

2D 图形加速器设计与实现

刘 鑫, 蒋 林

(西安邮电大学 研究生学院, 陕西 西安 710061)

摘 要: 设计并实现了 2D(2 Dimensional) 图形加速器电路, 提供基本图元: 点、线、多边形的绘制, 及任意大小和颜色的字符显示. 采用流水线结构, 流水线各级间采用握手信号进行信息交互, 缩短了处理时间, 提高了工作效率, 也便于各子模块间的时序配合. 对常用的两种二维绘图算法进行研究, 使用 Verilog HDL 硬件描述语言实现 2D 图形加速器的硬件电路. 与 System Verilog 搭建的电路行为模型对比仿真, 输出的图形数据信息结果一致, 并在 FPGA 上进行验证. 采用 SMIC 0.13 μm 标准 CMOS 工艺库综合, 工作频率达 205 MHz.

关键词: 2D 图形; 字体; ROP2 操作; FPGA

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2013)06-0075-05

Design and Implementation of 2D Graphics Accelerator

LIU Xin, JIANG Lin

(School of Gradient, Xi'an University of Posts and Telecommunications, Xi'an 710061, China)

Abstract: This paper designs and implements the 2D (2 Dimensional) Graphics Accelerator circuit. It provides the basic primitives point, line, polygon, and displays the words with any size and color. A pipeline architecture is used in the design. To exchange information handshake signals is used between the pipeline stages. It not only shortens the processing time and improves work efficiency, but also facilitates the co-ordination of the timing between the various sub-modules. The paper studies two kinds of high-frequency algorithms, uses the Verilog HDL language to describe 2D Graphics Accelerator hardware circuit. Compared with the simulation results of the behavior model, which is built by SystemVerilog, the results are consistent, and implements on FPGA. The operating frequency can reach 205 MHz with SMIC 0.13 μm standard CMOS cell library.

Key words: two-dimensional graphic; font; ROP2 operations; FPGA

1 引言

随着人们对图像显示质量和速度的要求日益提高, 使得图形处理器 (Graphics Processing Unit, GPU) 成为计算机显示系统的核心部件. 作为 GPU 的重要组成部分, 2D 图形加速器的研究和设计成为了热点问题^[1]. 如果说单纯的使用软件算法, 依靠 GPU 来进行图形的绘制, 图形显示需要占用大量的 CPU 时间, 不仅显示效率低下, 而且还会影响整个系统的性能, 因此使用硬件加速的处理方法实现图形的绘制是很有必要的.

文献[2]在对常用的二维画线和矩形填充算法进行研究的基础上, 使用 Verilog 硬件描述语言实现其功能, 并在 FPGA 上进行验证. 考虑到仅依靠直线和矩形实现二维绘图较单一化的问题, 本文增加了虚线的绘制, 矩形的渐变填充, 多边形的绘制及填充和字体/文本显示等功能, 并且提高了画线和矩形填充的工作效率. 设计采用并行流水线结构, 并在 Synopsys 公司的 VCS 环境下完成功能仿真, 选用 Xilinx 公司 Virtex6 系列的芯片进行了电路物理验证.

收稿日期: 2012-08-08; **修回日期:** 2012-10-16

基金项目: 国家自然科学基金面上项目“三维视频处理系统芯片动态可重构可编程体系结构研究”(61272120)

2 2D 图形处理相关算法的比较与选择

2.1 直线扫描算法

常用的绘制一个像素宽度直线的方法有:数值微分(DDA)画线算法、中点画线法和 Bresenham 算法^[3]. DDA 算法不利于硬件实现;Bresenham 算法不需要计算直线斜率,不用做除法;只需要整数加减法和乘 2 运算,乘 2 运算可以用硬件移位实现,算法速度快.

