



(12) 发明专利申请

(10) 申请公布号 CN 102497368 A

(43) 申请公布日 2012. 06. 13

(21) 申请号 201110409427. 3

(51) Int. Cl.

(22) 申请日 2004. 12. 08

H04L 29/06 (2006. 01)

(30) 优先权数据

60/527, 996 2003. 12. 08 US

(62) 分案原申请数据

200480041443. X 2004. 12. 08

(71) 申请人 高通股份有限公司

地址 美国加利福尼亚

(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

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 张扬 王英

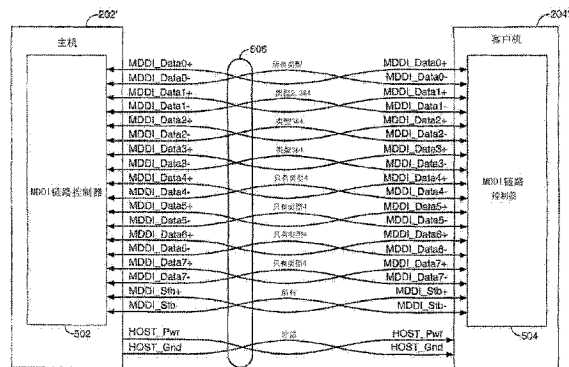
权利要求书 3 页 说明书 102 页 附图 62 页

(54) 发明名称

具有改进链路同步的高数据速率接口

(57) 摘要

一种数据接口,其使用链接在一起的分组结构来经由通信路径在主机和客户机之间传送数字数据,所述分组结构链接起来而形成用于传递数字控制和呈现数据的预选集合的通信协议。所述信号协议由链路控制器使用,所述链路控制器被配置成生成、传输并且接收形成通信协议的分组,并且使数字数据形成为一种或多种类型的数据分组,其中至少一个链路控制器驻留在主机设备中并且通过通信路径耦合至客户机。所述接口经由短程“串行”类型的数据链路提供划算的、低功率、双向、高速数据传送机制,这使其能够利用那些特别适用于将诸如可佩带的微型显示器的显示器元件连接到便携式计算机以及无线通信设备的小型连接器和细软电缆来实现。



1. 在移动显示数字接口 (MDDI) 通信系统中为反向数据链路封装分组提供安全容限的方法,所述方法包括下列步骤:

用往返行程延迟测量分组计算主机和客户机之间的往返行程延迟;

提供用于启用主机驱动器的第一预定时段;

提供用于禁用客户机驱动器的第二预定时段;

在所述反向数据链路封装分组中引入转向 2 字段长度,所述转向 2 字段长度大于计算出的往返行程延迟、所述第一预定时段和所述第二预定时段三者之和。

2. 在移动显示数字接口 (MDDI) 通信系统中为反向数据链路封装分组提供安全容限的系统,所述系统包括:

计算模块,用往返行程延迟测量分组计算主机和客户机之间的往返行程延迟;

第一预定时段提供模块,提供用于启用主机驱动器的第一预定时段;

第二预定时段提供模块,提供用于禁用客户机驱动器的第二预定时段;

引入模块,在所述反向数据链路封装分组中引入转向 2 字段长度,所述转向 2 字段长度大于计算出的往返行程延迟、所述第一预定时段和所述第二预定时段三者之和。

3. 计算机程序产品,包括:

计算机可读介质,包括:

用于在移动显示数字接口 (MDDI) 通信系统中为反向数据链路封装分组提供安全容限的代码,所述计算机代码包括:

计算代码,用往返行程延迟测量分组计算主机和客户机之间的往返行程延迟;

第一预定时段提供代码,提供用于启用主机驱动器的第一预定时段;

第二预定时段提供代码,提供用于禁用客户机驱动器的第二预定时段;

引入代码,在所述反向数据链路封装分组中引入转向 2 字段长度,所述转向 2 字段长度大于计算出的往返行程延迟、所述第一预定时段和所述第二预定时段三者之和。

4. 在移动显示数字接口 (MDDI) 通信系统中用于确保在反向链路封装分组传输期间所有数据信号都得到驱动的方法,所述方法包括下列步骤:

在所述反向链路封装分组的转向 1 时段的最后一个比特之前,禁用主机的数据线驱动器;

在所述反向链路封装分组的转向 1 时段开始时,启用客户机的线驱动器。

5. 权利要求 4 的方法,还包括下列步骤:

在从所述反向链路封装分组的全零 1 字段开始到全零 2 字段结束之间的每个比特边界处,都翻转选通信号。

6. 权利要求 4 的方法,其中,所述转向 1 时段包括:

用于让所述主机的线驱动器禁用和所述客户机的线驱动器启用基本同时进行的时间段。

7. 权利要求 4 的方法,其中,所述转向 1 时段包括:

大于计算出的往返行程延迟、主机驱动器禁用时间和客户机驱动器启用时间三者之和的时间段。

8. 在移动显示数字接口 (MDDI) 通信系统中用于确保在反向链路封装分组传输期间所有数据信号都得到驱动的系统,所述方法包括下列步骤:

禁用模块,在所述反向链路封装分组的转向 1 时段的最后一个比特之前,禁用主机的数据线驱动器;

启用模块,在所述反向链路封装分组的转向 1 时段开始时,启用客户机的线驱动器。

9. 权利要求 8 的系统,还包括:

翻转模块,在从所述反向链路封装分组的全零 1 字段开始到全零 2 字段结束之间的每个比特边界处,都翻转选通信号。

10. 权利要求 8 的系统,其中,所述转向 1 时段包括:

用于让所述主机的线驱动器禁用和所述客户机的线驱动器启用基本同时进行的时间段。

11. 权利要求 8 的系统,其中,所述转向 1 时段包括:

大于计算出的往返行程延迟、主机驱动器禁用时间和客户机驱动器启用时间三者之和的时间段。

12. 权利要求 8 的系统,其中,所述转向 1 时段包括:

大于计算出的往返行程延迟、主机驱动器禁用时间和客户机驱动器启用时间三者之和减去所述转向 1 时段持续时间的时间段。

13. 计算机程序产品,包括:

计算机可读介质,包括:

在移动显示数字接口 (MDDI) 通信系统中用于确保在反向链路封装分组传输期间所有数据信号都得到驱动的代码,所述计算机代码包括:

禁用代码,在所述反向链路封装分组的转向 1 时段的最后一个比特之前,禁用主机的数据线驱动器;

启用代码,在所述反向链路封装分组的转向 1 时段开始时,启用客户机的线驱动器。

14. 权利要求 13 的计算机程序产品,还包括:

翻转代码,在从所述反向链路封装分组的全零 1 字段开始到全零 2 字段结束之间的每个比特边界处,都翻转选通信号。

15. 权利要求 13 的计算机程序产品,其中,所述转向 1 时段包括:

用于让所述主机的线驱动器禁用和所述客户机的线驱动器启用基本同时进行的时间段。

16. 权利要求 13 的计算机程序产品,其中,所述转向 1 时段包括:

大于计算出的往返行程延迟、主机驱动器禁用时间和客户机驱动器启用时间三者之和的时间段。

17. 在移动显示数字接口 (MDDI) 通信系统中用于确保在反向链路封装分组传输期间所有数据信号都得到驱动的方法,所述方法包括下列步骤:

在所述反向链路封装分组的转向 2 时段的最后一个比特之前,禁用客户机的数据线驱动器;

在所述反向链路封装分组的转向 2 时段结束前,启用主机的线驱动器。

18. 权利要求 17 的方法,其中,在从所述反向链路封装分组的全零 1 字段开始到全零 2 字段结束之间的每个比特边界处,都翻转选通信号。

19. 权利要求 17 的方法,其中,在转向 2 时段开始后再经过与所测量往返行程延迟相等

的一段时间之后,启用所述主机的线驱动器。

20. 在移动显示数字接口 (MDDI) 通信系统中用于确保在反向链路封装分组传输期间所有数据信号都得到驱动的系统,所述方法包括下列步骤:

在所述反向链路封装分组的转向 2 时段的最后一个比特之前,禁用客户机的数据线驱动器;

在所述反向链路封装分组的转向 2 时段结束前,启用主机的线驱动器。

21. 权利要求 20 的系统,其中,在从所述反向链路封装分组的全零 1 字段开始到全零 2 字段结束之间的每个比特边界处,都翻转选通信号。

22. 权利要求 20 的系统,其中,在转向 2 时段开始后再经过与所测量往返行程延迟相等的一段时间后,启用所述主机的线驱动器。

23. 计算机程序产品,包括:

计算机可读介质,包括:

在移动显示数字接口 (MDDI) 通信系统中用于确保在反向链路封装分组传输期间所有数据信号都得到驱动的代码,所述计算机代码包括:

禁用代码,在所述反向链路封装分组的转向 2 时段的最后一个比特之前,禁用客户机的数据线驱动器;

启用代码,在所述反向链路封装分组的转向 2 时段结束前,启用主机的线驱动器。

24. 权利要求 20 的计算机程序产品,其中,在从所述反向链路封装分组的全零 1 字段开始到全零 2 字段结束之间的每个比特边界处,都翻转选通信号。

25. 权利要求 20 的计算机程序产品,其中,在转向 2 时段开始后再经过与所测量往返行程延迟相等的一段时间后,启用所述主机的线驱动器。



## 具有改进链路同步的高数据速率接口

[0001] 本申请是 2004 年 12 月 8 日提交的申请号为 200480041443.X、发明名称为“具有改进链路同步的高数据速率接口”的分案申请。根据 35U. S. C § 119 的优先权要求

[0002] 本专利申请要求于 2003 年 12 月 8 日提交、标题为“Switchable Threshold Differential Interface”、转让给本发明受让人的第 60/527, 996 号临时申请的优先权, 后一篇申请明确地以引用方式并入此处。

### 技术领域

[0003] 在本公开内容中, 本发明的实施例涉及用于在主机设备和客户机设备之间以高数据速率进行信号传递或者信号传送的一种数字信号协议和处理过程。更具体地说, 本公开内容涉及一种技术, 这种技术使用具有内部和外部设备应用的低功率、高数据速率传送机制, 把多媒体以及其它类型的数字信号从主机或者控制器设备传送到客户机设备, 以便呈现或者显示给最终用户。

### 背景技术

[0004] 近几年来, 计算机、电子游戏相关产品和各种视频技术 (例如 DVD 和高清晰度 VCR) 获得了显著的进展, 从而能够向这种装置的最终用户提供分辨率不断提高的静止、视频、视频点播和图形图像, 甚至还包括某些类型的文本。这些进展继而要求采用更高分辨率的电子观看设备, 例如高清晰度视频监视器、HDTV 监视器或者专用图像投影元件。例如, 在使用 CD 类型的声音再现、DVD 以及同样具有相关音频信号输出的其它设备时, 把这种可视图像与高清晰度或者高质量的音频数据相结合, 从而为最终用户产生更加逼真、内容更丰富或者更加真切的多媒体体验。另外, 已经开发了诸如 MP3 播放器之类的高移动性、高质量的音响系统和音乐传输机制, 以便向最终用户呈现单纯音频。这已经导致从计算机到电视甚至电话这种商业电子设备的典型用户, 对高品质或者一流质量的输出越来越习惯和并且越来越多的期望高品质或者优质的输出。

[0005] 在典型的视频呈现情形中, 涉及电子产品的视频数据通常是用当前技术以每秒一到数十千比特数量级上的速率来传送的, 这一速率被恰当地称为慢速或中速。然后, 把这些数据缓存或者储存在临时或者长期存储器设备中, 以便在所期望的观看设备上延迟 (稍后) 播出。例如, 可以利用驻留在具有调制解调器或者其他类型的互联网连接设备的计算机上的程序, “通过” 或者使用因特网来传送图像, 以便接收或者发送能够用来以数字方式表现图像的数据。使用诸如配备有无线调制解调器的便携式计算机、无线个人数字助理 (PDA) 或者无线电话之类的无线设备也能够进行类似的传送。

[0006] 一旦收到数据, 就在本地将数据储存在存储器元件、电路或者设备中, 例如储存在 RAM 或者快闪存储器中, 还包括存储在内部或外部存储设备中, 例如, 小尺寸硬盘, 以便进行回放。根据数据量和图像分辨率, 回放可以较快开始或者长时间延迟后再呈现回放。也就是说, 在某些情况下, 对于不需要大量数据或者使用某种缓存的小图像或者低分辨率图像而言, 图像呈现允许某种程度的实时回放, 因此经过了较小延迟之后, 一些内容呈现出来, 而

更多内容则还在传送过程中。假设传送链路不发生任何中断,或者没有来自相对于正在使用的传送信道的其他系统或用户的干扰,那么一旦呈现开始,这一传送对于观看设备的最终用户而言就是适当(reasonably)透明的。自然,当多个用户共享一条通信路径(例如,一条有线互联网连接)时,传输可能会中断,或者,比预期的要慢。

[0007] 用于产生静止图像或者运动视频的数据经常使用几种众所周知的技术之一来进行压缩,诸如由联合图像专家组(JPEG)、运动图像专家组(MPEG)以及介质、计算机和通信行业中众所周知的其它标准组织或公司规范的那些技术,以加速通信链路上数据的传送。这些技术使用较小数量的比特来传送给定信息量,从而允许更快地传送图像或者数据。

[0008] 一旦把数据传送到诸如计算机或者其它接收器设备之类的“本地”设备,其中,计算机具有存储机制,如存储器,或者磁或光存储元件,就将得到的信息解压缩(或者使用特殊解码播放器播放),如果需要,则进行解码,并且基于相应的可用呈现分辨率和控制要素为适当的呈现做好准备。例如,用 $X \times Y$ 像素屏幕分辨率表示的典型的计算机视频分辨率通常从低到 $480 \times 640$ 像素到 $600 \times 800$ 再到 $1024 \times 1024$ ,当然根据需要或要求,一般还可能有各种其它分辨率。

[0009] 就某些预定色彩等级或者色彩浓度(用于生成颜色的每一像素的位)和强度以及所采用的任何额外开销位而言,图像呈现还受图像内容和给定视频控制器操纵图像的能力的影响。例如,典型的计算机呈现预期每像素大约8至32位或更多位,从而表现各种颜色(浓淡和色调),但也会遇到其它值。

[0010] 从上述值可以看出,在从最低到最高的典型分辨率和浓度的范围上,给定的屏幕图像将要求分别传送从2.45兆比特(Mb)到33.55Mb左右的数据。以30帧每秒的速率观看视频或者运动类型的图像时,所需要的数据量大约73.7至1,006兆比特数据每秒(Mbps),或者大约9.21至125.75兆字节每秒(MBps)。另外,人们还可能希望将音频数据和图像一起呈现,例如多媒体呈现,或者希望将音频数据作为独立的高分辨率音频呈现,例如CD质量的音乐。还可以采用处理交互式命令、控制或者信号的额外的信号。这些选项中每一项都增加了更多的待传送数据。此外,涉及高清晰度(HD)电视和电影录制的更新传输技术可能会添加更多的数据和控制信息。在任何情况下,当人们希望向最终用户传送高质量或者高分辨率图像数据和高质量音频信息或者数据信号以便产生内容丰富的体验时,在呈现元件和配置成提供这种数据的源或主机设备之间需要具有高数据传送速率的链路。

[0011] 通常,115千字节每秒(KBps)或者920千比特每秒(Kbps)左右的数据速率能够由现代串行接口来处理。如USB串行接口的其它接口能够支持高达12MBps速率的数据传送,而诸如使用电气与电子工程师协会(IEEE)1394标准配置的那种专用高速传送则能够以100至400MBps数量级的速率进行。但是,这些速率未能达到所期望的上述高数据速率,所期望的这些速率的目的是能够供将来的无线数据设备和服务使用,以便提供高分辨率、内容丰富的输出信号来驱动便携式视频显示器或者音频设备。这包括用于商务和其他呈现的计算机、游戏设备等。另外,这些接口要求使用数目巨大的主机或者系统以及客户机软件来工作。它们的软件协议栈还产生不期望的大量开销,特别是在考虑到移动无线设备或者电话应用的情况下更为突出。这种设备具有严格的存储器和功耗限制,以及已经负担过重的计算能力。此外,一些接口使用大体积的电缆或者利用复杂的连接器,对于高度要求美观的移动应用而言,这些大体积的电缆太过笨重,不能令人满意,而这些复杂的连接器又增加

了成本,或者是功耗实在太大。

[0012] 还有其它已知接口,例如模拟视频图形适配器 (VGA)、数字视频交互 (DVI) 或者千兆位视频接口 (GVIF)。这些接口中的前两个是并行类型的接口,用于以更高的传送速率来处理数据,但是它们也采用笨重的电缆并且消耗大量功率,大约有几瓦的量级。这些特性没有一个可供便携式消费电子设备使用。甚至第三个接口也消耗太多的功率并且使用昂贵的或者体积庞大的连接器。

[0013] 对于上述接口中的一些以及其它非常高速率数据系统 / 协议或者与固定安装的计算机设备的数据传送相关联的传送机制来说,存在另一个主要缺陷。为了支持所需要的数据传送速率,还需要相当大的功率和 / 或用大电流工作。这样做极大地降低了将这种技术用于面向消费者的高移动性产品的可用性。

[0014] 一般说来,为了支持这种数据传输速率而使用诸如光纤类型的连接以及传送元件之类的替代物,还需要多个额外的转换器和元件,与真正面向消费者的商业产品所需求的相比,这样做复杂度更大、成本更高。到目前为止,除了光学系统一般都非常昂贵的本性以外,它们对功率的要求以及它们的复杂性同样妨碍了它们在轻便、低功率、便携式应用方面的普遍应用。

[0015] 便携式、无线或者移动应用行业所缺乏的是这样一种技术,即:向高移动性的最终用户提供高质量呈现体验,而不管它是基于音频、视频还是多媒体的。也就是说,当使用便携式计算机、无线电话、PDA 或者其它高移动性通信设备或者装置时,当前正在使用的视频和音频呈现系统或设备根本无法以所期望的高质量水平来传递输出。感觉到质量的缺乏往往是无法获得传送高质量呈现数据所需要的高数据速率的结果。这包括向更加有效、先进或充满特性的外部设备传送以便向最终用户呈现,或者在主机和如计算机、游戏机这种便携式设备内部的客户机之间传送,以及在主机和包括诸如移动电话之类的无线设备之间传送。

[0016] 在后一种情况下,在添加越来越高分辨率的内部视频屏幕以及其它专业输入和 / 或输出设备,并且连接至诸如所谓的第三代电话的无线设备以及所谓的膝上型计算机方面,已经迈进了一大步。然而,内部数据总线和连接可以包括桥接旋转或者滑动铰链或者类似铰链的结构,这些结构将视频屏幕或者其它元件安装或者连接至安装了主机和 / 或各种其它控制元件和输出部件的主外壳。这些通常是高带宽或高吞吐量的接口。使用现有技术来构造高吞吐量数据传送接口是非常难的,例如在无线电话中,现有技术要求高达 90 个导线或者更多,来实现所期望的吞吐量。当前的解决方案通常要采用具有较高信号电平的并行类型接口,而这会使互联成本更高、更不可靠,并可能会产生干扰设备功能的有辐射发射。这样做在有关制造、成本限制和可靠性方面存在许多难题尚待解决。

[0017] 对于固定位置的装置,也会有这种问题和需求,其中,例如,通信或计算类设备添加在家用电器和其他消费设备中,以提供先进的数据能力、互联网和数据传输连接,或内置在娱乐中。另一个例子是飞机和汽车,其中,单独的视频和音频呈现屏安装在座椅靠背中。但是,在这些情形中,更便利、高效和易用的做法是,让主存储、处理或通信控制单元远离可视屏或音频输出端,用互联的链路或信道来呈现信息。这种链路将需要处理相当数量的数据以实现如上所述的期望吞吐量。

[0018] 因此,需要一种新的传送机制来提高提供数据的主机设备和向最终用户呈现输出

的客户机显示器设备或元件之间的数据吞吐量。

[0019] 在美国专利申请 10/020,520 (现为美国专利 6,760,772,于 2004 年 7 月 6 日授予 Zou 等) 和 2003 年 9 月 6 日提交的共同待决的美国专利申请 10/236,657 中,申请人已经提出了这种新的传送机制,这两篇申请标题均为“Generating and Implementing A Communication Protocol and Interface for High Data Rate Signal Transfer”,这两个已授权的案子现在已经转让给本发明的受让人,在这里将它们引入作为参考。此外,还有 2004 年 6 月 2 日提交的、序号为 10/860,116、标题为“Generating and Implementing a Signal Protocol and Interface for Higher Data Rates”的共同待决的美国申请。在这些申请中讨论的技术能够极大地提高高速度数据信号中大量数据的传送速率。然而,不断提高数据速率,特别是涉及视频呈现的速率的需求不断增长。即使是利用数据信号技术领域正在开发的其它进展,仍然需要为进一步加快传送速率,提高通信链路效率,并且使通信链路更加强大而继续努力。因此,仍然需要开发用来提高主机和客户机设备之间数据吞吐量的新的或者改进的传送机制。

### 发明内容

[0020] 通过本发明的实施例,所述领域现存的上述以及其它缺陷将得以解决,在本发明的实施例中,已经开发出了用于在主机设备和接收方客户机设备之间以高数据速率传送数据的新协议和数据传送手段、方法和机制。

[0021] 本发明的实施例致力于一种在主机设备和客户机设备之间经由通信路径以高速率传送数字数据的移动数据数字接口 (Mobile Data Digital Interface, MDDI),它采用多个或一系列链接起来的分组结构来形成通信协议,以便在主机和客户机设备之间传递数字控制和呈现数据的预选集合。该信号通信协议或者链路层由主机或者客户机链路控制器、接收器或驱动器的物理层使用。驻留在主机设备中的至少一个链路控制器或驱动器经由通信路径或者链路耦合至客户机设备,并且被配置成生成、发射并且接收形成该通信协议的分组,并且将数字呈现数据形成为一个或多个类型的数据分组。所述接口在主机和客户机之间提供信息的双向传送,所述主机和客户机能够存在于公共的整体外壳或支撑结构内。

[0022] 除了差分驱动器和接收器之外,所述实现方式在本质上通常全部是数字化的,其中所述驱动器和接收器很容易就能够在数字 CMOS 芯片上实现,它们需要少数几个信号,如 6 个信号,并且几乎能够以对于系统设计员来说极为便利的任何数据速率工作。这种简单的物理和链路层协议使其便于集成,这种简单性加上休眠状态使得这种便携式系统具有非常低的系统功耗。

[0023] 为了有助于使用和接受,所述接口将增加极少的设备成本,能够在使用标准电池电压经由所述接口供电显示的同时,允许消耗极少的功率,并且能够使设备具有能装入口袋的外形因素。所述接口是可升级的,以便支持 HDTV 以上的分辨率,该接口还支持显示器设备的同步立体视频和 7.1 音频,对任何屏幕区域执行有条件的更新并且双向支持多种数据类型。

[0024] 按照本发明的实施例的其他方面,至少一个客户机链路控制器、接收器、设备或者驱动器被设置在客户机设备中,并且经由通信路径或者链路耦合至所述主机设备。所述客户机链路控制器也被配置成生成、传输并且接收形成通信协议的分组,并且将数字呈现数

据形成一个或多个类型的数据分组。一般说来,主机或者链路控制器采用状态机来处理用在命令或者某些类型的信号预备和查询处理中的数据分组,但是其也能够使用更慢的通用处理器来操纵数据和用于所述通信协议的某些不那么复杂的分组。所述主机控制器包括一个或多个差分线路驱动器;而客户机接收器包括一个或多个耦合至所述通信路径的差分线路接收器。

[0025] 所述分组在主机和客户机设备之间通信的介质帧内被组合在一起,这些介质帧具有预定义固定长度,其中预定数目的分组具有不同的可变长度。所述分组均包括分组长度字段、一个或多个分组数据字段和循环冗余校验字段。子帧报头分组被传送或者定位在来自主机链路控制器的其它分组传送的开头。所述通信协议使用一个或多个视频流类型分组和音频流类型分组,以便从主机经由正向链路向客户机分别传送视频类型数据和音频类型数据,从而向客户机设备用户呈现。所述通信协议使用一个或多个反向链路封装类型分组,以便把数据从客户机设备传送到主机链路控制器。在一些实施例中,这些传送包括把数据从具有至少一个 MDDI 设备的内部控制器传送至内部视频屏幕。其它实施例包括向内部音响系统传送,并且从包括游戏操纵杆和复杂键盘的各种输入设备向内部主机设备传送。

[0026] 由所述主机链路控制器生成填充符类型分组,以便占据没有数据的正向链路传输的周期。所述通信协议使用多个其它分组,以便传送视频信息。这些分组包括色彩映射、位块传送、位图区域填充、位图图案填充和透明色彩启用类型分组。所述通信协议使用用户定义的流类型分组以便传送接口-用户定义的数据。所述通信协议使用键盘数据和指示设备数据类型分组以便向与所述客户机设备相关联的用户输入设备传送数据或从这些设备中传送出来。所述通信协议使用链路关闭类型分组来终止沿任何一个方向经由所述通信路径的数据传送。

[0027] 所述通信路径通常包括或者采用具有一系列四个或更多导线和一个屏蔽的电缆。另外,根据需要,可以使用印刷线路或者导线,其中有一些线路或者导线驻留在柔性衬底上。

[0028] 所述主机链路控制器从客户机设备请求显示器能力信息,以便确定所述客户机经由所述接口能够支持什么类型的数据和数据速率。所述客户机链路控制器使用至少一个显示器能力类型分组向主机链路控制器传递显示或者呈现能力。通信协议使用多个传送模式,每个传送模式均允许在给定时间段内并行传送不同的最大数目数据位,每一模式均可以通过主机和客户机链路控制器之间的协商来选择。这些传送模式可以在数据传送期间动态地调整,并且在反向链路上不必使用与正向链路上所使用的传送模式那样使用相同的模式。

[0029] 在本发明某些实施例的其它方面中,所述主机设备包括无线通信设备,诸如无线电话、无线 PDA 或者具有其中设置有无线调制解调器的便携式计算机。典型的客户机设备包括便携式视频显示器,诸如微型显示器设备,和/或便携式音频呈现系统。此外,主机可以使用储存模块或元件以储存待传送以便呈现给客户机设备用户的呈现或者多媒体数据。

[0030] 在某些实施例的其他方面中,所述主机设备包括具有下述的存在于便携式电子设备内的驱动器的控制器或者通信链路控制设备,所述便携式电子设备诸如无线通信设备,诸如无线电话、无线 PDA 或者便携式计算机。这一配置的典型客户机设备包括客户机电路、集成电路或者模块,其耦合至主机并且与之驻留在同一设备中,并且耦合至如移动电话和/

或便携式音频呈现系统,或者一些可选类型的输入系统或设备中的高分辨率屏幕之类的内部视频显示器。

#### 附图说明

[0031] 下面对照附图详细说明本发明的其它特征和优点,以及本发明各种实施例的结构和操作。在这些附图中,相同的标号一般表示相同,功能上类似和/或结构上类似的元件或处理步骤,并且元件首次出现的附图用标号中最左边的数字来表示。

[0032] 图 1A 示出了本发明的实施例能够工作的基本环境,包括与便携式计算机或其他数据处理设备结合使用的微型显示器设备或投影仪的使用。

[0033] 图 1B 示出了本发明的实施例能够工作的基本环境,包括与无线收发器结合使用的微型显示器设备或投影仪和音频呈现元件的使用。

[0034] 图 1C 示出了本发明的实施例能够工作的基本环境,包括用在与便携式计算机中的内部显示器设备或音频呈现设备的使用。

[0035] 图 1D 示出了了本发明的实施例能够工作的基本环境,包括在无线收发器中使用内部显示器或音频呈现元件的使用。

[0036] 图 2 示出了具有主机和客户机互连的移动数字数据接口的总体构思。

[0037] 图 3 示出了用于实现从客户机设备到主机设备的数据传送的分组结构。

[0038] 图 4 示出了 MDDI 链路控制器的使用,以及在主机和客户机之间经由类型 1 接口的物理数据链路导线传递的信号类型。

[0039] 图 5 示出了 MDDI 链路控制器的使用,以及在主机和客户机之间经由类型 2、3、4 型接口的物理数据链路导线传递的信号类型。

[0040] 图 6 示出了用于实现所述接口协议的帧和子帧的结构。

[0041] 图 7 示出了用于实现所述接口协议的分组的通用结构。

[0042] 图 8 示出了子帧报头分组的格式。

[0043] 图 9 示出了填充符分组的格式和内容。

[0044] 图 10 示出了视频流分组的格式。

[0045] 图 11A-11E 示出了图 10 中使用的视频数据格式描述符的格式和内容。

[0046] 图 12 示出了数据的打包的和未打包的格式的使用。

[0047] 图 13 示出了音频流分组的格式。

[0048] 图 14 示出了数据的按字节对准和打包的 PCM 格式的使用。

[0049] 图 15 示出了用户定义的流分组的格式。

[0050] 图 16 示出了色彩映射分组的格式。

[0051] 图 17 示出了反向链路封装分组的格式。

[0052] 图 18 示出了客户机能力分组的格式。

[0053] 图 19 示出了键盘数据分组的格式。

[0054] 图 20 示出了指示设备数据分组的格式。

[0055] 图 21 示出了链路关闭分组的格式。

[0056] 图 22 示出了客户机请求和状态分组的格式。

[0057] 图 23 示出了位块传送分组的格式。

- [0058] 图 24 示出了位图区域充分组的格式。
- [0059] 图 25 示出了位图图案充分组的格式。
- [0060] 图 26 示出了通信链路数据通道分组的格式。
- [0061] 图 27 示出了接口类型切换请求分组的格式。
- [0062] 图 28 示出了接口类型确认分组的格式。
- [0063] 图 29 示出了执行类型切换分组的格式。
- [0064] 图 30 示出了正向音频通道启用分组的格式。
- [0065] 图 31 示出了反向音频采样速率分组的格式。
- [0066] 图 32 示出了数字内容保护开销分组的格式。
- [0067] 图 33 示出了透明色彩启用分组的格式。
- [0068] 图 34 示出了往返行程延迟测量分组的格式。
- [0069] 图 35 示出了往返行程延迟测量分组期间事件的时序。
- [0070] 图 36 示出了用于实现本发明的 CRC 生成器和校验器的示例性实现方式。
- [0071] 图 37A 示出了图 36 所示装置发送数据分组时 CRC 信号的时序。
- [0072] 图 37B 示出了图 36 所示装置接收数据分组时 CRC 信号的时序。
- [0073] 图 38 示出了没有竞争情况下典型服务请求的处理步骤。
- [0074] 图 39 示出了在链路重启序列开始之后维持 (assert) 的典型服务请求与链路启动进行竞争的处理步骤。
- [0075] 图 40 示出了如何使用 DATA-STB 编码来传输数据序列。
- [0076] 图 41 示出了可用于在主机中根据输入数据生成 DATA 和 STB 信号,然后在客户机中恢复数据的电路。
- [0077] 图 42 示出了可用于实现一个实施例的驱动器和终端电阻。
- [0078] 图 43 示出了由客户机用来保证得到来自主机的服务以及由主机用来提供这种服务的步骤和信号电平。
- [0079] 图 44 示出了 Data0、其它数据线 (DataX) 和选通线 (Stb) 上的跃迁 (transition) 之间的相对间隔。
- [0080] 图 45 示出了当主机传送分组之后禁用主机驱动器时会出现的,响应中存在的延迟。
- [0081] 图 46 示出了当主机启用主机驱动器传送分组时会出现的,响应中存在的延迟。
- [0082] 图 47 示出了漏电流分析。
- [0083] 图 48 示出了主机和客户机输出禁用和启用时间的切换特性和相对时序关系。
- [0084] 图 49 示出了能够使用状态机实现同步的信号处理步骤和条件的高级图表。
- [0085] 图 50 示出了在采用 MDDI 的系统中正向路径和反向路径上信号处理遇到的典型延迟量。
- [0086] 图 51 示出了临界的往返行程延迟测量。
- [0087] 图 52A 示出了反向链路数据速率的变化。
- [0088] 图 52B 示出了先进的反向数据采样的一个例子。
- [0089] 图 53 绘出了反向速率除数的值随正向链路数据速率变化的图形表示。
- [0090] 图 54A 和 54B 示出了接口操作中进行的步骤。

- [0091] 图 55 示出了处理分组的接口装置的概况。
- [0092] 图 56 示出了正向链路分组的格式。
- [0093] 图 57 示出了类型 1 链路接口中的传播延迟和偏离 (skew) 的典型值。
- [0094] 图 58 示出了对于经由所述接口的示例性信号处理而言, 在类型 1 链路上的 Data(数据)、Stb(选通) 和 Clock(时钟) 恢复时序。
- [0095] 图 59 示出了类型 2、3 或 4 链路接口中传播延迟和偏离的典型值。
- [0096] 图 60A、60B 和 60C 示出了两个数据信号和 MDDI\_Stb 之间相对时序的不同可能性, 它们分别对应于理想情况、提早情况和推迟情况。
- [0097] 图 61 示出了类型 1/ 类型 2 接口所用示例性连接器的接口管脚分配。
- [0098] 图 62A 和 62B 分别示出了类型 1 和类型 2 接口可能的 MDDI\_Data 和 MDDI\_Stb 波形。
- [0099] 图 63 示出了能够使用状态机实现同步的可选的信号处理步骤和条件的高级图表。
- [0100] 图 64 示出了一系列时钟周期与各种反向链路分组位的时序以及除数值之间的相对时序。
- [0101] 图 65 示出了示例性的错误代码传送处理。
- [0102] 图 66 示出了可用于错误代码传送处理的装置。
- [0103] 图 67A 示出了用于代码重载的错误代码传送处理。
- [0104] 图 67B 示出了用于代码接收的错误代码传送处理。
- [0105] 图 68A 示出了主机启动的唤醒的处理步骤。
- [0106] 图 68B 示出了客户机启动的唤醒的处理步骤。
- [0107] 图 68C 示出了带有竞争的主机和客户机启动的唤醒的处理步骤。
- [0108] 图 69 示出了请求 VCP 特征分组的格式。
- [0109] 图 70 示出了 VCP 特征应答分组的格式。
- [0110] 图 71 示出了 VCP 特征应答列表的格式。
- [0111] 图 72 示出了设置 VCP 特征分组的格式
- [0112] 图 73 示出了请求有效参数分组的格式。
- [0113] 图 74 示出了有效参数应答分组的格式。
- [0114] 图 75 示出了可缩放视频流能力分组的格式。
- [0115] 图 76 示出了可缩放视频流建立分组的格式。
- [0116] 图 77 示出了可缩放视频流确认分组的格式。
- [0117] 图 78 示出了可缩放视频流分组的格式。
- [0118] 图 79 示出了请求特定状态分组的格式。
- [0119] 图 80 示出了有效状态应答列表分组的格式。
- [0120] 图 81A 示出了分组处理延迟参数分组的格式。
- [0121] 图 81B 示出了延迟参数列表项的格式。
- [0122] 图 82 示出了个人显示器能力分组的格式。
- [0123] 图 83 示出了视场曲率点列表字段的点。
- [0124] 图 84A 示出了客户机错误报告分组的格式。



- [0125] 图 84B 示出了错误报告列表项的格式。
- [0126] 图 85 示出了客户机标识分组的格式。
- [0127] 图 86 示出了可选显示器能力分组的格式。
- [0128] 图 87 示出了寄存器访问分组的格式。
- [0129] 图 88A-88C 示出了使用两个显示器缓冲器来减少可见图像 (visible artifacts)。
- [0130] 图 89 示出了显示刷新快于图像传送的两个缓冲器。
- [0131] 图 90 示出了显示刷新慢于图像传送的两个缓冲器。
- [0132] 图 91 示出了显示刷新比图像传送快得多的两个缓冲器。
- [0133] 图 92 示出了显示刷新快于图像传送的三个缓冲器。
- [0134] 图 93 示出了显示刷新慢于图像传送的三个缓冲器。
- [0135] 图 94 示出了显示刷新快于图像传送的一个缓冲器。
- [0136] 图 95 示出了经由串级链 (daisy-chain) 和集线器的主机 - 客户机连接。
- [0137] 图 96 示出了经由集线器和串级链的组合连接的客户机设备。
- [0138] 图 97 示出了色彩映射。

## 具体实施方式

### [0139] I. 概述

[0140] 本发明总的目的在于提供一种移动显示数字接口 (MDDI, Mobile Display Digital Interface), 如下所述, 它能够得到或者提供具有性价比高、低功耗的传送机制, 这种传送机制使用“串行”类型的数据链路或者通道, 能够在主机设备和如显示器元件的这种客户机设备之间的近距离通信链路上实现高速或者非常高速的数据传送。这一机制适于利用小型连接器和细软电缆来实现, 这些小型连接器和细软电缆特别适合于把内部 (在外壳或者支架内) 显示器或者输出元件或设备或输入设备连接至中央控制器、通信元件或设备。此外, 把如可佩带的微型显示器 (护目镜或者投影仪) 之类的外部显示器元件或设备或其他类型的视觉、听觉、触觉信息呈现设备连接至便携式计算机、无线通信设备或者娱乐设备时, 这种连接机制非常有用。

[0141] 尽管术语移动和显示与所述协议的命名相关联, 但是应该理解, 这仅仅是为了便于让本领域中研究接口和协议的技术人员容易地理解标准的名称。它将涉及 VESA 标准以及该标准的各种应用。然而, 在阅读以下给出的实施例后, 很容易理解的是, 许多与非移动性和非显示相关的应用也会得益于应用本协议、所得到的接口结构或传送机制, 而 MDDI 标记并不意味着对本发明或其各种实施例施加任何限制。

[0142] 本发明的实施例的一个优点是数据传送提供一种技术, 它复杂度低、成本低、可靠性高, 非常适合于使用环境, 并且非常稳定, 同时保持了非常高的灵活性。

[0143] 可以将本发明的实施例用于各种情况, 以便把通常用于音频、视频或者多媒体应用的大量数据从生成、控制 (如, 传送到特定设备) 或处理或储存这种数据的主机或者源设备以高速率传递或者传送到客户机或者接收设备, 如视频显示器或投影元件、音频扬声器或其他呈现设备。下面讨论的一个典型应用是从便携式计算机、无线电话或调制解调器向可视显示器设备的数据传送, 其中可视显示器设备例如为小型视频屏幕或可佩带的微型显

示器用品 (micro-display appliance), 比如护目镜形式或者包含小型投影透镜和屏幕的头盔形式, 或者在这种部件内从主机向客户机设备的数据传送。也就是说, 从处理器到内部屏幕或其它呈现元件, 并且从各种内部输入设备或者采用客户机的外部输入设备到安装在内部的 (一起置在同一个设备外壳或者支撑结构内) 主机, 或通过线缆或导线连接到那里。

[0144] MDDI 的特性或属性不依赖于具体的显示或呈现技术。无论数据的内部结构, 还是数据或其执行的命令的功能方面如何, MDDI 都是用于以高速率传送数据的高灵活性机制。它允许调节正在传送的分组的时序以适应特定客户机设备的特性, 例如针对某些设备的独特显示需求的特性, 或者为满足某些 A-V 系统组合音频和视频的要求的特性, 或者针对如游戏操纵杆、触摸盘等等的某些输入设备的特性。这种接口不需要了解采用的是什么显示器元件或者客户机设备, 只要它们遵循所选的协议即可。另外, 总的串行链路数据或数据速率能够在几个数量级上改变, 这就使得通信系统或者主机设备的设计者能够对成本、功率要求、客户机设备的复杂程度以及客户机设备的更新速率进行优化。

[0145] 这种数据接口主要用于经由“有线”信号链路或者小型电缆传送大量高速率数据。然而, 某些应用也可以利用无线链路, 包括基于光的链路, 只要将这些链路配置成使用为这一接口协议开发的相同分组和数据结构, 并且能够以足够低的功耗或者复杂性来实现所期望的传送水平, 以便保持实用性。

[0146] II. 环境

[0147] 在图 1A 和 1B 中可以看到典型的应用, 其中示出了分别与显示器设备 104 和 106 以及音频再现系统 108 和 112 传递数据的便携式或者膝上型计算机 100 和无线设备或者 PDA 设备 102。另外, 图 1A 示出了与较大的显示器或者屏幕 114 或者图像投影仪 116 的潜在连接, 为了清楚, 仅仅在一个图中示出, 但是其也可以与无线设备 102 相连。无线设备有可能目前正在接收数据或者已经在存储元件或设备中预先储存了一定量的多媒体类型数据, 以便稍候呈现给无线设备的最终用户来观看和 / 或倾听。由于大多数时间使用典型的无线设备来进行语音和简单文本的通信, 所以它具有相当小的显示屏和简单的音频系统 (扬声器) 来向设备 102 的用户传递信息。

[0148] 计算机 100 具有非常大的屏幕, 但是还缺少外部音响系统, 并且仍达不到诸如高清晰度电视或者电影屏幕之类的其它多媒体呈现设备的水平。使用计算机 100 是出于说明的目的, 本发明也可以使用其它类型的处理器、交互式视频游戏或者消费电子设备。计算机 100 能够采用无线调制解调器或者其它嵌入设备来进行无线通信, 或者根据需要使用电缆或者无线链路与这种设备相连, 但是不局限于此。

[0149] 这样做不足以使得更加复杂或者“丰富”的数据的呈现提供有用或令人愉快的体验。因此, 该行业正在开发其它机制和设备来向最终用户呈现信息并且提供所期望的享受或积极体验的最低水平。

[0150] 正如先前所述的那样, 已经开发或者目前正在开发几种类型的显示器设备来向设备 100 的最终用户呈现信息。例如, 一个或多个公司已经开发了多组可佩带的护目镜, 用于在设备用户的眼睛前方投影图像以便呈现可视显示。正确定位的时候, 这种设备可以有效地“投射”一个虚拟图像, 就像用户眼睛感觉到的那样, 这个图像比提供视觉输出的元件大得多。也就是说, 非常小的投影元件使得用户的眼睛“看到”的图像比利用典型 LCD 屏幕等等可能看到的图像大得多。使用较大的虚拟屏幕图像还能够使用远高于利用更加有限的

LCD 屏幕显示器所能获得的分辨率的图像。其它显示器设备可以包括但不局限于小型 LCD 屏幕或者各种平板显示器元件、用于将图像投射在表面上的投影透镜和显示驱动器等等。

[0151] 还存在连接至无线设备 102 或计算机 100 或者与它们的使用相关联的附加元件, 以便向其它用户或者向其它设备呈现输出, 其中所述其它设备还把信号传送到其它地方或者储存信号。例如, 可以把数据储存在快闪存储器中, 以光学形式, 例如使用可写 CD 介质, 储存或者储存在诸如磁带记录器的磁介质上或类似设备中, 供以后使用。

[0152] 另外, 许多无线设备和计算机目前具有内置 MP3 音乐解码能力, 以及其它先进的声音解码器和系统。通常, 便携式计算机利用 CD 和 DVD 播放能力, 并且有一些还具有小型专用快闪存储器读取器来接收预先记录的音频文件。具有这种能力的问题在于: 数字音乐文件许诺能够提供高度增加的特性丰富的体验, 但是只有当解码和播放处理能够跟上时才可实现。这对于数字视频文件也是一样。

[0153] 为了帮助声音再现, 图 1A 中示出了外部扬声器 114, 其还附带有诸如超低音扬声器或者用于前后方声音投射的“环绕声”扬声器之类的附加元件。同时, 将扬声器或者耳机 108 示出为嵌入支架或者图 1B 的微型显示器设备 106 的机制的形式示出。众所周知, 还可以使用包括功率放大或者声音成形设备的其它音频或者声音再现元件。

[0154] 在任何情况下, 如上所述, 当人们希望经由一个或多个通信链路 110 从数据源向最终用户传送高质量或者高分辨率图像数据和高质量音频信息或者数据信号时, 要求高数据速率。也就是说, 传送链路 110 显然是数据通信过程中潜在的瓶颈, 就象早先讨论的那样, 并且由于当前的传送机制无法实现通常期望的高数据速率, 所以限制了系统性能。如上所述, 例如对于  $1024 \times 1024$  像素的较高图像分辨率、每像素 24 ~ 32 位的色彩浓度并且处于 30fps 的数据速率来说, 数据速率能够接近超过 755Mbps 的速率或者更高。另外, 这种图像可以作为多媒体呈现的一部分来呈现, 该多媒体呈现包括音频数据以及处理交互式游戏或通信、或者各种命令、控制或信号的潜在附加信号, 这进一步增加了数据量和数据速率。

[0155] 还显而易见的是, 建立数据链路所需的电缆或者互连越少, 意味着与显示器相关联的移动设备越易于使用, 并且越可能被更多的用户采用。在通常使用多个设备来建立完全的音频 - 视频体验的情况下, 尤为如此, 并且随着显示器和音频输出设备质量水平的提高而更为突出。

[0156] 涉及视频屏幕以及其它输出或者输入设备方面的上述以及其它改善的其它典型应用可以从图 1C 和 1D 中看到, 其中示出了分别与“内部”显示器设备 134 和 144 以及音频再现系统 136 和 146 传递数据的便携式或者膝上型计算机 130 以及无线电话或者 PDA 设备 140。

[0157] 在图 1C 和 1D 中, 用整个电子设备或者产品的小切开 (cut-away) 部分来示出设备的一部分中的一个或多个内部主机和控制器的位置, 通过整个电子行业目前使用的某些已知类型的转动铰链, 通用通信链路 (此处分别为 138 和 148) 把这些内部主机和控制器连接到具有相应客户机的视频显示器元件或者屏幕。人们可以看出, 这些传送所涉及的数据量要求大量导线来构成链路 138 和 148。由于并行接口或者其它已知接口技术的类型可用于传送这种数据, 因此据估计, 这种通信链路具有接近于 90 个或更多的导线, 以便满足现在对利用这种设备上的先进的彩色和图形接口、显示器元件的不断增长的需要。

[0158] 但是, 这种更高数据速率超出了当前用于传送数据的可用现有技术。就每单位时

间需要传送的原始数据量而言,并且就制造可靠的低成本的实用传送机制而言,均是如此。

[0159] 所需要的是这样一种用来在呈现元件和数据源之间的数据传送链路或者通信路径上以高速率传送数据的技术、结构、装置或方法,从而实现一贯低功耗、重量轻并且尽可能简单和经济的电缆结构。本申请人已经开发了一种新的技术、方法和装置来实现这些以及其它目的,以便使一系列移动式、便携式乃至固定位置的设备能够以非常高的数据速率向所期望的显示器、微型显示器或者音频传送元件传送数据,同时维持所期望的低功耗和低复杂性。

[0160] III. 高速率数字数据接口系统体系结构

[0161] 为了创建并且高效地利用新的设备接口,已经配制了使用低功率信号来提供很高的数据传送速率的信号协议和系统体系结构。所述协议基于分组和公共帧结构或者链接起来以便形成协议的结构,以用于传递预选数据集或者数据类型以及施加在接口上的命令或者操作结构。

[0162] A. 概述

[0163] 把经由 MDDI 链路连接或者通信的设备称为主机和客户机,所述客户机通常是某些类型的显示器设备,但是也可以考虑其它输出和输入设备。在由主机启用时,从主机到显示器的数据沿正向传送(称为正向业务或者链路),从客户机到主机的数据沿反向传送(称为反向业务或者链路)。在图 2 中示出的基本结构中说明这些。在图 2 中,主机 202 使用双向通信通道 206 与客户机 204 相连,所述双向通信通道是按照包括正向链路 208 和反向链路 210 的形式示出的。然而,这些通道是由公共导线集合形成的,其中这些公共导线集合的数据传送可在正向和反向链路操作之间有效切换。这样能够极大地减少导线数量,立即解决了在诸如移动式电子设备的低功耗环境下进行高速数据传送的当前方法所面对的许多问题之一。

[0164] 如同其它地方讨论的那样,所述主机包括能够得益于使用本发明的多种类型的设备之一。例如,主机 202 可以是具有手持、膝上型或者类似移动计算设备的形式的小型便携式计算机,它也可以是个人数字助理(PDA)、寻呼设备或者多种无线电话或者调制解调器之一。作为选择,主机 202 可以是便携式娱乐或者呈现设备,如便携式 DVD 或者 CD 播放器,或者是游戏设备。

[0165] 此外,所述主机可以作为主机设备或者控制元件而存在于各种其它广泛使用或者规划的商业产品中,这些产品与客户机之间需要建立高速通信链路。例如,主机可用于以高速率从视频记录设备向基于存储的客户机传送数据以改善响应,或者向高分辨率的大屏幕传送数据以便呈现。结合有板上清点(onboard inventory)或计算系统和/或与其它家庭设备的蓝牙连接的家电,如冰箱,当在互联网或者蓝牙连接的模式下工作时,具有改善了显示器能力,或者在电子计算机或者控制系统(主机)存在于室内其它地方的时候,降低了室内显示器(客户机)和小键盘或者扫描仪(客户机)的连线需要。总体上讲,本领域技术人员会了解,各种各样的现代电子设备和家电将得益于使用这种接口,而且通过利用新增加的或现有的连接器或者电缆中可用的有限数量的导线来实现信息的更高数据速率传输能够翻新旧式设备。

[0166] 同时,客户机 204 可以包括用于向最终用户呈现信息或者从用户向主机呈现信息的各种设备。例如,并入护目镜或者眼镜的微型显示器、嵌入帽子或者头盔的投影设备、嵌

入诸如车辆的窗户或者挡风玻璃中的小型屏幕乃至全息元件、或者用于呈现高质量声音或者音乐的各种扬声器、头戴耳机或者音响系统。其他呈现设备包括用于呈现会议信息或电影和电视图像的投影仪或投影设备。另一实例是使用触摸板或者敏感设备、语音识别输入设备、安全扫描仪,以及其他能够被调用以从设备或系统用户处传递大量信息,其中该设备或系统用户具有来自用户的除触摸和声音以外的不太实际的“输入”。此外,计算机和车辆配件或台式机配件的扩展坞(docking station)和无线电话的持有者也可以作为对于最终用户或其他设备和装置的接口设备,并且可以利用客户机(如鼠标的输出或输入设备)或主机来帮助传送数据,特别是在涉及高速网络的情况下。

[0167] 然而,本领域技术人员会很容易认识到,本发明不局限于这些设备,市场上还有许多其它设备供使用,这些设备或者以储存和传送的方式,或者以播放时呈现的方式,向最终用户提供高质量图像和声音。本发明在各种元件或者设备之间提高数据吞吐量方面十分有用,从而能够适应用于实现所期望的用户体验所需的高数据速率。

[0168] 可以将本发明的MDD接口和通信信号协议用来简化设备内(内部模式)的主机处理器、控制器或电路组件(例如)和设备或设备外壳或结构中的显示器之间的互联(称为内部模式),以降低这些连接的成本或复杂度以及相关功率和控制要求或这些连接的约束条件,并且提高可靠性,而不仅仅连接到或用于外部元件、设备或装置(外部模式)。

[0169] 由此接口结构使用的每一信号对上总的串行链路数据速率可以在多个数量级上改变,这一点允许系统或设备设计者易于对成本、功率、实现方式的复杂性和显示器更新速率进行优化。MDDI的属性不依赖显示器或其他呈现设备(目标客户机)的技术。可以容易地调节经由接口传送的数据分组的时序,以适应特定客户机的特性,该客户机如显示器设备、音响系统、存储器和控制元件,或者适应音频-视频系统的组合时序要求的特性。虽然这样做允许系统消耗尽可能小的功率,但是它并不要求各客户机具有帧缓冲器以便至少在一级别使用MDDI协议。

#### [0170] B. 接口类型

[0171] 所述MDD接口被构想为能够处理在通信和计算机行业中可以找到的多少有些不同的至少四个或者更多的接口的物理类型。将这些类型的接口简单标记为类型1、类型2、类型3类型4,当然本领域技术人员也可以根据所针对的具体应用或与之相关的行业而应用其它标记或者名称。例如,简单音频系统使用少于复杂多媒体系统的连接,并且可以不同地引用如“通道”的特性等等。

[0172] 所述类型1接口被配置成6线(6-wire)或其他类型导线或传导元件的接口,这种接口使其适用于移动式或者无线电话、PDA、电子游戏和诸如CD播放器或者MP3播放器的便携式媒体播放器,以及类似设备或在类似类型的电子消费技术上所使用的设备。在一个实施例中,一个被配置成8线(导线)接口的接口可更加适用于膝上型计算机、笔记本或者台式个人计算机以及类似设备或应用,这些设备不要求快速的数据更新,也没有嵌入式MDDI链路控制器。这种接口类型还能够通过使用额外的双线通用串行总线(USB)接口来区别此接口类型,其中的USB非常适用于支持在大多数个人计算机上常见的现存操作系统或者软件支持。

[0173] 类型2、类型3和类型4接口适用于高性能客户机或设备,并且使用带有额外的双绞线类型导线的更大的复杂电缆来为数据信号提供适当的屏蔽和低损耗传送。

[0174] 类型 1 接口传递包括显示、音频、控制和有限信令信息的信号,并且通常用于移动式客户机或者不要求高分辨率全速率视频数据的客户机设备。在 30fps 外加 5.1 声道音频的情况下,类型 1 接口可以容易地支持 SVGA 分辨率,并且在最小配置中,总计只使用三个线对,两对用于数据传输,一对用于功率传送。这种类型的接口主要用于如移动无线设备的设备,这种设备内通常没有 USB 主机端进行连接和传送信号。在这种配置中,所述移动无线设备是 MDDI 主机设备,并且充当用于控制来自所述主机的通信链路的“主控设备”,其中所述主机通常向客户机发送数据(正向业务或者链路)以用于呈现、显示或者播放。

[0175] 在这种接口中,通过向客户机发送特殊的命令或者分组类型,主机能够在该主机处接收来自客户机的通信数据(反向业务或者链路),从而允许客户机在指定的持续时间段内占据总线(链路),并且把数据作为反向分组发送到主机。这些内容在图 3 中示出,其中称为封装分组的分组类型(将在下面讨论)用来支持该传送链路上的反向分组传送,以创建所述反向链路。分配给主机用来轮询客户机的数据的时间间隔由主机预先确定,并且该时间间隔是基于每一指定应用的要求。在没有 USB 端口用来传送来自客户机的信息或者数据时,这种类型的半双工双向数据传送是特别有益的。

[0176] 能够支持 HDTV 类型或者类似高分辨率的高性能显示器要求 1.5Gbps 左右速率的数据流以支持全运动视频。所述类型 2 接口通过并行传输 2 位来支持高数据速率,所述类型 3 通过并行传输 4 位来支持,而类型 4 接口并行传送 8 位。类型 2 和类型 3 使用与类型 1 相同的电缆和连接器,但是能够工作在两倍和四倍的数据速率上以便支持便携式设备上的更高性能的视频应用。类型 4 接口适用于极高性能的客户机或者显示器,并且需要包含附加双绞线数据信号的稍大电缆。

[0177] 通常,通过协商议定所能使用的最高数据速率,由 MDDI 使用的协议允许类型 1、2、3 和 4 的主机中的每一个与类型 1、2、3 和 4 的客户机中的任意一个进行通信。将可以称为最小能力设备的能力或者可用特征被用来设置链路的性能。通常,即使是主机和客户机均能够是使用类型 2、类型 3 或者类型 4 接口的系统,二者也都以类型 1 接口开始工作。然后,主机确定目标客户机的能力,并且协商议定切换到或重新配置操作到类型 2、类型 3 或类型 4 中的一个模式,只要其对于特定的应用合适即可。

[0178] 对于主机而言,通常能够使用正确的链路层协议(下文进一步讨论)并且通常在任何时候都逐级下降或者再次重新配置操作到较慢模式以便节省功耗,或者逐级上升到较快模式来支持如高分辨率显示内容的更高速度传送。例如,当系统从如电池的电源切换至交流电源时,或者当显示媒体源切换到更低或者更高分辨率格式时,主机可以改变接口类型,或者可以将这些或者其它条件或者事件的组合认为是改变接口类型或者传送模式的基础。

[0179] 系统还能够在一个方向上使用一种模式而在另一个方向上使用另一模式来传递数据。例如,可以将类型 4 接口模式用于以高速率传送数据给显示器,而从如键盘或者指示设备这种外围设备向主机设备传送数据时则使用类型 1 模式。本领域普通技术人员会明白,主机和客户机能够以不同速率传递输出数据。

[0180] MDDI 协议的用户常常可以将“外部”模式和“内部”模式区分开来。外部模式描述的是利用该协议和接口把一个设备中的主机连接到该设备外部的且距离该设备最多 2 米左右的客户机。在这种情况下,主机还可以向外部客户机送电,以便使两个设备都易于在移

动环境下工作。内部模式描述的是主机与同一设备内部包含的客户机相连,例如处于公共外壳或者支架或者某种结构内。一个实例可以是无线电话或者其它无线设备内、或者便携式计算机或者游戏设备内的应用,其中客户机是显示器或者显示驱动器,或如键盘或触摸板的输入设备,或音响系统,而主机是中央控制器、图形引擎或者 CPU 元件。与外部模式应用相反,由于在内部模式应用中客户机位于非常靠近主机的地方,所以在这种结构中通常不要求将电源连接至客户机。

#### [0181] C. 物理接口结构

[0182] 图 4 和 5 中示出了用于在主机和客户机设备之间建立通信的设备或者链路控制器的一般配置。在图 4 和 5 中,所示出的 MDDI 链路控制器 402 和 502 安装在主机设备 202 中,并且所示出的 MDDI 链路控制器 404 和 504 安装在客户机设备 204 中。如前所述,主机 202 使用包括一系列导线的双向通信通道 406 与客户机 204 相连。如下所述,主机和客户机链路控制器两者都可以使用单一电路设计制作为集成电路,所述设计可以被设置、调节或者编程以作为主机控制器(驱动器)或者客户机控制器(接收器)来进行响应。由于需要更大规模地制造单个电路设备,这样做的成本更低。

[0183] 在图 5 中,所示出的 MDDI 链路控制器 502 安装在主机设备 202' 中,并且所示出的 MDDI 链路控制器 504 安装在客户机设备 204' 中。如前所述,主机 202' 使用包括一系列导线的双向通信通道 506 与客户机 204' 相连。如上所述,主机和客户机链路控制器两者都可以使用单个电路设计来加以制造。

[0184] 在图 4 和 5 中还说明在主机和如显示器设备之类的客户机之间经由 MDDI 链路或者所使用的物理导线来传递的信号。如图 4 和 5 所示,用于经由 MDDI 传送数据的主要路径或者机制使用标记为 MDDI\_Data0+/- 和 MDDI\_Stb+/- 的数据信号。这些信号的每一个均为经由电缆中的差分电线对来传送的低压数据信号。对于通过所述接口发送的每一位,MDDI\_Data0 对或者 MDDI\_Stb 对上只存在一个跃迁。这是基于电压而非基于电流的传送机制,因此静态电流消耗接近零。主机把 MDDI\_Stb 信号驱动到客户机显示器。

[0185] 当数据可以经由 MDDI\_Data 对沿正反方向流动时,也就是说它是双向传送路径时,主机是数据链路的主控设备或者控制器。所述 MDDI\_Data0 和 MDDI\_Stb 信号路径按照差分模式工作,以便使抗扰能力最强。这些线路上信号的数据速率是由主机发送的时钟速率来确定的,并且能够在 1kbps 直到 400Mbps 或更大的范围内改变。

[0186] 除了类型 1 接口的数据对或路径以外,类型 2 接口还包含一个附加数据对或者导线或者路径,称为 MDDI\_Data1+/-。除了类型 2 接口的那些数据对或路径以外,类型 3 接口包含两个附加数据对或者信号路径,称为 MDDI\_Data2+/- 和 MDDI\_Data3+/-。除类型 3 接口的数据对或路径以外,类型 4 接口包含另外四个数据对或者信号路径,分别称为:MDDI\_Data4+/-、MDDI\_Data5+/-、MDDI\_Data6+/- 和 MDDI\_Data7+/-。在上述接口配置的每一个中,主机可以使用线对或者指定为 HOST\_Pwr(主机电源)和 HOST\_Gnd(主机地)的信号来向客户机或者显示器提供电力。如下文将进一步讨论的,如果需要,当正在使用的接口“类型”采用的导线少于在其他模式中可用的或存在的导线时,在某些配置中 MDDI\_Data4+/-、MDDI\_Data5+/-、MDDI\_Data6+/- 或者 MDDI\_Data7+/- 导线,也可以用于进行电力传送。虽然一些应用存在差异,但电力传送通常为外部模式所使用,内部模式通常不需要电力传送。

[0187] 下面,按照接口类型,在表 I 中说明了在各种模式下,经由 MDDI 链路在主机和客户

机（显示器）之间传递的信号概要。

[0188] 表 I

[0189]

类型 1	类型 2	类型 3	类型 4
HOST_Pwr/Gnd	HOST_Pwr/Gnd	HOST_Pwr/Gnd	HOST_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+/-
可选 Pwr	可选 Pwr	可选 Pwr	MDDI_Data4+/-
可选 Pwr	可选 Pwr	可选 Pwr	MDDI_Data5+/-
可选 Pwr	可选 Pwr	可选 Pwr	MDDI_Data6+/-
可选 Pwr	可选 Pwr	可选 Pwr	MDDI_Data7+/-

[0190] 还应注意的是，用于从主机传送的 HOST\_Pwr/Gnd 连接通常是外部模式提供的。内部应用或者操作模式通常让客户机直接从其它内部资源接电源，而不使用 MDDI 来控制电源分布，正如对于本领域普通技术人员显而易见的那样，因此没有在此处进一步详细地描述这种分布。然而，正如本领域普通技术人员将理解的那样，当然能够经由 MDDI 接口来分配电源以方便例如某种电源控制、同步或者互连。

[0191] 通常用于实现上述结构和操作的电缆标称量级为大约 1.5 米的长度，一般为 2 米或者更短，并包括三对双绞线导线，每一导线还是多股 30AWG 线。把箔屏蔽覆盖物包裹或者形成在三对双绞线之上，作为另外的地线 (drain wire)。所述双绞线和屏蔽地线导线在显示器连接器中终止，屏蔽层与客户机的屏蔽层连接，还有一个绝缘层，覆盖整个电缆，这些都正如所属技术领域已知的那样。所述电线按以下方式配对：Host\_Gnd 与 Host\_Pwr；MDDI\_Stb+ 与 MDDI\_Stb-；MDDI\_Data0+ 与 MDDI\_Data0-；MDDI\_Data1+ 与 MDDI\_Data1-；等等。然而，如本领域已知的那样，可以根据特定的应用，使用各种导线和电缆来实现本发明的实施例。例如，在一些应用中，重的外部覆盖物或金属层可用于包含电缆，而细的、扁平的传导带型结构可能非常适合于其他应用。

[0192] D. 数据类型和速率

[0193] 为了实现用于用户体验和应用的有用接口，移动式数字数据接口 (MDDI, Mobile Digital Data Interface) 支持各种客户机和显示信息、音频转换器、键盘、指示设备和可以集成到移动式显示器设备或者与之协同工作的多种其它输入 / 输出设备，以及控制信息和它们的组合。所述 MDD 接口被设计成能够适应各种潜在类型的数据流，这些数据流使用最小数目的电缆或者导线沿正向或者反向链路方向在主机和客户机之间传送。同步流和异步流（更新）都得以支持。只要总计数数据速率小于或等于最大的期望 MDDI 链路速率，许多



数据类型的组合都是可能的。这一 MDDI 链路速率受到最大串行速率和所采用的数据对数目的限制。这些数据速率可以包括但不局限于下面表 II 和 III 中列出的那些项。

[0194] 表 II

从主机向客户机传送			
[0195]	同步视频数据	720×480,12 位,30f/s	~124.5 Mbps
	同步立体声音频数据	44.1kHz,16 位,立体声	~1.4 Mbps
	异步图形数据	800×600,12 位,10f/s,立体声	~115.2 Mbps
[0196]	异步控制	最小值	<< 1.0 Mbps

[0197] 表 III

从客户机向主机传送			
[0198]	同步语音数据	8 kHz, 8 位	<< 1.0 Mbps
	同步视频数据	640×480,12 位,24f/s	~ 88.5 Mbps
	异步状态、用户输入等等	最小值	<< 1.0 Mbps

[0199] 为了满足未来系统的灵活性要求,所述接口不是固定的,而是可扩展的,因此它可以支持包括用户定义的数据的各种信息“类型”的传送。所支持的数据的具体实例是:全运动视频,具有全屏幕或部分屏幕位图字段的形式或者压缩视频形式的视频;处于低速率以便节省功率并且降低实现成本的静态位图;以多种分辨率或者速率的 PCM 或者压缩音频数据;指示设备定位和选择,以及针对待定义能力的用户可定义数据。这种数据还可以随控制或状态信息一起传送,以便检测设备能力或设置操作参数。

[0200] 本发明的实施例给出了用于数据传送的技术,所述技术包括但不限于:观看电影(视频显示和音频);使用具有有限个人观看的个人计算机(图形显示、有时结合视频和音频);在 PC、控制台或者个人设备上玩视频游戏(运动图形显示或者合成的视频和音频);使用视频电话,静态数字图像照相机,或用于拍摄数字视频图像的摄录一体机形式的设备在因特网上“冲浪”(双向低速率视频和音频);使用扩展有投影仪的电话或者 PDA 来呈现或者与连接至视频监视器、键盘以及鼠标的台式扩展坞扩展连接;以及使用蜂窝电话、智能电话或者 PDA,包括无线指示设备和键盘数据,来增强生产力或者用于娱乐。

[0201] 如下所述的高速数据接口是以经由通信或者传送链路提供大量 A-V 类型数据的形式而提供的,所述通信或者传送链路通常被配置成电线线路或者电缆类型链路。然而,显而易见,可以调节信号结构、协议、时序或者传送机制以便提供光学或者无线介质形式的链路,只要该链路可以保持所需要的数据传送水平即可。

[0202] MDDI 信号把名为公共帧速率(CFR)的概念用于基本信号协议或者结构。支持使用公共帧速率的思想在于为同时的同步数据流提供同步脉冲。客户机设备可以使用这一公共帧速率作为时间基准。通过减少用于传输子帧报头的开销,低 CF 速率提高了通道效率。另一方面,高 CF 速率能够减少等待时间,并且允许将弹性更小的数据缓冲器用于音频采样。

本发明的接口的 CF 速率是可动态编程的,并且可以被设置为适合于特殊应用中使用的多个同步流的多个值之一。也就是说,根据需要,选择所述 CF 值来最佳地适应给定的客户机和主机结构。

[0203] 表 IV 中示出了例如视频或微型显示器之类的应用中很可能使用的同步数据流的每个子帧通常需要的字节数目,这个数目是可调节的或者是可编程的。

[0204] 表 IV

公共帧速率 (CFR) = 300 Hz							
	X	Y	位	帧速率	通道	速率 (Mbps)	字节/子帧
[0205] 计算机游戏	720	480	24	30	1	248.832	103680
计算机图形	800	600	24	10	1	115.200	48000
视频	640	480	12	29.97 或者 30	1	221.184	92160
CD 音频	1	1	16	44100	2	1.4112	588
语音	1	1	8	8000	1	0.064	26-2/3

