

面向多核的共享多通道 Cache 体系及原型构建

刘彩霞, 石 峰, 邓 宁, 宋 红, 薛立成

(北京理工大学 计算机科学技术学院, 北京 100081, lexhxb@gmail.com)

摘 要: 为构建一个合理高效的多核系统通信架构及数据存储路径, 提出了一种使用多通道 Cache 的可扩展可配置多核体系 (Architecture Utilizing Multi-Channel Cache, AUMCC), 并基于 FPGA 利用 LEON3 处理器建立了该体系的原型平台. AUMCC 体系中采用多通道 Cache 作为 L2 级 Cache, 并且基于多通道 Cache 的不同访问模式, AUMCC 可以配置成私有 L2 Cache 或共享 L2 Cache 两种结构. 原型系统中模拟测试结果表明, 相比传统共享 L2 Cache 体系, 性能提高了 37%, 同时 AUMCC 的系统层次化特性确保了良好的系统扩展性.

关键词: 面向 CMP 的共享多通道 Cache 体系; 多通道 Cache; 基于 FPGA 的原型系统

中图分类号: TP303

文献标志码: A

文章编号: 0367-6234(2010)11-1833-05

CMP-oriented shared multi-channel Cache architecture and its prototype construction

LIU Cai-xia, SHI Feng, DENG Ning, SONG Hong, XUE Li-cheng

(School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China, lexhxb@gmail.com)

Abstract: To construct a rational and efficient communication structure and a high-efficient data storage path for CMP system, a kind of scalable and configurable CMP architecture utilizing multi-channel Cache (AUMCC) is proposed in this article and its prototype system is established based on FPGA and LOEN3 processor. Multi-channel Cache is adopted as L2 Cache in AUMCC. AUMCC can be configured to be private L2 Cache structure or shared L2 Cache structure based on the access mode of multi-channel Cache. Simulation and testing results show that AUMCC architecture can achieve about 37% performance improvement compared with shared L2 Cache BUS-centric (SCA) architecture, and the hierarchy characteristic of AUMCC assures the good scalability.

Key words: shared multi-channel Cache for CMP architecture; multi-channel Cache; prototype system based on FPGA

多核 (Chip Multi-Processor, CMP) 处理器能够充分开发不同粒度的并行性, 因而成为微处理器的主流发展方向. 本质上多核设计的瓶颈还在于解决片内多核之间的互联和通信机制问题, 也就是需要一个快速的数据存储和传输路径^[1-3]. 存储系统在多核系统中不再仅仅意味着对数据的组织和存储, 合理的存储体系对于提高多核系统并行通信性能甚至系统整体性能起着至关重要的作用

用^[4-5].

基于将不同的存储体系的优势结合的目的, 本文提出一种使用多通道 Cache 作为 L2 Cache 的可扩展可配置 CMP 体系 Architecture Utilizing Multi-Channel Cache, AUMCC. 根据所设计的多通道 Cache 的不同工作模式, AUMCC 可以配置成私有 L2 Cache 结构或共享 L2 Cache 结构. AUMCC 的共享模式中, 多通道 Cache 的多个独立的访问通道确保了核间共享数据的高效并行传输, 提高了核间通信带宽; 同时多通道 Cache 的分离访问模式简化了对 L1 Cache 的一致性维护开销. 基于 LEON3 处理器的原型系统上性能模拟及测试表

收稿日期: 2009-03-15.

基金项目: 教育部博士点基金资助项目 (200700007070).

作者简介: 刘彩霞 (1973—), 女, 博士生, 副教授;

石 峰 (1961—), 男, 教授, 博士生导师.

明,AUMCC 体系能提供高效的并行通信,通信性能相对于基于总线共享 Cache 结构约高出 37%,系统的层次化特点使得系统具有良好的扩展性.

