计算机组成原理

第四章 指令系统

刘 超

中国地质大学计算机学院

主要内容

- 指令系统
- 指令格式
- 寻址方式
- 典型指令系统
- RISC 与 CISC

指令系统基本概念

- **计算机的程序是有一系列的机器指令组成的。**
- 机器指令（指令）
  - 计算机能直接识别、执行的某种操作命令。
- 指令系统（指令集）
  - 一台计算机中所有机器指令的集合。
  - 机器硬件设计的依据，也是软件设计的基础。
  - 硬件和软件间的界面，直接影响计算机系统性能

系列计算机

- 基本指令系统相同，基本系统结构相同的计算机。
- 一个系列往往有多种型号，它们在结构和性能上有所差异。同一系列的各机种有共同的指令集而且新推出的机种指令系统一定包含所有旧机种的全部指令,旧机种上运行的各种软件可以不加任何修改便可在新机种上运行，大大减少了软件开发费用。
- IBM，PDP-11，VAX-11，Cray，Intel-x86, Pentium
- 系列计算机主要是解决软件兼容的问题。新计算机中必须包含老计算机的指令系统，保证软件向上兼容，保护用户投资。

指令系统性能特性

- 完备性：指令丰富，功能齐全，使用方便。
- 有效性：程序占空间小，执行速度快。
- 规整性：
  - 对称性：在指令系统中所有的寄存器和存储器单元都可同等对待，所有的指令都可使用各种寻址方式
  - 匀齐性：一种操作性质的指令可以支持各种数据类型
  - 一致性：指令长度和数据长度有一定的关系，以方便处理和存取
- 兼容性：系列机软件向上兼容，即低档机上运行的软件可以在高档机上运行。

指令系统的发展与性能要求

- 低级语言与高级语言关系

	比较内容	高级语言	低级语言
1	对程序员的训练要求 (1)通用算法 (2)语言规则 (3)硬件知识	有 较少 不要	有 较多 要
2	对机器独立的程度	独立	不独立
3	编制程序的难易程度	易	难
4	编制程序所需时间	短	较长
5	程序执行时间	较长	短
6	编译过程中对计算机资源的要求	多	少

主要内容

- 指令系统
- 指令格式
- 寻址方式
- 典型指令系统
- RISC 与 CISC

指令格式

- 表示一条指令的机器字，称为指令字，简称指令。
- 指令格式：用二进制代码表示指令的结构形式。
  - 操作码字段表征指令的操作特性与功能；
  - 地址码字段通常指定参与操作的操作数的地址。
- 一条指令的结构可用如下形式来表示：

指令格式:操作码(OP)和地址码(AC)

- 指令系统的每一条指令都有一个操作码，它表示该指令应进行什么性质的操作。
- 不同的指令用操作码字段的不同编码来表示，每一种编码代表一种指令。
- 操作码字段的位数取决于计算机指令系统的规模
  - $L_{OP}=〔\log_2 n〕$
- **两种类型操作码：**
  - 固定长度操作码：便于译码，扩展性差
  - 可变长度操作码：能缩短指令平均长度
  - 例： IBM 370机，该机字长32位，16个通用寄存器R0～R15 ，共有183条指令；指令的长度可以分为16位、32位和48位等几种，所有指令的操作码都是8位固定长度。

指令格式:**操作码**(OP)和地址码(AC)

- 地址码通常指定参与操作的操作数的地址，包括：被操作数，操作数，操作结果 。

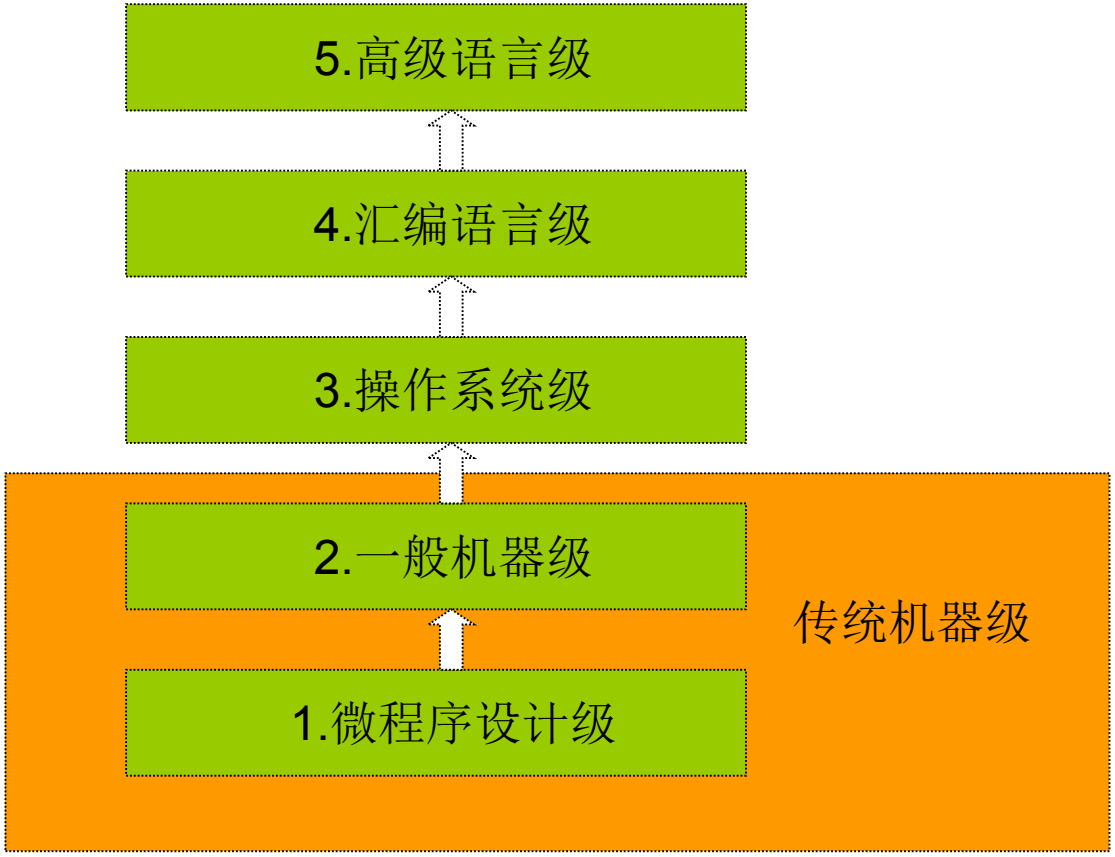
指令分类方法

- 按计算机系统的层次结构分类
- 按操作数个数分类
- 按操作数物理位置分类

计算机系统层次结构对指令分类

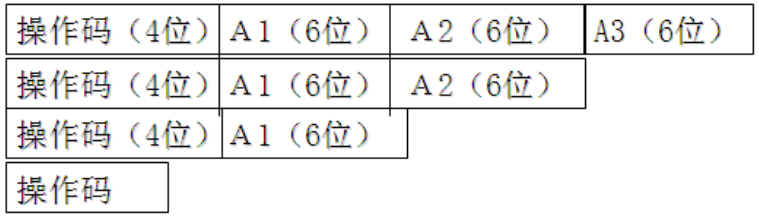
- 微指令：
  - 微程序级的指令，属于硬件层面。
- 机器指令：
  - 介于微指令和宏指令之间的指令，每一条指令可以完成一个独立的算术运算或逻辑运算操作。
- 宏指令：
  - 若干机器指令组成的软件指令，属于软件。
- **本章所讨论的指令，是机器指令**。一台计算机中所有机器指令的集合，称为这台计算机的指令系统。
  - 指令系统是表征一台计算机性能的重要因素，它的格式与功能不仅直接影响到机器的硬件结构，而且也直接影响到系统软件，影响到机器的适用范围。

