



(12)发明专利申请

(10)申请公布号 CN 106406936 A

(43)申请公布日 2017.02.15

(21)申请号 201610772503.X

(22)申请日 2016.08.31

(71)申请人 中国船舶重工集团公司第七一二研究所

地址 430064 湖北省武汉市洪山区狮子山街汽校一村

(72)发明人 梁宝明 赵晓宇 李海燕 江可扬 彭溪

(74)专利代理机构 武汉凌达知识产权事务所 (特殊普通合伙) 42221

代理人 刘念涛 宋国荣

(51)Int. Cl.

G06F 9/445(2006.01)

G06F 9/44(2006.01)

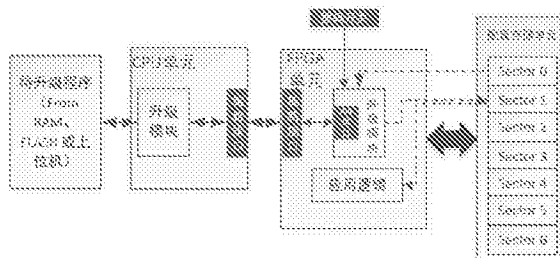
权利要求书2页 说明书4页 附图1页

(54)发明名称

一种FPGA程序多版本管理装置及方法

(57)摘要

本发明公开了一种FPGA程序多版本管理装置,包括依次相连的待升级程序单元、CPU单元、FPGA单元、配置存储单元以及拨码开关;所述的待升级程序单元用于向CPU单元传输升级程序;所述的CPU单元通过数据链路向FPGA单元发送升级指令,由升级模块和链路接口组成;所述的FPGA单元包括链路接口、升级模块和应用逻辑;所述的配置存储单元包括用来存储不同的FPGA程序版本的多个Sector块;所述的拨码开关为外部输入控制接口,主要用于进行配置程序的选择和远程升级功能的使能控制;还公开了其控制方法;本发明通过设置被保护的Sector0单元从而避免了远程升级程序被破坏的危险,通过多版本管理可以存储多个功能配置版本,丰富了模块的功能。



1. 一种FPGA程序多版本管理装置,其特征在于:包括依次相连的待升级程序单元、CPU单元、FPGA单元、配置存储单元以及拨码开关;

所述的待升级程序单元用于向CPU单元传输升级程序;

所述的CPU单元通过数据链路向FPGA单元发送升级指令,由升级模块和链路接口组成;所述的升级模块用于与待升级程序单元通信,收到升级指令后将升级程序以数据的形式传输到相应的FPGA单元;所述的链路接口用于建立CPU单元与FPGA单元的通信链路;

所述的FPGA单元包括链路接口、升级模块和应用逻辑;所述的链路接口功能与CPU单元的链路接口相对应,所述的升级模块用于将CPU单元传来的升级程序写至配置存储单元的对应Sector块中;所述的升级模块上设置有API接口,可以根据CPU发来的数据判断是否是升级指令;所述的应用逻辑为配置程序对FPGA单元进行配置后生成的应用逻辑;

所述的配置存储单元包括用来存储不同的FPGA程序版本的多个Sector块;

所述的拨码开关为外部输入控制接口,主要用于进行配置程序的选择和远程升级功能的使能控制。

2. 根据权利要求1所述的一种FPGA程序多版本管理装置,其特征在于,所述的待升级程序单元为上位机、FLASH芯片或CPU单元可访问的RAM单元。

3. 根据权利要求1所述的一种FPGA程序多版本管理装置,其特征在于,所述的配置存储单元为EPCS或QSPI。

4. 一种FPGA程序多版本管理方法,涉及权利要求1中所述的包括待升级程序单元、CPU单元、FPGA单元、配置存储单元以及拨码开关的管理装置,其特征在于,步骤为:

S01:待升级程序单元与CPU单元约定好传输协议;