1 使用多通道 Cache 的多核系统 AUMCC 总体结构

AUMCC 体系是一种采用多数据通道 Cache 作为 L2 Cache 的层次化可扩展可配置 CMP 结构^[6](如图 1 所示). 系统中每个核拥有私有分离 L1 指令和数据 Cache,L2 Cache 可以根据多通道 Cache 的访问模式动态配置成私有或共享两种方式,因而 AUMCC 体系相应的可以配置成私有 L2 Cache 结构或共享 L2 Cache 结构. 每个核对应一个通道,各个通道依次顺序编址. 内核间可以通过底层的互连网络通信,也可以通过共享模式中的多通道 L2 Cache 交换数据. 共享模式的 AUMCC 中,内核可以读写本地通道内的存储体,可以读其他通道内的存储体,因而不存在多核之间的写冲突问题,简化了对 L1 Cache 的一致性维护开销,提高了系统性能,同时 Cache 的多通道特性可以缓解传统基于总线共享 Cache 结构的总线带宽的瓶颈问题,提高系统访问带宽和可扩展性.

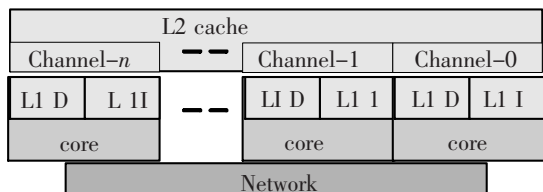


图 1 AUMCC 总体架构

图 1 中该结构具有明显的层次性. 各功能核及多通道 L2 Cache 可以看作系统的“超核”. 按照构建“超核”的方式,可以通过共享更高级的多通道 Cache 构成更大规模的系统,相似的构建方式可以应用于 memory 或辅助存储器.

2 多通道 Cache 体系结构

2.1 多通道 Cache 硬件组织结构

利用 CACTI 进行的对比实验表明,4 个 1 KB 大小的单端口 SRAM 在工作频率和面积上都优于一个 4 端口 4 KB 的全定制 SRAM 模块,而且单端口 SRAM 可以由 EDA 工具快速编译生成,便于设计实现. 因此本文采用存储体多体交叉的方式实现本体系中的多通道 Cache^[7-9],其硬件组织结构如图 2 所示.

多通道 Cache 由交叉开关和单端口 Cache 以多体交叉方式构成. 该方案中, m 个核通过一个 $m \times n$ 的交叉开关 (crossbar) 共享一个具有 m 个访

问通道、 n 个访问端口的 Cache 存储阵列. 每个通道内设置了 p 片单端口 Cache. 每一个 Cache 模块拥有独立的 Cache 控制器,可以提供独立的访问端口. 通道间以高位地址交叉方式编码形成多通道 Cache 地址,存储器的多体交叉是提高其数据带宽的有效方法^[10]. 通道间高位交叉使得多通道 Cache 的相邻地址分布在同一个 Cache 通道内,可以减少通道冲突. 而通道内各 Cache 则以低地址交叉方式形成通道地址,低地址交叉方式可以为访问同一通道的各请求提供基于“生产者-消费者”的流水并行访问方式,降低共享冲突.

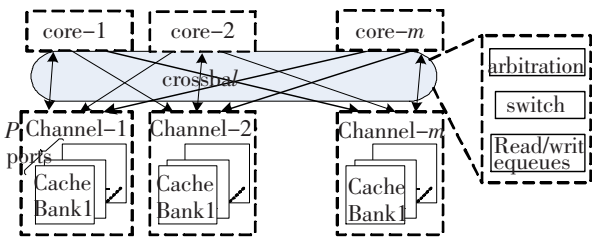


图 2 多通道 Cache 组织结构图

2.2 双模式操作与并行交叉访问

本文为多通道 Cache 设计了“私有”和“共享”两种工作模式. 1) 在私有模式下,任意核 $core-i$ 只能读/写与其对应通道内的多个存储体,不能访问其他的通道;2) 在共享模式下,每个核可以读取其他通道的数据,但不能向其中写入数据. 任意核必须通过其对应通道的存储体与其他核交换共享数据.

可见,在私有模式下,所有的存储体都不存在访问竞争. 在共享模式下,AUMCC 不存在多核写冲突的问题,简化了维护 L1 Cache 数据一致性的硬件开销,提高了核间共享数据的传输速度,有利于提高系统应用的实时性. 在共享模式下,通道内存储体的低地址交叉方式消除了核间的读冲突.

