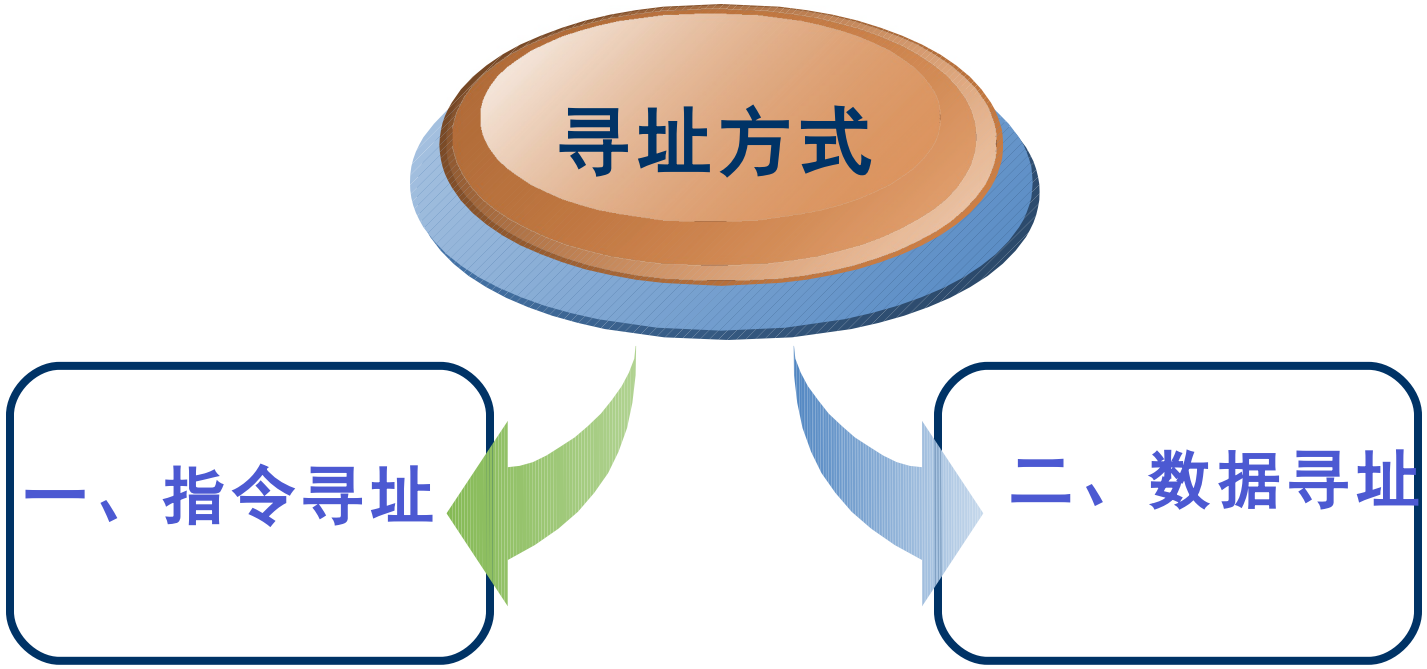
计算机组成原理与系统结构

第六章 指令系统

http://jpkc.hdu.edu.cn/computer/zcyl/dzkjdx/

第六章 指令系统

6.2 寻址方式



一、指令寻址

1. 顺序寻址方式

- 控制器中使用程序计数器 PC 来指示指令在内存中的地址。在程序顺序执行时，指令的地址码由 PC 自加 1 得出。
- 指令在内存中按顺序存放，当顺序执行一段程序时，根据 PC 从存储器取出当前指令， PC 自动＋ 1 ，然后执行这条指令；接着又根据 PC 指示从存储器取出下一条指令， PC 自动＋ 1 ， 执行……。

2. 跳跃寻址方式

- 当程序执行转移指令时，程序不再顺序执行，而是跳转到另一个地址去执行，此时，由该条转移指令的地址码字段可以得到新指令地址，然后将其置入 PC 中。

指令寻址（动画）

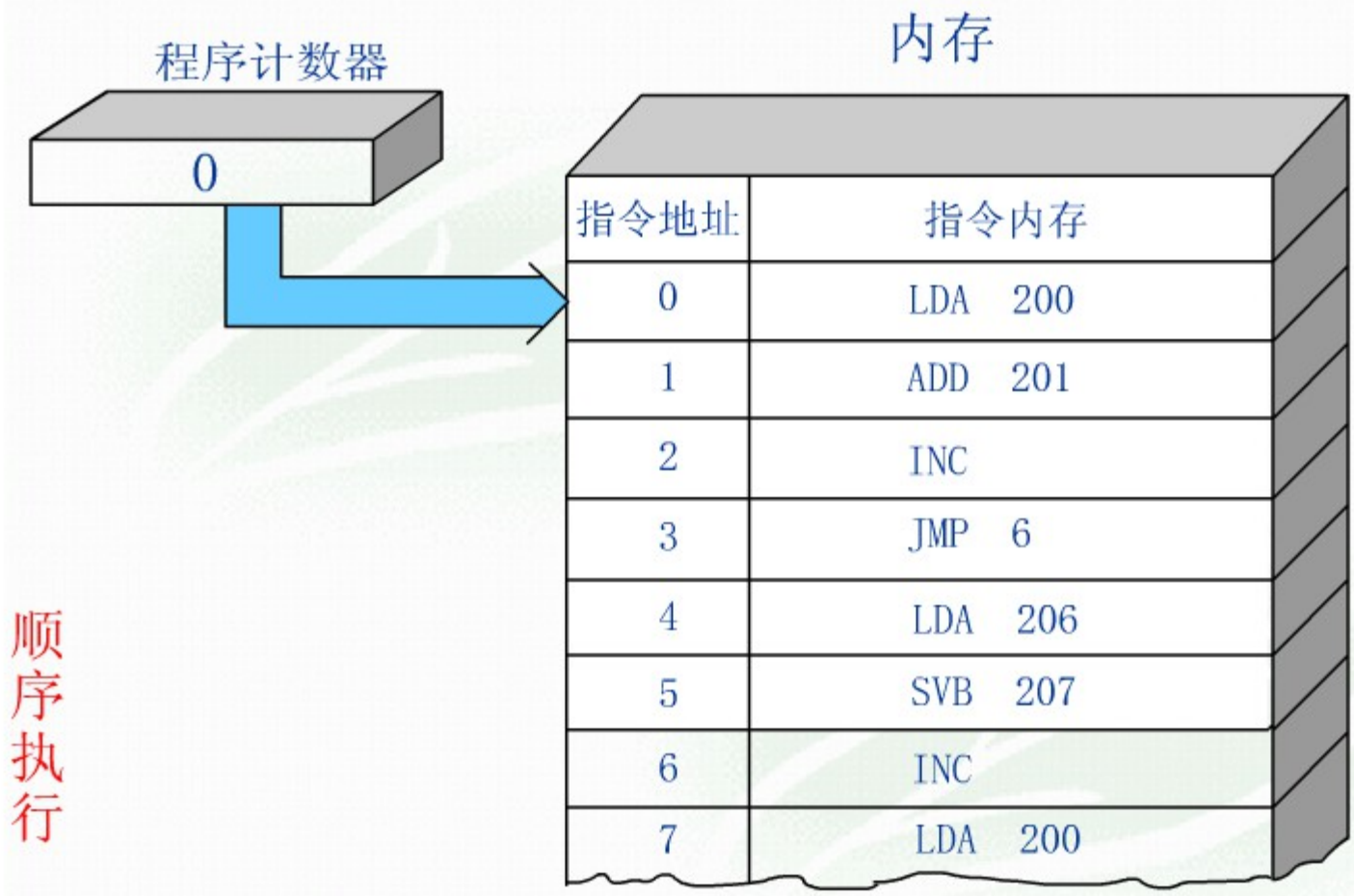
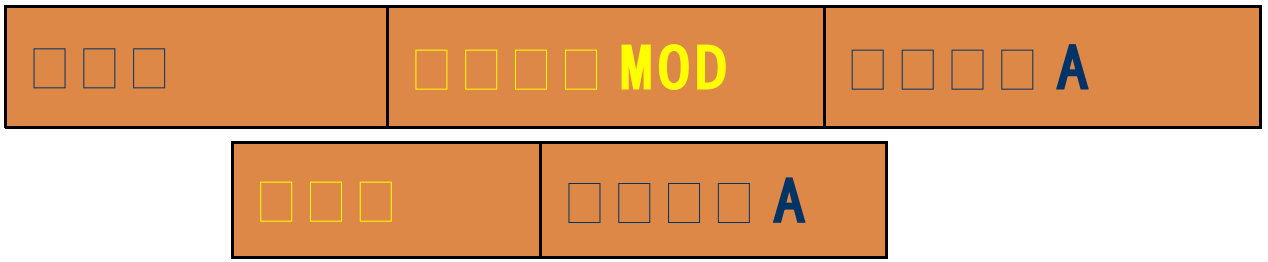


