

程序的机器级表示

指令系统简介

主讲人: 邓倩妮

上海交通大学



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

本节内容



- 指令系统
- 指令格式
- 寻址方式
- RISC 与 CISC

指令系统基本概念



- 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

计算机指令系统特性



- 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(所有寄存器、存储单元同等对待)、匀齐性（一种操作支持多种数据类型）：
 - 指令格式和数据格式的一致性（指令长度和数据长度通常是字节的整数倍）：
- 兼容性：系列机软件向上兼容

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。
- IBM, PDP-11, VAX-11, 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

地址码字段

操作码(OP)与地址码(AC)



操作码字段

地址码字段

- 指令系统中每一条指令对应一个操作码
- 操作码的长度取决于指令系统的规模
 - $L_{OP} = \lceil \log_2 n \rceil$
- 定长指令、变长指令
- 地址码包括被操作数, 操作数, 操作结果

指令分类方法



- 按操作数个数分类
- 按操作数物理位置分类
- 按指令长度分类

按操作数个数分类



三地址指令



$(A1)OP(A2) \rightarrow (A3)$

二地址指令



$(A1)OP(A2) \rightarrow (A1)$

一地址指令



$(AC)OP(A) \rightarrow AC$

零地址指令



如停机, 空操作, 堆栈指令

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

指令字长度



- 指令中包含二进制代码的位数
- 与机器字的长度有关：单字长,双字长,半字长。
- **等长指令**：结构简单，控制线路简单。
- **变长指令**：结构灵活，充分利用指令长度，控制复杂



指令字助记符



- ADD
- SUB
- MOV
- JMP
- STR
- LDA



寻址方式



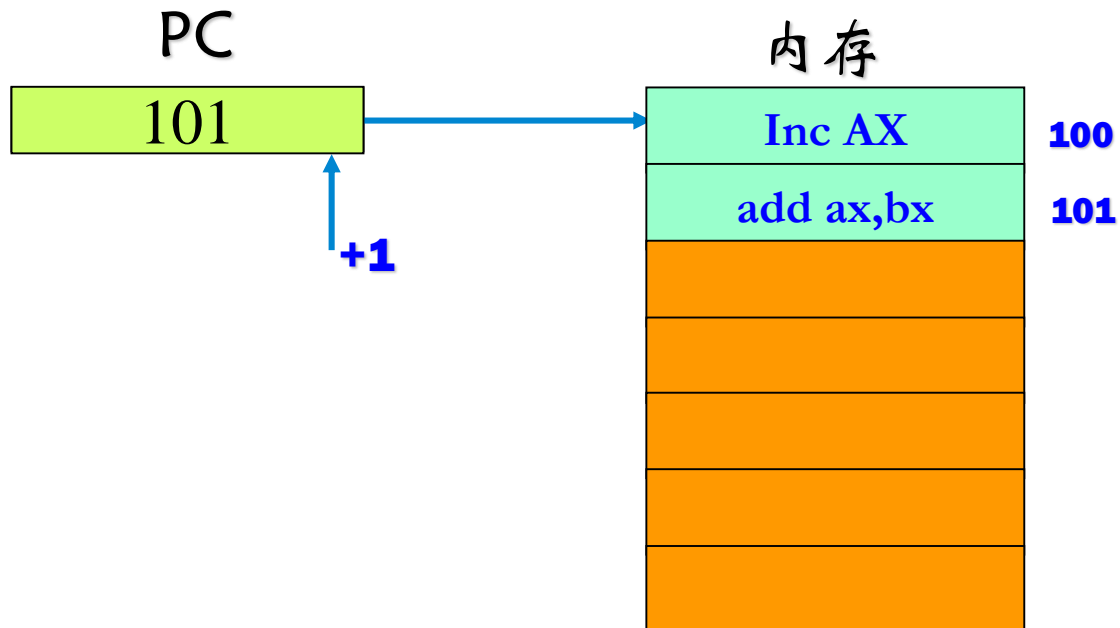
- 寻找指令或操作数有效地址的方式
 - 指令寻址
 - 顺序寻址
 - 跳跃寻址
 - 操作数寻址

