

图形处理器中双核顶点染色器的设计与实现

史少旭¹, 韩俊刚², 刘 欢²

(1 西安邮电大学 电子工程学院, 陕西 西安 710121; 2 西安邮电大学 计算机学院, 陕西 西安 710121)

摘 要: 为了提高图形处理器(graphic processing unit, GPU)中顶点染色器的工作效率, 设计了一款改进的可编程双核顶点染色器。它可以同时对来自命令处理器的两条命令进行解析, 然后交由 PE 双核进行处理, 这样既可以提高处理器的资源利用率又可以加强处理器的并行执行能力。采用基于 FPGA 的验证方式, 整个双核顶点染色器占用 Xilinx V6 系列 550T 型号 FPGA 芯片硬件资源的 7%, 最大工作频率能够达到 155.411 MHz。

关键词: 图形处理器; 顶点染色器; 可编程

中图分类号: TP302

文献标识码: A

文章编号: 1000—7180(2017)02—0077—06

Graphics Processor in the Design and Implementation of a Dual Vertex Shader

SHI Shao-xu¹, HAN Jun-gang², LIU Huan²

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

2 School of Computer, Xi'an University of Posts & Telecommunications, Xi'an 710121, China)

Abstract: In order to improve the graphics processing unit (graphic processing unit, GPU) vertex shader in the efficiency of work designed a dual-core improved programmable vertex shader. It could also resolve two commands from the command processor and then referred to PE dual-core processing, this way you can improve processor utilization can strengthen the parallel processor implementation capacity. Using FPGA-based authentication methods, the entire vertex shader use 7% Xilinx V6 series 550T FPGA chip hardware resources, the maximum operating frequency can reach 155.411 MHz.

Key words: GPU; Vertex Shader; programmable

1 引言

图形处理器(GPU)的发展主要集中在 3D 图形渲染领域, 如 PowerVR、Nvidia、ATI 和 BitBoys^[1]等产品中都具有 3D 引擎所必备的高性能染色处理单元。顶点染色器作为 GPU 的一个重要组成, 是高性能染色处理单元不可缺少的部分, 主要完成顶点、纹理坐标、光栅位置、法向量、源位置和聚光灯方向向量的旋转、平移和缩放操作, 以及顶点坐标的光照计算^[2]。

目前已经提出的顶点染色器的结构主要有: 单指令多数据流(Single Instruction Multiple Data,

SIMD)、超长指令字(Very Long Instruction Word, VLIW)、VLIW + SIMD、VLIW + SIMD + 多线程(Multithreading)^[3-4]。SIMD 主要提高了向量的处理速度, 提高了矩阵的运算, 对于顶点染色器的性能提高不是非常大; VLIW 通过将多个操作包含于一条指令的方法来提高处理器性能, 虽然在一定程度上提高了整体性能, 但是编译器编译时比较困难; VLIW + SIMD 虽然结合了上述两种处理器的优点, 但处理器的设计太复杂。VLIW + SIMD + 多线程在 VLIW + SIMD 基础上加入了 SMT 技术, 充分利用了空闲资源, 实现了数据处理的并行化, 但硬件复杂度相对较高。

收稿日期: 2016—04—29; **修回日期:** 2016—06—18

基金项目: 国家自然科学基金重点资助项目(61136002); 教育部科学研究计划重点资助项目(2111180)

本文通过分析顶点染色器的功能,设计了支持 OpenGL 命令处理器的可编程双核顶点染色器.与前面介绍的几种处理器相比较,本文中的处理器具有两个处理器内核,可以同时处理来自命令处理器的两条命令进行处理,两个内核共用一套指令和数据 Cache,提高了资源利用率,而且两个内核之间的工作是相互配合,提高了处理器的工作效率.

2 双核顶点染色器的整体结构

本文提出的双核顶点染色器由命令处理单元 (CMD)、2 个处理单元 (Processing Element, PE)、一个仲裁器 (Arbitration)、1 个 Dcache、1 个 Icache 和输出模块组成.整体结构如图 1 所示.

