哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

汇编语言程序设计

第2讲：80X86的寻址方式

裴文杰

第2讲：80X86的寻址方式

寻址的概念

与数据有关的寻址方式

- 立即寻址
- 寄存器寻址
- 直接寻址
- 寄存器间接寻址
- 寄存器相对寻址

与转移地址有关的寻址方式

- 基址变址寻址
- 相对基址变址寻址
- 段内直接寻址
- 段内间接寻址
- 段间直接寻址
- 段间间接寻址

wenjiecoder@outlook.com

寻址的概念

与数据有关的寻址方式

- 立即寻址
- 寄存器寻址
- 直接寻址
- 寄存器间接寻址
- 寄存器相对寻址

与转移地址有关的寻址方式

- 基址变址寻址
- 相对基址变址寻址
- 段内直接寻址
- 段内间接寻址
- 段间直接寻址
- 段间间接寻址

wenjiecoder@outlook.com

指令系统

计算机是通过执行指令序列来解决问题的，每条指令代表一种基本功能，这些基本功能是在程序运行期间由计算机硬件来实现的。

每种计算机都有对应于自己硬件的一组指令集，供用户使用，这组指令集就是这种计算机的指令系统。

本课程学习基于80x86 CPU的指令系统。

指令基本格式　　操作码　　操作数

操作码：计算机要执行的操作。

操作数：指明参与操作的数据或数据所在的地方。

汇编语言是一种用[illegible]采用指令系统的助记符来表示操作码和操作数，用符号地址表示操作数地址。相对于机器语言，因而易记、易读、易修改，给编程带来很大方便。

wenjiecoder@outlook.com

寻址的概念

汇编指令格式：

操作码 [操作数1 [,操作数2 [,操作数3]]] [;注

在使用两个操作数时，操作数1为目的操作数，操作数2为源操作数

寻址的目的：

每一个操作码都有对应的二进制代码，无需寻址。

操作数情况比较复杂：

- 可以直接在指令中指定，或者存放在寄存器或者内
- 操作数可以是单个数，或者是复杂的表格或者数组

如何在指令中用一套高效的机制来尽量以少的字段位数表示操作数的
从而在执行指令时根据地址找到操作数。

寻址、寻址方式的概念：

- 寻址就是寻找操作数的地址以便取得操作数。
- 寻址方式就是寻找操作数地址的方法。

例：result=a+b

```
data segment
  a       db  1
  b       db  2
  result  db  ?
  string  db  'result=$'
data ends
code segment
  assume cs:code, ds:data
start:  mov   ax,data
        mov   ds,ax
              mov   al,a
        add   al,b
        mov   result,al
        lea   dx,string
        mov   ah,09
        int   21h
        add   result,30h
        mov   dl,result
        mov   ah,2
        int   21h
              mov   ah,4ch
              int   21h
code    ends
        end   start
```

寻址的概念

操作数有三种来源：

① 操作数在指令中，称立即数操作数

如　MOV AL，9

② 操作数在寄存器中，称寄存器操作数

指令中给出用符号表示的寄存器名。

如　MOV AL，BL

③ 操作数在内存单元中，称存储器操作数或内存操作数

指令中给出该内存单元的地址。用[　]表示存储器操作数

如　MOV AL，[2000H]

寻址方式分类：

1)与数据有关的寻址方式：确定内存单元的地址

2)与转移地址有关的寻址方式：确定转移地址

wenjiecoder@outlook.com

与数据有关的寻址方式

- 立即寻址
- 寄存器寻址
- 直接寻址
- 寄存器间接寻址
- 寄存器相对寻址

与转移地址有关的寻址方式

- 基址变址寻址
- 相对基址变址寻址
- 段内直接寻址
- 段内间接寻址
- 段间直接寻址
- 段间间接寻址

wenjiecoder@outlook.com

与数据有关的寻址方式

以 MOV 指令为例：

- 立即寻址　　MOV AX , 3069H
- 寄存器寻址　　MOV AX , BX
- 直接寻址　　MOV AX , [2000H]
- 寄存器间接寻址　　MOV AX , [BX]
- 寄存器相对寻址　　MOV AX , COUNT [SI]
- 基址变址寻址　　MOV AX , [BP] [DI]
- 相对基址变址寻址　　MOV AX , MASK [BX] [SI]

wenjiecoder@outlook.com

与数据有关的寻址方式

1. 立即寻址方式

操作数是常数，直接存放在指令中，紧跟在操作码之后，作为指令的一部分，这种操作数称为立即数。

立即数可以是8位或16位(16位的立即数是高位字节放在高地址，低位字节放在低地址)。

应用场合：立即数常用来给寄存器或内存单元赋初值。

注意：只能用于源操作数字段，不能用于目的操作数字段。

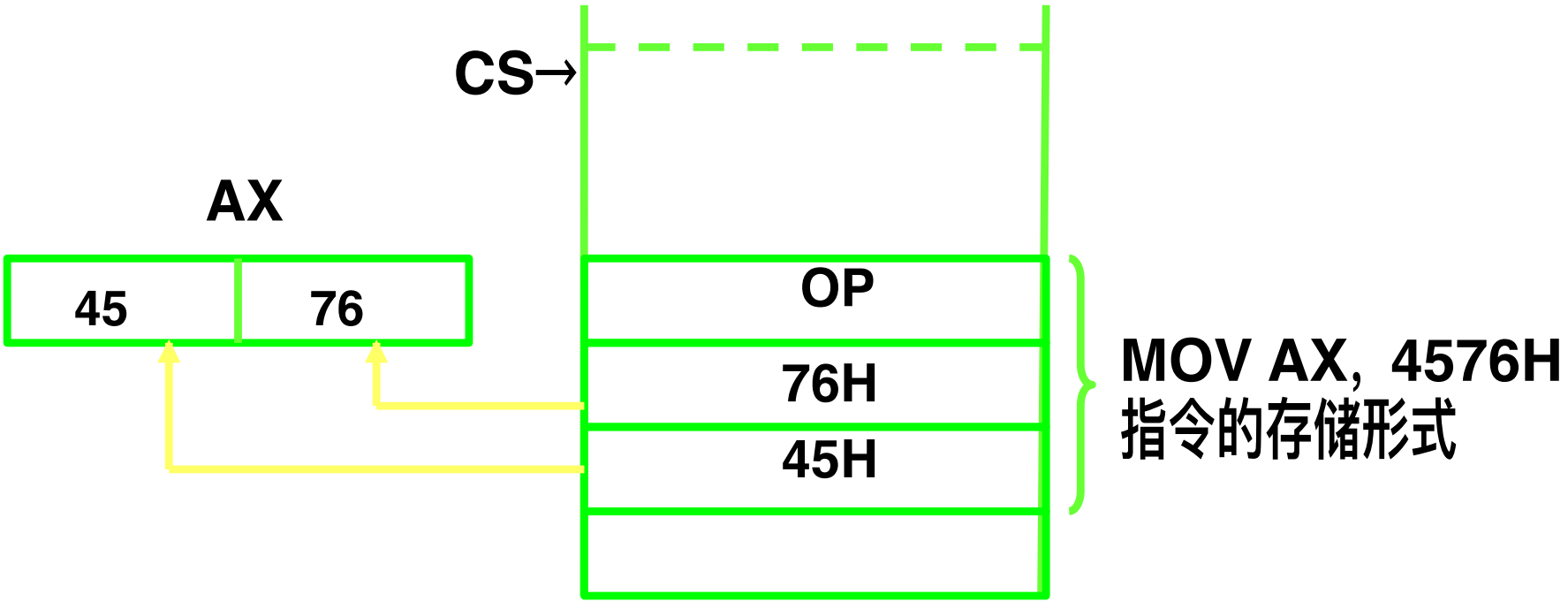
wenjiecoder@outlook.com

与数据有关的寻址方式

【例】 MOV AX，4576H 执行后（AX）=?

该例中源操作数为立即寻址方式，立即数为4576H，存放在指令的下一单元。

执行：4576H→AX 执行后：（AX）=4576H



例 MOV 78 H, AL (right?)

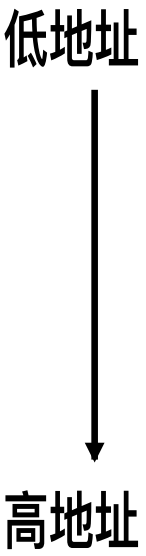
wenjiecoder@outlook.com

与数据有关的寻址方式

例　　MOV　AX , 2056H

结果　(AH) = 20H

(AL) = 56H



与数据有关的寻址方式

2.寄存器寻址方式

定义：指令所要的操作数已存储在某寄存器中，或把目标操作数存入寄存器。

指令中可以引用的寄存器如下：

8位寄存器：AH、AL、BH、BL、CH、CL、DH、DL等；

16位寄存器：AX、BX、CX、DX、SI、DI、SP、BP和段寄存器等。

注：由于指令所需的操作数已存储在寄存器中，或操作的结果存入寄存器，这样，在指令执行过程中，会减少读/写存储器单元的次数，所以，使用寄存器寻址方式的指令具有较快的执行速度。通常情况下，提倡在编写汇编语言程序时，应尽可能地使用寄存器寻址方式。

