

FPGA多重配置在LED显示控制系统中的应用

曹慧, 程宏斌, 汪洋, 郑喜凤

引用本文:

曹慧, 程宏斌, 汪洋, 等. FPGA多重配置在LED显示控制系统中的应用[J]. 液晶与显示, 2020, 35(5): 456–463.

CAO Hui, CHENG Hong-bin, WANG Yang, et al. Application of FPGA MultiBoot in LED display control system[J]. *Chinese Journal of Liquid Crystals and Displays*, 2020, 35(5): 456–463.

在线阅读 View online: <https://doi.org/10.3788/YJYXS20203505.0456>

您可能感兴趣的其他文章

Articles you may be interested in

基于FPGA的LED屏控制器设计

Design of LED screen controller based on FPGA

液晶与显示. 2015(5): 838–843 <https://doi.org/10.3788/YJYXS20153005.0838>

基于4G-LTE的脱机LED控制系统的设计

Design of off-line led control system based on 4G-LTE

液晶与显示. 2018, 33(1): 55–60 <https://doi.org/10.3788/YJYXS20183301.0055>

特定需求液晶显示控制器的设计实现

Design and implementation of LCD controller with special requirement

液晶与显示. 2015, 30(1): 109–113 <https://doi.org/10.3788/YJYXS20153001.0109>

基于FPGA的BiSS-C协议编码器接口技术研究及解码实现

Research and realization of BiSS-C protocol encoder interface based on FPGA

液晶与显示. 2016, 31(4): 386–391 <https://doi.org/10.3788/YJYXS20163104.0386>

基于FPGA的TFT液晶显示时序控制器设计

FPGA-based TFT LCD timing controller design

液晶与显示. 2015, 30(4): 647–654 <https://doi.org/10.3788/YJYXS20153004.0647>

文章编号:1007-2780(2020)05-0456-08

FPGA 多重配置在 LED 显示控制系统中的应用

曹 慧¹, 程宏斌^{1,2*}, 汪 洋^{1,2}, 郑喜凤^{1,2}

(1. 长春希达电子有限公司 吉林 长春 130103;

2. 中国科学院 长春光学精密机械与物理研究所 吉林 长春 130033)

摘要:在 LED 显示控制系统更新 FPGA 程序和显示参数过程中,可能会遇到掉电或误擦除等突发情况,导致程序出错后无法正常工作。为了提高系统的可靠性,本文对 FPGA 程序备份、系统功能自我恢复进行了研究。首先介绍了 Xilinx Spartan-6 FPGA 多重配置原理,然后针对 FPGA 程序和显示参数设置功能的需求,通过硬件设计研究了采用 Flash 多区域程序、显示参数备份存储方法。接着,通过 FPGA 多重配置模块的详细程序设计,实现了程序备份自动调用、自动判断和热重启的功能。最后,介绍了 FPGA 主备程序固化文件生成方式。通过模拟 FPGA 程序更新错误实验,验证了系统调用 ICAP 设置寄存器并触发 IPROG 命令和 FPGA 热启动并调用备份程序重新配置,实现了 FPGA 多重配置在 LED 显示控制系统上的实际应用,满足了系统更新和显示参数设置过程的稳定性、可靠性要求。

关键词: FPGA; 多重配置; LED 显示; 控制系统

中图分类号: TP273 **文献标识码:** A **doi:** 10.3788/YJYXS20203505.0456

Application of FPGA MultiBoot in LED display control system

CAO Hui¹, CHENG Hong-bin^{1,2*}, WANG Yang^{1,2}, ZHENG Xi-feng^{1,2}

(1. Changchun Cedar Electronics Technology Co., Ltd., Changchun 130103, China;

2. Changchun Institute of Optics, Fine Mechanics and Physics,
Chinese Academy of Sciences, Changchun 130033, China)

Abstract: During the process of updating the FPGA program and display parameters of the LED display control system, unexpected situations such as power failure or incorrect erasure may be encountered, which can result in the system to fail to work. In order to improve the reliability of the system, the FPGA program backup and system self-recovery are studied in this paper. Firstly, the MultiBoot configuration principle of Xilinx Spartan-6 FPGA is introduced. Then, according to the requirements of FPGA program and display parameter setting, the flash memory multi-region program and display parameter backup storage method are studied through the hardware design. Through the detailed programming of the FPGA MultiBoot configuration module, automatic call backup, automatic judgment, and hot restart of the program are realized. Finally, the generation method of the FPGA main and

收稿日期:2019-08-01;修订日期:2019-11-09.

