

# 计算机组织与体系结构

## 第四讲

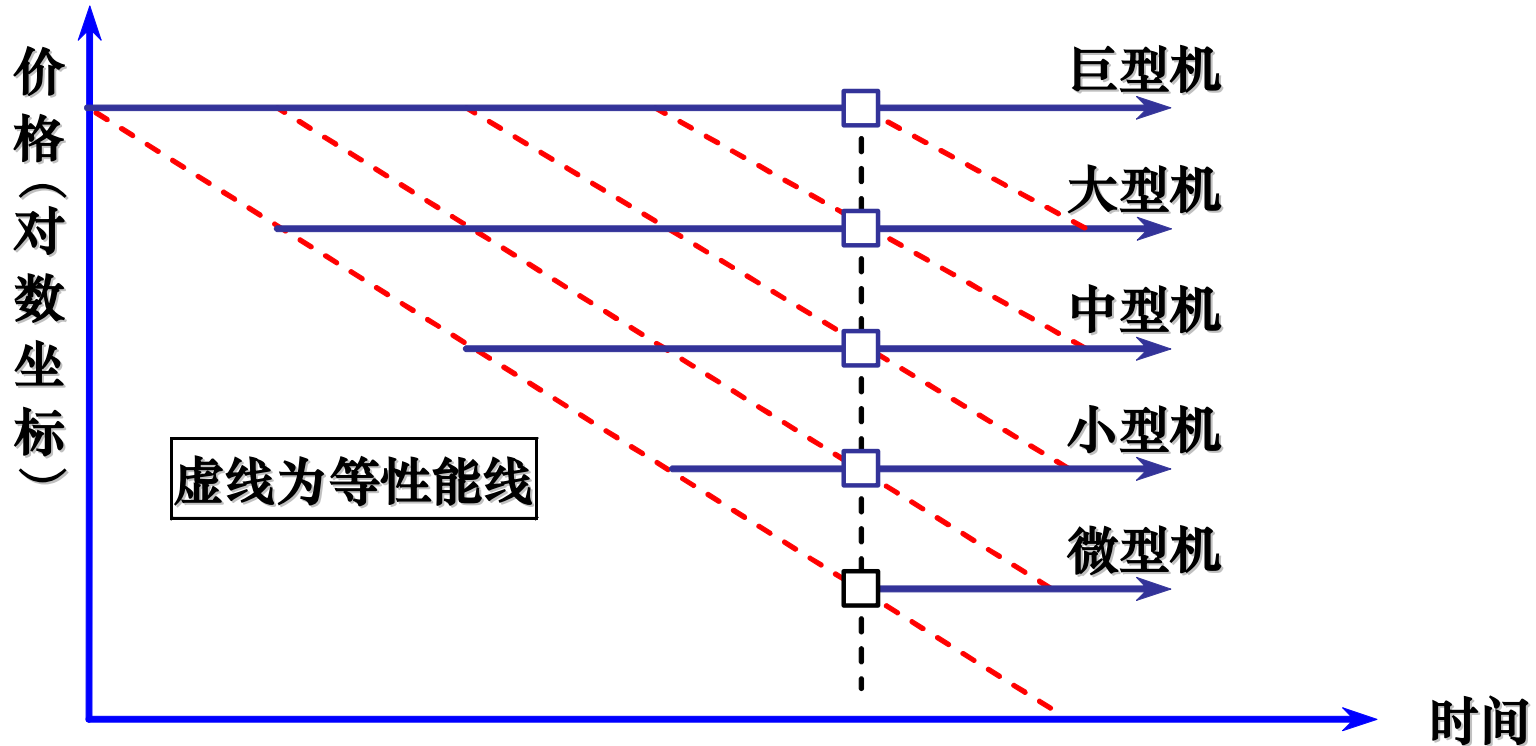
计算机科学与技术学院

舒燕君

# Recap

- 计算机的基本组成
  - ✓ 计算机的分代
  - ✓ 软件、应用的发展
  - ✓ 相关核心技术的发展
  - ✓ 体系结构的发展
- 并行体系结构的发展
  - ✓ 并行的概念、并行等级划分
  - ✓ Flynn分类
  - ✓ 提高并行性的技术途径

