

# Verilog

## 硬體描述語言

# VerilogHDL

A Guide  
to Digital  
Design  
and  
Synthesis



IEEE  
1364-2001  
Compliant

## 第11章 交換層次的模型

### 11.1 交換層次的元件

### 11.2 範 例

### 11.3 總 結

### 11.4 習 題

# 前言

11.1

11.2

11.3

11.4

- 前面的章節都是針對高層次的硬體模型進行描述，並未描述硬體細節模型(如LSI所使用的電晶體)，本章將介紹金氧半導體層次的Verilog描述方法。
- Verilog雖然支持這一層級的硬體模型，但只是把電晶體單純當作開關來看待，只能處理0、1、x、z這四種訊號位準與相關的推力關係。
- 本層級的描述方法缺點是:未能處理類比線路之設計。

## 11.1 交換層次的元件

11.1

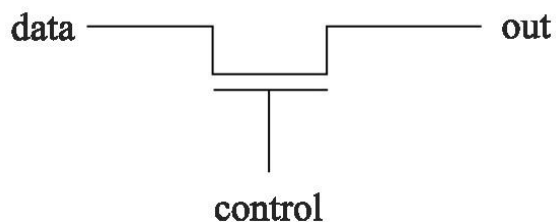
11.2

11.3

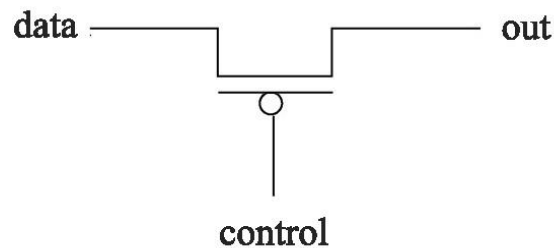
11.4

### • 11.1.1 金氧半導體式的開關

- 金氧半導體有兩種形式，分別為n型通道與p型通道金氧半導體，分別以關鍵字nmos 與 pmos 來做為代表。
- 我們常以底下的兩個符號來分別表示



N-通道金氧半導體



P-通道金氧半導體

圖11-1 N-通道金氧半導體與P-通道金氧半導體開關

## 11.1 交換層次的元件

11.1

11.2

11.3

11.4

- 11.1.1 金氧半導體式的開關

- 範例11-1 NMOS與PMOS開關的初始化

```
pmos n1(out, data, control ); // nmos 開關的範例  
nmos p1(out, data, control ); // pmos 開關的範例
```

- 開關屬於Verilog的原始關鍵字，所以可以直接使用，不一定需要給予別名，上例可修改如下。

```
nmos(out, data, control ); // nmos 開關的範例，無別名指定  
pmos(out, data, control ); // pmos 開關的範例，無別名指定
```

## 11.1 交換層次的元件

11.1

11.2

11.3

11.4

### • 11.1.1 金氧半導體式的開關

- 若以真值表的方式來表示，可以看到control 與 data 輸入分別是 1、0、z與 x 的值，其中 out 的 L 表示這一個值可能是 0 或者是未知 z，H 表示這一個值可能是 1 或者是未知 z。

