LAB - 08

陳培殷老師

國立成功大學　資訊工程系

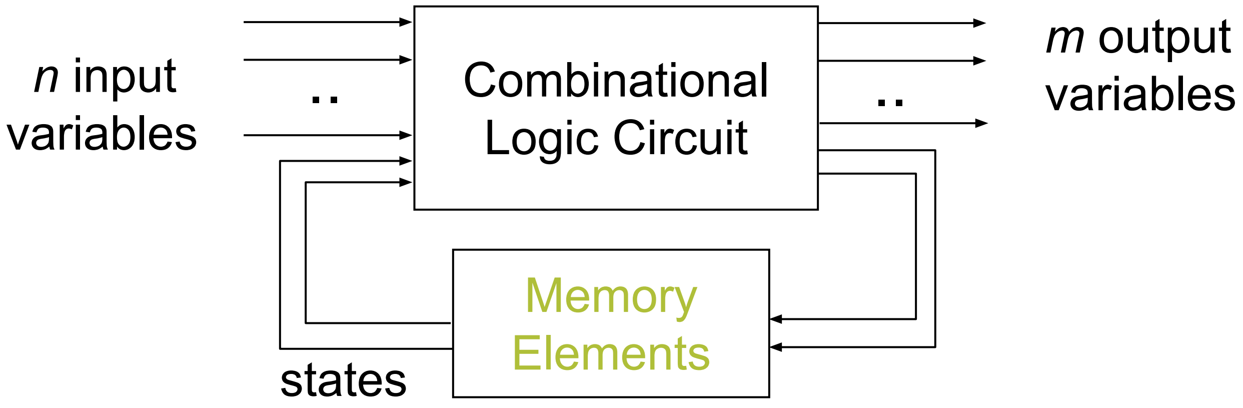
YOUTUBE

HDL Part 6 上(影片 9:50)

HDL Part 6 下(影片 11:49)

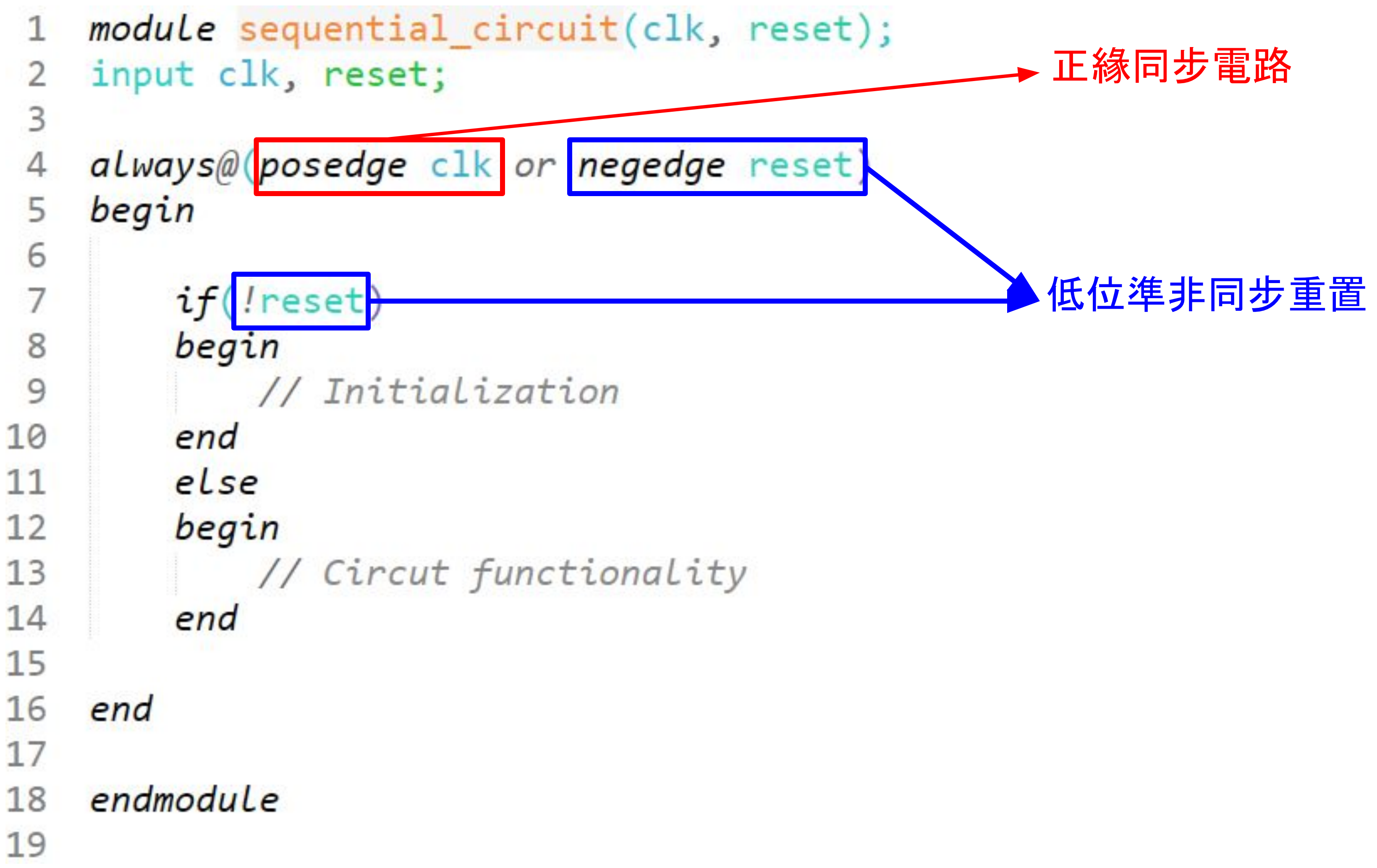
Sequential Circuit (1/3)

A sequential circuit is a system whose outputs at any time are determined *from the present combination of inputs and the previous inputs or outputs*.



- Sequential components contain memory elements
- The output values of sequential components depend on the input values and the values stored in the memory elements

Sequential Circuit (2/3)



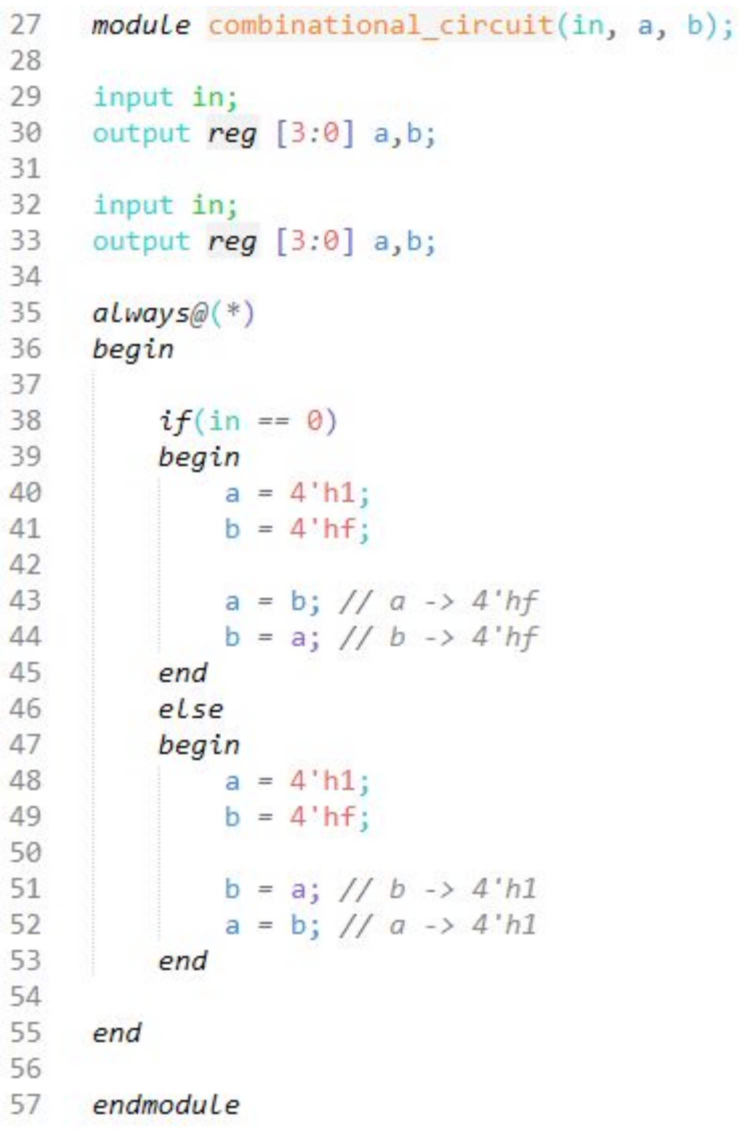
```
module sequential_circuit(clk, reset);
input clk, reset;

always@(posedge clk or negedge reset)
begin

    if(!reset)
    begin
        // Initialization
    end
    else
    begin
        // Circut functionality
    end

end

endmodule

