



(12)发明专利申请

(10)申请公布号 CN 105786752 A  
(43)申请公布日 2016.07.20

(21)申请号 201610044469.4

(22)申请日 2016.01.22

(71)申请人 北京大学

地址 100871 北京市海淀区颐和园路5号

(72)发明人 王韬 李志伟 丁博岩 吕松武

(74)专利代理机构 北京万象新悦知识产权代理  
事务所(普通合伙) 11360

代理人 苏爱华

(51)Int.Cl.

G06F 13/42(2006.01)

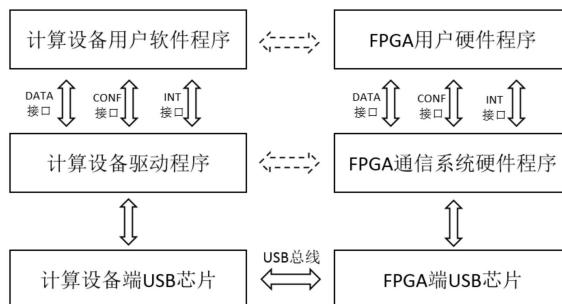
权利要求书4页 说明书11页 附图2页

(54)发明名称

一种计算设备与FPGA间的USB通信方法和通信系统

(57)摘要

本发明公布了一种计算设备与FPGA间的USB通信方法和通信系统,通过USB总线实际连接计算设备端USB芯片和FPGA端USB芯片;计算设备端USB芯片连接计算设备驱动程序,FPGA端USB芯片连接FPGA通信系统硬件程序;计算设备驱动程序通过通信接口连接计算设备用户软件程序,FPGA通信系统硬件程序通过通信接口连接FPGA用户硬件程序;通过USB总线的实际连接实现计算设备驱动程序与FPGA硬件程序的虚拟连接,进一步实现计算设备用户软件程序与FPGA用户硬件程序的虚拟连接;由此实现通过通信接口连接计算设备和FPGA硬件。本发明能够提供计算设备与FPGA间高效、可靠、强交互性的数据通信,通用性高。



1. 一种计算设备与FPGA间的USB通信方法,通过USB总线实际连接计算设备端USB芯片和FPGA端USB芯片;计算设备端USB芯片连接计算设备驱动程序,FPGA端USB芯片连接FPGA通信系统硬件程序;计算设备驱动程序通过通信接口连接计算设备用户软件程序,接FPGA通信系统硬件程序通过通信接口连接FPGA用户硬件程序;通过USB总线的实际连接实现计算设备驱动程序与FPGA硬件程序的虚拟连接,进一步实现计算设备用户软件程序与FPGA用户硬件程序的虚拟连接;其特征是,所述通信接口包括DATA通信接口、CONF通信接口和INT通信接口;由此实现通过通信接口连接计算设备和FPGA硬件;

当所述通信接口为DATA通信接口时,所述DATA通信接口在FPGA端的用户接口包括通信系统状态信号、请求与应答信号和DATA交互信号,所述DATA通信接口支持计算设备与FPGA间双向的、互不影响的、不同优先级的数据传输;所述DATA通信接口的信号交互过程包括步骤a11)~a13):

a11)通信系统状态信号向用户提供DATA通信接口信息;DATA通信接口信息包括DATA通信接口是否准备好、DATA通信接口是否存在错误和DATA通信接口有哪些优先级的通道可用信息;

a12)用户根据通信系统状态信号发出请求信号,当通信系统准备好时,向用户回复应答信号;请求与应答信号为通信系统与用户的握手信号,握手之后开始数据的传输;

a13)通过DATA交互信号进行DATA的数据传输;

当所述通信接口为CONF通信接口时,所述CONF接口在FPGA端的用户接口包括CONF写入的请求与应答信号、CONF写入的交互信号、CONF读出的请求与应答信号和CONF读出的交互信号;CONF通信接口支持计算设备对FPGA中配置寄存器的读写操作,包括计算设备对FPGA中配置寄存器写操作的交互与传输过程和计算设备对FPGA中配置寄存器读操作的交互与传输过程;所述CONF通信接口的信号交互过程包括步骤b11)~b14):

b11)CONF写入的请求与应答信号为通信系统与用户的握手信号,通信系统向用户发出CONF写入请求,当用户完成CONF写入时,向通信系统回复应答信号;此次握手之后,即完成CONF写入操作;

b12)CONF写入的交互信号包括寄存器写地址与写入值,不同的寄存器写地址对应不同的寄存器;通过写入FPGA中的寄存器实现CONF写入操作;

b13)CONF读出的请求与应答信号为通信系统与用户的握手信号,通信系统向用户发出CONF读出请求,当用户完成CONF读出时,向通信系统回复应答信号,此次握手之后,即完成CONF读出操作;

b14)CONF读出的交互信号包括寄存器读地址与读出值,寄存器读地址与寄存器一一对应,通信系统向用户给出寄存器读地址,用户向通信系统反馈该寄存器的值;通过读取FPGA中的寄存器实现CONF写;

当所述通信接口为INT通信接口时,所述INT通信接口在FPGA端的用户接口包括通信系统状态信号和INT请求与清除信号;所述INT通信接口支持FPGA向计算设备发送中断、支持上行传输中断、下行传输中断和用户自定义中断;所述INT通信接口的信号交互过程包括步骤c11)~c12):

c11)通信系统状态信号向用户提供INT接口是否使能信息;

c12)INT请求与清除信号包括INT中断请求信号和中断清除信号;用户根据通信系统状

态信号判断INT接口是否使能,当INT接口使能时,用户向通信系统发送INT中断请求信号,当通信系统中的计算设备收到INT中断请求信号后,通过通信系统向用户回复中断清除信号,表明INT请求已被处理。

2.如权利要求1所述计算设备与FPGA间的USB通信方法,其特征是,步骤a12)所述数据的传输包括下行方向在FPGA中的交互与传输过程、下行方向在计算设备中的交互与传输过程、上行方向在FPGA中的交互与传输过程和上行方向在计算设备中的交互与传输过程;

所述下行方向在FPGA中的交互与传输过程如下:

a21)通信系统实时更新系统当前状态,用户查看通信系统状态,当通信系统已准备好、且不同优先级的通道中存在待读取DATA时,用户向通信系统发出DATA读取请求和欲请求的通道号;

a22)通信系统与用户完成请求与应答握手;

a23)用户读取描述符;获得DATA的长度和对该DATA的描述信息;

a24)用户根据在描述符中获取的信息读取DATA;

a25)用户完成对该DATA处理后,向通信系统写入包含对该DATA的处理情况的状态描述符;

a26)用户通知计算设备对a25)中的状态描述符进行处理;

a27)开始下一次传输,重复执行a21)~a26);

所述上行方向在计算设备中通过设备驱动封装为用户态接口,交互与传输过程如下:

a31)根据通信系统软硬件的相关状态,向FPGA端用户硬件提供可支持的最大DATA长度信息,并在发生变化时进行更新;

a32)收到用户发送的中断后调用相关接口获取DATA,并进行处理;

a33)开始下一次传输,重复执行a31)~a32);

