

文章编号:1007-2780(2009)05-0718-05

基于 SOPC 的 TFT 触摸屏显示系统设计

黄 亮,杨景常

(西华大学 电气信息学院,四川 成都 610039,E-mail:huangliang.112@163.com)

摘 要:提出了一种基于 SOPC 系统的 TFT 触摸屏显示系统设计思路,介绍了对该触摸屏进行控制的硬件电路以及软件编写流程,给出了系统硬件构成与软件设计的主要程序。在软件设计完成并调试成功后,在 Quartus II 环境中将整个项目进行编译并最终下载到 EPCS4 芯片中,系统显示结果清晰、稳定,达到了设计的要求。

关 键 词:Nios II;可编程片上系统;TFT 触摸屏;现场可编程门阵列

中图分类号:TP216 **文献标识码:**A

1 引 言

Nios II 是 Altera 公司推出的第二代 IP 软核处理器,与其它 IP 核一起构成了 SOPC 系统的主要部分。SOPC Builder 集成在 EDA 工具 Quartus II 中,提供了 Nios II 处理器及一些常用外设接口,如 DMA 控制器、SDRAM 控制器、SPI 接口以及锁相环 PLL 等,对于一些库中没有提供的模块用户可以定义添加。用户可以通过自定义逻辑的方法在 SOPC 设计中添加自己开发的 IP 核。定制用户逻辑外设是使用 Nios II 嵌入式软核处理器的 SOPC 系统的重要特性。

液晶显示器(LCD)具有电压低、功耗小、易集成、轻巧便携以及显示效果好等突出优点,现已广泛应用于各种显示领域,在人机交互中扮演着重要的角色。

本文采用 T32QM6450 TFT-LCD 触摸屏显示模块(驱动器是 IS2102,分辨率为 240×320,26 万色,可以显示文字、彩图等),利用 SOPC Builder 中元件编辑器 Create New Component,实现了嵌入式 Nios II 软核处理器与液晶触摸屏显示模块的系统设计^[1],并给出了相应的系统硬、软件的流程和主要程序。

2 基于 Nios II 的 SOPC 系统的软硬件设计流程

SOPC 设计包括以 32 位 Nios II 软核处理器

为核心的嵌入式系统的硬件配置、硬件设计、硬件仿真、软件设计、软件调试等。SOPC 系统设计的基本工具软件有:Quartus II,用于完成 Nios II 系统的综合、硬件优化、适配、编程下载和硬件系统测试;SOPC Builder,是 Altera Nios II 嵌入式处理器开发软件包,用于实现 Nios II 系统的配置、生成;Nios II IDE,用于进行软件开发、调试及运行。

与传统的嵌入式系统设计不同,基于 Nios II 的 SOPC 系统开发分为硬件开发和软件开发两个流程。硬件开发过程包括由用户定制系统硬件的构建,然后由计算机完成硬件系统的生成;软件开发则与传统方式比较接近,在构建的硬件系统之上建立软件设计。图 1 为基于 Nios II 的 SOPC 系统开发整体设计流程^[2]。

从图 1 中可见,虽然 SOPC 开发流程比 FPGA 的开发流程增加了处理器、外设接口的定制步骤以及软件开发等步骤,但这些新增加的步骤在 SOPC Builder(定制处理器和外设接口)、Nios II IDE(软件集成开发环境)工具的协助下是很容易实现的。当用户完成在开发板上基本的 Nios II 处理器硬件原型设计之后,可以将设计下载至 FPGA,然后就可以建立在硬件上运行的软件原型。

Nios II IDE 是基于开发和可扩展的 Eclipse 平台,将通用用户界面和顶级开放相结合,与第三

图 1 基于 Nios II 的 SOPC 系统开发整体流程

Fig.1 Flowchart of the SOPC system development based on Nios II

方工具无缝地集成在一起。所有软件开发任务都可在 Nios II IDE 下完成,包括编辑、编译和调试程序。

系统硬件开发流程主要包括:

(1)使用 SOPC Builder 系统综合软件来选取合适的 CPU 以及外围器件(如片内存储器、PIO、UART 和片外存储器接口等),并定制它们的功能;

(2)使用 Quartus II 软件来选取具体的 Altera 可编程器件系列,并对 SOPC Builder 生成的 HDL 设计文件进行布局布线,再使用 Quartus II 软件选取目标器件并对 Nios II 系统上的各种 I/O 口分配管脚,在编译的过程中,Quartus II 从 HDL 源文件综合生成一个适合目标器件的网表,最后生成配置文件;

(3)使用 Quattus II 编程器和 Altera 下载电缆,将配置文件(用户定制的 Nios II 处理器系统的硬件设计)下载到 FPGA 开发板上,当校验完当前硬件设计后,还可再次将新的配置文件下载到开发板上的非易失存储器里。

系统软件设计流程包括:

(1)在使用 SOPC Builder 系统集成软件进行

硬件设计的同时,就可以开始编写独立于器件的 C/C++控制程序;

