



(12) 发明专利

(10) 授权公告号 CN 101938493 B

(45) 授权公告日 2013. 10. 16

(21) 申请号 201010287478. 9

H04L 1/00 (2006. 01)

(22) 申请日 2004. 06. 02

H04L 1/24 (2006. 01)

(30) 优先权数据

H04L 7/10 (2006. 01)

60/475, 459 2003. 06. 02 US

(62) 分案原申请数据

(56) 对比文件

200480021020. 1 2004. 06. 02

US 2003191809 A1, 2003. 10. 09, 摘要第 9-11 行、最后 2 行.

(73) 专利权人 高通股份有限公司

WO 03023587 A2, 2003. 03. 20, 全文.

地址 美国加利福尼亚州

US 6014705 A, 2000. 01. 11, 说明书第 8 栏第 50-52 行、第 10 栏第 7-12、38-40 行、16 栏第 66 行至第 17 栏第 3 行、第 17 栏第 3-21 行、第 20 栏第 30-32 行、第 20 栏第 30-32 行.

(72) 发明人 乔恩·詹姆斯·安德森

布赖恩·斯蒂尔 乔治·A·威利
沙尚克·谢卡尔

审查员 贺秀莲

(74) 专利代理机构 北京律盟知识产权代理有限
责任公司 11287

代理人 宋献涛

(51) Int. Cl.

H04L 29/06 (2006. 01)

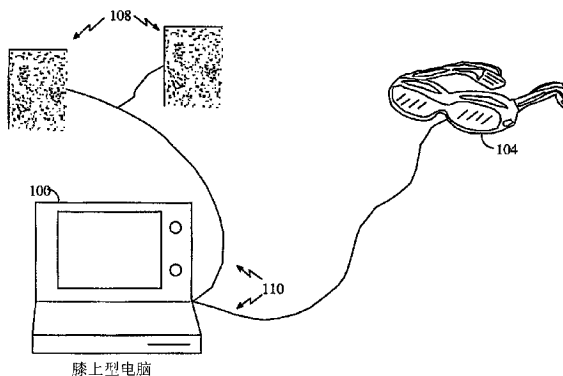
权利要求书1页 说明书62页 附图48页

(54) 发明名称

生成并实施一用于更高数据率的讯号协议和接口

(57) 摘要

本发明提供一种用于在主机和客户端之间经由一通讯路径传送数字数据的数据接口, 所述数据接口使用链接在一起的封包结构来形成一个通讯协议以传递预先选定的一组数字控制和呈现数据。所述讯号协议由链路控制器使用, 所述链路控制器经配置以生成、传输及接收形成所述通讯协议的封包, 并将数字数据形成为一或多种类型的数据封包, 其中至少一个链路控制器驻留在所述主机装置中且经由所述通讯路径耦合至所述客户端。所述接口提供了一种经由短程“串行”型数据链路的有成本效益、低功率、双向、高速的数据传送机制, 其适用于配合在将诸如可穿戴式微型显示器的显示元件连接到便携式电脑和无线通讯装置时特别有用的微型连接器和软性细电缆一起实施。



1. 一种经由一通讯路径在一主机装置和一客户端装置之间以一高速率传送数字数据以呈现给一用户的方法,其包含:

产生复数个预定义封包结构中的一个或一个以上封包结构,且将其链接在一起以形成一预定义的通讯协议;

使用所述通讯协议经由所述通讯路径在所述主机装置和所述客户端装置之间传递预先选定的一组数字控制和呈现数据;

经由所述通讯路径将驻留在所述主机装置中的至少一个主机链路控制器耦合到所述客户端装置,所述主机链路控制器被配置成产生、传输和接收形成所述通讯协议的封包且将数字呈现数据形成为一个或一个以上的数据封包类型;

使用所述链路控制器经由所述通讯路径传送呈封包形式的数据;

通过所述主机驱动一数据线到一高状态来唤醒一通讯链路,其中所述高状态持续至少 150 个时钟循环,并在将所述数据线驱动到所述高状态的 10 个时钟循环内,好像所述数据线为零,通过所述主机开始传输一选通讯号;及

在所述主机已将所述数据线驱动到高并持续所述至少 150 个时钟循环后,由所述主机将所述数据线驱动到低并持续 50 个时钟循环,同时继续传输一选通讯号。

2. 根据权利要求 1 所述的方法,其进一步包含由所述主机开始传输第一子帧标头封包。

3. 根据权利要求 1 所述的方法,其进一步包含由所述客户端计数所述数据线为高的至少 150 个连续时钟循环,接着是所述数据线的为低的至少 50 个连续时钟循环。

4. 根据权利要求 2 所述的方法,其进一步包含由所述客户端搜寻所述第一子帧标头封包的唯一字。

5. 根据权利要求 3 所述的方法,其进一步包含在所述客户端计数了所述数据线为高的 70 个连续时钟循环后,由所述客户端停止驱动所述数据线为高。

6. 根据权利要求 5 所述的方法,其进一步包含由所述客户端计数所述数据线为高的另外 80 个连续时钟循环以达到所述数据线为高的所述 150 个时钟循环,及寻找所述数据线为低的 50 个时钟循环,并寻找唯一字。

生成并实施一用于更高数据率的讯号协议和接口

[0001] 本申请是申请号为 PCT/US2004/017579, 申请日为 2004 年 6 月 2 日, 优先权日为 2003 年 6 月 2 日, 发明名称为“生成并实施一用于更高数据率的讯号协议和接口”的 PCT 申请进入国家阶段后申请号为 200480021021.1 的中国发明专利申请的分案申请。

[0002] 本专利申请案主张对 2003 年 6 月 2 日申请的名称为“Generating and Implementing a Signal Protocol and Interface for Higher Data Rates”的临时申请案第 60/475,459 号的优先权, 所述临时申请案让渡给本受让人且以引用的方式显明地并入本文中。

技术领域

[0003] 本发明关于一种用于以高数据率在主机装置和客户端音频/视觉呈现装置之间通讯或传送讯号的数字讯号协议和过程。更具体来说, 本发明关于一种用于使用低功率高数据率的传送机制将多媒体和其它类型的数字讯号自无线装置传送到微型显示单元或其它呈现装置的技术。

背景技术

[0004] 电脑、电子游戏相关产品和各种视频技术(例如 DVD 和高清晰度 VCR) 在过去几十年中发展迅猛, 使得可将分辨率越来越高的静止、视频、视频点播和图形影像(甚至在包括某些类型的文本时) 呈现给此种设备的最终用户。这些进步又要求使用更高分辨率的电子观察装置, 诸如高清晰度视频监视器、HDTV 监视器或专门的影像投影元件。将此等视觉影像和高清晰度或高质量音频数据(诸如在使用 CD 类声音重现、DVD 和其他也具有相关音频讯号输出的装置时) 的组合用来为最终用户创造一个更逼真、内容更丰富、或更真实的多媒体体验。此外, 已开发了诸如 MP3 播放器的高度移动的高质量声音系统和音乐传送机构, 以向最终用户进行仅有音频的呈现。

[0005] 在典型视频呈现环境中, 通常使用现有技术以可最贴切地称为缓慢或一般的速率(约为每秒一到几十千位) 来传送视频数据。接着这种数据被缓存或储存在短暂或长期存储装置中, 以延后(稍后) 在所观察装置上播放。例如, 可利用驻留于具有调制解调器或互联网连接装置的电脑上的程序来“跨越”或使用互联网传送影像, 以接收或传输对数字地表现一影像有用的数据。类似的传送可使用无线装置来进行, 诸如配有无线调制解调器的便携式电脑、或无线个人数字助理(Personal Data Assistant) (PDA)、或无线电话。

[0006] 一旦被接收到, 数据被在本地存储于存储器元件、电路或装置(诸如 RAM 或闪存) 包括外部存储装置中, 以供播放。取决于数据量和影像分辨率, 播放可能开始地较快或具有较长时间的延迟。意即, 在某些实例中, 影像呈现允许不需要许多数据的非常小或低分辨率影像的某种程度的实时播放, 或允许使用某种类型的缓存, 以使得在短暂延迟后, 在传送更多材料的同时呈现某些材料。倘若传送链路没有中断, 则一旦呈现过程开始, 所述传送对观察装置的最终用户来说是相当透明的。

[0007] 通常用若干熟知技术之一来压缩用于创建静止影像或动态视频的数据以加速

数据经由通讯链路的传送,所述技术可诸如由联合图像专家组 (Joint Photographic Experts Group) (JPEG)、运动图像专家组 (Motion Picture Experts Group) (MPEG) 以及媒体、电脑及通讯产业中的其他知名标准组织或公司所规定的技术。此通过使用较少数量的位来传送给定量的信息而允许更快地传送影像或数据。

[0008] 一旦数据被传送到诸如电脑或其它接收装置的“本地”装置,所得信息就被解压缩 (或使用特殊解码播放器播放),且 (若需要) 被解码,并被预备以基于对应的可用呈现分辨率和控制要素而进行适当呈现。例如,依据 X 乘 Y 像素的屏幕分辨率的典型电脑视频分辨率通常在从低至 480x640 像素、到 600x800 再到 1024x1024 的范围内,尽管在希望或需要的条件下一般可能使用多种其它分辨率。

[0009] 影像呈现还受到以下因素的影响:影像内容,及给定视频控制器在特定预定义彩色等级或色深 (用于产生彩色的每像素位) 及彩色强度方面操控影像的能力,及所使用的任何额外的开销位 (overhead bit)。例如,可预料典型的电脑呈现可使用从每像素大约 8 位到 32 位或更多位中的任何色深来表现各种色彩 (深浅和色调),尽管也会用到其它值。

[0010] 根据以上值,能看出在从最低到最高的典型分辨率和色深的范围上,给定屏幕影像将要求传送从 2.45 兆位 (Mb) 数据到约 33.55Mb 数据中的任一数据量。当以每秒 30 帧的速率观看视频或动态影像时,所需的数据量为约每秒 73.7 到 1,006 兆位 (Mbps) 数据,或约每秒 9.21 到 125.75 兆字节 (MBps)。此外,可能希望结合影像而提供音频数据 (诸如用于多媒体呈现),或提供音频数据作为单独的高分辨率音频呈现 (诸如 CD 音质的音乐)。也可采用处理互动命令、控制或讯号的额外讯号。这些选项中的每一种都会增加甚至更多的待传送数据。总之,当希望将高质量或高分辨率影像数据和高质量音频信息或数据讯号传送给最终用户以创造一种内容丰富的体验时,就会需要在呈现元件和经配置以提供此等类型数据的来源或主机装置之间的高数据传送率链路。

[0011] 约每秒 115 千字节 (KBps) 或每秒 920 千位 (Kbps) 的数据率可常规地由调制解调器串行接口处理。诸如 USB 串行接口的其它接口可适应速率高达 12MBps 的数据传送,且诸如使用美国电气电子工程师学会 (Institute of Electrical and Electronics Engineers) (IEEE) 1394 标准所配置的专门的高速传送可以约 100 到 400MBps 的速率进行。遗憾地,这些速率达不到上文论述的被设想用于未来的无线数据装置和服务的理想的高数据率,所述无线数据装置和服务用于提供高分辨率、内容丰富的输出讯号以驱动便携式视频显示器或音频装置。此外,这些接口要求使用大量主机 (或系统) 和客户端软件来进行操作。其软件协议堆栈也会产生过大量的开销,尤其是在考虑移动无线装置或电话应用的情况下。此等装置具有严格的存储器和功率消耗限制,以及已不堪重负的计算能力。此外,这些接口中的某些采用了对于高度追求美感的移动应用而言太重且不合要求的笨重电缆,采用了增高成本的复杂连接器,或只是耗电太多。

[0012] 存在其它熟知的接口,诸如模拟视频图形适配器 (Analog Video Graphics Adapter) (VGA) 接口、交互式数字视频 (Digital Video Interactive) (DVI) 接口或千兆位视频接口 (Gigabit Video Interface) (GVIF) 接口。头两种接口是并行类型的接口,其以更高传送率处理数据,但也采用沉重的电缆并消耗大量的功率 (约为几瓦)。这些特性都不能满足配合便携式消费性电子装置一起使用的需要。甚至第三种接口也消耗过多功率并使用昂贵或笨重的连接器。

[0013] 对于上述接口中的某些和其它与用于固定安装电脑设备的数据传送相关的极高速率数据系统 / 协议或传送机制来说, 存在另一主要缺点。为适应所要数据传送率还要求大量功率和 / 或在高电流电平下的操作。这极大地降低了此等技术在高度移动的消费型导向的产品中的适用性。

[0014] 一般而言, 使用诸如光纤型连接和传送元件的替代物来适应此等数据传送率也会要求使用导致更大复杂性和成本的若干额外的转接器和元件, 对于真正商用消费型导向的产品是不理想的。迄今光学系统除了一般较昂贵外, 其功率要求和复杂性也阻止了其在重量轻、低功率、便携式应用中的普遍使用。

[0015] 便携式或移动应用产业中缺乏一种为高度移动的最终用户提供高质量的呈现体验 (不管是基于音频、视频还是多媒体) 的技术。换句话说, 当使用便携式电脑、无线电话、PDA、或其它高度移动的通讯装置或设备时, 当前使用的视频和音频呈现系统或装置无法以理想的高质量水平提供输出。通常, 所察觉到的质量欠缺是无法获得传送高质量呈现数据所需的高数据率的结果。因此, 需要一种新的传送机制来增加在提供数据的主机装置和向最终用户呈现一输出的客户端显示装置或元件之间的数据吞吐量。

[0016] 本申请人已在美国专利申请案第 10/020, 520 号和第 10/236, 657 号中提出该等新的传送机制, 两案名称均为“Generating And Implementing A Communication Protocol And Interface For High Data Rate Signal Transfer”, 该等案被让渡给本发明的受让人且以引用的方式并入本文中。那些申请案中论述的技术可极大地提高高速数据讯号中的大量数据的传送率。然而, 尤其是在视频呈现方面, 对不断增大的数据率的需求还在增长。即使有了数据讯号技术中其它正在进行的发展, 仍需要争取获得甚至更快的传送率。因此开发一种增加主机和客户端装置之间的数据吞吐量所需的新的或改进的传送机制是一个持久的需要。

发明内容

[0017] 存在于此项技术中的上述缺点及其他问题由本发明的实施例解决, 在本发明中已开发了一种新的协议和数据传送机制, 以在主机装置和接收的客户端装置之间以高数据率传送数据。

[0018] 本发明之实施例是针对用于经由一通讯路径在主机装置和客户端装置之间以高速率传送数字数据的移动数据数字接口 (Mobile Data Digital Interface) (MDDI), 其采用链接在一起的复数个或一系列封包结构以形成一个用于在主机和客户端装置之间通讯预先选定的一组数字控制和呈现数据的通讯协议。所述讯号通讯协议或链路层由主机或客户端链路控制器的物理层使用。位在主机装置中的至少一链路控制器经由所述通讯路径或链路而耦合到客户端装置, 且经配置以生成、传输、及接收形成所述通讯协议的封包, 并将数字呈现数据形成为一或多类数据封包。所述接口提供了主机和客户端之间的双向信息传送。

[0019] 在本发明的实施例的其它态样中, 至少一客户端链路控制器 (或客户端接收器) 被安置在客户端装置中且经由通讯路径或链路而耦合到主机装置。客户端链路控制器也经配置以生成、传输、及接收形成通讯协议的封包, 并将数字呈现数据形成为一或多类数据封包。一般来说, 主机或链路控制器采用一状态机来处理用于命令或某些类型的讯号预备和

查询处理中的数据封包,但可使用一更慢的通用处理器来操控数据和通讯协议中所使用的较简单封包中的某些。主机控制器包含一个或一个以上差动线驱动器;而客户端接收器包含耦合到通讯路径的一个或一个以上差动线接收器。

[0020] 封包被分组集中于具有预先定义的固定长度且在主机和客户端装置之间予以通讯的媒体帧中,所述媒体帧具有预定数目的具有不同可变长度的封包。封包各包含一个封包长度域、一个或一个以上封包数据域、和一个循环冗余校验域。在从主机链路控制器传送其它封包的开始处传送或定位一个子帧标头封包(Sub-frame Header Packet)。通讯协议分别使用一个或一个以上视频流(Video Stream)类型封包和音频流(Audio Stream)类型封包来将视频类型数据和音频类型数据从主机经由一个正向链路传送到客户端,以呈现给客户端装置用户。所述通讯协议使用一个或一个以上反向链路封装(Reverse Link Encapsulation)类型封包来将数据从客户端装置传送到主机链路控制器。

[0021] 主机链路控制器生成填充符(Filler)类型封包来占据正向链路传输中的不具有数据的周期。通讯协议使用复数个其它封包来传送视频信息。此等封包包括色彩映射(Color Map)、位块传送(Bit Block Transfer)、位图区域填充(Bitmap Area Fill)、位图图案填充(BitmapPattern Fill)、和透明色彩启用(Transparent Color Enable)类型封包。通讯协议使用用户定义流(User-Defined Stream)类型封包来传送接口用户定义数据。通讯协议使用键盘数据(Keyboard Data)和指向装置数据(Pointing Device Data)类型封包来将数据传送到或传送自与所述客户端装置相关的用户输入装置。通讯协议使用链路关闭(Link Shutdown)类型封包来终止经由所述通讯路径的在任一方向上的数据传送。

[0022] 所述通讯路径一般包含或采用具有一系列四个或四个以上导体和一个屏蔽层的电缆。在某些实施例中,链路控制器包含一个USB数据接口且电缆使用USB型接口以及其它导体。此外,按需要可使用印刷导线或弹性导体。

[0023] 主机链路控制器从客户端装置请求显示器能力信息,以便确定所述客户端经由所述接口能适应何种数据和数据率。客户端链路控制器使用至少一个显示器能力(DisplayCapability)类型封包将显示或呈现能力传达到主机链路控制器。通讯协议使用多个传送模式,每一模式允许在一个给定时间周期中并行传送不同最大数目的数据位,同时每一模式可通过在主机和客户端链路控制器之间进行协商来加以选择。这些传送模式在数据传送期间是动态可调整的,且在正向链路上所使用的同一模式不一定要用在反向链路上。

[0024] 在本发明的某些实施例的其它态样中,主机装置包含一无线通讯装置,诸如无线电话、无线PDA、或其中设置有无线调制解调器的便携式电脑。典型的客户端装置包含诸如微型显示装置的便携式视频显示器和/或便携式音频呈现系统。此外,主机可使用存储构件或元件来存储待传送以呈现给客户端装置用户的呈现或多媒体数据。

附图说明

[0025] 参考附图详细描述本发明的其它特征和优点以及本发明的各种实施例的结构和操作。在图中,相同参考数字一般指示相同、功能类似和/或结构类似的元件或处理步骤,且一元件初次出现时所在的图由参考数字中最左方的数位指示。

[0026] 图1A说明本发明的实施例可于其中操作的基本环境,包括使用结合便携式电脑

来使用的微型显示装置。

[0027] 图 1B 说明本发明的实施例可于其中操作的基本环境,包括使用结合无线收发器来使用的微型显示装置和音频呈现元件。

[0028] 图 2 说明移动数字数据接口的总体概念,具有一个主机和客户端的互连。

[0029] 图 3 说明适用于实现从客户端装置到主机装置的数据传送的封包结构。

[0030] 图 4 说明 MDDI 链路控制器的使用和经由用于 I 型和 U 型接口的物理数据链路导体的在主机和客户端之间传递的讯号类型。

[0031] 图 5 说明 MDDI 链路控制器的使用和经由用于 II 型、III 型和 IV 型接口的物理数据链路导体的在主机和客户端之间传递的讯号类型。

[0032] 图 6 说明用于实施接口协议的帧和子帧的结构。

[0033] 图 7 说明用于实施接口协议的封包的大体结构。

[0034] 图 8 说明子帧标头封包的格式。

[0035] 图 9 说明填充符封包的格式和内容。

[0036] 图 10 说明视频流封包的格式。

[0037] 图 11 说明图 10 的视频数据格式描述符的格式和内容。

[0038] 图 12 说明数据的压缩格式和未压缩格式的使用。

[0039] 图 13 说明音频流封包的格式。

[0040] 图 14 说明数据的字节对准和压缩的 PCM 格式的使用。

[0041] 图 15 说明用户定义流封包的格式。

[0042] 图 16 说明色彩映射封包的格式。

[0043] 图 17 说明反向链路封装封包的格式。

[0044] 图 18 说明显示器能力封包的格式。

[0045] 图 19 说明键盘数据封包的格式。

[0046] 图 20 说明指向装置数据封包的格式。

[0047] 图 21 说明链路关闭封包的格式。

[0048] 图 22 说明显示器请求和状态封包的格式。

[0049] 图 23 说明位块传送封包的格式。

[0050] 图 24 说明位图区域填充封包的格式。

[0051] 图 25 说明位图图案填充封包的格式。

[0052] 图 26 说明通讯链路数据通道封包的格式。

[0053] 图 27 说明接口类型转递请求封包的格式。

[0054] 图 28 说明接口类型确认封包的格式。

[0055] 图 29 说明执行类型转递封包的格式。

[0056] 图 30 说明正向音频通道启用封包的格式。

[0057] 图 31 说明反向音频采样率封包的格式。

[0058] 图 32 说明数字内容保护开销封包的格式。

[0059] 图 33 说明透明色彩启用封包的格式。

[0060] 图 34 说明往返行程延迟测量封包的格式。

[0061] 图 35 说明在往返行程延迟测量封包期间的事件的时序。

- [0062] 图 36 说明适用于实施本发明的 CRC 产生器和校验器的样本实施。
- [0063] 图 37A 说明当发送数据封包时用于图 36 的设备的 CRC 讯号的时序。
- [0064] 图 37B 说明当接收数据封包时用于图 36 的设备的 CRC 讯号的时序。
- [0065] 图 38 说明在无争用情况下的典型服务请求的处理步骤。
- [0066] 图 39 说明在与链路开始 (link start) 有争用的情况下的典型服务请求的在链路重启动序列已开始后确立的处理步骤。
- [0067] 图 40 说明如何使用 DATA-STB 编码来传输一个数据序列。
- [0068] 图 41 说明适用于在主机处从输入数据生成 DATA 和 STB 讯号并接着在客户端处恢复数据的电路。
- [0069] 图 42 说明适用于实施一个实施例的驱动器和终端电阻器。
- [0070] 图 43 说明客户端从主机处获取服务时和主机提供此种服务时所采用的步骤和讯号电平。
- [0071] 图 44 说明在 Data0、其它数据线 (DataX) 和选通线 (Stb) 上的转变之间的相对间隔。
- [0072] 图 45 说明当主机在传送一个封包后禁用主机驱动器时会发生的响应延迟的存在。
- [0073] 图 46 说明当主机启用主机驱动器以传送一个封包时会发生的响应延迟的存在。
- [0074] 图 47 说明在主机接收器输入端, 被传送的数据的时序和选通脉冲的前缘与后缘之间的关系。
- [0075] 图 48 说明开关特性和由反向数据时序产生的对应客户端输出延迟。
- [0076] 图 49 说明使用状态机实施同步所用的讯号处理步骤和条件的高级图。
- [0077] 图 50 说明在采用 MDDI 的系统中的正向和反向路径上进行讯号处理时会遇到的典型延迟量。
- [0078] 图 51 说明边际往返行程延迟测量。
- [0079] 图 52 说明反向链路数据率变化。
- [0080] 图 53 说明反向速率除数比正向链路数据率的值的图形表示。
- [0081] 图 54A 和 54B 说明在接口操作中所采取的步骤。
- [0082] 图 55 说明了接口设备处理封包的概观。
- [0083] 图 56 说明了正向链路封包的格式。
- [0084] 图 57 说明了 I 型链路接口中的传播延迟和偏斜的典型值。
- [0085] 图 58 说明了经由接口的例示性讯号处理在 I 型链路路上的 Data、Stb、和时钟恢复时序。
- [0086] 图 59 说明了 II 型、III 型或 IV 型链路接口中的传播延迟和偏斜的典型值。
- [0087] 图 60A、60B 和 60C 说明了两个数据讯号和 MDDI_Stb 相对于彼此的时序的不同可能性, 分别为理想的、早的、和迟的。
- [0088] 图 61 说明配合 I 型 /II 型接口来使用的接口引脚配置例示性连接器。
- [0089] 图 62A 和 62B 分别说明用于 I 型和 II 型接口的可能的 MDDI_Data 和 MDDI_Stb 波形。
- [0090] 图 63 说明了使用状态机实施同步所用的替代性讯号处理步骤和条件的高级图。

[0091] 图 64 说明了一系列时钟循环和各种反向链路封包位及除数值的时序之间的例示性相对时序。

[0092] 图 65 说明了例示性错误码传送处理。

[0093] 图 66 说明了适用于错误码传送处理的设备。

[0094] 图 67A 说明了用于重载代码的错误码传送处理。

[0095] 图 67B 说明了用于接收代码的错误码传送处理。

[0096] 图 68A 说明了用于主机启始的唤醒的处理步骤。

[0097] 图 68B 说明了用于客户端启始的唤醒的处理步骤。

[0098] 图 68C 说明了在有争用的情况下用于主机和客户端启始的唤醒的处理步骤。

具体实施方式

[0099] I. 概述

[0100] 本发明的大体目的是提供一种如下文论述的移动显示数字接口 (Mobile Display DigitalInterface) (MDDI), 其导致或提供一种允许通过使用“串行”类型的数据链路或通道在主机装置和显示装置之间经由短程通讯链路进行高速或极高速数据传送的节省成本、低功耗的传送机制。这一机制适用于配合在将诸如可穿戴式微型显示器 (眼镜或投影仪) 的显示元件或装置连接到便携式电脑、无线通讯装置或娱乐装置时特别有用的微型连接器和软性细电缆一起实施。

[0101] 本发明的实施例的一个优点是提供了一种低复杂度、低成本、高可靠性、与使用环境配合良好、且非常坚固同时仍十分灵活的数据传送技术。

[0102] 本发明可用于各种情形下来以高速率将一般用于音频、视频或多媒体应用的大量数据从生成或存储该等数据的主机或源装置传达或传送到客户端显示器或呈现装置。下文论述的一个典型应用是将数据从便携式电脑或无线电话或调制解调器传送到诸如小型视频屏幕或可穿戴式微型显示装置 (诸如呈含有小型投影透镜和屏幕的眼镜或头盔的形式) 的视觉显示装置, 或将数据从主机传送到该等组件内的客户端装置。意即, 从处理器到内部屏幕或其它呈现元件。

[0103] MDDI 的特征或特性使得其独立于具体显示技术。这是一种可不考虑数据的内部结构也不考虑其实施的数据或命令的功能方面的高度灵活的高速数据传送机制。这允许调整所传送的数据封包的时序以适应于特定显示装置的特性、或适应于某些装置的独特显示要求、或满足某些 A-V 系统的结合的音频和视频要求。只要可遵循选定协议, 所述接口是显示元件或客户端装置无关的 (agnostic)。此外, 总的串行链路数据或数据率可在几个数量级上变化, 从而允许通讯系统或主机装置设计者最优化成本、功率要求、客户端装置复杂度和显示装置更新速率。

[0104] 所提供的接口主要是用于经由“有线”讯号链路或小型电缆来传送大量高速率数据。然而, 某些应用也可利用无线链路 (包括基于光学的链路), 只要其被配置成使用所开发的用于所述接口协议的相同封包和数据结构且可在功耗或复杂度足够低的条件下维持理想的传送水平来保持实用性。

[0105] II. 环境

[0106] 在图 1A 和 1B 中可看到一个典型的应用, 其中分别展示了与显示装置 104 和 106

以及音频重现系统 108 和 112 通讯数据的便携式或膝上型电脑 100 和无线电话或 PDA 装置 102。此外,图 1A 展示了到较大显示器或屏幕 114 或投影仪 116 的潜在连接,出于清晰的目的,其仅展示于一个图中,但同样可连接到无线装置 102。所述无线装置可正在接收数据或先前已在存储器元件或装置中存储了特定量的多媒体型数据以供随后呈现给无线装置的最终用户来观看和 / 或收听。因为典型的无线装置在多数时候是用于语音和简单文本通讯,其用相当小的显示屏和简单的音频系统(扬声器)来将信息传达给装置 102 的用户。

[0107] 电脑 100 具有大得多的屏幕,但外部音响系统仍然不足,且仍然比不上诸如高清晰度电视或电影屏幕的其它多媒体呈现装置。电脑 100 是用于说明目的,本发明也可配合其它类型的处理器、互动电视游戏或消费性电子装置使用。电脑 100 可采用(但不限于)无线调制解调器或其它用于无线通讯的内建装置,或可按需要使用一条电缆或无线链路连接到此种装置。

[0108] 这使呈现更复杂或“丰富”的数据成为一种不那么有用或愉快的体验。因此,业界正在开发其它机制和装置来向最终用户呈现信息并提供期望的最小程度的愉悦或正面的体验。

[0109] 如先前在上文中所论述,正在开发若干类型的显示装置以向装置 100 的最终用户呈现信息。例如,一个或一个以上公司已开发了多套将影像投射到装置用户的眼前来提供视觉显示的可穿戴眼镜。当此种装置被正确定位时,它可有效地“投射”一幅比提供视觉输出的元件大得多的由用户的眼睛所看到的虚拟影像。也就是说,一个非常小的投影元件允许用户的眼睛“看见”比在典型 LCD 屏幕及类似物的情况下比例大得多的影像。使用更大的虚拟屏幕影像也允许使用比在更有限的 LCD 屏幕显示器的情况下分辨率高得多的影像。其它显示装置可包括(但不限于)小型 LCD 屏幕或各种平板显示元件、用于将影像投射到一个表面上的投影透镜及显示驱动器,等等。

[0110] 也可存在连接到无线装置 102 或电脑 100 或者与其使用相关联的额外元件,以将输出提供给另一用户或提供给另一将讯号又转送到其它地方或加以存储的装置。例如,数据可存储在闪存中,以光学形式存储(例如,使用可写 CD 媒体)或存储在磁性媒体中(诸如在磁带录音机和类似装置中)以供随后使用。

[0111] 此外,现在许多无线装置和电脑具有内建的 MP3 音乐解码能力以及其它高级的音响解码器及系统。便携式电脑通常都具有 CD 和 DVD 播放能力,且有些还具有小型专用闪存读取器以接收预先记录的音频文件。伴随具有此等能力而来的问题是数字音乐文件有潜力提供特色极大增加的丰富体验,但这仅在解码和播放过程够快的情况下才能实现。对于数字视频文件来说也是一样。

[0112] 图 1A 中展示了用于协助声音重现的外部扬声器 114,其也可附带有额外的元件,诸如辅助低音扬声器、或用于前部和后部声音投射的“环绕声”扬声器。同时,扬声器或耳机 108 被显示为内建于图 1B 的微型显示装置 106 的支撑框架或机构中。如将了解的,可使用其它音频或声音重现元件,包括功率放大或声音成形装置。

[0113] 总之,如上文所论述,当希望经由一个或一个以上通讯链路 110 将高质量或高分辨率影像数据和高质量音频信息或数据讯号从一个数据源传送给最终用户时,需要高数据率。也就是说,传送链路 110 明显是如上文论述的数据通讯的一个潜在瓶颈,且限制了系统性能,因为现有传送机制达不到通常所要的高数据率。如上文举例论述的,就诸如 1024 乘

1024 像素的较高影像分辨率来说,当每像素色深为 24-32 位且数据率为 30fps 时,数据率可达到超过 755Mbps 或更高的速率。此外,此等影像可能是作为多媒体呈现的一部分来提供的,所述多媒体呈现包括音频数据和涉及互动游戏或通讯的潜在额外讯号、或各种命令、控制或讯号,从而进一步增加了数据量和数据率。

[0114] 同样明显地是要求用更少的电缆或互连来建立数据链路,这意味着与显示器相关联的移动装置可更易于使用,且更有可能被更多的人采用。这在共同使用多个装置以建立一种完整的音频-视觉体验时尤为正确,且在显示器和音频输出装置的质量水平增加时更是如此。

[0115] 不幸地是,所述更高的数据率超越了可用于传送数据的现有技术水平。需要一种用于在呈现元件和数据源之间的数据传送链路或通讯路径的技术以用高速率传送数据,其应涵盖以下特点:稳定地(较)低功率、轻重量以及尽可能简单和便宜的电缆结构。本申请人已发展出一种新颖技术,或方法和设备,来实现这些和其它目标以允许一系列移动、便携式或甚至是固定位置装置以极高数据率将数据传送到所要的显示器、微型显示器或音频传送元件,同时仍保持理想的低功耗和复杂度。

[0116] III. 高速率数字数据接口系统结构

[0117] 为了创造并有效利用一种新颖的装置接口,已设计一种可使用低功率讯号提供极高数据传送率的讯号协议和系统结构。所述协议是基于一封包和共用帧结构(或链接在一起以形成协议的结构),用于通讯预先选定的一组数据或数据类型以及应用于所述接口的命令或操作结构。

[0118] A. 概述

[0119] 通过 MDDI 链路连接或经由 MDDI 链路通讯的装置称为主机和客户端,客户端通常是某种类型的显示装置。从主机到显示器的数据正向行进(称为正向讯务或链路),且由主机启用的从显示器到主机的数据反向行进(反向讯务或链路)。这说明于图 2 所示的基本配置中。在图 2 中,主机 202 使用一个双向通讯通道 206 连接到客户端 204,所述通讯通道 206 被说明为包含一正向链路 208 和一反向链路 210。然而,这些通道是由共用的一组导体形成的,其数据传送有效地在正向链路操作或反向链路操作之间切换。

[0120] 如本文其它部分所论述,主机包含可从使用本发明中获益的若干类型装置之一。例如,主机 202 可为呈手提式电脑、膝上型电脑、或类似的移动计算装置的形式便携式电脑,其可为 PDA、寻呼装置、或许多无线电话或调制解调器之一。或者,主机 202 可为便携式娱乐或呈现装置,诸如便携式 DVD 或 CD 播放器、或游戏装置。同时,客户端 204 可包含各种适用于向最终用户呈现信息的装置。例如,结合入眼镜的微型显示器、内建于帽子或头盔中的投影装置、内建于车辆中的小型屏幕或甚至是全息元件(诸如在车窗或挡风玻璃中)、或各种扬声器、头戴受话器、或用于呈现高质量音响或音乐的音响系统。然而,所属领域的技术人员将易于了解本发明不限于这些装置,市场上存在许多预期在存储和运输方面或在播放时的呈现方面向最终用户提供高质量的影像和音响的及建议使用的其它装置。本发明适用于增加在各种装置之间的数据吞吐量,以适应实现理想的用户体验所需的高数据率。

[0121] B. 接口类型

[0122] 将 MDD 接口设想为处理在通讯和电脑产业中的五种或更多的相当不同的实际接口类型。此时将这些实际接口类型简单地标示为 I 型、II 型、III 型、IV 型和 U 型。

[0123] I 类接口配置为 6 线 (6-wire) (导体) 接口, 这使其适用于移动或无线电话、PDA、电子书 (e-Book)、电子游戏、及便携式媒体播放器 (诸如 CD 播放器或 MP3 播放器)、及基于类似类型的电子消费性技术的装置。U 类接口配置为 8 线 (8-wire) (导体) 接口, 其更适用于不需要快速更新显示且不具有内建 MDDI 链路控制器的膝上型电脑、笔记本电脑、或台式个人电脑及类似装置或应用。这种接口类型也可以通过所使用的额外的双线通用串行总线 (USB) 接口来识别, 其对于适应在多数个人电脑上的现有操作系统或软件支持极有帮助。U 型接口也可用于一种仅 USB (USB-only) 模式中, 其中显示器仅具有一连接到电脑或类似装置 (例如, 配备有标准 USB 端口的消费性电子装置, 诸如数码相机或视频播放器) 上的标准 USB 端口的 USB 连接器。

[0124] II 型、III 型及 IV 型接口适用于高性能显示器或装置, 且使用具有额外双绞线类型导线的更大更复杂的接线来为数据讯号提供适当的屏蔽和低损耗传送。

[0125] I 型接口传递可包含显示、视频、控制及有限讯令信息的讯号, 且通常用于不要求高分辨率全速率视频数据的装置。这类接口主要用于诸如移动无线装置的装置, 其中通常没有用于连接和传送讯号的 USB 主机 (USB host)。在这一配置中, 移动装置是 MDDI 主装置且充当控制源自主机的通讯链路的“主控 (master)”, 其一般发送显示数据到客户端 (正向讯务或链路)。

[0126] 在这种接口中, 主机通过将一种特殊命令或封包类型发送到客户端 (允许其在规定持续时间内接管总线 (链路) 并将数据作为反向封包发送到主机) 来启用在所述主机处自客户端接收通讯数据 (反向讯务或链路)。这说明于图 3 中, 其中将一类称为封装封包 (下文论述) 的封包用于适应在传送链路上传送反向封包, 从而创建反向链路。为主机轮询显示器以获得数据所分配的时间间隔由主机预定, 且基于每一特定应用的要求。这类半双工双向数据传送在无法使用 USB 端口来从客户端传送信息或数据的情况下尤其有利。

[0127] U 型接口传送十分适用于膝上型和台式应用中的讯号, 在此等应用中, USB 接口受到大量主板或其他硬件以及操作系统软件的广泛支持。新增的 USB 接口的使用允许了使用“即插即用”特性和容易的应用程序配置。包含 USB 也允许了命令、状态、音频数据等等的通用双向流动, 同时可使用双绞线以低功率和高速度传送以客户端装置为目标的视频和音频数据。如下文论述, 可使用其它导线来传送功率。使用 USB 接口的本发明实施例允许经由一组导体的高速传送, 同时主要经由 USB 连接来实施讯令和控制在不使用时关闭且消耗很少功率。

[0128] USB 接口是现代个人电脑设备中十分广泛使用的标准, 且 USB 接口的细节和其操作在此项技术中已熟知, 因此此处不再说明。就 USB 接口来说, 主机和显示器之间的通讯服从通用串行总线规范, 修订版 2.0 (Universal Serial Bus Specification, Revision 2.0)。在其中 USB 是主讯令通道并可能是语音返回通道的使用 U 型接口的应用中, 主机可选地经由 MDDI 串行数据讯号来论询客户端。

[0129] 有能力进行 HDTV 型或类似高分辨率显示的高性能显示器要求速率为约 1.5Gbps 的数据流, 以便支持全动视频 (full-motion video)。II 型接口通过并行传输 2 位来支持高数据率, III 型接口通过并行传输 4 位来支持高数据率, 而 IV 型接口通过并行传送 8 位来支持高数据率。MDDI 所使用的协议允许每一 I 型、II 型、III 型或 IV 型主机一般地与任一 I 型、II 型、III 型或 IV 型客户端或显示器通过协商可使用的最高可能数据率来通讯。将可

称为最无能的装置的能力或可用特性用于设定链路的性能。通常,即使是对于主机和客户端都能够使用 II 型、III 型或 IV 型接口的系统,主机和客户端仍以 I 型接口开始操作。主机接着确定目标客户端或显示器的能力,并以对特定应用适宜的方式协商一到 II 型、III 型抑或 IV 型模式的转递(hand-off)或重新配置操作。

[0130] 主机一般可能使用适当的链路层协议(下文进一步论述),且可能在任何时间减速(stepdown)或再次将操作重新配置到一个较慢模式来节省功率或加速(step up)到一个较快模式来支持更高速度的传送,诸如用于更高分辨率的显示内容。例如,当显示系统从诸如电池的电源切换到交流电源时或当显示媒体的来源切换到较低或较高分辨率格式时,主机可改变显示模式,或者可考虑这些或其他条件或事件的组合并以此为基础来改变显示或数据传送模式。

[0131] 系统也可能在一个方向上使用一种模式通讯数据,而在另一方向上使用另一种模式通讯数据。例如,IV 型接口模式可用于以高速率将数据传送到显示器,而当从外围设备(诸如键盘或指向装置)传送数据到主机装置时使用 I 型或 U 型模式。

[0132] C. 实际接口结构

[0133] 图 4 和 5 中展示了用于在主机和客户端装置之间建立通讯的装置或链路控制器的大体布置。在图 4 和 5 中,MDDI 链路控制器 402 和 502 被展示为安装在主机装置 202 中,且 MDDI 链路控制器 404 和 504 被展示为安装在客户端装置 204 中。如前所述,主机 202 通过包含一系列导体的双向通讯通道 406 而连接到客户端 204。如下文所论述,主机和客户端链路控制器都可作为使用单一电路设计的集成电路来制造,其中所述电路设计可经设定、调整或编程以作为主机控制器(驱动器)或客户端控制器(接收器)而响应。这提供了由较大规模制造单一电路装置所带来的低成本。

[0134] 在图 4 中,也展示了用于实施 MDDI 的 U 型接口版本的 USB 主机装置 408 和 USB 客户端装置 410。用于实施这些功能的电路和装置在此项技术中已熟知,且此处不再详述。

[0135] 在图 5 中,MDDI 链路控制器 502 被展示为安装在主机装置 202' 中且 MDDI 链路控制器 504 被展示为安装在客户端装置 204' 中。如前所述,主机 202' 通过包含一系列导体的双向通讯通道 506 而连接到客户端 204'。如上文所论述,主机链路控制器和客户端链路控制器都可用单一电路设计来制造。

[0136] 图 4 和 5 中也说明了经由 MDDI 链路或所使用的实际导体而在主机和客户端(诸如显示装置)之间所传递的讯号。如图 4 和 5 中所示,用于经由 MDDI 传送数据的主路径或机制使用了标示为 MDDI_Data0+/- 和 MDDI_Stb+/- 的数据讯号。这些讯号各自是经由电缆中的差分对导线传送的低电压数据讯号。对于经由接口发送的每一位,在 MDDI_Data0 的对抑或 MDDI_Stb 的对上仅存在一个转变。这是基于电压的传送机制而非基于电流的,所以静态电流消耗接近零。主机将 MDDI_Stb 讯号驱动到客户端显示器。

[0137] 尽管数据可经由 MDDI_Data 对在正向和反向方向上流动(意即,其为双向传送路径),但主机是数据链路的主控器或控制器。MDDI_Data0 和 MDDI-Stb 讯号路径工作在差分模式下以最大化抗扰性。这些线路上的讯号的数据率由主机所发送的时钟脉冲的速率来确定,且在约 1kbps 直到 400Mbps 或更大的范围内可变。

[0138] II 型接口除 I 型的数据对外还包含一个额外的数据对或导体或路径,称为 MDDI_Data1+/-。III 型接口除 II 型接口的数据对外还包含两个额外的数据对或讯号路径,称为

MDDI_Data2+/- 和 MDDI_Data3+/-。IV 型接口除 III 型接口的数据对外还包含四个额外的数据对或讯号路径,分别称为 :MDDI_data4+/-、MDDI_Data5+/-、MDDI_Data6+/-、和 MDDI_Data7+/-。在上述接口配置中的每一种配置中,主机可使用指定为 MDDI_Pwr 和 MDDI_Gnd 的线对或讯号将功率发送到客户端或显示器。

[0139] 一般仅可用于 U 型配置的一种传送类型是 MDDI USB 连接或讯号路径。MDDI USB 连接包含用于在主机和客户端显示器之间通讯的二级路径。在某些应用中,在主机和客户端之间以较低数据率发送某些信息可能更有利。使用 USB 传送链路允许不具有 MDDI 链路控制器但具有 USB 主机或有限主机能力的装置与配备有 U 型接口的 MDDI 兼容客户端或显示器通讯。可经由 USB 接口有效地传送到显示器的信息实例有 :静态位图、数字音频流、指向装置数据、键盘数据以及控制及状态信息。经由 USB 接口来支持的功能也可使用主 MDDI 高速串行数据路径来实施。尽管上文定义的数据 (参见下文的封包) 可经由 USB 类型接口发送,但是将数据串联成紧接着 (back-to-back) 的封包形式的要求不适用于此种 USB 接口,使用支持 MDDI 类型转递的封包的要求也不适用于此种 USB 接口。

[0140] 下文中,根据接口类型在表 I 中说明了经由 MDDI 链路在主机和客户端 (显示器) 之间传递的讯号的概要。

[0141] 表 I

[0142]

| I 型 | II 型 | III 型 | IV 型 |
|---------------|---------------|---------------|---------------|
| MDDI_Pwr/Gnd | MDDI_Pwr/Gnd | MDDI_Pwr/Gnd | MDDI_Pwr/Gnd |
| MDDI_Stb+/- | MDDI_Stb+/- | MDDI_Stb+/- | MDDI_Stb+/- |
| MDDI_Data0+/- | MDDI_Data0+/- | MDDI_Data0+/- | MDDI_Data0+/- |
| | MDDI_Data1+/- | MDDI_Data1+/- | MDDI_Data1+/- |
| | | MDDI_Data2+/- | MDDI_Data2+/- |
| | | MDDI_Data3+/- | MDDI_Data3+/- |
| U 型 | | | MDDI_Data4+/- |
| MDDI_Pwr/Gnd | | | MDDI_Data5+/- |
| MDDI_Stb+/- | | | MDDI_Data6+/- |
| MDDI_Data0+/- | | | MDDI_Data7+/- |
| MDDI_USB+/- | | | |

[0143] 一般用于实施以上结构及操作的接线标称长度约为 1.5 米且包含三根双绞线导体,每一根又是多股 30AWG 导线。在所述三根双绞线上卷绕或以其他方式形成一个箔片屏蔽覆层,作为额外的排扰线。双绞线和屏蔽排扰导体终止于显示器连接器中,同时所述屏蔽物连接到显示器 (客户端) 的屏蔽物,且如此项技术中熟知的,存在一个覆盖整个电缆的绝缘层。所述导线被配对为 :MDDI_Gnd 和 MDDI_Pwr ;MDDI_Stb+ 和 MDDL_Stb- ;MDDI_Data0+ 和 MDDI_Data0- ;MDDI_Data1+ 和 MDDI_Data1- ;等等。标称电缆直径为约 3.0 毫米,标称阻抗为 85 欧姆 ±10%,且标称 DC 电阻为每 1000 英尺 110 欧姆。讯号传播速度应标称地为 0.66c,经由电缆的最大延迟应少于约 8.0 纳秒。

[0144] D. 数据类型和数据率

[0145] 为实现适用于全部用户体验和应用范围的接口,移动数字数据接口 (MDDI) 提供各种显示器和显示信息、音频变换器、键盘、指向装置以及许多其他可集成入移动显示装置或与移动显示装置协同工作的输入装置、连同控制信息、及其组合的支持。MDD 接口经设

计成能够适应使用最小数目的电缆或导体在正向链路方向抑或反向链路方向上在主机和客户端之间传送的各种潜在数据流类型。同步流和异步流（更新）都受到支持。只要总计的数据率小于或等于最大所要 MDDI 链路速率，就可能使用许多数据类型组合。这些可包括（但不限于）下文表 II 和表 III 中列出的那些项目。

