

Stratix 系列 FPGA 可实现远程升级的系统设计

王晓辉,刘玉婷

(凯迈(洛阳)测控有限公司 光电事业部,洛阳 471000)

摘要: 在通信、航空、航天、医疗等众多领域,FPGA 作为主控器已无可替代,基于 JTAG(Joint Test Action Group)的 FPGA 传统升级方式依赖专用的 JTAG 下载线缆和本地开发环境,加载稳定性差,传输速率低。本文提出基于 Stratix 系列 FPGA 远程升级实现技术方案,此方案保证基本硬件电路与传统升级方式相同,通过软件功能扩展即可实现对 FPGA 软件的更新,使得产品的调试和升级维护更加便利。工程实践证明,利用远程升级方式可以使升级效率提升为原来的 10 倍左右,该技术方案稳定、可靠。

关键词: JTAG 升级;远程升级;传输速率

中图分类号: TN99 **文献标识码:** A

Remote Updating System Design Based on Stratix Series FPGA

Wang Xiaohui,Liu Yuting

(Optoelectronic Business Unit,Kaimai (Luoyang) Measurement and Control Co.,Ltd.,Luoyang 471000,China)

Abstract: In many fields such as communication,aviation,aerospace and medical treatment,FPGA as the main controller is irreplaceable. The traditional update method of FPGA based on JTAG (Joint Test Action Group) has the problems of relying on special JTAG download cable and local development environment,poor loading stability and low transmission rate. This paper presents a technical scheme for remote update of FPGA based on Stratix series. This scheme ensures that the basic hardware circuit is the same as the traditional update method. The FPGA software can be updated through software function expansion,which makes the debugging,updating and maintenance of products more convenient. The engineering practice has proved that the updating efficiency is increased by about 10 times by using the remote updating method,and the technical scheme is stable and reliable.

Keywords: JTAG update;remote update;transmission rate

0 引言

图像跟踪器是机载吊舱中的关键组件,每次对图像跟踪器更新 FPGA 程序时,首先需要拆卸设备,独立加电,再利用在线编程的方法,经多人一系列合作后才可完成 FPGA 程序的更新。在升级过程中,势必造成较大的人力成本投入。另外,每次拆开系统会影响产品的保密性;同时,也会对系统稳定性造成一定的风险。因此,有必要设计新的 FPGA 软件升级方法。

本文研究了 Altera 公司(现被 Intel 收购)Stratix 系列 FPGA 的在应用中编程的方法,并讨论了其应用在图像跟踪器中实现 FPGA 远程升级的方法。IAP(In Application Programming)即在应用编程,是指编程过程不会打断正常应用,对用户来讲是一种静默的编程方式^[1],既不需要专门的工具,也不需要拆机。

Altera 公司 FPGA 器件的配置主要有 3 种:主动模式、被动模式和联合测试行动组^[2](JTAG)模式,其具体说

明如表 1 所列。

表 1 Altera FPGA 配置简表

配置模式	配置方法	实现方式
主动模式	主动串行	EPCS 系列
	主动并行	EPC 系列
被动模式	同步被动串行	通用存储器等+微处理器+EPLD
	异步被动串行	
	同步被动并行	
	异步被动并行	
JTAG 模式	JTAG 下载	直接主机下载或微处理器+EPLD

1 关键技术

(1) 串口波特率动态切换

为了在现有系统资源基础上实现在线软件升级,需复用现有的串口资源,在正常应用状态下,串口通信波特率

为 38 400 b/s, 维持现有波特率可以实现在线程序升级, 但需要消耗较长时间, 故提出一种动态切换波特率的方案, 进入在线升级模式时, 波特率切换为较高波特率 (如 115 200 b/s、230 400 b/s 等), 当退出在线升级模式时, 将串口波特率改回原有工作模式对应的值即可。通过测试发现, 提升波特率后, 在线升级需等待的时间会明显缩短 N 倍, N 取决于升级模式下的波特率与正常工作模式所用波特率的倍数, 本方案选择 115 200 b/s 和 38 400 b/s 进行动态切换, 升级时间由原有的 18 min 缩短到 6 min。需注意, 波特率不能过高, 过高的波特率对系统抗干扰性提出更高的要求, 若抗干扰性不够强, 则会导致丢包、重传, 影响时效性。

