

适用于嵌入式应用的多线程顶点处理器设计

申智波¹, 陈 杰¹, 刘 建¹, 王 云¹, 朱 玥^{1,2}

(1 中国科学院微电子研究所, 北京 100029; 2 中国科学技术大学, 安徽 合肥 230026)

摘 要: 针对嵌入式应用中三维图形渲染的要求, 设计了一款可编程的多线程顶点处理器. 该顶点处理器采用单指令多数据结构, 一条指令能够同时处理 4 个单精度浮点数, 并采用多线程技术, 支持 4 个线程并发执行, 能够有效地减少发生数据读写冲突时的停顿周期数, 提高了处理效率. 相对于单线程结构, 4 线程顶点处理器在较小的硬件开销下, 可以实现 2.1~2.8 倍的性能提升. 该顶点处理器支持 OpenGL ES 1.1 和 Vertex Shader Model 1.1, 在 90nm CMOS 工艺库下可实现频率为 200 MHz, 性能为 50 Mvertices/s.

关键词: 图形处理器; 三维图形渲染; 可编程顶点处理器; 多线程; 单指令多数据; 嵌入式应用

中图分类号: TN391.41

文献标识码: A

文章编号: 1000-7180(2012)07-0076-05

Design of Multithreaded Vertex Processor for Embedded Applications

SHEN Zhi-bo¹, CHEN Jie¹, LIU Jian¹, WANG Yun¹, ZHU Yue^{1,2}

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

2 University of Science and Technology of China, Hefei 230026, China)

Abstract: A programmable multithreaded Vertex Processor is designed for 3D graphics rendering in embedded applications. The Single Instruction Multiple Data (SIMD) architecture is proposed, so four single precision floating-point numbers are processed simultaneously. The multithreaded datapath supports parallel execution of four threads, so the pipeline stalls are reduced when data hazards occur and the processing efficiency rises up. The proposed four-threaded Vertex Processor achieves 2.1~2.8 times of performance improvement at a relatively small hardware cost compared to the single-threaded architecture. The Vertex Processor supports OpenGL ES 1.1 and Vertex Shader Model 1.1. Implemented in 90nm CMOS process, it achieves 50 Mvertices/s of geometry performance at the operating frequency of 200MHz.

Key words: graphics processing unit; 3D graphics rendering; programmable vertex processor; multithreading; SIMD; embedded application

1 引言

图形处理器(Graphics Processing Unit, GPU)被广泛应用于个人电脑、工作站或者游戏机上,用于图形处理任务的硬件加速. 随着硬件性能的提升和价格的下降,以智能手机、平板电脑和掌上游戏机为代表的嵌入式移动设备逐渐普及, GPU 在嵌入式设备上也得到了广泛的应用,以满足这些设备图形计

算任务的需求. 早期的移动设备没有专用的图形处理硬件, 屏幕分辨率较低, 并且只有 2D 图形显示, 图形处理的任务由嵌入式中央处理器(CPU)完成. 在高分辨率和 3D 图形显示的情况下, 图形处理会占用大量的 CPU 资源, 因此必须采用 GPU 进行图形处理加速. GPU 协助嵌入式微处理器完成图形处理和显示, 节省了微处理器的资源消耗, 提高了图形处理性能和显示效果, 增强了电子产品的易用性和

收稿日期: 2011-11-25; 修回日期: 2011-12-22

基金项目: 国家“八六三”计划项目(2009AA011700)

用户体验。

GPU 中可编程处理器负责大部分的图形计算任务,一般分为顶点处理器和像素处理器(Pixel Processor)。顶点处理器负责顶点的几何变换、光照计算和纹理坐标计算等。针对嵌入式设备低功耗高性能的要求,本文设计了一款支持 4 线程的可编程顶点处理器,它支持 OpenGL ES 1.1^[1] 和 Vertex Shader Model 1.1^[2]。

2 嵌入式 GPU 图形处理

嵌入式 GPU 图形处理的流程如图 1 所示。分为 4 个步骤:顶点处理(Vertex Processing)、图元组装(Primitive Assembly)、光栅化(Rasterization)、片段操作(Per-Fragment Operations)。

