

采用 S-Tag 的 M-DSP 片上存储 DMA 访问优化*

鲁建壮¹, 孙书为¹, 陈胜刚¹, 陆文远²

(1. 国防科技大学 计算机学院, 湖南 长沙 410073; 2. 西安卫星测控中心, 陕西 西安 710043)

摘要:针对自主设计的 M-DSP, 提出并设计实现了一种基于 Tag 副本(S-Tag)的片上 SRAM DMA 访问数据相关性维护机制, 该机制以流水化方式实现, 在基本对 CPU 无打扰的前提下, 有效支撑了 DMA 数据的无阻塞传递。仿真和芯片实测结果表明, 该机制硬件开销较小, 并在有效带宽和带宽利用率上均优于已有典型同类芯片。

关键词: Cache + RAM 结构; S-Tag; DMA 传输; 数据一致性

中图分类号: TP302 **文献标志码:** A **文章编号:** 1001-2486(2018)06-112-06

S-Tag based DMA optimization for on-chip memory in M-DSP

LU Jianzhuang¹, SUN Shuwei¹, CHEN Shenggang¹, LU Wenyuan²

(1. College of Computer, National University of Defense Technology, Changsha 410073, China;

2. Xi'an Satellite Control Center, Xi'an 710043, China)

Abstract: The S-Tag (Shadow Tag) mechanism of SRAM (static random access memory) data consistency maintaining for DMA (direct memory access) accessing was proposed for the independent design of M-DSP (multi-core digital signal processor). The pipelining implementation can efficiently support DMA non-blocking data transfer, and release the CPU. Experimental results and tests in real chips show that the proposed mechanism outperforms the state-of-art ones with respect to bandwidth and bandwidth utilization while keeping relatively lower hardware cost.

Key words: Cache + RAM structure; S-Tag; DMA data transfer; data coherence

高性能数字信号处理器 (Digital Signal Processor, DSP) 在应对处理器领域普遍存在的处理性能提升过快而存储性能不足的“存储墙”问题时, 同样设计和实践了通用文献 [1] 多核结构和多层次存储组织方式, 但是由于其应用领域和使用方式的不同, 其存储架构和角色功能上也具有明显的特殊性。

目前采用“Cache + RAM”的多级存储架构是高性能 DSP 的主流^[2], 通常一级存储作为 Cache 实现, 容量小、速度快用以弥补存储系统与处理能力之间的性能差异; 二级存储采用 Cache/RAM 共存或可配置的混合结构, 这样既可以实现外部大批量数据的缓冲和较快速访问, 又可以便捷地支持实地址编程、显示的任务/数据调度和时间开销敏感的实时类程序等应用。直接存储器存取 (Direct Memory Access, DMA) 访问是处理器后台数据搬移的主要方式, 在具有 Cache 的存储系统中必须解决 DMA 传输过程中的数据相关性问题, 即 DMA 传递的必须是最新数据。

解决多核结构中 Cache 的数据相关性^[3]问题, 多侧重提高 Cache 性能。文献 [4] 中提出冗余 Cache 相关性结构, 主要面向通用多核处理器; 文献 [5] 与应用结合对替换策略调整, 降低末级 Cache 的失效率, 进而提高了系统整体性能; 文献 [6] 设计了软硬件结合的局部存储与 Cache 一致性协议, 但都不能解决多核 DSP 中后台实时数据传输中的数据相关性问题。文献 [7] 提出了 Qlink 机制实现多核数据传输, 增加了数据转换环节且面向内核的数据传输均需要侦听 L1, 特定条件下会对 CPU 执行产生较多的影响; 文献 [8] 采用冗余 Tag 的方式实现 DMA 传输中的数据相关性问题, 但没有对 Tag 自身的一致性进行优化。

