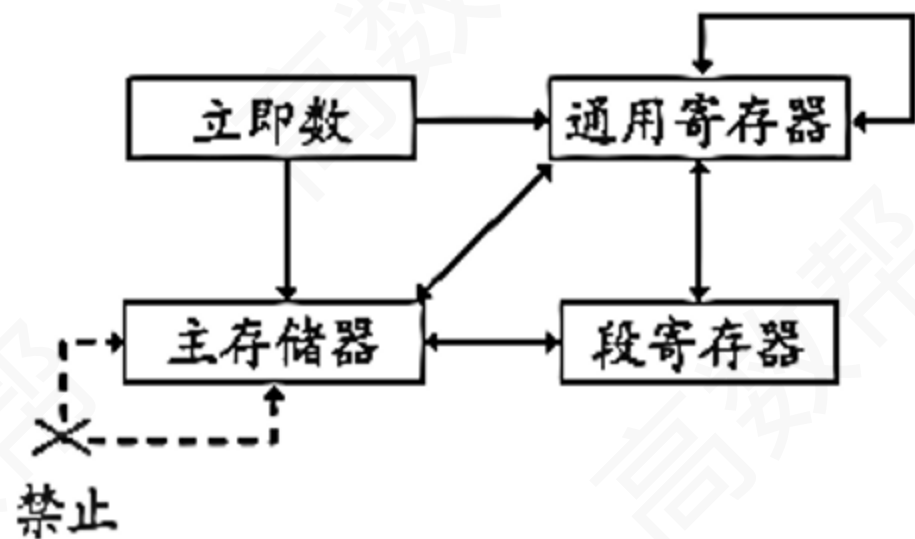


| 考点 | 重要程度 | 占分 | 题型 |
|-----------|-------|------|-------|
| 1.数据传送类指令 | ★★★★★ | 5~10 | 选择 填空 |
| 2.算术运算类指令 | 必考 | | 大题 |

