

MSP430位操作寄存器配置宏定义的方法

2021年7月7日

MSP430单片机的寄存器赋值都是直接给整个寄存器赋值，如

```
P1OUT = 0x01;
```

那如果只想给寄存器其中的一位赋值，能否直接对某位进行操作呢？答案是不可以。因为MSP430属于RISC精简指令集单片机，它有一个特点就是不能进行位操作。那么如果只想操作某一位，而不关心其它位，例如我只想将P1.0赋值为1，但我不知道也不想改变P1.1-P1.7的状态，该怎么办呢？

下面就来介绍MSP430中进行位操作的办法。

1) 写位操作

在对某字节使用“=”进行写操作时，所有位的值都将被改变。如果先将原字节读出来，再使用按位操作符对原字节进行赋值，则可“等效”实现对单个位的写操作。下面举例来说明MSP430中最重要的3个写位操作语句。

例：将P1.0置1，P1.1置0，P1.2取反，不影响其他位。

```
P1OUT |= 0x01;    //按位“或”，相当于置1
```

```
P1OUT &= ~0x02;    //取反后再按位“与”，相当于置0
```

```
P1OUT ^= 0x04;    //按位“异或”，相当于取反
```

在CCS新建工程时，都会包含“MSP430.h”，这个头文件其实相当于一个目录，会根据当前工程所选的芯片型号链接到对应的头文件，例如MSP430G2553芯片就会链接到“MSP430G2553.h”。这些头文件都在“C:\ti\ccsv5\ccs_base\msp430\include”目录下，其中包含了许多宏定义，利用这个宏定义可以帮助我们更直观的方式来写寄存器，不用每次都把要写的值换算成16进制了。

例如MSP430G2553.h中有如下宏定义：

```
#define BIT0      (0x0001)
#define BIT1      (0x0002)
#define BIT2      (0x0004)
#define BIT3      (0x0008)
#define BIT4      (0x0010)
#define BIT5      (0x0020)
#define BIT6      (0x0040)
#define BIT7      (0x0080)
#define BIT8      (0x0100)
#define BIT9      (0x0200)
#define BITA      (0x0400)
#define BITB      (0x0800)
#define BITC      (0x1000)
#define BITD      (0x2000)
#define BITE      (0x4000)
#define BITF      (0x8000)
```

有了BIT0-BITF的宏定义，上面例子中的语句就可以改写为

```
P1OUT |= BIT0;    //按位“或”，相当于置1
```

```
P1OUT &= ~BIT1;    //取反后再按位“与”，相当于置0
```

```
P1OUT ^= BIT2;    //按位“异或”，相当于取反
```

另外，也可以用加号对多位同时操作。例如要将P1.0、P1.1、P1.2均置1，不影响其他位，可以写

```
P1OUT |= BIT0 + BIT1 + BIT2;
```

请注意，采用按位操作并不意味着MSP430具备了位操作能力，按位操作实际上对整个字节的8位都进行了操作，只不过对其中7位的值没影响而已。

2) 读位操作

读位操作主要是通过if语句判断的方式得到的。同样，这种变通的方法不意味着MSP430单片机可以对位进行读取，这种方法同样需要对1个字节的8位都操作。

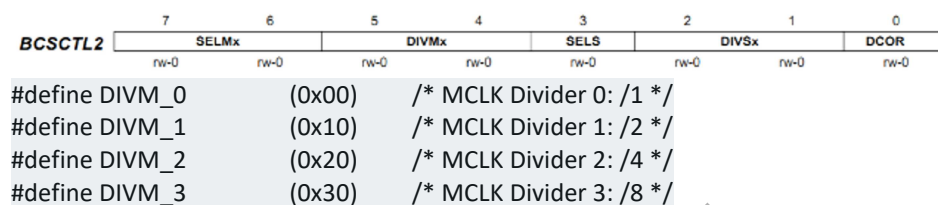
例：将P2.0的输出设置成P1.1输入相反，读取P1.0输入状态到变量temp

```
unsigned char temp = 0;
if((P1IN&BIT1) == 0) P2OUT |= BIT0;    //读P1.1 写P2.0
else                P2OUT &= ~BIT0;
if(P1IN&BIT0)      temp = 1;           //读P1.0 写temp
else                temp = 0;
```

1.5.3 MSP430寄存器的宏定义配置方法

上一节提到了MSP430G2553.h这个头文件，实际上里面除了BIT0-BIT7的宏定义之外还有很多的宏定义。这是因为MSP430的片上外设极为丰富，因此也有大量的寄存器来配置功能。MSP430上有数百个寄存器和近千个控制位，如果靠记忆这些寄存器的每一位的定义是不现实的，因此头文件的宏定义能帮助我们以更直观的方式来写程序。

例如下图中给出了BCSCTL2寄存器中DIVMx控制位的位置，下方代码中给出了MSP430头文件中两种DIVMx的宏定义，即DIVM0/1和DIVM_0/1/2/3。图中rw-0表示该位可读可写，复位后初始值为0。



这个8位寄存器中的DIVMx是用来控制MCLK的分频系数的，分频系数依次为1/2/4/8。例如我们要设置MCLK的时钟8分频，下列3条代码均可实现该功能：

```
BCSCTL2 |= 0x30;    //人脑记忆+数脚趾头，最笨的方式
BCSCTL2 |= BIT5 + BIT4; //需记忆DIVMx在寄存器中的位置，不推荐
BCSCTL2 |= DIVM_3;   //简洁明了，力荐
```

MSP430g2553教学：[\[MSP430G2 LaunchPad入门系列\] 第1篇 - MSP430简介及CCS开发环境 - MSP430 LAUNCHPAD - 全国大学生电子设计竞赛培训网 \(nuedc-training.com.cn\)](#)
[\(11条消息\) msp430内置函数总结 __bic_SR_register\(\), __delay_cycles\(x\)_u011123091的博客-CSDN博客](#)