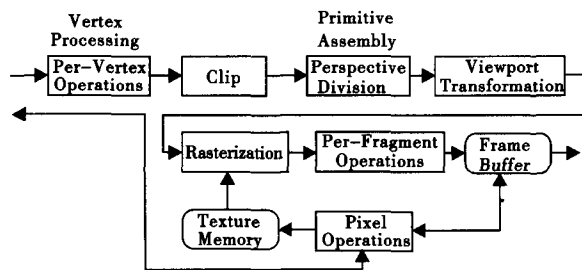


图 1 GPU 图形处理流程

早期的 GPU 中图形渲染管线的功能是固定的^[3],无法满足现实应用中各种各样复杂的场景渲染的要求。可编程图形管线允许用户编写程序实现图形处理算法,不但能够完成固定功能 GPU 所能完成的图形处理功能,还能通过编程实现新的算法和功能,具有灵活性和通用性。

3 相关工作

文献[4-5]设计的顶点处理器采用了定点数据运算单元和单线程结构,虽然硬件资源消耗相对较小,但是定点数据运算无法满足高精度屏幕显示的要求,并且单线程结构在发生数据写读冲突时会有较多的停顿周期;文献[6-7]设计的顶点处理器采用了单精度浮点数运算单元,满足了数据处理精度的要求,但仍然是采用单线程结构;文献[8]设计的顶点处理器采用了单精度浮点数运算单元,结构上采用了超长指令字(VLIW)和多线程,并且添加了输入输出缓存(Cache),虽然系统性能相对较高,但是硬件开销大。本文综合考虑了性能、硬件开销和数据处理精度这三个方面。

4 顶点处理器设计

4.1 系统结构

顶点处理器采用 SIMD 结构,每条指令可以处理 4 个 IEEE 754 单精度浮点数,并采用多线程结构,提高了处理效率。

顶点处理器分为以下几个部分^[9]:输入输出控制模块、逻辑功能模块、流水线寄存器、存储单元,如图 2 所示。输入输出控制模块负责顶点处理器与外界的数据传输;逻辑功能模块是组合逻辑,负责控制和运算功能;流水线寄存器是各级流水线之间暂存结果的寄存器;存储单元用于保存指令和寄存器数据。图 2 显示了顶点处理器的结构和流水线。

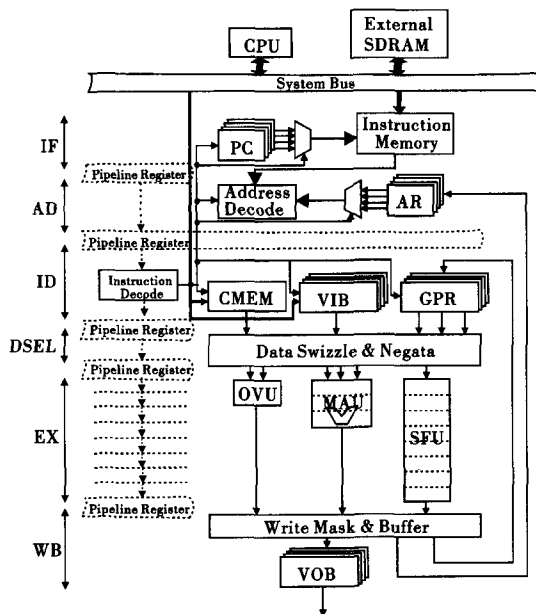


图 2 系统结构和流水线

指令和数据在 CPU 的控制下,通过系统总线写入顶点处理器的指令存储单元和寄存器。处理完毕的顶点数据保存在输出寄存器中,由 GPU 后续硬件处理。为了使 GPU 处理当前组顶点和传送下一组顶点数据可以同时进行,输入和输出寄存器采用双缓存(Double-buffered)结构。系统结构详细介绍见本节其它小节。

4.2 寄存器组

顶点处理器中寄存器组包含以下 5 类:

- (1) 输入寄存器(Vertex Input Buffer, VIB);
- (2) 输出寄存器(Vertex Output Buffer, VOB);
- (3) 常量寄存器(Constant Memory, CMEM);
- (4) 通用寄存器(General Purpose Register,

GPR);

(5) 地址寄存器(Address Register, AR).

