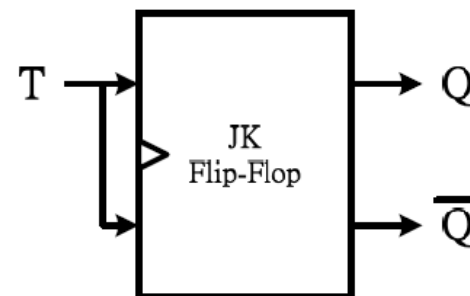
(3) JK正反器

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

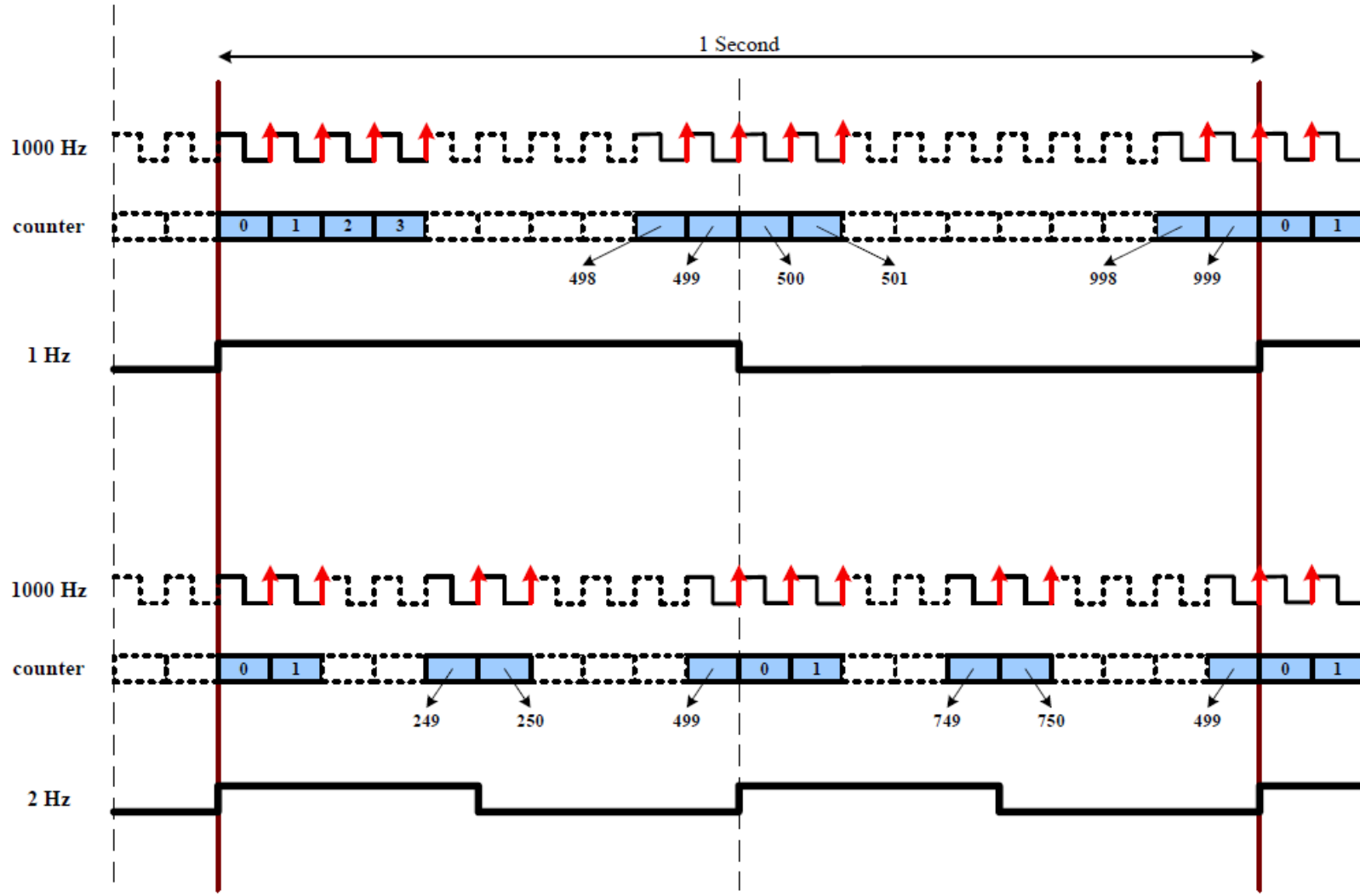


(4) T正反器

T	J	K	Q_{n+1}
0	0	0	Q_n
1	1	1	$\overline{Q_n}$



簡易的除頻電路

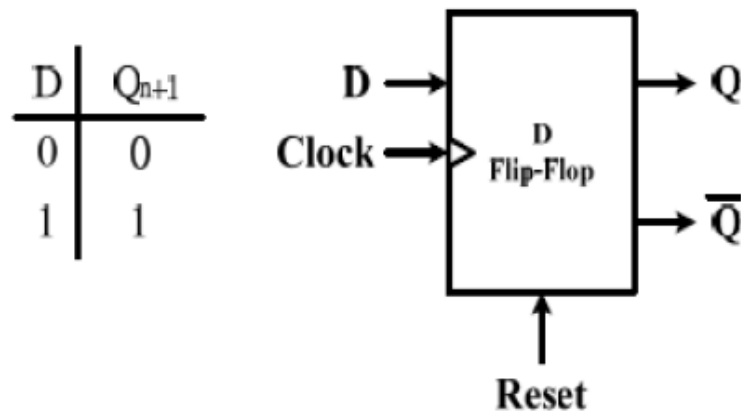


實作題

本次實作分為四個基本實作與一個結報問題

實作題(一) D型正反器 (1/5)

