



(12) 发明专利

(10) 授权公告号 CN 102541793 B

(45) 授权公告日 2014. 10. 08

(21) 申请号 201110440502. 2

(22) 申请日 2011. 12. 26

(73) 专利权人 浙江大学

地址 310027 浙江省杭州市西湖区浙大路 38 号

(72) 发明人 颜福才 叶炜 王超 孙文响 张卫杰 尤天容

(74) 专利代理机构 浙江杭州金通专利事务有限公司 33100

代理人 刘晓春

(51) Int. Cl.

G06F 13/38 (2006. 01)

(56) 对比文件

US 6899627 B2, 2005. 05. 31, 全文.

陆海, 胡伯涵. UDP 协议中分包与重组方法研究. 《无线通信技术》. 2010, 第 36 卷 (第 5 期), 第 20 页第 2 栏倒数第 1 段至第 21 页第 1 栏第 1 -

2 段, 图 1; 第 21 页第 2 栏 1. 2 节步骤 3.

吴剑, 王广志, 丁辉. 基于单片机的多并行接口设备数据传输的集成和控制. 《计算机工程与应用》. 2003, (第 35 期), 参见第 89 页第 1 栏第 1 段.

王峰. 基于 PL - 2301 实现网络隔离下的双机通信. 《信息安全与通信保密》. 2011, (第 6 期), 第 106 页第 2 节, 图 2 - 3, .

审查员 彭明明

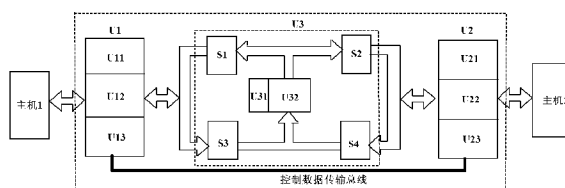
权利要求书 2 页 说明书 10 页 附图 4 页

(54) 发明名称

一种基于 USB 的高速双机数据交换方法及数据交换装置

(57) 摘要

本发明属于计算机通信技术领域, 公开了一种 USB2. 0 高速双机数据交换方法和基于该方法的双机数据交换装置, 所述方法通过分包再重组的方法, 采用多管道并发的方式传输数据, 以减少宽带空闲, 实现对 USB2. 0 带宽的充分利用。有效的提高了双机数据交换的速率。所述装置由两个外设控制器和一个数据中转控制单元组成, 所述两个外设控制器均设有用于与需要进行数据交换的主机进行连接的 USB 接口, 所述数据中转控制单元通过两个并行数据输入输出接口分别与两个外设控制器相连。该装置数据传输速度可达到 50MB/s, 达到了 USB2. 0 的极限传输速度, 是普通通过路由器数据交换速率的 5 至 100 倍, 是普通 USB 设备传输速度的 2 倍。



1. 一种基于 USB 的高速双机数据交换方法,包括如下步骤:

发送方主机将待传数据分包处理并连续编号后,经 USB 接口通过多个并发管道发送至发送方外设控制器;

发送方外设控制器将待传数据经并行数据接口发送至共享缓冲区;

接收方外设控制器经并行数据总线从共享缓冲区读取待传数据后,接收方主机经 USB 接口通过多个并发管道读取接收方外设控制器中的待传数据;

接收方主机按编号顺序重组从接收方外设控制器中读取的数据;

其特征在于,收发双方主机通过如下步骤实现数据连接:

发送方主机通过端点 0 向发送方外设控制器发送发送数据请求特征码,发送方外设控制器再通过控制数据传输总线将该特征码传送至接收方外设控制器,接收方主机轮询外设控制器时从其中接收该特征码;

接收方主机收到发送数据请求特征码后,通过端点 0 发送接收数据应答特征码,经接收方外设控制器、发送方外设控制器,最后到发送方主机;

发送方外设控制器开通其到共享缓冲区方向的数据线路,同时将内部的端点全部设置为输出,接收方外设控制器开通共享缓冲区到其方向的数据线路,同时将内部的端点全部设置为输入;

发送方和接收方外设控制器分别通过端点 0 向发送方主机和接收方主机发送通道已建成特征码;

发送方主机收到通道已建成特征码后开始数据发送,接收方主机收到通道已建成特征码后准备数据接收,并在数据到达后开始接收数据。

2. 根据权利要求 1 所述的基于 USB 的高速双机数据交换方法,其特征在于,还包括:

发送方主机设置应用程序缓冲区、USB 主机控制器缓冲区的步骤,

和接收方主机设置应用程序缓冲区、USB 主机控制器缓冲区的步骤,

所述 USB 主机控制器缓冲区数量与所连的外设控制器块传输端点数量相同,所述的应用程序缓冲区按传输端点数量分组,每组应用缓冲区固定连接到一个端点,一个应用程序缓冲区和一个端点形成一个管道,所有的应用程序缓冲区和所有的端点形成多个并发管道,主机通过多个并发管道与外设控制器传输数据时,按照固定的传输顺序与各个管道传输数据,所述的传输顺序以管道所连的端点地址为基本顺序,每次传输时选择与传输端点相连的一条管道。

3. 根据权利要求 1 所述的基于 USB 的高速双机数据交换方法,其特征在于,还包括错误数据重传步骤:

根据数据包编号判断错误数据包是否还在所述的共享缓冲区,若是,重新读取共享缓冲区中的数据包;

若否,则发送数据重传特征码和错误数据包编号至发送方主机;

发送方主机接收到数据重传特征码及错误数据包编号后,重新发送该编号数据包。

4. 根据权利要求 1 所述的基于 USB 的高速双机数据交换方法,其特征在于,发送方外设控制器与共享缓冲区之间、及接收方外设控制器与共享缓冲区之间均采用单向全带宽传输方式进行数据传输。

5. 根据权利要求 1 所述的基于 USB 的高速双机数据交换方法,其特征在于,收发双方外

设控制器的用于数据传输的端点设置为 4 个块传输端点；收发双方外设控制器与收发双方主机之间建立的管道数量为 12 个，每三个管道为一组共用一个外设控制器的块传输端点。

6. 根据权利要求 1 所述的基于 USB 的高速双机数据交换方法，其特征在于，收发双方与外设控制器各块端点相连的 USB 主机控制器缓冲区大小均设置为 3M；收发双方与各管道一一对应的应用程序缓冲区大小均设置为 1M；数据包长度设置为 512 字节。

7. 一种基于 USB 的高速双机数据交换装置，其特征在于，包括：

两个外设控制器，其分别通过 USB 接口与需要进行数据交换的两个主机连接并直接通信，其根据控制指令，执行接收直连主机控制器发送的数据包并转发至中转数据存储器，或者读取中转数据存储器内数据包并发送给直连主机控制器的操作；

一个数据中转控制单元，其包括至少一个中转数据存储器、一个数据重读模块和两个分别用于连接两个外设控制器的并行数据输入输出接口，其单周期单向接收一个外设控制器发来的数据包并转存入中转数据存储器供另一个外设控制器读取；

