清華大學
Tsinghua University

第二单元　第一讲

控制器概述
指令和指令系统

刘卫东

计算机科学与技术系

主要内容和教学安排

清華大學 Tsinghua University

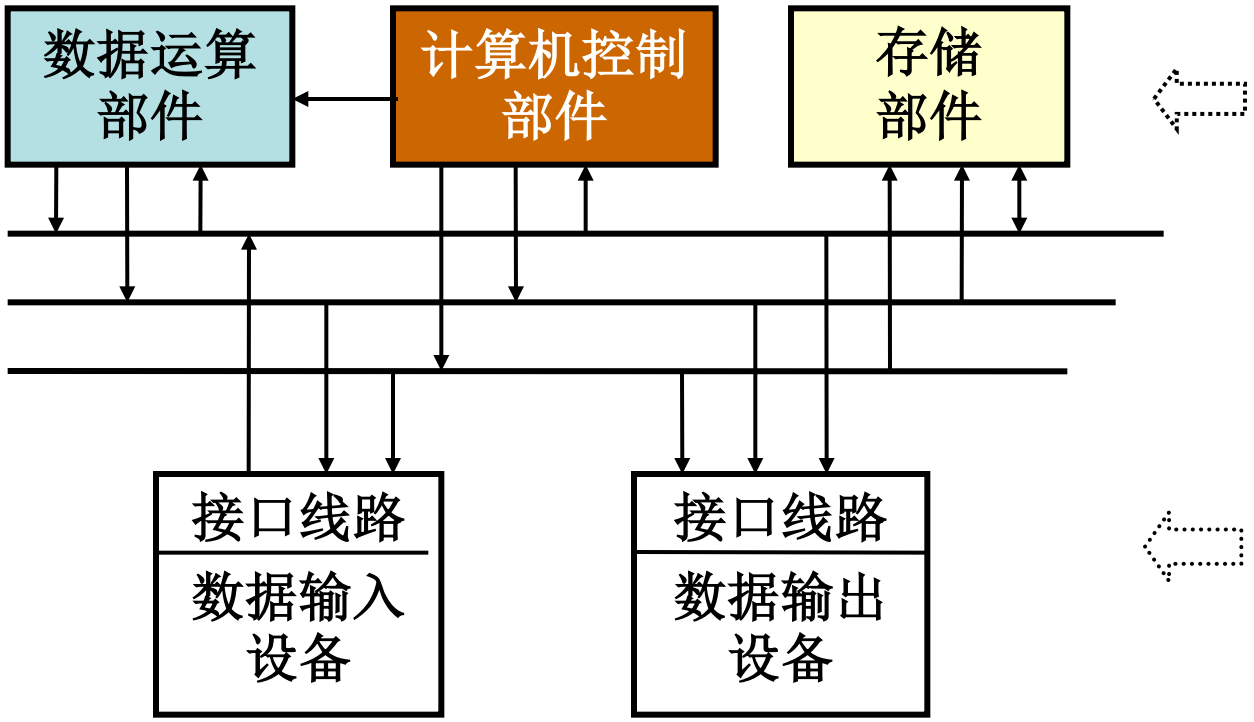
- 第一讲　控制器概述 指令系统 指令格式 寻址方式
- 第二讲　指令功能及实现
- 第三讲　单周期CPU设计
- 第四讲　多周期CPU设计
- 第五讲　指令流水基本概念
- 第六讲　流水中的结构冲突、数据冲突
- 第七讲　控制冲突、中断的解决方案
- 第八讲　THINPAD介绍 大实验
- 第九讲　大实验辅导及检查(1)
- 第十讲　大实验辅导及检查(2)
- 第十一讲 大实验辅导及检查(3)

重点和难点

- 单条指令功能的实现
  - 如何设计指令的数据通路?
  - 如何划分指令的执行步骤?
  - 如何根据指令得到控制信号?
- 机器语言程序的自动执行
  - 指令之间如何衔接?
- 提高性能
  - 在不增加太多硬件的情况下如何提高性能?
- 实现途径
  - 控制信号生成：组合逻辑或微程序
  - 程序自动执行： PC 、节拍和下地址
  - 指令系统：RISC和CISC
  - 提高性能：指令流水

计算机硬件系统功能部件

计算机的 CPU



计算机主机

数据总线
地址总线
控制总线

计算机外围设备

控制器的作用

- 计算机的基本功能
  - 执行程序
- 程序的构成
  - 指令序列
- 控制器的作用
  - 根据指令的要求，提供给各部件相应的控制信号，指挥、协调各部件共同完成指令规定的功能
  - 自动执行下一条指令

指令与指令系统

- 指令与指令系统的概念
- 指令系统设计要求
- 指令功能和分类
- 指令格式
  - 变长指令字/定长指令字
  - 操作码扩展
- 寻址方式

指令与指令系统

计算机系统由硬件和软件两大部分组成。硬件指由中央处理器、存储器以及外围设备等组成的实际装置。软件是为了使用计算机而编写的各种系统的和用户的程序，程序由一个序列的计算机指令组成。

