



# 基于 $\mu$ Clinux 的 FPGA 远程更新系统的实现

丁 丁<sup>1</sup>, 汤晓斌<sup>1</sup>, 陈立德<sup>2</sup>, 殷树根<sup>2</sup>

(1. 南京航空航天大学 材料科学与技术学院, 江苏 南京 211106;

2. 上海风格信息技术有限公司, 上海 200120)

**摘要:** 针对 Altera 公司的 Cyclone 系列 FPGA 在特殊环境中无法使用下载电缆来重新配置芯片的问题, 设计并实现了一个基于  $\mu$ Clinux 的 FPGA 远程更新系统。该系统利用 Altera 的片上可编程系统 (SoPC) 技术, 嵌入 Nios II 软核处理器, 移植  $\mu$ Clinux 系统, 通过 UART 串口接收升级数据, 通过 MTD 驱动将升级数据写入配置芯片 EPCS4 和 CFI 存储芯片。该方法利用 FPGA 的软硬件可编程的特点, 实现了系统软硬件同时在线升级, 完成了 FPGA 的远程更新。

**关键词:**  $\mu$ Clinux; FPGA; 远程更新; UART

中图分类号: TN47

文献标识码: B

文章编号: 0258-7998(2014)03-0006-03

## Implementation of FPGA remote update system based on $\mu$ Clinux

Ding Ding<sup>1</sup>, Tang Xiaobin<sup>1</sup>, Chen Lide<sup>2</sup>, Yin Shugen<sup>2</sup>

(1. College of Material Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China;

2. Shanghai Figure IT Company, Shanghai 200120, China)

**Abstract:** Aimed at the problema of Altera's Cyclone series FPGA which can't be used in a special environment using the download cable to reconfigure the chip, we designed and implemented a  $\mu$ Clinux-based FPGA remote update system. The system using Altera SoPC technology, which is embedded in a Nios II soft-core processor, to run a  $\mu$ Clinux system and transport the data via the UART serial port. The configuration data is written to the memory chip, EPCS4 and CFI, with MTD driver. The method utilizes the hardware and software programmable features of FPGA to complete the online upgrade of FPGA.

**Key words:**  $\mu$ Clinux; FPGA; remote update; UART

随着集成电路技术的进一步发展, FPGA 因具备功能灵活、开发成本低和上市速度快等特点, 而广泛应用于数字产品设计的各个领域。目前 FPGA 配置数据一般使用基于静态随机存储器 SRAM (Static Random Access Memory) 的存储方式, 掉电后数据消失。因此在实用中需要为 FPGA 芯片外加专用配置芯片, 在上电时, 由这个专用配置芯片把配置数据加载到 FPGA 中, 之后 FPGA 就可以正常工作。配置芯片通常是基于 Flash 存储器的器件。FPGA 配置数据更新的常用做法是: 通过专用的下载电缆把配置数据从 PC 下载到 FPGA 配置芯片中<sup>[1]</sup>。这种方法在早期的开发调试中显得非常便利, 然而在系统组装并形成产品后, 由于下载电缆接口通常不会作为系统的一个外部接口, 此时若想更新 FPGA 配置数据将会很麻烦, 需要把系统重新拆开才能进行<sup>[2]</sup>。因此, 在越来越多的系统中希望通过以配置数据的远程更新来提高

系统的可扩展性和升级的灵活性。

本文针对 Altera 公司 Cyclone IV 系列的一款低功耗、高性能的 FPGA 芯片 EP4CE15F23C8, 结合 Altera 公司提供的 SoPC 开发工具, 嵌入 Nios II 软核处理器技术, 移植  $\mu$ Clinux 系统, 通过 UART (Universal Asynchronous Receiver/Transmitter) 串口获取升级数据 (包括 FPGA 配置文件和  $\mu$ Clinux 系统镜像), 并在  $\mu$ Clinux 系统中添加 MTD 驱动以完成升级数据写入 Flash 存储器的任务。该方法利用 FPGA 的软硬件可编程的特点, 实现了系统软硬件同时在线升级, 完成了 FPGA 的远程更新。