考点1 数据传送类指令

一、mov指令

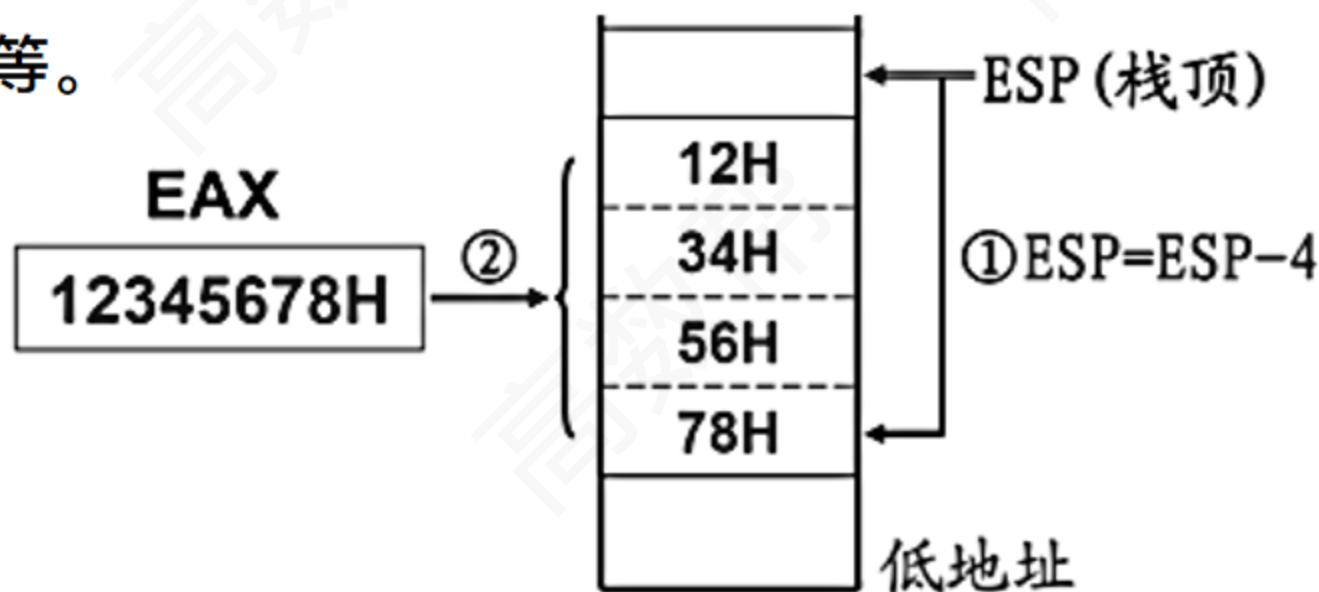


扫码观看
视频讲解更清晰

二、XCHG指令

XCHG指令不允许的情况有以下四种：

- 1.不能同时都为内存操作数；
- 2.任何一个操作数都不能为段寄存器；
- 3.任何一个操作数不能为立即数；
- 4.两个操作数的长度不能不相等。



三、PUSH指令和POP指令

堆栈一个特殊的存储区域

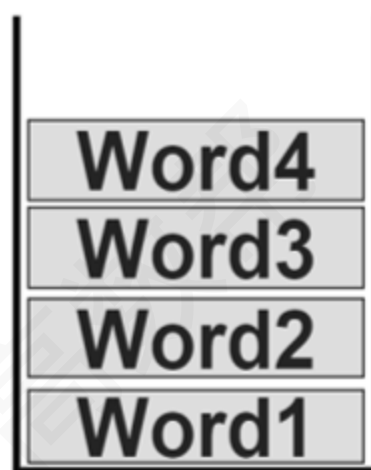
存取规则：先进后出FILO (First In Last out)

也可称为：先进后出LIFO (Last In First Out)