控制数据传输总线，用于两个外设控制器之间的控制数据传输；

所述外设控制器包括一个发送 / 接收缓存器、一个用于从发送 / 接收缓存器读取数据并写入所述的中转数据存储器或者从所述的中转数据存储器读取数据并写入发送 / 接收缓存器的读写控制模块、一个具有运算处理功能的微控制模块，所述发送 / 接收缓存器设有一个用于与主机控制器通信的串行数据输入输出接口和一个与中转数据存储器通信的并行数据输入输出接口，其内的缓存区为块传输端点缓冲区，设置为多个块传输端点；所述发送 / 接收缓冲区的并行数据输入输出接口与数据中转控制单元的并行数据输入输出接口通过并行数据总线连接，所述微控制模块与相连主机控制器通过端点 0 发送或接收特征码数据，所述微控制模块还被设定为通过设置块传输端点的输入输出状态控制主机控制器与发送 / 接收缓存器之间的数据传输及传输方向，通过控制读写控制模块的启停及读写方向来控制发送 / 接收缓存器与中转数据存储器之间的数据传输及传输方向，所述微控制模块的特征码输出脚通过控制数据传输总线与另一个外设控制器的微控制模块的特征码输入脚连接。

8. 根据权利要求 7 所述的双机数据交换装置，其特征在于，所述发送 / 接收缓存器的并行数据输入输出接口和数据中转控制单元的并行数据输入输出接口均为 16 位接口，两者通过 16 位并行数据总线连接。

9. 根据权利要求 7 所述的双机数据交换装置，其特征在于，所述的数据中转控制单元还包括四个具有连通关断控制功能的开关器件，其中，第一开关器件的数据输入端和第二开关器件的数据输入端同时连接到中转数据存储器的数据输出端，第三开关器件的数据输出端和第四开关器件的数据输出端同时连接到中转数据存储器的数据输入端，第一开关器件的数据输出端和第三开关器件的数据输入端连接共同做为数据中转控制单元的一个并行数据输入输出接口，第二开关器件的数据输出端和第四开关器件的数据输入端连接共同做为数据中转控制单元的另一个并行数据输入输出接口；

所述的数据中转控制单元中的中转数据存储器为 FIFO，容量设置在 512KB 及以上，所述的数据重读模块包括一个数据重读启动引脚，当在该引脚上出现数据重读信号时，将 FIFO 的读指针设置为 0；

所述的开关器件可为电子开关、总线选择器或输入缓冲器。

## 一种基于 USB 的高速双机数据交换方法及数据交换装置

### 技术领域

[0001] 本发明属于数据通信技术领域,特别涉及两套内嵌 CPU 设备之间的 USB 数据通信技术,具体是指一种基于 USB 的高速双机数据交换方法及其数据交换装置。

### 背景技术

[0002] USB 是一种应用在计算机领域的接口技术,具有热插拔、易携带、传输速度快等优点。由于 USB 的各种优点,各种操作系统如 windows 系列、linux 系列均支持这种接口技术,这使得 USB 技术应用更为广泛。随着 USB 技术的不断发展和完善,USB 传输速度越来越快,USB2.0 的比特率更是达到了 480Mbps,是普通 PC 机上以太网速度的 4.8 倍,是普通无线局域网速度的 8.9 倍。

[0003] 目前两台电脑之间的数据交换主要是通过路由器、交换机和双机互联网线完成,其数据交换速率均在 10MB/s 以下。在普通情况下,通过路由器连接的两台电脑之间的数据交换速率往往不到 500KB/S。

[0004] 在 USB2.0 协议中,数据传输是按照微帧进行,一个微帧的时间长度为 125 微妙。在 USB2.0 协议中除开握手协议,控制传输等小量的数据传输之外,大量的数据是将数据组成一个数据包进行传输,数据包的长度为 512 字节或 1024 字节。在前面提到的一个微帧里面插入若干个数据包、握手协议和控制传输,由于数据包的长度固定和控制传输数据量小,通常一个微帧里最多仅有 100 多个字节在进行握手协议和控制传输,因此一个微帧里面总会有时间处于空闲状态。例如,长度为 512 字节的数据包,每一个数据包要进行帧同步、PID 字段、端点地址等协议开销共计 55 个字节,那么传输一个数据包所需的传输时间  $T = (512 + 55) * 8 / 480 = 9.45$  微秒,那么一个微帧里面最多能传输 512 字节的数据包数量  $n = 125 / T = 125 / 9.45 = 12.227$ , 由于一个帧里所传输的数据包数量为整数,那么一个微帧里面仅能传输 12 个数据包,因此当采用 512 字节的数据包进行传输时,有效数据传输速率最高为  $V = 12 * 512 * 8 / 1000 = 53.248$  MB/s。同理可以计算出采用 1024 字节作为一个数据包时的最高数据传输速率为 49.152 MB/s。因此通过 USB2.0 进行数据传输有效数据最高速率为 53.248 MB/s。

[0005] 目前也有通过 USB 实现数据传输和双机数据交换的技术,但它们都没有充分利用 USB 带宽,USB2.0 系列产品中数据传输速率最高仅到 30MB/s。之所以没有充分利用 USB 带宽,是因为目前国内大部分 USB2.0 的研发人员不清楚数据帧的传输模式,而英文版 USB2.0 协议原著也未将其讲述透彻,很多人只知道理论最高速率值,但不知道为什么是这个值,因为他们不知道一个微帧的结构,当然也就不知道如何最大限度的开发 USB2.0 的带宽。

[0006] 由于 USB 模型是一种 Host-Slave(主机-外设)主从式结构,在 USB 组成的网络中,有且仅有一个主机控制器,其余均为外设控制器。主机控制器通过轮询的方式访问外设控制器内部的各个端点,看看有没有数据要进行传输,如果有数据要进行传输,主机控制器就根据端点的性质发起一次传输。当有一个外设控制器插入主机控制器所属的 USB 接口后,主机控制器会询问外设控制器的名称、类别、端点数量和端点性质等信息,这个过程叫

做枚举。一个外设控制器如果没有通过枚举，那主机控制器就无法与外设控制器进行通信，在我们的计算机上就会出现提示无法识别的 USB 设备。一个外设控制器通过枚举之后，主机控制器就把它加入自己的轮询对象之中。

[0007] 通过上段叙述可知，主机控制器没有外设控制器所具备的设备类别，没有端点，主机控制器不接受别的主机控制器的轮询，主机控制器只去轮询外设控制器，因此当把两台主机不通过外设而直接通过 USB 总线连接时，任意一台主机控制器都无法通过另一台主机控制器的枚举，甚至有可能因为接口电平不同造成电路烧毁。然而可通过增加外设控制器的方法使两台主机利用它们的 USB 端口通信。每个外设控制器连接到不同的主机，并利用两个外设控制器共享的缓冲器交换数据，这就是 USB2.0 双机数据交换的原理。