```

正緣同步電路

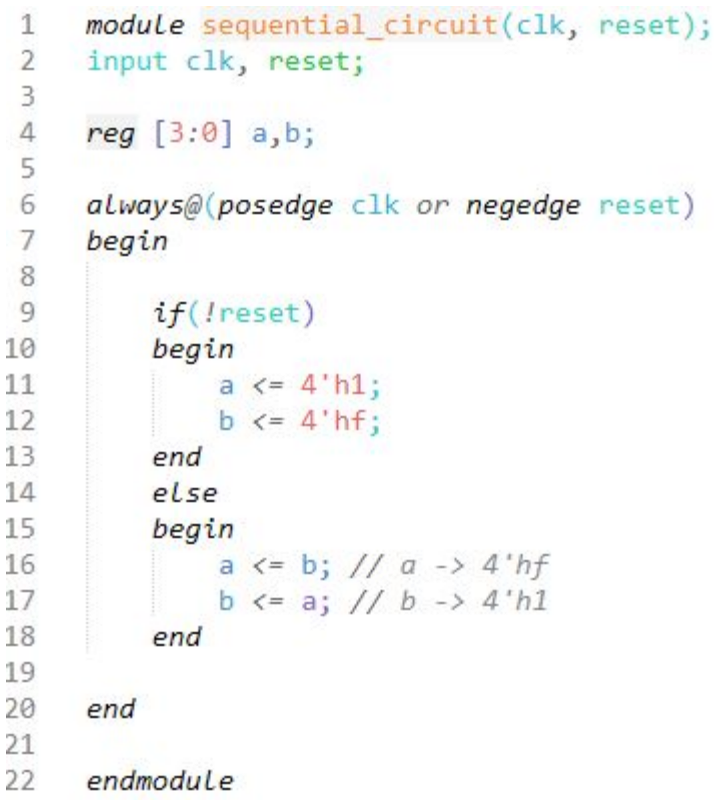
低位準非同步重置

Sequential Circuit (3/3)

- **Blocking statement vs nonblocking statement**



```
module combinational_circuit(in, a, b);

input in;
output reg [3:0] a,b;

input in;
output reg [3:0] a,b;

always@(*)
begin

    if(in == 0)
    begin
        a = 4'h1;
        b = 4'hf;

        a = b; // a -> 4'hf
        b = a; // b -> 4'hf
    end
    else
    begin
        a = 4'h1;
        b = 4'hf;

        b = a; // b -> 4'h1
        a = b; // a -> 4'h1
    end

end

endmodule
```



