

# 一种图形处理用的多格式定点运算器

易学渊<sup>1</sup>, 李涛<sup>2</sup>, 蒲林<sup>2</sup>

(1. 西安邮电大学 计算机学院, 陕西 西安 710061;

2. 西安邮电大学 电子工程学院, 陕西 西安 710061)

**摘要:**典型图形处理管线中的隐藏面消除、视窗变换、图元光栅化和像素着色等渲染阶段只需要定点运算,但是这些定点运算的数值范围需要变动。因此,设计了一种多格式定点处理单元来满足这些运算需求。设计中采用了指令双发射和多配置流水线等超标量处理机制,访存采用直接寻址,不含通用寄存器和LD/ST操作。值得一提的是,指令字中设置了阻塞标志位,用于控制互连通信模式,数据字中设置了格式标志位,用于控制多格式定点数据的处理。处理器实现了指令级并行、数据级并行等并行处理方式。

**关键词:**图形处理;超标量;阻塞标志位;指令级并行;数据级并行

中图分类号:TP332

文献标识码:A

文章编号:1673-629X(2014)10-0147-04

doi:10.3969/j.issn.1673-629X.2014.10.035

## A Multi-format Fixed Point ALU for Graphics Processing

YI Xue-yuan<sup>1</sup>, LI Tao<sup>2</sup>, PU Lin<sup>2</sup>

(1. School of Computer Science, Xi'an University of Posts and Telecommunications,  
Xi'an 710061, China;

2. School of Electronic Engineering, Xi'an University of Posts and Telecommunications,  
Xi'an 710061, China)

**Abstract:** In typical graphics rendering pipeline, the stages after projection such as back-surface culling, viewport transformation, rasterization and pixel shading, only need fixed point arithmetic operations. But the fixed point arithmetic operation range needs to change in different stage of operation. Hence, a multi-format fixed point Processing Element (PE) for graphics processing is designed. Such mechanisms of superscalar technology as dual-issue instruction word and multi-configuration pipeline are adopted when design, no general registers and LD/ST operations because of direct accessing mode. Particularly, a blocking bit is set in the instruction word for the control of interconnect communication mode, also a few format bits are set in data word for the control of multi-format fixed point data processing. Parallel processing means like Instruction Level Parallel (ILP) and Data Level Parallel (DLP) can be found in design of PE.

**Key words:** graphics processing; superscalar; blocking bit; instruction level parallel; data level parallel

## 0 引言

随着计算机科学技术的发展,图形处理器<sup>[1]</sup>不再仅仅用来完成图形处理相关的计算任务,而是逐渐向通用计算方向发展。大到整个图形处理器,小到其内部的着色器和引擎,它们所完成的功能不再是固定的,而是通过可编程来灵活实现。

图元光栅化是图形处理管线中的重要环节,负责把参数化描述的点、线、三角形等基本图元填充为具体的图形。文中设计了一种多格式定点运算器,能以可

编程的方式完成图元光栅化等渲染环节。与ASIC设计结构相比,可编程方式在灵活性和可控性方面大大加强,从而使得这些复杂渲染阶段的处理更容易维护和改进。另外,在电路面积上也有所降低,改善了ASIC设计存在的运算部件闲置和不可复用的不足。

文中主要介绍光栅化引擎的整体设计以及运算器设计,同时将ASIC设计方式和可编程设计方式做了简单的对比分析,并以此说明可编程设计的综合优越之处。

收稿日期:2013-11-16

修回日期:2014-02-20

网络出版时间:2014-07-17

基金项目:国家自然科学基金重点资助项目(61136002);陕西省科学技术研究发展计划资助项目(2011K06-47)

作者简介:易学渊(1987-),男,硕士研究生,研究方向为计算机系统结构和VLSI、图形处理;李涛,教授,研究方向为计算机图形学、专用集成电路设计与集成系统等。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140717.1226.005.html>