[0008] 在目前的 USB2.0 数据双机数据交换技术中，有三个因素制约了数据传输速率达不到极限速度。①难以控制 USB2.0 主机控制器的任务调度，主机控制器的任务调度程序没有外部接口，不能直接控制任务的调度，除非自己烧写任务调度程序到主机控制器内，但这只有理论上的意义。通常情况下，根据个人大量实验证明，主机控制器为每个外部设备分配的带宽仅为 6MB/s，这就是为什么我们的优盘在与计算机传输数据时，1 个 GB 的数据通常要花 2.5 分钟左右时间。②没有更高的速度来提供被传输的数据，也就是说两个外设控制器之间的数据交换速率低，限制了双机数据交换速率。③计算机上的主机控制器驱动程序在将主机控制器数据接收缓冲区内的数据转移到应用程序缓冲区内耗费了大量时间，进一步限制了双机数据交换速率。一台计算机上可外接 127 个 USB 外部设备，每个 USB 外部设备又可包含 16 个端点，由于主机控制器驱动程序必须开辟与每个端点对应的缓冲区，该缓冲区称为主机控制器数据缓冲区，由于一台计算机上的端点数量最多可达 2032 个，所以为保证主机控制器驱动程序的通用性和节省内存空间，主机控制器驱动程序为每个端点默认开辟的大小都比较小。这就导致在海量数据传输时，主机控制器数据缓冲区很容易被填满，填满后必须等待数据转移完后才继续与端点交换数据，这就进一步导致了从端点与主机控制器和主机控制器到应用程序这两个步骤没有并行运行。

## 发明内容

[0009] 本发明的目的是针对现有技术中存在的上述不足，提供一种能达到 USB2.0 极限速度、提高双机数据交换效率、方便实用、稳定可靠的 USB 高速双机数据交换方法及其数据交换装置。

[0010] 为达到上述目的，本发明采取如下技术方案：

[0011] 一种基于 USB 的高速双机数据交换方法，其特征在于：包括如下步骤：

[0012] 发送方主机将待传数据分包处理并连续编号后，经 USB 接口通过多个并发管道发送至发送方外设控制器；

[0013] 发送方外设控制器将待传数据经并行数据接口发送至共享缓冲区；

[0014] 接收方外设控制器经并行数据总线从共享缓冲区读取待传数据后，接收方主机经 USB 接口通过多个并发管道读取接收方外设控制器中的待传数据；

[0015] 接收方主机按编号顺序重组从接收方外设控制器中读取的数据。

[0016] 采用上述方法进行数据交换的收发双方主机通过如下步骤实现数据连接：

[0017] 发送方主机通过端点 0 向发送方外设控制器发送发送数据请求特征码，发送方外

设控制器再通过控制数据传输总线将该特征码传送至接收方外设控制器,接收方主机轮询外设控制器时从其中接收该特征码;

[0018] 接收方主机收到发送数据请求特征码后,通过端点 0 发送接收数据应答特征码,经接收方外设控制器、发送方外设控制器,最后到发送方主机;

[0019] 发送方外设控制器开通其到共享缓冲区方向的数据线路,同时将内部的端点全部设置为输出,接收方外设控制器开通共享缓冲区到其方向的数据线路,同时将内部的端点全部设置为输入;

[0020] 发送方和接收方外设控制器分别通过端点 0 向发送方主机和接收方主机发送通道已建成特征码;

[0021] 发送方主机收到通道已建成特征码后开始数据发送,接收方主机收到通道已建成特征码后准备数据接收,并在数据到达后开始接收数据。

[0022] 本发明提出的上述高速双机数据交换方法,通过分包再重组的方法,采用多管道并发的方式传输数据,以减少宽带空闲,实现对 USB2.0 带宽的充分利用。有效的提高了双机数据交换的速率。收发双方外设控制器采用并行数据接口进行数据传输,保证了外设控制器之间的高速数据交换,更有效的解决了外设控制器传输速率低限制双机数据交换速率的问题。

[0023] 为了进一步提高双机数据交换速率,本发明还提出了进一步改方案,即在上述方法中增加:

[0024] 发送方主机设置应用程序缓冲区、USB 主机控制器缓冲区的步骤,

[0025] 和接收方主机设置应用程序缓冲区、USB 主机控制器缓冲区的步骤,

[0026] 所述 USB 主机控制器缓冲区数量与所连的外设控制器块传输端点数量相同,所述的应用程序缓冲区按传输端点数量分组,每组应用缓冲区固定连接到一个端点,一个应用程序缓冲区和一个端点形成一个管道,所有的应用程序缓冲区和所有的端点形成多个并发管道,主机通过多个并发管道与外设控制器传输数据时,按照固定的传输顺序与各个管道传输数据,所述的传输顺序以管道所连的端点地址为基本顺序,每次传输时选择与传输端点相连的一条管道。

[0027] 为各管道设置应用程序缓冲区和为各块传输端点设置 USB 主机控制器缓冲区,可以有效避免在数据传输过程中,主机 CPU 和应用程序等待相应管道或块传输端点上单个数据包传输完毕,或减少等待单个数据包传输完毕的时间,可以进一步提高传输效率。

[0028] 为了防止数据传出错,提高双机数据交换的可靠性,作为更进一步改进,在上述方法再加入错误数据重传步骤:

[0029] 根据数据包编号判断错误数据包是否还在所述的共享缓冲区,若是,重新读取共享缓冲区中的数据包;

[0030] 若否,则发送数据重传特征码和错误数据包编号至发送方主机;

[0031] 发送方主机接收到数据重传特征码及错误数据包编号后,重新发送该编号数据包。

[0032] 在本发明的方法中,鉴于双机间的大数据量交换任务一般为一方到另一方的单向数据传送,为了降低数据传输对收发双方外设控制器间的带宽要求,压缩硬件成本,发送方外设控制器与共享缓冲区之间、及接收方外设控制器与共享缓冲区之间均优选采用单向全

带宽传输方式进行数据传输。

[0033] 由于 USB 接口属于串行数据接口,虽然增设数据管道可以有效减少带宽空闲,进而提高传输速率,但是毕竟 USB 接口带宽有限,传输速率并非随着管道数量的增加而同比增大。根据实验证实,当收发双方外设控制器设置的用于数据传输的块传输端点设置为 4 个;收发双方外设控制器与收发双方主机之间建立的管道数量设置为 12 个,每三个管道一组共用一个外设控制器的块端点时,数据传输速率基本达到 USB 接口带宽的极限。

[0034] 虽然为各块传输端点和管道设置足够大的主机控制器缓冲区和应用程序缓冲区可以有效避免因等待相应管道或块传输端点上单个数据包传输完毕而导致的 USB 带宽空闲,但是过大的缓冲区的设置也同样会造成浪费。

[0035] 实验证明,收发双方与外设控制器各块传输端点相连的 USB 主机控制器缓冲区大小均设置为 3M;收发双方与各管道一一对应的应用程序缓冲区大小均设置为 1M;并设定数据包长度为 512 字节时,双机传输速率更接近于极限且缓存浪费现象少。

[0036] 基于上述方法,本发明的另一个目的是提供一种基于 USB 的高速双机数据交换装置,其特征在于,包括:

[0037] 两个外设控制器,其分别通过 USB 接口与需要进行数据交换的两个主机连接并直接通信,其根据控制指令,执行接收直连主机控制器发送的数据包并转发至中转数据存储单元,或者读取中转数据存储单元内数据包并发送给直连主机控制器的操作;

[0038] 一个数据中转控制单元,其包括至少一个中转数据存储单元、一个数据重读模块和两个分别用于连接两个外设控制器的并行数据输入输出接口,其单周期单向接收一个外设控制器发来的数据包并转存入中转数据存储单元供另一个外设控制器读取;