2.2 区域填充扫描线算法

传统的区域填充算法有两种:一种是种子填充算法^[4],从给定的位置开始填充直到指定的边界为止;另一种是扫描线算法,通过确定横跨区域的扫描线的覆盖间隔来填充.扫描线算法^[5]主要利用扫描线与多边形边缘的交点来判断是否填充.该算法用来填充比较简单的标准多边形区域,它对轮廓线的形状有一定的要求,在处理复杂区域时往往失效.该算法在处理带水平边的凹拐点时不能正确填充,这是该算法的一个突出问题,但由于利用了扫描线上像素之间的连贯性,计算量较小,因此具有较高的效率.本设计采用扫描线算法,基本步骤如下:

- (1) 求出每一条水平扫描线与多边形各边的交点;
- (2) 将每一条水平扫描线上的交点按 x 值递增的次序排序;
- (3) 将排序好的交点配成交点对;
- (4) 将交点对之间进行填充.

3 2D 图形加速器体系结构设计

图形加速器的功能包括:笔模式的创建、绘制直线、矩形、多边形的轨迹及字体/文本的显示,按照指定的填充模式进行扫描填充(快速单色填充、模板填充、渐变填充).这里所有的扫描、填充操作均以屏幕左上角为原点(0,0),向右和向下分别为 x 轴正方向和 y 轴正方向;同时,本设计还支持源图形与目标的光栅操作.根据图形处理器的功能,采用两级流水线结构实现,如图 1 所示.

流水线第一级是四个并行的功能块,根据接收到的命令不同,执行相应的功能单元.直线扫描单元绘制单像素宽度的带样式的直线;矩形扫描单元支持绘制线框矩形、填充矩形、带线框的填充矩形(支持刷子、固色、水平渐变和垂直渐变填充);多边形单元绘制带边框的填充多边形(支持刷子和固色填充).字体/文本单元完成任意大小和颜色的字符显示.

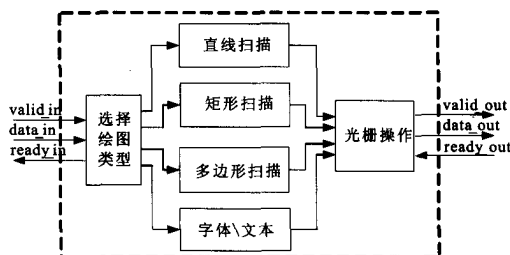


图 1 2D 图形加速器的总体结构图

第二级流水线实现 ROP2 操作^[6]功能,其中包括读像素模块、ROP2 操作模块、写控制模块.读像素模块根据目标图形存储的位置,将目标像素读出;ROP2 操作模块对源图形和目的图形的颜色进行与、或、异或、取反等逻辑运算操作,最多支持 16 种光栅操作(即全套 ROP2 操作);写控制模块根据图形显示的位置,计算出地址并将信息存储在对应的位置上.

3.1 直线扫描单元

直线扫描单元采用前面介绍的 Bresenham 算法,支持任意起点和终点的直线绘制,支持 32×1 的单色笔模式的设置.对于同一段使用虚线模式,扫描方向从左向右或者从右向左绘制出的线段效果是不同的,所以采用 Bresenham 算法交换坐标时,不能交换直线起点和终点的位置,以免影响绘制虚线的效果.直线扫描单元根据接收到的命令绘制不同模式的直线,端点坐标通过读取寄存器获得.命令中包含绘制直线的笔的颜色、模式等属性信息.扫描后产生的像素值送到流水线下一级 ROP2 操作单元.

