



*Department of Computer Science and Information Engineering*  
*National Cheng Kung University*

# **LAB - 08**

陳培殷老師  
國立成功大學 資訊工程系

*Digital Integrated Circuit Design Laboratory*



---

# YOUTUBE

[HDL Part 6 上\(影片 9:50\)](#)

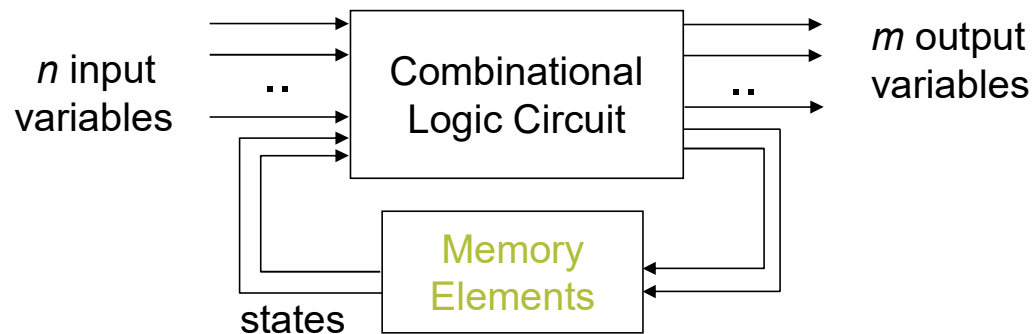
[HDL Part 6 下\(影片 11:49\)](#)

---

---

## 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)

```
1 module sequential_circuit(clk, reset);
2   input clk, reset;
3
4   always@(posedge clk or negedge reset)
5   begin
6
7       if(!reset)
8       begin
9           // Initialization
10        end
11        else
12        begin
13            // Circuit functionality
14        end
15    end
16 end
17
18 endmodule
19
```

正緣同步電路

低位準非同步重置

# Sequential Circuit (3/3)

## ■ Blocking statement vs nonblocking statement

```
27 module combinational_circuit(in, a, b);
28
29 input in;
30 output reg [3:0] a,b;
31
32 input in;
33 output reg [3:0] a,b;
34
35 always@(*)
36 begin
37     if(in == 0)
38     begin
39         a = 4'h1;
40         b = 4'hf;
41
42         a = b; // a -> 4'hf
43         b = a; // b -> 4'hf
44     end
45     else
46     begin
47         a = 4'h1;
48         b = 4'hf;
49
50         b = a; // b -> 4'h1
51         a = b; // a -> 4'h1
52     end
53 end
54
55 end
56
57 endmodule
```

```
1 module sequential_circuit(clk, reset);
2 input clk, reset;
3
4 reg [3:0] a,b;
5
6 always@(posedge clk or negedge reset)
7 begin
8
9     if(!reset)
10    begin
11        a <= 4'h1;
12        b <= 4'hf;
13    end
14    else
15    begin
16        a <= b; // a -> 4'hf
17        b <= a; // b -> 4'h1
18    end
19
20 end
21
22 endmodule
```

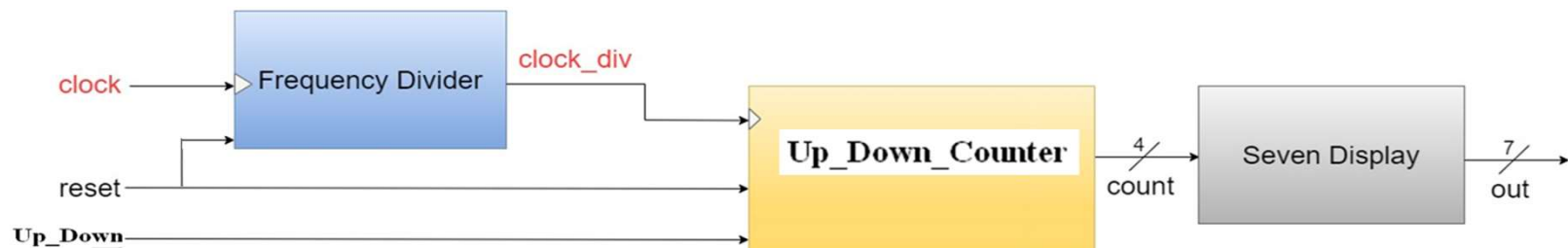
---

## 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，將目前狀態暫停，輸出維持0
    - 當reset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 .....
    - 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 $50 \times 10^6$  即代表經過一秒
-