所述上行方向在FPGA中的交互与传输过程如下:

a41)通信系统实时更新系统当前状态,用户查看通信系统状态,当通信系统已准备好、且不同优先级的通道中可以接收DATA时,用户可向通信系统发出DATA写入请求;

a42)通信系统与用户完成请求与应答握手;

a43)用户读取描述符;获得通信系统可支持的最大DATA长度;

a44)用户在可支持的最大DATA长度范围内确定本次写入的DATA长度、通道号和用户自定义信息,写入状态描述符;

a45)用户写入对应长度的DATA;

a46)用户发送中断,通知计算设备对a45)中的DATA进行处理;

a47)开始下一次传输,重复执行a41)~a46);

所述下行方向在计算设备中通过设备驱动封装为用户态接口,交互与传输过程如下:

a51)用户填充欲发送DATA的通道号、长度和用户自定义描述,与欲发送的DATA共同提交给设备驱动,由设备驱动按照约定的格式发送至USB总线;

a52)处理该DATA的状态描述符,读取FPGA端用户硬件对该DATA的处理情况;

a53)开始下一次传输,重复执行a51)~a52)。

3.如权利要求1所述计算设备与FPGA间的USB通信方法,其特征是,在所述CONF通信接口的信号交互过程中,所述计算设备对FPGA中配置寄存器写操作的交互与传输过程如下:

b21) 计算设备调用驱动提供的CONF写函数,并传递寄存器地址和写入值;

b22) 通信库FPGA端在收到CONF写报文后,向用户发出CONF写请求,并同时给出寄存器地址和写入值,用户在完成寄存器写入后使能确认应答信号;

b23) 开始下一次CONF写操作,重复执行b21)~b22);

所述计算设备对FPGA中配置寄存器读操作的交互与传输过程如下:

b31) 计算设备调用驱动提供的CONF读函数,并传递寄存器地址;

b31) 通信库驱动向FPGA发出CONF读报文;

b32) 通信库FPGA端在收到CONF读报文后,向用户发出CONF读请求,并同时给出寄存器地址,用户在完成寄存器读取后使能应答信号并同时给出确认应答信号;

b33) 通信库FPGA端发送CONF读报文的返回报文值通信库驱动,并将读取值作为CONF读函数的返回值提供给用户;

b34) 开始下一次传输,重复执行b31)~b33)。

4. 如权利要求1所述计算设备与FPGA间的USB通信方法,其特征是,在所述INT通信接口的信号交互过程中,所述上行传输中断与下行传输中断用于告知用户软件对传输至计算设备的报文进行处理;交互与传输过程如下:

c21) 在用户软件调用相关函数使能中断的情况下,用户硬件通过对应类型的INT接口发送中断请求;

c22) 通信库FPGA端发送INT报文至通信库驱动;

c23) 通信库驱动在收到INT报文后向FPGA发送中断清除报文,用户软件可通过相应函数阻塞至收到中断;

c24) 通信库FPGA端收到中断清除报文后向用户硬件给出相应类型的中断清除信号;

c25) 开始下一次传输,重复步骤c21)~c24)。

5. 一种使用权利要求1~4任一所述计算设备与FPGA间的USB通信方法实现的计算设备与FPGA间的USB通信系统,所述USB通信系统的硬件设备包括计算机、USB开发板、FMC接口转接板和FPGA开发板;所述计算机上运行通信系统软件程序,所述通信系统软件程序包括计算机驱动程序和计算机库程序和用户软件;所述USB开发板中运行板载芯片的固件程序;所述FPGA开发板中运行硬件程序,包括通信系统硬件程序和用户硬件程序;所述FPGA开发板的硬件部分包括传输控制模块、DATA报文处理模块、DATA引擎、CONF报文处理模块、CONF引擎和INT处理模块;所述FPGA开发板的硬件部分提供由软件控制的复位操作。

6. 如权利要求5所述计算设备与FPGA间的USB通信系统,其特征是,所述FPGA的硬件部分包括传输控制模块、DATA报文处理模块、DATA引擎、CONF报文处理模块、CONF引擎和INT处理模块;

B11. 所述传输控制模块,用于完成与USB芯片的交互,通过解析通信系统报文将USB芯片提供的接口封装为DATA、CONF和INT内部通信接口;

B12. 所述DATA报文处理模块,对于上行报文,完成DATA报文完整性的检查,并传输至传输控制模块;对于下行报文完成数据缓冲,传输至DATA引擎;

B13. 所述DATA引擎,用于向DATA报文处理模块提交上行报文和状态描述符,从DATA报文处理模块获取下行描述符和DATA,缓存数据,提供FPGA端用户接口,完成与用户时钟的时钟域转换;

B14. 所述CONF报文处理模块,对于CONF写入的请求报文,将报文解析为地址和数据,向后传输至CONF引擎;对于CONF读出的请求报文,将报文解析为地址,向后传输至CONF引擎,CONF引擎返回读取值后提交至传输控制模块;

B15. 所述CONF引擎,包括通信库内部CONF模块和用户CONF模块;所述CONF引擎接收来自CONF报文处理模块的读写请求,完成配置信息的读写,对于读请求报文,返回读取值;所述通信库内部CONF模块用于配置和读取通信库相关信息;所述用户CONF模块用于提供FPGA端CONF用户接口,实现与用户时钟的时钟域转换;

B16. 所述INT控制模块,用于将中断报文提交至传输控制模块,提供FPGA端INT用户接口,实现与用户时钟的时钟域转换。

7. 如权利要求5所述计算设备与FPGA间的USB通信系统,其特征是,所述USB开发板使用CYPRESS公司的CYUSB3KIT-003开发板;所述FMC接口转接板使用CYPRESS公司的CYUSB3ACC-005开发板。

## 一种计算设备与FPGA间的USB通信方法和通信系统

### 技术领域

[0001] 本发明属于计算机通信领域,涉及FPGA(现场可编程门阵列),尤其涉及一种高效、可靠、强交互性的计算设备与FPGA间的USB通信方法和通信系统。

### 背景技术

[0002] FPGA(现场可编程门阵列,Field-Programmable Gate Array)是一种可编程的高性能半定制电路器件,随着芯片技术的发展,FPGA性能不断提高,功能逐渐完善,也因此被越来越多的开发者和研究人员作为计算机系统中的加速器使用,现有的FPGA虽然为用户提供了PCIe、万兆以太网等高速接口,但也同时限制了计算机的机箱尺寸,使用者不得不携带大尺寸机箱来完成计算机与FPGA间的通信。计算机与FPGA间高速、便携、易用的通信成为一种需求。

[0003] USB(通用串行总线,Universal Serial Bus)是由USBIF(Universal Serial Bus Implement Forum)制定的一套外部总线标准,如今具备USB 3.0规范接口的设备已经大规模商用,可提供高达5Gbps的数据传输速率。USB协议具有良好的兼容性,不但对前期协议兼容,对不同主板也具有优秀的兼容性。