[0206] 使用简单的可编程 M/N 计数器结构可以很容易地计算出每子帧的分数字节。例如,通过传送 2 个 27 字节的帧,每次后面跟随一个 26 字节的帧,可以实现对每个 CF 26-2/3 字节的计数。可以选择更小的 CF 速率来产生每个子帧的整数字节。然而,一般说来,与较大音频采样 FIFO 缓冲器所需的区域相比,在用于实现本发明的部分或者所有实施例的集成电路芯片或者电子模块中,用硬件实现简单的 M/N 计数器需要的区域较小。

[0207] 说明不同数据传送速率和数据类型的影响的示例性应用是卡拉 OK 系统。对于卡拉 OK 系统来说,最终用户或者多个最终用户与音乐视频节目一起演唱。将歌曲的歌词显示在屏幕的某处,通常在屏幕底部,因此,用户知道将演唱的词,并且大致知道歌曲的时间安排。这一应用需要不经常进行图形更新的视频显示,并且把一个或者多个用户语音与立体声音频流混合。

[0208] 如果采用的公共帧速率是 300Hz,那么每一子帧将包括:经由正向链路至客户机显示器设备的 92,160 字节的视频内容和 588 字节的音频内容(在立体声中基于 14716-位采样),以及从麦克风送回到移动式卡拉 OK 机的平均值为 26.67(26-2/3)字节的语音。在主机和显示器(有可能是头带式的)之间发送异步分组。这包括最多 768 字节的图形数据(四分之一屏幕高度),以及少于约 200 字节(几字节)用于各种各样的控制和状态命令。

[0209] 表 V 示出了对于卡拉 OK 实例而言,在子帧内如何分配数据。将使用的总速率选择为约 279Mbps。稍高于 280Mbps 的速率允许传送每子帧大约又一 400 字节的数据,这样就允许利用偶尔出现的控制和状态信息。

[0210] 表 V

[0211]

元素速率	每个子帧的 开销字节	每个子帧的 介质字节
640×480 像素, 30fps 的音乐视频	$2*28=56$	92160
640×120 像素, 1fps 的歌词文本 10 个子帧中更新, 1/30 秒	28	768
44,100sps, 立体声, 16 位的 CD 音频	$2*16=32$	588
8,000sps, 单声道, 8 位的语音	$28+8+8+(4*16)+$ $(3*27)=125$	26.67
子帧报头	22	
总字节/CF	263	115815
总速率 (Mbps)	$(263+115815) * 8 * 300 = 278.5872$	

[0212]

[0213] III. 高速率数字数据接口系统体系结构 (续)

[0214] E. 链路层

[0215] 使用 MDD 接口高速串行数据信号传送的数据包括时间复用分组流, 其中这些分组一个接着一个地链接。即使传输设备没有要发送的数据, MDDI 链路控制器通常也自动地发送填充符分组, 由此维持分组流。因此, 简单分组结构的使用能够确保视频和音频信号或者数据流可靠的同步时序。

[0216] 多组分组包括在称为子帧的信号元素 (signal element) 或者结构内, 而子帧组则包含在称为介质帧的信号元素或者结构内。根据子帧各自的大小和数据传送用途, 子帧包含一个或多个分组, 而介质帧则包含一个或多个子帧。这里给出的实施例所采用的协议提供的最大子帧大约为  $2^{32}-1$  或者 4, 294, 967, 295 字节的量级, 而最大介质帧的大小大约为  $2^{16}-1$  或者 65, 535 个子帧的量级。

[0217] 一个特殊的子帧报头分组包含唯一的标识符, 该标识符在每一子帧开始时出现, 正如下面将讨论的那样。当启动主机和客户机之间的通信时, 该标识符还用于获取客户机设备处的帧时序。下面比较详细地讨论链路时序的获得。

[0218] 通常, 显示全运动视频时, 每一介质帧更新一次显示屏幕。显示帧速率与介质帧速率相同。根据所需要的应用, 链路协议支持整个显示器上的全运动视频, 或者仅仅支持由静态图像围绕的全运动视频内容的一个小区域。在某些低功率移动式应用中, 诸如观看网页或者电子邮件, 只须偶而更新显示屏。在这些情况下, 传输单个子帧, 然后关闭链路或者撤销链路以便使功耗最小是非常有益的。所述接口还支持诸如立体视觉的效果, 并且处理图形图元。

[0219] 子帧使得系统能够周期性地传输高优先级分组。这样做使得同时发生的同步流能够与最小量的数据缓存共存。这是提供给显示过程的一个有益实施例, 它允许多个数据流 (视频、语音、控制、状态、指示设备数据等等的高速通信) 基本上共享共同通道。该接口

使用相对少数的信号传送信息。该接口还允许存在特定于显示技术动作,诸如用于 CRT 监视器的水平同步脉冲和消隐间隔 (blanking intervals),或其他特定于客户机 - 技术的动作。

#### [0220] F. 链路控制器

[0221] 图 4 和 5 中示出的 MDDI 链路控制器将按照完全数字化实现方式来制造或者装配,其中不包括用来接收 MDDI 数据和选通信号的差分线路接收器。然而,例如当制作 CMOS 型 IC 时,差分线路驱动器和接收器甚至可以与链路控制器一起在同一数字集成电路中实现。不需要模拟功能或者锁相环 (PLL) 来进行位恢复或者实现链路控制器的硬件。除包含用于链路同步的状态机的客户机接口之外,主机和客户机链路控制器包含非常类似的功能。因此,本发明的实施例允许创建单个控制器设计或电路,并可以将它配置为主机或者客户机,这有利于实际应用,而且总的来说,这样做可以降低链路控制器的制造成本。

#### [0222] IV. 接口链路协议

##### [0223] A. 帧结构

[0224] 图 6 中示出了用于实现分组传送的正向链路通信的信号协议或者帧结构。如图 6 所示,把信息或者数字数据组合为公知为分组的元素 (element)。继而,多个分组又被组合在一起以形成所谓“子帧”,并且多个子帧又被组合在一起以形成“介质”帧。为了控制帧的形成和子帧的传送,每一子帧从专门预定的分组开始,该分组称为子帧报头分组 (SHP)。

[0225] 所述主机设备选择将用于给定传送的数据速率。主机设备可以基于主机的最大传送能力、由主机从信息源取回 (retrieve) 的数据,以及客户机或者该数据将要传送到的其它设备的最大能力,而动态地改变此速率。

[0226] 设计成或者能够与 MDDI 或者本发明的信号协议协同工作的接收方客户机设备能够由主机进行查询,以便确定该客户机可以使用的最大或者当前最大的数据传送速率,或者可以使用的默认较慢最小速率以及可用的数据类型和所支持的特征。这一信息可以使用显示器能力分组 (DCP) 来传送,如下文将进一步讨论的那样。所述客户机显示器设备能够使用所述接口以预选最小数据速率或者在最小数据速率范围内传送数据或者与其它设备通信,并且所述主机将使用此范围内的数据速率执行查询以便确定客户机设备的全部能力。

[0227] 可以在状态分组中向主机传送定义位图属性和客户机的视频帧速率能力的其它状态信息,因此主机可以按照在任何系统限制范围内的实际情况或需要,尽可能高效或优化地配置所述接口。

[0228] 当在本子帧中没有 (更多) 将要传送的分组时,或者当主机无法以足以跟上 (keep pace) 正向链路所选数据传输速率的速率进行传送时,所述主机发送填充符分组。由于每一子帧从子帧报头分组开始,所以前一子帧的末尾包含正好填充前一子帧的分组 (很可能是填充符分组)。在承载数据的分组本身缺乏空间的情况下,填充符分组很可能是子帧中最后的分组,或者在紧邻的先前 (next previous) 子帧的结尾并且在子帧报头分组之前。在主机设备中,控制操作的任务就是确保对于该子帧内待传输的每个分组在子帧中都保留有足够的空间。同时,一旦主机设备开始发送数据分组,主机就必须能够成功地使帧内该大小的分组完整,而不能引起数据欠载 (under-run) 情况。

[0229] 在实施例的一个方面中,子帧传输具有两种模式。一种模式是周期性的子帧模式,

或者周期性的时元 (periodic timing epochs), 用于传输实况播送的视频和音频流。在这种模式中, 把子帧长度定义为是非零的。第二种模式是异步或者非周期性模式, 其中当有新的信息时, 才使用帧来把位图数据提供给客户机。这种模式是通过在子帧报头分组中把子帧长度设置为零来定义的。使用周期性模式时, 当客户机与正向链路帧结构同步时, 可以开始子帧分组接收。这对应于按照下文就图 49 或图 63 讨论的状态图定义的“同步中 (in sync)”的状态。在异步非周期性子帧模式中, 在收到第一子帧报头分组之后开始接收。

#### [0230] B. 总体分组结构

[0231] 下面说明由所述实施例实现的用于规定通信或信令协议或传送数据的方法和手段的分组格式或者结构, 应记住的是, 所述接口是可扩展的, 根据需要可以添加附加分组结构。所述分组就其在接口中的功能, 即它们传送的或与之相关的命令、信息、数值或数据而言, 被标记为或者分为不同的“分组类型”。因此, 对于用于操纵被传送的分组和数据的给定分组, 每一分组类型表示一个预定义的分组结构。显而易见, 所述分组可以具有预选长度, 或者根据各自功能具有可变或者动态可变的长度。所述分组还可以拥有不同的名称, 虽然实现的仍然是相同的功能, 正如在将协议接受到标准里去的过程中协议发生改变时出现的情况一样。用于各种分组的字节或者字节值被配置成多位 (8 或 16 位) 无符号整数。所采用的分组概要以及它们的“类型”指定被按照类型顺序列出, 并在表 VI-1 至 VI-4 中示出。

[0232] 为了便于举例说明和理解, 每一表均给出了总体分组结构内的通用“类型”分组。这些分组不暗含或者表示对本发明的某些限制或者其它影响, 并且根据需要可以按照多个其它方式来组织所述分组。还指出了认为分组传送有效的方向。

#### [0233] 表 VI-1

#### [0234] 链路控制分组

#### [0235]

分组名称	分组类型	正向有效	反向有效
子帧报头分组	15359	x	
填充符分组	0	x	x
反向链路封装分组	65	x	
链路关闭分组	69	x	
接口类型切换请求分组	75	x	
接口类型确认分组	76		x
执行类型切换分组	77	x	
往返行程延迟测量分组	82	x	
正向链路偏离校准分组	83	x	

#### [0236]

[0237] 表 VI-2

[0238] 基本媒体流分组

[0239]

分组名称	分组类型	正向有效	反向有效
视频流分组	16	x	x
音频流分组	32	x	x
保留的流分组	1-15, 18-31, 33-35	x	x
用户定义的流分组	56-63	x	x
色彩映射分组	64	x	x
正向音频信道启用分组	78	x	
反向音频采样速率分组	79	x	
透明色彩启用分组	81	x	

[0240] 表 VI-3

[0241] 客户机状态和控制分组

[0242]

分组名称	分组类型	正向有效	反向有效
客户机能力分组	66		x
键盘数据分组	67	x	x
指示设备数据分组	68	x	x
客户机请求和状态分组	70		x
数字内容保护开销分组	80	x	x
请求 VCP 特征分组	128	x	
VCP 特征应答分组	129		x
设置 VCP 特征分组	130	x	
请求有效参数分组	131	x	
有效参数应答分组	132		x

[0243]

请求特定状态分组	138	x	
有效状态应答列表分组	139		x
分组处理延迟参数分组	140		x
个人显示器能力分组	141		x
客户机错误报告分组	142		x
可缩放视频流能力分组	143		x
显示器标识分组	144		x
可选显示器能力分组	145		x
寄存器访问分组	146	x	x

[0244] 表 VI-4

[0245] 高级图形和显示分组

[0246]

分组名称	分组类型	正向有效	反向有效
位块传送分组	71	x	
位图区域充分组	72	x	
位图图案充分组	73	x	
读取帧缓冲器分组	74	x	
可缩放视频流能力分组	143		x
可缩放视频流建立分组	136	x	
可缩放视频流确认分组	137		x
可缩放视频流分组	18	x	

[0247] 根据本文中的其它讨论可以清楚了解的是,虽然对于外部模式操作而言,反向封装分组、客户机能力分组以及客户机请求和状态分组被认为是非常重要的,甚至在通信接口的许多实施例中是需要的,但是可以认为它们对于内部模式操作而言是可选的。这样做产生了另一类型的 MDD 接口协议,该协议允许利用缩减的通信分组集合以非常高的速度来传递数据,并且相应地简化控制与时序。

[0248] 分组具有公共基本结构或者最少字段的总体集合,所述最少字段包括分组长度字

段、分组类型字段、数据字节字段和 CRC 字段,这些内容在图 7 中示出。如图 7 所示,分组长度字段包含具有多位或者字节值形式的信息,用于指定分组的总位数,或者指定分组长度字段和 CRC 字段之间的长度。在一个实施例中,所述分组长度字段包含 16 位或者 2 字节宽的无符号整数,它说明分组长度。分组类型字段是又一多位字段,它指明分组内包含的信息的类型。在一个示例性的实施例中,该分组类型字段是 16 位或者 2 字节宽、具有 16 位无符号整数形式的值,并且将这些数据类型说明为显示器能力、切换、视频或者音频流、状态等等。

[0249] 第三个字段是数据字节字段,它包含在主机和客户机设备之间作为分组的一部分传送或者发送的位或者数据。按照被传送数据的具体类型,为每个分组类型具体定义了数据格式,并且该分组格式可以分割成一系列附加字段,每一个都具有自身的格式要求。也就是说,每一分组类型将具有对这一部分或者字段的已定义格式。最后的字段是 CRC 字段,它包含在数据字节、分组类型和分组长度字段上计算出来的 16 位循环冗余校验的结果,该结果用来确认分组中信息的完整性。换言之,该结果是在除 CRC 字段本身之外的整个分组上计算出来的。客户机通常保存检测到的 CRC 错误的总计数,并且在客户机请求和状态分组中将此计数报告给主机(参见下文)。

[0250] 通常,这些字段宽度和组织结构被设计成能使 2 字节字段在偶数字节边界上对准,4 字节字段在 4 字节边界上对准。这样做允许分组结构易于嵌入主机和客户机的,或者与之相关联的主存储器空间,而不会违反大多数或者通常使用的处理器或者控制电路中遇到的数据类型对准规则。

[0251] 在传送所述分组期间,字段首先从最低有效位(LSB)开始传输,并且最后以传输最高有效位(MSB)结束。一个以上字节长度的参数首先传输最低有效字节,这样得到的用于传输长度大于 8 位的参数的位传输模式,与首先传输 LSB 的较短参数所使用的位传输模式相同。每一分组的数据字段通常按照在下文后续段落定义的顺序来传输,所列出的第一字段首先传输,而最后描述的字段最后传输。在类型 1、类型 2、类型 3 或者类型 4 这些模式中的任意一个模式里,MDDI\_Data0 信号路径上的数据与接口上传输的字节的‘0’位对准。

[0252] 当操纵数据来显示时,像素阵列的数据首先按照行,然后按照列来传输,就像电子技术领域中常规执行的那样。换言之,位图中同一行里出现的所有像素按照顺序传输,最左侧的像素首先传输,最右侧的像素最后传输。在一行中最右侧像素传输完之后,然后按照顺序下一像素是下一行最左侧的像素。对于大多数显示器而言,像素的行通常按照自上而下的顺序来传输,但也可以根据需要采用其它配置方式。此外,在处理位图的过程中,常规方法也就是此处遵循的方法是通过将位图左上角标记为位置或者偏移“0,0”来定义基准点的。当分别接近位图的右侧和底部时,用于定义或者确定在位图中的位置的 X 和 Y 坐标的值增大。第一行第一列(图像的左上角)从下标值零开始。从显示器用户的角度观看时,朝向图像右侧 X 坐标的数值增加,朝向图像的底部 Y 坐标的数值增加。

[0253] 显示窗是位图的可视部分,即,位图中用户能够在物理显示介质上看见的那部分像素。显示窗和位图经常大小相同。显示窗左上角始终显示位图像素位置 0,0。显示窗的宽度对应于位图的 X 轴,并且显示窗宽度小于或等于相应位图的宽度。窗的高度对应于位图的 Y 轴,并且显示窗高度小于或等于对应位图的高度。显示窗本身在协议中没有被涉及,这是因为仅仅将它定义为位图的可视部分。



[0254] 位图和显示窗之间的关系是计算机、电子领域,因特网通信和其他电子相关领域中公知的。因此这里未给出这些原理的进一步讨论或说明。

[0255] C. 分组定义

[0256] 1. 子帧报头分组

[0257] 子帧报头分组是每一子帧的第一个分组,并且具有如图 8 所示的基本结构。子帧报头分组用于主机-客户机同步,每一主机应该能够生成这个分组,同时每一客户机应该能够接收并且解释这个分组。正如可以在图 8 中看到的那样,通常将这种类型的分组构造为按顺序具有分组长度、分组类型、唯一字、保留 1、子帧长度、协议版本、子帧计数和介质帧计数字段。在一个实施例中,通常将这种类型的分组标识为类型 15359(十六进制 0x3bff)分组,并且使用 20 字节的预选固定长度,不包括分组长度字段。

[0258] 所述分组类型字段和唯一字字段均使用 2 字节值(16 位无符号整数)。将这两个字段的 4 字节组合在一起形成了具有良好的自相关性的 32 位唯一字。在一个实施例中,实际的唯一字是 0x005a3bff,其中低 16 位作为分组类型首先传输,然后传输最高 16 位。

[0259] 保留 1 字段包含 2 字节的保留空间以备将来之用,并且在这里一般将其配置成将所有位均置零。此字段的一个目的在于让后续 2 字节字段与 16 位字地址对准,让 4 字节字段与 32 位字地址对准。保留最低有效字节,以表明所述主机是否能够访问多个客户机设备。这个字节保留为零这个值,以表明这个主机只能够与单个客户机设备一起工作。

[0260] 子帧长度字段包含 4 字节的信息或者值,用于说明每个子帧的字节数。在一个实施例中,这个字段的长度可以被设置为等于零,以便表明在链路被关闭而进入空闲状态之前,主机将只传输一个子帧。从一个子帧跃迁到下一子帧时,这个字段中的值可以“在工作过程中(on-the-fly)”动态地改变。为了在同步脉冲中进行较小时序调整以便支持同步数据流,这一能力是十分有用的。如果子帧报头分组的 CRC 无效,那么链路控制器就应该使用已知完好的先前的子帧报头分组的子帧长度来估计当前子帧的长度。

[0261] 协议版本字段包含 2 字节,用于说明由主机使用的协议的版本。将协议版本字段设置为‘0’说明正在使用的协议的第一个或者当前版本。当创建了新版本的时候,此值将随时间而改变,并且对于一些版本字段已经更新到值 1。正如所公知的那样,版本值很可能或通常遵从包括如 MDDI 接口的已批准的标准文档的当前版本号。

[0262] 子帧计数字段包含 2 字节,指明一个序号,该序号表明自介质帧开始处已经传输的子帧的数目的序号。介质帧的第一子帧的子帧计数为零。介质帧最后子帧具有 n-1 的值,其中 n 是每介质帧的子帧数目。子帧计数字段的值等于在前一子帧分组中发送的子帧计数再加 1。应注意的是,如果子帧长度被设置为等于零(表明非周期性的子帧),那么子帧计数也必须被设置为等于零。

[0263] 介质帧计数字段包含 4 字节(32 位无符号整数),用于说明一个序号,该序号表明自正在传送的当前介质项或者数据的开始处,已经传输的介质帧的数目的序号。介质项目的第一介质帧的介质帧计数为零。介质帧计数刚好在每一介质帧的第一子帧之前增加,并且在使用了最大介质帧计数(例如,介质帧数目  $2^{32}-1 = 4,294,967,295$ )之后返回零。介质帧计数值通常随时可由主机重置,以便适应最终应用的需求。

[0264] 2. 填充符分组

[0265] 填充符分组是没有其它信息可以用来在正向或者反向链路上发送时,被传送到客

户机设备或从客户机设备传送的分组。建议填充符分组具有最小长度,以便当需要时,在发送其它分组的过程中具有最大的灵活性。在子帧或者反向链路封装分组(参见下文)的结尾,链路控制器设置填充符分组的大小以便填充剩余空间,从而保持分组的完整性。当主机或者客户机没有信息要发送或交换时,所述填充符分组对于保持链路上的时序十分有用。每一主机和客户机都应该能够发送并且接收这个分组,以便有效使用所述接口。

[0266] 图9中示出了填充符分组的格式和内容的一个示例性的实施例。如图9所示,将这种类型的分组构造为具有分组长度、分组类型、填充符字节和CRC字段。在一个实施例中,通常将这种类型的分组标识为类型0,这在2字节类型字段中表明。填充符字节字段中的位或者字节包括可变数目的全零的位值,以便允许填充符分组具有所期望的长度。最小填充符分组在这个字段中不包含字节。也就是说,所述分组只包括分组长度、分组类型和CRC,并且在一个实施例中,使用6字节的预选固定长度或者分组长度值4。为分组中的所有字节确定所述CRC值,其中包括分组长度,但是在其它分组类型中,该分组长度是被排除的。

### [0267] 3. 视频流分组

[0268] 视频流分组通常携带视频数据来更新显示器设备的通常为矩形的区域。此区域的大小可以像单个像素那么小,或者像整个显示器那么大。可以有几乎无限数目的流被同时显示,但其受系统资源的限制,这是因为视频流分组内包含显示一个流所需要的所有上下文。图10中示出了视频流分组(视频数据格式描述符)的一个实施例的格式。如图10所示,在一个实施例中,将这种类型的分组构造为具有分组长度(2字节)、分组类型、bClient ID、视频数据描述符、像素显示属性、X左沿、Y顶沿、X右沿、Y底沿、X和Y起始、像素计数、参数CRC、像素数据和CRC字段。通常将这种类型的分组标识为类型16,其在2字节类型字段中表明。在一个实施例中,客户机使用客户机能力分组中的RGB、单色和Y Cr Cb能力字段来表明接收视频流分组的能力。

[0269] 在一个实施例中,bClient ID字段包含为客户机ID保留的2字节信息。由于这是最新开发的通信协议,所以实际的客户机ID仍无法获知或者足以被传送。因此,这个字段中的位通常设置为等于零,直到获知这个ID值为止,当或者这个ID时,可以插入或者使用所述ID值,正如对于本领域技术人员显而易见的那样。对于以下讨论的客户机ID字段通常也执行与此相同的处理。

[0270] 上述的公共帧概念是用于最小化音频缓冲区大小并且减少等待时间的一种有效方式。然而,对于视频数据来说,可能必需将一个视频帧的像素扩展到介质帧内的多个视频流分组。还可能单个视频流分组中的像素不会正好对应于显示器上完整的矩形窗。对于30帧每秒的示例性视频帧速率来说,每秒有300子帧,这使得每一介质帧有10个子帧。如果每一帧中有480行像素,那么每一子帧中的每一视频流分组将包含48行像素。在其它情况中,视频流分组可能不包含整数个像素行。这对于每一介质帧的子帧数目不能被每一视频帧的行数(亦称视频行)均分的其它视频帧大小而言是这样的。为了有效操作,每一视频流分组通常必须包含整数个像素,即便它未必包含整数个像素行。如果每个像素均大于一个字节,或者如果它们具有如图12所示的分组格式,那么这是非常重要的。

[0271] 图11A-11E中示出了如上所述的用于实现示例性视频数据描述符字段的操作所采用的格式和内容。在图11A-11E中,视频数据格式描述符字段包含具有16位无符号整数形式的2字节,用于说明当前分组的当前流中像素数据内每一像素的格式。不同的视频流

分组可能使用不同的像素数据格式,即视频数据格式描述符中使用不同的值,同样,流(显示区域)可以在工作过程中(on-the-fly)改变其数据格式。像素数据格式应该符合按照客户机能力分组定义的至少一个客户机有效格式分组。只有在当前分组的像素格式暗含在特定视频流的使用期内,不再继续使用不变的格式的时候,视频数据格式描述符才定义当前分组的像素格式。

[0272] 图 11A 至 11D 说明视频数据格式描述符是如何编码的。正如这些图中所使用的那样,并且在这个实施例中,当位 [15:13] 等于‘000’时,如图 11A 所示,那么所述视频分组括单色像素阵列,其中每一像素的位数由视频数据格式描述符字的位 3 至位 0 定义。通常保留位 11 至 4 以便将来使用或者用于其它应用,并且在这种情况下被设置为零。当位 [15:13] 换成等于值‘001’时,如图 11B 所示,那么所述视频分组括一个彩色像素阵列,其中每一个像素都通过色彩映射(调色板)来指定一种颜色。在这种情况下,视频数据格式描述符字的位 5 至 0 定义每一像素的位数,并且位 11 至 6 通常被保留下来以供将来使用或应用,并且这些位被设置为等于零。当位 [15:13] 换成等于值‘010’时,如图 11C 所示,那么所述视频分组括彩色像素阵列,其中每一像素红色的位数由位 11 至 8 定义,每一像素绿色的位数由 7 至 4 位定义,并且每一像素蓝色的位数由位 3 至 0 来定义。在这种情况下,每一像素中位数的总数是红色、绿色和蓝色所使用的位数和。

[0273] 然而,当位 [15:13] 换成等于值或字串‘011’时,如图 11D 所示,那么视频分组括具有 4:2:2 的 YCbCr 格式的带有亮度和色度信息的一个视频数据阵列,其中每一像素亮度(Y)的位数由位 11 至 8 定义,Cb 分量的位数由位 7 至 4 定义,Cr 分量的位数由位 3 至 0 定义。每一像素中位数总数是红色、绿色和蓝色所使用的位数和。Cb 和 Cr 分量是以 Y 分量一半的速率发送。另外,这个分组的像素数据部分中的视频采样被组织为:Cb<sub>n</sub>、Y<sub>n</sub>、Cr<sub>n</sub>、Y<sub>n+1</sub>、Cb<sub>n+2</sub>、Y<sub>n+2</sub>、Cr<sub>n+2</sub>、Y<sub>n+3</sub>,...,其中 Cb<sub>n</sub> 和 Cr<sub>n</sub> 与 Y<sub>n</sub> 和 Y<sub>n+1</sub> 相关联,并且 Cb<sub>n+2</sub> 和 Cr<sub>n+2</sub> 与 Y<sub>n+2</sub> 和 Y<sub>n+3</sub> 相关联等等。

[0274] Y<sub>n</sub>、Y<sub>n+1</sub>、Y<sub>n+2</sub> 和 Y<sub>n+3</sub> 是单个行中从左至右的四个连续像素的亮度值。如果在视频流分组所定位的窗口的一行中存在有奇数个像素(X 右沿 -X 左沿 +1),那么对应于每一行中最后像素的 Y 值后面将跟随有下一行第一像素的 Cb 值,并且不发送该行中最后像素的 Cr 值。建议使用 Y Cb Cr 格式的窗口具有偶数个像素的宽度。分组中的像素数据应该包含偶数个像素。在像素数据的最后像素对应于视频流分组报头中指定的窗口里的行的最后像素的情况下,即,当像素数据中最后像素的 X 位置等于 X 右沿时,像素数据可以包含奇数或者偶数个像素。

[0275] 当位 [15:13] 换成等于‘100’时,那么视频分组括一个间隔(Bayer)像素阵列,其中每一像素的位数由视频数据格式描述符字的位 3 至 0 定义。像素组图案由位 5 和 4 定义,如图 11E 所示。像素数据的顺序可以是水平或者垂直的,处于行或列中的像素可以按照向前或向后的顺序来发送,并且由位 8 至 6 来定义。位 11 至 9 应该被设置为零。间隔(Bayer)格式中像素组内的四个像素的组类似于在一些显示技术中经常称作耽搁像素的情况。然而,在间隔(Bayer)格式中的一个像素只是该像素组拼接(mosaic)图案中四个彩色像素中的一个。

[0276] 对于图中示出的所有五个格式来说,指定为“P”的位 12 说明像素数据采样是打包的像素数据,还是字节对准的像素数据。这个字段中的值‘0’表明像素数据字段中的每一

像素都与 MDD 接口字节边界字节对准。值‘1’表明对像素数据中每一像素内的每一像素和每一颜色打包,从而相对于像素内的先前像素或色彩没有未使用位。图 12 更加具体地示出了字节对准的像素格式和打包的像素数据格式之间的差别,其中可以清楚地看到字节对准数据留有数据子帧的未使用部分,这与打包像素格式中未留有未使用部分的情况相对。

[0277] 用于特定显示窗的介质帧的第一个视频流分组中的第一个像素会进入由 X 左沿和 Y 顶沿定义的流窗口的左上角,而所接收的下一像素则放置在同一行的下一像素位置,等等。在介质帧的该第一个分组中,X 起始值通常等于 X 左沿,而 Y 起始值通常等于 Y 顶沿。在对应于同一屏幕窗口的后续分组中,通常将 X 和 Y 起始值设置为屏幕窗口中该像素的位置,该像素位置通常跟随在在先前子帧中传输的视频流分组中发送的最后一个像素之后。

#### [0278] 4. 音频流分组

[0279] 音频流分组携带将通过客户机音频系统播放的或者用于独立的音频呈现设备的音频数据。可以把不同的音频数据流分配给音响系统中独立的音频通道,例如:左前、右前、中央、左后和右后,这取决于正使用的音频系统的类型。为包含增强的空间声信号处理的耳机提供全部音频通道。客户机使用客户机能力分组的音频通道能力和音频采样速率字段来表明接收音频流分组的能力。图 13 中说明音频流分组的格式。

[0280] 如图 13 所示,在一个实施例中将这种类型的分组构造为具有分组长度、分组类型、bClient ID、音频通道 ID、保留 1、音频采样计数、每个采样的位及其打包、音频采样速率、参数 CRC、数字音频数据和音频数据 CRC 字段。在一个实施例中,通常将这种类型的分组标识为类型 32 分组。

[0281] bClient ID 字段包含为客户机 ID 保留的 2 字节信息,如先前使用的那样。保留 1 字段包含被保留下来以备将来之用的 2 字节,并且在这里配置为将所有位设置为零。

[0282] 每个采样的位及其打包字段包含具有 8 位无符号整数形式的 1 个字节,用于说明音频数据的打包格式。通常采用的格式是使用位 4 至 0 来定义每一 PCM 音频采样的位数。然后,位 5 说明数字音频数据采样是否被打包。图 14 中示出了已打包音频采样和字节对准音频采样之间的差异,这里使用 10- 位采样。值‘0’表明数字音频数据字段中的每一 PCM 音频采样与 MDDI 接口字节边界字节对准,并且值‘1’表明每一连续 PCM 音频采样被相对于先前音频采样而打包。通常只有当位 4 至 0(每一 PCM 音频采样的位数)定义的值不是 8 的倍数时,这一位才是有效的。位 7 至 6 被保留以供将来使用,并且通常设置为零。

#### [0283] 5. 保留的流分组

[0284] 在一个实施例中,根据遇到的各种应用的要求,分组类型 1 至 15、18 至 31 和 33 至 55 为待定义的保留的流分组,以供将来版本或者分组协议变化之用。与其它技术相比,这是使 MDD 接口更加灵活并且面对不断变化的技术和系统设计依然十分有用的一部分。

#### [0285] 6. 用户定义的流分组

[0286] 保留称为类型 56 至 63 的 8 个数据流类型,以便用于专有的应用,这些数据流类型可以由设备制造商来定义以便与 MDDI 链路一同使用。这些分组被称为用户定义的流分组。这种分组可以用于任何目的,但主机和客户机只有在非常了解或者知道这种使用结果的情况下才应该采用这种分组。将对这些分组类型的流参数和数据的具体定义留给实现这种分组类型或者探寻其用途的具体设备制造商。用户定义的流分组的某些示例性用途为传送测试参数和测试结果、工厂校准数据和专有的特殊数据。图 15 中示出了用于一个实施例的用

户定义的流分组的格式。如图 15 所示,将这种类型的分组构造为具有分组长度(2 字节)、分组类型、bClient ID 号、流参数、参数 CRC、流数据和流数据 CRC 字段。

#### [0287] 7. 色彩映射分组

[0288] 色彩映射分组说明用于向客户机呈现色彩的色彩映射查找表的内容。某些应用可能要求比单个分组能够传输的数据量更大的色彩映射。在这种情况下,通过使用如下所述的偏移和长度字段,可以传送多个色彩映射分组,每个色彩映射分组均具有色彩映射的不同子集。图 16 示出了一个实施例中的色彩映射分组的格式。如图 16 所示,将这种类型的分组构造为具有分组长度、分组类型、hClient ID、色彩映射项目计数、色彩映射偏移、参数 CRC、色彩映射数据和数据 CRC 字段。在一个实施例中,通常将这种类型的分组标识为类型 64 分组(视频数据格式和色彩映射分组),如分组类型字段(2 字节)中所说明的那样。客户机使用客户机能力分组中的色彩映射大小和色彩映射宽度字段来表明接收色彩映射分组的能力。

#### [0289] 8. 反向链路封装分组

[0290] 在一个示例性的实施例中,使用反向链路封装分组沿反向传送数据。发送一个正向链路分组,并且大约在这个分组的中部改变 MDDI 链路操作或者让它转向(传送方向),从而能够沿反向发送分组。图 17 示出一个实施例中的反向链路封装分组的格式。如图 17 所示,将这种类型的分组构造为具有分组长度、分组类型、hClient ID、反向链路标志、反向速率除数、转向(Turn-Around)1 长度、转向 2 长度、参数 CRC、全零 1、转向 1、反向数据分组、转向 2 和全零 2。在一个实施例中,通常将这种类型的分组标识为类型 65 分组。对于外部模式来说,每一主机必须能够生成这个分组并且接收数据,每一客户机必须能够接收并且向主机发送数据。这个分组的实现方式对于内部模式来说是可选的,但是反向链路封装分组用于主机接收来自客户机的数据。

[0291] 当发送反向链路封装分组时,MDDI 链路控制器按照特定方式运行。MDD 接口具有始终由作为链路控制器的主机驱动的选通信号。主机表现得就好像它正在为反向链路封装分组的转向和反向数据分组部分的每一位传输零。所述主机在两个转向期间并且在分配给反向数据分组的期间,在每一位的边界处翻转(toggle)MDDI 选通信号。也就是说,从全零 1 字段开始到全零 2 字段结束,主机翻转 MDDI\_Stb。(这与传输全零数据的行为相同。)

[0292] 所述主机禁用其 MDDI 数据信号线路驱动器,并且通常确保在转向 1 字段的最后一位之前它们是完全禁用的,然后,在转向 2 字段期间重新启用其线路驱动器,并且通常确保在转向 2 字段的最后一位之前它们是完全启用的。所述客户机读取转向长度参数,并且在转向 1 字段的最后一位之后,立即向主机驱动数据信号。也就是说,如下文以及其它地方的分组内容描述中说明的那样,在 MDDI 选通脉冲的某些上升沿处,所述客户机把新数据按照时钟输入(clock into)链路中。客户机使用分组长度和转向长度参数来了解其可以用来向主机发送分组的时间长度。当没有数据要发送到主机时,客户机可以发送填充符分组或者把数据线驱动至零状态。如果数据线被驱动为零,那么所述主机将其解释为具有零长度(不是一个有效长度)的分组,并且在当前反向链路封装分组期间,主机不再接收来自客户机的任何分组。

[0293] 在一个实施例中,可以利用客户机请求和状态分组的反向链路请求字段通知主机所述客户机在反向链路封装分组中需要把数据送回到主机的字节数。主机试图通过在反向

链路封装分组中分配至少该字节数来允许所述请求。主机可以在子帧中发送一个以上的反向链路封装分组。客户机几乎可以在任何时候发送客户机请求和状态分组,所述主机将把反向链路请求参数解释为一个子帧中请求的字节总数。

#### [0294] 9. 客户机能力分组

[0295] 主机必须知道与之通信的客户机(显示器)的能力,以便按照通常最优或者所需要的方式来配置主机至客户机的链路。建议在获得正向链路同步之后,显示器把客户机能力分组发送给主机。当由主机使用反向链路封装分组中的反向链路标志来请求时,应该考虑这种分组的传输。客户机能力分组用来通知主机客户机的能力。对于外部模式来说,每一主机必须能够接收这个分组,而每一客户机必须能够发送这个分组以便完全地利用这一接口和协议。这个分组的实现方式对于内部模式而言是可选的,这是由于在这种情况下,如显示器、键盘或其他输入/输出设备之类的客户机的能力应在制造的时候或者装配到某些类型的单个部件或单元时,就应该已经被明确定义并且为主机所知。

[0296] 图 18 中示出了一个实施例中客户机能力分组的格式。如图 18 所示,对于本实施例,将这种类型的分组构造为具有分组长度、分组类型、保留 cClient ID,协议版本、最小协议版本、数据速率能力,接口类型能力,可选显示器数目,保留 1,位图宽度、位图高度、显示窗宽度,显示窗高度,色彩映射大小,色彩映射 RGB 宽度,RGB 能力,单色能力、保留 2, Y Cr Cb 能力、间隔(Bayer)能力,阿尔法光标图像平面,客户机特征能力、最大视频帧速率,最小视频帧速率,最小子帧速率,音频缓冲区深度,音频通道能力,音频采用速率能力,音频采样分辨率,麦克风音频采样速率,麦克风采样速率能力,键盘数据格式,指示设备数据格式,内容保护类型,制造商名称,产品代码,保留 3,序号,制造星期,制造年份,以及 CRC 字段。在一个示例性的实施例中,通常将这种类型的分组标识为类型 66 分组。

#### [0297] 10. 键盘数据分组

[0298] 键盘数据分组用来从客户机设备向主机发送键盘数据。无线(或者有线)键盘可以与各种显示器或者音频设备结合使用,包括但不限于头戴视频显示器/音频呈现设备。键盘数据分组把从几个已知的类似键盘的设备收到的键盘数据中继至主机。这个分组还可以在正向链路上使用以便向键盘发送数据。客户机使用客户机能力分组中的键盘数据字段来表明发送和接收键盘数据分组的能力。

[0299] 图 19 中示出了键盘数据分组的格式,并且包含来自或者用于键盘的可变字节数量的信息。如图 19 所示,将这种类型的分组构造为具有分组长度、分组类型、bClient ID、键盘数据格式、键盘数据和 CRC 字段。在这里,通常将这种类型的分组标识为类型 67 分组。

[0300] 所述 bClient ID 是一个保留字段,如前所述,并且对分组的所有字节执行 CRC。键盘数据格式字段包含用于描述键盘数据格式的 2 字节值。位 6 至 0 应该与客户机能力分组中的键盘数据格式字段相同。这个值不等于 127。位 15 至 7 被保留以供将来使用,因此目前将它们设置为零。

#### [0301] 11. 指示设备数据分组

[0302] 指示设备数据分组用作一种用来将来自无线鼠标或者其它指示设备的位置信息从客户机发送给主机的方法、结构或手段。还可以使用这个分组在正向链路上将数据发送给指示设备。图 20 中示出了指示设备数据分组的示例性格式,并且该分组格式包含来自或者用于指示设备的可变字节数的信息。如图 20 所示,将这种类型的分组构造为具有分组长

度、分组类型、bClient ID, 指示设备格式、指示设备数据和 CRC 字段。在一个示例性的实施例中, 通常在 1 个字节的类型字段中, 将这种类型的分组标识为类型 68 分组。

#### [0303] 12. 链路关闭分组

[0304] 将链路关闭分组从主机发送到客户机, 作为一种用于表明 MDDI 数据和选通脉冲将被关闭, 并且进入低功耗“休眠”状态的方法和手段。在将静态位图从移动通信设备发送到显示器之后, 或者当暂时没有进一步的信息需要从主机传送到客户机时, 这个分组对关闭链路并且节省功率十分有用。当主机再次发送分组时, 恢复正常操作。休眠之后发送的第一分组是子帧报头分组。图 21 中示出了客户机状态分组的格式。如图 21 所示, 将这种类型的分组构造为具有分组长度、分组类型和 CRC 和全零字段。在一个实施例中, 通常在 1 个字节的类型字段中将这种类型的分组标识为类型 69 分组, 并且使用 3 字节的预选固定长度。

[0305] 分组长度字段用 2 个字节来指明该分组中不包括分组长度字段的字节数。在一个实施例中, 该分组的分组长度依赖于发送链路关闭分组时有有效的接口类型或链路模式。因此, 典型的分组长度如下: 对于类型 1 模式, 为 20 字节 (分组中总共 22 字节); 对于类型 2 模式, 为 36 字节 (分组中总共 38 字节); 对于类型 3 模式, 为 68 字节 (分组中总共 70 字节); 对于类型 4 模式, 为 132 字节 (分组中总共 134 字节)。

[0306] 全零字段使用可变数量的字节, 以保证 MDDI\_Data 处于逻辑零电平的时间长得足以使客户机在禁用主机线路驱动器之前能够仅仅使用 MDDI\_Stb 开始恢复时钟。全零字段的长度依赖于发送链路关闭分组时有有效的接口类型或链路模式。对于任何接口类型设定, 全零字段的长度意在 MDDI\_Stb 上产生 64 个脉冲。因此, 各种接口类型的全零字段如下: 对于类型 1, 为 16 字节; 对于类型 2, 为 32 字节; 对于类型 3, 为 64 字节; 对于类型 4, 为 128 字节。

[0307] CRC 字段使用 2 个字节, 其包含从分组长度到分组类型的字节的 16 位 CRC。

[0308] 在低功率休眠状态下, 在全零字段的最后一位后第 16 至第 48 个 MDDI\_Stb 周期或脉冲后开始, 将所述 MDDI\_Data0 驱动器禁用为高阻状态。对于类型 2、类型 3 或类型 4 链路而言, 在禁用 MDDI\_Data0 驱动器的同时, MDDI\_Data1 至 MDDI\_DataPwr7 信号也被置于高阻抗状态。如其它地方讨论的那样, 主机或者客户机都可以让 MDDI 链路从休眠状态中“唤醒”, 这是本发明的关键性进步和优势。

[0309] 在全零字段的定义中已经描述过, MDDI\_Stb 在链路关闭分组的 CRC 字段的 MSB 后的 64 个周期时翻转, 以便于及时关闭客户机控制器。一个周期是从低到高的跃迁, 后面跟着从高到低的跃迁, 或从高到低的跃迁, 后面跟着从低到高的跃迁。在发送全零字段之后, 禁用主机中的 MDDI\_Stb 驱动器。

#### [0310] 13. 客户机请求和状态分组

[0311] 主机需要来自客户机的少量信息, 以便使其可以按照总体来说最优的方式来配置主机至客户机的链路。建议客户机向主机每一子帧发送一个客户机状态分组。客户机应该把这个分组作为反向链路封装分组中的第一个分组来发送, 以便确保将该分组可靠地发送至主机。当由主机使用反向链路封装分组中的反向链路标志来请求该分组时, 也可实现该分组的转送。客户机请求和状态分组可用于向主机报告错误和状态。对于外部模式操作而言, 每个主机都应该能够接收该分组, 并且每个客户机应该能够发送该分组, 以便恰当和优

化地使用该 MDD 接口协议。还建议：对于内部操作，即内部主机和内部客户机，应当支持该分组，但这不是必须的。

[0312] 图 22 中示出了客户机请求和状态分组的格式。如图 22 所示，将这种类型的分组构造为具有分组长度、分组类型、cClient ID、反向链路请求、能力变化、客户机忙、CRC 错误计数和 CRC 字段。在 1 个字节的类型字段中，通常将这种类型的分组标识为类型 70 分组，并且使用 12 个字节的预选固定长度。

[0313] 所述反向链路请求字段可以用来通知主机所述客户机在反向链路封装分组中需要的字节数目，以便其把数据发送回主机。主机应该试图通过在反向链路封装分组中分配至少这个数目的字节来允许所述请求。主机可以在子帧中发送一个以上的反向链路封装分组以便接纳数据。客户机可以在任何时候发送客户机请求和状态分组，并且所述主机可将该反向链路请求参数解释为一个子帧中请求的字节总数。下文示出如何将反向链路数据送回至主机的其它细节和具体实例。

#### [0314] 14. 位块传送分组

[0315] 位块传送分组提供了用于沿任意方向滚动显示器区域的方法。具有这一能力的显示器将在客户机能力分组的显示器特征能力指示符字段的位 0 中报告所述能力。图 23 中示出了一个实施例的位块传送分组的格式。如图 23 所示，将这种类型的分组构造为具有分组长度、分组类型、hClient ID、左上 X 值、左上 Y 值、窗口宽度、窗口高度、窗口 X 移动、窗口 Y 移动和 CRC 字段。通常将这种类型的分组标识为类型 71 分组，并且使用 15 个字节的预选固定长度。

[0316] 这些字段用来说明待移动的窗口左上角的坐标的 X 和 Y 值，待移动的窗口的宽度和高度，以及窗口将要分别水平并且垂直移动的像素数目。后两个字段的正值使得窗口将向下向右移动，而负值让窗口向上向左移动。

#### [0317] 15. 位图区域充分组

[0318] 位图区域充分组提供了很容易地把显示区域初始化为单色的手段、结构或方法。具有这一能力的显示器将在客户机能力分组的客户机特征能力指示符字段的位 1 中报告所述能力。图 24 中示出了位图区域充分组的格式的一个实施例。如图 24 所示，在这种情况下，将这种类型的分组构造为具有分组长度、分组类型、hClient ID、左上 X 值、左上 Y 值、窗口宽度、窗口高度、数据格式描述符、像素区域填充值和 CRC 字段。在 1 个字节的类型字段中，通常将这种类型的分组标识为类型 72 分组，并且使用 17 个字节的预选固定长度。

#### [0319] 16. 位图图案充分组

[0320] 位图图案充分组提供了很容易地把显示区域初始化为预选图案的手段和结构。具有这一能力的显示器将在客户机能力分组的客户机特征能力指示符字段的位 2 中报告所述能力。填充图案的左上角与待填充的窗口的左上角对准，除非水平或垂直图案偏移为非零。如果待填充的窗口比填充图案更宽或者更高，那么可以多次水平或者垂直地重复所述图案以便填充窗口。根据需要最后重复的图案的右侧或底部截断。如果窗口比填充图案小，那么可以截断填充图案的右侧或者底部以便适应所述窗口。

[0321] 如果水平图案偏移为非零，那么在窗口左边和该左边加上该水平图案偏移之间的像素用该图案最右侧的像素填充。该水平图案偏移将小于该图案的宽度。类似地，如果垂直图案偏移为非零，则在该窗口顶边和该顶边加上垂直图案偏移之间的像素用该图案的最



下侧的像素填充。该垂直图案偏移小于该图案高度。

[0322] 图 25 中示出了位图图案填充分组的格式的一个实施例。如图 25 所示,将这种类型的分组构造为具有分组长度、分组类型、hClient ID、左上 X 值、左上 Y 值、窗口宽度、窗口高度、图案宽度、图案高度、水平图案偏移、垂直图案偏移、数据格式描述符、参数 CRC、图案像素数据和像素数据 CRC 字段。在 1 个字节类型字段中,通常将这种类型的分组标识为类型 73 分组。

#### [0323] 17. 通信链路数据通道分组

[0324] 通信链路数据通道分组提供了用于使如 PDA 的这种具有高级计算能力的客户机与如蜂窝电话或者无线数据端口设备的这种无线收发器通信的结构、手段或方法。在这种情况下,所述 MDDI 链路充当通信设备和具有移动式显示器的计算设备之间的便利高速接口,其中这种分组在设备的操作系统的链路层传送数据。例如,如果网页浏览器、电子邮件客户机或者整个 PDA 被嵌入移动式显示器中,就可以使用这个分组。具有这一能力的显示器将在客户机能力分组的客户机特征能力指示符字段的位 3 中报告所述能力。

[0325] 图 26 中示出了通信链路数据通道分组的一个实施例的格式。如图 26 所示,将这种类型的分组构造为具有分组长度、分组类型、hClient ID、参数 CRC、通信链路数据和通信数据 CRC 字段。在一个实施例中,在类型字段中,通常将这种类型的分组标识为类型 74 分组。

#### [0326] 18. 接口类型切换请求分组

[0327] 接口类型切换请求分组提供了一种手段、方法或结构,其使得主机能够请求客户机或者显示器从现有或者当前模式切换至类型 1(串行)、类型 2(2 位并行)、类型 3(4 位并行)或者类型 4(8 位并行)模式。在主机请求特定模式以前,它应该通过检查客户机能力分组的显示器特征能力指示符字段的位 6 和 7,来确认客户机能够工作在所期望的模式下。图 27 中示出了接口类型切换请求分组格式的一个实施例。如图 27 所示,将这种类型的分组构造为具有分组长度、分组类型、接口类型、保留 1 和 CRC 字段。通常将这种类型的分组标识为类型 75 分组,并且使用 4 字节的预选固定长度。

#### [0328] 19. 接口类型确认分组

[0329] 接口类型确认分组由客户机发送,并提供使得客户机能够确认接收到接口类型切换请求分组的手段、方法和结构。所请求的模式,即:类型 1(串行)、类型 2(2 位并行)、类型 3(4 位并行)或者类型 4(8 位并行)模式,作为这个分组中的一个参数返回给主机。图 28 中示出了接口类型确认分组的一个实施例的格式。如图 28 所示,将这种类型的分组构造为具有分组长度、分组类型、cClient ID、接口类型、保留 1 和 CRC 字段。通常将这种类型的分组标识为类型 76 分组,并且使用 4 字节的预选固定长度。

#### [0330] 20. 执行类型切换分组

[0331] 执行类型切换分组是用于使主机命令客户机切换至这个分组中指定的模式的一种手段、结构或方法。这个模式与接口类型切换请求分组和接口类型确认分组先前请求并且确认的模式相同。在发送这个分组之后,主机和客户机应该切换到达成一致的 mode。客户机可能在模式改变期间失去并且重新获得链路同步。图 29 中示出了执行类型切换分组的一个实施例的格式。如图 29 所示,将这种类型的分组构造为具有分组长度、分组类型、分组类型、保留 1 和 CRC 字段。在 1 个字节的类型字段中,通常将这种类型的分组标识为类型

77 分组,并且使用 4 字节的预选固定长度。

#### [0332] 21. 正向音频通道启用分组

[0333] 这个分组提供使得主机能够启用或者禁用客户机中的音频通道的结构、方法或手段。这一能力十分有用,因为在主机没有音频输出时,客户机(如显示器)可以使音频放大器或者类似电路元件断电以便节省功率。仅仅使用音频流的存在与否作为指示符显然较难实现。客户机系统上电时的缺省状态是启用所有音频通道。图 30 中示出了正向音频通道启用分组的一个实施例的格式。如图 30 所示,将这种类型的分组构造为具有分组长度、分组类型、hClient ID、音频通道启用掩码和 CRC 字段。在 1 个字节的类型字段中,通常将这种类型的分组标识为类型 78 分组,并且使用 4 字节的预选固定长度。

#### [0334] 22. 反向音频采样速率分组

[0335] 这个分组使得主机能够启用或者禁用反向链路音频通道,并且设置这一流的音频数据采样速率。所述主机选择在客户机能力分组中被定义为有效的采样速率。如果主机选择无效的采样速率,那么客户机不会向主机发送音频流,并且在客户机错误报告分组中可向主机发送适当的错误、错误值、或错误信号。主机可以通过把采样速率设置为值 255 来禁用反向链路音频流。当客户机系统最初上电或者连接时假定的缺省状态为禁用反向链路音频流。图 31 中示出了反向音频采样速率分组的一个实施例的格式。如图 31 所示,将这种类型的分组构造为具有分组长度、分组类型、hClient ID、音频采样速率、保留 1 和 CRC 字段。通常将这种类型的分组标识为类型 79 分组,并且使用 4 字节的预选固定长度。

#### [0336] 23. 数字内容保护开销分组

[0337] 这个分组提供了一种使得主机和客户机能够交换与正使用的数字内容保护方法相关的消息的结构、方法或手段。目前设想了两种类型的内容保护,数字传输内容保护(DTCP)或者高带宽数字内容保护系统(HDCP),并且为将来可替代的保护方案名称保留了空间。所使用的方法由这个分组中的内容保护类型参数说明。图 32 中示出了数字内容保护开销分组的一个实施例格式。如图 32 所示,将这种类型的分组构造为具有分组长度、分组类型、bClient ID、内容保护类型、内容保护开销消息和 CRC 字段。通常将这种类型的分组标识为类型 80 分组。

#### [0338] 24. 透明色彩启用分组

[0339] 透明色彩启用分组是一种用来说明在显示器中哪些色彩是透明的,并且启用或者禁用利用透明色彩来显示图像的结构、方法或手段。具有这一能力的显示器在客户机能力分组的客户机特征能力指示符字段的位 4 中报告该能力。当具有透明色彩值的像素被写入位图时,色彩不会从先前值发生变化。图 33 中示出了透明色彩启用分组的格式。如图 33 所示,将这种类型的分组构造为具有分组长度、分组类型、hClient ID、透明色彩启用、保留 1、阿尔法光标标识符、数据格式描述符、透明像素值和 CRC 字段。在 1 个字节的类型字段中,通常将这种类型的分组标识为类型 81 分组,并且使用 10 个字节的预选固定长度。

#### [0340] 25. 往返行程延迟测量分组

[0341] 往返行程延迟测量分组提供了一种用来测量从主机到客户机(显示器)的传播延迟加上从客户机(显示器)回到主机的延迟的结构、方法或手段。这一测量固有地包括线路驱动器和接收器以及互联子系统中存在的延迟。这一测量用来设置上文大体描述的反向链路封装分组中的转向延迟和反向链路速率除数参数。当 MDDI 链路以用于特定应用的最

大速度运行时,这种分组是最有用的。可以在类型 1 模式下以较低数据速率发送分组,从而增加往返形成延迟测量的范围。所述 MDDI\_Stb 信号的行为是如同在如下字段发送全零数据:两个保护时间、全零和测量周期。这样做使得 MDDI\_Stb 以一半的数据速率翻转,因此可以在测量周期中在客户机里将 MDDI\_Stb 用作周期性的时钟。

[0342] 在一个实施例中,客户机通常通过客户机能力分组的客户机特征能力指示符字段中的位 18 来指示支持往返行程延迟测量分组的能力。建议所有客户机都支持往返行程延迟测量,但主机也可能基于最大电缆延迟和最大驱动器和接收器延迟来获知最差情况下的往返行程延迟。主机还可在内部模式下预先获 MDDI 链路的往返行程延迟,这是因为这是使用该接口的设备的已知设计要素(导线长度、电路类型和特性,等等)中的一个方面。

[0343] 图 34 中示出了往返行程延迟测量分组的格式。如图 34 所示,在一个实施例中,将这种类型的分组构造为具有分组长度、分组类型、hClient ID、参数 CRC、保护时间 1、测量周期、全零和保护时间 2 字段。通常将这种类型的分组标识为类型 82 分组,并且使用 159 位的预选固定长度。

[0344] 图 35 中示出了在往返行程延迟测量分组期间发生的事件的时序。在图 35 中,主机发送往返行程延迟测量分组,这通过其后跟随有全零和保护时间 1 字段的参数 CRC 和选通对准字段的出现来示出。在所述分组到达客户机显示器设备或者处理电路以前,延迟 3502 发生。当客户机收到所述分组时,客户机在尽可能准确到实际的测量周期的开始时刻传输 0xff、0xff、和 30 个字节的 0x0 模式(pattern),其中该测量周期开始是由客户机确定的。客户机开始发送这一序列的实际时间距从主机的角度看的测量周期的开始有延迟。这一延迟量基本上是所述分组用来通过线路驱动器和接收器以及互联子系统(电缆、导线)进行传送所花费的时间。对于把该模式从客户机传送到主机来说,经历了类似的延迟量 3504。

[0345] 为了准确地确定往返于客户机的信号的往返行程延迟时间,主机对测量周期开始之后出现的正向链路位时间周期的数目进行计数,直到检测到 0xff、0xff、和 30 个字节的 0x0 序列的开头到达为止。这一信息用来确定往返信号从主机传递至客户机并且再返回的时间量。然后,将这一数量的一半左右归因于到达客户机的单向信号通道所产生的延迟。

[0346] 在两个保护时间期间,主机和客户机都将线路驱动到逻辑零电平,以保持 MDDI\_DATA 线路处于所定义的状态。在两个保护时间期间,主机和客户机的启用和禁用时间使得 MDDI\_Data 信号对于任何有效的往返行程延迟时间都处于有效低电平。

[0347] 26. 正向链路偏离校准分组

[0348] 正向链路偏离校准分组允许客户机或者显示器根据所述 MDDI\_Stb 信号来自己校准 MDDI\_Data 信号的传播延迟差异。在没有延迟偏离补偿的情况下,最大数据速率通常是受限的以解决这些延迟中潜在的最坏情况下的变化。通常只有当正向链路数据速率被配置成 50Mbps 左右或者更低的速率时,才发送这个分组。在发送这个分组以便校准显示器之后,数据速率可以逐级提高至 50Mbps 以上。如果在偏离校准过程期间数据速率被设置得过高,那么显示器有可能同步到一个该位周期的假信号(alias)上,这可以令延迟偏离补偿被设置为偏离一个以上的位时间,从而导致数据时钟错误。在发送正向链路偏离校准分组之前,选择接口的最高数据速率类型或者最大可能性的接口类型,从而校准所有存在的数据位。

[0349] 图 56 中示出了正向链路偏离校准分组格式的一个实施例。如图 56 所示,将这种

类型的分组构造为具有分组长度 (2 字节)、分组类型、hClient ID、参数 CRC、全零、校准数据序列和 CRC 字段。在类型字段中,通常将这种类型的分组标识为类型 82 分组,并且在—个实施例中具有 515 的预选长度。

#### [0350] 虚拟控制面板

[0351] 虚拟控制面板 (VCP) 的使用允许主机设置客户机中的某些用户控制。通过允许由主机来调节这些参数,客户机中的用户界面得以简化,这是因为可以通过主机软件而不是通过客户机中的一个或多个微处理器来生成允许用户调节诸如音量或显示亮度之类的参数的屏幕画面。主机具有读取客户机中的参数设置,并且确定每一控制的有效值范围的能力。客户机具有向主机返回其所能调节的控制参数的报告的能力。

[0352] 通常指定的控制代码 (VCP 代码) 和相关联的数据值被用于说明客户机中的控制和设置。MDDI 规范中的 VCP 代码被扩展为 16 位,以便在分组定义中保持正确的数据字段对准,并且将来支持补充值,所述补充值对于这一接口或者将来的增强是唯一的。

#### [0353] 27. 请求 VCP 特征分组

[0354] 请求 VCP 特征分组提供了一种使主机请求具体控制参数或者所有有效控制参数的当前设置的手段、机制或方法。通常,客户机利用 VCP 特征应答分组中适当的信息来响应 VCP 分组。在一个实施例中,客户机使用客户机能力分组中的客户机特征能力指示符字段的位 20 来表明支持请求 VCP 特征分组的能力。

[0355] 图 69 中示出了一个实施例中请求 VCP 特征分组的格式。如图 69 所示,将这种类型的分组构造为具有分组长度、分组类型、hClientID、MCCS VCP 代码和 CRC 字段。在一个实施例中,通常将这种类型的分组标识为类型 128,这是在 2 字节的类型字段中表明的。分组长度说明分组中不包括分组长度字段的字节总数,并且对于这种类型的分组而言,该分组长度通常被固定在 8 字节的长度。

[0356] 所述 hClient ID 字段被保留以供未来实现中客户机 ID 使用并通常设置为零。所述 MCCS VCP 代码字段包括 2 字节的信息,用于说明 MCCS VCP 控制代码参数。0 至 255 的范围内的值令 VCP 特征应答分组以单个项返回,该单个表项为在对应于指定的 MCCS 代码的 VCP 特征应答列表中的表项。为 65535 (0xffff) 的 MCCS VCP 代码用于请求一个具有 VCP 特征应答列表的 VCP 特征应答分组,所述 VCP 特征应答列表包含由客户机支持的每个控制的特征应答列表项。对于这个字段而言,256 至 65534 的值被保留以供将来使用并且目前没有使用。

#### [0357] 28. VCP 特征应答分组

[0358] VCP 特征应答分组提供了一种使客户机利用具体控制参数或者所有有效控制参数的当前设置来响应主机请求的手段、机制或方法。总体而言,客户机响应于请求 VCP 特征分组而发送所述 VCP 特征应答分组。这个分组对确定具体参数的当前设置,确定具体控制的有效范围,确定客户机是否支持具体的控制或者确定由客户机支持的控制集合十分有用。如果发送涉及客户机中没有实现的具体控制的请求 VCP 特征,那么利用与包含适当错误代码的未实现控制相对应的单个 VCP 特征应答列表项来返回 VCP 特征应答分组。在一个实施例中,所述客户机使用客户机能力分组的客户机特征能力指示符字段的位 20 来表明支持 VCP 特征应答分组的能力。

[0359] 图 70 中示出了一个实施例中的 VCP 特征应答分组的格式。如图 70 所示,将这种

类型的分组构造为具有分组长度、分组类型、cClientID、MCCS 版本、应答序号、VCP 特征应答列表和 CRC 字段。在一个实施例中,通常将这种类型的分组标识为类型 129,如在 2 字节的类型字段中表明的那样。

[0360] 所述 cClient ID 字段包含保留给客户机 ID 的信息。这个字段被保留以供将来使用并且通常设置为零。MCCS 版本字段包含 2 字节的信息,用于说明由客户机实现的 VESAMCCS 规范的版本。

[0361] 2 字节的应答序号字段包含说明由客户机返回的 VCP 特征应答分组的序号的信息或者数据。所述客户机,响应于具有值为 65535 的 MCCS 控制代码的请求 VCP 特征分组,返回一个或多个 VCP 特征应答分组。所述客户机可以经由多个 VCP 特征应答分组扩展特征应答列表。在这种情况下,客户机向每一连续分组分配序号,并且响应单个请求 VCP 特征分组而发送的 VCP 特征应答分组的序号是从零开始并且每次加 1。最后的 VCP 特征应答分组中的最后一个 VCP 特征列表项应该包含等于 0xffff 的 MCCS VCP 控制代码值,以便确定所述分组是最后一个并且包含返回的分组的最高序号。如果响应于请求 VCP 特征分组只发送了一个 VCP 特征应答分组,那么该单个分组中的应答序号就为零,并且 VCP 特征应答列表包含具有等于 0xffff 的 MCCS VCP 控制代码的记录。

[0362] 列表中特征数目字段包含 2 字节,用于说明该分组的 VCP 特征应答列表中的 VCP 特征列表项的数目,而所述 VCP 特征应答列表字段则是包含一个或多个 VCP 特征应答列表项的一组字节。图 71 中示出了一个实施例中单个 VCP 特征应答列表项的格式。

[0363] 如图 71 所示,每一 VCP 特征应答列表项的长度是 12 字节,并且包括 MCCS VCP 代码、结果代码、最大值和当前值字段。2 字节的 MCCS VCP 代码字段包含说明与这一列表项相关联的 MCCS VCP 控制代码参数的数据或者信息。对于本实施例,只是在 VESA MCCS 规范版本 2 和后来版本中定义的控制代码值被认为是有效的。2 字节的结果代码字段包含用于说明错误代码的信息,所述错误代码与关于所指定的 MCCS VCP 控制的信息请求相关。这个字段中的值 '0' 意味着没有错误,而值 '1' 则意味着指定的控制未被客户机中实现。这个字段的其它值 2 至 65535 目前被保留以便将来使用以及用于实现所述领域设想的其它应用,但是现在不使用。

[0364] 4 字节的最大值字段包含 32 位无符号整数,用于说明可以设置为指定的 MCCS 控制的最大可能值。如果所请求的控制没有被客户机中实现,就将这一值设置为零。如果返回值在长度上小于 32 位 (4 字节),就把所述值换算为 32 位整数,将最高有效 (未使用) 字节设置为零。4 字节的当前值字段包含用于说明指定的 MCCS VCP 连续 (C) 或者不连续 (NC) 控制的当前值的信息。如果所请求的控制没有被客户机中实现,或者实现了所述控制但是是表 (T) 数据类型,就将这一值设置为零。如果就每一 VESAMCCS 规范而言,返回值在长度上小于 32 位 (4 字节),就把所述值换算为 32 位整数,将最高有效 (未使用) 字节设置为零。

[0365] 29. 设置 VCP 特征分组

[0366] 设置 VCP 特征分组提供用于使主机为客户机中连续和不连续的控制设置 VCP 控制值的手段、机制或者方法。在一个实施例中,所述客户机使用客户机能力分组的客户机特征能力指示符字段的位 13 来表明支持设置 VCP 特征分组的能力。

[0367] 图 72 中示出了一个实施例中设置 VCP 特征分组的格式。如图 72 所示,将这种类型的分组构造为具有分组长度、分组类型、hClientID、MCCS VCP 代码、列表中值的数量、控

制值列表和 CRC 字段。通常将这种类型的分组标识为类型 130, 如 2 字节的类型字段中所示那样, 且该类型的分组不包括分组长度字段是 20 字节长。

[0368] 所述 hClient ID 字段再次使用 2 字节值来说明或者充当客户机 ID。将这个字段保留下来以供将来使用并且目前将其设置为零。所述 MCCS VCP 代码字段使用 2 字节的信息或者值来说明要调节的 MCCS VCP 控制代码参数。所述 2 字节的列表中值的数量字段包含用于说明控制值列表中存在的 16 位值的数目的信息或者值。所述控制值列表通常包含一项, 除非 MCCS 控制代码涉及客户机中的表。在不涉及表的控制的情况下, 所述控制值列表将包含一个值, 该值用于说明将要写入由 MCCS VCP 代码字段指定的控制参数的新值。对于涉及表的控制来说, 控制值列表中数据的格式是由指定的 MCCS VCP 代码的参数描述来指定的。如果所述列表包含大于一个字节的值, 那么首先传输最低有效字节, 这与其它地方定义的方法相一致。最后, 2 字节的 CRC 字段包含分组中所有字节的 16 位 CRC, 其中包括分组长度在内。

### [0369] 30. 请求有效参数分组

[0370] 请求有效参数分组用作请求该客户机返回有效参数应答分组的手段或者机制, 该有效参数应答分组包含由所指定的不连续 (NC) 或者表 (T) 控制支持的一参数列表。这个分组只应当指定不连续的控制或者涉及客户机中的表的控制, 而不指定用于指定所有控制的值为 65535 (0xffff) 的 MCCS VCP 代码值。如果指定了不支持或者无效的 MCCS VCP 代码, 就在有效参数应答分组中返回适当的错误值。在一个实施例中, 所述客户机使用客户机能力分组的客户机特征能力指示符字段的位 20 来表明支持请求有效参数分组的能力。

[0371] 图 73 中示出了一个实施例中的请求有效参数分组的格式。如图 73 所示, 将这种类型的分组构造为具有分组长度、分组类型、hClient ID、MCCS VCP 代码和 CRC 字段。在一个实施例中, 通常将这种类型的分组标识为类型 131, 如在 2 字节的类型字段中表明的那样。

