



[12] 发明专利申请公布说明书

[21] 申请号 200480030030.1

[43] 公开日 2008年6月4日

[11] 公开号 CN 101194482A

[22] 申请日 2004.8.12

[21] 申请号 200480030030.1

[30] 优先权

[32] 2003.8.13 [33] US [31] 60/494,983

[86] 国际申请 PCT/US2004/026264 2004.8.12

[87] 国际公布 WO2005/018191 英 2005.2.24

[85] 进入国家阶段日期 2006.4.13

[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·艾伦·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

代理人 覃鸣燕

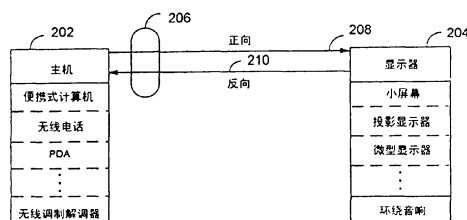
权利要求书 12 页 说明书 156 页 附图 84 页

[54] 发明名称

用于更高数据速率的信号接口

[57] 摘要

一种使用链接在一起的数据包结构来经由通信路径在主机和客户机之间传送数字数据的数据接口，所述数据包结构链接起来以便形成用于交换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数据包，并且使数字数据形成一个或多个类型的数据数据包，其中至少一个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节约成本、低功率、双向、高速数据传输机制，可以借助特别适用于连接诸如可佩带的微型显示器的显示器元件和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. 一种用于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呈现数据的数字数据接口，包括：

多个数据包结构，链接在一起以便形成通信协议，用于经由所述通信路径在主机和客户机之间传递数字控制和呈现数据的预选集合；以及

驻留在所述主机设备中的至少一个链路控制器，经由所述通信路径耦合至所述客户机，被配置成生成、传输并且接收形成所述通信协议的数据包，并且将数字呈现数据形成一个或多个类型的数据数据包。

2. 如权利要求 1 所述的接口，还包括在介质帧内组合在一起的所述数据包，其中所述介质帧在所述主机和客户机之间传递，且具有预定义固定长度，预定数目的所述数据包具有不同的并且可变的长度。

3. 如权利要求 1 所述的接口，还包括位于从所述主机传送数据包的开始的子帧报头数据包。

4. 如权利要求 1 所述的接口，其中所述链路控制器是主机链路控制器，并且还至少包括一个客户机链路控制器，该客户机链路控制器驻留在经由所述通信路径耦合至所述主机的所述客户机设备中，被配置成生成、传输并且接收形成所述通信协议的数据包，并且将数字呈现数据形成为一个或多个类型的数据数据包。

5. 如权利要求 1 所述的接口，还包括用于视频类型数据的一个或多个视频流数据包，以及用于音频类型数据的音频流数据包，以便把数据从所述主机经由正向链路传送到所述客户机，从而呈现给客户机用户。

6. 如权利要求 2 所述的接口, 还包括:

多个传送模式, 每个传送模式均允许在给定时间段上并行传送最大数目不同的数据位, 每一模式都可以通过所述主机和客户机链路驱动器之间的协商来选择; 以及

其中所述传送模式在传送数据期间是在所述模式之间动态地调整的。

7. 如权利要求 1 所述的接口, 还包括多个数据包, 这些数据包能够用来传送从色彩映射、位块传送、位图区域填充、位图图案填充以及透明色彩启用类型数据包构成的组中选择出来的视频信息。

8. 如权利要求 1 所述的接口, 还包括由所述主机生成的填充符类型数据包, 用于占据没有数据的正向链路传输的周期。

9. 如权利要求 1 所述的接口, 还包括用户定义的流类型数据包, 用于传送接口用户定义的数据。

10. 如权利要求 1 所述的接口, 还包括链路关闭类型数据包, 用于由所述主机传输给所述客户机, 以便终止沿任何一个方向经由所述通信路径的数据传送。

11. 如权利要求 1 所述的接口, 还包括用于让所述客户机把所述主机从休眠状态唤醒的模块。

12. 一种用于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以便呈现给用户的方法, 包括:

生成多个预定义数据包结构的一个或多个, 并且把它们链接在一起以便形成预定义通信协议;

使用所述通信协议经由所述通信路径在所述主机和所述客户机

设备之间传递数字控制和呈现数据的预选集合；

经由所述通信路径把驻留在所述主机设备中的至少一个主机链路控制器耦合至所述客户机设备，所述主机链路控制器被配置成生成、传输并且接收形成所述通信协议的数据包，并且将数字呈现数据形成为一个或多个类型的数据数据包；并且

使用所述链路控制器经由所述通信路径以数据包的形式来传送数据。

13. 如权利要求 12 所述的方法，还包括在介质帧内把所述数据包组合在一起，以便在所述主机和客户机之间通信，所述介质帧具有预定义的固定长度，且预定数目的所述数据包具有不同的并且可变的长度。

14. 如权利要求 12 所述的方法，还包括利用子帧报头类型数据包开始从所述主机传送数据包。

15. 如权利要求 12 所述的方法，还包括经由所述通信链路在所述主机和客户机之间双向地传送信息。

16. 如权利要求 12 所述的方法，还包括至少一个客户机链路控制器，该客户机链路控制器驻留在经由所述通信路径耦合至所述主机设备的所述客户机设备中，且被配置成生成、传输并且接收形成所述通信协议的数据包，并且将数字呈现数据形成为一个或多个类型的数据数据包。

17. 如权利要求 16 所述的方法，其中所述主机链路控制器包括一个或多个差分线路驱动器；并且所述客户机链路控制器包括耦合至所述通信路径的一个或多个差分线路接收器。

18. 如权利要求 12 所述的方法，还包括由主机链路控制器从所

述客户机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。

19. 如权利要求 12 所述的方法，还包括由所述链路控制器的每个来将 USB 数据接口用作所述通信路径的一部分。

20. 如权利要求 12 所述的方法，其中每一个所述数据包均包括数据包长度字段、一个或多个数据包数据字段和循环冗余校验字段。

21. 如权利要求 13 所述的方法，还包括：

在所述主机和客户机链路控制器之间协商每一方向上多个传送模式之一的使用，每一个模式均允许在给定时间段上并行传送最大数目不同的数据位；并且

在传送数据期间，在所述传送模式之间进行动态调整。

22. 如权利要求 12 所述的方法，还包括使用多个数据包的一个或多个来传送从色彩映射、位块传送、位图区域填充、位图图案填充以及透明色彩启用类型数据包构成的组中选择出来的视频信息。

23. 如权利要求 12 所述的方法，还包括由所述主机生成填充符类型数据包，用于占据没有数据的正向链路传输的周期。

24. 如权利要求 12 所述的方法，还包括使用由所述主机传输到所述客户机的链路关闭类型数据包，终止沿任何一个方向经由所述通信路径的数据传送。

25. 如权利要求 12 所述的方法，还包括通过与所述客户机的通信将所述主机从休眠状态中唤醒。

26. 用于在主机设备和客户机设备之间经由通信路径以高速率传

送数字数据以便呈现给用户的装置，包括：

至少一个主机链路控制器，设置在所述主机设备中，用于生成多个预定义数据包结构的一个或多个，并且把它们链接在一起以便形成预定义的通信协议，并且用于使用所述通信协议在所述主机和所述客户机设备之间经由所述通信路径来传递数字控制和呈现数据的预选集合；

至少一个客户机控制器，设置在所述客户机设备中，并且经由所述通信路径耦合至所述主机链路控制器；并且

每个链路控制器被配置成生成、传输并且接收形成所述通信协议的数据包，并且将数字呈现数据形成为一个或多个类型的数据数据包。

27. 如权利要求 26 所述的装置，其中所述主机控制器包括状态机。

28. 如权利要求 26 所述的装置，其中所述主机控制器包括通用信号处理器。

29. 如权利要求 26 所述的装置，还包括位于从所述主机传送的数据包开始处的子帧报头类型的数据包。

30. 如权利要求 26 所述的装置，其中所述链路控制器被配置成经由所述通信链路在所述主机和客户机设备之间双向地传送信息。

31. 如权利要求 30 所述的装置，其中所述主机控制器包括一个或多个差分线路驱动器；并且所述客户机接收器包括耦合至所述通信路径的一个或多个差分线路接收器。

32. 如权利要求 26 所述的装置，还包括把数据从所述主机传送到所述客户机以便呈现给客户机用户时，用于视频类型数据的视频流

类型数据包以及用于音频类型的音频流类型数据包。

33. 如权利要求 26 所述的装置，还包括一个或多个反向链路封装类型数据包，用于把数据从所述客户机传送到所述主机。

34. 如权利要求 33 所述的装置，还包括至少一个显示器能力类型数据包，用于把显示或者呈现能力从客户机链路控制器传递给所述主机链路控制器。

35. 如权利要求 26 所述的装置，其中所述数据包每一个均包括数据包长度字段、一个或多个数据包数据字段和循环冗余校验字段。

36. 如权利要求 26 所述的装置，其中所述主机和客户机链路控制器被配置成沿每个方向使用多个传送模式之一，每一个均允许在给定时间段上并行传送最大数目不同的数据位；并且能够在传送数据期间在所述传送模式之间进行动态调整。

37. 如权利要求 26 所述的装置，还包括多个数据包的一个或多个，用于传送从色彩映射、位块传送、位图区域填充、位图图案填充以及透明色彩启用类型数据包的组中选择的视频信息。

38. 如权利要求 26 所述的装置，还包括由所述主机传送的填充符类型数据包，用于占据没有数据的正向链路传输的周期。

39. 如权利要求 26 所述的装置，其中所述主机控制器被配置成向所述客户机模块传输链路关闭类型数据包，用于终止沿任何一个方向经由所述通信路径的数据传送。

40. 一种计算机程序产品，用于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以便呈现给用户的电子系统，包

括：

计算机可用介质，具有嵌入所述介质的计算机可读程序代码模块，用于让应用程序在所述计算机系统上执行，所述计算机可读程序代码模块包括：

计算机可读第一程序代码模块，用于让所述计算机系统生成多个预定义数据包结构的一个或多个，并且把它们链接在一起以便形成预定义通信协议；

计算机可读第二程序代码模块，用于让所述计算机系统使用所述通信协议在所述主机和所述客户机设备之间经由所述通信路径传递数字控制和呈现数据的预选集合；

计算机可读第三程序代码模块，用于让所述计算机系统把设置在所述主机设备中的至少一个主机链路控制器经由所述通信路径耦合到设置在所述客户机设备中的至少一个客户机控制器，所述链路控制器被配置成生成、传输并且接收形成所述通信协议的数据包，并且将数字呈现数据形成为一个或多个类型的数据数据包；以及

计算机可读第四程序代码模块，用于让所述计算机系统使用所述链路控制器经由所述通信路径以数据包的形式传送数据。

41. 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以便呈现给用户的装置，包括：

用于生成多个预定义数据包结构的一个或多个并且把它们链接在一起以便形成预定义通信协议的模块；

用于使用所述通信协议在所述主机和所述客户机设备之间经由所述通信路径来传递数字控制和呈现数据的预选集合的模块；

用于把至少两个链路控制器经由所述通信路径耦合在一起的模块，在所述主机和客户机的每一个中各有一个该模块，并且均被配置成生成、传输并且接收形成所述通信协议的数据包，并且将数字呈现数据形成为一个或多个类型的数据数据包；以及

用于使用所述链路控制器经由所述通信路径以数据包的形式来传送数据的模块。

42. 如权利要求 41 所述的装置，还包括用于利用子帧报头类型数据包开始从所述主机传送数据包的模块。

43. 如权利要求 41 所述的装置，还包括经由所述通信链路在所述主机和客户机之间双向地传送信息的模块。

44. 如权利要求 41 所述的装置，还包括用于通过主机链路控制器从客户机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的模块。

45. 如权利要求 44 所述的装置，还包括用于使用至少一个显示器能力类型数据包把显示或者呈现能力从客户机链路控制器传递给所述主机链路控制器的模块。

46. 如权利要求 42 所述的装置，还包括：

用于在所述主机和客户机链路驱动器之间协商沿每一方向多个传送模式之一的使用的模块，每一个模式允许在给定时间段上并行传送最大数目不同的数据位；

以及用于在传送数据期间，在所述传送模式之间进行动态调节的模块。

47. 如权利要求 41 所述的装置，还包括用于使用多个数据包的一个或多个来传送从色彩映射、位块传送、位图区域填充、位图图案填充以及透明色彩启用类型数据包的组中选择的视频信息的模块。

48. 一种用于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电子系统的处理器，所述处理器被配置成生成多个预定义数据包结构的一个或多个并且把它们链接在一起以便形成预定义通信协议；将数字呈现数据形成为一个或多个类型的数据数据

包；使用所述通信协议在所述主机和所述客户机设备之间经由所述通信路径来传递数字控制和呈现数据的预选集合；并且经由所述通信路径以数据包的形式传送数据。

49. 一种用于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电子系统中用来获得同步的状态机，所述状态机被配置成具有至少一个异步帧状态的同步状态、至少两个获得同步状态的同步状态以及至少三个同步中状态的同步状态。

50. 一种用于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电子系统中用来获得同步的状态机，所述状态机被配置成具有至少一个获得同步状态的同步状态以及至少两个同步中状态的同步状态。

51. 如权利要求 50 所述的状态机，其中在获得同步状态和第一同步中状态之间变换的一个条件是在通信链路中检测到存在同步模式。

52. 如权利要求 51 所述的状态机，其中在获得同步状态和第一同步中状态之间变换的第二个条件是在帧边界处检测到存在子帧报头数据包和良好的 CRC 值。

53. 如权利要求 50 所述的状态机，其中在第一同步中状态和获得同步状态之间变换的一个条件是在子帧边界处检测到不存在同步模式或者存在坏 CRC 值。

54. 如权利要求 50 所述的状态机，其中在第一同步中状态和第二同步中状态之间变换的一个条件是在子帧边界处检测到不存在同步模式或者存在坏 CRC 值。

55. 如权利要求 50 所述的状态机, 其中在获得同步状态和第一同步中状态之间变换的一个条件是在通信链路中检测到存在同步模式, 并且检测到存在良好的数据包 CRC 值。

56. 如权利要求 50 所述的状态机, 其中在第一同步中状态和获得同步状态之间变换的条件是在数据包中检测到存在坏的 CRC 值。

57. 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电子系统中用来获得同步的状态机, 所述状态机被配置成具有至少一个获得同步状态的同步状态和至少两个同步中状态的同步状态, 其中在第一同步中状态和获得同步状态之间直接变换的条件是在一个数据包系列中任意一个里检测到存在坏的 CRC 值。

58. 如权利要求 57 所述的状态机, 其中在第一同步中状态和获得同步状态之间直接变换的条件是检测到唯一字没有出现在期望其到达的时候。

59. 如权利要求 26 所述的方法, 还包括由所述主机通过把数据线驱动至高状态且至少持续 10 个时钟周期, 并且按照就所述数据线好像是零一样的方式开始传输选通信号, 从而唤醒通信链路。

60. 如权利要求 59 所述的方法, 还包括在所述主机已经把所述数据线驱动为高状态达 150 个时钟周期之后, 在继续传输选通信号期间, 由所述主机把数据线驱动为低状态且持续 50 个时钟周期。

61. 如权利要求 59 所述的方法, 还包括由所述主机开始传输第一个子帧报头数据包。

62. 如权利要求 60 所述的方法, 还包括由所述客户机对处于高状态的数据线的至少 150 个连续时钟周期进行计数, 继之对处于低状

态的数据线的至少 50 个连续时钟周期进行计数。

63. 如权利要求 62 所述的方法, 还包括由所述客户机搜索第一子帧的唯一字。

64. 如权利要求 60 所述的方法, 还包括所述客户机对处于高状态的数据的 70 个连续时钟周期计数之后, 停止把数据线驱动为高状态。

65. 如权利要求 64 所述的方法, 还包括由所述客户机对处于高状态的数据线的另外 80 个连续时钟周期进行计数, 以便达到处于高的数据线的 150 个时钟周期, 并且寻找处于低状态的数据线的 50 个时钟周期, 并且寻找所述唯一字。

66. 如权利要求 26 所述的方法, 还包括通过在反向时序数据包期间, 在上升沿和下降沿上对数据线采样, 对发生的时钟周期数目进行计数, 直到一个 1 被所述主机采到为止。

67. 一种在通信系统中传送错误代码的方法, 在所述通信系统中, 数字数据以具有 CRC 值的数据包形式在主机设备和客户机设备之间经由通信路径传送, 所述方法包括检测错误的存在, 选择对应于所述错误的预定错误代码, 并且利用所述代码来重写所述 CRC 值。

68. 如权利要求 67 所述的方法, 还包括在被传送的连续数据包中重写所述 CRC 值, 直到所述错误被纠正为止。

69. 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以便呈现给用户的方法, 包括:

生成多个预定义数据包结构的一个或多个, 每一个都包括至少一个 CRC 字段, 并且把它们链接在一起以便形成预定义通信协议;

使用所述通信协议在所述主机和所述客户机设备之间经由所述通信路径来传递数字控制和呈现数据的预选集合；

经由所述通信路径把驻留在所述主机设备中的至少一个主机链路控制器耦合至所述客户机设备，所述主机链路控制器被配置成生成、传输并且接收形成所述通信协议的数据包，并且将数字呈现数据形成为一个或多个类型的数据数据包；

使用所述链路控制器经由所述通信路径以数据包的形式来传送数据；

检测通信链路的错误的存在；

选择对应于所述错误的预定错误代码；并且利用所述代码重写所述 CRC 值。

70. 如权利要求 69 所述的方法，还包括重写传送的连续数据包中的所述 CRC 值，直到所述错误被纠正为止。

用于更高数据速率的信号接口

相关申请的交叉引用

本专利申请要求于 2003 年 8 月 13 日提交，标题为“MDDI Specification Advancements”，转让给本受让人的第 60/494,983 号临时专利申请的优先权，在此明确将其引入作为参考。

技术领域

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

近几年，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）获得了明显的进展，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感受。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越来越多的期望。

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，最好将这一速率称为慢速或中速。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。

一旦收到数据，就把数据就地储存在存储器元件、电路或者设备中，例如储存在 RAM 或者快闪存储器中，还包括储存在外部存储设备中，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，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，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（reasonable）透明的。

用于产生静止图像或者运动视频的数据经常使用几个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。

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，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元件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 480×640 像素到 600×800 再到 1024×1024 ，当然根据需要或要求，一般还可能有各种其它分辨率。

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），虽然也会遇到其它值。

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 2.45 兆比特（Mb）到 33.55 Mb 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给传送增加了更多的数据。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感受时，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。

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 12 MBps 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 400 MBps 数量级的速率进行。遗憾的是，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。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必要的大量开销，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，对于要求美观的移动应用而言，这些大体积的电缆太重，不能令人满意，这些复杂的连接器增加了成本，或者是功耗太大。

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。

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。

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。

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感受，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。

在后者的情况下，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 and 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。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 个导线来实现所期望的吞吐量。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。

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。

在序号为 10/020,520 和 10/236,657，标题均为“Generating and Implementing a Communication Protocol and Interface for High Data Rate Signal Transfer”，目前已授权的美国专利申请中，申请人已经提出了这种新的传送机制，这两篇申请已经转让给本发明的受让人，在这里将它们引入作为参考。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

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。

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数据包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。信号通信协议或者链路层由主机或者客户机链路控制器的物理层使用。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、传输并且接收形成通信协议的数据包, 并且将数字呈现数据形成为一个或多个类型的数据数据包。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。

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 6 个左右的信号, 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。

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。

按照本发明的实施例的另一方面, 至少一个客户机链路控制器或者客户机接收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数据包, 并且将数字呈现数据形成为一个或多个类型的数据数据包。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数据包, 但是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数据包。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。

所述数据包被组合在介质帧内，这些具有预定义固定长度的介质帧利用具有不同的可变长度的预定数目数据包在主机和客户机设备之间通信。所述数据包均包括数据包长度字段、一个或多个数据包数据字段和循环冗余校验字段。子帧报头数据包被传送或者定位在来自主机链路控制器的其它数据包传送的开头。所述通信协议使用一个或多个视频流类型数据包和音频流类型数据包，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数据包，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。

由所述主机链路控制器生成填充符类型数据包，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数据包，以便传送视频信息。这些数据包包括色彩映射、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数据包。所述通信协议使用用户定义的流类型数据包以便传送接口-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数据包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数据包来终止沿任何一个方向经由所述通信路径的数据传送。

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，其中有一些驻留在柔性衬底上。

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数据包向主机链路控制器传递显示或者呈现能力信息。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上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如同在正向链路上那样使用同一模式。

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。

在某些实施例的又一方面，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。这一结构的典型客户机设备包括客户机电路、集成电路或者耦合至主机的组件，并且驻留在同一设备中，并且耦合至移动电话的高分辨率屏幕之类的内部视频显示器和/或便携式音频呈现系统。

附图说明

下面参考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。

图 1A 说明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结合使用的微型显示器设备的使用。

图 1B 说明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和音频呈现元件的使用。

图 1C 说明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结合使用的微型显示器设备的使用。

图 1D 说明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和音频呈现元件的使用。

图 2 说明具有主机和客户机互连的移动数字数据接口的总体概念。

图 3 说明用于实现从客户机设备到主机设备的数据传送的数据包结构。

图 4 说明 MDDI 链路控制器的使用，以及在主机和客户机之间经由 I 型和 U 型接口的物理数据链路导线传递的信号类型。

图 5 说明 MDDI 链路控制器的使用，以及在主机和客户机之间经由 II 型、II 型和 IV 型接口的物理数据链路导线传递的信号类型。

图 6 说明用于实现所述接口协议的帧和子帧的帧结构。

图 7 说明用于实现所述接口协议的数据包的通用结构。

图 8 说明子帧报头数据包的格式。

图 9 说明填充符数据包的格式和内容。

图 10 说明视频流数据包的格式。

图 11 说明图 10 的视频数据格式描述符的格式和内容。

图 12 说明数据的已打包和已拆包格式的使用。

图 13 说明音频流数据包的格式。

图 14 说明数据的按字节对准和已打包 PCM 格式的使用。

图 15 说明用户定义的流数据包的格式。

图 16 说明色彩映射数据包的格式。

图 17 说明反向链路封装数据包的格式。

图 18 说明显示器能力数据包的格式。

图 19 说明键盘数据数据包的格式。

图 20 说明指示设备数据数据包的格式。

图 21 说明链路关闭数据包的格式。

图 22 说明显示器请求和状态数据包的格式。

图 23 说明位块传送数据包的格式。

图 24 说明位图区域填充数据包的格式。

图 25 说明位图图案填充数据包的格式。

图 26 说明通信链路数据通道数据包的格式。

图 27 说明接口类型切换请求数据包的格式。

图 28 说明接口类型确认数据包的格式。

图 29 说明执行类型切换数据包的格式。

图 30 说明正向音频通道启用数据包的格式。

图 31 说明反向音频采样速率数据包的格式。

图 32 说明数字内容保护开销数据包的格式。

图 33 说明透明色彩启用数据包的格式。

图 34 说明往返行程延迟测量数据包的格式。

图 35 说明往返行程延迟测量数据包期间事件的时序。

图 36 说明用于实现本发明的 CRC 生成器和校验器的示例性实现方式。

图 37A 说明图 36 所示装置发送数据数据包时 CRC 信号的时序。

图 37B 说明图 36 所示装置接收数据数据包时 CRC 信号的时序。

图 38 说明没有竞争的典型服务请求的处理步骤。

图 39 说明在链路重启序列开始之后维持 (assert) 典型服务请求的处理步骤, 以及链路启动竞争的步骤。

图 40 说明如何使用 DATA-STB 编码来传输数据序列。

图 41 说明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。

图 42 说明可用于实现一个实施例的驱动器和终端电阻。

图 43 说明由客户机采用以便保证来自主机的服务的安全, 以及主机用来提供这种服务的步骤和信号电平。

图 44 说明 Data0、其它数据线 (DataX) 和选通线 (Stb) 上的转变之间的相对间隔。

图 45 说明传送数据包之后主机禁用主机驱动器时会出现的, 作为响应而存在的延迟。

图 46 说明主机启用主机驱动器传送数据包时会出现的, 作为响应而存在的延迟。

图 47 说明主机接收器输入端正在传送的数据的时序与选通脉冲前后沿之间的关系。

图 48 说明反向数据时序引发的切换特性和相应的客户机输出延迟。

图 49 说明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。

图 50 说明在采用 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。

图 51 说明边缘往返行程延迟测量。

图 52 说明反向链路数据速率的变化。

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。

图 54A 和 54B 说明接口操作中进行的步骤。

图 55 说明处理数据包的接口装置的概况。

图 56 说明正向链路数据包的格式。

图 57 说明 I 型链路接口中的传播延迟和偏离 (skew) 的典型值。

图 58 说明对经由所述接口的示例性信号处理而言, 在 I 型链路上的 Data、Stb 和时钟恢复时序。

图 59 说明 II 型、III 型或者 IV 型链路接口中传播延迟和偏离的典型值。

图 60A、60B 和 60C 说明两个数据信号和 MDDI_Stb 之间时续的不同可能性, 分别对应于理想、提早和推迟。

图 61 说明 I 型/II 型接口所用示例性连接器的接口管脚分配。

图 62A 和 62B 分别说明 I 型和 II 型接口可能的 MDDI_Data 和 MDDI_Stb 波形。

图 63 说明能够使用状态机实现同步的替代信号处理步骤和条件的高级图表。

图 64 说明一系列时钟周期之间的示例性相对时续和各种反向链路数据包位和除数值的时序。

图 65 说明示例性的错误代码传送处理。

图 66 说明可用于错误代码传送处理的装置。

图 67A 说明用于代码重载的错误代码传送处理。

图 67B 说明用于接收代码的错误代码传送处理。

图 68A 说明主机启动的唤醒的处理步骤。

图 68B 说明客户机启动的唤醒的处理步骤。

图 68C 说明带有竞争的主机和客户机启动的唤醒的处理步骤。

图 69 说明请求 VCP 特征数据包的格式。

图 70 说明 VCP 特征应答数据包的格式。

图 71 说明 VCP 特征应答列表的格式。

图 72 说明设置 VCP 特征数据包的格式

图 73 说明请求有效参数数据包的格式。

图 74 说明有效参数应答数据包的格式。

图 75 说明阿尔法光标图像能力数据包的格式。

图 76 说明阿尔法光标透明度映射数据包的格式。

图 77 说明阿尔法光标图像偏移数据包的格式。

图 78 说明阿尔法光标视频流数据包的格式。

图 79 说明可缩放视频流能力数据包的格式。

图 80 说明可缩放视频流设置数据包的格式。

图 81 说明可缩放视频流确认数据包的格式。

图 82 说明可缩放视频流数据包的格式。

图 83 说明请求专用状态数据包的格式。

图 84 说明有效状态应答列表数据包的格式。

图 85 说明数据包处理延迟参数数据包的格式。

图 86 说明个人显示器能力数据包的格式。

图 87 说明显示器错误报告数据包的格式。

图 88 说明显示器标识数据包的格式。

图 89 说明可选显示器能力数据包的格式。

图 90 说明寄存器访问数据包的格式。

图 91A-91C 说明使用两个显示缓冲器来减少可视非自然信号
(visible artifacts)。

图 92 说明显示刷新快于图像传送的两个缓冲器。

图 93 说明显示刷新慢于图像传送的两个缓冲器。

图 94 说明显示刷新比图像传送快得多的两个缓冲器。

图 95 说明显示刷新快于图像传送的三个缓冲器。

图 96 说明显示刷新慢于图像传送的三个缓冲器。

图 97 说明显示刷新快于图像传送的一个缓冲器。

图 98 说明经由串级链和集线器的主机-客户机连接。

图 99 说明经由集线器和串级链相连的客户机设备。

具体实施方式

I. 概述

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成本效益、低功耗的传送机制, 使用“串行”类型的数据链路或者通道, 能够在主机设备和显示器元件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(至外壳或者支架)显示器元件或者输入设备连接至中央控制器, 或者把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连接至便携式计算机、无线通信设备或者娱乐设备。

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。

可以将本发明的实施例用于各种情况, 把通常用于音频、视频或者多媒体应用的大量数据从生成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显示器或者呈现设备。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（一起在同一个设备外壳或者支撑结构内）主机。

MDDI 的特性或属性不依赖于具体的显示技术。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，这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数据包的时序以适应特定客户机设备的特性，例如某些设备的独特显示需求，或者满足某些 A-V 系统或者象游戏操纵杆、触摸盘等等的某些输入设备的组合音频和视频的要求。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优化成本、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度。

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，包括基于光学的链路，只要将这些链路配置成使用为这一接口协议开发的相同数据包和数据结构，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。

II. 环境

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 和无线电话或者 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。

计算机 100 具有非常大的屏幕，但是外部音响系统仍不够，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此种设备相连，但是不局限于此。

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益或令人愉快的感受。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感受的最低水平。

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，非常小的投影元件允许用户的眼睛“看到”比利用典型液晶屏幕等等看到的大得多的图像。使用较大的虚拟屏幕图像还允许使用比利用更加有限的 LCD 屏幕显示器能够获得的高得多的分辨率的图像。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。

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。

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：数字音乐文件允许高度增加的特性丰富的感受，但是只有当解码和播放处理能够跟上时才行。这对于数字视频文件也是一样。

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 作为嵌入支架或者图 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。

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 1024×1024 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 30 fps 的数据速率来说，数据速率能够接近超过 755 Mbps 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。

此外显而易见的是，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，并且越可能有更多的用户采用。在通常使用多个设备来建立完全的音频-视频感受的情况下，这一点是尤其正确的，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。