[0004] 文献1(Usui,Takuma,Ryohei Kobayashi,and Kenji Kise."A Challenge of Portable and High-Speed FPGA Accelerator."Applied Reconfigurable Computing.Springer International Publishing,2015.383-392)提供一款基于CYUSB3014芯片设计的计算机与FPGA间的USB 3.0通信系统。该系统使用USB连接FPGA与计算机,该系统的传输速率尚可,但只为用户提供了流式数据接口,无法传输帧数据,也无法满足更加多样化的接口需求,如寄存器读写和中断等。文献2(Jolfaei,Fatemeh Arbab,et al."High speed USB 2.0interface for FPGA based embedded systems."Embedded and Multimedia Computing,2009.EM-Com 2009.4th International Conference on.IEEE,2009.)提供一款基于CY7C68001芯片设计的计算机与FPGA间的USB 2.0通信系统,由于其仅支持USB 2.0协议,大大限制了该通信系统的传输速率,无法满足如今高速数据传输的需求。文献3(Zaboloŧny,Wojciech M.,et.al."Low cost USB-local bus interface for FPGA based systems."Photonics Applications in Astronomy,Communications, Industry,and High-Energy Physics Experiments 2012.International Society for Optics and Photonics,2012.)提供一款基于FT2232H芯片开发的计算机与FPGA间通信系统,支持帧数据传输与寄存器读写,为用户提供了数据、I2C与JTAG接口,可方便用户进行数据读写与调试。该通信系统虽然为用户提供了多样化的接口,但传输速率很低,且系统工作不稳定。

### 发明内容

[0005] 为了克服上述现有技术的不足,本发明提供一种基于USB总线的通信方法和系统,能够提供计算设备与FPGA间高效、可靠、强交互性的数据通信,本发明的通用性高,所述计

算设备包括但不限于支持USB接口的、能够运行操作系统的计算设备,如台式机、笔记本电脑、手机等,通过接口转换器将USB接口转换为其他接口连接的不支持USB接口的计算设备也同样适用于本发明。

[0006] 为了便于说明,本文约定:“DATA”表示帧数据;“CONF”表示配置信息;“INT”表示中断;“FIFO”表示FPGA中使用随机存取存储器实现的先入先出硬件电路模块;“上行”表示FPGA向计算设备方向;“下行”表示计算设备向FPGA方向。

[0007] 本发明的原理是通过USB总线连接计算设备和FPGA开发板,分别在计算设备与FPGA中通过编写程序进行双向的数据通信,并分别在计算设备与FPGA端提供方便用户使用的多样化接口。

[0008] 本发明提供的技术方案是:

[0009] 一种计算设备与FPGA间的USB通信方法,通过USB总线实际连接计算设备端USB芯片和FPGA端USB芯片;计算设备端USB芯片连接计算设备驱动程序,FPGA端USB芯片连接FPGA通信系统硬件程序;计算设备驱动程序通过通信接口连接计算设备用户软件程序,FPGA通信系统硬件程序通过通信接口连接FPGA用户硬件程序;通过USB总线的实际连接实现计算设备驱动程序与FPGA硬件程序的虚拟连接,进一步实现计算设备用户软件程序与FPGA用户硬件程序的虚拟连接;所述通信接口包括DATA通信接口、CONF通信接口和INT通信接口;由此实现通过通信接口连接计算设备和FPGA硬件。

[0010] 针对上述计算设备与FPGA间的USB通信方法,进一步地,所述通信接口为DATA通信接口,所述DATA通信接口在FPGA端的用户接口包括通信系统状态信号、请求与应答信号和DATA交互信号,所述DATA通信接口支持计算设备与FPGA间双向的、互不影响的、不同优先级的数据传输;所述DATA通信接口的信号交互过程如下:

[0011] a11)通信系统状态信号向用户提供DATA通信接口信息;DATA通信接口信息包括DATA通信接口是否准备好、DATA通信接口是否存在错误和DATA通信接口有哪些优先级的通道可用等信息;

[0012] a12)用户根据通信系统状态信号发出请求信号,当通信系统准备好时,向用户回复应答信号;请求与应答信号为通信系统与用户的握手信号,握手之后开始数据的传输;

[0013] a13)通过DATA交互信号进行DATA的数据传输。

[0014] 针对上述计算设备与FPGA间的USB通信方法,进一步地,步骤a12)所述数据的传输包括下行方向在FPGA中的交互与传输过程、下行方向在计算设备中的交互与传输过程、上行方向在FPGA中的交互与传输过程和上行方向在计算设备中的交互与传输过程;

[0015] 所述下行方向在FPGA中的交互与传输过程如下:

[0016] a21)通信系统实时更新系统当前状态,用户查看通信系统状态,当通信系统已准备好、且不同优先级的通道中存在待读取DATA时,用户向通信系统发出DATA读取请求和欲请求的通道号;

[0017] a22)通信系统与用户完成请求与应答握手;

[0018] a23)用户读取描述符;获得DATA的长度和对该DATA的描述信息;

[0019] a24)用户根据在描述符中获取的信息读取DATA;

[0020] a25)用户完成对该DATA处理后,向通信系统写入包含对该DATA的处理情况的状态描述符;

- [0021] a26)用户发送中断,通知计算设备对a25)中的状态描述符进行处理;
- [0022] a27)开始下一次传输,重复执行a21)~a26);
- [0023] 所述下行方向在计算设备中通过设备驱动封装为用户态接口,交互与传输过程如下:
- [0024] a31)用户填充欲发送DATA的通道号、长度和用户自定义描述符,与欲发送的DATA共同提交给设备驱动,由设备驱动按照约定的格式发送至USB总线;
- [0025] a32)处理该DATA的状态描述符,读取FPGA端用户硬件对该DATA的处理情况;
- [0026] a33)开始下一次传输,重复执行a31)~a32)。
- [0027] 所述上行方向在FPGA中的交互与传输过程如下:
- [0028] a41)通信系统实时更新系统当前状态,用户查看通信系统状态,当通信系统已准备好、且不同优先级的通道中可以接收DATA时,用户可向通信系统发出DATA写入请求;
- [0029] a42)通信系统与用户完成请求与应答握手;
- [0030] a43)用户读取描述符;获得通信系统可支持的最大DATA长度;
- [0031] a44)用户在可支持的最大DATA长度范围内确定本次写入的DATA长度、通道号 and 用户自定义信息,写入状态描述符;
- [0032] a45)用户写入对应长度的DATA;
- [0033] a46)用户发送中断,通知计算设备对a45)中的DATA进行处理;
- [0034] a47)开始下一次传输,重复执行a41)~a46);
- [0035] 所述上行方向在计算设备中通过设备驱动封装为用户态接口,交互与传输过程如下:
- [0036] a51)根据通信系统软硬件的相关状态,向FPGA端用户硬件提供可支持的最大DATA长度信息,并在发生变化时进行更新;
- [0037] a52)收到用户发送的中断后调用相关接口获取DATA,并进行处理;
- [0038] a53)开始下一次传输,重复执行a51)~a52);
- [0039] 针对上述计算设备与FPGA间的USB通信方法,进一步地,所述通信接口为CONF通信接口,所述CONF通信接口在FPGA端的用户接口包括CONF写入的请求与应答信号、CONF写入的交互信号、CONF读出的请求与应答信号和CONF读出的交互信号;CONF通信接口支持计算设备对FPGA中配置寄存器的读写操作,包括计算设备对FPGA中配置寄存器写操作的交互与传输过程和计算设备对FPGA中配置寄存器读操作的交互与传输过程;所述CONF通信接口的信号交互过程如下:
- [0040] b11)CONF写入的请求与应答信号为通信系统与用户的握手信号,通信系统向用户发出CONF写入请求,当用户完成CONF写入时,向通信系统回复应答信号,此次握手之后,即完成CONF写入操作;
- [0041] b12)CONF写入的交互信号包括寄存器写地址与写入值,不同的寄存器写地址对应不同的寄存器;通过写入FPGA中的寄存器实现CONF写入操作;
- [0042] b13)CONF读出的请求与应答信号为通信系统与用户的握手信号,通信系统向用户发出CONF读出请求,当用户完成CONF读出时,向通信系统回复应答信号,此次握手之后,即完成CONF读出操作;
- [0043] b14)CONF读出的交互信号包括寄存器读地址与读出值,寄存器读地址与寄存器一



