3.4 数据传送类指令

这些指令又可以分成七个子类：

传送类指令	通用传送类指令	MOV
	获取有效地址指令	LEA
	获取地址指针指令	LDS, LES
	标志传送指令	LAHF, SAHF
	数据交换指令	XCHG
	字节转换指令	XLAT
	堆栈操作指令	PUSH, POP, PUSHF, POPF

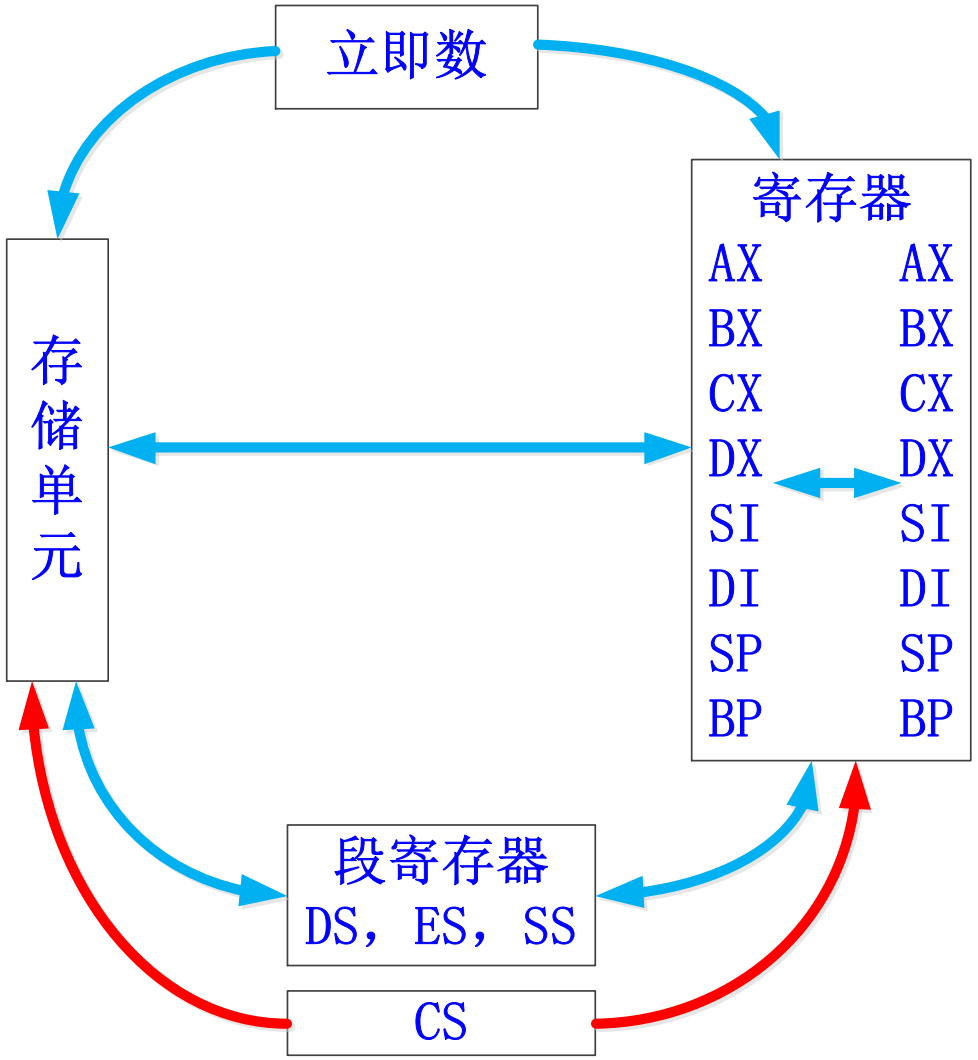
3.4 数据传送类指令

1. 通用传送指令（MOV）

格式：MOV DST，SRC；

(DST)←(SRC)