基金项目:十三五国家重点研发计划(No.2017YFB0404800);吉林省重点科技研发项目(No.20180201116GX)

Supported by National Key Research and Development Program of China(No.2017YFB0404800); Jilin Province Key Science and Technology Research and Development Project (No.20180201116GX)

* 通信联系人, E-mail: chenghb@ccxida.com

backup program firmware files is introduced. By simulating the FPGA program update error, it is verified that the system could call the ICAP setting register to trigger the IPROG command, and make sure the FPGA hot start to call the backup program to reconfigure. As a result, the stability and reliability of the LED control system updating and display parameter setting process are met by the practical application of FPGA MultiBoot configuration.

Key words: FPGA; MultiBoot; LED display; control system

1 引言

常规 LED 显示控制系统主要由上位机、发送卡和接收卡组成,通过网络链路,实现 LED 显示屏的显示和控制。发送卡接收视频和控制信号,并将数据打包传输接收卡。接收卡截取自己需要的数据完成灰度调制或控制响应,并通过千兆网将数据串行转发给下一张接收卡^[1-3]。

在指挥、监控中心等工程项目中,LED 显示屏要求(7×24)h 不间断运行,因此确保控制系统程序升级正确越发重要。系统通过网络传输更新发送卡、接收卡外置 Flash 存储的程序文件,在过程中存在因干扰产生错误数据或遇到掉电、误擦除等破坏程序文件完整性的情况,致使现场可编程门阵列(Field Programmable Gate Array, FPGA)^[4-12]不能再次调用,人工逐一板卡烧录程序费时费力,造成重大显示故障。通过 Xilinx Spartan-6 FPGA 的 MultiBoot 特性,完成 FPGA 多重配置设计,在 Flash 某一固定区域存储备用程序,当 FPGA 更新程序出错时,调用备用程序保障常规显示和网络链路功能,随后在线更新程序,提高控制系统的容错率。

2 多重配置原理

Spartan-6 FPGA 在通电时从外部 Flash 加载存储的项目程序文件(mcs),完成 FPGA 的配置后执行系统功能。mcs 文件一般只包含一个比特流配置文件,但也允许有多个。Spartan-6 FPGA 具有 MultiBoot 逻辑配置模块,来实现 MultiBoot 和 Fallback 操作。MultiBoot 操作可以有选择地从 Flash 的多个比特流文件中加载其中的一个,通过 ICAP 设置寄存器并触发 IPROG 命令,在系统不重复上电情况下对 FPGA 重新配置^[4-6],实现不同的功能。Fallback 操作是在切换

加载比特流文件过程中出错时,用预存的安全比特流文件替代并进行 FPGA 的重新配置。

MultiBoot 逻辑配置流程中有 Header、MultiBoot 和 Golden 3 个比特流配置文件,此时外部 Flash 存储器中的比特流文件组织模式及 FPGA 配置流程如图 1 所示。

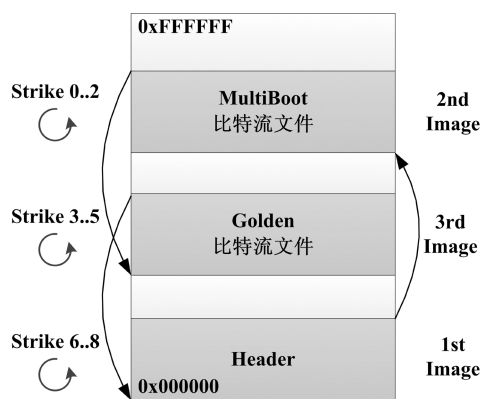


图 1 MultiBoot 逻辑配置流程

Fig.1 MultiBoot logic configuration process

Header 配置文件包含同步字、MultiBoot 和 Golden 配置文件在 Flash 存储的起始地址等信息^[7]。系统上电时,FPGA 从 0 地址加载 Header 配置文件,根据存储的地址跳转并加载 MultiBoot 配置,这是满足用户使用的系统常规程序。Golden 配置文件为备用的安全程序,当加载 MultiBoot 配置出错时,触发 Fallback 操作,加载该安全程序配置 FPGA。