VIB 存放待处理顶点的输入数据,包括顶点坐标、法向量、纹理坐标等,每条指令只能读取一个 VIB 寄存器;VOB 存放处理后顶点的输出数据,用于光栅化等后续处理,包括变换后的顶点坐标、颜色、纹理坐标等;CMEM 存放一组顶点共用的参数,包括变换矩阵、光照参数等,每条指令只能读取一个 CMEM 寄存器,可以采用直接寻址或者用 AR 相对寻址;GPR 保存顶点处理器运行过程中的中间变量,每条指令最多可以同时读取 3 个和写回 1 个 GPR 寄存器;AR 只用于 CMEM 的寻址.如表 1 所示.

表 1 寄存器组

寄存器类型	数目	比特数	符号	读写
VIB	8	$2 \times 4 \times 8 \times 128$	v0~v7	RO
VOB	6	$2 \times 4 \times 6 \times 128$	o0~o5	WO
CMEM	128	128×128	c0~c127	RO
GPR	12	$4 \times 12 \times 128$	r0~r11	R/W
AR	1	4×10	a0	R/W

VIB 和 VOB 为双缓存结构.在顶点处理器处理顶点之前,CPU 通过系统总线将顶点数据和常量数据写入 VIB 和 CMEM 寄存器.

4.3 指令集

根据顶点处理器所要完成的功能,指令集包含 5 类指令:(1)乘加指令(Multiply and Add Unit, MAU),包括 ADD、MUL、DST、MAD、DP3、DP4;(2)特殊函数求值指令(Special Function Unit, SFU),包括 RCP、RSQ、LOG、EXP、LIT;(3)普通向量指令(Ordinary Vector Unit, OVU),包括 MOV、FRC、MIN、MAX、SLT、SGE;(4)地址寄存器加载指令 ARL (Address Register Load);(5)空操作指令 NOP.指令的位宽固定,为 64bit.

所有指令(NOP 除外)按源操作数的类型分为两类指令:向量指令(Vector Instruction)和标量指令(Scalar Instruction).MAU 类型、OVU 类型和 LIT 指令属于向量指令,它们的源操作数是 4 维向量;SFU 类型(LIT 除外)和 ARL 指令属于标量指令,它们的源操作数只能是 4 维向量的某一个分量.MAU、OVU 和 SFU 类型指令的目的操作数都是 4

维向量,如果源操作数是标量,则指令计算的结果拓展成 4 维向量.ARL 指令的目的操作数只能是地址寄存器 AR.

4.4 指令流水线

指令流水线如图 2 所示,分为 6 个阶段:取指(Instruction Fetch, IF)、地址译码(Address Decode, AD)、指令译码(Instruction Decode, ID)、数据选择(Data Select, DSEL)、执行(Execute, EX)、写回(Write Back, WB).

AD 阶段生成读取 CMEM、VIB、GPR 寄存器的地址,如果是使用 AR 寄存器相对寻址,则将 AR 寄存器中的基址值加上偏移地址,得出读取 CMEM 寄存器的地址.ID 阶段^[10-11]控制整个系统流水线的运行,并且源操作数的读取也在这个阶段进行.DSEL 阶段完成源操作数中各个数据分量的重组和取负,并将数据分量分配给运算功能单元 MAU、SFU 或 OVU.EX 阶段,执行实际的向量或者标量数据运算,执行周期数最多为 8 个周期;OVU 单元负责 OVU 类型指令的计算,需要 1 个执行周期;MAU 单元负责 MAU 类型指令的计算,需要 1 个到 3 个执行周期;SFU 单元负责 SFU 类型指令的计算,除了 LIT 指令需要 8 个执行周期之外,其它指令需要 4 个执行周期.WB 阶段中含有写缓存(Write Buffer),缓存 EX 阶段的输出结果,以避免在发生数据写写冲突的情况下,流水线产生停顿.

4.5 单指令多数据

计算机图形中所要处理的数据大部分都包含 3 个或者 4 个分量,例如向量或点的齐次坐标、法向量、纹理坐标、颜色分量 RGBA,因此系统采用 SIMD 结构,每条指令包含 4 个数据分量,表示为 (x, y, z, w) .为了满足数据处理精度和范围的要求,采用 IEEE 754 单精度浮点数.顶点处理器中所有寄存器的位宽都是 128bit(AR 除外).