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。

在图 1C 和 1D 中，用整个电子设备或者产品的切开 (cut-away) 部分来示出设备的一部分中一个或多个内部主机和控制器的位置，穿过某些已知类型的旋转连接，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，其中所使用的旋转连接的类型可以是整个电子行业目前使

用的那些。人们可以看出，这些传送包含的数据量要求大量导线来构成链路 138 和 148。据估计，这种通信链路接近于 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，这是由于并行或者其它已知接口技术的类型可用于传送这种数据。

使人遗憾的是，这种更高数据速率超出了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。

所需要的是这样一种技术，即，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，这种技术允许一贯的低功耗、重量轻并且尽可能简单并且经济的电缆结构。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，以便允许一系列移动式、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。

III. 高速率数字数据接口系统体系结构

为了创建并且有效地利用新的设备接口，已经配制了使用低功率信号提供很高的数据传送速率的信号协议和系统体系结构。所述协议基于数据包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。

A. 概述

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，尽管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而形成的, 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, 立即解决了当前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方式所面对的许多问题之一。

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便携式计算机, 它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, 诸如便携式 DVD 或者 CD 播放器, 或者是游戏设备。

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, 可以将主机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, 具有改善了的显示器能力, 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, 而且通过利用连接器或者电缆中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。

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。另一实例是使用触摸板或者敏感设备、语音识别输入设备、安全扫描仪等等, 这些设备可能利用除

了来自用户的触摸和声音以外的不太实际的“输入”，而想要从用户那里传送数量巨大的信息。

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，从而能够适应实现所期望的用户感受所需的高数据速率。

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）的主机处理器和显示器之间的互联，以降低这些连接的成本并且提高可靠性，而不仅仅用于外部元件（外部模式）。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 的属性不依赖显示技术。可以容易地调节经由接口传送的数据包的时序，以适应特定显示器设备或者音频-视频系统的综合时序要求的特性。虽然这样做允许系统消耗尽可能小的功率，但是它并不要求显示器具有帧缓冲器以便使用 MDDI。

B. 接口类型

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物理上多少有些不同的五个或者更多类型的接口。将这些类型的接口简单标记为 I 型、II 型、III 型、IV 型和 U 型，当然本领域技术人员也可以应用其它标记或者标识，这取决于这些接口的应用环境。

所述 I 型接口被配置成 6 线（导线）接口，这使其适用于移动式或者无线电话、PDA、电子书籍、电子游戏和诸如 CD 播放器或者 MP3 播放器的便携式介质播放器，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，U 型接口被配置成 8 线（导线）接口，它更加适用于膝上型、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地更新显示，也没有嵌入式 MDDI 链路控制器。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 在支持大多数个人计算机上常见的现存操作系统或者软件支持方面非常有用。U 型接口还可以用于 USB 单一模式，其中显示器只有一个 USB 连接器，该连接器与计算机或者类似设备上的标准 USB 端口连接，其中所述类似设备例如是配备有这种端口的消费电子设备，诸如数字照相机或者视频播放器。

II 型、III 型和 IV 型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。

所述 I 型接口传递包括显示、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 30 fps 外加 5.1 声道音频的情况下，I 型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是为诸如移动无线设备之类的设备而设计的，这种设备内通常没有 USB 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”，其中的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。

在这种接口中，通过向客户机发送特殊的命令或者数据包类型，主机能够在该处接收来自客户机的通信数据(反向业务或者链路)，其中这些命令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数据包发送到主机。这些内容在图 3 中示出，其中称为封装数据包的数据包类型(将在下面讨论)用来支持经由传送链路传送反向数据包，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。

能够显示 HDTV 类型或者类似高分辨率的高性能显示器要求 1.5 Gbps 左右速率的数据流以支持全运动视频。所述 II 型接口通过并行

传输 2 位来支持高数据速率，所述 Type-I/III 通过并行传输 4 位来支持，而 IV 型接口并行传送 8 位。II 型和 III 型使用与 I 型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。IV 型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。

通常，通过议定什么是可以使用的最高数据速率，由 MDDI 使用的协议允许 I 型、II、III 或者 IV 主机的每一个与任何 I 型、II、III 或者 IV 客户机通信。将可以称为最小能力设备的能力或者可用特征用来设置链路的性能。通常，即使是主机和客户机均能够使用 II 型、III 型或者 IV 型接口的系统，二者都以 I 型接口开始工作。然后，主机确定目标客户机的能力，并且议定对 II 型、III 型或 IV 型模式执行切换还是重新配置操作，只要对于特定的应用合适就行。

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逐级下降或者再次重新配置为较慢模式以便节省功耗，或者逐级上升为较快模式来支持诸如高分辨率显示内容的更高速率传送。例如，当系统从诸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。

系统还能够在一个方向上使用一种模式在另一个方向上使用另一模式来传递数据。例如，可以将 IV 型接口模式用于以高速率传送数据给显示器，而从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 I 型或者 U 型模式。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。

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 MDDI 把一个设备中的主机连接到该设备外部的，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，例如在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，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。

C. 物理接口结构

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。由于更大规模地制造单个电路设备，这样做的成本更低。

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。

在图 4 和 5 中还说明在主机和诸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低压数据信号，且经由电缆中的差分电线对来传送。对于通过所述接口发送的每一位，MDDI_Data0 对或者 MDDI_Stb 对上只存在一个转变。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。

当数据可以经由 MDDI_Data 对沿正反方向流动，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 1 kbps 直到 400 Mbps 或更大的范围内改变。

除了 I 型接口的数据对或路径以外，II 型接口还包含另外一个数据对或者导线或者路径，称为 MDDI_Data1 +/-。除了 II 型接口的那些数据对或路径以外，III 型接口包含另外两个数据对或者信号路径，称为 MDDI_Data2 +/- 和 MDDI_Data3 +/-。除 III 型接口的数据对或路径以外，IV 型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 +/-。在每一上述接口配置中，主机可以使用线对或者指定为 MDDI_Pwr 和 MDDI_Gnd 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”采用的导线比在其他模式中可用的或存在的导线少时，在基于 MDDI_Data4 +/-、MDDI_Data5 +/-、MDDI_Data6 +/- 或者 MDDI_Data7 +/- 导线的某些配置中，也可以进行电力传送。

下面，按照接口类型在表 I 中说明在各种模式下，经由 MDDI 链路在主机和客户机（显示器）之间传递的信号概要。

表 I

<i>I 型</i>	<i>II 型</i>	<i>III 型</i>	<i>IV 型</i>
MDDI_Pwr/Gnd	MDDI_Pwr/Gnd	MDDI_Pwr/Gnd	MDDI_Pwr/Gnd
MDDI_Stb+/-	MDDI_Stb+/-	MDDI_Stb+/-	MDDI_Stb+/-
MDDI_Data0+/-	MDDI_Data0+/-	MDDI_Data0+/-	MDDI_Data0+/-
	MDDI_Data1+/-	MDDI_Data1+/-	MDDI_Data1+/-
		MDDI_Data2+/-	MDDI_Data2+/-
		MDDI_Data3+/-	MDDI_Data3+/-
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+/-

还应注意的，用于从主机传送的 MDDI 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某种电源控制、同步或者互连。

通常用于实现上述结构和操作的电缆标称为大约 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 30 AWG 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，屏蔽层与显示器（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：MDDI_Gnd 与 MDDI_Pwr；MDDI_Stb+ 与 MDDI_Stb-；MDDI_Data0+ 与 MDDI_Data0-；MDDI_Data1+ 与 MDDI_Data1-；等等。

D. 数据类型和速率

为了实现用于用户感受和应用的有用接口，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，只要总的的数据速率低于需要的最大 MDDI 链路速率就行，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。这些可以包括但不局限于下面表 II 和 III 中列出的那些项。

表 II

从主机向客户机传送		
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
异步控制	最小值	<< 1.0 Mbps

表 III

从客户机向主机传送		
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

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，因此它可以支持各种信息“类型”的传送，所述信息类型包括用户定义的数据。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；处于低速率以便节省功率并且降低实现成本的静态位图；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。

本发明的实施例推进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）；使用个人计算机进行有限个人观看（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）；使用视频电话形式的设备在因特网上“冲浪”（双向低速率视频和音频），使用照相机进行静态数字图像拍摄或者使用摄录一体机来获取数字视频图像；使用装有投影仪的电话或者 PDA 来呈现或者装配连接至视频监视器、键盘以及鼠标的台式装配站；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。

如下所述的移动式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。

MDD 接口信号把通称为公共帧（CF）的概念用于基本信号协议或者结构。使用公共帧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用于特殊应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。

表 IV 中示出了例如头戴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公共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。

表 IV

公共帧速率 (CFR) = 300 Hz							
	X	Y	位	帧速率	通道	速率 (Mbps)	字节/CF
计算机游戏	720	480	24	30	1	248.832	27000
计算机图形	800	600	24	10	1	115.200	12500
视频	640	480	12	29.97 或者 30	1	221.184	92160
CD 音频	1	1	16	44100	2	1.4112	294
语音	1	1	8	8000	1	0.064	26-2/3

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公共帧的分数字节。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 CF 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。

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节奏。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。

如果人们假定公共帧速率是 300 Hz，那么每一 CF 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 147 16 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（有可能是头戴式的）之间发送异步数据包。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。

表 V 示出了对于卡拉 OK 实例而言，在公共帧内如何分配数据。将使用的总速率选择为约 225 Mbps。226 Mbps 的稍高速率允许每子帧传送又一 400 字节的数据，这样就能够偶尔进行控制和传送状态信息。

表 V

元素速率	字节/CF
640×480 像素, 30fps 的音乐视频	92160
640×120 像素, 1fps 的歌词文本	768
44,100sps, 立体声, 16 位的 CD 音频	588
8,000sps, 单声道, 8 位的语音	26.67
子帧报头	19
反向链路开销	26.67+2*9+20
总字节/CF	93626.33
总速率 (Mbps)	224.7032

III. (继续) 高速率数字数据接口系统体系结构

E. 链路层

使用 MDD 接口高速串行数据信号传送的数据包括时分复用数据包流，其中这些数据包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数据包，由此维持数据包流。因此，简单数据包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。

数据包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数据包，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。

特殊的报头数据包包含唯一的标识符，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捕获。

通常，显示全运动视频时，每一介质帧更新显示屏幕一次。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。

子帧允许系统能够周期性地传输高优先级数据包。这样做允许同时发生的同步流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）基本上共享公共通道。该接口使用相对少数的信号传送信息。该接口还允许存在显示技术专用动作，诸如 CRT 监视器的水平同步脉冲和消隐间隔（blanking intervals）。

F. 链路控制器

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，总的来说，这样做可以降低链路控制器的制造成本。

IV. 接口链路协议

A. 帧结构

图 6 中说明用于实现数据包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数据包的元素 (element)。继而，多个数据包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，每一子帧从特别预定数据包开始，该数据包称为子帧报头数据包 (SHP)。

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。

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有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数据包 (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。

可以在状态数据包中向主机传送定义位图属性和显示器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。

当在本子帧中没有 (更多) 将要传送的数据包时，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数据包。由于每一子帧从子帧报头数据包开始，所以前一子帧的末尾包含正好填充前一子帧的数据包 (很可能是填充符数据包)。在承载数据的数据包本身缺乏空间的情况下，填充符数据包很可能是子帧中最后的数据包，或者在紧邻的先前 (next previous) 子帧的结尾并且在子帧报头数据包之前。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数据包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包，主机就必须能够成功地完成帧内该大小的数据包传送，而不能引起数据欠载（under-run）情况。

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，或者周期性的时序初相（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，其中只有在有新的信息时，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数据包中把子帧长度设置为零来定义的。使用周期性模式时，当显示器与正向链路帧结构同步时，可以开始子帧数据包接收。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数据包之后开始接收。

B. 总体数据包结构

下面说明用于规定所述实施例实现的信令协议的数据包格式或者结构，应记住的是，所述接口是可扩展的，根据需要可以添加附加数据包结构。所述数据包就其在接口中的功能，即它们传送的是命令还是数据而言，被标记为或者分为不同的“数据包类型”。因此，对于用于操纵被传送的数据包和数据的给定数据包，每一数据包类型表示一个预定义的数据包结构。显而易见，所述数据包可以具有预选长度，或者根据各自功能具有可变或者动态可变的长度。所述数据包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。用于各种数据包的字节或者字节值被配置成多位（8 或 16 位）无符号整数。所采用的数据包概要以及它们的“类型”指定被按照类型顺序列出，并在表 VI-1 至 VI-4 中示出。

为了便于举例说明和理解，每一表均给出总体数据包结构内的通用“类型”数据包。这些分组不暗含或者表示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数据包。还指出了认为数据包传送有效的方向。

表 VI-1
链路控制数据包

数据包名称	数据包类型	正向有效	反向有效
子帧报头数据包	15359	X	
填充符数据包	0	x	x
反向链路封装数据包	65	x	
链路关闭数据包	69	x	
接口类型切换请求数据包	75	x	
接口类型确认数据包	76		x
执行类型切换数据包	77	x	
往返行程延迟测量数据包	82	x	
正向链路偏离校准数据包	83	x	

表 VI-2
基本媒体流数据包

数据包名称	数据包类型	正向有效	反向有效
视频流数据包	16	x	x
音频流数据包	32	x	x
反向流数据包	1-15,18-31,33-35	x	x
用户定义的流数据包	56-63	x	x
色彩映射数据包	64	x	x
正向音频通道启用数据包	78	x	
反向音频采样速率数据包	79	x	
透明色彩启用数据包	81	x	

表 VI-3
显示器状态和控制数据包

数据包名称	数据包类型	正向有效	反向有效
显示器能力数据包	66		X
键盘数据数据包	67	X	X
指示设备数据数据包	68	X	X
显示器请求和状态数据包	70	X	X
数字内容保护开销数据包	80	X	X
请求 VCP 特征数据包	128	X	
VCP 特征应答数据包	129	X	X
设置 VCP 特征数据包	130	X	
请求有效参数数据包	131	X	
有效参数应答数据包	132		X
请求专用状态数据包	138	X	
有效状态应答列表数据包	139		X
数据包处理延迟参数数据包	140		X
个人显示器能力数据包	141		X
显示器错误报告数据包	142		X
可缩放视频流能力数据包	143		X
显示器标识数据包	144		X
备用显示器能力数据包	145		X
寄存器访问数据包	146		X

表 VI-4
高级图形和显示数据包

数据包名称	数据包类型	正向有效	反向有效
位块传送数据包	71	x	
位图区域填充数据包	72	x	
位图图案填充数据包	73	x	
读取帧缓冲器数据包	74	x	
阿尔法-光标图像能力数据包	133		x
阿尔法-光标透明度映射数据包	134	x	
阿尔法-光标图像偏移数据包	135	x	
阿尔法-光标视频流数据包	17	x	
可缩放视频流能力数据包	143		x
可缩放视频流设置数据包	136	x	
可缩放视频流确认数据包	137		x
可缩放视频流数据包	18	x	

根据本文中的其它讨论可以清楚了解的是，虽然反向封装数据包、显示器能力数据包以及显示器请求和状态数据包也被认为是非常重要的，甚至是外部模式操作所必需的，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数据包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。

数据包具有公共基本结构或者最少字段的总体集合，所述最少字段包括数据包长度字段、数据包类型字段、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，数据包长度字段包含具有多位或者字节值形式的信息，用于指定数据包的总位数，或者指定数据包长度字段和 CRC 字段之间的长度。在一个实施例中，所述数据包长度字段包含 16 位或者 2 字节宽的无符号整数，它说明数据包长

度。数据包类型字段是又一多位字段，它指明数据包内包含的信息的类型。在一个示例性的实施例中，该数据包类型字段是 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。

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数据包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，为每个数据包类型具体定义了数据格式，并且该数据包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，每一数据包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、数据包类型和数据包长度字段上计算出来的 16 位循环冗余校验的结果，该结果用来确认数据包中信息的完整性。换言之，该结果是在除 CRC 字段本身之外的整个数据包上计算出来的。客户机通常保存检测到的 CRC 错误的总计数，并且在显示器请求和状态数据包（参见下文）中将此计数报告给主机。

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。这样做允许数据包结构易于嵌入主机和客户机的主存储器空间或者与之相关联，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。

在传送所述数据包期间，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。每一数据包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 I 型、II 型、III 型或者 IV 型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。

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，尽管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。

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对应位图的宽度。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。位图和显示窗之间的关系示出在。

C. 数据包定义

1. 子帧报头数据包

子帧报头数据包是每一子帧的第一个数据包，并且具有如图 8 所示的基本结构。子帧报头数据包用于主机-客户机同步，每一主机应该能够生成这个数据包，同时每一客户机应该能够接收并且解释这个数据包。正如可以在图 8 中看到的那样，通常将这种类型的数据包按顺序构造为具有数据包长度、数据包类型、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，通常将这种类型的数据包标识为类型 15359（十六进制 0x3bff）数据包，并且使用 20 字节的预选固定长度，不包括数据包长度字段。

所述数据包类型字段和唯一字字段均使用 2 字节值（16 位无符号整数）。这种将两个字段的 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 位作为数据包类型首先传输,然后传输最高 16 位。

保留 1 字段包含 2 字节的保留空间以备将来之用,并且在这里一般配置成将所有位置零。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,以表明所述主机能够访问多个客户机设备。保留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。

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,以便表明在把链路关闭为空闲状态之前,主机将只传输一个子帧。从一个子帧转变到下一子帧时,这个字段中的值可以“在工作过程中”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。如果子帧报头数据包的 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数据包的子帧长度来估计当前子帧的长度。

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。子帧计数字段包含 2 字节,说明表明从介质帧开始所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。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。

介质帧计数字段包含 4 字节(32 位无符号整数),用于说明表明从传送的当前介质项或者数据开始,已经传输的介质帧的数目的序号。介质项目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。

2. 填充符数据包

填充符数据包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数据包。填充符数据包最好具有最小长度，以便当需要时，在发送其它数据包的过程中具有最大的灵活性。在子帧或者反向链路封装数据包（参见下文）的结尾，链路控制器设置填充符数据包的大小以便填充剩余空间，从而维护数据包完整性。当主机或者客户机没有信息要发送或交换时，所述填充符数据包对于维护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数据包，以便有效使用所述接口。

图9中示出了填充符数据包的格式和内容。如图9所示，将这种类型的数据包构造为具有数据包长度、数据包类型、填充符字节和CRC字段。在一个实施例中，通常将这种类型的数据包标识为类型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位值，以便允许填充符数据包具有所期望的长度。最小填充符数据包在这个字段中不包含字节。也就是说，所述数据包只包括数据包长度、数据包类型和CRC，并且在一个实施例中，使用预选6字节的固定长度或者数据包长度值4。为数据包中的所有字节确定所述CRC值，其中包括数据包长度，但是在其它数据包类型中，该数据包长度是被排除的。

3. 视频流数据包

视频流数据包通常携带视频数据来更新显示器设备的矩形区域。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，其受系统资源的限制，这是因为视频流数据包内包含显示流所需要的所有上下文。图10中示出了视频流数据包（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，将这种类型的数据包构造为具有数据包长度（2字节）、数据包类型、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。通常将这种类型的数据包标识为类

型 16，这在 2 字节类型字段中表明。在一个实施例中，客户机使用显示器能力数据包的 RGB、单色和 Y Cr Cb 能力字段来表明接收视频流数据包的能力。

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通信。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，此时，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。

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数据包。还可能单个视频流数据包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 子帧。如果每一帧中有 480 像素行，那么每一子帧中的每一视频流数据包将包含 48 行像素。在其它情况中，视频流数据包可能不包含整数像素行。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正确的。每一视频流数据包通常必须包含整数个像素，即便它未必包含整数行像素。如果每个像素均大于一个字节，或者如果它们具有如图 12 所示的数据包格式，那么这是非常重要的。

图 11a-11d 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格式和内容。在图 11a-11d 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，用于说明当前数据包的当前流中像素数据中每一像素的格式。不同的视频流数据包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）可以在工作过程中改变其数据格式。像素数据格式应该符合至少一个对于客户机有效的格式，如在显示器能力数据包中所定义的那样。只有在当前数据包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数据包的像素格式。

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，那么所述视频数据包括单色像素阵列，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‘001’时，如图 11b 所示，那么所述视频数据包括一个彩色像素阵列，其中每一个象素都通过色彩映射（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‘010’时，如图 11c 所示，那么所述视频数据包括彩色像素阵列，其中红色的每一像素位数由位 11 至 8 定义，绿色的每一像素位数由 7 至 4 位定义，并且蓝色的每一像素位数由位 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、绿色和蓝色的位数和。

然而，当位[15:13]换成等于‘011’时，如图 11d 所示，那么视频数据包括具有 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，其中亮度（Y）的每一像素位数由位 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，这个数据包的像素数据部分中的视频采样被组织为： $C_{bn}, Y_n, C_{rn}, Y_{n+1}, C_{bn+2}, Y_{n+2}, C_{rn+2}, Y_{n+3}, \dots$ ，其中 C_{bn} 和 C_{rn} 与 Y_n 和 Y_{n+1} 相关联，并且 C_{bn+2} 和 C_{rn+2} 与 Y_{n+2} 和 Y_{n+3} 相关联等等。

Y_n, Y_{n+1}, Y_{n+2} 和 Y_{n+3} 是单个行中从左至右的四个连续像素的亮度值。通常将色彩分量的顺序选为与微软公司在其软件中使用的 UYVY FOURCC 格式相同的格式，当然本发明不局限于这种格式。如果在视频流数据包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。数据包中的像素

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数据包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，它可以包含奇数或者偶数个像素。

当位[15:13]换成等于‘100’时，那么视频数据包括一个间隔（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。像素图案由位 5 和 4 定义，如“错误!未发现参考源（间隔（Bayer））”所示那样。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，如“错误!未发现引用源”所示那样。位 11 至 9 应该被设置为零。

对于图中示出的所有四个格式来说，指定为“P”的位 12 说明像素数据采样是打包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’表明像素数据中每一像素内的每一像素和每一颜色相对于未留有未使用位的像素内的先前像素或色彩而打包。

用于特定显示窗的介质帧的第一个视频流数据包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。在介质帧的此第一个数据包中，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数据包中，通常将 X 和 Y 起始值设置为屏幕窗口中特定像素位置，该像素位置为在先前子帧中传输的视频流数据包中发送的最后一个像素以后的像素位置。

4. 音频流数据包

音频流数据包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。客户机使用显示器能力数据包的音频通道能力和音频采样速率字段来表明接收音频流数据包的能力。图 13 中说明音

频流数据包的格式。

如图 13 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、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，通常将这种类型的数据包标识为类型 32 数据包。

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。

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 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 中说明已打包音频采样和字节对准音频采样之间的差异。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 音频采样相对于先前音频采样打包。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。

5. 保留的流数据包

在一个实施例中，根据遇到的各种应用的要求，数据包类型 1 至 15、18 至 31 和 33 至 55 为待定义的保留流数据包，以供将来版本或者数据包协议变化之用。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。

6. 用户定义的流数据包

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 MDDI 链路使用。这些被称为用户定义的流数据包。这种数据包可以用于任何目的，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

采用这种数据包。针对这些数据包类型的流参数和数据的具体定义留给实现这种数据包类型或者探寻其用途的具体设备制造商。用户定义的流数据包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 中说明用于一个实施例的用户定义的流数据包的格式。如图 15 所示，将这种类型的数据包构造为具有数据包长度（2 字节）、数据包类型、bClient ID 号、流参数、参数 CRC、流数据和流数据 CRC 字段。

7. 色彩映射数据包

色彩映射数据包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数据包能够传输的数据量大的色彩映射。这样的话，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数据包，每个色彩映射数据包均具有色彩映射的不同子集。图 16 中说明一个实施例中的色彩映射数据包的格式。如图 16 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、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，通常将这种类型的数据包标识为类型 64 数据包（视频数据格式和色彩映射数据包），如数据包类型字段（2 字节）中所说明的那样。客户机使用显示器能力数据包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数据包的能力。

8. 反向链路封装数据包

在一个示例性的实施例中，使用反向链路封装数据包沿反向传送数据。发送一个正向链路数据包，并且大约在这个数据包的中部改变 MDDI 链路操作或者让它转向（传送方向），从而能够沿反向发送数据包。图 17 说明一个实施例中的反向链路封装数据包的格式。如图 17 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、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、反向数据数据包、全零 2、转向 2 以及驱动器重新启用字段。通常将这种类型的数据包

标识为类型 65 数据包。对于外部模式来说，每一主机必须能够生成这个数据包并且接收数据，每一客户机必须能够接收并且向主机发送数据。这个数据包的实现方式对于内部模式来说是可选的。

当发送反向链路封装数据包时 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。主机的行为好像它正在为反向链路封装数据包的转向和反向数据数据包部分的每位传输零。所述主机在两个换向期间并且在分配给反向数据数据包的期间在每一位的边界处发出 MDDI 选通信号。（这与传输全零数据的行为相同。）所述主机在由转向 1 指定的时间段期间禁用其 MDDI 数据信号线路驱动器，并且客户机在由转向 2 字段指定的时间段之后的，驱动器重新启用字段期间重新启用其线路驱动器。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数据包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。客户机使用数据包长度和转向长度参数来了解其可以用来向主机发送数据包的时间长度。当没有数据要发送到主机时，客户机可以发送填充符数据包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）的数据包，并且在当前反向链路封装数据包期间，主机不再接收来自客户机的任何数据包。

在全零 1 字段期间，所述主机把 MDDI_Data 信号驱动为逻辑零电平，并且在转向 2 字段开始以前，也就是在全零 2 字段期间，客户机在至少一个反向链路时钟周期内把 MDDI 数据线驱动为逻辑零电平。这样做使数据线在转向 1 和转向 2 字段的时间段期间保持在确定的状态中。如果客户机没有更多数据包要发送，那么它甚至可以在把数据线驱动为逻辑零电平之后禁用它们，这是因为在反向数据数据包字段的剩余时间，或者在约 16 个或更多正向链路字节的期间，休眠偏置电阻（其它地方讨论）使数据线保持在逻辑零电平。

在一个实施例中，可以利用显示器请求和状态数据包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数据包中需要把数据

送回到主机的字节数。主机试图通过在反向链路封装数据包中分配至少那么多的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数据包。显示器几乎可以在任何时候发送显示器请求和状态数据包，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。

9. 显示器能力数据包

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，显示器把显示器能力数据包发送给主机。当由主机使用反向链路封装数据包中的反向链路标志来请求时，应该考虑这种数据包的传输。显示器能力数据包用来通知主机显示器的能力。对于外部模式来说，每一主机必须能够接收这个数据包，而每一显示器必须能够发送这个数据包以便完全地利用这一接口和协议。这个数据包的实现方式对于内部模式而言是可选的，这是由于制造的时候或者装配到某些类型的单个部件或单元时，显示器的能力应该已经被明确定义并且为主机所知。

图 18 中说明一个实施例中显示器能力数据包的格式。如图 18 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、协议版本、最小协议版本、位图宽度、位图高度、单色能力、色彩映射能力、RGB 能力、Y Cr Cb 能力、显示器特征能力、数据速率能力、帧速率能力、音频缓冲器深度、音频流能力、音频速率能力、最小子帧速率和 CRC 字段。在一个示例性的实施例中，通常将这种类型的数据包标识为类型 66 数据包。

10. 键盘数据数据包

键盘数据数据包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。键盘数据数据包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。这个数据包还可以

在正向链路上使用以便向键盘发送数据。客户机使用显示器能力数据包中的键盘数据字段来表明发送和接收键盘数据数据包的能力。

图 19 中示出了键盘数据数据包的格式，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，将这种类型的数据包构造为具有数据包长度、数据包类型、bClient ID、键盘数据格式、键盘数据和 CRC 字段。在这里，通常将这种类型的数据包标识为类型 67 数据包。

所述 bClient ID 是一个保留字段，如前所述，并且对数据包的所有字节执行 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 应该与显示器能力数据包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。

11. 指示设备数据数据包

指示设备数据数据包用来从无线鼠标或者其它来自显示器的指示设备的位置信息发送给主机。还可以使用这个数据包把数据发送给正向链路上的指示设备。图 20 中示出了指示设备数据数据包的示例性格式，并且该数据包格式包含来自或者用于指示设备的可变数目字节的信息。如图 20 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、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 字节的类型字段中，将这种类型的数据包标识为类型 68 数据包。

12. 链路关闭数据包

链路关闭数据包从主机发送到客户机显示器，以便表明 MDDI 数据和选通脉冲将被关闭，并且进入低功耗“休眠”状态。在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，这个数据包对关闭链路并且节省功率十分有用。当主机再次发送数据包时，恢复正常操作。休眠之后发送的第一数据包是子帧报头数据包。图 21 中示出了显示状态数据包的格

式。如图 21 所示，将这种类型的数据包构造为具有数据包长度、数据包类型和 CRC 字段。在一个实施例中，通常在 1 字节的类型字段中将这种类型的数据包标识为类型 69 数据包，并且使用 3 字节的预选固定长度。

在低功率休眠状态中，所述 MDDI_Data 驱动器被禁用为高阻状态，并且使用可以由显示器过度驱动的高阻偏置网络把 MDDI_Data 信号拉至逻辑零状态。在休眠状态中，由所述接口使用的选通信号被设置为逻辑零电平以便最小化功耗。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“苏醒”，这是本发明的关键性进步和优势。

13. 显示器请求和状态数据包

主机需要来自显示器的少量信息，以便使其可以按照总的来说最优的方式来配置主机至客户机的链路。建议显示器向主机每一子帧发送一个显示状态数据包。显示器应该把这个数据包作为反向链路封装数据包中的第一个数据包来发送，以便确保将该数据包可靠地发送至主机。图 22 中示出了显示状态数据包的格式。如图 22 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、反向链路请求、CRC 错误计数和 CRC 字段。在 1 字节的类型字段中，通常将这种类型的数据包标识为类型 70 数据包，并且使用 8 字节的预选固定长度。