[0146] 表 II

[0147]

| 从主机传送到客户端 | | |
|-----------|---------------------------|-------------|
| 同步视频数据 | 720x480, 12bit, 30f/s | ~124.5 Mbps |
| 同步立体声音频数据 | 44.1kHz, 16bit, 立体声 | ~1.4 Mbps |
| 异步图形数据 | 800x600, 12bit, 10f/s, 立体 | ~115.2 Mbps |
| 异步控制 | 最小值 | <<1.0 Mbps |

[0148] 表 III

[0149]

| 从客户端传送到主机 | | |
|-------------|-----------------------|------------|
| 同步语音数据 | 8 kHz, 8bit | <<1.0 Mbps |
| 同步视频数据 | 640x480, 12bit, 24f/s | ~88.5 Mbps |
| 异步状态, 用户输入等 | 最小值 | <<1.0 Mbps |

[0150] 所述接口并非固定不变的而是可扩展的, 以使得其可支持各种信息“类型”（包括用户定义数据）的传送, 以实现未来系统的灵活性。将包含的数据的具体实例为: 全动视频, 形式为全屏或部分屏幕位图场 (bitmap field) 或经压缩的视频; 低速率静态位图, 以节约功率并减小实施成本; 各种分辨率或速率的压缩音频数据或 PCM; 指向装置位置和选择, 及用于尚未定义的能力的用户可定义数据。也可将此等数据连同控制或状态信息一起传送以侦测装置能力或设定操作参数。

[0151] 本发明推进了用于数据传送中的技术, 包括（但不限于）: 观看电影（视频显示和音频）; 使用具有有限个人检视 (personal viewing)（图形显示器, 有时与视频和音频组合）的个人电脑; 在 PC、游戏机、或个人装置上玩电子游戏（运动图形显示、或人造视频和音频）; 浏览互联网; 使用形式为视频电话（双向低速率视频和音频）、用于拍摄静态数字像片的相机、或用于摄取数字视频影像的可携式摄像机的装置; 及用于移动电话、智能电话或 PDA 的生产率增强或娱乐用途。

[0152] 下文论述的移动数据接口是就经由一般配置为有线线路或电缆类链路的通讯或传送链路而提供大量 A-V 类型数据方面来说的。但是, 显而易见的是所述讯号结构、协议、时序或传送机制可经调整以提供形式为光学或无线媒体的链路（如果它能维持理想的数据传送水平）。

[0153] MDD 接口讯号将称为共同帧 (Common Frame) (CF) 的概念用于了基本讯号协议或结构。使用共同帧所基于的思想是为同时的同步数据流提供同步脉冲。显示装置可使用这种共同帧速率作为时间基准。低 CF 速率通过减少用以传输子帧标头的开销来增加通道效率。另一方面, 高 CF 速率减少了等待时间, 且允许用于音频采样的弹性数据缓冲区较小。本

发明性接口的 CF 速率是可动态编程的,且可设定在对用于特定应用中的同步流为适宜的许多值中的一个值处。也就是说,按需要,CF 值经选择以最适应特定显示装置和主机配置。

[0154] 表 IV 中展示了最有可能用于某应用(诸如头戴式微型显示器)的同步数据流的每共同帧一般所需的字节数(其为可调整或可编程的)。

[0155] 表 IV

[0156]

| 共同帧速率 (CFR) = 1200 Hz | | | | | | | |
|-----------------------|-----|-----|----|-------|----|----------|---------|
| | X | Y | 位 | 帧速率 | 通道 | 速率(Mbps) | 字节/ CFR |
| DVD 电影 | 720 | 480 | 12 | 30 | 1 | 124.4 | 12960 |
| 立体图形 | 800 | 600 | 12 | 10 | 2 | 115.2 | 12000 |
| 便携式摄像机 | 640 | 480 | 12 | 24 | 1 | 88.5 | 9216 |
| CD 音频 | 1 | 1 | 16 | 44100 | 2 | 1.4 | 147 |
| 语音 | 1 | 1 | 8 | 8000 | 1 | 0.1 | 6.7 |

[0157] 通过使用简单的可编程 M/N 计数器结构,易于获得每共同帧字节的分数计数值。例如,通过传送两个 27 字节的帧且每一帧后为一个 26 字节的帧,可实现每 CF 26-2/3 字节的计数值。可选择较小的 CF 速率来产生每 CF 整数个字节。然而,一般来说,在用于实施部分或所有本发明的集成电路芯片或电子模块内,以硬件实施简单 M/N 计数器所需的区域会小于较大音频采样 FIFO 缓冲器所需的区域。

[0158] 一个说明不同数据传送率和数据类型的影响的例示性应用是卡拉 OK 系统。在卡拉 OK 中,系统用户伴着音乐视频节目而歌唱。歌曲的歌词显示在屏幕的底部,使用户知道要唱的歌词和歌唱的大概时机。这种应用要求图形更新不频繁的视频显示,和将用户的语音与立体声音频流混合。

[0159] 如果假定共同帧速率为 300Hz,那么每一 CF 将由经由正向链路到显示装置的 92,160 字节的视频内容和 588 字节的音频内容(基于 147 个 16 位样本,立体声)组成,且平均 29.67(26-2/3)字节的语音被从麦克风发送回到移动卡拉 OK 机。在主机和显示器之间发送异步封包。这包括最多 768 字节的图形数据(四分之一屏幕高度)和少于约 200 字节的用于杂项控制和状态命令的(若干)字节。

[0160] 表 V 展示了如何在用于卡拉 OK 实例的共同帧中分配数据。所使用的总速率被选择为约 225Mbps。略高的 226Mbps 的速率允许传送每子帧另外的约 400 字节数据,其允许使用非经常的控制和状态消息。

[0161] 表 V

[0162]

| 要素速率 | 字节 /CF |
|----------------------------|--------------|
| 640x480 像素和 30fps 下的音乐视频 | 92160 |
| 640x120 像素和 1fps 下的歌词文字 | 768 |
| 44,100sps,立体声,16 位下的 CD 音频 | 588 |
| 8,000sps,单声道,8 位下的语音 | 26.67 |
| 子帧标头 | 19 |
| 反向链路开销 | 26.67+2*9+20 |
| 总字节 /CF | 93626.33 |
| 总速率 (Mbps) | 224.7032 |

[0163] III. 高速率数字数据接口系统基础结构

[0164] E. 链路层

[0165] 使用 MDD 接口高速串行数据讯号传送的数据由彼此链接的分时多工的封包流组成。即使传输装置无数据要发送, MDDI 链路控制器一般也自动发送填充符封包, 因此, 维持了封包流。对简单的封包结构的使用使得视频和音频讯号或数据流具有可靠的同步时序。

[0166] 成组的封包被包含在称为子帧的讯号要素或结构中, 且成组的子帧被包含在称为媒体帧的讯号要素或结构中。取决于封包的各别大小和数据传送用途, 子帧包含一个或一个以上封包, 且媒体帧包含一个或一个以上子帧。由本发明使用的协议所提供的最大的子帧约为 232-1 或 4, 294, 967, 295 字节, 且最大的媒体帧大小就变为约 216-1 或 65, 535 个子帧。

[0167] 如下文论述, 一种特殊的标头封包包含一个出现在每一子帧的开头的唯一识别符。所述识别符也用来当启始了主机和客户端之间的通讯时在客户端装置处获取帧时序 (frametimeing)。下文更详细地论述了链路时序获取。

[0168] 一般来说, 当在显示全动视频时, 针对每媒体帧更新一次显示屏。显示帧速率与媒体帧速率相同。取决于所要的应用, 链路协议支持在整个显示器上的全动视频或只是在由静态影像包围的较小区域中的全动视频内容。在某些低功率移动应用中, 诸如观看网页或电子邮件, 显示屏可能只需要被偶尔更新。在那些情况下, 传输单一子帧并接着关闭或撤消链路来最小化功耗是有利的。所述接口也支持诸如立体视觉的效果, 并处理图形基元 (graphics primitive)。

[0169] 采用子帧的目的是使得可周期地传输高优先权封包。这允许同时的同步流与最小量的数据缓冲并存。这是本发明为显示过程提供的一个优点, 允许了多个数据流 (视频、语音、控制、状态、指向装置数据等等的高速通讯) 实质上共享一个共同通道。其使用较少的讯号来传送信息。其也允许存在特定针对显示技术的动作, 诸如用于 CRT 监视器的水平同步脉冲和消隐信号间隔 (blanking interval)。

[0170] F. 链路控制器

[0171] 图 4 和 5 所示的 MDDI 链路控制器被制造或装配成一种全数字化的实施, 除了用于接收 MDDI 数据和选通讯号的差分线接收器。然而, 甚至是差分线驱动器和接收器也可和链路控制器一起实施于相同数字集成电路中。不需要模拟功能或锁相回路 (PLL) 来实施用于链路控制器的硬件。除了包含用于链路同步的状态机的显示器接口, 主机链路控制器和客户端链路控制器具有非常相似的功能。因此, 本发明提供了以下实际优点, 即能够创造一种可配置为主机抑或客户端的单一控制器设计或电路, 这能在整体上减少链路控制器的制造成本。

[0172] IV. 接口链路协议

[0173] A. 帧结构

[0174] 图 6 中说明了用于实施封包传送的正向链路通讯的讯号协议或帧结构。如图 6 所示, 信息或数字数据被分组为称为封包的要素。多个封包又分组到一起而形成称为“子帧”的要素, 且多个子帧又分组到一起而形成“媒体”帧。为控制帧的形成和子帧的传送, 每一子帧以一个称为子帧标头封包 (SHP) 的特殊预定义封包开始。

[0175] 主机装置选择待用于特定传送的数据率。此速率可由主机装置基于主机的最大传

送能力或主机从数据源提取数据的能力、和显示器或数据传送到的其他装置的最大能力两者来动态地改变。

[0176] 设计用于（或能够）配合 MDDI 或本发明性讯号协议一起工作的接收客户端装置可由主机查询以确定其可使用的最大（或当前最大）的数据传送率（或者可使用缺省的较慢最小速率），以及可用数据类型和所支持的特性。这种信息可使用显示器能力封包（DCP）来传送，如下文进一步论述。客户端显示装置能够使用所述接口以预先选定的最小数据率或在一个最小数据率范围内与其它装置传送数据或通讯，且主机将使用在此范围内的一个数据率来执行查询以确定客户端装置的全部能力。

[0177] 可在状态封包中将定义位图的性质和显示器的视频帧速率能力的其他状态信息传送到主机，使得主机可将接口配置得尽可能地有效或最优，或在任何系统限制下为理想的。

[0178] 当在当前子帧中没有（更多）待传送的数据封包或当主机的传送速率不足以跟上为正向链路选定的数据传送率时，主机发送填充符封包。由于每一子帧以一个子帧标头封包开始，所以前一个子帧的末尾包含一个正好填充前一个子帧的封包（最有可能是一个填充符封包）。在缺少用于数据承载封包本身的空间的情况下，填充符封包将最有可能是子帧中的最后一个封包，或在下一个前一子帧的末端且在子帧标头封包之前。确保在一个子帧中余留了足够用于将在所述子帧中传输的每一封包的空间是主机装置中的控制操作的任务。同时，一旦主机装置启始了数据封包的发送，主机必须能够成功地在一个帧内完成所述尺寸的封包，而不导致数据不足（data under-run）情况。

[0179] 在实施例的一个方面，子帧传输具有两个模式。一个模式是用于传输实况视频和音频流的周期性子帧模式。在这一模式中，子帧长度定义为非零。第二种模式是异步或非周期性模式，其中仅在新信息可用时将帧用于把位图数据提供给显示装置。通过在子帧标头封包中将子帧长度设定为零来定义这一模式。当使用周期模式时，当显示器已同步于正向链路帧结构时可开始子帧封包接收。这对应于根据下文参看图 49 或 63 所论述的状态图而定义的“同步”状态。在非同步非周期性子帧模式中，在接收到第一个子帧标头封包后开始接收。

[0180] B. 整体封包结构

[0181] 用于系统阐述本发明所实施的讯令协议的封包结构的格式陈述在下文中，请注意所述接口是可扩展的且按需要可添加额外的封包结构。根据封包在接口中的功能（换句话说，其传送的命令或数据），所述封包被标示为（或划分成）不同“封包类型”。因此，每一封包类型表示用于一特定封包的预定义封包结构，其用于操控所传送的封包和数据。如将显而易见地，封包可具有预先选定的长度或具有取决于其各别功能而可变或可动态改变的长度。尽管仍会实现相同的功能，所述封包也可带有不同名称，如在协议被接受到一标准中的期间受到改变的情况下会发生的。用于各种封包中的字节或字节值被配置为多位（8 位或 16 位）无符号整数。表 VI 中展示了以类型顺序列出的所使用封包的概要以及其“类型”标号。还注明了封包的传送被认为有效时的方向，以及其是否用于 U 型接口。

[0182] 表 VI

[0183]

| 封包名称 | 封包类型 | 有效方向 | | |
|------------|-------|------|----|----|
| | | 正向 | 反向 | U型 |
| 子帧标头封包 | 255 | X | | X |
| 填充符封包 | 0 | X | X | |
| 视频流封包 | 1 | X | X | X |
| 音频流封包 | 2 | X | X | X |
| 保留流封包 | 3-55 | | | |
| 用户定义流封包 | 56-63 | X | X | X |
| 色彩映射封包 | 64 | X | X | X |
| 反向链路封装封包 | 65 | X | | |
| 显示器能力封包 | 66 | | X | X |
| 键盘数据封包 | 67 | X | X | X |
| 封包名称 | 封包类型 | 有效方向 | | |
| | | 正向 | 反向 | U型 |
| 指向装置数据封包 | 68 | X | X | X |
| 链路关闭封包 | 69 | X | | |
| 显示器请求和状态封包 | 70 | | X | X |
| 位块传送封包 | 71 | X | | X |
| 位图区域填充封包 | 72 | X | | X |
| 位图图案填充封包 | 73 | X | | X |
| 通讯链路数据通道封包 | 74 | X | X | X |
| 接口类型转递请求封包 | 75 | X | | |
| 接口类型确认封包 | 76 | | X | |
| 执行类型转递封包 | 77 | X | | |
| 正向音频通道启用封包 | 78 | X | | X |
| 反向音频采样率封包 | 79 | X | | X |
| 数字内容保护开销封包 | 80 | X | X | X |
| 透明色彩启用封包 | 81 | X | | X |
| 往返行程延迟测量封包 | 82 | X | | |
| 正向链路偏斜校正封包 | 83 | X | | |

[0184] 封包具有一个共同基本结构或总体上最少的一组域,包含封包长度域、封包类型域、数据字节域、和CRC域,其说明在图7中。如图7中所示,封包长度域包含形式为多位或多字节值的信息,所述信息规定了封包中的总位数,或其封包长度域和CRC域之间的长度。在一个实施例中,封包长度域含有规定了封包长度的16位或2字节宽的无符号整数。封包类型域是另一多位域,其标示了封包内所含有的信息类型。在一例示性实施例中,它是8位或1字节宽的值,形式为8位无符号整数,且规定了诸如显示器能力、转递、视频或音频流、状态、等等的数据类型。

[0185] 第三种域是数据字节域,其包含作为所述封包的部分的在主机和客户端装置之间

传送或发送的位或数据。根据所传送的特定类型数据为每一封包类型特定地定义数据格式，且数据格式可分为一系列额外的域，每一个域具有其自己的格式要求。也就是说，每一封包类型将具有用于此部分或域的已定义的格式。最后一个域是 CRC 域，其包含对数据字节域、封包类型域及封包长度域进行计算所得的 16 位循环冗余校验结果，其用于确认封包中的信息的完整性。换句话说，对除 CRC 域本身的整个封包进行计算。客户端一般保存检测到的 CRC 错误的总计数值，且在显示器请求和状态封包（参看下文）中将这一计数值回报给主机。

[0186] 在传送封包期间，以最初传输的最低有效位 (LSB) 为开始且以最后传输的最高有效位 (MSB) 为结尾来传输域。长度大于一个字节的参数通过最低有效字节最先传输的方式来传输，这导致了与用于较短参数的其中 LSB 被首先传输的位传输模式相同的用于长度大于 8 位的参数的位传输模式。MDDI_Data0 讯号路径上的数据与 I 型、II 型、III 型或 IV 型模式中的任意模式下的接口上所传输的字节位 '0' 对准。

[0187] 当在操控用于显示器的数据时，通过先行后列的方式传输用于像素阵列的数据，如在电子技术中习惯的一样。换句话说，出现在位图中同一行中的所有像素被以最左端像素首先传输而最右端像素最后传输的方式按顺序传输。在一行的最右端像素被传输后，序列中的下一个像素是下一行的最左端像素。对于多数显示器来说，像素的行一般以从顶至底的顺序传输，尽管按需要也可包含其它配置。此外，在处理位图时，常规方法（此处亦遵循）是通过把位图的左上角标示为位置或偏移 "0,0" 来定义一个参考点。随着位图中的位置分别趋近于位图的右端和底端，用于定义或确定所述位置的 X 和 Y 座标的值增加。第一行和第一列以指数值零开始。

[0188] C. 封包定义

[0189] 1. 子帧标头封包

[0190] 子帧标头封包是每一子帧的第一封包，且具有如图 8 所示的基本结构。如图 8 中所示，这一类型的封包被构造成具有封包长度域、封包类型域、唯一字域、子帧长度域、协议版本域、子帧计数域和媒体帧计数域，其一般按所列顺序。这一类型的封包一般被识别为 255 型 (0xff 十六进制) 封包，且使用预先选定的 17 字节的固定长度。

[0191] 封包类型域使用一字节值，而唯一字域使用 3 字节值。这两个域的 4 字节组合共同形成一个具有良好自相关的 32 位唯一字。在一个例示性实施例中，实际的唯一字是 0x005a3bff，其中较低的 8 位首先作为封包类型传输，且最高有效的 24 位被随后传输。

[0192] 子帧长度域包含规定每子帧的字节数的 4 字节信息。可将这个域的长度设定为零，以指示在链路被关闭到空闲状态前主机仅会传输一个子帧。这个域中的值可在从一个子帧过渡到下一子帧时被即时地动态改变。此能力适用于在同步脉冲中进行微小时序调整，以适应同步数据流。如果子帧标头封包的 CRC 无效，那么链路控制器应使用前一已知为好的子帧标头封包的子帧长度来估计当前子帧的长度。

[0193] 协议版本域包含规定主机所使用的协议版本的 2 个字节。将协议版本域设定为 '0' 来把协议的第一或当前版本规定为正在使用的。随着时间流逝当产生了新版本时，此值将改变。子帧计数域包含 2 字节，其规定了一个序号，所述序号指示从媒体帧的开始起已传输的子帧数。媒体帧的第一子帧具有为零的子帧计数。媒体帧的最后一个子帧具有为 n-1 的值，其中 n 为每媒体帧的子帧数。请注意，如果子帧长度设定为零（指示非周期性子帧），

那么子帧计数必须也设定为零。

[0194] 媒体帧计数域包含 3 字节,其规定了一个序号,所述序号指示从传送当前媒体项目或数据的开始起已传输的媒体帧的数目。媒体项目的第一媒体帧具有为零的媒体帧计数。在每一媒体帧的第一子帧之前,媒体帧计数加 1,且在使用了最大媒体帧计数(例如,媒体帧数 $224-1 = 16,777,215$)后,媒体帧计数返回到零。媒体帧计数值一般可在任何时候由主机复位以适应最终应用的需要。

[0195] 2. 填充符封包

[0196] 填充符封包是当在正向抑或反向链路上没有其它信息可发送时传送到或传送自客户端装置的封包。建议填充符封包具有一个最小长度,以便在需要时发送其它封包时具有最大的灵活性。在子帧或反向链路封装封包(参见下文)的末尾处,链路控制器设定填充符封包的大小以填满剩余空间,以便保持封包完整性。

[0197] 填充符封包的格式和内容展示于图 9 中。如图 9 中所示,这一类型的封包被构造成具有封包长度域、封包类型域、填充符字节域和 CRC 域。此类型的封包一般被识别为 0 型,并指示于 1 字节类型域中。填充符字节域中的位或字节包含可变数目的全零位值,以允许填充符封包为所要长度。最小的填充符封包在这个域中不包含字节。也就是说,所述封包仅由封包长度、封包类型和 CRC 组成,且使用预先选定的 3 字节固定长度。

[0198] 3. 视频流封包

[0199] 视频流封包承载视频数据以更新显示装置的通常为矩形的区域。此区域的大小可小至单一像素或大至整个显示器。可能存在同时显示的几乎无限数目的流,其由系统资源限制,因为显示一个流所需的所有背景(context)都包含于视频流封包中。视频流封包的格式(视频数据格式描述符)展示在图 10 中。如图 10 中所示,这一类型的封包被构造成具有封包长度(2 字节)域、封包类型域、视频数据描述符域、显示属性域、X 左边缘域、Y 顶边缘域、X 右边缘域、Y 底边缘域、X 和 Y 起始域、像素计数域、参数 CRC 域、像素数据域和 CRC 域。此类封包一般被识别为 1 型,并指示于 1 字节类型域中。

[0200] 上文所论述的共同帧概念是一种最小化音频缓冲区大小并减小等待时间的有效方式。但是,对于视频数据,可能有必要将一个视频帧的像素分散在一媒体帧中的多个视频流封包中。同样很有可能在单个视频流封包中的像素并非恰好对应于显示器上的一个完美矩形窗口。对于每秒 30 帧的例示性视频帧速率来说,每秒有 300 个子帧,从而导致每媒体帧 10 个子帧。如果在每一帧中有 480 行像素,在每一子帧中的每一视频流封包将包含 48 行像素。在其它情况下,视频流封包不一定包含整数行像素。这对于其中每媒体帧的子帧数不能均匀地除每视频帧的行(也称为视频线)数目的其它视频帧尺寸来说是正确的。尽管视频流封包不一定包含整数行像素,但每一视频流封包一般必须包含整数个像素。这在像素各具有一个以上字节或其处于如图 12 所示的压缩格式中时是重要的。

[0201] 图 11a-11d 中展示了用于实现如上文所述的例示性视频数据描述符域的操作所采用的格式和内容。在图 11a-11d 中,视频数据格式描述符域包含形式为 16 位无符号整数的 2 字节,所述整数规定了当前封包中的当前流中的像素数据(Pixel Data)中的每一像素的格式。不同视频流封包可能使用不同像素数据格式,也就是说在视频数据格式描述符中使用不同的值,且类似地,一个流(显示器的区域)可即时改变其数据格式。视频数据格式描述符仅定义了用于当前封包的像素格式,其并不意味着某个不变的格式将持续用于特

定视频流的整个存在期中。

[0202] 图 11a 至 11d 说明了如何编码视频数据格式描述符。如在这些图中所用,且在此实施例中,当如图 11a 所示位 [15:13] 等于 ‘000’ 时,视频数据就由单色像素的阵列组成,其中每像素的位数由视频数据格式描述符字的 3 到 0 位定义。在此情况下 11 至 4 位设定为零。当位 [15:13] 等于 ‘001’ 时,如图 11b 所示,则视频数据由彩色像素阵列组成,所述像素各通过一个色彩映射规定一种色彩。在这种情况下,视频数据格式描述符字的 5 至 0 位定义每像素的位数目,且 11 至 6 位被设定为零。当位 [15:13] 等于 ‘010’ 时,如图 11c 所示,则视频数据由彩色像素阵列组成,其中每像素红色的位数目由 11 至 8 位定义,每像素绿色的位数目由 7 至 4 位定义,且每像素蓝色的位数目由 3 至 0 位定义。在此情况下,每一像素中的位的总数是用于红、绿、蓝的位的数目的总和。

[0203] 但是,当位 [15:13] 等于 ‘011’ 时,如图 11d 所示,则视频数据由具有亮度和色度信息的为 4:2:2 格式的视频数据阵列组成,其中每像素亮度 (Y) 的位数目由 11 至 8 位定义,Cr 分量的位数目由 7 至 4 位定义,且 Cb 分量的位数目由 3 至 0 位定义。每一像素中的位的总数是用于红、绿和蓝的位的数目的总和。以发送 Y 的速率的一半的速率发送 Cr 和 Cb 分量。此外,在此封包的像素数据部分中的视频样本被组织为如下:Y_n、Cr_n、Cb_n、Y_{n+1}、Y_{n+2}、Cr_{n+2}、Cb_{n+2}、Y_{n+3}、……,其中 Cr_n 和 Cb_n 与 Y_n 和 Y_{n+1} 相关联,且 Cr_{n+2} 和 Cb_{n+2} 与 Y_{n+2} 和 Y_{n+3} 相关联,等等。如果在当前流中的一行中存在奇数个像素 (X 右边缘 -X 左边缘 +1),则对应于每一行中最后一个像素的 Cb 值之后将是下一行的第一个像素的 Y 值。

[0204] 对于图中所示的所有四个格式来说,标示为“P”的 12 位规定了像素数据样本是否是压缩的或字节对准的像素数据。在这个域中的值 ‘0’ 指示在所述像素数据域中的每一像素和每一像素中的每一色彩是与一个 MDD 接口字节边界 (byte boundary) 字节对准的。值 ‘1’ 指示像素数据域中的每一像素和每一像素内的每一色彩是与前一像素或一像素内的色彩压缩到一起的,从而消除未使用的位。

[0205] 用于特定显示窗口的媒体帧的第一视频流封包中的第一像素将前往所述流窗口的由 X 左边缘和 Y 顶边缘定义的左上角,且接收到的下一个像素被置于相同行中的下一个像素位置,等等。在媒体帧的这个第一封包中,X 起始值通常将等于 X 左边缘,且 Y 起始值通常将等于 Y 顶边缘。在对应于相同屏幕窗口的随后封包中,X 和 Y 起始值通常将设定到屏幕窗口中通常紧接着在前一子帧中传输的视频流封包中所发送的上一像素的像素位置。

[0206] 4. 音频流封包

[0207] 音频流封包承载了将通过显示器的音频系统播发或用于独立的音频呈现装置的音频数据。不同音频数据流可被分配用于一个音响系统中的分开的音频通道,例如:取决于所使用的音频系统的类型,为左前、右前、中央、左后和右后通道。为含有增强的空间声学讯号处理的头戴受话器提供了一整套音频通道。图 13 中说明了音频流封包的格式。如图 13 中所示,这一类型的封包被构造成具有封包长度域、封包类型域、音频通道 ID 域、音频采样计数域、每样本位和压缩域、音频采样率域、参数 CRC 域、数字音频数据域和音频数据 CRC 域。在一个实施例中,此类型封包一般被识别为 2 型封包。

[0208] 每样本位和压缩域包含形式为 8 位无符号整数的 1 字节,其规定了音频数据的压缩格式。一般采用的格式是 4 至 0 位定义每 PCM 音频样本的位数目。接着 5 位规定是否压缩数字音频数据样本。图 14 中说明了压缩的音频样本和字节对准的音频样本之间的差异。

值‘0’指示数字音频数据域中的每一 PCM 音频样本与一个 MDDI 接口字节边界字节对准,且值‘1’指示每一连续的 PCM 音频样本被相对于前一音频样本压缩。这个位仅在 4 至 0 位中定义的值(每 PCM 音频样本的位数目)不为八的倍数时是有效的。7 至 6 位被保留用于未来的用途且一般设定为零值。

[0209] 5. 保留流封包

[0210] 在一个实施例中,保留了封包类型 3 至 55,以用于将按会遇到的各种应用的需要被定义用于封包协议的未版本或变体的流封包。再一次,这是面临不断变化的技术和系统设计与其他技术相比较来说使所述 MDD 接口更灵活且更有用的方面。

[0211] 6. 用户定义流封包

[0212] 称为 56 至 63 型的八个数据流类型被保留用于可由设备制造商定义以配合 MDDI 链路使用的专用应用。这些数据流类型称为用户定义流封包。视频流封包承载视频数据以更新(或不更新)显示器的一个矩形区域。用于这些封包类型的流参数和数据要留给寻求使用它们的特定设备制造商来定义。图 15 中说明了用户定义流封包的格式。如图 15 中所示,这类封包被构造成具有封包长度域(2 字节)、封包类型域、流 ID 号域、流参数域、参数 CRC 域、流数据域和流数据 CRC 域。

[0213] 7. 色彩映射封包

[0214] 色彩映射封包规定了用以呈现用于显示器的色彩的色彩映射查找表的内容。某些应用可能会要求比在单一封包中可传输的数据量更大的色彩映射。在这些情况下,可传送多个色彩映射封包,通过使用下文所述的偏移和长度域,每一个封包具有色彩映射的一个不同子集。图 16 中说明了色彩映射封包的格式。如图 16 中所示,这一类型的封包被构造成具有封包长度域、封包类型域、色彩映射数据大小域、色彩映射偏移域、参数 CRC 域、色彩映射数据域、和数据 CRC 域。此类封包一般被识别为 64 型封包。

[0215] 8. 反向链路封装封包

[0216] 在一个例示性实施例中,使用反向链路封装封包在反向上传送数据。发送一个正向链路封包,且在这一封包的中途改变或逆转 MDDI 链路操作(传送方向),使得可在反向上传送封包。图 17 中说明了反向链路封装封包的格式。如图 17 中所示,这一类型的封包被构造成具有封包长度域、封包类型域、反向链路旗标域、逆转长度域、参数 CRC 域、逆转 1 域、反向数据封包域、及逆转 2 域。此类型的封包一般被识别为 65 型封包。

[0217] MDDI 链路控制器在发送反向链路封装封包的同时以一种特殊方式动作。MDD 接口具有一个一直由主机驱动的选通讯号。主机以似乎其正在针对反向链路封装封包的逆转和反向数据封包的每一位传输一个零的方式动作。主机在所述两个逆转时间期间及在分配用于反向数据封包的时间期间于每一个位界(bit boundary)处双态触发一个 MDDI_Strobe 讯号(这是与其传输全零数据时相同的动作)。主机在由逆转 1 域规定的时间周期期间禁用其 MDDI 数据讯号线驱动器,且客户端在由逆转 2 域规定的时间周期之后的驱动器重启用时域期间重启用其线驱动器。显示器读取逆转长度参数并在逆转 1 域中的最后一个位后立刻将数据讯号驱动向主机。显示器使用封包长度和逆转长度参数来获知其所具有的用以将封包发送到主机的时间长度。当客户端没有要发送到主机的数据时,客户端可发送填充封包或将数据线驱动到零状态。如果数据线被驱动到零状态,主机就将这种情况解释为零长度(不是有效长度)封包,且主机在当前反向链路封装封包的持续时间中不接受任何更

多的来自客户端的封包。

[0218] 客户端显示器在逆转 2 域开始前在至少一个反向链路时钟周期中将 MDDI 数据线驱动到零电平。这将所述数据线在逆转 2 时间周期期间保持在确定状态中。如果客户端没有更多要发送的封包,其甚至可在将数据线驱动到零电平后禁用所述数据线,因为休眠偏压电阻器(在文中其他地方论述)会在反向数据封包域的剩余部分中将数据线保持在零电平。

[0219] 显示器请求和状态封包的反向链路请求域可用于通知主机在反向链路封装封包中显示器需要用以将数据发送回到主机的字节数。主机尝试通过在反向链路封装封包中分配至少 此数目的字节来批准所述请求。主机可在一个子帧中发送多于一个的反向链路封装封包。显示器可在几乎任何时候发送显示器请求和状态封包,且主机将把反向链路请求参数解释为在一个子帧中请求的字节总数。

[0220] 9. 显示器能力封包

[0221] 主机需要知晓其正与之通讯的显示器(客户端)的能力,以便以一种大体上最优或理想的方式配置主机至显示器链路。建议在获得正向链路同步后显示器才发送显示器能力封包到主机。此封包的传输在主机使用反向链路封装封包中的反向链路旗标进行请求后被视为必需的。图 18 中说明了显示器能力封包的格式。如图 18 中所示,这一类型的封包被构造成具有封包长度域、封包类型域、协议版本域、最小协议版本域、位图宽度域、位图高度域、单色能力域、色彩映射能力域、RGB 能力域、Y Cr Cb 能力域、显示特性能力域、数据率能力域、帧速率能力域、音频缓冲深度域、音频流能力域、音频速率能力域、最小子帧速率域和 CRC 域。在一个例示性实施例中,此类型的封包一般被识别为 66 型封包。

[0222] 10. 键盘数据封包

[0223] 键盘数据封包用于将键盘数据从客户端装置发送到主机。无线(或有线)键盘可结合各种显示器或音频装置使用,包括(但不限于)头戴式视频显示器/音频呈现装置。键盘数据封包将从若干已知的类似键盘的装置之一接收的键盘数据中继到主机。这种封包也可用于正向链路上以将数据发送到键盘。键盘数据封包的格式展示在图 19 中,且包含来自键盘或用于键盘的字节数可变的信息。如图 19 所示,这一类型的封包被构造成具有封包长度域、封包类型域、键盘数据域和 CRC 域。在此处,此类型的封包一般被识别为 67 型封包。

[0224] 11. 指向装置数据封包

[0225] 指向装置数据封包用于将来自无线鼠标或其他指向装置的在显示器上的位置信息发送到主机。也可用这种封包在正向链路上将数据发送到指向装置。指向装置数据封包的例示性格式展示在图 20 中,且包含来自指向装置或用于指向装置的字节数可变的信息。如图 20 中所示,这一类型的封包被构造成具有封包长度域、封包类型域、指向装置数据域和 CRC 域。在一个例示性实施例中,此类型封包在 1 字节类型域中一般被识别为 68 型封包。

[0226] 12. 链路关闭封包

[0227] 将链路关闭封包从主机发送到客户端显示器以指示 MDDI 数据和选通脉冲将被关闭且进入低功耗的“休眠”状态。这种封包可在静态位图被从移动通讯装置发送到显示器后,或者当暂时没有进一步的信息要从主机传送到客户端时用于关闭链路并节省功率。当主机 再次发送封包时恢复正常操作。在休眠状态后发送的第一个封包是子帧标头封包。

图 21 中展示了显示状态封包的格式。如图 21 中所示,这一类型的封包被构造成具有封包长度域、封包类型域和 CRC 域。在一个实施例中,此类型封包在 1 字节类型域中一般被识别为 69 型封包,且使用预先选定的 3 字节固定长度。

[0228] 在低功率休眠状态中,MDDI_Data 驱动器被禁用而处于高阻抗状态,且通过使用可由显示器过驱动的高阻抗偏压网络而将 MDDI_Data 讯号拉到逻辑零状态。在休眠状态中接口使用的选通讯号被设定到逻辑零电平,从而最小化功耗。如本文其它地方所描述,主机或客户端都可将 MDDI 链路从休眠状态唤醒,这是本发明的一个关键进步和优点。

[0229] 13. 显示器请求和状态封包

[0230] 主机需要来自显示器的少量信息以便能以大体上最优的方式配置主机至显示器链路。建议显示器在每一子帧中将一个显示状态封包发送到主机。显示器应将此封包作为反向链路封装封包中的第一个封包来发送,以确保其被可靠地传送到主机。图 22 中展示了显示状态封包的格式。如图 22 所示,这一类型的封包被构造成具有封包长度域、封包类型域、反向链路请求域、CRC 错误计数域和 CRC 域。此类型封包在 1 字节类型域中一般被识别为 70 型封包,且使用预先选定的 8 字节固定长度。

[0231] 反向链路请求域可用于通知主机在反向链路封装封包中显示器需要用以将数据发送回到主机的字节数。主机应尝试通过在反向链路封装封包中分配至少此数目的字节来批准所述请求。主机可在一个子帧中发送多于一个的反向链路封装封包以便容纳数据。客户端可在任何时候发送显示器请求和状态封包,且主机将把反向链路请求参数解释为在一个子帧中请求的字节总数。下文展示如何将反向链路数据发送回到主机的额外细节和具体实例。

[0232] 14. 位块传送封包

[0233] 位块传送封包提供了一种在任意方向上滚动显示区域的手段。具有这种能力的显示器将在显示器能力封包的显示特性能力指示符域的 0 位中记录该能力。图 23 中展示了位块传送封包的格式。如图 23 中所示,这一类型的封包被构造成具有封包长度域、封包类型域、左上部 X 值域、左上部 Y 值域、窗口宽度域、窗口高度域、窗口 X 移动域、窗口 Y 移动域和 CRC 域。此类型封包一般被识别为 71 型封包,且使用预先选定的 15 字节固定长度。

[0234] 所述域分别用于规定待移动的窗口的左上角的座标的 X 和 Y 值、待移动的窗口的宽度和高度、以及水平及垂直移动窗口的像素数。后两个域的正值分别导致窗口向右和向下移动,而负值分别导致向左和向上移动。

[0235] 15. 位图区域填充封包

[0236] 位图区域填充封包提供了一种容易地将显示器的一个区域初始化为单一色彩的手段。具有这种能力的显示器将在显示器能力封包的显示特性能力指示符域的 1 位中记录该能力。图 24 中展示了位图区域填充封包的格式。如图 24 所示,这一类型的封包被构造成具有封包长度域、封包类型域、左上部 X 值域、左上部 Y 值域、窗口宽度域、窗口高度域、数据格式描述符域、像素区域填充值域和 CRC 域。此类型封包在 1 字节类型域中一般被识别为 72 型封包,且使用预先选定的 17 字节的固定长度。

[0237] 16. 位图图案填充封包

[0238] 位图图案填充封包提供了一种容易地将显示器的一个区域初始化为一个预先选定的图案的手段。具有这种能力的显示器将在显示器能力封包的显示特性能力指示符域的

2 位中记录该能力。填充图案的左上角与待填充的窗口的左上角对准。如果待填充窗口比填充图案更宽或更高,则所述图案可被水平或垂直地重复若干次以填满所述窗口。按需要截除最后一个重复的图案的右端或底端。如果所述窗口小于填充图案,则可截除填充图案的右边或底端以配合所述窗口。

[0239] 图 25 中展示了位图图案填充封包的格式。如图 25 所示,这一类型的封包被构造成具有封包长度域、封包类型域、左上部 X 值域、左上部 Y 值域、窗口宽度域、窗口高度域、图案宽度域、图案高度域、数据格式描述符域、参数 CRC 域、图案像素数据域、和像素数据 CRC 域。此类型封包在 1 字节类型域中一般被识别为 73 型封包。

[0240] 17. 通讯链路数据通道封包

[0241] 通讯链路数据通道封包为具有高水平计算能力的显示器(诸如 PDA)提供了一种与诸如手机或无线数据端口装置的无线收发器通讯的手段。在这种情况下,MDDI 链路充当介于通讯装置和具有移动显示器的计算装置之间的方便的高速接口,其中此封包在用于所述装置的操作系统的通讯链路层上传输数据。例如,如果网络浏览器、电子邮件客户端或整个 PDA 被内建在移动显示器中,则可使用此种封包。具有这种能力的显示器将在显示器能力封包的显示特性能力指示符域的 3 位中记录该能力。

[0242] 图 26 中展示了通讯链路数据通道封包的格式。如图 26 所示,这一类型的封包被构造成具有封包长度域、封包类型域、参数 CRC 域、通讯链路数据域和通讯数据 CRC 域。此类型封包在所述类型域中一般被识别为 74 型封包。

[0243] 18. 接口类型转递请求封包

[0244] 接口类型转递请求封包使得主机可请求客户端或显示器从现有或当前模式转移到 I 型(串行)模式、II 型(2 位并行)模式、III 型(4 位并行)模式或 IV 型(8 位并行)模式。在主机请求一个特定模式之前,其应通过检查显示器能力封包的显示特性能力指示符域的 6 位和 7 位来确认显示器能够在所要模式下操作。图 27 中展示了接口类型转递请求封包的格式。如图 27 所示,这一类型的封包被构造成具有封包长度域、封包类型域、接口类型域和 CRC 域。此类型封包一般被识别为 75 型封包,且使用预先选定的 4 字节的固定长度。

[0245] 19. 接口类型确认封包

[0246] 接口类型确认封包由显示器发送以确认收到接口类型转递封包。所请求的模式(I 型(串行)模式、II 型(2 位并行)模式、III 型(4 位并行)模式或 IV 型(8 位并行)模式)被作为此封包中的一个参数返回给主机。图 28 中展示了接口类型确认封包的格式。如图 28 所示,这一类型的封包被构造成具有封包长度域、封包类型域、接口类型域和 CRC 域。此类型封包一般被识别为 76 型封包,且使用预先选定的 4 字节的固定长度。

[0247] 20. 执行类型转递封包

[0248] 执行类型转递封包是一种供主机命令显示器转递到这种封包中所规定的模式中的手段。所述模式将是先前通过接口类型转递请求封包和接口类型确认封包予以请求和确认过的相同模式。在发送了此封包后,主机和显示器应切换到已达成一致的模式。显示器可能在模式改变期间丢失并重获链路同步性。图 29 中展示了执行类型转递封包的格式。如图 29 所示,这一类型的封包被构造成具有封包长度域、封包类型域、封包类型域和 CRC 域。此类型封包在 1 字节类型域中一般被识别为 77 型封包,且使用预先选定的 4 字节的固定长

度。

[0249] 21. 正向音频通道启用封包

[0250] 这种封包允许主机启用或禁用显示器中的音频通道。此能力的用途在于当主机不输出音频时显示器（客户端）可关闭音频放大器或类似电路元件以节省功率。这在只将音频流存在与否用作指示物的情况下是隐含地更难以实施的。当显示系统加电后的缺省状态是所有音频通道都被启用。图 30 中展示了正向音频通道启用封包的格式。如图 30 所示，这一类型的封包被构造成具有封包长度域、封包类型域、音频通道启用屏蔽域和 CRC 域。此类型封包在 1 字节类型域中一般被识别为 78 型封包，且使用预先选定的 4 字节的固定长度。

[0251] 22. 反向音频采样率封包

[0252] 这种封包允许主机启用或禁用反向链路音频通道，且设定这个流的音频数据采样率。主机选择一个在显示器能力封包中被定义为有效的采样率。如果主机选择了无效的采样率，那么显示器将不会把音频流发送给主机。主机可通过将采样率设定为 255 来禁用反向链路音频流。当显示系统最初加电或连接时所采取的缺省状态是反向链路音频流被禁用。图 31 中展示了反向音频采样率封包的格式。如图 31 所示，这一类型的封包被构造成具有封包长度域、封包类型域、音频采样率域和 CRC 域。此类型封包一般被识别为 79 型封包，且使用预先选定的 4 字节的固定长度。

[0253] 23. 数字内容保护开销封包

[0254] 这种封包允许主机和显示器互换与所使用的数字内容保护方法相关的消息。现在设想了两类内容保护，即数字传输内容保护 (Digital Transmission Content Protection) (DTCP)、或高带宽数字内容保护系统 (High-bandwidth Digital Content Protection System) (HDCP)，同时为未来的替代性保护方案标志保留了空间。所使用的方法由这种封包中的内容保护类型参数来规定。图 32 中展示了数字内容保护开销封包的格式。如图 32 所示，这一类型的封包被构造成具有封包长度域、封包类型域、内容保护类型域、内容保护开销消息域和 CRC 域。此类型封包一般被识别为 80 型封包。

[0255] 24. 透明色彩启用封包

[0256] 透明色彩启用封包用于规定在显示器中哪个色彩是透明的，且用于启用或禁用将透明色彩用于显示影像。具有此能力的显示器将在显示器能力封包的显示特性能力指示符域的 4 位中记录该能力。当具有透明色彩的值的像素被写入位图时，色彩不会从先前值变化。图 33 中展示了透明色彩启用封包的格式。如图 33 所示，这一类型的封包被构造成具有封包长度域、封包类型域、透明色彩启用域、数据格式描述符域、透明像素值域和 CRC 域。此类型封包在 1 字节类型域中一般被识别为 81 型封包，且使用预先选定的 10 字节的固定长度。

[0257] 25. 往返行程延迟测量封包

[0258] 往返行程延迟测量封包用于测量从主机到客户端（显示器）的传播延迟与从客户端（显示器）回到主机的延迟的总和。这一测量固有地包含存在于线驱动器和接收器及互连子系统中的延迟。此测量用于设定上文概述的反向链路封装封包中的逆转延迟和反向链路速率除数参数。这种封包在 MDDI 链路运行在希望一个特定应用达到的最大速度下时最为有用。MDDI_Stb 讯号的动作如同在以下时域期间发送全零数据：全零 (All Zero)、两个保护时间、及测量周期。这导致 MDDI_Stb 以数据率一半的速率双态触发，所以其可在测量

周期期间 在显示器中用作周期时钟。

[0259] 图 34 中展示了往返行程延迟测量封包的格式。如图 34 所示,这一类型的封包被构造成具有封包长度域、封包类型域、参数 CRC 域、全零域、保护时间 1 域、测量周期域、保护时间 2 域和驱动器重启用域。此类型封包一般被识别为 82 型封包,且使用预先选定的 533 位的固定长度。

[0260] 图 35 中说明了在往返行程延迟测量封包期间发生的事件时序。在图 35 中,主机传输往返行程延迟测量封包,其以继之以全零域和保护时间 1 域的参数 CRC 域和选通对准域来展示。在封包到达客户端显示装置或处理电路之前发生延迟 3502。当显示器接收所述封包时,其在由显示器确定的测量周期开始处尽可能精确地传输 0xff、0xff、0x0 模式。从主机的观点来看,显示器开始传输此序列的实际时间是延迟于测量周期的开始处的。这个延迟的量大体上是封包传播经过线驱动器和接收器及互连子系统所花费的时间。所述模式从显示器传播回到主机会发生类似量的延迟 3504。

[0261] 为了准确确定讯号传送到并传送自客户端的往返行程延迟时间,主机计数在测量周期开始后直到讯号到达时侦测到 0xff、0xff、0x0 序列的开始为止所发生的位时间周期的数目。此信息用于确定往返行程讯号从主机传递到客户端并返回所用的时间量。接着,将这个量的大约一半归因于讯号到客户端的单程传送所产生的延迟。

[0262] 显示器大体上在发送了 0xff、0xff、0x0 模式的最后一个位后立刻禁用其线驱动器。保护时间 2 为显示器的线驱动器提供了在主机传输下一封包的封包长度之前完全进入高阻抗状态的时间。休眠上拉电阻和休眠下拉电阻(见图 42)确保了在主机和显示器中线驱动器被禁用的时间间隔中,MDDI_Data 讯号保持在有效低电平。