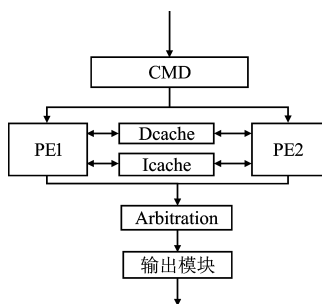


图 1 双核顶点染色器的整体结构

双核顶点染色器完成的功能描述如下:

- (1) 完成顶点、纹理坐标、光栅位置、法向量光源位置和聚光灯方向向量的旋转、平移和缩放操作;
- (2) 完成顶点坐标的光照计算,最多支持 8 盏灯的光照计算,同时支持单面和双面光照计算^[5];
- (3) 完成模型视图矩阵和纹理矩阵的基本操作;
- (4) 支持深度为 32 的模型视图矩阵堆栈,支持深度为 16 的纹理矩阵堆栈,支持深度为 8 的属性堆栈;
- (5) 完成法向量标准化计算;
- (6) 完成当前模型视图矩阵的逆矩阵变换操作;
- (7) 提供用户查询相关属性的信息.

3 双核顶点染色器的设计

双核顶点染色器最大的优点在于通过设计双核来减轻单核处理器中遇到的处理速度慢,并行计算能力差等问题.对于命令处理器解析到的命令,可以交由两个 PE 核分别处理,这样就节省了命令等待时间,而仲裁器的作用是会让它们的输出出现紊乱^[6].本文主要就双核的结构以及核内具体结构作出详细介绍.

3.1 E 核的整体结构

本文用到的 PE 核主要由 VS_CMD、VS、VS_IM、VS_MM 四部分组成.各模块的连接关系如图 2 所示. VS_CMD 主要用于解析传到 PE 核的命令,提取命令中的参数,输出每个固定程序的入口地址. VS 用于运行基本的函数和着色程序. VS_IM 分为两部分,一部分为固定程序存储区,用于存放固定染色程序;另一部分为 Icache 存储区,用于存放用户自定义染色程序(实际上是从 DDR2 中读入 Icache 存储区). VS_MM 用于管理 VS 所涉及的数据存储区,包括 ram 存储区、统一编址的寄存器以及 Dcache.

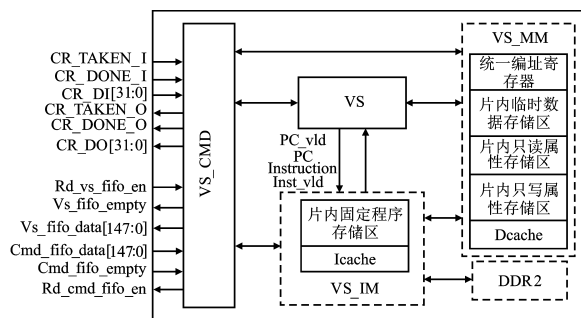


图 2 PE 核的整体结构

3.2 VS_CMD 模块的设计与实现

VS_CMD 模块主要由 CMD_PROCESS、FIFO 和 FEEDBACK 三个模块组成.它们之间的连接关系如图 3 所示.

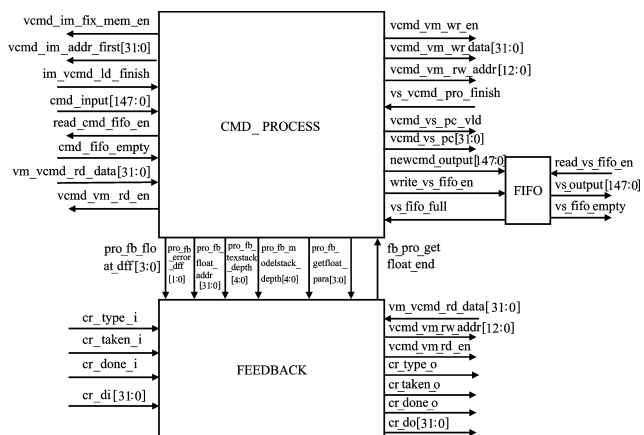


图 3 VS_CMD 模块顶层结构图

3.2.1 CMD_PROCESS 模块完成的功能

CMD_PROCESS 模块控制命令中参数的存储与读取,并根据不同命令的要求对命令进行直接透传、只存储参数并下传、修改参数并下传以及启动顶点染色程序产生新命令并下传等操作;本模块对命令译码,将不同命令送入其相应的命令处理状态;其

处理步骤如下所示.

