

基于 FPGA 的嵌入式系统设计

王俊雄, 黄 铨, 刘正义

(西南交通大学电气工程学院, 四川省成都市 610031)

摘要:提出了一种基于 FPGA (现场可编程门阵列) 和软核 CPU 的嵌入式系统设计的新方法。FPGA 芯片选用 Altera 公司的 Cyclone 系列芯片作为处理器, 配合 Altera 公司的 NIOS 软核嵌入式处理器构成整个系统的核心; 操作系统采用 μ COS 嵌入式系统, 并移植到 FPGA 平台上; 外围添加 USB 接口作为扩展接口, 使用 Cypress 公司的 EZ_USB 系列芯片; 整个系统以 FPGA 和 NIOS 为中心进行设计, 外围接口包括 JTAG、串口、USB 口等。给出了系统硬件架构以及与 USB 芯片的接口应用电路。

关键词: FPGA; NIOS; μ COS; USB; 嵌入式系统

中图分类号: TP368

0 引言

嵌入式系统是以应用为中心, 以计算机技术为基础, 并且软硬件可裁剪, 适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。嵌入式系统是一个外延极广的名词, 凡是与产品结合在一起的具有嵌入式特点的系统都可以叫嵌入式系统, 很难给它下一个准确的定义。因此, 目前通常把嵌入式系统的中心放在“系统”(即操作系统)上, 是指能够运行操作系统的软硬件综合体。总体上, 嵌入式系统可以划分成硬件和软件两部分。硬件一般由高性能的微处理器和外围接口电路组成, 软件一般由实时操作系统和其上运行的应用软件构成, 软件和硬件之间由所谓的中间层即 BSP(板级支持包)连接。

传统的嵌入式处理器一般都选用 ASIC(专用集成电路)。ASIC 是一种为具体任务而特殊设计的专用器件, 由于在设计过程中进行了专门优化, 其性能、性价比都非常高, 减少了系统软件和硬件设计的复杂程度, 降低了系统成本, 但还是存在一些缺点如前期设计费用高, 且一旦设计完成就无法升级和扩展等。

近年来, 各种可编程芯片的出现, 改变了嵌入式硬件模块的设计方法, 出现了“可重构计算”的概念, 它允许在不改变硬件电路板的情况下, 实现不同的控制接口和控制功能。可重构计算是通过 FPGA (现场可编程门阵列) 实现的。FPGA 是 20 世纪 80 年代中期以后发展起来的一种可编程的大规模集成器件, 具有保密性好、体积小、重量轻、可靠性高等一系列专用大规模 ASIC 的优点, 同时, 随着系统设计引入 SOPC(片上可编程系统), FPGA 足以与传统的 ASIC 抗衡, 具有

作为嵌入式处理器的能力。

1 硬件设计

1.1 系统整体设计

整个设计的硬件系统结构见图 1。微处理器选用 Altera 公司的 Cyclone 系列 EP1C1240C8 存储设备选用 AM29LV320D; 外围接口包括 USB 接口、以太网接口、JTAG 口以及串口; 选用分辨率为 128×64 像素的液晶显示器 (LCD) 模块做为输出显示。

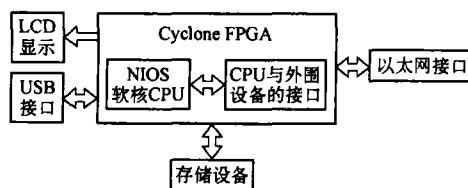


图 1 系统硬件结构

1) EP1C12Q240C8

EP1C12Q240C8 拥有 12 060 个 LE(逻辑单元)、52 个 4 kB 存储块、2 个锁相环、249 个可用的 I/O 脚。通过 Altera 公司的 QuartusII 软件可以对 EP1C12 的 I/O 脚很方便地进行定义、配置; 允许通过编程下载到 FPGA 中, 在不改变硬件电路板的基础上得到需要的功能电路、接口或者控制功能; Cyclone 系列芯片完全支持 Altera 公司的 NIOS 处理器, 并且 SOPC 软件对这个系列的芯片进行了专门的优化, 使其性能得到更进一步提高。

2) CY7C68013

选用美国 Cypress 公司的 USB2.0 芯片 CY7C68013 作为 USB 接口芯片, 它占用更少的电路板空间, 并缩短开发时间。该芯片是针对 USB2.0 的, 而且和 USB 1.1 兼容。它支持两种传输速率: 全速 12Mbit/s

