第33卷 第16期 计 算 机 工 程 2007年8月
Vol.33 No.16 Computer Engineering August 2007
·工程应用技术与实现· 文章编号：1000—3428(2007)16—0208—03 文献标识码：A 中图分类号：TN47

多核处理器的结构设计研究

何 军，王 飙
（国家高性能集成电路上海设计中心，上海 201204）

摘 要：围绕如何进行多核处理器的结构设计，提高处理器性能这一问题，结合传统多处理机设计原理对多核处理器结构设计进行了研究，并对当前主要商业多核处理器进行了研究，揭示了其发展趋势，探索了未来多核处理器设计的发展方向。

关键词：多核处理器；ILP；TLP；处理器结构

Research on Architecture Design of Multi-core Processor

HE Jun, WANG Biao
(National High Performance IC Shanghai Design Center, Shanghai 201204)

【Abstract】 On the issue how to design the architecture of multi-core processor as to improve its performance, this paper researches on the multi-core processor architecture design referring to the theory of traditional multiprocessor, discloses the development trends of commercial multi-core processors by making a study of current commercial multi-core processors, and reflects on the future of multi-core processor design.

【Key words】 multi-core processor; ILP; TLP; processor architecture

在过去的几十年中，一般通过增加发射宽度和提高时钟频率来提高处理器性能，利用超标量发射、乱序发射执行、超级流水、动态转移预测、大容量片内 Cache 等技术来开发程序的指令级并行性(ILP)。但是，增加发射宽度使设计的复杂性急剧增加，这一方面不利于时钟频率的提高，另一方面也难以适应深亚微米工艺下日益显现的线延大于门延的负面影响，也使处理器设计验证成本难以承受。更重要的是，指令之间的数据和控制相关，可以开发的 ILP 也有限，当发射宽度大于 4 时，大多数应用可以获得的性能提升十分有限。提高时钟频率，使流水线级数增加，转移预测失败和 Cache 失效的代价也随之增加，这样提高时钟频率获得的性能十分有限，另外频率提升使功耗也随之增加，使芯片封装和冷却代价也水涨船高。这些都使经典的超标量结构处理器难以进一步提高处理器性能。

从应用的角度看，如在线事务处理(OLTP)、决策支持系统(DSS)和 Web 服务等这类应用的特点是具有丰富的线程级并行性(TLP)而缺少 ILP，传统处理器结构难以开发，一般是采用多处理机系统开发。随着集成电路工艺的发展，利用先进工艺将多个处理器核集成到单个芯片上以开发 TLP 成为可能。因此，在 1996 年斯坦福大学的研究人员提出了单片多处理器(CMP)结构，并进行了研究[1]。

CMP 结构的主要思想是通过简化超标量结构设计，将多个相对简单的超标量处理器核集成到一个芯片上，从而避免线延的影响，并充分开发 TLP，提高吞吐量。通过简化单核设计，不仅有利于避免线延的影响，而且有利于时钟频率的提升，还有利于减少处理器研制的时间周期。

另外，CMP 结构能够适应工艺尺寸比例缩放，具有较好的可扩展性。因此，CMP 结构成为当前和未来处理器发展的主要方向。但是，如何进行多核处理器结构设计，以充分提高处理器性能是一个值得仔细研究和探索的问题。

1 多核处理器设计

从某种意义上讲，多核处理器就是一个芯片级的多处理机系统，是板级多处理机系统的“微缩版”，成熟的多处理机系统设计原理对多核处理器结构设计具有重要的参考价值。从硬件角度看，多核处理器结构设计主要包括两个方面：单核结构设计与多核组织结构设计。

1.1 单核结构设计

虽然多核处理器结构最初的动机是利用简化的超标量处理器核组成多核处理器，但是广义地来看，单核结构并不局限于超标量结构，完全可以利用现有的各种处理器结构来实现。纵观处理器结构的发展历程，可以采用经典的超标量结构，也可以采用超长指令字(VLIW)结构和多线程结构(技术)。其中多线程结构主要有 3 种：细粒度多线程，粗粒度多线程和同时多线程(SMT)[2]。在此，先对各种处理器结构进行简单回顾。