如图 1 所示,Header、MultiBoot 和 Golden (Fallback)配置文件在加载过程中出错时,根据相应寄存器内设置的起始地址值 FPGA 会重新加载 3 次,加载次数的计数器名称为 Strike。若 Strike 的计数值为 6,则从 0 地址开始,FPGA 执行 Header→MultiBoot→Golden (Fallback)→Header 的循环加载。若期间 MultiBoot 或 Golden(Fallback)配置文件加载成功,则停止循环。每次循环 Strike 值加 1,最多 3 次,Strike 值

为 9 则停止所有配置加载。

Fallback 操作一旦被触发且加载 Golden 配置成功,只有重启系统或使用 IPROG 命令才能实现重新加载其他 FPGA 配置文件。MultiBoot 特性提高了器件利用率及系统安全性、降低了复杂系统的成本。

3 系统应用设计

LED 显示系统方案设计如图 2 所示,增加 MultiBoot 控制模块,采用状态机编码,按照指令流程通过 Xilinx 内置的 ICAP 核向配置文件发送 IPROG 命令,热启动调用 Flash 中存储的备用程序,实现对 FPGA 重新配置。

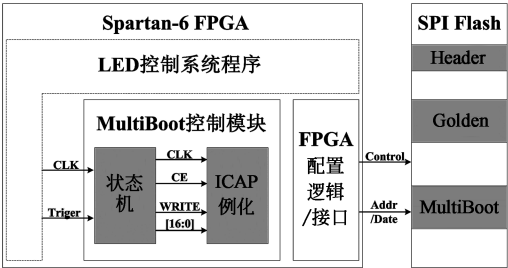


图 2 系统方案设计
Fig.2 System scheme design

3.1 硬件设计

采用 FPGA 型号为 Xilinx Spartan-6 Lx16、Flash 型号 M25P16,FPGA 与 SPI Flash 物理连接的硬件电路如图 3 所示,数据宽度为×1 SPI 配置的方式, $M[1:0] = 2'b01^{[8-9]}$ 。

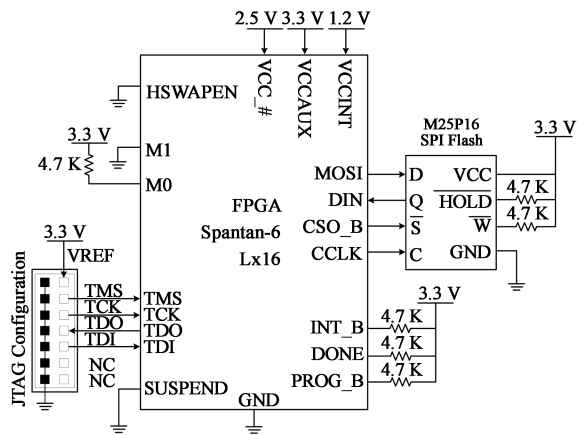


图 3 配置接口硬件电路
Fig.3 Configuration interface hardware circuit

LED 显示控制系统针对不同的显示屏需要进行相匹配的显示参数设置,在 SPI Flash 内采用显示参数设置与程序运行功能相独立地址存储,系统上电时加载程序比特流文件和显示参数设置文件,完成 FPGA 配置后实现不同显示屏的正常显示。

发送卡、接收卡的 Flash 存储区分为程序文件区和设置文件区。程序文件区保存实现各自功能的程序比特流数据;发送卡设置文件主要包含视频信号截取、屏幕带载分辨率、数据分包编码信息等;接收卡设置文件主要包含单卡带载分辨率、级联编码、显示灰度和刷新率、Gamma 表、校正系数表等。

SPI Flash 的容量选择与系统设计的加载模式和 FPGA 芯片的型号有关。在发送、接收端采用相同的主备两种 FPGA 配置加载设计方案,SPI Flash 内存储 Header、主程序 MultiBoot、备用程序 Golden 比特流文件和 2 个程序附属显示设置文件(Config.hex)。所用型号为 M25P16,16 Mbit(0x000000-0x1FFFFFF)容量的 Flash 详细的存储空间如表 1 所示。

