课时4 指令系统

考点	重要程度	占分	题型
1.指令格式	★★★	0～3	选择、填空
2.指令的寻址方式	必考	6～10	大题
3.CISC和RISC的基本概念			

4.1 指令格式

指令（又称机器指令）是指示计算机执行某种操作的命令，程序由一条条指令构成。

1.指令的基本格式

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。一条指令通常包括操作码字段和地址码字段两部分。

操作码指出指令中该指令应该执行什么性质的操作和具有何种功能。

地址码给出被操作的信息（指令或数据）的地址，包括参加运算的一个或多个操作数所在的地址、运算结果的保存地址、程序的转移地址、被调用的子程序的入口地址等。

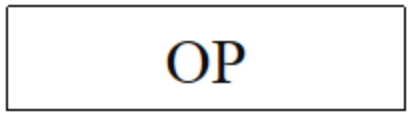
指令的长度是指一条指令中所包含的二进制代码的位数。指令字长取决于操作码的长度、操作数地址码的长度和操作数地址的个数。

1.指令的基本格式

根据指令中操作数地址码的数目的不同，指令分：

一.零地址指令

零地址



只给出操作码OP，没有显式地址。这种指令有两种可能：

1）不需要操作数的指令，如空操作指令、停机指令、关中断指令等。

2）零地址的运算类指令仅用在堆栈计算机中。通常参与运算的两个操作数隐含地从栈顶和次栈顶弹出，送到运算器进行运算，运算结果再隐含地压入堆栈。

二.一地址指令

一地址

这种指令也有两种常见的形态，要根据操作码的含义确定究竟是哪一种。

1.指令的基本格式

1）只有目的操作数的单操作数指令，按A1地址读取操作数，进行OP操作后，结果存回原地址。

指令含义：OP（A1）→A1

如操作码含义是加1、减1、求反、求补等。

2）隐含约定目的地址的双操作数指令，按指令地址A1可读取源操作数，指令可隐含约定另一个操作数由ACC（累加器）提供，运算结果也将存放在ACC中。

指令含义：（ACC）OP（A1）→ACC

若指令字长为32位，操作码占8位，1个地址码字段占24位，则指令操作数的直接寻址范围为224 = 16M。

1.指令的基本格式

三.二地址指令

二地址

OP	A_1	A_2

指令含义：（A1）OP（A2）→A1

对于常用的算术和逻辑运算指令，往往要求使用两个操作数，需分别给出目的操作数和源操作数的地址，其中目的操作数地址还用于保存本次的运算结果。

若指令字长为32位，操作码占8位，两个地址码字段各占12位，则指令操作数的直接寻址范围为212 = 4K。

1.指令的基本格式

四.三地址指令

三地址	OP	A_1	A_2	A_3(结果)

指令含义：（A1）OP（A2）→A3

若指令字长为32位，操作码占8位，3个地址码字段各占8位，则指令操作数的直接寻址范围为28 = 256。

若地址字段均为主存地址，则完成一条三地址需要4次访问存储器（取指令1次，取两个操作数2次，存放结果1次）.