顺序寻址



- 程序的指令序列在主存顺序存放。程序执行时从第一条指令开始，逐条取出并逐条执行，这种程序的顺序执行过程，称为顺序寻址方式。
- 为了达到顺序寻址的目的，CPU中必须有一个程序计数器（PC）对指令的顺序号进行计数。PC中开始时存放程序的首地址，每执行一条指令，PC 加1，以指出下条指令的地址，直到程序结束。
- PC存放下一条指令的地址

顺序寻址过程

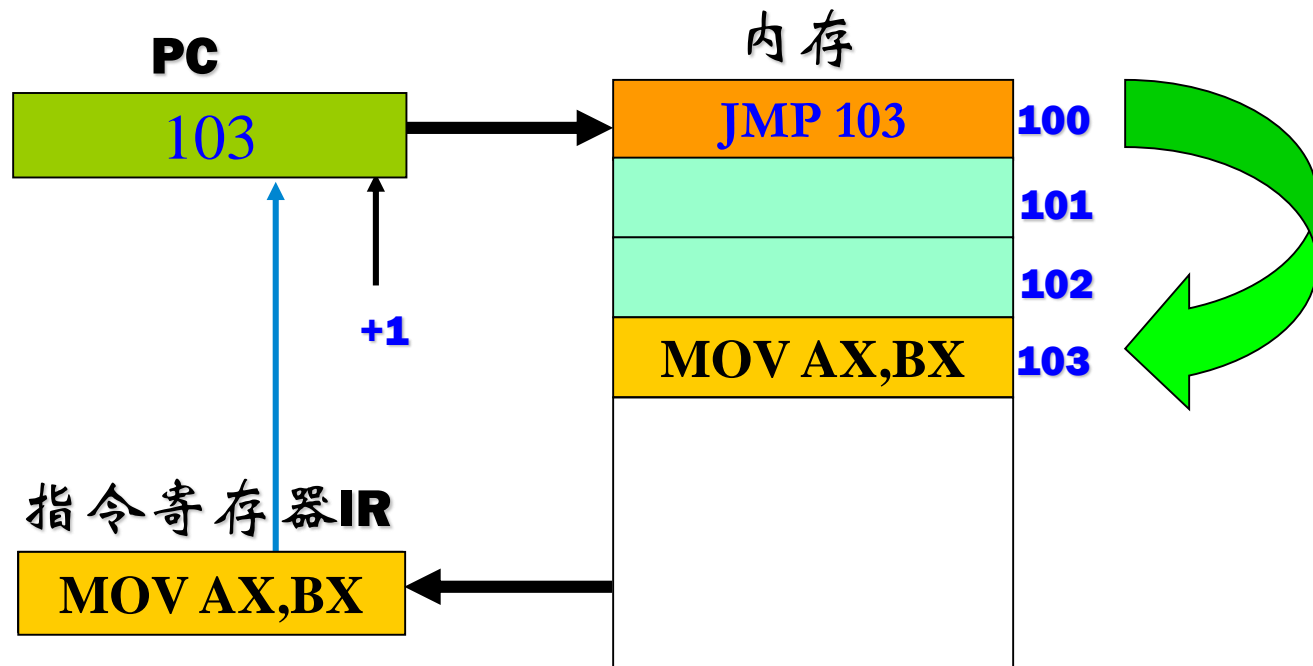


跳跃寻址



- 当程序中出现分支或循环时，就会改变程序的执行顺序。此时对指令寻址就要采取跳跃寻址方式。
- 所谓跳跃，就是指下条指令的地址不是通过程序计数器PC当前值获得的，而是由指令本身给出。
- 跳跃的处理方式是重新修改PC的内容。然后进入取指令阶段。

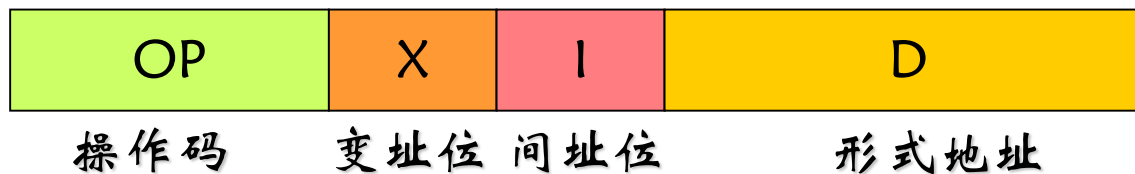
跳跃寻址过程



操作数的寻址方式



- 形成操作数有效地址的方法。
 - 单地址指令地址码的构成: X, I, D
 - 实际有效地址为 E , 实际操作数 S
 - $S = (E)$





寻址方式分类

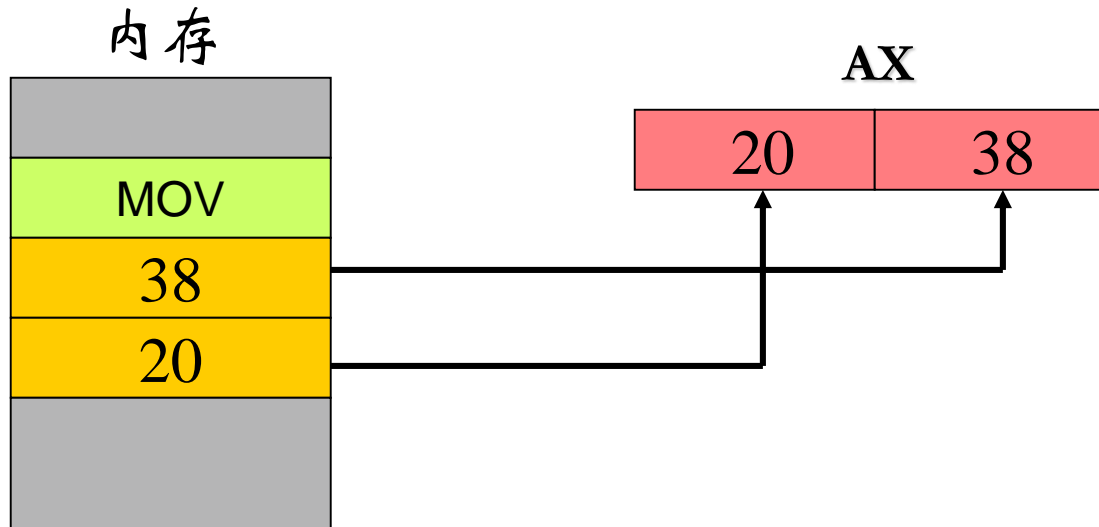


- 立即寻址
- 直接寻址
- 间接寻址（已经不使用）
- 寄存器寻址
- 寄存器间接寻址
- 相对寻址
- 变址寻址
- 复合寻址

立即寻址



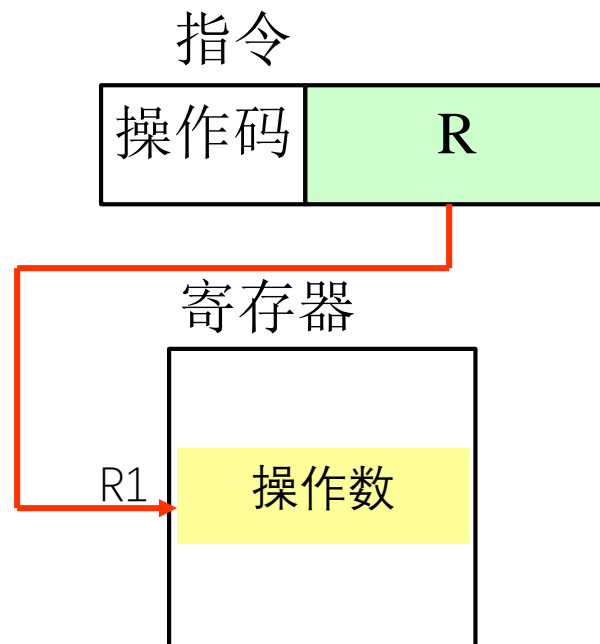
- 地址码字段是操作数本身
 - 例: `MOV AX,2038H` (2038H→AX)