- 請寫出一個正緣觸發，且具有非同步 Reset 功能的 D Flip-Flop。
- 利用虛擬裝置(VerilInstrument)，產生一個指撥開關來代表輸入訊號 D，一個按鈕代表 Reset 訊號，並且使用七段顯示器顯示 D Flip-flop的內容。



實作題(一) D型正反器 (2/5)

步驟

1. 請在 “c:\logiclab\<自己的學號>” 的路徑下新增一資料夾 lab9_1
2. 完成下列*.v檔
 - d_ff.v
 - clk_div.v
 - SevenSeg.v
 - lab9_1.v
3. 在虛擬儀表板上，產生一個按鈕、一個指撥開關與七段顯示器，並且設定對應的接腳。

實作題(一) D型正反器 (3/5)

```

module clock_div (clk_48MHZ , clk_1HZ) ;

    input clk_48MHZ ;
    output clk_1HZ ;

    reg [31:0] counter_1HZ ;
    reg clk_1HZ ;

    always@(posedge clk_48MHZ)
    begin
        if(counter_1HZ<32'd48000_000)
            counter_1HZ<=counter_1HZ+32'd1 ;
        else
            counter_1HZ<=32'd1 ;
        end

        always@(posedge clk_48MHZ)
        begin
            if(counter_1HZ<=32'd24000_000)
                clk_1HZ<=32'd1 ;
            else
                clk_1HZ<=32'd0 ;
            end
        end
    endmodule

```

clk_div.v

```

module d_ff(clock,reset,din_d,dout_q,dout_nq) ;

    input clock ;
    input reset ;
    input din_d ;
    output dout_q ;
    output dout_nq ;

    reg data ;

    always@(posedge clock or posedge reset )
    begin
        if(reset)
            data <= 1'b0 ;
        else
            
        end

    assign dout_q = data ;
    assign dout_nq = ~data ;

endmodule

```

d_ff.v

實作題(一) D型正反器 (4/5)

```

module SevenSeg (dout,din) ;

    input [3:0] din ;

    output [6:0] dout ;
    reg [6:0] dout ;

    always@(din)
    begin
        case(din)
            4'b0000 : dout = 7'b0111111 ;
            4'b0001 : dout = 7'b0000110 ;
            4'b0010 : dout = 7'b1011011 ;
            4'b0011 : dout = 7'b1001111 ;
            4'b0100 : dout = 7'b1100110 ;
            4'b0101 : dout = 7'b1101101 ;
            4'b0110 : dout = 7'b1111101 ;
            4'b0111 : dout = 7'b0000111 ;
            4'b1000 : dout = 7'b1111111 ;
            4'b1001 : dout = 7'b1101111 ;
            4'b1010 : dout = 7'b1110111 ;
            4'b1011 : dout = 7'b1111100 ;
            4'b1100 : dout = 7'b0111001 ;
            4'b1101 : dout = 7'b1011110 ;
            4'b1110 : dout = 7'b1111001 ;
            4'b1111 : dout = 7'b1110001 ;
            default : dout = 7'b0000000 ;
        endcase
    end
endmodule

```

SevenSeg.v

```

module lab9_1 (clock,reset, din , dout_digit);

    input clock , reset , din ;
    output [6:0] dout_digit ;

    wire [3:0] value ;

    clock_div clock_div(
        .clk_48MHZ (clock),
        .clk_1HZ (clk_1HZ)
    );

    d_ff d_ff(
        .clock (clk_1HZ),
        .reset (reset),
        .din_d (din),
        .dout_q (value[0])
    );

    assign value[3:1] = 3'b000 ;

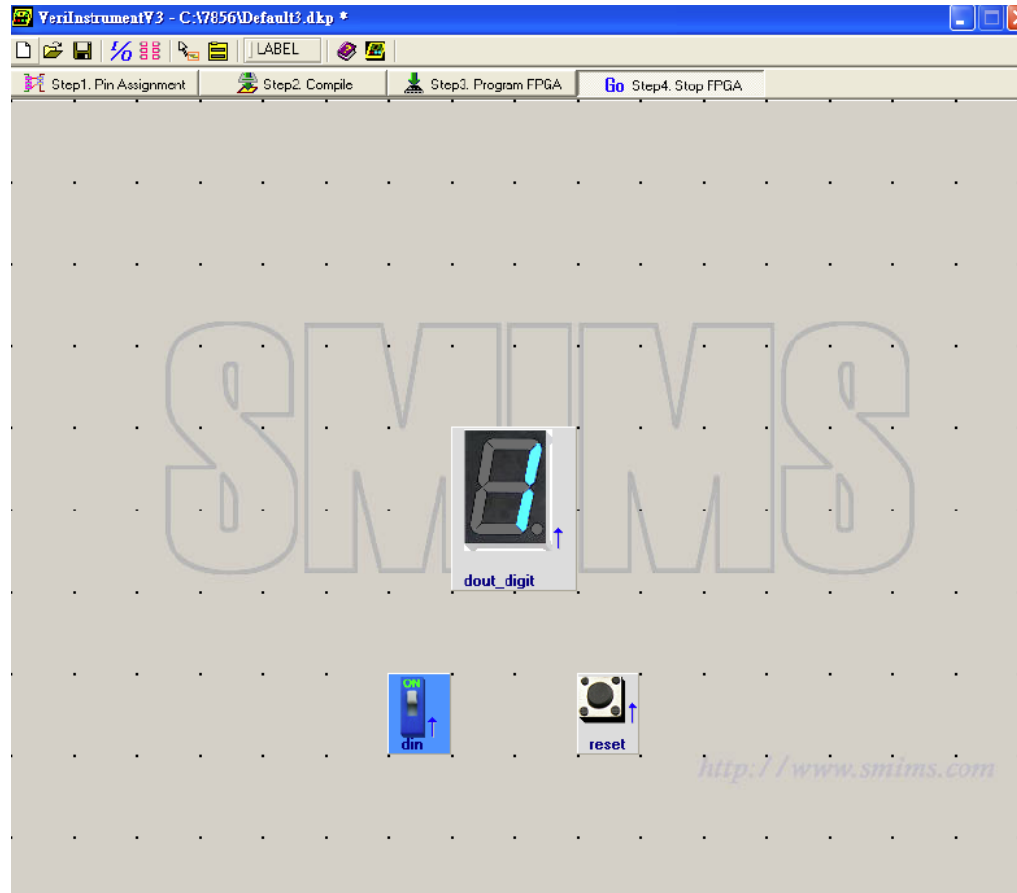
    SevenSeg SevenSeg(
        .din (value[3:0]),
        .dout (dout_digit)
    );

endmodule

