

文章编号: 1001- 2486(2009) 05- 0019- 05

# 一种面向异构多核 DSP 的数据流传输控制引擎设计与评测\*

汪东, 陈书明

(国防科技大学 计算机学院, 湖南 长沙 410073)

**摘要:** 根据多核 DSP 的应用特点, 介绍了一种面向异构多核 DSP 的数据流传输控制引擎(DSTCE)的结构与设计, 并实现了基于 DSTCE 的数据流前瞻传输技术。该技术以可编程的 DSTCE 模块为核心, 实现了超节点内各个端口之间以及超节点之间的高效数据流传输, 而且将传输过程与多核间的数据前瞻和编程模型有机结合起来。该技术避免了复杂的 Cache 一致性问题, 显著提高了大规模多核 DSP 计算与访存的并行性。实验结果表明, 采用 DSTCE 的多核 DSP 单个超节点的性能平均提高了 22%, 两个超节点情况下基于 DSTCE 结构的系统比基于 CG-NUMA 结构的系统性能平均提高了 14%。

**关键词:** 多核 DSP; 数据流; 传输控制; 数据前瞻

中图分类号: TP368.1 文献标识码: A

## Design and Evaluation of a Data Streams Transmission Control Engine Customized for Heterogeneous MC-DSPs

WANG Dong, CHEN Shu-ming

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract:** This paper introduces a large-scale parallel MC-DSP prototype, PolyDSP constructed with heterogeneous super-nodes, and proposes a data speculation technique based on customized data streams transmission control engine (DSTCE). This technique implements high efficient data streams transmissions among different data ports within a super-node and transmissions between neighbor super-nodes. The superiority of this technique is that it can combine a data speculation solution with the programming model of PolyDSP, avoid complex cache coherence problems, thus improving the parallelism between data processing and memory accessing for MC-DSPs significantly. Our simulation results show that, the performance of single super-node of PolyDSP is improved by 22% on average, and the performance of PolyDSP based on DSTCE is 14% higher than that of PolyDSP based on CG-NUMA structure with two super-nodes.

**Key words:** multi-core DSP; data streams; transmission control; data speculation

### 1 相关工作

面向多处理器的数据前瞻技术通过将访存操作与计算过程重叠进行, 能够非常有效地隐藏远程访问延迟, 主要包括一致性预测技术、数据预取技术和数据前向技术三类。其中, 一致性预测技术<sup>[1-2]</sup>能够根据程序执行的历史信息, 预测下一条一致性消息(或操作), 提前执行一致性协议。在数据预取技术中, 由消费者处理器在使用数据之前请求数据, 通过预取指令等方式将远程数据预取到距离自己较近的本地 Cache 或存储器内。在数据前向技术中<sup>[3-4]</sup>, 由生产者处理器将数据拷贝主动发送到一个或多个消费者处理器的局部存储器中。数据预取和数据前向技术都能够大幅度减少远程访问延迟, 但是必须防止将过多的数据取入消费者的本地 Cache 内而造成污染, 或者由于数据传输时机控制不好而发生阻塞。

多核 DSP 是新一代无线通信、多媒体处理等高性能嵌入式应用的核心处理器。大量分析表明<sup>[5-7]</sup>, 多核 DSP 的应用主要体现出实时性强、功耗要求低、数据流特征显著等特点。多核 DSP 存储通路的设计应该针对上述应用特征进行优化和权衡。

\* 收稿日期: 2009-07-03

基金项目: 国家 863 计划资助项目(2007AA01Z108); 教育部“高性能微处理器技术”创新团队资助项目(IRT0614)

作者简介: 汪东(1979-), 男, 助理研究员, 博士。

## 2 PolyDSP: 基于异构超节点的大规模多核 DSP

组成 PolyDSP 的单个异构超节点 SDSP 由 4 个 32 位浮点 DSP 核与 1 个 32 位 RISC 处理器内核构成, 如图 1 所示。其中, DSP 核采用八流出超长指令字结构的 32 位浮点 DSP 内核“YHFF-D3B”<sup>[8]</sup>。每个 DSP 核私有 4KB 一级指令 Cache (L1P) 和一级数据 Cache (L1D)。4 个 DSP 核共享总容量为 320KB 的二级 Cache (L2), L2 的一部分或全部可以配置为 SRAM 存储器, 映射全局地址空间来使用。RISC 核采用兼容 SPARC 指令集的 32 位 LEON 处理器, 主要用于实现前端人机接口、数据 I/O 和超节点之间的通信控制。4 个 DSP 核则主要负责后端数据处理与计算加速。利用四维互连接口 (NI) 可以实现超节点之间的互连与扩展, 构成包含多个超节点的大规模多核 DSP。

