Chapter 1

计算机概要与技术

David A. Patterson

John L. Hennessy

王喆 284418373@qq.com

课程要求（2023年）

- 课时：64学时
- 期末考试方式：闭卷
- 成绩权重分布：平时30%，期末70%
- 平时成绩包括：作业、课堂测验

1.1计算机革命

- 计算机技术的飞速发展
  - 以摩尔定律为基础（18-24个月）
- 使得许多新应用成为可行
  - 车载计算机（Computers in automobiles）
  - 手机（Cell phones）
  - 人类基因项目（Human genome project）
  - 万维网（World Wide Web）
  - 搜索引擎（Search Engines）

1.1.1计算机的分类

- 个人计算机
  - 广泛使用，运行大量第三方软件
  - 受性价比的制约，对单用户提供良好性能，价格低廉
- 服务器
  - 基于网络访问
  - 容量大，性能良好，可靠性高
  - 服务器的功能和价格有很大的伸缩范围
- 超级计算机
  - 用于高端科学和工程计算
  - 虽然代表了最高的计算能力，但是他们在整个计算机市场份额中所占比例很小。
- 嵌入式计算机
  - 通常和硬件集成在一起
  - 面向单一需求的嵌入式应用通常被严格限制成本或功耗

1.1.2后PC时代

- 个人移动设备 (PMD)
  - 由电池供电
  - 可以连接到网络上
  - 价格只有几百美元
  - 智能手机、平板、电子眼镜
- 云计算
  - 仓储规模计算机 (WSC)
  - 软件即服务(SaaS)
  - 开发者通常在PMD和云上各运行应用的一部分。
  - Amazon and Google

1.1.3你（可）能从本书学到什么

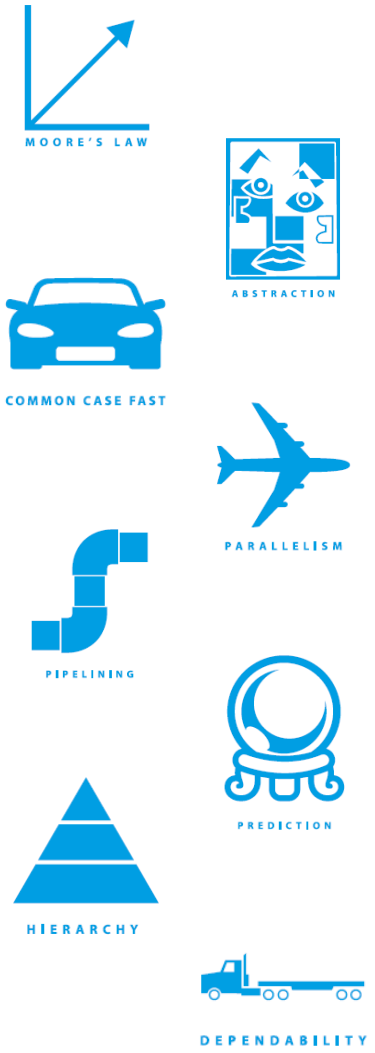
- 内存容量的制约—>处理器的并行性与内存的层次性
- 程序如何翻译成机器语言
  - 硬件如何执行程序
- 软硬件之间的接口
- 哪些因素决定了程序的性能
  - 它的性能如何被改进
- 硬件设计者怎样改进性能
- 什么是并行处理

理解程序性能

- 算法
  - 决定源代码的数量和执行I/O操作的数量
- 程序语言，编译器和体系结构
  - 决定每行源代码对应机器指令的数量
- 处理器和存储器系统
  - 决定指令执行的速度快慢
- I/O系统（硬件和操作系统）
  - 决定I/O操作执行的速度快慢

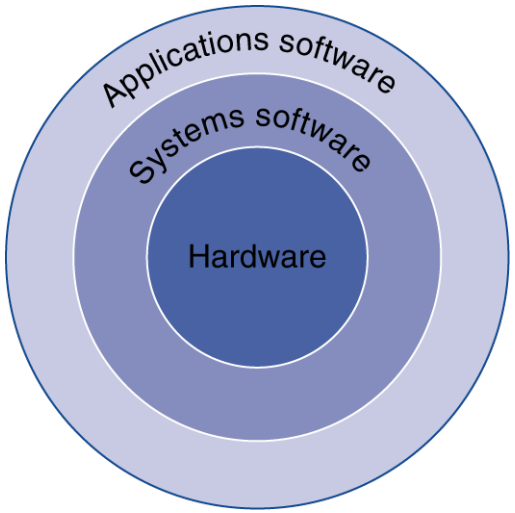
计算机系统结构中的8个伟大思想！

- 面向摩尔定律的设计（前瞻性）
- 使用抽象简化设计
- 加速大概率事件
- 通过并行提高性能
- 通过流水线提高性能
- 通过预测提高性能
- 存储器层次
- 通过冗余提高可靠性



