

3.6 8086/8088的寻址方式

所谓寻址方式，就是指令中用于说明操作数所在地或者所在地地址的方法。

8088/8086的寻址方式分为两类：

关于寻找数据的寻址方式

关于寻找转移地址的寻址方式

3.6 8086/8088的寻址方式

下面讲关于数据的寻址方式时，均以数据传送指令MOV为例讲解。MOV指令格式如下：



指令完成的功能： (DST) ←—— (SRC)

3.6 8086/8088的寻址方式

一. 关于寻找数据的寻址方式（共8种）

数据的寻址方式就是告诉CPU存/取数据的地方。

1. 立即寻址

操作数直接存放在指令中，紧跟在操作码之后，作为指令的一部分，存放在代码段里，这种操作数称为立即数。

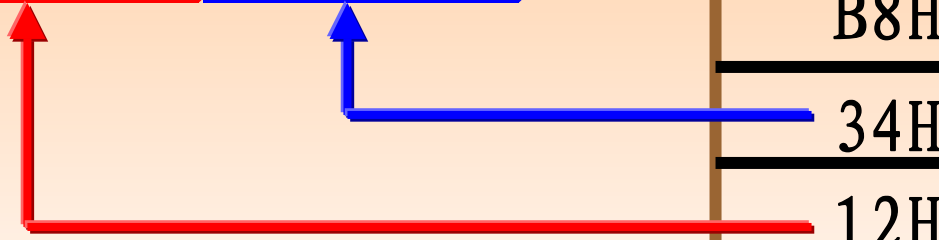
立即寻址主要用来给REG或M赋初值。

注意：只能用于源操作数字段，不能用于目的操作数字段。如：MOV 12H, AL ~~×~~ (语法错误)

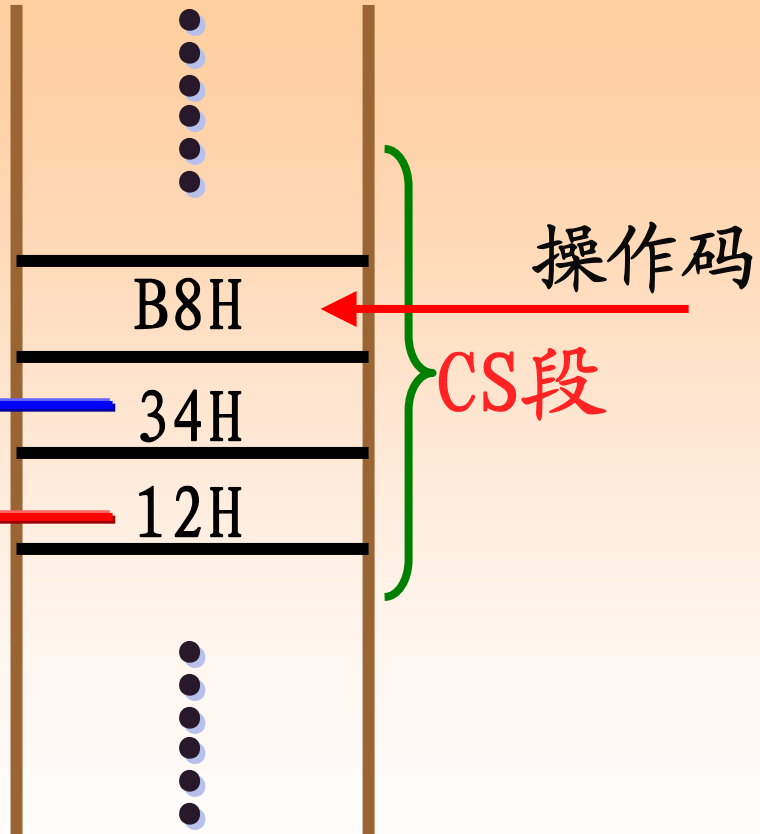
3.6 8086/8088的寻址方式

例: MOV AX, 1234H

A X



∴ (AX) = 1234H



存储器

3.6 8086/8088的寻址方式

2. 寄存器寻址

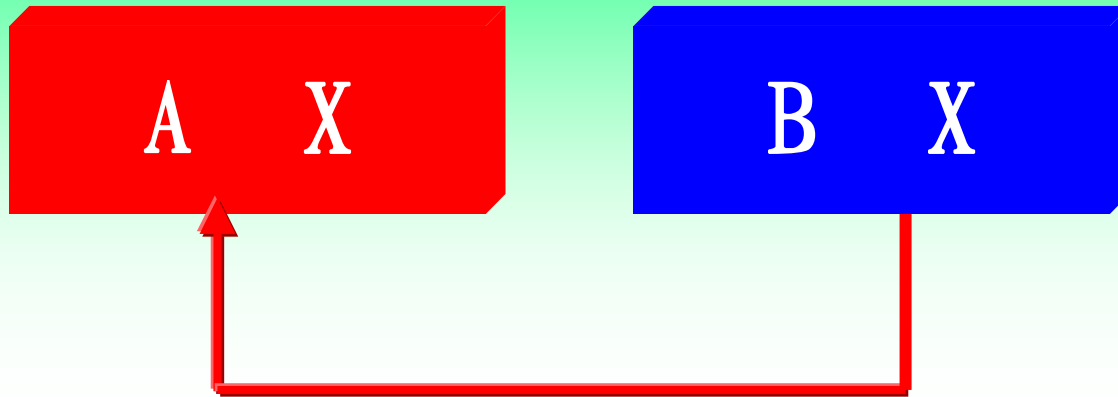
数据放在指令规定的寄存器中，对16位数据，REG可以是AX、BX、CX、DX、SI、DI、SP、BP以及段寄存器，而对于8位数据，REG可以是AH、AL、BH、BL、CH、CL、DH、DL。

在程序设计中，一般存放数据时，寄存器选择通用寄存器，而存放结果时尽可能的使用AX累加器，因为使用AX累加器要比用其它寄存器存放结果，指令执行时间要短一些。

寄存器寻址既可以作DST,也可以作SRC。

3.6 8086/8088的寻址方式

例: MOV AX, BX



若 $(AX) = 1234H$, $(BX) = 5678H$, 则CPU执行上条指令后, $(AX) = 5678H$, 而 (BX) 不变。

又如: MOV CX, DL ~~×~~ (语法错误)

错误原因: 类型不一致。

3.6 8086/8088的寻址方式

3. 存储器寻址

这类寻址方式，操作数在存储器中，而存储器单元的地址由以下五种寻址方式的任何一种均可以找到。但在指令中给出的只是要寻找的操作数所在单元的段内偏移地址，而操作数所在单元的段地址除非指令中用段超越前缀特别指明，否则是默认的。

3.6 8086/8088的寻址方式

① 直接寻址

指令中直接给出了要寻找操作数所在单元的16位偏移地址。

➡ 指令中直接给出的操作数所在单元的16位偏移地址默认在数据段。也可以通过增加段超越前缀来改变操作数所在的段地址。

➡ 操作数所在单元的物理地址：

$$PA = (\text{段寄存器}) \times 16 + \text{指令中给出的偏移地址}$$

3.6 8086/8088的寻址方式

例1: MOV AX, [2000H]