在任务流水的计算模式下,核间的共享数据相继构成“生产者-消费者”关系:前一个核的计算输出直接作为下一个核的计算输入. 为了有效支持这种传输模式,在 AUMCC 中采用了交叉访问的机制:1) 当“生产者” $core-i$ 向其对应通道内的存储体 Bank- i 写入第 1 块共享数据之后,释放该存储体,转而向 Bank- $i+1$ 写入第 2 块共享数据;2) “消费者” $core-j(j \neq i)$ 启动读访问,从 Bank- i 读出第 1 块共享数据; $core-i$ 释放 Bank- $i+1$,向 Bank- $i+2$ 写入第 3 块数据. 依此类推,直至全部并行流水传输完成.

因此,在写入第 1 块共享数据之后,读写操作就可以并行执行. 当多方的计算负载均衡,速度匹配的时候,核之间可以进行流水并行传输. 同步等

待延迟最小,传输效率达到最高.

2.3 AUMCC Cache 体系结构

在 AUMCC 体系的私有工作模式下,每一个功能核只读写 L2 Cache 本地通道的数据,因而不存在 L1 Cache 的一致性问题的. 在共享工作模式下,任意功能核只能读取 L2 Cache 非本地通道的数据,因而共享模式下 L1 Cache 的一致性协议可以采用简化的 MESI 协议——M 协议:为每一个 L1 Cache 块设置一位“M”位,用来标识在共享模式中读取的 L2 非本地通道内的数据是否被通道拥有核修改. 当任意功能核修改了 L2 Cache 本地

通道内的数据时,会通过广播方式通知其他所有核,其他核接收到通知后会根据 Tag 位和通道标识(CID)将相应的 L1 Cache 中的 M 状态位置“1”. L1 Cache 在进行 Tag 比较的同时会查看状态位是否为“1”,若状态位为“1”,则即使 Tag 命中,也会发出 L1 Cache “缺失”请求,进而产生共享读请求,访问 L2 非本地通道. L2 Cache 会根据核标识(CID)将命中的数据送入相应的 CPU 及 L1 Cache 中. 依据这样的 L1 Cache 一致性协议, AUMCC Cache 体系结构如图 3 所示.