收稿日期: 2005 10 17

和高速 480 Mbit/s 不支持低速 1.5 Mbit/s。CY7C68013 主要结构包括 1 个 8051 处理器、1 个智能 SIE(串行接口引擎)、1 个 USB 收发器、16 kB 片上 RAM(其中包括 4 kB FIFO)存储器以及 1 个 GPIF(通用可编程接口)。

3) RTL8019

采用 RTL8019 作为网络接口芯片, 该器件是一个以太网控制器, 实现了网络七层协议栈中的传输层和 MAC(媒体访问控制)层的功能, 并且有很好的网络兼容性。

4) 存储设备

存储设备采用 Flash 存储器和 SDRAM 两种。Flash 存储器选用了 AM29LV320D, 它具有 4 MB 空间, 主要用来存放软件代码以及一些需要保存的参数; 由于 NIOS 是 32 位的软核 CPU, 所以 SDRAM 选用了 2 块 HY57V641620 串接形成 32 位数据存储空间, 用在系统运行时的代码和数据存储。

5) LCD

选用带中文字库、具有 4 位/8 位并行、2 线/3 线串行多种接口方式、内部含有国标一级和二级简体中文字库的点阵图形液晶显示模块, 其显示分辨率为 128×64 像素, 内置 8192 个 16×16 点汉字和 128 个 16×8 点 ASCII 字符集, 利用该模块灵活的接口方式和简单、方便的操作指令, 可构成全中文人机交互图形界面。

1.2 微处理器外围的接口

USB 接口与 FPGA 的连接如图 2 所示。

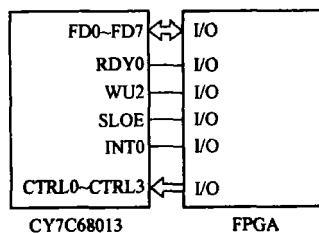


图 2 USB 与 FPGA 的连接

USB 芯片采用 GPIF 方式。GPIF 是主机方式, 可以软件编程读写控制波形, 几乎可以对任何 8 bit/16 bit 接口的控制器、存储器和总线进行数据的主动读写。

USB 2.0 协议定义了端点作为数据的接收器和发送器。主机发送 4 bit 的地址和 1 bit 的方向来选择端点, 因此 USB 最多可有 32 个端点定义: N0~N15 和 OUT0~OUT15。FX2 定义了 7 个端点, 在高速模式下的端点缓存结构如图 3 所示。

图 3 中: EP0IN 和 OUT、EP1IN、EP1OUT 是 64 B 的端点缓存; EP0 是默认的控制传输端点, 既是 N 端

点也是 OUT 端点; EP1IN、EP1OUT 支持块、中断和同步传输; EP0、EP1IN 和 EP1OUT 只能由 FX2 的固件访问, 而 EP2、EP4、EP6、EP8 无需固件干涉即可与片外互传高速数据; FX2 端点配置方式非常灵活; EP2、EP4、EP6、EP8 是大容量高带宽的数据传输端点, 可设为 N 或 OUT 端点的一种, 能配置成多种形式以适应带宽需要。图 3 中, 每一列代表一种配置方式。EP2、EP4、EP6、EP8 可包括 2 个~4 个 512 B 或 1 024 B 的缓存, 分别表示端点可配置成双重、三重和四重缓存。

