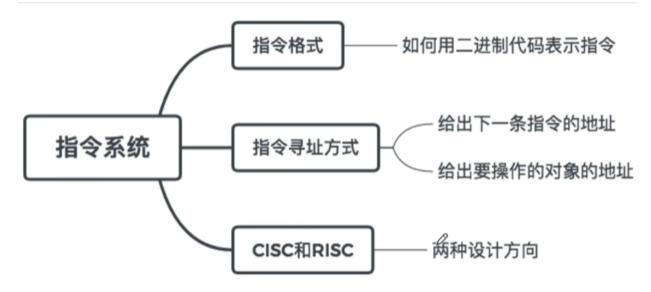
CISC 和 RISC 的比较



CISC 和 RISC

CISC: Complex Instruction Set Computer 设计思路: 一条指令完成一个复杂的基本功能。

代表: x86架构, 主要用于笔记本、台式机等

RISC: Reduced Instruction Set Computer 设计思路: 一条指令完成一个基本"动作"; 多条指令组合完成一个复杂的基本功能。

代表: ARM架构, 主要用于手机、平板等

80-20规律: 典型程序中 80% 的语句仅仅使用处理机中 20% 的指令

比如设计一套能实现整数、矩阵加/减/乘运算的指令集:

CISC的思路:除了提供整数的加减乘指令除之外,还提供矩阵的加法指令、矩阵的减法指令、矩阵的乘法指令

一条指令可以由一个专门的电路完成 有的复杂指令用纯硬件实现很困难 → 采用"存储程序"的设计思想,由一个比 较通用的电路配合存储部件完成一条指令 RISC的思路: 只提供整数的加减乘指令

一条指令一个电路,电路设计相对简单,功耗更低 "并行"、"流水线"

类 别对比项目	CISC	RISC
指令系统	复杂,庞大	简单,精简
指令数目	一般大于200条	一般小于100条
指令字长	不固定	定长
可访存指令	不加限制	只有Load/Store指令
各种指令执行时间	相差较大	绝大多数在一个周期内完成
各种指令使用频度	相差很大	都比较常用
通用寄存器数量	较少	多
目标代码	难以用优化编译生成高效的目标代码程序	采用优化的编译程序,生成代码较为高效
控制方式	绝大多数为微程序控制	绝大多数为组合逻辑控制
指令流水线	可以通过一定方式实现	必须实现