所述反向链路请求字段可以用来通知主机所述显示器在反向链路封装数据包中需要的字节数目，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数据包中分配至少这个数目的字节来允许所述请求。主机可以在子帧中可发送一个以上的反向链路封装数据包以便接纳数据。客户机可以在任何时候发送显示器请求和状态数据包，并且所述主机可将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具体实例的其它细节。

14. 位块传送数据包

位块传送数据包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显示器能力数据包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 中示出了位块传送数据包的格式。如图 23 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、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。通常将这种类型的数据包标识为类型 71 数据包，并且使用 15 字节的预选固定长度。

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。

15. 位图区域填充数据包

位图区域填充数据包提供了很容易地把显示区域初始化为单色的方法。具有这一能力的显示器将在显示器能力数据包的显示器特征能力指示符字段的位 1 中报告所述能力。图 24 中示出了位图区域填充数据包的格式。如图 24 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、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 字节的类型字段中，通常将这种类型的数据包标识为类型 72 数据包，并且使用 17 字节的预选固定长度。

16. 位图图案填充数据包

位图图案填充数据包提供了很容易地把显示区域初始化为预选图案的方法。具有这一能力的显示器将在显示器能力数据包的显示器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。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。根据需要将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。

图 25 中示出了位图图案填充数据包的格式。如图 25 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、左上 X 值、左上 Y 值、窗口宽度、窗口高度、图案宽度、图案高度、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 字节类型字段中，通常将这种类型的数据包标识为类型 73 数据包。

17. 通信链路数据通道数据包

通信链路数据通道数据包提供了用于使如 PDA 的这种具有高级计算能力的显示器与如便携电话或者无线数据端口设备的这种无线收发器通信的方法。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，其中这种数据包在设备的操作系统的通信链路层传送数据。例如，如果网络浏览器、电子邮件客户机程序或者整个 PDA 被嵌入移动式显示器中，就可以使用这个数据包。具有这一能力的显示器将在显示器能力数据包的显示器特征能力指示符字段的位 3 中报告所述能力。

图 26 中示出了通信链路数据通道数据包的格式。如图 26 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、参数 CRC、通信链路数据和通信数据 CRC 字段。在类型字段中，通常将这种类型的数据包标识为类型 74 数据包。

18. 接口类型切换请求数据包

接口类型切换请求数据包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 I 型（串行）、II 型（2 位并行）、III 型（4 位并行）或者 IV 型（8 位并行）模式。在主机请求特定模式以前，它应该通过检查显示器能力数据包的显示器特征能力指示符字段的位 6 和 7，来确认显示器能够工作在所需要的模式下。图 27 中示出了接口类型切换请求数据包的格式。如图 27 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、接口类型和 CRC 字段。通常将这种类型的数据包标识为类型 75 数据包，并且使用 4 字节的预选固定长度。

19. 接口类型确认数据包

接口类型确认数据包由显示器发送以便确认接收到接口类型切换请求数据包。所请求的模式，即：I 型（串行）、II 型（2 位并行）、III 型（4 位并行）或者 IV 型（8 位并行）模式，作为这个数据包中的一个参数返回给主机。图 28 中示出了接口类型确认数据包的格式。如图 28 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、接口类型和 CRC 字段。通常将这种类型的数据包标识为类型 76 数据包，并且使用 4 字节的预选固定长度。

20. 执行类型切换数据包

执行类型切换数据包是用于使主机命令显示器切换至这个数据包中指定的模式的一种方法。这个模式与接口类型切换请求数据包和接口类型确认数据包先前请求并且确认的模式相同。在发送这个数据包之后，主机和显示器应该切换到达成一致的 mode。显示器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数据包的格式。如图 29 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、数据包类型和 CRC 字段。在 1 字节的类型字段中，通常将这种类型的数据包标识为类型 77 数据包，并且使用 4 字节的预选固定长度。

21. 正向音频通道启用数据包

这个数据包使得主机能够启用或者禁用显示器中的音频通道。这一能力十分有用，因为在主机没有音频输出时，显示器（客户机）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。显示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数据包的格式。如图 30 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、音频通道启用掩码和 CRC 字段。在 1 字节的类型字段中，通常将这种类型的数据包标识为类型 78 数据包，并且使用 4 字

节的预选固定长度。

22. 反向音频采样速率数据包

这个数据包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。所述主机选择在显示器能力数据包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，那么显示器不会向主机发送音频流。主机可以通过把采样速率设置为 255 来禁用反向链路音频流。当显示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数据包的格式。如图 31 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、音频采样速率和 CRC 字段。通常将这种类型的数据包标识为类型 79 数据包，并且使用 4 字节的预选固定长度。

23. 数字内容保护开销数据包

这个数据包使得主机能够和显示器交换与正使用的数字内容保护方法相关的消息。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标记保留了空间。所使用的方法由这个数据包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数据包的格式。如图 32 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、内容保护类型、内容保护开销消息和 CRC 字段。通常将这种类型的数据包标识为类型 80 数据包。

24. 透明色彩启用数据包

透明色彩启用数据包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。具有这一能力的显示器在显示器能力数据包的显示器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数据包的格式。如图 33 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、透明色彩启

用、数据格式描述符、透明像素值和 CRC 字段。在 1 字节的类型字段中，通常将这种类型的数据包标识为类型 81 数据包，并且使用 10 字节的预选固定长度。

25. 往返行程延迟测量数据包

往返行程延迟测量数据包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。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数据包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，这种数据包是最有用的。所述 MDDI_Stb 信号的作用是如同在如下字段发送全零数据：全零、两个保护时间和测量周期。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显示器里将 MDDI_Stb 用作周期性的时钟。

图 34 中示出了的往返行程延迟测量数据包的格式。如图 34 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、参数 CRC、全零、保护时间 1、测量周期、保护时间 2 和驱动器重新启用字段。通常将这种类型的数据包标识为类型 82 数据包，并且使用 533 位的预选固定长度。

图 35 中说明在往返行程延迟测量数据包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数据包，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。在所述数据包到达客户机显示器设备或者处理电路以前，延迟 3502 发生。当显示器收到所述数据包时，在通过显示器确定的测量周期开始时，显示器以尽可能准确到实际时间的时刻传输 0xff、0xff、0x0 模式（pattern）。从主机的角度看，显示器开始发送这一序列的实际时间距测量周期的开始有延迟。这一延迟量基本上是所述数据包用来通过线路驱动器和接收器以及互联子系统进行传送所花费的时间。对于把该模式从显示器传送到主机来说，经历了类似的延迟量 3504。

为了准确地确定信号往返于客户机的往返行程延迟时间，主机对测量周期开始之后出现的位时间周期数计数，直到检测到 0xff、0xff、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。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。

发送 0xff、0xff、0x0 模式的最后位之后，显示器基本上立即禁用其线路驱动器。保护时间 2 允许显示器的线路驱动器在主机传输下一数据包的数据包长度之前的时间完全进入高阻状态。休眠上拉和下拉电阻（参见图 42）确保 MDDI_Data 信号在主机和显示器中禁用线路驱动器的这些间隔内保持在有效的低电平上。

26. 正向链路偏离校准数据包

正向链路偏离校准数据包允许客户机或者显示器针对 MDDI_Data 信号相对于所述 MDDI_Stb 信号的传播延迟差异来校准自己。在没有延迟偏离补偿的情况下，最大数据速率通常受限于针对这些延迟中潜在的最坏情况下的变化量的值。通常只有当正向链路数据速率被配置成 50 Mbps 左右或者更低的速率时，才发送这个数据包。在发送这个数据包以便校准显示器之后，数据速率可以提高至 50 Mbps 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数据包之前，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。

图 56 中示出了正向链路偏离校准数据包的格式。如图 56 所示，将这种类型的数据包构造为具有数据包长度（2 字节）、数据包类型、参数 CRC、校准数据序列和 CRC 字段。在类型字段中，通常将这种类型的数据包标识为类型 82 数据包，并且具有 515 的预选长度。

虚拟控制面板

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。

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，以便在数据包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改进是唯一的。

27. 请求 VCP 特征数据包

请求 VCP 特征数据包提供了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 特征应答数据包中适当的信息来响应 VCP 数据包。在一个实施例中，客户机使用显示器能力数据包的显示器特征能力指示符字段的位 20 来表明支持请求 VCP 特征数据包的能力。

图 69 中示出了一个实施例中请求 VCP 特征数据包的格式。如图 69 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、hClient ID、MCCS VCP 代码和 CRC 字段。在一个实施例中，通常将这种类型的数据包标识为类型 128，这是在 2 字节的类型字段中表明的。数据包长度说明数据包中不包括数据包长度字段的字节总数，并且对于这种类型的数据包而言，该数据包长度通常被固定在 8 字节的长度。

所述 hClient ID 字段包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留下来以供将来使用并且通常设置为零。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 特征应答数据包以单个项返回，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。65535 (0xffff) 的 MCCS VCP 代码请求一个具有 VCP 特征应

答列表的 VCP 特征应答数据包, 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。

28. VCP 特征应答数据包

VCP 特征应答数据包提供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。通常, 客户机响应于请求 VCP 特征数据包而发送所述 VCP 特征应答数据包。这个数据包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 特征应答数据包。在一个实施例中, 所述客户机使用显示器能力数据包的显示器特征能力指示符字段的位 20 来表明支持 VCP 特征应答数据包的能力。

图 70 中示出了一个实施例中的 VCP 特征应答数据包的格式。如图 70 所示, 将这种类型的数据包构造为具有数据包长度、数据包类型、cClient ID、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, 通常将这种类型的数据包标识为类型 129, 如在 2 字节的类型字段中表明的那样。

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 VESA MCCS 规范的版本。

2 字节的应答序号字段包含说明由客户机返回的 VCP 特征应答数据包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 特征数据包, 返回一个或多个 VCP 特征应答数据包。所述客户机可以经由多个 VCP 特征应答数据包扩展特征应答列表。在这种情况下, 客户机向每一连续数据包分配序号, 并且响应单个请求 VCP 特征数据包发送的 VCP 特征应答数据包的序号从零开始并且每次加 1。最后 VCP 特征应答数据包中的最后 VCP

特征列表项应该包含等于 0xffff 的 MCCS VCP 控制代码值，以便确定所述数据包是最后一个并且包含返回的数据包组的最高序号。如果响应于请求 VCP 特征数据包只发送了一个 VCP 特征应答数据包，那么该单个数据包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。

列表中特征数目字段包含 2 字节，用于说明该数据包的 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。

如图 71 所示，每一 VCP 特征应答列表项的长度严格地是 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。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，所述错误代码与对所指定的 MCCS VCP 控制的信息的请求相关。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在客户机中实现。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。

4 字节的最大值字段包含 32 位无符号整数，用于说明可以设置指定的 MCCS 控制的最大可能值。如果所请求的控制没有在客户机中实现，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），就把所述值转换为 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已指定的 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在客户机中实现，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 VESA MCCS 规范而言，返回值在长度上小于 32 位（4 字节），就把所述值转换为 32 位整数，将最高有效（未使用）字节设置为零。

29. 设置 VCP 特征数据包

设置 VCP 特征数据包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，所述客户机使用显示器能力数据包的显示器特征能力指示符字段的位 20 来表明支持设置 VCP 特征数据包的能力。

图 72 中示出了一个实施例中设置 VCP 特征数据包的格式。如图 72 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、hClient ID、MCCS VCP 代码、列表中值的数量、控制值列表和 CRC 字段。通常将这种类型的数据包标识为类型 130，如 2 字节的类型字段中所示那样，且该类型的数据包不包括数据包长度字段是 20 字节长。

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将包含一项，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 字段包含数据包中所有字节的 16 位 CRC，其中包括数据包长度在内。

30. 请求有效参数数据包

请求有效参数数据包用作请求该客户机返回有效参数应答数据包的手段或者机制，该有效参数应答数据包包含由所指定的不连续（NC）或者表（T）控制支持的一系列参数。这个数据包只应当指定不连续的控制或者涉及客户机中的表的控制，而不指定值为 65535

(0xffff) 的 MCCS VCP 代码值以便指定所有控制。如果指定了不支持或者无效的 MCCS VCP 代码，就在有效参数应答数据包中返回适当的错误值。在一个实施例中，所述客户机使用显示器能力数据包的显示器特征能力指示符字段的位 20 来表明支持请求有效参数数据包的能力。

图 73 中示出了一个实施例中的请求有效参数数据包的格式。如图 73 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、hClient ID、MCCS VCP 代码和 CRC 字段。在一个实施例中，通常将这种类型的数据包标识为类型 131，如在 2 字节的类型字段中表明的那样。

如 2 字节数据包长度字段中表明的数据包长度通常被设置为具有数据包中的字节总数，但不包括数据包长度字段 8。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。

31. 有效参数应答数据包

有效参数应答数据包是响应于请求有效参数数据包而发送的。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数据包，就可以由客户机发送具有顺序的应答序号的多个数据包。在一个实施例中，客户机使用显示器能力数据包的显示器特征能力指示符字段的位 20 来表明支持有效参数应答数据包的能力。

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 特征数据包，诸如读/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数据包；然后客户机返回包含表数据的一个或多个有效参数应答数据包。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。

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，这个数据包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。

图 74 中示出了用于一个实施例的有效参数应答数据包的格式。如图 74 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、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，通常将这种类型的数据包标识为类型 132，如在 2 字节的类型字段中表明的那样。

将所述 cClient ID 字段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 代码数据包则包含用于说明由这个数据包描述的不连续 MCCS VCP 控制代码参数的值。如果请求有效参数数据包指定了无效的 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。

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，所述请求为对关于所指定的 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数据包，该数据包具有最高应答序号。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数据包。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。

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数据包的序号。所述客户机，响应于一个请求有效参数数据包，返回一个或多个有效参数应答数据包。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数据包中。在这后一情况下，客户机将向每一连续的数据包分配序号，并且在除序列中最后一个数据包之外的所有数据包中，把响应代码设置为 1。序列中最后的有效参数应答数据包将具有最高的应答序号，并且所述响应代码包含的值为 0。

所述 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 字段包含数据包中所有字节的 16 位 CRC，其中包括数据包长度在内。

阿尔法光标图像

用于经由通信链路传递数据的 MDD 接口和关联的本发明的协议及机制提供了对互相重叠并且具有不同程度透明度的多重图像平面的支持。硬件光标可以使用具有可变 X-Y 偏移的重叠图像来实现。下面提供对阿尔法光标功能和相关协议支持的概述。在响应于请求专用状态数据包而发送的阿尔法光标图像能力数据包中定义了支持阿尔法光标图像数据包的能力。

32. 阿尔法光标图像能力数据包

阿尔法光标图像能力数据包用来定义客户机中的阿尔法光标图像和相关联的透明度映射的特性。在一个实施例中，客户机使用有效状态应答列表数据包的有效参数应答列表中参数值 133 来表明支持阿尔法光标图像能力数据包的能力。对于一个实施例而言，把数据包长度字段中指定的数据包长度设置为固定值 20，其不包括所述数据

包长度字段。

图 75 中示出一个实施例中的阿尔法光标图像能力数据包的格式。如图 75 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、cClient ID、阿尔法光标标识符、阿尔法光标位图宽度、阿尔法光标位图高度、RGB 能力、单色能力、保留 1、Y Cr Cb 能力、透明度映射分辨率、能力位（Capability Bits）和 CRC 字段。通常保留所述 cClient ID 字段以供将来客户机 ID 使用，并且目前将它设置为零。

阿尔法光标标识符字段（2 字节）包含用于识别具体阿尔法光标平面的值。如果客户机支持 n 个阿尔法光标图像平面，那么阿尔法光标标识符具有 0 至 $n-1$ 的有效范围。在一个实施例中，值 n 是由显示器能力数据包的阿尔法光标图像平面字段指定的。所述客户机为每个阿尔法光标图像平面返回唯一的阿尔法光标图像能力数据包。

2 字节的阿尔法光标位图宽度字段值说明以像素个数表示的阿尔法光标位图图像的宽度，而 2 字节的阿尔法光标位图高度字段值则说明以像素个数表示的阿尔法光标位图图像的高度。

RGB 能力字段 - 2 字节，它包含一个 16 位无符号整数，用于说明可以按照 RGB 格式显示的分辩率的位数。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值，在一个实施例中，它是这样实现的：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 15 至 12 被保留下来以便将来用于呈现 RGB 能力信息，而现在被设置为零。

1 字节的单色能力字段用来指定可以按照单色格式显示的分辩率的位数。如果客户机无法使用单色格式，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。这四位能够指定包括每个像素由 1 至 15 位构成。如果所述值是零，那么客户机不支持所述单色格式。

1 字节的保留 1 字段包含通常被保留以备将来之用的值，并且因

此，这个字段中的所有位都被设置为零。这将令后续 2 字节字段与 16 位字地址 (address) 对准，并且令 4 字节字段与 32 位字地址对准。

2 字节的 Y Cb Cr 能力字段包含用于说明可以按照 Y Cb Cr 格式显示的分辩率的位数的值或者信息。如果客户机无法使用 Y Cr Cb 格式，那么这一值是零。通常，在一个实施例中，所述 Y Cb Cr 能力字包括三个独立的值：位 3 至 0 定义用于说明 Cr 采样的最大位数；位 7 至 4 定义用于说明 Cb 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；并且位 15 至 12 被保留以供将来用于呈现 Y Cb Cr 能力信息或者值，但是目前被设置为零。

1 字节的透明度映射分辩率字段包含用于说明阿尔法光标图像透明度映射的每一像素位置中的位数 (深度) 的值或者信息。这一值在从 1 至 8 的范围内。如果所述值是零，那么对于这一阿尔法光标图像缓冲器 (由阿尔法光标标识符字段指定的缓冲器) 来说，不支持透明度映射。

1 字节的能力位字段提供包含一组标志的值或者信息，所述标志说明与阿尔法光标图像缓冲器相关联的能力。在一个实施例中，所述标志是如此定义的：0 位用于选择将要置于数据包格式中的阿尔法光标视频流数据包中的像素数据。位 1 用于表明阿尔法光标透明度数据包中的透明度映射数据是数据包格式。图 76 中示出了字节对准并且打包的透明度映射数据的实例。位 2 用于表明阿尔法光标图像平面使用阿尔法光标图像偏移数据包能够支持图像偏移能力。位 3 用于表明阿尔法光标图像平面能够支持色彩映射数据格式。与用于主图像缓冲器和可缩放视频流使用的一样，把同样的色彩映射表用于阿尔法光标图像平面。所述色彩映射是使用其它地方描述的色彩映射数据包来配置的。

保留位 7 至 4 以供将来使用，并且因此通常将其设置为零值或逻辑电平低。

33. 阿尔法光标透明度映射数据包

阿尔法光标透明度映射数据包定义了指定的阿尔法光标图像平

面的图像透明度映射的内容。某些应用可能要求一个透明度映射，且该透明度映射的数据量大于能够在单个数据包中传输的数据量。这样的话，通过使用如下所述的透明度映射 X 和 Y 起始字段，可以发送多个阿尔法光标透明度映射数据包，每个均具有透明度映射的不同子集。这些字段按照类似于视频流数据包的 X 起始和 Y 起始字段的方式进行工作。在一个实施例中，客户机使用每个具体阿尔法光标平面的阿尔法光标图像能力数据包的透明度映射分辨率字段来表明其支持阿尔法光标透明度映射数据包的能力，其中阿尔法光标平面是由阿尔法光标图像能力数据包的阿尔法光标标识符字段指定的。所述数据包长度和客户机 ID 字段如上文就其它数据包所述那样使用。在一个实施例中，数据包类型字段中的值 134 用来把数据包标识为阿尔法光标透明度映射数据包。

图 76 中示出了用于一个实施例的阿尔法光标透明度映射数据包的格式。如图 76 所示，将这种类型的数据包构造为具有数据包长度、数据包类型、hClient ID、阿尔法光标标识符、透明度映射 X 起始、透明度映射 Y 起始、透明度映射分辨率、保留 1、参数 CRC、透明度映射介质和透明度映射数据 CRC 字段。

2 字节的阿尔法光标标识符字段具有用于标识具体阿尔法光标平面的值。如果客户机支持 n 个阿尔法光标图像平面，那么阿尔法光标标识符具有 0 至 $n-1$ 的有效范围。

2 字节的透明度映射 X 和 Y 起始字段均指定绝对 X 和 Y 坐标，其中所述点（透明度映射 X 起始，透明度映射 Y 起始）是下面的透明度映射数据字段中的第一像素。

所述透明度映射分辨率字段（1 字节）包含用于说明透明度映射的分辨率以及所述数据是否被打包的值。在这个字段的一个实施例中，位 3 至 0 定义所有透明度映射表项中存在的分辨率的位数。有效的值说明宽度是从 1 至 8 位。值 0 和 9 至 15 被认为是无效的。这一值应该与客户机在阿尔法光标图像能力数据包中的透明度映射分辨率字段中返回的值匹配。位 6 至 4 被保留以供将来使用，并且因此此时通常设置为逻辑零。这一字节的位 7 说明透明度映射数据是打包的

还是具有字节对准形式。如果位 7 等于 ‘1’，那么透明度映射数据是打包的形式，而如果是 ‘0’，所述数据就是字节对准形式。在“错误！未发现参考源。”中示出了打包和字节对准的透明度映射数据的一个实例。这一位的值必须与阿尔法光标图像能力数据包的能力位字段的位 1 值匹配。

1 字节的保留 1 字段被保留以备将来之用，因此，这个字段中的所有位通常被设置为等于逻辑零电平。这个字段的一个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。

参数 CRC 字段包含从数据包长度到保留 1 字段的所有字节的 16 位 CRC。如果这一 CRC 未能通过校验，就丢弃整个数据包。

对于透明度映射数据字段来说，每一透明度映射位置在宽度上是 1 至 8 位。如果单个透明度映射无法刚好放进一个阿尔法和光标透明度映射数据包，那么整个透明度映射可以通过发送多个数据包来指定，每一数据包中具有不同透明度映射数据和透明度映射 X 和 Y 起始值。

2 字节的透明度映射数据 CRC 字段包含只有透明度映射数据的 16 位 CRC。如果这一 CRC 未能通过校验，那么透明度映射数据仍可被使用，不过 CRC 错误计数将增加。

34. 阿尔法光标图像偏移数据包

阿尔法光标图像偏移数据包指定光标距离主显示图像左上角的 X 和 Y 偏移。图 77 中说明阿尔法光标图像偏移数据包的格式。如图 77 所示，在一个实施例中，所述阿尔法光标图像偏移数据包被构造为具有数据包长度、数据包类型、hClient ID、阿尔法光标 X 偏移、阿尔法光标 Y 偏移和 CRC 字段。在一个实施例中，客户机使用每个具体阿尔法光标平面的阿尔法光标图像能力数据包中的能力字段的位 2 来表明其支持阿尔法光标图像偏移数据包的能力，其中该阿尔法光标平面是由阿尔法光标图像能力数据包的阿尔法光标标识符字段指定的。在一个实施例中，所述数据包长度被固定在 10，如 2 字

节的数据包长度字段所示。在一个实施例中，数据包类型 135 把该数据包标识为阿尔法光标图像偏移数据包。

2 字节的阿尔法光标 X 和 Y 偏移字段包含用于分别说明光标图像的像素的最左列和顶行距离主图像左侧和顶部的水平和垂直偏移的值。所述 2 字节的 hClient ID 包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留供将来使用并且设置为零。

35. 阿尔法光标视频流数据包

阿尔法光标视频流数据包承载视频数据来更新阿尔法光标图像平面的矩形区域。这一区域的大小可以象单个像素那么小，或者象整个显示器那么大。图 78 说明阿尔法光标视频流数据包的格式。如图 78 所示，在一个实施例中，阿尔法光标视频流数据包被构造为具有数据包长度、数据包类型、bClient ID、视频数据格式属性、X 左沿、Y 顶沿、X 右沿、Y 底沿、X 起始、Y 起始、像素计数、参数 Crc 像素数据和像素数据 CRC 字段。在一个实施例中，客户机通过使用每个具体阿尔法光标平面的阿尔法光标图像能力数据包来表明其支持阿尔法光标视频流数据包及其相关参数的能力，其中所述阿尔法光标平面由阿尔法光标图像能力数据包的阿尔法光标标识符字段指定，并且数据包类型字段中的值 17 表明或者把该数据包标识为阿尔法光标视频流数据包。所述 hClient ID 字段（2 字节）被保留以供将来用作客户机 ID，并且同时将其通常设置为零，正如所述领域将了解的那样。

2 字节的视频数据格式描述符字段包含用于说明当前数据包的当前流中像素数据内的每一像素的格式的信息或者值。像素数据格式必须符合阿尔法光标图像平面的至少一个有效格式，如在阿尔法光标图像能力数据包中所定义的那样。所述视频数据格式描述符字段包含只用于定义当前数据包的像素格式的值，并且不暗指在特定视频流的可使用期间 (lifetime) 将持续使用不变的格式。“错误！未发现引用源。”说明视频数据格式描述符是如何编码的。所述格式如下：

在一个实施例中，当位[15:13]是‘000’时，那么所述视频数据

包括单色像素阵列，其中每一像素的位数由视频数据格式描述符字的位 3 至 0 定义。然后位 11 至 4 被设置为零。当位[15:13]是‘001’时，那么所述视频数据包括彩色像素阵列，其均通过色彩映射（调色板）来指定颜色。视频数据格式描述符字的 5 至 0 位定义每一像素的位数，而位 11 至 6 被设置为零。当位[15:13]是‘010’时，那么所述视频数据包括具有原始 RGB 格式的彩色像素阵列，其中红色的每一像素位数由位 11 至 8 定义，绿色的每一像素位数由位 7 至 4 定义，蓝色的每一像素位数由位 3 至 0 定义。每一像素中位数总数是红色、绿色和蓝色的位数和。

当位[15:13]是‘011’时，那么视频数据包括具有亮度和色度信息的 4:2:2 的 YCbCr 格式的视频数据阵列。亮度（Y）的每像素位数是由位 11 至 8 定义的，Cb 分量的位数是由位 7 至 4 定义的，而 Cr 分量的位数是由位 3 至 0 定义的。Cb 和 Cr 分量是以 Y 分量一半的速率发送的。这个数据包的像素数据部分中的视频采样按照如下方式组织：Cb_n, Y_n, Cr_n, Y_{n+1}, Cb_{n+2}, Y_{n+2}, Cr_{n+2}, Y_{n+3}, ……，其中 Cb_n 和 Cr_n 与 Y_n 和 Y_{n+1} 相关联，并且 Cb_{n+2} 和 Cr_{n+2} 与 Y_{n+2} 和 Y_{n+3} 相关联等等。Y_n, Y_{n+1}, Y_{n+2} 和 Y_{n+3} 是单个行中从左至右的四个连续像素的亮度值。色彩分量的顺序与微软公司 UYVY FOURCC 格式相同。如果在由视频流数据包所定位的窗口的一行中存在有奇数个（X 右沿 - X 左沿 + 1）像素，那么对应于每一行中最后像素的 Cb 值后面将跟随有下一行第一像素的 Y 值。建议使用 YCbCr 格式的窗口具有偶数个像素的宽度。数据包中的像素数据包含偶数个像素。在像素数据的最后像素对应于视频流数据包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。对于所有四个格式来说，位 12（在图中标为“P”）指定像素数据采样是否被打包。当位 12 的值是‘0’时，那么把像素数据字段中每一像素和每一像素内的每一色彩与 MDDI 接口字节边界形成字节对准。当位 12 的值是‘1’时，那么像素数据中的每一像素和每一像素内的每一颜色相对于未留有未使用位的像素内的先前像素或色彩而打包。

在一个实施例中，像素数据属性字段（2 字节）具有按照如下方式解释的一系列位值。位 1 和 0 选择如何路由显示像素数据。对于位值 ‘11’ 来说，向两只眼睛显示数据，对于位值 ‘10’ 来说，只把数据路由至左眼，而对于位值 ‘01’ 来说，只把位值路由至右眼。

位 2 表明像素数据是否以隔行格式提供，值 ‘0’ 意味着像素数据处于标准的逐行格式，并且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 允许的标准隔行模式相似，但是这里隔行是垂直的，而不是水平的。当位 3 是 ‘0’ 时，像素数据处于标准逐个像素格式，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。

位 4 表明像素数据涉及显示器还是照相机，就像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中传送出来一样。当位 4 是 ‘0’ 时，像素数据被传送到显示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被传送出来，这种设备是所属领域众所周知的。

位 5 被保留以供将来使用或者用于 MDD 接口的应用，因此通常被设定为零值或者 ‘0’。

位 7 和 6 是显示器更新位，用于说明将被写入像素数据的帧缓冲器。其它地方讨论了更多的具体效果。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更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。位值或者组合 ‘10’ 被当做无效值或者标记，并且将像素数据忽略，而不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是有用的。

位 8 至 15 被保留以供将来使用，并且因此通常设置为零。

在一个实施例中，2 字节的 X 起始和 Y 起始字段说明像素数据

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 顶沿字段说明由像素数据字段填充的阿尔法光标图像窗口的左沿的 X 坐标以及顶沿的 Y 坐标，而 X 右沿和 Y 底沿字段则说明被更新的阿尔法光标图像窗口的右沿的 X 坐标以及底沿的 Y 坐标。

像素计数字段（2 字节）说明下面的像素数据字段中的像素数目。

2 字节的参数 CRC 字段包含从数据包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，就丢弃整个数据包。

所述像素数据字段包含将被显示的原始视频信息，并且按照由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”。

