

第四章

- 对操作数的要求
- 尤其应注意算术运算指令对标志位的影响
- 无符号和有符号数的区别

CF=1, 则表示无符号数运算产生溢出

OF=1, 则表示有符号数运算产生溢出

算术运算

加法

ADD (ADDition) 加法指令

格式: ADD DST, SRC

执行的操作: $(DST) \leftarrow (SRC) + (DST)$

说明:

目的操作数: 8/16/32位的寄存器/存储器

源操作数: 与目的操作数同类型的reg/mem/imm

指令执行的结果影响标志位AF、CF、OF、PF、SF和ZF。

ADC (ADd with Carry) 带进位加法指令

格式: ADC DST, SRC

执行的操作: $(DST) \leftarrow (SRC) + (DST) + CF$

这条指令主要用于多字节(或多字)相加。

[例3] 两个无符号的双精度数(双字)的加法的指令序列。

ADD AX, CX ; 低位字相加

ADC DX, BX ; 高位字带进位相加

INC (INCrement) 加1指令

格式: INC OPR

执行的操作: $(OPR) \leftarrow (OPR) + 1$

说明: 其中OPR可以是reg/mem, 但不能是立即数和段寄存器操作数。

该指令影响OF、SF、ZF、PF和AF, 但是CF标志位不受影响。

减法指令

SUB (SUBtruction) 减法指令

格式: SUB DST, SRC

执行的操作: $(DST) \leftarrow (DST) - (SRC)$

说明: DST和SRC的寻址方式的规定同ADD指令

指令执行的结果影响标志位AF、CF、OF、PF、SF和ZF。

SBB (SuBtruct with Borrow) 带借位减法

格式: SBB DST, SRC

执行的操作: $(DST) \leftarrow (DST) - (SRC) - CF$

主要用于多字节相减。

指令执行的结果影响条件标志位。

DEC (Decrement) 减1指令

格式: DEC OPR

执行的操作: $(OPR) \leftarrow (OPR) - 1$

说明: 其中OPR可以是寄存器和存储器操作数, 但不能是立即数和段寄存器操作数。

该指令影响OF、SF、ZF、PF和AF, 但是CF标志位不受影响(同INC)

NEG (Negate) 求补指令

格式: NEG OPR

执行的操作: $(OPR) \leftarrow (OPR)$ 补

或 $(OPR) \leftarrow 0 \text{ -- } (OPR)$

对操作数的规定同INC指令

影响6个标志位, 只有对0求补时, $CF=0$, 其余 $CF=1$

CMP (Compare) 比较指令

格式: CMP OPR1, OPR2

执行的操作: $(OPR1) - (OPR2)$

该指令同SUB指令一样执行减法运算, 唯一不同的是, 该指令不保存结果。

指令执行的结果影响标志位。

乘法指令

MUL 无符号数乘法指令

格式: MUL SRC ;B/W

执行的操作: $AX = AL \times (SRC)$; 字节操作

DX、 $AX = AX \times (SRC)$; 字操作

其中: 隐含目的操作数

SRC可以是寄存器操作数和内存操作数

SRC不能使用立即数和段寄存器

IMUL 带符号数乘法指令

参加乘法运算的是有符号数, 格式、操作方法和规定同MUL指令

乘法指令对标志位的影响

除CF、OF以外无定义。

- 对MUL :

乘积的高一半是0, $CF=OF=0$, 否则为1。

- 对IMUL

乘积的高一半是低一半的符号扩展, $CF=OF=0$, 否则为1。

除法和符号扩展指令

•DIV 无符号数除法指令

格式: DIV SRC ;B/W

执行的操作: $AL \leftarrow AX / (SRC)$ 的商 ; 字节操作

$AH \leftarrow AX / (SRC)$ 的余数

$AX \leftarrow DXAX / (SRC)$ 的商 ; 字操作

$DX \leftarrow DXAX / (SRC)$ 的余数

其中: SRC的规定同乘法指令

DIV指令的被除数、除数、商和余数都是无符号数。

•IDIV 带符号数除法指令

格式: IDIV SRC ;B/W

执行的操作: 与DIV相同, 即:

AL \div AX/(SRC)的商 ; 字节操作

AH \div AX/(SRC)的余数

AX \div DXAX'(SRC)的商 ; 字操作

DX \div DXAX'(SRC)的余数

其中: SRC的规定同乘法指令

IDIV指令的被除数、除数、商和余数都是带符号数

余数的符号同被除数的符号

除法指令对标志位的影响

标志位AF、CF、OF、PF、SF、ZF

均无定义

1.用IDIV指令时, 商的范围应在-128~127或-32768~32767。若超出该范围则作为除数为0处理, 产生0号中断, 而不会使OF=1。

2.用IDIV指令时, 余数的符号一定要和被除数相同。例如-30,8, 得到商为-3, 余数为-6, 而不是商-4, 余2。

3. 除法运算中, 要求16位数除以8位数, 当被除数只有8位时, 必须将此8位数据放在AL中, 并对高8位AH进行符号扩展。同样, 要求32位数除以16位数, 当被除数只有16位时, 必须将此16位数据放在AX中, 并对高16位DX进行符号扩展。

