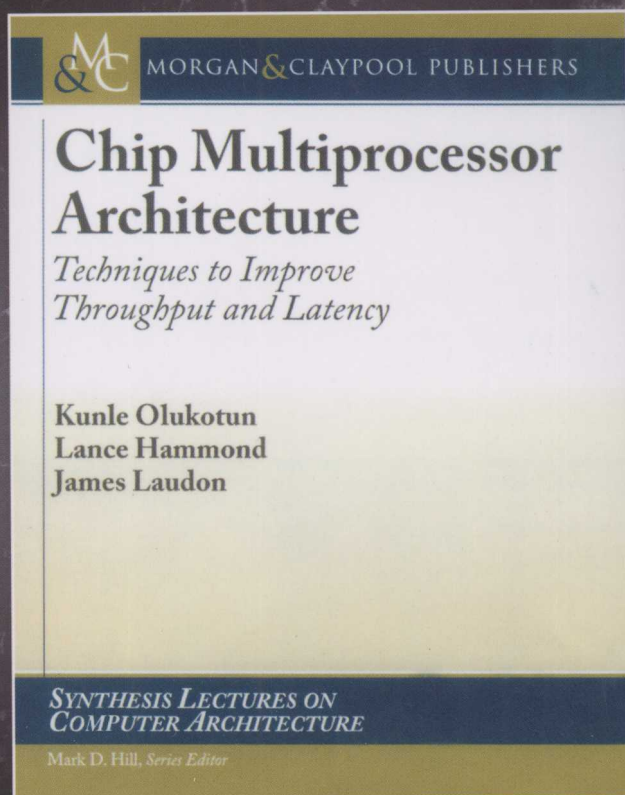


片上多处理器体系结构 改善吞吐率和延迟的技术

(美) Kunle Olukotun Lance Hammond James Laudon 著 汪东升 王海霞 李鹏 等译
斯坦福大学 斯坦福大学 Sun公司 清华大学



Chip Multiprocessor Architecture

Techniques to Improve Throughput and Latency



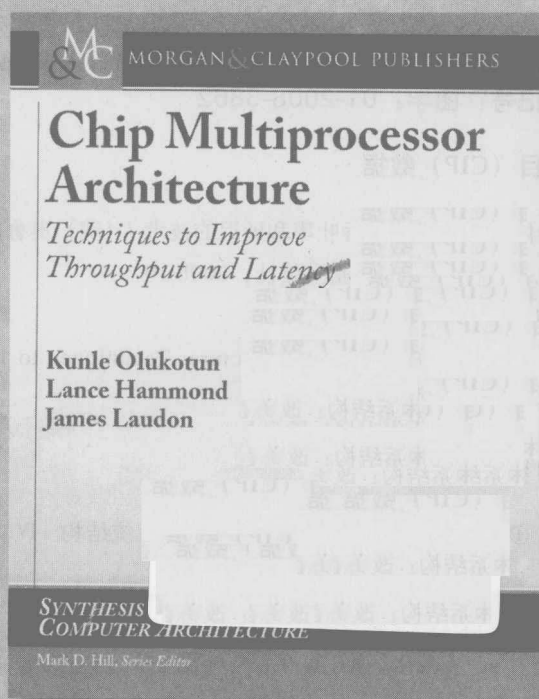
机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

片上多处理器体系结构

改善吞吐率和延迟的技术

(美) Kunle Olukotun Lance Hammond James Laudon 著 汪东升 王海霞 李鹏 等译
斯坦福大学 斯坦福大学 Sun公司 清华大学



Chip Multiprocessor Architecture
Techniques to Improve Throughput and Latency



机械工业出版社
China Machine Press

片上多处理器 (chip multiprocessor), 又称多核微处理器或简称CMP, 已成为构造现代高性能微处理器的唯一技术途径。本书在简单介绍了片上多处理器的基本概念后, 着重于从提高吞吐率和缩短响应时延两方面探讨片上多处理器的基本技术与设计方法。同时介绍了多核处理器的编程技巧, 包括线程级猜测和事务型内存等热点技术。通过阅读本书, 读者可以在较短时间内熟悉和掌握片上多处理器研究的主流技术和最新的研究成果, 为片上多处理器领域的科研和应用带来新的思路和灵感。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2008-3862