(2) 数据帧重传机制设计

为了保证接收程序文件的完整性, 一是需在上位机发送数据帧中添加帧标号, 终端设备在接收数据帧时判断帧标号的连续性。若是理想的帧标号, 则向上位机反馈当前帧接收成功且为有效帧; 若出现帧标号为非理想帧, 则向上位机反馈理想帧标号, 待上位机收到应答后, 分析成功与否, 并根据结果对数据进行重取或直接传输下一帧, 直到终端设备收到最后一帧, 并将成功结果反馈给上位机, 待上位机收到最后一帧反馈后, 升级结束。

(3) 应用模式与系统在线升级模式切换设计

为了适应二者的实时智能切换, 需将系统工作模式设计为应用模式和在线升级模式。正常运行模式下, 可以接收在线升级的请求命令, 一旦收到升级请求, 则将系统工作模式切换到在线升级模式, 此功能需上位机和终端设备配合实现。上位机中专门设计升级模式切换开关, 以消息帧形式进行传递, 终端设备收到升级消息帧后, 系统工作模式切换到在线升级状态, 等待接收上位机传送的数据帧, 待收到最后一帧数据并完成了数据文件向 Flash 的写入操作后, 向上位机反馈接收完成标识, 并切换工作模式到正常工作模式即可。

(4) Flash 器件数据写入策略设计

为了保证向 Flash 写入数据文件的完整性, 同时考虑到系统资源有限, 需在软件中设计了特定大小的缓存用于存储串口接收的程序数据。Flash 写入操作可以按扇区形式或字节形式写入, 由于字节写入效率太低, 本方案选择了前者, Flash 扇区大小为 512 KB, 故设计了大小为 512 KB 的数据缓存。当接收到有效的 512 KB 数据时, 可启动 Flash 写入操作流程, 待写入完成后清除缓存, 并向上位机反馈接收结果, 否则不启动 Flash 写入操作。

2 远程升级系统方案

Altera 各系列 FPGA 升级方案存在较大差异, 需要针对具体型号进行专门的方案设计。远程升级成功的关键

在于将升级数据文件准确无误地写入 Flash。通常 Altera 的 FPGA 远程升级方案设计分为: ①在 PL 端实现, 利用 ALT_REMOTE_UPDATE_IP、ALT_ASIM_PARALLEL_IP, 以及逻辑软件配合实现; ②在 PS 端实现, 利用 Qsys/SOPC 开发工具, 搭载 NIOS 软核系统, 利用串行接口通信实现远程升级 FPGA 程序; ③PL+PS 的混合开发模式, 即一部分功能在 PL 端实现, 另一部分则需借助于 PS 端的软核代码实现^[3]。

这 3 种方式各有优缺点, 开发者需根据自己所使用的 FPGA 型号, 通过查阅资料确定适合自己的方案。

考虑到当前公司产品硬件状态, 方案①和方案③均无法实现本型号 FPGA 的远程升级。故本文选择方案②——基于 NIOS 软核的高可靠 FPGA 远程升级方案, 后期会在硬件支持的情况下考虑其余两种升级方式。

2.1 物理接口

在 PC 和 FPGA 端实现数据交互, 可以采用 UART、以太网以及 SPI 等多种形式串行接口, 如图 1 所示。当前系统中对外交互的接口为 UART, 为不改变硬件状态, 更好地适应现有系统, 选择 UART 作为升级接口更合适, 也更容易实现。

