

计算机体系结构--

指令系统的设计

zhaofangbupt@163.com



北京邮电大学

BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

主要内容

- ◆ 1. 指令系统结构的分类.....●
- ◆ 2. 操作数的类型和数据表示.....●
- ◆ 3. 寻址方式.....●
- ◆ 4. 指令系统的设计和优化.....●
- ◆ 5. 指令系统的发展和改进.....●
- ◆ 6. MIPS/DLX指令系统结构.....●

指令系统的改进方向

□ 指令系统的优化设计有两个截然相反的方向：

➤ 复杂指令系统计算机CISC

- 增强指令功能，设置功能复杂的指令
- 面向目标代码、面向高级语言、面向操作系统
- 用一条指令代替一串指令
 - 传统主流：IBM370、VAX、微机IntelX86、AMD

➤ 精简指令系统计算机RISC

- 简化指令功能，只保留功能简单的指令
- 较复杂的功能用子程序来实现
 - 服务器主流：SUN SPARC、IBM PowerPC

沿着CISC方向发展和改进

□ CISC结构追求的 **目标**

- 强化指令功能，减少程序的指令条数，以达到提高性能的目的

□ 增强指令功能主要是从以下几个方面着手：

- 面向目标程序增强指令功能
- 增强运算型指令的功能
- 增强数据传送指令的功能
- 增强程序控制指令的功能
- 丰富的程序控制指令为编程提供了多种选择

沿着CISC方向发展和改进

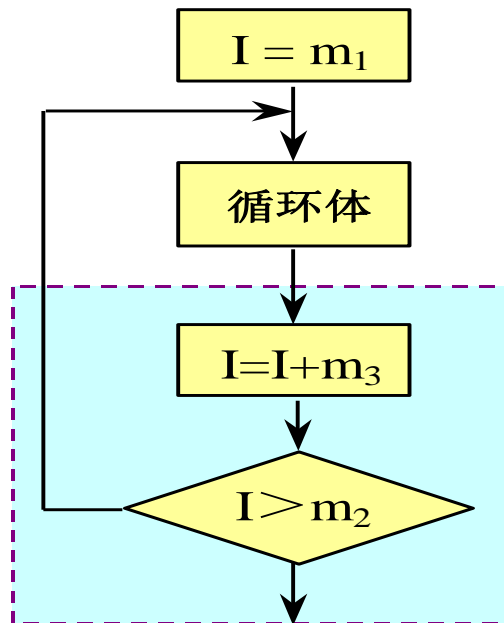
□ 举例：循环程序

➤ 循环控制部分：

- 一条加法指令
- 一条比较指令
- 一条分支指令
- 设置循环控制指令

□ 增加了硬件的成本和复杂度

□ 仅用于频繁使用的子程序或指令串



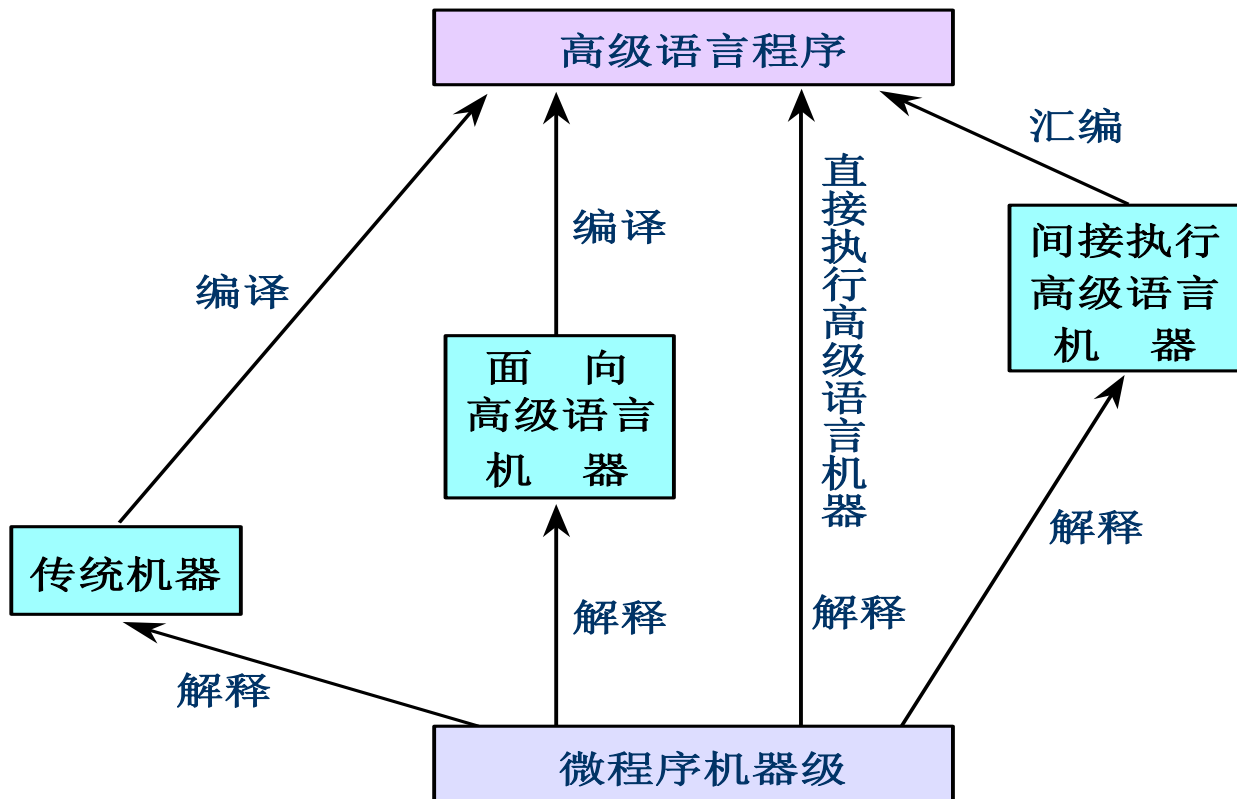
沿着CISC方向发展和改进