1.指令的基本格式

五.四地址指令

四地址

OP	A_1	A_2	A_3(结果)	A_4（下址）

指令含义：（A1）OP（A2）→A3，A4 = 下一条将要执行指令的地址。

若指令字长为32位，操作码占8位，4个地址码字段各占6位，则指令操作数的直接寻址范围为26 = 64。

2.扩展操作码指令格式

为了在指令字长有限的前提下仍保持比较丰富的指令种类，

可采取可变长度操作码，

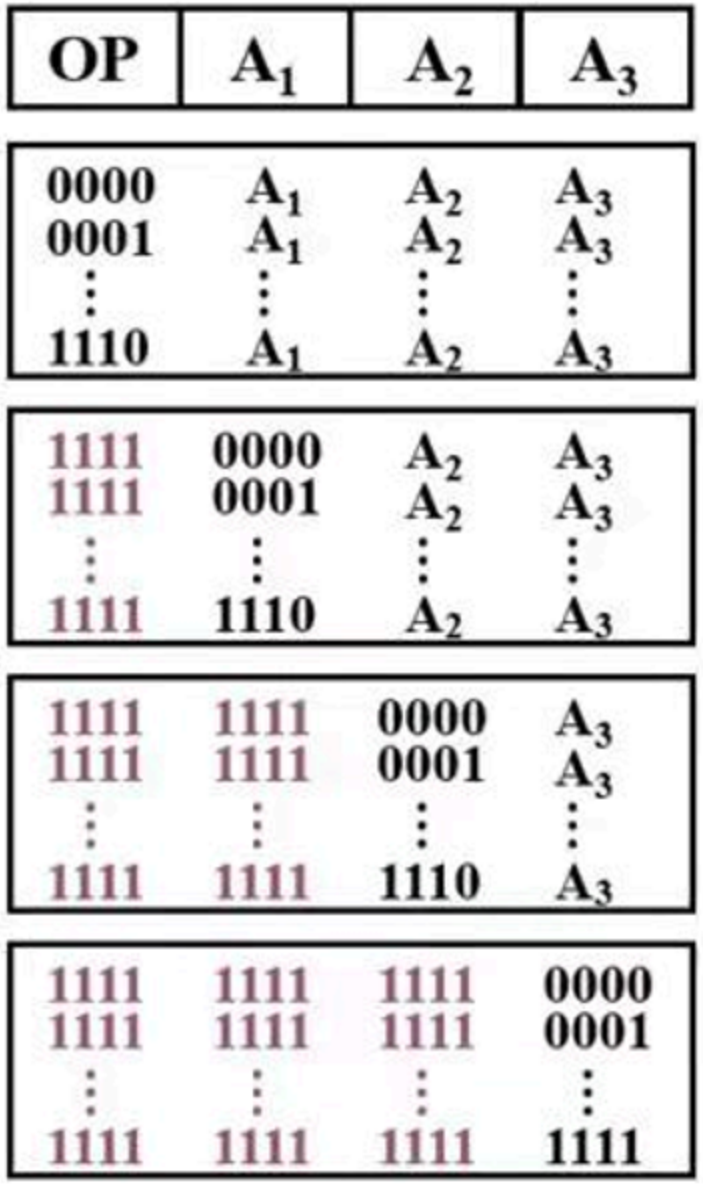
即全部指令的操作码字段的位数不固定，且分散地放在指令字的不同位置上。

最常见的变长操作码方法是扩展操作码，它使操作码的长度随地址码的减少而增加，不同地址数的指令可具有不同长度的操作码，

从而在满足需要的前提下，有效地缩短指令字长。

扫码听课，视频讲解更清晰

2.扩展操作码指令格式



三地址指令操作码每减少一种可多构成2^4种二地址指令

二地址指令操作码每减少一种可多构成2^4种一地址指令

2.扩展操作码指令格式

上图中，指令字长为16位，其中4位为基本操作码字段OP，

另有3个4位长的地址字段A1、A2和A3。

4位基本操作码若全部用于三地址指令，则有16条。

图中所示的三地址指令为15条，1111 留作扩展操作码之用；

二地址指令为15条，1111 1111留作扩展操作码之用；

一地址指令为15条，1111 1111 1111留作扩展操作码之用；

零地址指令为16条。

2.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。

2）各指令的操作码一定不能重复。

4.2 指令的寻址方式

寻址方式是指寻找指令或操作数有效地址的方式，即确定本条指令的数据地址及下一条待执行指令的地址的方法。寻址方式分为指令寻址和数据寻址两大类。

指令中的地址码字段并不代表操作数的真实地址，这种地址称为形式地址（A）；形式地址结合寻址方式，可以计算出操作数在存储器中的真实地址，这种地址称为有效地址（EA）。

注意：（A）表示地址为A的数值，

例如，EA =（A）意思是有效地址是地址A中的数值。

1. 指令寻址和数据寻址

寻找下一条将要执行的指令地址称为 指令寻址；寻找操作数的地址称为数据寻址。

一.指令寻址

指令寻址方式有两种：一种是顺序寻址方式，另一种是跳跃寻址方式。

1）顺序寻址可通过程序计数器（PC）加1（1个指令字长），自动形成下一条指令的地址。

2）跳跃寻址通过转移类指令实现。所谓跳跃，是指下条指令的地址码不由程序计数器给出，而由本条指令给出下条指令地址的计算方式。

注意: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（相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通过程序计数器（PC）给出。