表 1 Flash 存储空间
Tab.1 Flash storage space

名称	发送卡	接收卡	地址空间
Header	Header	Header	000000
FPGA 备用配置	Golden 备用程序	Golden 备用程序	000044
			07FFFF
	设置文件区	设置文件区	显示参数
			080000
			08FFFF
			GAMMA 表
			090000
FPGA 主配置	MultiBoot 主程序	MultiBoot 主程序	09FFFF
			0A0000
	设置文件区	设置文件区	校正系数表
			0FFFFF
			100000
	显示参数	显示参数	17FFFF
			180000
			18FFFF
			GAMMA 表
FPGA 主配置	设置文件区	设置文件区	190000
			19FFFF
			校正系数表
FPGA 主配置	亮色度表	校正系数表	1A0000
			1FFFFF

3.2 软件设计

软件设计主要分为两个部分:第一部分是在 FPGA 上开发的应用模块。实现系统发送卡、接收卡各自的功能,同时也为配置重载模块提供时

钟信号和触发信号,本文采用在线 Flash 程序写入完成后输出的标识信号作为触发信号。第二部分是 FPGA 配置重载模块。包括 Header 文件编写、MultiBoot 和 Golden 程序文件中例化 ICAP 核,满足触发条件后,发送 IPROG 命令,实现 FPGA 重新配置。程序备份启动及升级流程如图 4 所示。

根据程序运行流程,Header 配置文件作为启动引导作用,参照 Xilinx Spartan-6 指令对内部相关寄存器的规则^[7](如表 2 所示),逐条编译指令数据形成 Header 配置文件并保存为 HEX 形式,图 5 为向内部寄存器 GENERAL2 写入指令数据的详解示例。

MultiBoot 和 Golden 程序功能完全相同,需要注意的是:Golden 备用程序内对应上位机程序升级指令的 Flash 擦除、写入的偏移地址为 MultiBoot 主程序地址,且各程序在 SPI Flash 起始地址需要与 Header 程序文件内地址设置值对应。在主备程序的 MultiBoot 控制模块中同时例化 ICAP 核, Xilinx 的编译平台 ISE 中调用 ICAP 原语路径为 Edit → Language Templates, Verilog → Device Primitive Instantiation → Spartan-6 → Config/BSCAN Components → Internal Configuration Access Port (ICAP _ SPARTAN6),调用接口如下:

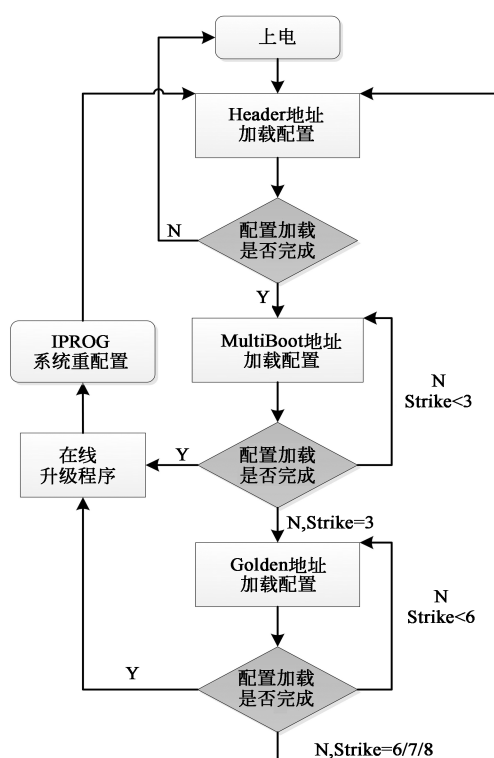


图 4 程序运行流程图

Fig.4 Program operation flowchart

```
ICAP_SPARTAN6_inst (
    .BUSY(BUSY), // Busy/Ready output
    .O(O),       // 16Bit Configuration data output
    .CE(CE),     // ICAP Enable input(Active-Low)
    .CLK(CLK),   // Clock input
    .I(I),       // 16Bit Configuration data input
    .WRITE(WRITE) // Read/Write control input
);
```