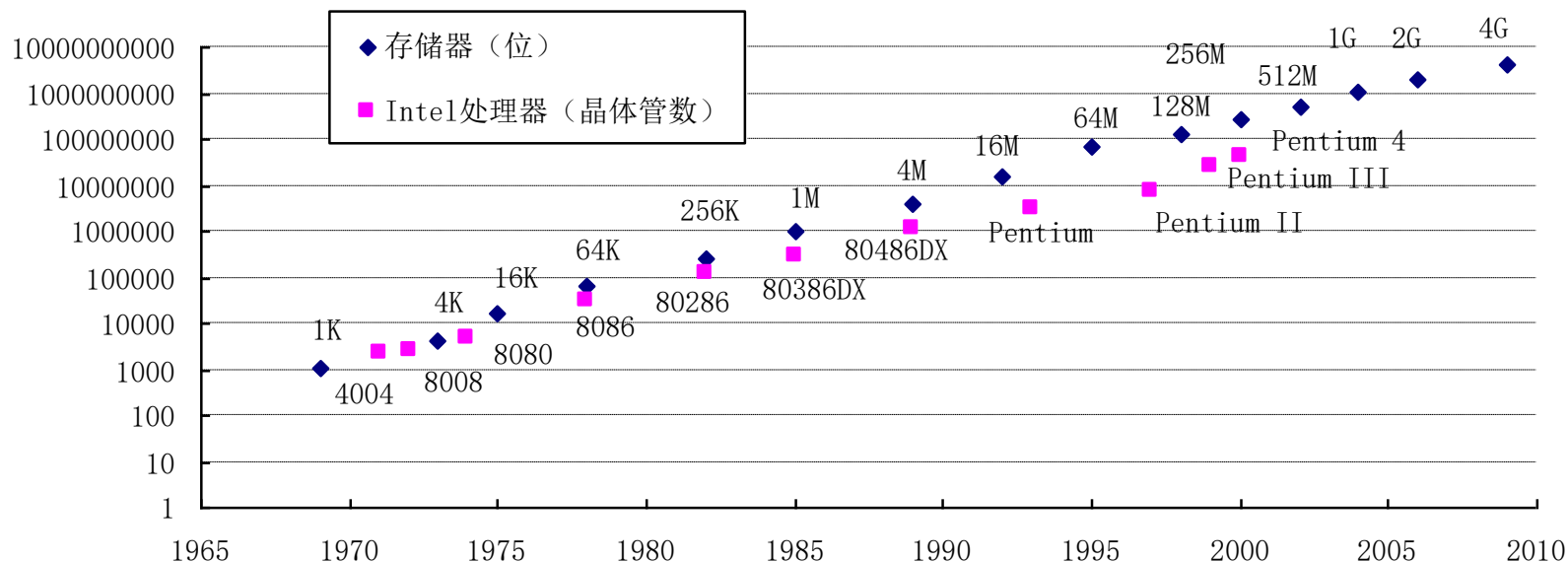
## 2.2.1 计算机的分代



**技术和性能的“下移”**。新型体系结构的设计一方面是合理地增加计算机系统中硬件的功能比例。另一方面则是通过多种途径提高计算机体系结构中的并行性。

## 2.2.4 相关核心技术的发展

- 1965年，时任仙童公司研发实验室主任的摩尔（Gordon Mooer）在《Electronics》上撰文，认为集成电路密度大约每两年翻一番
- 40年来，摩尔定律不但印证了集成电路技术的发展，也印证了计算机技术的发展



## 2.2.6 并行处理技术的发展

- 计算机系统结构的Flynn分类法
  - Flynn分类法 按照指令和数据的关系，把计算机系统的结构分为4类
    - 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)