直线扫描单元通过状态机实现:在 init 状态,接收并保存 Pen 属性信息;如果收到 DrawLine 命令跳转到 exchange 状态,计算 dx 和 dy 的值,交换直线的坐标使得直线斜率在 $0 \sim 1$ 范围内.在 exchange 状态,计算决策参数 P_0 (即 $d_1 - d_2$)的初始值,状态机跳转到 update 状态.在 update 状态,如果扫描点是直线的第一个点并且笔的模式表示显示该点,则装配输出数据,等待输出,跳转到 wait 状态;如果不是第一个扫描点,分别计算直线的新的坐标值及 P_i 的值,当 dy 是负数时,跳转到 cur_y 状态;如果 dy 是正数,跳转到 yinc 状态.在 cur_y 和 yinc 状态,判断是否将计算得到的坐标进行交换,状态机跳转到 coordinate 状态.在 coordinate 状态,需要更新画笔的模式,判断画笔的模式是否为显示该点,如果不显示,跳转到 update 状态,重新扫描直线的下一个像素点;如果该点需要显示,则按照指定的格式装配数据,等待输出,状态机跳转到 wait 状态.在 wait 状态,等待下一级流水线的允许输出信号有

效(高电平),如果信号无效,状态机保持此状态;如果信号有效,输出预先装配的数据,同时数据有效信号置为低电平,在这一过程中需要判断输出的像素点是否为直线的最后一个点,如果是最后一个点,状态机跳转到 init 状态;如果不是最后一个像素点,状态机跳转到 update 状态. 直线扫描单元的状态跳转过程如图 2 所示.

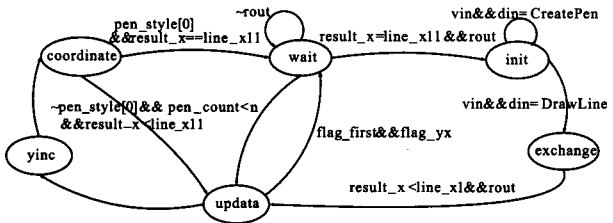


图 2 直线生成模块状态机

3.2 矩形扫描单元

矩形扫描单元包括线框和填充两种绘制模式. 填充模式又可以分为固色填充, 8 * 8 刷子填充, 水平渐变填充, 垂直渐变填充四种模式, 每次扫描绘制一个矩形时, 只支持选择一种填充模式. 根据矩形左上角和右下角的两个顶点坐标可以确定矩形的左右边界和上下边界的坐标, 从而在每次行扫描时, 矩形的右边界可以作为纵坐标递增的判断条件, 下边界可以作为终止扫描的判断条件.

矩形扫描单元的结构图如图 3 所示. 首先, 根据命令选择矩形的类型, 进入到线框模式或者填充模式; 其次, 按照选择的类型进入各个填充单元. 笔的模式填充, 是使用 32 * 1 的画笔绘制矩形边框. 8 * 8 刷子填充模式, 从矩形的左上角开始使用刷子模式, 进行行扫描, 在同一行重复使用刷子的 8 个像素点, 直到一行结束. 渐变填充^[7]的基本思想是采用 Gouraud 着色方法, 假设顶点 A、B 的颜色是已知的, 分别为 C_A、C_B, 所以线段 A_B 之间的任意一点 X 的颜色值如式(1)所示.

$$C_x = C_A \cdot \frac{Y_1 - Y_B}{Y_A - Y_B} + C_B \cdot \frac{Y_A - Y_1}{Y_A - Y_B}$$

(1)

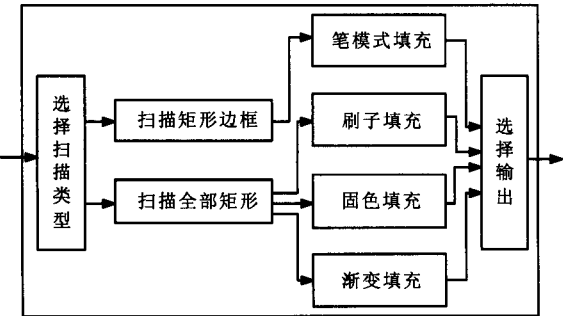


图 3 矩形扫描单元实现结构图

3.3 多边形扫描单元

多边形扫描单元支持凸多边形的绘制及内部填充(固色填充和刷子填充两种填充模式). 多边形的顶点信息由用户指定, 刷子的颜色信息通过配置寄存器得到. 填充模式及画笔模式由命令提供.