[0372] 2 字节分组长度字段中表明的分组长度通常被设置为具有分组中的字节总数, 但不包括分组长度字段的 8 个字节。所述 hClient ID 再一次指定客户机 ID, 但是正如对于本领域普通技术人员显而易见的那样, 目前被保留下来以供将来之用, 并且被设置为零。2 字节 MCCS VCP 代码字段包含用于说明待查询的不连续的 MCCS VCP 控制代码参数的值。这个字段中的值应该对应于客户机中实现的不连续的控制。值 256 至 65535 (0xffff) 通常被保留下来或者被认为是无效的, 并且被认为是错误响应中未实现的控制。

### [0373] 31. 有效参数应答分组

[0374] 有效参数应答分组是响应于请求有效参数分组而发送的。它用作识别不连续的 MCCS VCP 控制或者用于返回表的内容的控制的有效设置的手段、方法或机制。如果所述控制涉及客户机中的表, 那么 VCP 参数应答列表仅仅包含被请求的顺序表值的具体列表。如果所述表的内容无法正好放进单个有效参数应答分组, 就可以由客户机发送具有顺序的应答序号的多个分组。在一个实施例中, 客户机使用客户机能力分组的客户机特征能力指示符字段的位 20 来表明支持有效参数应答分组的能力。

[0375] 主机可以按照以下方式请求表的内容: 主机发送一个包含必要或者需要的参数的设置 VCP 特征分组, 如读/写参数、LUT 偏移和 RGB 选择; 然后由主机发送用于说明所需控制的请求有效参数分组; 然后客户机返回包含该表数据的一个或多个有效参数应答分组。这一操作序列执行与 MCCS 操作模型中描述的表读取功能相似的功能。

[0376] 如果客户机不支持具体的客户机参数,那么在一个实施例中,这个分组的相应字段将包含值 255。对于客户机中使用的参数来说,相应的字段应该包含客户机中的参数值。

[0377] 图 74 中示出了用于一个实施例的有效参数应答分组的格式。如图 74 所示,将这种类型的分组构造为具有分组长度、分组类型、cClient ID、MCCS VCP 代码、响应代码、应答序号、列表中值的数量、VCP 参数应答列表和 CRC 字段。对于一个实施例而言,通常将这种类型的分组标识为类型 132,如在 2 字节的类型字段中表明的那样。

[0378] 所述 cClient ID 字段被保留给将来的客户机 ID,正如根据上文讨论可以获知的那样,而 3 字节的 MCCS VCP 代码分组则包含用于说明由这个分组描述的不连续 MCCS VCP 控制代码参数的值。如果请求有效参数分组指定了无效的 MCCS VCP 控制代码,那么用响应代码字段中的适当值来指定这个字段中同样的无效参数值。如果所述 MCCS 控制代码是无效的,那么 VCP 参数应答列表将具有零长度。

[0379] 所述响应代码字段包含 2 字节的信息或者值,它说明与请求相关的响应的属性,所述请求为关于所指定的 MCCS VCP 控制的信息的请求。如果这个字段中的值等于 0,那么认为对于该数据类型而言不存在错误,并且发送了序列中的最后的有效参数应答分组,该分组具有最高应答序号。如果这个字段中的值等于 1,那么认为不存在错误,但是将要发送具有更高序号的其它有效参数应答分组。如果这个字段中的值等于 2,那么认为所指定的控制在客户机中没有实现。如果这个字段中的值等于 3,那么所指定的控制不是不连续控制(它是一个连续控制且始终具有从零到其最大值的所有值的有效集合)。等于 4 至 65535 的这个字段的值被保留下来以供将来使用并且通常不使用。

[0380] 2 字节的应答序号字段指定由客户机返回的有效参数应答分组的序号。所述客户机,响应于一个请求有效参数分组,返回一个或多个有效参数应答分组。所述客户机可以将所述 VCP 参数应答列表扩展到整个多个有效参数应答分组中。在后一情况下,客户机将向每一连续的分组合分配序号,并且在除序列中最后一个分组之外的所有分组中,把响应代码设置为 1。序列中最后的有效参数应答分组将具有最高的应答序号,并且所述响应代码包含的值为 0。

[0381] 2 字节的列表中值的数量字段说明所述 VCP 参数应答列表中存在的 16 位值的数目。如果所述响应代码不等于零,那么列表中值的数量的参数是零。所述 VCP 参数应答列表字段包含一系列 0 至 32760 的 2 字节值,用于表明由 MCCS 控制代码字段指定的不连续控制的有效值集合。在 VESA MCCS 规范中给出了不连续控制代码的定义。最后,在这个实施例中,所述 CRC 字段包含分组中所有字节的 16 位 CRC,其中包括分组长度在内。

[0382] 可缩放视频流图像

[0383] 所述 MDD 接口或者协议机制、结构、手段或者方法对可缩放视频流图像提供支持,它允许主机向客户机发送与原始图像相比按比例放大或缩小的图像,并且所述可缩放图像(the scaled image)被复制到主图像缓冲器。对可缩放视频流(scaled video stream)的功能以及相关协议支持的概述在其他地方给出。支持可缩放视频流的能力被定义在响应于请求特定状态分组而发送的可缩放视频流能力分组内或者由其定义。

[0384] 32. 可缩放视频流能力分组

[0385] 可缩放视频流能力分组定义了客户机中的或者由其使用的可缩放视频流源图像的特性。图 75 中总体上示出了可缩放视频流能力分组的格式。如图 75 所示,在一个实施

例中,可缩放视频流能力分组被构造为具有分组长度、分组类型、cClient ID、最大流数、源最大 X 大小、源最大 Y 大小、RGB 能力、单色能力、保留 1、Y Cr Cb 能力、保留 2 和 CRC 字段。在一个实施例中,分组长度被选择为固定的 20 字节,如在长度字段中所示,其包括 2 字节的 cClient ID 字段和 CRC 字段,其中 cClient ID 字段被保留以供客户机 ID 之用,否则被设置为零。在一个实施例中,所述客户机使用有效状态应答列表分组中有效参数应答列表内的参数值 143 来表明其支持可缩放视频流能力分组的能力。

[0386] 2 字节的最大流数字段包含用于标识可以一次分配的同步可缩放视频流的最大数目的值。在一个实施例中,如果可缩放视频流的最大数目已被分配,那么客户机应该拒绝分配可缩放视频流的请求。如果分配了小于最大数目的可缩放视频流,那么客户机还可以基于客户机中的其它资源限制来拒绝分配请求。

[0387] 源最大 X 大小以及 Y 大小字段(2 字节)分别指定以像素数量表示可缩放视频流源图像的最大宽度和高度的值。

[0388] RGB 能力字段使用一些值来指定可以按照 RGB 格式显示的分辨率的位数。如果可缩放视频流无法使用 RGB 格式,那么这一值被设置为等于零。所述 RGB 能力字包括三个独立的无符号值:位 3 至 0 定义每一像素中蓝色(蓝色强度)的最大位数,位 7 至 4 定义每一像素中绿色(绿色强度)的最大位数,位 11 至 8 定义每一像素中红色(红色强度)的最大位数,而位 15 至 12 被保留下来以供今后用于将来的能力定义,通常设置为零。

[0389] 1 字节的单色能力字段包含用来说明可以按照单色格式显示的分辨率的位数的值。如果可缩放视频流无法使用单色格式,那么这一值是零。位 7 至 4 被保留以供将来使用,并且因此对于当前应用而言应该设置为零('0'),当然该值可以随时间而改变,如同本领域技术人员将理解的那样。位 3 至 0 定义每一像素中可以存在的灰度级的最大位数。这四位能够说明每一像素包括 1 至 15 位。如果所述值是零,那么可缩放视频流不支持所述单色格式。

[0390] 保留 1 字段(在这里是 1 字节)被保留以供将来用于提供与可缩放视频流分组信息或者数据相关的值。因此,目前,这个字段中的所有位都被设置为逻辑'0'。这个字段的一个目的在于让所有后续 2 字节字段与 16 位地址对准,让 4 字节字段与 32 位地址对准。

[0391] 2 字节的 Y Cb Cr 能力字段包含用于说明可以按照 Y Cb Cr 格式显示的分辨率的位数的值。如果可缩放视频流无法使用 Y Cb Cr 格式,那么这一值是零。所述 Y Cb Cr 能力字包括三个独立的无符号值:位 3 至 0 定义了用于说明 Cr 采样的最大位数;位 7 至 4 定义了用于说明 Cb 采样的最大位数;位 11 至 8 定义了用于说明 Y 采样的最大位数;位 15 至 12 被保留以供将来使用并且通常设置为零。

[0392] 1 字节的能力位字段包含 8 位无符号整数,它包含一组用于说明与可缩放视频流相关联的能力的标志。所述标志定义如下:位 0 覆盖了可缩放视频流分组中的像素数据能够为打包格式的情况。在先前图 12 中示出了打包的和字节对准的像素数据的实例。位 1 被保留以供将来使用并且被设置为零;位 2 也被保留供将来使用并且设置为零;位 3 覆盖了可缩放视频流能够按照色彩映射数据格式指定的情况。用于可缩放视频流的色彩映射表与用于主要图像缓冲器和所述阿尔法光标图像平面的色彩映射表相同。使用其它地方描述的色彩映射分组可配置所述色彩映射;并且位 7 至 4 被保留以供将来使用并且通常设置为



零。

[0393] 保留 2 字节（在这里是 1 字节）被保留以供将来用于提供与可缩放视频流分组信息或者数据相关的值。因此，目前，这个字段中的所有位都被设置为逻辑‘0’。这个字段的一个目的在于让所有后续 2 字节字段与 16 位地址对准，让 4 字节字段与 32 位地址对准。

### [0394] 33. 可缩放视频流建立分组

[0395] 可缩放视频流建立分组用来定义可缩放视频流的参数，并且客户机使用所述信息来分配内部存储器以便缓存并且缩放所述图像。通过利用发送其中 X 图像大小和 Y 图像大小字段为零的分组，可以解除对流的分配。解除分配的可缩放视频流稍后可以利用相同的或者不同的流参数来重新分配。在一个实施例中，客户机使用有效状态应答列表分组的有效参数应答列表中的参数值 143，并且通过使用可缩放视频流能力分组的最大流数字段中的非零值，来表明其支持可缩放视频流建立分组的能力。

[0396] 图 76 总体示出了可缩放视频流建立分组的格式。如图 76 所示，在一个实施例中，可缩放视频流建立分组被构造为具有分组长度、分组类型、hClient ID、流 ID、可视数据格式描述符、像素数据属性、X 左沿、Y 顶沿、X 右沿、Y 底沿、X 图像大小、Y 图像大小和 CRC 字段。

[0397] 2 字节的分组长度字段说明了分组中不包括分组长度字段的字节总数。在一个实施例中，这个分组的分组长度固定为 24。2 字节的分组类型字段使用值 136 将所述分组标识为可缩放视频流设置分组。2 字节的 hClient ID 字段被保留以供将来使用，并且暂时通常以逻辑零值设置所有位，或者知道协议用户确定了将要使用的 ID 值，正如将会了解的情况。

[0398] 流 ID 字段使用 2 字节说明流 ID 的唯一标识符。这一值是由主机分配的，并且在从零到客户机能力分组中指定的最大流 ID 值的范围内变化。所述主机必须小心管理流 ID 值的使用，以便确保每一激活的流被分配有唯一的值，并且确保不再激活的流被解除分配或者重新分配。

[0399] 在一个实施例中，视频数据格式描述符使用 2 字节说明当前分组的当前流中像素数据的每一像素的格式。像素数据格式必须符合在阿尔法光标图像能力分组中所定义的阿尔法光标图像平面的至少一个有效格式。所述视频数据格式描述符只定义当前分组的像素格式，而不暗含在特定视频流的使用期间将会持续使用不变的格式。图 11 示出了视频数据格式描述符是如何编码的一个实施例，以及以上针对其他分组的讨论。

[0400] 在一个实施例中，2 字节的像素数据属性字段具有的值可解释如下：位 1 和 0 选择像素数据将要被路由至的显示器。对于比特值 11 或者 00，把像素数据显示给两个眼睛，对于比特值 10，只把像素数据路由到左眼，对于比特值 01，只把数据路由到右眼。

[0401] 位 2 表明像素数据处于隔行格式。当位 2 是 0 时，像素数据处于标准逐行格式。当从一行前进到下一行时，行号（像素 Y 坐标）加 1。当位 2 是 1 时像素数据处于隔行格式。当从一行前进到下一行时，行号（像素 Y 坐标）加 2。

[0402] 位 3 表明像素数据处于交替像素格式。这与位 2 允许的标准隔行模式相似，但是这里的隔行是垂直的而不是水平的。当位 3 是 0- 时像素数据处于标准逐行像素格式。当接收到每一连续像素时，列号（像素 X 坐标）加 1。位 3 是 1- 时，像素数据处于交替像素格

式。当接收到每一像素时,列号(像素 X 坐标)加 2。

[0403] 位 4 表明像素数据涉及显示器还是照相机。当位 4 是 0- 时,像素数据被送入显示帧缓冲器或者从中送出。当位 4 是 1- 时,像素数据被送入照相机或者从中送出。位 5 被保留供将来使用并且由此通常设置为零。

[0404] 位 7 和 6 是显示器更新位,它用于说明将要写入像素数据的帧缓冲器。帧更新位的效果在其他地方作了更加详细的描述。当位 [7:6] 为 01 时,像素数据被写入离线图像缓冲器。当位 [7:6] 为 00 时,像素数据被写入用于刷新显示器的图像缓冲器。当位 [7:6] 为 11 时,像素数据被写入所有图像缓冲器。如果位 [7:6] 为 10,则视作无效值。这些位当前被保留供将来之用。在这种情况下,像素数据被忽略并且不写入任何图像缓冲器。

[0405] 位 8 至 15 被保留供将来使用,并且通常设置为逻辑零电平或值。

[0406] 34. 可缩放视频流确认分组

[0407] 可缩放视频流确认分组允许客户机确认接收到可缩放视频流设置分组。所述客户机经由有效状态应答列表分组的有效参数应答列表中的参数值 143 并且经由可缩放视频流能力分组的最大流数字段中的非零值来表明其支持可缩放视频流确认分组的能力。

[0408] 图 77 示出了可缩放视频流确认分组的格式。如图 77 所示,在一个实施例中,将可缩放视频流确认分组构造成具有分组长度分组、分组类型、cClient ID、流 ID、ACK 代码和 CRC 字段。2 字节的分组长度说明了分组中不包括分组长度字段的字节总数,且对于这个分组类型而言,分组长度值是 10,而分组类型 137 将该分组标识为可缩放视频流确认分组。

[0409] 2 字节的 cClient ID 字段 - 被保留供将来使用,并且通常设置为零。2 字节的流 ID 字段说明了流 ID 的唯一标识符。这与可缩放视频流设置分组中由主机分配的值相同。

[0410] 2 字节的确认代码字段提供了包含一个代码的值,该代码用于描述试图更新指定的可缩放视频流的结果。在一个实施例中,所述代码按如下方式定义:

[0411] 0- 流分配尝试成功。

[0412] 1- 流解除分配尝试成功。

[0413] 2- 对一个已经分配了的流 ID 的分配尝试无效。

[0414] 3- 对一个已经解除分配的流 ID 的解除分配尝试无效。

[0415] 4- 客户机不支持可缩放视频流

[0416] 5- 流参数与客户机能力不一致。

[0417] 6- 流 ID 值大于客户机允许的最大值。

[0418] 7- 客户机没有足够资源可以用来分配指定的流。

[0419] 2 字节的 CRC 字段包含了该分组中包括分组长度在内的所有字节的 CRC。

[0420] 35. 可缩放视频流分组

[0421] 可缩放视频流分组用来传输与特定的可缩放视频流相关联的像素数据。这个分组涉及的区域大小是由可缩放视频流设置分组定义的。所述客户机经由有效状态应答列表分组的有效参数应答列表中的参数值 143,并且经由可缩放视频流确认分组的确认代码字段中的成功的可缩放视频流分配响应,来表明其支持可缩放视频流分组的能力。

[0422] 图 78 示出了可缩放视频流分组的一个实施例的格式。如图 78 所示,将可缩放视频流分组构造为具有分组长度、分组类型、hClient ID、流 ID、参数 CRC、像素计数、像素数据和像素数据 CRC 字段。2 字节的分组类型字段使用值 18 将该分组标识为可缩放视频流分

组。hClient ID 字段是为客户机 ID 保留的,并且通常设置为零。如前所述,2 字节的流 ID 字段说明了流 ID 的唯一标识符。这一值是在可缩放视频流设置分组中由主机指定的,并且在可缩放视频流确认分组中被确认。

[0423] 2 字节的像素计数字段说明了像素数据字段中像素数目。2 字节的参数 CRC 字段具有从分组长度到像素计数的所有字节的 CRC。如果这一 CRC 未能通过校验,就丢弃整个分组。2 字节的像素数据字段包含将要缩放并随后被显示的原始视频信息。数据是按照由视频数据格式描述符字段所述的方式来进行格式化的。按照前面定义的方式每次传输一行所述数据。

[0424] 2 字节的像素数据 CRC 字段包含只对像素数据的 CRC。如果这一 CRC 未能通过校验,那么所述像素数据仍可被使用,不过 CRC 错误计数增加。

[0425] 36. 请求特定状态分组

[0426] 请求特定状态分组为主机提供了用于请求客户机按照这个分组中指定的方式把能力或者状态分组送回到主机的手段、机制或方法。客户机在下一个反向链路封装分组中返回指定类型的分组。如果所述客户机具有响应该请求特定状态分组的能力,那么客户机将设置分组客户机能力分组中的客户机特征能力字段中的位 17。客户机可使用分组客户机能力分组的客户机特征能力字段的位 21 来表明其支持请求特定状态分组的能力。

[0427] 图 79 示出了请求特定状态分组的一个实施例的格式。分组如图 79 所示,将请求特定状态分组构造为具有分组长度、分组类型、hClient ID、状态分组 ID 和 CRC 字段。分组长度字段说明了分组中不包括分组长度字段的字节总数,并且对于这个分组类型而言,分组长度通常固定为值 10。分组类型 138 分组把所述分组标识为请求特定状态分组。hClient ID(2 字节)字段被保留以供将来客户机 ID 使用,并且现在设置为零,而 2 字节的状态分组 ID 字段说明了客户机将要发送到主机的能力或者状态分组的类型。典型的分组类型为:

[0428] 66- 由客户机发送分组客户机能力分组。

[0429] 133- 由客户机发送阿尔法光标图像能力分组。

[0430] 139- 发送有效状态应答列表分组,用于标识客户机能够发送的能力和状态分组的准确类型。

[0431] 140- 由客户机发送分组处理延迟参数分组。

[0432] 141- 由客户机发送个人分组客户机能力分组。

[0433] 142- 由客户机发送客户机错误报告分组。

[0434] 143- 由客户机发送可缩放视频流能力分组。

[0435] 144- 由客户机发送客户机标识分组。

[0436] 分组类型 56 至 63 可以用于特定于制造商能力和状态标识符。

[0437] CRC 字段包含包括分组长度在内的分组中所有字节的 CRC。

[0438] 37. 有效状态应答列表分组

[0439] 有效状态应答列表分组提供了使得主机具有客户机有能力对其做出响应的一系列状态和能力分组的结构、手段或方法。客户机能够使用分组客户机能力分组中的客户机特征能力字段的位 21 来表明其支持有效状态应答列表分组的能力。

[0440] 图 80 示出了有效状态应答列表分组的一个实施例的格式。如图 80 所示,将有效状态应答列表分组构造为具有分组长度、分组类型、cClient ID、列表中值的数量、有效参

数应答列表和 CRC 字段。这一类型的分组的分组长度通常固定为数值 10, 并且类型值 139 将该分组标识为有效状态应答分组。cClient ID 字段被保留供将来用作客户机 ID, 并通常设置为零。2 字节的列表中值的数量字段说明了随后的有效参数应答列表中项目的数目。

[0441] 有效参数应答列表字段包含 2 字节参数的列表, 其用于说明客户机可以发送到主机的能力或者状态分组的类型。如果客户机已经表明它可以响应请求特定状态分组 (利用分组客户机能力分组中的客户机特征能力字段的位 21), 那么它能够发送至少分组客户机能力分组 (分组类型 = 66) 和有效状态应答列表分组 (分组类型 = 139)。可由客户机发送并且可包括在该列表中的分组类型以及本实施例中其各自分配目的是:

[0442] 66- 分组客户机能力分组。

[0443] 133- 阿尔法光标图像能力分组。

[0444] 139- 有效状态应答列表分组, 用于标识客户机能够发送的能力和状态分组的准确类型。

[0445] 140- 分组处理延迟参数分组。

[0446] 141- 个人显示器能力分组。

[0447] 142- 客户机错误报告分组。

[0448] 143- 可缩放视频流能力分组。

[0449] 144- 客户机标识分组。

[0450] 145- 其他显示器能力分组。

[0451] 分组类型 56 至 63 可用于特定于制造商能力和状态标识符。

[0452] CRC 字段包含分组包括分组长度在内分组中所有字节的 CRC。

[0453] 38. 分组处理延迟参数分组

[0454] 分组处理延迟参数分组提供一组参数来允许主机计算完成与特定分组类型的接收相关联的处理需要的时间。由主机发送的某些命令无法由客户机在零时间内完成。主机可以轮询客户机请求和状态分组中的状态位, 以便确定某些功能是否已经由客户机完成, 或者主机可以使用由客户机在分组处理延迟参数分组中返回的参数来计算完成时间。所述客户机能够利用有效状态应答列表分组中有效参数应答列表中的参数值 140 来表明其支持分组处理延迟参数分组的能力。

[0455] 图 81A 总体示出了分组处理延迟参数分组的一个实施例的格式。如图 81A 所示, 将分组处理延迟参数分组构造为具有分组长度、分组类型、cClient ID、列表项数、延迟参数应答列表和 CRC 字段。这个分组类型的分组长度通常固定为数值 10, 并且类型值 140 把所述分组标识为分组处理延迟参数分组。cClient ID 字段被保留以供将来用作客户机 ID, 并且通常设置为零。2 字节的列表项数说明了随后有效参数应答列表中项目的数目。

[0456] 延迟参数应答列表字段是一个或多个延迟参数列表项的列表。图 81B 示出了单个延迟参数列表项的一个实施例的格式, 其中示出了延迟分组类型、像素延迟、水平像素延迟、垂直像素延迟和固定延迟字段。

[0457] 每一延迟参数列表项在长度上通常限定为 6 字节, 并且按如下定义。2 字节的延迟分组类型字段说明了随后延迟参数应用的分组类型。像素延迟字段 (1 字节) 包括延迟值索引 (index)。将从表中读取的值乘以该分组的目的字段中的像素总数。该像素总数是由该分组定位的位图的目的区域的宽度乘高度。1 字节的水平像素延迟字段包含作为延迟值

表（与 DPVL 相同的表）的索引的值。将从表中读取的这个值乘以分组中目的字段中的宽度（以像素为单位）。1 字节的垂直像素延迟字段包含作为延迟值表（通常使用与 DPVL 相同的表）的索引的值。将从表中读取的该值乘以分组目的字段中的高度（以像素为单位）。

[0458] 固定延迟字段使用 1 字节作为延迟值表（与 DPVL 相同的表）的索引。从所述表中读取的该值是固定延迟参数，其表示处理与分组中指定的任何参数值无关的分组所需的时间。总延迟，或分组处理完成的时间延迟可根据如下关系加以确定：

[0459] 延迟 = ( 分组处理延迟 ( 像素延迟 ) · 像素总数 ) +

[0460] ( 分组处理延迟 ( 水平像素延迟 ) · 宽度 ) +

[0461] ( 分组处理延迟 ( 垂直像素延迟 ) · 高度 ) +

[0462] 分组处理延迟 ( 固定延迟 )

[0463] 对于某些分组来说，不使用总像素数、宽度或者高度，这是因为在相应的分组中没有引用这些参数。在这些情况下，相应的像素延迟参数通常设置为零。

[0464] 39. 个人显示器能力分组

[0465] 个人显示器能力分组提供一组参数，用于描述诸如头戴显示器或者显示器眼镜的个人显示器设备的能力。这使得主机能够按照客户机的具体能力来自定义显示信息。另一方面，客户机通过使用有效状态应答列表分组的有效参数应答列表中的相应参数来表明其能够发送个人显示器能力分组的能力。

[0466] 图 82 示出了个人显示器能力分组的一个实施例的格式。如图 82 所示，将个人显示器能力分组构造为具有分组长度、分组类型、cClient ID、所述子像素布局、像素形状、水平视场、垂直视场、可视轴交叉、左 / 右图像、透明度 (see through)、最大亮度、光学能力、最小 IPD、最大 IPD、视场曲率点列表和 CRC 字段。在一个实施例中，该分组长度固定为 68。分组类型值 141 将分组标识为个人显示器能力分组。cClient ID 字段被保留以供将来使用，并且现在通常设置为零。

[0467] 子像素布局字段使用以下值来说明子像素自上而下并且从左到右的物理布局，其中：使用 0 表明子像素布局并未定义；使用 1 表明红色、绿色、蓝色条纹；使用 2 表明蓝色、绿色、红色条纹；使用 3 表明具有 2×2 子像素排列的四个像素，该排列为红色处于顶部左侧，蓝色处于底部右侧以及两个绿色子像素，其中一个绿色子像素位于底部左侧，另一个位于顶部右侧；使用 4 来表明具有 2×2 的子像素排列的四个像素，该排列包括红色处于底部左侧、蓝色处于顶部右侧以及两个绿色子像素，其中一个位于顶部左侧，另一个处于底部右侧；使用 5 来表明 Δ (Delta) (三元组)；使用 6 来表明以红色、绿色和蓝色重叠（例如，利用场序色彩的 LCOS 显示）的马赛克；并且值 7 至 255 通常被保留供将来之用。

[0468] 像素形状字段使用以下值说明了由特定配置的子像素组成的每一像素的形状，其中：使用 0 来表明子像素形状未定义；使用 1 表明圆形；使用 2 表明正方形；使用 3 表明矩形；使用 4 来表明卵形；使用 5 来表明椭圆形；并且值 6 至 255 被保留以供将来用于表明所需要的形状，正如本领域普通技术人员理解的那样。

[0469] 1 字节的水平视场 (HFOV) 字段说明了以 0.5 度递增的水平视场（例如，如果 HFOV 是 30 度，那么其值是 60）。如果其值是零，那么不指定 HFOV。

[0470] 1 字节的垂直视场 (VFOV) 字段说明了以 0.5 度递增的垂直视场（例如，如果 VFOV 是 30 度，那么其值是 60）。如果其值是零，那么不指定 VFOV。

[0471] 1字节的视轴交叉字段说明了以0.01屈光度(1/m)递增的视轴交叉(例如,如果视轴交叉是2.22米,那么其值是45)。如果其值是零,视轴交叉未指定。

[0472] 1字节的左/右图像重叠字段说明了左右图像重叠百分比。图像重叠的容许范围的百分比是1到100。值101到255是无效的并且通常不使用。如果其值是零,那么不指定图像重叠。

[0473] 1字节的透明度(see through)字段说明了图像的透明度百分比。透明度的容许范围的百分比是0到100。值101到254是无效的并且将不会使用。如果其值是255,那么透明度百分比就未指定。1字节的最大亮度字段说明了以20尼特递增的最大亮度(例如,如果最大亮度是100尼特,那么其值是5)。如果其值是零,那么最大亮度就未被指定。

[0474] 2字节的光学能力标志字段-包含说明显示器光学能力的各种字段。通常按照如下情况分配这些位的值:

[0475] 位15至5被保留供将来之用,通常设置为零。

[0476] 位4选择眼镜聚焦调节,且0值意味着显示器不具有眼镜聚焦调节。1值意味着显示器具有眼镜聚焦调节。

[0477] 位3至2按照如下情况选择双目功能:数值0意味着显示器是双目的,并且只可以显示2维(2D)图像;数值1意味着显示器是双目的,并且可以显示3维(3D)图像;数值2-意味着所述显示器是单目的,而数值3被保留供将来之用。

[0478] 位1至0选择左右视场曲率对称性,数值0-意味着视场曲率没有定义。如果这个字段是零,那么除了点C3以外的从A1至E5的所有视场曲率值设置为零,点C3指定显示器的焦距,或者其被设置为零以便表明焦距没有被指定。数值1-意味着左右显示器具有相同对称性。数值2-意味着左右显示器在垂直轴(列C)上互为镜像。数值3被保留以供将来使用。

[0479] 1字节的最小瞳孔间距(IPD)字段用于以毫米(mm)为单位来指定最小瞳孔间距。如果其值是零,那么最小瞳孔间距没有被指定。1字节的最大瞳孔间距(IPD)用于以毫米(mm)为单位来指定最大瞳孔间距。如果其值是零,那么最大瞳孔间距没有被指定。

[0480] 视场曲率点列表字段包含一系列25个2字节的参数,用于在1到65535的范围以千分之一的屈光度(1/m)为单位来指定焦距(例如,1是0.001屈光度,并且65535是65.535屈光度)。视场曲率点列表中的25个元素被标记为A1至E5,如图83所示。所述点应在显示器的激活区域上均匀分布。列C对应于显示器的纵轴,行3对应于显示器的横轴。列A和E分别对应于显示器的左右沿。并且行1和5分别对应于显示器的顶部和底沿。列表中25个点的顺序是:A1, B1, C1, D1, E1, A2, B2, C2, D2, E2, A3, B3, C3, D3, E3, A4, B4, C4, D4, E4, A5, B5, C5, D5, E5。

[0481] CRC字段包含分组中包括分组长度在内的所有字节的CRC。

[0482] 40. 客户机错误报告分组

[0483] 客户机错误报告分组充当用于允许客户机向主机提供一系列操作错误的机制或者手段。作为从主机接收到某些命令的结果,客户机可以检测其正常操作情况下的各种各样的错误。这些错误的实例包括:客户机已经得到命令在其不支持的模式中操作;所述客户机可能已经接收到包含某些参数的分组,而这些参数超出客户机能力或者范围;所述客户机可能得到命令按照不正确的序列进入一种模式。所述客户机错误报告分组可以用来检测

正常操作期间的错误,但是对系统设计员和集成商最有用的是在主机和客户机系统的开发和集成过程中诊断问题。所述客户机使用有效状态应答列表分组的有效参数应答列表中的参数值 142 来表明其发送显示器错误报告分组的能力。

[0484] 图 84A 示出了客户机错误报告数据的一个实施例的格式。如图 84A 所示,将客户机错误报告数据构造为具有分组长度、分组类型、cClient ID、列表项数、错误代码列表和 CRC 字段。分组类型值 142 将一个分组标识为客户机错误报告分组。cClient ID 字段被保留供将来使用,并现在通常设置为零。列表项数(2 字节)字段说明了随后的错误代码列表中项目的数目。错误代码列表字段(这里 8 个字节)是一个包含一个或多个错误报告列表项的列表。图 87B 示出了单个错误报告列表项的格式。

[0485] 在一个实施例中,如图 87B 所示,每一错误报告列表项在长度上正好是 4 字节,并且在一个实施例中,其具有这样的结构,包括:用于说明所报告的错误类型的 2 字节显示器错误代码字段,用于说明与由客户机错误代码分组定义的错误有关的更加详细的细节的 2 字节错误子代码字段。每一客户机错误代码的具体定义由客户机的制造商定义。不必为每一显示器错误代码定义错误子代码,并且在错误子代码未被定义的情况下,所述值被设置为零。每一错误子代码的具体定义由客户机的制造商定义。

[0486] 41. 客户机标识分组

[0487] 客户机标识分组允许客户机响应于请求特定状态分组来返回标识数据。在一个实施例中,客户机使用有效状态应答列表分组的有效参数应答列表中的参数值 144 来表明发送客户机标识分组的能力。能够通过读取来自客户机的这些数据,而确定客户机设备的制造商名称和型号,对于主机而言是十分有用的。所述信息可以用来确定客户机是否具有分组客户机能力分组中无法描述的特殊能力。大概存在两种可能的方法、手段或机制用于从客户机读取识别信息。一种是通过使用分组客户机能力分组来实现,该分组客户机能力分组包含的字段类似于基本 EDID 结构中的那些字段。另一个方法是通过使用包含比分组客户机能力分组中类似字段更加丰富的信息集合的客户机标识分组。它允许主机识别没有被分配 3 字符 EISA 代码的制造商,并且允许序号包含字母数字字符。

[0488] 图 85 示出了客户机标识分组的一个实施例的格式。如图 85 所示,将客户机标识分组构造为具有分组长度、分组类型、cClient ID、制造星期、制造年份、制造商名称长度、产品名称长度、序号长度、制造商名称字符串、序号字符串和 CRC 字段。

[0489] 2 字节的分组类型字段包含用于把分组标识为客户机标识分组的值。在一个实施例中,将这一值选择为 144。所述 cClient ID 字段(2 字节)也被保留以供将来用于客户机 ID,并且通常设置为零。CRC 字段(2 字节)包含包括分组长度在内的分组中的所有字节的 16 位 CRC。

[0490] 1 字节的制造星期字段包含用于定义显示器制造的星期的值。在至少一个实施例中,如果客户机支持这个字段,那么这一值在 1 到 53 的范围之内。如果客户机不支持这个字段,那么通常将其设置为零。1 字节的制造年份字段包含用于定义客户机(显示器)制造的年份的值。这一值是从作为起点的 1990 年的偏离值,当然也可以使用其它基准年。这个字段可以表示 1991 到 2245 范围内的年份。例如,2003 年对应于制造年份值 13。如果客户机不支持这个字段,那么通常将其设置为零。

[0491] 制造商名称长度、产品名称长度以及序号长度字段均包含 2 字节的值,分别用于

说明包括任何空终止符或者空填充字符的制造商名称字符串字段的长度,包括任何空终止符或空填充字符的产品名称字符串字段的长度,以及包括任何空终止符或者空填充字符的序号字符串字段的长度。

[0492] 制造商名称字符串、产品名称字符串以及序号字符串字段均包含分别由制造商名称、产品名称和序号的长度字段指定的可变字节数,且包含 ASCII 字符串,分别用于说明制造商、产品名称以及显示器字母数字序号。这些字符串的每个均由至少一个空字符来终止。

[0493] 42. 可选 (alternate) 显示器能力分组

[0494] 可选显示器能力分组表明连接于 MDDI 客户机控制器的可选显示器的能力。该分组是响应于请求特定状态分组而发送的。当被提示时,客户机设备为所支持的每个可选显示器发送可选显示器能力分组。所述客户机能够经由有效状态应答列表分组的有效参数应答列表中的参数值 145 来表明其发送可选显示器能力分组的能力。

[0495] 对于按照内部模式操作的 MDDI 系统来说,具有一个以上连接至 MDDI 客户机控制器的显示器是非常普遍的。一个示例性应用是移动电话在翻盖内部具有较大显示器而在外部具有较小显示器。内部模式的客户机不必返回可选显示器能力分组,原因有二。其一,在维护期间,主机可能已经被编程或得知这些能力,因为它们用在公共设备或外壳中。其二,由于合而为一,客户机不能很容易地与到主机的连接断开或分离,主机可以包含客户机能力的硬编码副本,或者,至少知道它们不随客户机的变化而改变,否则就可能会发生这种情况。

[0496] 客户机能力分组的可选显示器数目字段用来报告所连接的一个以上的显示器,并且可选显示器能力分组报告每一可选显示器的能力。所述视频流分组在像素数据属性字段中包含 4 位以便访问客户机设备中的每一可选显示器。

[0497] 图 89 示出了可选显示器能力分组的一个实施例的格式。如图 86 所示,将可选显示器能力分组构造为具有分组长度、分组类型、cClient ID、可选显示器号、保留 1、位图宽度、位图高度、显示器窗口宽度、显示器窗口高度、色彩映射 RGB 宽度、RGB 能力、单色能力、保留 2、Y Cb Cr 能力、显示器特征能力、保留 3 和 CRC 字段。分组类型值 145 将一个分组标识为可选显示器能力分组。cClient ID 字段被保留供将来客户机 ID 使用,并通常设置为零。

[0498] 可选显示器号字段使用 1 字节,用 0 到 15 范围内的整数表明可选显示器的身份。第一可选显示器通常指定为 0 号,而其它可选显示器用唯一可选显示器号的值来标识,且最大值是可选显示器总数减去 1。不应该使用大于可选显示器总数减去 1 的值。例如:具有主显示器和连接至 MDDI 客户机的主叫 ID 显示器的移动电话具有一个可选显示器,因此主叫 ID 显示器的可选显示器号是零,并且显示器能力分组的可选显示器数目字段的值为 1。

[0499] 保留 1 字段 (1 字节) 被保留供将来之用。这个字段中的所有位都设置为零。这个字段的目的在于让所有后续 2 字节字段与 16 位地址对准,让 4 字节字段与 32 位地址对准。

[0500] 位图宽度字段使用 2 字节来说明以像素个数表示的位图宽度。位图高度字段使用 2 字节来说明以像素个数表示的位图高度。显示器窗口宽度字段使用 2 字节来说明以像素个数表示的显示器窗口宽度。显示器窗口高度字段使用 2 字节来说明以像素个数表示的显示器窗口高度。

[0501] 色彩映射 RGB 宽度字段使用 2 字节来说明可以按照色彩映射 (调色板) 显示模式显示的红色、绿色和蓝色色彩分量的位数。每个色彩分量 (红色、绿色和蓝色) 最多可以使



用 8 位。即便在色彩映射分组中发送每一色彩分量的 8 位,也仅仅使用在这个字段中定义的每一色彩分量的最低有效位的位数。如果显示器客户机无法使用色彩映射(调色板)格式,那么这一值是零。所述色彩映射 RGB 宽度字包括三个独立的无符号值:

[0502] 位 3 至 0 定义每一像素中蓝色的最大位数,且数值 0 至 8 被认为是有效的。位 7 至 4 定义每一像素中绿色的最大位数,且数值 0 至 8 被认为是有效的。位 11 至 8 定义每一像素中红色的最大位数,且数值 0 至 8 被认为是有效的。当位 14 至 12 被保留供将来使用并且通常设置为零。位 15 用于表明客户机接受打包或未打包格式的色彩映射像素数据的能力。当位 15 被设置为逻辑 1 电平时,它表明客户机能够接受打包或未打包格式的色彩映射像素数据。如果位 15 被设置为逻辑 0 电平,它表明客户机只能接受未打包格式的色彩映射像素数据。

[0503] RGB 能力字段使用 2 字节来说明可以按照 RGB 格式显示的分辨率的位数。在一个实施例中,如果客户机无法使用 RGB 格式,那么这一值是零。所述 RGB 能力字包括三个独立的无符号值:位 3 至 0 定义每一像素中蓝色(蓝色强度)的最大位数。位 7 至 4 定义每一像素中绿色(绿色强度)的最大位数。位 11 至 8 定义每一像素中红色(红色强度)的最大位数。位 14 至 12 被保留供将来使用并且设置为零。位 15 用于表明客户机接受打包或未打包格式的 RGB 像素数据的能力。当位 15 被设置为逻辑 1 电平时,它表明客户机能够接受打包或未打包格式的 RGB 像素数据。如果位 15 被设置为逻辑 0 电平,它表明客户机只能接受未打包格式色彩映射像素数据。

[0504] 1 字节的单色能力字段包含用来指定可以按照单色格式显示的分辨率的位数的值或信息。如果客户机无法使用单色格式,那么这一值是零。位 6 至 4 被保留供将来使用并且通常设置为零。位 3 至 0 定义每一像素中可以存在的灰度级的最大位数。这四位能够指定每一像素包括 1 至 15 位。如果所述值是零,那么客户机不支持所述单色格式。当位 7 被设置为 1 时,它表明客户机能够接受打包或未打包的单色像素数据。如果位 7 被设置为 0,它表明客户机只能接受未打包格式的单色像素数据。

[0505] 保留 2 字段是为将来之用保留的 1 字节宽的字段,并且通常将这个字段中的所有位都应设置为零。在一个实施例中,这个字段的目的在于让所有后续 2 字节字段与 16 位字地址对准,让 4 字节字段与 32 位字地址对准。

[0506] 2 字节的 Y Cb Cr 能力字段说明了可以按照 Y Cb Cr 格式显示的分辨率的位数。如果客户机无法使用 Y Cb Cr 格式,那么这一值是零。所述 Y Cb Cr 能力字包括三个独立的无符号值:位 3 至 0 定义用于说明 Cb 采样的最大位数。位 7 至 4 定义用于说明 Cr 采样的最大位数。位 11 至 8 定义用于说明 Y 采样的最大位数。位 14 至 12 被保留供将来使用并且设置为零。当位 15 被设置为 1 时,它表明客户机能够接受打包或未打包的 Y Cb Cr 像素数据。如果位 15 被设置为 0,它表明客户机只能接受未打包格式的 Y Cb Cr 像素数据。

[0507] 2 字节的 Bayer 能力字段说明能够以 Bayer 格式传送的分辨率的位数、像素组和像素顺序。如果客户机不能使用间隔格式,那么这个值被置为零。Bayer 能力字段由以下值组成:位 3 至 0 定义每个像素中存在的强度的最大位数;位 5 至 4 定义所需要的像素组图案;位 8 至 6 定义用于所需要的像素顺序;位 14 至 9 被保留供将来使用并且设置为零。当位 15 被设置为 1 时,它表明客户机能够接受打包或未打包的 Bayer 像素数据。如果位 15 被设置为 0,它表明客户机只能接受未打包格式的 Bayer 像素数据。

[0508] 2字节的CRC字段包含包括分组长度在内的分组中所有字节的16位CRC。

### [0509] 43. 寄存器访问分组

[0510] 寄存器访问分组向主机或者客户机提供用于访问MDDI链路对端的配置和状态寄存器的手段、机制或方法。这些寄存器对于每个显示器或设备控制器而言很可能是唯一的。这些寄存器已存在于许多显示器中，它们要求设置配置、操作模式并且具有其它有益的和必需的设置。所述寄存器访问分组允许MDDI主机或者客户机对寄存器进行写操作，并且使用MDDI链路请求读取寄存器。当主机或者客户机请求读取寄存器时，对端应该通过以相同的分组类型发送寄存器数据，并且通过利用读/写信息字段来表明这是从特定寄存器读取的数据而作为响应。寄存器访问分组可以用来通过指定大于1的寄存器计数来读取或者写入多个寄存器。客户机使用分组客户机能力分组的客户机特征能力字段的位22表明支持寄存器访问分组的能力。

[0511] 图87示出寄存器访问分组的一个实施例的格式。如图87所示，将寄存器访问分组构造为具有分组长度、分组类型、bClient ID、读/写标志、寄存器地址、参数CRC、寄存器数据列表和寄存器数据CRC字段。分组类型值146将一个分组标识为寄存器访问分组。bClient ID字段被保留供将来之用，并且通常设置为零。

[0512] 2字节的读/写标志字段将这个特定分组说明为写、读或者对读的响应，并且提供这些数据值的计数。

[0513] 位15至14用作读/写标志。如果位[15:14]为00，那么该分组包含待写入由寄存器地址字段寻址的寄存器的数据。待写入指定寄存器的分组含在寄存器数据列表字段中。如果位[15:14]为10，那么这是从由寄存器地址字段寻址的一个或多个寄存器请求数据。如果位[15:14]为11，那么这个分组响应于具有位15:14的读/写标志设置为10的寄存器访问分组而包含所请求的数据。寄存器地址字段包含对应于第一寄存器数据列表项的寄存器地址，而寄存器数据列表字段包含从所述一个或多个地址中读取的数据。如果位[15:14]位01，则这一值被保留以供将来使用并且不在此时使用，但本领域技术人员将会明白如何对为了应用而采用它。

[0514] 位13:0使用14位无符号整数来说明将于寄存器数据列表字段中传送的32位寄存器数据项的数目。如果分组位15:14等于“00”，那么位13:0指定32-位寄存器数据项的数目，该寄存器数据项包含在寄存器数据列表字段，且待写入从寄存器地址字段指定的寄存器开始的寄存器中。如果分组位15:14是“10”，那么位13:0指定接收设备发送至请求读取该寄存器的设备的32-位寄存器数据项的数目。该分组中的寄存器数据列表字段应该不包含项目并且长度是零。位15:14等于“11”，那么位13:0分组指定已经从寄存器中读出的包含在寄存器数据列表字段中的寄存器数据项的数目。目前，位15:14不会被设置为等于“01”，这被视为无效值，否则被保留以供将来指定或使用。

[0515] 寄存器地址字段使用4字节来表明待将要被写入或者将要从中读取的寄存器的地址。为了对地址小于32位的寄存器寻址，高位应被设置为零。

[0516] 2字节的参数CRC字段包含从分组长度到寄存器地址在分组所有字节的CRC。如果该CRC未能通过校验，则丢弃整个分组。

[0517] 寄存器数据列表字段包含待写入客户机寄存器的一系列4字节寄存器数据值，或已经从客户机设备寄存器中读取的值。

[0518] 2字节的寄存器数据CRC字段包含只对寄存器数据列表的CRC。如果该CRC未能通过校验,则寄存器数据仍可使用,不过CRC错误计数将递增。

#### [0519] D. 分组CRC

[0520] CRC字段出现在分组的结尾,有时出现在分组中的某些更关键的参数之后,这些参数具有明显较大数据字段,从而增大了传送期间出错的可能性。在具有两个CRC字段的分组中,当只有一个CRC被使用时,CRC生成器在第一CRC之后被重新初始化,因此跟随长数据字段的CRC计算不受分组开始时的参数影响。

[0521] 在一个示例性的实施例中,用于CRC计算的多项式为CRC-16或者 $X^{16}+X^{15}+X^2+X^0$ 。图36说明可用于实现本发明的CRC生成器和校验器3600的示例性实现方式。在图36中,CRC寄存器3602正好在传送被输入到Tx\_MDDI\_Data\_Before\_CRC(发送在CRC之前的MDDI数据)线路上的分组的第一位之前被初始化为值0x0001,然后分组的字节首先从LSB开始被移入寄存器。应注意的是,该图中的寄存器位数对应于所使用的多项式的阶数,而不是由MDDI使用的位的位置。CRC寄存器沿单向移位更加有效,这使得CRC位15出现在MDDI CRC字段的位位置0处,CRC寄存器位14出现在MDDI CRC字段位位置1处等等,直到到达MDDI位位置14。

[0522] 举例来说,如果客户机请求和状态分组的分组内容是:0x000c、0x0046、0x000、0x0400、0x00、0x0000(或者表示为一个字节序列:0x0c,0x00,0x46,0x00,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x00),并且使用多路复用器3604和3606以及与非(NAND)门3608的输入来提交(submit),Tx\_MDDI\_Data\_With\_CRC(发送具有CRC的MDDI数据)线路上输出的最终CRC是0xd9aa(或者表示为一个序列0xaa,0xd9序列)。

[0523] 当CRC生成器和校验器3600被配置成CRC校验器时,在Rx\_MDDI\_Data(接收MDDI数据)线路上接收的CRC被输入至多路复用器3604和NAND门3608,并且使用或非(NOR)门3610、异或(XOR)门3612和与(AND)门3614逐位与CRC寄存器中发现的值进行比较。如果存在任何错误,如AND门3614输出的那样,那么针对包含CRC错误的每个分组,通过把门3614的输出连接至寄存器3602的输入,CRC增加一次。应注意的是,图36中示出的示例性电路可以输出给定CHECK\_CRC\_NOW(现在校验CRC)窗口内一个以上的CRC错误信号(参见图37B)。因此,所述CRC错误计数器通常只对在校验CRC窗口内每一间隔内的第一个CRC错误事例计数。如果被配置成CRC生成器,那么在与分组末尾对齐的时刻,将CRC按照时钟从所述CRC寄存器输出。

[0524] 图37A和37B中以图形方式说明输入和输出信号以及使能信号的时序。图37A用Gen\_Reset(生成复位)、Check\_CRC\_Now(现在校验CRC)、Generate\_CRC\_Now(现在生成CRC)和Sending\_MDDI\_Data(发送MDDI数据)信号的状态以及Tx\_MDDI\_Data\_Before\_CRC(发送CRC之前的MDDI数据)和Tx\_MDDI\_Data\_With\_CRC(发送具有CRC的MDDI数据)信号的状态(0或者1)示出了CRC的生成和数据的分组的传输。图37B中以Gen\_Reset、Check\_CRC\_Now、Generate\_CRC\_Now和Sending\_MDDI\_Data信号以及Rx\_MDDI\_Data(接收MDDI数据)和CRC\_Error(CRC错误)信号的状态示出了数据的分组的接收和CRC值的校验。

#### [0525] E. 分组CRC的错误代码重载(overload)

[0526] 每当在主机和客户机之间只传送数据的分组和CRC时,其中不容纳错误代码。唯

一的错误是失去同步。否则,必须等待链路因缺乏良好的数据传送路径或者流水线而超时,然后重置所述链路并且继续进行。但是,这样做既耗时效率又低。

[0527] 针对一个实施例中的使用,已经开发了这样一种新技术,其中分组的 CRC 部分用来传送错误代码信息。这在图 65 中大体上示出。也就是说,由处理数据传送的处理器或设备生成一个或多个错误代码,其表明通信处理或者链路内可能出现的特定预定错误或者缺陷。遇到错误时,使用分组的 CRC 位来生成并且传送适当的错误代码。也就是说,利用所需要的错误代码来重载或者重写所述 CRC 值,该所需的错误代码可以在接收端由用于监视 CRC 字段值的错误监视器或者校验器检测到。对于错误代码与 CRC 值因为某些原因而相匹配的情况,传送错误码的补码以便防止混淆。

[0528] 在一个实施例中,为了提供稳固 (robust) 的错误警告和检测系统,使用一系列分组、通常是所有分组传送若干次错误代码,这些分组是在已经检测到错误之后才被传送或者发送的。这种情况将一直出现,直到从系统中清除产生错误的条件时为止,此时传送正常的 CRC 位,而不用另一个值来重载。

[0529] 重载 CRC 值的这种技术对系统错误提供非常快的响应,同时使用了最小量的额外位或字段。

[0530] 如图 66 所示,利用错误检测器或者检测模块 6602 示出了 CRC 重写机制或者装置 6600,它能够形成先前描述或者了解的其它电路的一部分,用于检测通信链路或者过程内错误的存在或出现。可以形成为其它电路的一部分或者使用诸如查阅表来储存预选错误消息的技术的错误代码生成器或者模块 6604 生成一个或多个错误代码,以表明已经检测到的出现的特定的预定错误或者缺陷。易于理解的是,设备 6602 和 6604 根据需要可以形成为单个电路或者设备,或者作为用于其它已知处理器和元件的被编程的步骤序列的一部分。

[0531] 示出的 CRC 值比较器或者比较模块 6606,用于检查所选择的一个或多个错误代码是否与被传送的 CRC 值相同。如果是,就利用代码补码生成器或者生成模块或设备来提供错误代码的补码,以免错误代码被误认为是原来的 CRC 模式 (pattern) 或者值,从而使检测方案混淆或者使其复杂。然后,错误代码选择器或者选择模块元件或设备 6610 选择期望插入或者重写的错误代码,或者它们各自的补码,这视情况而定。错误代码 CRC 重写器或者重写机制或者模块 6612 是一种设备,用于接收数据流、分组和待插入的期望的代码,并且重写相应的或者适当的 CRC 值,以便把期望的错误代码传送至接收设备。

[0532] 如上所述,可以使用一系列分组多次传送错误代码,因此重写器 6612 可以在处理或者调用来自先前元件或者其它已知存储单元的这些代码期间,利用存储器储存元件来保存代码的副本,其中这些存储单元根据需要或期望用于储存或者保留它们的值。

[0533] 图 67A 和 67B 中更加详细地示出了图 66 的重写机制实现的总体处理的实现。在 67A 中,在步骤 6702,在通信数据或者过程中检测到一个或多个错误,并且在步骤 6704,选择错误代码来表明这一条件。同时,或者在适当点,在步骤 6706 校验待更换的 CRC 值,并且在步骤 6708 将其与需要的错误代码进行比较。如早先讨论的那样,这一比较的结果是确定所需要的代码或者其它代表值是否与当前 CRC 值相同。如果是,那么处理进行到步骤 6712,把所述补码或者在某些情况下根据需要是其它代表值,选为代码来插入。一旦在步骤 6710 和 6714 中确定了将会插入什么样的错误代码或者值,就选择这个适当的代码以便插入。为了清楚起见,把这些步骤作为独立的步骤示出,但是它通常表现为基于步骤 6708 判定的输

出所作的单一选择。最后,在步骤 6716,在 CRC 位置中重写这些适当值,以便与作为所述过程的目标的分组一起传送。

[0534] 在分组接收侧,如图 67B 所示,在步骤 6722 中监视分组 CRC 值。通常,所述 CRC 值由系统内的一个或多个过程监视,以便确定数据传送中的错误是否已经出现,并且确定是否请求重新传送一个或多个分组,或是禁止进一步操作等等,其中一些上文已经讨论过了。作为这种监视的一部分,还可以使用所述信息与已知的或者预选错误码或者代表值进行比较,并检测错误的出现。或者,也可以实现独立的错误检测过程和监视器。如果好像存在这样一个代码,就在步骤 6724 提取它或者标注它,以便进一步处理。在步骤 6726 中可以确定这是实际代码还是补码,在这种情况下,将附加步骤 6728 用来把所述值转换为所需要的代码值。不论是哪种情况,在步骤 6730 最终提取的代码、补码或者其它恢复值都可用于从已传输的代码中检测已经出现了哪些错误。

#### [0535] V. 链路休眠

[0536] MDDI 链路可以快速进入休眠状态,并且快速从休眠中苏醒。由于 MDDI 链路能够非常快速地再次苏醒以便使用,因而这种响应允许通信系统和设备强制 MDDI 链路经常进入休眠以降低功耗。在一个实施例中,当外部模式的客户机第一次从休眠中苏醒时,它以数据速率以及选通脉冲时序进行如此操作,其中选通脉冲时序与 1Mbps 的速率一致,也就是说, MDDI\_Stb 对应以 500kHz 的速率翻转。一旦客户机的特性已经被主机发现或者其特性已经传递到主机,那么主机通常可以以从 1Mbps 到客户机能够运行的最大速率之间的任意速率唤醒链路。内部模式的客户机能够以主机和客户机双方均可运行的任意速率苏醒。这通常也适用于内部模式客户机第一次苏醒。

[0537] 在一个实施例中,当链路从休眠中苏醒时,主机和客户机交换脉冲序列。使用低速线路接收器可检测到这些脉冲,该低速线路接收器仅消耗差分接收器以最大链路运行速度接收信号所需的电流的一部分。主机或者客户机都能够唤醒链路,从而唤醒协议被设计成能够处理主机和客户机同时试图进行唤醒时出现的可能的竞争。

[0538] 在休眠状态期间, MDDI\_Data 和 MDDI\_Stb 差分驱动器被禁用并且跨在所有差分对上的差分电压为零电压。在从休眠苏醒期间用于检测脉冲序列的差分线路接收器具有有意的电压偏移。在一个实施例中,这些接收器中逻辑 1 和逻辑 0 电平之间的阈值近似为 125mV。这使得未驱动的差分线对在链路苏醒序列期间将被视为逻辑零电平。

[0539] 为了进入休眠状态,主机在链路关闭分组的 CRC 之后发送 64 个 MDDI\_Stb 周期。主机禁用 CRC 之后的在 15 至 56 个 MDDI\_Stb 周期(包括输出禁用传播延迟)的范围内主机的 MDDI\_Data0 输出。主机在链路关闭分组的 CRC 之后且在其发起唤醒序列之前发送 64 个 MDDI\_Stb 周期。在一个实施例中,主机发起的唤醒被定义为主机在 MDDI\_Data0 达到有效逻辑 1 电平之后且在驱动 MDDI\_Stb 上的脉冲之前必须等待 100 纳秒。在一个实施例中,客户机在链路关闭分组的 CRC 之后且在将等待 MDDI\_Data0 驱动到逻辑 1 电平以试图唤醒主机之前需要等待至少 60 个 MDDI\_Stb 周期。

[0540] 为了从休眠状态中“苏醒”,需要执行几个动作和过程。当客户机,这里是显示器,需要来自主机的数据或通信、服务时,客户机将 MDDI\_Data0 线路驱动到逻辑 1 状态且持续大约 70 到 1000 微秒,而 MDDI\_Stb 处于非激活状态,且在 MDDI\_Stb 变为激活状态之后仍保持 MDDI\_Data0 被驱动为逻辑 1 电平大约 70 个 MDDI\_Stb 周期(在 60 到 80 的范围上),但

也可根据需要使用其他时段。然后客户机通过将 MDDI\_Data0 驱动器置为高阻状态而禁用 MDDI\_Data0 驱动器。

[0541] 如果在休眠期间 MDDI\_Stb 处于激活状态,尽管这不太可能,那么客户机只能将 MDDI\_Data0 驱动到逻辑 1 状态大约 70 个 MDDI\_Stb 周期(在 60 到 80 的范围上)。这一动作使得主机启动和重启在正向链路(208)上的数据业务并询问客户机的状态。

[0542] 主机必须检测请求脉冲的出现并开始启动序列,即,先将 MDDI\_Stb 驱动为逻辑 0 电平和将 MDDI\_Data0 驱动为逻辑高电平达至少 100nsec。然后,在翻转 MDDI\_Stb 的同时,继续将 MDDI\_Data0 驱动为逻辑 1 电平且持续 150 个 MDDI\_Stb 周期(在 140 到 160 的范围上),并将 MDDI\_Data0 驱动为逻辑 0 电平且持续 50 个 MDDI\_Stb 周期(在 40 到 60 的范围上)。如果客户机检测到 MDDI\_Data0 处于逻辑 1 状态多于 80 个 MDDI\_Stb 周期,那么客户机不应发送服务请求。当客户机检测到 MDDI\_Data0 处于逻辑 1 电平达 60 至 80 个 MDDI\_Stb 周期时,那么客户机搜索主机将 MDDI\_Data0 驱动为逻辑 0 电平达 50 个 MDDI\_Stb 周期的间隔。在主机将 MDDI\_Data0 驱动到逻辑 0 电平且持续 50 个 MDDI\_Stb 周期的时间后,主机随后开始在链路上发送分组。发送的第一个分组是子帧报头分组。在 MDDI\_Data0 处于逻辑 0 电平达 50 周期间隔的 40 个 MDDI\_Stb 周期后,客户机开始寻找子帧报头分组。以下进一步讨论与休眠处理和启动序列相关的世界选择属性和时间间隔容差。(参见以下的图 68A-C)

[0543] 主机可通过首先启用 MDDI\_Stb 而发起唤醒,并同时将其驱动至逻辑 0 电平。直至脉冲按照如下所述方式输出,MDDI\_Stb 才被驱动至逻辑 1 电平。在 MDDI\_Stb 达到有效逻辑 0 电平后,主机启用 MDDI\_Data0 并同时将其驱动到逻辑 1 电平。直到如下所述 MDDI\_Data0 被驱动到逻辑 0 电平且持续 50 个 MDDI\_Stb 脉冲的间隔时,MDDI\_Data0 才应被驱动到逻辑 0 电平。主机应该在 MDDI\_Data0 达到逻辑 1 电平之后且在驱动 MDDI\_Stb 上脉冲之前等待至少 100 纳秒。当考虑到最差情况的输出启用延迟时,出现这种时序关系。这基本确保了客户机在其 MDDI\_Stb 接收器由被主机驱动的 MDDI\_Data0 上的逻辑 1 电平唤醒之后有充足的时间来完全启用该 MDDI\_Stb 接收器。

[0544] 图 38 示出了在没有竞争的情况下典型服务请求事件 3800 的处理步骤的实例,其中为了说明的方便,使用字母 A、B、C、D、E、F 和 G 来标记所述事件。所述过程在点 A 处开始,此时主机向客户机设备发送链路关闭分组,以便通知它所述链路将转变为低功率休眠状态。在下一步,通过禁用 MDDI\_Data0 驱动器并且把 MDDI\_Stb 驱动器设置为逻辑 0,所述主机进入低功率休眠状态,如 B 点处所示。通过高阻抗偏置网络将 MDDI\_Data0 驱动为逻辑 0 电平。在某时间段之后,客户机通过把 MDDI\_Data0 驱动至逻辑 1 电平来向主机发送服务请求,如点 C 处所示那样。所述主机仍然使用高阻抗偏置网络来维持(asserts)逻辑 0 电平,但是客户机中的驱动器强制所述线路进入逻辑 1 电平。在 50 微秒内,所述主机识别这一服务请求脉冲,并且通过启用其驱动器在 MDDI\_Data0 上维持逻辑 1,如点 D 处所示。然后,客户机停止试图维持服务请求脉冲的努力,并且所述客户机将其驱动器置于高阻状态,如点 E 处所示。主机把 MDDI\_Data0 驱动至逻辑 0 电平且持续 50 微秒,如点 F 处所示,并且按照与 MDDI\_Data0 上的逻辑 0 电平相一致的方式来生成 MDDI\_Stb。在 MDDI\_Data0 处于逻辑 0 电平达 40 个 MDDI\_Stb 周期后,客户机开始寻找子帧报头分组。在把 MDDI\_Data0 置于逻辑 0 电平并且驱动 MDDI\_Stb 50 微秒之后,所述主机开始通过发送子帧报头分组在正向

链路上传输数据,如点 G 所示。

[0545] 图 39 中说明类似的实例,其中在已经开始链路重启序列之后维持 (assert) 服务请求,并且也使用字母 A、B、C、D、E、F 和 G 来标记所述事件。这代表来自客户机的请求脉冲或者信号非常接近破坏子帧报头分组的最坏情况。所述过程在点 A 处开始,此时主机向客户机设备再次发送链路关闭分组,通知它所述链路将转变到低功率休眠状态。在下一步,所述主机通过禁用 MDDI\_Data0 驱动器并且把 MDDI\_Stb 驱动器设置为逻辑 0 电平,而进入低功率休眠状态,如 B 点处所示。如前所述,通过高阻抗偏置网络将 MDDI\_Data0 驱动为逻辑 0 电平。经过一段时间以后,主机通过把 MDDI\_Data0 驱动为逻辑 1 电平且持续 150 微秒来开始所述链路重启序列,如点 C 处所示。在链路重启序列开始之后且在经过 50 微秒之前,所述显示器还维持 MDDI\_Data0 为 1 达 70 微秒的持续时间,如点 D 处所示。发生这种情况是因为显示器需要向主机请求服务并且尚未识别出所述主机已开始链路重启序列。然后,客户机不再试图维持服务请求脉冲,并且所述客户机将其驱动器置于高阻状态,如点 E 处所示。主机继续把 MDDI\_Data0 驱动至逻辑 1 电平。主机把 MDDI\_Data0 驱动至逻辑 0 电平且持续 50 微秒,如点 F 所示,并且按照与 MDDI\_Data0 上为逻辑 0 电平相一致的方式来生成 MDDI\_Stb。在把 MDDI\_Data0 维持到逻辑 0 电平并且驱动 MDDI\_Stb 达 50 微秒之后,所述主机通过发送子帧报头分组开始在正向链路上传输数据,如点 G 所示。

[0546] 根据上面的讨论,人们可以看出:作为唤醒序列的一部分,先前解决方案涉及让主机经历两个状态。对于第一状态来说,主机把 MDDI\_Data0 信号驱动为高且持续 150 微秒,然后把 MDDI\_Data0 信号驱动为低且持续 50 微秒,同时激活 MDDI\_Stb 线路,然后开始传输 MDDI 分组。就使用 MDDI 装置和方法可获得的数据速率而言,这一过程可很好地用于提高现有技术水平。然而,如前所述,就减少对条件的响应时间或能够更快选择下一步骤或过程(即,简化处理或元件的能力)而言,始终要求更快的速度。

[0547] 申请人已经发现了一种新的创新方法来唤醒处理和时序,其中主机使用基于时钟周期的时序进行信号翻转 (toggle)。按照这一配置,在主机于唤醒序列开始时把 MDDI\_Data0 信号驱动为高之后,所述主机从 0 至 10 微秒开始翻转 MDDI\_Stb,并且直到信号被驱动为低才等待。在唤醒序列期间,主机翻转 MDDI\_Stb,就好象 MDDI\_Data0 信号始终处于逻辑 0 电平。这样做有效地把时间概念从客户机侧去除,并且对于这些周期而言,主机从最初两个状态的 150 微秒和 50 微秒时间段改变为针对这些时间段的 150 个时钟周期和 50 个时钟周期。

[0548] 现在,主机负责把该数据线驱动为高,并且在 10 个时钟周期内开始传输选通信号,就好像所述数据线是零。在主机已经把数据线驱动为高且持续 150 个时钟周期之后,主机把数据线驱动为低并持续 50 个时钟周期,同时继续传输选通信号。在主机已经完成这两个过程之后,主机可以开始传输第一子帧报头分组。

[0549] 在客户机侧,现在,客户机实现方式可以使用所生成的时钟计算数据线首先为高、然后为低的时钟周期的数目。在驱动为高状态的数据线上需要存在的时钟周期数目是 150,在驱动为低状态的数据线上的时钟周期数目是 50。这意味着对于正确的唤醒序列来说,客户机应该能够对数据线处于高状态的至少 150 个连续时钟周期,及其后跟随的数据线处于低状态的至少 50 个连续时钟周期进行计数。一旦这两个条件得到满足,客户机就可以开始搜索第一子帧的唯一字。把这一模式中的中断用作把计数器返回初始状态的基础,其中客

户机再次寻找数据线处于高状态的最初 150 个连续时钟周期。

[0550] 如先前所讨论的那样,基于主机的从休眠中唤醒的本发明的客户机实现方式,除了没有强制时钟速率起始于 1Mbps 以外,非常类似于初次启动的情况。相反,可以将时钟速率设置为从任何先前速率重新开始,该先前速率为通信链路进入休眠时处于激活状态的速率。如果主机如上所述开始传输选通信号,那么客户机应该能够再次对数据线处于高状态的至少 150 个连续时钟周期,和其后跟随的数据线处于低状态的至少 50 个连续时钟周期进行计数。一旦已经符合这两个条件,客户机就可以开始搜索唯一字。

[0551] 基于客户机的从休眠中唤醒的本发明的客户机实现方式,除了它通过让客户机驱动数据线来启动之外,与基于主机的唤醒相似。客户机可以在没有时钟的情况下异步地驱动数据线来唤醒主机设备。一旦主机识别出所述数据线被客户机驱动为高状态,它就可以开始其唤醒序列。客户机可以对开始时或者在其唤醒过程期间的主机生成的时钟周期的数目进行计数。一旦客户机数出数据线处于高状态的 70 个连续时钟周期,它就可以停止将数据线驱动为高状态。在这一刻,主机也应该已经把数据线驱动为高状态。于是客户机可以对数据线处于高状态的另外 80 个连续时钟周期进行计数,以便使数据线处于高状态的时钟周期达到 150 个,并且因此能寻找数据线处于低状态的 50 个时钟周期。一旦符合这三个条件,客户机就可以开始寻找唯一字。

