



# 邏輯系統實習

Testbench

testbench  
國立成功大學 電機系  
**2022**

# 大綱

- 數字規格
- 延長與註解
- 系統任務
- 測試平台
- 時間經度與定義
- Verilog編碼風格

# 數字規格

- 規定長度之數字 `<size>'<base format><number>`
  - `<size>`：以十進位表示數字的bit數。
  - `<base format>`：定義此數為二進位('b或 'B)、八進位('o或 'O)、十進位('d或 'D)、十六進位('h或 'H)。
  - ex : `4'b0111`; //4bit的二進位數
  - ex : `12'habc`; //12bit的十六進位數
- 不定長度之數字
  - 若省略`<size>`等效於模擬器內定規格(ex : 32bit)。
  - 若省略`<base format>`等效於使用十進制。
  - ex : `23456`; //為32bit的十進位數
  - ex : `'hc3`; //為32bit的十六進位數
- 底線
  - 用於增加數值的可讀性。
  - ex : `12'1111_0000_1010`;

# 延遲與註解

## ■ 延遲

- ex :    a = 1'b0;
- #5 b = 1'b1; //在a被指定為0的5個單位時間之後，b才被指定為1。

## ■ 註解

- 可以為單行註解(//...)。
- 或多行註解(/\*...\*/)。

# 系統任務

## ■ \$display \$write

- 用於顯示變數值或字串內容(只執行一次)，類似C語言中的printf。
- Ex : \$display("Hello World");
- \$write 輸出的結果顯示後會自動換行
- \$display則不會

## ■ \$monitor

- 用於監控訊號變化，當被監控的訊號有變化時，便會輸出新的值(觸發就執行)。
- ex : \$monitor("address = %b", addr);
- 注意:一個程式執行中只能有一個monitor
- 可以使用\$monitoron \$monitoroff操作

格式定義表

格式定義表	
%b 或 %B	二進制變數
%o 或 %O	八進制變數
%d 或 %D	十進制變數
%h 或 %H	十六進制變數
%f 或 %F	十進制實數變數
%c 或 %C	ASCII字元
%t 或 %T	Time 時間