图书在版编目 (CIP) 数据

片上多处理器体系结构: 改善吞吐率和延迟的技术 / (美) 奥鲁克顿 (Olukotun, K.) 等著; 汪东升等译. —北京: 机械工业出版社, 2008.10

(计算机科学丛书)

书名原文: Chip Multiprocessor Architecture: Techniques to Improve Throughput and Latency

ISBN 978-7-111-25381-5

I. 片… II. ①奥… ②汪… III. 微处理器—系统结构 IV. TP332

中国版本图书馆CIP数据核字 (2008) 第160892号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 杨庆燕

三河市明辉印装有限公司印刷·新华书店北京发行所发行

2009年1月第1版第1次印刷

184mm×260mm · 10.25印张

标准书号: ISBN 978-7-111-25381-5

定价: 26.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线 (010) 68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章分社较早意识到“出版要为教育服务”。自1998年开始，华章分社就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brain W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



译者序

多核处理器是处理器发展的必然趋势。无论是移动/嵌入式应用、桌面应用还是服务器应用，都将采用多核的架构。然而多核处理器的研发和应用还有问题，诸如多核处理器设计、验证和调试、设计空间、编程模型等挑战性问题还有待于深入的探讨。本书是讲授多核体系结构设计和优化的第一本著作。

本书在简单介绍了片上多处理器的基本概念后，着重从提高吞吐量和缩短响应时延两方面探讨片上多处理器的基本技术与设计方法。本书还介绍了多核处理器的编程技巧，包括线程级猜测和事务型内存等热点技术。通过阅读本书，读者可以在较短时间内熟悉和掌握片上多处理器研究的主流技术和最新的研究成果，为片上多处理器领域的科研和应用带来新的思路和灵感。

本书的作者Kunle Olukotun博士是美国斯坦福大学计算机系教授，是当今计算机设计领域著名的学者和开拓者。他领导了第一个片上多处理器系统Hydra的开发，推动了片上多处理器的研究热潮。同时，他创建了Afara Websystems公司，该公司最终被Sun公司收购并将其处理器重命名为Niagara。Lance Hammond是事务型缓存一致性与内存一致性(TCC)的主要设计者之一。Lames Laudon提出了交叉多线程(interleaved multithreading)的概念并在UltraSPARC T1中得以应用。

感谢机械工业出版社对出版本书的支持，感谢清华大学微处理器与SoC技术研究中心的顾瑜、嵩天（现在北京理工大学）、郭三川和李崇民等博士在翻译和校对过程中所付出的辛勤努力。由于时间仓促及译者水平有限，文中难免有不当之处，还望读者批评指正。

汪东升

2008年9月于清华园

摘要

由于受诸多因素的影响,片上多处理器(chip multiprocessor),又称多核微处理器或简称CMP,已成为构造现代高性能微处理器的唯一技术途径。传统超标量指令发射技术无法从典型程序指令流中发掘出足够多的并行性,使得单核微处理器的性能再无法有效扩展。另外,处理器的功耗与散热问题日趋严重,除非使用水冷散热系统,否则很难继续依靠简单提高时钟频率的方法来改善微处理器性能。综合这些问题可以得到一个简单结论:目前单个微处理器芯片集成了数量空前的晶体管,如果继续保持每年或两年就完成新一代更大规模处理器的设计与调试工作,将会使成本直线上升以至无法完成设计工作。

CMP通过在单个芯片中放入多个结构相对简单的处理器内核而不是使用一个巨大的处理器内核避免了上述问题。CMP内核既可采用简单流水线结构,又可以使用中度复杂的超标量处理器,无论选定哪种内核,CMP系统都可以随着半导体工艺的进步,在每一代新版处理器芯片中加入更多数量的高速处理器内核,来有效扩展处理器性能。此外,并程序将多线程任务分发给CMP系统中几个内核并行执行,与单内核处理器相比可以取得显著的性能提升。尽管在许多实用的应用负载中并行线程已经非常常见,但仍有一些很重要的应用负载很难被划分为几个线程来并行执行。与传统多处理器系统相比,CMP系统中内核之间的通信延迟更低,从而使更多应用负载适于并行执行。但是某些关键应用中缺乏足够的并行度,可能成为这些系统中推广CMP应用的主要障碍。

通过对比CMP与传统单处理器的优缺点,本书仔细研究了如何针对两种常见但截然不同的工作负载来更好地设计CMP:并行度较高且对吞吐率敏感的应用和并行度较低且对延迟敏感的应用。对吞吐率敏感的应用,如可迅速处理多个独立事务的服务器工作负载,需要综合考虑CMP中所有可能限制吞吐率的部件,如处理器核心、片上缓存和片外存储器接口。书中展示了在几个研究与实例系统(如Sun Niagara)中如何进行设计折中。对延迟较敏感的应用,如桌面应用,其关注的焦点是如何降低内核之间的通信延迟,以及如何帮助程序员简化已有程序代码多线程化。本书介绍了多种可用于CMP系统的并行编程简化技术,并重点讨论了斯坦福大学在该领域的相关研究工作。为了证明CMP的潜在优势,书中给出了一些典型例子来加以说明。本书的其它关注焦点还包括线程级猜测(thread-level speculation,简称TLS)和事务型内存(transactional memory)。线程级猜测是一种将标准的串行程序自动切分成CMP上多个并行线程的方法。事务型内存模型使用硬件而非传统软件锁机制来保证一段指令的

原子代码执行,可以显著简化并行编程工作量,有效降低并行代码的出错概率。

关键字

基本术语:片上多处理器 (CMP),多核微处理器,微处理器功耗,并行处理,基本线程执行。

应用的类别:吞吐率敏感的应用,服务器应用,延迟敏感的应用,桌面应用, SPEC测试程序集, Java应用。

技术:线程级猜测 (TLS), JRPM虚拟机,提取猜测线程的踪迹器 (TEST), 事务型内存,事务型缓存一致性与内存一致性 (TCC),事务型锁消除 (TLR)。

系统名称: DEC Piranha, Sun Niagara, Sun Niagara 2, Stanford Hydra。

目 录

出版者的话
译者序
摘要

第1章 CMP简介	1
1.1 一个新途径：片上多处理器	5
1.2 应用程序的并行性图景	6
1.3 一个简单的例子：超标量与CMP	9
1.4 本书：超越基本的CMP	19
参考文献	20
第2章 吞吐率优化技术	21
2.1 简单内核与服务器应用	24
2.1.1 处理器内部对多线程的需求	25
2.1.2 最大化单个芯片上的处理器 内核数量	27
2.1.3 提供足够的缓存和主存访问 带宽	27
2.2 与吞吐率有关的片上多处理器 案例研究	27
2.2.1 案例1：Piranha服务器CMP	28
2.2.2 案例2：Niagara服务器CMP	37
2.2.3 案例3：Niagara-2服务器CMP	49
2.2.4 基于简单内核的CMP系统的 局限性	52
2.3 通用服务器CMP分析	53
2.3.1 模拟大规模设计空间	53
2.3.2 选择设计要点	56
2.3.3 实验结果	58
2.3.4 讨论	60
参考文献	63

第3章 自动的延迟优化技术	65
3.1 伪并行：“帮手”线程	66
3.2 采用线程级猜测实现 自动并行	68
3.3 TLS系统的一个实例：Hydra	75
3.3.1 Hydra基础设计	76
3.3.2 在Hydra中加入TLS	78
3.3.3 使用冲突统计获得的反馈信息	87
3.3.4 性能分析	92
3.3.5 完全自动化的TLS支持： Jrpm系统	95
3.4 自动并行化的总结思考	107
参考文献	108
第4章 基于手工并行编程的延迟优化 技术	110
4.1 使用TLS支持事务型内存	112
4.1.1 一个例子：使用TLS对堆排序 进行并行化	112
4.1.2 使用TLS对SPEC2000进行并行化	123
4.2 事务型一致性：更普遍的 事务型内存	125
4.2.1 TCC硬件	127
4.2.2 TCC软件	131
4.2.3 TCC性能	137
4.3 事务型内存与常规共享内存的 混合方案	147
参考文献	148
第5章 一个多核的世界：CMP的 未来	150
参考文献	153
作者介绍	154