[0552] 这种唤醒处理的新实现方式的优势在于它不需要时间测量设备。无论它是振荡器、电容器放电电路还是其它这种已知的设备,客户机不再需要这种外部设备来确定启动条件。当在客户机设备板上实现控制器、计数器等的时候,这样做节省资金和电路面积。虽然对于主机而言,未必象对于客户机那样有益,但是就用于核心电路的超高密度逻辑 (VHDL) 而言,这一技术应该也能够简化所述主机。使用数据和选通线路作为唤醒通知和测量源的功耗也非常低,这是因为不需要为核心元件运行任何外部电路,以使其等待基于主机的唤醒。所使用的周期或者时钟周期的数目是示例性的,还可以使用其它周期,这对于本领域普通技术人员是显而易见的。

[0553] 这种唤醒处理的新实现方式的优势在于它不需要时间测量设备。无论它是振荡器、电容器放电电路还是其它这种已知的设备,客户机不再需要这种外部设备来确定启动条件。当实现控制器、计数器等的时候,这样做节省资金和电路面积,因此节省客户机设备板上的面积。虽然对于主机而言,未必象对于客户机那样有益,但就用于核心电路的超高密度逻辑 (VHDL) 而言,这一技术应该也能够简化所述主机。使用数据和选通线路作为唤醒通知和测量源时的功耗也非常低,这是因为核心元件等待基于主机的唤醒不需要任何外部电路。

[0554] 为了阐明并且举例说明这一新技术的操作,图 68A、68B 和 68C 中示出了 MDDI\_Data0、MDDI\_Stb 和相对于所述时钟周期的各种操作的时序。

[0555] 图 68A 中示出了在没有竞争的情况下典型主机启动的唤醒的处理步骤的实例,其中为了便于说明,也使用字母 A、B、C、D、E、F 和 G 来标记所述事件。所述过程在点 A 处开始,此时主机向客户机设备发送链路关闭分组以便通知它所述链路将转变到低功率休眠状态。在下一步中,B 点处,主机翻转 (toggle)MDDI\_Stb 且持续翻转约 64 个周期(根据系统设计需要),以便在停止 MDDI\_Stb 翻转之前允许客户机的处理完成,其停止客户机设备中的已恢复时钟。主机最初也把 MDDI\_Data0 设置为逻辑 0 电平,然后在 CRC 之后,在 16 ~ 48



个周期的范围内（通常包括输出禁用传播延迟）禁用 MDDI\_Data0 输出。这可能需要在 CRC 后的 48 周期之后，并且在下一阶段 (C) 之前的某时，把客户机中的 MDDI\_Data0 和 MDDI\_Stb 的高速接收器置于低功率状态。在链路关闭分组的 CRC 之后的第 48 个 MDDI\_Stb 周期的上升沿之后的任意时刻，客户机将其 MDDI\_Data0 和 MDDI\_Stb 的高速接收器置于休眠。所建议的是，在链路关闭分组的 CRC 之后的第 64 个 MDDI\_Stb 周期的上升沿之前，客户机将其 MDDI\_Data0 和 MDDI\_Stb 的高速接收器置于休眠。

[0556] 通过禁用 MDDI\_Data0 和 MDDI\_Stb 驱动器并且把主机控制器置于低功率休眠状态，所述主机在点 C 或者步骤 C 进入低功率休眠状态。还可以根据需要把 MDDI\_Stb 驱动器设置为逻辑 0 电平（使用高阻抗偏置网络）或者设置为在休眠期间连续翻转。所述客户机也处于低功率电平的休眠状态。

[0557] 在某一段时间之后，所述主机通过启用 MDDI\_Data0 和 MDDI\_Stb 驱动器输出，在点 D 开始链路重启序列。所述主机把 MDDI\_Data0 驱动为逻辑 1 电平，并且把 MDDI\_Stb 驱动至逻辑 0 电平，且其持续时间与驱动器完全启用它们各自的输出所应占用的时间一样长。在这些输出达到所需要的逻辑电平之后，并且在驱动 MDDI\_Stb 上的脉冲之前，主机通常等待 200 纳秒左右。这样，客户机有时间准备接收。

[0558] 当主机驱动器被启用并且 MDDI\_Data0 被驱动为逻辑 1 电平时，主机开始翻转 MDDI\_Stb 且使其持续翻转 150 个 MDDI\_Stb 周期，如点 E 处所示。主机把 MDDI\_Data0 驱动至逻辑 0 电平且持续 50 个周期，如点 F 所示，并且在 MDDI\_Data0 处于逻辑 0 电平达 40 个 MDDI\_Stb 周期之久后，客户机开始寻找子帧报头分组。主机通过发送子帧报头分组开始在正向链路上传输数据，如点 G 所示。

[0559] 图 68B 中说明在没有竞争的情况下典型客户机启动的唤醒的处理步骤的实例，其中为了便于说明，也使用字母 A、B、C、D、E、F、G、H 和 I 来标记所述事件。如前所述，所述过程在点 A 处开始，此时主机向客户机发送链路关闭分组，通知它所述链路将转变为低功率状态。

[0560] 在 B 点，主机翻转 MDDI\_Stb 且持续翻转 64 个周期（根据系统设计需要），以便允许在停止 MDDI\_Stb 翻转之前完成客户机处理，这一信号用于停止客户机设备中的已恢复时钟。主机最初也把 MDDI\_Data0 设置为逻辑 0 电平，然后在 CRC 之后，在 16 至 48 个周期（通常包括输出禁用传播延迟）的范围内，禁用 MDDI\_Data0 输出。在 CRC 后的 48 个周期之后，并且在下一阶段 (C) 之前的某时，需要把客户机中的 MDDI\_Data0 和 MDDI\_Stb 的高速接收器置于低功率状态中。

[0561] 通过禁用 MDDI\_Data0 和 MDDI\_Stb 驱动器并且把主机控制器置于低功率休眠状态，所述主机在点 C 或者步骤 C 进入低功率休眠状态。还可以根据需要把 MDDI\_Stb 驱动器设置为逻辑 0 电平（使用高阻抗偏置网络）或者设置为在休眠期间继续翻转。所述客户机也处于低功率电平休眠状态。

[0562] 在某一段时间之后，客户机通过启用 MDDI\_Stb 接收器并且启用在 MDDI\_Stb 接收器中的一个偏移，在点 D 开始链路重启序列，以便确保在主机启用其 MDDI\_Stb 驱动之前，在客户机中所接收的 MDDI\_Stb 的版本的的状态是逻辑 0 电平。根据需要，可能需要客户机稍微超前于启用接收器时启用该偏移 (offset)，以便确保接收到有效的差分信号并且防止错误信号。客户机启用 MDDI\_Data0 驱动器，同时把 MDDI\_Data0 线路驱动至逻辑 1 电平。如果

启动偏移和标准 MDDI\_Data0 差分接收器的时间为 200 纳秒,则可以同时启用 MDDI\_Data0 和 MDDI\_Stb。

[0563] 在约 1 毫秒内,在点 E 处,主机识别来自客户机的服务请求脉冲,并且主机通过启用 MDDI\_Data0 和 MDDI\_Stb 驱动器输出来开始链路重启序列。所述主机把 MDDI\_Data0 驱动为逻辑 1 电平,并把 MDDI\_Stb 驱动至逻辑 0 电平,且其持续时间与驱动器启用它们各自的输出所应占用的时间一样长。在这些输出达到所需要的逻辑电平之后,并且在驱动 MDDI\_Stb 上的脉冲之前,主机通常等待 200 纳秒左右。这样,客户机就有时间准备接收。

[0564] 当主机驱动器被启用并且 MDDI\_Data0 被驱动为逻辑 1 电平时,主机在 MDDI\_Stb 上输出脉冲且持续 150 个 MDDI\_Stb 周期的持续时间,如点 F 处所示。当客户机识别出 MDDI\_Stb 上的第一个脉冲时,它禁用其 MDDI\_Stb 接收器中的偏移 (offset)。客户机继续把 MDDI\_Data0 驱动至逻辑 1 电平且持续 70 个 MDDI\_Stb 周期,并且在点 G 禁用其 MDDI\_Data0 驱动器。客户机继续把 MDDI\_Data0 驱动至逻辑 1 电平且持续 80 个额外的 MDDI\_Stb 脉冲,并且在点 H 将 MDDI\_Data0 驱动至逻辑零电平。

[0565] 如点 G 和 H 处所示,主机把 MDDI\_Data0 驱动至逻辑 0 电平且持续 50 个周期,并且客户机在 MDDI\_Data0 处于逻辑 0 电平达 40 个 MDDI\_Stb 周期后,客户机开始寻找子帧报头分组。在将 MDDI\_Stb 驱动达 50 个周期后,主机通过发送子帧报头分组开始在正向链路上传输数据,如点 I 所示。

[0566] 图 68C 中说明具有来自客户机的竞争,也就是客户机也希望唤醒所述链路的典型主机启动的唤醒的处理步骤的实例。为了便于说明,也使用字母 A、B、C、D、E、F、G、H 和 I 来标记所述事件。如前所述,所述过程在点 A 处开始,此时主机向客户机设备发送链路关闭分组,以便通知客户机所述链路将转变到低功率状态,然后进行到 B 点,在点 B 处 MDDI\_Stb 被翻转且持续翻转约 64 个周期(或者根据系统设计需要),以便允许由客户机执行的处理完成,然后进行到点 C,此时所述主机通过禁用 MDDI\_Data0 和 MDDI\_Stb 驱动器并且把主机控制器置于低功率休眠状态来进入低功率休眠状态。在某时间段之后,所述主机通过启用 MDDI\_Data0 和 MDDI\_Stb 驱动器输出在点 D 开始链路重启序列,并且开始翻转 MDDI\_Stb 且使其持续翻转 150 个 MDDI\_Stb 周期的持续时间,如点 E 处所示。

[0567] 在点 E 之后的多达 70 个 MDDI\_Stb 周期处,在这里为点 F,客户机还没有识别到主机正在把 MDDI\_Data0 驱动至逻辑 1 电平,因此客户机也把 MDDI\_Data0 驱动至逻辑 1 电平。出现这种情况是因为客户机需要请求服务,但是没有识别到其正在试图与之通信的主机已经开始链路重启序列。在点 G,客户机停止驱动 MDDI\_Data0,并且通过禁用其输出将其驱动器置于高阻状态。所述主机继续把 MDDI\_Data0 驱动至逻辑 1 电平且使其持续 80 个附加周期。

[0568] 主机把 MDDI\_Data0 驱动至逻辑 0 电平且使其持续 50 个周期,如点 H 所示,并且客户机在 MDDI\_Data0 处于逻辑 0 电平达 40 个 MDDI\_Stb 周期之久后,该客户机开始寻找子帧报头分组。主机通过发送子帧报头分组开始在正向链路上传输数据,如点 I 所示。

[0569] VI. 接口电气规范

[0570] 在一个示例性的实施例中,使用数据选通信号或者 DATA-STB 格式对非归零 (NRZ) 格式的数据进行编码,这样就将时钟信息嵌入数据和选通信号中。所述时钟可以在没有复杂的锁相环的情况下加以恢复。数据由双向差分链路承载,这通常是使用有线线路电缆来

实现的,当然如前所述也可以使用其它导线、印刷线路或者传送元件。选通信号 (STB) 是在仅由主机驱动的单向链路上承载的。每当存在背靠背 (back-to-back) 状态,0 或 1 时,即数据线或者信号上保持不变时,选通信号将值翻转 (0 或者 1)。

[0571] 图 40 中以图形形式示出了如何使用 DATA-STB(数据-选通)编码来传输如位“1110001011”的数据序列的实例。在图 40 中,DATA(数据)信号 4002 显示在信号时序图的顶行上,并且 STB(选通)信号 4004 显示在第二行上,根据适当的需要(公共起点)二者每次都对准。随着时间推移,当 DATA 线路 4002(信号)上出现的状态发生改变时,那么 STB 线路 4004(信号)保持先前状态,从而,DATA 信号的第一个‘1’状态与 STB 信号的第一个‘0’状态也就是它的起始值相互关联。然而,如果 DATA 信号的状态、电平不改变时,那么 STB 信号翻转至相对状态,即在本实例中为‘1’状态,如同图 40 中 DATA 提供又一‘1’值的情况那样。也就是说,每个位周期在 DATA 和 STB 之间有并且只有一个转变。因此,当 DATA 信号保持在‘1’时,STB 信号再次转变,此时为‘0’,并且在 DATA 信号的电平改变为‘0’时,STB 保持这一电平。当 DATA 信号保持在‘1’时,STB 信号翻转至相反状态,即当前实例中的‘1’,以及随着 DATA 信号改变或者保持电平或者值,STB 信号保持或者转变,等等。

[0572] 收到这些信号时,对 DATA 和 STB 信号执行异或 (XOR) 运算以便产生时钟信号 4006,该时钟信号显示在时序图的底部,以便与所需要的数据和选通信号相比较。图 41 中示出了一电路实例,该电路可用于在主机处根据输入数据生成 DATA 和 STB 输出或者信号,然后在客户机处根据 DATA 和 STB 信号恢复或者重获数据。

[0573] 在图 41 中,传输部分 4100 用来生成并且经由中间信号路径 4102 传输原始 DATA 和 STB 信号,而接收部分 4120 用来接收信号并且恢复数据。如图 41 所示,为了从主机向客户机传送数据,把 DATA 信号输入到两个 D 型触发器电路元件 4104 和 4106,并且输入时钟信号以便触发所述电路。然后,使用两个差分线路驱动器 4108 和 4110(电压模式)把两个触发电路的输出(Q)分别分为 MDDI\_Data0+,MDDI\_Data0- 和 MDDI\_Stb+,MDDI\_Stb- 的信号差分对。三输入同或 (XNOR) 门、电路或逻辑元件 4112 被连接以便接收 DATA 以及两个触发器的输出,并且生成用于向第二触发器提供数据输入的输出,这第二触发器则生成所述 MDDI\_Stb+,MDDI\_Stb- 信号。为方便起见,在 XNOR 门上画了一个反相泡泡,以便表明它有效地使生成选通信号的触发器的 Q 输出反相。

[0574] 在图 41 的接收部分 4120 中,由两个差分线路接收器 4122 和 4124 中的每一个接收 MDDI\_Data0+,MDDI\_Data0- 和 MDDI\_Stb+,MDDI\_Stb- 信号,差分线路接收器根据差分信号生成单端输出。然后,放大器的输出被输入至双输入异或 (XOR) 门、电路或逻辑元件 4126 的每一输入端,所述逻辑元件 4126 用于产生时钟信号。所述时钟信号用来触发两个 D 型触发器电路 4128 和 4130 中的每一个,所述 D 型触发器电路用于通过延迟元件 4132 接收 DATA 信号的延迟版本,所述 D 型触发器之一 (4128) 生成数据‘0’值,而另一个 (4130) 生成数据‘1’值。所述时钟还具有来自 XOR 逻辑的独立输出。由于时钟信息被分配给 DATA 和 STB 线路,所以状态之间的信号转变均慢于一半的时钟速率。由于使用 DATA 和 STB 信号的异或处理来再现时钟,所以与直接经由单根专用数据线发送时钟信号的情况相比,系统实际上容许在输入数据和时钟之间存在两倍的偏离量。

[0575] MDDI 数据对, MDDI\_Stb+ 和 MDDI\_Stb- 信号,采用差分模式工作,以便最大程度上免除噪声的负面影响。差分对中的每一个都是利用传送信号的电缆或者导线的特性阻抗来

并联端接 (parallel terminated) 的。通常,所有的并联终端电阻都驻留在客户机设备中。这近似于正向业务 (从主机向客户机发送数据) 的差分接收器,但其在电缆或其他导线或传送元件的驱动端部是用于反向业务的 (从客户机向主机发送数据)。对于反向业务,信号由客户机驱动,由主机处的高阻抗接收器反射,并在客户机处端接。这避免了对于将增加电流消耗的双终端电阻的需要。其也工作在大于电缆往复行程延迟的倒数的数据速率上。所述 MDDI\_Stb+ 和 MDDI\_Stb- 导线或信号只由主机驱动。

[0576] 图 42 中示出了作为本发明的 MDDI 接口一部分,可以用来实现传送信号的驱动器、接收器和终端电阻的元件的示例性配置。这一示例性接口使用低压检测,在这里是 200mV,具有小于 1 伏的电压摆幅和低功耗。每个信号对的驱动器具有一个差分电流输出。当接收 MDDI 分组时,该 MDDI\_Data 和 MDDI\_Stb 对使用传统的差分接收器且电压阈值为零伏。在休眠状态,驱动器输出被禁用,并且并联终端电阻将每个信号对上的电压上拉到零伏。在休眠期间,在 MDDI\_Data0 对上的专用接收器具有正 125mV 的偏移输入阈值,该阈值使得休眠线路接收器将未驱动信号对解释为逻辑零电平。

[0577] 有时,主机或客户机同时将差分对驱动到逻辑 1 电平或逻辑 0 电平,以便确保当数据流方向改变时 (从主机到客户机或从客户机到主机) 在该差分对上的逻辑电平有效。输出电压范围和输出规范仍然应满足驱动到相同逻辑电平的驱动输出。在一些系统中,还必须将小电流驱动到端接的差分对中,以在休眠期间的特定时刻和链路从休眠状态被唤醒时创建的小偏移电压。在这些情况下,启用的偏移电流偏置驱动的电平表示为:  $I_{\text{ESD-and-Rx}}$  - 内部 ESD 二极管和差分接收器输入,其中  $I_{\text{ESD-and-Rx}} \leq 1 \mu\text{A}$ ;  $I_{\text{Tx-Hi-Z}}$  - 在高阻态的差分驱动器输出,其中  $I_{\text{Tx-Hi-Z}} \leq 1 \mu\text{A}$ ;  $I_{\text{external-ESD}}$  - 通过外部 ESD 保护二极管的泄漏,其中通常  $I_{\text{external-ESD}} \leq 3 \mu\text{A}$ 。

[0578] 图 47 示出了这些泄漏电流中的每一种。在所有泄漏都同时出现时,上拉和下拉电路必须达到上述最差情况的泄漏条件下的最小的差分电压。对于没有外部 ESD 保护二极管的内部模式,总泄漏是  $\leq 4 \mu\text{A}$ ,对于具有外部 ESD 保护的外部模式,总泄漏是  $\leq 10 \mu\text{A}$ 。

[0579] 表 VIIa-VIIId 描述了一个示例性实施例,差分线路驱动器和线路接收器的电气参数和特性。在功能上,驱动器将输入端上的逻辑电平直接传送到正输出端,并且将所述输入端的反相传送到负输出端。从输入端到输出端的延迟与以差分方式驱动的差分线路很好地匹配。在大多数实现中,输出端上的电压摆幅小于输入端上的摆幅,从而使得功耗和电磁辐射最小化。在一个实施例中,最小电压摆幅大约为 0.5V。然而,其他值也是可用的,正如本领域技术人员所公知的那样,并且发明人考虑到在一些实施例中根据设计约束条件还可以是更小的值。

[0580] 差分线路接收器具有与高速电压比较器相同的特性。在图 41 中,没有泡泡的输入端是正输入端,而有泡泡的输入端是负输入端。如果  $(V_{\text{输入}+}) - (V_{\text{输入}-})$  大于 0,那么输出端是逻辑 1。用于描述它的另一方式是具有非常大 (事实上无限的) 增益,输出端钳制在逻辑 0 和 1 电压电平的差分放大器。

[0581] 不同对之间的延迟偏离应被最小化,以便以最高的速度来操作差分传输系统。

[0582] 表 VIIa

[0583] 主机发射机电气规范

[0584]

参数	描述	最小值	最大值	单位
$V_{\text{output-range}}$	相对于主机地所允许的主机驱动器输出电压范围	0.35	1.60	V
$I_{\text{OD+}}$	驱动器差分输出高电流 (当驱动终端传输线时)	2.5	4.5	mA
$I_{\text{OD-}}$	驱动器差分输出低电压 (当驱动终端传输线时)	-4.5	-2.5	mA
$T_{\text{Rise-Fall}}$	在差分模式下测量的驱动器输出的升降时间(在20%和80%幅度间)	425	注意事项 1	psec
$T_{\text{skew-pair}}$	同一差分对的正负输出间的偏离(对内偏离)		125	psec
$T_{\text{Differential-skew}}$	一个差分对和另一差分对之间的峰值延迟偏离		参见上面	psec
$T_{\text{A}}$	到中间交叉点的抖动、位边界	0	$T_{\text{B-283}}$	psec
$T_{\text{B-TP0-DRVR}}$	到最小输出电平的抖动、位边界	0	参见上面	psec

[0585] 注意事项 1:最大升降时间要么是在一个差分对上发送一位的间隔的 30%,要么是 100nsec,取二者中的较小者。

[0586] 表 VIIb

[0587] 客户机发射机电气规范

[0588]

参数	描述	最小值	最大值	单位
$V_{\text{output-Range-Ext}}$	相对于客户机地所允许的客户机驱动器输出电压范围（外部模式）	0	1.25	V
$V_{\text{output-Range-Int}}$	相对于客户机地所允许的客户机驱动器输出电压范围（内部模式）	0.35	1.60	V
$I_{\text{OD+}}$	驱动器差分输出高电压（当驱动主机和客户机中存在的上拉和下拉电路等效电路时）	2.5	4.5	mA
$I_{\text{OD-}}$	驱动器差分输出低电压（当驱动主机和客户机中存在的上拉和下拉电路等效电路时）	-4.5	-2.5	mA
$T_{\text{Rise-Fall}}$	在差分模式下测量的驱动器输出的升降时间（在 20% 和 80% 幅度间）	425	注 意 事 项 1	psec
$T_{\text{skew-pair}}$	同一差分对的正负输出间的偏离（对内偏离）		125	psec
$T_{\text{Differential-skew}}$	一个差分对和另一差分对之间的峰值延迟偏离		参 见 上 面	psec
$T_{\text{A}}$	到中间交叉点的抖动、位边界	0	$T_{\text{B}}-283$	psec
$T_{\text{B-TP4-DRVR}}$	到最小输出电平的抖动、位边界	0	参 见 上 面	psec

[0589] 注意事项 1：最大升降时间要么是在一个差分对上发送一位的间隔的 30%，要么是 100nsec，取二者中的较小者。

[0590] 表 VIIc

[0591] 客户机接收器电气规范

[0592]

参数	描述	最小值	典型值	最大值	单位
$V_{IT+}$	接收器差分输入高门限电压		0	50	mV
$V_{IT-}$	接收器差分输入低门限电压	-50	0		mV
$V_{IT+}$	接收器差分输入高门限电压（偏移用于休眠唤醒）		125	175	mV
$V_{IT-}$	接收器差分输入低门限电压（偏移用于休眠唤醒）	75	125		mV
$V_{Input-Range}$	相对于客户机地所允许的接收器输入电压范围	0		1.65	V
$R_{term}$	并行终端电阻值	98	100	102	$\Omega$
$I_{in}$	输入漏极电流	-10		10	$\mu A$
$C_{pad}$	垫到客户机地的电容（注意事项 1）			5	pF
$C_{diff}$	差分对的两个信号之间的电容（注意事项 1）			1	pF
$T_{skew-pair-INT}$	同一差分对的差分接收器的正负输入间的差分接收器导致的偏离（对内偏离）。内部模式			250	psec
$T_{skew-pair-EXT}$	对内偏离，外部模式			50	psec

[0593]

$T_{\text{Differential-Skew}}$	一个差分对和另一差分对之间的峰值延迟偏离			参见上面	
$T_A$	到中间交叉点的抖动、位边界			$T_B-38.5$	psec
$T_{B-TP4-DRVR-INT}$	到最小输出电平的抖动、位边界（内部模式）	0		参见上面	psec
$T_{B-TP4-DRVR-EXT}$	到最小输出电平的抖动、位边界（外部模式）	0		参见上面	psec

[0594] 表 VIId

[0595] 主机接收器电气规范

[0596]

参数	描述	最小值	典型值	最大值	单位
$V_{IT+}$	接收器差分输入高门限电压（无偏移）		0	50	mV
$V_{IT-}$	接收器差分输入低门限电压（无偏移）	-50	0		mV
$V_{IT+}$	接收器差分输入高门限电压（休眠唤醒的偏移）		125	175	mV
$V_{IT-}$	接收器差分输入低门限电压（休眠唤醒的偏移）	75	125		mV
$V_{\text{Input-Range}}$	相对于主机地所允许的接收器输入电压范围	0		1.65	V
$I_{in}$	输入漏极电流（排除掉休眠偏压）	-10		10	$\mu A$
$C_{\text{pad}}$	垫到主机地的电容			5	pF
$C_{\text{diff}}$	差分对的两个信号之间的电容			1	pF
$T_{\text{skew-pair}}$	同一差分对的差分接收			250	psec

[0597]



	器的正负输入间的差分接收器导致的偏离(对内偏离)				
$T_{\text{skew-pair-EXT}}$	对内偏离, 外部模式			50	psec
$T_A$	到中间交叉点的抖动、位边界			$T_B-38.5$	psec
$T_{B-TP0-DRVR-INT}$	到最小输出电平的抖动、位边界(外部模式)			参见上面	psec
$T_{B-TP0-DRVR-EXT}$	到最小输出电平的抖动、位边界(外部模式)			参见上面	psec

[0598] 在图 42 中, 主机控制器 4202 和客户机或者显示器控制器 4204 经由通信链路 4206 传送分组。主机控制器采用一系列三个驱动器 4210、4212 和 4214 来接收待传送的主机 DATA 和 STB 信号, 并且接收待传送的客户机数据信号, 而客户机采用三个驱动器 4230、4232 和 4234。负责主机 DATA 传递的驱动器 (4212) 采用使能信号输入端以便通常只有当需要从主机向客户机传送时, 才允许激活通信链路。由于 STB 信号是作为数据传送的一部分来形成的, 所以对于该驱动器 (4212) 而言没有采用附加使能信号。客户机 DATA 和 STB 接收器 (4132、4230) 驱动器中每一个的输入端都具有分别跨接 (pace accross) 在其上的终端阻抗或电阻 4218 和 4220。客户机控制器中的驱动器 4234 用于准备从客户机传送到主机的数据信号, 其中输入侧的驱动器 4214 处理这些数据。

[0599] 专用接收器 (驱动器) 4216 和 4236 与 DATA 线路耦合或连接, 并产生或使用先前所讨论的 125mV 的电压偏移, 作为其他地方所讨论的休眠控制的一部分。该偏移使得休眠线路接收器将未驱动信号对解释为逻辑 0 电平。

[0600] 上述驱动器和阻抗可以作为分立元件或者作为电路模块的一部分来形成, 或者作为专用集成电路 (ASIC) 来形成, 该专用集成电路充当更节省成本的编码器或者解码器方案。

[0601] 可以很容易看出: 使用标记为 HOST\_Pwr 和 HOST\_Gnd 的信号经由一对导线从主机设备向客户机设备或者显示器传送功率。信号的 HOST\_Gnd 部分充当客户机设备的基准地和电源返回路径或者信号。HOST\_Pwr 信号充当客户机设备的电源, 该电源由主机设备驱动。在示例性的配置中, 对于低功率应用, 所述客户机设备被允许使用最多 500mA。可以从便携式电源提供 HOST\_Pwr 信号, 例如但不局限于驻留在主机设备的锂离子类型的电池或者电池组, 并且相对于 HOST\_Gnd 可以在 3.2 到 4.3 伏的范围内。

[0602] VII. 时序特性

[0603] A. 概述

[0604] 图 43a、43b 和 43c 分别示出的步骤和信号电平由主机或客户机使用, 以进入休眠状态 (不请求、预期或需要服务) 和为客户机获得来自主机的服务。在图 43a、43b 和 43c

中,所示出的第一部分信号示出了从主机传送的链路关闭分组,然后使用高阻抗偏置电路把数据线驱动为逻辑 0 状态。已经禁用其驱动器的客户机或者主机不传送数据。由于在链路关闭分组期间 MDDI\_Stb 是激活的,所以可以在底部看见 MDDI\_Stb 信号线路的一系列选通脉冲。当主机把偏置电路和逻辑驱动至零时,一旦这一分组结束,逻辑电平就变成 0。这种情况代表从主机传送的最后信号或者来自主机的服务结束,并且这可能发生在过去的任意时刻,而且这被包括以便示出服务在先中止以及在服务开始前的信号状态。如果需要的话,可以发送这样的信号,只是为了把通信链路重置为适当的状态,而无需一个已经由这一主机设备进行的‘已知’的在先通信。

[0605] 如图 43a 所示,并且如上针对链路关闭分组所讨论的那样,在低功率休眠状态下,在链路关闭分组中的全零字段的最后一位后的第 16 个至第 48 个 MDDI\_Stb 周期或脉冲后开始, MDDI\_Data0 驱动器被禁用而进入高阻抗状态。对于类型 2、类型 3 或类型 4 链路,在 MDDI\_Data0 驱动器被禁用的同时, MDDI\_Data1 至 MDDI\_DataPwr7 信号也被置入高阻抗状态。在全零字段的定义中已经描述过,在链路关闭分组的 CRC 字段的 MSB 后的 64 个周期(或者,根据系统设计所希望的那样), MDDI\_Stb 翻转,从而能够完成客户机的处理和客户机控制器中的有序关闭。一个周期是一次低到高转换,后面接着一次高到低转换,或者是一次高到低转换,后面接着一次低到高转换。在发送全零字段后,主机中的 MDDI\_Data0 和 MDDI\_Stb 驱动器被禁用,主机进入低功率休眠状态。在一段时间后,启用 MDDI\_Data0 和 MDDI\_Stb 线路或驱动器输出,主机开始如图 43b 和 43c 所示的链路重启序列,并开始翻转 MDDI\_Stb,作为主机或客户机发起的唤醒请求的一部分。

[0606] 如图 43b 所示,在用于 MDDI\_Data0 和 MDDI\_Stb 的驱动器的输出信号被禁用的情况下经过一段时间以后,主机通过将其 MDDI\_Stb 驱动器启用时间段  $t_{\text{stb-data-enbl}}$ ,启动服务或从休眠中唤醒,在此时间段内,线路被驱动为逻辑零电平,直到它被完全启用然后启用其 MDDI\_Data0 驱动器为止。在 MDDI\_Data0 达到高或逻辑 1 电平后(经过时间段  $t_{\text{client-startup}}$  发生),主机将 MDDI\_Stb 维持在逻辑零电平。在时间段  $t_{\text{client-startup}}$  结束时,主机翻转 MDDI\_Stb 信号或线路。在时间段  $t_{\text{restart-high}}$  内,主机将 MDDI\_Data0 线路驱动为高,即逻辑 1 电平,而客户机不驱动 MDDI\_Data0,然后,在时间段  $t_{\text{restart-low}}$  内,将 MDDI\_Data0 线路驱动为低,即逻辑零电平。此后,第一前向业务与子帧报头分组一起开始,然后,传输前向业务分组。在时段  $t_{\text{restart-low}}$  和后续的子帧报头分组期间, MDDI\_Stb 信号是激活的。

[0607] 如图 43c 所示,在用于 MDDI\_Data0 和 MDDI\_Stb 的驱动器的输出信号被禁用的情况下经过一段时间后,通过在主机启用其 MDDI\_Stb 驱动器之前在时间段  $t_{\text{stb-data-enbl}}$  内启用 MDDI\_Stb 接收器中的偏移或输出信号,客户机发出服务请求或从休眠中唤醒。然后,客户机启用其 MDDI\_Data0 驱动器持续时间段  $t_{\text{host-detect}}$ ,在此期间,在主机开始 MDDI\_Stb 翻转之前,线路被驱动至逻辑零电平。

[0608] 在主机检测到所述请求之前,可能会经过或者需要一定量的时间,这段时间表示为“ $t_{\text{host-detect}}$ ”,此后,主机将 MDDI\_Stb 保持在逻辑零电平达时间段  $t_{\text{stb-startup}}$  来做出响应,然后,主机在时间段  $t_{\text{restart-high}}$  内将 MDDI\_Data0 驱动到逻辑 1 或高电平,从而开始用链路启动序列翻转 MDDI\_Stb。当客户机发现 MDDI\_Stb 上的第一脉冲时,它禁用其 MDDI\_Stb 接收器中的偏移。客户机继续将 MDDI\_Data0 驱动到逻辑 1 电平或时间段  $t_{\text{client-detect}}$ ,直到它检测到主机驱动该线路为止。在这一刻,客户机不再维持(de-assert)所述请求,并且禁用其

MDDI\_Data0 驱动器,因此来自客户机的输出也进入 0 逻辑电平,主机驱动 MDDI\_Data0。如前所述,在时间段“ $t_{\text{restart-high}}$ ”内,主机继续把 MDDI\_Data0 驱动为逻辑‘1’电平,此后,主机把 MDDI\_Data0 线路驱动至低并且持续“ $t_{\text{restart-low}}$ ”时间段,此后,第一正向业务以子帧报头分组开始。在  $t_{\text{restart-low}}$  时间段期间以及后续子帧报头分组期间,所述 MDDI\_Stb 信号是处于激活状态的。

[0609] 表 VIII 示出了上述各种时间段的长度的代表时间或处理时间段,以及示例性的最小和最大数据速率之间的关系,其中  $t_{\text{bit}} = \frac{1}{\text{Link\_Data\_Rate}}$ ,其中 Link\_Data\_Rate 是单个数据对的位速率。

[0610] 表 VIII

[0611]

参数	描述	最小值	典型值	最大值	单位
$1/t_{\text{BIT-min-perf}}$	最低性能设备的链路数据速率	0.001		1.1	Mbps
$1/t_{\text{BIT-max-perf}}$	外部设备的最大链路数据速率范围	0.001		400	Mbps
$1/t_{\text{BIT-max-perf}}$	内部设备的最大链路数据速率范围	0.001		550	Mbps
	反向链路数据速率	0.0005		50	Mbps
$t_{\text{BIT}}$	外部模式下前向链路数据位的周期	2.5		$10^6$	nsec
$t_{\text{BIT}}$	内部模式下前向链路数据位的周期	1.8		$10^6$	nsec
$t_{\text{restart-high}}$	主机链路重启高脉冲期间	140	150	160	Stb 时钟
$t_{\text{restart-low}}$	主机链路重启低脉冲期间	50	50	50	Stb 时钟
$t_{\text{stb-data-enabl}}$	MDDI_Stb 完全被启用到 MDDI_Data0 被启用的链路	0			微秒

[0612]

	重启序列				
$t_{\text{client-startup}}$	主机在 MDDI_Data0 达到逻辑高电平之后保持 MDDI_Stb 在逻辑 0 电平的时间	200			纳秒
$t_{\text{host-detect}}$	从 MDDI_Data0 高到 MDDI_Stb 翻转的时间	0		1000	微秒
$t_{\text{client-detect}}$	客户机检测 MDDI_Data0 为逻辑高电平性能设备的时间	60		80	stb 时钟
$t_{\text{stb-startup}}$	在主机开始 MDDI_Stb 翻转之前保持 MDDI_Stb 为逻辑 0 电平的时间	100			纳秒

[0613] 本领域技术人员将易于理解,图 41 和 42 中举例说明的各个元件的功能是众所周知的,并且通过图 43a、43b 和 43c 中的时序图确认图 42 中元件的功能。把图 42 中示出的串联的终端电阻和休眠电阻的细节从图 41 省略,这是因为对于描述如何执行数据-选通编码并且根据它恢复时钟而言,所述信息是不必要的。

[0614] B. 数据-选通时序正向链路

[0615] 表 IX-1 中示出了在正向链路上从主机驱动器输出端传送数据的切换特性。表 IX 用表格形式对照发生的某些信号转变的典型时间给出了所需要的最小和最大值。例如,从数据值(‘0’或者‘1’的输出)的开始到结束发生转变(称为  $t_{\text{tdd-(host-output)}}$  的 Data0 到 Data0 转变)的典型时间长度是  $t_{\text{tbit}}$ ,而最小时间大约为  $t_{\text{tbit}}-0.5$  纳秒,而最大时间大约为  $t_{\text{tbit}}+0.5$  纳秒。图 44 中说明 Data0、其它数据线(DataX)和选通线路(Stb)上的转变之间的相对间隔,其中示出了 Data0 至选通、选通至选通、选通至 Data0、Data0 至非 Data0、非 Data0 至非 Data0、非 Data0 至选通以及选通至非 Data0 的转变间隔,将其分别称为  $t_{\text{tds-(host-output)}}$ 、 $t_{\text{tss-(host-output)}}$ 、 $t_{\text{tsd-(host-output)}}$ 、 $t_{\text{tddx-(host-output)}}$ 、 $t_{\text{tdxdx-(host-output)}}$ 、 $t_{\text{tdxs-(host-output)}}$  以及  $t_{\text{tsdx-(host-output)}}$ 。

[0616] 表 IX-1

[0617]

参数	描述	最小值	典型值	最大值	单位
----	----	-----	-----	-----	----

[0618]

$t_{\text{tdd-(host-output)}}$	Data0 至 Data0 转变	$t_{\text{tbit}}-0.5$	$t_{\text{tbit}}$	$t_{\text{tbit}}+0.5$	纳秒
$t_{\text{tds-(host-output)}}$	Data0 至选通转变	$t_{\text{tbit}}-0.8$	$t_{\text{tbit}}$	$t_{\text{tbit}}+0.8$	纳秒
$t_{\text{tss-(host-output)}}$	选通至选通转变	$t_{\text{tbit}}-0.5$	$t_{\text{tbit}}$	$t_{\text{tbit}}+0.5$	纳秒
$t_{\text{tsd-(host-output)}}$	选通至 Data0 转变	$t_{\text{tbit}}-0.8$	$t_{\text{tbit}}$	$t_{\text{tbit}}+0.8$	纳秒
$t_{\text{tddx-(host-output)}}$	Data0 至非 Data0 转变		$t_{\text{tbit}}$		纳秒

$t_{tdxdx}(\text{host-output})$	非 Data0 至非 Data0 转变	$t_{tbit}-0.5$	$t_{tbit}$	$t_{tbit}+0.5$	纳秒
$t_{tdxs}(\text{host-output})$	非 Data0 至选通转变		$t_{tbit}$		纳秒
$t_{tsdx}(\text{host-output})$	选通至非 Data0 转变		$t_{tbit}$		纳秒

[0619] 表 IX-2 中示出了用于在正向链路上传送同一信号的客户机接收器输入的典型 MDDI 时序要求。由于除了时间延迟之外讨论的是相同的信号,所以不需要新的附图来举例说明各个标记的信号特性或者意义,正如本领域技术人员将理解的那样。

[0620] 表 IX-2

[0621]

参数	描述	最小值	典型值	最大值	单位
$t_{tdd}(\text{display-input})$	Data0 至 Data0 转变	$t_{tbit}-1.0$	$t_{tbit}$	$t_{tbit}+1.0$	纳秒
$t_{tds}(\text{display-input})$	Data0 至选通转变	$t_{tbit}-1.5$	$t_{tbit}$	$t_{tbit}+1.5$	纳秒
$t_{tss}(\text{display-input})$	选通至选通转变	$t_{tbit}-1.0$	$t_{tbit}$	$t_{tbit}+1.0$	纳秒
$t_{tsd}(\text{display-input})$	选通至 Data0 转变	$t_{tbit}-1.5$	$t_{tbit}$	$t_{tbit}+1.5$	纳秒
$t_{tddx}(\text{host-output})$	Data0 至非 Data0 转变		$t_{tbit}$		纳秒
$t_{tdxdx}(\text{host-output})$	非 Data0 至非 Data0 转变		$t_{tbit}$		纳秒
$t_{tdxs}(\text{host-output})$	非 Data0 至选通转变		$t_{tbit}$		纳秒
$t_{tsdx}(\text{host-output})$	选通至非 Data0 转变		$t_{tbit}$		纳秒

[0622] 图 45 和 46 分别说明当主机禁用或启用主驱动器时作为响应存在的延迟。在主机传送某些分组的情况下,诸如反向链路封装分组或者往返行程延迟测量分组,主机在所需分组被传送之后,禁用所述线路驱动器,所需分组例如为图 45 中举例说明的已经传送的参数 CRC、选通对准以及全零分组。然而,如图 45 所示,所述线路的状态不必即刻从‘0’切换至所需要的高位值,尽管这是利用现有某些控制或电路元件就能够实现的,而是要经历被称为主机驱动器禁用延迟周期的时间段来做出响应。虽然这种切换事实上可以立即进行,从而使此时间段在长度上是 0 纳秒 (nsec),但是它也可以很容易地延续更长的周期,即作为需要的最大周期长度的 10 纳秒,这是在保护时间 1 或者转向 1 分组周期期间进行的。

[0623] 参见图 46,当主机驱动器被启用以便传送诸如反向链路封装分组或者往返行程延迟测量分组这样的分组时,人们会看到信号电平发生改变。在这里,在保护时间 2 或者转向 2 分组周期之后,主机驱动器被启用并且开始驱动电平(在这里是‘0’),在经历称为主机驱动器启用延迟周期的时间段后接近或者达到该值,该主机驱动器启用延迟周期是在第一分组被发送之前,在驱动器重新启用周期期间出现的。

[0624] 客户机设备的驱动器和信号传送出现类似的过程,这里的客户机为显示器。下面,

表 X 中示出了这些周期的长度的通用准则以及它们各自的关系。

[0625] 表 X

[0626]

描述	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

[0627] C. 主机和客户机输出启用和禁用时间

[0628] 图 48 中示出了相对于反向链路封装分组结构和周期的主机和客户机输出启用和禁用时间或操作的切换特性和相对时序关系。驱动器输出功能或操作标为： $t_{\text{host-enable}}$ ，代表主机输出启用时间； $t_{\text{host-disable}}$ ，代表主机输出禁用时间； $t_{\text{client-enable}}$ ，代表客户机输出启用时间； $t_{\text{client-disable}}$ ，代表客户机输出禁用时间。下面讨论特定信号转换的典型时间。这些操作的最小周期将是 0 纳秒，典型值或最大值是根据采用该接口的系统设计而确定的，量级可能是 8 纳秒或更多。

[0629] 下面的表 XI 示出了这些周期（主机和客户机启用 / 禁用时间）的长度及其各自关系的一般指导方针。

[0630]

参数	描述	最小值	典型值	最大值	单位
$t_{\text{host-enable}}$	主机输出启用时间	0		$24 \cdot t_{\text{BIT}}$	纳秒
$t_{\text{host-disable}}$	主机输出禁用时间， 转向 1 字段的全长	0		$24 \cdot t_{\text{BIT}}$	纳秒
$t_{\text{client-enable}}$	客户机输出启用时间， 转向 1 字段的全长	0		$24 \cdot t_{\text{BIT}}$	纳秒
$t_{\text{client-disable}}$	从转向 2 字段最后一位 结束测量的客户机 输出禁用时间	0		$24 \cdot t_{\text{BIT}}$	纳秒

[0631] VIII. 链路控制（链路控制器操作）的实现方式

[0632] A. 状态机分组处理器

[0633] 经由 MDDI 链路传送的分组通常以大约 300Mbps 或更大的速率，如 400Mbps，被非常快速地传递，当然需要的时候肯定也能支持更低的速率。对于目前可以购买到的（经济

的)通用微处理器等等而言,这种类型的总线或者传送链路的速度太高以至无法控制。因此,用于实现这种类型的信号传送的实际实现方式在于利用可编程状态机来解析输入分组流以便产生被传送或被重定向至它们应该去的适当的音频-视频子系统的分组。这种设备是公知的,并且使用通常专用于有限数目的操作、功能或者状态的电路,以便实现所需要的高速度或者超高速操作。

[0634] 可以将通用控制器、处理器或者处理元件用于更适当地作用于或者操纵具有较低速度要求的某些信息,诸如控制或者状态分组。收到那些分组(控制、状态或者其他预定义分组)时,状态机应该让它们经由数据缓冲器或者类似处理元件传递至通用处理器,从而能够对所述分组进行操作以便提供所需要的结果(效果),同时音频和视觉分组被传送到其适当目的地以便进行动作。如果将来,制造出来微处理器或者其他通用控制器、处理器或者处理元件能够实现更高数据速率的处理能力,那么还可以使用这种设备的软件控制来实现下面讨论的状态或者状态机,所述设备通常作为诸如储存在存储元件或者介质上的程序。

[0635] 在一些实施例中,所述通用处理器功能可以利用处理能力或者多余的周期来实现,所述处理能力或者多余的周期可以从计算机应用中的微处理器(CPU)或者控制器、处理器、数字信号处理器(DSP)、专用电路或者无线设备中发现的ASIC得到,其实现的方式非常类似于某些调制解调器或者图像处理器利用计算机中发现的CPU的处理能力来执行某些功能,并且减少硬件的复杂性和成本。然而,这种周期的共享或者使用会给处理速度、时序或者这种元件的整个操作带来消极影响,因此在许多应用中,专用电路或元件更适合于这种通用处理。

[0636] 为了能够在显示器(微型显示器)上观看图像数据,或者可靠地接收由主机设备发送的所有分组,所述客户机信号处理与正向链路通道时序同步。也就是说,到达客户机和客户机电路的信号在时间上基本上要同步,以便正确进行信号处理。图49中说明可以实现这种同步的信号处理步骤或者方法实现的状态的高级图表。在图49中示出用于状态机4900的可能正向链路同步“状态”被分类为一个“异步帧状态”4904,两个“获得同步状态”4902和4906以及三个“同步中状态”4908、4910和4912。

[0637] 如开始步骤或者状态4902所示,诸如呈现设备之类的显示器或者客户机以预选“非同步(no sync)”状态开始,并且搜索检测到的第一子帧报头分组中的唯一字。应当注意,这种非同步状态表示选择类型1接口时的最小通信设置或者“后退(fall-back)”设置。当在搜索期间发现所述唯一字时,客户机保存子帧长度字段。在处理这一第一帧时不对CRC位进行校验,或者直到获得同步为止才校验。如果这一子帧长度是零,那么同步状态处理因此继续进行至状态4904,在这里标记为“异步帧(async frame)”状态,这表明还没有实现同步。在图49中,在处理中把这一步骤标记为具有cond 3,或者条件3。另外,如果帧长大于零,那么同步状态处理进行到状态4906,其中接口状态被设定为“发现一个同步帧”。在图49中,把处理中的这一步骤标记为具有cond 5,或者条件5。另外,如果对于帧长度大于零的帧,状态机看到帧报头分组并且确定具有良好的CRC,那么处理进行到“发现一个同步帧”状态。在图49中,将其标记为符合cond6,或者条件6。

[0638] 在系统处于除“非同步”之外的状态的每一种情况下,当确定了子帧报头分组有良好的CRC结果,接口状态改变为“同步中”状态4908。在图49中,把处理中的这一步骤标记

为符合 cond 1, 或者条件 1。另一方面, 如果任何分组中的 CRC 不正确, 那么同步状态处理继续进行或者返回到“非同步帧”状态的接口状态 4902。在图 49 的状态图中, 将处理中的这一部分标记为符合 cond 2, 或者条件 2。

#### [0639] B. 同步获得时间

[0640] 所述接口可以被配置成在判定失去同步之前允许特定数量的“同步错误”, 并且返回到“非同步帧”状态。在图 49 中, 一旦状态机已经达到“同步中状态”并且未发现错误, 那么它将连续地遇到 cond 1 的结果, 并且保持在“同步中”状态。然而, 一旦检测到一个 cond 2 结果, 那么处理过程把状态改变为“一个同步错误 (one-sync-error)”状态 4910。在这一刻, 如果处理过程检测到又一 cond 1 的结果, 那么状态机返回到“同步中”状态, 否则它遇到又一 cond 2 的结果, 并且移动到“两个同步错误”状态 4912。此外, 如果 cond 1 发生, 那么处理把状态机返回至“同步中”状态。否则, 遇到又一 cond 2, 并且状态机返回到“非同步”状态。还可以理解的是, 如果接口遇到“链路关闭分组”, 那么这将让链路终止数据传送并且返回到“非同步帧”状态, 就像没有什么可以与之同步似的, 在图 49 所示的状态图中, 将其称为符合 cond 4。

[0641] 应该理解的是, 有可能存在唯一字的重复“错误副本 (false copy)”, 它可能出现在子帧内的某个固定位置处。在那些情况下, 状态机极其不可能与这个子帧同步, 这是因为进行处理时为了使 MDD 处理进行到“同步中”状态, 子帧报头分组上的 CRC 也必须是有效的。

[0642] 子帧报头分组中的子帧长度可以被设置为零, 以表明在链路关闭, 并且将 MDD 接口设置在或者配置为空闲的休眠状态之前, 主机只传输一个子帧。在这种情况下, 客户机在检测到子帧报头分组之后必须立即经由正向链路接收分组, 这是因为在链路转变至空闲状态以前只发送了一个子帧。在正常的或者典型的操作中, 子帧长度是非零的, 并且客户机只处理正向链路分组, 而所述接口处于在图 49 中统一显示为“同步中”状态的那些状态。

[0643] 外部模式的客户机设备可连接到主机, 而主机已正在发送正向链路序列。在这种情况下, 客户机必须同步到主机。客户机同步到正向链路信号所需的时间随着子帧大小和正向链路数据速率而改变。当子帧尺寸较大时, 在正向链路中检测到作为所述随机数据或者更多随机数据一部分的唯一字的“错误副本”的可能性较大。同时, 当正向链路数据速率较慢时, 从错误检测中恢复的能力更低, 而执行这一操作所花费的时间更长。

[0644] 对于一个或多个实施例而言, 所推荐的或理解的是, MDDI 主机应当执行一些附加步骤, 以确保在它停止前向链路传输而进入低功率模式或完全关闭链路之前 MDDI 反向链路是稳定的。

[0645] 可能会出现这样一个问题: 如果主机使用的往返行程延迟的测量值不正确, 这可能会导致后面从客户机接收所有反向数据传输流失败, 即便前向链路看起来是良好的。当客户机与前向链路不同步时, 或者, 由于极端的环境温度变化导致差分驱动器和接收器的传播时延产生相应大变化而影响往返行程延迟, 如果主机发送往返行程延迟测量分组, 就会发生这种现象。间歇性的线缆或接头接触故障也可能使客户机暂时失去同步然后再次获得同步, 在此期间, 它可能收不到往返行程延迟测量分组。后续的反向链路分组无法被主机正确地解码。

[0646] 可能发生的另一类问题是: 如果客户机暂时失去同步, 那么, 在客户机能够重新获取同步之前, 主机发送链路关闭分组。在客户机无法进入休眠状态时, 主机将处于休眠状



态,因为它没有收到链路关闭分组和没有时钟,因为链路处于休眠状态。

[0647] 可用于克服这些问题的一种技术或实施例是:在将链路置入休眠状态前,使主机确保客户机与前向链路同步。如果 MDDI 主机无法做到这一点或者没有这样的机会,例如,当其没电或由于在工作期间出现线缆、导体或接头分开、断开或断连接而造成链路中断或故障时,那么,主机在开始往返行程延迟测量过程或发送反向链路封装分组之前应当先尽量确保客户机是同步的。

[0648] 主机可以观察客户机发送的客户机请求和状态分组中的 CRC 错误计数字段,以确定前向链路的完整性。主机从客户机请求该分组。但是,如果有主要的链路故障或中断,该请求将很可能得不到响应,因为客户机无法正确地对该分组进行解码,或者可能根本无法收到它。使用反向链路封装分组中发送的客户机请求和状态分组对 CRC 错误计数的请求充当第一完整性检查,相当于第一道防线。此外,主机可以发送一个往返行程延迟测量分组,以确认关于客户机失去同步的假设是否有效。如果客户机不对往返行程延迟测量分组做出响应,主机可以得出客户机失去同步的结论,然后就可以启动使其恢复同步的处理。

[0649] 一旦主机得出主机很可能与前向链路失去同步的结论,它就在试图发送填充符分组之外的任何分组之前等待,直到下一子帧报头。这样做的目的在于,使客户机有足够的时间去检测或寻找子帧报头分组中包含的唯一字。此后,主机可以假设,客户机已经自身复位,因为它没有在正确的位置发现唯一字。在这一刻,主机可以在子帧报头分组后面跟一个往返行程延迟测量分组。如果客户机仍不对往返行程延迟测量分组做出正确响应,则主机可以重复执行再同步过程。正确的响应是,客户机在往返行程延迟测量分组中,将指定的序列回送给主机。如果没有收到该序列,接收反向链路封装分组中的反向数据将失败。这种性质的继续失败可能表明一些其他系统错误,这必须以其他方式解决,故在此时刻不是链路同步的一部分。

[0650] 但是,如果在成功的往返行程延迟测量分组中主机仍看到被破坏的数据,或在反向链路封装分组中没有响应,它应当通过重新发送一个往返行程延迟测量分组,来确认反向数据采样是正确的。如果数次尝试之后这还不成功,对于一个实施例,所建议的是,主机通过增加反向速率除数值,来降低反向数据速率。

[0651] 在将 MDDI 链路置入休眠状态之前,主机应当执行上述的链路故障检测和链路再同步步骤。这通常可以确保,在链路稍后重新启动时执行的往返行程延迟测量分组是成功的。如果主机没有理由怀疑链路故障,并且客户机报告了对反向链路封装分组和零个前向链路 CRC 错误的正确响应,那么,主机可以假定,一切工作或运转正常(例如,没有链路故障),并继续功率下调/休眠过程。

[0652] 主机可以测试同步的另一种方式是:让主机发送往返行程延迟测量分组,并确认来自客户机的正确响应。如果主机收到正确的响应,则可以合理地假定,客户机正在成功地解释前向链路分组。

[0653] C. 初始化

[0654] 如前所述,在“启动”时,主机配置正向链路以便以低于最小要求或者需要的 1Mbps 的数据速率工作,或者就以此速率工作,并且针对给定应用适当地配置子帧长度和介质帧速率。也就是说,正反链路都使用类型 1 接口开始工作。当主机确定客户机显示器(或者其它类型的客户机设备)的能力或者所需要的配置时,通常只是临时使用这些参数。所述

主机经由正向链路发送或者传送子帧报头分组,其后跟随有反向链路封装分组,该分组将请求标志的位‘0’设置为值一(1),以便请求显示器或者客户机以客户机能力分组来响应。一旦显示器在(或者与)正向链路上获得同步,它就经由反向链路或者通道发送客户机能力分组和客户机请求和状态分组。

[0655] 主机检查客户机能力分组的内容,以便出于最优考虑或者所需要的性能级别,确定如何重新配置链路。主机检查协议版本和最小协议版本字段以便确认所述主机和客户机使用了彼此兼容的协议版本。所述协议版本通常是保存为客户机能力分组的开头两个参数,因此即使协议的其它元素不兼容或者不能完全理解为兼容,也可以确定兼容性。

[0656] 在内部模式下,主机能够提前知道客户机参数,而无需接收到客户机能力分组。链路可以按照使主机和客户机都能工作的任何数据速率重新启动。在很多实施例中,系统设计者很可能选择以可实现的最高数据速率来启动链路,以加快数据传输,但是,这不是必须的,并且在很多情况下也不需要这种方式使用。对于内部模式操作来说,在链路从休眠状态重启期间使用的选通脉冲的频率通常与该预期速率一致。

[0657] D. CRC 处理

[0658] 对于所有分组类型来说,分组处理器状态机确保 CRC 校验器被适当或正确地控制。当检测到 CRC 比较产生了一个或多个错误时,还增加 CRC 错误计数器,并且在被处理的每一子帧开始时重置所述 CRC 计数器。

[0659] E. 可选失去同步校验

[0660] 虽然上述系列步骤或者状态能够产生更高数据速率或者吞吐速度,但是申请人已经发现:还可以使用其它的替换设置或者条件的改变来有效地实现甚至更高的数据速率或者吞吐量,其中客户机使用这些条件来声明与主机失去同步。本发明新的实施例具有相同的基本结构,但是改变状态的条件发生了变化。另外,实现新的计数器以便有助于进行子帧同步检查。相对于图 63 给出了这些步骤和条件,该图示出用于建立方法或者状态机的操作的一系列状态和条件。为了清楚,只示出了“获得同步状态”和“同步中状态”部分。另外,由于得到的状态与状态机本身基本上是相同的,所以它们使用相同的编号。然而,改变状态(和状态机操作)的条件多少有些改变,因此为了使两个图都清楚,全部进行重新编号(1, 2, 3, 4, 5 和 6 相对于 61, 62, 63, 64, 和 65),以便于识别差异。由于在这一讨论中不考虑异步帧状态,所以在这个图中有一个状态(4904)和条件(6)不再使用。

[0661] 在图 63 中,所述系统或者客户机(用于显示或者呈现)从预选“非同步”状态 4902 中的状态机 5000 开始,这与图 49 相同。从非同步条件 4902 改变状态的第一次状态改变是在发现同步模式的条件 64 下进行的。假定子帧报头的 CRC 也在这个分组上传递(符合条件 61),分组处理器状态机的状态可以改变为同步中状态 4908。同步错误也就是条件 62 将让状态机切换至状态 4910,并且当第二次出现这个条件时切换至状态 4912。然而,已经发现 MDDI 分组的任何 CRC 失败都将让状态机移出同步中状态 4908,进入一个同步错误状态 4910。任何 MDDI 分组的又一 CRC 失败会导致移到两个同步失败状态 4912。利用正确的 CRC 值解码的分组将导致状态机返回同步中状态 4908。

[0662] 发生了变化的地方是利用了 CRC 值或者对‘每一’分组的 CRC 确定。也就是说,要让状态机注意(look at)每一分组的 CRC 值以便确定失去同步,而不是只观察子帧报头分组。在这一结构或者过程中,不是使用唯一字和仅仅使用子帧报头 CRC 值来确定失去同步。

[0663] 这种新的接口实现方式允许 MDD 接口链路更加快速地识别同步失败,从而同样更加迅速地从中恢复。

[0664] 为了使这种系统更加稳固,客户机还应该添加或利用子帧计数器。于是,在信号中唯一字预期到达或者出现时,客户机检查唯一字的存在。如果唯一字没有在正确时间出现,那么客户机可以更加快速地识别出同步失败已经出现,这种识别出同步失败的速度要比它必须等待几(在这里是三)个分组时间或者周期快的多,且其中所述分组要比子帧长度大得多。如果对唯一字的测试表明它未出现,换言之,时序是不正确的,那么客户机立即声明链路失去同步,并且移到非同步状态。检查正确的唯一字出现的过程向状态机添加了条件 65(cond65),说明唯一字不正确。如果期望在客户机上接收到子帧分组但是不相配,那么客户机可以立即进入非同步状态 4902,从而节省等待多个同步错误(条件 62)的附加时间,该多个错误同步通常是通过状态 4910 和 4912 而遇到的。

[0665] 这种改变使用附加计数器或者客户机核心中的计数功能来对子帧长度计数。在一个实施例中,使用递减计数功能,如果计数器期满,就中断目前正处理的任何分组的传送,以便检查子帧唯一字。作为选择,所述计数器也可以递增计数,将所述计数与所需要的最大或者特定需要值相比,在这个值对应的时刻校验当前分组。这一过程保护客户机不用对客户机上不正确地接收的长度特别长的分组解码。如果子帧长度计数器需要中断正被解码的某些其它分组,那么由于分组不应跨越子帧边界,所以可以确定已经失去同步。

[0666] IX. 分组处理

[0667] 对于状态机接收的上述每个类型的分组来说,它进行特定的处理步骤或者系列步骤来实现接口的操作。通常按照下面表 XII 列出的示例性处理过程来处理正向链路分组。

[0668] 表 XII

分组类型	分组处理器状态机响应
子帧报头 (SH)	确认良好的分组,捕获子帧长度字段并且向通用处理器发送分组参数。
填充符 (F)	忽略数据。
视频流 (VS)	解释视频数据格式描述符以及其它参数,必要时对已打包的像素数据,必要时通过色彩映射变换像素,并且把像素数据写入位图中的适当位置。
[0669] 音频流 (AS)	向音频采样时钟发生器发送音频采样速率设置,分离指定大小的音频采样,必要时拆开音频采样数据,并且把音频采样路由至适当音频采样FIFO
色彩映射 (CM)	读取色彩映射大小和偏移参数,并且把色彩映射数据写入色彩映射存储器或者存储单元。
反向链路封装 (REL)	便于在适当时间按照反向发送分组。检查反向链路标志,需要的时候发送客户机能力分组。只要合适,还发送客户机请求和状态分组。
客户机能力 (CC)	当主机请求时,使用反向链路封装分组的反向链路标志字段来发送这种类型的分组。

分组类型	分组处理器状态机响应
键盘 (K)	如果存在键盘型设备并且需要使用,就把这些分组传递至与该键盘类型设备通信的通用处理器并且从该通用处理器接收这些分组。
指示设备 (PD)	如果存在指示型设备并且需要使用,就把这些分组传递至与该指示类型设备通信的通用处理器并且从该通用处理器接收这些分组。
链路关闭 (LS)	记录链路被关闭的事实并且通知通用处理器。
客户机服务请求和状态 (CSRS)	发送这个分组作为反向链路封装分组中的第一分组。
位块传送 (BPT)	解释诸如视频数据格式描述符之类的分组参数,确定哪些像素首先移动,并且根据需要移动位图中的像素。
位图区域填充 (BAF)	解释分组参数,必要时通过色彩映射来变换像素,并且把像素数据写入位图中的适当位置,
位图图案填充 (BPF)	解释分组参数,必要时拆开已打包的像素数据,必要时通过色彩映射来变换像素,并且把像素数据写入位图中的适当位置。
通信链路通道 (CLC)	直接把这一数据发送至通用处理器。
休眠期间的客户机服务请求 (CSR)	通用处理器控制发送请求的低级功能并且检测与其自身上的链路重启的竞争。
接口类型切换请求 (ITHR) 和接口类型确认 (ITA)	可以将这些分组传递至通用处理器和从中传递出来。用于接收这种类型的分组并且以确认来构成响应的逻辑基本上是最小的。因此,这一操作还可以在分组处理器状态机内实现。得到的切换作为低级物理层动作存在,并且不可能影响通用处理器的功能。
执行类型切换 (PTH)	可以直接作用于这种分组或者通过把这种分组传送至通用处理器,以及命令硬件进行模式变换来实现。

[0671] X. 降低反向链路数据速率

[0672] 发明人已经观察到:用于主机链路控制器的某些参数可以按照特定的方式进行调节或者配置,以便实现最大或更优化(规模)的反向链路数据速率,这是非常需要的。例如,在用于传送反向链路封装分组的反向数据分组字段期间,MDDI\_Stb 信号对进行翻转(toggle)从而以正向链路数据速率的一半产生周期性的数据时钟。出现这种情况是因为在好像 MDDI\_Data0 正在发送全零的情况下,所述主机链路控制器生成对应于 MDDI\_Data0 信号的 MDDI\_Stb 信号。MDDI\_Stb 信号从主机被传送至客户机,在客户机中将它用来生成时钟信号以便从客户机传送反向链路数据,反向数据借此被送回主机。图 50 中示出了在采用 MDDI 的系统中在正向和反向路径上的信号传送和处理遇到的典型延迟量。在图 50 中,在 Stb+/- 生成、电缆传送至客户机、客户机接收器、时钟生成、信号时钟、Data0+/- 生成、电缆传送至主机和主机接收器级的处理部分附近,分别示出了一系列延迟值,即: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 纳秒以及 2.5 纳秒。

[0673] 根据正向链路数据速率以及遇到的信号处理延迟,可能需要 MDDI\_Stb 信号上一周期以上的时间来用于“往返行程”效应或者完成一组事件,这导致了时间或者周期量的不期望的消耗。为了防止这一问题的发生,反向速率除数使反向链路上的一位时间得以跨越 MDDI\_Stb 信号的多个周期。这意味着反向链路数据速率小于正向链路速率。

[0674] 应该注意的是,通过所述接口的信号延迟的实际长度随着每一特定主机-客户机系统或者使用的硬件不同而有所不同。尽管不是必需的,但是通过使用往返行程延迟测量分组来测量系统中的实际延迟,由此将反向速率除数设置为最佳值,每一系统通常可以更好地运行。主机可以支持较简单但以低速度运行的基本数据采样,或者也可支持较复杂但支持更高反向数据速率的高级数据采样。同样,也考虑了支持这两种方法的客户机能力。

[0675] 往返行程延迟是通过让主机把往返行程延迟测量分组发送至客户机来测量的。客户机通过在该分组中称为测量周期字段的预选测量窗内或者在这个预选测量窗期间里,把一个 1 序列发送回主机,以响应这个分组。先前描述了这一测量的详细时序。往返行程延迟用来确定可以对反向链路数据安全采样的速率。

[0676] 往返行程延迟测量包括确定、检测在测量周期字段开始和在主机接收到来自客户机的 0xff、0xff、0x00 响应序列这个时间段开始之间出现的正向链路数据时钟间隔的数目,或者对其进行计数。应注意的是,有可能在测量计数即将增加以前的一小部分正向链路时钟周期,收到来自客户机的响应。如果将这一未修改的值用来计算反向速率除数,那么它会导致由于不可靠的数据采样而在反向链路上产生位错误。图 51 中说明这种情况的一个实例,其中按照图形方式示出了代表主机处 MDDI\_Data、主机处 MDDI\_Stb、主机内部的正向链路数据时钟以及延迟计数的信号。在图 51 中,在延迟计数即将从 6 增加到 7 之前的一部分正向链路时钟周期,从客户机收到响应序列。如果把延迟假定为 6,那么主机将刚好转变之后或者可能在位转变当中对反向数据采样。这样做会导致主机处采样错误。为此,测量出的延迟在用来计算反向速率除数以前,通常应该加 1。

[0677] 反向速率除数是主机对反向链路数据采样以前应该等待的 MDDI\_Stb 周期数目。由于 MDDI\_Stb 是以正向链路速率一半来周期性循环的,所以正确的往返行程延迟测量结果需要除以 2,然后进位舍入至下一整数。这一关系表示为公式是:

[0678]

$$\text{反向\_速率\_除数} = \text{进位舍入到下一整数} \left( \frac{\text{往返\_行程\_延迟} + 1}{2} \right)$$

[0679] 对于给定实例来说,它变为:

[0680]

$$\text{反向\_速率\_除数} = \text{进位舍入到下一整数} \left( \frac{6+1}{2} \right) = 4$$

[0681] 如果用于这一实例的往返行程延迟测量结果是 7 而不是 6,那么反向速率除数也等于 4。

[0682] 所述反向链路数据由主机在反向链路时钟的上升沿时刻采样。主机和客户机(显示器)中都存在计数器或者类似已知的电路或设备用来生成反向链路时钟。计数器被初始化,因此反向链路时钟的第一上升沿出现在反向链路封装分组的反向链路分组字段的第一位开始处。在图 52A 中说明用于以下给出的实例的情况。所述计数器在 MDDI\_Stb 信号每

一上升沿增加 1, 并且计数数目一直增加直到反向链路封装分组中的反向速率除数参数设定该计数返回 (wrap around) 零。由于 MDDI\_Stb 信号是以正向链路速率的一半来翻转的, 所以反向链路速率是正向链路速率的一半除以反向速率除数。例如, 如果正向链路速率是 200Mbps, 而反向速率除数是 4, 那么反向链路数据速率可以表示为:

$$[0683] \quad \frac{1}{2} \cdot \frac{200\text{Mbps}}{4} = 25\text{Mbps}$$

[0684] 图 52 中示出了反向链路封装分组中的 MDDI\_Data0 和 MDDI\_Stb 信号线的时序的实例, 其中用于举例说明的分组参数具有如下值:

- [0685] 分组长度 = 1024 (0x0400)                      转向 1 长度 = 1  
 [0686] 分组类型 = 65 (0x41)                        转向 2 长度 = 1  
 [0687] 反向链路标志 = 0                            反向速率除数 = 2  
 [0688] 参数 CRC = 0xdb43                        全零是 0x00  
 [0689] 分组长度和参数 CRC 字段之间的分组数据是:

[0690] 0x00, 0x04, 0x41, 0x00, 0x02, 0x01, 0x01, 0x43, 0xdb, 0x00, .....