计算机系统的层次结构



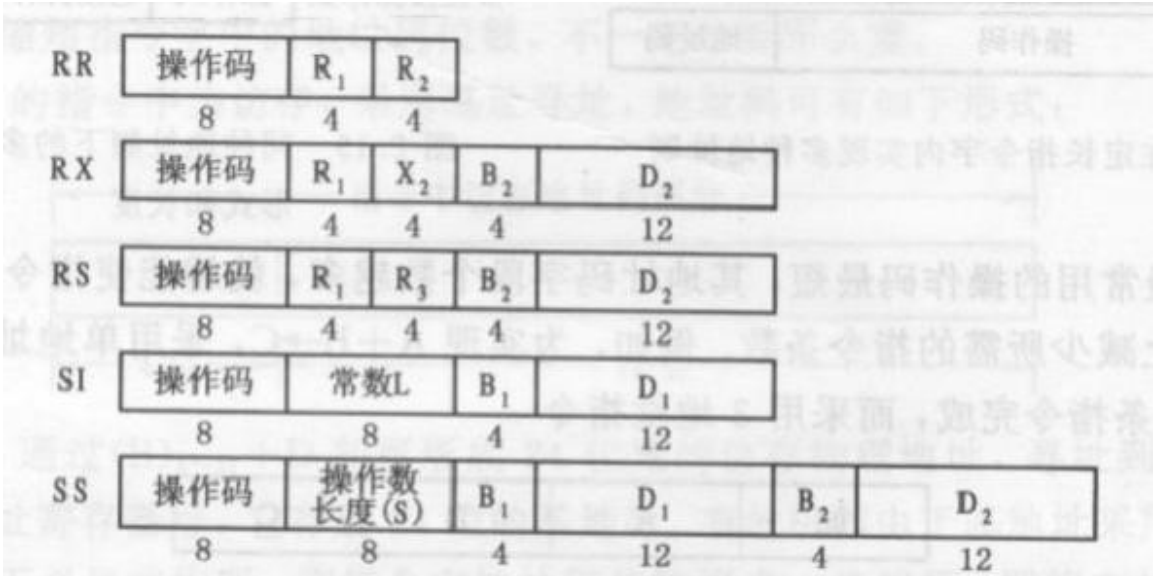
按操作数个数分类

- 三地址指令，有三个操作数地址A1，A2和A3。
- 二地址指令，常称双操作数指令，它有两个地址码字段A1和A2，分别指明参与操作的两个数在内存中或运算器通用寄存器的地址，其中地址A1兼做存放操作结果的地址。
- 一地址指令，单操作数指令。
- 零地址指令
- 举例：
  - 零地址指令，如CLR，STP，NULL
  - 一地址指令，如INC R1
  - 二地址指令，如ADD R1，R2
  - 三地址指令，如ADD R1，R2，R3



按操作数的物理位置分类

- 访问内存
  - 存储器—存储器（SS）型
- 访问寄存器
  - 寄存器—寄存器（RR）型
- 访问内存和寄存器型
  - 寄存器—存储器（RS）型



访问速度上来看：RR最快，SS最慢

指令字长度

- 指令字中包含二进制代码的位数，称为**指令字长度**。
- **机器字长**是指计算机能直接处理的二进制数据的位数，它决定了计算机的运算精度。
- 指令字长度=机器字长度的指令称为**单字长指令**
- 指令字长度=半个机器字长度的指令称为**半字长指令**
- 指令字长度=两个机器字长度的指令称为**双字长指令**
- 用多字长指令的目的？
  - 优点是提供足够的地址位来解决访问内存任何单元的寻址问题。
  - 缺点是必须两次或多次访问内存以取出整条指令，这就降低了CPU的运算速度，同时又占用了更多的存储空间。

指令字长度

- 等长指令字结构
  - 在指令系统中，各种指令字长度是相等的。
  - 指令字结构简单，且指令字长度是不变的。
- 变长指令字结构
  - 在指令系统中，各种指令字长度随指令功能而异。
  - 指令字结构灵活，能充分利用指令长度，但指令的控制较复杂。

指令字助记符

- 由于硬件只能识别1和0，所以采用二进制操作码是必要的，但是我们用二进制来书写程序却非常麻烦。
- 为便于书写和记忆而设定的，与机器指令一一对应。每条指令通常用3个或4个英文缩写字母来表示。这种缩写码叫做**指令助记符**。
- 指令助记符由汇编程序转换成它们相对应的二进制操作码。不同的计算机中，指令助记符的规定是不一样的。必须用汇编语言翻译成二进制代码。
- 当指令的操作码用助记符表示，而地址及其寻址特征也用符号表示时，就成为**汇编语言**，这些符号称为汇编符号，用汇编符号表示的指令格式，就称为汇编格式。
- 例子：ADD， SUB， MOV， JMP， STR， LDA

指令格式举例

八位微型计算机的指令格式

8位微型机字长只有8位，指令结构是一种可变字长形式，包含单字长、双字长、三字长指令等多种。

操作码　　单字长指令

操作码　操作数地址　　双字长指令

操作码　操作数地址 1　操作数地址2　三字长指令

内存按字节编址，所以单字长指令每执行一条指令后，指令地址加 1 。双字长指令或三字长指令每执行一条指令时，指令地址要加2或加3，可见多字长的指令格式不利于提高机器速度。

PDP-11 指令格式举例

PDP/11系列机指令字长16位，其指令格式如下表所示。

4.3 PDP/11系列机指令格式

指令类型 \ 指令位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
单操作数指令	操作码（10位）										目标地址(6位)					
双操作数指令	操作码(4位)				源地址(6位)						目标地址(6位)					
转移指令	操作码(8位)								位移量(8位)							
转子指令	操作码(7位)							寄存器号								
子程序返回指令	操作码(13位)															
条件码操作指令	操作码(11位)											S	N	Z	V	C

从表中看出，在**PDP/11**中，操作码字段是不固定的，其长度也是不相同的。这样做可以扩展操作码以包含较多的指令。但是操作码字段不固定，对控制器的设计来说必将复杂化。

pentium指令格式

- pentium机的指令字长度是可变的：从 1 字节到12字节，还可以带前缀，指令格式如下所示。

0或1　　0或1　　0或1　　0或1(字节数)

指令前缀	段取代	操作数长度取代	地址长度取代

1或2　　0或1　　0或1　　0,1,2,4　0,1,2,4(B)

操作码	Mod	Reg或操作码	R/M	比例S	变址I	基址B	位移量	立即数

2位　　3位　　3位　　2位　　3位　　3位

这种非固定长度的指令格式是典型的CICS结构特征。一是为了与它的前身80486保持兼容，二是希望能给编译程序写作者以更多灵活的编程支持。

操作数长度取代前缀和地址长度取代前缀　在实地址模式下，操作数和地址的默认长度是16位；在保护模式下，若段描述符的D＝1，操作数和地址的默认长度是32位，若D＝0，二者的默认长度是16位。

指令本身由操作码字段、Mod-R/M字段、SIB字段、位移量字段、立即数字段组成。除操作码字段外，其他四个字段都是可选字段。

Pentium 4 指令格式

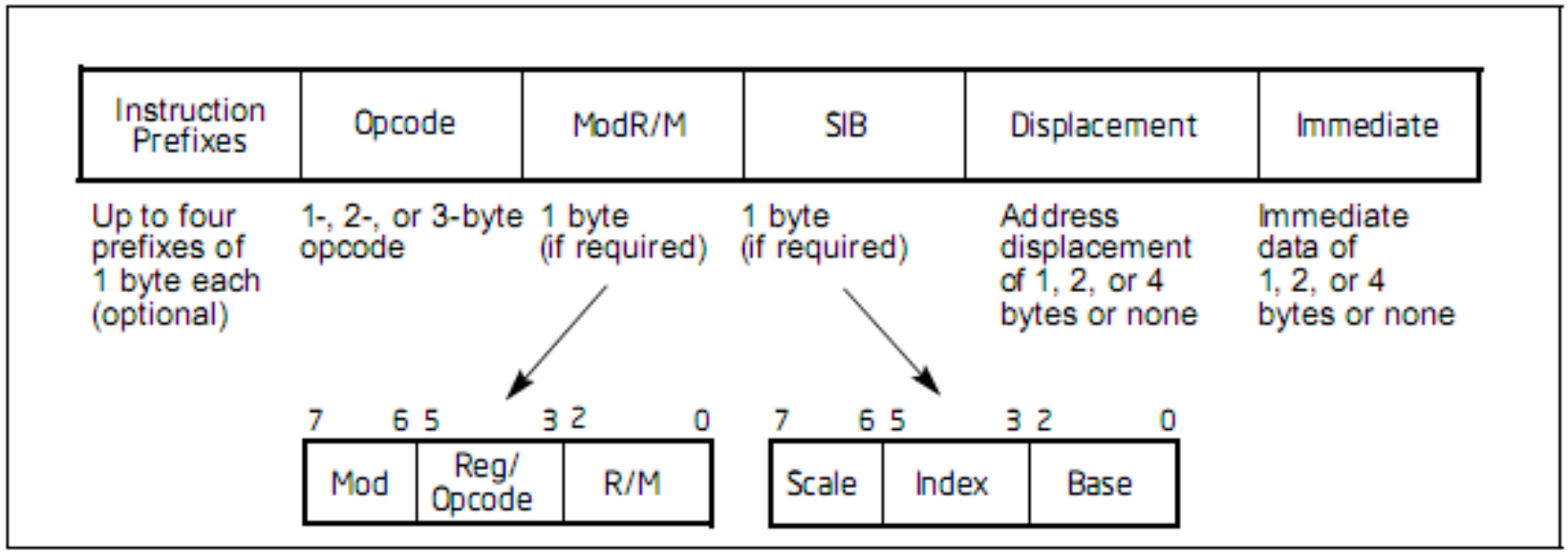


Figure 2-1. Intel 64 and IA-32 Architectures Instruction Format

- 详见Intel CPU文档:Intel 64 and IA-32 Architectures Software Developer's ManualVolume 2A: Instruction Set Reference

