

# 第十二章

# 位运算

- 主要内容

**12.1 位运算符和位运算**

**12.2 位运算举例（略）**

**12.3 位段(略)**

## 概念

- 位运算是指按二进制位进行的运算。因为在系统软件中，常要处理二进制位的问题。
- 例如：将一个存储单元中的各二进制位左移或右移一位，两个数按位相加等。
- C语言提供位运算的功能，与其他高级语言（如PASCAL）相比，具有很大的优越性。

## 12.1 位运算符和位运算

C 语言提供的位运算符有：

运算符	含义	运算符	含义
&	按位与	~	取反
	按位或	<<	左移
^	按位异或	>>	右移

### 说明：

- (1) 位运算符中除~以外，均为二目（元）运算符，即要求两侧各有一个运算量。
- (2) 运算量只能是整型或字符型的数据，不能为实型数据。



## 12.1.1 “按位与”运算符 (&)

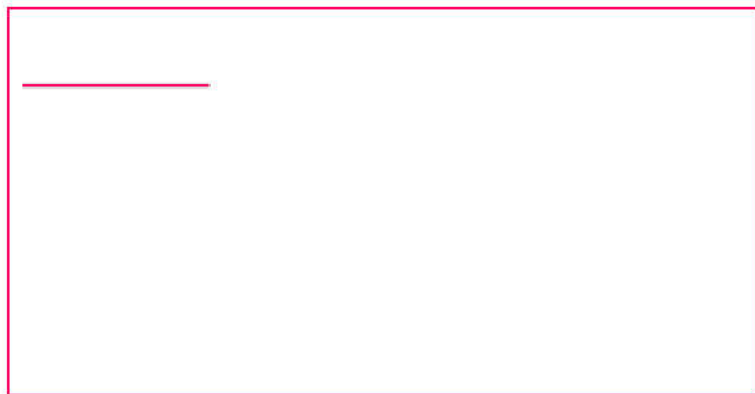
按位与是指：参加运算的两个数据，按二进制位进行“与”运算。如果两个相应的二进制位都为1，则该位的结果值为1；否则为0。即：

$$0 \& 0 = 0, \quad 0 \& 1 = 0, \quad 1 \& 0 = 0, \quad 1 \& 1 = 1$$

例：3 & 5 并不等于 8，应该是按位与运算：

```
00000011 (3)
& 00000101 (5)
-----
00000001 (1)
```

3&5的值得 1



## 12.1.2 “按位或”运算符（|）

两个相应的二进制位中只要有一个为1，该位的结果值为1。

即  $0 | 0 = 0$ ， $0 | 1 = 1$ ， $1 | 0 = 1$ ， $1 | 1 = 1$

**例：**  $060 | 017$ ，将八进制数60与八进制数17进行

```
00110000
| 00001111
-----
00111111
```

### 12.1.3“异或”运算符（ $\wedge$ ）

异或运算符 $\wedge$ 也称XOR运算符。它的规则是：  
若参加运算的两个二进制位同号则结果为0（假）  
异号则结果为1（真）

即： $0 \wedge 0 = 0$ ， $0 \wedge 1 = 1$ ， $1 \wedge 0 = 1$ ， $1 \wedge 1 = 0$

例：

```
00111001
^ 00101010
—————
00010011
```

即： $071 \wedge 052 = 023$ （八进制数）



## 12.1.4 “取反”运算符（~）

~是一个单目（元）运算符，用来对一个二进制数按位取反，即将0变1，将1变0。例如，~025是对八进制数25（即二进制数00010101）按位求反。

0000000000010101

(~)

---

1111111111101010（八进制数177752）



## 12.1.5 左移运算符 (<<)

左移运算符是用来将一个数的各二进制位全部左移若干位。

例如： $a \ll 2$  将  $a$  的二进制数左移 2 位，右补 0。  
若  $a = 15$ ，即二进制数  $00001111$ ，  
左移 2 位得  $00111100$ ，(十进制数  $60$ )

高位左移后溢出，舍弃。

## 12.1.5 左移运算符 (<<)

左移 1 位相当于该数乘以 2，左移 2 位相当于该数乘以  $2^2=4$ ， $15<<2=60$ ，即乘了 4。但此结论只适用于该数左移时被溢出舍弃的高位中不包含 1 的情况。

假设以一个字节（8 位）存一个整数，若 a 为无符号整型变量，则 a = 64 时，左移一位时溢出的是 0，而左移 2 位时，溢出的高位中包含 1。

## 12.1.6 右移运算符(>>)

右移运算符是 $a \gg 2$ 表示将 $a$ 的各二进制位右移2位，移到右端的低位被舍弃，对无符号数，高位补0。

例如： $a=017$ 时：

$a$ 的值用二进制形式表示为00001111，  
舍弃低2位11： $a \gg 2 = 00000011$

右移一位相当于除以2  
右移 $n$ 位相当于除以 $2^n$ 。



## 12.1.7 位运算赋值运算符

位运算符与赋值运算符可以组成复合赋值运算符。

例如：&=, |=, >>=, <<=, ^=

例：  $a \&= b$  相当于  $a = a \& b$

$a \ll = 2$  相当于  $a = a \ll 2$