指令是计算机运行的最小的功能单元，是指挥计算机硬件运行的命令，是由多个二进制位组成的位串，是计算机硬件可以直接识别和执行的一个信息体。

一台计算机提供的全部指令构成该计算机的指令系统。指令用于程序设计人员告知计算机执行一个最基本运算、处理功能，多条指令可以组成一个程序，完成一项预期的任务。

指令系统的地位

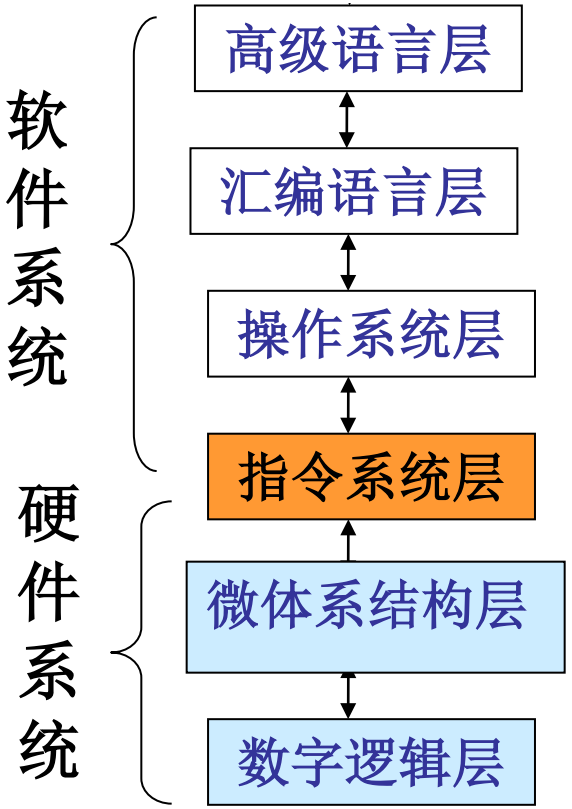
可以从6个层次分析和看待计算机系统的基本组成。

指令系统层处在硬件系统和软件系统之间，是硬、软件之间的接口部分，对两部分都有重要影响。

硬件系统用于实现每条指令的功能，解决指令之间的衔接关系；

软件由按一定规则组织起来的许多条指令组成，完成一定的数据运算或者事务处理功能。

指令系统优劣是一个计算机系统是否成功的关键因素。



计算机系统的层次结构

指令系统设计要求

- 完备性
  - 指令功能齐全、编程方便
- 规整性
  - 指令格式简单、统一
- 高效性
  - 占内存少，运行高效
- 兼容性
  - 同一系列软件兼容

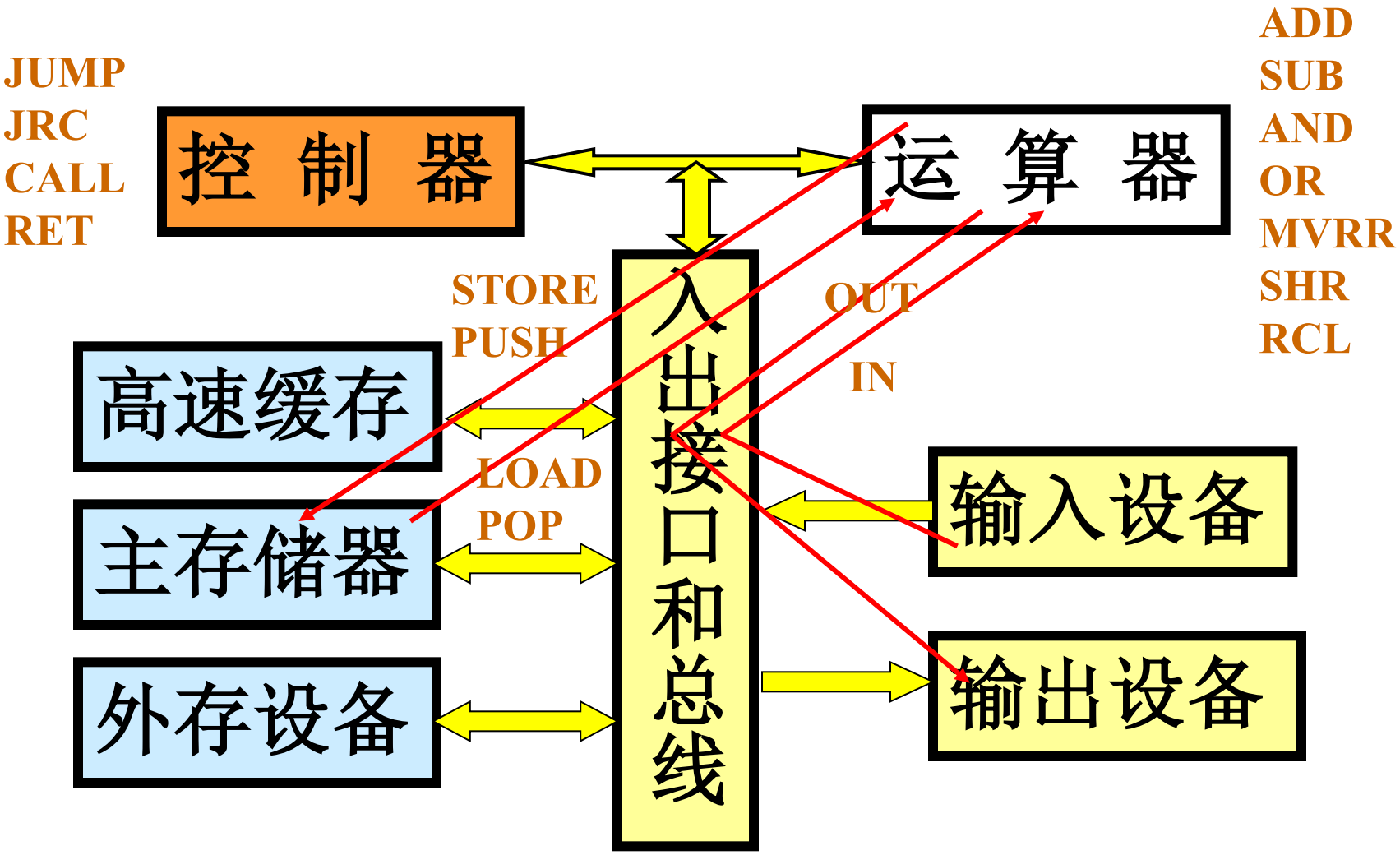
计算机中需配备的指令

清華大學 Tsinghua University

指令是用户使用计算机和计算机本身运行的最小的功能单元：① 指令是由多个二进制位组成的数串，② 用于设计程序，③ 计算机硬件可直接识别和执行。通常情况下一台计算机需要提供哪些指令呢?

计算机用于计算和处理数据，为此，要在计算机硬件系统中设置 5 种类型的部件：运算器部件、控制器部件、存储器部件、输入设备、输出设备，各自承担数据运算、系统指挥控制、保存当前程序和数据、执行输入和执行输出的功能。需要在计算机中设置为使用和控制这几个部件运行的相应指令。

使用硬件系统的基本指令



指令的功能和分类

清華大學
Tsinghua University

指令用于设计程序，指令系统构成最低级别的程序设计语言，程序设计人员通过指令直接指挥计算机的硬件完成某一个基本的运算、处理功能，例如：

对数值数据的算术运算，对逻辑数据的逻辑运算，
在计算机部件之间传送、保存数据，
从外部向计算机内输入数据，
把计算机内部计算结果输出出来，
按照某种条件控制计算机选择执行某段程序，
当然还有另外一些方面的更深层次的要求 等；