[0263] 26. 正向链路偏斜校正封包

[0264] 正向链路偏斜校正封包允许客户端或显示器针对相对于 MDDI_Stb 讯号的 MDDI_Data 讯号的传播延迟方面的差异而校正客户端或显示器本身。在没有延迟偏斜补偿的情况下,最大数据率一般被限制于顾及这些延迟中的潜在最差情况偏差。一般来说,仅在正向链路数据率被设置成约 50Mbps 或更低的速率时才发送这种封包。在发送这一封包来校正显示器后,数据率可被提升到 50Mbps 以上。如果在偏斜校正过程期间数据率设定得太高,显示器可能同步到位周期的假信号(alias),其可导致延迟偏斜补偿设定偏移大于一个位时间的量,从而导致错误的的数据定时。在发送正向链路偏斜校正封包之前选择数据率最高的接口类型或最大可能的接口类型,以便可校正所有现有的数据位。

[0265] 图 56 中展示了正向链路偏斜校正封包的格式。如图 56 所示,这一类型的封包被构造成具有封包长度域(2 字节)、封包类型域、参数 CRC 域、校正数据序列域和 CRC 域。此类型封包在类型域中一般被识别为 83 型封包,且具有预先选定的为 515 的长度。

[0266] D. 封包 CRC

[0267] CRC 域出现在封包的末端,且有时出现在封包中的可能具有相当大的数据域并因此在此在传送期间发生错误的的可能性较大的某些较关键参数之后。在具有两个 CRC 域的封包中,CRC 产生器(当仅使用一个时)在第一个 CRC 后重新初始化,以便在一个长数据域后的 CRC 计算不受封包开始处的参数的影响。

[0268] 在一个例示性实施例中,用于 CRC 计算的多项式称为 CRC-16,或 $X^{16}+X^{15}+X^2+X^0$ 。图 36 中展示了适用于实施本发明的 CRC 产生器和校验器 3600 的样本实施。在图 36 中,在

传送一个封包（在 Tx_MDDI_Data_Before_CRC 线上输入）的第一位之前，CRC 寄存器 3602 被初始化到值 0x0001，接着将所述封包的字节移位（从 LSB 开始）到寄存器中。注意在此图中的寄存器位数对应于所使用的多项式的次数，而不是 MDDI 使用的位位置（bitposition）。以单向来移位 CRC 寄存器更有效率，且这导致 CRC 位 15 出现在 MDDI CRC 域的位位置 0 中，且 CRC 寄存器位 14 出现在 MDDI CRC 域的位位置 1 中，以及等等直到达到 MDDI 位位置 14。

[0269] 举例来说，如果显示器请求和状态封包的封包内容为 :0x07、0x46、0x000400、0x00（或以字节序列表示为 :0x07、0x00、0x46、0x00、0x04、0x00、0x00），且使用多工器 3604 和 3606 及与非门 3608 的输入来提供（submit），则在 Tx_MDDI_Data_With_CRC 线上所得的 CRC 输出为 0x0eal（或表示为序列 0xa1、0x0e）。

[0270] 当 CRC 产生器和校验器 3600 被配置为 CRC 校验器时，在 Rx_MDDI_Data 线上接收到的 CRC 被输入到多工器 3604 和与非门 3608，并通过或非门 3610、异或（XOR）门 3612 和与门 3614 而与 CRC 寄存器中的值逐位比较。如果有任何错误（如与门 3614 所输出），则通过将门 3614 的输出连接到寄存器 3602 的输入来针对每个含有 CRC 错误的封包而将 CRC 加一。应注意图 36 中所示的实例性电路可在给定 CHECK_CRC_NOW 窗口内输出一个以上 CRC 错误讯号（见图 37B）。因此，CRC 错误计数器一般仅计数 CHECK_CRC_NOW 有效的每一时间间隔内的第一个 CRC 错误实例。如果配置为 CRC 产生器，则在与封包末尾一致的時刻将 CRC 从 CRC 寄存器时钟输出。

[0271] 在图 37A 和 37B 中图形地说明了输入和输出讯号及启用讯号的时序。通过 Gen_Reset、Check_CRC_Now、Generate_CRC_Now 及 Sending_MDDI_Data 讯号连同 Tx_MDDI_Data_Before_CRC 讯号和 Tx_MDDI_Data_With_CRC 讯号的状态（0 或 1），图 37A 中展示了 CRC 的产生和数据封包的传输。通过 Gen_Reset、Check_CRC_Now、Generate_CRC_Now 及 Sending_MDDI_Data 讯号连同 Rx_MDDI_Data 讯号和 CRC 错误讯号的状态，图 37B 中展示了数据封包的接收和 CRC 值的校验。

[0272] E. 用于封包 CRC 的错误码重载

[0273] 在主机和客户端之间只传送数据封包和 CRC 时，不包含错误码。唯一的误差是同步的损失。否则，不得不等候链路由于缺乏良好的数据传送路径或管线而超时且接着复位所述链路并继续。不幸地是，这相当耗时且效率较低。

[0274] 为用于一个实施例中，已发展了一种新技术，其中封包的 CRC 部分被用于传送错误码信息。这被一般性地展示于图 65 中，且更详细地展示于图 66A、66B 和 67 中。也就是说，处理器或处理数据传送的装置产生一个或一个以上错误码，所述错误码指示了可能在通讯处理或链路中发生的特定预定义错误或缺陷。当遇到错误时，产生适当的错误码并使用封包的 CRC 的位来传送。也就是说，用所要的错误码或某个其他预先选定的值来重载（或重写）所述 CRC 值以表示存在错误，其可由一个监视 CRC 域的值错误监视器或校验器在接受端予以侦测。对于由于某种原因而使错误码与 CRC 值匹配的情况，传送错误的补码（compliment）来防止混淆。

[0275] 在一个实施例中，为提供一种健壮的（robust）错误警告和侦测系统，可使用在侦测到错误后传送或发送的一系列（一般为所有）封包将错误码传送若干次。这种情况持续直到产生所述错误的条件已从系统中清除为止，此时传送常规的 CRC 位而不由另一值重载。

[0276] 这种重载 CRC 值的技术提供了对系统错误的快得多的响应同时使用了最小量的额外位或域。

[0277] 如图 66 所示,展示了 CRC 重写机构或装置 6600,其使用一个错误侦测器或侦测构件 6602,所述错误侦测器或侦测构件 6602 可形成先前所述或已知的其他电路的部分,侦测通讯链路或过程中错误的出现或存在。可形成为其他电路的部分或使用诸如查找表的技术来存储预先选定的错误消息的错误码产生器或构件 6604 产生一个或一个以上错误码来指示已侦测到的特定的预定义错误或缺陷。容易理解的是,装置 6602 和 6604 可形成为所要的单一电路或装置,或形成为用于其他已知处理器和元件的编程的步骤顺序的部分。

[0278] 展示了 CRC 值比较器或比较构件 6606,其用以校验所选定的一个或一个以上错误码 是否与所传送的 CRC 值相同。如果相同,则将补码产生器或产生构件或装置用于提供错误码的补码,以便不会被误认为原始的 CRC 模式或值及使侦测机制混乱或复杂化。错误码选择器或选择构件元件或装置 6610 接着选择其需要插入或重写的错误码或值、或其相应的适当补码。错误码 CRC 重写器或重写机构或构件 6612 是一种接收数据流、封包和待插入的所要码,并重写对应或适当的 CRC 值,以便传送所要的错误码到接收装置的装置。

[0279] 如所提及,可使用一系列封包将错误码传送若干次,所以重写器 6612 可利用存储器存储元件,从而在处理期间保持所述码的复本,或从先前元件或可用于按需要或理想地存储或保持这些码的值的其他已知存储位置再调用这些码。

[0280] 图 67A 和 67B 更详细地展示了图 66 的重写机构所实施的一般处理。在图 67A 中,在步骤 6702 中侦测通讯数据或过程中的一个或一个以上错误,且在步骤 6704 中选择错误码来指示此情况。同时或在适当的时间,在步骤 6706 中校验将被替换的 CRC 值,且在步骤 6708 中将其与所要的错误码相比较。如上文所论述,此比较的结果是确定所要的码或其它代表性值是否会与现有的 CRC 值相同的依据。如果相同,那么处理继续到步骤 6712,在此步骤中选择补码或在某些情况下按需要选择另一种代表性值作为待插入的码。一旦在步骤 6710 和 6714 中已确定将插入何种错误码或值,就选择适当的码用于插入。为了清晰的目的,这些步骤被单独说明,但它们大体上代表以步骤 6708 决策的输出为基础的单一选择。最后,在步骤 6716 中,在 CRC 位置重写适当的值,以与此过程所针对的封包一起传送。

[0281] 如图 67B 所示,在封包接收方,在步骤 6722 中监视封包 CRC 值。一般来说, CRC 值由系统内的一个或一个以上过程监视,以确定是否已发生数据传送错误,及是请求重新传输所述封包还是禁止进一步的操作等等,其中的某些在上文已论述。作为此监视过程的一部分,也可将信息用于将值与已知或预先选定的错误码或代表性值比较并侦测错误的存在。或者,可实施单独的错误侦测过程和监视器。如果看来似乎存在一个码,就在步骤 6724 中抽取或以其它所提及的方式获取其以用于进一步处理。可在步骤 6726 中确定它是实际的码还是补码(在补码的情况下,用一个额外的步骤 6728 来将所述值转译为所要的码值)。在任何一种状况下,接着在步骤 6730 中将所得的已抽取码、补码或其它已恢复的值用于侦测所传输的码中发生了何错误。

[0282] V. 从休眠状态的链路重新启动

[0283] 当主机从休眠状态重新启动正向链路时,其将 MDDI_Data 驱动到逻辑 1 状态并保持约 150 微秒,接着启动 MDDI_Stb 并同时 MDDI_Data 驱动到逻辑 0 状态并保持 50 微秒,且接着通过发送子帧标头封包来开始正向链路讯务。通过在讯号间提供足够的整

定时间 (settling time), 这大体上允许在发送子帧标头封包之前解决总线争用 (bus contention)。

[0284] 当客户端 (此处为显示器) 需要来自主机的数据或通讯时, 其将 MDDI_Data0 线驱动到逻辑 1 状态并保持约 70 微秒 (尽管按需要可使用其它时间段), 接着通过将驱动器置于高阻抗状态来禁用所述驱动器。此动作导致主机开始或重新启动正向链路 (208) 上的数据讯务并轮询客户端的状态。主机必须在 50 微秒内侦测请求脉冲的存在, 并接着开始启动序列, 即, 驱动 MDDI_Data0 到逻辑 1 并保持 150 微秒, 及驱动 MDDI_Data0 到逻辑 0 并保持 50 微秒。如果显示器侦测到 MDDI_Data0 处在逻辑 1 状态中超过 50 微秒, 那么它不应发送服务请求脉冲。下文进一步论述与休眠处理和启动序列相关的时间间隔的时间和容限的选择原理。

[0285] 图 38 说明了无争用的典型服务请求事件 3800 的处理步骤的实例, 在图示中为方便起见使用字母 A、B、C、D、E、F 和 G 将所述事件标号。过程开始于 A 点处, 此时主机将链路关闭封包发送到客户端装置以通知其链路将转换到低功率休眠状态。如点 B 处所示, 在下一步骤中, 主机通过禁用 MDDI_Data0 驱动器并将 MDDI_Stb 驱动器设定至逻辑 0 来进入低功率休眠状态。通过一个高阻抗偏压网络将 MDDI_Data0 驱动到零电平。如在 C 点所示, 在某段时间后, 客户端通过将 MDDI_Data0 驱动到逻辑 1 电平而发送服务请求脉冲到主机。通过使用高阻抗偏压网络, 主机仍确立零电平, 但客户端中的驱动器迫使所述线变为逻辑 1 电平。如 D 点所示, 在 50 微秒内, 主机识别出所述服务请求脉冲, 并通过启用其驱动器而在 MDDI_Data0 上确立逻辑 1 电平。如 E 点所示, 客户端接着停止尝试确立服务请求脉冲, 且客户端将其驱动器置于高阻抗状态。如 F 点所示, 主机驱动 MDDI_Data0 到逻辑 0 电平并保持 50 微秒, 且也开始以与 MDDI_Data0 上的逻辑 0 电平一致的方式产生 MDDI_Stb。如 G 点所示, 在将 MDDI_Data0 确立为零电平并驱动 MDDI_Stb 达 50 微秒后, 主机通过发送子帧标头封包而开始在正向链路上传输数据。

[0286] 图 39 中说明了一个类似实例, 其中在链路重新启动序列开始后确立服务请求, 且再次使用字母 A、B、C、D、E、F 和 G 来对事件标号。这代表了一种最差情况, 其中来自客户端的请求脉冲或讯号最接近于破坏子帧标头封包。过程开始于 A 点, 此时主机再次发送链路关闭封包到客户端装置, 以通知其链路将转换到低功率休眠状态。如点 B 处所示, 在下一步骤中, 主机通过禁用 MDDI_Data0 驱动器并将 MDDI_Stb 驱动器设定至逻辑 0 来进入低功率休眠状态。和上文一样, 通过一个高阻抗偏压网络将 MDDI_Data0 驱动到零电平。如在 C 点所示, 在某段时间后, 客户端通过将 MDDI_Data0 驱动到逻辑 1 电平并保持 150 微秒而开始链路重新启动序列。如 D 点所示, 在链路重新启动序列开始后经过 50 微秒之前, 显示器也在 70 微秒的持续时间内确立 MDDI_Data0。这样做的原因是显示器需要从主机请求服务且不会识别出主机已开始链路重新启动序列。如点 E 所示, 客户端接着停止尝试确立服务请求脉冲, 且客户端将其驱动器置于高阻抗状态。主机继续将 MDDI_Data0 驱动到逻辑 1 电平。如 F 点所示, 主机驱动 MDDI_Data0 至逻辑 0 电平达 50 微秒, 且也开始以与 MDDI_Data0 上的逻辑 0 电平一致的方式产生 MDDI_Stb。如 G 点所示, 在将 MDDI_Data0 确立为零电平并驱动 MDDI_Stb 达 50 微秒后, 主机通过发送子帧标头封包而开始在正向链路上传输数据。

[0287] 由以上论述, 可了解所涉及的前述解决方案使主机经历作为唤醒序列的部分的两个状态。对于第一个状态, 主机将 MDDI_Data0 讯号驱动到高并持续 150 微秒, 接着将 MDDI_

Data0 讯号驱动到低并持续 50 微秒同时启动 MDDI_Stb 线,且接着开始传输 MDDI 封包。此过程在使用 MDDI 设备和方法可达到的数据率方面很好地推进了此项技术。但是,如上文所述,仍然需要更高的速度(如对条件的响应时间缩短或更快地选择下一步骤或过程)或者简化处理或元件的能力。

[0288] 申请人已发现一种用于唤醒处理和时序的新颖的发明性方法,其中主机使用基于时钟循环的时序来进行讯号双态切换。在此配置中,在主机于唤醒序列开始处驱动 MDDI_Data0 讯号为高后 0 到 10 微秒时间中,主机开始双态切换 MDDI_Stb,且不会等待直到讯号被驱动到低。在唤醒序列期间,主机就像 MDDI_Data0 讯号总在逻辑 0 电平一般双态切换 MDDI_Stb。这有效地从客户端侧消除了时间的概念,且主机从用于头两个状态的前述 150 微秒和 50 微秒周期变化到用于这些周期的 150 个时钟循环和 50 个时钟循环。

[0289] 主机现负责将该数据线驱动到高,且负责在 10 个时钟循环内开始传输选通讯号(好像数据线为零)。在主机已将数据线驱动到高达 150 个时钟循环后,主机将数据线驱动到低并持续 50 个时钟循环,同时继续传输选通讯号。在完成所有这些处理后,主机可开始传输第一个子帧标头封包。

[0290] 在客户端方,客户端实施现在可使用所产生的时钟来计算数据线首先高接着低所经历的时钟循环的数目。在数据线被驱动到高的状态中所需发生的时钟循环的数目为 150,且在数据线被驱动到低的状态中所需发生的时钟循环的数目为 50。这意味着对于正确的唤醒序列,客户端应能够计数数据线为高时的至少 150 个连续时钟循环,接着是数据线为低时至少 50 个连续时钟循环。一旦满足了这两个条件,客户端可开始搜索第一个子帧的唯一字。基于这一模式中的中断来将计数器返回到初始状态,在所述状态中客户端再度寻找数据线为高时的头 150 个连续时钟循环。

[0291] 用于基于主机的休眠唤醒的本发明的客户端实施非常类似于最初的启动案例,不同之处是时钟速率未被强迫开始于 1Mbps,如上文所论述。相反,可将时钟速率设定为恢复至通讯链路进入休眠时有效的任何先前速率。如果主机如上文所述开始传输选通讯号,客户端应能够再次计数数据线为高时的至少 150 个连续时钟循环,接着是数据线为低时的至少 50 个连续时钟循环。一旦满足了这两个条件,客户端可开始搜索所述唯一字。

[0292] 用于基于客户端的休眠唤醒的本发明的客户端实施类似于基于主机的唤醒,不同之处是其通过使客户端驱动数据线来开始。客户端可在无时钟的情况下异步地驱动数据线以唤醒主机装置。一旦主机识别出数据线正由客户端驱动到高,其可开始其唤醒序列。客户端可计数由主机开始或在其唤醒过程中产生的时钟循环的数目。一旦客户端计数了数据线为高时的 70 个连续时钟循环,其可停止将数据线驱动到高。此时,主机也应已在将数据线驱动到高。客户端接着可计数数据线为高时的另外 80 个连续时钟循环以达到数据线为高时的 150 个时钟循环,且接着可寻找数据线为低时的 50 个时钟循环。一旦满足了这三个条件,客户端可开始寻找所述唯一字。

[0293] 唤醒处理的这一新颖的实施方式的一个优点是,其消除了时间度量装置的必要性。无论是振荡器或电容器放电电路或其它此类已知装置,客户端不再需要此类外部装置来确定启动条件。这当在客户端装置电路板上建构控制器、计数器等节省了金钱和电路面积。对于主机来说,此技术也在用于核心电路的极高密度逻辑(very high density logic, VHDL)方面潜在地简化了主机,尽管这对于客户端不一定同样有利。使用数据线和

选通线作为唤醒提示和测量源的功耗也会较低,因为外部电路不需要为将等待基于主机的唤醒的核心元件而运行。

[0294] 所使用的循环或时钟周期的数目是例示性的,且如对所属领域的技术人员显而易见地,可使用其它周期。

[0295] 唤醒处理的这一新颖的实施方式的一个优点是,其消除了时间度量装置的必要性。无论是振荡器或电容器放电电路或其它此类已知装置,客户端不再需要此类外部装置来确定启动条件。这当在客户端装置电路板上建构控制器、计数器等时节省了金钱和电路面积。对于主机来说,此技术也在用于核心电路的极高密度逻辑(very high density logic, VHDL)方面潜在地简化了主机,尽管这对于客户端不一定同样有利。使用数据线和选通线作为唤醒提示和测量源的功耗也会较低,因为外部电路不需要为将等待基于主机的唤醒的核心元件而运行。

[0296] 为阐释及说明此新技术的操作,图 68A、68B 和 68C 中展示了 MDDI_Data0、MDDI_Stb 和与时钟循环有关的各种操作的时序。

[0297] 图 68A 中说明了无争用的典型主机起始唤醒的处理步骤的实例,在说明中为方便起见再次使用字母 A、B、C、D、E、F 和 G 来对事件标号。过程开始于 A 点,此时主机发送一个链路关闭封包给客户端装置以通知其链路将转换到低功率休眠状态。在 B 点,在下一步骤中,主机在约 64 个循环中(或按系统设计的需要)双态切换 MDDI_Stb,以允许在停止 MDDI_Stb 双态切换(其停止客户端装置中的经恢复的时钟)前客户端的处理可完成。主机最初也将 MDDI_Data0 设定到逻辑 0 电平并接着在 CRC 后 16 到 48 个循环的范围内(一般包括输出禁用传播延迟)禁用 MDDI_Data0 输出。可能需要将客户端中的用于 MDDI_Data0 和 MDDI_Stb 的高速接收器在 CRC 后的 48 个循环后且在下一阶段(C)之前的某时刻置于低功率状态。

[0298] 通过禁用 MDDI_Data0 和 MDDI_Stb 驱动器并将主机控制器置于低功率休眠状态中,在 C 点或步骤 C 处,主机进入低功率休眠状态。按需要,也可将 MDDI_Stb 驱动器设定到逻辑 0 电平(使用一个高阻抗偏压网络),或将 MDDI_Stb 驱动器设定为在休眠期间继续双态切换。客户端也处在低功率水平的休眠状态中。

[0299] 在某段时间后,在 D 点处,主机通过启用 MDDI_Data0 和 MDDI_Stb 驱动器输出而开始链路重新启动序列。主机将 MDDI_Data0 驱动到逻辑 1 电平并将 MDDI_Stb 驱动到逻辑 0 电平,直到驱动器可完全启用其各别输出为止。在这些输出到达所要逻辑电平后,并在 MDDI_Stb 上驱动脉冲之前,主机一般等待约 200 纳秒。这为客户端提供了准备进行接收的时间。

[0300] 如 E 点所示,当主机驱动器已经启用且 MDDI_Data0 被驱动到逻辑 1 电平时,主机开始在 150 个 MDDI_Stb 循环的持续时间中双态切换 MDDI_Stb。如 F 点所示,主机在 50 个循环中驱动 MDDI_Data0 到逻辑 0 电平,且客户端在 MDDI_Data0 已在逻辑 0 电平持续了 40 个 MDDI_Stb 循环后开始寻找子帧标头封包。如 G 点所示,主机通过发送子帧标头封包而开始在正向链路上传输数据。

[0301] 图 68B 中说明了无争用的典型客户端起始唤醒的处理步骤的实例,在说明中为方便起见再次使用字母 A、B、C、D、E、F、G、H 和 I 来对事件标号。如前所述,过程开始于 A 点,此时主机发送一个链路关闭封包以通知客户端链路将转换到低功率状态。

[0302] 在 B 点,主机双态切换 MDDI_Stb 持续约 64 个循环(或按系统设计的需要),以允

许在停止 MDDI_Stb 双态切换（其停止客户端装置中的恢复的时钟）之前完成客户端所实施的处理。主机最初还将 MDDI_Data0 设定到逻辑 0 电平并接着在 CRC 后 16 到 48 个循环的范围内（一般包括输出禁用传播延迟）禁用 MDDI_Data0 输出。可能需要将客户端中用于 MDDI_Data0 和 MDDI_Stb 的高速接收器在 CRC 后的 48 个循环后且在下一阶段（C）之前的某时刻置于低功率状态。

[0303] 通过禁用 MDDI_Data0 和 MDDI_Stb 驱动器并将主机控制器置于低功率休眠状态中，在 C 点或步骤 C 处，主机进入低功率休眠状态。按需要，也可将 MDDI_Stb 驱动器设定到逻辑 0 电平（使用一个高阻抗偏压网络），或将 MDDI_Stb 驱动器设定为在休眠期间继续双态切换。客户端也处在低功率水平的休眠状态中。

[0304] 在某段时间后，在 D 点处，客户端通过下述方式开始链路重新启动序列：启用 MDDI_Stb 接收器且也启用 MDDI_Stb 接收器中的一个偏移，以保证在主机启用其 MDDI_Stb 驱动器之前所接收的 MDDI_Stb 版本在客户端中是逻辑 0 电平。客户端可能需要比启用接收器稍早地启用偏移，以按需要确保接收到一个有效的差分讯号并禁止错误讯号。客户端启用 MDDI_Data0 驱动器，同时将 MDDI_Data0 线驱动到逻辑 1 电平。

[0305] 在约 1 毫秒内，在 E 点处，主机识别来自客户端的服务请求脉冲，且主机通过启用 MDDI_Data0 和 MDDI_Stb 驱动器输出而开始链路重新启动序列。主机将 MDDI_Data0 驱动到逻辑 1 电平并将 MDDI_Stb 驱动到逻辑 0 电平，直到驱动器完全启用其各自的输出为止。在这些输出达到所要逻辑电平后，并在 MDDI_Stb 上驱动脉冲之前，主机一般等待约 200 纳秒。这为客户端提供了准备进行接收的时间。

[0306] 如 F 点所示，当主机驱动器已经启用且 MDDI_Data0 被驱动到逻辑 1 电平时，主机开始在 MDDI_Stb 上输出脉冲达 150 个 MDDI_Stb 循环的持续时间。当客户端识别出 MDDI_Stb 上的第一个脉冲时，其禁用其 MDDI_Stb 接收器中的偏移。客户端继续驱动 MDDI_Data0 到逻辑 1 电平达 70 个 MDDI_Stb 循环，且在 G 点处禁用其 MDDI_Data0 驱动器。

[0307] 如 G 和 H 点所示，主机在 50 个循环中将 MDDI_Data0 驱动到逻辑 0 电平，且客户端在 MDDI_Data0 在逻辑 0 电平中持续了 40 个 MDDI_Stb 循环后开始寻找子帧标头封包。如 I 点所示，主机通过发送子帧标头封包而开始在正向链路上传输数据。

[0308] 图 68C 展示了典型主机起始唤醒的处理步骤的实例，其具有来自客户端的争用，也就是说客户端也想要唤醒所述链路。在说明中为方便起见再次使用字母 A、B、C、D、E、F、G、H 和 I 来对事件标号。如前所述，过程开始于 A 点，此时主机发送一个链路关闭封包以通知客户端链路将转换到低功率状态；继续到 B 点，此时 MDDI_Stb 被双态切换约 64 个循环（或按系统设计的需要），以允许完成客户端的处理；且接着到 C 点，此时通过禁用 MDDI_Data0 和 MDDI_Stb 驱动器并将主机控制器置于低功率休眠状态中，主机进入低功率休眠状态。在某段时间后，在 D 点处，主机通过启用 MDDI_Data0 和 MDDI_Stb 驱动器输出而开始链路重新启动序列；且如 E 点所示，主机开始在 150 个 MDDI_Stb 循环的持续时间中双态切换 MDDI_Stb。

[0309] 在 E 点后的多达 70 个 MDDI_Stb 循环处（本文为 F 点），客户端仍未识别出主机正将 MDDI_Data0 驱动到逻辑 1 电平，所以客户端也将 MDDI_Data0 驱动到逻辑 1 电平。发生这种情况的原因是，客户端想请求服务，但未识别出其试图通讯的主机已经开始了链路重新启动序列。在 G 点处，客户端停止驱动 MDDI_Data0，且通过禁用其驱动器的输出而将其驱动

器置于高阻抗状态。主机在另外的 80 个循环中继续将 MDDI_Data0 驱动到逻辑 1 电平。

[0310] 如 H 点所示,主机在 50 个循环中将 MDDI_Data0 驱动到逻辑 0 电平,且客户端在 MDDI_Data0 在逻辑 0 电平中持续了 40 个 MDDI_Stb 循环后开始寻找子帧标头封包。如 I 点所示,主机通过发送子帧标头封包而开始在正向链路上传输数据。

[0311] VI. 接口电气规格

[0312] 在实例性实施例中,使用数据选通讯号或 DATA-STB 格式来编码不归零 (NRZ) 格式的数据,这允许将时钟信息嵌入数据和选通讯号中。可在无复杂锁相回路电路的情况下恢复时钟。经由一般用有线电缆实施的双向差动链路 (differential link) 来运载数据,尽管如上文所述也可使用其它导体、印刷线路或传送元件。经由一条仅由主机驱动的单向链路来运载选通讯号 (STB)。不管何时存在 0 或 1 的交替相邻 (back-to-back) 状态,选通讯号都将值 (0 或 1) 双向切换,对于数据线或讯号也是一样。

[0313] 图 40 中以图形形式展示了如何用 DATA-STB 编码法传输诸如多个位“1110001011”的数据序列的实例。在图 40 中,DATA 讯号 4002 展示于讯号时序图的上方的线上,而 STB 讯号 4004 展示于第二条线上,其时间经适当地对准 (共同的开始点)。随着时间经过,当 DATA 线 4002 (讯号) 上发生状态变化时,STB 线 4004 (讯号) 保持先前状态,因此,DATA 讯号的第一个“1”状态相关于 STB 讯号的第一个“0”状态 (其开始值)。然而,如果或当 DATA 讯号的状态、电平不改变时,在本实例中 STB 讯号就双态切换到相反状态或“1”,如在图 40 中 DATA 呈另一个“1”值的情况下。也就是说,在 DATA 和 STB 中每个位循环中有一个且仅有一个转换。因此,STB 讯号再次转换,这次当 DATA 讯号保持在“1”时 STB 讯号转换到“0”,且当 DATA 讯号电平改变到“0”时 STB 讯号保持此电平或值 (“0”)。在本实例中,当 DATA 讯号保持在“1”时,STB 讯号双态切换到相反状态或“1”,且随着 DATA 讯号改变或保持电平或值,STB 讯号进行诸如此类的变化。

[0314] 在接收到这些讯号时,对 DATA 和 STB 讯号执行异或 (XOR) 运算以产生时钟讯号 4006,其展示于时序图的底部以与所要数据和选通讯号进行相对比较。图 41 展示了一个实例电路,其适用于在主机处从输入数据产生 DATA 和 STB 输出或讯号,并接着在客户端处从 DATA 和 STB 讯号恢复或重获所述数据。

[0315] 在图 41 中,使用传输部分 4100 来产生并经由中间讯号路径 4102 传输原始 DATA 和 STB 讯号,同时使用接收部分 4120 来接收所述讯号并恢复数据。如图 41 所示,为了将数据从主机传送到客户端,将 DATA 讯号连同用于触发所述电路的时钟讯号一起输入到两个 D 型触发电路元件 4104 和 4106。接着使用两个差动线驱动器 4108 和 4110 (电压模式) 分别将所述两个触发电路输出 (Q) 分成差动的讯号对 MDDI_Data0+, MDDI_Data0- 和 MDDI_Stb+, MDDI_Stb-。连接了一个三输入同 (XNOR) 门、电路或逻辑元件 4112 以接收 DATA 及两个触发器的输出,且产生作为第二触发器的数据输入的输出,所述第二触发器又产生 MDDI_Stb+, MDDI_Stb- 讯号。为了方便起见,同门上标有反向圆圈以指示其可有效地将产生选通讯号的触发器的 Q 输出反向。

[0316] 在图 41 的接收部分 4120 中, MDDI_Data0+, MDDI_Data0- 和 MDDI_Stb+, MDDI_Stb- 被两个差动线接收器 4122 和 4124 各自接收,所述差动线接收器从差动讯号产生单一输出。放大器的输出接着输入到产生时钟讯号的两输入异或 (XOR) 门、电路或逻辑元件 4126 的各个输入端。所述时钟讯号被用于触发两个 D 型触发电路 4128 和 4130 中的每一

者,所述触发电路经由延迟元件 4132 而接收 DATA 讯号的经延迟版本,所述触发电路之一(4128)产生数据“0”值且另一个(4130)产生数据“1”值。时钟也具有来自 XOR 逻辑的独立输出。由于时钟信息被分布在 DATA 和 STB 线中,所以讯号在状态间的转换都不会快于时钟速率的一半。由于通过对 DATA 和 STB 讯号的异或处理而再生了时钟,所以与其中时钟讯号被直接经由单一专用数据线发送的情况相比较,所述系统可有效地容忍两倍的输入数据和时钟间偏斜量。

[0317] MDDI 数据对, MDDI_Stb+ 和 MDDI_Stb- 讯号操作于差动模式中以最大化对噪音的抗扰性。差动讯号路径的每一部分被源端接 (source terminated), 其中电缆或导体的一半特征阻抗被用于传送讯号。MDDI 数据对在主机和客户端侧都被源端接。由于在某给定时刻这两个驱动器中仅有一个是有效的,所以在传送链路的源处持续存在端接。MDDI_Stb+ 和 MDDI_Stb- 讯号仅由主机驱动。

[0318] 图 42 展示了适用于实现作为发明性 MDD 接口的一部分的用于传送讯号的驱动器、接收器和端接的元件的例示性配置,同时表 VII 中展示了 MDDI_Data 和 MDDI_Stb 的对应 DC 电气规格。此例示性接口使用低电压感应,此处为 200mV,具有小于 1 伏的功率摆动和低耗用功率。

[0319] 表 VII

[0320]

| 参数 | 说明 | 最小值 | 典型值 | 最大值 | 单位 |
|-------------------|----------------------|------|------|------|------|
| R _{端接} | 串联端接 | 41.3 | 42.2 | 43.0 | 欧姆 |
| R _{休眠} | 休眠状态偏压端接 | 8 | 10 | 12 | K-欧姆 |
| V _{休眠} | 休眠状态开路电压 | 0.5 | | 2.8 | V |
| V _{输出范围} | 相对于 GND 的驱动器输出电压容许范围 | 0 | | 2.8 | V |
| V _{OD+} | 驱动器差动输出高电压 | 0.5 | | | V |
| V _{OD-} | 驱动器差动输出低电压 | | | -0.5 | V |
| V _{IT+} | 接收器差动输入高阈电压 | | | 10 | mV |
| V _{IT-} | 接收器差动输入低阈电压 | -10 | | | mV |
| V _{输入范围} | 相对于 GND 的接收器输入电压容许范围 | 0 | | 3.0 | V |
| I _{输入} | 输入漏电流 (不包括休眠偏压) | -25 | | 25 | μA |

[0321] 表 VIII 描述了差动线驱动器和差动线接收器的电气参数和特征。在功能上,所述驱动器将输入端上的逻辑电平直接传送到正输出端,且将所述输入的反向传送到负输出端。自输入端到输出端的延迟良好地匹配于差动线(其被差动地驱动)。在大多数实施中,输出端上的电压摆动小于输入端上的摆动,以最小化功耗和电磁发射。表 VIII 提供了约为 0.5V 的最小电压摆动。但是,如所属领域的技术人员所知,也可使用其它值,且发明者在某些实施例中根据设计限制而构想了更小的值。

[0322] 差动线接收器与高速电压比较器具有相同的特征。在图 41 中,没有圆圈的输入端是正输入端,而有圆圈的输入端是负输入端。如果 (V_{输入+}) - (V_{输入-}) 大于零,则输出为逻辑 1。另一种方式是将其描述为具有极大(实际上是无限)增益的差动放大器,其输出被限幅于逻辑 0 和 1 电压电平。

[0323] 应最小化差动对之间的延迟偏斜,以在最高可能速度下操作所述差动传输系统。

[0324] 在图 42 中,展示了经由通讯链路 4206 传送封包的主机控制器 4202 和客户端或显示器控制器 4204。主机控制器采用一系列的三个驱动器 4210、4212 和 4214 来接收待传送的主机 DATA 和 STB 讯号,以及接收待传送的客户端数据讯号。负责传递主机 DATA 的驱动器

采用一个启用讯号输入来允许一般仅当希望发生从主机到客户端的传送时启动通讯链路。由于 STB 讯号是作为数据传送的一部分形成的,所以未将额外的启用讯号用于那个驱动器(4212)。DATA 和 STB 驱动器中每一驱动器的输出端分别连接到端接阻抗或电阻器 4216a、4216b、4216c 和 4216d。

[0325] 端接电阻器 4216a 和 4216b 也将充当用于 STB 讯号处理的客户端侧接收器 4220 的输入端上的阻抗,而在客户端数据处理接收器 4222 的输入端上额外的端接电阻器 4216e 和 4216f 分别与电阻器 4216c 和 4216d 串联。客户端控制器中的第六驱动器 4226 用于预备从客户端传送到主机的数据讯号,其中驱动器 4214(经由在输入端侧的端接电阻器 4216c 和 4216d) 处理数据以传送到主机供处理。

[0326] 分别在端接电阻器和接地端及端接电阻器和电压源 4220 之间设置了两个额外的电阻器 4218a 及 4218b,作为本文其它地方论述的休眠控制的一部分。所述电压源用于将传送线驱动到先前论述的高或低电平,以管理数据流。

[0327] 以上驱动器和阻抗可作为离散组件或作为电路模块的一部分形成,或作为专用集成电路(ASIC)(其充当了一种更节省成本的编码器或解码器解决方案)的一部分形成。

[0328] 可容易地发现,功率是利用经由一对导体的标号为 MDDI_Pwr 和 MDDI_Gnd 的讯号而从主机装置传送到客户端装置或显示器的。讯号的 MDDI_Gnd 部分充当参考接地电平和用于显示装置的讯号的供电返回路径。MDDI_Pwr 讯号充当显示装置电源,其由主机装置驱动。在一个例示性配置中,对于低功率应用来说,允许显示装置汲取多达 500mA 的电流。可从便携式电源,诸如(但不限于)位于主机装置处的锂离子型电池或电池组提供 MDDI_Pwr 讯号,且 MDDI_Pwr 讯号相对于 MDDI_Gnd 的电压差可在 3.2 至 4.3 伏的范围内。

[0329] VII. 时序特征

[0330] A. 概观

[0331] 图 43 说明了由客户端用于从主机处获取服务且由主机用于提供此等服务的步骤和讯号电平。在图 43 中,所说明的讯号的第一部分展示了正从主机传送的链路关闭封包,且接着数据线被用高阻抗偏压电路驱动到逻辑 0 状态。客户端显示器或主机(其驱动器被禁用)不传输任何数据。由于 MDDI_Stb 在链路关闭封包期间是有效的,所以在底部可看到一系列用于 MDDI_Stb 讯号线的选通脉冲。一旦此封包结束且逻辑电平随着主机将偏压电路和逻辑驱动到 0 而改变到 0, MDDI_Stb 讯号线也改变到 0 电平。这代表来自主机的上一次讯号传送或服务的终止,且可在过去的任何时间发生;将其包括在内以展示服务的先前终止和在服务开始前的讯号状态。如果需要,可在不要求此主机装置已进行“已知”的先前通讯的情况下,仅为将通讯链路复位到正确状态而发送此讯号。

[0332] 如图 43 所示,自客户端输出的讯号最初被设定在逻辑 0 电平。换句话说,客户端输出在高阻抗下,且驱动器被禁用。当请求了服务时,客户端启用其驱动器并将服务请求发送给主机,其是标记为 t 服务的一个时间段,在该时间段内线路被驱动到逻辑 1 电平。在主机侦测到所述服务请求前,经过或可需要特定量的时间(称为 t_{主机-侦测}),其后主机通过将讯号驱动到逻辑 1 电平而以链路启动序列进行响应。此时,客户端撤销请求,且禁用服务请求驱动器,使得源自客户端的输出线再次成为逻辑 0 电平。在此时间期间, MDDI_Stb 讯号处于逻辑 0 电平。

[0333] 主机在称为 t_{重新启动-高}的时间段中将主机数据输出端驱动到“1”电平,其后主

机在称为 t 重新启动 - 低的时间段中将所述逻辑电平驱动到 0 并启动 MDDI_Stb, 其后通过子帧标头封包开始第一正向讯务, 且接着传送正向讯务封包。MDDI_Stb 讯号在 t 重新启动 - 低的时间段和随后的子帧标头封包期间有效。

[0334] 表 VIII 展示了上文论述的各个时间段长度的代表性时间, 及与例示性最小和最大数据率的关系, 其中:

$$[0335] \quad t_{\text{bit}} = \frac{1}{\text{Link_Data_Rate}}$$

[0336] 表 VIII

[0337]

| 参数 | 说明 | 最小值 | 典型值 | 最大值 | 单位 |
|------------------------|------------------|--------|-----|-----|------|
| $t_{\text{服务}}$ | 显示器服务请求脉冲的持续时间 | 60 | 70 | 80 | 秒 |
| $t_{\text{重启动-高}}$ | 主机链路重新启动高脉冲的持续时间 | 140 | 150 | 160 | 微秒 |
| $t_{\text{重启动-低}}$ | 主机链路重新启动低脉冲的持续时间 | 40 | 50 | 60 | 微秒 |
| $t_{\text{显示器-检测}}$ | 显示器侦测链路重新启动序列的时间 | 1 | | 50 | 微秒 |
| $t_{\text{主机-检测}}$ | 主机侦测服务请求脉冲的时间 | 1 | | 50 | 微秒 |
| $1/t_{\text{位-最小-性能}}$ | 最小性能装置的链路数据率 | 0.001 | | 1 | Mbps |
| $1/t_{\text{位-最大-性能}}$ | 装置的最大链路数据率范围 | 0.001 | | 450 | Mbps |
| | 反向链路数据率 | 0.0005 | | 50 | Mbps |
| t_{bit} | 一个正向链路数据位的周期 | 2.2 | | 10° | 纳秒 |

[0338] 所属领域的技术人员将易于了解图 41 和 42 中说明的个别元件的功能是众所周知的,且图 42 中的元件的功能得到图 43 中的时序图的证实。图 41 中省略了关于图 42 所示的串联端接和休眠电阻器的细节,因为此种信息对于说明如何执行数据 - 选通编码并自其恢复时钟来说并非必要。

[0339] B. 数据 - 选通定时正向链路

[0340] 表 IX 展示了在正向链路上从主机驱动器输出端传送数据的开关特征。表 IX 提供了某些讯号转换发生所需的最小和最大时间以及典型时间的表格。例如,从一个数据值(为“0”或“1”的输出)的开始到结束发生的转换(Data0 到 Data0 的转换,称为 ttdd-(主机-输出))的典型时间长度是 ttbit,同时最小时间是约 ttbit-0.5 纳秒而最大时间是约 ttbit+0.5 纳秒。图 44 说明了 Data0、其它数据线(DataX)、及选通线(Stb)上的转换的相对间距,其中展示了 Data0 至选通、选通至选通、选通至 Data0、Data0 至非 Data0、非 Data0 至非 Data0、非 Data0 至选通、及选通至非 Data0 的转换,所述转换分别称为 ttdds-(主机-输出)、ttss-(主机-输出)、ttsd-(主机-输出)、ttddx-(主机-输出)、ttddx-(主机-输出)、ttdds-(主机-输出)、及 ttssdx-(主机-输出)。

[0341] 表 IX

[0342]

| 参数 | 说明 | 最小值 | 典型值 | 最大值 | 单位 |
|---------------------|---------------------|-------------------|------------|------------------|----|
| t_{tdd-} (主机-输出) | Data0 至 Data0 转换 | $t_{tbit-} - 0.5$ | t_{tbit} | $t_{tbit} + 0.5$ | 纳秒 |
| t_{tds-} (主机-输出) | Data0 至选通转换 | $t_{tbit-} - 0.8$ | t_{tbit} | $t_{tbit} + 0.8$ | 纳秒 |
| t_{tss-} (主机-输出) | 选通至选通转换 | $t_{tbit} - 0.5$ | t_{tbit} | $t_{tbit} + 0.5$ | 纳秒 |
| t_{tsd-} (主机-输出) | 选通至 Data0 转换 | $t_{tbit} - 0.8$ | t_{tbit} | $t_{tbit} + 0.8$ | 纳秒 |
| t_{tddx-} (主机-输出) | Data0 至非 Data0 转换 | | t_{tbit} | | 纳秒 |
| t_{tdsx-} (主机-输出) | 非 Data0 至非 Data0 转换 | $t_{tbit} - 0.5$ | t_{tbit} | $t_{tbit} + 0.5$ | 纳秒 |
| t_{tdxs-} (主机-输出) | 非 Data0 至选通转换 | | t_{tbit} | | 纳秒 |
| t_{tsdx-} (主机-输出) | 选通至非 Data0 转换 | | t_{tbit} | | 纳秒 |

[0343] 表 X 展示了在正向链路上传送数据的相同讯号的客户端接收器输入的典型 MDDI 时序要求。如所属领域的技术人员会了解的,由于所论述的是经过时延的相同讯号,所以不需要新的图来说明各别标号的讯号特征或意义。

[0344] 表 X

[0345]

| 参数 | 说明 | 最小值 | 典型值 | 最大值 | 单位 |
|---------------------|---------------------|----------------|------------|----------------|----|
| t_{tdd-} (显示器-输入) | Data0 至 Data0 转换 | $t_{tbit-1.0}$ | t_{tbit} | $t_{tbit+1.0}$ | 纳秒 |
| t_{tds-} (显示器-输入) | Data0 至选通转换 | $t_{tbit-1.5}$ | t_{tbit} | $t_{tbit+1.5}$ | 纳秒 |
| t_{tss-} (显示器-输入) | 选通至选通转换 | $t_{tbit-1.0}$ | t_{tbit} | $t_{tbit+1.0}$ | 纳秒 |
| t_{tsd-} (显示器-输入) | 选通至 Data0 转换 | $t_{tbit-1.5}$ | t_{tbit} | $t_{tbit+1.5}$ | 纳秒 |
| t_{tddx-} (主机-输出) | Data0 至非 Data0 转换 | | t_{tbit} | | 纳秒 |
| t_{tdxs-} (主机-输出) | 非 Data0 至非 Data0 转换 | | t_{tbit} | | 纳秒 |
| t_{tdxs-} (主机-输出) | 非 Data0 至选通转换 | | t_{tbit} | | 纳秒 |
| t_{tsdx-} (主机-输出) | 选通至非 Data0 转换 | | t_{tbit} | | 纳秒 |

[0346] 图 45 和 46 分别说明了当主机禁用或启用主机驱动器时可发生的响应延迟的存在。在主机转发某些封包的情况下,诸如反向链路封装封包或往返行程延迟测量封包,在转发所要的封包后,诸如图 45 所说明的参数 CRC 封包、选通对准封包和全零封包(作为已传送的),主机禁用线驱动器。但是,如图 45 所示,所述线路的状态不一定立刻从“0”切换到所要的较高值(尽管通过现有的某些控制或电路元件,这是有可能实现的),而是要花费称

为主机驱动器禁用延迟周期的一段时间来响应。尽管所述切换可实际上即刻发生使得此时间段长度为 0 纳秒 (nsec.)，但其更易于跨越某个更长的时期，10 纳秒是理想的最大周期长度，其发生在保护时间 1 或逆转 1 封包周期期间。

[0347] 观察图 46，可看出当主机驱动器被启用以传送诸如反向链路封装封包或往返行程延迟测量封包的封包时会经历讯号电平改变。此处，在保护时间 2 或逆转 2 封包周期后，主机驱动器被启用且开始驱动一个电平（此处为“0”），在经过一个称为主机驱动器启用延迟周期的时间段后接近或到达所述值，其发生在驱动器重新启用周期期间，先于发送第一个封包。

[0348] 对于客户端装置（此处为显示器）的驱动器和讯号传送来说，会发生类似的过程。以下表 XI 展示了这些周期的长度及其各自关系的一般准则。

[0349] 表 XI

[0350]

| 说明 | 最小值 | 最大值 | 单位 |
|------------|-----|-----|----|
| 主机驱动器禁用延迟 | 0 | 10 | 纳秒 |
| 主机驱动器启用延迟 | 0 | 2.0 | 纳秒 |
| 显示器驱动器禁用延迟 | 0 | 10 | 纳秒 |
| 显示器驱动器启用延迟 | 0 | 2.0 | 纳秒 |

