

图元装配的硬件实现与验证

周高财,李 涛,贺 琼

(西安邮电大学 电子工程学院,陕西 西安 710121)

摘 要: 图形流水线是 GPU (Graphics Processing Unit)工作的通用模型,在图形流水线中,图元装配位于关键环节,为了提高改善图形流水线的速度和渲染效果,快速准确地产生图元成为主要因素之一.立足当前图形流水线的要求,首先采用 SystemC 语言建立图元装配单元的行为模型.然后采用 Verilog 语言实现其硬件电路描述,同时采用 SystemVerilog 语言搭建验证平台,完成图元装配单元的功能验证.通过各方面验证,表明所设计的图元装配单元满足系统设计要求.

关键词: 图形处理器;SystemC 行为模型;硬件实现;SystemVerilog

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2017)12-0079-04

Hardware Primitive Assembler Implementation and Verification

ZHOU Gao-cai, LI Tao, HE Qiong

(School of Electronic Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710121, China)

Abstract: Graphics pipeline is a common model of GPU (Graphics Processing Unit) work. primitive assembler is one of the key stage in the graphics pipeline, in order to improve the efficiency and quality of rendering by the graphics pipeline, quickly and accurately generate primitives as one of the main factors. This paper meets the requirements of the current graphics pipeline, this paper uses SystemC language to build the behavior model of primitive assembler. And uses Verilog language to perform hardware design of primitive assembler unit. And further more, this paper uses SystemVerilog language to build the hierarchical verification environment which facilitates the design modules. Through all aspects of verification, this paper shows that the design of the primitive assembler unit metaphor to meet the system design requirements.

Key words: graphics processing unit; systemC behavioral modeling; hardware implementation; SystemVerilog

1 引言

随着半导体工艺的不断进步,中央处理器(Central Processing Unit, CPU)的频率和速度都快速提高.但伴随而来的却是高功耗的阻碍,使得其发展速度受到很大的压制.然而图形处理器(GPU)以大大超出摩尔定律的速度高速发展,图形处理器的发展极大地提高了计算机图形处理的速度和图形质量,并促进了与计算机图形相关应用领域的快速发展^[1].在传统计算机中,对于图形图像的处理依赖于CPU的处理能力,由于CPU自

身限制导致其处理图形图像的速度与质量无法满足需求. GPU的出现减少了在图形处理方面对于CPU的依赖, GPU将图形图像处理功能集成在显卡内部,从而在处理图形方面具有更快更好的处理速度与效果.

随着科技的进一步发展,种类繁多的电子智能设备已经深入到人类生活的众多领域,不论是通用计算机领域或嵌入式应用领域.对多媒体和图形图像处理技术的需求都在日益增加^[2].近年来, GPU受到顶点染色、像素染色、几何渲染、物理计算和通用计算等市场需求的推动,日益成为行业产业和科学研

究的焦点.

本文讨论设计的自主研发的 GPU 不同于其他 GPU 的是,采用顶点染色器与像素染色器分离的结构和多种硬件加速器,没有采用统一着色器结构.其加速器结构简单、速度快、功耗低;同时采用不同于传统结构的并行处理和存储分配方法^[3]. 本文重点讨论的是自主研发 GPU 硬件加速器中的图元装配组件的硬件电路设计和验证.

2 SystemC 模型

SystemC 既是系统级语言,也是硬件描述语言,是建立在 C++ 编程语言基础之上的. 它由一组描述硬件行为的 C++“类“库和一个支持硬件时序仿真的 C++ 内核“类”组成,实现 C、C++ 语言的 ASIC 设计与 RTL 电路模型代码自动综合^[4]. 本文利用 SystemC 在建模能力上的优点,对图元装配进行行为建模,便于对于硬件设计提供参考,而且能保证硬件的实现.

图元装配是通过算法对客户端指点的顶点根据相关参数进行装配,得到所需的一系列图形,涉及点、线、三角形三大基本形态,以及连续、离散两大类共九种图元类型. 本文描述的图元装配单元是与 OpenGL 相兼容的. 根据图元装配单元的功能,其行为模型建模流程如图 1 所示.

使用 SystemC 搭建电路行为级模型,对于后续 RTL 设计可以提供准确无误的标准参考模型 (Golden Reerence Model, GRM). 通过对模型仿真验证可以消除算法或具体电路实现中可能会出现 的错误,提高设计质量和速度. 本文设计的图元装配组件支持图元类包括 Points、Lines、Line_strip、Line_loop、Triangles、Triangles_strip、Triangles_fan 等.

通过对模型的验证得到结论是:该算法满足设计需求,并能高效地进行图元装配.

3 硬件实现

3.1 总体架构

本文所设计的图元装配是指通过算法对客户端指定的顶点数据进行装配,得到点、线、三角形等组成的各种图元. 经过变换的顶点数据按照客户端给定的参数需求发送到图元装配组件,在图元装配阶段把顶点数据根据指定的参数装配成一系列的点、线段和三角形等几何图形.

本文设计的图元装配器由三部分组成:预装配器、装配控制器以及图元装配存储管理;总体结构图

如图 2 所示.

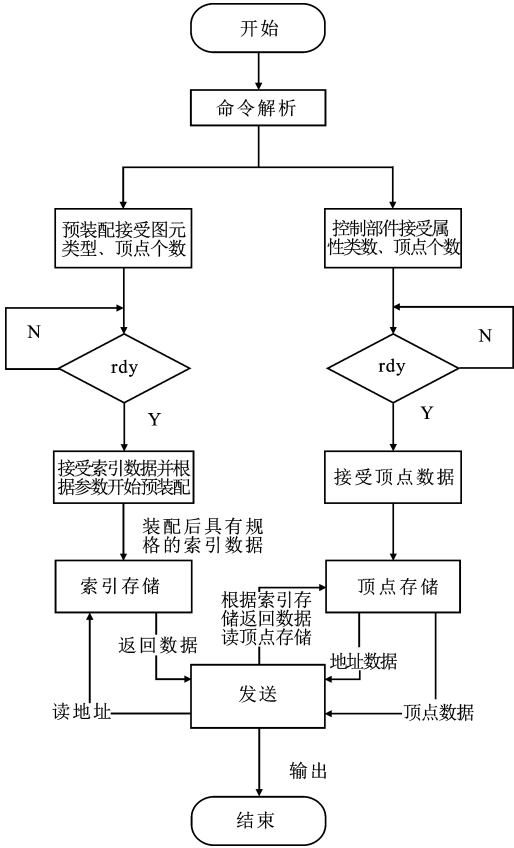


图 1 图元装配模型流程图

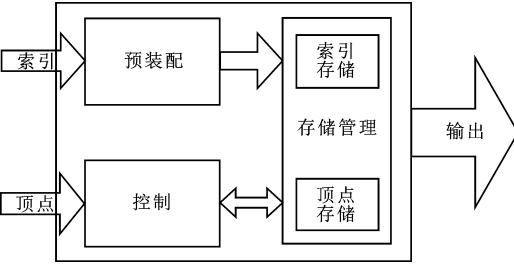


图 2 总体结构

3.2 预装配单元

预装配单元对图元进行预装配处理,其从外部模块中接受顶点索引数据,以及顶点个数、图元个数、装配类型等配置信息数据,对配置信息进行初步处理传送至控制模块;同时根据图元信息对顶点索引数据进行装配处理. 预装配处理后得到的索引数据存放在图元索引存储中,所以预装配单元还需控制图元索引存储. 经过预装配存放在图元索引存储中的索引数据是具有一定规则格式的数据布局. 在表 1 中举例列出了不同图元类型的索引数据的布局图.