程序概念入门

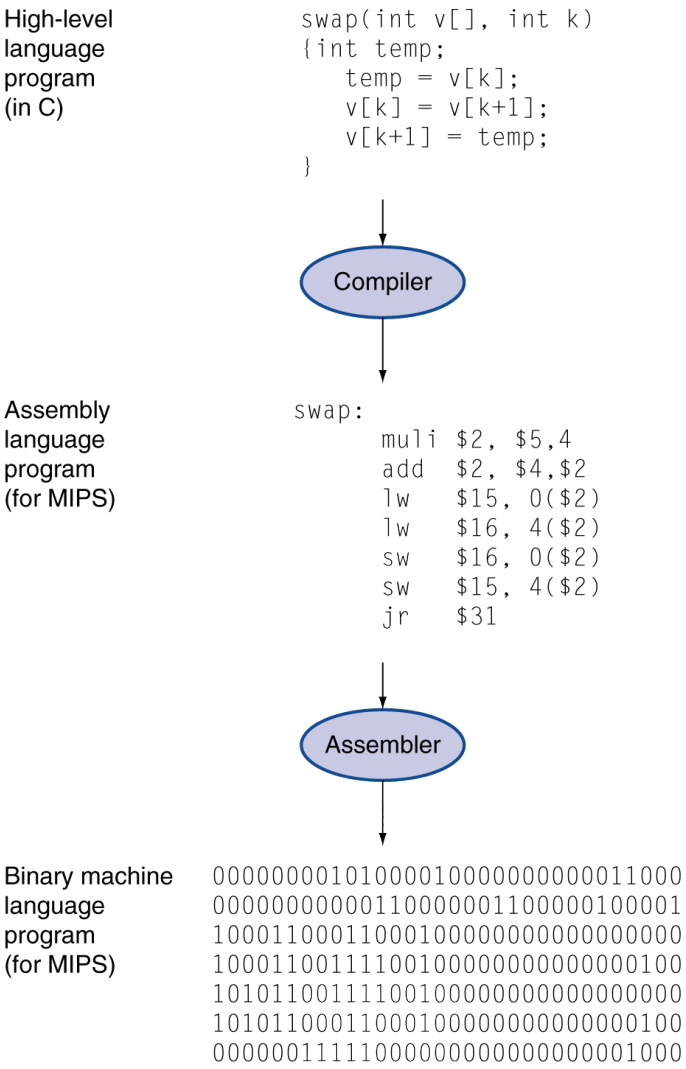
你理解的程序是什么样子？如何执行的？



- 应用软件
  - 用高级语言编写
- 系统软件
  - 编译程序：把用高级语言编写的程序翻译成机器语言
  - 操作系统：提供各种服务和监控
- 硬件
  - 处理器，存储器，输入和输出设备
  - **经过几个软件层次将复杂的高层操作逐步解释或翻译成简单的计算机指令，这是抽象的例子。**