表11-1 NMOS和PMOS的真值表

	nmos	control			
		0	1	x	z
data	0	z	0	L	L
	1	z	1	H	H
	x	z	x	x	x
	z	z	z	z	z

	pmos	control			
		0	1	x	z
data	0	0	z	L	L
	1	1	z	H	H
	x	x	z	x	x
	z	z	z	z	z

## 11.1 交換層次的元件

11.1

11.2

11.3

11.4

### • 11.1.2 互補式金氧半導體開關(CMOS)

- 一般來說，我們以關鍵字cmos來代表互補式金氧半導體開關，一個cmos應該由nmos與pmos組合而成。

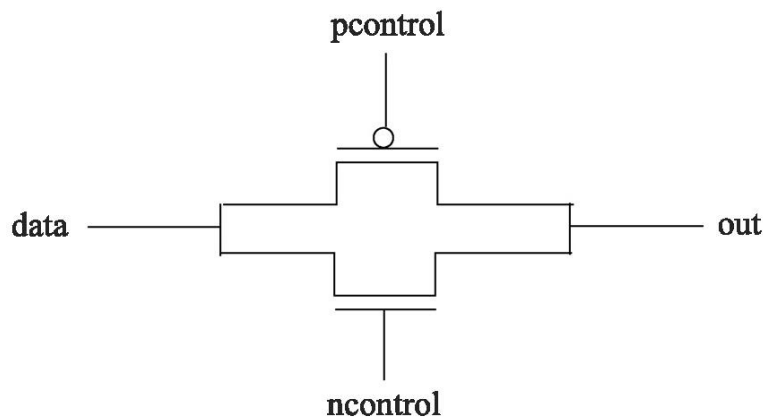


圖 11-2 CMOS開關

## 11.1 交換層次的元件

11.1

11.2

11.3

11.4

### • 11.1.2 互補式金氧半導體開關(CMOS)

#### • 範例11-2 使用CMOS開關

```
cmos cl(out, data, ncontrol, pcontrol ); // 使用別名的方式  
cmos(out, data, ncontrol, pcontrol );    // 不使用別名
```

- ncontrol 與 pcontrol 這兩個訊號應該彼此為互補值，一個是 0 則另一個就是 1，或是一個是 1 則另一個就是 0。若套上 pmos 與 nmos的關鍵字，上例可修改如下。

```
nmos(out, data, ncontrol ); // nmos 開關的範例  
pmos(out, data, pcontrol ); // pmos 開關的範例
```

- 同理，因為cmos是來自pmos與nmos的組合，會有一致的真值表。

## 11.1 交換層次的元件

### • 11.1.3 雙向開關

- N型、P型半導體與互補式型半導體，都是將源端(source)導通至目的端。所以應該可以設計一個開關，使得元件的兩端都有可能是源端。在Verilog中，我們可以使用三個關鍵字來處理，分別是 tran 、tranif0 與 tranif1 。

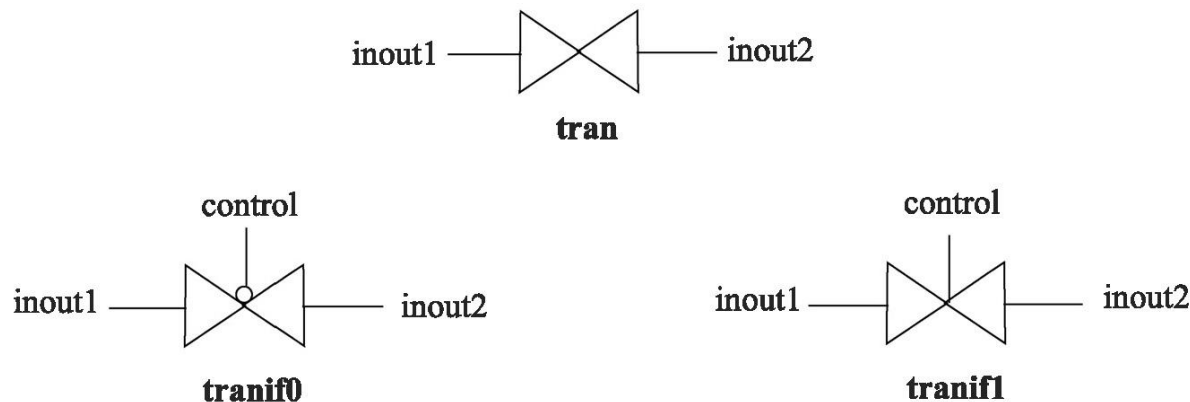


圖11-3 雙向開關

11.1

11.2

11.3

11.4

## 11.1 交換層次的元件

11.1

11.2

11.3

11.4

### • 11.1.3 雙向開關

- tran型態的雙向開關只是單純地當成一個雙向緩衝器，兩端都可以當作源端。而tranif0型態的雙向開關只有在control訊號為0時，才會當成一個雙向緩衝器。如果是1時，非源端側的值將為高阻抗狀態，而tranif1型態則剛好是相反。
- 範例11-3 使用雙向開關

```
tran t1(inout1, inout2 );           // 使用別名的方式
tranif0(inout1, inout2 control ); // 不使用別名
tranif1(inout1, inout2, control);  // 不使用別名
```

