汇编笔记

条件码标志flag

1. ZF 零标志

   运算结果为0，ZF=1.否则ZF-0

2. SF 符号标志

   若运算结果为负数，SF=1,否则SF=0

3. CF 进位标志

   若加法时（无符号数）最高位向前有进位或减法时最高位向前有借位，则CF=1,否则CF=0

4. OF 溢出标志

   若带符号数的运算结果超出了补码的表示范围，则OF=1，否则OF=0

进位和溢出的区别

进位标志表示无符号数运算结果是否超出范围，超出范围后加上进位或借位运算结果仍然正确； 溢出标志表示有符号数运算结果是否超出范围，超出范围后运算结果不正确。

---

1. AF 辅助进位标志

   若加法时结果低4位向前有进位或减法时低4位向前有借位，则AF=1，否则AF=0（用户无需关注）

2. PF 奇偶标志

   若结果最低字节（8位）中1的个数位偶数，则PF=1，否则PF=0

3. DF 方向标志

   用于串操作指令，控制地址的变化方向

   DF=0，储存器地址自动增加

   DF=1，储存器地址自动减少

   CLD指令复位方向标志：DF＝0 STD指令置位方向标志：DF＝1

4. IF 中断允许标志

   用于控制外部可屏蔽中断是否可以被处理器响应： 设置IF＝1，则允许中断； 设置IF＝0，则禁止中断。

   CLI指令复位中断标志：IF＝0 STI指令置位中断标志：IF＝1

5. TF 陷阱标志

   用于控制处理器进入单步操作方式： 设置TF＝0，处理器正常工作； 设置TF＝1，处理器单步执行指令。

寻址方式

1. 立即寻址

   只能用于源操作数，不能用于目的操作数。 源操作数 和 寄存器字长一致

2. 寄存器寻址

   这种方式执行速度快，因为不需访问内存。

   字节寄存器只有 AH AL BH BL CH CL DH DL

3. 内存寻址

   操作数是某个内存单元的值，指令中给出有效地址EA，段地址在某个段寄存器中

   1. 直接寻址
   2. 寄存器间接寻址。 [reg]

      reg只能是BX、BP、SI、DI，有效地址是reg的值

   3. 变址寻址。 disp [reg]

      disp可以是常量或变量，汇编后为一个常数，若是变量，则取其偏移地址。 reg只能是BX、BP、SI、DI。

   4. 基址变址寻址。 [base] [index]

      base为BX或BP，index为SI或DI。 有效地址=base+index

   5. 相对基址变址寻址。 disp[base] [index]

      有效地址(EA) =base+index+disp disp可以是常量或变量，汇编后为一个常数，若是变量，则取其偏移地址。 base为BX或BP，index为SI或DI。

注意

数值地址通常用作调试程序（debug)中，在源程序中要想使用数值地址，通常要给出段超越前缀，如DS:[1000H],否则汇编时会把它当成立即数处理。

计算出的有效地址以16位表示，若超过0FFFF，CPU将忽略溢出。

例如，设BX=2000H，SI=1000H，则 MOV AX,0F000H[BX] [SI] 源EA=0F000H+2000H+1000H=2000H

注意，高字节在前，低字节在后，注意十进制数的十六进制表示

指令系统

符号约定：

dest — 目的操作数 src — 源操作数 oprdn— 第n个操作数，如oprd1, oprd2, oprd3 = — 赋值 — 或者 reg8 — 8位通用寄存器AH/AL/BH/BL/CH/CL/DH/DL

reg16 — 16位通用寄存器AX/BX/CX/DX/SI/DI/BP/SP reg32 — 32位通用寄存器 EAX/EBX/ECX/EDX/ESI/EDI/EBP/ESP reg — reg8/reg16/reg32 seg — 段寄存器CS/DS/SS/ES/FS/GS mem8 — 8位内存操作数 mem16 — 16位内存操作数 mem32 — 32位内存操作数

mem — mem8/mem16/mem32 mem64 — 64位内存操作数 imm8 — 8位立即数 imm16 — 16位立即数 imm32 — 32位立即数 imm — imm8/imm16/imm32

PPT中出现的指令

重点掌握 MOV XCHG XLAT（翻译字节表查找） PUSH POP LEA（装载有效地址到寄存器）

请注意算术运算类指令对标志的影响 掌握： ADD/ADC/INC、SUB/SBB/DEC/ NEG/CMP 理解：MUL/IMUL、DIV/IDIV 了解：CBW/CWD

逻辑指令 一般形式： AND dest, src ; dest = dest and src OR dest, src ; dest = dest or src XOR dest, src ; dest = dest xor src NOT dest ; dest = not dest TEST dest, src ; dest and src，执行AND操作但不存储结果到dest

移位指令 移位指令包括： SHL（Shift Left）： 逻辑左移 SAL（Shift Arithmetic Left）： 算术左移 SHR（Shift Right）： 逻辑右移 SAR（Shift Arithmetic Right）： 算术右移

循环移位 循环移位指令包括： ROL（Rotate Left）： 循环左移 ROL dest, count ; dest循环左移，count为移位次数 ROR（Rotate Right）： 循环右移 ROR dest, count RCL（Rotate through Carry Left）： 带进位循环左移 RCL dest, count RCR（Rotate through Carry Right）： 带进位循环右移 RCR dest, count

控制转移类指令用于实现分支、循环、过程等程序结构，是仅次于传送指令的常用指令 重点掌握：JMP/Jcc/LOOP CALL/RET INT n/IRET 常用系统功能调用

串操作指令是8086指令系统中比较独特的一类指令，采用比较特殊的数据串寻址方式，常用在操作主存连续区域的数据时有两类 串传送：MOVS STOS LODS REP 串比较：CMPS SCAS REPZ REPNZ

重复前缀分2类，3条指令： 配合不影响标志的MOVS、STOS（和LODS）指令的REP前缀 配合影响标志的CMPS和SCAS指令的REPZ和REPNZ前缀

01H：从键盘读入一个字符 02H：显示一个字符 09H：显示一个字符串 0AH：从键盘读入一个字符串

标志处理指令包括 CLC（Clear Carry Flag，CF清0） STC（Set Carry Flag，CF置1） CMC（Complement Carry Flag，CF取反）、CLD（Clear Direction Flag，DF清0） STD（Set Direction Flag，DF置1） CLI（Clear Interrupt Flag，关中断） STI（Set Interrupt Flag，开中断）。

1．NOP（No Operation）：无操作

XCHG

XLAT（翻译字节表查找）

PUSH

POP

LEA（装载有效地址到寄存器）