

数据传送指令

- 标志寄存器传送指令

标志送AH指令: LAHF (Load AH with Flags)

执行操作: $(AH) \leftarrow (\text{FLAGS的低字节})$

AH送标志寄存器指令: SAHF (Save AH into Flags)

执行操作: $(\text{FLAGS的低字节}) \leftarrow (AH)$

标志进栈指令: PUSHF

执行操作: $(SP) \leftarrow (SP) - 2$

$((SP)+1, (SP)) \leftarrow (\text{FLAGS})$

标志出栈指令: POPF

执行操作: $(\text{FLAGS}) \leftarrow ((SP)+1, (SP))$

$(SP) \leftarrow (SP) + 2$

* SAHF, POPF影响标志位

				OF	DF	IF	TF	SF	ZF		AF		PF		CF
--	--	--	--	----	----	----	----	----	----	--	----	--	----	--	----

数据传送指令

- 类型转换指令

CBW AL → AX (AL中的内容符号扩展到AX)

执行操作: 若(AL)的最高有效位为0, 则(AH)= 00H

 若(AL)的最高有效位为1, 则(AH)= FFH

CWD AX → (DX, AX) (AX中的内容符号扩展到DX: AX)

执行操作: 若(AX)的最高有效位为0, 则(DX)= 0000H

 若(AX)的最高有效位为1, 则(DX)= FFFFH

例: (AX) = 0BA45H

CBW ; (AX)=0045H

CWD ; (DX)=0FFFFH (AX)=0BA45H

注意: * 无操作数指令
 * 隐含对AL 或AX 进行符号扩展
 * 不影响条件标志位

第4&5讲：8086/8088的指令系统

- 8086/8088指令概述
- 数据传送指令
- 算术指令
- 逻辑指令
- 串处理指令
- 处理机控制与杂项操作指令

算术指令

- 加法指令
ADD、ADC、INC
- 减法指令
SUB、SBB、DEC、NEG、CMP
- 乘法指令
MUL、IMUL
- 除法指令
DIV、IDIV

算术指令

- 加法指令

加法指令: `ADD DST, SRC`
执行操作: $(DST) \leftarrow (SRC) + (DST)$

带进位加法指令: `ADC DST, SRC`
执行操作: $(DST) \leftarrow (SRC) + (DST) + CF$

加1指令: `INC OPR`
执行操作: $(OPR) \leftarrow (OPR) + 1$

注意:

* 除INC指令不影响CF标志外, 均对条件标志位有影响。

算术指令

举例: $n=8$ bit 带符号数(-128~127), 无符号数(0~255)

```
  0 0 0 0 0 1 0 0
+ 0 0 0 0 1 0 1 1
-----
  0 0 0 0 1 1 1 1
```

带: $(+4)+(+11)=+15$ OF=0

无: $4+11=15$ CF=0

带符号数和无符号数都不溢出

```
  0 0 0 0 0 1 1 1
+ 1 1 1 1 1 0 1 1
-----
  1 0 0 0 0 0 1 0
```

带: $(+7)+(-5)=+2$ OF=0

无: $7+251=258$

现为: 2 CF=1 ?

无符号数溢出

```
  1 0 0 0 0 1 1 1
+ 1 1 1 1 0 1 0 1
-----
  1 0 1 1 1 1 1 0 0
```

带: $(-121)+(-11)=-132$

现为: 124 OF=1 ?

无: $135+245=380$

现为: 124 CF=1 ?

带符号数和无符号数都溢出

```
  0 0 0 0 1 0 0 1
+ 0 1 1 1 1 1 0 0
-----
  1 0 0 0 0 1 0 1
```

带: $(+9)+(+124)=133$

现为: -123 OF=1 ?