第1章 CMP简介



微处理器作为现代计算机系统的核心，在过去的许多年中，其性能一直呈指数增长，图1-1所示的Intel处理器就是一个典型例子。微处理器性能快速增长的主要原因有两点。首先，在摩尔定律^[1]的作用下，处理器和存储芯片的基本单元——晶体管的速度越来越快，从而使由众多晶体管搭建的处理器性能得到迅速提升。其次，利用芯片上数量众多的晶体管，现代微处理器设计者能从软件代码中挖掘更多的并行性来改善程序性能，因此微处理器的实际性能增长速度甚至比摩尔定律所预测的还要快^[2]。

长期以来，挖掘和利用程序代码中并行性的各种策略具有一个有趣的共同点，即对软件程序员保持透明。从20世纪70年代微处理器问世至今，除了少量改动外，其实现都遵从传统的冯·诺

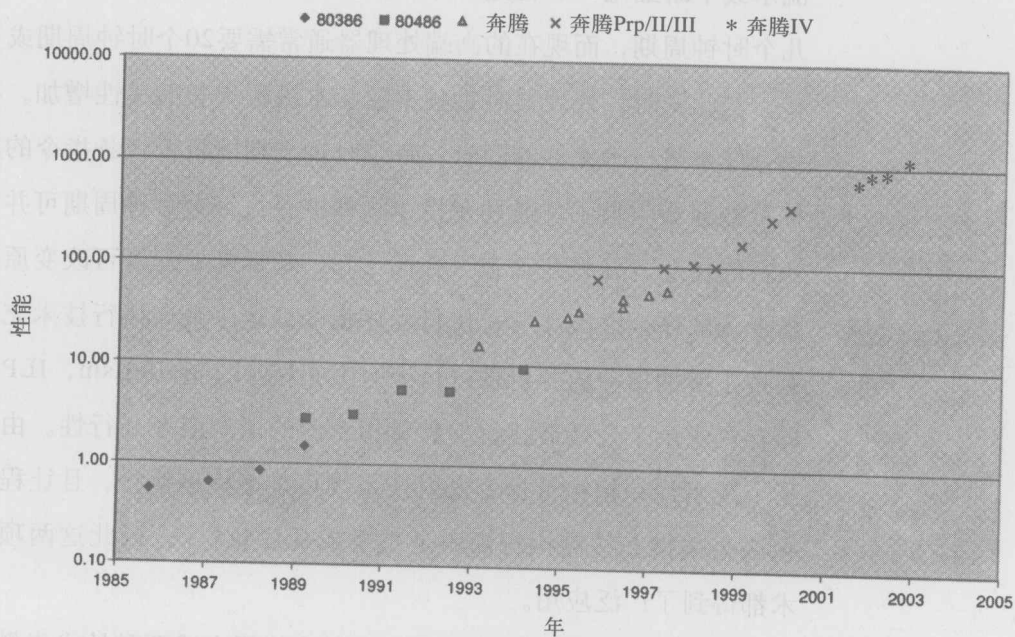


图1-1 Intel处理器性能随时间的变化，数据来源于Intel公开发表的SPEC CPU数字，根据不同测试版本（89，92，95，2000）进行了归一化



伊曼计算模型。对程序员来说，计算机系统就是由执行串行指令流的单核处理器，以及存放程序代码和数据的“存储器”所组成。处理器设计通常采用与前代处理器向前兼容的策略，其根本原因是这种策略更为经济。数十年来，硬件设计者的工作被限制在冯·诺伊曼抽象模型下，并在已有基础上改进系统性能。从存储器的角度看，为了继续维护冯·诺伊曼模型，设计者在处理器中增加了更大的缓存（cache）和寄存器堆，前者可以将“存储器”中频繁访问的部分数据存放在物理上更接近处理器的小型快速存储器中；后者则可以将最频繁使用的少量数据存放在更小、更快的、由编译器管理的“存储器”区域中。大多数处理器内部结构优化和改进的主要目的是实现以下两个目标或其中之一：增加处理器指令队列中每时钟周期可发射的指令数，以及超越摩尔定律，更快地提升处理器时钟频率。多级指令流水线技术将单条指令的执行过程切分为多个更小的执行步骤，减少了每个时钟周期需要开闭的电路逻辑数量，从而提高了处理器的时钟频率。由于指令流水线不断细化，20世纪80年代的微处理器执行单条指令时只需几个时钟周期，而现在的高端处理器通常需要20个时钟周期或更多，与此同时，处理器时钟频率随流水级数增长而线性增加。在指令流水线技术发展的同时，每个时钟周期能执行多条指令的超标量处理器问世了。通过在指令队列中寻找每个时钟周期可并发执行的指令，并将其动态组合在一起，超标量处理器可改变原有指令流顺序，以乱序方式执行程序指令。这种乱序执行技术充分利用了所谓指令级并行性（instruction-level parallelism, ILP），即存在于单个处理器执行的大量指令间的所有潜在并行性。由于指令流水线和超标量指令发射技术均可加速程序执行，且让程序员认为“指令按序执行而不是重叠或乱序执行”，因此这两项技术都得到了广泛应用。