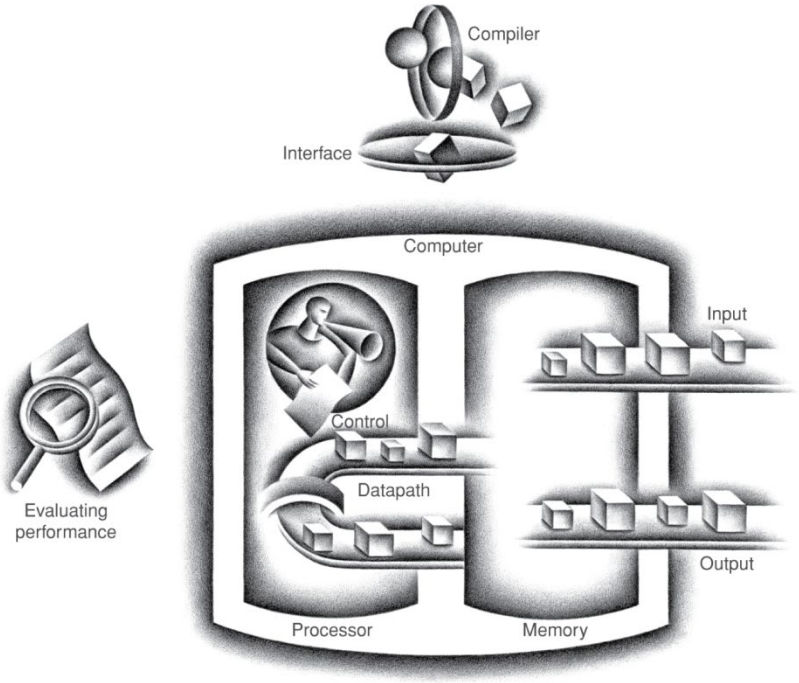
程序语言级别

- 高级语言
  - 最接近问题领域的抽象级别
  - 提供了语言的高效性和可移植性
- 汇编语言
  - 指令的文本表示，助记符形式的指令
- 机器语言
  - 二进制数字（位）
  - 可编码指令和数据
  - **指令与数据都是二进制的**



1.4计算机组成部件

The BIG Picture

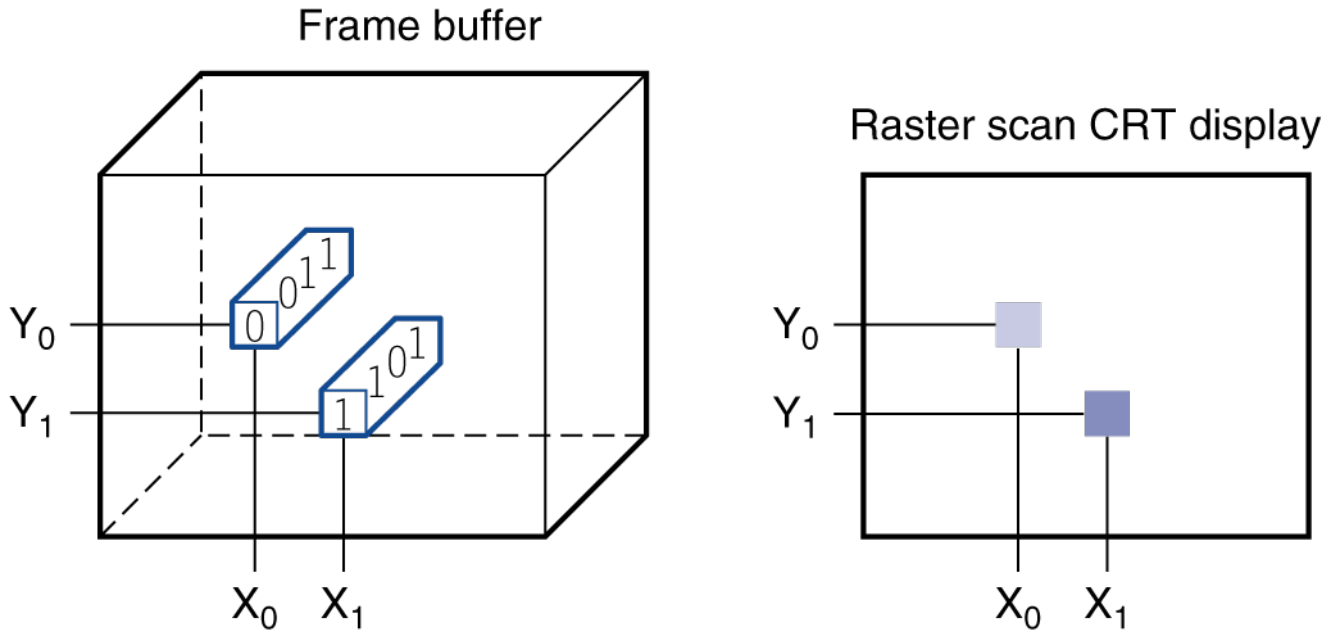


计算机都由哪些部分构成？

- 所有种类的计算机都有相同的组成部件（冯 诺依曼
  - 桌面计算机, 服务器,嵌入式计算机
- 输入和输出部分包括
  - 用户界面设备
    - 显示器，键盘，鼠标
  - 存储设备
    - 硬盘, CD/DVD, 闪存flash
  - 网络适配器
    - 与其他计算机通信

