位运算

按位运算

- C有这些按位运算的运算符：

- **&** 按位的与
- **|** 按位的或
- **~** 按位取反
- **^** 按位的异或
- **<<** 左移
- **>>** 右移

按位与 &

- 如果 $(x)_i == 1$ 并且 $(y)_i == 1$，那么 $(x \ \& \ y)_i = 1$
- 否则的话 $(x \ \& \ y)_i = 0$
- 按位与常用于两种应用：
  - 让某一位或某些位为0：`x & 0xFE`
  - 取一个数中的一段：`x & 0xFF`

按位或 |

- 如果 $(x)_i == 1$ 或 $(y)_i == 1$，那么 $(x \mid y)_i = 1$
- 否则的话，$(x \mid y)_i == 0$
- 按位或常用于两种应用：
  - 使得一位或几个位为1：x | 0x01
  - 把两个数拼起来：0x00FF | 0xFF00

按位取反 ~

- $(\sim x)_i = 1 - (x)_i$

- **把1位变0，0位变1**

- 想得到全部位为1的数：~0

- 7的二进制是0111，x | 7使得低3位为1，而

- x & ~7，就使得低3位为0

逻辑运算vs按位运算

- 对于逻辑运算，它只看到两个值：0和1
- 可以认为逻辑运算相当于把所有非0值都变成1，然后做按位运算
  - 5 & 4 —>4而 5 && 4 —> 1 & 1 —> 1
  - 5 | 4 —> 5而 5 || 4 —> 1 | 1 —> 1
  - ~4 —> 3而 !4 —> !1 —> 0

按位异或^

- 如果 $(x)_i == (y)_i$，那么 $(x \hat{} y)_i = 0$
- 否则的话，$(x \hat{} y)_i == 1$
- 如果两个位相等，那么结果为0；不相等，结果为1
- 如果x和y相等，那么x ^ y的结果为0
- 对一个变量用同一个值异或两次，等于什么也没做
  - x ^ y ^ y —> x

左移 <<

- **`i << j`**
- i中所有的位向左移动j个位置，而右边填入0
- 所有小于int的类型，移位以int的方式来做，结果是int
- **`x <<= 1`** 等价于 **`x *= 2`**
- **`x <<= n`** 等价于 **`x *= 2`**^{n}.

右移 >>

- **`i >> j`**
- i中所有的位向右移j位
- 所有小于int的类型，移位以int的方式来做，结果是int
- 对于unsigned的类型，左边填入0
- 对于signed的类型，左边填入原来的最高位（保持符号不变）
- **`x >>= 1`** 等价于 **`x /= 2`**
- **`x >>= n`** 等价于 **`x /= 2`**^{n}.

no zuo no die

- 移位的位数不要用负数，这是没有定义的行为
  - `x << -2  //!!NO!!`

输出一个数的二进制

```c
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int number;
    scanf("%d", &number);
    unsigned mask = 1u<<31;
    for ( ; mask ; mask >>=1 ) {
        printf("%d", number & mask?1:0);
    }
    printf("\n");

    return 0;
}
```

MCU的SFR

Table 70: UART0 Line Control Register (U0LCR - 0xE000C00C)

U0LCR	Function	Description	Reset Value
1:0	Word Length Select	00: 5 bit character length 01: 6 bit character length 10: 7 bit character length 11: 8 bit character length	0
2	Stop Bit Select	0: 1 stop bit 1: 2 stop bits (1.5 if U0LCR[1:0]=00)	0
3	Parity Enable	0: Disable parity generation and checking 1: Enable parity generation and checking	0
5:4	Parity Select	00: Odd parity 01: Even parity 10: Forced “1” stick parity 11: Forced “0” stick parity	0
6	Break Control	0: Disable break transmission 1: Enable break transmission. Output pin UART0 TxD is forced to logic 0 when U0LCR6 is active high.	0
7	Divisor Latch Access Bit	0: Disable access to Divisor Latches 1: Enable access to Divisor Latches	0

MCU的SFR

Table 70: UART0 Line Control Register (U0LCR - 0xE000C00C)

U0LCR	Function	Description	Reset Value
1:0	Word Length Select	00: 5 bit character length 01: 6 bit character length 10: 7 bit character length 11: 8 bit character length	0
2	Stop Bit Select	0: 1 stop bit 1: 2 stop bits (1.5 if U0LCR[1:0]=00)	0
3	Parity Enable	0: Disable parity generation and checking 1: Enable parity generation and checking	0
5:4	Parity Select	00: Odd parity 01: Even parity 10: Forced “1” stick parity 11: Forced “0” stick parity	0
6	Break Control	0: Disable break transmission 1: Enable break transmission. Output pin UART0 TxD is forced to logic 0 when U0LCR6 is active high.	0
7	Divisor Latch Access Bit	0: Disable access to Divisor Latches 1: Enable access to Divisor Latches	0

- const unsigned int SBS = 1u << 2;
- const unsigned int PE = 1u << 3;
- U0LCR |= SBS | PE;
- U0LCR &= ~SBS;
- U0LCR &= ~(SBS | PE);

位段

- 把一个int的若干位组合成一个结构

```
struct {
    unsigned int leading : 3;
    unsigned int FLAG1: 1;
    unsigned int FLAG2: 1;
    int trailing: 11;
};
```

位段

- 可以直接用位段的成员名称来访问
  - 比移位、与、或还方便
- 编译器会安排其中的位的排列，不具有可移植性
- 当所需的位超过一个int时会采用多个int