## 11.1 交換層次的元件

11.1

11.2

11.3

11.4

### • 11.1.4 電源與接地

- 在電晶體層級的設計中，一定會需要電源端(Vdd，邏輯值1)，或是接地端(GND，邏輯值0)。在Verilog中，我們以supply1來代表電源端，以supply0來代表接地端。
- 使用範例

```
supply1 vdd;  
supply0 gnd;
```

```
assign a = vdd; // 將a連接到電源端  
assign b = gnd; // 將b連接到接地端
```

## 11.1 交換層次的元件

11.1

11.2

11.3

11.4

### • 11.1.5 電阻式開關

- 前述的三種開關元件都可以再加上電阻值的設定，一個有電阻的開關就不再是理想開關。首先，它的阻抗值會增加，而且通過的訊號強度會降低，實際用的元件通常都具有此特性。在Verilog中，對於有電阻性的開關，都在關鍵字前加上r這一個字首。

```
rnmos rpmos      // 電阻式 pmos 與 nmos 開關  
Rcmos            // 電阻式 互補開關  
  
rtran rtranf0 rtranf1 //電阻式 雙向開關
```

- 開關有無電阻值設定之差異
  - 源端到目的端的阻抗不同
  - 通過訊號的處理方式

## 11.1 交換層次的元件

11.1

11.2

11.3

11.4

### • 11.1.5 電阻式開關

- 電阻式開關有較大的源端到目的端的阻抗值，理想開關通常有著近乎零的阻抗值。電阻式開關會降低通過訊號的強度，理想開關則不會。

表11-2 訊號強度衰減對照表

輸入強度	輸出強度
supply	pull
strong	pull
pull	weak
weak	medium
large	medium
medium	small
small	small
high	high

# 11.1 交換層次的元件

11.1

11.2

11.3

11.4

## • 11.1.6 在開關中的延遲設定

### • 金氧半導體與互補式金氧半導體開關部份

- 相同於一般的線路設計，在開關中一樣可以給予延遲時間設定。延遲的時間可以是有值或是無值(延遲時間為零)，也可以分成是上升延遲、下降延遲與關閉延遲。同樣有一、二、三種值的指定方法。

表11-3 延遲時間設定與半導體與互補式半導體開關的關係

開關型態	延遲設定種類	範例
pmos, nmos, rpmos, rnmos	無值(不延遲)	pmos p1(out,data,control);
	一個值(所有的延遲型態延遲值都一樣)	pmos #(1)p1(out,data,control);
	二個值(上升延遲與下降延遲)	nmos #(1,2)p2(out,data,control);
	三個值(上升延遲、下降延遲與關閉延遲)	nmos #(1,3,2)p2(out,data,control);
cmos, rcmos	同上	cmos #(5)c2(out,data,nctrl,pctrl); cmos #(1,2)c1(out,data,nctrl,pctrl);

# 11.1 交換層次的元件

11.1

11.2

11.3

11.4

## • 11.1.6 在開關中的延遲設定

### • 雙向開關部份

- 雙向開關不同於其他開關種類，一般只有開啟延遲與關閉延遲這兩種。所以描述方式只有三種:無延遲值、一個延遲值(開啟與關閉共用)及兩個延遲值。

### • 指定區塊

- 接腳對接腳的延遲時間設定及相關的時間檢查，都可以用在開關上。

表11-4 延遲時間設定與雙向開關的關係

開關種類	延遲設定	範例
tran,rtran	無延遲設定	
tranif1, rtranif1, tranif0, rtranif0	無延遲值 一個延遲值(開啟與關閉共用) 兩個延遲值(分別針對開啟與關閉)	rtranif0 rt1(inout1,inout2,control); tranif0 #(3)T(inout1,inout2,control); tranif1 #(1,2)t1(inout1,inout2,control);

