



参考教材:

1. 计算机系统结构教程(第2版) 张晨曦 清华大学出版社
2. **Computer Architecture – A Quantitative Approach (Fourth Edition or later)** John L.Hennessy, David A.Patterson, Morgan Kaufmann Publish, 2007-2012
3. 计算机系统结构(第三版) 白中英 科学出版社

- 研究内容:

从系统外部研究使用者所关心到的物理计算机的抽象、软硬件功能分配及分界面的确定。

系统结构的相关概念

流水线技术

向量处理机

指令级并行

互连网络

多处理机（多核*）

阵列处理机

机群系统*

- 学习目的:

建立计算机系统的完整概念、学习计算机系统的分析方法和设计方法、掌握新型计算机系统的基本结构及其工作原理

计算机是全国重点学科，是如何设置的？

- 一级学科：计算机科学与技术
- 二级学科：计算机系统结构、*计算机软件与理论、计算机应用技术*

本课程所要具备的知识：

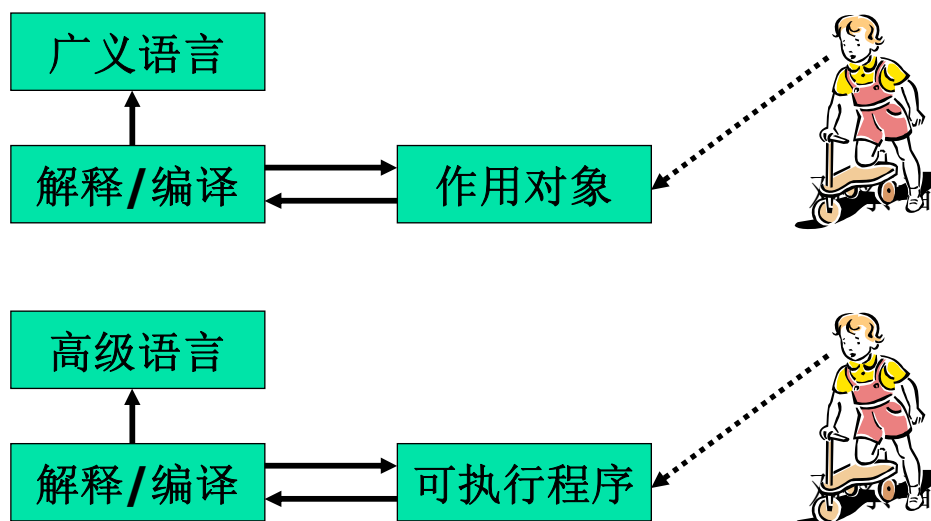
计算机组成原理、计算机操作系统、汇编语言、
数据结构、微机原理、高级语言

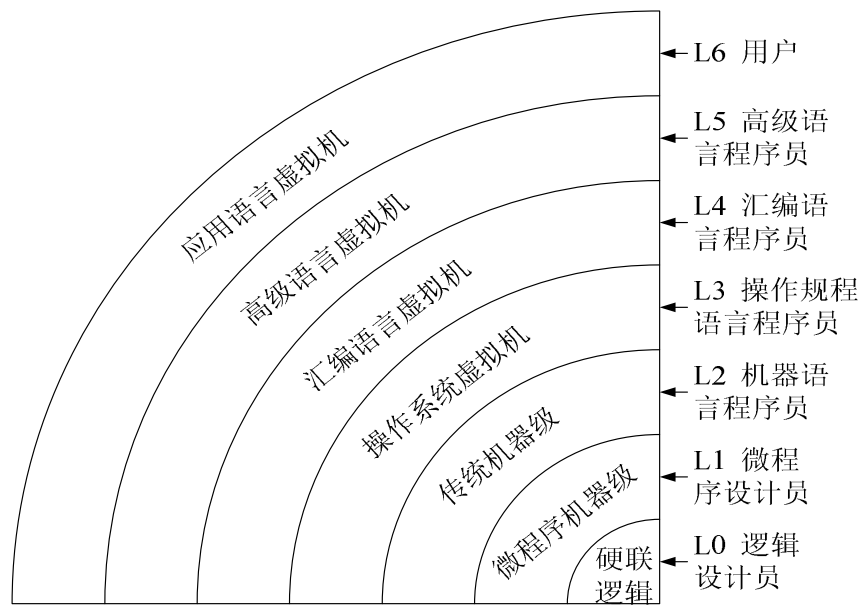
本课程教学形式、考核方式和考核结构

Concepts

虚拟机(Virtual Machine)概念

虚拟机是一个抽象的计算机，由软件实现，并与实际机器一样，都具有指令集并使用不同的存储区域





L0和L1级属于计算机组织与实现。

什么是计算机系统结构？

计算机体系结构的定义之一：

The attributes of a [computing] system as seen by the programmer, i.e.. The conceptual structure and functional behavior, as distinct from the organization of the data flows and controls the logic design, and the physical implementation. Amdahl, Blaaw, and Brooks, 1964 (IBM 360)