具有两种基本的数据传输操作

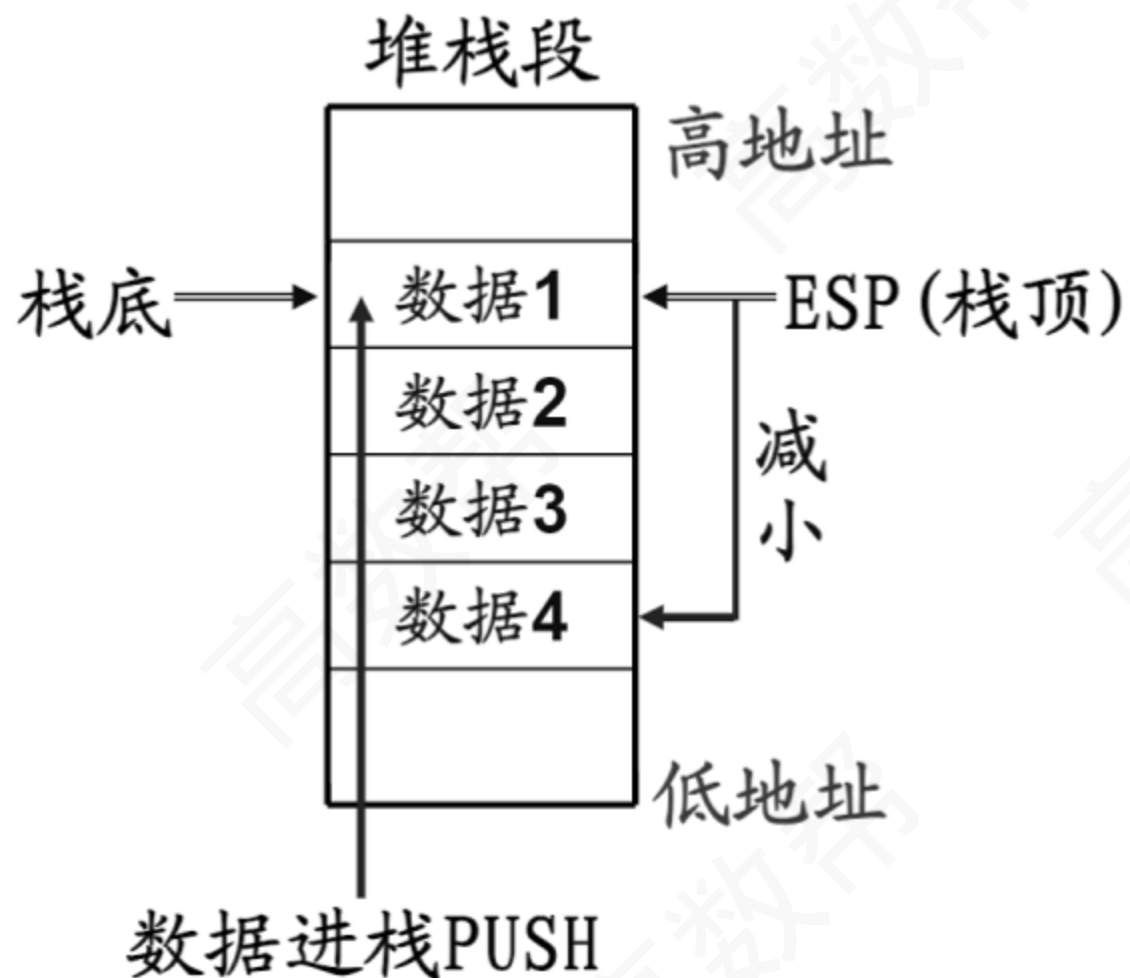
1.数据压进堆栈 PUSH

2.数据弹出堆栈 POP

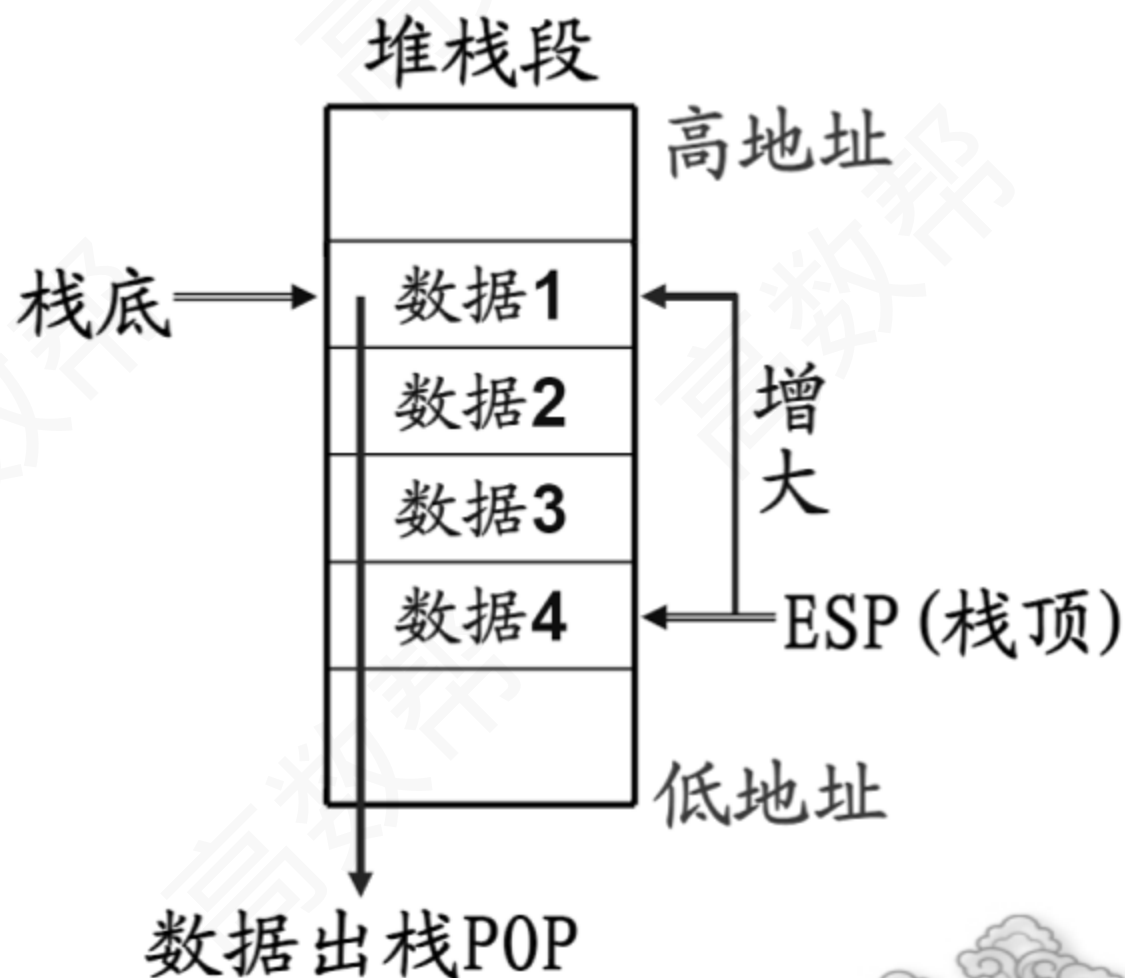


Stack

数据压入堆栈、ESP逐渐减小



数据弹出堆栈、ESP逐渐增大



1.PUSH指令

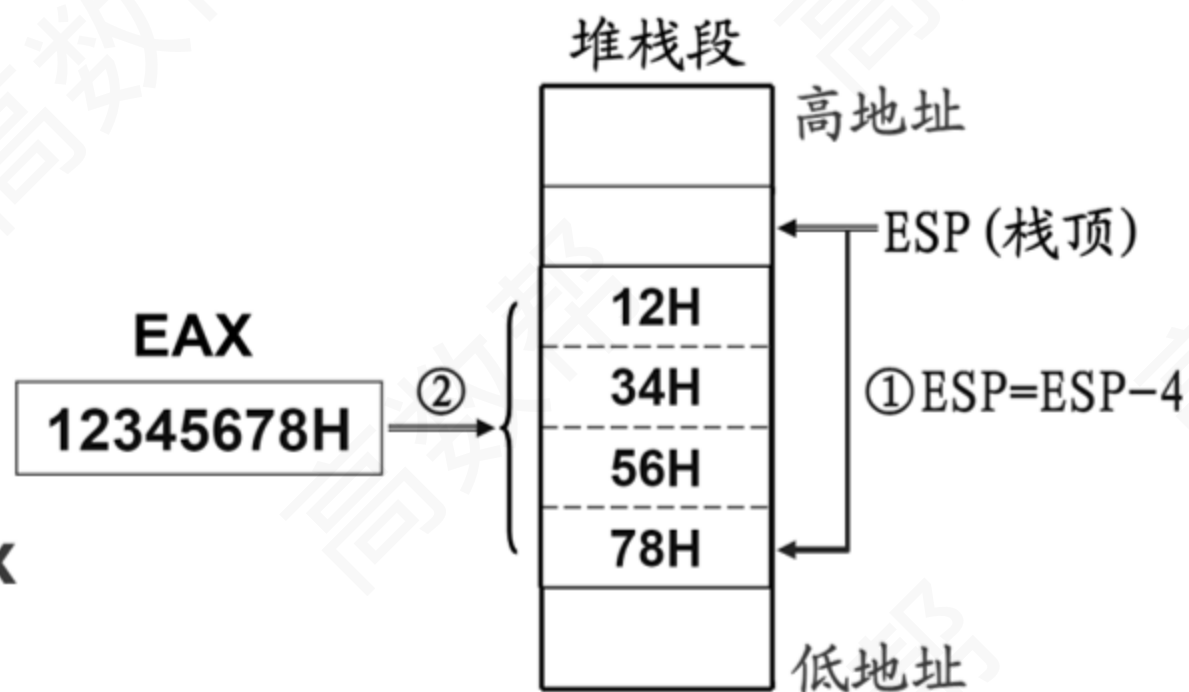
;进栈指令

push eax

;等同于如下2条指令