[0691] 从客户机返回的第一反向链路分组是具有分组长度 7 和分组类型 70 的客户机请求和状态分组。这个分组从字节值 0x07, 0x00, 0x46, ..... 等等开始。然而, 图 52 中只可以看到第一字节 (0x07)。在图中这个第一反向链路分组时间上移动了一个反向链路时钟周期, 以便举例说明实际的反向链路延迟。把主机至客户机往返行程延迟为零的理想波形显示为虚线轨迹。

[0692] 传送参数 CRC 字段的 MS 字节, CRC 的前面是分组类型, 然后是全零字段。当来自主机的数据改变电平时, 来自主机的选通从一切换至零, 并且回到一, 从而形成较宽的脉冲。当数据变为零时, 选通信号以更高速率切换, 只有数据线上的数据改变导致在对准字段的末尾附近产生改变。对于图中其余部分而言, 由于数据信号具有固定 0 或 1 电平且延续若干时间周期, 所以选通信号以所述更高速率切换, 并且这种转变落在脉冲图案 (沿) 上。

[0693] 主机的反向链路时钟处于零, 直到转向 1 周期结束为止, 此时时钟开始支持反向链路分组。图中较低部分的箭头表示数据采样的时刻, 正如从公开内容的其余部分可明显看出的一样。图中示出正传送的分组字段的第一字节 (在这里是 11000000) 在转向 1 之后开始出现, 并且线路电平因为主机驱动器被禁用而稳定。在数据信号的虚线中可以看出第一位的传递 (passage) 中的延迟, 并且正如位三所看到的那样。

[0694] 在图 53 中可以看到基于正向链路数据速率的反向速率除数的典型值。实际的反向速率除数是作为往返行程链路测量结果而确定的, 以便确保正确的反向链路操作。第一区域 5302 对应于安全操作区域, 第二区域 5304 对应于边缘性能区域, 而第三区域 5306 则表明大不可能正确工作的设置。

[0695] 在正向或者反向链路上利用所述接口类型设置中的任意设置工作的时候, 往返行程延迟测量和反向速率除数设置是相同的, 这是因为它们是根据实际时钟周期而不是传输或者接收的位数来表示并且操作的。

[0696] 通常, 最大可能的反向速率除数是能够用类型 I 接口在往返行程测量分组的测量窗口中发送的位数的一半, 或者, 对于该例子:

[0697]

$$\left( \frac{512 \text{字节} \cdot 8 \text{位/字节}}{2} \right) = 2048$$

[0698] 也可以采用一种先进的反向数据采样方法作为一种使反向位时间小于往返行程延迟的选择。对于这种技术,主机仅仅测量往返行程延迟,但也可以确定来自客户机的响应相对于具有零延迟的客户机和链路“理想”位边界的相位。知道了客户机设备响应的相位,主机就可以确定对来自客户机的反向数据进行采样的较安全时间。往返行程延迟测量向客户机指示反向数据的第一位相对于反向数据分组字段开始的位置。

[0699] 图 52B 中的示意图给出了先进反向数据采样的一个实施例。图中将具有零往返行程延迟的理想反向数据信号显示为点线波形。3.5 和 4 个 MDDI\_Stb 周期之间的实际往返行程延迟可以被观测为实线波形和理想波形之间的延迟差值。这与用往返行程延迟测量分组测量的延迟相同,是等于 7 个正向链路位时间的测量往返行程延迟值。在本实施例中,反向数据位是 2 个 MDDI\_Stb 脉冲长,即 4 个正向链路位时间,相当于等于 2 的反向速率除数。对于先进的反向数据采样,使用预选的反向速率除数 2、而不像其他地方那样计算它,是很便利的。这看起来是针对先进反向数据采样的基本上最佳的选择,因为使用上述的传统测量可以很容易地确定理想的采样点。

[0700] 反向数据的理想采样点可以通过如下方式很容易地进行计算:将总往返行程延迟的提示信号除以每个反向位的前向链路时钟数目或每个反向位的往返行程延迟以前向链路时钟为模。然后,减去 1 或 2,得到远离数据转换的一个安全点。在本例中,  $7 \bmod 4 = 3$ , 所以,  $3-1 = 2$  或  $3-2 = 1$ 。安全采样点是 1 个或 2 个正向链路位时间,从零往返行程延迟的“理想”位边界边沿开始。该图示出的采样点在从理想位边界开始的 2 个正向链路位时间处,如该时序图底部的一系列垂直箭头所示。第一采样点是测量往返行程延迟后的第一理想位边界加上安全采样的偏移量。在本例中,往返行程延迟测量是 7,所以,下一理想位边界处于第 8 个位时间,然后对于安全采样点加 1 或 2,所以,第一位应当在反向数据分组字段开始后的 9 个或 10 个正向链路位时间进行采样。

[0701] XI. 转向和保护时间

[0702] 反向链路封装分组中的转向 1 字段使主机驱动器禁用和客户机驱动器启用有时间同时进行。往返行程延迟测量分组中的保护时间 1 字段使主机和客户机能重叠,所以,客户机驱动器能够在主机接口驱动器禁用之前启用。反向链路封装分组中的转向 2 字段使得前一段中的数据在主机驱动器启用之前能够从客户机完全发送出去。保护时间 2 字段提供一个时间值或周期,以使客户机和主机驱动器能够在逻辑零电平同时驱动。保护时间 1 和保护时间 2 字段通常以不能调节的预置长度或者预选长度值来填充。根据正在使用的接口硬件,可以使用经验数据来获得这些值并且在某些情况下调节它们以便改善操作。

[0703] 转向 1

[0704] 有几个因素有助于确定转向 1 的长度,这些因素是正向链路数据速率、主机中 MDDI\_Data 驱动器的最大禁用时间和客户机驱动器的启用时间(通常与主机禁用时间相同)。选择转向 1 字段的长度为  $24 \cdot t_{BIT}$ 。(表 XI) 转向 1 字段的正向链路字节数的长度用接口类型因子(InterfaceTypeFactor)确定,并且用以下关系计算:

$$[0705] \quad \text{Length}_{\text{TurnAround1}} = \frac{24}{8 \text{bits/byte}} \cdot \text{InterfaceTypeFactor}_{\text{FWD}} = 3 \cdot \text{InterfaceTypeFactor}_{\text{FWD}}$$

[0706] 其中接口类型因子对于类型 1 为 1,对于类型 2 为 2,对于类型 3 为 4,对于类型 4 为 8。

[0707] 转向 2

[0708] 确定通常用于转向 2 的时间长度的因素是通信链路的往返行程延迟、客户机中 MDDI\_Data 驱动器的最大禁用时间以及主机驱动器的启用时间（被指定为与客户机驱动器禁用时间相同）。最大主机驱动器启用时间和客户机驱动器禁用时间在中给出。往返行程延迟的测量单位是  $t_{BIT}$ 。以转向 2 字段的正向链路字节数指定的最小长度根据以下关系计算：

$$[0709] \quad Length_{TurnAround2} \geq RoundUpToInteger \left( \frac{RoundTripDelay + 24}{8bits / byte} \cdot InterfaceTypeFactor_{FWD} \right)$$

[0710] 例如,往返行程延迟为 10 个正向链路时钟的类型 3 前向链路通常使用量级如下的转向 2 延迟：

$$[0711] \quad Length_{TurnAround2} \geq RoundUpToInteger \left( \frac{11 + 24}{8} \cdot 4 \right) = 18bytes$$

[0712] XII. 可选反向链路时序

[0713] 虽然使用上述的时序和保护带 (guard band) 能够实现高数据传送速率接口,但是本发明的发明人已经发现了一种技术,通过改变反向时序发现 (reverse timing discovery) 来提供短于往返行程时间的反向位长。

[0714] 如上所述,反向链路时序的先前方式被如此配置,即,从反向时序分组的保护时间 1 的最后位开始计数时钟周期的数目,直到在 10 时钟的上升沿对第一位进行采样为止。这就是用于给 MDD 接口的输入和输出进行定时的时钟信号。于是,反向速率除数的计算由以下等式给出：

[0715]

$$\text{反向速率除数} = \text{进位舍入到下一整数} \left( \frac{\text{往返行程延迟} + 1}{2} \right)$$

[0716] 这样做提供了等于往返行程延迟的位宽度,由此能够得到非常可靠的反向链路。然而,已经说明了反向链路能够更快地运行,或者以更高数据传送速率运行,这是发明人希望利用的。本发明新的技术允许利用接口的附加能力来达到更高速度。

[0717] 这是通过让主机对时钟周期数目进行计数,直到对一个 1 完成采样来实现的,但是在反向时序分组期间,主机在上升沿和下降沿上均对数据线采样。这样做允许主机选择反向位内的最有用的甚至最佳采样点,以便确保所述位是稳定的。也就是说,发现最有用的或者最佳的上升沿来对反向业务的反向封装分组上的数据进行采样。最佳采样点取决于反向链路除数和是在上升沿还是下降沿检测到第一个 1。新的时序方法允许主机只寻找在反向链路时序内由客户机发送的 0xFF0xFF 0x00 模式的第一沿,以便确定在反向封装分组中的什么地方采样。

[0718] 在图 64 中说明到达的反向位并且该位将如何寻找各种反向速率除数的实例,以及从保护时间 1 的最后位开始已经出现的时钟周期数目。在图 64 中可以看出,如果第一沿出现在上升和下降沿 (标记为上升 / 下降) 之间,那么对于反向速率除数为 1 的最佳采样点,即该最佳采样点是标记为 'b' 的时钟周期沿,因为它是反向位周期内出现的唯一上升



沿。对于反向速率除数为 2 的情况来说,由于周期沿 ‘c’ 比 ‘b’ 更接近位沿,所以最佳采样点可能仍是时钟周期前沿 ‘b’。对于反向速率除数为 4 的情况来说,最佳采样点可能是时钟周期沿 ‘d’,因为它更加接近其值可能已经稳定的反向位的后沿。

[0719] 回到图 64,然而,如果第一沿出现在下降和上升沿(标记为下降/上升)之间,那么对于反向速率除数为 1 的最佳采样点是采样点时钟周期沿 ‘a’,因为这是反向位时间周期内出现的唯一上升沿。对于反向速率除数为 2 的情况来说,最佳采样点是沿 ‘b’,并且对于反向速率除数为 4 的情况来说,最佳采样点是沿 ‘c’。

[0720] 可以看出,由于最佳采样点应该是最靠近中部的上升沿,所以随着反向速率除数越来越大,最佳采样点变得更便于确定或者选择。

[0721] 主机使用这一技术可以找到在数据线上观察到的时序分组的上升数据沿之前的上升时钟沿的数目。因此,基于所述沿是出现在上升和下降沿之间还是下降和上升沿之间,以及反向速率除数是什么,有多少附加时钟周期要添加到计数器,就可做出决定,以便合理地确保所述位的采样始终尽可能靠近中部。

[0722] 一旦主机已经选择或者确定时钟周期的数目,它就可以利用客户机来“调查(explore)”各种反向速率除数,以便确定特定反向速率除数是否能用。所述主机(和客户机)可以以除数 1 启动,并且校验从客户机接收的反向状态分组的 CRC,以便确定这一反向速率是否能够正确地传送数据。如果 CRC 被破坏,那么有可能存在采样错误,主机可以增大反向速率除数,并试图再次请求状态分组。如果第二次请求的分组被破坏,那么可以再次增大除数,并且再次进行请求。如果这个分组被正确地解码,那么这一反向速率除数可以用于所有将来的反向分组。

[0723] 这一方法是有效的并且是有用的,因为自从初始的往返行程时序估算开始反向时序不应该发生改变。如果正向链路是稳定的,那么即使存在反向链路故障,客户机也应该继续对正向链路分组解码。当然,由于这一方法不确保反向链路是理想的,所以设置所述链路的反向链路除数仍是主机的职责。另外,所述除数将主要取决于用于生成 IO 时钟的时钟质量。如果该时钟具有很大的抖动量,那么存在采样误差的可能性更大。这一错误概率随着往返行程延迟中的时钟周期量的增加而增加。

[0724] 这种实现方式看来似乎最适用于类型 1 的反向数据,但是对类型 2 到类型 4 的反向数据来说可能会出现,这是由于数据线之间的偏离可能过大,以至于无法以最适用于仅仅一个数据对的速率运行所述链路。然而,即使是采用类型 2 至类型 4 进行操作,所述数据速率也许不需要降低到先前方法。如果在每一数据线上重复该方法以便选择理想的或者最佳时钟采样位置,那么这种方法也可能起到最好的作用。如果对于每个数据对来说,它们处于相同的采样时间,那么这种方法能够继续工作。如果它们处于不同的采样周期,那么可以使用两个不同的方式。第一种是为每个数据点选择所需要的或更优化的采样位置,即使对于每一数据对而言该采样位置并不同。于是,所述主机能够在对来自所述数据对集合的所有位进行采样之后重构数据流:类型 2 是两位,类型 3 是四位而类型 4 是八位。对于主机而言,另一个选项是增大反向速率除数,从而使每一数据对的数据位可以在相同的时钟沿采样。

[0725] XIII. 链路延迟和偏离的效应

[0726] 正向链路上 MDDI\_Data 对和 MDDI\_Stb 之间的延迟偏离会限制可能的最大数据速

率,除非采用延迟偏离补偿。产生时序偏离的延迟差异起因于以下概述的控制器逻辑、线路驱动器和接收器以及电缆和连接器。

[0727] A. 由偏离限制的链路时序分析 (MDDI 类型 1)

[0728] 1. 类型 1 链路的延迟和偏离实例

[0729] 图 57 中示出了类似于图 41 所示的,用于支持类型 1 接口链路的典型接口电路。在图 57 中,为 MDDI 类型 1 的正向链路的多个处理或接口级的每个示出了传播延迟和偏离的示例性值或者典型值。MDDI\_Stb 和 MDDI\_Data0 之间的延迟中的偏离导致输出时钟的占空比 (duty-cycle) 扭曲。使用触发器 5728、5732 的接收器触发器 (RXFF) 级的 D 输入端的数据在时钟沿之后稍微改变,从而该数据能够被可靠地采样。所述图示出了两个级联延迟线 5732a 和 5732b,其利用产生这一时序关系来解决两个不同问题。在实际实现方式中,可以将它们合并成单个延迟元件。

[0730] 图 58 示出了用于经由所述接口进行示例性信号处理的类型 1 链路上的数据、Stb 和时钟恢复时序。

[0731] 通常明显的总延迟偏离出现或者来自如下级中偏离的总和:具有触发器 5704、5706 的发射器触发器 (TXFF);具有驱动器 5708、5710 的发射器驱动器 (TXDRVR);电缆 (CABLE) 5702;具有接收器 5722、5724 的接收器线路接收器 (RXRCVR) 以及接收器 XOR 逻辑 (RXXOR)。延迟 15732a 应该匹配或者超过 RXXOR 级的 XOR 门 5736 的延迟,通过如下关系可确定该延迟:

$$[0732] \quad t_{PD-min(\text{延迟 } 1)} \geq t_{PD-max(XOR)}$$

[0733] 最好符合这一要求,以便接收器触发器 5728、5732 的 D 输入端不会在其时钟输入以前改变。如果 RXFF 的保持时间是零,那么这是有效的。

[0734] 延迟 2 (Delay2) 的目的或功能在于按照如下关系来补偿 RXFF 触发器的保持时间:

$$[0735] \quad t_{PD-min(\text{延迟 } 2)} = t_{H(RXFF)}$$

[0736] 在许多系统中,这个值将是零,因为保持时间是零,在这种情况下,延迟 2 的最大延迟当然也是零。

[0737] 接收器 XOR 级中偏离的最坏情况是数据 - 迟到 / 选通 - 提早这种情况,其中延迟 1 处于最大值,而来自 XOR 门的时钟输出按照如下关系尽可能早地到达:

$$[0738] \quad t_{\text{偏离-max}(RXXOR)} = t_{PD-max(\text{Delay } 1)} - t_{PD-min(XOR)}$$

[0739] 在这种情况下,数据可以在两个位周期 n 和 n+1 之间改变,且改变时刻非常接近位 n+1 被按照时钟打入接收器触发器的时刻。

[0740] MDDI 类型 1 链路的最大数据速率 (最小位周期) 是 MDDI 链路中通过所有驱动器、电缆和接收器而遇到的最大偏离加上设置到 RXFF 级中的总计数据的函数。链路中直到 RXRCVR 级输出端的总延迟偏离可以表示为:

$$[0741] \quad t_{\text{偏离-max}(LINK)} = t_{\text{偏离-max}(TXFF)} + t_{\text{偏离-max}(TXDRVR)} + t_{\text{偏离-max}(电缆)} + t_{\text{偏离-max}(RXRCVR)}$$

[0742] 其中“线缆”代表多种导体或互联或线和相应的延迟,并且最小位周期由下式给出:

$$[0743] \quad t_{\text{位-min}} = t_{\text{偏离-max}(链路)} + 2 \cdot t_{B-TP4} + t_{\text{Asymmetry}} + t_{\text{偏离-max}(RXXOR)} + t_{\text{jitter-host}} + t_{PD-max(\text{Delay } 2)} + t_{SU(R}$$

XFF)

[0744] 在图 57 所示的实例中,对于外部模式, $t_{\text{偏离-max(链路)}} = 1000\text{psec}$ ,并且最小位周期可以表示为:

[0745]  $t_{\text{位-min}} = 1000 + 2 \cdot 125 + 625 + 125 + 200 + 0 + 100 = 2300\text{psec}$  或者近似表示为 434Mbps。在图 57 所示的实例中,对于内部模式, $t_{\text{偏离-max(链路)}} = 500\text{psec}$ ,并且最小位周期可以表示为:

[0746]  $t_{\text{位-min}} = 500 + 2 \cdot 125 + 625 + 125 + 200 + 0 + 100 = 1800\text{psec}$  或者近似表示为 555Mbps。

[0747] B. MDDI 类型 2、类型 3 和类型 4 的链路时序分析

[0748] 在图 59 中示出了类似于图 41 和 57 中示出的典型接口电路,用于支持类型 2、3、和 4 的接口链路。在 TXFF (5904)、TXDRVR (5908)、RXRCVCR (5922) 和 RXFF (5932, 5928, 5930) 级中使用附加元件,以便支持附加信号处理。在图 59 中,为 MDDI 类型 2 正向链路的多个处理或接口级的每个示出了传播延迟和偏离的示例性值或者典型值。除影响输出时钟占空比 (duty-cycle) 的 MDDI\_Stb 和 MDDI\_Data0 之间的延迟中的偏离以外,在这两个信号和其它 MDDI 数据信号之间也存在偏离。由触发器 5928 和 5930 组成的接收器触发器 B (RXFFB) 级的 D 输入端的数据,在时钟沿之后稍微改变,从而能够可靠地对该数据进行采样。如果 MDDI\_Data1 比 MDDI\_Stb 或者 MDDI\_Data0 提早到达,那么 MDDI\_Data1 应被延迟,以便至少以延迟偏离量来对 MDDI\_Data1 进行采样。为了实现这些,使用延迟 3 (Delay3) 延迟线来延迟数据。如果 MDDI\_Data1 比 MDDI\_Stb 和 MDDI\_Data0 晚到,并且它也通过延迟 3 来延迟,那么 MDDI\_Data1 改变处的点被移动至更接近下一时钟沿。这一过程决定了 MDDI 类型 2、类型 3 或者类型 4 链路的数据速率的上限。图 60A、60B 和 60C 中示出了两个数据信号和 MDDI\_Stb 相对于彼此的时序或者偏离关系的某些示例性的不同可能性。

[0749] 当 MDDI\_DataX 尽早到达时,为了在 RXFFB 中可靠地对数据采样,按照如下关系来设置延迟:

[0750]  $t_{\text{PD-min(Delay3)}} \geq t_{\text{偏离-max(LINK)}} + t_{\text{H(RXFFB)}} + t_{\text{PD-max(XOR)}}$

[0751] 最大链路速度由允许的最小位周期确定。当 MDDI\_DataX 尽可能晚到达时,该速度最受影响。在这种情况下,允许的最小周期时间通过如下等式给出:

[0752]  $t_{\text{位-min}} = t_{\text{偏离-max(LINK)}} + t_{\text{PD-max(Delay3)}} + t_{\text{SU(RXFFB)}} - t_{\text{PD-min(XOR)}}$

[0753] 于是,链路速度的上限为:

[0754]  $t_{\text{PD-max(Delay3)}} = t_{\text{PD-min(Delay3)}}$

[0755] 并且在该假定下:

[0756]  $t_{\text{位-min(lower-bound)}} = 2 \cdot t_{\text{偏离-max(LINK)}} + t_{\text{PD-max(XOR)}} + t_{\text{SU(RXFFB)}} + t_{\text{H(RXFFB)}}$

[0757] 在上文给出的实例中,最小位周期的下限通过如下关系给出:

[0758]  $t_{\text{位-min(lower-bound)}} = 2 \cdot (1000 + 2 \cdot 125 + 625 + 200) + 1500 + 100 + 0 = 5750\text{psec}$ ,这大约是 174Mbps。

[0759] 这比可以用于类型 1 链路的最大数据速率慢得多。MDDI 的自动延迟偏离补偿能力明显降低了延迟偏离对最大链路速率因子的影响,仅仅是有效数据建立的边缘。MDDI\_Data0 和 MDDI\_Stb 之间的校准偏离是:

[0760]  $t_{\text{SKEW-max(calibrated)}} = 2 \cdot t_{\text{TAP-SPACING-max}}$

[0761] 并且最小位周期是:

[0762]  $t_{\text{BIT-min-calibrated}} = t_{\text{SKEW-max(calibrated)}} + 2 \cdot t_{\text{B-TP4}} + t_{\text{Asymmetry}} + t_{\text{jitter-host}} + t_{\text{SKEW-max(RXAND+RXOR)}} + t_{\text{SU(RXF)}}$

F)

[0763] 其中,“TB”或  $t_B$  代表从一个位边界到最小输出电平的信号抖动。非对称仅仅表示差分接收器内部延迟的非对称特性。“TP4”关联于或有效地定义用于电气表征和测试目的,作为客户机差分线路驱动器和接收器的连接或接口(客户机中 MDDI 控制器设备的管脚)。对于系统剩余部分的链路,它代表一个便利的或预定的点,据此测量和表征信号延迟。在一个实施例中,对于内部模式而言,在 TP4 处参数  $T_B$  的最大值通过关系  $t_{\text{Differential-Skew-TP4-DRVR-EXT}} = 0.3 \cdot t_{\text{BIT}}$  来定义,对于客户机发射机的内部模式来说,是  $t_{\text{Differential-Skew-TP4-DRVR-INT}} = 0.6 \cdot t_{\text{BIT}}$ ;对于客户机接收器的外部模式来说,是  $t_{\text{B-TP4-RCVR-EXT}} = 0.051 \cdot t_{\text{BIT}} + 175\text{ps}$ 。

[0764] 标记 TP4 仅仅用于对接口和链路中不同的测试点 (TP) 进行编号。在一个实施例中,对于内部模式和外部模式该测试点定义成相同。存在一个相应的“TP0”测试点,用于或关联于一个主机中 MDDI 控制器设备的连接或接口管脚,该主机包含差分线路驱动器和接收器。在本实施例中,对于主机接收器而言,对于内部模式,TP0 处的参数  $T_B$  的最大值由关系  $t_{\text{B-TP0-RCVR-INT}} = 0.051 \cdot t_{\text{BIT}} + 50\text{ps}$  来定义,对于外部模式,关系为  $t_{\text{B-TP0-RCVR-EXT}} = 0.051 \cdot t_{\text{BIT}} + 175\text{ps}$ ;对于主机发射机,关系为  $t_{\text{B-TP0}} = 0.102 \cdot t_{\text{BIT}}$ 。

[0765] 在图 59 给出的示例中,  $t_{\text{SKEW-max(Data0-Stb-Calibrated)}} = 300\text{psec}$ ,而最小位周期是:

[0766]  $t_{\text{BIT-min-calibrated}} = 300 + 2 \cdot 125 + 625 + 200 + 175 + 100 = 1650\text{psec}$ ,这大约是 606Mbps。

[0767] 当 MDDI\_Data1 尽早到达时,为了在 RXFFB 中可靠地对数据采样,将关联的可编程延迟调整到最佳设定,精度为一个抽头,为安全起见,增加附加的抽头延迟。最大的链路速度取决于最小的允许位时段。当 MDDI\_Data1 尽晚到达时,影响最大。在这种情况下,最小允许循环时间为:

[0768]  $t_{\text{BIT-min-Data1-calibrated}} = 2 \cdot t_{\text{TAP-Spacing-max}} + 2 \cdot t_{\text{TA-TP4}}$

[0769] 其中,“TA”或  $t_A$  表示从位边界到中间交叉的信号抖动。

[0770] 在图 59 给出的示例中,基于采样 MDDI\_Data1 的最小位时段的下限是:

[0771]  $t_{\text{BIT-min-Data1-calibrated}} = 2 \cdot 150 + 2 \cdot 125 = 550\text{psec}$

[0772] 在一个实施例中,对于内部模式而言,主机中的延迟偏离、延迟非对称性和时钟抖动的典型总延迟时间定义成:

[0773]  $t_{\text{Asymmetry-TXFF}} + t_{\text{Asymmetry-TXDVR}} + t_{\text{Skew-TXFF}} + t_{\text{Skew-TXDVR}} + t_{\text{jitter-host}} = 0.467 \cdot (t_{\text{BIT}} - 150\text{ps})$

[0774] 对于外部模式为:

[0775]  $t_{\text{Asymmetry-TXFF}} + t_{\text{Asymmetry-TXDVR}} + t_{\text{Skew-TXFF}} + t_{\text{Skew-TXDVR}} + t_{\text{jitter-host}} = 0. \text{TBD} \cdot (t_{\text{BIT}} - 150\text{TBDps})$

[0776] 对于内部模式而言,客户机中的延迟偏离、延迟非对称性和建立时间的典型总延迟时间 ( $t_{\text{B-TP4}}$ ) 定义成:

[0777]  $t_{\text{Asymmetry-RXRCVR}} + t_{\text{Asymmetry-RXXOR}} + t_{\text{Skew-RXRCVR}} + t_{\text{Skew-RXXOR}} + t_{\text{setup-RXFF}} = 0.307 \cdot (t_{\text{BIT}} - 150\text{ps})$

[0778] 对于外部模式,为:

[0779]  $t_{\text{Asymmetry-RXRCVR}} + t_{\text{Asymmetry-RXXOR}} + t_{\text{Skew-RXRCVR}} + t_{\text{Skew-RXXOR}} + t_{\text{setup-RXFF}} = 0. \text{TBD} \cdot (t_{\text{BIT}} - \text{TBDps})$

[0780] 其中,项 TBD 是用于未来待定值的一个灵活位置保持标记,其取决于外部模式连接的多种公知特性和操作要求。

[0781] XIV. 物理层互连描述

[0782] 可用于实现本发明的接口的物理连接可以使用能够购买到的零件来实现,诸如在主机侧,可以是由 Hirose Electric Company Ltd. 制造的零件号为 3260-8S2(01) 的

零件,并且在客户机设备侧,可以是由 Hirose Electric Company Ltd. 制造的零件号为 3240-8P-C 的零件。表 XIII 中列出了供类型 1/ 类型 2 接口使用的这种连接器的示例性接口引脚分配或者“引脚引出线”,并且在图 61 中举例说明。

[0783] 表 XIII

[0784]

信号名称	引脚号	信号名称	引脚号
MDDI_Pwr	1	MDDI_Gnd	11
MDDI_Stb+	2	MDDI_Stb-	12
MDDI_Data0+	4	MDDI_Data0-	14
MDDI_Data1+	6	MDDI_Data1-	16
MDDI_Data2+	8	MDDI_Data2-	18
MDDI_Data3+	10	MDDI_Data3-	20
MDDI_Data4+	9	MDDI_Data4-	19
MDDI_Data5+	7	MDDI_Data5-	17
MDDI_Data6+	5	MDDI_Data6-	15
MDDI_Data7+	3	MDDI_Data7-	13
		屏蔽	

[0785]

[0786] 所述屏蔽与主机接口中的 HOST\_Gnd 相连,并且电缆中的屏蔽地线与客户机连接器的屏蔽相连。然而,这个屏蔽和地线不与客户机内部的电路地相连。

[0787] 选择或者设计互连元件或设备,以便使其足够小以用于移动通信和计算设备,如 PDA 和无线电话,或者便携式游戏设备,而在与相关设备尺寸的比较中不会超出或者显得不美观。任何连接器和电缆都应该在典型用电设备环境下足够耐用,并且尺寸要小,对电缆而言更是如此,并且成本相对要低。传送元件应该支持作为差分 NRZ 数据的数据和选通信号,对于类型 1 和类型 2 而言,该差分 NRZ 数据具有高达 450Mbps 左右的传送速率,而对于 8 位并行类型 4 的版本而言,具有高达 3.6Gbps 的传送速率。

[0788] 对于内部模式应用来说,同样对于所使用的导线没有连接器,或者这种连接元件倾向于小型化。一个实例为零插入力“插座”,用于接受插入主机或者客户机设备的集成电路或元件。另一实例是主机和客户机驻留在具有各种互联导线的印刷电路板上的情况,并且具有从外壳中伸出的“引脚”或触点,这些引脚或触点被焊接至用于集成电路互联的导线上。

**[0789]** XV. 操作

**[0790]** 图 54A 和 54B 中示出了使用本发明实施例的接口操作期间,在处理数据和分组的时候进行的通用步骤的概况,并且图 55 中示出了处理所述分组的接口装置的概况。在这些图中,所述过程在步骤 5402 开始,确定客户机和主机是否使用通信路径(在这里是电缆)连接。这可以通过主机使用软件或者硬件来周期性轮询而进行,所述软件或硬件用于检测在主机的输入端是否存在连接器、电缆或者信号(例如 USB 接口),或者这也通过其它已知的技术进行。如果没有客户机连接至主机,那么根据应用,它只可以进入某个预定长度的等待状态,进入休眠模式,或者处于非激活状态以便等待将来使用,这会要求用户采取行动来重新激活主机。例如,当主机驻留在计算机类型的设备上时,用户可能必须点击屏幕图标或者请求程序,从而激活主机处理以寻找客户机。此外,仅仅插入 USB 类型连接就可以激活主机处理,这取决于主机或者驻留主机软件的能力和配置。

**[0791]** 一旦客户机与主机相连,或者反过来,或者客户机被检测为存在,那么在步骤 5404 和 5406,客户机或者主机发送适当的请求服务分组。在步骤 5404,客户机可以发送客户机服务请求或者状态分组。应注意的是,如上所述,所述链路先前可能已经关闭或者处于休眠模式,因此随后的未必是通信链路的完整初始化。一旦通信链路是同步的,并且主机试图与客户机通信,客户机就也向主机提供客户机能力分组,如步骤 5408。主机现在可以开始确定客户机能够支持的支持类型,包括传送速率。

**[0792]** 通常,在步骤 5410,主机和客户机还议定待使用的服务模式的类型(速率/速度),例如类型 1、类型 2 等等。一旦服务类型被建立,主机就能开始传送信息。另外,主机可以与其它信号处理并行地使用往返行程延迟测量分组来优化通信链路的时序,如步骤 5411 所示。

**[0793]** 如前所述,所有传送从步骤 5412 示出的正在传送的子帧报头分组开始,继之为在步骤 5414 示出的正在传送的数据类型和填充符分组,在这里数据类型是视频和音频流分组。所述音频和视频数据已经预先预备好或者映射到分组中,并且根据需要或者期望插入填充符分组以便填充介质帧所需要的位数。主机能发送如正向音频通道启用分组的分组来激活音响设备。另外,在步骤 5416 中,主机能使用上述其它分组类型来传送命令和信息,在这里显示为色彩映射、位块传送或者其它分组的传送。此外,所述主机和客户机能使用适当的分组交换涉及键盘或者指示设备的数据。

**[0794]** 在操作期间,多个不同事件之一会出现,这将导致主机或者客户机需要不同的数据速率或者不同的接口模式类型。例如,用于传递数据的计算机或者其它设备可能在处理数据的过程中遇到装载条件 (loading conditions),所述条件导致分组的预备或者呈现速度减慢。接收数据的客户机有可能从专用交流电源换成更加有限的电池电源,并且在有限的功率设定下,可能不能迅速地传送数据,容易地处理命令,也无法使用同一分辨率或者色彩浓度。或者,限制条件也可能减少或者消失,从而允许设备以更高的速率传送数据。这更加需要做出请求以改变为更高传送速率模式。

**[0795]** 如果这些或者其它类型的已知条件出现或者改变,主机或者客户机会检测到它们并且试图重新协商议定接口模式。这一点在步骤 5420 中示出,其中所述主机向客户机发送接口类型切换请求分组以请求切换到另一模式,所述客户机发送接口类型确认分组,以便确认寻求改变,并且所述主机发送执行类型切换分组以便改变为指定的模式。

[0796] 虽然不要求有特定的处理顺序,但是所述客户机和主机还可以交换涉及某些数据的分组,这些数据是要发送给主要与客户机连接的指示设备、键盘或其它用户类型输入设备的,或者是从这些设备中收到的数据,当然这些元件还可以存在于主机一侧。这些分组通常使用通用处理器类型的元件而非状态机来处理(5502)。另外,上述某些命令也由通用处理器(5504,5508)处理。

[0797] 在主机和客户机间已经交换了数据和命令之后,在某一点,就是否传送附加数据或者主机或客户机是否要停止为所述传送提供的服务而做出决定。这在步骤5422中示出。如果所述链路将进入休眠状态或者完全关闭,那么主机向客户机发送链路关闭分组,并且两侧终止传送数据。

[0798] 使用先前相对于主机和客户机控制器讨论的驱动器和接收器来传送在上述操作处理中传送的分组。这些线路驱动器以及其它逻辑元件与上述状态机和通用处理器相连,如图55的概述所示。在图55中,状态机5502和通用处理器5504和5508可以进一步与未示出的其它元件相连,如专用USB接口、存储器元件或驻留在与它们相互作用的链路控制器外部的其它部件,包括但不限于数据源和用于观看显示器设备的视频控制芯片。

[0799] 处理器和状态机提供对驱动器的启用和禁用控制,正如上文就保护时间等等所讨论的那样,以便确保有效建立或终止通信链路,并且传送分组。

[0800] XVI 显示器帧缓冲器

[0801] 与计算机图形相比,视频数据的缓存要求对于运动视频图像是不同的。像素数据经常储存在客户机的本地帧缓冲器中,因此客户机上的图像能本地刷新。

[0802] 在显示全运动视频(每一介质帧中显示器中几乎每一像素都要改变)时,通常优选在一个帧缓冲器中储存正在输入的像素数据,而从第二帧缓冲器中刷新显示器上的图像。可以用两个以上的显示缓冲器来消除可视假象(visible artifact),如下所述。当已经在一个帧缓冲器中接收了整个图像时,就可以交换缓冲器的角色,将最新接收的图像用于刷新所述显示器,并且将图像的下一帧填充另一个缓冲器。图88A中示出了这一概念,其中通过把显示器更新位设置为“01”来把像素数据写入离线图像缓冲器。

[0803] 在其它应用中,所述主机只需要更新小部分图像,而不必重新绘制整个图像。在这种情况下,需要把新的像素直接写入用于刷新显示器的缓冲器,如图88B中所示。

[0804] 在具有较小视频窗口的固定图像的应用中,把固定图像写入两个缓冲器(显示器更新位等于“11”)是最容易的,如图88C所示,并且随后通过把显示器更新位设置为“01”把运动图像像素写入离线缓冲器。

[0805] 如下规则描述了缓冲器指针的有益操纵,同时把新的信息写入客户机并且刷新显示器。存在三个缓冲器指针:current\_fill(当前填充)指向当前正在由MDDI链路上的数据填充的缓冲器;just\_filled(刚刚填充)指向最近被填充的缓冲器;being\_displayed(正在显示)指向目前正用于刷新显示器的缓冲器。所有三个缓冲器指针可以包含从0到N-1的值,其中N是显示器缓冲器的数目,而 $N \geq 2$ 。对缓冲器指针的运算是取模(mod N),例如,当 $N = 3$ 并且 $current\_fill = 2$ 时,增加current\_fill让current\_fill被设置为0。在 $N = 2$ 的简单情况下,just\_filled始终是current\_fill的补码。在每一MDDI介质帧边界上(子帧计数字段的子帧报头分组等于零)按照指定的顺序来执行如下操作:将just\_filled设置为等于current\_fill,并且设置current\_fill等于current\_

fill+1。

[0806] MDDI 视频流分组按照如下结构或方法来更新缓冲器：当显示器更新位等于‘01’时，把像素数据写入由 current\_fill 指定的缓冲器；当显示器更新位等于‘00’时，把像素数据写入由 just\_filled 指定的缓冲器；当显示器更新位等于“11”时，把像素数据写入所有缓冲器。由 being\_displayed 指针指定的缓冲器来刷新显示器。在显示器刷新一个帧刷新时代 (epoch) 中的最后像素之后并且在它开始刷新下一帧刷新时代中的第一像素以前，显示器更新处理执行把 being\_refreshed (正在刷新) 设置为等于 just\_filled 的操作。

[0807] 所述视频流分组包含一对显示器更新位，说明将写入像素数据的帧缓冲器。所述客户机能力分组具有三个附加位，其表明客户机中所支持的显示器更新位的组合。在许多情况下，计算机生成的图像需要基于用户输入递增地刷新，或者根据从计算机网络接收的信息来导出。显示器更新位组合“00”和“11”通过让像素数据写入正在显示的帧缓冲器或者写入两个帧缓冲器来支持这一操作模式。

[0808] 当支持视频图像时，图 89 说明当显示器更新位等于“01”，且经由 MDDI 链路传输视频数据时，如何使用一对帧缓冲器来显示视频图像。在 MDDI 链路上检测到介质帧边界之后，在当前正刷新的帧的刷新处理完成时，显示器刷新处理将开始从下一帧缓冲器刷新。

[0809] 涉及图 89 的一个重要假设是：图像是作为像素的连续流从主机接收的，所述像素的连续流是按照客户机从帧缓冲器读取像素以便刷新显示器的顺序来传输的（通常从左上开始，逐行读取，直到屏幕的右下角）。在显示器刷新和图像传送操作涉及相同的帧缓冲器的情况下，这是重要的细节。

[0810] 显示器刷新帧速率需要大于图像传送帧速率，以免显示出部分图像。图 90 示出了在显示器刷新速率较慢情况下，也就是说，显示器刷新比图像传送慢的情况下，图像碎片是如何出现的。

[0811] 在包含计算机图形图像和运动视频图像的组中的图像中，视频像素数据可能占据小部分介质帧。在显示器刷新操作和图像传送涉及相同的帧缓冲器的情况下，这是十分重要的。这些情况通过图 91 中的交叉影线的阴影示出，其中从缓冲器中读取以便刷新显示器的像素可能是两帧以前写入缓冲器的像素，或者它们可以对应于正要被写入同一帧缓冲器的帧。

[0812] 客户机中使用三个帧缓冲器来解决访问帧缓冲器的小窗口竞争问题，如图 92 所示。

[0813] 然而，如果显示器刷新速率小于 MDDI 链路上的介质帧速率，那么仍存在问题，如图 93 所示。

[0814] 运动视频图像使用单个缓冲器多少存在问题，如图 94 所示。当显示器刷新快于图像传送到缓冲器中时，正刷新的图像有时将显示正写入的帧的上部，而图像的较低部分则是先前传送的帧。当显示器刷新快于图像传送（优选操作模式）时，示出类似分裂图像的帧的情况将更加频繁地出现。

#### [0815] XVII. 延迟值表

[0816] 分组处理延迟参数分组使用表查找功能来计算所预测的延迟，以便处理客户机中的某些命令。表中的值按照对数方式增加，以便提供动态范围非常宽的延迟值。在下面的表 XX 中可以找出用于实现本发明实施例的延迟值的示例性表，相应的索引与延迟值对照。



[0817] 表 XX

[0818]

0-无延迟	37-1.5ns	74-51ns	111-1.8us	148-62us	185-2.2ms	222-75ms
1-46ps	38-1.6ns	75-56ns	112-2.0us	149-68us	186-2.4ms	223-83ms
2-51ps	39-1.8ns	76-62ns	113-2.2us	150-75us	187-2.6ms	224-91ms
3-56ps	40-2.0ns	77-68ns	114-2.4us	151-83us	188-2.9ms	225-100ms
4-62ps	41-2.2ns	78-75ns	115-2.6us	152-91us	189-3.2ms	226-110ms
5-68ps	42-2.4ns	79-83ns	116-2.9us	153-100us	190-3.5ms	227-120ms
6-75ps	43-2.6ns	80-91ns	117-3.2us	154-110us	191-3.8ms	228-130ms
7-83ps	44-2.9ns	81-100ns	118-3.5us	155-120us	192-4.2ms	229-150ms
8-91ps	45-3.2ns	82-110ns	119-3.8us	156-130us	193-4.6ms	230-160ms
9-100ps	46-3.5ns	83-120ns	120-4.2us	157-150us	194-5.1ms	231-180ms
10-110ps	47-3.8ns	84-130ns	121-4.6us	158-160us	195-5.6ms	232-200ms
11-120ps	48-4.2ns	85-150ns	122-5.1us	159-180us	196-6.2ms	233-220ms
12-130ps	49-4.6ns	86-160ns	123-5.6us	160-200us	197-6.8ms	234-240ms
13-150ps	50-5.1ns	87-180ns	124-6.2us	161-220us	198-7.5ms	235-260ms
14-160ps	51-5.6ns	88-200ns	125-6.8us	162-240us	199-8.3ms	236-290ms
15-180ps	52-6.2ns	89-220ns	126-7.5us	163-260us	200-9.1ms	237-320ms
16-200ps	53-6.8ns	90-240ns	127-8.3us	164-290us	201-10ms	238-350ms
17-220ps	54-7.5ns	91-260ns	128-9.1us	165-320us	202-11ms	239-380ms
18-240ps	55-8.3ns	92-290ns	129-10us	166-350us	203-12ms	240-420ms
19-260ps	56-9.1ns	93-320ns	130-11us	167-380us	204-13ms	241-460ms
20-290ps	57-10ns	94-350ns	131-12us	168-420us	205-15ms	242-510ms
21-320ps	58-11ns	95-380ns	132-13us	169-460us	206-16ms	243-560ms
22-350ps	59-12ns	96-420ns	133-15us	170-510us	207-18ms	244-620ms
23-380ps	60-13ns	97-460ns	134-16us	171-560us	208-20ms	245-680ms
24-420ps	61-15ns	98-510ns	135-18us	172-620us	209-22ms	246-750ms

25-460ps	62-16ns	99-560ns	136-20us	173-680us	210-24ms	247-830ms
26-510ps	63-18ns	100-620ns	137-22us	174-750us	211-26ms	248-910ms
27-560ps	64-20ns	101-680ns	138-24us	175-830us	212-29ms	249-1.0sec
28-620ps	65-22ns	102-750ns	139-26us	176-910us	213-32ms	250-1.1sec
29-680ps	66-24ns	103-830ns	140-29us	177-1.0ms	214-35ms	251-1.2sec
30-750ps	67-26ns	104-910ns	141-32us	178-1.1ms	215-38ms	252-1.3sec
31-830ps	68-29ns	105-1.0us	142-35us	179-1.2ms	216-42ms	253-1.5sec
32-910ps	69-32ns	106-1.1us	143-38us	180-1.3ms	217-46ms	254-1.6s
33-1.0ns	70-35ns	107-1.2us	144-42us	181-1.5ms	218-51ms	255- 不确定的
34-1.1ns	71-38ns	108-1.3us	145-46us	182-1.6ms	219-56ms	
35-1.2ns	72-42ns	109-1.5us	146-51us	183-1.8ms	220-62ms	
36-1.3ns	73-46ns	110-1.6us	147-56us	184-2.0ms	221-68ms	

[0819] 通过将指定参数作为表的索引进行表格查找来计算延迟。这意味着延迟等于分组处理表（索引）。例如：如果来自延迟参数列表项的参数之一是等于 134 的 8 位值，那么延迟等于分组处理表 (134)，即，16 微秒。值 255 表明无法通过计算确定命令完成时间，主机必须检查客户机请求和状态分组中的图形繁忙标志或者 MCCS VCP 控制参数 B7h。

[0820] 在某些情况下，把这一延迟乘以目标图像的高度、宽度或者像素数目，并且将其加到其它延迟上去，以计算整个分组处理延迟。

[0821] XVIII 多客户机支持

[0822] 当前协议版本看起来无法直接支持多个客户机设备。然而，大部分分组包含保留的客户机 ID 字段，该客户机 ID 可用于对具有多个客户机的系统中的特定客户机设备进行寻址。目前，对于许多应用来说，这一客户机 ID 或者这些客户机 ID 被设置为零。所述子帧报头分组还包含用于表明主机是否支持多个客户机系统的字段。因此，存在这样一种方式，其中在 MDD 接口或者协议的未应用中有可能会将多个客户机设备连接起来并进行寻址，以便帮助系统设计师规划将来与多个客户机主机和客户机的兼容性。

[0823] 在具有多个客户机的系统中，把客户机经由客户机的串级链 (Daisy-chained) 或者使用集线器连接至主机，如图 95 所示，或者使用如图 96 所示的这些技术的组合将客户机连接至主机都是十分有益的。

[0824] XVIII. 附录

[0825] 除了上面讨论的用于实现本发明实施例的体系结构和协议的格式、结构和内容以外，在这里对某些分组类型提供了更加详细的字段内容或操作。在这里给出这些内容是为了进一步阐明它们各自的用途或操作，以便使本领域技术人员更加易于理解以及将本发明用于各种应用。在这里只讨论没有讨论过的少数几个字段。另外，这些字段是相对于上文

给出的实施例以示例性的定义和值给出的。然而,不应该把这些值看作是对本发明的限制,而是它们代表实现这些接口和协议有用的一个或多个实施例,并且无需一起或者同时实施所有的实施例。其它实施例也可以采用其它值以便实现所需要的数据或数据速率传送结果的呈现,正如本领域技术人员将理解的那样。

#### [0826] A. 关于视频流分组

[0827] 在一个实施例中,像素数据属性字段(2字节)具有如下解释的一系列位值。位1和0选择如何路由该显示器像素数据。对于位值‘11’来说,向或为两只眼睛显示数据,对于位值‘10’来说,只把数据路由至左眼,而对于位值‘01’来说,只把数据路由至右眼,而对于位值‘00’来说,把数据路由至可选显示器,正如下面讨论的由位8至11所指定的那样。

[0828] 位2表明是否以隔行的格式提供像素数据,值‘0’意味着像素数据处于标准的逐行格式,从一行前进至下一行时,行号(像素Y坐标)加1。当这一位具有值‘1’时,像素数据处于隔行的格式,并且从一行前进到下一行时,行号加2。位3表明像素数据处于交替像素格式。这与位2启用的标准隔行模式相似,但是这里的隔行是垂直的而不是水平的。当位3是‘0’时,像素数据处于标准的逐个像素格式,当接收到每一连续像素时,列号(像素X坐标)加1。当位3是‘1’时,像素数据处于交替像素格式,接收到每一像素时,列号加2。

[0829] 位4表明像素数据涉及显示器还是照相机,即,其中数据被传送至无线电话或者类似设备甚至便携式计算机或如上所述的其它设备的内部显示器,或者从这些设备中传送出来,或者数据被传送到嵌入或者直接耦合到所述设备的照相机或者从该照相机中传送出来。当位4是‘0’时,像素数据正传送到显示器帧缓冲器或者从中传送出来。当位4是‘1’时,像素数据被传送到某些类型的照相机或者视频设备或者从中传送出来,这种设备是所属领域众所周知的。

[0830] 位5用来表明何时像素分组含显示器中下一连续行的像素。当位5被设置为等于‘1’时,被认为是这样的情况。当位5被设置为‘1’时,那么X左沿、Y顶沿、X右沿、Y底沿、X起始和Y起始参数不被定义并且被客户机忽略。当位15被设置为逻辑1电平时,它表明该分组中的像素数据是所述图像中最后一行像素的像素数据。客户机能力分组的客户机特征能力标识符字段的位8表明支持该特征。

[0831] 位7和6是显示器更新位,用于说明将写入像素数据的帧缓冲器。其它地方讨论了更加具体的效果。对于位值‘01’来说,像素数据被写入离线图像缓冲器中。对于位值‘00’来说,像素数据被写入用于刷新显示器的图像缓冲器中。对于位值‘11’来说,像素数据被写入所有图像缓冲器中。所述位值或者组合‘10’被当做无效值或者指定,并且像素数据被忽略,而不被写入任何图像缓冲器。这一值可能对于所述接口的将来应用具有用途。

[0832] 位8至11形成4位无符号整数,用于说明要将要像素数据路由到的可选显示器或者显示器位置。位0和1被设置为等于00,以便使显示器客户机把位8至11解释为可选显示器号。如果位0和1不等于00,那么位8至11被设置为零。

[0833] 位12至14被保留以供将来使用并且通常应设定为逻辑零。所讨论的位15与位5结合使用,并且将位15设置位逻辑1表明像素数据段中像素的行是数据帧中的最后一行像素。位5设置为逻辑1的下一个视频流分组对应于下一个视频帧的第一行像素。

[0834] 2字节的X起始和Y起始字段指定像素数据字段中第一像素的点(X起始,Y起始)的绝对X和Y坐标。2字节的X左沿和Y顶沿字段指定由像素数据字段填充的屏幕窗口的

左沿的 X 坐标以及顶沿的 Y 坐标,而 X 右沿和 Y 底沿字段则指定正在更新的窗口的右沿的 X 坐标以及底沿的 Y 坐标。

[0835] 像素计数字段 (2 字节) 指定以下像素数据字段中的像素数目。

[0836] 参数 CRC 字段 (2 字节) 包含从分组长度到像素计数的所有字节的 CRC。如果这一 CRC 未能通过校验,就丢弃整个分组。

[0837] 所述像素数据字段包含将被显示的原始视频信息,并且按照视频数据格式描述符字段描述的方式格式化。如同其它地方讨论的那样,数据每次传输一“行”。当像素数据属性字段的位 5 设置为逻辑 1 电平时,像素数据字段刚好包含一行像素,且正在传输的第一个像素对应于最左像素,而传输的最后一个像素对应于最右像素。

[0838] 像素数据 CRC 字段 (2 字节) 只包含像素数据的 16 位 CRC。如果这一值的 CRC 校验失败,那么像素数据仍可使用,但是 CRC 错误计数增加。

[0839] B. 关于音频流分组

[0840] 在一个实施例中,音频通道 ID 字段 (1 字节) 使用 8 位无符号整数值来标识特定音频通道,其中客户机设备把音频数据发送至该通道。物理音频通道通过这个字段按照值 0, 1, 2, 3, 4, 5, 6 或者 7 被指定或者映射到物理通道,这些值分别表明左前、右前、左后、右后、前方中心、超低音扬声器、环绕左以及环绕右通道。音频通道 ID 值 254 表明数字音频采样的单个流被发送给左前和右前通道。这样做简化了一些应用的通信,如将立体声耳机用于语音通信的应用,对 PDA 使用的生产力改进的应用,或者简单用户接口生成报警声的其它应用。在从 8 至 253 和 255 的范围内的 ID 字段值目前被保留,以便在新设计需要附加标记的情况下使用,正如本领域技术人员可预见的那样。

[0841] 保留 1 字段 (1 字节) 通常被保留以备将来之用,并且将这个字段中的所有位都设置为零。这个字段的一个功能是让所有后续 2 字节字段与 16 位字地址对准,让 4 字节字段与 32 位字地址对准。

[0842] 音频采样计数字段 (2 字节) 指定这个分组中音频采样的数目。

[0843] 每个采样位数和打包字段包含 1 字节,用于说明音频数据的打包格式。在一个实施例中,通常采用的格式是使用位 4 至 0 来定义每一 PCM 音频采样的位数。然后,位 5 说明所述数字音频数据采样是否被打包。如上所述,图 12 说明打包的音频采样和字节对准音频采样之间的差异。位 5 的值‘0’表明数字音频数据字段中每一 PCM 音频采样与接口的字节边界是字节对准的,并且值‘1’表明每一连续 PCM 音频采样相对于先前音频采样被打包。通常只有当位 4 至 0 (每一 PCM 音频采样的位数) 中定义的值不是 8 的倍数时,这一位才是有效的。位 7 至 6 被保留以供系统设计需要附加标识时使用,并且通常设置为零值。

[0844] 音频采样速率字段 (1 字节) 指定音频 PCM 采样速率。所采用的格式分别是:值 0 时速率是 8,000 个采样每秒 (sps),值 1 表明 16,000sps,值 2 表明 24,000sps,值 3 表明 32,000sps,值 4 表明 40,000sps,值 5 表明 48,000sps,值 6 表明 11,025sps,值 7 表明 22,050sps 并且值 8 表明 44,100sps,而值 9 至 255 被保留供将来之用,因此它们目前被设置为零。

[0845] 参数 CRC 字段 (2 字节) 包含从分组长度到音频采样速率的所有字节的 16 位 CRC。如果这一 CRC 未能适当地通过校验,就丢弃整个分组。数字音频数据字段包含待播放的原始音频采样,并且通常具有无符号整数的线性格式的形式。音频数据 CRC 字段 (2 字节) 包

含只针对音频数据的 16 位 CRC。如果这一 CRC 未能通过校验,那么所述音频数据仍可被使用,不过 CRC 错误计数增加。

[0846] C. 关于用户定义的流分组

[0847] 在一个实施例中,2 字节的流 ID 号字段用来标识特定的用户定义的流。流参数和流数据字段的内容通常由 MDDI 设备制造商定义。2 字节的流参数 CRC 字段包含从分组长度开始到音频编码字节的流参数的所有字节的 16 位 CRC。如果这一 CRC 未能通过校验,就丢弃整个分组。如果 MDD 接口的最终应用不需要流参数和流参数 CRC 字段,即它们被看作可选的,那么流参数和流参数 CRC 字段都可以被丢弃。2 字节的流数据 CRC 字段包含只针对流数据的 CRC。如果这一 CRC 未能适当地通过校验,那么流数据的使用是可选的,这取决于所述应用的要求。流数据的使用随 CRC 的良好性而定,通常要求缓存流数据直到 CRC 被确认为良好的为止。如果 CRC 没有通过校验,那么所述 CRC 错误计数增加。

[0848] D. 关于色彩映射分组

[0849] 2 字节的 hClient ID 字段包含为客户机 ID 保留的信息或值,如先前使用的那样。由于这个字段通常被保留供将来之用,所以当前通过把这些位设置为‘0’来将其设置为零。

[0850] 2 字节的色彩映射项目计数字段使用值来指定 3- 字节色彩映射项目的总数,这些项目包含在色彩映射数据字段中,或者指定存在于这个分组的色彩映射数据中的色彩映射表的条目的总数。在这个实施例中,色彩映射数据中的字节数目是色彩映射项目计数的 3 倍。色彩映射项目计数被设置为等于零以便不发送色彩映射数据。如果色彩映射大小是零,那么色彩映射偏移值通常仍被发送,但是被显示器忽略。色彩映射偏移字段(4 字节)指定这个分组中色彩映射数据距客户机设备中的色彩映射表的开始的偏移量。

[0851] 2 字节的参数 CRC 字段包含从分组长度到音频编码字节的所有字节的 CRC。如果这一 CRC 未能通过校验,就丢弃整个分组。

[0852] 对于色彩映射数据字段来说,每一色彩映射位置的宽度是由色彩映射项大小字段指定的,其中在一个实施例中,第一部分指定蓝色数值,第二部分指定绿色数值而第三部分指定红色数值。色彩映射大小字段指定存在于色彩映射数据字段中的 3 字节色彩映射表项的数目。如果单个色彩映射无法符合(fit into)视频数据格式和色彩映射分组,那么可以通过发送多个分组来指定整个色彩映射,其中每一分组中具有不同色彩映射数据和色彩映射偏移。每一色彩映射数据项中蓝色、绿色和红色的位数应该与显示器能力分组的色彩映射 RGB 宽度字段指定的相同。

[0853] 2 字节的色彩映射数据 CRC 字段只包含色彩映射数据的 CRC。如果这一 CRC 未能通过校验,那么所述色彩映射数据仍可使用,但是 CRC 错误计数增加。

[0854] 每一色彩映射数据项将会按照如下的顺序传输:蓝色、绿色、红色,每一分量的最低有效位首先传输。将每一色彩映射项目的单个红色、绿色和蓝色分量打包,但是每一色彩映射项目(蓝色分量的最低有效位)应该按字节对准。图 100 示出了具有 6 位蓝色、8 位绿色和 7 位红色的色彩映射数据项的实例。对于这一实例来说,色彩映射分组中的色彩映射项大小等于 21,而客户机能力分组的色彩映射 RGB 宽度字段等于 0x0786。

[0855] E. 关于反向链路封装分组

[0856] 参数 CRC 字段(2 字节)包含从分组长度到转向长度的所有字节的 16 位 CRC。如果这一 CRC 未能通过校验,就丢弃整个分组。

[0857] 在一个实施例中,反向链路标志字段(1字节)包含用于请求来自客户机的信息的一组标志。如果位(例如位0)被设置为逻辑1电平,那么主机使用客户机能力分组从显示器请求指定信息。如果所述位被设置为逻辑零电平,那么主机不需要来自客户机的信息。剩余位(在这里是位1至7)被保留以供将来使用并且被设置为零。然而,还可以根据需要使用更多位来设置用于反向链路的标志。

[0858] 反向速率除数字段(1字节)指定与反向链路数据时钟相关而出现的MDDI\_Stb周期的数目。所述反向链路数据时钟等于正向链路数据时钟除以两倍的反向速率除数。反向链路数据速率涉及反向链路数据时钟和反向链路上的接口类型。在一个实施例中,对于类型1的接口来说,反向数据速率等于反向链路数据时钟,对于类型2、类型3和类型4的接口来说,反向数据速率分别等于2倍、4倍和8倍的反向链路数据时钟。

[0859] 全零1字段包含一组字节,这里是8,通过将各个位设置成逻辑0电平可将全零1字段设置为等于零,并且该字段可用来确保所有MDDI\_Data信号在转向1字段期间禁用主机线路驱动器之前能够处于逻辑零电平达足够的时间,以便允许客户机仅利用MDDI\_Stb开始恢复时钟。在一个实施例中,全零1字段的长度大于或等于电缆的往返行程延迟中的正向链路字节传输次数的数目。

[0860] 转向1长度字段(1字节)指定分配给转向1的字节总数,用于建立第一转向周期。在主机中的线路驱动器被禁用之前,转向1字段采用由转向1长度参数指定的字节数目,以便允许启用客户机中的MDDI\_Data线路驱动器。在转向1的位0期间,所述客户机启用其MDDI\_Data线路驱动器,并且主机禁用其输出端以便在转向1的最后位之前完全禁用其输出端。MDDI\_Stb信号表现得如同在整个转向1周期期间MDDI\_Data0均处于逻辑0电平。上面给出了转向1设置的更加完整的说明。

[0861] 反向数据分组字段包含一系列从客户机传送到主机的数据的分组。当客户机没有数据要发送到主机时,客户机可发送填充分组或将MDDI\_Data线路驱动到逻辑0状态或电平。在本实施例中,如果MDDI\_Data线路被驱动到0,主机则将其解释为具有0长度的分组(不是有效长度),并且主机在当前反向链路封装分组期间将不再从客户机接收另外的分组。

[0862] 转向2长度字段(1字节)指定分配给转向2的字节总数,用于建立第二转向周期。转向2的推荐长度是往返行程延迟加上主机启用其MDDI\_Data驱动器所需的时间所需的字节数。转向2长度值也可以大于最小所需或计算值,以使得能有足够的时间来处理主机中的反向链路分组。

[0863] 转向2字段包括转向长度参数指定的字节数。主机在启用其MDDI\_Data线路驱动器之前在转向2期间等待至少往返行程延迟时间。主机启用其MDDI\_Data线路驱动器,从而它们在转向2的最后位之前通常完全启用,客户机禁用其输出,从而它们在转向2的最后位之前通常完全禁用。转向2字段的目的是能够从客户机发送或传输自反向数据分组字段开始的剩余数据量。考虑到不同系统实现该接口和分配的安全容量方面的差异,有可能的是,主机和客户机都未在转向2字段周期的某些部分期间将MDDI\_Data信号驱动到逻辑零电平,如主机中的线路接收器所示。MDDI\_Stb信号表现得如同MDDI\_Data0基本上在整个转向2周期期间处于逻辑0电平。上面给出了转向2设置的描述。

[0864] 反向数据分组字段包含从客户机传送到主机的一系列数据分组。如前所述,发送

填充符分组以便填充其它分组类型未使用的剩余空间。

[0865] 全零 2 字节包含一组被设置为等于零值的字节（在本实施例中为 8 个），这是通过将这些位设置成逻辑 0 电平来实现的，并且该字段用来确保在跟随转向 2 字节启动了主机线路驱动器之后，所有 MDDI\_Data 信号能够处于零状态达足够的时间，以便客户机能够使用 MDDI\_Data0 和 MDDI\_Stb 二者开始恢复时钟。

[0866] F. 关于客户机能力分组

[0867] 如一个实施例所示，协议版本字段使用 2 字节来指定由客户机所使用的协议版本。初始版本目前被设置为等于 1，并且随着新版本的产生，该初始版本也会随时间而改变，正如将会知道的那样，而最小协议版本字段使用 2 字节来指定客户机能采用或者解释的最小协议版本。在这种情况下，0 值也是有效值。数据速率能力字段（2 字节）指定客户机在接口的正向链路上的每个数据对上能接收的最大数据速率，并且以每秒百万位（Mbps）的形式来指定。接口类型能力字段（1 字节）指定正反链路上支持的接口类型。设置为“1”的位表明支持指定的接口，而设置为“0”的位表明不支持该指定类型。主机和客户机应该在正反向链路上至少支持类型 1。不需要支持接口类型的相邻范围。例如，只支持接口中的类型 1 和类型 3 而不支持类型 3 和类型 4 将是非常有效的。也不需要正反向链路使用相同的接口类型来运行。然而，当链路从休眠中苏醒时，正反向链路应该在类型 1 模式开始运行直到协商议定或选择出其他模式，或者主机和由客户机二者均同意使用其他模式为止。

[0868] 在一个实施例中，通过选择位 0、位 1 或者位 2 表明支持的接口类型，以便分别选择正向链路上的类型 2（2 位）、类型 3（4 位）或者类型 4（8 位）模式，而位 3、位 4 或位 5 则分别选择反向链路上的类型 2、类型 3 或者类型 4 模式；位 6 和 7 被保留并且通常设置为零。位图宽度和高度字段（2 字节）分别以像素为单位指定位图的宽度和高度。

[0869] 单色能力字段（1 字节）用来指定可以按照单色格式显示的分辨率的位数。如果显示器无法使用单色格式，那么这一值被设置为零。位 7 至 4 被保留以供将来使用，从而设置为零。位 3 至 0 定义每一像素中可以存在的灰度级的最大位数。这四位能够为每一像素指定值 1 至 15。如果所述值是零，那么该显示器不支持所述单色格式。

[0870] Bayer 能力字段使用 1 个字节能够以 Bayer 格式传送的分辨率的位数、像素组和像素顺序。如果客户机不能使用 Bayer 格式那么该值为零。Bayer 能力字段由以下值构成：位 3 至 0 定义了每个像素中存在的强度的最大位数，而位 5 至 4 定义了必要的像素组模式，而位 8 至 6 定义了必要的像素顺序；位 14 至 9 被保留以供将来使用并且同时通常设置为零。位 15 设置为 1 时，表明客户机能够接受打包的或未打包格式的 Bayer 像素数据。如果位 15 被设置为 0，这表明客户机只能够接受未打包格式的 Bayer 像素数据。

[0871] 色彩映射能力字段（3 字节）指定显示器色彩映射表中存在的表项的最大数目。如果显示器无法使用色彩映射格式，那么这一值是零。

[0872] RGB 能力字段（2 字节）指定可以按照 RGB 格式显示的分辨率的位数。如果显示器无法使用 RGB 格式，那么这一值等于零。RGB 能力字包括三个独立的无符号值，其中：在每一像素中，位 3 至 0 定义蓝色的最大位数，位 7 至 4 定义绿色的最大位数，而位 11 至 8 定义红色的最大位数。目前，位 14 至 12 被保留以供将来使用并且通常设置为零。位 15 设置为 1 时，表明客户机能够接受打包的或未打包格式的 RGB 像素数据。如果位 15 被设置为逻辑 0 电平，这表明客户机只能够接受未打包格式的 RGB 像素数据。

[0873] Y Cr Cb 能力字段 (2 字节) 指定可以按照 Y Cr Cb 格式显示的分辨率的位数。如果显示器无法使用 Y Cr Cb 格式,那么这一值被设置为等于零。所述 Y Cr Cb 能力字包括三个独立的无符号值,其中:位 3 至 0 定义 Cb 采样中的最大位数,位 7 至 4 定义 Cr 采样中的最大位数,位 11 至 8 定义 Y 采样中的最大位数,而位 15 至 12 目前被保留供将来之用并且被设置为零。

[0874] 客户机特征能力指示符字段使用 4 字节,它包含一组用于表明客户机中所支持的特定特征的标志。设置为逻辑 1 电平的位表明支持所述能力,而设置为逻辑零电平的位表明不支持所述能力。在一个实施例中,位 0 的值表明是否支持位图块传送分组 (分组类型 71)。位 1、2 和 3 的值分别表明是否支持位图区域填充分组 (分组类型 72)、位图图案填充分组 (分组类型 73) 或者通信链路数据通道分组 (分组类型 74)。位 4 的值表明客户机是否有能力使一种颜色透明,而位 5 和 6 的值分别表明客户机能否接受打包格式的视频数据或者音频数据,而位 7 的值表明客户机能否从照相机发送反向链路视频流。位 8 的值表明客户机是否有能力接收全线路 (full line) 的像素数据并忽略如视频流分组的像素数据属性字段的位 5 指定的显示器寻址,并且客户机还可使用像素数据属性字段的位 15 检测帧同步或视频帧数据的末尾。