对计算机系统而言，是指那些由程序员可见的系统属性。

所指的程序员：

汇编语言、机器语言、编译程序、操作系统

程序员所看到的：

为了程序能在机器上正确运行所必须了解到的内容

- **数据表示**：硬件能够直接识别和处理的数据类型和格式
- **寻址方式**：最小寻址单位、寻址方式的种类和地址运算等
- **寄存器组织**：操作数寄存器、变址寄存器、控制寄存器及专用寄存器的定义、数量和使用规则等
- **指令系统**：机器指令的操作类型、格式，指令间的排序和控制机制等
- **中断系统**：中断类型、中断级别和中断响应方式等
- **存储系统**：最小编址单位、编址方式、主存容量、最大寻址空间等
- **处理机工作状态**：定义和切换方式，如管态和目态等
- **输入输出系统**：连接方式、数据交换方式、数据交换过程的控制等
- **信息保护**：包括信息保护方式和硬件对信息保护的支持等

本来存在的事物或属性，从某种角度看似乎不存在

浮点数表示、乘法指令

对高级语言程序员、应用程序员 透明

对汇编语言程序员、机器语言程序员 不透明

数据总线宽度、微程序

对汇编语言程序员、机器语言程序员 透明

对硬件设计者、计算机维修人员 不透明

计算机体系结构的定义之二：

计算机系统的软、硬件的界面，即机器语言程序员所看到的传统机器级所具有的属性

—— 计算机系统结构主要研究软硬件功能分配和对软硬件界面的确定

- 计算机系统由软件、硬件和固件组成，它们在功能上是同等的
- 同一种功能可以用硬件实现，也可以用软件或固件实现
- 不同的组成只是性能和价格不同

计算机组成：指计算机系统结构的逻辑实现

主要包括：

- 确定数据通路的宽度
- 确定各种操作对功能部件的共享程度
- 确定专用的功能部件
- 确定功能部件的并行度
- 设计缓冲和排队策略
- 设计控制机构
- 确定采用何种可靠性技术

计算机实现：指计算机组成的物理实现

- 处理机、主存储器等部件的物理结构
- 器件的集成度和速度
- 专用器件的设计
- 器件、模块、插件、底版的划分与连接
- 信号传输技术
- 电源、冷却及装配技术，相关制造工艺及技术等

Are you clear ?

- The difference between **Architecture** with **Organization**
- refer to the operational units and their interconnections that realize the architecture specifications.

Example?

- 计算机系统结构、计算机组成和计算机实现是三个不同的概念，但随着技术、器件和应用的发展，三者之间的界限越来越模糊。

常见的分类方法：

1. 按大小 — 巨型机、大型机、中型机、小型机、微型机等

以性能为表征，按价格划分

划分的标准随时间变化，随时间推移会降低等级

设计方法：

最高性能

特殊用途

最佳性能价格比

一般商用计算机

最低价格

家用计算机

2. 按用途 — 科学计算、事务处理、实时控制、工作站、服务器、家用计算机，以及特殊应用的：**嵌入式处理机**、单片机等
3. 按数据类型 — 定点机、浮点机、向量机、堆栈机等
4. 按处理机个数和种类 — 单处理机、并行处理机、多处理机、分布处理机、关联处理机、超标量处理机、超流水线处理机、SMP（对称多处理机）、MPP（大规模并行处理机）、机群（Cluster）系统等
5. 按所使用的器件 — 第一代电子管（Valve）、第二代晶体管（Transistor）、第三代集成电路（LSI）、第四代大规模集成电路（VLSI）、第五代（智能计算机）等