(2)在 Nios II IDE 中建立新的软件工程时,IDE 会根据 SOPC Builder 对系统的硬件配置自动生成一个定制的 HAL(硬件抽象层)系统库,该库能为程序和底层硬件的通信提供接口程序,它类似于创建 Nios 系统时 SOPC Builder 生成的 SDK 文件;

(3)使用 Nios II IDE 对软件工程进行编译、调试;

(4)将硬件设计下载到 FPGA 开发板上后,就可以将软件下载到开发板上并在硬件上运行^[3]。

3 建立基于 SOPC 的 TFT 显示系统

3.1 系统硬件构成

3.1.1 FPGA 与外部设备

采用 Altera 的 FPGA-Cyclone II 系列的 EP2C8 器件,其逻辑单元共 8 256 个,含 36 个 M4K RAM 块,2 个锁相环,最大 I/O 接口为 182 个,等效门数为 42 万门(利用 Nios II IDE 定制一个完整的 32 位 CPU 只需占用约 700 个逻辑单元)。本设计采用的 FPGA 核心板其外部设备主

要有指示灯、4 MB EPCS4 芯片(存储软、硬件程序)、8 MB SDRAM 芯片(程序运行时将其导入 SDRAM)、T32QM6450 TFT-LCD 触摸屏显示模块等。

3.1.2 T32QM6450 TFT-LCD 触摸屏显示模块

T32QM6450 TFF-LCD 触摸屏显示模块带手写输入功能,屏幕分辨率为 240×320 ,屏幕大小为 8.1 cm(3.2 in),屏幕颜色为 26 万色,3.3 V 供电,16 位并行数据接口,驱动器是 IS2102。其系统接口结构如图 2 所示。

图 2 TFT 触摸屏系统接口结构

Fig. 2 Interface structure of the TFT touch-screen system

3.2 系统硬件设计

系统的硬件结构如图 3 所示。FPGA 片内包括 Nios II 处理器以及通过 Avalon 总线与处理器相连的外设接口,主要有 JTAG UART、TFT 接口、SDRAM Controller;片外设备包括 T32QM-6450 TFT-LCD 触摸屏、SDRAM 和 LED 等^[4]。

JTAG UART 实现 PC 和开发板通信,主要用于调试,从键盘输入字符,然后处理器将字符通过 LCD 接口传输到 TFT 触摸屏上显示出来。脱

图 3 系统的硬件结构框图

Fig. 3 Hardware diagram of the system

机时,可以用小键盘或按键等作为输入设备取代该模块。SDRAM Controller 实现处理器和 SDRAM 之间的数据存取,包括 SDRAM 存储程序和字符、图形以及颜色等数据;T32QM6450 TFT-LCD 触摸屏显示方式为点阵方式,由驱动程序根据输入信号,通过查询方式确定要显示的字符、图形以及颜色,并产生地址信号及控制信号,将点阵数据送到触摸屏显示^[6]。

根据所用到的外设和器件特性,在 SOPC Builder 中建立系统要添加的外设模块,包括: Nios II CPU、外部 SDRAM、JTAG UART 和 PIO 等。设定好各个参数,然后添加 LCD_Controller,将 LCD 控制器的 avalon_master_0 接口连接到 SDRAM 上。然后按照设计流程创建 Quartus II 新工程,添加 CPU 和外设,指定基地址,设置系统,生成系统模块,建立顶层文件,加入锁相环,编译并配置 FPGA 引脚,最后编译后配置到 FPGA 中^[6,7]。

4 软件控制部分设计

Nios II IDE 集成开发环境由 IDE 基于用户在 SOPC Builder 中创建 Nios II 处理器系统时自动生成硬件抽象层(Hardware Abstraction Layer, HAL)系统库。HAL 的主要作用是为用户的嵌入式系统上的外围设备提供与之相一致的接口程序,如果外设配置有了变化,则 HAL 的硬件设备驱动配置也会随之自动更新,从而避免了由于底层硬件的变化而产生的程序错误。由于 T32QM6450 TFT-LCD 显示模块属 HAL 支持的外围设备,因此可以利用 ANSI C 标准库函数与 HAL API 接口平台实现对设备的访问。

