

## 微机原理与系统 B 第二周作业 9 月 22 日 周二

PB18151866 龚小航

6.11 比较 JMP DI 与 JMP [DI] 指令的操作。

解：JMP 指令是无条件跳转指令，转至指令所寻址的地址开始执行指令。第一条指令为寄存器寻址方式，第二条为寄存器间接寻址方式。DI 为目的地址寄存器，默认寻址数据段 DS。

JMP DI 指令表示把 DI 寄存器内存储的值赋给 IP，实现代码段内的近转移；

JMP [DI] 表示把内存单元中 DS:DI 地址处的值赋给 IP，实现代码段内的近转移。

6.25 解释 LOOPE 指令如何操作。【课本 148 页】

解：这是条件 LOOP 指令，如果 CX 不等于零而且等于条件成立，则 LOOPE(等于则循环) 指令转移；如果条件不成立或者 CX 寄存器减 1 后为 0，则跳出循环。LOOPE 指令和 LOOPZ 指令相同。

6.27 设计指令系列，在 100H 字节的存储块内检索。这个程序必须统计所有高于 42H 的无符号数的数目和低于 42H 的无符号数的数目。其中高于 42H 的计数值放在数据段存储单元 UP 中，而低于 42H 的计数值放在数据段单元 DOWN 中。【也可以使用 SCANB，课本 137 页】

解：指令系列如下所示： 其中已经省略了汇编文件的头尾，返回值等，只写出代码段中部分。

```
LEA DI, BLOCK    ;利用 DI 寻址
MOV UP, 0
MOV DOWN, 0
MOV CX, 100H
MOV AL, 42H
L1:
    CMP [DI], 42H
    JZ  L3
    JS  L2        ;大于 42H 的情况
    INC DOWN      ;小于 42H 的情况
    JMP L3
L2:
    INC UP
L3:  INC DI
    LOOP L1
```

6.41 写出求 EAX,EBX,ECX,EDX 之和的过程。如果出现进位,将逻辑 1 放入 EDI,如果不出现进位,将 0 放入 EDI。程序执行以后,和放在 EAX 中。

解: 程序如下所示:

```
MOV  EDI, 0 ;存储进位, 只要有进位就是 1
ADD  EAX, EBX
JNC  AC
MOV  EDI, 1

AC:
ADD  EAX, ECX
JNC  AD
MOV  EDI, 1

AD:
ADD  EAX, EDX
JNC  END
MOV  EDI, 1

END:
RET
```

6.47 IRET 指令和 RET 指令有什么区别? 【课本 156 页、158, 159 页】

解: RET 是普通的子程序的返回指令,从堆栈中取出 16 位数字(近返回)放入 IP 或取出 32 位数字(远返回)放入 IP 和 CS 中。在近返回时,处理器从堆栈中弹出 IP 或者 EIP,然后根据当前的 CS: IP 跳转到新的执行地址。如果之前压栈的还有其余的参数,则这些参数也会被弹出。

IRET 是中断返回指令,只用于软件或硬件的中断服务程序中。与简单的返回指令不同,IRET 指令能够弹出堆栈数据返还到 IP;弹出堆栈数据返还到 CS;弹出堆栈数据返还到标志寄存器。

这两条指令也有一定的关系:IRET 指令与后面跟随 POPF 的远 RET 指令实现相同的功能。

【补充题 1】 下列程序段执行完以后，程序转移分别到了哪里？

程序段1	程序段2
MOV AX, 147BH	MOV AX, 99D8H
MOV BX, 80DCH	MOV BX, 9847H
ADD AX, BX	SUB AX, BX
JNO L1	JNC L3
JNC L2	JNO L4

解：对这两个程序段分别分析：

① 跳转指令关于 of,cf, 写出 ADD 指令执行的结果以及标志寄存器的情况：

$AX + BX = 147B\text{H} + 80DC\text{H} = 9557\text{H}$

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

cf = 0(无进/借位);      of = 0(未溢出)

因此执行至 JNO 指令时即跳转成功，程序转移到了 L1 所指的地址继续执行。

② 跳转指令关于 of,cf, 写出 SUB 指令执行的结果以及标志寄存器的情况：

$$AX - BX = 99D8\text{H} - 9847\text{H} = 0191\text{H}$$

cf = 0(无进/借位);      of = 0(未溢出)

因此执行至 JNC 指令时即跳转成功，程序转移到了 L3 所指的地址继续执行。