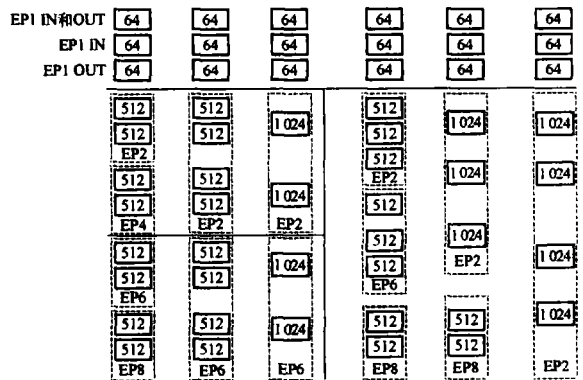


图 3 端点缓存结构

由于 FPGA 提供的标准 I/O 口, 所以只需要将对应的引脚与 I/O 口连接上, 可以通过编程来产生需要的时序, 十分灵活。

2 软件设计

2.1 核心处理器 NDS

采用 Altera 公司的 NDS 软核嵌入式处理器。它是一种可特许的通用 RISC CPU。Altera 公司以 IP 核的方式将它提供给设计者。它可以与各种各样的外设、定制指令和硬件加速单元相结合, 构成一个定制的 SOPC。该处理器具有可由用户配置的 L1(第 1 层)指令与数据超高速缓存。NDS 处理器还具有一种基于 JTAG 的 OC1(片上仪器)芯核, 使软件开发人员在实时调试方面具有更明显的优势。该处理器的软件支持可扩展到对 APR、IP、ICMP、TCP、UDP 和以太网的网络协议支持。

CPU 与其他器件的接口的设计使用 Quartus II 软件, 它是 Altera 公司的第 4 代可编程逻辑器件集成开发环境, 提供从设计输入到器件编程的全部功能。该软件提供一个名为 SOPC Builder 的工具, 允许用户在图形界面下直接搭建自己的系统, 直观且方便。在 SOPC Builder 中包含了一些基本器件与 NDS CPU 的接口, 如 SDRAM 控制器、Flash 存储器的接口等。

在本设计过程中, 采用 SOPC Builder 在图形化界面内定制了一个从硬件到软件的完整系统。

首先, 根据设计需要使用 SOPC Builder 来对 NDS CPU 进行定制, 包括 CPU 是否使用数据和代码缓冲、CPU 寄存器数量 (NDS CPU 最多可以有 512 个寄存器) 等。本设计中的 NDS CPU 有 256 个寄存器, 未使用数据和代码缓冲。除了 NDS CPU 外, 还需要用 SOPC Builder 将 CPU 与外围设备的接口添加进来。对于本设计来说, 就是将 USB 接口以用户自定义接口的方式添加进来, 其余器件与 CPU 的接口只需在 SOPC Builder 中将对应的接口模块添加进来即可。

然后, 用 SOPC Builder 将定制转换为 VHDL 等具体的设计文件。此外, SOPC Builder 还根据定制的结果, 自动生成针对特定硬件环境的 C 语言和汇编语言的头文件以及函数库, 非常方便、灵活。

2.2 操作系统移植

该设计采用 μ C OS 操作系统。 μ C OS 设计时就充分考虑到在不同平台上移植的需求, 将同平台相关的部分局限在一个很小范围内, 对于不同的平台只需对下面一些函数和宏进行重写。

需要改写的宏如下:

OS_ENTER_CRITICAL 和 OS_EXIT_CRITICAL, 这是两个宏, 用来进行临界段保护。在本设计中使用汇编代码中断来实现。

OS_TASK_SW: 这是一个用于进行任务切换的宏。本设计中利用 CPU 的软件中断方式实现。也就是说调用此宏产生软件中断, 然后由相应的中断处理程序来具体实现任务上下文保护和任务切换。

OS_InCriticalSection: 实现中断级任务切换, 用纯汇编实现。

OS_UserLevelTaskSw: 实现用户级上下文切换, 用纯汇编实现。

OSTickISR: 为系统定时器中断的处理函数, 用纯汇编实现。

OSTaskStackInit 用来创建任务时对任务堆栈进行初始化。

2.3 USB 软件设计

2.3.1 驱动程序开发

开发 USB 设备驱动程序, 可采用 Numege 公司的 DriveWorks, Microsoft 公司的 2000DDK, 并以 VC++ 6.0 作为辅助开发环境。Driver works 提供的驱动向导可根据用户的需要, 自动生成代码框架, 减少了开发的难度, 缩短了开发的周期。在 Cypress 公司的 EZ-USB FX2 开发包中, 有一个通用的驱动程序, 该程序可不加修改经 DDK 编译后直接使用。此驱动程序是用来与基于 EZ-USB 的外设连接的通用设备驱动程序,

它提供公共 USB 总线设备请求和数据传输的用户态接口。

2.3.2 固件设计

HD (人机接口设备) 类是 Windows 完全支持的 USB 总线设备类型中的一种。在运行 Windows 98 或更高版本操作系统的 PC 机上, 应用程序可以与 HD 进行通信。对于主机的驱动程序和 HD 通信, 设备必须使用相应的固件程序来满足一定的要求, 通过固件程序的调度, 主机才可以得到设备的描述符以及完成断点的传输。

