

文章编号:1007-2780(2012)03-0364-07

基于 FPGA 的 LCoS 显示驱动系统的设计与实现

马 飞¹,黄 蓓¹,赵博华¹,郝丽芳²,卢颖飞²,
杜 寰^{1*},韩郑生¹,林 斌²,倪旭翔²

(1. 中国科学院 微电子研究所,北京 100029,E-mail:mafei@ime.ac.cn;
2. 浙江大学 现代光学仪器国家重点实验室 国家光学仪器工程中心,浙江 杭州 310027)

摘 要:研究了硅基液晶(LCoS)场序彩色显示驱动系统的设计与实现。该系统以 FPGA 作为主控芯片,用两片高速 DDR2 SDRAM 作为帧图像存储器。通过对图像数据以帧为单位进行处理,系统将并行输入的红、绿、蓝数据转换成串行输出的红、绿、蓝单色子帧。将该驱动系统与投影光机配合,实现了分辨率为 800×600 的 LCoS 场序彩色显示。

关 键 词: LCoS;场序彩色显示;FPGA; DDR SDRAM

中图分类号: TN141.9 **文献标识码:** A **DOI:** 10.3788/YJYXS20122703.0364

Design and Implementation of LCoS Display Driver System Based on FPGA

MA Fei¹, HUAGN Ran¹, ZHAO Bo-hua¹, HAO Li-fang², LU Ying-fei²,
DU Huan^{1*}, HAN Zheng-sheng¹, LIN Bin², NI Xu-xiang²

(1. *Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China, E-mail:mafei@ime.ac.cn;*
2. *State Key Laboratory of Optical Instrument, Chinese National Engineering Center of Optical Instrument, Zhejiang University, Hangzhou 310027, China*)

Abstract: This paper introduces the design and implementation of LCoS field-sequential-colour (FSC) display driver system. The driver system uses FPGA as main controller, and uses two high speed DDR2 SDRAM to store image data. By means of regarding one frame image as a process unit, the driver system convert parallel input RGB signal into three series output homochromy image. By coordinating the driver system with the optical projector, FSC display on LCoS with the resolution of 800×600 is realized.

Key words: LCoS; FSC; FPGA; DDR SDRAM

1 引 言

硅基液晶(Liquid Crystal on Silicon,LCoS)显示器是一种微型反射式液晶显示器,其控制液晶的像素驱动电路用硅基 CMOS 制成。由于硅

集成电路工艺线宽在微米量级,所以可以把像素面积做的很小,使得 LCoS 芯片同时具有显示面积小和分辨率高的特点^[1]。

LCoS 用于彩色显示通常有两种方法,一种是空间混色法,另一种是时间混色法。空间混色

收稿日期: 2011-11-15; 修订日期: 2012-02-13

基金项目: 国家“863”计划资助项目(No. 2009AA01Z325)

作者简介: 马飞(1987—),男,山西原平人,硕士研究生,研究方向为微显示驱动技术研究、数字集成电路设计。

* 通讯联系人,E-mail:duhuan@ime.ac.cn

法有两种方式:一种是在显示芯片上对每个像素做出 3 个子像素,每个子像素分别显示红、绿、蓝不同的基色,这种混色方式控制简单,但是牺牲了显示屏的分辨率。另一种是使用 3 块微型显示屏,分别调制红、绿、蓝光源,然后再利用光学系统合成一幅彩色图像,这种混色方式缺点是成本较高^[2]。

时间混色法是把一帧彩色图像分成 3 个子帧,分别为红、绿、蓝三基色的图像,利用人眼的视觉惰性在人脑中还原成彩色图像。与空间混色法相比,时间混色法具有较高的显示面积利用率,但要求作为光阀的液晶响应速度较快,驱动电路频率较高^[3]。

针对自行研制的 LCoS 显示芯片,本文设计了相应的时序彩色显示驱动系统,可将输入的 VGA 信号转换成符合 LCoS 显示芯片接收协议的场序串行数字信号,并控制 LED 光源发出与单色图像相匹配的红、绿、蓝三基色脉冲光。经过电路仿真和板级调试,实现了时序彩色图像显示。

2 系统整体方案设计

驱动系统的硬件框架结构如图 1 所示,系统选用电脑的 VGA 接口作为信号源,经过视频解码芯片模数转换后输出分辨率为 800×600 的 SVGA 数字视频信号,包括像素时钟 CLK,帧同步信号 VS,行同步信号 HS,以及 R、G、B 各 8 位像素数据。

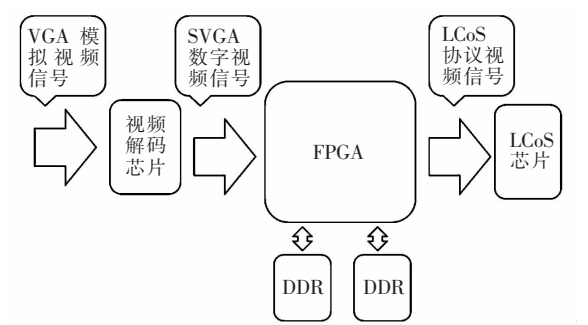


图 1 系统硬件结构框图

Fig. 1 Frame of the system hardware

核心控制模块选用 FPGA 来完成,FPGA 具有丰富的寄存器资源,适合实现复杂的时序逻辑电路。这里选用的 FPGA 芯片型号为 Altera 公司的 EP3C16F484C8,属于低端 Cyclone III 系列,内部资源有 15 408 个逻辑单元,516 kbit 的

RAM,4 个 PLL,346 个 I/O 端口,满足系统要求^[4]。SVGA 数字信号经过 FPGA 处理后,转换为符合 LCoS 芯片输入协议的视频信号。

存储器选用 DDR2 SDRAM,它具有容量大、吞吐速度快、成本低的特点,适合视频驱动系统使用。这里选用的 DDR 为 Miron 公司的 MT47H64M16,存储单元位宽为 16 bit,存储容量共 128 Mbit^[5]。一帧图像的数据量为 $800 \times 600 \times 24 \text{ bit} = 11.52 \text{ Mbit}$,所用 DDR 满足存储容量要求。由于 DDR 的读写操作复杂,时序要求严格,所以要设计 DDR 控制器来对 DDR 进行读写控制。

系统设计的基本思路是通过用存储器缓存的方法将输入的视频数据进行并串转换,并用两片 DDR 进行乒乓操作交替读写来保证数据流的无缝传输。具体实现方法是将输入的一帧数据存到一片 DDR 中,当下一帧到来时,把当前的数据存入另一片 DDR 中,同时将上一帧数据中相同颜色的数据串行读出,这样就把输入的一帧并行的彩色数据分成 3 个串行单色子帧输出。由于输入视频信号帧频为 60 Hz,所以输出的单色子帧帧频为 180 Hz。

3 FPGA 设计

FPGA 设计中存在跨时钟域数据传输的问题:输入的视频信号频率为 66.67 MHz,DDR 工作频率为 125 MHz,输出的 LCoS 视频信号频率为 50 MHz。为了解决这个问题,电路在 3 个时钟域间分别插入异步 FIFO,通过设置 FIFO 深度和空满标志位,保证数据的可靠传输。

FPGA 内部设计框图如图 2 所示,由 9 个基本模块构成,图中箭头方向标识了电路中数据流的方向。

该电路的基本运行过程是:

输入信号处理模块把 SVGA 视频信号中的

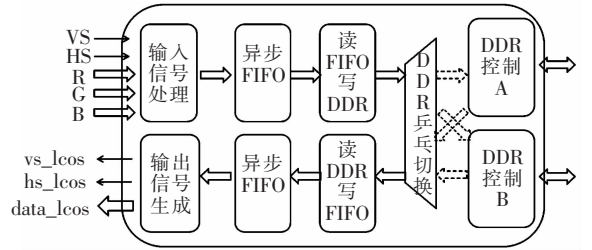


图 2 FPGA 内部设计框图

Fig. 2 FPGA internal design diagram

有效像素数据提取出来,并将 8 位像素数据中相邻的 4 个合并成 32 位写入异步 FIFO。后级模块将 FIFO 中的数据读出并生成写 DDR 地址和写有效信号,送入 DDR 乒乓切换模块。DDR 乒乓切换模块在帧同步信号 VS 的上升沿切换两片 DDR 的读写状态,选择一片 DDR 写入数据,并把写有效信号、写地址及数据送给相应的 DDR 控制器,同时通过另一个 DDR 控制器读出数据给“读 DDR 写 FIFO 模块”。“读 DDR 写 FIFO 模块”根据 FIFO 的将满信号适时将数据写入 FIFO。输出信号生成模块将 FIFO 中的数据连续读出,与附加的 LCoS 控制信号一起输出到 LCoS 芯片。