- (1)接收来自命令处理器的 OpenGL 命令;
- (2)拼接地址,提取数据;
- (3)产生 read_cmd_fifo_en 信号;
- (4)确定处理本条命令是否透传、是否对系统进行初始化、是否修改参数、是否产生新命令,发出相关的控制信号;
- (5)给出相应待启动程序的 pc 值(程序入口地址)并置高 pc_vld 信号,表示当前 pc 有效;
- (6)存储相应的命令行参数;
- (7)产生新命令并下发.

3.2.2 Feedback 模块完成的功能

- (1)反馈环单元用于处理 glGetFloatv 命令,根据命令要求把流水线对应各级的状态信息返回客户端;
- (2)反馈环传递堆栈上溢和下溢造成的错误;
- (3)反馈环单元由位于命令处理器的处理电路和位于流水线各级的传输电路共同组成;
- (4)反馈环与顶点染色器命令处理器并行执行,顶点染色器命令处理器接收到 glGetFloatv 命令后两者才会相关;
- (5)命令处理器优先级最高,一旦命令处理器向反馈环发送数据,glGetFloatv 命令处理过程必须等待直到反馈环为空;
- (6)反馈环将 glGetFloatv 的反馈信息与堆栈上溢和下溢产生的错误反馈信息按照不同的信息编码上传.

3.2.3 FIFO 模块完成的功能

- (1)接收从 cmd_process 模块获得下发的命令;
- (2)发送命令到图元装配单元;
- (3)空满信号指定 fifo 是否可读写.

3.3 VS_MM 模块的设计与实现

VS_MM 模块主要完成了对参数存储区完全管理和数据 Cache 初步管理,这些管理操作包括了顶点染色命令处理单元和顶点染色处理器对参数存储区的读写控制操作,以及顶点染色指令存储控制器对数据 Cache 的管理.

顶点染色存储管理顶层电路结构以及地址映射如图 4 所示,顶点染色器的存储区主要分为三类:

- (1)统一编址的寄存器,这类主要存储所有的使能状态信号,顶点染色命令处理模块和顶点染色处理器模块可以对它进行读写操作;
- (2)片内 ram 区,这类主要存储与 OpenGL 命令相关的参数,大部分存储区域是由顶点染色命令处理模块进行写入操作,由

顶点染色处理器模块进行读出操作;

- (3)数据 cache 区,这是一个简单的数据 cache,不进行回写操作,这部分主要是顶点染色处理器模块进行读出操作,写操作是在不命中的情况下,从 DDR 中,读取 1 kbyte 数据,将 cache 替换,数据 cache 的总大小为 1 kbyte.

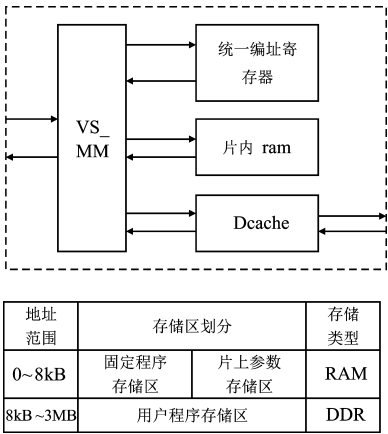


图 4 顶点染色存储管理顶层结构和地址映射图

顶点染色存储管理详细电路图如下图所示. 图 5 所示为写 ram 操作电路,对 ram 的写操作是由顶点染色命令处理模块和顶点染色处理器模块进行的,这两者对 ram 的写操作是互斥的,即要么命令处理模块进行写,要么染色处理器模块进行写,不可能同时进行,因此用写使能信号进行选择即可. 由于染色处理器模块访问的是全空间,因此,需要对其地址进行初次判断,判断是否对 ram 进行操作,ram 区的地址范围是 000H~FFFH,只需判断高 19 位地址是否为 0 即可. 选择的结果还需要判断是否是对统一编址的寄存器进行写操作,这个判断只需将选择的地址同 08AH(统一编址的寄存器的地址)进行比较,不相等,则访问的是 ram 区,相等则访问的是统一编址的寄存器.