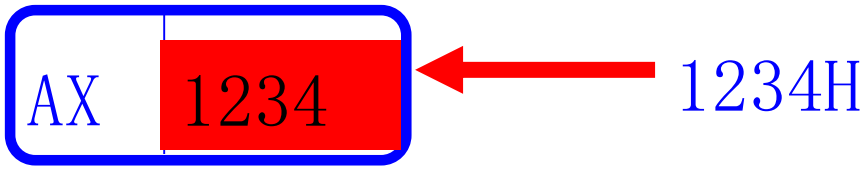
说明：将SRC（源操作数）中的一个字节或一个字传送到DST（目的操作数）所指定的位置。MOV指令可以在立即数、存储单元、寄存器和段寄存器之间传送数据，其传送路径如图所示。



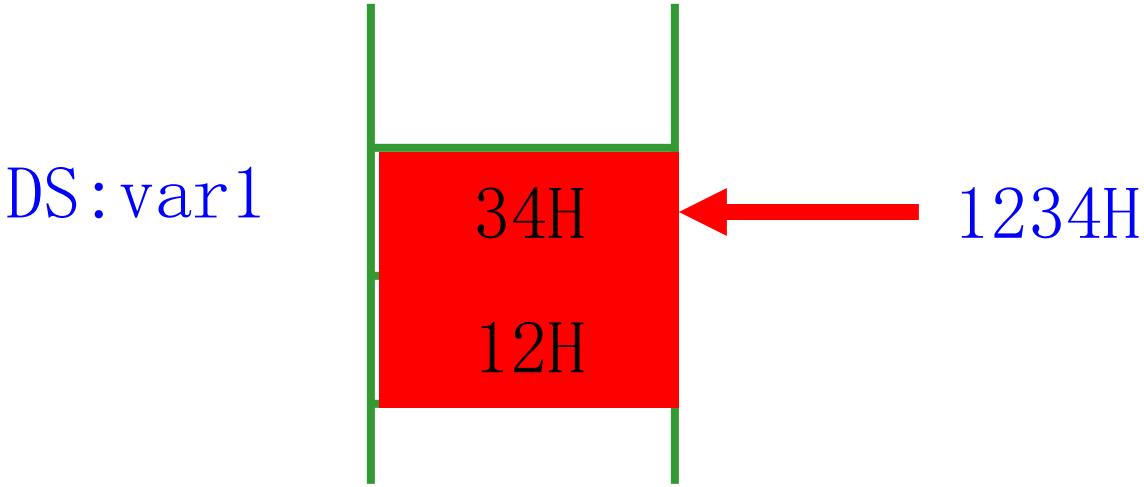
3.4 数据传送类指令

① 立即数→通用寄存器或存储单元

例：MOV AX, 1234H； 将1234H传送到AX中



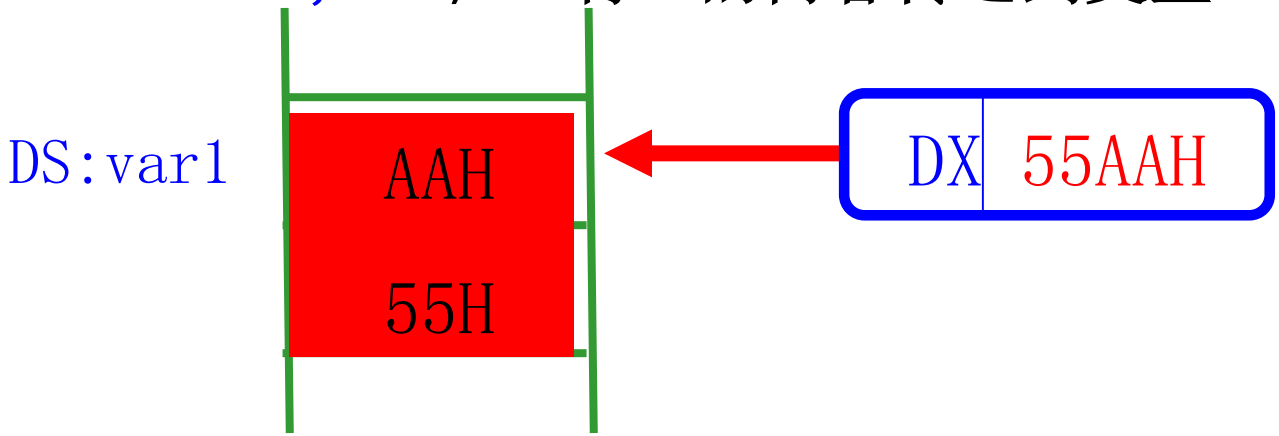
例：MOV var1, 1234H； 将1234H传送到变量var1中



3.4 数据传送类指令

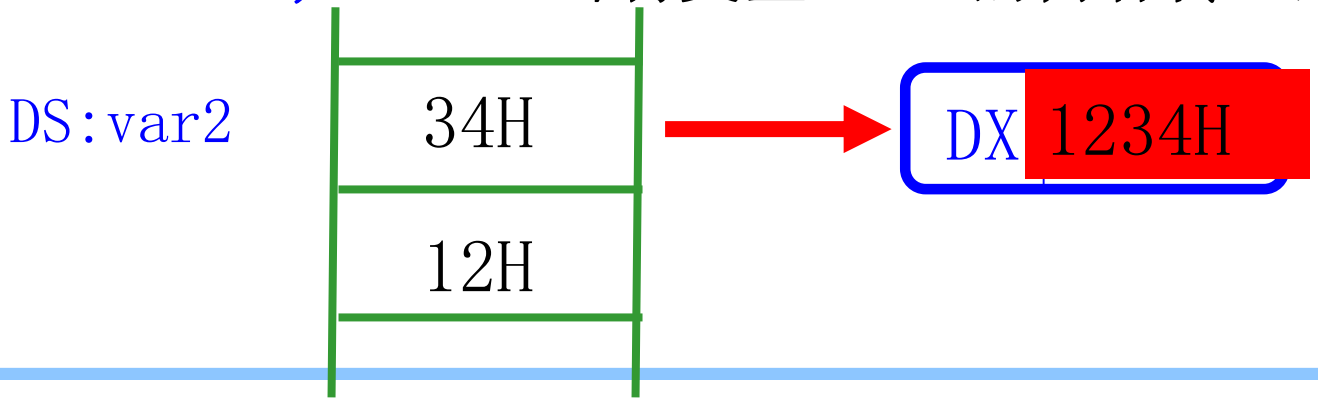
