

计算机体系结构--

# 指令系统的设计

[zhaofangbupt@163.com](mailto:zhaofangbupt@163.com)



北京邮电大学

BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

# 指令系统基本概念

- 指令系统是计算机**所有指令的集合**
- 指令系统是计算机系统结构的主要内容，是硬件和软件之间的接口
  - 计算机具有哪些硬件功能，是硬件逻辑设计的基础
- 指令系统也需要为编译器提供明确的编译目标，使编译结果具有规律性和完整性
- 指令系统是计算机系统软硬件功能分配的界面，也是计算机系统结构设计的核心
  - 其功能对计算机系统的性能有直接的影响

# 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.....●

# 指令系统结构的分类

## □ 指令系统结构：

- 指令系统的结构 (Instruction Set Architecture)

## □ 区别不同指令集结构的主要因素

- CPU中用来存储操作数的存储单元的类型
- 指令中显式表示的操作数个数
- 操作数的寻址方式
- 指令集所提供的操作类型
- 操作数的类型和大小

# 指令系统结构的分类

□ CPU中用于存储操作数的存储单元类型主要有：

- 堆栈
- 累加器
- 通用寄存器组

□ 将指令系统结构分为三种类型：

- 堆栈结构
- 累加器结构
- 通用寄存器结构

# 指令系统结构的分类

## □ CPU对操作数的不同存取方式

CPU提供的 暂存器	每条ALU指令显式表示 的操作数个数	运算结果的 目的地	访问显式操作数 的过程
堆栈	0	堆栈	Push/Pop
累加器	1	累加器	Load/Store 累加器
寄存器组	2/3	寄存器或存 储器	Load/Store寄存 器或存储器

# 指令系统结构的分类

□ 根据操作数的来源不同，又可进一步分为：

➤ 寄存器-存储器结构 (RM结构)

- 操作数来自可以来自存储器

➤ 寄存器-寄存器结构 (RR结构)

- 所有操作数都是来自通用寄存器组
- 也称为 **load-store** 结构，这个名称强调：只有 load 指令和 store 指令能够访问存储器

➤ 存储器-存储器结构 (MM结构)

