

## 第四讲 指令系统

刘财政

启航教育

# 本讲内容

考点一：机器指令

考点二：指令设计



考点三：寻址方式



考点四：CISC和RISC



启航教育 刘财政

## 考点一：机器指令

启航教育

刘财政

## 考点框架



指令的基本格式



指令类型

启航教育 刘财政

# 考点一：机器指令

## 指令的基本格式

启航教育 刘财政

## 考点一：机器指令



### 冯·诺依曼计算机

- (1) 计算机由五大部件组成
- (2) 指令和数据以同等地位存于存储器
- (3) 指令和数据用二进制表示
- (4) 指令由操作码和地址码组成
- (5) 存储程序
- (6) 以运算器为中心
- (7) **控制流驱动的工作方式**
- (8) 按地址寻访，寻址有数据寻址和指令寻址两种

## 考点一：机器指令

### 机器指令格式

一条指令就是机器语言的一个语句，它是一组有意义的二进制代码。

操作码	地址码
-----	-----

**操作码：**指明操作的性质及功能。

**地址码：**指明操作数的地址。

## 考点一：机器指令

指令长度取决于操作码的长度、操作数的地址数以及操作数地址的个数

操作码：指定操作类型

(操作码长度：固定 / 可变)

源操作数参照：一个或多个源操作数所在的地址

(操作数来源：主(虚)存/寄存器/I/O端口/指令本身)

结果值参照：产生的结果存放何处 (目的操作数)

(结果地址：主(虚)存/寄存器/I/O端口)

下一条指令地址：下条指令存放何处

(下条指令地址：主(虚)存)

(正常情况隐含在PC中，改变顺序时由指令给出)



# 考点一：机器指令

## 指令类型

启航教育 刘财政

## 考点一：机器指令

### 零地址指令

- (1) 无需操作数 如：空操作 / 停机等
- (2) 所需操作数为默认的 如：堆栈 / 累加器等

形式：

OP
----

### 一地址指令

其地址既是操作数的地址，也是结果的地址

- (1) 单目运算：如：取反 / 取负等
- (2) 双目运算：另一操作数为默认的 如：累加器等

形式：

OP
----

A1
----

## 考点一：机器指令

### 二地址指令（最常用）

分别存放双目运算中两个操作数，并将其中一个地址作为结果的地址。

形式：

OP	A1	A2
----	----	----

### 三地址指令（RISC风格）

分别作为双目运算中两个源操作数的地址和一个结果的地址。

形式：

OP	A1	A2	A3
----	----	----	----

### 多地址指令

用于成批数据处理的指令，如：向量 / 矩阵等运算的SIMD指令。

**【谢谢大家】**

启航教育 刘财政

## 考点二：指令设计

启航教育 刘财政



## 考点框架



定长指令格式



扩展操作码指令设计



操作数据类型和操作类型

启航教育

刘财政

## 考点一：机器指令



### 机器指令格式设计原则

指令格式的选择应遵循的几条基本原则

- 应尽量短
- 要有足够的操作码位数
- 指令编码必须有唯一的解释，否则是不合法的指令
- 指令字长应是字节的整数倍
- 合理地选择地址字段的个数
- 指令尽量规整

## 考点一：机器指令

### 机器指令格式

一条指令就是机器语言的一个语句，它是一组有意义的二进制代码。

操作码	地址码
-----	-----

**操作码：**指明操作的性质及功能。

**地址码：**指明操作数的地址。



## 考点一：机器指令

### 定长指令格式

#### 基本思想

操作码的长度决定了指令系统中完成不同操作的指令条数

若操作码长度为 $k$ 位，最多只能有 $2^k$ 条不同指令

#### 特点

译码方便，但有信息冗余

## 考点一：机器指令

### 定长指令格式

#### 基本思想

指令的操作码部分采用固定长度的编码

如：假设操作码固定为6位，则系统最多可表示64种指令

#### 特点

译码方便，但有信息冗余

## 考点一：机器指令



### 扩展操作码指令格式

- 操作码的长度可变，且分散地放在指令字的不同字段中,通过逐步扩展操作码来实现指令格式的扩展
- 通常是在指令字中用一个固定长度的字段来表示基本操作码，而对于一部分不需要某个地址码的指令，把它们的操作码扩充到该地址码字段

## 考点一：机器指令



### 扩展操作码指令格式

在设计扩展操作码指令格式时，必须注意以下两点：

- (1) 不允许短码是长码的前缀，即短操作码不能与长操作码的前面部分的代码相同。【哈夫曼编码】
- (2) 各指令的操作码不能重复。

**【政哥点拨】**

启航教育 刘财政

1、以一个例子来说明扩展操作码指令格式的功能原理。假设某机器的指令格式用三地址格式表示，其中操作码是4位，A1、A2、A3都是4位。如果要设计15条一地址指令，15条二地址指令，15条三地址指令，16条零地址指令，该如何进行设计。

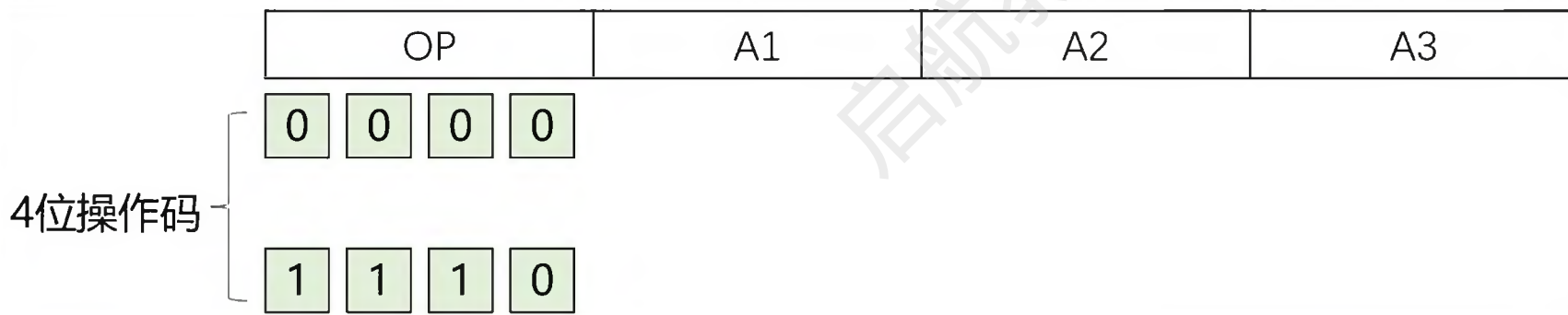
解析：若都是三地址指令，有16条。

OP	A1	A2	A3
----	----	----	----

1、以一个例子来说明扩展操作码指令格式的功能原理。假设某机器的指令格式用三地址格式表示，其中操作码是4位，A1、A2、A3都是4位。如果要设计15条一地址指令，15条二地址指令，15条三地址指令，16条零地址指令，该如何进行设计。

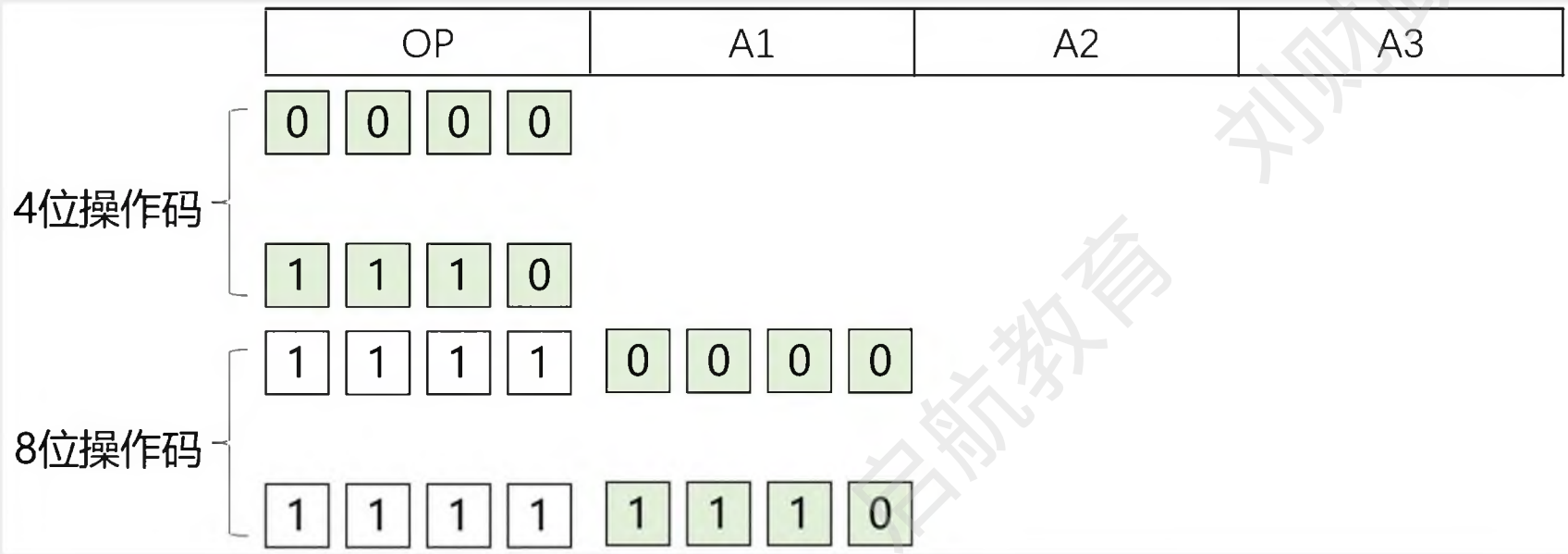
解析：若三地址指令需15条，两地址指令需要15条，一地址指令需要15条，零地址指令需要16条，共61条指令，则操作码的长度需向地址码扩展。

(1) 15条三地址指令的操作码由4位基本操作码从0000~1110给出，剩下一个码点（组合）1111用于把操作码扩展到A1，即4位扩展到8位。



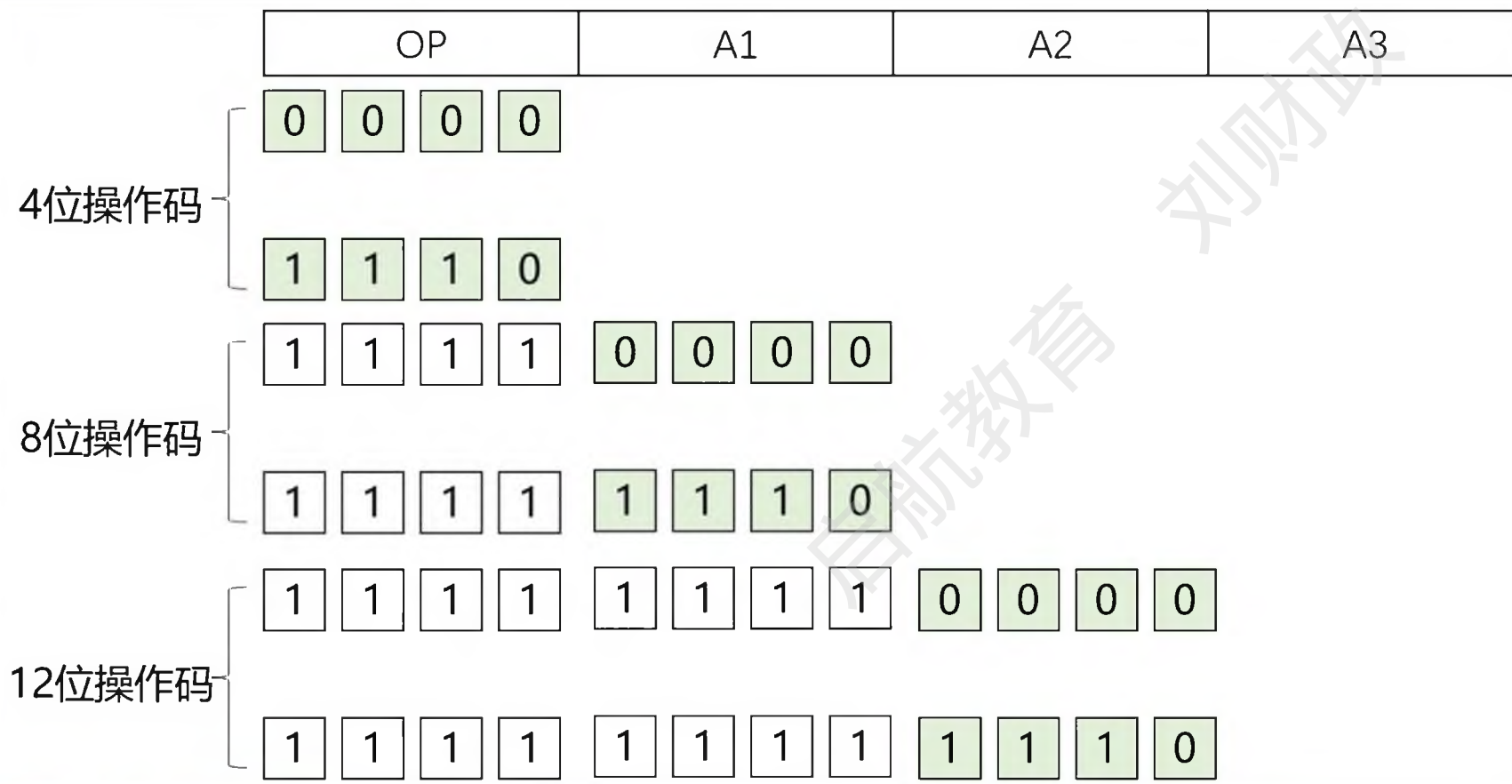
解析：若三地址指令需15条，两地址指令需要15条，一地址指令需要15条，零地址指令需要16条，共61条指令，则操作码的长度需向地址码扩展。

(2) 15条二地址指令的操作码由8位操作码从11110000~11111110给出，剩下一个码点11111111用于把操作码扩展到A2，即从8位扩展到12位。

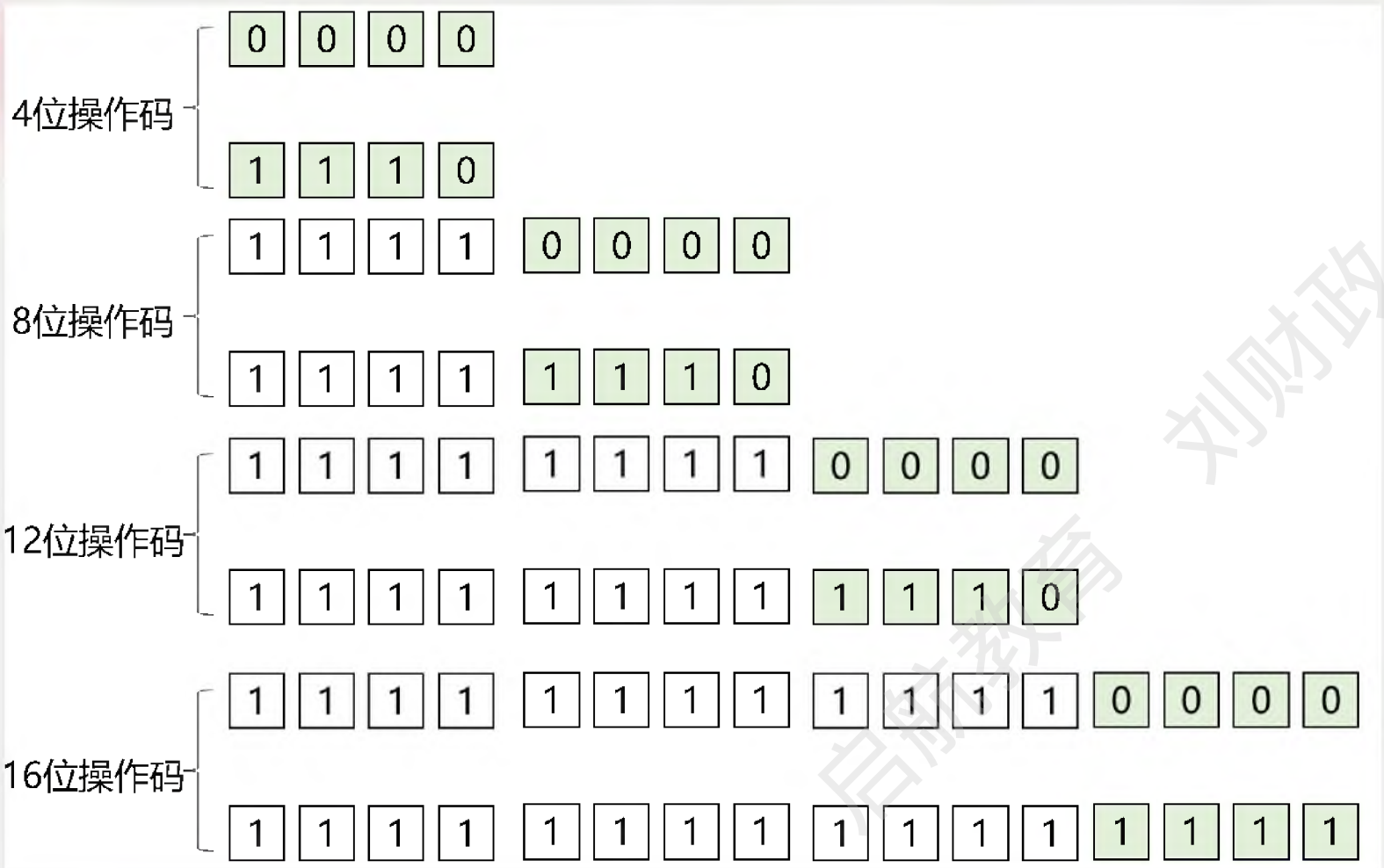




(3) 15条一地址指令的操作码由12位操作码从111111110000 ~ 111111111110给出，剩下一个码点111111111111用于把操作码扩展到A3，即从12位扩展到16位。







在设计时，有一个结论大家，需要记住，上一级的地址每留出一条指令，下一级就扩展 $2^K$ 条指令，其中K是地址码位数。

2、【航天系统，2018】假设指令字长为16位，操作数地址码为6位，指令有零地址、一地址和二地址3种格式。设操作码固定，若零地址指令有M条，一地址指令有N条，则二地址指令最多有( )。

A.  $10-M-N$ 条    B.  $12-M-N$ 条    C.  $16-M-N$ 条    D.  $32-M-N$ 条

C 【解析】因为采用指令具有两个地址码，且每个地址码的位数是6位，所以两个地址码的位数是12，而指令字长是16位，那么操作码是4位，共计有16条指令；由于零地址指令有M条、一地址指令有N条，因此二地址指令的条数是 $16-M-N$ 。

## 考点一：机器指令

### 操作数类型和操作类型

- 操作数类型：包括地址、数字、字符、逻辑数据等。
- 不同机器具有不同的操作类型，通用的操作包括数据传送、算术逻辑操作、移位、转移、输入/输出、其他（等待指令、停机指令、开中断指令等）。

【牛刀小试】

启航教育 刘财政

1. 程序控制类指令的功能是( )。

- A. 进行算术运算和逻辑运算
- B. 进行主存与CPU之间的数据传送
- C. 进行CPU和I/O设备之间的数据传送
- D. 改变程序执行顺序

D 【解析】 一般情况下，CPU执行程序是按照指令的顺序逐条执行的，但实际上程序不可能总是顺序执行，而经常需要改变程序的执行流程，转移到所要求的目标地址去执行，这就必须安排一条程序转移类指令(程序控制类指令的一种)来完成。

【经典总结】程序控制类指令就是专门用来控制程序执行顺序的，包括无条件转移、条件转移、循环控制及中断控制4种类型。

2. 在单地址指令中，有两个参加运算的数，指令中给出一个操作数的地址，另一个操作数的地址需采用( )。

- A. 堆栈寻址方式      B. 立即寻址方式
- C. 隐含寻址方式      D. 间接寻址方式

C【解析】采用隐含寻址可以简化地址结构，减少指令中的显式地址数。在单地址指令中，若有两个参加运算的数，指令中给出了一个操作数的地址，另一个操作数的地址通常采用隐含寻址的方式。