### 1 FPGA 远程更新系统的结构与更新流程

#### 1.1 远程更新系统的系统结构

FPGA 远程更新系统的结构如图 1 所示, 该系统主要包括本地系统和远程系统。本地系统包括目标板和交换板。

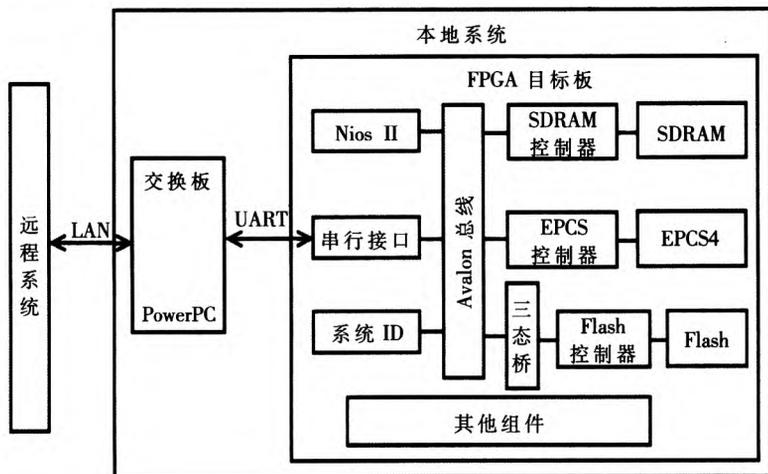


图1 远程更新系统结构

目标板的FPGA采用Altera公司Cyclone IV系列的EP4CE15F23C8芯片。该芯片采用65 nm工艺,内部包含了15 408个逻辑单元LE(Logic Elements)、504 KB片内随机存储器、56个嵌入式乘法器、20个全局时钟网络,具有低功耗、高性能的特点<sup>[3]</sup>。通过Altera Quartus II软件中的SoPC Builder工具,在FPGA芯片上构建了Nios II软核处理器、外设和外设接口,包括EPCS控制器、SDRAM控制器、三态桥、UART、通用I/O等<sup>[4]</sup>。EPCS是Altera提供的专用串行配置芯片,本系统使用EPCS4芯片,它支持多种可配置的时钟源,最高支持100 MHz的外部时钟源,并且具有4 MB的存储空间,完全满足此款FPGA配置数据的存储需要。在本系统中,以Altera Wiki社区发布的nios2-linux-20090929.tar内核开发包为基础,在Nios II上移植了μClinux系统<sup>[5]</sup>。FPGA上还外接了一个NOR型Flash芯片,用于存储μClinux系统镜像。

交换板采用PowerPC的微处理器,运行嵌入式Linux系统,交换板与FPGA目标板间采用UART串口通信方式。

远程系统通过网络接口与交换板相连,是整个升级系统的控制中心。

### 1.2 远程更新流程

FPGA上电启动时,自动产生时序从EPCS中读取配置文件并进行配置。配置完成后,NiosII处理器硬件就已经在FPGA内部生成,这时FPGA进入主复位状态,在内部的复位逻辑作用下,NiosII从其设定的复位地址开始执行程序。在本系统中设计NiosII处理器上电后从外部Flash开始启动、解压并加载μClinux的文件系统,运行相应的应用程序。因此,FPGA的远程更新主要涉及到两个文件,分别为

FPGA的配置文件和μClinux的系统镜像。对于Nios II处理器,完成FPGA的远程更新主要需要解决两个问题:(1)远程更新数据的获取;(2)将配置文件写入EPCS配置芯片。为了不影响原有系统的运行,这两个任务均需在μClinux系统下完成。为此设计如下的更新流程。

(1)升级文件的处理:FPGA的配置文件需在Quartus II软件的配置文件输出选项中添加\*.rbf(Raw Binary File,原始二进制文件)格式输出,并作MSB到LSB的字节变换得到\*.img文件。在Linux下构建μClinux的开发环境,通过make交叉编译生成elf格式的内核镜像文件zImage,并在Nios II Command Shell中运行“nios2-elf-ob-

copy -O binaryzImagezImage.bin”,得到二进制格式的μClinux系统镜像。

(2)升级文件的传输:远程控制系统完成升级文件处理后,通过LAN或Internet发送给PowerPC交换板,最后由交换板通过UART串口发送给FPGA目标板。

(3)升级文件的写入:目标板接收升级文件,完成数据校验后将升级文件写入相应的配置芯片,并在写入完成后重启系统。

## 2 串口文件传输的实现

### 2.1 串口文件传输协议

如前所述,FPGA升级数据在交换板与目标板之间通过UART串口传输,为了顺利完成升级数据的传输,定义了一种串口文件传输协议。该协议采用基于帧传输的方式,将升级数据进行分帧发送,并在传输过程中对单帧进行和校验而对整个升级数据进行CRC校验。相关的帧包括:控制帧、信息帧和数据帧<sup>[6]</sup>。

