

第四章

例 4.12

通过以下两条指令将 80H 加 80H 后送到 AL，试写出运算结果及其标志位。

```
MOV    AL, 80H
```

```
ADD    AL, 80H
```

ADD 指令运算结束后：

- (AL) = 00H;
- ZF=1：因为运算结果为全零；
- CF=1：因为加法运算的最高位产生了进位，表明无符号数加运算发生溢出。实际上是：把操作数 80H 和 80H 均当作无符号数 128 和 128，做加法运算的正确结果为 256，超出了 8 位无符号数的表示范围（0~255）。
- OF=1：因为 $C_1 \oplus C_7 = 1$ ，按照单符号判溢方法，表明有符号数运算发生溢出。实际上是：把操作数 80H 和 80H 均当作有符号数（补码）-128 和 -128，做加法运算的正确结果为 -256，超出了 8 位补码机器数的表示范围（-128~+127）。
- SF=0：因为运算结果的最高位为 0。显然，因为 OF=1（发生了溢出），所以 SF 是错误的。
- PF=1：结果中“1”的个数为 0 个，所以 PF=1。

对于某个运算，CPU 在操作运算结果的 SF 符号位和 OF 溢出位时总是把操作数当作是有符号数，而在处理 CF 符号位时总是视其为无符号数，而只有程序员自己才知道究竟是想处理无符号数还是有符号数。譬如，假设上面两条指令后用 JC L1 指令（等价于 JB 指令）发生跳转，则表明程序员想处理无符号数。