

Lab4

Register x Wire

2022.04.27

指導教授：范志鵬

主講人：王星耀



Outline

- ❖ Register、Wire 的使用時機介紹
- ❖ Lab4作業

Reg vs. Wire

❖ Register ? Wire ?

```
3
4 //模組名稱
5 module seg7_switch(
6     input clk,
7     input rst_n,
8     input [7:0] switch,
9     output [7:0] seg7,
10    output [3:0] seg7_sel
11);
12
13 //暫存器宣告
14 reg [7:0] seg7;
15 reg [3:0] seg7_sel;
16 reg [7:0] seg7_temp [0:31];
17 reg [ Warning: Redeclaration of ansi port seg7_sel is not allowed
18
19 reg [29:0] count;
20 wire d_clk;
```

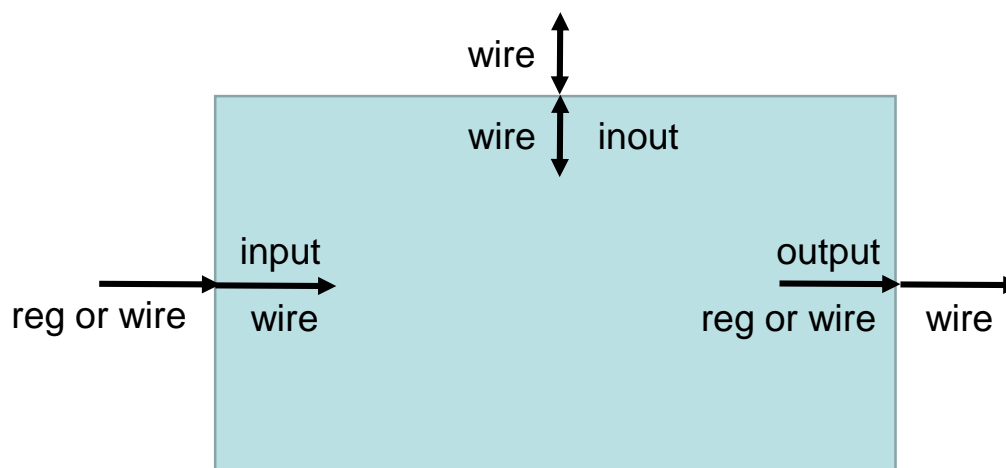
Reg vs. Wire

- ❖ 大致上來說，wire 和 reg 都類似於 C/C++ 的變數，但若此變數要放在 **always block** 內，該變數就須使用 reg，在 **always block** 之外，則使用 wire。
- ❖ 使用 wire 時，須搭配 **assign**，reg 則不必。
- ❖ input、output、inout 預設值都是 wire。
- ❖ 在 Verilog 中使用 reg，並不表示合成後就是暫存器(register)。若在組合電路中使用 reg，合成後仍只是 wire，唯有在循序電路中使用 reg，合成後才會以 flip-flop 形式表示成 register。
 - ◆ Ex:

```
always @(a or b)begin
    c = a & b;
end
```

Reg vs. Wire

- ❖ 輸入輸出應該使用reg還是wire呢？
 - ◆ 因為無法確定上一級為暫存器輸出還是組合邏輯輸出，所以以wire作為輸入。
 - ◆ 而輸出信號則是由自己決定以暫存器輸出或是組合邏輯輸出，reg或是wire都可以。



Reg vs. Wire

❖ 那應該如何編寫 Verilog code 呢?

◆ 因為預設皆為wire，在此示範將輸出改為reg

◆ 常見有兩種方法

✦ 另外設定為reg

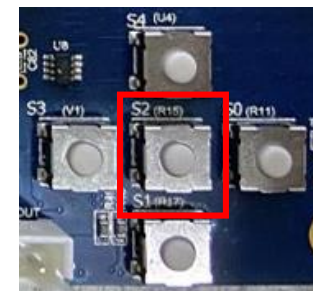
```
module top(  
    input a,  
    input b,  
    output c  
)  
    reg c;
```

✦ 在宣告時直接設為reg

```
module top(  
    input a,  
    input b,  
    output reg c  
)
```

Lab4 作業(下午班)

- ❖ 合併Lab1.2.3所學到的東西，設計功能如下：
 - ◆ 以4個七段顯示器作為輸出，顯示出貪吃蛇跑八字形
 - ◆ 以SW0(R1)作為輸入，未上撥為向右，上撥為向左
 - ◆ 以中間按鍵S2(R15)作為輸入，按下時，提高貪吃蛇的速度

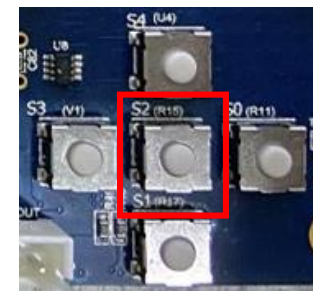


Lab4 作業(下午班)

