

## 1.4 计算机设计的量化准则

### 1.4.1 计算机系统设计的定量原理

#### 1.Amdahl定律

系统对某一部件采用某种更快执行方式所能获得的系统性能改进程度，取决于这种执行方式被使用的频率或所占总执行时间的比例。

首先，**Amdahl**定律定义了**加速比**的概念。假设对机器进行某种改进，那么机器系统的加速比为：

$$\text{加速比} = \frac{\text{改进后的性能}}{\text{改进前的性能}}$$



## 1.4 计算机设计的量化准则

或

$$\text{加速比} = \frac{\text{改进前的总执行时间}}{\text{改进后的总执行时间}}$$

系统加速比告诉我们改进后的机器比改进前快多少。Amdahl定律使我们能快速得出改进所获得的效益。系统加速比依赖于两个因素：

可改进比例 ( $F_e$ )，它总是小于1的。

性能提高比 ( $S_e$ )，它总是大于1的。



## 1.4 计算机设计的量化准则

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



## 1.4 计算机设计的量化准则

改进后整个任务的执行时间为：

$$T_n = T_0 \cdot (1 - F_e + \frac{F_e}{S_e})$$

其中 $T_0$ 为改进前的整个任务的执行时间。

改进后整个系统的加速比为：

$$S_n = \frac{T_0}{T_n} = \frac{1}{(1 - F_e) + \frac{F_e}{S_e}}$$

其中 **(1-Fe)**表示不可改进部分。



## 1.4 计算机设计的量化准则

实际上，**Amdahl**定律还表达了一种性能增加的递减规则：如果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。**Amdahl**定律的一个重要推论是：如果只针对整个任务的一部分进行优化，那么所获得的加速比不大于：

$$\frac{1}{(1 - F_e)}$$



## 1.4 计算机设计的量化准则

**例1：**假设将某一部件的处理速度加快到**10**倍，该部件的原处理时间仅为整个运行时间的**40%**，则采用加快措施后能使整个系统的性能提高多少？

解：

由题意可知： **$F_e=0.4$** ,  **$S_e=10$** ，根据**Amdahl**定律，加速比为：

$$S_n = \frac{1}{(1 - 0.4) + \frac{0.4}{10}} = \frac{1}{0.64} = 1.56$$



## 1.4 计算机设计的量化准则

**例2：**某计算机系统采用浮点运算部件后，浮点运算速度提高到原来的**25**倍，而系统运行某一程序的整体性能提高到原来的**4**倍，试计算该程序中浮点操作所占的比例。

解：由题意可知： **$S_e=25$** ， **$S_n=4$** ，根据**Amdahl定律**：

$$4 = \frac{1}{(1 - Fe) + \frac{Fe}{25}}$$

由此可得： **$Fe \approx 78.1\%$** 。



## 1.4 计算机设计的量化准则

**例3：**求浮点数（**FP**）平方根的不同实现方法在性能上可能有很大差异。假设在程序中求浮点平方根（**FPSQR**）操作占总执行时间的**20%**，一种方法是增加专门的**FPSQR** 硬件，可提高速度**10**倍；另一种方法是提高所有**FP**指令的速度，**FP**指令占总执行时间的**50%**，**FP**指令的速度提高为原来的**1.6**倍。试比较这两种方法。

$$S_{n(FPSQR)} = \frac{1}{(1-0.2) + \frac{0.2}{10}} = \frac{1}{0.82} = 1.22$$

$$S_{n(FP)} = \frac{1}{(1-0.5) + \frac{0.5}{1.6}} = \frac{1}{0.8125} = 1.23$$





## 1.4 计算机设计的量化准则

### 2. CPU性能公式

程序执行的**CPU**时间为：

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

若将程序执行过程中所处理的指令数，记为**IC**。这样可以获得一个与计算机系统结构有关的参数，即“指令时钟数**CPI**”。

$$\text{CPI} = \frac{\text{CPU时钟周期数}}{\text{IC}}$$



## 1.4 计算机设计的量化准则

### 主频和**CPU**时钟周期

主频是衡量**CPU**速度的重要参数。**CPU**的主频又称为时钟频率，表示在**CPU**内数字脉冲信号振荡的速度，与**CPU**实际的运算能力并没有直接关系。主频的倒数就是**CPU**时钟周期，这是**CPU**中最小的时间元素。每个动作至少需要一个时钟周期。



## 1.4 计算机设计的量化准则

### CPI

**CPI (Cycles per Instruction)** 就是每条指令执行所用的时钟周期数。由于不同指令的功能不同，造成指令执行时间不同，也即指令执行所用的时钟数不同，所以**CPI**是一个平均值。在现代高性能计算机中，由于采用各种并行技术，使指令执行高度并行化，常常是一个系统时钟周期内可以处理若干条指令，所以**CPI**参数经常用**IPC**

(**Instructions per Cycle**) 表示，即每个时钟周期执行的指令数。



## 1.4 计算机设计的量化准则

$$\text{CPU时间} = \frac{\text{IC} \times \text{CPI}}{\text{时钟频率}}$$

这个公式通常称为**CPU**性能公式。它的三个参数反映了与系统结构相关的三种技术：

- ① 时钟频率：反映了计算机实现技术、生产工艺和计算机组织。
- ② **CPI**：反映了计算机实现技术、计算机指令系统的结构和组织。
- ③ **IC**：反映了计算机指令级的结构和编译技术。



## 1.4 计算机设计的量化准则

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<sub>i</sub>**，在程序中第**i**种指令出现的次数为**I<sub>i</sub>**，则有：

$$CPI = \frac{\sum_{i=1}^n CPI_i \times I_i}{IC} = \sum_{i=1}^n (CPI_i \times \frac{I_i}{\underline{IC}})$$



## 1.4 计算机设计的量化准则

### 1.4.2 衡量计算机系统性能的主要标准

#### 1. 吞吐量和响应时间

吞吐量和响应时间是描述计算机系统性能常用的参数，也是用户所关心的。**吞吐量是指系统在单位时间内处理请求的数量。响应时间是指系统对请求作出响应的的时间，响应时间包括CPU时间（运行一个程序所花费的时间）与等待时间（用于磁盘访问、存储器访问、I/O操作、操作系统开销等时间）的总和。**



## 1.4 计算机设计的量化准则

## 2. 运算速度

(1) 时钟频率（主频）：用于同类处理机之间 如：PentiumII/450 比 PentiumII/300快50%，...

(2) 指令执行速度 一种很经典的表示方法 MIPS (Million Instructions Per Second), KIPS, GIPS, TIPS

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



## 1.4 计算机设计的量化准则

主要缺点：

(1) 不同指令的速度差别很大

(2) 指令使用频度差别很大

(3) 有相当多的非功能性指令

**功能性指令**：加、减、乘、除等

另一种替代标准

$$\text{MFLOPS} = \frac{\text{程序中的浮点操作次数}}{\text{执行时间} \times 10^6}$$





## 1.4 计算机设计的量化准则

其中，**Fz**为处理机的工作主频；**CPI (Cycles Per Instruction)**为每条指令所需的平均时钟周期数；**IPC (Instruction Per Cycle)**为每个时钟周期平均执行的指令条数。

**例4：**计算**Pentium II 450**处理机的运算速度。

解：由于**PentiumII 450**处理机的**IPC = 2** (或**CPI = 0.5**), **Fz = 450MHz**, 因此,

$$\text{MIPS}_{\text{Pentium II 450}} = \text{Fz} \times \text{IPC} = 450 \times 2 = 900(\text{MIPS})$$



## 1.4 计算机设计的量化准则

**例5：**微机**A**和**B**是采用不同主频的**CPU**芯片，片内逻辑电路完全相同。

**(1)** 若**A**机的**CPU**主频为**8MHz**，**B**机为**12MHz**，则**A**机的**CPU**时钟周期为多少？

**(2)** 如**A**机的平均指令执行速度为**0.4MIPS**，那么**A**机的平均指令周期为多少？

**(3)** **B**机的平均指令执行速度为多少？

解：**(1)** **A**机的**CPU**主频为**8MHz**，所以**A**机的**CPU**时钟周期  
 $= 1 \div 8\text{MHz} = 0.125\mu\text{s}$ 。

**(2)** **A**机的平均指令执行速度为**0.4MIPS**，所以**A**机的平均指令周期  
 $= 1 \div 0.4\text{MIPS} = 2.5\mu\text{s}$ 。

**(3)** **A**机平均每条指令时钟周期数  $= 2.5\mu\text{s} \div 0.125\mu\text{s} = 20$   
而微机**A**和**B**片内逻辑电路完全相同，所以**B**机平均每条指令的时钟周期数也为**20**。

**B**机的平均指令执行速度  $= \text{主频} \div \text{CPI} = 12 \div 20 \text{ MIPS}$   
 $= 0.6\text{MIPS}$ 。



## 1.4 计算机设计的量化准则

## 3、峰值速度

峰值指令速度**MIPS**、**GIPS**、**TIPS**

**Pentium III 500**有**3**条指令流水线，则其峰值指令速度为：

$$3 \times 500\text{MHz} = 1500 \text{ (MIPS)}$$

即每秒**15**亿次

**例8**：一个由**8**台机器组成的**Cluster**系统，每台机器是**4**个**PentiumIII 500**组成的**SMP**系统；计算这个**Cluster**系统的指令峰值速度。



## 1.4 计算机设计的量化准则

解：

峰值指令速度：

$$500\text{MHz} \times 8 \times 4 \times 3 = 48(\text{GIPS})$$

即每秒480亿次。



## 1.4 计算机设计的量化准则

4. 等效指令速度：吉普森（**Gibson**）法

$$\text{等效指令执行时间 } T = \sum_{i=1}^n (W_i \times T_i)$$

$$\text{等效指令速度 } MIPS = 1 / \sum_{i=1}^n \frac{W_i}{MIPS_i}$$

$$\text{等效 } CPI = \sum_{i=1}^n (CPI_i \times \underline{W_i})$$



## 1.4 计算机设计的量化准则

其中，

**Wi**: 指令使用频度, **i**: 指令种类  
静态指令使用频度: 在程序中直接

统计

动态指令使用频度: 在程序执行过程中统计

在计算机发展的早期, 用加法指令的运算速度来衡量计算机的速度。通常: 加、减法**50%**, 乘法**15%**, 除法**5%**, 程序控制**15%**, 其他**15%**。



## 1.4 计算机设计的量化准则

**例6：**我国最早研制的小型计算机**DJS-130**，定点**16**位，加法每秒**50**万次，但没有硬件乘法和除法指令，用软件实现乘法和除法，速度低**100**倍左右。求等效速度。

解：

定点等效速度为：

$$\text{等效指令速度 MIPS} = 1 / \left( \frac{0.80}{0.5} + \frac{0.20}{0.5/100} \right) = 0.02 \text{ MIPS}$$

即每秒2万次，由于乘法和除法用软件实现，等效速度降低了25倍。



## 1.4 计算机设计的量化准则

**例7：**假设在程序中浮点开平方操作**FPSQR**的比例为**2%**，它的**CPI**为**100**；其他浮点操作**FP**的比例为**23%**，它的**CPI=4.0**；其余**75%**指令的**CPI=1.33**，计算该处理机的等效**CPI**。如果**FPSQR**操作的**CPI**也为**4.0**，重新计算等效**CPI**。





## 1.4 计算机设计的量化准则

解：

$$\begin{aligned}\text{等效CPI}_1 &= 100 \times 2\% + 4 \times 23\% + \\ &1.33 \times 75\% \\ &= 3.92\end{aligned}$$

$$\begin{aligned}\text{等效CPI}_2 &= 4 \times \underline{25\%} + 1.33 \times 75\% \\ &= 2.00\end{aligned}$$

由于改进了仅占2%的FPSQR操作的CPI，使等效速度提高了近一倍。



## 1.4 计算机设计的量化准则

### 1.4.3 计算机性能的比较

	计算机A	计算机B	计算机C
程序P1 (s)	1	10	20
程序P2 (s)	1000	100	20
总计 (s)	1001	110	40



## 1.4 计算机设计的量化准则

总执行时间

最简单的相对性能综合评价方法

算术平均速度

$$\frac{1}{n} \sum_{i=1}^n T_i$$



## 1.4 计算机设计的量化准则

## 加权执行时间

为每一个程序赋予一个权重值 $w_i$ ，将各个程序的权重值与执行时间的乘积加起来。

	A	B	C	<b>w1</b>	<b>w2</b>	<b>w3</b>
程序P1 (s)	1	10	20	0.50	0.909	0.999
程序P2 (s)	1000	100	20	0.50	0.091	0.001
平均执行时间 <b>w1</b>	500.5	55	20			
平均执行时间 <b>w2</b>	91.91	18.19	20			
平均执行时间 <b>w3</b>	2	10.09	20			



## 1.4 计算机设计的量化准则

## 加权平均速度

$$\sum_{i=1}^n W_i \bullet T_i$$



## 1.4 计算机设计的量化准则

### 归一化执行时间

将执行时间对一台参考机器归一化，然后取归一化执行时间的平均值。平均归一化执行时间可以表示成算术平均值，也可以表示成几何平均值。



## 1.4 计算机设计的量化准则

几何平均速度:

$$G = \sqrt[n]{\prod_{i=1}^n ETR_i}$$



## 1.4 计算机设计的量化准则

算术平均值因参考机器不同而不同，  
几何平均值不随参考机器的变化而变化。

	A	B	C	A	B	C	A	B	C
程序P1	1.0	10.0	20.0	0.1	1.0	2.0	0.05	0.5	1.0
程序P2	1.0	0.1	0.02	10.0	1.0	0.2	50.0	5.0	1.0
算术平均值	1.0	5.05	10.01	5.05	1.0	1.1	25.3	2.75	1.0
几何平均值	1.0	1.0	0.63	1.0	1.0	0.63	1.58	1.58	1.0
总时间	1.0	0.11	0.04	9.1	1.0	0.36	25.03	2.75	1.0





## 1.4 计算机设计的量化准则

其中，**ETR(execution time ratio)**中的**n** 指不同的程序。

几何平均速度与机器无关，与程序的执行时间无关。



## 1.4 计算机设计的量化准则

### 1.4.4 计算机系统的性能评价

用于评价计算机系统性能的程序称为评测程序（**benchmark**）。评测程序能够揭示计算机系统对于某类应用的优势或不足。五种评测程序，评测的准确度依次递减。

真实的程序

修改过的程序

程序内核

小型基准程序

综合基准程序



## 1.4 计算机设计的量化准则

把应用程序中用得最频繁的那部分核心程序作为评价计算机性能的标准程序。称为基准程序 (**benchmark**)。

### 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, 各种数据类型和数据区,  
各种控制语句, 过程调用和参数传送, 整数运算和逻辑操作。



## 1.4 计算机设计的量化准则

**VAX-11/780**的测试结果为每秒  
**1757个Dhrystones**，即：**1VAX  
MIPS=1757 Dhrystones/Second**

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**（**Million Floating Point Operations Per Second**）表示；  
**GFLOPS**、**TFLOPS**



## 1.4 计算机设计的量化准则

### 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。
- 测试结果用**Kwips**表示。



## 1.4 计算机设计的量化准则

### **SPEC基准测试程序 (System performance evaluation Cooperative)**

由**30**个左右世界知名计算机大厂商所支持的非盈利的合作组织，包括：**IBM、AT&T、BELL、Compaq、CDC、DG、DEC、Fujitsu、HP、Intel、MIPS、Motolola、SGI、SUN、Unisys**等；



## 1.4 计算机设计的量化准则

**SPEC**能够全面反映机器的性能，具有很高的参考价值；

以**VAX-11/780**的测试结果作为基数。

**SPEC1.0** 1989年10月宣布，程序量超过**15**万行，包含**10**个测试程序，**4**个定点程序，**6**个浮点程序；测试结果用**SPECint'89**和**SPECfp'89**表示。

**1992年**，又增加**10**个测试程序，共有**6**个定点程序和**14**个浮点程序，测试结果用**SPECint'92**和**SPECfp'92**表示。

**1995年**，推出**SPECint'95**和**SPECfp'95**。

**2000年**，推出**SPECint2000**（**11**个整数基准程序）和**SPECfp2000**（**14**个浮点基准程序）。



## 1.4 计算机设计的量化准则

处理机	SPECint'95	SPECfp'95
PentiumII 400	18.5	13.3
PentiumII 450	18.7	13.7
PentiumIII 500	20.6	14.7
PientiumIII 550	22.3	15.6
Celeron 300A	12.0	9.66
Celeron 333	13.1	10.20
Celeron 366	14.1	10.70
Celeron 400	15.1	11.20
Celeron 433	16.1	11.60
Celeron 466	17.0	12.00





## 1.4 计算机设计的量化准则

### TPC基准程序

**Transaction Processing Council**（事务处理委员会）

成立于**1988**年，已有**40**多个成员；

用于评测计算机的事务处理、数据库处理、企业管理与决策支持等方面的性能。

**1989**年**10**月、**1990**年**8**月和**1992**年**7**月发表了**TPC-A**、**TPC-B**和**TPC-C**。