上述研究对于提高 DSP 中临近内核的“Cache + RAM”的 DMA 传输速度贡献都不大, 为支持实时应用、提高 DMA 效率, 需采用更为高效的相关性维护机制, 在实现快速 DMA 的同时减少对内核/一级 Cache 的打扰。

本文以自主设计的多核 DSP, 即 M-DSP 为基

* 收稿日期: 2017-09-16

基金项目: 国家自然科学基金资助项目 (61402499, 61402500, 61602493)

作者简介: 鲁建壮 (1977—), 男, 河北深州人, 副研究员, 博士, 硕士生导师, E-mail: lujz1977@163.com

础,深入研究“Cache + RAM”结构下的 DMA 访问流程和操作特点,提出并设计实现了一种基于 Tag 副本——S-Tag 的片上 SRAM DMA 访问数据相关性维护机制,该机制以流水化方式实现,在尽量降低对 CPU 打扰的情况下,有效支撑了 DMA 方式的高效数据传递。

1 M-DSP 的存储结构和 DMA 访问

1.1 M-DSP 的存储结构

M-DSP 采用经典的存储结构(见图 1):由核内两级缓存和外部共享存储器构成。一级缓存采用哈佛结构,分为程序存储器(L1P)和数据存储器(L1D),与 CPU 同频,主要解决速度问题,在命中的情况下无须停顿内核的流水线。

二级缓存是程序与数据混合式存储器(L2),是一级缓存与外部存储器的枢纽,可配置成“Cache + SRAM 模式”;L2 容量较大,M-DSP 的单核 Cache 最大可配置为 512 KB;内核对 L2 的访问采用流水化实现,其访问延迟为 2~8 个系统时钟周期不等。L2 Cache 属核内私有缓存,其他核心不可见;L2 静态随机存储器(Static Random Access Memory, SRAM)具有全局地址,支持核间共享存储,其他核心可通过 DMA 或全局地址访问本核的 L2 SRAM。

核外共享存储是 M-DSP 的第三级存储,可以通过配置使其部分空间可缓存,进入 L1 或者 L2 Cache,它包括片内多核共享存储器和片外存储器等,其中片内共享存储可达 4 MB,访问延迟在 20 个系统时钟周期左右;片外访问延迟一般在 100 个系统时钟周期以上。

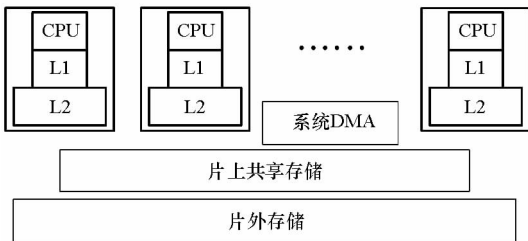


图 1 M-DSP 结构简图

Fig. 1 Structure of M-DSP

在以数据密集型和实时性为特征的 DSP 类应用中,以 DMA 方式访问可编址片内 RAM,将数据放在在内核就近的存储空间,在内核完成处理后再通过 DMA 将处理结果移走是典型的处理模式。此模式下,挖掘数据处理和数据传输之间的并行性是保证整体性能的关键。

1.2 典型 DMA 访问模式

双缓冲模式是一种常用的数据处理方式:将存储空间分为 A、B 两组输入/输出缓冲, DMA 对一组缓冲导入导出,内核在同一时间段内处理另一组数据。在类 M-DSP 的高性能 DSP 中,通常在 L2 设置缓冲,实现“乒乓”或多核“生产者-消费者”模式的流式数据处理,如图 2 所示。