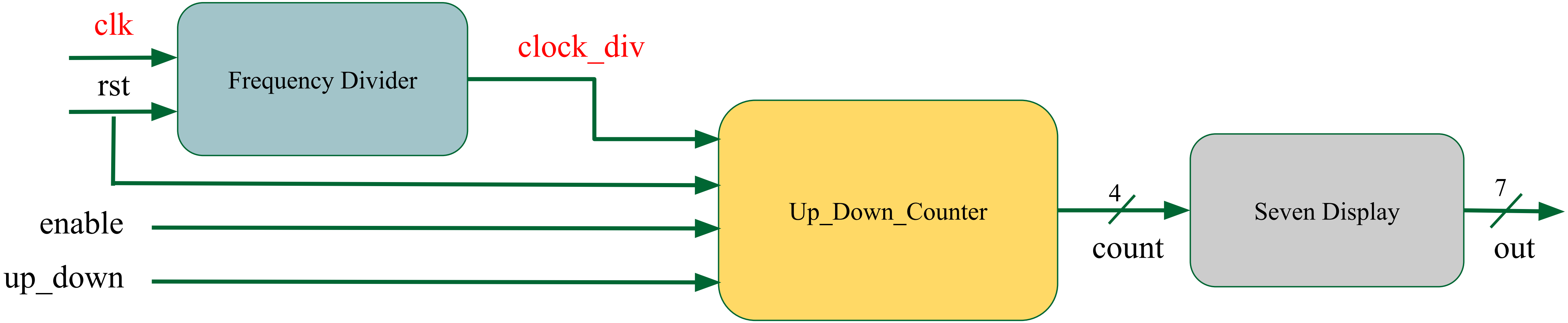
```
module sequential_circuit(clk, reset);
input clk, reset;

reg [3:0] a,b;

always@(posedge clk or negedge reset)
begin

    if(!reset)
    begin
        a <= 4'h1;
        b <= 4'hf;
    end
    else
    begin
        a <= b; // a -> 4'hf
        b <= a; // b -> 4'h1
    end

end

endmodule
```

Lab I (1/3)

- 請設計一個具備下列功能的計數器:
  - 正緣同步電路，低位準非同步 重置
  - 當reset訊號為0，將計數器歸零，輸出維持 0
  - 當reset訊號為1，且enable訊號為1:
    1) up_down = 0，每秒計數減1
       - EX:0 -> F -> E -> …… -> 2 -> 1 -> 0 -> F ……
    2) up_down = 1，每秒計數加1
       - EX:0 -> 1 -> 2 -> …… -> E -> F -> 0 -> 1 ……
  - 當enable訊號為0時，保持目前數 值不變，暫停計數。
  - FPGA版之clock頻率為50MHz，需藉由除頻器將 clock訊號降為1Hz
    - 實現方式為透過一個計數器，計算經過幾個 時脈正緣，當計數到50 × 10^6 即代表經過一秒

Lab I (2/3)

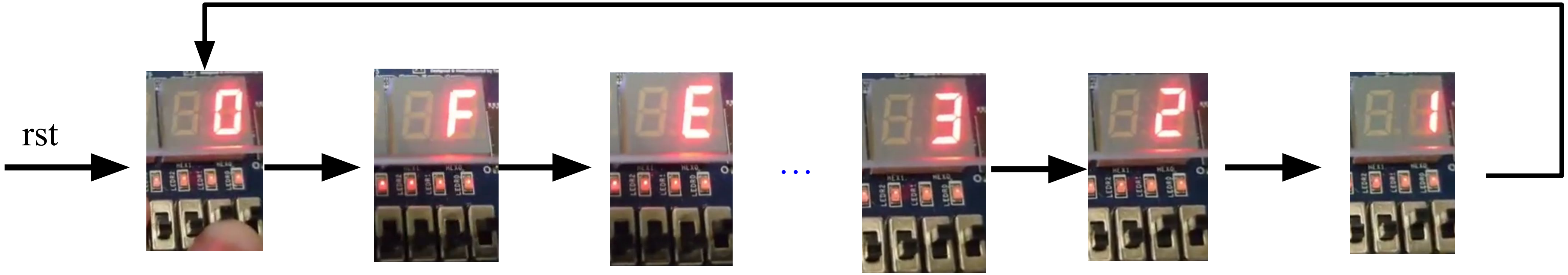
- 請將計數的數 值顯示於七段顯示器
- 系統架構圖請參考下方
  - ❑ Input : clk(MAX10_CLK1_50) , rst(SW0), up_down(SW1), enable(SW2)
  - ❑ Output : out(7 bits, HEX06~HEX00)
- 可使用structural description設計，其中，除頻器及計數模組為循序電路，七段顯示器控制模組則為組合電路