## 2.2.6 并行处理技术的发展

### 2 提高并行性的技术途径

#### 三种途径：

#### (1) 时间重叠

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。

#### (2) 资源重复

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。

#### (3) 资源共享

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。

# 第 2 章 计算机系统量化分析基础

## 2.1 计算机体系结构的概念

## 2.2 计算机体系结构的发展

## 2.3 计算机系统设计和分析

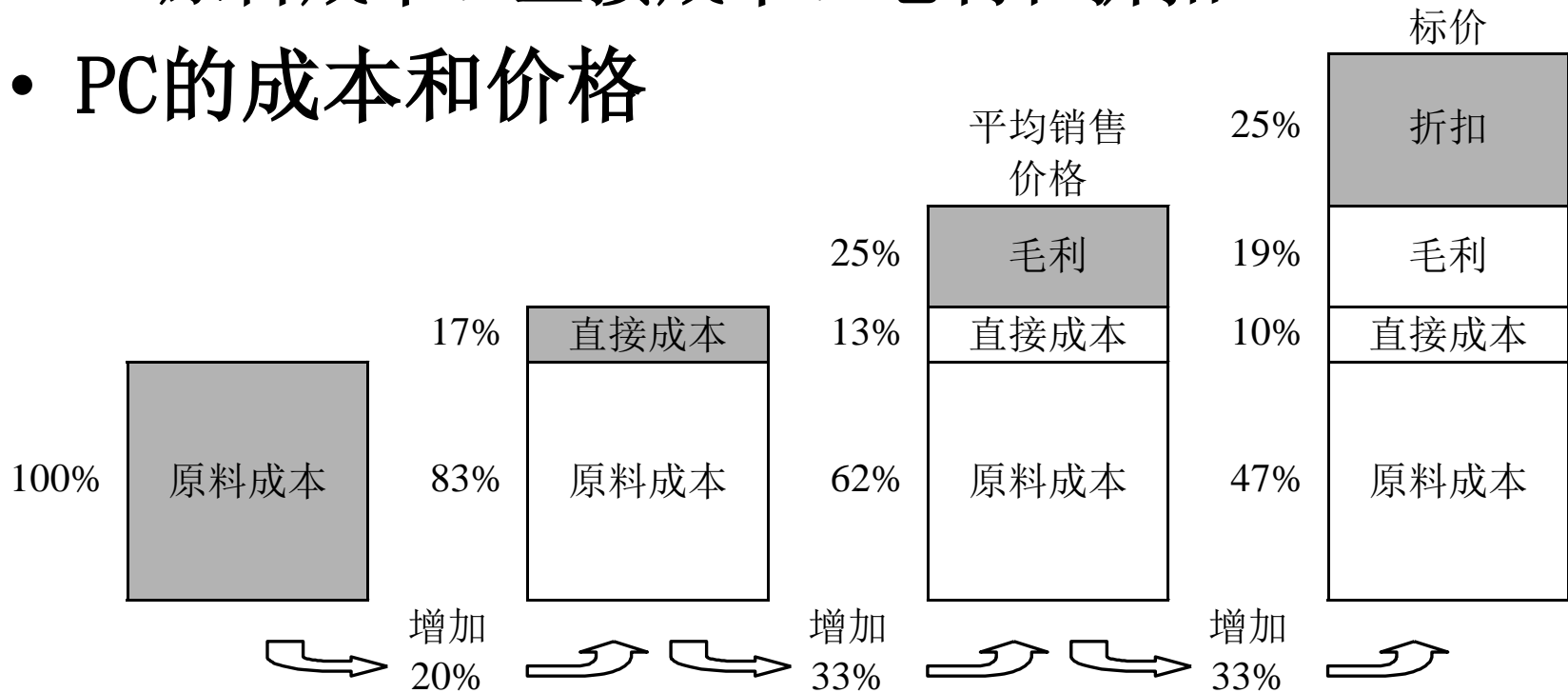
## 2.3 计算机系统设计和分析

- 2.3.1 成本与价格
- 2.3.2 基准测试程序
- 2.3.3 量化设计的基本原则



## 2.3.1 成本与价格

- 商品的标价（价格）由这样一些因素构成
  - 原料成本、直接成本、毛利和折扣
- PC的成本和价格



# 装机部件的成本分布（2015）

	型号	价格（元）	比例
处理器	Intel 酷睿 i7 5820k（盒）	2999	15.1%
散热器	安钛克 KuHLER H20 650	339	1.7%
主板	华硕 X99-A	2999	15.1%
显卡	七彩虹 iGame980-4GD5	4099	20.7%
内存	DDR4 2666 4GB 4条	1999	10.1%
硬盘1	三星 850EVO 250G SSD 固态硬盘	799	4.0%
硬盘2	希捷 2TB ST2000DM001	500	2.5%
显示器	戴尔 UltraSharpU2713HM	4099	20.7%
鼠标	罗技 G502	499	2.5%
键盘	雷蛇 终极版 2014	799	4.0%
机箱	安钛克 P280	399	2.0%
电源	鑫谷 RP PLUS650	299	1.5%
总价		19829	

