

指令系统

▼ 重点题型

- 操作码编码方案（特别是扩展操作码编码） 5.9
- 结合地址码的情况，分析扩展操作码的可行性和指令数量的变化情况 5.10，5.12
- 数据存储<大端，小端> 5.14 **注意可能会结合后面寻址结果**
- 依据基本寻址方式，分析新的寻址方式（主要是寻址方式的名称，最后的寻址结果）
- 设计指令格式
- 根据指令功能，设计某一具体指令

▼ 术语

- 指令：控制计算机硬件完成指定基本操作的命令
- 指令系统：指令集合
- 指令字长：指令的位数
- 操作码：指定指令要完成的功能
- 地址码（操作数）：提供该指令的操作对象
- 寻址方式：指令获取操作数的方式
- CISC：复杂指令系统
- RISC：精简指令系统

▼ 指令系统概述

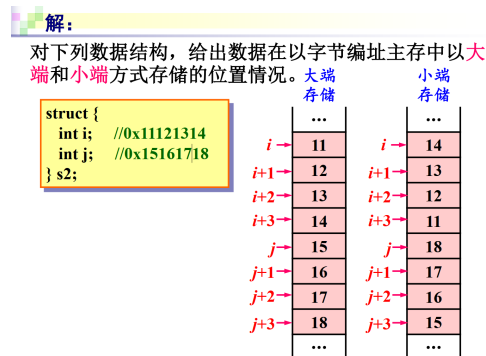
- 指令系统是从程序设计者看到的机器的主要属性，是软、硬件的主要界面
- 指令系统的设计主要包括**指令功能**和**指令格式**的设计。

▼ 指令系统结构层定义

▼ 存储模式

- 数据存储顺序
 - 大端存储【字符打印】

- 小端存储【算术运算】
- 例题



- 边界对齐
 - 按字节编址，边界对齐
- 堆栈
 - 管理堆栈<堆栈指针，堆栈基址，堆栈界限>



堆栈界限 > 堆栈基址 VS 堆栈界限 < 堆栈基址

- 冯·诺依曼结构和哈佛结构
 - 优缺点分析【主存空间利用，数据操作对指令的破坏性，】
- 加载/存储体系结构
 - Load，Store指令
- 寄存器组织
 - 软件设计者唯一能操作的CPU内部资源



标志寄存器常作为条件跳转指令的跳转条件

- 数据类型
- I/O模式
 - 统一编址
 - 独立编址

- 指令类型
 - 程序控制类
 - 转移指令<无条件转移指令, 条件转移指令>
 - 循环控制指令
 - 过程调用和返回指令
 - 程序自中断指令

▼ 指令设计

- 指令格式<操作码, 操作数>
- 指令长度
 - 指令长度 = 操作码 + 地址码 (注意按字节向上取整)

▼ 地址码设计

源操作数	目的操作数	下条指令地址
------	-------	--------

四地址指令: `op rd, rs1, rs2, ni`
 ; `rs1 op rs2 → rd`; `ni` 提供顺序或转移地址
 三地址指令: `op rd, rs1, rs2`
 ; `rs1 op rs2 → rd`; `PC` 提供顺序地址
 二地址指令: `op rd, rs1`
 ; `rd op rs1 → rd`; `PC` 提供顺序地址 或
 ; `rs1 op ACC → rd`; `PC` 提供顺序地址

 一地址指令: `op rd`
 ; `rd op ACC → ACC`; `PC` 提供顺序地址 或
 ; `rd` 自身操作 `→ rd`; `PC` 提供顺序地址 或
 ; 由 `rd` 提供转移地址
 零地址指令: `op`
 ; 由操作码指定操作数(隐含寻址) 或
 ; 无需操作数

▼ 操作码设计

- 定长操作码
- 变长操作码
 - 霍夫曼编码

- 扩展操作码（为充分利用指令的二进制位）



拓展操作码的设计

这只是其中一种扩展方法，还有其他多种扩展方法。

操作码编码	说明
0000 0001 1110	4位长度的操作码 共有15种
1111 0000 1111 0001 1111 1110	8位长度的操作码 共有15种
1111 1111 0000 1111 1111 0001 1111 1111 1110	12位长度的操作码 共有15种

等长 15/15/15...扩展法

操作码编码	说明
0000 0001 0111	4位长度的操作码 共有8种
1000 0000 1000 0001 1111 0111	8位长度的操作码 共有64种
1000 1000 0000 1000 1000 0001 1111 1111 0111	12位长度的操作码 共有512种

等长 8/64/512...扩展法

▼ 寻址方式

- 基本寻址方式



寻址方式在指令中的体现：

由操作码决定（隐含寻址），操作数中设置寻址方式字段（显式寻址）

隐含寻址	
立即寻址	
寄存器寻址	寄存器编号
直接寻址	操作数地址
间接寻址	操作数地址的地址→操作数地址
寄存器间接寻址	寄存器编号→操作数地址
相对寻址	地址偏移 (+PC)
基址寻址	基址寄存器寻址位+地址偏移
变址寻址	变址寄存器寻址位+地址偏移
堆栈寻址	

• RISC-V 和 x86 寻址方式



额外内容

RISC-V		x86	
立即寻址		立即寻址	
寄存器寻址		寄存器寻址	
基址寻址		主存储器寻址	
相对寻址			

• 例题

习题 5.20 (按字节编址) (小端存储)

说明下列指令执行后R1的值。R1为16位寄存器。

寄存器	地址	内存
...
RB 0100H	0100H	12H
RI 0002H	0101H	34H
...	0102H	56H
...	0103H	78H
...
...	1200H	2AH
...	1201H	4CH
...	1202H	B7H
...	1203H	65H
...

① MOV R1, #1200H
 ② MOV R1, RB
 ③ MOV R1, (1200H)
 ④ MOV R1, (RB)
 ⑤ MOV R1, 1100H(RB)
 ⑥ MOV R1, (RB)(RI)
 ⑦ MOV R1, 1100H(RB)(RI)

【解】 ① 1200H ⑤ 4C2AH
 R1 = ② 0100H ⑥ 7856H
 ③ 4C2AH ⑦ 65B7H
 ④ 3412H

▼ 指令系统结构的发展

▼ CISC 复杂指令系统计算机

- 指令系统复杂庞大
- 指令长度不固定，指令格式种类多，寻址方式种类多
- 可以访存的指令不受限制
- 控制器大多数采用微程序控制

▼ RISC 精简指令系统计算机

- 指令系统简单
- 采用Load/Store结构

- 重视提高流水线的执行效率
- 强调优化编译技术的作用

▼ 指令系统实例



新增，考试重点

- RISC-V 指令系统
- Intel CPU 指令系统
 - Intel 64 和 IA-32
 - IA-32e
 - Intel AVX
- MIPS 指令系统