像素数据 CRC 字段（2 字节）包含只对像素数据的 16 位 CRC。如果这一值的 CRC 验证失败，那么像素数据仍可以使用，但是 CRC 错误计数增加。

可缩放视频流图像

所述 MDD 接口或者协议机制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图像，所述图像可以是比原始图像按比例放大或缩小的，并且所述可缩放图像（the scaled image）被复制到主图像缓冲器。其它地方对可缩放视频流（scaled video stream）的功能以及相关协议支持提供了概述。在可缩放视频流能力数据包内或者由其定义了支持可缩放视频流的能力，响应于请求专用状态数据包而发送该可缩放视频流能力数据包。

36. 可缩放视频流能力数据包

可缩放视频流能力数据包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 79 中总体上示出了可缩放视频流能力数据包的格式。如图 79 所示，在一个实施例中，可缩放视频流能力数据包被构造为具有数据包长度、数据包类型、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，数据包长度被选择为固定的 20 字节，如在长度字段中所示那样，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，不然被设置为零。在一个实施例中，所述客户机使用有效状态应答列表数据包的有效参数应答列表中的参数值 143 来表明其支持可缩放视频流能力数据包的能力。

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。

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。

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。

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。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。

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数据包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 位字地址对准，让 4 字节字段与 32 位字地址对准。

2 字节的 Y Cb Cr 能力字段包含用于说明可以按照 Y Cb Cr 格式显示的分辩率的位数的值。如果可缩放视频流无法使用 Y Cb Cr 格式，那么这一值是零。通常，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。

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数据包中的像素数据能够为打包格式的情况。在“错误!未发现参考源。”中示出了数据包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 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数据包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。

保留 2 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数据包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 位字地址对准，让 4 字节字段与 32 位字地址对准。

37. 可缩放视频流设置数据包

可缩放视频流设置数据包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数据包，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数据包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数据包的最大流数字段中的非零值，来表明其支持可

缩放视频流设置数据包的能力。

在“错误！未发现引用源。”中说明数据包定义。

可缩放视频流设置数据包

数据包长度	数据包类型 =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

数据包内容：

数据包长度 - 2 字节，其包含 16 位无符号整数，用于说明数据包中不包括数据包长度字段的字节总数。这个数据包的数据包长度始终是 24。

数据包类型 - 2 字节，其包含 16 位无符号整数。数据包类型 136 把所述数据包标识为可缩放视频流设置数据包。

hClient ID - 2 字节，其包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留以供将来使用并且被设置为零。

流 ID - 2 字节，其包含用于说明流 ID 的唯一标识符的 16 位无符号整数。这一值是由主机分配的，并且应该是从零到显示器能力数据包中指定的最大流 ID 值的值。所述主机必须小心管理流 ID 值的使用，以便确保每一处于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派。

视频数据格式描述符 - 2 字节，其包含 16 位无符号整数用于说明当前数据包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阿尔法光标图像平面的至少一个有效格式，如在阿尔法光标图像能力数据包中所定义的那样。所述视频数据格式描述符只定义当前数据包的像素格式，而不暗含在特定视频流的使用期将会持续使用

不变的格式。“错误！未发现引用源。”说明视频数据格式描述符是如何编码的。所述格式如下：

如果位[15:13]=000，那么所述视频数据包括单色像素阵列，其中每一像素的位数由视频数据格式描述符字的 3 至 0 位定义。位 11 至 4 应该被设置为零。

如果位[15:13]=001，那么所述视频数据包括彩色像素阵列，其中每一个彩色像素都通过色彩映射（调色板）来指定颜色。视频数据格式描述符字的位 5 至 0 定义每一像素的位数。位 11 至 6 应该被设置为零。

如果位[15:13]=010，那么所述视频数据包括具有原始 RGB 格式的彩色像素阵列，其中红色的每一像素位数由位 11 至 8 定义，绿色的每一像素位数由位 7 至 4 定义，蓝色的每一像素位数由位 3 至 0 定义。每一像素中位数总数是红色、绿色和蓝色的位数和。

如果[15:13]= 011，那么视频数据包括具有亮度和色度信息的具有 4:2:2 的 Y Cb Cr 格式的视频数据阵列。亮度（Y）的每像素的位数是由位 11 至 8 定义的，Cb 分量的位数是由位 7 至 4 定义的，而 Cr 分量的位数是由位 3 至 0 定义的。Cb 和 Cr 分量是以 Y 分量一半的速率发送的。另外，这个数据包的像素数据部分中的视频采样将组织为：Cb_n, Y_n, Cr_n, Y_{n+1}, Cb_{n+2}, Y_{n+2}, Cr_{n+2}, Y_{n+3}, ……，其中 Cb_n 和 Cr_n 与 Y_n 和 Y_{n+1} 相关联，并且 Cb_{n+2} 和 Cr_{n+2} 与 Y_{n+2} 和 Y_{n+3} 相关联等等。Y_n, Y_{n+1}, Y_{n+2} 和 Y_{n+3} 是单个行中从左至右的四个连续像素的亮度值。色彩分量的顺序与微软公司的 UYVY FOURCC 格式相同。如果在由视频流数据包所涉及的窗口的一行中存在有奇数个像素（X 右沿- X 左沿+ 1），那么对应于每行中最后像素的 Cb 值后将跟随有下一行第一像素的 Y 值。建议使用 Y Cb Cr 格式的窗口具有偶数个像素的宽度。数据包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数据包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。

对于所有四个格式来说，位 12（在“错误！未发现引用源。”中

标为“P”)指定像素数据采样是否被打包。“错误!未发现引用源。”说明打包和字节对准像素数据之间的差异。

0 — 像素数据字段中的每一像素和每一像素内的每一颜色与MDDI接口字节边界形成字节对准。

1 — 像素数据中的每一像素和每一像素内的每一颜色相对于未留有未使用位的像素内的先前像素或色彩而打包。

像素数据属性-2字节,它包含按如下方式解释的16位无符号整数:

位1和0选择像素数据应被路由至的显示器。

位[1:0] = 11 或者 00 - 把数据显示给两个眼睛

位[1:0] = 10 -只把数据路由到左眼。

位[1:0] = 01 -只把数据路由到右眼。

位2表明像素数据处于隔行格式。

位2是0 - 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。

位2是1 - 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。

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。

位3是0 - 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 -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。

位4表明像素数据涉及显示器还是照相机。

位4是0 - 像素数据被送入显示帧缓冲器或者来自其中。

位4是1 - 像素数据被送入照相机或者来自其中。

位5被保留供将来使用并且设置为零。

位7和6是显示器更新位,它用于说明要写入像素数据的帧缓冲器。在“错误!未发现引用源”和“错误!未发现引用源”部分比较详细地描述了帧更新位的效用。

位[7:6] = 01 - 像素数据被写入离线图像缓冲器。

位[7:6] = 00 - 像素数据被写入用于更新显示器的图像缓冲器。

位[7:6] = 11 - 像素数据被写入所有图像缓冲器。

位[7:6] = 10 - 无效的。被保留供将来之用。像素数据被忽略并且不写入任何图像缓冲器。

位 8 至 15 被保留供将来使用并且设置为零。

X 左沿 - 2 字节，它包含用于说明目标（destination）图像的左沿的 X 坐标的 16 位无符号整数。

Y 顶沿 - 2 字节，它包含用于说明目标图像的顶沿的 Y 坐标的 16 位无符号整数。

X 右沿 - 2 字节，它包含用于说明目标图像的右沿的 X 坐标的 16 位无符号整数。

Y 底沿 - 2 字节，它包含用于说明目标图像的底沿的 Y 坐标的 16 位无符号整数。

X 图像大小 - 2 字节，它包含用于说明源图像的宽度的 16 位无符号整数。

Y 图像大小 - 2 字节，它包含用于说明源图像的高度的 16 位无符号整数。

CRC - 2 字节，它包含数据包中所有字节的 16 位 CRC，其中包括数据包长度在内。

可缩放视频流确认数据包

可缩放视频流确认数据包允许客户机确认接收到可缩放视频流设置数据包。所述客户机经由有效状态应答列表数据包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数据包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数据包的能力。

数据包定义说明在。

可缩放视频流确认数据包

数据包长度	数据包类型=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

数据包内容:

数据包长度 - 2 字节, 它包含 16 位无符号整数, 用于说明数据包中不包括数据包长度字段的字节总数。这个数据包的数据包长度始终是 10。

数据包类型 - 数据包类型 137 把所述数据包标识为可缩放视频流确认数据包。

cClient ID - 2 字节, 它包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留供将来使用并且设置为零。

流 ID - 2 字节, 它包含用于说明流 ID 的唯一标识符的 16 位无符号整数。这与可缩放视频流设置数据包中由主机分配的值相同。

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

- 0 - 流分配尝试成功。
- 1 - 流解除分配尝试成功。
- 2 - 对一个已经分配了的流 ID 的分配尝试无效。
- 3 - 对一个已经解除分配的流 ID 的解除分配尝试无效。
- 4 - 客户机不支持可缩放视频流
- 5 - 流参数与客户机能力不一致。
- 6 - 流 ID 值大于客户机允许的最大值。
- 7 - 客户机没有足够资源可以用来分配指定的流。

CRC - 2 字节, 它包含数据包中包括数据包长度在内的所有字节的 16 位 CRC。

可缩放视频流数据包

可缩放视频流数据包用来传输与特定的可缩放视频流相关联的像素数据。这个数据包涉及的区域大小是由可缩放视频流设置数据包定义的。所述客户机经由有效状态应答列表数据包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数据包的确认代码字

段中的成功可缩放视频流分配响应，来表明其支持可缩放视频流数据包的能力。

在图-1 中说明所述数据包定义。

可缩放视频流数据包

数据包 长度	数据包 类型=18	hClient ID	流 ID	参数 CRC	像素 计数	像素数据 -12 字节	像素数 据 CRC
2 字节	2 字节	2 字节	2 字节	2 字节	2 字节	数据包长度 -12 字节	2 字节

图-1 可缩放视频流数据包

数据包内容：

数据包长度 - 2 字节，它包含 16 位无符号整数，用于说明数据包中不包括数据包长度字段的字节总数。

数据包类型 - 2 字节，它包含 16 位无符号整数。数据包类型 18 把所述数据包标识为可缩放视频流数据包。

hClient ID - 2 字节，它包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留供将来使用并且设置为零。

流 ID - 2 字节，它包含用于给出流 ID 的唯一标识符的 16 位无符号整数。这一值是在可缩放视频流设置数据包中由主机指定的，并且在可缩放视频流确认数据包中被确认。

像素计数 - 2 字节，它包含用于说明像素数据字段中像素数目的 16 位无符号整数。

参数 CRC - 2 字节，它包含从数据包长度到像素计数的所有字节的 16 位 CRC。如果这一 CRC 未能通过校验，就丢弃整个数据包。

像素数据 - 待缩放和随后被显示的原始视频信息。这里的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。所述数据每次被传输一行，如在“错误！未发现引用源。”部分中定义的那样。

像素数据 CRC - 2 字节，它包含只有像素数据的 16 位 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。

请求专用状态数据包

请求专用状态数据包为主机提供了用于请求客户机按照这个数据包中指定的那样把能力或者状态数据包送回到主机的手段。客户机在下一个反向链路封装数据包中返回指定类型的数据包。如果所述客户机具有响应请求专用状态数据包的能力，那么客户机将设置显示器能力数据包的显示器特征能力指示符字段中的位 17。客户机应该经由显示器能力数据包的显示器特征能力指示符字段的位 21 来表明其支持请求专用状态数据包的能力。

请求专用状态数据包

数据包长度	数据包类型=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，请求专用状态数据包

数据包内容：

数据包长度 - 2 字节，它包含 16 位无符号整数，用于说明数据包中不包括数据包长度字段的字节总数。这个数据包的数据包长度始终是 10。

数据包类型 - 2 字节，它包含 16 位无符号整数。数据包类型 138 把所述数据包标识为请求专用状态数据包。

hClient ID - 2 字节，它包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留供将来使用并且设置为零。

状态数据包 ID - 2 字节，它包含 16 位无符号整数，用于说明客户机应该送到主机的能力或者状态数据包的类型，具体如下：

66 - 应由客户机发送显示器能力数据包。

133 - 应由客户机发送阿尔法光标图像能力数据包。

139 - 应发送有效状态应答列表数据包，用于标识客户机能够发送的能力和状态数据包的准确类型。

140 - 应由客户机发送数据包处理延迟参数数据包。

141 - 应由客户机发送个人显示器能力数据包。

142 - 应由客户机发送显示错误报告数据包。

143 - 应由客户机发送可缩放视频流能力数据包。

144 - 应由客户机发送显示器标识数据包。

位 56 至 63 - 可以用于制造商专用能力和状态标识符。

CRC - 2 字节，它包含包括数据包长度在内的数据包中所有字节的 16 位 CRC。

有效状态应答列表数据包

有效状态应答列表数据包向主机提供了客户机有能力进行响应的一系列状态和能力数据包。客户机应该经由显示器能力数据包的显示器特征能力指示符字段的位 21 来表明其支持有效状态应答列表数据包的能力。

有效状态应答列表数据包

数据包 长度	数据包类型=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	数据包长度-8 字节	2 字节

数据包内容：

数据包长度 - 2 字节，它包含 16 位无符号整数，用于说明数据包中不包括数据包长度字段的字节总数。这个数据包的数据包长度始终是 10。

数据包类型 - 2 字节，它包含 16 位无符号整数。数据包类型 139 把所述数据包标识为有效状态应答数据包。

cClient ID - 2 字节，它包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留供将来使用并且设置为零。

列表中值的数量 - 2 字节，它包含 16 位无符号整数，用于说明随后有效参数应答列表中项目的数目。

有效参数应答列表 - 一系列 2 字节参数，用于说明客户机可以发送到主机的能力或者状态数据包的类型。如果客户机已经表明它可以响应请求专用状态数据包（经由显示器能力数据包中的显示器特征能力指示符字段的位 21），那么它始终应该能够发送至少显示器能力数据包（数据包类型 == 66）和有效状态应答列表数据包（数据包类型 == 139）。这一列表中可以包括的数据包类型及其含义是：

66 - 客户机能够发送显示器能力数据包。

133 - 客户机能够发送阿尔法光标图像能力数据包。

139 - 能够发送有效状态应答列表数据包，用于标识客户机能够发送的能力和状态数据包的准确类型。

140 - 客户机能够发送数据包处理延迟参数数据包。

141 - 客户机能够发送个人显示器能力数据包。

142 - 客户机能够发送显示错误报告数据包。

143 - 客户机能够发送可缩放视频流能力数据包。

144 - 客户机能够发送显示器标识数据包。

位 56 至 63 - 可以用于制造商专用能力和状态标识符。

CRC - 2 字节，它包含数据包中包括数据包长度在内的所有字节的 16 位 CRC。

数据包处理延迟参数数据包

数据包处理延迟参数数据包提供一组参数来允许主机计算完成与特定数据包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。主机可以轮询显示器请求和状态数据包中的状态位，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数据包处理延迟参数数据包中返回的参数来计算完成时间。所述客户机应该经由有效状态应答列表数据包的有效参数应答列表中的参数值 140 来表明其支持数据包处理延迟参数数据包的能力。

数据包处理延迟参数数据包

数据包长度	数据包类型 =140	cClinet ID	列表项 数	延迟参数列表	CRC
2 字节	2 字节	2 字节	2 字节	数据包长度-8 字节	2 字节

数据包内容:

数据包长度 - 2 字节, 它包含 16 位无符号整数, 用于说明数据包中不包括数据包长度字段的字节总数。这个数据包的数据包长度始终是 10。

数据包类型 - 2 字节, 它包含 16 位无符号整数。数据包类型 140 把所述数据包标识为数据包处理延迟参数数据包。

cClient ID - 2 字节, 它包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留供将来使用并且设置为零。

列表项数 - 2 字节, 它包含 16 位无符号整数, 用于说明随后有效参数应答列表中项目的数目。

有效参数应答列表 - 包含一个或多个延迟参数列表项的列表。在“错误! 未发现引用源。”中示出了单个延迟参数列表项的格式。

CRC - 2 字节, 它包含数据包中包括数据包长度在内的所有字节的 16 位 CRC。

延迟参数列表项

延迟数据包类型	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

每一延迟参数列表项在长度上正好是 6 字节, 并且定义为:

延迟数据包类型 - 2 字节, 它包含 16 位无符号整数, 用于说明随后延迟参数应用的数据包类型。

像素延迟 - 1 字节，它包含作为延迟值索引 (index) 的 8 位无符号整数。把从表中读取的值乘以数据包目的字段中的像素总数。像素总数是由该数据包定位的位图的目的区域的宽度乘高度。等式 0 - 1 用来计算总延迟。

水平像素延迟 - 1 字节，它包含作为延迟值表 (与 DPVL 相同的表) 的索引的 8 位无符号整数。把从表中读取的值乘以数据包目的字段中的宽度 (以像素为单位)。等式 0 - 1 用来计算总延迟。

垂直像素延迟 - 1 字节，它包含作为延迟值表 (与 DPVL 相同的表) 的索引的 8 位无符号整数。把从表中读取的值乘以数据包目的字段中的高度 (以像素为单位)。等式 0 - 1 用来计算总延迟。

固定延迟 - 1 字节，它包含作为延迟值表 (与 DPVL 相同的表) 的索引的 8 位无符号整数。从所述表中读取的值是固定延迟参数，用于表示处理与数据包中指定的任何参数值无关的数据包所需的时间。等式 0 - 1 用来计算总延迟。

$$\begin{aligned} \text{延迟} = & (\text{数据包处理延迟(像素延迟)} \cdot \text{像素总数}) + \\ & (\text{数据包处理延迟(水平像素延迟)} \cdot \text{宽度}) + \\ & (\text{数据包处理延迟(垂直像素延迟)} \cdot \text{高度}) + \\ & \text{数据包处理延迟(固定延迟)} \end{aligned}$$

等式 0 - 1, 数据包处理完成时间延迟

对于某些数据包来说，不使用总像素数、宽度或者高度，这是因为在相应的数据包中没有引用这些参数。在这些情况下，相应的像素延迟参数是零。

个人显示器能力数据包

个人显示器能力数据包提供一组参数，用于描述诸如头带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数据包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数据包的能力。

个人显示器能力数据包

数据包	数据包类	cClient	子像素	像素形状	水平视场	垂直视场	视轴交叉
长度	型=141	ID	布局				
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

数据包内容:

数据包长度 - 2 字节, 它包含 16 位无符号整数, 用于说明数据包中不包括数据包长度字段的字节总数。这个数据包的数据包长度始终是 68。

数据包类型 - 2 字节, 它包含 16 位无符号整数。数据包类型 141 把所述数据包标识为个人显示器能力数据包。

cClient ID - 2 字节, 它包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留供将来使用并且设置为零。

所述子像素布局字段包含 8 位无符号整数, 用于说明子像素自上而下并且从左到右的物理布局, 并且: 使用 0 来表明子像素布局没有定义; 使用 1 来表明红色、绿色、蓝色条纹; 使用 2 来表明蓝色、绿色、红色条纹; 使用 3 来表明具有 2×2 子像素布局的四个像素, 该布局包括红色处于顶部左侧, 蓝色处于底部右侧以及两个绿色子像素, 其中两个绿色子像素的一个位于底部左侧, 另一个位于顶部右侧; 使用 4 来表明具有 2×2 的子像素布局的四个像素, 该布局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。

所述像素形状字段包含 8 位无符号整数，用于说明由特定配置子像素组成的每一像素的形状，其中：使用 0 来表明子像素形状没有被定义；使用 1 来表明圆形；使用 2 来表明正方形；使用 3 来表明矩形；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。

水平视场（HFOV）字段 - 1 字节，它包含 8 位无符号整数，用于说明以 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。

垂直视场（VFOV）字段 - 1 字节，它包含 8 位无符号整数，用于说明以 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。

视轴交叉字段 - 1 字节，它包含 8 位无符号整数，用于说明以 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，那么不指定视轴交叉。{注意：这个参数的指定适合于大多数应用中所需要的范围吗？}

左/右图像重叠字段 - 1 字节，它包含用于说明左右图像重叠百分比的 8 位无符号整数。图像重叠的容许范围的百分比是 1 到 100。值 101 到 255 是无效的并且不应该使用。如果其值是零，那么不指定图像重叠。

透明度（see through）字段 - 1 字节，它包含用于说明图像的透明度百分比的 8 位无符号整数。透明度的容许范围的百分比是 0 到 100。值 101 到 254 是无效的并且不应该使用。如果其值是 255，透明度百分比就没有指定。

最大亮度字段 - 1 字节，它包含 8 位无符号整数，用于说明以 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，那么最大亮度就没有被指定。

光学能力标志字段 - 2 字节，它包含 16 位无符号整数，它包含用于说明显示器光学能力的各种字段。

位 15 至 5 - 被保留供将来之用，设置为零。

位 4 - 眼镜聚焦调节

- 0 - 显示器不具有眼镜聚焦调节。
- 1 - 显示器具有眼镜聚焦调节。

位 3 至 2 - 双目功能

- 0 - 显示器是双目的并且只可以显示 2 维 (2D) 图像。
- 1 - 显示器是双目的并且可以显示 3 维 (3D) 图像。
- 2 - 所述显示器是单目的。
- 3 - 被保留供将来之用。

位 1 至 0 - 左右视场曲率对称性

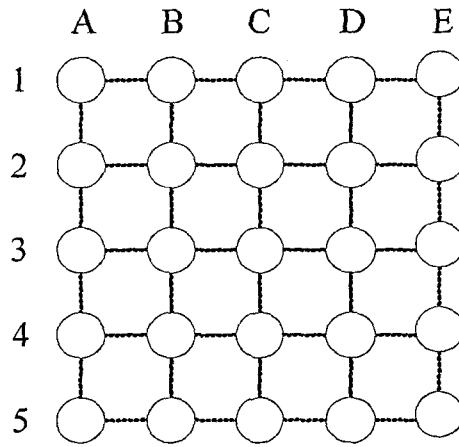
- 0 - 视场曲率没有定义。如果这个字段是零，那么除了点 C3 以外的从 A1 至 E5 的所有视场曲率值设置为零，点 C3 指定显示器的焦距，或者其被设置为零以便表明焦距没有被指定。
- 1 - 左右显示器具有相同对称性。
- 2 - 左右显示器在垂直轴 (列 C) 上互为镜像。
- 3 - 保留以供将来使用。

最小瞳孔间距 (IPD) - 1 字节，它包含用于以毫米 (mm) 为单位来指定最小瞳孔间距的 8 位无符号整数。如果其值是零，那么最小瞳孔间距没有被指定。

最大瞳孔间距 (IPD) - 1 字节，它包含用于以毫米 (mm) 为单位来指定最大瞳孔间距的 8 位无符号整数。如果其值是零，那么最大瞳孔间距没有被指定。

视场曲率点列表 - 一系列 25 个 2 字节参数，用于在 1 到 65535 的范围按照千分之一的屈光度 (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，如下文“错误！未发现引用源。”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。

CRC - 2 字节，它包含数据包中包括数据包长度在内的所有字节的 16 位 CRC。



显示器错误报告数据包

显示器错误报告数据包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数据包，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。所述显示器错误报告数据包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在主机和客户机系统的开发和集成过程中诊断问题最有用。所述客户机应该经由有效状态应答列表数据包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数据包的能力。

显示器错误报告数据包

数据包长度	数据包类型	cClinet ID	列表项数	错误代码列表	CRC
2 字节	=142 2 字节	2 字节	2 字节	数据包长度-8 字节	2 字节

数据包内容：

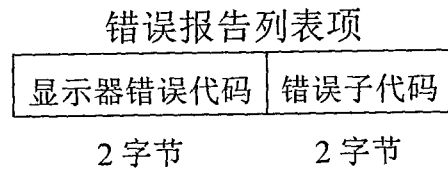
数据包长度 - 2 字节，它包含 16 位无符号整数，用于说明数据包中不包括数据包长度字段的字节总数。

数据包类型 - 2 字节，它包含 16 位无符号整数。数据包类型 142 把所述数据包标识为显示器错误报告数据包。

cClient ID - 2 字节，它包含保留给客户机 ID。这个字段被保留供将来使用并且设置为零。

列表项数 - 2 字节，它包含 16 位无符号整数，用于说明随后错误代码列表中项目的数目。

错误代码列表 - 包含一个或多个错误报告列表项的列表。在“错误！未发现引用源。”中示出了单个错误报告列表项的格式。



在一个实施例中，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，具有这样的结构，包括：用于说明所报告的错误的类型的 2 字节显示器错误代码字段，用于说明与由显示器错误代码数据包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。每一显示器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。

显示器标识数据包

显示器标识数据包允许客户机响应于请求专用状态数据包来返回标识数据。在一个实施例中，客户机使用有效状态应答列表数据包的有效参数应答列表中的参数值 144 来表明发送显示器标识数据包的能力。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显示器能力数据包中无法描述的特殊能力。大概存在两种方法、手段或机制来从客户机读取识别信息。一种是通过使用显示器能力数据包来实现，该显示器能力数据包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显示器能力数据包中类似字段更加丰富的信息集合的显示器标识数据包。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。

显示器标识数据包

数据包 长度	数据包类 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

2 字节的数据包类型字段包含用于把数据包标识为显示器标识数据包的值。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) 包含包括数据包长度在内的数据包中的所有字节的 16 位 CRC。

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，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。

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。

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。

可选 (alternate) 显示器能力数据包

可选显示器能力数据包表明附属于 MDDI 客户机控制器的可选显示器的能力。该数据包是响应于请求专用状态数据包而发送的。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数据包。所述客户机应该经由有效状态应答列表数据包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数据包的能力。

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。显示器能力数据包的可选显示器数目字段用来报告所附属的一个以上的显示器，并且可选显示器能力数据包报告每一可选显示器的能力。所述视频流数据包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。

可选显示器能力数据包

数据包长度	数据包类型=145	cClinet 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

色彩映射 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

数据包内容：

数据包类型 - 2 字节，它包含 16 位无符号整数。数据包类型 145 把所述数据包标识为可选显示器能力数据包。

cClient ID - 2 字节，它包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留供将来使用并且设置为零。

可选显示器号 - 1 字节，包含 8 位无符号整数，用 0 到 15 范围内的整数表明可选显示器的身份。第一可选显示器应是 0 号，而其它可选显示器应该用最大值是可选显示器总数减去 1 的唯一可选显示器号标识。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，并且显示器能力数据包的可选显示器数目字段具有值 1。

保留 1 - 1 字节，它包含被保留供将来之用的 8 位无符号整数。这个字段中的所有位都应设置为零。这个字段的目的在于让所有后续 2 字节字段与 16 位字地址对准，让 4 字节字段与 32 位字地址对准。

位图宽度 - 2 字节，它包含 16 位无符号整数，用于说明以像素个数表示的位图宽度。

位图高度 - 2 字节，它包含 16 位无符号整数，用于说明以像素个数表示的位图高度。

显示窗宽度 - 2 字节，它包含 16 位无符号整数，用于说明以像素个数表示的显示窗宽度。

显示窗高度 - 2 字节，它包含 16 位无符号整数，用于说明以像素个数表示的显示窗高度。

色彩映射 RGB 宽度 - 2 字节，它包含 16 位无符号整数，用于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。即便在色彩映射数据包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：

位 3 至 0 定义每一像素中蓝色的最大位数。0 至 8 是有效的。

位 7 至 4 定义每一像素中绿色的最大位数。0 至 8 是有效的。

位 11 至 8 定义每一像素中红色的最大位数。0 至 8 是有效的。

位 15 至 12 被保留供将来使用并且设置为零。

RGB 能力 - 2 字节，它包含 16 位无符号整数，用于说明可以按照 RGB 格式显示的分辩率的位数。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 15 至 12 被保留供将来使用并且设置为零。

单色能力 - 1 字节，包含 8 位无符号整数，用来指定可以按照单色格式显示的分辩率的位数。如果客户机无法使用单色格式，那么这一值是零。位 7 至 4 被保留供将来使用并且设置为零。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。

保留 2 - 1 字节，它包含被保留供将来之用的 8 位无符号整数。这个字段中的所有位都应设置为零。这个字段的目的在于让所有后续 2 字节字段与 16 位字地址对准，让 4 字节字段与 32 位字地址对准。

Y Cb Cr 能力 - 2 字节，它包含 16 位无符号整数，用于说明可以按照 Y Cb Cr 格式显示的分辩率的位数。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 15 至 12 被保留供将来使用并且设置为零。

显示器特征能力指示符 - 1 字节，它包含 8 位无符号整数，包含一组用于表明客户机中的特定特征是否被支持的标志。设置为 1 的位表明支持所述能力，设置为零的位表明不支持所述能力。

位 0 - 客户机可以接受数据包格式的视频数据。

位 1 至 7 被保留供将来之用，应设置为零。

保留 3 - 1 字节，它包含被保留供将来之用的 8 位无符号整数。这个字段中的所有位都应设置为零。这个字段的目的在于让所有后续 2 字节字段与 16 位字地址对准，让 4 字节字段与 32 位字地址对准。

CRC - 2 字节，它包含包括数据包长度在内的数据包中所有字节的 16 位 CRC。

寄存器访问数据包

寄存器访问数据包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。所述寄存器访问数据包允许 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，对端应该通过按照相同的数据包类型来发送寄存器数据，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来作为响应。寄存器访问数据包可以用来通过指定寄存器计数大于 1 来读取或者写入多个寄存器。客户机使用显示器能力数据包的显示器特征能力指示符字段的位 22 表明支持寄存器访问数据包的能力。

