



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-2

主讲：燕延

目录

1、输入输出指令：

IN、OUT指令

2、目标地址传送指令：

LEA LDS LES指令

3、标志传送指令：

LAHF、SAHF PUSHF POPF

1、输入输出指令

输入输出指令共两条IN和OUT。输入指令IN用于从I/O端口接收数据，输出指令OUT向端口发送数据。无论接收到的数据和准备发送的数据都必须在累加器AX或AL中。

指令格式： **IN acc, port** ;(acc) (port)

OUT port , acc ; (port) ← (acc)

I/O指令可以采用8位(单字节)或16位(双字节)地址两种寻址方式。如采用单字节作为端口地址,则最多可以有256个端口(端口地址号从00H~FFH),可用**直接寻址**(直接端口寻址)方式。

IN AL, 20H

IN AX, 0FCH; 将端口0FCH输入的内容→AL
;将端口0FDH输入的内容→AH

OUT 21H, AL

OUT 0F0H, AX; 将(AL)→端口0F0H;
将(AH)→端口0F1H。

如用双字节地址作为端口地址，则最多可以有64 K个端口(端口地址号从0000H~FFFFH)，并且是间接寻址方式，即把端口地址放在DX寄存器内(间接端口寻址)。

输入：MOV DX, 200H ; 16位地址

IN AX, DX ; 16位传送

或 IN AL, DX ; 8位传送

输出：MOV DX, 1000H

OUT DX, AX ; 16位传送

或 OUT DX, AL ; 8位传送

当端口地址>0FFH时必须用DX间接寻址外设端口。

2、目标地址传送指令(Address-Object Transfer)

1) LEA (Load Effective Address)

格式: **LEA reg16, mem16**

功能: 把源操作数的有效地址传送到目标寄存器。

执行指令: **LEA BX, TABLE**

BX=1450H

2000:1450

23
56

TABLE

MOV BX, TABLE 执行后(BX)?

BX=5623H

LEA BX, TABLE 与 MOV BX, OFFSET TABLE 等效

指令体验:

假设(BP)=0010H, (DI)=0100H, (SS)=1000H,

(10110H)=3AH, (10111H)=48H

LEA BX, [BP][DI] → (BX)=0110H

有效地址 EA=(BP)+(DI)=0110H

MOV BX, [BP][DI] → (BX)=483AH

```

10000
  0010
+  0100
-----
10110
    
```

10110H

10111H

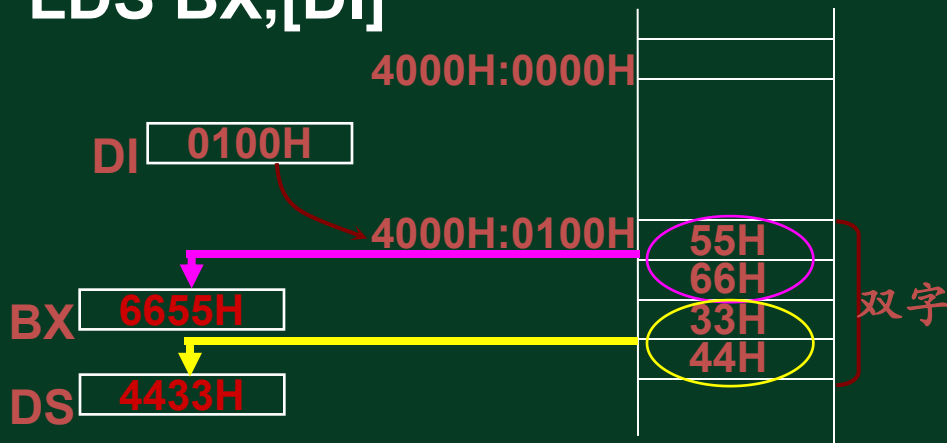
3A
48

2) LDS(Load Pointer using DS)

格式：**LDS reg16, men32**

功能：将寻址到的存储单元的第一个源操作数(字)送16位寄存器，第二个源操作数(字)送DS寄存器。

执行指令：**LDS BX,[DI]**



- 说明：
- ① 本条指令中的16位寄存器不允许是段寄存器。
 - ② 本条指令不影响标志位。

【例1】 LDS BX, TABLE[SI]