可以按照指令执行的功能对它们进行分类。

指令的功能和分类

清華大學 Tsinghua University

- **算术与逻辑运算指令**

  加、减、乘、除、变符号 等算术运算

  与、或、非、异或 等逻辑运算

- **移位操作指令**

  算术移位（一般只右移）、逻辑移位、循环移位

- **数据传送指令**

  通用寄存器之间传送

  通用寄存器与主存储器存储单元之间传送

  主存储器不同存储单元之间传送

- **输入输出指令**

  通用寄存器与输入输出设备（接口）之间传送

指令的功能和分类

清華大學 Tsinghua University

- **转移指令**

  变动程序中指令执行次序的指令，分为无条件转移指令和条件转移指令

- **子程序调用与返回指令**

  调用指令与返回指令二者要配合使用，子程序的最后一条指令一定是返回指令，执行结束后返回主程序断点

- **堆栈操作指令**

  堆栈（stack）是由若干个连续存储单元组成的先进后出的存储区，有压入（即进栈）和弹出（即退栈）操作

- **其他指令**

  置条件码指令、开中断指令、关中断指令

  停机指令、空操作指令、特权指令

指令表示

指令中的内容，包括指令操作码（指出指令完成的运算处理功能和数据类型）和 操作数或指令的地址（指明用到的数据或地址）两部分。例如：

算逻运算中的运算功能，数据来源或结果去向

数据传送指令中的数据原来位置和新的存储位置

输入输出指令中用到的设备和数据来、去的位置

转移指令的转移类别、转移条件和转移地址 等

每一条指令必须指明它需要完成的功能，通常用几位指令操作码表示；还需要指明用到的数据、地址或设备，通常在地址字段给出，可能是：

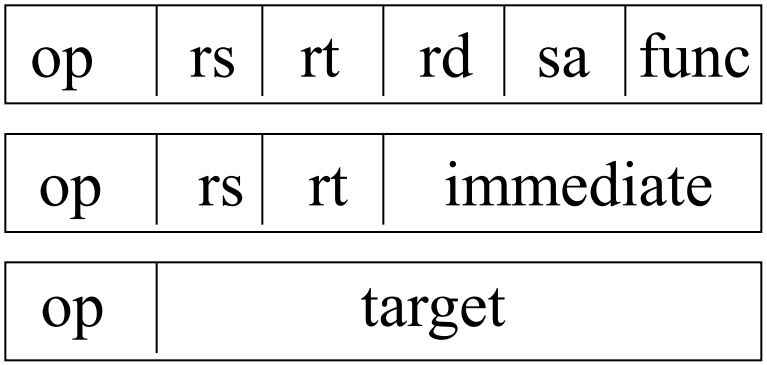
(1) 寄存器编号，　　(2) 设备端口地址，

(3) 存储器的单元地址　　(4) 数值　等几种信息。

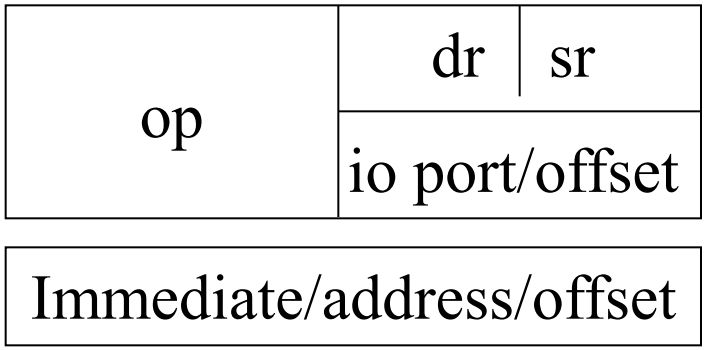
指令格式与指令字长

清華大學 Tsinghua University

指令字长是指组成一条指令的二进制数的位数，例如 8 bits、16 bits、32 bits、64 bits等，指令格式与指令字长密切相关，指令字越长可以给出的信息越多 。一个指令字通常由指令操作码和操作数地址两部分组成，如何把一个指令字划分成多个字段并分配各字段所表示的内容大有学问。



MIPS 指令格式 (32位)



TEC-2000 指令格式 (16位)

指令格式

- **指令字**：完整的一条指令的二进制表示
- **指令字长**：指令字中二进制代码的位数

  机器字长：计算机能直接处理的二进制数据的位数

  指令字长（字节倍数）＝0.5、1、2…个机器字长

  定长指令字结构 vs. 变长指令字结构
- **指令格式**：指令字中**操作码**和**操作数地址**的二进制位的分配方案

操作码：指明本条指令的操作功能，
每条指令有一个确定的操作码

操作数地址：说明操作数存放的地址，有时是操作数本身

操作码组织与编码

清華大學 Tsinghua University

定长的操作码的组织方案

在指令字最高位部分分配固定若干位用于表示操作码，有利于简化计算机硬件设计，提高指令译码和识别速度

例如：IBM360机、THINPAD教学机

变长的操作码的组织方案

在指令字最高位部分用一固定长度的字段来表示基本操作码，而对于部分操作数地址位数可以少的指令，则把另外多位辅助操作码扩充到该操作数地址字段，即操作码位数可变。

这种方法在不增加指令字长的情况下，可表示更多的指令，但增加了译码和分析难度，要求更多的硬件支持

例如：PDP-11计算机、TEC-2000的 8位机

操作码组织与编码

清華大學 Tsinghua University

- 操作码字段与操作数地址字段有所交叉的方案

不同指令的操作码长度可以不同，表示操作码所用到的一些二进制位不再集中在指令字的最高位部分，而是与用于表示操作数地址的一些字段有所交叉，操作码还被区分为主操作码和辅助操作码这样不同的两部分，这是一种比较特殊、不很常用的方案。

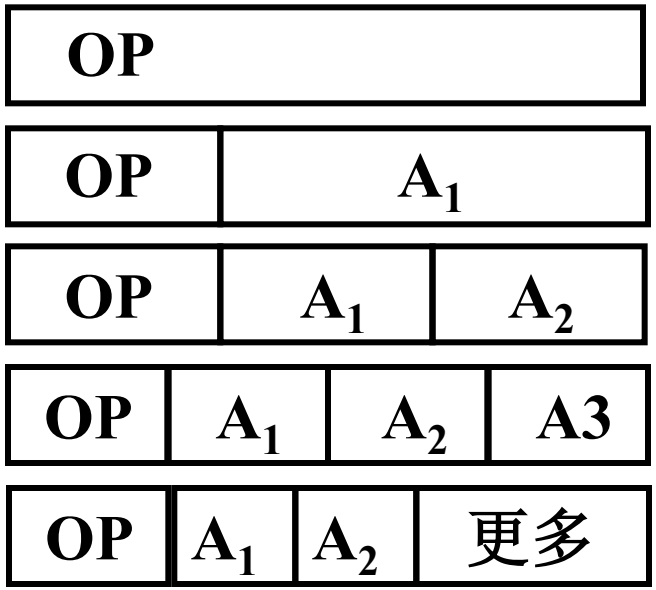
例如：NOVA (DJS-130) 计算机就采用这种方案

指令操作码的位数限制指令系统中的指令条数！

操作数个数与来源

清華大學 Tsinghua University

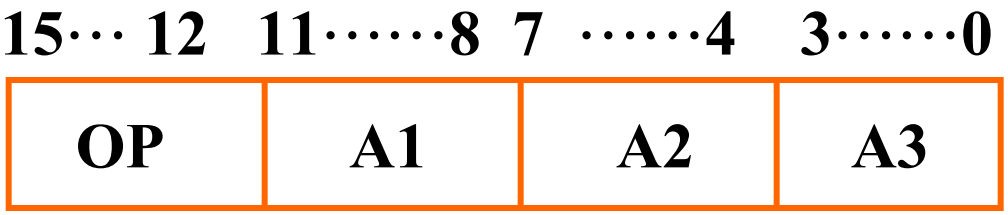
- 指令操作数个数
  - 无操作数指令（零地址指令）
  - 单操作数指令（一地址指令）
  - 双操作数指令（二地址指令）
  - 三操作数指令（三地址指令）
  - 多操作数指令（多地址指令）



- 指令操作数来源和去向
  - CPU内部的通用寄存器
  - 输入输出设备（接口）的一个寄存器
  - 主存储器的一个存储单元

指令操作码的扩展技术

假设某机器的指令长度为16位，包括 4位基本操作码和三个 4位地址码段。



4 位基本操作码可表示 16个状态，

如用 4 位操作码，则能表示 16 条三地址指令，

若用 8 位操作码，则可表示 256 条二地址指令，

而用12位操作码，则可表示 4096条一地址指令，

若16位全用作操作码，则可表示 65536条零地址指令

指令操作码的扩展技术

清華大學 Tsinghua University

若需要在16位字长的指令中能够同时支持三地址、二地址、一地址指令各15条，零地址指令16条，则可以选用如下方案的变长操作码实现:

15条三地址指令的操作码为：0000 ~ 1110

15条二地址指令的操作码的高 4 位选用 1111，低 4 位用 0000 ~ 1110，即得到: 11110000 ~ 11111110

15条一地址指令的操作码的高 8位选用11111111，低 4 位用 0000 ~ 1110，即: 111111110000 ~111111111110

16条零地址指令的操作码的高 12位每位均用1，低 4 位随意，即: 111111111110000~111111111111111

指令操作码扩展技术

前面介绍的操作码扩展方案中，每次扩展4位并仅保留了一个编码用于接下来的扩展过程，当每次扩展的位数和保留的位数变化时，后面可扩展的指令条数就可以变化。例如在16位字中的指令字中，可以选用如下方案支持三地址指令、二地址指令、一地址指令和零地址指令14、30、31、16条：

14条三地址为：0000 ~ 1101 (保留1110、1111 两个码)

30条二地址为：11100000 ~11111101 （保留 2个码）

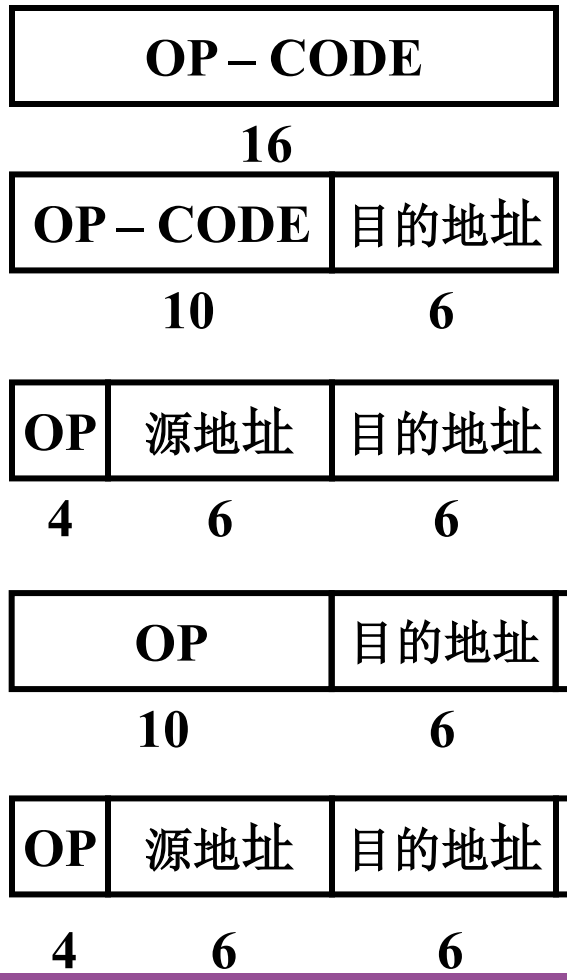
31条一地址为：1111111100000 ~1111111111110 (保留 1个码)

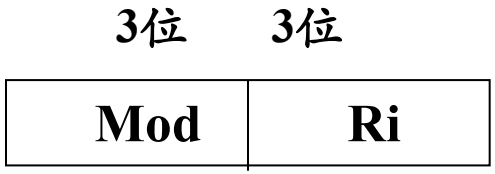
16条零地址为：1111111111110000~1111111111111111

指令操作码扩展技术

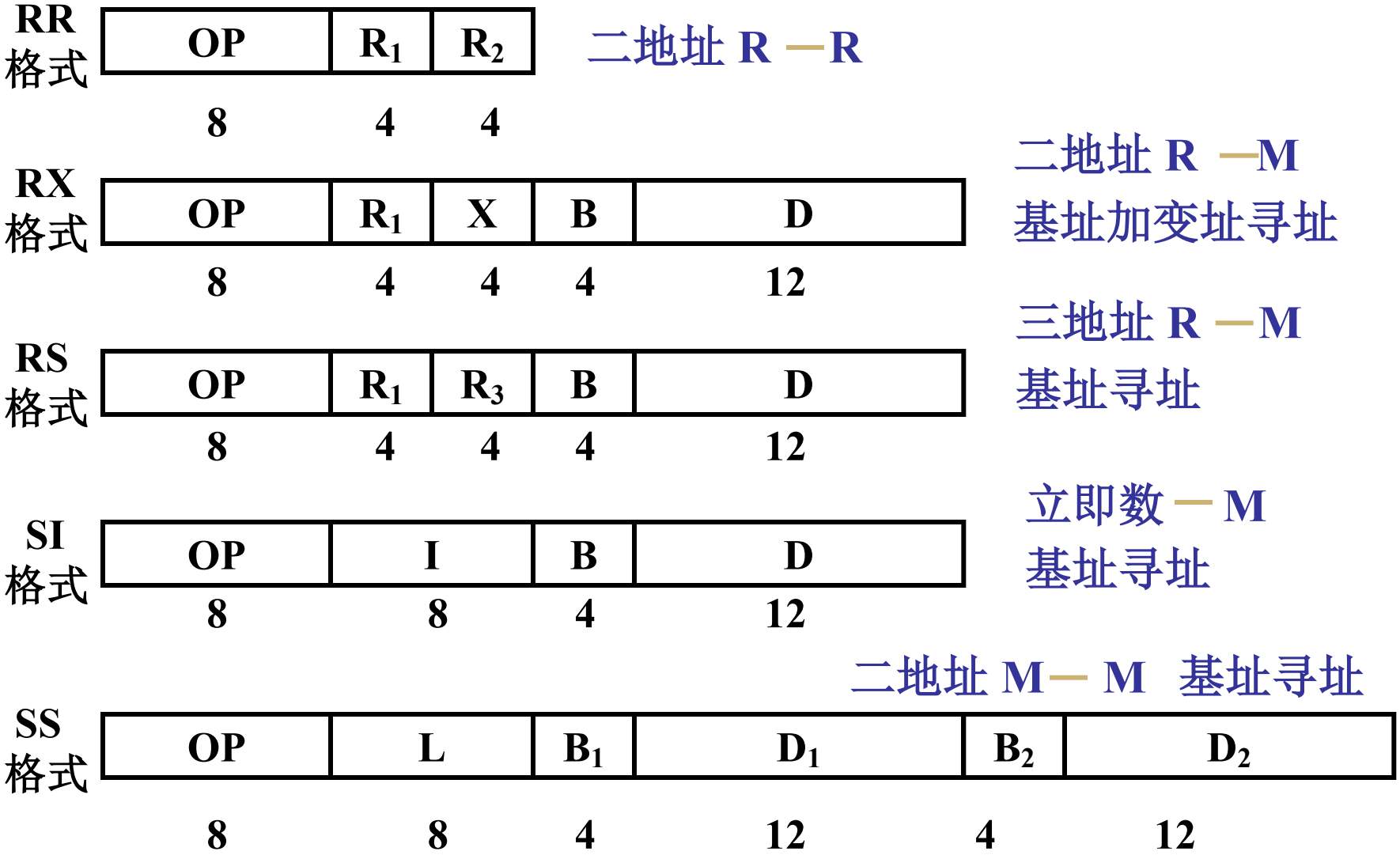
(PDP－11 指令为例)

指令字长有 16 位、32 位、48 位三种 (1字、2字、3字)





IBM 360指令格式

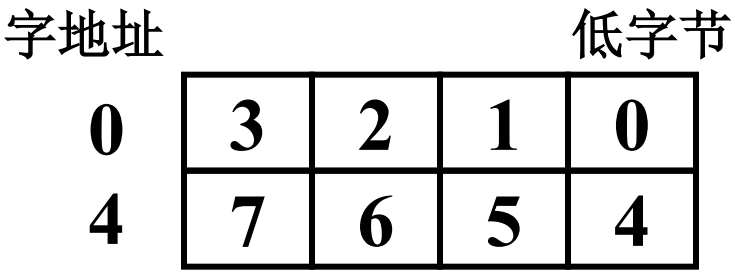

RR
格式
OP
R1
R2
8
4
4
二地址 R —R
RX
格式
OP
R1
X
B
D
8
4
4
4
12
二地址 R —M
基址加变址寻址
RS
格式
OP
R1
R3
B
D
8
4
4
4
12
三地址 R —M
基址寻址
SI
格式
OP
I
B
D
8
8
4
12
立即数 — M
基址寻址
二地址 M— M 基址寻址
SS
格式
OP
L
B1
D1
B2
D2
8
8
4
12
4
12

操作数类型与数据存储方式

1、 操作数类型

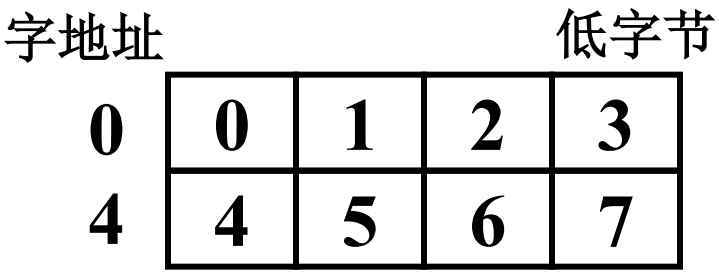
地址	无符号整数
数字	定点数、浮点数、十进制数
字符	ASCII
逻辑数	逻辑运算，bit

2、 数据在存储器中的存放方式



字地址 为 低字节 地址

Intel



字地址 为 高字节 地址

Motorola

寻址方式

寻址方式（又称编址方式）指的是确定本条指令的操作数地址及下一条要执行的指令地址的方法。

不同的计算机系统，使用数目和功能不同的寻址方式，其实现的复杂程度和运行性能各不相同。有的计算机寻址方式较少，而有些计算机采用多种寻址方式。

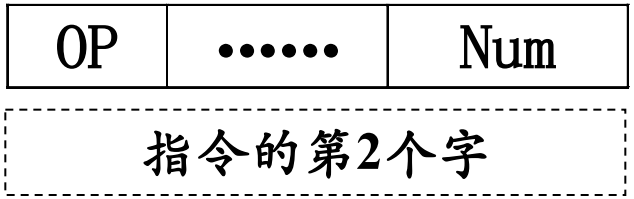
通常需要在指令中为每一个操作数专设一个地址字段，用来表示数据的来源或去向的地址。在指令中给出的操作数（或指令）的地址被称为形式地址，使用形式地址信息并按一定规则计算出来或读操作得到的一个数值才是数据（或指令）的实际地址。在指令的操作数地址字段，可能要指出：

① 运算器中的累加器的编号或专用寄存器名称（编号）
② 输入/输出指令中用到的 I/O 设备的入出端口地址
③ 内存储器的一个存储单元（或一 I/O设备 ）的地址

有多种 基本寻址方式 和某些 复合寻址方式 ，简介如下。

立即数寻址

所需的一个操作数在指令的地址字段部分直接给出。



Num 即为操作数的值

适用于操作数固定的情况，取指同时取得操作数，指令执行阶段不必到存储器中取操作数，提高了指令执行速度；

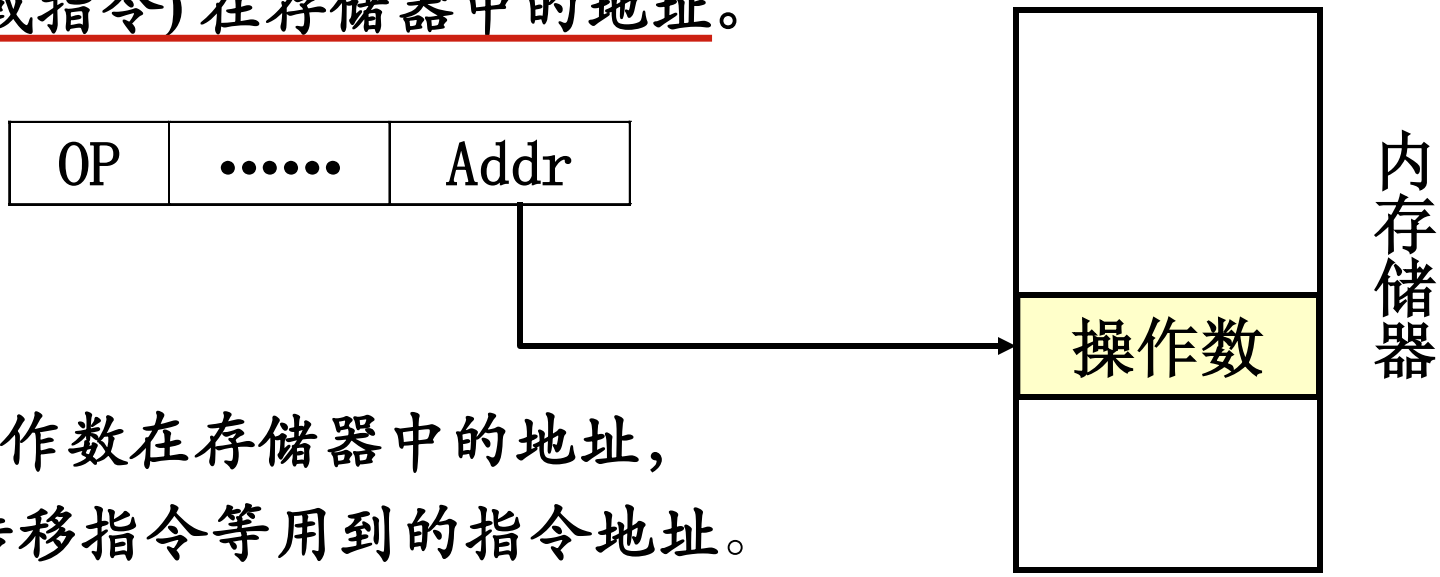
当该立即数的值较小(占用位数少)时，可在指令字第一个字中直接给出，否则需要用指令的第二个字提供。

例：Num = 1234H，指令的一个操作数就是 1234H

这里的 H 表示 1234 是 16 进制的值

(2) 直接寻址

在指令的地址码字段，直接给出所需的操作数(或指令)在存储器中的地址。



Addr 为操作数在存储器中的地址，

或转移指令等用到的指令地址。

例：Addr = 5718H，可能是下一条指令的地址或一个操作数的地址，若 [5718H] = 3，则用 5718H 作地址，从内存储器单元中读出的操作数就是 3。

寄存器寻址/寄存器间址

计算机的CPU中设置有一定数量的通用寄存器，用于存放操作数、操作数地址或中间结果。假如指令地址码字段给出某一通用寄存器的编号(地址)，且所需的操作数就在这一寄存器中，这就是寄存器寻址方式；若该寄存器中存放的是操作数在内存储器中所在单元的地址，这就是寄存器间接寻址方式。可通过指令的操作码或另设一个字段，来区分这两种不同的寻址方式。

OP	……	RegNo.