```

lab9_1.v

實作題(一) D型正反器 (5/5)



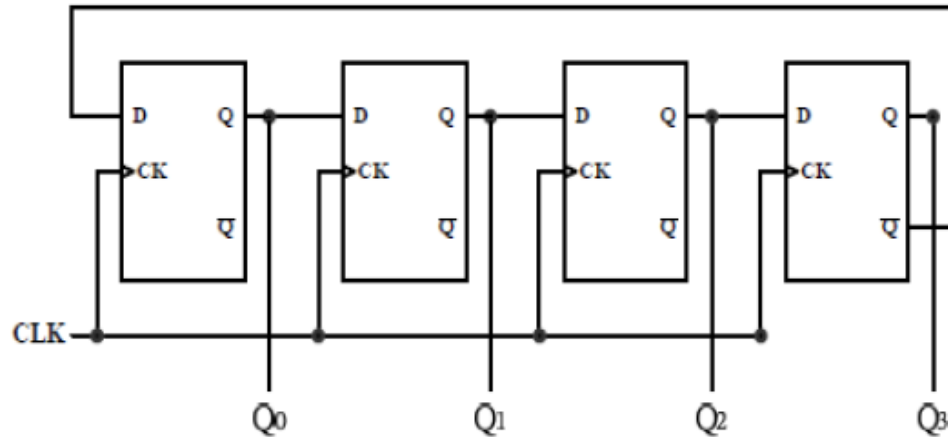
配置圖

實作題(二) Johnson Counter (1/6)

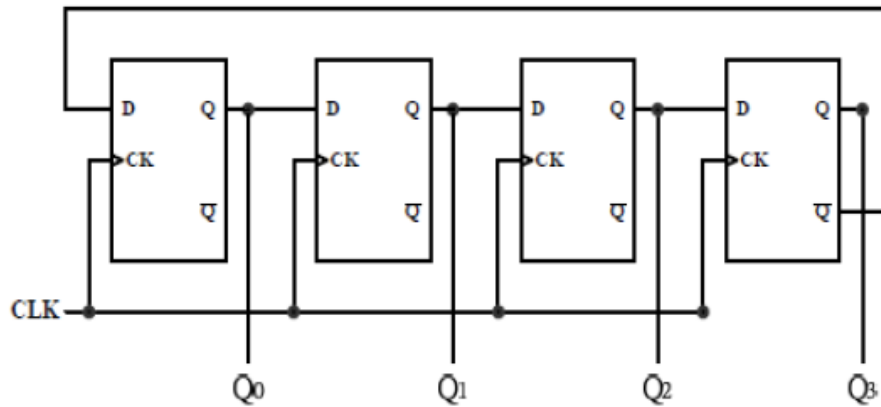
➤請利用 D Flip-Flop 設計具有非同步 Reset 功能的四位元

Johnson Counter

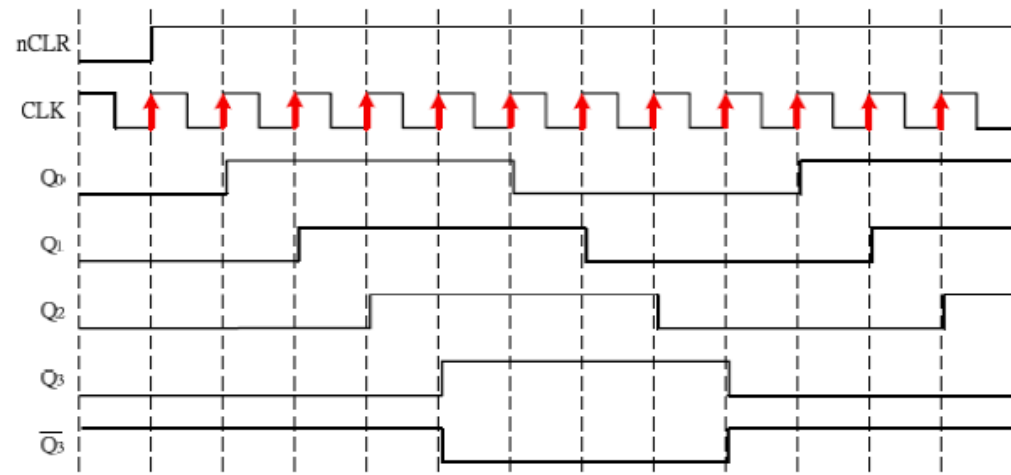
- 使用一個按鈕代表 Reset 訊號。
- D Flip-flop 的內容(Q) 請以 4 顆 LED 顯示。
- Johnson Counter 的最後輸出 (J) 則以 8 顆 LED 顯示。



實作題(二) Johnson Counter (2/6)



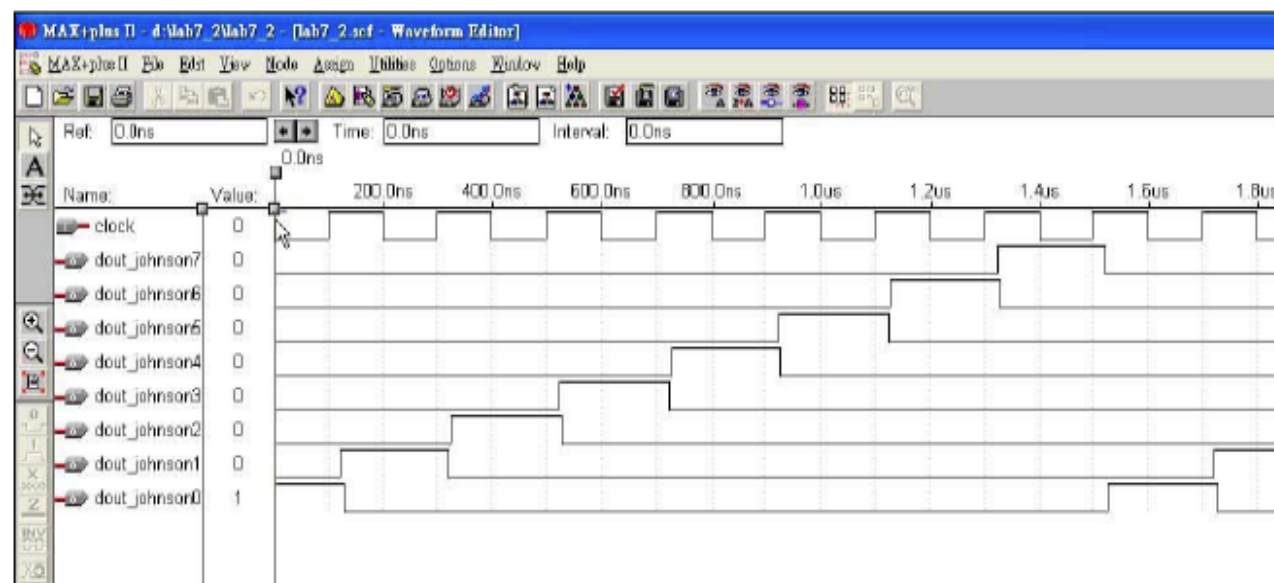
T = 0	0	0	0	0
T = 1	1	0	0	0
T = 2	1	1	0	0
T = 3	1	1	1	0
T = 4	1	1	1	1
T = 5	0	1	1	1
T = 6	0	0	1	1
T = 7	0	0	0	1
T = 8	0	0	0	0



實作題(二) Johnson Counter (3/6)

4-bits Johnson Counter Output Function

	Q_0	Q_1	Q_2	Q_3	Function
J_0	0	0	0	0	$\overline{Q_0} \cdot \overline{Q_3}$
J_1	1	0	0	0	$Q_0 \cdot \overline{Q_1}$
J_2	1	1	0	0	$Q_1 \cdot \overline{Q_2}$
J_3	1	1	1	0	$Q_2 \cdot \overline{Q_3}$
J_4	1	1	1	1	$Q_0 \cdot Q_3$
J_5	0	1	1	1	$\overline{Q_0} \cdot Q_1$
J_6	0	0	1	1	$\overline{Q_1} \cdot Q_2$
J_7	0	0	0	1	$\overline{Q_2} \cdot Q_3$



實作題(二) Johnson Counter (4/6)

步驟

1. 請在 “c:\logiclab**<自己的學號>**” 的路徑下新增一資料夾 **lab9_2**
2. 完成下列*.v檔
 - d_ff.v
 - clk_div.v
 - johnson_counter.v
 - lab9_2.v
3. 在虛擬儀表板上，產生一個按鈕與**十二個LED燈**(**四個給輸出訊號Q**、**八個給輸出訊號J**)，並且設定對應的接腳。

實作題(二) Johnson Counter (5/6)

```

module johnson_counter (clock , reset,dout_q,dout_j);
    input clock,reset ;
    output [3:0] dout_q ;
    output [7:0] dout_j ;

    wire q0,q1,q2,q3 ;
    wire nq3 ;

    d_ff dff_b0(
        .clock      (clock) ,
        .reset      (reset) ,
        .din_d      (nq3) ,
        .dout_q     (q0)
    );
    d_ff dff_b1(
        .clock      (clock) ,
        .reset      (reset) ,
        .din_d      (q0) ,
        .dout_q     (q1)
    );
    d_ff dff_b2(
        .clock      (clock) ,
        .reset      (reset) ,
        .din_d      ( ) ,
        .dout_q     ( )
    );
    d_ff dff_b3(
        .clock      (clock) ,
        .reset      (reset) ,
        .din_d      ( ) ,
        .dout_q     ( ) ,
        .dout_nq   (nq3)
    );

    assign dout_q = {q3,q2,q1,q0} ;
    assign dout_j[0] = (~q3) & (~q0) ;
    assign dout_j[1] = (~q1) & (q0) ;
    assign dout_j[2] = (~q2) & (q1) ;
    assign dout_j[3] = (~q3) & (q2) ;
    assign dout_j[4] = (q3) & (q0) ;
    assign dout_j[5] = (q1) & (~q0) ;
    assign dout_j[6] = (q2) & (~q1) ;
    assign dout_j[7] = (q3) & (~q2) ;

endmodule