# 系統任務

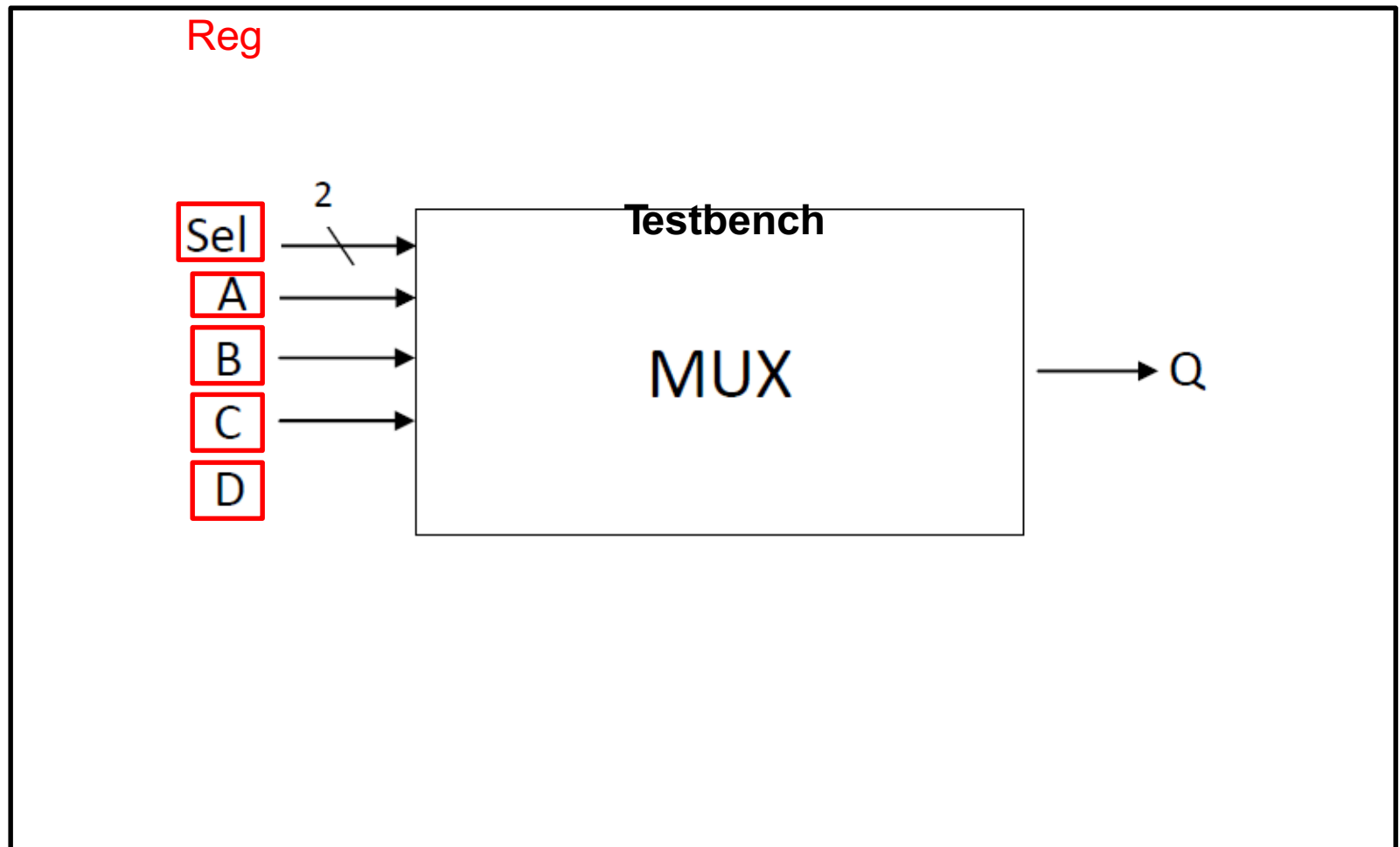
- f代表file 表示寫入你想要寫入的檔案中
  - \$fopen:將檔案打開才可以做寫入
  - \$fclose:將寫完的檔案關掉，在程式結尾務必寫入此任務
  - \$fdisplay:用於顯示變數值或字串內容(只執行一次)，類似C語言中的printf。  
顯示完自動換行。
  - \$fwrite:與\$fdisplay同但是顯示完不會自動換行
  
- \$readmemb:從檔案中讀取binary存到memory中
- \$readmemh:從檔案中讀取十六進位存到memory中



# 系統任務

- **\$stop**
  - 暫時終止模擬運算的進行。
- **\$finish**
  - 結束模擬運算。
- **\$time**
  - 回傳目前的模擬時間。

# 測試平台 (*testbench*)(1/3)





# 測試平台 (*testbench*)(2/3)

- 寫一個 `mux_4to1` 的測試平台。

```
module testbench();  
    reg in1, in2, in3, in4;  
    reg [1:0] sel;  
    wire out;  
  
    mux_4to1 x1(in1, in2, in3, in4, sel, out);
```

行為層次的語法，一個 **initial** 區塊啟動於模擬時間零，而且僅執行一次

產生輸入訊號

```
    initial begin  
        sel=2'b00; in1=1'b1; in2=1'b0; in3=1'b0; in4=1'b0;  
        #1 sel=2'b00; in1=1'b0; in2=1'b1; in3=1'b1; in4=1'b1;  
  
        #1 sel=2'b01; in1=1'b0; in2=1'b1; in3=1'b0; in4=1'b0;  
        #1 sel=2'b01; in1=1'b1; in2=1'b0; in3=1'b1; in4=1'b1;  
  
        #1 sel=2'b10; in1=1'b0; in2=1'b0; in3=1'b1; in4=1'b0;  
        #1 sel=2'b10; in1=1'b1; in2=1'b1; in3=1'b0; in4=1'b1;  
  
        #1 sel=2'b11; in1=1'b0; in2=1'b0; in3=1'b0; in4=1'b1;  
        #1 sel=2'b11; in1=1'b1; in2=1'b1; in3=1'b1; in4=1'b0;  
  
        #1 $finish;  
    end
```