写固件程序是比较复杂的, 需要用到大量的函数, 但其基本结构却相对简单, 包括下面 3 个过程:

- a) 初始化, 包括处理器和外围电路的初始化;
- b) 主函数, 包括完成符合设备特定要求的代码;
- c) 中断处理, 包括处理各种中断的程序代码。

该设计已经制作成印制电路板, 并调试通过, 系统能够正常运行, 各种接口能够正常工作。

3 结束语

可以看出, 采用 FPGA 进行嵌入式系统设计有两个突出的特点。首先, 降低了硬件设计的难度, 缩短了硬件开发周期, 提高了设计的可靠性。CPU 本身是以软核的方式实现, 其功能可根据需要进行定制, 相当的灵活、快速。其次, 软件开发容易, 由于 SOPC Builder 开发环境的完备功能, 可以把注意力集中在系统整体构架和功能上来, 而无需过多地考虑细节性的电路设计, 同时还可以得到比较好的系统稳定性和可靠性。

该设计为嵌入式系统设计提供了一条新思路, 体现了 FPGA 的灵活性、高集成性等特性, 大大降低了成本, 缩短了开发时间。

参 考 文 献

- [1] 彭澄廉. 挑战 SOC——基于 NDS 的 SOPC 设计与实践 [M]. 北京: 清华大学出版社, 2004
- [2] 潘松, 黄继业, 曾毓. SOPC 技术实用教程 [M]. 北京: 清华大学出版社, 2005
- [3] 赵曙光, 郭万有, 杨颂华. 可编程逻辑器件原理、开发与应用 [M]. 西安: 西安电子科技大学出版社, 2001
- [4] 吴涛, 韩伟. 一种基于 USB 2.0 接口的数据采集系统的设计方案与实现 [J]. 空间电子技术, 2004 1(2): 54-60
- [5] 常晓明. Verilog HDL 实践与应用系统设计 [M]. 北京: 北京航空航天大学出版社, 2003
- [6] 王田苗. 嵌入式系统设计 with 实例开发 [M]. 北京: 清华大学出版社, 2003

Design of Embedded System Based on FPGA

WANG Junxiong HUANG Xuan LIU Zhengyi

(Southwest Jiaotong University, Chengdu 610031, China)

Abstract This paper proposes a new method for embedded system designing based on FPGA and soft core CPU. The system uses Cyclone series devices and NDS provided by Altera company to be the core of system. The μ C OS is chosen as the operating system and transplanted to FPGA. The system has USB interface using EZ_USB series USB interface control chip provided by Cypress company. The core of system is FPGA and NDS. The system has several interfaces including JTAG, COM, USB. This paper also expatiates the frame of hardware and its interface circuit with USB.

Keywords FPGA; NDS; μ C OS; USB; embedded system

(上接第 57页)

A Fingerprint Image Enhancement Algorithm

LI Qingrong

(University of Electronic Science and Technology, Chengdu 610054, China)

Abstract Fingerprint image enhancement is a key procedure in an automatic fingerprint identification system. An algorithm based on the local direction of fingerprint is proposed to improve the quality of the fingerprint image in noise background. Firstly, local ridge orientation is calculated by making use of the local gray distribution. Then, the gray projection on the local orthogonal axis was used for estimating the local ridge frequency. At last, a set of adaptive Gabor filter which have both orientation-selective and frequency-selective properties was used to enhance the fingerprint image. Simulation shows this algorithm can suppress noise effectively and preserve true ridge structure.

Keywords fingerprint; local ridge orientation; local ridge frequency; image enhancement; Gabor filter

(上接第 60页)

A Fuzzy Comprehensive Evaluation of Software Function Testing

WANG Yunjun¹, YANG Lingling^{1,2}

(1. Southeast University, Nanjing 210096, China)

2. The 28th Research Institute of China Electronics Technology Group Corporation, Nanjing 210007, China)

Abstract This paper studies the software quality evaluation method for function testing software. It applies Fuzzy Mathematics method to the software quality comprehensive evaluation of function testing software and suggest a quantitative evaluating method of software quality. It uses a practical case study to validate the feasibility of the method and provides a new idea for software quality evaluation.

Keywords software quality evaluation; fuzzy comprehensive evaluation; function testing method; function points analysis