

<<64位微处理器系统编程和应用编程>>

图书基本信息

书名：<<64位微处理器系统编程和应用编程>>

13位ISBN编号：9787302207757

10位ISBN编号：7302207755

出版时间：2009-11

出版时间：周明德 清华大学出版社 (2009-11出版)

作者：周明德

页数：725

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>

前言

电子计算机是人类在20世纪最重大的科技创造之一。

自20世纪40年代诞生第一台用电子管制造的现代电子计算机以来,电子计算机的主要元件先后经历了真空管、晶体管、中小规模集成电路、大规模超大规模集成电路4代。

超大规模集成电路的发展,把计算机的主体——中央处理单元(CPU)集成在一个芯片上,称为微处理器,使计算机进入了微型计算机时代。

自从1981年IBM公司进入了微型计算机领域并推出了IBM—PC以后,计算机的发展开创了一个新的时代——微型计算机时代。

微型计算机的迅速普及,使计算机真正广泛应用于工业、农业、科学技术领域以及社会生活的各个方面。

以前的大型机(MainFrame)、中型机、小型机的界线已经日益模糊与消失。

随着微型机应用的普及及技术的发展,芯片与微型机的功能和性能迅速提高,其功能已经远远超过了20世纪80年代以前的中小型机甚至超过了大型机。

到了20世纪90年代,随着局域网、广域网、城域网以及Internet的迅速普及与发展,微型计算机从功能上可分为网络工作站(客户端——Client)和网络服务器(Server)两大类型。

网络客户端又称为个人计算机(台式PC或笔记本PC)。

在个人计算机中,其核心是中央处理单元(CPU)。

Intel公司的芯片在个人计算机中占据了统治地位(约占80%)。

形成了个人计算机芯片的主流——x86系列结构。

x86系列结构从8位的8088、16位的8086发展到32位的80386、80486直至奔腾4(P4)以及最近推出的双核心、四核心的处理器,其功能与性能都有近千倍的提升。

由于广泛的应用,x86系列结构已成了事实上的工业标准。

其他厂商,例如AMD也推出依从这样的体系结构的兼容的CPU,并得到了大量应用。

网络时代的来临、多媒体信息的数字化等,都使信息爆炸般增长。

信息的存储、处理、交换都强烈地促进微处理器向64位时代过渡。

当前主流的64位处理器是AMD公司的x86—64系列,它是与32位x86处理器兼容的,得到了迅速的发展和广泛的认可与应用。

为此,Intel公司也推出了与IA—32(32位x86系列处理器)兼容的Intel64位体系结构处理器。

这两种64位处理器,无论从应用编程还是系统编程的角度来看都是一致的,所以作者把这两者综合起来统一进行介绍。

<<64位微处理器系统编程和应用编程>>

内容概要

《64位微处理器系统编程和应用编程》综合了Intel公司和AMD公司的x86系列的64位处理器的应用和系统编程，是作者《微型计算机系统原理及应用（第五版）》一书向64位处理器的扩展与提高。反映了当代微处理器的最新发展。