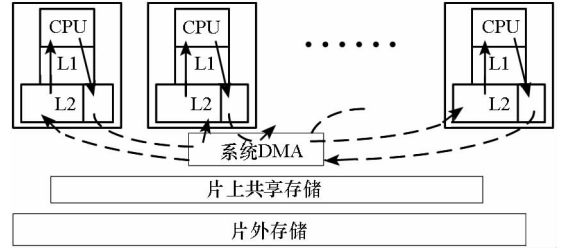


图 2 多核“生产者-消费者”模式

Fig. 2 Producer-consum pattern in multi-core

核内采用两级存储结构,使内核处理集中访问 L1 Cache, DMA 主要访问 L2 SRAM,从而使数据处理和传输获得更好的并行性。但 L1 Cache 会导致数据一致性问题:DMA 从 L2 SRAM 读取数据时,如该数据在 L1D Cache 中存在副本且已经被 CPU 修改, DMA 仅访问 L2 会得到旧数据; DMA 向 L2 SRAM 写入数据时,数据如在 L1D Cache 中有副本, DMA 写入不更新/作废 L1D, CPU 处理旧数据将造成错误。

为了避免上述情况,需采用特定机制维护两级存储之间的数据一致性,即 DMA 在访问 L2 时 SRAM 时完成如下功能:

- 1) 对 L2 进行读操作时,如果数据不在 L1D Cache 或者在 L1D Cache 中,但是未被修改,直接读取 L2,否则从 L1D 中获取数据(即侦听读);
- 2) 对 L2 进行写操作时,如果数据不在 L1D 中,直接写入 L2,否则写入 L2 的同时把数据写入 L1D(即侦听写)。

为此 DMA 访问 L2 时需额外完成两个功能:一是判别数据是否在 L1D 中,二是从 L1D 中获取数据或者将新数据更新到 L1D 中。传统的处理方式是直接查询 L1D 的 Tag 判别数据状态,这样每个 DMA 请求都会访问 L1D Tag,形成与 CPU 访问的竞争,多流出的 CPU 微体系结构此情况会进一步恶化,影响 CPU 处理效率。从实现角度看,查询 L1D Tag 属跨部件操作,在无须与 L1D 进行数据传递时会有 2~3 个周期的开销,有数据交互时一般有 4 周期以上的开销,会显著加长 DMA 访问的处理延迟。

为解决上述问题, M-DSP 为 L2 设计实现了基于 S-Tag 的流水化 DMA 处理机制。

2 基于 S-Tag 的 DMA 访问优化与实现

2.1 基于 S-Tag 的 DMA 访问

在 L2 中设置一个 L1D Tag 的副本即 S-Tag, 与 L1D Tag 同步记录 L2 SRAM 中数据在 L1D 中的状态: 是否存在有效副本、副本是否为脏(被 CPU 修改)。每个 DMA 访问首先查询 S-Tag, 确定相应数据是否在 L1D Cache 中有副本、是否为脏; 而后对于 DMA 读, 若请求数据在 L1D 中存在副本且为脏, 则 L2 发送监听 - 读到 L1D, 将最新数据返回给 DMA, 否则直接从 L2 SRAM 读取数据; 对于 DMA 写, 所写数据在 L1D 中存在副本, L2 将发出监听 - 写命令到 L1D, 同时对 L1D 和 L2 SRAM 进行数据更新, 否则仅更新 L2 SRAM。

