

1) 计算机性能和功耗

2019年4月10日 23:06

目的：减少CPU time，可以考虑软件和硬件优化设计--- 算法 程序语言 编译器 指令集 等因素，也可从硬件本身考虑----提高frequency。

然而功率墙阻碍了频率的继续增大，考虑多处理器来提高性能

为什么提高了transistor数量会提高性能

本质公式：CPU time=IC * CPI/ Rate CPI 等于各类指令加权平均

Power= $1/2 * capacity\ load * voltage^2 * frequency$ 其中capacity load = capacity * activity

1. Relative performance: CPU相对性能

Define Performance = $1/Execution\ Time$

“X is n time faster than Y”

Elapsed time VS CPU time

Elapsed time 是运行时间

间和空闲时间.它决定了系统性能

CPU time 不包含I/O，其他时间同上
算法本身有关

除了CPUtime外还包含IO，操作系统时

其不仅和CPU性能有关，还和程序和

Clock period & Clock frequency

CPU Time = No. of Clock Cycles × Clock Period

$$= \frac{\text{No. of Clock Cycles}}{\text{Clock Rate}}$$

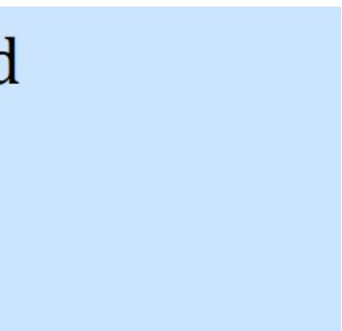
减少时钟周期数，增大频率以提高性能

IC和CPI

IC取决于算法程序、编译器和指令集

单个的指令CPI取决于CPU硬件实现，平均CPI取决于算法程序，语言，编译器，ISA，和CPU硬件实现

CPU time = IC* CPI/ Rate



$$CPI = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^n \left(CPI_i \times \frac{\text{Instruction Count}_i}{\text{Instruction Count}} \right)$$

Performance summary

■ Performance depends on

- ◆ Algorithm: affects IC, possibly CPI
- ◆ Programming language: affects IC, CPI
- ◆ Compiler: affects IC, CPI
- ◆ Instruction set architecture: affects CPI

2. 功耗Power:

$$\text{Power} = \frac{1}{2} \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

3. 多处理器（并行处理）

由于功率墙的存在，我们无法继续提高frequency了

因此考虑：多处理器

需要显式并行编程 难以针对硬件实现方式进行更优化编程

阿姆达尔定律：估计多处理器状态下的理论提速比上界

4. Benchsutes 衡量performance的标准：基准套件 每个商家都会发布自己的衡量标准
基准套件

标准：SPEC CPU 2006：

给定程序集合 与标准机比较运行时间（I/O除外），求执行时间比的几何平均数

$$\frac{\text{ion Count}_i}{\text{ion Count}})$$

CPI

s IC, CPI, T_c

Frequency

$$\sqrt[n]{\prod_{i=1}^n \text{Execution time ratio}_i}$$

SPEC POWER BENCHSUITE:

$$\text{Overall ssj_ops per Watt} = \left(\sum_{i=0}^{10} \text{ssj}_o \right)$$

5. 谬误:闲置花费能量更少 事实上, 闲置性价比更低

$$\text{ops}_i) / \left(\sum_{i=0}^{10} \text{power}_i \right)$$