[0351] C. 数据 - 选通定时反向链路

[0352] 图 47 和 48 中展示了用于在反向链路上从客户端驱动器输出端传送数据的数据讯号和选通讯号的开关特征和时序关系。下文论述某些讯号转换的典型时间。图 47 说明了在主机接收器输入端处的所传送数据的时序和选通脉冲的前缘和后缘之间的关系。也就是说，选通讯号的上升缘或前缘的建立时间称为 t_{su-sr} ，且选通讯号的后缘或下降缘的建立时间称为 t_{su-sf} 。这些建立周期的典型时间长度约为最小 8 纳秒。

[0353] 图 48 说明了开关特征和由反向数据时序产生的对应客户端输出延迟。在图 48 中，可看到所传送数据的时序和选通脉冲的前缘及后缘（考虑到所诱发的延迟）之间的关系。也就是说，选通讯号的上升缘或前缘和数据（有效）之间的传播延迟称为 t_{pd-sr} ，且数据和选通讯号的后缘或下降缘之间的传播延迟称为 t_{pd-sf} 。这些传播延迟周期的典型最大时间长度约为 8 纳秒。

[0354] VIII. 链路控制的实施（链路控制器操作）

[0355] A. 状态机封包处理器

[0356] 经由 MDDI 链路传送的封包的分配速度极快，通常速率约为 300Mbps 或更大（诸如 400Mbps），尽管按需要当然可包含较低的速率。此类总线或传送链路速度过快使得当前可（经济地）购得的通用微处理器或类似物无法控制。因此，为实现此类讯号传送的一个实际的实施方式是利用可编程的状态机来分析所输入的封包流以产生封包，所述封包被传送或重定向到希望其到达的适当音频 - 视频子系统。此等装置已是熟知的，且使用了一般专用于有限量的操作、功能或状态的电路以实现理想的高速或极高速操作。

[0357] 通用控制器、处理器、或处理元件可用来更合适地按某些具有较低速度需求的信息行动或操纵所述信息，诸如控制或状态封包。在将音频和视频封包传送到其适当的目的地以进行动作的同时，当接收到这些封包（控制、状态或其他预定义的封包）时，状态机应将其经由数据缓冲区或类似处理元件而传递到通用处理器，以便可按照所述封包而提供所

要的结果（效果）。如果在未来制造的微处理器或其他通用控制器、处理器或处理元件达到了更高的数据率处理能力，那么下文论述的状态机也可用对此等装置的软件控制来实施，一般作为存储在存储元件或介质上的程序。

[0358] 在某些实施例中，通用处理器功能可通过利用电脑应用中的微处理器（CPU）或无线装置中的控制器、处理器、数字讯号处理器（DSP）、特殊电路或 ASIC 中可用的处理能力或过剩循环来实现，其方式与某些调制解调器或图形处理器利用电脑中的 CPU 的处理能力来执行某些功能并减少硬件复杂性和成本一样。但是，这种循环共享或使用会不利地影响此等元件的处理速度、时序或总体操作，所以在许多应用中，最好将专用电路或元件用于这种一般处理。

[0359] 为了在显示器（微型显示器）上观看影像数据，或可靠地接收由主机装置发送的所有封包，将显示讯号处理与正向链路通道时序同步。也就是说，为进行适当的讯号处理，到达显示器和显示器电路的讯号需要大体在时间上同步。图 49 说明了借以实施此种同步的一种方法或多个讯号处理步骤所达成的状态的高级图。在图 49 中，展示了状态机 4900 的可能的正向链路同步“状态”，所述状态被划分为：一个异步帧状态（ASYNC FRAMES STATE）4904、两个获取同步状态（ACQUIRING SYNC STATE）4902 和 4906，及三个同步状态（IN-SYNC STATE）4908、4910 和 4912。

[0360] 如起始步骤或状态 4902 所示，诸如一个呈现装置的显示器或客户端开始于一个预先选定的“非同步”状态，且在所侦测到的第一个子帧标头封包中搜寻唯一字。应注意这个非同步状态代表了最小通讯设定或“后退（fall-back）”设定（在其中选择 I 型接口）。当在所述搜寻过程中发现了唯一字时，显示器保存该子帧长度域。对于对此第一帧的处理来说，不校验 CRC 位，或者直到实现同步后才校验 CRC 位。如果此子帧长度为零，则同步状态处理相应地继续到此处标为“异步帧”状态的状态 4904，其指示还没有实现同步。在图 49 中，此处理步骤被标为遇到 cond 3 或条件 3。否则，如果帧长度大于零，则同步状态处理继续到状态 4906，其中接口状态被设定为“发现一个同步帧”。在图 49 中，此处理步骤被标为遇到 cond 5 或条件 5。此外，如果状态机发现帧标头封包且判定 CRC 良好且帧长度大于零，则处理继续到“发现一个同步帧”状态。在图 49 中，此被标为遇到 cond 6 或条件 6。

[0361] 在系统处于“非同步”以外的状态中的各种情况下，当侦测到唯一字且针对子帧标头封包判定 CRC 结果良好，且子帧长度大于零，则接口状态改变到“同步”状态 4908。在图 49 中，此处理步骤被标为遇到 cond 1 或条件 1。另一方面，如果子帧标头封包中的唯一字或 CRC 中任一者不正确，则同步状态处理继续或返回到接口状态 4902，“非同步帧”状态。在图 49 的状态图中，此处理部分被标为遇到 cond 2 或条件 2。

[0362] B. 获取同步时间

[0363] 接口可经配置以在判定同步已丢失且返回到“非同步帧”状态前接纳特定数目的“同步错误”。在图 49 中，一旦状态机到达“同步状态”且未发现错误，则其继续遇到 cond 1 结果，且保持在“同步”状态。然而，一旦侦测到一个 cond 2 结果，处理就将状态改变到“一个同步错误”状态 4910。此时，如果处理导致侦测到另一个 cond 1 结果，则状态机返回到“同步”状态，否则其遇到另一个 cond 2 结果，且移至“两个同步错误”状态 4912。再次，如果发生 cond 1，则处理将状态机返回到“同步”状态。否则，遇到另一个 cond 2 且状态机返回到“非同步”状态。也可了解的是，如果接口遇到“链路关闭封包”，则这会导致链路终止

数据传送并返回到“非同步帧”状态,因为已不存在同步的对象,在图 49 的状态图中,这被称为遇到 cond 4 或条件 4。

[0364] 应了解可能存在可出现于子帧中的某固定位置处的唯一字的重复的“虚假复本”。在此情况下,将状态机与子帧同步是非常不可能的,因为为了使 MDD 接口处理继续到“同步”状态,子帧标头封包上的 CRC 在处理时必须也有效。

[0365] 子帧标头封包中的子帧长度可设定为零以指示在链路关闭之前主机将仅传输一个子帧,且 MDD 接口被置于或配置成闲置休眠状态。在此情况下,显示器必须在侦测到子帧标头封包后立刻经由正向链路接收封包,因为在链路转换到闲置状态前仅发送单个子帧。在正常或典型操作中,子帧长度是非零的,且显示器仅处理正向链路封包,同时接口处于图 49 中汇总地展示为“同步”状态的那些状态中。

[0366] 取决于子帧大小和正向链路数据率,显示器与正向链路讯号同步所需的时间是可变的。当子帧大小较大时,侦测到唯一字的作为正向链路中的随机(或更随机)数据的部分的“虚假复本”的可能性更大。同时,当正向链路数据率更低时,从虚假侦测中恢复的能力更低,且这样做所花的时间更长。

[0367] C. 初始化

[0368] 如先前所述,在“启动”时,主机配置正向链路以在 1Mbps 的最小所要求(或所需)数据率处或以下操作,且适当地配置子帧长度和媒体帧速率以用于给定应用。也就是说,正向链路和反向链路都使用 I 型接口开始操作。这些参数一般仅会在主机确定客户端显示器(或其他类型的客户端装置)的能力或所要配置时暂时使用。主机经由正向链路发送或传送一个子帧标头封包,接着是一个反向链路封装封包,其请求旗标的“0”位设定为值 1,以便请求显示器或客户端用一个显示器能力封包来响应。一旦显示器实现了对(或与)正向链路的同步,其经由反向链路或通道发送一个显示器能力封包和一个显示器请求和状态封包。

[0369] 主机检查显示器能力封包的内容以便确定如何重新配置链路以达到最优或理想的性能水平。主机检查协议版本域和最小协议版本域以确认主机和显示器使用了彼此兼容的协议版本。协议版本一般保持为显示器能力封包的头两个参数,以使得甚至在协议的其它要素不一定兼容或不一定完全理解为兼容时确定兼容性。

[0370] D. CRC 处理

[0371] 对于所有封包类型,封包处理器状态机确保 CRC 校验器受到适当或正确地控制。其也在 CRC 比较过程导致侦测到一个或一个以上错误时将 CRC 错误计数器加一,且在所处理的每一子帧的开始将 CRC 计数器复位。

[0372] E. 替代性同步丢失检查

[0373] 尽管以上一系列步骤或状态皆可作用以产生更高的数据率或吞吐速度,申请人已发现一种对客户端用以宣称丢失了和主机的同步的条件的替代性安排或改变可被有效地用于实现甚至更高的数据率或吞吐量。所述新的发明性实施例具有相同基本结构,但用于改变状态的条件变化了。此外,实施了一种新的计数器以帮助检查子帧同步。参照图 63 呈现这些步骤和条件,该图说明了适用于建立所述方法或状态机的操作的一系列状态和条件。为清晰起见,仅展示了“获取同步状态”和“同步状态”部分。此外,由于所得状态和状态机本身都大体相同,所以其使用相同编号。然而,用于改变状态(及状态机操作)的条件

有所变化,所以为清晰起见所有条件均在两个数字(1、2、3、4、5和6对61、62、63、64和65)间进行了重新编号,以方便识别差异。由于在此论述中不考虑异步帧状态,所以在该图中不再使用一个状态(4904)和一个条件(6)。

[0374] 在图63中,系统或客户端(用于显示或呈现)以状态机5000处于预先选定的“非同步”状态4902中(如图49)而开始。从非同步条件4902改变状态的第一次状态改变在条件64中,其是发现同步模式。假定对此封包的子帧标头的CRC也通过(满足条件61),封包处理器状态机的状态可改变到同步状态4908。一个同步错误(条件62)会导致状态机变化到状态4910,且第二个同步错误会导致状态机变化到状态4912。然而,已发现,MDDI封包的任何CRC失败将导致状态机离开同步状态4908,进入一个同步错误状态4910。任何MDDI封包的另一CRC失败将导致移动到两个同步失败状态4912。一个经解码具有正确CRC值的封包将导致状态机返回同步状态4908。

[0375] 已改变的是利用了针对“每个”封包的CRC值或判定。也就是说,使状态机检查每个封包的CRC值来判定同步丢失,而不是只观察子帧标头封包。在此配置或过程中,不是只使用唯一字和子帧标头CRC值来判定同步丢失。

[0376] 此新的接口实施方式允许MDD接口链路更快地识别出同步失败,且因此,也更快地从失败中恢复。

[0377] 为使此系统更健壮,客户端也应添加或利用一个子帧计数器。客户端接着在预期唯一字将到达或出现的时间在讯号中检查所述唯一字是否存在。如果唯一字未在正确时间出现,客户端可识别出发生了同步失败,这比客户端必须等待若干(此处为3个)封包时间或周期(其大于一个子帧长度)快很多。如果对唯一字的测试指示其不存在,换句话说,时序不正确,那么客户端可立刻宣布链路丢失了同步并进入非同步状态。检查适当的唯一字存在的过程添加条件65(cond 65)至状态机,意谓唯一字不正确。如果期望在客户端上接收到子帧封包而子帧封包未出现,客户端可立刻进入非同步状态4902,从而节省了等待通常在穿过状态4910和4912时遇到的多个同步错误(条件62)的额外时间。

[0378] 此改变使用了客户端核心中的额外的计数器或计数功能来计数子帧长度。在一个实施例中,使用一个倒计时功能,且在计数器期满的情况下中断当前正在处理的任一封包的传送以检查子帧唯一字。或者,计数器可正数,将计数与所要的最大值或特定所要值相比较,且在此点检查当前封包。此过程使客户端免于解码在所述客户端上错误地接收到的具有特别长的封包长度的封包。如果子帧长度计数器需要中断正被解码的某个其它封包,则可判定同步丢失,因为不应有封包跨越子帧边界。

[0379] IX. 封包处理

[0380] 就状态机接收的上文论述的每一类型的封包来说,其承受一个特殊处理步骤或一系列步骤以实施接口的操作。正向链路封包一般根据下文表XII中所列的例示性处理来加以处理。

[0381] 表XII

[0382]

| | |
|------------------|---|
| 封包类型 | 封包处理器状态机响应 |
| 子帧标头(SH) | 确认良好封包，获取子帧长度域，且将封包参数发送到通用处理器。 |
| 填充符(F) | 忽略数据。 |
| 视频流(VS) | 解译视频数据格式描述符及其他参数，必要时解包压缩的像素数据，如果必要则经由色彩映射转译像素，且将像素数据写至位图中的适当位置。 |
| 音频流(AS) | 发送音频采样率设定到音频采样时钟产生器，分离规定大小的音频样本，在必要时解包音频样本数据，且将音频样本路由到适当的音频样本 FIFO。 |
| 色彩映射(CM) | 读取色彩映射大小和偏移参数，且将色彩映射数据写至色彩映射存储器或存储位置。 |
| 反向链路封装(REL) | 促进在适当时间反向发送封包。检查反向链路旗标，且如果必要则发送显示器能力封包。亦适当地发送显示器请求和状态封包。 |
| 显示器能力(DC) | 当主机使用反向链路封装封包的反向链路旗标域来请求时，发送此类型封包。 |
| 键盘(K) | 如果存在所述封包且需要使用，则将这些封包传递至并传递自一个与键盘类型装置通讯的通用处理器。 |
| 指向装置(PD) | 如果存在所述封包且需要使用，则将这些封包传递至并传递自一个与指向类型装置通讯的通用处理器。 |
| 链路关闭(LS) | 记录链路关闭的事实并通知通用处理器。 |
| 显示器服务请求和状态(DSRS) | 将此封包作为反向链路封装封包中的第一个封包发送。 |
| 位块传送(BPT) | 解译封包参数，诸如视频数据格式描述符，确定首先移动哪些像素，且按要求移动位图中的像素。 |

[0383]

| | |
|----------------------------|--|
| 封包类型 | 封包处理器状态机响应 |
| 位图区域填充(BAF) | 解译封包参数，如果必要则经由色彩映射转译像素，且将像素数据写至位图中的适当位置。 |
| 位图图案填充(BPF) | 解译封包参数，如果必要则解包压缩的像素数据，如果必要则经由色彩映射转译像素，且将像素数据写至位图中的适当位置。 |
| 通讯链路通道(CLC) | 将此数据直接发送到通用处理器。 |
| 休眠期间的显示器服务请求(DSR) | 通用处理器控制着发送请求的低级功能且侦测争用，链路自身进行重新启动。 |
| 接口类型转递请求(ITHR)和接口类型确认(ITA) | 可将这些封包传递至并传递自通用处理器。接收此类型封包并形成具有一个确认的响应的逻辑大体上最小。因此，此操作也可实施于封包处理器状态机中。所导致的转递作为低级物理层动作进行且不太可能影响通用处理器的功能或运行。 |
| 执行类型转递(PTH) | 可直接按此等封包行动或通过将其传送到通用处理器来按此等封包行动，也命令硬件进行模式改变。 |

[0384] X. 减小反向链路数据率

[0385] 发明人已观察到可以特定方式调整或配置用于主机链路控制器的某些参数,以便实现非常理想的最大或更加优化(比例)的反向链路数据率。例如,在用于传送反向链路封装封包的反向数据封包域的时间期间,MDDI_Stb 讯号对双态切换以创造速率为正向链路数据率一半的周期数据时钟。此发生的原因是主机链路控制器以好像在发送全零一样的方式产生对应于 MDDI_Data0 讯号的 MDDI_Stb 讯号。MDDI_Stb 讯号被从主机传送到客户端,在客户端处其被用以产生一个用于从显示器传送反向链路数据的时钟讯号,通过所述时钟讯号,反向数据被发送回主机。图 50 展示了在采用 MDDI 的系统中的正向和反向路径上的讯号传送和处理中所遇到的典型延迟量的说明。在图 50 中,分别在用于 Stb+/- 产生阶段、电缆传送到显示器阶段、显示器接收器阶段、时钟产生阶段、讯号定时阶段、Data0+/- 产生阶段、电缆传送到主机阶段和主机接收器阶段的处理部分附近处展示了一系列延迟值 1.5 纳秒、8.0 纳秒、2.5 纳秒、2.0 纳秒、1.0 纳秒、1.5 纳秒、8.0 纳秒和 2.5 纳秒。

[0386] 取决于所遇到的正向链路数据率和讯号处理延迟,此“往返行程”效应或事件集合可能需要 MDDI_Stb 讯号上的一个循环以上的时间来完成,从而导致消耗了不当的时间或循环量。为防止此问题发生,反向速率除数(Reverse Rate Divisor)使得反向链路上的一个位时间可能跨越 MDDI_Stb 讯号的多个循环。这意味着反向链路数据率低于正向链路速率。

[0387] 应注意穿过接口的讯号延迟的实际长度可取决于所使用的每一具体主机-客户端系统或硬件的不同而不同。尽管并非必要,一般可通过使用往返行程延迟测量封包来测量系统中的实际延迟以便将反向速率除数设定到最优值来使每一个系统性能更好。

[0388] 通过使主机发送一个往返行程延迟测量封包给显示器来测量往返行程延迟。显示器通过在所述封包中的称为测量周期域的一个预先选定的测量窗口中或期间发送一序列 1 回到主机而回应于此封包。先前描述了此测量的详细时序。将往返行程延迟用于确定安全地采样反向链路数据所能达到的速率。

[0389] 往返行程延迟测量由确定、侦测或计数在测量周期域的开始和主机从客户端接收到 0xff、0xff、0x00 响应序列的时间周期的开始之间发生的正向链路数据时钟间隔数。应注意在测量计数开始增加之前的一小部分正向链路时钟周期时可接收来自客户端的响应。如果将此未修改的值用来计算反向速率除数,其可能由于不可靠的数据采样而在反向链路上造成位错误。图 51 中说明了此情况的一个实例,其中以图形形式说明了代表主机处的 MDDI_Data、主机处的 MDDI_Stb、主机内的正向链路数据时钟、及延迟计数的讯号。在图 51 中,在延迟计数将从 6 增加到 7 之前一小部分正向链路时钟周期时从显示器接收到响应序列。如果假定延迟为 6,则主机将刚好在位转换后或可能在位转换中间采样反向数据。这可在主机处导致错误的采样。由于此原因,在将所测量的延迟用于计算反向速率除数之前一般应将所述延迟加一。

[0390] 反向速率除数是主机在采样反向链路数据前应等待的 MDDI_Stb 循环数。由于 MDDI_Stb 的循环速率是正向链路速率的一半,所以经修正的往返行程延迟测量值需要除以 2 且接着四舍五入到下一整数。以公式表达,此关系为:

[0391]

$$\text{反向_速率_除数} = \text{四舍五入为下一整数} \left(\frac{\text{往返_行程_延迟} + 1}{2} \right)$$

[0392] 对于所给出的实例,此关系变为:

[0393]

$$\text{反向_速率_除数} = \text{四舍五入为下一整数} \left(\frac{6+1}{2} \right) = 4$$

[0394] 如果此实例中所用的往返行程延迟测量值是 7 而不是 6,那么反向速率除数也会等于 4。

[0395] 主机在反向链路时钟的上升缘采样反向链路数据。在主机和客户端(显示器)中都有一个计数器或类似已知电路或装置以产生反向链路时钟。将计数器初始化以使得反向链路时钟的第一个上升缘发生在反向链路封装封包的反向链路封包域中的第一个位的开始处。此就下文给出的实例而说明于图 52 中。计数器在 MDDI_Stb 讯号的每一个上升缘加一,且反向链路封装封包中的反向速率除数参数设定了直到计数绕回(wrap around)为止发生的计数次数。由于 MDDI_Stb 讯号以正向链路速率的一半双态切换,反向链路速率就是正向链路速率的一半除以反向速率除数。例如,如果正向链路速率为 200Mbps 且反向速率除数为 4,则反向链路数据率表达为:

$$[0396] \quad \frac{1}{2} \cdot \frac{200Mbps}{4} = 25Mbps$$

[0397] 图 52 中展示了一实例,其展示了反向链路封装封包中的 MDDI_Data0 和 MDDI_Stb 讯号线的时序,其中用于说明的封包参数具有以下值:

[0398] 封包长度 = 1024 (0x0400) 逆转 1 长度 = 1

[0399] 封包类型 = 65 (0x41) 逆转 2 长度 = 1

[0400] 反向链路旗标 = 0 反向速率除数 = 2

[0401] 参数 CRC = 0xdb4 3 全零为 0x00

[0402] 封包长度域和参数 CRC 域间的封包数据为:

[0403] 0x00、0x04、0x41、0x00、0x02、0x01、0x01、0x43、0xdb、0x00、...

[0404] 从显示器返回的第一个反向链路封包是显示器请求和状态封包,其封包长度为 7 且封包类型为 70。此封包以字节值 0x07、0x00、0x46、... 等等开始。然而,在图 52 中仅可看见第一个字节(0x07)。在图中,此第一反向链路封包时移了接近一个反向链路时钟周期,以说明实际反向链路延迟。以虚线展示了主机至显示器往返行程延迟为零的理想波形。

[0405] 先传送封包类型域,然后传送参数 CRC 域的 MS 字节,接着是全零域。来自主机的选通脉冲随着来自主机的数据改变电平而从 1 切换到 0 并切换回 1,从而形成更宽的脉冲。当数据变为 0 时,选通脉冲以更高速率切换,仅在数据线上的数据的变化才导致在接近对准域的末端处的变化。由于数据讯号在长时间中的固定的 0 或 1 电平,及脉冲模式(边缘)上的转换下降,选通脉冲在该图的剩余部分中以更高速率切换。

[0406] 用于主机的反向链路时钟为零直到逆转 1 周期的末端为止,此时时钟开始适应反向链路封包。图中下部部分中的箭头指示了何时采样数据,如从本揭示案的其余部分可了解的。正被传送的封包域的第一个字节(此处为 11000000)被展示为开始于逆转 1 之后,且线电平从主机驱动器被禁用以来已稳定。第一位的传递延迟(及针对 3 位所见)可从用于数据讯号的虚线看出。

[0407] 在图 53 中,可观察到基于正向链路数据率的反向速率除数的典型值。将实际反向

速率除数确定为往返行程链路测量的结果,以保证适当的反向链路操作。第一区域 5302 对应于一个安全操作区域,第二区域 5304 对应于一个边缘性能区域,而第三区域 5306 指示了不可能正常工作的设定。

[0408] 当在正向链路或反向链路上用任意接口类型设定操作时,往返行程延迟测量和反向速率除数设定是相同的,因为是根据实际时钟周期的单位而不是所传输或接收的位数目来表达其或按其操作的。

[0409] XI. 逆转和保护时间

[0410] 如上文所论述,反向链路封装封包中的逆转 1 域和往返行程延迟测量封包中的保护时间 1 域指定了在启用显示器接口驱动器前允许主机接口驱动器被禁用的时间长度值。逆转 2 域和保护时间 2 域提供了在启用主机驱动器前允许显示器驱动器被禁用的时间值。保护时间 1 域和保护时间 2 域一般填充有预设或预先选定的不希望调整的长度值。取决于所使用的接口硬件,可使用经验数据改进这些值,并在某些情况下加以调整,以改善操作。

[0411] 若干因素有助于确定逆转 1 的长度,且这些是正向链路数据率和主机中的 MDDI_Daa 驱动器的最大禁用时间。最大主机驱动器禁用时间规定于表 XI 中,在该表中展示了驱动器花费最大约 10 纳秒来禁用且花费约 2 纳秒来启用。禁用主机驱动器所需的最小正向链路时钟数根据以下关系式来表达:

[0412]

$$\text{禁用_所需_时钟}_{TA1} = \frac{\text{正向链路数据率}}{\text{接口类型因素}_{FWD}} \cdot \text{主机驱动器禁用延迟}_{\max}$$

[0413] 逆转 1 的允许值范围根据以下关系式来表达:

[0414]

$$\text{逆转_1} \geq \text{四舍五入至下一整数} \left(\frac{\text{禁用_所需_时钟}_{TA1} \cdot \text{接口类型因素}_{FWD}}{8} \right)$$

[0415] 其中对于 I 型来说接口类型因素为 1、对于 II 型来说为 2、对于 III 型来说为 4、且对于 IV 型来说为 8。

[0416] 组合以上两个等式,可发现接口类型因素项被消去,且逆转 1 定义为:

[0417]

$$\text{逆转_1} = \text{四舍五入至下一整数} \left(\frac{\text{正向链路数据率} \cdot \text{主机驱动器禁用延迟}_{\max}}{8} \right)$$

[0418] 例如,1500Mbps 的 III 型正向链路将使用的逆转 1 延迟为:

[0419]

$$\text{逆转_1} = \text{四舍五入至下一整数} \left(\frac{1500\text{Mbps} \cdot 10\text{nsec}}{8} \right) = 2 \text{ 字节}$$

[0420] 随着往返行程延迟增加,时序容限从主机被禁用的时间点提高到显示器被启用的时间。确定一般用于逆转 2 的时间长度的因素是正向链路数据率、显示器中的 MDDI_Data 驱动器的最大禁用时间、及通讯链路的往返行程延迟。对禁用显示器驱动器所需时间的计算基本上与上文论述的针对主机驱动器的计算相同,且根据以下关系式来定义:

[0421]

$$\text{禁用_所需_时钟}_{TA2} = \frac{\text{正向链路数据率}}{\text{接口类型因素}_{FWD}} \cdot \text{显示器驱动器禁用延迟}_{\max}$$

[0422] 且逆转 2 的允许值范围表达为：

[0423]

$$\text{逆转_2} \geq \text{四舍五入至下一整数} \left(\frac{\text{禁用_所需_时钟}_{TA2} + \text{往返_行程_延迟} + 1}{\left(\frac{8}{\text{接口类型因素}_{FWD}} \right)} \right)$$

[0424] 例如,往返行程延迟为 10 个正向链路时钟的 1500Mbps 的 III 型正向链路通常使用的逆转 2 延迟约为：

[0425]

$$\text{禁用_所需_时钟}_{TA2} = \frac{1500Mbps}{4} \cdot 10nsec = 3.75$$

[0426]

$$\text{逆转_2} \geq \text{四舍五入至下一整数} \left(\frac{3.75 + 10 + 1}{\left(\frac{8}{4} \right)} \right) = 8$$

[0427] XII. 替代性反向链路时序

[0428] 尽管使用上文论述的时序和保护间隔 (guard band) 有实现高数据传送率接口的作用,发明人已发现一种通过改变反向时序发现而允许比往返行程时间更短的反向位长度的技术。

[0429] 如上文所述,定时反向链路的前一方法经配置,使得从反向时序封包的保护时间 1 的最后一位开始计数时钟循环的数目直至第一个位在 IO 时钟的上升缘被采样。那就是用于定时 MDD 接口的输入和输出的时钟讯号。反向速率除数的计算由下式给出：

[0430]

$$\text{反向_速率_除数} = \text{四舍五入至下一整数} \left(\frac{\text{往返_行程_延迟} + 1}{2} \right)$$

[0431] 这提供了等于往返行程延迟的位宽度,从而导致非常可靠的反向链路。然而,反向链路已显示出有能力运行地更快(或以更高数据传送率运行),这正是发明人想加以利用的。一种新的发明性技术允许利用接口的额外能力以达到更高的速度。

[0432] 这是通过使主机计数时钟循环数直到采样到一个 1,但同时主机是在反向时序封包期间的上升缘和下降缘两者上采样数据线来实现的。这允许主机挑选反向位中的最有用或甚至是最优的采样点以确保所述位为稳定的。也就是说,为反向讯务反向封装封包找到采样数据的最有用或最优上升缘。最优采样点取决于反向链路除数和第一个 1 是在上升缘上还是在下降缘上被侦测到的。新的时序方法允许主机只寻找客户端为反向链路时序发送的 0xFF 0xFF 0x00 模式的第一边缘,以确定在反向封装封包中的何处采样。

[0433] 图 64 中说明了到达的反向位和所述位将如何寻找各种反向速率除数的实例,以及自从保护时间 1 的最后一位开始已发生的若干时钟循环。在图 64 中,对于反向速率除数为 1 的最优采样点,可发现如果第一边缘出现在上升缘和下降缘(标为上升/下降)之间,

最优采样点是标为“b”的时钟循环边缘,因为它是出现在反向位的周期中的唯一上升缘。对于为 2 的反向速率除数,最优采样点可能仍是时钟循环前缘“b”,因为循环边缘“c”比“b”更接近位边缘。对于为 4 的反向速率除数,最优采样点可能是时钟循环边缘“d”,因为它更接近反向位的后缘(在该处值可能已稳定化)。

[0434] 返回图 64,然而,如果第一边缘出现在下降缘和上升缘(标为下降/上升)之间,那么反向速率除数为 1 时的最优采样点是采样点时钟循环边缘“a”,因为它是反向位时间周期中唯一的上升缘。对于为 2 的反向速率除数,最优采样点为边缘“b”,且对于为 4 的反向速率除数,最优采样点为边缘“c”。

[0435] 可看出随着反向速率除数变得越来越大,最优采样点变得更易于确定或选择,因为它应是最接近中间的上升缘。

[0436] 主机可使用这一技术来在时序封包数据的上升数据边缘在数据线上被观察到之前发现上升时钟边缘的数目。其接着可基于边缘是出现在上升与下降缘之间还是出现在下降与上升缘之间和反向速率除数是什么来决定将多少额外的时钟循环添加到数字计数器,以便合理地确保总是尽可能接近中间地采样所述位。

[0437] 一旦主机已选择或确定了时钟循环的数目,其可与客户端“考察”各种反向速率除数,以确定特定反向速率除数是否会起作用。主机(和客户端)可以除数 1 开始,并校验从客户端接收的反向状态封包的 CRC 以确定此反向速率是否可正确工作以传送数据。如果 CRC 是坏的,就可能存在采样错误,且主机可增加反向速率除数并试图再次请求状态封包。如果第二个所请求的封包是坏的,可再次增加除数并再次发出请求。如果此封包被正确地解码,就可将此反向速率除数用于所有未来的反向封包。

[0438] 此方法有效且有用,因为反向时序不应从最初往返行程时序估计值变化。如果正向链路稳定,即使存在反向链路错误,客户端仍应继续解码正向链路封包。当然,主机仍然有责任设定链路的反向链路除数,因为此方法不保证有完美的反向链路。此外,除数将主要取决于用于产生 IO 时钟的时钟的质量。如果该时钟具有大量不稳定性,出现采样错误的可能性就更大。此错误可能性随着往返行程延迟中的时钟循环的量增加而增加。

[0439] 此实施表现得对 I 型反向数据效果最好,但由于数据线之间的偏斜可能太大而不能以对仅仅一个数据对效果最好的速率运行链路,此实施可对 II 型到 IV 型反向数据造成问题。但是,甚至是在将 II 型到 IV 型用于操作时,数据率可能不必减少到先前的方法中的情况。如果在每一数据线上重复以选择理想或最优的时钟采样位置,此方法也可起到最好的作用。如果它们在用于每一数据对的相同采样时间下,此方法将继续起作用。如果它们在不同采样周期下,可使用两个不同办法。第一个办法是选择每一数据点的理想或更优化的采样位置,即使其对于每一数据对并不相同。主机接着可在从数据对的集合中采样了所有位后重构数据流:II 型时两位、III 型时四位、IV 型时八位。另一选项是,主机增加反向速率除数,使得可在相同时钟边缘采样每个数据对的数据位。

[0440] XIII. 链路延迟和偏斜的影响

[0441] 正向链路上的在 MDDI_Data 对和 MDDI_Stb 之间的延迟偏斜可限制最大可能的数据率,除非使用了延迟偏斜补偿。导致时序偏斜的延迟方面的差异是由于控制器逻辑、线驱动器和接收器、及电缆和连接器,如下文概述。

[0442] A. 受偏斜限制的链路时序分析(MDDI I 型)

[0443] 1. I 型链路的延迟和偏斜实例

[0444] 图 57 中展示了类似于图 41 所示电路的典型接口电路,用于适应 I 型接口链路。在图 57 中,针对 MDDI I 型正向链路的若干处理或接口阶段中的每一者展示了传播延迟和偏斜的例示性或典型值。MDDI_Stb 和 MDDI_Data0 之间的延迟中的偏斜会导致输出时钟的工作循环 (duty-cycle) 失真。在使用触发器 5728、5732 的接收器触发器 (RXFF) 级的 D 输入端处的数据必须在时钟边缘后略微改变,以便能可靠地采样其。该图展示了两个级联的延迟线 5732a 和 5732b,其被用于解决与建立此时序关系有关的两个不同问题。在实际实施中,这些延迟线可组合为单个延迟元件。

[0445] 图 58 中说明了 I 型链路上的用于穿过所述接口的例示性讯号处理的数据、Stb、和时钟恢复时序。

[0446] 相当大的总延迟偏斜大体上源自于以下各级中的偏斜的总和:具有触发器 5704、5706 的传输器触发器 (TXFF);具有驱动器 5708、5710 的传输器驱动器 (TXDRVR);电缆 5702;具有接收器 5722、5724 的接收器线接收器 (RXRCVR);及接收器异或逻辑 (RXXOR)。延迟 15732a 应匹配或超过 RXXOR 级中的异或门 5736 的延迟,其由以下关系式确定:

$$[0447] \quad t_{PD-min}(\text{延迟 } 1) \geq t_{PD-max}(XOR)$$

[0448] 需要满足此要求,以使得接收器触发器 5728、5732 的 D 输入不在其时钟输入之前改变。此在 RXFF 的保持时间为零的情况下有效。

[0449] 延迟 2 的目的或功能是根据以下关系式补偿 RXFF 触发器的保持时间:

$$[0450] \quad t_{PD-min}(\text{延迟 } 2) = t_H(RXFF)$$

[0451] 这在许多系统中将为零,因为保持时间为零,在那种情况下延迟 2 的最大延迟当然也可为零。

[0452] 对接收器异或级中的偏斜的最坏情况影响 (worst-case contribution) 发生在数据脉冲迟 / 选通脉冲早 (data-late/strobe-early) 的情况下,其中延迟 1 处在最大值且根据以下关系式来自异或门的时钟输出尽可能早地到达:

$$[0453] \quad t_{\text{偏斜-max}}(RXXOR) = t_{PD-max}(\text{延迟 } 1) - t_{PD-min}(XOR)$$

[0454] 在此情况下,数据可在两个位周期, n 和 n+1, 之间变化,非常接近于 n+1 位被时钟处理 (clocked) 入接收器触发器的时间。

[0455] MDDI I 型链路的最大数据率 (最小位周期) 是在 MDDI 链路中的所有驱动器、电缆和接收器中遇到的最大偏斜加上到 RXFF 级的总数据设定的函数。在直到 RXRCVR 级的输出端的链路中的总延迟偏斜可表示为:

$$[0456] \quad t_{\text{偏斜-max}}(\text{链路}) = t_{\text{偏斜-max}}(TXFF) + t_{\text{偏斜-max}}(TXDRVR) + t_{\text{偏斜-max}}(\text{电缆}) + t_{\text{偏斜-max}}(RXRCVR)$$

[0457] 且最小位周期由下式给出:

$$[0458] \quad t_{\text{位-min}} = t_{\text{偏斜-max}}(\text{链路}) + t_{\text{偏斜-max}}(RXXOR) + t_{PD-max}(\text{延迟 } 2) + t_{SU}(RXFF)$$

[0459] 在图 57 所示实例中, $t_{SKEW-max}(\text{LINK}) = 1.4$ 纳秒且最小位周期可表示为:

$$[0460] \quad t_{\text{位-min}} = 1.4 + 0.3 + 0.2 + 0.5 = 2.4 \text{ nsec}, \text{ 或陈述为约 } 416 \text{ Mbps}.$$

[0461] B. MDDI II 型、III 型和 IV 型的链路时序分析

[0462] 图 59 展示了类似于图 41 和 57 中所示者的典型接口电路,用于适应 II 型、III 型和 IV 型接口链路。在 TXFF (5904) 级、TXDRVR (5908) 级、RXRCVCR (5922) 级和 RXFF (5932、5928、5930) 级中使用了额外元件以适应额外的讯号处理。在图 59 中,针对 MDDIII 型正向

链路的若干处理或接口阶段中的每一者展示了传播延迟和偏斜的例示性或典型值。除了会影响输出时钟的工作循环的在 MDDI_Stb 和 MDDI-Data0 之间的延迟中的偏斜外,还有这两个讯号和其他 MDDI_Data 讯号之间的偏斜。在由触发器 5928 和 5930 组成的接收器触发器 B (RXFFB) 级的 D 输入端处的数据在时钟边缘后被略微改变,以便能可靠地采样其。如果 MDDI-Data1 比 MDDI_Stb 或 MDDI-Data0 到达地更早,则 MDDI-Data1 应被延迟至少该延迟偏斜的量以被采样。为实现此,使用延迟 3 延迟线来延迟数据。如果 MDDI-Data1 比 MDDI_Stb 和 MDDI-Data0 到达地更晚且其又被延迟 3 所延迟,则 MDDI-Data1 改变的时点移近下一时钟边缘。此过程确定了 II 型、III 型、或 IV 型链路的数据率的上限。图 60A、60B 和 60C 中说明了两个资料讯号和 MDDI_Stb 相对于彼此的时序或偏斜关系的一些不同的例示性可能性。

[0463] 为了在 MDDI_DataX 尽可能早地到达时在 RXFFB 中可靠地采样数据,根据以下关系式设定延迟 3:

$$[0464] \quad t_{PD-min(\text{延迟 } 3)} \geq t_{\text{偏斜-max(链路)}} + t_{H(RXFFB)} + t_{PD-max(XOR)}$$

[0465] 最大链路速度由最小可允许的位周期判定。此在 MDDI_DataX 尽可能迟地到达时受到的影响最大。在那种情况下,最小可允许循环时间由下式给定:

$$[0466] \quad t_{\text{位-min}} = t_{\text{偏斜-max(链路)}} + t_{PD-max(\text{延迟 } 3)} + t_{SU(PXFFB)} + t_{PD-min(XOR)}$$

[0467] 链路速度的上限就为:

$$[0468] \quad t_{PD-max(\text{延迟 } 3)} = t_{PD-min(\text{延迟 } 3)}$$

[0469] 且假定:

$$[0470] \quad t_{\text{位-min(lower-bound)}} = 2 \cdot t_{\text{SKEW-max(链接)}} + t_{PD-max(XOR)} + t_{SU(RXFFB)} + t_{H(RXFFB)}$$

[0471] 在上文提供的实例中,最小位周期的下限由以下关系式给出:

$$[0472] \quad t_{\text{位-min(lower-level)}} = 2 \cdot 1.4 + 1.5 + 0.5 + 0.1 = 4.8 \text{ n sec, 其约为 } 208 \text{ Mbps.}$$

[0473] 这比 I 型链路中可使用的最大数据率慢得多。MDDI 的自动延迟偏斜补偿能力显著减少了延迟偏斜对最大链路速率的影响。

[0474] XIV. 物理层互连描述

[0475] 适用于实施根据本发明之接口的物理连接可使用可购得的零件来实现,诸如在主机侧使用 Hirose Electric Company Ltd. 制造的零件号 3260-8S2(01),而在显示装置侧使用 Hirose Electric Company Ltd. 制造的零件号 3240-8P-C。表 XIII 中列出了此等配合 I 型 / II 型接口使用的连接器的例示性接口引脚配置或“引出线”,且其说明于图 61 中。

[0476] 表 XIII

[0477]

| 讯号名称 | 引脚数 | 颜色 | 讯号名称 | 引脚数 | 颜色 |
|-------------|-----|----|-------------|-----|------|
| MDDI_Pwr | 1 | 红色 | MDDI_Gnd | 2 | 黑红成对 |
| MDDI_Stb+ | 3 | 绿色 | MDDI_Stb- | 4 | 黑绿成对 |
| MDDI_Data0+ | 5 | 蓝色 | MDDI_Data0- | 6 | 黑蓝成对 |
| MDDI_Data1+ | 7 | 白色 | MDDI_Data1- | 8 | 黑白成对 |
| | | | 屏蔽物 | | |

[0478] 屏蔽物连接到主机接口中的 MDDI_Gnd,且电缆中的屏蔽排扰线连接到显示器连接器的屏蔽物。然而,屏蔽物和排扰线不连接到显示器内的电路接地端。

[0479] 互连元件或装置经选择或设计以便可足够小以配合移动通讯和计算装置(诸如 PDA 和无线电话)或便携式游戏装置使用,而不会与装置的相对大小相比显得突兀或不美

观。任何连接器和接线应足够耐用以用于典型消费性环境中,且允许有小尺寸(尤其是接线)和较低的成本。传送元件应适应数据和选通讯号,所述讯号是在 I 型和 II 型的情况下传送率高达约 450Mbps 且在 8 位平行 IV 型型式下传送率高达 3.6Gbps 的差动 NRZ 数据。

[0480] XV. 操作

[0481] 图 54A 和 54B 中展示了在使用本发明的实施例的接口的操作期间在处理数据和封包时采取的大体步骤的概述,及图 55 中展示了处理所述封包的接口设备的概观。在这些图中,过程开始于步骤 5402 中,其中确定客户端和主机是否用一通讯路径(此处为一电缆)相连。这可通过主机使用可侦测连接器或电缆或到主机的输入端处的讯号(诸如在 USB 接口中所见)的存在的软件或硬件或其他已知技术来周期性轮询而进行。如果没有客户端连接到主机,则主机可只是进入某预定长度(取决于应用)的等待状态,进入休眠模式或停止活动以等待未来的使用(可要求用户采取行动重新激活主机)。例如,当主机驻存在电脑型装置中时,用户可能必须点击屏幕图标或请求一个程序,从而激活主机处理以寻找客户端。而且,取决于主机或驻留的主机软件的能力和配置,简单地插入 USB 型连接(诸如用于 U 型接口者)可激活主机处理。

[0482] 一旦客户端连接到主机或反之或客户端被侦测到,在步骤 5404 和 5406 中,客户端抑或主机发送适当的封包请求服务。在步骤 5404 中,客户端可发送显示器服务请求或状态封包。应注意,如上文所论述,链路可能在先前已被关闭或处于休眠模式,所以这可不是接下来的对通讯链路的完全初始化。一旦通讯链路被同步且主机试图与客户端通讯,客户端也提供一个显示器能力封包给主机,如步骤 5408 中。主机现在可开始确定客户端可适应的支持的类型,包括传送速率。

[0483] 大体上,在步骤 5410 中,主机和客户端也协商将使用的服务模式的类型(速率/速度),例如, I 型、U 型、II 型等等。在步骤 5410 中,一旦确立了服务类型,主机可开始传送信息。此外,主机可使用往返行程延迟测量封包来最优化与其他讯号处理并行的通讯链路的时序,如步骤 5411 所示。

[0484] 如上文所述,所有传送以子帧标头封包开始,如图示其在步骤 5412 中被传送,接着是数据类型(此处为视频和音频流封包)和填充符封包,如图示其在步骤 5414 中被传送。音频和视频数据应在先前已被预备或映射入封包中,且按必要或需要插入填充符封包以填满媒体帧的所要求数目的位。主机可发送诸如正向音频通道启用封包的封包来启动音响装置。此外,主机可使用上文论述的其它封包类型来传送命令或信息,此处展示为在步骤 5416 中传送色彩映射封包、位块传送封包或其它封包。此外,主机和客户端可使用适当封包来互换与键盘或指向装置有关的数据。

[0485] 在操作期间,可能发生若干会导致主机或客户端需采用不同数据率或接口类型模式的不同事件之一。例如,电脑或其它通讯数据的装置可在处理数据过程中遇到装入的情况,从而在预备或呈现封包的过程中造成减慢。一台接收数据的显示器可能从专用 AC 电源变为用更有限的电池电源供电,并且在更有限的供电设定下不能同样快地传送入数据、不能同样快地处理命令、抑或不能使用相同的分辨率或色深。或者,限制性条件可能缓和或消失,从而允许任一装置以更高速率传送数据。这是更期望的,可发出请求以改变到更高传送率模式下。

[0486] 如果这些或其它类型的已知条件发生或改变,主机抑或客户端可侦测到它们并试

图就接口模式进行协商。这展示于步骤 5420 中,其中主机发送接口类型转递请求封包给客户端,请求转递至另一模式,客户端发送接口类型确认封包以确认已找到一改变,且主机发送执行类型转递封包以改变到规定模式。

[0487] 尽管不要求特定处理顺序,客户端和主机也可互换封包,所述封包是关于以指向装置、键盘或主要与客户端相关联的其他用户型输入装置(尽管此等元件也可存在于主机侧)为目的地的数据或从所述装置接收的数据。这些封包通常用通用处理器型元件而不是状态机加以处理(5502)。另外,上文论述的某些命令也将用通用处理器来处理。(5504、5508)

[0488] 在主机和客户端之间互换了数据和命令后,在某时点处决定是否要传送额外的数据或主机或客户端是否要停止向传送提供服务。这展示于步骤 5422 中。如果链路将进入休眠状态或被完全关闭,则主机发送链路关闭封包给客户端,且两方都终止数据传送。

[0489] 将使用先前关于主机和客户端控制器所论述的驱动器和接收器来传送在上文操作处理中所传送的封包。这些线驱动器和其它逻辑元件连接到上文论述的状态机和通用处理器,如图 55 的概观所说明。在图 55 中,状态机 5502 和通用处理器 5504 和 5508 可进一步连接到其它未图示的元件,诸如专用 USB 接口、存储器元件、或驻存在所述状态机和通用处理器借以互动的链路控制器外部的其他组件,包括(但不限于)数据源,以及用于显示装置的视频控制芯片。

[0490] 处理器和状态机提供与保护时间等等有关的对如上文所论述的驱动器的启用和禁用的控制,以便确保能高效率地建立或终止通讯链路及传送封包。

[0491] XVI. 附录

[0492] 除了用于实施本发明的实施例的体系结构和协议的各种封包的上述格式、结构及内容外,此处提供某些封包类型的更详细的域内容或操作。此处提供这些内容或操作以进一步阐明其各别用途或操作,以便使所属领域的技术人员可更容易地理解本发明并将其用于各种应用中。此处进一步论述尚未论及的少数域。另外,使用与上文提供的实施例有关的例示性定义和值来提供这些域。但是,此等值不应被当作是对本发明的限制,而是代表了适用于实施所述接口和协议的一个或一个以上实施例,且并非所有实施例都需要一起或同时实行。如所属领域的技术人员所了解,在其它实施例中可使用其它值以实现数据或数据率传送结果的所要表述。