图 6 所示的电路是对统一编址寄存器进行写操作. 这个寄存器主要存储的是相关的使能信号,这个使能信号分配如图 6 中的表格所示.

该寄存器是由命令处理模块进行写操作,输入的 32 数据中,如果是对某一位进行操作,就将这位位置高,其余位置低. 如果要置位这位,则将第 0 位置高,第 1 位置低;如果要清除这位,则将第 0 位置低,第 1 位置高;如果要对矩阵模式进行操作,则将第 1 位置高. 电路实现上,如果要置位,则把原数据同写入的数据进行或操作,再写入寄存器中即可;如果要清零则把写入的数据取反,再同源数据进行与操作,

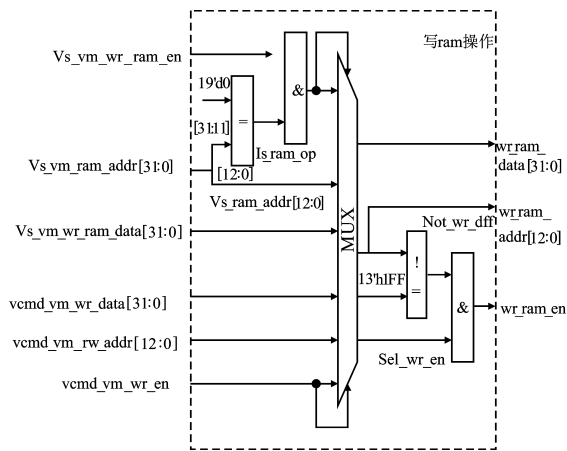


图 5 写 ram 操作电路图

操作过程类似,可以参见写 ram 操作电路说明.

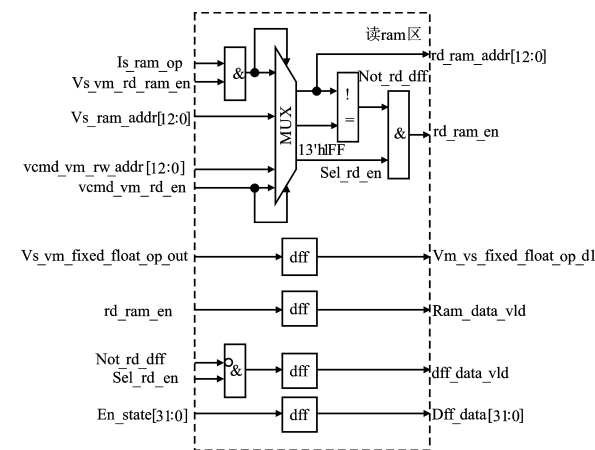


图 8 读 ram 操作电路图

图 9 是读操作数据选择电路,读取的数据来自三部分,一是来自数据 cache;二是来自 ram 区;三是来自统一编址的寄存器.如果是数据 cache 的话,其读取数据的时钟数是不确定的;如果是 ram 区的话,只需要一个节拍;如果是统一编址的寄存器的话,则可以立即输出.由于这三部分都是互斥的,因此,只需要用各自的数据有效信号进行选择即可.

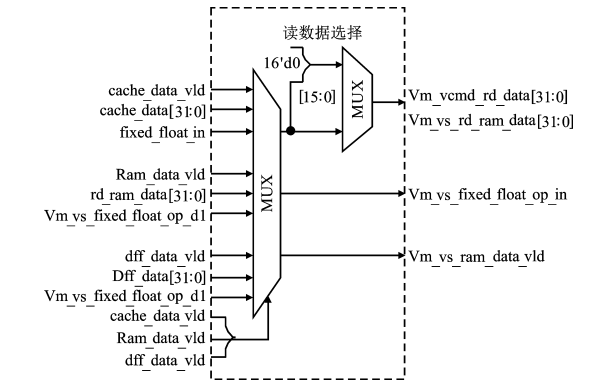


图 9 读操作数据选择电路图

统一编址寄存器各位表示的参数意义 (地址 01FFH)								
31	30~23	22	21	20	11~6	5~2	1	0
lighting	八盏灯的使能信号	颜色材料	使能归一化法向量	使能缩放法向量	保留	Matrix mode	写矩阵模式	置位/清零