若DS为3000H, 则:

DS 3 0 0 0 0 H

+ 2 0 0 0 H

PA=3 2 0 0 0 H

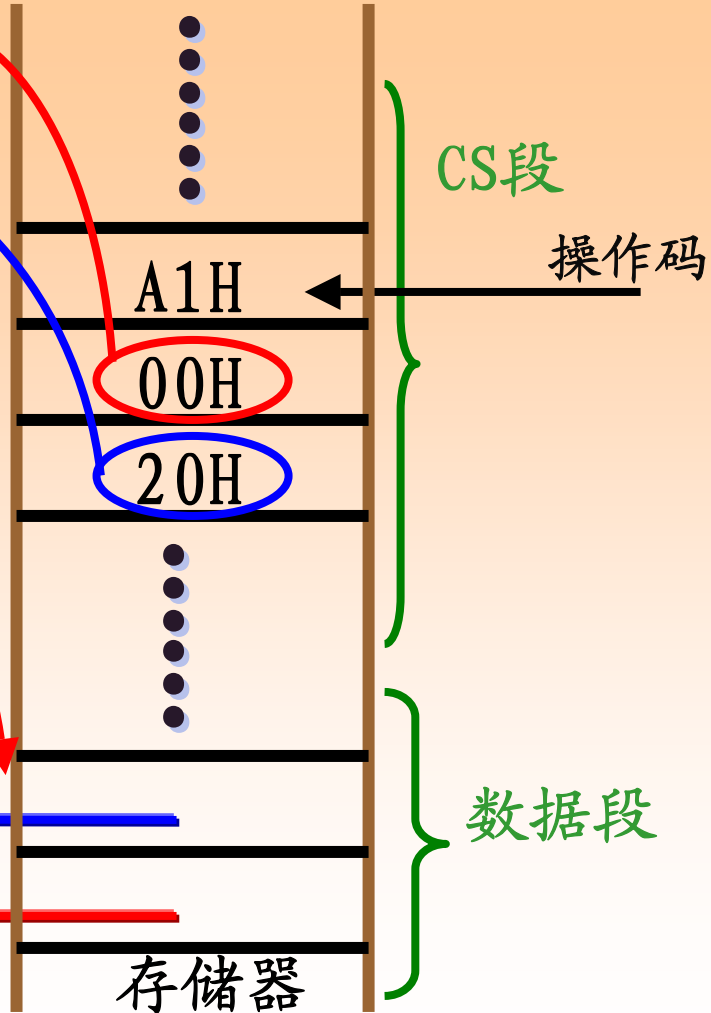
A X

A H

A L

32000H

32001H



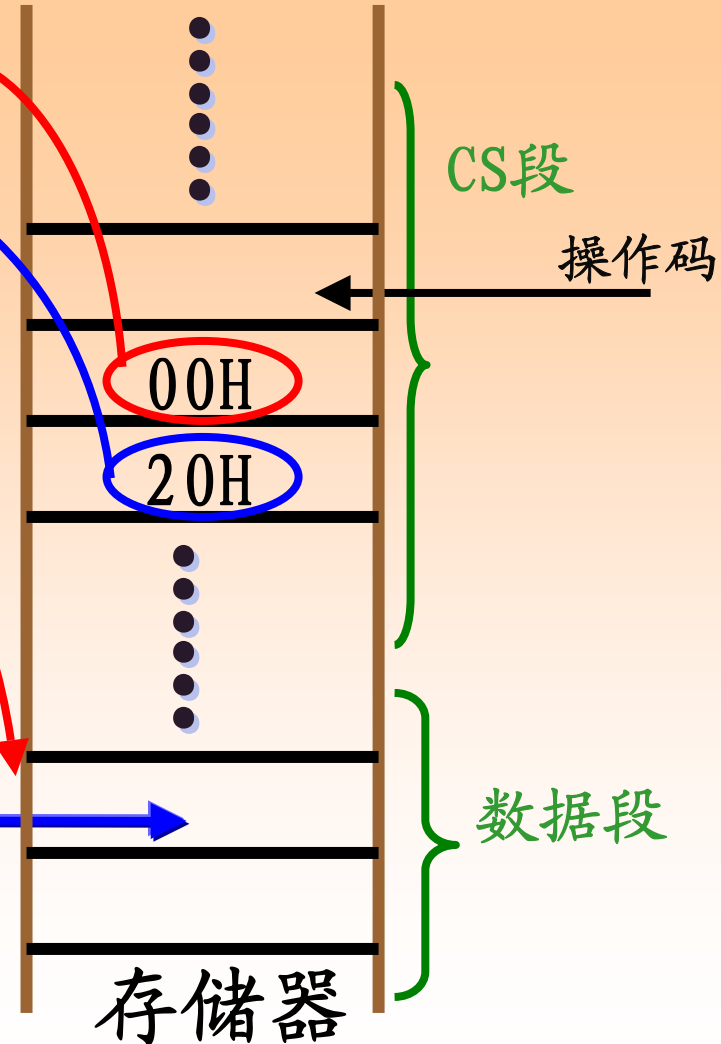
3.6 8086/8088的寻址方式

例2: MOV [2000H], AL
若DS为3000H, 则:

DS	3	0	0	0	0	H
+		2	0	0	0	H
<hr/>						
PA=	3	2	0	0	0	H

AL

32000H



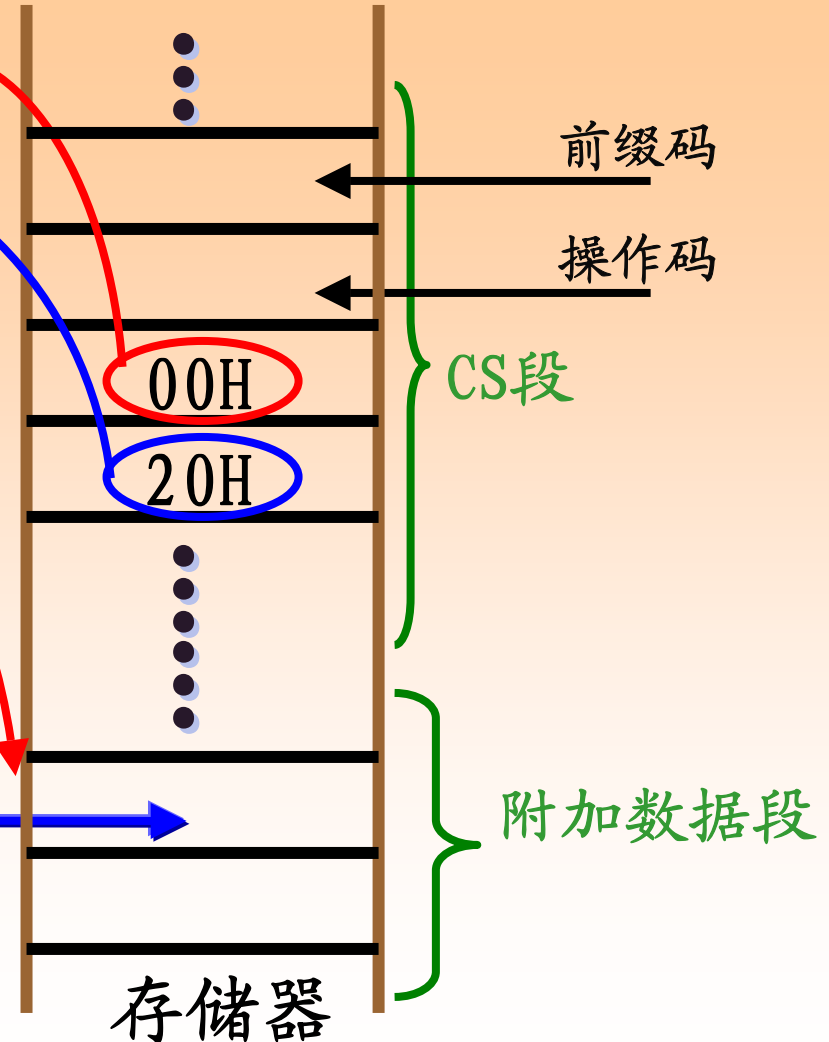
3.6 8086/8088的寻址方式

例3: MOV ES: [2000H], AL
若ES为2050H, 则:

ES	2	0	5	0	0	H
+	2	0	0	0	0	H
<hr/>						
PA=	2	2	5	0	0	H

AL

22500H



3.6 8086/8088的寻址方式

在实际的汇编语言程序设计中，如果程序比较复杂，而用到的存放数据的单元又很多，那么在直接寻址方式当中，用户就要记住存放数据的每个单元的地址，同时还要记住该地址单元存放的数据的意义，这样对设计程序带来了很大的困难。所以在实际的汇编语言程序设计中，常常采用给存放数据的单元，定义一个符号地址名，即**变量名/变量**。

3.6 8086/8088的寻址方式

➡ 变量名一旦定义了，就具有了：

该单元的段地址

该单元的偏移地址

类型

长度

大小

五个属性

这样，在程序设计中就可以用这个变量名代替原来的存储器单元的实际地址。

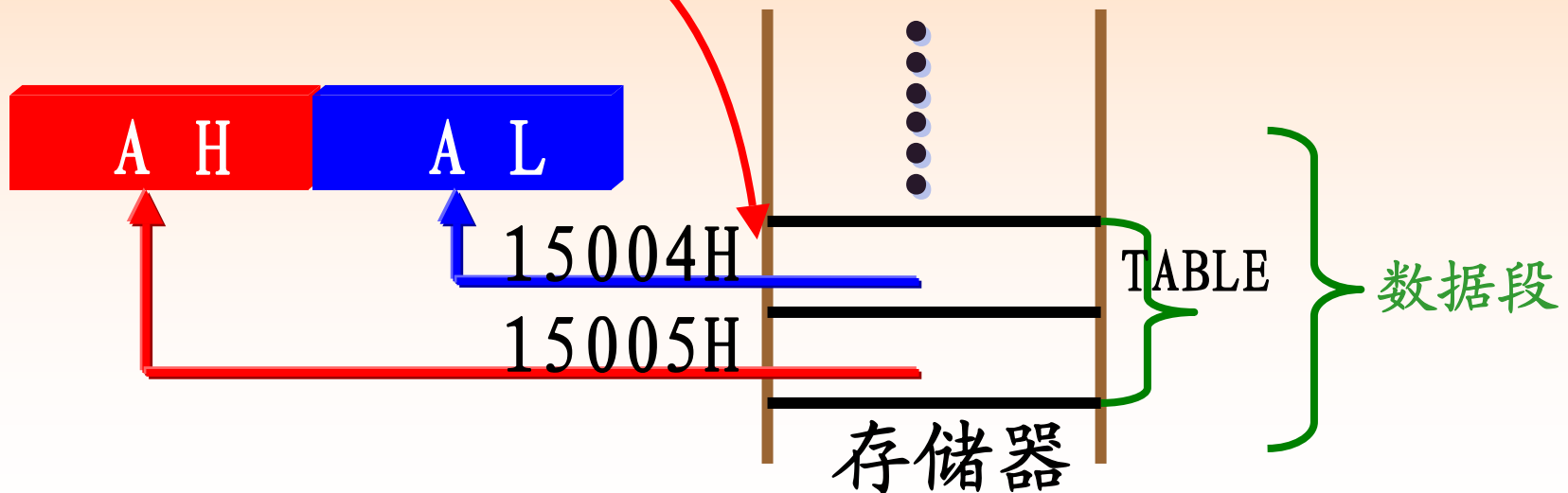
3.6 8086/8088的寻址方式

例4: 若 (DS)=1500H, TABLE为在DS段定义的一个字变量, 且偏移地址为0004H。则CPU执行 MOV AX, TABLE 指令完成的操作如下:

DS 1 5 0 0 0 H

+ 0 0 0 4 H

PA=1 5 0 0 4 H



3.6 8086/8088的寻址方式

例5: 若VAR1为字变量, VAR2和VAR3为字节变量, 判断下列指令的书写格式是否正确, 正确的说出SRC和DST的寻址方式, 不正确说出错误原因。

MOV AX , VAR1

✓ SRC为直接寻址
DST为寄存器寻址

MOV AX , VAR2

✗ 类型不一致

MOV VAR2, VAR3

✗ 两存储器单元之间不能直接传送数据

MOV [0200H] , 12H

✗ 类型不明确

3.6 8086/8088的寻址方式

例6: 将例5中语法不正确的语句改对。

MOV AX , VAR2 × 类型不一致

改: MOV AL , VAR2

MOV VAR2, VAR3 × 两存储器单元之间不

改: { MOV AL , VAR3 能直接传送数据
 MOV VAR2 , AL

MOV [0200H] , 12H × 类型不明确

改: MOV BYTE PTR [0200H] , 12H

或者: MOV WORD PTR [0200H] , 12H

注: PTR为临时属性修改符。

3.6 8086/8088的寻址方式

② 寄存器间接寻址

这种寻址方式，要寻找的操作数在某存储器单元中，该存储器单元地址的段内16位偏移地址在指令中以BX、SI、DI某一个寄存器给出。其段地址默认在DS段。

$$EA = \begin{pmatrix} BX \\ SI \\ DI \end{pmatrix}$$

3.6 8086/8088的寻址方式

例1: MOV AX , [BX]

其SRC为寄存器间接寻址;

DST为寄存器寻址;

指令完成的功能为:

$AX \longleftarrow (DS: (BX))$

若: DS=3000H , BX=1050H

则: SRC所在单元的物理地址为:

$$PA = (DS) \times 16 + (BX)$$

$$= 3000H + 1050H$$

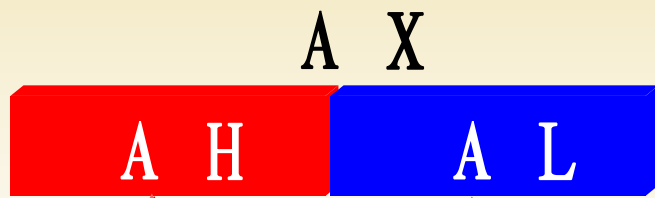
$$= 31050H$$

3.6 8086/8088的寻址方式

DS: 3 0 0 0 0 H

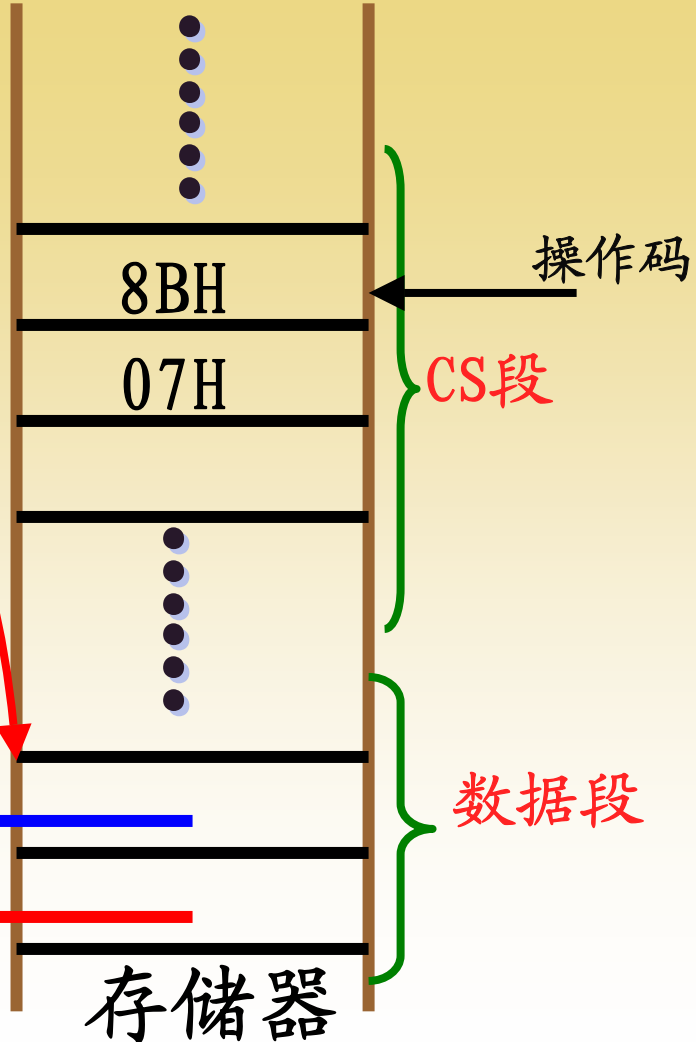
+BX: 1 0 5 0 H

PA: 3 1 0 5 0 H



31050H

31051H



3.6 8086/8088的寻址方式

例2: MOV ES: [SI] , AL

指令完成的功能为:

(ES: (SI)) ← (AL)

若: ES=4000H , SI=1234H, (AL)=23H

则: DST所在单元的物理地址为:

$$PA = (ES) \times 16 + (SI)$$

$$= 40000H + 1234H$$

$$= 41234H$$

∴ 指令执行后 (41234H) = 23H.

3.6 8086/8088的寻址方式

例3: 判断下列指令的书写格式是否正确, 正确的说出SRC和DST的寻址方式, 不正确说出错误原因, 并改正。

① MOV [BX], [SI]

✗ 两存储器单元之间不能直接传送数据; 类型也不明确。

改正: MOV AL, [SI]

MOV [BX], AL

3.6 8086/8088的寻址方式

② MOV [DI] , 12H × 类型不明确

改正: MOV WORD PTR [DI] , 12H

③ MOV [SI] , CX ✓

DST为寄存器间接寻址

SRC为寄存器寻址

3.6 8086/8088的寻址方式

③ 寄存器相对寻址

要寻找的操作数在某存储器单元之中，该单元

的有效地址的一部分在 $\left[\begin{array}{c} \text{BX} \\ \text{BP} \\ \text{SI} \\ \text{DI} \end{array} \right]$ 中，另一部分为一个 $\left\{ \begin{array}{l} 8\text{位} \\ 16\text{位} \end{array} \right\}$ DISP。

其中，DISP \rightarrow 相对位移量

3.6 8086/8088的寻址方式

$$EA = \begin{bmatrix} BX \\ BP \\ SI \\ DI \end{bmatrix} + \begin{bmatrix} 8\text{位DISP} \\ 16\text{位DISP} \end{bmatrix}$$

在DISP为常数时，操作数所在单元的段地址以寄存器为准，若寄存器为BX、SI、DI，操作数默认在DS段中。若寄存器为BP，操作数默认在SS段中。

在DISP为变量时，操作数所在单元的段地址以变量为准，变量在哪个段定义的，就取该段的段地址。

3.6 8086/8088的寻址方式

例1: MOV AX, [BX]+05H

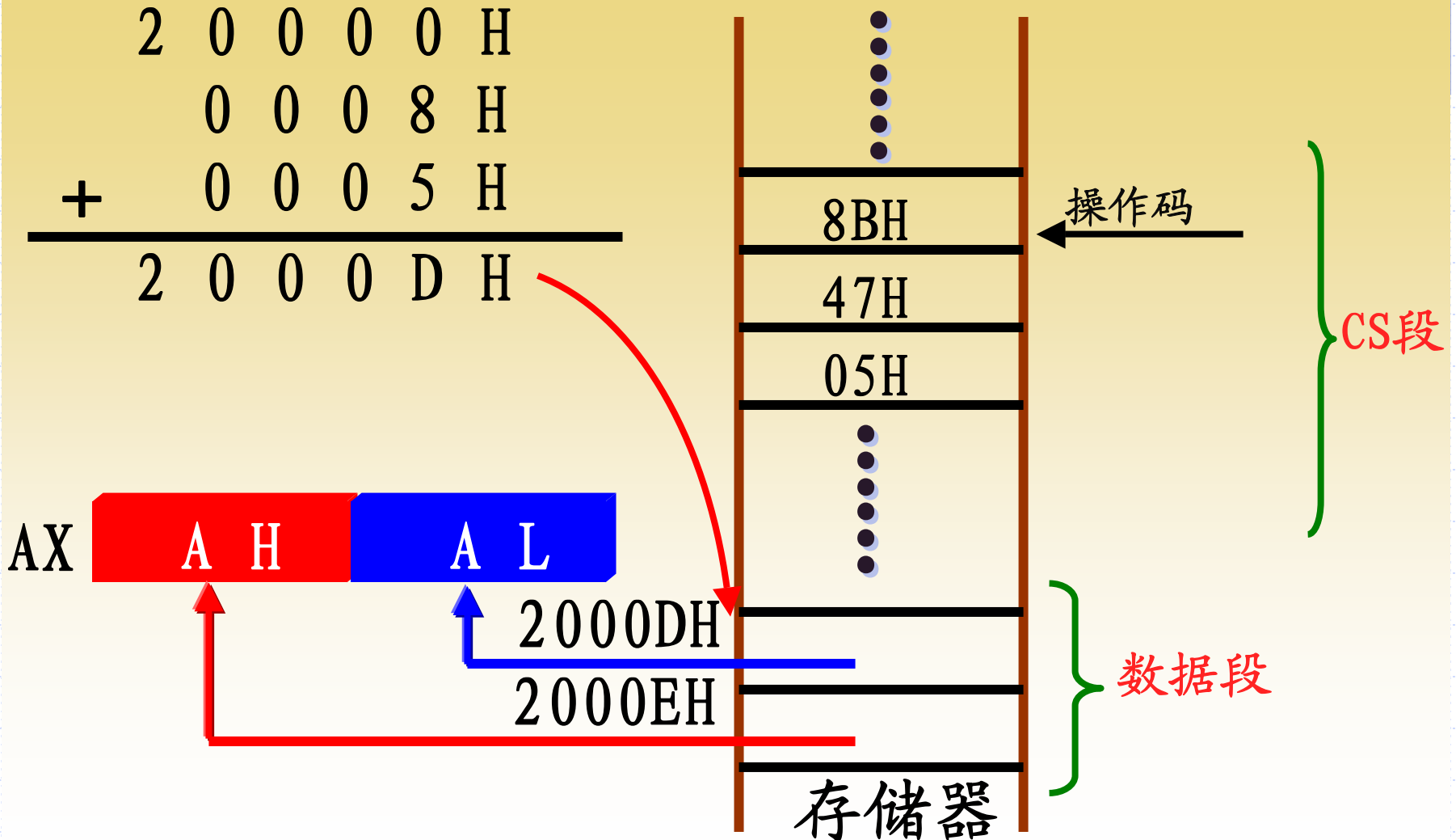
其中, SRC也可以写成:

$\left\{ \begin{array}{l} [BX+05H] \\ 05H[BX] \\ 05H+[BX] \end{array} \right.$

若: DS=2000H, BX=0008H, 存放操作数单元的物理地址为:

$$\begin{aligned} PA &= (DS) \times 16 + (BX) + 05H \\ &= 20000H + 0008H + 05H \\ &= 2000DH \end{aligned}$$

3.6 8086/8088的寻址方式



3.6 8086/8088的寻址方式

例2: MOV AX, [BP]

若: SS=1050H, BP=0050H, 存放操作数的存储单元的物理地址为:

$$\begin{aligned} \text{PA} &= (\text{SS}) \times 16 + (\text{BP}) + 00\text{H} \\ &= 10500\text{H} + 0050\text{H} + 00\text{H} \\ &= 10550\text{H} \end{aligned}$$

3.6 8086/8088的寻址方式

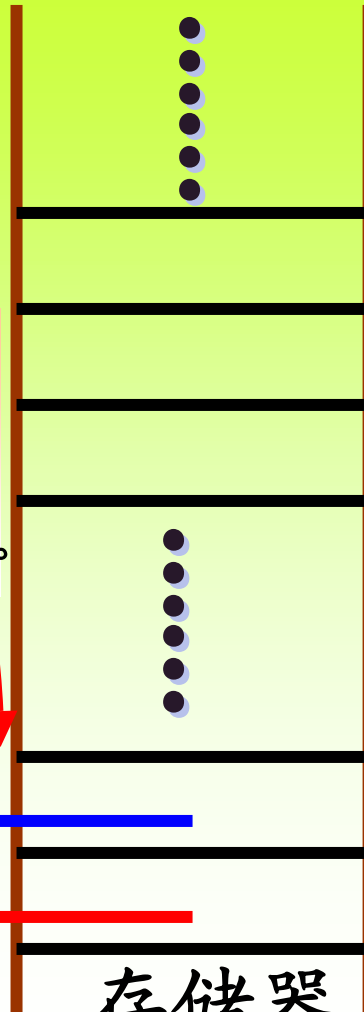
1 0 5 0 0 H
+ 0 0 5 0 H

1 0 5 5 0 H

相对偏移量DISP为0，这里特别指出，BP寄存器无间接寻址，只不过再相对寻址时，DISP为0。



10550H
10551H



操作码

CS段

SS段

存储器

3.6 8086/8088的寻址方式

例3: 若 (DS)=1500H, TABLE为在DS段定义的一个字变量, 且偏移地址为0004H, (BP)=0003H。

MOV AX, TABLE [BP]

SRC的寻址方式为寄存器相对寻址。

指令完成的操作为:

(AX) ← (DS: OFFSET TABLE + (BP))

3.6 8086/8088的寻址方式

④ 基址变址寻址

要寻找的操作数在某存储器单元之中，该单元

有效地址的一部分在 $\begin{bmatrix} \text{BX} \\ \text{BP} \end{bmatrix}$ 中，另一部分在 $\begin{bmatrix} \text{SI} \\ \text{DI} \end{bmatrix}$ 中。

$$EA = \begin{bmatrix} \text{BX} \\ \text{BP} \end{bmatrix} + \begin{bmatrix} \text{SI} \\ \text{DI} \end{bmatrix}$$

该单元的段地址以基址寄存器为准，若基址寄存器为BX，则段地址默认在DS中，若基址寄存器为BP，则段地址默认在SS中。

3.6 8086/8088的寻址方式

例1: MOV AX, [BX] [SI]

(AX) ← (DS: (BX+SI))

例2: MOV AX, [BP] [SI]

(AX) ← (SS: (BX+SI))

例3: MOV [BP] [DI], AL

(SS: (BX+DI)) ← (AL)

3.6 8086/8088的寻址方式

⑤ 基址、变址相对寻址

它是基址变址寻址的扩充，操作数仍在存储器中，存储器单元的有效地址为：

$$EA = \begin{bmatrix} BX \\ BP \end{bmatrix} + \begin{bmatrix} SI \\ DI \end{bmatrix} + \begin{bmatrix} 8\text{位DISP} \\ 16\text{位DISP} \end{bmatrix}$$

同样，如果用BX作为基地址，操作数默认在DS段中；如果用BP作为基地址，则在SS段中。

在DISP为变量时，操作数所在单元的段地址以变量为准，变量在哪个段定义的，就取该段的段地址。

3.6 8086/8088的寻址方式

例1: MOV AX, [BX][DI] 04

(AX) ← (DS: (BX+DI+04H))

例2: MOV AX, [BP][DI] 04

(AX) ← (SS: (BX+DI+04H))

例3: MOV DS: [BP][DI]+04H, AL

(DS: (BX+DI+04H)) ← (AL)

3.6 8086/8088的寻址方式

8. 隐含寻址

有些指令的指令码中不包含指明操作数地址的部分，而其操作码本身隐含的指明了操作数地址。

如：乘除法指令、字符串操作类指令等。

3.6 8086/8088的寻址方式

例1. 若 $(BX) = 0158H$, $(DI) = 10A5H$,
 $(DS) = 2100H$, $DISP = 1B57H$,
 $(BP) = 0100H$, $(SS) = 1100H$,
段寄存器按默认段寄存器, 则相对于各种寻址方式的EA的求法如下:

➡ 直接寻址:

$$EA = 1B57H$$

$$PA = 21000H + 1B57H = 22B57H$$

3.6 8086/8088的寻址方式

例1. 若 $(BX) = 0158H$, $(DI) = 10A5H$,
 $(DS) = 2100H$, $DISP = 1B57H$,
 $(BP) = 0100H$, $(SS) = 1100H$,
段寄存器按默认段寄存器, 则相对于各种寻址方式的EA的求法如下:

➡ 寄存器间接寻址 (设寄存器为BX):

$$EA = 0158H$$

$$PA = 21000H + 0158H = 21158H$$

3.6 8086/8088的寻址方式

例1. 若 $(BX) = 0158H$, $(DI) = 10A5H$,
 $(DS) = 2100H$, $DISP = 1B57H$,
 $(BP) = 0100H$, $(SS) = 1100H$,
段寄存器按默认段寄存器, 则相对于各种寻址方式的EA的求法如下:

➡ 寄存器相对寻址 (以BP为例):

$$EA = 0100H + 1B57H = 1C57H$$

$$PA = 11000H + 1C57H = 12C57H$$

3.6 8086/8088的寻址方式

例1. 若 $(BX) = 0158H$, $(DI) = 10A5H$,
 $(DS) = 2100H$, $DISP = 1B57H$,
 $(BP) = 0100H$, $(SS) = 1100H$,
段寄存器按默认段寄存器, 则相对于各种寻址方式的EA的求法如下:

► 基址变址寻址 (BX、DI) :

$$EA = 0158H + 10A5H = 11FDH$$

$$PA = 21000H + 11FDH = 221FDH$$

3.6 8086/8088的寻址方式

例1. 若 $(BX) = 0158H$, $(DI) = 10A5H$,
 $(DS) = 2100H$, $DISP = 1B57H$,
 $(BP) = 0100H$, $(SS) = 1100H$,
段寄存器按默认段寄存器, 则相对于各种寻址方式的EA的求法如下:

➡ 基址变址相对寻址 (BP、DI) :

$$EA = 0100H + 10A5H + 1B57H = 2CFCH$$

$$PA = 11000H + 2CFCH = 13CFCH$$

3.6 8086/8088的寻址方式

例2: 判断下列指令的书写格式是否正确, 正确的说出SRC和DST的寻址方式, 不正确说出错误原因。

MOV AX, [BX] [SI] ✓ SRC为基址变址寻址
DST为寄存器寻址

MOV AX, BL ✗ 类型不一致

MOV [BP], [DI+01H] ✗ 两存储器单元之间
不能直接传送数据

MOV [BX] [DI]+02H, 12H ✗ 类型不明确

3.6 8086/8088的寻址方式

作业: P61-P62

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
7. (1) (3)
8. (2)
- 9.
- 10.
- 11.

3.6 8086/8088的寻址方式

二. 说明转移地址的寻址方式

寻找的操作数作地址用，给IP或给CS: IP，从而实现程序的转移。

如果程序转移后**只有IP**发生了改变，则称为**段内转移**或者称为**近程转移**（也称为NEAR型转移）。

如果程序转移后**CS、IP**均发生了改变，则称为**段间转移**或者称为**远程转移**（也称为FAR型转移）。

3.6 8086/8088的寻址方式

8086指令系统中的转移指令有两大类：

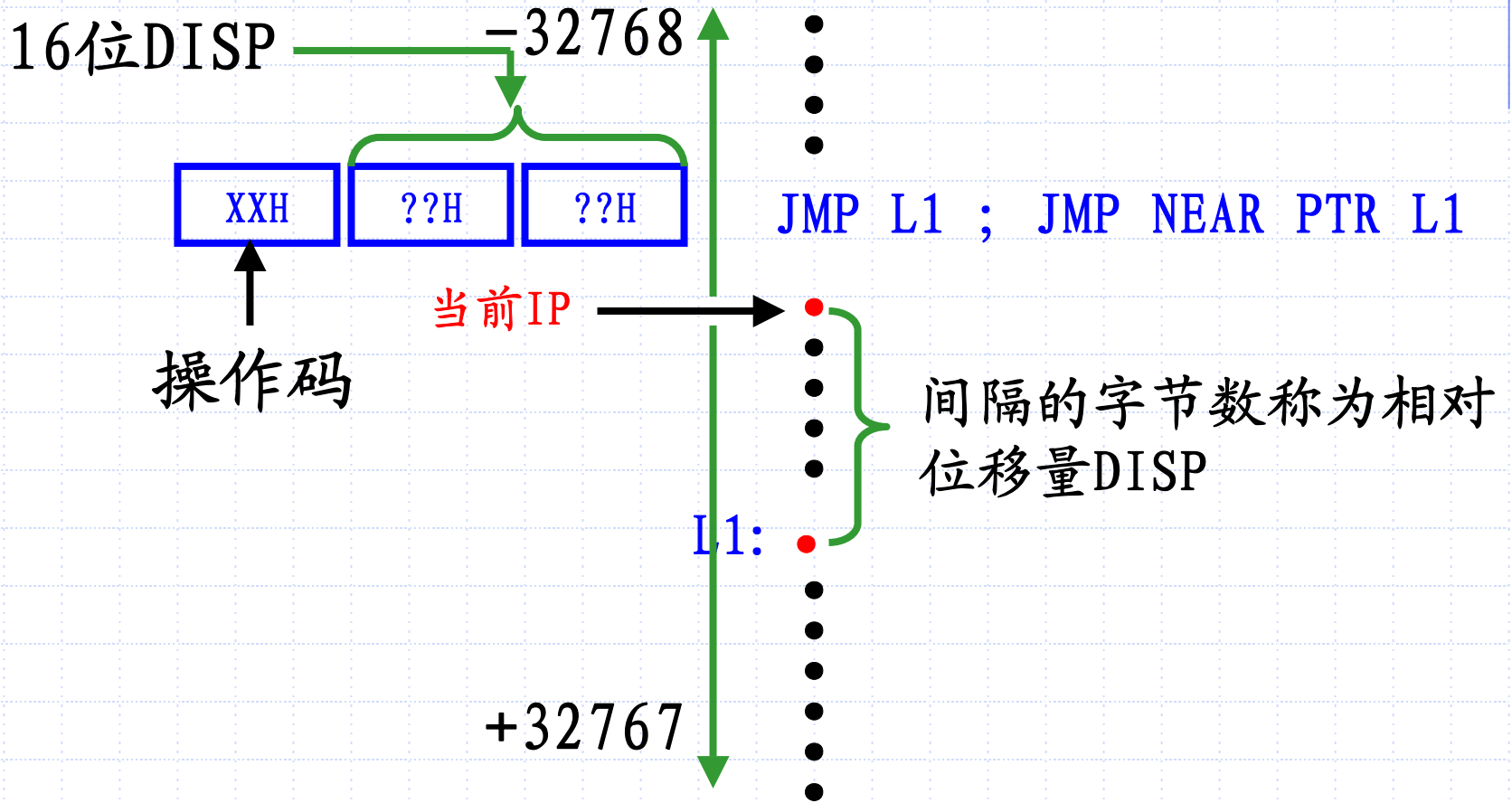
- 无条件转移指令：有JMP、CALL、RET、IRET
- 条件转移指令：如JZ、JC、JCXZ、LOOP等。