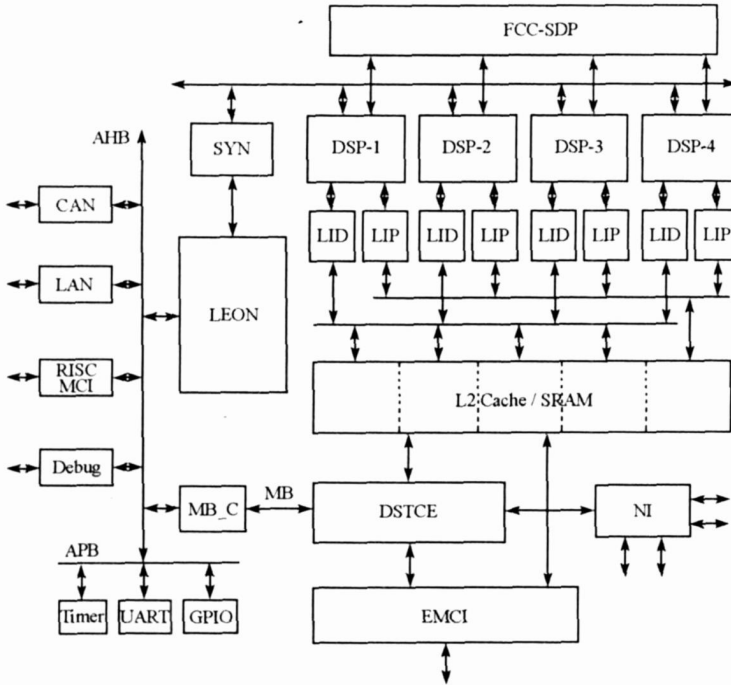


图 1 PolyDSP 单个超节点的体系结构

Fig. 1 The architecture of single super-node of PolyDSP

每个超节点都具有自己独立的地址空间。超节点内采用基于目录的协议维护 4 个 DSP 核的 Cache 一致性, 超节点之间采用消息传递的方式, 通过片上网络实现数据通信和同步控制。本文利用 DSTCE 对超节点内部和超节点之间的数据流传输进行了优化设计, 有效提高了 PolyDSP 结构的可扩展性。

## 3 DSTCE 的结构与设计

数据流传输控制引擎 (DSTCE) 是 PolyDSP 内置的一个数据流传输控制模块, 在系统中扮演数据流“搬运工”的角色。它不但支持超节点内部各个 I/O 端口之间、存储器之间以及超节点到超节点之间的数据流传输, 而且将超节点之间的数据迁移与任务同步有机结合起来, 能够很好地支持数据流的前瞻传输, 提高了全系统的计算并行性和可扩展性。

DSTCE 主要由同步事件触发寄存器 ETR、传输参数存储器 TPRAM、TPRAM 控制器、传输请求仲裁模块、请求队列 Q1/Q2/Q3、传输控制模块 TC、数据流缓冲 DSB 和传输中断寄存器 TIR 组成, 如图 2 所示。DSTCE 能够响应三类传输请求: 16 个同步事件触发的传输请求、RISC 核发起的传输请求和 NI 接口发出的传输请求。这三类的响应优先级依次降低。

ETR 用于记录同步事件, TPRAM 用于存储各个同步事件对应的传输控制参数。当某个同步事件发生时, 从 TPRAM 中读出预先配置好的一组传输控制参数, 提交给仲裁模块进行处理。仲裁模块根据响