```

johnson_counter.v

```

module lab9_2 (clock,reset,led_a,led_b);
    input clock ,reset ;
    output [3:0] led_a ;
    output [7:0] led_b ;

    wire reset ;
    wire clock_1Hz ;

    wire [3:0] q ;
    wire [7:0] j ;

    clock_div clock_div(
        .clk_48MHZ (clock),
        .clk_1HZ (clock_1Hz)
    );

    johnson_counter jc(
        .clock      (clock_1Hz),
        .reset      (reset),
        .dout_q     (q),
        .dout_j     (j)
    );

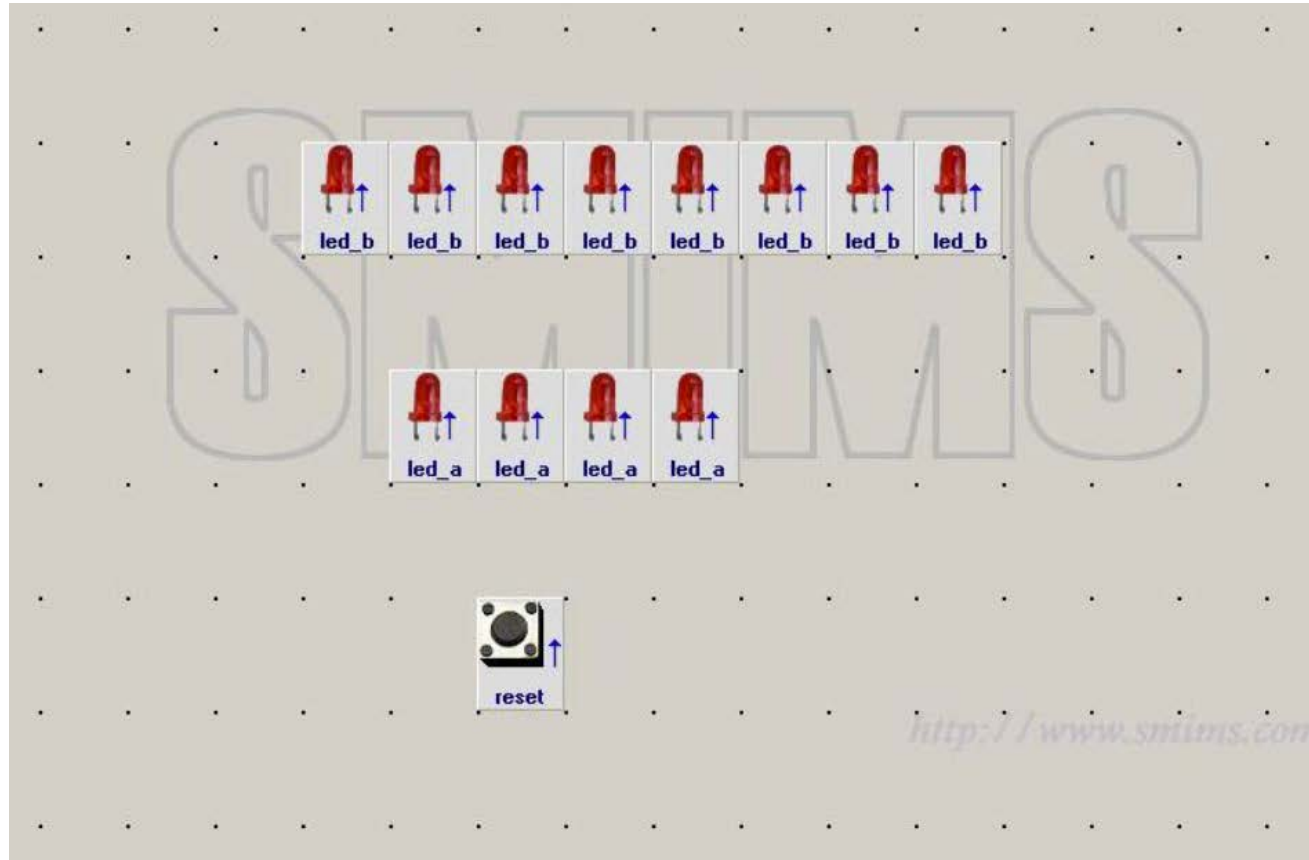
    assign led_a = q ;
    assign led_b = j ;

endmodule

```

lab9_2.v

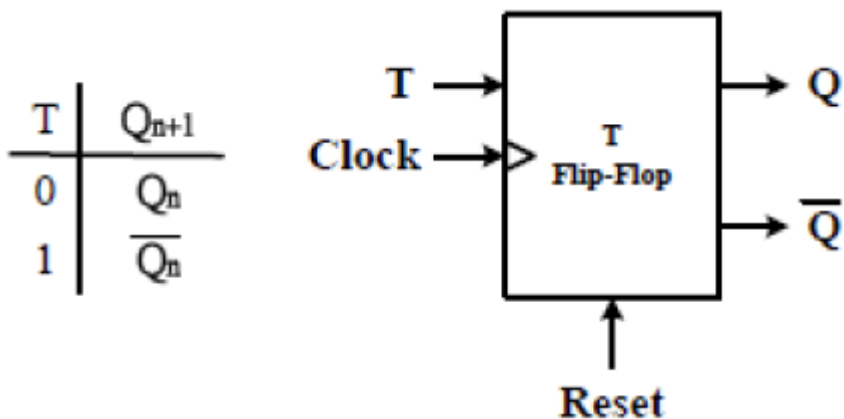
實作題(二) Johnson Counter (6/6)



配置圖

實作題(三) T型正反器 (1/4)

- 請寫出一個正緣觸發，且具有非同步 Reset 功能的 T Flip-flop。
- 利用虛擬裝置(VerilInstrument)，產生一個指撥開關來代表輸入訊號 T，一個按鈕代表 Reset 訊號，並且使用七段顯示器顯示 T Flip-flop的內容。



實作題(三) T型正反器 (2/4)

步驟

1. 請在 “c:\logiclab\<自己的學號>” 的路徑下新增一資料夾 lab9_3
2. 完成下列*.v檔
 - t_ff.v
 - clk_div.v
 - SevenSeg.v
 - lab9_3.v
3. 在虛擬儀表板上，產生一個按鈕、一個指撥開關與七段顯示器，並且設定對應的接腳。