超标量结构主要目标是利用硬件动态地从指令窗口中寻找出多条不相关的指令，由多个独立功能部件并行执行，以充分开发 ILP。但正如前文所述，随着工艺的进步，复杂的超标量结构难以进一步提高处理器性能。VLIW 结构企图通过编译器静态调度避免复杂的硬件动态调度，从而简化硬件设计，进一步开发 ILP。但随着发射宽度的增加，VLIW 结构需要的集中的寄存器文件的大小以及旁路逻辑等设计复杂性也急剧增加，此外，VLIW 结构性能对编译器的过分依赖，以及与主流指令集结构(ISA)不兼容，也使 VLIW 结构不堪重负。

超标量和 VLIW 结构目标都是为了开发 ILP，而多线程结构则是在开发 ILP 的基础上开发 TLP，以提高昂贵的处理器资源的利用率。多线程结构的思想类似于早期分时共享计

作者简介：何 军(1980－)，男，硕士研究生，主研方向：计算机体系结构，微处理器设计，王 飙，高级工程师

收稿日期：2006-09-20 **E-mail：**joyhejun@yahoo.com.cn

算机系统，执行多个线程的处理器在遇到某个线程由于Cache 失效或者转移预测失败而停顿时，可以切换到另一个线程执行。这些结构的差别在于线程间共享的资源以及线程切换机制(参见表 1，在超标量处理器中，当某个时钟周期指令发射槽未填满时就发生“水平浪费”，当由于某种原因在某个时钟周期没有指令发射时就发生“垂直浪费”)。其中，SMT 结构能在每一时钟周期从不同的线程发射指令到多个执行部件，允许多个线程在共享的执行资源中细粒度地动态交叉执行，因而具有最大的灵活性和最高的资源利用率，但其实现也最复杂。实际上，多核结构也是一种多线程结构，只是其主要硬件资源是空间静态划分的，由单线程独享，其资源利用率也较低。从开发并行性的角度看，超标量和 VLIW 结构充分开发了单线程 ILP，SMT 等多线程结构在开发单线程 ILP 的基础上进一步开发了 TLP，而多核结构则是牺牲部分单线程 ILP 换取了 TLP。

表 1　多线程结构

多线程技术	线程间共享资源	线程切换机制	资源利用率
细粒度	除寄存器文件和控制逻辑外	每个时钟周期	消除垂直浪费
粗粒度	除取指缓冲、寄存器文件和控制逻辑外	流水线停顿时	消除垂直浪费
SMT	除取指缓冲、返回地址堆栈、寄存器文件和控制逻辑、重排序缓冲、Store 队列外	所有上下文同时处于活跃状态，无切换	消除垂直浪费和水平浪费

单核结构设计解决的主要问题是如何充分利用一定的芯片面积，在单核结构的复杂性(单线程性能)与单核数量(多线程性能)之间进行取舍，以获得最佳性能。这与设计的应用目标密切相关，对于高 TLP 应用来说，多个简化的单核设计优势明显，但是对于低 TLP 的应用来说却很不利，适度复杂的单核设计才是更好的选择。核数还受到片内 Cache 容量、片外带宽和多核结构组织等因素的限制。一个折中的方法是多个核采用不同的结构设计，称为“异构多核”，而所有处理器核采用同一结构设计称为“同构多核”。同构能简化设计验证的代价，而异构能提高多核处理器对应用的适应性。

单核结构设计还需要考虑的一个问题是：是否采用以及采用何种多线程技术提单核资源利用率。单核多线程技术的使用能够弥补多核结构资源利用率低的缺点，虽然采用多线程技术会使单核设计复杂化，但是若能增加少量硬件实现多线程技术显然是值得的，尤其是 SMT 技术独特的灵活性十分具有吸引力。这需要在硬件代价与可获得的收益之间进行权衡。

1.2　多核组织结构设计

在当今的服务器市场上，共享存储器的多处理机系统是主流，这些系统硬件支持单存储器空间，且具有大量应用，主要包括对称多处理机(SMP)和 Cache 一致性非均匀存储访问(CC-NUMA)系统。其中 CC-NUMA 是 SMP 系统的扩展，这二者是均匀存储访问(UMA)和非均匀存储访问(NUMA)的典型代表。这些系统的设计原理对多处理器核的组织具有重要的参考价值。与 SMP 和 CC-NUMA 类似，多处理器核组织结构可以有两种(如图 1 所示)：均匀 Cache 访问(UCA)和非均匀 Cache 访问(NUCA)。UCA 结构中，多个处理器核与二级 Cache 通过互联系统(总线或交叉开关)互联，所有处理器和对二级 Cache 访问延迟相同；在 NUCA 结构中，每个处理器核具有本地二级 Cache，通过互联系统对其他处理器核的二级 Cache 访问(延迟变长)。随着集成度的提高，也可以将三级 Cache 集成到片内，如图 2 所示。需要强调的一点是，以上 4 种组织中，片内二级或三级 Cache 均为所有处理器核共享。