[0039] 控制数据传输总线,用于两个外设控制器之间的控制数据传输;

[0040] 其中,所述外设控制器包括一个发送/接收缓存器、一个用于从发送/接收缓存器读取数据并写入所述的中转数据存储单元或者从所述的中转数据存储单元读取数据并写入发送/接收缓存器的读写控制模块、一个具有运算处理功能的微控制模块,所述发送/接收缓存器设有一个用于与主机控制器通信的串行数据输入输出接口和一个与中转数据存储单元通信的并行数据输入输出接口,其内的缓存区为块传输端点缓冲区,设置为多个块传输端点;所述发送/接收缓冲区的并行数据输入输出接口与数据中转控制单元的并行数据输入输出接口通过并行数据总线连接,所述微控制模块与相连主机控制器通过端点 0 发送或接收特征码数据,所述微控制模块还被设定为通过设置块传输端点的输入输出状态控制主机控制器与发送/接收缓存器之间的数据传输及传输方向,通过控制读写控制模块的启停及读写方向来控制发送/接收缓存器与中转数据存储单元之间的数据传输及传输方向,所述微控制模块的特征码输出脚通过控制数据传输总线与另一个外设控制器的微控制模块的特征码输入脚连接。

[0041] 在上述双机数据交换装置中,所述发送/接收缓冲区的并行数据输入输出接口和数据中转控制单元的并行数据输入输出接口均优选为 16 位接口,两者通过 16 位并行数据总线连接。

[0042] 实验证明采用单周期单向读写的方式进行数据交换时,两外设控制器间选用 16 位并行接口已经可以达到 96MB/s 的传输速率,完全可以满足 USB 接口全带宽传输需求。

[0043] 由于现有可作为中转数据存储单元的存储芯片,均为单向输入输出接口。即只有两

个数据接口,分别用于写入数据和读出数据。要实现两个外设控制器之间双向数据传输,至少有两种方案。

[0044] 其一方案是:选用两块存储芯片,分别负责实现第一方外设控制到另一方外设控制器的传输和另一方外设控制器回传给第一方外设控制器的数据传输,即一块存储芯片输入接口接第一方外设控制器,输出接口接另一方外设控制器,而另一块存储芯片的输出接口接第一方外设控制器,输入接口接另一方外设控制器。

[0045] 其二方案是:选用一块存储芯片和四个具有连通关断控制功能的开关器件,通过四个开关器件的天关控制实现第一方外设控制到另一方外设控制器的传输和另一方外设控制器回传给第一方外设控制器的数据传输。

[0046] 其连接结构为:第一开关器件的数据输入端和第二开关器件的数据输入端同时连接到中转数据存储器的数据输出端,第三开关器件的数据输出端和第四开关器件的数据输出端同时连接到中转数据存储器的数据输入端,第一开关器件的数据输出端和第三开关器件的数据输入端连接共同做为数据中转控制单元的一个并行数据输入输出接口,第二开关器件的数据输出端和第四开关器件的数据输入端连接共同做为数据中转控制单元的另一个并行数据输入输出接口;

[0047] 在本发明的双机数据交换装置中,所述的数据中转控制单元中的中转数据存储器优选为 FIFO,容量设置在 512KB 及以上,所述的数据重读模块应包括一个数据重读启动引脚,当在该引脚上出现数据重读信号时,将 FIFO 的读指针设置为 0。所述的开关器件可为电子开关、总线选择器或输入缓冲器;

[0048] 与现有技术相比,本发明的高速双机数据交换装置的优点是:

[0049] 1) 传输速度快,数据传输速度可达到 50MB/s,达到了 USB2.0 的极限传输速度,是普通通过路由器数据交换速率的 5 至 100 倍,是普通 USB 设备传输速度的 2 倍;

[0050] 2) 有错误数据重传模块,数据传输稳定可靠;

[0051] 3) 特别适合由多台计算机组成的系统内部频繁的数据交换;

[0052] 4) 双机互联方便,支持热插拔;

[0053] 5) 携带方便。

## 附图说明

[0054] 图 1 是本发明所述的 USB 高速双机数据交换示意图;

[0055] 图 2 是本发明 USB 高速双机数据交换装置结构示意图;

[0056] 图 3 是本发明 USB 高速双机数据交换装置原理图;

[0057] 图 4 是本发明所使用的数据交换特征码表;

[0058] 图 5 是本发明的数据交换方法操作流程;

[0059] 图 6 是本发明的数据错误处理步骤操作流程。

## 具体实施方式

[0060] 本发明公开了一种基于 USB 的高速双机数据交换方法及其数据交换装置,用于实现两台计算机之间的高速数据交换。为了更清楚地理解本发明的技术内容,特举以下实施例详细说明。



[0061] 参照图 1,该图是本发明实施例的 USB2.0 高速双机数据交换示意图。

[0062] 在该图中,有两台要进行数据交换的主机,分别为主机 1 和主机 2。中间为 USB2.0 高速双机数据交换装置。在图中两台主机上的用户程序都开辟了 12 个大小为 1M 字节的应用程序缓冲区,同时都将与每个端点相连的 USB 主机控制器缓冲区大小设置为 3M 字节,在图中两个 USB2.0 外设控制器配置为 4 个块传输端点。用户程序中的应用程序缓冲区与外设控制器中的端点形成 12 个虚拟管道。发送方主机用户程序将要发送的数据添加数据包编号,分发到用户程序中的应用程序缓冲区通过管道传输到与发送方主机相连接的发送方外设控制器中,发送方外设控制器将数据放到中转数据存储器中,接收方外设控制器从中转数据存储器中读取发送过来的数据,再通过虚拟管道将数据发送到接收方主机用户程序中的应用程序缓冲区中,接收方主机应用程序根据数据包编号将数据重组,从而实现了两台主机之间的数据交换。

[0063] 为方便阐述本发明的 USB2.0 高速双机数据交换方法,先讲述使用本方法进行双机数据交换的 USB2.0 高速双机数据交换装置。

[0064] 请参阅图 2 所示,该 USB2.0 高速双机数据交换装置,包括与主机 1 相连的外设控制器 U1 和与主机 2 相连的外设控制器 U2,U1 包括读写控制模块 U11、发送/接收缓存器 U12、微控制模块单元 U12;U2 包括读写控制模块 U21、发送/接收缓存器 U22、微控制模块 U23;该 USB2.0 高速双机数据交换装置还包括数据中转控制单元 U3;U3 包括一个中转数据存储器 U32、一个数据重读模块 U31、和四个具有连通关断控制功能的开关器件 S,U32 用于存放两台主机之间的交换数据,U31 用于错误数据的及时重新传送。

[0065] 所述四个开关器件 S 中,第一开关器件 S1 的数据输入端和第二开关器件 S2 的数据输入端同时连接到 U32 的数据输出端,第三开关器件 S3 的数据输出端和第四开关器件 S4 的数据输出端同时连接到 U32 的数据输入端,S1 的数据输出端和 S3 的数据输入端连接共同做为 U32 的一个并行数据输入输出接口,S2 的数据输出端和 S4 的数据输入端连接共同做为 U32 的另一个并行数据输入输出接口;