在对多边形扫描时, 对于一个给定的多边形, 用一组水平(垂直)的扫描线进行扫描, 对每一条扫描线均可以求出与多边形边的交点, 这些交点将扫描线分割成落在多边形内部的线段和落在多边形外部的线段, 并且二者相间排列. 于是, 将落在多边形内部的线段上的所有像素点赋以给定的色彩值. 可见, 多边形扫描算法中不需要检验每个像素点, 而只考虑与多边形相交的交点分割后的扫描线段. 对于一条扫描线的处理, 首先求出与多边形各边的交点, 然后将这些交点按递增顺序排序, 即可以从左到右确定落在多边形内部的线段. 如第一个交点与第二个交点, 第三个交点与第四个交点等等, 然后在进行像素的置色, 即采用扫描线填充算法的四个步骤: 求交点、交点排序、交点匹配以及区间填充颜色. 扫描流程依次为: 获取多边形顶点信息, 坐标排序, 生成活性边表, 计算行扫描交点, 坐标配对, 像素输出六个过程. 流程的每一级均表示独立的一级流水线, 流水线之间采用握手信号, 流水线内部采用状态机实现.

3.4 字体/文本单元

主要完成任意大小字体和文本的显示, 支持字体/文本的颜色设置. 字体/文本的数据信息(由软件生成的字模)会预先存储到 ROM 和主存储中, 根据文本/字体处理单元所要绘制的字体选择相应的字模信息.

此单元采用三级流水线结构实现, 具体设计结构如图 4 所示. 第一级流水线读控制模块根据字码的值从存储里读字模信息, 然后装配发送给下一级; 第二级流水线缩放模块完成对字体的放大或缩小操作, 采用线性插值算法对上一级输出的像素进行插值计算, 得到缩放后的像素并传送给下级流水线; 第三级流水线写模块根据字体/文本的显示坐标, 计算出地址并将像素写到相应的存储里.

3.5 ROP2 操作

ROP2(二元光栅操作)操作的操作对象是笔和目标图像. 它指定了图像设备接口(Graphics Device Interface, GDI)如何将已选择的笔与目标图像结合在一起. 一共有 16 种光栅操作码, 每种光栅操作码代表了唯一一种逻辑运算, 包括按位与、按位或、按位取反及按位异或四种操作, 描述了笔的像素值与

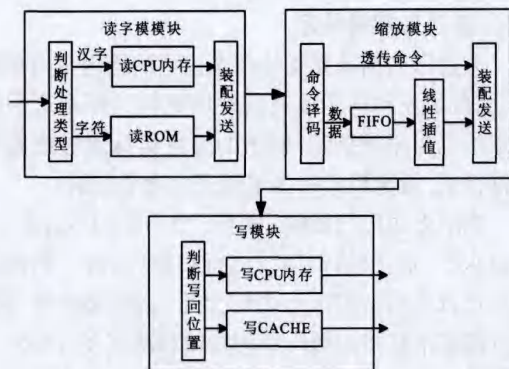


图4 字体/文本处理单元的流水线结构图

目标图像的像素值的结合方式.很多位块传输(blit)函数,如 BitBlt,用来修改传输位图,这些函数就是通过把图像与笔和刷子结合起来,经过某些逻辑操作达到预期的效果.这样的工作反复进行就会形成一系列的动画效果.

ROP2 模块要将笔的像素值与目标图像的像素值进行逻辑操作,笔的像素值是从前一级流水线扫描得到的.在本系统中,目标图像默认存储在 CPU 中,每一次 ROP2 操作都需要将目标图像的像素值读取到本模块.笔的像素带有坐标和颜色值等信息,根据像素对应的坐标就可以计算出目标图像像素的地址,这样就得到了 ROP2 操作的另外一个操作数——目标图像的像素值.再根据指定的 ROP2 操作码进行逻辑操作,便得到了本次图形操作的结果.

4 验证

电路设计采用 Verilog 硬件描述语言,在 Synopsys 公司的 VCS 环境下完成功能仿真,使用 VCS 内置的可视化环境 DVE 工具观测仿真波形.采用 SMIC0.13 μm CMOS 工艺标准单元库进行综合,使用 TCL(ToolCommandLanguage)脚本语言来描述综合命令,在 Design Compiler 环境下完成逻辑综合,电路工作频率可达到 205 MHz.

设计选用 Xilinx 公司的 Virtex6 系列的芯片 Virtex XC6VLX550T 进行了电路物理验证,该系列器件具有内嵌处理器、速度快、资源丰富、连续式布线结构等优点,能满足本设计电路高速工作频率及资源充足的要求. FPGA 布局布线后的芯片资源使用报告如表 1 所示,2D 图形加速器消耗了 2 130 个寄存器和 3 899 个 LUT 资源,占总资源的 1%,极大地节约了 FPGA 的可编程资源.

表1 FPGA 芯片资源使用报告表

Logic Utilization	Utilization
Target Device	xc6vlx550t-1ff1760
Slice Registers	2 130/687 360(0 %)
Slice LUTs	3 899/343 680(1 %)
LUT-FF pairs	2 798/4 400(63 %)

为了更好地观察硬件设计的结果,使用 FPGA 将结果输出到屏幕上,以屏幕左上角为原点,最终显示结果如图 5 所示.

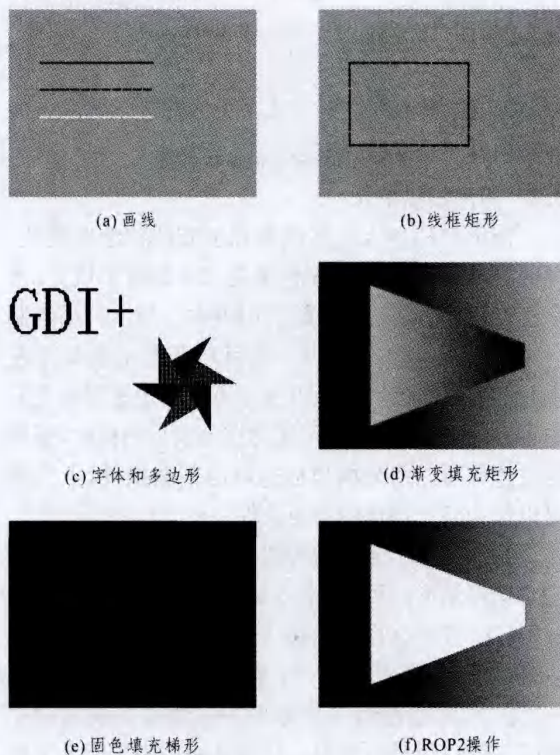


图5 显示结果

图 5 中,(a)为图形的画线功能,包括直线、虚线,及将虚线做 ROP2 操作的结果;(b)为矩形的线框模式,未进行矩形内部填充,画笔的模式设置为 16'h0fff;(c)启用文本和多边形功能,使用多边形算法绘制的三角形,包括刷子填充和固色填充两种模式;(d)矩形的垂直渐变填充功能;(e)多边形扫描单元绘制的梯形,使用白色填充多边形内部,多边形边框使用绿色画笔绘制;(f)在(e)图绘制绿框白色梯形的基础上,增加与目的图形(渐变背景)做 ROP2 操作的结果.

