



第三部分 微机原理与接口

第12章 8086系列指令系统





12.1 概述

能够被计算机识别并执行的二进制代码称为指令。

所有指令的集合称为指令系统。

按功能分类

数据传送类
算术运算类
逻辑运算和移位
串操作
控制转移类
处理器控制



12.1.1 8086指令格式

1 指令的一般格式

操作码	运算数据或结果的地址，参加运算的数据或数据的地址
-----	--------------------------

8086指令的长度在1~7字节之间，操作码为1~2字节，指令的长度决定于操作数的个数以及所采用的寻址方式。

按操作数数量分类 {

- 零操作数指令
- 单操作数指令
- 双操作数指令



2 操作数类型

- 立即数操作数
- 寄存器操作数
- 存储器操作数

◆ 立即数是运算数据本身，无地址含义，故无法作为目标操作数。

◆ 存储器操作数的表现形式：[内存偏移地址]。



12.2 8086寻址方式

寻址方式，是指获得操作数所在地址的方法。

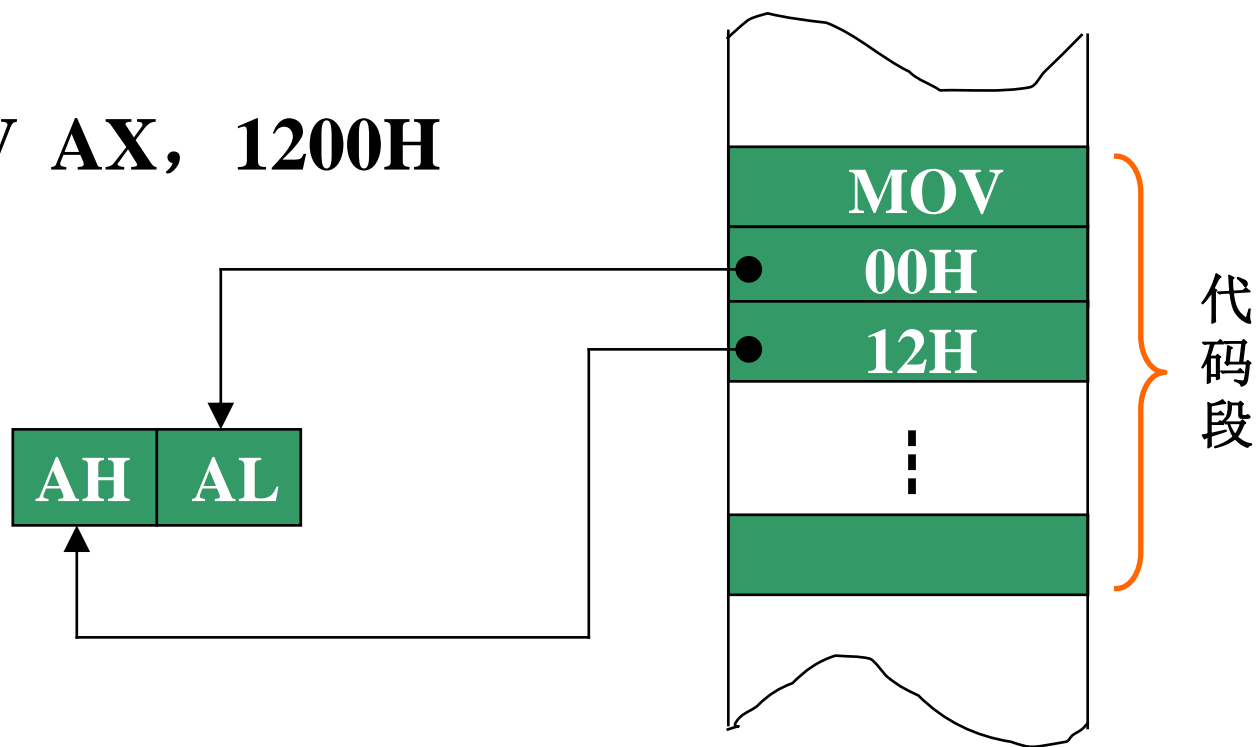
寻址 { 寻找操作数的地址
寻找下一条指令的地址，即程序的地址



1 立即寻址

仅适用于源操作数。

例：MOV AX, 1200H





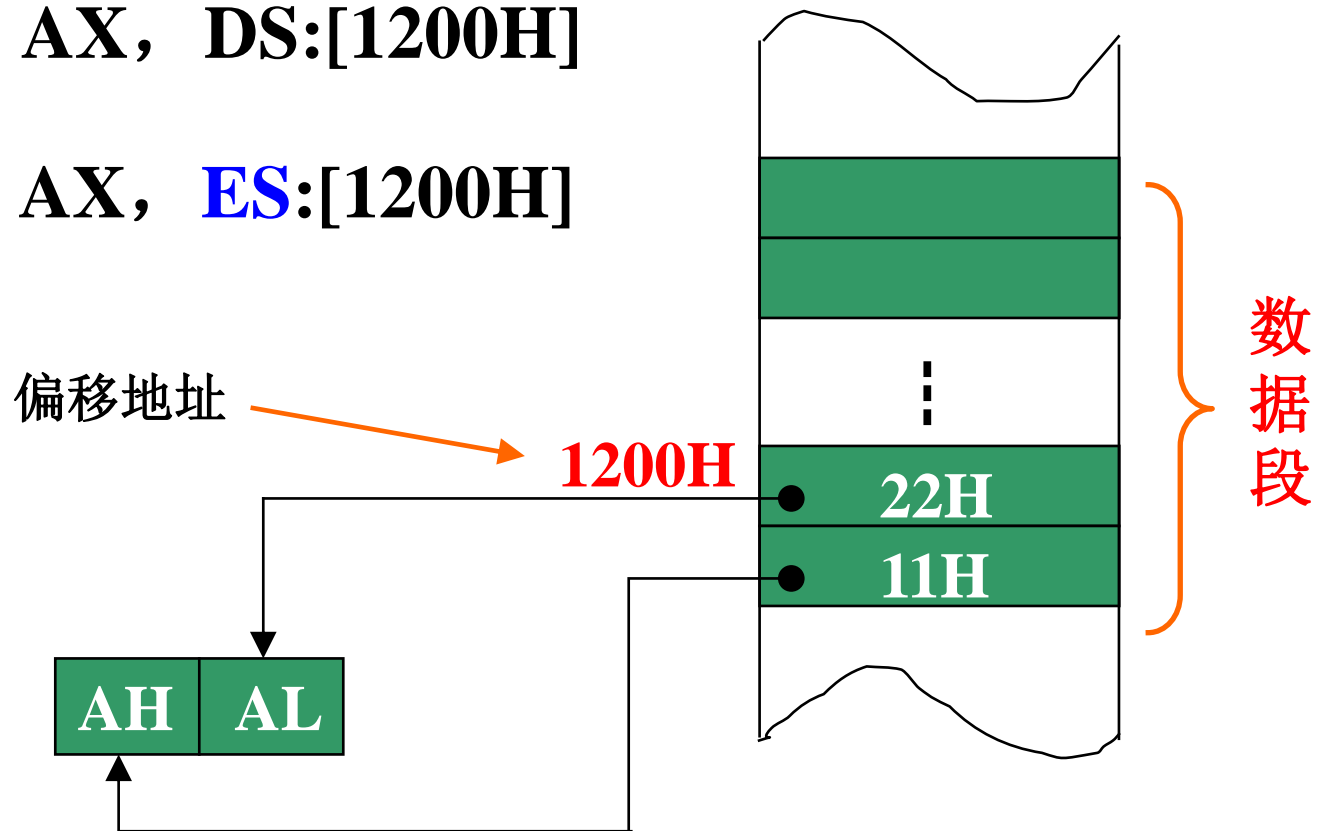
2

直接寻址

指令中直接给出操作数的偏移地址，默认在数据段，允许段重设。即可以增加段前缀实现段超越。

例：MOV AX, DS:[1200H]

MOV AX, ES:[1200H]