## 11.2 範 例(Examples)

### • 11.2.1 互補式金氧半導體非或(NOR)邏輯閘

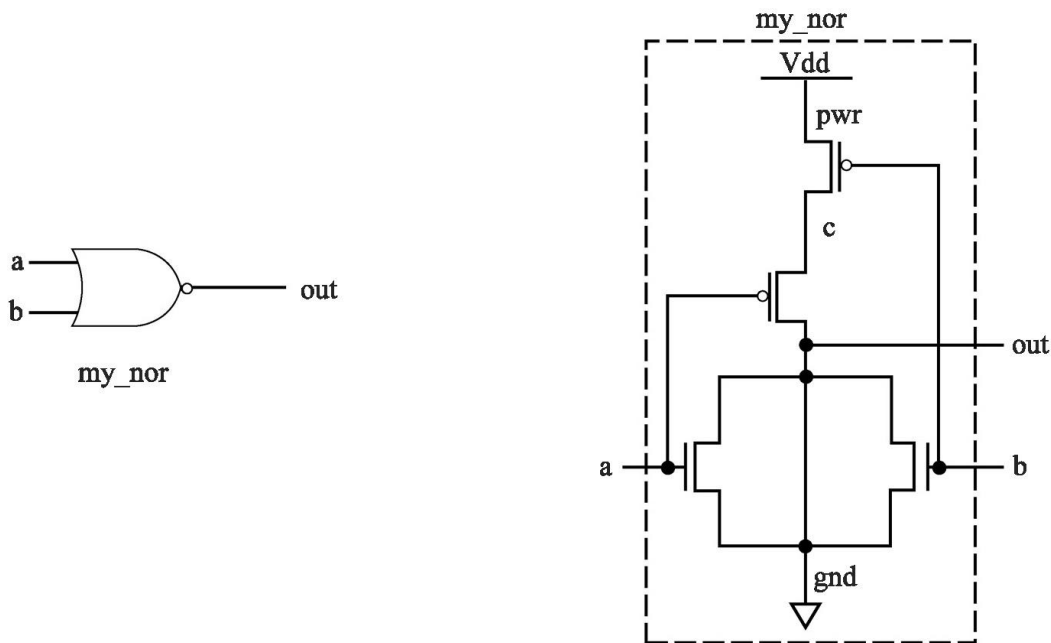


圖11-4 Nor Gate與開關示意圖

11.1

11.2

11.3

11.4

## 11.2 範例(Examples)

[11.1](#)[11.2](#)[11.3](#)[11.4](#)

- 11.2.1 互補式金氧半導體非或(NOR)邏輯閘

- 範例11-4 非或邏輯閘的Verilog描述

```
// 定義這一個模組為 my_nor  
module my_nor(out, a, b);
```

```
output out;  
input a, b;
```

```
// 內部連線  
wire c;
```

```
// 設定電源端與接地端  
supply1 pwr;  
supply0 gnd;
```

```
// 初始化開關  
pmos(c, pwr, b);  
pmos(out, c, a);  
nmos(out, gnd, a);  
nmos(out, gnd, b);
```

```
endmodule
```

