



## 2.3 SystemVerilog運算子

Mao-Hsu Yen  
ymh@mail.ntou.edu.tw



# 目錄

1. 運算符號種類
2. 優現順序
3. 基本的算術運算子
4. 關係運算子
5. 等式運算子
6. 移位運算子
7. 位拚接運算子
8. 縮減運算子

# ● 運算子

種類	符號	運算功能	需要運算元數目	種類	符號	運算功能	需要運算元數目
算數 運算	*	乘法	2	位元 運算	~	位元的NOT	1
	/	除法	2		&	位元的AND	2
	+	加法	2			位元的OR	2
	-	減法	2		^	位元的XOR	2
	%	取餘數	2		~^	位元的XNOR	2
邏輯 運算	!	邏輯的NOT	1	化簡 運算	&	化簡的位元AND	1
	&&	邏輯的AND	2		~&	化簡的位元NAND	1
		邏輯的OR	2			化簡的位元OR	1
比較 運算	>	大於	2		~	化簡的位元NOR	1
	<	小於	2		^	化簡的位元XOR	1
	>=	大於或等於	2		~^	化簡的位元XNOR	1
	<=	小於或等於	2	移位 運算	>>	向右移位	2
相等 運算	==	等於	2		<<	向左移位	2
	!=	不等於	2	連結 運算	{ }	連結	任意數目
條件 運算	?:	做條件運算	3		{ { }	複製	任意數目

## ● 運算符號種類

---

- SystemVerilog HDL的語言的運算符的範圍很廣，

按照其功能大概可以分為以下幾類：

- (1)算術運算子 +、-、\*、/、%
- (2)賦值運算子 =、<=
- (3)關係運算子 >、<、>=、<=
- (4)邏輯運算子 &&、||、!
- (5)條件運算子 ? :
- (6)位運算子 ~、|、^、&、^~
- (7)移位運算子 <<、>>
- (8)位拼接運算子 {}
- (9)其他。



## ● 優先順序

---

! ~  
\* / %  
+ -  
<< >>  
< <= > >=  
== != === !==  
&  
& ^ ^~  
|  
&&  
||  
?:

最高優先順序別



最低優先順序別

## ● 關係運算符

---

- $<$ 、 $>$ 、 $\leq$ 、 $\geq$  如果關係運算不為真，則返回值是0，如果關係是真的，則返回值是1。
- 關係運算子的優先順序別低於算數運算子。
- 如：

$a < \text{size}-1$                     //這種表達方式與下面的表達方式相同

$a < (\text{size}-1)$                 //

$\text{size}-1 < a$                     //這種表達方式與下面的表達方式不同

$\text{size} - (1 < a)$                 //

## ● 等式運算符

---

- `==`、`!=`、`===`、`!==` 符號之間不能有空格。
- “`==`”和“`!=`”稱作邏輯等式運算子，其結果由兩個操作數的值決定。由於操作數可能是x或z，其結果可能為x；
- “`===`”和“`!==`”常用於case表達式的判別，又稱作case等式運算子。其結果只為0和1。如果操作數中存在x和z，那麼操作數必須完全相同結果才為1，否則為0。
- 邏輯等式運算子和case等式運算子的區別：



## ● 移位運算符

---

- << 、 >>

a>>n其中a代表要進行移位的操作數，n代表要移幾位。這兩種移位運算都用0來填補移出的空位。

- 如果操作數已經定義了位寬，則進行移位後操作數改變，但是其位寬不變。

- << 左移 //=>乘二

- >> 右移 //=>除二

b=10; //(1010)a=b<<1; //b左移一位元=(10100)=(14)h=20

b=10; //(1010)a=b>>1; // b右移一位元=(0101)=5



## ● 位拼接運算符

---

- {信號1的某幾位，信號2的某幾位，.....信號n的某幾位} 將某些信號的某些為列出來中間用逗號分開，最後用大括弧括起來表示一個整體的信號。
- 在位拼接的表達式中不允許存在沒有指明位數的信號。

{a, b[3:0], w, 3'b101}                      //等同於{a, b[3], b[2], b[1], b[0], w, 1'b1, 1'b0, 1'b1}

{4{w}}    //等同於{w, w, w, w}

{b, {3{a, b}}}

//等同於{b, a, b, a, b, a, b} 這裡面的3、4必須是常量表達式

## ● 縮減運算符

---

- 這是單目運算子，也包括~, |, ^, &, ^~。運算規則與位運算相似，不過是對單個運算子的每一位逐步運算，最後的運算結果是一位的二進位數。

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

```
reg      C;
```

```
assign C = &B;
```

```
C=&B;      //意思同c=((B[0]&B[1]) &B[2]) & B[3];
```



THANKS

END