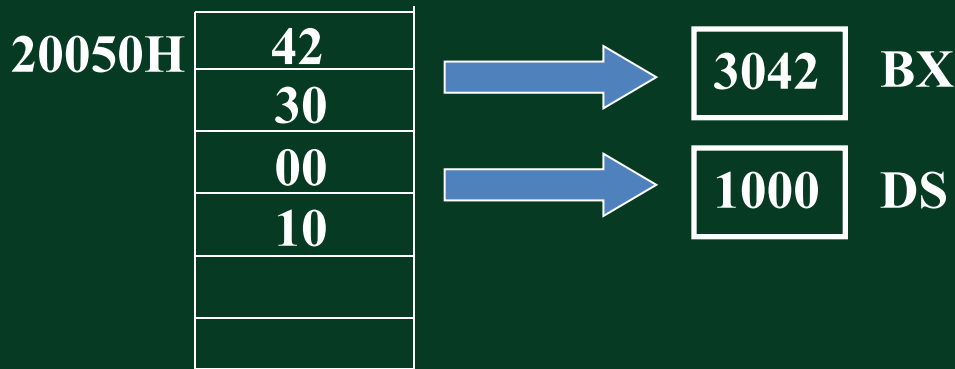
指令执行前，如果假设

(20050H)=10003042H, (BX)=0000H,

(DS)=2000H, TABLE=0020H, (SI)=0030H,

则 EA=0020H+0030H=0050H

物理地址=20000H+0050H=20050H



命令指令执行后，(BX)=3042H, (DS)=1000H。

(DS) = 1000H
2080H	1EH
2081H	00H
2082H	3DH
2080H	20H

(DS) = 203DH
001EH	88H

【例2】执行下列指令：

MOV BX, 2080H

LDS SI, [BX]

MOV AL, [SI]

(SI) = **001EH**

(DS) = **203DH**

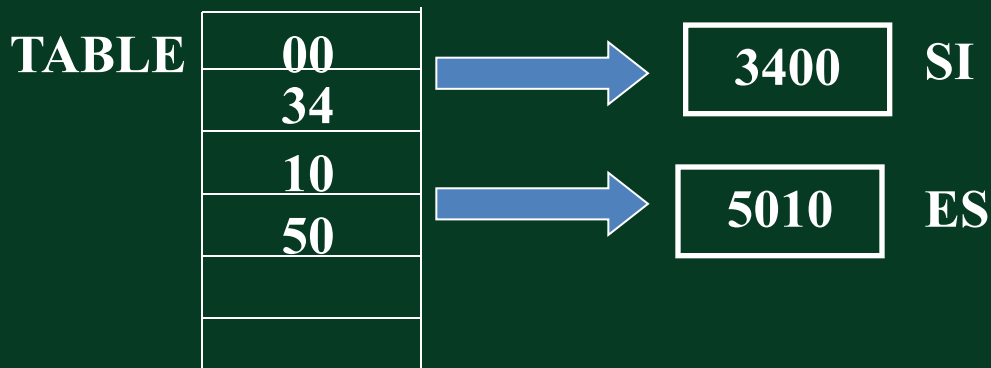
(AL) = **88H**

3) LES (Load pointer using ES)

格式: **LES reg16, men32**

功能: 将寻址到的存储单元的第一个源操作数(字)送16位寄存器, 第二个源操作数(字)送ES寄存器。

例如: **LES SI, TABLE**



- 说明: ① 本条指令中的16位寄存器不允许是段寄存器。
② 本条指令不影响标志位。

3、标志传送指令（Flag Register Transfer）

1) LAHF(Load AH from Flags)

格式：**LAHF**

功能： $(AH) \leftarrow (\text{FLAG的低8位})$



2) SAHF(Store AH into Flags)

格式: **SAHF**

功能: (FLAGS的低8位) \leftarrow (AH)



此指令改变标志寄存器低8位内容。

3) PUSHF (Push Flags onto stack)

格式: **PUSHF**

功能: $(SP) \leftarrow (SP) - 2$, $((SP) + 1) \leftarrow (FLAGS_H)$
 $((SP)) \leftarrow (FLAGS_L)$

将标志寄存器内容压入堆栈。

执行指令 **PUSHF**

SP →

$FLAGS_L$

SP →

$FLAGS_H$



4) POPF (Pop Flags off stack)

格式: **POPF**

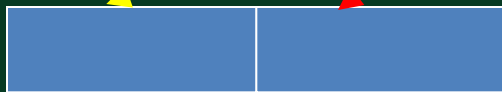
功能: $(\text{FLAGS}) \leftarrow ((\text{SP})+1, (\text{SP}))$

$(\text{SP}) \leftarrow (\text{SP})+2$

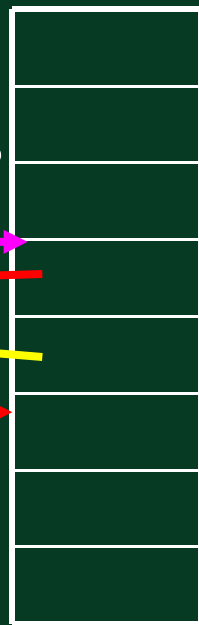
将16位堆栈数据弹出送入标志寄存器中。

执行指令 **POPF**

FLAGS



SP



说明：8086/8088指令系统中没有设置改变TF标志位的指令。若要改变TF值，先用PUSHF指令将标志压栈，然后设法改变对应TF标志位的位值，再用POPF指令弹出送给FLAGS，即可完成改变FLAG中TF标志位的值。

D ₁₅	D ₁₄	D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

【例3】 若想设置TF=1，程序段如下：

PUSHF

POP AX

OR AH, 01H ; 修改TF位

PUSH AX

POPF

AND AH, 0FEH

思考题：若想设置TF=0，程序如何设计？

D ₁₅	D ₁₄	D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

小结

- 1、IN、OUT指令及使用要点
- 2、LEA指令及使用要点
- 3、LDS、LES指令及使用要点
- 4、LAHF、SAHF、PUSHF、POPF指令及使用要点