寄存器访问数据包

数据包 长度	数据包类型 =146	bClient ID	读/写 信息	寄存器 地址	寄存器数据 列表	CRC
2 字节	2 字节	2 字节	2 字节	4 字节	数据包长度 -12 字节	2 字节

寄存器访问数据包

2 字节的数据包长度字段包含 16 位无符号整数，用于说明不包

括数据包长度字段的数据包中的字节总数。

数据包类型 - 2 字节, 它包含 16 位无符号整数。数据包类型 146 把所述数据包标识为寄存器访问数据包。

bClient ID - 2 字节, 它包含保留给客户机 ID 的 16 位无符号整数。这个字段被保留供将来使用并且设置为零。

读/写信息 - 2 字节, 它包含 16 位无符号整数, 用于将这个特定数据包说明为写、读或者对读的响应, 并且提供这些数据值的计数。

位 15 至 14 - 读/写标志。

位 [15:14] = 10 - 这是对主机从寄存器地址字段寻址的一个或多个寄存器数据的请求。

位 [15:14] = 00 - 写入这个数据包, 该数据包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数据包包含在寄存器数据字段中。

位 [15:14] = 11 - 包含这个数据包, 这个数据包包含响应于读/写标志设置为 10 的寄存器访问数据包而被请求的数据。寄存器地址字段应该包含对应于第一寄存器数据项的寄存器地址, 而寄存器数据字段应该包含从所述一个或多个地址中读取的数据。

位 [15:14] = 10 - 这一值被保留以供将来使用并且不应该使用。

位 13:0 - 14 位无符号整数, 用于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。

如果在主机发送数据包中位 15 是 0, 那么位 13:0 指定起始于由寄存器地址字段指定的寄存器、且待写入客户机寄存器的寄存器数据列表字段中包含的寄存器数据项的数目。

如果在主机发送的数据包中位 15 是 1, 那么位 13:0 指定客户机应该发送至主机的寄存器数据项的数目。由主机发送的数据包中的寄存器数据字段应该不包含项目并且长度是零。

如果在客户机发送的数据包中位 15 是 1, 那么位 13:0 指定寄存器数据列表字段中包含的寄存器数据项的数目。

在客户机发送的数据包中位 15 不应该被设置为 0。这不是有效

值。

寄存器地址 - 4 字节，包含 32 位无符号整数，它包含要写入其中或者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。

寄存器数据列表 - 一系列待写入客户机寄存器的 4 字节寄存器数据值，或者从客户机设备寄存器中读取的值。

CRC - 2 字节，它包含包括数据包长度在内的数据包中所有字节的 16 位 CRC。

帧同步数据包

帧同步数据包向主机提供用于表明光栅图像起始的能力，并且允许客户机避免使用视频流数据包中的像素和窗口可寻址性特征。这是通过在图像的每一帧开始时发送帧同步数据包、然后为图像数据的每行发送视频流数据包(将像素数据属性字段中的位 5 置位)来实现的。在一个实施例中，客户机使用显示器能力数据包的显示器特征能力指示符字段的位 23 来表明支持帧同步数据包的能力。

所述格式为：

帧同步数据包			
数据包长度	数据包类型=147	bClient ID	CRC
2 字节	2 字节	2 字节	2 字节

数据包类型 - 2 字节，它包含 16 位无符号整数。数据包类型 147 把所述数据包标识为帧同步数据包。

bClient ID - 2 字节，它包含保留给客户机 ID 的 16 位无符号整数。

D. 数据包 CRC

CRC 字段出现在数据包的结尾，有时出现在数据包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 字段的数据包中，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 计算不受数据包开始时的参数影响。

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)线路上的数据包的第一位之前被初始化为值 0x0001，然后数据包的字节首先从 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。

举例来说，如果显示器请求和状态数据包的数据包内容是：0x07，0x46，0x000400，0x00(或者表示为一个字节序列：0x07，0x00，0x46，0x00，0x04，0x00，0x00)，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，Tx_MDDI_Data_With_CRC 线路上输出的最终 CRC 是 0x0eal(或者表示为一个序列 0xa1，0x0e 序列)。

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 错误的每个数据包，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 CHECK_CRC_NOW 激活的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，那么在与数据包末尾对齐的时刻，将 CRC 按照时钟从所述 CRC 寄存器输出。

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 的生成和数据的数据包的传输。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数据包的接收和 CRC 值的校验。

数据包 CRC 的错误代码重载（overload）

每当在主机和客户机之间只传送数据的数据包和 CRC 时，其中不容纳错误代码。唯一的错误是失去同步。否则，人们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。使人遗憾的是这样做既耗时效率又低。

针对一个实施例中的使用，已经开发了这样一种新技术，其中数据包 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，使用数据包的 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。

在一个实施例中，为了提供稳固的错误警告和检测系统，错误代码可以使用一系列数据包、通常是所有数据包被传送若干次，这些数据包是在已经检测到错误之后才被传送或者发送的。进行如此处理，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。

当使用最小量的额外位或字段时，重载 CRC 值的这种技术对系统错误提供非常快的响应。

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与否。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，以表明检测到已经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。

示出了 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 选择希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，用于接收待插入的数据流、数据包和需要的代码，并且重写对应或者适当的 CRC 值，以便把需要的错误代码传送至接收设备。

如上所述，错误代码可以使用一系列数据包被传送若干次，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位置的这些代码期间，利用存储器元件来保存代码的副本，以便根据需要这些存储单元可用于储存或者保留它们的值。

图 67A 和 67B 中更加详细地示出了图 66 的重写机制实现的通用处理。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，根据需要把所述补码或者在某

些情况下是其它代表值，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 判定输出所作的单个选择。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数据包一起传送。

在数据包接收侧，如图 67B 所示，在步骤 6722 中监视数据包 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，并且确定是请求重新传送一个或多个数据包，还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，检测错误的存在。也可以实现独立的错误检测过程和监视。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，最终提取的代码、补码或者其它恢复值都可用于在步骤 6730 从已传输的代码中检测已经出现了哪些错误。

V. 链路从休眠重启

当主机从休眠状态重启正向链路时，它把 MDDI_Data 驱动为逻辑 1 状态约 150 微秒，然后激活 MDDI_Stb，并且同时把 MDDI_Data 驱动为逻辑 0 状态 50 微秒，然后通过发送子帧报头数据包来开始正向链路通信。这样，通过在信号间提供足够的稳定（settling）时间，通常可使总线竞争得以在发送子帧报头数据包之前解决。

当客户机，在这里是显示器，需要来自主机的数据或者通信时，它把 MDDI_Data0 线路驱动至逻辑 1 状态大约 70 微秒，当然需要的时候还可以使用其它周期，然后通过将驱动器置于高阻状态来禁用该驱动器。这一动作让主机在正向链路（208）上开始或者重新启用数据通信并且轮询客户机的状态。主机必须在 50 微秒内检测到请求脉

冲的存在，然后开始启动序列，即把 MDDI_Data0 驱动至逻辑 1 状态 150 微秒并且驱动至逻辑 0 状态 50 微秒。如果显示器检测到 MDDI_Data0 处于逻辑 1 状态多于 50 微秒，那么显示器不应该发送服务请求脉冲。下面进一步讨论涉及休眠处理和启动序列的时间和时间间隔的容差的选择属性。

图 38 中说明没有竞争的典型服务请求事件 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数据包，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。在某些时间段之后，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 处所示那样。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 处所示那样。主机把 MDDI_Data0 驱动至逻辑 0 电平 50 微秒，如点 F 处所示那样，并且按照在某种意义上与 MDDI_Data0 上的逻辑 0 电平相容的方式来生成 MDDI_Stb。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数据包在正向链路上传输数据，如点 G 所示。

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及其接近破坏的子帧报头数据包的最坏情况。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数据包，通知它所述链路将转变到低功率休眠状态。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。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 150 微秒来开始所述链路重启序列，如点 C 处所示那样。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 处所示那样。这是因为显示器需要向主机请求服务并且没有识别出所述主机已开始链路重启序列。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 处所示那样。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 50 微秒，如点 F 所示，并且按照在某种意义上与 MDDI_Data0 上的逻辑 0 电平相容的方式来生成 MDDI_Stb。在把 MDDI_Data0 维持到逻辑 0 电平并且驱动 MDDI_Stb 50 微秒之后，所述主机通过发送子帧报头数据包开始在正向链路上传输数据，如点 G 所示。

从上面的讨论可知，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 150 微秒，然后把 MDDI_Data0 信号驱动为低 50 微秒，同时激活 MDDI_Stb 线路，然后开始传输 MDDI 数据包。就使用 MDDI 装置和方法可获得的数据速率而言，这一过程适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（简化处理或元件的能力）而言，始终要求更快的速度。

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）。按照这一结构，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 微秒周期改变为 150 个时钟周期和 50 时钟周期。

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 150 个时钟周期之后，主机把数据线驱动为低 50 时钟周期，同时继续传输选通信号。在主机已经完成这些过程之后，主机可以开始传输第一子帧报头数据包。

在客户机侧，客户机实现方式现在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，客户机应该能够对处于高状态的数据线的至少 150 个连续时钟周期进行计数，其后跟随有处于低状态的数据线的至少 50 个连续时钟周期。一旦符合这两个条件，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处于高状态的数据线的最初 150 个连续时钟周期。

如早先讨论的那样，用于基于主机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恢复出来，该先前速率为通信链路进入休眠时处于活动状态的速率。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对处于高状态的数据线的至少 150 个连续时钟周期再次进行计数，其后跟随有处于低状态的数据线的至少 50 个连续时钟周期。一旦已经符合这两个条件，客户机就可以开始搜索唯一字。

用于基于客户机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开始之外，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。客户机可以对开始或者处于其唤醒过程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。

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。虽然未必象对于客户机那样有益，对于主机而言，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。

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。虽然未必象对于客户机那样有益，对于主机而言，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。

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。虽然未必象对于客户机那样有益，对于主机而言，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。

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。

图 68A 中说明没有竞争的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数据包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，主机翻转 (toggle) MDDI_Stb 且持续翻转 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，禁用 MDDI_Data0 输出 16~48 周期 (通常包括输出禁用传播延迟)。可能需要在 CRC 后的 48 周期之后，并且在下一阶段 (C) 之前，把客户机中 MDDI_Data0 和 MDDI_Stb 的高速接收器置于低功率状态。

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。

在某些时间段之后，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。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，只要它使驱动器完全启用它们各自的输出即可。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。这允许客户机有时间准备接收。

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 处所示那样。主机把 MDDI_Data0 驱动至逻辑 0 电平 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数据包。主机通过发送子帧报头数据包来开始在正向链路上传输数据，如点 G 所示。

图 68B 中说明没有竞争的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数据包，通知它所述链路将转变为低功率状态时。

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 翻转之前完成待完成的客户机处理，其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，禁用 MDDI_Data0 输出 16 至 48 周期（通常包括输出禁用传播延迟）。在 CRC 后的 48 周期之后，并且在下一阶段(C)之前，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一段时间。

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。

在某些时间段之后，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。可能需要客户机稍微超前于启用接收器时启用该偏移，以便确保接收到有效的差分信号并且防止错误信号，这根据需要而定。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。

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，并且把 MDDI_Stb 驱动至逻辑 0 电平，只要驱动器能启用它们各自的输出即可。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。这允许客户机有时间准

备接收。

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 周期期，如点 F 处所示那样。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。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。

如点 G 和 H 处所示那样，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数据包。主机通过发送子帧报头数据包来开始在正向链路上传输数据，如点 I 所示。

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数据包，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 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。在某些时间段之后，所述主机通过启用 MDDI_Data0 和 MDDI_Stb 驱动器输出来在点 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，如点 E 处所示那样。

在点 E 之后的最多 70 个 MDDI_Stb 周期处，在这里为点 F，客户机还没有意识到主机正在把 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已经需要请求服务，但是没有意识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。

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数据包。主机通过发送子帧报头数据包来开始在正向链路上传输数据，如点 I 所示。

VI. 接口电气规范

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，这允许时钟信息将被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。每当存在背对背 (back-to-back) 状态，0 或 1 时，选通信号将值翻转 (0 或者 1)，这在数据线或者信号上保持相同。

图 40 中以图形形式示出了如何使用 DATA-STB (数据-选通) 编码来传输诸如位“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) 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，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。

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，将其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。

在图 41 中，发送部分 4100 用来生成并且经由中间信号路径 4102 发送原始 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。

在图 41 的接收部分 4120 中，由两个差分线路接收器 4122 和 4124 的每个接收 MDDI_Data0+、MDDI_Data0-和 MDDI_Stb+、MDDI_Stb-信号，差分线路接收器从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 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。所述时钟也具有来自 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没有一个比时钟速率的一半快。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。

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。差分信号路径的每一部分利用用于传送信号的电缆或者导线的特性阻抗的一半的终端电阻来进行源端接。MDDI 数据对在主机和客户机处均进行源端接。由于在给定时间这两个驱动器只有一个是激活的，所以传送链路的源端一直存在终端电阻。所述 MDDI_Stb+和 MDDI_Stb-信号只由主机驱动。

图 42 中示出了作为本发明的 MDD 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，同时在表 VII 中示出了 MDDI_Data 和 MDDI_Stb 的相应 DC 电气特性。这一示例性接口使用低压检测，在这里是 200 mV，具有小于 1 伏的电压摆幅和低功耗。

表 VII

参数	描述	最小值	典型值	最大值	单位
R_{term}	串行终端电阻	41.3	42.2	43.0	欧姆
$R_{hibernate}$	休眠状态偏置终端电阻	8	10	12	千欧
$V_{hibernate}$	休眠状态开路电压	0.5		2.8	伏
$V_{Output-Range}$	相对于 GND, 容许的驱动器输出电压范围	0		2.8	伏
V_{OD+}	驱动器差分输出高电压	0.5			伏
V_{OD-}	驱动器差分输出低电压			-0.5	伏
V_{IT+}	接收器差分输入高门限电压			10	毫伏
V_{IT-}	接收器差分输入低门限电压	-10			毫伏
$V_{Input-Range}$	相对于 GND, 容许的接收器输入电压范围	0		3.0	伏
I_{in}	输入泄漏电流 (休眠偏置除外)	-25		25	微安

在表 VIII 中描述了差分线路驱动器和线路接收器的电气参数和特性。功能上，所述驱动器直接把输入端上的逻辑电平传送至正输出端，并且把输入端的反相直接传送至负输出端。从输入端至输出端的

延迟与差分驱动的差分线路充分匹配。按照大部分实现方式，输出端上的电压摆幅小于输入端上的摆幅，以便使功耗和电磁发射最小化。表 VIII 给出 0.5V 左右的最小电压摆幅。然而，本领域技术人员众所周知的是，还可以使用其它值，并且在一些实施例中，发明人设想了更小的值，这取决于设计限制。

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。

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。

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 传送数据包。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 DATA 信号。负责主机 DATA 传递的驱动器采用使能信号输入端以便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。DATA 和 STB 驱动器中每一个的输出端分别与终端阻抗或者电阻 4216a、4216b、4216c 和 4216d 相连。

终端电阻 4216a 和 4216b 还将充当用于 STB 信号处理的客户机侧接收器 4220 的输入端的阻抗，而附加终端电阻 4216e 和 4216f 分别与电阻 4216c 和 4216d 串联设置在客户机数据处理接收器 4222 的输入端上。客户机控制器中的第六驱动器 4226 用来预备从客户机传送到主机的数据信号，其中驱动器 4214 通过输入侧上的终端电阻 4216c 和 4216d 处理所述数据以便传送到主机来处理。

作为其它地方讨论的休眠控制的一部分，在终端电阻和地以及电压源 4220 之间分别放置另外两个电阻 4218a 和 4218b。电压源用来驱动传送线路至先前讨论的高或低电平，以便管理数据流。

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。

很容易看出：使用标记为 MDDI_Pwr 和 MDDI_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 MDDI_Gnd 部分充当基准地和电源返回路径或者显示器设备的信号。MDDI_Pwr 信号充当显示器设备电源，在示例性的配置中，该电源由主机设备驱动，以便用于低功率应用，所述显示器设备被允许使用高达 500 mA。可以从便携式电源提供 MDDI_Pwr 信号，诸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 MDDI_Gnd 可以在 3.2 到 4.3 伏的范围内。

VII. 时序特性

A. 概述

图 43 说明由客户机采用以便确保来自主机的服务安全步骤和信号电平，以及由主机采用以便提供这种服务的步骤和信号电平。在图 43 中，所示出的第一部分信号示出了从主机传送的链路关闭数据包，然后使用高阻抗偏置电路把数据线驱动为逻辑 0 状态。此时，已经禁用其驱动器的客户机显示器或者主机不传送数据。由于在链路关闭数据包期间 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，一旦这一数据包结束并且逻辑电平变成 0，所述 MDDI_Stb 信号线路也变为逻辑 0 电平。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，例如可以只发送信号以便把通信链路重置为适当的状态，而无需一个由这一主机设备进行的‘已知’的在先通信。

如图 43 所示，来自客户机的信号输出的逻辑电平最初被设置为零。换言之，客户机输出端处于高阻抗，驱动器被禁用。当请求服务时，客户机启用其驱动器并且向主机发送服务请求，这是一个时间段，

表示为“t_{service}”，在这期间，所述线路被驱动为逻辑 1 电平。然后，在主机检测到所述请求之前，经过或者可能需要一定量的时间，这段时间表示为“t_{host-detect}”，此后，主机通过把所述信号驱动至逻辑 1 电平来以链路启动序列做出响应。在这一刻，客户机不再维持（de-assert）所述请求，并且禁用服务请求驱动器，因此来自客户机的输出线路也进入 0 逻辑电平。在这个时候，所述 MDDI_Stb 信号处于逻辑 0 电平。

在称为“t_{restart-high}”的周期内，所述主机把主机数据输出端驱动为‘1’电平，此后，所述主机把逻辑电平驱动至零并且激活 MDDI_Stb 且在称为“t_{restart-low}”的周期内保持，此后，第一正向业务以子帧报头数据包开始，然后传送正向业务数据包。在 t_{restart-low} 周期期间以及后续子帧报头数据包期间，所述 MDDI_Stb 信号是处于激活状态的。

表 VIII 示出了上述各种周期长度的典型时间以及示例性的最小和最大数据速率之间的关系，其中：

$$t_{\text{bit}} = \frac{1}{\text{Link_Data_Rate}}$$

表 VIII

参数	描述	最小值	典型值	最大值	单位
t _{service}	显示器服务请求脉冲时期	60	70	80	微秒
t _{restart-high}	主机链路重启高脉冲时期	140	150	160	微秒
t _{restart-low}	主机链路重启低脉冲时期	40	50	60	微秒
t _{display-detect}	显示器检测链路重启序列的时间	1		50	微秒
t _{host-detect}	主机检测服务请求脉冲的时间	1		50	微秒
1/t _{bit-min-perf}	最小性能设备的链路数据速率	0.001		1	Mbps
1/t _{bit-max-perf}	设备的最大链路数据速率范围	0.001		450	Mbps
	反向链路数据速率	0.0005		50	Mbps
t _{bit}	一个正向链路数据位的周期	2.2		10 ⁶	纳秒

本领域技术人员将易于理解，图 41 和 42 中举例说明的单个元件的功能是众所周知的，并且通过图 43 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。

B. 数据-选通时序正向链路

表 IX 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 $t_{tdd}(\text{host-output})$ 的 Data0 到 Data0 转变）的典型持续时间是 t_{tbit} ，而最小时间大约为 $t_{tbit} - 0.5$ 纳秒，而最大时间大约为 $t_{tbit} + 0.5$ 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，将其分别称为 $t_{tds}(\text{host-output})$ 、 $t_{tss}(\text{host-output})$ 、 $t_{tsd}(\text{host-output})$ 、 $t_{tddx}(\text{host-output})$ 、 $t_{tdxdx}(\text{host-output})$ 、 $t_{tdxs}(\text{host-output})$ 以及 $t_{tsdx}(\text{host-output})$ 。

表 IX

参数	描述	最小值	典型值	最大值	单位
$t_{tdd}(\text{host-output})$	Data0 至 Data0 转变	$t_{tbit} - 0.5$	t_{tbit}	$t_{tbit} + 0.5$	纳秒
$t_{tds}(\text{host-output})$	Data0 至选通转变	$t_{tbit} - 0.8$	t_{tbit}	$t_{tbit} + 0.8$	纳秒
$t_{tss}(\text{host-output})$	选通至选通转变	$t_{tbit} - 0.5$	t_{tbit}	$t_{tbit} + 0.5$	纳秒
$t_{tsd}(\text{host-output})$	选通至 Data0 转变	$t_{tbit} - 0.8$	t_{tbit}	$t_{tbit} + 0.8$	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	$t_{tbit} - 0.5$	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

表 X 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。

表 X

参数	描述	最小值	典型值	最大值	单位
$t_{idd}(\text{display-input})$	Data0 至 Data0 转变	$t_{bit}-1.0$	t_{bit}	$t_{bit}+1.0$	纳秒
$t_{ids}(\text{display-input})$	Data0 至选通转变	$t_{bit}-1.5$	t_{bit}	$t_{bit}+1.5$	纳秒
$t_{iss}(\text{display-input})$	选通至选通转变	$t_{bit}-1.0$	t_{bit}	$t_{bit}+1.0$	纳秒
$t_{isd}(\text{display-input})$	选通至 Data0 转变	$t_{bit}-1.5$	t_{bit}	$t_{bit}+1.5$	纳秒
$t_{iddx}(\text{host-output})$	Data0 至非 Data0 转变		t_{bit}		纳秒
$t_{idx}(\text{host-output})$	非 Data0 至非 Data0 转变		t_{bit}		纳秒
$t_{idxs}(\text{host-output})$	非 Data0 至选通转变		t_{bit}		纳秒
$t_{isdx}(\text{host-output})$	选通至非 Data0 转变		t_{bit}		纳秒

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。在主机转发某些数据包的情况下，诸如反向链路封装数据包或者往返行程延迟测量数据包，在所需要的数据包被转发之后，主机禁用所述线路驱动器，所需数据包例如为图 45 中举例说明的已经传送的参数 CRC、选通对准以及全零数据包。然而，如图 45 所示，所述线路的状态未必即刻从‘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，但是它也可以轻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 数据包周期期间进行的。

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数据包或者往返行程延迟测量数据包这样的数据包时，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 数据包周期之后，主机驱动器被启用并且开始驱动电平（在这里是‘0’），在称为主机

驱动器启用延迟周期的时间段内接近或者达到该值，该周期是在第一数据包被发送之前，在驱动器重新启用周期期间进行的。

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 XI 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。

表 XI

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

C. 数据-选通时序反向链路

图 47 和 48 中示出了用于在反向链路上从客户机驱动器输出端传送数据的数据和选通信号的切换特性以及时序关系。下面讨论某些信号转变的典型时间。图 47 说明在主机接收器输入端，正在传送数据的时序和选通脉冲的前后沿之间的关系。也就是叫做选通信号的上升沿和前沿的建立时间 t_{su-sr} ，以及叫做选通信号的后沿或者下降沿的建立时间 t_{su-sf} 。这些建立周期的典型持续时间最少是在大约 8 纳秒的量级。

图 48 说明由反向数据时序呈现的切换特性和相应的客户机输出延迟。在图 48 中，人们可以看出被传送数据的时序和选通脉冲的前后沿之间的关系，以便说明导致延迟的原因。也就是叫做选通信号的上升沿或者前沿与数据（有效）之间的传播延迟 t_{pd-sr} ，以及叫做数据和选通信号的后沿或者下降沿之间的传播延迟 t_{pd-sf} 。这些传播延迟周期的典型最大时间长度大约是在 8 纳秒的量级。

VIII. 链路控制（链路控制器操作）的实现方式

A. 状态机数据包处理器

经由 MDDI 链路传送的数据包通常以大约 300 Mbps 或更大的速率，如 400 Mbps，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数据包流以便产生数据包，该数据包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。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。

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，诸如控制或者状态数据包。收到那些数据包（控制、状态或者其他预定义数据包）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，从而能够对所述数据包进行操作以便提供所需要的结果（效果），同时音频和视觉数据包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。

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。

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数据包，所述显示器信号处理与正向链路

通道时序同步。也就是说，到达显示器和显示器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。

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数据包中的唯一字。应当注意，这种非同步状态表示选择 I 型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，显示器保存子帧长度字段。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，状态机看到帧报头数据包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 cond 6，或者条件 6。

在系统处于除“非同步”之外的状态的每一种情况下，当检测到唯一字，确定了子帧报头数据包有良好的 CRC 结果，并且子帧长度大于零的时候，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，如果子帧报头数据包中的唯一字或者 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。

B. 同步获得时间

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，那么它将连续地遭遇 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，并且转换到“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“链路关闭数据包”，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。

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，子帧报头数据包上的 CRC 也必须有效的。

子帧报头数据包中的子帧长度可以被设置为零，以便表明在链路关闭，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，显示器在检测到子帧报头数据包之后必须立即经由正向链路接收数据包，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，并且显示器只处理正向链路数据包，而所述接口处于在图 49 中集中显示为“同步中”状态的那些状态。

显示器与正向链路信号同步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”的可能性更大。同时，当正向链路数据速率更慢时，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。

C. 初始化

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 I 型接口开始工作。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数据包，其跟随有反向链路封装数据包，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显示器能力数据包来响应。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显示器能力数据包和显示器请求和状态数据包。

主机检查显示器能力数据包的内容，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显示器使用彼此兼容的协议版本。所述协议版本通常是作为显示器能力数据包的开头两个参数，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。

D. CRC 处理

对于所有数据包类型来说，数据包处理器状态机确保 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。

E. 可选失去同步校验

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，该图说明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。

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，就象图 49 中一样。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 也在这个数据包上传递（符合条件 61），数据包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 数据包的任何 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 数据包的又一 CRC 失败会导致移到两个同步失败状态 4912。利用正确的 CRC 值解码的数据包将导致状态机返回同步中状态 4908。

发生了变化的地方是利用了 CRC 值或者对‘每一’数据包的 CRC 确定。也就是说，要让状态机注意（look at）每一数据包的 CRC 值以便确定失去同步，而不是只观察子帧报头数据包。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。

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。

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意识到同步失败已经出现，这种意识到同步失败的速度要比它必须等待几（在这里是三）个数据包时间或者周期快的多，且其中所述数据包要比子帧长度大得多。如果对唯一字的测试表明它不存在，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。检查正确的唯一字存在的过程向状态机添加了条件 65（cond 65），说明唯一字不正确。如果期望在客户机上接收到子帧数据包但是不相配，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 所要遇到的。

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数据包的传送，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数据包。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数据包解码。如果子帧长度计数器需要中断正被解码的某些其它数据包，那么由于数据包不应跨越子帧边界，所以可以确定已经失去同步。

IX. 数据包处理

对于状态机接收的上述每个类型的数据包来说，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数据包。

表 XII

数据包类型	数据包处理器状态机响应
子帧报头 (SH)	确认良好的数据包，获取子帧长度字段并且向通用处理器发送数据包参数。
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，必要时拆开已打包的像素数据，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。

数据包类型	数据包处理器状态机响应
音频流 (AS)	向音频采样时钟发生器发送音频采样速率设置, 区分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位置。
反向链路封装 (REL)	便于在适当时间按照反向发送数据包。检查反向链路标志, 需要的时候发送显示器能力数据包。只要合适, 还发送显示器请求和状态数据包。
显示器能力 (DC)	当主机请求时, 使用反向链路封装数据包的反向链路标志字段来发送这种类型的数据包。
键盘 (K)	如果存在并且需要使用, 就把这些数据包传递至与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数据包。
指示设备 (PD)	如果存在并且需要使用, 就把这些数据包传递至与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数据包。
链路关闭 (LS)	记录链路被关闭的事实并且通知通用处理器。
显示器服务请求和状态 (DSRS)	发送这个数据包作为反向链路封装数据包中的第一数据包。
位块传送 (BPT)	解释诸如视频数据格式描述符之类的数据包参数, 确定哪些像素首先移动, 并且在根据需移动位图中的像素。
位图区域填充 (BAF)	解释数据包参数, 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)	解释数据包参数, 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。
休眠期间的显示器服务请求 (DSR)	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)	可以把这些数据包传递至通用处理器并从中接收。用于接收这种类型的数据包并且以确认来构成响应的逻辑基本上是最小的。因此, 这一操作还可以在数据包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)	可能直接作用于这种数据包或者通过把这种数据包传送至通用处理器, 以及命令硬件进行模式变换来实现。

X. 降低反向链路数据速率

本发明人已经观察到：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，这是非常合乎需要的。例如，在用于传送反向链路封装数据包的反向数据数据包字段期间，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所述主机链路控制器生成对应于 MDDI_Data0 信号的 MDDI_Stb 信号，好像 MDDI_Data0 正在发送全零。MDDI_Stb 信号从主机被传送至客户机，在那里将它用来生成时钟信号以便从显示器传送反向链路数据，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，在接近 Stb+/-生成、电缆传送至显示器、显示器接收器、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，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。

根据正向链路数据速率以及遇到的信号处理延迟，需要 MDDI_Stb 信号上一周期以上的时间来用于“往返行程”效果或者待完成的事件集合，这导致了时间或者周期量的不需要的消耗。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。

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。尽管不需要，但是通过使用往返行程延迟测量数据包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，每一系统通常可以表现得更好。

往返行程延迟是通过让主机把往返行程延迟测量数据包发送至显示器来测量的。显示器通过在称为测量周期字段的数据包中的预选测量窗内或者在这个期间里，把一个 1 序列发送回主机来响应这个数据包。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。

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，或者对其计数。应注意的是，在测量计数即将增加以前，有可能收到来自客户机的响应一小部分正向链路时钟周期。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，其中按照图形方式说明代表主机处 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已经从显示器收到响应序列。如果把延迟假定为 6，那么主机将在位刚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，在用来计算反向速率除数以前，测量得到的延迟通常应该加 1。

