文章编号：2096-1618(2020)05-0493-06

一种基于 FPGA 的在线升级方案

万 垚，李 鑫
(成都信息工程大学通信工程学院，四川 成都 610225)

摘要：一般对 FPGA 程序升级，需要使用下载器通过 JTAG 接口与 FPGA 连接，在一些不方便开盖的环境下，此种升级方案非常困难。基于 XILINX 公司的 XC6SLX9 芯片，利用 FPGA 的 MultiBoot 将多个配置文件下载入 Flash 中的特性，介绍一种基于 Flash、FPGA 和 RS232 串行通信的在线升级方案。FPGA 通过 SPI 总线配置 Flash，ICAP 接口使 FPGA 跳转到 Flash 的对应地址读取烧写到 Flash 中的. bin 文件。方案可在不增加额外器件且不开盖的情况下仅通过一个 RS232 通信接口，上位机软件就能完成对用户设计的功能程序或产品程序的升级，对程序存储芯片 Flash 的操作均由 FPGA 内部逻辑实现。结果表明，该方法有效且具有很好的移植性和可扩展性。

关 键 词：SPI；Flash；RS232 串行通信；ICAP；MultiBoot
中图分类号：TP332.1 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2020.05.002

0 引言

如今，电子产品的复杂度和集成度越来越高，产品的迭代速度也在变快。FPGA 以其低时延、速度快和可重配置的优势已经被应用在通信、工业控制和图像处理等领域。这些领域的程序算法为适应产品需求的提升也在不断更新。因此对 FPGA 能进行在线升级从而快速更新产品功能的需求被提出。

传统的对 FPGA 配置的方案需要对设备进行拆卸，使用 JTAG 接口对 FPGA 进行重配置进而对程序进行升级。由于 JTAG 模式是将程序配置到 FPGA 的 SRAM，掉电丢失。Flash 作为非易失的存储器，可以用来存储程序，不过需用相应的开发工具将 FPGA 硬件综合生成的比特流文件转换为. msc 文件然后再烧写到 Flash 中。这种做法需要对产品进行拆卸且需要专业技术人员进行操作，升级成本大。为实现便捷地对 FPGA 相应产品的程序进行升级，提出一种用上位机软件的人机交互界面，将二进制格式的程序文件通过 SPI 总线固化到 Flash 中完成对 FPGA 进行在线升级的方法。此方法操作简单，只需厂家提供对应程序的二进制格式的升级包，用户自身就能完成对 FPGA 程序的升级，且掉电程序不丢失。此方案操作简单，用户自身即可完成，大大降低了升级成本。

1 系统方案设计

采用 Spartan-6 系列的 XC6SLX9[1] 芯片，该芯片有 144 个引脚，Flash 又称闪存，属于一种数据非易失的存储器，在掉电的情况下 Flash 内部数据可以得到保存，因此可以用其作为 FPGA 的配置芯片。SPI 总线可作为 FPGA 与 Flash 芯片通信的桥梁[2]，连接如图 1 所示。SPI 总线时序的模式有两种，如图 2 所示。

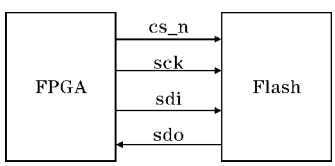

图 1 FPGA 与 Flash 连线示意图

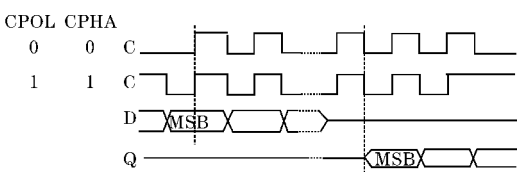

图 2 SPI 模式

SPI 总线由 4 根信号线构成[3]，分别为cs_n片选信号线，该信号线拉低表示 FPGA 选中对应的器件、Sck 为时钟信号线，SPI 总线就是根据此时钟传输数据。Sdi 和 Sdo 都为数据信号线，数据在此数据信号线上进行串行传输，在 Sdi 信号线上数据从 FPGA 芯片传输到 Flash 中，Flash 中的数据传输到 FPGA 中则是通过 Sdo 信号线。两根数据线上串行数据的切换和锁存由 Sck 时钟线控制。