➤ 扩展累加器或者专用寄存器结构

# 指令系统结构的分类

□ 对于不同类型的指令集结构，操作数的位置、个数以及操作数的给出方式（**显式**或**隐式**）也会不同

➤ **隐式给出**：使用事先约定好的存储单元

- 堆栈结构：操作数隐式放在栈的顶部
- 累加器结构：部分操作数是隐式的

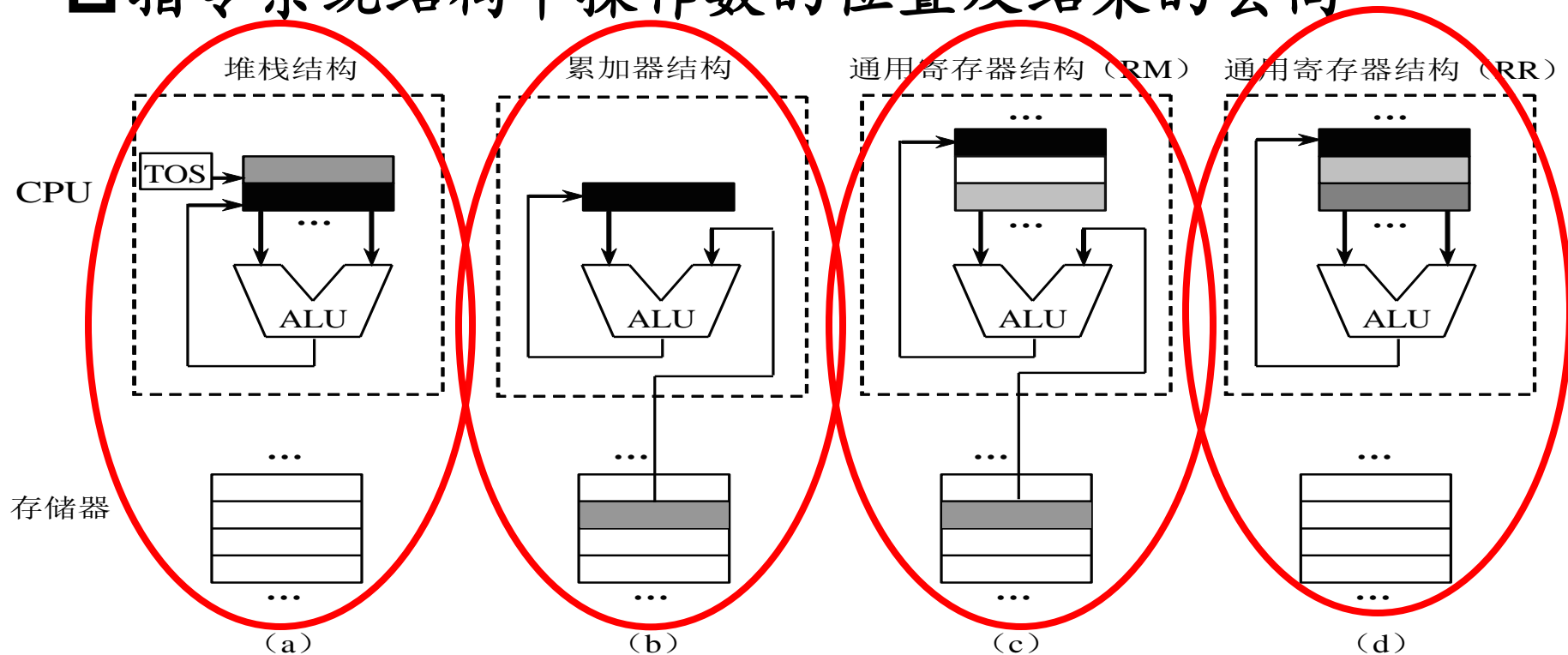
➤ **显式给出**：用指令字中的操作数字段给出

- 通用寄存器体系结构：只有显式操作数



# 指令系统结构的分类

## □ 指令系统结构中操作数的位置及结果的去向



# 指令系统结构的分类

□ 【例1】表达式 $C=A+B$ 在4种类型指令集结构上的代码

➤ 假设：A、B、C均保存在存储器单元中，并且不能破坏A和B的值

堆 栈	累加器	寄存器（RM型）	寄存器（RR型）
push A	load A	load R1, A	load R1, A
push B	add B	add R1, B	load R2, B
add	store C	store R1, C	add R3, R1, R2
pop C			store R3, C

# 指令系统结构的分类

## □ 各种指令系统类型比较

### ➤ 堆栈型

- **优点**：指令字比较短，程序占用的空间比较小
- **缺点**：不能随机地访问堆栈，难以生成有效的代码，而对栈顶的访问是个瓶颈

### ➤ 累加器型

- **优点**：指令字比较短，程序占用的空间比较小
- **缺点**：只有一个中间结果暂存器，需要频繁访问存储器

# 指令系统结构的分类

## □ 通用寄存器结构

- 现代指令系统结构的**主流**
- 在灵活性和提高性能方面有明显的优势：
  - 寄存器的访问速度比存储器快的多
  - 对编译器而言，能更加容易、有效地分配和使用寄存器
  - 在表达式求值方面，通用寄存器型结构具有更大的灵活性和更高的效率： $(A*B) - (C*D) - (E*F)$

# 指令系统结构的分类

□ 寄存器可以用来**存放变量**:

➤ 能加快程序的执行速度

- 寄存器比存储器快

➤ 能够减少对存储器的访问

➤ 用更少的地址位来对寄存器进行寻址，从而有效地减少程序的目标代码所占用的空间

- 相对于存储器地址来说

# 指令系统结构的分类

## □ CPU需要设置寄存器的个数：

➤ 主要由编译器使用寄存器的情况来决定

- 为表达式求值保留一些寄存器
- 为传递参数保留一些寄存器
- 用剩下的寄存器来保存变量

## □ 通用寄存器和专用寄存器个数的选择：

- 早期计算机，相当一部分用作专用寄存器
- 现代计算机中通用寄存器的个数已越来越多

# 指令系统结构的分类

□ 根据ALU指令的操作数特征对通用寄存器型指令集结构进一步细分：

➤ ALU指令的操作数个数

- 3个操作数的指令（两个源操作数、一个目的操作数）
- 2个操作数的指令（其中一个操作数既作为源操作数，又作为目的操作数）

➤ ALU指令中存储器操作数的个数

- 可以是0~3中的某一个值，0表示没有存储器操作数

# 指令系统结构的分类

## □ 存储器操作数个数和操作数个数的所有可能组合

ALU指令中存储器操作数的个数	ALU指令中操作数的最多个数	结构类型	机器实例
0	3	RR	MIPS, SPARC, Alpha, PowerPC, ARM
1	2	RM	IBM 360/370, Intel 80x86, Motorola 68000
	3	RM	IBM 360/370
2	2	MM	VAX
3	3	MM	VAX



# 指令系统结构的分类

## □ 通用寄存器型指令集结构进一步细分为3种类型

### ➤ 寄存器-寄存器型

- **RR型**: register-register

### ➤ 寄存器-存储器型

- **RM型**: register-memory

### ➤ 存储器-存储器型

- **MM型**: memory-memory

# 指令系统结构的分类

指令集结构类型	优 点	缺 点
寄存器—寄存器型 (0, 3)	指令字长固定，指令结构简洁，是一种简单的代码生成模型，各种指令的执行时钟周期数相近	与指令中含存储器操作数的指令集结构相比，指令条数多，目标代码不够紧凑，因而程序占用的空间比较大
寄存器—存储器型 (1, 2)	可以在ALU指令中直接对存储器操作数进行引用，而不必先用load指令进行加载。容易对指令进行编码，目标代码比较紧凑	指令中的两个操作数不对称。在一条指令中同时对寄存器操作数和存储器操作数进行编码，有可能限制指令所能够表示的寄存器个数。指令的执行时钟周期数因操作数的来源（寄存器或存储器）不同而差别比较大
存储器—存储器型 (2, 2) 或 (3, 3)	目标代码最紧凑，不需要设置寄存器来保存变量	指令字长变化很大，特别是3操作数指令。而且每条指令完成的工作也差别很大。对存储器的频繁访问会使存储器成为瓶颈。这种类型的指令集结构现在已不用了

# 指令系统分类

## □ 指令格式与指令字长对编译的影响

- 指令格式和指令字长越单一，编译器的工作就越简单
- 指令格式和指令字长具有多样性，可有效减少目标代码所占的空间
- 多样性也可能会增加编译器和CPU实现的难度
- CPU寄存器的个数也会影响指令的字长

## □ 通用寄存器型结构比堆栈型结构和累加器型结构更具有优势

- 寄存器-寄存器型结构备受青睐

# 内容小结

□ 指令系统基本概念

□ 指令系统结构的分类

- 在CPU中操作数的存储方法
- 指令中显式表示的操作数个数
- 操作数的寻址方式
- 指令系统所提供的操作类型
- 操作数的类型和大小

# 内容小结

## □ 知识要点

堆栈型结构	累加器型结构	通用寄存器型结构
寄存器-存储器型结构	存储器-存储器型结构	寄存器-寄存器型结构

# 练习题

- 1. 区别不同指令系统结构的主要因素是什么？根据这个主要因素可将指令系统结构分为哪三类？
- 2. 常见的三种通用寄存器型机器的优缺点各有哪些？

# Thank You !

[zhaofangbupt@163.com](mailto:zhaofangbupt@163.com)



北京邮电大学

BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS