

指令集和体系结构

本文主要探讨一些体系结构层面的东西，如指令集架构（ISA）等。

CISC 和 RISC

CISC（Complex Instruction Set Computer，复杂指令集计算机结构）是早期常用的一种结构。到现在为止 Intel 的 x86 仍属于它。

它的特点在于：

- 指令系统复杂庞大，各种指令使用频度相差大。
- 指令长度不固定、指令格式种类多、寻址方式多。
- CPU 中多为专用寄存器，通用寄存器比较少。
- 采用微程序控制器实现控制。微程序控制器可称为 CPU 中的 CPU，通过它可以实现兼容老版本的指令。
- 指令执行速度不一，有的指令只用一个时钟周期，有的要用很多个。

虽然这样的设计有利于编程，但是缺点也有很多：

- 其指令太多了，大约 80% 的语句只用了 20% 的命令。
- 指令的灵活多样加大了 CPU 设计的难度。
- 执行频度高的简单指令由于复杂指令的存在难以提高执行速度。这是因为因为指令都要受到指令周期的限制。最小周期不能无限缩短，要受到其他指令的牵制。
- 难以采用优化编译程序生成高效的目标代码。

RISC（Reduced Instruction Set Computer，精简指令集计算机结构）是现在发展的比较快的一种结构，MIPS 就属于它。

它的特点在于：

- 选择了使用频度较高的一些简单指令来构成指令集，复杂指令通过简单指令的组合实现。
- 指令定长，种类少，寻址方式少。只有 Load/Store 能访问存储器。
- CPU 中有多个通用寄存器，一般比 CISC 多。
- 采用流水线技术，一个时钟周期完成一条指令。
- 使用组合逻辑实现控制。
- 采用优化的编译程序。

它有很多好处：

- RISC 更能充分利用 VLSI 芯片的面积。
- 借助流水线一类技术，RISC 更能提高计算机运算速度。
- RISC 便于设计，可降低成本，提高可靠性。
- RISC 有利于编译程序代码优化。

当然，这两种设计风格各有长有短。两种技术也在相互的融合。例如 RISC 系统因为芯片集成度的增大和硬件速度的提升变得更加复杂，CISC 的微程序控制器的设计采用了 RISC 的一些思路等。

也有将两者的优点结合起来的办法。例如：

- 对外：对程序员支持 CISC 的指令集，方便编程。
- 对内：在芯片内部将这些指令转换为类 RISC 的指令，然后由硬件设计者实现 RISC 风格的处理器以获取高性能。

指令集架构和指令

之前提过，ISA 是硬件和软件之间的桥梁，可以看作是对程序员的硬件抽象。是体系结构中与设计程序有关的部分。

有的芯片 ISA 相同，但硬件有不同的实现。例如 x86 可以由 Intel 和 AMD 的芯片实现，MIPS 可以由龙芯实现。

指令和数据一样，都以二进制方式保存。指令一般由多个字段，也叫**位域 (Bit Field)** 组成，位域的分布方式就是**指令格式 (Instruction Format)**。以二进制数方式表示的指令为**机器语言 (Machine Language)**。

致谢

本文的主要内容参考自：

- 上海交通大学《计算机组成》（课程代号：EI332）一课的课程材料。
- [一文解读RISC与CISC\(转\)](#)