无: $9+124=133$ CF=0

带符号数溢出

算术指令

对无符号数，考虑进位标志CF，则结果正确。

$$\begin{array}{r} 0000\ 0100 \\ +\ 0000\ 1011 \\ \hline 0000\ 1111 \end{array}$$

带: $(+4)+(+11)=+15$ OF=0

无: $4+11=15$ CF=0

带符号数和无符号数都不溢出

$$\begin{array}{r} 1000\ 0111 \\ +\ 1111\ 0101 \\ \hline 1\ 0111\ 1100 \end{array}$$

带: $(-121)+(-11)=-132$

现为: 124 OF=1 ?

无: $135+245=380$

现为: 124 CF=1 ✓

带符号数和无符号数都溢出

$$\begin{array}{r} 0000\ 0111 \\ +\ 1111\ 1011 \\ \hline 1\ 0000\ 0010 \end{array}$$

带: $(+7)+(-5)=+2$ OF=0

无: $7+251=258$

现为: 2 CF=1 ✓

无符号数溢出

$$\begin{array}{r} 0000\ 1001 \\ +\ 0111\ 1100 \\ \hline 1000\ 0101 \end{array}$$

带: $(+9)+(+124)=133$

现为: -123 OF=1 ?

无: $9+124=133$ CF=0

带符号数溢出

算术指令

```
  1 0 0 0 0 1 1 1
+ 1 1 1 1 0 1 0 1
-----
  1 0 1 1 1 1 0 0
```

带: $(-121)+(-11)=-132$
现为: 124 **OF=1 ?**

无: $135+245=380$
现为: 124 **CF=1 ✓**

带符号数和无符号数都溢出

```
  0 0 0 0 1 0 0 1
+ 0 1 1 1 1 1 0 0
-----
  1 0 0 0 0 1 0 1
```

带: $(+9)+(+124)=133$
现为: -123 **OF=1 ?**

无: $9+124=133$ **CF=0**

带符号数溢出

对带符号数，若溢出标志OF为1，则计算结果错误。

溢出产生的原因是所表示的数超过的所给位数的最大能力，8位带符号数的表示范围在-128~+127。

处理方法：增加位数。右侧例题只要把位数扩展到16位就不会溢出了。

扩展方法：若最高位为0，则用0扩展，若最高位为1，则用1扩展。

算术指令

加法指令对**条件标志位**的影响

$$\begin{aligned} \text{SF} &= \begin{cases} 1 & \text{结果为负} \\ 0 & \text{否则} \end{cases} & \text{ZF} &= \begin{cases} 1 & \text{结果为0} \\ 0 & \text{否则} \end{cases} \\ \\ \text{CF} &= \begin{cases} 1 & \text{和的最高有效位有向高位的进位} \\ 0 & \text{否则} \end{cases} \\ \\ \text{OF} &= \begin{cases} 1 & \text{两个操作数符号相同，而结果符号与之相反} \\ 0 & \text{否则} \end{cases} \end{aligned}$$

CF 位表示 无符号数 相加的溢出。

OF 位表示 带符号数 相加的溢出。

算术指令

例：双精度数的加法DX、AX与BX、CX，两个双字长的数相加。DX、BX分别存放高位字。

(DX) = 0002H (AX) = 0F365H
(BX) = 0005H (CX) = 0E024H

指令序列 ADD AX, CX ; (1)
 ADC DX, BX ; (2)

(1) 执行后, (AX) = 0D389H
 CF= OF= SF= ZF=

(2) 执行后, (DX) = 0008H
 CF= OF= SF= ZF=

算术指令

例：双精度数的加法DX、AX与BX、CX，两个双字长的数相加。DX、BX分别存放高位字。

(DX) = 0002H (AX) = 0F365H
(BX) = 0005H (CX) = 0E024H

指令序列 ADD AX, CX ; (1)
 ADC DX, BX ; (2)

