文章编号:1673-0062(2017)03-0055-05

基于 FPGA 的图像叠加及 VGA 显示设计

陈　权,朱卫华*,曹　亮,陈志勇

(南华大学 电气工程学院,湖南 衡阳 421001)

摘　要:为了实现两幅图像的半透明叠加的 VGA 显示,根据图像叠加原理和 VGA 显示原理,使用 Altera 公司的 FPGA 芯片 CycloneⅣEP4CE15F17C8 作为控制核心,采用 VerilogHDL 语言实现图像叠加 Alpha 算法,采用移位操作解决 FPGA 处理浮点运算问题,通过外部按钮来控制图像的透明度.用 FPGA 控制实现图像叠加的 VGA 实时显示,该设计方案具有体积小、成本低、速度快、工作稳定等优点.Modelsim 仿真结果和实际显示效果验证了设计的正确性与可行性.

关键词:FPGA;图像叠加;VGA 显示;移位操作;Modelsim 仿真

中图分类号:TN919.5　　**文献标志码**:A

The Design of Image Overlay and VGA Display Based on FPGA

CHEN Quan, ZHU Wei-hua*, CAO Liang, CHEN Zhi-yong

(School of Electrical Engineering, University of South China, Hengyang, Hunan 421001, China)

Abstract: In order to achieve the semi-transparent overlay of two images and VGA real-time display, according to the principle of image overlay and the principle of VGA display, Altera's chip CycloneⅣEP4CE15F17C8 is used as the control core. The image overlay Alpha algorithm is achieved by VerilogHDL language. And the shift operation is adopted to solve the problem of FPGA processing floating point. And the transparency of the image is controlled by external buttons. The image overlay VGA display is realized by FPGA control, which has advantages such as smaller size, low cost, fast speed and more reliable stability. The Modelsim simulation results and the actual display results verify the correctness and feasibility of the design.

key words: FPGA; image overlay; VGA display; shift operation; Modelsim simulation

收稿日期:2017-05-29

基金项目:湖南省自然科学基金资助项目(16C1375)

作者简介:陈　权(1989-),男,硕士研究生,主要从事电子系统集成、FPGA 的图像采集处理等方向的研究.E-mail:479559745@qq.com. * 通讯作者:朱卫华,E-mail:zhuweihua64@163.com

0 引 言

视频图像叠加技术是当前在屏幕显示(on screen display,OSD)技术的一个重要部分.视频图像叠加的应用越来越广,如雷达监测[1],汽车智能[2],医学成像[3]等方面.目前大多数视频图像叠加都采用专业视频图像叠加芯片完成,虽能实现各种视频图像处理功能,但具有一定的局限性[4],如不能叠加汉字或系统较复杂,不能灵活的适应各种环境.而 PC (personal computer,个人计算机)处理大分辨率的高清图像时速度较慢,图像透明叠加效果无法实时呈现.利用 FPGA (field-programmable gate array,现场可编程门阵列)芯片适应能力强和并行处理的优势,用移位操作处理设计算法中的浮点运算,提高运算速度,实现图像的透明叠加效果的实时显示.

1 图像叠加

1.1 叠加原理

视频图像叠加是指在原视频图像信号中叠加上另一视频图像信号,使得原视频图像中叠加上字符或图像.其基本原理是将存放在存储器里的字符或图像取出来转换成满足一定格式要求的信号,同输入的另一视频或图像信号合并,使被叠加的电信号与原视频图像信号在指定的位置上同步显示.因为图像的基本组成单元是像素,所以图像叠加的实质是在像素级为每个像素点选择信号,完成像素点的定位(确定行、列位置)和图像信号的输出(选择显示哪幅图像像素的电信号)[5-6].

1.2 叠加算法

Alpha 透明混叠算法公式主要有下面三种[7-8]:

$$imgL(x,y)=alpha1(x,y)*imgM(x,y)+(1-alpha1(x,y)*alpha2(x,y)*imgN(x,y) \quad (1)$$

$$imgL(x,y)=imgM(x,y)-imgM(x,y)/alpha(x,y)+imgN(x,y)/alpha(x,y) \quad (2)$$

$$imgL(x,y)=alpha(x,y)*imgM(x,y)+(1-alpha(x,y)*imgN(x,y)) \quad (3)$$

公式(1)、(2)、(3)中,imgL(x,y)为混叠后的的图像,imgM(x,y),imgN(x,y)为混叠前相同分辨率的两幅图像,alpha1(x,y)前景透明值,alpha2(x,y)为背景透明值,alpha(x,y)为透明度值.三种算法都可以实现透明图像的叠加,用算法公式(3)来实现半透明图像叠加.公式(3)中,alpha(x,y)是图像 imgM 的透明度系数,取值在 0 到 1 之间.Alpha(x,y)取 0 表示图像 imgM 完全透明即只显示图像 imgN,alpha(x,y)取 1 表示图像 imgM 完全不透明即只显示图像 imgM.改变 alpha(x,y) 值可以得到两幅图不同透明度的叠加效果.