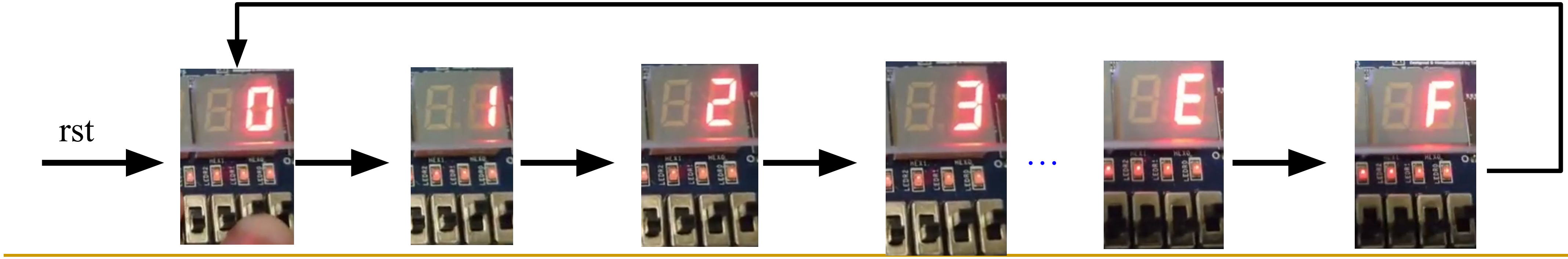
Lab I (3/3)

- **輸出範例 (rst=1, enable=1)**

1) up_down = 0



2) up_down = 1



Lab I (3/3)

- 狀態範例：

rst(SW0)	up_down(SW1)	enable(SW2)	輸出結果
0	0	0	維持輸出0
0	0	1	維持輸出0
0	1	0	維持輸出0
0	1	1	維持輸出0
1	0	0	暫停在當前狀態(ex. 下數狀態。如果數到3，就停在3)
1	0	1	倒數(0->f->e...->1->0->f)
1	1	0	暫停在當前狀態(ex. 上數狀態。如果數到5，就停在5)
1	1	1	上數(0->1->2....->e->f->0)

Lab - Hint(1/3)

- Frequency Divider (sequential circuit)
  - 將clock頻率從50MHz降為1Hz
- Counter (sequential circuit)
  - 加/減計數器模組，支援 enable 控制暫停功能
- Seven Display (combinational circuit)
  - 將Counter數值轉為七段顯示器控制訊號

- Module呼叫範例 (Structural description)
  - FrequencyDivider u_FreqDiv (.clk(clk), .rst(rst), .clk_div(clk_div));
  - Counter u_counter(.clk(clk_div), .rst(rst), .up_down(up_down), .enable(enable), .count(count));
  - SevenDisplay u_display(.in(count), .out(out));

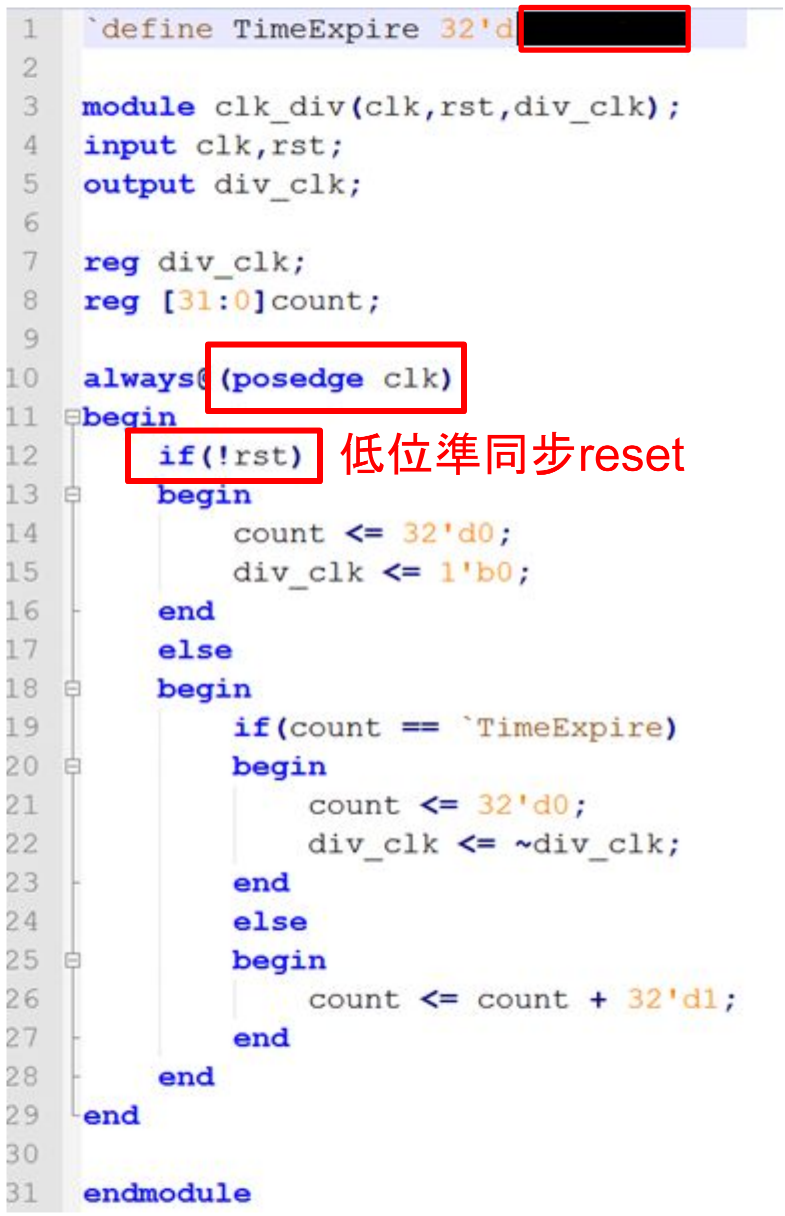
```
module_name unit_name(.port_name(signal_name),.port_name(signal_name)……)
```