图4.1 指令的寻址方式

二、数据寻址

- **形式地址**：指指令的地址码字段，通常都不代表操作数的真实地址，记为 A 。
- **有效地址**：指操作数的真实地址，记作 EA ，它是由寻址方式和形式地址共同来确定的。
  - 问题：在实地址模式和虚拟地址模式下， EA 是？
- 常见的有 9 种基本的寻址方式
  - 复合寻址？
- 所有的计算机 CPU 均采用多种寻址方式
  - 问题：如何识别？



1、立即寻址（Immediate Addressin

- 操作数在指令的地址码字段，即：

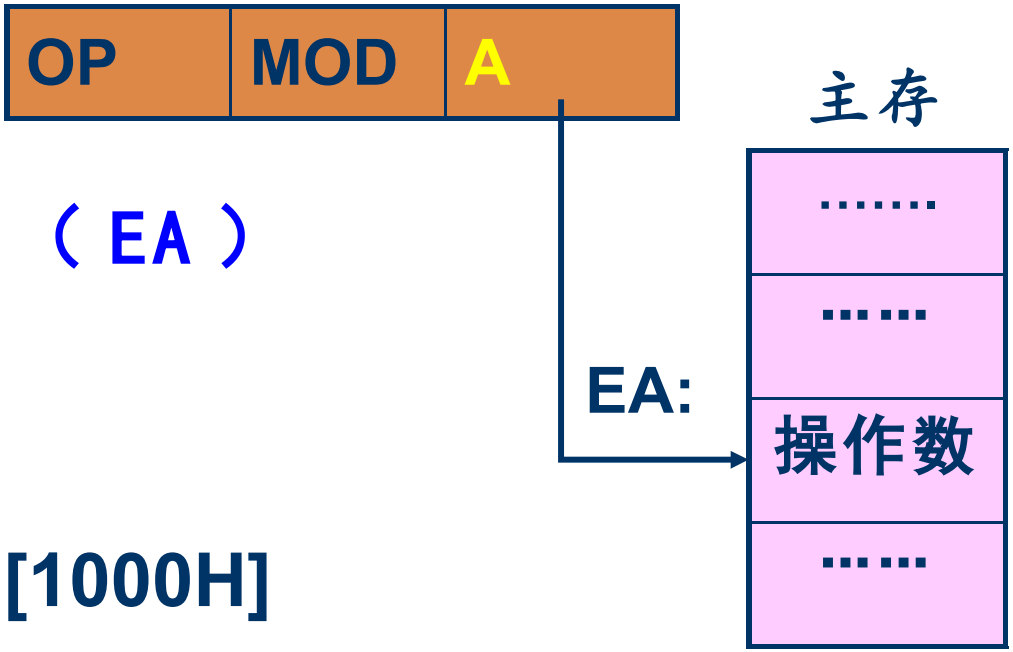
DATA ＝ A

- 例如：

```
MOV   AL, 5
MOV   AX, 3064H
MOV   AL, ‘A’
```

2．直接寻址（Direct Addressi