3. 直接转移指令JMP，其功能是将指令中的地址部分送入( )。

- A. ACC累加器
- B. PC程序计数器
- C. MR地址寄存器
- D. DR数据缓冲寄存器

**B【解析】**无条件转移指令的功能是使程序无条件地转移到指令指定的地址去执行，分为无条件转移(JMP)、调用过程(CALL)及从过程返回(RET)3种指令格式。

JMP指令可以使程序无条件地转移到目标标号指定的地址去执行。

**【经典总结】**CPU要执行的下一条指令地址都是从PC获取的，无条件转移指令执行完毕之后，应该能将下一条指令的地址放到PC。那么，JMP指令的功能应该是将指令中的地址部分送入PC，否则当前的JMP指令执行完毕，程序无法跳转到要转向的程序位置执行。

4. 零地址运算指令在指令格式中不给出操作数地址，它的操作数来自( )。

- A. 立即数和栈顶
- B. 暂存器
- C. 栈顶和次栈顶
- D. 累加器

**C【解析】**零地址指令只给出了操作码，没有显式地址码部分。这种指令通常有以下两种可能。

(1)不需要操作数，主要用来完成某种控制功能，如暂停指令(HLT)、空操作指令(NOP)、开中断指令(STD)等。

(2)零地址的算术逻辑类指令是用于堆栈计算机中的，堆栈计算机没有一般计算机中必备的通用寄存器，因此堆栈成为提供操作数和保存运算结果的唯一场所。通常，参加算术逻辑运算的两个操作数隐含地从堆栈顶部弹出，送到运算器中进行运算，运算的结果再隐含地压入堆栈。

5. 下列说法中( )是正确的。

- A. 加法指令的执行周期一定要访存
- B. 加法指令的执行周期一定不访存
- C. 指令的地址码给出存储器地址的加法指令, 在执行周期一定访存
- D. 指令的地址码给出存储器地址的加法指令, 在执行周期不一定访存

C【解析】加法指令的操作数可以在寄存器或累加器中, 在执行周期不一定要访存, 所以A、B选项均不正确。若加法指令的地址码部分给出了存储器地址, 说明有操作数在内存中, 在指令执行阶段为了获取操作数, 必须访问内存。所以C选项正确。

6. 一条指令中包含的信息有( )。

- A. 操作码、控制码 B. 操作码、向量地址
- C. 操作码、地址码 D. 控制码、地址码

**C【解析】**通常一条指令包含操作码和地址码两个部分，操作码指出该指令应该执行什么性质的操作和具有何种功能，地址码指出操作数所在的寄存器地址或存储器地址。故而，本题选择C选项。

启航教育

7. 在一地址格式的指令中, 下列( )是正确的。

- A. 仅有一个操作数, 其地址由指令的地址码提供
- B. 可能有一个操作数, 也可能有两个操作数
- C. 一定有两个操作数, 另一个是隐含的
- D. 指令的地址码字段存放的一定是操作码

B【解析】地址指令, 即只有一个地址码的指令。但是, 一地址指令未必只有一个操作数, 另一个操作数可以隐含指定, 如累加器ACC中。

若只有一个操作数, 一地址指令给出的一个地址是操作数的地址, 通常也是运算结果的存放地址, 如加1、减1、移位等单操作数指令。若不止一个操作数, 则指令中提供的是其中一个操作数的地址, 另一个操作数由机器内硬件存储器“隐含”地自动提供, 如乘法、除法指令。

8. 在二地址指令中( )是正确的。

- A. 指令的地址码字段存放的一定是操作数
- B. 指令的地址码字段存放的一定是操作数地址
- C. 运算结果通常存放在其中一个地址码所提供的地址中
- D. 指令的地址码字段存放的一定是操作码

**C【解析】**通常算术逻辑运算指令需要两个操作数。二地址指令给出了源操作数和目的操作数的地址，并规定将操作结果放回目的操作数的地址位置，通常也将二地址指令称为双操作数指令。

谢谢大家

启航教育 刘财政

### 考点三：寻址方式

启航教育 刘财政



## 考点框架



寻址方式概述



数据寻址



指令寻址

启航教育

刘财政

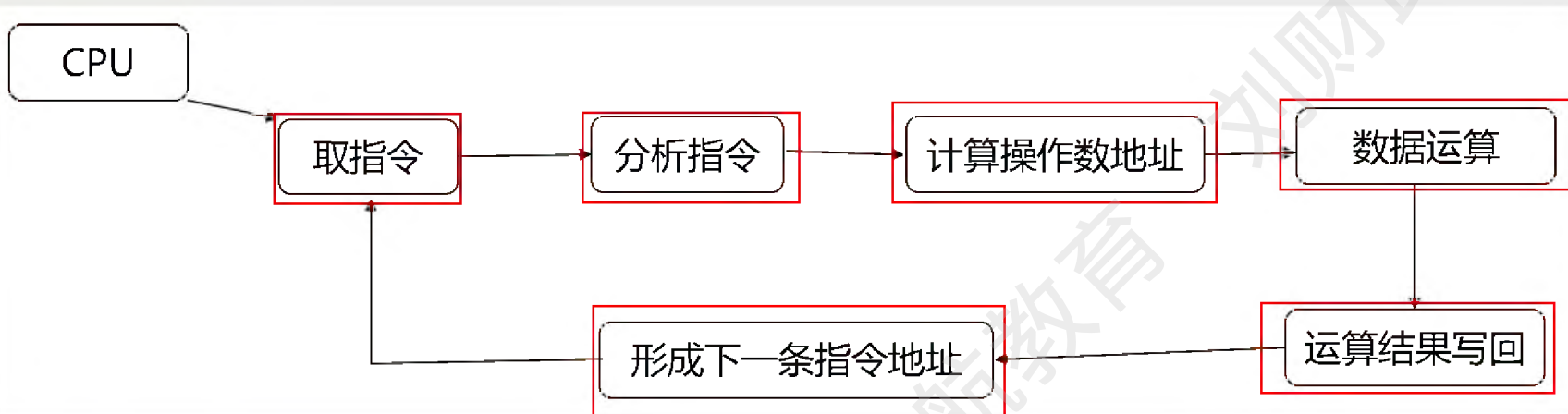
## 考点三：寻址方式

### 寻址方式概述

启航教育 刘财政

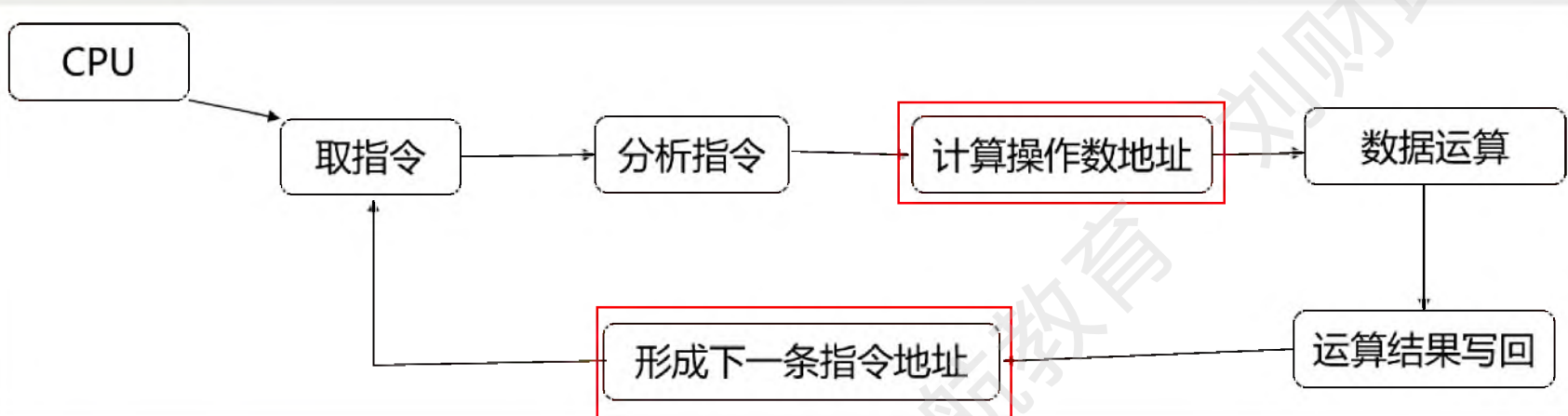
### 考点三：寻址方式

CPU给出指令的地址，通过访存得到指令，分析指令功能，计算操作数的地址，从该地址中读出数据，再进行数据运算，并把运算结果写回，最后形成下一条指令的地址，以便于继续执行指令。



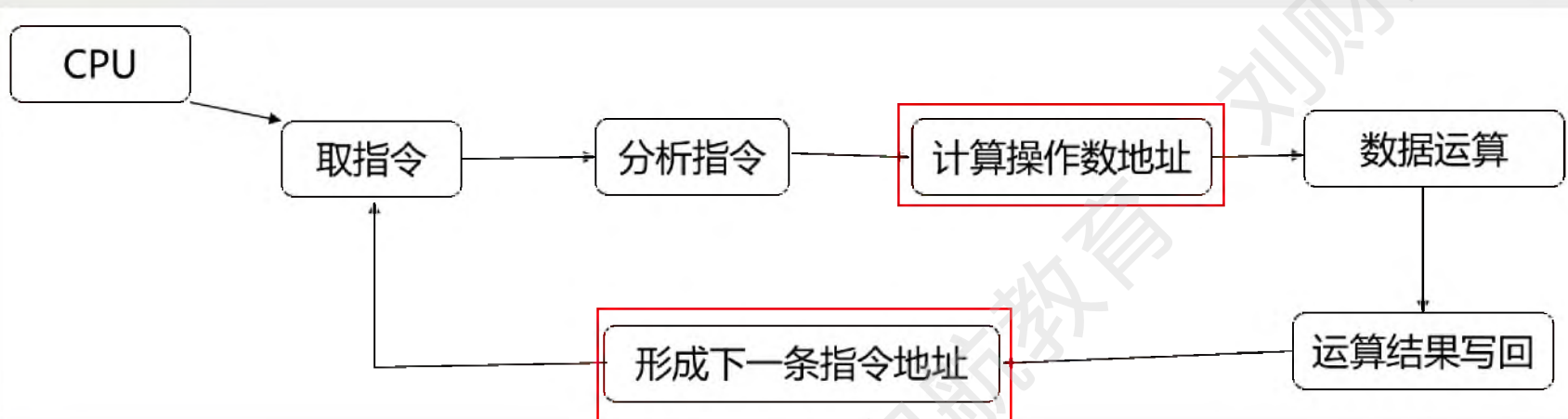
### 考点三：寻址方式

分析指令是CPU的功能之一，而数据运算依赖于分析指令，得到指令的功能而后控制ALU进行运算。那么问题是如何计算操作数的地址，如何形成下一条指令的地址呢？这就是**寻址**的应用背景。



## 考点三：寻址方式

- 寻址方式是确定本条指令的数据地址以及下一条要执行的指令地址的方法，它与硬件结构紧密相关，而且直接影响指令格式和指令功能。
- 寻址方式分为指令寻址和数据寻址两大类。



## 考点三：寻址方式

约定几个概念和表达方式：

- (1) 指令中的操作数字段提供的值为形式地址，记做A。
- (2) 按照特定的寻址方式计算得到的地址称为有效地址，记做EA。
- (3) (X) 表示获取以X地址的内存中的数据，或者以X为编号的寄存器中的值。
- (4) 假定指令字长=存储字长=机器字长，这里的假定是为了方便讲解，事实上，我们知道他们之间没有必然联系。

## 考点三：寻址方式

数据寻址

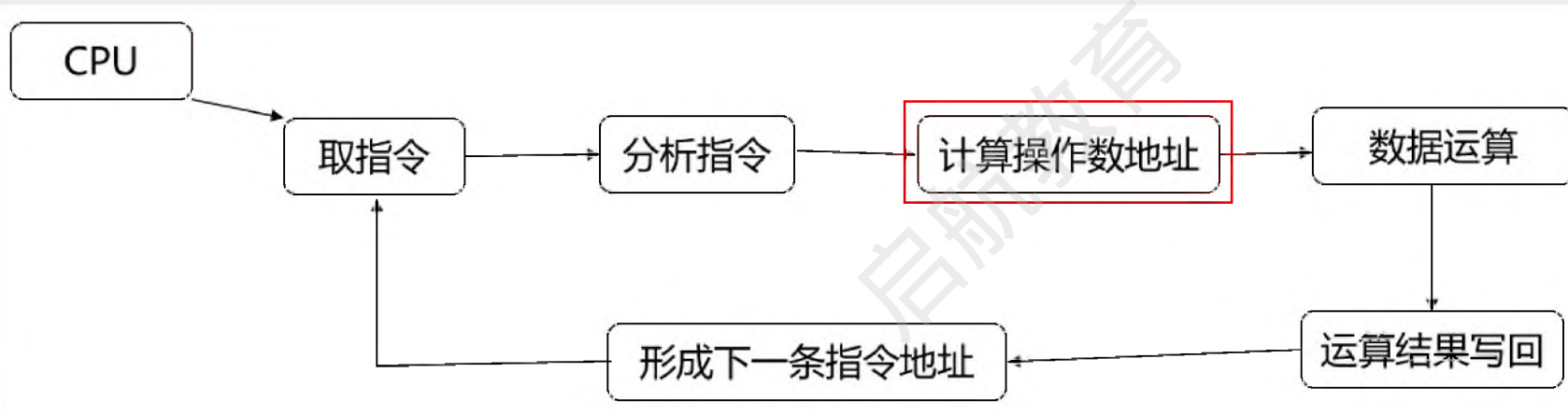
启航教育 刘财政



## 考点三：寻址方式

### 数据寻址

- 所谓数据寻址是根据指令分析的结果，确定本条指令中所有**操作数的地址**，并通过这个地址寻找指令要操作的数据的方法。
- 数据寻址的方法一共有10种，这10种方法大家务必要区分清楚每一种**寻址方式的特点、形式地址、有效地址和数据**。





# 考点三：寻址方式

## 数据寻址

假设：A=地址字段值，R=寄存器编号，  
EA=有效地址，(X)=X中的内容



方式	算法	主要优点	主要缺点
立即	操作数=A	指令执行速度快	操作数幅值有限
直接	EA=A	有效地址计算简单	地址范围有限
间接	EA=(A)	有效地址范围大	多次存储器访问
寄存器	操作数=(R)	指令执行快，指令短	地址范围有限
寄间接	EA=(R)	地址范围大	额外存储器访问
偏移	EA=A+(R)	灵活	复杂
堆栈	EA=栈顶	指令短	应用有限

## 考点三：寻址方式

### 数据寻址

操作码	寻址特征	形式地址 A
-----	------	--------

形式地址                  指令字中的地址

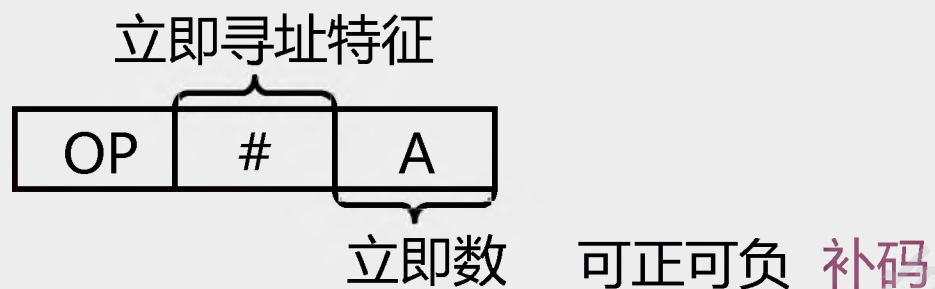
有效地址                  操作数的真实地址

约定                  指令字长 = 存储字长 = 机器字长

## 考点三：寻址方式

### 立即寻址

形式地址 A 就是操作数



指令的操作数字段给出的是**操作数本身**（**操作数在指令中**）。数据随指令由内存读入CPU,此时**形式地址A**就是**数据本身**，没有有效地址。

## 考点三：寻址方式

### 立即寻址

形式地址 A 就是操作数



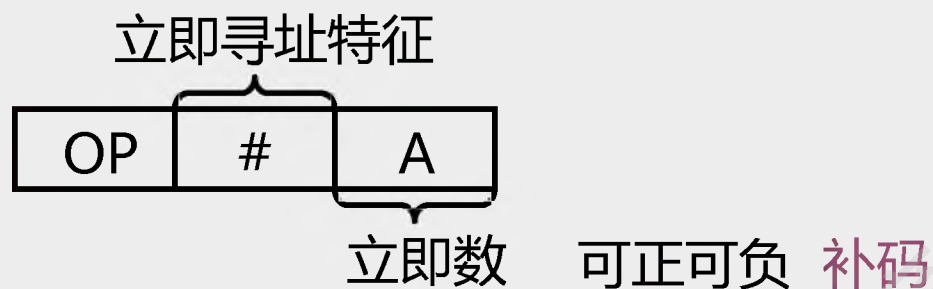
立即数 可正可负 补码

好处：立即数寻址的优点是简单、明确、指令执行阶段不访存，速度快

## 考点三：寻址方式

### 立即寻址

形式地址 A 就是操作数



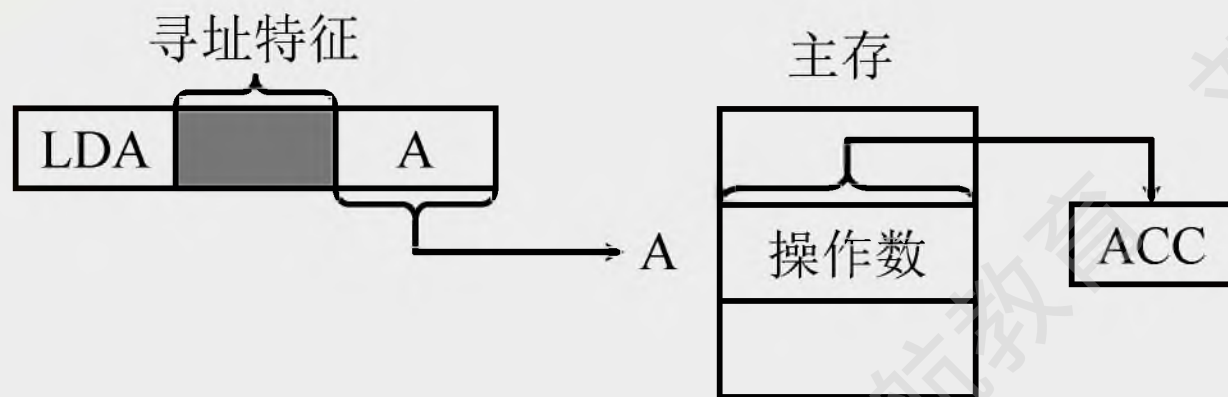
坏处:因为操作数是指令的一部分, **不能被修改**, 而且立即数的大小受到指令长度的限制, 所以这种寻址方式**灵活性最差**, 通常用于给某一寄存器或主存单元赋初值或提供一个常数。

## 考点三：寻址方式

### 🕒 直接寻址

$$EA = A$$

有效地址由形式地址直接给出



好处:

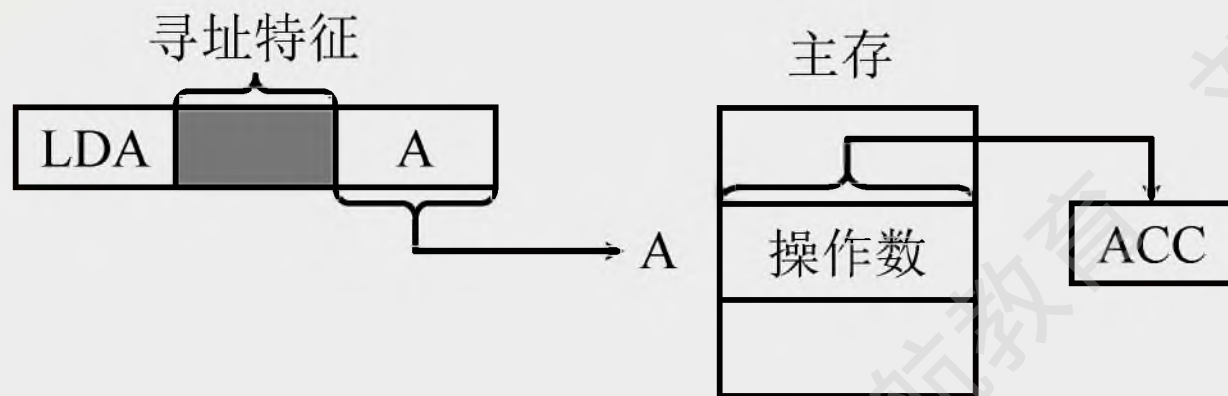
- 执行阶段访问一次存储器
- 这种寻址方式不需作任何寻址运算，简单直观，也便于硬件实现