选用的 Flash 芯片为 M25p16。M25p16 属于意法半导体公司生产的 Flash 芯片，其最大的时钟频率为 50 MHz。设计进行擦除时采用12.5 MHz的时钟，即 sck 的时钟频率为12.5 MHz。该款 Flash 芯片的存储空间为16 Mbit，每个地址内存储 1byte(8bit)的数据，

收稿日期：2019-11-26
基金项目：国家自然科学基金资助项目(61601065)

共2 M的存储深度,其地址分为 32 扇区(sector)、每个扇区包含 256 页(page)、每一页包含 256 字节(byte),因此该 Flash 芯片需要用到 21 位地址线,加上扩展的 3bit 地址线,一共用到 24bit 地址线。

系统包括上位机软件、串口、FPGA、Flash 芯片和 LED 灯,图 3 为系统框图。在图 3 中,当 FPGA 板卡接通电源,自动加载 Flash 中的程序开始配置 FPGA,同时 LED 灯亮,LED 灯点亮期间若 UART 接口接收到用户通过 PC 端的上位机软件发送的. bin 文件,即可完成对 Flash 中程序的在线升级。当 FPGA 板卡上电后,自动加载 Flash 中的程序开始配置 FPGA,同时 LED 灯亮,LED 灯点亮期间若 UART 接口接收到用户通过 PC 端的上位机软件发送的. bin 文件,即可完成对 Flash 中程序的在线升级。

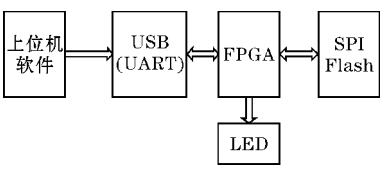

图 3 系统框图

2 FPGA 内部模块设计

FPGA 的顶层文件包括 10 个模块,图 4 为 FPGA 内部构成框图,复位信号可对 FPGA 全局进行复位,时钟的控制有 PLL 模块,FPGA 通过 SPI 总线发来的数据由串口接收模块进行串并转换后缓存到 FIFO 中。Flash 控制模块对来自串口接收模块的数据按照指定的数据包协议进行解析,使 FPGA 跳转到对应的擦除读写模块从而对外部的 Flash 进行擦除读写操作。程序升级计时配置模块预留给用户 20 秒的升级时间,在 20 秒内若检测到串口接收模块的擦除指令,即对 Flash 对应区域的程序进行擦除升级。重新上电板卡,20 秒内不对程序进行升级,FPGA 启动 ICAP 跳转,跳转到 Flash 对应区域加载程序。

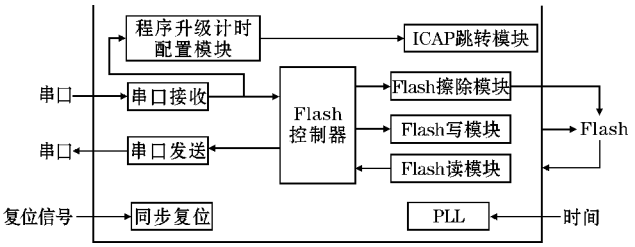

图 4 FPGA 内部构成框图

2.1 Flash 擦除模块设计

对 Flash 进行写入数据时,首先需要初始化 Flash,即对 Flash 中已经存在的数据进行擦除。Flash 的擦除分为扇区擦除(sector erase)和全擦除(bulk erase)[4]。

扇区擦除又称为 SE(sector erase),在进行 SE 之前需要给出一个写使能指令(WREN)[5],然后再进行扇区擦除。首先需要将 cs_n 置为低,然后产生 12.5 MHz的时钟 sck,Flash 根据 sck 的上升沿锁存 sdi 的数据,因此给出的 sdi 的每一位最好都能被 sck 的上升沿采集到稳定的时刻,即 sck 的上升沿对着 sdi 每一位数据的中心位置,WREN 的指令为 0x06。由于在 WREN 状态下只需要发送8 bit数据,因此只需要产生 8 个可以锁存 sdi 数据的 sck 上升沿即可,在确定 sdi 数据被锁存结束后,就可将 cs_n 拉高。图 5 为 Flash 扇区擦除仿真波形图,可以看出当来了擦除标志后,首先进入 WREN 状态,拉低cs_n。在 WREN 结束后,cs_n需要被拉高,为了确保 WREN 被 Flash 存储,cs_n 拉高的时间至少需要100 ns,之后再次拉低cs_n为发送 SE 指令做准备。进行擦除设计时首先拉低cs_n选中 Flash,然后发送 SE 指令(0xd8)以及 3 个 byte 的地址位,根据所发送的 bit 个数给出对应数量的 sck 上升沿,并确保每个 sck 的上升沿都能采集到 sdi 稳定的时刻。由于是扇区擦除,因此给出的地址位只有高8 bit 有效,低16 bit无论是何值都对该扇区的擦除没有影响。在给出最后1 byte地址位后,确保被选中的扇区可以被擦除,cs_n 需要保持至少 3 秒高电平的状态。