- 操作数位于存储器中，操作数所在的存储器单元的地址存放在指令的地址字段 A 中，即：



DATA ＝（EA）

EA=A

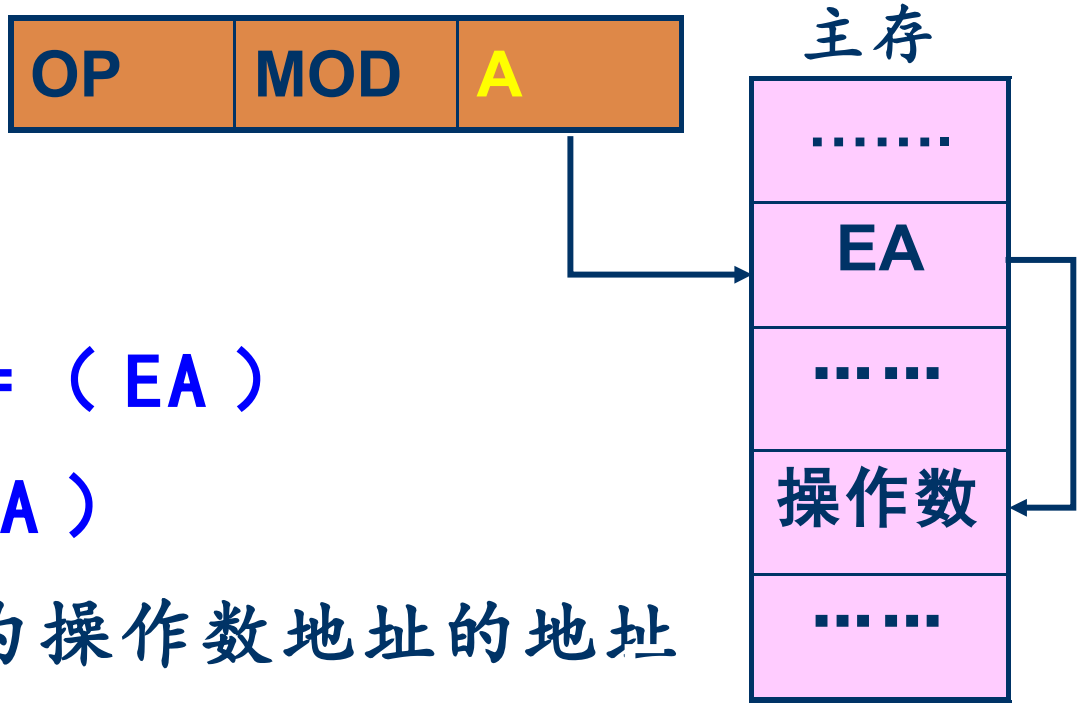
例如：

- MOV AX,[1000H]
- ADD [2000H],BX

直接寻址（动画）

3、间接寻址（ Indirect Addressing

- 操作数位于存储器中，操作数所在的存储器单元地址也存放在存储器中，该存储器地址则存放在指令的地址字段中，即：

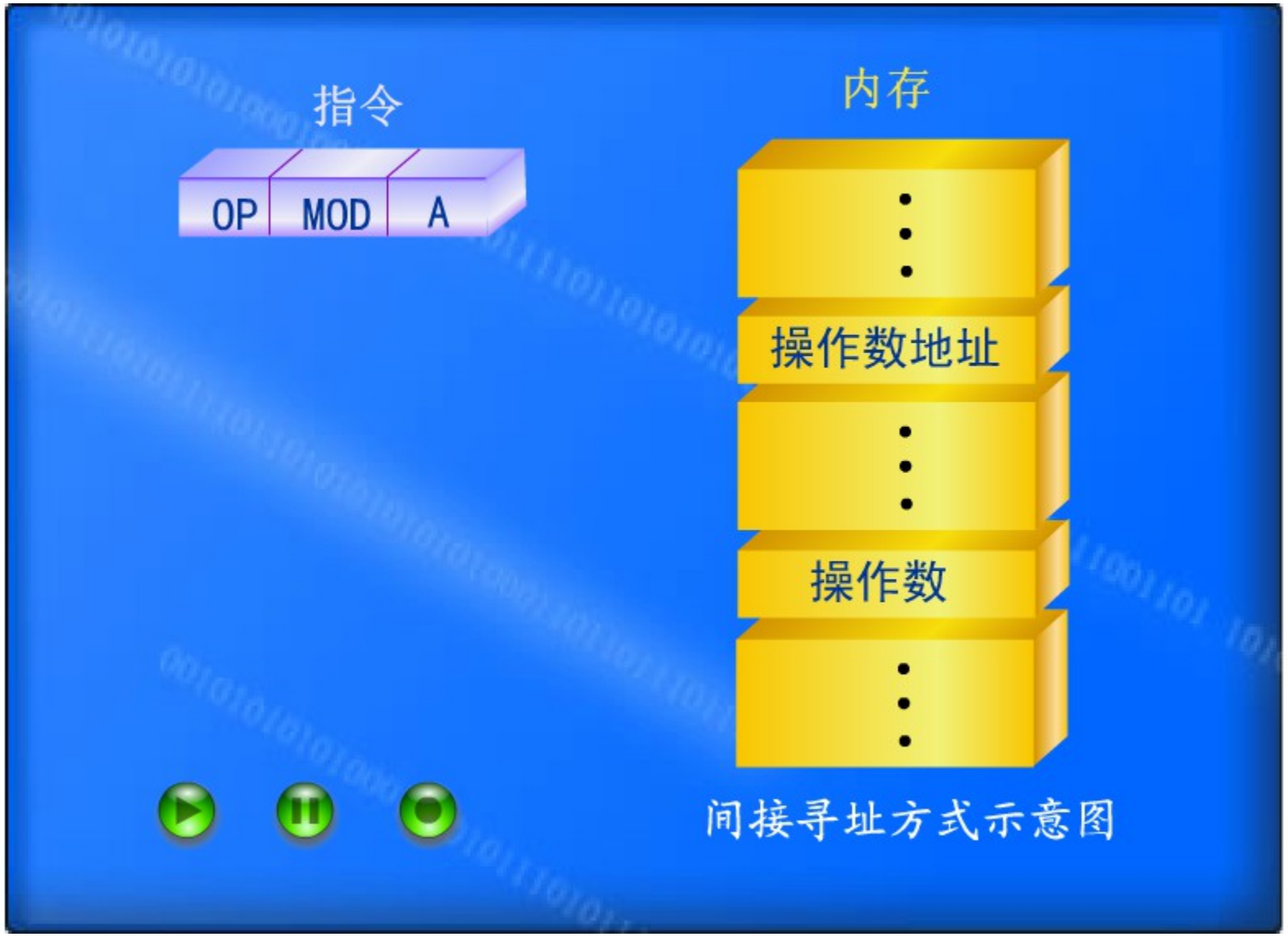


DATA ＝（EA）

EA=（A）

- 即： A 为操作数地址的地址

间接寻址（动画）



存储器

指令

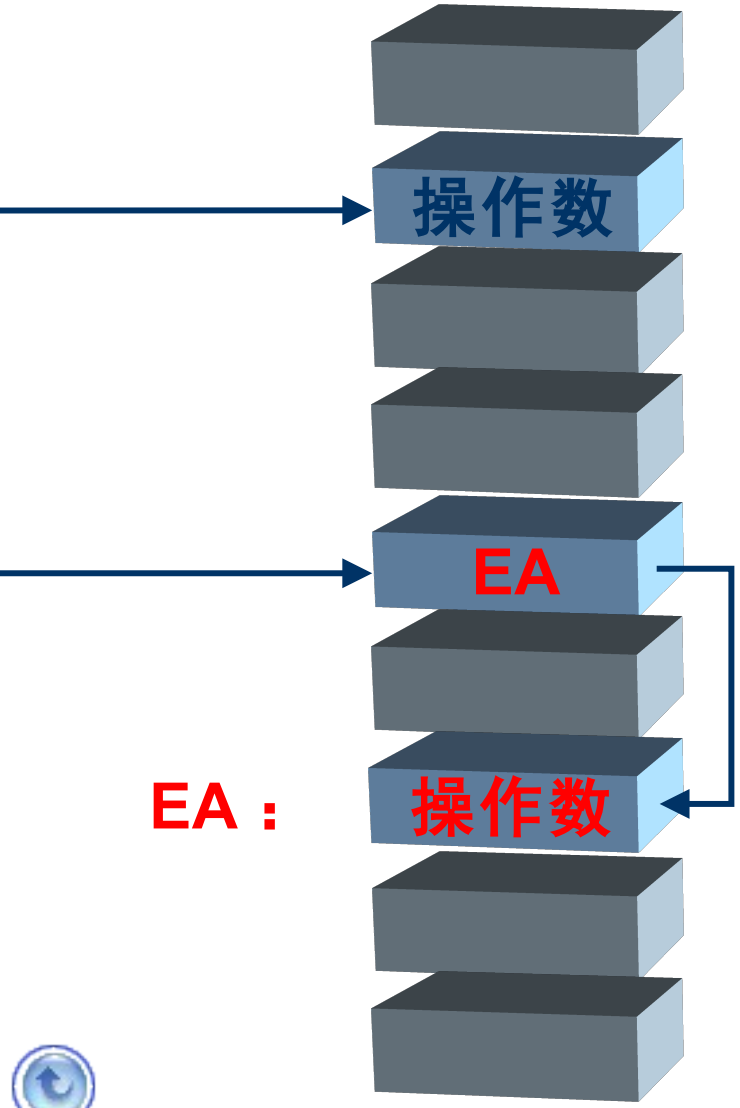
直接寻址方式 A

：

指令

间接寻址方式 A