### 1 图形处理管线和光栅化

图形处理器主要用来给图形计算提供硬件加速。管线处理结构是早期的一种较为经典的设计方式,这种结构按照图形处理流程来设计,各个部件具有明确而特定的功能。

在管线结构中,命令处理器接收 CPU 经总线传送的 API 驱动命令,分别下发到 3D 和 2D 渲染引擎。在管线中的相邻两级通过握手协议完成数据传递,每个流水级完成不同的渲染功能,并最终将计算的像素数据送至帧缓存。在整个管线中,像素染色器完成的是图元的光栅化<sup>[2]</sup>和片段的纹理光照计算。

图元光栅化也称扫描转换,主要完成图元的填充计算。将通过顶点坐标描述的理想图元,显示在网格状的显示屏幕上,需要通过扫描算法<sup>[3]</sup>来对顶点以外的空白处做插值计算,这个计算过程就是图元的光栅化。

针对扫描的功能特点,扫描单元可以设计成 ASIC 结构。这种设计方式的优点是数据流动多为定向,无需指令控制,速度较快。但由于电路结构一般是根据特定扫描算法的处理流程来设计,实现方式是固化的,不利于设计的维护和改进。

由于扫描算法多种多样,并且各个算法在效率和渲染效果上各有优缺点,实现方式的灵活性成了一个新的要求。具有可编程能力的扫描单元,刚好能满足这一点。可编程扫描单元的实质是,用指令驱动的运算器来代替 ASIC,对图元的扫描处理通过在运算器上编写扫描程序来实现。

### 2 运算器设计

通过分析隐藏面消除、视窗变换、图元光栅化和像素着色等阶段的渲染算法可以发现,采用合理格式的定点数即能满足运算需求,并且算法中的运算多为加法和乘法,含少量除法和开方等。文中设计的定点运算器,在指令系统、微体系结构和具体实现上,都依据算法的需求来展开。

#### 2.1 指令系统

文中指令系统的设计,采用当前较为流行的 RISC 形式。单条指令字长为 43 位,其中包含阻塞标志位、操作码、源操作数和目的操作数。指令字最高位为阻塞标志位,操作码占 6 位编码,目的操作数占 10 位编码,两个源操作数分别占 10 位编码和 16 位编码。

一般情况下,阻塞标志位置 0,指令流正常执行而不会发生阻塞。当阻塞标志位置 1 时,指令流可能发生阻塞,需要通过判断访存地址是否为特定通信地址来决定。

两个源操作数一般为读访存地址,目的操作数一

般为写访存地址。立即数和 PC 值在 16 位的源操作数中给出。

指令的功能由指令中的操作码指定,分为算术操作、逻辑操作、跳转操作、本地数据移动、数据设置、调用和返回、空操作等几大类。

#### 2.2 微体系结构

图 1 是处理单元的总体设计结构图。处理单元流水线<sup>[4-5]</sup>大致分为指令预取<sup>[6-7]</sup>,译码<sup>[7-8]</sup>,数据发射,执行,回写 5 个阶段。

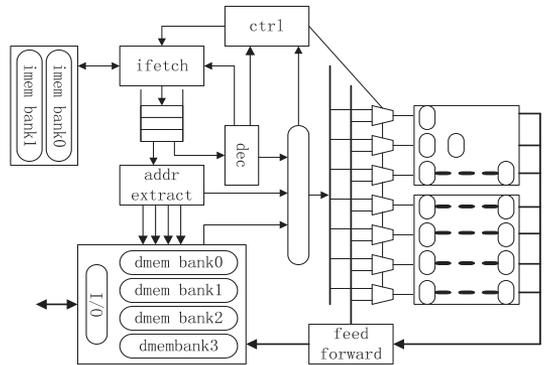


图 1 总体结构图

指令和数据在初始化时通过外部接口分别载入到指令存储和数据存储。指令预取每次读取两条指令字,送给译码单元做译码处理和控制。发射级将来自数据存储,指令字,或者前馈路径的数据分送到不同执行管线。地址流水负责控制数据和地址的匹配,并向存储提交数据。