32	16	8	4
5	4	3	2
1~0			
颜色	纹理	投影矩阵	模型视图

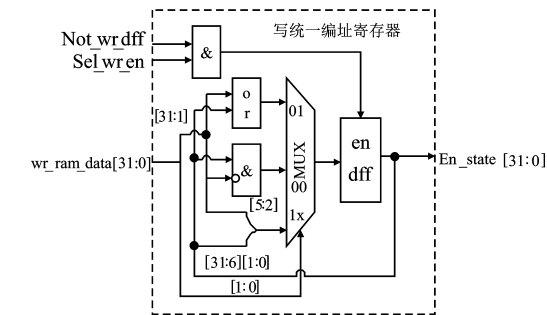


图 6 统一编址寄存器

结果写入寄存器中;如果是对矩阵模式进行操作,则直接将写入的数据的第 5 到第 2 位数据写入到寄存器的第 5 到第 2 位.

图 7 是读数据 cache 进行读操作,只有染色处理器才能对 cache 进行读操作,只需判断染色处理器访问的地址不在 ram 区即可.

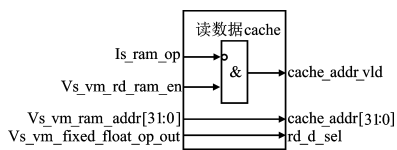


图 7 读数据 cache 电路图

图 8 是读 ram 操作电路,其操作过程同写 ram

3.4 VS 模块的设计与实现

3.4.1 VS 模块完成的功能

(1)染色处理单元包含一个浮点单元和一个定点运算单元;浮点单元支持单精度运算,定点运算单元支持 32 位算术和逻辑运算;

(2)支持数据 Cache (Dcache) 和指令 Cache (ICache),大小为 1 kByte.

(3)支持浮点和定点指令发射,定点和浮点单元有各自的流水线.定点单元支持五级流水线:

IFetch/Decode/Exe/Memory/Write_Back 等五级;

(4)指令顺序发送,除了 load 和浮点除法命令

外,在执行期间支持乱序;

(5)采用 Load-Store 结构.

3.4.2 代码实现逻辑

VS 模块的整体结构如图 10 所示,该部分电路与外部的 vs_cmd 相接,在 vs_vcmd_pro_finish 为高时,vs_cmd 向 pc 模块输入 vcmd_vs_pc_vld 和 vcmd_vs_pc[31:0],pc 模块指示 icache 输送对应指令给 fifo,并进行译码等后续相关操作.

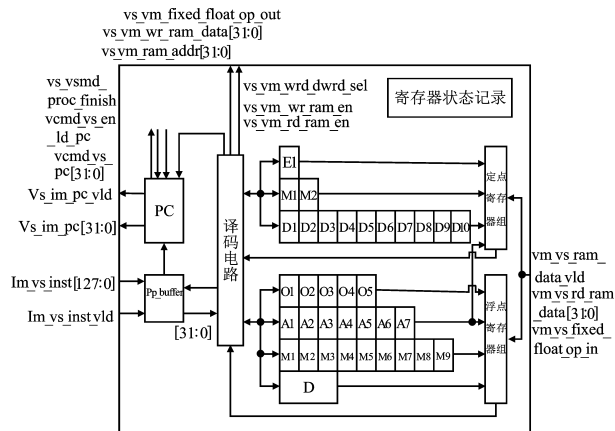


图 10 VS 模块的整体结构

3.5 VS_IM 模块的设计与实现

3.5.1 VS_IM 模块完成的功能

VS_IM 完成顶点染色器中固定程序的加载以及读取,完成指令 Icache 的实现,完成 cache 控制器的实现,能够减少与内存交换的速度差异,从而提高性能. VS_IM 能够通过使能信号从 ddr 中加载固定程序块,能够从固定程序区或者 Icache 中读取相应的指令,能够从 ddr 中加载 Icache 块,能够完成数据 Dcache 和指令 Icache 的控制,能够完成数据和指令的整块调入.

3.5.2 代码实现逻辑

按照功能描述可以将 VS_IM 划分为 Icache 与固定存储体块 (Icache_fix)、cache 控制器 (cache_ctrl) 两个部分,结构框图如图 11 所示.

