

片上异构多核DSP同步与通信的实现

刘建¹, 陈杰¹, 敖天勇^{1,2}, 许汉荆¹

(1. 中国科学院微电子研究所 北京 朝阳区 100029; 2. 河南大学物理与电子学院 河南 开封 475001)

【摘要】设计了一个硬件信号量模块,可实现互斥和栅障等同步功能。与使用处理器原子操作指令相比,该方法具有指令数目少、执行效率高的优点。为提高存储器使用效率,基于便笺式存储器的结构特点,设计了具有绝对地址映射和虚拟地址映射两种寻址模式的共享程序存储器以支持指令存储空间复用。FPGA实验结果证明,该设计与传统的采用L2缓存方式相比,可以将多核处理器系统的程序性能提高14.7%。

关键词 数字信号处理器; 硬件信号量; 共享存储; 同步

中图分类号 TN 914.42

文献标识码 A

doi:10.3969/j.issn.1001-0548.2010.04.011

Implementations of Synchronization and Communication in Heterogeneous Multi-Core DSP

LIU Jian¹, CHEN Jie¹, AO Tian-yong^{1,2}, and XU Han-jing¹

(1. Institute of Microelectronics, Chinese Academy of Sciences Chaoyang Beijing 100029;

2. School of Physics and Electronics, Henan University Kaifeng Henan 475001)

Abstract A hardware semaphore module is designed to support the synchronization primitives, such as mutex and barrier. Compared with the atomic instruction realization, the method executes efficiently and uses fewer instructions. Based on the structure of scratch-pad memory, a shared program memory with two addressing mode of absolute address mapping and virtual address mapping is designed to implement instruction space sharing, resulting in higher utility of memory. The result of FPGA simulation demonstrates that, the performance of the proposed design can achieve speed-up 14.7% compared with traditional shared L2 caches.

Key words digital signal processors; hardware semaphore; shared memory; synchronization

多媒体及通信技术的发展对数字信号处理器提出了越来越高的性能要求,通常采用提高指令发射宽度和时钟频率的方法实现。但是,随着频率的提高以及发射宽度的增加,处理器的功耗和面积也随之增加。通过在处理器中增加内核构成片上多处理器实现并行任务处理^[1-2],可实现更佳效能。片上多处理器分为同构多核和异构多核两类。与同构多核处理器芯片相比,异构多核处理器具有更高的并行计算加速性能^[3],可以针对不同应用运算种类、性能、功耗^[4]以及处理内容的复杂程度等进行优化^[5-6]。

多核处理器对带宽和存储结构等要求更高^[7]。为实现多线程同步,大多数处理器都采用原子操作指令,但是在异构多核处理器中,实现总线、存储控制等模块对各个处理器核心不尽相同的原子操作指令的支持非常困难。此外,在多核处理器的常规设计方法中,cache的使用将带来数字信号处理实时性及数据一致性问题^[8]。

为实现异构处理器同步和共享存储,本文提出了一种基于硬件信号量的多核同步方法,无需处理器原子操作指令的支持。此外,通过对传统便笺式存储器结构(scratch-pad memory, SPM)的改进,实现了支持多核高效共享的SPM存储结构。

1 异构多核DSP系统结构

异构多核DSP是在同芯系列DSP基础上开发的多核DSP结构,如图1所示。该结构以同芯II 16位DSP为处理核心,每个DSP都具有私有的8 K字数据和程序便笺式存储器,以及共享的32 K字数据和32 K字程序便笺式存储器。使用1个32位单发射的RISC CPU作为辅助内核,负责运行嵌入式操作系统,同时管理DSP任务队列和共享存储空间,CPU通过AMBA总线和外部设备互联。该系统利用分层的存储结构和交叉开关互联,实现了高带宽核间通信和高效的数据共享方式。

收稿日期: 2009-01-19; 修回日期: 2009-06-12

基金项目: 国家863计划重点项目(2009AA011700)