无论是对于 DMA 读还是写, L1D 中的数据状态保持不变, 也不进行 Cache 行作废或写回操作, 从而减少 CPU 访问时的数据缺失。

2.2 DMA 访问的流水化实现

结合 M-DSP 的结构特点和功能需求, DMA 访问 L2 SRAM 采用 5 站流水化实现, 功能划分如图 3 所示。

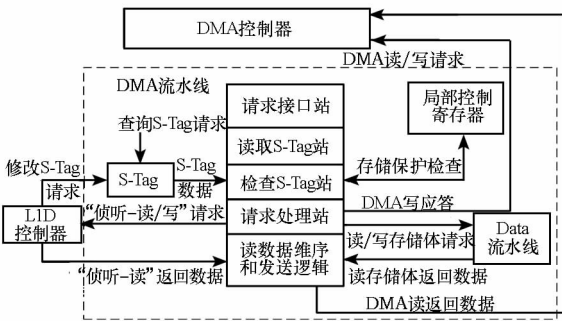


图 3 L2 的 DMA 请求处理流水线

Fig. 3 Pipeline of DMA accessing in L2

“请求接口站”接收来自 DMA 的读写请求, 并将请求参数缓存、拆解而后发至下一站。

“读取 S-Tag 站”接收上一站的 DMA 读写请求参数, 产生查询 S-Tag 请求, 并完成查询 S-Tag 和 L1D 修改 S-Tag 的请求仲裁。

“检查 S-Tag 站”接收上一站的 DMA 请求参数、S-Tag 的查询结果, 完成该请求在 L1D 中的命中判别; 同时完成存储保护, 即 DMA 请求的权限检查。S-Tag 采用存储体实现, 其输出在查询请求的下一个节拍才能获得, 流水线停顿时读出和

判别结果不能进入本站, 为节省功耗、降低访问冲突, 读出结果在站间缓存。

“请求处理站”完成对 DMA 请求的处理。如果权限异常, 则仅产生给 DMA 的反馈, 不进行真实读写操作; 否则, 当满足侦听条件时, 发出“侦听 - 读”或“侦听 - 写”请求, 对于 DMA 写和不满

足侦听条件的读向 L2 的 Data 流水线发出写或读请求。向 DMA 发出写响应或者传递读数据返回参数给下一站。

2.3 S-Tag 的相关性维护

S-Tag 是 L1D Tag 的副本, 两者之间同样存在一致性问题: L1D 发生行缺失时会更新 L1D Tag, 此时需要保证 S-Tag 及时更新, 使后续的 DMA 请求能够做出正确的侦听判别; 反之如果某 DMA 请求已经查询 S-Tag 的状态, L1D 应保证相关的数据行状态不变直至 DMA 完成必要的侦听请求。例如, L1D 发生行缺失准备读取 B 地址的数据, 并将该行原有的 A 地址脏数据替换出去, 同时 DMA 写操作查询并发现了 A 地址的数据在 L1D 中存在副本, 正确的结果应该是将 DMA 写数据(不是一个完整 L1D 行)覆盖到 L1D 回写的 A 地址所在行, 并最终写到 L2, 如果此时不能正确维护被操作 Cache 行的状态, 则可能 L1D 将原有的 A 地址脏数据写到 L2, 而 DMA“侦听 - 写”数据错误地写入该 Cache 行新返回的 B 地址数据, 造成错误。

L1D 的缺失请求和 DMA 请求来源不同, 访问时机和访问地址完全独立, 两者之间的访问冲突难以避免, 流水化实现使 S-Tag 的查询、侦听操作以及 L1D 行缺失过程中对 Tag 的修改都成为多周期操作, 为维护 Tag 和数据的一致性需要保证操作原子性。在多核交叉数据搬移中, 还需避免原子性操作导致的互锁问题。

为此, M-DSP 设计了交互式协同处理机制, 将 L1D 的 Tag 修改分为两个阶段: 作废 Tag 并写回脏数据, 接收新数据并修改 Tag。这两个阶段开始时分别会阻止新 DMA 请求查询 S-Tag, 在结束后释放。同时设置一个 DMA 流水线忙标志, 标志流水线中存在已查询 S-Tag 但未处理的侦听操作。具体处理流程如图 4 所示。

