

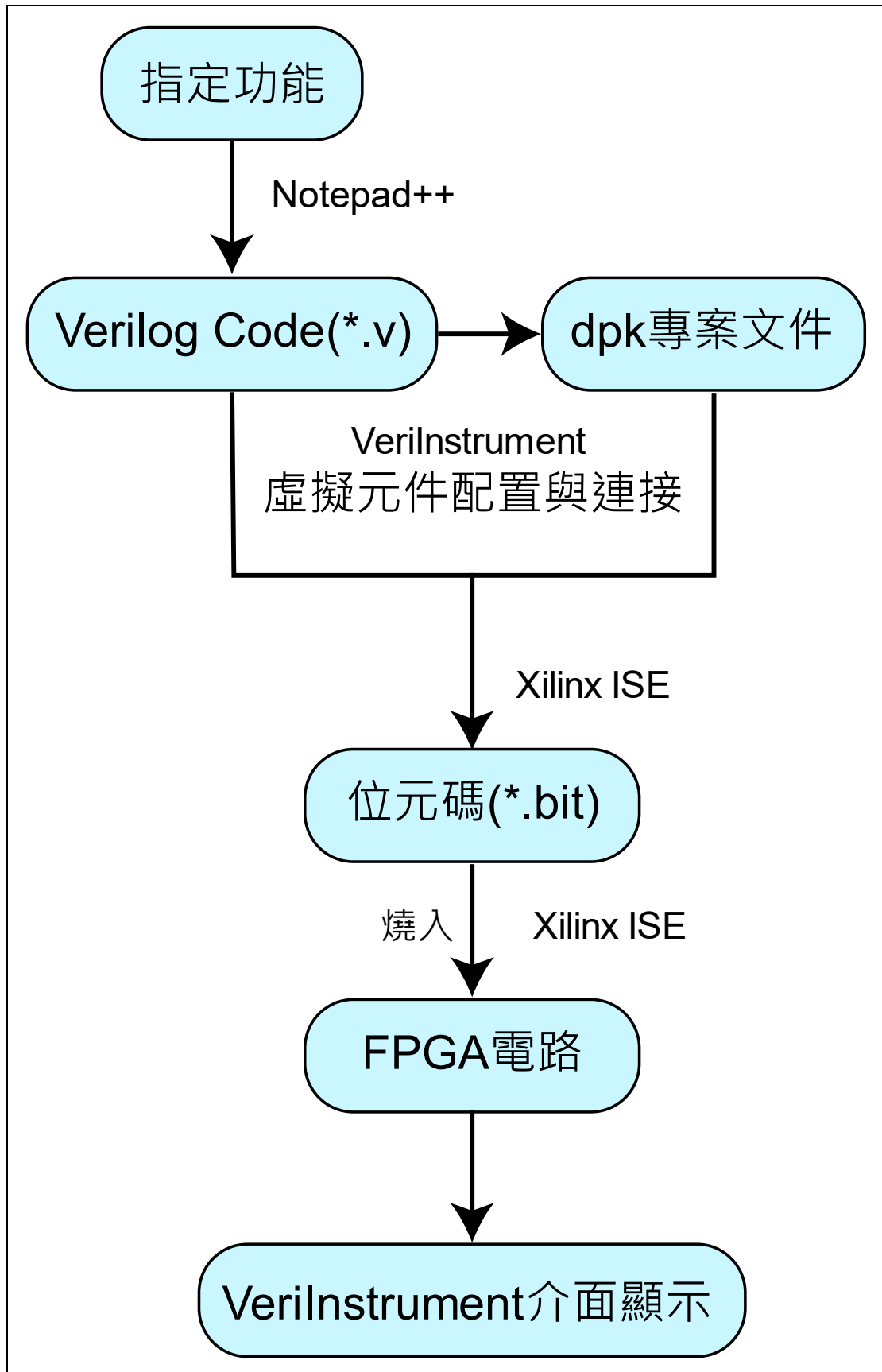
邏輯系統實驗

Lab 9

2021/05/06 (四)

第 1 組	
組員姓名	學號
陳旭祺	E24099059
張振杰	E24085034
何啟造	E34085337

● 實驗流程圖(同實驗 Lab7、Lab8)