## 2.3.1 成本与价格

- 系统的成本
  - 处理器板：~ 47%
  - I/O设备：~ 47%
  - 附件：~ 6%



## 2.3.1 成本与价格

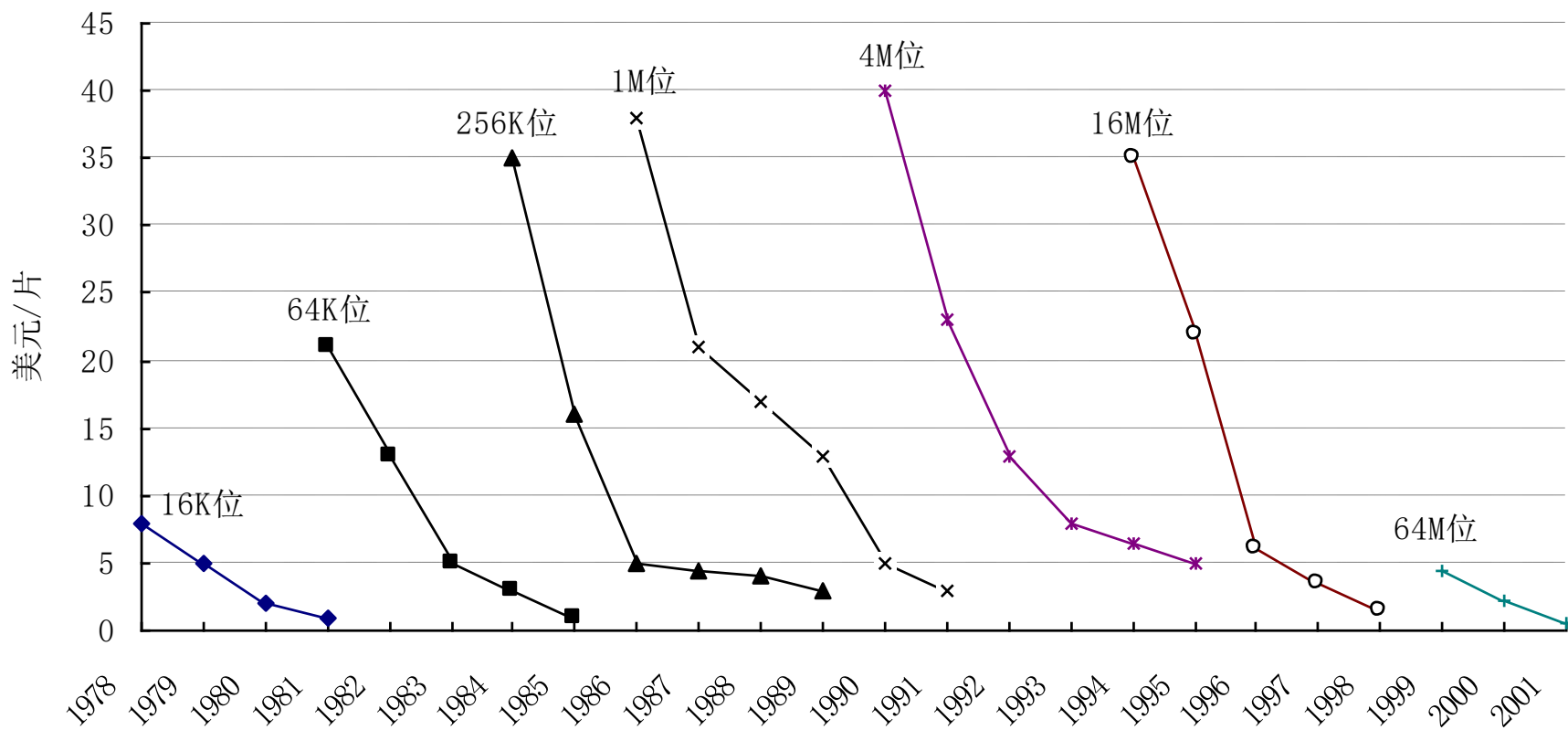
- **性能/成本设计**：设计者需取得性能与成本之间的平衡。
- 在过去的几十年中，计算机尺寸变小，因此**低成本设计**和**性能/成本设计**就显得日益重要，即使是巨型计算机制造商也发觉成本问题已日益重要。

## 2.3.1 成本与价格

- 对计算机系统成本产生影响的**主要因素**有：
  - 时间、产量、商品化等
- 对成本产生最直接影响的是**时间**。即使实现技术没有变动，计算机系统的制造成本也会不断下降。随着时间的推移，生产工艺会日渐稳定，产品的成品率会不断提高。产品的成本与成品率成反比。

# 成本-时间因素：学习曲线

- 产品价格随时间变化的特性，就是价格随时间下降的趋势



## 2.3 计算机系统设计和分析

- 2.3.1 成本与价格
- 2.3.2 基准测试程序
- 2.3.3 量化设计的基本原则

## 2.3.2 基准测试程序

- 性能与测试程序的执行时间相关，那么用什么做测试程序呢？
- 五类测试程序
  - 真实程序
  - 修正的（或者脚本化）应用程序
  - 核心程序
  - 小测试程序
  - 合成测试程序
- 测试程序包(组件, benchmark suites)
  - 选择一组各个方面有代表性的测试程序组成
  - 尽可能全面地测试了一个计算机系统的性能