根据功能描述可以将 Icache_fix 模块划分为 tag 控制模块、地址映射块、译码控制块、存储体块、输出控制块 5 个模块.

Tag 控制模块:完成加载固定程序的判断,完成所要取指令的地址判断,判断出读取固定程序区还是 cache 区,如果是固定程序区,直接访问固定程序去;如果是 cache 区需要判断 cache 是否命中,若命中,则直接访问 cache ram 读取指令,若不命中,则需要送入地址映射模块.

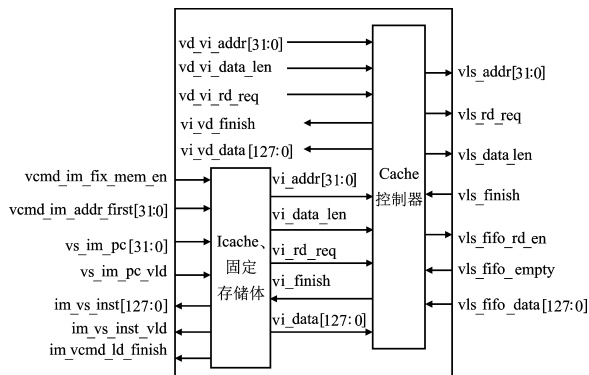


图 11 VS_IM 结构框图

地址映射模块:完成加载固定程序首地址和 ddr2 地址的映射,完成替换 cache 块地址的映射,并且根据是能选择所要读取 ddr2 的模块.

译码控制模块:完成从 ddr2 读取的数据写入选择是固定存储区还是 cache 存储体区,完成存储体部分地址的产生,并且判断数据是否已写满.

存储体模块:完成 1 kB 大小的 Icache 的指令存储,完成 8 kB 大小的固定程序的存储.

输出控制块:完成存储体中输出 128 位指令(四条),完成固定存储区和 cache 存储体取数据的选择.

4 验证与结果分析

4.1 测试平台介绍

如果采用软件仿真的方法进行测试,不仅效率低而且花费的时间也多,更别说是完全测试了.因此,本文采用软硬件结合的方法进行测试,不仅测试的完备性、正确性大大提高,而且效率也提高不少.

FPGA 测试的结构图如图 12 所示,主要由三部分构成:第一部分是上位机部分,主要是软件编程的工作,完成将 OpenGL 库的实现、驱动程序的实现.编译器将测试程序编译成可执行的文件,通过 JTAG 接口或者 URAT 接口加载到 LEON 的 DDR 中;第二部分是 CPU 部分,该部分采用开源的 LEON3 处理器,主要完成测试程序的执行,以及 CPU 的驱动;第三部分就是 GPU 部分,该部分就是对 OpenGL 程序进行加速的电路实现部分^[7-8].

FPGA 测试版采用的是 Dini 公司的 DNV6_F2PCIE 测试板,该板提供了两颗 Xilinx 公司的 V6 系列的 550T 型号的 FPGA.其中 A 片放的是 CPU 电路,B 片放的是 GPU 电路,两片的通信采用了异步握手加滤波的方式,保证了两片 FPGA 通信的正确性.CPU 只占 A 片 LUT 的 20%,单核能跑 150

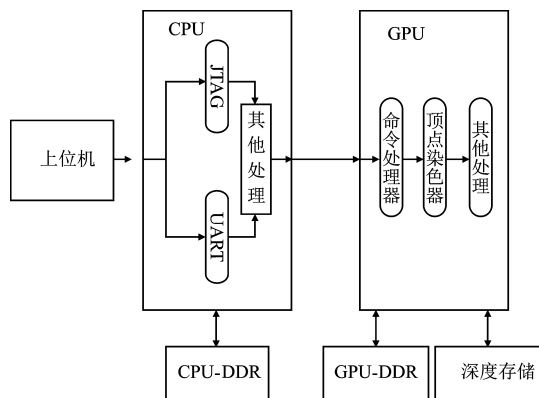


图 12 FPGA 测试结构图

MHz, 多核能跑 100 MHz. GPU 只占 B 片 LUT 的 54%, 综合报告电路速度能跑到 100 MHz, 实际测试电路能跑 150 MHz.