作者简介: 刘建(1981-),男,博士生,主要从事异构多处理器方面的研究。

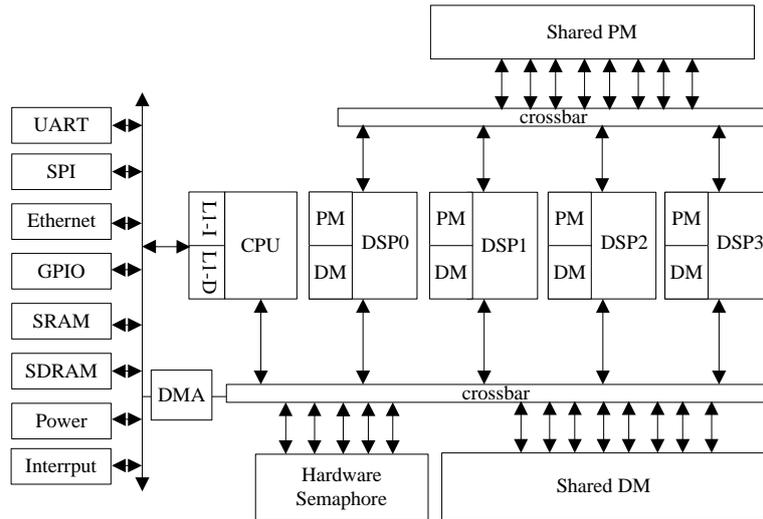


图1 异构DSP系统结构

2 多处理器硬件信号量结构

运行在不同处理器上的进程需要相互协调, 实现通信和同步才能共同完成任务。在共享存储的程序中, 通信通过读/写共享变量的方法实现, 同步则使多个处理器的进程相互等待, 保证对共享变量的访问为互斥访问。处理器之间的同步事件功能可以有效地对共享资源进行保护。

本文通过模拟Dijkstra信号量的申请(P)和释放(V)操作实现了硬件信号量, 代替比较并交换等处理器的原子操作指令。硬件信号量模块和中断相结合, 可以在软件的支持下实现DSP运行多任务时的任务阻塞和进程切换功能。将该硬件信号量模块和系统的功耗控制模块相结合, 可实现处理器阻塞时的低功耗切换。下面将详细介绍硬件信号量单元的释放、申请、初始化及优先级管理, 并介绍申请失败时的等待模式和中断模式。

硬件信号量模块由多个信号量单元组成, 每一个信号量单元对应多处理器系统中的一个或者多个共享资源。硬件信号量的核心是一个“加1减1”计数器。处理器对硬件信号量的操作表达为处理器对一个地址区域的只读访问。信号量模块解析处理器发送的命令(即处理器读操作的低位地址), 实现计数器加减操作, 完成等效于处理器原子操作指令“读-修改-写”的操作, 并把结果返回处理器。硬件信号量命令格式如图2所示, 通过地址的中间4位“semaphore_id”区别16个硬件信号量单元, 通过地址的低4位“command”区别包括申请、释放在内的不同命令。

处理器读操作地址映射

Hardware_semaphore	semaphore_id	command
--------------------	--------------	---------

图2 硬件信号量命令格式

处理器对硬件信号量单元的释放操作实现了计数器的“加1”操作。如果此操作后计数器值小于1, 则表示有其他处理器处于申请资源但是被阻塞状态, 此时解除对其他处理器的阻塞状态。

处理器对硬件信号量单元的申请操作实现了对计数器的“减1”操作。如果“减1”后计数器值非负, 则执行申请操作的处理器获得资源, 若“减1”后计数器值为负, 则表示信号量单元对应的资源处于不可用状态, 执行申请操作的处理器等待其他处理器释放资源。该信号量单元在“减1”操作后计数器值为负时, 根据申请命令可以执行多种处理器的阻塞式等待模式、低功耗等待模式、以及中断模式等后续操作, 如图3所示。

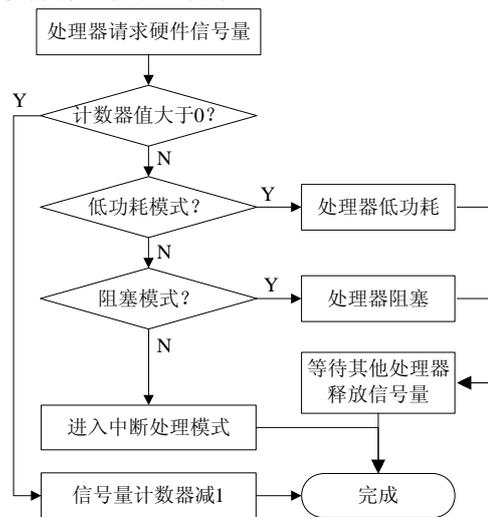


图3 信号量操作状态转移图

两种等待模式通过阻塞处理器对总线的读操作

达到阻塞处理器继续执行后续指令的目的, 实现了申请共享资源的等待, 低功耗模式时, 信号量向功耗控制模块发送相应的请求。在处理器申请信号量单元失败后, 如进入中断模式, 硬件信号量单元记录该处理器编号, 但不阻塞处理器, 当其他处理器执行释放硬件信号量命令后, 硬件信号量单元以中断的方式对其进行通知。

信号量模块除了支持以上申请和释放操作以外, 同时支持初始化信号量计数器、查询信号量单元计数器值等命令。为实现具备优先级的多处理器信号量申请, CPU在初始化时对硬件信号量单元中的多处理器优先级寄存器进行配置, 从而实现多处理器优先级的管理。硬件信号量单元的设计, 使得软件可以调用其申请和释放等操作, 实现复杂的读写锁、栅障等功能, 从而实现多处理器进程间通信。