# 测试程序包： [www.SPEC.org](http://www.SPEC.org)

- 基于UNIX，诞生于20世纪80年代
- 由真实程序和核心程序构成
- 采用C和Fortran两种语言，后增加C++
- 包括整数部分SPECint和浮点部分SPECfp
- 主要版本包括SPEC89、SPEC92、SPEC95、SPEC2000和SPEC2006 … SPEC2017等
  - SECP2017功能进一步细化
    - 台式计测试：SPEC CPU2000
    - 图像测试：SPECviewperf, SPECapc
    - NFS性能测试：SPECsfs
    - Web服务测试：SPECweb

# 其它测试包

- TPC-x
  - 测量事务处理、排队系统、决策支持、数据库应用等的性能
  - 1985年发布第一个TPC测试程序TPC-A，并先后发布多个修改版本并补充了四个不同的测试程序，构成TPC测试程序组件
- 嵌入式处理器
  - EEMBC: EDN 嵌入式微处理器测试程序联盟发布的测试程序包

## 2.3 计算机系统设计和分析

- 2.3.1 成本与价格
- 2.3.2 基准测试程序
- 2.3.3 量化设计的基本原则

## 2.3.3 量化设计的基本原则

### 1. 大概率事件优先原则

- 追求全局的最优结果

### 2. Amdahl定律

- 系统性能加速比，受限于该部件在系统中所占的重要性
- 可以定量计算

### 3. 程序的局部性原理

- 程序执行时所访问存储器在时-空上是相对地簇聚
- 这种簇聚包括指令和数据两部分

# 大概率事件优先的原则

- 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
- 要能够确定什么是大概率事件，同时要说明针对该事件进行的改进将如何提高机器的性能
- “好钢用在刀刃上”，事半功倍

# Amdahl定律

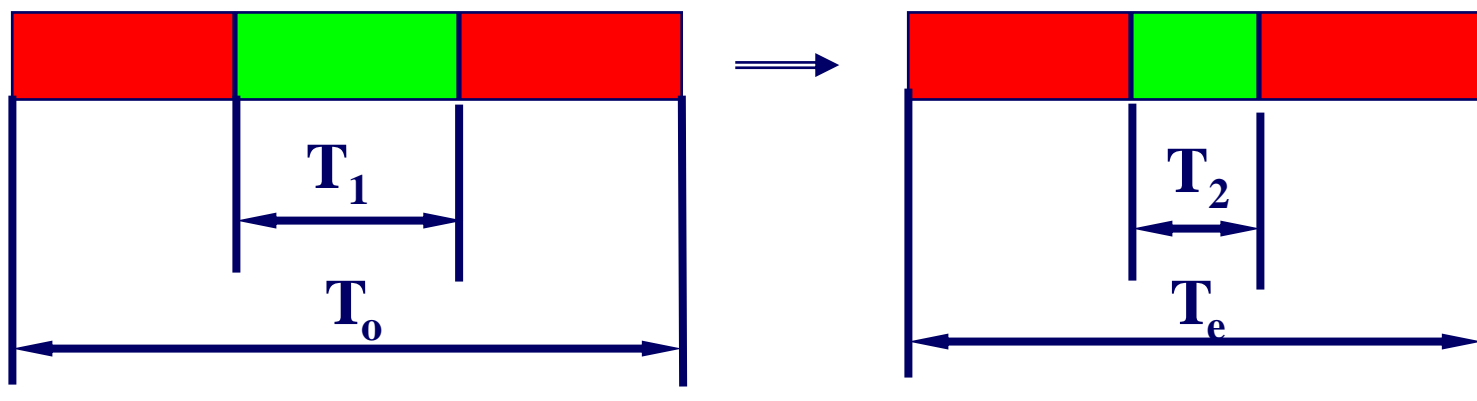
- 假设我们对机器（部件）进行某种改进，那么机器系统（部件）的加速比就是

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

- 核心概念：时间
- 系统加速比告诉我们改进后的机器比改进前快多少

# Amdahl定律

- 系统加速比依赖于两个因素
  - “**可改进比例**”：可改进部分在原系统计算时间中所占的比例，它总是小于等于1的
    - $T_1/T_0$
  - “**部件加速比**”可改进部分改进以后的性能提高，一般情况下它是大于1的
    - $T_1/T_2$



# Amdahl的系统执行时间

- 部件改进后，系统的总执行时间等于不可改进部分的执行时间加上可改进部分改进后的执行时间，即：
- 总执行时间<sub>改进后</sub>

$$= (1 - \text{可改进比例}) \times \text{总执行时间}_{\text{改进前}} + \frac{\text{可改进比例} \times \text{总执行时间}_{\text{改进前}}}{\text{部件加速比}}$$

$$= \text{总执行时间}_{\text{改进前}} \times \left[ (1 - \text{可改进比例}) + \frac{\text{可改进比例}}{\text{部件加速比}} \right]$$