如果不对AH或DX进行符号扩展, 可能会得到错误的结果。

4. 除法运算中的符号扩展:

•对于无符号数除法指令DIV:

只要将AH或DX寄存器清0即可

•对于带符号数除法指令IDIV:

AH或DX的扩展就是低位字节(AL) 或低位字(AX) 的符号扩展, 具体是:

把AL中的最高位扩展到AH的8位中(正数为00H, 负数为FFH);

把AX中的最高位扩展到DX的16位中（正数为0000H，负数为FFFFH）；

•CBW字节转换为字指令

(Convert Byte to Word)

格式：CBW

执行的操作：AL中的符号位扩展到8位的AH中。若AL中最高位为0，则AH=00H；若AL中最高位为1，则AH=FFH；

•CWD字转换为双字指令

(Convert Word to Double-word)

格式：CBW

执行的操作：AX中的符号位扩展到16位的DX中。若AX中最高位为0，则DX=0000H；若AX中最高位为1，则DXH=FFFFH；

符号扩展指令不影响标志位

BCD数运算

算术运算指令都是二进制数的运算，为了便于十进制运算，8088指令系统有一组用于十进制调整的指令，将二进制的运算结果调整得到十进制的结果。

•压缩的BCD码

用4位二进制数表示一个十进制数位，整个十进制数形成为一个顺序的以4位为一组的数串。如

4567的压缩型BCD码表示为4567H

或 0100 0101 0110 0111B

•非压缩的BCD码

以8位二进制数表示一个十进制数位，8位中的低4位表示BCD码，而高4位没有意义。如

4567的非压缩型BCD码表示为04050607H等

或 ''''0100''''0101''''0110''''0111B

逻辑运算

逻辑运算指令的特点:

- 1.按位操作
- 2.NOT为单操作数指令，不允许**立即数和段寄存器**。
- 3.其余为双操作数指令，操作数的规定同算术运算指令。

逻辑指令对标志位的影响

NOT指令不影响标志位

AND、OR、XOR、TEST指令：

使CF=OF=0，AF无定义

影响SF、ZF、PF。

AND 逻辑与指令

格式：AND DST, SRC ;B/W

执行的操作：

$(DST) = (DST) \text{AND} (SRC)$

用途：

使某操作数中的若干位不变、若干位清0，称之为屏蔽某些位。

OR 逻辑或指令

XOR 逻辑异或指令

格式：XOR DST, SRC ;B/W

执行的操作： $(DST) \leftarrow (DST) \text{XOR} (SRC)$

用途：

可使某操作数清0，同时使CF=0；

也可使某个操作数中的若干位不变、若干位取反。

NOT 逻辑非指令

格式：NOT OPR ;B/W

执行的操作：

(OPR) $\bar{U}(\text{OPR})$ 取反

注意:

不允许立即数和段寄存器;

不影响标志位。

TEST 测试指令

格式: TEST DST, SRC ;B/W

执行的操作:

(DST)**AND**(SRC), 不回送结果, 影响标志位

用途:

在不希望改变原有的操作数的情况下, 用来检测某一位或某几位是否满足条件, 可作为条件转移指令的先行指令

移位和循环移位

一般移位指令

SHL SHR SAL SAR

循环移位

ROL ROR RCL RCR

双精度移位指令 (386以上)

SHLD SHRD

- SHL (SHift logical Left) 逻辑左移
- SHR (SHift logical Right) 逻辑右移
- SAL (Shift Arithmetic Left) 算术左移
- SAR (Shift Arithmetic Right) 算术右移
-
- ROL (ROtate Left) 循环左移
- ROR (ROtate Right) 循环右移

- RCL (Rotate Left through Carry) 带进位循环左移
- RCR (Rotate Right through Carry) 带进位循环右移
-
- SHLD (Shift Left Double) 双精度左移
- SHRD (Shift Right Double) 双精度右移

