

文章编号:1007-2780(2013)01-0120-07

基于 SOPC 技术的 VGA 字符和图像显示系统

姜 漫^{1,2}, 吴志勇^{1*}, 曹 腾³

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;
2. 中国科学院大学, 北京 100049; 3. 浙江大学 信息与通信工程研究所, 浙江 杭州 310027)

摘 要:为解决 VGA 显示系统因基于计算机或专用芯片而不易携带和更改的局限性,采用可实现用户自定义内核功能的 SOPC 技术,完成 PS/2 键盘控制下 VGA 字符和图像显示系统设计。在 FPGA 中构建 32 位 NIOS II 内核,C 语言编写主程序,采用 AVALON 总线连接外设驱动控制器和基本 PIO 接口,连同 VGA 时序模块与缓存器组成的 VGA 控制器,实现系统软件;基本硬件电路、外部存储芯片、PS/2 键盘、VGA 接口和串口电路构成硬件平台。系统最终实现 SOPC 内核移植和在 PS/2 键盘控制下由 CRT 显示器显示字符和图片的功能。

关 键 词: SOPC; NIOS II; VGA; PS/2; SDRAM

中图分类号: TN911.73 **文献标识码:** A **DOI:** 10.3788/YJYXS20132801.0120

Characters and Images VGA Display System Based on SOPC Technology

JIANG Man^{1,2}, WU Zhi-yong^{1*}, CAO Teng³

(1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China;
3. Institute of Information and Communication Engineering, Zhejiang University, Hangzhou 310027, China)

Abstract: To solve the localization that VGA display system based on computer or ASIC lacks portability and modification, a novel VGA display system to display any characters and images was designed under PS/2 keyboard control, which is based on user custom kernel SOPC technology. In FPGA, a 32-bits NIOS II hardcore was built, main program was compiled with C language, and a AVALON BUS was used to connect the external drive controller and basic PIO interface, combined with VGA controller which composed by VGA timing module and buffers, the software part was completed through the above. Hardware platform is consisted of basic hardware circuit, external memory chips, PS/2 keyboard, VGA interface and UART circuit. The system finally achieved the migration of SOPC kernel and function of displaying character and image on CRT monitor under PS/2 keyboard control.

Key words: SOPC; NIOS II; VGA; PS/2; SDRAM

1 引 言

随着科技的发展进步,片上可编程系统(Sys-

tem on Programmable Chip, SOPC)^[1-2]技术作为嵌入式的一种,被广泛应用到各个领域,依附现场可编程门阵列(Field Programmable Gate Array,

收稿日期:2012-05-14; 修订日期:2012-10-08

基金项目:国家 863 高技术研究发展计划资助项目(No. 2008AA7034320B)

作者简介:姜漫(1987—),女,黑龙江齐齐哈尔人,硕士研究生,主要研究方向:高速光纤通信系统研究。

* 通信联系人, E-mail: wuzy@ciomp. ac. cn

FPGA)^[1-2]实现,将处理器、存储器、I/O 口等系统设计需要的功能模块集成到一个可编程器件上,构成可编程的片上系统,由单个芯片完成整个系统的主要逻辑功能,具有灵活的设计方式,可裁减、可扩充、可升级,并具备软硬件在系统可编程功能。视频图像阵列^[3](Video Graphics Array, VGA)是 IBM 在 1987 年随 PS/2 一起推出的一种视频传输标准,具有分辨率高、显示速率快、颜色丰富等优点,在彩色显示器领域得到广泛应用。目前 VGA 显示控制方法增多,但主要是基于带有 VGA 显示卡的计算机等大型设备上,设计复杂,成本高;在一些既需要显示彩色高分辨率图像又不需要计算机控制场合,受到一定局限性。

近几年有关 VGA 显示的研究论文主要基于 FPGA 实现^[4-7],部分采用 SOPC 技术实现^[3,8],但系统只显示汉字或者捕获显示图像,本文设计了基于 SOPC 技术的 VGA 显示控制系统,加入 PS/2 键盘控制,实现了在不使用 VGA 显示卡情况下切换显示字符和图像的功能,解决了 VGA 显示控制局限于 PC 机或 VGA 专用芯片不易携带和改动的问题,采用 Avalon 总线控制^[9],各功能模块下载到一片 FPGA 中,具有设计可靠灵活,硬件尺寸小的优点,应用前景广泛。