2.常见的数据寻址方式

一.隐含寻址

不明显地给出操作数的地址，而在指令中隐含操作数的地址。

例如，单地址的指令格式就不明显地在地址字段中指出第二操作数的地址，

而规定累加器（ACC）作为第二操作数地址，

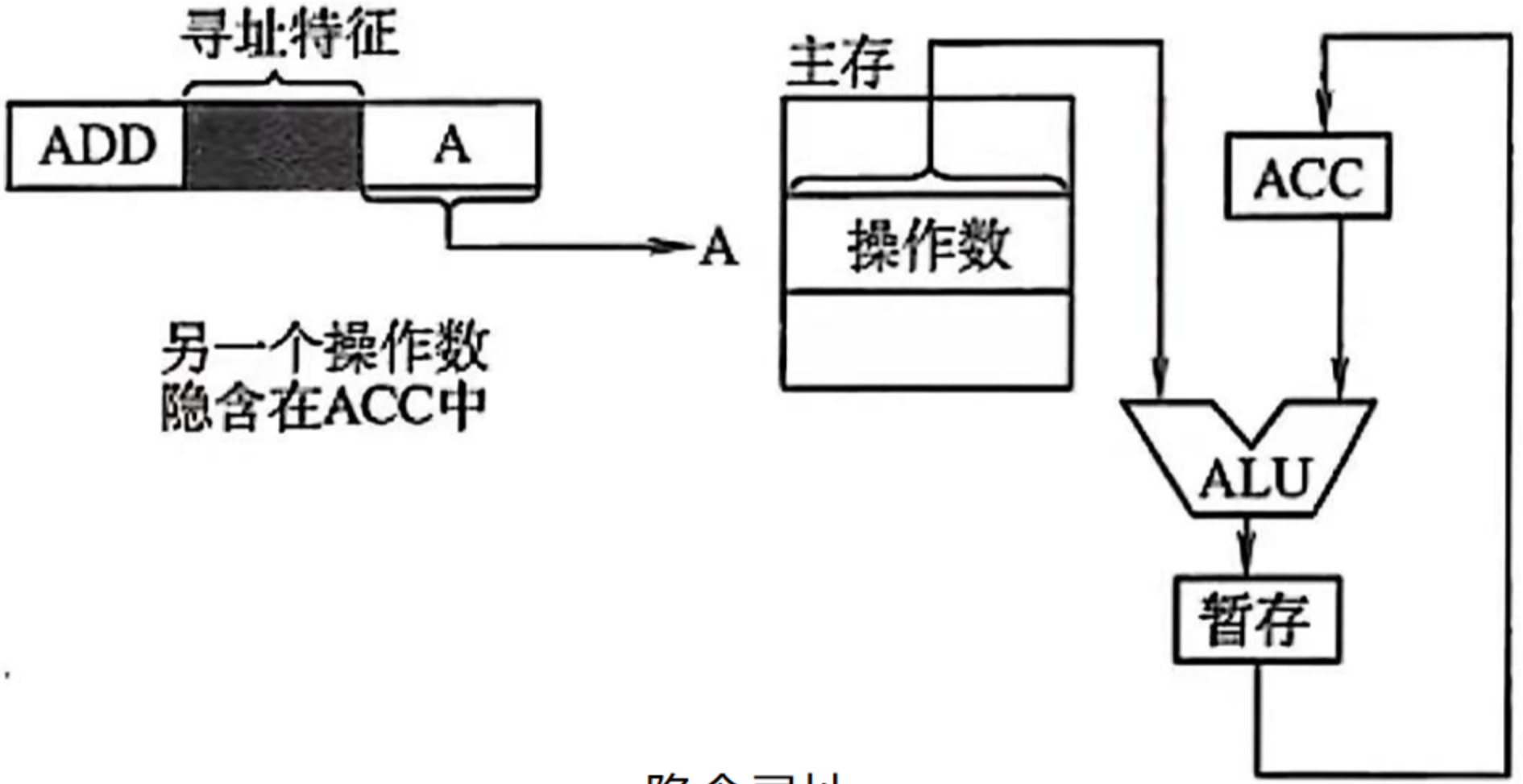
指令格式明显指出的仅是第一操作数的地址。

因此，累加器（ACC）对单地址指令格式来说是隐含寻址，如图所示。

隐含寻址的优点是有利于缩短指令字长；

缺点是需增加存储操作数或隐含地址的硬件.