(1) 执行后, (AX) = 0D389H
 CF=1 OF=0 SF=1 ZF=0

(2) 执行后, (DX) = 0008H
 CF=0 OF=0 SF=0 ZF=0

算术指令

- 减法指令

减法指令: SUB DST, SRC
执行操作: $(DST) \leftarrow (DST) - (SRC)$

带借位减法指令: SBB DST, SRC
执行操作: $(DST) \leftarrow (DST) - (SRC) - CF$

减1指令: DEC OPR
执行操作: $(OPR) \leftarrow (OPR) - 1$

求补指令: NEG OPR
执行操作: $(OPR) \leftarrow 0FFFFH - (OPR) + 1$

比较指令: CMP OPR1, OPR2
执行操作: $(OPR1) - (OPR2)$, 不保存结果, 只是根据结果设置标志位。

注意:
* 除DEC指令不影响CF标志外, 均对条件标志位有影响。

算术指令

减法指令对条件标志位 (CF/OF/ZF/SF) 的影响:

CF= $\begin{cases} 1 & \text{被减数的最高有效位有向高位的借位} \\ 0 & \text{否则} \end{cases}$

或转换成加法:

CF= $\begin{cases} 1 & \text{减法转换为加法运算时无进位} \\ 0 & \text{否则} \end{cases}$

OF= $\begin{cases} 1 & \text{两个操作数符号相反, 而结果的符号与减数相同} \\ 0 & \text{否则} \end{cases}$

或转换成加法:

OF= $\begin{cases} 1 & \text{两个操作数符号相同, 而结果符号与之相反} \\ 0 & \text{否则} \end{cases}$

CF 位表示 无符号数 减法的溢出。

OF 位表示 带符号数 减法的溢出。

算术指令

NEG 指令对CF/OF的影响

$$CF = \begin{cases} 0 & \text{操作数为0} \\ 1 & \text{否则} \end{cases}$$

$$OF = \begin{cases} 1 & \text{操作数为 -128 (字节运算) 或} \\ & \text{操作数为 -32768 (字运算)} \\ 0 & \text{否则} \end{cases}$$

$$\begin{array}{r} 10000000 \\ 01111111 \\ + 00000001 \\ \hline 10000000 \end{array}$$

算术指令

例：SUB [SI+14H], 0136H

若 (DS) = 3000H, (SI) = 0040H, (30054H) = 4336H

则，指令执行后：(30054H) =

SF= , CF= , OF= , ZF=

SUB DH, [BP+4]

若 (SS) = 0000H, (BP) = 00E4H,

(DH) = 41H, (000E8) = 5AH

则，指令执行后：(DH) =

SF= , CF= , OF= , ZF=

算术指令

例：SUB [SI+14H], 0136H

若 (DS) =3000H, (SI) =0040H, (30054H) =4336H

则，指令执行后：(30054H) =4200H

SF=0, CF=0, OF=0, ZF=0

SUB DH, [BP+4]

若 (SS) =0000H, (BP) =00E4H,

(DH) =41H, (000E8) =5AH

则，指令执行后：(DH) =0E7H

SF=1, CF=1, OF=0, ZF=0

算术指令

例：x、y、z 均为双精度数，分别存放在地址为X, X+2; Y, Y+2; Z, Z+2的存储单元中，存放时，高字节存放在高地址中，低字节存放在低地址中，用指令序列实现：

$W \leftarrow x + y + 24 - z$ ，并用W, W+2单元存放W

MOV AX, X

MOV DX, X+2

ADD AX, Y

ADC DX, Y+2 ; x+y

ADD AX, 24

ADC DX, 0 ; x+y+24

SUB AX, Z

SBB DX, Z+2 ; x+y+24-z

MOV W, AX

MOV W+2, DX ; 结果存入W, W+2单元

算术指令

- 乘法指令

无符号数乘法指令: `MUL SRC`

带符号数乘法指令: `IMUL SRC`

执行操作:

字节操作数 $(AX) \leftarrow (AL) * (SRC)$

字操作数 $(DX, AX) \leftarrow (AX) * (SRC)$

注意:

- * `AL (AX)` 为隐含的乘数寄存器。
- * `AX (DX,AX)` 为隐含的乘积寄存器。
- * `SRC` 不能为立即数。
- * 除 `CF` 和 `OF` 外, 对条件标志位无定义。

算术指令

注意：MUL和IMUL指令的使用条件是由数的格式决定的。如：
 $(11111111b) * (11111111b)$

若是无符号数： $255d * 255d = 65025d$

若是带符号数： $(-1) * (-1) = 1$

因此，必须根据所要相乘的数的格式来决定选用哪一种命令。

算术指令

例：如 (AL) =0B4H, (BL) =11H, 求：

IMUL BL

MUL BL

分析：

(AL) =0B4H为无符号数的180，有符号数的-76；

(BL) =11H为无符号数的17，有符号数的17.

因此：

MUL BL的执行结果为：(AX) =0BF4H (3060)

IMUL BL的执行结果为：(AX) =0FAF4H (-1292)

算术指令

乘法指令对 CF/OF 的影响:

MUL指令: CF, OF =	$\begin{cases} 00 & \text{乘积的高一半为零} \\ 11 & \text{否则} \end{cases}$
IMUL指令: CF, OF =	$\begin{cases} 00 & \text{乘积的高一半是低一半的符号扩展} \\ 11 & \text{否则} \end{cases}$

例: (AX) = 16A5H, (BX) = 0611H

1010 0101

0101 1011

- (1) IMUL BL ; (AX) \leftarrow (AL) * (BL)
; A5*11 \Rightarrow 5B*11=060B \Rightarrow F9F5
; (AX) = 0F9F5H CF=OF=1
- (2) MUL BX ; (DX,AX) \leftarrow (AX) * (BX)
; 16A5*0611=0089 5EF5
; (DX)=0089H (AX)=5EF5H CF=OF=1

算术指令

- 除法指令

无符号数除法指令: `DIV SRC`

带符号数除法指令: `IDIV SRC` (余数的符号和被除数相同)

执行操作:

字节操作 $(AL) \leftarrow (AX) / (SRC)$ 的商
 $(AH) \leftarrow (AX) / (SRC)$ 的余数

字操作 $(AX) \leftarrow (DX, AX) / (SRC)$ 的商
 $(DX) \leftarrow (DX, AX) / (SRC)$ 的余数

注意:

- * `AX (DX,AX)` 为隐含的被除数寄存器。
- * `AL (AX)` 为隐含的商寄存器。
- * `AH (DX)` 为隐含的余数寄存器。
- * `SRC` 不能为立即数。
- * 对所有条件标志位均无定义。

算术指令

- 思考：

```
MOV AX, 2EE0H    ;12000
MOV BL, 8
DIV  BL
```

计算结果：
商: 1500=>AL
余数: 0=>AH

?

问题：

只要字节除法操作AH大于等于除数、或者字除法操作DX大于等于除数，都会发生除法溢出！

如何解决？

操作数先扩展，再做除法。

```
MOV AX, 2EE0H    ;12000
CWD              ???
MOV BX, 8
DIV  BX
```

计算结果：

商: 1500=>AX
余数: 0=>DX

算术指令

例：x, y, z, v 均为16位带符号数，计算 $(v-(x*y+z-540))/x$

```
MOV  AX, X
IMUL Y      ; x*y → (DX,AX)
MOV  CX, AX
MOV  BX, DX
MOV  AX, Z
CWD          ; Z → (DX, AX)
ADD  CX, AX
ADC  BX, DX  ; x*y+z → (BX, CX)
SUB  CX, 540
SBB  BX, 0   ; x*y+z-540
MOV  AX, V
CWD          ; V → (DX, AX)
SUB  AX, CX
SBB  DX, BX  ; v-(x*y+z-540)
IDIV X      ; (v-(x*y+z-540))/x → (AX), 余数 → (DX)
```