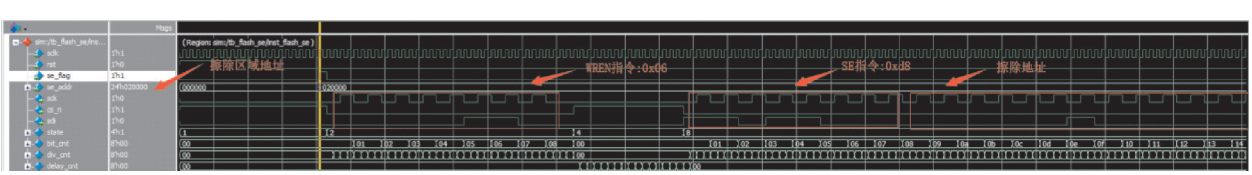

图 5 Flash 扇区擦除的仿真波形

2.2 Flash 写模块设计

Flash 页写功能,PP(page program)命令为 0x02[6]。同擦除指令一样,需要在发送 PP 指令之前,执行 Write EN 操作,当执行完 PP 后 Write EN 会自动复位,所以每次 PP 之前必须发送 Write EN 操作。模块中首先给写入页起始地址,将要写入 Flash 的一页数据缓存到 FIFO 中,缓存完毕后根据给出写 Flash 标志启动 Flash 页写操作,该功能只支持单页写,多页写可以在本模块外增加额外控制缓存和页写标志pp_flag

实现[7]。

Flash 写控制器时序与擦除控制器类似,这里写入 Flash 的字节数是由 PP 指令 1 字节、地址 3 字节和数据 256 字节组成,最后额外增加1 bit的cs_n为低的冗余周期,一共是 2081 个比特周期。在执行页写 PP 指令之前就把需要写入 Flash 一页的 256 字节的数据缓存到 FIFO 缓冲器中,当来了页写标志 pp_flag 就从 FIFO 缓冲器中读取一个字节数据通过 SPI 总线发送到 Flash 中,直到一个页数据发送完毕。从 FIFO 读取的 8 位数据提前一拍读取并赋值给移位寄存器,最高位即输出到 sdi 线中,其他 7 位需要按照 pp_shift_flag 标志移位到最高位并输出到 sdi 中。当 8bit 数据都移位到 sdi 数据线中的同时读取下一个 8 位数据,以此类推读取全部的 FIFO 中缓冲的 256 字节数据。图 6 为 Flash 写模块的仿真波形图。

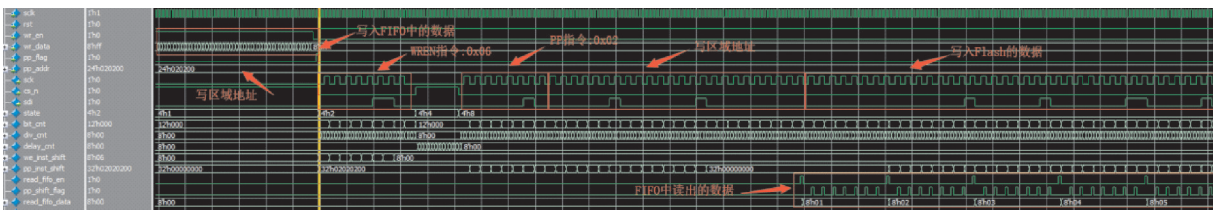

图 6　Flash 写模块的仿真波形图

2.3　Flash 读模块设计

Read data byte 读数据字节指令,简称 RDB 指令,指令码为 0x03[8],读指令可以读取 Flash 地址空间的任何一个字节数据,不受扇区和页的限制,而且可以连续读取出所有数据,读数据首先要给出读命令码,然后紧跟24 bit的起始地址,之后会在 sdo 行输出读出数据,直到 cs_n 中断读出数据操作,否则地址在 Flash 内部递增读出数据[9]。

