

51单片机汇编语言教程：第15课-单片机位操作指令

(基于HJ-1G、HJ-3G实验板)

前面那些流水灯的例程，我们已经习惯了“位”一位就是一盏灯的亮和灭，而我们学的指令却全都是用“字节”来介绍的：字节的移动、加法、减法、逻辑运算、移位等等。用字节来处理一些数学问题，比如说：控制冰箱的温度、电视的音量等等很直观，能直接用数值来表在。可是如果用它来控制一些开关的打开和合上，灯的亮和灭，就有些不直接了，记得我们上次课上的流水灯的例程吗？我们知道送往 P1 口的数值后并不能马上知道哪个灯亮和来灭，而是要化成二进制才知道。工业中有很多场合需要处理这类开关输出，继电器吸合，用字节来处理就显示有些麻烦，所以在 8031 单片机中特意引入一个位处理机制。

位寻址区在 8031 中，有一部份 RAM 和一部份 SFR 是具有位寻址功能的，也就是说这些 RAM 的每一个位都有自己的地址，能直接用这个地址来对此进行操作。

内部 RAM 的 20H-2FH 这 16 个字节，就是 8031 的位寻址区。看图 1。可见这里的每一个 RAM 中的每个位我们都可能直接用位地址来找到它们，而不必用字节地址，然后再用逻辑指令的方式。

能位寻址的特殊功能寄存器 8031 中有一些 SFR 是能进行位寻址的，这些 SFR 的特点是其字节地址均可被 8 整除，如 A 累加器，B 寄存器、PSW、IP（中断优先级控制寄存器）、IE（中断允许控制寄存器）、SCON（串行口控制寄存器）、TCON（定时器/计数器控制寄存器）、P0-P3（I/O 端口锁存器）。以上的一些 SFR 我们还不熟，等我们讲解相关内容时再作详细解释。

位操作指令 MCS-51 单片机的硬件结构中，有一个位处理器（又称布尔处理器），它有一套位变量处理的指令集。在进行位处理时，CY（就是我们前面讲的进位位）称“位累加器”。有自己的位 RAM，也就是我们刚讲的内部 RAM 的 20H-2FH 这 16 个字节单元即 128 个位单元，还有自己的位 I/O 空间（即 P0.0...P0.7, P1.0...P1.7, P2.0...P2.7, P3.0...P3.7）。当然在物理实体上它们与原来的以字节寻址用的 RAM，及端口是完全相同的，或者说这些 RAM 及端口都能有两种使用办法。

位传送指令

```
MOV C, BIT
```

```
MOV BIT, C
```

这组指令的功能是实现位累加器（CY）和其它位地址之间的数据传递。

例：MOV P1.0, CY ;将 CY 中的状态送到 P1.0 管脚上去（如果是做算术运算，我们就能通过观察知道现在 CY 是多少啦）。

51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

MOV P1.0, CY ;将 P1.0 的状态送给 CY。

位修正指令

位清0指令

CLR C ;使 CY=0

CLR bit ;使指令的位地址等于0。例：CLR P1.0 ;即使 P1.0 变为0

位置1指令

SETB C ;使 CY=1

SETB bit ;使指定的位地址等于1。例：SETB P1.0 ;使 P.0 变为1

位取反指令

CPL C ;使 CY 等于原来的相反的值，由1变为0，由0变为1。

CPL bit ;使指定的位的值等于原来相反的值，由0变为1，由1变为0。

例：CPL P1.0

以我们做过的实验为例，如果原来灯是亮的，则执行本指令后灯灭，反之原来灯是灭的，执行本指令后灯亮。

位逻辑运算指令

位与指令

ANL C, bit ;CY 与指定的位地址的值相与，结果送回 CY

ANL C, /bit ;先将指定的位地址中的值取出后取反，再和 CY 相与，结果送回 CY，但注意，指定的位地址中的值本身并不发生变化。

例：ANL C, /P1.0

设执行本指令前，CY=1，P1.0 等于1（灯灭），则执行完本指令后 CY=0，而 P1.0 也是等于1。

可用下列程序验证：

```
ORG 0000H
```

```
AJMP START
```

```
ORG 30H
```

```
START: MOV SP, #5FH
```

```
MOV P1, #0FFH
```

```
SETB C
```

```
ANL C, /P1.0
```

```
MOV P1.1, C ;将做完的结果送 P1.1, 结果应当是 P1.1 上的灯亮，而 P1.0 上的灯还是不亮
```

推荐使用慧净 51 实验板。推荐 51 学习网 WWW.HLMCU.COM 淘宝网: <http://shop37031453.taobao.com/>

51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

位或指令

```
ORL C, bit
```

```
ORL C, /bit
```

这个的功能大家自行分析吧，然后对照上面的例程，编一个验证程序，看看你相得对吗？

位条件转移指令

判 CY 转移指令

```
JC rel
```

```
JNC rel
```

第一条指令的功能是如果 CY 等于1就转移，如果不等于1就次序执行。那么转移到什么地方去呢？我们能这样理解：JC 标号，如果等于1就转到标号处执行。这条指令我们在上节课中已讲到，不再重复。

第二条指令则和第一条指令相反，即如果 CY=0就转移，不等于0就次序执行，当然，我们也同样理解：JNC 标号

判位变量转移指令

```
JB bit, rel
```

```
JNB bit, rel
```

第一条指令是如果指定的 bit 位中的值是1，则转移，不然次序执行。同样，我们能这样理解这条指令：JB bit, 标号

第二条指令请大家先自行分析

下面我们举个例程说明：

```
ORG 0000H
```

```
LJMP START
```

```
ORG 30H
```

```
START: MOV SP, #5FH
```

```
MOV P1, #0FFH
```

```
MOV P3, #0FFH
```

```
L1: JNB P3. 2, L2 ;P3. 2上接有一只按钮，它按下时，P3. 2=0
```

```
JNB P3. 3, L3 ;P3. 3上接有一只按钮，它按下时，P3. 3=0
```

```
LJM P L1
```

```
L2: MOV P1, #00H
```

推荐使用慧净 51 实验板。推荐 51 学习网 WWW.HLMCU.COM 淘宝网: <http://shop37031453.taobao.com/>

[51 单片机汇编语言教程-慧净电子会员收集整理 \(全部 28 课\)](#)

```
LJMP L1
```

```
L3: MOV P1, #0FFH
```

```
LJMP L1
```

```
END
```

把上面的例程写入片子，看看有什么现象……

按下接在 P3.2 上的按钮，P1 口的灯全亮了，松开或再按，灯并不熄灭，然后按下接在 P3.3 上的按钮，灯就全灭了。这像什么？这不就是工业现场经常用到的“启动”、“停止”的功能吗？

怎么做到的呢？一开始，将 0FFH 送入 P3 口，这样，P3 的所有引线都处于高电平，然后执行 L1，如果 P3.2 是高电平（键没有按下），则次序执行 JNB P3.3, L3 语句，同样，如果 P3.3 是高电平（键没有按下），则次序执行 LJMP L1 语句。这样就不停地检测 P3.2、P3.3，如果有一次 P3.2 上的按钮按下去了，则转移到 L2，执行 MOV P1, #00H，使灯全亮，然后又转去 L1，再次循环，直到检测到 P3.3 为 0，则转 L3，执行 MOV P1, #0FFH，例灯全灭，再转去 L1，如此循环不已。大家能否稍加改动，将本程序用 JB 指令改写？

[51 实验板推荐\(点击下面的图片可以进入下载资料链接\)](#)