#### 2.1.1 控制帧

控制帧由交换板在开始传输前发送给目标板,帧内容包括控制字(升级命令)、升级文件的参数等信息,其构成如图2所示。其中,文件参数由升级文件号、相应的总分帧数、单帧大小、最后一帧大小及文件的CRC校验码构成。文件1对应为FPGA的配置文件,文件2对应为μClinux的系统镜像。

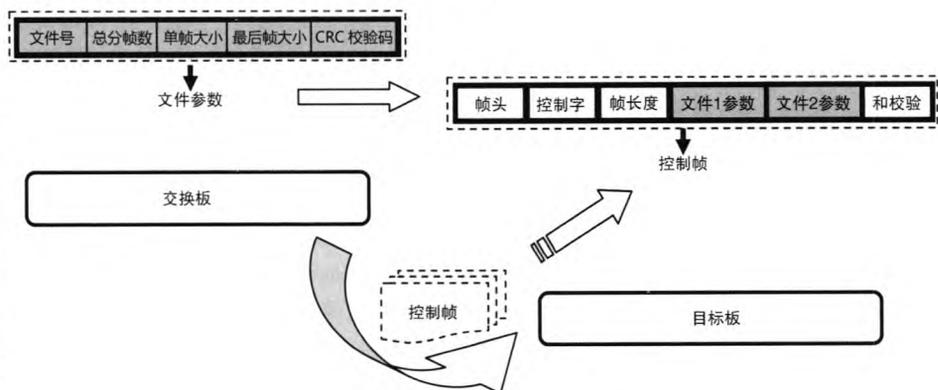


图2 控制帧的构成

### 2.1.2 信息帧

信息帧由目标板反馈给交换板,用于请求分帧数据、返回 FPGA 升级状态等,帧内容如图 3 所示。

帧头	控制字	帧内容		和校验	含义
0x31 0x1	0x1	文件号	帧号	-	帧请求
0x31 0x1	0x2	文件号	0x1 0	-	文件 CRC 校验成功
0x31 0x1	0x2	文件号	0x0 0	-	文件 CRC 校验失败
0x31 0x1	0x3	文件号	0x1 0	-	Flash 写入成功
0x31 0x1	0x3	文件号	0x0 0	-	Flash 写入失败
0x31 0x1	0x4	0x1	0x1 0	-	重启,退出升级
0x31 0x1	0x4	0x1	0x0 0	-	非重启,退出升级

图 3 信息帧的构成

### 2.1.3 数据帧

数据帧由交换板发送给目标板,用于完成升级数据的传输,帧内容包括分帧号、分帧数据等。

在升级数据的传输过程中,首先由交换板向目标板发送控制帧,用于通知目标板进入升级程序,同时发送所需升级的数据类型、分帧帧数及升级数据的 CRC 校验码。目标板在接收到升级控制帧后,启动相应的接收程序,并向交换板返回信息帧,用于请求分帧数据。目标板在接收到帧号请求信息帧后,开始向目标板发送数据帧。在传输中如有错帧或漏帧,目标板均可重发请求信息帧,这样不断重复即可完成整个升级数据的传输。

### 2.2 串口传输过程中的数据丢失现象

在完成了上述串口传输协议相关代码的编写后,在实际测试中却发现,传输过程中会频繁发生数据丢失的现象。在分析各种可能原因后,确定为目标板的 UART 串口 FIFO 深度过低所导致。FIFO 缓冲区的深度直接影响通过 UART 传输数据所需的 CPU 开销,是影响 UART 性能的关键因素。FIFO 的深度越低,在传输过程中就需要越多的 CPU 中断,尤其在短时大数据量的传输过程中就可能出现 CPU 无法响应中断而发生数据丢失的现象。

然而 Altera 的 SoPC Builder 工具默认所添加的 UART 设备没有配带 FIFO,仅有 1 B 的深度,为此参考 Altera Wiki 社区所发布了带 FIFO 的 UART 组件。这里为 FPGA 重新配置了 FIFOed Uart 设备<sup>[7]</sup>,并将其接收 FIFO 的深度设为 256 B。