3 程序和数据共享的便笺存储器

在多核处理器中, 共享缓存会导致数据一致性问题, 而采用监听一致性协议或目录一致性协议维护缓存的数据一致性会导致硬件复杂度和功耗的增加, DSP运行所需的实时性也受到影响。此外, 相对于CPU, 缓存对于DSP的作用要小很多。文献[9]针对便笺式存储器和缓存的研究结果表明, 便笺式存储器比缓存减少约40%的功耗, 并且由于减少了缓存的附带管理单元, 面积减少到66%, 极大降低了DSP的功耗和面积。

本文在异构多核处理器中实现了缓存和便笺式存储器的混合结构, CPU采用缓存, 而DSP则采用便笺式存储器实现数据的存储。同时, 为实现CPU-DSP以及DSP-DSP的通信, 利用共享便笺式数据存储器进行处理器间的数据交互。利用便笺式共享程序存储器实现DSP共享程序空间, 该共享空间可以工作于绝对寻址和虚拟寻址两种模式。

采用绝对地址空间模式访问程序共享存储区时, 各个DSP程序寻址相同的地址, 4个DSP可以复用相同的程序, 减少多份程序在DSP私有程序空间的拷贝, 从而减少总的程序空间需求。

采用虚拟地址访问程序共享存储区时, 在程序基址上加一个偏移量得到实际的共享程序存储区物理地址, 实现DSP需要大于8 K的本地存储空间时, 对DSP本地程序空间进行扩展。

异构多核处理器根据每个DSP所需执行的程序大小对共享程序区进行动态划分, 实现虚拟地址访问。图4展示了虚拟地址空间的动态划分, 其中P0~

P3表示4个DSP可访问的共享程序区间。每个DSP程序指令低8 K字保存在私有便笺式存储器中, 而剩余部分则保存在共享存储器。在DSP加载任务时, CPU配置每个DSP在共享区域的偏移量。

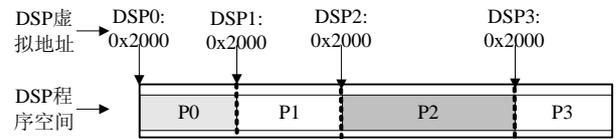


图4 动态分配示意图

为解决多个DSP同时读取共享存储器时的冲突问题, 在每个DSP的读存储器端口设计了一个缓冲区。当发生读操作时, 首先检查缓冲区中是否已有相应的指令, 如果缓冲区中无对应地址的指令, 那么从存储器中读出该地址的指令, 并将该指令及其地址标记写入缓冲区, 供后续读取该指令时使用, 即该缓冲区起到一个微型缓存的作用。

共享数据存储器可以接受16位DSP和32位CPU的并发访问, 实现多处理器的数据存储和交互。为解决存储器读写冲突, 每个DSP写通道具有深度为4的写缓冲区。当DSP发生读操作时, 首先检查自身缓冲区的内容, 如果缓冲区中没有相应数据, 则读共享存储区。为降低硬件复杂性, 每个DSP并不主动检查其他DSP缓冲区中的内容, 由此引发的共享数据一致性问题, 由DSP向其他处理器释放共享数据时刷新缓冲区来解决。

4 实验结果与性能分析

文献[10]的快速共享数据池利用二元硬件信号量和SPM的直接映射, 实现了处理器核间数据传递, 但是一个处理器不能通过多个信号量通知其他处理器同时获得多份共享资源。而本文实现了硬件信号量对任意共享资源的映射, 而非局限于共享存储器的互斥访问。

图5给出了一个利用本文硬件信号量实现互斥访问共享资源的实例: 两个DSP均利用PRINT函数实现输出字符串。为防止两个输出串扰, 必须同一时刻只能有一个处理器输出字符串。因此, 利用信号量单元6表示共享的I/O口资源, 初始状态下其计数器值为1。当DSP0和DSP1同时进行申请操作, 只有DSP0成功, 而DSP1被阻塞。DSP0执行PRINT()后释放信号量, DSP1字符串输出。

图6给出了一个利用本文硬件信号量实现全局同步功能的实例。在该例中, 信号量单元7的计数器初始化为0。

```
#define semaphore(id,pv) (*(char*)(0x302000+(id<<4)+pv))
#define sem_V 0 // up
#define sem_P 1 // down
#define sem_id_6 6 // shared resource: I/O

DSP0:          DSP1:
i=semaphore(sem_id_6,sem_P);i=semaphore(sem_id_6,sem_P);
PRINT("Hello,DSP0");    PRINT("Hello,DSP1");
i=semaphore(sem_id_6,sem_V);i=semaphore(sem_id_6,sem_V);
```