[0066] 所述外设控制器 U1、U2 与开关器件 S1-S4 之间,以及开关器件 S1-S4 与中转数据存储器 U32 之间,均采用 16 位并行数据总线进行通信。

[0067] 由于 USB2.0 协议是半双工数据通信,即在任一时刻只有一个方向上的数据传输,因此中转数据存储器可以考虑只用一块芯片,降低成本。但是价格便宜的存储芯片不支持数据输入输出端的反向,开关器件 S 便解决了此难题,降低了成本。

[0068] 其中 U1 和 U2 为 CY7C68012A-56PVXC 芯片,该芯片具有本发明中数据传输方法所要求的各个特点,内部可配置为 4 个端点,每个端点都具有双缓冲,该芯片内部还集成了主频 48MHZ 的读写控制模块,可实现单周期的数据总线读写,该芯片的数据总线可在 8 位和 16 位之间切换,此外该芯片价格便宜,有利于降低成本。

[0069] U31 和 U32 为 1 片 SN74V293 芯片,该芯片内部集成了 1M\*16 位 FIFO,该芯片还支持多片芯片之间的串并联,适合装置的升级,该芯片内部集成了数据重读模块。S1 至 S4 为 4 片 74LVC16244A 芯片,该芯片驱动能力强,响应速度快,价格便宜。

[0070] 在实际应用中,请参阅图 3 所示,本发明 USB2.0 高速双机数据交换装置原理图,U1 和 U2 的第 8 脚连接到 U3 的 62 脚,U1 和 U2 的第 9 脚连接到 U3 的 75 脚,U1 和 U2 的第 36 脚连接到 U3 的 61 脚,U1 和 U2 的第 37 脚连接到所述的 U3 的 1 脚,U1 的第 20 脚连接到 U3

的 64 脚和 80 脚, U1 的第 40 脚连到 U2 的第 41 脚, U1 的第 41 脚连接到 U2 的第 40 脚, U1 和 U2 的第 42 脚连接到 U31 的第 60 脚, U1 的第 43 脚和 S1、S4 的第 1、24、25、48 脚相连, U1 的第 43 脚和 S2、S3 的第 1、24、25、48 脚相连,

[0071] U12 和 U22 设有端点 2、端点 4、端点 6 和端点 8, U12 的这些端点暂存与主机 1 交换的数据, U11 的读写控制功能实现 U12 内的数据与 U3 进行数据交换, U3 内部的数据按照先进先出的模式通过 U21 的读写控制功能与 U22 进行数据交换, U22 内部的端点 2、4、6 和 8 再与主机 2 进行数据交换。从而实现两台主机之间的数据交换。

[0072] U13 和 U23 设有端点 0, 通过端点 0 可以与主机交互所述的数据交换特征码、枚举信息和错误数据包编号等信息。

[0073] S1、S2、S3 和 S4 的第 1、24、25、48 脚为使能信号, 当使能信号为低时, S 两端的数据总线连通, 当使能信号为高时, S 的输出数据总线呈现高阻状态, 利用此特性, U12 和 U23 可利用使能信号控制数据流动方向。例如当 U1 为发送方时, U1 的第 43 脚设置为高电平, U2 的第 43 脚设置为低电平, 此时 S2 和 S3 连通, S1 和 S4 断开, 此时数据流向为: U12→S2→U32→S3→U22。

[0074] 如图 3 所示, U31 和 U32 采用 SN74V293 芯片, 该芯片的第 75 引脚为 IR (输入准备好) 信号, IR 信号为低表示 U3 还可以写入数据, 第 64 引脚为 OR (输出准备好) 信号, OR 信号为低表示 U3 还可以数据没有读完, 所述的读写控制模块通过采集 IR 和 OR 信号, 决定是否继续读写。

[0075] U31 和 U32 采用 SN74V293 芯片, 该芯片的第 60 脚为重读引脚, 当第 60 脚上出现一个读写时钟周期宽度的负脉冲, U31 便将 U32 内部的读指针设置为 0, 其后处于接收状态的读写控制模块将从 U32 内部的第一个数据开始重新读取, 从而实现了数据重读功能。

[0076] 该数据重读功能可使错误的及时得到重新传输。也就是说如果错误数据经过判断后得知仍然在 U32 内部时, 就可以直接从 U32 内部读取, 不用通知发送方重新发送数据。

[0077] U1 和 U2 采用 CY7C68013A-56PVXC 芯片, 芯片的第 40 脚为中断引脚, 如图 2 所示, U1 的第 41 脚连到 U2 的第 40 脚, U1 的第 40 脚连接到 U2 的第 41 脚, CY7C68012A-56PVXC 芯片的第 44 和 45 引脚为通用 I/O 口, 在此应用中, CY7C68012A-56PVXC 芯片的第 44 和 45 引脚用来模拟 SPI 总线时序形成控制数据传输总线, SPI 总线的中断功能通过 U1 和 U2 的第 41 引脚去中断对方来实现, 通过 SPI 总线实现所述的数据交换特征码的传达, 实现 U1、U2、主机 1 和主机 2 之间的协调运转。为讲述上述的 SPI 总线操作过程, 现假设 U1 需要向 U2 发送特征码, 而 U2 不知道 U1 要发送特征码, 所以 U2 不会去接收特征码, 此时 U1 通过第 41 脚产生一个中断信号, U2 立即进入中断状态, 即等待特征码的发送, 此时 U1 通过 SPI 总线发送特征码, U2 收到特征码后, 根据预设定的特征码, 执行具体操作。

[0078] 为实现 U1、U2、主机 1 和主机 2 之间的协调运转, 需要在 U1、U2、主机 1 和主机 2 之间传递特征码和数据重传时的错误数据包编号, 如上文所述主机 1 和 U1 通过 U1 内部的端点 0 传递特征码和错误数据包编号, U1 和 U2 通过上文所述的 SPI 总线传递特征码和错误数据包编号, U2 再通过端点 0 和主机 2 传递信息, 由于上述传递过程均可双向可逆, 据此可实现 U1、U2、主机 1 和主机 2 之间的协调运转。

[0079] 参照图 4, 本发明的 USB2.0 高速双机数据交换方法中, 使用到的数据交换特征码

分别为：发送数据请求、接收数据应答、发送完数据、接收完数据、通道已建成、通道已撤销、数据重读和数据重传。各数据交换特征码对应的码值如图中所示。为方便描述以下简称特征码。

[0080] 主机应用程序打开时需进行初始化，其流程如下：发送方主机和接收方主机应用程序设置 12 个缓冲区，设置 USB2.0 主机控制器内的数据交换缓冲区大小。U1 和 U2 将内部端点配置为 4 个块传输端点，数据包大小设置为 512 字节。

[0081] 初始化流程仅在主机应用程序打开时进行，之后进行的数据交换操作不再执行此步。

[0082] 图 5 所示为主机 1 为发送方主机，主机 2 为接收方主机的数据交换操作流程图，数据交换过程如下：

[0083] ①数据交换的双机建立起数据交换事务，主机 1 通过端点 0 发送发送数据请求特征码到所连的 U1，U1 再将发送数据请求特征码通过控制数据传输总线发送到 U2，U2 再通过 U2 的端点 0 在主机 2 轮询的时候将发送数据请求特征码发送到接收方主机，同时 U2 将接收数据应答特征码通过控制数据传输总线发送到 U1，U1 再将接收数据应答特征码通过 U1 的端点 0 发送到主机 1 上。主机 1 和主机 2 等待所连的 U1 和 U2 发送通道已建成特征码。U1 和 U2 进入第②步。