## 考点三：寻址方式

### 🕒 直接寻址

$$EA = A$$

有效地址由形式地址直接给出



坏处:

- A 的位数决定了该指令操作数的寻址范围
- 操作数的地址不易修改（必须修改A）

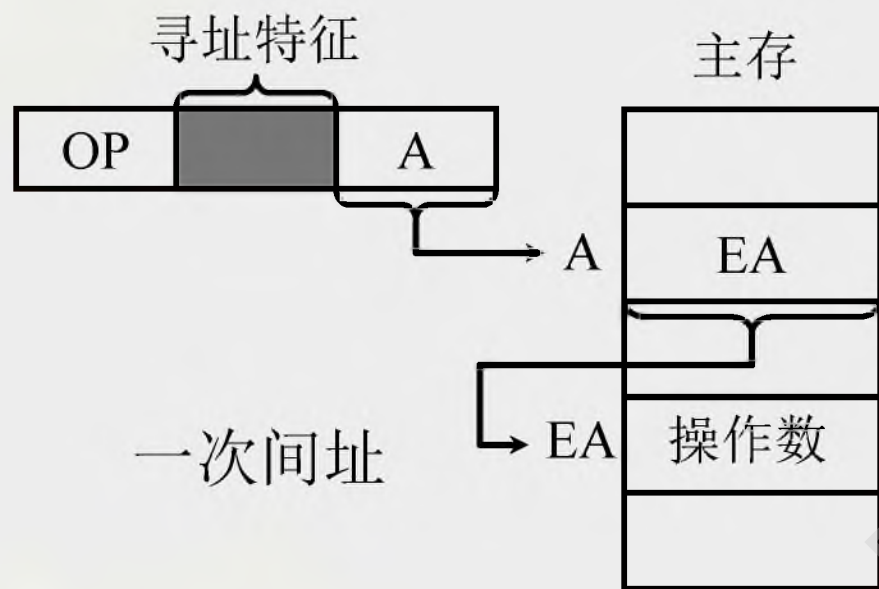


## 考点三：寻址方式

### 🕒 间接寻址

$$EA = (A)$$

有效地址由形式地址间接提供



在一级间接寻址中

□  $EA = (A)$

□  $S = ((A))$

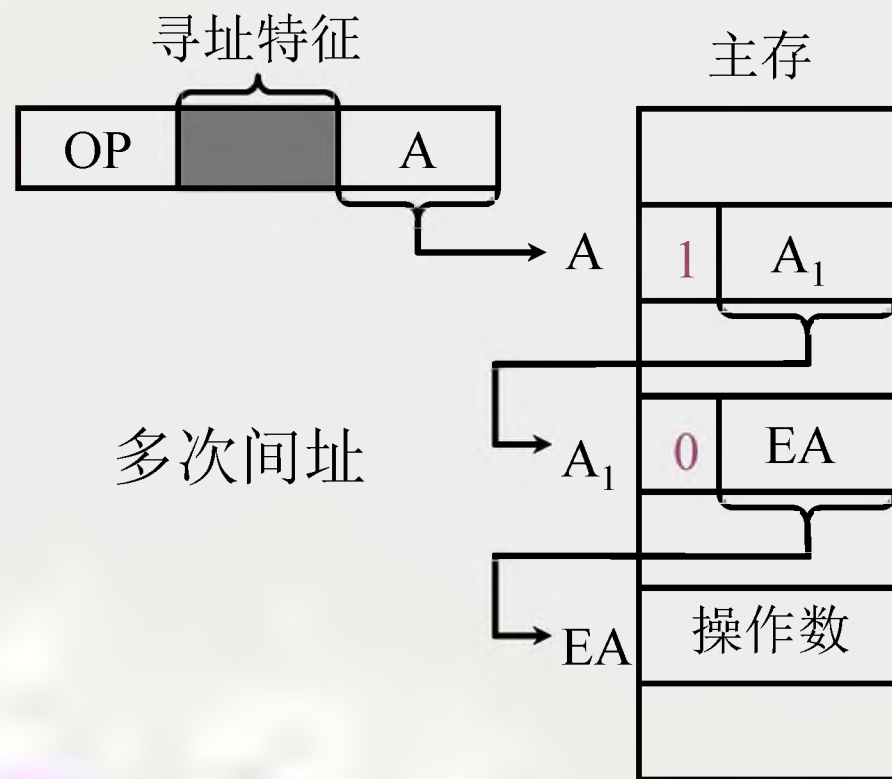


## 考点三：寻址方式

### 🕒 间接寻址

$$EA = (A)$$

有效地址由形式地址间接提供

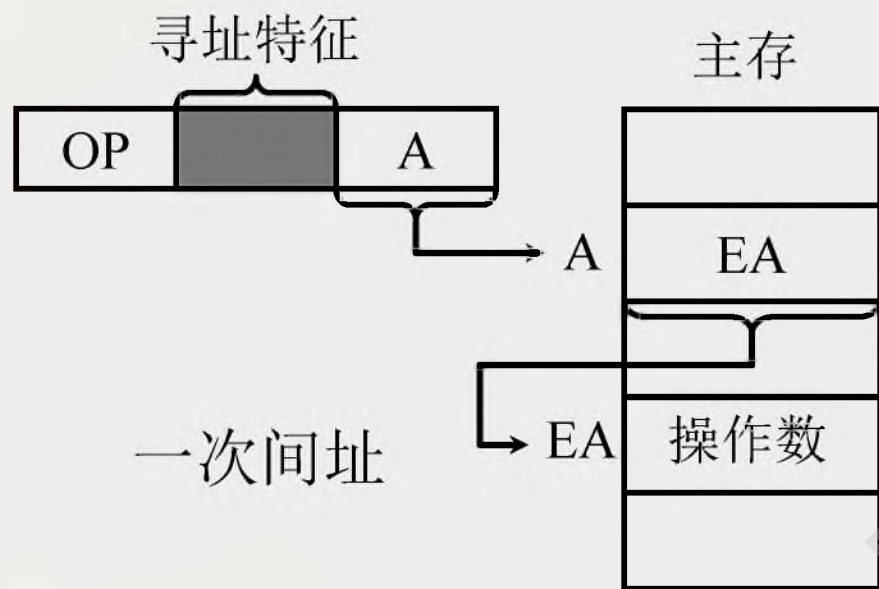


## 考点三：寻址方式

### 🕒 间接寻址

$$EA = (A)$$

有效地址由形式地址间接提供



在一级间接寻址中

□  $EA = (A)$

□  $S = ((A))$

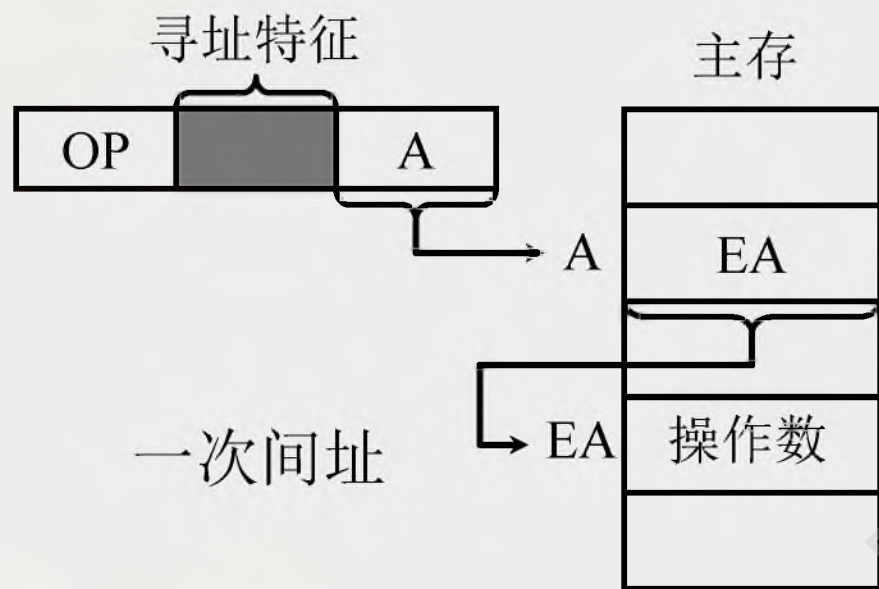
好处:间接寻址的优点是由于操作数在内存, 操作数所占的内存单元数由数据的类型决定, 所以数据的内存地址以及范围可以较大

## 考点三：寻址方式

### 🕒 间接寻址

$$EA = (A)$$

有效地址由形式地址间接提供



在一级间接寻址中

□  $EA = (A)$

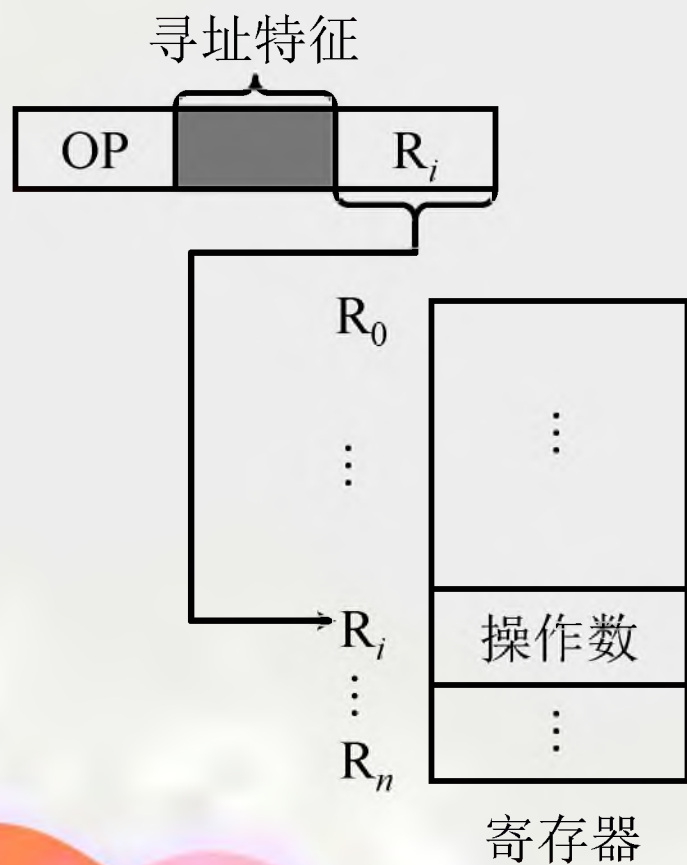
□  $S = ((A))$

坏处:间接寻址的缺点是读写操作数需要多次读写内存, 会导致系统速度变慢。

## 考点三：寻址方式

### 🕒 寄存器寻址

$EA = R_i$       有效地址即为寄存器编号

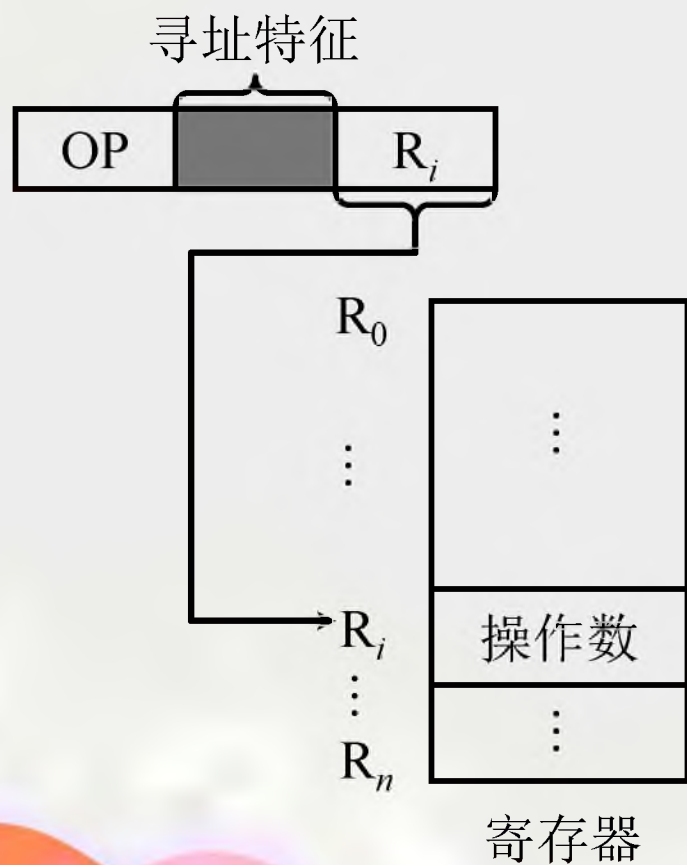


好处:寄存器寻址的优点是由于操作数在CPU中, 取数速度较快;由于CPU中寄存器的数量是有限的所以需要的寄存器编号也较短, 可以有效减少操作数字段的位数。

## 考点三：寻址方式

### 🕒 寄存器寻址

$EA = R_i$       有效地址即为寄存器编号



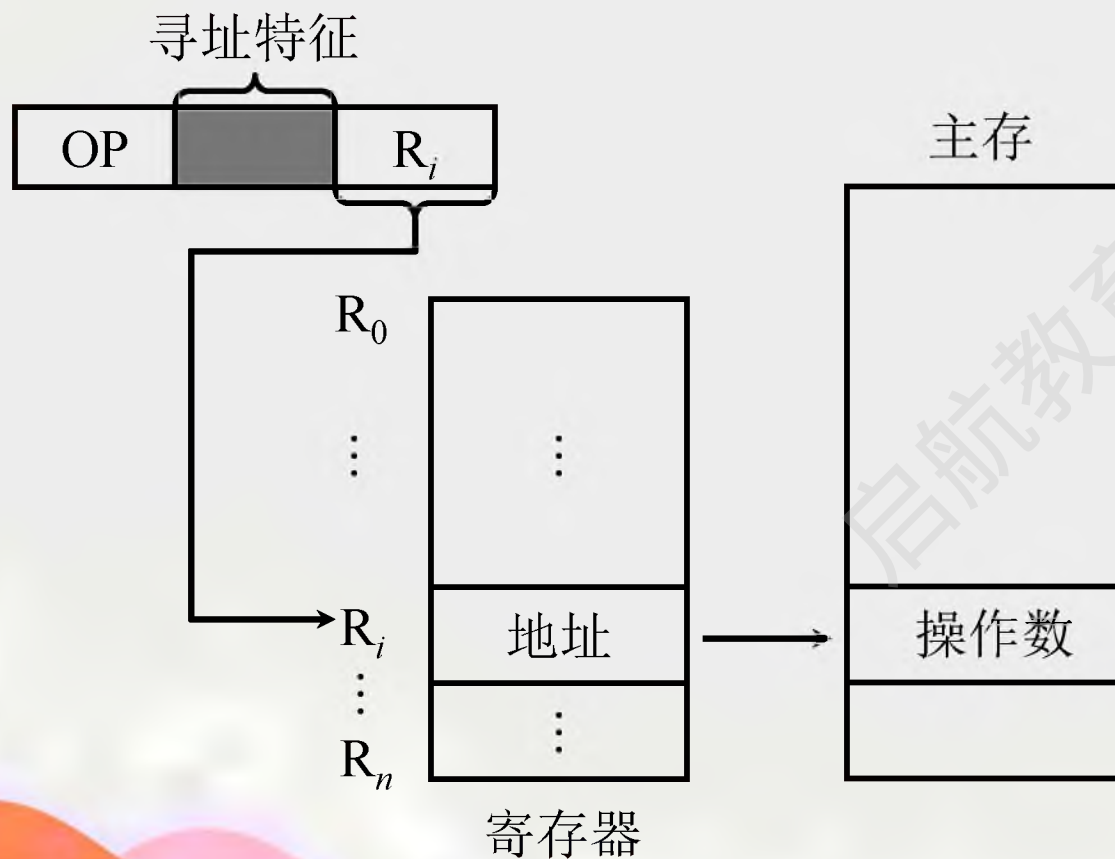
坏处:寄存器寻址的缺点是由于CPU中寄存器的数量是有限的,所以同时进行寄存器寻址的操作数个数是有限的。

## 考点三：寻址方式

### 🕒 寄存器间接寻址

$$EA = (R_i)$$

有效地址在寄存器中



好处:寄存器间接寻址的优点是由于操作数的地址在CPU中获得操作数内存地址的速度较快。由于CPU中寄存器的数量是有限的，所以需要的寄存器编号也较短可以有效减少操作数字段的位数。