3

寄存器寻址

参加操作的操作数在CPU的寄存器中。

例：MOV AX, BX





4 寄存器间接寻址

寄存器的内容表示操作数的偏移地址。

允许的寄存器为**BX, BP, SI, DI**，又称为间址寄存器或地址指针。

操作数的段地址取决于间址寄存器

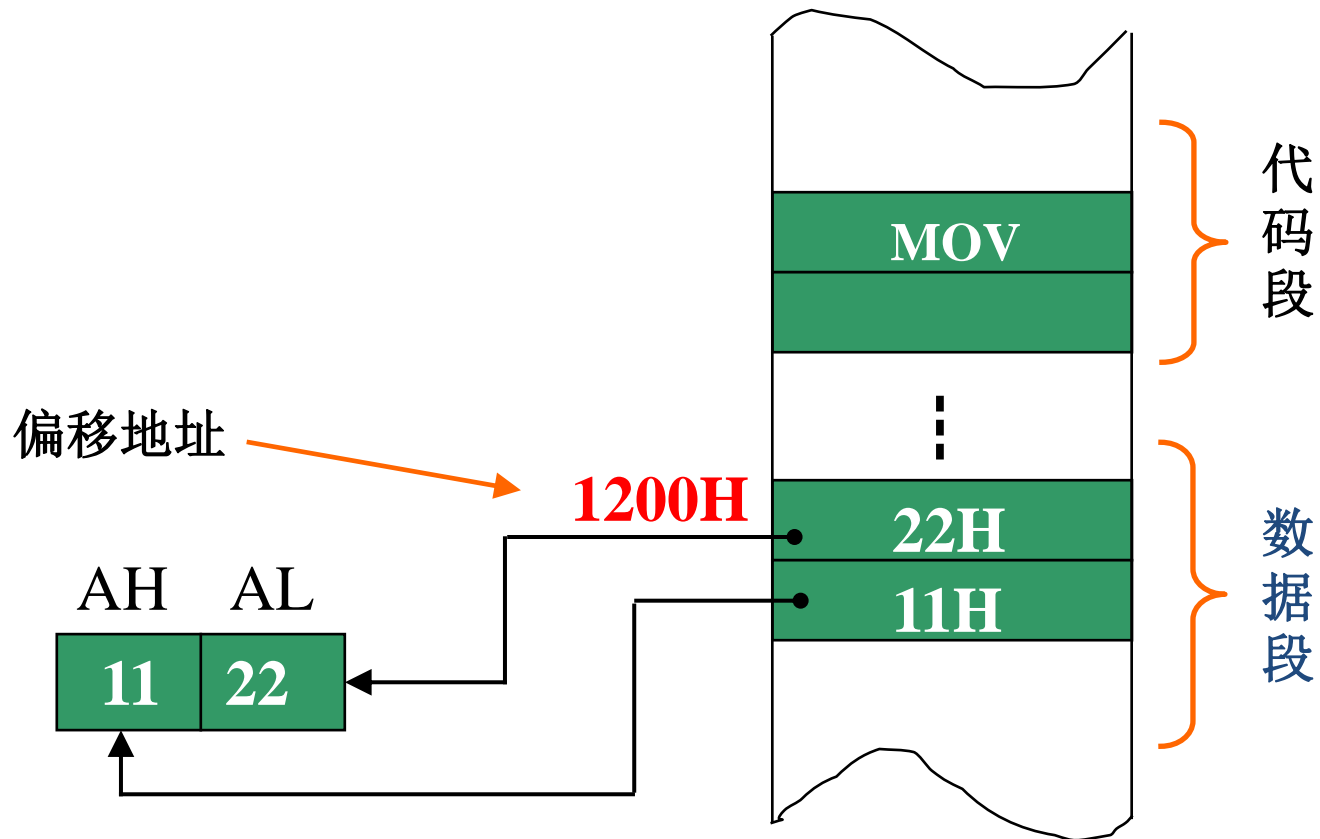
BX, SI, DI: 默认为数据段**DS**

BP: 默认为堆栈段**SS**

因为寄存器的内容表示操作数的偏移地址，所以指令中的间址寄存器必须加上方括号。



例, **MOV BX, 1200H**
MOV AX, [BX]





5 寄存器相对寻址

操作数在内存中的偏移地址由间址寄存器的内容加上指令中给出的一个8位或16位的位移量组成。

操作数所在的段由所使用的间址寄存器决定。

例，指令 **MOV AX, Disp[BX]**

假设，**DS=6000H, BX=1000H, Disp=08H**

则偏移地址为， **BX+Disp=1008H**

操作数物理地址为**61008H**



可以使用的寄存器为：**BX、BP、SI和DI**



格式可以为：

MOV AX, [BX+Disp]

