异质存储系统中的高速缓存机制研究

王 超[1,2],张惠臻[1,2],周学海[1,2],马宏星[1,2]
(1.中国科学技术大学计算机学院,安徽合肥 230027; 2.中国科学技术大学苏州研究院,江苏苏州 215123)

摘 要: 存储子系统是嵌入式系统的重要组成部分.由于传统存储系统的设计已经无法满足容量日益增长的需求,固态存储器的应用越来越广泛.针对固态存储系统的存取速率慢的问题,目前常用的优化技术主要有缓存和并行读写技术.然而,在大多数应用和研究中,如何将这两类技术进行融合是目前遇到的一个重大挑战.对此,本文提出一种融合了缓存技术和并行读写技术的基于循环缓冲的新型缓存机制.本方案将缓冲和固态存储模块通过交叉矩阵进行互联,并用专门的缓冲来进行读写过程中的错误处理.理论计算和仿真实验表明该机制能够有效地提升大容量固态存储系统的访问速率.原型系统证明本方案具有直接在电路板布局布线实现的高可行性.

关键词: 异构存储系统; 交叉矩阵; 缓冲调度
中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112 (2011) 06-1267-05

A Study on Cache Mechanism in Heterogeneous Memory System

WANG Chao[1,2], ZHANG Hui-zhen[1,2], ZHOU Xue-hai[1,2], MA Hong-xing[1,2]
(1. *School of Computer Science, University of Science and Technology of China, Hefei, Anhui* 230027, *China*;
2. *Suzhou Institute for Advanced Study, University of Science and Technology of China, Suzhou, Jiangsu* 215123, *China*)

Abstract: Memory subsystem plays a vital role in embedded systems. Since it's hard to obtain increasingly capacity for traditional memory systems, NAND Flash based SSM (solid state memory), especially heterogeneous SSM with RAM (read only memory), is becoming more and more widely used. Corresponding to low speed problem in SSM, cache and parallel technologies are commonly used for optimization. However, it poses a significant challenge to integrate them into a single solution in most applications. This article presents a mechanism based on cyclic buffer and integrates caches with parallel technologies. Cache and Storage modules are connected through a switch-matrix. A unified buffer is introduced to handle errors. Benefiting from dual bus infrastructure, also flexible addressing and scheduling strategies, this approach can largely improve data level parallelism. Verification and empirical experiment results demonstrate the concept with high speed and satisfactory performance. Prototyping system shows the feasibility of the placement layout and routing within circuit board.

Key words: heterogeneous memory system; switch matrix; cache scheduling

1 引言

随着嵌入式系统在日常生活中应用的日益广泛,网络、数字化技术的普及和高速数据采集技术的发展使人们在通讯、工作和娱乐等方面发生了革命性的变化,大规模数据应用如数字图书馆、地理信息系统、媒体库和遥感数据中心等得到了越来越广泛的应用.目前,海量数据的收集大多采用嵌入式设备,如视频采集器等.而传统的硬盘、磁带等存储介质由于容量小、存取速度慢、功耗大、易损坏、可靠性低及适应性差等缺点,无法满足大规模数据应用的需求.因此,固态存储器(Solid State Memory,SSM)在数据存储中尤其是嵌入式存储系统中起到了越来越大的作用.

SSM具有可靠性好、存储密度高、高速随机访问、功耗小、成本低等优点,倍受学术界和工业界的关注.目前的SSM[1,2]在规模和存取速度上,相对于大规模数据应用的需求而言,仍然存在差距,因此其应用范围受到了一定限制.

2 研究现状与分析

目前最常用于SSM的半导体存储介质是DRAM(Dynamic Random Access Memory)和NAND Flash.其中,DRAM的存储密度相对较低,且具有易失性,需有定期刷新机制以维持数据信息,因此以DRAM为介质的固态存储器的容量极为有限,在新型存储设备中应用较少[3,4].而NAND Flash是基于与非门的闪存芯片,存储

收稿日期:2010-03-21;修回日期:2010-06-18
基金项目:国家高技术研究发展计划(863计划)课题(No.2008AA01Z101)

密度较高,同时其数据在断电后依然能够保持,应用非常广泛.但它的控制逻辑比较复杂,直接访问速度较低,如何根据 NAND Flash 的特征实现存储阵列的高速访问是目前学术界和工业界亟待解决的问题.