印出輸出結果

```
    initial begin  
        $monitor($time, " sel=%d, in1=%b, in2=%b, in3=%b, in4=%b, out=%b",  
            sel, in1, in2, in3, in4, out);  
    end
```

```
endmodule
```

testbench

# 測試平台 (*testbench*)(3/3)

```
module testbench();
    reg in1, in2, in3, in4;
    reg [1:0] sel;

0 sel=0, in1=1, in2=0, in3=0, in4=0, out=1
1 sel=0, in1=0, in2=1, in3=1, in4=1, out=0
2 sel=1, in1=0, in2=1, in3=0, in4=0, out=1 in1, in2, in3, in4, sel, out);
3 sel=1, in1=1, in2=0, in3=1, in4=1, out=0
4 sel=2, in1=0, in2=0, in3=1, in4=0, out=1 in
5 sel=2, in1=1, in2=1, in3=0, in4=1, out=0 b00; in1=1'b1; in2=1'b0; in3=1'b0; in4=1'b0;
6 sel=3, in1=0, in2=0, in3=0, in4=1, out=1 b00; in1=1'b0; in2=1'b1; in3=1'b1; in4=1'b1;
7 sel=3, in1=1, in2=1, in3=1, in4=0, out=0
    #1 sel=2'b01; in1=1'b0; in2=1'b1; in3=1'b0; in4=1'b0;
    #1 sel=2'b01; in1=1'b1; in2=1'b0; in3=1'b1; in4=1'b1;

    #1 sel=2'b10; in1=1'b0; in2=1'b0; in3=1'b1; in4=1'b0;
    #1 sel=2'b10; in1=1'b1; in2=1'b1; in3=1'b0; in4=1'b1;

    #1 sel=2'b11; in1=1'b0; in2=1'b0; in3=1'b0; in4=1'b1;
    #1 sel=2'b11; in1=1'b1; in2=1'b1; in3=1'b1; in4=1'b0;

    #1 $finish;
end

    { initial begin
        $monitor($time, " sel=%d, in1=%b, in2=%b, in3=%b, in4=%b, out=%b",
            sel, in1, in2, in3, in4, out);
    }
end

endmodule
```

模擬後，主控台的螢幕顯示結果

testbench

# 時間經度與定義

- ``timescale` 仿真時間單位/時間精度:時間經度，用來定義模塊的仿真時間單位和時間精度
  - 時間精度不能比時間單位還要大
  - 時間單位和時間經度只能是1,10,100這三種整數，單位有s,ms,us,ns,ps,fs;
- ``define` 大寫名字 代表的東西:定義一些在此程式常用到的東西，可以減去手誤的問題，程式中使用時(```大寫名字)即可使用
  - ex : ``define WIDTH 32`
  - ``WIDTH`

# Verilog 編碼風格

## ■ 命名風格

- 1) 使用小寫描述I/O埠、訊號、變數；使用大寫描述常數。
  - ex : input data\_in;
  - ex : `define WIDTH 32
- 2) 使用有意義的命名。
  - ex : data\_in代表資料輸入；a、b代表...意義不明？
- 3) 使用clk命名clock；使用rst命名reset。

## ■ 撰寫風格

- 1) 適當的使用縮排與空白，以增進原始碼的可讀性。



# 實驗題

- 選擇一堂之前實驗課中所有實驗的testbench來做報告
- 分析testbench的程式碼邏輯、意義(越詳細越高分)