2 系统设计

为实现系统显示功能,设计使用 PS/2 键盘作为控制器和字符数据的输入设备,系统上电后选择显示模式。字符显示时 NIOS II 内核直接获取 PS/2 键值并发送到双端口 SRAM 缓存;图像显示时 NIOS II 内核不断通过串口获取上位机数据由 SDRAM 以字节数组的形式缓存(每次缓存一幅图像数据)。最后 VGA 显示控制器产生时序和 RGB 信号驱动 VGA 终端显示。采用图 1 所示设计方案,分为硬件电路设计和 SOPC 模块设计。

硬件电路包括:主控芯片(Altera 公司 Cyclone 系列 FPGA EP1C12Q240C8)^[10],PS/2 键盘作为控制命令和字符数据输入,CRT 显示器输出,SDRAM、Flash 数据和程序存储器,保证系统正常工作的 FPGA 最小系统电路。以及 C# 编写的上位机作为图像数据输入。

SOPC 模块设计包括:在 FPGA 内部构建的 NIOS II 内核和 verilog 编写 VGA 显示控制器。

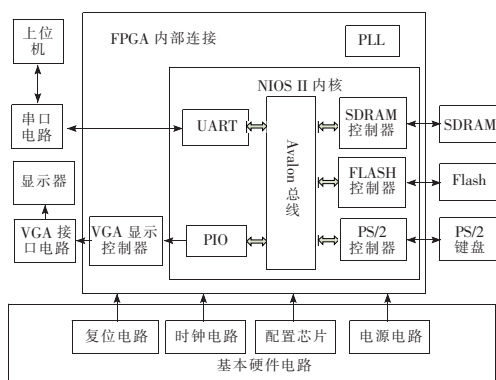


图 1 系统整体结构

Fig. 1 Structure of the overall system

系统以 NIOS II 嵌入式内核为设计核心,C 语言编写内核程序,采用 Avalon 总线,添加 CFI 三态桥控制器进行仲裁,驱动 SDRAM 和 Flash 控制器,添加 UART 和 PS/2 控制器与外部硬件通信,PIO 接口与 VGA 逻辑通信。

3 硬件平台构建

3.1 基本硬件电路

复位电路:使用一个电压管理集成控制芯片 TL7705,监视自身 SENSE 引脚电压,上电时,当 V_{CC} 达到 1 V 左右,复位信号使能(低电平有效), V_{CC} 接近 3 V 时,延时一段时间,RESET 和 RESET# 输出无效,正常工作时,供电电压过低,RESET 和 RESET# 输出有效。

时钟电路:为系统提供多个不同频率时钟,以确保各模块正常工作,使用 PI6C103 芯片,14.318 MHz 的晶振为内部振荡器,通过电容电阻匹配,设计实现了以下时钟:PCI 时钟(33.33 MHz)、USB 时钟(48 MHz)、REFERENCE 时钟(14.318 MHz)、CPU 时钟(100 MHz/66.66 MHz)。

配置芯片:针对 FPGA 断电后程序丢失的特点,使用 FLASH 对程序进行存储、串行接口芯片 EPCS4 为配置芯片,为 FPGA 储存结构数据,保存基本配置,下一次上电或重启时,将所储存的数据再次发送到 FPGA 芯片上。

电源电路:系统中 FPGA 芯片、PS/2 键盘和 VGA 接口使用 +5 V 电源供电,存储器设备芯片均使用 3.3 V 电源。系统采用带负载最大可以输出 3 A 电流的三端稳压芯片 LD1085,该芯片可以提供两种电压。