4 关键模块设计

4.1 异步 FIFO 设计

FIFO 的设计难点在于确定它的位宽、深度以及空满标志位^[6]。下面介绍输入信号处理模块和“读 FIFO 写 DDR 模块”间的异步 FIFO 设计。

4.1.1 FIFO 位宽的确定

由于 DDR 的双沿采样数据位宽为 16,而单沿采样位宽为双沿的 2 倍,所以“读 FIFO 写 DDR 模块”的数据宽度为 32。因此,为了便于后级模块处理,设置 FIFO 中存储器的数据宽度为 32。

4.1.2 FIFO 空满标志位

DDR 的读写是基于突发模式的,每次读写都会连续传输多个地址单元的数据。对于突发长度为 8,每次读写连续要传输 8 个 16 位数,即 4 个 32 位数。为了满足写 DDR 的数据量要求,每读一次 FIFO 要能够连续读出 4 个 32 位单色数据,所以单色数据必须连续存放。因此在单个 FIFO 中放置了 3 个双口 RAM,分别存放红绿蓝三色数据,如图 3 所示。

因为输入的视频数据一直要写入 FIFO,不能停顿,所以无需设定写满标志位,并且必须保证读出数据的速率比写入快,使 FIFO 始终处于未

满状态。

写信号 wr_fifo 有效时,3 个 RAM 的地址指针同时各加 1,在相同的写地址写入相应的单色数据。设置输出允许读标志位 data_ready,它在写指针 wp 大于等于读指针 rp+4 时有效,通知后级模块可以连续读出 4 个 32 位数。异步 FIFO