② 通用寄存器→存储单元

例： MOV var1, DX; 将DX的内容传送到变量var1中



③ 存储单元→通用寄存器

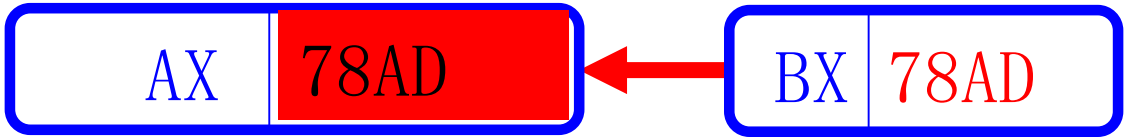
例： MOV DX, var2 ;将变量var2 的内容传送到DX中



3.4 数据传送类指令

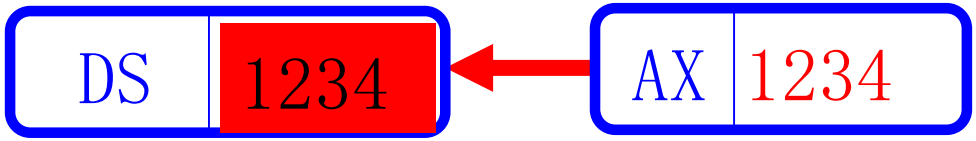
④ 通用寄存器←→通用寄存器

例：MOV　AX，BX；　　将BX的内容传送到AX中

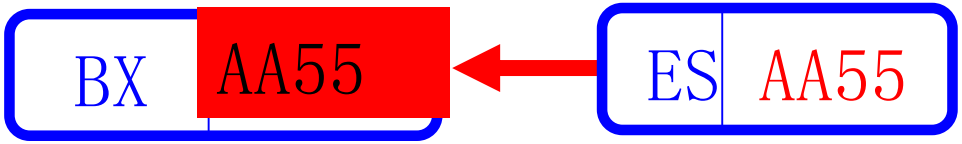


⑤ 通用寄存器←→段寄存器

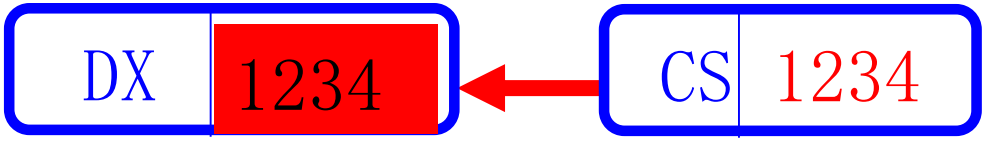
例　　MOV　DS，AX



