第七章 分支与循环设计

1. 无条件转移指令

无条件转移指令是指能够无条件改变程序执行流程，改变 CPU 顺序执行下一条指令的模式，而将程序流程转移到代码段指定位置的指令，该指定的转移位置称为目标地址。

功能是通过无条件修改IP寄存器的内容或修改IP、CS寄存器的内容来实现的。

可以分类为段内直接短转移、段内直接长转移、段内间接转移、段间直接转移、段间间接转移。

但在汇编指令系统中，它们使用相同的指令助记符“JMP”，即 jump 之意。

1. 段内直接转移



第一种格式中的标号与 JMP 指令位于同一代码段，并且定义的标号类型为NEAR；第二种格式中的标号也与 JMP 指令位于同一代码段，但定义的标号类型允许为FAR，PTR 运算符将标号类型在 JMP 指令中暂时的修改为NEAR。

段内直接转移指令的机器指令由操作码字段、位移量字段（DISP 字段）构成

在汇编阶段（注意，指汇编程序将 我们的源程序翻译为目标代码的阶段），汇编程序会生成机器指令中的位移量，生成方法为：DISP = 标号的段内偏移量 − JMP指令后那条指令的段内偏移量 。

尽管这两个偏移量均是无符号数编码，由于标号的段内偏移量既可能大于 JMP 指令后那条指令的段内偏移量，也有可能小于它，因此所计算出的DISP可能为正数，也可能为负数，其编码一定是带符号数编码， 即补码。

将 16 位的无符号偏移量看作 17 位的正数补码，最高位即第16位是不可见的，我们假定它总为0，那么两个17位正数补码相减得到一个补码运算结果就不再是一件奇怪的事情了。

但是，与算术运算指令中的补码加、减运算类似，在计算DISP过程中也有可能出现补码运算溢出的情况，如果溢出发生，则汇编程序会在屏幕上报告一个语法错误， 指出JMP 指令的目标地址已超出其转移范围，程序员必须修改该错误后才能得到目标代码。

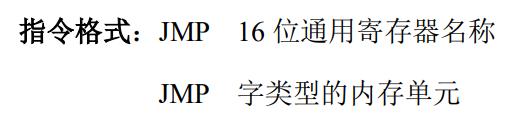
当所计算出的 DISP 位于-128~127 范围内时，它可用 8 位补码表示，此时汇编程序生成的机器指令为短转移指令，机器指令的操作码与位移量字段总共占据 2个字节。

而当所计算出的DISP超出-128~127 范围时，DISP 必须使用 16 位补码表示，此时汇编程序生成的机器指令为长转移指令，机器指令的操作码与位移量字段总共占据 3 个字节。段内直接转移最大范围为：向低地址端最大转移-32768字节（约 32K），向高地址端最大转移 32767字节（约 32K），若转移范围超出，则不能使用直接转移方式。

指令功能：(IP) + DISP ⇒ IP ，导致程序流程无条件地转移到目标地址，CPU 下一条即将执行的指令改变为目标地址处的指令。

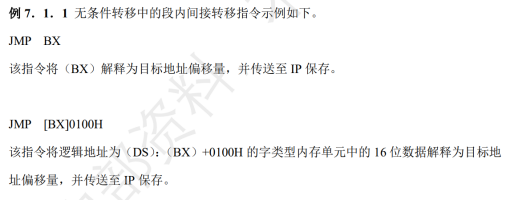
指令功能中所述操作是在程序执行阶段，即执行 JMP 指令时完成的。指令功能中修改（IP）的操作是一个加法运算，与生成 DISP 的原理类似，在作加法时将（IP）理解为 17 位正数补码，并且由于汇编程序生成 DISP 时是保证了没有溢出发生的，因此 JMP 指令中使用加法来修改（IP）同样不会导致溢出产生，我们应将所得到的加法结果重新理解为 16 位的无符号偏移量。

1. 段内间接转移



指令功能：(16位寄存器) ⇒ IP 或 （字类型的内存单元）⇒ IP ，JMP 指令将16位通用寄存器或字类型内存单元中的数据解释为目标地址偏移量，并将其传送至IP寄存器保存，从而使程序流程转移至目标地址。