Lab – Hint(2/3)

- **除頻器範例：**

  要達到產出1Hz的div_clk，

  要每0.5秒flip一次div_clk。



```verilog
`define TimeExpire 32'd[redacted]

module clk_div(clk,rst,div_clk);
input clk,rst;
output div_clk;

reg div_clk;
reg [31:0]count;

always@(posedge clk)
begin
    if(!rst)
    begin
        count <= 32'd0;
        div_clk <= 1'b0;
    end
    else
    begin
        if(count == `TimeExpire)
        begin
            count <= 32'd0;
            div_clk <= ~div_clk;
        end
        else
        begin
            count <= count + 32'd1;
        end
    end
end

endmodule
```

低位準同步reset

Lab - Hint(3/3)

```
module SevenDisplay(in,out);

input [3:0] in;
output [6:0] out;

reg [6:0] out;
```

```
always@(in)
begin
    case(in)
        4'd0: out = 7'b1000000;
        4'd1: out = 7'b1111001;
        4'd2: out = 7'b0100100;
        4'd3: out = 7'b0110000;
        4'd4: out = 7'b0011001;
        4'd5: out = 7'b0010010;
        4'd6: out = 7'b0000010;
        4'd7: out = 7'b1111000;
        4'd8: out = 7'b0000000;
        4'd9: out = 7'b0010000;
        4'd10: out = 7'b0001000;
        4'd11: out = 7'b0000011;
        4'd12: out = 7'b1000110;
        4'd13: out = 7'b0100001;
        4'd14: out = 7'b0000110;
        default: out = 7'b0001110;
    endcase
end
endmodule
```

```
always@(in)
begin
    case(in)
        4'd0: out = 7'b1000000;
        4'd1: out = 7'b1111001;
        4'd2: out = 7'b0100100;
        4'd3: out = 7'b0110000;
        4'd4: out = 7'b0011001;
        4'd5: out = 7'b0010010;
        4'd6: out = 7'b0000010;
        4'd7: out = 7'b1111000;
        4'd8: out = 7'b0000000;
        4'd9: out = 7'b0010000;
        4'd10: out = 7'b0001000;
        4'd11: out = 7'b0000011;
        4'd12: out = 7'b1000110;
        4'd13: out = 7'b0100001;
        4'd14: out = 7'b0000110;
        default: out = 7'b0001110;
    endcase
end
endmodule
```

Notice

- 請勿命名中文或數字開頭的資料夾

- Device family 請確認與 FPGA Chip 符合 (**10M50DAF484C7G**)

- Top module name & Top entity name需要一致

- 在組合電路中，case、if…else…若沒有寫滿，合成後會產生latch