4.2 验证结果分析

综合测试结果如图 13 所示, 这些测试例子都需要进行光照计算和坐标变换, 包含了各种顶点数据以及各种变换的运算, 顶点数据多达上万, 充分地测试了双核顶点染色器的各种功能和稳定性. 结果表明, 处理器在处理速度上比单核处理器处理速度提高了 40%, 而且稳定性也比单核的强.

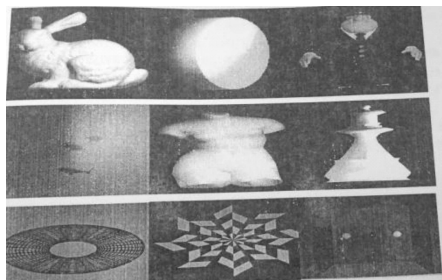


图 13 综合测试结果

5 结束语

本文提出了一种双核顶点染色器的结构, 在 FPGA 板子上完成了测试, 并分析了结果. 该双核顶点染色器与之前的相比, 不仅在处理速度上提高了, 而且稳定性也得到了保障, 同时有效地提高处理器的资源利用率并加强处理器的并行执行能力.

参考文献:

- [1] 韩俊刚, 蒋林, 杜慧敏, 等. 一种图形加速器和着色器的体系结构[J]. 计算机辅助设计与图形学学报, 2010, 22(3): 363-372.
- [2] 胡伟武, 陈云霄, 肖俊华, 等. 计算机体系结构[M]. 北京: 清华大学出版社, 2011: 55-139.
- [3] 周江磊, 李涛, 韩俊刚. 图形处理器中顶点染色仿真器的设计[J]. 数字技术与应用, 2012, 30(12): 109-110.
- [4] 杜慧敏, 李萌. 面向顶点染色算法的专用指令集优化处理器[J]. 西安邮电大学学报, 2014, 19(2): 60-66.
- [5] 范彩霞, 吴宁. 图形处理器光照引擎的设计[J]. 计算机工程应用, 2009, 45(7): 86-88.
- [6] 韩俊刚, 刘友耀, 张晓, 等. 嵌入式图形处理器中几何变换引擎的设计[J]. 计算机仿真, 2010, 27(1): 206-209.
- [7] Luebke D, Humphreys G. How GPUs work[J]. IEEE Computer, 2007, 40(2): 96-100.
- [8] George Lai Yuan. GPU compute memory system[D]. Vancouver: The University of British Columbia, 2006.

作者简介:

史少旭 男, (1992-), 硕士研究生. 研究方向为电路与系统. E-mail: 120868871@qq.com.

刘欢 男, (1992-), 硕士研究生. 研究方向为计算机体系结构.

韩俊刚 男, (1943-), 教授. 研究方向为软件和硬件的形式化验证、图形处理器和新型计算机体系结构.

(上接第 76 页)

- [2] Sohi G S, Breach S E, Vijaykumar T N. Multiscalar processors[C]//Proceedings of the 22nd Annual International Symposium on Computer Architecture. ACM, 1995, 23(2): 414-425.
- [3] 岳峰, 利润丰, 陈田, 等. 基于随机路由的高性能片上路由器设计与仿真[J]. 电子测量与仪器学报, 2013, 27(7): 699-675.
- [4] 周东红. 基于 3x3NoC 的路由器设计与网络性能分析[D]. 西安: 西安电子科技大学, 2011.
- [5] 林世俊, 张凡, 金德鹏. 分布式同步的 GALS 片上网络

及其接口设计[J]. 清华大学学报, 2008, 48(1): 32-38.

- [6] 杨婷. 多态并行阵列处理器中的数据通信及路由器的设计与实现[D]. 西安: 西安邮电大学, 2011.

作者简介:

杜丽娜 女, (1989-), 硕士研究生. 研究方向为数字集成电路. E-mail: 1403224191@qq.com.

韩俊刚 男, (1943-), 博士, 教授. 研究方向为集成电路系统设计和验证、计算机体系结构.

李卯良 男, (1991-), 硕士研究生. 研究方向为数字集成电路.