译码单元每次能同时处理两条指令的译码操作。数据存储包含 4 个 bank,最多可同时读取 4 个操作数,每个 bank 大小为 256 B,总大小为 1 kB。多配置执行管线包括整数加减逻辑运算、整数乘法运算<sup>[9]</sup>、整数除法运算<sup>[10]</sup>、定点加减逻辑运算、定点单目运算(例如取整、取绝对值)、定点乘法运算、定点除法运算共 7 条并行执行管线,不同管线的的数据可以并行处理。

#### 2.3 定点格式

定义的多格式定点数据需要编码前缀来区分,32 bit 数据的高两位被用来作为数据格式编码位,低 30 位是有效数据编码位。数据格式编码见图 2。

	格式编码		数据编码			
	比特位	31:30	29:24	23:15	14:6	6:0
6.24 定点数	00		整数	小		数
15.15 定点数	01		整	数	小	数
24.6 定点数	10		整		数	小数
30.0 整数	11		整			数

图 2 多格式定点定义

对于无符号数和整数,处理时不需要对前缀编码进行判断。对于多格式定点数,处理时则需要对前缀编码进行判断。

当数据结果发生异常时,运算执行单元会给出运算结果异常信号,异常的运算结果不会被提交到数据存储管理单元。

在进行算术运算时,定点运算结果需要进行截位处理。在处理时,根据输入数据携带的格式编码,确定计算结果的定点位置并对结果进行截取和舍入。

截位处理主要针对定点乘除运算。对于定点乘法,由于操作数为 30 位有效,未截位前的积最大为 60 位。若为两个 6.24 定点数相乘,则去掉高 6 位和低 24 位。若为两个 15.15 定点数相乘,则去掉高 15 位和低 15 位。若为两个 24.6 定点数相乘,则去掉高 24 位和低 6 位。若为两个 30.0 定点数相乘,则去掉高 30 位。数据发送上溢时,取该定点格式的最大值,数据发生下溢时,采取“就近舍入”方式。截位处理如图 3 所示。

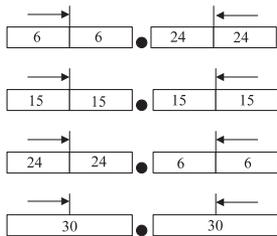


图 3 截位处理

定点除法是一个逐步产生商的过程,能否除尽以及商的位数都不可预知。因此做除法时,除满相应格式位数的商即停止,未除的后续位相当于做了截断,会带来误差。具体做法是,若为两个 6.24 定点数相除,除完被除数 30 位后,取商低 6 位作整数部分,再添 0 除得 24 位小数为止。若为两个 15.15 定点数相除,除完被除数 30 位后,取商低 15 位作整数部分,再添 0 除得 15 位小数为止。若为两个 24.6 定点数相除,除完被除数 30 位后,取商低 24 位作整数部分,再添 0 除得 6 位小数为止。若为两个 30.0 定点数相除,除完被除数 30 位即止。

## 2.4 跳转和函数调用

在执行条件跳转指令时,指令流的执行会进入阻塞状态,并根据 ALU 最终的计算结果来判断是否执行跳转操作。对于无条件跳转和函数调用操作,则直接执行跳转操作。其中,函数调用一般由 CALL 和 RET 指令配对完成,由 CALL 指定调用函数 PC 引起跳转,再由 RET 返回继续执行。

对于数据冒险和访存冲突等问题,统一交给软件层来处理。这样,硬件层不需要提供复杂的控制机制,设计也相对简单。

## 2.5 互连和阻塞模式

为了便于阵列扩展,运算器的设计包含了互连接口,当不需要阵列扩展时,这些接口也可用来和外围电路作通信。

互连时采用 mesh 拓扑结构,通过阻塞位来控制互连通信<sup>[11]</sup>。非阻塞模式下,互连通信在一个时钟周期可完成读写操作。阻塞模式<sup>[12-13]</sup>下,则需要通过通信寄存器的数据有效标志位来判断能否进行读写操作,以免造成数据读写错误。通信寄存器位于数据存储单元,每个大小均为 33 bit,最高位为数据有效标志位,余下为数据位。将 1020 到 1023 四个连续的存储地址,设为互连时访问通信寄存器的接口地址。