MOV　BX，ES



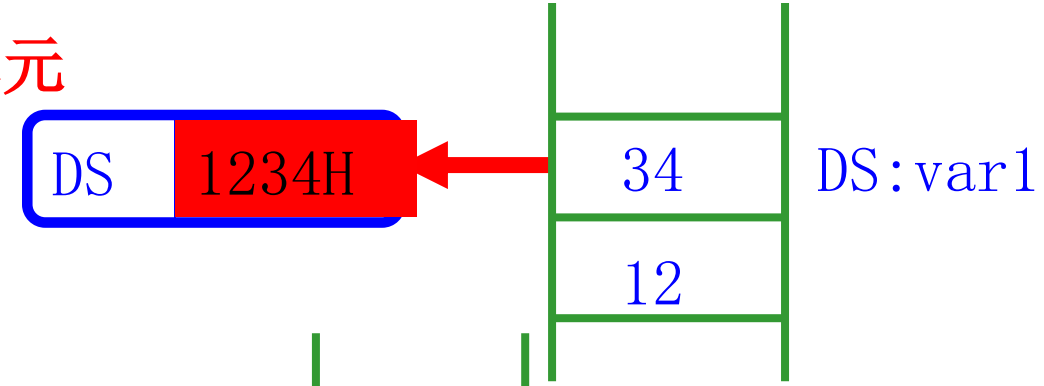
MOV　DX，CS



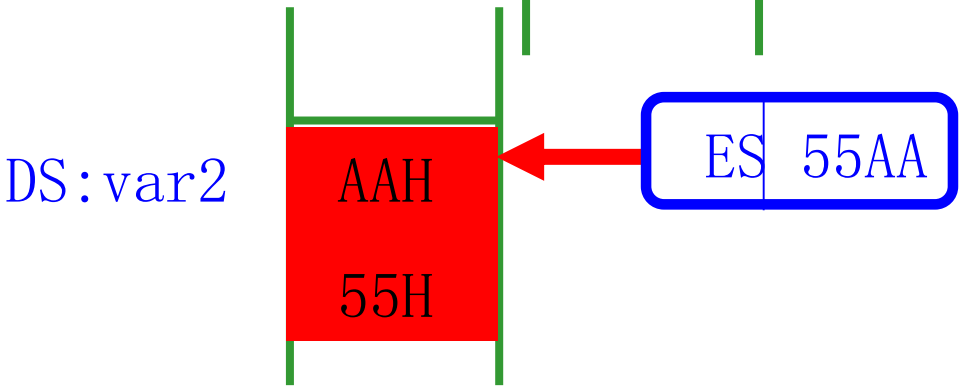
3.4 数据传送类指令

⑥ 段寄存器←→存储单元

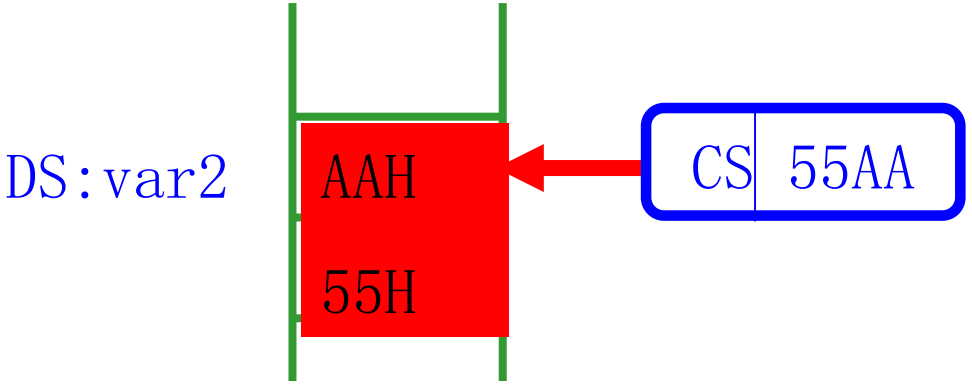
例：MOV DS, var1



例：MOV var2, ES



例：MOV var2, CS



3.4 数据传送类指令

MOV指令不能直达的路径：

（1）立即数×→段寄存器；（2）存储单元×→存储单元

（3）段寄存器×→段寄存器

如果要完成数据在这些路径上的传送，则应该分两步操作

（1）立即数→段寄存器

一般可以通过立即数→通用寄存器→段寄存器来完成

（2）存储单元→存储单元

一般可以通过存储单元→通用寄存器→存储单元来完成

（3）段寄存器→段寄存器

一般可以通过段寄存器→通用寄存器→段寄存器来完成

注：CS不能作为目的寄存器

3.4 数据传送类指令

① 立即数→段寄存器

可以通过立即数→通用寄存器→段寄存器来完成

例：　　MOV　AX，3A01H

　　MOV　DS，AX；　　(DS)←3A01H

② 存储单元→存储单元

可以通过存储单元→通用寄存器→存储单元来完成

例：　　MOV　AX，VAR1

　　MOV　[DI＋10]，AX；　((DI)＋10)←VAR1

③ 段寄存器→段寄存器

可以通过段寄存器→通用寄存器→段寄存器来完成

例：　　MOV　AX，CS

　　MOV　DS，AX；　　(DS)←(CS)