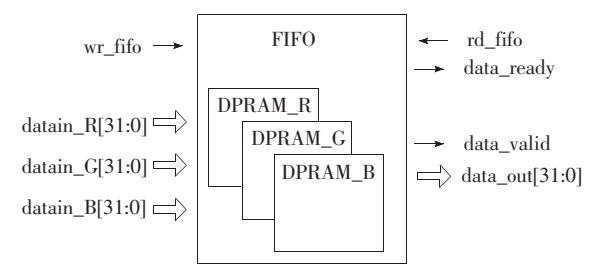


图 3 异步 FIFO 模块示意图
Fig. 3 Asynchronous FIFO module

在 rd_fifo 有效时,开始依次读出 3 个 RAM 中的数据,其控制状态机流程图如图 4 所示。

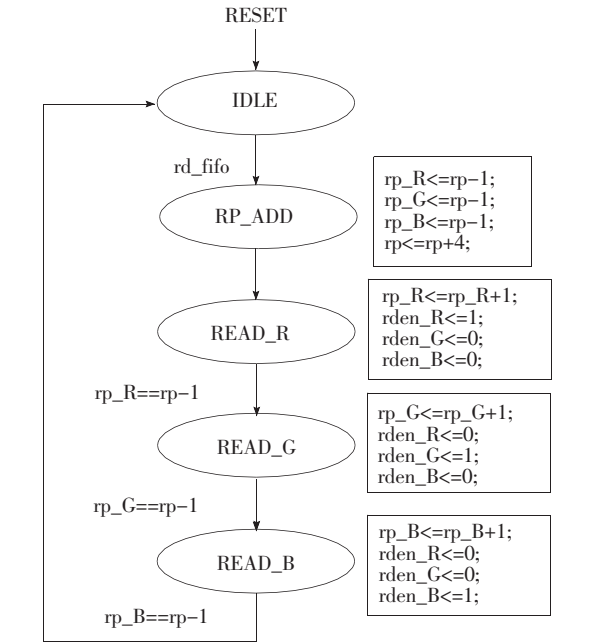


图 4 读 FIFO 状态机流程图
Fig. 4 Flow chart of reading

设读指针为 rp,红绿蓝三色 RAM 的指针分别为 rp_R,rp_G,rp_B。复位后状态机进入 IDLE 状态,当读 fifo 信号有效后状态进入 RP_ADD,当前 rp 的值减去 1 赋给三色 RAM 指针,同时 rp 加 4。之后进入 READ_R 状态,rp_R 加 1,读出 4 个 32 位红色数据。当 rp_R 与 rp-1 相等时,rp_R 已经加了 4 次,红色数据已读完,就跳到状态 READ_G 以同样的方法连续读出 4 个 32 位绿色数据,读完蓝色数据后跳回到 IDLE 状态,这样每读一次 FIFO 就能连续读出 12 个 32 位数据。状态机中的关键代码在图 4 中右侧标出。

4.1.3 FIFO 深度的确定

FIFO 深度要根据 FIFO 的输入输出数据率

来计算。FIFO 的输入数据率为每 240 ns 写入 12 个 32 数。输出数据的时钟周期 $T=8\text{ ns}$,连续读出 12 个 32 位数需要 $12T$,再加上握手通信时间 $2T$,FIFO 传递地址的格雷码转换延迟 $4T+15\text{ ns}$,共读出 12 个 32 位数需要的时间为 $18T+15\text{ ns}=159\text{ ns}<240\text{ ns}$ 。所以输入的数据率小于输出的数据率,输入的视频数据流可以不停顿,符合设计预期。

按照前面的计算,读出比写入快,故最小深度可设为 8,就能保证写 FIFO 不溢出。然而 DDR 在隔 7 812 ns 就要刷新一次,这时读出数据的时间需要加上刷新所用的时间。刷新一次所用的时间为 $20T=160\text{ ns}$,所以总的读出时间为 319 ns,比写入时间长。如果 FIFO 深度为 8 就会有溢出,所以要增加 FIFO 深度来缓存未读出的数据。

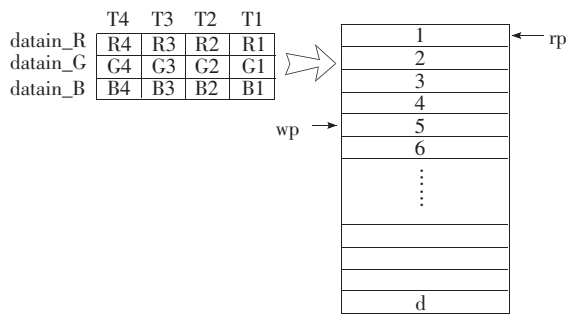


图 5 写 FIFO 指针变化示意图
Fig. 5 Sketch map of writing FIFO

设 FIFO 深度为 d ,如图 5 所示。开始写 FIFO 后,当 FIFO 写入 12 个 32 位的三色数据,写指针从 1 加到 5,比读指针 rp 大 4,此时 $data_ready$ 变为高电平,通知后级模块可以读 FIFO。如果此时 DDR 要刷新,则必须先等待刷新完成,再连续读出 12 个 32 位数据。在刷新期间,写指

针一直在增加,但最大不得超过 FIFO 深度 d 。因此可以得出 FIFO 深度与刷新完成时间关系为:刷新完成时间+读出 12 个 32 位数的时间 $\leq (d-4)\times 60\text{ ns}$,通过计算得出 $d\geq 319/60+4=9.3$ 。考虑到 FIFO 内部用格雷码传递地址,深度应为 2 的整数次幂,故 FIFO 最小深度可设为 16。

4.2 DDR 控制器设计

4.2.1 DDR 存储方案设计

DDR 的存储结构分为三层——Bank、Row、Column。系统所用 DDR 中含有 8 个 Bank,每个 Bank 中有 8 192 个 Row,每个 Row 中有 1 024 个 Column。