一对应,通信系统向用户给出寄存器读地址,用户向通信系统反馈该寄存器的值;通过读取FPGA中的寄存器实现CONF读出操作。

[0044] 针对上述计算设备与FPGA间的USB通信方法,进一步地,所述计算设备对FPGA中配置寄存器写操作的交互与传输过程如下:

[0045] b21)计算设备调用驱动提供的CONF写函数,并传递寄存器地址和写入值;

[0046] b22)通信系统FPGA端在收到CONF写报文后,向用户发出CONF写请求,并同时给出寄存器地址和写入值,用户在完成寄存器写入后使能确认应答信号;

[0047] b23)开始下一次CONF写操作,重复执行b21)~b22);

[0048] 所述计算设备对FPGA中配置寄存器读操作的交互与传输过程如下:

[0049] b31)计算设备调用驱动提供的CONF读函数,并传递寄存器地址;

[0050] b32)通信系统驱动程序向FPGA发出CONF读报文;

[0051] b33)通信系统FPGA端在收到CONF读报文后,向用户发出CONF读请求,并同时给出寄存器地址,用户在完成寄存器读取后使能应答信号并同时给出确认应答信号;

[0052] b34)通信系统FPGA端发送CONF读报文的返回报文至通信系统驱动程序,并将读取值作为CONF读函数的返回值提供给用户;

[0053] b35)开始下一次传输,重复执行b31)~b34)。

[0054] 针对上述计算设备与FPGA间的USB通信方法,进一步地,所述通信接口为INT通信接口,所述INT通信接口在FPGA端的用户接口包括通信系统状态信号和INT请求与清除信号;所述INT通信接口支持FPGA向计算设备发送中断、支持上行传输中断、下行传输中断和用户自定义中断;所述INT通信接口的信号交互过程如下:

[0055] c11)通信系统状态信号向用户提供INT接口是否使能信息;

[0056] c12)INT请求与清除信号包括INT中断请求信号和中断清除信号;用户根据通信系统状态信号判断INT接口是否使能,当INT接口使能时,用户向通信系统发送INT中断请求信号,当通信系统中的计算设备收到INT中断请求信号后,通过通信系统向用户回复中断清除信号,表明INT请求已被处理。

[0057] 针对上述计算设备与FPGA间的USB通信方法,进一步地,所述上行传输中断与下行传输中断用于告知用户软件对传输至计算设备的报文进行处理;交互与传输过程如下:

[0058] c21)在用户软件调用相关函数使能中断的情况下,用户硬件通过对应类型的INT接口发送中断请求;

[0059] c22)通信系统FPGA端发送INT报文至通信系统驱动程序;

[0060] c23)通信系统驱动程序在收到INT报文后向FPGA发送中断清除报文,用户软件可通过相应函数阻塞至收到中断;

[0061] c24)通信系统FPGA端收到中断清除报文后向用户硬件给出相应类型的中断清除信号;

[0062] c25)开始下一次传输,重复步骤c21)~c24)。

[0063] 针对上述计算设备与FPGA间的USB通信方法,进一步地,所述FPGA的硬件部分包括传输控制模块、DATA报文处理模块、DATA引擎、CONF报文处理模块、CONF引擎和INT处理模块;

[0064] B11.所述传输控制模块,用于完成与USB芯片的交互,通过解析通信系统报文将

USB芯片提供的接口封装为DATA、CONF和INT内部通信接口；

[0065] B12.所述DATA报文处理模块,对于上行报文,完成DATA报文完整性的检查,并传输至传输控制模块;对于下行报文完成数据缓冲,传输至DATA引擎;

[0066] B13.所述DATA引擎,用于向DATA报文处理模块提交上行报文和状态描述符,从DATA报文处理模块获取下行描述符和DATA,缓存数据,提供FPGA端用户接口,完成与用户时钟的时钟域转换;

[0067] B14.所述CONF报文处理模块,对于CONF写入的请求报文,将报文解析为地址和数据,向后传输至CONF引擎;对于CONF读出的请求报文,将报文解析为地址,向后传输至CONF引擎,CONF引擎返回读取值后提交至传输控制模块;

[0068] B15.所述CONF引擎,包括通信系统内部CONF模块和用户CONF模块;所述CONF引擎接收来自CONF报文处理模块的读写请求,完成配置信息的读写,对于读请求报文,返回读取值;所述通信系统内部CONF模块用于配置和读取通信系统相关信息;所述用户CONF模块用于提供FPGA端CONF用户接口,实现与用户时钟的时钟域转换;

[0069] B16.所述INT控制模块,用于将中断报文提交至传输控制模块,提供FPGA端INT用户接口,实现与用户时钟的时钟域转换。

[0070] 基于上述计算设备与FPGA间的USB通信方法,本发明还提供一种计算设备与FPGA间的USB通信系统,所述USB通信系统的硬件设备包括计算机、USB开发板、FMC接口转接板和FPGA开发板;所述计算机上运行通信系统软件程序,所述通信系统软件程序包括计算机驱动程序和计算机库程序和用户软件;所述USB开发板中运行板载芯片的固件程序;所述FPGA开发板中运行硬件程序,包括通信系统硬件程序和用户硬件程序;所述FPGA开发板的硬件部分包括传输控制模块、DATA报文处理模块、DATA引擎、CONF报文处理模块、CONF引擎和INT处理模块;所述FPGA开发板的硬件部分提供由软件控制的复位操作。

[0071] 针对上述计算设备与FPGA间的USB通信系统,其中,所述USB开发板使用CYPRESS公司的CYUSB3KIT-003开发板;所述FMC接口转接板使用CYPRESS公司的CYUSB3ACC-005开发板。

[0072] 与现有技术相比,本发明的有益效果是:

[0073] 本发明提供一套高效、可靠的通信方法,同时支持三种功能和接口,分别为DATA、CONF和INT接口。DATA功能适用于大量数据传输,CONF功能适用于计算设备对FPGA中的寄存器进行配置和读取,INT功能适用于FPGA向计算机发起中断,以通知计算机对一些事务进行处理。本发明提供的多种功能,能够满足使用者不同的需求,并可以根据自身需求进行不同功能间的协同工作,体现了本发明的通用性、灵活性;而且,本发明方法的通信交互过程使得通信系统能够及时向计算设备反馈FPGA中的数据处理情况,体现了本发明的可靠性、强交互性。利用本发明提供的技术方案,可以完成一套通用、高效、可靠、强交互性的计算设备与FPGA间的通信系统,为FPGA开发者提供了便利,节约了人力成本和时间成本,降低了开发难度,提高了开发效率。