- [例1] 指令格式如下所示，其中OP为操作码，试分析指令格式的特点。

15 9 7 4 3 0

OP -------- 源寄存器 目标寄存器

- [解]:

(1)单字长二地址指令。

(2)操作码字段OP可以指定128条指令。

(3)源寄存器和目标寄存器都是通用寄存器（可分别指定16个），所以是RR型指令，两个操作数均在寄存器中。

(4)这种指令结构常用于算术逻辑运算类指令。

- [例2] 指令格式如下所示，OP为操作码字段，试分析指令格式特点。

15 10 7 4 3 0

OP -------- 源寄存器 变址寄存器

位移量(16位)

- [解]:

(1)双字长二地址指令，用于访问存储器。

(2)操作码字段OP为6位，可以指定64种操作。

(3)一个操作数在源寄存器（共16个），另一个操作数在存储器中（由变址寄存器和位移量决定）所以是RS型指令。

主要内容

- 指令系统
- 指令格式
- 寻址方式
- 典型指令系统
- RISC 与 CISC

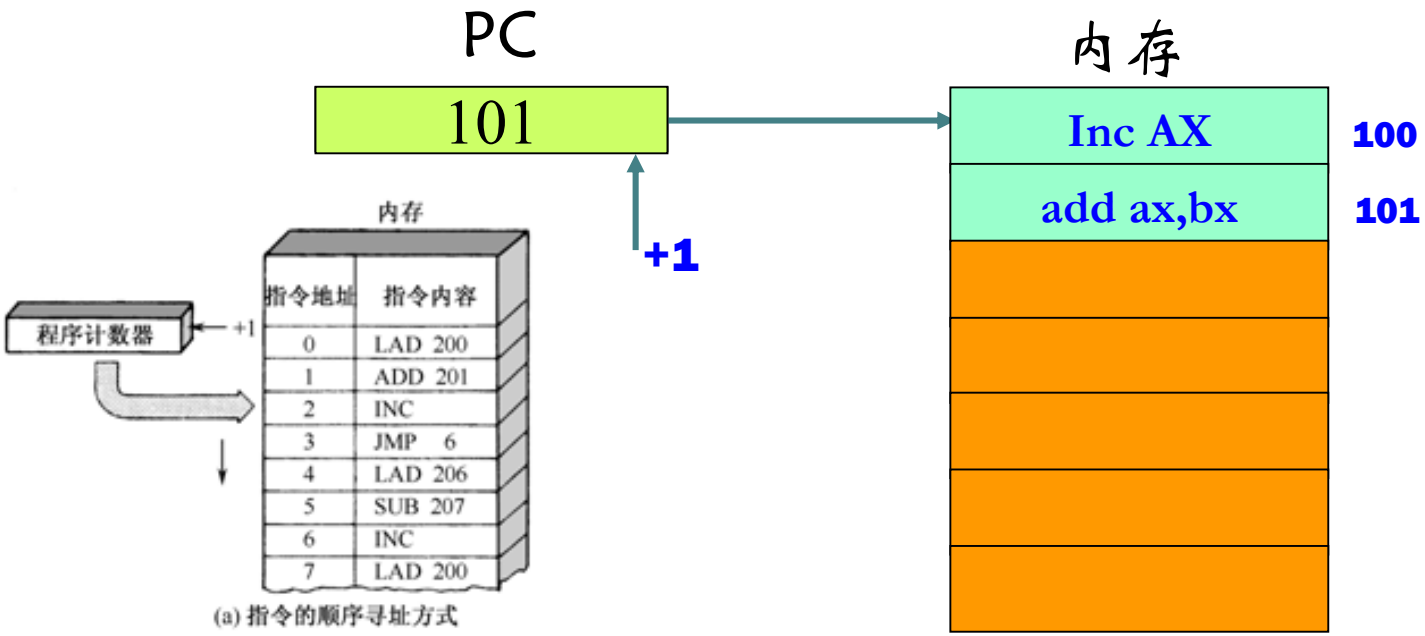
寻址方式

- 操作数或指令在存储器中的地址：某个操作数或某条指令存放在某个存储单元时其存储单元的编号。
- 在存储器中，操作数或指令写入或读出的方式，有三种：地址指定方式、相联存储方式和 堆栈存储方式。
- 当采用地址指定方式时，形成操作数或指令地址的方式，称为寻址方式。
- 寻址方式主要解决2个问题：确定本条指令中各操作数的地址；下一条指令的地址。
  - 即：寻址方式分为指令寻址方式和数据寻址方式。
  - 指令的寻址方式有两种，一种是顺序寻址方式，另一种是跳跃寻址方式。

指令的寻址方式

顺序寻址方式

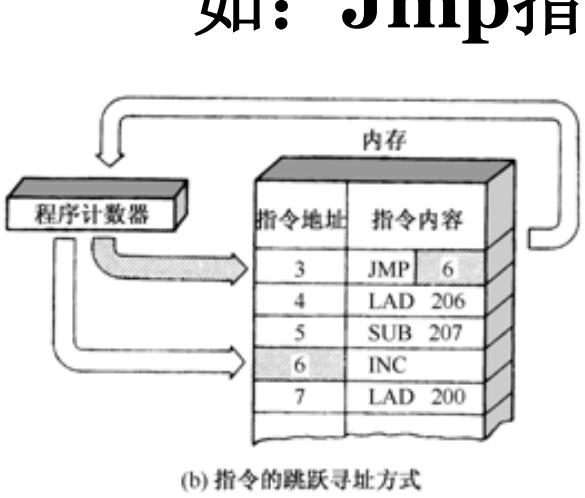
- 指令地址在内存中按顺序安排，当执行一段程序时，通常是一条指令接一条指令的顺序执行。为此，必须使用程序计数器（又称指令指针寄存器）PC来计数指令的顺序号，该顺序号就是指令在内存中的地址。(PC)+1->PC



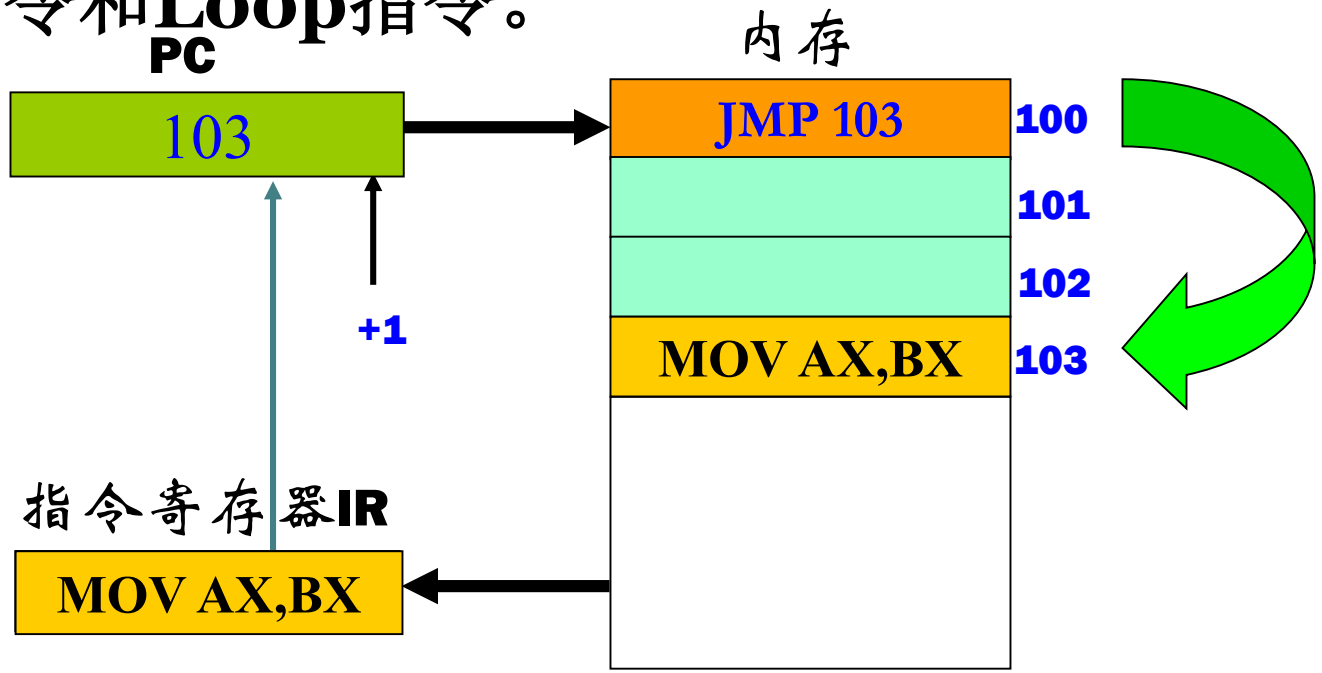
指令的寻址方式

跳跃寻址方式

- 当程序转移执行的顺序时，指令的寻址就采取跳跃寻址方式。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。程序跳跃后，按新的指令地址开始顺序执行。如：Jmp指令和Loop指令。