[0084] ② USB2.0 高速双机数据交换装置建立起数据交换通道，U1 和 U2 通过控制各自的第 43 引脚，分别控制 S2 和 S4 关断、S1 和 S3 连通，由此开通 U1 向 U2 传输的数据线路，同时 U1 将内部的四个端点设置为输出，U2 将内部的四个端点设置为输入。主机 1 和主机 2 建立 12 个应用程序缓冲区和 U1 和 U2 四个端点之间的管道。其中管道 1、5、9 连接到端点 2，管道 2、6、10 连接到端点 4，管道 3、7、11 连接到端点 6，管道 4、8、12 连接到端点 8。U1 和 U2 分别通过端点 0 向主机 1 和主机 2 发送通道已建成特征码，之后 U1 等待主机 1 发送数据到端点，U2 等待 U31 的 OR（输出准备好）信号变为有效，根据前文所述 OR 信号变为有效表明 U31 中已有数据可供读取，主机 2 则开始轮询 U2 是否有数据要接收。

[0085] ③启动数据交换，主机 1 收到通道已搭建好特征码后，将要发送的数据在每个数据包中添加进数据包编号，通过 12 个管道按管道编号不断向 U1 的 4 个端点发送，主机 1 对所发送的数据包计数，U1 依次循环将 4 个端点的数据包写入到 U31 中，一旦有数据写入，U31 的 OR 信号变为有效，U2 检测到该信号后，从 U31 中读取数据，不断将数据写入到 4 个端点内，此时主机 2 一直在轮询，轮询到 U2 的端点内已有数据，通过 12 个管道按管道编号不断将 U2 端点内的数据读入到用户程序中的 12 个数据缓冲区内，主机 2 内的用户程序再根据数据包编号重新连接，主机 2 在接收的过程中对数据包个数计数。上述过程循环往复，主机 1、U1、U2 和主机 2 并行地执行各自的程序，直至主机 1 宣布数据传输完毕，若数据交换中发现数据错误，申请所述的数据重传操作。在数据发送的过程中，发送方主机不断按照如下顺序分发到各个管道：管道 1—>管道 2—>管道 3—>管道 4—>管道 5—>管道 6—>管道 7—>管道 8—>管道 9—>管道 10—>管道 11—>管道 12。在数据接收的过程中，接收方主机按照同样的顺序从各个管道接收数据。

[0086] ④结束数据交换，主机 1 发现要发送的数据已经发送完，发送发送完数据特征码、数据包个数和最后一个数据包长度到 U1 中，U1 通过控制数据传输总线将上面 3 个信息发送到 U2，U2 再通过端点 0 传输到主机 2 上。主机 2 根据收到的数据包个数判断数据是否接

受完,由此决定是否继续轮询,当主机 2 接收完最后一个数据包后,通过端点 0 发送接收完数据特征码到 U2,U2 再通过控制数据传输总线发送到 U1,U1 通过端点 0 发送到主机 1 上。此时 U1 和 U2 进入第⑤步

[0087] ⑤撤销数据交换通道,U1 和 U2 通过控制各自的第 43 引脚,分别控制 S1、S2、S3、S4 关断,由此撤销 U1 向 U2 传输的数据线路,之后 U1 和 U2 分别通过端点 0 向主机 1 和主机 2 发送通道已撤销特征码。主机 1 和主机 2 断开各个管道。之后主机 1、主机 2、U1 和 U2 进入第⑥步

[0088] ⑥结束数据交换事务。主机 1、主机 2、U1 和 U2 进入空闲状态。

[0089] 图 6 所示为主机 1 为发送方主机,主机 2 为接收方主机的数据错误处理操作流程,上述的数据错误处理操作包括以下步骤:

[0090] ①收方主机发现错误数据包,

[0091] ②判断数据包是否还在所述的中转数据存储器,若是,启动数据重读,并跳到第⑤步。数据重读为重新读取中转数据存储器内的数据。

[0092] ③若错误数据不在所述的中转数据存储器,则发送错误数据包编号至发送方,

[0093] ④发送方重新发送所述的错误数据包编号的数据,

[0094] ⑤结束数据错误处理操作。

[0095] 在背景技术中已谈及,现有的 USB2.0 数据传输技术中,有三个制约因素:①难以控制 USB2.0 主机控制器的任务调度,主机控制器的任务调度程序没有外部接口,不能直接控制任务的调度。②没有更高的速度来提供被传输的数据,也就是说两个外设控制器之间的数据交换速率低,限制了双机数据交换速率。③计算机上的主机控制器驱动程序在将主机控制器数据接收缓冲区内的数据转移到应用程序缓冲区内耗费了大量时间,进一步限制了双机数据交换速率。

[0096] 针对制约因素①,本发明采用了使用多管道并发传输的方法进行改进。在本实施例中,虚拟管道数量设置为 12 个,分别为 1 至 12 号管道,端点数量设置为 4 个,分别为端点 2、4、6、8;其中 1、5、9 号管道使用端点 2,2、6、10 号管道使用端点 4,3、7、11 号管道使用端点 6,其余管道使用端点 8;在主机通过管道与端点之间传输数据时,按照管道编号顺序依次进行分时传输。在本实施例中,端点数量为 4 个,管道也分成 4 组,每组管道包含 3 个管道,在主机通过管道与端点之间传输数据时,按照固定的顺序进行分时传输,该固定顺序可看做一个大循环,一个大循环内包含 3 个小循环,每个小循环中按照端点地址递增的顺序选择与该端点相连的一条管道,而每个小循环选择端点相连的不同的管道,由此一个大循环中使得各个管道均参与到传输中,又不出现重复的管道。

[0097] 虽然主机控制器任务调度程序不可控,但是主机控制器是按照外设地址端点地址递增地轮询各个端点,该多管道并发传输技术的优点就在于,在主机应用程序内进行各个管道的调度,使该调度与主机控制器任务调度顺序尽量吻合,此外通过增加管道数量来增加主机控制器的任务数量,然而这些增加的任务数量全都是用于同一个源和目标之间的数据传输,这就使得一个源和目标之间的数据传输速度迅速上升。

[0098] 但这同时带来一个问题,由于数据是分发到多个管道,最终数据接收方将无法重新连接数据,所以在本发明中为每个数据包添加了数据包编号,接收方可根据数据包编号重新连接数据。

[0099] 通常情况下,一台计算机上外接的设备为 USB 鼠标和 USB 键盘,它们传输的数据量都很少,一秒钟内的数据传输量在 100 字节以下,所占据的带宽基本可以忽略,在这种外围设备少的情况下(也是最普通的情况下),本发明实施例中的 CY7C68013A 将所有端点都用于双机数据交换,而且每个端点都有 3 个管道,这就使得主机控制器为同一个数据传输任务分配了更多的带宽,经实验证明当管道数量多到 8 个时,总的数据传输带宽就达到了 50MB/s 左右,