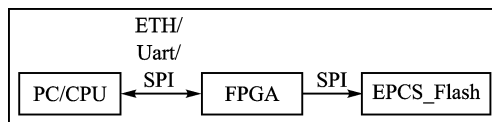


图 1 PC 和 FPGA 端数据交互框图

上位机通过 UART 将 PC 机和 FPGA 的软核或硬逻辑接口进行连接, 下载用户设计的新应用程序并通过擦除、重写 Flash 来达到更新 FPGA 程序的目的。

2.2 总体流程

FPGA 远程升级逻辑设计总体结构框图如图 2 所示, 主要由 4 个模块组成: 上位机 (Upper computer) 模块负责升级过程中命令的发出、状态的切换, 以及升级数据包的分包和发送等操作; 协议解析模块 (Protocol_analysis) 用于适配外部接口, 解析命令和数据后转换为本地总线结构, 访问控制各功能模块; 擦除更新控制模块 (Erase/Write) 用于接收升级控制信息和升级数据包后缓存, 并给出必要的指示信息及数据信息, EPCS 控制模块根据这些信息通过 SPI 协议读写控制 Flash, 将升级配置文件正确写入到 Flash 中。

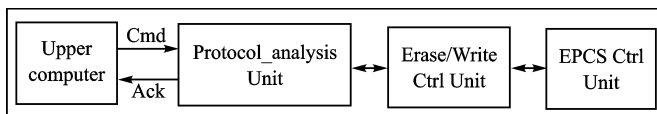


图 2 总体结构框图

2.3 PC 端处理流程

PC 端主要负责升级流程的发起、配置文件的获取,以及升级流程的控制与处理,具体流程如图 3 所示。

2.4 FPGA 端升级流程

FPGA 端默认运行在应用模式下,一旦收到升级请求命令,则会获取命令包中需要保存的升级文件 Size 参数,并立刻切换系统工作模式,进入到升级配置模式下,准备好接收有效的配置文件数据,当数据包到来

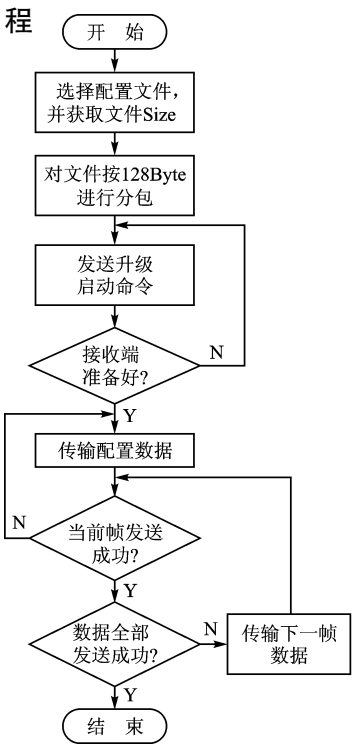


图 3 PC 端升级流程

时,主动判断数据包中帧号是否为当前帧号。若帧号不一致,则将理想帧号反馈给上位机;否则,将当前帧数据进行缓存。具体过程如图 4 所示。

2.5 配置文件生成流程

Altera 公司设计的 JTAG 软件在固化方式下文件格式为 jic 或 pof 两种。但在线软件升级模式下,最终需要将程序文件下载到 Flash 内,Flash 只认 bin 文件类型,故需要设计一种方法将原有的程序文件转换成 bin 格式并能正常运行。本文专门设计一种方法,即通过 Quartus 软件中的 Command shell 工具和 shell 脚本文件先将逻辑部分的 SOF 文件转换为 Flash 文件,然后将 elf 文件转换为 Flash 文件,并将二者拼接在一起,最后将 Flash 格式文件转换为 bin 文件,bin 文件生成图如图 5 所示。

2.6 升级异常处理

对于升级过程中的异常情况,PC 端的上位机软件和 FPGA 端通过分析数据处理流程和引发常见问题的异常情况分别设计了不同的异常处理机制来解决远程升级失败的问题。对异常处理机制进行了充分测试,并已在系统实践中验证了方案的可靠性。