(b) 指令的跳跃寻址方式



操作数寻址方式

形成操作数有效地址的方法，称为操作数寻址方式。
操作数通常放在哪儿呢？

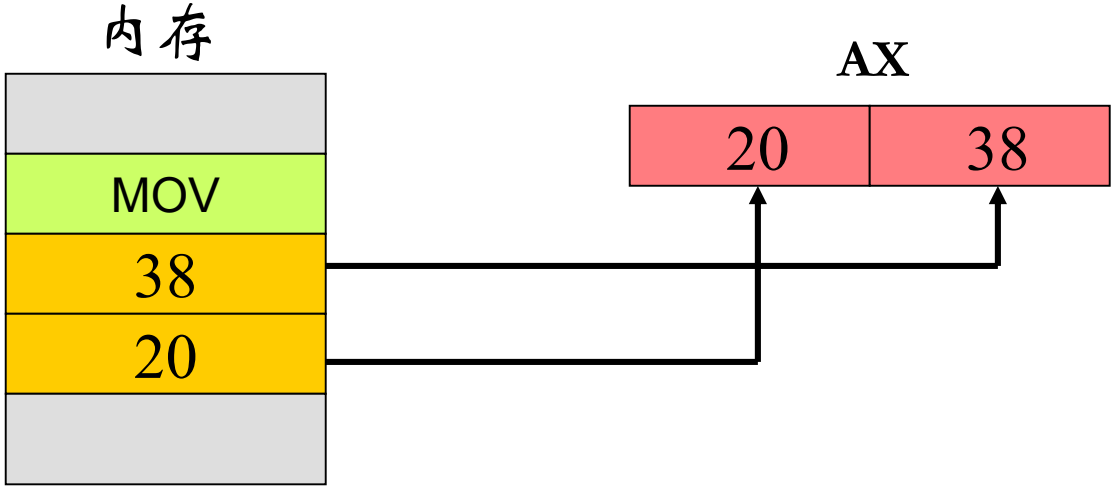
- 隐含寻址
- 立即寻址
- 寄存器寻址
- 寄存器间接寻址
- 直接寻址
- 间接寻址
- 相对寻址
- 基址和变址寻址
- 块寻址
- 段寻址

隐含方式

- 操作数的寻址方式，就是形成操作数的有效地址的方法。
- 指令中隐含着操作数的地址。
- 如ADD A中的累加器

立即寻址(Immediate Addressing)

- 地址码字段是操作数本身。
  - 例： MOV AX, 2038H （2038H→AX）



寄存器寻址(Register Addressing)

- 操作数在CPU的内部寄存器中
  - AX,BX,CX,DX
  - MOV AX, BX

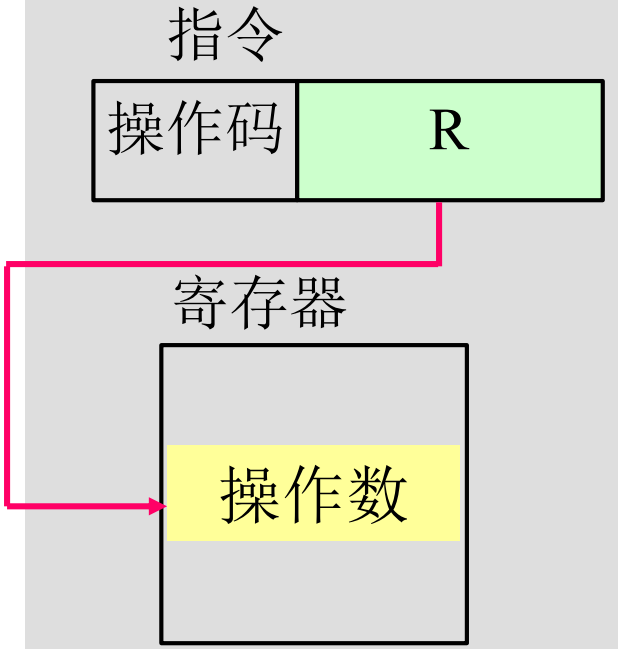
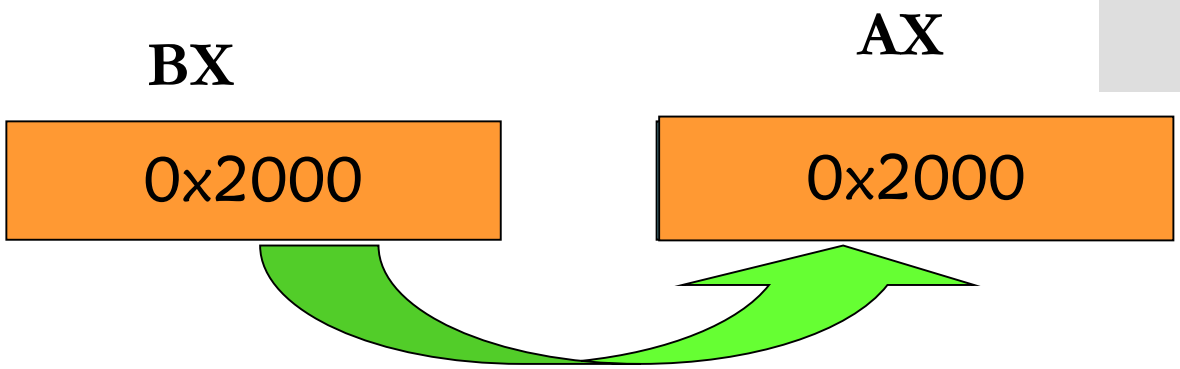


图 4-3 寄存器寻址



寄存器间接寻址(Register Indirect Addressing)

- 寄存器间址，看做指针。
- MOV R1, #0
- MOV R2,#A
- ADD R1,(R2)

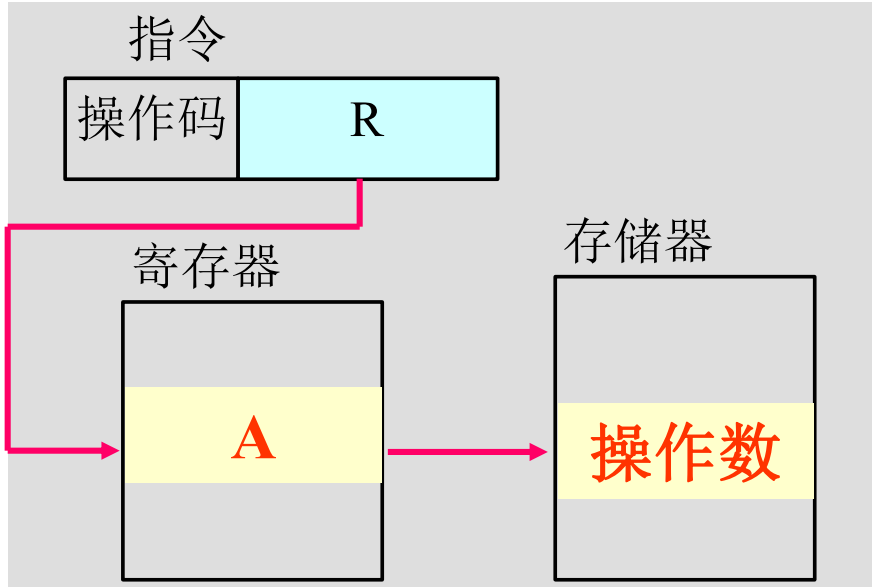


图 4-6　寄存器间接寻址

直接寻址(Direct Addressing)

- 地址码字段直接给出操作数在内存的地址
- **INC 1000**
- MOV AX , [200]