① **sub esp,4**

② **mov [esp],eax**



2.POP指令

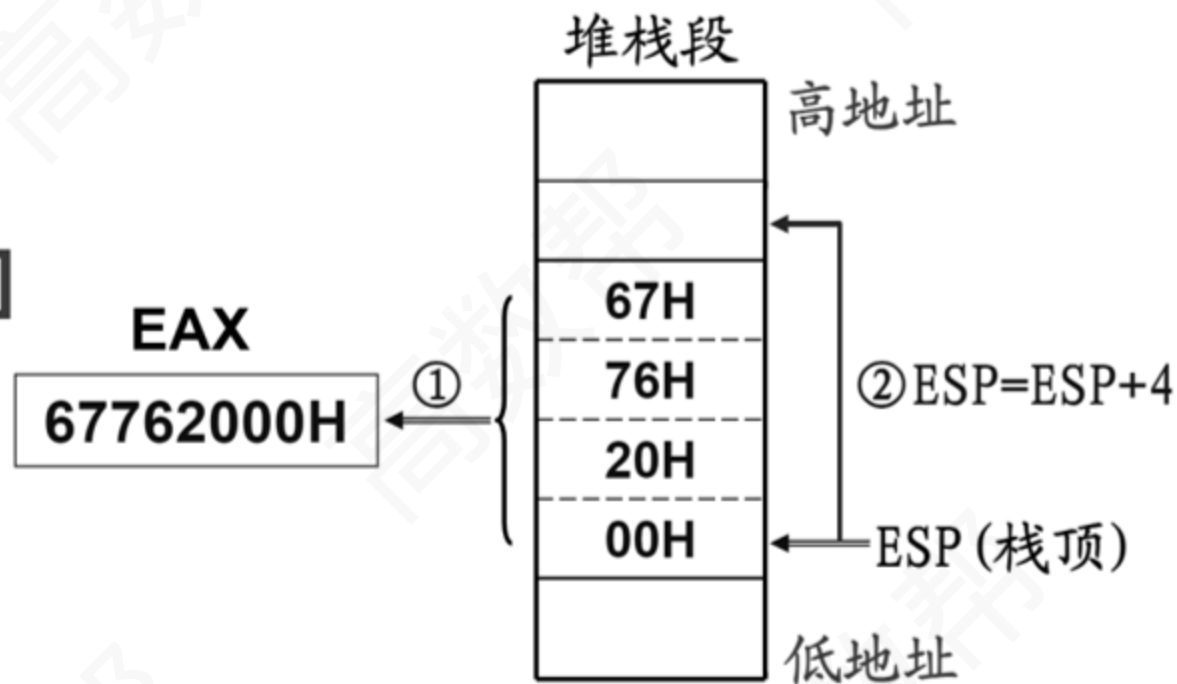
;进栈指令

pop eax

;等同于如下2条指令

① **mov eax,[esp]**

② **add esp,4**



扫码观看
视频讲解更清晰

【题1】 设SP初值为2000H，执行指令“PUSH AX”后，SP的值是(D)

A.1FFFH; B.1998H; C.2002H; D.1FFE H。

【题2】 若栈顶的物理地址为 20100H，当执行完指令 PUSH AX 后，栈顶的物理地址为 (D)

A. 20098H B. 20102H C. 20100H D. 200FEH

【题3】 下列关于堆栈的说法，错误的是(D)

A.以“先入后出”为原则。

B.栈区最高地址单元的前一个单元为栈底。

C.运行中 SP 寄存器动态跟踪栈顶位置。

D.压栈和弹出都是以字节为单位

【题4】 A DW 1234H
B DW 5678H
:
PUSH A
PUSH B
POP A
POP B

试回答:

1. 上述程序段执行后(A) = 5678H ,
2. 设执行前 SP = 200H, 执行后 SP = 2000H 。



扫码观看
视频讲解更清晰

三、LEA

LEA: 获取存储器操作数的地址

LEA指令在指令执行时计算出偏移地址

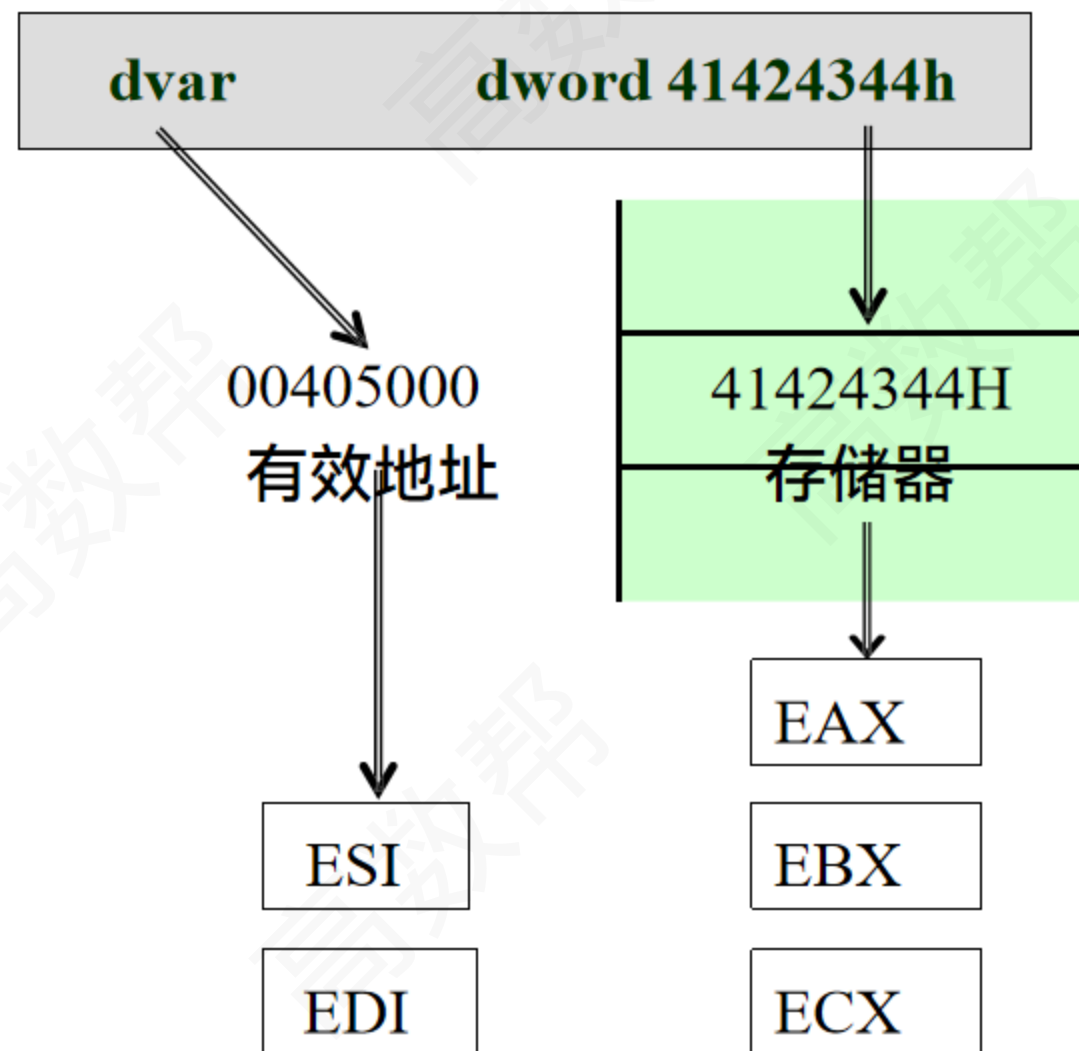
OFFSET操作符在汇编阶段取得变量的偏移地址，指令执行速度更快

LEA指令能获取汇编阶段无法确定的偏移地址



三、LEA

```
mov eax,dvar lea, esi,dvar mov ebx,[esi]  
mov edi,offset dvar mov ecx,[edi]
```



考点2 算术运算类指令

一、标志位

状态标志是处理器最基本的标志。

一方面，作为加减运算和逻辑运算的辅助结果

另一方面构成各种条件实现程序分支

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

8086的标志

1.进位标准CF: 看做无符号数

2.溢出标志OF: 看做有符号数

两个相同符号数相加 (含两个不同符号数相减)

正+正 负+负 正-负 (中间看成加号, 左右符号同)

负-正 (中间看成加号, 左右符号同)

运算结果与原数据符号相反时, 产生溢出; 其他条件不会产生溢出

3.零标志ZF: 运算结果为0, 是1

4.符号标志SF: 结果最高位为1, SF=1

5.奇偶标志PF: 运算结果最低字节“1”的个数为偶数时, PF=1

【题5】当标志位CF=1 时表示无符号数运算产生溢出, 而当标志位 OF=1 是表示带符号数运算产生溢出。

二、加法运算指令

1. 加法指令 ADD

add eax,88000000h

;32位加法

;最高位是D₃₁

| | |
|---------|------------|
| 加法之前EAX | AAFFB36EH |
| 32位加法 | +88000000H |
| 加法之后EAX | 32FFB36EH |

0011 0010 1111 1111 1011 0011 0110 1110

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

【题6】 下列程序执行后, (AX)=? , (BX)=?