2.常见的数据寻址方式



隐含寻址

2.常见的数据寻址方式

二.立即（数）寻址

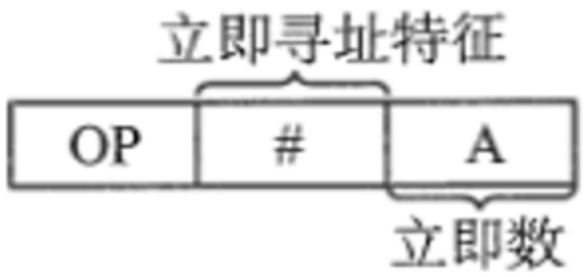
此指令的地址字段指出的不是操作数的地址，而是操作数本身，又称立即数。

数据采用补码形式存放。

图示为立即寻址示意图，图中＃表示立即寻址特征，A就是操作数本身。

立即寻址的优点是指令在执行阶段不访问主存，指令执行时间最短；

缺点是A的位数限制了立即数的范围。

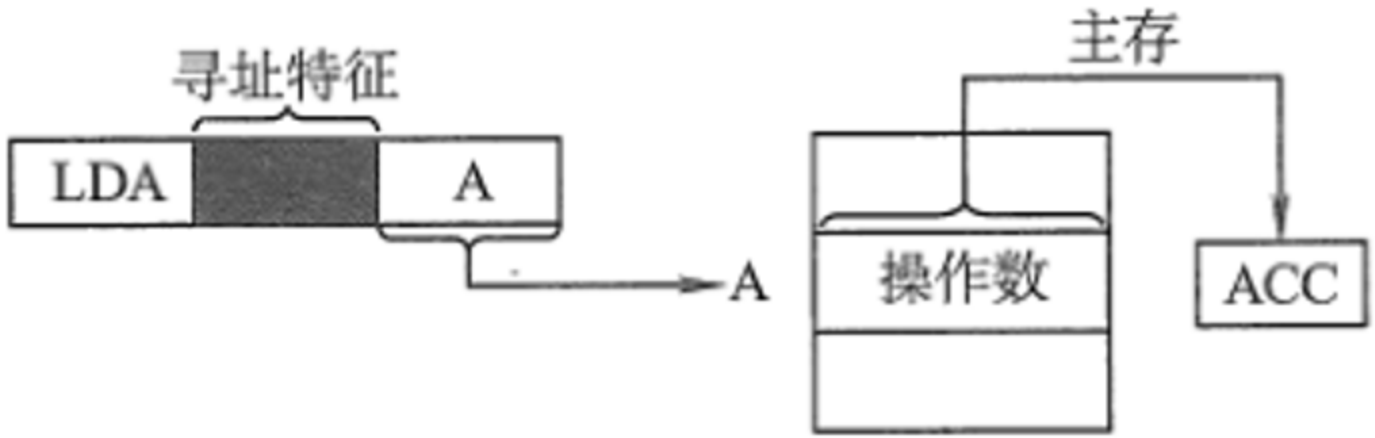


立即寻址

2.常见的数据寻址方式

三.直接寻址

指令字中的形式地址A是操作数的真实地址EA，即EA = A，如图所示。



.4　直接寻址

其优点是简单，指令在执行阶段仅访问一次主存，不需专门计算操作数的地址；

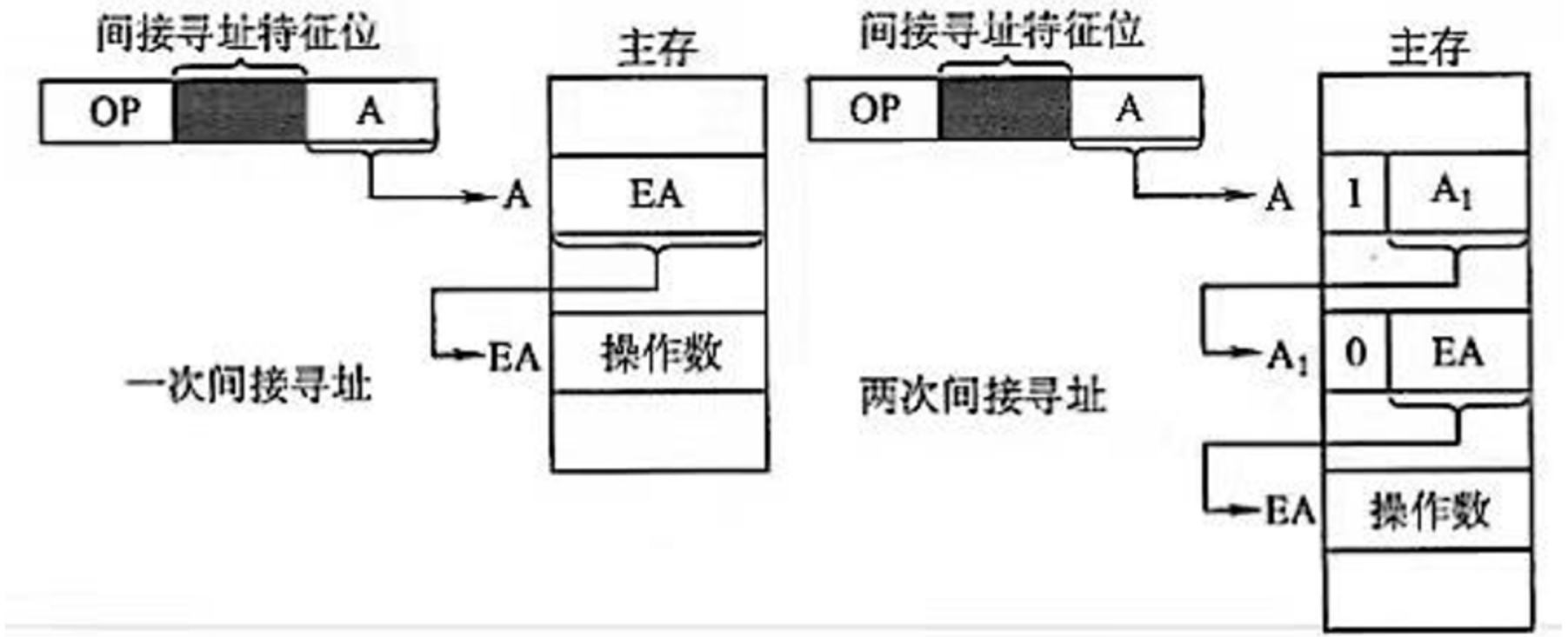
缺点是A的位数决定了该指令操作数的寻址范围，操作数的地址不易修改。

2.常见的数据寻址方式

四.间接寻址

相对于直接寻址而言的，指令的地址字段给出的形式地址不是操作数的真正地址，而是操作数有效地址所在的存储单元的地址，也就是操作数地址的地址，

即EA = （A），如图所示。间接寻址可以是一次间接寻址，也可是多次间接寻址。



2.常见的数据寻址方式

在图中，主存字第一位为1时，表示取出的仍不是操作数的地址，即多次间址；主存字

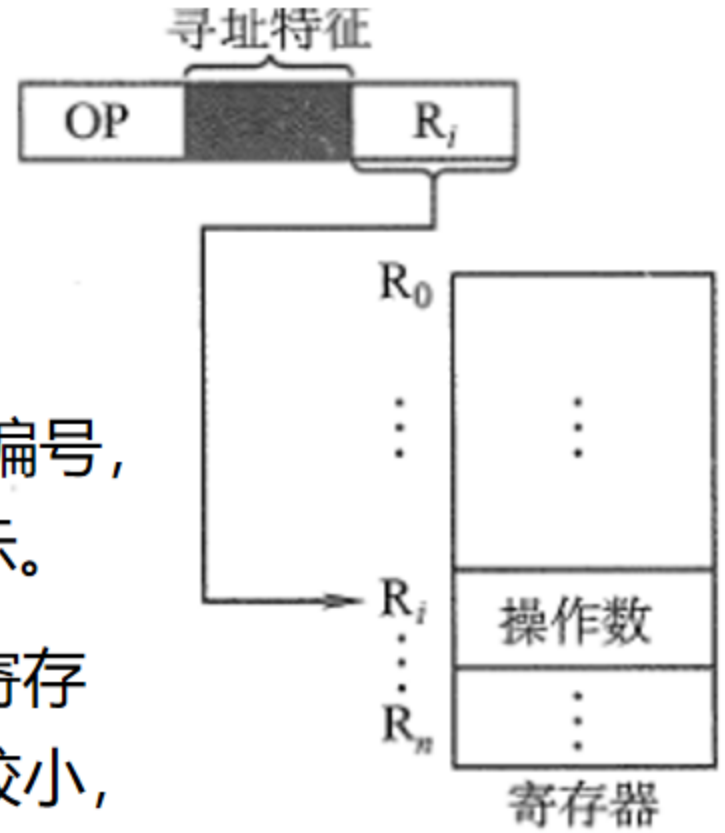
间接寻址的优点是可扩大寻址范围（有效地址EA的位数大于形式地址A的位数），
便于编制程序（用间接寻址可方便地完成子程序返回）；
缺点是指令在执行阶段要多次访存（一次间接寻址需两次访存，多次
间接寻址需根据存储字的最高位确定访存次数）。

访问速度过慢，这种寻址方式并不常用。一般问到扩大寻址范围时，
通常指的是寄存器间接寻址。

2.常见的数据寻址方式

五.寄存器寻址

寄存器寻址是指在指令字中直接给出操作数所在的寄存器编号，即EA = R；，其操作数在由R；所指的寄存器内，如图所示。



寄存器寻址

寄存器寻址的优点是指令在执行阶段不访问主存，只访问寄存器，因寄存器数量较少，对应地址码长度较小，使得指令字短且因不用访存，所以执行速度快，支持向量 / 矩阵运算；