# Amdahl的系统加速比

- 系统加速比为改进前与改进后总执行时间之比，为：

系统加速比

$$= \frac{\text{总执行时间}_{\text{改进前}}}{\text{总执行时间}_{\text{改进后}}}$$

$$= \frac{1}{(1 - \text{可改进比例}) + \frac{\text{可改进比例}}{\text{部件加速比}}}$$

# Amdahl定律的观点

## 1. 性能增加的递减规则

- 仅仅对计算机中的一部分做性能改进，则改进越多，系统获得的效果越小

## 2. Amdahl定律的一个重要推论

- 针对整个任务的一部分进行优化，则最大加速比不大于

$$\frac{1}{1 - \text{可改进比例}}$$

## 3. Amdahl定律衡量一个“好”的计算机系统

- 具有高性能价格比的计算机系统是一个带宽平衡的系统，而不是看它使用的某些部件的性能

# Amdahl定律练习

例1：假设在某程序的执行过程中，浮点操作时间占整个执行时间的10%，现希望对浮点操作加速。

- 设对浮点操作的加速比为 $S_f$ ，请推导出程序总的加速比 $S$ 和 $S_f$  之间的关系表达式；
- 请问程序的最大加速比可达多少？

# Amdahl定律练习



$S$ 与 $S_f$ 的关系表达式为:

$$\begin{aligned} S &= \frac{1}{(1 - f_f) + \frac{f_f}{S_f}} \\ &= \frac{1}{(1 - 10\%) + \frac{10\%}{S_f}} \\ &= \frac{1}{0.9 + \frac{0.1}{S_f}} \end{aligned}$$

# Amdahl定律练习



程序的最大加速比 $S_{max}$ 为:

$$S_{max} = \lim_{S_f \rightarrow \infty} \frac{1}{0.9 + \frac{0.1}{S_f}}$$
$$= 10/9$$

# 程序局部性

- 程序访问地址的分布不是随机的，而是相对地簇聚
  - 包括时间局部性和程序的空间局部性
- 程序的时间局部性
  - 程序即将用到的信息很可能就是目前正在使用的信息
- 程序的空间局部性
  - 程序即将用到的信息很可能与目前正在使用的信息在空间上相邻或者临近
- 其他局部性
  - 生产-消费局部性

# CPU性能公式

- 执行一个程序所需的CPU时间

**CPU时间 = 执行程序所需的时钟周期数 × 时钟周期时间**

其中：时钟周期时间是系统时钟频率的倒数。

- 每条指令执行的平均时钟周期数CPI

**CPI = 执行程序所需的时钟周期数 / IC**

IC：所执行的指令条数

- 程序执行的CPU时间可以写成

**CPU时间 = IC × CPI × 时钟周期时间**

时钟周期时间：取决于硬件实现技术和计算机组成

CPI：取决于计算机组成和指令系统的结构；

IC：取决于指令系统的结构和编译技术

# CPU性能公式

- 假设计算机系统有 $n$  种指令，其中第 $i$  种指令的处理时间为 $CPI_i$ ，在程序中第 $i$  种指令出现的次数为 $IC_i$ 。

$$\begin{aligned}T_{CPU} &= \sum (IC_i \times CPI_i) / f \\ &= \sum (IC_i \times CPI_i) \times T_{CLK}\end{aligned}$$

$$\begin{aligned}CPI &= \sum (IC_i \times CPI_i) / IC \\ &= \sum [(IC_i / IC) \times CPI_i]\end{aligned}$$

其中：  $IC_i / IC$  反映了第 $i$ 种指令在程序中所占的比例。



# CPU性能公式

例：假设我们考虑条件分支指令的两种不同设计方法如下：

- $\text{CPU}_A$ ：通过比较指令设置条件码，然后测试条件码进行分支。
- $\text{CPU}_B$ ：在分支指令中包括比较过程。

**$\text{CPU}_A$**

比较指令

测试分支

**$\text{CPU}_B$**

分支指令

# CPU性能公式

- 哪一个CPU更快?

$$T_{CPU_A} = CPI_{CPU_A} \times IC_{CPU_A} \times T_{CLK_A}$$

$$T_{CPU_B} = CPI_{CPU_B} \times IC_{CPU_B} \times T_{CLK_B}$$

**CPU<sub>A</sub>**

比较指令

测试分支

**CPU<sub>B</sub>**

分支指令

# CPU性能公式

在两种CPU中，条件分支指令都占用2个时钟周期而所有其它指令占用1个时钟周期。

**CPU<sub>A</sub>**

其它指令

比较指令

测试分支

**CPU<sub>B</sub>**

其它指令

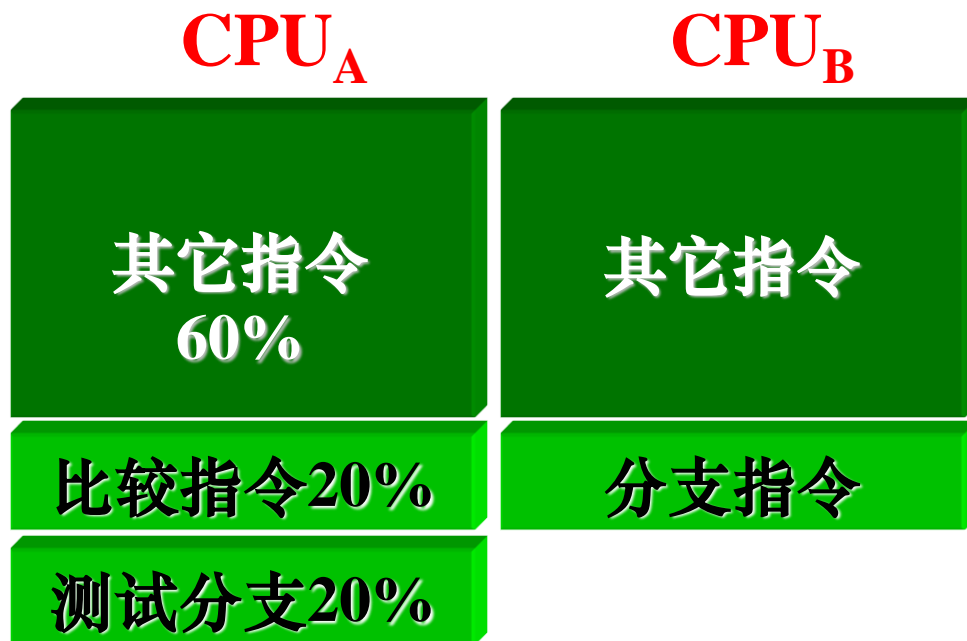
分支指令

$$CPI_{\text{分支}} = 2$$

$$CPI_{\text{其它}} = 1$$

# CPU性能公式

对于CPU<sub>A</sub>，执行的指令中分支指令占20%；由于每个分支指令之前都需要有比较指令，因此比较指令也占20%。



$$CPI_{\text{分支}} = 2$$

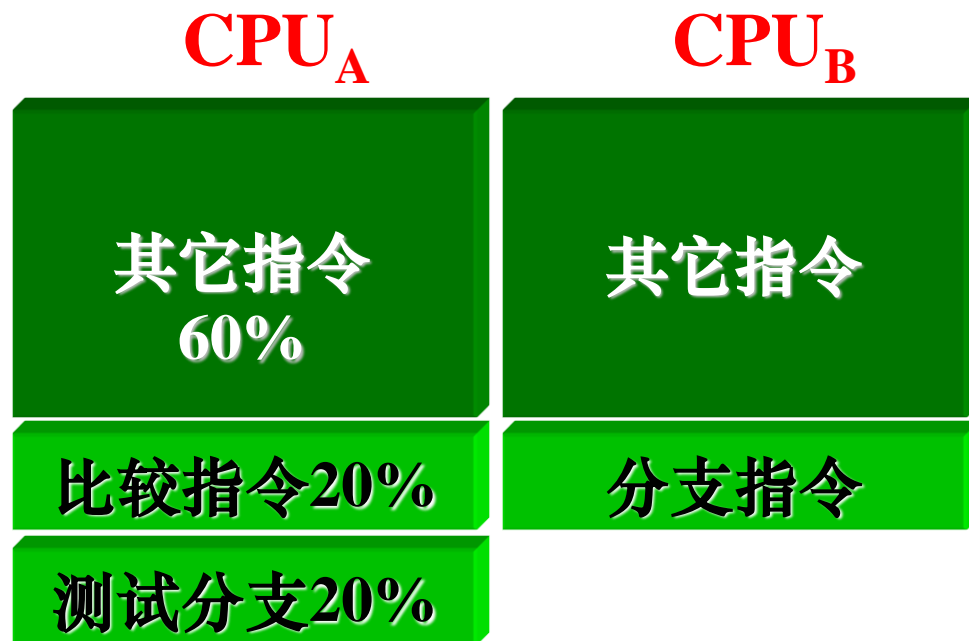
$$CPI_{\text{其他}} = 1$$

$$IC_B = (1 - 20\%) IC_A$$

$$IC_{B_{\text{分支}}} = IC_{A_{\text{分支}}}$$

# CPU性能公式

由于CPU<sub>A</sub>在分支时不需要比较, 假设它的时钟频率比CPU<sub>B</sub>快1.25倍。



$$CPI_{\text{分支}} = 2$$

$$CPI_{\text{其他}} = 1$$

$$IC_B = (1 - 20\%) IC_A$$

$$IC_{B_{\text{分支}}} = IC_{A_{\text{分支}}}$$

$$T_{\text{clk}_B} = 1.25 T_{\text{clk}_A}$$

# CPU性能公式

$$T_{CPU} = \sum (IC_i \times CPI_i) \times T_{CLK}$$

**CPU<sub>A</sub>**

其它指令  
60%

比较指令20%

测试分支20%

**CPU<sub>B</sub>**

其它指令

分支指令

$$CPI_{\text{分支}} = 2$$

$$CPI_{\text{其他}} = 1$$

$$IC_B = (1 - 20\%) IC_A$$

$$IC_B_{\text{分支}} = IC_A_{\text{分支}}$$

$$T_{\text{clk}_B} = 1.25 T_{\text{clk}_A}$$

# CPU性能公式

$$T_{CPU} = \sum (IC_i \times CPI_i) \times T_{CLK}$$

条件:

$$CPI_{\text{分支}} = 2 \quad CPI_{\text{其他}} = 1$$

$$IC_B = (1 - 20\%) IC_A \quad IC_{B_{\text{分支}}} = IC_{A_{\text{分支}}}$$

$$T_{\text{clk}_B} = 1.25 T_{\text{clk}_A}$$

求解CPU<sub>A</sub>的执行时间:

$$\begin{aligned} T_{CPU_A} &= (80\% \times 1 + 20\% \times 2) \times IC_A \times T_{CLK_A} \\ &= IC_A \times 1.2 \times T_{CLK_A} \end{aligned}$$

# CPU性能公式

$$T_{CPU} = \sum (IC_i \times CPI_i) \times T_{CLK}$$

条件:

$$CPI_{\text{分支}} = 2 \qquad CPI_{\text{其他}} = 1$$

$$IC_B = (1 - 20\%) IC_A \qquad IC_{B_{\text{分支}}} = IC_{A_{\text{分支}}}$$

$$T_{\text{clk}_B} = 1.25 T_{\text{clk}_A}$$

求解CPU<sub>B</sub>的执行时间:

$$\begin{aligned} T_{CPU_B} &= (75\% \times 1 + 25\% \times 2) \times IC_B \times T_{CLK_B} \\ &= 1.25 \times 0.8 \times IC_A \times 1.25 \times T_{CLK_A} \\ &= IC_A \times 1.25 \times T_{CLK_A} \end{aligned}$$



# CPU性能公式

**CPU<sub>A</sub>**

其它指令

比较指令

测试分支

**CPU<sub>B</sub>**

其它指令

分支指令

$$T_{CPU_A} = 1.2 \times IC_A \times T_{CLK_A}$$

$$T_{CPU_B} = 1.25 \times IC_A \times T_{CLK_A}$$

**CPU<sub>A</sub>快，即使其执行的指令数较CPU<sub>B</sub>多。**

# CPU性能公式

- 如果CPUA的时钟周期时间仅仅比CPUB快1.1倍，  
哪一个CPU更快呢？

**CPU<sub>A</sub>**

其它指令

比较指令

测试分支

**CPU<sub>B</sub>**

其它指令

分支指令

$$T_{CPU_A} = 1.2 \times IC_A \times T_{CLK_A}$$

$$T_{CPU_B} = 1.1 \times IC_A \times T_{CLK_A}$$

# 本章小结

- 讨论计算机体系结构的基本概念
  - 计算机系统层次结构概念
  - 经典计算机体系结构概念
  - 计算机组成和计算机实现技术
  - 现代计算机体系结构所研究的范围和内容
- 计算机体系结构的发展
  - 计算机的分代和分型
  - 计算机应用需求和实现技术等方面的发展对计算机体系结构发展的促进作用

# 本章小结

- 影响体系结构设计的成本和价格因素
  - 加深对计算机体系结构技术的理解
- 并行性技术提高计算机系统性能
  - 并行性技术的基本概念、分类、实例
  - 时间重叠、资源重复、资源共享
- 定量分析技术
  - 大率率事件优先原则
  - Amdahl定律
  - 程序的局部性原理
- 性能公式

## 第二章作业

- 王志英教材，P42，T5，T7

- 阅读报告：

基于“Requirements Bottlenecks and Good Fortune Agents for Microprocessor Evolution”, Y. Patt, IEEE 2001，对当前计算机领域新技术的调研和展望。