受篇幅所限,下面仅给出了结构体 lcd_frame_buffer_struct 程序、打点函数程序和显示一幅图片的代码。结构体 lcd_frame_buffer_struct 用于定义结构体变量,对 TFT 的操作就是通过此结构体来完成的。

```
typedef struct {  
    lcd_frame_array * lcd_frame0;    //显存 0 的指针  
    lcd_frame_array * lcd_frame1;    //显存 1 的指针  
    int width;                        //LCD 控制核支持的宽度  
    int height;                       //LCD 控制核支持的高度  
    int color_depth;                 //LCD 控制核支持的颜色数据宽度
```

```
int bytes_per_pixel;           //每个点的颜色数据所占的字节数
int bytes_per_frame;           //一帧数据所占字节数
int frame_buffers;             //LCD 控制核支持的缓存数
int lcd_controller_base;       //LCD 控制寄存器的基地址
} lcd_frame_buffer_struct;

打点函数的定义为:

void lcd_print_pixel ( int row, int line, char r, char g, char b, lcd_frame_buffer_struct * lcd_frame_
buffer, int buffer_number )
{
    int addr1, addr2, addr3;
    addr1=row * lcd_frame_buffer->width * 3+line * 3;      //计算出该点的 R 的位置
    addr2=addr1+1;      //计算出该点的 G 的位置
    addr3=addr2+2;      //计算出该点的 B 的位置
    if (buffer_number==0)
    {
        IOWR_8DIRECT ( lcd_frame_buffer-> lcd_frame0, addr1, r );
        IOWR_8DIRECT ( lcd_frame_buffer-> lcd_frame0, addr2, g );
        IOWR_8DIRECT ( lcd_frame_buffer-> lcd_frame0, addr3, b );
    }
    else if ( buffer_number==1 )
    {
        IOWR_8DIRECT ( lcd_frame_buffer-> lcd_frame1, addr1, r );
        IOWR_8DIRECT ( lcd_frame_buffer-> lcd_frame1, addr2, g );
        IOWR_8DIRECT ( lcd_frame_buffer-> lcd_frame1, addr3, b );
    }
}
```

由这个打点函数很容易设计出画图函数,其代码如下所述。其中,Picture_data 为图片数组的指针。显示一幅图片的代码为:

```
void LCD_print_picture ( lcd_frame_buffer_struct * lcd_frame_buffer, char * Picture_data )
{
    int row, line;
    int addr;

    for(row = 0; row < lcd_frame_buffer-> height; row++)
    {
        for(line=0; line< lcd_frame_buffer-> width; line++)
        {
            addr=row * lcd_frame_buffer-> width * 3+line * 3;
            lcd_print_pixel ( row, line, Picture_data(addr),
```

图 4 系统显示效果图
Fig. 4 Design sketch of system display

```
Picture_data(addr+1), Picture_data(addr+2),  
lcd_frame_buffer, 0 );  
}  
}  
}
```

软件设计完成并调试成功后,在 Quartus II 环境中将整个项目进行编译,即将硬件配置信息与软件信息编译在一起,并最终下载到 EPCS4 芯片中,系统显示结果清晰、稳定,达到了设计的要求,显示的图片如图 4 所示。

5 结 论

基于 SOPC 系统,对 TFT 触摸屏显示系统

软硬件进行了设计。软件设计完成并调试成功后,在 Quartus II 环境中将整个项目进行编译,即将硬件配置信息与软件信息编译在一起,并最终下载到 EPCS4 芯片中,系统显示结果清晰、稳定,达到了设计的要求。

从设计中可以看出采用 Nios II 嵌入式处理器与使用 Nios II IDE 集成开发环境进行软硬件开发的优越性。SOPC 技术代表了电子设计自动化领域的主流方向,新一代嵌入式处理器 Nios II 必将在嵌入式系统设计与应用中得到广泛应用,本文的设计思路与实例开发具有一定的工程应用与参考价值。

参 考 文 献:

[1] 王晓迪,张景秀. SOPC 系统设计与实践 [M]. 北京:北京航空航天大学出版社,2008.
[2] Altera 公司组/Altera 公司. Nios II 嵌入式处理器设计大赛 2006 年优秀作品精选集 [M]. 西安:西安电子科技大学出版社,2007.
[3] 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.
[4] 龙燕,李剑峰. 以 FPGA 为核心的液晶显示电路设计与实现 [J]. 液晶与显示,2006,21(3):274-278.
[5] 郭书军,侯文杰,贾克斌. Nios II 液晶屏接口组件设计 [J]. 液晶与显示,2007,22(2):213-217.
[6] 岳帮辉,魏廷存,樊晓桢. 手机用 TFT-LCD 驱动芯片内置 SRAM 的研究与设计 [J]. 液晶与显示,2006,21(5):566-570.
[7] 王骞,丁铁夫. 基于 FPGA 的液晶显示驱动 IP 核的实现 [J]. 液晶与显示,2005,20(4):324-327.

Design of Display System for TFT Touch-Screen Based on SOPC

HUANG Liang, YANG Jing-chang

(College of Electric and Information, Xihua University, Chengdou 610039, China, E-mail: huangliang.112@163.com)

Abstract

An idea of TFT touch-screen display system design based on SOPC was presented. The hardware interface and the flowchart of software with relevant programs were introduced, the main procedure that the system hardware constitutes to design with software was given. After the design and debugging of the software were completed, the entire project was compiled in Quartus II environment and was downloaded in EPCS4 chip. The display result of the system was clear and stable, the design requirements was achieved entirely.

Key words: Nios II ; SOPC system; TFT touch-screen; FPGA

作者简介:黄亮(1984—),男,广西灵山县人,硕士研究生,研究方向为测试技术与仪器。