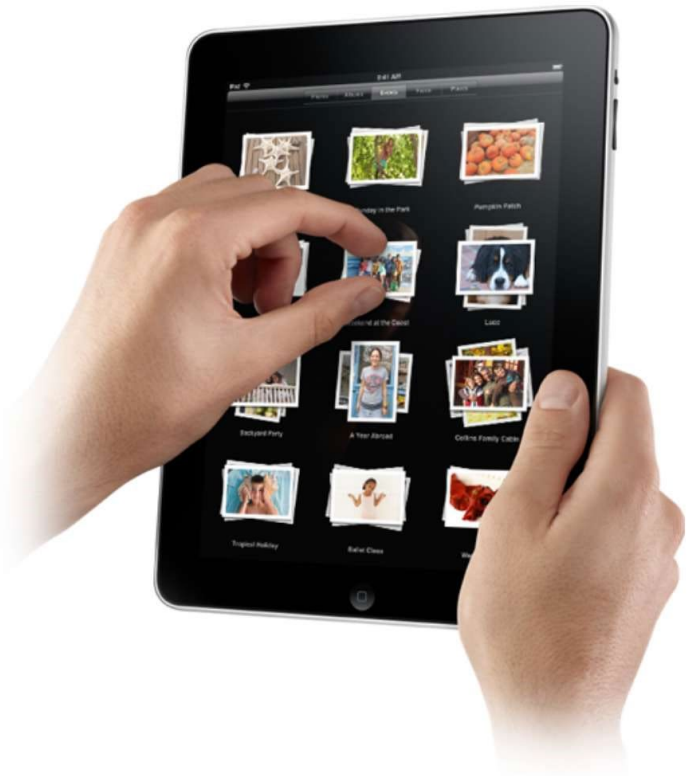
显示器s

- LCD 液晶显示器: 图像元素（像素）
  - 将帧缓冲中的图像在屏幕上浮现出来

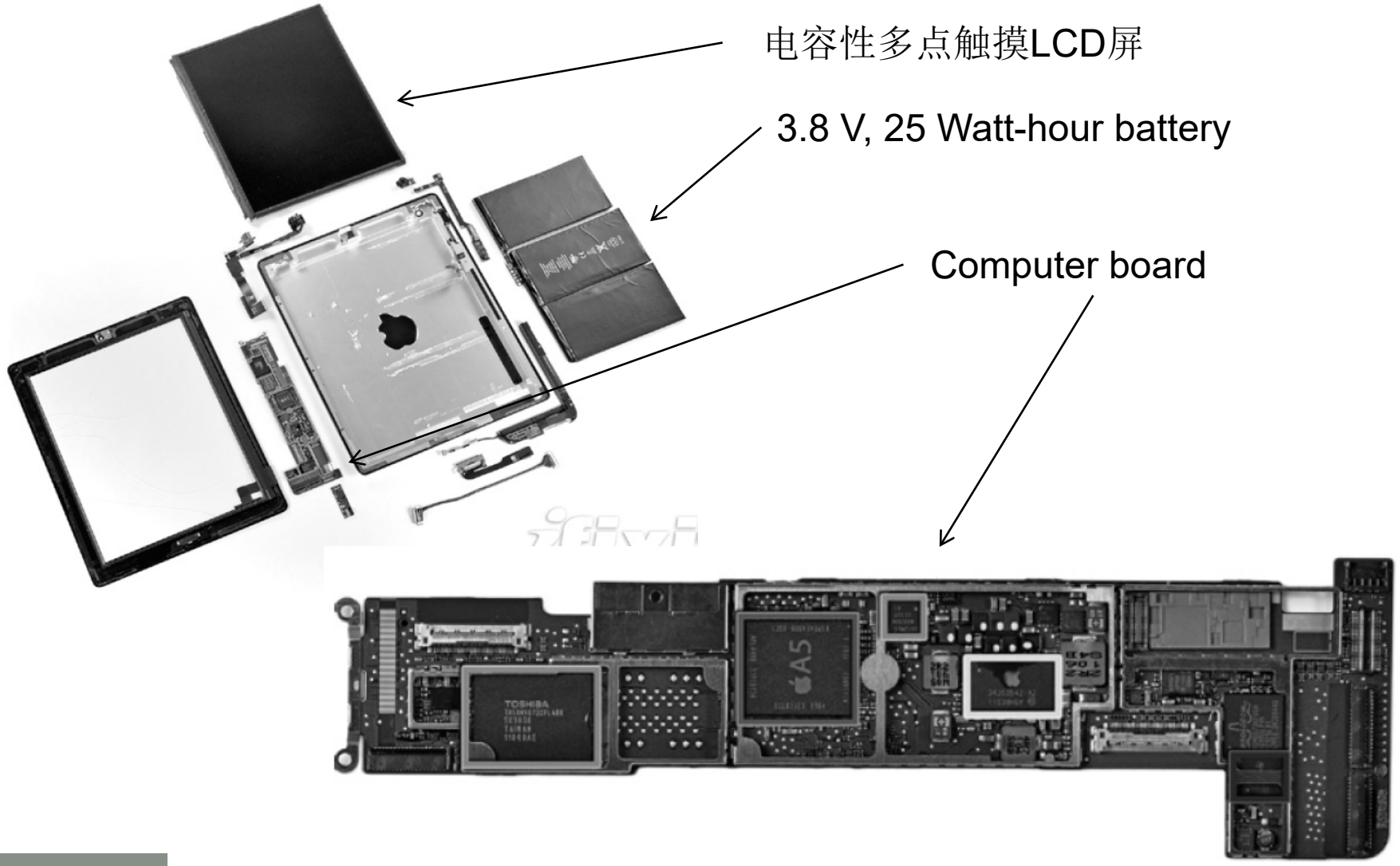


触摸屏s

- 后PC时代设备
- 替代了键盘和鼠标
- 通常采用电阻感应和电容感应
  - 许多平板和智能电话采用电容感应实现
  - 电容感应技术允许同时接触多个点



打开机箱s

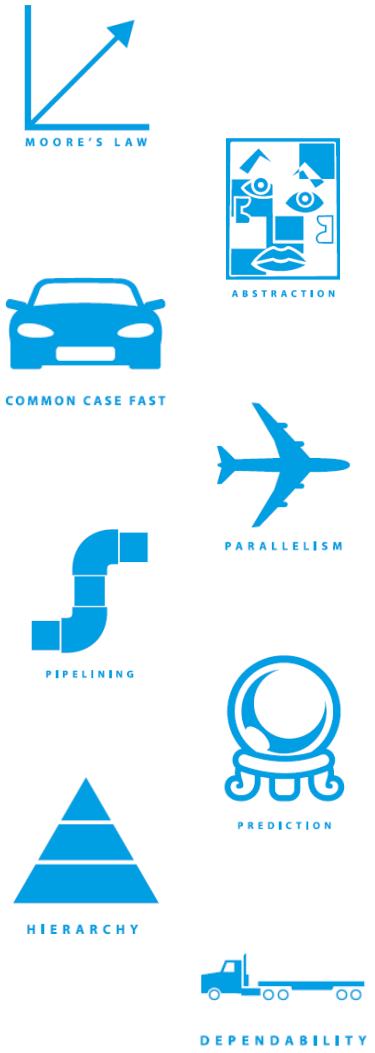


理解程序性能

- 算法
  - 决定源代码的数量和执行I/O操作的数量
- 程序语言，编译器和体系结构
  - 决定每行源代码对应机器指令的数量
- 处理器和存储器系统
  - 决定指令执行的速度快慢
- I/O系统（硬件和操作系统）
  - 决定I/O操作执行的速度快慢

计算机系统结构中的8个伟大思想！

- 面向摩尔定律的设计（前瞻性）
- 使用抽象简化设计
- 加速大概率事件
- 通过并行提高性能
- 通过流水线提高性能