[0493] A. 对于视频流封包

[0494] 在一个实施例中,显示器属性域(1 字节)具有一系列位值,其解释如下。1 位和 0 位选择如何投送显示器像素数据。位值为“00”或“11”时,为双眼显示数据,位值为“10”时,仅将数据投送到左眼,且位值为“01”时,仅将数据投送到右眼。2 位指示像素数据是否以隔行格式提供的,值为“0”时意味着像素数据为标准的逐次格式,且当从一行前进到下一行时行号(像素 Y 座标)加 1。当此位的值为“1”时,像素数据为隔行格式,且当从一行前进到下一行时行号加 2。3 位指示像素数据处于交替像素格式。此类似于由 2 位启用的标准的隔行模式,但隔行操作是垂直的而非水平的。当 3 位为 0 时,像素数据为标准的逐次格式,且随着每一连续的像素被接收到,列号(像素 X 座标)加 1。当 3 位为 1 时,像素数据为交替像素格式,且随着接收到每一像素,列号加 2。7 至 4 位被保留用于未来的用途且一般设为零。

[0495] 2字节的 X 起始域和 Y 起始域规定了像素数据域中的第一个像素的点 (X 起始、Y 起始) 的绝对 X 和 Y 座标。2字节的 X 左边缘域和 Y 顶边缘域规定了由像素数据域填充的屏幕窗口的左边缘的 X 座标和顶边缘的 Y 座标, 而 X 右边缘域和 Y 底边缘域规定了正被更新的窗口的右边缘的 X 座标和底边缘的 Y 座标。

[0496] 像素计数域 (2 字节) 规定了下述像素数据域中的像素数目。

[0497] 参数 CRC 域 (2 字节) 包含从封包长度域到像素计数域的所有字节的 CRC。如果此 CRC 未通过校验, 则将整个封包丢弃。

[0498] 像素数据域包含将被显示的原始视频信息, 且所述信息以视频数据格式描述符域描述的方式受到格式化。如本文其它地方所论述, 数据被一次一“行”地传输。

[0499] 像素数据 CRC 域 (2 字节) 仅包含像素数据域的 16 位 CRC。如果此值的 CRC 验证失败, 则仍可使用所述像素数据域, 但将 CRC 错误计数加 1。

[0500] B. 对于音频流封包

[0501] 在一个实施例中, 音频通道 ID 域 (1 字节) 识别了客户端装置发送的音频数据所要前往的特定音频通道。物理音频通道在此域中规定为或被此域映射为值 0、1、2、3、4、5、6 或 7, 其分别指示左前、右前、左后、右后、正中、亚低音扬声器 (sub-woofer)、环绕声左、环绕声右通道。为 254 的音频通道 ID 值指示单一数字音频样本流被发送到左前和右前通道。这简化了将立体声耳机用于语音通讯的应用、将生产率增强应用程序 (productivity enhancement apps) 用于 PDA 上的应用或其它其中简单的用户接口产生报警音的应用。

[0502] ID 域的值在 8 至 253 之间变动, 且当前将 255 保留用于其中新设计需要额外的标志的情况下。

[0503] 音频采样计数域 (2 字节) 规定了此封包中的音频样本的数目。

[0504] 每样本位和压缩域包含 1 字节, 其规定了音频数据的压缩格式。一般采用的格式是 4 至 0 位定义每 PCM 音频样本的位数。5 位则规定数字音频数据样本是否被压缩。如上文所述, 图 12 说明了压缩的音频样本和字节对准的音频样本之间的差异。5 位的值为“0”指示了数字音频数据域中的每一 PCM 音频样本与接口字节边界字节对准, 且值为“1”指示了每一连续的 PCM 音频样本是与前一音频样本压缩到一起的。此位仅在 4 至 0 位中所定义的值 (每 PCM 音频样本的位数) 不是 8 的倍数时有效。7 至 6 位被保留用于其中系统设计需要额外的标志的情况下, 且一般设为零值。

[0505] 音频采样率域 (1 字节) 规定了音频 PCM 采样率。所采用的格式是, 值为 0 指示每秒 8,000 个样本 (sps) 的速率, 值为 1 指示 16,000sps, 值为 2 指示 24,000sps, 值为 3 指示 32,000sps, 值为 4 指示 40,000sps, 值为 5 指示 48,000sps, 值为 6 指示 11,025sps, 值为 7 指示 22,050sps 且值为 8 指示 44,100sps, 同时将值 9 至 15 保留用于未来用途, 所以它们当前设为零。

[0506] 参数 CRC 域 (2 字节) 包含从封包长度域到音频采样率域的所有字节的 16 位 CRC。如果此 CRC 未正确通过校验, 则将整个封包丢弃。数字音频数据域包含将要播放的原始音频样本, 且通常形式为如无符号整数的线性格式。音频数据 CRC 域 (2 字节) 仅含有音频数据的 16 位 CRC。如果此 CRC 未通过校验, 则仍可使用音频数据域, 但 CRC 错误计数加 1。

[0507] C. 对于用户定义流封包

[0508] 在一个实施例中,将 2 字节的流 ID 号域用于识别特定的用户定义流。流参数域和流数据域的内容通常由 MDDI 设备制造商定义。2 字节的流参数 CRC 域包含从封包长度域开始到音频编码字节的流参数的所有字节的 16 位 CRC。如果此 CRC 未通过校验,则丢弃整个封包。2 字节的流数据 CRC 域仅含有流数据域的 CRC。如果此 CRC 未正确通过校验,则取决于该应用的要求,此流数据域的使用与否是可选的。使用流数据的条件是 CRC 良好,一般要求将流数据存储于缓冲区中直到确认 CRC 良好。如果 CRC 未通过校验,则 CRC 错误计数加 1。

[0509] D. 对于色彩映射封包

[0510] 色彩映射数据大小域(2 字节)规定了存在于此封包中的色彩映射数据域中的色彩映射表条目的总数。在此实施例中,色彩映射数据域中的字节数是色彩映射大小的 3 倍。色彩映射大小被设定为等于零以便不发送色彩映射数据。如果色彩映射大小为零,则一般仍发送色彩映射偏移值,但其被显示器忽略。色彩映射偏移域(2 字节)规定了此封包中的色彩映射数据相对于显示装置中的色彩映射表的开始的偏移。

[0511] 2 字节的参数 CRC 域包含从封包长度域到音频编码字节的所有字节的 CRC。如果此 CRC 未通过校验,则丢弃整个封包。

[0512] 对于色彩映射数据域,每一色彩映射位置为一个 3 字节值,其中第一个字节规定蓝色的量值,第二个字节规定绿色的量值且第三个字节规定红色的量值。色彩映射大小域规定了存在于色彩映射数据域中的 3 字节色彩映射表项的数目。如果不能将单个色彩映射装入一个视频数据格式和色彩映射封包,则可通过发送多个各具有不同色彩映射数据和色彩映射偏移的封包来规定整个色彩映射。

[0513] 2 字节的色彩映射数据 CRC 域仅包含色彩映射数据的 CRC。如果此 CRC 未通过校验,则色彩映射数据仍可使用,但 CRC 错误计数加 1。

[0514] E. 对于反向链路封装封包

[0515] 在一个实施例中,反向链路旗标域(1 字节)包含一组旗标以便从显示器请求信息。如果一个位(例如,0 位)被设为 1,则主机使用显示器能力封包从显示器请求此规定信息。如果所述位为 0,则主机不需要所述来自显示器的信息。将剩余的位(此处为 1 至 7 位)保留用于未来用途并设为零。但是,按需要可使用更多的位来设定用于反向链路的旗标。

[0516] 反向速率除数域(1 字节)规定了关于反向链路数据时钟所发生的 MDDI_Stb 循环的数目。反向链路数据时钟等于正向链路数据时钟除以反向速率除数的两倍。反向链路数据率与反向链路上的反向链路数据时钟和接口类型有关。对于 I 型接口,反向数据率等于反向链路数据时钟,对于 II 型、III 型和 IV 型接口,反向数据率分别等于反向链路数据时钟的两倍、四倍和八倍。

[0517] 逆转 1 长度域(1 字节)规定了为逆转 1 分配的总字节数。推荐的逆转 1 长度是主机中的 MDDI_Data 驱动器禁用输出端所需的字节数。此基于上述输出禁用时间、正向链路数据率、及所选择的正向链路接口类型。上文提供了设定逆转 1 的更完整说明。

[0518] 逆转 2 长度域(1 字节)规定了为逆转(Turn-Around)分配的总字节数。推荐的逆转 2 长度是显示器中的 MDDI_Data 驱动器禁用其输出端所需的字节数加上往返行程延迟。上文提供了设定逆转 2 的说明。

[0519] 参数 CRC 域 (2 字节) 包含从封包长度域到逆转长度域的所有字节的 16 位 CRC。如果此 CRC 未通过校验,则丢弃整个封包。

[0520] 全零域 (1 字节) 被设定为等于 0,且用于确保在第一保护时间周期期间禁用线驱动器之前所有 MDDI_Data 讯号处于零状态。

[0521] 逆转 1 域用于建立第一逆转周期。此域分配由逆转长度参数所规定的数目的字节以允许在启用客户端 (显示器) 中的线驱动器之前禁用主机中的 MDDI_Data 线驱动器。主机在逆转 1 的 0 位期间禁用其 MDDI_Data 线驱动器,且客户端 (显示器) 在逆转 1 的最后一位后立刻启用其线驱动器。MDDI_Stb 讯号以好像逆转周期为全零一般的方式动作。

[0522] 反向数据封包域包含从客户端传送到主机的一系列数据封包。如上文所述,发送填充符封包以填充其它封包类型未使用的剩余空间。

[0523] 将逆转 2 域用于建立第二逆转周期。此域分配由逆转长度参数所规定的数目的字节。

[0524] 驱动器重启用域使用等于零的 1 字节来确保在下一封包的封包长度域之前重启用所有 MDDI_Data 讯号。

[0525] F. 对于显示器能力封包

[0526] 在一个实施例中,协议版本域使用 2 个字节来规定客户端使用的协议版本。将初始版本设定为零,而最小协议版本域使用 2 个字节来规定客户端可使用或解译的最小协议版本。显示器数据率能力域 (2 字节) 规定显示器在接口的正向链路上可接收的最大数据率,且以百万位秒 (Mbps) 的形式来规定。接口类型能力域 (1 字节) 规定在正向和反向链路上可支持的接口类型。这在当前通过以下方式来指示,选择 0 位、1 位或 2 位来分别选择正向链路上的 II 型、III 型或 IV 型模式,且选择 3 位、4 位或 5 位来分别选择反向链路上的 II 型、III 型或 IV 型模式;同时保留 6 位和 7 位并将其设为零。位图宽度和高度域 (2 字节) 以像素为单位规定了位图的宽度和高度。

[0527] 单色能力域 (1 字节) 用于规定可以单色格式显示的分辨率的位数。如果显示器不能使用单色格式,则此值设为零。将 7 至 4 位保留用于未来用途且因此设为零。3 至 0 位定义了对于每一像素可存在的灰度级的最大位数。这 4 个位使得可能规定每一像素的 1 到 15 的值。如果值为零,则所述显示器不支持单色格式。

[0528] 色彩映射能力域 (3 字节) 规定了存在于显示器中的色彩映射表中的表项的最大数目。如果显示器不能使用色彩映射格式,则此值为零。

[0529] RGB 能力域 (2 字节) 规定了可以 RGB 格式显示的分辨率的位数。如果显示器不能使用 RGB 格式,则此值等于零。RGB 能力字由三个独立的无符号值构成,其中:在每一像素中,3 至 0 位定义蓝色的最大位数,7 至 4 位定义绿色的最大位数,且 11 至 8 位定义红色的最大位数。当前,将 15 至 12 位保留用于未来用途,且一般将其设为零。

[0530] Y Cr Cb 能力域 (2 字节) 规定可以 Y Cr Cb 格式显示的分辨率的位数。如果显示器不能使用 Y Cr Cb 格式,则将此值设为零。Y Cr Cb 能力字由三个独立的无符号值构成,其中:3 至 0 位定义 Cb 样本中的最大位数,7 至 4 位定义 Cr 样本中的最大位数,11 至 8 位定义 Y 样本中的最大位数,且当前将 15 至 12 位保留用于未来用途,且一般将其设为零。

[0531] 显示特性能力指示符域使用 4 个字节,其包含一组指示显示器中支持的具体特性的旗标。一个位设定为 1 就指示支持该能力,且一个位设定为 0 就指示不支持该能力。0 位

的值指示是否支持位图块传送封包（封包类型 71）。1 位、2 位和 3 位的值分别指示是否支持位图区域填充封包（封包类型 72）、位图图案填充封包（封包类型 73）或通讯链路数据通道封包（封包类型 74）。4 位的值指示显示器是否有能力使一种色彩透明，而 5 位和 6 位的值分别指示显示器是否可接受压缩格式的视频数据或音频数据，且 7 位的值指示显示器是否可发送来自一摄像机的反向链路视频流。11 位和 12 位的值分别指示了何时客户端与指向装置通讯并可发送和接收指向装置数据封包，及何时客户端与键盘通讯并可发送和接收键盘数据封包。当前将 13 至 31 位保留用于未来用途或对于系统设计者有用的替代标志，且一般将其设为零。

[0532] 显示器视频帧速率能力域（1 字节）规定了以帧每秒为单位的显示器的最大视频帧更新能力。主机可选择以低于此域中规定的值的速率来更新影像。

[0533] 音频缓冲深度域（2 字节）规定了显示器中的专用于每一音频流的弹性缓冲区的深度。

[0534] 音频通道能力域（2 字节）包含一组旗标，其指示哪些音频通道受到显示器（客户端）的支持。设为 1 的位指示该通道受到支持，而设为 0 的位指示该通道不受支持。所述位位置被指派给不同通道，例如，位位置 0、1、2、3、4、5、6 和 7 分别指示了左前、右前、左后、右后、正中、亚低音扬声器、环绕声左和环绕声右通道。当前将 8 至 15 位保留用于未来用途，且一般将其设为零。

[0535] 用于正向链路的 2 字节的音频采样率能力域包含一组旗标，以指示客户端装置的音频采样率能力。相应地将位位置指派给不同速率，诸如分别将 0、1、2、3、4、5、6、7 和 8 位指派给 8,000、16,000、24,000、32,000、40,000、48,000、11,025、22,050 和 44,100 样本每秒（SPS），同时将 9 至 15 位保留用于未来用途或按需要用于替代速率用途，所以当前将其设为“0”。将这些位中之一的位值设定为“1”指示了支持该特定采样率，且将所述位设为“0”就指示不支持所述采样率。

[0536] 最小子帧速率域（2 字节）规定了以帧每秒为单位的最小子帧速率。所述最小子帧速率使显示器状态更新速率充分大而足以读取显示器中的某些感应器或指向装置。

[0537] 用于反向链路的 2 字节的麦克风采样率能力域包含一组旗标，其指示客户端装置中的麦克风的音频采样率能力。用于 MDDI 的目的时，将客户端装置麦克风配置成最少支持至少 8,000 样本每秒的速率。将此域的位位置指派给不同速率，其中位位置 0、1、2、3、4、5、6、7 和 8（例如）分别用于代表 8,000、16,000、24,000、32,000、40,000、48,000、11,025、22,050 和 44,100 样本每秒（SPS），同时将 9 至 15 位保留用于未来用途或按需要用于替代速率用途，所以当前将其设为“0”。将这些位中之一的位值设定为“1”指示了支持该特定采样率，且将所述位设为“0”就指示不支持所述采样率。如果未连接麦克风，则将每一个麦克风采样率能力位设为零。

[0538] 内容保护类型域（2 字节）包含一组旗标，其指示显示器支持的数字内容保护的类型。当前，将位位置 0 用于指示何时 DTCP 受到支持，且将位位置 1 用于指示何时 HDCP 受到支持，同时将位位置 2 至 15 保留用于需要或可用的其他保护机制，所以当前将其设为零。

[0539] G. 对于显示器请求和状态封包

[0540] 反向链路请求域（3 字节）规定了显示器在反向链路中在下一子帧中发送信息到主机所需要的字节数。

[0541] CRC 错误计数域 (1 字节) 指示自从媒体帧的开始起发生了多少 CRC 错误。当发送了子帧计数为零的子帧包头封包时将 CRC 计数复位。如果 CRC 错误的实际数目超过 255, 则此值一般在 255 饱和。

[0542] 能力变化域使用 1 字节来指示显示器能力中的变化。此会在用户连接了诸如麦克风、键盘或显示器的周边装置的情况下或因为某种其他原因而发生。当位 [7:0] 等于 0 时, 所述能力自从发送上一个显示器能力封包以来没有变化。但是, 当位 [7:0] 等于 1 到 255 时, 所述能力已改变。检查显示器能力封包来确定新的显示器特征。

[0543] H. 对于位块传送封包

[0544] 窗口左上部座标 X 值域和 Y 值域各使用 2 字节来规定待移动的窗口的左上角的座标的 X 值和 Y 值。窗口宽度域和高度域各使用 2 字节来规定待移动的窗口的宽度和高度。窗口 X 移动域和 Y 移动域各使用 2 字节来规定窗口分别在水平和垂直方向上将要移动的像素数。通常, 配置这些座标以使得 X 的正值使窗口向右移动且负值引起向左的移动, 同时 Y 的正值使窗口向下移动且负值引起向上移动。

[0545] I. 对于位图区域填充封包

[0546] 窗口左上部座标 X 值域和 Y 值域各使用 2 字节来规定待填充的窗口的左上角的座标的 X 值和 Y 值。窗口宽度域和高度域 (各为 2 字节) 规定待填充的窗口的宽度和高度。视频数据格式描述符域 (2 字节) 规定像素区域填充值的格式。所述格式与视频流封包中的所述同样格式相同。像素区域填充值域 (4 字节) 包含待填充到由上文论述的域所规定的窗口中的像素值。在视频数据格式描述符域中规定了此像素的格式。

[0547] J. 对于位图图案填充封包

[0548] 窗口左上部座标 X 值域和 Y 值域各使用 2 字节来规定待填充的窗口的左上角的座标的 X 值和 Y 值。窗口宽度域和高度域 (各为 2 字节) 规定待填充的窗口的宽度和高度。图案宽度域和图案高度域 (各为 2 字节) 分别规定填充图案的宽度和高度。2 字节的视频数据格式描述符域规定像素区域填充值的格式。图 11 说明如何编码视频数据格式描述符。所述格式与视频流封包中的同样域相同。

[0549] 参数 CRC 域 (2 字节) 包含从封包长度域到视频格式描述符域的所有字节的 CRC。如果此 CRC 未通过校验, 则丢弃整个封包。图案像素数据域包含原始视频信息, 其规定了由视频数据格式描述符所规定的格式中的填充图案。数据被压缩为字节, 且每一行的第一个像素必须是字节对准的。每次一行地传输填充图案数据。图案像素数据 CRC 域 (2 字节) 仅包含图案像素数据的 CRC。如果此 CRC 未通过校验, 则仍可使用图案像素数据, 但将 CRC 错误计数加 1。

[0550] K. 通讯链路数据通道封包

[0551] 参数 CRC 域 (2 字节) 包含从封包长度域到封包类型域的所有字节的 16 位 CRC。如果此 CRC 未通过校验, 则丢弃整个封包。

[0552] 通讯链路数据域包含来自通讯通道的原始数据。此数据被简单地传递到显示器中的计算装置。

[0553] 通讯链路数据 CRC 域 (2 字节) 仅包含通讯链路数据的 16 位 CRC。如果此 CRC 未通过校验, 则通讯链路数据仍被使用或有用, 但将 CRC 错误计数加 1。

[0554] L. 对于接口类型转递请求封包

[0555] 接口类型域 (1 字节) 规定了将使用的新接口类型。此域中的值以如下方式规定了接口类型。如果 7 位中的值等于“0”,则所述类型转递请求是用于正向链路,如果其等于“1”,则类型转递请求是用于反向链路。将 6 至 3 位保留用于未来用途,且一般将其设为零。2 至 0 位用于定义待使用的接口类型,其中值为 1 意味着转递到 I 型模式,值为 2 意味着转递到 II 型模式,值为 3 意味着转递到 III 型模式,且值为 4 意味着转递到 IV 型模式。将“0”值和 5 到 7 的值保留用于未来的替代模式或模式组合的标志。

[0556] M. 对于接口类型确认封包

[0557] 接口类型域 (1 字节) 具有一个值,其确认将使用的新接口类型。此域中的值以如下方式规定了接口类型。如果 7 位等于“0”,则类型转递请求是用于正向链路,或者,如果其等于“1”,则类型转递请求是用于反向链路。当前将位位置 6 至 3 保留用于按需要标识其他转递类型,且一般将其设为零。但是,位位置 2 至 0 用于定义将使用的接口类型,其中值为“0”指示一个否定确认(或者所请求的转递不能被执行),值为“1”、“2”、“3”和“4”分别指示到 I 型、II 型、III 型和 IV 型模式的转递。将值 5 到 7 保留,按需要用于替代的模式标志。

[0558] N. 对于执行类型转递封包

[0559] 1 字节接口类型域指示了将使用的新接口类型。此域中的值通过如下方式规定了接口类型,首先使用 7 位的值来确定类型转递是用于正向链路还是用于反向链路。值为“0”指示类型转递请求是用于正向链路,且值为“1”指示类型转递请求是用于反向链路。将 6 至 3 位保留用于未来用途,且因而一般将其设为零值。然而,2 至 0 位用于定义待使用的接口类型,其中值 1、2、3 和 4 分别规定使用到 I 型、II 型、III 型和 IV 型模式的转递。将这些位的 0 值和 5 到 7 的值保留用于未来用途。

[0560] O. 对于正向音频通道启用封包

[0561] 音频通道启用屏蔽域 (1 字节) 包含一组旗标,其指示将在客户端中启用哪些音频通道。一个设为 1 的位启用对应的通道,且一个设为 0 的位禁用对应的通道。0 至 5 位标示通道 0 至 5,其分别针对左前、右前、左后、右后、正中、亚低音扬声器 (sub-woofer) 通道。将 6 位和 7 位保留用于未来用途,同时一般将其设为零。

[0562] P. 对于反向音频采样率封包

[0563] 音频采样率域 (1 字节) 规定了数字音频采样率。将此域的值指派至不同速率,其中分别将 0、1、2、3、4、5、6、7 和 8 的值用于标识 8,000、16,000、24,000、32,000、40,000、48,000、11,025、22,050 和 44,100 样本每秒 (SPS),同时将 9 至 254 的值保留并按需要用于替代速率,所以当前将其设为“0”。将值 255 用于禁用反向链路音频流。

[0564] 样本格式域 (1 字节) 规定了数字音频样本的格式。当位 [1:0] 等于“0”时,数字音频样本为线性格式,当其等于 1 时,数字音频样本为 μ -Law 格式,且当其等于 2 时,数字音频样本为 A-Law 格式。将位 [7:2] 保留并按需要用于替代地识别音频格式,且一般将其设为零。

[0565] Q. 对于数字内容保护开销封包

[0566] 内容保护类型域 (1 字节) 规定了所使用的数字内容保护方法。“0”值指示数字传输内容保护 (DTCP),而值为 1 指示高带宽数字内容保护系统 (HDCP)。当前未对 2 至 255 的值范围做出规定,但将其保留并按需要用于替代的保护机制。内容保护开销消息域是一个

可变长度域,其包含在主机和客户端之间发送的内容保护消息。

[0567] R. 对于透明色彩启用封包

[0568] 透明色彩启用域(1字节)规定了何时启用或禁用透明色彩模式。如果0位等于0,则禁用透明色彩模式,如果其等于1,则启用透明色彩模式,且透明色彩由以下两个参数规定。将此字节的1至7位保留用于未来用途且通常将其设为零。

[0569] 视频数据格式描述符域(2字节)规定了像素区域填充值得格式。图11说明了如何将视频数据格式描述符编码。所述格式一般与视频流封包中的同样域相同。

[0570] 像素区域填充值域使用为待填充到上文规定的窗口中的像素值而分配的4字节。在视频数据格式描述符域中规定了此像素的格式。

[0571] S. 对于往返行程延迟测量封包

[0572] 在一个实施例中,参数CRC域(2字节)包含从封包长度域到封包类型域的所有字节的16位CRC。如果此CRC未通过校验,则丢弃整个封包。

[0573] 全零域(1字节)包含零,以确保在第一保护时间周期期间禁用线驱动器之前所有MDDI_Data讯号都处于零状态。

[0574] 保护时间1域(8字节)用于允许在客户端(显示器)中的线驱动器被启用之前禁用主机中的MDDI_Data线驱动器。主机在保护时间1的0位期间禁用其MDDI_Data线驱动器,且显示器在保护时间1的最后一个位后立刻启用其线驱动器。

[0575] 测量周期域是一个512字节的窗口,其用来允许显示器在正向链路上所用的数据率的一半的速率下以0xff、0xff、0x0响应。此速率对应于为1的反向链路速率除数。显示器在测量周期的开始处立刻返回此响应。将精确地在主机处的测量周期的第一个位开始后加上链路的往返行程延迟的时间处在主机处接收到此响应。在来自显示器的0xff、0xff、0x00响应之前和之后立刻禁用显示器中的MDDI_Data线驱动器。

[0576] 保护时间2域(8字节)中的值允许在主机中的线驱动器被启用之前禁用客户端MDDI_Data线驱动器。保护时间2总存在,但仅当往返行程延迟处于在测量周期中可测量的最大量时才需要。客户端在保护时间2的0位期间禁用其线驱动器,且主机在保护时间2的最后一个位后立刻启用其线驱动器。

[0577] 驱动器重启用域(1字节)被设为零,以确保在下一个封包的封包长度域之前重启用所有MDDI_Data讯号。

[0578] T. 对于正向链路偏斜校正封包

[0579] 在一个实施例中,参数CRC域(2字节)包含从封包长度域到封包类型域的所有字节的16位CRC。如果此CRC未通过校验,则丢弃整个封包。

[0580] 校正数据序列域包含一个512字节数据序列,其导致MDDI_Data讯号在每个数据周期中双态切换。在校正数据序列的处理期间,MDDI主机控制器将所有MDDI_Data讯号设为等于选通讯号。当客户端显示器正在接收校正数据序列域时,显示器时钟恢复电路应仅使用MDDI_Stb而不是MDDI_Stb与MDDI_Data0的异或运算结果来恢复数据时钟。取决于在校正数据序列域的开始时MDDI_Stb讯号的确切相位,基于发送此封包时所使用的接口类型,校正数据序列一般将为以下序列之一:

[0581] I型 -0xaa、0xaa... 或 0x55、0x55...

[0582] II型 -0xcc、0xcc... 或 0x33、0x33...

[0583] III 型 -0xf0、0xf0... 或 0x0f、0x0f...

[0584] IV 型 -0xff、0x00、0xff、0x00... 或 0x00、0xff、0x00、0xff...

[0585] 图 62A 和 62B 中分别展示了用于 I 型和 II 型接口的可能的 MDDI_Data 和 MDDI_Stb 波形的实例。

[0586] XVII. 结论

[0587] 尽管上文已描述本发明的各种实施例,应了解其仅是以例示性而不是以限制性的方式提供的。因此本发明的广度和范畴不应受到任何上述例示性实施例的限制,而是仅应根据随后的权利要求及其均等物来加以界定。