一帧图像数据在 DDR 中的存储安排为:将红绿蓝三色图像数据分别写入 3 个 Bank 中,从 Row0、Col0 开始顺序存放,存满一行后换下一行继续存入。每次写操作要把三色数据各 4 个 32 位分别写入 3 个 Bank 中。读出时依次把每个 Bank 中的单色数据读完,每次读操作读出 4 个 32 位单色数据。

对 DDR 读或写的操作流程为:

- (1)发出激活 (Active) 命令,同时给出 Bank 地址和 Row 地址。
- (2)经过 t_{RCD} 时间后,发出读或写命令,同时给出 Column 地址。此时令 A10 为 1 可在读写完毕后自动预充电 (Auto Precharge)。
- (3)换 Row 读写时要先对当前行预充电 (Precharge),之后再对新的 Row 进行激活操作。

由于每次读写都要包括行激活和预充电,因此为了提高传输效率,就要设法把行激活和预充电占用读写的时间尽量减少。本文采用交错 Bank 写入数据的方法,在写入当前 Bank 时对另一个 Bank 进行激活或预充电操作,从而提高了数据总线利用效率^[7],如图 6 所示。

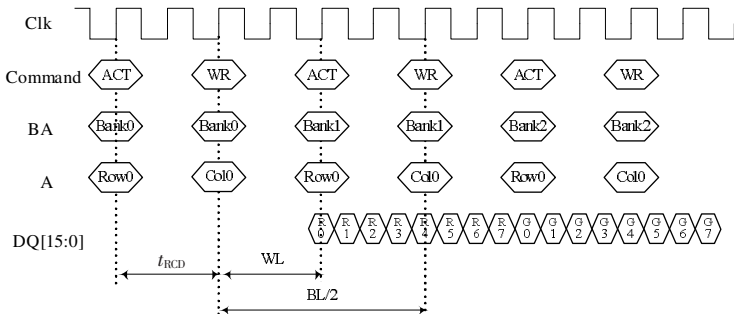


图 6 交错 Bank 写操作示意图
Fig. 6 Sketch map of crisscross bank writing

将红绿蓝三色各 8 个 16 位数据分别写入 Bank0、Bank1、Bank2 中,起始写地址都为 Row0, Col0。在 ACT 发出后,写命令 WR 应在其 t_{RCD} 后发出。 t_{RCD} 为 15 ns,对于频率为 125 MHz 的时钟,相当于间隔 2 个时钟周期。发出写命令 WR 的同时令 A10 为高电平,则 DDR 会自动完成预充电操作。在 WR 发出后 WL 个时钟周期,写数据通过 DQ 输出,此处 WL 设为 2。如果下一个写命令在前一个写命令的 BL/2=4 个周期后发出,则两次突发写入的数据能够连接起来,这样下一个 ACT 应在 WR 后 2 个周期发出。所以按照图 5 中的操作,可以将红绿蓝三色的各 4 个 32 位数据连续写入 3 个 Bank 中相同的地址,并且仅用了一次写操作的准备时间,提高写入效率。

4.2.2 DDR 控制器设计

DDR 控制器主要由 5 个模块构成:时钟产生模块、初始化模块、刷新模块、控制模块和数据通路模块^[8],如图 7 所示。

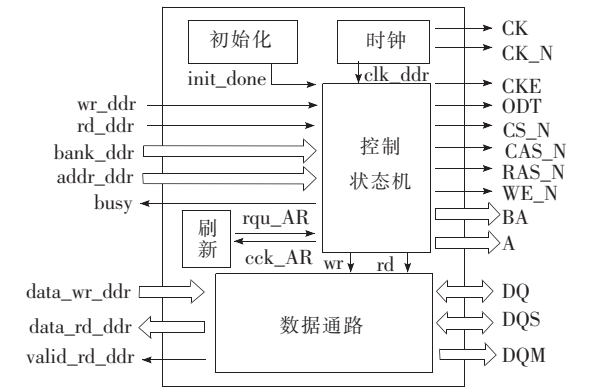


图 7 DDR 控制器总体设计框图

Fig. 7 Block diagram of DDR controller

为了避免在读写过程中受到刷新或其他读写的中断影响,控制器设置了握手信号 busy。当刷新或正在读写时,busy 为 1,否则为 0。每次对 DDR 进行读写前要查询 busy 信号是否为高有效,如果为高电平,则要等到其变为低才能进行读写。