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：

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

对于给定实例来说，它变为：

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

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。

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在反向链路封装数据包的反向链路数据包字段的第一位开始时出现。在图 52 中说明用于以下给出的实例的情况。所述计数器在 MDDI_Stb 信号每一上升沿的增加 1，并且直到计数返回（wrap

around) 零的计数, 是由反向链路封装数据包中的反向速率除数参数设定的。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 200 Mbps, 而反向速率除数是 4, 那么反向链路数据速率可以表示为:

$$\frac{1}{2} \cdot \frac{200\text{Mbps}}{4} = 25\text{Mbps}$$

图 52 中示出了反向链路封装数据包中的 MDDI_Data0 和 MDDI_Stb 信号线的时序的实例, 其中用于举例说明的数据包参数具有如下值:

数据包长度 = 1024 (0x0400)	转向 1 长度 = 1
数据包类型 = 65 (0x41)	转向 2 长度 = 1
反向链路标志 = 0	反向速率除数 = 2
参数 CRC = 0xdb43	全零是 0x00

数据包长度和参数 CRC 字段之间的数据包数据是:

0x00, 0x04, 0x41, 0x00, 0x02, 0x01, 0x01, 0x43, 0xdb, 0x00, ……

从显示器返回的第一反向链路数据包是具有数据包长度 7 和数据包类型 70 的显示器请求和状态数据包。这个数据包从字节值 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数据包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。把主机至显示器往返行程延迟为零的理想波形显示为虚线轨迹。

传送参数 CRC 字段的 MS 字节, CRC 的前面是数据包类型, 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。

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数据包。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。图中示出正传送的数据包字段的第一字节（在这里是 11000000）在转向 1 之后出现，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转变（passage）中的延迟，并且正如位三所看到的那样。

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。

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。

XI. 转向和保护时间

如早先讨论的那样，反向链路封装数据包中的转向 1 字段和往返行程延迟测量数据包中的保护时间 1 字段，指明允许在启用显示器接口驱动器以前禁用主机接口驱动器的持续时间值。转向 2 和保护时间 2 字段提供用于允许在启用主机驱动器以前显示驱动器被禁用的时间值。保护时间 1 和保护时间 2 字段通常以预置长度或者预选长度值来填充，这些长度不能被调节。根据正使用的接口硬件，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。

有几个因素有助于确定转向 1 的长度，这些因素包括正向链路数据速率以及主机中 MDDI_Data 驱动器的最大禁用时间。在表 XI 中指定了最大主机驱动器禁用时间，其中表明驱动器花费最多约 10 纳秒的时间来完成禁用并且约 2 纳秒来完成启用。主机驱动器被禁用所需的正向链路时钟的最小数目按照如下关系来表示：

$$\text{要禁用的时钟}_{TA1} = \frac{\text{正向链路数据速率}}{\text{接口类型因子}_{FWD}} \cdot \text{主机驱动器禁用延迟}_{\max}$$

转向 1 所允许的值的范围按照如下关系来表示:

$$\text{转向}1 \geq \text{进位舍入到下一整数} \left(\frac{\text{要禁用的时钟}_{TA1} \cdot \text{接口类型因子}_{FWD}}{8} \right)$$

其中对于 Type-1 来说, 接口类型因子是 1, 对于 II 型来说是 2, 对于 III 型来说是 4, 而对于 IV 型来说是 8。

结合上述两个等式可以看出, 接口类型因子可以抵消, 于是转向 1 被定义为:

$$\text{转向}1 = \text{进位舍入到下一整数} \left(\frac{\text{正向链路数据速率} \cdot \text{主机驱动器禁用延迟}_{\max}}{8} \right)$$

例如, 1500 Mbps 的 III 型正向链路会使用以下转向 1 延迟:

$$\text{转向}1 = \text{进位舍入到下一整数} \left(\frac{1500\text{Mbps} \cdot 10\text{ns}}{8} \right) = 2 \text{ 字节}$$

随着往返行程延迟增加, 时序边缘从主机被禁用的时间点改进到显示器被启用的时间点。

用于确定通常用于转向 2 的持续时间的因素是正向链路数据速率、显示器中 MDDI_Data 驱动器的最大禁用时间以及通信链路的往返行程延迟。禁用显示驱动器需要的时间的计算基本上与上述主机驱动器的相同, 并且按照如下关系被定义:

$$\text{要禁用的时钟}_{TA2} = \frac{\text{正向链路数据速率}}{\text{接口类型因子}_{FWD}} \cdot \text{显示器驱动器禁用延迟}_{\max}$$

并且转向 2 的允许值范围可以表示为:

$$\text{转向}2 \geq \text{进位舍入到下一整数} \left(\frac{\text{要禁用的时钟}_{TA2} + \text{往返行程延迟} + 1}{\left(\frac{8}{\text{接口类型因子}_{FWD}} \right)} \right)$$

例如, 具有 10 个正向链路时钟的往返行程延迟的 1500 Mbps 的 III 型正向链路通常使用以下数量级的转向 2 延迟:

$$\text{要禁用的时钟}_{TA2} = \frac{1500\text{Mbps}}{4} \cdot 10\text{ns} = 3.75$$

$$\text{转向}2 \geq \text{进位舍入到下一整数} \left(\frac{3.75 + 10 + 1}{\left(\frac{8}{4}\right)} \right) = 8$$

XII. 可选反向链路时序

虽然使用上述的时序和保护带 (guard band) 能够实现高数据传输速率接口, 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。

如上所述, 反向链路时序的先前方式被如此配置, 即使时钟周期的数目从反向时序数据包的保护时间 1 的最后位开始计数, 直到在 IO 时钟的上升沿对第一位采样为止。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:

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

这样做提供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, 或者以更高数据传输速率运行, 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。

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, 但是在反向时序数据包期间, 主机在上升沿和下降沿上均对数据线采样。这样做允许主机采集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通信反向封装数据包上的数据采样。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由客户机在反向链路时序内发送的 0xFF 0xFF 0x00 模式的第一沿, 以便确定在反向封装数据包中的什么地方

采样。

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，因为反向位周期内出现的唯一上升沿。对于反向速率除数 2 来说，由于周期沿 ‘c’ 比 ‘b’ 接近位沿时，所以最佳采样点也许仍是时钟周期前沿 ‘b’。对于反向速率除数 4 来说，最佳采样点也许是时钟周期沿 ‘d’，由于它更加接近值已经稳定的反向位的后沿。

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，那么反向速率除数 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 2 来说，最佳采样点是沿 ‘b’，并且对于反向速率除数 4 来说，最佳采样点是沿 ‘c’。

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

主机使用这一技术可以在数据线上观察到时序数据包的上升数据沿之前找到上升时钟沿的数目。因此，能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，来做出决定，以便合理地确保所述位的采样始终尽可能靠近中部。

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 (explor)”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）可以从除数 1 开始，并且校验从客户机接收的反向状态数据包的 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，试图再次请求状态数据包。如果第二次请求的数据包被破坏，那么可以再次增大除数，并且再次进行请求。如果这个数据包被正确地解码，那么这一反向速率除数可以用于所有

将来的反向数据包。

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数据包解码。当然，由于这一方法不确保反向链路完美，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，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增加而增加。

这种实现方式看来似乎最适用于 I 型反向数据，但是对 II 型到 IV 型反向数据来说可能会出现問題，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一个数据对的速率运行所述链路。然而，所述数据速率也许不需要降低到先前方法，即使是采用 II 型至 IV 型的操作。如果在每一数据线上复制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最适用。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，所述主机能够在对所有位采样之后根据所述数据对集合来重构数据流：对于 II 型是两位，对于 III 型是四位而对于 IV 型是八位。主机的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。

XIII. 链路延迟和偏离的效果

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控制器逻辑、线路驱动器和接收器以及电缆和连接器，如同下面略述的一样。

A. 由偏离限制的链路时序分析 (MDDI I 型)

1. I 型链路的延迟和偏离实例

图 57 中示出了类似于图 41 所示，用于支持 I 型接口链路的典型接口电路。在图 57 中，为 MDDI I 型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或者典型值。MDDI_Stb 和 MDDI_Data0 之间延迟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 输入端的数据必须在时钟沿之后稍微改变，从而能够对它可靠地采样。所述图示出了用于利用产生这一时序关系来解决两个不同问题的两个级联延迟线 5732a 和 5732b。在实际实现方式中，可以将它们合并成单个延迟元件。

图 58 说明对经由所述接口的示例性信号处理的 I 型链路上的数据、Stb 和时钟恢复时序。

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 1 5732a 应该匹配或者超过 RXXOR 级的 XOR 门 5736 的延迟，通过如下关系确定该延迟：

$$t_{PD-min(延迟\ 1)} \geq t_{PD-max(XOR)}$$

需要符合这一要求，这样一来，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。

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：

$$t_{PD-min(延迟\ 2)} = t_{H(RXFF)}$$

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。

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：

$$t_{\text{偏离-max(RXXOR)}} = t_{\text{PD-max(Delay1)}} - t_{\text{PD-min(XOR)}}$$

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

MDDI I 型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 RXFF 级设置的总计数据的函数。链路中直到 RXRCVR 级输出端的总延迟偏离可以表示为：

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

并且最小位周期由如下等式给出：

$$t_{\text{位-min}} = t_{\text{偏离-max(链路)}} + t_{\text{偏离-max(RXXOR)}} + t_{\text{PD-max(Delay2)}} + t_{\text{SU(RXFF)}}$$

在图 57 所示的实例中， $t_{\text{偏离-max(链路)}} = 1.4$ 纳秒，并且最小位周期可以表示为：

$$t_{\text{位-min}} = 1.4 + 0.3 + 0.2 + 0.5 = 2.4 \text{ 纳秒，或者近似表示为 } 416 \text{ Mbps。}$$

B. MDDI II 型、III 型和 IV 型的链路时序分析

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 II 型、III 型和 IV 型接口链路。在 TXFF (5904)、TXDRVR (5908)、RXRCVCR (5922) 和 RXFF (5932, 5928, 5930) 级使用附加元件，以便支持附加信号处理。在图 59 中，为 MDDI II 型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 数据信号之间还有偏离。包括触发器 5928 和 5930 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，从而能够可靠地对其采样。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 II 型、III 型或者 IV 型链路的数据速率的上限。

图 60A、60B 和 60C 中说明两个数据信号和 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。

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 3：

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

最大链路速度取决于允许的最小位周期。当 MDDI_DataX 尽可能晚的到达时，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：

$$t_{位-min} = t_{偏离-max}(LINK) + t_{PD-max}(Delay3) + t_{SU}(RXFFB) - t_{PD-min}(XOR)$$

于是链路速度的上限为：

$$t_{PD-max}(Delay3) = t_{PD-min}(Delay3)$$

并且在以上假定下：

$$t_{位-min(lower-bound)} = 2 t_{偏离-max}(LINK) + t_{PD-max}(XOR) + t_{SU}(RXFFB) + t_{H}(RXFFB)$$

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：

$$t_{位-min}(lower-level) = 2 \cdot 1.4 + 1.5 + 0.5 + 0.1 = 4.8 \text{ 纳秒，大约是 } 208$$

Mbps。

这比可以用于 I 型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的影响。

XIV. 物理层互连描述

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，可以由 Hirose Electric Company Ltd. 制造的零件号为 3260 - 8S2 (01) 的零件，并且在显示器设备侧，可以由 Hirose Electric Company Ltd. 制造的零件号为 3240-8P-C 的零件。表 XIII 中列出了供 I 型/II 型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。

表 XIII

信号名称	引脚号	颜色	信号名称	引脚号	颜色
MDDI_Pwr	1	白色	MDDI_Gnd	2	与白色配对的黑色
MDDI_Stb+	3	绿色	MDDI_Stb-	4	与绿色配对的黑色
MDDI_Data0+	7	蓝色	MDDI_Data0-	8	与蓝色配对的黑色
MDDI_Data1+	11	棕色	MDDI_Data1-	12	与棕色配对的黑色
MDDI_Data2+	15	红色	MDDI_Data2-	16	与红色配对的黑色
MDDI_Data3+	19	桔红色	MDDI_Data3-	20	与桔红色配对的黑色
MDDI_Data4+	17	TBD1	MDDI_Data4-	18	与 TBD1 配对的黑色
MDDI_Data5+	13	TBD2	MDDI_Data5-	14	与 TBD2 配对的黑色
MDDI_Data6+	9	TBD3	MDDI_Data6-	10	与 TBD3 配对的黑色
MDDI_Data7+	5	TBD4	MDDI_Data7-	6	与 TBD4 配对的黑色
			屏蔽		

所述屏蔽与主机接口中的 MDDI_Gnd 相连，并且电缆中的屏蔽地线与显示器连接器的屏蔽相连。然而，这个屏蔽和地线不与显示器内部的电路地相连。

选择或者设计互连元件或设备，使其足够小以用于移动通信和计算设备，诸如 PDA 和无线电话，或者便携式游戏设备，而与相对的设备大小相比不会超出或者不美观。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 I 型和 II 型而言，该差分数据具有高达 450 Mbps 左右的传送速率，而对于 8 位并行 IV 型版本而言，具有高达 3.6 Gbps 的传送速率。

对于内部模式应用来说，同样对于使用的导线而言没有连接器，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

于接受安装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互联集成电路的导线上。

XV. 操作

图 54A 和 54B 中示出了使用本发明实施例的接口操作期间，在处理数据和数据包的时候进行的通用步骤的概况，并且图 55 中示出了处理所述数据包的接口装置的概况。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，那么它只可以进入等待状态某个预定长度，这取决于应用，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，用户也许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，诸如用于 U 型接口的连接，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。

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数据包。在步骤 5404，客户机可以发送显示器服务请求或者状态数据包。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。一旦通信链路同步化，并且主机试图与客户机通信，客户机就也向主机提供显示器能力数据包，如同在步骤 5408 一样。主机现在可以开始确定客户机可能支持的支持类型，包括传送速率。

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 I 型、U 型、II 型等等。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数据包来优化通信链路的时序，如步骤 5411 所示。

如前所述，所有传送从步骤 5412 示出的正在传送的子帧报头数据包开始，继之为在步骤 5414 示出的正在传送的数据类型和填充符数据包，在这里数据类型是视频和音频流数据包。所述音频和视频数据已经预先预备好或者映射到数据包中，并且根据需要或者期望插入填充符数据包以便填写介质帧所需要的位数。主机能发送诸如正向音频通道启用数据包的数据包来激活音响设备。另外，在步骤 5416 中，主机能使用上述其它数据包类型来传送命令和信息，在这里显示为色彩映射、位块传送或者其它数据包的传送。此外，所述主机和客户机可以使用适当的数据包交换涉及键盘或者指示设备的数据。

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），所述条件导致数据包的预备或者呈现速度减慢。接收数据的显示器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。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。请求改变为更高传送速率模式是更加合乎需要的。

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议定接口模式。这一点在步骤 5420 中示出，其中所述主机客户机发送接口类型切换请求数据包以请求切换到另一模式，所述客户机发送接口类型确认数据包，以便确认寻求改变，并且所述主机发送执行类型切换数据包以便改变为指定模式。

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数据的数据包，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，或者是从它们收到的数据，当然这些元件还可以存在于主机一侧。这些数据包通常使用通用处理

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。

在数据和命令已经在主机和客户机间交换之后，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服务于所述传送而做出决定。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数据包，并且两侧终止传送数据。

在上述操作处理中传送的数据包将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。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，诸如专用 USB 接口、存储器元件或驻留在于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。

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，并且传送数据包。

XVI 显示帧缓冲器

与计算机图形相比，视频数据的缓存要求对于移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，因此显示器上的图像能本地刷新。

显示全运动视频（每一介质帧中显示器的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输入像素数据，而从第二帧缓冲器更新显示器上的图像。可以用两个以上的显示缓冲器来消除可视假象，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 91A 中说明这一概念，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。

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 91B 中所示那样。

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 91C 所示，并且随后通过把显示器更新位设置为“01”来把运动图像像素写入离线缓冲器。

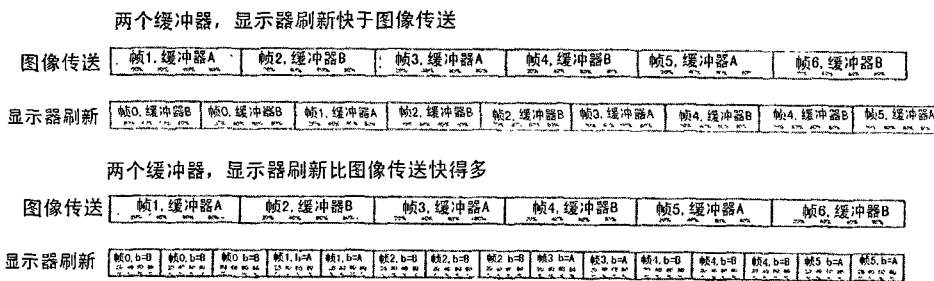
如下规则描述了缓冲器指针的有益操纵，同时把新的信息写入客户机并且刷新显示器。存在三个缓冲器指针：`current_fill`（当前填充）指向当前正在由 MDDI 链路上的数据填充的缓冲器；`just_filled`（刚刚填充）指向最近被填充的缓冲器；`being_displayed`（正在显示）指向目前正用于刷新显示器的缓冲器。所有三个缓冲器指针可以包含从 0 到 $N-1$ 的值，其中 N 是显示器缓冲器的数目，而 $N \geq 2$ 。对缓冲器指针的运算是对 N 取模（ $\text{mod } N$ ），例如，当 $N = 3$ 并且 `current_fill` = 2 时，增加 `current_fill` 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数据包等于零）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`。

MDDI 视频流数据包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。根据 `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期（epoch）中的最后像素之后并且在它开始刷新下一帧刷新时期中的第一像素以前，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。

所述视频流数据包包含一对显示器更新位，说明应写入像素数据的帧缓冲器。所述显示器能力数据包具有三个附加位，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

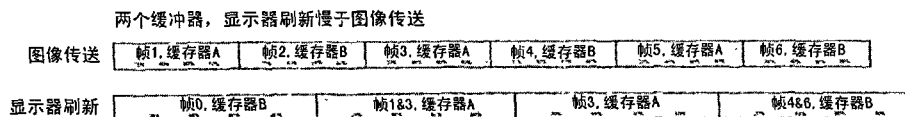
要基于用户输入递增地刷新，或者根据从计算机网络接收的信息来导出。显示器更新位组合“00”和“11”通过让像素数据写入正在显示的帧缓冲器或者写入两个帧缓冲器来支持这一操作模式。

当支持视频图像时，图 92 说明当显示器更新位等于“01”，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，当用于目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。



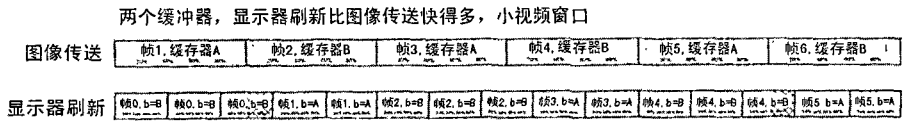
涉及图 92 的一个重要假设是：图像是作为像素的连续流从主机接收的，所述像素的连续流是按照客户机从帧缓冲器读取像素以便刷新显示器的顺序来传输的（通常从左上开始，逐行读取，直到屏幕的右下角）。在显示器刷新和图像传送操作涉及相同的帧缓冲器的情况下，这是重要的细节。

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 93 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，图像碎片是如何出现的。

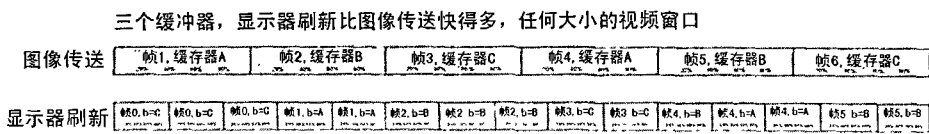


在包含计算机图形图像和运动视频图像的组中的图像中，视频像

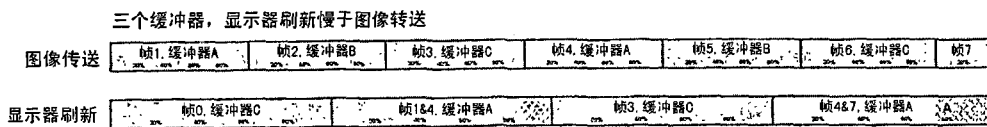
素数据可能占据小部分介质帧。在显示器刷新操作和图像传送涉及相同的帧缓冲器的情况下，这是十分明显的。这些情况通过图 94 中的交叉影线的阴影示出，其中从缓冲器中读取以便刷新显示器的像素可能是两帧以前写入缓冲器的像素，或者它们可以对应于正被写入同一帧缓冲器的帧。



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 95 所示。

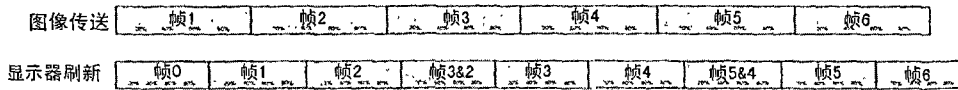


然而，如果显示器刷新速率小于 MDDI 链路上的介质帧速率，那么仍存在问题，如图 96 所示。

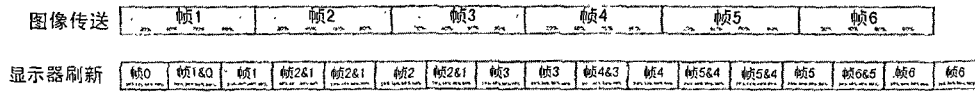


运动视频图像使用单个缓冲器多少存在问题，如图 97 所示。当显示器刷新快于图像传送到缓冲器中时，正刷新的图像有时将显示正写入的帧的上部，而图像的较低部分则是先前传送的帧。当显示器刷新快于图像传送（优选操作模式）时，示出类似分裂图像的帧的情况将更加频繁地出现。

一个缓冲器，显示器刷新快于图传转送



一个缓冲器，显示器刷新比图像传送快得多



XVII. 延迟值表

数据包处理延迟参数数据包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。

表 XX (添加整个新表)

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	

通过将指定参数作为表的索引进行表格查找来计算延迟。这意味着延迟等于数据包处理表(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，那么延迟等于数据包处理表(134)，是 16 微秒。值 255 表明无法通过计算确定命令完成时间，主机必须检查显示器请求和状态数据包中的图形繁忙标志或者 MCCS VCP 控制参数 B7h。

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，以计算整个数据包处理延迟。

XVIII 多客户机支持

当前协议版本看起来无法直接支持多个客户机设备。然而，大部分数据包包含保留的客户机 ID 字段，可以将它用于对具有多个客户机的系统中的特定客户机设备寻址。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。所述子帧报头数据包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。

在具有多个客户机的系统中，把客户机经由客户机的菊花链（Daisy-chained）或者使用集线器连接至主机是十分有益的。

XVIII. 附录

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，在这里对某些数据包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。

A. 关于视频流数据包

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 选择如何路由显示像素数据。对于位值 ‘11’ 来说，向两只眼睛显示数据，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。

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 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。

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帧缓冲器或者从中传送出。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。

位 5 用来表明何时像素数据包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。所述帧同步数据包定义下一行作为图像的顶行。

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有用。

位 8 至 11 形成 4 位无符号整数，用于说明要将数据路由到的可选显示器或者显示位置。位 0 和 1 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。

位 12 至 15 被保留以供将来使用并且通常应设定为零。

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 X 坐标以及顶沿 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 X 坐标以及底沿 Y 坐标。

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。

参数 CRC 字段 (2 字节) 包含从数据包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, 就丢弃整个数据包。

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”。

像素数据 CRC 字段 (2 字节) 包含只有像素数据的 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。

B. 关于音频流数据包

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。这样做简化了应用的通信, 诸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, 以便用于新设计需要附加标识的情况, 正如本领域技术人员预料的那样。

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。

音频采样计数字段（2 字节）指定这个数据包中音频采样的数目。

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。

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

参数 CRC 字段（2 字节）包含从数据包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，就丢弃整个数据包。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。

C. 关于用户定义的流数据包

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流。流参数和流数据字段的内容通常由 MDDI 装备制造制造商定义。2 字节的流参数 CRC 字段包含从数据包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，就丢弃整个数据包。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，通常要求流数据被缓存，直到 CRC 被确认为是良好的为止。如果 CRC 没有通过校验，那么所述 CRC 错误计数增加。

D. 关于色彩映射数据包

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。

2 字节的色彩映射项目计数字段使用值来指定总数为 3 的字节色彩映射项目，这些项目包含在色彩映射数据字段中，或者存在于这个数据包的色彩映射数据中的色彩映射表的条目。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）指定这个数据包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。

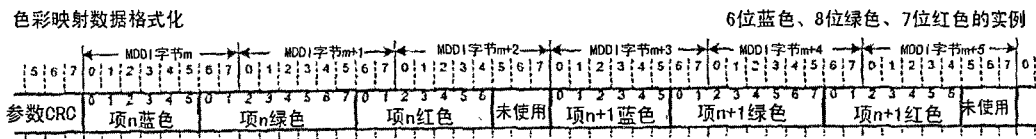
2 字节的参数 CRC 字段包含从数据包长度到音频编码字节的所有字节的 CRC。如果这一 CRC 未能通过校验，就丢弃整个数据包。

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数据包，那么可以通过发送每一数据包中具有不同色彩映射数据和色彩映射偏移的多个数据包来指定整个色彩映射。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数据包的色彩映射 RGB 宽度字段指定的相同。

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。

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。“错误!未发现引用源。(新的 Z??)” 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，色彩映射数据包中的色彩映射项大小等于 21，而显示器能力数据包的色彩映射 RGB 宽度字段等于 0x0786。



E. 关于反向链路封装数据包

参数 CRC 字段（2 字节）包含从数据包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，就丢弃整个数据包。

在一个实施例中，反向链路标志字段（1 字节）包含用于请求来自显示器的信息的一组标志。如果位（例如位 0）被设置为 1，那么主机使用显示器能力数据包从显示器请求指定信息。如果所述位是零，那么主机不需要来自显示器的信息。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。

反向速率除数字段（1 字节）指定与反向链路数据时钟相关的出现的 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。对于 I 型接口来说，反向数据速率等于反向链路数据时钟，对于 II 型、III 型和 IV 型接口来说，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。

全零 1 字段包含一组被设置为零值的字节，这是通过将 these 位设置成逻辑 0 电平来实现的，并且用来确保所有 MDDI_Data 信号在第一保护时间周期内禁用线路驱动器之前都处于零状态，以便允许反映出的逻辑 1 电平在转向 1 字段期间禁用主机的线路驱动器之前消耗掉。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。

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。分配由转向长度参数指定的字节数目，以便允许主机中的 MDDI_Data 线路驱动器在客户机中的线路驱动器被启用以前被禁用。在转向 1 的位 0 期间，所述主机禁用其 MDDI_Data 线路驱动器，并且客户机启用其输出端并且在转向 1 的最后位期间把 MDDI_Data0 驱动为逻辑 0。MDDI_Stb 信号表现得如同转向 1 周期全部是零。转向 1 的建议长度是主机中 MDDI_Data 驱动器禁用输出端所需的字节数目。这基于上述输出禁用时间、正向链路数据速率以及正使用的正向链路接口类型选择。上面给出了转向 1 设置的更加完整的说明。

全零 2 字段包含一组被设置为零值的字节，这是通过将 these 位设置成逻辑 0 电平来实现的，并且用来确保所有 MDDI_Data 信号都处于零状态，以便允许反映出的逻辑 1 电平在转向 1 字段期间禁用主机的线路驱动器之前消耗掉。在一个实施例中，全零 2 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。

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。分配由转向长度参数指定的字节数目，以便允许客户机中的 MDDI_Data 线路驱动器在主机中的线路驱动器被启用以前被禁用。在转向 2 的位 0 期间，所述客户机禁用其 MDDI_Data 线路驱动器，并且主机启用其输出端并且在转向 2 的最后位期间把

MDDI_Data0 驱动为逻辑 0。MDDI_Stb 信号表现得如同 MDDI_Data0 在整个转向 2 周期期间处于逻辑 0 电平。转向 2 的建议长度是显示器中 MDDI_Data 驱动器禁用输出所需的字节数目加上所述往返行程延迟。上面给出了转向 2 设置的说明。

反向数据数据包字段包含从客户机传送到主机的一系列数据数据包。如前所述，发送填充符数据包以便填充其它数据包类型未使用的剩余空间。

驱动器重新启用字段使用等于逻辑零的 1 字节来确保所有 MDDI_Data 信号在下一数据包的数据包长度字段之前被重新启用。

F. 关于显示器能力数据包

在一个实施例中，协议版本字段使用 2 字节来指定由客户机使用的协议版本。初始版本被设置为等于零，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。显示数据速率能力字段（2 字节）指定显示器在接口的正向链路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。目前这是通过选择位 0、位 1 或者位 2 来分别选择正向链路上的 II 型、III 型或者 IV 型模式来表明的，而位 3、位 4 或位 5 则分别选择反向链路上的 II 型、III 型或者 IV 型模式；位 6 和 7 被保留并且设置为零。位图宽度和高度字段（2 字节）以像素为单位指定位图的宽度和高度。

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，那么显示器不支持所述单色格式。

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。

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 15 至 12 被保留以供将来使用并且通常设置为零。

Y Cr Cb 能力字段（2 字节）指定可以按照 Y Cr Cb 格式显示的分辩率的位数。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。