3.2 外部硬件电路

PS/2 键盘:通过一个 6 针连接器与系统通信,在时钟和数据线上是开路,需大阻值上拉电阻。平时保持高电平,有输出时拉低,之后自动上浮到高电平。

串口电路:上位机采用 RS-232 串口标准,使用美信公司 MAX3243 芯片完成逻辑电平转换。

VGA 接口^[4]:接收来自 FPGA 的数字信号,转换成可驱动 VGA 显示的模拟信号。这些信号是与 TTL 逻辑电平兼容的行同步信号、场同步信号;红、黄、蓝三种 RGB 模拟信号。

SDRAM:作为系统图像数据的缓存器,使用 ISSI 系列的 IS42S16400B7T 芯片,是一个 8 Mbyte 同步接口高速动态存储器,其内部是一个存储阵列,具有 64 Mbits 容量。同步动态存储器采用流水线结构传输数据来提高速度。

Flash:使用与 Cyclone 器件连接的容量为 2 Mbyte 芯片 ES29LV160DB70RTC。深度 16 Mbits,3.0 V 可擦除可编程闪存,其功能命令包含读取程序和擦除操作,微处理器与接口易连接。

4 基于 SOPC 技术的软件实现

4.1 NIOS II 内核建立

4.1.1 NIOS II 工程建立

在 Quartus II 环境中创建工程,使用 SOPC Builder 选择 32 bits 内核,确定偏置地址,添加 JTAG UART 和 ON CHIP MEMORY 生成一个 NIOS II 的 Symbol,工程中保存为 *.bif 文件,再次打开 QUARTUS II 软件,建立 Symbol 的工程,编译后下载到 FPGA 的芯片上,NIOS II 嵌入式硬件内核构建成功。添加需要的外设,绑定 FPGA

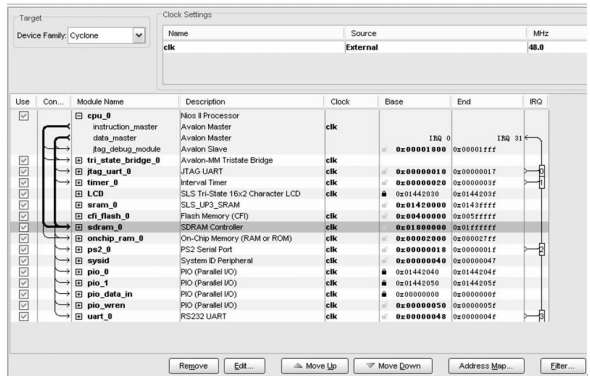


图 2 NIOS II 系统软件配置

Fig. 2 NIOS II software system configuration

引脚,完成 Quartus II 工程的配置,生成 NIOS II 顶层 BDF 文件^[1-2]。系统中 NIOS II 软件配置如图 2。

4.1.2 NIOS II 外设添加及接口设计

PS/2 控制器:控制系统显示模式和内容的功能,提供 ps2_clk 和 ps2_data 两个接口。PS/2 键盘作为从设备使用,向主机传送同步时钟信号和键盘数据信号。控制器和键盘遵循 PS/2 键盘的全双工通信时序。

UART:与外部串口硬件电路通信,接收上位机的图像数据并写入 SDRAM 中,存储数据后等待读取。提供 RXD 和 TXD 两个信号线。

PIO 接口:完成 NIOS II 内核与 VGA 显示控制器的通信。系统设置 4 个 PIO 接口:VGA 控制器中缓存的写地址 wraddress[9:0]、数据传输信号线 data_in[7:0]、使能信号 wren 和调试用 Led[3:0]。

SDRAM 控制器^[10]:自动完成 SDRAM 的初始化、自动定时刷新、存储单元地址映射管理的功能。同样基于 Avalon 接口,SOPC Builder 提供了现场的 SDRAM 控制器核,支持 PCI00 标准中的所有 SDRAM,NIOS II 通过该控制器使用片外 SDRAM,扩大了存储空间,控制器触发时钟由 PLL 锁相环产生,通过时钟接口、地址线、数据线等实现 SDRAM 的读写操作。