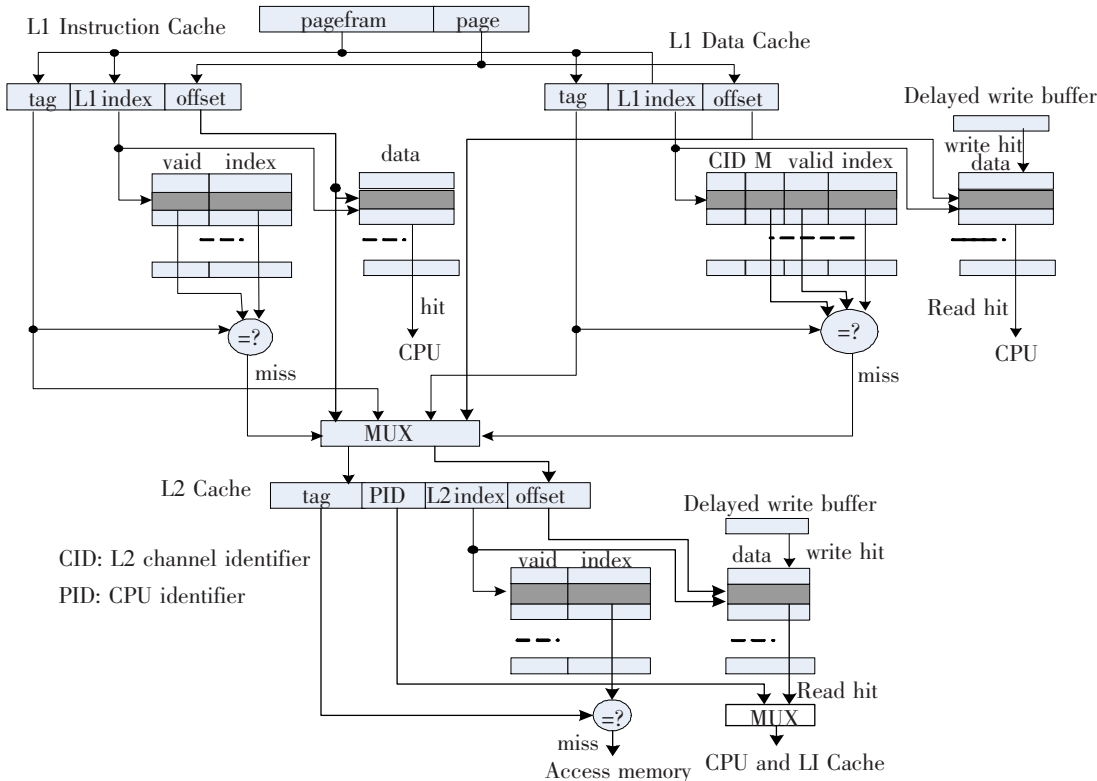


图 3 AUMCC Cache 体系结构

3 AUMCC 原型系统的构建及性能分析

3.1 原型系统的构建平台及构建方案

AUMCC 原型系统的构建平台及构建方案设置为:系统环境为 Windows with Cygwin,仿真器为 Modelsim 6. 1f,综合工具为 Xilinx ISE 8. 1,开发板为 Xilinx 的 DS-KIT-4VLX60 MB.

本文在 Xilinx 的开发板 DS-KIT-4VLX60 MB 上利用 Xilinx FPGA XC4VLX60-10 FF1148C 建立了基于 LEON3^[11]处理器的 AUMCC 体系以及基于总线共享 Cache 结构(SCA)的原形平台. 首先利用 VHDL 语言设计实现多通道 Cache 控制器,并将其嵌入编译到 Grlib 库的总线控制器中,实例化该库中三端口 SRAM 生成多端口 Cache 存储阵列. 将 LEON3 软核和 VHDL 设计文件一起利用

make 工具编译生成库 IP 核. 之后由 XST 综合工具生成的 FPGA 网表文件并由 Xilinx 公司的布局布线和下载工具生成相应的 SOF 文件,通过 JTAG 端口将 SOF 文件下载到 FPGA 上进行硬件配置. 使用 LEON3 的基于 GCC 的 LECCS 交叉编译系统对基准程序进行编译,得到二进制代码以后,通过串行口下载到 FPGA 开发板上执行,系统配置参数如图 4 所示.

3.2 AUMCC 体系性能测试

本文以基准程序集 MediaBench^[12]和 OO-PACK^[13]为基础,从不同的分类中选择了 7 个典型的基准测试程序 basicmath, bitcount, blowfish, matrix, dijkstra, fft 和 stringsearch. 测试结果给出了 8 核 AUMCC 体系及 SCA 体系中各基准测试程序运行时间(如图 5 所示). 相比 SCA 结构,AUM-

CC 体系中不存在 L1 Cache 的一致性维护开销,而且 L2 Cache 的多通道支持高效的并行访问,因而 AUMCC 体系中各种测试程序的运行性能均有所提高,平均加速比可以达到 37%. 通信与交互比较多的应用在两种体系中的性能都最好,说明共享 Cache 结构能满足核间高速、低延迟的通信

需求. 对于通信和并行交互比较多的测试程序, AUMCC 的性能加速比更高, 分别达到 1.52, 1.46,因而相对 SCA 体系, AUMCC 体系可以提供更高效的并行通信性能,支持核间共享数据的高效并行传输.