控制状态机是整个 DDR 控制器的核心部分,其状态转移流程如图 8 所示。

状态机的输入信号有:初始化完成标志位 init_done,刷新计数器的刷新请求信号 rqu_AR,写 DDR 信号 wr_ddr,读 DDR 信号 rd_ddr,均为高电平有效。

上电复位开始后,状态机停留在 DDR_WAIT

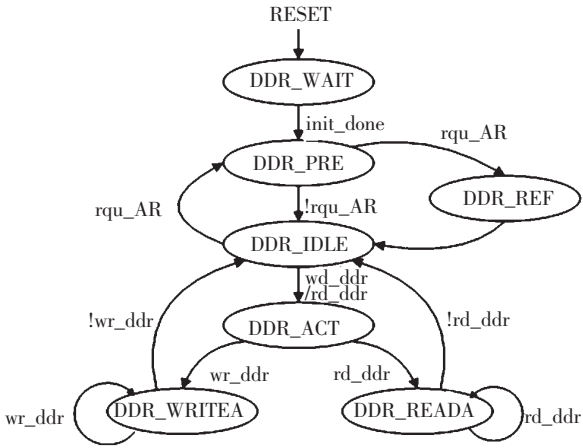


图 8 DDR 控制状态机流程图

Fig. 8 Flow chart of DDR control state machine

状态。等初始化完成,init_done 变为高电平后,状态机进入 DDR_PRE 状态,对所有 Bank 预充电。如果此时无刷新请求则直接进入 IDLE 状态。如果刷新计数器发来刷新请求信号 rqu_AR,则进行预充电并刷新。刷新完毕,对刷新计数器发出应答信号并回到 IDLE 状态。

要对 DDR 进行读写操作,就对 DDR 控制器发出 rd_ddr 信号或 wr_ddr 信号。状态机接收到读写信号后先进入 DDR_ACT 状态,对 DDR 进行激活操作,等待 t_{RCD} 后,根据 rd_ddr 或 wr_ddr 进入 DDR_READA 状态或 DDR_WRITEA 状态,对 DDR 进行读写并自动预充电。写入时每次写入 12 个 32 位数,采用交错 Bank 的方式连续发出 3 个写命令。读出时每次仅发出 1 个读命令,读出 4 个 32 位数。读写完毕后状态机返回 DDR_IDLE 状态。如果在读写期间刷新计数器发来刷新请求,则不予处理,直到读写操作完成后返回到 DDR_IDLE 状态再进行刷新。通过这样的状态转移,能够完成对 DDR 的读写控制。

5 结果与分析

5.1 电路仿真结果

整个驱动电路采用 RTL 级的 Verilog 实现,验证是在 Modelsim 环境中进行,DDR 的仿真模型采用 Micron 公司给出的器件仿真模型。

5.1.1 FIFO 的仿真结果

FIFO 的读写仿真波形图如图 9 所示,将连续 4 个红绿蓝三色并行数据送入 FIFO 后,data_ready 为 1,此时发出读 FIFO 信号,将存入 FIFO

的 12 个数据串行读出,同时在有效数据出现时,数据有效信号 valid_data 为高电平。从波形图看出电路的仿真测试结果符合设计要求。

5.1.2 DDR 控制器的仿真结果

图 10、图 11、图 12 为所设计的 DDR 控制器读写操作仿真波形图。令所用的两片 DDR 标号分别为 DDR0、DDR1,图中后缀为 0 表示 DDR0 的控制器的相关信号,后缀为 1 表示对 DDR1 的控制器的相关信号。图 10 为写操作的输入信号,当 busy0

为 0 时,发出写操作,每次写操作连续写入 12 个 32 位数。DDR 控制器通过切换 Bank 将 aaaa0001-aaaa0004、bbbb0001-bbbb0004、cccc0001-cccc0004 分别写入 Bank0、Bank1、Bank2,如图 11 所示。

图 12 为读操作波形图,当 busy1 为低电平时,发出读 DDR 信号以及读地址,经过 12 个时钟周期后,从 DDR 读出的数据出现在数据总线上(CL 设置为 3),同时数据有效信号变为高电平。