只需给定读的起始地址和需要读取的数据的数量,读控制器就会自动从 Flash 读出数据缓存到 FIFO 中,以供其他模块读取调用。根据 Flash 读时序可知读操作无需前置 Write EN 操作,只需要分频计数器和比特计数器即可完成设计。通过串口发送的字节数推导出计数器的最终计数值,把 sdo 上的数据根据分频计数器和比特计数器进行移位锁存就可以实现串转并操作并完成 FIFO 的数据写入。图 7 为 Flash 读模块的仿真波形。

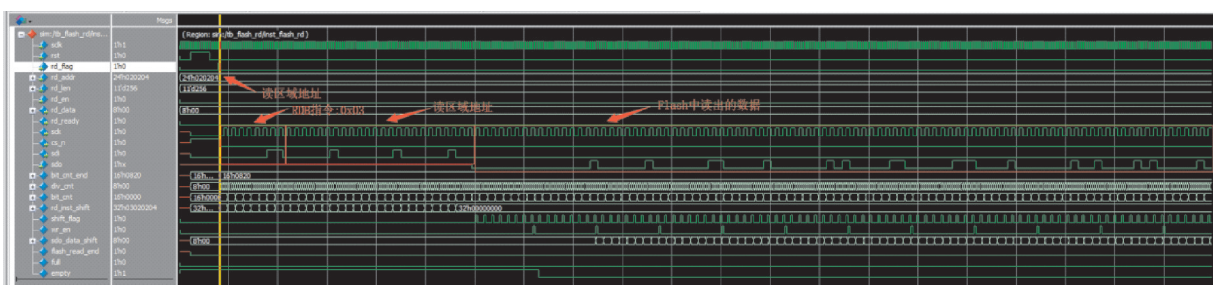

图 7　Flash 读模块的仿真波形

2.4　Flash 擦除读写控制器设计

完成了对 Flash 擦除和读写模块的设计后,还需设计一个对 Flash 的擦除和读写进行控制和仲裁的模块。模块中串口给 FPGA 发送指令,FPGA 解析指令后,进而对 Flash 进行对应的操作。

由于要与上位机串口交互,模块中制定了一种包括 3 种数据包类型的协议,分别为擦除命令包、写命令包和读命令包以及读数据返回包[10],如表 1 ~4 所示。当串口传递的数据为 0xcc 跳转到 WRITE 状态,WRITE 数据结束跳回 IDLE 状态。当串口传递的数据为 0xee 跳转到 ERASE 状态,完成擦除后跳回 IDLE 状态。当串口传递的数据为 0xdd 跳转 READ 状态,完成读取数据后跳回 IDLE 状态,命令解析状态机[11]如图 8 所示。

Flash 擦除读写控制器的功能框图如 9 所示,如何把 sck、cs_n、sdi、sdo 从 Flash 擦除(Flash_se)、读(Flash_rd)和写(Flash_pp)模块中引入 Flash 擦除读写控制模块(state_ctrl),并把这些信号在不同的时刻正确的路由到顶层接口与 Flash 通信是关键。

Flash 擦除模块和 Flash 写模块中 sck 信号在空闲无数据时间保持为 0 状态,而 cs_n 空闲无数据状态保持为 1,因此可以通过组合逻辑将两个信号输出到顶层。在 state = = READ 状态下正在读可以直接把 state

==READ 作为 Flash 读模块路由到顶层的条件。sdo 信号只有 Flash 读模块用到所以直接路由到 Flash 读模块中。

经过串口调试助手验证,发送相应的指令能够完成对 Flash 的擦除和读、写操作。

表 1 擦除命令包

字节编号	协议字段	定义
0	0xee	擦除包头
1	Addr[23:16]	擦除地址高字节
2	Addr[15:8]	擦除地址中字节
3	Addr[7:0]	擦除地址低字节

表 2 读命令包

字节编号	协议字段	定义
0	0xdd	读包头
1	Addr[23:16]	读地址高字节
2	Addr[15:8]	读地址中字节
3	Addr[7:0]	读地址低字节
4	Rd_len[15:8]	读长度高字节
5	Rd_led[7:0]	读长度低字节

表 3 读数据返回包

字节编号	协议字段	定义
0	Data0	第一字节
1	Data1	第二字节
2 ~ rd_len	Data2 ~ end	第三字节至最后的字节

表 4 页写命令包

字节编号	协议字段	定义
0	0xcc	写包头
1	Addr[23:16]	写地址高字节
2	Addr[15:8]	写地址中字节
3	Addr[7:0]	写地址低字节
4 ~ 259	W_data[7:0]	写数据 256 字节