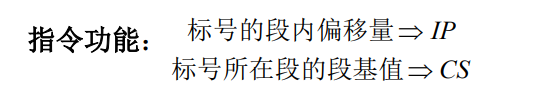
段内间接转移中没有目标地址超出转移范围的情况发生。



1. 段间直接转移



第一种格式中的标号与 JMP 指令不在同一个代码段中，并且定义类型为 FAR；第二种 格式中的标号与 JMP 指令可能在不同代码段，也可能在同一代码段，但其定义类型可能为NEAR，PTR 运算符将 JMP 指令中的标号类型暂时修改为 FAR。

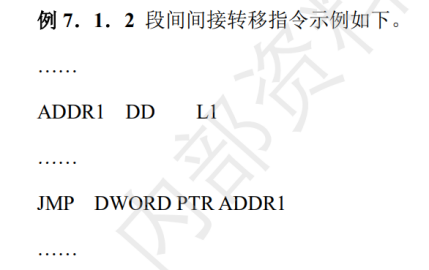


段间直接转移没有任何转移范围限制，理论上可在 1M 字节内存空间中任意实施流程转移。

1. 段间间接转移

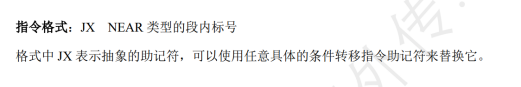


指令功能：将双字内存单元中的低地址字解释为目标地址偏移量，传送至 IP 保存，将双字内存单元中的高地址字解释为目标地址段基值，传送至 CS 保存，从而使程序流程转移至目标地址。与段间直接转移类似，由于使用完整逻辑地址作为目标地址，转移范围在 1M 字节 空间内不受限制。



1. 条件转移指令

在硬件上，多数条件转移指令所使用的判断条件都来自于状态标志位的当前状态，因此，我们可以称状态标志是实现分支、循环结构程序的硬件基础。



1. 单标志位转移指令

单标志位转移指令是指判断条件基于单个标志位状态的条件转移指令。这种转移指令一般成对出现。



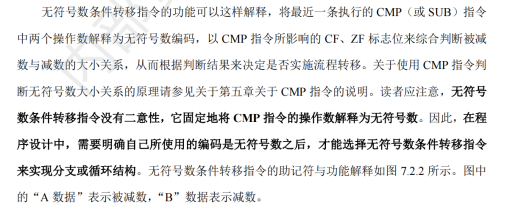


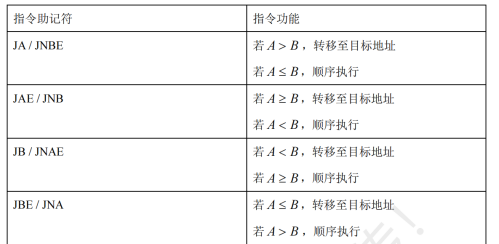
1. JCXZ指令

JCXZ 指令所判断的条件不是标志位的状态，而是 CX 寄存器中的数据，若 (CX ) = 0 则转移至目标地址，若 (CX ) ≠ 0 则顺序执行。在 8086/8088 指令系统中，该指令是唯一不以标志位为判断条件的条件转移指令。

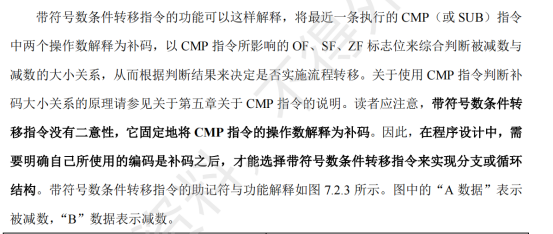
该指令的用途是配合循环控制指令形成计数循环结构，在进入循环体前用于判断计数器CX中的数据是否已经为零，如果已为零则不进入循环体，直接转移至循环出口。

1. 无符号数条件转移指令





1. 带符号数条件转移指令





1. 分支程序设计