具体说来,如果某个 PE 要给它周围的 PE 写数据,可以把指令中的目的地址设置为 1020-1023 等通信地址。此时,地址流水单元在控制数据回写时,会根据目的地址确定数据的去向。下面是一段简单的汇编片段示例:

```
ADD 1020, 0, 1; //本地 PE 给互连 PE0 写数据
SUB 1021, 2, 3; //本地 PE 给互连 PE1 写数据
MULT 1022, 4, 5; //本地 PE 给互连 PE2 写数据
DIV 1023, 6, 7; //本地 PE 给互连 PE3 写数据
```

如果某个 PE 要从它周围的 PE 读数据,可以把指令中的源地址设置为 1020-1023 等通信地址。此时,在访问通信地址时,存储控制会把通信接口寄存器中的值读出。下面是一段简单的汇编片段示例:

```
ADD 1, 0, 1020; //本地 PE 从互连 PE0 读数据
SUB 3, 2, 1021; //本地 PE 从互连 PE1 读数据
MULT 5, 4, 1022; //本地 PE 从互连 PE2 读数据
DIV 7, 6, 1023; //本地 PE 从互连 PE3 读数据
```

阻塞模式可以通过增加关键字来标识和指定,比如在汇编程序中增加关键字 BLOCK 标识的区块,汇编工具会自动把某一个区域的指令阻塞位置 1。通信时,遇到阻塞模式位有效,对通信寄存器的读写操作会根据寄存器数据有效标志位进行。

## 3 功能验证和应用分析

### 3.1 功能验证

设计中建立了处理单元的 SV<sup>[14]</sup>模型,并将 SV 模型和 RTL 模型的仿真结果进行了对比,通过了处理单元的基本功能测试。

通过载入图元的扫描程序对处理单元作运行较为复杂的算法测试。图 4 显示的是经过处理单元计算绘制出的点、直线、三角形。其中,扫描程序是基于自定义汇编并根据相关图形扫描算法来编写的。



图 4 扫描程序测试

### 3.2 应用分析

通过同时绘制多个较为复杂的三角形图元,来查看后级片段处理对扫描的响应情况。图 5(a)和(b)是两种不同结构扫描单元的处理过程对比。由图可以看出,在同时绘制 3 个三角形时,ASIC 结构的 1 号扫描单元处于阻滞状态,而可编程结构的 3 个扫描单元则处于均衡状态。

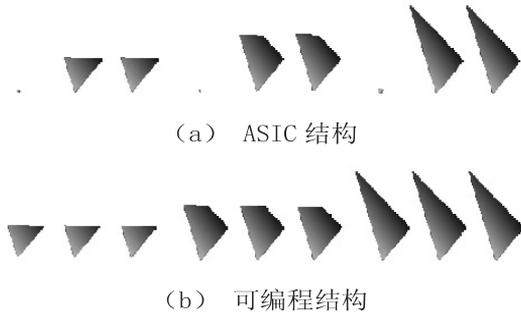


图 5 两种不同结构扫描单元的处理过程

表 1 分别对 ASIC 结构和可编程结构绘制不同个数的三角形图元的仿真时间做了统计(仿真在 Modelsim SE 6.5 下进行)。统计结果显示,ASIC 结构扫描单元在绘制单个三角形时比可编程结构快,使得后级的片段处理处于满载,在绘制多个三角形时绘制时间呈线性递增。可编程扫描单元在绘制单个三角形时较慢,后级的片段处理未处于满载,因此在绘制多个三角形时几乎不会增加时间。

表 1 三角形绘制仿真时间统计

三角形数目	ASIC 结构/kns	可编程结构/kns
1	1 550	5 165
2	3 385	5 155
3	4 735	5 150

由于图形处理中经常需要大量绘制图元,因此片段处理一般处于满载状态,可编程结构较之 ASIC 结构不会影响总体速度。

## 4 结束语

多格式定点运算器能够满足光栅化等阶段的运算处理要求,在处理速度上不如 ASIC 结构,但能够带来更大的灵活性,支持不同的实现方式;同时,减少了闲置的运算单元,意味着在一定程度上降低了电路面积。在负载均衡上,可编程结构比 ASIC 结构效果要更好一些。在绘制大量图元时,两者总体速度一致。