wenjiecoder@outlook.com

与数据有关的寻址方式

【例】下列程序执行后，（AX）=？，（BX）=？

```
MOV  AX, 1234H
MOV  BX, 5678H
ADD  AX, BX
```

执行：**1234H→AX**

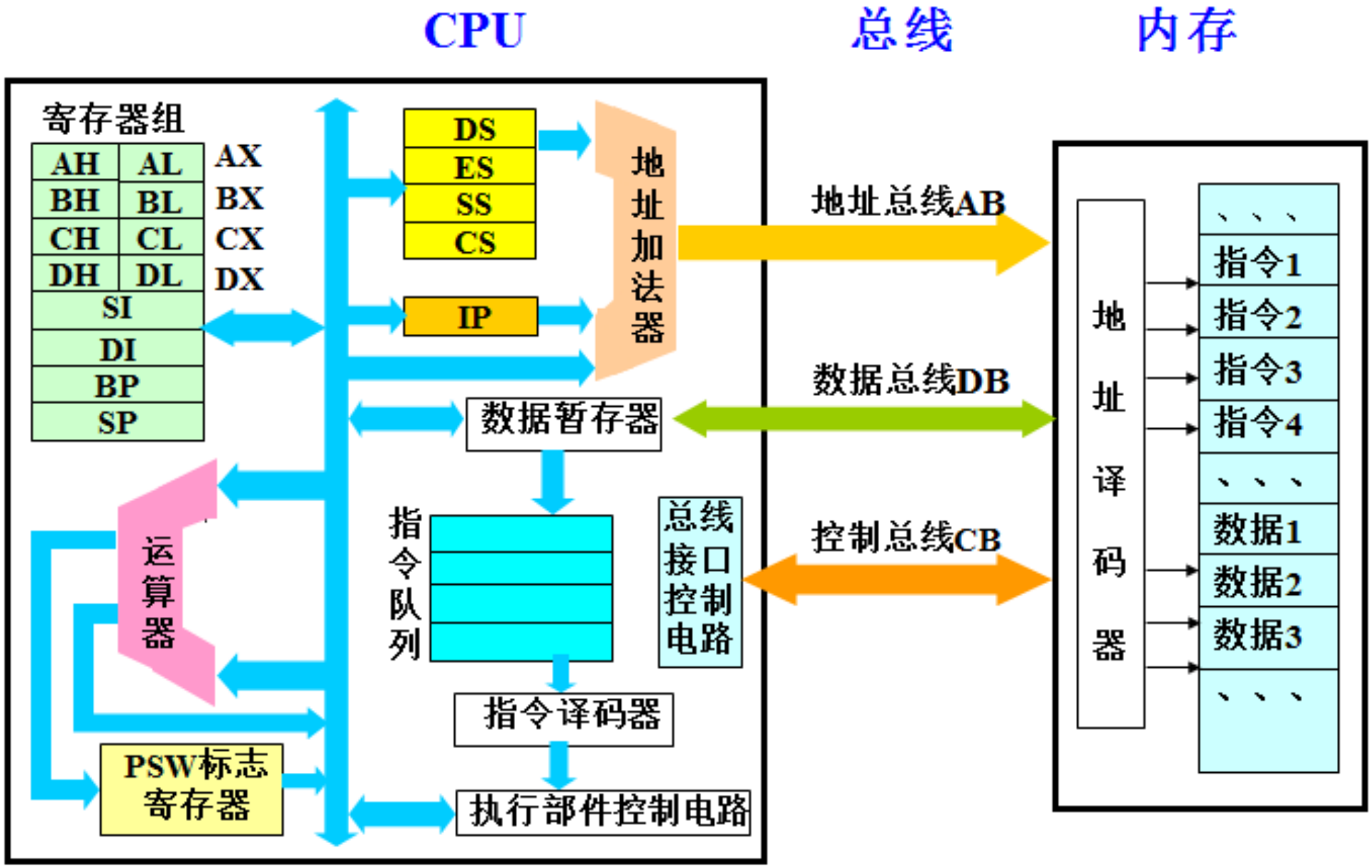
5678H→BX

“ADD AX, BX”：**(AX) + (BX) →AX**

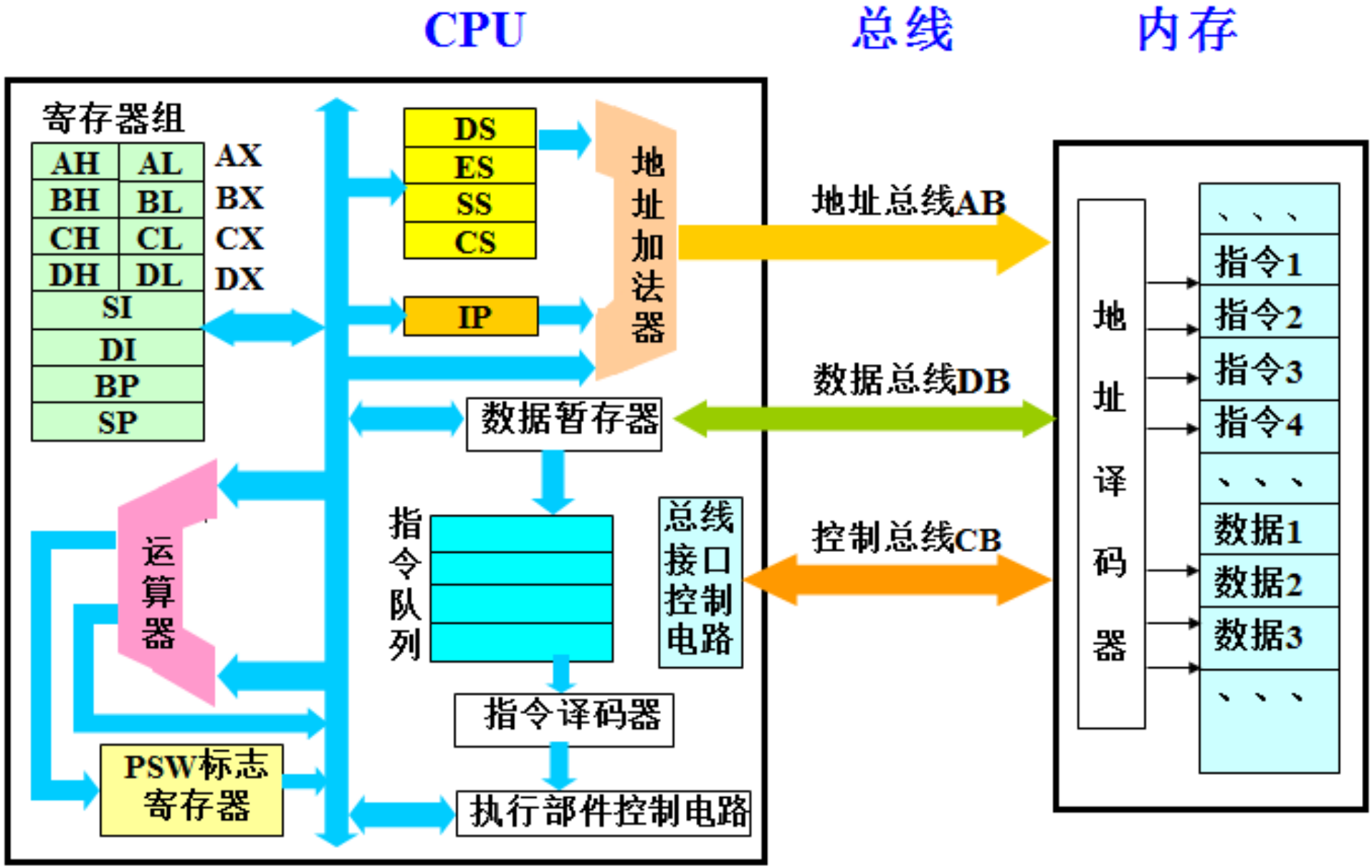
执行后：（AX）=68ACH，（BX）=5678H

▲ 以上立即数寻址、寄存器寻址的操作数，

不用在取完指令后再到内存中取数。



wenjiecoder@outlook.com



▲ 以下 5 种寻址方式：

操作数存放在内存中，取完指令后，还需到内存取数。

指令中给出的是该操作数的地址，包括段地址和偏移地址。

与数据有关的寻址方式

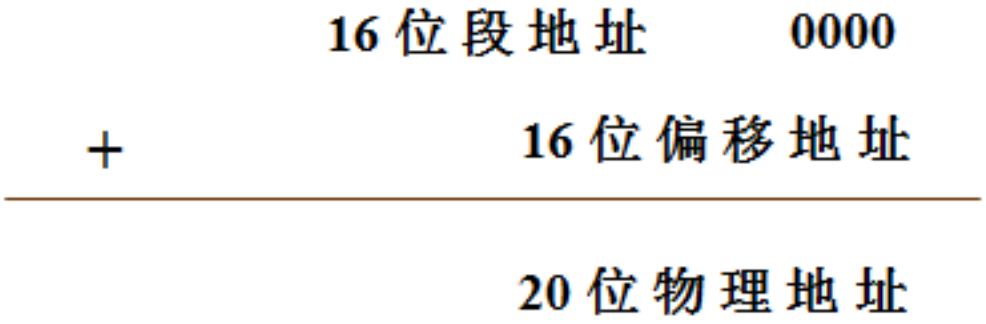
3. 存储器寻址

双操作数指令中只能有一个使用存储器寻址方式：

考虑指令长度，解码难度和硬件实现复杂性等因素，比如两个存储器之间传送，那么两个存储器都需要寻址，指令过长，解码过于复杂等。

除上述寻址方式外，以下各种寻址方式，操作数都在

在前面的章节中，我们知道操作数的物理地址（Phy Address，PA）等于段基址左移4位，再加上偏移地址：



段地址可以从段寄存器直接获得，所以下面主要解决的问题是如何得到操作数的偏移地址。在8086/8088里，将操作数的偏移地址又称为有效地址（effective address，即EA），因此以下的各种寻址方式即为求有效地址（EA）的不同途径。

wenjiecoder@outlook.com

与数据有关的寻址方式

如无特别指定段寄存器，指令中的寻址方式一般只显式指定有效地址（EA)，即偏移地址，而采用默认方式选择段寄存器来获得段地址：

- 若有效地址用SI、DI和BX等之一来指定或者直接提供有效地址，则其缺省的段寄存器为DS。寻址方式物理地址的计算方法如下：

$$PA=16\times DS + 有效（偏移）地址$$

- 若有效地址用BP来指定，则其缺省的段寄存器为SS（即：堆栈段）。物理地址：

$$PA=16\times SS + 有效（偏移）地址$$

wenjiecoder@outlook.com

与数据有关的寻址方式

8086/8088指令中有效地址（EA）由以下3部分组成：

EA=基址＋变址＋位移量

①位移量（displacement）：

存放在指令中的一个8位或16位的数，但它不是立即数，而是一个地址；

②基址（base）：

存放在基址寄存器（BX、BP）中的内容。它是有效地址中的基址部分，通常用于指向数据段中数组或字符串的首地址。

③变址（index）：

存放在变址寄存器（SI、DI）中的内容。通常用来指向数组中某个元素或字符串的某个字符。

根据有效地址中含有的成分不同，分别构成不同的寻址方式 。

三种成分	16位寻址	32位寻址
位移量	0、8、16位	0、8、32位
基址寄存器	BX、BP	任何32位通用寄存器
变址寄存器	SI、DI	除ESP外的32位通用寄存器

与数据有关的寻址方式

8086/8088指令中有效地址（EA）由以下3部分组成：

EA=基址＋变址＋位移量

用法举例：

①一维数组或者表格或者字符串：用基址或者位移量表示首地址，而用变址来表示数组，表格或者字符串中的某一个值（相对于首地址的偏移量）；

②二维数组：用位移量指向整个二维数组的首地址，用基址指向某一行的首地址，变址指向这一行的某一个元素。

三个地址并不一定同时需要，根据实际需求来指定。

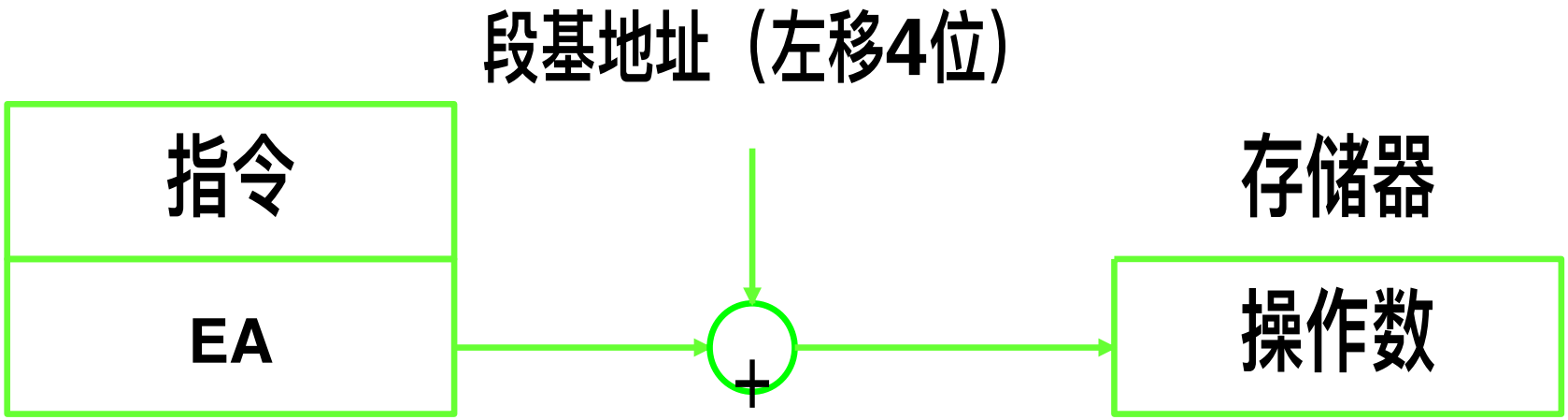
wenjiecoder@outlook.com

与数据有关的寻址方式

(1) 直接寻址

定义：指令所要的操作数存放在内存中，在指令中直接给出该操作数的有效地址，这种寻址方式为直接寻址方式。

图形表示：



- 存储器寻址关键：
计算有效地址，然后与段地址相加，得到物理地址

wenjiecoder@outlook.com

与数据有关的寻址方式

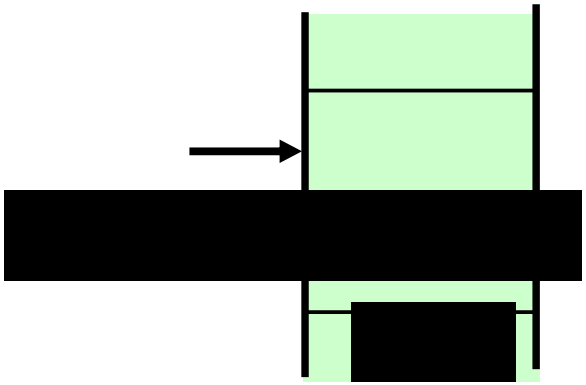
在汇编语言程序中，一般不直接用数值表示偏移地址，而用符号代替数值表示地址，称**符号地址**(变量名)。