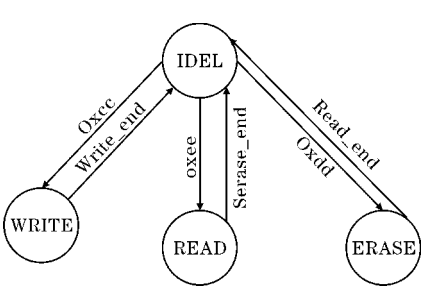

图 8 命令解析状态机

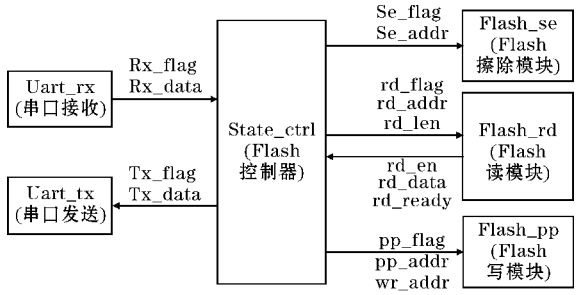

图 9 功能总体框图

2.5 ICap 在线升级 MultiBoot

Multiboot start address 是指通过 ICAP 接口使 FPGA 跳转到 Flash 的对应地址把程序的二进制文件烧录到 Flash 中来配置 FPGA[12]。当 MultiBoot start address 读取. bin 文件失败或跳转失败时,就回到 Fallback start address,这个地址设置为 0x000000,这样跳转失败回到 0 地址继续加载,如图 10 所示。区域 1 中 0x000000 为起始地址,FPGA 上电自动从此地址加载数据,区域 1 中的程序执行读写擦除 Flash,因此区域 1 中的程序就不能被擦除和升级,只对区域 2 的程序升级。区域 2 的地址为 0x100000。ICAP(the internal configuration access port)即内部配置访问端口[13],如图 11 所示。

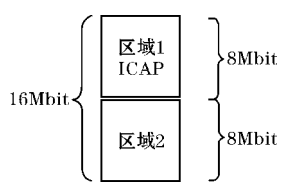

图 10 Flash 程序区域分布图

```
ICAP_SPARTAN6 #(
   .DEVICE_ID(0'h4000093),     // Specifies the pre-programmed Device ID value
   .SIM_CFG_FILE_NAME("NONE")  // Specifies the Raw Bitstream (RBT) file to be parsed by
                               // model
)
ICAP_SPARTAN6_inst (
   .BUSY(BUSY),   // 1-bit output: Busy/Ready output
   .O(O),         // 16-bit output: Configuartion data output bus
   .CE(CE),       // 1-bit input: Active-Low ICAP Enable input
   .CLK(CLK),     // 1-bit input: Clock input
   .I(I),         // 16-bit input: Configuration data input bus
   .WRITE(WRITE)  // 1-bit input: Read/Write control input
);

// End of ICAP_SPARTAN6_inst instantiation
```

图 11 ICAP 实例化原语

Device_id 指的是器件的编号,需要根据不同的芯片型号选择不同的 ID[14]。BUSY 和 O 属于输出,无需管理。CE(输入)为 ICAP 使能信号,低有效,在整个过程中需要保持为低;CLK(输入)时钟;I(输入)配置的数据,,需要按照表 5 步骤传输数据,如图 12 所示。其中 opcode 指的是器件 read 的命令(基于 SPI 的 Flash read 命令为 0x03)。在传输这些配置数据时,需要将这些配置数据按照 byte 为单位,进行高低位互换。按

照 ICAP 具体的操作指令指定相应的跳转地址,程序即可跳转到对应的存储空间[15]。

表 5　配置命令字

Configuration Data(hex)	Explanation
FFFF	Dummy Word
AA99	Sync Word
5566	Sync Word
3261	Type 1 Write 1 Words to GENERAL_1
XXXX	MultiBoot Start Address[15:0]
3281	Type 1 Write 1 Word to GENERAL2
XXXX	Opcode and MultiBoot Start Address[23:16]
32A1	Type 1 Write 1 Word to GENERAL3
XXXX	Fallback Start Address[15:0]
32C1	Type 1 Write 1 Word to GENERAL4
XXXX	Opcode and Fallback Start Address[23:16]
30A1	Type 1 Write 1 Word to CMD
000E	IPROG Command
2000	Type 1 NO OP