S02:要升级FPGA单元的程序时,CPU单元对FPGA单元发出升级指令,FPGA单元收到升级指令后对比升级指令ID地址与自身ID是否匹配,匹配时则开始准备升级工作,当其准备就绪后将给CPU单元一个ready信号并开始升级;

S03:CPU单元向FPGA单元传输升级程序,FPGA单元收到数据以后,若校验通过则写入相应的Sector中,不同的Sector对应不同的程序版本;

S04:当校验出错时FPGA程序会回到初始状态并重新加载Sector 0中的程序,同时CPU单元标记升级错误供上层程序处理;

S05:当配置数据成功写入到配置存储单元以后,升级程序会根据上位控制字节设置内部状态字节,从而完成整个升级过程。

5. 根据权利要求4所述的一种FPGA程序多版本管理方法,其特征在于,所述的FPGA单元在首次烧写时将包含链路接口、升级模块的初始程序烧写至Sector 0中,便于系统上电后首先被加载。

6. 根据权利要求4所述的一种FPGA程序多版本管理方法,其特征在于,所述的步骤S02中程序单元根据读取的拨码开关状态加载相应的程序:当为远程模式时,支持远程升级;当为本地模式时,屏蔽远程升级。

7. 根据权利要求6所述的一种FPGA程序多版本管理方法,其特征在于,当拨码开关最高位为1时,程序单元将接受远程控制指令加载相应Sector中的程序。

8. 根据权利要求6所述的一种FPGA程序多版本管理方法,其特征在于,当拨码开关最高位为0时,程序单元直接将相应Sector中的程序加载到FPGA单元中进行配置。

9. 根据权利要求5所述的一种FPGA程序多版本管理装置,其特征在于,所述的FPGA单元的升级模块首次烧写由USB-Blaster烧写至Sector0,并将Sector0作为不允许升级程序对其进行升级的存储段。

## 一种FPGA程序多版本管理装置及方法

### 技术领域

[0001] 本发明属于电力电子技术领域,尤其涉及一种FPGA程序多版本管理装置,以及其管理方法,主要用于实现FPGA程序多个版本的管理功能。

### 背景技术

[0002] FPGA因其应用灵活、功能能够多次重配等特点,使其在设计开发验证和延长产品寿命等方面都非常有优势。而且随着近年来FPGA价格的不断下降其应用也越来越广泛。

[0003] 很多应用中FPGA都是以协处理器的身份出现的,且一个系统中有时会包含多个FPGA芯片,此时若通过USB-Bluster来逐个烧写非常麻烦,尤其是当电路板安装在设备上或需对大批产品升级时,往往需要繁琐的拆装过程,工作量是非常巨大的。所以针对这种情况,需要更高效的程序更新以及版本管理方法来提高烧写效率。

[0004] 目前虽然有采用NIOS软核进行远程升级的技术方案,但其并没有解决当对同一系统中的多个FPGA芯片进行烧写的问题,并且当程序在升级过程中出现升级失败(如意外断电等)时,就会造成存储芯片中原始程序的损坏,即使再重新上电也无法支持远程烧写功能,只能重新通过USB-Bluster来烧写。

### 发明内容

[0005] 为了克服现有技术的缺点和不足,本发明的目的之一是提供了一种FPGA程序多版本管理装置。

[0006] 本发明解决其技术问题所采用的技术方案是:一种FPGA程序多版本管理装置,包括依次相连的待升级程序单元、CPU单元、FPGA单元、配置存储单元以及拨码开关;

所述的待升级程序单元用于向CPU单元传输升级程序;

所述的CPU单元通过数据链路向FPGA单元发送升级指令,由升级模块和链路接口组成;所述的升级模块用于与待升级程序单元通信,收到升级指令后将升级程序以数据的形式传输到相应的FPGA单元;所述的链路接口用于建立CPU单元与FPGA单元的通信链路;

所述的FPGA单元包括链路接口、升级模块和应用逻辑;所述的链路接口功能与CPU单元的链路接口相对应,所述的升级模块用于将CPU单元传来的升级程序写至配置存储单元的对应Sector块中;所述的升级模块上设置有API接口,可以根据CPU发来的数据判断是否是升级指令;所述的应用逻辑为配置程序对FPGA单元进行配置后生成的应用逻辑;

