

<<支撑处理器的技术>>

图书基本信息

书名：<<支撑处理器的技术>>

13位ISBN编号：9787121180927

10位ISBN编号：7121180928

出版时间：2012-10

出版时间：电子工业出版社

作者：(日) Hisa Ando

页数：342

字数：500000

译者：李剑

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<支撑处理器的技术>>

前言

关于本书 2010年,全世界每天都会售出100万台个人计算机,这些计算机的核心就是处理器。其实不仅是个人计算机,汽车、家电,特别是手机中也有处理器的身影。

几乎所有的手机中嵌入的ARM处理器都是由英国的ARM公司开发的,而据ARM公司2010年的报告,当年ARM处理器的累计销售总量超过了200亿个。

仅ARM一家公司,每年就会生产40亿个ARM处理器,这个数值超过全世界人口的一半。

所以,我们生活的世界,是由地球总人口好几倍多的处理器支撑着的。

此外,Google这类搜索引擎、银行ATM、铁路运管系统等社会基础设施也离不开处理器,我们无法想象没有这些系统的生活。

回顾历史,人们从古代就为了实现简单、高速的计算而花费了很大精力。

从算盘等计算辅助工具,到使用齿轮制作机械式加法器和乘法器,后来又采用电子线路代替齿轮以实现更快的速度。

第一台计算机ENIAC于1946年完成并投入使用,它是个像房子那么大的庞然大物,每秒能进行5000次计算,耗电量约为150kW。

与之相比,现代微处理器的性能为每秒100多亿次以上计算,个人计算机的整体最大耗电量约为300W。

这就是说,计算性能提升了200万倍,耗电量下降到了1/500。

其实,从最初的电子计算机到现在仅用了60年时间。

细数飞跃进步的原因,功劳最大的当然要数从真空管到晶体管再到LSI (Large Scale Integration, 大规模集成电路)的电路技术的进步。

而另一方面,追求高速运行的计算机体系结构的进步也为其做出了巨大贡献。

尽管处理器性能一直在飞速发展,但最近由于耗电量的限制,单核心的性能提高速度变慢了。

人们开始从多线程、多核心化方面提升性能,并通过虚拟机、安全等技术努力提高可用性。

另外,不仅是传统的计算机,智能手机、上网本等移动设备也使用了强大的处理器,由于电池续航能力的要求,处理器还在朝着节能方向发展。

如上所述,处理器在持续发展以适应时代与需求的变化。

个人计算机被贴上了“Intel Inside”的标志,就连对硬件技术毫无兴趣的人也知道微处理器,至少知道处理器这个名字。

另一方面,随着科技的发展,处理器也变得极端复杂,其构造与使用技巧很难得到广泛的认知。

但是,是理解处理器的构造和运行原理,还是把它当做黑盒使用而不知其所以然,两者的使用效果难道没有差别吗?

就像开车,要是只想把车开走,当然不必了解汽车的结构及结构带来的性质差异;但要是能够理解这些,不就能开得比较“拉风”,或比较省油了?

本书主要以希望提高计算机系统运行效率的程序员为对象,讲解处理器中使用的高性能化技术与构造,并介绍怎样写程序才能让处理器高效运行。

提到处理器的设计,有些人可能会联想到与门、或门等逻辑电路之类的硬件设计,但本书内容处于逻辑电路的上层,主要以“微体系架构”和“指令架构”这些程序员使用的接口为中心,介绍处理器的结构与运行原理。

处理器处于各种计算机的核心位置,如今在移动设备乃至社会基础设施中都扮演着举足轻重的角色。

希望本书能在理解处理器的发展、熟练掌握处理器方面助你一臂之力。

2010年11月 Hisa Ando 本书的结构 本书以程序员为主要对象,以通俗易懂的方式说明处理器的结构和使用技巧。

以下是各章内容。

第1章 处理器与计算机系统的基础 第1章作为绪论,通俗易懂地介绍了处理器、使用处理器的计算机的概要及术语,以便初次接触处理器、计算机硬件的读者也能够理解第2章以后的内容。