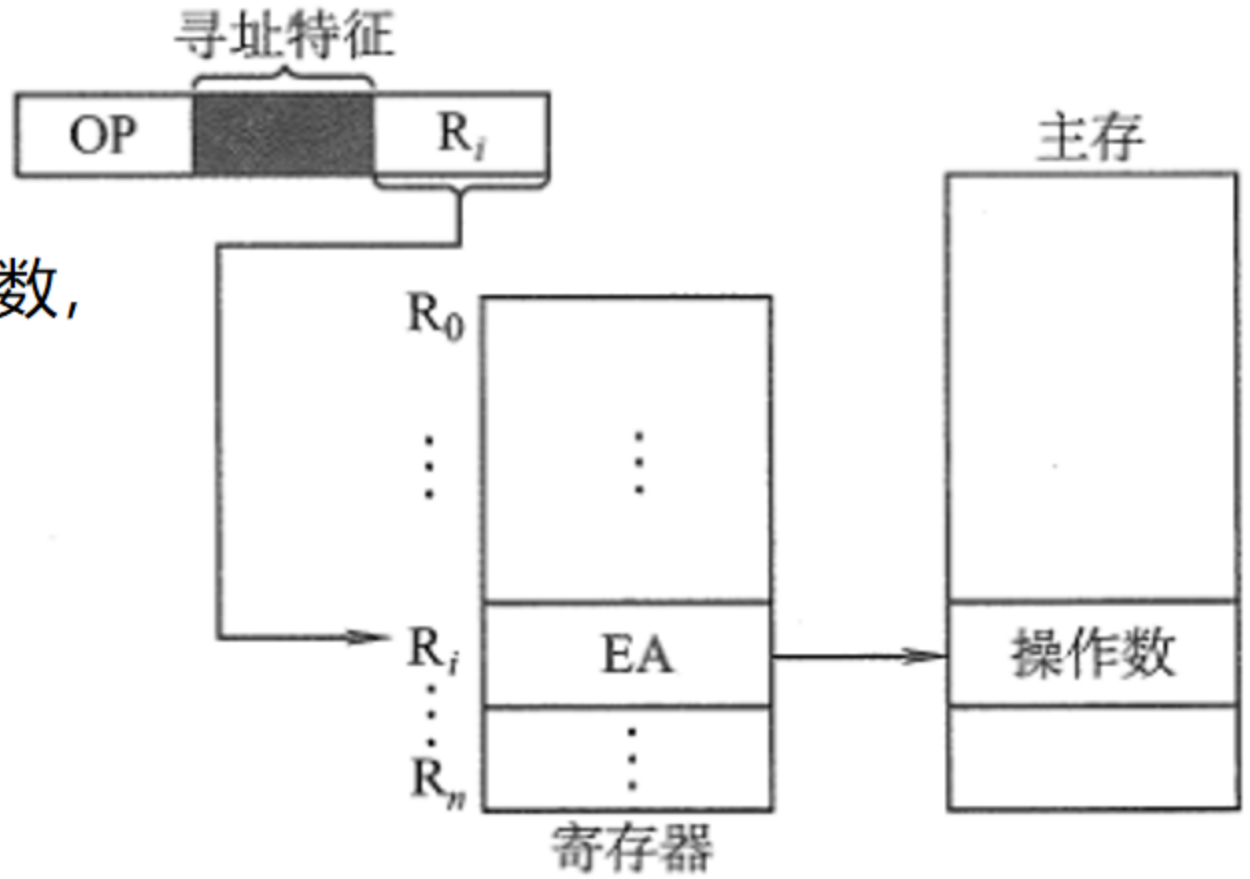
缺点是寄存器价格昂贵，计算机中的寄存器个数有限。

2.常见的数据寻址方式

六.寄存器间接寻址

指在寄存器 R；中给出的不是一个操作数，
而是操作数所在主存单元的地址，
即EA = （R；），如图所示。

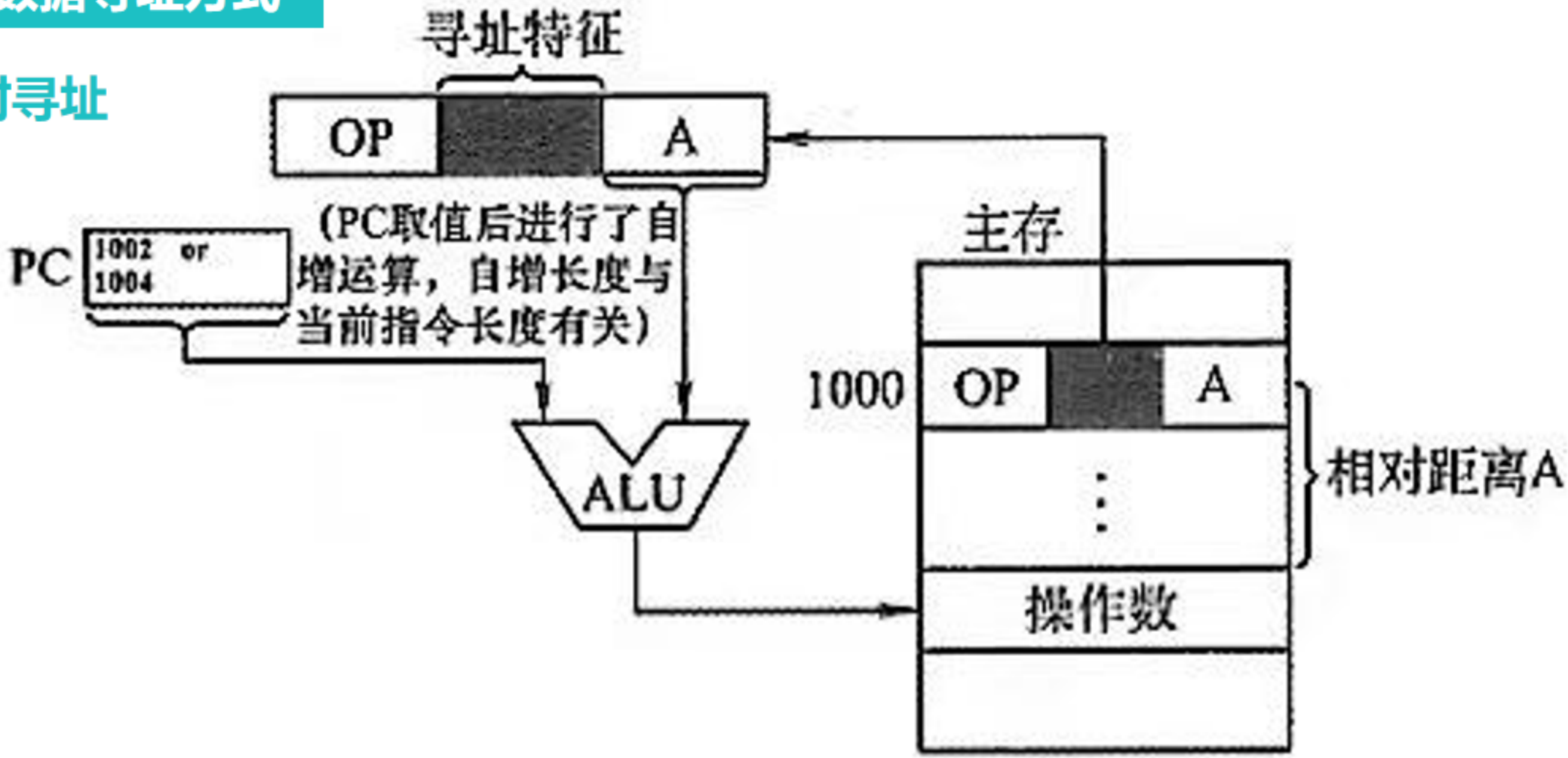
特点：
与一般间接寻址相比速度更快，
但指令的执行阶 段需要访问主存
(因为操作数在主存中）。