上电后自动加载区域 1 的程序,此时开始计时,如果 20 秒内没有检测到串口发送的擦除指令,即用户没有执行升级操作,ICAP 接口控制 FPGA 跳转到 Flash 的区域 2 中加载程序。如果希望再次升级的话必须重新给板卡上电使得程序回到区域 1 中。图 12 为 ICAP 接口的跳转程序的状态机。其中在 S_LOW_ADDR_BACK 状态下需给出区域 1 的起始低 16 位地址 0x0000;在 S_HIGH_ADDR_BACK 状态下给出 8 位读操作码和起始高 8 位地址 0x0300;在 S_LOW_ADDR 状态下需给出区域 2 的 16 位地址 0x0000,在 S_HIGH_ADDR 状态下需给出 8 位读操作码和高 8 位地址,这里为 0x0310。其余部分对照手册配置即可。

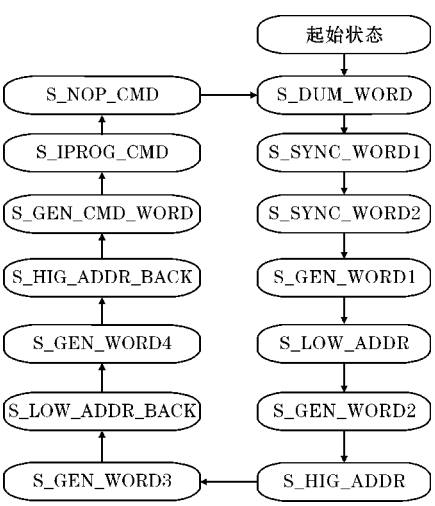

图 12　ICAP 指令状态机

3　实验结果

如图 13 上位机软件所示,首先通过上位机软件的加载 BIN 选项将 Flash 读写擦除控制程序的. bin 文件选中,镜像地址选择 0x000000 及对应烧写到 Flash 中地址为 0x000000 的区域,所有配置完成后点击下载 bin 的按钮即开始往 Flash 中下载程序,进度条实时显示下载进度。下载完成板卡重新上电后自动加载 Flash 此地址区域的程序。在预留的可升级时间内把需要升级的用户程序以相同的操作通过上位机软件固化到 Flash 的 0x100000 中,完成对 Flash 中程序的升级,如图 14 所示。选用的 Flash 芯片最大存储容量为 16Mbit,其中划分了 8Mbit 的空间存储用户程序,所以. bin 文件最大支持 8Mbit。此区域可以灵活划分,只要预留出足够区域存储 Flash 读写擦除控制程序即可。结果表明,板卡重新上电后,待升级时间结束,FPGA 会自动加载升级后的程序。实验表明,该设计操作简单,能够便捷的完成对 FPGA 的在线升级工作。