或：**MOV AX, Disp[BX]**

或：**MOV AX, [BX]+Disp**

或：**MOV AX, Disp+[BX]**



6 基址变址寻址

◆ 操作数的偏移地址为，一个基址寄存器的内容 + 一个变址寄存器的内容；

◆ 操作数的段地址由选择的**基址寄存器决定**，**允许重新设定**。

基址寄存器为BX，默认在数据段

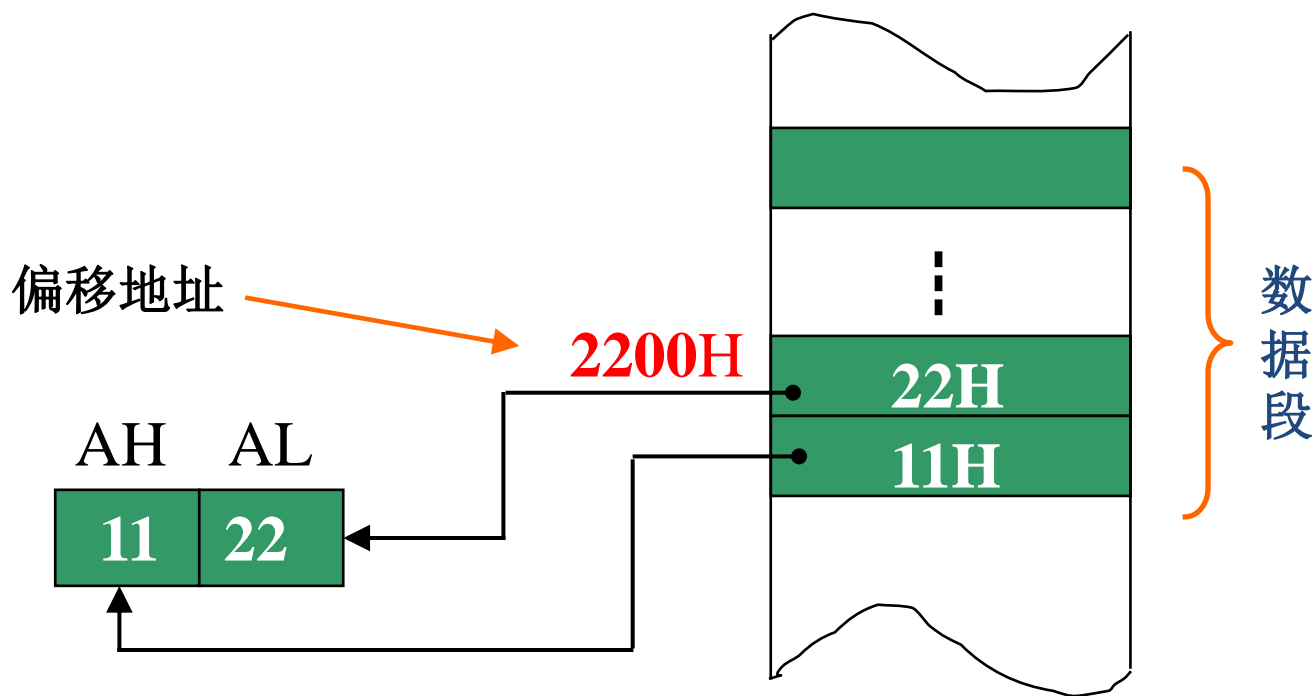
基址寄存器为BP，默认在堆栈段

例，**MOV AX, [BX][SI]**

MOV AX, [BP][SI]



例, **MOV SI, 1100H**
MOV BX, SI
MOV AX, [SI+BX]





7 基址-变址-相对寻址

- ❖ 操作数的偏移地址为，基址寄存器的内容 + 变址寄存器的内容+位移量；
- ❖ 操作数的段地址由选择的基址寄存器决定，允许重新设定。
- ❖ 指令中不能同时出现两个基址寄存器或两个变址寄存器。



8

隐含寻址

有些指令的操作码不仅包含了操作的性质，同时还隐含规定了部分操作数。

例：

MUL BL

指令执行：

$AL \times BL \rightarrow AX$

这条指令隐含规定了被乘数在**AL**中，同时乘积的结果存放于**AX**中。



12.3 8086指令系统

后续所用的符号说明：

OPRD	泛指各种类型的操作数
Mem	存储器操作数
Acc	累加器操作数
Dest	目标操作数
Src	源操作数
Disp	8位或16位位移量，可用符号地址表示
Data	8位或16位立即数
Port	输入输出端口，可用数字或表达式表示
[]	表示存储器操作数