寄存器间接寻址

2.常见的数据寻址方式

七.相对寻址



在图中，A的位数决定操作数的寻址范围。

七.相对寻址

把程序计数器（PC）的内容加上指令格式中的形式地址A而形成操作数的有效地址，
即EA = （PC） + A，
其中A是相对于当前指令地址的位移量，可正可负，补码表示，如图所示。

其优点是操作数的地址不是固定的，它随PC值的变化而变化，且与指令
地址之间总是相差一个固定值，因此便于程序浮动。相对寻址广泛应用于转移指令。

注意:对于转移指令JMPA，当CPU从存储器中取出一字节时，
会自动执行（PC） + 1→PC。
若转移指令的地址为X，且占2B，在取出该指令后，PC的值会增2，
即（PC） = X + 2，这样在执行完该指令后，
会自动跳转到X + 2 + A的地址继续执行。

2.常见的数据寻址方式

八.基址寻址

指将CPU中基址寄存器（BR）的内容加上指令格式中的形式地址A而形成操作数的有效地址，即EA = （BR） + A。

其中基址寄存器既可采用专用寄存器，又可采用通用寄存器，如图所示。

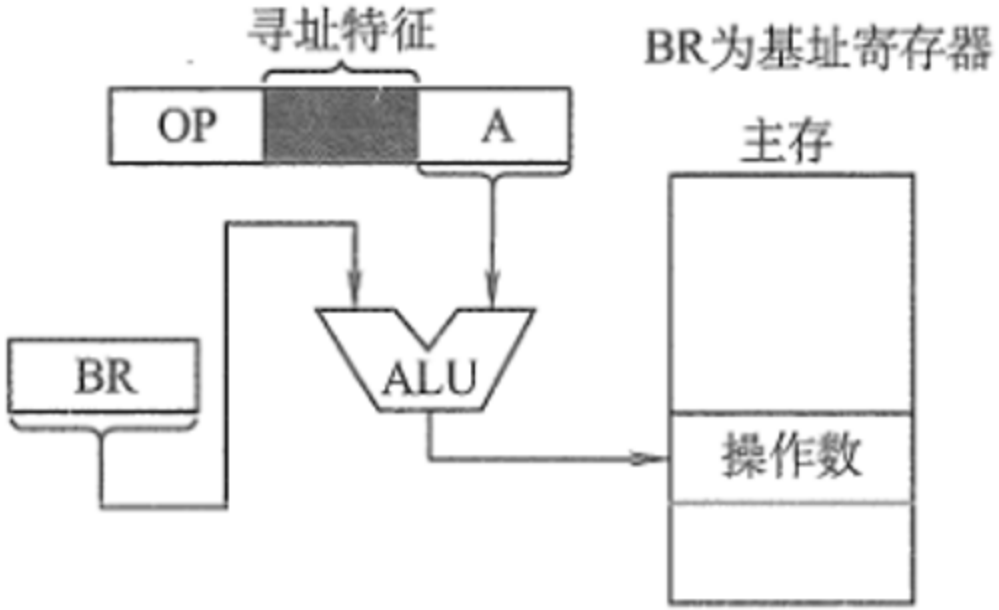
基址寄存器是面向操作系统的，其内容由操作系统或管理程序确定，主要用于解决程序逻辑空间与存储器物理空间的无关性。