實作題(三) T型正反器 (3/4)

```

module t_ff(clock,reset,din_t,dout_q,dout_nq) ;

input clock ;
input reset ;
input din_t ;
output dout_q ;
output dout_nq ;

reg data ;

always@(posedge clock or posedge reset )
begin
    if(reset)
        data <= 1'b0 ;
    else if(din_t)
        
    end

assign dout_q = data ;
assign dout_nq = ~data ;
endmodule

```

t_ff.v

```

module lab9_3 (clock,reset, din , dout_digit);

input clock , reset , din ;
output [6:0] dout_digit ;

wire [3:0] value ;
wire clock_1Hz ;

clk_div clk_div(
    .clk_48MHZ (clock),
    .clk_1HZ (clock_1Hz)
);

t_ff t_ff(
    .clock (clock_1Hz),
    .reset (reset),
    .din_t (din),
    .dout_q (value[0])
);

assign value[3:1] = 3'b000 ;

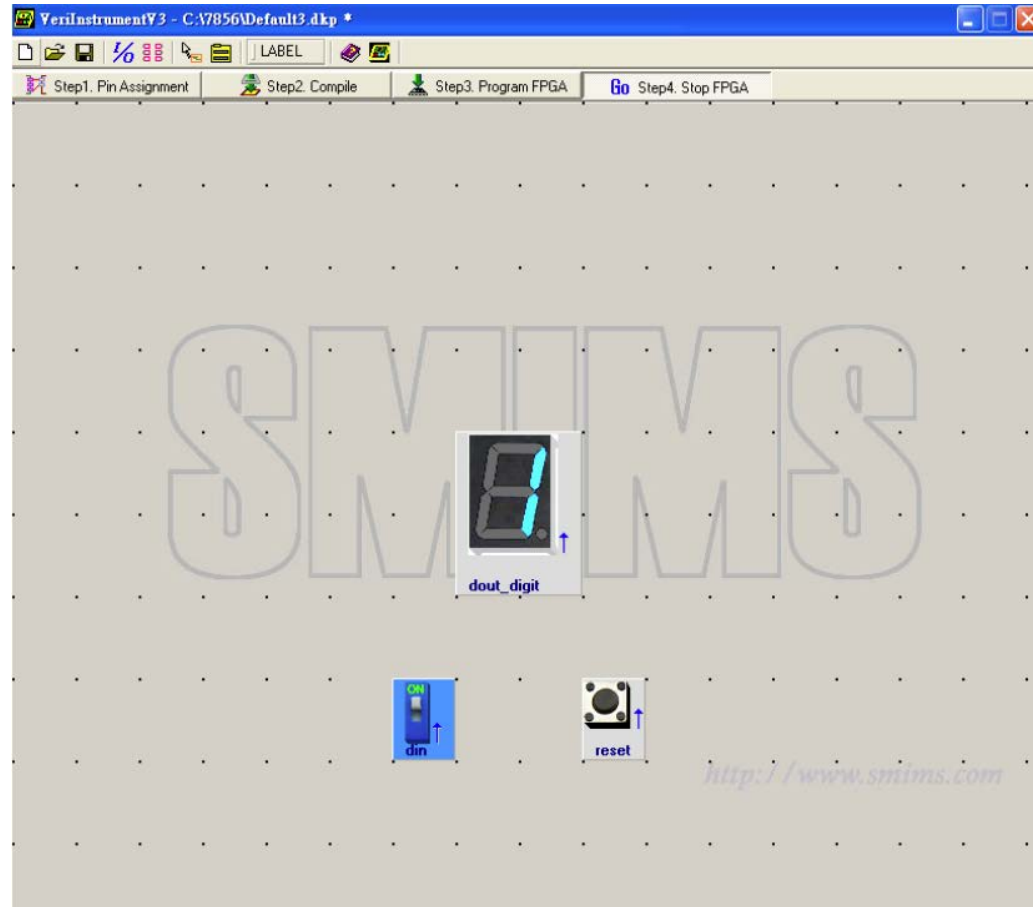
SevenSeg SevenSeg(
    .din (value[3:0]),
    .dout (dout_digit)
);

endmodule

```

lab9_3.v

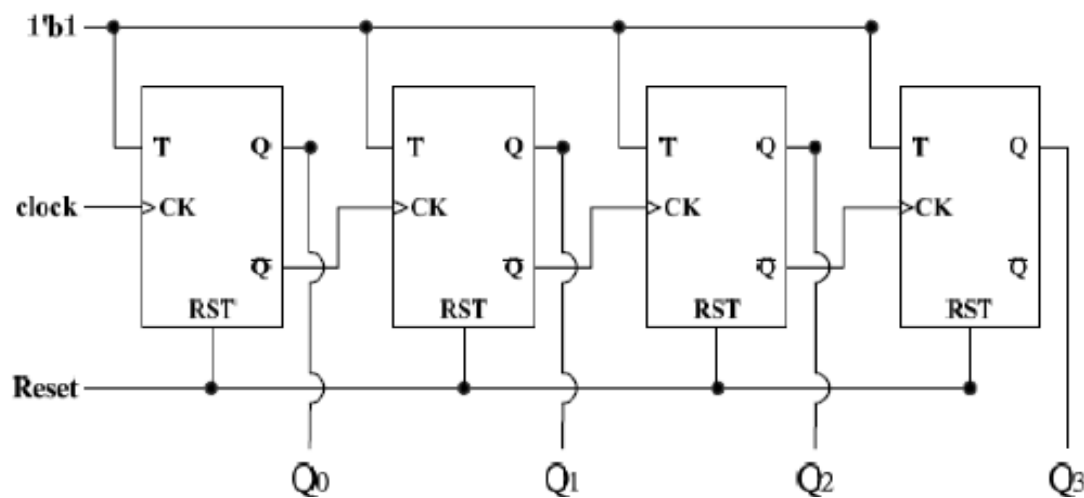
實作題(三) T型正反器 (4/4)



