



Lab3

7-Seg_Display x Switch

2022.04.20



指導教授：范志鵬

主講人：方藝橋



Outline

- ❖ 實驗介紹
- ❖ 七段顯示器介紹
- ❖ Lab3 作業

Lab3

❖ 此次實驗使用到的 I/O

- ◆ 以八個switch作為輸入
- ◆ 以四顆七段顯示器作為輸出

❖ 功能

- ◆ 學會同時顯示多個「七段顯示器」
- ◆ 將八個switch表示的二進位數字，轉為十進位數字，並顯示在七段顯示器上。
- ◆ Ex: 8'b0001_0011 → 19



Verilog

設定單一cycle長度

```
`define CYCLE 100000 // 單一cycle 長度
```

```
// 模組名稱
```

```
module seg7_switch(  
    input clk,  
    input rst_n,  
    input [7:0] switch,  
    output [7:0] seg7,  
    output [3:0] seg7_sel  
);
```

```
// 暫存器宣告
```

```
reg [7:0] seg7;  
reg [3:0] seg7_sel;  
reg [3:0] seg7_temp [0:3];  
reg [1:0] seg7_count;  
  
reg [29:0] count;  
wire d_clk;
```

除頻

```
// 除頻
```

```
always @(posedge clk or negedge rst_n) begin  
    if(!rst_n)  
        count <= 0;  
    else if (count >= `CYCLE)  
        count <= 0;  
    else  
        count <= count + 1;  
end  
assign d_clk = count > (`CYCLE/2) ? 0 : 1;
```

```
// switch 二進位轉十進位
```

```
always @(posedge clk or negedge rst_n) begin  
    if(!rst_n) begin  
        seg7_temp[0] <= 0;  
        seg7_temp[1] <= 0;  
        seg7_temp[2] <= 0;  
        seg7_temp[3] <= 0;  
    end  
    else begin  
        seg7_temp[3] <= 0;  
        seg7_temp[2] <= (switch % 1000) / 100;  
        seg7_temp[1] <= (switch % 100) / 10;  
        seg7_temp[0] <= switch % 10;  
    end  
end
```

Binary
To
Decimal

```
// 七段顯示器顯示
```

```
always @(posedge d_clk or negedge rst_n) begin  
    if(!rst_n) begin  
        seg7_count <= 0;  
    end  
    else begin  
        seg7_count <= seg7_count + 1;  
    end  
end
```

七段顯示器顯示

```
always @(posedge d_clk or negedge rst_n) begin  
    if(!rst_n) begin  
        seg7_sel <= 0;  
        seg7 <= 0;  
    end  
    else begin  
        case(seg7_count)  
            0: seg7_sel <= 4'b0001;  
            1: seg7_sel <= 4'b0010;  
            2: seg7_sel <= 4'b0100;  
            3: seg7_sel <= 4'b1000;  
        endcase  
        case(seg7_temp[seg7_count])  
            0: seg7 <= 8'b0011_1111;  
            1: seg7 <= 8'b0000_0110;  
            2: seg7 <= 8'b0101_1011;  
            3: seg7 <= 8'b0100_1111;  
            4: seg7 <= 8'b0110_0110;  
            5: seg7 <= 8'b0110_1101;  
            6: seg7 <= 8'b0111_1101;  
            7: seg7 <= 8'b0000_0111;  
            8: seg7 <= 8'b0111_1111;  
            9: seg7 <= 8'b0110_1111;  
        endcase  
    end  
end  
endmodule
```

除頻 ver.2

❖ 設定一計數器，並觀察計數器是否超過 CYCLE一半，若超過，則除頻訊號為(1)，反之則為(0)

❖ EX:

- ◆ `define CYCLE 100
- ◆ assign d_clk = count > (`CYCLE/2) ? 0 : 1;

<NOTE> 「三元運算子 ? : 」使用方法
assign 輸出 = (條件) ? 符合條件的值 : 不符合條件的值;

```
//除頻
always @(posedge clk or negedge rst_n)begin
    if(!rst_n)
        count <= 0;
    else if (count >= `CYCLE)
        count <= 0;
    else
        count <= count + 1;
end
assign d_clk = count > (`CYCLE/2) ? 0 : 1;
```

Cycle	0	1	2	...	49	50	51	52	...	99	100	101
count	0	1	2	...	49	50	51	52	...	99	100	0
d_clk												