## 11.2 範 例(Examples)

11.1

11.2

11.3

11.4

### • 11.2.1 互補式金氧半導體非或(NOR)邏輯閘

#### • 範例11-4之模擬測試程式

```
// 邏輯閘的測試輸入
module stimulus;
reg A, B;
wire OUT;

// 初始化非或邏輯
my_nor n1(OUT, A, B);
```

結果輸出如下：

```
0 OUT = 1, A= 0, B=0
5 OUT = 0, A= 0, B=1
10 OUT = 0, A= 1, B=0
15 OUT = 0, A= 1, B=1
```

```
// 灌入測試訊號組合，兩組輸入，四種輸入訊號組合。
```

```
initial
begin
    A=1'b0; B=1'b0;
    #5 A=1'b0; B=1'b1;
    #5 A=1'b1; B=1'b0;
    #5 A=1'b1; B=1'b1;
end
```

```
// 檢查輸出
```

```
initial
    $monitor($time, " OUT = %b , A= %b, B=%b", OUT,A,B);

endmodule
```

## 11.2 範 例(Examples)

### • 11.2.2 2 對 1 多工器

- 當 S 值為零時，OUT 的值等於 I0，S 值為零時，OUT 的值等於 I1。

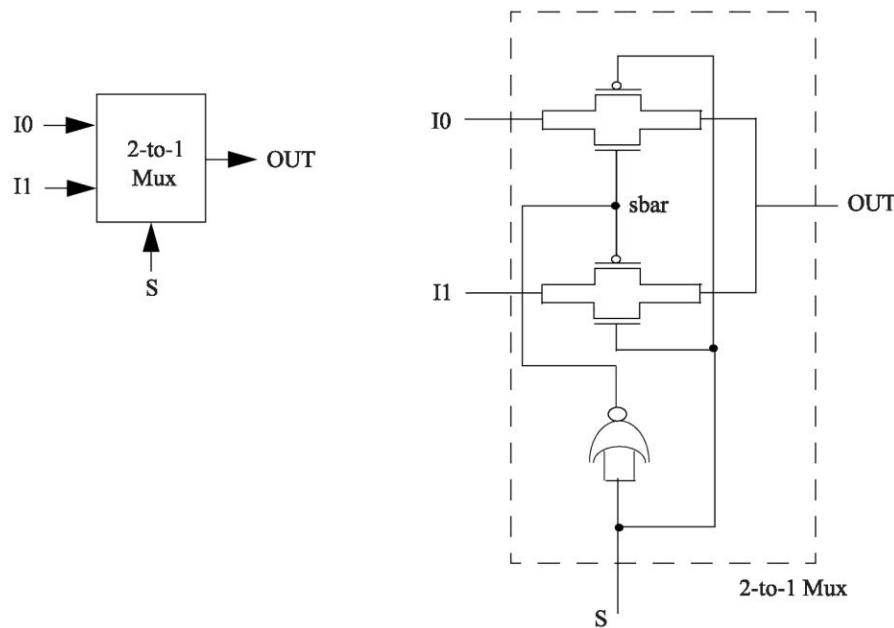


圖11-5 二對一多工器

11.1

11.2

11.3

11.4

## 11.2 範 例(Examples)

11.1

11.2

11.3

11.4

### • 11.2.2 2 對 1 多工器

- 範例11-5 以交換層次的寫法來描述一個多工器

```
// 定義一個 2 對 1 的多工器
module my_mux(out, s, i0, i1);

output out;
input s, i0,i1;

// 內部接線
```

```
wire sbar; // s 的互補值

// 利用前面的 my_nor 模組
my_nor nt(sbar, s, s); // 與一個反閘功能相同

cmos(out, i0, sbar, s);
cmos(out, i1, s, sbar);

endmodule
```

## 11.2 範 例(Examples)

### • 11.2.3 簡單的互補式電晶體正反器

- 這個正反器是用來當反相器使用。
- 範例11-6 反相器

```
// 定義一個反相器
module my_not(out, in);

output out;
input in;

// 宣告電源端與接地端
supply1 pwr;
supply0 gnd;

// 初始化開關
pmos(out,pwr,in);
nmos(out,gnd,in);

endmodule
```