在使用 FIFOed Uart 设备并配置相关的驱动后,串口传输过程中的数据丢失的问题得以解决。

### 3 μClinux 下 Flash 的写入操作

在接收完升级数据后,需要完成升级数据写入相应配置芯片的操作。由于配置芯片是基于 Flash 的设备,在 μClinux 下可以通过调用内存技术设备 MTD (Memory Technology Device) 驱动完成相应的写入操作。MTD 驱动是 Linux 下标准的 Flash 驱动器,它支持大多数 Flash 存储设备,兼有功能强大的分区定义和映像工具。MTD 在硬件与内核之间提供一个抽象的接口,可理解为 Flash

的设备驱动程序。通过该接口,可像读写普通文件一样对 Flash 设备进行读写操作。

Altera Wiki 社区发布的 μClinux 基础开发包中 MTD 驱动的分区配置文件为 linux-2.6/arch/nios2/kernel/config.c,根据相应的硬件芯片修改此文件并运行“make menuconfig”修改内核编译设置后,完成 MTD 驱动的添加<sup>[8]</sup>。添加驱动成功后,会在 μClinux 的 dev 目录下生成字符设备 mtd0 或块设备 mtblock0,通过系统调用 dd 命令,“dd if=/tmp/zImage.bin of=/dev/mtdblock0”,即可完成 Flash 的写入操作。

在完成升级数据写入 Flash 的操作后,还需要对写入的数据进行 CRC 校验,校验完成后即可重启系统,从而完成了整个 FPGA 升级的任务。

本文介绍了一种基于 μClinux 的 FPGA 远程更新系统的实现方法,使用该方法不用打开机箱即可随时更新 FPGA 配置程序,方便了产品在客户端维护升级,具有灵活方便、配置时间短、易于操作的特点。通过长期的实际运行验证了这种升级方式的正确性和稳定性。另外,虽然本文是针对 Altera 公司 Cyclone 系列的 FPGA 器件而设计,但也可适用于 Altera 其他系列的 FPGA 器件,故其有一定的通用性,并为其他厂商的 FPGA 器件的远程升级提供参考。

#### 参考文献

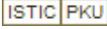
- [1] 李兰英. Nios II 嵌入式软核 SOPC 设计原理及应用[M]. 北京:北京航空航天大学出版社,2006.
- [2] 李鹏,兰正龙. 用 CPLD 和 Flash 实现 FPGA 配置[J]. 电子技术应用,2006,32(6):101-3.
- [3] Altera Conproation. Cyclone IV device data sheet[EB/OL]. (2013-05-01)[2013-10-20]. <http://www.altera.com.cn/literature/hb/cyclone-iv/cyiv53001.pdf>, 2012.
- [4] Altera Conproation. Cyclone IV device handbook[EB/OL]. (2013-05-01)[2013-10-20]. <http://www.altera.com.cn/literature/hb/cyclone-iv/cyclone4handbook.pdf>, 2012.
- [5] Altera Wiki. Install Nios II Linux[EB/OL]. (2013-01-14)[2013-10-20]. [http://www.alterawiki.com/wiki/Install\\_Nios\\_II\\_Linux](http://www.alterawiki.com/wiki/Install_Nios_II_Linux), 2013.
- [6] 王媛媛,刘树林. 基于 FPGA 的串行通信控制系统的设计[J]. 微型机与应用,2011,30(5):57-59.
- [7] Altera Wiki. FIFOed Avalon Uart[EB/OL]. (2013-06-11)[2013-10-20]. [http://www.alterawiki.com/wiki/FIFOed\\_Avalon\\_Uart](http://www.alterawiki.com/wiki/FIFOed_Avalon_Uart), 2012.
- [8] Altera Wiki. Memory technology device[EB/OL]. (2011-11-07)[2013-10-20]. <http://www.alterawiki.com/wiki/MTD>, 2013.

(收稿日期:2013-10-28)

#### 作者简介:

丁丁,男,1988年生,硕士研究生,主要研究方向:核能与核技术工程。

## 基于 $\mu$ Clinux 的 FPGA 远程更新系统的实现

作者: 丁丁, 汤晓斌, 陈立德, 殷树根, Ding Ding, Tang Xiaobin, Chen Lide, Yin Shugen  
作者单位: 丁丁, 汤晓斌, Ding Ding, Tang Xiaobin(南京航空航天大学材料科学与技术学院, 江苏南京, 211106), 陈立德, 殷树根, Chen Lide, Yin Shugen(上海风格信息技术有限公司, 上海, 200120)  
刊名: 电子技术应用   
英文刊名: Application of Electronic Technique  
年, 卷(期): 2014, 40(3)

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_dzjsyy201403003.aspx](http://d.g.wanfangdata.com.cn/Periodical_dzjsyy201403003.aspx)