寄存器寻址(Register Addressing)



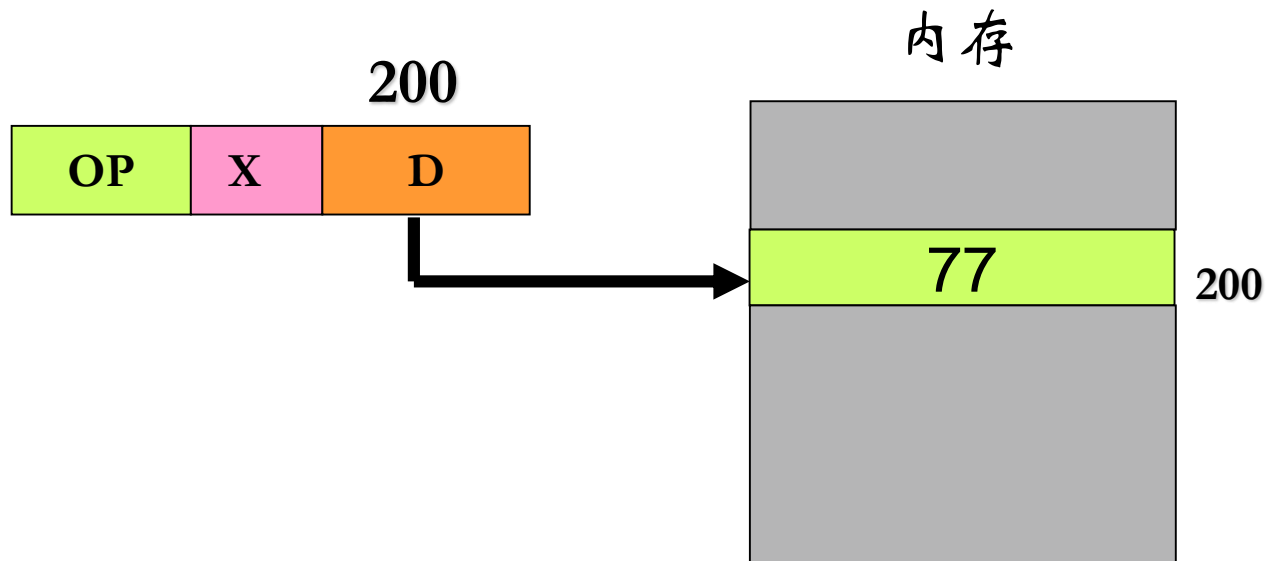
- 操作数在CPU的内部寄存器中。
 - 例如 INC R1



直接寻址(Direct Addressing)