显示器特征能力指示符字段使用 4 字节，它包含一组用于表明显示器中所支持的特定特征的标志。设置为 1 的位表明支持所述能力，并且设置为零的位表明不支持所述能力。位 0 的值表明是否支持位图块传送数据包（数据包类型 71）。位 1、2 和 3 的值分别表明是否支持位图区域填充数据包（数据包类型 72）、位图图案填充数据包（数据包类型 73）或者通信链路数据通道数据包（数据包类型 74）。位 4 的值表明显示器是否有能力使一种颜色透明，而位 5 和 6 的值分别表明显示器能否接受打包格式的视频数据或者音频数据，而位 7 的值表明显示器能否从照相机发送反向链路视频流。位 11 和 12 的值分别表明客户机正在与指示设备通信并且能发送和接收指示设备数据数据包，还是正在与键盘通信并且能发送和接收键盘数据数据包的情况。位 13 至 31 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识，并且通常设置为等于零。

显示视频帧速率能力字段（1 字节）指定每秒帧中显示器的最大视频帧更新能力。主机可以选择以慢于字段中指定的值的速率来更新所述图像。

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弹性缓冲器的深度。

音频通道能力字段（2 字节）包含一组标志，用于表明显示器（客户机）支持哪些音频通道。设置为 1 的位表明通道被支持，而设置为

零的位表明通道不被支持。将位的位置分配给不同的通道，例如位位置 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 15 目前保留以供将来使用并且通常设置为零。

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设置为 '1' 表明该特定的采样速率被支持，而把该位设置为 '0' 表明该采样速率不被支持。

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。

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地支持最少 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，其中位位置 0、1、2、3、4、5、6、7 和 8 例如分别用于代表 8,000、16,000、24,000、32,000、40,000、48,000、11,025、22,050 和 44,100 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被支持，而把该位设置为 '0' 表明该采样速率不被支持。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。

内容保护类型字段 (2 字节) 包含一组标志，用于表明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，而位位置 2 至 15 被保留以供其它保护方案之用，或者可利用，因此它们目前被设置为零。

G. 关于显示器请求和状态数据包

反向链路请求字段（3 字节）指定显示器在下一子帧中在反向链路上用于向主机发送信息所需要的字节数目。

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数据包时，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。

能力改变字段使用 1 字节来表明显示器能力的改变。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，那么自从发送出最后的显示器能力数据包以后，能力没有改变。然而，当位[7:0]等于 1 至 255 时，能力已经改变。检查显示器能力数据包以便确定新的显示特性。

H. 关于位块传送数据包

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，以便使 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。

I. 关于位图区域填充数据包

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。所述格式与视频流数据包中的同一字段相同。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。

J. 关于位图图案填充数据包

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。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。所述格式与视频流数据包中的同一字段相同。

参数 CRC 字段（2 字节）包含从数据包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，就丢弃整个数据包。图案像素数据字段包含原始视频信息，用于说明由视频数据格式描述符指定的填充图案。数据被打包为字节，并且每一行的第一像素必须按字节对准。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）只包含图案像素数据的 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。

K. 通信链路数据通道数据包

参数 CRC 字段（2 字节）包含从数据包长度到数据包类型的所有字节的 16 位 CRC。如果这一 CRC 未能通过校验，就丢弃整个数据包。

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。

通信链路数据 CRC 字段（2 字节）只包含通信链路数据的 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。

L. 关于接口类型切换请求数据包

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 I 型模式，值 2 意味着切换至 II 型模式，值 3 意味着切换至 III 型模式，而值 4 意味着切换至 IV 型模式。值‘0’以及 5 至 7 被保留以供将来标识可选模式或者模式组合之用。

M. 关于接口类型确认数据包

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，反过来，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，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 I 型、II 型、III 型和 IV 型模式。值 5 至 7 被保留以便根据需要用作模式的可选标识。

N. 关于执行类型切换数据包

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 I 型、II 型、III 型和 IV 型模式。这些位的值 0 以及 5 至 7 的使用被保留以供将来使用。

O. 关于正向音频通道启用数据包

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。

P. 关于反向音频采样速率数据包

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。

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 μ -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。

Q. 关于数字内容保护开销数据包

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）。目前不指定 2 至 255 的值范围，但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。

R. 关于透明色彩启用数据包

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。

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数据包中的同一字段相同。

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。

S. 关于往返行程延迟测量数据包

在一个实施例中,参数 CRC 字段(2字节)包含从数据包长度到数据包类型的所有字节的 16 位 CRC。如果这一 CRC 未能通过校验,就丢弃整个数据包。

全零字段(1字节)包含零,以便确保所有 MDDI_Data 信号在第一保护时间周期期间禁用线路驱动器之前处于零状态。

保护时间 1 字段(8字节)用来允许主机中的 MDDI_Data 线路驱动器在客户机(显示器)中的线路驱动器启用以前被禁用。在保护时间 1 的位 0 期间,所述主机禁用 MDDI_Data 线路驱动器,并且在保护时间 1 的最后位之后,显示器立即启用其线路驱动器。

测量周期字段是 512 字节窗口,用于允许显示器以正向链路上使用的一半数据速率来利用 0xff、0xff、0x0 来作出响应。这一速率对应于反向链路速率除数为 1。在测量周期开始时,显示器立即返回这一响应。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的时刻,主机接收这一响应。所述显示器中的 MDDI_Data 线路驱动器在来自显示器的 0xff、0xff、0x00 响应之前和之后被立即禁用。

保护时间 2 字段(8字节)中的值允许客户机 MDDI_Data 线路驱动器在启用主机中的线路驱动器以前被禁用。保护时间 2 始终有,但是只有当往返行程延迟处于最大数量时才需要,所述往返行程延迟量可以在测量周期中被测量。在保护时间 2 的位 0 期间,所述客户机禁用其线路驱动器,并且在保护时间 2 的最后位之后,主机立即启用其线路驱动器。

驱动器重新启用字段(1字节)被设置为等于零,以便确保所有

MDDI_Data 信号在下一数据包的数据包长度字段之前被重新启用。

T. 关于正向链路偏离校准数据包

在一个实施例中，参数 CRC 字段（2 字节）包含从数据包长度到数据包类型的所有字节的 16 位 CRC。如果这一 CRC 未能通过校验，就丢弃整个数据包。

校准数据序列字段包含 512 字节数据序列，用于让 MDDI_Data 信号在每一数据周期翻转。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，所述显示器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，基于发送这个数据包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：

I 型 - 0xaa, 0xaa……或者 0x55, 0x55……

II 型 - 0xcc, 0xcc……或者 0x33, 0x33……

III 型 - 0xf0, 0xf0……或者 0x0f, 0x0f……

IV 型 - 0xff, 0x00, 0xff, 0x00……或者 0x00, 0xff, 0x00, 0xff……

图 62A 和 62B 中分别示出了 I 型和 II 型接口的可能的 MDDI_Data 和 MDDI_Stb 波形的实例。

XVII. 结束语

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。

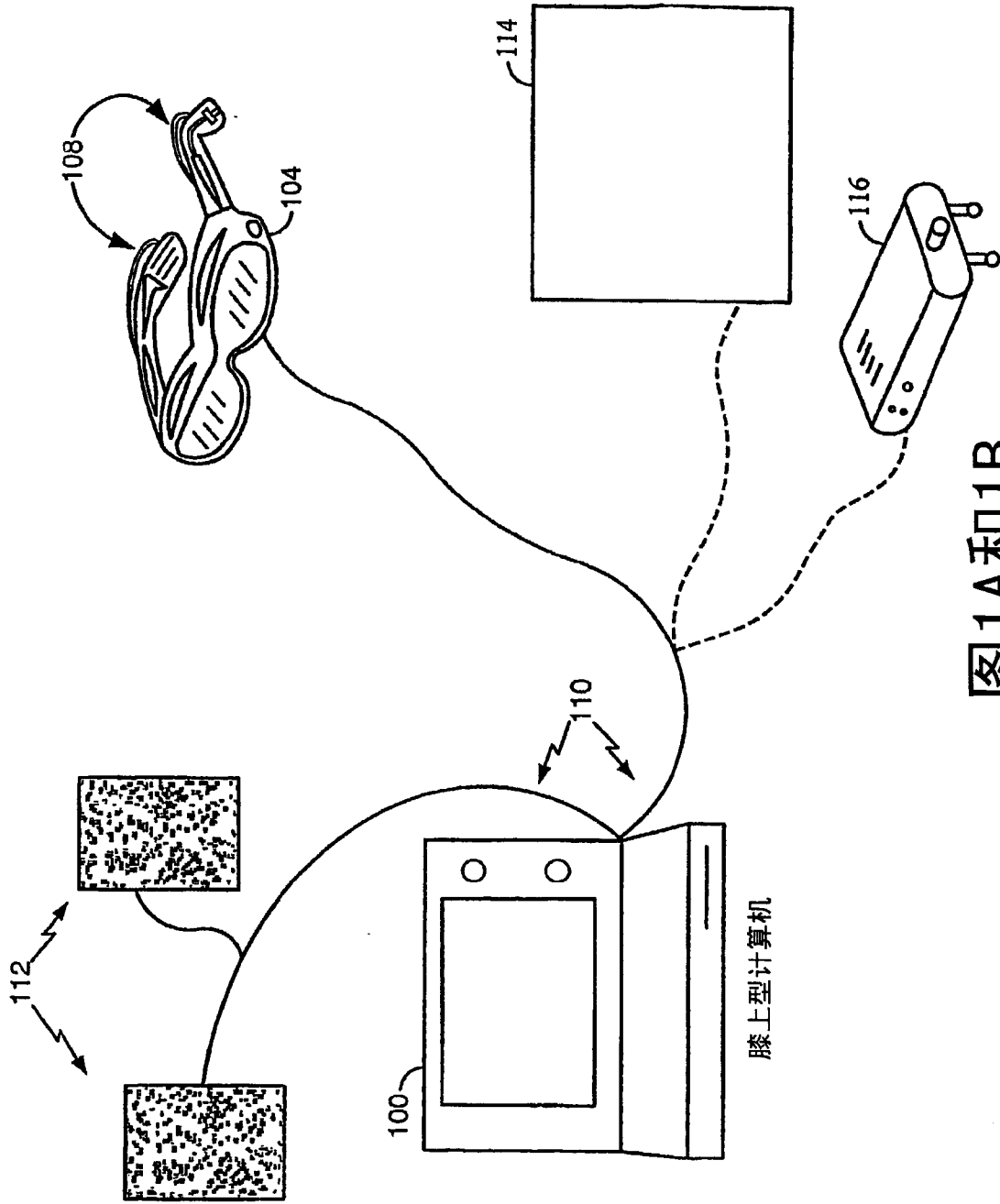


图1A和1B

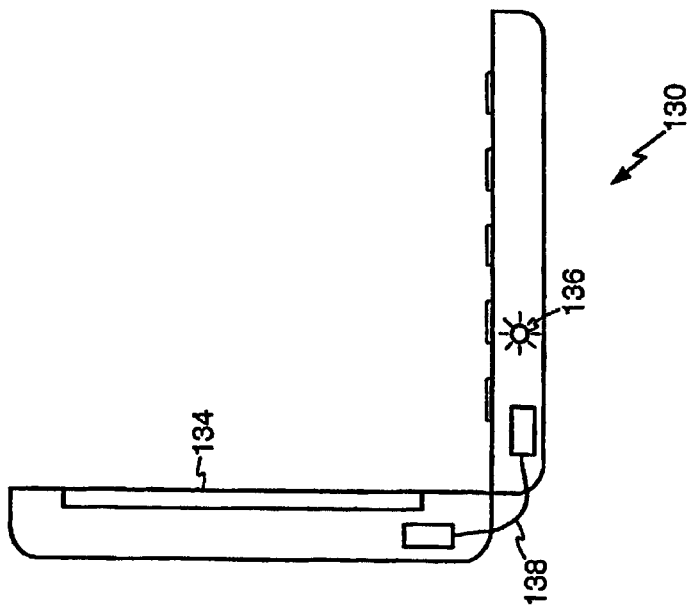


图1C

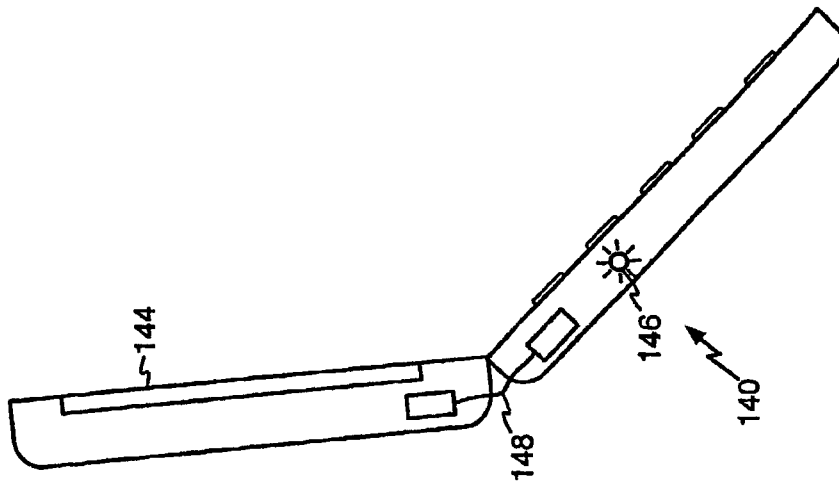


图1D

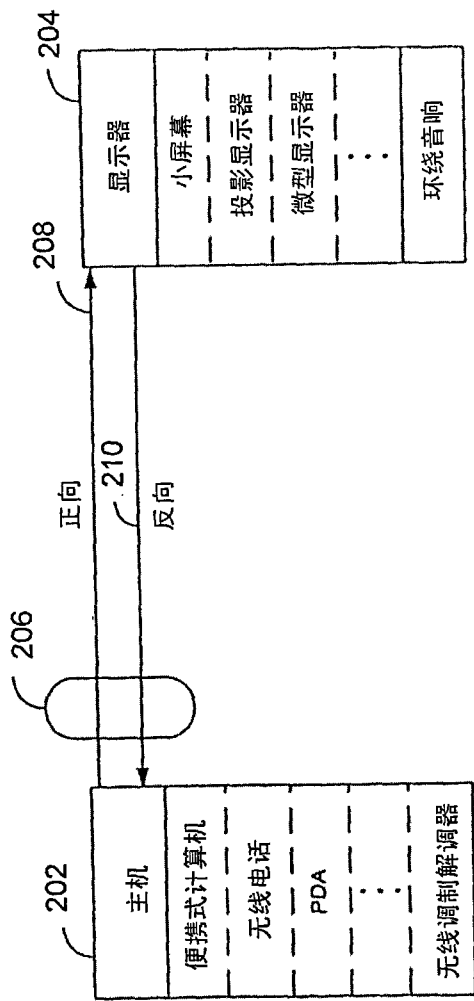


图2

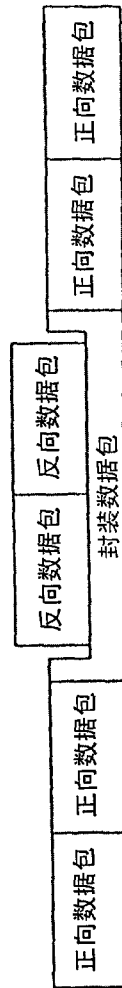


图3

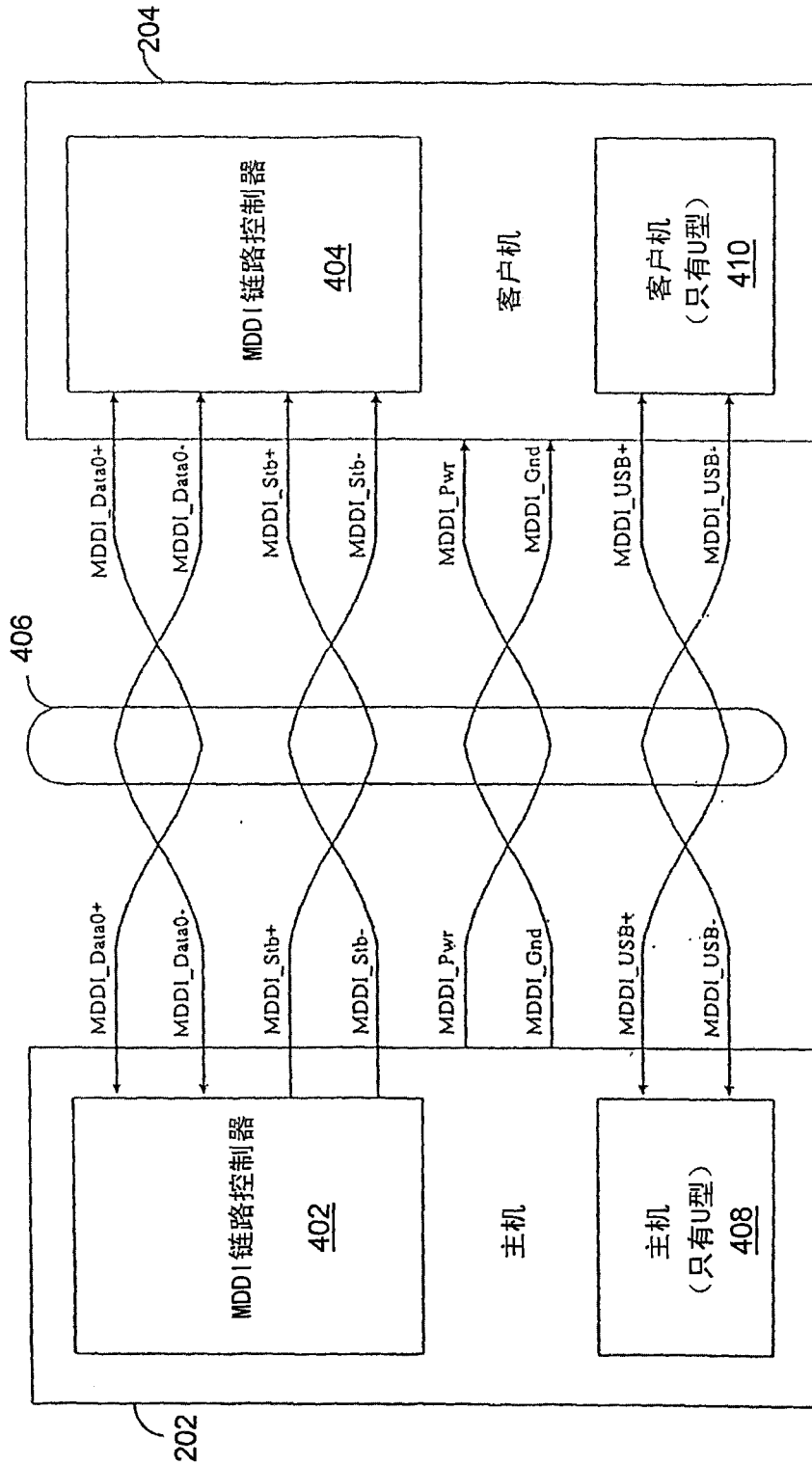


图4

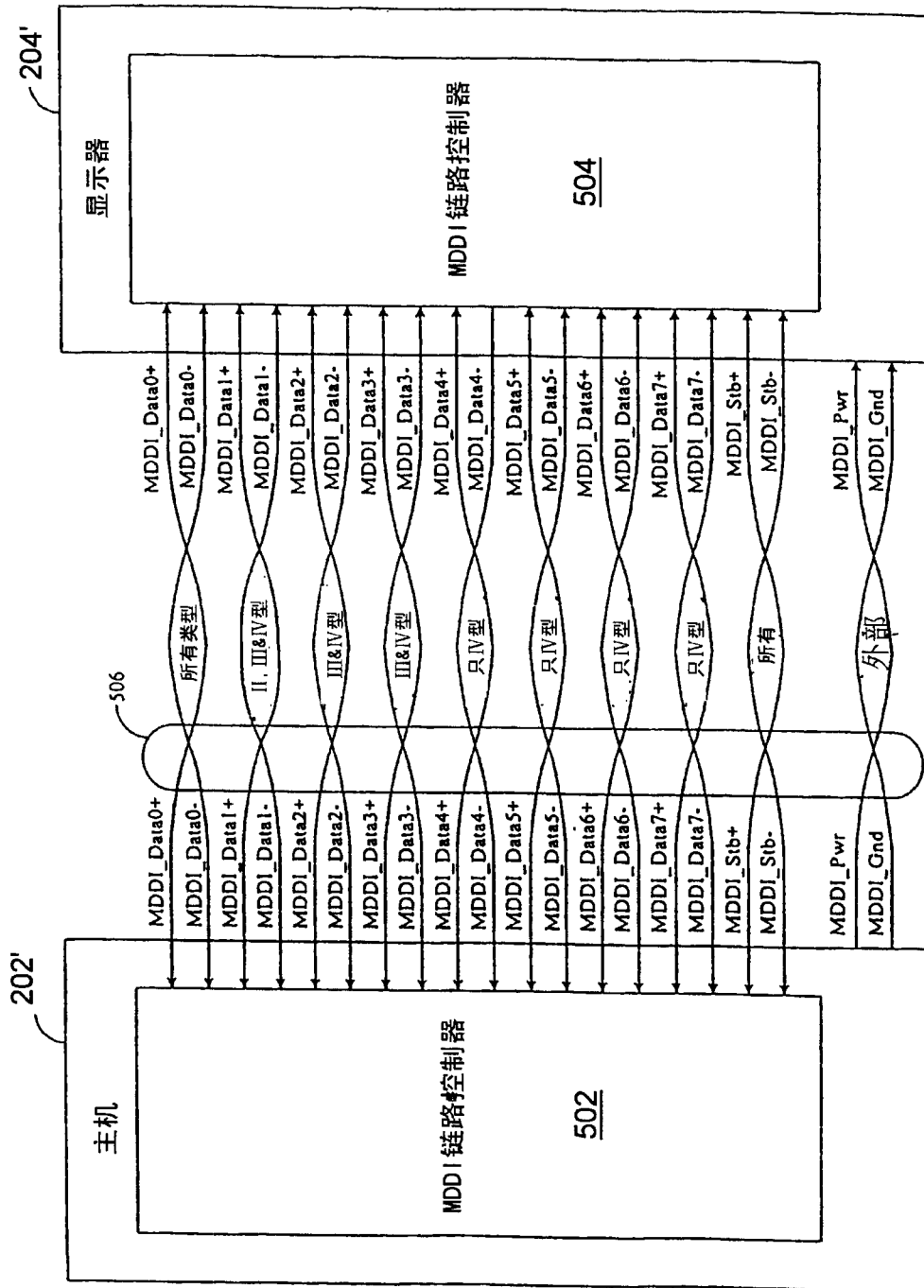


图5

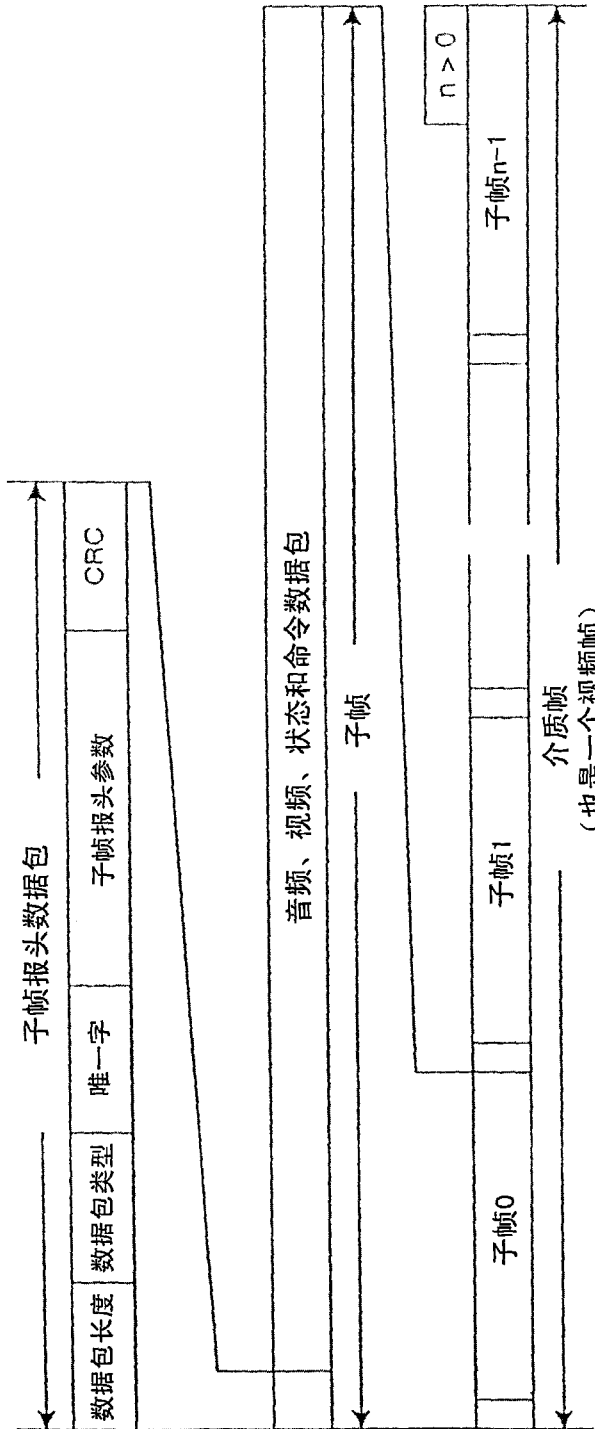


图6

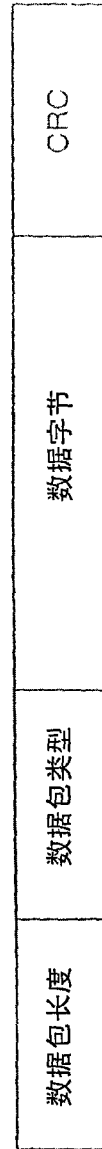


图7

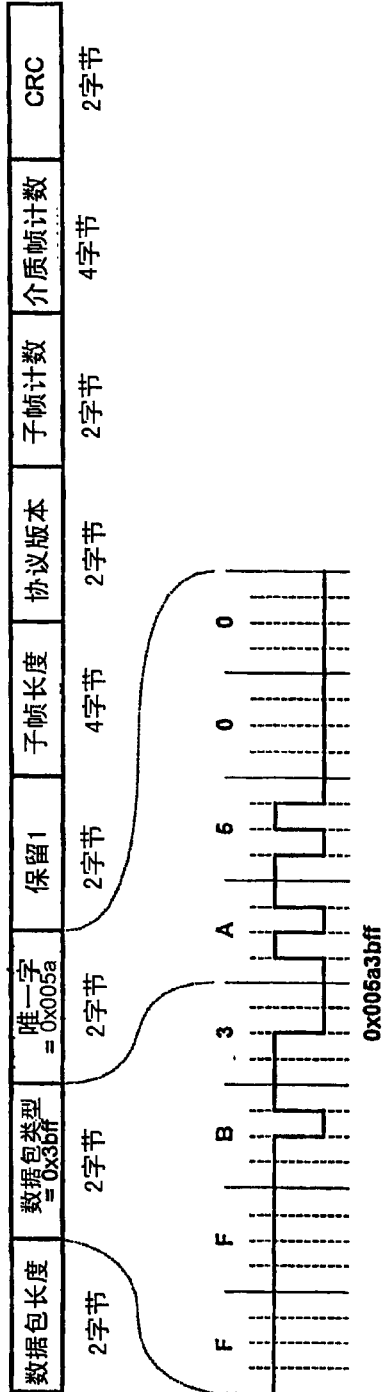


图8

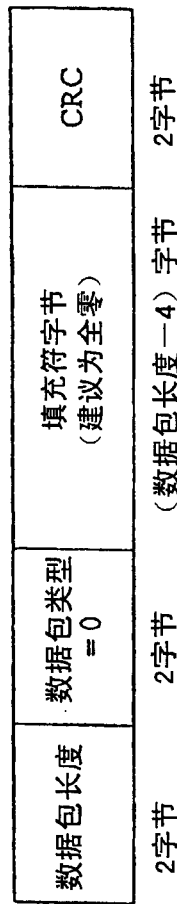


图9

视频流数据包

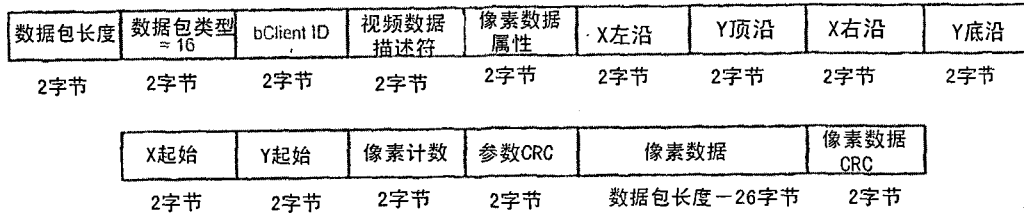
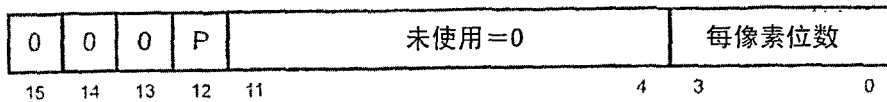
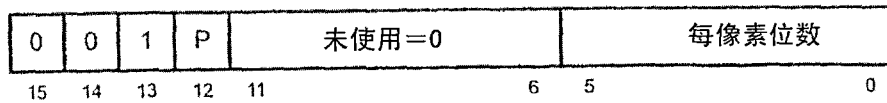


