

基于 FPGA 的图像预处理单元的硬件实现^①

朱喜 王玲 郑善贤 郭湘勇 (湖南大学 电气与信息工程学院 湖南 长沙 410082)

摘要: 为了实现图像的实时处理,常采用现场可编程门阵列 FPGA 对采集到的图像数据进行预处理。以对 Micron MT9V112 传感器的 Bayer 图像数据处理为例,首先就 Bayer 数据坏点修正、Bayer 转 RGB888 及 RGB888 降噪进行了介绍,然后应用 Verilog HDL 语言设计出相应的硬件模块,最后结合 MATLAB 对硬件模块处理后的数据做了相应的测试。仿真结果表明,硬件模块对 640x480 数据的处理满足系统实时性要求。

关键词: 图像预处理单元; Verilog HDL; 现场可编程门阵列

Hardware Implementation of PPU Based on FPGA

ZHU Xi, WANG Ling, ZHENG Shan-Xian, GUO Xiang-Yong

(College of Electrical and Information Engineering, Hunan University, Changsha 410082, China)

Abstract: During the procedure of real-time image processing, FPGA is often utilized to preprocess the collected digital image. This project looks at how to process the Bayerdata, which comes from Micron MT9V112-1/6-Inch SOC VGA CMOS DIGITAL IMAGE SENSOR. The study first introduces the bad pixel correction of Bayerdata, the procedures of converting an image from the Bayer format, to an RGB888 format and RGB888 images denoising methods. The paper then examines the data from an image preprocessing module based on Verilog HDL via Matlab. The discovery demonstrates that this module can successfully process 640x480 pixels with desirable real-time results.

Keywords: image preprocessing unit; Verilog HDL; FPGA

1 引言

图像传感器的众多输出格式中, Bayer 格式被广泛的运用于 CCD 和 CMOS 摄像头,它允许从单一平面中得到彩色图像。本文以 Bayer 图像预处理为例,介绍了图像的坏点修正、Bayer 转 RGB888、及 RGB888 降噪,并设计出相应的硬件模块。由于整个设计,采用流水线结构,所以仿真时钟频率可以达到 71.4MHz,另外模块内部全部采用移位累加的方法替代硬核乘法器,所以门电路的数目也是相对较少的。

2 图像预处理模块实现原理

2.1 坏点修正实现原理

由于现有技术的不足,坏点不可避免。我们所采

用的坏点修正过程如下,首先在坏点检测时记录好坏点位置,并把坏点信息写入到寄存器中,然后在坏点修正模块中依据这些信息对数据进行实时处理。图 1 给出了 Micron MT9V112 传感器的 Bayer 图像输出格式,下面结合图 1 给出坏点修正时,主体部分的实现原理。这里列出四种典型位置坏点修正方法,这四个典型位置的像素点分别记为 $p_{C(46,63)}$, $p_{C(46,64)}$, $p_{C(46,63)}$, $p_{C(47,64)}$, 以下是相应的修正算法:

$$p_{C(46,63)} = p_{(46,61)} * 0.375 + p_{(44,63)} * 0.375 + p_{(44,65)} * 0.25$$

$$p_{C(46,64)} = p_{(46,62)} * 0.25 + p_{(45,63)} * 0.375 + p_{(45,65)} * 0.375$$

$$p_{C(47,63)} = p_{(47,61)} * 0.25 + p_{(46,62)} * 0.375 + p_{(46,64)} * 0.375$$

$$p_{C(47,64)} = p_{(47,62)} * 0.375 + p_{(45,64)} * 0.375 + p_{(45,66)} * 0.25$$

^① 基金项目:湖南省科技计划(2009GK3082)