12.3.1 数据传送指令

按
功
能
分
类

通用数据传送指令
输入输出指令
目标地址传送指令
标志传送指令



1

通用数据传送指令

一般传送指令**MOV**

堆栈操作指令**PUSH**和**POP**

交换指令**XCHG**

查表转换指令**XLAT**

字位扩展指令**CBW**和**CWD**

通用数据传送指令不影响标志位

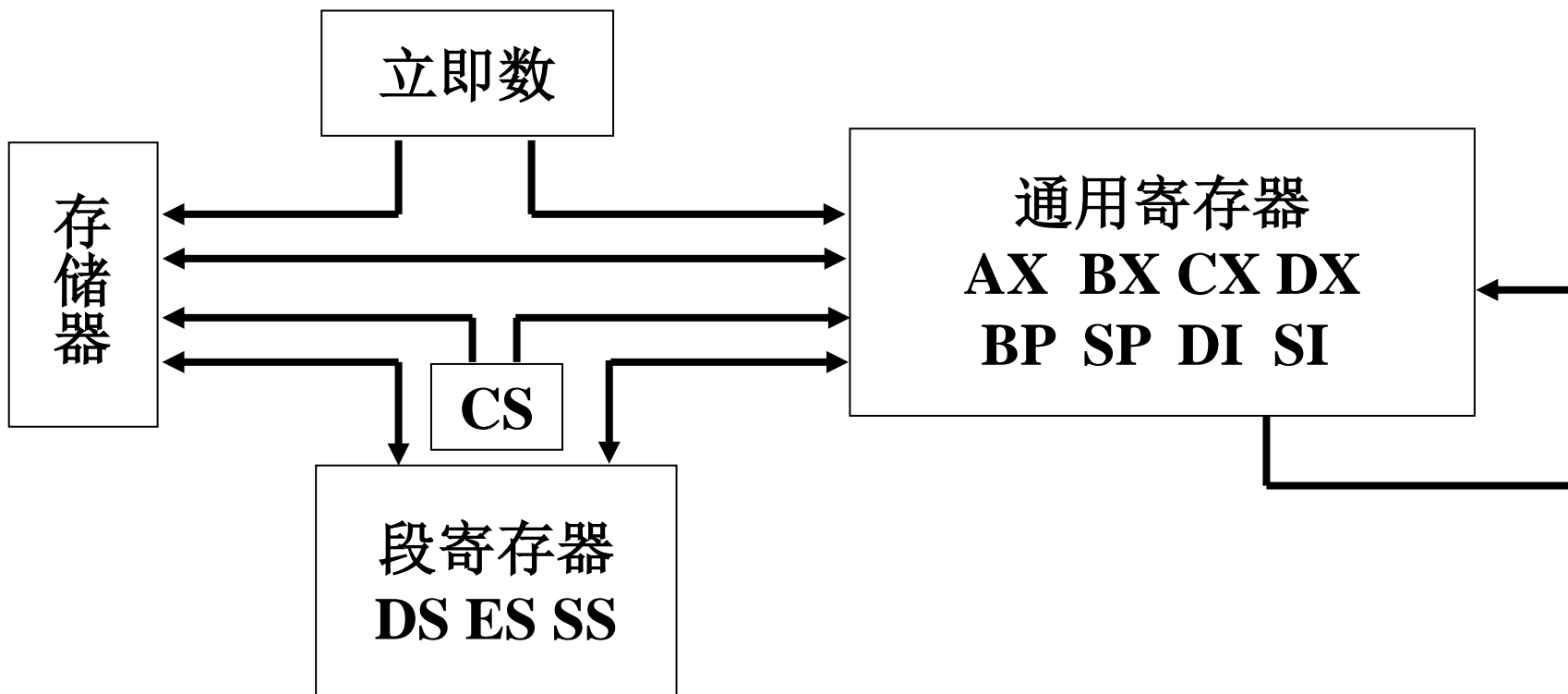


1) 一般数据传送指令 **MOV**

格式: **MOV Dest, Src**

操作: **Src**→**Dest**

例: **MOV AL, BL**



MOV传送指令的操作数



对操作数的要求:

- 两操作数长度必须相同;
- 存储单元之间不能直接传送;
- 段寄存器CS只能作源操作数, 段寄存器之间不能直接传送。;
- 在源操作数是立即数时, 目标操作数不能是段寄存器;
- **FLAGS**一般不作为操作数在指令中出现。



例，判断下列指令的正确性：

MOV AL, BX (×)

MOV AX, 05H[SI] (√)

MOV [BX][BP], BX (×)

MOV DS, 1000H (×)

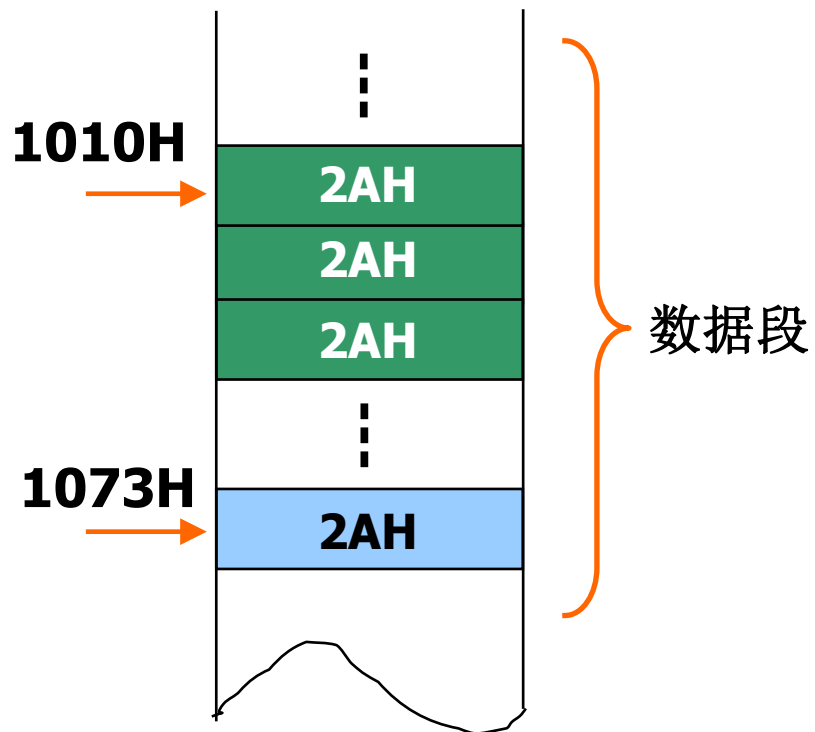
MOV DX, 09H (√)

MOV [1200], [SI] (×)



例，将(*)的ASCII码2AH送入内存数据段1010H开始的100个单元中。

```
MOV DI, 1010H
MOV CX, 100
MOV AL, '*'
AGAIN: MOV [DI], AL
INC DI
DEC CX
JNZ AGAIN
HLT
```





2) 堆栈操作指令

压栈指令

格式: **PUSH OPRD**

出栈指令

格式: **POP OPRD**

①指令的操作数必须是**16位**;

立即数不适用于单操作数指令

②操作数可以是寄存器或存储器字单元，但不能是立即数;

③不能从栈顶弹出一个字给CS;

④**PUSH**和**POP**指令在程序中一般成对出现;

