

一种面向写穿透 Cache 的写合并设计及验证

梅魁志, 李国辉, 张斌

(西安交通大学人工智能与机器人研究所, 710049, 西安)

摘要: 为了利用片上缓冲技术来提高处理器应用性能,提出一种面向写穿透 Cache 的写合并设计方法.使用同步动态随机存储器(SDRAM)的单个写方式和片上写缓冲器,对 SDRAM 一行内的局部数据采用写合并策略,由此提高了外部存储的访问效率,同时给出了连续和单个 Cache 读写的缓存与内存的数据一致性策略.在寄存器传输语言(RTL)仿真环境下使用 mp3 解码对 Leon2 处理器进行数据测试,结果表明:在缓冲区优化为 3 行 8 列的参数下,SDRAM 每次行开启平均进行 7.8 个字的写入操作,外存的读写效率由 12% 提高到 19%;在 TSMC 0.18 μm 工艺下,综合后面积为 0.263 mm^2 ,流片后工作主频为 100 MHz.

关键词: 写穿透;写合并;处理器;同步读写存储器;读写效率

中图分类号: TP302 **文献标志码:** A **文章编号:** 0253-987X(2010)04-0001-04

A Write Coalescing Design and Verification for Write-Through Cache

MEI Kuizhi, LI Guohui, ZHANG Bin

(Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A write coalescing design for write-through cache is proposed to promote the performance of application processors using on-chip buffer. The scheme of write coalition for the local data in the same row of SDRAM (synchronous dynamic random access memory) is designed to improve accessing efficiency by employing the single write mode of SDRAM and on-chip buffer. A coherence scheme for the single or multiple cache data reading or writing is also presented. Simulation for mp3 decoding data is implemented in the RTL (register transmit language) simulation environment for Leon2 processor. The simulation results show that when the on-chip buffers are optimized at 3 rows and 8 columns, average 7.8 words are written into the SDRAM after every row pre-charge, and the accessing efficiency increases from 12% to 19% at 100 MHz, and that the area of the proposed design is 0.263 mm^2 under TSMC 0.18 μm process.

Keywords: write through; write coalescing; processor; synchronous dynamic random access memory; access efficiency

写缓冲区的实现方式是当前较多处理器采用的一种提高 Cache 读写效率的方式,如 Intel Xscale 处理器对读和写均使用了缓冲存储方式^[1]. Leon 处理器是欧洲航空航天局利用 VHDL 开发的基于 SPARC V8 指令集的开源处理器核,在嵌入式、SoC (System on Chip)以及航天领域得到了广泛的应用,可参见 NASA 的 2009 MAPLD (Military and Aerospace Programmable Logic Devices) 和 EuroSpace 的 2009 DASIA (Data Systems in Aerospace) 等^[2]. 开源处理器 Leon2 设计为写穿透 Cache 方式,且存储控制逻辑对同步读写存储器 (SDRAM) 的操作为单次的读写方式,具有较大的读写延迟.

为了提高 Leon 处理器在 SoC 集成应用中的存储访问效率,本文给出了一种面向写穿透 Cache 的

行和 $N(N>0)$ 列数据,每个缓冲区的存储单元由 32 b 数据和列地址(位数等同于 SDRAM 的列地址位宽)构成。行地址信息由 $R_x(x=1,\cdots,M)$ (SDRAM 中所属的第 k 行位置的行地址)及标志信号构成。标志信号标识了缓冲区中行存储单元的信息,由 4 个信号构成:FULL(1 b)表示行是否为满;EMPTY(1 b)表示行是否为空;Bank 号(2 b)对应 SDRAM 的 Bank 地址。

2.2 数据缓冲区满时的替换算法

图 3 所示的数据缓冲区的行均对应 SDRAM 的行,但列宽小于 SDRAM 的列宽,因此缓冲区中的同一行数据在写入时只需要发出一次 SDRAM 行开启命令,同一行数据的公共数据项的行地址确定了缓冲区的行数据与 SDRAM 行的对应关系。缓冲区的数据读写查询可以通过 AHB 发出的地址总线经行地址和列地址来解析,并通过与 M 行的公共行地址及行内列地址的比较来进行。

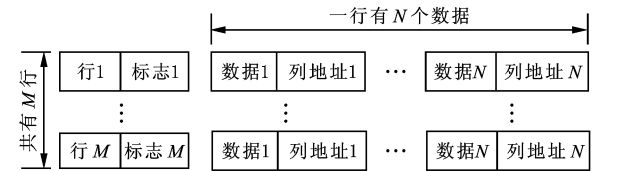


图 3 缓冲区数据存储结构图

缓冲区替换算法可采用最少使用(LRU)算法来淘汰最近最久未使用的缓冲区行。该算法需要缓冲区的每一行增加 1 个访问字段,以记录该行自上次被访问以来所经历的时间 t 。当需要替换行数据时,淘汰 t 值最大的行。LRU 算法如图 4 所示,其首先将每一行的编号存入栈中,如果某一行(如图 4 中的第 0 行或第 1 行)已写入数据,则将该行置入栈底,其他行依次向栈顶移动。在每一次的行替换中,数据应取自于栈顶的行。

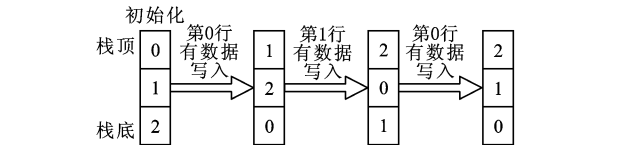


图 4 LRU 算法替换示意图

3 数据一致性下的缓冲数据读写控制

当数据存入缓冲区之后,由于存在同一数据的多份拷贝,因此需要确保数据的一致性。内存中的内容在 Cache、写合并缓冲器(Buffer)中存有副本,数据一致性需要保证从内存中读取的数据始终为最新

的数据,包括 Cache 中的与内存中的一致性,写合并 Buffer 中的与内存中的一致性等。图 2 中写合并模块可作为总线从设备(Slave)接收总线传过来的各种读写请求,因此需对设备写合并模块设置 SDRAM 访问规则,以保证数据的一致性。数据一致性下的写合并对 SDRAM 的缓冲数据读写控制的机制描述如下。

(1)单个数据读操作。先查看数据,如果数据已在缓冲区中,则读取缓冲区中的数据,此时无需访问 SDRAM,由缓冲读控制电路实现。否则,从 SDRAM 中读取数据,并由控制主状态机电路实现。

(2)单个数据写操作。如果数据已在缓冲区中,则更新数据,此时无需访问 SDRAM,由缓冲写控制电路实现。否则,查看条件①,即是否存有一行的行地址与本次写操作的相同且该行不满,如果存在,执行写操作,如果不存在,查看条件②,即缓冲区中是否有空行,如果有,将数据写入该空行,同时更新这一行的行地址信息。如果不满足条件①和②,则等待空行再执行条件②。

(3)连续数据读操作。开启 SDRAM,以便读取其中的数据,在读到数据后,需要查看缓冲区中是否存在该数据的拷贝:如果存在,该拷贝必定为最新的拷贝,应将缓冲区中的数据作为结果返回;如果不存在,应将 SDRAM 中的数据作为结果返回。也就是说,读缓冲区与 SDRAM 的读操作是同时进行的。

(4)连续数据写操作。开启 SDRAM,以便写入数据,在写数据的过程中同时查看缓冲区中是否存在该数据的拷贝:如果存在,则将数据同时写入缓冲区中,即写缓冲区和 SDRAM 的写操作是同时进行的;否则,将数据只写入 SDRAM。

(5)检查缓冲区中的行信息状态。如果出现行满,则根据替换算法将某一行的数据全部写入 SDRAM,使缓冲区存有空行,以供写数据时使用,保证缓冲区中的数据为最新数据。

4 设计验证与结论

增加写合并功能来改变处理器(Leon2)Cache 的访存行为,可用单个数据的传输效率来体现。对于 DCache,需考虑读写策略和命中情况对 AHB 总线的单字读写操作的影响,描述如下。

(1)单字读操作且命中缓存。要读的数据已经存放于写合并 Buffer 之中,当读地址的行地址与 Buffer 中某一行的匹配,且与该行中某一数据的列地址匹配,读取命中,而 Cache 经一个周期后得到

数据.

(2)单字读操作且不命中缓存. 这需要申请存储总线占用,得到响应后,SDRAM 控制器进行开启行操作,在经过 t_{RCD} 、 t_{CL} 延迟之后,数据读出,此时的传输效率最低(在本文设计应用的芯片中,该操作周期为 11 s^{-1}).

(3)单字写操作且命中缓存. 此时,数据可以直接写入缓存中,不需要申请存储总线占用. 写命中有 3 种情况:①缓存中有空行;②缓存中无空行,但有一行的行地址与写操作的相同,且该行不满;③行已满,但有一个数据的列地址与本次写操作的列地址相同.

(4)单字写操作且不命中缓存. 这需要进行行替换操作,并向存储总线提出申请,以便把替换出来的行写入内存之中. 由于行替换可以在存储器总线空闲时进行,因此这种情况发生的概率比较小.

通过仿真实验对本文提出的写合并策略下的 SDRAM 读写效率进行了验证,具体统计所用的测试数据流来自于 mp3 解码程序,并经仿真软件 Modelsim 对整个芯片 SoC(主处理器为基于 Leon2 的寄存器传输语言(RTL)级代码)系统仿真得到. 依据统计数据流中 SDRAM 写数据的请求次数与实际发生的 SDRAM 写次数,来计算每一次写 SDRAM 时的平均写入的数据数 λ . 表 1 为数据缓冲区间为 M 行、 N 列在不同组合情况下的 λ . 综合考虑了芯片实现时的硬件资源与工作频率(数据缓冲区的存储单元均为寄存器的实现方式),当 $M=3$ 、 $N=8$ 时, $\lambda=7.8$ (较优化的组合参数),即一次 SDRAM 写操作的行开启,可平均写入 7.8 个字.

表 1 不同行列值下的一次 SDRAM 写操作时平均写入的数据数

N	λ		
	M=2	M=3	M=4
8	4.23	7.80	7.90
16	5.70	15.20	15.80

使用 Synopsys Design Compiler 对写合并模块 ($M=3$, $N=8$) 的 RTL 级电路进行了综合(TSMC $0.18\text{ }\mu\text{m}$ 工艺库),综合面积:组合电路的面积为 $148\,231.859\,375\text{ }\mu\text{m}^2$,非组合电路的面积为 $114\,640.570\,312\text{ }\mu\text{m}^2$,综合后总的面积为 $262\,872\text{ }\mu\text{m}^2$. 在 TSMC $0.18\text{ }\mu\text{m}$ 工艺下经流片后,该电路的工作频率稳定在 100 MHz . SDRAM 读写效率(也称为 SDRAM 利用率)

$$\eta = \lambda_{\text{send}}/\omega$$

式中: λ_{send} 为传递的数据数; ω 为 SDRAM 处于非空闲周期状态的周期数

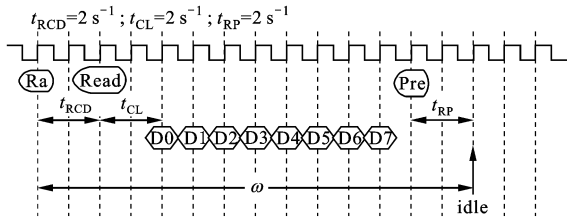


图 5 SDRAM 读写效率计算示意图

非空闲状态的时钟周期数除了包括传送数据的时钟周期数之外,还包括 t_{RCD} 、 t_{CL} 、 t_{RP} 等延迟. 如图 5 所示,传输 8 个数据,包括 t_{RCD} 、 t_{CL} 、 t_{RP} 在内,总共用了 14 个周期,传输的效率为 57%. 统计 Cache 模块对 SDRAM 的访问可知,在采用写合并机制之后,SDRAM 的读写效率由 12% 提高到 19%. 上述 SDRAM 效率仍然低的原因在于,单字读取且不命中引起了低效率的 SDRAM 传输且读延迟过大,这是由于 Leon2 DCache 只更新读取缺失的数据. 如果将 DCache 在发生读缺失时从内存中读取一行更新 Cache,则 SDRAM 的访问效率可由 19% 提升到 46%.

参考文献:

[1] Intel Inc. Intel XScale technology overview [EB/OL]. [2009-07011]. <http://www.intel.com/design/intelxscale/>.

[2] Gaisler. Leon SPARC V8 processor [EB/OL]. [2009-07-11]. <http://www.gaisler.com>.

[3] 林伟,叶笑春,宋风龙,等. 众核处理器中使用写掩码实现混合写回/写穿透策略[J]. 计算机学报, 2008, 31(11): 1918-1928.

LIN Wei, YE Xiaochun, SONG Fenglong, et al. Using write mask to support hybrid write-back and write-through cache policy on many-core architectures [J]. Chinese Journal of Computers, 2008, 31(11): 1918-1928.

[4] KANG S Y, PARK S, JUNG H Y, et al. Performance tradeoffs in using NVRAM write buffer for flash memory-based storage devices[J]. IEEE Transactions on Computers, 2009, 58(6): 44-758.

[5] KANG W Y, SON S H, STANKOVIC J A. Power-aware data buffer cache management in real-time embedded databases[C]// The 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications. Piscataway, NJ, USA: IEEE, 2008: 35-44.

(编辑 苗凌)