设绘制一条 480 像素的线段以及填充一个 640×480 的矩形,与文献[4]的对比结果如表 2 所示.从表 2 可以看出在使用相同的算法实现直线与矩形填充的情况下,本设计绘制直线所使用的时间仅为

文献[4]的 58.8%,矩形填充的时间仅为文献[4]的 69.1%,对比表明,本设计在占用较少的 FPGA 资源的基础上,极大地提高了绘制二维图形的效率。

表 2 时间对比

	Draw Line (480 pixels)	Fill rectangle (640 * 480 pixels)
This design	51 492.5 ns	13 841 154.5 ns
References[4] design	87 587.5 ns	20 043 587.5 ns (20 ms)
This design/ References[4] design	58.8%	69.1%

5 结束语

本设计在研究了扫描算法的基础上使用硬件程序实现了二维基本图形的绘制,并且做了 ROP2 操作,以便日后优化设计,使这些简单的图形经过一定的 ROP2 形成丰富多彩的 2D 显示效果. 设计采用典型的 Top-Down 设计方法,对电路进行了模块划分和详细设计,使用 Verilog 硬件语言描述设计,并对设计电路进行了功能验证和逻辑综合及 FPGA 验证. 综合及验证结果表明设计电路的正确性,且工作频率可以达到 205 MHz. 除此之外,本设计还采用 SystemVerilog 搭建的测试平台对 2D 图形加速器的总体电路进行了功能对比,输出结果完全一致。

参考文献:

[1] 袁扬智,韦明,曾献军. 2D 图像硬件加速引擎的设计优化[J]. 微电子学与计算机,2009,26(4):241-248.

[2] 高齐,熊庭刚,秦信刚. 二维图形加速技术与 FPGA 实现[J]. 计算机与数字工程,2011,39(4): 142-145.

[3] 刘勇奎,周晓敏. 逐点生成参数曲线的双步算法[J]. 计算机辅助设计与图形学学报,2000,14(7):619-623.

[4] 任继成,刘慎权. 区域填充扫描线算法的改进[J]. 计算机辅助设计与图形学学报,1998,10(6):481-486.

[5] 张玉芳,刘君,彭燕. 一种改进的扫描线多边形填充算法[J]. 计算机科学,2005,32(6):164-166.

[6] Windows Microsoft. Binary raster oiperations[EB/OL]. [2010-8-15]. [http://msdn.microsoft.com/en-us/library/windows/desktop/dd183369\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd183369(v=vs.85).aspx).

[7] Francis S Hill Jr, Stephen M Kelley. 计算机图形学 (OpenGL 版)[M]. 3 版. 胡事民,刘利刚,刘永进,等译. 北京,清华大学出版社,2009.

作者简介:

刘鑫女,(1988-),硕士研究生. 研究方向为专用集成电路设计与系统集成.

蒋林男,(1970-),教授,硕士生导师. 研究方向为专用集成电路设计.

(上接第 74 页)

[6] Edward Stott,Pete Sedcole,Peter Y K. Fault tolerant methods for reliability In FPGAs[C]//FPL 2008. London: Electrical and Electronic Engineering Imperial College, 2008:415-420.

[7] Brenda Lin,Mai Irie. Evolvable Hardware[R]. CRAB Lab:CIT595 Research Project, 2008.

[8] 刘文冰,刘尚合,原亮. 用于静电放电 EMP 实验的三模冗余技术[J]. 军械工程学院学报,2006,18(4):29-31.

[9] 满梦华,褚杰,施威,等. 嵌入式数字电路故障自修复技术研究[J]. 河北科技大学学报,2011(32):142-144.

[10] Miller, Julian F. Cartesian genetic programming[M]. Natural Computing Series, New York: Springer, 2011.

[11] 林勇,罗文坚,王煦法. 基于遗传算法的异构硬件电路冗余系统构造方法[J]. 电路与系统学报,2009,14(3): 61-65.

作者简介:

施威男,(1988-),硕士研究生. 研究方向为智能检测与故障诊断.

原亮男,(1977-),教授. 研究方向为智能检测与故障诊断、电磁防护仿生技术.

解双建男,(1986-),硕士研究生. 研究方向为嵌入式系统设计.

满梦华男,(1984-),博士研究生. 研究方向为电磁防护理论与技术.