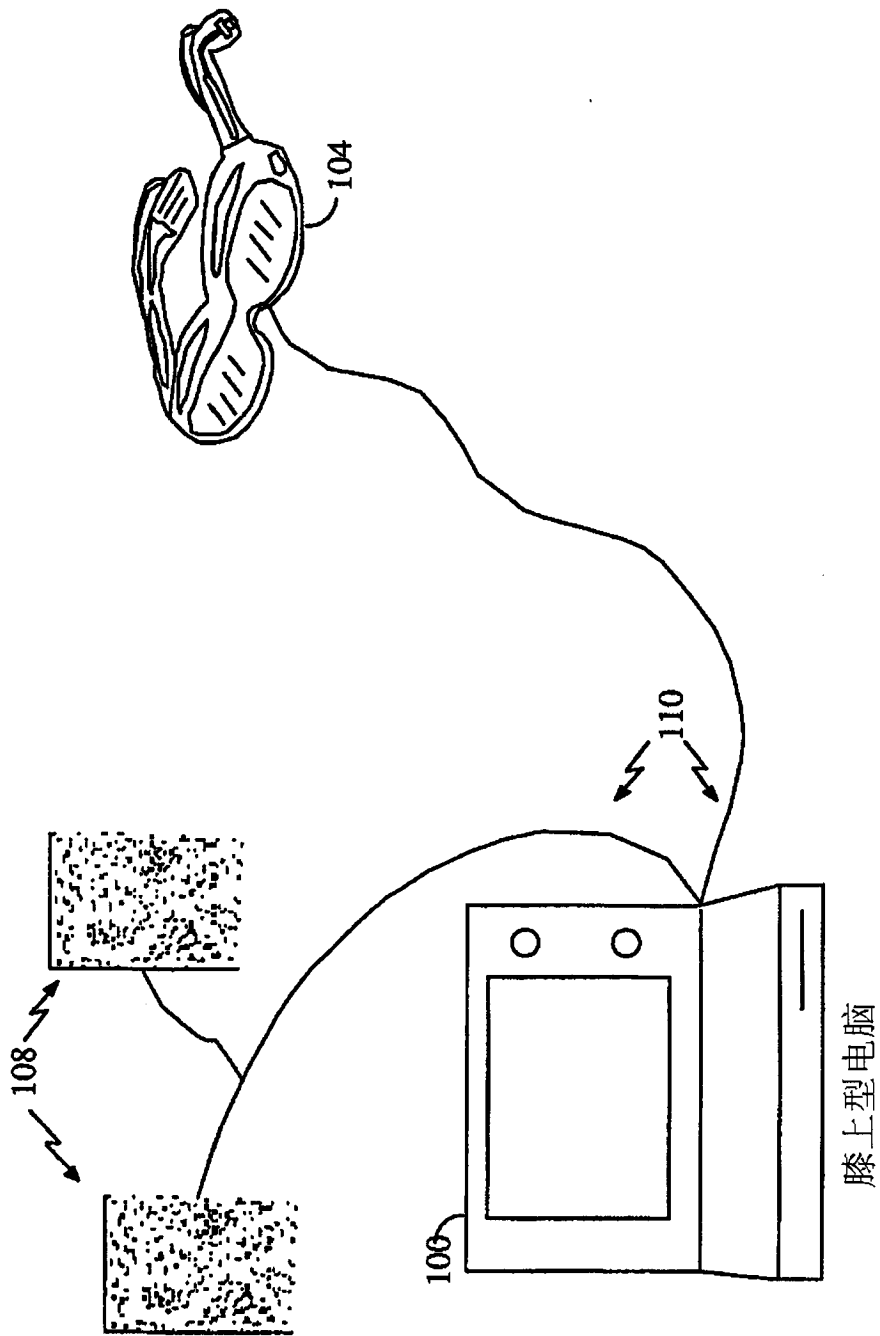


图 1A

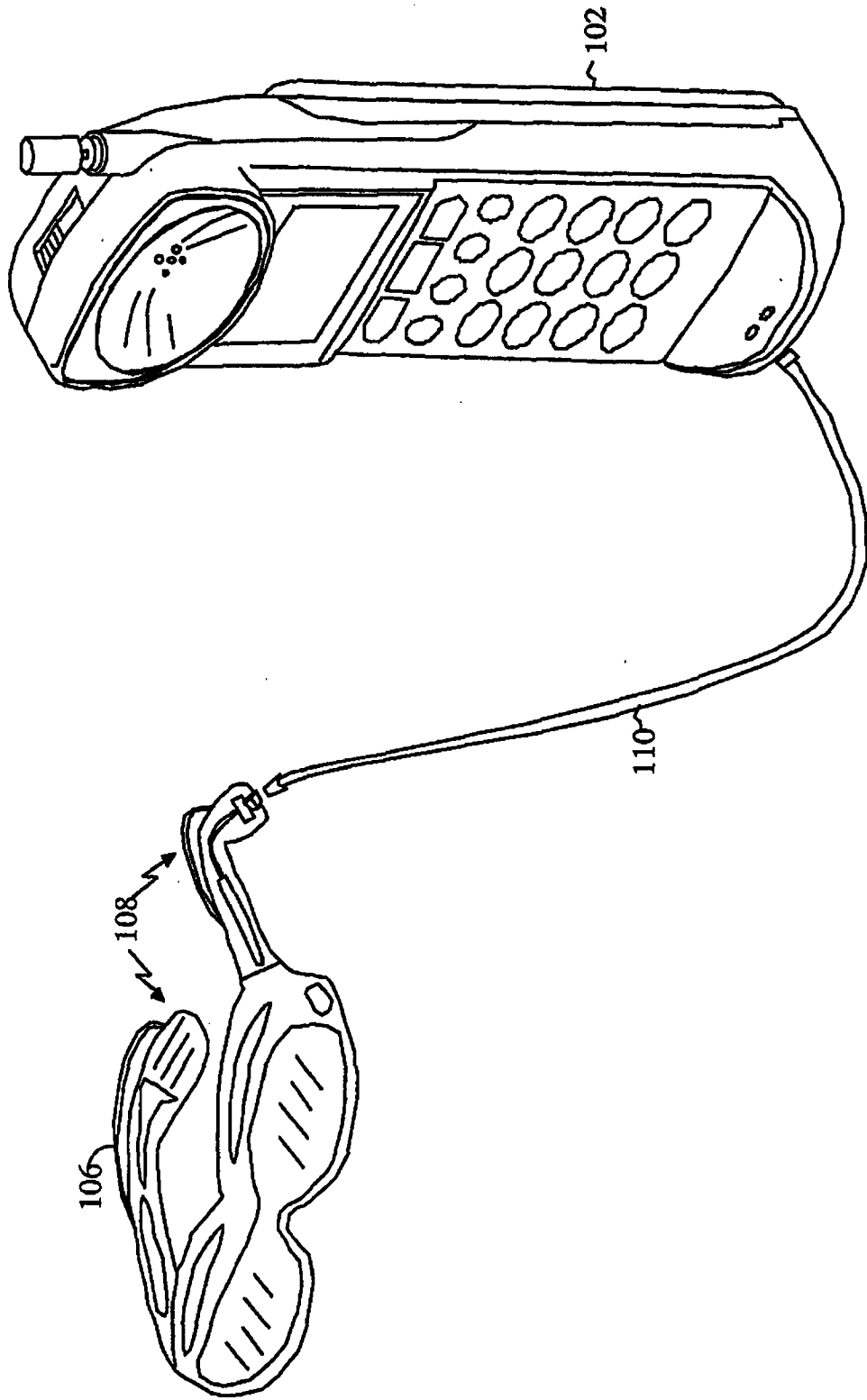


图 1B

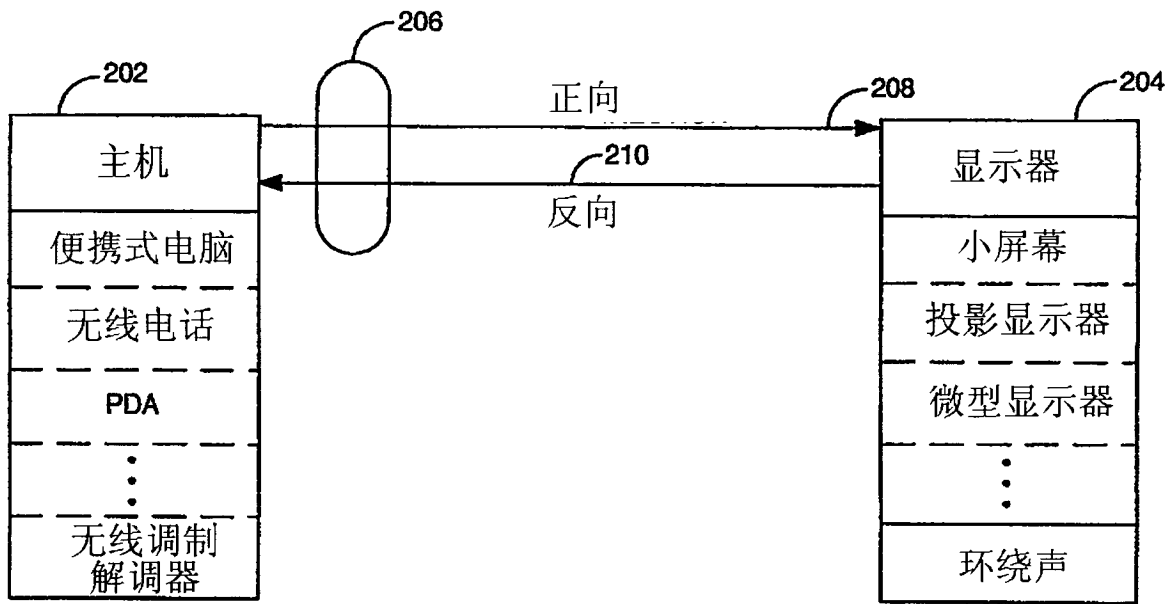


图 2

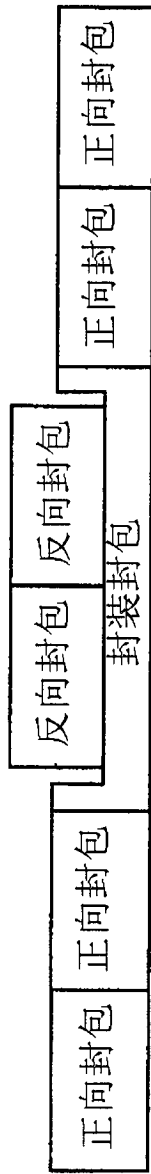


图 3

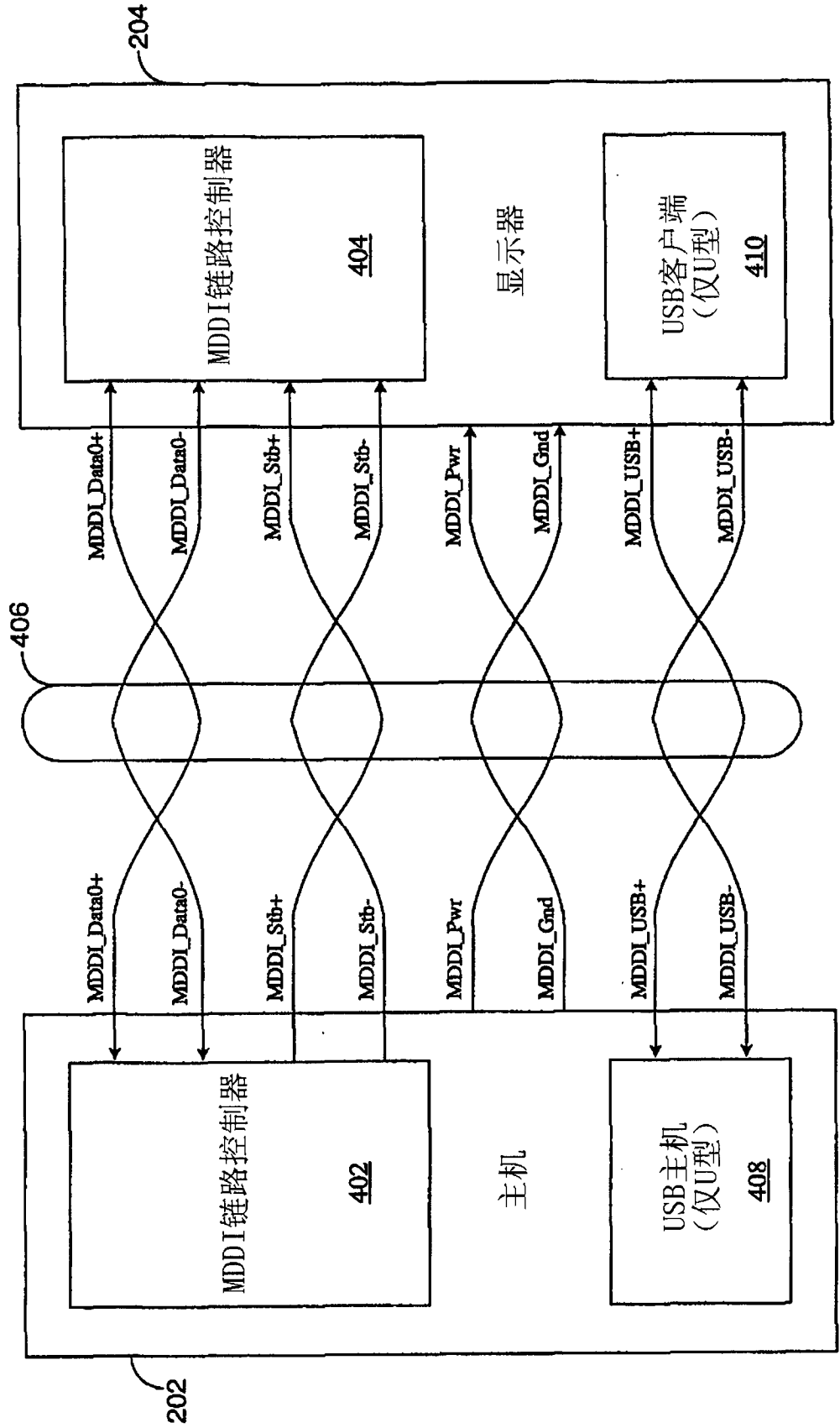


图 4

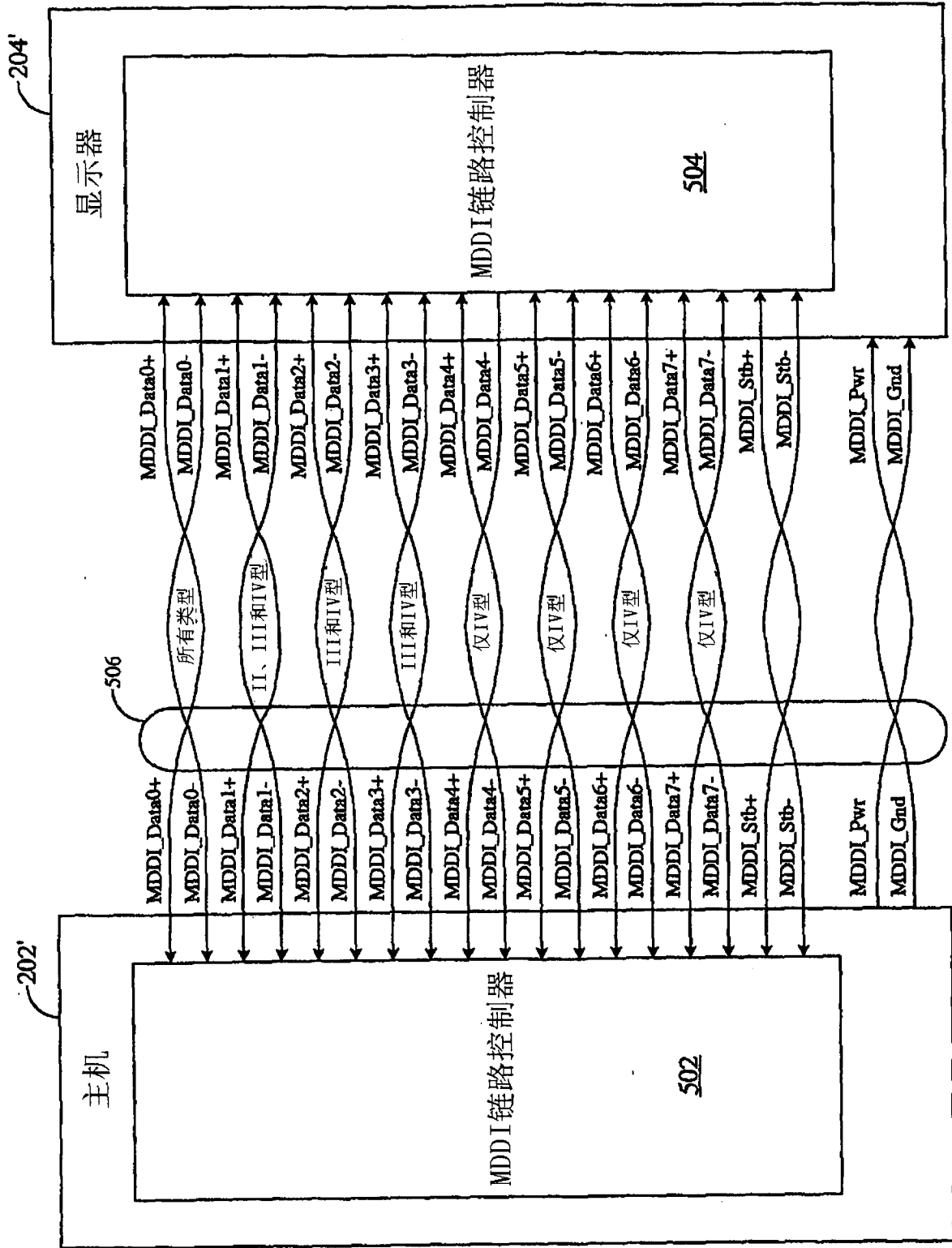


图 5

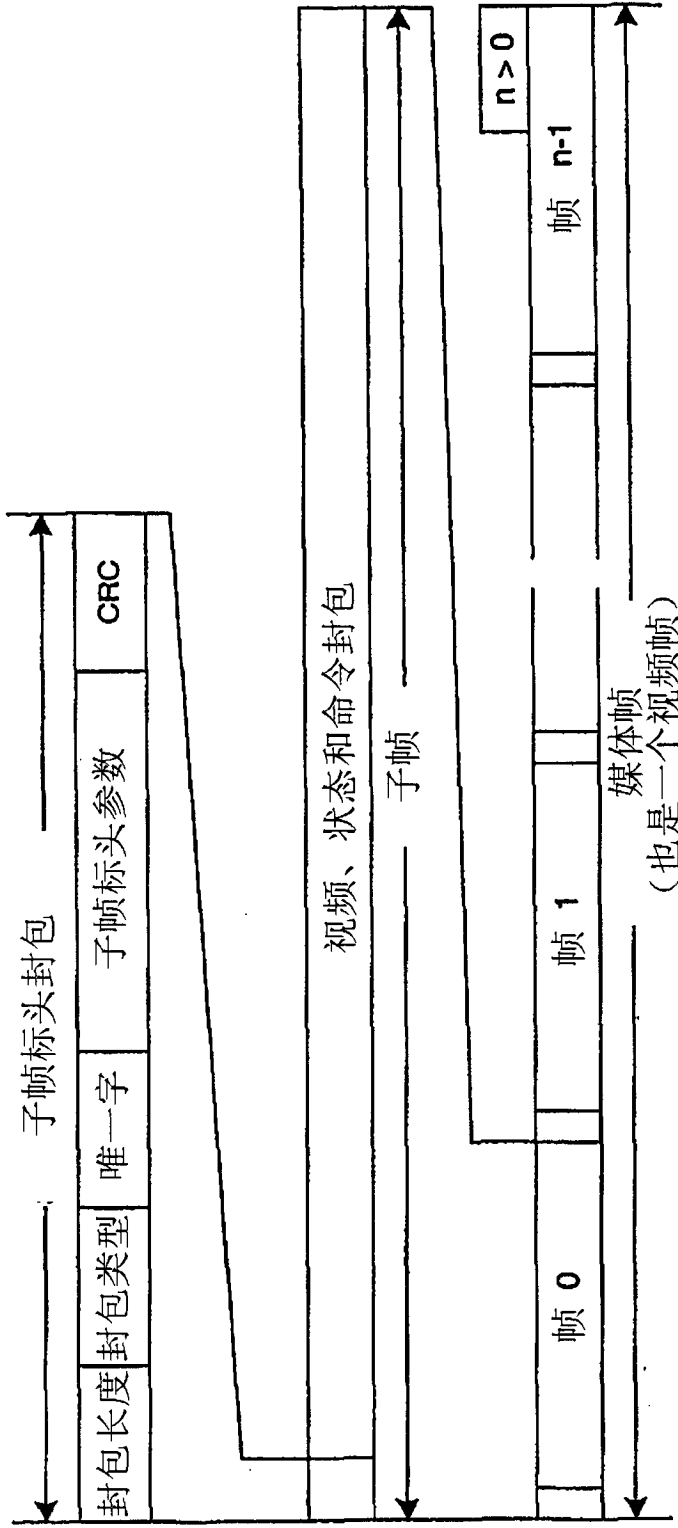


图 6



图 7

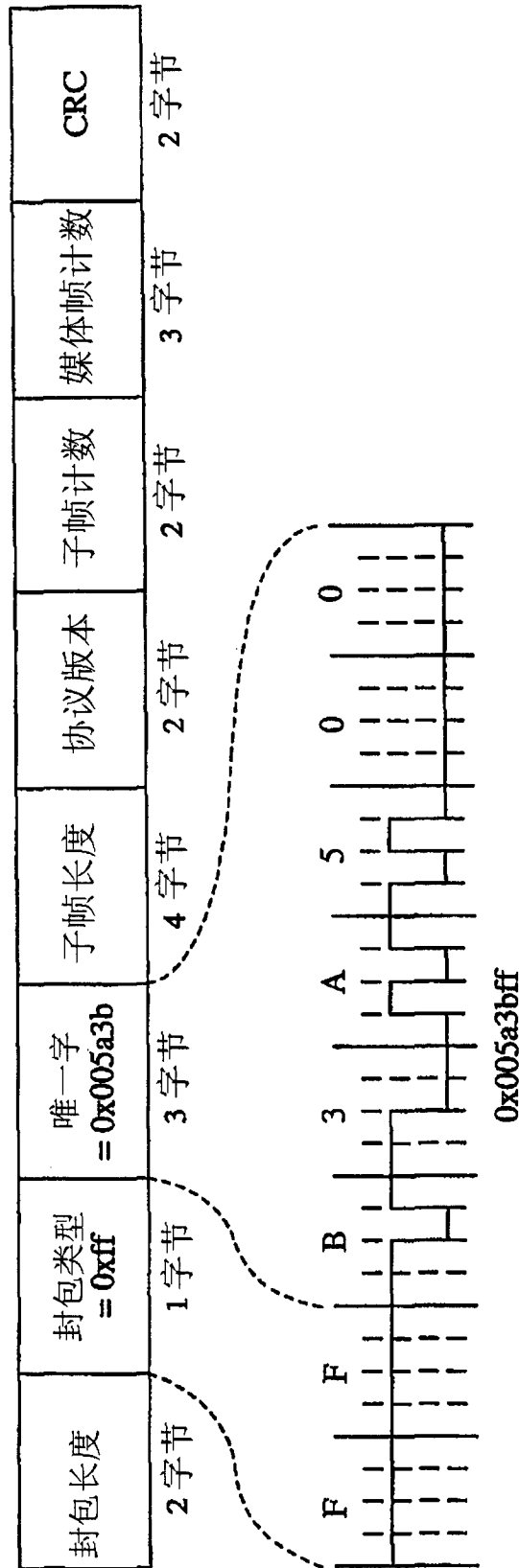


图 8

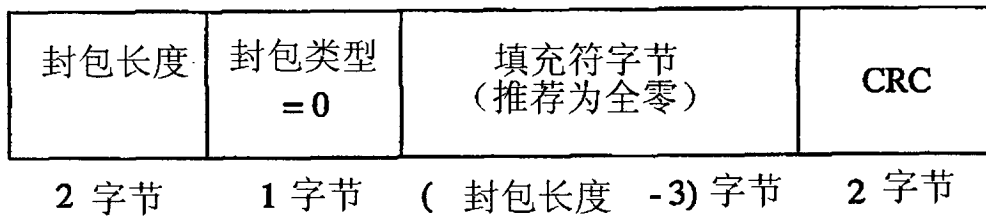


图 9

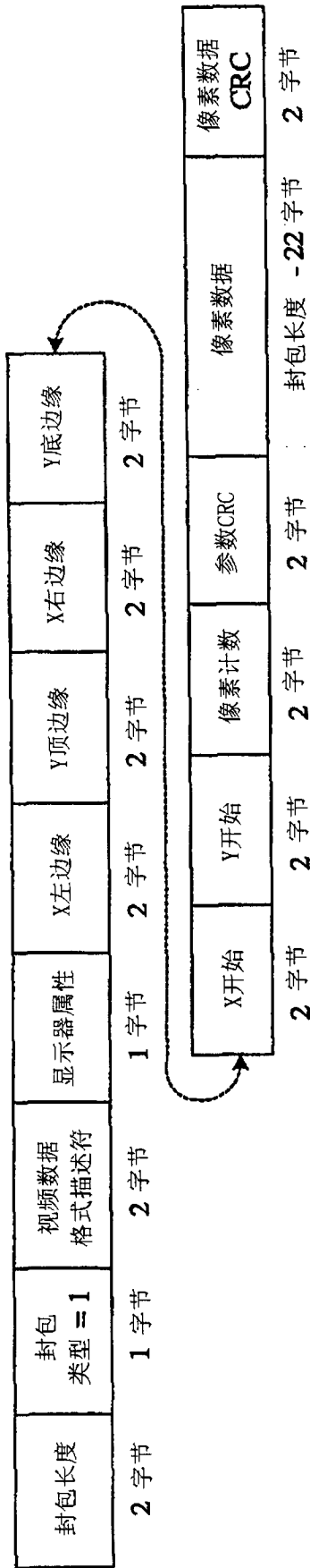


图 10

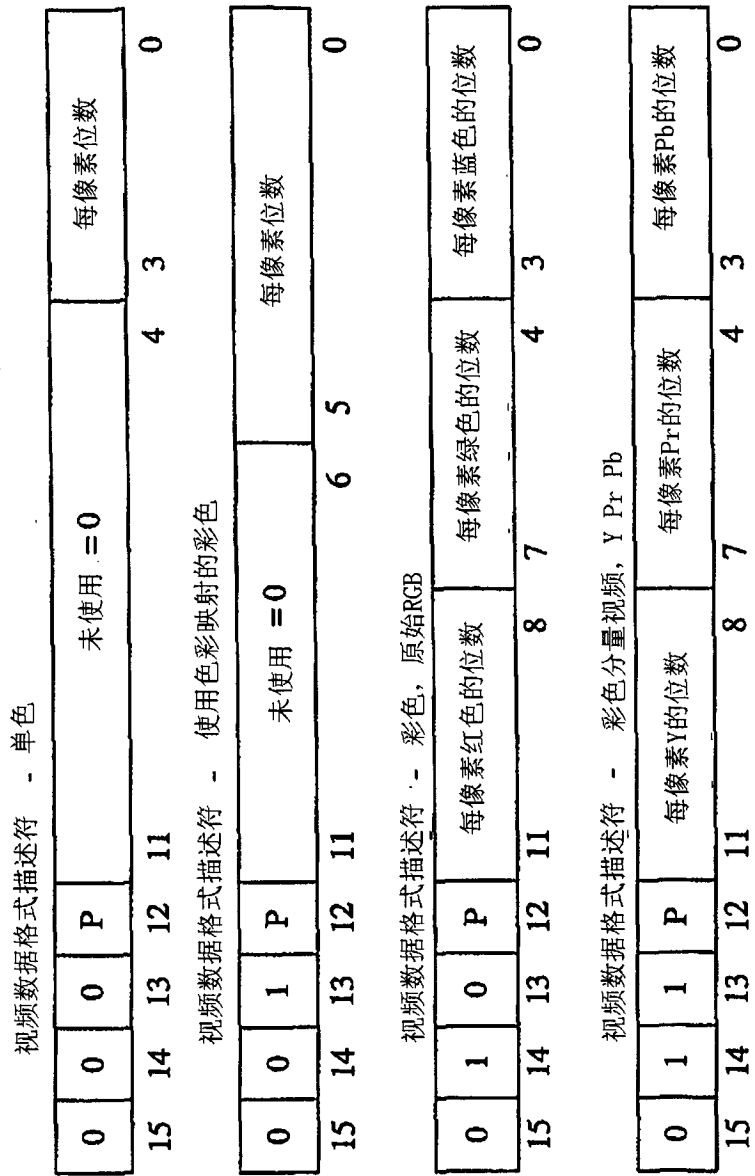


图 11

字节对准 (未压缩) 像素数据, P=0

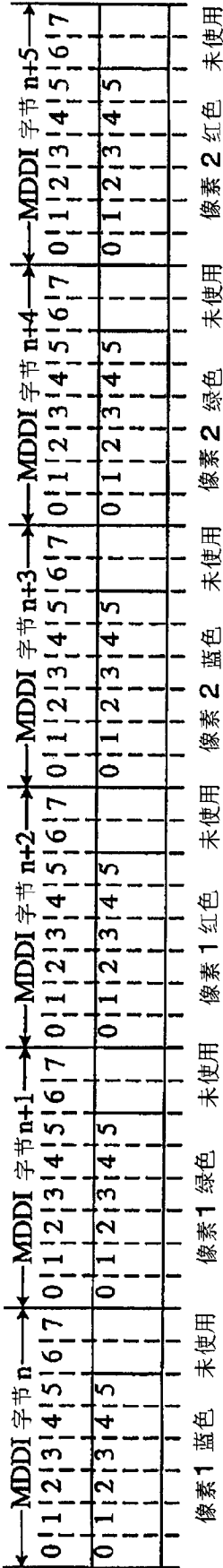


图 12

压缩的像素数据, P=1

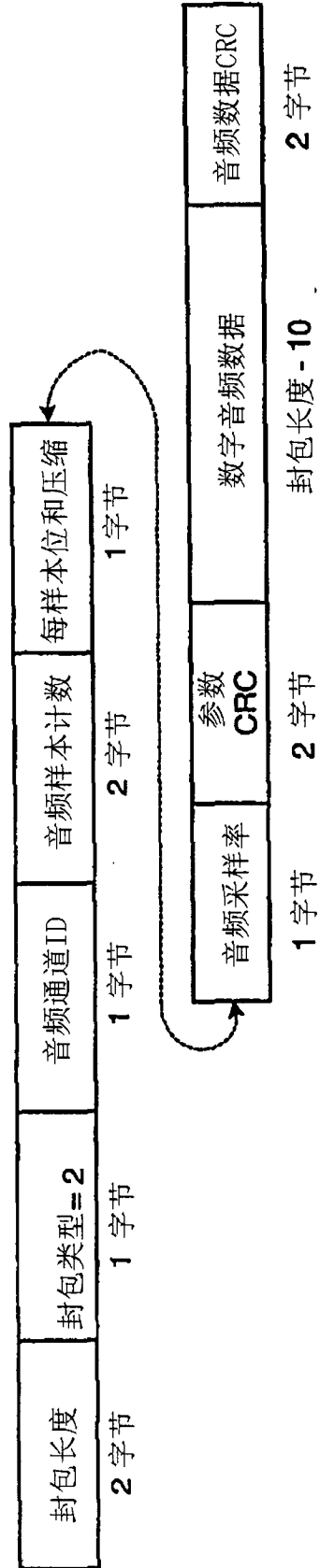
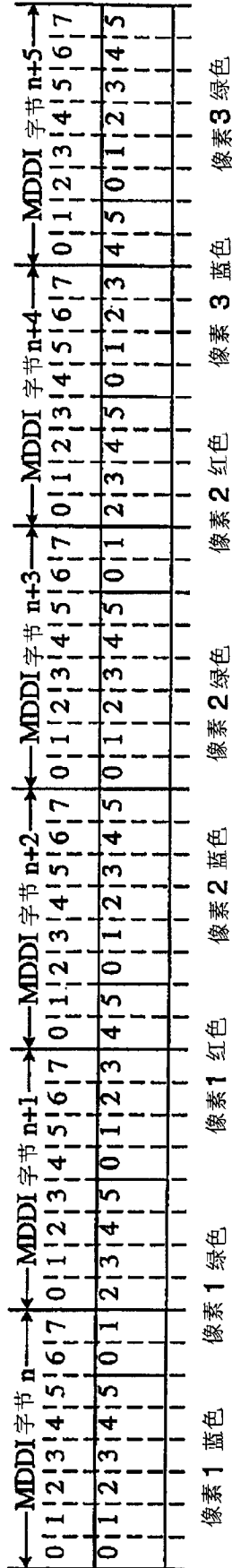


图 13

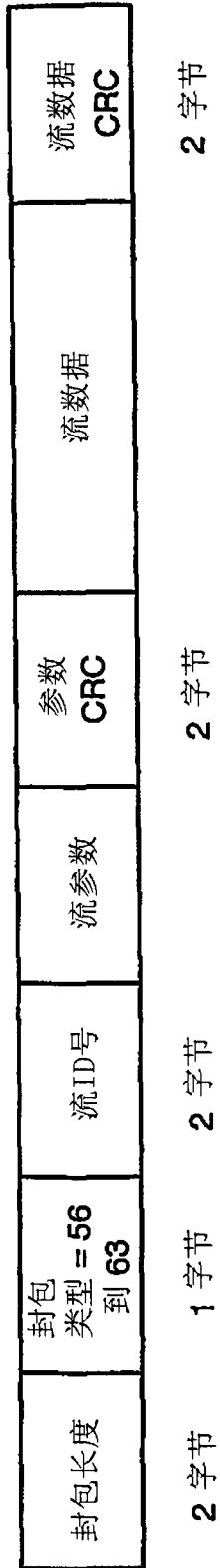


图 15

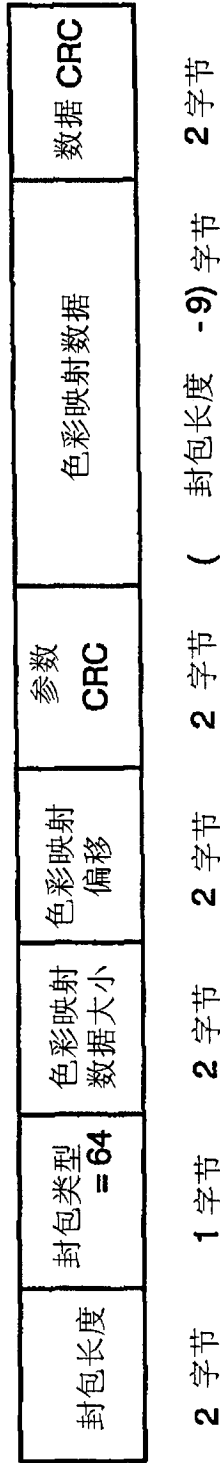
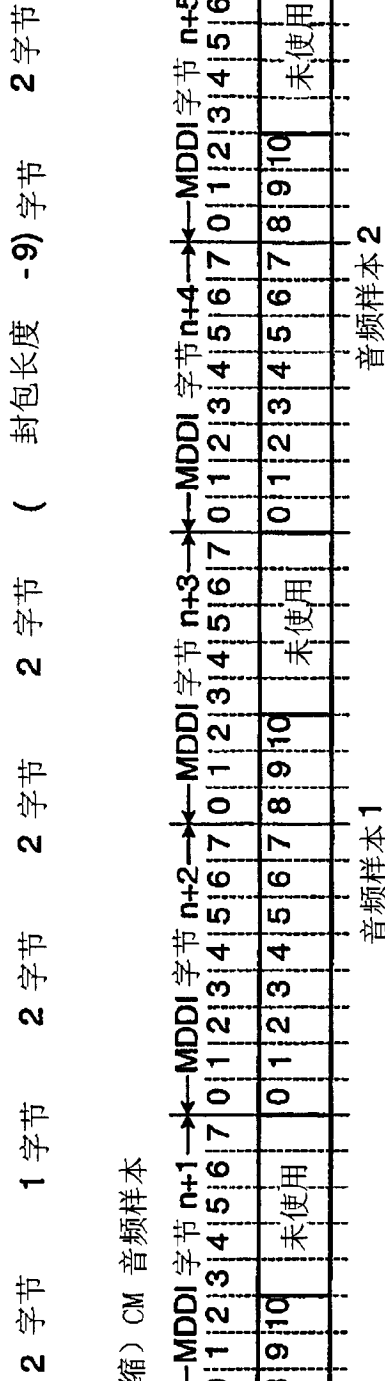
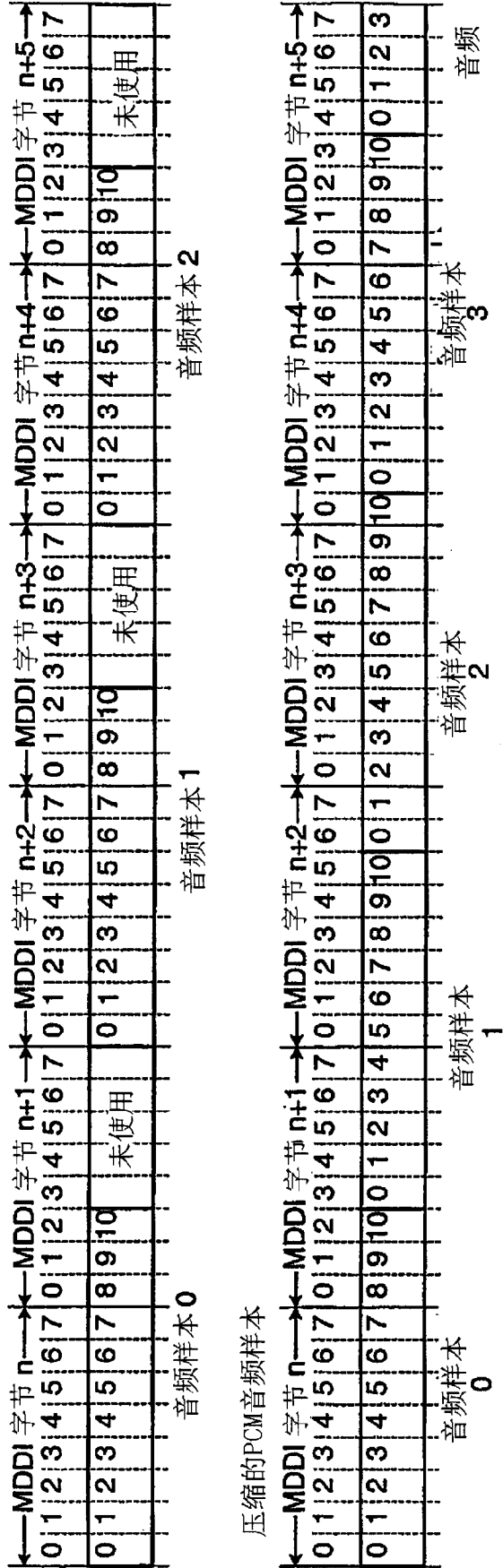