Flynn（佛林）分类法：

按照指令流和数据流的多倍性特征对计算机系统进行分类

指令流： 机器执行的指令序列

数据流： 由指令流调用的数据序列，含输入数据和中间结果

多倍性（multiplicity）： 在系统性能瓶颈部件上同时处于同一执行阶段的指令或数据的最大可能个数

单指令流单数据流 **SISD**(single instruction stream single data stream)

单指令流多数据流 **SIMD**(single instruction stream multiple data stream)

多指令流单数据流 **MISD**(multiple instruction stream single data stream)

多指令流多数据流 **MIMD**(multiple instruction stream multiple data stream)



是否都存在？分类是否有缺陷？

按并行度的分类方法

Handler分类法

根据并行度和流水线分类，计算机的硬件结构分成三个层次，并分别考虑它们的可并行性和流水处理程度

PCU 程序控制部件的个数k

ALU 算术逻辑部件 (或运算部件PE)的个数d

ELC 每个算数逻辑部件包含基本逻辑线路电路的套数 ω

t(系统型号) = (k, d, ω) 或

$$t(\text{系统型号}) = (k \times k', d \times d', \omega \times \omega')$$

k':宏流水线中程序控制部件的个数

d':指令流水线中算数逻辑部件的个数

ω' :操作流水线中基本逻辑线路的套数

例: Cray-1

1个CPU； 12个ALU/PE部件，可最多实现8级流水；字长64位，可实现1~14位流水线处理；

t(Cray-1) = [1, 12 x 8, 64 x (1~14)]

冯氏分类法 (冯云泽, 1972)

计算机系统在单位时间内能够处理的最大二进制位数（字/位的串/并）

Embedded Immanence



- 设计Intel 4004 的最初目的是为了应用于计算器 —— 一个嵌入式应用
- 当今的处理器中有95%是用于嵌入式应用
- 通常面向特殊应用

Microcontrollers

Media Processors

Network & Communication Processors

DSPs

Graphics Processors

...

What is Embedded Microprocessor?

A programmable processor whose programming interface is not accessible to end-user of the product.

The only user-interaction is through the actual application.



Make the Common Case Fast

In making a design trade-off, favor the frequent case over the infrequent case.

大概率事件优先原理

对于大概率事件(最常见的事件), 赋予它优先的处理权和资源使用权, 以获得全局的最优结果。

—— 系统结构设计中最常用的思想方法

❖ 大概率事件往往比小概率事件简单?

Example?

- The performance improvement to be gained from using some faster mode of execution is limited by the fraction of the time the faster mode can be used.

系统中某一部件由于采用某种更快的执行方式后整个系统性能的提高与这种执行方式的使用频率或占总执行时间的比例有关。

$$\text{系统加速比} = \frac{\text{改进后系统性能}}{\text{改进前系统性能}} = \frac{\text{改进前总执行时间}}{\text{改进后总执行时间}}$$

$$Fe = \frac{\text{可改进部分占用的时间}}{\text{改进前整个任务的执行时间}} < 1 \quad Se = \frac{\text{改进前改进部分的执行时间}}{\text{改进后改进部分的执行时间}} > 1$$

$$\text{改进后整个任务的执行时间为: } T_n = T_o \cdot (1 - Fe + \frac{Fe}{Se})$$

其中 T_o 为改进前的整个任务的执行时间。

未改进部分

改进后

Amdahl定律

改进后整个系统的加速比为：
$$S_n = \frac{T_o}{T_n} = \frac{1}{(1 - Fe) + \frac{Fe}{Se}}$$

$Se \rightarrow \infty$ 时，则 $S_n = \frac{1}{1 - Fe}$ ，可改近极限受 Fe 的约束。

例：将某一部件的处理速度加快10倍，该部件的原处理时间占整个运行时间的40%，整个系统的性能提高多少？

解： $Fe=0.4$, $Se=10$ ，根据Amdahl定律，

$$S_n = \frac{1}{0.6 + \frac{0.4}{10}} = \frac{1}{0.64} \approx 1.56$$

CPU性能公式

一个程序所花的CPU时间 $T_{CPU} = \overset{\text{周期数}}{N_C} \times \overset{\text{周期}}{t} = \frac{N_C}{f}$

其中 N_C 表示CPU时钟周期数

每条指令的平均时钟周期数CPI $CPI = \frac{N_C}{I_N}$ ↑ 指令数

$$\therefore T_{CPU} = \underbrace{I_N \times CPI}_{N_C = CPI \times I_N} \times t = \frac{(I_N \times CPI)}{f}$$