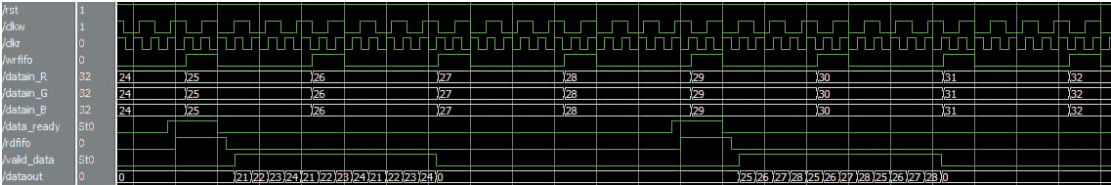


图 9 FIFO 读写仿真波形图
Fig. 9 Simulation wave figures of FIFO reading and writing

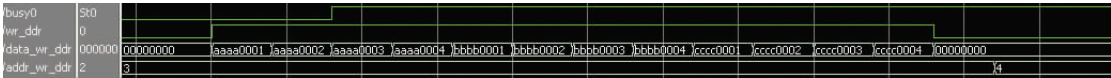


图 10 DDR 控制器写操作输入信号仿真波形图
Fig. 10 DDR controller for writing operation simulation input signal

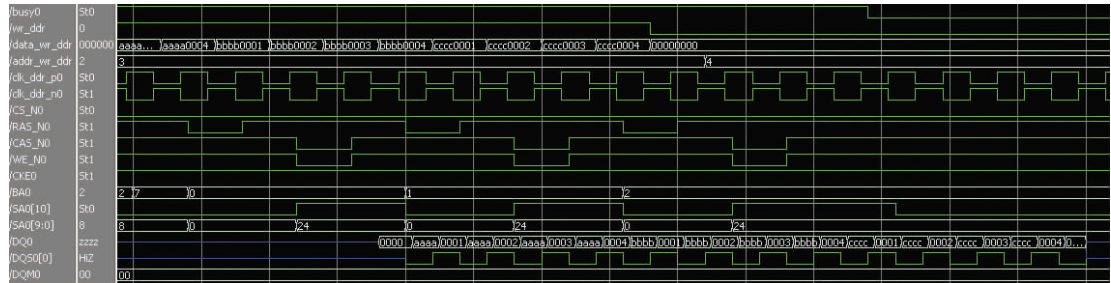


图 11 DDR 控制器写操作仿真波形图
Fig. 11 DDR controller for writing operation simulation waveform

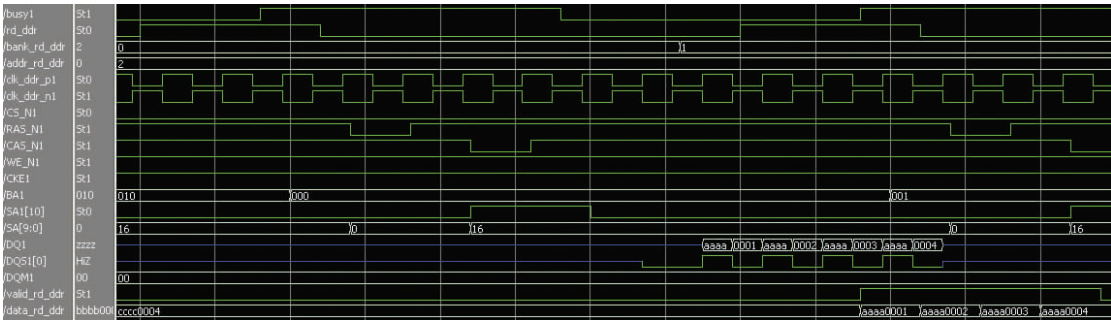


图 12 DDR 控制器读操作仿真波形图
Fig. 12 DDR controller for reading operation simulation waveform