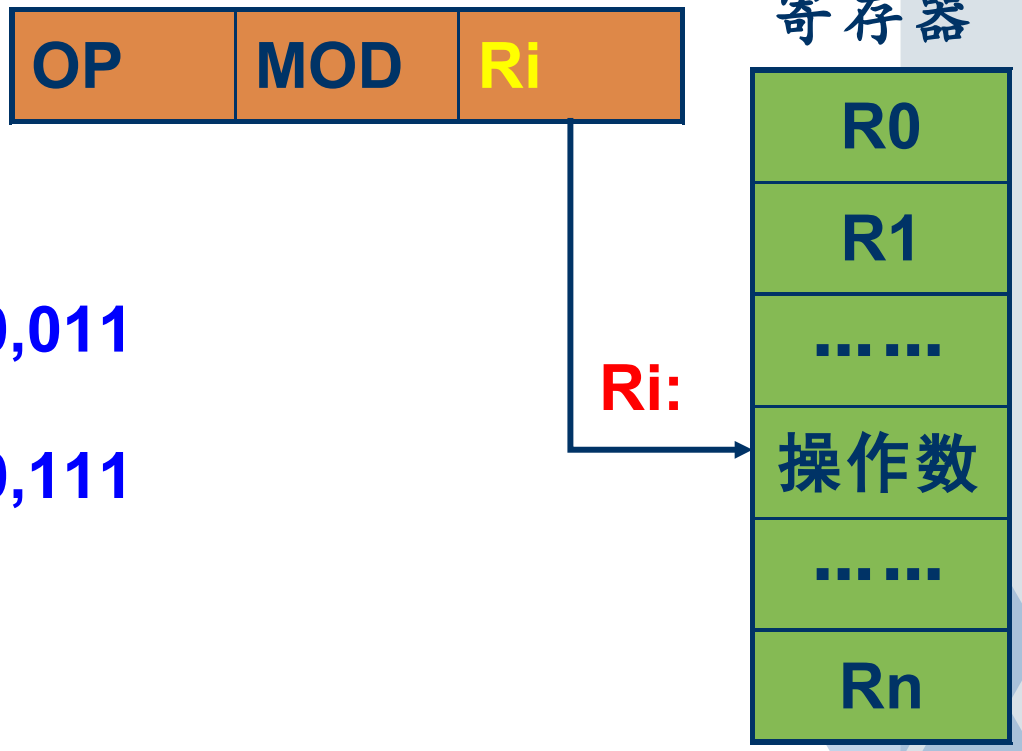
：



4、寄存器寻址方式 （ Register Addres

- 操作数位于寄存器中，操作数所在的寄存器编号存放在指令的地址字段 A 中，即：

DATA ＝（R_i）

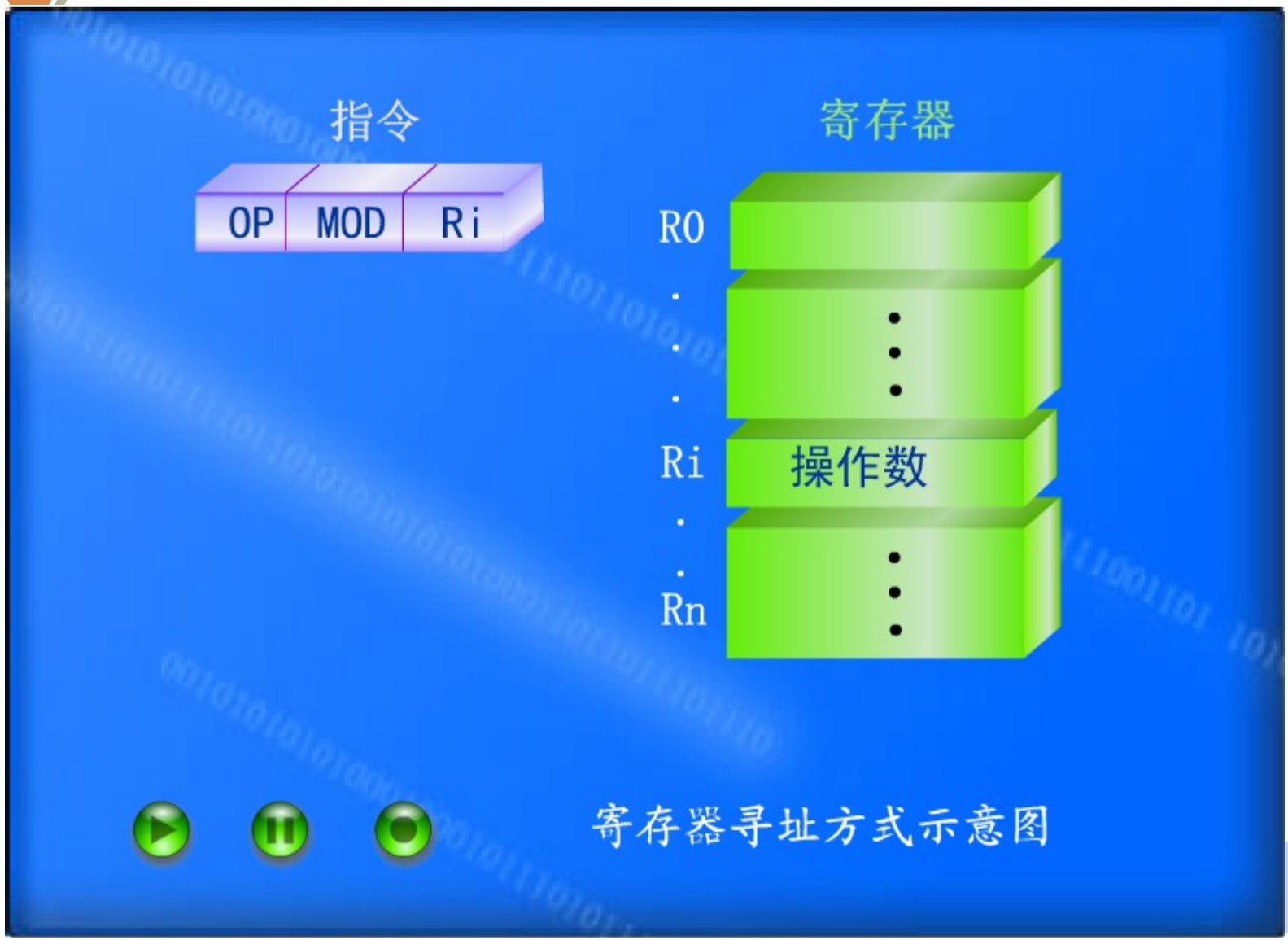


例如：

MOV AX, BX ;000,011

MOV AL, BH ;000,111

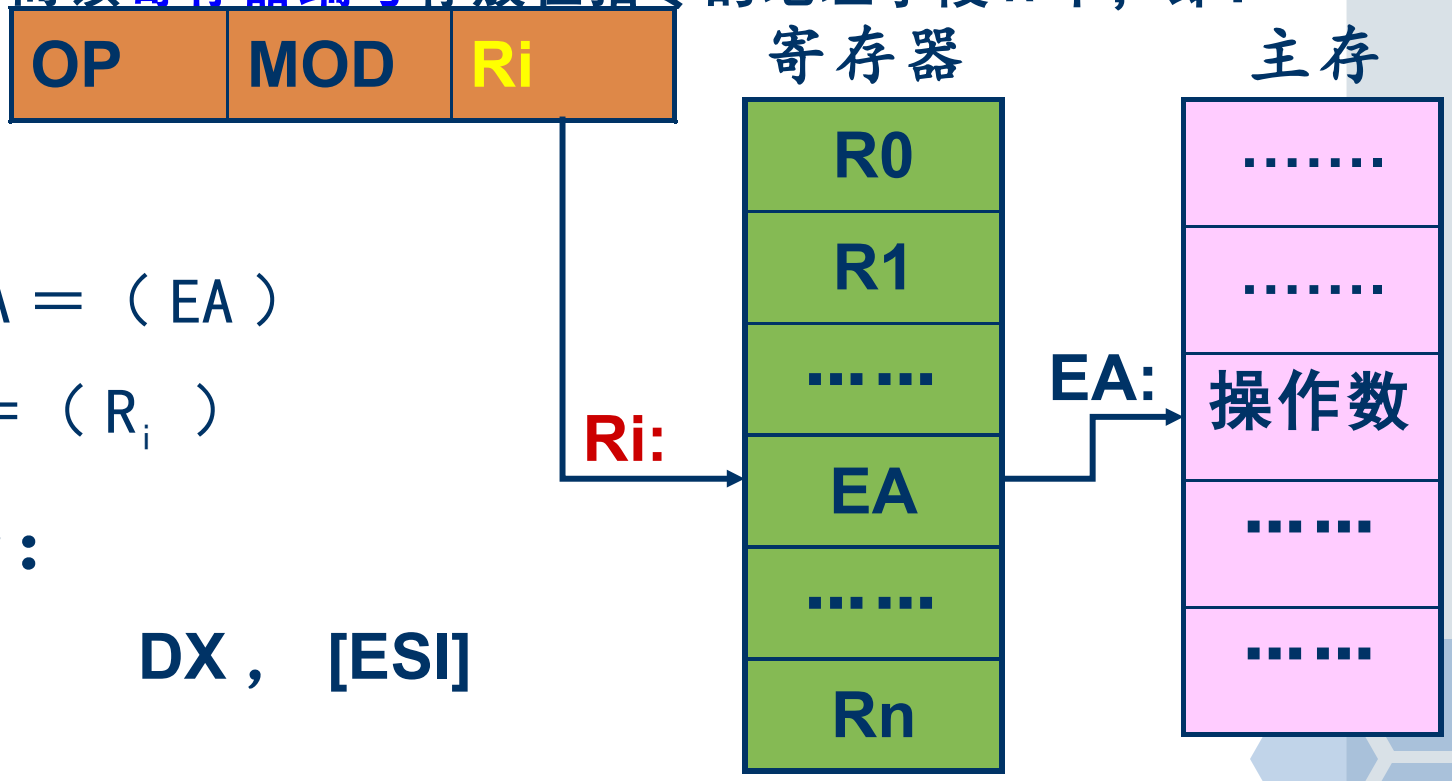
寄存器寻址方式（动画）



5、寄存器间接寻址方式

- 操作数位于存储器中，操作数所在的存储器地址存放在寄存器

中，而该寄存器编号存放在指令的地址字段 A 中，即：

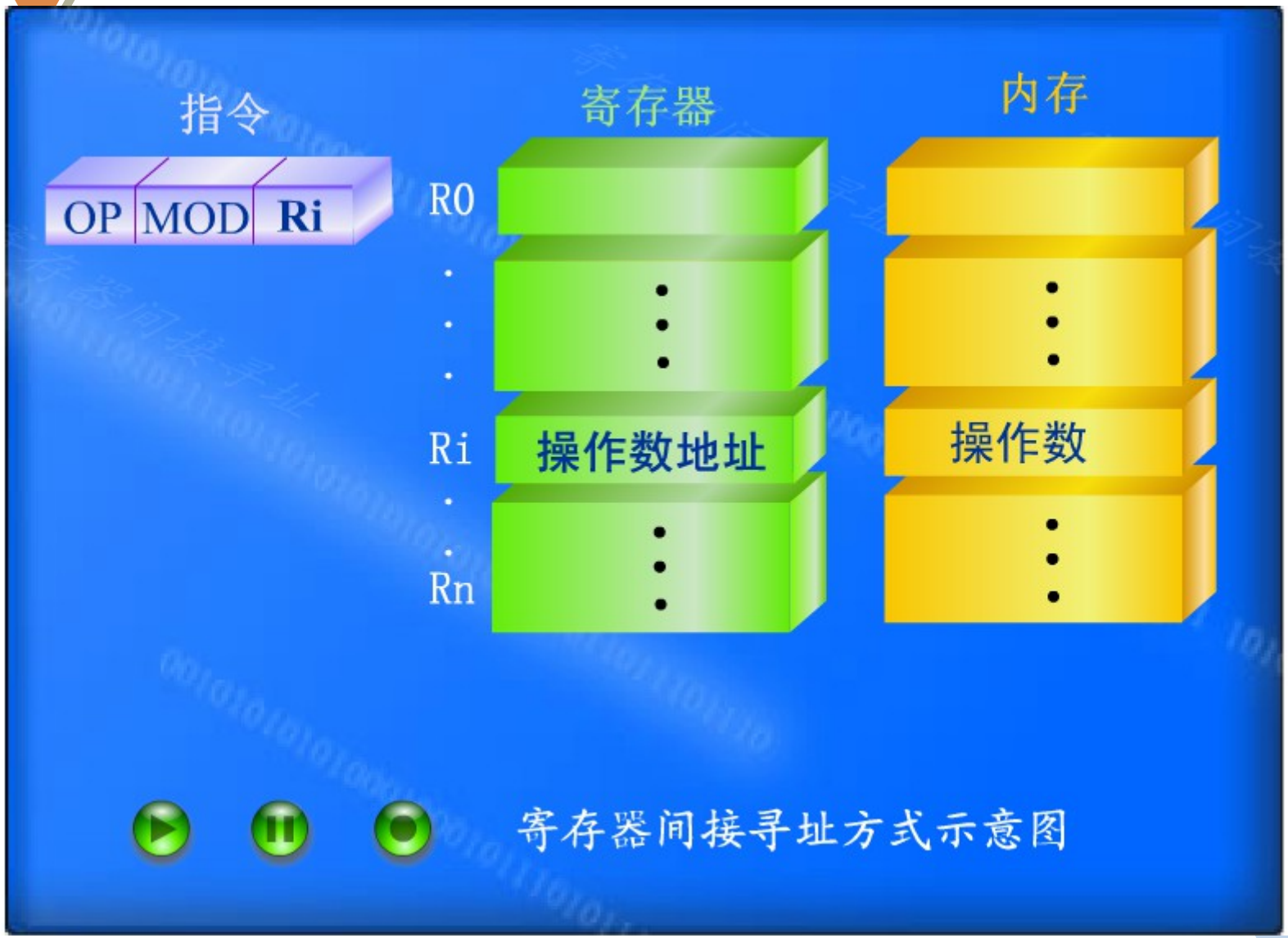


- DATA ＝（EA）