《64位微处理器系统编程和应用编程》详细介绍了64位处理器的应用编程和64处理器体系结构系统编程的具体内容，对于应用编程和系统编程的开发者都是十分重要和有用的。

《64位微处理器系统编程和应用编程》适合希望了解和应用64位处理器的科技人员使用，也可作为高等院校相关课程的教材。

作者简介

周明德，教授，1959年毕业于清华大学计算机专业。

毕业后留校。

讲授过“脉冲技术”、“微型计算机原理及应用”等课程。

在全国普及微型计算机的高潮中，应邀在全国各地讲授了数十次“微型计算机原理”课程，并制作了录像带在各地播放，受到了极高的评价，为普及微型计算机起到了重要作用。

曾任中国计算机技术服务公司总工程师，为电子部有突出贡献专家，享受政府津贴。

主持了国家“八五”、“九五”重点科技攻关项目“国产操作系统开发”，任副总设计师。

此项目获电子部科技进步特等奖，国家科技进步二等奖。

著有《微型计算机系统原理及应用》、《64位微处理器应用编程》、《64位微处理器系统编程》、《UNIX / Linux核心》等书，共约三百余万字。

累计发行三百余万册。

书籍目录

第1章 引言1.1 符号约定1.1.1 位与字节顺序1.1.2 保留位与软件兼容性1.1.3 十六进制与二进制数1.1.4 分段的寻址1.1.5 对CPUID、CR和MSR值的新句法1.1.6 异常1.2 Intel 64和IA-32体系结构的简要历史1.2.1 16位处理器与分段1.2.2 Intel 286处理器1.2.3 Intel 386处理器1.2.4 Intel 486处理器1.2.5 Intel Pentium(奔腾)处理器1.2.6 处理器的P6族(系列)1.2.7 Intel Pentium 4处理器系列1.2.8 Intel Xeon处理器1.2.9 Intel Pentium M处理器1.2.10 Intel Pentium处理器极品版1.2.11 Intel Core Du0和Intel Core Sol0处理器1.2.12 Intel Xeon处理器5100系列和Intel Core 2处理器系列1.3 更多的关于特定的先进技术1.3.1 P6族微体系结构1.3.2 Intel NetBurst微体系结构1.3.3 Intel Core微体系结构1.3.4 SIMD指令1.3.5 超线程技术1.3.6 多核技术1.3.7 Intel 64体系结构1.3.8 Intel虚拟化技术第2章 基本执行环境2.1 操作方式2.1.1 Intel 64体系结构2.1.2 x86—64的操作方式2.2 基本执行环境概要2.2.1 64位方式执行环境2.3 内存组织2.3.1 IA-32方式2.3.2 分页和虚拟存储器2.3.3 64位方式的内存组织2.3.4 操作方式与内存模型2.3.5 32位和16位地址与操作数尺寸2.3.6 在保护方式扩展的物理寻址2.3.7 在64位方式的地址计算2.4 基本程序执行寄存器2.4.1 通用寄存器2.4.2 段寄存器2.4.3 EFLAGS寄存器2.5 指令指针2.6 操作数尺寸和地址尺寸属性2.7 操作数寻址2.7.1 立即数2.7.2 寄存器操作数2.7.3 存储器操作数2.7.4 规定段选择子2.7.5 规定偏移量2.7.6 汇编程序和编译器寻址方式2.7.7 I/O端口寻址第3章 数据类型3.1 基本数据类型3.2 数字数据类型3.2.1 整数3.2.2 浮点数据类型3.3 指针数据类型3.4 位字段数据类型3.5 串数据类型3.6 组合的SIMD数据类型3.6.1 64位SIMD组合的数据类型3.6.2 128位组合的SIMD数据类型3.7. BCD和组合的BCD整数3.8 实数和浮点格式3.8.1 实数系统3.8.2 浮点格式3.8.3 实数和非数字编码3.8.4 舍入3.9 浮点异常概要3.9.1 浮点异常条件3.9.2 浮点异常优先权3.9.3 浮点异常处理程序的典型动作第4章 Intel64(AMDx86.64)和IA-32系统体系结构概要4.1 系统级体系结构概要4.1.1 全局和局部描述符表4.1.2 系统段、段描述符和门4.1.3 任务状态段和任务门4.1.4 中断和异常处理4.1.5 存储管理4.1.6 系统寄存器4.1.7 其他系统资源4.2 操作方式4.3 在EFLAGS寄存器中的系统标志和字段4.4 存储管理寄存器4.4.1 全局描述符表寄存器(GDTR)4.4.2 局部描述符表寄存器(LDTR)4.4.3 中断描述符表寄存器(IDTR)4.4.4 任务寄存器(TR)4.5 控制寄存器4.6 系统指令概要4.6.1 装入和存储系统寄存器4.6.2 验证访问权4.6.3 装入和存储调试寄存器4.6.4 无效缓存和TLB4.6.5 控制处理器4.6.6 读性能监视和时戳计数器4.6.7 读和写模型特定寄存器第5章 保护方式存储管理5.1 存储管理概要5.2 使用分段5.2.1 基本平面模型5.2.2 保护的平面模型5.2.3 多段模型5.2.4 在IA-32e方式中的分段5.2.5 分页和段5.3 物理地址空间5.4 逻辑和线性地址5.4.1 在IA-32e方式中逻辑地址的转换5.4.2 段选择子5.4.3 段寄存器5.4.4 在IA-32e方式中的段装入指令5.4.5 段描述符5.5 系统描述符类型5.5.1 段描述符表5.5.2 在IA-32e方式中的段描述符表5.6 分页(虚拟存储器)概要5.6.1 分页选项5.6.2 页表和页目录5.7 使用32位物理寻址的页转换5.7.1 线性地址转换(4KB页)5.7.2 线性地址转换(4MB页)5.7.3 混合4KB和4MB页5.7.4 内存混合5.7.5 页目录的基地址5.7.6 页目录和页表项5.7.7 页目录和页项不存在5.8 使用PAE分页机制的36位物理寻址5.8.1 增强的传统PAE分页5.8.2 具有PAE启用的(4KB页)的线性地址转换5.8.3 具有PAE启用的(2MB页)的线性地址转换5.8.4 访问具有扩展的页表结构的全扩展物理地址空间5.8.5 具有扩展的地址启用的页目录和页表5.9 使用PSE-36分页机制的36位物理寻址5.10 在IA-32e方式中PAE启用的分页5.10.1 IA-32e方式线性地址转换(4KB页)5.10.2 IA-32e方式线性地址转换(2MB页).....第6章 内存缓存控制第7章 保护第8章 中断和异常处理第9章 指令集小结第10章 用x87 FPU编程第11章 64位和128位媒体与科学编程第12章 用SSE3和追加的SSE3编程第13章 任务管理第14章 调试与性能监视第15章 先进的可编程的中断控制器第16章 多处理器管理第17章 处理器标识及特性第18章 处理器管理和初始化参考文献