- 通过预测提高性能
- 存储器层次
- 通过冗余提高可靠性



处理器（CPU）内部构成

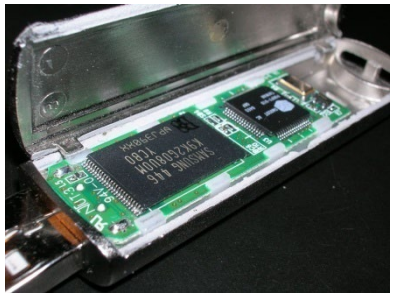
- 数据通路: 负责完成算术运算
- 控制器: 负责指导数据通路、存储器和I/O设备按照程序的指令正确执行
- 高速缓冲存储器Cache
  - 快速访问数据的小而快的SRAM存储器
- CPU内部构成对程序设计有什么影响？

抽象Abstractions

- 抽象便于构建复杂系统
  - 掩盖底层计算机系统细节
- 指令集体系结构（ISA）
  - 硬件和底层软件之间的接口
- 应用二进制接口（ABI）
  - 基本指令集和操作系统接口
- 实现
  - 相关的细节和接口，遵循体系结构抽象的硬件

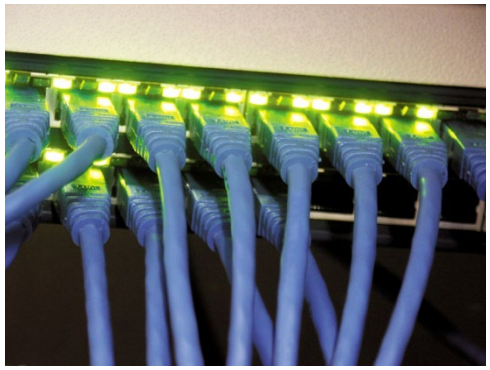
数据安全（易失性）s

- 易失性主存储器
  - 断电时会丢掉所有数据(DRAM、SRAM)
- 非易失性二级存储器
  - 磁盘
  - 闪存
  - 光盘 (CDROM, DVD)



