

01-概述

通用电子数字计算机

- 通用：不是一种专用设备
 - 所有计算机在给予足够时间和容量存储器的条件下，都可以完成同样的计算
 - 当希望完成新的计算时，不需要对计算机重新设计
- 电子（非机械）：采用电子元器件
- 数字（非模拟）：信息采用数字化的形式表

结构和组织

	对程序员	特点	包括	例子
组织	不可见	操作单元及其相互连接（硬件的内部实现）	控制信号、存储技术	使用乘法器/加法器完成乘法
结构	可见	直接影响程序逻辑执行的属性（硬件给编程者的“接口”）	指令集、各类数据类型的大小	提供/不提供乘法指令

两台计算机组织相同、结构不同：硬件一样，一台机器指令集有乘法，另一台没

两台计算机结构相同、组织不同：指令集一致，实现指令集的硬件细节不同（上文乘法的例子）

计算机发展历史

- 真空管→晶体管→集成电路

冯·诺伊曼架构

- 运算器、控制器、存储器、I、O
- 指令和数据以同等地位存放在存储器内，并可按地址访问
- 指令和数据均用二进制表示

 无法区分内存中的内容是指令还是数据。

I/O模块指的是计算机内部负责和I/O外设交互的部分，而不是外设本身。

摩尔定律

- 每过18个月，单芯片上所能包含的晶体管数量翻一番
- 摩尔定律意义：
 - 计算机逻辑电路和存储器的成本显著下降。（原料成本下降、制作成本增加）
 - 更小的尺寸带来更多灵活性和可能性
 - 更强的计算能力带来更多可能性
 - 减小了对电能消耗和冷却的要求
 - 集成电路上的内部连接比焊接更可靠，芯片间的连接更少

计算机性能评估

- CPU：速度
- 内存：速度、容量
- I/O：速度、容量

系统时钟

- 时钟频率（时钟速度，Hz）：单位时间内执行最基本操作的次数
- 时钟周期（周期时间，s）：执行每次基本操作的时间

CPU性能评估指标

- $I_C = \sum_{i=1}^n I_i$ I_C = $\sum_{i=1}^n I_i$: 指令条数
- $CPI = \frac{\sum_{i=1}^n CPI_i \times I_i}{I_C}$ $CPI = \frac{\sum_{i=1}^n CPI_i \times I_i}{I_C}$
- $T = I_C \times CPI \times t$ $T = I_C \times CPI \times t$: 给定程序的执行时间
- $MIPS = \frac{I_C}{T \times 10^6} = \frac{f}{CPI \times 10^6}$ $MIPS = \frac{I_C}{T \times 10^6} = \frac{f}{CPI \times 10^6}$
- $MFLOPS = \frac{N_{FLO}}{T \times 10^6}$ $MFLOPS = \frac{N_{FLO}}{T \times 10^6}$

基准程序 Benchmark

- “跑分软件”
- 一系列基准程序评估性能
- 平均结果：算术/调和平均值？
 - 调和平均数用在 x 的平均无实际意义， $\frac{1}{x}$ 的平均有意义的情况下
 - benchmark的结果是速度，使用调和平均才能反应性能（时间的倒数）
 - 如果测量的是时间，直接算术平均即可

性能设计的基本原则

- 大概率事件优先原则：大概率事件赋予优先处理权、资源使用权
- Amdahl定律：系统性能的最大潜在改进受到系统中无法改进的部分的限制
 - $Speedup = \frac{1}{(1-f) + \frac{f}{N}}$ $Speedup = \frac{1}{(1-f) + \frac{f}{N}}$ (f : 能并行的部分的占比 N : 处理器个数)
 - 提高部件性能对于整体性能的贡献取决于部件在整体中的重要性占比

- 若仅对对计算机中的一部分做性能改进，改进越多，系统获得的效果越小

上一页
计算机组织结构

下一页
02-计算机的顶层视图

最后更新于11个月前