例：RegNo.=5，使用 5# 累加器，

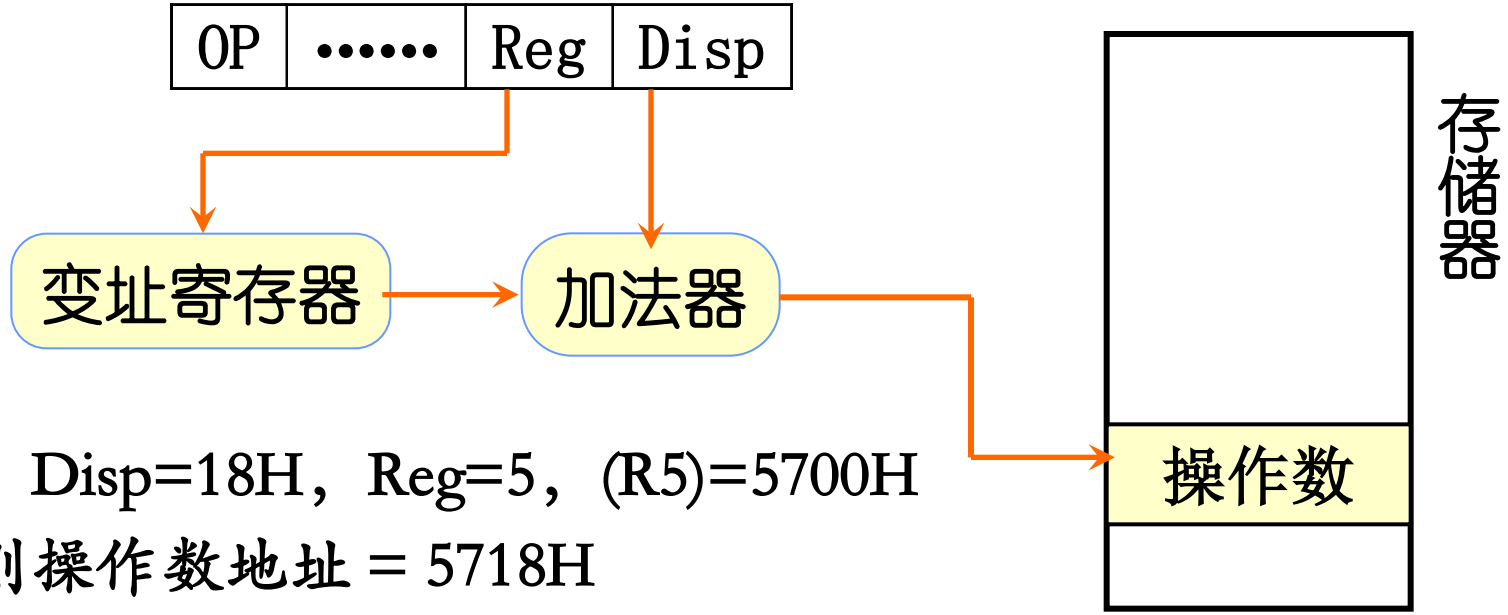
此时若 5# 累加器中的内容为 7，可记为 (R5)=7，

对寄存器寻址，操作数就是寄存器中的数值 7

对寄存器间接寻址，从内存 7# 单元读出来的数才是操作数

变址寻址

操作数的地址由指定的变址寄存器（由Reg指定）的内容和指令中的变址偏移量（Disp）相加得到。



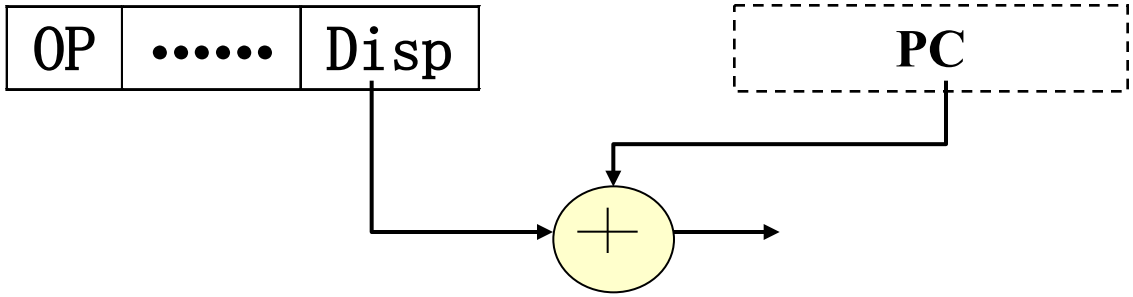
例：Disp=18H，Reg=5，(R5)=5700H

则操作数地址 = 5718H

变址寄存器内容变化，变址偏移量不变，便于读写数组中的元素，是计算机中常用的一种寻址方式。