所述的配置存储单元包括用来存储不同的FPGA程序版本的多个Sector块;

所述的拨码开关为外部输入控制接口,主要用于进行配置程序的选择和远程升级功能的使能控制。

[0007] 所述的一种FPGA程序多版本管理装置,其待升级程序单元为上位机、FLASH芯片或CPU单元可访问的RAM单元。

[0008] 所述的一种FPGA程序多版本管理装置,其配置存储单元为EPCS或QSPI。

[0009] 本发明的目的之二是提供了一种FPGA程序多版本管理装置的管理方法。

[0010] 本发明解决其技术问题所采用的技术方案是：一种FPGA程序多版本管理装置的管理方法，涉及包括待升级程序单元、CPU单元、FPGA单元、配置存储单元以及拨码开关的管理装置，步骤为：

S01：待升级程序单元与CPU单元约定好传输协议；

S02：要升级FPGA单元的程序时，CPU单元对FPGA单元发出升级指令，FPGA单元收到升级指令后对比升级指令ID地址与自身ID是否匹配，匹配时则开始准备升级工作，当其准备就绪后将给CPU单元一个ready信号并开始升级；

S03：CPU单元向FPGA单元传输升级程序，FPGA单元收到数据以后，若校验通过则写入相应的Sector中，不同的Sector对应不同的程序版本；

S04：当校验出错时FPGA程序会回到初始状态并重新加载Sector 0中的程序，同时CPU单元标记升级错误供上层程序处理；

S05：当配置数据成功写入到配置存储单元以后，升级程序会根据上位控制字节设置内部状态字节，从而完成整个升级过程。

[0011] 所述的一种FPGA程序多版本管理方法，其FPGA单元在首次烧写时将包含链路接口、升级模块的初始程序烧写至Sector 0中，便于系统上电后首先被加载。

[0012] 进一步，FPGA单元的升级模块首次烧写由USB-Blaster烧写至Sector0，并将Sector0作为不允许升级程序对其进行升级的存储段。

[0013] 所述的一种FPGA程序多版本管理方法，步骤S02中程序单元根据读取的拨码开关状态加载相应的程序：当为远程模式时，支持远程升级；当为本地模式时，屏蔽远程升级。

[0014] 进一步，当拨码开关最高位为1时，程序单元将接受远程控制指令加载相应Sector中的程序。当拨码开关最高位为0时，程序单元直接将相应Sector中的程序加载到FPGA单元中进行配置。

[0015] 本发明的有益效果是：通过设置被保护的Sector0单元从而避免了远程升级程序被破坏的危险，通过多版本管理可以存储多个功能配置版本，丰富了模块的功能；所以说本发明提供的技术更加安全可靠，且具有实用性，与现有技术相比，本发明能够根据配置选择加载不同版本的程序，增加了系统的灵活性；解决了程序在升级过程中，因意外造成的升级失败而破坏原始程序，导致无法再进行远程升级的问题；增加拨码开关进行升级功能的控制，避免误升级，增强了系统的安全性；此外，本发明在不增加现有成本的基础上实现了FPGA程序的升级和版本管理，方便高效且应用灵活。

## 附图说明

[0016] 图1是本发明装置的结构框图；

图2是本发明管理方法的工作流程示意图。

## 具体实施方式

[0017] 为了更加清楚的阐述本发明方案，下面结合附图和实例对发明内容进行进一步的说明。应当理解，此处所描述的具体是实施例仅用于解释本发明，保护范围并不局限于所述实例。

[0018] 参照图1所示，本发明公开了一种FPGA程序多版本管理装置，包括依次相连的待升

级程序单元、CPU单元、FPGA单元、配置存储单元以及拨码开关。

[0019] 所述的待升级程序单元用于向CPU单元传输升级程序,待升级程序单元为上位机、FLASH芯片或CPU单元可访问的RAM单元,待升级程序单元需要与CPU单元约定好传输协议。