- EA ＝（R_i）

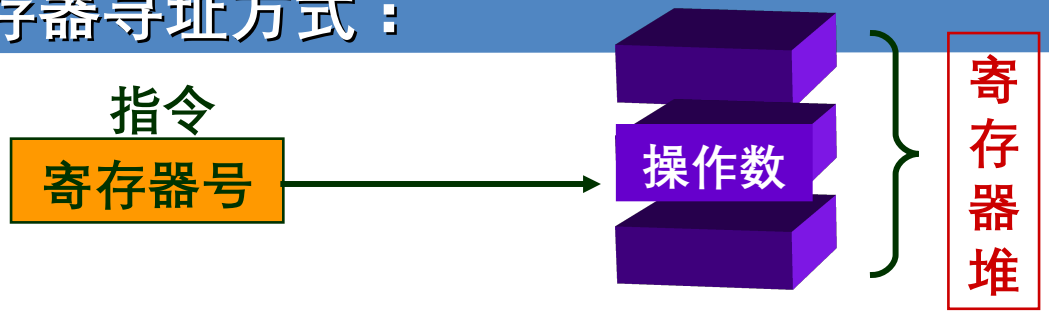
例如：

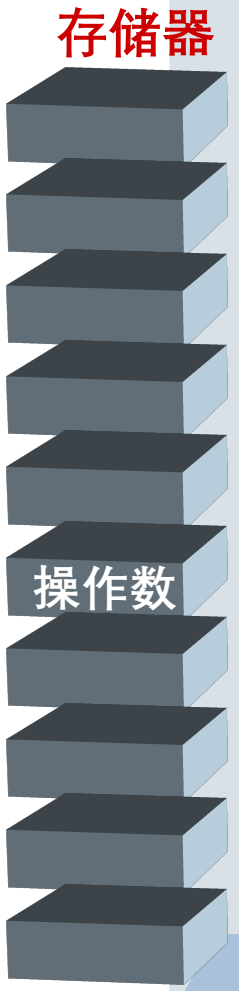
SUB　　DX，[ESI]

寄存器间接寻址方式（动画）

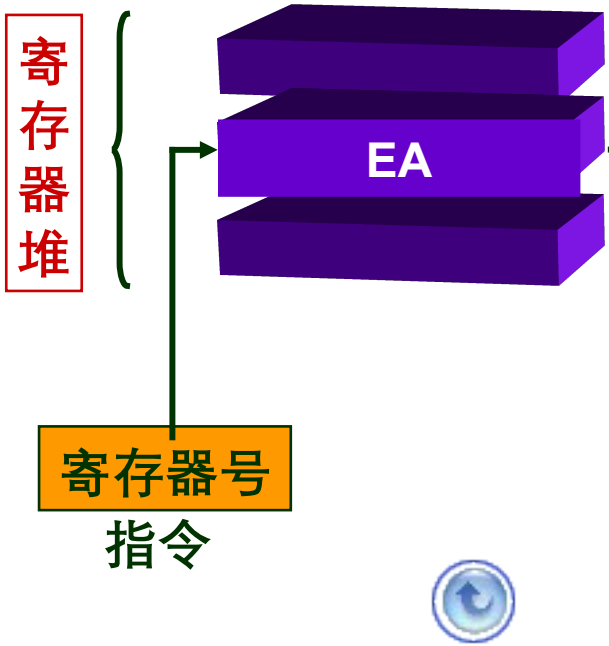


寄存器寻址方式：





寄存器间接寻址方式：



6、变址寻址（Indexed Addressin

- 操作数位于存储器中，操作数所在的存储器地址 EA 由变址寄存器 RI 和指令的地址字段 A 指出：

$$DATA = (EA)$$

$$EA = (RI) + A$$

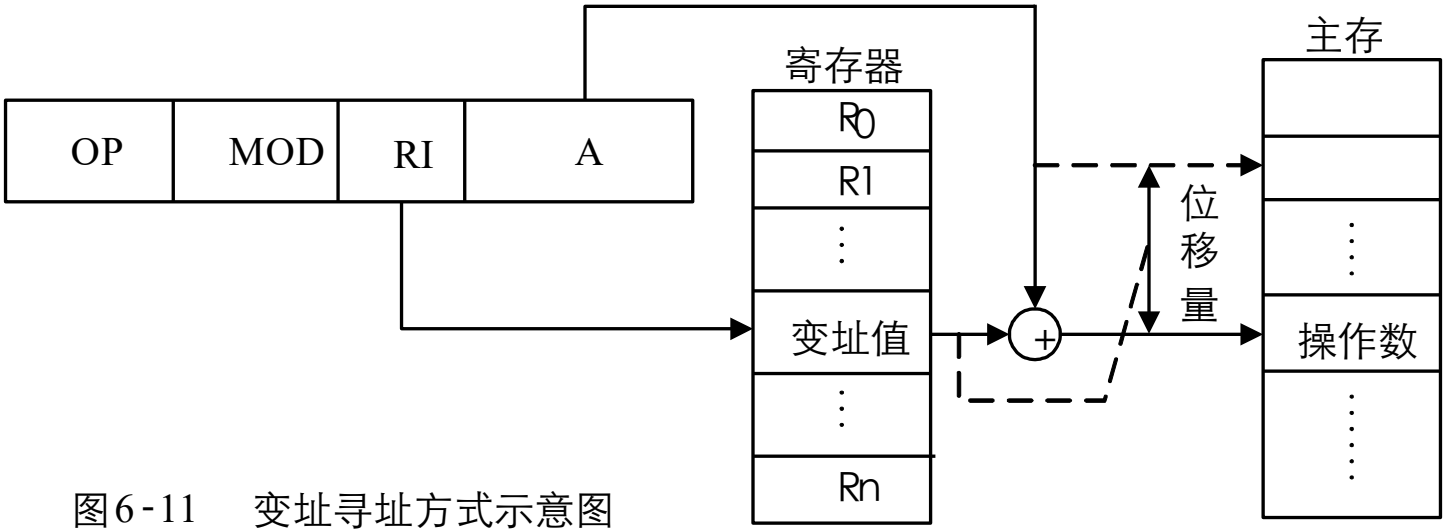
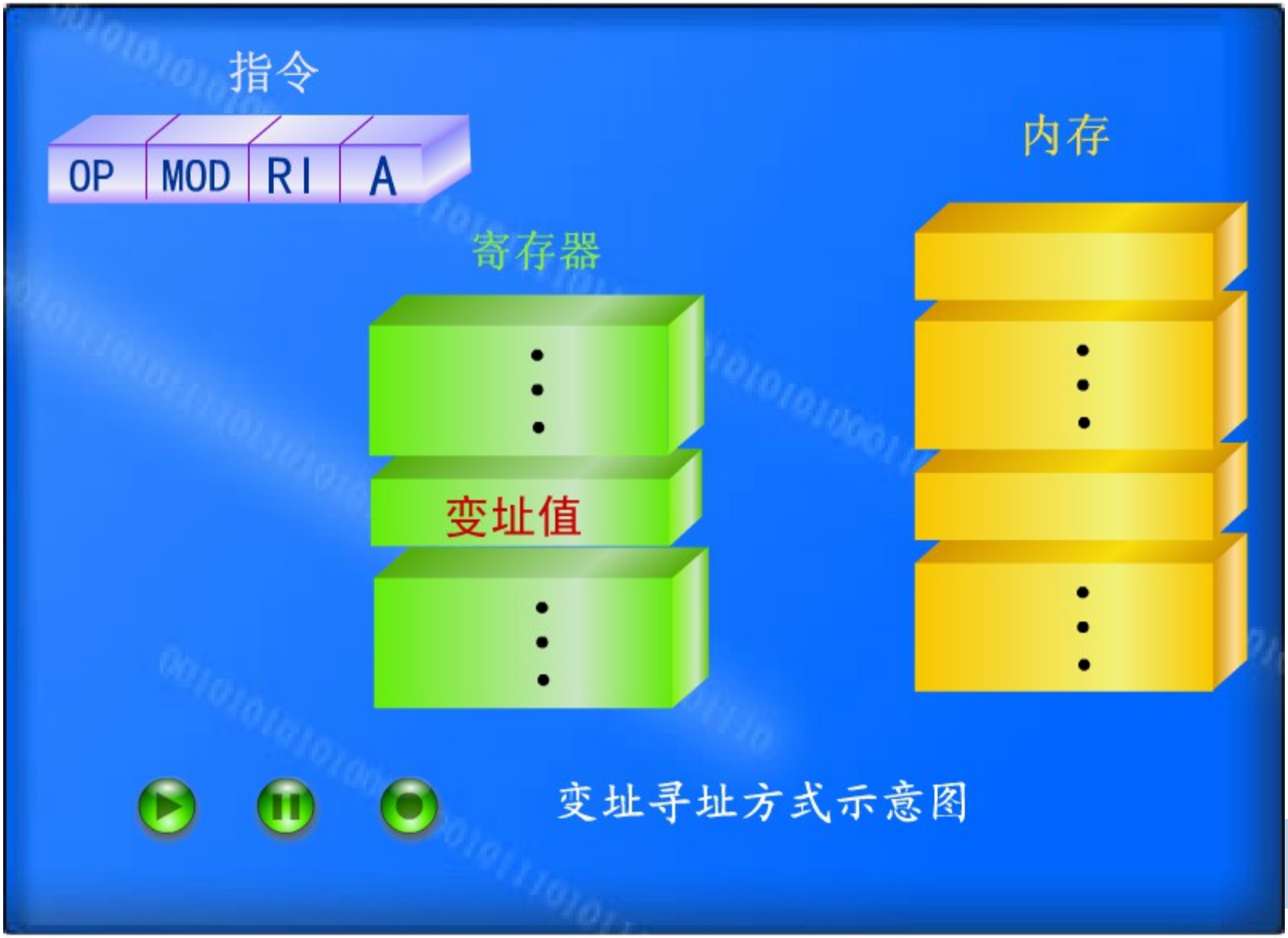