章节摘录

插图：此流水线设计以高速、流水线的微处理器解决公共问题。

这些问题中的两个构成主要的延迟源。

从目标读取的指令的译码时间；由于在缓存行的中间的分支目标，浪费了译码带宽。

流水线的追踪缓存的操作解决这些问题。

指令由转换引擎（取指 / 译码逻辑部分）经常地读取和译码并构造为称为追踪的微操作序列。

在任何时候多追踪（代表预取分支）存放在追踪缓存中。

为了跟在有效分支后的指令，搜索追踪缓存。

若指令也作为预取分支中的第一条指令出现，从内存层次中读取和译码指令，且预取的分支变为指令的新的源。

追踪缓存和转换引擎共同操作分支预测硬件。

分支目标基于它们的线性地址用分支目标缓冲器（BTB）被预测，尽可能快地被读取。

2. 超序执行核超序执行核超序执行指令的能力是允许并行的一关键因素。

此特性允许处理器重排序指令，因此，一个微操作被延迟围绕它的其他微操作可能处理。

处理器使用若干缓冲器以平滑微操作流。

核心设计以推动平行执行。

每个周期它能分派多至6个微操作（这超过了追踪缓存和撤出微操作带宽）。

大多数流水线每个周期能开始执行一个新的微操作，所以对于每个流水线，若干个指令同时在处理中。

一些算术逻辑单元（ALU）指令每周期能开始两个；许多浮点指令每两个周期启动一个。

3. 撤出单元撤出单元接收从超序执行核执行的微操作的结果并处理此结果，所以，体系结构状态按照原始的程序顺序更新。

当一微操作完成和写它的结果时，它被撤出。

每个周期可能撤出多至3个微操作。

重新排序缓冲器（ROB）是处理器中缓冲完成的微操作、按序更新体系结构状态和管理执行顺序的单元。

撤出节也保持分支的踪迹和送更新的分支目标信息至BTB。

然后清除不再需要的预取的踪迹。

编辑推荐

《64位微处理器系统编程和应用编程》特色针对当前应用最广泛的两种64位处理器——Intel和AMD的四核处理器作为基础进行介绍分析。

反映了当代微处理器的最新发展。

把Intel和AMD两种处理器综合起来进行分析。

不论是应用编程还是系统编程，《64位微处理器系统编程和应用编程》的内容对两种处理器都适用。

应用编程全面而完整：既有通用编程，又有利用x86处理器的浮点和数学运算编程以及利用MMX技术和SSE2、SSE3、SSSE3的多媒体和科学计算的应用编程。

《64位微处理器系统编程和应用编程》包括了应用编程的详尽内容，也包含了系统编程的主要内容，对于绝大多数应用者自如地编程和应用是足够的。

《64位微处理器系统编程和应用编程》立足于应用。

给出了丰富的编程实例。

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>