例　符号buffer表示一个地址。

MOV AX ,[buffer]

或写成　　MOV AX ,buffer

源操作数为buffer指向的内存单元的内容



符号地址(变量名)经汇编连接后，与一个确定的数值地址相对应。可用操作符Offset 获取变量的偏移地址。

故 PA = (DS)×10H + Offset buffer

指令执行结果　(AX) = 0B0A H

注：

如没有特别指定，默认选择在DS段中寻址。

显式指定段跨越前缀：

MOV AX, ES:[buffer]

wenjiecoder@outlook.com

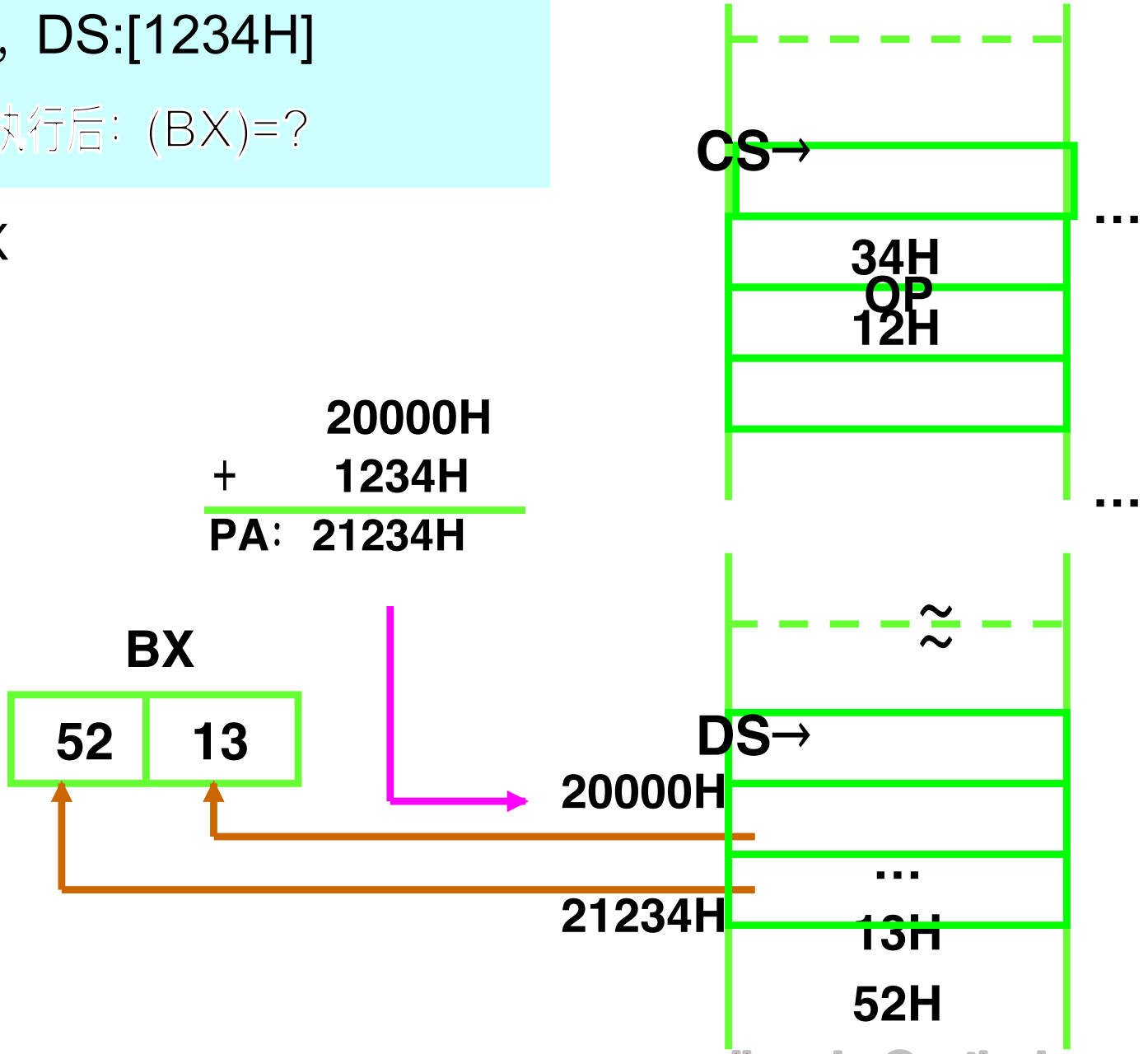
与数据有关的寻址方式

例：执行指令MOV BX，DS:[1234H]

设（DS）=2000H。执行后：(BX)=?

执行：（21234H）→BX

执行后：（BX）=5213H



与数据有关的寻址方式

例：MOV AX , DS:[1000 H]

若 (DS) = 2000H

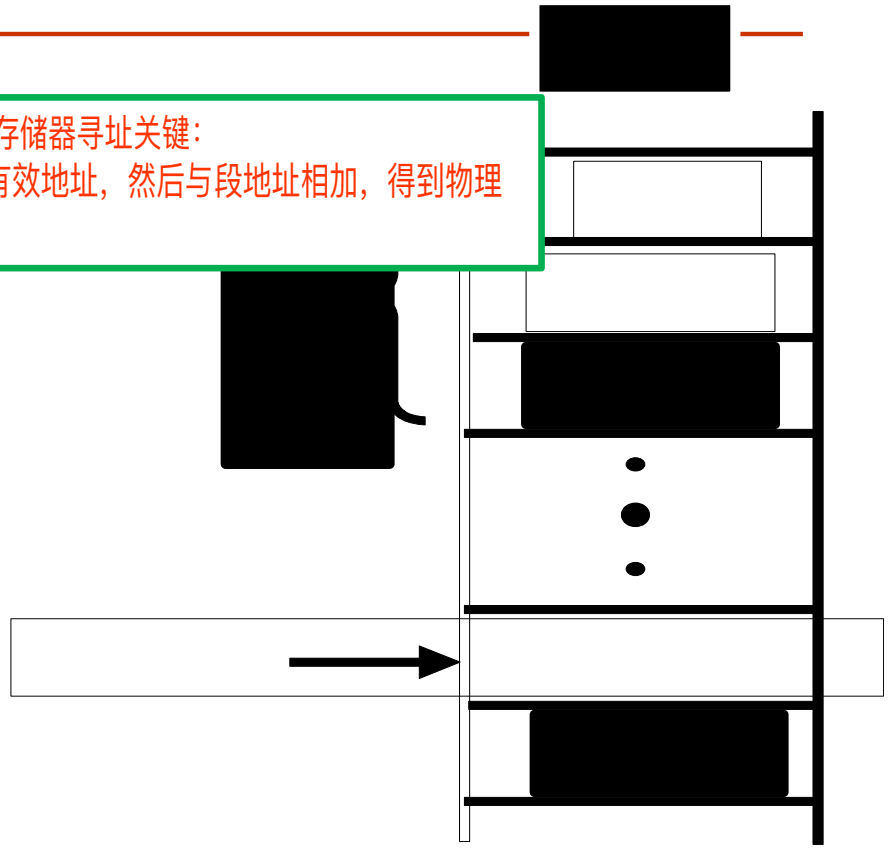
内存操作数的物理地址为：

PA = (DS)×10H +EA

= 2000H × 10H + 1000H

= 21000H

执行后 (AX)= 3040H



指令 MOV AX, [1000H] 与MOV AX, 1000H有什么不同?