然而不幸的是，处理器设计者很难使用上述两种技术来继续提高现代处理器的速度。典型程序指令中的可用并行性十分有限^[3]，

因此在运行大多数应用程序时，每周期发射指令数超过4条的超标量处理器对应用性能的改善效果非常有限。图1-2描述了真实的Intel处理器在不同年代里提取指令并行性的效率。当指令级并行性尚未被大量开发前，效率曲线（性能/每周期）显得比较平坦；而后随着指令并行性逐步得到有效利用，效率曲线急剧上升；最后由于可用的指令并行性已被基本开发完毕，近年来该曲线逐渐变得比较平缓。实现超标量处理器，需要在芯片上构造用于动态寻找并发执行指令的额外电路逻辑。该电路的复杂度与可同时发射指令数的平方成正比，因此构造每周期可发射多条指令的超标量处理器的成本非常昂贵。与之类似，流水线级数超过10~20的处理器的设计难度也相当大，因为此时每个流水级都太短，甚至都来不及完成整数加之类的基本逻辑操作，若继续细分流水线，电路就会变得异常复杂。另外，如果流水线超过30级，那些额外增加的流水线寄存器和旁路多路选择器产生的延迟，以及分支跳转导致流水线状态被冲刷产生的延迟，可能会抵消流水线级增加带来的所有性能提升。由于需要在每个处理器内核的中央逻辑里

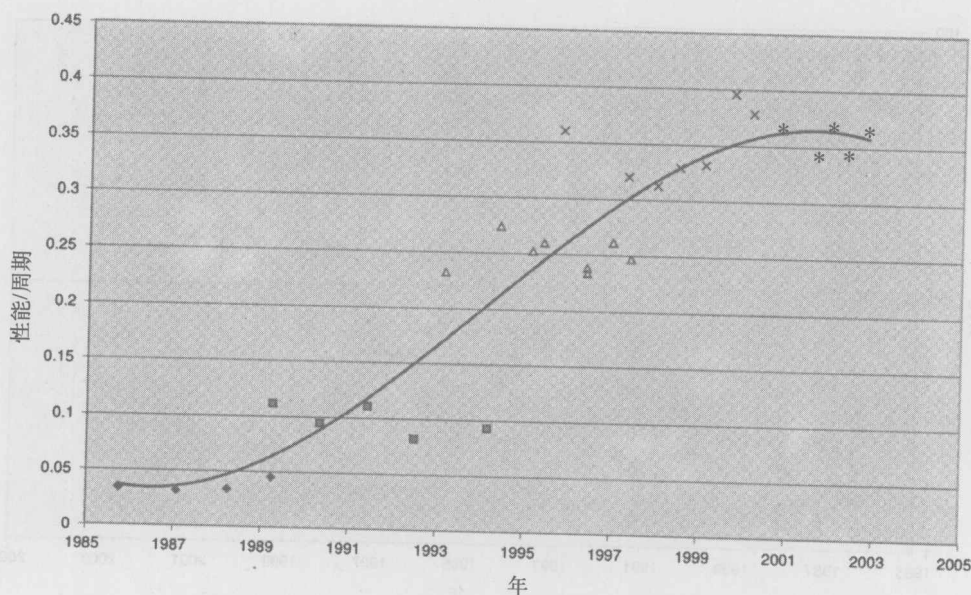


图1-2 Intel处理器每周期归一化性能随时间的变化，通过Intel发布的（归一化的）SPEC CPU数字和时钟频率计算而成



集成更多数量的晶体管，超标量发射与流水线技术的进一步发展遇到了另外一个瓶颈——增加的晶体管实在太多，以至于几乎没有哪家公司能雇得起足够数量的工程师在合理时间内完成这种处理器的设计和验证工作。上述这些问题使处理器的性能增长速度明显趋缓，同时迫使那些已无法承担进行有效竞争所必需成本的小厂商主动放弃了高端处理器市场。