表 1 不同图元类型索引数据的布局图						
图元类型	索引数值					
POINTS	0	1	2	3	4	5
LINES	0	1	2	3	4	5
LINES_STRIP	0	1	1	2	2	3
LINES_LOOP	0	1	1	2	2	0
TRIANGLES	0	1	2	3	4	5
TRIANGLES_STIP	0	1	2	1	2	3
TRIANGLES_FAN	0	1	2	0	2	3

3.3 装配控制器

装配控制器是图元装配组件核心控制模块. 装配控制器主要功能是根据所接收到的顶点数据, 完成对顶点数据的存储控制. 接收来自预装配模块的配置信息, 如顶点个数、装配类型等. 等待 rdy 信号开始装配, 从图元索引中经过预装配的图元索引数据, 返回本模块. 通过先前配置信息以及返回的索引数据情况输出对顶点数据存储的相关控制信号.

经过着色的每个顶点, 将根据其顶点索引数据号进行顺序编制, 上层模块接受到顶点数据后, 通过Burst 方式将数据发送给装配器. 不同应用中的每个顶点, 其所对应空间大小各不一致, 下面给出一个实例: 顶点属性启用 Color/Texcoord/Position, 共 3 个顶点, 其对应的地址空间数据如表 2 所示.

在顶点进行发送数据之前, 客户端将配置每个顶点所启用的属性至装配器中. 装配器接收到连续数据后, 将根据所启用的顶点数据属性信息对数据进行解析处理. 如表 2 所述, 装配器接到 3 个数据后, 就知道当前数据位顶点坐标, 即第一个顶点的数据传送完成.

表 2 地址空间数据									
属性	C0	T0	V0	C1	T1	V1	C2	T2	V2
地址	0	1	2	3	4	5	6	7	8

对装配控制器的设计主要使用状态机完成, 实现该状态机有 10 个状态, 具体如下所示.

IDLE: 空闲状态, 若 start 信号位低则采集顶点数据, 否则读取图元索引存储, 跳转至 INDE_RD_T;

INDEX_RD_T: 等待索引存储数据返回; 跳转至 INDEX_RD;

INDEX_RD: 等待索引存储数据返回; 跳转至 VD_RD;

VD_RD: 对索引存储返回数据处理, 跳转至 VD_RD_0;

VD_RD_0: 输出顶点存储读地址, 跳转至 VD_RD_1;

VD_RD_1: 输出顶点存储读地址, 跳转至 VD_RD_2;

VD_RD_2: 输出顶点存储读地址, 跳转至 VD_RD_3;

VD_RD_3: 输出顶点存储读地址, 当检测到下一次装配准备信号为高且本次装配结束信号为高, 则跳转至 IDLE; 否则若本次装配结束信号为高, 跳转至 IDLE_WAIT, 否则若下一次装配准备信号为高, 跳转至 IDLE_RD;

IDLE_WAIT: 当检测到下一次装配准备信号为高, 则跳转至 IDLE; 否则则等待(连续单次装配模式);

IDLE_RD: 当检测到下一次装配准备信号为高, 则跳转至 IDLE; 否则则等待(单次连续装配模式);

3.4 图元装配存储管理

图元装配存储管理由预装配顶点索引存储、顶点数据存储以及存储控制组成. 该模块总体结构如图 3 所示.

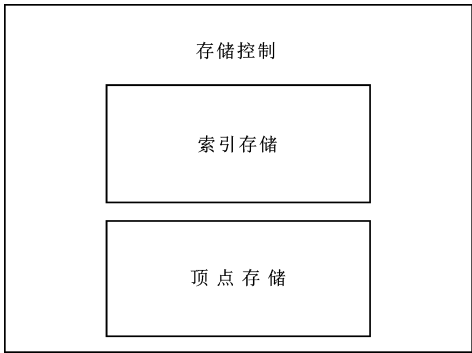


图 3 图元装配存储管理结构图

其中 IMem 存放预装配器所装配的顶点索引数据, DMem 存放经过染色后的顶点数据. 从外部传送到图元装配器的数据, 将存放在顶点数据存储中, 由于每个顶点的属性数量根据应用的不同, 数量也不尽相同. 在目前设计的芯片中, 支持定点的最大属性值位 16, 最小属性为 2, 顶点数据存储大小为 4 k; 因此, 图元装配器中, 对于最大属性数量的顶点一次最多存放 64 定点; 对于最小属性数量的顶点, 一次最多可存放 512 个顶点. 当顶点数据存储存满或者外部引入启动信号, 图元装配器将从预装配器中所得到的索引存储中以地址递增的顺序提取索引数据, 以后根据索引号以及顶点属性数量, 计算当前所