□ 面向高级语言的优化实现来改进指令系统

➤ 目标

- 缩小高级语言与机器语言的语义差距
- 高级语言与一般的机器语言的语义差距非常大，为高级语言程序的编译带来了一些问题
 - 编译器本身比较复杂
 - 编译生成的目标代码比较难以达到很好的优化

沿着CISC方向发展和改进



沿着CISC方向发展和改进

□ 具体优化的**方法**：统计指令和指令串的使用频度

➤ 静态使用频度：侧重减少占用空间

- 哈夫曼方法

➤ 动态使用频度：侧重减少运行时间

- 减少访存

➤ 高频指令：增强功能，加快执行，缩短长度

➤ 高频指令串：增加新指令，软件向后兼容

➤ 低频指令：取消或合并到其他指令中

沿着CISC方向发展和改进

□ 增强对高级语言和编译器的支持

- 增强高级语言中使用频度高、执行时间长的语句，应增强有关指令的功能，加快这些指令的执行速度
 - 一元赋值语句在高级语言程序中所占的比例最大
 - 条件转移和无条件转移语句所占的比例也比较高
- 增加专门的指令，可以达到减少目标程序的执行时间和减少目标程序长度的目的
- 增强系统结构的规整性，减少系统结构中的各种例外情况

沿着CISC方向发展和改进

□ 面向高级语言的计算机

➤ 间接执行高级语言机器

- 高级语言成为机器的汇编语言，高级语言和机器语言是一一对应的
- 用汇编方法把高级语言源程序翻译成机器语言程序

➤ 直接执行高级语言的机器

- 直接把高级语言作为机器语言，直接由固件/硬件对高级语言源程序的语句逐条进行解释执行
- 这时既不用编译，也不用汇编

沿着CISC方向发展和改进

□ 面向操作系统的优化实现改进指令系统

- 操作系统和计算机系统结构是紧密联系的，操作系统的实现在很大程度上取决于系统结构的支持
- 指令系统对操作系统的支持主要有：
 - 处理机工作状态和访问方式的切换
 - 进程的管理和切换
 - 存储管理和信息保护
 - 进程的同步与互斥，信号灯的管理等
- 操作系统指令属于特权指令，一般用户程序不能使用