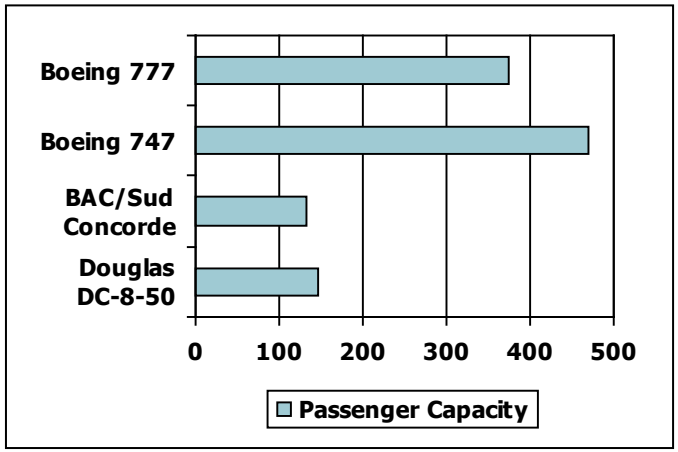
网络s

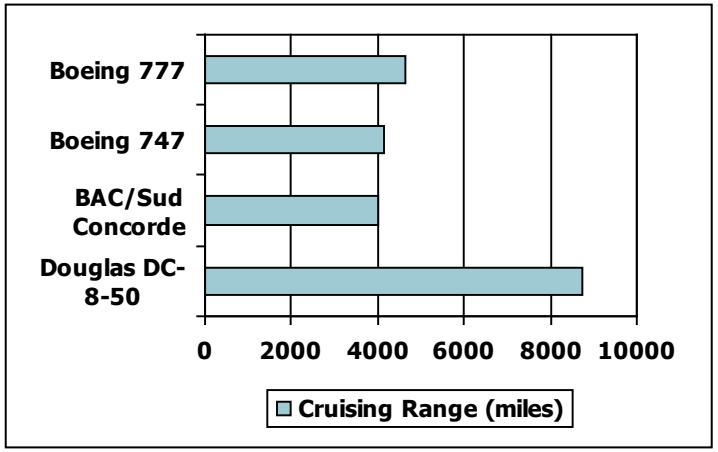
- 通信， 资源共享，远距离访问
- 局域网 (LAN): 以太网Ethernet
  - 同一个建筑物内
- 广域网 (WAN）: the Internet
- 无线网: WiFi, Bluetooth