段内转移地址的寻址方式（只有IP发生改变）

1. 段内直接寻址（也叫段内相对寻址）

这种寻址方式中，指令中直接写出了转移目的地的符号地址（NEAR型标号名）。

3.6 8086/8088的寻址方式



转移目的地的IP=当前 (IP) +16位DISP

3.6 8086/8088的寻址方式

指令中指定的8bit/16bit的相对位移量DISP是相对于当前IP来计算的，有正、负转移，向地址增加的方向转移，为正转移，DISP用原码表示；向地址减小的方向转移，为负转移，DISP用补码表示。

转移目的地物理地址的求法：

$$PA = (CS) \times 16 + \text{当前}(IP) + DISP$$

注：8086指令系统当中的所有条件转移指令只能在段内转移，且转移范围以当前IP为基准，不能超出-128~+127之间，其寻址方式为段内相对寻址。

3.6 8086/8088的寻址方式

2. 段内间接寻址

这种寻址方式中，转移地址的段内偏移地址要么存放在一个16bit的寄存器中，要么放在存储器的一个字单元之中。这个寄存器或字单元的地址，由前面讲过的数据的寻址方式指明。只不过寻址所得到的不是数据，而是偏移地址。

3.6 8086/8088的寻址方式

例：若 $(DS) = 2000H$, $(BX) = 0100H$, $(SI) = 0002H$,
 $(20100H) = 1200H$, $(20102H) = 1250H$,
则CPU执行:

JMP BX 指令后, $(IP) = 0100H$

JMP WORD PTR [BX] 指令后, $(IP) = 1200H$

JMP WORD PTR [BX][SI] 指令后, $(IP) = 1250H$

3.6 8086/8088的寻址方式

段间转移地址的寻址方式(CS、IP均发生改变)

——只适合于无条件转移指令

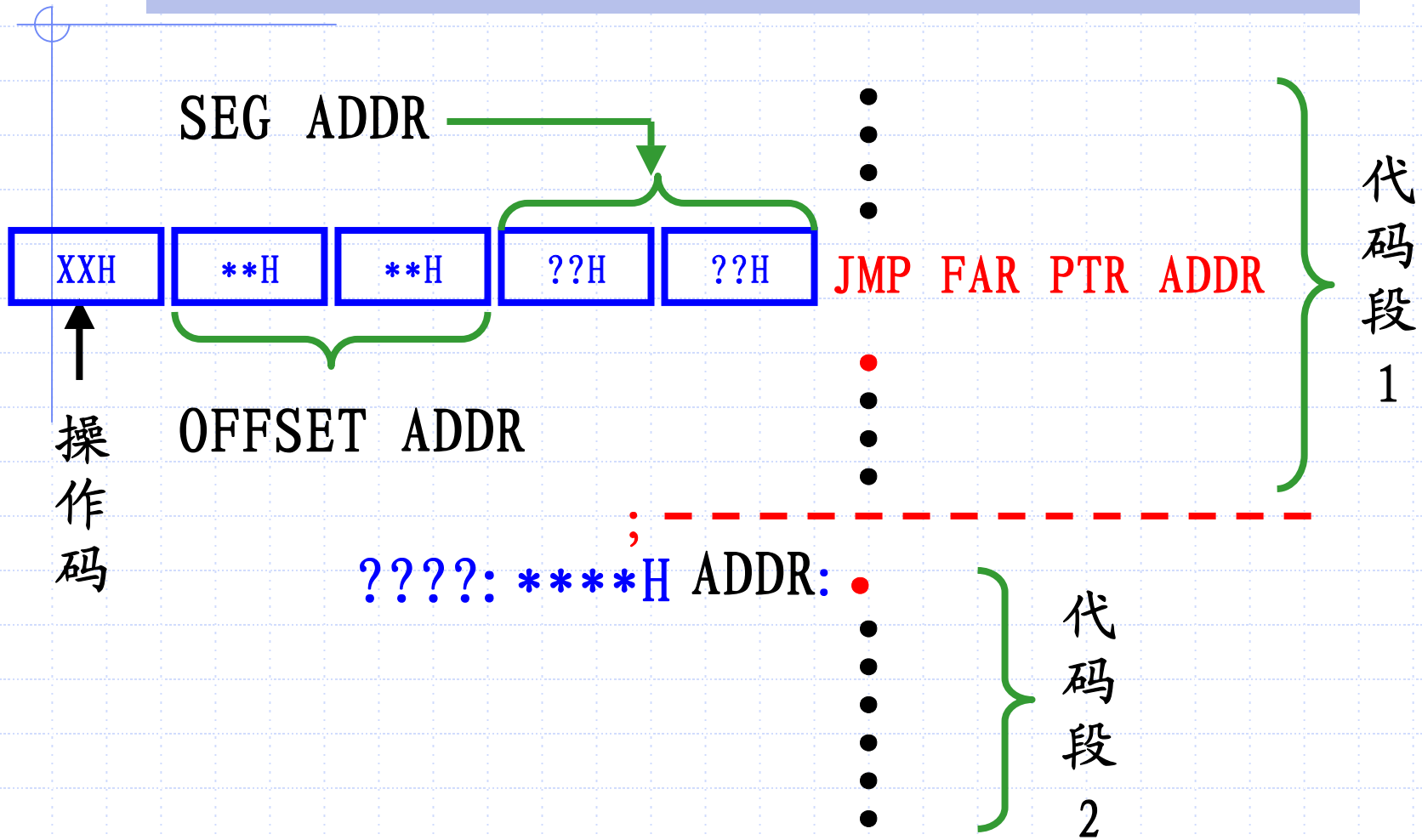
1. 段间直接寻址

指令语句书写时直接写出转移目的地的符号地址(FAR型标号名)。

书写格式为:

JMP FAR PTR 标号名

3.6 8086/8088的寻址方式



3.6 8086/8088的寻址方式

如：JMP FAR PTP ADDR

CS 4 0 0 0 H
+ IP 3 0 0 0 H

4 3 0 0 0 H

ADDR

操作码

OFFSET低字节

OFFSET高字节

SEG低字节

SEG高字节

⋮

OP CODE

存储器

00

30

00

40

IP

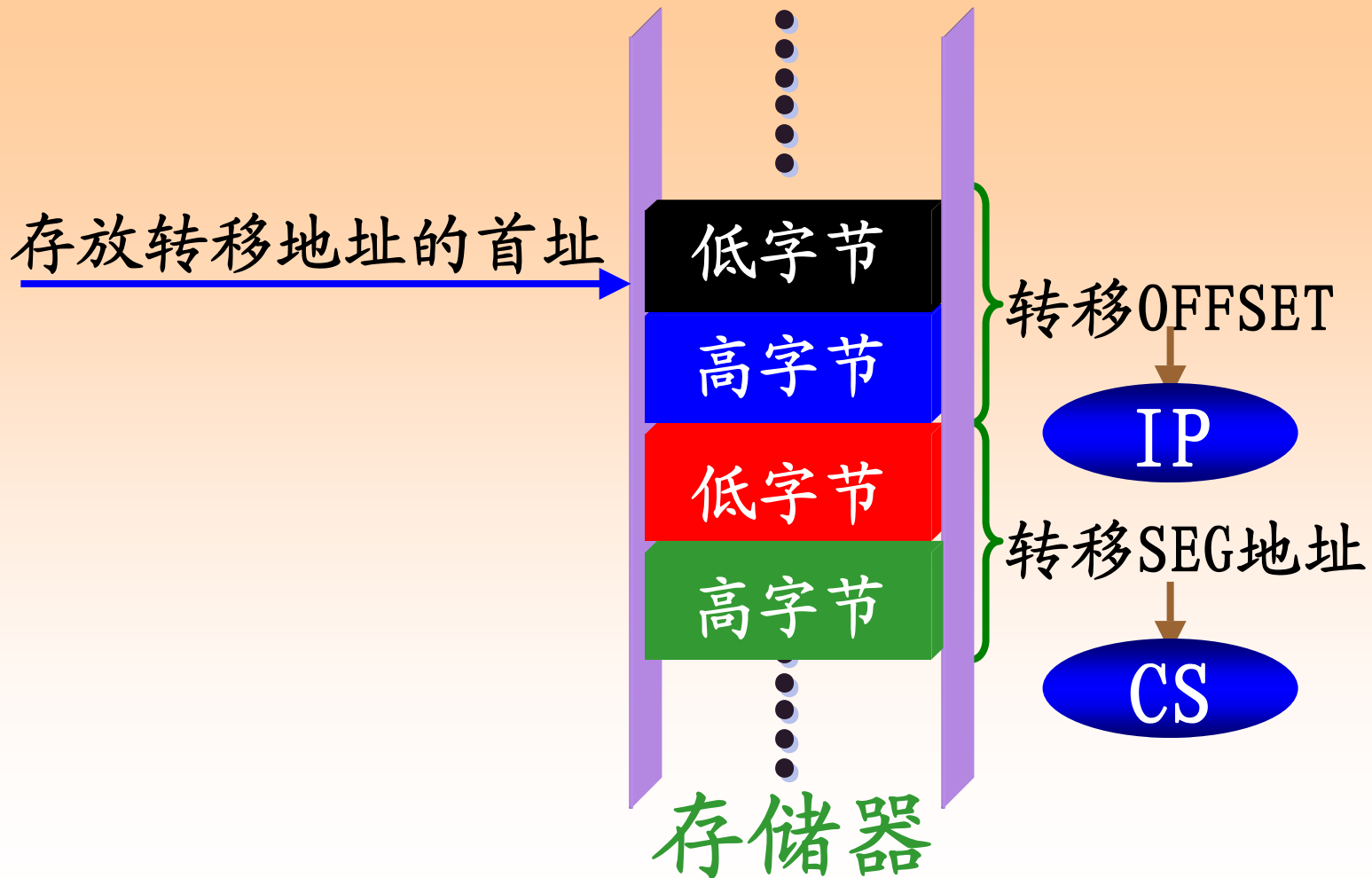
CS

3.6 8086/8088的寻址方式

2. 段间间接寻址

这种寻址方式和段内间接寻址方式相似，但不可能有寄存器间接寻址了，因为要得到的转移地址是32bit（16bit SEG地址和16bit EA地址）。因此转移目的地的地址只能间接的存放在一个双字的存储器单元当中。这个双字存储器单元可以用五种存储器寻址方式的任一种寻找到。找到的这个双字单元（连续的4个字节单元）存放的就是转移地址如下图所示：

3.6 8086/8088的寻址方式



3.6 8086/8088的寻址方式

如：JMP DWORD PTR [BX] [DI]

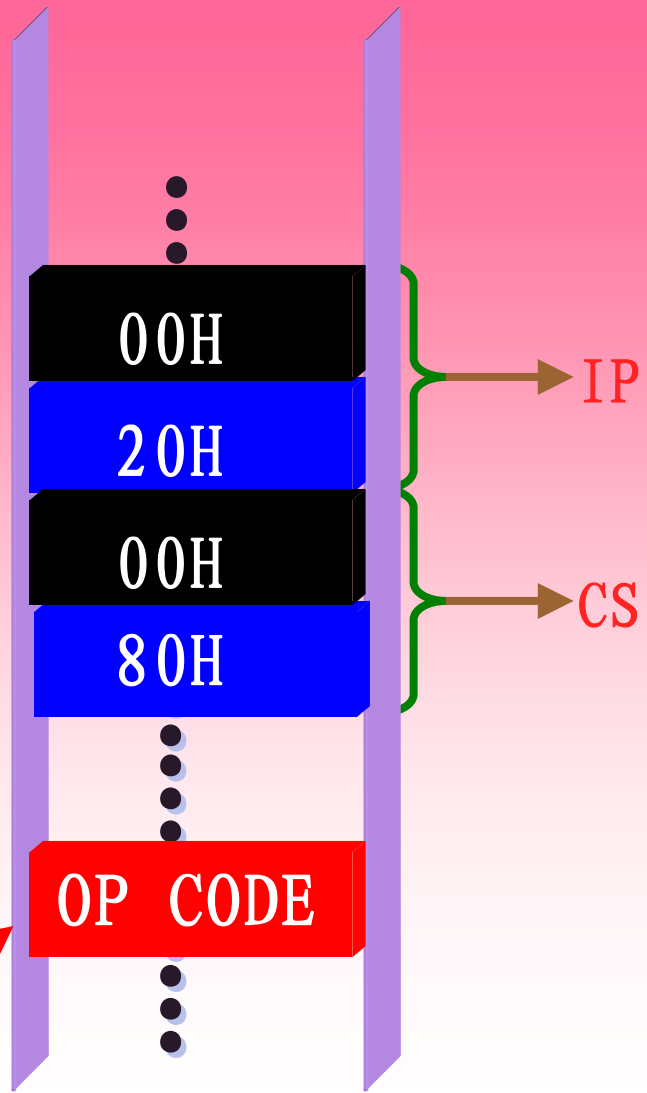
若 (DS)=3000H, (BX)=1000H,
(DI)=2000H, 则:

DS:	3	0	0	0
BX:	1	0	0	0
DI:+	2	0	0	0
<hr/>				
	3	3	0	0
				0 H

33000H

CS:	8	0	0	0
IP: +	2	0	0	0
<hr/>				
	8	2	0	0
				0 H

82000H



小结

8086
寻址方式

寻找数据

立即数寻址
寄存器寻址

存储器寻址

隐含寻址

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

寻找地址

段内

直接寻址 (相对寻址)

间接寻址

段间

直接寻址

间接寻址

小结

8086指令系统当中的所有条件转移指令只能在段内转移，且转移范围以当前IP为基准，不能超出-128~+127之间，其寻址方式为段内相对寻址。

作业：

12. 13.