## 附图说明

[0074] 图1是本发明的系统结构图。

[0075] 图2是本发明FPGA硬件部分的结构框图。

[0076] 图3是本发明实施例中详细设计的FPGA硬件部分的结构框图。

### 具体实施方式

[0077] 下面结合附图,通过实施例进一步描述本发明,但不以任何方式限制本发明的范围。

[0078] 本发明提供一种计算设备与FPGA间的USB通信方法,如图1所示,该方法的实现由计算设备驱动程序和FPGA硬件程序构成;计算设备端USB芯片和FPGA端USB芯片通过USB总线实际连接;计算设备端USB芯片连接计算设备驱动程序,FPGA端USB芯片连接FPGA通信系统硬件程序;计算设备驱动程序通过通信接口连接计算设备用户软件程序,FPGA通信系统硬件程序通过通信接口连接FPGA用户硬件程序;通过USB总线的实际连接,实现了计算设备驱动程序与FPGA硬件程序的虚拟连接,并进一步实现了计算设备用户软件程序与FPGA用户硬件程序的虚拟连接。

[0079] 如图1所示,本发明提供的通信方法支持三种接口的通信方法,分别为DATA、CONF与INT。DATA通信接口提供的通信方法功能适用于大量数据传输,CONF通信接口提供的通信方法功能适用于计算设备对FPGA中的寄存器进行配置和读取,INT通信接口提供的通信方法功能适用于FPGA向计算机发起中断,以通知计算机对一些事务进行处理。上述功能能够满足使用者不同的需求,并可以根据自身需求进行不同功能间的协同工作,体现了本发明的通用性、灵活性;其中的交互过程使得通信系统能够及时向计算设备反馈FPGA中的数据处理情况,体现了本发明的可靠性、强交互性。

[0080] 针对上述DATA、CONF与INT三种接口的通信方法和功能,其中,DATA接口在FPGA端的用户接口包括通信系统状态信号、请求与应答信号、描述符交互信号、DATA交互信号、状态描述符交互信号,所述DATA接口的交互信号如下:

[0081] a11)通信系统状态信号;该部分信号向用户提供通信系统DATA接口是否准备好、是否存在错误、有哪些优先级的通道可用等信息。

[0082] a12)请求与应答信号;该部分信号为通信系统与用户的握手信号,用户根据通信系统状态信号发出请求,通信系统若已准备好,向用户回复应答信号。此次握手之后,即可开始DATA传输。

[0083] a13)描述符交互信号;描述符中包含计算设备提供的DATA相关信息,如DATA的优先级、长度和其他用户自定义信息。

[0084] a14)DATA交互信号;通过该交互信号可进行DATA传输。

[0085] a15)状态描述符交互信号;状态描述符中包含FPGA向计算设备提供的DATA相关信息,如DATA长度及其他用户自定义信息。

[0086] 上述DATA功能和接口支持计算设备与FPGA间双向的、互不影响的、不同优先级的数据传输;其中,下行方向在FPGA中的交互与传输过程如下:

[0087] a21)通信系统实时更新系统当前状态,用户查看通信系统状态,当通信系统已准备好、且不同优先级的通道中存在待读取DATA时,用户可向通信系统发出DATA读取请求和欲请求的通道号;

[0088] a22)通信系统与用户完成请求与应答握手;

[0089] a23)用户读取描述符;获得DATA的长度和对该DATA的描述信息;

- [0090] a24)用户根据在描述符中获取的信息读取DATA;
- [0091] a25)用户完成对该DATA处理后,向通信系统写入状态描述符,状态描述符中包含对该DATA的处理情况;用户硬件程序与用户软件程序约定好的情况下,该步骤在通信方法的具体实现中可省略,若省略状态描述符的传输,用户软件将无法通过此途径获知DATA的处理情况,但仍可通过CONF通信接口完成用户硬件程序向用户软件程序的信息反馈;
- [0092] a26)用户通过INT接口发送中断,通知计算设备对a25)中的状态描述符进行处理;用户硬件程序与用户软件程序约定好的情况下,该步骤在通信方法的具体实现中可省略,若省略中断的传输,用户软件将无法及时处理a25)中的状态描述符,且可能发生用户软件读取状态描述符错误的情况;
- [0093] a27)开始下一次传输。
- [0094] 下行方向在计算设备中通过设备驱动封装为用户态接口,交互与传输过程如下:
- [0095] a31)用户填充欲发送DATA的通道号、长度和用户自定义描述,与欲发送的DATA共同提交给设备驱动,由设备驱动按照约定的格式发送至USB总线;
- [0096] a32)等待接收下行方向中断,收到中断后处理a33)中DATA的状态描述符;用户硬件程序与用户软件程序约定好的情况下,该步骤在通信方法的具体实现中可省略,若省略中断的传输,用户软件将无法及时处理a33)中的状态描述符,且可能发生用户软件读取状态描述符错误的情况;
- [0097] a33)调用相关接口获取该DATA的状态描述符并进行解析,获得FPGA端用户硬件对该DATA的处理情况;用户硬件程序与用户软件程序约定好的情况下,该步骤在通信方法的具体实现中可省略,若省略状态描述符的传输,用户软件将无法通过此途径获知DATA的处理情况,但仍可通过CONF通信接口完成用户硬件程序向用户软件程序的信息反馈;
- [0098] a34)开始下一次传输。
- [0099] 上行方向在FPGA中的交互与传输过程如下:
- [0100] a41)通信系统实时更新系统当前状态,用户查看通信系统状态,当通信系统已准备好、且不同优先级的通道中可以接收DATA时,用户可向通信系统发出DATA写入请求;
- [0101] a42)通信系统与用户完成请求与应答握手;
- [0102] a43)用户读取描述符,获得通信系统可支持的最大DATA长度;在用户硬件程序与用户软件程序约定好固定不变的可支持的最大DATA长度的情况下,该步骤在通信方法的具体实现中可省略;
- [0103] a44)用户在可支持的最大DATA长度范围内确定本次写入的DATA长度、通道号 and 用户自定义信息,写入状态描述符;
- [0104] a45)用户写入对应长度的DATA;
- [0105] a46)用户通过INT接口发送中断,通知计算设备对a45)中的DATA进行处理;用户硬件程序与用户软件程序约定好的情况下,该步骤在通信方法的具体实现中可省略,若省略中断的传输,用户软件将无法及时处理a46)中的DATA,且可能发生用户软件读取DATA错误的情况;
- [0106] a47)开始下一次传输。
- [0107] 上行方向在计算设备中通过设备驱动封装为用户态接口,交互与传输过程如下:
- [0108] a51)根据通信系统软硬件的相关状态,向FPGA端用户硬件提供可支持的最大DATA

长度信息,并在发生变化时进行更新;在用户硬件程序与用户软件程序约定好固定不变的可支持的最大DATA长度的情况下,该步骤在通信方法的具体实现中可省略;