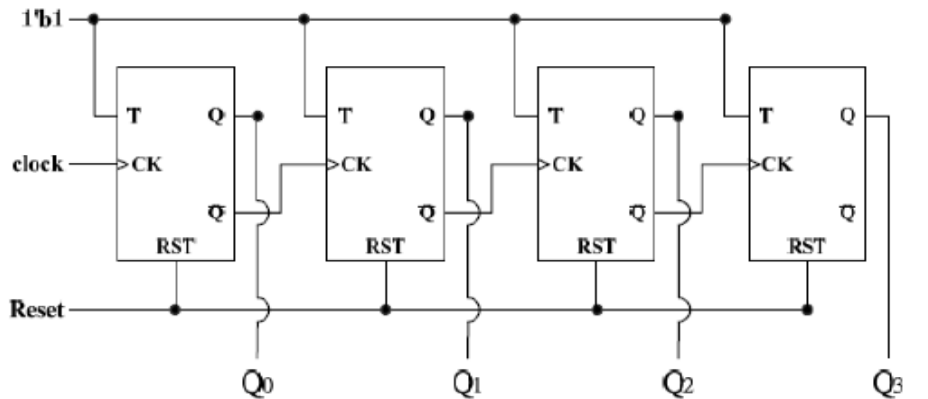
配置圖

實作題(四) 四位元非同步計數器 (1/4)

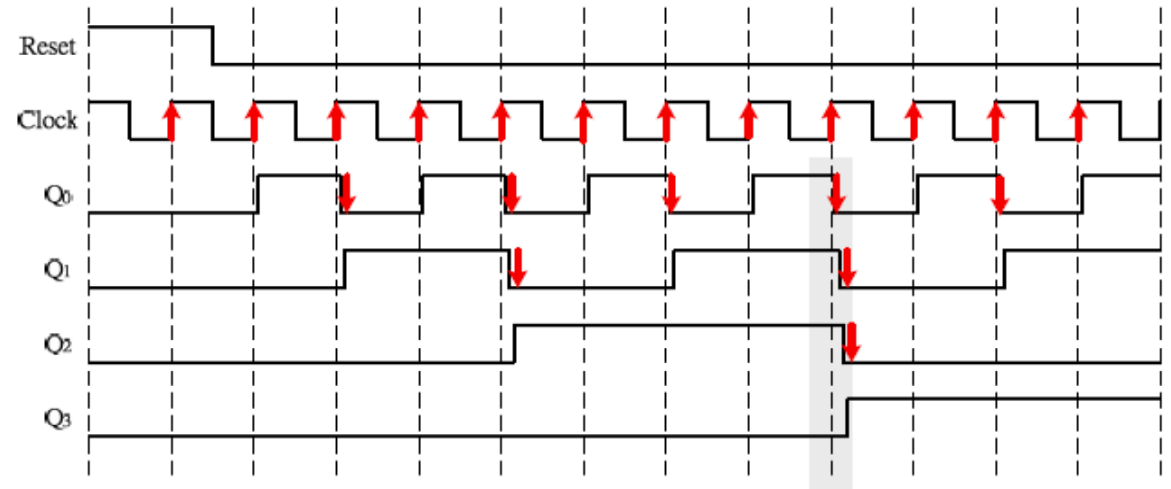
- 請利用T 型正反器(T Flip-flop) 設計一個四位元的非同步計數器。
- 使用一個按鈕代表 Reset 訊號。
 - 使用七段顯示器顯示目前計數器的值。



實作題(四) 四位元非同步計數器 (2/4)



	Q ₀	Q ₁	Q ₂	Q ₃
T = 0	0	0	0	0
T = 1	1	0	0	0
T = 2	0	1	0	0
T = 3	1	1	0	0
T = 4	0	0	1	0
T = 5	1	0	1	0
T = 6	0	1	1	0
T = 7	1	1	1	0
T = 8	0	0	0	1
T = 9	1	0	0	1



實作題(四) 四位元非同步計數器 (3/4)

```

module asyn_counter (clock,reset,dout_q);
    input    clock,reset ;
    output   [3:0] dout_q ;

    wire logic1 ;
    wire q0,q1,q2,q3 ;
    wire nq0,nq1,nq2 ;

    assign logic1 = 1'b1 ;

    t_ff dff_b0(
        .clock    (clock) ,
        .reset    (reset) ,
        .din_t    (logic1) ,
        .dout_q   (q0) ,
        .dout_nq  (nq0)
    );
    t_ff tff_b1(
        .clock    (nq0) ,
        .reset    (reset) ,
        .din_t    (logic1) ,
        .dout_q   (q1) ,
        .dout_nq  (nq1)
    );
    t_ff tff_b2(
        .clock    (nq1) ,
        .reset    (reset) ,
        .din_t    (logic1) ,
        .dout_q   (q2) ,
        .dout_nq  (nq2)
    );

    t_ff tff_b3(
        .clock    ( ) ,
        .reset    ( ) ,
        .din_t    ( ) ,
        .dout_q   ( )
    );
    assign dout_q = {q3,q2,q1,q0} ;
endmodule