<<支撑处理器的技术>>

第2章 处理器发展史 第2章 说明计算机的发展史，以及那些提高处理器性能、简化使用方法的技术。

本章的目标是让读者理解处理器的发展历程和采用技术的概要。

第3章 [详解] 面向程序员的处理器架构 第4章 虚拟化支持 第5章 多处理器的出现和普及

第3章到第5章是本书的中心。

第3章具体介绍流水线处理、缓存、乱序执行等提高处理器速度的技术，以及内存管理、虚拟化、多媒体处理等扩大处理器使用范围的技术。

我们将给出程序例子，通俗易懂地介绍程序员应当如何高效地使用处理器。

另外，第3章还介绍了个人计算机和服务中广泛使用的Intel处理器的构造。

第4章说明最近的云计算系统等必不可少的虚拟化，以及虚拟化的实现机制，与提高虚拟机系统效率的注意事项。

第5章就最近流行的多线程、多核心处理器，介绍其出现的背景和使用的技术，介绍高效利用多核心处理器时的要点。

第6章 处理器周边技术 第6章 介绍处理器周边技术，它与处理器同样是计算机系统的重要构成因素，包括负责存储的内存和处理器相连的输入/输出设备接口的结构与运作方式。

第7章 GPGPU与超并行处理 第7章 着眼于最近以其高计算处理能力而备受关注的GPU (Graphics Processing Unit)、GPGPU (General Purpose GPU)，本章介绍其结构，以及将它作为计算引擎时的注意事项。

第8章 处理器的发展趋势 最后在第8章中以今后处理器的发展为焦点，探讨技术的发展方向和各种用途的处理器的发展方向。

中文版书中切口处的“ ”表示原书页码，便于读者与原日文版图书对照阅读，本书索引所列页码为原日文版页码。

<<支撑处理器的技术>>

内容概要

本书用通俗易懂的语言和大量的插图，介绍了处理器的历史、基本结构、实现原理等，还对时下流行的虚拟化技术、多任务、多核心、GPGPU等进行了全面的讲解，并介绍了有效利用处理器的各种功能来提高应用程序性能的方法。对于普通读者来说，这是本学习知识的科普读物；对于程序员来说，这是一把打开现代处理器宝库的钥匙。

本书最后还介绍了处理器在移动设备、汽车、家电等方面的应用，并展望处理器的未来发展趋势，希望能对相关硬件的开发者有所帮助。

<<支撑处理器的技术>>

作者简介

Hisa Ando, 拥有40年尖端处理器的开发经验。
曾作为SPARC架构委员会委员参与了SPARC
V9架构的开发, 后作为HAL公司副总裁兼处理器开发部总经理参与了SPARC64的开发。
后来又担任过富士通株式会社计算机事业本部的技师长等职务。
现在作为技术作家, 主要从事处理器、超级计算机相关的报道和介绍等。
此外, 还担任了几所大学的客座讲师, 负责计算机架构的课程。
拥有工学博士学位。