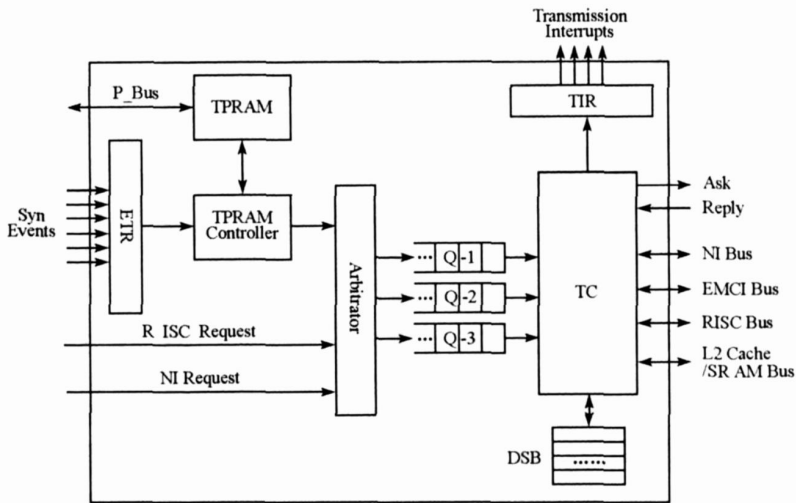


图2 DSTCE 的组成结构

Fig. 2 The block diagram of DSTCE

应优先级将传输请求分别送入相应的队列。TC 模块内含 3 个并行的请求处理子模块,每次从请求队列的头部开始处理传输请求。当一次传输请求处理完成后,TC 模块向 TIR 发出中断信号,通知处理器执行后续响应和处理工作。

数据单元是 DSTCE 传输的最小单位,可以是一个字节、半字或字。多个数据单元组成一帧,多个帧构成二维数据流。DSTCE 支持一维的数据流传输,即每次传输一个数据单元或一个帧;也支持二维数据流的传输,即每次触发多个帧的传输。

#### 4 基于 DSTCE 的数据流前瞻传输

DSTCE 在开始传输每一帧数据之前,TC 模块和目标端口之间通过查询、应答的方式建立传输通道,然后根据控制参数设定的传输模式,连续从源端口读出数据,发送给目标端口。本文对二维数据流的传输过程进行了优化设计,利用数据流缓冲器实现多帧流水传输。

PolyDSP 的编程模型将程序的执行环境划分为多核 DSP 操作系统、子任务、模块和函数 4 个抽象层次。图 3 给出了一个由两个子任务构成的多核 DSP 数据流图(DFG)的实例。其中子任务 1 包含  $f_{1a}$ 、 $f_{1b}$ 、 $f_{1c}$ 、 $f_{1d}$  四个模块,对原始数据 A 进行处理,得到中间结果 B。子任务 2 包含  $f_{2a}$ 、 $f_{2b}$ 、 $f_{2c}$ 、 $f_{2d}$  四个模块,对中间结果 B 进行处理,得到最终结果 C。

将该数据流图映射到 PolyDSP 的 1 号和 2 号超节点上。其中原始数据 A 存放在超节点 1 的片外存储器中,中间结果从 1 号超节点传输给 2 号超节点,最终结果 C 通过超节点 2 的 LAN 接口输出。在这种映射方式下,DSTCE 的作用如下:

- (1) DSTCE-1 响应 RISC 核的请求,将原始数据 A 和程序代码从外存搬移到 1 号超节点的本地共享 SRAM 中,然后通过传输中断启动子任务 1 的执行;
- (2) 当子任务 1 执行完毕时,DSTCE-1 响应 DSP-4 的请求,将超节点 1 内的中间结果 B 通过互连接口 NI 搬移到超节点 2 的本地 SRAM 中;
- (3) 当子任务 2 执行完毕时,DSTCE-2 响应超节点 2 的 DSP-4 的同步事件,将最终结果 C 通过 LAN 接口输出。

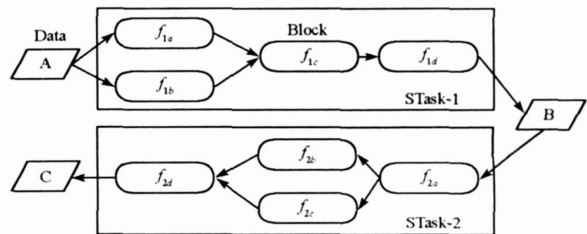


图3 一个包含两个子任务的 PolyDSP 数据流图例子

Fig. 3 A DFG case of PolyDSP containing two sub-tasks

利用 DSTCE 的事件触发功能,可以非常方便地实现数据流预取和数据流前向传输。本文为 PolyDSP 设计了两个前瞻操作原语,分别是 DSTCE 预取原语:

$$\text{DSTCE\_Prefetch Array, SRC\_ID(Addr), Size;}$$

和 DSTCE 前向原语:

$$\text{DSTCE\_Forward Array, DST\_ID(Addr), Size;}$$

DSTCE\_Prefetch 用于请求 DSTCE 从超节点 SRC\_ID 的地址 Addr 开始预取大小为 Size 的数据块到本地共享存储变量 Array 中。DSTCE\_Forward 用于请求 DSTCE 将本地共享存储变量 Array 前向传输到超节点 DST\_ID 的地址 Addr 开始大小为 Size 的一片存储空间中。