当前的处理器内核研发工作被另一个物理因素所制约：功耗（power）。在过去二十年中，基于流水线与超标量技术的典型高端微处理器的功耗已从低于1瓦增加到100多瓦。即使每一代新型硅工艺都号称可以降低功耗（因为更小尺寸的晶体管开闭所需功耗更低），然而在实际应用中，除非直接使用新工艺和已有芯片设计来“缩小已有芯片尺寸”，这些新工艺所宣称的效果才可能出现。然而，处理器设计者总是试图在处理器内核中使用更多的晶体管来增加流水线阶段与超标量发射宽度，并在越来越高的频率下开闭这些晶体管，因此最终的效果就是相邻两代处理器的功耗呈指数增长（如图1-3所示），而相关的冷却技术却很难按指数

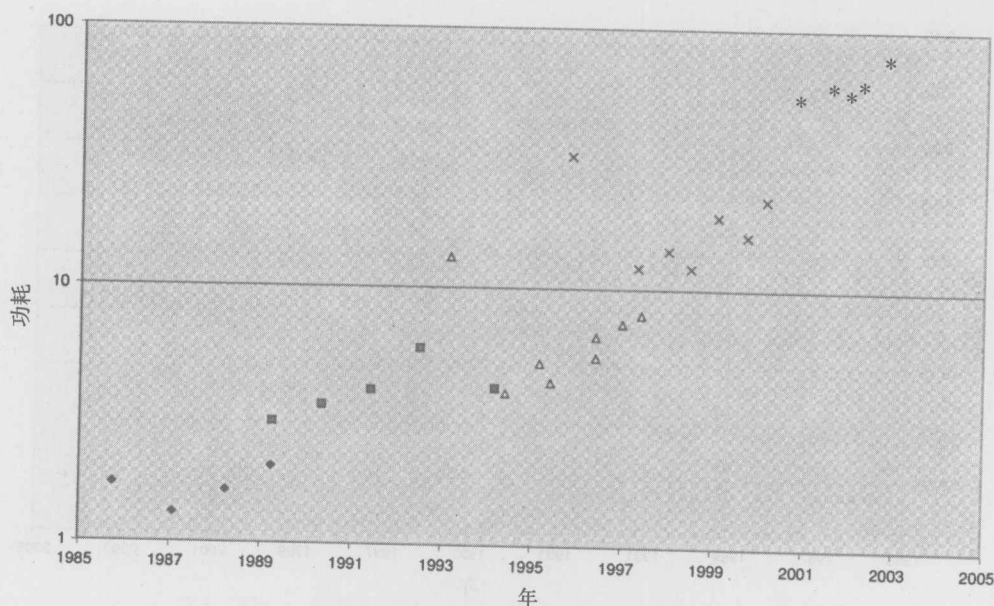


图1-3 Intel处理器的功耗随时间的变化，通过Intel发布的（归一化的）SPEC CPU数字和功耗计算而成

扩展。处理器从20世纪80年代不需要使用散热片,发展到90年代需要中等尺寸的散热片,再到今天需要安装巨大的散热片,以及一个或几个专用风扇来加快处理器周围的空气流动来降温。若照此趋势继续发展下去,那么下一代微处理器可能会需要类似水冷系统这样非常夸张的冷却系统来降温,这种冷却系统只能用于价格最昂贵的微处理器,若用于一般微处理器则价格太高,不太现实。



程序指令中有限的指令并行性、流水线的物理极限,以及冷却手段与成本的限制所造成的“功耗天花板”等因素混合在一起,致使传统处理器内核未来的性能增长再也无法与摩尔定律所预测的晶体管数量增长速度相匹配。尽管,使用更大的缓存可加快传统计算模型中“存储器”的访问速度,在某种程度上改善处理器性能,但显而易见的是,如果处理器设计方面没有更为深刻的变革,未来微处理器的性能提升速度将会急剧下降,除非处理器设计者们能找到有效利用高端硅片上大量晶体管的新途径,可以在减少额外功耗与设计复杂度的同时提升处理器性能。