在程序执行过程中，基址寄存器的内容不变（作为基地址），形式地址可变（作为偏移量）。

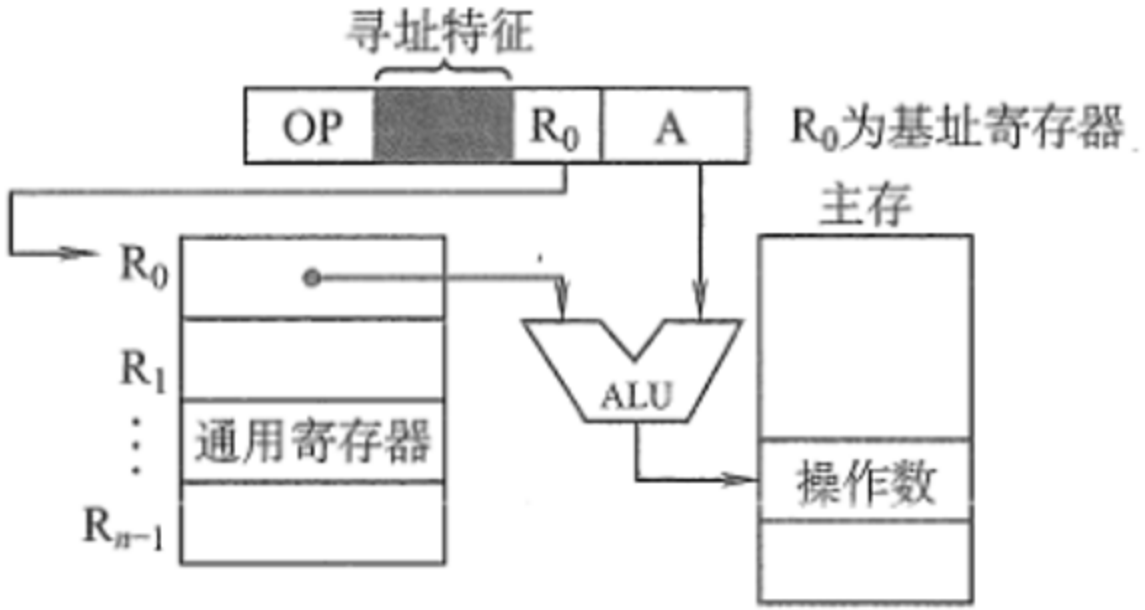
采用通用寄存器作为基址寄存器时，可由用户决定哪个寄存器作为基址寄存器，但其内容仍由操作系统确定。

八.基址寻址

优点是可扩大寻址范围（基址寄存器的位数大于形式地址A的位数）；
用户不必考虑自己的程序存于主存的哪个空间区域，因此有利于多道程序设计，
并可用于编制浮动程序，但偏移量（形式地址A）的位数较短。



(a) 采用专用寄存器BR作为基址寄存器



(b) 采用通用寄存器 R_0 作为基址寄存器

2.常见的数据寻址方式

九.变址寻址

变址寻址是指有效地址EA等于指令字中的形式地址A与变址寄存器IX的内容之和，即EA = （IX） + A，其中IX为变址寄存器（专用），也可用通用寄存器作为变址寄存器。图示为采用专用寄存器IX的变址寻址示意图。

变址寄存器是面向用户的，在程序执行过程中，变址寄存器的内容可由用户改变（作为偏移量），形式地址 A不变（作为基地址）。

其优点是可扩大寻址范围（变址寄存器的位数大于形式地址A的位数）；

在数组处理过程中，可设定A为数组的首地址，

不断改变变址寄存 器IX的内容，

便可很容易形成数组中任一数据的地址，特别适合编制循环程序。

偏移量（变址寄存器IX）的位数足以表示整个存储空间。

九.变址寻址

变址寻址与基址寻址的有效地址形成过程极为相似。本质上，两者区别较大。

基址寻址面向系统，主要用于为多道程序或数据分配存储空间，
因此基址寄存器的内容通常由操作系统或管理程序确定，
在程序的执行过程中其值不可变，而指令字中的A是可变的；

变址寻址立足于用户，主要用于处理数组问题，
在变址寻址中，变址寄存器的内容由用户设定，
在程序执行过程中其值可变，
而指令字中的A是不可变的。

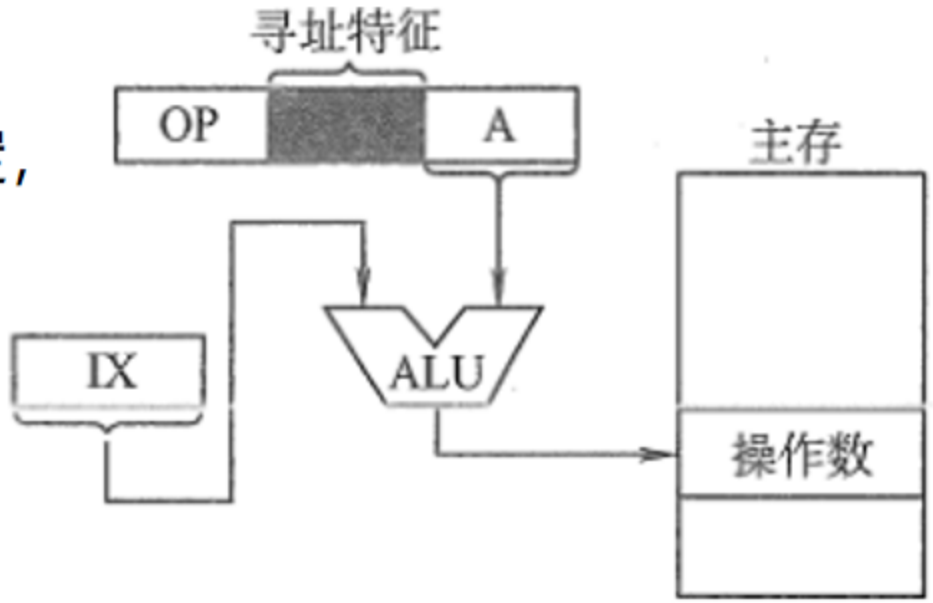


图 4.10　变址寻址

4.3 CISC和RISC的基本概念

1.复杂指令系统计算机（CISC）

随着VLSI技术的发展，硬件成本不断下降，软件成本不断上升，促使人们在指令系统中增加更多、更复杂的指令，以适应不同的应用领域，这样就构成了复杂指令系统计算机（CISC）。