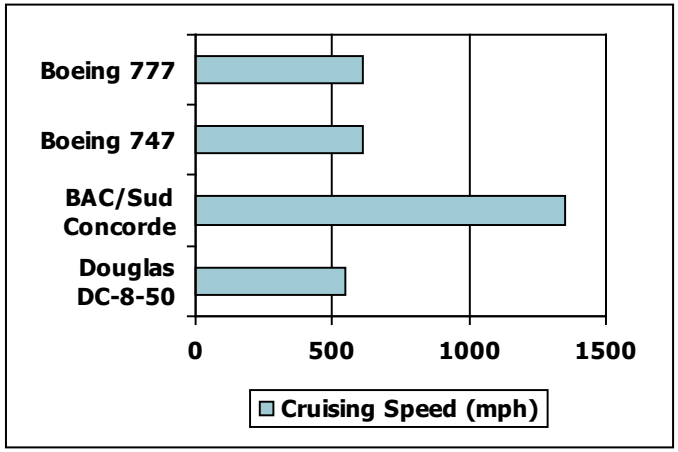


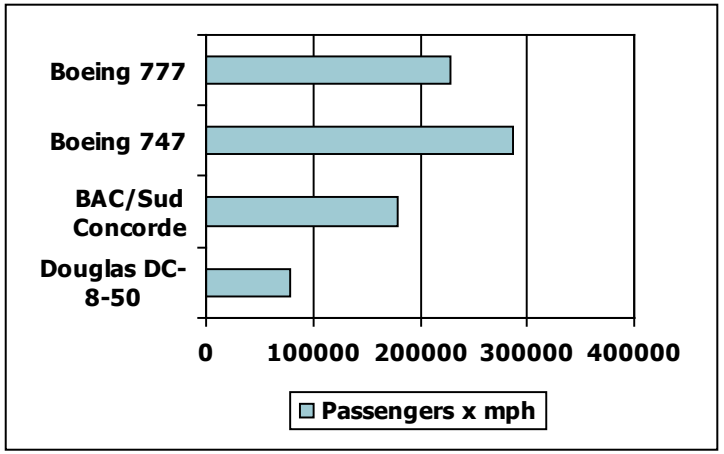
1.6性能的定义

- 哪个客机的性能最好？性能的定义存在多种可能









响应时间和吞吐率

- 响应时间（执行时间）
  - 完成某任务需要的总时间，包括硬盘访问、内存访问、I/O活动、操作系统开销和CPU执行时间等。
- 吞吐率（带宽）
  - 单位时间内完成的任务数量
    - 例如，任务数，交易数… /小时
- 响应时间和吞吐率受哪些因素影响？（例题）
  - 更换更高型号的处理器？
  - 增加多个处理器来分别处理独立的任务?
- 现在我们把重点放在响应时间上…

相对性能(例题)

- 定义：性能 = 1/执行时间
- “X是Y的 n 倍快”

$$\text{性能}_X / \text{性能}_Y = \text{执行时间}_Y / \text{执行时间}_X = n$$

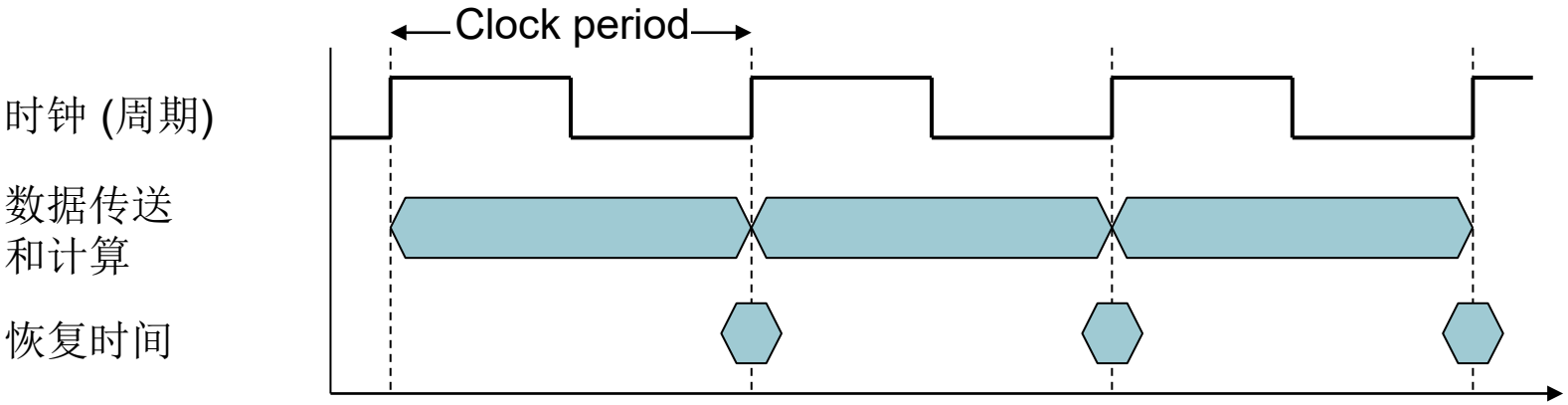
- 举例: 运行一个程序
  - 计算机A需要10s, B需要15s
  - 执行时间$_B$ / 执行时间$_A$= 15s / 10s = 1.5
  - 因此 A 是 B 的1.5倍快。

1.6.2性能的度量

- 响应时间（执行时间、墙上时钟时间、消逝时间）
  - 总的响应时间，包括
    - 处理时间, I/O操作, OS开销, 空闲时间等
  - 决定了系统的性能
- CPU 时间（CPU执行时间）
  - 程序本身花费的时间
    - 不包括等待I/O或者运行其他程序的时间
  - 包括运行用户程序的时间和操作系统为用户服务花去的CPU时间（用户CPU时间/系统CPU时间）
    - CPU和系统性能对不同的程序有不同的影响。

CPU时钟

- 主时钟不断产生固定频率的时钟



  - 时钟周期：一个时钟持续的时间
    - 例如：250ps = 0.25ns = 250×10^{-12}s
  - 时钟频率: 每秒包含的时钟周期数
    - 例如：4.0GHz = 4000MHz = 4.0×10^{9}Hz

1.6.3CPU性能及因素

某段程序的CPU时间：

$$\text{CPU时间} = \text{CPU时钟周期数} \times \text{时钟周期} = \frac{\text{CPU时钟周期数}}{\text{时钟频率}}$$