[0875] 位 11 和 12 的值分别表明客户机是正在与指示设备通信并且能发送和接收指示设备数据的分组,还是正在与键盘通信并且能发送和接收键盘数据的分组。位 13 的值表明客户机是否有能力通过支持 VCP 特征分组来设置一个或多个音频或视频参数,其中 VCP 特征分组如:请求 VCP 特征分组、VCP 特征应答分组、设置 VCP 特征分组、请求有效参数分组和有效参数应答分组。位 14 的值表明客户机是否有能力将像素数据写入离线显示器帧缓冲器。如果该位的值被设置位逻辑 1 电平,那么显示器更新位 (视频流分组的像素数据属性字段的位 7 和 6) 可被设置为值“01”。

[0876] 位 15 的值表明客户机是否有能力将像素数据只写入当前正在用于刷新显示器图像的显示器帧缓冲器。如果该位被设置为 1,那么显示器更新位 (视频流分组的像素数据属性字段的位 7 和 6) 可被设置为值“00”。位 16 的值表明客户机有能力将来自单个视频流分组的像素数据写入所以显示器帧缓冲器的情况。如果该位被设置为 1,那么显示器更新位 (视频流分组的像素数据属性字段的位 7 和 6) 可被设置为值“11”。

[0877] 位 17 的值表明何时客户机有能力响应于请求特定状态分组,位 18 的值表明何时客户机有能力响应于往复行程延迟测量分组,位 19 的值表明何时客户机有能力响应正向链路偏离校准分组。

[0878] 位 21 的值表明何时客户机有能力解释请求特定状态分组,并利用有效状态应答列表分组做出响应。如在其他地方所述,客户机表明有能力在有效状态应答列表分组的有效参数应答列表字段中返回另外的状态。

[0879] 位 22 表明客户机是否有能力响应于寄存器访问分组。位 9 至 10 和位 23 至 31 表明目前被保留供将来之用,或者用作对系统设计员有益的可选标记,并且通常被设置为等于零。

[0880] 显示器视频帧速率能力字段 (1 字节) 指定以每秒帧数为单位的显示器的最大视频帧更新能力。主机可以选择以慢于该字段中指定的值的速率来更新所述图像。

[0881] 所述音频缓冲器深度字段 (2 字节) 指定专用于每一音频流的显示器中的弹性缓



冲器的深度。

[0882] 音频通道能力字段 (2 字节) 包含一组标志,用于表明客户机或客户机连接设备支持哪些音频通道。设置为 1 的位表明该通道是可支持的,而设置为零的位表明该通道是不支持的。将位的位置分配给不同的通道,例如在一个实施例中,位位置 0、1、2、3、4、5、6 或者 7 分别表明左前、右前、左后、右后、前方中心、超低音扬声器、环绕左以及环绕右通道。位 8 至 14 目前保留以供将来使用并且通常设置为零。在一个实施例中,位 15 用于表明客户机是否提供对正向音频通道启用分组的支持。如果是这种情况,位 15 被设置为逻辑 1 电平。然而,如果作为正向音频通道启用分组的结果,客户机不能禁用音频通道,或者如果客户机不支持任何音频能力,那么这一位被设置为逻辑 0 电平或值。

[0883] 用于正向链路的 2 字节的音频采样速率能力字段包含一组标志,用于表明客户机设备的音频采样速率能力。因此,把位位置分配给不同速率,诸如位 0、1、2、3、4、5、6、7 和 8 被分别分配给 8,000、16,000、24,000、32,000、40,000、48,000、11,025、22,050 和 44,100 采样数每秒 (SPS),位 9 至 15 被保留以供将来使用或者根据需要用于可选速率,因此它们目前被设置为 '0'。把这些位之一的位值被设置为 '1' 表明可支持该特定的采样速率,而把该位设置为 '0' 表明该采样速率是不支持的。

[0884] 最小子帧速率字段 (2 字节) 指定以每秒帧数为单位的最小子帧速率。最小子帧速率保持足以读取显示器中的某些传感器或者指示设备的显示状态刷新速率。

[0885] 用于反向链路的 2 字节的麦克风采样速率能力字段包含一组标志,用于表明客户机设备的麦克风的音频采样速率能力。出于利用 MDDI 的目的,客户机设备麦克风被配置成最低限度支持最少 8,000 个采样每秒的速率。把用于这个字段的位位置分配给不同速率,例如,其中位位置 0、1、2、3、4、5、6、7 和 8 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 采样数每秒 (SPS),位 9 至 15 被保留以供将来使用或者根据需要用于可选速率,因此它们目前被设置为 '0'。把这些位之一的位值设置为 '1' 表明该特定的采样速率是支持的,而把该位设置为 '0' 表明该采样速率是不支持的。如果没有连接麦克风,那么麦克风采样速率能力位的每位均被设置为等于零。

[0886] 键盘数据格式字段 (这里为 1 个字节) 说明键盘是否连接到客户机系统以及所连接的键盘的类型。在一个实施例中,由位 6 至 0 确立的值用于定义所连接的键盘的类型。如果该值为零,则键盘类型被视作未知。当为 1 值时,键盘数据格式被认为是标准 PS-2 方式。目前未使用 2 至 125 范围内的值,这些值被保留下来以供系统设计人员或接口公司 (incorporator) 或产品开发商用来定义与 MDD 接口或相应的客户机或主机一同使用的特定的键盘或输入设备。值 126 用于表明用户定义的键盘数据格式,而值 127 用于表明不能连接到客户机的键盘。此外,位 7 能够用于表明键盘是否能够与客户机通信。该位的预期使用是用于表明键盘使用无线链路与客户机通信的情况。如果位 6 至 0 表明键盘不能连接的客户机,那么位 7 将被设置为 0 电平。因此,对于一个实施例而言,当位 7 的值是 0 时,键盘和客户机不能通信,而如果位 7 的值是 1,那么键盘和客户机已经确认它们能够彼此通信。

[0887] 指示设备数据格式字段 (这里是 1 字节) 说明了指示设备是否连接到客户机系统以及所连接的指示设备的类型。在一个实施例中,由位 6 至 0 建立的值用于定义所连接的指示设备的类型。如果该值为零 (0),则指示设备类型被视作未知。当为 1 值时,指示设备数据格式被认为是标准 PS-2 方式。目前未使用 2 至 125 范围内的值,这些值被保留下来以

供系统设计人员或接口公司 (incorporator) 或产品开发商用来定义与 MDD 接口或相应的客户机或主机一同使用的特定的指示设备或输入设备。值 126 用于表明用户定义的指示设备数据格式, 而值 127 用于表明不能连接到客户机的指示设备。此外, 位 7 能够用于表明指示设备是否能够与客户机通信。该位的预期使用是用于表明指示设备使用无线链路与客户机通信的情况。如果位 6 至 0 表明指示设备不能连接的客户机, 那么位 7 将被设置为 0 电平。因此, 对于一个实施例而言, 当位 7 的值是 0 时, 指示设备和客户机不能通信, 而如果位 7 的值是 1, 那么指示设备和客户机已经确认它们能够彼此通信。

[0888] 内容保护类型字段 (2 字节) 包含一组标志, 用于表明由显示器支持的数字内容保护类型。目前, 位位置 0 用来表明支持 DTCP 的情况, 而位位置 1 用来表明支持 HDCP 的情况, 且位位置 2 至 15 被保留以供所需要的或可用的其他保护方案之用, 因此它们目前被设置为零。

[0889] 制造商名称字段 (这里是 2 个字节) 包括制造商的 EISA 3- 字符 ID, 并且按照与 VESA EDID 规范中相同的方式将其打包成 3 个 5- 位的字符。字符 “A” 被表示为二进制的 00001, 字符 “Z” 被表示为二进制的 11010, 而且所有 “A” 和 “Z” 之间的所有字母都被表示为对应于 “A” 和 “Z” 之间的字母表顺序的顺序的二进制值。制造商名称的最高有效位未被使用, 并且通常设置为逻辑 0, 直到用于未来实现为止。例如, 由字符串 “XYZ” 表示的制造商将具有值为 0x633a 的制造商名称。如果客户机不支持该字段, 那么其将被设置为 0。产品代码使用 2 个字节, 来包含由显示器制造商分配的产品代码。如果客户机不支持该字段, 那么该字段将被设置为 0。

[0890] 保留 1、保留 2 和保留 3 字段 (这里是 2 个字节) 保留, 以便将来用于透露信息。这些字段中的所有位通常被设置为 0。这些字段的目的是现在是使得所有连续的 2 字节字段与 16 位地址对准, 并且使得 4 字节字段与 32 位地址对准。

[0891] 在该实施例中, 序列号字段用 4 个字节来表明显示器的数字形式序号。如果客户机不支持该字段, 那么该字段将被设置为 0。制造星期字段用 1 个字节, 来定义该显示器的制造星期。如果客户机支持该字段, 那么该值在 1-53 的范围内。如果客户机不支持该字段, 那么该字段将被设置为 0。制造年份字段是 1 个字节, 其定义了该显示器的制造年份。该值是距离 1990 年的偏移量。该字段表示的年份的范围是 1991-2245。例如, 2003 年对应于值为 13 的制造年份。如果客户机不支持该字段, 那么该字段将被设置为 0。

[0892] CRC 字段 (这里为 2 字节) 包含包括分组长度在内的分组中所有字节的一个 16 位的 CRC 值。

[0893] G. 关于客户机请求和状态分组

[0894] 反向链路请求字段 (3 字节) 指定了客户机在下一子帧中在反向链路上向主机发送信息所需要的字节数目。

[0895] CRC 错误计数字段 (1 字节) 表明自从介质帧开始以来已经出现有多少 CRC 错误。发送子帧计数为零的子帧报头分组时, CRC 计数重置。如果 CRC 错误的实际数目超出 255, 那么这一值通常在 255 饱和。

[0896] 能力改变字段使用 1 字节来表明客户机能力的改变。如果用户连接诸如麦克风、键盘或者显示器之类的外围设备, 或者出于其它原因, 可能出现这种情况。当位 [7:0] 等于 0 时, 那么自从发送了上次的客户机能力分组以后, 能力没有改变。然而, 当位 [7:0] 等于 1

至 255 时,能力已经改变。检查客户机能力分组,以便确定新的显示器特性。

[0897] 客户机繁忙字段使用 2 个字节,表明客户机正在执行特定的功能,尚未准备好接受与该功能相关的其他分组。设置成逻辑 1 电平或值的位表示,客户机当前正在执行该特定功能,客户机中的相关功能部件繁忙。如果客户机中的相关功能部件就绪,则将该位设为逻辑 0。对于客户机中不支持的所有功能,客户机应当返回繁忙状态(位设为 1)。

[0898] 在一个实施例中,这些字节根据如下关系进行解释:如果位 0 是“1”,则位图块传送功能繁忙,如果位 1 是“1”,则位图区域填充功能繁忙,如果位 2 是“1”,则位图模式填充功能繁忙。当前,位 3 至 15 预留将来用,如果将来分配这些位时,通常设为逻辑 1 电平或状态,以表明繁忙状态。

[0899] H. 关于位块传送分组

[0900] 窗口左上坐标 X 值和 Y 值字段均使用 2 字节来指定待移动的窗口左上角的坐标的 X 和 Y 值。窗口宽度和高度字段均使用 2 字节用于说明待移动的窗口的宽度和高度。窗口 X 移动和 Y 移动字段均使用 2 字节用于分别指定窗口水平和垂直移动的像素数目。通常,这些坐标被如此配置,即,X 的正值让窗口将向右侧移动,负值让其向左侧移动,而 Y 的正值让窗口向下移动,且负值使其向上移动。

[0901] I. 关于位图区域填充分组

[0902] 窗口左上坐标 X 值和 Y 值字段均使用 2 字节用于说明待填充的窗口左上角坐标的 X 和 Y 值。窗口宽度和高度字段(均 2 字节)用于说明待填充的窗口的宽度和高度。视频数据格式描述符字段(2 字节)指定像素区域填充值的格式。所述格式与视频流分组中的同一字段相同。像素区域填充值字段(4 字节)包含待填充到由上述字段指定的窗口中的像素值。在视频数据格式描述符字段中指定了这一像素的格式。

[0903] J. 关于位图图案填充分组

[0904] 窗口左上坐标 X 值和 Y 值字段均使用 2 字节来指定待填充的窗口左上角坐标的 X 和 Y 值。窗口宽度和高度字段(均 2 字节)用于说明待填充的窗口的宽度和高度。图案宽度和图案高度字段(均 2 字节)分别用于说明填充图案的宽度和高度。水平图案偏移字段(2 字节)指定了距离待填充的指定窗口的左沿的像素数据图案的水平偏移。该值被指定为小于图案宽度字段中的值。垂直图案偏移字段(2 字节)指定了距离待填充的指定窗口的顶沿的像素数据图案的垂直偏移。该值被指定为小于图案高度字段中的值。

[0905] 2 字节的视频数据格式描述符字段指定像素区域填充值的格式。图 11 说明视频数据格式描述符是如何编码的。所述格式与视频流分组中的同一字段相同。

[0906] 参数 CRC 字段(2 字节)包含从分组长度到视频格式描述符的所有字节的 CRC。如果这一 CRC 未能通过校验,就丢弃整个分组。图案像素数据字段包含原始视频信息,用于说明以视频数据格式描述符指定的格式的填充图案。数据被打包为字节,并且每一行的第一像素将按字节对准。所述填充图案数据每次传输一行。图案像素数据 CRC 字段(2 字节)包含只对图案像素数据的 CRC。如果这一 CRC 未能通过校验,那么所述图案像素数据仍可被使用,但是 CRC 错误计数增加。

[0907] K. 通信链路数据通道分组

[0908] 参数 CRC 字段(2 字节)包含从分组长度到分组类型的所有字节的一个 16-位 CRC。如果这一 CRC 未能通过校验,就丢弃整个分组。

[0909] 通信链路数据字段包含来自通信信道的原始数据。这一数据仅仅被传递到显示器中的计算设备上。

[0910] 通信链路数据 CRC 字段 (2 字节) 包含只对通信链路数据的 16- 位 CRC。如果这一 CRC 未能通过校验,那么所述通信链路数据仍可使用或者仍是有用的,不过 CRC 错误计数增加。

[0911] L. 关于接口类型切换请求分组

[0912] 所述接口类型字段 (1 字节) 指定要使用的新的接口类型。这个字段中的值按照以下方式指定接口类型。如果位 7 的值等于 '0',那么类型切换请求是针对正向链路的,如果等于 '1',那么类型切换请求是针对反向链路的。位 6 至 3 被保留以供将来使用并且通常设置为零。位 2 至 0 用来定义将使用的接口类型,值 1 意味着切换至类型 1 模式,值 2 意味着切换至类型 2 模式,值 3 意味着切换至类型 3 模式,而值 4 意味着切换至类型 4 模式。值 '0' 以及 5 至 7 被保留以供将来可选模式或者模式组合的标记使用。

[0913] M. 关于接口类型确认分组

[0914] 接口类型字段 (1 字节) 具有用于确认要使用的新的接口类型的值。这个字段中的值按照以下方式指定接口类型。如果位 7 的值等于 '0',那么类型切换请求是针对正向链路的,或者,如果等于 '1',那么类型切换请求是针对反向链路的。位位置 6 至 3 目前被保留以便根据需要用于指明其它切换类型,并且通常设置为零。然而,位位置 2 至 0 用于定义将使用的接口类型,其中值 '0' 表示否定确认,或者所请求的切换无法执行,值 '1'、'2'、'3' 和 '4' 分别表示切换至类型 1、类型 2、类型 3 和类型 4 模式。值 5 至 7 被保留以便根据需要用作模式的可选标记。

[0915] N. 关于执行类型切换分组

[0916] 1 字节的接口类型字段表明要使用的新的接口类型。这个字段中存在的值通过首先使用位 7 的值确定是对正向还是反向链路进行类型切换来指定接口类型。值 '0' 表明类型切换请求是针对正向链路的,而值 '1' 表明类型切换请求是针对向反向链路的。位 6 至 3 被保留以供将来使用并且通常设置为零值。然而,位 2 至 0 用来定义将使用的接口类型,值 1、2、3 和 4 分别指定切换至类型 1、类型 2、类型 3 和类型 4 模式。这些位的值 0 以及 5 至 7 的使用被保留以供将来使用。

[0917] O. 关于正向音频通道启用分组

[0918] 音频通道启用掩码字段 (1 字节) 包含一组标志,用于表明哪些音频通道将会在客户机中被启用。设置为 1 的位启用相应的通道,设置为零的位禁用相应的通道。位 0 至 5 分别指明位于左前、右前、左后、右后、前方中心和超低音扬声器通道的通道 0 到通道 5。位 7 至 6 被保留以供将来使用,同时通常设置为零。

[0919] P. 关于反向音频采样速率分组

[0920] 音频采样速率字段 (1 字节) 指定数字音频采样速率。把这个字段的值分配给不同的速率,其中值 0、1、2、3、4、5、6、7 和 8 分别用于指明 8,000、16,000、24,000、32,000、40,000、48,000、11,025、22,050 和 44,100 采样数每秒 (SPS),值 9 至 254 被保留以便根据需要供可选速率使用,因此它们目前被设置为 '0'。值 255 用来禁用反向链路音频流。

[0921] 采样格式字段 (1 字节) 指定数字音频采样的格式。当位 [1:0] 等于 '0' 时,数字音频采样是线性格式的,当它们等于 1 时,数字音频采样是  $\mu$ -律格式的,而当它们等于 2

时,数字音频采样是 A-律格式的。位 [7:2] 被保留以便根据需要可替代地用于指明音频格式,并且通常设置为等于零。

[0922] Q. 关于数字内容保护开销分组

[0923] 内容保护类型字段 (1 字节) 指定所使用的数字内容保护方法。值 '0' 表明数字传输内容保护 (DTCP), 而值 1 表明高带宽数字内容保护系统 (HDCP)。2 至 255 的值范围目前为被指定, 而是被保留以便根据需要用于可选保护方案。内容保护开销消息字段是包含在主机和客户机之间发送的内容保护消息的长度可变字段。

[0924] R. 关于透明色彩启用分组

[0925] 透明色彩启用字段 (1 字节) 指定启用或者禁用透明色彩模式的情况。如果位 0 等于 0, 那么透明色彩模式被禁用, 如果等于 1, 那么透明色彩模式被启用, 并且透明色彩通过如下两个参数指定。这一字节的位 1 至 7 被保留以供将来使用并且通常设置为等于零。

[0926] 视频数据格式描述符字段 (2 字节) 指定像素区域填充值的格式。图 11 说明视频数据格式描述符是如何编码的。所述格式通常与视频流分组中的同一字段相同。

[0927] 像素区域填充值字段使用 4 字节分配给待填充到如上指定的窗口中的像素值。在视频数据格式描述符字段中指定这一像素的格式。

[0928] S. 关于往返行程延迟测量分组

[0929] 2 字节的分组长度字段指定了不包括分组长度字段的分组中的字节总数, 并且在实施例中选择 159 的固定长度。2 字节的分组类型字段用值 82 标识该分组类型, 其将一个分组标识为往返行程延迟测量分组。如前一样, hClient ID 字段被保留以供将来用作 Client ID, 且通常被设为 0。

[0930] 在一个实施例中, 参数 CRC 字段 (2 字节) 包含从分组长度到分组类型的所有字节的 16- 位 CRC。如果这一 CRC 未能通过校验, 就丢弃整个分组。

[0931] 保护时间 1 字段 (这里是 64 字节) 用来允许客户机中的 MDDI\_Data 线路驱动器在主机中的线路驱动器被禁用以前被启用。在保护时间 1 的位 0 期间, 所述客户机启用 MDDI\_Data 线路驱动器, 并且主机禁用其线路驱动器以便在保护时间 1 的最后位之前, 该线路驱动器被完全禁用。在保护时间 1 期间, 主机和客户机在其未被禁用时都驱动逻辑零电平。该字段的另一个目的是确保: 在禁用主机线路驱动器之前, 所有 MDDI\_Data 信号处于逻辑零电平达足够的时间, 以便允许客户机仅仅利用 MDDI\_Stb 开始恢复时钟或时钟信号

[0932] 测量周期字段是 64 字节窗口, 用于允许客户机以正向链路上使用的一半数据速率利用 0xff、0xff 和 30 个字节的 0x0 来作出响应。这一数据速率对应于反向链路速率除数为 1 的情况。客户机一旦发觉处于测量周期开始时, 就立即返回这一响应。在测量周期的第一位在主机处开始之后经过精确的链路往返行程延迟加上客户机的逻辑延迟的时刻, 主机接收来自客户机的这一响应。

[0933] 全零 1 字段 (2 字节) 包含多个零, 以便允许主机和客户机中的 MDDI\_Data 线路驱动器交迭, 从而 MDDI\_Data 总是被驱动。主机在保护时间 2 的位 0 期间启用 MDDI\_Data, 而客户机按照其在测量周期末尾的做法, 也将信号驱动到逻辑 0 电平。

[0934] 保护时间 2 字段 (64 字节) 中的值在往返行程延迟处于测量周期能够测量的最大值时, 允许由客户机驱动测量周期交迭。客户机在保护时间 2 的位 0 期间禁用其线路驱动器, 而主机在保护时间 2 的最后一位之后立即启用其线路驱动器。在保护时间 2 期间, 主

机和客户机二者在其未被禁用时都驱动逻辑 0 电平。该字段的另一目的是确保所有 MDDI\_Data 信号处于逻辑 0 电平达足够的时间,以便允许客户机在启用主机的线路驱动器之后,使用 MDDI\_Data0 和 MDDI\_Stb 开始恢复时钟信号。

[0935] T. 关于正向链路偏离校准分组

[0936] 在一个实施例中,参数 CRC 字段(2 字节)包含从分组长度到分组类型的所有字节的 16-位 CRC。如果这一 CRC 未能通过校验,就丢弃整个分组。

[0937] 全零 1 字段使用 1 个字节,确保在参数 CRC 字段结束时在 MDDI\_Stb 上有转变。

[0938] 校准数据序列字段包含数据序列,用于让 MDDI\_Data 信号在每一数据周期翻转。校准数据序列字段的长度取决于上行链路上使用的接口。在处理校准数据序列期间,所述 MDDI 主机控制器把所有 MDDI\_Data 信号设置为等于选通信号。校准数据序列字段由客户机显示器接收的时候,所述客户机时钟恢复电路应该只使用 MDDI\_Stb 而不是 MDDI\_Stb 与 MDDI\_Data0 的异或来恢复数据时钟。根据校准数据序列字段开始时 MDDI\_Stb 信号的准确相位,基于发送这个分组时正在使用的接口类型,校准数据序列通常将是如下情况之一:

[0939] 类型 1-(64 字节数据序列)0xaa,0xaa……或者 0x55,0x55……

[0940] 类型 2-(128 字节数据序列)0xcc,0xcc……或者 0x33,0x33……

[0941] 类型 3-(256 字节数据序列)0xf0,0xf0……或者 0x0f,0x0f……

[0942] 类型 4-(512 字节数据序列)0xff,0x00,0xff,0x00……或者 0x00,0xff,0x00,0xff……

[0943] 图 62A 和 62B 中分别示出了类型 1 和类型 2 接口的可能的 MDDI\_Data 和 MDDI\_Stb 波形的实例。

[0944] XVII. 结束语

[0945] 虽然已经描述了本发明的各种实施例,但是应该理解的是,仅仅是以举例方式来给出它们,而不是限制性的。因此,本发明的广度和范围不应该受到上述任何示例性实施例的限制,而是仅仅根据如下的权利要求书及其等效物来限定。

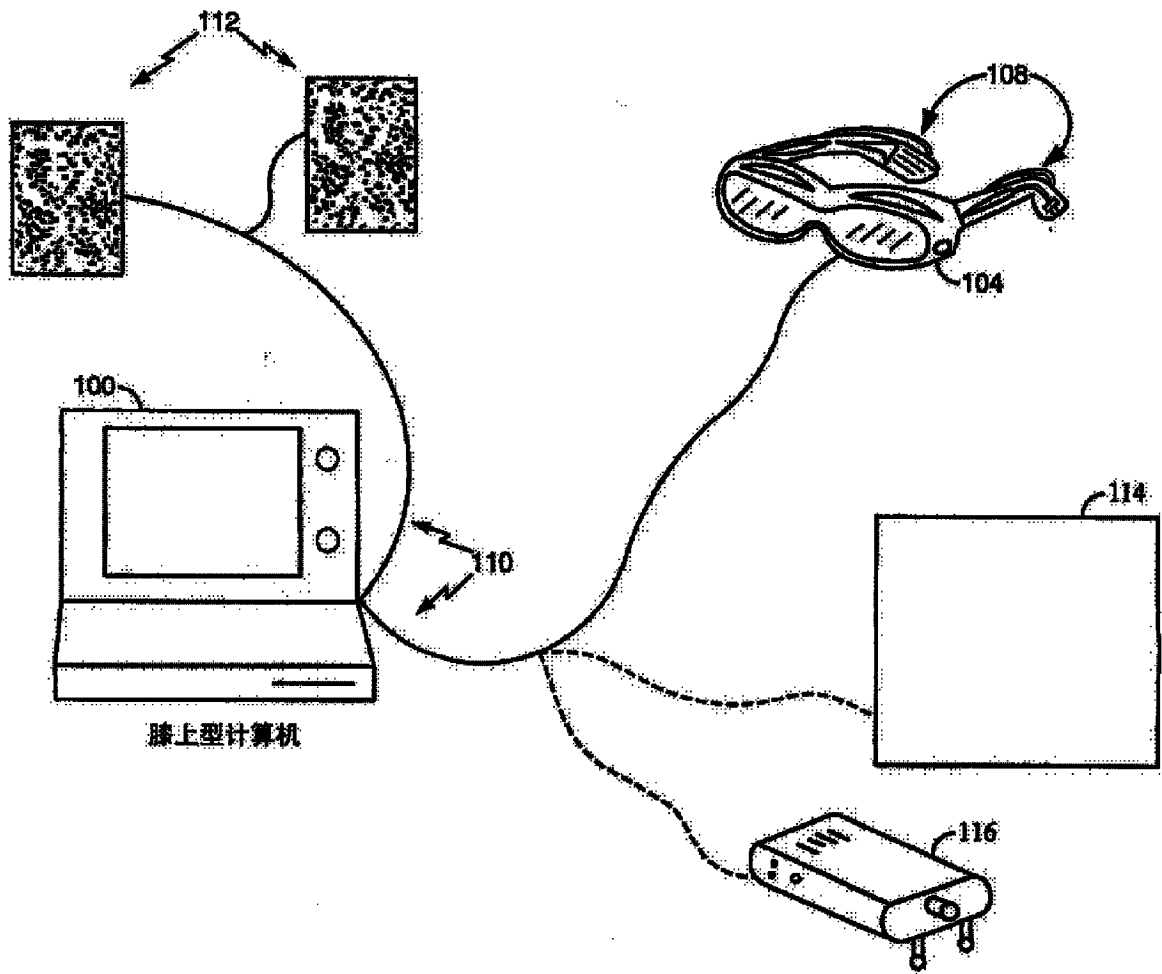


图 1A

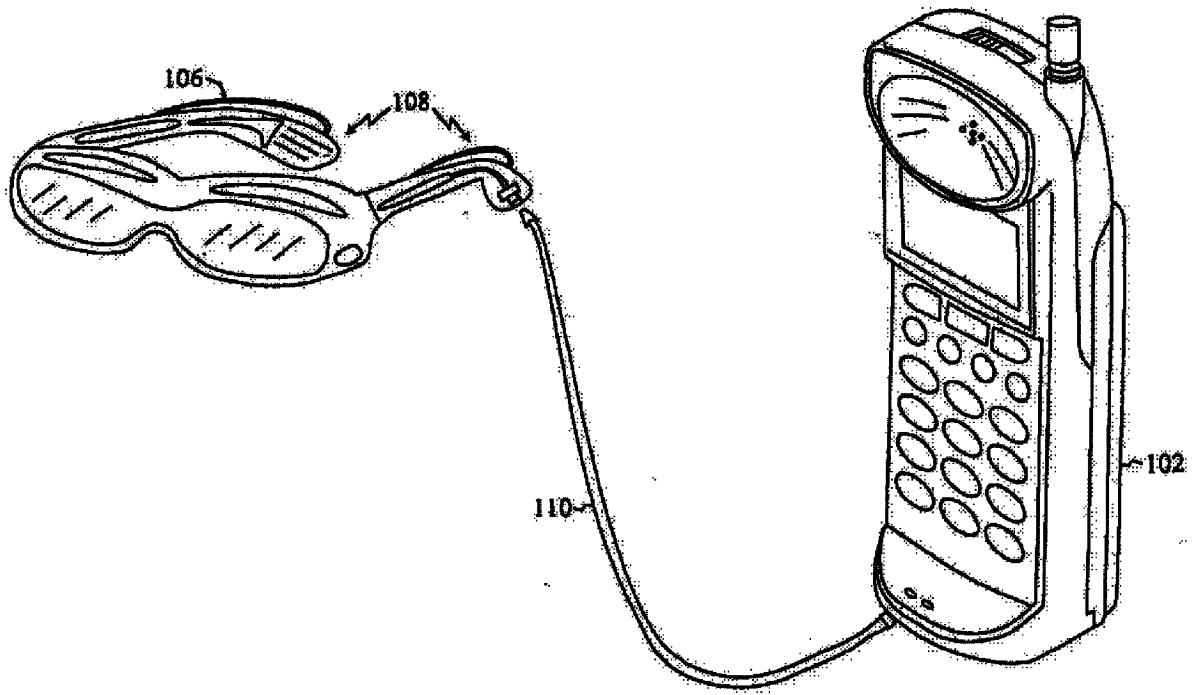


图 1B

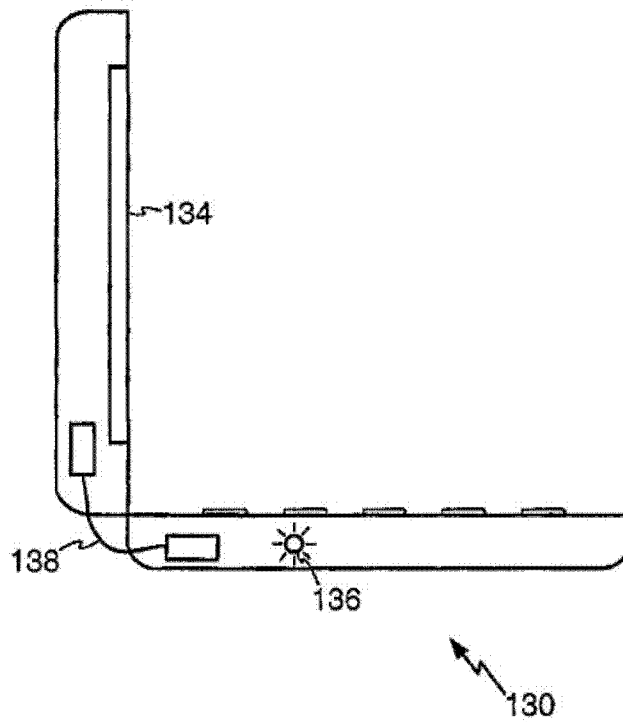


图 1C



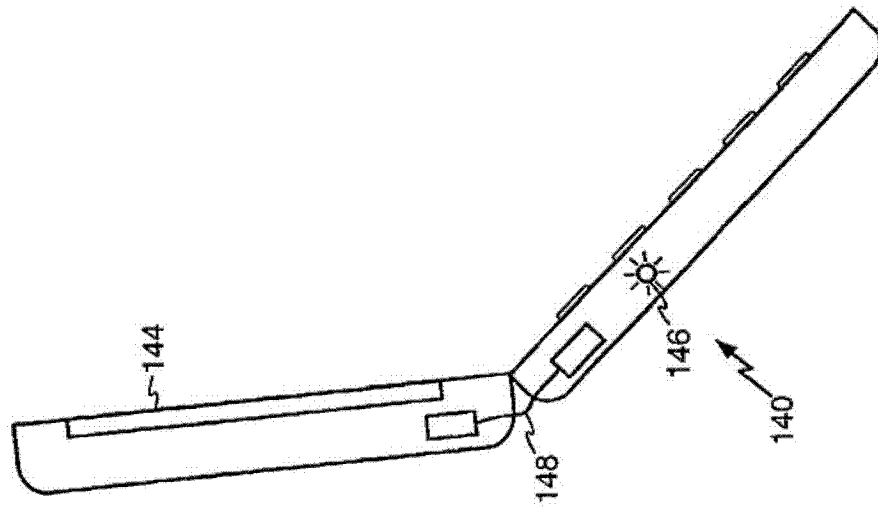


图 1D

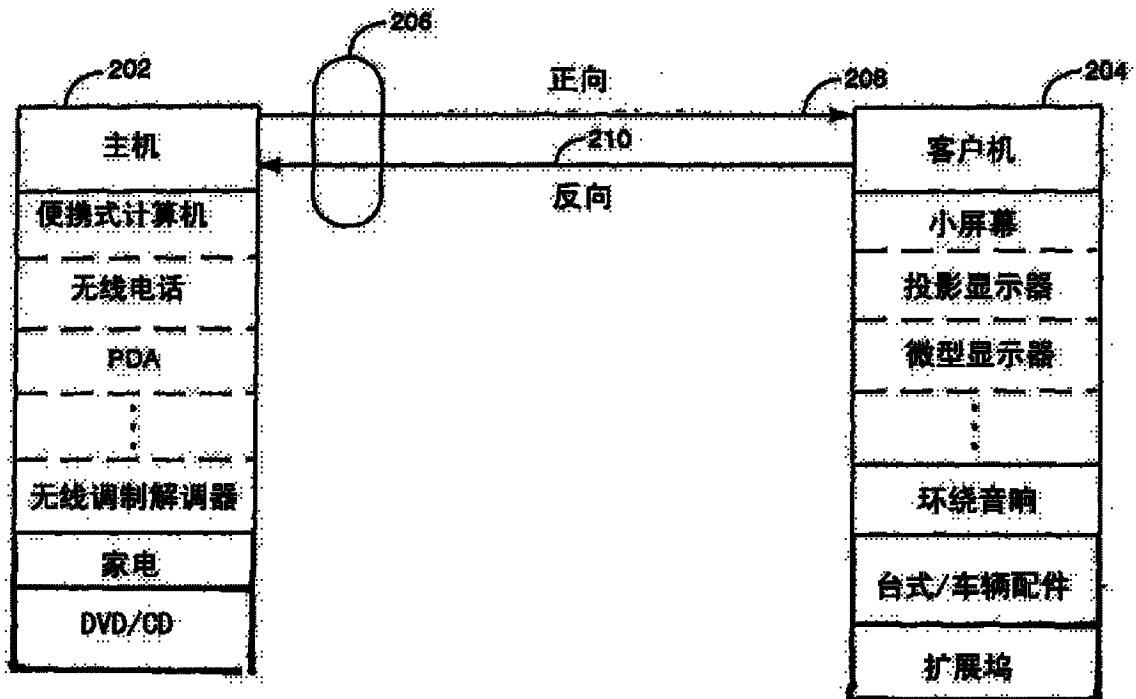


图 2

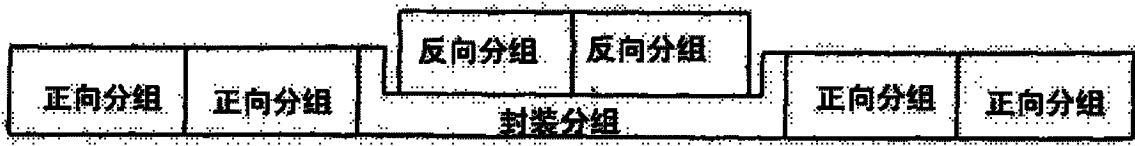


图 3

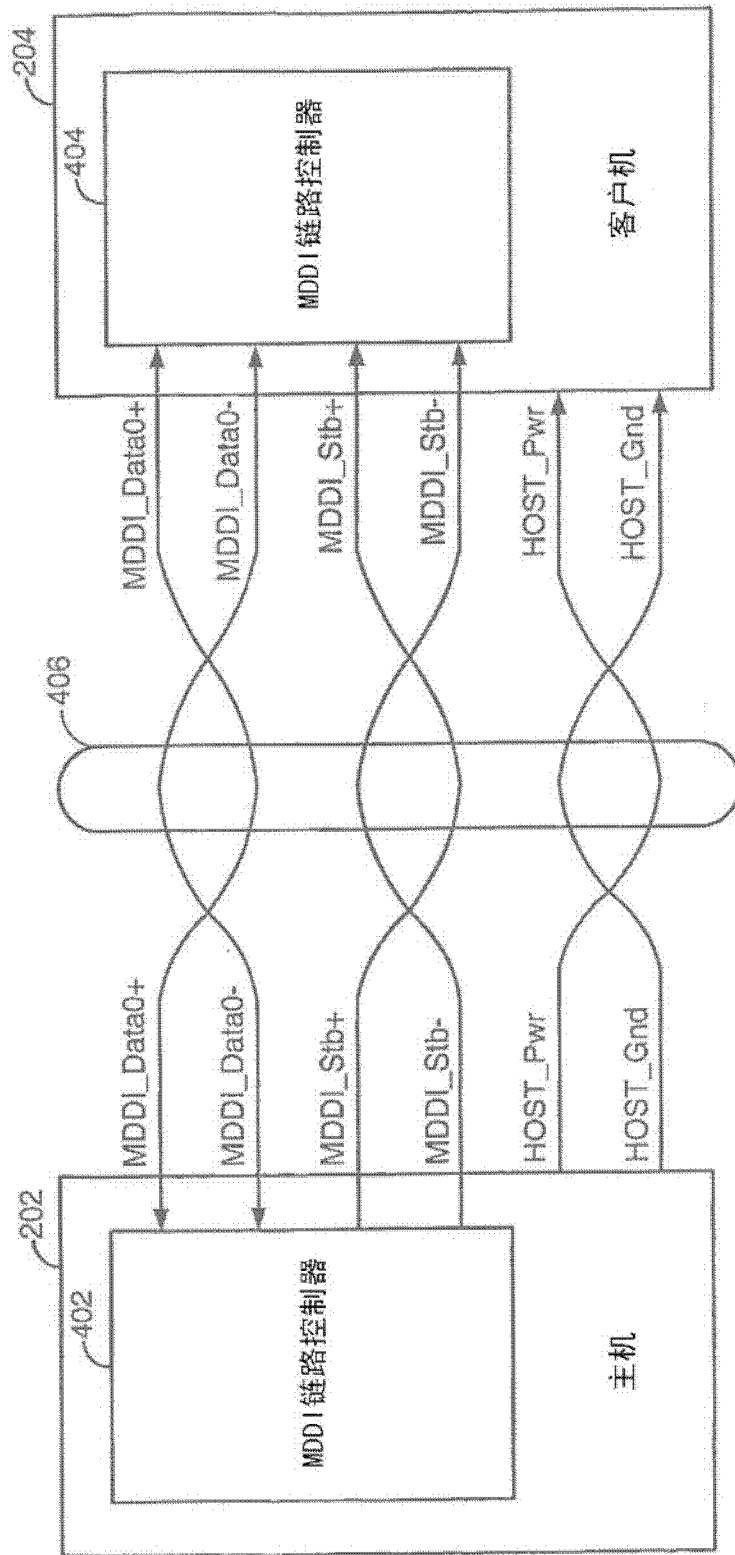


图 4

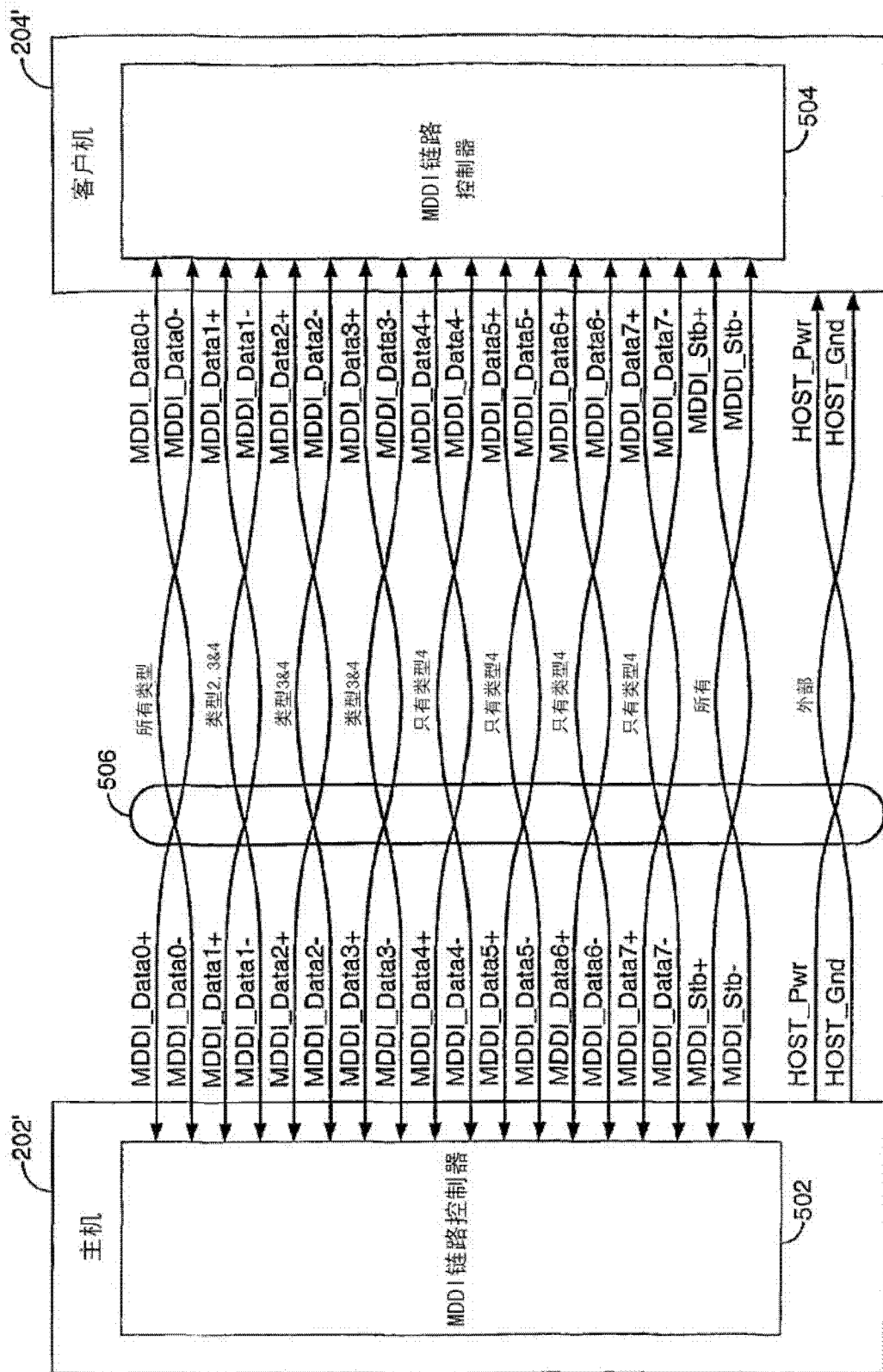


图 5

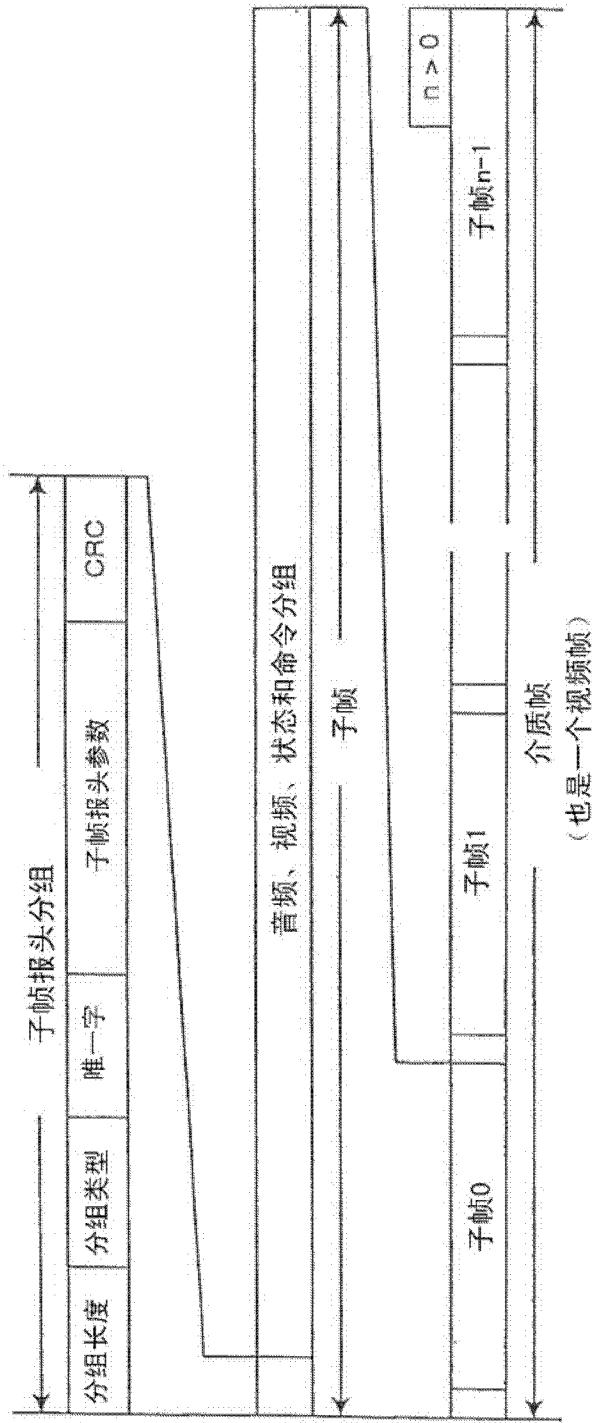


图 6



图 7

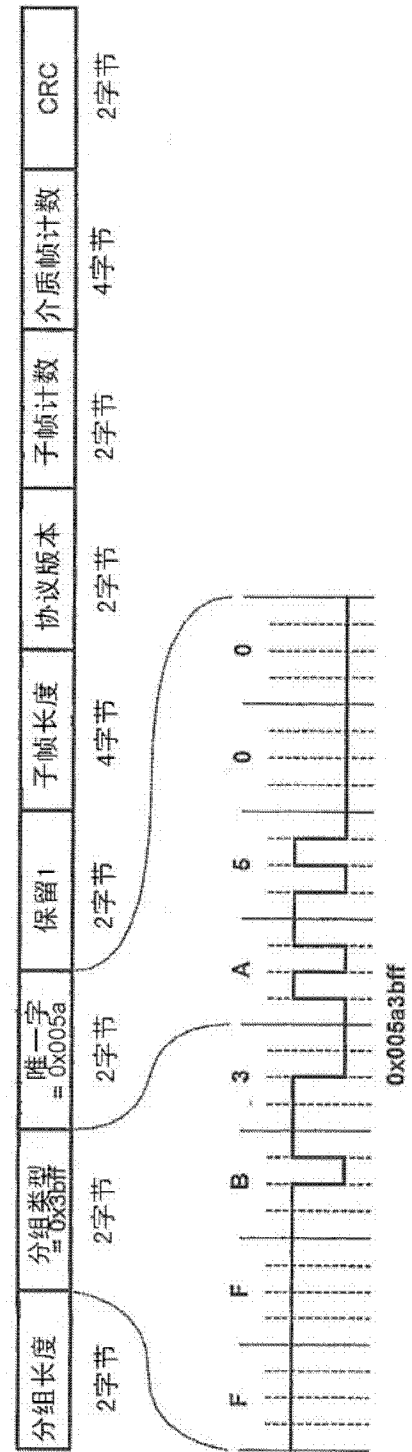


图 8

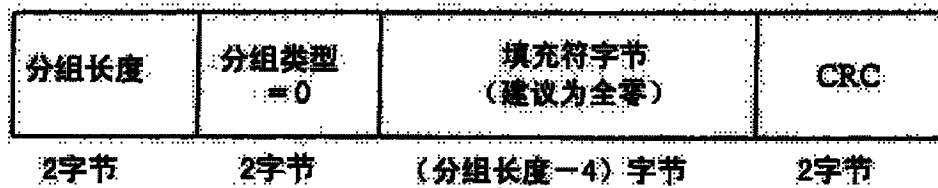


图 9

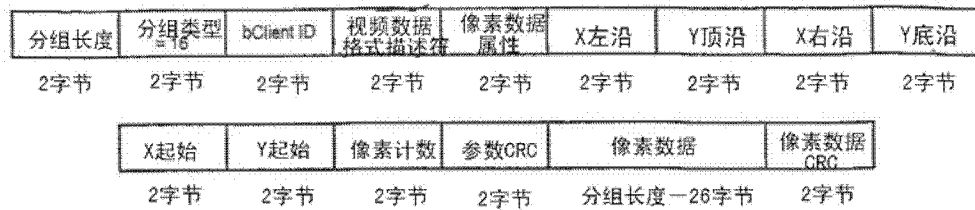


图 10

视频数据格式描述符—单色

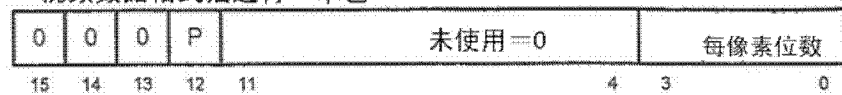


图 11A

视频数据格式描述符—使用色彩映射的色彩

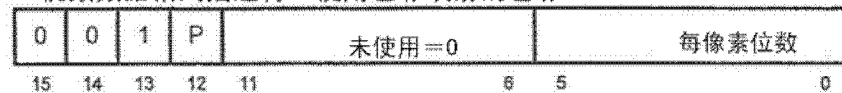


图 11B

视频数据格式描述符—色彩、原始RGB

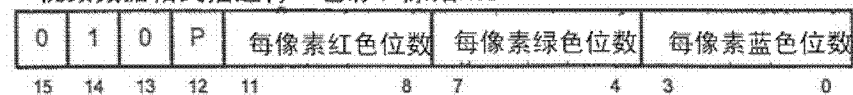


图 11C

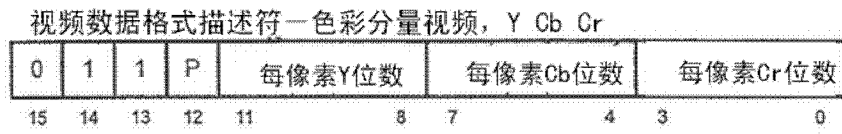


图 11D

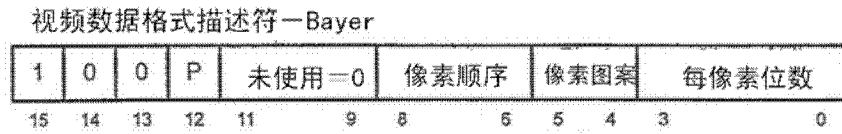


图 11E

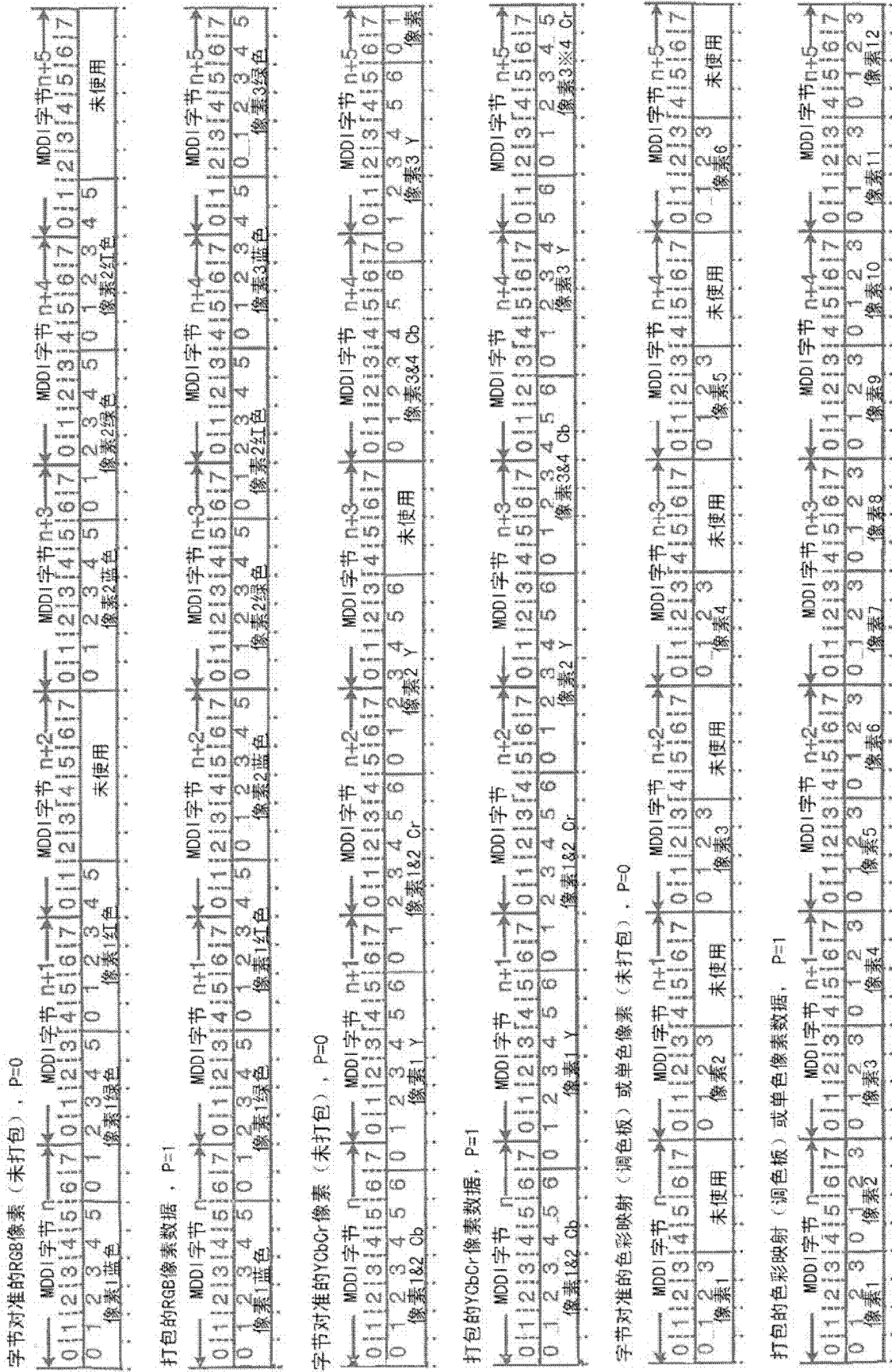
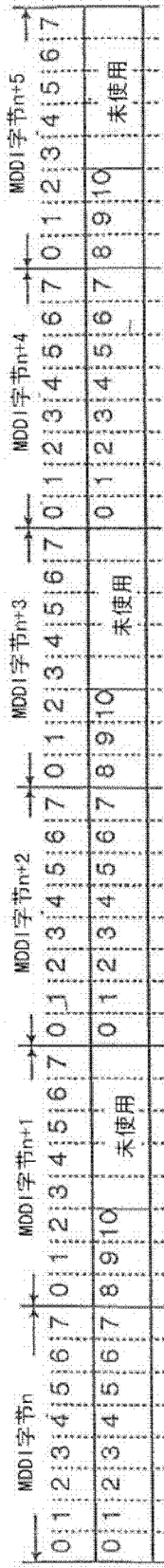


图 12



字节对准的（未打包）PCM音频采样

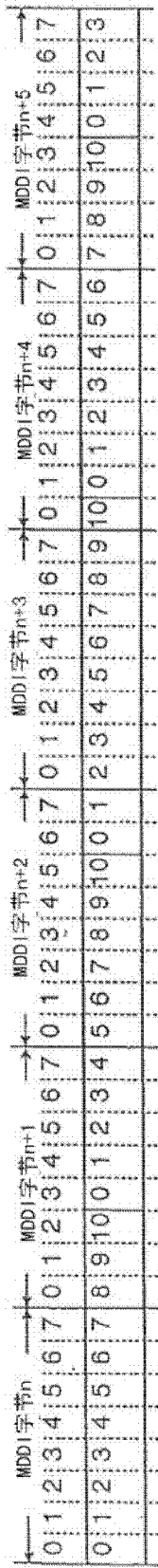


音频采样2

音频采样1

音频采样0

打包的PCM音频采样



音频采样3

音频采样2

音频采样1

音频采样0

图 14

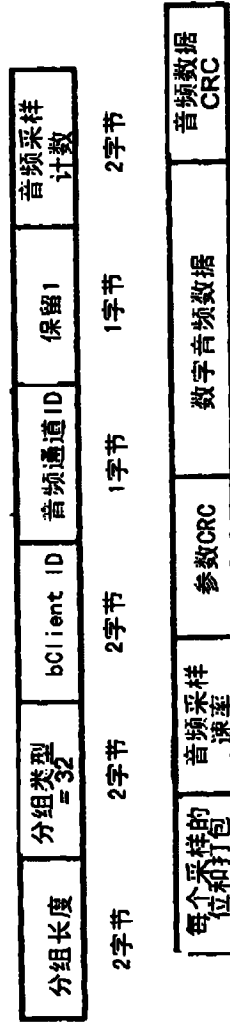


图 13

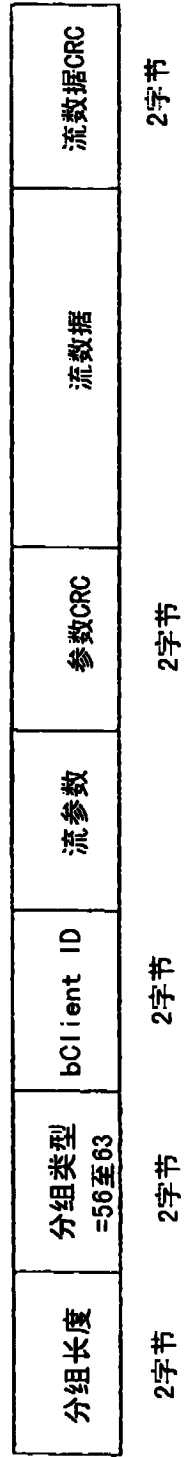


图 15



图 16

反向链路封装分组

分组长度	分组类型 =65	hClient ID	反向链路标志	反向速率除数	转向1长度	转向2长度
2字节	2字节	2字节	1字节	1字节	1字节	1字节
参数CRC	全零1	转向1	反向数据分组		转向2	全零2
2字节	8字节	x字节	(分组_长度-X-Y-26)字节		y字节	8字节

图 17

客户机能力分组

分组长度	分组类型 Type = 66	cClient ID	协议版本	最小协议版本	数据速率能力	接口类型能力
2字节	2字节	2字节	2字节	2字节	2字节	1字节
可选显示器数目	保留1	位图宽度	位图高度	显示窗宽度	显示窗高度	色彩映射大小
1字节	2字节	2字节	2字节	2字节	2字节	4字节
色彩映射 RGB宽度	RGB能力	单色能力	保留2	Y Cb Cr能力	Bayer能力	阿尔法光标 图像平面
2字节	2字节	1字节	1字节	2字节	2字节	2字节
客户机特征能力	最大视频帧速率	最小视频帧速率	最小子帧速率	音频缓冲器深度	音频信道能力	音频采样速率能力
4字节	1字节	1字节	2字节	2字节	2字节	2字节
音频采样分辨率	麦克风采样分辨率	麦克风采样速率能力	键盘数据格式	指示设备数据格式	内容保护类型	制造商名称
1字节	1字节	2字节	1字节	1字节	2字节	2字节
产品代码	保留3	序号	制造星期	制造年份	CRC	
2字节	2字节	4字节	1字节	1字节	2字节	

