

# 计算机体系结构

## 第四讲

计算机科学与技术学院

舒燕君

# Recap

- 计算机体系结构的发展
  - ✓ 核心技术的发展
  - ✓ 体系结构的发展
  - ✓ 并行体系结构的发展
  - ✓ 处理器技术的发展
- 成本与价格：学习曲线
- 基准测试程序

## 1.2.6 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)

## 1.2.6 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

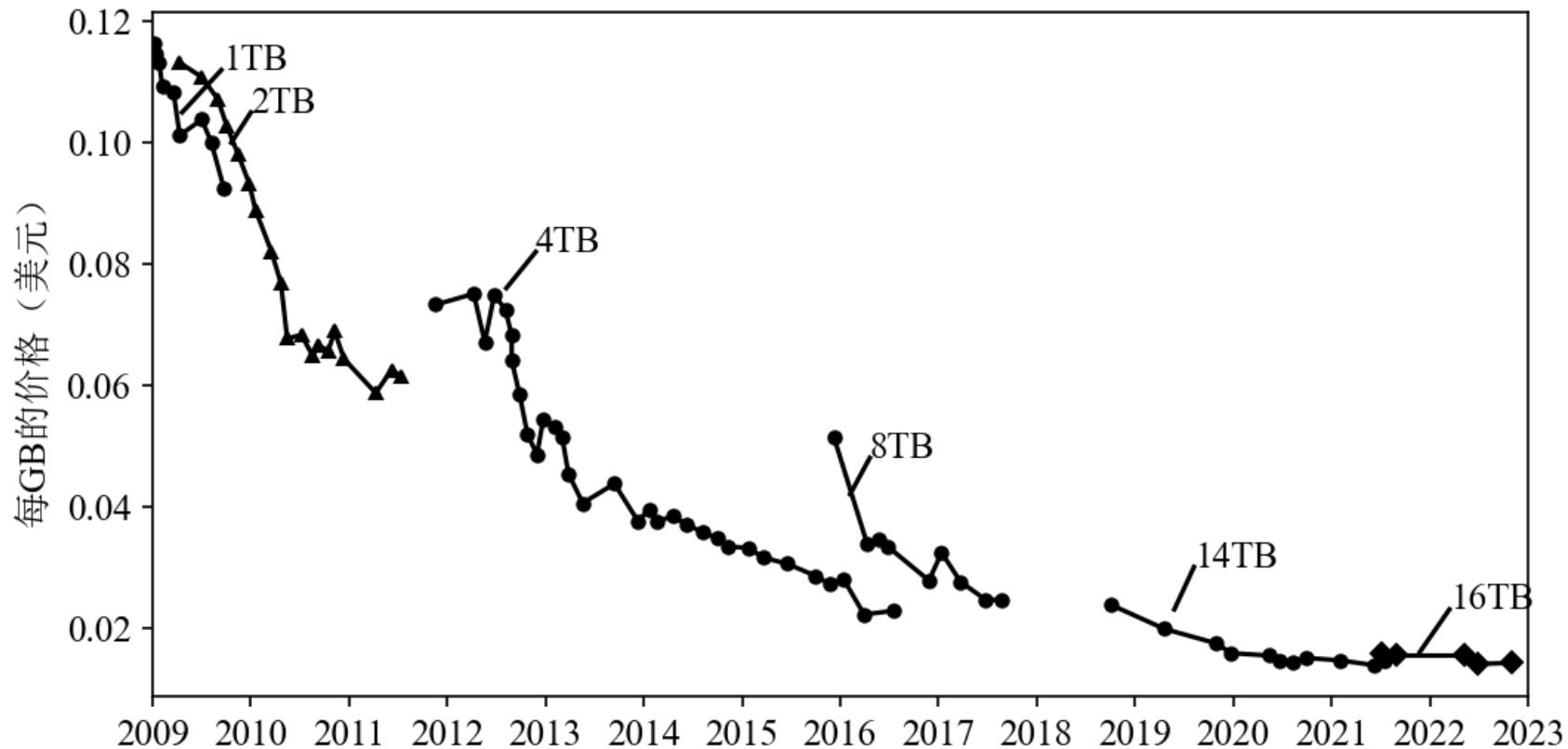
引入空间因素，以数量取胜。通过重复设置硬件资源，大幅度地提高计算机系统的性能。

(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。

# 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



## 1.3.2 基准测试程序

关键因素	分析建模	模拟	实际测量
系统所处的生命周期阶段	任何阶段	任何阶段	做出原型以后
时间消耗	短	中等	不一定
可用工具	分析师	模拟语言	测量仪器
准确性	低	中等	变化
参数修改（设计调整）	方便	中等	困难
成本	低	中等	高
结果的可接受程度	低	中等	高

## 1.3.2 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

# 第1章 计算机系统量化分析基础

## 1.1 计算机体系结构的概念

## 1.2 计算机体系结构的发展

## 1.3 计算机系统设计和分析

# 1.3 计算机系统设计和分析

- 1.3.1 成本与价格
- 1.3.2 基准测试程序
- 1.3.3 量化设计的基本原则

### 1.3.3 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
- 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

### 1.3.3 量化设计的基本原则

#### 2. 并行性原理

- 提升计算机系统性能的最常用技术手段
- 尽可能多的让程序中可以同时工作的部分并行工作，不仅可以提高性能，而且可以提高资源利用率
- 可以划分为元器件级、部件级、指令级、数据级、请求级、系统级、应用级
- 挖掘并行性和利用并行性原则是计算机体系结构研究的重要方面，尤其是摩尔定律速度放缓（甚至失效）的今天

## 1.3.3 量化设计的基本原则

### 3. 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

### 4. 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

# Amdahl定律

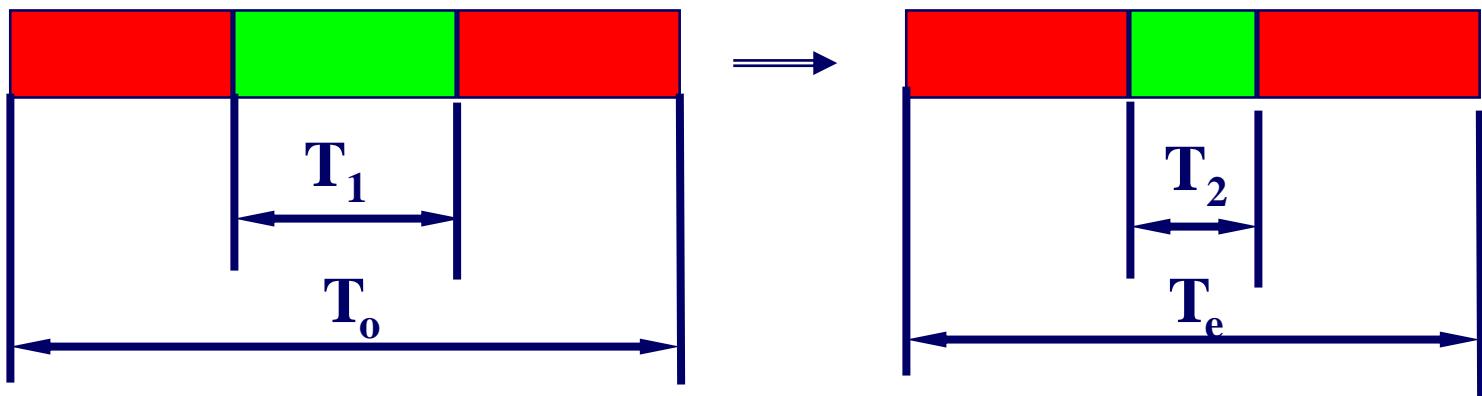
- 假设我们对机器（部件）进行某种改进，那么机器系统（部件）的加速比就是

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

- 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>

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

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

# 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

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

# 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

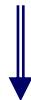
- 具有高性能价格比的计算机系统是一个带宽平衡的系统，而不是看它使用的某些部件的性能

# 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



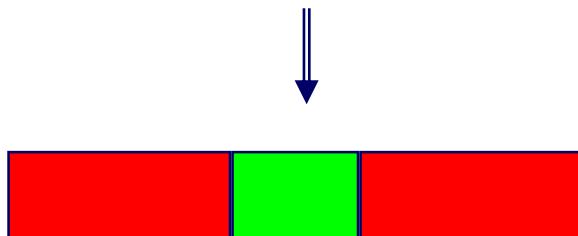
$S$ 与 $S_f$ 的关系表达式为：

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

# 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
  - 包括程序的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

# 局部性举例

```
sum = 0;  
for (i = 0; i < n; i++)  
    sum += a[i];  
return sum;
```

## ■ 对数据的引用

- 顺序访问数组元素  
(步长为1的引用模式)
- 变量sum在每次循环迭代中被引用一次

空间局部性

时间局部性

## ■ 对指令的引用

- 顺序读取指令
- 重复循环执行for循环体

空间局部性

时间局部性

### 1.3.3 量化设计的基本原则

#### 5. 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：假设我们考虑分支指令的两种不同设计方法如下：

- CPU<sub>A</sub>: 通过比较指令设置条件码，然后测试条件码进行分支。
- CPU<sub>B</sub>: 在分支指令中包括条件比较过程。

CPU<sub>A</sub>

比较指令

测试分支

CPU<sub>B</sub>

条件分支

# 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

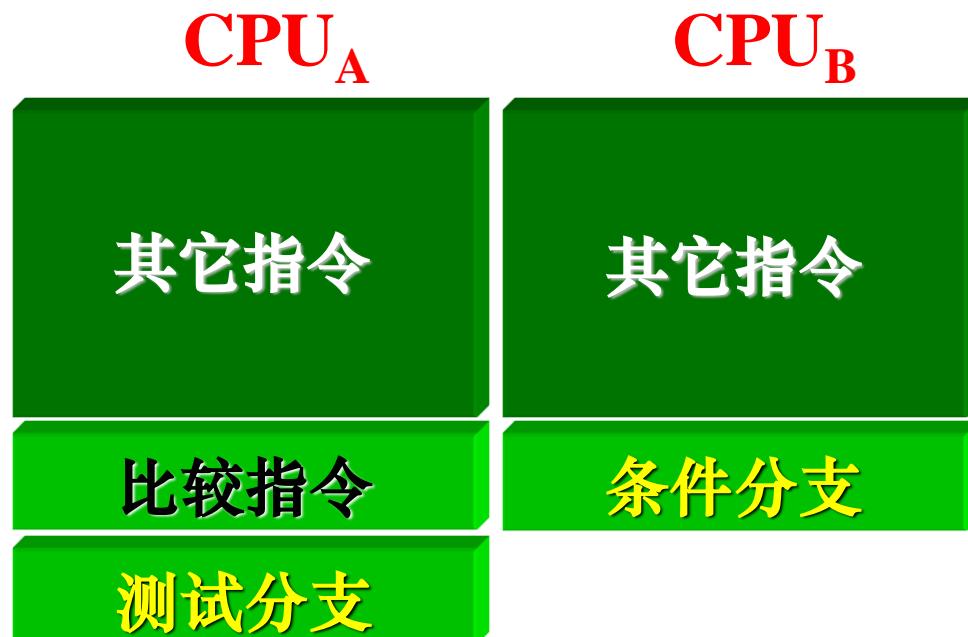
测试分支

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

条件分支

# CPU性能公式

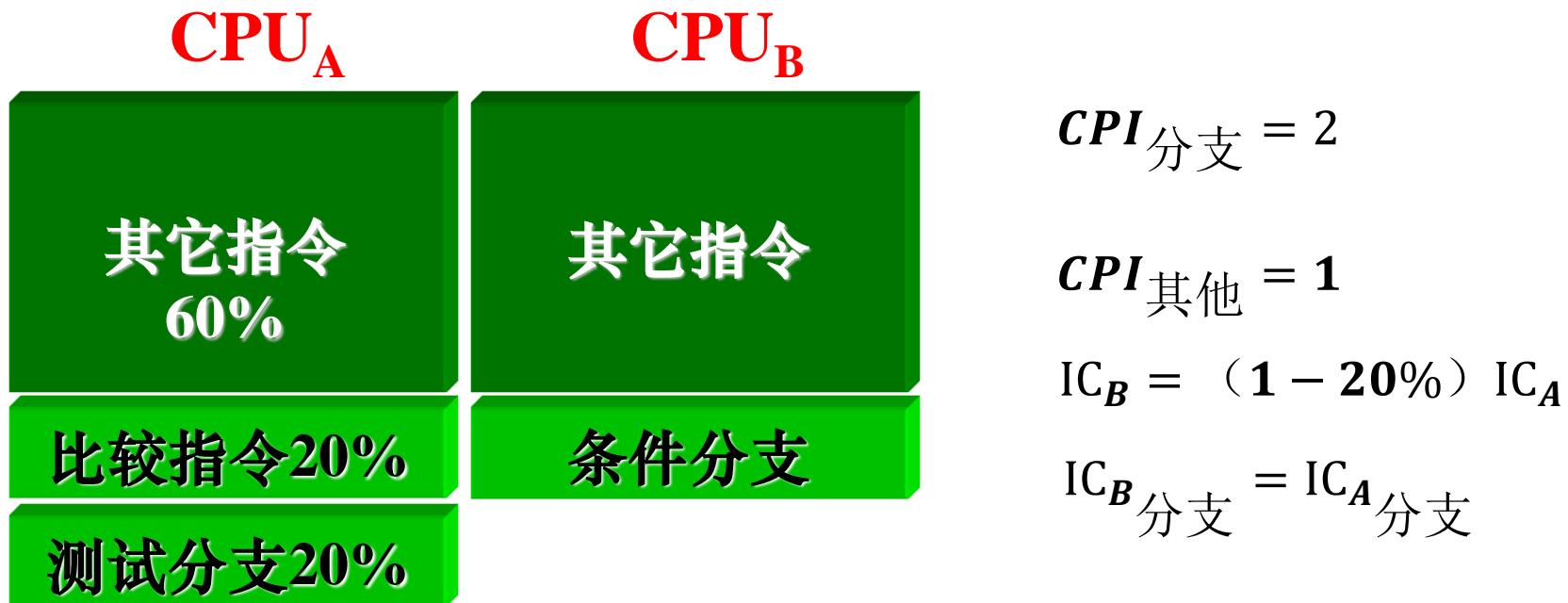
在两种CPU中，分支指令都占用2个时钟周期而所有其它指令占用1个时钟周期。



$$CPI_{\text{分支}} = 2$$
$$CPI_{\text{其它}} = 1$$

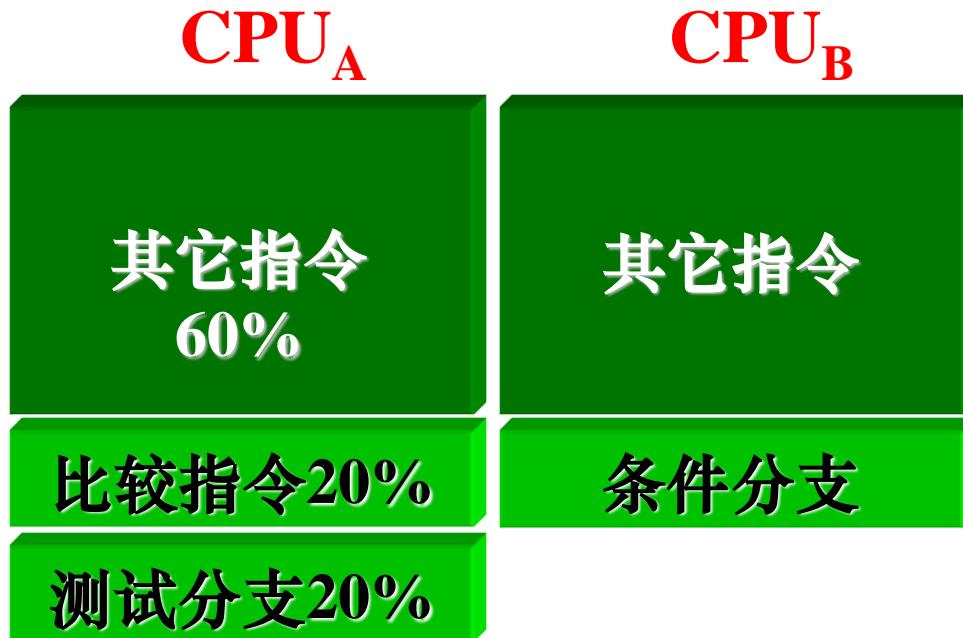
# CPU性能公式

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%。



# 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。



$$CPI_{\text{分支}} = 2$$

$$CPI_{\text{其他}} = 1$$

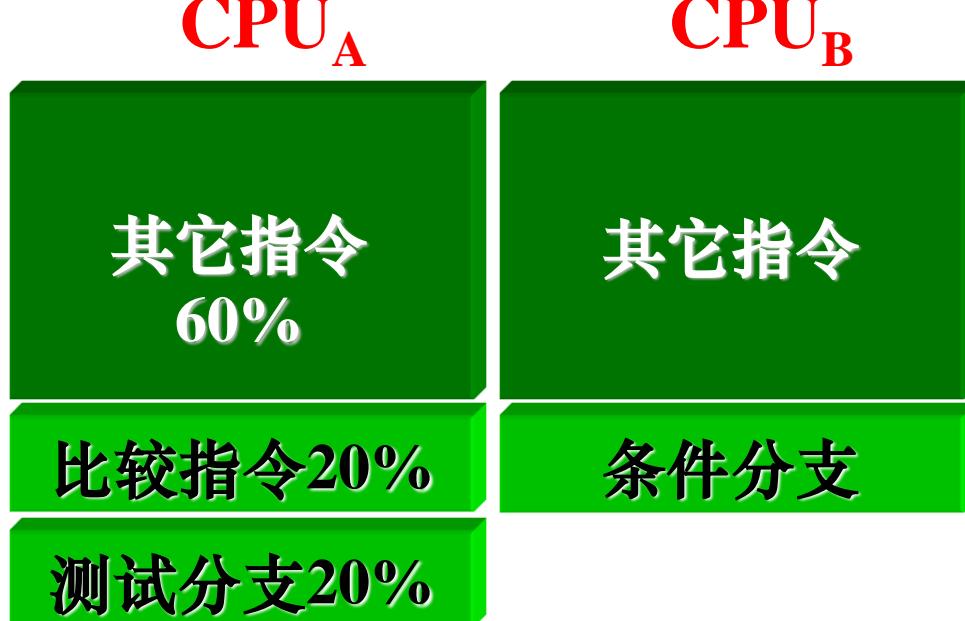
$$IC_B = (1 - 20\%) IC_A$$

$$IC_B^{\text{分支}} = IC_A^{\text{分支}}$$

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

# CPU性能公式

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



$$CPI_{\text{分支}} = 2$$

$$CPI_{\text{其他}} = 1$$

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

$$IC_B^{\text{分支}} = IC_A^{\text{分支}}$$

$$T_{clk_B} = 1.25 T_{clk_A}$$

# CPU性能公式

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

条件：

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

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

$$T_{clk_B} = 1.25 T_{clk_A}$$

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：

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

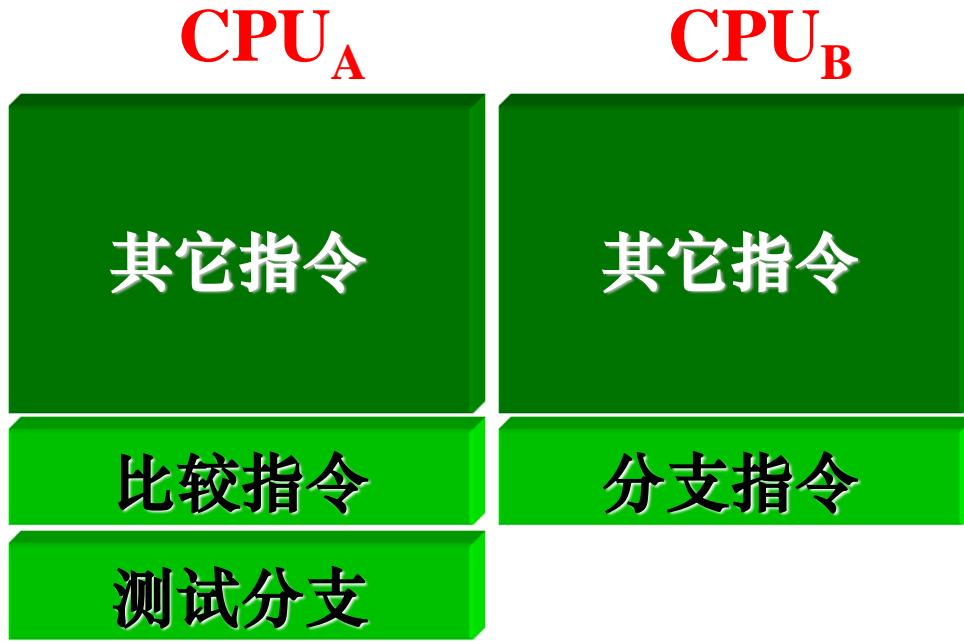
$$IC_B = (1 - 20\%) IC_A \quad IC_B^{\text{分支}} = IC_A^{\text{分支}}$$

$$T_{clk_B} = 1.25 T_{clk_A}$$

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



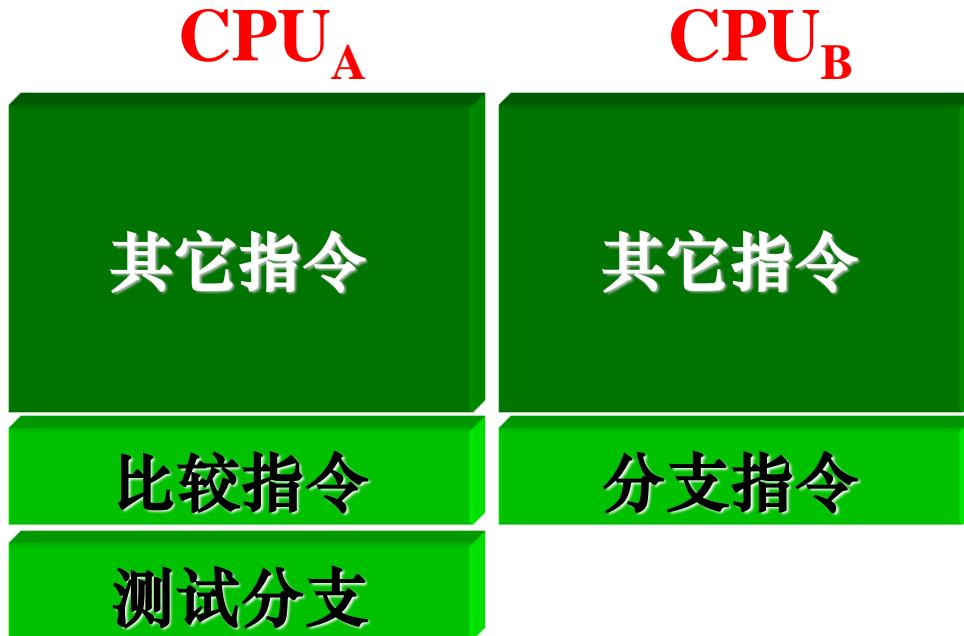
$$T_{CPU_A} = 1.2 \times IC_A \times T_{CLK_A}$$

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

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CPU<sub>A</sub>的时钟周期时间仅仅比CPU<sub>B</sub>快1.1倍，  
哪一个CPU更快呢？



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

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

# 本章小结

- 讨论计算机体系结构的基本概念
  - 经典计算机体系结构概念
  - 计算机系统层次结构概念
  - 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结构发展的促进作用

# 本章小结

- 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
- 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
- 定量分析技术
  - 大概率事件优先原则
  - 并行性原理
  - Amdahl定律
  - 程序的局部性原理
  - CPU性能公式

# 第一章作业

- 王志英教材， P42， T5， T7 （张春元教材， P49， T5， T7）
- 阅读以下论文，结合第一章的课程内容完成有关对计算机系统发展趋势展望的报告（3000字以内）：
  - [1] Y. Patt. Requirements Bottlenecks and Good Fortune Agents for Microprocessor Evolution, IEEE 2001.
  - [2] G. Amdahl. Architecture of the IBM System/360, IBM Journal, 1964.

# Quiz 1

1. 假设某台机器运行一个测试程序的执行时间为100秒，其中CPU处理时间占90%，I/O处理时间占10%，若CPU的执行速度每年能够提高50%，请问5年后在这台机器上，运行该测试程序将耗费多少秒？I/O处理时间占多少百分比？
2. 某程序的条件判断的分支占10%，考虑运行该程序的CPU中条件分支的两种设计方案：（1）CPU1通过比较指令设置条件码，然后测试条件码进行分支（测试分支）；（2）CPU2在分支指令中就包含了比较过程（条件分支）。假设条件分支指令和测试分支指令都需要2个时钟周期，其它所有指令需要1个时钟周期，由于CPU2在分支时需要比较，因此CPU1的时钟频率为CPU2的1.2倍，请对比两种CPU的设计方案。

**注意：写上学号-姓名、自觉自主完成**