[0109] a52)等待接收上行方向中断,收到中断后处理a53)中的DATA;用户硬件程序与用户软件程序约定好的情况下,该步骤在通信方法的具体实现中可省略,若省略中断的传输,用户软件将无法及时处理a46)中的DATA,且可能发生用户软件读取DATA错误的情况;

[0110] a52)调用相关接口获取DATA,并进行处理;

[0111] a53)开始下一次传输。

[0112] CONF接口在FPGA端的用户接口包括CONF写入的请求与应答信号、CONF写入的交互信号、CONF读出的请求与应答信号和CONF读出的交互信号,其具体通信过程如下:

[0113] b11)CONF写入的请求与应答信号;该部分信号为通信系统与用户的握手信号,通信系统向用户发出CONF写入请求,用户完成CONF写入时,向通信系统回复应答信号。此次握手之后,即完成CONF写入操作。

[0114] b12)CONF写入的交互信号;CONF写通过写入FPGA中的寄存器实现。该交互信号包括寄存器写地址与写入值。不同的寄存器写地址需对应不同的寄存器。

[0115] b13)CONF读出的请求与应答信号;该部分信号为通信系统与用户的握手信号,通信系统向用户发出CONF读出请求,用户完成CONF读出时,向通信系统回复应答信号。此次握手之后,即完成CONF读出操作。

[0116] b14)CONF读出的交互信号。CONF写通过读取FPGA中的寄存器实现。该交互信号包括寄存器读地址与读出值。通信系统向用户给出寄存器读地址,用户向通信系统反馈该寄存器的值。不同的寄存器读地址需对应不同的寄存器,相同的寄存器写地址与寄存器读地址对应相同的寄存器。

[0117] 上述CONF功能和接口支持计算设备对FPGA中配置寄存器的读写操作。计算设备对FPGA中配置寄存器写操作的交互与传输过程如下:

[0118] b21)计算设备调用驱动提供的CONF写函数,并传递寄存器地址和写入值;

[0119] b22)通信系统FPGA端在收到CONF写报文后,向用户发出CONF写请求,并同时给出寄存器地址和写入值,用户在完成寄存器写入后使能确认应答信号;

[0120] b23)开始下一次CONF写操作。

[0121] 计算设备对FPGA中配置寄存器读操作的交互与传输过程如下:

[0122] b31)计算设备调用驱动提供的CONF读函数,并传递寄存器地址;

[0123] b31)通信系统驱动程序向FPGA发出CONF读报文;

[0124] b32)通信系统FPGA端在收到CONF读报文后,向用户发出CONF读请求,并同时给出寄存器地址,用户在完成寄存器读取后使能应答信号并同时给出确认应答信号;

[0125] b33)通信系统FPGA端发送CONF读报文的返回报文至通信系统驱动程序,并将读取值作为CONF读函数的返回值提供给用户;

[0126] b34)开始下一次传输。

[0127] INT接口在FPGA端的用户接口包括通信系统状态信号和INT请求与清除信号,其具体通信过程如下:

[0128] c11)通信系统状态信号;该部分信号向用户提供INT接口是否使能等信息。

[0129] c12)INT请求与清除信号;用户根据通信系统状态信号判断INT接口是否使能,当

INT接口使能时,用户可向通信系统发送INT中断请求,当通信系统中的计算设备收到INT请求后,会通过通信系统向用户回复中断清除信号,表明该次INT请求已被处理。

[0130] 上述INT功能和接口支持FPGA向计算设备发送中断,支持上行传输中断,下行传输中断和用户自定义中断,其中上行传输中断与下行传输中断用于告知用户软件对传输至计算设备的报文进行处理。交互与传输过程如下:

[0131] c21)在用户软件调用相关函数使能中断的情况下,用户硬件通过对应类型的INT接口发送中断请求;

[0132] c22)通信系统FPGA端发送INT报文至通信系统驱动程序;

[0133] c23)通信系统驱动程序在收到INT报文后向FPGA发送中断清除报文,用户软件可通过相应函数阻塞至收到中断;

[0134] c24)通信系统FPGA端收到中断清除报文后向用户硬件给出相应类型的中断清除信号;

[0135] c25)开始下一次传输。

[0136] 为了方便快捷地实现上述通信过程,本发明设计了FPGA端硬件结构,如图2所示,FPGA硬件部分包括传输控制模块、DATA报文处理模块、DATA引擎、CONF报文处理模块、CONF引擎和INT处理模块。本发明提供的FPGA端硬件结构体现了分层与不同功能不同分区的设计思想,对报文进行了高速高效处理,体现了本发明的高效性;各模块具体如下:

[0137] B11.所述传输控制模块,完成与USB芯片的交互,通过解析通信系统报文将USB芯片提供的接口封装为DATA、CONF和INT三种内部接口;

[0138] B12.所述DATA报文处理模块,对于上行报文,完成DATA报文完整性的检查,并传输至传输控制模块,对于下行报文完成数据缓冲,传输至DATA引擎。上述对报文完整性的检查,避免了因用户错误传输导致通信系统卡死的问题,使得通信系统能够在出现错误后复位,体现了本发明的可靠性。

[0139] B13.所述DATA引擎,向DATA报文处理模块提交上行报文和状态描述符,从DATA报文处理模块获取下行描述符和DATA,缓存数据,提供FPGA端用户接口,并完成与用户时钟的时钟域转换;

[0140] B14.所述CONF报文处理模块,对于CONF写入的请求报文,将报文解析为地址和数据,向后传输至CONF引擎;对于CONF读出的请求报文,将报文解析为地址,向后传输至CONF引擎,CONF引擎返回读取值后提交至传输控制模块;

[0141] B15.所述CONF引擎,如图3所示,包括通信系统内部CONF模块与用户CONF模块,CONF引擎接收来自CONF报文处理模块的读写请求,完成配置信息的读写,对于读请求报文,返回读取值。其中通信系统内部CONF模块用于配置和读取通信系统相关信息,用户CONF模块提供FPGA端CONF用户接口,并完成了与用户时钟的时钟域转换;

[0142] B16.所述INT控制模块,将中断报文提交至传输控制模块,提供FPGA端INT用户接口,并完成了与用户时钟的时钟域转换。

[0143] 基于上述通信方法,本发明还实现了一套基于Xilinx FPGA的计算机与FPGA间的USB3.0通信系统,用于验证了上述通信方法的可行性和有效性。

[0144] 本发明实施例中实现的计算机与FPGA间的USB 3.0通信系统,在硬件设备上由计算机、USB开发板、FMC接口转接板和FPGA开发板构成,其中,FMC接口是Xilinx FPGA提供的