收稿时间:2010-01-17;收到修改稿时间:2010-03-20

这里的 $pc_{(46,63)}$ 表示处理后用来替代坏点 $pc_{(46,63)}$ 的数据。

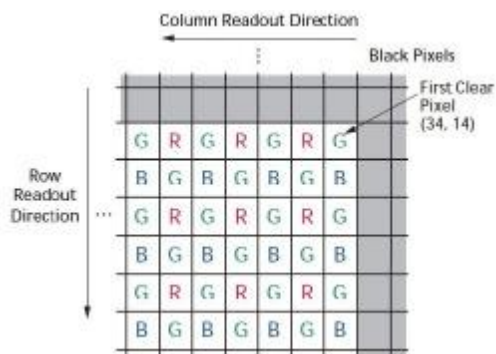


图 1 Micron MT9V112 传感器的 Bayer 图像输出格式

2.2 格式转换实现原理

本部分主要实现 Bayer 格式到 RGB888 格式转换, 结合图 1 给出了格式转换时, 主体部分的实现原理。下面写出了四种典型位置格式转换的算法:

$$pt_{(46,63)} = G1(p_{(45,62)}, p_{(45,63)}, p_{(45,64)}, p_{(46,62)}, p_{(46,63)}, p_{(46,64)}, p_{(47,62)}, p_{(47,63)}, p_{(47,64)});$$

$$pt_{(46,64)} = G2(p_{(45,63)}, p_{(45,64)}, p_{(45,65)}, p_{(46,63)}, p_{(46,64)}, p_{(46,65)}, p_{(47,63)}, p_{(47,64)}, p_{(47,65)});$$

$$pt_{(47,63)} = G3(p_{(46,62)}, p_{(46,63)}, p_{(46,64)}, p_{(47,62)}, p_{(47,63)}, p_{(47,64)}, p_{(48,62)}, p_{(48,63)}, p_{(48,64)});$$

$$pt_{(47,64)} = G4(p_{(46,63)}, p_{(46,64)}, p_{(46,65)}, p_{(47,63)}, p_{(47,64)}, p_{(47,65)}, p_{(48,63)}, p_{(48,64)}, p_{(48,65)});$$

这里的 $pt_{(46,63)}$ 是 $p_{(46,63)}$ 在 RGB888 格式中映射出来的像素点, 而 G1 表示相应的转换算法。

2.3 降噪实现原理

本部分主要实现 RGB888 的降噪, 采用两条线来滤波, 结合图 1 及格式转换算法的表达式, 给出降噪模块主体部分的实现原理。如果需要降噪的点为 $pt_{(47,63)}$, 那么该点的降噪算法为:

$$pn_{(46,63)} = pt_{(46,63)} * 0.5 + pt_{(46,62)} * 0.21875 + pt_{(47,62)} * 0.03125 + pt_{(47,63)} * 0.15625 + pt_{(47,64)} * 0.09375$$

这里 $pn_{(46,63)}$ 表示对应于像素点 $pc_{(46,63)}$ 降噪后的数据。

3 图像预处理模块的硬件实现

本模块的实现如图 2 所示, 首先将抓取到的行数数据(Bayer color pattern), 按照 2.1 所提出的修正算法在第一个 5 级流水线中进行坏点处理, 并将正确数

据回写入 5 条行缓冲对应的位置, 延迟一定数量的时钟(比如 64 个时钟)后, 按照 2.2 所提出的方法在第二个 5 级流水线中进行 Bayer 到 RGB888 的转换, 同时按照 2.3 所提出的方法在同一个 5 级流水线中进行降噪, 即可得到所需的 RGB888 数据。

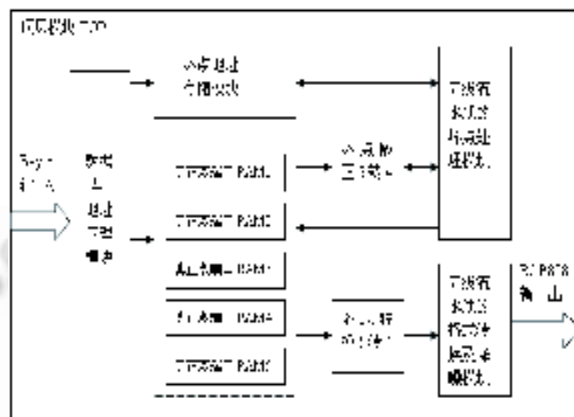


图 2 图像预处理模块的硬件实现原理图

4 图像预处理模块的仿真与验证原理

本模块的验证采用了灰盒测试法, 从模块中引出关键信息, 并以.txt 文本文件保存。图 3 给出了图像预处理模块 Testbench 示意图, 其实现分为三步:

- ①由 MATLAB 产生测试输入激励。
- ②测试输入激励分别输入图像预处理模块和 MATLAB 程序中, 可以分别得到大量的关键信息。
- ③比较得到的这两类关键信息, 如果一致, 则运用 MATLAB 函数显示图像; 如果不一致, 则检查错误, 直到两者的关键信息一致。

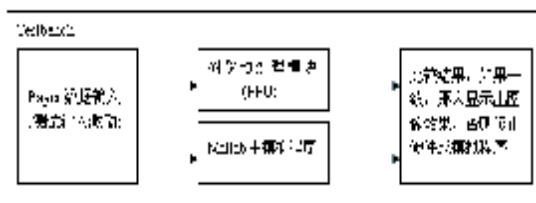


图 3 图像预处理模块 Testbench 示意图

5 仿真结果

本模块的软件开发工具是 Xilinx 的 ISE Foundation 10.1, 采用的综合工具是软件中自带的 XST(VHDL/Verilog), 采用的仿真工具是 Model

Tech 公司的 ModelSim XE III 6.3c, 以及美国 MathWorks 公司的 MATLAB R2007a, 采用的 FPGA 芯片是 Xilinx 公司的 Virtex5 系列 XC5VLX220(这里只作为开发用途)。图像预处理模块仿真时, 采用的 PIXCLK 是 71.4MHz。预处理后的效果如图 5 所示, 图 4 给出的坏点位置是一种极为苛刻的情况(左下方有一条刺眼的坏点线)。由两幅图的比较可知原始图像得到了很好的坏点修正及复原。



图 4 原始图像



图 5 对原始图像 Bayer 格式预处理后复原的图

6 结束语

本项目中, 对 Bayer 格式的图像处理, 由两个模块来完成。一个是图像预处理单元, 另一个是图像处理单元。图像处理单元完成的功能主要有 Contrast, Brightness, Saturation, Hue, Sharpness, Gamma 等。在图像预处理模块的仿真中, 我们所用的时钟频率远大于实际传感器(Micron MT9V112)在 Bayer 模式下的频率(13.5 MHz), 所以很容易满足当前图像预处理的时序性要求。下一步我们将从两个方向对预处理模块做出改进, 一方面把降噪处理方法由两条线滤波改为一条线滤波, 另一方面在图像预处理模块中采用寄存器控制图像格式, 使得图像预处理单元能够同时兼容 640x480(VGA), 352x288(CIF), 800x600(SVGA), 1280x1024(SXGA), 1280x720, 以及 1920x1080。

参考文献

- 1 阮秋琦. 数字图像处理学. 北京: 电子工业出版社, 2001. 212 - 229.
- 2 Dierichx B, Meynants G. Missing Pixel Correction Algorithm for Image Sensors. SPIE, 1998, 34(10): 200 - 203.
- 3 王金明. 数字系统设计与 Verilog HDL. 第三版, 北京: 电子工业出版社, 2009. 241 - 242.
- 4 云创工作室. Verilog HDL 程序设计与实践. 北京: 人民邮电出版社, 2009. 25 - 55.
- 5 乔庐峰. Verilog HDL 数字系统设计与验证. 北京: 电子工业出版社, 2009. 91 - 105.
- 6 刘秋云, 王佳. Verilog HDL 设计实践与指导. 北京: 机械工业出版社, 2005. 135 - 149.
- 7 袁文波, 张皓, 唐振中. FPGA 应用开发从实践到提高. 北京: 北京航空航天大学出版社, 2007. 88 - 98.
- 8 Mentor Graphics Corporation. ModelSim Se Datasheet. Mentor Graphics Corporation, 2004.
- 9 陈怀琛. 数字信号处理教程——MATLAB 释义与实现. 第 2 版, 北京: 电子工业出版社, 2008. 252 - 264