

3.5 算术运算类指令



- 算术运算类指令可以分成6个子类:

算术运算指令	加减法指令	ADD, ADC, SUB, SBB, NEG
	比较指令	CMP
	增量减量指令	INC, DEC
	乘除法指令	MUL, IMUL, DIV, IDIV
	符号扩展指令	CBW, CWD
	BCD数运算调整指令	AAA, DAA, AAS, DAS, AAM, AAD

注：只要**ALU**涉及到运算，就不能使用段**REG**，其运算结果会影响**6**个状态标志位。

3.5 算术运算类指令



3.5.1 加减法指令

1. 加法指令（ADD、ADC）

格式： **ADD DST, SRC;** $(DST) \leftarrow (SRC) + (DST)$

ADC DST, SRC; $(DST) \leftarrow (SRC) + (DST) + (CF)$

说明：

- **SRC**可以取立即数、通用寄存器和存储单元，**DST**可以取通用寄存器和存储单元，但**SRC**和**DST**不能同时取存储单元。
- **ADD**和**ADC**指令会正常影响**PSW**中的6个状态标志位：**CF**、**AF**、**OF**、**SF**、**ZF**和**PF**。

3.5 算术运算类指令



例：设变量VAR1为字型变量，VAR2为字节变量，则

ADD AX, 56A0H ; (AX) \leftarrow (AX)+56A0H

ADC AX, BX ; (AX) \leftarrow (AX)+(BX) + (CF)

ADD AX, VAR1 ; (AX) \leftarrow (AX)+ (VAR1)

ADD BX, VAR1[DI] ; (BX) \leftarrow (BX)+ ((VAR1) +(DI))

ADD BL, VAR2 ; (BL) \leftarrow (BL)+(VAR2)

ADD BYTE PTR VAR1, AL

; (VAR1的低位字节) \leftarrow (VAR1的低位字节)+(AL)

3.5 算术运算类指令



状态标志位影响：

例：设 $(AX) = 5439H$ ，则执行 **ADD AX, 476AH** 后，

$(AX) = 9BA3H$,

0101 0100 0011 1001 B

+ 0100 0111 0110 1010 B

1001 1011 1010 0011 B

CF=0, AF=1, OF=1, SF=1, ZF=0, PF=1。

3.5 算术运算类指令



2. 减法指令 (**SUB**、**SBB**)

格式: **SUB DST, SRC;** $(DST) \leftarrow (DST) - (SRC)$

SBB DST, SRC; $(DST) \leftarrow (DST) - (SRC) - (CF)$

说明:

- 指令中**DST**、**SRC**的来源与**ADD**指令相同;
- 减法指令**SUB**和**SBB**指令会正常影响**PSW**中的6个状态标志位: **CF**、**AF**、**OF**、**SF**、**ZF**和**PF**。

3.5 算术运算类指令



例：设VAR1为字型变量，VAR2为字节变量，则

SUB AL, 2CH ; (AL) \leftarrow (AL) - 2CH

SBB AX, BX ; (AX) \leftarrow (AX) - (BX) - (CF)

SUB AX, VAR1 ; (AX) \leftarrow (AX) - (VAR1)

SBB DL, VAR2[SI] ; (DL) \leftarrow (DL) - ((VAR2) + (SI)) - (CF)

3.5 算术运算类指令



例: **MOV AX, 565BH**

SUB AX, C546H

结果: **(AX) = 9115H**

变补或求负求法:

已知 $[Y]_{\text{补}} = Y_{n-1}Y_{n-2} \dots Y_1Y_0$, 则对 $[Y]_{\text{补}}$ 的每一位 (包括符号位) 都按位取反, 然后再加1, 结果即 $[-Y]_{\text{补}}$ 。 $C546H \rightarrow$ 变补

0101 0110 0101 1011 B	
- 1100 0101 0100 0110 B	
<hr/>	
[1]1001 0001 0001 0101 B	

0101 0110 0101 1011 B	
+ 0011 1010 1011 1010 B	
<hr/>	
1001 0001 0001 0101 B	

注: 减法运算时, 实际二进制运算时的进位状态取反后影响AF和CF

标志位: **CF=1, AF=0, OF=1, SF=1, ZF=0, PF=0**

3.5 算术运算类指令



3. 取负指令（NEG）

格式：NEG DST; $(DST) \leftarrow 0 - (DST)$

说明：

- 是特殊的减法运算，其被减数为0，相减结果存放在DST中
- DST可以取通用寄存器和存储单元，可以直接利用NEG指令完成求负数的补码
- NEG指令将正常影响PSW中的6个状态标志位