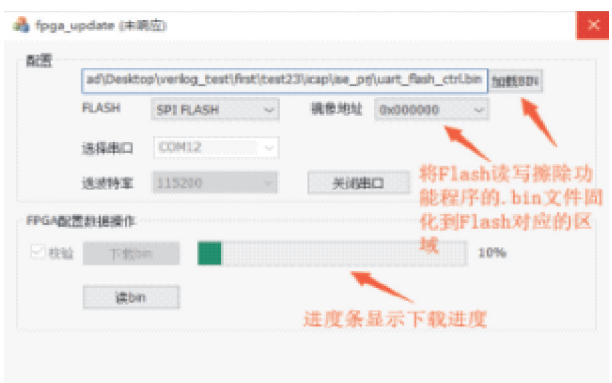

图 13　固化 Flash 擦除读写程序

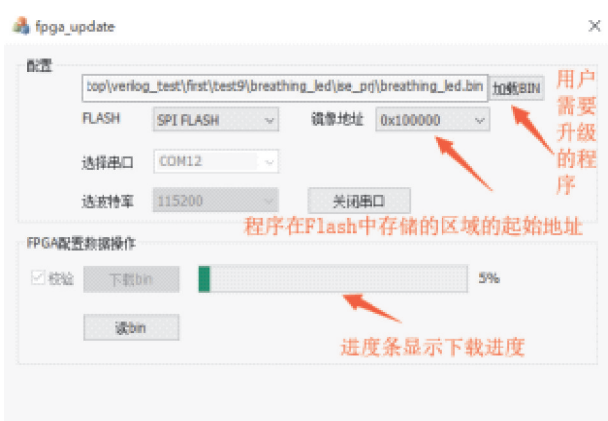

图 14　固化升级程序

4　结论

提出一种用上位机软件的人机交互界面,将二进

制格式的程序文件通过 SPI 总线固化到 Flash 中完成对 FPGA 进行在线升级的方法。可在不增加额外器件且不开盖的情况下仅通过一个 RS232 通信接口,上位机软件就能完成对用户设计的功能程序或产品程序的升级。该方法有效且具有很好的移植性和可扩展性。

参考文献:

[1] 陈炳成. 基于 FPGA 的 SPI Flash 控制器的设计与实现[J]. 电子世界,2013(12):137+231.

[2] 陈明义,连帅军,周建国. 基于 FPGA 的 FLASH 控制器系统设计及实现[J]. 电子科技,2008(7): 11-13+16.

[3] 刘俊. 基于 FPGA 的 FLASH 控制器的设计[J]. 电子技术与软件工程,2016(23):125-126.

[4] 罗莉,夏军,邓宇. 通用 SPI Flash 控制器的设计与验证[J]. 计算机工程,2011,37(8):22-24+27.

[5] 赵庆平,李素文,杜伟宁,等. 基于 FPGA 的 SPI 接口 Flash 控制器设计及其在存储配置数据中的应用[J]. 吉林大学学报(理学版),2014,52(5):1022-1026.

[6] 贾嘉,王新安,雍珊珊. 基于 W25Q80BL 的 FPGA 配置控制器的设计与验证[J]. 电子器件,2014,37(3):474-477.

[7] 张立为,钟慧敏. 实现基于 FPGA 的 SPI Flash 控制器设计[J]. 微计算机信息,2010,26(17):124-126.

[8] 林天静,阮翔,刘春. 基于 Flash 控制器的 FPGA 在线加载功能设计[J]. 电子技术应用,2019,45(1):88-91.

[9] 关珊珊,周洁敏. 基于 Xilinx FPGA 的 SPI Flash 控制器设计与验证[J]. 电子器件,2012,35(2):216-220.

[10] 冯明发,卢锦川. 基于 FPGA 的 NAND Flash 控制器设计[J]. 煤炭技术,2010,29(11):201-202+205.

[11] 徐立国,李德建,于宝东,et al. 一种支持在线升级的 NOR Flash 控制器设计[J]. 电子技术应用,2019,45(10):50-57.

[12] 刘钊,杜永锋,许知博. 基于 Xilinx-Spartan6 FPGA 的 MultiBoot 设计的实现[J]. 电子科技,2012,25(3):28-31.

[13] 李平,吴晓,山寿. 基于 SPI FLASH 的 FPGA 多重配置[J]. 现代电子技术,2013,36(22):127-130.

[14] 邱金蕙,李振全. 基于 Max Ⅱ CPLD 和 Flash 实现 FPGA 的多重配置[J]. 河北科技大学学报,2008(2):158-160.

[15] 张江伟. 基于 Virtex-5 和 FLASH 实现 FPGA 的多重配置[J]. 计算机与网络,2012,38(Z1):130-132.

An Online Upgrade Scheme based on FPGA

WAN Yao, LI Li

(College of Communication Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: In general, to upgrade FPGA program, we need to use the downloader to connect with FPGA through JTAG interface, which is very difficult when it is inconvenient to uncover the device. Based on XC6SLX9chip of Xilinx company, this paper introduces an online upgrade scheme based on flash, FPGA and RS232 serial communication, which uses FPGA's MultiBoot to download multiple configuration files into flash. FPGA configures flash through SPI bus, and ICAP interface makes FPGA jump to the corresponding address of flash to read. bin file written into flash. This scheme can upgrade the function program or product program designed by the userby the Upper computer software through only one RS232 communication interface without adding additional devices and opening the cover. Through only one RS232 communication interface the upper computer software can upgrade the function program or product program designed by the user at which in this scheme without the additional devices and without opening the cover. Inthis scheme, the operation of program memory chip Flash is implemented by FPGA internal logic. The result shows that this method is effective and has good portability and expansibility.

Keywords: SPI; Flash; RS232 serial communication; ICAP; MultiBoot