相对寻址

指令的地址由程序计数器 PC 的内容（即当前执行指令的地址）和指令的相对寻址偏移量相加得到。



例：Disp = 48H　　(PC) = 5600H

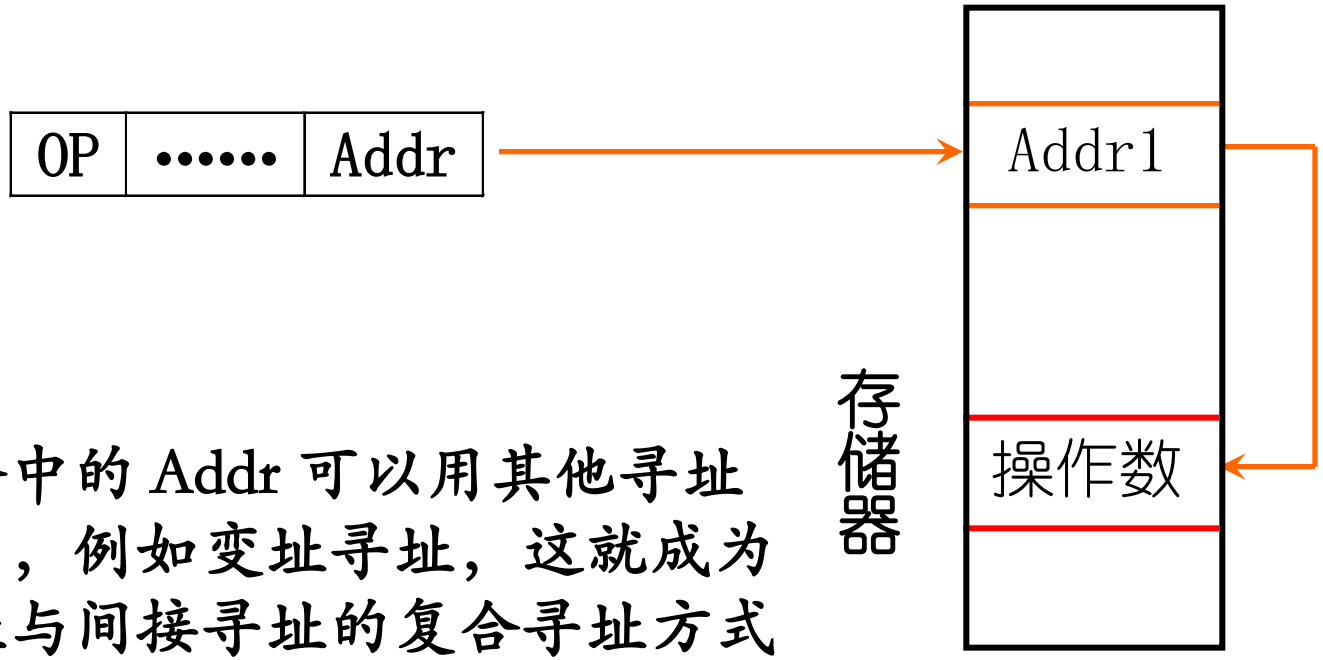
则实际地址 = 5648H

（1）主要用于转移指令，对浮动程序很有用。

（2）偏移量可正可负，通常用补码表示。

间接寻址

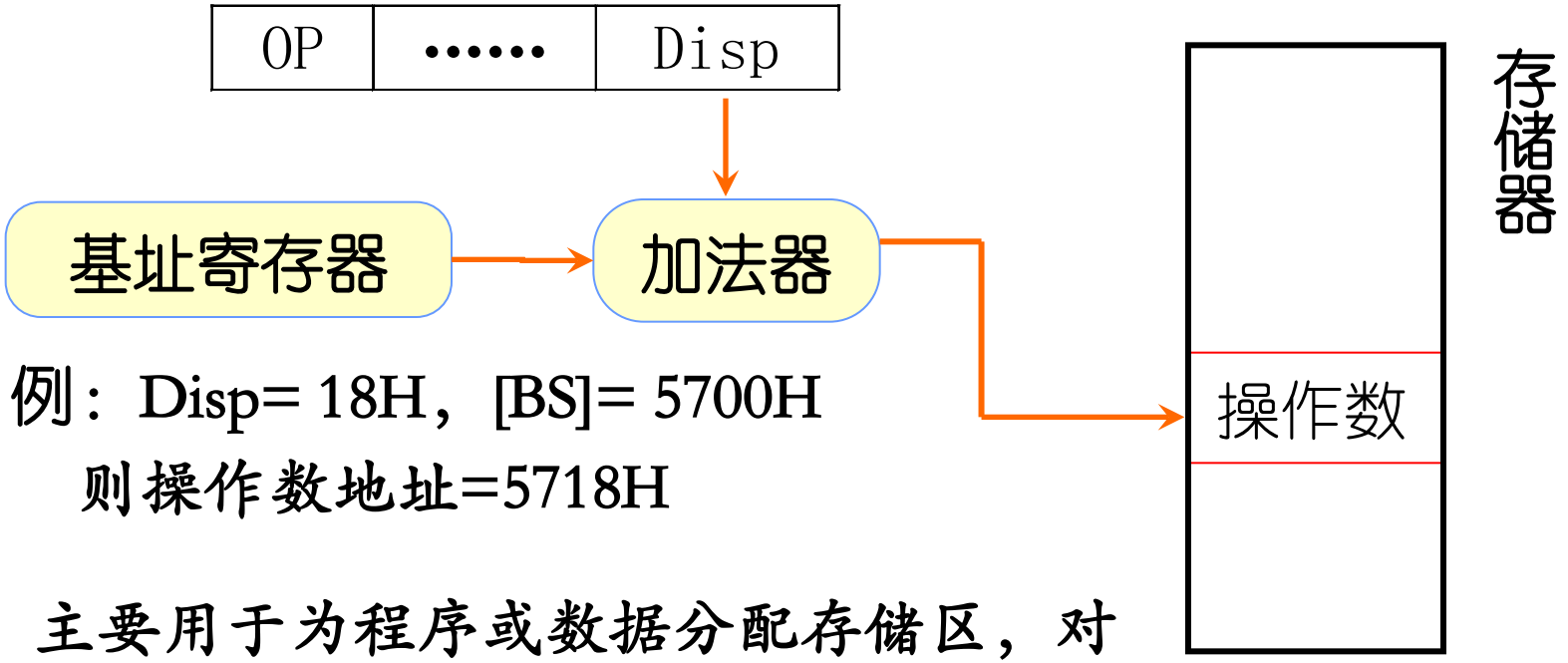
指令的地址码字段给出的内容既不是操作数，也不是操作数的地址，而是操作数（或指令）地址的地址，这被称为间接寻址方式，多一次读内存储器的操作。