3.5 算术运算类指令



例：设**VAR1**为字型变量，则有下列指令：

NEG BX **;(BX) \leftarrow 0 - (BX)**

NEG VAR1 **;(VAR1) \leftarrow 0 - (VAR1)**

NEG BYTE PTR 2[BX] **;(BX)+2 \leftarrow 0 - ((BX)+2)**

3.5 算术运算类指令



3.5.2 比较指令（CMP）

格式：CMP DST, SRC; (DST)－(SRC)

说明：

- **CMP**指令与减法指令类似，完成**DST**的内容减去**SRC**的内容，但其结果不保存到**DST**，指令中的**SRC**、**DST**的来源与**ADD**指令相同。
- **CMP**指令将正常影响**PSW**中的6个状态标志位：**CF**、**AF**、**OF**、**SF**、**ZF**和**PF**。

3.5 算术运算类指令



两个无符号数进行比较：

- 当**CF=0**、**ZF=0**时，**(DST) > (SRC)**；
- 当**CF=1**时，**(DST) < (SRC)**。

两个有符号数进行比较：

大小的比较要根据**OF**和**SF**共同来决定

3.5 算术运算类指令



- 当 $(OF \oplus SF) \wedge ZF = 0$ 时, $(DST) > (SRC)$;
(理解: 正常时 $OF=0$ 且 $SF=0$; 溢出时, 正数减负数超出范围得到负数结果, $OF=1$ 且 $SF=1$)

DST=70D, SRC=60D

0100 0110

- 0011 1100

0000 1010 ($OF=0, SF=0$)

DST=-60D, SRC=-70D

1100 0100

- 1011 1010

0000 1010 ($OF=0, SF=0$)

DST=70D, SRC=-50D

0100 0110

- 1100 1110

0111 1000 ($OF=0, SF=0$)

DST=70D, SRC=-60D

0100 0110

- 1100 0100

1000 0010 ($OF=1, SF=1$)

3.5 算术运算类指令



- 当 $(OF \text{ XOR } SF) = 1$ 时, $(DST) < (SRC)$;
(理解: 正常时 $OF=0$ 且 $SF=1$; 溢出时, 负数减正数超出范围得到正数结果, $OF=1$ 且 $SF=0$)
- 溢出发生在异号数比较

3.5 算术运算类指令



例：设VAR1为字型变量，VAR2为字节变量，则有

CMP CX, 2000 ;(CX)－2000，并置FLAG

CMP BX, DX ;(BX)－(DX)，并置FLAG

CMP AL, VAR2 ;(AL)－(VAR2)，并置FLAG

CMP VAR1[SI], AX ;((VAR1)+(SI))－(AX)，并置FLAG

3.5 算术运算类指令



例: 两个无符号数 (AL) = 49H、(BL) = 28H 进行大小比较。

程序片段:

```
MOV  AL, 49H
```

```
MOV  BL, 28H
```

```
CMP  AL, BL
```

执行后, CF=0, 说明 49H > 28H。

注: 如果采用 **CMP BL, AL** 语句, 则 CF=1, 说明 28H < 49H。

3.5 算术运算类指令



例：两个有符号数-104、-113进行大小比较。

程序片段：

```
MOV AL, -104
```

```
MOV BL, -113
```

```
CMP AL, BL
```

执行后的标志位：OF=0，SF=0，说明-104>-113。

注：如果采用CMP BL, AL语句，则OF=0，SF=1，这说明-113<-104。

3.5 算术运算类指令



3.5.3 增量减量指令 (INC, DEC)

格式: **INC DST;** $(DST) \leftarrow (DST) + 1$

DEC DST; $(DST) \leftarrow (DST) - 1$

说明:

- **INC**为增量指令，每次对**DST**的内容增加**1**；
- **DEC**为减量指令，每次对**DST**的内容减去**1**；
- **DST**可以取通用寄存器和存储单元；
- **INC**和**DEC**指令正常影响**PSW**中的**5**个标志位：**AF**、**OF**、**SF**、**ZF**和**PF**，但不会影响**CF**位。