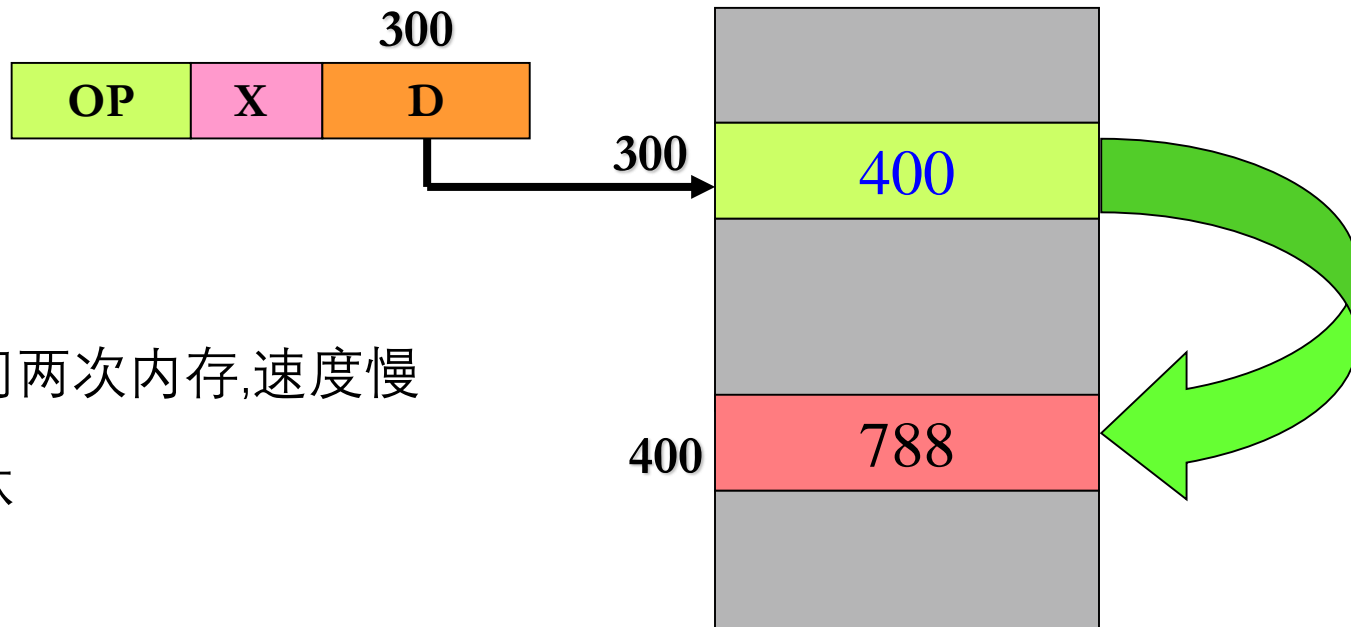
- 地址码字段直接给出操作数在内存的地址
- MOV AX , [200]



间接寻址(Indirect Addressing)



- D单元的内容是操作数地址, D是操作数地址的地址
 - $E=(D)$

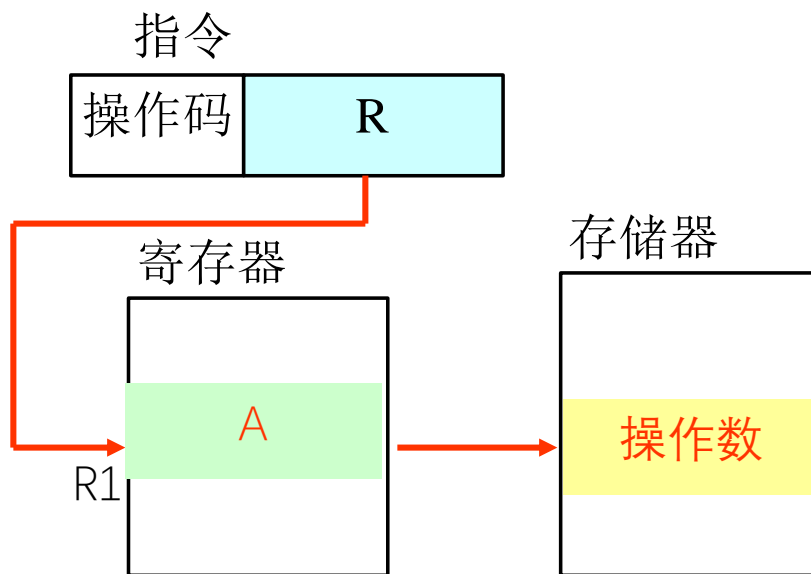


需要访问两次内存,速度慢
已被淘汰

寄存器间接寻址 (Register Indirect Addressing)