实际应用中,源操作数任意一个数据分量的出现次数或者位置都是任意的,目的操作数任意一个数据分量可以有选择性地被写回,这样才能够提供给程序员足够的灵活性.因此,顶点处理器中,指令的源操作数支持数据分量的重组(Data Swizzle),目的操作数的数据分量支持写掩码(Writing Mask).例如指令,

ADD r0. xyz, v0. zwxx, c0. zwxx

该指令表示寄存器 v0 和 c0 的分量 (z, w, x, x) 对应相加,结果只写回寄存器 r0 的 (x, y, z) 分量, w 分量保持原来的值不变.

指令的源操作数可以取负(Negate),因此不需要减法指令.指令集中没有计算 3 维向量叉积的指令,但是可以用如下两条指令替代.

```
MUL r0, v0, zxyw, c[a0+4].yzxw
MAD r0, v0, yzxw, c[a0+4].zxyw, -r0
```

4.6 多线程

传统的单线程处理器在发生数据读写冲突时,流水线需要停顿以等待正在执行的指令结果的写回,因此浪费了许多时钟周期,系统的效率降低.相对于 CPU 的数据运算,GPU 中顶点处理器的数据运算更加复杂,DP3 和 DP4 的执行周期数为 3 个,

RCP、RSQ、LOG、EXP 的执行周期数为 4 个,LIT 指令由于要进行幂运算,执行周期数最多,为 8 个.因此,相对于 CPU 而言,顶点处理器中发生数据读写冲突时,将会有更多的停顿周期,损耗了系统性能.

采用多线程结构则可以改善这种问题,不同线程之间的指令交替执行,能够减少停顿周期,提高了处理效率.执行图 3 所示的指令,单线程处理器会产生 5×4 共 20 个周期的停顿,而 4 线程处理器只有 2 个停顿周期.

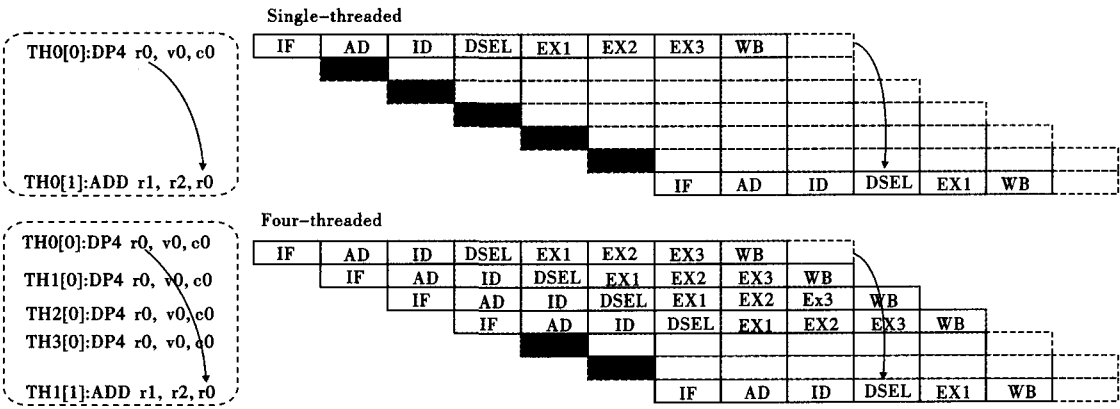


图 3 单线程和多线程比较

相对于单线程结构,多线程结构花费的硬件开销较小.本设计中,只需要为每个线程提供一套相应的寄存器即可,即 VIB、GPR、VOB、AR 寄存器. CMEM 寄存器由 4 个线程共享,因此只需要 1 份,运算单元 MAU、SFU、OVU 由 4 个线程共享.

5 设计结果

采用 90nm CMOS 工艺库实现该顶点处理器,工作频率可达到 200MHz,等效逻辑门数量为 11.6 万,片内存储器大小为 5.7 KB,顶点处理性能可达 50 Mvertices/s,.

表 3 显示了单线程与多线程结构顶点处理器综合后等效逻辑门数量和片内存储器大小的对比.图 4 是两种结构的性能对比,采用了一些典型的顶点程序作为测试程序,归一化至单线程结构运行的周期数.从测试数据可以看出,相对于单线程结构,4 线程顶点处理器可以实现 2.1~2.8 倍的性能提升.