3.5 算术运算类指令



例：设VAR1为字型变量，则有

DEC AX ;(AX) \leftarrow (AX) - 1

INC DL ;(DL) \leftarrow (DL) + 1

INC VAR1 ;(VAR1) \leftarrow (VAR1) + 1

DEC WORD PTR [BX] ;((BX)) \leftarrow ((BX)) - 1

3.5 算术运算类指令



3.5.4 乘法和除法指令

1. 乘法运算

- 两个无符号二进制数的乘法运算与十进制数的乘法类似，可以采用列竖式的方法计算，只是在相加运算时，按逢二进一的规则。
- 两个有符号二进制数进行乘法运算时，先将负数采用变补的方法变换成正数，进行两个正数相乘，然后统一考虑符号。

3.5 算术运算类指令



2. 乘法指令（**MUL**，**IMUL**）

格式：**MUL SRC**； 无符号数相乘

IMUL SRC； 有符号数相乘

说明：指令的目的操作数隐含在**AX**（或**AL**）中，**SRC**可以取通用寄存器或存储单元，只能是字节和字，它们决定了乘法操作的类型。

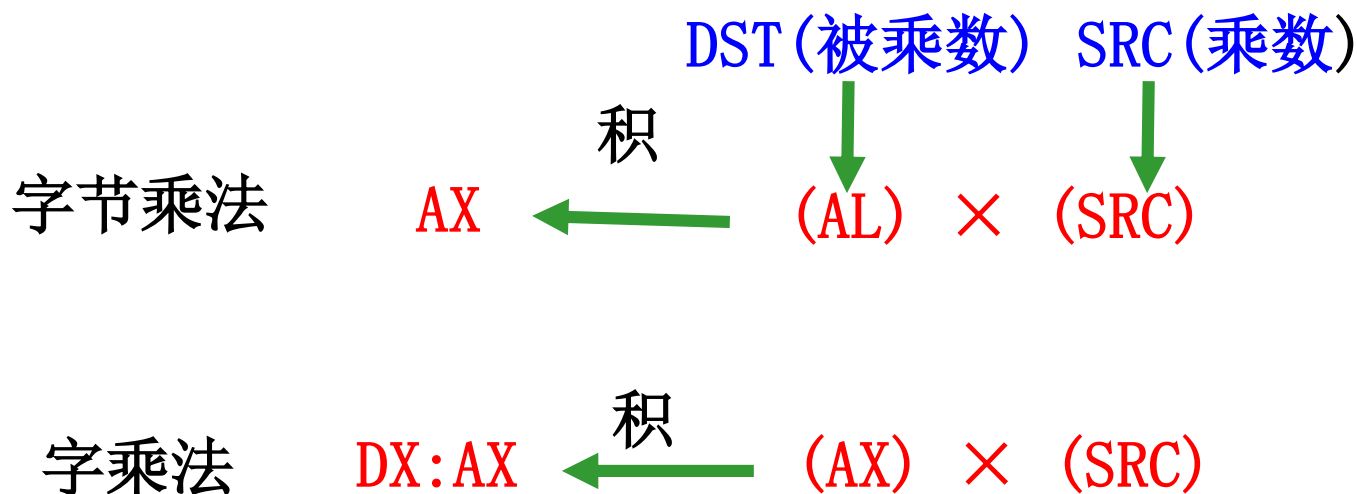
3.5 算术运算类指令



SRC为寄存器或存储单元，只能是字节和字，它们决定了乘法操作的类型。

乘法操作的类型：

- 当**SRC**为字节时，**MUL**和**IMUL**为字节运算，表示将**AL**中的数与（**SRC**）相乘，其结果放入（**AX**）中；
- 当**SRC**为字时，**MUL**和**IMUL**为字运算，表示将**AX**中的数与（**SRC**）相乘，其结果的高**16**位保存在**(DX)**，低**16**位保存在（**AX**）



3.5 算术运算类指令



MUL和IMUL指令只对PSW的CF、OF位有影响

影响规则：

- 当采用**MUL**运算时，字运算结果的**DX**为**0**时，**CF=0**，**OF=0**，表示两个字相乘其结果也为一个字；字节运算结果的**AH**为**0**时，**CF=0**，**OF=0**，表示两个字节相乘其结果也为一个字节；否则**CF=1**，**OF=1**。
- 当采用**IMUL**运算时，字运算结果的**DX**为符号扩展时，**CF=0**，**OF=0**，表示两个字相乘其结果可以用一个字表示；字节运算结果的**AH**为符号扩展时，**CF=0**，**OF=0**，表示两个字节相乘其结果也可以用一个字节表示；否则**CF=1**，**OF=1**。