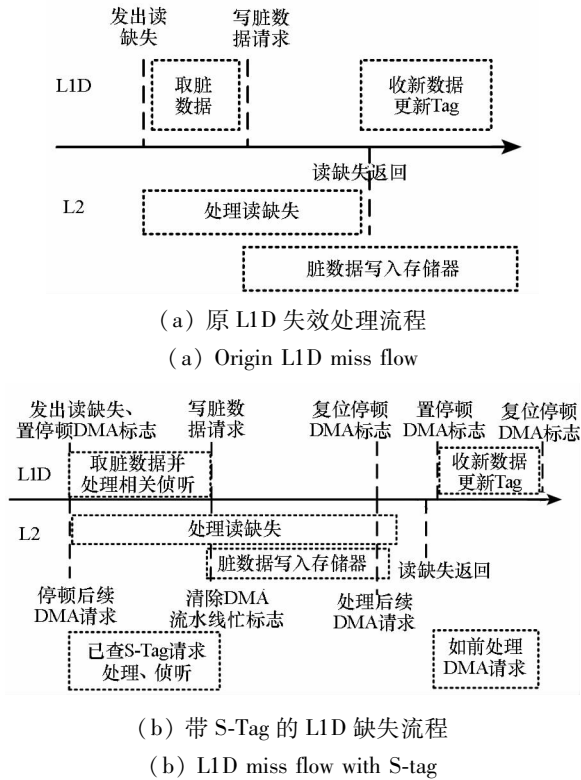


图 4 原 L1D 缺失与带 S-Tag 的 L1D 缺失流程对比
Fig. 4 Origin L1D miss flow vsersus L1D miss flow with S-tag

L1D 发生数据缺失时,发出缺失请求,同时设置停顿 DMA 请求标志,以阻止新 DMA 请求查询 L2 中 S-Tag,而后将本行的脏数据读出到牺牲缓冲;L2 继续完成已经查询了 S-Tag 的 DMA 请求,必要时发出侦听读或写操作,完成后清除 DMA 流水线忙标志;如果 DMA 请求命中缺失所在行,则针对牺牲缓冲完成读或写操作;L1D 监测到 DMA 不忙标志后,回写脏数据并作废 S-Tag 中相应行,最后清除停顿 DMA 标志。当 L1D 的读缺失数据返回时,L1D 按上述过程再次设置和清除停顿 DMA 标志。

将 L1D 读缺失过程中的 Tag 修改分为原内容作废和新内容写入,可有效简化单个原子性操作内的操作个数,避免死锁,但修改后可能增加 L1D 行缺失的开销。为避免这一情况,首先将 DMA 流水线忙设计为状态信号,在没有 DMA 请求时,不会增加 L1D 读缺失的处理时间;其次 L1D 和 L2 的处理尽可能并行,如读取脏数据和 DMA 流水线的处理;最后,图 4 是最复杂的情况,如果 L1D 行干净,读数据返回较快,L1D 的两次停顿将自动合并为一次。

2.4 读数据返回的维序处理

DMA 模块要求按请求顺序将读数据发送给 DMA,本机制中数据返回有三种来源:读取 L2

SRAM,与 CPU 请求竞争访问数据流水线并返回数据,可能因为 CPU 访问 L2 而产生停顿;无效读请求,直接返回;“侦听-读”L1D,数据返回受 L1D 忙闲状态影响,延迟不定。

本文设计了一种伴随流水加适时插入的处理机制,如图 5 所示。

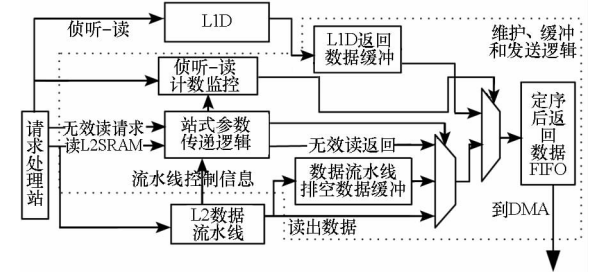


图 5 多源返回数据维序机制

Fig. 5 Ordering mechanism for multi-source data

图 5 中的“站式请求参数传递逻辑”,传输节拍与数据流水线相同,顺序接收读取 L2 SRAM/无效读的参数,依据数据流水线停/走控制信息同步推进相应请求的参数;返回数据先入先出(First In First Out, FIFO)满时停止 DMA 请求处理,排空数据流水线中的 DMA 请求,数据缓存在排空数据缓冲,以减少 DMA 访问对 CPU 的影响,采用上述机制可维护读 L2 SRAM 请求和无效请求的顺序。