访问顶点的具体地址,最终根据计算所得地址,从数据存储中提取顶点数据,并传送给外部模块。

对于整个图元装配过程,总共存在四次存储访问:

(1)预装配阶段,预装配器接收顶点索引值,并将装配后的顶点索引值存放于 IMem 中。

(2)经过顶点染色后的顶点,将结果存放于 DMem 中。

(3)当顶点数据完成存放后,装配控制器将从 IMem 中按照顺序读取顶点索引值。

(4)根据上一步骤所读取出来的定点索引值,计算对应的顶点数据,最终从 DMem 中读取顶点数据,发送给下一模块。

4 仿真验证

至此以上完成的是图元装配模块的行为建模和硬件电路设计,下面将使用 SystemVerilog 验证语言搭建可重用自动化验证平台。验证的主要目的就是检查设计是否满足设计需求,是否实现了设计所需的功能。具体就是在设计需求规定的条件下,检验电路是否产生了符合预计功能要求的输出^[5]。在设计调试前期主要使用定向测试:根据具体需要和要求,编写对应的测试激励,定点测试可以提前预测结果,有助于前期电路调试;在设计后期主要使用随机测试:通过对输入数据采用带约束的随机化产生,与硬件电路相连接,对电路进行更加全面和完善的测试,以保证硬件电路的正确性。

4.1 仿真结果

根据设计要求和性能需求编写对应的测试激励,在仿真工具 Modelsim 下进行定点测试。在 Modelsim 中以 Triangles_strip 图元类型为例子说明,结果如图 4 所示,可以看到其结果正确。经过充分测试,硬件电路满足图元装配所需的功能且满足时序要求。

4.2 SystemVerilog 验证平台架构

本文设计的图元装配器件采用 SystemVerilog 语言搭建功能验证平台,验证平台架构如图 5 所示,其中参考模型为 SystemC 行为模型。

SystemVerilog 语言提供了高层抽象的接口 (Interface),接口独立于功能模块,包含了连接、同步等多模块之间的通信。本文使用接口连接设计电路与测试平台。本文所使用的验证平台简介模块功能如下。

Packet:声明数据变量并设置约束条件等。



图 4 硬件电路仿真结果

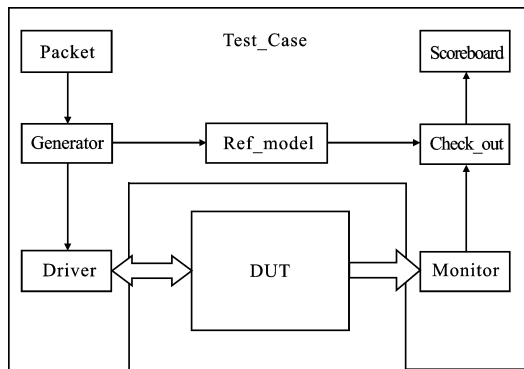


图 5 验证平台架构

Generator:根据已设置的约束,随机化 Packet 中的数据并通过邮箱发送给 Driver。

Driver:接收 Generator 随机化的数据包,通过接口对 DUT 发送数据。

Ref_model:参考模型,接收 Generator 数据,模拟 DUT 处理数据。

DUT:待测试设计(图元装配硬件电路)。

Monitor:通过接口采集 DUT 输出结果,并发送到比较器。

Check_out:接收参考模型与监视器的数据包进行比较处理,结果发送到记分板。

Scorboard:统计检验器的比较结果。

因为数据位数为 32 位,故随机化功能验证没有对数据覆盖率采集。对其图元类型、属性数量、顶点个数等进行随机化。

5 结束语

本文完成了所讨论的图元装配算法的硬件实现以及验证。同时采用 SystemVerilog 语言搭建验证平台,对所设计的硬件电路进行了全面的功能验证。

(下转第 87 页)

度,且图像的景深信息有利于判断显著性目标的位置.这也是下一步努力的方向.