图 16



字节对准 (未压缩) CM 音频样本



压缩的PCM音频样本

图 14

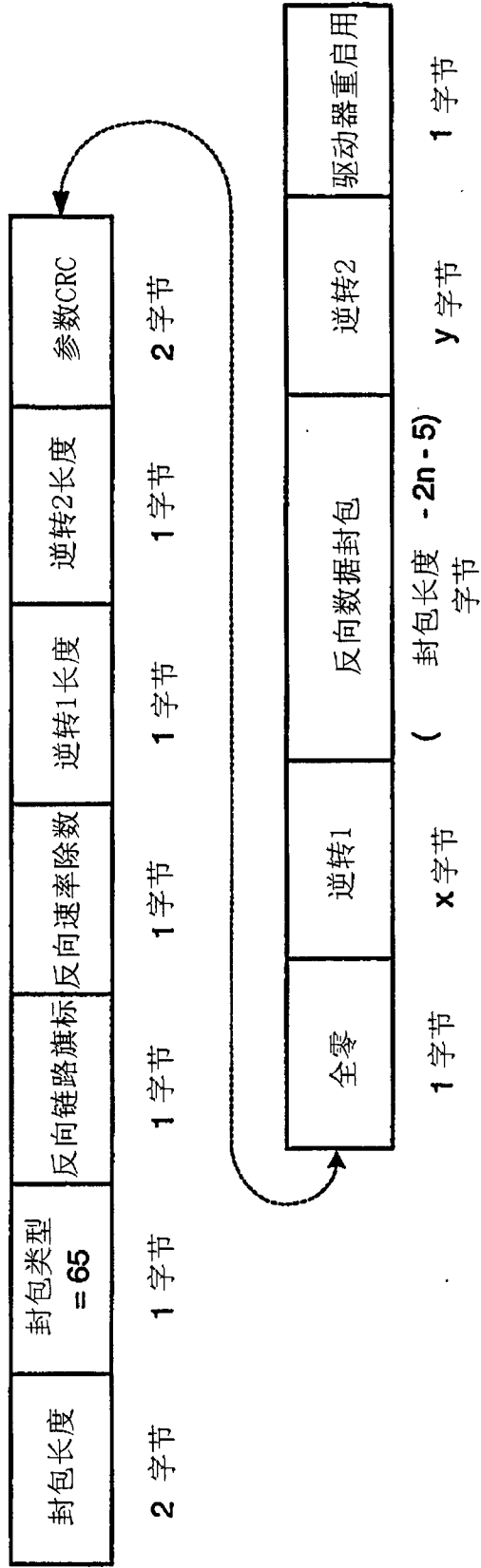


图 17

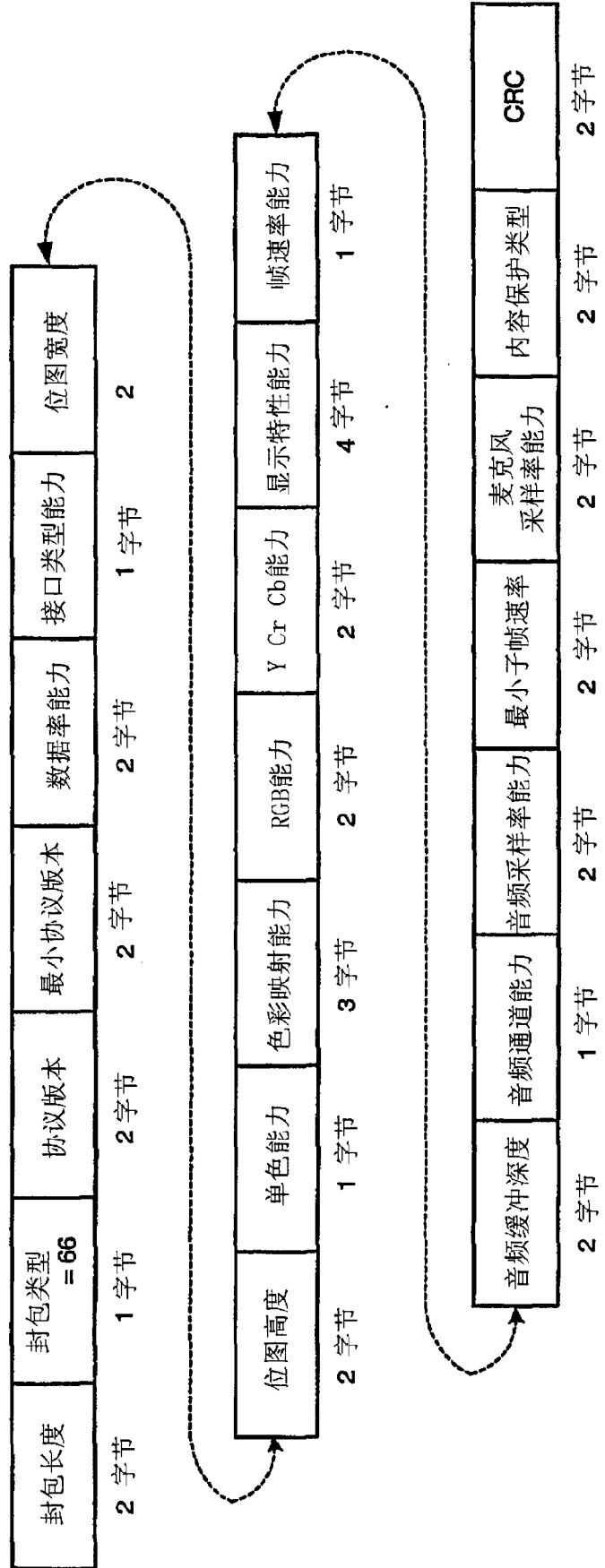


图 18

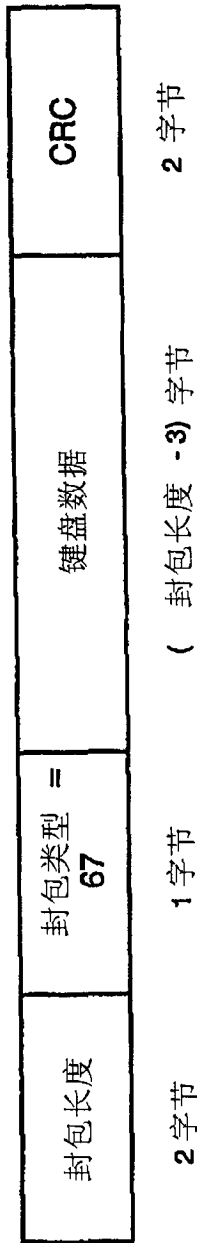


图 19

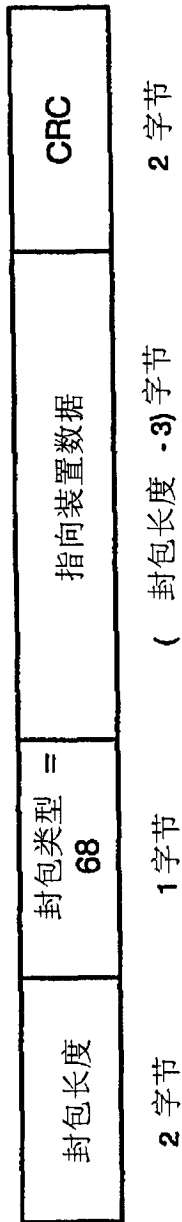


图 20

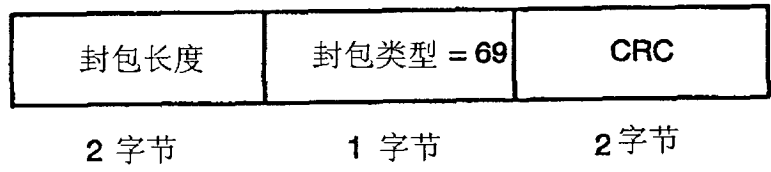


图 21

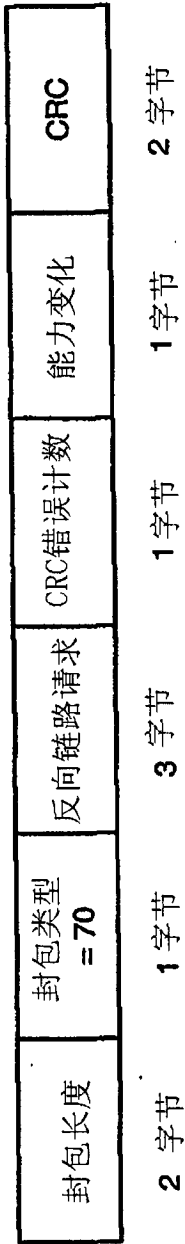


图 22

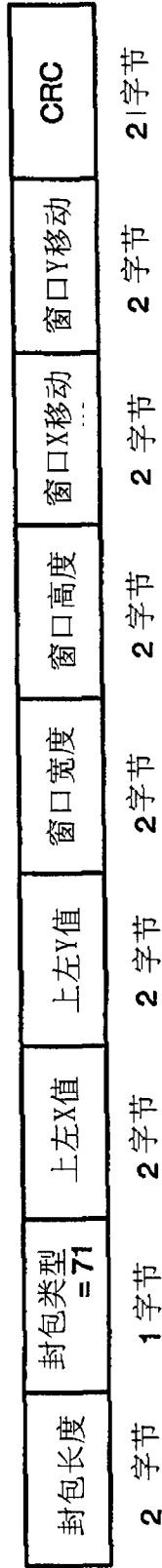


图 23

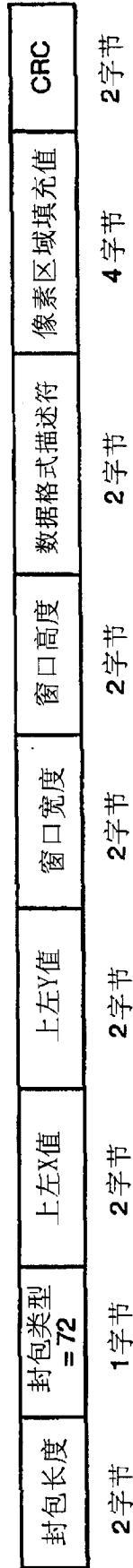


图 24

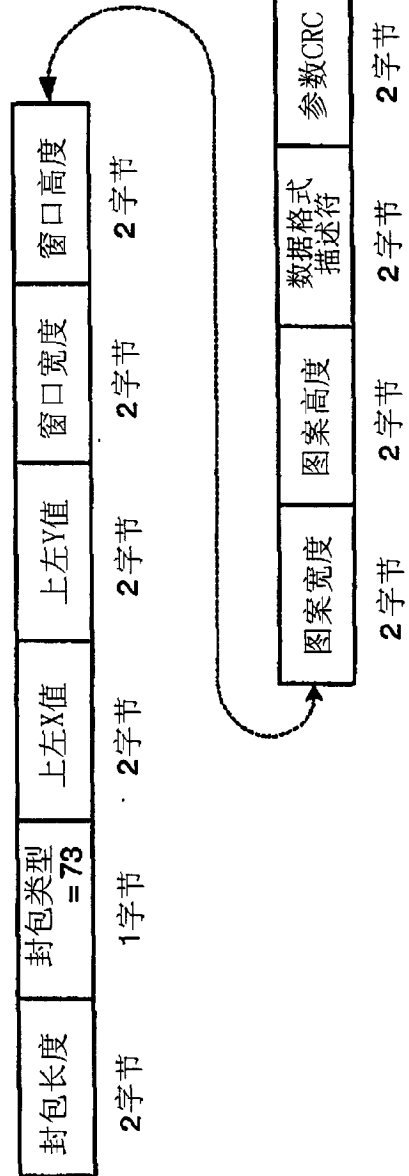


图 25

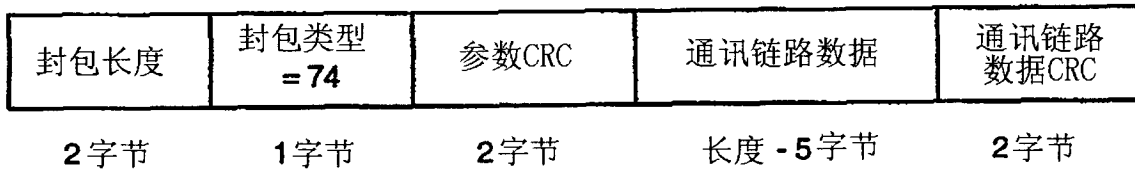


图 26

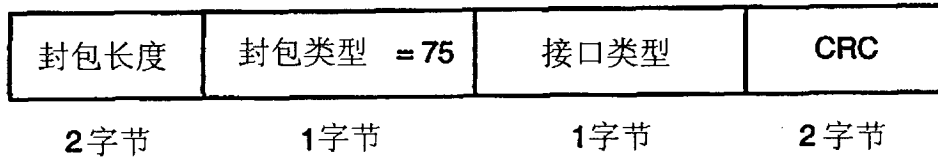


图 27

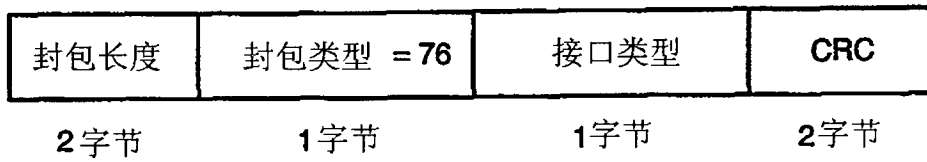


图 28

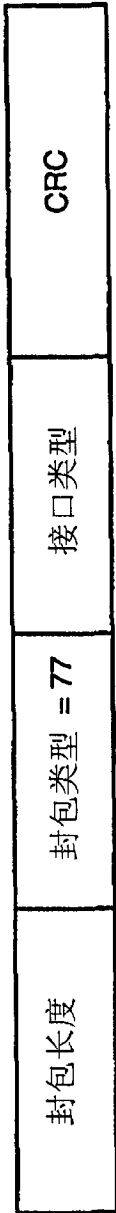


图 29



图 30



图 31

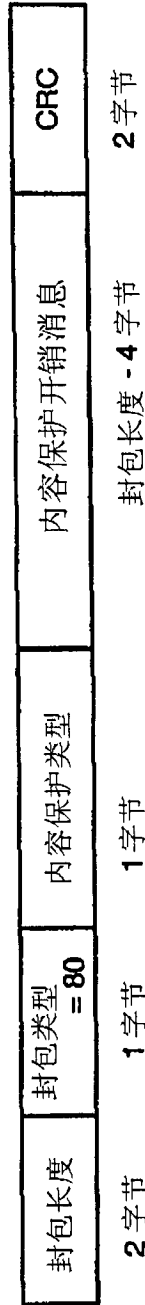


图 32



图 33

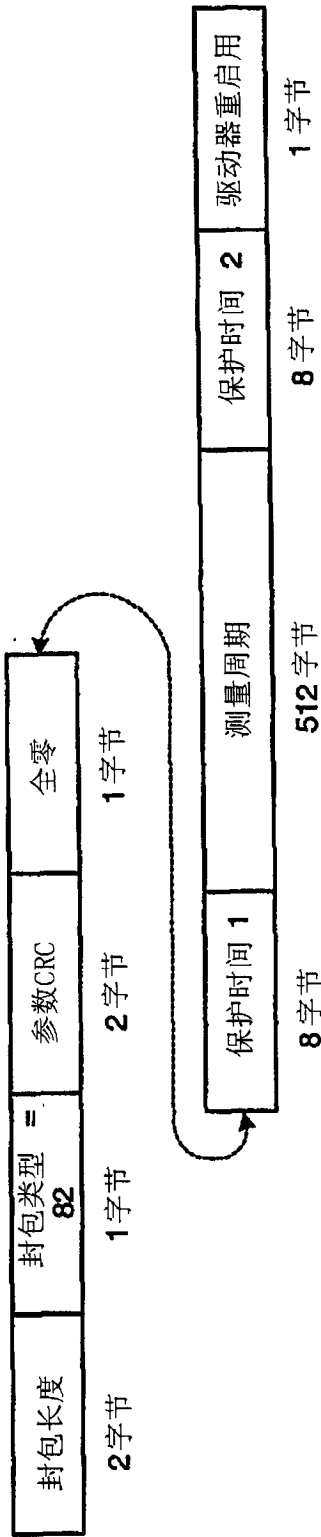


图 34

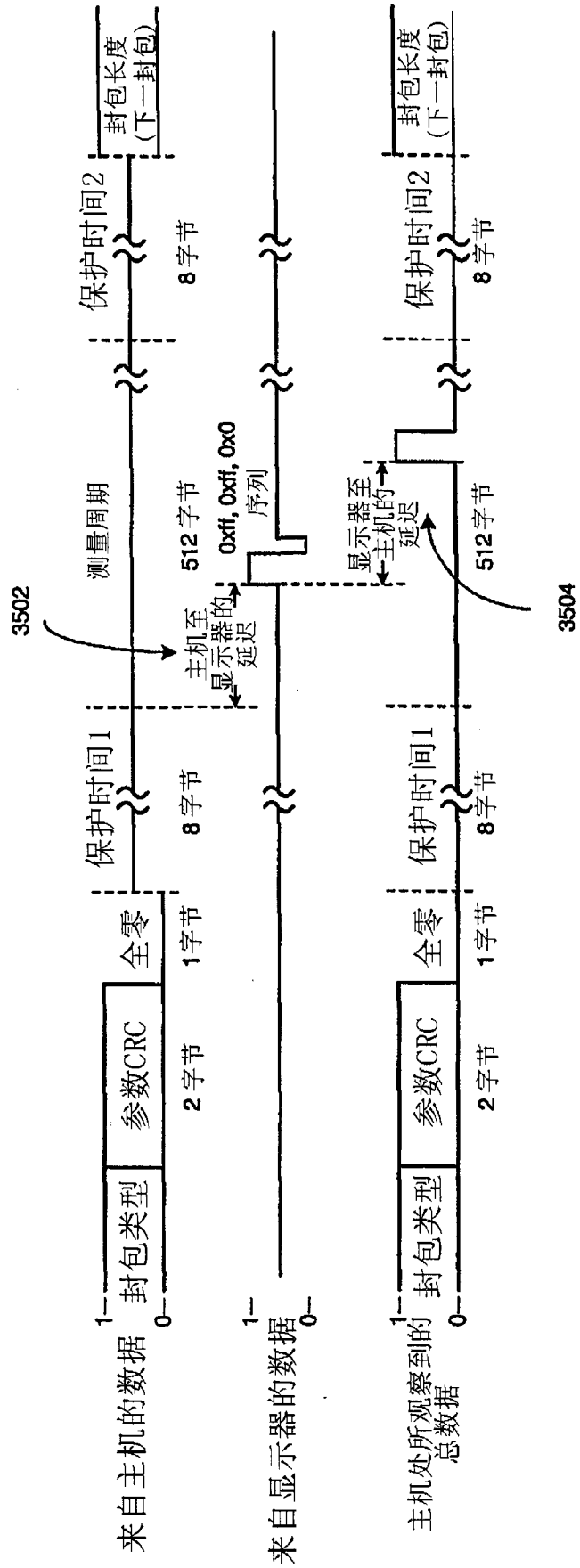


图 35

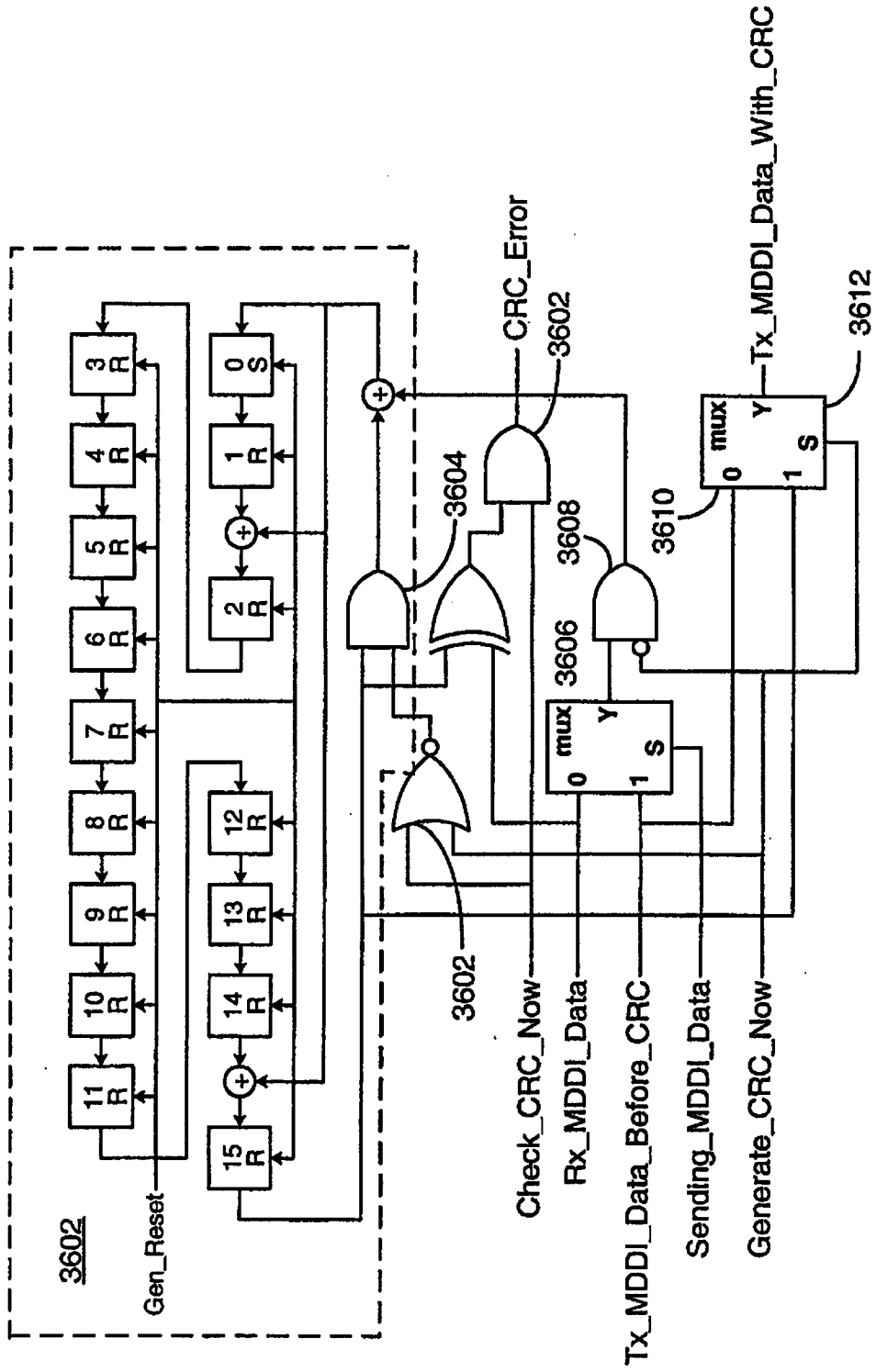


图 36

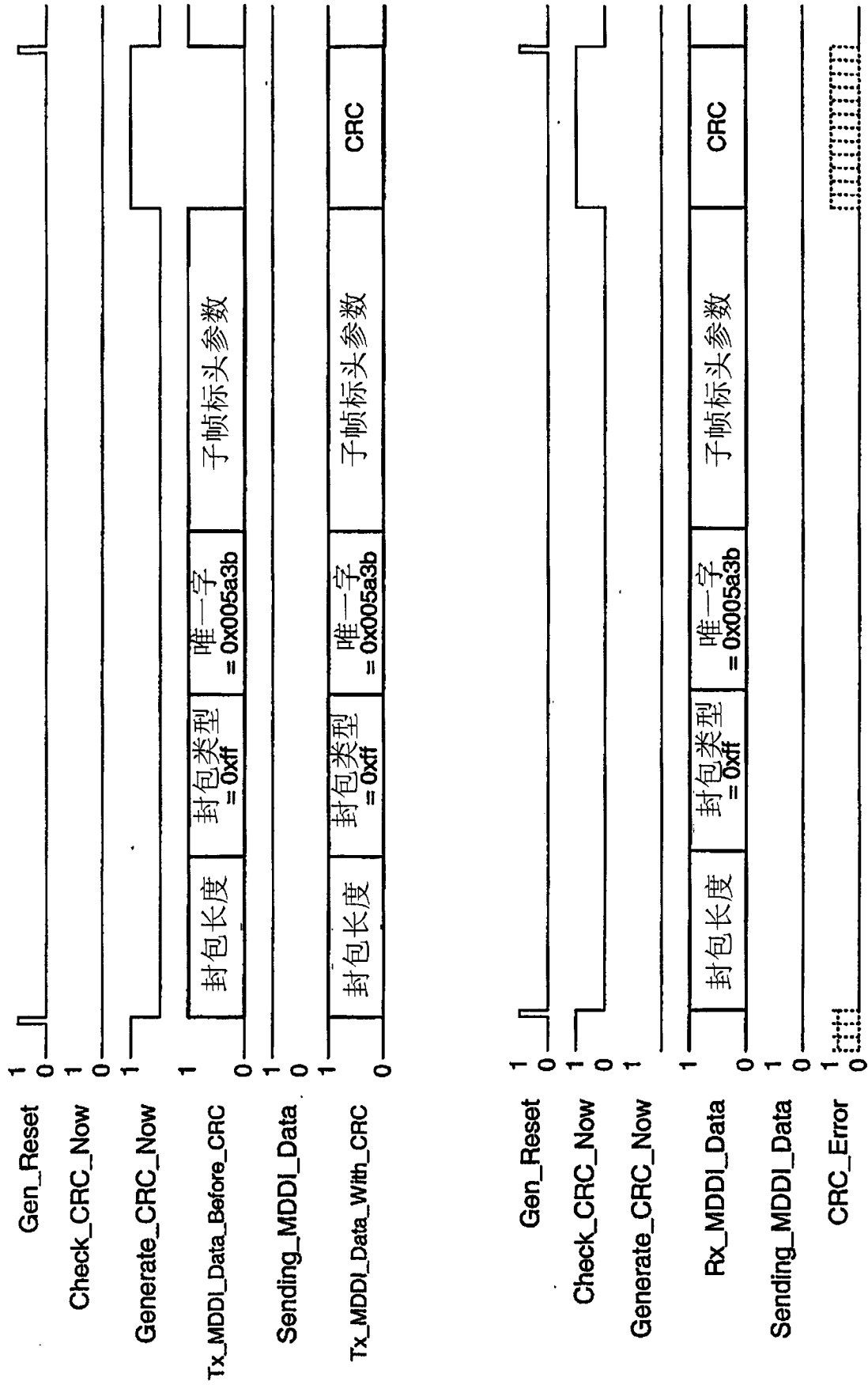


图 37A

图 37B

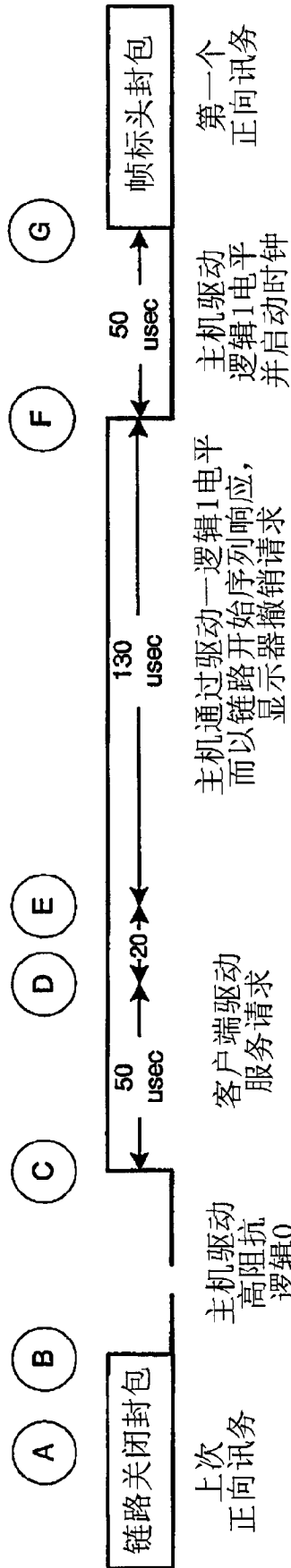


图 38

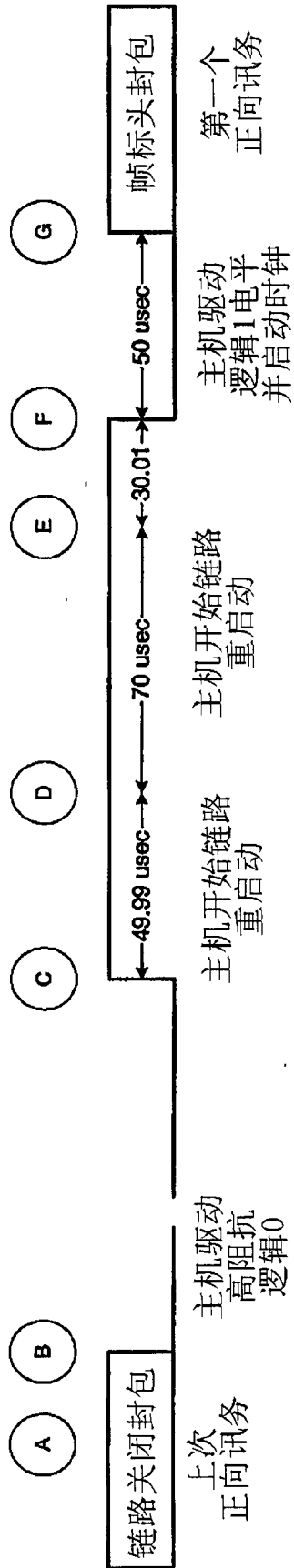


图 39

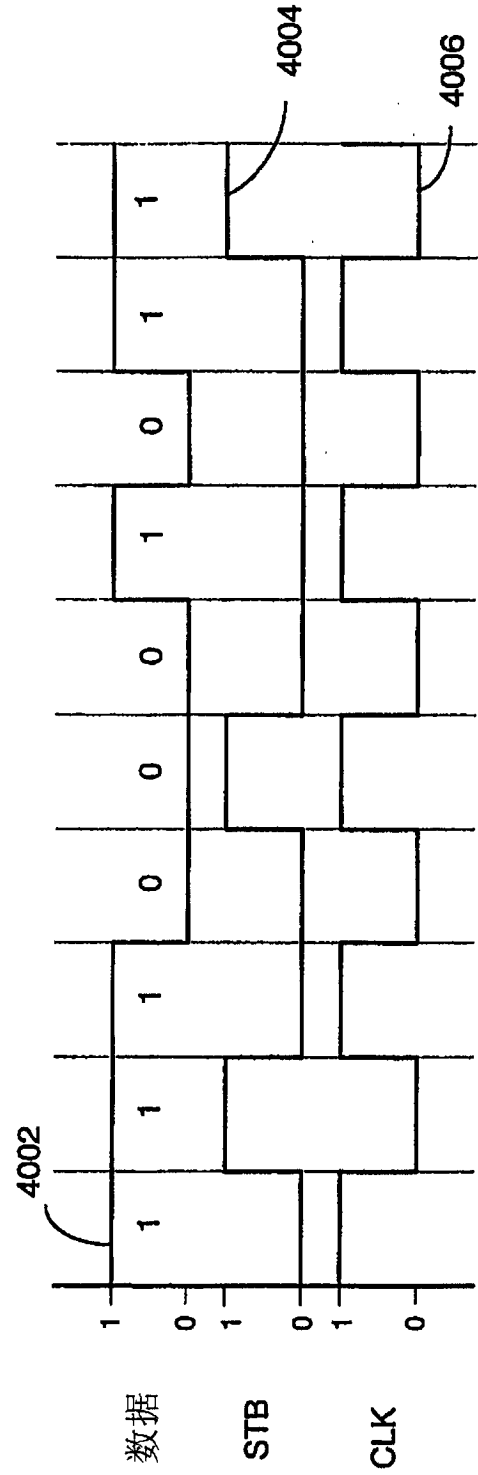


图 40

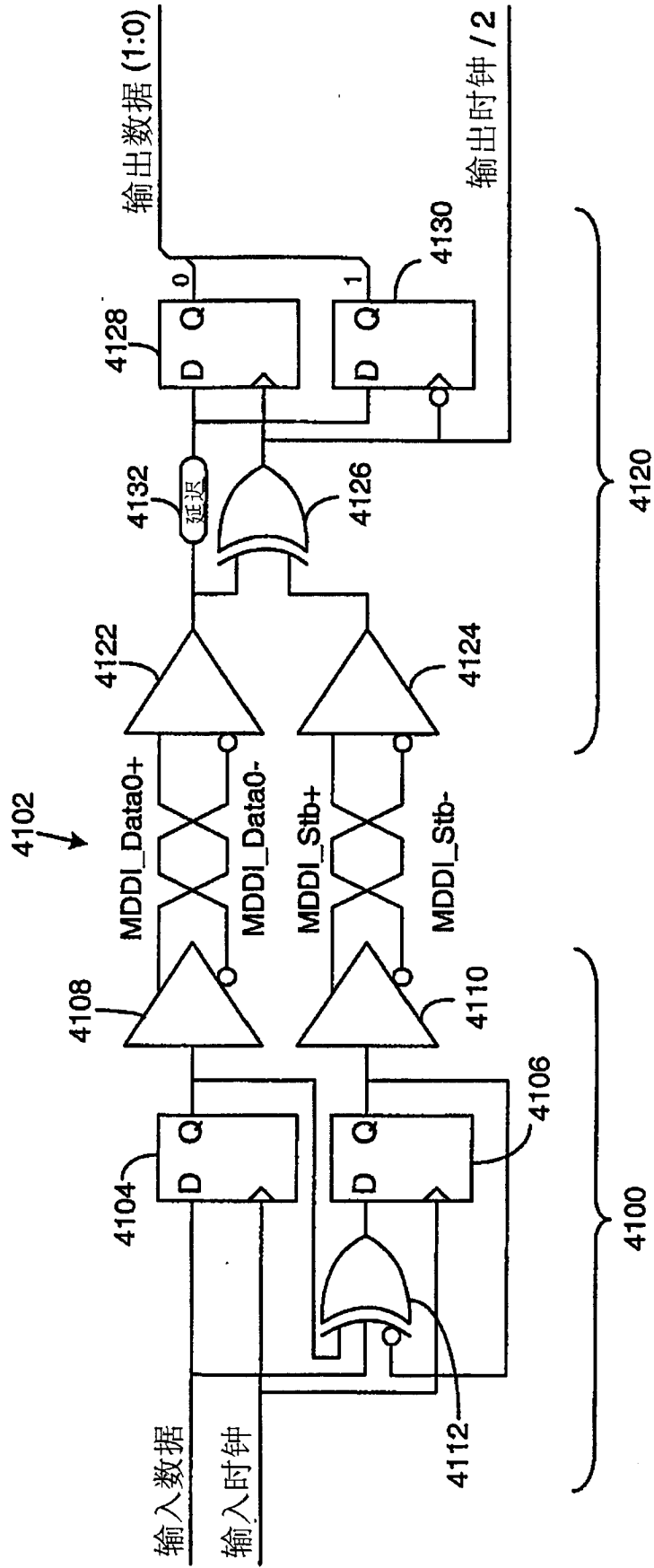


图 41

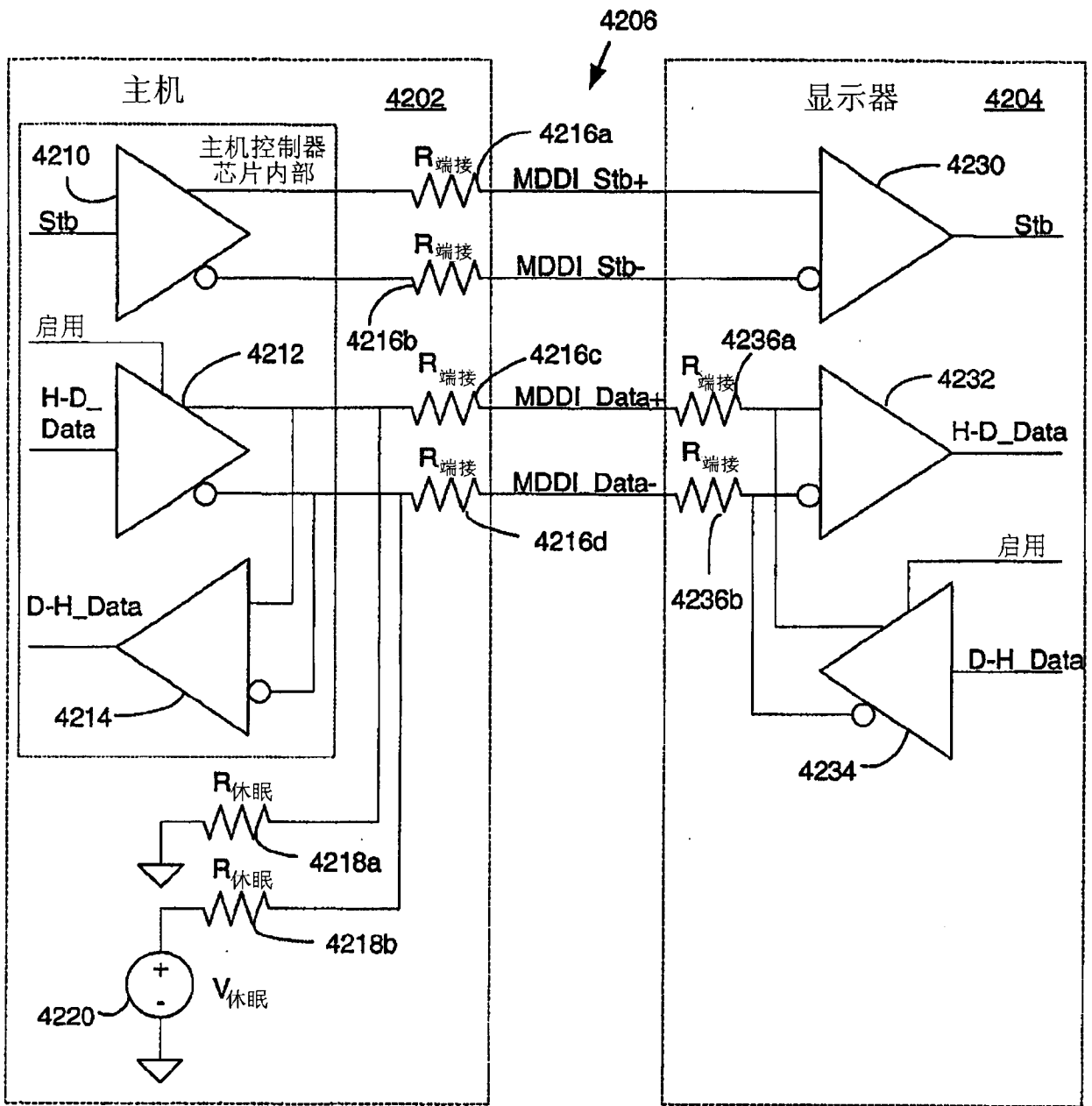


图 42

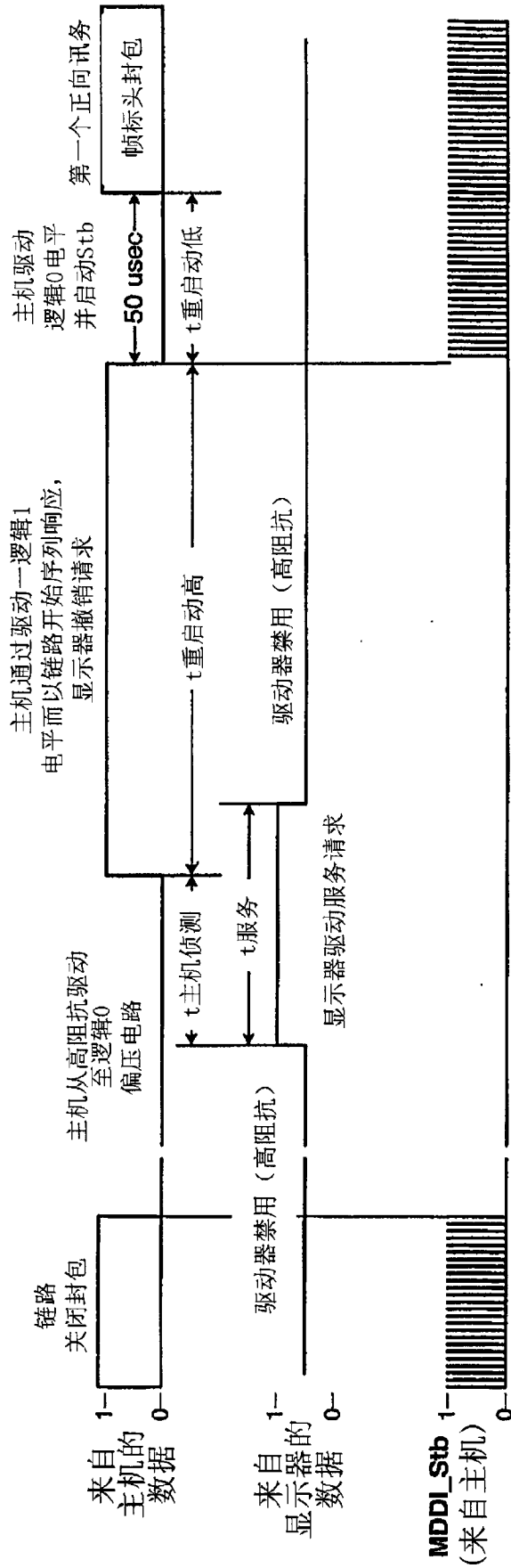


图 43

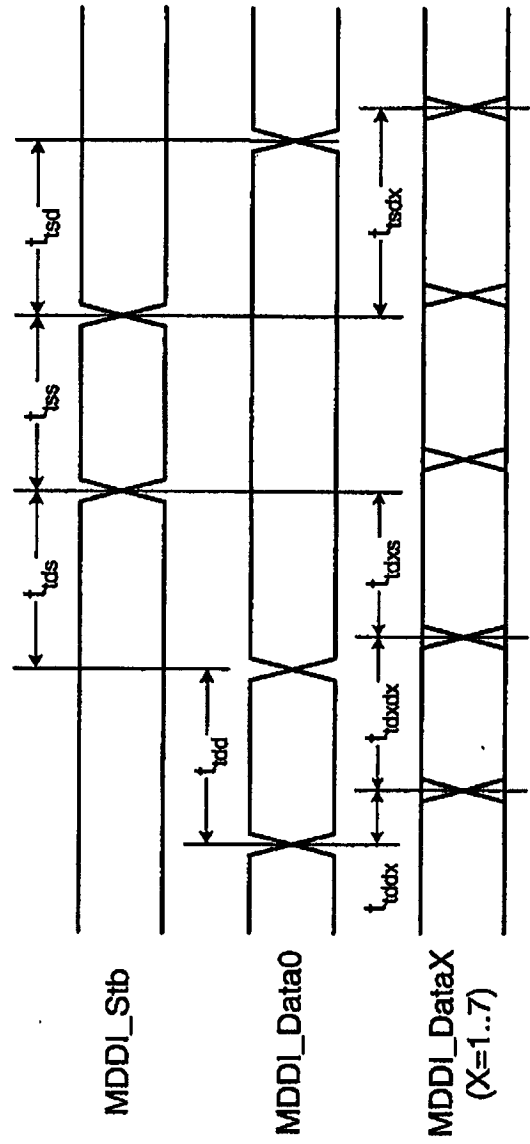


图 44

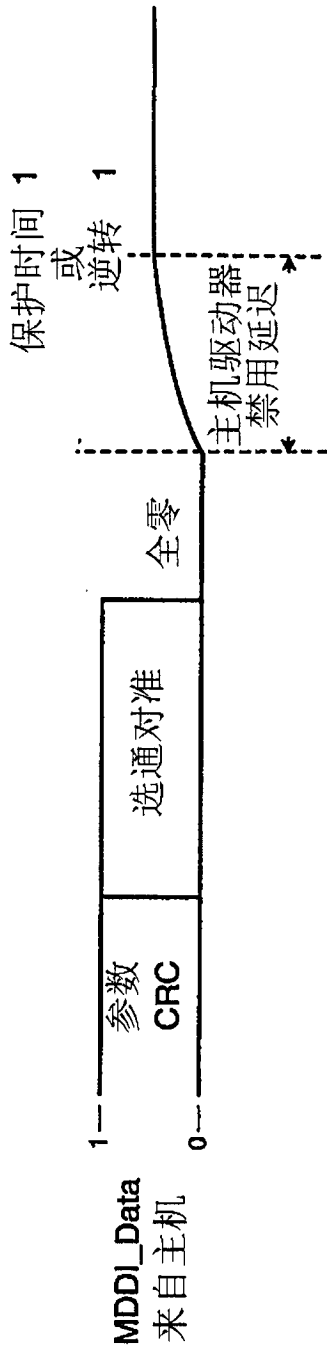


图 45

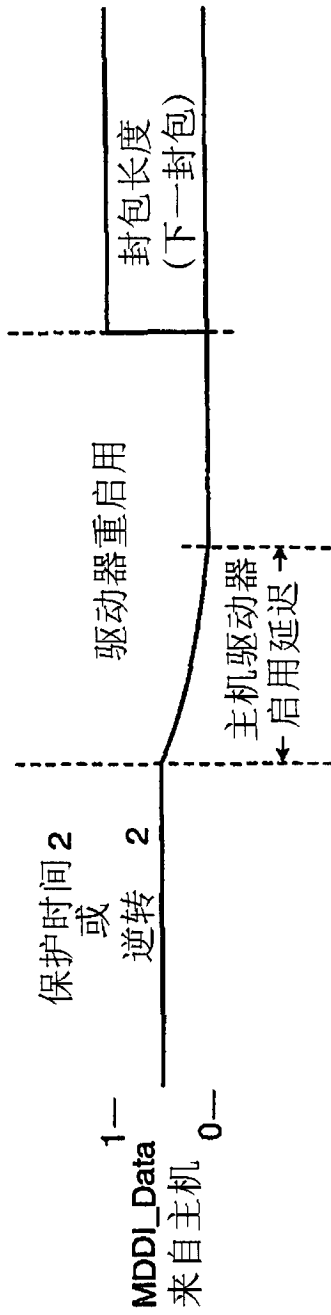


图 46

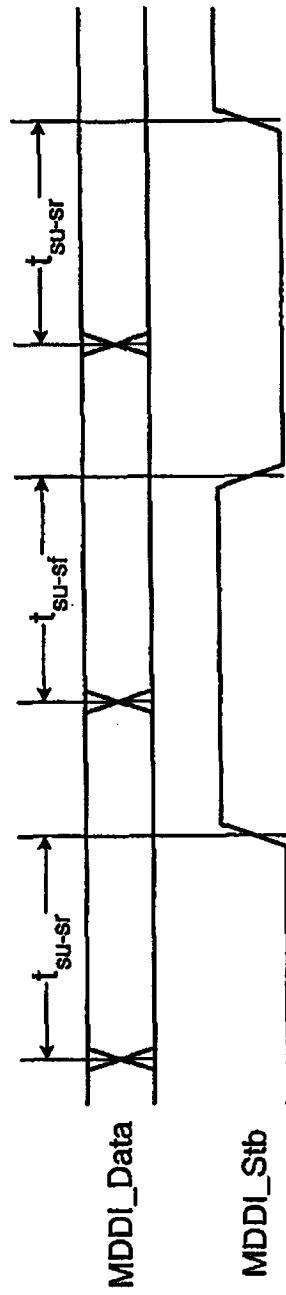


图 47

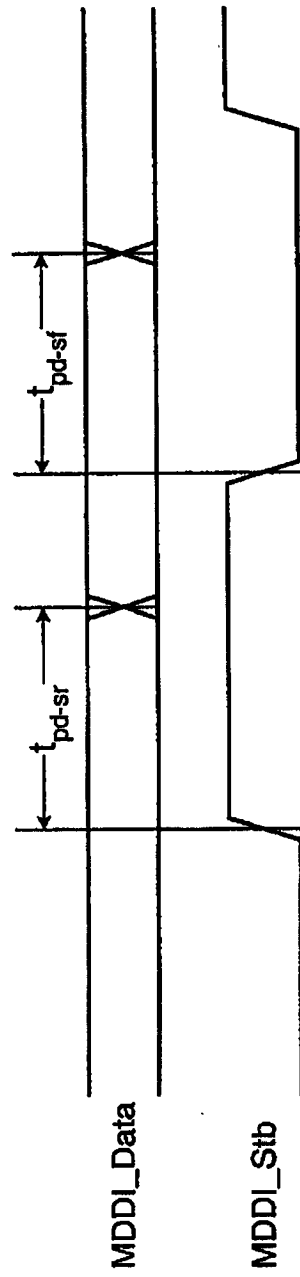
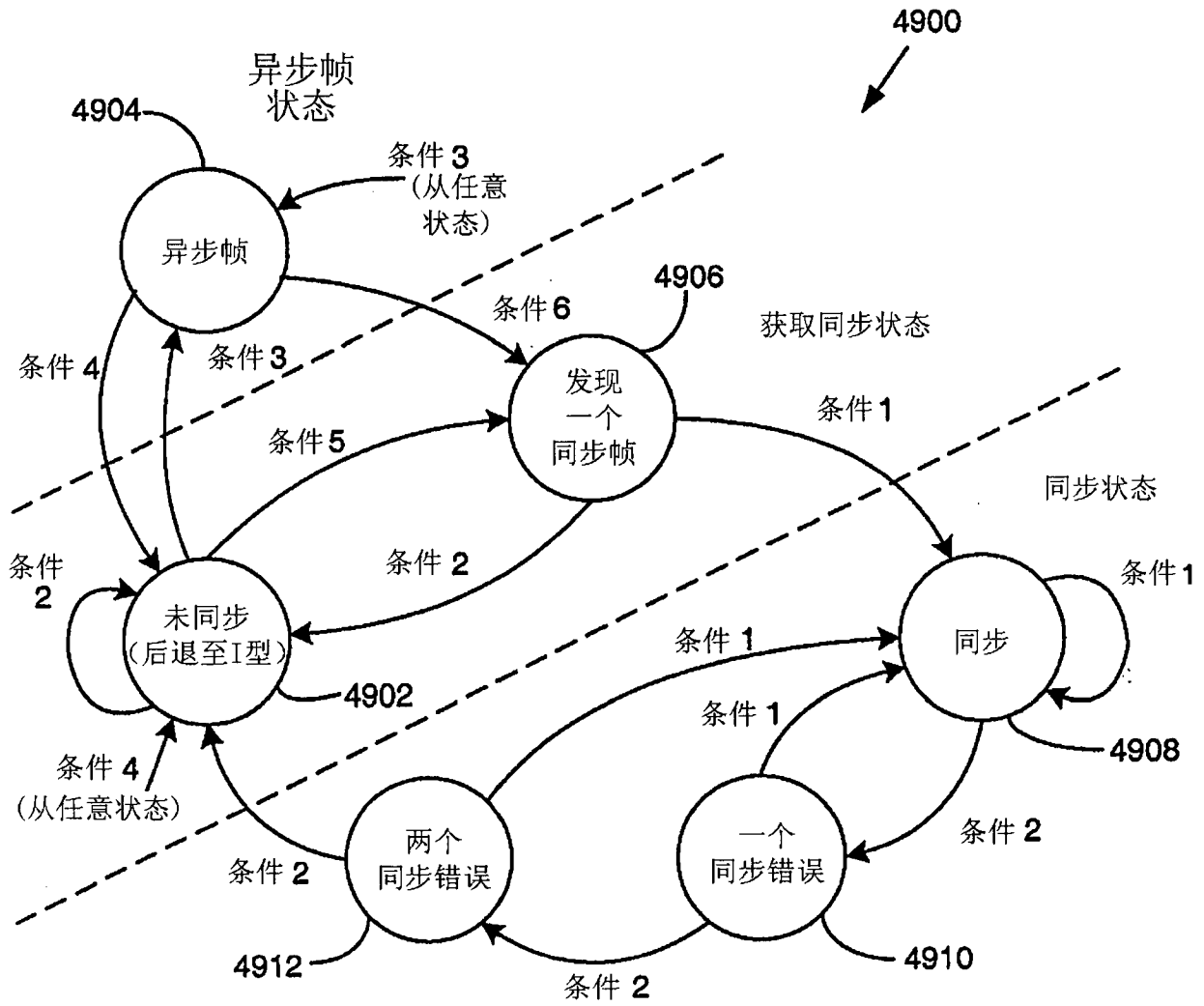