### 参考文献:

- [1] Owens J, Davis U C. GPU architecture overview [C]//Proc of SIGGRAPH. Santiago de Chile; [s. n.], 2007.
- [2] Sun Chih-Hao, Tsao You-Ming, Lok Ka-Hang, et al. Universal Rasterizer with edge equations and tile-scan triangle traversal algorithm for graphics processing units [C]//Proceedings of the 2009 IEEE international conference on multimedia and expo. New York: IEEE, 2009: 1358-1361.
- [3] 赫恩, 巴克. 计算机图形学 [M]. 蔡士杰, 宋继强, 蔡敏, 译. 第 3 版. 北京: 电子工业出版社, 2010.
- [4] Shen J P, Lipasti M H. 现代处理器设计: 超标量处理器基础 [M]. 张承义, 译. 北京: 电子工业出版社, 2004.
- [5] Hennessy J L, Patterson D A. 计算机系统结构: 量化研究方法 [M]. 白跃彬, 译. 第 4 版. 北京: 电子工业出版社, 2007.
- [6] Yi Kui, Ding Yuehua. 32-bit RISC CPU based on MIPS instruction fetch module design [C]//Proc of international joint conference on artificial intelligence. Hainan Island: IEEE, 2009: 754-760.
- [7] Balpande R S, Keote R S. Design of FPGA based instruction fetch & decode module of 32-bit RISC (MIPS) processor [C]//Proc of 2011 international conference on communication systems and network technologies. Katra, Jammu: IEEE, 2011: 409-413.
- [8] Yi Kui, Ding Yuehua. 32-bit RISC CPU based on MIPS-instruction decoder module design [C]//Proc of second Pacific-Asia conference on web mining and web-based application. Wuhan: IEEE, 2009: 124-128.
- [9] Vijayalakshmi V, Seshadri R, Ramakrishnan S. Design and implementation of 32 bit unsigned multiplier using CLAA and CSLA [C]//Proc of 2013 international conference on emerging trends in VLSI, embedded system, nano electronics and telecommunication system. Tiruvannamalai: IEEE, 2013: 1-5.
- [10] 黄秀荪, 叶青, 仇玉林. 高速除法器设计及 ASIC 实现 [J]. 微电子学与计算机, 2008, 25(2): 133-135.
- [11] 李涛, 肖灵芝. 面向图形和图像处理的轻核阵列机结构 [J]. 西安邮电学院学报, 2012, 17(3): 41-47.
- [12] Li Tao, Xiao Lingzhi, Huang Hucai, et al. PAAG: a polymorphic array architecture for graphics and image processing [C]//Proc of 2012 fifth international symposium on parallel architectures, algorithms and programming. Taipei: IEEE, 2012: 242-249.
- [13] Li Tao, Xiao Lingzhi. A thin-core array architecture for graphics and image processing [J]. Journal of Xi'an Institute of Posts and Telecommunications, 2012, 17(3): 41-47.
- [14] 克里斯·斯皮尔. SystemVerilog 验证 [M]. 张春, 译. 第 2 版. 北京: 科学出版社, 2009.

# 一种图形处理用的多格式定点运算器

作者: [易学渊](#), [李涛](#), [蒲林](#), [YI Xue-yuan](#), [LI Tao](#), [PU Lin](#)  
作者单位: [易学渊, YI Xue-yuan\(西安邮电大学 计算机学院, 陕西 西安, 710061\)](#), [李涛, 蒲林, LI Tao, PU Lin\(西安邮电大学 电子工程学院, 陕西 西安, 710061\)](#)  
刊名: [计算机技术与发展](#)   
英文刊名: [Computer Technology and Development](#)  
年, 卷(期): 2014(10)

引用本文格式: [易学渊. 李涛. 蒲林. YI Xue-yuan. LI Tao. PU Lin 一种图形处理用的多格式定点运算器](#)[期刊论文]-  
[计算机技术与发展](#) 2014(10)