图 18

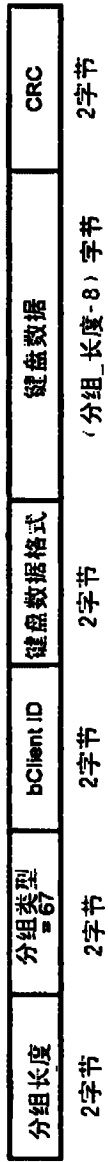


图 19

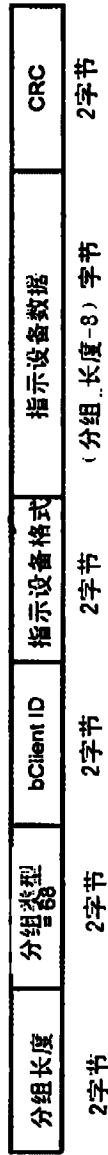


图 20

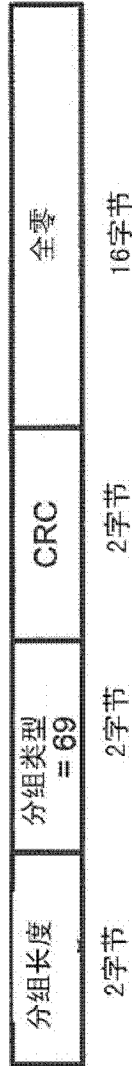


图 21

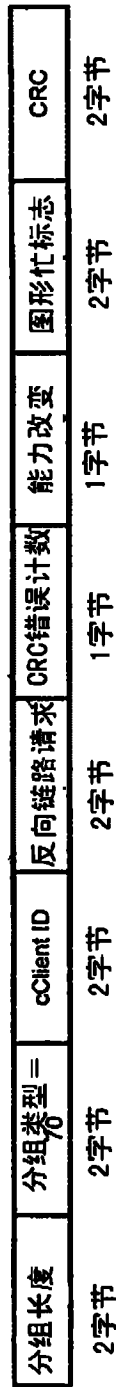


图 22

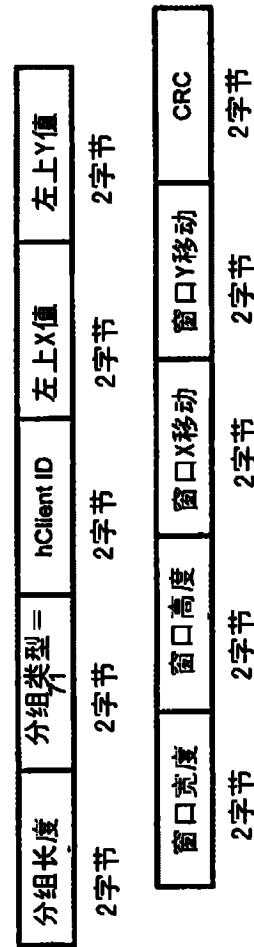


图 23

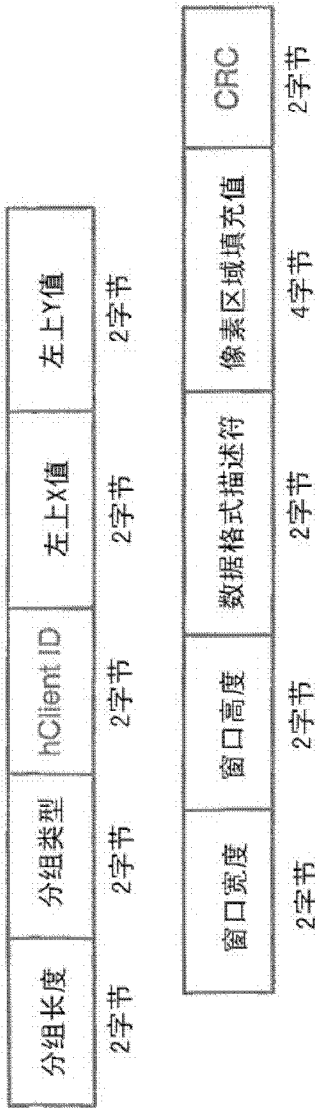


图 24

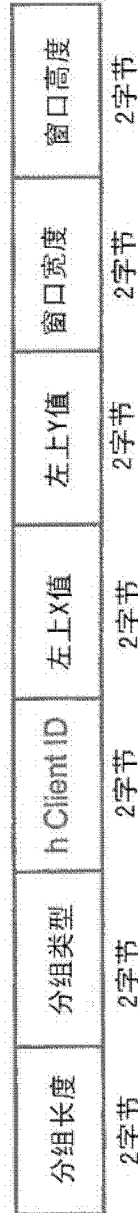


图 25

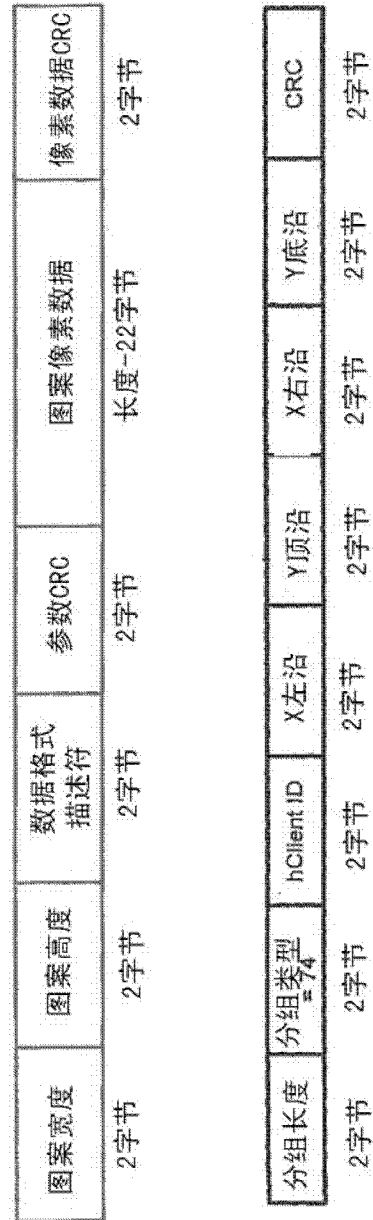


图 26

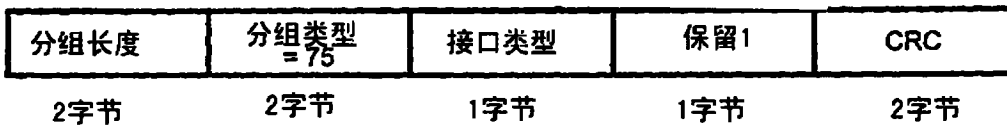


图 27

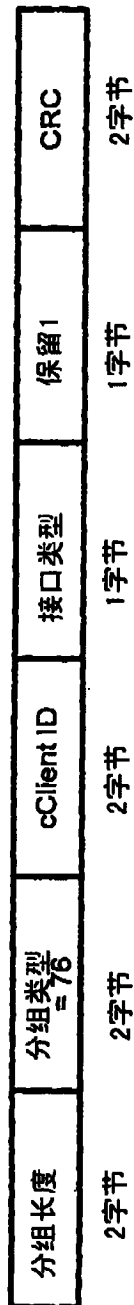


图 28

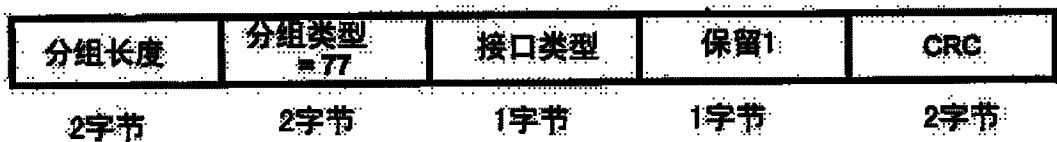


图 29

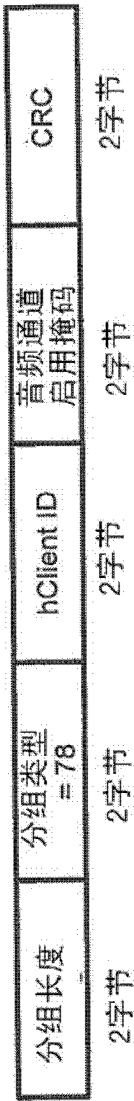


图 30

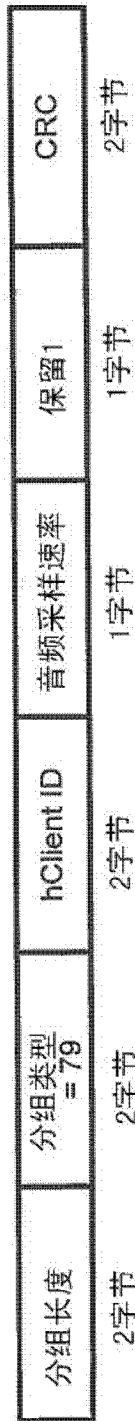


图 31



图 32

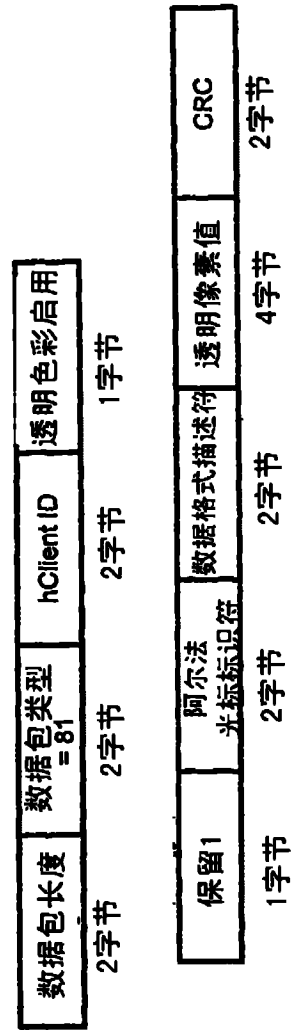


图 33

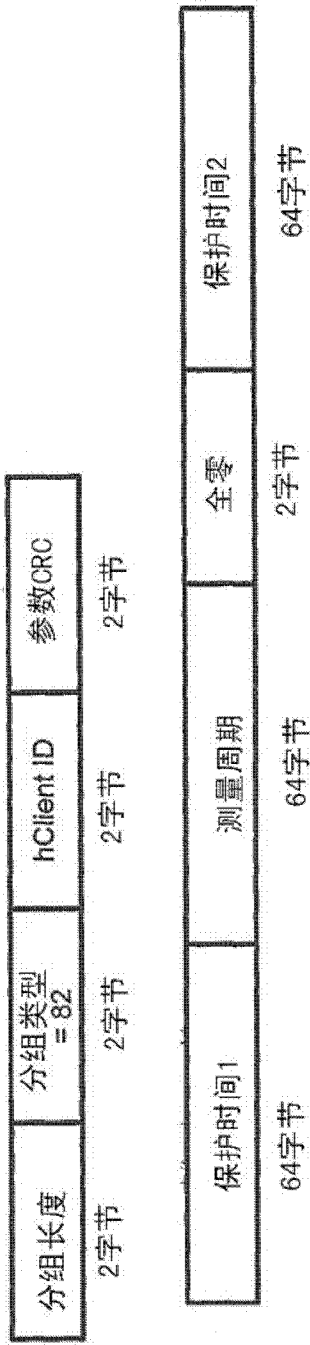


图 34

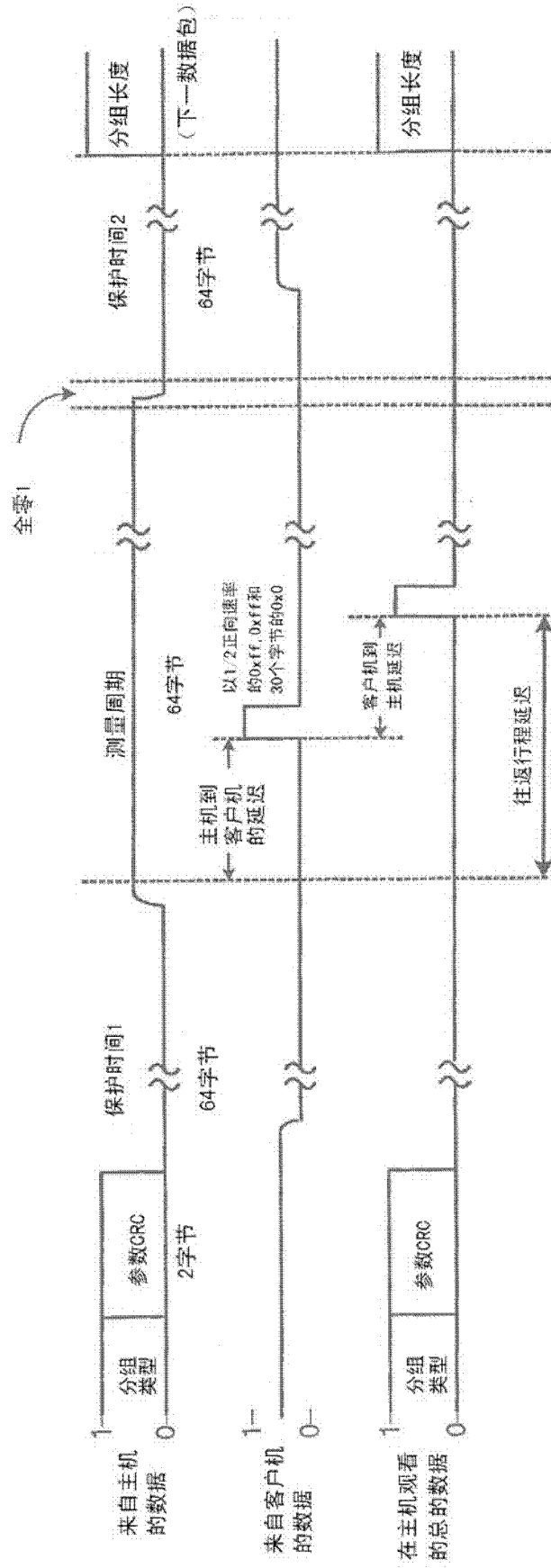


图 35



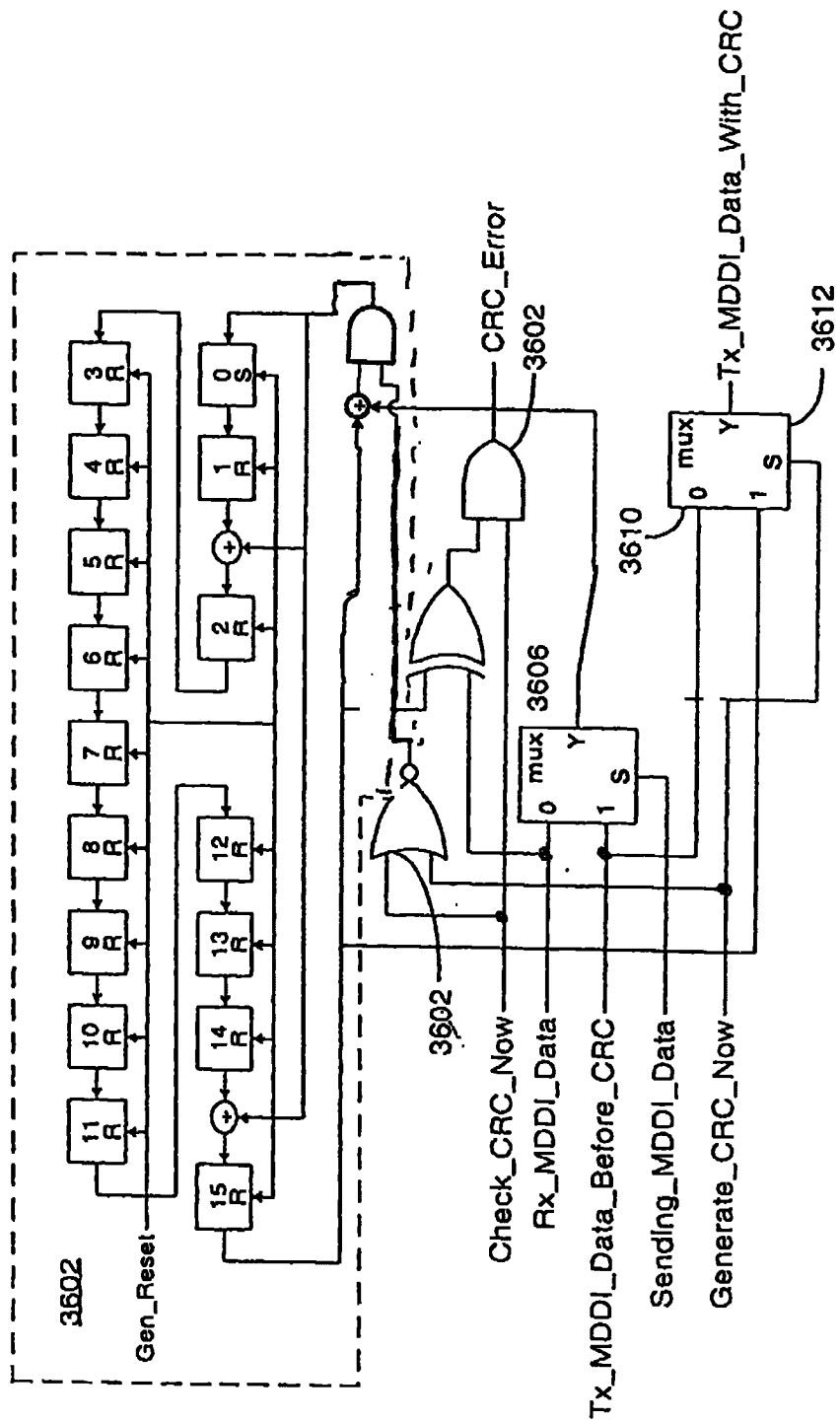


图 36

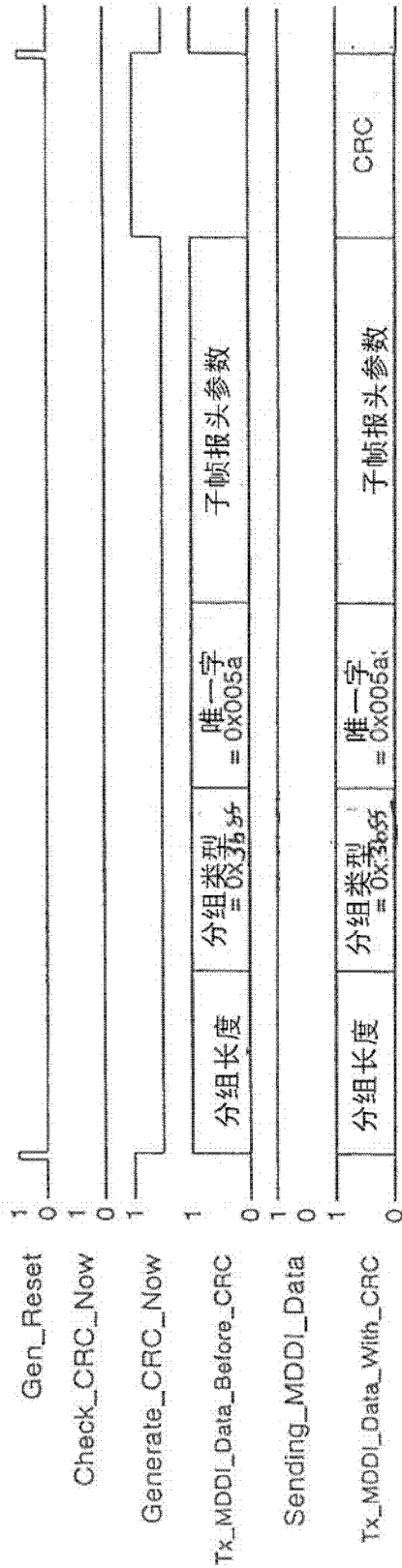


图 37A

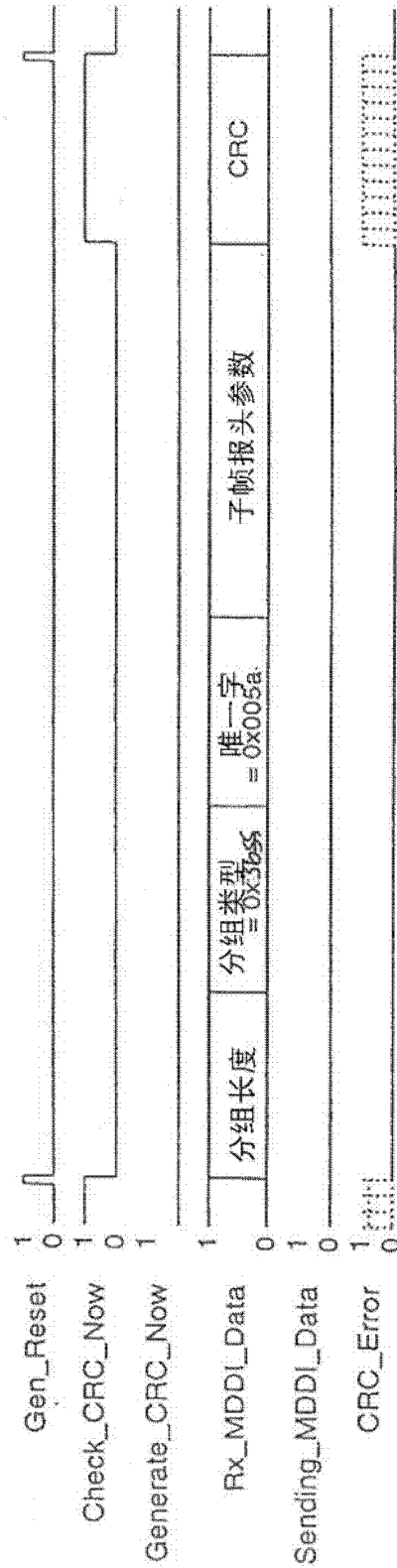


图 37B

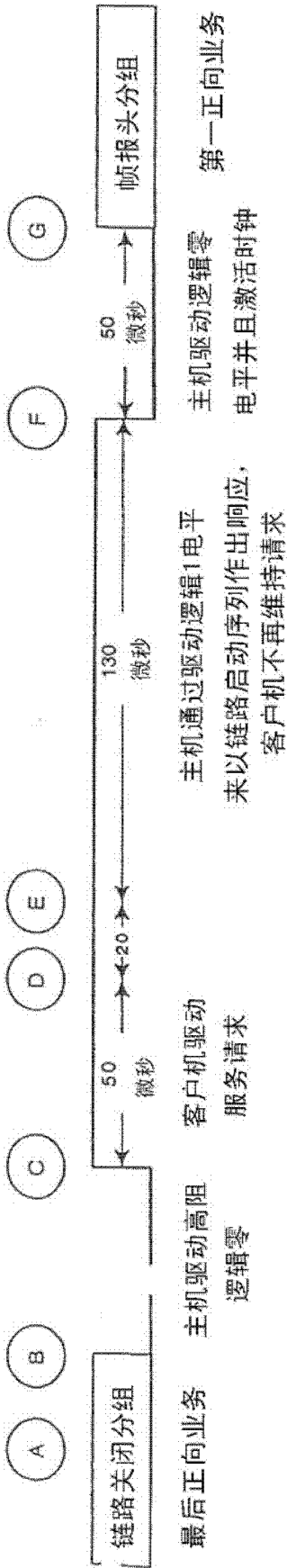


图 38

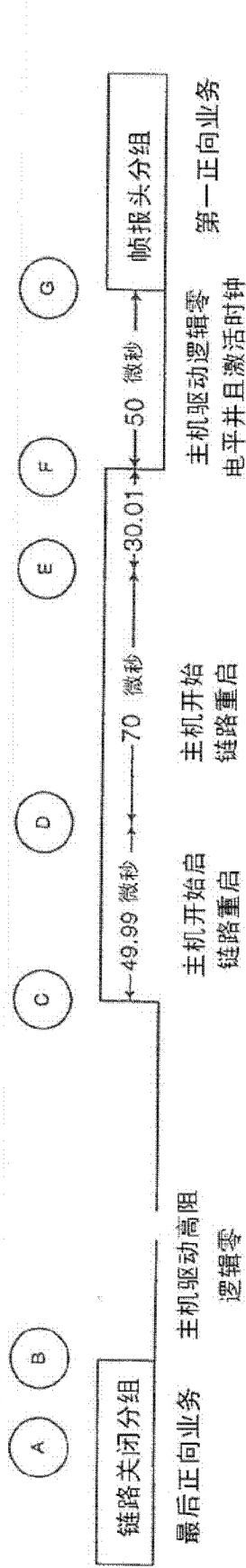


图 39

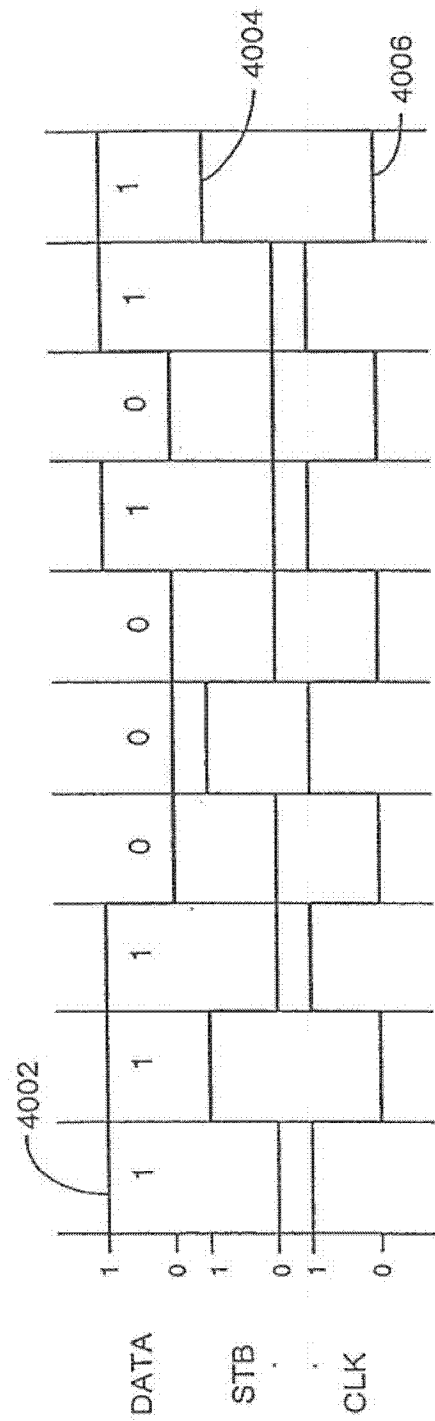


图 40

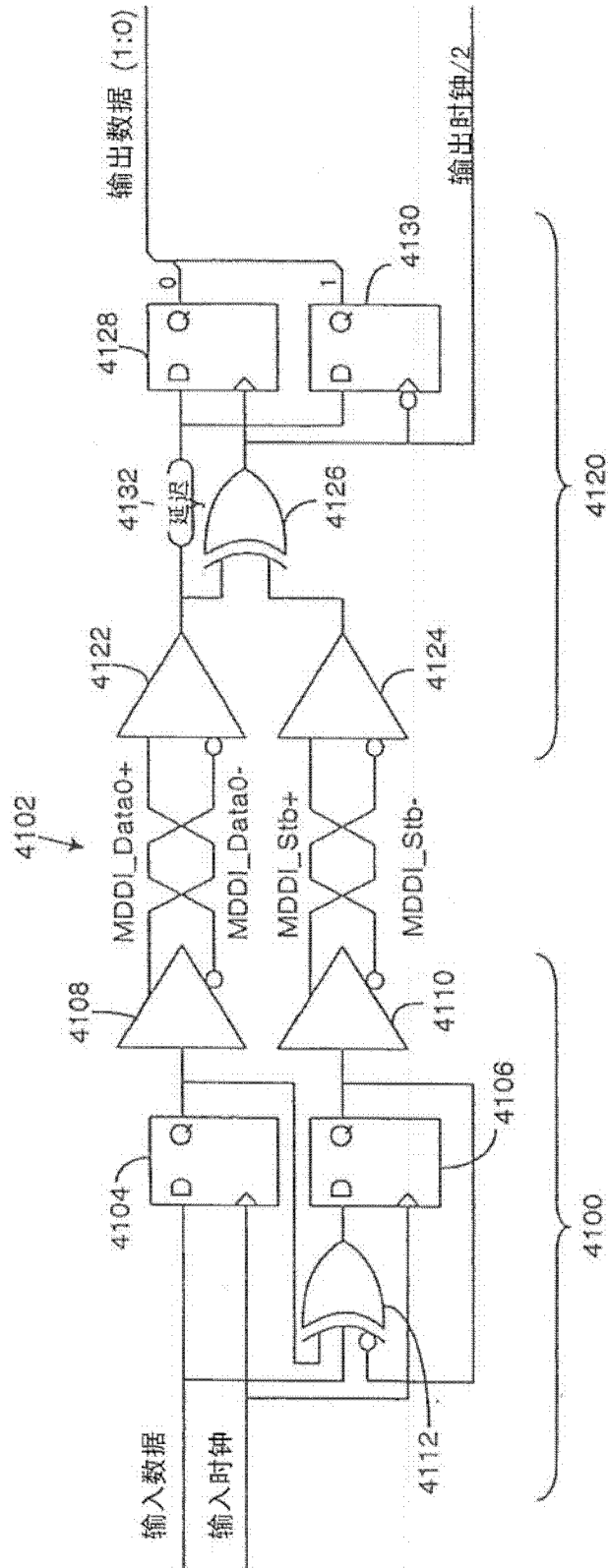


图 41

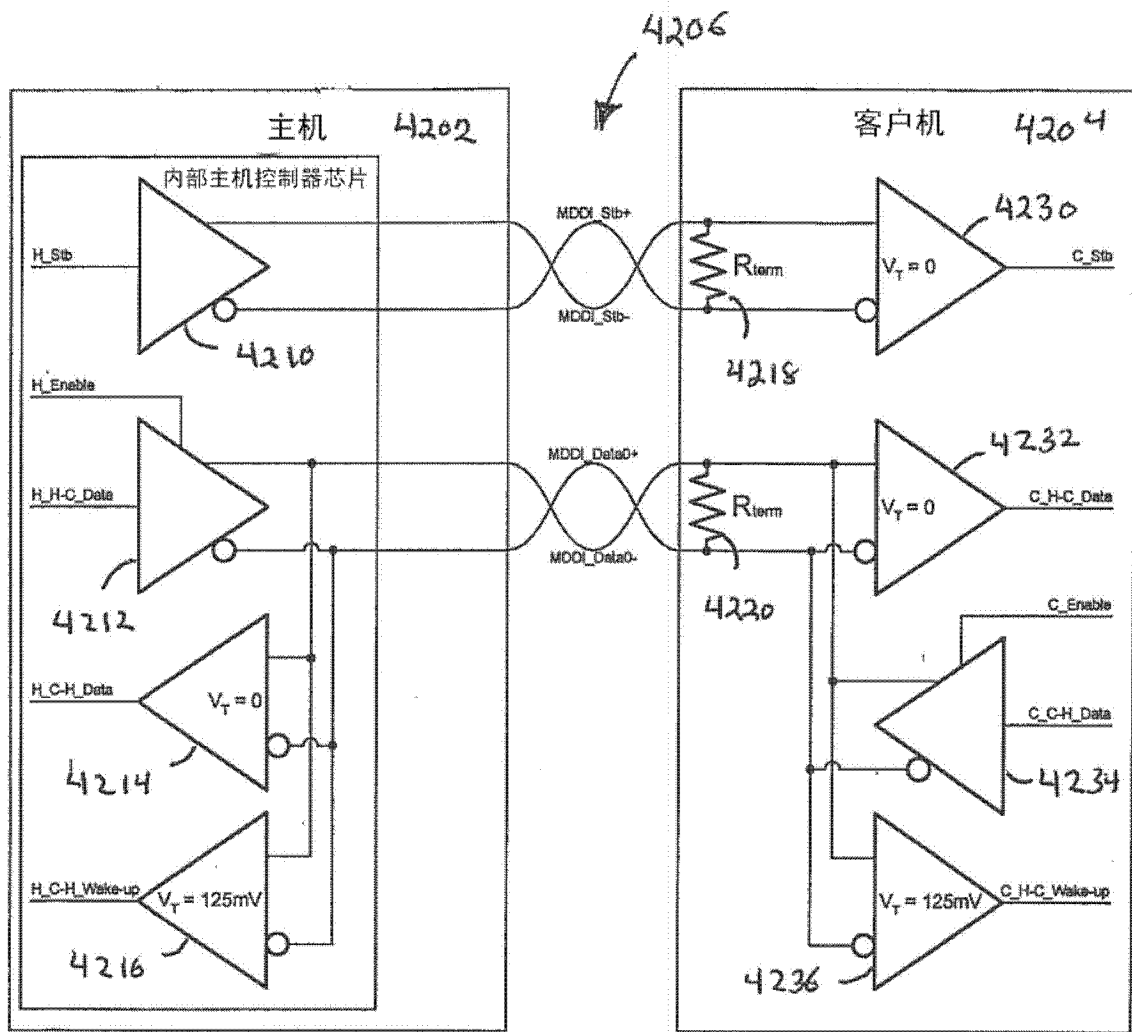


图 42

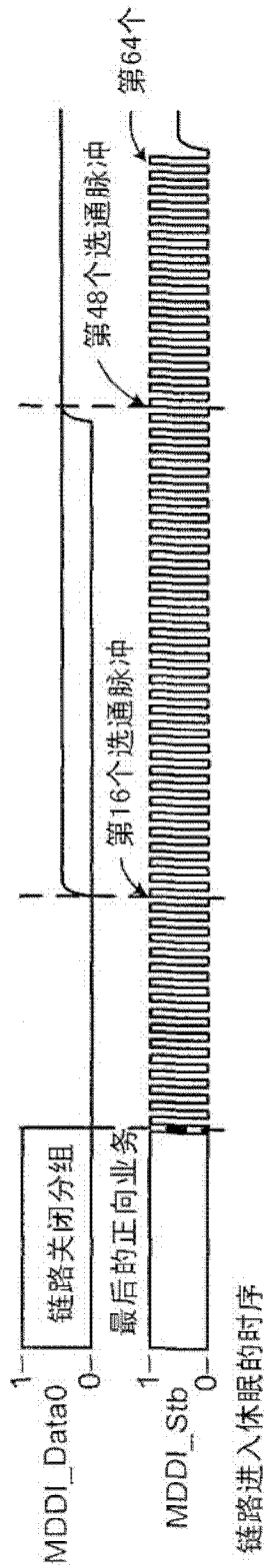


图 43a

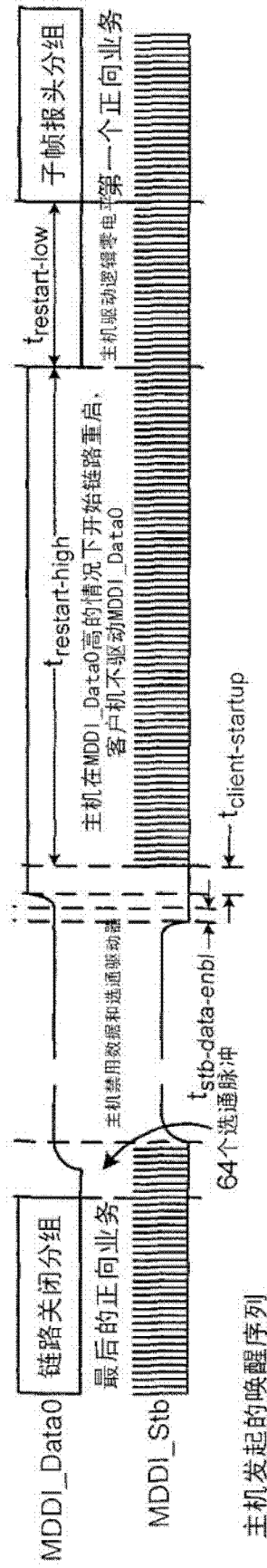


图 43b

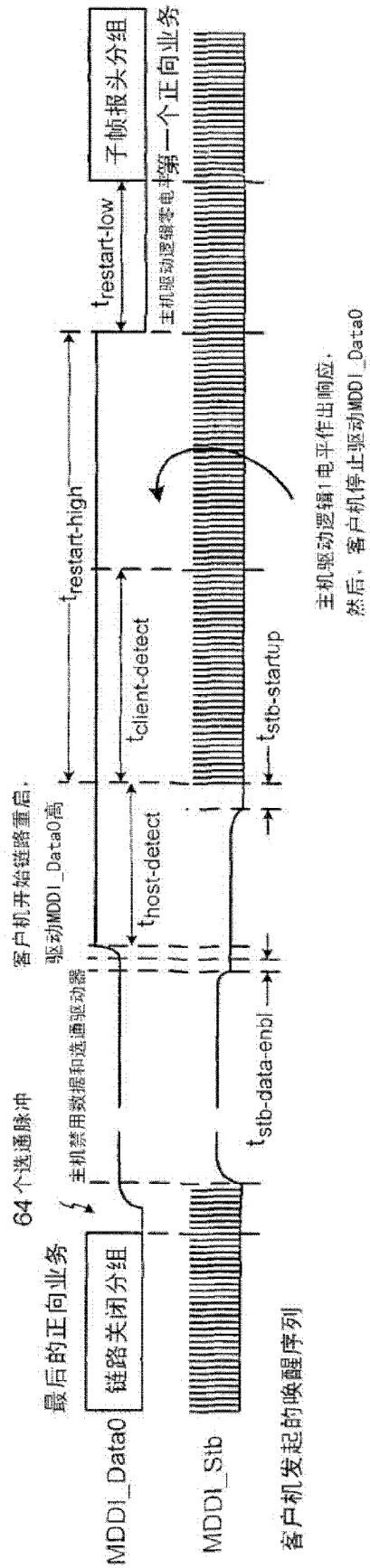


图 43c

主机驱动逻辑1电平作出响应。  
然后，客户机停止驱动MDDI\_Data0

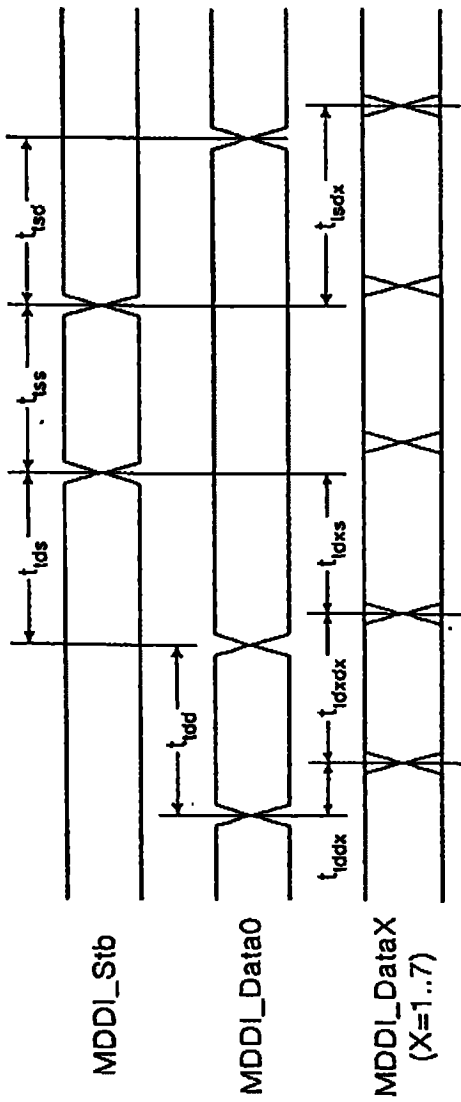


图 44

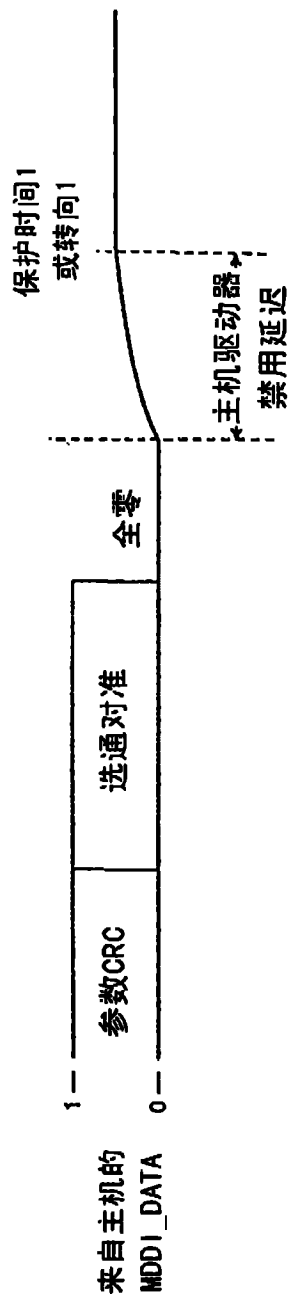


图 45

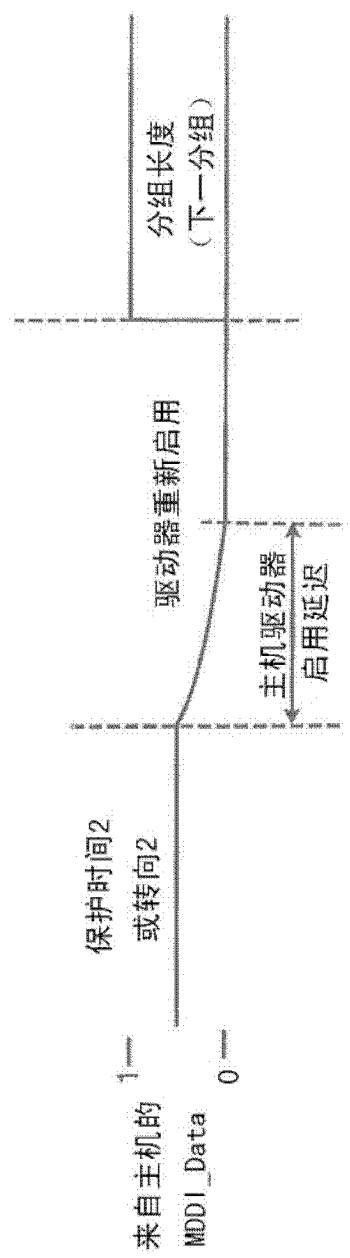


图 46

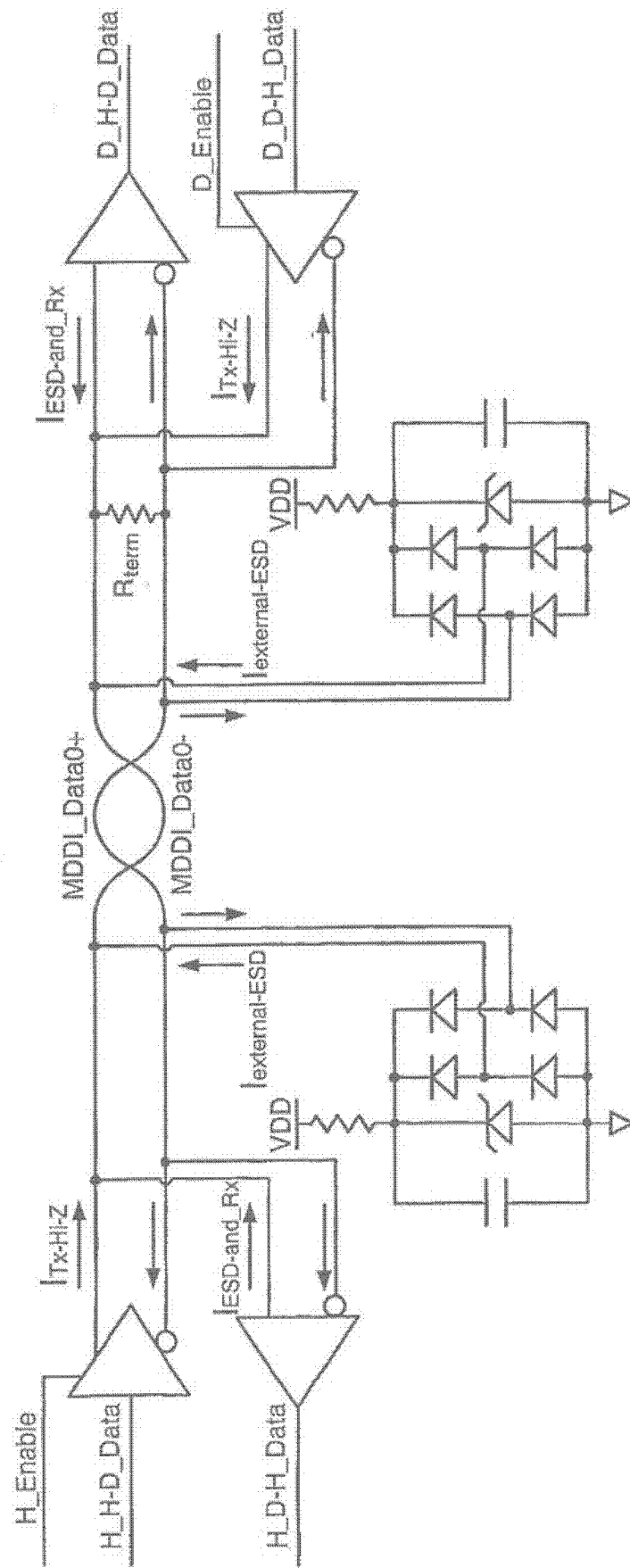
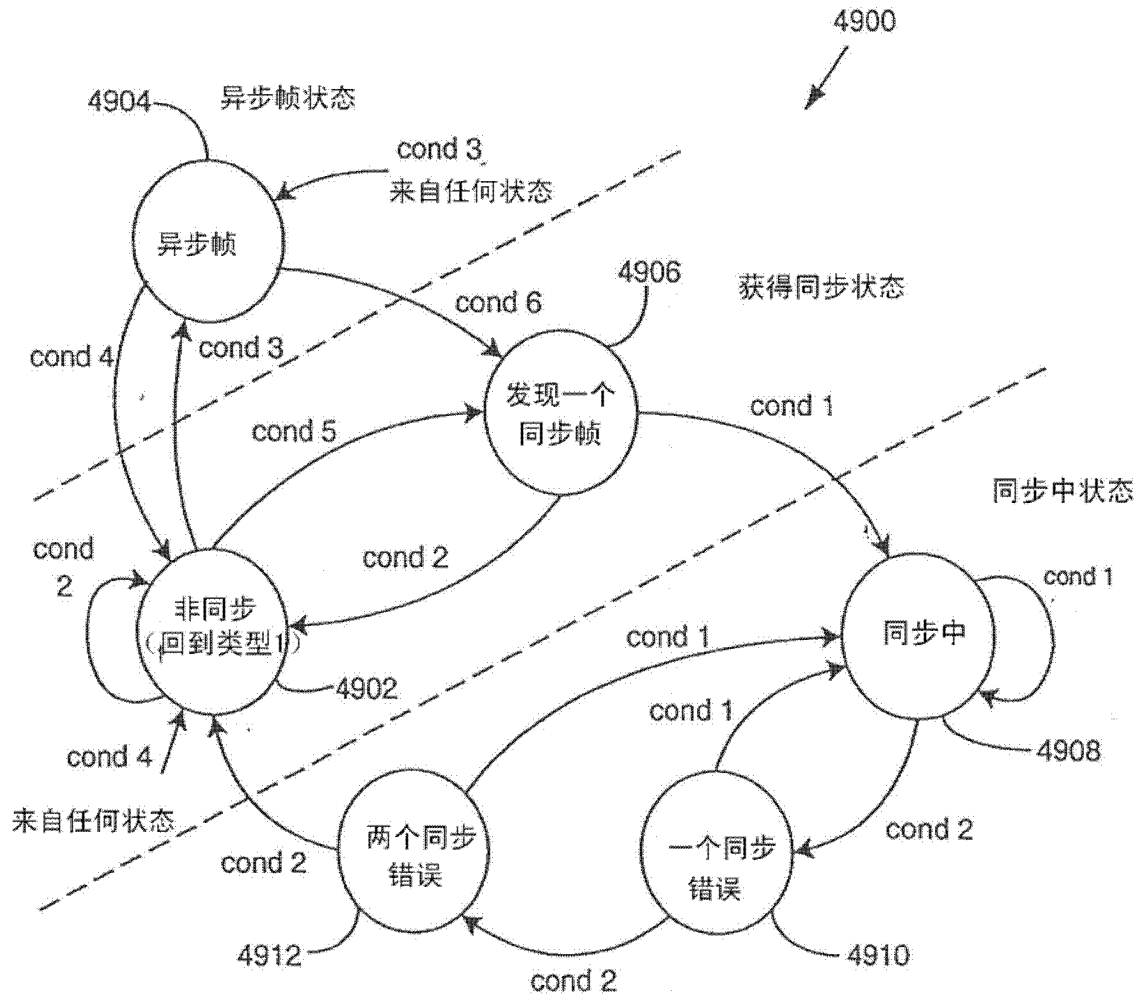


图 47







- cond 1 = 子帧报头分组&处于帧边界的正确CRC, 帧长度>0
- cond 2 = 没有同步模式或帧边界处有错误CRC
- cond 3 = 发现同步模式, 帧长度=0
- cond 4 = 接收到链路关闭分组
- cond 5 = 发现同步模式, 帧长度>0
- cond 6 = 帧报头分组&正确CRC, 帧长度>0

图 49

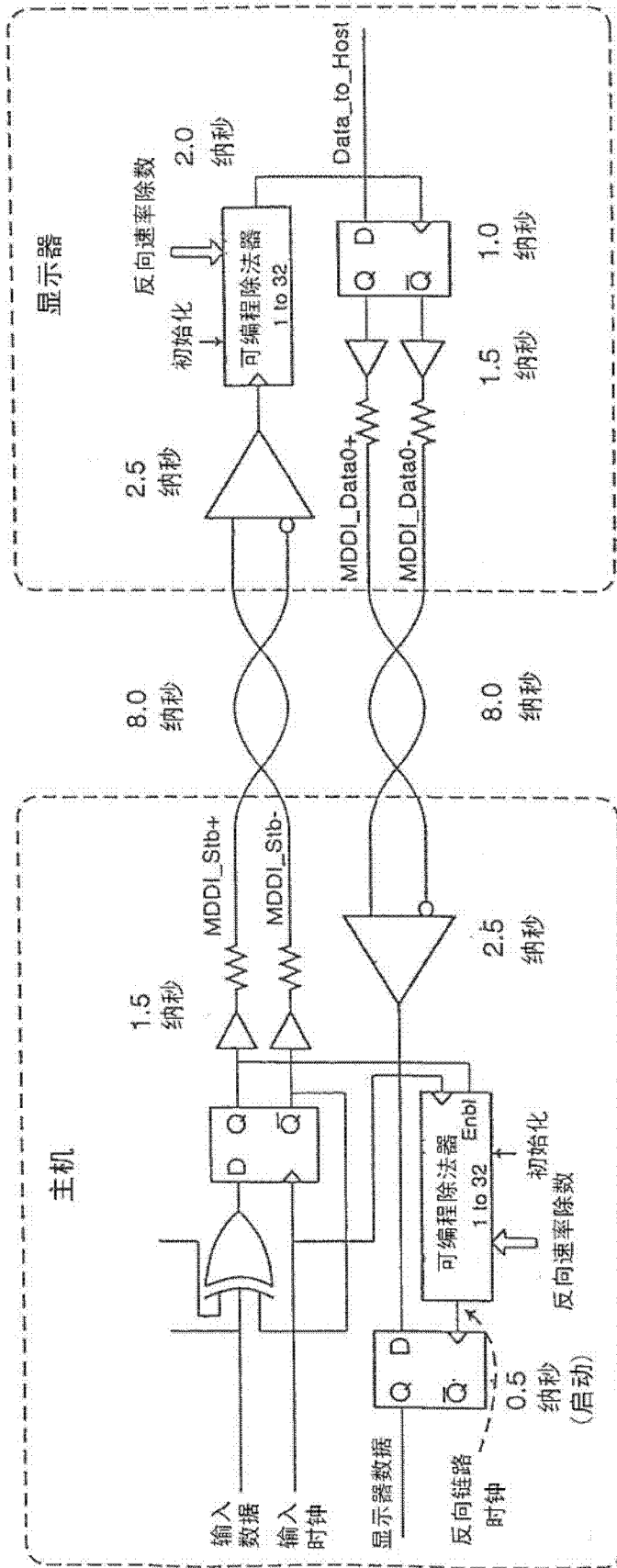


图 50

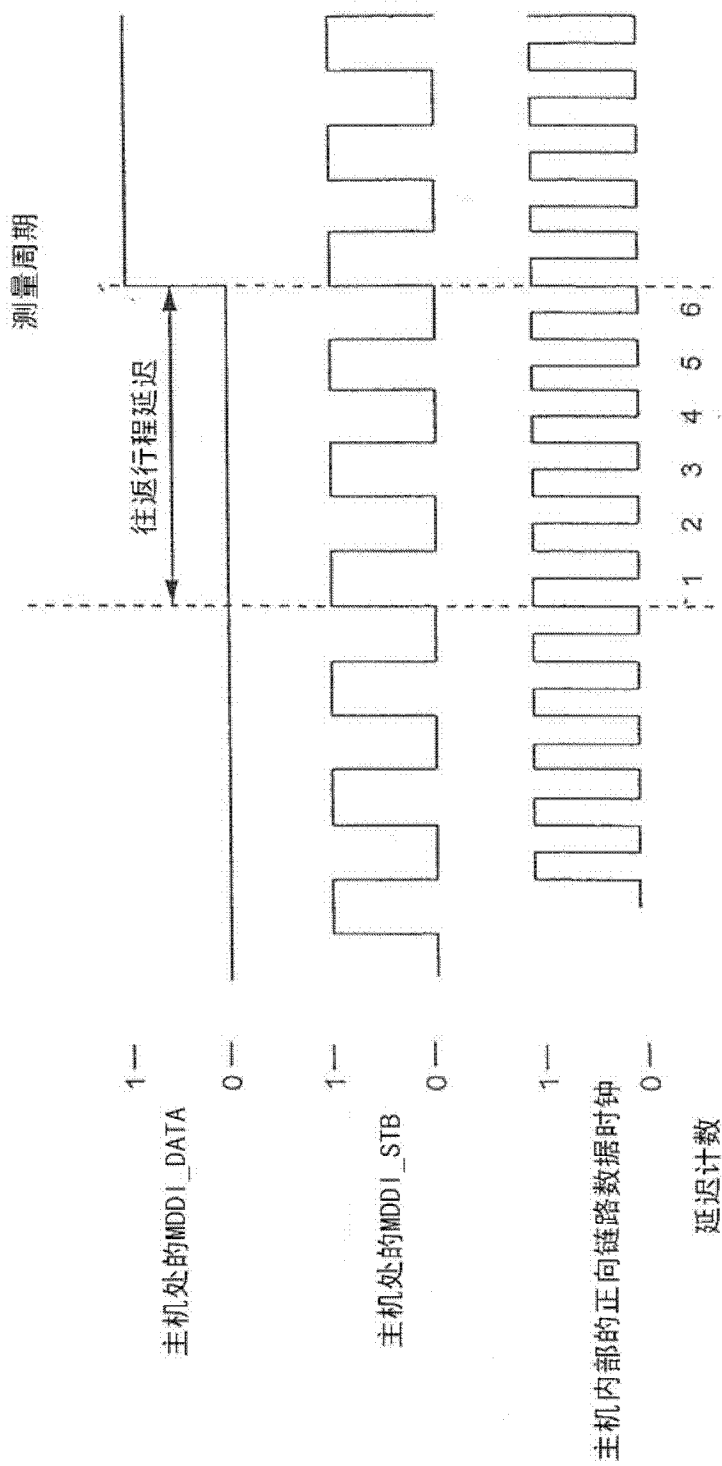


图 51

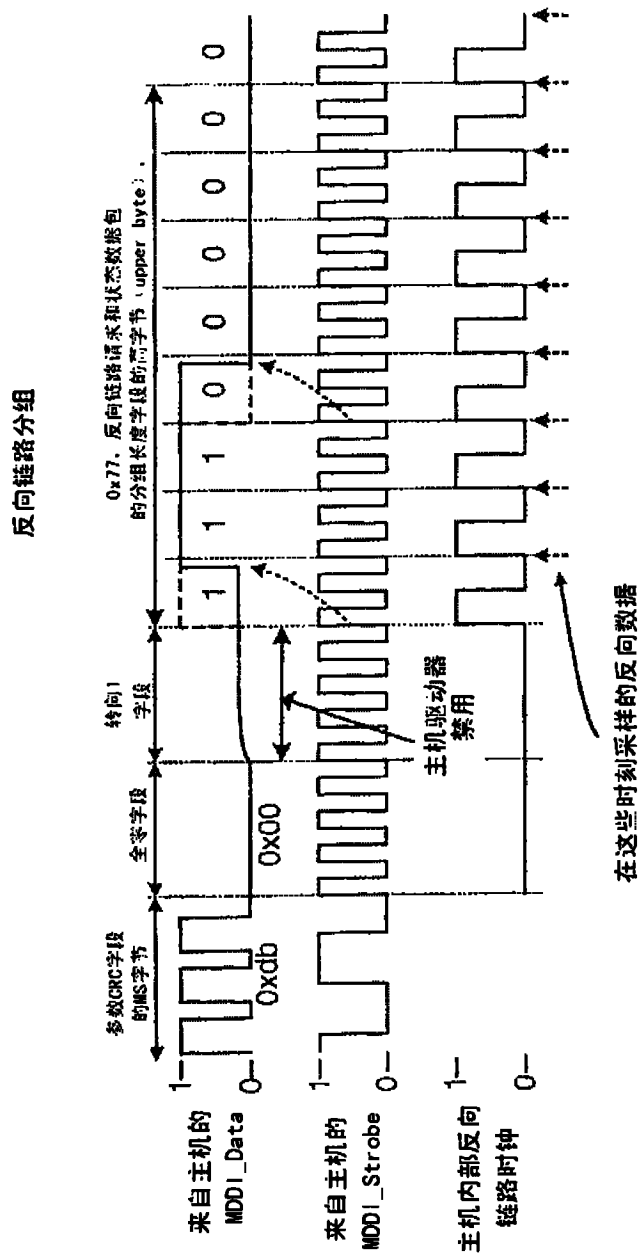


图 52A

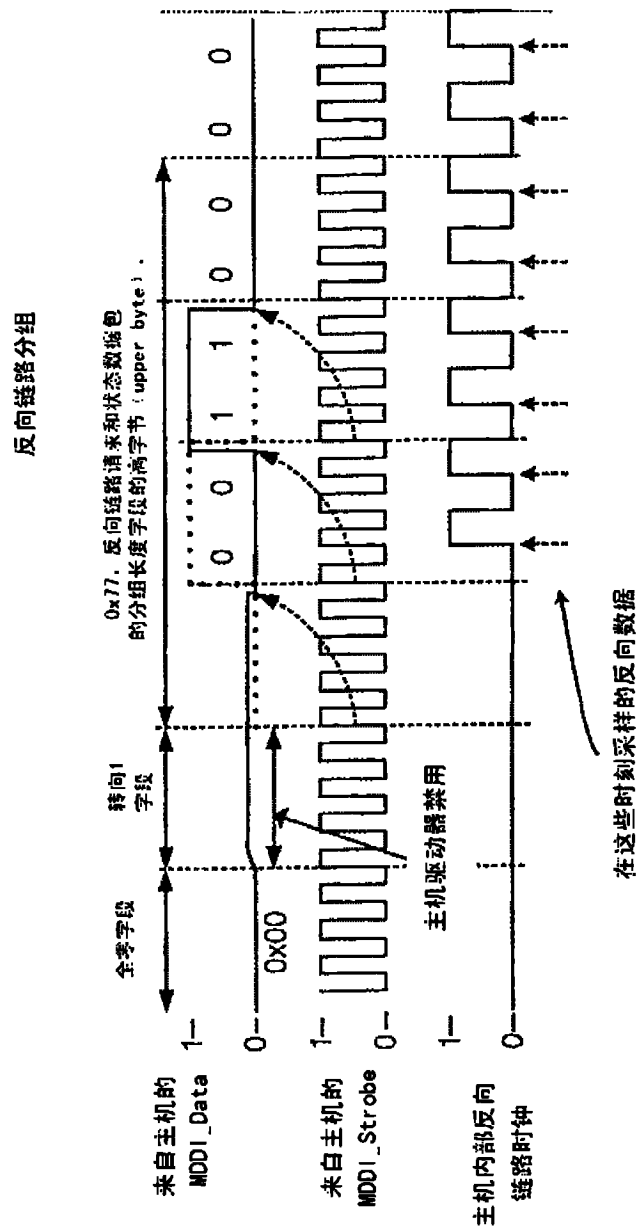


图 52B

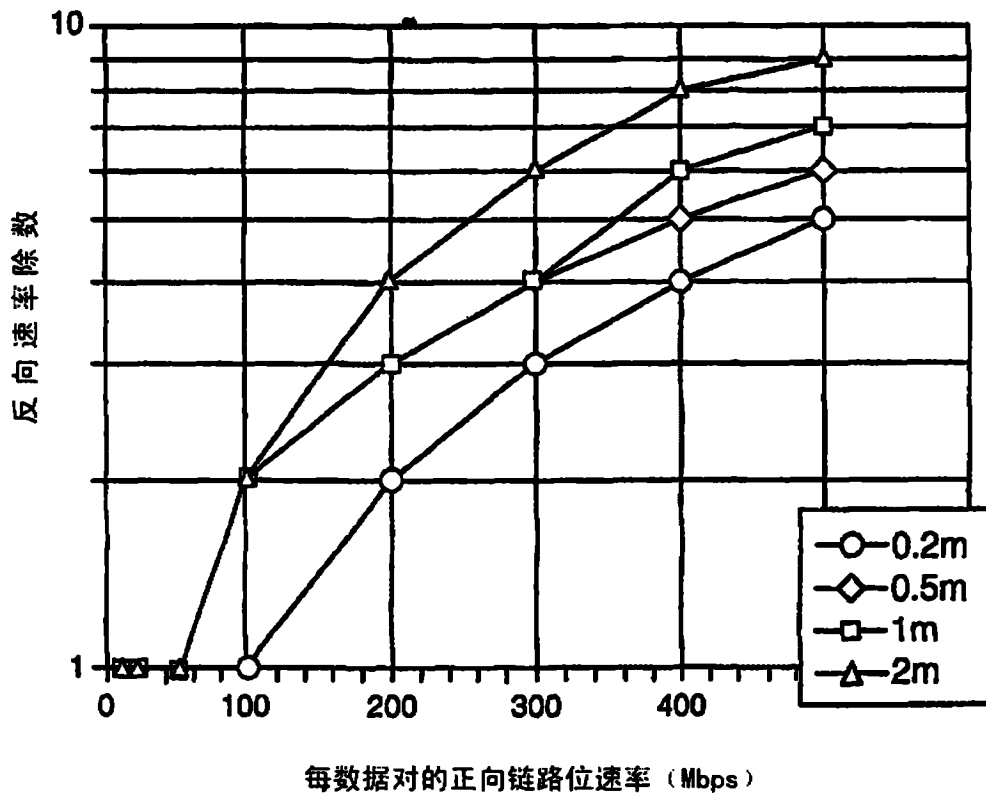


图 53

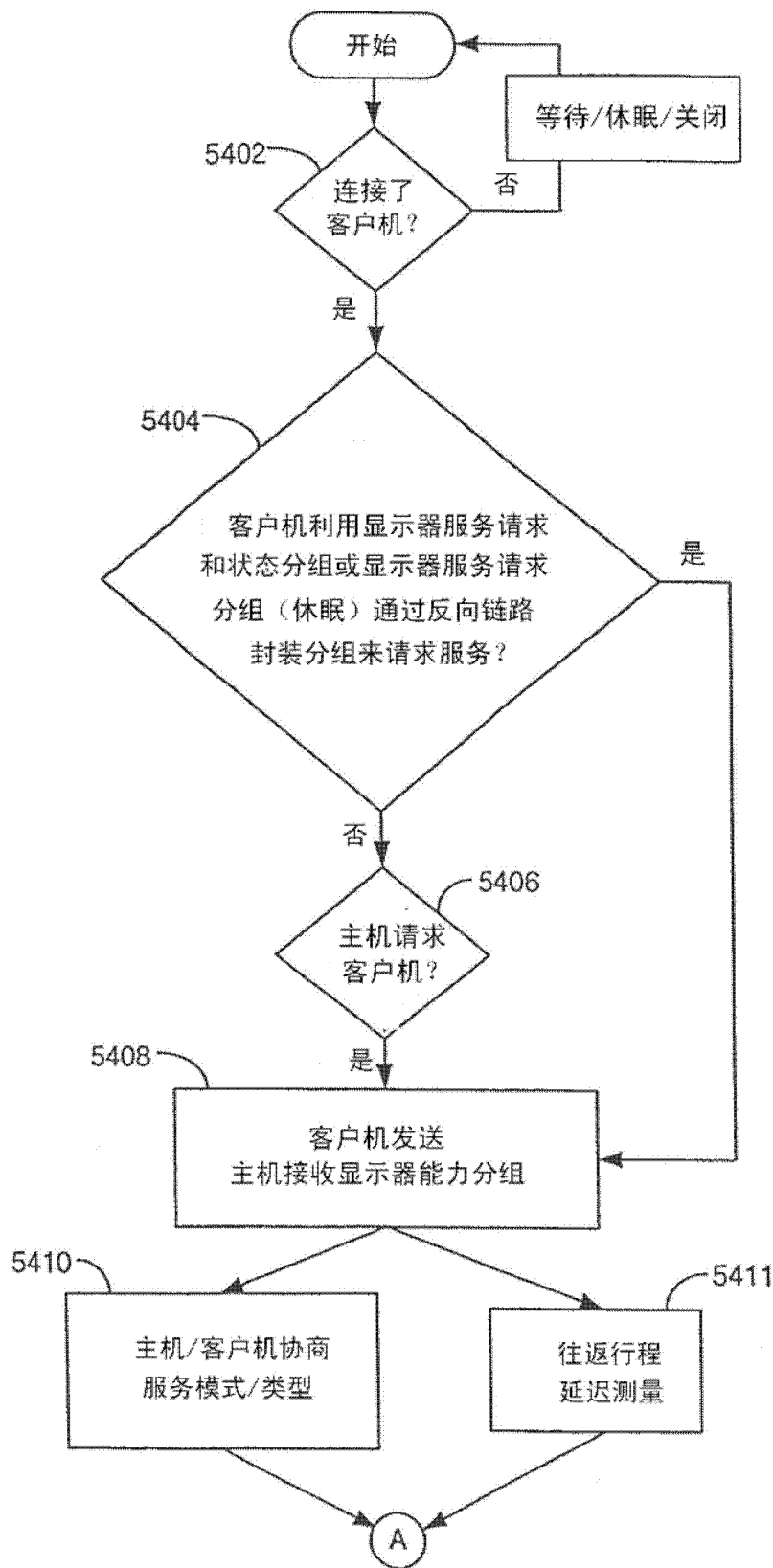


图 54A



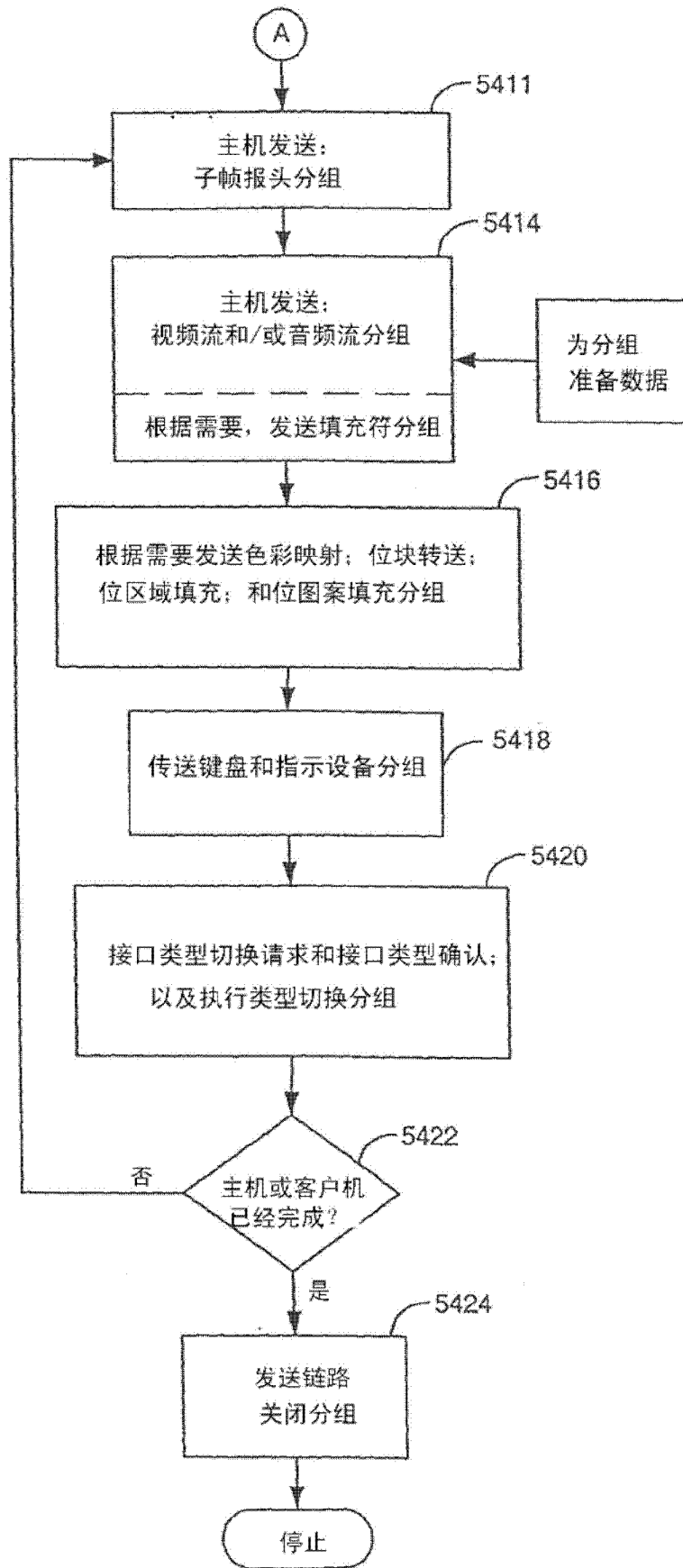


图 54B

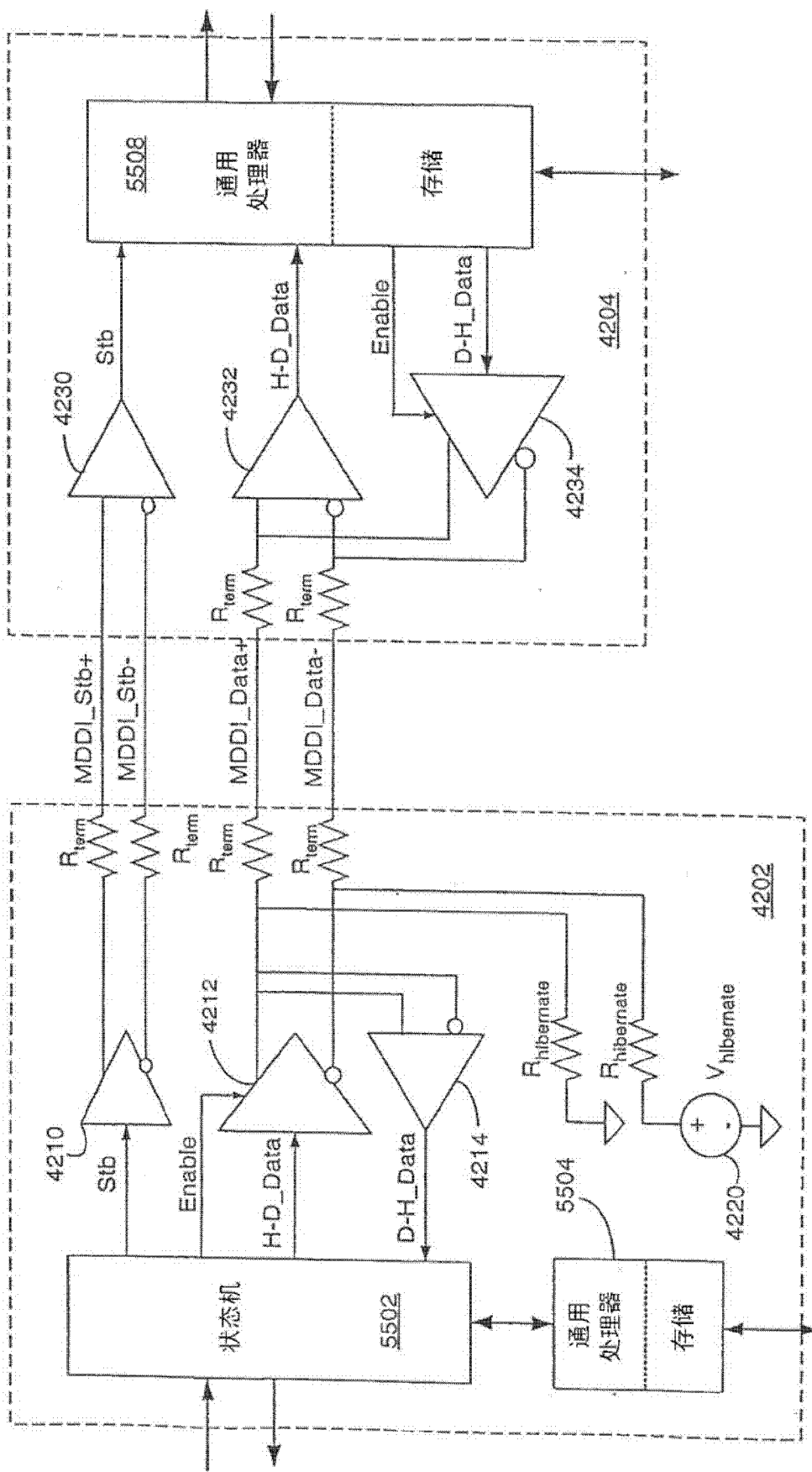


图 55

正向链路偏离校准分组

分组长度	分组类型 = 83	hClient ID	参数 CRC	全零1	校准数据序列	全零2
------	-----------	------------	--------	-----	--------	-----