```

asyn_counter.v

```

module lab9_4 (clock,reset , dout_digit);

    input clock , reset ;
    output [6:0] dout_digit ;

    wire [3:0] value ;
    wire clock_1Hz ;

    clock_div clock_div(
        .clk_48MHZ    (clock) ,
        .clk_1HZ     (clock_1Hz)
    );

    asyn_counter ac(
        .clock    (clock_1Hz) ,
        .reset    (reset) ,
        .dout_q   (value)
    );

    // assign value[3:1] = 3'b000 ;

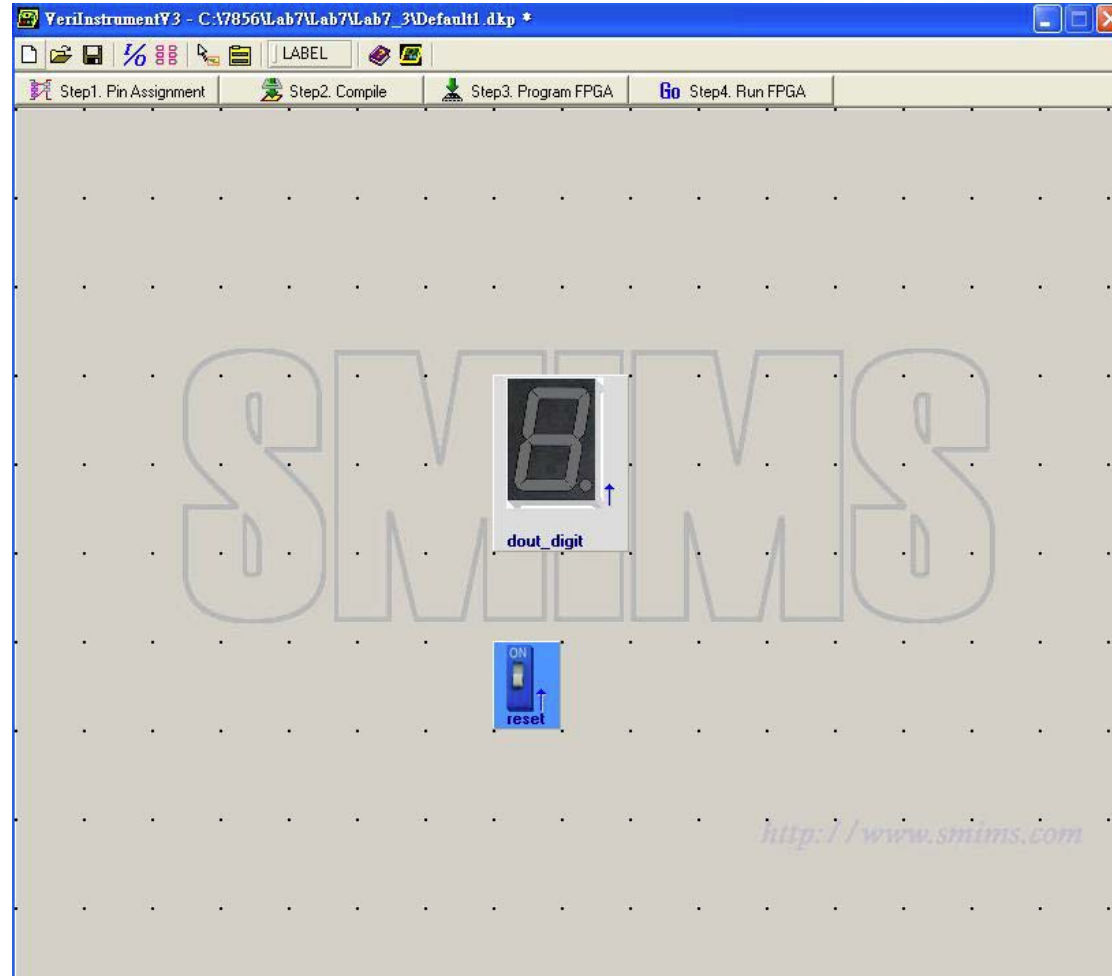
    SevenSeg SevenSeg(
        .din      (value) ,
        .dout     (dout_digit)
    );

endmodule

```

lab9_4.v

實作題(四) 四位元非同步計數器 (4/4)



配置圖

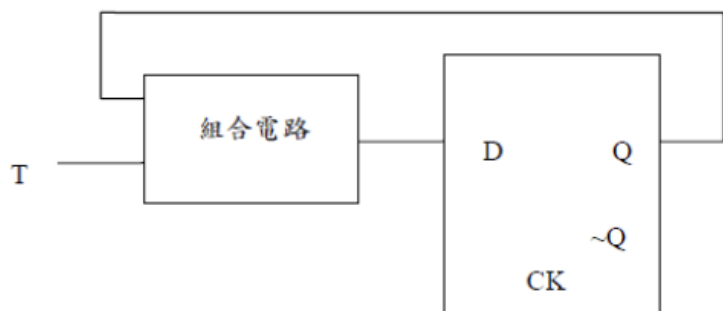
結報問題

本週的結報，請每組的同學：

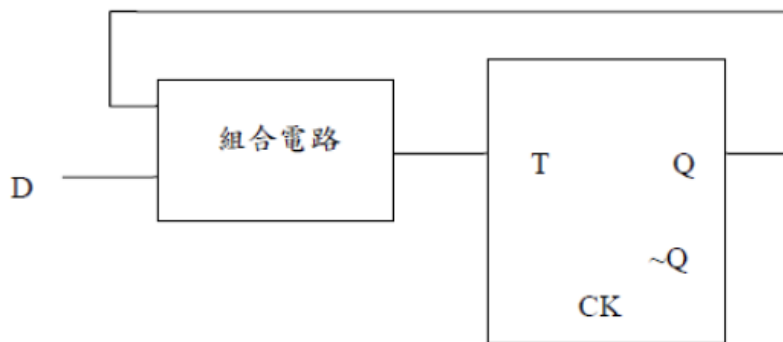
1. 在 D 型正反器外，加一組合電路，來產生 T 型正反器。
2. 在 T 型正反器外，加一組合電路，來產生 D 型正反器。

(提示：第一題，只要在輸入訊號 D 前面加入一組合電路，如圖一所示，即可完成。第二題同第一題的作法，可以參考圖二)

結報的內容需說明此組合電路為何，如何找到此電路，以及為何此組合電路可以達成以上目的。如果能在 Quartus II 上模擬，並顯示驗證的波型圖，另外加分。



圖一



圖二