● 實作題(一): D flip flop

1. Verilog Code

clock_div.v
<pre> module clock_div(clk_48MHZ, clk_1HZ); //reduce clock frequency input clk_48MHZ; output reg clk_1HZ; reg[31:0] counter_1HZ; always@(posedge clk_48MHZ) begin if(counter_1HZ == 32'd48000_000) counter_1HZ <= 32'd0; //return to 0 else counter_1HZ <= counter_1HZ + 32'b1; //plus 1 repeatedly if(counter_1HZ < 32'd24000_000) clk_1HZ <= 1'd1; else clk_1HZ <= 1'd0; end endmodule </pre>
d_ff.v
<pre> module d_ff(clock,reset,din_d,dout_q,dout_nq); input clock; input reset; input din_d; //D output dout_q; //Q_n output dout_nq; //Q_n+1 reg data; always@(posedge clock or posedge reset) begin if(reset) //reset data <= 1'b0; else begin </pre>

```

        if(din_d) //D==1
            data <= 1'b1; //Q_n+1=1
        else      //D==0
            data <= 1'b0; //Q_n+1=0
        end
    end

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

// D flip-flop truth table
// +-----+
// | CK | D | Q_n+1 |
// +=====+
// | 0 | x | Q_n |
// +-----+
// | 1 | 0 | 0 |
// +-----+
// | 1 | 1 | 1 |
// +-----+
// (transform of SR flip flop and D stands for Data)

```

SevenSeg.v

```

module SevenSeg(din,dout);
    input [3:0] din;
    output 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

```

lab9_1.v

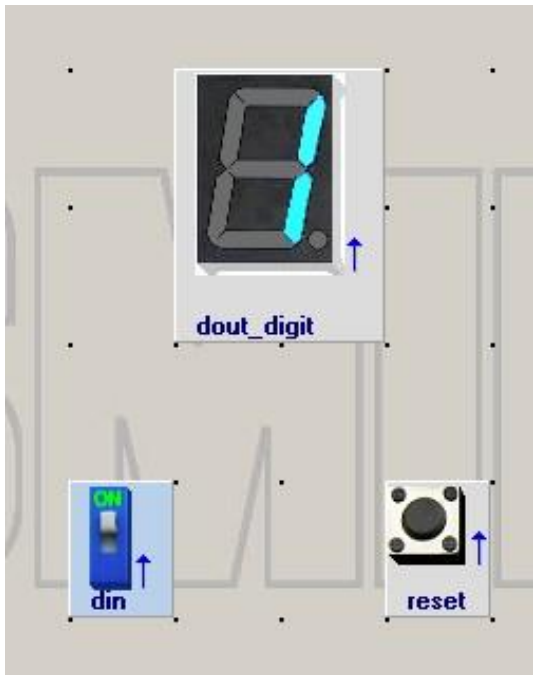
```

module lab9_1(clock,reset,din,dout_digit);
    input clock,reset,din;
    output[6:0] dout_digit;
    wire[3:0] value;
    wire clk_1HZ;
    //reduce clock frequency
    clock_div clock_div(
        .clk_48MHZ (clock),
        .clk_1HZ   (clk_1HZ)
    );
    //D-flip flop
    d_ff d_ff(
        .clock (clk_1HZ),
        .reset (reset),
        .din_d (din),
        .dout_q (value[0]),
        .dout_nq ()
    );

    //decode into Sevenseg Signal
    assign value[3:1] = 3'b000;
    SevenSeg SevenSeg(
        .din (value[3:0]),
        .dout (dout_digit)
    );
endmodule

```

2. FPGA result

圖片	說明
	<p>D flip-flop，輸出會和輸入相同。因為正緣觸發，在切換輸入的瞬間，需要等 clock 的正緣到時，輸出值才會變</p>

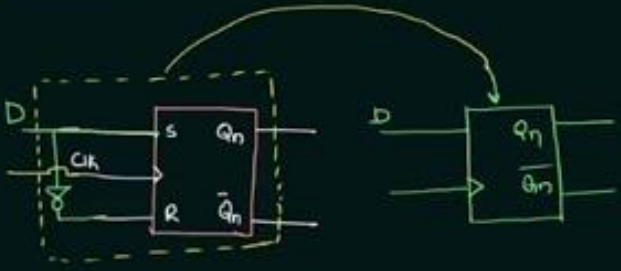
3. 硬體架構與功能解釋(以下 Youtube 影片說明)