2字节    2字节    2字节    2字节    8字节    分组长度-22字节    8字节

偏离 --- 100 ps    300 ps    400 ps    600 ps    300 ps    总计 = 2.4 ns

延迟 --- 1.0 ns 典型 1.5 ns 最大    1.0 ns 典型 1.5 ns 最大    2.4 ns 典型 3.3 ns 最大    7.7 ns 典型 10.0 ns 最大    1.0 ns 典型 1.5 ns 最大    0.1 ns 最小 0.2 ns 最大     $t_{SU} = 0.5 \text{ ns}$      $t_{H} = 0.1 \text{ ns}$

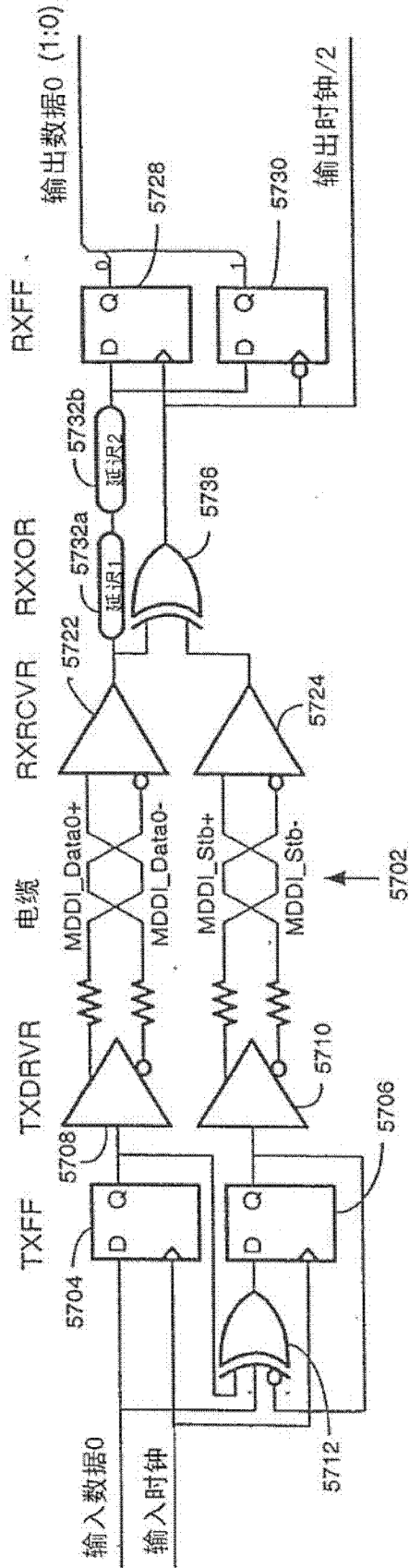


图 56

图 57

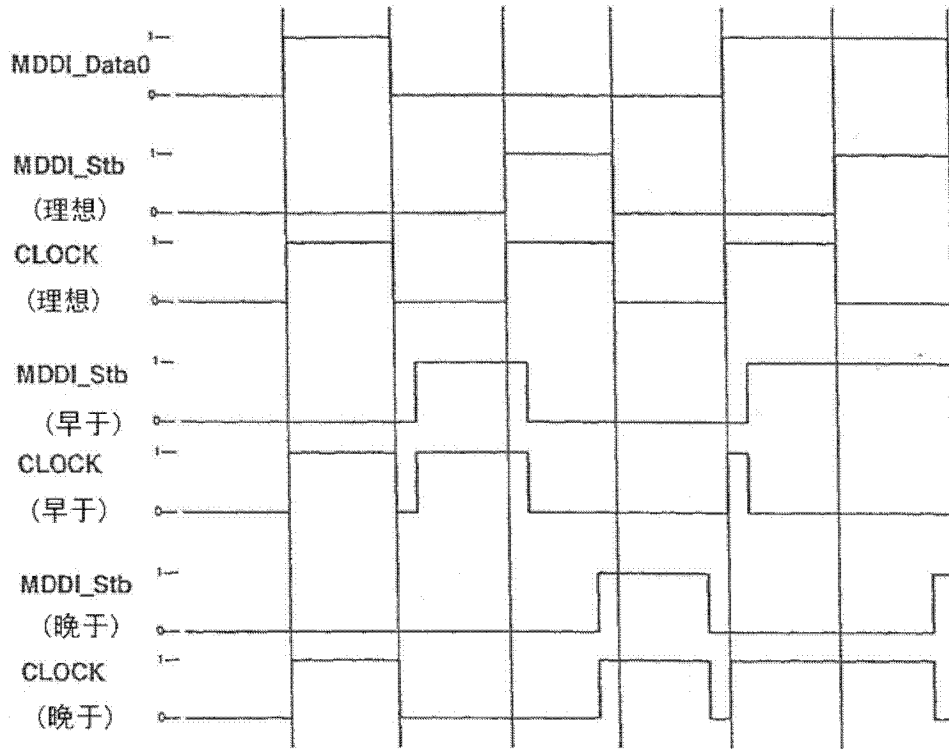


图 58

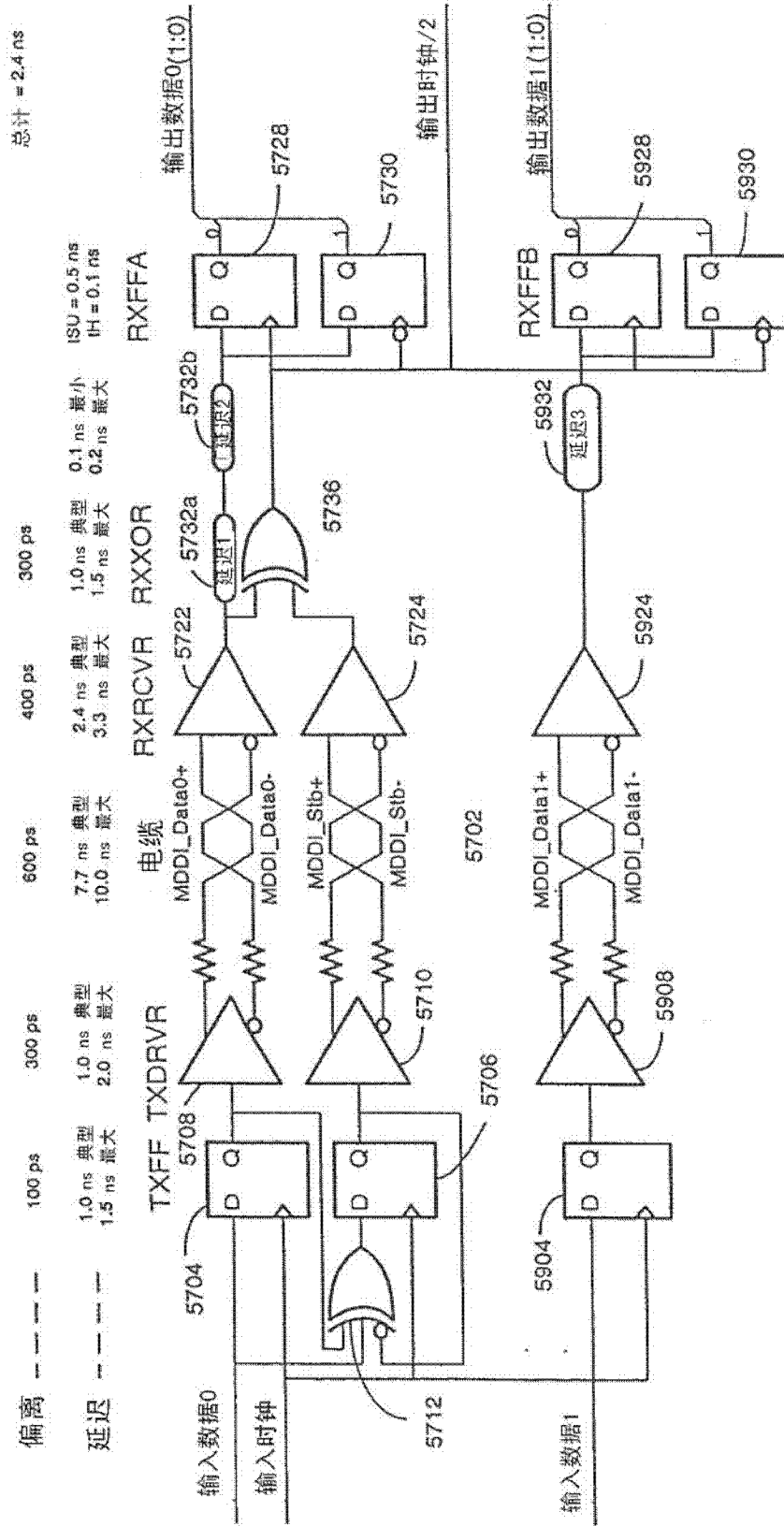


图 59

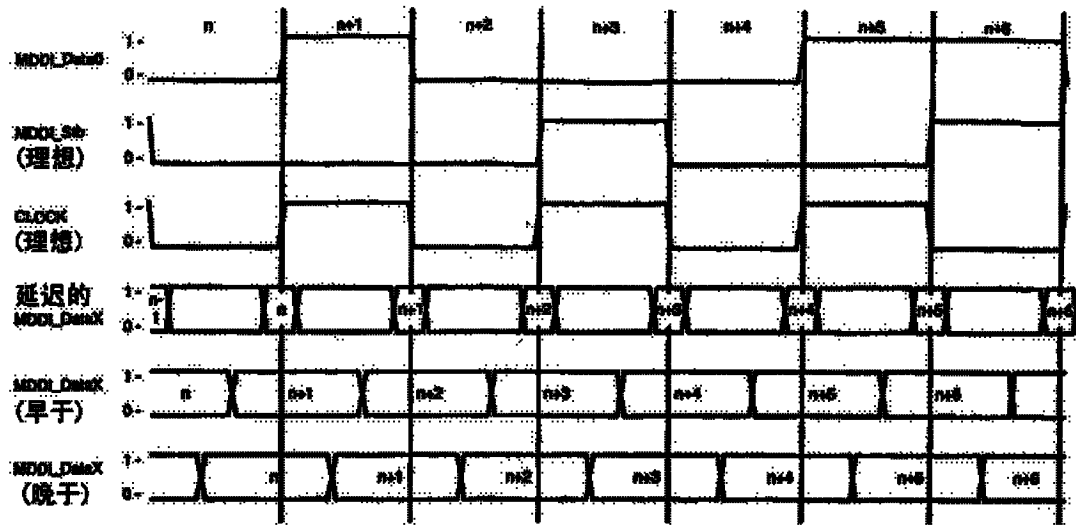


图 60A

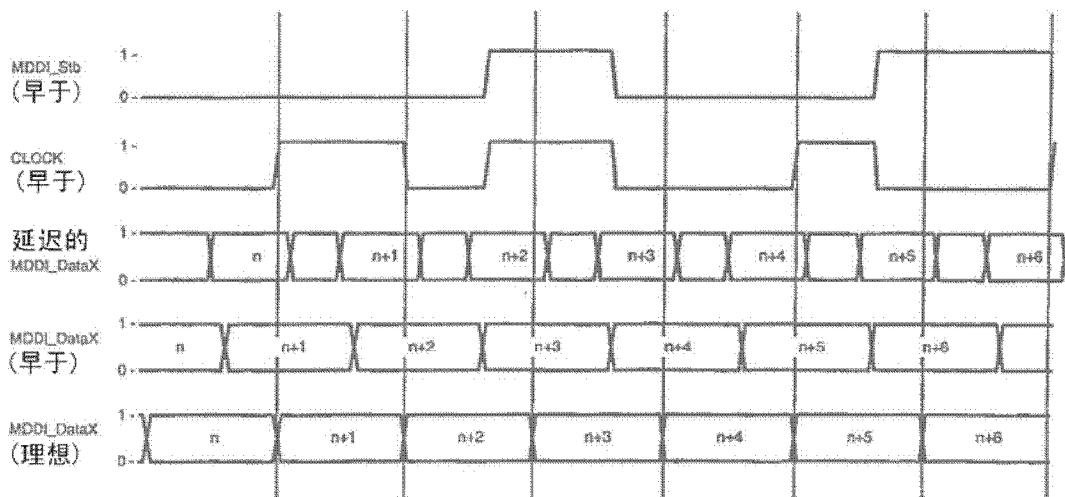


图 60B

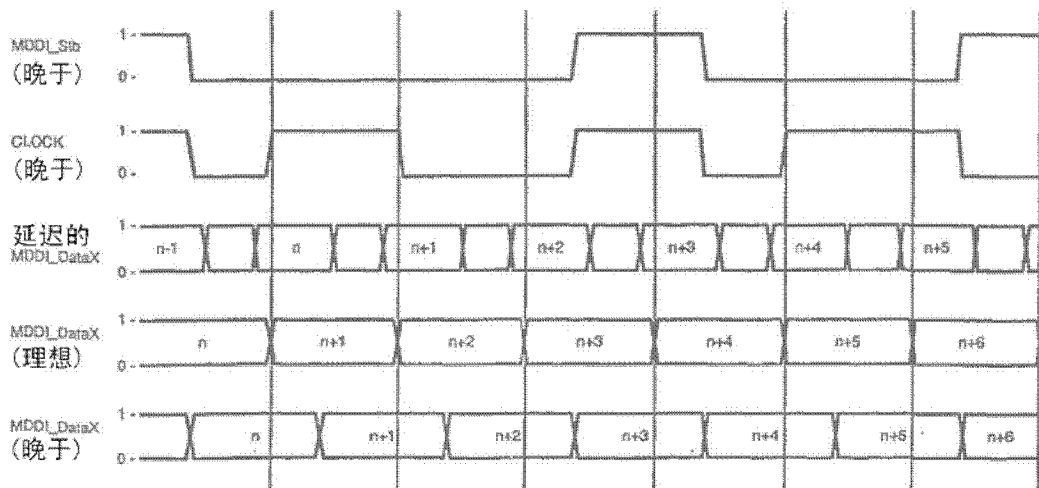


图 60C

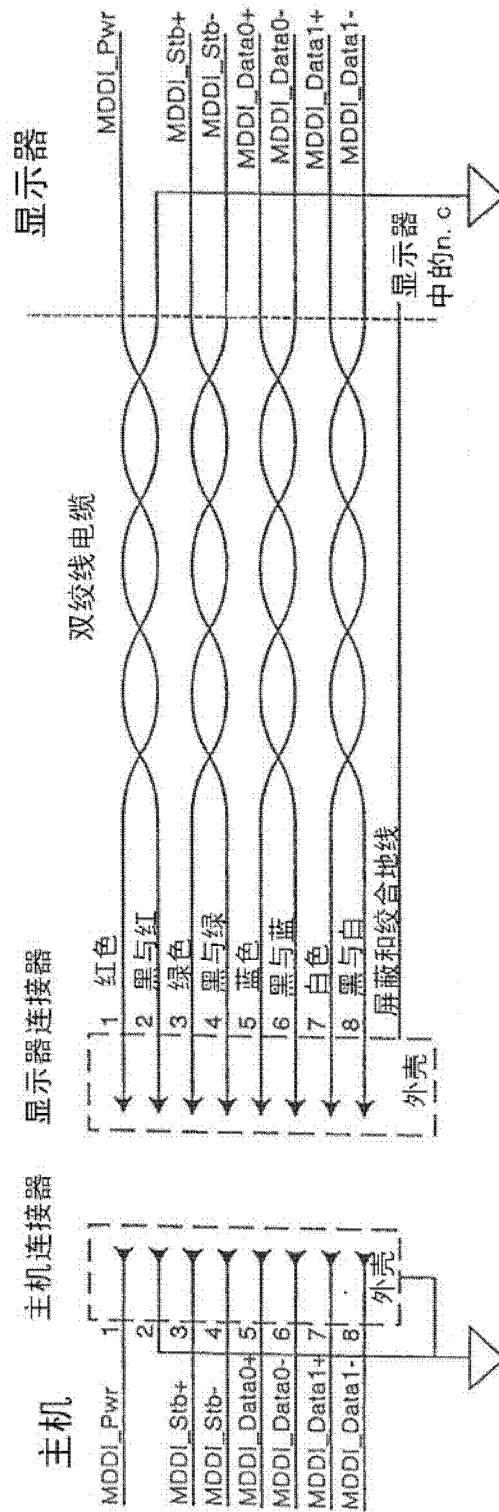


图 61



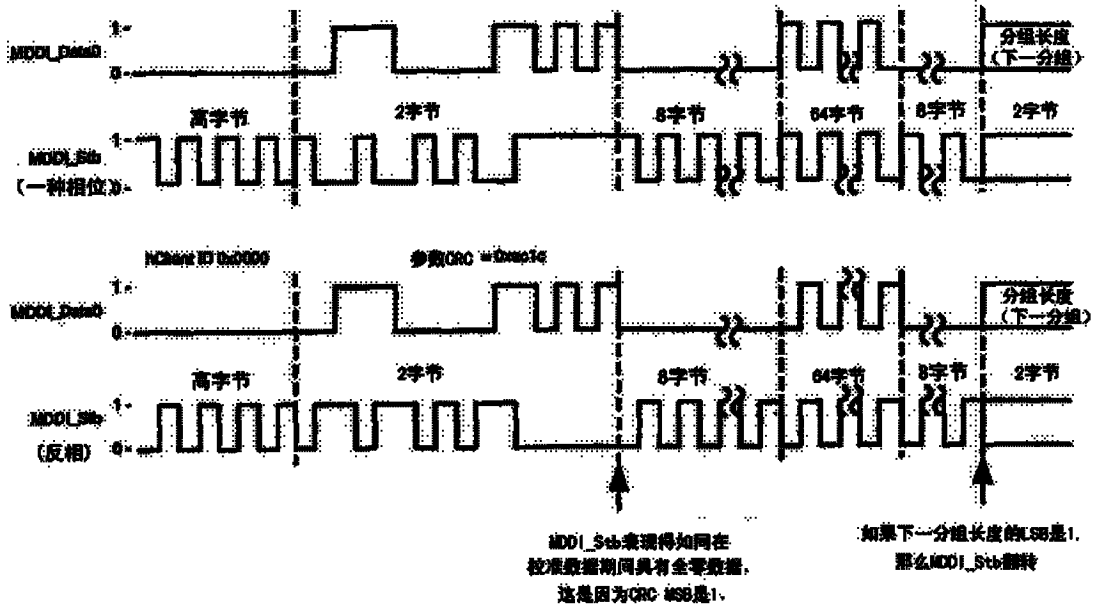


图 62A

类型 II 数据实例

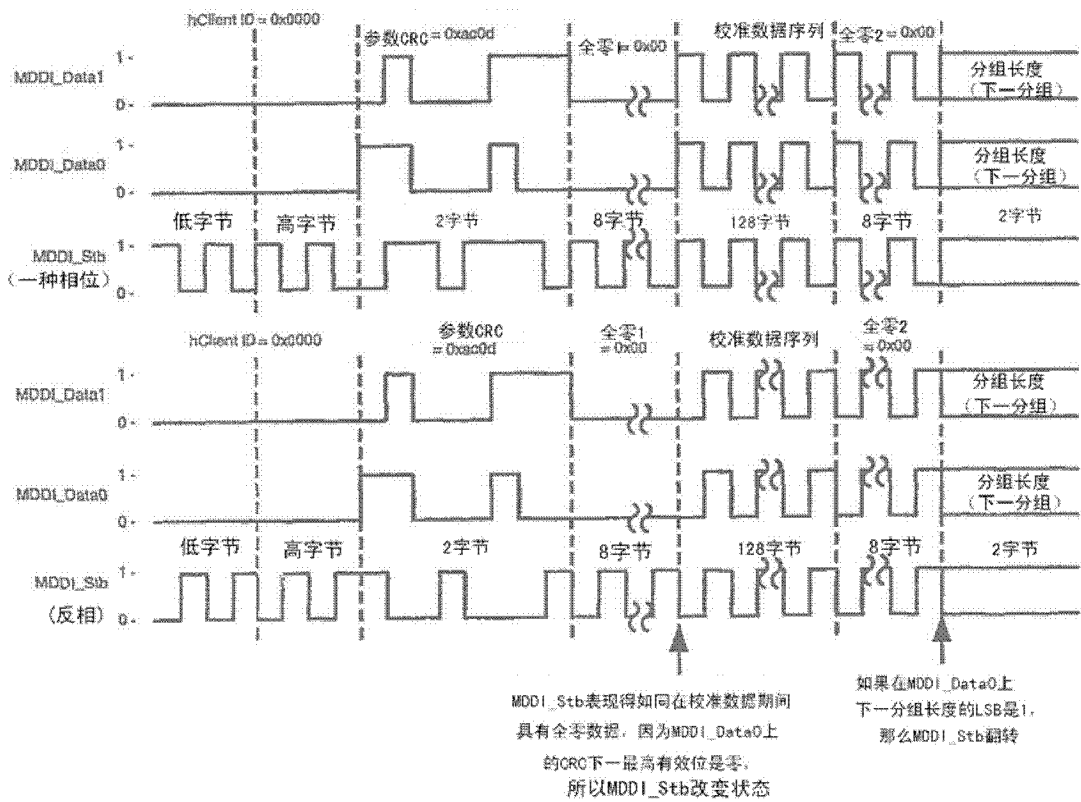
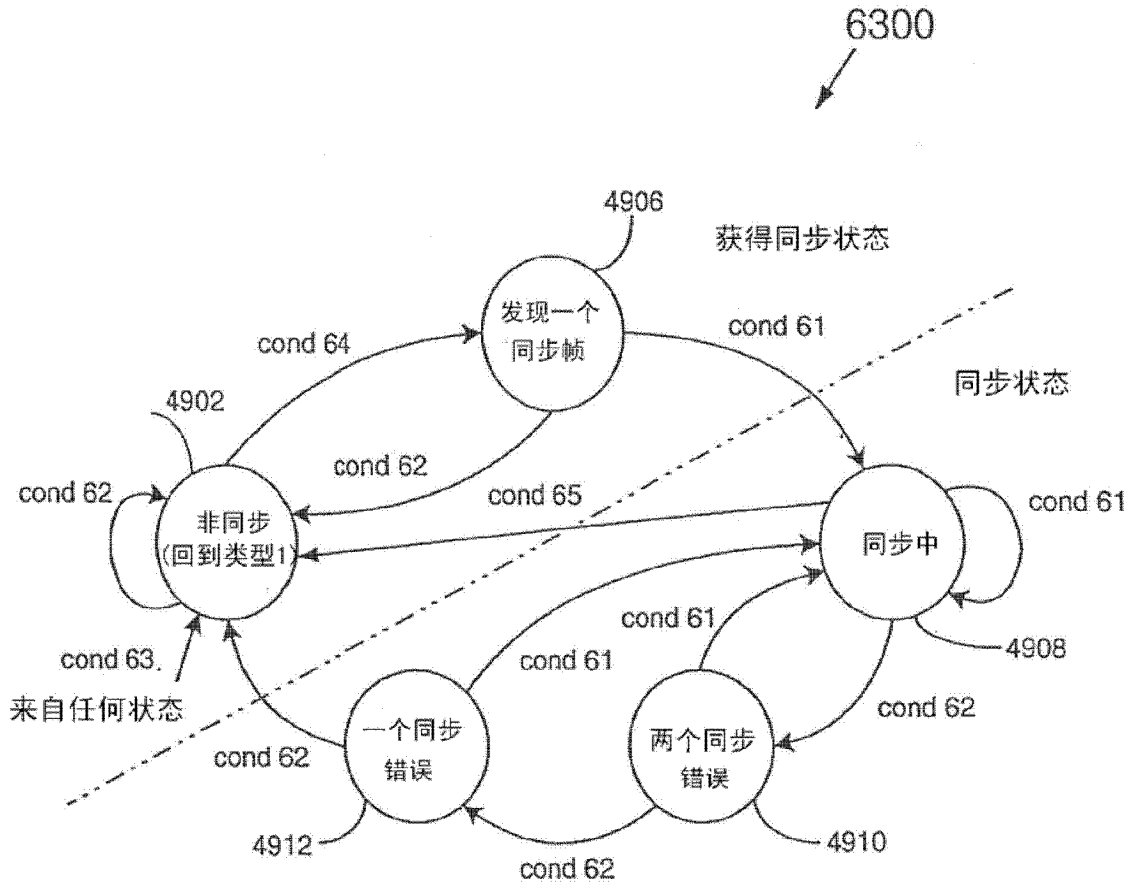


图 62B



- cond 61 = 子帧报头数据包&处于帧边界的正确CRC, 帧长度>0
- cond 62 = 非同步模式或帧边界处的错误CRC
- cond 63 = 接收链路关闭数据包
- cond 64 = 发现同步模式
- cond 65 = 唯一字不正确

图 63

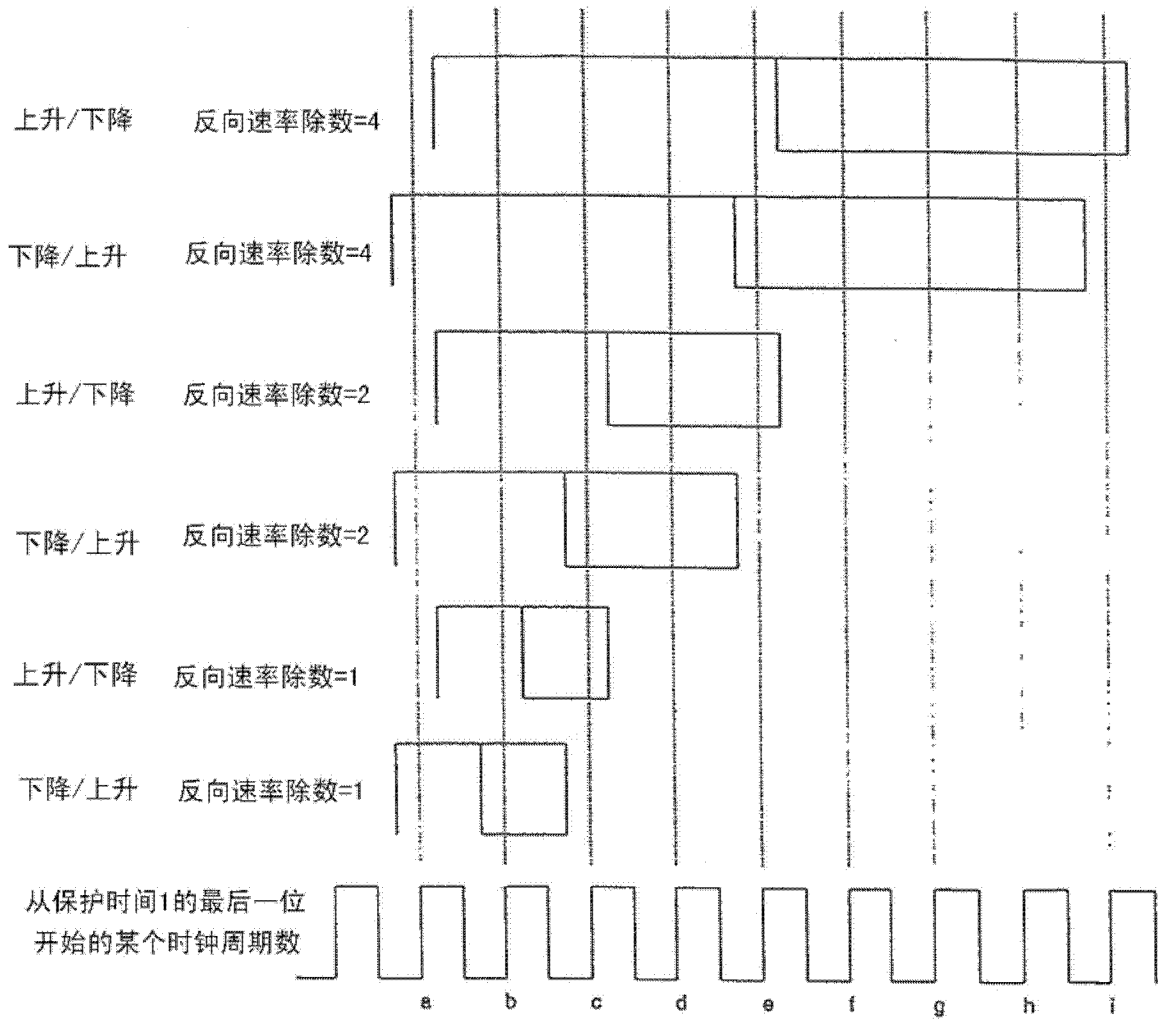


图 64

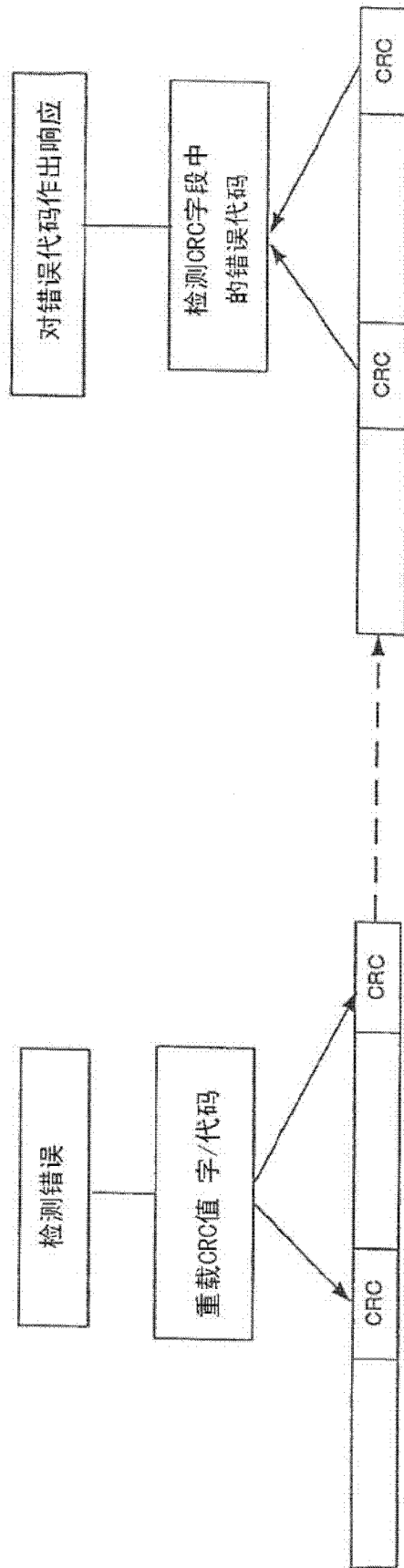


图 65

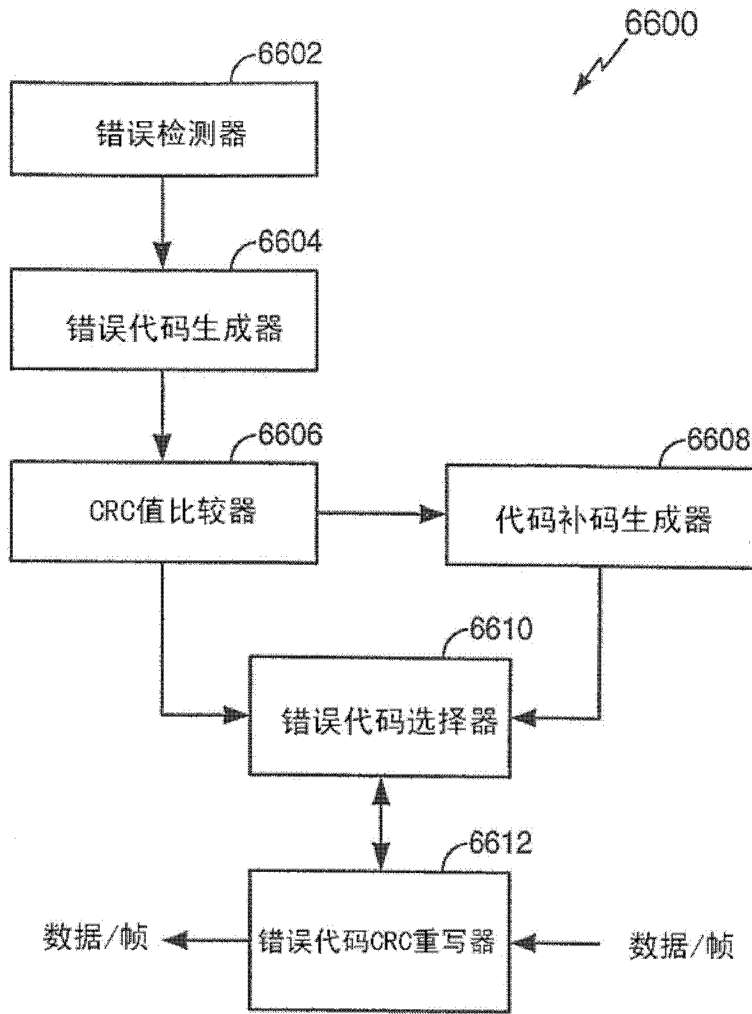


图 66

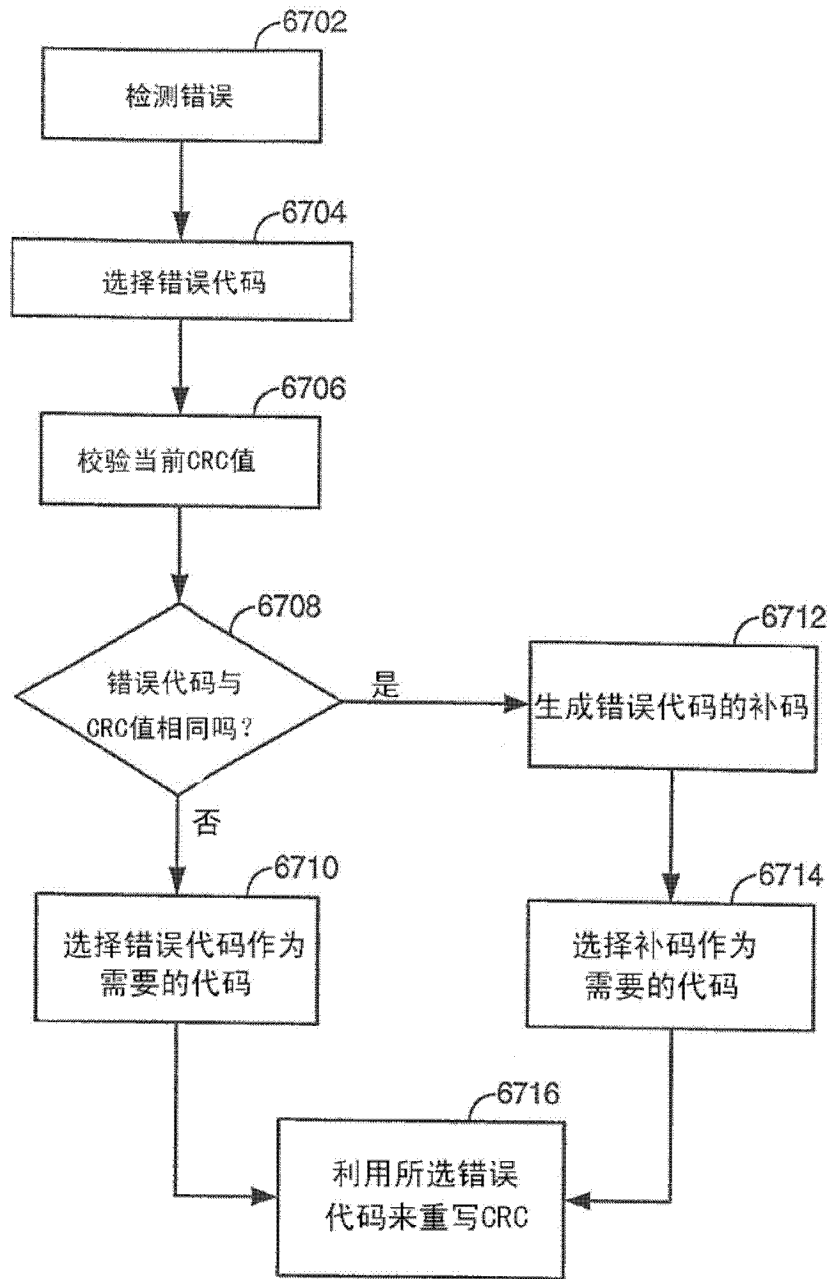


图 67A

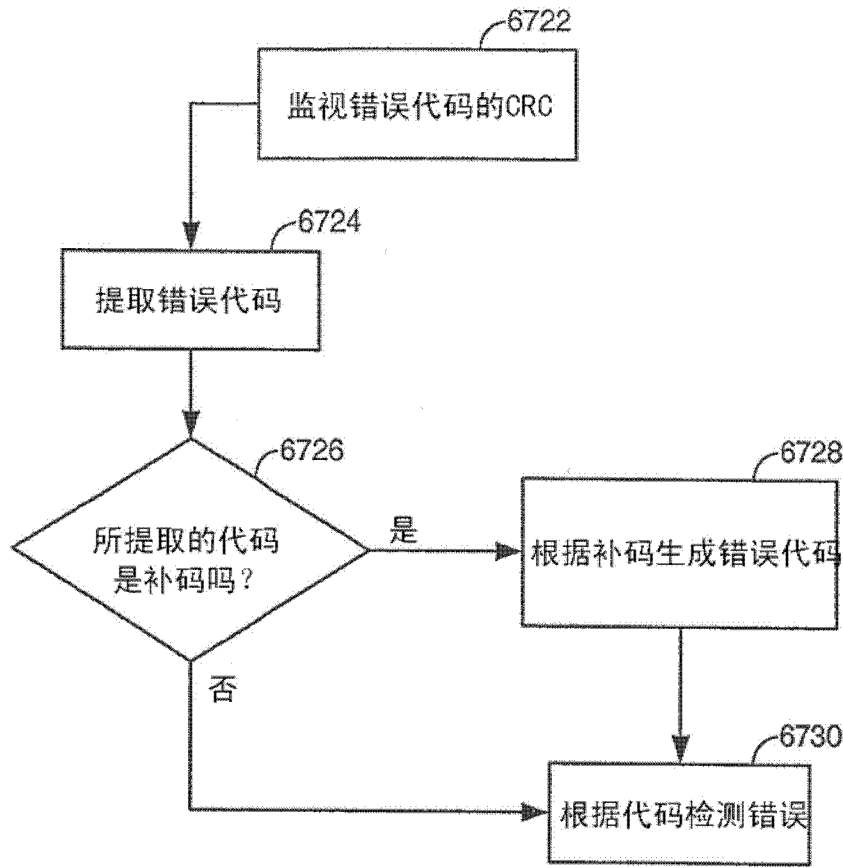


图 67B

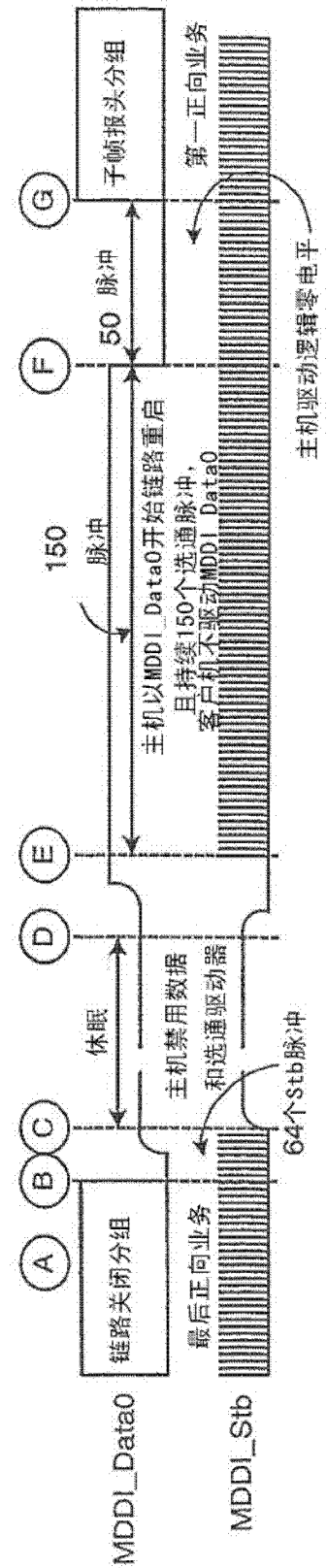


图 68A

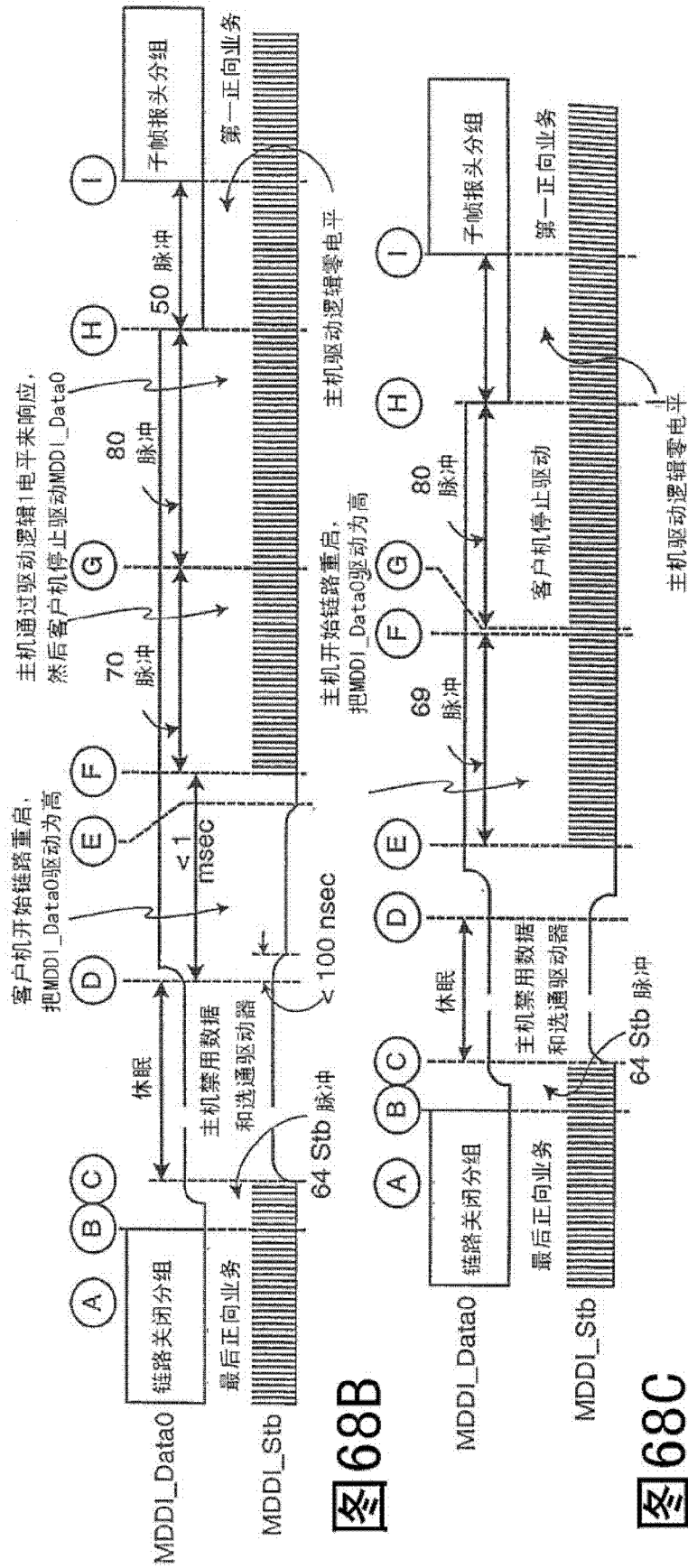


图 68B 图 68C



请求VCP特征分组

分组长度	分组类型 = 128	hClient ID	MCCS VCF 代码	CRC
2字节	2字节	2字节	2字节	2字节

图 69

VCP特征应答分组

分组长度	分组类型 = 129	cClient ID	MCCS版本	应答序列号	列表中特征 数目	VCP特征应答列表	CRC
2字节	2字节	2字节	2字节	2字节	2字节	(分组长度-12) 字节	2字节

图 70

VCP特征应答列表项

MCCS VCP代码	结果代码	最大值	当前值
2字节	2字节	4字节	4字节

图 71

设置VCP特征分组

分组长度	分组类型 = 130	hClient ID	MCCS VCP代码	列表中值的数量	控制值列表	CRC
2字节	2字节	2字节	2字节	2字节	(分组长度-10) 字节	2字节

图 72

请求有效参数分组

分组长度	分组类型 = 131	hClient ID	MCCS VCP代码	CRC
2字节	2字节	2字节	2字节	2字节

图 73

有效参数应答分组

分组长度	分组类型 = 132	cClient ID	MCCS VCP代码	响应代码	应答序列号	列表中值的 数量	VCP参数应答列表	CRC
2字节	2字节	2字节	2字节	2字节	2字节	2字节	(分组长度-14) 字节	2字节

图 74

可缩放视频流能力分组

分组长度	分组类型 = 143	cClient ID	最大流数	源最大X大小	源最大Y大小
2字节	2字节	2字节	2字节	2字节	2字节

RGB能力	单色能力	保留1	Y Cr Cb能力	能力位	保留2	CRC
2字节	1字节	1字节	2字节	1字节	1字节	2字节

图 75

可缩放视频流设置分组

分组长度	分组类型 = 136	hClient ID	流ID	视频数据 格式描述符	像素数据属性
2字节	2字节	2字节	2字节	2字节	2字节

X左沿	Y顶沿	X右沿	Y底沿	X图像大小	Y图像大小	CRC
2字节	2字节	2字节	2字节	2字节	2字节	2字节

图 76

可缩放视频流确认分组

分组长度	分组类型 = 137	cClient ID	流ID	确认代码	CRC
2字节	2字节	2字节	2字节	2字节	2字节

图 77

可缩放视频流分组

分组长度	分组类型 = 18	hClient ID	流ID	参数CRC	像素计数	像素数据	像素数据CRC
2字节	2字节	2字节	2字节	2字节	2字节	分组长度-12字节	2字节

图 78

请求特定状态分组

分组长度	分组类型 = 138	hClient ID	状态数据包ID	CRC
2字节	2字节	2字节	2字节	2字节

图 79

## 有效状态应答列表分组

分组长度	分组类型 = 139	cClient ID	列表中值的数量	有效参数应答列表	CRC
2字节	2字节	2字节	2字节	分组长度-8字节	2字节

图 80

## 分组处理延迟参数分组

分组长度	分组类型 = 140	cClient ID	列表项数目	延迟参数列表	CRC
2字节	2字节	2字节	2字节	分组长度-8字节	2字节

图 81A

## 延迟参数列表项

延迟分组类型	像素延迟	水平像素延迟	垂直像素延迟	固定延迟
2字节	1字节	1字节	1字节	1字节

图 81B

## 个人显示器能力分组

分组长度	分组类型 = 141	cClient ID	子像素布局	像素形状	水平视场	垂直视场	视轴交叉
2字节	2字节	2字节	1字节	1字节	1字节	1字节	1字节
左右图像重叠	透明度	最大亮度	光学能力	最小IPD	最大IPD	视场曲率点列表 (25个2字节值)	CRC
1字节	1字节	1字节	2字节	1字节	1字节	50字节	2字节

图 82

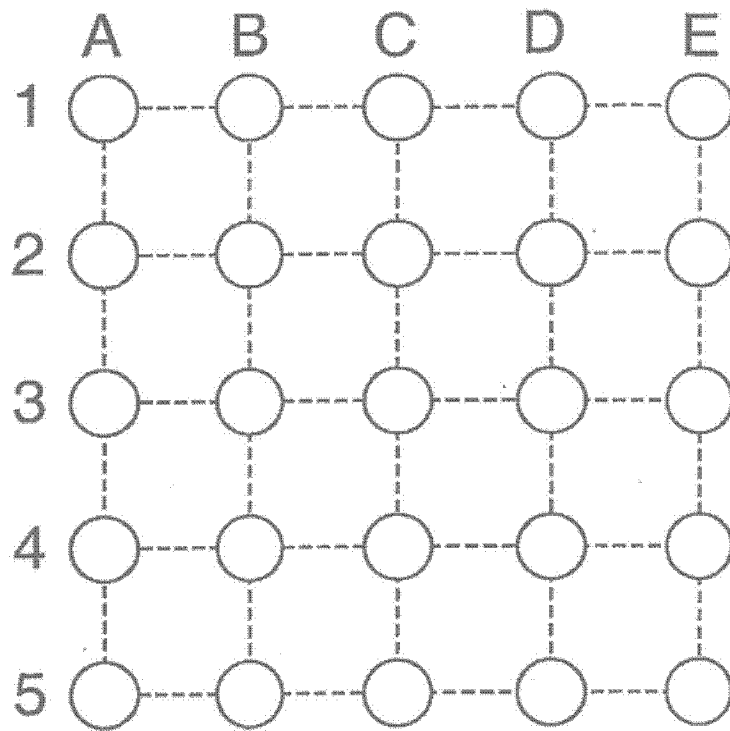


图 83

客户机错误报告分组

分组长度	分组类型 = 142	cClient ID	列表项数	错误代码列表	CRC
2字节	2字节	2字节	2字节	分组长度-8字节	2字节

图 84A

错误报告列表项

显示器 错误代码	错误子代码
2字节	2字节

图 84B

客户机标识分组

分组长度	分组类型 = 144	cClient ID	制造星期	制造年份	制造商 名称长度	产品名称长度	序号长度
2字节	2字节	2字节	1字节	1字节	2字节	2字节	2字节

制造商名称字符串	产品名称字符串	序号字符串	CRC
制造商名称字节长度	制造商名称字节长度	制造商名称字节长度	2字节

图 85

可选显示器能力分组

分组长度	分组类型 = 145	cClient ID	可选显示器号	保留1	位图宽度	位图高度	显示窗宽度	显示窗高度
2字节	2字节	2字节	1字节	1字节	2字节	2字节	2字节	2字节

彩色映射 RGB宽度	RGB能力	单色能力	保留2	YCbCr能力	显示器 特征能力	保留3	CRC
2字节	2字节	1字节	1字节	2字节	1字节	1字节	2字节

图 86

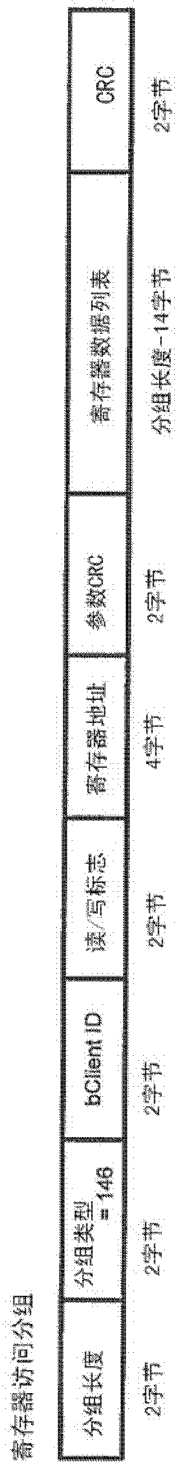


图 87

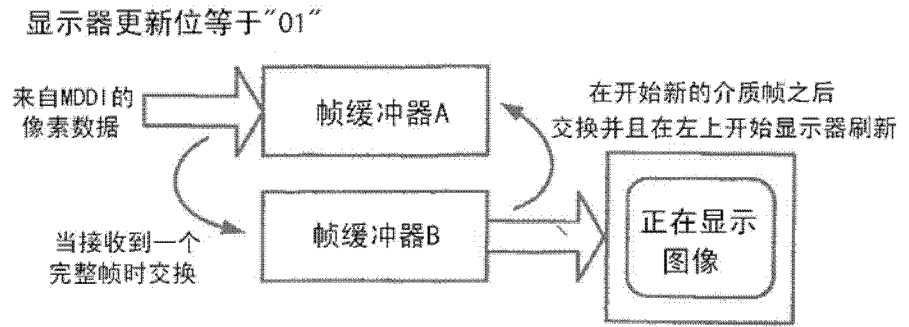


图 88A

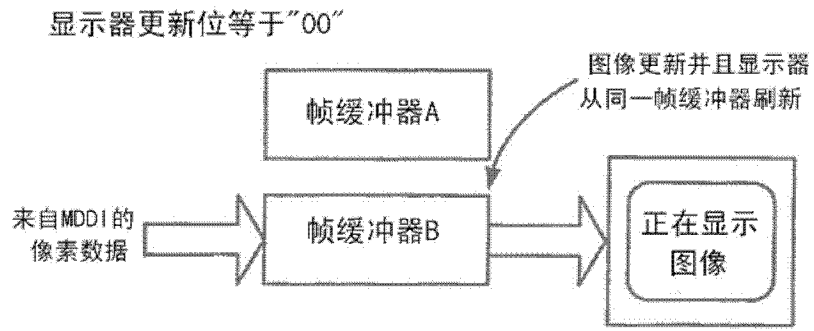


图 88B

显示器更新位等于“11”

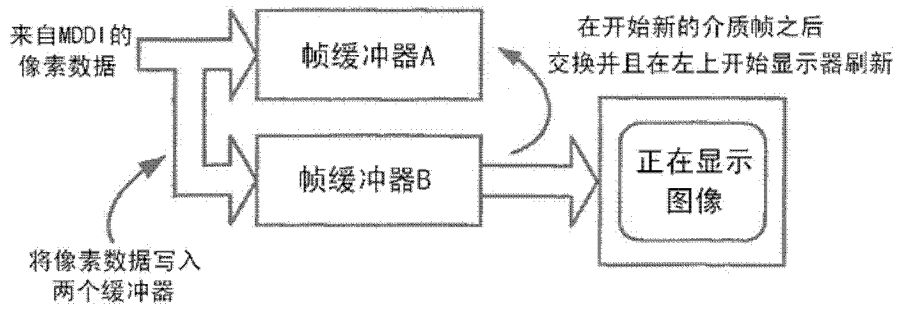


图 88C

两个缓冲器，显示器刷新快于图像传送

图像传送

帧1, 缓冲器A 20% 40% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器A 20% 40% 60% 80%	帧4, 缓冲器B 20% 40% 60% 80%	帧5, 缓冲器A 20% 40% 60% 80%	帧6, 缓冲器B 20% 40% 60% 80%
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

显示器刷新

帧0, 缓冲器B 20% 40% 60% 80%	帧1, 缓冲器A 20% 40% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器A 20% 40% 60% 80%	帧4, 缓冲器B 20% 40% 60% 80%	帧5, 缓冲器A 20% 40% 60% 80%
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

两个缓冲器，显示器刷新比图像传送快得多

图像传送

帧1, 缓冲器A 20% 40% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器A 20% 40% 60% 80%	帧4, 缓冲器B 20% 40% 60% 80%	帧5, 缓冲器A 20% 40% 60% 80%	帧6, 缓冲器B 20% 40% 60% 80%
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

显示器刷新

帧0, b=B 20 40 60 80	帧1, b=B 20 40 60 80	帧2, b=B 20 40 60 80	帧3, b=A 20 40 60 80	帧4, b=B 20 40 60 80	帧5, b=A 20 40 60 80	帧6, b=A 20 40 60 80
------------------------	------------------------	------------------------	------------------------	------------------------	------------------------	------------------------

两个缓冲器，显示器刷新慢于图像传送

图像传送

帧1, 缓冲器A 20% 40% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器A 20% 40% 60% 80%	帧4, 缓冲器B 20% 40% 60% 80%	帧5, 缓冲器A 20% 40% 60% 80%	帧6, 缓冲器B 20% 40% 60% 80%
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

显示器刷新

帧0, 缓冲器B 20% 40% 60% 80%	帧1&3, 缓冲器A 40% 20% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器A 40% 20% 60% 80%	帧4&6, 缓冲器B 40% 20% 60% 80%
-----------------------------	-------------------------------	-----------------------------	-----------------------------	-------------------------------

图 89

图 90



三个缓冲器，显示器刷新比图像传送快得多，任何大小的视频窗口

图像传送

帧1, 缓冲器A 20% 40% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器C 20% 40% 60% 80%	帧4, 缓冲器A 20% 40% 60% 80%	帧5, 缓冲器B 20% 40% 60% 80%	帧6, 缓冲器C 20% 40% 60% 80%
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

显示器刷新

fm0, b=B 20 40 60 80	fm0, b=A 20 40 60 80	fm1, b=B 20 40 60 80	fm1, b=A 20 40 60 80	fm2, b=B 20 40 60 80	fm2, b=A 20 40 60 80	fm3, b=B 20 40 60 80	fm3, b=A 20 40 60 80	fm4, b=B 20 40 60 80	fm4, b=A 20 40 60 80	fm5, b=B 20 40 60 80	fm5, b=A 20 40 60 80
-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------

图 91

三个缓冲器，显示器刷新比图像传送快得多，任何大小的视频窗口

图像传送

帧1, 缓冲器A 20% 40% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器C 20% 40% 60% 80%	帧4, 缓冲器A 20% 40% 60% 80%	帧5, 缓冲器B 20% 40% 60% 80%	帧6, 缓冲器C 20% 40% 60% 80%
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

显示器刷新

fm0, b=C 20 40 60 80	fm0, b=B 20 40 60 80	fm0, b=A 20 40 60 80	fm1, b=C 20 40 60 80	fm1, b=B 20 40 60 80	fm1, b=A 20 40 60 80	fm2, b=C 20 40 60 80	fm2, b=B 20 40 60 80	fm2, b=A 20 40 60 80	fm3, b=C 20 40 60 80	fm3, b=B 20 40 60 80	fm3, b=A 20 40 60 80	fm4, b=C 20 40 60 80	fm4, b=B 20 40 60 80	fm4, b=A 20 40 60 80	fm5, b=C 20 40 60 80	fm5, b=B 20 40 60 80	fm5, b=A 20 40 60 80
-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------

图 92

三个缓冲器，显示器刷新慢于图像传送

图像传送

帧1, 缓冲器A 20% 40% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器C 20% 40% 60% 80%	帧4, 缓冲器A 20% 40% 60% 80%	帧5, 缓冲器B 20% 40% 60% 80%	帧6, 缓冲器C 20% 40% 60% 80%	帧7 20%
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------

显示器刷新

帧0, 缓冲器C 20% 40% 60% 80%	帧1&4, 缓冲器A 20% 40% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器C 20% 40% 60% 80%	帧4&7, 缓冲器A 20% 40% 60% 80%
-----------------------------	-------------------------------	-----------------------------	-----------------------------	-------------------------------

图 93

一个缓冲器，显示器刷新快于图像传送

图像传送

帧1 20% 40% 60% 80%	帧2 20% 40% 60% 80%	帧3 20% 40% 60% 80%	帧4 20% 40% 60% 80%	帧5 20% 40% 60% 80%	帧6 20% 40% 60% 80%
-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------

显示器刷新

帧0 20% 40% 60% 80%	帧1 20% 40% 60% 80%	帧2 20% 40% 60% 80%	帧3&2 20% 40% 60% 80%	帧4 20% 40% 60% 80%	帧5&4 20% 40% 60% 80%	帧6 20% 40% 60% 80%
-----------------------	-----------------------	-----------------------	-------------------------	-----------------------	-------------------------	-----------------------

一个缓冲器，显示器刷新比图像传送快得多

图像传送

帧1 20% 40% 60% 80%	帧2 20% 40% 60% 80%	帧3 20% 40% 60% 80%	帧4 20% 40% 60% 80%	帧5 20% 40% 60% 80%	帧6 20% 40% 60% 80%
-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------

显示器刷新

帧0 20% 40% 60% 80%	帧1 20% 40% 60% 80%	帧2 20% 40% 60% 80%	帧2&1 20% 40% 60% 80%	帧3 20% 40% 60% 80%	帧3 20% 40% 60% 80%	帧3 20% 40% 60% 80%	帧4 20% 40% 60% 80%	帧4 20% 40% 60% 80%	帧4 20% 40% 60% 80%	帧5 20% 40% 60% 80%	帧5 20% 40% 60% 80%	帧5 20% 40% 60% 80%	帧6 20% 40% 60% 80%	帧6 20% 40% 60% 80%
-----------------------	-----------------------	-----------------------	-------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------

图 94

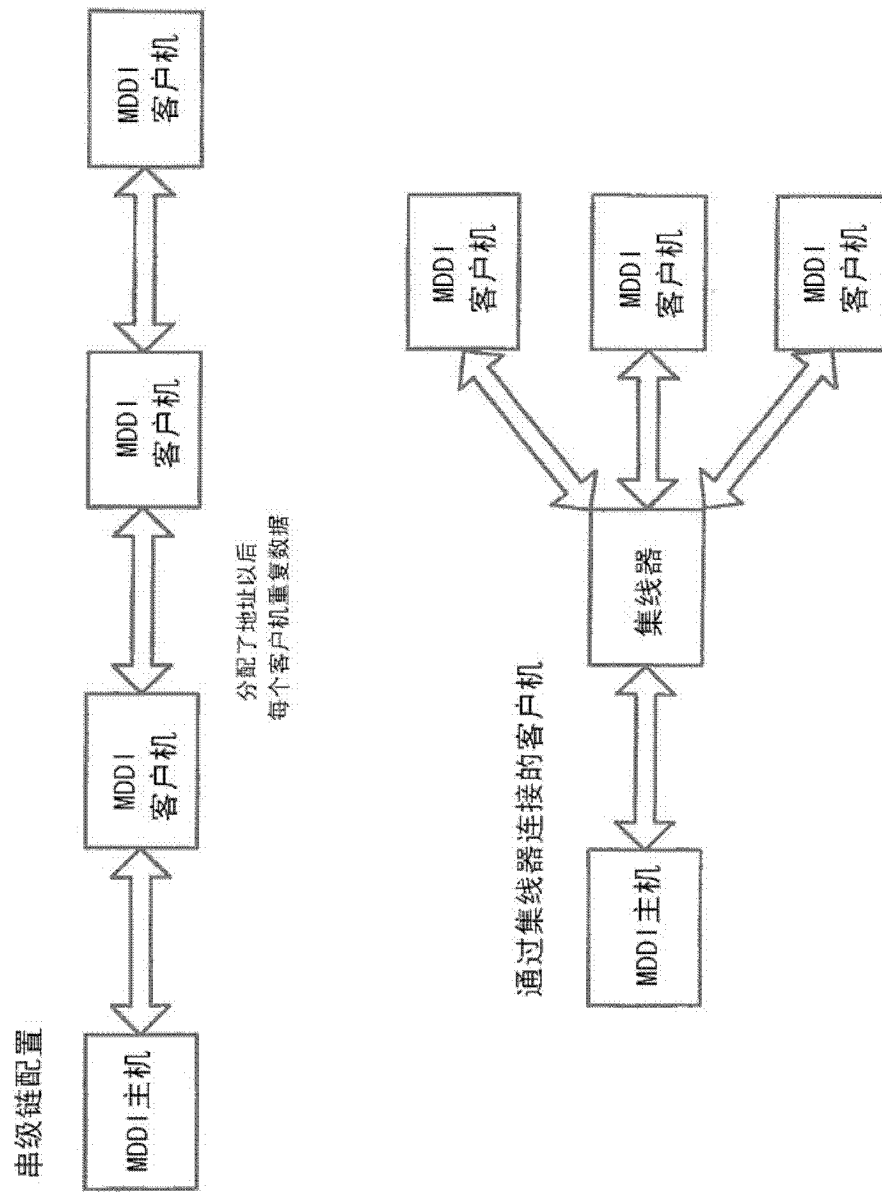


图 95

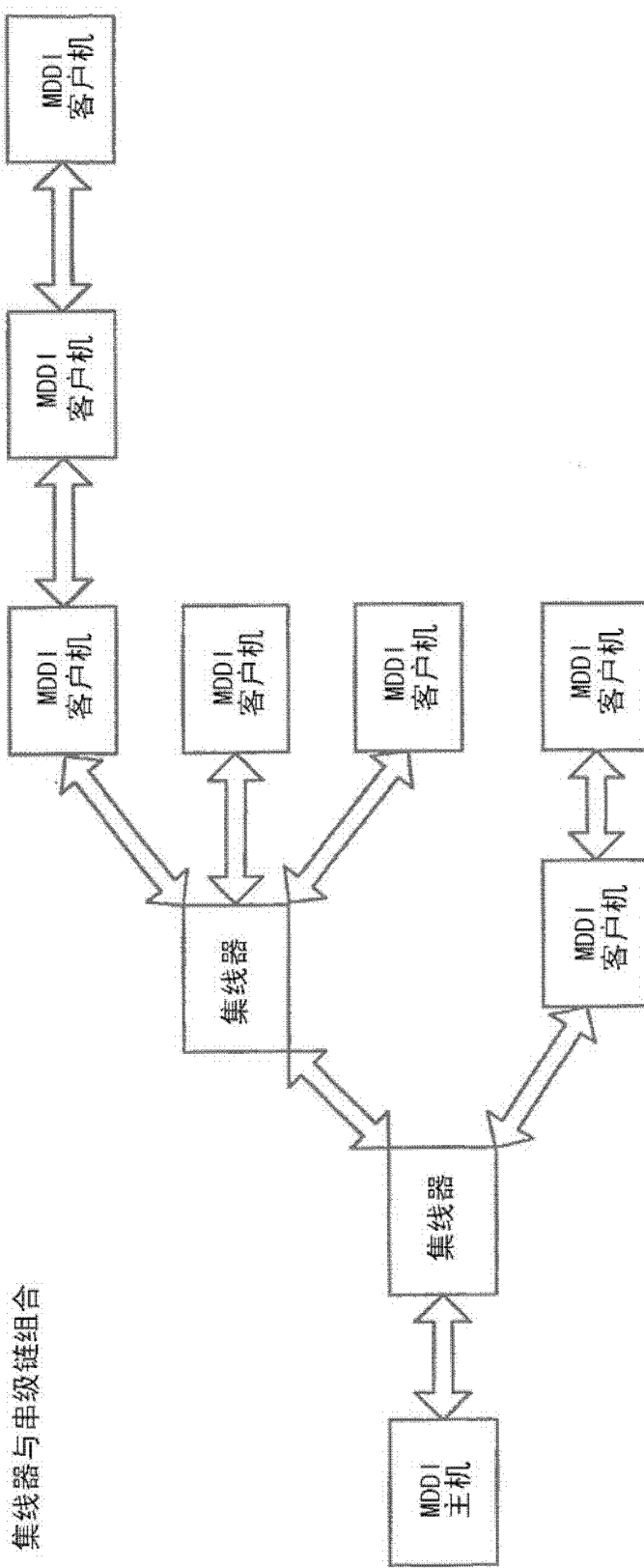


图 96

6位蓝色、8位绿色、7位红色的实例

色彩映射数据格式化

←	MDDI 字节 <sub>m</sub>						→	MDDI 字节 <sub>m+1</sub>						→	MDDI 字节 <sub>m+2</sub>						→	MDDI 字节 <sub>m+3</sub>						→	MDDI 字节 <sub>m+4</sub>						→	MDDI 字节 <sub>m+5</sub>											
5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0				
参数CRC						项 <sub>n</sub> 蓝色						项 <sub>n</sub> 绿色						项 <sub>n</sub> 红色						项 <sub>n+1</sub> 蓝色						项 <sub>n+1</sub> 绿色						项 <sub>n+1</sub> 红色						未使用					

图 97