指令中的 Addr 可以用其他寻址方式给出，例如变址寻址，这就成为变址寻址与间接寻址的复合寻址方式

基址寻址

在计算机中设置一个专用的基址寄存器，操作数（或指令）的地址通过基址寄存器的内容和指令中的地址码相加得到。



例：Disp= 18H，[BS]= 5700H

则操作数地址=5718H

主要用于为程序或数据分配存储区，对多道程序或浮动程序很有用，解决了程序在存储器中的定位和扩大寻址空间等问题。

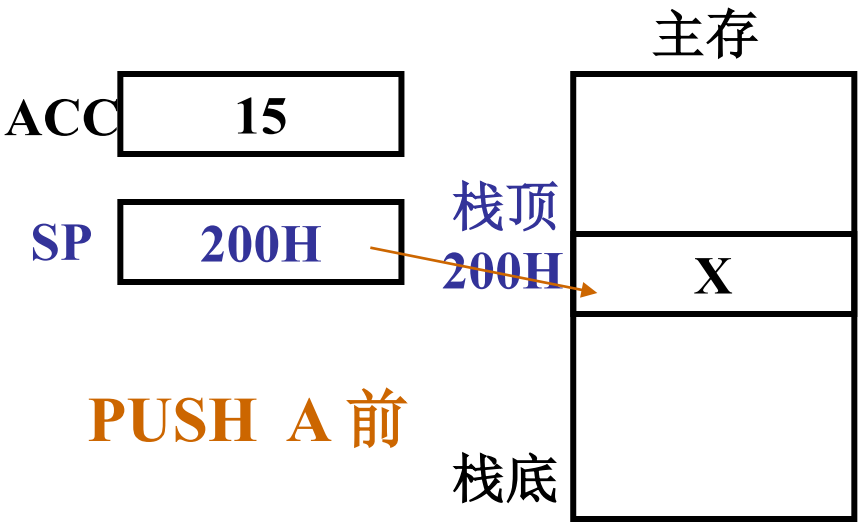
堆栈寻址

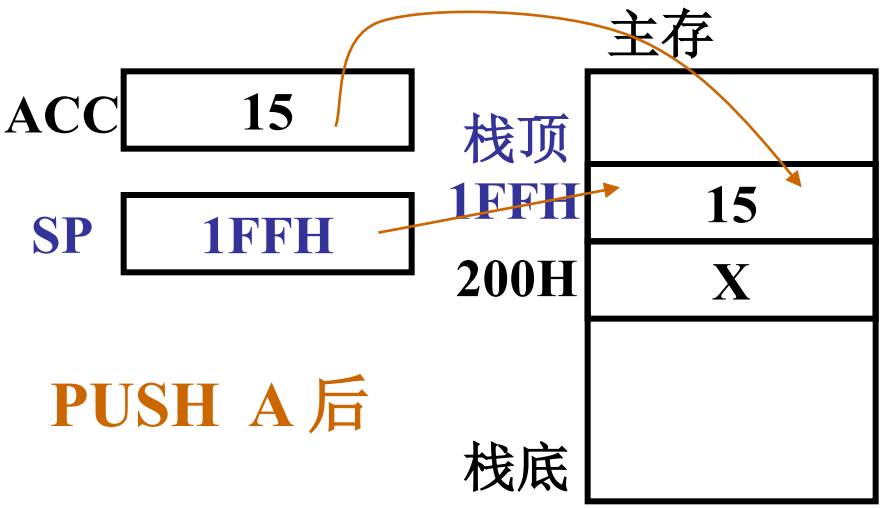
堆栈是内存储器中一块按“后进先出”原则进行读写的存储区，并通过一个专用的寄存器(称为堆栈指针 SP)给出堆栈的栈顶地址，执行读写堆栈操作通常总在栈顶进行，故不必在指令中给出堆栈地址，而且在读写操作的前后伴随有自动修改 SP 内容的动作，确保使 SP 总是指向堆栈的栈顶。例如，按字寻址时：

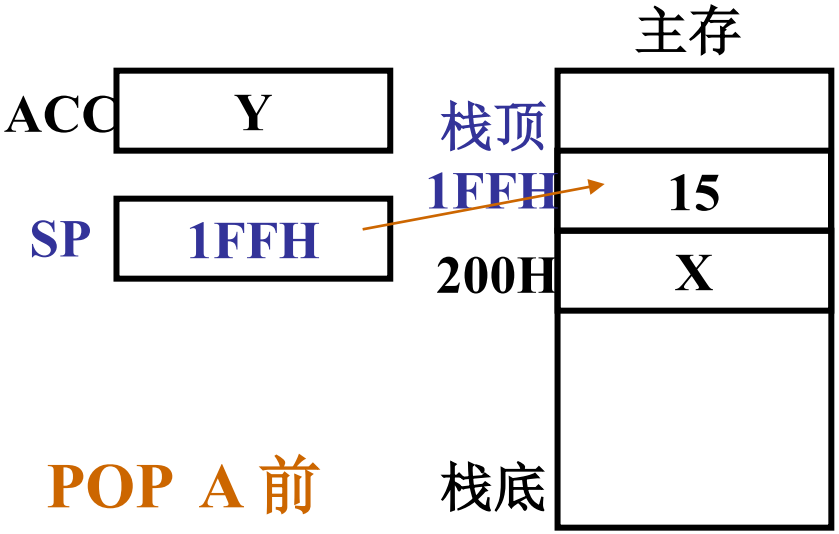
入栈操作：SP -1→ SP 和 AR，即 SP 的内容减 1 存回 SP,并送入内存地址寄存器，接下来才可以把数据写到堆栈中，这是因为需要把数据写到新开辟出来的栈顶单元中。

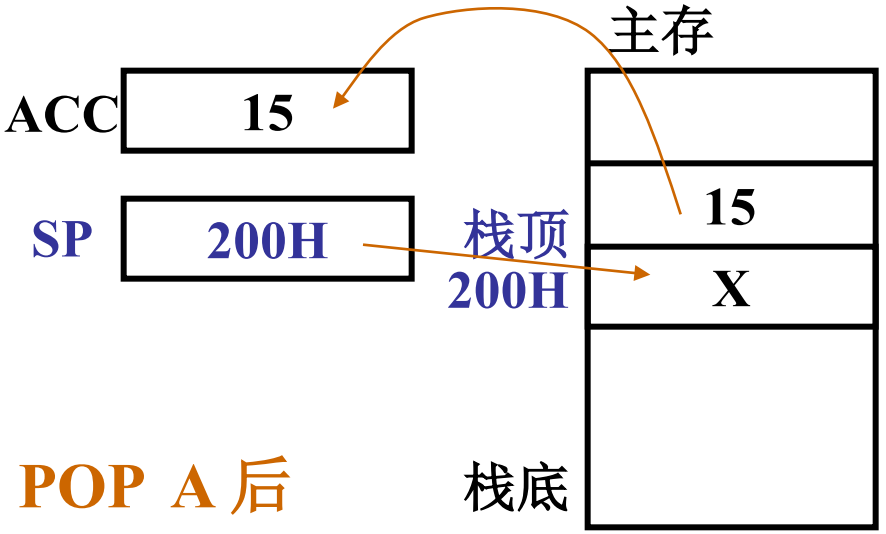
出栈操作：SP → AR，完成一次读堆栈操作后，还要执行一次 SP +1→ SP 的操作，用于修改 SP 内容，这是因为数据读出后原来它的下一个相邻单元变成为栈顶。

堆栈寻址举例


主存
ACC
15
SP
200H
栈顶
200H
X
栈底
PUSH A 前


主存
ACC
15
SP
1FFH
栈顶
1FFH
15
200H
X
栈底
PUSH A 后


主存
ACC
Y
SP
1FFH
栈顶
1FFH
15
200H
X
栈底
POP A 前


主存
ACC
15
SP
200H
栈顶
200H
15
X
栈底
POP A 后

小结

- 控制器的主要功能
  - 正确执行指令规定的功能
  - 自动、连续执行指令
- 指令系统
  - 是硬件和软件的接口
  - CISC和RISC
- 指令格式
  - 指令如何用二进制编码表示，主要是操作码和操作数地址的安排方案
- 寻址方式
  - 操作数寻址方式
  - 对操作系统、编译程序提供支持，方便程序员使用计算机
  - 寻址方式的选择

阅读和思考

- 阅读
  - 教材相关章节
  - 计算机硬件系统实验教程：第2章
- 思考
  - 计算机指令系统中哪些是必备指令?为什么?
  - 指令寻址方式有哪些?这些寻址方式可以在高级语言程序中找到哪些影子?
  - 分析THCO MIPS指令系统的在寻址方式和指令格式方面的特点。
  - 根据THCO MIPS指令系统要求，确定ALU应具备的功能。