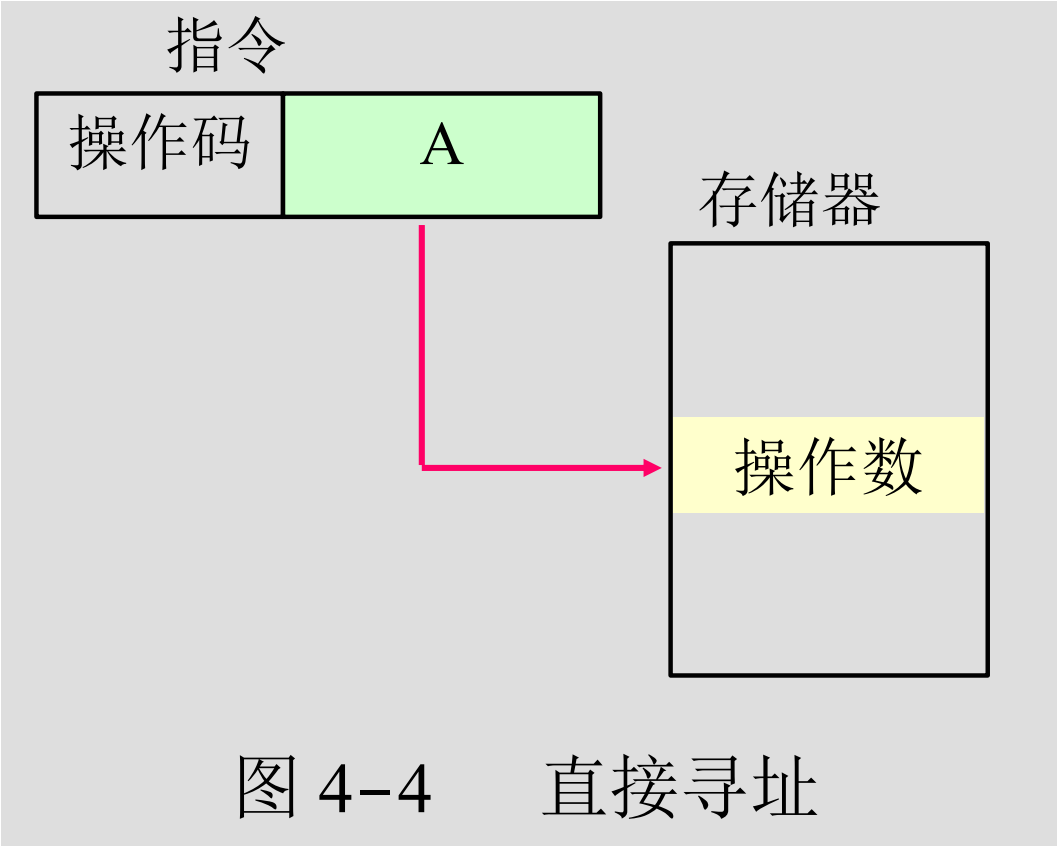


图 4-4　　直接寻址

间接寻址(Indirect Addressing)

- A单元的内容是操作数地址, A是操作数地址的地址。
- 是需要访问两次内存,速度慢， 已被淘汰。

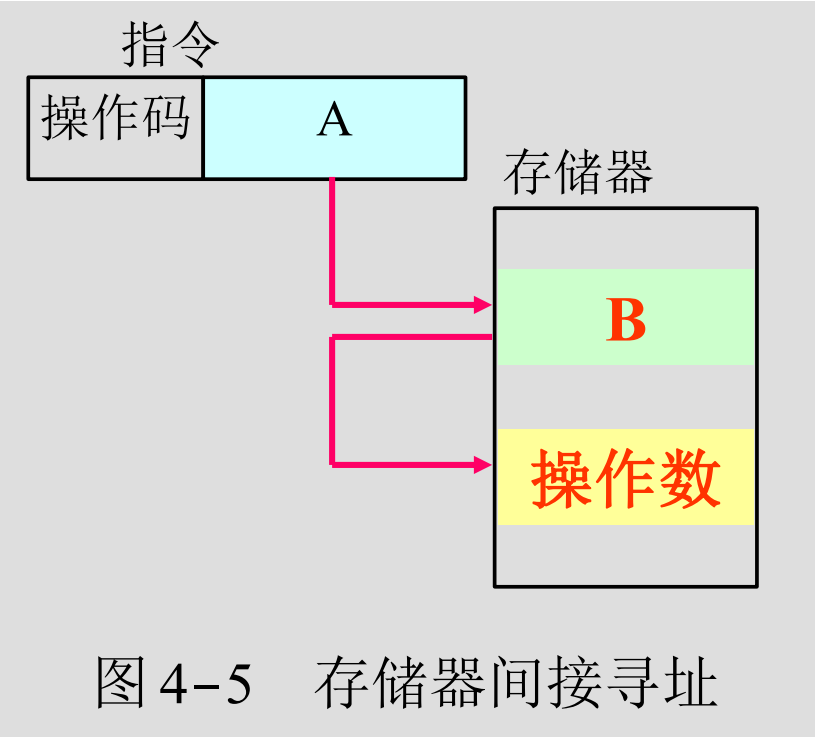
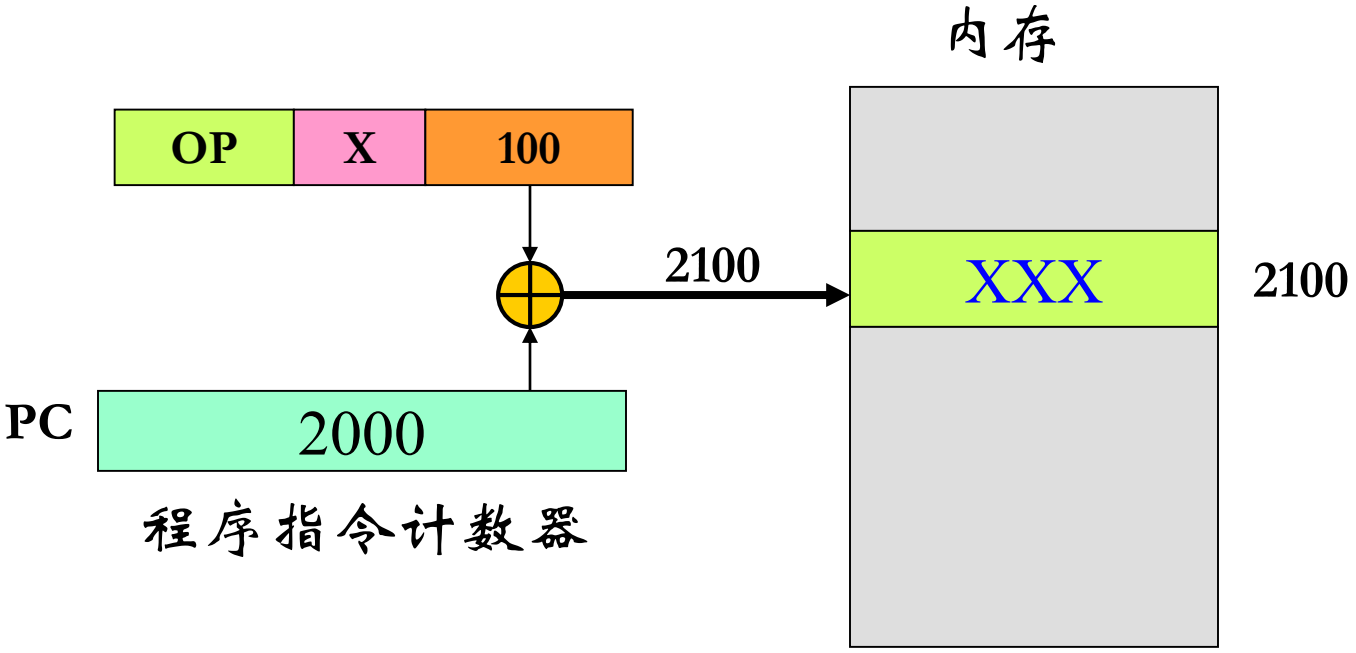


图 4-5　存储器间接寻址

相对寻址 (Relative Addressing)

- 指令中的操作数地址部分加上PC的内容作为操作数的实际地址。如： INC 100(PC)



基址和变址寻址(Indexed Addressing)