表 2 Header 配置文件内指令

Tab.2 Header configuration directive definition

序号	配置数据	指令说明
1	FFFFFFFF	空闲
2	AA995566	同步字
3	32610000	0x3261=向 GENERAL1 写 1 个 TYPE1 的字数据; 0x0000=MultiBoot 程序的低 16 位地址[15:0]

续 表

序号	配置数据	指令说明
4	32810310	0x3281=向 GENERAL2 写 1 个 TYPE1 的字数据; 0x03=SPI×1 读指令操作码, 0x10=MultiBoot 程序的高 8 位地址[23:16]
5	32A10044	0x32A1=向 GENERAL3 写 1 个 TYPE1 的字数据; 0x0044=Golden 程序的低 16 位地址[15:0]
6	32C10300	0x32C1=向 GENERAL4 写 1 个 TYPE1 的字数据; 0x03=SPI×1 读指令操作码, 0x00=Golden 程序的高 8 位地址[23:16]
7	32E10000	0x32E1=向 GENERAL5 写 1 个 TYPE1 的字数据; 0x0000=用户存储和访问故障安全方案所需的信息
8	30A10000	0x30A1=向 CMD 写 1 个 TYPE1 的字数据; 0x0000=空操作命令
9	33012100	0x3301=向 MODE_REG 写 1 个 TYPE1 的字数据; 0x2100=设置 NEW_MODE 位值[13]=0, BUSWIDTH 位值[12:11]=00, BOOTMODE[10:8] 位值=001, BOOTVSEL 位值[7:0]=0
10	3201201F	0x3201=向 HC_OPT_REG 写 1 个 TYPE1 的字数据; 0x201F=设置 INIT_SKIP[6]位值=0
11	30A1000E	0x30A1=向 CMD 写 1 个 TYPE1 的字数据; 0x000E=IPROG 命令
12	20002000	TYPE1 类型的空操作命令
13	20002000	TYPE1 类型的空操作命令

TYPE1 数据包						
HEX	32810310					
	包头16位			数据16位		
BIN	001	10	010100	00001	00000011	00010000
位宽	[15:13]	[12:11]	[10:5]	[4:0]	[15:8]	[7:0]
指令说明	TYPE1 标识	写操作指令	GENERAL2 寄存器标识 0×14	操作字节数为1	GENERAL2 寄存器SPI×1 读模式	GENERAL2 寄存器高8位 地址为0×10

图 5 指令数据详解示例
Fig.5 Detailed examples of command data

在使用该接口时,通过如图 2 所示的 Multi-Boot 控制模块中状态机只对 CE、CLK、I、WRITE 4 个信号不断赋值来发送指令数据,不用关注 BUSY、O 的值^[7]。ICAP 写数据的指令时序图如图 6 所示。

按照 Xilinx Spartan-6 指令对内部相关寄存器的规则,通过状态机完成表 2 中序号 1→2→8→11→12→13 的 6 个指令 ICAP 赋值,生成

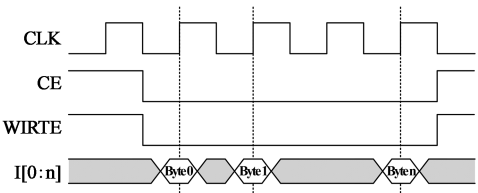


图 6 ICAP 写数据指令时序图(I 总线宽度为 16 bit)
Fig.6 ICAP write data instruction timing chart (I bus width is 16 bit)

IPROG 指令。ICAP_SPARTAN6 数据总线宽度为 16 bit,需要把引用的 6 个 32 bit 指令数据依次拆分为 16 bit 数据,以主备程序内 Flash 写入完成标识 Spi_wda=1 为触发信号,实现主备程序在线更新后按照运行流程自动重启对 FPGA 重新配置。写 IPROG 指令状态机如图 7 所示。