如上例,在开始处理数据 A 的第  $i$  块之前,可以在 1 号超节点的 DSP 程序中调用一条 DSTCE\_Prefetch 原语,启动 DSTCE 从片外存储器中预取第  $i+1$  块数据到片内。另外,每当 1 号超节点产生中间结果 B 的一块数据时,可以在 DSP 程序中调用一条 DSTCE\_Forward 原语,启动 DSTCE-1 将 B 的这块数据提前传送到 2 号超节点,实现数据流传输过程和 DSP 计算过程的部分并行化,隐藏一定的访存延迟。

## 5 性能评测

首先构建了 PolyDSP 单个超节点的 C 语言模拟器 SDSP-Sim。该模拟器是一个周期精确的行为级模拟器,能够执行经过编译优化和任务分配的 C 程序,DSP 核之间采用共享变量的方式进行通信,DSP 核与 RISC 核之间采用进程通信的方式进行同步。进一步地,利用消息传递接口函数实现多个超节点之间的通信,构成包含多个超节点的系统模拟器 PolyDSP\_Sim。

表 1 用于性能评测的 Benchmark

Tab. 1 The benchmarks selected for performance evaluation

Benchmarks	Comments
ADPCM_e	The audio encoder for adaptive differential pulse code modulation.
PEGWIT	A program for public key encryption and authentication.
RASTA	A program for speech recognition that supports PLP, RASTA, and Jaf RASTA.
GSM	European GSM 06 10 provisional standard for full-rate speech transcoding.
2MP3_d	Two ways of MP3 audio decoder with 128Kbps of code rate.
MPEG2_e	MPEG-2 video encoder with 352×288 resolution (CIF, 4:2:0) and 25fps of frame rate.

以 DSP 专用评测程序集 MediaBench 为基础,选择了 8 组典型的多媒体与无线通信类的测试程序,如表 1 所示。采用手工划分的方式对测试程序进行了子任务划分和分配,然后利用 YHFT-D3B 单核处理器的 C 语言编译器对每个子任务分别进行编译优化。在进行子任务映射时,主要采用“生产—消费”模式,使 DSP 核之间以及超节点之间构成流水操作。

### 5.1 单个超节点下的性能对比

首先对比了采用 DSTCE 前瞻传输前后,单个超节点性能的变化情况。两种实验环境如下:

- 在不采用 DSTCE 的结构中,将 L2 全部配置为 Cache 模式,所有访存操作均通过 L2 Cache 直接请求 EMCI 接口;
- 在采用 DSTCE 的结构中,将 L2 的一半容量配置为 SRAM,利用 DSTCE 的前瞻操作原语对可以分块处理的数据进行优化,在外存和 SRAM 之间进行前瞻传输。

在上述两种实验环境下,利用模拟器模拟运行了 6 组测试程序,分别得到不采用 DSTCE 情况下的执行时间和启用 DSTCE 前瞻传输方式的执行时间,然后计算出后者相对于前者的计算加速,如图 4 所示。

实验结果表明,除了 PEGWIT 程序外,其他 5 个测试程序都表现出了不同程度的性能提升,平均加速比为 1.22。其中 MPEG2\_e 中的数据流最为密集,也可以进行分块处理,达到了最高的加速比 1.35。而 PEGWIT 中的不规则数据较多,不易分块处理,难以发挥 DSTCE 的优势,实验效果不理想。

### 5.2 与 CG-NUMA 结构的性能对比

对比了基于 DSTCE 和 NI 接口的 PolyDSP 扩展方案与基于 CG-NUMA 结构的 PolyDSP 扩展方案在性

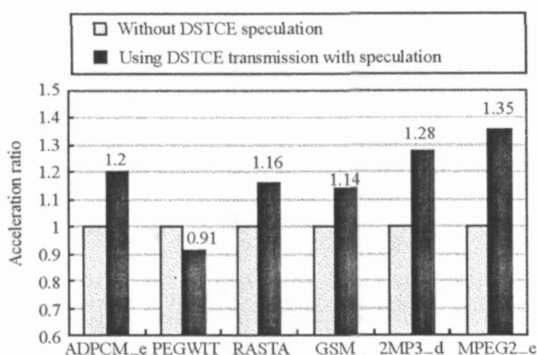


图4 使用 DSTCE 前瞻传输对单个超节点带来的性能提升

Fig. 4 The performance acceleration of single super-node when using DSTCE speculation