“侦听-读”计数监控逻辑在“侦听-读”发出时,记录当前参数传递逻辑中有效 DMA 请求的个数,但不阻止后续请求进入该逻辑;当传递逻辑中有请求进入返回 FIFO 时计数器减 1;当 L1D 侦听数据返回后,需等计数器为 0,再将其写入返回 FIFO;若计数器已为 0,但 L1D 数据未返回,则阻止另两种请求进入返回 FIFO。

3 性能与开销分析

以 M-DSP 为基础,对本机制进行了性能评估和开销分析。M-DSP 部分设计参数如表 1 所示。

表 1 M-DSP 的存储配置

Tab. 1 Configuration of M-DSP's storage system

| 参数项 | 参数值 |
|-----------------|---------|
| L1D 行大小 | 64 B |
| L1D Cache 大小 | 32 KB |
| L2→L1D 读数据总线宽度 | 256 bit |
| L1D→L2 写数据总线宽度 | 256 bit |
| L2 SRAM 大小 | 512 KB |
| L2→EDMA 读数据总线宽度 | 128 bit |
| L2→EDMA 写数据总线宽度 | 128 bit |

3.1 性能分析

基于上述配置,本文构造了一个非流水化结构,分别针对数据仅在 L2 中、在 L1D 且需要侦听 L1D 等条件下的读写情况,对比测试了本文所提结构的流水化效果,运行情况如表 2 所示。流水化处理比非流水化处理至少可以获得 30% 左右的性能提升,读请求能够获得更高的性能提升,是因为读操作中请求可以突发,流水化数据返回及维序处理等会发挥更大作用。

表 2 流水化与非流水化开销对比

Tab. 2 Overhead of pipelined transfer versus non-pipelined

| 操作类型 | 设计方式 | 请求数目 | 耗时/ns | 比值 |
|------------------|------|--------|-------|-------|
| 写 L2 SRAM | 流水化 | 64 | 206 | 1.296 |
| | 非流水化 | 64 | 267 | |
| Burst 读 L2 SRAM | 流水化 | 4 × 16 | 136 | 3.853 |
| | 非流水化 | 4 × 16 | 524 | |
| “侦听 - 写” L1D | 流水化 | 64 | 206 | 1.311 |
| | 非流水化 | 64 | 270 | |
| Burst“侦听 - 读”L1D | 流水化 | 4 × 16 | 556 | 2.385 |
| | 非流水化 | 4 × 16 | 1326 | |

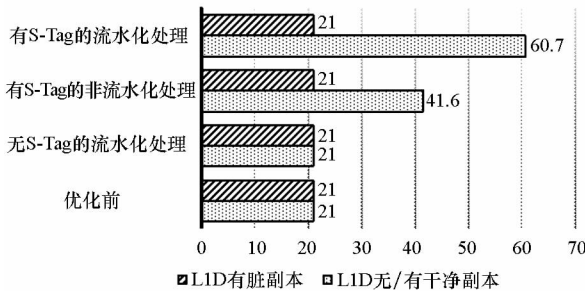
为进一步评测 DMA 访问的优化效果,本文

设置了四个对照组:无优化、不引入 S-Tag 有流水线优化、引入 S-Tag 无流水线优化、全优化,分别以 1 KB 和 32 KB 数据传输对优化前后的 DMA 数据访存进行了仿真,数据分为全部在 L1D 为脏,不在 L1D 或在 L1D 中仅有干净副本两种情况,仿真结果如图 6 所示。