注：CS不能作为目的寄存器

3.4 数据传送类指令

对于双操作数指令，两个操作数的类型必须匹配：

（1）两者都指定了类型，则必须一致，否则指令出错（类型不一致）

（2）两者之一指定了类型，一般指令无错

（3）两者都无类型，则指令出错（类型不定）。

对于操作数的类型，还需注意：

（1）立即数是无类型的

（2）不含变量名的直接寻址、寄存器间接寻址、寄存器相对寻址、基址变址寻址、基址变址且相对寻址的操作数也是无类型的

（3）利用PTR操作符可指定或暂时改变存储单元的类型。

什么样的操作数为立即数，从形式上看，立即数有：

（1）由常数等组成的表达式

（2）所有由属性操作符得到的标号或变量的属性。

3.4 数据传送类指令

常见错误指令：（DATA1为字变量）

MOV 10H, AL; ✗ DST不能为立即数寻址

MOV AL, CX; ✗ 类型不一致

MOV [BX][SI], 78H; ✗ 类型不明确

MOV DATA1, AH; ✗ 类型不一致

MOV [DI]+02H, DATA1; ✗ 两单元之间不能直接传送数据

MOV CS, AX; ✗ CS不能做DST

MOV DS, 0100H; ✗ 当DS作DST时，SRC不能为立即数

3.4 数据传送类指令

2. 取有效地址指令（LEA）

格式：LEA REG，MEM

功能：指令的功能是将源操作数MEM（存储单元）的有效地址（偏移地址）传送到寄存器REG。

这是一条特殊指令，它传送的不是操作数本身，而是操作数的有效地址。

注：目的REG为通用REG，一般用BX、BP、SI、DI。

3.4 数据传送类指令

例： LEA DI, VAR1; (DI) ← VAR1的偏移地址

等效于：

```
MOV  DI, OFFSET VAR1
```

LEA BX, VAR1＋15; (BX) ←VAR1的偏移地址＋15

等效于：

```
MOV  BX , OFFSET VAR1+15
```

3.4 数据传送类指令

3. 取地址指针指令（LDS，LES）

格式：LDS REG16，MEM；

(DS)←(MEM+2), (REG16)←(MEM)

LES REG16，MEM；

(ES)←(MEM+2), (REG16)←(MEM)

功能：

- 取地址指针指令LDS 可以将双字变量MEM内容中的高16位送入DS，低16位送入指定的REG16中；
- 取地址指针指令LES 可以将双字变量MEM内容中的高16位送入ES，低16位送入指定的REG16中。

3.4 数据传送类指令

例：定义变量

```
TABLE    DB 10H, 20H, ……
POINT1   DD 02001000H
POINT2   DD TABLE
```

则：

```
LDS   DI,  POINT1;   (DS)←0200H, (DI)←1000H
LES   SI,  POINT2;   （ES）← TABLE的段地址
                  ;   （SI）← TABLE的偏移地址
```

3.4 数据传送类指令

4. 标志传送指令（LAHF，SAHF）

格式：LAHF　　；(AH) ← PSW寄存器的低8位

　　SAHF　　；PSW寄存器的低8位 ← (AH)

说明：

- 指令LAHF可以将PSW寄存器中的低8位传送到寄存器AH中
- 指令SAHF可以将AH中的内容传送到PSW寄存器中的低8位中
- 源操作数和目的操作数的寻址方式均为隐含寻址方式。

3.4 数据传送类指令

5. 数据交换指令（XCHG）

格式：XCHG DST，SRC； DST ←→ SRC

功能：

- 完成寄存器与寄存器或寄存器与存储单元之间内容交换
- 要求两个操作数之一必须是寄存器，允许两个操作数都是寄存器，但不允许是段寄存器。

注：段REG和立即数不能参加交换。

3.4 数据传送类指令

例：合法指令示例

```
XCHG  AX, BX        ;  (AX) ←→(BX)
XCHG  CX, [DI]      ;  (CX) ←→((DI))
XCHG  DX, VAR1      ;  (DX) ←→(VAR1)
```

例：错误的指令书写格式

```
XCHG  AX, 1234H     ; ✗ 立即数不能参加交换
XCHG  BX, ES        ; ✗ ES段寄存不能参加交换
XCHG  AL, CX        ; ✗ 类型不一致
XCHG  DAT1, DAT2    ; ✗ 两存储器单元不能直接交换
```

例：若（CX）=9A8BH，将CX的高8位与低8位互相交换。

```
XCHG  CH , CL
```

结果(CX) = 8B9AH

3.4 数据传送类指令

6. 字节交换指令（XLAT）

格式：XLAT　；(AL) ←((BX)+(AL))

功能：

- 该指令的寻址方式是隐含的，其有效地址EA = (BX) + (AL)
- 指令的功能是将EA为偏移地址所对应的内存单元中的一个字节的内容送入AL 。

典型实例：指令XLAT非常适合于两个代码之间的转换。

3.4 数据传送类指令

例：如下表所示，将Code1转换成Code2

```
LEA  BX, TABLE
MOV AL, 2
XLAT      ; (AL) = 5BH

          ; EA=(BX)+(AL)
          ; (AL) ←((BX)+(AL))
```