- 指定一个寄存器B/或者I,其存放基址或者变址。例如：**MOV R2,A(R1)，目的操作数是寄存器寻址，源操作数变址寻址，A是偏移量。**

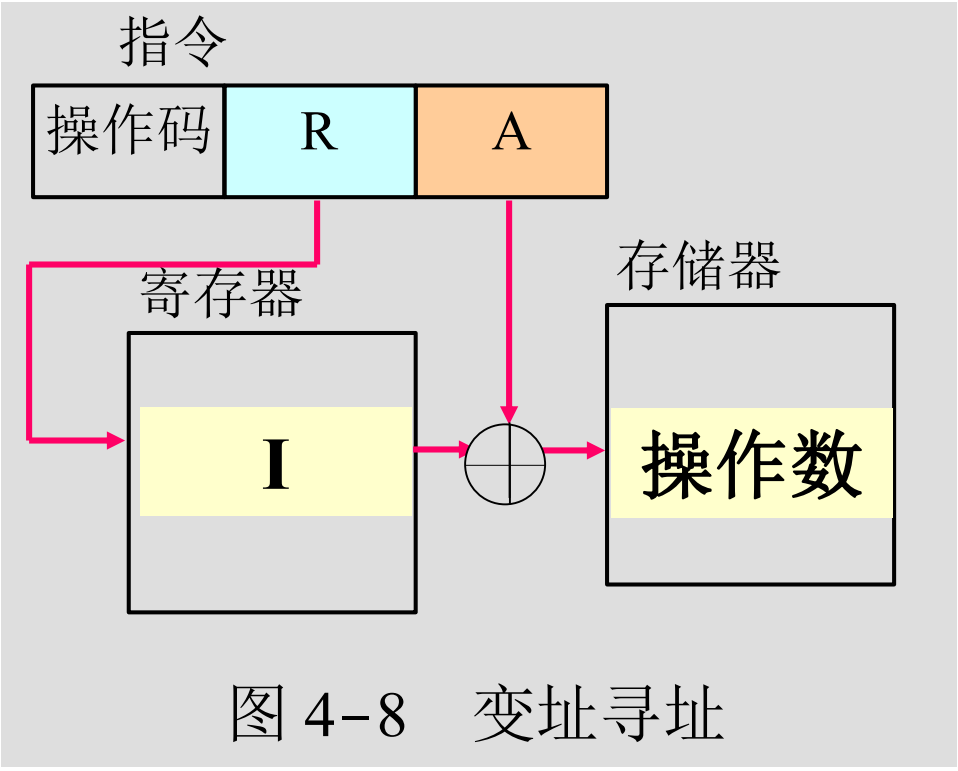
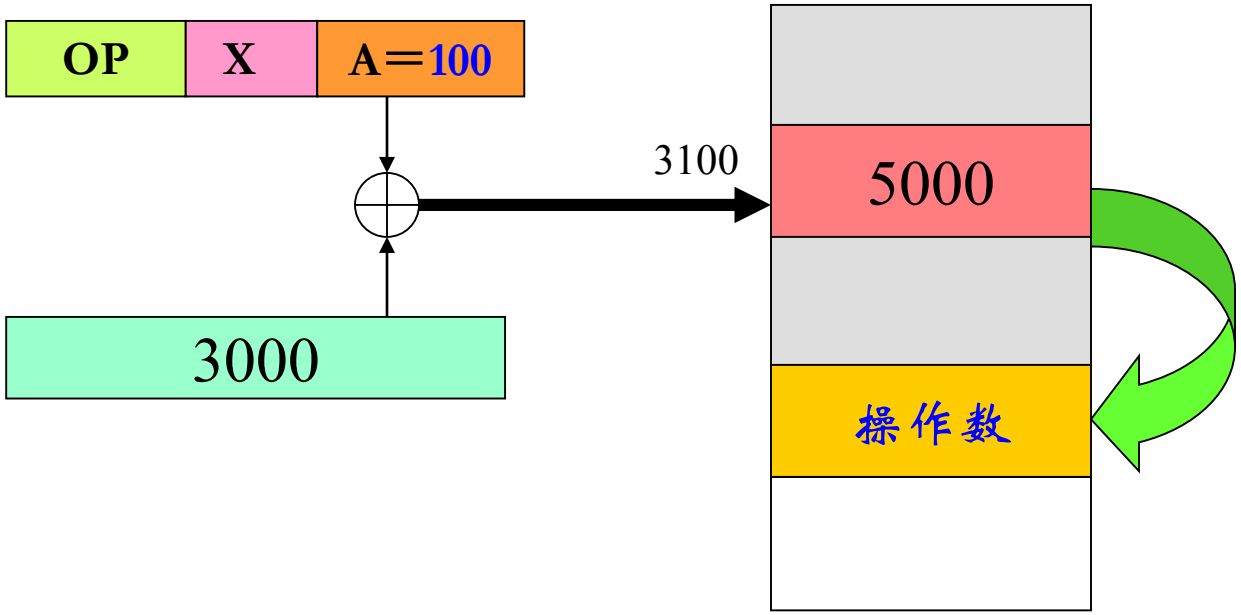


图 4-8　变址寻址

- 变址寻址：便于数组访问
- 基址寻址：可扩大寻址范围，可实现程序浮动

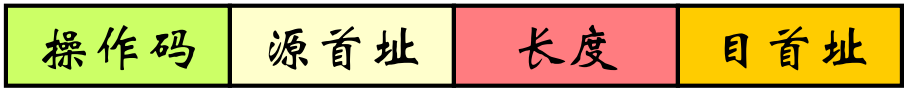
复合寻址(Composite Addressing)

- 将间址,相对,变址,基址等寻址方式组合。
- 变址间址: 先变址,后间址。 **E=((R)+A)**

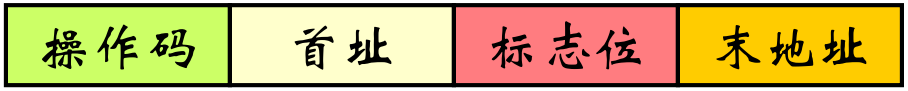


块寻址

- 用于I/O指令，对顺序连续的成块数据字进行寻址。
  - 压缩程序的长度，加快执行速度。在两个部件间的数据交换；程序、数据块的浮动。数据搬移子程序。
- 若块的长度可变，格式如下：

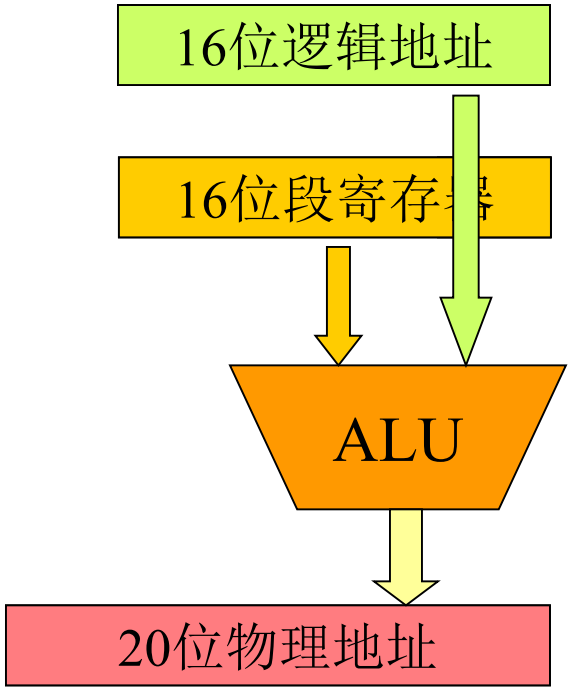


数据块定长时：



段寻址

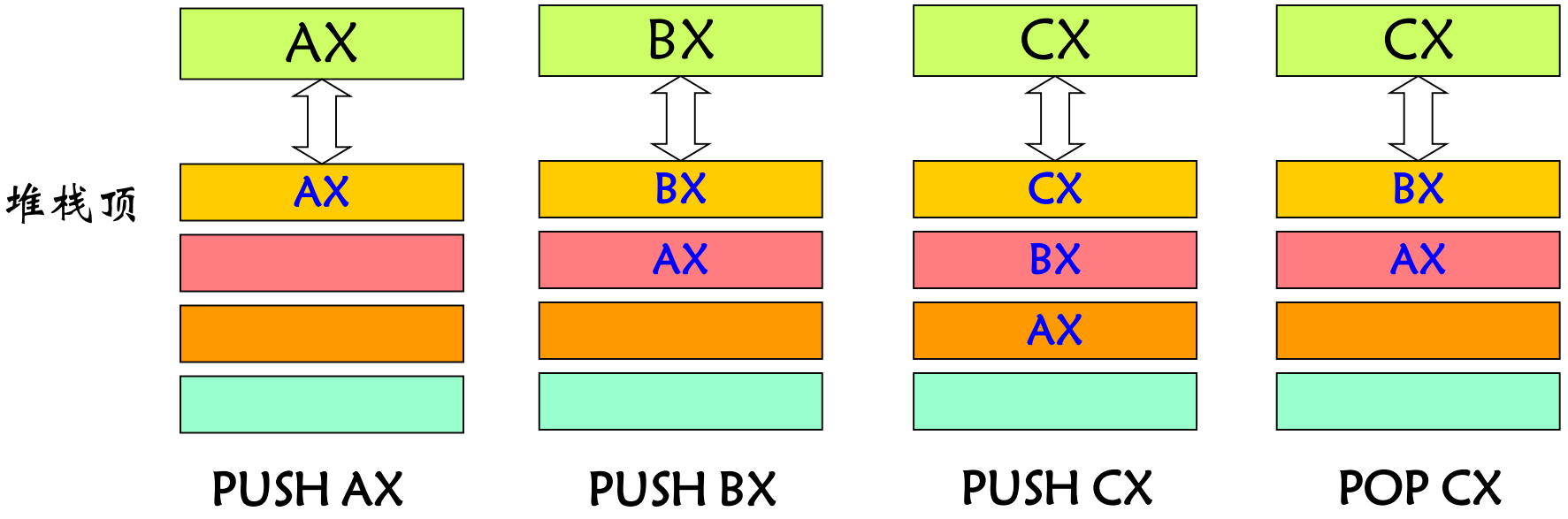
- Intel 8086/8088微机中，ALU16位运算，但其寻址范围可到1M，即地址有20位。16位段寄存器左移4位，加上16位偏移量计算的到实际20位物理地址。
- 实质是基值寻址。