沿着CISC方向发展和改进

□ 核心：缩短OS与系统结构语义差距，减少OS辅助操作时间，节省OS占用空间

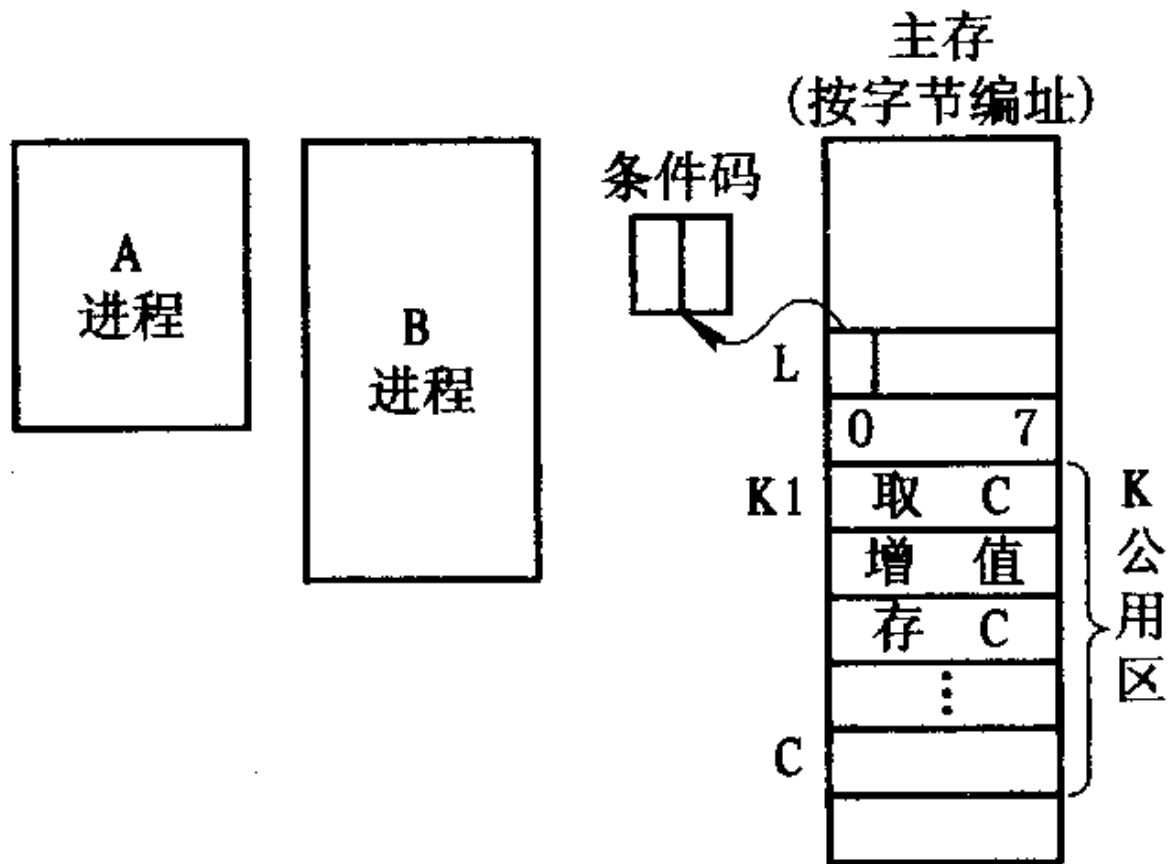
➤ 思路1：统计OS常用指令和指令串的使用频度，进行分析改进

- 效果有限

➤ 思路2：增设专用于支持OS的新指令

- 进程使用公用区：测试与设置指令
- 多进程防止死锁：比较与交换指令

沿着CISC方向发展和改进



沿着CISC方向发展和改进

➤ **思路3**：硬件、固件实现OS的某些功能——使用频繁、影响速度

- 为进程切换设置相关指令
 - 保存进程关联信息、恢复进程关联信息
- 堆栈机HP-300设置PCAL、EXIT指令
 - 支持程序嵌套与递归调用

➤ **思路4**：发展由专门的处理机完成OS功能的分布式系统结构

从CISC到RISC

□ CISC指令集结构存在的问题

➤ 各种指令的**使用频度相差悬殊**

- 据统计：只有20%的指令使用频度比较高，占运行时间的80%，而其余80%的指令只在20%的运行时间内才会用到
- 使用频度高的指令也是最简单的指令

从CISC到RISC

执行频度排序	80x86指令	指令执行频度（占执行指令总数的百分比）
1	load	22%
2	条件分支	20%
3	比较	16%
4	store	12%
5	加	8%
6	与	6%
7	减	5%
8	寄存器-寄存器间数据移动	4%
9	调用子程序	1%
10	返回	1%
合 计		95%