在状态机发送 IPROG 指令数据到 ICAP 核

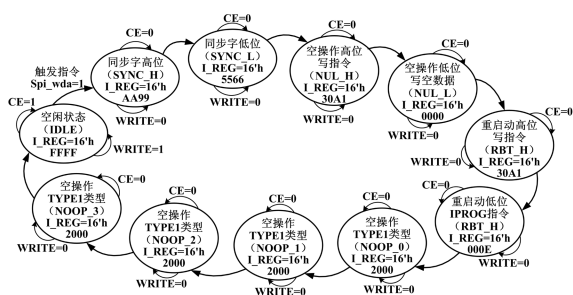


图7 写 IPROG 指令状态机

Fig.7 Write IPROG instruction state machine

的过程中,需要先将 16 bit 位宽度数据分为高低各 8 bit 位宽字,在每 8 bit 位宽字内进行一次位转换(Bit Swapping)操作,确保 ICAP_SPARTAN6 核接收到位转换后的正确指令。位转换操作参照如下 Verilog 语句:

```
wire [15:0] I; // 进行 Swap 之后的有效数据
```

```
reg [15:0] I_REG; // 进行 Swap 前的 I 寄存器数据
```

```
assign I = {I_REG[8], I_REG[9], I_REG[10], I_REG[11], I_REG[12], I_REG[13], I_REG[14], I_REG[15], I_REG[0], I_REG[1], I_REG[2], I_REG[3], I_REG[4], I_REG[5], I_REG[6], I_REG[7]}; // Flash 主 SPI 模式下,数据需要按照每 8 bit 进行一次 Swap 操作(即每个字 8 个 Bit 数据进行高低位互换)。
```

同理,按照上述方法,增加外部触发信号,参照表 2 中所有指令数据,增加、修改图 8 状态机流程,则可以在 MultiBoot 控制模块中根据内部触发、外部触发信号输出给 ICAP 核不同数据,实现系统 FPGA 被动和主动两种不同的热启动重新配置方式。

4 生成固化文件

Xilinx Spartan-6 系列的 FPGA 需要将后缀名为 mcs 的内存镜像文件固化到外部配置存储器中,上电后 FPGA 自动加载并完成配置。本文的 mcs 文件包含 3 个比特流文件和 2 个 HEX 文件,在整合到 mcs 文件的过程中,需要指定每个比特流、HEX 文件在 Flash 中存放的物理地址,且必须与系统程序设计内有关 Flash 擦除、写入、读取等指令设置的起始地址相同,这样 FPGA 才

能根据地址找到对应的程序和数据。

采用 Xilinx ISE 平台完成主备两个程序的工程项目文件设计后,通过软件的 Generate Programming File 生成主程序 MultiBoot.bit 和备程序 Golden.bit。把相关文件放在同一目录下,使用 ISE 平台内的命令行窗口 ISE Design Suite 32/64 Bit Command Prompt 工具^[10],按照以下步骤完成 Flash 存储芯片中初始可烧录的内存镜像(mcs)文件。

(1)promgen -p bin -r Header_SPIx1.hex -o Header_SPIx1.bin。如图 8 所示,把编写后的 Header 配置文件进行格式转换。

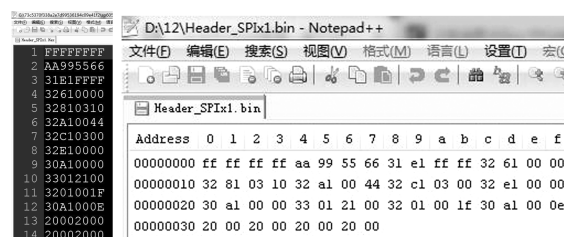


图8 Header 配置文件格式转换

Fig.8 Header configuration file format conversion

(2)promgen -w -p mcs -spi -s 2048 -data file up 000000 Header_SPIx1.bin -u 000044 Golden.bit -data file up 080000 Config.hex -u 100000 MultiBoot.bit -data file up 180000 Config.hex -o D:/H_M_G.mcs。

命令中各参数说明:

— spi: 生成的 mcs 文件要固化到 SPI Flash 中;

— p mcs:生成文件的格式是 mcs 格式;

— s 2048:定义外部存储器的容量,2 048 kbytes $\times$ 8/1 024=16 Mbit;

— data_file up 000000 Header_SPIx1.bin:名字为 Header_SPIx1 的 BIN 文件在 Flash 的起始地址为 0x0;

— u 000044Golden.bit:名字为 Golden 的比特率文件在 Flash 的起始地址为 0x44;