图 3 为 SDRAM 接口图 and 状态图,SDRAM

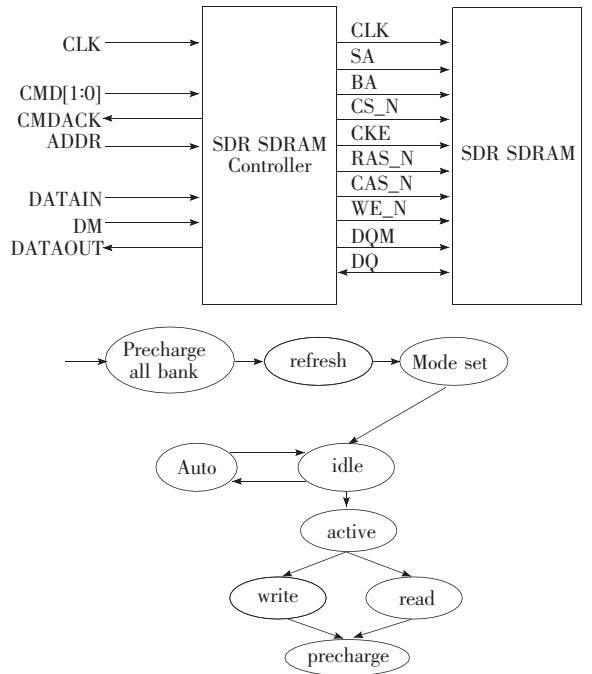


图 3 SDRAM 接口图 and 状态图

Fig. 3 SDRAM interface and state diagram

控制器对 SDRAM 进行操作的状态图, 实现 SDRAM 信息存储。首先系统所有逻辑 bank 进入预充电状态, 进行自刷新后开始模块设置, 之后进入空闲状态并不断自动刷新 AR, 保证数据被存储, 若空闲状态变为激活状态, 根据外界信号确定执行读或写操作, 操作后再次进入预充电状态。

CFI 控制器: 一是实现 NIOS II 内核通过 Flash Programmer 下载到片外 Flash 中; 二是在系统上电初始化时, 将 Flash 中程序写回 SDRAM。提供 DQ0~DQ14, 19 位的 DATA 线, RY/BY, WE#, CE#, BYTE#, OE#, RESET# 等, 连接示意图如图 4 所示接口。

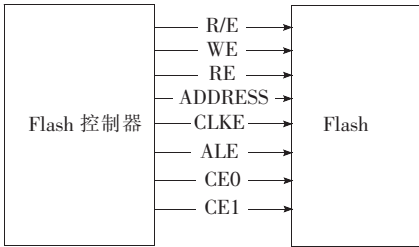


图 4 CIF 控制器与 Flash 连接关系图
Fig. 4 CIF controller with flash connection diagram

4.2 VGA 显示控制器设计

4.2.1 VGA 显示原理

标准的 VGA 画面尺寸是 640×480 像素、大约每秒 60 帧、像素时钟最高 25.175 MHz。VGA 显示器最终显示的一帧图像的数据是一个 640 行 480 列的像素点的数组^[8]。

显示一帧数据的细节是: VGA 控制器先从其所拥有的一组显存中读取一行数据, 将其发送到显示器响应的像素点上, 下一行与之相同。前一帧的所有行都显示结束后才会开始显示下一帧的图像。每秒钟 VGA 显示器显示的图像的帧数就是刷新率。当刷新率足够高时, 人眼会认为图像是连续的, 而不是一行接一行地被显示出来。

对于 VGA 显示器, 输出线有 5 个信号: R、G、B 三基色信号, HS 行同步信号, VS 场同步信号^[11-12]。信号时序需遵守“VGA 工业标准” $640 \times 480 \times 60\text{Hz}$ 。图 5 所示是 VGA 行扫描和场扫描时序^[4]。

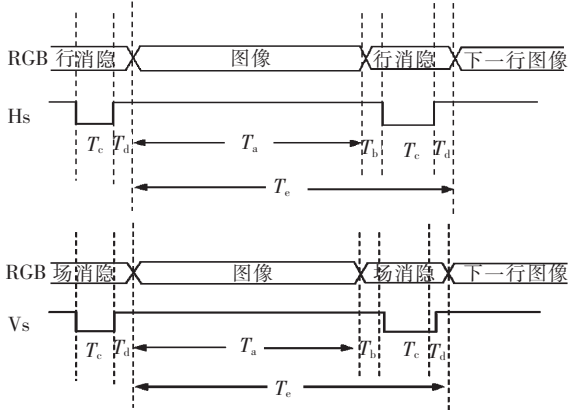


图 5 行扫描、场扫描时序图
Fig. 5 Timing diagram of line and field scan

表 1、表 2 为其时序参数, T_a 是行扫描的开头, T_d 是行图像, T_g 是行循环。

表 1 行扫描时序要求(单位: 像素)