3.5 算术运算类指令



例.下列指令是合法的。

MUL CX ;(DX:AX) ← (AX)*(CX), 为字操作

IMUL CL ;(AX) ← (AL)*(CL), 为字节操作

下列指令是非法的。

MUL AL,BL ;✗ DST(被乘数)应为隐含寻址

IMUL 05H ;✗ SRC(乘数)不能为立即数寻址

3.5 算术运算类指令



例. 乘法的字节运算。要计算两个无符号数**2CH**、**42H**的乘积，结果保存在**AX**中。

程序片段：

```
MOV AL, 2CH
```

```
MOV BL, 42H
```

```
MUL BL
```

结果：（**AX**）=**0B58H**，**CF**=1，**OF**=1。

3.5 算术运算类指令



例. 乘法的字运算。要计算两个有符号数1000、-12345的乘积。

程序片段:

```
MOV AX, 1000
```

```
MOV BX, -12345
```

```
IMUL BX
```

结果: (DX) = FF43H, (AX) = A158H;

CF=1, OF=1

3.5 算术运算类指令



例. 字节×字运算。要计算两个有符号数**15H**、**FB78H**的乘积。

有符号数**FB78H**为负数(**-1160**)。

程序片段：

```
MOV AL, 15H
```

```
CBW
```

```
MOV BX, 0FB78H
```

```
IMUL BX
```

结果 **(DX) = FFFFH**, **(AX) = A0D8H**;

CF=0, **OF=0**, 表示相乘结果只需要用一个字表示。

3.5 算术运算类指令



3. 除法指令（DIV，IDIV）

格式： **DIV SRC;** 无符号数的除法

IDIV SRC; 有符号数的除法

说明：

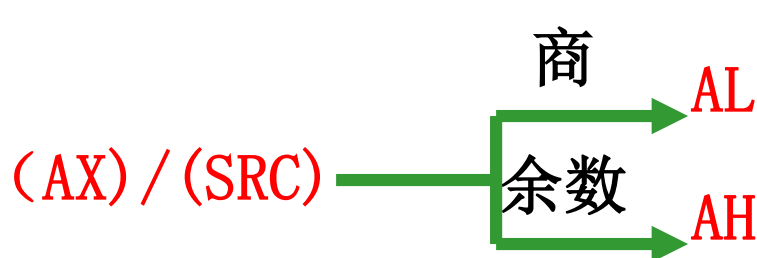
- 指令的目的操作数隐含在**DX: AX**（或**AX**）中，**SRC**可以取通用寄存器和存储单元，它只能是字节和字，这也决定了除法操作的类型。
- **DIV**和**IDIV**指令不影响**PSW**中的标志位
- 除法不允许出现除数为**0**或商溢出，若发生除数为**0**或商溢出则其结果没有意义，并引起中断。

3.5 算术运算类指令



(1) 除法操作类型：

- 当**SRC**为字节时，**DIV**和**IDIV**为字节运算，表示将**AX**中的**16**位二进制数除以**8**位二进制数（**SRC**），其结果的商保存在（**AL**）中，余数保存在（**AH**）中；
- 当**SRC**为字时，**DIV**和**IDIV**为字运算，表示将**DX**与**AX**联合构成的**32**位二进制数除以**16**位二进制数（**SRC**），其结果的商保存在（**AX**）中，余数保存在（**DX**）中。



字节除法



字除法

3.5 算术运算类指令



例. 字÷字节的除法运算。

设要完成除法运算**12345÷156**。

思路：被除数**12345**可以用一个字表示，除数**156**可以用一个字节表示，可以直接采用除法的字节运算。

程序片段：

```
MOV AX, 12345
```

```
MOV BL, 156
```

```
DIV BL
```

结果：商（**AL**）=**4FH**，余数（**AH**）=**15H**

3.5 算术运算类指令



例. 双字÷字的除法运算

完成除法运算**28901240H**÷（**-6528H**）。

思路：是有符号数的除法运算，被除数应该用一个双字表示，
除数可以用一个字表示，

程序片段：

MOV AX, 1240H

MOV DX, 2890H

MOV BX, -6528H

IDIV BX

结果：商（**AX**）=**9959H**，余数（**DX**）=**2528H**。

3.5 算术运算类指令



例. 下列指令是合法的。

DIV BL

IDIV CX

DIV WORD PTR [BX][SI]