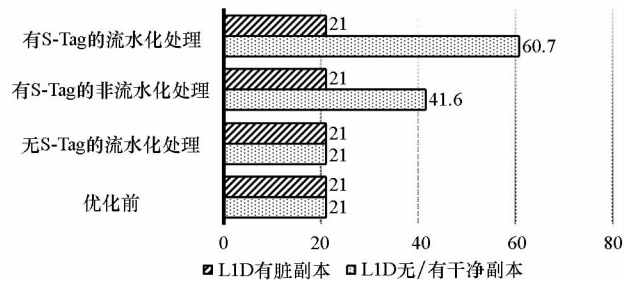
图 6 中“纵向访存”是指垂直存储层次的 DMA 传输,如将 L2 SRAM 中的数据搬移至多核共享存储器中。“横向访存”是指在水平存储层次的 DMA 传输,如将某核 L2 SRAM 中的数据搬移至其他核的 L2 SRAM 中。结果表明:

1)在同一优化条件下,当传输数据量较小时(1 KB),纵向与横向访存带宽基本保持一致;当数据量较大时(32 KB),DMA 纵向访存带宽略高于横向。这是由于小批量的数据在连续搬移时,突发传输之间的间隙开销较小且不易受到其他因素,如内核运行的影响,而大批量数据传输则相反。

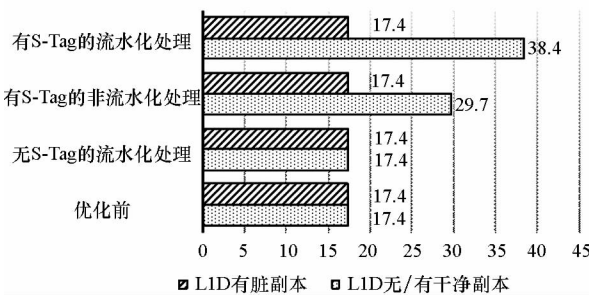
2)当被访问的数据在本地 L1D Cache 中存在脏副本,即 L2 SRAM 中的数据无效时,DMA 访存带宽的提升效果较小甚至没有提升,是由于 DMA 访问的最新数据全部在 L1D 中时,DMA 访存速率主要取决于 L1D Cache 侦听处理开销,而侦听操作未流水化,从而弱化了 S-Tag 和 L2 流水化效果。



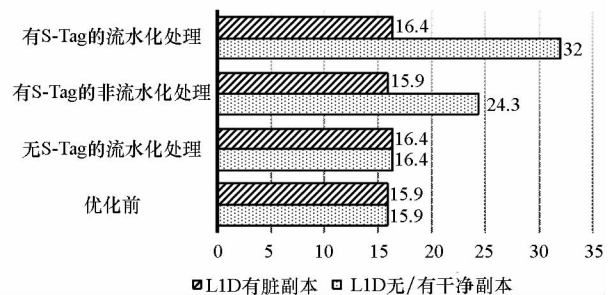
(a) DMA 纵向访存本地 L2 SRAM 1 KB 数据
(a) DMA vertically transfer 1 KB data with local L2



(b) DMA 横向访存本地 L2 SRAM 1 KB 数据
(b) DMA horizontally transfer 1 KB data with local L2



(c) DMA 纵向访存本地 L2 SRAM 32 KB 数据
(c) DMA vertically transfer 32 KB data with local L2



(d) DMA 横向访存本地 L2 SRAM 32 KB 数据
(d) DMA horizontally transfer 32 KB data with local L2

图 6 不同优化级别的 DMA 性能比较

Fig. 6 Comparison of performances of different optimization level

3)当L1D中不存在L2 SRAM中存储数据的脏副本时,DMA访存带宽为:全优化 > 引入S-Tag但无流水线优化 > 有流水线优化但不引入S-Tag ≥ 无优化。全优化时的DMA数据访存带宽是无优化时的2~3倍。由此表明本文所采用的DMA数据访存一致性优化手段是有效的。