<<支撑处理器的技术>>

书籍目录

第1章 处理器与计算机系统的基础

1.1 计算机的结构

计算机组成部分——处理器、内存、输入/输出设备

专栏 处理器封装的物理结构

计算机使用二进制进行计算

专栏 2值和多值 非二进制不可吗？

不一定！

处理器——计算机的大脑，负责解释程序

专栏 处理器、微处理器的定义 计算机最重要的组件

——处理器

专栏 流水线级

专栏 表示数量级的前缀

内存——存储计算机的指令和数据

输入/输出设备——计算机的眼睛、耳朵和嘴巴

北桥和南桥——逐渐淡化的称呼

1.2 支撑计算机高速化的半导体技术

摩尔（Moore）定律——更多的晶体管，更高的并行度

缩放定律——晶体的性能提升

源源不断的开发投资带来的半导体微型化——微型化的步伐

还将长期维持

提升性能的三大支柱——提高频率、并行处理和功能扩展

1.3 计算机和数据表示方法

字节、半字、字……——比特组及用途

文字编码

专栏 日语与文字编码

无符号整数与有符号整数、反码与补码表示法

浮点小数——IEEE 754标准

Big Endian与Little Endian

1.4 处理器和指令

指令架构入门

机器指令——处理器的指令

地址空间

专栏 AMD和Intel的64位扩展架构

1.5 机器指令程序编写方法

用汇编语言编程

利用编译器进行高级语言编程

解释语言编程

1.6 小结

专栏 半导体的微型化

第2章 处理器发展史

2.1 计算机之前的计算设备

算盘——最早的计算工具

纳皮尔算筹（Napier ' s bones）——乘法辅助工具

机械式计算机——Schikard计算机、Pascaline

<<支撑处理器的技术>>

巴贝奇的差分机——使用齿轮的超级计算机

2.2 最早的电子计算机

最早的电子计算机——ABC和ENIAC

FUJIC——日本最早的电子计算机

2.3 处理器组成元件的变迁

根据主要部件划分计算机的时代

第1代：真空管

第2代：晶体管

第3代：集成电路

第4代：大规模集成电路（VLSI）

VLSI处理器的元件数目和时钟频率的发展趋势

2.4 指令架构的变迁

指令架构的发展之路

存储程序的计算机——程序也从内存中读入

虚拟内存——更加丰富、取之不尽的内存

多任务——需要改变程序的部署

分时系统与内存管理设施的出现

专栏 MULTICS之后的事

内存管理机构、特权态——多用户需要解决安全问题

指令架构扩展——指令架构的确立与指令兼容性的实现

2.5 微架构的发展

微架构的发展之道

流水线处理——通过流水线寄存器有效利用硬件

运算器的高速化——整数运算器、浮点数运算器

高速缓存——解决内存访问问题的锦囊

RISC的出现——RISC与CISC

超标量执行——一个周期内执行多条指令

乱序执行——改变指令顺序以提高性能

专栏 乱序执行（Out of Order）：请注意！

？

分支预测与预测执行——“条件分支指令”对策

多核心——耗电量的限制推动了多核心的发展

2.6 处理器的用途越来越广泛

活跃在各种舞台的处理器

节能、高可靠技术、实时性——各种各样的需求

2.7 小结

第3章 [详解] 面向程序员的处理器架构

3.1 支撑微架构的技术

流水线执行的原理——并行处理指令

高速缓存的构造——提高内存访问速度

RISC与CISC——定长指令与变长指令

提高运算器的速度——处理器中因处理复杂而处理时间长的组件之一

超标量执行的原理——一个周期内并行执行多条指令

乱序执行的原理——减轻数据冒险的影响

分支预测的原理——降低控制冒险造成的损失

<<支撑处理器的技术>>

现代处理器会消除或减轻各种冒险以提高性能
内存、I/O与输入/输出接口
性能计数器——性能分析器和处理器内部执行状况的信息
3.2 架构扩展扩大了处理器的使用范围
多任务和内存管理机构
中断处理机构
专栏 异常、中断、陷阱——术语整理
虚拟化支持——内存访问和内存管理机构
多媒体、加密等支持——需要大量数据的计算
3.3 x86 Nehalem架构处理器
x86的指令体系和Intel 64架构
Core i7处理器的结构
内存管理采用4级页表
新的处理器接口QPI
3.4 小结
第4章 虚拟化支持
4.1 虚拟化的目的和优缺点
虚拟化的基础知识（复习）
虚拟化的目的
实现用户间强隔离——虚拟化的优点
集合多台服务器，提高利用率——虚拟化的优点
VMM运行时的额外开销——虚拟化的缺点、注意点
4.2 实现虚拟化的条件
为操作系统提供独立的（虚拟）硬件的VMM
4.3 支撑虚拟化的硬件设施
检测硬件操作命令
硬件状态的保存和还原——切换虚拟机
双重地址转换、TLB
I/O的虚拟化
实时迁移（Live Migration）
4.4 小结
专栏 虚拟化的前世今生
第5章 多处理器的出现和普及
5.1 多线程处理器
线程、多线程的纷繁芜杂（!）——首先总结一下术语
多线程的两种方式
VMT的原理——短时间内切换线程
SMT的原理——混杂执行多个线程的指令
专栏 历史悠久的SMT
SMT必需的机制
多线程的效果如何——通过Windows任务管理器查看效果
5.2 多处理器系统
多处理器、多核心是什么意思
多核心处理器的结构
专栏 众核处理器（Many-core processor）的结构
缓存一致性控制——多处理器之间缓存的一致性
多插槽系统

<<支撑处理器的技术>>

专栏 插槽还是芯片

专栏 多核心时代的处理器、CPU的含义

提高多处理器系统的性能——问题和解决方案

共享内存系统和分布式内存系统

5.3 小结

第6章 处理器周边技术

6.1 内存技术

内存历史概览

DRAM内存的工作原理——利用电荷存储信息

DRAM芯片和内存DIMM

DRAM芯片的内部结构——内存单元阵列、行/列解码器、

检测放大器、Bank

处理器和DIMM的连接

内存系统的错误处理

6.2 输入/输出设备的连接

处理器的I/O连接

PCI总线

专栏 通向PCI总线之路

Intel Core i系列处理器的I/O结构

Intel PCH的I/O控制结构

6.3 小结

专栏 DRAM的刷新

第7章 GPGPU和超并行处理

7.1 GPGPU的原理

3D图形和GPU——需要大量计算

GPU系统

从GPU到GPGPU

作为“超并行SIMD处理器”的GPGPU

作为“超多线程处理器”的GPGPU

专栏 Warp !