Table 1 Timing requirements of the line scan (unit: pixel)

	行同步	行消隐后肩	行图像	行消隐前肩	行周期
对应位置	T_c	T_d	T_a	T_b	T_c
像素	656~751 (96 个像素)	752~799 (48 个像素)	0~639 (640 个像素)	640~655 (16 个像素)	800

表 2 场扫描时序要求(单位: 行)

Table 2 Field scanning timing requirements (unit: line)

	场同步	场消隐后肩	场图像	场消隐前肩	场周期
对应位置	T_c	T_d	T_a	T_b	T_c
行数	490~491 (2 行)	492~524 (33 行)	0~479 (480 行)	480~489 (10 行)	800

4.2.2 VGA 显示控制器实现

设计 VGA 显示控制器如图 6 所示。缓存器

中 SRAM 缓存字符数据, 在其中构建 .mif 文件, 对相应的行列地址进行像素点编码, 作为数据查

找表,用于查找颜色信息,将接收到的像素数据转换成 RGB 信息输出;SDRAM 缓存上位机载入的图像数据。最后由 VGA 时序生成器直接读取显示。

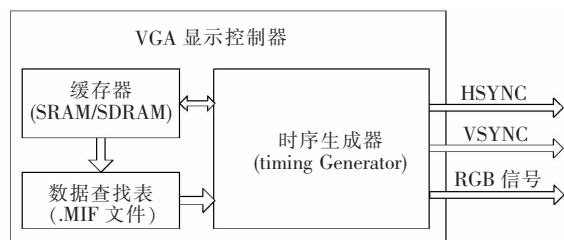


图 6 VGA 显示控制器框图

Fig. 6 VGA display controller block diagram

根据 VGA 显示原理和时序,使用 Verilog 语言编写 VGA 时序生成器,根据表 1、表 2 所示行、场有效时间和消隐的节点,定义常量 $H_pixels_across = 640$; $H_sync_low = 664$; $H_sync_high = 760$; $H_end_count = 800$; $V_pixels_down = 480$; $V_sync_low = 491$; $V_sync_high = 493$; $V_end_count = 525$; 以 $pixel_clock$ 为时钟,使用计数器,在对应节点控制 HS 和 VS 产生图 5 所示高低电平变换,输出 HS 和 VS 信号;向显存发送读地址数据,从 .mif 文件或 SDRAM 缓存器获取 RGB 基本信息,跟变量 $video_on_int$ 进行 && 操作后作为 VGA 显示器的三基色信号输出。程序为:

```
assign video_on_int = video_on_h && video_on_v;
assign pixel_clock = pixel_clock_int;
assign video_on = video_on_int;
assign horiz_sync_out = horiz_sync;
assign vert_sync_out = vert_sync;
assign red_out = red && video_on_int;
assign green_out = green && video_on_int;
assign blue_out = blue && video_on_int;
always @(posedge pixel_clock_int)
begin
    if (h_count == H_end_count)
        h_count <= 0;
    else h_count <= h_count + 1;
    if (h_count <= H_sync_high &&
        h_count >= H_sync_low).....
        horiz_sync <= 0;
```

```
else horiz_sync <= 1;
if (v_count >= V_end_count &&
    h_count >= H_sync_low)
    v_count <= 0;
else if (h_count == H_sync_low)
    v_count <= v_count + 1;
if (v_count <= V_sync_high &&
    v_count >= V_sync_low)
    vert_sync <= 0;
else vert_sync <= 1;
if (h_count < H_pixels_across)
begin
    video_on_h <= 1;
    pixel_column <= h_count;
end
else video_on_h <= 0;
if (v_count <= V_pixels_down)
begin
    video_on_v <= 1;
    pixel_row <= v_count;
end
else video_on_v <= 0;
```

end

endmodule

VGA 控制器与 NIOS II 内核工作关系如图 7 所示, VGA 时序发生器发送地址信号到 NIOS II 内核获取显存 SRAM 中 PS/2 键值或 SDRAM 图像对应的行列数据进行显示。