参考文献:

- [1] Itti L, Koch C, Niebur E. A model of saliency-based visual attention for rapid scene analysis[J]. IEEE Transactions on pattern analysis and machine intelligence, 1998, 20(11): 1254-1259.
- [2] Ma Q, Zhang L. Saliency-based image quality assessment criterion[C]//International Conference on Intelligent Computing. Springer Berlin Heidelberg, 2008: 1124-1133.
- [3] Achanta R, Hemami S, Estrada F, et al. Frequency-tuned salient region detection[C]//Computer vision and pattern recognition, 2009. cvpr 2009. iee conference on. IEEE, 2009: 1597-1604.
- [4] Luo Y, Yuan J, Xue P, et al. Saliency density maximization for efficient visual objects discovery[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2011, 21(12): 1822-1834.
- [5] Cheng M M, Mitra N J, Huang X, et al. Global contrast based salient region detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 569-582.
- [6] Borji A, Cheng M M, Jiang H, et al. Salient object detection: A benchmark[J]. IEEE Transactions on Image Processing, 2015, 24(12): 5706-5722.
- [7] Hunter R S. Photoelectric color difference meter[J].

Josa, 1958, 48(12): 985-995.

- [8] Hunter R S. Accuracy, precision, and stability of new photoelectric color-difference meter[J]. Journal of the Optical Society of America, 1948, 38(12): 1094.
- [9] Liu T, Yuan Z, Sun J, et al. Learning to detect a salient object[J]. IEEE Transactions on Pattern analysis and machine intelligence, 2011, 33(2): 353-367.
- [10] 黄虹, 张建秋. 彩色自然场景统计显著图模型[J]. 复旦学报(自然科学版), 2014, 53(1): 51-58.
- [11] Bundy, Alan, Lincoln Wallen. Breadth-First Search[M]. Springer Berlin Heidelberg, 1984:13.
- [12] Zhai Y, Shah M. Visual attention detection in video sequences using spatiotemporal cues[C]//Proceedings of the 14th ACM international conference on Multimedia. ACM, 2006: 815-824.
- [13] Imamoglu N, Lin W, Fang Y. A saliency detection model using low-level features based on wavelet transform[J]. IEEE transactions on multimedia, 2013, 15(1): 96-105.
- [14] Bai P, Cui Z, Gan Z, et al. A novel saliency detection model based on curvelet transform[C]//Wireless Communications & Signal Processing (WCSP), 2016 8th International Conference on. IEEE, 2016: 1-5.

作者简介:

刘臣辰 女,(1992-),研究生.研究方向为图像处理.

E-mail:14210720037@fudan.edu.cn.

张建秋 男,(1962-),教授.研究方向为阵列信号处理.

(上接第 82 页)

综上所述,本文讨论的图元装配组件能够很好地实现对图元快速装配的功能并且满足设计要求,在很大程度上提高了处理器的速率.

参考文献:

- [1] 吴恩华. 图形处理器用于通用计算的技术、现状及其挑战[J]. 软件学报, 2004, 15(10):1493-1504.
- [2] 邓军勇, 李涛, 蒋林,等. 面向 OpenGL 的图形加速器设计与实现[J]. 西安电子科技大学学报(自然科学版), 2015, 42(6):124-130.
- [3] 王丹, 韩俊刚, 杨世旺,等. 图元装配的硬件设计与验证[C] //中国计算机图形学大会. 中国计算机学

会, 2012.

- [4] 罗琨, 曹阳, 尹建华,等. 数字专用集成电路设计中的 SystemC 建模验证方法[J]. 武汉大学学报(理学版), 2002, 48(3):306-310.
- [5] 韩伟. 图形处理芯片的研究与设计验证[D]. 上海:上海交通大学, 2008.

作者简介:

周高财 男,(1992-),硕士研究生.研究方向为电路与系统.

E-mail:374661708@qq.com.

李涛 男,(1954-),教授.研究方向为计算机体系结构、计算机图形学.

贺琼 女,(1992-),硕士研究生.研究方向为电路与系统.