1.1 一个新途径: 片上多处理器

前面已经阐明,在多种因素的合力作用下,人们目前几乎已不再具备建造更大更快的单核处理器的能力。现在处理器制造厂商开始采用一种新的微处理器设计模式: 片上多处理器 (chip multiprocessor, CMP)。本书通篇使用这一术语,它与“多核微处理器 (multicore microprocessor)”是同义词,后者通常被工业界所使用 (有些人也使用更专业的术语“众核微处理器,即 manycore microprocessor”来描述一个由大量十分简单的内核组成的CMP。在第2章中将作详细讨论,但这个词没那么流行)。正如名字所暗示的那样,片上多处理器就是集成到一个处理器芯片上,作为一个整体工作的一组单核处理器,此时几个较小的处理器“内核”填满了原本被单个大型单核处理器占用的芯片面积。



CMP处理器的更新换代所需工程量相对较小,只需将多个相同处理器内核像贴邮票一样放在芯片上,然后对处理器内核间的慢速连线逻辑进行适量修改以满足处理器内核的通信带宽与延迟需要即可。因此,相对重新设计一个拥有高速流水线逻辑的处理器,CMP处理器的升级换代所需的工程量要小得多。此外,与在芯片中封装单个处理器内核的传统多处理器不同的是,不同代CMP处理器的外观近似,因此虽然不同代CMP处理器中包含的处理器内核数目不同,但其主板设计只需进行细微修改即可。不同代CMP处理器间唯一的真正区别是,随着CMP处理器规模的扩展,其主板需要提供更高的内存与I/O带宽,并能逐渐采用那些新出现的I/O技术标准。当硅工艺发生几代变革之后,基于CMP的处理器设计能节省不菲的工程代价,因为处理器的升级换代设计只需多贴几个处理器内核,这相对较为容易。而且共性工程工作可以分摊到一系列相关处理器家族的设计工作中,处理器数目与时钟频率的变化就可使几乎相同的硬件满足不同的价格与性能需求。

由于在程序员看来,CMP系统中的多个处理器内核是不同的实体,因此我们需要将传统的冯·诺伊曼计算模型替换为全新的并行编程模型(parallel programming model)。使用这种模型,程序员必须将应用程序划分为多个线程(thread),每个线程是一个“半”独立的实体,多个线程可在CMP系统中不同处理器内核上并行执行;否则这些程序就无法充分利用CMP系统的处理能力。一旦被线程化,程序就可以充分利用线程级并行性(thread-level parallelism, TLP)。此外,每个线程还可开发其指令级并行性(ILP)。然而不幸的是,改写那些针对“传统”冯·诺伊曼单处理器编写的应用程序所花的代价和难度各不相同。

1.2 应用程序的并行性图景

为了更好地理解CMP的潜力,我们对应用程序中的并行性进

行了详细的调查。图1-4展示了一些典型应用程序中存在的并行性图景。X轴展示了程序并行性的各种概念级别，Y轴以指令为单位展示了并行性的粒度（granularity），也就是通信“和/或”同步点之间每个并行块的平均机器指令条数。该图表明，随着并行性概念级别的提升，并行性的粒度也倾向于增加（尽管不同级别间的粒度有很大部分的重叠）。