3 测试结果与分析

3.1 测试环境

测试硬件环境为图像跟踪器中对应的 Stratix 系列

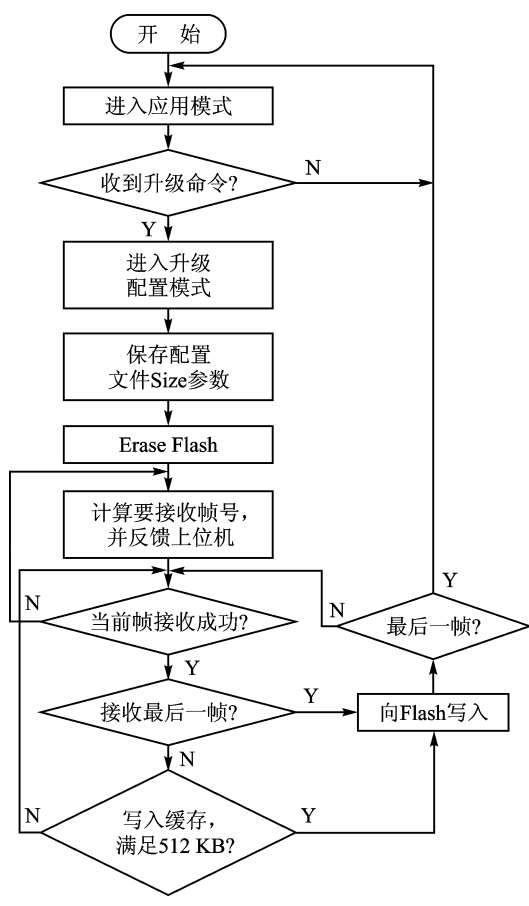


图 4 FPGA 端处理流程图

```
sof="D:\wxh\VC53_FAC_1\output_files\VC53_FAC.sof"
elf="D:\wxh\VC53_FAC_1\software\Simple1_test\Simple1_test.elf"

echo "sof>flash ..."; sof2flash --input=$sof --output=sof.flash --quiet

echo "elf>flash ..."; elf2flash --epcs --after=sof.flash --input=$elf --output=elf.flash

echo "cat flash ..."; cp sof.flash sof_elf.flash; cat elf.flash >> sof_elf.flash

echo "flashbin ..."; nios2-elf-objcopy --input-target srec --output-target binary sof_elf.flash sof_elf.bin

echo "del flash ..."; rm -f *.flash
```

图 5 bin 文件生成图

FPGA,上位机软件是自行开发产品调试软件。上位机通过获取 FPGA 反馈的运行状态验证升级成功与否。

3.2 测试用例及结果

在传统的升级方式下,鉴于图像跟踪器功能的特殊性,安装位置必须在吊舱舱体内部,每次对跟踪图像跟踪器 FPGA 程序进行升级时,都需要拆开球壳,尤其是在外场,需要将吊舱从飞机上拆下,然后拆开球壳,再进行升级,升级验证完成后,需再次将系统安装还原,整个过程异常繁琐复杂。而本文方式则只需借助上位机软件将生成的配置文件打开并加载后启动升级命令,即可自动完成升级过程。

```
USB Device Connected
USB Device Reset Completed
PID: 5740h
VID: 483h
Address (#1) assigned.
Manufacturer : EmuFlight
Product : CrazyBee F4 FR
Serial Number : 0x8000000
Enumeration done.
This device has only 1 configuration.
Default configuration set.
No registered class for this device.
access
USB Device disconnected
class:0x2
```

图6 飞控板连接时串口输出信息

由 MCU 调试串口输出的信息可以发现,STM32 的 USB 主机协议准确识别了上述 3 个 USB 设备的信息,得到上述 3 个设备的类型,根据程序禁止 HID 类的 USB 设备连接计算机,放行 MSC 类和 CDC 类的设备与计算机连接,有效防护了可能存在 badUSB 设备的 HID 类 USB 设备。判断成功后,USB 设备直接通过 PCB 线路与计算机直接相连,由于没有经过 USB 协议解析,所以通信的速率没有降低,不影响设备的正常使用。