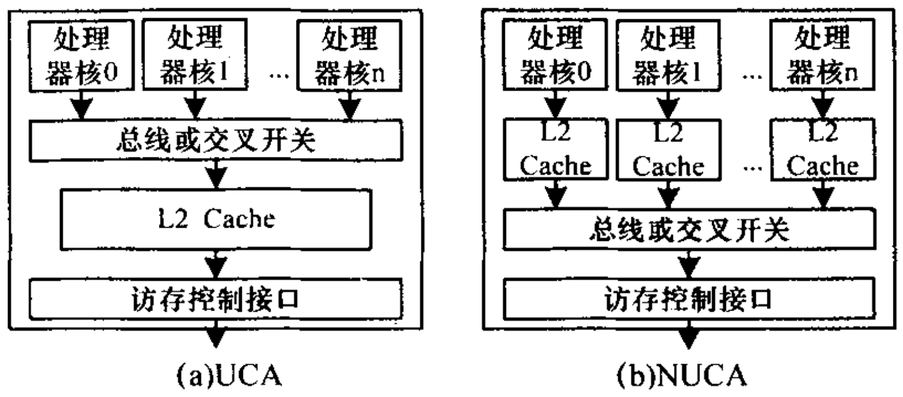

图 1　多核组织结构示意图(1)

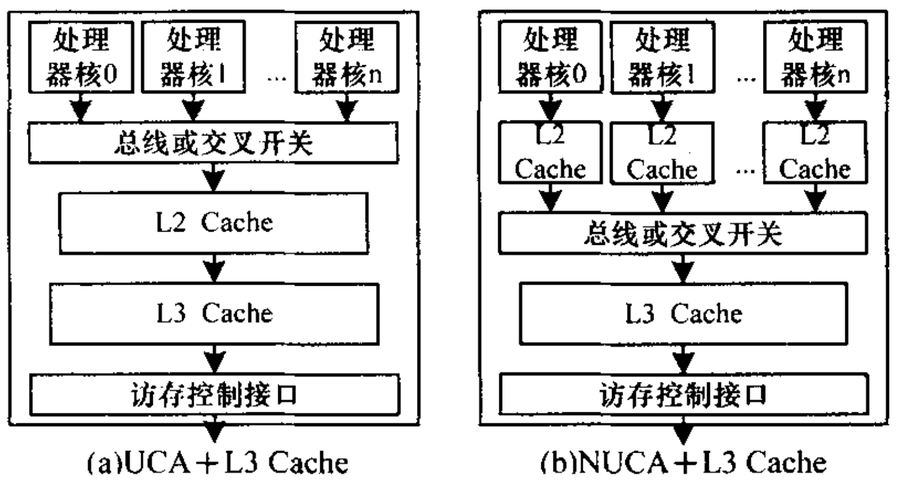

图 2　多核组织结构示意图(2)

多核组织结构设计主要解决片内多核之间的互联和通信机制问题。与多处理机系统类似，多核之间的互联可以采用总线或者交叉开关甚至多级网络实现，需要在可扩展性和复杂性之间进行权衡。多个处理器核通过共享 Cache 数据单元实现信息交换和同步，除了传统的板级多处理机的 Cache 一致性问题外，还需要考虑芯片级多处理器核的 Cache 一致性问题。参考多处理机系统，可用两种方法：监听协议和基于目录的协议。在多核间采用总线互联时，可以采用监听协议，此时可以采用两种保持一致性的策略：写无效和写更新策略。由于写更新策略会使互联总线负载过重，降低通信效率，因而一般采用写无效策略。当多核之间采用非总线互联时，可以采用基于目录的协议。另外，板级多处理机的 Cache 一致性接口可以由多处理器核共享，需要考虑的是该接口能否满足多处理器核的带宽。