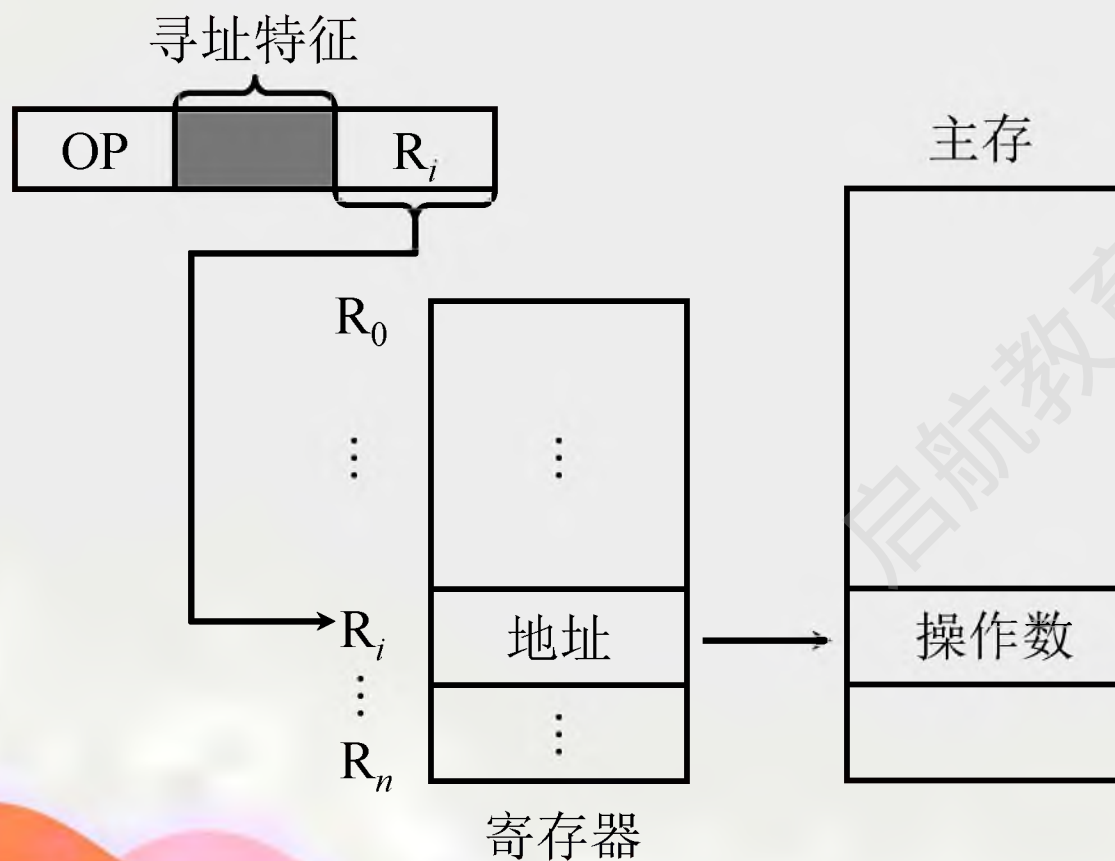
## 考点三：寻址方式



### 寄存器间接寻址

$$EA = (R_i)$$

有效地址在寄存器中



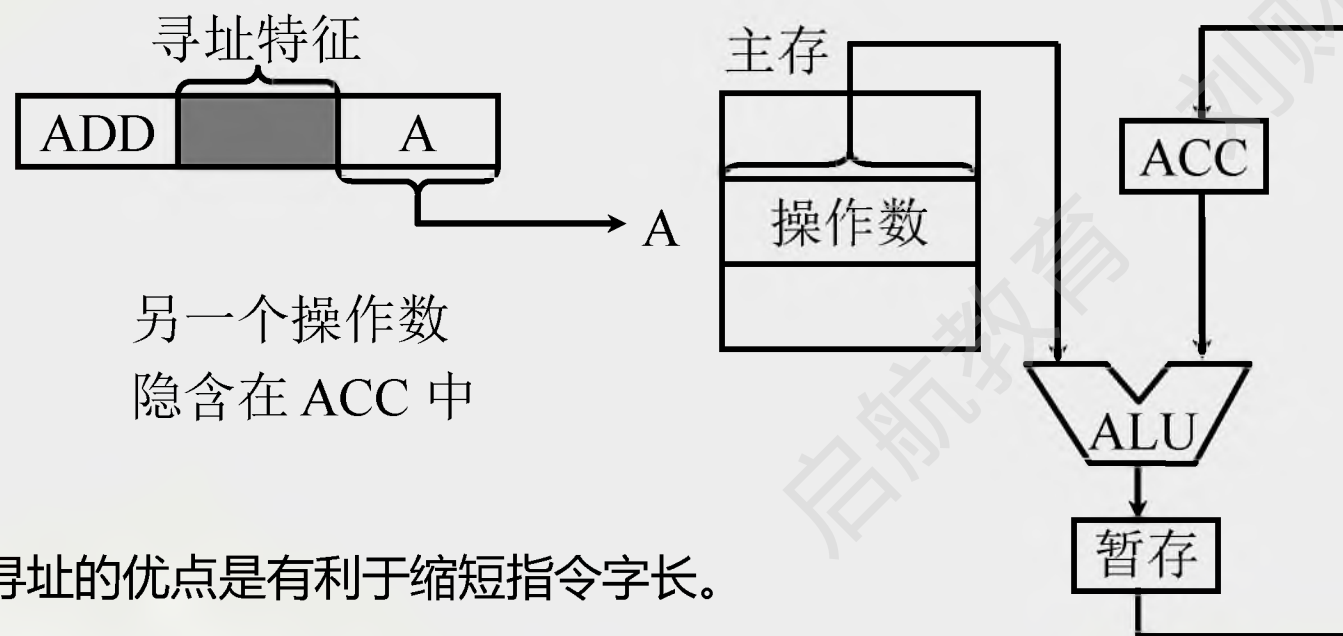
坏处:寄存器间接寻址的缺点是由于CPU中寄存器的数量是有限的,所以同时进行寄存器间接寻址的操作数个数是有限的;并且同寄存器寻址方式相比,由于需要读内存才能得到操作数,速度比寄存器寻址慢。

## 考点三：寻址方式



### 隐含寻址

操作数地址隐含在操作码中



隐含寻址的优点是有利于缩短指令字长。

隐含寻址的缺点是需增加存储操作数或隐含地址的硬件

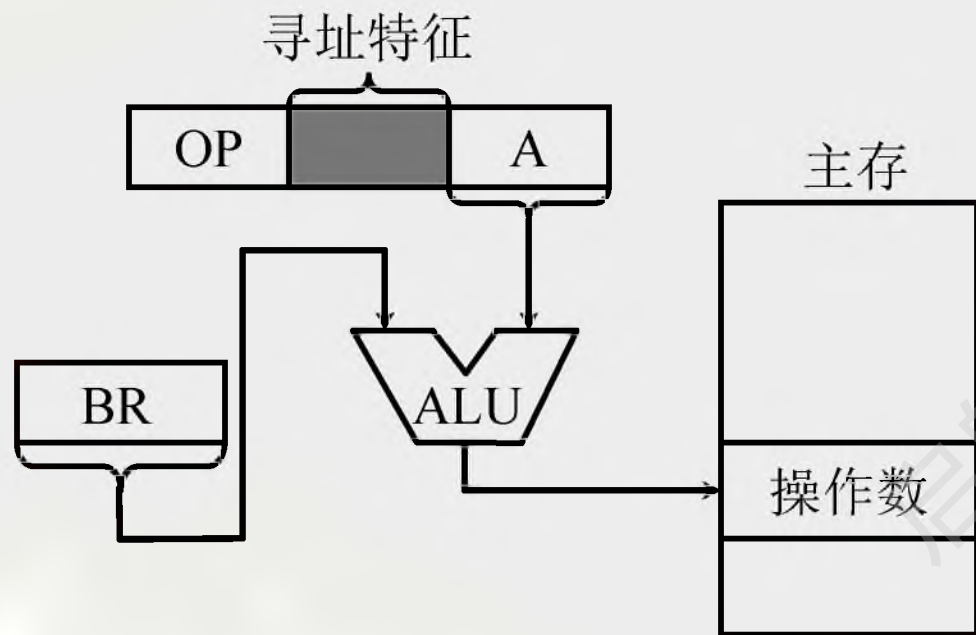


## 考点三：寻址方式

### 🕒 基址寻址

$$EA = (BR) + A$$

BR 为基址寄存器



- 可扩大寻址范围
- 有利于多道程序
- BR 内容由操作系统或管理程序确定
- 在程序的执行过程中 BR 内容不变，形式地址 A 可变

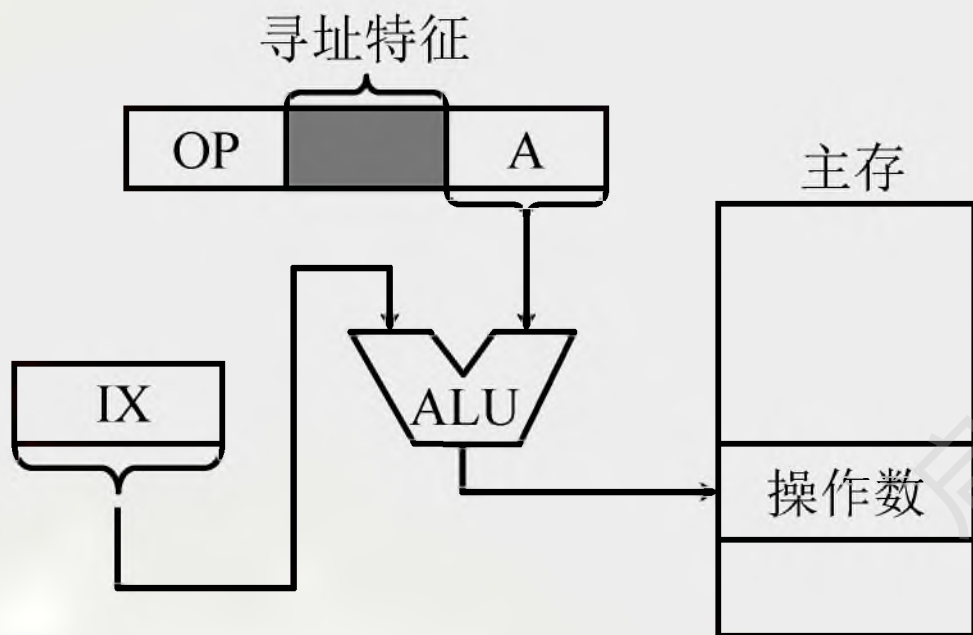
## 考点三：寻址方式

### 🕒 变址寻址

$$EA = (IX) + A$$

IX 为变址寄存器（专用）

通用寄存器也可以作为变址寄存器



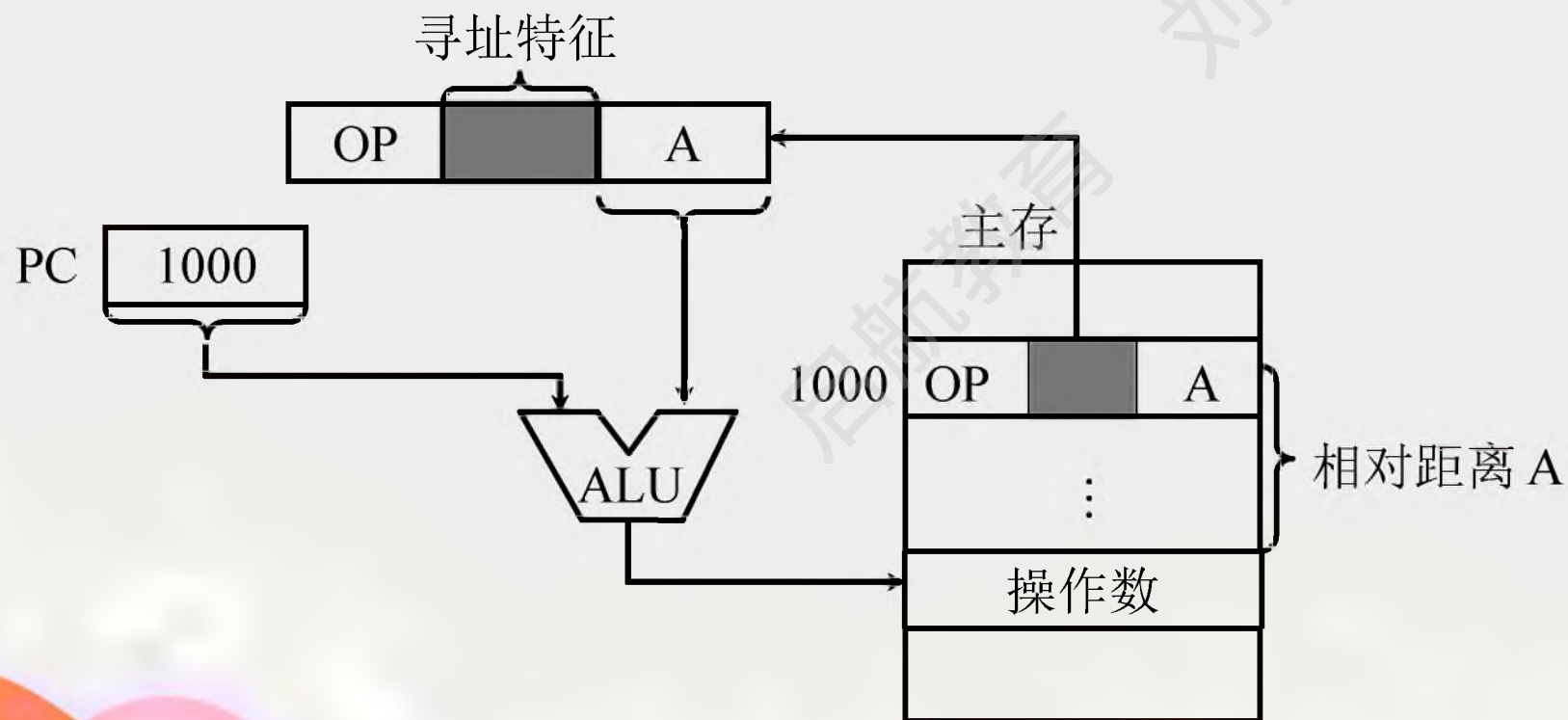
- 可扩大寻址范围
- IX 的内容由用户给定
- 在程序的执行过程中 IX 内容可变，形式地址 A 不变
- 便于处理数组问题

## 考点三：寻址方式

### 🕒 相对寻址

$$EA = (PC) + A$$

A 是相对于当前指令的位移量（可正可负，补码）



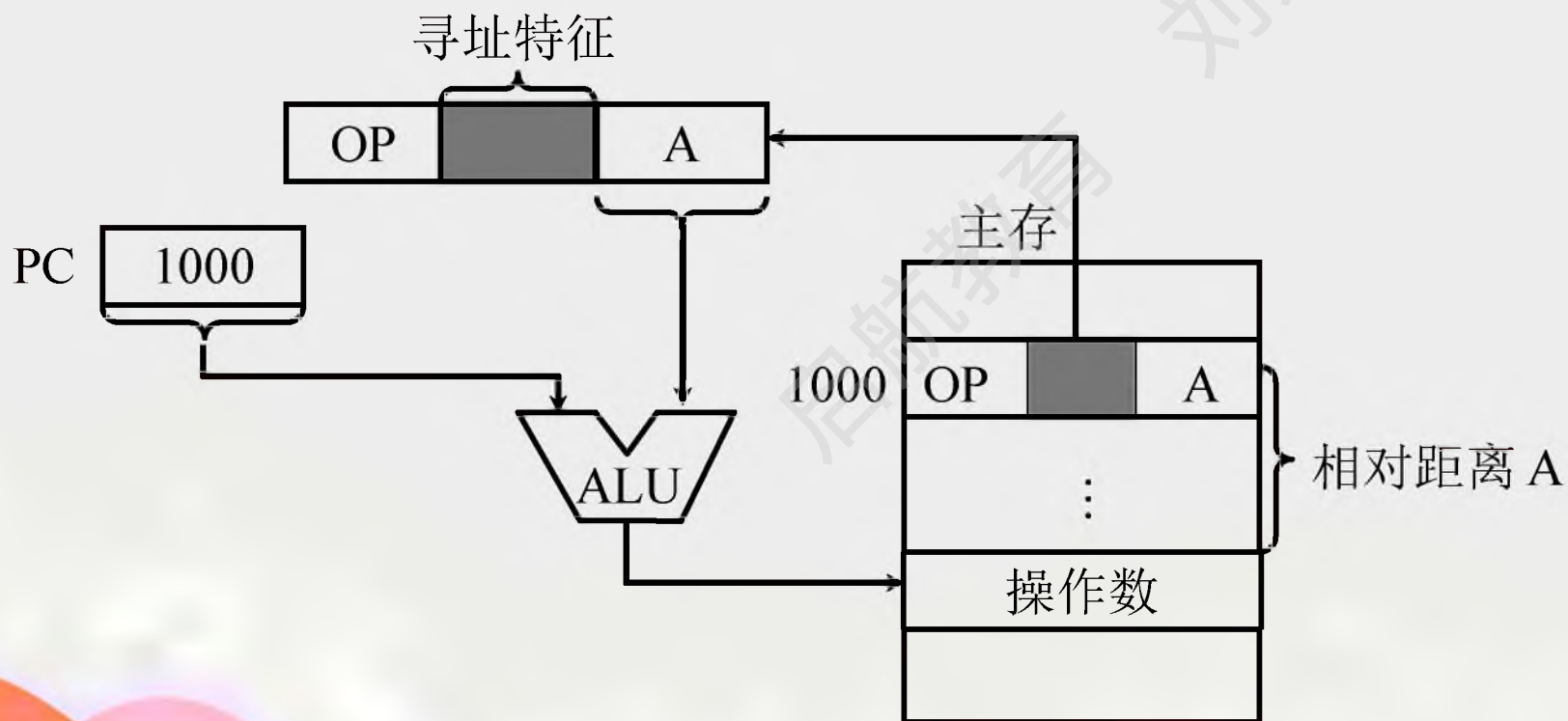
## 考点三：寻址方式

### 🕒 相对寻址

$$EA = (PC) + A$$

A 是相对于当前指令的位移量（可正可负，补码）

- A 的位数决定操作数的寻址范围
- 程序浮动
- 广泛用于转移指令



## 考点三：寻址方式



堆栈寻址  
堆栈

硬堆栈  
软堆栈

多个寄存器

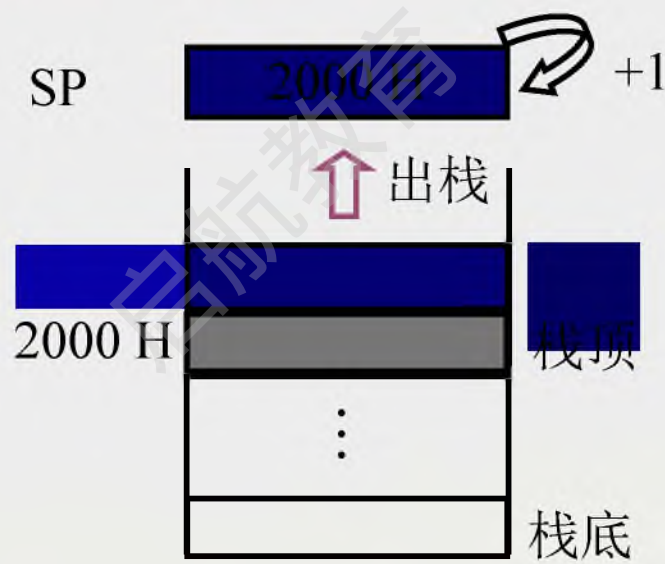
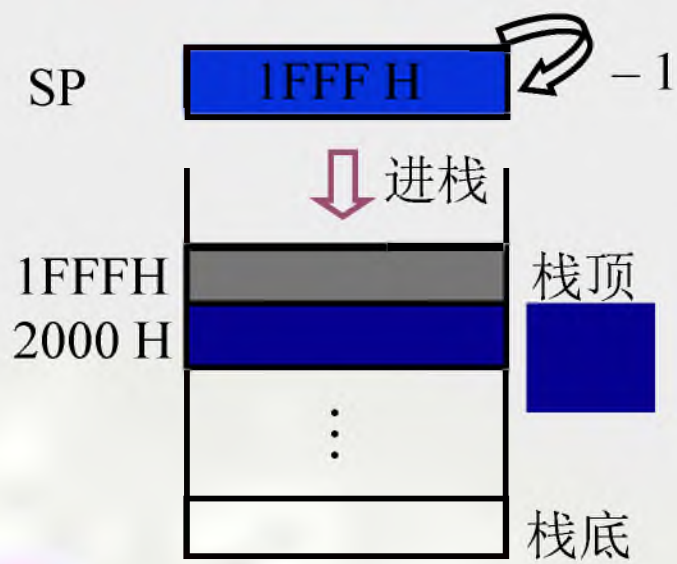
指定的存储空间

先进后出（一个入出口）

进栈  $(SP) - 1 \rightarrow SP$

栈顶地址 由 SP 指出

出栈  $(SP) + 1 \rightarrow SP$





### 考点三：寻址方式

寻址方式	有效地址	访存次数
隐含寻址	程序指定	0
立即寻址	A即是操作数	0
直接寻址	$EA=A$	1
一次间接寻址	$EA=(A)$	2
N次间接寻址	$EA((((A))))$	N+1
寄存器寻址	$EA=R_i$	0
寄存器间接一次寻址	$EA=(R_i)$	1
相对寻址	$EA=(PC)+A$	1
基址寻址	$EA=(BR)+A$	1
变址寻址	$EA=(IX)+A$	1
堆栈寻址	硬堆栈	0
堆栈寻址	软堆栈	1

## 考点三：寻址方式

但是这里会有一个比较深刻的问题，这10种寻址方式，如何来区分到底是哪一种寻址方式呢？其实在寻址方式中，有寻址特征，会明确告知是哪一种寻址方式

操作码	寻址特征	形式地址 A
-----	------	--------

形式地址                  指令字中的地址

有效地址                  操作数的真实地址

约定                  指令字长 = 存储字长 = 机器字长

**【政哥点拨】**

启航教育 刘财政



1. **基址寄存器**的内容为3000H, **变址寄存器**的内容为02B0H, 指令的**地址码**为002BH, **程序计数器**(存放当前正在执行的指令的地址)的内容为4500H, 且存储器内存放的内容如表所示。

地址	002BH	302BH	32B0H	32DBH	3500H	452BH
内容	3500H	3500H	5600H	2800H	2600H	2500H

(1)若采用基址寻址方式, 则取出的操作数是什么?