D Flip Flop

T.T for SR Flip Flop:-

clk	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	invalid

Q_n } Memory



152
Digital Electronics

● 實作題(二): Johnson Counter

1. Verilog Code

`include "clock_div.v" "d_ff.v"	
johnson_counter.v	
<pre> 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 d_ffb0(.clock (clock), .reset (reset), .din_d (nq3), .dout_q (q0), .dout_nq ()); d_ff d_ffb1(.clock (clock), .reset (reset), .din_d (q0), .dout_q (q1), .dout_nq ()); d_ff d_ffb2(.clock (clock), .reset (reset), .din_d (q1), .dout_q (q2), .dout_nq ()); d_ff d_ffb3(.clock (clock), .reset (reset), .din_d (q2), </pre>	

```

        .dout_q (q3),
        .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

```

lab9_2.v

```

module lab9_2(clock,reset,led_a,led_b);
    input      clock,reset;
    output [3:0] led_a;
    output [7:0] led_b;
    wire       clock_1HZ;
    wire [3:0] q;
    wire [7:0] j;
    wire       reset;

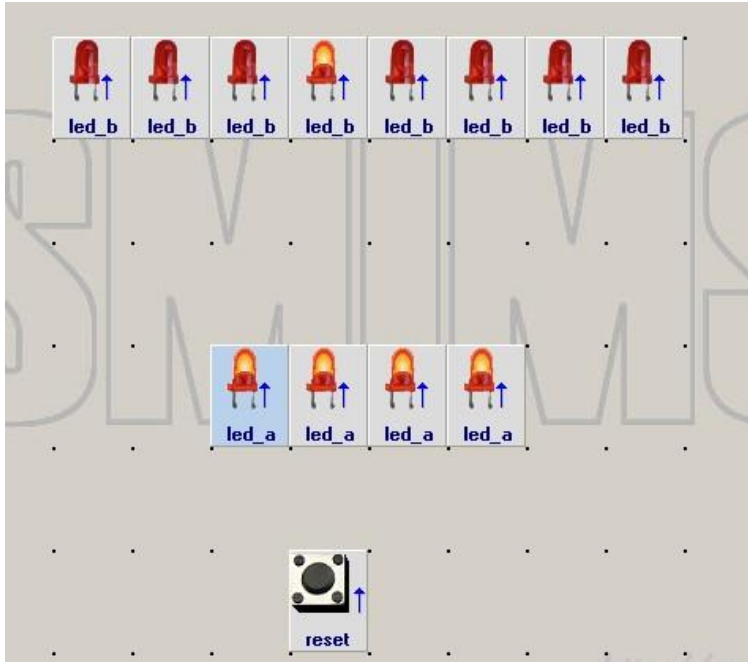
    clock_div clock_div(
        .clk_48MHZ (clock),
        .clk_1HZ   (clock_1HZ)
    );

    johnson_counter jc(
        .clock (clock_1HZ),
        .reset (reset),
        .dout_q (q),
        .dout_j (j)
    );

    assign led_a = q;
    assign led_b = j;
endmodule

```

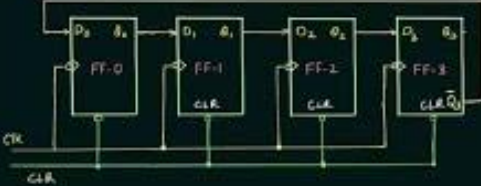

2. FPGA result


圖片	說明
 <p>The image shows the physical implementation of a Johnson Counter on an FPGA board. There are 8 red LEDs labeled 'led_b' and 4 red LEDs labeled 'led_a'. A 'reset' button is also visible. The LEDs are arranged in two rows: the top row has 8 LEDs, and the bottom row has 4 LEDs. The 'reset' button is located below the bottom row of LEDs.</p>	<p>Johnson Counter 由 4 個 D flip-flop 組成，每個 flip-flop 的輸入為前一個的輸出，4 個 flip-flop 對應 4 個 LED_A，4 個 LED_A 的訊號會依序出現和消失產生 8 種狀態，各對應 1 個 LED_B</p>

3. 硬體架構與功能解釋(以下 Youtube 影片說明)