[0020] 所述的CPU单元由升级模块和链路接口两部分组成,通过与FPGA单元之间的数据链路向FPGA单元发送升级指令。

[0021] CPU单元内部的升级模块用于与待升级程序单元通信,其负责监测升级指令,收到升级指令后负责将升级程序以数据的形式传输到相应的FPGA单元,将指定地址中的配置程序传输到FPGA中,并控制传输过程和检测传输结果,直至传输结束。

[0022] 所述的链路接口用于建立CPU单元与FPGA单元的通信链路;包括协议约定、空间分配等,当有特殊需求是也可以建立独立的通信接口如SPI、RS232等,利用CPU与FPGA间原有的通信信道即可,不需另建链路,其可支持以太网、USB、SPI等协议传输。

[0023] 所述的FPGA单元包括链路接口、升级模块和应用逻辑等几个部分,所述的链路接口功能与CPU单元的链路接口相对应,所述的升级模块用于将CPU单元传来的升级程序写至配置存储单元的对应Sector块中,升级模块负责将配置存储单元划分成不同的Sector块,并将不同版本的配置程序依据ID号存储到不同的Sector中,上电时根据配置加载不同块中的程序,FPGA单元根据ID来判断是否升级,ID匹配后,CPU单元向FPGA单元传输升级数据,FPGA单元根据指令将升级数据存储到对应的空间中,不同空间对应不同的程序版本,上电时可根据上位指令或本板拨码开关选择性加载,实现升级和版本管理。

[0024] 所述的升级模块上设置有API接口,API接口是由FPGA程序自身的限制而设计的,主要负责与CPU单元的升级过程向对接,API接口能够根据CPU单元发来的数据判断是否是升级指令,且根据ID判断是否是对本单元的升级,从而支持网络式升级。

[0025] 所述的升级模块还负责对配置存储单元进行Sector块的划分,即将配置存储单元划分成不同的Sector块,比如Sector 0、Sector 1、Sector 2、Sector 3、Sector 4、Sector 5、Sector 6……,配置存储单元将不同版本的配置程序依据ID号存储到不同的Sector中,上电时根据配置加载不同块中的程序,其中升级模块是由Sector 0中的程序配置生成,FPGA单元中的升级模块首次烧写由USB-Blaster烧写至Sector0,并且此存储段不允许升级程序对其进行升级。

[0026] 配置存储单元为EPCS或QSPI,可根据需要自行选定,使用时应根据程序量选择合适的大小的芯片。

[0027] 上电时首先由Sector0配置生成升级模块,然后再根据配置调用其它配置Sector来生成应用逻辑。

[0028] 所述的应用逻辑为配置程序对FPGA单元进行配置后生成的应用逻辑;

所述的拨码开关为外部输入控制接口,主要用于进行配置程序的选择和远程升级功能的使能控制。

[0029] 本发明管理装置采用CPU+FPGA+配置芯片的系统结构方式,支持一主多从的网络升级方式;在本实例中待升级程序单元为运行于PC的升级软件和待升级程序,升级指令由升级软件发起,并负责将待升级程序传输至CPU单元;待升级程序是由QUARTUS II生成的FPGA程序,Sector0中的升级模块程序需首先通过USB-Bluster下载至配置存储单元中。

[0030] 本实例中的CPU单元为运行有嵌入式Linux系统的SoC-FPGA系统,CPU单元在收到

待升级程序后,现将程序缓存至RAM中,然后向FPGA单元发起升级指令。

[0031] 本地模式时,程序将根据拨码开关的值直接将相应Sector中的程序加载到FPGA中进行配置。

[0032] 参照图2所示,本发明公开了一种FPGA程序多版本管理装置的管理方法,步骤为:

S01:待升级程序单元与CPU单元约定好传输协议。

[0033] S02:要升级FPGA单元的程序时,CPU单元对FPGA单元发出升级指令,FPGA单元收到升级指令后对比升级指令ID地址与自身ID是否匹配,匹配时则开始准备升级工作,当其准备就绪后将通过其与FPGA间的数据链路给CPU单元一个ready信号并开始升级。

[0034] 程序单元根据读取的拨码开关状态,对程序升级和加载进行硬件控制:当为远程模式时,系统支持远程升级;当为本地模式时,系统屏蔽远程升级;当拨码开关最高位为1时,程序单元将接受远程控制指令加载相应Sector中的程序;当拨码开关最高位为0时,程序单元直接将相应Sector中的程序加载到FPGA单元中进行配置,这样方便现场操作,也增强了安全性能。

[0035] S03:CPU单元向FPGA单元传输升级程序,CPU单元负责将升级程序以数据的形式传输到FPGA中,利用CPU与FPGA间原有的通信信道即可,不需另建链路,其可支持以太网、USB、SPI等协议传输。

[0036] FPGA单元在首次烧写时必须将包含链路接口、升级模块的初始程序烧写至Sector 0中,用于实现远程升级支持,便于系统上电后首先被加载,系统上电后配置芯片首先加载Sector 0中的程序来配置FPGA芯片,上电时可根据上位指令或本板拨码开关选择性加载,实现升级和版本管理。

[0037] FPGA单元收到数据以后,若校验通过则写入相应的Sector中,不同的Sector对应不同的程序版本。

[0038] S04:当校验出错时FPGA程序会回到初始状态并重新加载Sector 0中的程序,同时CPU单元标记升级错误供上层程序处理。

[0039] S05:当配置数据成功写入到配置存储单元以后,升级程序会根据上位控制字节设置内部状态字节,从而完成整个升级过程。

[0040] 其中FPGA单元的升级模块首次烧写由USB-Blaster烧写至Sector0,并将Sector0作为不允许升级程序对其进行升级的存储段。

[0041] 上电时首先由Sector0配置生成升级模块,然后再根据配置调用其它配置Sector来生成应用逻辑。升级模块还负责对配置存储单元进行Sector块的划分。

[0042] 上述实施例仅例示性说明本发明的原理及其功效,以及部分运用的实施例,对于本领域的普通技术人员来说,在不脱离本发明创造构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。