IPC=1/CPI

• 结论：CPU的性能与3个要素有关：

①时钟频率 f ；② 每条指令所花的时钟周期数 N_C ；③ 指令条数 I_N 。其中时钟频率取决于硬件技术和组织，CPI (Cycles Per Instruction)取决于系统结构组织和指令集，指令数目取决于系统结构的指令集和编译技术。

计算程序的CPU时钟周期总数： $N_c = \sum_{i=1}^n (CPI_i \times IC_i)$

其中 IC_i 为i指令在程序中执行的次数； CPI_i 为i指令平均时钟周期数； n 为指令种类数。

因而有：

$$T_{CPU} = \left[\sum_{i=1}^n (CPI_i \times IC_i) \right] \times t$$

$$CPI = \sum_{i=1}^n (CPI_i \times IC_i) / IC = \sum_{i=1}^n \left(CPI_i \times \frac{IC_i}{IC} \right)$$

其中 IC_i/IC 为i指令在程序中所占的比例。

- Determine what attributes are important for a new machine, then design a machine to maximize performance while staying within cost and power constraints.
- 确定用户对计算机系统的功能、价格和性能需求
 - 应用领域
 - 软件兼容
 - OS要求
 - 标准
- 软硬件功能分配
- 生命周期

冯·诺依曼（**Van Neumann**）结构 — 1936～46年形成，
冯·诺依曼等人于1946年提出

特点：存储程序，运算器为中心，集中控制

- 存储器是字长固定的、顺序线性编址的一维结构
- 存储器提供可按地址访问的一级地址空间，每个地址是唯一定义的
- 由指令形式的低级机器语言驱动。
- 指令顺序执行，即一般按照指令在存储器中存放的顺序执行，程序分支由转移指令实现。
- 运算器为中心，输入输出设备与存储器之间的数据传送都途经运算器。运算器、存储器、输入输出设备的操作以及它们之间的联系都由控制器集中控制。

改进：存储程序，存储器为中心，分散控制 ✨

非冯计算机

- 数据驱动型，出现了数据流机计算机
- 需求驱动型，出现各种图归约计算机
- 处理非数值化信息的智能计算机，例如自然语言、声音、图形和图象处理、虚拟现实处理等。
- 第五代计算机，由推理机和知识库机等组成
- 神经网络计算机，脉动式计算机，仿生计算机，.....



系统结构发展的影响因素

软件对系统结构的影响 – 移植（优化）

统一的高级语言 – 不同计算机能采用同一种高级语言（如 JAVA）。

系列机 - 具有相同的系统结构，但具有不同组成和实现的一系列不同型号的机器（不同时间/不同档次）

向上(下)兼容 - 按某档机器编写的程序，不加修改就能运行于比它高(低)档的机器。

向前(后)兼容 - 某个时期投入市场的某种型号机器编写的程序，不加修改就能运行于在它之前(后)投入市场的机器。

模拟和仿真 - 在不同系统结构的机器之间实现软件移植

模拟 - 用软件方法在一台现有的计算机上实现另一台计算机的指令系统。

仿真 - 微程序直接解释另一种机器指令系统的方法



系统结构发展的影响因素

应用对系统结构的影响

应用需求是促使计算机系统结构发展的最根本动力。

计算密集、存储密集、I/O密集、实时、人工智能等

器件对系统结构的影响

摩尔定律

问题:

设计Cache机制的原因和依据是什么?

Programs tend to reuse data and instructions they have used recently.