4 结 语

本文利用 STM32 MCU 和 USB 双路复用芯片设计并实现了一种 badUSB 防护设备,实验结果表明,该防护设备能够有效防护可能存在 badUSB 的 HID 类的 USB 设备。本方案成本低、体积小、安全性高、通信速率高,但目前这个防护设备仅支持 USB2.0 及以下设备。后续可以改进设计,从而设备支持 USB3.2 和 USB4.0 协议,使应用更

69 测量结果如表 2 所列。通过传统 JTAG 方式完成一次图像跟踪器的 FPGA 升级需要两个小时,而利用本文的升级方式只需要 0.2 个小时,在效率上整整提升了 10 倍左右;另外,本文方案只需系统电缆连接正常即可,而不需要使用专用的仿真器,升级距离和稳定性都得到了较大提升。

表 2 测试比较结果

升级方式	电缆	距离/m	时间/h
传统方式	USB-JTAG 线缆	3	2
本文方式	RS422 串口线	1200	0.2

4 结 语

本文提出的基于 Stratix 系列 FPGA 可远程实现软件升级的方法,缩短了产品升级过程中消耗的时间,提升了

加广泛;此外,意法半导体提供了 AI 代码库,可以利用代码库进行训练,在 MCU 中部署一个神经网络模型,进一步识别 HID 类设备是否正常设备还是 badUSB 设备,以便于对 HID 类设备做更加详细的判断。

参考文献

- [1] Shin S, Gu G. Conficker and beyond: a large-scale empirical study[C]//Proceedings of the 26th Annual Computer Security Applications Conference, 2010: 151-160.
- [2] Nohl K, Lell J. Bad USB - On accessories that turn evil[C]//Black Hat USA, 2014: 1-22.
- [3] 陈威, 陈乐然, 徐小天, 等. USB 设备对电力工控系统的威胁[J]. 华北电力技术, 2016(7): 5.
- [4] 赵松银, 郁滨. USB 安全连接方案设计与实现[J]. 系统仿真学报, 2016, 28(6): 6.
- [5] Loe E L, Hsiao H C, Kim T H J, et al. Sand USB: An installation-free sandbox for USB peripherals[C]//2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). IEEE, 2016: 621-626.
- [6] Kang M, Saiedian H. USB Wall: A novel security mechanism to protect against maliciously reprogrammed USB devices[J]. Information Security Journal: A Global Perspective, 2017, 26(4): 166-185.
- [7] 张乃珩. 一种基于 Hypervisor 的 BadUSB 攻击防御方法[J]. 河南科学, 2020, 38(9): 1400-1408.
- [8] usb.org. Defined Class Codes[EB/OL]. [2023-06]. <https://www.usb.org/defined-class-codes>.

胡金亮(工程师), 主要研究方向为嵌入式系统、软件无线电、移动通信。通信作者: 胡金亮, predator_hjl@foxmail.com。

(责任编辑: 薛士然 收稿日期: 2023-06-16)

维修效率, 并将产品升级过程中造成的风险降到最低, 同时节约了系统的维护成本。

参考文献

- [1] 周琼, 周鹏. 单核 FPGA 系统的可靠性远程升级设计[J]. 中国新通信, 2018, 20(2): 47-49.
- [2] 刘永恩, 王俊芳. FPGA 远程升级技术的分析与实现[J]. 无线电工程, 2012, 42(9): 48-50, 60.
- [3] 吴冬冬, 杨晓君, 张佩珩. 一种 FPGA 的远程系统升级方法[J]. 计算机工程与应用, 2006, 18(1): 86-89.

王晓辉(硕士研究生), 主要从事图像信息处理及嵌入式系统设计与开发。通信作者: 王晓辉, xmwangxiaohui@163.com。

(责任编辑: 薛士然 收稿日期: 2023-05-17)