- 计算机性能改进：
  - 减少时钟周期数量
  - 增加时钟效率？
  - 硬件设计者需要权衡时钟频率和时钟周期数量

CPU 时间举例

- 计算机 A: 2GHz 时钟频率, 10s CPU 时间
- 设计计算机 B
  - 要求6s CPU 时间
  - 可以提高时钟频率，但时钟周期数将变成A的1.2倍
- 计算机B的时钟频率将提高到多少？

$$\text{时钟频率}_B = \frac{\text{时钟周期数}_B}{\text{CPU 时间}_B} = \frac{1.2 \times \text{时钟周期数}_A}{6\text{s}}$$

$$\text{时钟周期数}_A = \text{CPU 时间}_A \times \text{时钟频率}_A$$

$$= 10\text{s} \times 2\text{GHz} = 20 \times 10^9$$

$$\text{时钟频率}_B = \frac{1.2 \times 20 \times 10^9}{6\text{s}} = \frac{24 \times 10^9}{6\text{s}} = 4\text{GHz}$$

指令数和每条指令的平均时钟周期数:CPI

CPI: clock cycle per instruction

$$\text{时钟周期数} = \text{程序的指令数} \times \text{每条指令的平均时钟周期数CPI}$$

$$\text{CPU时间} = \text{程序的指令数} \times \text{CPI} \times \text{时钟周期}$$

$$= \frac{\text{程序的指令数} \times \text{CPI}}{\text{时钟频率}}$$

- 一个程序的指令数
  - 由程序自身，指令集体系结构和编译器决定
- 每条指令的平均时钟周期数
  - 由CPU硬件决定
  - 如果不同的指令有不同的CPI
    - 平均CPI受全部指令影响

CPI 举例

- 计算机A: 时钟周期 = 250ps, CPI = 2.0
- 计算机B: 时钟周期 = 500ps, CPI = 1.2
- 相同的指令集体系结构
- 那台计算机执行速度更快？快多少？

$$\text{CPU时间}_A = \text{指令数} \times \text{CPI}_A \times \text{时钟周期}_A$$
$$= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps}$$

A 更快...

$$\text{CPU时间}_B = \text{指令数} \times \text{CPI}_B \times \text{时钟周期}_B$$
$$= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}$$

$$\frac{\text{CPU时间}_B}{\text{CPU时间}_A} = \frac{I \times 600\text{ps}}{I \times 500\text{ps}} = 1.2$$

...快1.2倍

CPI 的更多细节

- 如果不同指令类型需要不同数量的时钟周期

$$\text{CPU的时钟周期数} = \sum_{i=1}^{n} (\text{CPI}_i \times \text{指令数}_i)$$

- 加权平均Weighted average CPI

$$\text{CPI} = \frac{\text{CPU时钟周期数}}{\text{指令数}} = \sum_{i=1}^{n} \left(\text{CPI}_i \times \underbrace{\frac{\text{指令数}_i}{\text{指令数}}}_{\text{相对频率}} \right)$$

CPI 举例

- 两个代码序列，对某高级语言，不同编译代码序列使用指令集A, B, C的指令数如下：

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

- 序列 1: IC = 5
  - 时钟周期数
    = 2×1 + 1×2 + 2×3
    = 10
  - Avg. CPI = 10/5 = 2.0
- 序列 2: IC = 6
  - 时钟周期数
    = 4×1 + 1×2 + 1×3
    = 9
  - Avg. CPI = 9/6 = 1.5

性能总结

The BIG Picture

$$\text{CPU 时间} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

- 性能由如下因素决定
  - 算法: 影响指令数 IC, 也可能影响CPI
  - 编程语言: 影响 IC, CPI
  - 编译程序: 影响 IC, CPI
  - 指令集体系结构: 影响 IC, CPI, 时钟周期T_c

陷阱: 用性能公式的一个子集去度量性能

- MIPS: 每秒百万条指令
  - 不能表示
    - 不同计算机指令集体系结构的差异
    - 不同指令之间复杂性的差异

$$\text{MIPS} = \frac{\text{指令条数}}{\text{执行时间} \times 10^6} = \frac{\text{指令条数}}{\frac{\text{指令条数} \times \text{CPI}}{\text{时钟周期}} \times 10^6} = \frac{\text{时钟周期}}{\text{CPI} \times 10^6}$$

  - 一台计算机的不同程序中，CPI可以变化

总结

- 计算机成本/性能将会不断改进
  - 归因于技术的发展
- 抽象的层次结构
  - 同时体现在硬件和软件之间
- 指令集体系结构
  - 硬件和软件的接口
- 执行时间：最好的性能度量方法
- 功耗是一个限定因素
  - 可以采用并行技术改进性能

- 8个伟大思想
- 软硬件的层次
- 冯.诺依曼 计算机的逻辑构成
- 性能