

## 51 单片机汇编语言教程：第 11 课-单片机算术运算指令

(基于 HJ-1G、HJ-3G 实验板)

不带进位位的单片机加法指令

ADD A,#DATA ;例: ADD A, #10H

ADD A,direct ;例: ADD A, 10H

ADD A,Rn ;例: ADD A, R7

ADD A,@Ri ;例: ADD A, @R0

用途: 将 A 中的值与其后面的值相加, 最终结果否是回到 A 中。

例: MOV A, #30H

ADD A, #10H

则执行完本条指令后, A 中的值为 40H。

下面的题目自行练习

MOV 34H, #10H

MOV R0, #13H

MOV A, 34H

ADD A, R0

MOV R1, #34H

ADD A, @R1

带进位位的加法指令

ADDC A, Rn

ADDC A,direct

ADDC A,@Ri

ADDC A,#data

用途: 将 A 中的值和其后面的值相加, 并且加上进位位 C 中的值。

说明: 由于 51 单片机是一种 8 位机, 所以只能做 8 位的数学运算, 但 8 位运算的范围只有 0-255, 这在实际工作中是不够的, 因此就要进行扩展, 一般是将 2 个 8 位的数学运算合起来, 成为一个 16 位的运算, 这样, 能表达的数的范围就能达到 0-65535。如何合并呢? 其实很简单, 让我们看一个 10 进制数的例程:

66+78。

这两个数相加, 我们根本不在意这的过程, 但事实上我们是这样做的: 先做 6+8 (低位), 然后再做 6+7, 这是高位。做了两次加法, 只是我们做的时候并没有刻意分成两次加法来做罢了, 或者说我们并没有意识到我们做了两次加法。之所以要分成两次来做, 是因为这两个数超过了一位所能表达的范置 (0-9)。

在做低位时产生了进位, 我们做的时候是在适当的位置点一下, 然后在做高位加法是将这一点加进去。那么计算机中做 16 位加法时同样如此, 先做低 8 位的, 如果两数相加产生了进位, 也要“点一下”做个标记, 这个标记就是进位位 C, 在 PSW 中。在进行高位加法是将这个 C 加进去。例: 1067H+10A0H, 先做 67H+A0H=107H, 而 107H 显然超过了 0FFH, 因此最终保存在 A 中的是 7, 而 1 则到了 PSW 中的 CY 位了, 换言之, CY 就相当于 100H。然后再做 10H+10H+CY, 结果是 21H, 所以最终的结果是 2107H。

带借位的单片机减法指令

SUBB A, Rn

SUBB A,direct

SUBB A,@Ri

SUBB A,#data

## 51 单片机汇编语言教程-慧净电子会员收集整理 (全部 28 课)

设 (每个 H, (R2) = 55H, CY=1, 执行指令 SUBB A, R2 之后, A 中的值为 73H。

说明: 没有不带借位的单片机减法指令, 如果需要做不带位的减法指令(在做第一次相减时), 只要将 CY 清零即可。

乘法指令

MUL AB

此单片机指令的功能是将 A 和 B 中的两个 8 位无符号数相乘, 两数相乘结果一般比较大, 因此最终结果用 1 个 16 位数来表达, 其中高 8 位放在 B 中, 低 8 位放在 A 中。在乘积大于 FFFFH (65535) 时, OV 置 1 (溢出), 不然 OV 为 0, 而 CY 总是 0。

例: (A) = 4EH, (B) = 5DH, 执行指令

MUL AB 后, 乘积是 1C56H, 所以在 B 中放的是 1CH, 而 A 中放的则是 56H。

除法指令

DIV AB

此单片机指令的功能是将 A 中的 8 位无符号数除了 B 中的 8 位无符号数 (A/B)。除法一般会出现小数, 但计算机中可没法直接表达小数, 它用的是我们小学生还没接触到小数时用的商和余数的概念, 如 13/5, 其商是 2, 余数是 3。除了以后, 商放在 A 中, 余数放在 B 中。CY 和 OV 都是 0。如果在做除法前 B 中的值是 00H, 也就是除数为 0, 那么 OV=1。

加 1 指令

INC A

INC Rn

INC direct

INC @Ri

INC DPTR

用途很简单, 就是将后面目标中的值加 1。例: (A) = 12H, (R0) = 33H, (21H) = 32H, (34H) = 22H, DPTR = 1234H。执行下面的指令:

INC A (A) = 13H

INC R2 (R0) = 34H

INC 21H (21H) = 33H

INC @R0 (34H) = 23H

INC DPTR (DPTR) = 1235H

后结果如上所示。

说明: 从结果上看 INC A 和 ADD A, #1 差不多, 但 INC A 是单字节, 单周期指令, 而 ADD #1 则是双字节, 双周期指令, 而且 INC A 不会影响 PSW 位, 如 (A) = 0FFH, INC A 后 (A) = 00H, 而 CY 依然保持不变。如果是 ADD A, #1, 则 (A) = 00H, 而 CY 一定是 1。因此加 1 指令并不适合做加法, 事实上它主要是用来做计数、地址增加等用途。另外, 加法类指令都是以 A 为核心的, 其中一个数必须放在 A 中, 而运算结果也必须放在 A 中, 而加 1 类指令的对象则广泛得多, 能是寄存器、内存地址、间址寻址的地址等等。

减 1 指令

减 1 指令

DEC A

DEC Rn

DEC direct

DEC @Ri

与加 1 指令类似, 就不多说了。

综合练习:

## 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 A, #12H
MOV R0, #24H
MOV 21H, #56H
ADD A, #12H
MOV DPTR, #4316H
ADD A, DPH
ADD A, R0
CLR C
SUBB A, DPL
SUBB A, #25H
INC A
SETB C
ADDC A, 21H
INC R0
SUBB A, R0
MOV 24H, #16H
CLR C
ADD A, @R0
```

先写出每步运行结果，然后将以上题目建入，并在软件仿真中运行，观察寄存器及有关单元的内容的变化，是否与自己的预想结果相同。

### 51 实验板推荐(点击下面的图片可以进入下载资料链接)