扫码听课，视频讲解更清晰

1.复杂指令系统计算机（CISC）

一.CISC的主要特点如下：

1）指令系统复杂庞大，指令数目一般为200条以上。

2）指令的长度不固定，指令格式多，寻址方式多。

3）可以访存的指令不受限制。

4）各种指令使用频度相差很大。

5）各种指令执行时间相差很大，大多数指令需多个时钟周期才能完成。

6）控制器大多数采用微程序控制。有些指令非常复杂，以至于无法采用硬连线控制。

7）难以用优化编译生成高效的目标代码程序。

2.精简指令系统计算机（RISC）

精简指令系统计算机（RISC）的中心思想是要求指令系统简化，尽量使用寄存器-寄存器操作指令，指令格式力求一致。

2.精简指令系统计算机（RISC）

一.RISC的主要特点如下：

1）选取使用频率最高的一些简单指令，复杂指令的功能由简单指令的组合来实现。

2）指令长度固定，指令格式种类少，寻址方式种类少。

3）只有Load / Store（取数 / 存数）指令访存，其余指令的操作都在寄存器之间进行。

4）CPU中通用寄存器的数量相当多。

5）RISC一定采用指令流水线技术，大部分指令在一个时钟周期内完成。

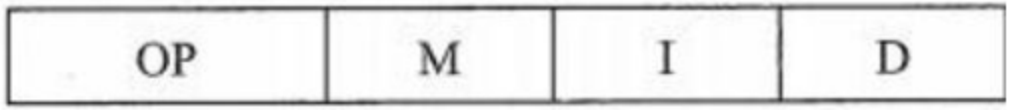
6）以硬布线控制为主，不用或少用微程序控制。

7）特别重视编译优化工作，以减少程序执行时间。

3.CISC和RISC的比较

类别 / 对比项目	CISC	PISC
指令系统	复杂，庞大	简单，精简
指令数目	一般大于200条	一般小于100条
指令字长	不固定	定长
可访存指令	不加限制	只有Load/Store指令
各种指令执行时间	相差较大	绝大多数在同一个周期内完成
各种指令使用频率	相差很大	都比较常用
通用寄存器数量	较少	多
目标代码	难以用优化编译生成高效的目标代码程序	采用优化的编译程序，生成代码较为高效
控制方式	绝大多数为微程序控制	绝大多数为组合u偶极控制
指令流水线	可以通过一定方式实现	必须实现

【题1】某指令格式如下所示。



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(C)。

A.I+D　　B.(I)+D　　C.((I)+D)　　D.((I))+D

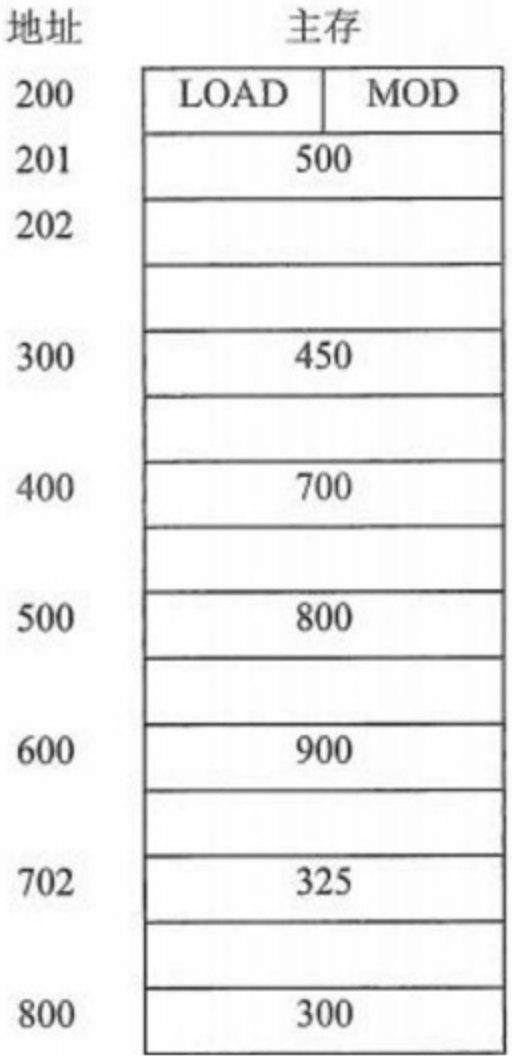
解析：

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的内容之和，即EA = （I） + D

间接寻址是相对于直接寻址而言的，指令的地址字段给出的形式地址不是操作数的真正地址，而是操作数地址的地址，即EA = （D）。

从而该操作数的有效地址是（（I） + D）。

【题2】一条双字长的Load指令存储在地址为200和201的存储位置，该指令将指定的内容装入累加器（ACC）中。指令的第一个字指定操作码和寻址方式，第二个字是地址部分。主存内容示意图如下图所示。PC值为200，R1值为400，XR值为100。



指令的寻址方式字段可指定任何一种寻址方式。请在下列寻址方式中，装入ACC的值。

1）直接寻址。
2）立即寻址。
3）间接寻址。
4）相对寻址。
5）变址寻址。
6）寄存器R1寻址。
7）寄存器R1间接寻址。

扫码听课，视频讲解更清晰

解：1)直接寻址时，有效地址是指令中的地址码部分500，装入ACC的是800。

2)立即寻址时，指令的地址码部分是操作数而不是地址，所以将500装入ACC。

3)间接寻址时，操作数的有效地址存储在地址为500的单元中，由此得到有效地址为800，操作数是300。

4)相对寻址时，有效地址EA = （PC） + A = 202 + 500 = 702，所以装入ACC的操作数是325。这是因为指令是双字长，在该指令的执行阶段，PC的内容已经加2，更新为下一条指令的地址202。

5)变址寻址时，有效地址EA = （XR） + A = 100 + 500 = 600，所以装入ACC的操作数是900。

6)寄存器寻址时，R1的内容400装入ACC。

7)寄存器间接寻址时，有效地址是R1的内容400，装入ACC的操作数是700。