程序访存的局部性

- 实验统计表明: 一个程序用90%的执行时间去执行仅占10%的程序代码。

数据访问的局部性

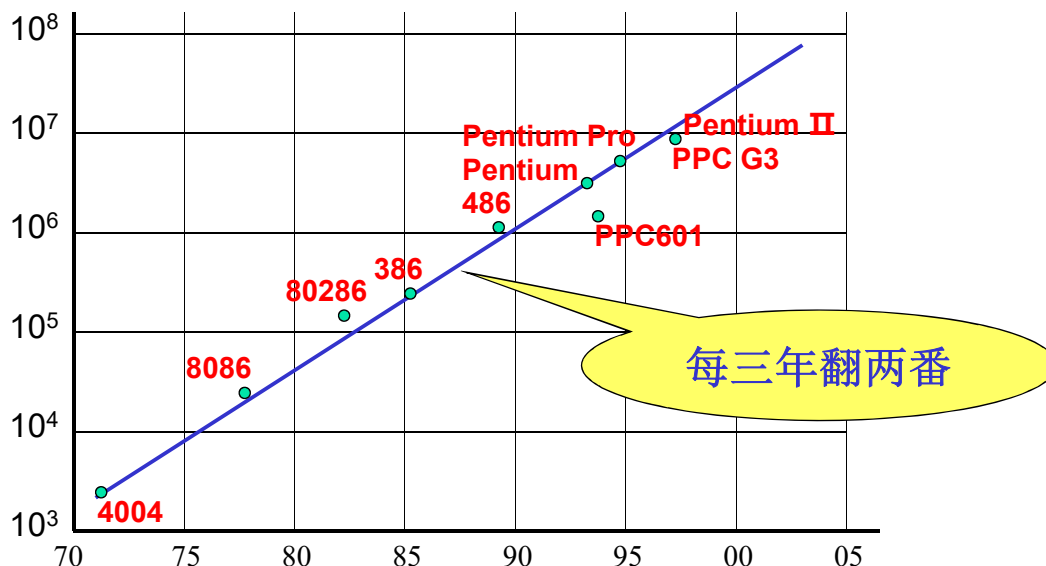
- 空间上的局部性(Spatial locality)
- 时间上的局部性(Temporal locality)

如何利用?