与多处理机系统类似，多核间的互联系统和共享的二级(或三级)Cache 将因多核争用而成为瓶颈，多核间的片内延迟比多处理机的板级延迟小得多，尽量提高带宽、降低通信开销是多核组织结构设计的关键，也直接影响多核结构的性能。另外，受多核组织结构的影响，可能需要仔细考虑整片处理器的 Cache 结构设计，包括层次结构、大小、独立与共享以及对 Cache 一致性的支持等。最后，多核结构可能对访存和 I/O 带宽要求更高，加剧访存瓶颈，需要仔细设计整片访存接口和 I/O 接口。

1.3　当前商业多核处理器

如何进行多核处理器结构设计，是一个开放性的问题，是当前研究和实践的热点。对当前商业多核处理器进行研究，有助于未来多核处理器的发展。为此，对主要的商业多核处理器，如 STI(SONY，TOSHIBA，IBM)Cell[3]，Sun Niagara[4]，HP Piranha[5]，IBM POWER5[6]，Intel Montecito[7]和 AMD Opteron 等进行了研究，从表 2 可发现以下几个重要趋势：(1)简化单核设计，采用顺序单/双发射；(2)采用多种单核多线程技术；(3)采用有效的总线或交叉开关互联，提供高核间带宽；(4)采用先进访存接口技术，提高访存带宽。

表 2　商业多核处理器

	POWER 5	Piranha	Niagara	Cell	Montecito	Opteron
核数	2	8	8	9	2	2
同构异构	同构	同构	同构	异构	同构	同构
线程数	2×2	8×1	8×4	1×2+8	2×2	2×1
多线程	SMT	无	细粒度+SMT	SMT	TMT+SMT	无
发射	乱序 8 发射	顺序 1 发射	顺序 1 发射	顺序 2 发射	顺序 6 发射	6 发射
L2 Cache	共享	共享	共享	独立	独立	独立
访存带宽	未知	12.8GB/s	20GB/s	25.6GB/s	10.66GB/s	6.4GB/s
访存接口	DDR	Rambus	DDR2	Rambus XDR	DDR	DDR
核间互联	交叉开关	交叉开关	交叉开关	环(EIB)	总线	交叉开关
核间带宽	未知	32GB/s	200GB/s	256GB/s	未知	未知

这些多核处理器大多都是针对商业服务器应用，多采用了简单的顺序单/双发射，简化了单核结构，但是也有一些直接继承了前一代的设计，这可能是考虑到兼容性、研制周期等方面的因素。3 种多线程技术在这些处理器中都有采用，而且都不是单纯的某一种技术，而是在处理器结构的不同阶段部分采用或者综合采用了这些技术，比如 Piranha 综合采用细粒度和 SMT 技术，Montecito 前端流水线采用了分时多线程(TMT，类似粗粒度多线程)，而在访存时采用了 SMT。在多核组织上，一般都考虑采用有效的总线或交叉开关互联，提高核间带宽，另外，也采用了先进访存接口技术，提高访存带宽，以缓解访存带宽，如 Niagara 利用 4 条 DDR2 访存通道，提供了 20GB/s 的访存带宽，Cell 采用两个 Rambus XDR 接口使访存带宽高达 25.6GB/s。

值得一提的是 Niagara 和 Cell。Niagara 通过简化单核，采用了 8 个顺序单发射单核，每个单核支持 4 线程的结构，能提供 32 个线程，该处理器综合利用多核多线程结构提供了大量线程，是主要针对商业应用的典型。Cell 试图为游戏和多媒体、实时响应系统和科学计算等多种应用提供一个宽阔的平台，因而采用了异构设计：顺序双发射的 PowerPC 核(PPE)和 8 个双发射增效核(SPE)，其中 PPE 采用了 SMT 技术，SPE 有专用的 128 个 128 位宽寄存器和 256KB 本地 SRAM 存储器，支持整数和浮点 SIMD 运算，每个 SPE 通过专用的 DMA 通道与 PPE 相连，Cell 的异构设计和独特的 SPE 设计值得研究和参考。