- 寄存器中的内容是操作数的内存地址,

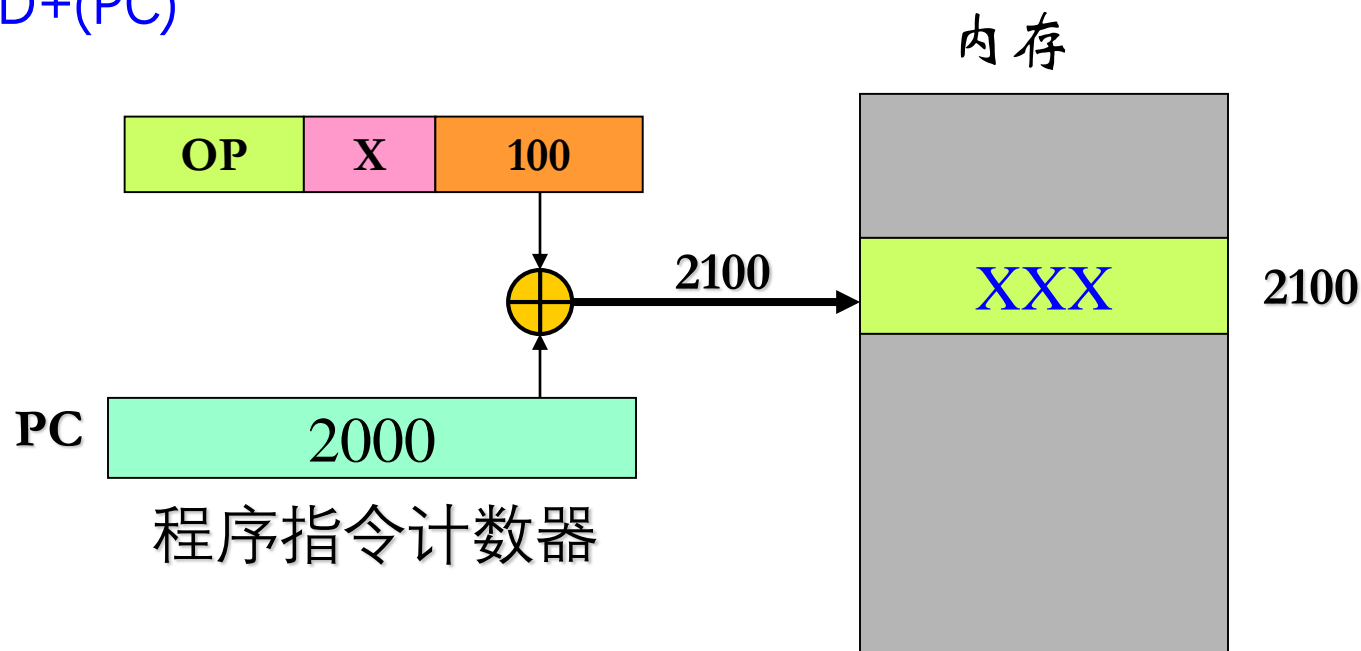
例如：INC (R1)



相对寻址 (Relative Addressing)



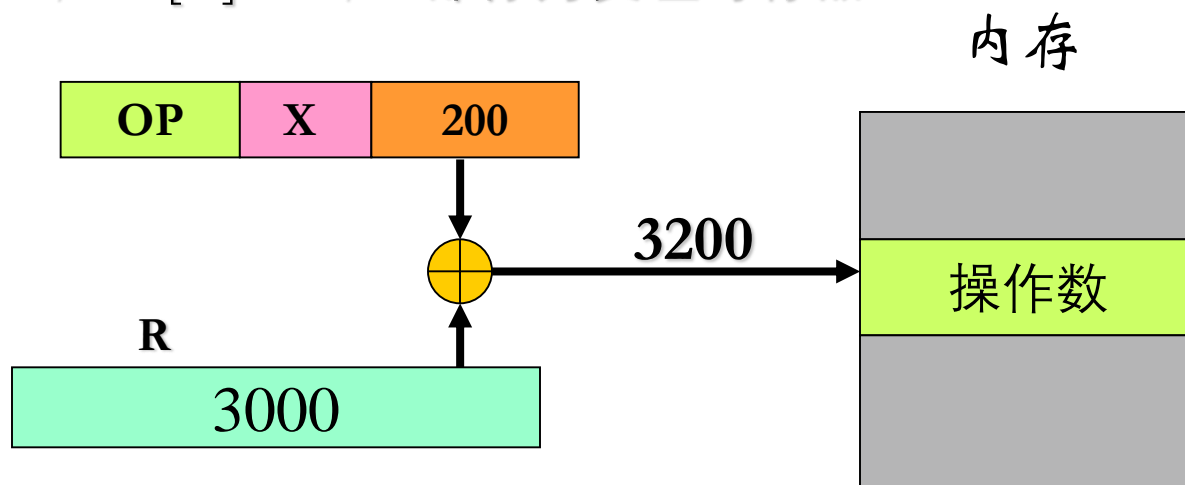
- 指令中的D加上PC的内容作为操作数的地址.
- $E = D + (PC)$