计算机性能的发展——摩尔定律

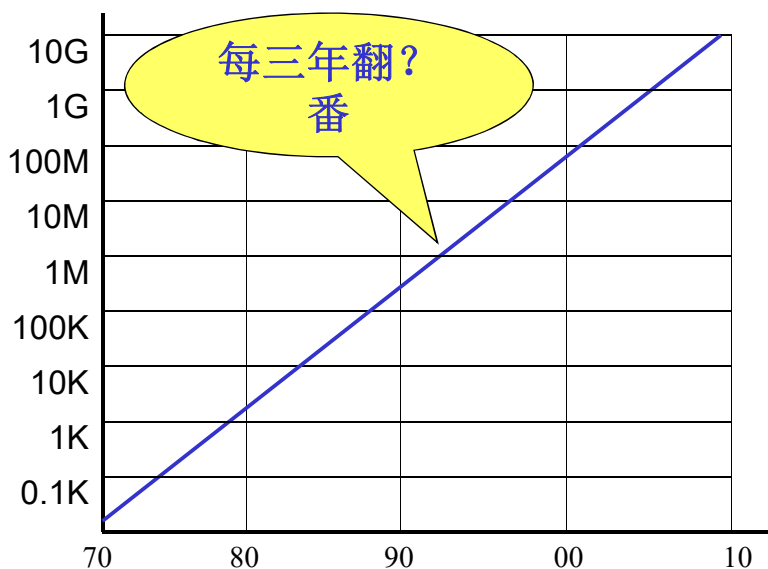
高登·摩尔(Gordon Moore) —— Intel的创始人之一

断言芯片的晶体管数量每年翻一番，而且这种态势将在不久的将来仍继续下去（1965年）



摩尔定律

存储器容量：60%/year

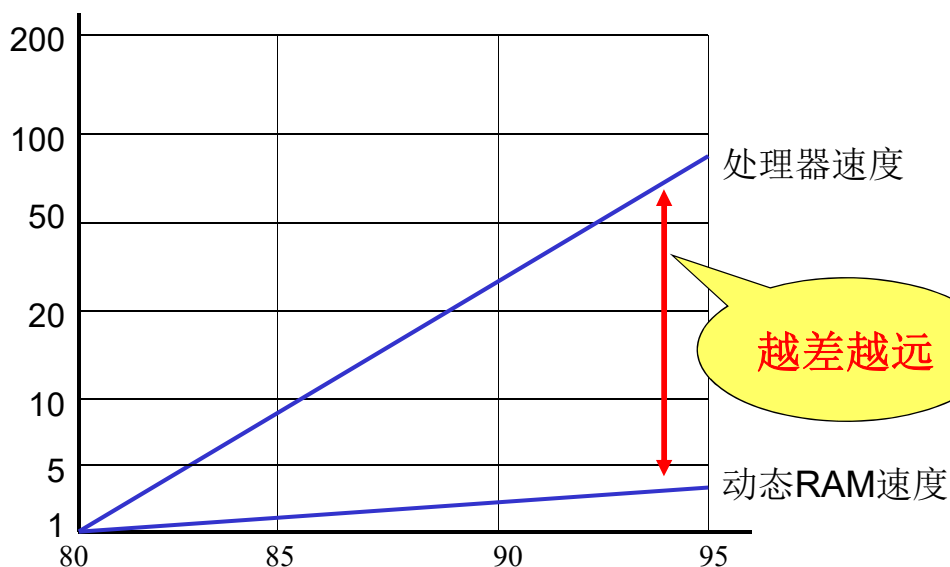


year	Mb
1980	.0625
1983	.25
1986	1
1989	4
1992	16
1996	64
2000	256

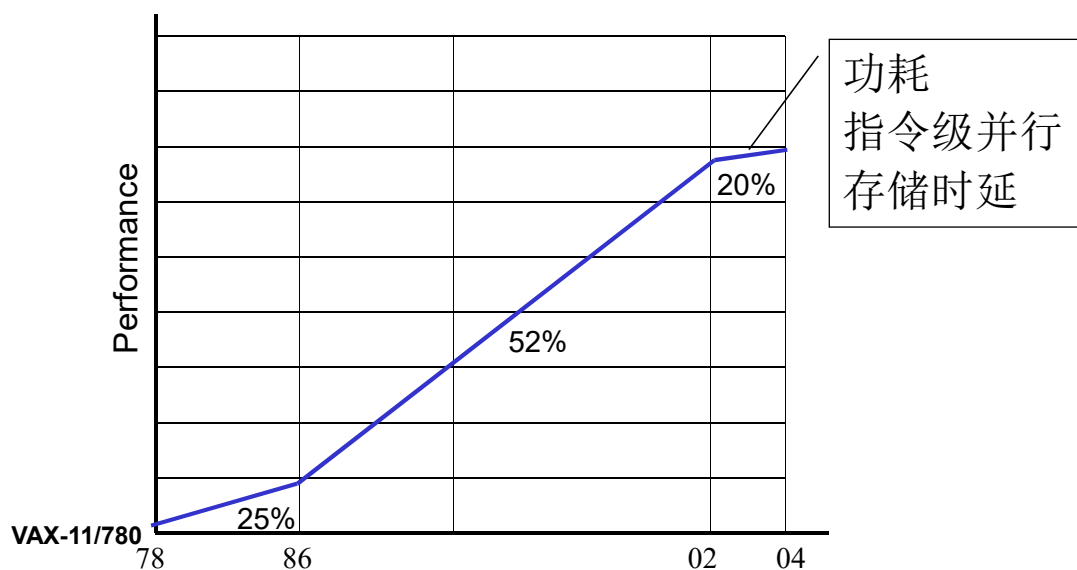
技术工艺发展趋势：

项目	速度(响应时间)
Logic	每三年两倍
DRAM	每十年两倍
DISK	每十年两倍

这中间会出什么问题？



当处理器的性能以惊人的速度向前发展的时候，其他部件（包括I/O系统）的速度没有跟上——寻求性能平衡的需要：调整组织与结构



性能评价的一个重点问题是对性能**量度**的选择。
对系统设计者来说，执行时间和吞吐量是两个重要的性能量度值。

1. 响应时间（response time），也可称为**执行时间（execution time）**，即事件从发生到结束所花费的时间。

“甲机器比乙机器快n倍”的含义应该是：

$$\frac{\text{机器性能（甲）}}{\text{机器性能（乙）}} = \frac{\text{执行时间（乙）}}{\text{执行时间（甲）}} = n$$

2. 对用于事务处理的计算机系统进行性能评价时，有一个常用的词是“**吞吐量 (throughput)**”。如果我们说“甲的吞吐量是乙的2倍”，那么则意味着在每个时间单元内，甲完成任务的数量是乙的2倍。

通常用每分钟的事务处理量 (**Transactions Per Minute - tpm**) 来表述事务处理系统的性能。

CPU性能与系统性能

$$T_{CPU} = I_N \times CPI \times t$$

$$= \frac{N_c}{t} \times t$$

↑
总时钟周期数

每个指令平均时钟周期数

1. 时钟频率 (主频)：一般用于同类处理机之间

如：Pentium III/800比Pentium III /500快60%

2. 指令执行速度：一种经典表示方法

MIPS(Million Instructions Per Second), GIPS、TIPS

$$MIPS = \frac{\text{指令条数}}{\text{执行时间} \times 10^6} = \frac{F_z}{CPI} = IPC \times F_z$$