从CISC到RISC

➤ 指令集庞大，指令条数很多，许多指令的功能又很复杂，使得控制器硬件非常复杂

➤ 导致的问题：

- 占用了大量芯片面积，给VLSI设计造成很大的困难
- 增加了研制时间和成本，容易造成设计错误
- 许多指令由于操作繁杂，CPI值比较大，执行速度慢
- 由于指令功能复杂，规整性不好，不利于采用流水技术来提高性能

沿着RISC方向发展和改进

□ 设计RISC机器遵循的原则

➤ 指令条数少而简单

- 只选取使用频度很高的指令，在此基础上补充一些最有用的指令

➤ 采用简单而又统一的指令格式，并减少寻址方式

- 指令字长都为32位或64位

➤ 指令的执行在单个机器周期内完成

- 采用流水线机制

沿着RISC方向发展和改进

- 只有 **load** 和 **store** 指令才能访问存储器，其他指令的操作都是在寄存器之间进行
 - 即采用 **load-store** 结构
- 大多数指令都采用硬连逻辑来实现
- 强调优化编译器的作用，为高级语言程序生成优化的代码
- 充分利用流水技术来提高性能

沿着RISC方向发展和改进

□ 早期的RISC微处理器

- 1981年，Patterson等人的32位微处理器RISC I：
 - 31条指令，指令字长都是32位，78个通用寄存器，时钟频率为8 MHz
 - 控制部分所占的芯片面积只有约6%。商品化微处理器MC68000和Z8000分别为50%和53%
 - RISC I性能比MC68000和Z8000快3~4倍
- 1983年的RISC II
- 1981年，Stanford大学Hennessy等人的MIPS
- IBM的801

沿着RISC方向发展和改进

□RISC为了使流水线高效率执行，应具有以下特征：

- 简单而统一的指令格式，加快指令译码
- 大部分指令在单周期完成
- 只有LOAD/STORE指令能够访问主存
- 简单的寻址方式
- 采用延迟转移技术
- LOAD延迟技术

沿着RISC方向发展和改进

□ **减少CPI**是RISC思想的精华

➤ 程序执行时间的计算公式：

$$P = I \cdot CPI \cdot T$$

类型	指令条数IC	指令平均周期数CPI	时钟周期
CISC	1	$2 \sim 15$	$33 \sim 5\text{ns}$
RISC	$1.3 \sim 1.4$	$1.1 \sim 1.4$	$10 \sim 2\text{ns}$

➤ RISC速度要比CISC快3倍，关键是RISC的CPI减小了

CISC和RISC发展的融合

□ RISC的出现是计算机系统结构发展最重要的变革之一

➤ CISC通过吸收RISC设计思想提高了传统机器的性能

➤ RISC指令系统也开始采用一些复杂而必要的指令，使RISC计算机结构也日益复杂

□ 发展趋势来看，RISC与CISC的优势互补和技术交融将会持续下去

内容小结

□ 指令系统的发展和改进

➤ 沿着CISC方向发展和改进

- 面向目标程序增强指令功能
- 面向高级语言的优化
- 面向OS的优化实现

➤ 从CISC到RISC的演变

- CISC指令系统结构存在的问题

➤ 沿着RISC方向发展和改进

➤ 发展趋势，RISC与CISC的优势互补和技术交融

内容小结

□ 知识要点

➤ CISC

➤ RISC

练习题

- 1. 简述CISC指令集结构功能设计的主要目标。从当前的计算机技术观点来看，CISC指令集结构的计算机有什么缺点？
- 2. 简述RISC指令集结构的设计原则？
- 3. 根据CPU性能公式简述RISC指令集结构计算机和CISC指令集结构计算机的性能特点。

Thank You !

zhaofangbupt@163.com



北京邮电大学

BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS