

引用格式:张淑,田泽,郑新建,等.面向低抖动 GPU 像素 Cache 的像素写合并缓冲技术[J].微电子学与计算机,2019,36(7):93-97. [ZHANG S, TIAN Z, ZHENG X J, et al. Pixel write combining buffer technology for pixel cache low thrash of GPU[J]. Microelectronics & Computer, 2019, 36(7):93-97.]

面向低抖动 GPU 像素 Cache 的像素写合并缓冲技术

张 淑¹, 田 泽^{2,3}, 郑新建^{2,3}, 张 骏^{2,3}, 许宏杰^{2,3}, 王 维¹

(1 西安翔腾微电子科技有限公司, 陕西 西安 710068;

2 中国航空工业集团 西安航空计算技术研究所, 陕西 西安 710068;

3 集成电路与微系统设计航空科技重点实验室, 陕西 西安 710068)

摘 要: 图形处理器像素 Cache 访问时抖动发生频繁, 很大程度的影响了图形处理器的性能. 通过研究图形处理器中多数数据流处理的并行化特征, 提出了一种像素写合并缓冲技术, 在数据写入像素 Cache 前, 根据地址对像素进行合并后再对 Cache 进行访问, 能够减少对 Cache 的访问次数, 降低 Cache 抖动, 提高 Cache 性能. 使用写合并缓冲技术能将 Cache 抖动平均降低 60%, 最大降低 70%, 测试表明像素 Cache 的写合并缓冲技术能够降低 Cache 的抖动, 非常适用于嵌入式图形处理器像素 Cache 的设计.

关键词: Cache 抖动; 像素缓冲; 数据合并; 图形处理器

中图分类号: TP391.4

文献标识码: A

文章编号: 1000-7180(2019)07-0093-05

Pixel write combining buffer technology for pixel cache low thrash of GPU

ZHANG Shu¹, TIAN Ze^{2,3}, ZHENG Xin-jian^{2,3},
ZHANG Jun^{2,3}, XU Hong-jie^{2,3}, WANG Wei¹

(1 Xi'an Xiangteng Microelectronics Technology Company Limited, Xi'an 710068, China;

2 Aeronautics computing technique research institute, Aviation industry corporation of China, Xi'an 710068, China;

3 Key Laboratory of Aviation Science and Technology on Integrated Circuit and Micro-System Design, Xi'an 710068, China)

Abstract: The thrashing of pixel cache in GPU is frequent, which greatly decrease the performance of GPU. Based on the research of the parallelization characteristics of multi data stream processing in GPU, a pixel write merging buffer technique is proposed. The cache access is merged according to the address before. It can reduce the number of cache access, reduce the cache thrashing, and improve the performance of the cache. The pixel write combining buffer technology can reduce the average cache trash by 60% and the maximum reduction of 70%. The result shows that the write merge buffer technology of pixel cache can reduce the cache thrash, and is very suitable for the design of the pixel cache of the embedded graphic processor.

Key words: Cache thrash; pixel buffer; data merge; GPU

1 引言

随着计算机系统的迅速发展, 现代计算机系统

越来越受主存储器性能的限制. 处理器的性能以每年 60% 的速度在增长, 而主存储器芯片带宽仅仅以每年 10% 的速度在增长. 在速度方面, 主存储器和

处理器一直保持了大约一个数量级的差距. 由于处理器和存储器的速度差距越来越大,Cache 作为弥补处理器与主存之间速度巨大差异的桥梁,正成为影响系统性能的关键因素之一^[1].

针对处理器中 Cache 性能提高的研究有很多^[2]. 这些研究或针对 Cache 本身的替换算法进行优化和改进^[3-7],或面向数据对象采用不同的 Cache 管理策略相结合的方法^[8-11]. GPU 拥有复杂的 Cache 系统^[12],除了针对 Cache 设计本身的性能提高策略以外,还可以针对 GPU 数据对象的特点,在数据写入 Cache 前对数据进行处理,提高数据的连续性,从而在原有 Cache 结构的基础上进一步提高 Cache 的访存性能.

本文分析了像素 Cache 在图形处理器(Graphic Process Unit, GPU)中的应用,并对 3D 图形处理流程进行了研究:由于图元的并行处理和串行输出造成了同一图元的像素乱序输出,进而造成了 Cache 抖动. 针对这一问题提出了像素写合并缓冲技术,降低了像素 Cache 的抖动.

2 研究动机

图 1 是一种比较通用的图形处理器阵列结

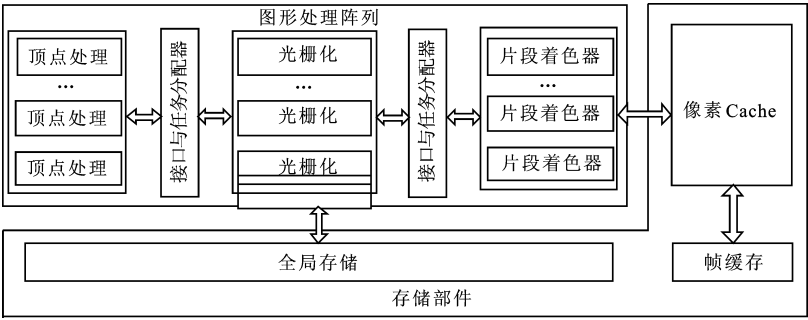


图 1 GPU 工作流程图

在这种并行化的处理结构中,光栅化阵列会同时处理多个图元,处于不同屏幕位置的点、直线、三角形这三种基本图元的一个或多个可以同时光栅化^[17-18]. 当多个光栅化单元同时工作时,将会有地址空间跨度较大的数据经过片段着色器阵列依次连续写入像素 Cache. 因此,不难理解,图形处理器在进行大量的图元绘制时必然会造成 Cache 抖动的频繁发生.

图 2 给出了绘制多个不连续存储的图元时造成的 Cache 抖动. 图中 Cache 包含 2 个块,每个块可以容纳图中四分之一的像素数据,数据宽度是 128 bits. 每个像素的数据宽度是 32 bits(RGBA 为 8×8

构^[13-16],采用多级流水的方式进行图形处理,包含多级处理部件,每一级处理部件包含多个并行的处理单元,对多组不同的数据进行相同的处理,完成图形处理的一个阶段. 绘图信息经过多级流水部件的处理后最终产生屏幕可显示的像素.

3D 图形处理的具体工作流程如下:来自 API 的绘图信息经过处理以后产生顶点数据,首先顶点处理部件对输入的顶点信息进行分析处理,确定 3D 图形的形状及位置关系,建立基本类型的图元;接着由光栅化部件进行基本图元的扫描——对不同类型的图元采用不同的光栅化算法,将已经建立的 3D 图形转化成屏幕上对应的像素点;然后由片段着色器计算光栅化后每个像素点的颜色并进行后续处理. 最后将处理好的颜色数据经由像素 Cache 发送到显存中.

值得注意的是,在图形流水线的各个部件之间都存在一个“接口与任务分配器”,设立该分配器的目的是实现数据在不同处理单元之间的调度,实现数据处理的并行化,同时有效分配处理阵列中的各个数据处理单元.

$\times 8 \times 8$ bits),则每个地址可以存 4 个地址连续的像素数据. 当图中 A、B、C、D、E 五个图元同时被五个不同的光栅化单元处理时,将会有多个地址不连续的数据依次写入像素 Cache.

假设数据写入像素 Cache 的先后顺序为 $A_0-B_0-C_0-E_0-D_0-B_1-C_1$ (数据同时产生,但由于存储带宽的限制,数据访问存储的先后顺序完全有这样的可能). 图 2(a)表示访存开始时 A_0 和 B_0 所在的 Cache 块依次被调入;图 2(b)展示了写像素 C_0 的过程,按照 LRU 替换算法,处理器访问 C_0 时块 0 被替换出 Cache, C_0 所在的块 2 进入 Cache,图 2(c)至图 2(f)展示了随后 6 次访存操作在 LRU 替换算法下的

Cache 变化过程. 块 0 在图 2(b)被替换出 Cache 后, 在 2(c)又重新进入 Cache,这期间 Cache 没有任何其他数据被访问过,说明 Cache 发生了抖动. 同理, 块 1 在图 2(c)离开像素 Cache,在图 2(e)进入像素 Cache,期间除块 3 发生变化其他数据均没有被访问,发生了 Cache 抖动. 此例中,由于 LRU 算法的替换策略以及图形处理器的并行处理特性导致 Cache 永远没有办法命中. 而且每次回写的数据带宽利用率很低,只有 1/4 是有效的.

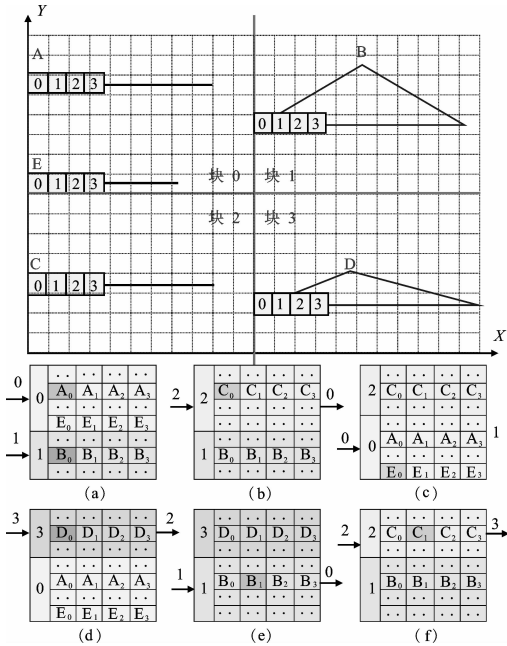


图 2 图元绘制引起的像素 Cache 抖动

针对每一个光栅化单元来说,扫描产生的像素坐标是连续的,在存储中地址也是连续的. 这样产生的数据可以根据地址连续设立像素 Cache 收集缓冲区,数量与光栅化处理单元数量相同,每个缓冲区的宽度与存储宽度相同,深度为 1. 将属于同一个物理地址的数据合并后统一发送到像素 Cache. 例如,将图 2 中 A 图形的像素数据分别进行收集,当收集完 $A_0 A_1 A_2 A_3$ 后写像素 Cache,如此对像素 Cache 的访存次数降低 75%,同时数据带宽的利用率提高 75%. 很大程度上降低了 Cache 抖动.

3 像素写合并算法描述

像素写合并技术是将写入像素 Cache 的数据进行分类和收集,暂存到一个容量较小的 sram 缓冲区中. 该缓冲区根据图形处理器光栅化单元的数量、数据宽度以及 Cache 的数据宽度和存储地址划分进行设定. 缓冲区的行数与光栅化单元数量相同. 每一

行存储的数据对应一个存储地址. 每一行存储的像素个数是 Cache 的数据宽度/像素数据宽度. 为每行的数据设置写入标记位,数据写入像素 Cache 时选择写入数据最多的行.

本设计的嵌入式 GPU 光栅化单元的数量为 4, 处理器每个像素数据的位宽为 32 bits(RGBA 分别为 $8 \times 8 \times 8 \times 8$ bits),写像素 Cache 的数据宽度为 128 bits,存储按照显示器屏幕的坐标连续进行划分(每个物理地址可以存储屏幕上连续四个像素值). 像素写合并算法具体描述如下: 在写像素 Cache 前设置 4×128 bits(64 B)的像素收集缓冲区. 每像素占用 32 bits,一行缓冲 4 个像素,与 Cache 写入的数据宽度相同. 每行对应 4 位有效位,每位对应该行中的一个像素(共 2 B).

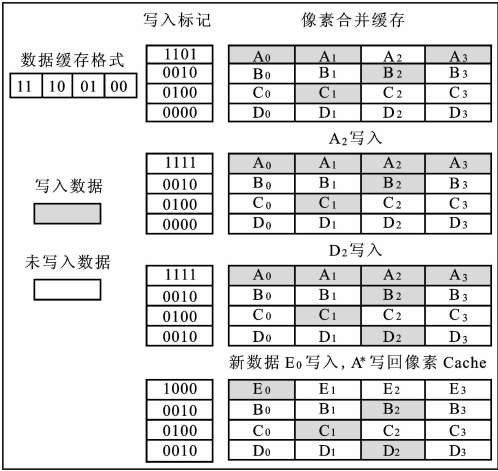


图 3 像素缓存与回写

图 3 显示了具体的像素缓存与回写策略. 根据像素的坐标值判断连续输入像素的位置是否连续, 如果 Y 坐标相同, X 坐标低两位以外的其它位相同则输入像素位置连续. 像素按照 X 坐标的低两位进行数据缓存,缓存格式如图 3 所示: 11 在最高位, 00 在最低位. 如果接收到的像素是连续的, 则把数据直接写入缓冲区, 继续接收新的像素, 如图中数据 A_2 . 如果接收到的像素与已经缓冲的数据不连续, 则选择新的缓冲区写入, 如图中数据 D_2 . 当有新的地址不连续的像素输入, 且所有写入标记不为 0 时, 选择有效像素最多的行写回像素 Cache, 并将新像素写入该行, 如图中数据 E_0 . 如果有效像素最多的行有多个, 则选择地址较小的写回.

像素写合并算法流程如图 4 所示. 首先把输入像素的坐标与缓冲区中已存在的坐标进行比较, 判断是否需要进行数据合并, 如果需要则根据坐标位

置判断(X 坐标的低两位进行判断)缓存位置并更新缓存标记位,否则查看缓冲区中是否有空的存储位置,如果有则根据坐标值写入相应位置并更新缓存标记位,否则选择一个缓存数据最多的存储区回写到像素 Cache,同时将数据写入相应位置并更新缓存标记位.

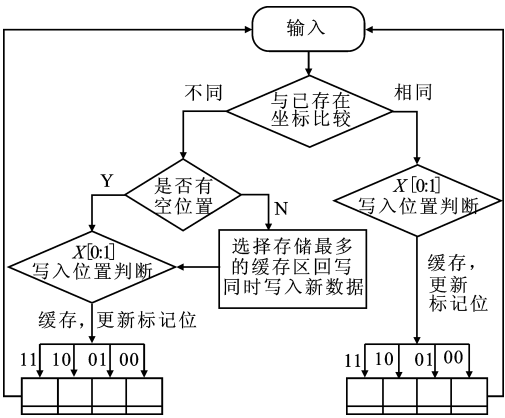


图 4 像素写合并算法流程

4 像素写合并缓冲技术的实现

像素写合并缓冲的硬件结构如图 5 所示. 数据由图形处理阵列产生,经过像素写合并单元,最终写入到像素 Cache.

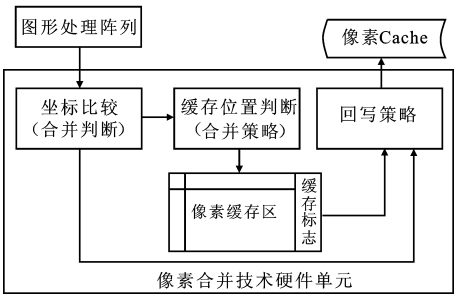


图 5 像素写合并硬件结构

图中,坐标比较单元根据像素的屏幕坐标(X,Y)(比较位根据具体设计选择,本文用例中不对 X 坐标的低两位进行判断),判断新的像素是否可以与已存在的像素进行合并;像素合并的方法则是根据坐标值选择写入存储的位置;回写策略则主要根据缓存标志位选择回写的缓冲区地址.

5 虚拟仿真与结果分析

采用 System Verilog 构建虚拟仿真平台,并进行功能验证. 虚拟仿真平台如图 6 所示,激励产生模块负责产生测试激励驱动图形处理单元,监控模块

记录 Cache 的访存次数、命中率和替换次数.

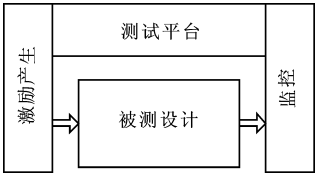


图 6 虚拟仿真平台

本文采用的测试用例是绘制图 2 所示的五个图形,记录从绘制开始到像素完全写入像素 Cache 的这段时间内图形处理器对 Cache 的访存次数、命中率和替换次数. 测试结果如图 7 所示. 结果表明本设计可以使 Cache 访存次数减少 70%,Cache 替换次数减少 60%. 有效降低了 60%的 Cache 抖动(最大可降低 70%),提高了 60%的存储带宽利用率.

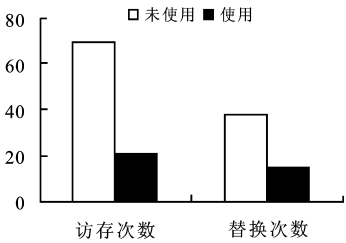


图 7 仿真结果统计

6 结束语

本文对图形处理器 3D 图形处理结构造成的像素 Cache 抖动问题进行了分析研究,提出一种像素写合并缓冲技术,在像素写入 Cache 前进行收集合并,减少了像素 Cache 的访存次数,降低了像素 Cache 的抖动,提高了存储带宽的利用率. 验证结果表明,Cache 抖动平均降低 60%,最大降低 70%,存储带宽利用率平均提高 60%.

另外,针对写入数据进行像素合并缓冲,会产生读写数据的一致性问 题,使得从 Cache 中读到的数据可能不是最新数据. 所以使用写合并缓冲技术时,数据读操作也要经过数据缓冲区的检索,如果读数据不在数据缓冲区中,再对 Cache 进行操作. 从这个角度来看,该技术同时降低了读写数据产生的 Cache 抖动.

参考文献:

[1] 段宗涛,周兴社,董明峰. 主存储器调度技术研究[J]. 微电子学与计算机,2004,21(10): 55-59.
[2] 陈占龙,张丁文,吴亮,等. 面向多核处理器的共享 cache 优化研究进展[J]. 计算机应用研究, 2014, 31

- (10):2881-2887.
- [3] GHASEMZADEH H, MAZROUEE S, KAKOEE M R. Modified pseudo LRU replacement algorithm [J]. Engineering of Computer Based Systems, 2006(52): 27-30.
- [4] 吴俊杰,杨学军,曾坤,等. DOOC:一种能够有效消除抖动的软硬件合作管理 Cache[J]. 计算机研究与发展, 2008,12(12):2020-2032.
- [5] 李洪,毛志刚. PLRU 替换算法在嵌入式系统 cache 中的实现[J]. 微处理机, 2010,2(1): 6-19.
- [6] 刘松鹤,宋焕生,等. 基于分支路径跟踪的猜测访存数据 Cache 污染控制技术[J]. 计算机应用研究, 2013,30(7):2064-2068.
- [7] 孙玉强,王文闻,巢碧霞,等. 基于预取的 Cache 替换策略[J]. 微电子学与计算机, 2017, 34(1):85-89.
- [8] SORYANI MOHSEN, SHARIFI MOHSEN. Performance evaluation of cache memory organizations in embedded systems [J]. Information Technology, 2007: 1045-1050. DOI:10.1109/ITNG.2007.150
- [9] 张骏,田泽,梅魁志,等. 基于节点预测的直接 Cache 一致性协议[J], 计算机学报, 2014, 37(3):700-703.
- [10] 郑新建,田泽,张骏. 一种低功耗指令 Cache 的设计与实现[J]. 微电子学与计算机, 2015, 32(7):25-28.
- [11] 郑新建,龙强,王维. 一种多端口非阻塞纹理 Cache 设计与实现[J]. 无线电工程, 2018, 3(48):198-202.
- [12] ZHANG J. XY-Type GPU cache: exploiting spatial localities in both X and Y directions to avoid conflict miss [J]. Chinese Journal of Electronics, 2015, 24(1): 88-95.
- [13] 辉达公司. 用于图形处理器的并行阵列结构[P]. 中国. CN101371247A. 2009. 2. 8.
- [14] 姜晓丽,王伟,隋毅. GPU 工作流程简介[J]. 科技创新导报, 2010(24):46-47.
- [15] SHREINER D. OpenGL programming guide [M]. 7 版. 李军,译. 北京:机械工业出版社, 2010.
- [16] 刘坚. 嵌入式多核 GPU 渲染流水线的研究与实现 [D]. 四川:电子科技大学, 2015.
- [17] 田泽,刘天江,张骏,等. 基于扫描线填充的三角形图元双向光栅化技术[J]. 小型微型计算机系统, 2015, 6(6): 1398-1402.
- [18] 王志方. 嵌入式 GPU 中图形光栅化的设计与实现 [D]. 长沙:湖南大学, 2016.

作者简介:

张淑女, (1989-), 硕士, 工程师. 研究方向为 GPU 设计、SoC 设计. E-mail: zhangshuxyz@163.com.

田泽男, (1965-), 博士, 研究员. 研究方向为 SoC 设计方法学、航空专用集成电路设计.

郑新建男, (1980-), 博士, 工程师. 研究方向为 GPU 设计、SOC 设计.

张骏男, (1978-), 博士后, 高级工程师. 研究方向为计算机体系结构、GPU 设计、微处理器设计、SoC 设计.

许宏杰男, (1981-), 高级工程师. 研究方向为航空总线、视频编码及图形处理器设计.

王维女, (1988-), 硕士, 工程师. 研究方向为 GPU 设计、SOC 设计.