— u 100000MultiBoot.bit:名字为 MultiBoot 的比特率文件在 Flash 的起始地址为 0x100000;

— data_file up 080000 Config.hex:名字为 Config 的 HEX 文件在 Flash 的起始地址为 0x80000;

— data_file up 180000 Config.hex:名字为

Config 的 HEX 文件在 Flash 的起始地址为 0x180000;

—o D: /H_M_G.mcs;合成后的文件名和保存的目录。

合并完成 mcs 文件后,通过 ISE 设计平台中 IMPACT 软件将 mcs 文件固化到 SPI Flash 中。

5 测试分析与结果

通过 ISE 平台中集成的 Simulation 软件对设计进行仿真,通过添加 Test Bench,模拟激励信号为升级程序写完 Flash 的标识信号 Spi_wda,对 MultiBoot 控制模块进行仿真。图 9 是 Flash 重新加载时对 ICAP 核输入信号的仿真结果,可以看到,状态机成功生成了 IPROG 命令,并将位转换后的数据发送给 ICAP 模块。



图 9 ICAP_SPARTAN6 IPROG 命令仿真

Fig.9 ICAP_SPARTAN6 IPROG command simulation

如图 10 所示,本设计通过 LED 显示平台进行主备升级程序的测试。测试过程及结果如下:

升级程序格式转换为 HEX 文件,由上位机



图 10 LED 显示平台(软件、控制系统、显示屏)

Fig.10 LED display platform (software, control system, display)

软件通过网络传输升级指令和程序数据,分别发送给 LED 控制系统的发送卡、接收卡。通过修改程序 HEX 文件内数据和在上位机传输数据过程中拔掉网线、掉电,模拟升级过程中出现的突发情况,对发送卡、接收卡进行独立的测试。通过 LED 显示屏能否正常显示图像和系统指示灯状态观测 FPGA 是否正确加载配置。

系统主备程序的显示功能一致,显示配置文件一致,主程序工作状态指示灯闪烁频率是备程序的 2 倍。在系统主程序正常工作时,模拟主程序升级故障,观测到板卡自动重启后系统指示灯从高频变为低频闪烁,LED 显示屏显示正常,确认板卡备程序自动启用;此时正常进行程序升级,观测到板卡自动重启后系统指示灯由低频变为高频闪烁,LED 显示屏显示正常,确认板卡主程序升级成功并自动启用;当备程序已经启用,模拟升级故障时,观测到板卡自动重启后系统指示灯维持低频闪烁状态,LED 显示屏显示正常,确认板卡主程序升级失败,自动保持启用备程序。

6 结 论

本文 LED 显示控制系统的发送卡、接收卡程序采用主备程序热启动设计,在 LED 显示屏使用过程中极大地保障了系统在线升级的安全性和时效性。同时,显示程序文件与显示配置文件分区存储、灵活调用,通过上位机软件对 Flash 擦除、写入显示配置时,既满足不同 LED 显示屏需要灵活配置,又规避了对显示程序直接操作的风险。在 Xilinx Spartan-6 FPGA 系统设计中编写状态机调用 ICAP 核,通过内部、外部触发信号可以控制 FPGA 重新配置。该设计方法灵活方便,易于操作,可分时复用提高编程资源的利用率和系统稳定性,在工程应用中有很高的应用价值。

参 考 文 献:

- [1] 严飞.LED 显示屏灰度控制关键技术的研究 [D]. 长春: 长春光学精密机械与物理研究所, 2013
YAN F. Research on key technology design of LED display gray-scale control [D]. Changchun: Changchun Institute of Optics, Fine Mechanics and Physics, 2013. (in Chinese)
- [2] 李爽, 郑喜凤, 严飞. 千兆以太网介质转换器在 LED 显示屏传输系统中的应用[J]. 液晶与显示, 2012(01):79-84.