(1)采用基址寻址方式, 有效地址EA等于基址寄存器的内容加上形式地址。本题中, 基址寄存器的内容为3000H, 指令的地址码为002BH, 所以操作数的有效地址为302BH。查表可知, 地址302BH的内容是3500H, 即操作数是3500H。

1. **基址寄存器**的内容为3000H, **变址寄存器**的内容为02B0H, 指令的**地址码**为002BH, **程序计数器**(存放当前正在执行的指令的地址)的内容为4500H, 且存储器内存放的内容如表所示。

地址	002BH	302BH	32B0H	32DBH	3500H	452BH
内容	3500H	3500H	5600H	2800H	2600H	2500H

(2)若采用变址寻址(考虑基址)方式, 则取出的操作数是什么?

(2)采用变址寻址方式, 有效地址 $EA = \text{基址寄存器的内容} + \text{变址寄存器的内容} + \text{偏移量} = 3000H + 02B0H + 002BH = 32DBH$ 。查表可知, 该存储单元的内容(操作数)是2800H。

1. **基址寄存器**的内容为3000H, **变址寄存器**的内容为02B0H, 指令的**地址码**为002BH, **程序计数器**(存放当前正在执行的指令的地址)的内容为4500H, 且存储器内存放的内容如表所示。

地址	002BH	302BH	32B0H	32DBH	3500H	452BH
内容	3500H	3500H	5600H	2800H	2600H	2500H

(3)若采用立即寻址方式, 则取出的操作数是什么?

(3)采用立即寻址方式, 指令的形式地址字段的值就是操作数本身, 所以操作数为002BH。

1. **基址寄存器**的内容为3000H, **变址寄存器**的内容为02B0H, 指令的**地址码**为002BH, **程序计数器**(存放当前正在执行的指令的地址)的内容为4500H, 且存储器内存放的内容如表所示。

地址	002BH	302BH	32B0H	32DBH	3500H	452BH
内容	3500H	3500H	5600H	2800H	2600H	2500H

(4)若采用存储器间接寻址(不考虑基址)方式, 则取出的操作数是什么?

(4)采用存储器间接寻址方式, 指令的地址码字段给出的是操作数的有效地址, 而不是操作数本身。查表可知, 操作数的地址为3500H, 存放在该物理位置的操作数为2600H。

1. **基址寄存器**的内容为3000H, **变址寄存器**的内容为02B0H, 指令的**地址码**为002BH, **程序计数器**(存放当前正在执行的指令的地址)的内容为4500H, 且存储器内存放的内容如表所示。

地址	002BH	302BH	32B0H	32DBH	3500H	452BH
内容	3500H	3500H	5600H	2800H	2600H	2500H

(5)若相对寻址用于转移指令, 则转移地址是多少?

(5)采用相对寻址的方式, 指令的有效地址为PC的值加上形式地址, 故而  
 $EA = 4500H + 002BH = 452BH$ 。

2. 【航天系统, 2017】某机主存容量为4 M×16位, 且存储字长等于指令字长, 若该机能完成**97种操作**, **操作码位数固定**, 且其有直接、间接、基址、变址、相对、立即**6种寻址方式**, 则相对寻址指令的偏移量为( )。

A. -32 ~ +31    B. -64 ~ +63    C. -128 ~ +127    D. -256 ~ +255

A【解析】题目中操作码固定, 且该机需要完成97种操作, 那么操作码的位数是 $\log_2 97=7$ 位。具有直接、间接、基址、变址、相对、立即6种寻址方式, 所以寻址方式(特征)是 $\log_2 6=3$ 位(这三位就是指令的寻址特征位, 通过这个寻址特征位的取值, 可以确定指令的寻址方式)。指令字长是16位, 那么地址码是 $16-3-7=6$ (位), 所以偏移量是-32 ~ +31。

OP (7位)	M (3位)	A (6位)
---------	--------	--------



3. 【航天系统，2017】某计算机功能完成120种操作，CPU有8个通用寄存器(12位)，主存容量为16 K字。采用寄存器-存储器类型指令，欲使指令可任意访问主存任意地址，指令字长应取( )位。

A. 12      B. 14      C. 22      D. 24

D【解析】计算机功能完成120种操作，共需要 $\log_2 120=7$ 位，通用寄存器有8个，那么寄存器编号是3位。由于主存容量为16 K字，欲使指令可任意访问主存任意地址，那么地址码的位数至少是 $\log_2 16\text{ K}=14$ 位，所以指令字长至少是 $14+7+3=24$ (位)。

OP (7位)	M (3位)	A (14位)
---------	--------	---------

## 考点三：寻址方式

指令寻址

启航教育 刘财政



## 考点三：寻址方式

- **相对寻址**指的是下一条将要执行的指令所在的内存地址由CPU内的程序计数器PC的内容（直接用或加1后使用）和指令中给出的偏移量求和得到，注意偏移量为有符号数，用补码表示。
- 特点是可实现循环和选择结构。指令寻址分为**顺序寻址**和**跳跃寻址**两种。

## 考点三：寻址方式

□ 顺序寻址可以通过程序计数器PC加1自动形成下一条指令的地址。



## 考点三：寻址方式

- 跳跃寻址则通过转移类指令实现。
- 所谓跳跃，是指下条指令的地址码不是由程序计数器PC给出，而是由本条指令给出下条指令地址的计算方式。是否跳跃可能受到状态寄存器和操作数的控制，而跳跃到的地址分为绝对地址（由标记符直接得到）和相对地址（对于当前指令地址的偏移量），跳跃的结果是当前指令修改PC值，所以下一条指令仍然是通过程序计数器PC给出

## 考点三：寻址方式

□ 跳跃寻址则通过转移类指令实现。



**【政哥点拨】**

启航教育 刘财政

1、某机器字长为16 位，主存按字节编址，转移指令采用相对寻址，由两个字节组成，第一字节为操作码字段，第二字节为相对位移量字段。假定取指令时，每取一个字节 PC自动加 1。若某转移指令所在主存地址为 2000H，相对位移量字段的内容为 06H，则该转移指令成功转移后的目标地址是（ ）。

A. 2006H                      B. 2007H                      C. 2008H   D. 2009H

C【解析】主存按字节编址，取指令时，每取一个字节PC自动加 1。由于转移指令由两个字节组成，取出这条转移指令之后的PC值等于2002H，所以转移指令成功转移后的目标地址为 $PC = 2000H + 2H + 06H = 2008H$ 。

此题容易误选A或B。原因是没有考虑PC值的自动更新，或虽然考虑了PC要自动更新，但没有注意到这条转移指令是一条两个字节的指令，PC值仅仅加1而不是加2。



2、堆栈寻址方式中，设A为累加器，SP为堆栈指示器，MSP为SP指示的栈顶单元，如果进栈操作的动作顺序是 $(A) \rightarrow MSP$ ， $(SP)-1 \rightarrow SP$ ，那么出栈操作的动作顺序应为（ ）。

- A.  $(MSP) \rightarrow A$ ， $(SP)+1 \rightarrow SP$     B.  $(SP)+1 \rightarrow SP$ ， $(MSP) \rightarrow A$   
C.  $(SP)-1 \rightarrow SP$ ， $(MSP) \rightarrow A$     D. 以上都不对

B【解析】 进、出堆栈时对栈顶指针的操作顺序是不同的。如果进栈时是先压入数据，说明栈指针是指向栈顶的空单元，所以出栈时，就要先修改栈指针，然后才能弹出数据。

3、设相对寻址的转移指令占两个字节，第一个字节是操作码，第二个字节是相对位移量（用补码表示），若每当CPU从存储器中取出一个字节时，即自动完成 $(PC)+1 \rightarrow PC$ ，设当前PC的内容为2009H，要求转移到2000H地址，则该转移指令第二字节的内容应为（ ）。

A. F5H                      B. F7H                      C. 09H    D. 00H

A【解析】因为PC的内容是2009H，每次读取一个字节，PC的内容加1，本题中，指令的长度是2个字节，因此读取指令之后，PC的值是 $2009H+2H=200BH$ 。题目中需要跳转到2000H，因此操作码的值是-BH，即地址码是-11。把-11转化为补码，得到答案A。



4、 直接、间接、立即三种寻址方式指令的执行速度，由快至慢的排序是（ ）。

- A. 直接、立即、间接
- B. 直接、间接、立即
- C. 立即、直接、间接
- D. 立即、间接、直接

C【解析】通过我们总结的表格，可以根据不同寻址方式的访存次数决定他们的速度。立即寻址最快、间接寻址最慢。

寻址方式	有效地址	访存次数
隐含寻址	程序指定	0
立即寻址	A即是操作数	0
直接寻址	$EA=A$	1
一次间接寻址	$EA=(A)$	2
N次间接寻址	$EA=(((A)))$	N+1
寄存器寻址	$EA=R_i$	0
寄存器间接一次寻址	$EA=(R_i)$	1
相对寻址	$EA=(PC)+A$	1
基址寻址	$EA=(BR)+A$	1
变址寻址	$EA=(IX)+A$	1
堆栈寻址	硬堆栈	0
堆栈寻址	软堆栈	1

5.偏移寻址通过将某个寄存器中内容与一个形式地址相加而生成有效地址。下列寻址方式中，**不属于**偏移寻址方式的是（ ）。

A. 间接寻址      B. 基址寻址      C. 相对寻址      D. 变址寻址

A【解析】间接寻址不需要寄存器， $EA=(A)$ 。

基址寻址： $EA=A+$ 基址寄存器内容；

相对寻址： $EA = A+PC$ 内容；变址寻址：

$EA = A+$ 变址寄存器内容。

后三个都是将某个寄存器内容与一个形式地址相加而形成有效地址。

6.【北京大学，2016年】假设某条指令的一个操作数采用变址寻址方式，变址寄存器的内容为8H,指令中给出的形式地址为1200H,地址为1200H的内存单元的内容为12FCH,地址为12FCH 的内存单元的内容为38B8H，则该操作数的有效地址为（ ）

A. 1200H      B.12FCH      C.1208H      D.38B8H

C【解析】变址寻址方式的有效地址是 $EA = (IX) + A$ ，变址寄存器的内容为8H,指令中给出的形式地址为1200H，所以 $EA = 1200H + 8H = 1208H$ 。

7、【航空系统，2018年】假设寄存器R中的数值为600，地址为600和 700的主存单元中存放的内容分别是700和800。则( ) 访问到的操作数为600。

I 直接寻址600

II. 寄存器间接寻址(R)

III.存储器间接寻址 (600)

IV.寄存器寻址

A.只有I      B.II、III      C.III、IV      D.只有IV

D 【解析】根据题目的条件，如果采用直接寻址，其地址是600，取得内存的数据是700；寄存器间接寻址(R)，R的值是600，通过600访问内存，取得内存的数据是700；存储器间接寻址 (600)，得到内存地址中600中的数据是700，再以700访问内存，得到数据是800。

**【牛刀小试】**

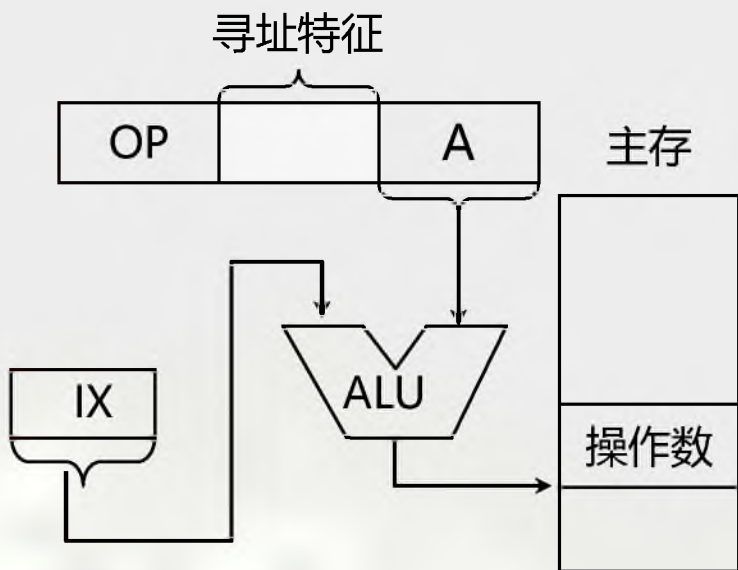
启航教育 刘财政

1. **变址寻址**时，操作数有效地址是由指令中给出的形式地址(偏移量)与( )相加得到的。

- A. 基址寄存器      B. 变址寄存器  
C. 变址值          D. 指令计数器内容

C【解析】有效地址EA等于指令字中的形式地址A与变址寄存器IX的内容相加之和

$$EA = (IX) + A \quad IX \text{ 为变址寄存器}$$



- 可扩大寻址范围
- IX 的内容由用户给定
- 在程序的执行过程中 IX 内容可变, 形式地址 A 不变
- 便于处理数组问题



2. 在指令的地址字段中，直接给出操作数本身的寻址方式，称为( )。

- A. 隐含寻址      B. 立即寻址      C. 寄存器寻址      D. 直接寻址

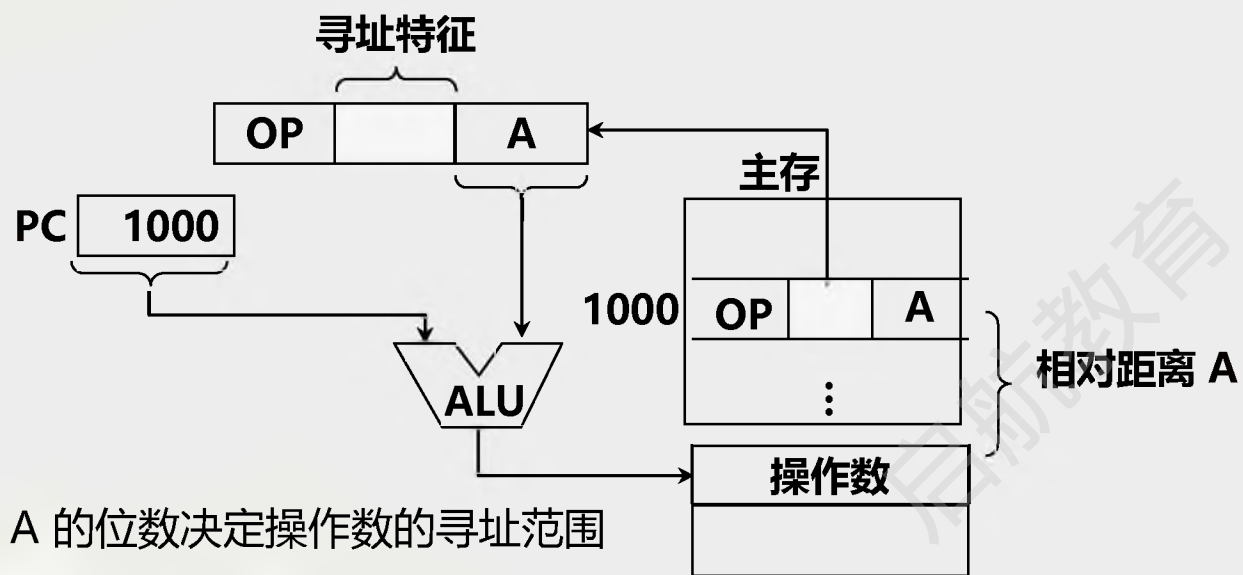
寻址方式	有效地址	访存次数
隐含寻址	程序指定	0
立即寻址	A即是操作数	0
直接寻址	$EA=A$	1
一次间接寻址	$EA=(A)$	2
N次间接寻址	$EA=(((A)))$	N+1
寄存器寻址	$EA=R_i$	0
寄存器间接一次寻址	$EA=(R_i)$	1
相对寻址	$EA=(PC)+A$	1
基址寻址	$EA=(BR)+A$	1
变址寻址	$EA=(IX)+A$	1
堆栈寻址	硬堆栈	0
堆栈寻址	软堆栈	1

**B【解析】**立即寻址的指令中地址码字段直接给出了操作数本身。也就是说，在取指令时，操作码和操作数同时被取出，不必再次访问存储器，从而提高了指令的执行速度。显然，这种寻址方式也有明显的缺点，那就是操作数是指令的一部分，不能被修改，且立即数的大小将受到指令长度的限制。

3. 设相对寻址的转移指令占两个字节，第一个字节是操作码，第二个字节是相对位移量(可正可负)，则转移的地址范围是( )。

A. 255      B. 256      C. 254      D. 0

B 【解析】  $EA = (PC) + A$      $A$  是相对于当前指令的位移量（可正可负，补码）



- $A$  的位数决定操作数的寻址范围
- 程序浮动
- 广泛用于转移指令

因为相对寻址的偏移量字段是一个字节，  
所以转移的地址范围是256。

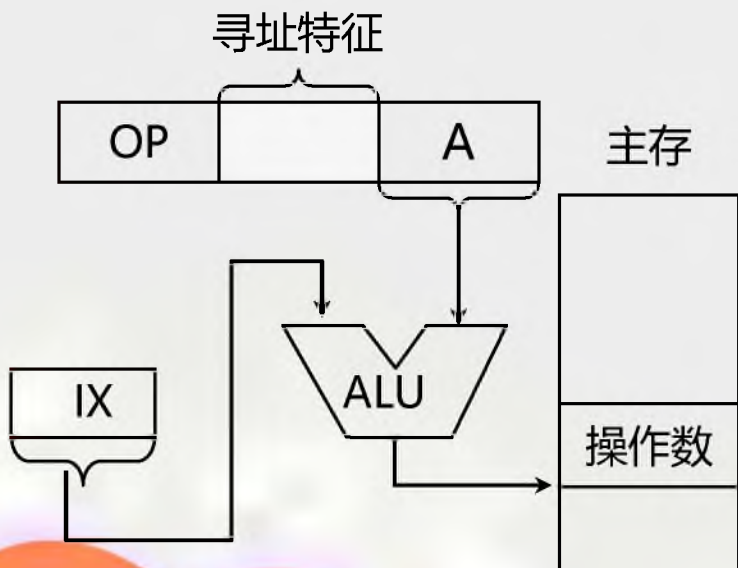


4. 采用**变址寻址**可扩大寻址范围，且( )。

- A. 变址寄存器内容由用户确定，在程序执行过程中不可变
- B. 变址寄存器内容由操作系统确定，在程序执行过程中不可变
- C. 变址寄存器内容由操作系统确定，在程序执行过程中可变
- D. 变址寄存器内容由用户确定，在程序执行过程中可变

D【解析】有效地址EA等于指令字中的形式地址A与变址寄存器IX的内容相加之和

$$EA = (IX) + A \quad IX \text{ 为变址寄存器}$$



- 可扩大寻址范围
- IX 的内容由用户给定
- 在程序的执行过程中 IX 内容可变，形式地址 A 不变
- 便于处理数组问题

4. 采用**变址寻址**可扩大寻址范围，且( )。

- A. 变址寄存器内容由用户确定，在程序执行过程中不可变
- B. 变址寄存器内容由操作系统确定，在程序执行过程中不可变
- C. 变址寄存器内容由操作系统确定，在程序执行过程中可变
- D. 变址寄存器内容由用户确定，在程序执行过程中可变

D【解析】基址寻址和变址寻址在形成有效地址时所用的算法是相同的：

(1)变址寻址是面向用户的，用于访问字符串、向量和数组等成批数据。变址寄存器是面向用户的，在指令的执行过程中，变址寄存器的内容可由用户改变，但是形式地址不变。

(2)基址寻址面向系统，主要用于逻辑地址和物理地址的变换，用于解决程序在主存中的再定位和扩大寻址空间等问题。基址寄存器是面向操作系统的，其内容只能由操作系统或管理程序来管理，用户指令无权操作和修改。在指令的执行过程中，基址寄存器内容不变，但是形式地址可变。