Johnson's Counter

no of state = $2 \times \text{no. of df}$





192 Digital Electronics

● 實作題(三): T flip flop

1. Verilog Code

```

`include "clock_div.v" "SevenSeg.v"

t_ff.v

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 begin
            if(din_t)
                data <= ~data;
            else
                data <= data;
        end
    end
    assign dout_q = data;
    assign dout_nq = ~data;
endmodule

// T flip-flop truth table
// +-----+-----+
// | CK | T | Qn+1 |
// +=====+
// | 0 | x | Qn |
// +-----+
// | 1 | 0 | Qn |
// +-----+
// | 1 | 1 | Qn' |
// +-----+
//((trasform of JK flip flip and T stands for Toggle)

```

```

lab9_3.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;

    clock_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]),
        .dout_nq ( )
    );

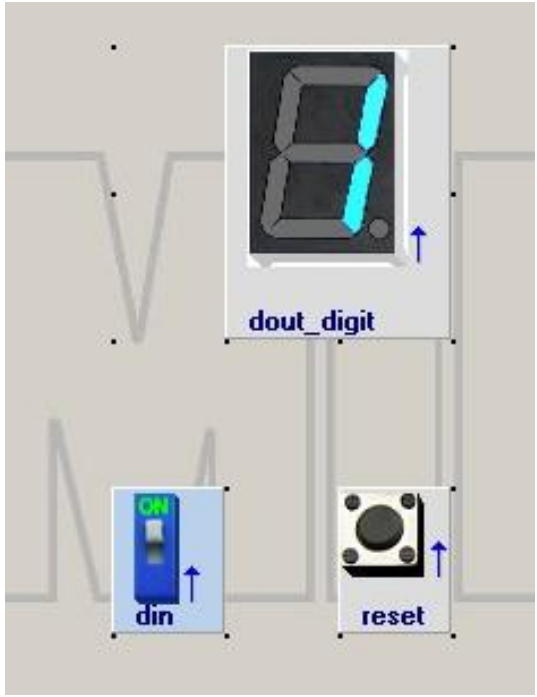
    assign value[3:1] = 3'b000;

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

endmodule

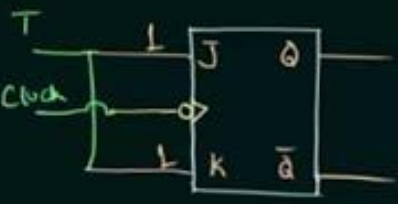
```

2. FPGA result

圖片	說明
 <p>The image shows the physical implementation of the T flip-flop circuit on an FPGA. A 7-segment display, labeled 'dout_digit', shows the digit '8'. Below it are two buttons: 'din' (data input) and 'reset'. Arrows indicate the signal flow from the buttons to the display.</p>	<p>此實驗要求為非同步的 T 型正反器，toggle 訊號為 1 時，T 型正反器的輸出會一直變，達到非同步的效果。而當 toggle 為 0 時，則輸出保持不變，達到 flip-flop 的記憶效果，另外還有一個 reset，當 reset 觸發，T flip-flop 歸零</p>

3. 硬體架構與功能解釋(以下 Youtube 影片說明)

T Flip Flop



The diagram shows a T Flip Flop circuit. It has a clock input, a T input, and two data inputs J and K. The outputs are Q and Q-bar. The T input is connected to the J and K inputs.

T.T. for T FF -

clk	T	Q _{n+1}
0	X	Q _n (memory)
1	0	Q _n (memory)
1	1	$\overline{Q_n}$ (toggling)

159
Digital Electronics

● 實作題(四): 4-bit Asynchronous Up Counter

1. Verilog Code

<code>`include "clock_div.v" "SevenSeg.v" "t_ff"</code>
<code>asyn_counter.v</code>
<pre> module asyn_counter(clock,reset,dout_q); input clock,reset; output [3:0]dout_q; wire q0,q1,q2,q3; wire nq0,nq1,nq2; t_ff diff_b0(.clock (clock), .reset (reset), .din_t (1'b1), .dout_q (q0), .dout_nq (nq0)); t_ff diff_b1(.clock (nq0), .reset (reset), .din_t (1'b1), .dout_q (q1), .dout_nq (nq1)); t_ff diff_b2(.clock (nq1), .reset (reset), .din_t (1'b1), .dout_q (q2), .dout_nq (nq2)); t_ff diff_b3(.clock (nq2), .reset (reset), .din_t (1'b1), .dout_q (q3), </pre>

```

        .dout_nq    () //unused
    );
    assign dout_q = {q3,q2,q1,q0};
endmodule