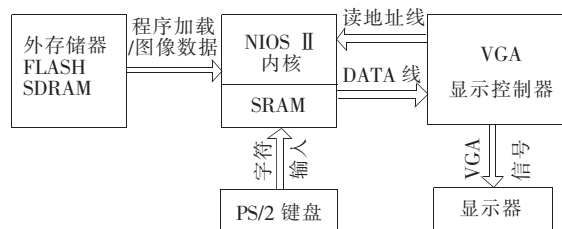


图 7 VGA 控制器工作示意图

Fig. 7 Working diagram of the VGA controller

5 系统实现及结果

5.1 系统功能实现

一是字符显示,使用 NIOS II 的 PIO 接口、PS/2 键盘控制器和 VGA 显示控制器实现;二是图像显示,使用 SDRAM、PS/2 键盘、串口和 VGA

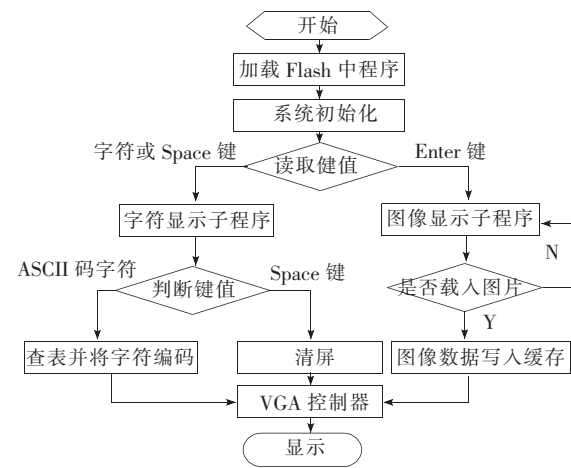


图 8 主程序流程图

Fig. 8 Main program flow chart

显示控制器实现。实现流程如图 8。

FLASH 存储器存储系统信息, 上电后将 FLASH 中的 C 程序写入内核, 完成系统初始化, 判断键值, 为 ASCII 码字符或 Space 键进入字符显示, Enter 键进入图像显示。之后由两个子程序通过缓存器存储二进制数据, VGA 控制器读取后显示。代码如下:

```
void main_loop(alt_up_ps/2 * ps2_kb)
{KB_CODE_TYPE decode_mode;
 alt_u8 data;
 char ascii;
 while(1)
 {if(ps2_decode_scancode(ps2_kb, &deco
 de_mode, &data, &ascii) == 0 )
 {if(decode_mode == KB_ASCII_MAKE_
 CODE&&.IS_CHAR(ascii))
 { printf("VGA display character
 %c\n", ascii);
 vga_disp_char(ascii);
 }
 else if(decode_mode == KB_BINARY_
 MAKE_CODE&&.data == ASCII_SPACE)
 {printf("VGA display clear screen\n");
 vga_clear( );
 }
 else if(decode_mode == KB_BINARY_
 MAKE_CODE&&.data == ASCII_ENTER)
 {printf("VGA display pre-loaded image\n");
 vag_disp_img( );
```

```
}
}
}
}

int main( )
{ alt_up_ps2_dev * ps2;
 printf("Initializing...\n");
 initialize_ps2(ps2);
 initialize_vga( );
 if(ps2->decice_type == PS2_KEYBOARD)
 {printf("System start workn,press any Char or
 Enter Key (for image)
 \n");
 main_loop(ps2);
 }
 else
 {printf("PS2 Device Init Error\n");
 exit(-1);
 }
 return 0;
}
```

5.2 显示结果分析

系统上电后自动加载程序, 键盘控制进入字符显示子程序后, 出现图 9(a)所示界面, 说明从缓存的 .mif 文件中正确读取数值, 对设备初始化操作成功。按下普通字符, 显示如图 9(b)界面, 说明系统正确查表, 字符显示成功。按下 space 键出现 9(c)