格式

格式: SHL OPR, CNT ;B/W

- OPR为通用寄存器和内存操作数, 不允许立即数和段寄存器;
- 移位次数由CNT决定, CNT可取1或CL寄存器数 (8086) ,

逻辑右移单纯右移,算术移动不变符号位,相当于乘除

移位对标志位的影响

- 一般移位指令影响标志位CF、OF、PF、SF和ZF。
- 循环移位指令只影响CF和OF。
- AF无定义
- OF位只有当CNT=1时才有效。在执行一位移位后, 如果移位前后符号位发生了变化, 则OF=1, 否则OF=0。

cpu控制指令

CLC CF =0

STC CF =1

CMC CF取反

CLD DF =0

STD DF =1

CLI IF =0

STI IF =1

HLT：暂停指令：使CPU暂停程序执行，进入暂停状态。复位信号RESET可使CPU退出暂停状态。

NOP：空操作：不执行如何对数据的操作，但是使用了3个时钟周期，1个字节，并使IP加1。用于软件延时或者在程序调试时取代其他指令的位置。

输入输出

输出单个字符

DL \leftarrow 待输出字符的ASCII代码

AH \leftarrow 02H

INT 21H

输出字符串

输出一个字符串

DS: DX \leftarrow 待输出字符串的首地址

AH \leftarrow 09H

INT 21H

字符串以字符"\$"为结束标志，该字符本身不输出。

字符输入

字符的输入

(1) AH \leftarrow 01

INT 21H

键盘输入字符后返回，ASCII码在AL中，并回显。

(2) AH \leftarrow 07

INT 21H

键盘输入字符后返回，ASCII码在AL中，无回显。

(3) AH \leftarrow 08

INT 21H

键盘输入字符后返回，ASCII码在AL中，无回显。同时检测 Ctrl+Break和 Ctrl+C键的组合。

输入一个字符串

DS: DX ← 输入缓冲区首地址

AH ← 0AH

INT 21H

输入缓冲区格式：

BUFFER DB 81, ?, 81 DUP(?)

字符串操作指令

串操作指令的共同特点：

- 源操作数由DS: [SI]提供（有时也可以由AL、AX、EAX提供）；

目的操作数由ES: [DI]提供

- 每执行一次串操作，自动修改SI和（或）DI，使其指向下一个字节、字或双字
- 方向标志DF控制对SI、DI递增或是递减

指令STD将DF置1，指令CLD将DF清0

与无条件重复前缀REP配合使用的指令

1. MOVS 字符串传送指令

- 格式：MOVSB / MOVSW / MOVSD

- 执行的操作：

(1) 目的操作数单元 ← 源操作数，即ES:[DI] ← DS:[SI]

(2) 修改SI和DI值

2. REP 串重复操作前缀

- 格式：REP 串操作指令

- 执行的操作：重复执行串操作指令，直到CX的值为零

如果 $(CX) \neq 0$:

(1) $(CX) = (CX) - 1$

(2) 执行串指令

(3) 转向 (1)

如果 $(CX) = 0$ ，结束本指令操作

3. STOS 存字符串指令

- 格式：STOSB/ STOSW/STOSD

- 执行的操作：

(1) 目的存储单元 \leftarrow 累加器

字节操作：ES:[DI] \leftarrow (AL)

字操作：ES:[DI] \leftarrow (AX)

双字操作：ES:[DI] \leftarrow (EAX)

(2) 修改DI值IN

STOS 可以与REP联用

LODS 取字符串指令

- 格式：LODSB/ LODSW/LODSD

- 执行的操作：

(1) 累加器 \leftarrow 源存储单元

字节操作：(AL) \leftarrow DS:[SI]

字操作：(AX) \leftarrow DS:[SI]

双字操作：(EAX) \leftarrow DS:[SI]

(2) 修改SI值

与有条件重复前缀配合使用的指令

1. CMPS 串比较指令

- 格式：CMPSB/CMPSW/CMPSD

- 执行的操作：

(1) 源操作数 - 目的操作数

即：(DS:[SI]) - (ES:[DI]) ,

不保存减法得到的差，但产生新的状态标志

(2) 修改SI和DI值

2. REPZ / REPE 当为零 / 相等时重复操作前缀

- 格式：REPZ / REPE 串指令

- 执行的操作：如ZF = 1且 (CX) ≠ 0，重复执行串指令

(1) 如ZF = 1且 (CX) ≠ 0，则：

(a) 执行串操作指令

(b) (CX) = (CX) - 1

(c) 返回 (1)

(2) ZF = 0或(CX) = 0，执行“REPE/REPZ 串指令”的后续指令。

REPNZ / REPNE 当不为零 / 不相等时重复操作前缀

- 该前缀与REPZ / REPE对标志位ZF的判别条件刚好相反

- 格式：REPNZ / REPNE 串指令

- 执行的操作：如果ZF = 0且 (CX) ≠ 0，重复串操作

如ZF = 0且 (CX) ≠ 0，则：

(1) 执行串操作指令

(2) (CX) = (CX) - 1

ZF = 1或 (CX) = 0, 停止执行本指令, 执行下一条指令

SCAS 串扫描指令

- 格式: SCASB/ SCASW/SCASD

- 执行的操作:

- (1) 累加器 - 目的操作数

- 字节操作: (AL) - (ES:[DI])

- 字操作: (AX) - (ES:[DI])

- 双字操作: (EAX) - (ES:[DI])

- (2) 修改DI值