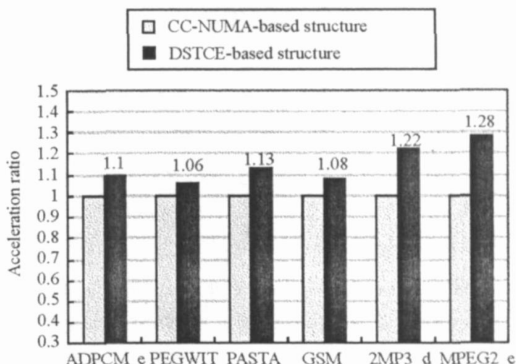


图5 两种 PolyDSP 结构下的性能对比

Fig. 5 Performance comparison of two structures of PolyDSP

能上的差异。在构建 CG-NUMA 结构的软件模拟器时,去掉了 DSTCE 引擎,将各个超节点的本地 L2 Cache 通过交叉开关连接在一起,采用基于写作废的目录协议维护各个超节点之间的 Cache 一致性。然后重新映射测试程序,在统一的地址空间内重新编译各个子任务程序。

由于 PolyDSP 系统比较复杂,为了提高软件模拟器的运行效率,仅仅对比了在两个超节点的情况下上述两种扩展方案的性能。6 组程序的性能对比结果如所示图 5 所示。

实验结果表明,在采用基于 DSTCE 和 NI 接口的两个超节点互连结构下,6 组测试程序的计算性能都高于基于 CG-NUMA 结构的扩展方案,平均加速比为 1.14,最高加速比达到了 1.28。

分析认为,采用支持数据流前瞻传输的 DSTCE 模块实现超节点内部和超节点之间的数据通信,无论对于单个超节点还是对于多个超节点都具有较好的性能提升,这主要归因于以下几点:

(1) DSTCE 针对数据流的特点进行了专门的设计优化,支持一维、二维数据流的传输,能够按照控制参数自动进行传输计数和地址更新;

(2) DSTCE 为编程提供了专用的预取和前向操作原语,支持数据流的前瞻传输,进一步提高了数据传输与计算的并行性;

(3) 在 DSTCE 的支持下,超节点内部采用耦合度较高的共享存储结构,而在超节点之间采用消息传递方式,更有利于程序员显式地管理数据流,提高并行编程的效率。

## 6 结束语

以 DSTCE 为核心,将事件触发、同步控制、前瞻传输等多个并行系统的要素有机结合起来,构建了一条高性能的多核 DSP 数据通路。今后,将就 DSTCE 对不同结构多核 DSP 的适应能力、数据传输优化等问题展开进一步的研究。

## 参考文献:

- [1] Lai A, Falsafi B. Memory Sharing Predictor: The Key to a Speculative Coherent DSM [C]//Proceedings of the 26<sup>th</sup> International Symposium on Computer Architecture (ISCA), U. S. A., 1999.
- [2] Kaxiras S, Goodman J R. Improving CG-NUMA Performance Using Instruction-based Prediction [C]//Proceedings of 5<sup>th</sup> International Symposium on High-performance Computer Architecture (HPCA), U. S. A., 1999.
- [3] Koufaty D A, Chen X, Poulsen D K, et al. Data Forwarding in Scalable Shared-memory Multiprocessors [J]. IEEE Transactions on Parallel and Distributed Systems, 1996, 7(12): 1250-1264.
- [4] Koufaty D, Torrellas J. Comparing Data Forwarding and Prefetching for Communication-induced Misses in Shared-memory MPs [C]//Proceedings of International Conference on Supercomputing, Australia, 1998: 53-60.
- [5] 王光,沈绪榜.一种用于多媒体流处理器的缓冲器体系结构设计[J].黑龙江大学自然科学学报,2005,22(2):248-253.
- [6] Ravikumar C P. Multiprocessor Architectures for Embedded System-on-chip Applications [C]//Proceedings of the 17<sup>th</sup> International Conference on VLSI Design (VLSI'D), India, 2004: 512-519.
- [7] Rutten M, Pol E J, van Eijndhoven J, et al. Dynamic Reconfiguration of Streaming Graphs on a Heterogeneous Multiprocessor Architecture [J]. SPIE Electronic Imaging: Embedded Processors for Multimedia and Communications, 2005, 5683: 17-18.
- [8] 陈书明,李振涛,万江华,等.“银河龙腾”高性能数字信号处理器研究进展[J].计算机研究与发展,2006,43(6):993-1000.