



石家莊鐵道大學
SHIJIAZHUANG TIEDAO UNIVERSITY

在线开放课程

微型计算机指令系统

数据传送指令-1

主讲：燕延

目录

- 1、MOV 指令
- 2、PUSH与POP指令
- 3、XCHG指令
- 4、XLAT 指令

8086指令系统

8086指令系统有90多种基本指令，按功能可分为6类：

- ①数据传送指令
- ②算术运算指令
- ③逻辑运算和移位指令
- ④串操作指令
- ⑤控制转移指令
- ⑥处理器控制指令

1、 数据传送指令(Data Transfer)

除个别指令外，大部分数据传送指令对标志位无影响。

8086/8088有四类传送指令，分别是：通用数据传送指令、输入输出指令、目标地址传送指令和标志传送指令。

1. 通用传送指令（General Purpose Transfer）

助记符	MOV	PUSH	POP	PUSHF	POPF	XCHG	XLAT
功能	一般 传送	入栈	出栈	标志 压栈	标志 出栈	交换	查表 转换

1) MOV (Movement)

MOV指令可实现寄存器之间、寄存器和存储器之间传送数据，还可实现将立即数送至寄存器或存储单元的操作。

汇编格式：MOV 目的操作数，源操作数

执行的操作： $(\text{目的操作数}) \leftarrow \text{源操作数}$

功能：将源操作数存入目的操作数的寄存器或存储单元中去。

如果把CPU内部的寄存器细分为段寄存器和寄存器的话，则MOV指令有九种形式：

① 从寄存器到寄存器；

MOV AX, BX

② 从寄存器到段寄存器；

MOV DS, AX

③ 从寄存器到存储器；

MOV [SI], AX

④ 从段寄存器到寄存器；

MOV BX, DS

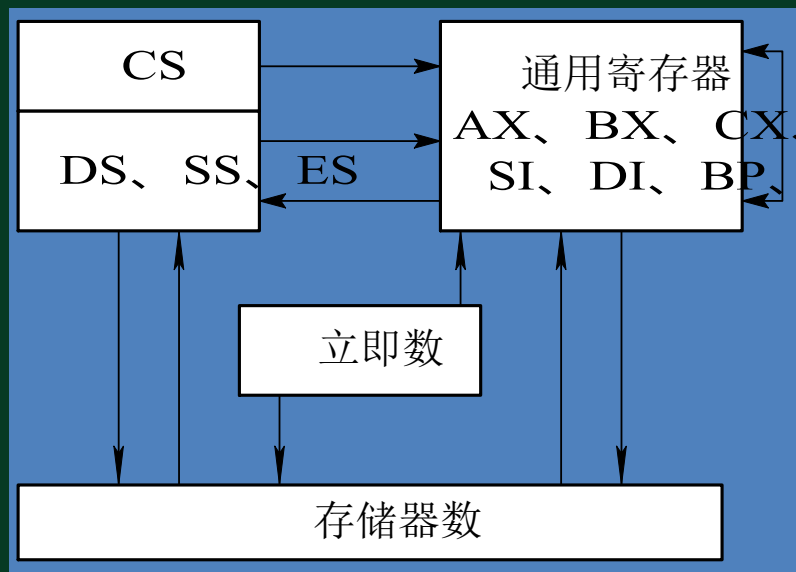


图3.7 MOV指令的九种形式

⑤ 从存储器到寄存器；

```
MOV CX, BUFF[DI]
```

⑥ 从段寄存器到存储器；

```
MOV [BX][SI], ES
```

⑦ 从存储器到段寄存器；

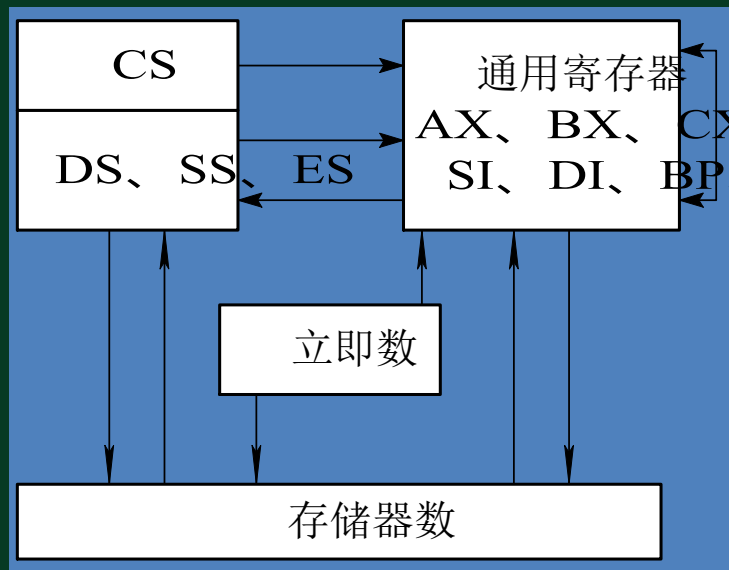
```
MOV SS, [SI],
```

⑧ 从立即数到寄存器；

```
MOV CL, 64H
```

⑨ 从立即数到存储器。

```
MOV TABLE[BX][SI], 7856H
```



注意:

MOV 1234H,AX

① 目的操作数不能是立即数。

② 源操作数不能同时为
存储器寻址方式。寄存器与寄存器之间不能
直接传送数据。

MOV AX, DATA

MOV DS, AX

③ 立即数只能送入寄存器，即段寄存器只能向寄存器单元传送数据。

MOV AX, MEM1

MOV MEM2, AX

- ④ 两个段寄存器之间不允许直接传送数据。
- ⑤ 不允许给**CS**、**IP**、**FLAG**三个寄存器传送数据，即这3个寄存器的值用户无权改变。
- ⑥ 源操作数和目的操作数数据类型必须匹配。
- ⑦ **MOV**指令不影响标志位。

MOV AL,BX MOV DS, SS

MOV SI, BH MOV CS, AX

都是错误指令，类型匹配问题是常见错误。

【例1】 编程实现将内存BUFFER1开始的200个字节数据传送到BUFFER2开始的另一片存储区中。

```
MOV SI, OFFSET BUFFER1  BUFFER1
```

```
MOV DI, OFFSET BUFFER2
```

```
MOV CX, 200
```

```
NEXT: MOV AL, [SI]
```

```
MOV [DI], AL
```

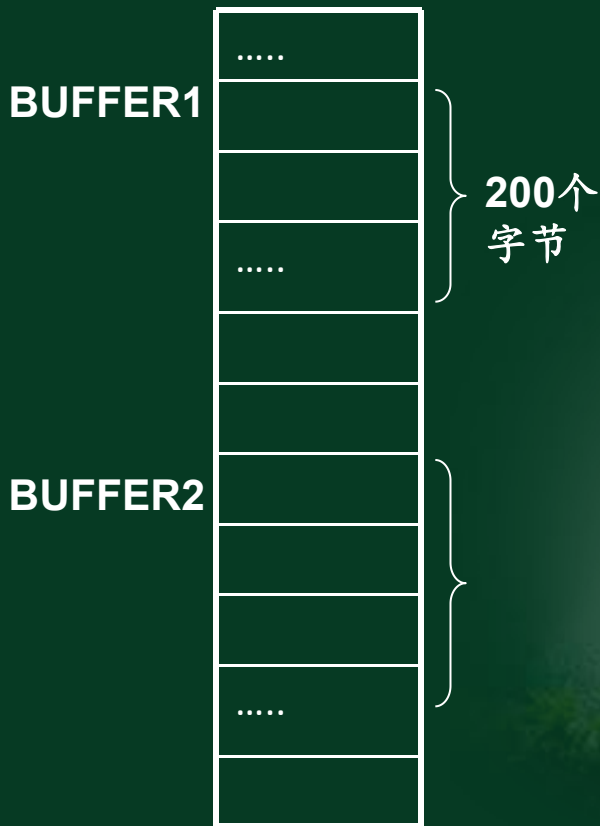
```
INC SI
```

```
INC DI
```

```
DEC CX
```

```
JNZ NEXT
```

```
.....
```