TABLE

3F
06
5B
4F
66
6D
7D
07
7F
6F

Code1	Code2
0	3F
1	06
2	5B
3	4F
4	66
5	6D
6	7D
7	07
8	7F
9	6F

代码变换表的存储　　（共阴）

3.4 数据传送类指令

7. 堆栈操作指令

（1）堆栈

- 堆栈是以后进先出（LIFO）的原则存取信息的一种存储机构。堆栈通常是存储器的一部分。为了保证堆栈区的存储器能按后进先出的规则存取信息，该存储区的存取地址由一个专门的地址寄存器来管理，这个地址寄存器称为堆栈指示器或称堆栈指针SP。
- 在8088/8086系统中，堆栈段的段地址由SS提供，堆栈操作的偏移地址由SP提供。

压入堆栈操作：将信息送入堆栈的过程；

弹出堆栈操作：从堆栈中取出信息的过程。

3.4 数据传送类指令

（2）压入堆栈指令（PUSH）

格式：PUSH SRC； 将SRC压入堆栈，

；（SP）←（SP）- 2，（SP）←（SRC）

PUSHF ； 将PSW压入堆栈，

；（SP）←（SP）- 2，（SP）←（PSW）

说明：压入堆栈指令PUSH先修正堆栈指针SP的内容，然后再将SRC或PSW的内容送入堆栈。SRC必须是字型的，它可以是通用寄存器和段寄存器，也可以是某种寻址方式所指定的存储单元，但不能是立即数。

3.4 数据传送类指令

（3）弹出堆栈指令（POP）

格式：POP DST ；从堆栈弹出DST，

；（DST）←（SP），（SP）←（SP）+ 2

POPF ；从堆栈弹出PSW，

；（PSW）←（SP），（SP）←（SP）+ 2

说明：弹出堆栈指令POP可以取出堆栈的内容，送入DST所指定的寄存器、存储单元或PSW，然后修正SP的内容。DST也必须是字型的，它可以是通用寄存器、段寄存器（CS除外），也可以是存储单元，但不能是立即数。

3.4 数据传送类指令

例：
```
PUSH  AX      ；将（AX）压入堆栈
PUSH  DS      ；将（DS）压入堆栈
PUSH  [SI]    ；将（（SI））压入堆栈
PUSHF         ；将（PSW）压入堆栈
```

例：
```
POP  BX       ；从堆栈弹出一个字，送给（BX）
POP  ES       ；从堆栈弹出一个字，送给（ES）
POP  [SI]     ；从堆栈弹出一个字，送给（（SI））
POPF          ；从堆栈弹出一个字，送给（PSW）
```