堆栈寻址方式

- 堆栈------一组能存取数据的暂时存储单元。
- 串联堆栈
  - 一组专门的寄存器，一个R保存一个数据。
  - 数据的传送在栈顶和通用寄存器之间进行。
  - 快速：在CPU内部实现
  - 串行：进栈和出栈涉及到栈内所有其它数据的移动；
  - 破坏性读出：读数据的同时也离开了堆栈；
  - 栈容量有限：取决于CPU内堆栈专用寄存器的数量；
  - 栈顶不动，数据移动。

堆栈寻址方式

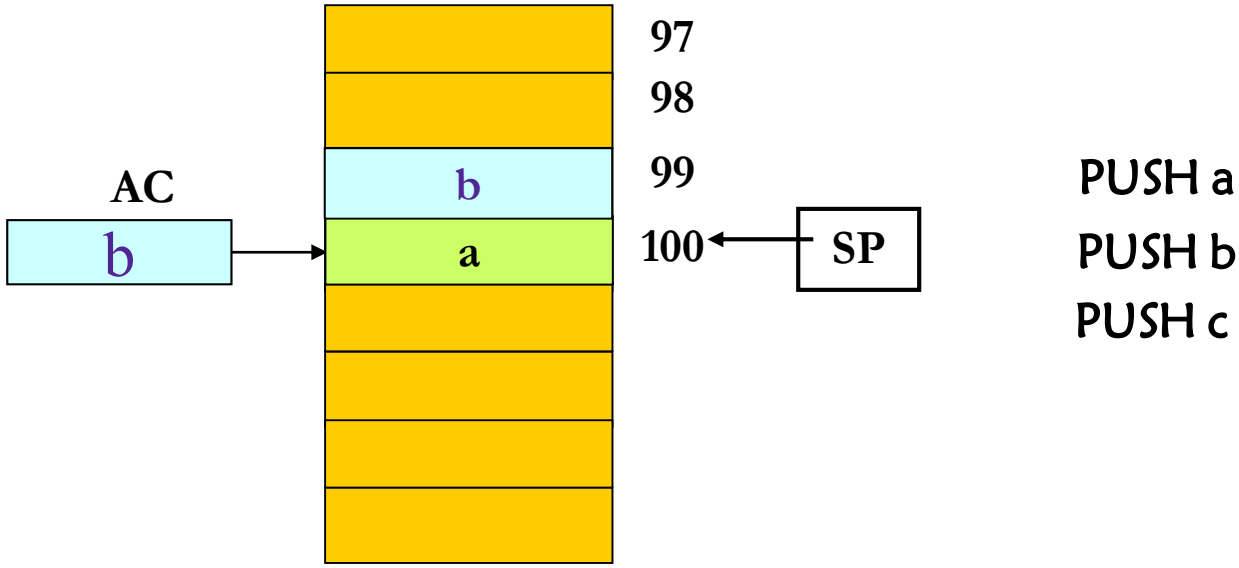

堆栈顶
AX
AX
BX
BX
AX
CX
CX
BX
AX
CX
BX
AX
PUSH AX
PUSH BX
PUSH CX
POP CX

存储器堆栈

- 用一部分主存空间作堆栈称为存储器堆栈。
  - 堆栈的数目、长度可随意指定
  - SP---堆栈指示器(栈指针)， CPU中一个专门寄存器，SP内容是栈顶单元地址。改变SP内容即可移动栈顶的位置。
  - 堆栈操作期间，堆栈中数据不动，栈顶移动
  - 非破坏性读出

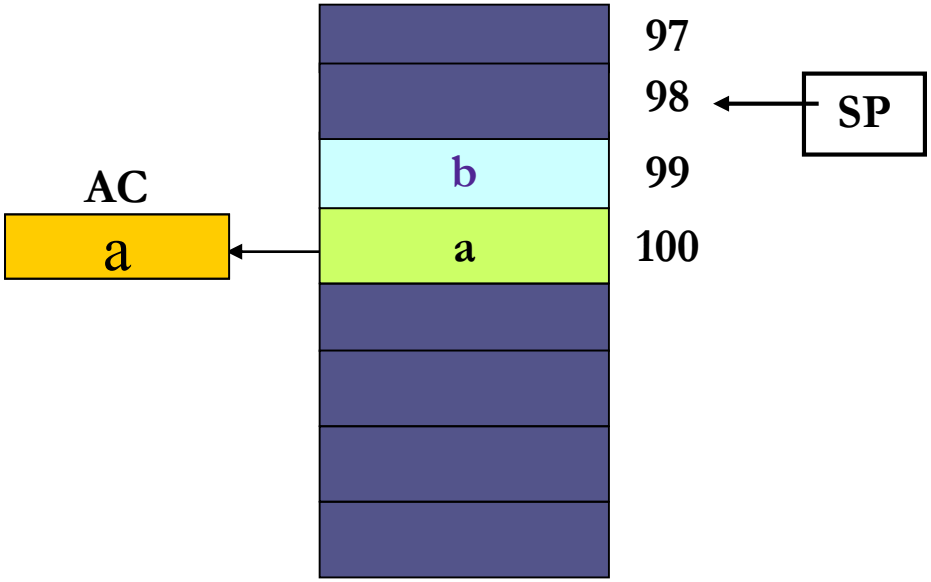
进栈

- 进栈--------累加器中的数送堆栈保存.
- (AC) →堆栈MSP　堆栈指针(sp) －1 →sp



出栈

- 出栈--------将堆栈中的数取出送累加器
- 堆栈指针(sp) ＋1 → sp　（堆栈MSP）→ AC



Pentium寻址方式

方式	算法
立即	作数=A
寄存器	LA=R
偏移量	LA=(SR)+A
基址	LA=(SR)+(B)
基址带偏移量	LA=(SR)+(B)+A
比例变址带偏移量	LA=(SR)+(I)×S+A
基址带变址和偏移量	LA=(SR)+(B)+(I)+A
基址带比例变址和偏移量	LA=(SR)+(B)+(I)×S+A
相对	LA=(PC)+A

Pentium 4 &UltraSPARC 寻址方式

- Pentium 4支持立即寻址，直接寻址，寄存器寻址，寄存器间接寻址，变址寻址等。
- UltraSPARC III 除了对内存进行的寻址指令(Load，Store，Asyn)，所有指令都是立即寻址和寄存器寻址。

寻址方式举例

[例3] 一种二地址RS型指令的结构如下所示：

6位	4位	1位	2位	16位
OP	---通用寄存器	I	X	偏移量D

其中**I为间接寻址标志位，X为寻址模式字段，D位偏移量字段。**通过I，X，D的组合，可构成下表所示的寻址方式。请写出六种寻址方式的名称。

寻址方式	I	X	有效地址E算法	说明
(1)	0	00	E=D	
(2)	0	01	E=(PC)+D	PC为程序计数器
(3)	0	10	E=(R_2)+D	R_2为变址寄存器
(4)	1	11	E=(R3)	
(5)	1	00	E=(D)	
(6)	0	11	E=(R1)+D	R_1为基址寄存器

[解]:

(1)直接寻址　(2)相对寻址　(3)变址寻址

(4)寄存器间接寻址　(5)间接寻址　(6)基址寻址

寻址方式举例

[例4] 某16位机器所使用的指令格式和寻址方式如下所示，该机有两个20位基址寄存器，四个16位变址寄存器，十六个16位通用寄存器，指令汇编格式中的S（源），D（目标）都是通用寄存器，M是主存中的一个单元。三种指令的操作码分别是MOV（OP）＝（A）H，STA（OP）＝（1B）H，LDA（OP）＝（3C）H。MOV是传送指令，STA为写数指令，LDA为读数指令。

要求：⑴分析三种指令的指令格式与寻址方式特点。

⑵CPU[illegible]花时间最长？第二种指令的[illegible]

⑶下列[illegible]其中如果有编码不正确，[illegible]

①(FoF1[illegible]

③（6FD[illegible]

[解]：

⑴第一种指令是单字长二地址指令，RR型；第二种指令是双字长二地址指令，RS型，其中S采用基址寻址或变址寻址，R由源寄存器决定；第三种也是双字长二地址指令，RS型，其中R由目标寄存器决定，S由20位地址（直接寻址）决定。

⑵处理机完成第一种指令所花时间最短，因为是RR型指令，不需要访问存储器。第二种指令所花时间最长，因为是RS型指令，需要访问存储器，同时要进行寻址方式的变换运算（基址或变址），这也需要时间。第二种指令的执行时间不会等于第三种指令，因为第三种指令虽然也访问存储器，但节省了求有效地址运算的时间开销。

⑶根据已知条件：MOV（OP）＝AH=001010，STA（OP）＝1BH=011011，LDA（OP）＝3CH=111100,将指令的十六进制格式转换成二进制代码且比较后可知：①（FoF1）H （3CD2）H 指令代表LDA指令，编码正确，其含义是把主存（13CD2）H 地址单元的内容取至15号寄存器。