2 图像叠加的 FPGA 实现

2.1 系统硬件结构

采用 Altera 公司的 Cyclone Ⅳ EP4CE15F17C8 作为控制芯片,采用成都国腾电子公司的 GM7123 作为数模转换芯片.系统由 FPGA 生成水平彩条和垂直彩条图像,并进行透明化图像叠加,通过外部按键改变 alpha 的值来控制两幅图像间的透明度,最终显示在 VGA (video graphics array,视频图形阵列)显示器上.整个系统包括五个模块,lcd_display 模块、lcd_driver 模块、systime_ctrl_pll 模块、alpha_key 模块、alpha_adj 模块,如图 1 所示.lcd_display 模块:控制生成水平彩条图像和垂直彩条图像,对数据进行移位操作,实现两幅图像的透明叠加.lcd_driver 模块:实现 1 280×1 024@60 Hz 分辨率下 VGA 驱动显示.systime_ctrl_pll 模块:利用 FPGA 的 IP(intellectual property,知识产权)核生成所需的时钟频率.alpha_key 模块和 alpha_adj 模块:实现利用外部按键控制 alpha 的值,实时改变两幅图像的透明度.

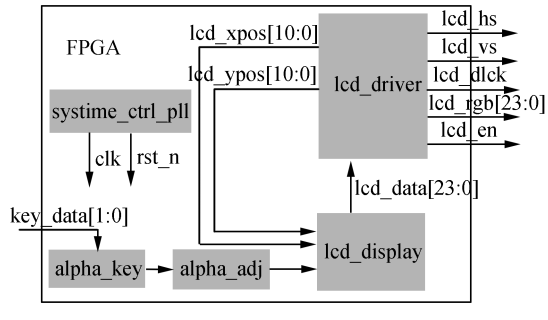


图 1 系统硬件框图

Fig.1 Structure of system hardware

2.2 VGA 显示原理

VGA 接口采用 15 个接口信号,其中用于显示的 5 个主要接口信号为:3 个色彩线(红绿蓝)接口信号和 1 个行同步信号、1 个场同步信号[9-10].实现 VGA 正确地显示图像实质是在一定时钟频率下,控制产生准确的 HSYNC 行同步信号和 VSYNC 场同步信号及行、场消隐信号.采用了的 VGA 分辨率为 1280×1024@60 Hz 的模式.依据 VGA 工业标准,每场(V_TOTAl)扫描 1 066 个行,包括场同步时期(V_SYNC)3 行,场消隐后

肩(V_BACK) 38 行,场显示时期(V_DISP) 1 024 行, 场消隐前肩(V_FRONT)1 行;每行(H_TOTAl)扫描 1 688 个像素,包括行同步时期(H_SYNC) 112 个像素,行消隐后肩(H_BACK) 248 个像素,行显示时期(H_DISP)1 280 个像素,行消隐前肩(H_FRONT)48 个像素,如图 2 所示.因此需要的像素时钟频率为 1066 * 1688 * 60 Hz 等于 108 MHz.

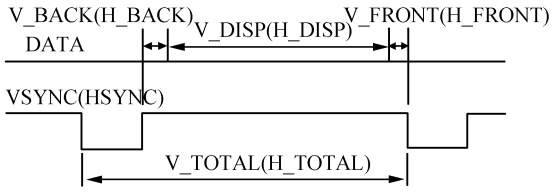


图 2　VGA 扫描时序

Fig.2　VGA scan sequence

2.3　图像叠加的实现

用 8 位二进制数表示 alpha、Red、Green 和 Blue,其中 Red、Green 和 Blue 分别表示红绿蓝三原色的颜色值,alpha 变量表示颜色的透明程度,那么它们的值定义在 0 到 256 之间.透明的比例按 2 的 N 次幂来分级,这样可以利用移位操作实现快速运算.这里透明的比例按 256 级分级,依据公式(3),则各通道图像叠加算法公式为[11]:

R(L) = alpha * R(M) + (256 - alpha) * R(N)>>8;　　(4)

G(L) = alpha * G(M) + (256 - alpha) * G(N)>>8;　　(5)

B(L) = alpha * B(M) + (256 - alpha) * B(N)>>8;　　(6)

公式(4)、(5)、(6)中,R(L)、G(L)、B(L)分别指图像 L 的 RGB 分量原色值,R(M)、G(M)、B(M)分别指图像 M 的 RGB 分量原色值,R(N)、G(N)、B(N)分别指图像 N 的 RGB 分量原色值,易知 alpha 值取 0 表示图像 M 完全透明,取 256 则表示图像 M 完全不透明.