一种扩展接口, FMC接口转接板可将USB开发板提供的接口转换为FMC接口, 完成USB开发板与FPGA的连接。如图1所示, 计算机上运行通信系统软件程序, 包括计算机驱动程序、计算机库程序和用户软件, USB开发板中运行板载芯片的固件程序, FPGA中运行硬件程序, 包括通信系统硬件程序和用户硬件程序。其中, USB开发板使用CYPRESS公司的CYUSB3KIT-003开发板(CYUSB3KIT-003EZ-USB FX3SuperSpeed Explorer Kit.<http://www.cypress.com/?rID=99916>), FMC接口转接板使用CYPRESS公司的CYUSB3ACC-005开发板(CYUSB3ACC-005FMC Interconnect Board for the EZ-USB FX3SuperSpeed Explorer Kit.<http://www.cypress.com/?rID=99921>)。如果使用Altera FPGA, 可以对应使用HSMC接口转接板, 即CYPRESS公司的CYUSB3ACC-006开发板(CYUSB3ACC-006HSMC Interconnect Board for the EZ-USB® FX3SuperSpeed Explorer Kit.<http://www.cypress.com/?rID=99922>)。

[0145] 本实施例的FPGA开发板的硬件部分, 基于发明内容中的通信方法进行设计, 如图2所示, 包括传输控制模块、DATA报文处理模块、DATA引擎、CONF报文处理模块、CONF引擎和INT处理模块; 具体结构如下:

[0146] B21. 所述传输控制模块, 如图3所示, 包括发送引擎模块、接收引擎模块和传输引擎模块。发送引擎用来控制上行方向(FPGA向计算机方向)的数据传输, 将前级模块提供的INT报文与CONF报文两路合一, 并与DATA报文一起提交给传输引擎模块; 接收引擎模块用来控制下行方向(计算机向FPGA方向)的数据传输, 将传输引擎提供的数据解析为DATA报文、CONF读报文与CONF写报文并向DATA报文处理模块或CONF报文处理模块传输; 传输引擎模块完成了与USB开发板间的交互并将USB开发板提供的半双工单接口封装为两个全双工接口, 该模块包含一个不断循环检查的状态机, 轮询发送引擎与USB开发板是否有发送或接收请求, 如果发送引擎有发送请求, 则将数据发送至USB开发板, 如果USB开发板中有数据待接收, 则将数据接收至接收引擎模块。

[0147] B22. 所述DATA报文处理模块, 如图3所示, 对于上行报文, 完成DATA报文完整性的检查, 在确定报文完整后传输至发送引擎模块, 以防止报文不完整导致长时间占用与USB开发板的交互接口; 对于下行报文, 仅完成数据缓冲, 传输至DATA引擎。

[0148] B23. 所述DATA引擎, 如图3所示, 包括DATA上行引擎与DATA下行引擎。DATA下行引擎模块提供了下行DATA用户接口, 与用户程序进行交互, 并完成了与用户时钟的时钟域转换, 该模块接收来自DATA引擎的数据, 并缓存至多16通道, 每通道8帧的数据, 每个通道中的数据具有不同优先级, 为用户提供更加多样化的接口; DATA上行引擎模块提供了上行DATA用户接口, 与用户程序进行交互, 并完成了与用户时钟的时钟域转换, 该模块与用户程序交互, 接收用户数据, 并向前传输至DATA报文处理模块, 上行方向可为用户提供4通道不同优先级的选择。

[0149] B24. 所述CONF报文处理模块, 如图3所示, 对于寄存器写请求报文, 将报文解析为地址与数据, 并向后传输至CONF引擎, CONF引擎完成寄存器写操作后返回完成信号, 此时CONF报文处理模块可以对下一个报文进行处理; 对于寄存器读请求报文, 将报文解析为地址, 并向后传输至CONF引擎, CONF引擎返回读取值和寄存器读完成信号后向前提交至传输控制模块, 并由传输控制模块提交至USB开发板。

[0150] B25. 所述CONF引擎, 如图3所示, 包括通信系统内部寄存器模块与用户寄存器模块, CONF引擎接收来自CONF报文处理模块的读写请求, 完成配置寄存器的读写, 对于写请求

报文,返回寄存器写完成信号,对于读请求报文,返回读取值和寄存器读完成信号。其中通信系统内部寄存器模块为通信系统内部使用,提供了版本信息,中断控制等配置信息,用户寄存器模块为用户配置寄存器模块,该模块提供了CONF用户接口,与用户程序进行交互,并完成了与用户时钟的时钟域转换。

[0151] B26.所述INT控制模块,如图3所示,提供了INT用户接口,与用户程序进行交互,并完成了与用户时钟的时钟域转换,该模块将中断请求封装为中断报文,提交至发送引擎模块并最终到达用户软件。

[0152] 上述FPGA开发板的硬件部分提供由软件控制的复位操作,可支持两种硬件复位操作,分别为强复位与弱复位。两种复位均可复位通信系统内部逻辑,并向用户程序提供复位信号以复位用户逻辑,此外,强复位能够复位USB开发板固件。

[0153] 本实施例中的USB开发板硬件部分需完成板载芯片的固件编写。固件代码通过调用CYPRESS公司提供的各种函数,完成了板载芯片中实时操作系统的初始化、线程创建、芯片配置、错误处理等操作,在USB 3.0工作模式下,支持2个双向端口,最大数据包长度1024Byte,最大支持连续16次突发传输;另外,该固件支持由计算机软件程序复位USB开发板固件,并同时向FPGA发出复位信号,复位FPGA硬件。

[0154] 本实施例中的USB开发板固件工作于SlaveFIFO模式,并自动完成由SlaveFIFO接口至USB总线的直接内存访问操作,保证了USB开发板的高速传输。SlaveFIFO模式是USB开发板提供的一种类FIFO的接口,可以方便地进行数据传输,大大降低了开发难度。

[0155] 本实施例中的USB开发板提供两个双向端口,其中一个端口传输DATA报文,另一个端口传输CONF与INT报文,通过传输端口区分不同报文,简化了报文解析的难度,也增加了USB通信系统的工作效率。

[0156] 本实施例中的USB开发板最大支持连续16次突发传输。突发传输为USB 3.0规范支持的新特性,即在USB总线传输中,可以单方向连续传输16个数据包再接收每个数据包对应的确认报文。本实例中对该特性的支持加快了USB通信系统的传输速度。

[0157] 需要注意的是,公布实施例的目的在于帮助进一步理解本发明,但是本领域的技术人员可以理解:在不脱离本发明及所附权利要求的精神和范围内,各种替换和修改都是可能的。因此,本发明不应局限于实施例所公开的内容,本发明要求保护的范围以权利要求书界定的范围为准。