②（2856)H 指令代表MOV指令，编码正确，含义是把6号源寄存器的内容传送至5号目标寄存器。

③（6FD6）H 是单字长指令，一定是MOV指令，但编码错误，可改正为（28D6）H

④（1C2）H 是单字长指令，代表MOV指令，但编码错误，可改正为（28C2）H 。

8088/8086典型指令

- 数据传送指令
  - 取数　　LDA　AX，TEMP
  - 存数　　STA　TEMP，AX
  - 传送　　MOV　AX，CX
- 算术运算指令
  - 定点＋，－，×，÷
  - ADD，INC，SUB，DEC MUL，DIV等
  - 浮点＋，－，×，÷ ，求反，求补 NEG，比较
- 逻辑运算指令
  - NOT，AND，OR，XOR

8088/8086典型指令

- 程序控制指令
  - 无条件转移 JMP 条件转移 C，Z，N，P，V
  - 转子程序 JSR 子程序返回 RET 中断返回 IRET
- 输入/输出指令
  - IN AX，n OUT n, AX
- 字符串处理指令: 字符串传送、转换、比较、查找
- 特权指令: 系统资源的分配和管理
- 其他指令
  - CLC（clear carry flag)/CLI (clear interrupt elable flag)/HLT/WAIT/ESC/LOCK/TEST/RESSET

指令系统发展方向

- CISC---复杂指令系统计算机
  - Complex Instruction System Computer
  - 指令数量多，指令功能，复杂的计算机。
- RISC---精简指令系统计算机
  - Reduced Instruction System Computer
  - 指令数量少，指令功能单一的计算机。

高级语言中各种语句的动态出现频度

	Pascal	C
赋值语句	45	38
循环语句	5	3
程序调用语句	15	12
判断语句	29	43
直接转移语句	–	3
其它	6	1

精减指令系统(RISC)

- 选取使用频率最高的一些简单指令,指令条数少
- 寻址方式简单
- 指令长度固定,指令格式简单
- CPU设置大量寄存器
- 只有存/取数指令才能访问存储器,
- 其余指令的操作都在寄存器之间进行
- 每一个机器周期完成一条机器指令

CISC与RISC的比较

特　　征	CISC			RISC	
	IBM 370/168	VAX 11/780	Intel 80486	SPARC	MIPS R4000
开发年份	1973	1978	1989	1987	1991
指令数量/条	208	303	235	69	94
指令长度/B	2～6	2～5	1～11	4	4
寻址方式	4	22	11	2	2
通用寄存器数/个	16	16	8	40～520	32
控制存储器大小/Kb	420	480	246	—	—
Cache大小/KB	64	64	8	32	128

典型RISC机指令系统

机器型号	指令数	寻址方式	指令格式	通用寄存器数
RISC-I	31	2	2	78
RISC-II	39	2	2	138
MIPS	55	3	4	16
SPARC	75	4	3	120-136
MIPS R3000	91	3	3	32
i860	65	3	4	32

例子：字长16位，主存64K，指令单字长单地址，80条指令。寻址方式有直接、间接、相对、变址。请设计指令格式。

解：

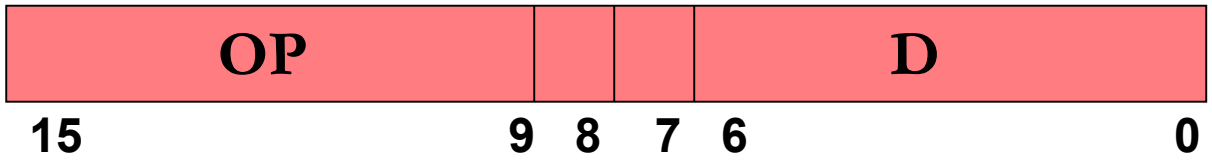
80条　　log[80]=7　　采用7位操作码

寻址方式4种　　采用2位

PC为16位　　变址寄存器16位

相对寻址　E=（PC）+D

变址寻址　E=（R）+D



例子

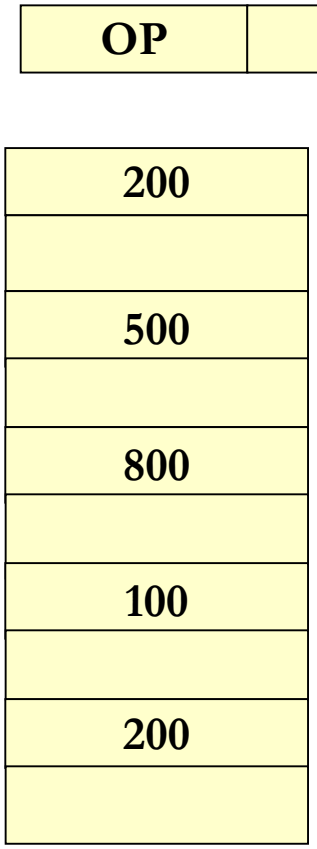
下列关于 RISC 的叙述中，错误的是（ ）

A. RISC 普遍采用微程序控制器
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、寻址方式和指令格式种类相对 CISC 少

偏移寻址通过将某个寄存器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

设某机的指令格式、有关寄存器和主存内容如下，X为寻址方式，D为形式地址，请在下表中填入有效地址E及操作数的值？



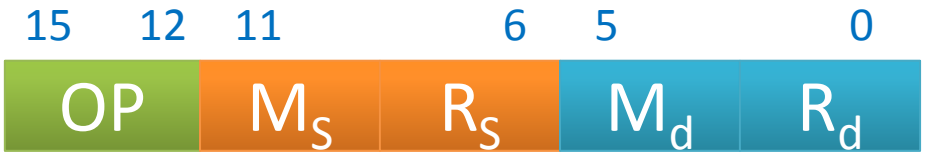
PC=1000

$R_{基}$=2000

100	200
200	500
500	800
1100	100
2100	200

寻址方式	X	有效地址E	操作数
立即	0	——	100
直接	1	E=D=100	200
间接	2	E=(D)=200	500
相对	3	E=PC+D=1100	100
变址	4	E=(R)+D=2100	200
变址间址	5	E=((R)+D)=200	500

(11分)某计算机字长为16位，主存地址空间大小为128KB，按字编址。采用单字长指令格式，指令各字段定义如下：10年研究生统考



（1）、该指令系统最多支持2^4=16条指令；支持2^3=8个通用寄存器；16位字长的计算机的MAR、MDR至少是16位字长；

000B	寄存器直接	R_n	操作数=(R_n)
001B	寄存器间接	(R_n)	操作数=((R_n))
010B	寄存器间接，自增	(R_n)+	操作数=((R_n)),(R_n)+1→(R_n)
011B	相对	D(R_n)	转移目标地址=(PC)+(R_n)

注(x)表示存储器地址x或寄存器x的内容

（1）该指令系统最多可有多少条指令？该计算机最多有多少个通用寄存器？存储器地址寄存器MAR和存储器数据寄存器MDR至少需要多少位？

(11分)某计算机字长为16位，主存地址空间大小为128KB，按字编址。采用单字长指令格式，指令各字段定义如下：

（2）、转移指令的目标地址范围应该是整个内存的寻址空间0~65535；

000B	寄存器直接	R_n	操作数=(R_n)
001B	寄存器间接	(R_n)	操作数=((R_n))
010B	寄存器间接，自增	(R_n)+	操作数=((R_n)),(R_n)+1→(R_n)
011B	相对	D(R_n)	转移目标地址=(PC)+(R_n)

注(x)表示存储器地址x或寄存器x的内容

（2）转移指令的目标地址范围是多少？

（3、指令的机器码：0010 001100 010101B，即2315H；指令功能是：把内存1234H单元中的数据与内存5678H单元中的数据相加，结果写回到5678H单元;R5的内容用作内存地址之后，还要执行R5的内容加1的操作。因此，指令执行后，R5的内容将变为5679H，内存5678H单元的内容将变为该加法指令计算得到的和：5678H+1234H=68ACH。

000B	寄存器直接	R_n	操作数=(R_n)
001B	寄存器间接	(R_n)	操作数=((R_n))
010B	寄存器间接，自增	(R_n)+	操作数=((R_n)),(R_n)+1→(R_n)
011B	相对	D(R_n)	转移目标地址=(PC)+(R_n)

（3）若操作码0010B表示加法操作，助记符为add，寄存器R4，R5的编号分别为100B和101B，R4的内容为1234H，R5的内容为5678H，地址1234H中的内容为5678H，地址5678H中的内容为1234H，则汇编语句**add (R4),(R5)+**逗号前为源操作数，逗号后为目的操作数，对应的机器码是多少？用十六进制表示。该指令执行以后，哪些寄存器和存储单元的内容会发生改变？改变后的内容是什么？

作业

- 138页：**5、6、7、8**