图6-11　变址寻址方式示意图

变址寻址（动画）



6、变址寻址（Indexed Addres

❖例如：

```
.data
   str_tb db   ‘Abort，  Retry？’，  0
.code
   MOV   ESI,0
   MOV   AL, str_tb[ESI]
   ……
   INC    ESI
```

串首址

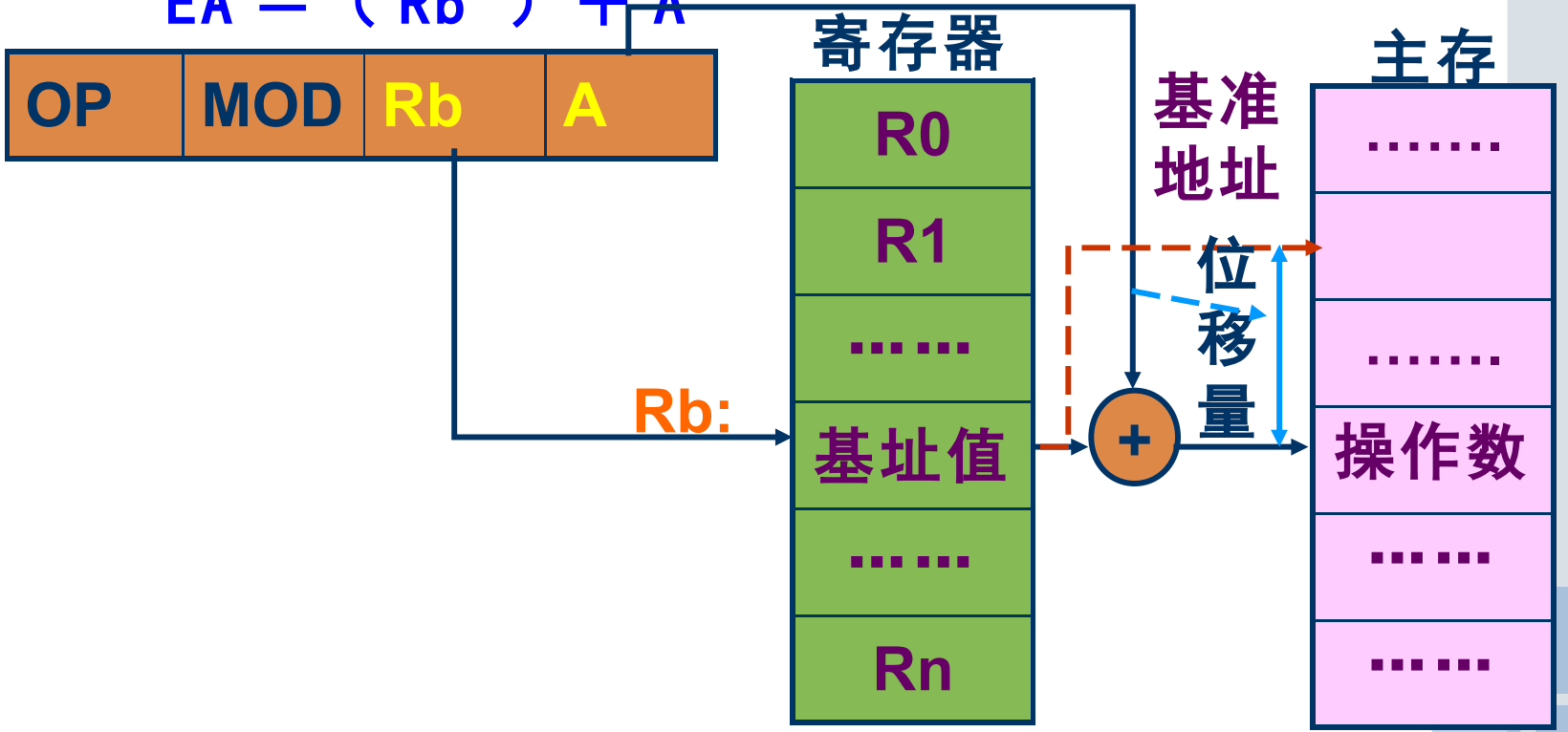
变址寄存器

7、基址寻址 （ Based Address

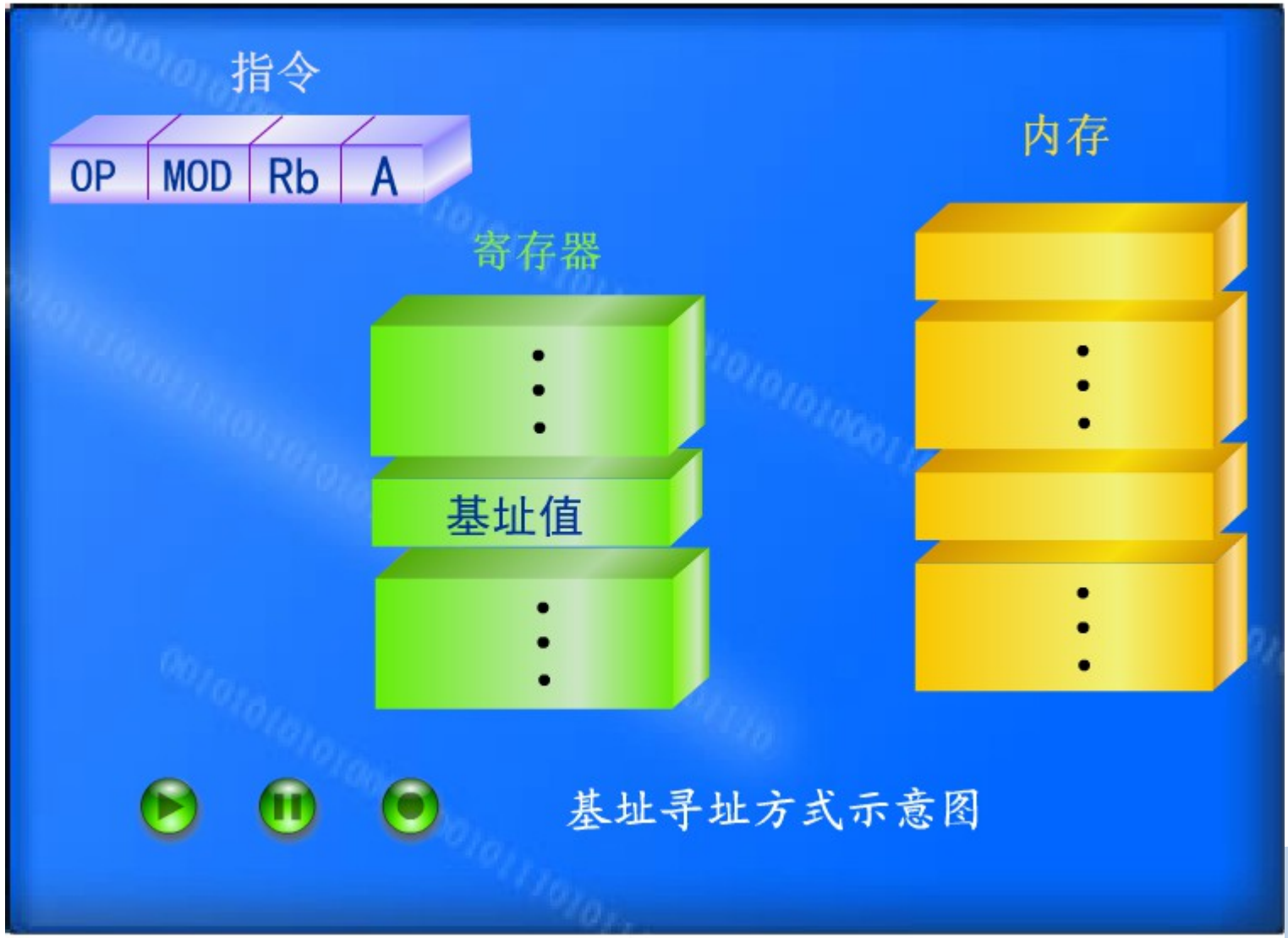
- 操作数位于存储器中，操作数所在的存储器地址 EA 由基址寄存器 Rb 和指令的地址字段 A 指出：