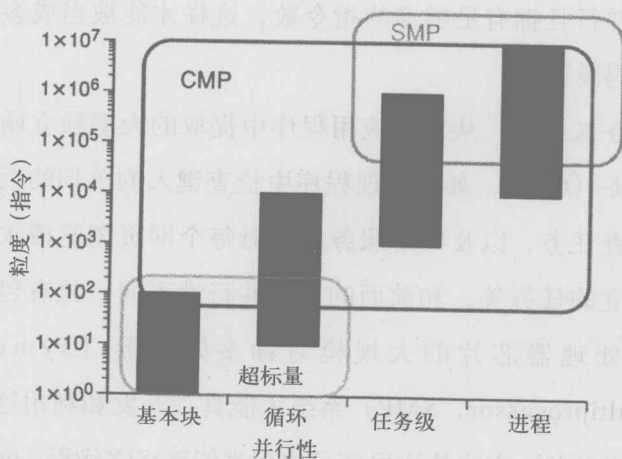


图1-4 不同处理器体系结构可能开发的不同“范围”的并行性小结

- 指令级并行：所有应用程序在多条指令之间都存在一些并行性。由于这一级别的粒度是单条指令，因此图1-4中没有列出。根据前面的讨论，超标量结构可以很好地利用这种类型的并行性。
- 基本块并行：以分支跳转结束的一小段指令称为基本块。除了在这些小块内提取ILP外，传统体系结构几乎无法开发其他的并行性。然而，高效的分支预测算法可支持同时从几个基本块中提取ILP，从而显著提高了超标量体系结构从多个基本块中寻找潜在并行指令的能力。
- 循环迭代级并行：典型循环的每次迭代常处理独立的数据元素，因此循环的多次迭代实质是一组独立的可并行执行的工作（这显然不适用于像指针追逐那样的高度相关循环迭代）。传统系统利用这种并行性的唯一途径是使用具有较



大指令窗口的超标量处理器，以发现多次循环迭代之间的指令级并行性，或者使用一个聪明的编译器，利用软件流水（software pipelining）技术，交叉调度不同循环迭代间的指令执行，以弥补硬件不能直接并行化循环的缺陷。使用OpenMP这样的软件工具，程序员可将这一级别的并行性转化为有限的TLP，前提条件是循环的多次迭代必须高度并行且拥有足够多的指令数，这样才能被当成多个独立代码段使用。

- 任务级并行：从单一应用程序中提取的大型独立函数称为任务（task），如字处理程序中检查键入的单词的后台拼写检查任务，以及网络服务器中对每个网页浏览请求分配的独立的任务等。和前面的各种并行性不同，只有包含多个微处理器芯片的大规模对称多处理器（symmetric multiprocessor, SMP）系统才能真正开发和利用这种级别的并行性。方法是让程序员使用类似POSIX线程（pthreads）这样的软件机制，将代码手工切分为可以显式利用TLP的线程，这种级别的并行性规模太大，传统超标量处理器无法在ILP级别开发其并行性。
- 进程级并行：比任务更大的是完全独立的OS进程，它们来自于不同的应用程序，而且具有独立的虚拟地址空间。开发这一级别的并行性类似于开发任务间的并行性，但粒度更大。

人们常使用任务的延迟（latency）来度量具有基本块和循环级并行性的应用程序性能，而对具有更高级别并行性，如任务或进程级并行性的程序性能进行度量常使用任务或应用程序的吞吐率（throughput）这一参数，原因在于此时程序员通常更关心单位时间内完成的任务总数，而非每个任务完成的时间。

CMP的诞生改变了应用程序并行性的图景。与传统单处理器

不同，多核处理器芯片可以直接利用应用中的TLP，因此也可像SMP一样挖掘和利用传统大粒度任务和进程级并行性。另外，由于内核间通信延迟远低于SMP，并能集成猜测线程机制之类的新特性，CMP处理器能够充分开发循环、任务甚至基本块等细粒度并行性。



1.3 一个简单的例子：超标量与CMP

展现CMP系统优缺点最好的办法是将其与一个各方面均相当的超标量单核处理器进行比较。在底层系统结构存在较大差异的前提下，选择一对“等价”的处理器芯片显然非常困难，不免会带入一些主观臆测。定义“等价”的一种方法是使用相同工艺，设计两款硅片面积不同的不同处理器芯片，这两个处理器芯片拥有相同的片外I/O资源，并运行在相同时钟频率下。基于这两种体系结构模型，我们就可以通过模拟代码执行来比较在这两种处理器上运行多个不同类型的应用程序时的性能结果。

在此例中，我们构建了一个由四个简单小型处理器内核构成的CMP处理器，其面积与一个大型单核超标量处理器相当。表1-1展示了当真实构造这两款“等价”处理器时，它们所具有的关键体系结构特征。这个大型单核超标量处理器（SS）实质上是MIPS R10 000处理器^[4]的6路发射超标量版本，R10 000是20世纪90年代设计出的经典处理器，它几乎具备之后其他所有高端乱序发射超标量微处理器的全部特性，非常类似于今天的高端乱序发射超标量微处理器。这里的片上多处理器（CMP）是一个4路CMP，由4个类似于Intel奔腾和DEC Alpha的早期2路顺序发射超标量处理器构成，Intel奔腾和DEC Alpha出现于1992年前后。表1-1给出的两种体系结构中，我们都将整数与浮点单元数目以及重复延迟设置为真实R10 000的相关数值。