5.2 整机显示结果

实际的视频驱动系统如图 13 所示,由两块电路板组成:一片是视频解码电路板,放置了视频解

码芯片和 VGA 接口;另一片是视频驱动电路板,主要由 FPGA 芯片和两片 DDR 存储器组成,该电路板的 PCB 版图如图 14 所示。

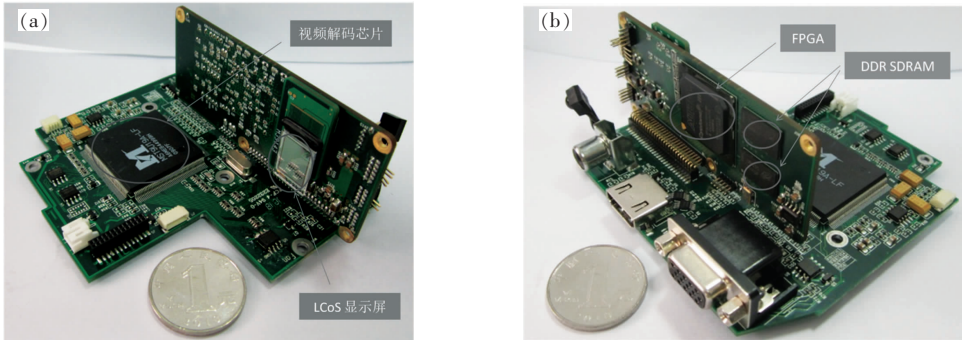


图 13 (a)视频驱动系统正面;(b)视频驱动系统背面。
Fig. 13 (a)The obverse side of video driver, (b)The reverse side of video driver.

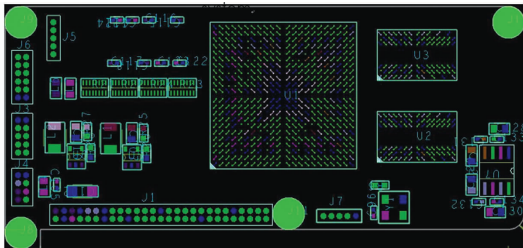


图 14 视频驱动电路板的 PCB 版图
Fig. 14 PCB layout of video driver system



图 15 实际显示效果
Fig. 15 Actual display effect

在显示图像时,将电脑输出的分辨率设置为 800×600 ,帧频调为 60 Hz,通过 VGA 接口与视频解码电路板对接,即可实时显示电脑的输出图像,显示效果如图 15 所示。结果表明图像显示线条正确,画面清晰稳定,无拖影。

6 结 论

设计并实现了分辨率为 800×600 ,子帧频为 180 Hz 的 LCoS 场序彩色驱动系统。系统以 FPGA 作为主控芯片,用两片 DDR SDRAM 作为帧图像存储器。在 FPGA 设计中,对于不同的时钟域插入了异步 FIFO 解决了跨时钟域数据传输问题,并就异步 FIFO 参数的设置进行了讨论。此外,本文根据实际电路需求设计了专用的 DDR 读写控制器,完成了对 DDR 的数据存取。用该驱动系统与投影光机配合,实现了 LCoS 场序彩色显示。

参 考 文 献:

[1] 耿卫东,代永平,任立儒,等. LCoS 场序彩色显示控制器的设计 [J]. 液晶与显示,2003,18(3):188-192.
[2] 宋丹娜,代永平,刘艳艳. LCoS 接口专用集成电路的设计 [J]. 液晶与显示,2009,24(5):670-674.
[3] 代永平. LCoS(硅基液晶)显示器设计 [D]. 天津:南开大学,2003.
[4] ALTERA corporation. Cyclone III Design Guidelines [EB/OL]. [2008-11-20]. <http://www.altera.com.cn/literature/lit-cyc3.jsp>.
[5] Micron Technology. MT47H64M16HR Handbook [EB/OL]. [2011-10-15]. <http://www.micron.com/parts/dram/ddr2-sdram/mt47h64m16h4-25e?pc={7BBF7165-1B7B-4D9E-B35A-5D4412F647FD}>.
[6] 苏进. 异步 FIFO 控制器的设计 [D]. 合肥:合肥工业大学,2007.
[7] 范泽明. DDR2 SDRAM 控制器的设计与验证 [D]. 西安:西安电子科技大学,2009.
[8] 朱炜,刘新宁. DDR SDRAM 控制器的设计与实现 [J]. 电子器件,2009,32(3):592-595,600.