图 48



条件 1 = 子帧标头封包和在帧边界的良好CRC，帧长度>0

条件 2 = 无同步模式或在帧边界的坏的CRC

条件 3 = 发现同步模式，帧长度=0

条件 4 = 接收链路关闭封包

条件 5 = 发现同步模式，帧长度>0

条件 6 = 帧标头封包和良好CRC，帧长度>0

图 49

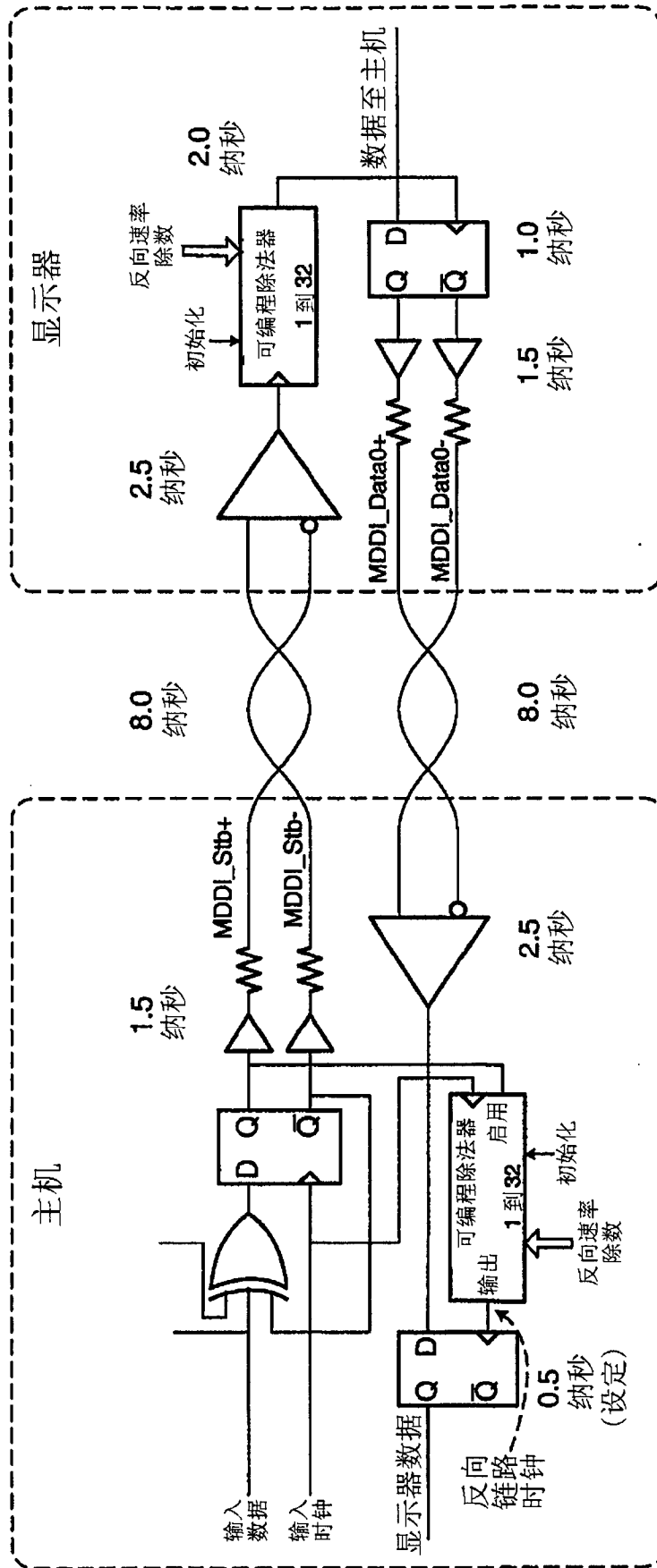


图 50

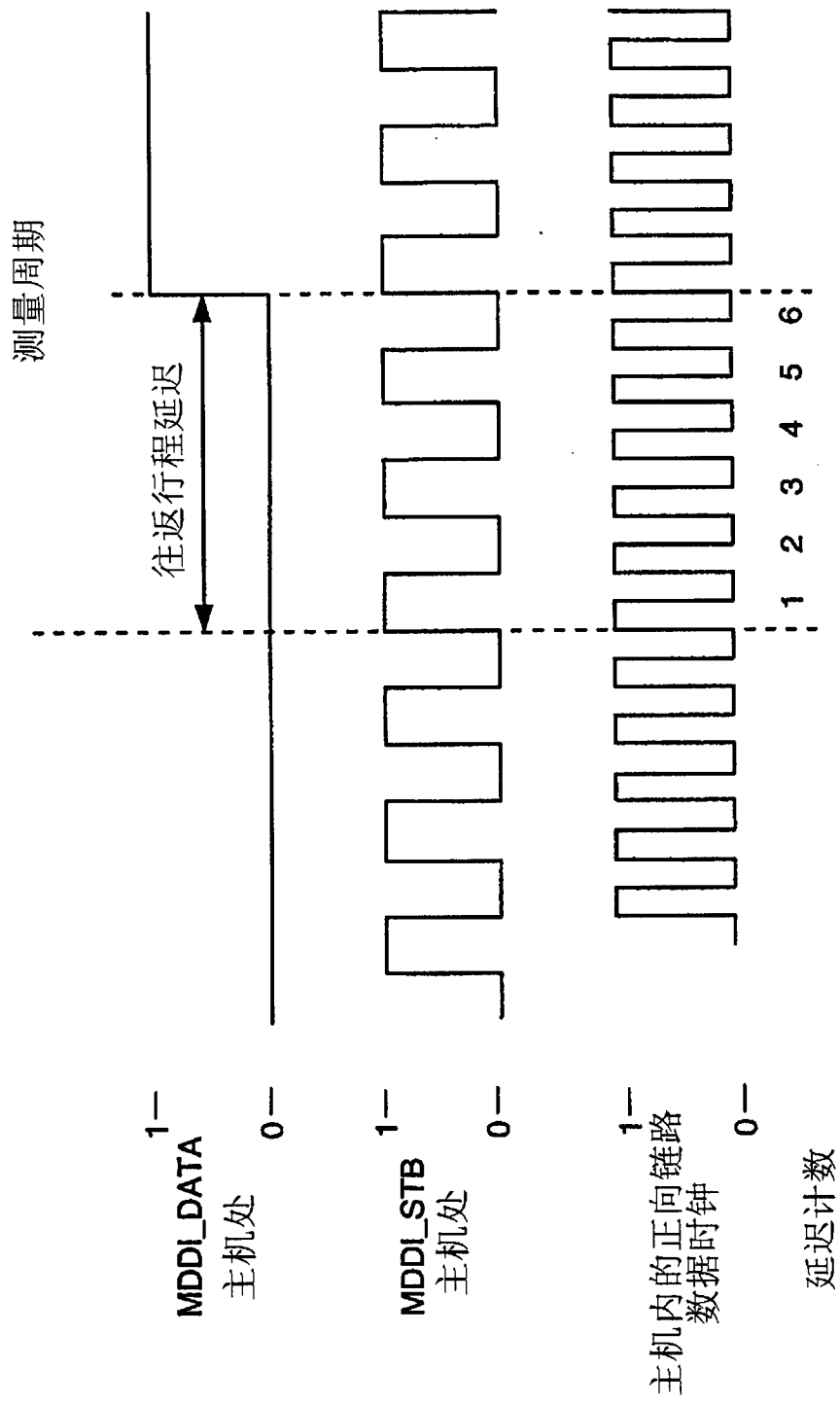


图 51

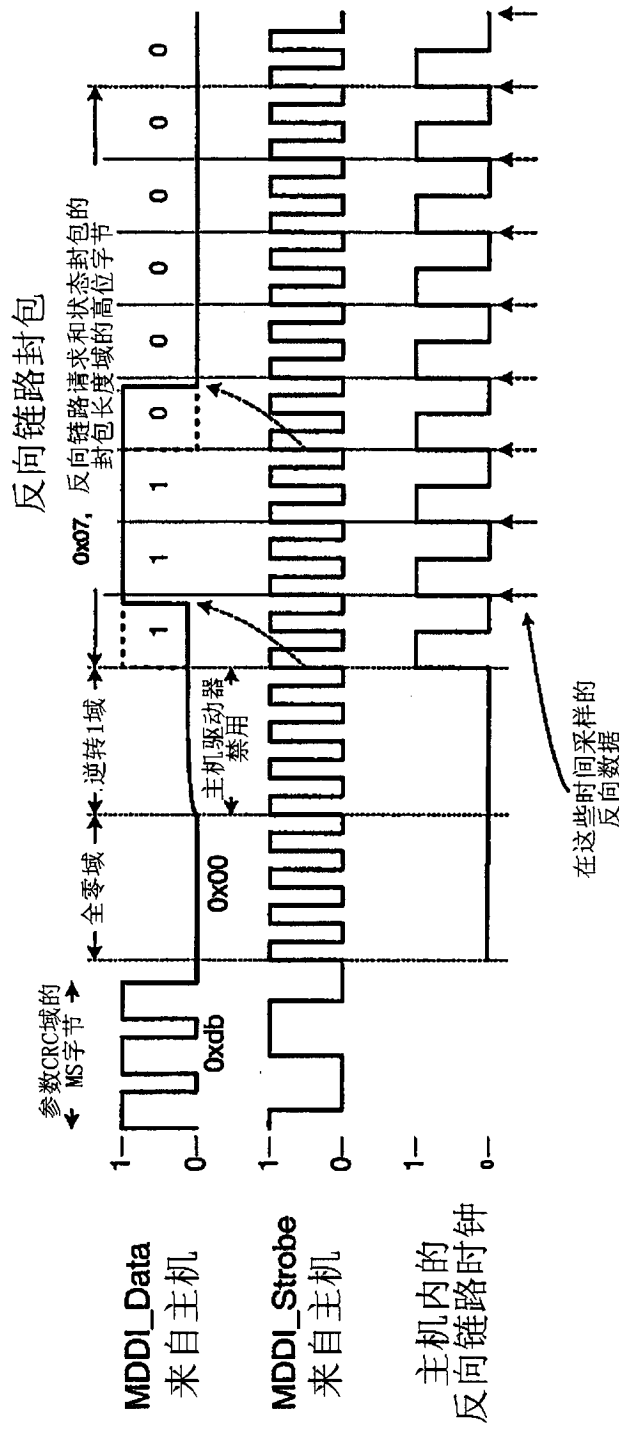


图 52

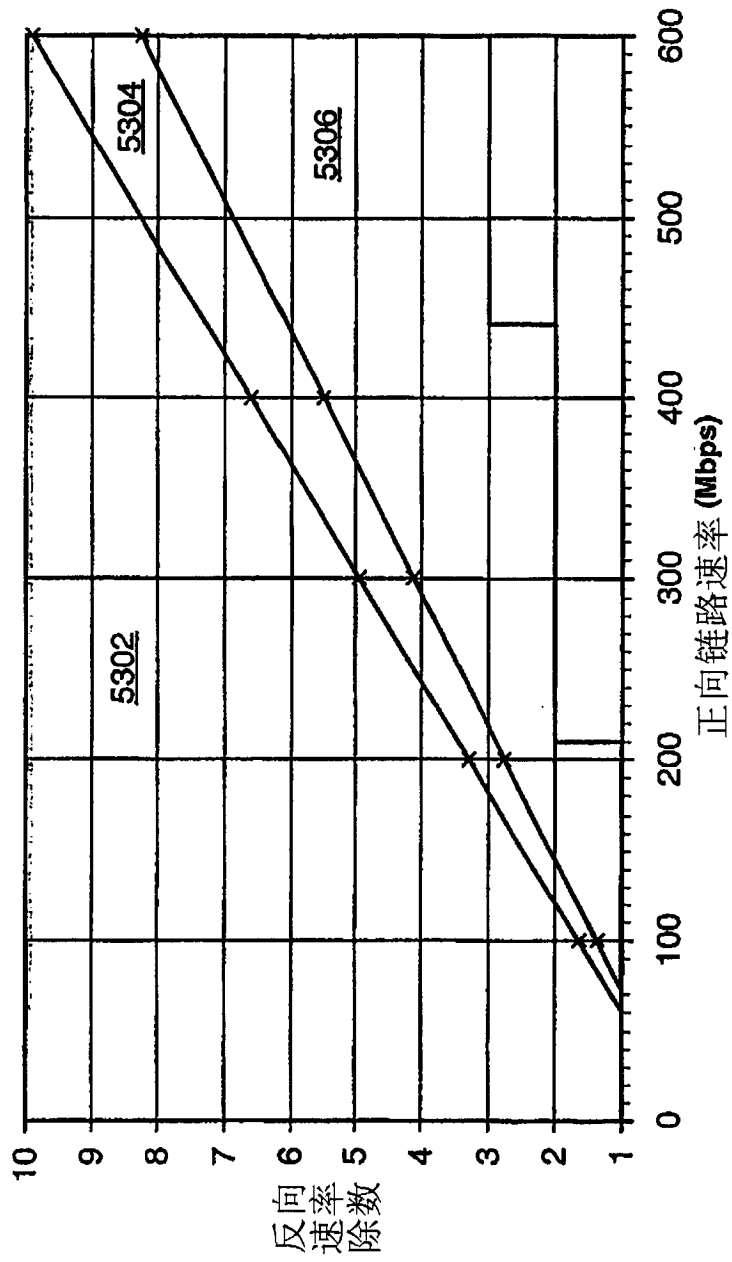


图 53

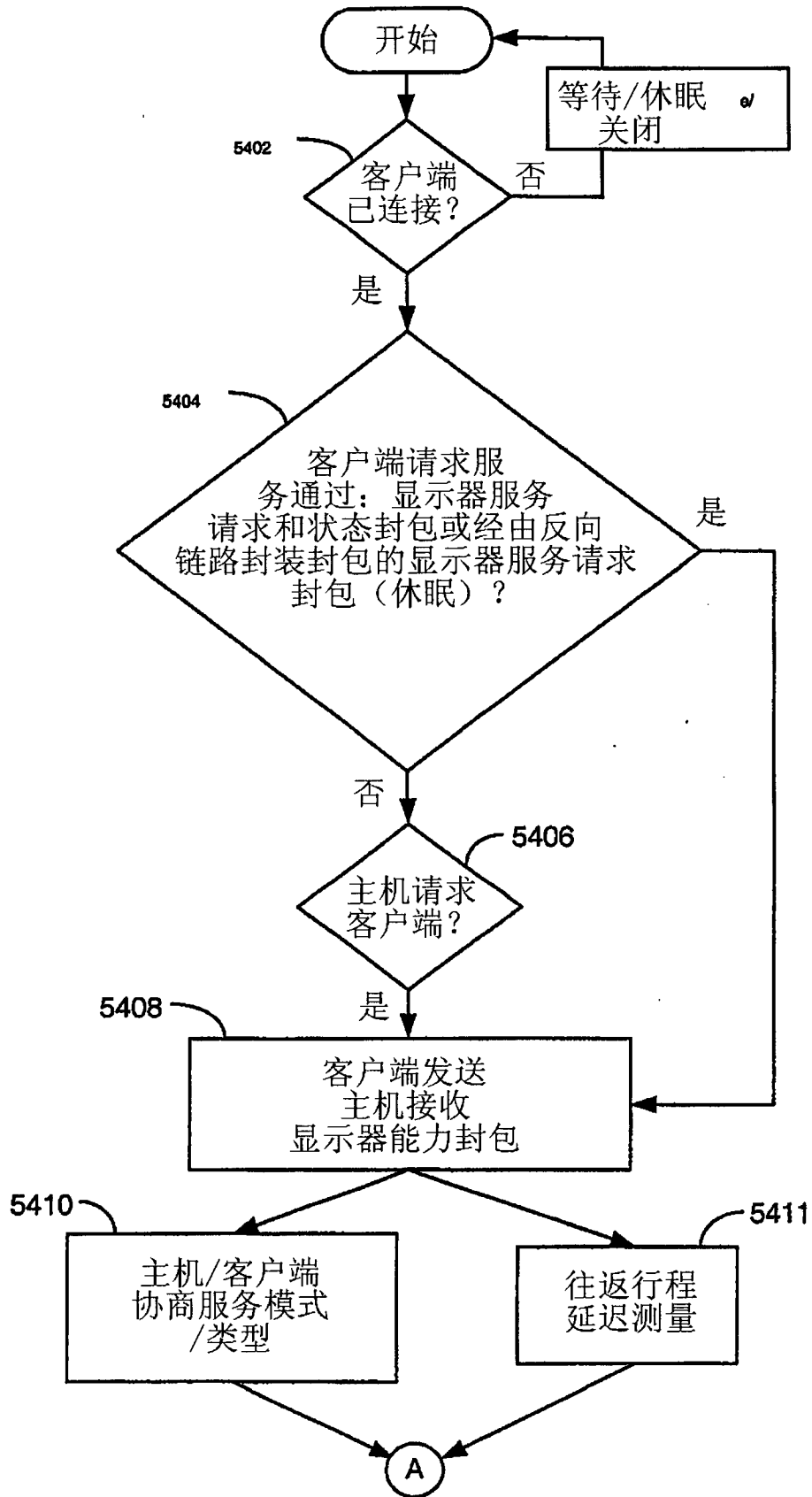


图 54A

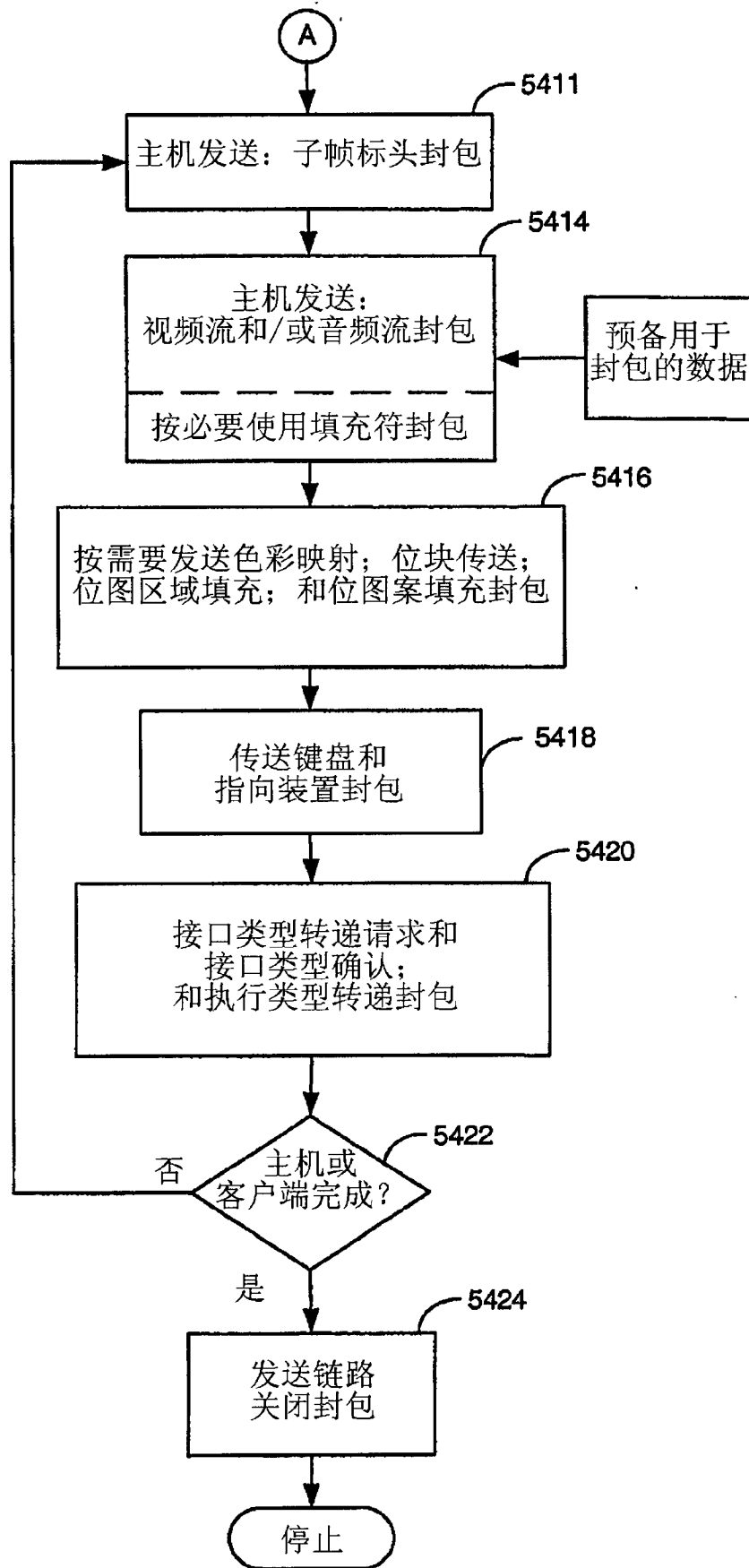


图 54B

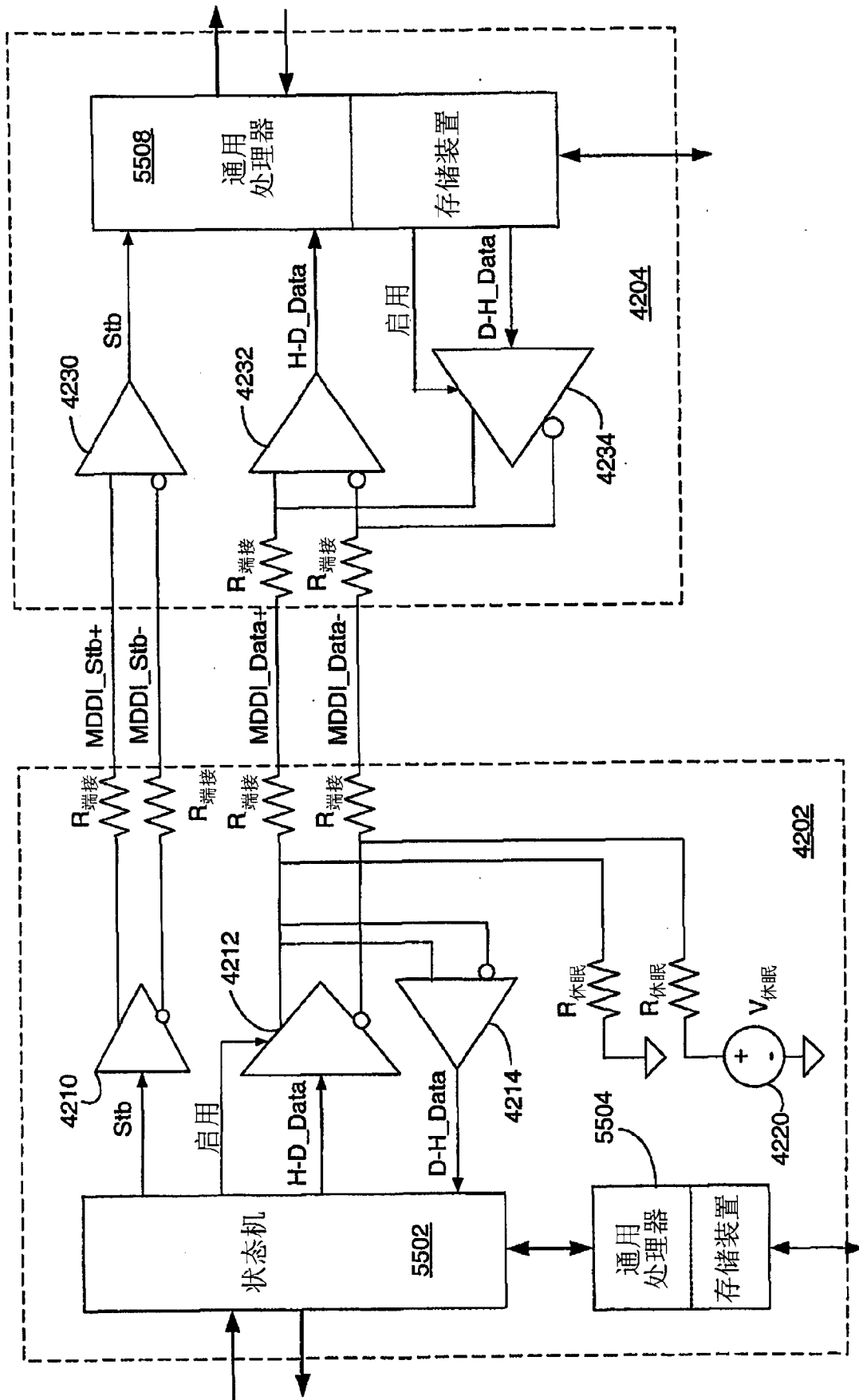


图 55



图 56

512 字节

2 字节

1 字节

2 字节

总数 = 2.4 ns

偏斜

100 ps

300 ps

600 ps

400 ps

300 ps

延迟

1.0 ns 典型
1.5 ns 最大

1.0 ns 典型
2.0 ns 最大

7.7 ns 典型
10.0 ns 最大

2.4 ns 典型
3.3 ns 最大

1.0 ns 典型
1.5 ns 最大

0.1 ns 最小
0.2 ns 最大

tSU = 0.5 ns
tH = 0.1 ns

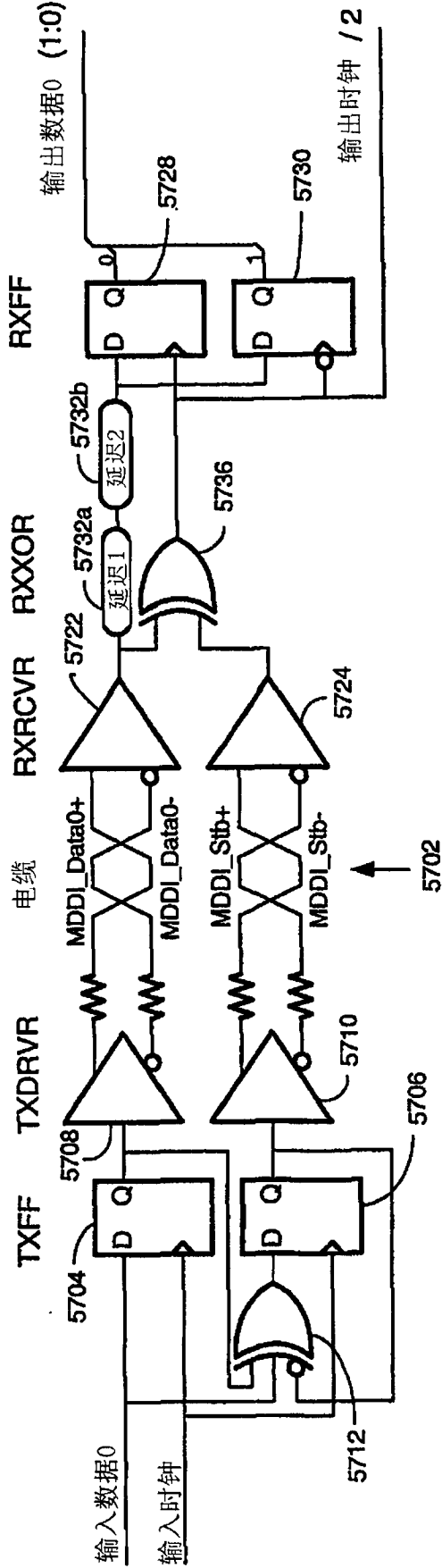


图 57

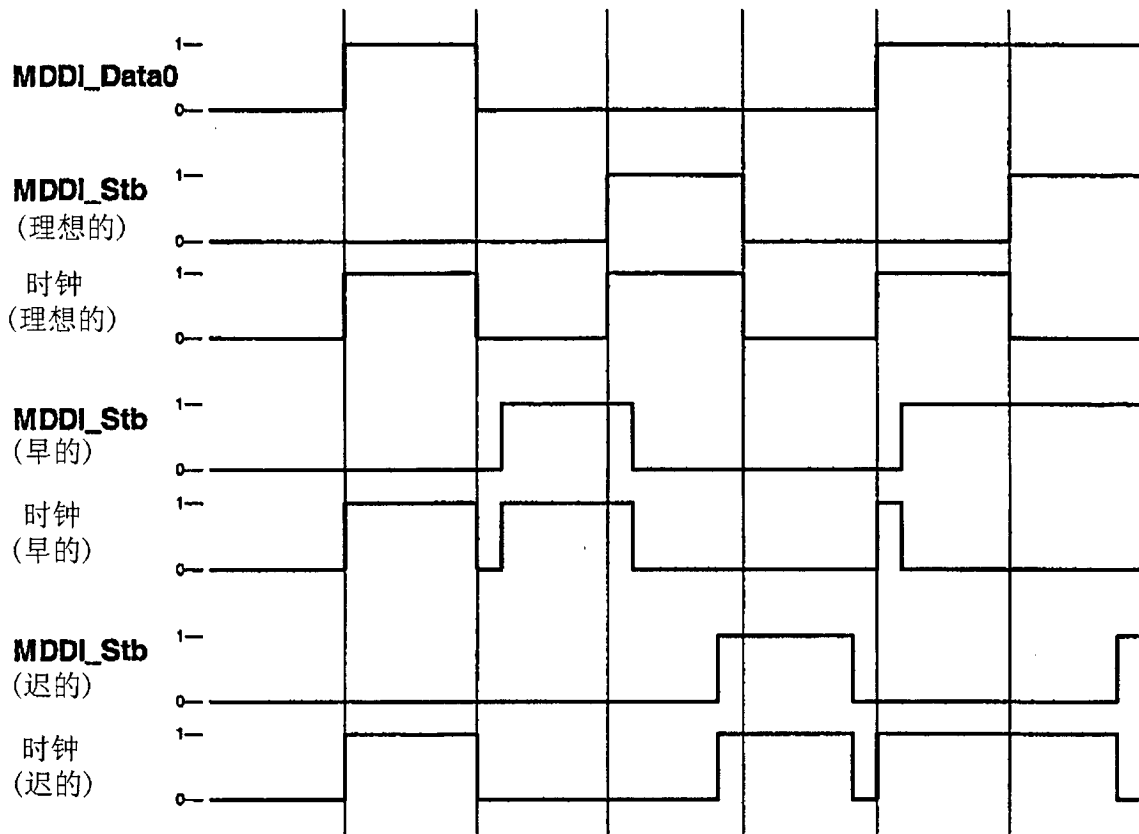


图 58

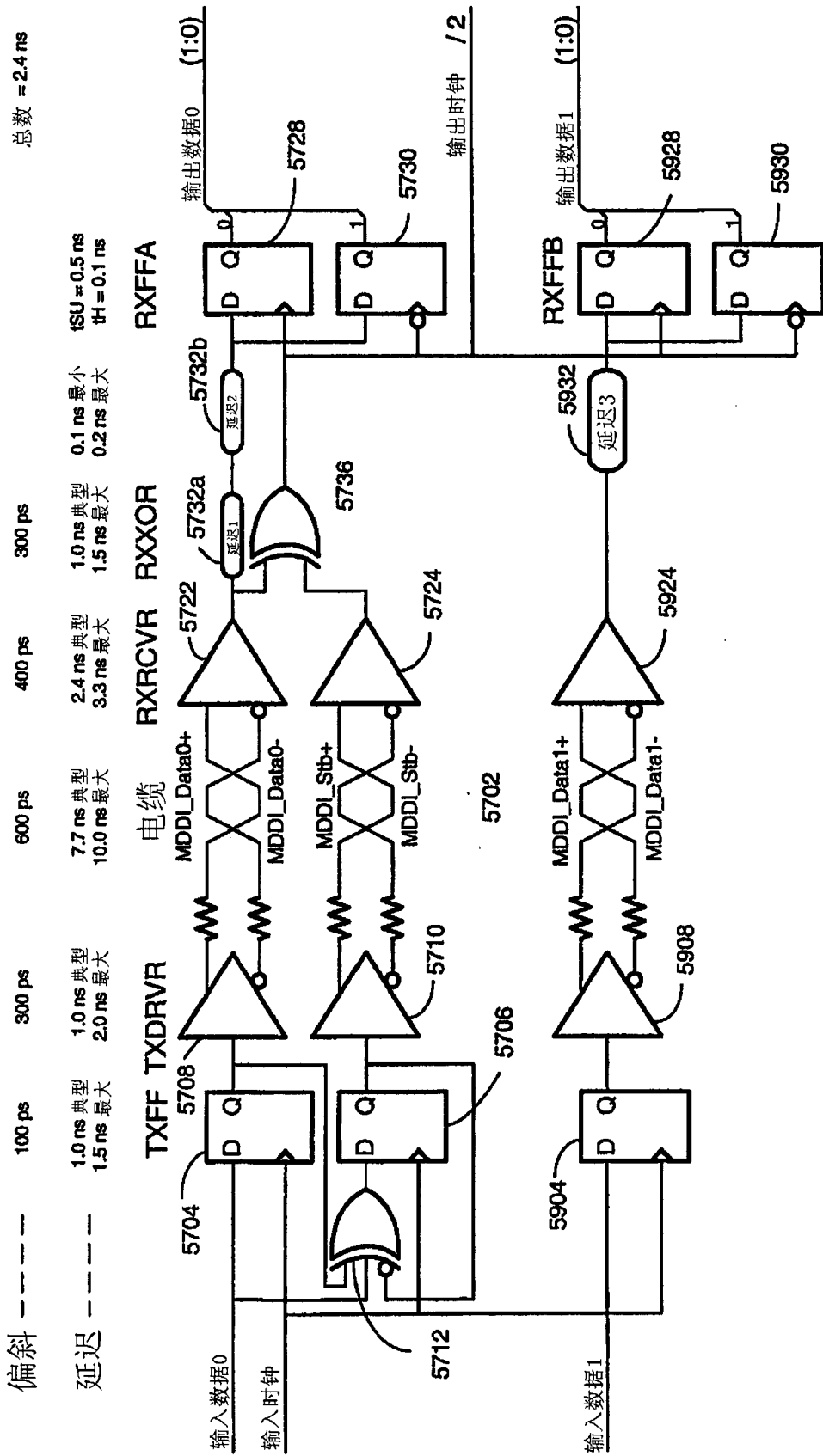


图 59

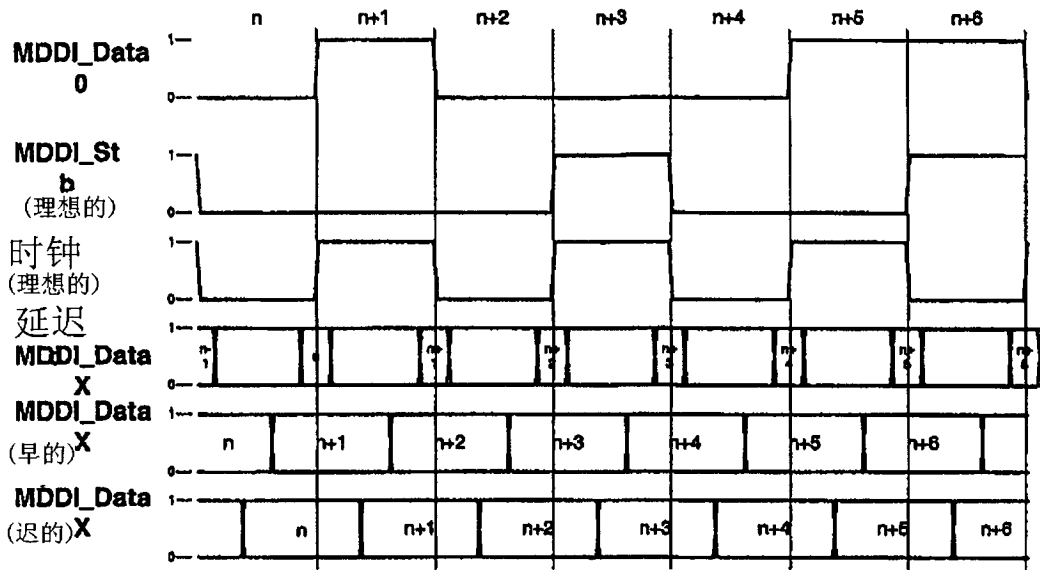


图 60A

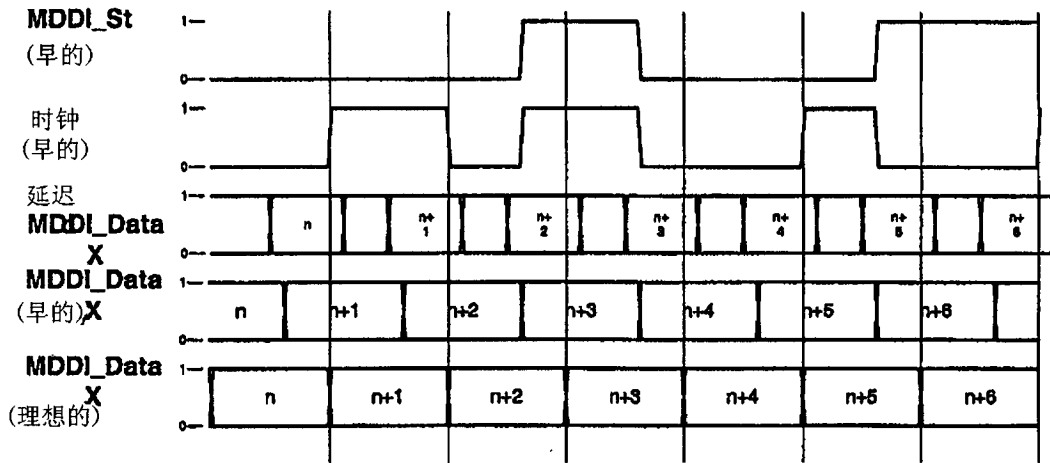


图 60B

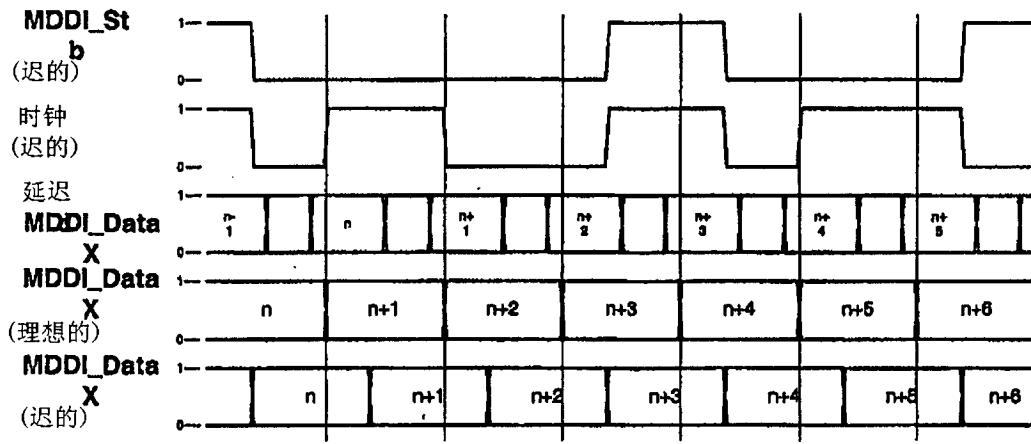


图 60C

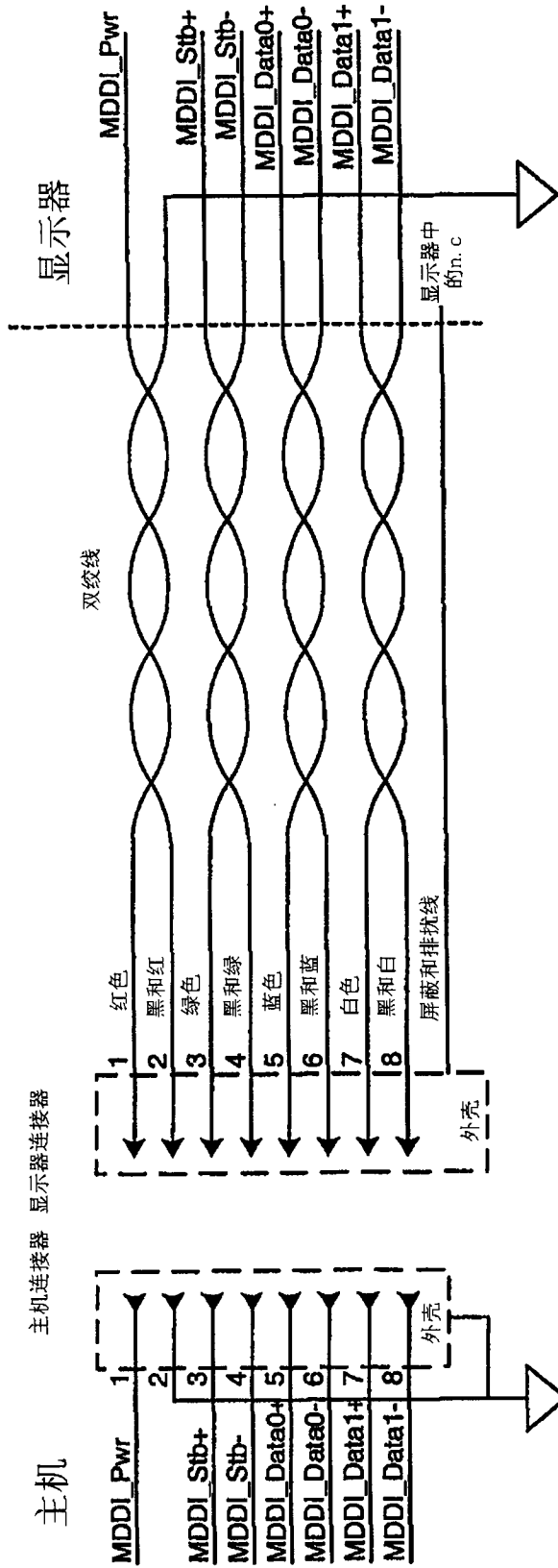


图 61

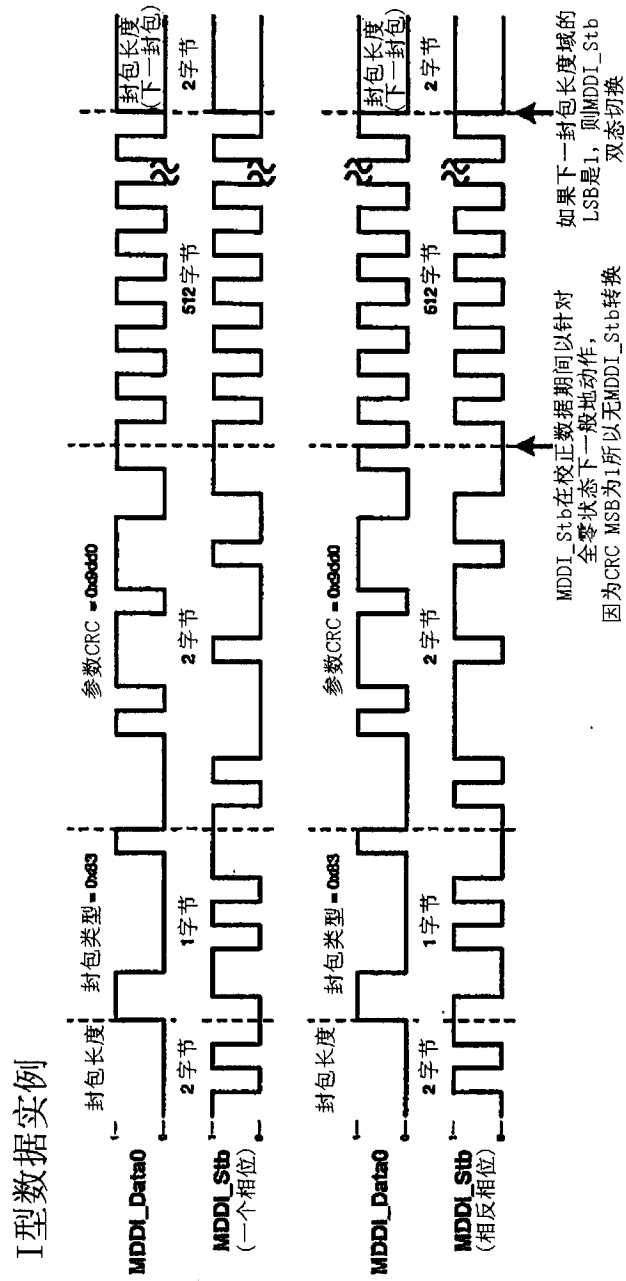


图 62A

II型数据实例

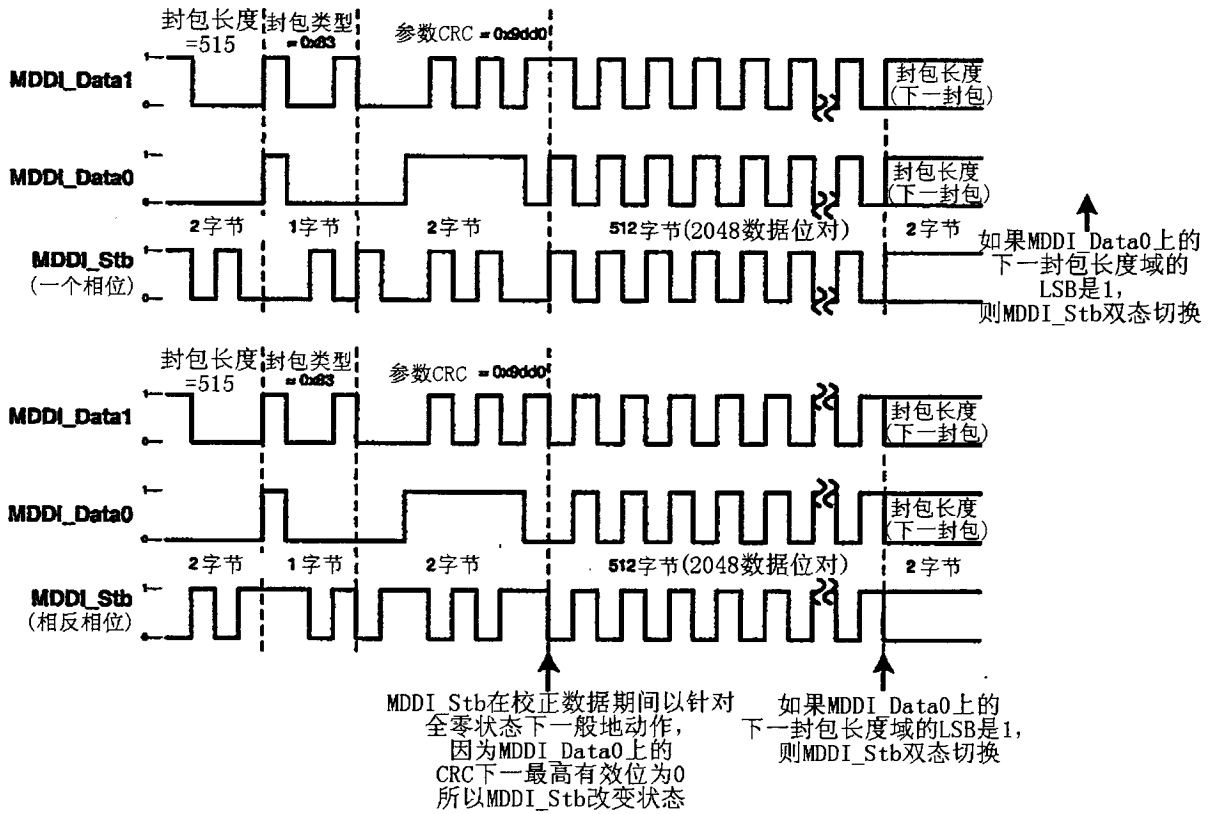
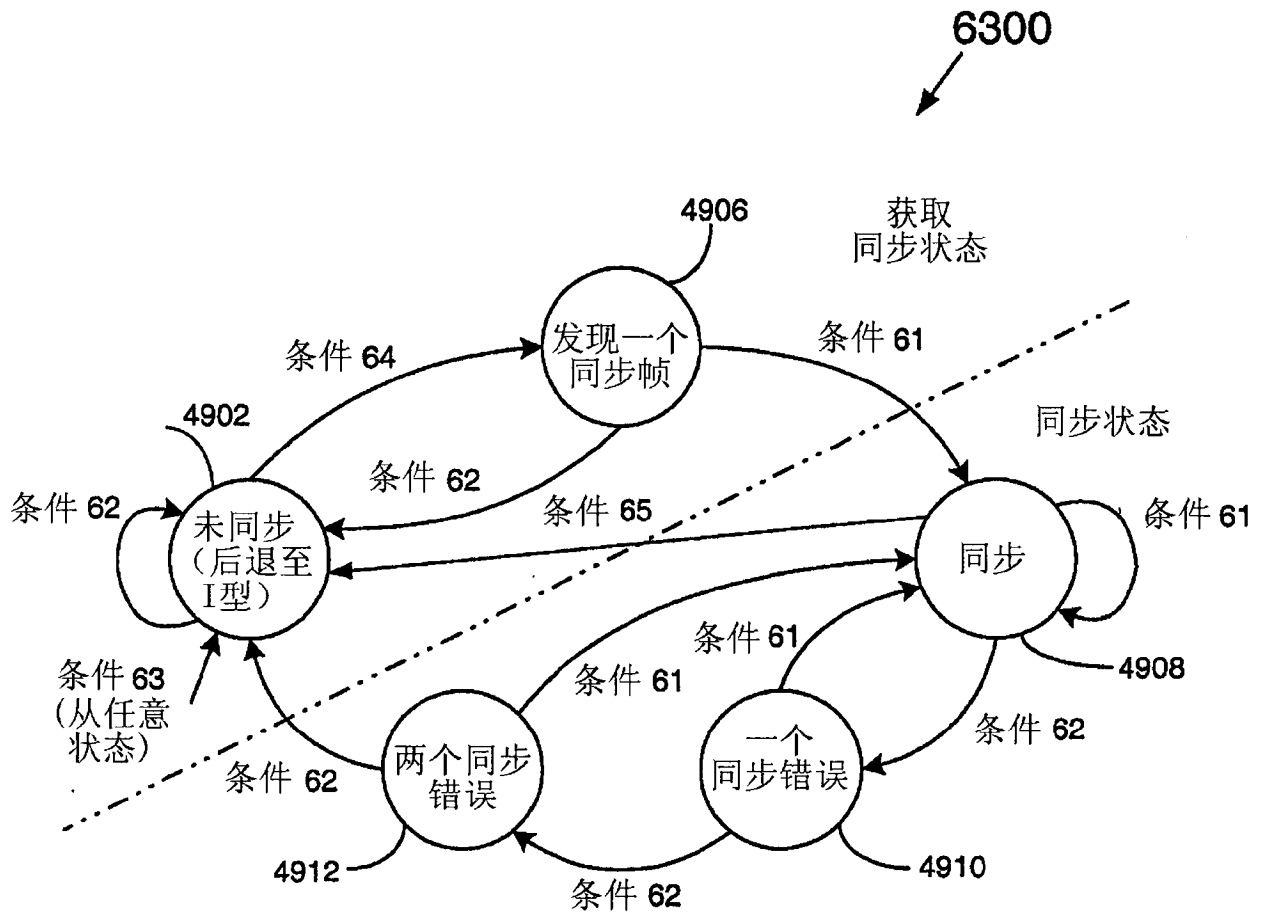


图 62B



条件 61 = 子帧标头封包和在子帧边界的良好CRC
条件 62 = 无同步模式或在子帧边界的坏的CRC
条件 63 = 接收链路关闭封包
条件 64 = 发现同步模式
条件 65 = 唯一字不正确

图 63

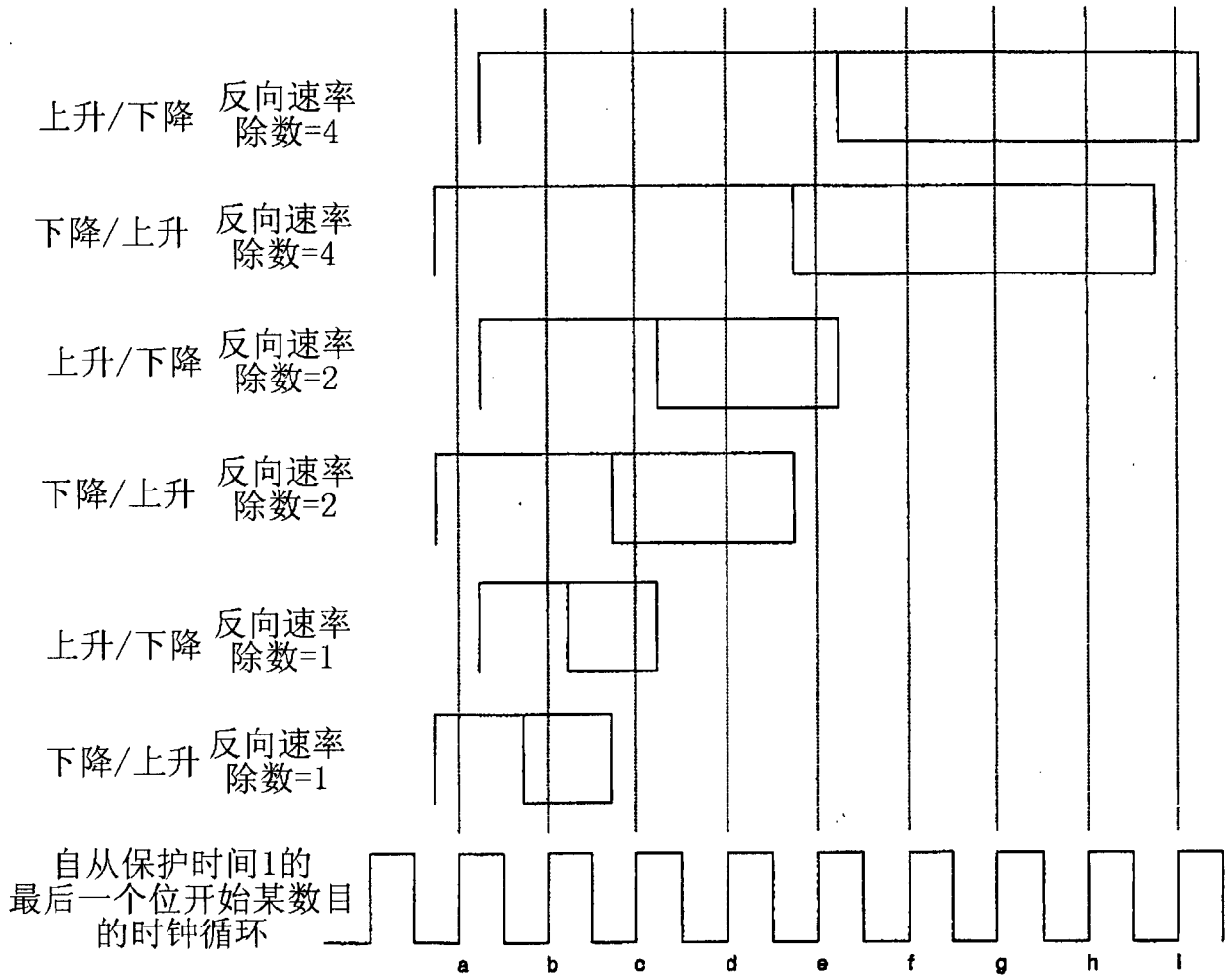


图 64

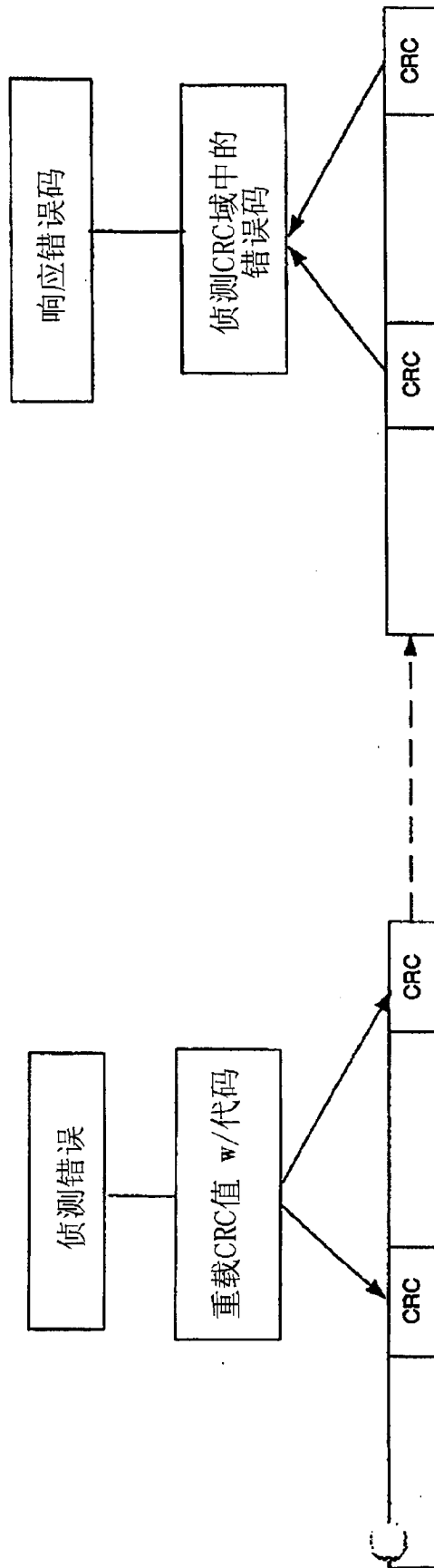


图 65

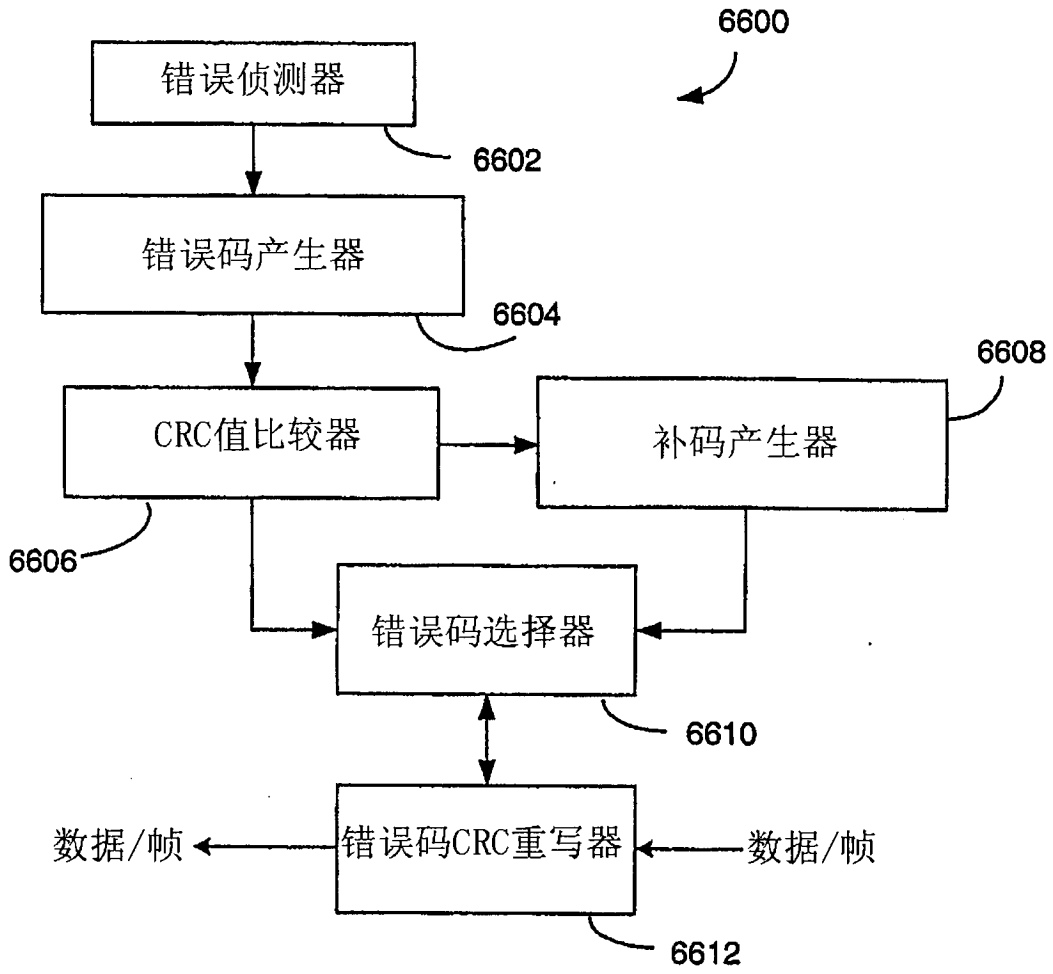


图 66

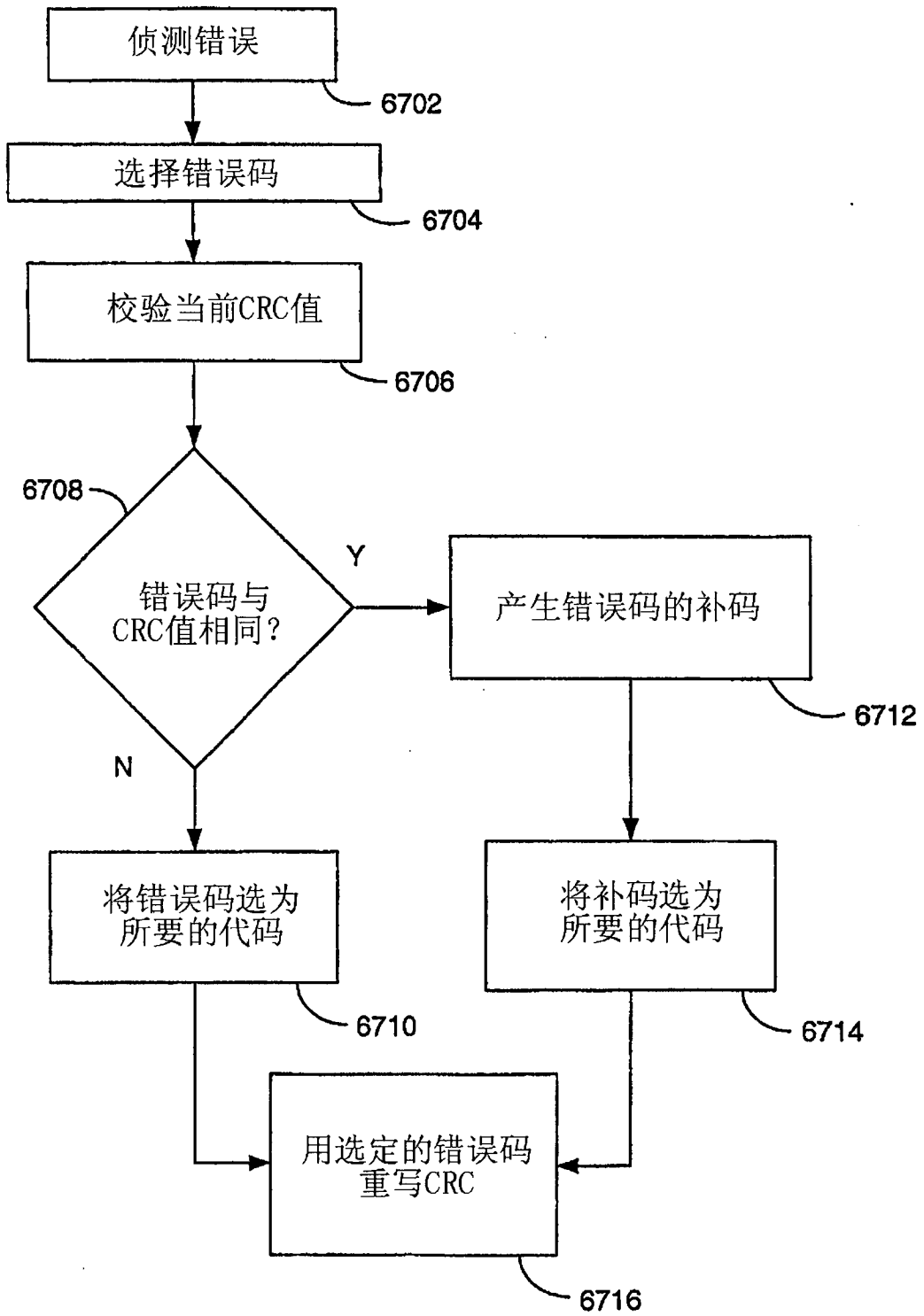


图 67A

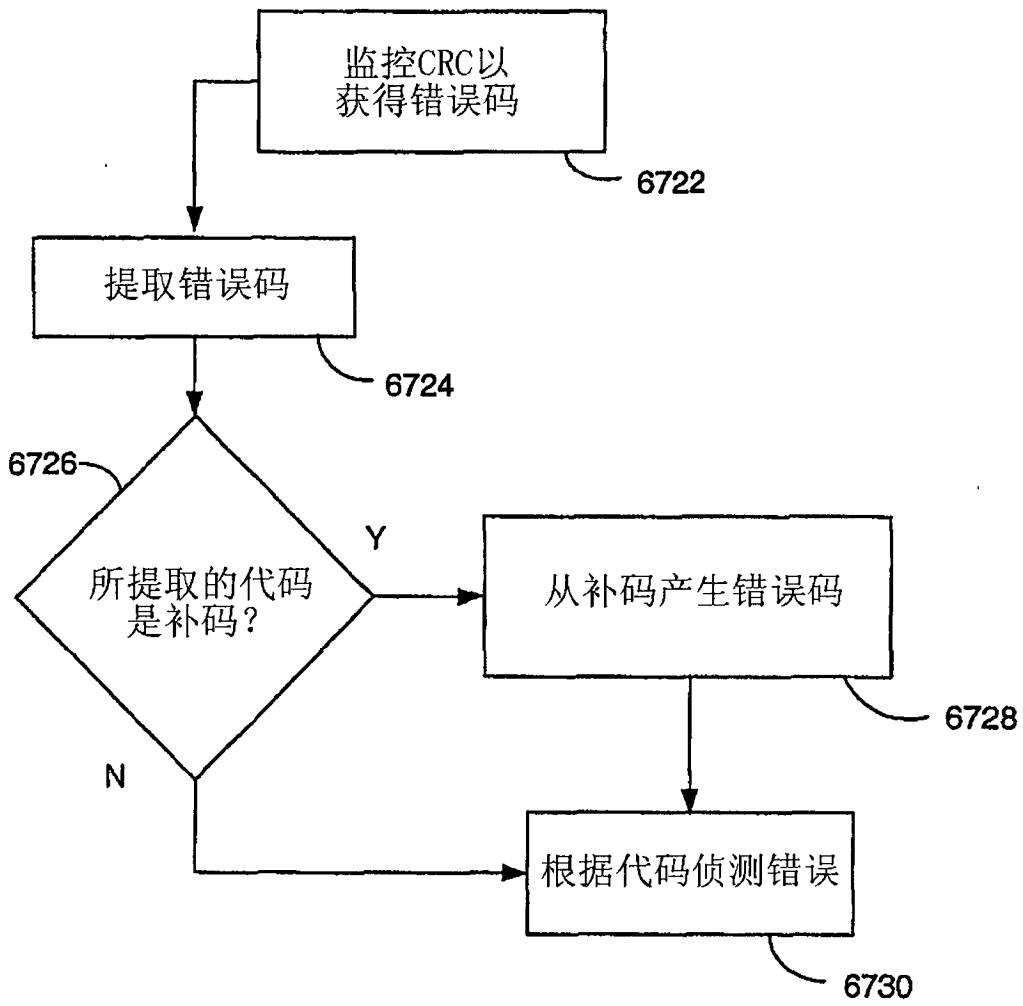


图 67B