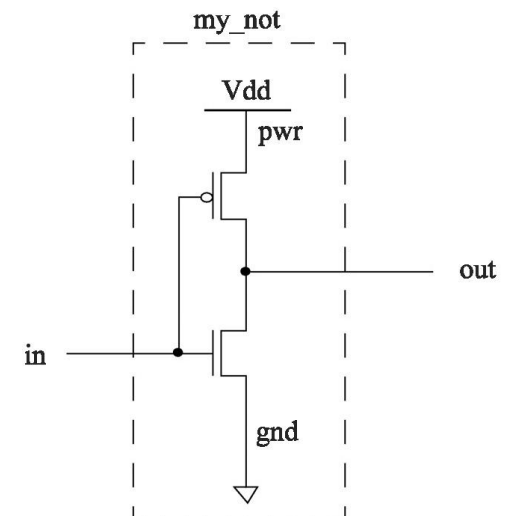
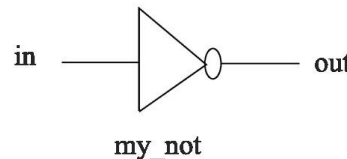


圖11-7 CMOS正反器 (反相器)

11.1

11.2

11.3

11.4

## 11.2 範 例(Examples)

11.1

11.2

11.3

11.4

### • 11.2.3 簡單的互補式電晶體正反器

- 這個正反器是用來儲存資料，是一個與訊號位準有關的循序電路，而非只是組合電路。
- C1與C2兩個模組使用前述之互補式電晶體開關，當clk為1時，開關C1打開。反之，當clk為0時，開關C2打開，C1與C2的clk輸入互為互補值。

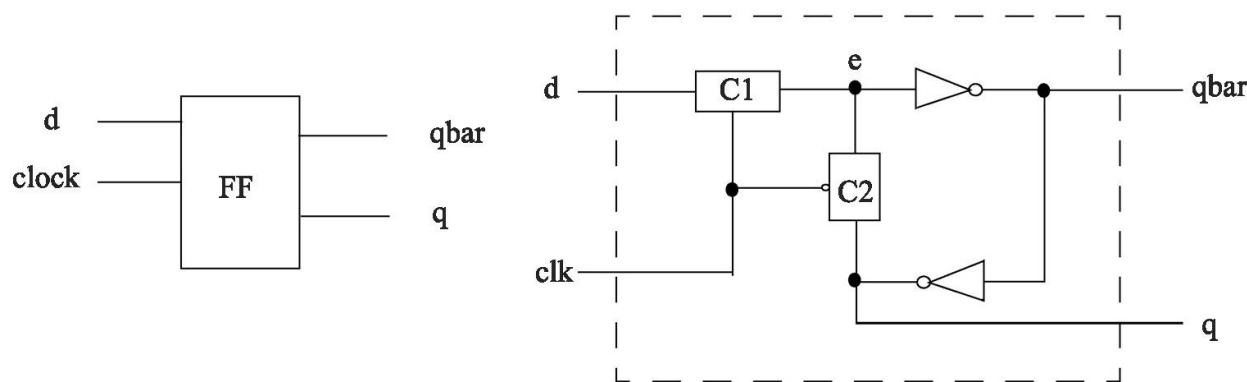


圖11-6 CMOS正反器

## 11.2 範 例(Examples)

11.1

11.2

### • 11.2.3 簡單的互補式電晶體正反器

11.3

#### • 範例11-7 互補式電晶體正反器

11.4

```
// 定義正反器
module cff(q, qbar, d, clk);

output q, qbar;
input d, clk;

// 內部連線
wire e;
wire nclk; //clk 的互補訊號

// 使用反相器
my_not nt(nclk, clk);
```

```
// 使用互補式電晶體開關
cmos(e, d, clk, nclk); // switch c1 closed i.e. e = d,
                        // when clk = 1
cmos(e, q, nclk, clk); // switch c2 closed i.e. e = q,
                        // when clk = 0

// 使用反相器
my_not nt1(qbar, e);
my_not nt2(q, qbar);

endmodule
```