下列指令是非法的。

DIV 12 ;✗ SRC(除数)不能为立即数寻址

DIV [SI]+02H ;✗ SRC类型不明确

IDIV AX, BL ;✗ DST(被除数)应为隐含寻址

3.5 算术运算类指令



3.5.5 符号扩展指令（CBW, CWD）

格式：

CBW ;将**AL**中的符号扩展到**AH**中,形成一个字**AX**

CWD ;将**AX**中的符号扩展到**DX**中,形成双字（**DX:AX**）

说明：**CBW**为字节到字的符号扩展指令，表示将**AL**中的有符号数扩展到**AH**，即根据**AL**中的符号位**D7**来扩展：

- 当**D7=0**时，**AH=00H**；
- 当**D7=1**时，**AH=FFH**

这样可以保证有符号数**AL**与有符号数**AX**所表示的值相同。

3.5 算术运算类指令



CWD (Convert word to doubleword) 为字到双字的符号扩展指令，表示将**AX**中的有符号数扩展到**DX**，即根据**AX**中的符号位**D15**来扩展：

- 当**D15=0**时，**DX=0000H**
- 当**D15=1**时，**DX=FFFFH**

可以保证有符号数**AX**与有符号数**DX:AX**所表示的值相同。

3.5 算术运算类指令



例. 写出计算 $Y=a*b+c-18$ 的程序。

在数据段定义变量如下：

DAT1 DB 34H; 为a的一个设定值

DAT2 DB 56H; 为b的一个设定值

DAT3 DB 0E7H ; 为c的一个设定值

DATY DW ? ; DATY单元存放结果

代码段的程序片段为：

MOV AL, DAT1 ;取a

MOV BL, DAT2 ;取b

IMUL BL ; (AL)*(BL) \rightarrow (AX)

MOV BX, AX ; (AX) \rightarrow (BX)

MOV AL, DAT3 ;取c

CBW ;扩展AL \rightarrow AX

ADD AX, BX ; (AX)+(BX) \rightarrow (AX)

SUB AX, 18

MOV DATY, AX

3.5 算术运算类指令



3.5.6 BCD数运算调整指令

- **BCD码（数）**是用四位二进制码来表示一位十进制数。利用**BCD码**可进行加、减、乘、除运算，但为了得到正确的结果，必须进行修正。
 - **BCD码表示**可分为两类：
 - **分离BCD码**：8位的寄存器中只包含一位**BCD码**（**D0~D3**）
 - **组合BCD码**：8位的寄存器中包含了两位**BCD码**。
- 两类修正的方法是不同的，其调整指令也不同。

3.5 算术运算类指令



- **BCD数调整指令共6条**

助记符格式	功能说明
AAA	加法分离BCD调整
AAS	减法分离BCD调整
DAA	加法组合BCD调整
DAS	减法组合BCD调整
AAM	乘法分离BCD调整
AAD	除法分离BCD调整

3.5 算术运算类指令



1. 加法调整指令（**AAA**, **DAA**）

格式: **AAA** ; 加法分离**BCD**码调整

DAA ; 加法组合**BCD**码调整

说明:

□ **AAA**为分离**BCD**码加法运算后的调整指令，表示对相加结果**AL**的低4位进行加6修正。根据运算结果及修正结果的**AF**有无进位，进行下列操作：

- **AF**无进位，则**AH**=0，**CF**=0，**AF**=0；
- **AF**有进位，则**AH**=1，**CF**=1，**AF**=1；
- 并清掉**AL**中的高4位。

3.5 算术运算类指令



□ **DAA**为组合**BCD**码加法运算后的调整指令，表示对相加结果**AL**的低**4**位和高**4**位分别进行加**6**修正。**DAA**指令对**PSW**中的**AF**、**CF**、**SF**、**ZF**、**PF**都有影响，其效果等同于**ADD**指令。