DATA ＝（ EA ）

EA ＝（ Rb ）＋ A



基址寻址（动画）

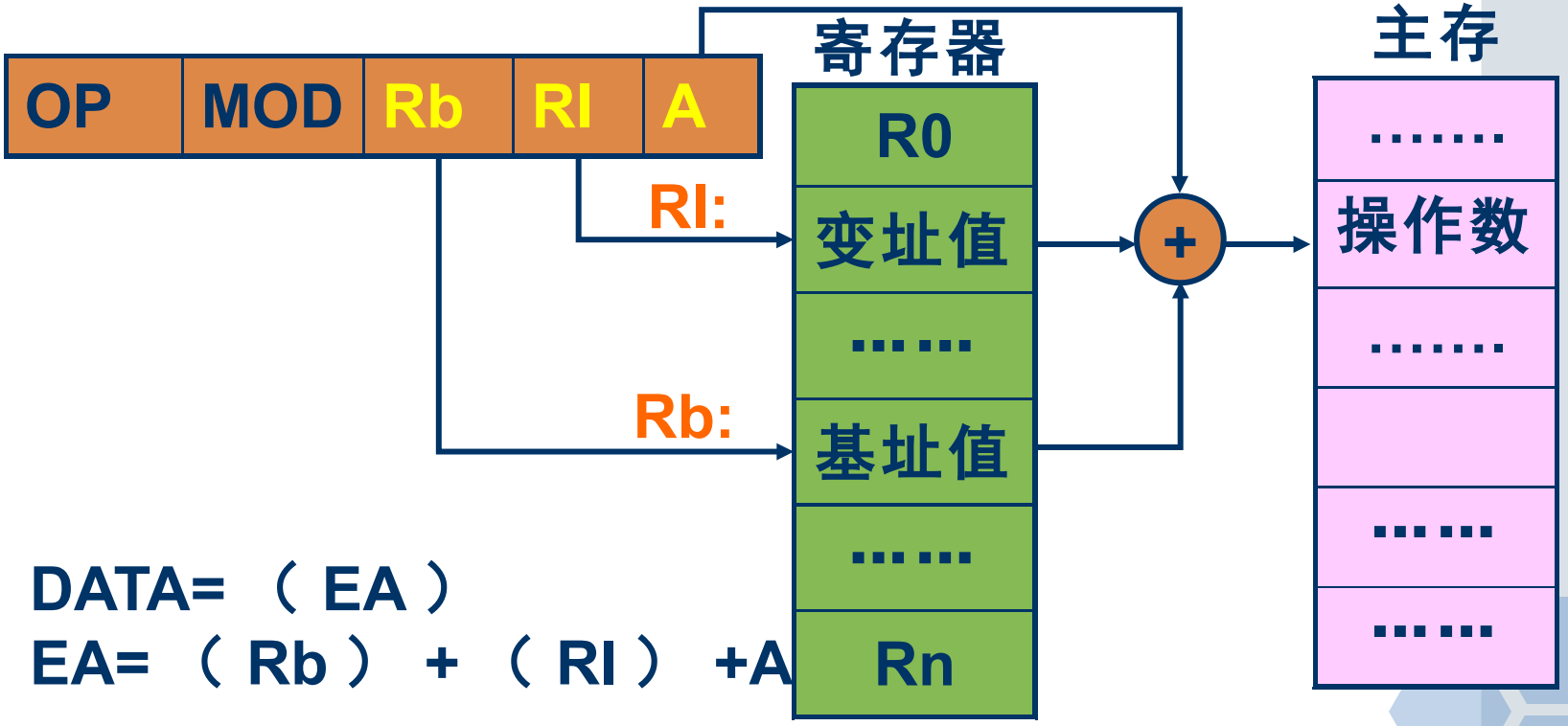


7、基址寻址 （ Based Address

- 基址寻址适合于多用户计算机系统，当操作系统为多道程序分配主存空间，将用户程序装入主存时，需要进行逻辑地址到物理地址的变换。
- 操作系统给每个用户一个基地址并放入其相应的基址寄存器，在程序执行时，以基址为基准自动进行逻辑地址到物理地址的变换。
- 在应用场合上，基址寻址面向系统，可以用来解决程序在主存中的重定位和扩大寻址空间等问题。而变址寻址是面向用户编程，用来访问字符串、向量和成批数据。

8、基址变址寻址

- 在指令中指定一个基址寄存器和一个变址寄存器，指令中的地址码给出位移量。有效地址是由基址寄存器中的值、变址寄存器中的值和位移量三者相加而成。



- DATA=（EA）
- EA=（Rb）+（RI）+A

9、相对寻址 （ Relative Addressin

- 操作数位于存储器中，操作数所在的存储器地址 EA 由程序计数器 PC 和指令的地址字段 A 指出：

  DATA ＝（ EA ）

  EA ＝（ PC ）＋ A

  ① A 通常称作相对偏移量 DISP 。

  ②相对寻址主要用于转移指令，执行之后，程序将转移到（ PC ）＋偏移量为地址的指令去执行。

  ③偏移量可正、可负，通常用补码表示，即可相对 PC 值向后或向前转移。

10、堆栈寻址 （ Stack Addressi

- 操作数位于存储器中，操作数所在的存储器地址 EA 由堆栈指针寄存器 SP 隐含指出，通常用于堆栈指令。
- 堆栈是由若干个连续主存单元组成的先进后出（first in last out，即FILO）存储区
- 第一个放入堆栈的数据存放在栈底，栈底是固定不变的。
- 最近放入的数据存放在栈顶，栈顶随着数据的入栈和出栈在时刻变化。
- 栈顶的地址由堆栈指针 SP 指明。

10、堆栈寻址 （ Stack Addressi

- 计算机中，堆栈从高地址向低地址扩展，即栈底的地址总是大于或等于栈顶的地址，称为**上推堆栈**。也有少数计算机相反，称为**下推堆栈**。
- 堆栈寻址主要用来暂存中断和子程序调用时现场数据及返回地址。
- 堆栈指针的管理：

  ①SP 总是指向最后压入的有效数据

  ②SP 总是指向栈顶的空单元

堆栈的结构

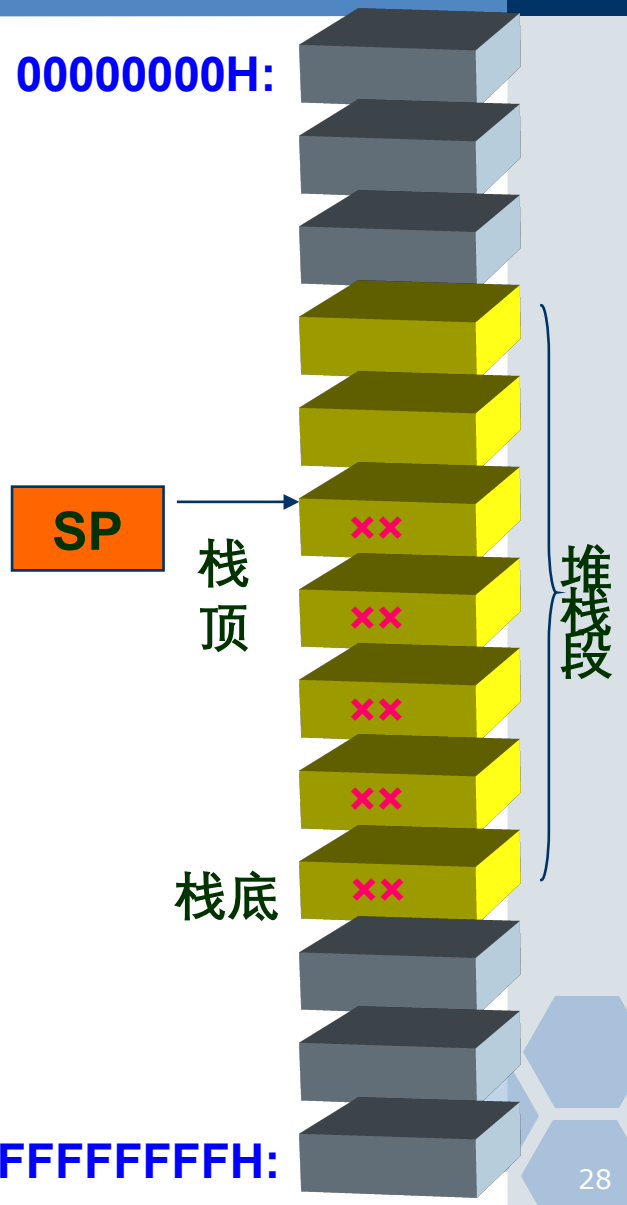
存储器

- 堆栈的操作：压入（PUSH）和弹出（POP），对应 PUSH 和 POP 指令，假设数据字长为 1B

  ①压入指令 PUSH Ri：将 Ri 寄存器内容压入堆栈。其操作是：

  (SP) – 1→SP，

  (Ri) → (SP)



堆栈的结构

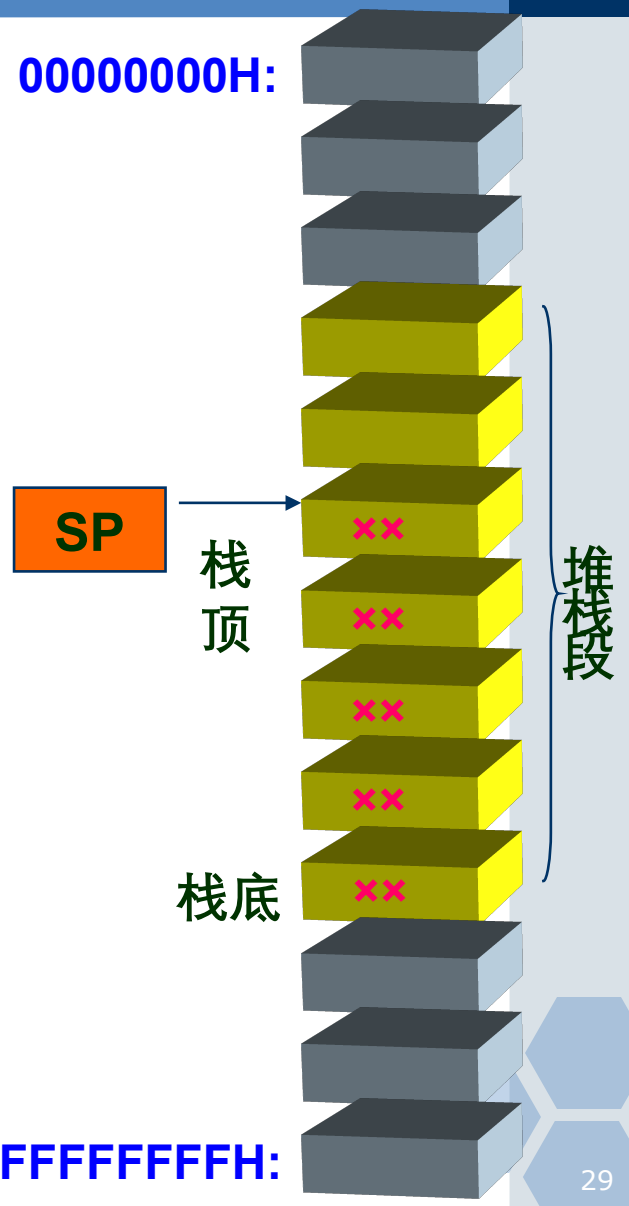
存储器

②弹出指令 POP Ri：从堆栈中弹出 1 个数据送 Ri 寄存器，其操作是：

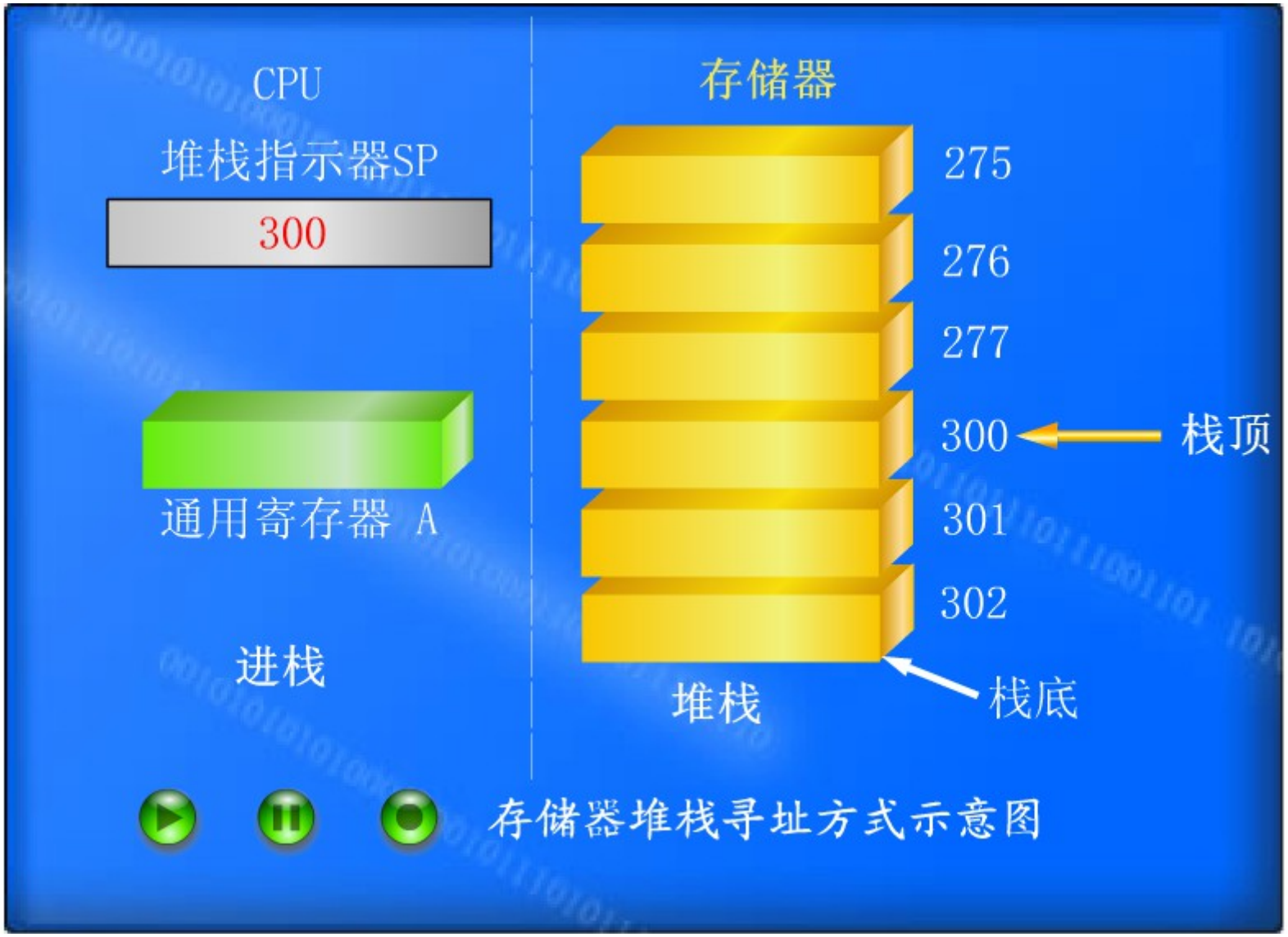
((SP)) → Ri，

(SP) + 1→SP

③(SP) 表示堆栈指针 SP 的内容；((SP)) 表示 SP 所指的栈顶的内容。



堆栈寻址（动画）


CPU
堆栈指示器SP
300
通用寄存器 A
进栈
存储器
275
276
277
300
栈顶
301
302
堆栈
栈底
存储器堆栈寻址方式示意图

总结

寻址方式	操作数类型	地址码字段 A	EA
立即寻址	立即数	立即数 Data	——
直接寻址	存储器	直接地址 A	EA=A
间接寻址	存储器	间接地址 A	EA=(A)
寄存器寻址	寄存器	寄存器号 n	——
寄存器间接寻址	存储器	寄存器号 n	EA=（Rn）
变址寻址	存储器	形式地址 X	EA=（RI）+X
基址寻址	存储器	相对偏移量 Disp	EA=（Rb）+Disp
相对寻址	存储器	相对偏移量 Disp	EA=（PC）+Disp

The End !