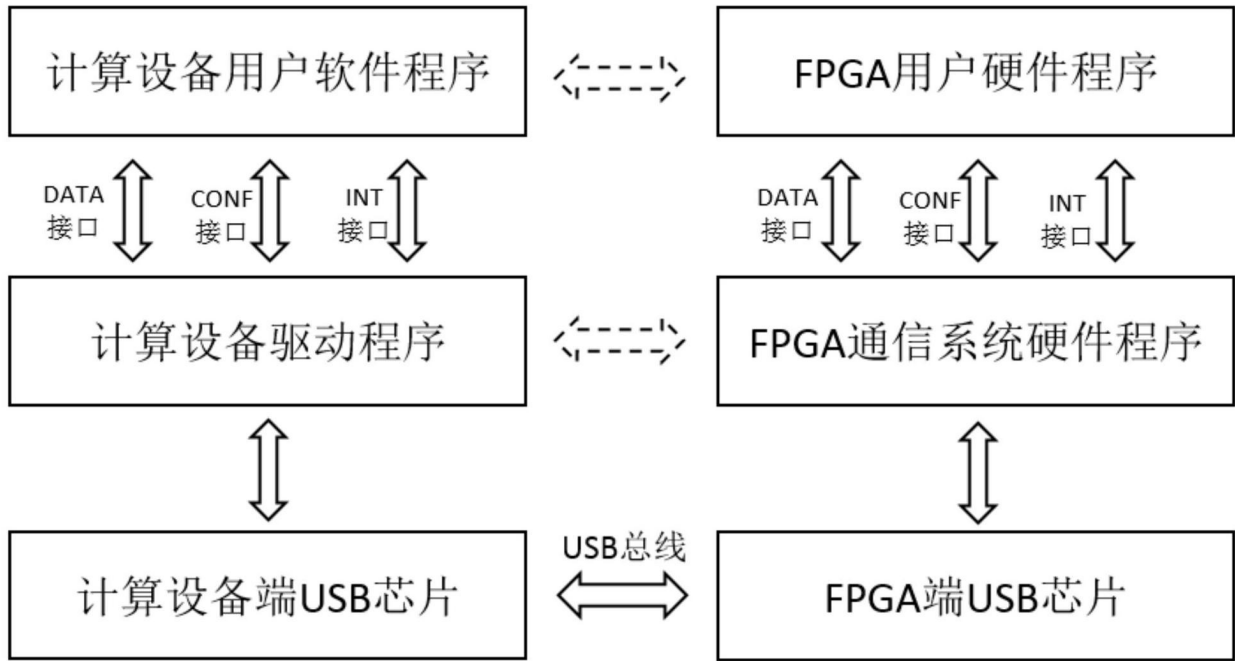


图1

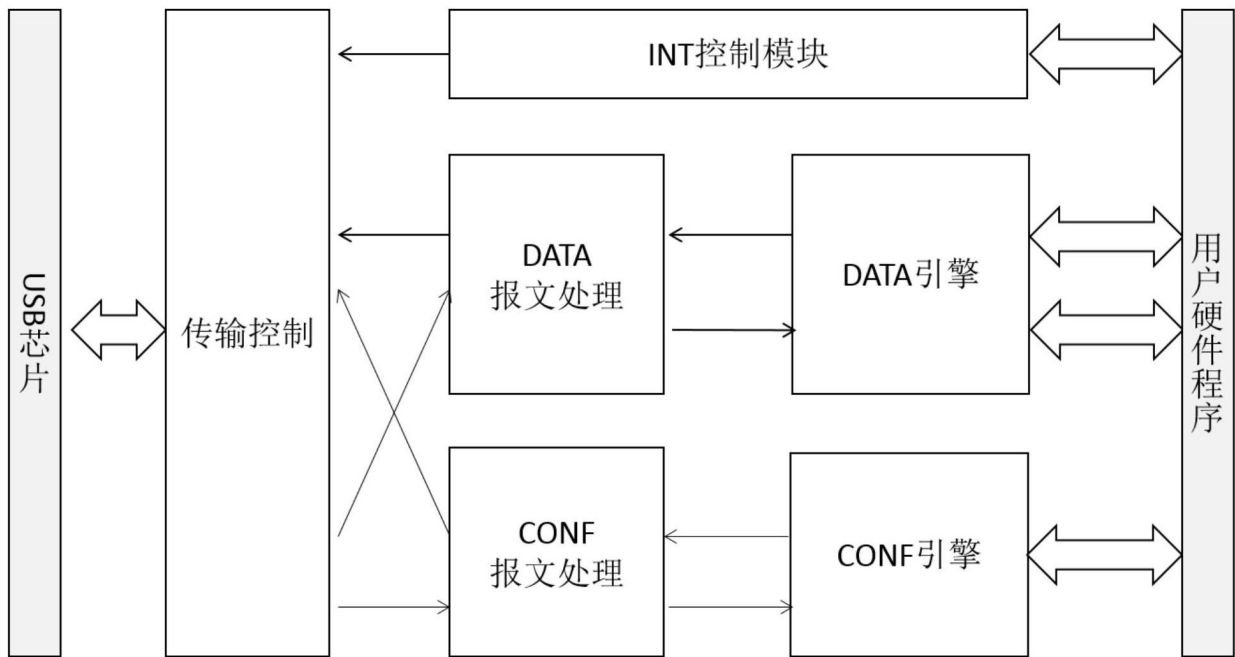


图2

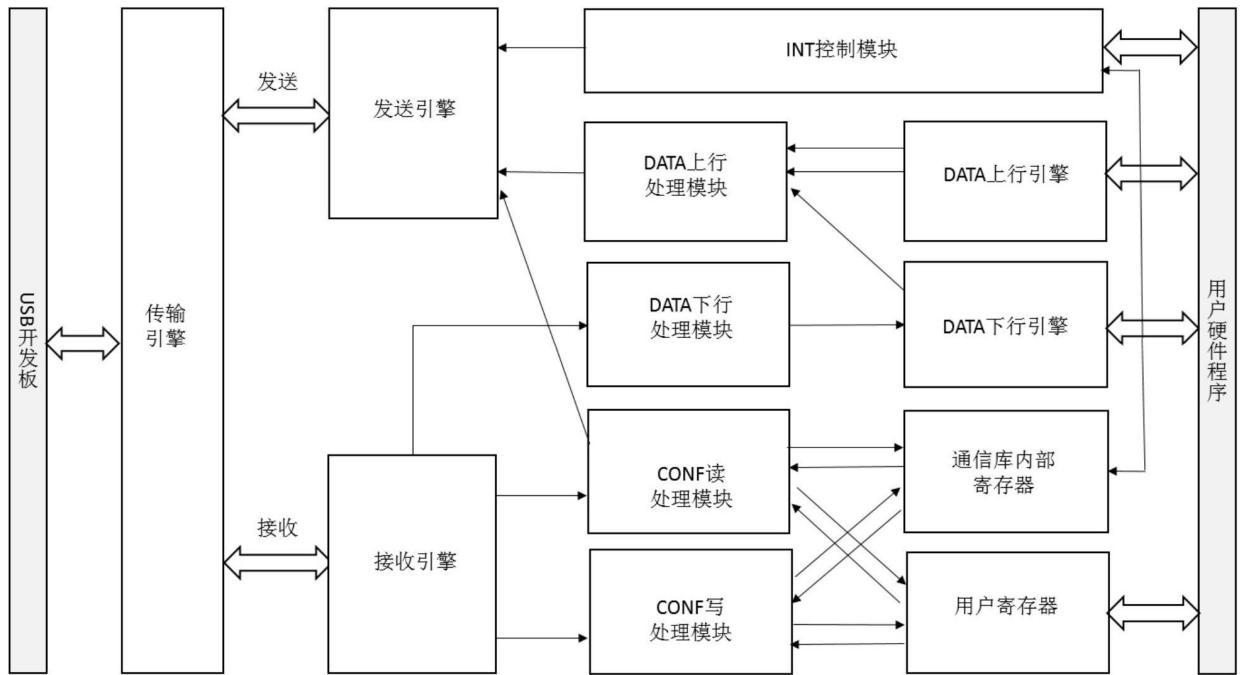


图3