[0100] 在本发明的实施例中,使用了多达 12 个管道,4 个端点,使一个微帧中填满了数据包,而且每个数据包长度采用 512 字节,空闲带宽少,一个微帧内传输的数据包数量基本上都是 13 个,因此数据传输速度接近了极限速度  $13 \times 512 \times 8 \times 1000 \text{B/s}$ , 等于 53.2MB/s。

[0101] 而在大部分现有的数据传输技术中,没有使用多个管道,一方面原因是没有解决多个数据管道带来的数据重新连接问题,另一方面原因是 USB 主机控制器任务调度程序确实不能有直接控制的接口,所以对同一个数据传输只采用了一个管道,所以分配到的数据带宽少。这就是为什么优盘与计算机传输数据时,一个 GB 的数据往往要传 2.5 分钟的原因。

[0102] 针对制约因素②,在本发明中,发送/接收缓冲区的并行数据输入输出接口和数据中转控制单元的并行数据输入输出接口均优选为 16 位接口,两者通过 16 位并行数据总线连接。外设控制器包括一个用于从发送/接收缓存器读取数据并写入所述的中转数据存储单元或者从所述的中转数据存储单元读取数据并写入发送/接收缓存器的读写控制模块。在本实施例中,该读写控制模块可实现单周期读写,频率高达 48MHz,USB2.0 外设控制器 16 位并行数据总线带宽达到 96MB/s 以上,为实现 USB2.0 极限传输提供强大的后盾。

[0103] 针对制约因素③,在本发明中,与每个端点相连的主机控制器缓冲区都人为设定了较大的容量;在本发明实施例中为与每个端点相连的主机控制器缓冲区分配了 3MB 的空间,该空间可同时容纳 6144 个数据包,为主机控制器与端点和主机控制器与应用程序的并行运行提供了宽裕的数据缓冲能力,使它们完全可以并发运行。在 USB2.0 协议规范中并没有提及它们之间的并行运行,因此在大多数现有的技术中,忽略了该环节,导致它们之间的是串行运行,这就是为什么现有的技术中几乎没有突破 30MB/s 数据传输能力的原因之一 ( $30 \text{MB/s} \approx 53.2/2 = 26.6 \text{MB/s}$ )。

[0104] 在实际的数据传输过程中,由于充分利用了 USB2.0 总线带宽,传输速率达到了 50MB/s。由于采取了数据错误重传机制,数据误码率为 0。

[0105] 在此说明书中,本发明已参照其特定的实施例做了描述。但是,很显然仍可以做出各种修改和变换而不背离本发明的精神和范围。因此说明书和附图应被认为是说明性的而非限定性的。

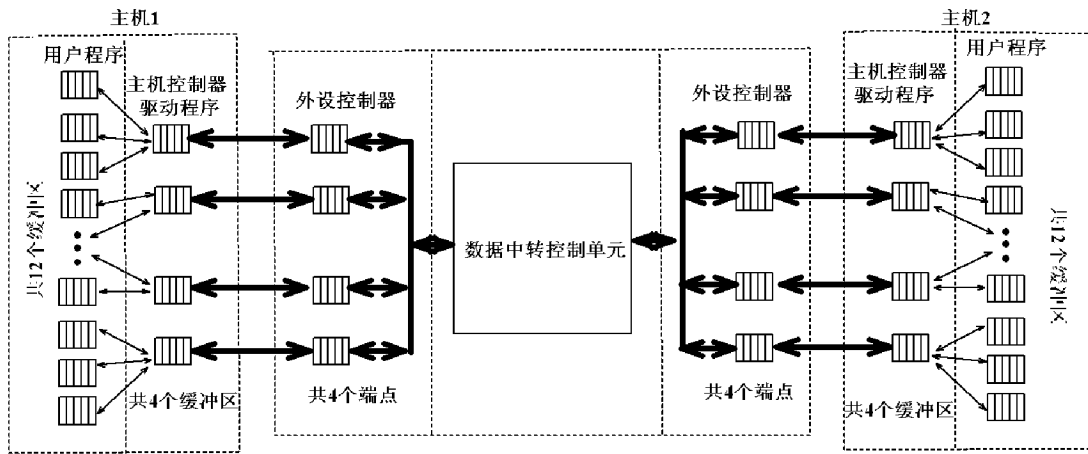


图 1

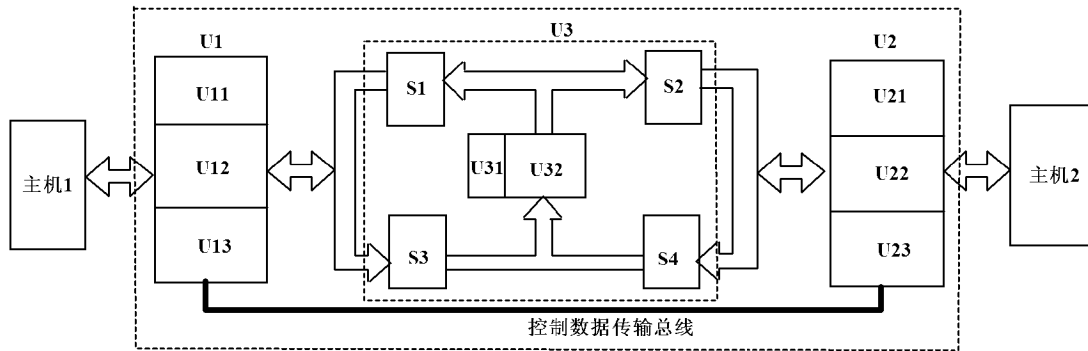


图 2

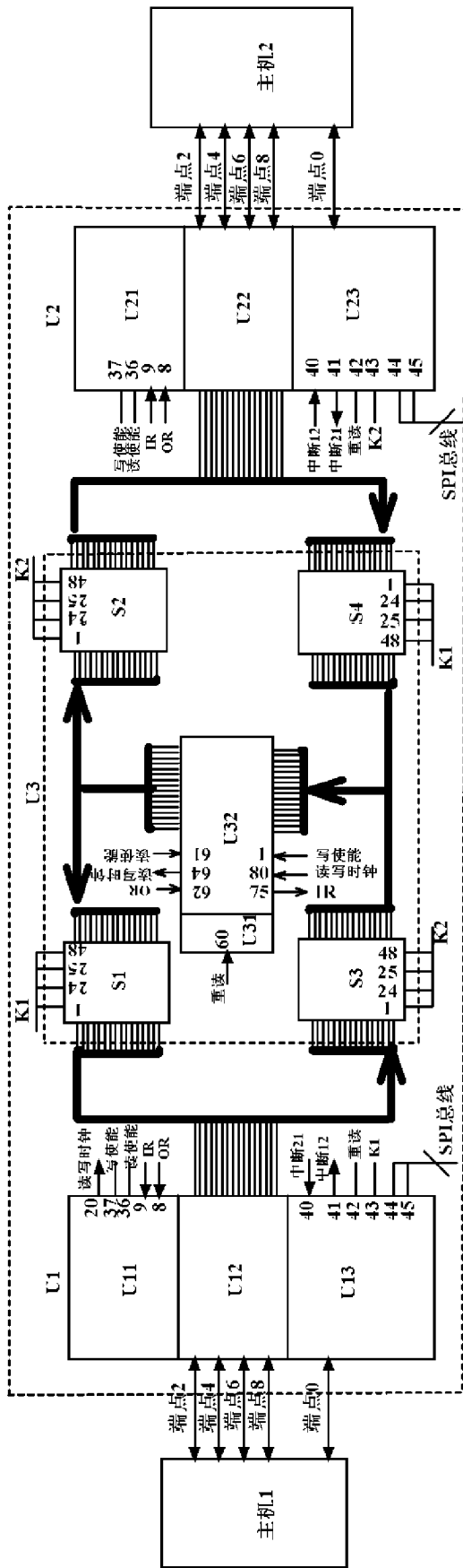


图 3

特征码含义	特征码值
发送数据请求	A0
接收数据应答	A1
发送完数据	A2
接收完数据	A3
通道已建成	A4
通道已撤销	A5
数据重读	A6
数据重传	A7