Binary to Decimal

❖ 將二進位數字轉為十進位數字

- ◆ seg7_temp[3]：千位
- ◆ seg7_temp[2]：百位
- ◆ seg7_temp[1]：十位
- ◆ seg7_temp[0]：個位

❖ 藉由switch輸入一個8bits二進位數字，並經過取餘數及除法，取得每一位數



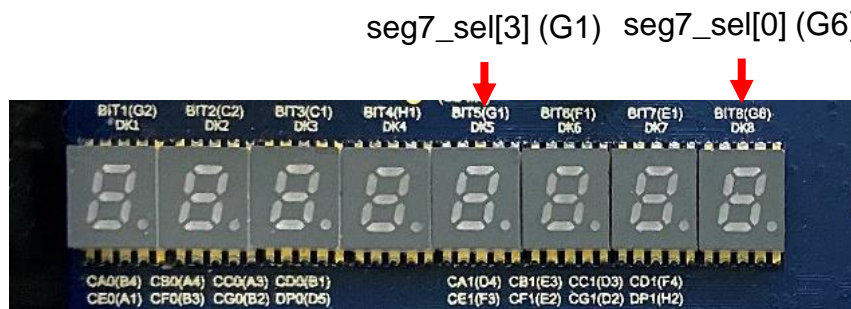
```
reg [3:0] seg7_temp [0:3];
```

```
//switch 二進位轉十進位
always @(posedge clk or negedge rst_n)begin
    if(!rst_n)begin
        seg7_temp[0] <= 0;
        seg7_temp[1] <= 0;
        seg7_temp[2] <= 0;
        seg7_temp[3] <= 0;
    end
    else begin
        seg7_temp[3] <= 0;
        seg7_temp[2] <= (switch % 1000) / 100;
        seg7_temp[1] <= (switch % 100) / 10;
        seg7_temp[0] <= switch % 10;
    end
end
```

Ex: 8'b0001_0011 → 19

Multiple 7-Seg Display

- ❖ 透過「掃描」實現多七段顯示器顯示
- ❖ 設定 2bits seg7_count計數器，0~3不停計數
- ❖ 一方面，輪流顯示每一個七段顯示器
- ❖ 另一方面，將對應的段選訊號輸出至 [seg7]



```
//七段顯示器顯示
always @(posedge d_clk or negedge rst_n)begin
    if(!rst_n)begin
        seg7_count <= 0;
    end
    else begin
        seg7_count <= seg7_count + 1;
    end
end

always @(posedge d_clk or negedge rst_n)begin
    if(!rst_n)begin
        seg7_sel <= 0;
        seg7 <= 0;
    end
    else begin
        case(seg7_count)
            0: seg7_sel <= 4'b0001;
            1: seg7_sel <= 4'b0010;
            2: seg7_sel <= 4'b0100;
            3: seg7_sel <= 4'b1000;
        endcase

        case(seg7_temp[seg7_count])
            0: seg7 <= 8'b0011_1111;
            1: seg7 <= 8'b0000_0110;
            2: seg7 <= 8'b0101_1011;
            3: seg7 <= 8'b0100_1111;
            4: seg7 <= 8'b0110_0110;
            5: seg7 <= 8'b0110_1101;
            6: seg7 <= 8'b0111_1101;
            7: seg7 <= 8'b0000_0111;
            8: seg7 <= 8'b0111_1111;
            9: seg7 <= 8'b0110_1111;
        endcase
    end
end
```

Lab3 constraint file

名稱	電路圖訊號	FPGA IO PIN
時鐘接腳	SYS_CLK	P17
名稱	電路圖訊號	FPGA IO PIN
重置接腳	FPGA_RESET	P15

段選信號

名稱	電路圖訊號	FPGA IO PIN
A0	LED0_CA	B4
B0	LED0_CB	A4
C0	LED0_CC	A3
D0	LED0_CD	B1
E0	LED0_CE	A1
F0	LED0_CF	B3
G0	LED0_CG	B2
DP0	LED0_DP	D5
A1	LED1_CA	D4
B1	LED1_CB	E3
C1	LED1_CC	D3
D1	LED1_CD	F4
E1	LED1_CE	F3
F1	LED1_CF	E2
G1	LED1_CG	D2
DP1	LED1_DP	H2

片選信號

DN0_K1	LED_BIT1	G2
DN0_K2	LED_BIT2	C2
DN0_K3	LED_BIT3	C1
DN0_K4	LED_BIT4	H1
DN1_K1	LED_BIT5	G1
DN1_K2	LED_BIT6	F1
DN1_K3	LED_BIT7	E1
DN1_K4	LED_BIT8	G6

Switch

SW8	SW_DIP0	U3
	SW_DIP1	U2
	SW_DIP2	V2
	SW_DIP3	V5
	SW_DIP4	V4
	SW_DIP5	R3
	SW_DIP6	T3
	SW_DIP7	T5

Lab3 作業(下午班)

❖ 將Lab3稍作修改，設計功能如下：

- ◆ 將前4bit作為2's補數的輸入，輸出為有號十進制數字
- ◆ 將後4bit作為2進制無號數，顯示十進制數字

★ Ex:

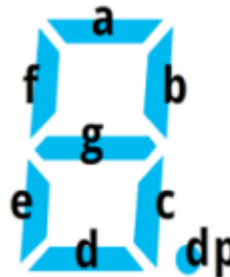
0011_1001，顯示

0	3	0	9
-	8	1	5
-	1	0	1

1000_1111，顯示

1111_0001，顯示

- ◆ 負號以片段“g”表示
- ◆ 未用到的七段顯示器，不亮



2's complement	Signed Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1

Lab3 作業(晚上班)

❖ 將Lab3稍作修改，設計功能如下：

- ◆ 輸入為8bit二進制無號數，輸出為十進制科學記號。

◆ Ex:

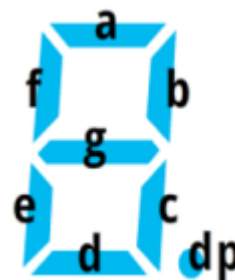
1111_1111，十進制為255= 2.55×10^2 ，顯示

0000_1111，十進制為15= 1.5×10^1 ，顯示

0000_0001，十進制為1= 1.00×10^0 ，顯示

2.	5	5	2
1.	5	0	1
1.	0	0	0

- ◆ 小數點以片段“dp”表示
- ◆ 未用到的七段顯示器，**不亮**





Thank you for your attention!