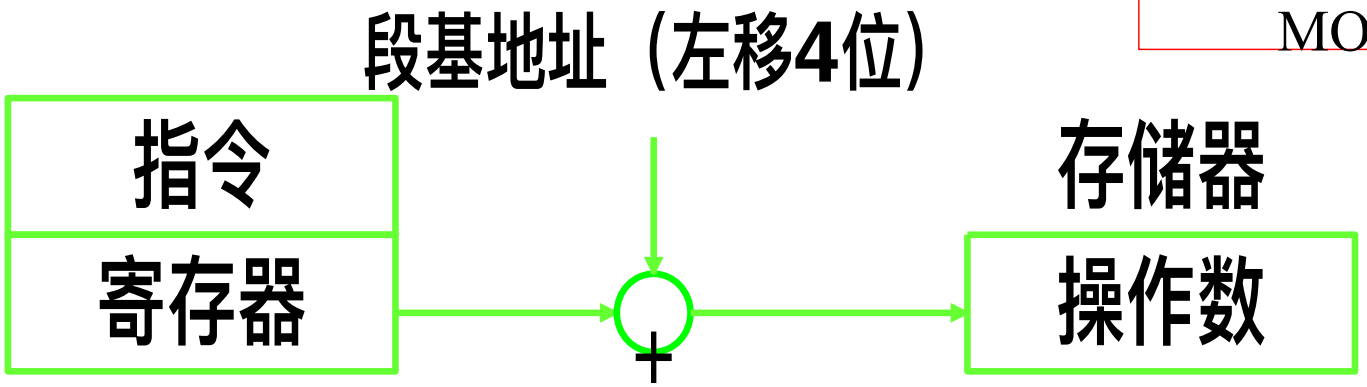
与数据有关的寻址方式

(2) 寄存器间接寻址

定义：操作数的有效地址存储在基址寄存器或变址寄存器当中。

可用的寄存器有 BX 、BP、SI、DI

如：
```
MOV   AL, [BX]
MOV   DH, [BP]
MOV   AH, [SI]
MOV   DL, [DI]
```



wenjiecoder@outlook.com

与数据有关的寻址方式

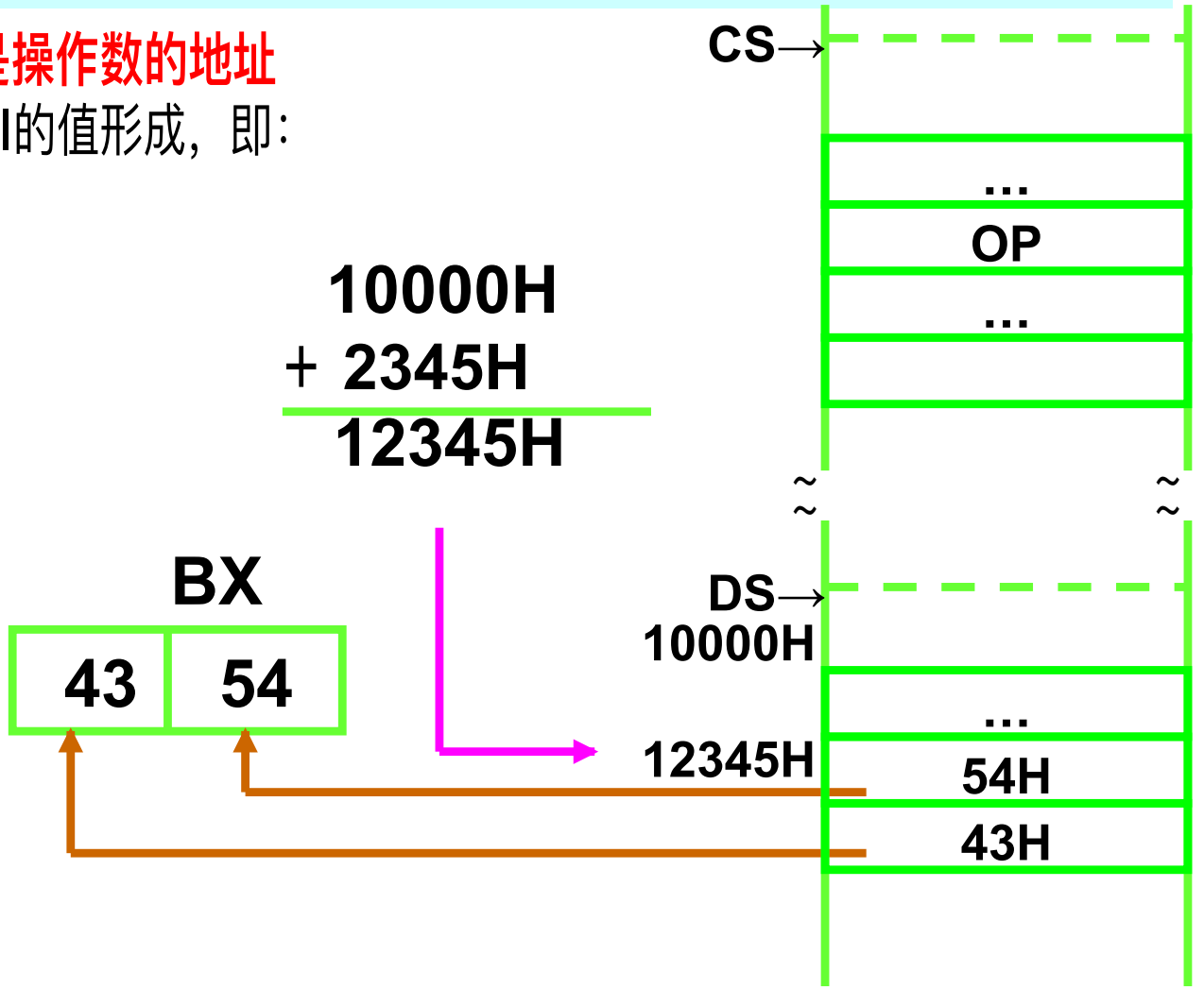
【例3.4】假设有指令：MOV　BX,[DI]，在执行时，（DS）=1000H，（DI）=2345H，存储单元（12345H）=4354H。问执行指令后，BX的值是什么?

解：**寄存器DI的值不是操作数，而是操作数的地址**。该操作数的物理地址应由DS和DI的值形成，即：

PA=（DS）*16+DI

=1000H*16+2345H

=12345H。



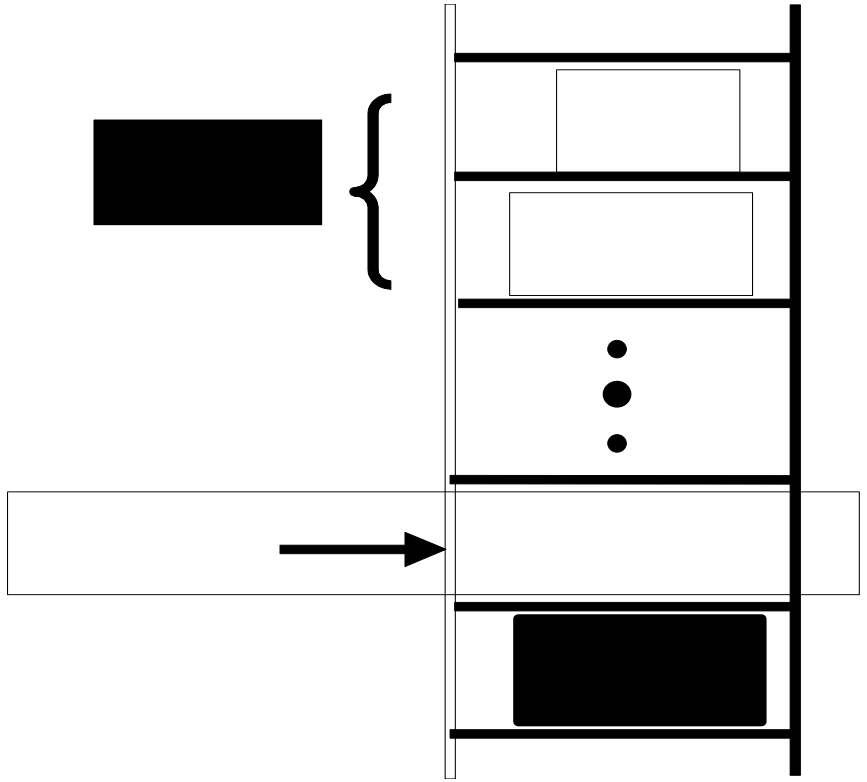
该指令的执行效果是：把从物理地址为12345H开始的一个字的值传送给BX。

与数据有关的寻址方式

例: MOV AX , [DI]

若 (DS) = 3000H

(DI) = 2000H



则内存操作数的物理地址为:

PA = (DS)×10H + (DI)

= 32000H

执行后 (AX) = (32000H) = 400BH

思考: 指令 MOV AX,[DI] 与指令MOV AX, DI有什么不同?

wenjiecoder@outlook.com

与数据有关的寻址方式

(3) 寄存器相对寻址方式

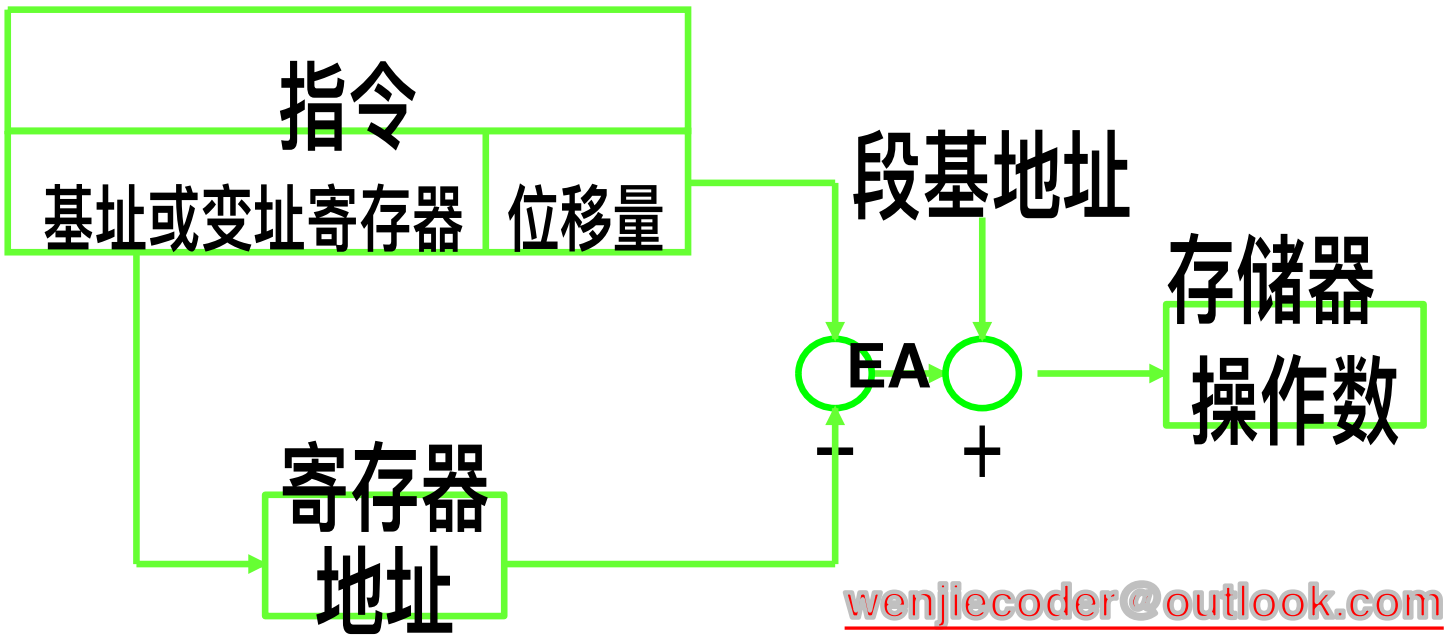
定义：操作数在存储器中，其有效地址是一个基址寄存器（BX、BP）或变址寄存器（SI、DI）的内容和指令中的8位/16位偏移量之和。

使用BP时，其默认段是SS段，其他寄存器默认为DS段。

汇编格式：X[R]或[R+X]（X表示位移量，是8位或16位有符号数）

功能：操作数存放在存储器，寄存器R的内容加位移量X为操作数的偏移地址EA。

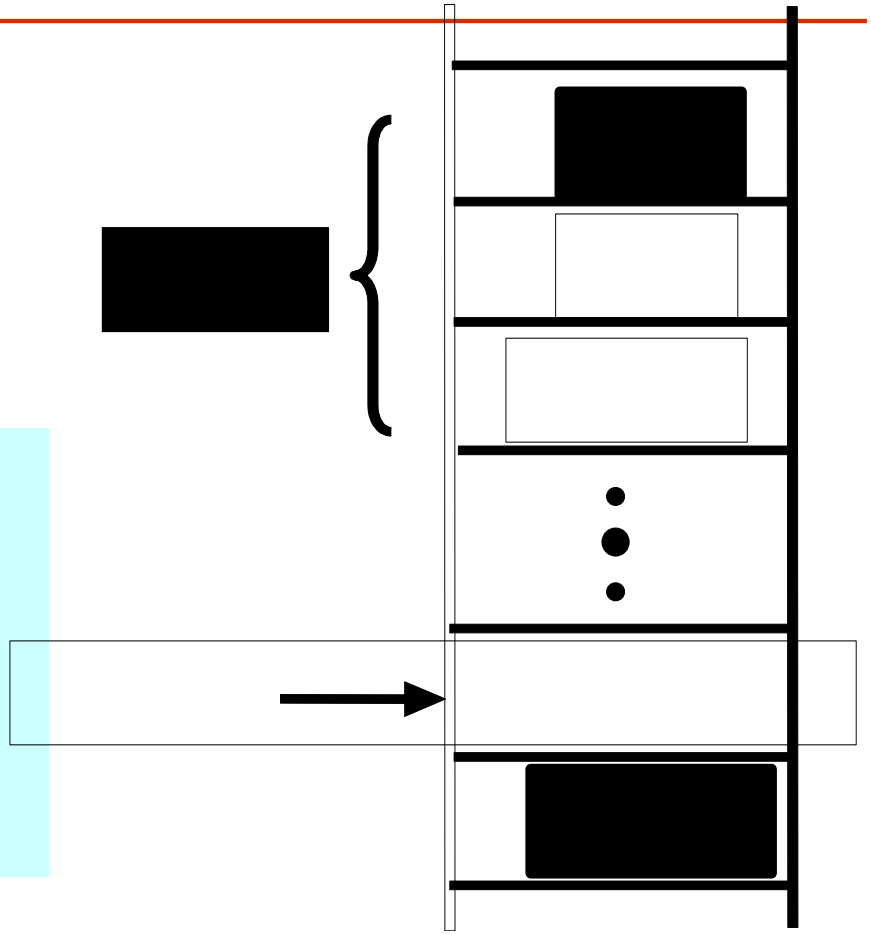
一维数组或者表格或者字符串：用**位移量**表示首地址，而用**基址**或者**变址**来表示数组，表格或者字符串中的某一个值



wenjiecoder@outlook.com

与数据有关的寻址方式

如:
MOV AL, [BX +10H]
MOV AH, [DI+20H]
MOV DL, 30H [SI]
MOV DH, 40H [BP]



【例】 MOV AX , 60H [BP]

若 (DS) =2000H, (SS) = 3000H

(BP) = 200H

则内存操作数的物理地址为?

PA = (SS)×10H + (BP) + 60H

= 30260H

指令执行后:

(AX) = (30260H) = 0ABCH

注:

因为有效地址用BP指定，因此按照默认方式选择SS作为段寄存器。

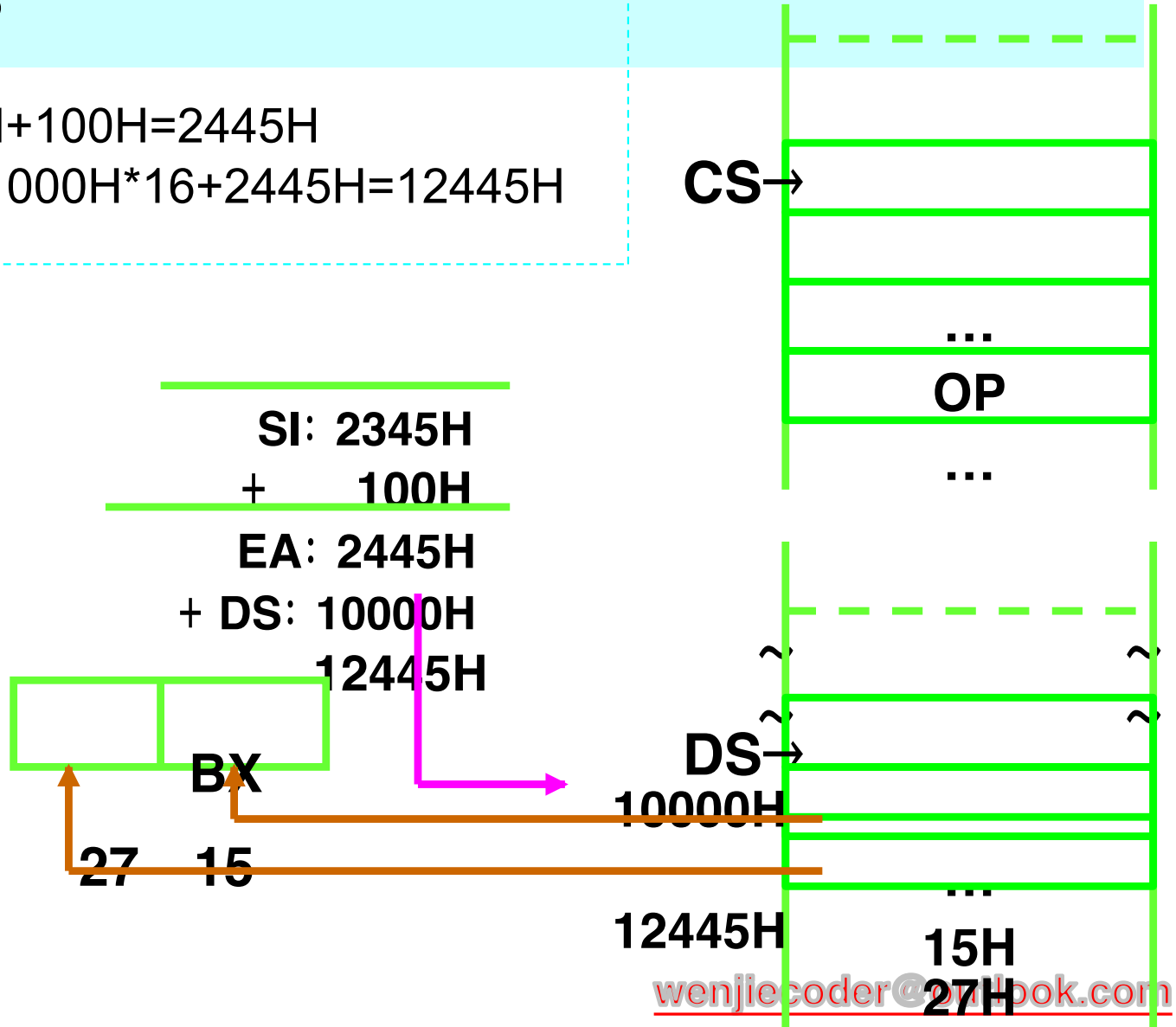
与数据有关的寻址方式

【例】假设指令：MOV BX, [SI+100H]，在执行它时，（DS）=1000H，（SI）=2345H，内存单元12445H的内容为2715H，问该指令执行后，BX的值是什么?

解：EA=（SI）+100H=2345H+100H=2445H

PA=（DS）*16+EA=1000H*16+2445H=12445H

所以，该指令的执行效果是：把从物理地址为12445H开始的一个字的值传送给BX。



(4) 基址变址寻址方式

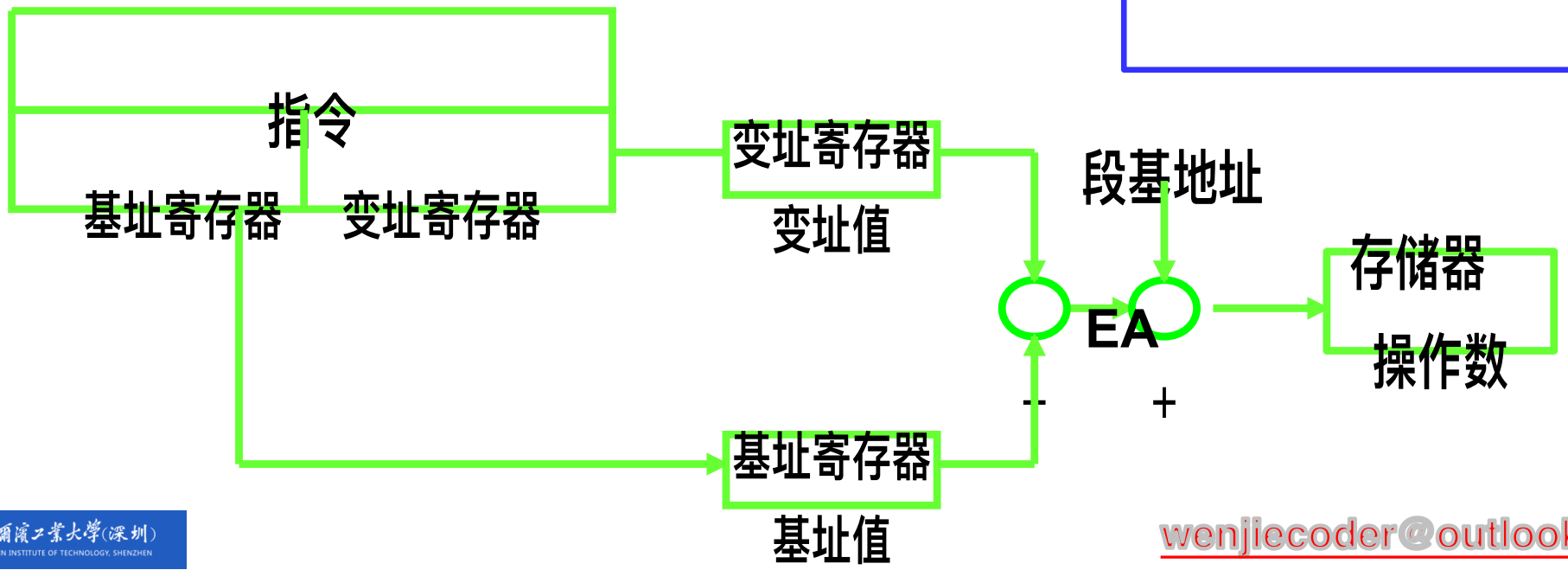
定义：操作数在存储器中，其有效地址是一个基址寄存器（BX、BP）和一个变址寄存器（SI、DI）的内容之和。

汇编格式： [BR+IR]

功能：操作数存放在存储器，BR的内容加IR的内容是操作数的偏移地址EA。

图形表示：

一维数组或者表格或者字符串：
用**基址**表示首地址，而用**变址**来表示数组，表格或者字符串中的某一个值



与数据有关的寻址方式

【例】假设指令：MOV BX, [BX+SI]，在执行时，（DS）=1000H，（BX）=2100H，（SI）=0011H，内存单元12111H的内容为1234H。问该指令执行后，BX的值是什么?

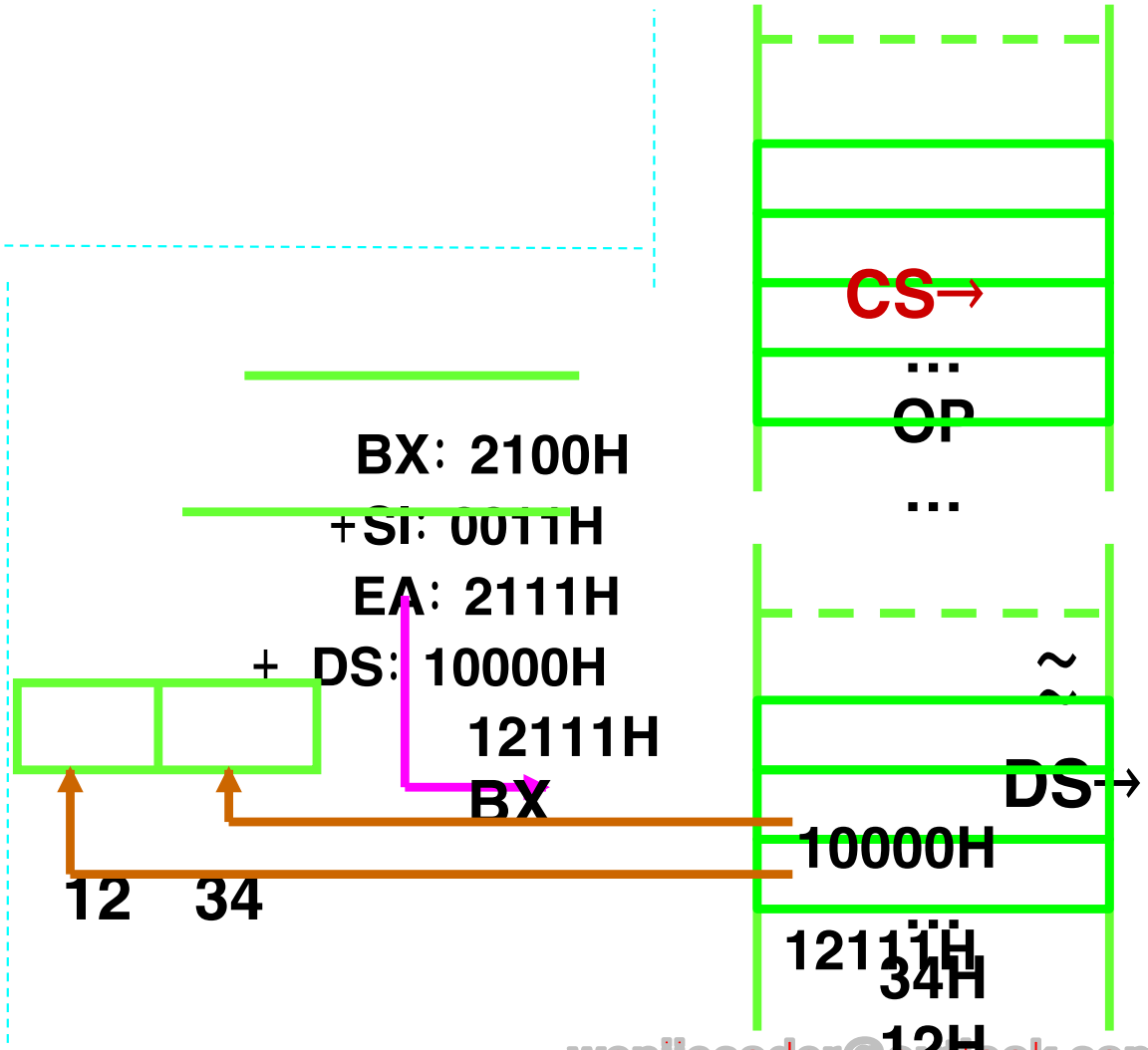
解：操作数的物理地址PA为：

PA=(DS)*16+(BX)+(SI)

=1000H*16+2100H+0011H

=12111H

所以，该指令的执行效果是：把从物理地址为12111H开始的一个字的值传送给BX。



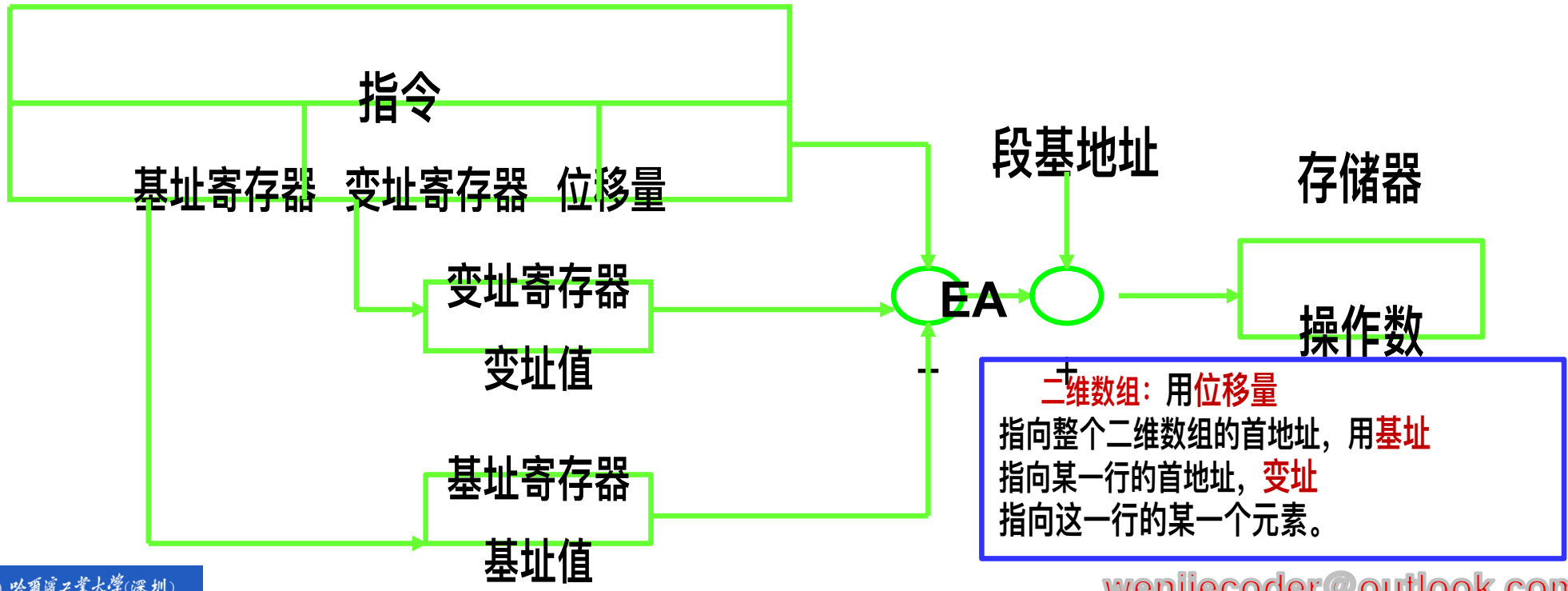
与数据有关的寻址方式

(5) 相对基址变址寻址方式

定义：操作数在存储器中，其有效地址是一个基址寄存器（BX、BP）的值、一个变址寄存器（SI、DI）的值和指令中的8位/16位位移量之和。

汇编格式：X [BR+IR]或[BR+IR+X]

功能：操作数存放在存储器，BR内容加IR内容加位移量X是操作数的偏移地址EA。



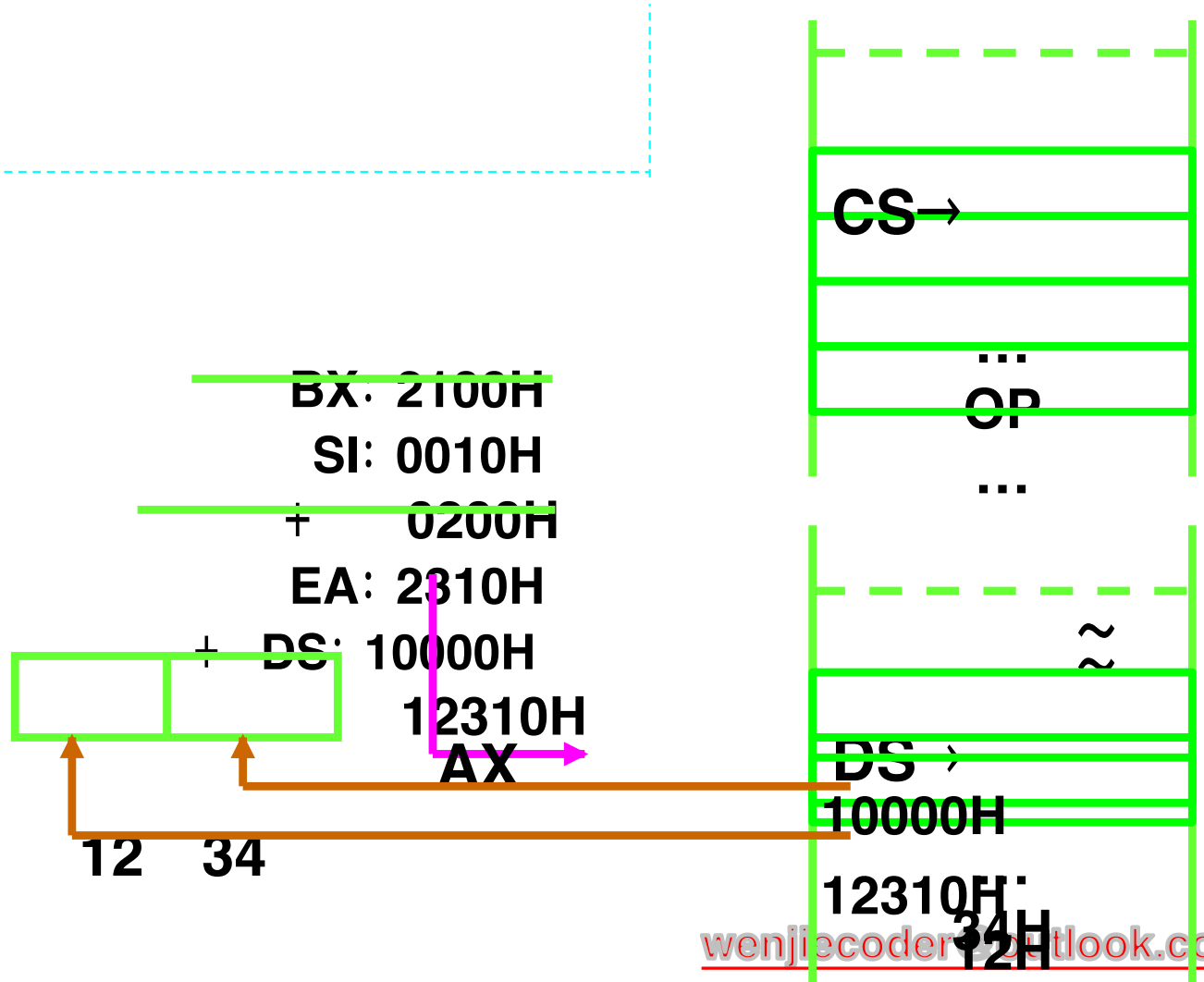
wenjiecoder@outlook.com

与数据有关的寻址方式

【例】假设指令：MOV AX, [BX+SI+200H]，在执行时，（DS）=1000H，（BX）=2100H，（SI）=0010H，内存单元12310H的内容为1234H。问该指令执行后，AX的值是什么？

解：该操作数的物理地址应由DS和EA的值形成，即：
PA=12310H

所以，该指令的执行效果是：把从物理地址为12310H开始的一个字的值传送给AX。



与数据有关的寻址方式

- 综上，按给出偏移地址方式的不同，分为以下5种：

直接寻址	MOV AL,[buffer]
寄存器间接寻址	MOV AL,[BX]
寄存器相对寻址	MOV AL,[BX + 10H]
基址变址寻址	MOV AL,[BX + SI]
相对基址变址寻址	MOV AL, [BX + SI + 10H]

其中：

寄存器相对寻址、基址变址寻址可用于表格或数组。

相对基址变址可用于二维数组。

wenjiecoder@outlook.com

与数据有关的寻址方式

- **段寄存器默认规则总结：**
-

若有效地址用SI、DI和BX等之一来指定或者直接提供有效地址，则其缺省的段寄存器为DS。寻址方式物理地址的计算方法如下：

$$PA=16\times DS+\begin{cases}(BX)\\(SI)\\(DI)\\\text{直接提供的有效地址}\end{cases}$$

-

若有效地址用BP来指定，则其缺省的段寄存器为SS（即：堆栈段）。物理地址：

$$PA=16\times SS+(BP)$$

wenjiecoder@outlook.com

与数据有关的寻址方式

例：MOV AL, [BX]

PA= (DS) ×10H + (BX)

默认选择DS寄存器的内容为段地址。

MOV AX, [BP]

PA= (SS) ×10H + (BP)

默认选择SS寄存器的内容为段地址。

wenjiecoder@outlook.com

与数据有关的寻址方式

❖ 段跨越问题

当要否定默认状态，到非约定段寻找操作数时，必须用跨段前缀指明操作数的段寄存器名。

汇编格式：段寄存器名：操作数地址。

功能：冒号“：”之前的段寄存器名指明操作数所在的段。

【例】 MOV AX, DS:[BP]

MOV CX, SS :[SI]

该例中“DS:”，“SS:”均为跨段前缀，此时默认状态无效，操作数的物理地址PA由段寄存器内容左移4位加偏移EA形成。上述2条指令的源操作数物理地址分别为：

PA1 = (DS) 左移4位+(BP)

PA2 = (SS) 左移4位+ (SI)

wenjiecoder@outlook.com

与数据有关的寻址方式

在某些情况下，8086/8088允许程序员使用跨越段前缀来改变系统指定的默认段。但以下3种情况不允许使用跨越段前缀。

①指令必须在代码段CS中；

②PUSH指令的源操作数和POP指令的目的操作数必须使用SS段；

③串处理指令的目的串必须使用附加段ES。

访存类型	所用段及寄存器	缺省选择规则
指令	代码段：CS	用于取指令
堆栈	堆栈段：SS	进栈或出栈，用SP、BP作为基址寄存器的访存
局部数据	数据段：DS	除堆栈或串处理操作以外的所有数据访问
目的串	附加段：ES	串处理指令的目的串

wenjiecoder@outlook.com

与数据有关的寻址方式

注意：1）不能自创寻址方式

寄存器操作数地址只能由BX、BP、SI、DI 给出，它们的组合也不是任意的。

寄存器间接
- [SI]
- [DI]
- [BX]
- [BP]

寄存器相对
- [SI + X]
- [DI + X]
- [BX + X]
- [BP + X]

基址加变址
- [BX+SI]
- [BX+DI]
- [BP+SI]
- [BP+DI]

相对基址加变址
- [BX+SI+X]
- [BX+DI+X]
- [BP+SI+X]
- [BP+DI+X]

其中X为8位或16位偏移量

wenjiecoder@outlook.com

常见错误：

如　MOV　CL, [AX]

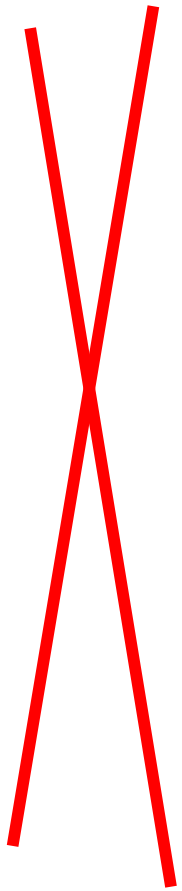
MOV　AX, [DX]

MOV　AL, [CX]

MOV　CX, [BP+BX]

MOV　AH, [SI+DI]

MOV　BL, [AX+CX]



与数据有关的寻址方式

2）只有存储器操作数需用段跨越的前缀

MOV ES:[SI], CX 

MOV AL, DS:[BP] 

MOV ES:AX, 0 

MOV DS:CX, 22H 

wenjiecoder@outlook.com

寻址的概念

与数据有关的寻址方式

- 立即寻址
- 寄存器寻址
- 直接寻址
- 寄存器间接寻址
- 寄存器相对寻址
- 基址变址寻址
- 相对基址变址寻址

与转移地址有关的寻址方式

- 段内直接寻址
- 段内间接寻址
- 段间直接寻址
- 段间间接寻址

wenjiecoder@outlook.com

与转移地址有关的寻址方式

用来确定转移指令（条件转移指令或无条件转移指令）及call指令的转向地址。

转移地址是由各种寻址方式得到的有效地址和段地址相加而成的，有效地址存入IP寄存器中，段地址指定为CS段寄存器内容。

- 段内寻址：**段内转移只需改变（IP）**

  段内直接寻址　JMP　NEAR PTR　NEXT

  段内间接寻址　JMP　TABLE [BX]

- 段间寻址：**段间转移改变（CS）、（IP）**

  段间直接寻址　JMP　FAR PTR　NEXT

  段间间接寻址　JMP　DWORD PTR [BX]

wenjiecoder@outlook.com

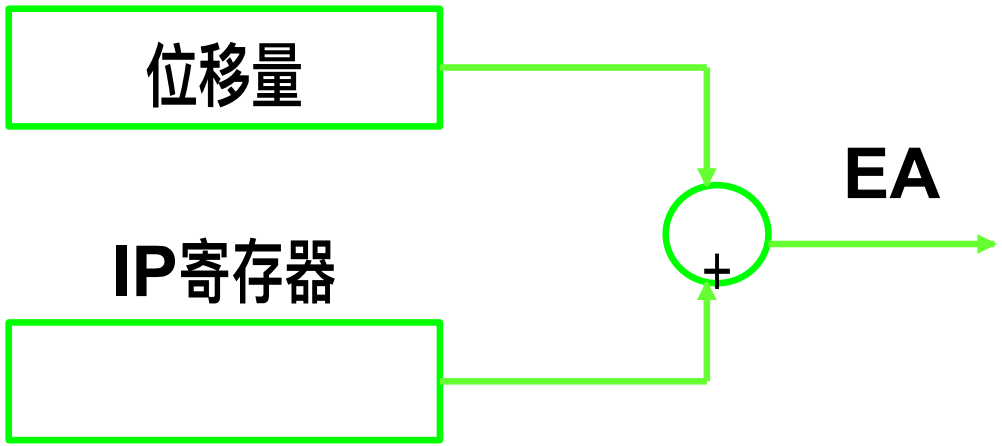
与转移地址有关的寻址方式

1. 段内转移

段内转移是指程序在同一代码段内，仅改变IP寄存器的值，而不改变CS寄存器的值所发生的转移。段内转移又分为段内直接转移和段内间接转移。

(1) 段内直接寻址

转向的有效地址 = 当前（IP） + 位移量（8bit/16bit)



wenjiecoder@outlook.com

与转移地址有关的寻址方式

位移量：位移量 = EA -（IP）当前

- 对于短转：8位位移量可正可负，范围是 −128～+127，前面加“SHORT”；
- 对于近转：16位位移量可正可负，范围是−32768～+32767，前面加“NEAR PTR ”。

这种寻址方式适用于条件转移及无条转移指令。

- 当用于条件转移指令时，位移量只允许 8 位；
- 用于无条件转移指令时，位移量 8 位时称为 短跳转，16位时则称为近跳转。

```
JMP SHORT QUEST          ；8位短跳转
JMP   NEAR PTR NEXT      ；16位近跳转
```

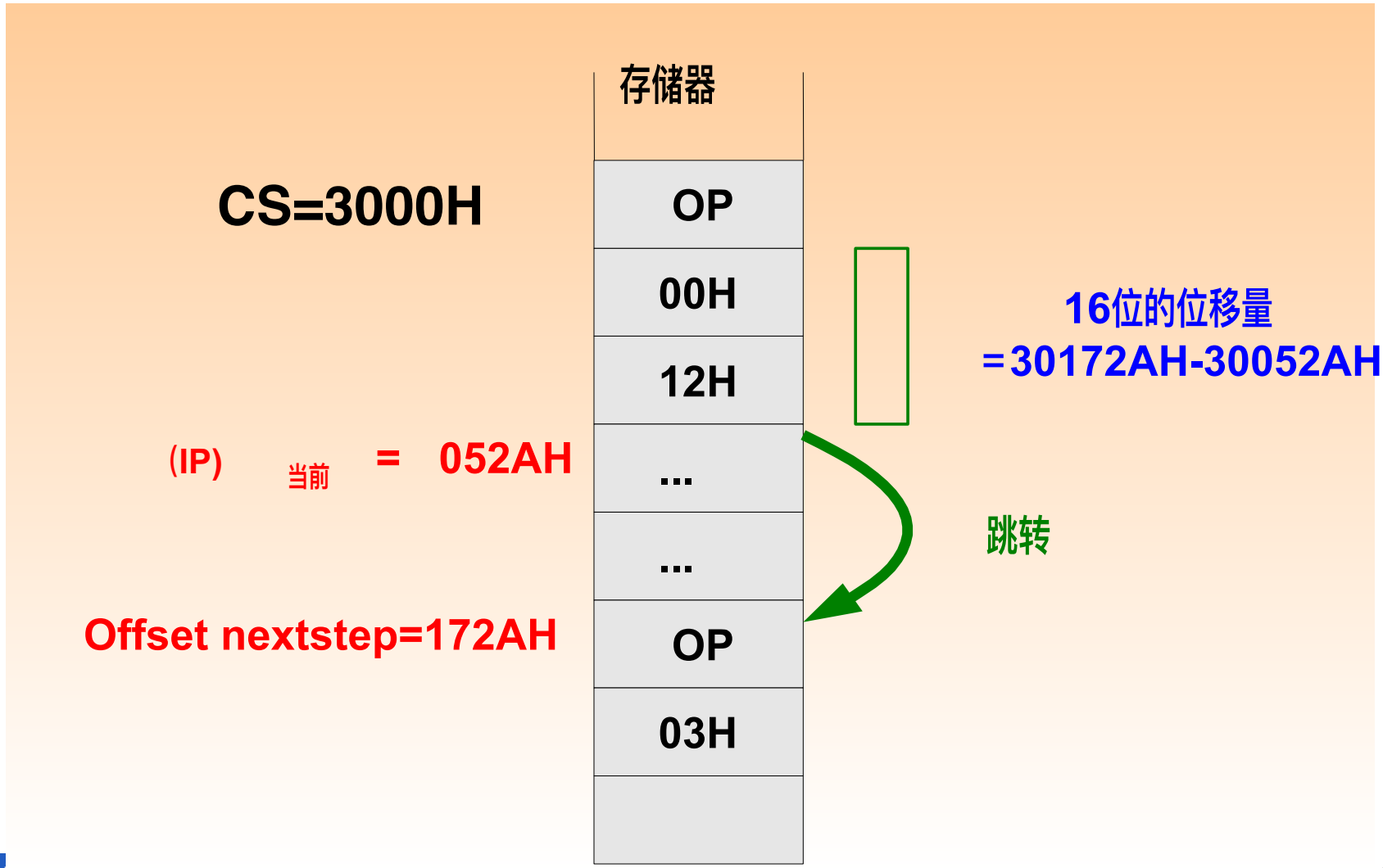
其中NEXT、QUEST均为转向的符号地址，机器中用位移量表示。

wenjiecoder@outlook.com

与转移地址有关的寻址方式

段内直接寻址：JMP near ptr nextstep

(IP) ← (IP) 当前+16位位移量



与转移地址有关的寻址方式

在段内直接寻址中：
汇编语言中操作数为转向的目标符号地址，
但是在机器指令中，操作数为位移量。

```
                1060:000D EB04 JMP SHORT NEXT
IP当前值→ 1060:000F … …
                1060:0011 … …
                1060:0013 0207 NEXT: ADD AL,[BX]
```

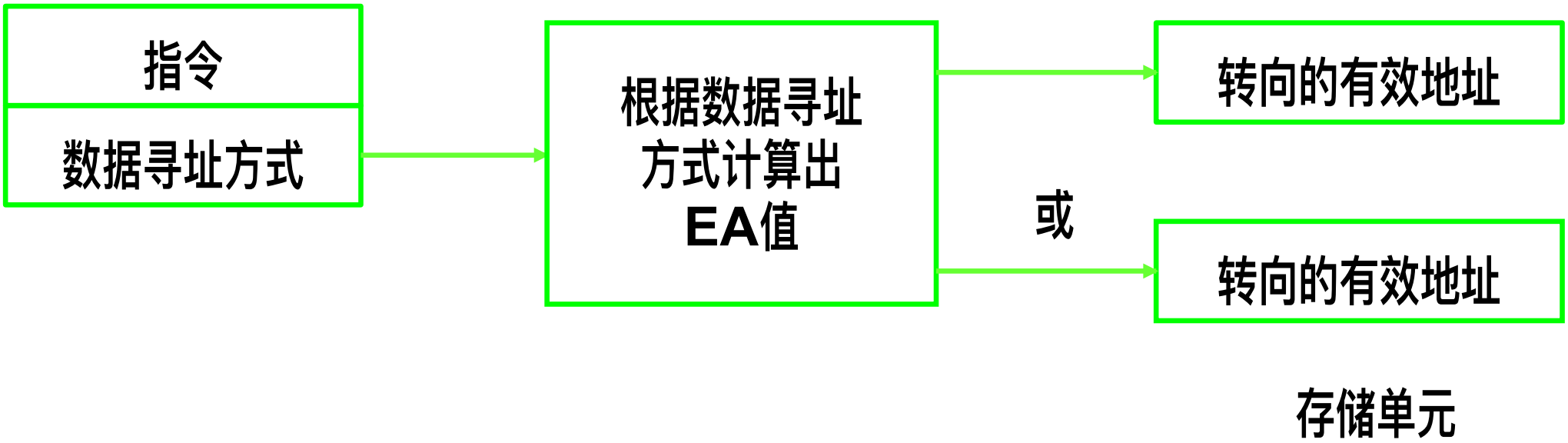
EB04 为机器码
EB为操作码，**04**为位移量，所以转向的有效地址是
000F + 0004 = 0013

wenjiecoder@outlook.com

与转移地址有关的寻址方式

(2) 段内间接寻址

转向的有效地址是一个寄存器或存储单元的内容。可用除立即数以外的任何一种数据寻址方式得到，所得到的转向的有效地址取代IP寄存器的内容。



wenjiecoder@outlook.com

与转移地址有关的寻址方式

汇编格式：JMP　　BX

JMP word ptr [BP + table]

其中：　word ptr ——属性定义符

(BP) + table 寻址所得地址是一个字的有效地址，

由有效地址形成物理地址里的内容即为转向的有效地址（IP）。

转向物理地址的计算公式：PA=10H□（CS）+(IP)

wenjiecoder@outlook.com

与转移地址有关的寻址方式

【例】
已知 TABLE=20A1H，（BX）=1256H，（SI）=528EH，
（DS）=2000H，（232F7H）= 3280H，（264E5H）=2450H

```
JMP  BX                          ; （IP）=1256H

JMP  WORD PTR  TABLE[BX]         ; （IP）=3280H

JMP  WORD PTR  [BX][SI]          ; （IP）=2450H
```

注：如果指令操作数已被定义为16位的存储器，则：

word ptr 可以省去。

wenjiecoder@outlook.com

与转移地址有关的寻址方式

2. 段间转移

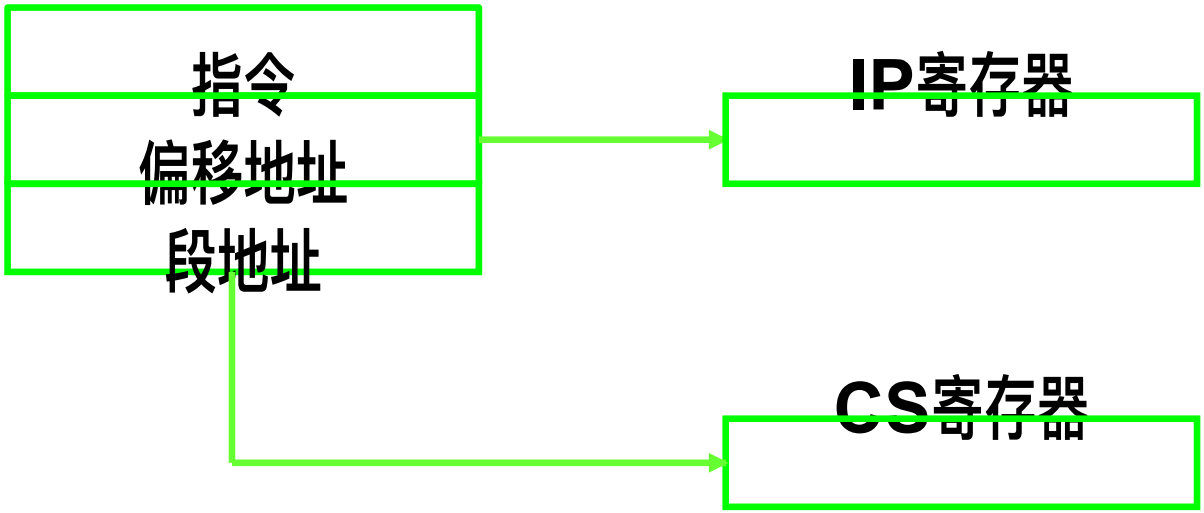
当程序从一个代码段转移到另一个代码段时，

不仅要改变IP寄存器的值，同时也要改变CS寄存器的值

，这种转移称为段间转移。段间转移又分为段间直接转移和段间间接转移。

（1） 段间直接寻址

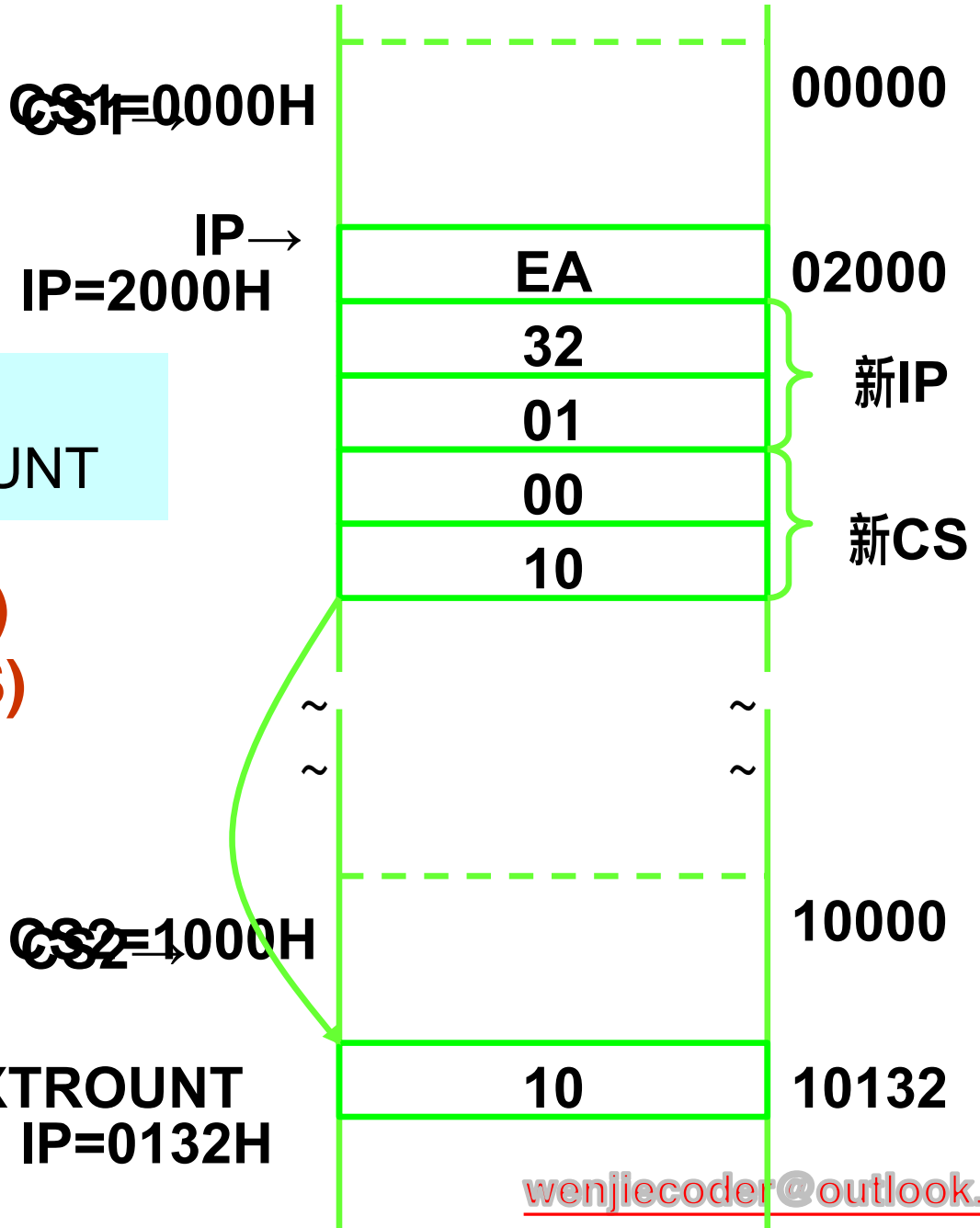
用指令中提供的转向段地址和偏移地址取代CS 和 IP。



与转移地址有关的寻址方式

【例】
JMP FAR PTR NEXTROUNT

offset NEXTROUNT? (IP)
seg NEXTROUNT? (CS)



与转移地址有关的寻址方式

(2) 段间间接寻址

用存储器中的两个相继字的内容取代CS 和IP，存储单元的地址可用存储器寻址方式得到。

与转移地址有关的寻址方式

【例】 JMP DWORD PTR [INTERS+BX]

DWORD PTR——双字操作符，转向地址双字（段间转移）

如DS=3000H，BX=1200H，INTERS=0020H，

则存储单元的物理地址PA=30000+0020+1200=31220H

指令执行前，CS=0000H，IP=1000H，（31220H）=40H，

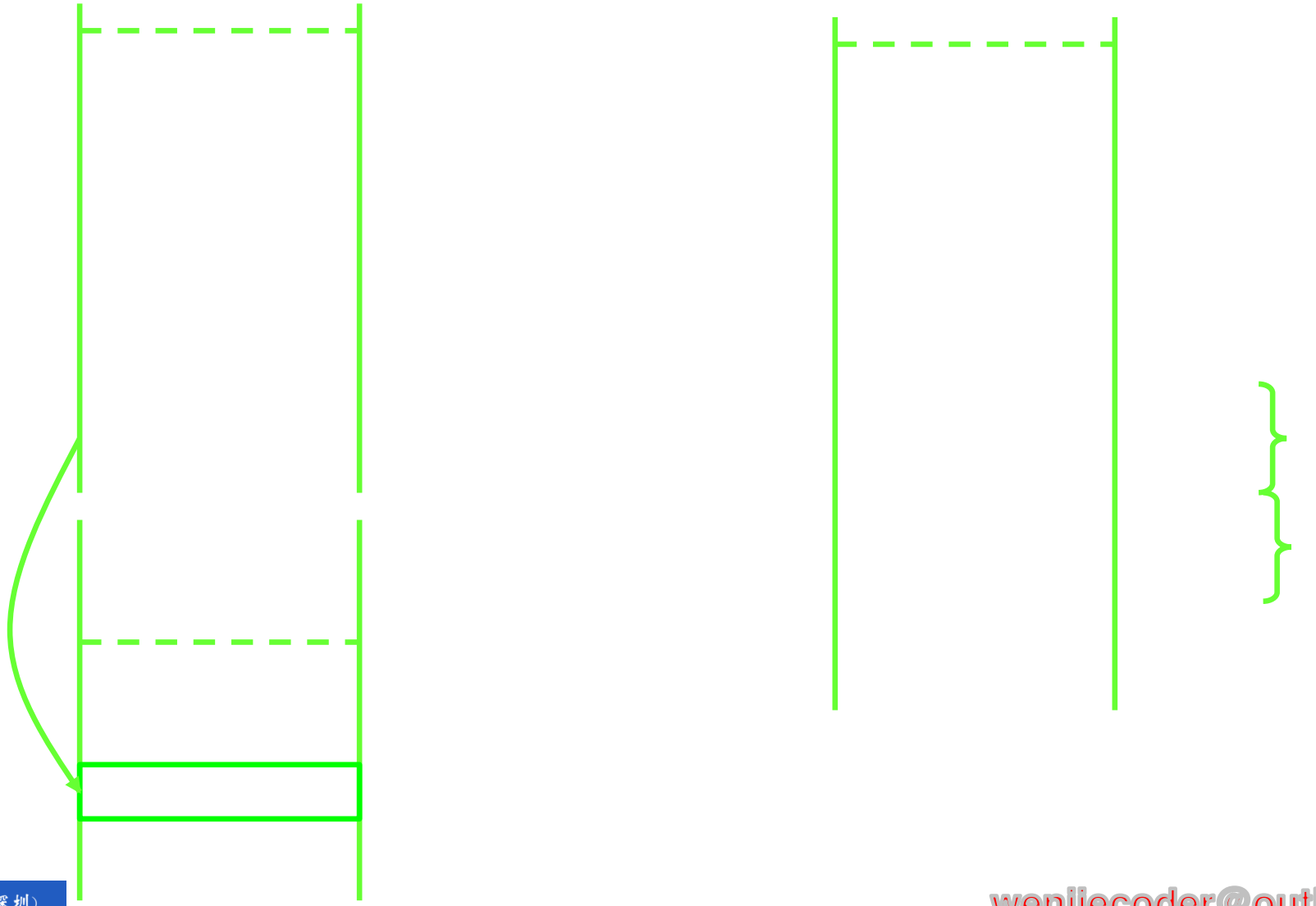
（31221）=01H，（31222H）=00H，（31223）=10H。

指令执行后，**IP=0140H，CS=1000H。先IP、后CS。**

wenjiecoder@outlook.com

与转移地址有关的寻址方式

指令存储和执行情况：



wenjiecoder@outlook.com

第二讲作业

Page 107-108：3.4、3.5、3.8

wenjiecoder@outlook.com