堆栈基本知识

堆栈的段地址由堆栈指针(SI)指出，当前的堆栈指针(SI)指向堆栈的顶部。如果有新的数据要入栈，必须及时修改堆栈指针(SI)的位置。

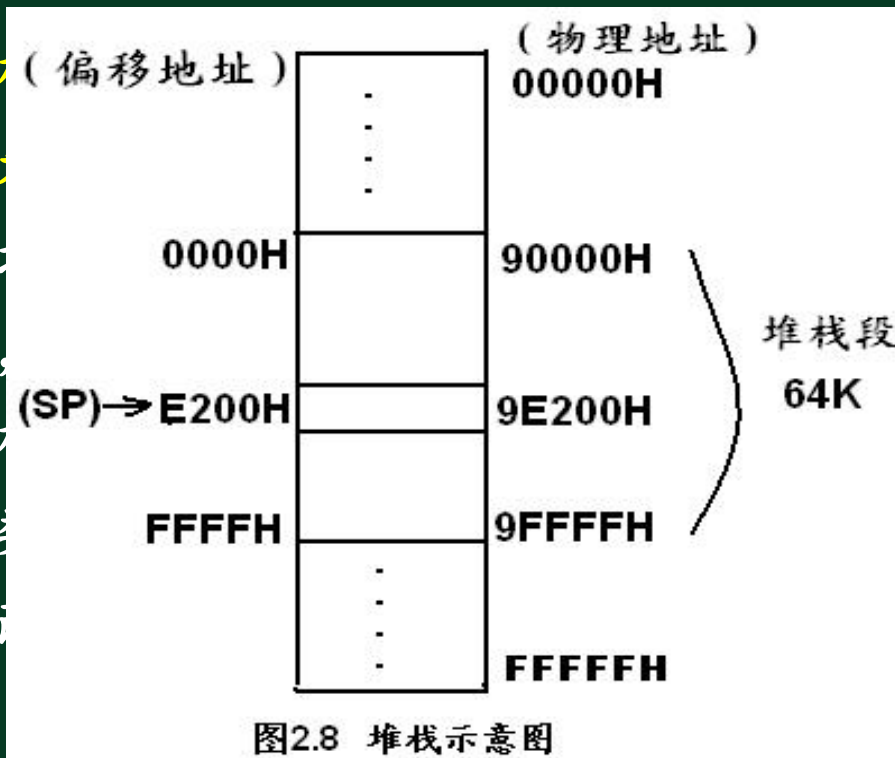


图2.8 堆栈示意图

的，它数据指向确定。必须栈顶

堆栈操作：
入栈（压栈）
出栈（弹出）



2) 进栈指令PUSH及出栈指令POP

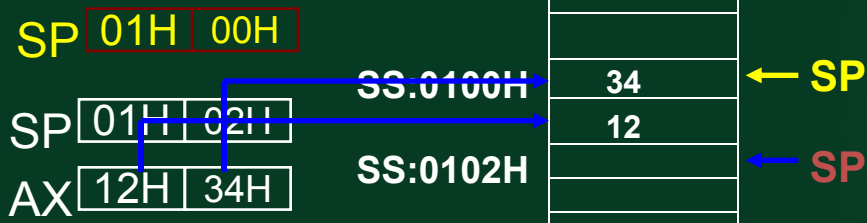
压栈指令PUSH: **PUSH src**

功能: $SP \leftarrow SP - 2$, $[SP] \leftarrow src$,
将栈顶指针上移两个字节后将
源操作数压入堆栈

PUSH AX

PUSH BP

PUSH DATA[SI]



入栈操作先修改指针数据后入栈

出栈指令POP: **POP dest**

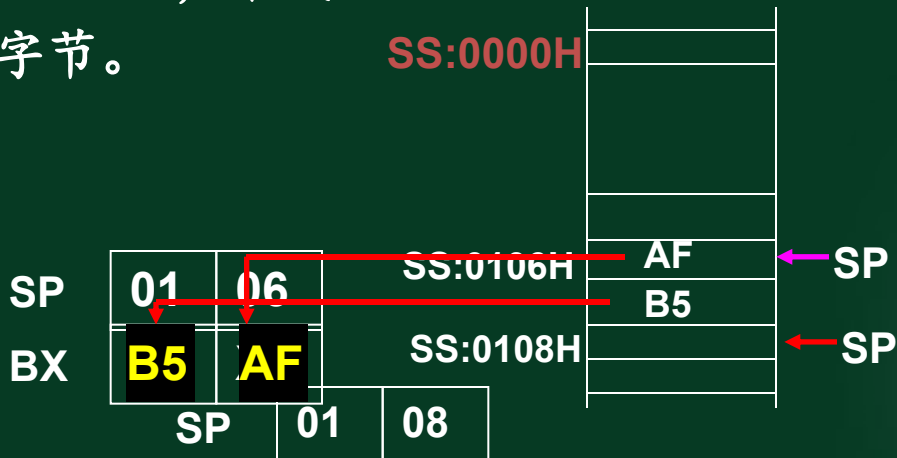
功能: $DEST \leftarrow [SP]$ 、将栈顶内容弹出到目的操作数中, $SP \leftarrow SP + 2$, 堆栈的栈顶指针下移两个字节。

POP BX

POP DI

POP ES

POP ALPHA[BX]



出栈操作数据先出栈后修改指针

说明:

① 在8086/8088中，PUSH、POP指令的操作数不能使用立即寻址方式。POP指令的操作数还不能使用CS寄存器。

PUSH 1234H, POP 5678H

POP CS 均是错误的指令

② 堆栈中数据的压入、弹出必须以字为单位，所以PUSH和POP指令只能作字操作。

PUSH AL PUSH DL POP CL

POP BH 均是错误指令

③ 这两条堆栈指令不影响标志位。

3) XCHG 指令

互换指令 XCHG 可以实现字互换或字节互换。互换可以在寄存器之间进行，也可以在寄存器和存储单元之间进行。

汇编格式：**XCHG** 目的操作数，源操作数

执行的操作：互换源、目的两个操作数的存放位置。

XCHG BL, DL

XCHG AX, SI

XCHG COUNT[DI], AX

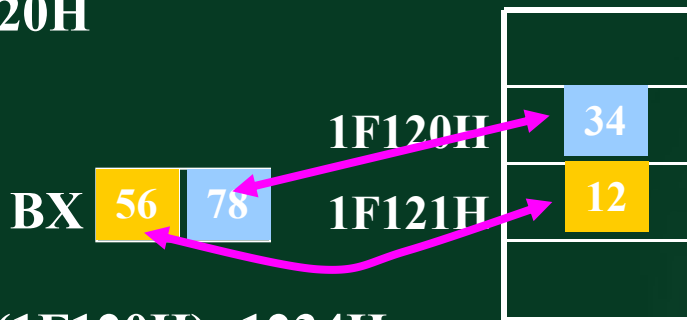
说明：

- ① 源、目的操作数不允许是立即数。
- ② 两个存储单元之间不能直接互换数据。
- ③ 所有的段寄存器以及IP寄存器不允许使用本条指令与其他寄存器互换数据。
- ④ 源、目的操作数类型要匹配。
- ⑤ 此指令不影响标志位。

【例2】 XCHG BX, [BP][SI]

指令执行前, (BX)=1234H, (BP)=0100H, (SI)=0020H,
(SS)=1F00H, (1F120H)=5678H, 则

源操作数物理地址=1F00H×10H+0100H+0020H
=1F00H×10H+0120H=1F120H



指令执行后: (BX)=5678H, (1F120H)=1234H

比较 MOV BX, [BP][SI] 执行后, BX 寄存器的内容是什么?

或者 MOV [BP][SI], BX 执行后 [BP][SI] 所指向单元的内容?

4) XLAT换码指令

汇编格式：**XLAT** 或XLAT 地址标号

执行的操作： $(AL) \leftarrow ((BX) + (AL))$

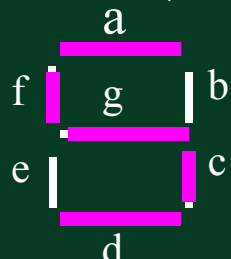
XLAT指令是将AL的内容替换成存储单元中的一个数，常用于代码转换，例如，把数字和字符转换成ASCII码或者把十六进制数0~F转换成七段数码管显示代码，平方、立方表等。

使用换码指令XLAT前需要做的工作：

- ①利用存储地址的连续性和规律性先在数据段建立一个表格，
- ②表格首地址存入BX寄存器，
- ③欲取代码的表内位移量存入AL寄存器中。

共阴极数码管显示原理及字型表：

0	1	2	3	4	5	6	7	8	9
3F	06	5B	4F	66	6D	7D	03	7F	6F



g f e d c b a
d₆ d₅ d₄ d₃ d₂ d₁ d₀

1 1 0 1 1 0 1=6DH

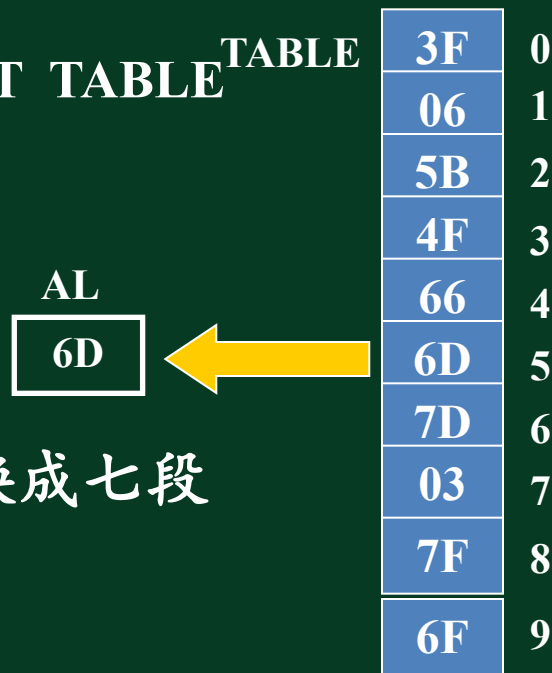
【例3】 一个七段LED显示代码转

换表存于TABLE开始的存储区，则：

```
MOV AL, 5
```

```
MOV BX, OFFSET TABLETABLE
```

```
XLAT
```



完成了将数字 5 转换成七段
LED显示代码的工作。

小结

- 1、MOV指令及使用要点
- 2、PUSH、POP指令及使用要点
- 3、XCHG指令及使用要点
- 4、XLAT指令及使用要点