位掩码就是方便对寄存器进行置位或者清零而设立的

原理就是:

对某八位寄存器的第三位置位则可以使用逻辑运算来操作

1 /\* 将REG第5位置位 \*/

2 REG |= (1<<5);

3

4 /\* 将REG第2和6位置位 \*/

5 REG |= (1<<2) |(1<<6);

/\* 将REG第5位清零 \*/

REG &= ~(1<<5);

/\* 将REG第2和6位清零 \*/

REG &= ~( (1<<2) | (1<<6) ) ;

然后就可以使用宏定义进行快捷操作了

#define BIT2\_MASK 0b00000100

#define BIT5\_MASK 0b00100000

#define BIT6\_MASK 0b01000000

程序就变为

/\* 将REG第5位置位 \*/

REG |= BIT5\_MASK;

/\* 将REG第2和6位置位，两段代码效果一样 \*/

REG |= BIT2\_MASK | BIT6\_MASK;

REG |= BIT2\_MASK + BIT6\_MASK;

/\* 将REG第5位清零 \*/

REG &= ~BIT5\_MASK;

/\* 将REG第2和6位清零，两段代码效果一样\*/

REG &= ~(BIT2\_MASK | BIT6\_MASK) ;

REG &= ~(BIT2\_MASK + BIT6\_MASK) ;

掩码的两个应用：**取出一个变量或者寄存器中某一位的值和作为判断条件**

比较十位千位百位是多大怎么操作