3.2 硬件开销与芯片测试

M-DSP采用28 nm工艺实现,基于表1的配置对硬件开销进行了评估,采用本方案的L2总面积为1 870 781 μm^2 ,其中存储体约占94%;DMA处理逻辑面积为26 708 μm^2 ,约占L2总面积的1.4%;S-Tag相关逻辑面积开销为15 047 μm^2 (其中一半是S-Tag存储体),约占L2总面积的0.8%。由此可见,本机制的硬件开销较小,相对于L2总体面积基本可忽略。

采用本机制的M-DSP芯片已经成功流片,经测试各项功能正确,L2在宽温区稳定运行频率达1 GHz以上。成功流片后在板级用专门的批量数据(不小于512 KB)传输程序对L2的DMA性能进行评测,与文献[7]的C6678对比如表3所示。

表3 M-DSP与C6678的有效带宽对比

Tab.3 Effective bandwidth of M-DSP versus C6678

| 传输类型 | L2到片上共享存储 | | L2与DDR | | 核间L2传递 | 单核L2内部搬移 |
|-------|-----------|------|--------|------|--------|----------|
| | 上行 | 下行 | 上行 | 下行 | | |
| | | | | | | |
| M-DSP | 7.41 | 7.41 | 4.89 | 4.89 | 7.34 | 3.96 |
| C6678 | 4.93 | 4.93 | 4.91 | 4.92 | 4.94 | 4.51 |

就M-DSP自身而言,L2的DMA总线宽度128 bit,理论峰值带宽可达16 GB/s,但如果数据访问在L2中存在体冲突,将会插入1拍等待,再考虑到片上远程传输的控制开销等,在实际运行中获得7.2 GB/s以上的有效带宽达到了预期目标,而且利用率高于文献[6]的40%。

M-DSP与TI C6678结构相似、配置相同,有

效传输带宽优于或与之相当,其中双倍数据速率(Double Data Rate, DDR)传输主要受限于外端口,核内搬移带宽几乎减半是因为读写共用流水线。结果表明,本机制将DMA传输的数据相关性判别局部化优势明显。

4 结论

本文以M-DSP为基础,提出并设计实现的基于L1D Tag副本,即S-Tag的流水化片上SRAM DMA访问处理和数据相关性维护机制,对CPU干扰少,以较小的硬件开销实现了DMA数据的高效传输,提升了系统带宽。

参考文献(References)

- [1] Patterson D A, Hennessy J L. Computer architecture; a quantitative[M]. 5th ed. USA: Morgan Kaufmann Publishers Inc, 2007.
- [2] Sorin D, Hill M, Wood D. A primer on memory consistency and Cache coherence [M]. USA: Morgan & Claypool Publisher, 2011.
- [3] Khan O, Hoffmann H. ARCC: a case for an architecturally redundant cache-coherence architecture for large multicores[C]// Proceedings of 29th International Conference on Computer Design, 2011.
- [4] 刘胜,陈海燕,葛磊磊,等.面向访问模式的多核末级Cache优化方法[J].国防科技大学学报,2015,37(2):79-85.
LIU Sheng, CHEN Haiyan, GE Leilei, et al. Optimization method for multi-core last level Cache considering the memory access modes [J]. Journal of National University of Defense Technology, 2015, 37(2): 79-85. (in Chinese)
- [5] Alvarez L, Gonzalez M, Martorell X, et al. Hardware-software coherence protocol for the coexistence of Caches and local memories [C]// Proceedings of International Conference on High Performance Computing, Networking, Storage and Analysis, 2012.
- [6] Chen S M, Wan J H, Lu J Z, et al. YHFT-QDSP: high-performance heterogeneous multi-core DSP [J]. Journal of Computer Science and Technology, 2010, 25(2): 214-224.
- [7] TMS320C6678. CorePac. user guide; SPRUGWOC [Z]. TEXAS Instruments, 2013.
- [8] TMS320C6678. Multi-core fixed and floating-point digital signal processor[Z]. TEXAS Instruments, 2014.