5. 指令系统中，操作数寻址采用不同寻址方式的主要目的是( )。

A. 实现程序控制

B. 降低指令移码的难度

C. 扩大寻址空间，提高编程的灵活性

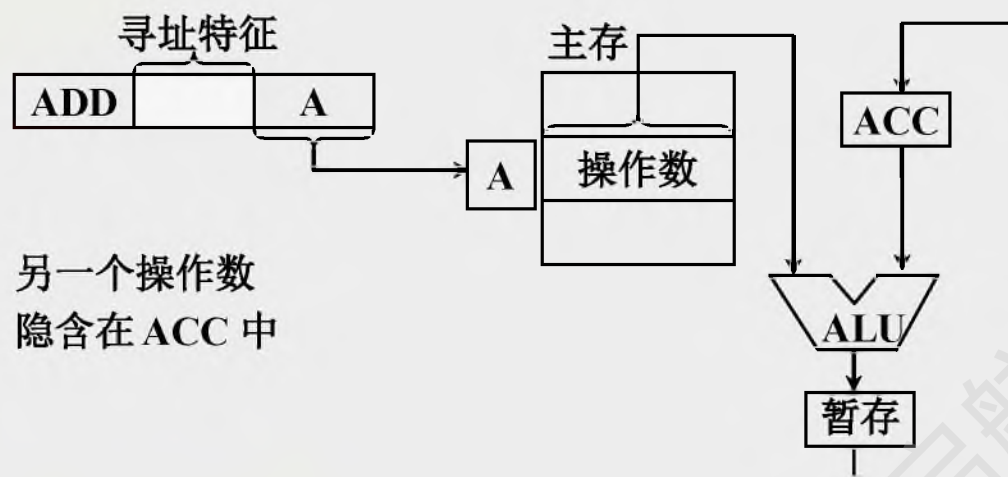
D. 加快访存的速度

C【解析】指令系统中，操作数寻址采用不同寻址方式的主要目的是扩大寻址空间，提高编程的灵活性。

6. **单地址指令**中为了完成两个数的算术运算，除地址码指明的一个操作数外，另一个**操作数**需采用( )。

A. 堆栈寻址方式 B. 立即寻址方式 C. 隐含寻址方式 D. 间接寻址方式

C【解析】操作数地址隐含在操作码中

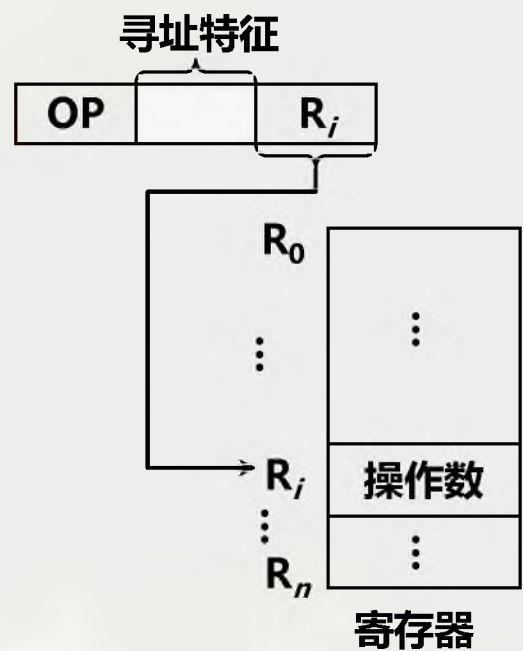


- 指令字中少了一个地址字段，可缩短指令字长

7. 用某个寄存器中操作数的寻址方式称为( )寻址。

- A. 直接      B. 间接      C. 寄存器直接      D. 寄存器间接

C【解析】 $EA = R_i$       有效地址即为寄存器编号

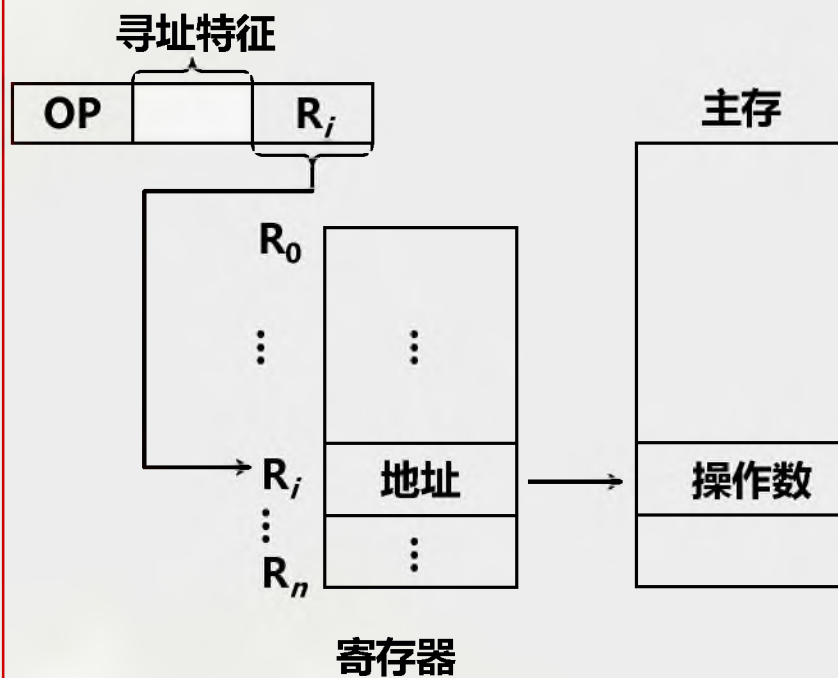


- 执行阶段不访存，只访问寄存器，执行速度快
- 寄存器个数有限，可缩短指令字长

8. 寄存器间接寻址方式中, 操作数在( )。

- A. 通用寄存器      B. 主存单元      C. 程序计数器      D. 堆栈

B【解析】 $EA = (R_i)$  有效地址在寄存器中



- 有效地址在寄存器中, 操作数在存储器中, 执行阶段访存
- 便于编制循环程序



9. 设变址寄存器为X, 形式地址为D, (X)表示寄存器X的内容, 则这种寻址方式的有效地址为( )。

- A.  $EA=(X)+D$                       B.  $EA=(X)+(D)$   
C.  $EA=((X)+D)$                       D.  $EA=((X)+(D))$

A【解析】变址寻址把指令给出的形式地址D与变址寄存器X的内容相加, 形成操作数有效地址。

$$EA=D+(X)$$

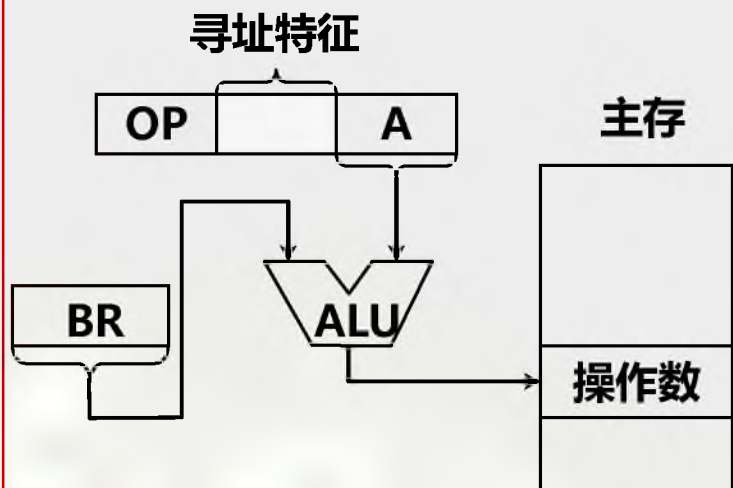
其中, X的内容为变址值。

10. 在基址寻址方式中，操作数的有效地址是( )。

- A. 基址寄存器内容加上形式地址(位移量)
- B. 程序计数器内容加上形式地址
- C. 变址寄存器内容加上形式地址
- D. 寄存器内容加上形式地址

A【解析】其有效地址EA等于指令字中的形式地址A与基址寄存器BR的内容相加之和

$$EA = (BR) + A \quad BR \text{ 为基址寄存器}$$



- 可扩大寻址范围
- 有利于多道程序
- BR 内容由操作系统或管理程序确定
- 在程序的执行过程中 BR 内容不变，形式地址 A 可变



11. ( )便于处理数组问题。

A. 间接寻址    B. 变址寻址    C. 相对寻址    D. 立即寻址

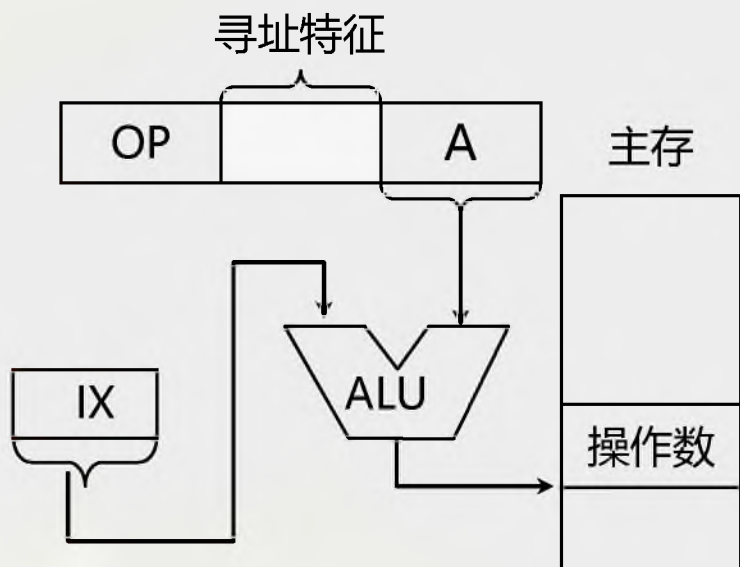
寻址方式	应用场景/优点
隐含寻址	可以有效缩短指令长度
变址寻址	数组问题
基数寻址	多道程序
相对寻址	程序浮动，转移指令
堆栈寻址	执行期间可以不访存

11. ( )便于处理数组问题。

A. 间接寻址 B. 变址寻址 C. 相对寻址 D. 立即寻址

B【解析】有效地址EA等于指令字中的形式地址A与变址寄存器IX的内容相加之和

$EA = (IX) + A$  IX 为变址寄存器



- 可扩大寻址范围
- IX 的内容由用户给定
- 在程序的执行过程中 IX 内容可变, 形式地址 A 不变
- 便于处理数组问题

12. ( )对于实现程序浮动提供了较好的支持。

- A. 间接寻址    B. 变址寻址    C. 相对寻址    D. 直接寻址

寻址方式	应用场景/优点
隐含寻址	可以有效缩短指令长度
变址寻址	数组问题
基数寻址	多道程序
相对寻址	程序浮动，转移指令
堆栈寻址	执行期间可以不访存

C选项正确。

13. **变址寻址**和**基址寻址**的有效地址形成方式类似，但是( )。

- A. 变址寄存器的内容在程序执行过程中是不可变的
- B. 在程序执行过程中，变址寄存器和基址寄存器的内容都是可变的
- C. 在程序执行过程中，基址寄存器的内容不可变，变址寄存器的内容可变
- D. 基址寄存器的内容在程序执行过程中是可变的

**C 【解析】**基址寻址和变址寻址在形成有效地址时所用的算法是相同的：

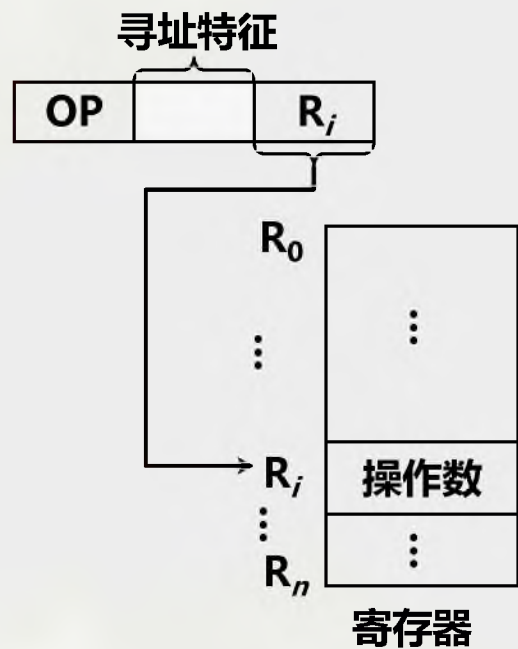
(1)变址寻址是面向用户的，用于访问字符串、向量和数组等成批数据。变址寄存器是面向用户的，在指令的执行过程中，变址寄存器的内容可由用户改变，但是形式地址不变。

(2)基址寻址面向系统，主要用于逻辑地址和物理地址的变换，用于解决程序在主存中的再定位和扩大寻址空间等问题。基址寄存器是面向操作系统的，其内容只能由操作系统或管理程序来管理，用户指令无权操作和修改。在指令的执行过程中，基址寄存器内容不变，但是形式地址可变。

14. 为了缩短指令中某个地址段的位数, 有效的方法是采取( )。

A. 立即寻址    B. 变址寻址    C. 间接寻址    D. 寄存器寻址

D 【解析】 $EA = R_i$       有效地址即为寄存器编号



- 执行阶段不访存, 只访问寄存器, 执行速度快
- 寄存器个数有限, 可缩短指令字长

15. 变址寄存器为X, 形式地址为D, 某机具有先间址后变址的寻址方式, 则这种寻址方式的有效地址为( )。

A.  $EA=(X)+D$     B.  $EA=(X)+(D)$

C.  $EA=((X)+D)$     D.  $EA=X+D$

B【解析】先间址后变址, 即先把寄存器单元内容当作地址, 再加上形式地址的内容得到操作数的地址, 即 $EA=(X)+(D)$ 。



16. 【吉林大学，2014】某机器字长16位，主存按字节编址。转移指令采用相对寻址，由两个字节组成，第一个字节为操作码字段，第二个字节为相对偏移量字段。若某转移指令所在的主存地址为4000H，相对偏移量字段的内容为06H，则该转移指令的地址码为( )。

A. 4002H    B. 4004H    C. 4006H    D. 4008H

D【解析】主存按字节编址，取指令时，每取一个字节PC自动加1。由于转移指令由两个字节组成，取出这条转移指令之后的PC值等于4002H，因此转移指令成功转移后的目标地址为

$$PC=4000H+2H+06H=4008H$$



**【大显身手】**

启航教育 刘财政

1. **基址寄存器**的内容为2000H, **变址寄存器**的内容为03A0H, **指令地址码**部分是3FH, 当前**正在执行的指令所在地址为2B00H**, 请求出:

(1)变址编址(考虑基址)访存有效地址。

(2)相对编址访存有效地址。

**【解析】**(1)变址寻址方式的有效地址为变址寄存器的内容加上指令中的形式地址字段的值, 故而, 变址编址(考虑基址)访存有效地址  $= 2000H + 03A0H + 3FH = 23DFH$ 。

(2)相对寻址的有效地址为当前PC的内容与形式地址的和。本题中, 相对编址访存有效地址  $= 2B00H + 3FH = 2B3FH$  (注意, 当前执行指令在IR中, 不在PC中)。

2. 一条**双字长**的取数指令( LDA)存于存储器的100和101单元, 其中第一个字为操作码和寻址特征M, 第二个字为形式地址。假设**PC当前值为100**, **变址寄存器XR的内容为100**, **基址寄存器的内容为200**, 存储器各单元的内容如表所示。

100		101	102		300		400	401	402		500		800
LDA	M	300		...	800		700	400	500	...	200	...	600

写出在下列寻址方式中, 取数指令执行结束后, **累加器ACC的内容**。

(1)直接寻址。

【解析】(1)直接寻址的指令地址码字段直接给出了数据的内存地址。根据指令的结构可知指令的第二个字为300, 表示形式地址。那么, 在直接寻址方式下, 操作数应该是800。在取指令结束之后, 该数据被取到累加器ACC中, 所以ACC的内容为800。下面各题类似, 不再赘述。

2. 一条**双字长**的取数指令( LDA)存于存储器的100和101单元, 其中第一个字为操作码和寻址特征M, 第二个字为形式地址。假设**PC当前值为100**, **变址寄存器XR的内容为100**, **基址寄存器的内容为200**, 存储器各单元的内容如表所示。

100		101	102		300		400	401	402		500		800
LDA	M	300		...	800		700	400	500	...	200	...	600

写出在下列寻址方式中, 取数指令执行结束后, **累加器ACC的内容**。

(2)立即寻址。

(2)在立即寻址方式下, 指令的地址字段300表示的就是操作数本身。

2. 一条**双字长**的取数指令( LDA)存于存储器的100和101单元, 其中第一个字为操作码和寻址特征M, 第二个字为形式地址。假设**PC当前值为100**, **变址寄存器XR的内容为100**, **基址寄存器的内容为200**, 存储器各单元的内容如表所示。

100		101	102		300		400	401	402		500		800
LDA	M	300		...	800		700	400	500	...	200	...	600

写出在下列寻址方式中, 取数指令执行结束后, **累加器ACC的内容**。

(3)间接寻址。

(3)在间接寻址方式下, 地址码字段给出的是操作数地址的地址。所以, 操作数的有效地址为800, 存放在该位置的操作数为600。

2. 一条**双字长**的取数指令( LDA)存于存储器的100和101单元, 其中第一个字为操作码和寻址特征M, 第二个字为形式地址。假设**PC当前值为100**, **变址寄存器XR的内容为100**, **基址寄存器的内容为200**, 存储器各单元的内容如表所示。

100		101	102		300		400	401	402		500		800
LDA	M	300		...	800		700	400	500	...	200	...	600

写出在下列寻址方式中, 取数指令执行结束后, 累加器ACC的内容。

(4)相对寻址。

(4)相对寻址的操作数有效地址为PC值与形式地址之和。显然, PC值为100, 在执行完取指令操作之后, PC的值加2(因为指令是双字长指令)变成102。形式地址为300, 有效地址为402。查表可知, 存放在地址402的操作数是500。



2. 一条**双字长**的取数指令( LDA)存于存储器的100和101单元, 其中第一个字为操作码和寻址特征M, 第二个字为形式地址。假设**PC当前值为100**, **变址寄存器XR的内容为100**, **基址寄存器的内容为200**, 存储器各单元的内容如表所示。

100		101	102		300		400	401	402		500		800
LDA	M	300		...	800		700	400	500	...	200	...	600

写出在下列寻址方式中, 取数指令执行结束后, 累加器ACC的内容。

(5)变址寻址。

(5)在变址寻址方式下, 操作数的有效地址等于变址寄存器的内容加上形式地址, 即 $100+300=400$ 。查表可知, 存放在地址400的操作数是700。



2. 一条**双字长**的取数指令( LDA)存于存储器的100和101单元, 其中第一个字为操作码和寻址特征M, 第二个字为形式地址。假设**PC当前值为100**, **变址寄存器XR的内容为100**, **基址寄存器的内容为200**, 存储器各单元的内容如表所示。

100		101	102		300		400	401	402		500		800
LDA	M	300		...	800		700	400	500	...	200	...	600

写出在下列寻址方式中, 取数指令执行结束后, 累加器ACC的内容。

(6)基址寻址。

(6)在基址寻址方式下, 操作数的有效地址等于基址寄存器的内容加上形式地址, 即 $200+300=500$ 。查表可知, 存放在地址500的操作数是200。

3. 在某16位机器中，存储器按**字节**编址。一条**双字长直接寻址**的子程序调用指令，其第一个字为操作码和寻址特征，第二个字为**地址码5000H**。假设**PC当前值为2000H**，**SP的内容为0100H**，**栈顶内容为2746H**，存储器按字节编址，而且进栈操作是先 $(SP) - \Delta \rightarrow SP$ ，后存入数据。试回答下列几种情况下，PC、SP及栈顶内容各为多少？

(1)CALL指令被读取前。

(2)CALL指令被执行后。

(3)子程序返回后

3. 在某16位机器中，存储器按**字节**编址。一条**双字长直接寻址**的子程序调用指令，其第一个字为操作码和寻址特征，第二个字为**地址码5000H**。假设**PC当前值为2000H**，**SP的内容为0100H**，**栈顶内容为2746H**，存储器按字节编址，而且进栈操作是先 $(SP) - \Delta \rightarrow SP$ ，后存入数据。试回答下列几种情况下，PC、SP及栈顶内容各为多少？