- LI S, ZHENG X F, YAN F. Application of gigabit ethernet media converter in LED display transmission system [J]. *Chinese Journal of Liquid Crystals and Displays*, 2012, 26(1): 79-84.(in Chinese)
- [3] 郝亚茹, 邓招奇, 邓春健. LED 显示控制系统移存频率加速方法研究 [J]. *液晶与显示*, 2016, 30(5): 470-476.
- HAO Y R, DENG Z Q, DENG C J. Research on accelerating method of transfer frequency of LED display control system [J]. *Chinese Journal of Liquid Crystals and Displays*, 2016, 30(5): 470-476.(in Chinese)
- [4] 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.
- LIU Z, DU Y F, XU Z B. Implementation of MultiBoot design based on Xilinx-Spartan6 FPGA [J]. *Electronic Science and Technology*, 2012, 25 (3): 28-31.(in Chinese)
- [5] 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.
- LING P, WU X, SHAN S. FPGA Multi-configuration based on SPI FLASH [J]. *Modern Electronics Technique*, 2013, 36(22): 127-130.(in Chinese)
- [6] 陈岚, 李纪云, 朱人杰. 基于 Kintex7 和 SPI Flash 实现 FPGA 的多重加载 [J]. *电子技术应用*, 2014, 40(7): 24-26.
- CHEN L, LI J Y, ZHU R J. Multiple loads of FPGA based on Kintex7 and SPI flash [J]. *Application of Electronic Technique*, 2014, 40 (7): 24-26.(in Chinese)
- [7] 赛灵思. Spartan-6 现场可编程门阵列配置用户指南 (v2.11) [EB/OL]. (2019-03-22) [2019-08-01]. https://china.xilinx.com/support/documentation/user_guides/ug380.pdf.
- Xilinx. Spartan-6 FPGA Configuration User Guide UG380 (v2.11) [EB/OL]. (2019-03-22) [2019-08-01]. https://china.xilinx.com/support/documentation/user_guides/ug380.pdf. (in Chinese)
- [8] 李嘉琛, 杨光. 基于 FPGA 的 SPI FLASH 数据存储系统设计 [J]. *仪器仪表用户*, 2017, 24(6): 16-19.
- LI J C, YANG G. Design of SPI FLASH data storage system based on FPGA [J]. *Instrumentation Users*, 2017, 24 (6): 16-19.(in Chinese)
- [9] 陈燕文, 韩焱, 徐磊, 等. 基于 FPGA 的 SPI Flash 配置存储器复用的实现 [J]. *测试技术学报*, 2017, 31(6): 37-43.
- CHEN Y W, HAN Y, XU L, et al. Implementation of SPI flash configuration memory multiplexing based on FPGA [J]. *Journal of Test and Measurement Technology*, 2017, 31(6): 37-43.(in Chinese)
- [10] 赛灵思. 命令行工具用户指南 (v14.7) [EB/OL]. (2013-10-02) [2019-08-01]. https://china.xilinx.com/support/documentation/sw_manuals/xilinx14_7/devref.pdf.
- Xilinx. Command Line Tools User Guide UG628 (v14.7) [EB/OL]. (2013-10-02) [2019-08-01]. https://china.xilinx.com/support/documentation/sw_manuals/xilinx14_7/devref.pdf.
- [11] 王向军, 张继龙, 阴雷. 光流法运动估计在 FPGA 上的实现与性能分析 [J]. *光学精密工程*, 2019, 27(1): 211-220.
- WANG X J, ZHANG J L, YIN L. Implementation and performance analysis of optical flow based motion estimation on FPGA [J]. *Optics and Precision Engineering*, 2019, 27(1): 211-220.(in Chinese)
- [12] 杨金宝, 李飞, 郝军伟. 基于视频行场消隐期的大容量 FLASH 存储控制器 [J]. *光学精密工程*, 2015, 23(4): 1153-1160.
- YANG J B, LI F, QIE J W. Large capacity FLASH video storage controller base on blanking period of line and field signals [J]. *Optics and Precision Engineering*, 2015, 23(4): 1153-1160.(in Chinese)

作者简介:



曹 慧(1983—),男,江苏连云港人,硕士,高级工程师,2009 年于长春理工大学获得硕士学位,主要从事 LED 显示与控制方面的研究。E-mail: caoh@ccxida.com



程宏斌(1972—),男,吉林蛟河人,博士,研究员,2009 年于美国匹兹堡大学获得博士学位,主要从事光电显示技术方面的研究。E-mail: chenghb@ccxida.com