其中：Fz为处理机的工作主频。

如：Pentium II 450处理机的IPC=2 (或CPI=0.5)，
Fz=450MHz

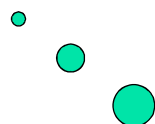
$$MIPS_{Pentium II 450} = IPC \times F_z = 2 \times 450 = 900 MIPS$$

$$CPI = \frac{1}{IPC}$$

↑ 每个时钟周期执行的指令数
每个指令平均时钟周期数

MFLOPS(Million Floating Point Operations Per Second),
每秒百万次浮点运算。(GFLOPS、TFLOPS)

$$MFLOPS = \frac{\text{浮点运算次数}}{\text{执行时间} \times 10^6}$$



MIPS和MFLOPS的不足之处?

3. 基准测试程序 (benchmark) :

把应用程序中用得最多频繁的那部分核心程序作为评价计算机性能的标准程序。称为基准测试程序。

1. 核心测试程序 – 从真实程序中选出的关键代码段;
2. 小测试程序 – 简单的只有几十行的小程序, 也可以是测试者自编的针对某些特定指标测试小程序;
3. 合成的测试程序 – 人工合成, 一般基于大量应用程序的操作统计比例;

在1988年以前就出现了许多被广泛使用的小型基准测试程序, 如Dhrystone、Whetstone、Linpack、Sorting等。从1988年开始, 以SPEC (Standard Performance Evaluation Cooperative) 和TPC (Transactions Processing Council) 等为代表的组织提供了一系列基准测试套件和基准测试指导方针, 大大地改进了基准测试程序的质量。

整数测试程序：Dhrystone

用C语言编写，100条语句。包括各种赋值语句、数据类型和数据区、控制语句、过程调用和参数传送、整数运算和逻辑操作。

浮点测试程序：Linpack

用FORTRAN语言编写，主要是浮点加法和浮点乘法操作。用MFLOPS、GFLOPS 及TFLOPS表示。

Whetstone基准测试程序

用FORTRAN语言编写的综合性测试程序，主要包括浮点运算、整数算术运算、功能调用、数组变址、条件转移、超越函数。结果用KMIPS表示

SPEC CPU2000

是工业标准化的CPU基准测试套件，可用于测试处理器、内存和编译器的性能。套件包括14个用C和Fortran编写的浮点程序和11个整数程序（10个C程序和1个C++程序）。在SPEC CPU2000之前有SPEC89、SPEC92、SPEC95

SPEC CPU2006

分为测试整数性能的CINT2006和测试浮点性能的CFP2006两个部分，分别包含12个测试场景和17个测试场景，均为基于应用程序的测试。

<http://www.spec.org/cpu2006/>

EEMBC系列-EEMBC (EDN Embedded Microprocessor Benchmark Consortium / **EDN**嵌入式微处理器基准测试联盟)

建立于1997年4月。**EEMBC**系列基准测试程序面向嵌入式处理器。由于联盟主要由半导体工业中的主流企业组成，因此发布的相关内容自然成为实际的工业标准。其内容涵盖汽车工业、消费电子、数字娱乐、**JAVA**、网络、办公室自动化和电信市场，应用包括引擎控制、数码相机、打印机、蜂窝电话、调制解调器等包含嵌入式处理器的系统，以数十个独立的算法程序组成的套件。

<http://www.eembc.org/>

1. **8080**: 世界上第一个通用微处理器。**8位**。曾用于第一台个人计算机Altair。
2. **8086**: **16位**。**8086** 开辟了指令高速缓存或称队列，在指令被实际执行之前能预取几条指令。变形形式是**8088**，曾用于最初的**IBMPC**机，并确保了**Intel**的成功。
3. **80286**: **8086**扩展产品，可寻址**16MB**存储区间，不再受**1MB**存储区间的限制。
4. **80386**: **Intel**的第一个**32位处理器**，一个有重大改进的产品。**32位**的结构，可以与几年前的小型机和大型机相抗衡。也是**Intel**的第一个支持多任务的处理器。
5. **80486**: 采用更为复杂、功能更强的**Cache**和指令流水线技术。内置数字协处理器，减轻了主**CPU**复杂算术运算的负担。