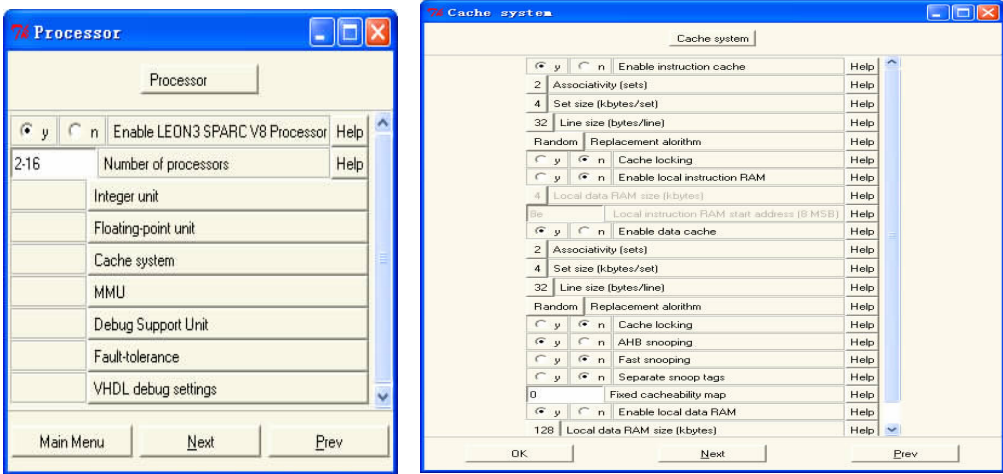


图 4 系统参数配置

3.3 系统扩展性测试分析

在消除了多通道 Cache 的访问冲突之后,所有核可以同时访问不同的存储体. 理论上, AUMCC 的访问带宽可以随着核数量的增长而线性增长,具有良好的可扩展性. 为此,定义 AUMCC 体系的带宽 B 为在某个单位时间内所能完成的访问请求的数目,模拟时以每拍实际完成的访问请求的数目 DPC (Demands per cycle) 来衡量共享数据 Cache 的带宽, N 为核的数量. 7 个基准程序的扩展性模拟结果如图 6 所示.

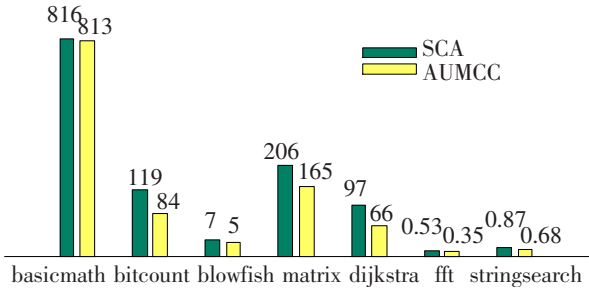


图 5 两种体系下基准程序得运行时间

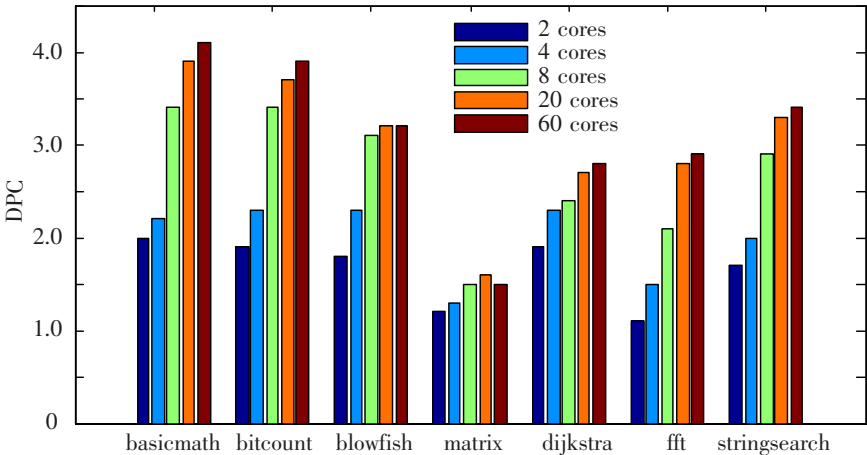


图 6 AUMCC 系统有效访问带宽与核数量的关系

由实验结果可见,当 $N < 12$ 时, B 快速增长,与 N 呈近似的线性关系. 随着 N 的进一步增大,核与存储体之间的控制逻辑开销、互连总线和交叉开关端口数量以 $O(N^2)$ 数量级增长, AUMCC

的工作频率开始下降,访问延迟越来越大,有效访问带宽增长十分缓慢. 当 $N < 8$ 时,带宽的平均增长率为 30%; 当 $8 < N < 12$ 时,平均增长率为 13%; 当 $N > 12$ 时,带宽增长率低于 5%. 因此,

AUMCC 适合于 12 核以内的多核 CMP. 当核数量超过 8 核以上时, 将以 4 核为一个超节点进行结构扩展. 超节点内部采用 AUMCC 实现紧耦合的数据传输, 超节点之间通过片上网络或者其他共享存储结构进行数据传输.