GPGPU的内存层次结构

GPGPU中的多线程需要分割使用通用寄存器

CPU和GPGPU的巨大区别

7.2 GPGPU编程

现代GPGPU编程

CUDA的运行模型——线程块、网格

CUDA中的函数声明和变量声明

OpenCL编程

如何发挥GPGPU的性能

7.3 小结

专栏 Top500和GPU计算

第8章 处理器的发展趋势

8.1 耗电量是决定因素

为何处理器要消耗电力

节能技术的发展

8.2 更高、更快、更强

<<支撑处理器的技术>>

半导体细微化技术

如何有效利用增加的晶体管

利用部分晶体管降低生产成本

有效利用部分块不合格的芯片

8.3 更可靠、更安全的处理器设计

为何处理器会有错误行为——故障、噪声

防止错误行为，确保安全运转

8.4 未来处理器的发展方向

无处不在的处理器

家电用处理器

汽车用处理器

个人计算机用处理器

智能手机用处理器

服务器用处理器

8.5 小结

索引

<<支撑处理器的技术>>

章节摘录

版权页：插图：流水线处理——通过流水线寄存器有效利用硬件 第1章说过，处理器以流水作业的方式处理指令。

但是，第1代处理器元件很少，无法实现流水作业，只能顺序完成指令读取到写回结果这一连串操作，然后再开始读取下个指令。

负责指令读取、指令解释、操作数（指令运算对象）读取、运算、写回结果的硬件各自独立，执行一个操作时，负责其他功能的单元处于无事可做的状态。

这很浪费，于是人们涉及了流水线处理方式，将各个功能单元的处理结果保存在寄存器中，下个时钟周期开始处理下一条指令。

这样，每个时钟周期都可以开始处理新指令，与顺序执行相比，性能得到了大幅度提升。

流水线处理方式就是流水作业，就像原油的管道（pipeline）一样，可以把一连串的功能单元看成管道，而指令就相当于原油在这些管道中流动，流水线处理也因此而得名。

流水线处理的实现难点——随着元件数目增加，流水线处理得到普及 流水线处理能大幅度提升性能，但实现起来也有难点。

例如，如果下一条指令要使用上一条指令的计算结果，那么前一条指令写回结果之前，下一条指令就无法开始；遇到分支指令，下一条指令就会发生跳转，从而必须读取跳转目的地的指令。

这些情况都无法进行流水线处理，需要检测这些情况并暂时中断流水作业。

因此，要实现流水线处理，就要在功能单元之间增加流水线寄存器，还要添加上述控制电路，整体来看需要的元件数会大幅度增加。

由于这些原因，直到1961年IBM开发的Stretch（IBM7030）超大型计算机才第一次采用了流水线处理。

不过，随着可用元件数的增加，小型处理器也开始广泛应用流水线处理。

运算器的高速化——整数运算器、浮点数运算器 处理器的运算对象大致可分为整数和浮点数。

整数有时表示数字，有时只是当做比特序列使用。

另一方面，如第1章所述，浮点数包含尾数部分和指数部分，能够表示非常大和非常小的数。

整数运算包括加减乘除运算，以及把整数当做比特序列的AND、OR、NOT等逻辑运算，左右移位处理等运算。

加减法、逻辑运算一般由一个运算器执行，这个运算器称为ALU（Arithmetic Logic Unit，算术逻辑运算单元）。

乘除法与加减法差别很大，因此分别采用独立的运算器。

此外，有的处理器把移位运算器作为ALU的一部分，不过最近也有移位运算器独立于ALU的情况。

<<支撑处理器的技术>>

编辑推荐

《支撑处理器的技术:永无止境地追求速度的世界》对于普通读者来说，是本学习知识的科普读物；对于程序员来说，是一把打开现代处理器宝库的钥匙。

<<支撑处理器的技术>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介, 请支持正版图书。

更多资源请访问:<http://www.tushu007.com>