(1)CALL指令被读取前。

(1)CALL指令被读取前，PC的内容为2000H，栈顶指针(栈顶的地址) $SP=0100H$ ，栈顶内容(存放在栈顶地址的内容)为2746H。

3. 在某16位机器中，存储器按**字节**编址。一条**双字长直接寻址**的子程序调用指令，其第一个字为操作码和寻址特征，第二个字为**地址码5000H**。假设**PC当前值为2000H**，**SP的内容为0100H**，**栈顶内容为2746H**，存储器按字节编址，而且进栈操作是先 $(SP) - \Delta \rightarrow SP$ ，后存入数据。试回答下列几种情况下，PC、SP及栈顶内容各为多少？

(2)CALL指令被执行后。

(2)CALL指令被执行后，由于存储器按字节编址，CALL指令共占4个字节，当前的PC值为2000H，在执行完CALL指令之后，PC的内容为下一条被执行的指令地址，即完成 $PC+4$ 。显然， $PC+4$ 的值为2004H。

故而，程序断点2004H需进栈，此时栈顶指针 $SP = (SP) - 2 = 00FEH$ ，栈顶内容为2004H。

断点入栈之后，将PC的内容更新为子程序入口地址5000H。

3. 在某16位机器中，存储器按**字节**编址。一条**双字长直接寻址**的子程序调用指令，其第一个字为操作码和寻址特征，第二个字为**地址码5000H**。假设**PC当前值为2000H**，**SP的内容为0100H**，**栈顶内容为2746H**，存储器按字节编址，而且进栈操作是先 $(SP) - \Delta \rightarrow SP$ ，后存入数据。试回答下列几种情况下，PC、SP及栈顶内容各为多少？

(3)子程序返回后

(3)子程序返回后，程序断点出栈，PC的内容为2004H，栈顶指针SP被修改为0100H，栈顶内容为2746H。程序返回断点处继续执行。

4. 某机主存容量为**4 M×16位**，且**存储字长等于指令字长**。若该机的指令系统具备**85种操作**，**操作码位数固定**，且具有直接、间接、立即、相对、基址、变址**6种寻址方式**。

(1)画出一地址指令格式并指出各字段的作用。

【解析】因为系统支持85种操作，所以OP操作码字段为7位。因为有直接、间接、立即、相对、基址、变址6种寻址方式，所以M寻址方式特征字段为3位。因为机器字长是16位，而主存容量是4 M×16位，所以需要访问4M的空间，A形式地址字段需 $16-7-3=6$ (位)。

OP (7位)	M (3位)	A (6位)
---------	--------	--------



4. 某机主存容量为**4 M×16位**，且**存储字长等于指令字长**。若该机的指令系统具备**85种操作**，**操作码位数固定**，且具有直接、间接、立即、相对、基址、变址**6种寻址方式**。

(2)该指令直接寻址的最大范围(十进制表示)。

(3)一次间址的寻址范围(十进制表示)。

(4)相对寻址的位移量(十进制表示)。

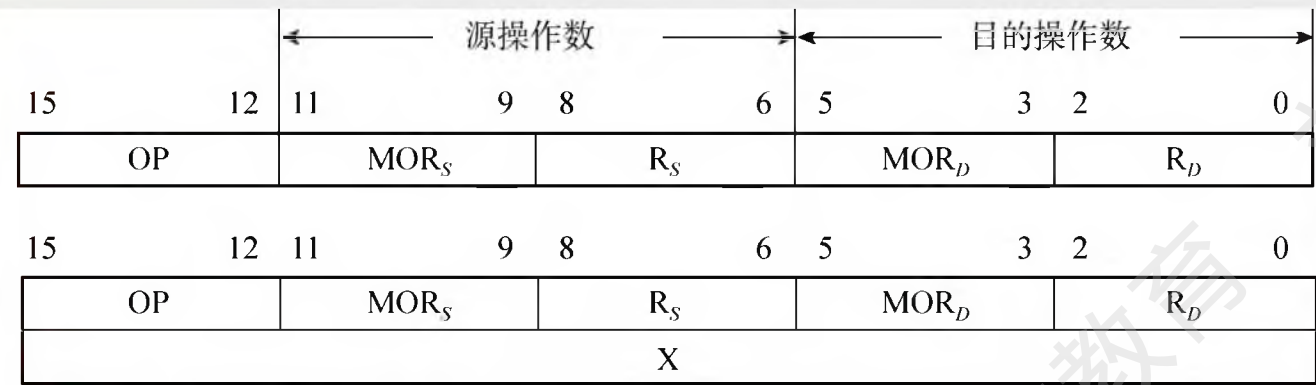
(2)因为形式地址A一共6位，所以直接寻址的最大范围为 $2^6=64$ 。

(3)由于存储字长为32位，因此一次间址的寻址范围为 $2^{16}=65536$ 。

(4)因为形式地址A是6位，所以相对寻址的位移量为 $-32 \sim +31$ 。



5. 假设以下各条指令在执行前均存放在地址为500的单元中，存储器按字节编址，字地址为偶数。每条指令执行前  $(R_0)=100$ ， $(100)=200$ ， $(200)=500$ ， $(604)=200$ ， $MOV(OP)=1001$ (二进制)。MOV指令的功能是将源操作数传到目的地址，指令格式和寻址方式及其二进制如图所示。



- MOD=000寄存器寻址 汇编符号: R<sub>n</sub>
- MOD=001寄存器间接寻址 汇编符号: ( R<sub>n</sub> )
- MOD=010变址寻址 汇编符号: X ( R<sub>n</sub> )
- MOD=011变址间接寻址 汇编符号: @X ( R<sub>n</sub> )
- MOD=100相对寻址 汇编符号: X
- MOD=101相对间接寻址 汇编符号: @X
- MOD=110立即寻址 汇编符号: #X
- MOD=111直接寻址 汇编符号: @#X

请将以下每条指令译成机器代码，并确定每条指令执行后 $(R_1)$ 等于多少。

(1)  $\text{MOV}(R_0), R_1$ 。

(1)  $\text{MOV}(R_0), R_1$ ，将 $R_0$ 的地址所指向的内存单元的内容送到 $R_1$ ，如图所示。

1001	001	000	000	001
------	-----	-----	-----	-----

源操作数有效地址  $EA=(R_0)=100$ ，源操作数  $D=((R_0))=200$ 。指令执行后  $(R_1)=D=200$ 。

请将以下每条指令译成机器代码，并确定每条指令执行后( $R_1$ )等于多少。

(2)MOV @100( $R_0$ ),  $R_1$ 。

(2)MOV @100( $R_0$ ),  $R_1$ , 将与 $R_0$ 内容所表示的地址再加上偏移100的内存单元的内容送到 $R_1$ , 如图所示。

1001	011	000	000	001
100				

源操作数有效地址  $EA=((R_0)+100)=500$  , 源操作数  $D=(EA)=(500)=1001011000000001$ (二进制)。指令执行后  $(R_1)=D=1001011000000001$ (二进制)。

请将以下每条指令译成机器代码，并确定每条指令执行后( $R_1$ )等于多少。

(3)MOV 100,  $R_1$ 。

(3)MOV 100,  $R_1$ ，将内存单元偏移地址100的内容送到寄存器 $R_1$ ，如图所示。

1001	100	000	000	001
100				

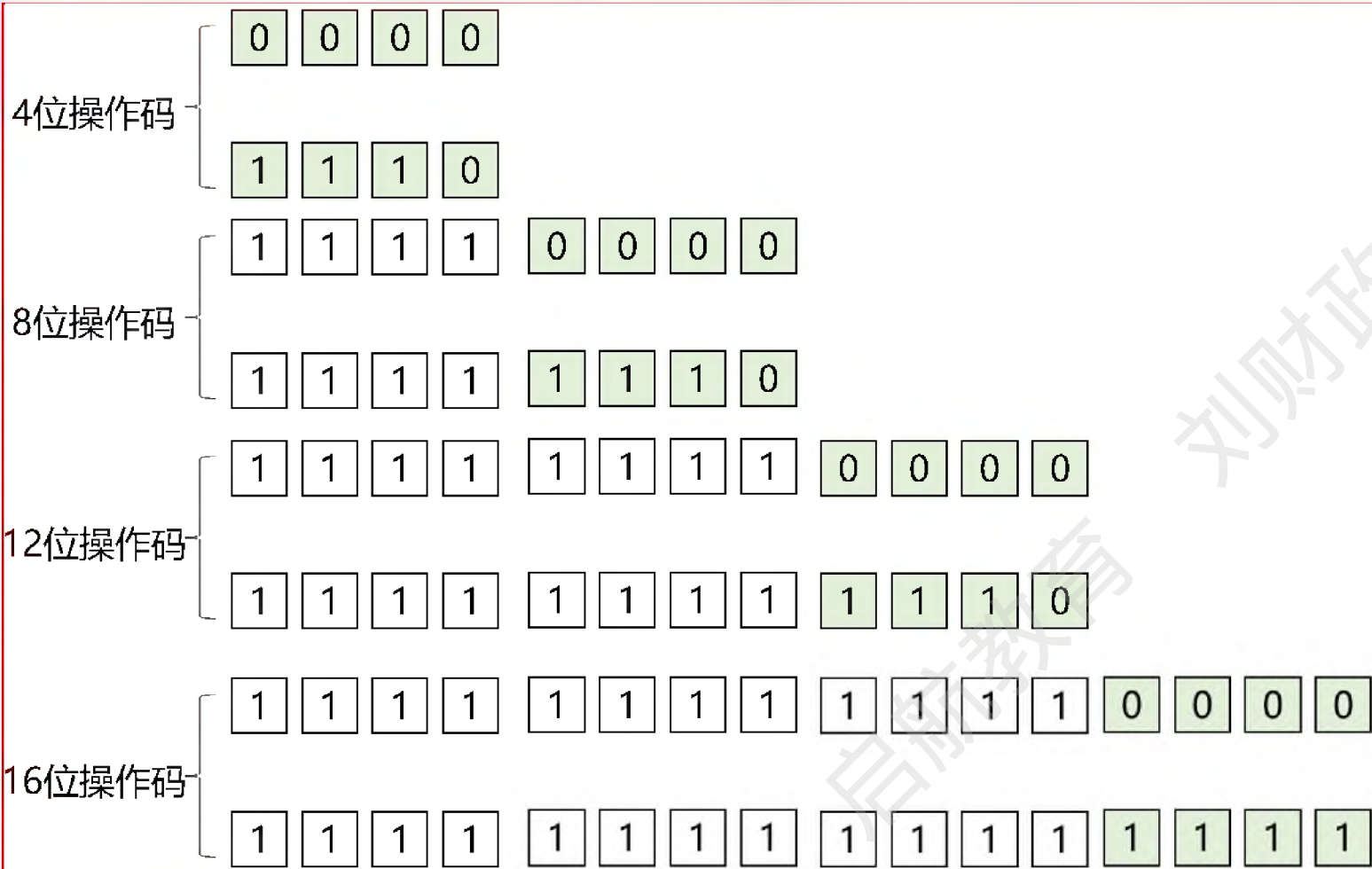
源操作数有效地址  $EA=(PC)+100=504+100=604$ ，源操作数  $D=(EA)=(604)=200$ 。

指令执行后( $R_1$ )= $D=200$ 。

## 考点二

### 【大显身手】

启航教育 刘财政



在设计时，有一个结论大家，需要记住，上一级的地址每留出一条指令，下一级就扩展 $2^K$ 条指令，其中K是地址码位数。



1. 某机器字长16位，存储字长等于指令字长，若存储器直接寻址空间为128字，变址时的位移量为 $-64 \sim +63$ ，16个通用寄存器可作为变址寄存器。设计一套指令格式，满足下列寻址类型的要求。

(1)直接寻址的二地址指令3条。

(2)变址寻址的一地址指令6条。

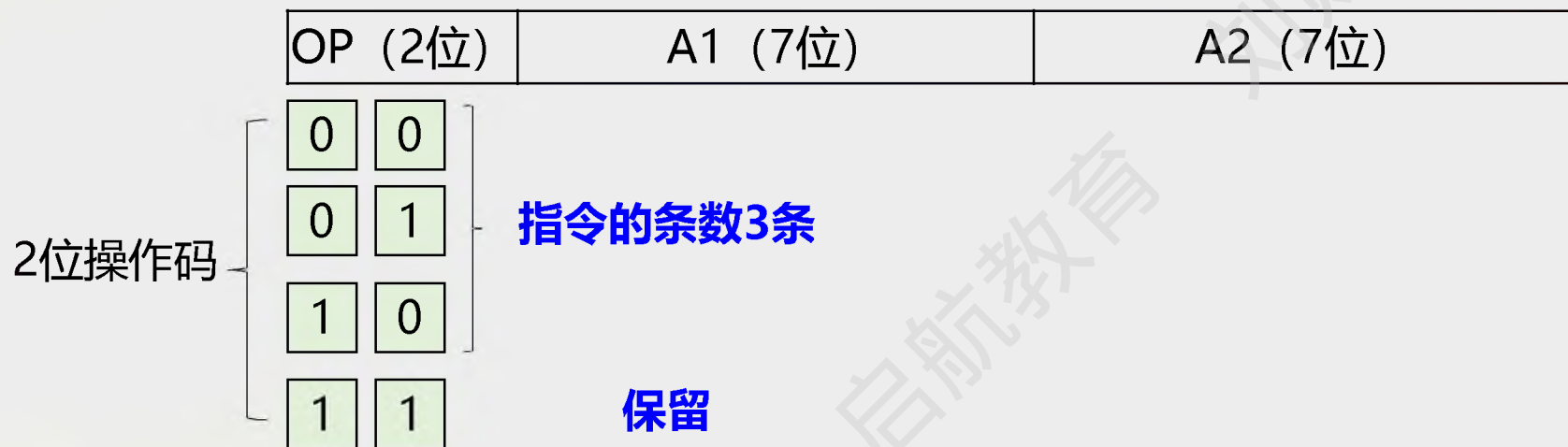
(3)寄存器寻址的二地址指令9条。

(4)直接寻址的一地址指令13条。



(1)直接寻址的二地址指令3条。

(1)因为存储字长等于指令字长，而且机器字长为16位，可取指令字长为16位。  
对于直接寻址的二地址指令，因为**指令的条数是3条，需要操作码3个**，用**两位**二进制数表示**操作码**。而指令的直接寻址范围是128字，需要7位来表示。那么，直接寻址的二地址指令格式如图所示。



(2)变址寻址的一地址指令6条。

(2) 变址寻址的位移量为 - 64 ~ + 63，需要7位二进制数来表示这个范围。对于1地址指令，若是寄存器变址寻址，16个寄存器需要4位二进制数来表示。故而，指令格式可设计为如表所示。

OP (5位)	Resigter (4位)	A1 (7位)
---------	---------------	---------

2位操作码	0	0
	0	1
	1	0

5位操作码	1	1	0	0	0
	1	1	1	0	1
	1	1	1	1	0
	1	1	1	1	1
	1	1	1	1	1

指令的条数6条

保留

(3)寄存器寻址的二地址指令9条。

(3) 因为共有16个寄存器，寄存器寻址的二地址指令有两个寄存器地址字段，每个字段用4位二进制数表示。该寄存器寻址的二地址指令格式如图4-11所示。

OP (8位)					Resigter (4位)			Resigter (4位)		
0	0									
0	1									
1	0									
1	1	0	0	0						
1	1	1	0	1						
1	1	1	1	0	0	0	0	}	指令的条数8条	
1	1	1	1	0	1	1	1			
1	1	1	1	1	0	0	0			
1	1	1	1	1	0	0	1		指令的条数1条	
1	1	1	1	1	0	0	1		保留	
1	1	1	1	1	1	1	1		保留	

(4)直接寻址的一地址指令13条。

(4) 直接寻址的一地址指令，需要7位来表示地址，故而可以采用如表所示的指令格式。

		OP (9位)					A1 (7位)				
2位操作码	{	0	0								
		0	1								
		1	0								
5位操作码	{	1	1	0	0	0					
		1	1	1	0	1					
		1	1	1	1	0	0	0	0		
8位操作码	{	1	1	1	1	0	1	1	1		
		1	1	1	1	1	0	0	0		
9位操作码	{	1	1	1	1	1	0	0	1	0	指令的条数13条
		1	1	1	1	1	1	1	1	0	

指令的条数13条

2. 某机器字长16位，存储器直接寻址空间为128字，变址时的位移量为-64 ~ +63，16个通用寄存器均可作为变址寄存器。设计一套指令格式，满足下列寻址类型的要求。

(1)直接寻址的二地址指令3条。

(2)变址寻址的一地址指令6条。

(3)寄存器寻址的二地址指令8条。

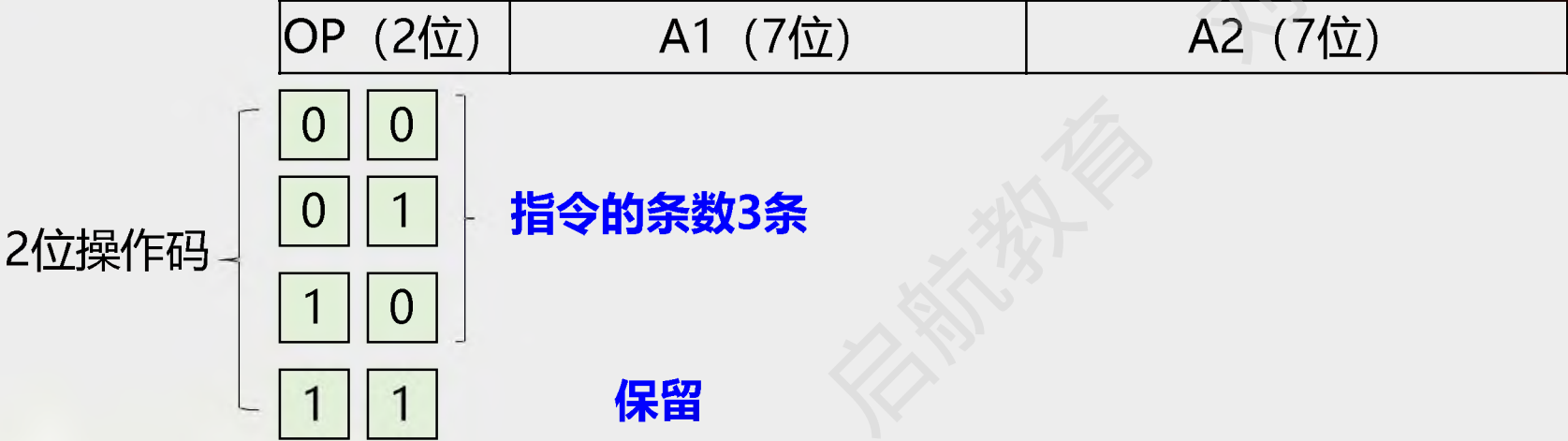
(4)直接寻址的一地址指令12条。

(5)零地址指令32条。



(1)直接寻址的二地址指令3条。

**【解析】**(1)根据题意，直接寻址的每个地址都应该能访问到128字的空间，即7位。对于直接寻址的二地址指令，需要两个字段的7位地址字段。因为机器字长是16位，剩下的2位操作码可以用00、01、10来表示直接寻址的3条指令，11用来作为下一种格式指令的操作码扩展使用，如图所示。