从容量和读写速度的角度看,现有的 SSM 解决方案主要包括例如 SAMSUNG[5]、Toshiba[6]、SANDISK[7]等厂商推出的基于 NAND Flash 的 256GB 的大容量固态存储器,其读写速度最高可达 200~220 MB/s,但其对于大规模的存储密集型应用来说还是不够的.目前针对 SSM 的研究主要集中在通过并行技术即通过增加总线宽度的方法来提高存取速度.例如文献[8]中提出固态存储器的性能瓶颈一般在内部的核心接口上,可以采取系统级或设备级的并行访问来解决.文献[9]中通过一个多通道的存储结构实现了访问的流水化,但由于受到总线宽度的限制,无法满足更大规模数据的同时写入、直接读取的要求.

缓存技术是提高 SSM 吞吐率的另外一个研究热点[10].网络处理器及其应用的研究[11]以及多媒体存储系统[12],经常采用高速缓存 Cache 作为高速访问的媒介.同时在海量数据存取中缓存技术也是经常采用的优化策略.目前的这些缓存技术的主要缺点是需要对缓存进行信息读取来构建索引,从而会引入额外的读事务而增大系统的开销.另外一些系统[13]虽然利用 FPGA(Field Programming Gate Array)或者 Flash 自带的部分缓存来加速读写,但对整个系统来说缓存资源太少,无法进行整体调度,会导致缓冲的频繁失效而增大系统的响应时间.

总的来说,目前的固态存储器优化技术可扩展性较差,无法适用于可变应用的大容量存储系统中.因此我们提出了基于交叉循环缓冲的缓存机制,该机制在缓存的基础上支持并行读写,有很强的扩展性.

3 基于交叉循环缓冲的缓存机制

3.1 缓存机制组织架构

图 1 显示了基于交叉矩阵的缓存机制的组织架构,包含了 Flash 阵列、多组静态 SRAM 缓冲、交叉开关矩阵、以及用来进行交叉矩阵连接的总线和对应的控制逻辑.其中 Flash 阵列是存储的主要介质;循环缓冲是外部接口与存储阵列的通信的中间媒介,缓冲还负责进行 Flash 编程的错误转移;交叉矩阵在不同的读写阶段可以通过配置可以实现不同的功能.

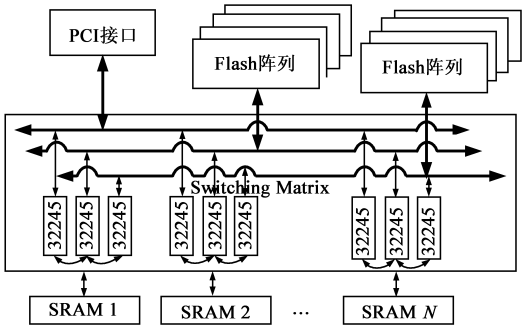


图1 基于交叉矩阵的缓存架构

3.2 读写流程

缓存机制的主要工作流程包括数据写入和数据读取,分别如图 2 中(*a*)和(*b*)所示.

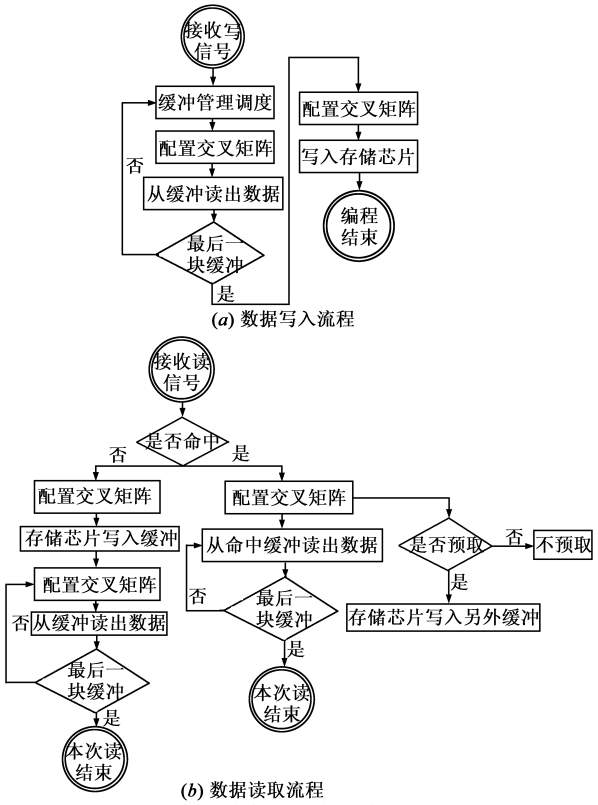


图2 缓存机制的主要读写流程

在数据写入时,如(*a*)所示,系统接收写请求,然后通过缓冲调度模块获得当前要写入的缓冲组编号,之后配置交叉矩阵使缓冲与访问源相连.在将数据写满第一组缓冲之后,重新配置矩阵使外部访问源将数据写入.同时,已写满的缓冲通过交叉矩阵与 Flash 阵列相连,进行从缓冲到 Flash 阵列的数据传输.

在数据读取时,如(*b*)所示,系统首先接收读请求,然后判断在当前缓冲中是否存在数据命中.若命中,则直接配置交叉矩阵使缓冲与外部访问源相连进行数据读取,同时查看预取标志位,若需要预取,则将 Flash 芯片中的下一组数据预取到下一组缓冲中;若未命中,则需要先将数据从 Flash 芯片中读取到缓冲中,然后再从缓冲中读取.

3.3 编址机制和调度策略

为提高 Flash 的访问速率,将存储阵列在逻辑上分为奇偶两个体以提高并行度.整个 Flash 阵列对外提供

两套总线,分别拥有各自独立的控制信号,在个体内以分列的方式对 Flash 阵列进行控制.其中每列 8 块芯片,不同列的 4 块芯片进行位扩展为一套总线.

Flash 的存储以页为单位,因此按整页编程的效率比按字节编程的效率高.设计中使用多组(每组两片)和 Flash 页大小相同的 SRAM 芯片作为缓冲.在写入时,外部接口直接将数据写入缓冲.当该组缓冲写满后,外部接口换到下一组缓冲进行数据写入,同时将刚刚写满的数据写入 Flash.在读过程中,对于未命中的情况首先将数据从存储阵列读到缓冲芯片,然后从缓冲芯片中读出.

考虑到缓冲的组数太多会导致系统的硬件开销会比较大,若太少则无法进行操作的流水化,因此本方案在读写过程中,使用 3 组缓冲芯片作为典型值.调度中使用写标志寄存器 WReg 和读标志寄存器 RReg 来判断系统是否空闲.另外有 3 个地址寄存器存放 3 组缓冲芯片中各自数据的起始地址,用来判断读操作是否命中.缓冲调度过程如图 3 所示.

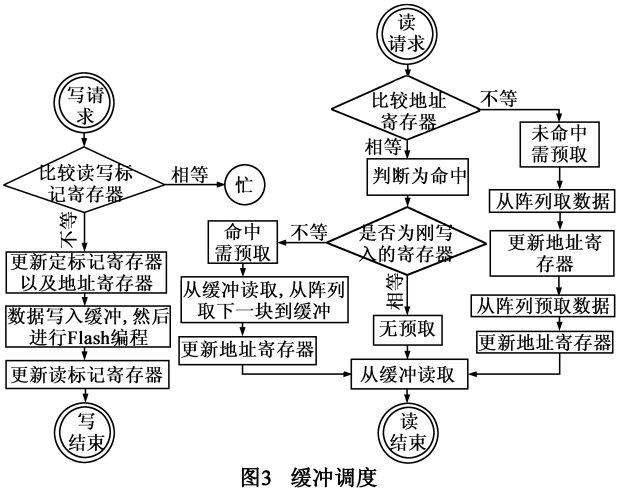


图3　缓冲调度

在系统接收到写请求之后,首先比较读写标志寄存器是否相等,若相等,则表明当前全部缓冲中的数据尚未写入 Flash 芯片中,系统处于繁忙状态;若不等,则更新写标记寄存器,标记有新的数据进入缓冲,在将数据写入缓冲之后,更新读寄存器,标记有新的数据需要进行从缓冲到 Flash 芯片的转移.

同理,在接收到读请求之后,通过比较地址寄存器来判断读请求是否命中:若命中,则可以直接从缓冲中读取数据,同时判断是否要进行数据的预取以提高效率(若命中的缓冲为刚刚写入的数据,则不进行预取,理由是刚写入的数据被读出,则很有可能是对数据写入正确性的验证),若需要预取,则将 Flash 中下一块地址的数据从阵列取到下一组缓冲中;若请求未命中,则需要一定的响应时间使数据先从 Flash 阵列中转移到缓冲中,然后从缓冲中传送给外部访问源的同时预取下一块数据.

由于 Flash 存储介质存在坏块,因此向 Flash 进行编程有可能失败,为了在失败时不中断写入,系统在错误反馈的同时将该页的数据从工作缓冲转移到错误缓冲进行存储.为了保证错误记录和处理顺序的一致性,错误管理采用先来先服务(First Come First Serve)的原则,从缓冲向 Flash 写入的时候出现错误时,此时读写访问仍在继续,则需要将发生错误的数据保存,同时将错误编号及出错地址作为反馈给文件系统,等 Flash 写入结束后再将数据由工作缓冲转移到错误缓冲.当接收到文件系统的处理错误命令时,再将错误的数据从错误缓冲写入到 Flash 芯片中.

4　实验与分析

4.1　效率分析

在不采取任何加速措施的情况下,Flash 典型的读时序由发送命令和传送数据两阶段组成,约为 78μs,在页大小为 2048byte 的情况下,典型的读带宽为 216Mbps.Flash 的写时序主要可以分为 3 个阶段:命令阶段(Command),传送数据阶段(Data)和内部编程阶段(Program).典型的写周期最小为 25ns,编程时间在 220μs ~ 500μs 左右,因此其平均写入时间为 275μs,最大带宽为 61.4Mbps.

系统中数据总线扩展为 32 位,则读带宽为 216 × 4 = 864Mbps.而在编程时,考虑到 Flash 进行阵列编程期间完全由其控制器自身控制,不需要外部逻辑的参与,所以可采取多路流水线写入的方式来提高访存带宽.编程时间大概是发送命令和数据写入两阶段时间之和的 4 倍(220μs:60μs),可以采取 5 级多路并行流水写的方式以掩盖 Flash 芯片的编程时间.在这种情况下,经过扩展之后的写带宽为 307.5Mbps.另外 Flash 单片数据总线宽度为 8 位,通过位扩展可将数据总线扩展成 32 位(每次写入 4 片),则带宽可达 307.5Mbps × 4 = 1230Mbps.

4.2　实验结果

仿真原型系统采用的 Flash 模型为 64 片 Micron Technology,Inc.的 MT29F 芯片(8Gbit),页大小为 2048 byte,整个系统的容量可达 8Gbit × 64 = 512Gbit;缓冲芯片的 VHDL 模型采用 CY7C1010DV33(2-Mbit Static RAM (256Kbit × 8)),6 片为正常的工作缓冲,1 片作为专用的错误缓冲.实验利用 ModelSim 工具测试了复位、擦除、读写数据和错误处理等操作流程.针对不同类型的应用,实验共测试了全读(100%读)、多读少写(70%读 + 30%写)、读写均衡(50%读 + 50%写)、多写少读(70%写 + 30%读)和全写(100%写)几种不同的访问情况.

4.2.1 访问速率

仿真实验的结果如图4所示,横坐标表示了五组不同类型的应用(全部读、多读少写、读写平衡、多写少读、全部写),纵坐标表示了不同应用的访问速率.三个柱体分别为未经加速的速率、加速之后的理论计算值以及具体实验数据.从该数据我们可以分析得到,由于优化策略主要针对效率相对较低的写入操作,故当目标应用中写操作所占比例较高时,系统获得的加速比也相对较高.

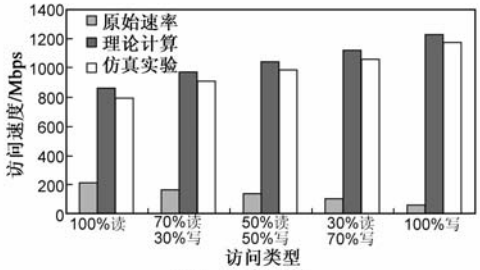


图4 速率比较

4.2.2 响应时间

当进行一次写操作之后再进行读操作,且操作的地址和正在编程的 Flash 地址处在一块芯片中,则会产生写后读冲突.系统采取使读等待之后中断写的策略,由于写操作以逻辑块(假设为 256KB)为单位,连续的读操作软件查询被分成多次进行,每次以一个逻辑块为单位,所以每次需要等待的时间最坏情况为:$t_{\mathrm{MaxWait}} = \mathrm{Size}_{\mathrm{LogicBlock}}/\mathrm{BW}_{\mathrm{Write(pipeline+32bitextension)}} = 256\mathrm{Kb} \times 8 \div 1175\mathrm{Mbps} \approx 1.70\mathrm{ms}$,因此由写后读冲突而引起的平均等待时间为 $t_{\mathrm{AverWait\text{-}RAW}} = t_{\mathrm{MaxWait}}/2 = 0.85\mathrm{ms}$.

当进行一次 Flash 读操作时,如果命中则响应时间很短可以忽略;如果未命中则需要先将数据从存储阵列转移到缓冲中.读操作以逻辑块为单位,则等待的时间为 $t_{\mathrm{MaxWait}} = \mathrm{Size}_{\mathrm{LogicBlock}}/\mathrm{BW}_{\mathrm{Write(pipeline+32bitextension)}} = 256\mathrm{Kb} \times 8 \div 796\mathrm{Mbps} \approx 2.51\mathrm{ms}$,所以由读缺失而引起的平均等待时间为 $t_{\mathrm{AverWait\text{-}RAW}} = t_{\mathrm{MaxWait}}/2 = 1.26\mathrm{ms}$.

4.2.3 原型系统

本项目实现了原型电路板,如图5所示.其中采用 Xilinx Virtex II Pro 作为主控 FPGA,交叉矩阵芯片、缓冲芯片和存储阵列芯片分别采用 FST3245 8-Bit Bus Switch、CY7C1010DV33 和 Samsung 的 K9K8G08UOA-PCBO.

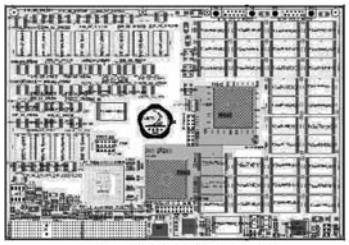


图5 原型系统电路板

5 总结

本文提出了一种基于 NAND Flash 闪存芯片和 SRAM 缓冲芯片实现大容量数据存储的方案,设计了一种新的缓存机制对数据的读取进行缓存.整个方案分成缓冲、存储阵列、错误管理等不同模块,通过可以动态配置的交叉矩阵进行互连.多组缓存通过基于循环缓冲的缓冲管理和调度机制来提高读写速率、命中率及减少响应时间.存储阵列存取时采用了灵活的编址技术和调度策略、并行双总线架构及扩展总线位宽的方式以提高存取速度.仿真实验和原型系统实现表明该机制可以有效地提升大容量固态存储区的访问速率,支持错误管理和并行存取,并且系统的响应时间也较小.

致谢 感谢陈香兰博士和纪金松博士给本文提出的参考意见.

参考文献

[1] Adams L, K Chao, M Fehringer, C Miller, et al. Challenges in implementing commercial non-volatile memory in spacecraft solid state recorders[A]. 1st Annual Non-Volatile Memory Technology Symposium[C]. Arlington, Virginia, USA: IEEE Computer Society Press, 2000. 23 – 25.

[2] European Space Agency. A3SysDef: Aurora Avionics Architecture System Definition[EB/OL]. http://www.esa.int/esaMI/Aurora/SE-MZ87A5QCE_0.html, 2006 – 05 – 10/2010 – 03 – 21.

[3] 王芳,李恪,苏林,耿立红.空间太阳望远镜的星载固态存储器研制[J].电子学报,2004,32(3):472 – 475.
WANG Fang, LI Ke, SU Lin, GENG Li-hong. Development of onboard solid state recorder for space solar telescope[J]. Acta Electronica Sinica, 2004, 32(3): 472 – 475. (in Chinese)

[4] 张科,郝智泉,王贞松.一种基于新体系结构的空间固态记录器原型系统[J].电子学报,2008,36(2):285 – 290.
ZHANG Ke, HAO Zhi-quan, WANG Zhen-song. A novel architecture prototype of space slid-state recorder[J]. Acta Electronica Sinica, 2008, 36(2): 285 – 290. (in Chinese)

[5] Samsung. Flash SSD for Enterprise[EB/OL]. http://www.samsung.com/global/business/semiconductor/products/SSD/Products_Enterp-rise_SSD.html, 2010 – 01 – 08/2010 – 03 – 21.

[6] Toshiba Inc High Performance SSDs (HG3 series)[EB/OL]. http://ssd.toshiba.com/SSD-product-guide.html, 2010 – 01/2010 – 03 – 10.

[7] SanDisk. SanDisk SSD G4[EB/OL]. http://www.sandisk.com/business-solutions/ssd/g4-solid-state-drive, 2010 – 02/2010/03 – 21.

[8] Cagdas D, J Bruce. The performance of PC solid-state disks

(SSDs) as a function of bandwidth, concurrency, device architecture, and system organization[A]. Steve Keckler. International Symposium on Computer Architecture Proceedings of the 36th Annual International Symposium on Computer Architecture[C]. Austin, USA: ACM Press, 2009. 279 – 289.

[9] Kang J-U, J-S Kim, C Park, H Park, et al. A multi-channel architecture for high-performance NAND flash-based storage system[J]. Journal of Systems Architecture, 2007, 53(9): 644 – 658.

[10] Ramakrishna K, J S Love, G W Bradley. Caching strategies to improve disk system performance[J]. Computer, 1994, 27(3): 38 – 46.

[11] 刘祯,刘斌,郑凯,陈善真.网络处理器中的高速缓冲机制及其有效性分析[J].清华大学学报(自然科学版), 2008, 48(1): 113 – 116.
LIU Zhen, LIU Bin, ZHENG Kai, CHEN Shanzhen. Trace driven study of the effectiveness of caching mechanisms for network processors[J]. Journal of Tsinghua University (Science & Technology), 2008, 48(1): 113 – 116. (in Chinese)

[12] 朱晴波,乔浩,陈道蓄.分布式多媒体存储系统中的全局缓存管理[J].电子学报, 2002, 30(12): 1832 – 1835.
ZHU Qing-bo, QIAO Hao, CHEN Dao-xu. Global memory management in distributed multimedia storage systems[J]. Acta Electronica Sinica, 2002, 30(12): 1832 – 1835. (in Chinese)

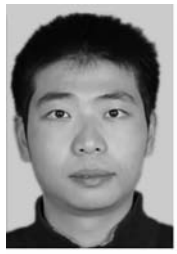
[13] 张燕,焦新泉,熊继军.超大容量高速存储技术研究[J].微计算机信息(嵌入式与 SOC), 2008, 24(2 – 2): 171 – 172.

作者简介

王　超　男,1985 年 8 月出生于河北保定,中国科学技术大学博士研究生,主要研究方向为:可重构存储系统设计、可重构计算.
E-mail: saintwc@mail.ustc.edu.cn

周学海　男,1966 年生于安徽淮南,中国科学技术大学教授,博士生导师,主要研究方向为嵌入式系统设计、计算机体系结构和可重构计算.
E-mail: xhzhou@ustc.edu.cn

张惠臻　男,1983 年生于福建龙岩,中国科学技术大学博士研究生,主要研究方向为可重构计算、编译优化技术.
E-mail: zhanghz@mail.ustc.edu.cn



马宏星　男,1983 年生于安徽明光,博士研究生,主要研究方向为嵌入式操作系统,可重构计算.
E-mail: mhx@mail.ustc.edu.cn