图5 硬件信号量例子—互斥

```
#define semaphore(id,pv) (*(char*)(0x302000+(id<<4)+pv))
#define sem_V 0
#define sem_P 1
#define sem_CNT_read 3 // get semaphore counter值value
#define sem_id_7 7 // shared resource: Barrier

DSP0-2:        DSP3:
i=semaphore(sem_id_7,sem_P);while(true) {
    i=semaphore(sem_id_7,sem_CNT_read);
    if(i== -3) break;
}
i=semaphore(sem_id_7,sem_V);
i=semaphore(sem_id_7,sem_V);
i=semaphore(sem_id_7,sem_V);
```

图6 硬件信号量例子—栅障同步

以上2个例程利用信号量申请和释放只需执行一条指令，而基于原子操作指令的互斥和栅障同步操作则需要多达几十条指令来实现。此外，硬件信号量也避免了由于申请失败导致处理器重复读取指令占用总线的问题。此外，硬件信号量使处理器处于阻塞状态，减少了功耗。

为测试片上异构多核DSP共享指令存储器执行相同任务的性能，本文运行了Jacobi迭代^[11]、PSRS排序^[11]与JPEG解码3个程序，并对比程序存储于DSP本地存储空间(LM)、共享存储空间(SPM optimization)以及无优化的共享存储空间(SPM)的结果，得到相对于LM归一化的DSP每周指令数(IPC)，如图7所示。由图7可见，本文对类似Jacobi迭代的短循环密集计算实现了非常好的优化，这是由于循环体可以保存在小容量缓存中，避免了4个处理器读存储器的冲突。对于PSRS排序，核心运算也为短循环指令，并且4个DSP计算量不平衡使得同步操作占了较大比例。本文采用了阻塞式同步设计，在DSP执行同步操作时，不会和其他DSP抢占总线，因此在共享存储空间时也有很好的性能。而对于JPEG等复杂程序，共享存储器结构相对于私有存储器结构性能下降仅3%，且不需要4份私有程序。

本文基于FPGA测试了4个DSP在CPU控制下进行JPEG码流的宏块级并行解码，实现了相对于单个DSP解码1:4.1的加速比，该结果相比文献[12]提高了14.7%的性能。

最后，在SMIC 0.18um CMOS工艺下，本文使用Synopsys公司的DC综合工具对该设计进行综合

的结果如表1所示。可见，引入硬件信号量模块所需要付出的面积代价极小。

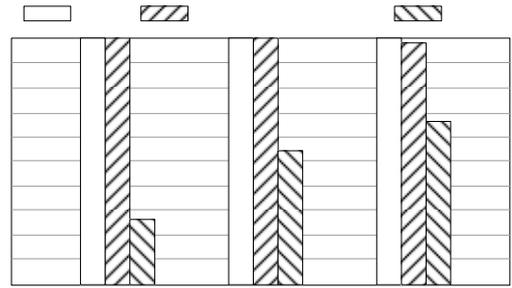


图7 共享程序存储器性能对比

表1 DC综合结果

性能指标	设计结果
工作频率/MHz	150
芯片面积/门	约300万
信号量单元面积/门	1 603

5 结束语

本文提出了一种新颖的异构多核DSP系统结构，利用硬件信号量模块取代处理器原子操作指令实现简单、高效的同步功能，可以使异构多处理器系统高效地实现核间的同步与通信，并且只需要付出极小的面积代价。此外，利用便笺式存储器实现了多处理器程序和数据共享。FPGA验证结果表明，该结构具有很高的编程灵活性，易于并行任务开发，实现高的性能加速比。

参 考 文 献

- [1] BROOKS D, BOSE P, SCHUSTER S, et al. Power-aware micro architecture: design and modeling challenges for next generation microprocessors[J]. IEEE Micro, 2000, 20(6): 24-44.
- [2] AGARWAL A, LEVY M. The kill rule for multicore [C]//Design Automation Conference. San Diego, CA: ACM/IEEE, 2007: 750-753.
- [3] HILL D, MARTY R. Amdahl's law in the multicore era[J]. IEEE Computer, 2008, 41(7): 33-38.
- [4] 桑楠, 李保宇, 马红. 多处理器的节能调度算法[J]. 电子科技大学学报, 2008, 37(1): 116-119. SANG Nan, LI Bao-yu, MA Hong. A scheduling algorithm of saving energy for multiprocessors[J]. Journal of University of Electronic Science and Technology of China, 2008, 37(1): 116-119.
- [5] TI. OMAP-L137 low-power applications processor [EB/OL]. [2009-01-12]. <http://focus.ti.com.cn/cn/docs/prod/folders/print/omap-l137.html>, 2009-1-14.

(下转第536页)