4 结 论

1) 提出一种面向多核系统的使用多通道 Cache 作为 L2 Cache 的高效存储架构 AUMCC.

2) 针对性能需求设计了多通道 Cache 的体系, 其两种分离访问模式不仅简化了 L1 Cache 的一致性维护开销, 同时使得 AUMCC 可以分别配置成共享 L2 Cache 和私有 L2 Cache 两种架构.

3) 采用 LEON3 处理器基于 FPGA 构建了 AUMCC 体系原型系统并进行了系统性能仿真模拟, 结果表明, AUMCC 体系相对于传统基于总线共享的存储架构而言有 37% 的性能提升. 同时系统扩展性试验表明, 该体系具有明显的层次化特点, 易于扩展.

参考文献:

- [1] OLUKOTUN K, HAMMOND L. QUEUE: The future of microprocessors[J]. ACM, 2005, 3(7):26–29.
- [2] Costin Iancu, Steven Hofmeyr. Runtime optimization of vector operations on large scale SMP clusters[C]//Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques. New York, NY: ACM, 2008:122–132.
- [3] HOEFLE T, GOTTSCHLING P, LUMSDAINE A. Leveraging non-blocking collective communication in high-performance applications[C]//Proceedings of the Twentieth Annual Symposium on Parallelism in Algorithms and Architectures. New York, NY: ACM, 2008:113–115.
- [4] LEE Jaejin, SEO Sangmin, KIM Chihun, *et al.* COM-IC: A coherent shared memory interface for cell be[C]//Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques. New York, NY: ACM, 2008:303–314.
- [5] OZTURK O, KANDEMIR M, CHEN G, *et al.* Customized on-chip memories for embedded chip multiprocessors[C]//Proceedings of the 2005 Asia and South Pacific Design Automation Conference. New York, NY: ACM, 2005: 743–748.
- [6] Haroon-Ur-Rashid, SHI Feng, JI Weixing, *et al.* TriBA: A novel scalable architecture for high performance parallel computing applications[C]//Proceedings of the 6th Conference on WSEAS International Conference on Applied Computer Science. Stevens Point, Wisconsin: World Scientific and Engineering Academy and Society (WSEAS), 2007:396–401.
- [7] AMD. AMD Athlon™64 处理器[EB/OL]. [2005–03–08]. [http://www.amd.com.cn/CHCN/Processors/Product Information/0,30_118_9484,00.html](http://www.amd.com.cn/CHCN/Processors/Product%20Information/0,30_118_9484,00.html).
- [8] OZTURK O, KANDEMIR M. Data replication in banked DRAMs for reducing energy consumption[C]//Proceedings of the 7th International Symposium on Quality Electronic Design. Washington, DC: IEEE Computer Society, 2006:551–556.
- [9] 刘彩霞, 石峰, 薛立诚, 等. 一种块传输多端口存储控制器: 中国, 200710098503. 7[P/OL]. [2007–10–09]. <http://search.sipo.gov.cn/sipo/zljs/hyjs-jieguo.jsp>.
- [10] HENNESSY J L, PATTERSON D A. 计算机体系结构: 量化研究方法[M]. 3 版. 北京: 机械工业出版社, 2002.
- [11] Glib-gpl-1.0.19-b3188. tar. gz[OL]. [2008–09–30]. http://www.gaisler.com/cms/index.php?option=com_content&task=view&id=156&Itemid=104.
- [12] GUTHAUS M R, RINGENBERG J S, EMST D, *et al.* MiBench: A free, commercially representative embedded benchmark suite[C]//Proceedings of the 4th Annual Workshop on Workload Characterization. Washington, DC: IEEE Computer Society, 2001:3–14.
- [13] Evaluating Performance and Power of Object-Oriented vs. Procedural Programming in Embedded Processors. [EB/OL]. [2008–9–19]. <http://www.auto.tuwien.ac.at/AE2002/Presentations/chatzigeorgion/ADA.ppt>.

(编辑 张 红)