图 4

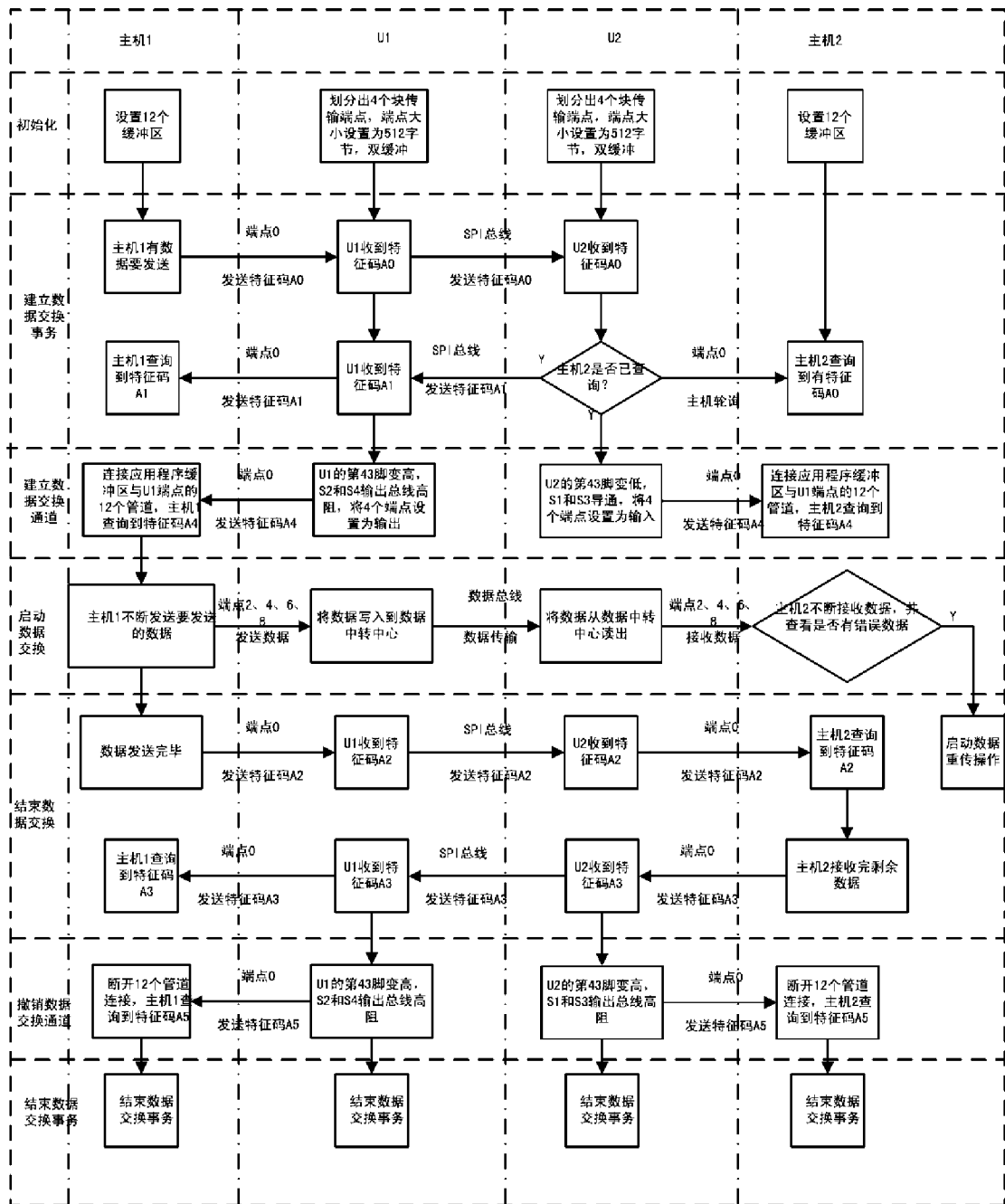


图 5



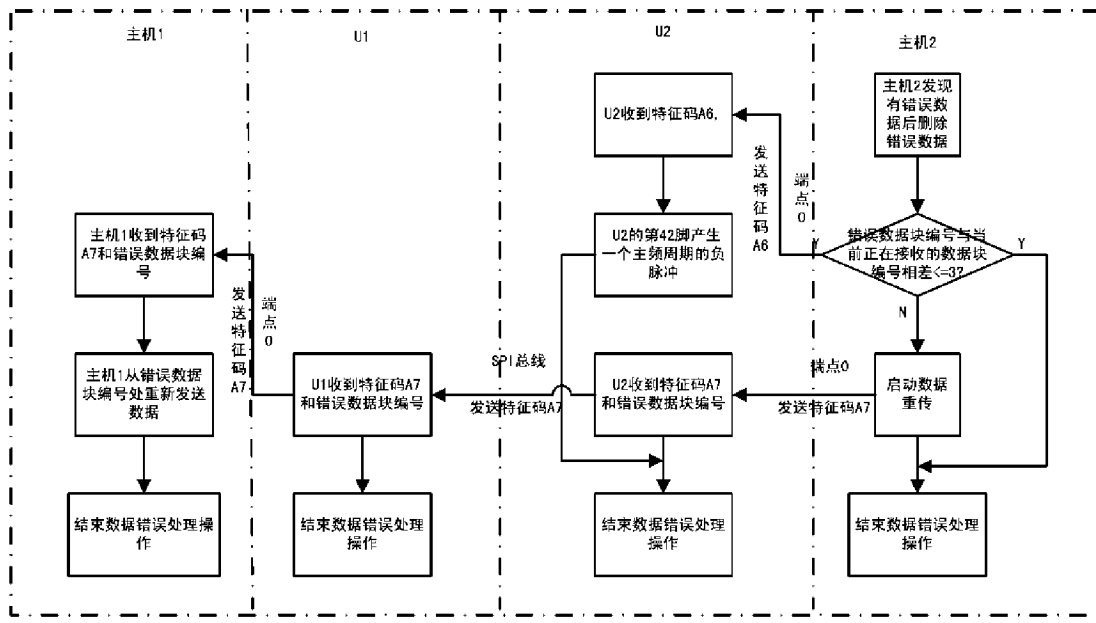


图 6