表 3 综合结果

	单线程	4 线程	4 线程/单线程
等效逻辑门	11.2 万	11.6 万	1.04
片内存储器	3.7KB	5.7KB	1.54

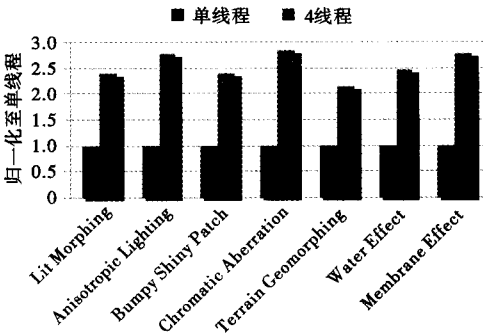


图 4 性能对比

6 结束语

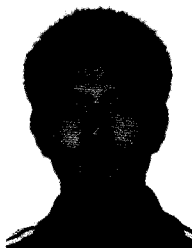
针对嵌入式应用,本文设计了一款支持 4 线程

的顶点处理器,采用多线程结构,可以有效地减少发生数据读写冲突时的流水线停顿周期数,提高了处理效率.在相对较少硬件开销的前提下,性能有了较大地提升.

参考文献:

- [1] Khronos Group. OpenGL ES common profile specification[M]. USA: Khronos Group, 2008.
- [2] Microsoft. Direct 3D[EB/OL]. [2011-05-17]. [http://msdn.microsoft.com/en-us/library/windows/desktop/hh309466\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/hh309466(v=VS.85).aspx).
- [3] Hill F S, Kelley S M. 计算机图形学(OpenGL 版)[M]. 3 版. 胡事民,译. 北京:清华大学出版社,2009.
- [4] Sohn Ju-Ho, Woo Jeong-Ho, Lee Min-Wuk, et al. A 155-mW 50-mvertices/s graphics processor with fixed-point programmable vertex shader for mobile applications[J]. IEEE Journal of Solid-State Circuits, 2006, 41(5): 1081-1091.
- [5] 杨毅,郭立,史鸿声,等. 一个面向移动设备的可编程顶点处理器的设计[J]. 中国科学技术大学学报, 2010, 40(2): 140-145.
- [6] Arakawa Fumio, Yoshinaga Takeshi, Hayashi Tomochi, et al. An embedded processor core for consumer appliances with 2.8GFLOPS and 36M polygons/s FPU[C]//Digest of Technical Papers. USA: IEEE ISSCC, 2004: 334-343.
- [7] Kim Donghyun, Chung Kyusik, Yu Chang-Hyo, et al. An SoC with 1.3 gtexels/s 3-D graphics full pipeline for consumer applications[J]. IEEE Journal of Solid-State Circuits, 2006, 41(1): 71-84.
- [8] Yu Chang-Hyo, Chung Kyusik, Kim Donghyun, et al. An energy-efficient mobile vertex processor with multithread expanded VLIW architecture and vertex caches[J]. IEEE Journal of Solid-State Circuits, 2007, 42(10): 2257-2269.
- [9] Lindholm Erik, Kilgard Mark J, Moreton Henry. A user-programmable vertex engine[C]//Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. Los Angeles, CA, USA: ACM SIGGRAPH'01, 2001: 149-158.
- [10] 韩亮,陈杰,史鸿声,等. 嵌入式可重构 DSP 处理器的指令译码器设计[J]. 微电子学与计算机, 2004, 21(7): 91-94.
- [11] 敖天勇,陈杰,刘建,等. 基于 Costar II 的异构多核 DSP 设计与实现[J]. 微电子学与计算机, 2007, 21(3): 59-62.

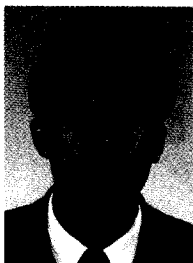
作者简介:



申智波 男,(1987—),硕士研究生.研究方向为集成电路设计.

(上接第 75 页)

作者简介:



秦怀斌 男,(1980—),博士研究生,副教授.研究方向为软件工程、复杂网络.



郭理 男,(1968—),硕士,副教授.研究方向为软件工程、人工智能.