

Appendix B 位运算

——bitwise operations

B.1 位运算符

B.2 位段

B.1 位运算符 bitwise operators

➤ 整数的表示

◆ **15**， **-15** 的二进制表示

➤ 提示

◆ 在计算机中，数值一律以补码表示

◆ 绝对值相同的数，原码除符号位外都相同

◆ 正数的补码、原码、反码相同

◆ 负数的补码，是原码的反码加1

➤ 注意：最高位是符号位，1表示负数，0表示正数



B.1.1 移位运算符

——bitwise shift operators

➤ 左移位运算符 <<

◆ 向左移动指定位，空位补0

● Eg. $E1 \ll E2$ 表示 $E1$ 向左移 $E2$ 位，结果为：

$$E1 \times 2^{E2}$$

➤ 右移位运算符 >>

◆ 向右移动指定位，空位补0

● Eg. $E1 \gg E2$ 表示 $E1$ 向右移 $E2$ 位，结果为：

$$E1 / 2^{E2}$$

B.1.2 位与运算符

——bitwise AND operator

➤ 位与运算符 &

- ◆ 运算数必须为整型
- ◆ 逐位计算，对应位均为1时结果为1

➤ 主要用途

- ◆ 清零（用**0**相**&**）
- ◆ 保留指定位（用**1**相**&**）
- ◆ 检测某位的值（用**1**相**&**）

B.1.3 位或运算符

——bitwise inclusive OR operator

➤ 位或运算符 |

◆ 运算数必须为整型

◆ 逐位计算，对应位均为0时结果为0

B.1.4 异或运算符

——bitwise exclusive OR operator

➤ 异或运算符 \wedge

◆ 运算数必须为整型

◆ 逐位计算，对应位不同时结果为1

B.1.5 位反运算符

——bitwise complement operator

➤ 位反运算符 \sim

◆ 运算数必须为整型

◆ 逐位取反, $0 \rightarrow 1, 1 \rightarrow 0$



B.2 位段 ——bit field

- 位段(**bit field**),就是结构(**struct**)中的一种字段(成员),这种成员是以二进制位作为长度单位的。

```
struct p_data{  
    unsigned a:2;  
    unsigned b:3;  
    int i;  
}data;
```


- 位段成员的类型必须指定为**unsigned** 或**int**类型;
- 可定义长度为**0**的位段, 使某一位段从另一个字开始存放
- 可定义无名位段, 表明该几位空间不用
- 一个位段不能跨单元存储。如果第一个单元空间不能容纳下一个位段, 则放弃该空间, 而从下一个单元起存放该位段
- **5**、因为最小的存储单位是字节, 所以不允许取位段的地址
- **6**、位段也只能在结构中定义
- **7**、位段可以用整型格式符输出。例如:
 - ◆**printf("%d,%d,%d",data.a,data.b,data.c);**
- **8**、位段可以在数值表达式中引用, 它会被系统自动地转换成整型数。