## Lab I (2/3)

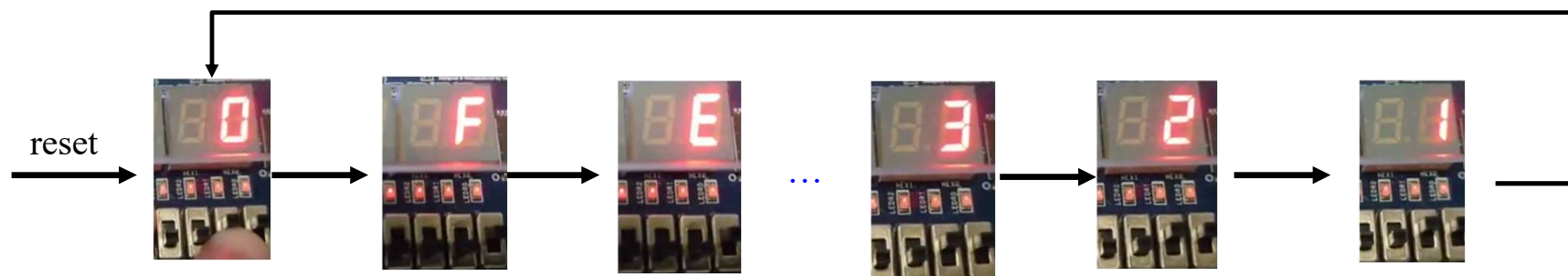
- 請將計數的數值顯示於七段顯示器
- 系統架構圖請參考下方
  - Input : clock(**MAX10\_CLK1\_50**) , reset(SW0), Up\_Down(SW1)
  - Output : out(7 bits, HEX06~HEX00)
- 可使用**structural description**設計，其中，除頻器及計數模組為循序電路，七段顯示器控制模組則為組合電路



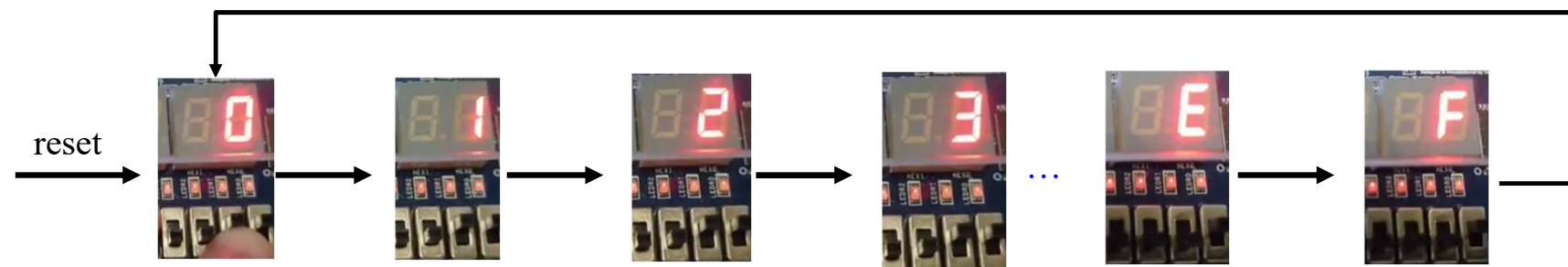
# Lab I (3/3)

## ■ 輸出範例

1) Up\_Down = 0



2) Up\_Down = 1





## Lab - Hint(1/3)

- **Frequency Divider (sequential circuit)**
  - 將clock頻率從50MHz降為1Hz
- **Counter (sequential circuit)**
  - 每秒進行計數
- **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(clock), .reset(reset), .clk\_div(clock\_div));
  - **Counter** u\_counter(.clk(clock\_div), .reset(reset), .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。

```
1  `define TimeExpire 32'd[REDACTED]
2
3  module clk_div(clk,rst,div_clk);
4  input clk,rst;
5  output div_clk;
6
7  reg div_clk;
8  reg [31:0]count;
9
10 always@(posedge clk)
11 begin
12     if(!rst) 低位準同步reset
13     begin
14         count <= 32'd0;
15         div_clk <= 1'b0;
16     end
17     else
18     begin
19         if(count == `TimeExpire)
20         begin
21             count <= 32'd0;
22             div_clk <= ~div_clk;
23         end
24         else
25         begin
26             count <= count + 32'd1;
27         end
28     end
29 end
30
31 endmodule
```

## 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
```

---

## 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