⑤**PUSH**指令的操作方向是从高地址向低地址，而**POP**指令的操作正好相反。

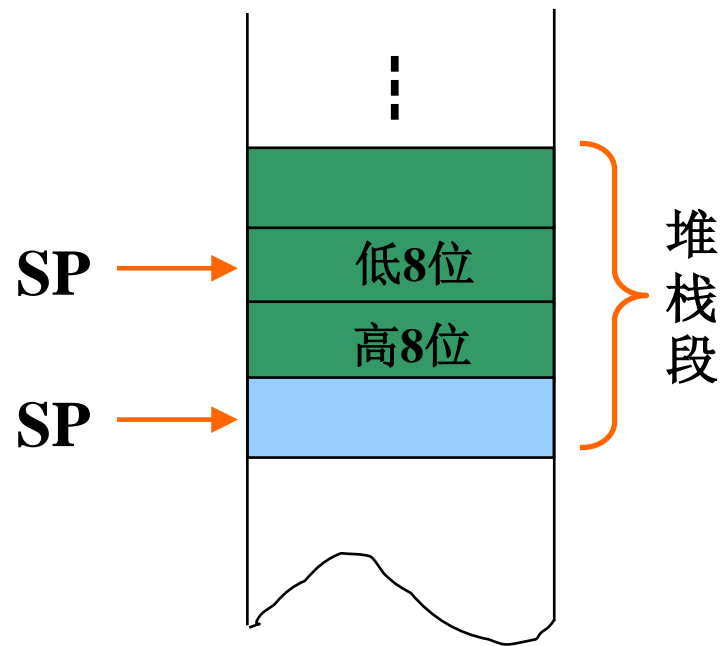


◆ PUSH指令执行过程:

$SP - 2 \rightarrow SP$

操作数高字节 $\rightarrow [SP+1]$

操作数低字节 $\rightarrow [SP]$



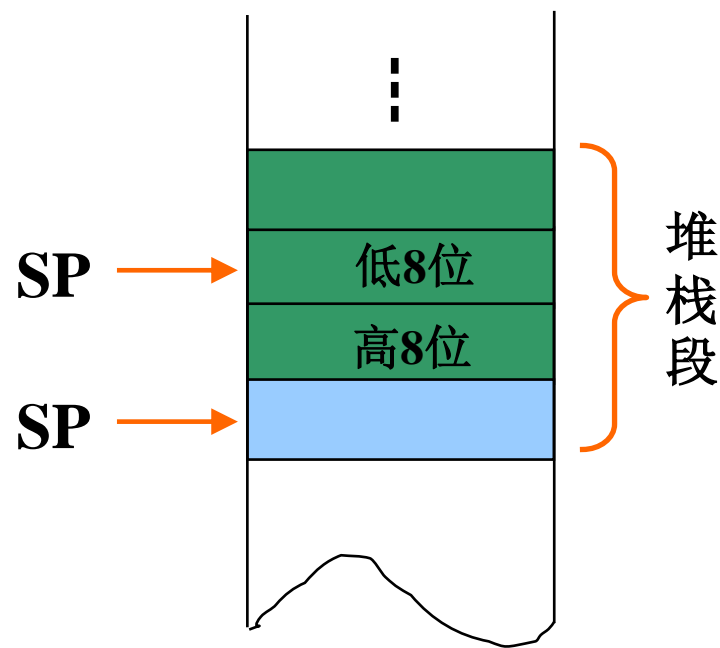


POP指令执行过程:

$[SP] \rightarrow$ 操作数低字节

$[SP+1] \rightarrow$ 操作数高字节

$SP+2 \rightarrow SP$





例, **PUSH AX**
 PUSH BX
 PUSH WORD PTR[BX]
 ⋮
 POP WORD PTR[BX]
 POP AX
 POP BX

$AX \longleftrightarrow BX$



3) 交换指令

XCHG OPRD1, OPRD2

操作: **OPRD1 \longleftrightarrow OPRD2**

将源地址与目标地址中的内容进行互换

①操作数中至少有一个是通用寄存器，即不能同时为存储器。

②段寄存器不能作为操作数。段寄存器主要适用于一般传送和堆栈操作

③两个操作数字长必须相同。

例, **XCHG AX, BX**
 XCHG [2000], CL



4) 查表指令

XLAT

操作：[BX+AL] → AL

- ①指令为零操作数指令，采用隐含寻址。
- ②用**BX**的内容代表表格首地址，**AL**内容为表内位移量，**BX+AL**得到要查找元素的偏移地址。
- ③表格的最大长度不超过**256**字节。



例，假设数据段中存放有一张ASCII码转换表，若首地址为2000H，现欲查出表中第11个代码的ASCII码。

2000H+0	30H	'0'
	31H	'1'
	32H	'2'
	...	
	39H	'9'
2000H+11	41H	'A'
	42H	'B'
	...	
	45H	'E'
	46H	'F'



可用如下指令实现：

MOV BX, 2000H ; **BX**←表首地址

MOV AL, 11 ; **AL**←序号

XALT ; 查表转换

执行后：**AL = 42H**

还可用其他方法实现，如：

MOV BX, 2000H

MOV AL, [BX+0BH]