6. **Pentium**: Intel引入了**超标量(Superscalar)技术**, 允许更多的指令并行执行。
7. **PentiumPro**: 继续推进由Pentium开始的超标量结构, 极富进取性地使用了包括**寄存器重命名、转移预测、数据流分析、推测执行**等技术。
8. **Pentium II**: 融入了专门用于有效处理视频、音频和图形数据的Intel MMX技术。
9. **Pentium III**: 融入了新的浮点指令, 以支持三维图形软件。
10. **Pentium IV**:

	601	603/603e	604/604e	740/750(G3)	G4
发布时间	1993	1994	1994	1997	1999
时钟频率(MHz)	50-120	100-300	166-350	200-366	500
一级高速缓存	-	16KB指令	32KB	32KB	32KB
	-	16KB数据	32KB	32KB	32KB
二级高速缓存	-	-	-	256KB-1MB	256KB-1MB
晶体管数 (单位: 10万)	2.8	1.6-2.6	3.6-5.1	6.35	不详

1. 601: 快速推出601(32位机)的目的是, 将PowerPC结构尽可能快地推向市场。
2. 603: 面向低端的台式机和便携机, 32位, 与601性能相当, 但价格更低, 效率更高。
3. 604: 面向台式机和低端服务器, 32位机, 使用了非常先进的超标量设计技术, 达到更高的性能。
4. 620: 面向高端服务器。PowerPC家族中第一个实现全64位结构的成员, 包含64位的寄存器和数据路径。
5. 740/750(G3): 在主处理器芯片内集成了两级高速缓存, 与带片外高速存的系统相比, 性能显著地改善。
6. G4: 继续提高了处理器芯片的并行度和内部速度。

- 超标量处理机、超流水线处理机、向量处理机
- 并行处理机
 - ✓ SIMD
 - ✓ 并行向量处理机PVP(Parallel Vector Processor)
 - ✓ 对称多处理机SMP(Symmetric Multiprocessor)
 - ✓ 大规模并行处理机MPP(Massively Parallel Processor)
 - ✓ 工作站机群COW(Cluster of Workstation)
 - ✓ 分布共享DSM(Distributed Shared Memory)多处理机

并行处理——高等计算机的核心技术

所涉及的主要问题:

互联网络与通信

粒度划分与调度

并行存储器系统

Cache一致性问题

存储器一致性问题

指令的并行性

并行性概念：计算机系统所具有的能在同一时刻或同一时间间隔进行多种运算或操作的特性

- **同时性**(simultaneity)：两个或两个以上的事件在同一时刻发生
- **并发性**(concurrency)：两个或两个以上的事件在同一时间间隔发生

并行处理是系统结构、硬件、软件、算法、语言等多方面综合研究的领域

数据处理的并行性：

字串位串（无并行）

字串位并（初步并行）

字并位串（较高并行）

全并行（更高并行）

程序执行的并行性：

指令内部并行（各微操作之间） **指令级并行**（ILP，流水线）

线程级并行（TLP，结合多核，通常以进程内派生的多个线程为调度单位）

任务级/过程级并行（以子程序/进程为单元）

作业/程序级并行（单机系统通过软件，多机系统大多通过硬件）

提高并行性的技术途径:

- **时间重叠**(Time Interleaving) - 引入时间因素, 让多个处理过程在时间上相互错开, 轮流重叠地使用同一套硬件设备的各个部分, 以加快硬件周转而赢得速度
- **资源重复**(Resource Replication) - 引入空间因素, 以数量取胜。通过重复设置硬件资源, 大幅度地提高计算机系统的性能
- **资源共享**(Resource Sharing) - 这是一种软件方法, 它使多个任务按一定时间顺序轮流使用同一套硬件设备

- 耦合度
 - **紧耦合(Tightly Coupled)/直接耦合(Directly Coupled)**: 物理连接频带高, 总线/高速开关互联, 可共享主存;
 - **松耦合(Loosely Coupled)/间接耦合(Indirectly Coupled)**: 通道/通信线互联, 可共享外存/外设
- 同构(Homogeneous)和异构(Heterogeneous)
- 容错(Fault Tolerant)/可靠性和可重构(Reconfigurable)
- 分布式系统与多机系统?