(2)变址寻址的一地址指令6条。

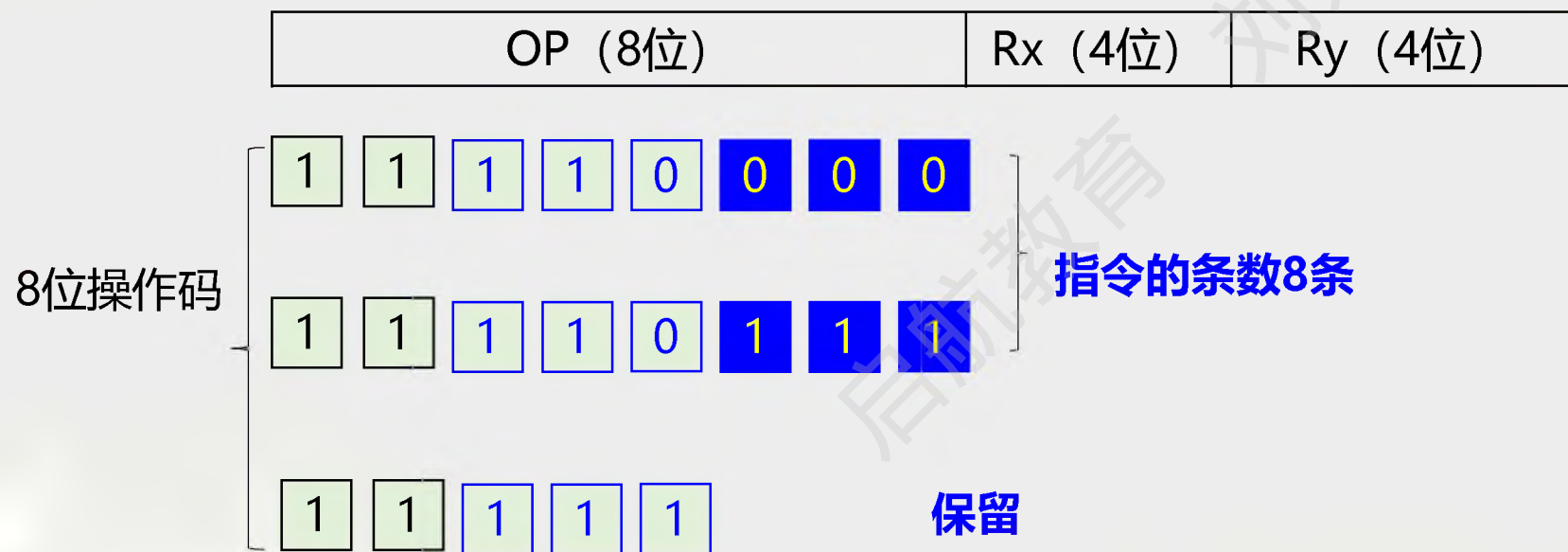
(2)指令可以直接寻址128个字，变址寻址指令的地址段位移量范围为-64 ~ +63，形式地址字段A取7位。另外，16个寄存器也应该能访问到，这16个通用寄存器可作为变址寄存器，取4位用来表示变址寄存器Rx的编号。那么，只有5位可用作操作码了。因为需要变址寻址的指令总共6条，可以将1 1000 ~ 1 1101这6个编码作为6条变址寻址指令的操作码，剩下的两个编码1 1110和1 1111作为扩展用。这6条变址寻址的一地址指令的格式如图所示。





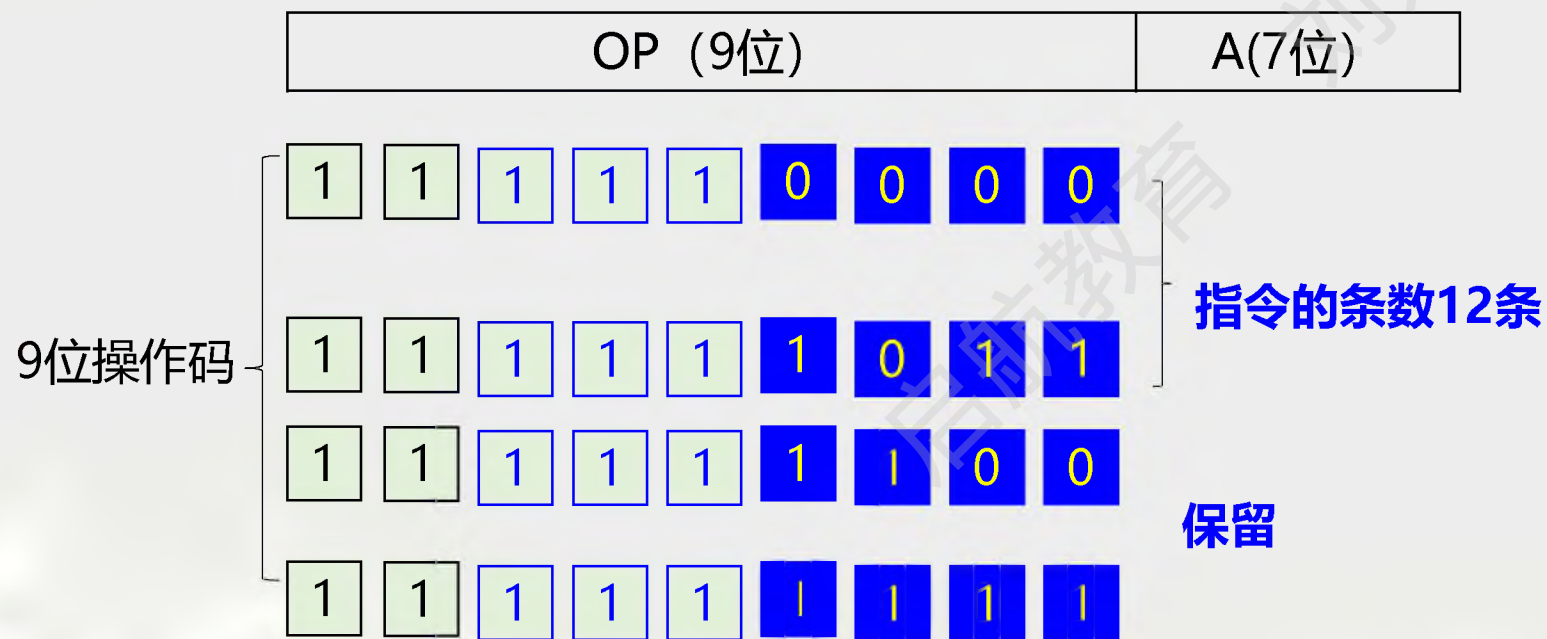
(3)寄存器寻址的二地址指令8条。

(3)寄存器寻址的二地址指令，需要两个地址字段，每个字段4位，用来表示16个寄存器的编号，剩下的8位可用作操作码。显然，可以分配1111 0000 ~ 1111 0111给这8条指令，作为它们的操作码。剩下的1111 1000 ~ 1111 1111共8个编码作为扩展用，如图所示。



(4)直接寻址的一地址指令12条。

(4)直接寻址的一地址指令，其地址码字段每个字段应该为7位，显然当直接寻址的指令是一地址指令时，地址码字段是7位，操作码可以为9位。显然，可以采用11111 0000 ~ 11111 1011作为12条一地址直接寻址指令的操作码，剩下的11111 1100 ~ 11111 1111作为扩展用，如图所示。



(5)在零地址指令中，指令的16位都作为操作码。显然，可以采用1111 1110 0000 0000 ~ 1111 11100001 1111来构成32条零地址指令的操作码，剩下的其他操作码暂时不用，如图所示。







9位操作码

1 1 1 1 1 0 0 0 0

1 1 1 1 1 1 0 1 1

1 1 1 1 1 1 1 0 0

1 1 1 1 1 1 1 1 1

指令的条数12条

保留

16位操作码

1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1

1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

指令的条数32条

保留

**【谢谢大家】**

启航教育 刘财政

## 考点四：CISC和RISC

启航教育 刘财政



## 考点四：CISC和RISC

按指令格式的复杂度来分，有两种类型计算机：

复杂指令集计算机CISC (Complex Instruction Set Computer)

精简指令集计算机RISC (Reduce Instruction Set Computer)

早期CISC设计风格的主要特点

### (1) 指令系统复杂

变长操作码 / 变长指令字 / 指令多 / 寻址方式多 / 指令格式多

### (2) 指令周期长

绝大多数指令需要多个时钟周期才能完成

### (3) 各种指令都能访问存储器

除了专门的存储器读写指令外，运算指令也能访问存储器

### (4) 采用微程序控制

### (5) 有专用寄存器

### (6) 难以进行编译优化来生成高效目标代码

## 考点四：CISC和RISC

### ◆ CISC的缺陷

- 日趋庞大的指令系统不但使计算机的研制周期变长，而且难以保证设计的正确性，难以调试和维护，并且因指令操作复杂而增加机器周期，从而降低了系统性能。

- ◆ 1975年IBM公司开始研究指令系统的合理性问题，John Cocks提出精简指令系统计算机 RISC ( Reduce Instruction Set Computer )。

## 考点四：CISC和RISC

### RISC设计风格的主要特点

#### (1) 简化的指令系统

指令少 / 寻址方式少 / 指令格式少 / 指令长度一致

#### (2) 以RR方式工作

除Load/Store指令可访问存储器外，其余指令都只访问寄存器。

#### (3) 指令周期短

以流水线方式工作，因而除Load/Store指令外，其他简单指令都只需一个或一个不到的时钟周期就可完成。

#### (4) 采用大量通用寄存器，以减少访存次数

#### (5) 采用组合逻辑电路控制，不用或少用微程序控制

#### (6) 采用优化的编译系统，力求有效地支持高级语言程序

	CISC	RISC
指令系统	复杂,庞大	简单,精简
指令数	一般大于200	一般小于100
指令格式	一般大于4	一般小于4
指令字长	一般大于4	一般小于4
寻址方式	不固定	固定32位
可访问指令	不加限制	只有LOAD/STORE指令
各种指令使用频率	相差很大	相差不大绝大多数在一个机器
各种指令执行时间	相差很大	周期完成
优化编译实现	很难	较容易
程序源代码长度	较短	较长
控制逻辑实现方式	绝大多数为微程序控制	绝大多数为硬连线控制
寄存器	少	多

**【政哥点拨】**

启航教育 刘财政

【北京科技大学，2019】下列关于RISC的说法中，错误的是()。

- A. RISC普遍采用微程序控制器【错判】
- B. RISC大多数指令在一个时钟周期内完成
- C. RISC的内部通用寄存器数量相对CISC多
- D. RISC的指令数、寻址方式和指令合适种类相对CISC少



	CISC	RISC
指令系统	复杂,庞大	简单,精简
指令数	一般大于200	一般小于100
指令格式	一般大于4	一般小于4
指令字长	一般大于4	一般小于4
寻址方式	不固定	固定32位
可访问指令	不加限制	只有LOAD/STORE指令
各种指令使用频率	相差很大	相差不大绝大多数在一个机器
各种指令执行时间	相差很大	周期完成
优化编译实现	很难	较容易
程序源代码长度	较短	<b>较长</b>
控制逻辑实现方式	绝大多数为微程序控制	<b>绝大多数为硬连线控制</b>
寄存器	少	<b>多</b>



**【牛刀小试】**

启航教育 刘财政

1. 下列关于RISC的描述中，**不正确**的是( )。

- A. 指令条数比CISC少
- B. 指令长度固定，指令格式种类少，寻址方式种类少
- C. 在程序中出现频度占80%的指令占指令总数的20%
- D. 只有取数/存数指令访问存储器

**C【解析】**与CISC相比，RISC的指令系统比较简单，指令条数也比较少。而且RISC的指令长度固定，指令格式种类少，寻址方式种类也少。除此之外，RISC机器只有LOAD和STORE指令能访问内存。

在CISC指令系统中，指令丰富、功能强。大量测试表明，CISC最常使用的是一些比较简单的指令，这类指令仅占指令总数的20%，但在各种程序中出现的频度占80%，其余大多数指令是功能复杂的指令，这类指令占指令总数的80%，但使用频度很低，仅占20%。因此，人们把这种情况称为“20%~80%定律”。

但是，这个定律不是针对RISC的，而是针对CISC的。

2. 下面关于RISC技术的描述中，**正确**的是()。

A. 采用RISC技术后，计算机的体系结构又恢复到早期的比较简单情况

B. 为了实现兼容性设计，新RISC是从原来的CISC系统的指令系统中挑选一部分实现的

C. RISC的主要目标是减少指令数

D. RISC设有乘法指令和浮点运算指令

**C【解析】**采用RISC技术之后，计算机的指令变简单了，这是为了并行处理，提高系统指令执行速度和系统效率，并没有使计算机的体系结构恢复到早期的比较简单情况。故而，A选项错误。

RISC的指令系统并不是都来源于CISC，也不是从CISC中抽取比较简单的指令，这二者没有必然的联系。

RISC中没有复杂的指令，乘法指令和浮点数运算指令属于复杂指令，运算速度慢，RISC一般不引入该类指令。

3. 【杭州电子科技大学，2017】下面关于RISC和CISC的描述中，**不正确**的是( )。

A. RISC指令格式种类、寻址方式少，指令长度固定，所以更容易用硬布线电路实现，执行速度要比CISC快

B. CISC的指令功能强大、寻址方式种类多，便于汇编程序员编程

**C. CISC指令种类多，所以更有利于编译、优化【错误】**

D. RISC多数指令能够在在一个机器周期内完成，特别适合流水线工作

	CISC	RISC
指令系统	复杂,庞大	简单,精简
指令数	一般大于200	一般小于100
指令格式	一般大于4	一般小于4
指令字长	一般大于4	一般小于4
寻址方式	不固定	固定32位
可访问指令	不加限制	只有LOAD/STORE指令
各种指令使用频率	相差很大	相差不大绝大多数在一个机器
各种指令执行时间	相差很大	周期完成
优化编译实现	很难	较容易
程序源代码长度	较短	较长
控制逻辑实现方式	绝大多数为微程序控制	绝大多数为硬连线控制
寄存器	少	多

**【真题实战】**

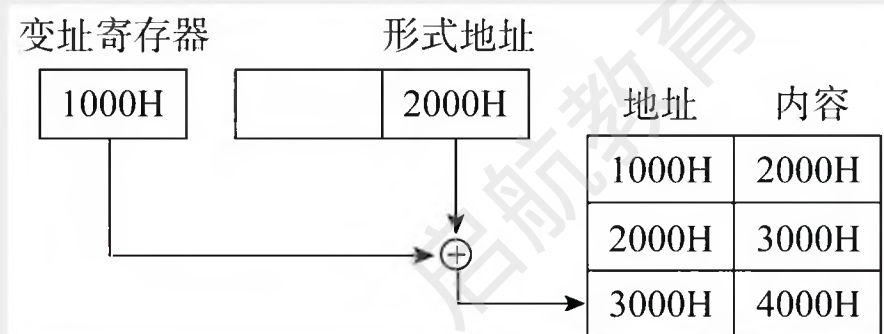
启航教育 刘财政



1、假设变址寄存器 R 的内容为 1000H，指令中的形式地址为 2000 H；地址 1000H 中的内容为 2000H，地址 2000H 中的内容为 3000H，地址 3000 H 中的内容为 4000H，则变址寻址方式下访问到的操作数是\_\_\_\_\_。

A. 1000H          B. 2000H          C. 3000H          D. 4000 H

D【解析】据变址寻址的主要方法，变址寄存器的内容（1000H）与形式地址的内容（2000H）相加之后，得到操作数的实际地址（3000H），根据实际地址访问内存，获取操作数4000H。





2、某计算机有 **16 个通用寄存器**，采用 32 位定长指令字，**操作码字段（含寻址方式位）为 8 位**，Store 指令的源操作数和目的操作数分别采用**寄存器直接寻址和基址寻址方式**。若基址寄存器可使用任一通用寄存器，且**偏移量用补码表示**，则 Store 指令中偏移量的取值范围是\_\_\_\_\_。

- A. -32768 ~ +32767      B. -32767 ~ +32768  
C. -65536 ~ +65535      D. -65535 ~ +65536

A【解析】采用32位定长指令字，其中操作码为8位，而 Store指令的源操作数和目的操作数分别采用寄存器直接寻址和基址寻址，机器中共有16个通用寄存器，则寻址一个寄存器编号需要 $\log_2 16 = 4$ 位，源操作数中的寄存器直接寻址和目的操作数采用基址寻址，各需要4位。则留给偏移址的位数为 $32 - 4 - 4 - 8 = 16$ 位，而偏移址用补码表示，因此16位补码的表示范围为-32768~+32767。

OP(8位)	R1(4位)【寄存器直接寻址】	R2(4位)【寄存器直接寻址】
--------	-----------------	-----------------

3、某指令格式如下图所示。

OP	M	I	D
----	---	---	---

其中M为寻址方式, I为变址寄存器编号, D为形式地址。若采用**先变址后间址**的寻址方式, 则操作数的有效地址是\_\_\_\_\_。

- A.  $I+D$                   B.  $(I)+D$                   C.  $((I)+D)$                   D.  $((I))+D$

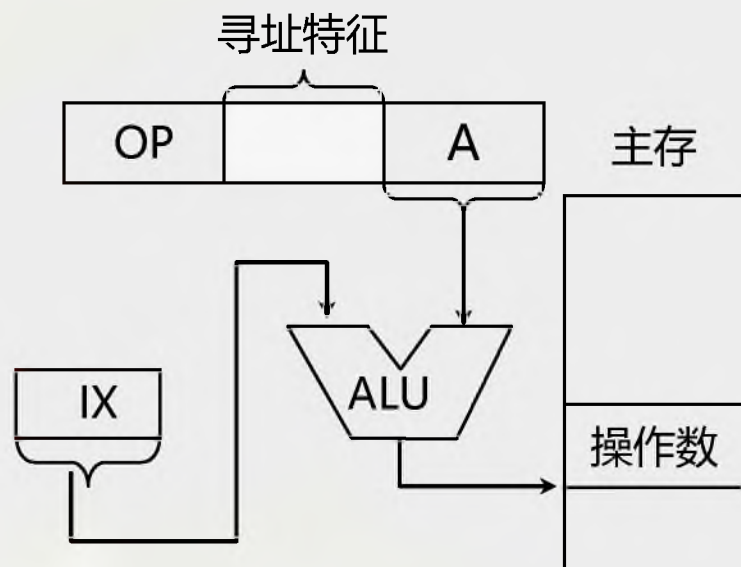
C 【解析】变址寻址中, 有效地址EA等于指令字中的形式地址D与变址寄存器I的内容相加之和, 即 $EA = (I) + D$ 。间接寻址是指令的地址字段给出的是操作数地址的地址, 即 $EA = (D)$ 。所以系统采用先变址后间址的寻址方式, 操作数有效地址是 $EA = ((I) + D)$ 。

4、下列寻址方式中，最适合**按下标顺序访问一维数组**元素的是\_\_\_\_\_。

A. 相对寻址      B. 寄存器寻址      C. 直接寻址      D. 变址寻址

D【解析】有效地址EA等于指令字中的形式地址A与变址寄存器IX的内容相加之和

$EA = (IX) + A$       IX 为基址寄存器



- 可扩大寻址范围
- IX 的内容由用户给定
- 在程序的执行过程中 IX 内容可变, 形式地址 A 不变
- 便于处理数组问题

5、某计算机按字节编址，**指令字长固定**且只有**两种指令格式**，其中三地址指令29条，二地址指令107条，每个地址字段为6位，则指令字长至少应该是\_\_\_\_\_。

- A. 24位            B. 26位            C. 28位            D. 32位

A【解析】题目中的地址有三地址指令，且每个地址字段为6位，因此地址字段有18位。设操作码字段有X位，那么三地址指令最多有 $2^X$ 条。因为实际的三地址指令有29条，因此可以用来扩展二地址指令的三地址指令有 $(2^X - 29)$ 条。根据变长指令编码对着，可以扩展的二地址指令有 $(2^X - 29) * 2^6$ 。由于二地址指令107条，因此 $(2^X - 29) * 2^6 > 107$ ，解出 $X = 5$ 。所以指令字长最短可以是23位，结合选项，当按照字节编址是，最少取24位。

6、按字节编址的计算机中，某double型数组A的首地址为2000H，使用变址寻址和循环结构访问数组元素A，保存数组下标的变址寄存器初值为0，每次循环取一个数组元素，其偏移地址为变址值乘以sizeof (double)，取完后变址寄存器内容自动加1。若某次循环所取元素的地址为2100H，则进入该次循环时变址寄存器的内容是

A, 25,      B, 32      C, 64      D100

B【解析】变址寻址方式最适用于按照下标访问数组元素，其中形式地址A对应于数组首地址，变址寄存器对应于数组下标。结合数据结构的知识， $\text{Loc}(A[i]) = \text{Loc}(A[0]) + i * L$ ，其中i表示数组下标，L表示每个数组元素的大小，本题中对应的是8。于是 $2100\text{H} = 2000\text{H} + i * 8$ ；可以解出 $i = 32$ ，也就是变址寄存器的值。

**【谢谢大家】**

启航教育 刘财政