3.4 数据传送类指令

常见的错误堆栈操作指令

PUSH AL ；✗ 堆栈只能按字操作

PUSH 5678H ；✗ 不能为立即数寻址

POP CS ；✗ CS不能作DST

3.4 数据传送类指令

（4）堆栈结构

当开辟一块存储区域用作为堆栈时，必须将其段地址置入SS，将SP指向栈底（最后单元的下一个单元地址）。

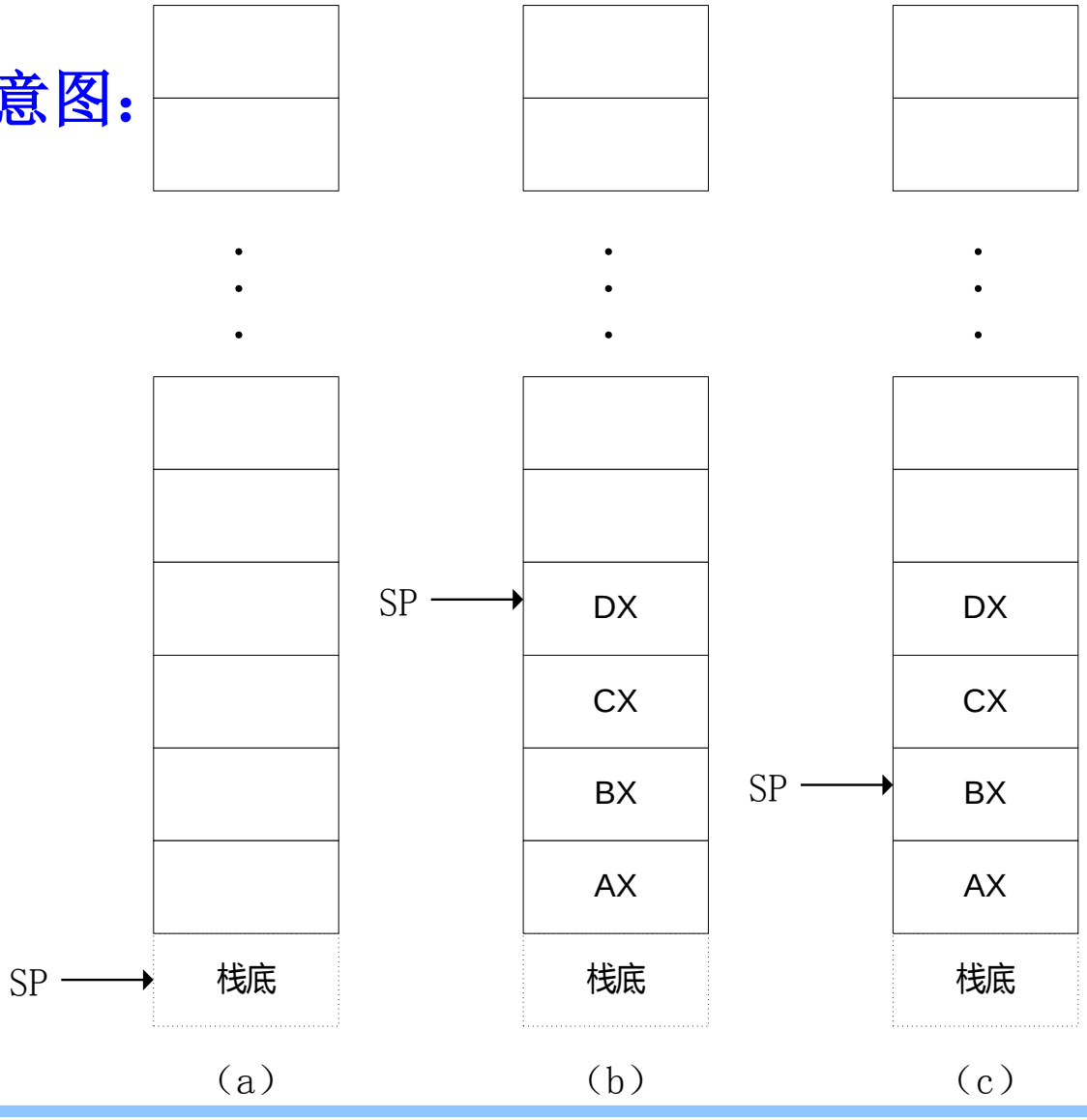
注：在程序设计时，PUSH、POP必须配对使用，尤其是在分支中应该确保PUSH与POP成对出现，以保证堆栈操作的正常次序，这是子程序能够正常返回的条件。

3.4 数据传送类指令

堆栈操作示意图：

```
PUSH   AX
PUSH   BX
PUSH   CX
PUSH   DX

POP    DX
POP    CX
```



3.4 数据传送类指令

应用示例

交换DS和ES的内容

```
PUSH  DS
PUSH  ES
POP   DS
POP   ES
```