3.5 算术运算类指令



例. 计算十进制数的加法运算，设要计算**4+8**。

思路：当通过键盘输入这两个十进制数时，我们得到的是其**ASCII**码，如果将其看作为分离**BCD**码，则高**4**位为无效部分，因此不需要将高**4**位清除。

程序片段：

```
MOV AL, '4'
```

```
MOV BL, '8'
```

```
ADD AL, BL
```

```
AAA
```

AAA指令调整之前，**(AL) = 6CH**，执行**AAA**指令后，其结果为**(AX) = 0102H**，**CF=1**，**AF=1**；

这说明“**4+8=12**”。

3.5 算术运算类指令



例. 计算十进制数的加法运算，设要计算**34+28**。

程序片段：

MOV AL, 34H

MOV BL, 28H

ADD AL, BL

DAA

DAA指令调整之前，**(AL) = 5CH**，执行**DAA**指令后，

结果：**(AL) = 62H**，**CF=0**，**AF=1**，**SF=0**，**PF=0**，**ZF=0**

说明 “**34+28=62**”。

3.5 算术运算类指令



2. 减法调整指令（**AAS**，**DAS**）

格式：**AAS**； 减法分离**BCD**码调整

DAS； 减法组合**BCD**码调整

说明：

□ **AAS**为分离**BCD**码减法运算后的调整指令，表示对相减结果**AL**的低4位进行减6修正。根据运算结果及修正结果的**AF**有无借位，进行下列操作：

- **AF**有借位，则**CF**=1，**AF**=1；
- **AF**无借位，则**CF**=0，**AF**=0；
- 并清掉**AL**中的高4位。

3.5 算术运算类指令



□ **DAS**为组合**BCD**码减法运算后的调整指令，表示对相减结果**AL**的低4位和高4位分别进行减6修正。**DAS**指令对**PSW**中的**AF**、**CF**、**SF**、**ZF**、**PF**都有影响，其效果等同于**SUB**指令。

3.5 算术运算类指令



例. 计算十进制数的减法运算，设要计算**5－9**。

程序片段：

MOV AL, 05H

MOV BL, 09H

SUB AL, BL

AAS

AAS指令调整之前，**(AL) = 0FCH**，**AF = 1**，执行**AAS**指令后，其结果为**(AL) = 06H**，**CF = 1**，**AF = 1**；

这说明调整结果为负数对**10**补码，为求取原码必须对其求一次**10**补码（**[[X]补]补 = [X]原**），因此结果为 **6H - 10H = - 4H**（负数的补码 = 模 + 负数）

3.5 算术运算类指令



3. 乘法分离BCD码调整

格式： **AAM**； 乘法分离BCD码调整

说明： **AAM**为分离BCD码乘法的调整指令，两个分离BCD码相乘之后，需用**AAM**调整，调整方法是将结果（**AL**）除以10（**0AH**），其商为（**AH**）的低4位，余数为（**AL**）的低4位。

AAM指令会影响PSW的**SF**、**ZF**：

- **SF=0**，因为**AL**的高4位为0；
- 当余数为0，即**AL=0**时，**ZF=1**；
- 当余数不为0时，**ZF=0**。

3.5 算术运算类指令



例. 计算十进制数的乘法运算，设要计算 7×8 。

程序片段：

MOV AL, 7

MOV BL, 8

MUL BL

AAM

AAM指令执行之前，**(AX) = 38H**，**AAM**指令执行后，
(AX) = 0506H，**SF=0**，**ZF=0**

3.5 算术运算类指令



4. 除法分离BCD码调整

格式：**AAD**； 除法分离BCD码调整

说明：**AAD**为分离BCD码除法的调整指令，两个分离BCD码除法指令之前应采用**AAD**进行调整。被除数**AH**和**AL**中分别保存有一位分离BCD码，**AAD**指令进行除法调整：
 $(AL) \leftarrow (AH) \times 10 + (AL)$ ，并使**(AH)←0**，然后进行正常的除法运算。

➤ **AAD**指令会影响**PSW**的**PF**、**SF**、**ZF**标志位。

3.5 算术运算类指令



例. 计算十进制数的除法运算，设要计算 $27 \div 4$ 。

程序片段：

```
MOV AX, 0207H
```

```
MOV BL, 4
```

```
AAD      ; (AX) ← 001BH
```

```
DIV BL
```

结果：AL=06H，AH=03H，这说明十进制数的除法运算 $27 \div 4$ 的商为6，余数为3。