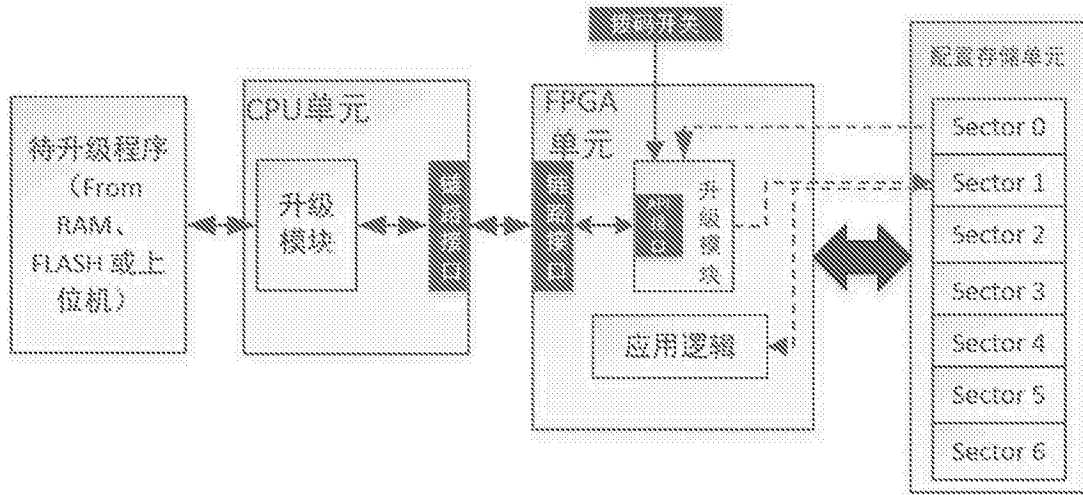


图1

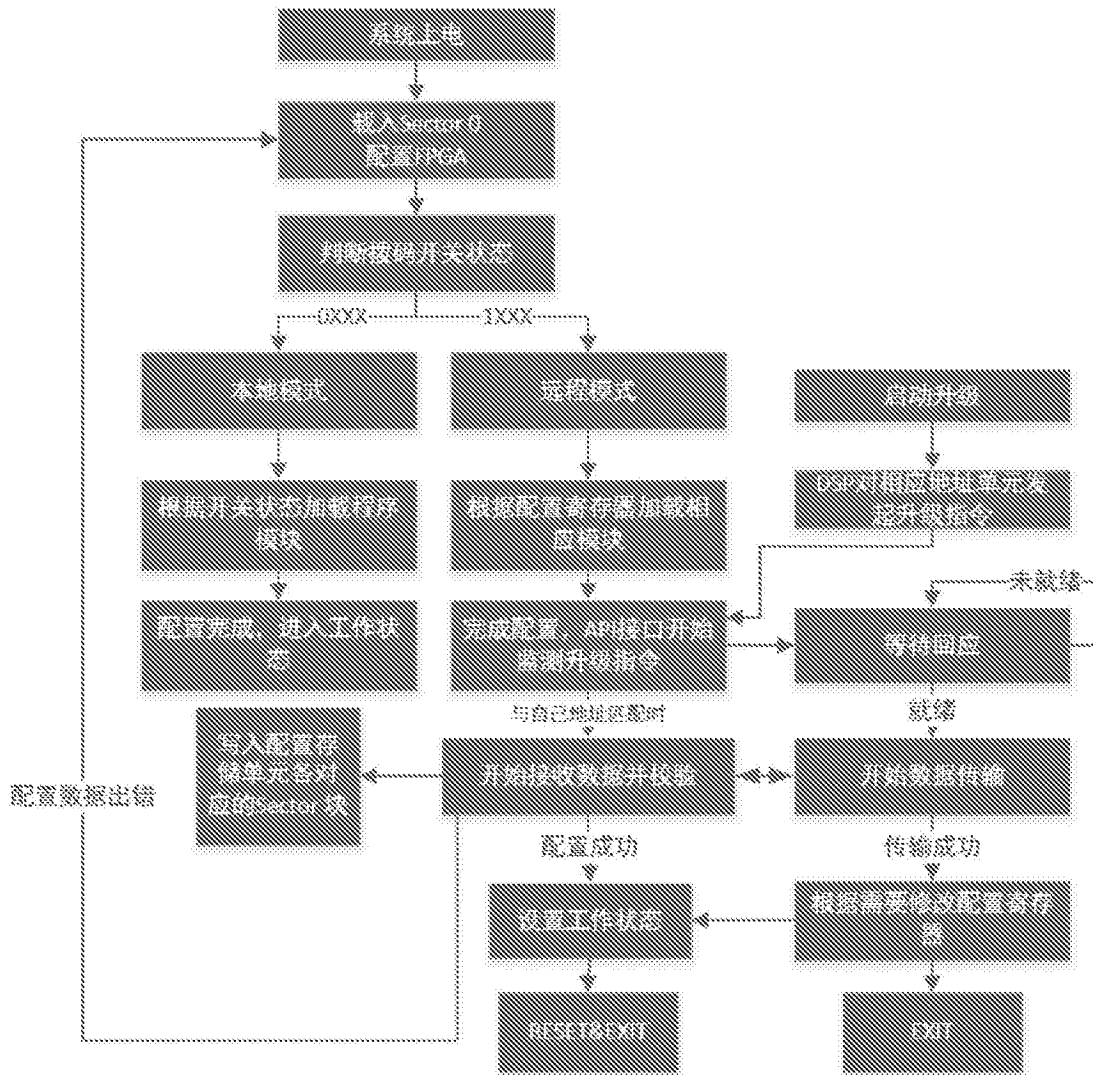


图2