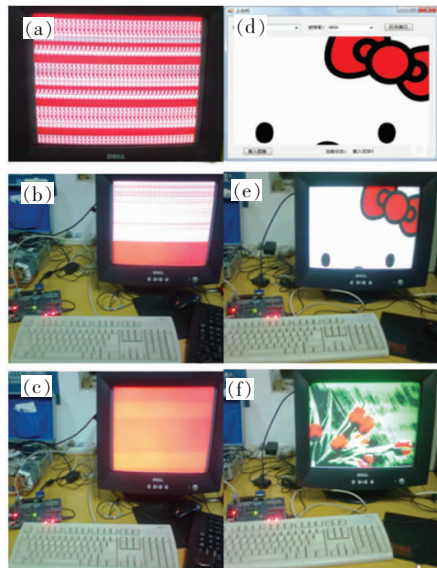


图 9 系统显示结果

Fig. 9 Results of the system displays

界面,执行清屏操作。系统字符显示功能正确。

打开上位机,设置图片格式 640×480 ,设置串口号和波特率,固定 8 位数据位、1 位停止位,点击上位机“打开串口”按钮,如图 9(d)所示,点击“载入图片”后,CRT 显示器显示上位机所载入的图像,如图 9(e)所示。说明系统可以接收串口数据并写入 SDRAM 存储区,并由 VGA 控制器读取后驱动显示。重新载入图片后系统显示如图 9(f),说明 SDRAM 读写正确。系统图像显示功能正确。

6 结 论

系统采用可实现用户自定制内核功能的 SOPC 技术,实现嵌入式 SOPC 内核移植、在 PS/2 键盘控制下由 CRT 显示器显示字符和图片。完成 32 bits NIOS II 内核构建,实现 Avalon 总线控制下各控制器模块的添加和配合工作。解决了 VGA 显示基于计算机或专用芯片不易携带和更改这一局限性。并具有电路尺寸小,设计方法灵活、可移植性强、通用性好的优点。

参 考 文 献:

- [1] 李兰英. Nios II 嵌入式软核 SOPC 设计原理及应用 [M]. 北京:北京航空航天大学出版社,2006.
- [2] 潘松,黄继业. EDA 技术使用教程(第三版) [M]. 北京:科学出版社,2006.
- [3] 刘敏,戴曙光,穆平安. 采用 SOPC IP 核技术实现液晶屏显示 [J]. 液晶与显示,2011,26(5):665-672.
- [4] 王亮等. VGA 汉字显示的 FPGA 设计与实现 [J]. 计算机工程与设计,2009,30(2):275-277.
- [5] 朱奕丹,方怡冰. 基于 FPGA 的图像采集与 VGA 显示系统 [J]. 计算机应用,2011,31(5):1258-1260.
- [6] 胡建革,宋海声,陶中幸. 基于 FPGA 的数据采集与显示系统的设计 [J]. 仪表技术与传感器,2012,(3):53-55.
- [7] 熊文彬,蒋泉,曲建军,等. 基于 FPGA 实现的视频显示系统 [J]. 液晶与显示,2011,26(1):92-95.
- [8] 王鸣浩,王志,吴小霞. 基于 SOPC 的高帧频数字图像采集显示系统 [J]. 液晶与显示,2011,26(5):650-654.
- [9] 郭小清,谢忠红. 一种基于 AVALON 总线的 VGA 控制器的设计与实现 [J]. 科学技术与工程,2010,10(13):3236-3240.
- [10] Altera. SDR SDRAM Controller [M/CD]. USA:Altera,2002.
- [11] Wang Ziting,Guo Haili,Sun Yan. Design of VGA image controller based on SOPC technology [C]//2009 International Conference on New Trends in Information and Service Science, Washington DC, USA: IEEE, 2009: 825-827.
- [12] 姚善学,王代强. 基于 FPGA 的串口通信与 VGA 显示 [J]. 电子设计工程,2011,19(17):146-148.