变址寻址(Index Addressing)



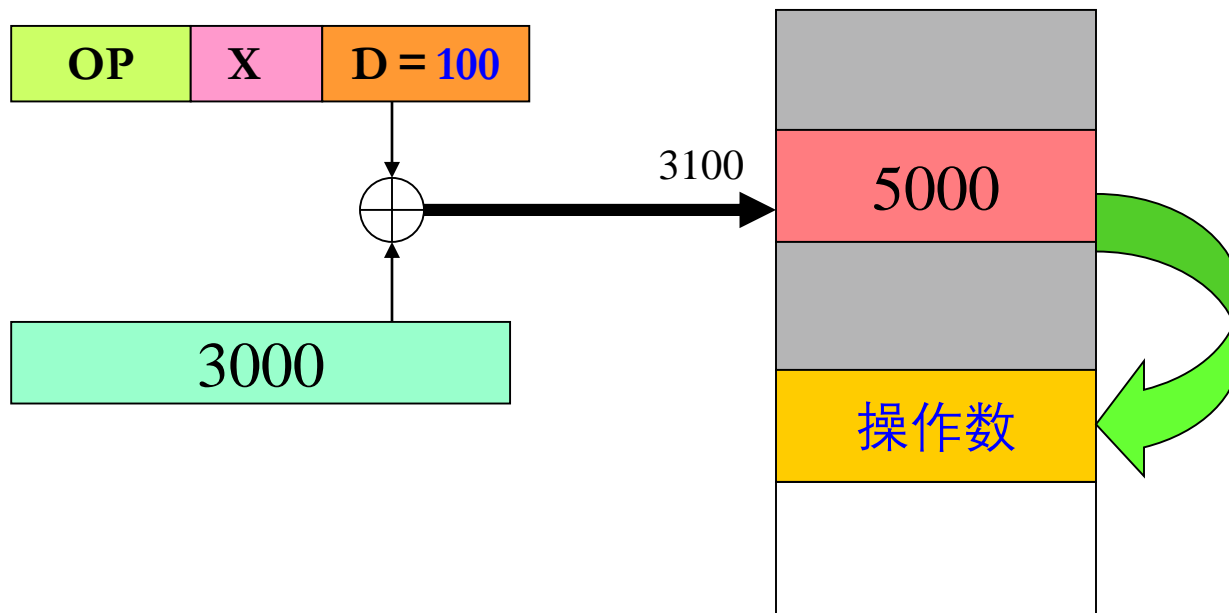
- 指定一个寄存器R,其存放基址,R被称为变址寄存器。它与本指令的地址无关, R的内容可以随要求填入。
- $E = D + (R)$
- MOV AX, 200[SI] SI, DI 都称为变址寄存器



复合寻址(Composite Addressing)



- 将间址,相对,变址,基值等寻址方式组合.
- 例如：变址间址 先变址,后间址. $E=((R)+D)$



各种常见寻址方式的汇编指令表示



寻址方式	汇编表示	操作内容
直接寻址	load adr	$ac \leftarrow \text{mem}[\text{adr}]$
间接寻址	load (adr)	$ac \leftarrow \text{mem}[\text{mem}[\text{adr}]]$
相对寻址	load adr(pc)	$ac \leftarrow \text{mem}[\text{pc} + \text{adr}]$
立即寻址	load #n	$ac \leftarrow n$
变址寻址	load adr(rn)	$ac \leftarrow \text{mem}[\text{adr} + \text{rn}]$
寄存器寻址	load rn	$ac \leftarrow \text{rn}$
寄存器间接寻址	load (rn)	$ac \leftarrow \text{mem}[\text{rn}]$

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	VAX 11	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

谢谢！