图像叠加的 VerilogHDL 核心代码如下:

```
always@ (posedge clk or negedge rst_n)
begin
  if(! rst_n)
    begin
     lcd_data1 <= 24'h0;
     lcd_data2 <= 24'h0;
    end
  else
    begin
//VGA_Horizontal_Color
    lcd_data1 <=(lcd_ypos >= 0 &&
lcd_ypos < (V_DISP/8) * 1)? RED:
    (lcd_ypos >= (V_DISP/8) * 1 &&
lcd_ypos < (V_DISP/8) * 2)? GREEN:
    (lcd_ypos >= (V_DISP/8) * 2 &&
lcd_ypos < (V_DISP/8) * 3)? BLUE:
    (lcd_ypos >= (V_DISP/8) * 3 &&
lcd_ypos < (V_DISP/8) * 4)? WHITE:
    (lcd_ypos >= (V_DISP/8) * 4 &&
lcd_ypos < (V_DISP/8) * 5)? BLACK:
    (lcd_ypos >= (V_DISP/8) * 5 &&
lcd_ypos < (V_DISP/8) * 6)? YELLOW:
    (lcd_ypos>= (V_DISP/8) * 6  &&
    lcd_ypos<(V_DISP/8) * 7)? CYAN:ROYAL;
    //VGA_Vertical_Color
    lcd_data2 <=(lcd_xpos >= 0 &&
lcd_xpos < (H_DISP/8) * 1)? RED:
    (lcd_xpos >= (H_DISP/8) * 1 &&
lcd_xpos < (H_DISP/8) * 2)? GREEN:
    (lcd_xpos >= (H_DISP/8) * 2 &&
lcd_xpos < (H_DISP/8) * 3)? BLUE:
    (lcd_xpos >= (H_DISP/8) * 3 &&
lcd_xpos < (H_DISP/8) * 4)? WHITE:
    (lcd_xpos >= (H_DISP/8) * 4 &&
lcd_xpos < (H_DISP/8) * 5)? BLACK:
    (lcd_xpos >= (H_DISP/8) * 5 &&
lcd_xpos < (H_DISP/8) * 6)? YELLOW:
    (lcd_xpos>=(H_DISP/8) * 6 &&
lcd_xpos<(H_DISP/8) * 7)? CYAN:ROYAL;
    end
end
wire[23:0] red_data = ((9'd256 - alpha) * lcd_
data1[23:16]+alpha * lcd_data2[23:16])>> 8;
wire[23:0] gre_data = ((9'd256 - alpha) * lcd_
data1[15:8]+alpha * lcd_data2[15:8])>> 8;
wire[23:0] blu_data = ((9'd256 - alpha) * lcd_
data1[7:0]+alpha * lcd_data2[7:0])>> 8;
assign     lcd_data = {red_data[7:0],gre_data[7:
0],blu_data[7:0]};
```

上述代码中首先由 FPGA 内部产生两幅图像,lcd_data1 为水平彩条图像,lcd_ypos 分为 8 部分,由上到下分别为红、绿、蓝、白、黑、黄、青、品

蓝;lcd_data2 为垂直彩条图像,lcd_xpos 亦分为 8 部分,由左到右分别为红、绿、蓝、白、黑、黄、青、品蓝.接着对红绿蓝三通道的每一像素进行叠加,实现图像叠加算法.最后取红色通道的低 8 位,绿色通道的低 8 位,蓝色通道的低 8 位,完成 RGB888 格式图像 VGA 显示.

3 实验及结果

3.1 仿真结果

将 alpha 的值设置为 160,利用 Modelsim 软件进行仿真,仿真结果如图 3.从仿真结果可以得到 lcd_data 和 lcd_data1、lcd_data2 的值,以黑竖线位置为例分析,此时 lcd_data = 9f9fff, lcd_data1 = 0000ff, lcd_data2 = ffffff, 依据前述混合算法 red_data = {(256-160) * 00+160 * ff}/256,经过计算和换算后等于 9f(十六进制,下同),同理,算得 gre_data 等于 9f、blu_data 等于 ff,组合起来就是 9f9fff,与仿真出来 lcd_data 数据的结果一致,可知设计符合要求.