2　多核处理器的未来

随着集成度的提高，芯片上将集成更多的处理器核，另外随着单核多线程技术的应用，多核结构开发 TLP 的能力将更加强大，这可能推动并行编程模型的普及。但是，并行编程模型由于其复杂性，远没有顺序编程容易，而且大量应用都是基于顺序编程模型的。因此，串行程序自动并行化成为一种的理想方法。目前，对于具有可预测的控制流(如具有确定迭代次数的嵌套循环)和可静态确定的访存模式(如顺序读取一个浮点多维数组)的大量科学数据计算应用，可以利用编译器实现串行程序的自动并行化。但是对于一些非数值计算串行程序，由于其不规则的控制流、不可预测的访存方式，或者难以静态确定的存储器数据相关性，编译器难以将其划分为多个并行线程。对这类应用，如果能增加一些硬件支持，可能有利于实现串行程序的并行化。

从另一个角度看，ILP 可以通过编译器或者硬件来发掘，利用转移预测、寄存器重命名、动态调度和推测执行等超标量技术来解决指令之间的控制和数据相关，TLP 同样也可以通过编译器或者硬件来发掘。可以利用编译器静态解决并行线程之间的控制和数据相关性，实现串行程序的并行化，当然也可以采用一些硬件机制来解决这一问题。如果能够实现，将弥补多核结构在单线程性能上的缺点，更重要的是，有助于实现串行程序自动并行化，化解并行编程的困境。实际上，并行编程模型是将并行化的任务交给程序员完成，若能用硬件解决线程之间的相关性，将大大减轻程序员的负担。

目前，已经有一些这方面相关的研究和探索，主要有 3 种方法：动态多线程，多标量和线程级推测(参见表 3)。对于线程间控制相关性，这些方法都是通过利用在程序控制流图中控制无关的节点(无论控制流图中分支条件如何化解都会执行到某一个节点，则称该节点是控制无关的)处创建线程而解决的；对于线程间其他两种相关性，一般采用编译器和硬件推测的方式解决，但具体方式各不相同，硬件复杂性也不同。值得一提的是线程级推测，利用编译器来避免寄存器数据相关，通过对 Cache 一致性(MESI)协议进行简单扩展来解决存储器数据相关，对硬件改动最小，更有潜力。美国斯坦福大学的 Hydra(一种多核处理器)研究小组[8]，一直致力于这方面的研究。

表 3　线程相关性解决方法

	动态多线程	多标量	线程级推测
线程来源	循环退出，子程序返回	循环体和控制流汇合点	循环体
线程创建机制	硬件	软件或编译器	软件或编译器
寄存器数据相关	硬件，预测和推测	软硬件推测支持	编译器避免
存储器数据相关	硬件，预测和推测	基于硬件推测	硬件相关推测

3　结束语

多核处理器结构通过简化单核结构避免线延影响，适应了工艺发展和商业应用的特点而成为未来的发展方向，如何充分发掘多核结构的潜能仍然需要仔细研究和探索。本文结合传统多处理机设计原理，从单核结构设计和多核组织结构两方面进行了研究探讨，分析指出了设计中的关键问题。据此还结合当前商业多核处理器进行了分析，指出多核处理器的发展的 4 点重要趋势。探讨了通过硬件解决线程间的相关性，实现串行程序自动并行化这一富有挑战性和重要意义的问题，也许这将是多核处理器未来发展的重要方向。

参考文献

1 Kunle O K, Basem A N, Hammond L, et al. The Case for a Single-chip Multiprocessor[C]//Proc. of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems, New York. 1996-10-02.

2 Tullsen D M, Eggers S J, Levy H M. Simultaneous Multithreading: Maximizing On-chip Parallelism[C]//Proc. of the 22nd Ann. Int'l Symp. on Computer Architecture. 1995: 392-403.

3 Kahle J A. Introduction to the Cell Multiprocessor[J]. IBM Journal Res. & Dev., 2005, 49(4/5): 589-604.

4 Kongetira P. A 32-Way Multithreaded SPARC Processor[J]. IEEE Micro, 2005, 25(2): 21-29.

5 Barroso L A. Piranha: a Scalable Architecture Based on Single-chip Multiprocessing[C]//Proc. of Int'l Symp. on Computer Architecture. 2000: 165-175.

6 Kalla R. IBM Power5 Chip: A Dual-core Multithreaded Processor[J]. IEEE Micro, 2004, 24(2): 40-47.

7 McNairy C, Bhatia R. Montecito: A Dual-core, Dual-thread Itanium Processor[J]. IEEE Micro, 2005, 25(2): 10-20.

8 Hammond L. The Stanford Hydra CMP[J]. IEEE Micro, 2000, 20(2): 71-84.