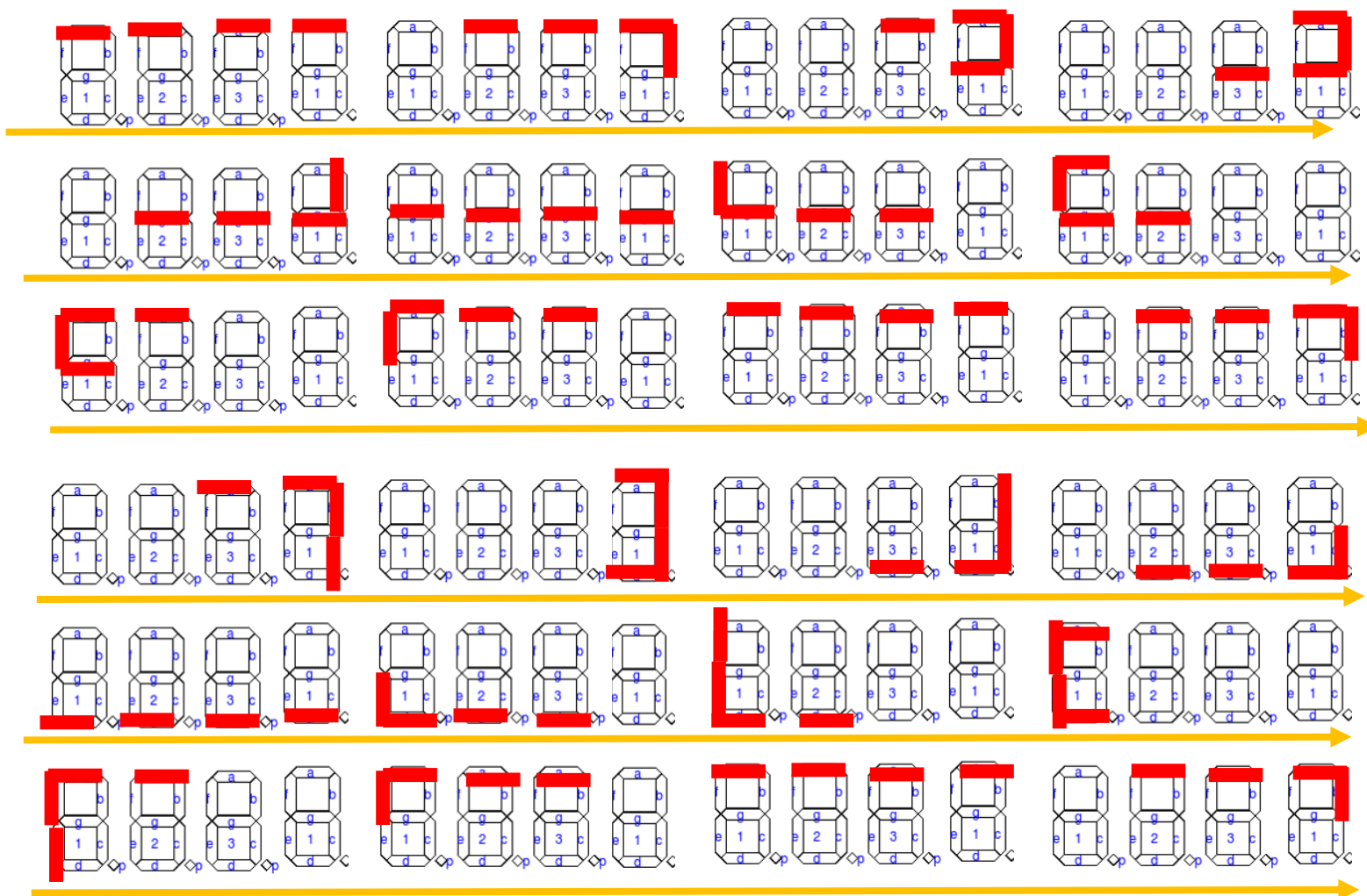


Lab4 作業(晚上班)

- ❖ 合併Lab1.2.3所學到的東西，設計功能如下：
 - ◆ 以4個七段顯示器作為輸出，顯示出貪吃蛇跑八字形
 - ◆ 以SW0(R1)作為輸入，未上撥為向右，上撥為向左
 - ◆ 以中間按鍵S2(R15)作為輸入，按下時，降低貪吃蛇的速度



Lab4 作業(晚上班)



作業提示

- ❖ 將每顆七段顯示器所對應的狀態建成表格
- ❖ 透過switch改變計數器順序，依序將表格輸出至顯示暫存器，達成左右移動的狀態
- ❖ 加速（減速）功能可使用之前 Lab 所學的除頻概念

Lab4 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

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

Button

Button	S2	R15
--------	----	-----

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

SW0	SW_0	R1
-----	------	----



**Thank you for your
attention!**