5) 字位扩展指令

将带符号数的符号位扩展到高位。

指令为零操作数指令，采用隐含寻址，隐含的操作数为AX及AX，DX。

CBW

将AL的符号位扩展到AH。

若AL最高位=1，则执行后AH=0FFH

若AL最高位=0，则执行后AH=00H



CWD

将AX的符号位扩展到DX。

若AX最高位=1，则执行后DX=0FFFFH

若AX最高位=0，则执行后DX=0000H

例，判断以下指令执行后的结果：

MOV AL, 44H

CBW

MOV AX, 0AFDEH

CWD



2 输入输出指令

专门面向I/O端口操作的指令

指令格式：

输入指令： **IN Acc, Port**

输出指令： **OUT Port, Acc**

Port为端口地址，**Acc**为累加器**AL**或**AX**



根据端口地址码的长度，指令具有两种不同的寻址方式。

➤ 直接寻址：

- 端口地址为8位时，指令中直接给出8位端口地址；
- 可寻址256个端口。

➤ 间接寻址：

- 端口地址为16位时，指令中的端口地址必须由**DX**指定；
- 可寻址64K个端口。



例，

IN AX, 80H

MOV DX, 2400H

IN AL, DX

OUT 35H , AX

OUT DX, AX



3 取偏移地址指令

LEA Reg16, Mem

LEA指令将存储器操作数的16位偏移地址送到指定的通用寄存器。

源操作数必须是存储器操作数，目标操作数必须是16位通用寄存器，而且最好是间址寄存器。



例，若设

$BX=1000H, DS=6000H, [61050H]=33H, [61051H]=44H$ 。

比较以下指令的执行结果。

① **LEA BX, [BX+50H]**

② **MOV BX, [BX+50H]**

① **$BX=1000H+50H=1050H$**

② **$BX=DS:[1050H]=4433H$**



装入地址指针指令LDS和LES

格式:

LDS Reg16, Mem32

LES Reg16, Mem32



Reg16 任意一个16位通用寄存器。



Mem32 必须是一个存储器操作数。



作用 把Mem32存储单元开始的4个字节单元的内容送入通用寄存器和段寄存器**DS**（LDS指令）或**ES**（LES指令）。



其他 低字单元内容为偏移量送通用寄存器，高字单元内容为段基值送**DS**或**ES**。



4

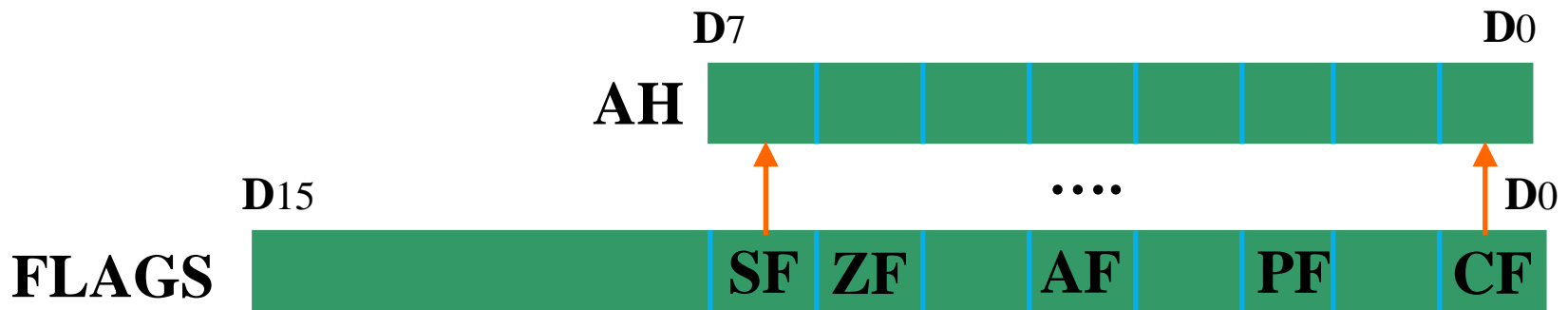
标志传送指令

LAHF
SAHF } 隐含操作数AH和FLAGS

PUSHF
POPF } 隐含操作数FLAGS和堆栈单元



LAHF, 将**FLAGS**低8位的内容装入**AH**。



SAHF, 将**AH**的内容送到**FLAGS**的低8位。

PUSHF, 将**FLAGS**的内容压栈保存。

POPF, 将当前堆栈栈顶的内容送到**FLAGS**。