MOV AX, 1234H

MOV BX, 5678H

ADD AX, BX

解: 执行: $AX \leftarrow 1234H$

执行: $BX \leftarrow 5678H$

执行: $AX \leftarrow (AX) + (BX)$

执行后: (AX)=68ACH, (BX)=5678H



扫码观看
视频讲解更清晰

2.增量指令INC

只有一个操作数，寄存器或存储单元

对操作数加1(增量)，再将结果返回原处

i++

INC reg/mem ; 加1: $\text{reg/mem} \leftarrow \text{reg/mem} + 1$

用于计数器和地址指针的调整

不影响进位CF标志，影响其他状态标志位

```
inc ecx
```

```
inc dword ptr [ebx]
```

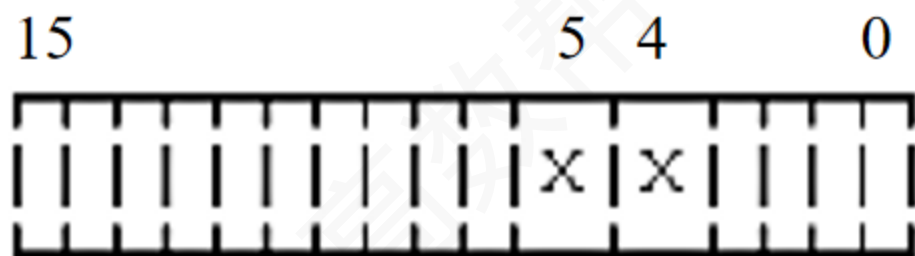
```
inc wvar
```

```
inc bl
```

```
inc word ptr [esi]
```

```
inc wvar[edi]
```


【题7】 请用一条指令实现如下操作： 将AX中划‘X’的位清0



解：指令：AND AX,0FFCFH。



扫码观看
视频讲解更清晰

三、减法运算指令

1. 减法指令SUB

sub eax,0bb000000h

;32位减法

;最高位是D₃₁

| | |
|---------|------------------|
| 减法之前EAX | AAFF3322H |
| 32位减法 | -BB000000H |
| 减法之后EAX | EFFF3322H |

1110 1111 1111 1111 0011 0011 0010 0010

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

2.减量指令DEC

只有一个操作数，寄存器或存储单元

对操作数减1 (减量)，再将结果返回原处

DEC reg/mem ; 减1: $\text{reg/mem} \leftarrow \text{reg/mem} - 1$

用于计数器和地址指针的调整

不影响进位CF标志，影响其他状态标志位

dec cx

dec byte ptr [ebx]

dec wvar

dec bl

dec word ptr [esi]

dec wvar[edi]

i--

四、求补指令NEG

对操作数执行求补运算即用零减去操作数

NEG reg/mem ; $\text{reg/mem} \leftarrow \text{reg/mem}$

neg al

neg byte ptr [ebx]

neg wvar[esi]

neg ax

neg word ptr [ebx]

neg wvar[edi]

五、乘法运算指令

1. 无符号数乘法指令MUL

2. 带符号乘法指令IMUL

【题8】 设AH=04H, CL=0F8H, 执行MUL CL指令后, 结果为(C)

A. AX=0032H B. AX=00E0H C. AX=03E0H D. AX=0FFE0H

六、除法运算指令

无符号数乘法指令DIV

带符号除指令IDIV



扫码观看
视频讲解更清晰

七、字节/字扩展指令CBW/CWD

这两条指令主要用于除法指令前，形成双倍长度的被除数，他们都是无操作数指令，隐含使用AX或DX

指令功能： CBW：扩展AL中符号味道AH中

CWD：扩展AX中符号味道DS中

两条指令对标志都无影响

如：

MOV AL, 3EH ;AL=0011 1110B

CBW ;AX=0000 0000 0011 1110B

MOV AL, 93H

CBW ;AX=1111 1111 1001 0011B

ADD

INC 加1：不影响CF位

SUB

DEC 减1：不影响CF位

NEG 求补，0减去操作数

MUL无符号数乘法指令：只影响CF和OF，其他不确定

字节运算： $AX \leftarrow (AL) \times (OPRD)$

字运算： $DX:AX \leftarrow (AX) \times (OPRD)$

IMUL带符号数乘法指令

DIV无符号数除法指令：对标志各位无有效影响

IDIV带符号数除法指令

CBW/CWD字节/字扩展指令：对标志位无影响

【题9】 MUL为(无符号乘法指令)，IMUL为(有符号乘法指令)，AND为(逻辑与操作)。