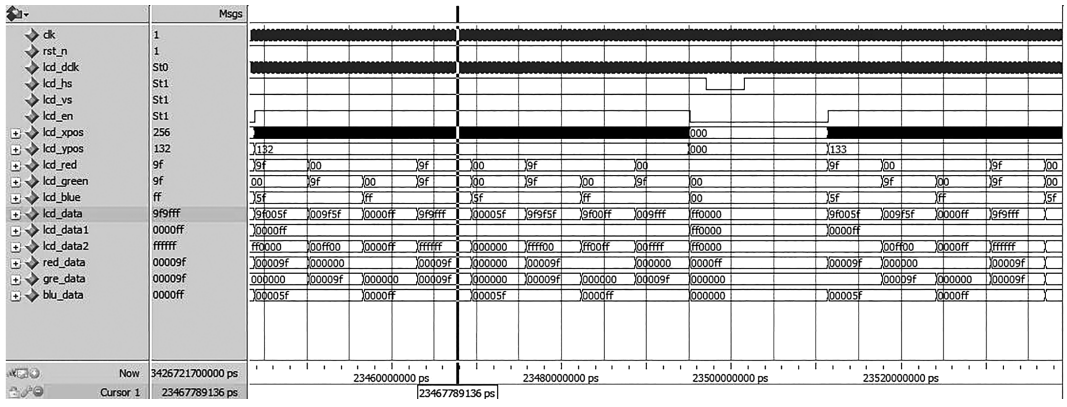


图 3 alpha=160 时 Modelsim 仿真结果

Fig.3 The Modelsim simulation results when alpha=160

3.2 显示效果

通过外部按键控制实现图像叠加的部分 VGA 的显示效果,如图 4.从图 4 中可以看到,当 alpha 值等于 256 时完全只显示垂直彩条,彩条颜色从左到右分别为红、绿、蓝、白、黑、黄、青、品蓝,随着 alpha 值的减少,垂直彩条越来越透明,水平彩条越来越不透明,直至 alpha 等于 0 时完全只显示水平彩条,彩条颜色从上到下分别为红、绿、蓝、白、黑、黄、青、品蓝,符合预期效果.

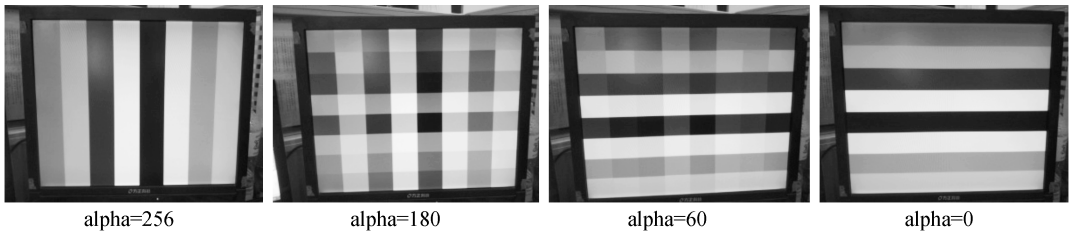


图 4 VGA 显示效果图

Fig4.VGA display effect

4 结 论

利用 FPGA 并行处理数据的优势,通过移位操作处理算法中的浮点运算,实现了两幅图像的透明叠加及其效果的实时显示.采用 FPGA 设计集成度高,设计灵活方便,还可以根据需求叠加各种字符或 Logo 图标及多种分辨率的图像.这对在高清图像显示领域中实现视频图像和半透明字幕图像的叠加处理有参考作用.

参考文献:

[1] LUBCZONEK J. Analysis of accuracy of surveillance radar image overlay by using georeferencing method [C]//Radar symposium,IEEE,2015:876-881.

[2] ZHANG S,ZHANG H,CHEN B,et al.On-screen-display (OSD) and SPI interface on CMOS image sensor for automobile application[C]//Fifth international conference on computational intelligence, communication systems and networks,IEEE computer society,2013:405-408.

[3] ZHANG X,CHEN G,LIAO H.A high-accuracy surgical augmented reality system using enhanced integral videography image overlay[C]//Engineering in medicine & biology society. conf proc IEEE eng med biolsoc, 2015:4210.

[4] 李华英,向怀坤,曹泉,等.基于 DSP/CPLD 的视频叠加系统设计与实现[J].交通信息与安全,2005,23(6):79-82.

[5] 王旭东,王新赛,李坚.基于 FPGA 的视频图像叠加系统的设计与实现[J].电子技术应用,2007,33(11):38-40.

[6] 高军.视频图像叠加的 FPGA 设计与实现[D].长沙:湖南大学,2008.

[7] 胡妍,李逸仙,张婕,等.基于 FPGA 的图像处理及叠加系统的设计与实现[J].弹箭与制导学报,2016,36(6):154-156.

[8] CRAZY B.RGBA alpha 透明度混合算法[DB/OL].http://www. cnblogs. com/crazybingo/archive/2012/05/11/2496440.html(2012-05-11).

[9] 杜宗展,王振河,冯迎春.基于 FPGA 的 VGA 图像显示系统的设计[J].现代电子技术,2015,38(16):95-99.

[10] 陈启武,白天蕊.基于 FPGA 的 VGA 显示系统的设计与实现[J].电子产品世界,2016,23(2/3):48-50.

[11] 曹山,刘继勇.基于 FPGA 的 Alpha 半透明图像叠加算法硬件实现[J].电子设计工程,2015,23(10):172-174.