```

lab9_4.v

```

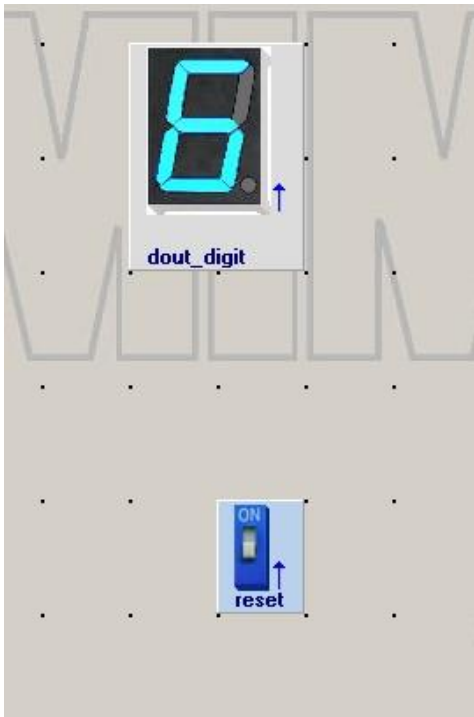
module lab9_4(clock,reset,dout_digit);
    input clock,reset;
    output[6:0] dout_digit;
    wire[3:0] value;
    wire clock_1HZ;

    clock_div clock_div(
        .clk_48MHZ (clock),
        .clk_1HZ   (clock_1HZ)
    );

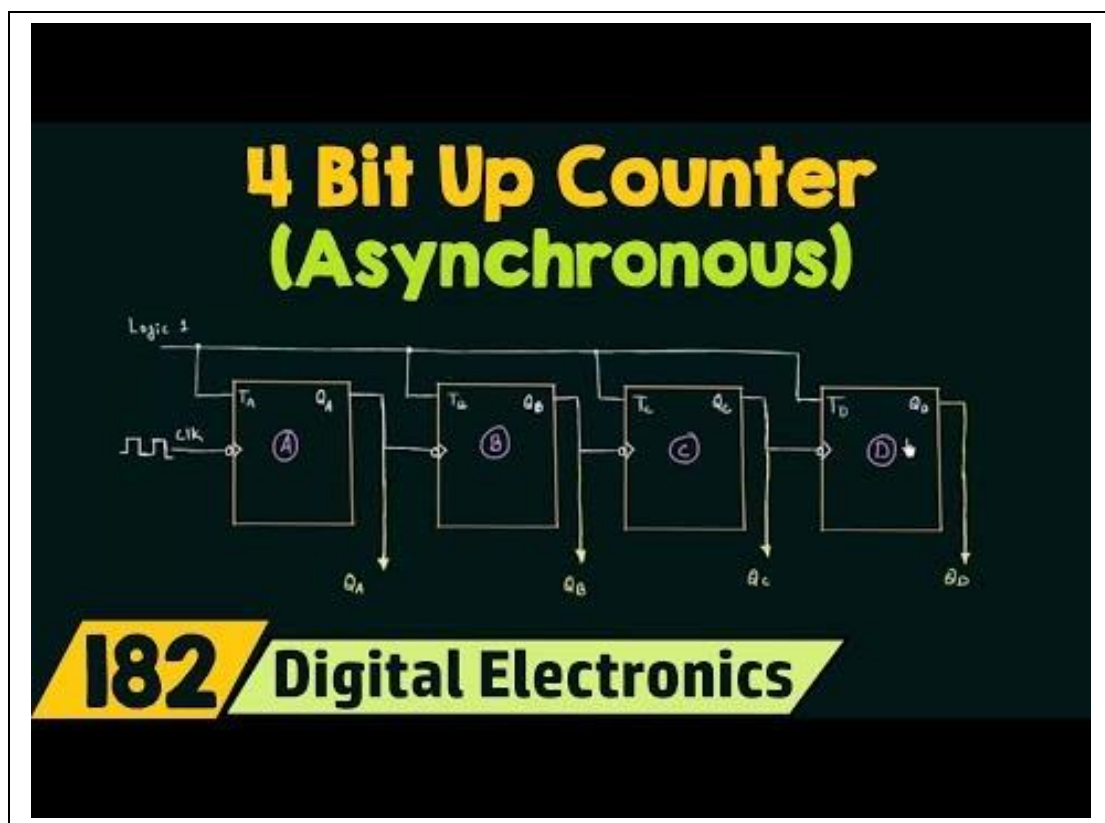
    asyn_counter ac(
        .clock      (clock_1HZ),
        .reset      (reset),
        .dout_q     (value)
    );
    //assign vlaue[3:1] = 3'b000;
    SevenSeg SevenSeg(
        .din      (value),
        .dout     (dout_digit)
    );
endmodule

```

2. FPGA result

圖片	說明
	<p>由 4 個 T flip-flop 組成， $T=1$ 時每個 flip-flop 的輸出一直變， 由 asyn_counter 去做連接，Q' 會成為 下一個 Clk 的輸入，因此每個 Clk 頻 率不相同。由 SevenSeg 作為最後的 dout。最後數字會一直變，由 0~F 依 序輸出</p>

3. 硬體架構與功能解釋(以下 Youtube 影片說明)



● 心得

1. 組員一 陳旭祺

這次實驗共有兩個部分，一是實作 D flip-flop，並以此 block 進一步實作 Johnson Counter；二是實作 T flip-flop，並以此 block 進一步實作 4-bit Asynchronous Up Counter。

module	功能
clock_div.v	降低 Clk 的頻率為一秒一次，如此我們才能肉眼觀察到變化
SevenSeg.v	將訊號解碼出七段顯示器的訊號並顯示
d_ff.v	D flip-flop 是 RS flip-flop 之變形，用一個訊號接 S，另一個由原本訊號反向再接 R，如此避免原本 S=1、R=1 無效訊號，又能實現 RS flip-flop 所有功能
johnson_counter.v	將使用 4 個 d_ff.v，利用 D flip flop 輸入 0 下一個 state 是 0；輸入 1 下一個 state 是 1，從一開始歸 0 狀態，每過一個 Cycle 就一直擴增 1 的訊號，直到最後一個 D flip flop 的 Q'=0 回授到一開始的輸入又變回 0，如此每過一個 Cycle 就一直擴增 0 的訊號，直到回到全 0 的狀態，共經過 8 個 state
lab9_2.v	需另外新增 led_b [7:0]，讓訊號解碼出我們想要的 LED1-8
t_ff.v	T flip-flop 是 JK flip-flop 之變形，輸入 0，hold 住原本的值；輸入 1，反轉原本的值，即為 T 代表 toggle 之原意
asyn_counter.v	Up 或 down 只是單純代表訊號是輸出 Q 或是 Q'，記得要初始化全 0 才能正常運作，第一個 T flip flop 會在 clk 一起來，輸出值變為原本值的倒數，如此就變為原本 clk 的倍；第二級 T flip flop 吃的 clk 是前一級的輸出 Q，因此就變為上一級 T flip flop 輸出值的 2 倍，也就是原本 clk 的 4 倍，如此不斷下去，實現 Clock Divider 的功能。

詳細圖示解說請參考以上影片會比較清楚，主要是學懂 Clk 一來會如何變化與接線、回授，如此就了解 johnson_counter.v 與 asyn_counter.v 的工作原理，而在本次實驗我也實際在各個 module 上接線，搞清楚各個 module 的功能，並做 FPGA 燒入與 VerilInstrument 虛擬裝置的驗證。

2.組員二 張振杰

這次實驗的正反器主要訊號輸入是時間訊號和一個自控訊號，而我們可以通過不同的 flipflop 結構來觀察不同的訊號輸出，就如實作 2 的 Johnson Counter 一排 LED 表示的是每一種情況，而另一排 LED 則是表示每種情況什麼輸出會有值。而其理論也值得深究。

3.組員三 何啟造

這次的實驗是做關於正反器以及除頻電路的，可讓數字、LED 燈不停地閃爍跳動。在實驗中花最多時間的部分還是去理解程式碼的邏輯，而只要一搞懂邏輯實驗就不會難了。