图10

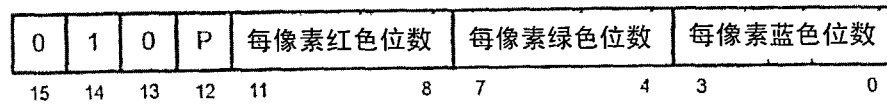
视频数据格式描述符—单色



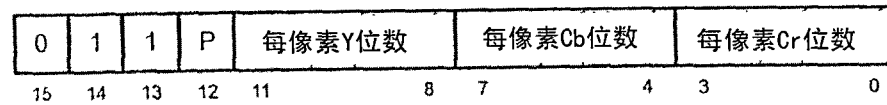
视频数据格式描述符—使用色彩映射的色彩



视频数据格式描述符—色彩、原始RGB



视频数据格式描述符—色彩分量视频, YCbCr



视频数据格式描述符—间隔 (Bayer)

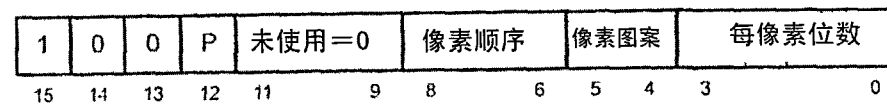


图11

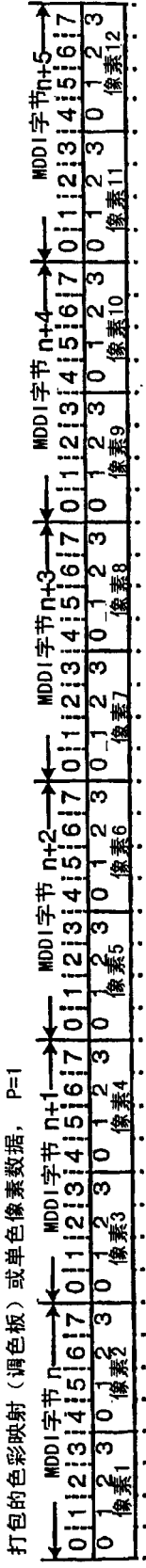
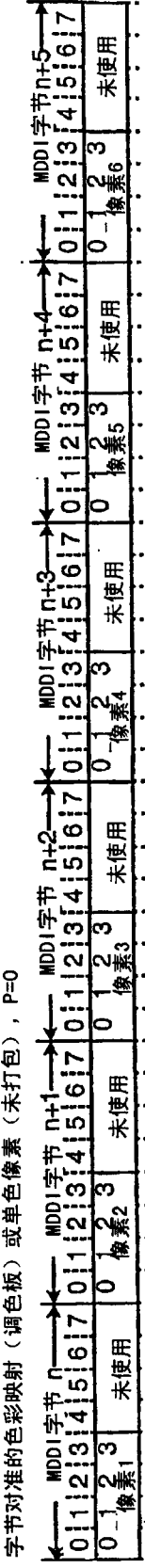
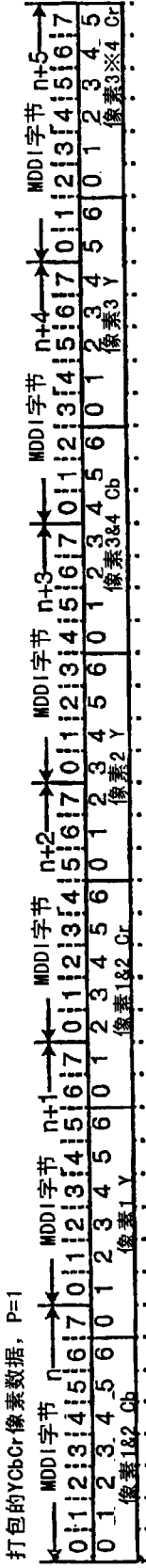
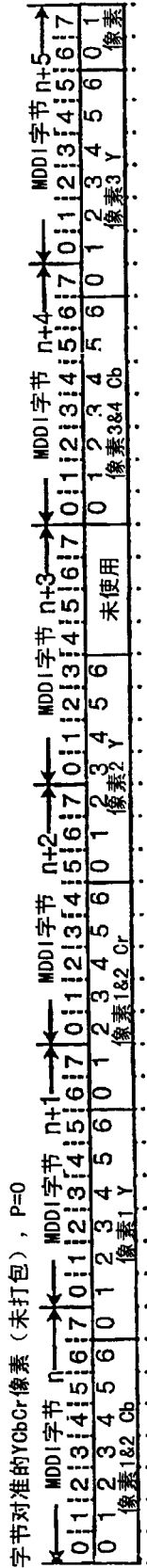
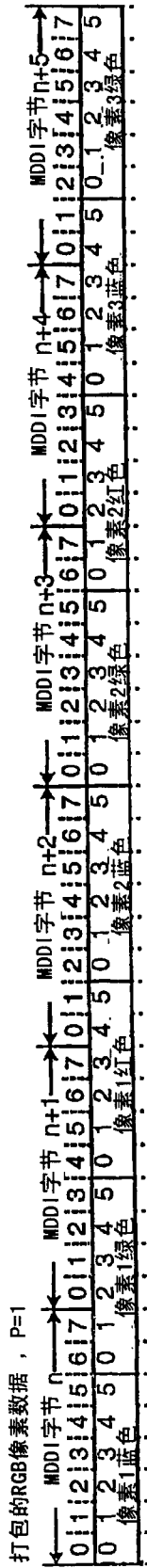
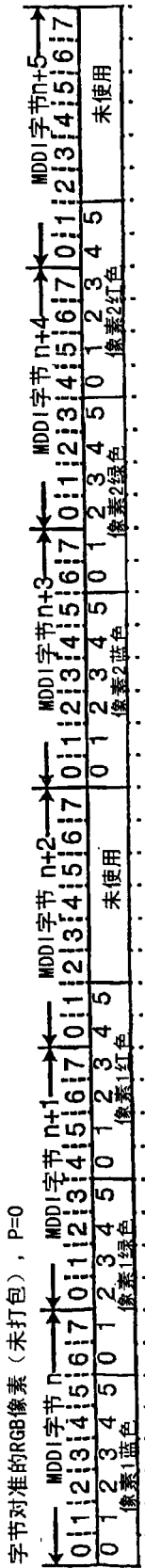


图12

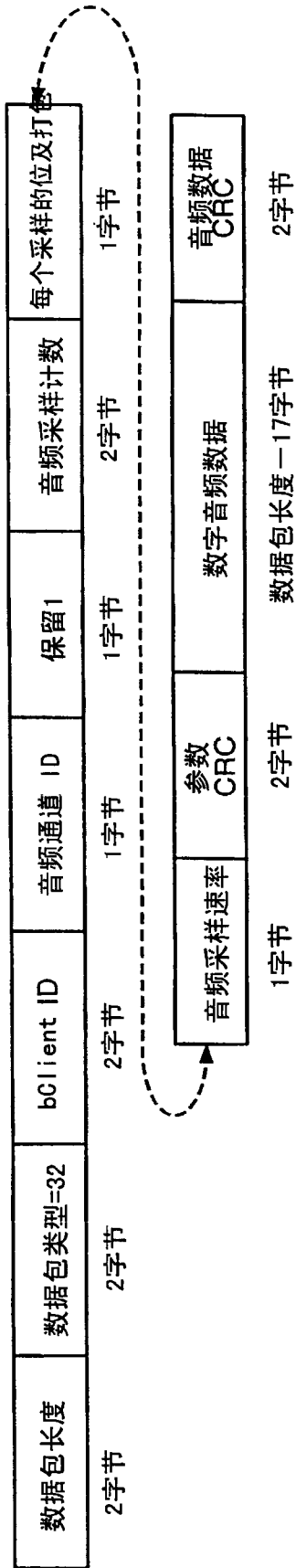


图13

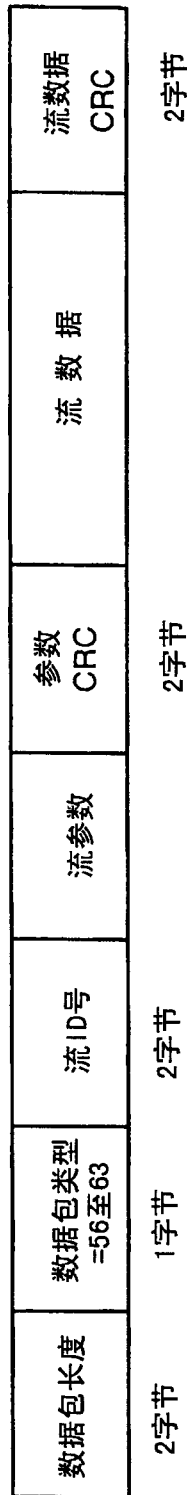


图15

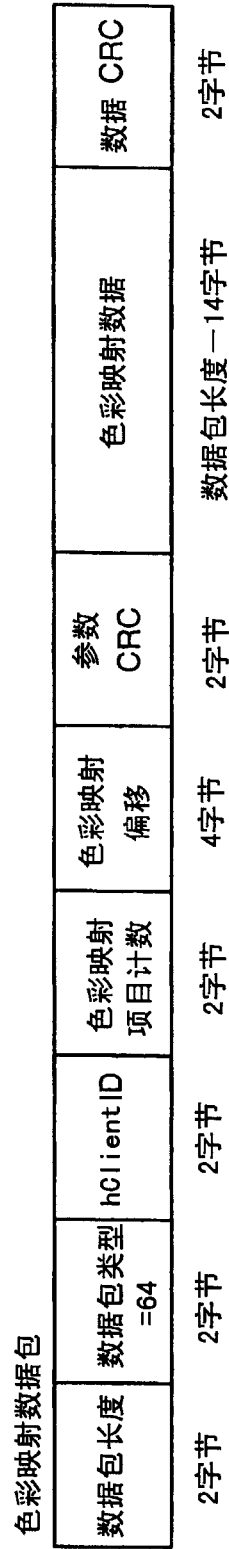
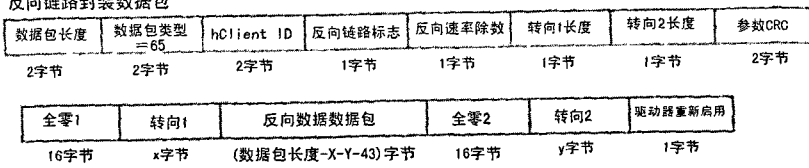


图16

反向链路封装数据包



显示器能力数据包



图17

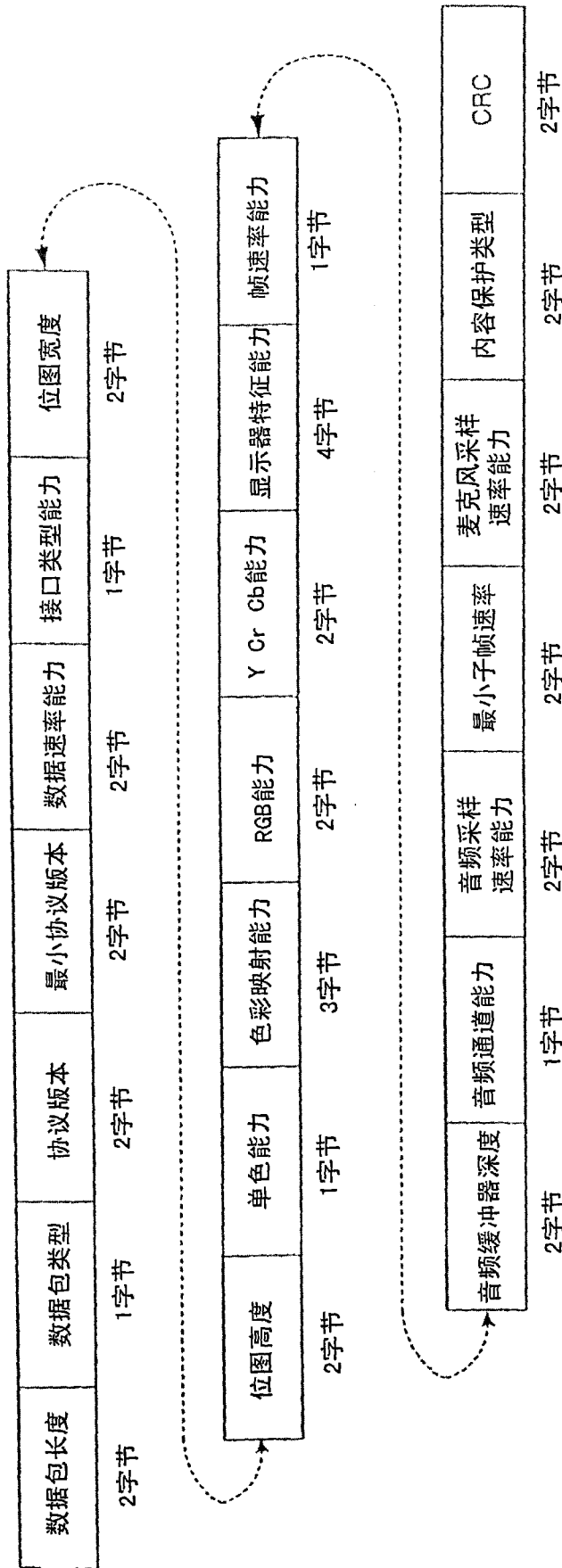


图18

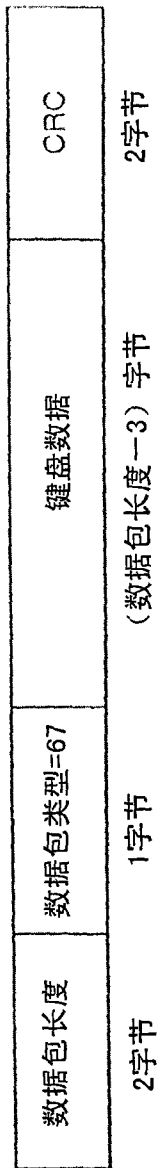


图19

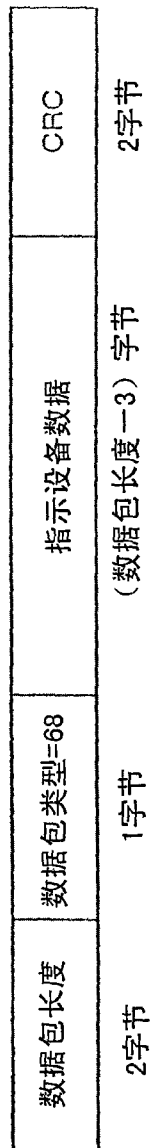


图20

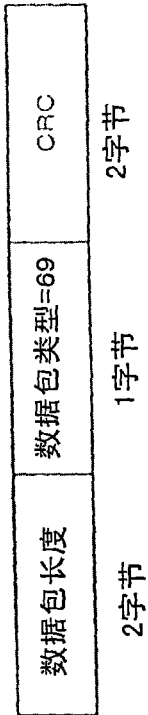


图21

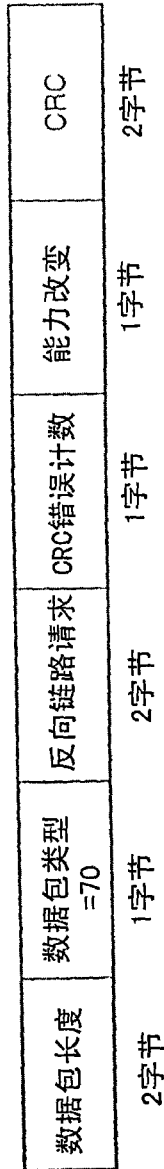


图22

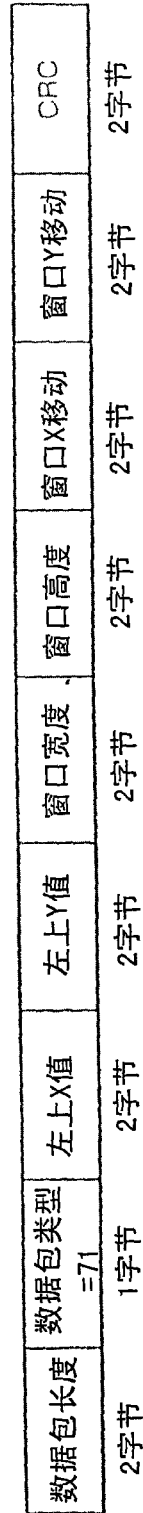


图23



图24

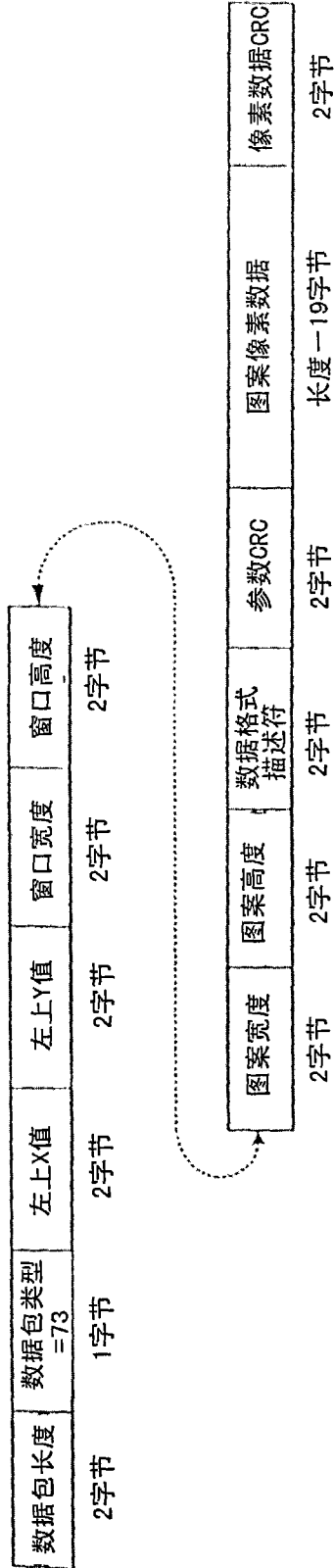


图25

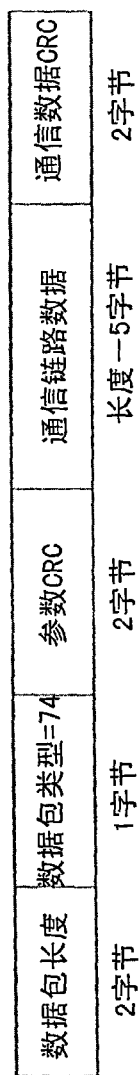


图26

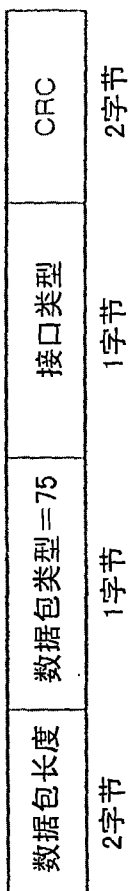


图27

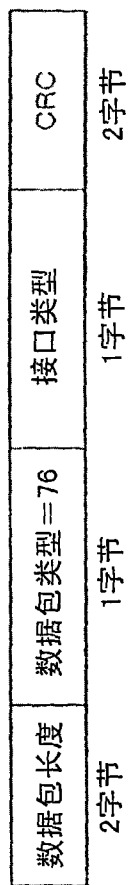


图28

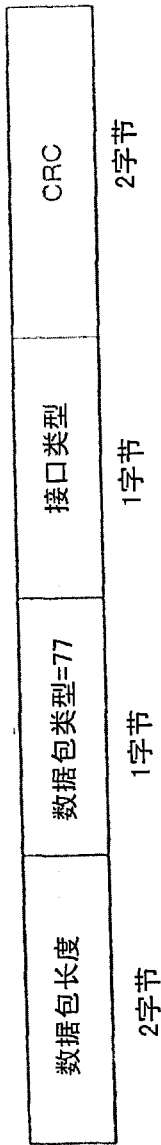


图29

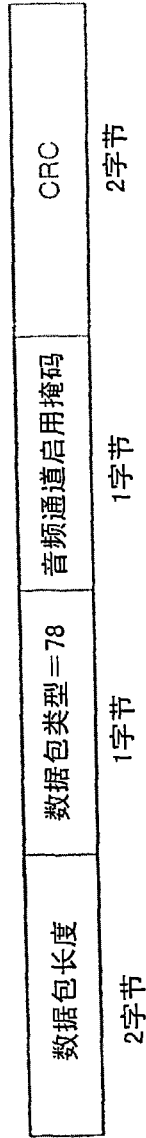


图30

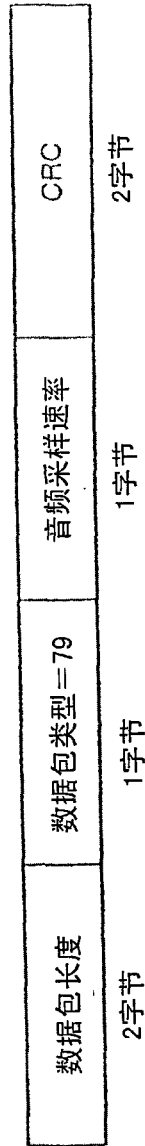


图31

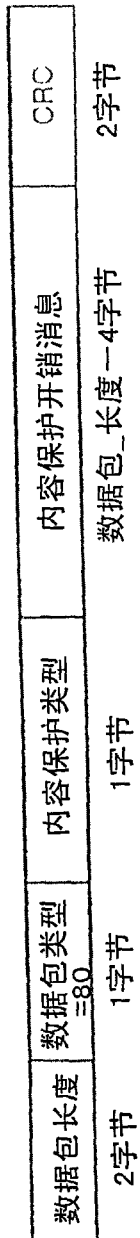


图32

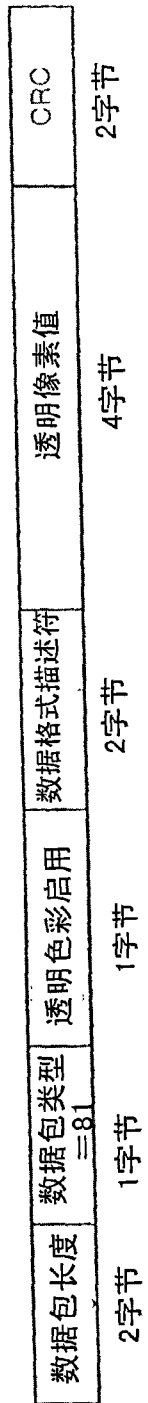


图33

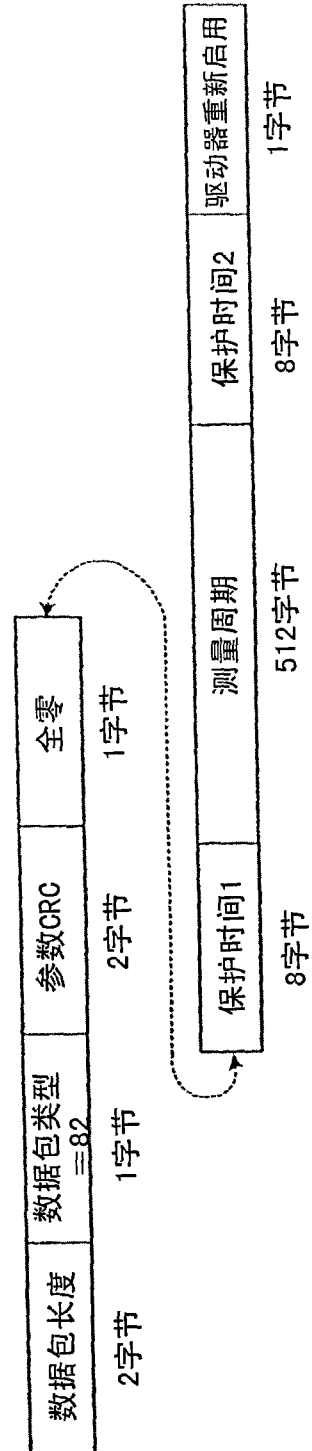


图34

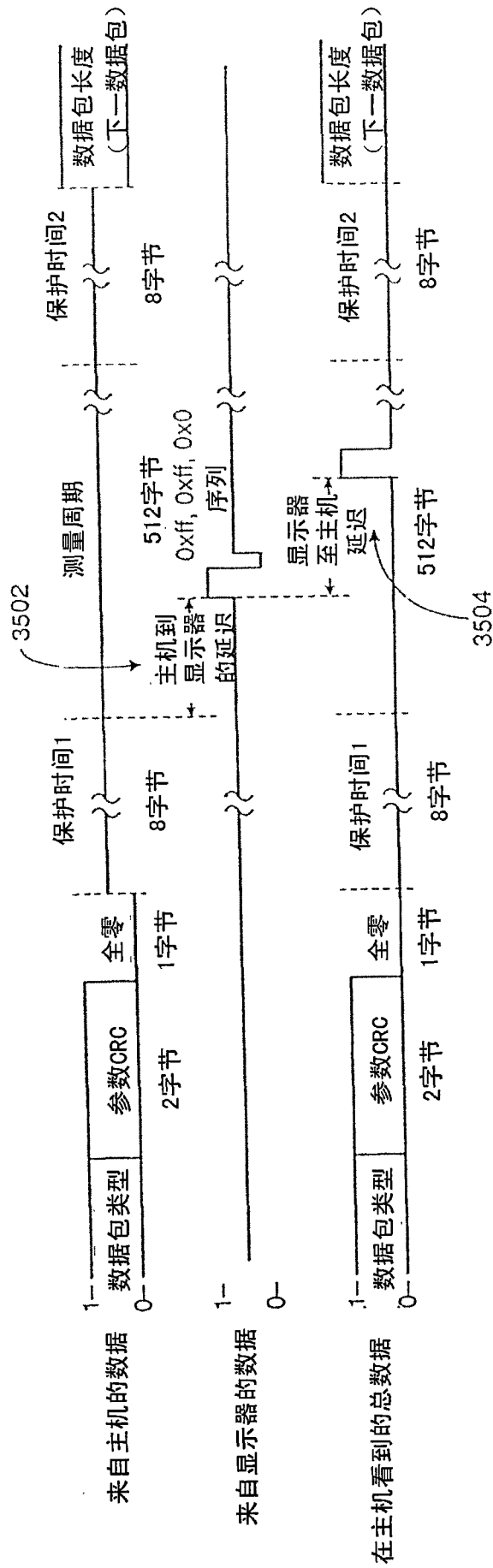


图35

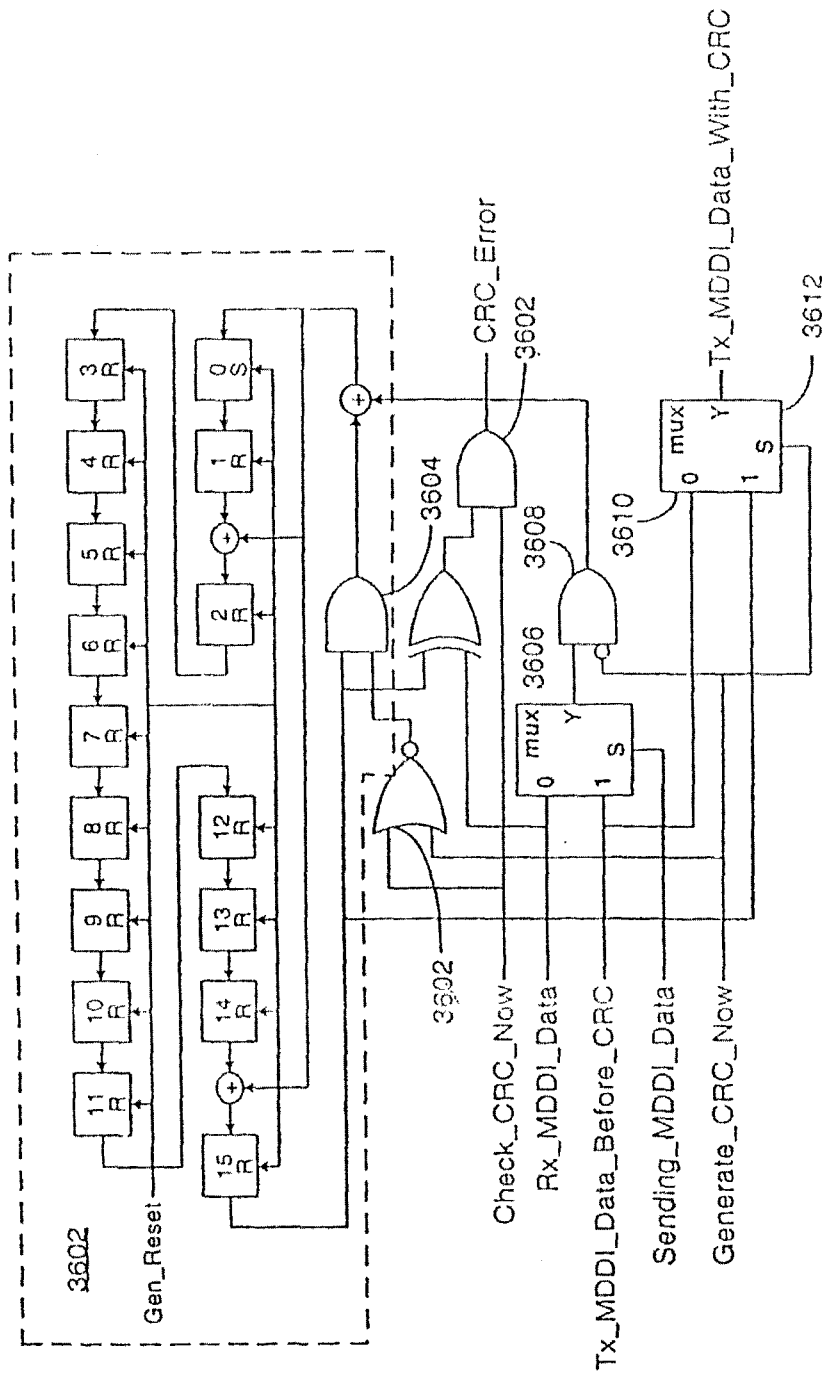


图 36

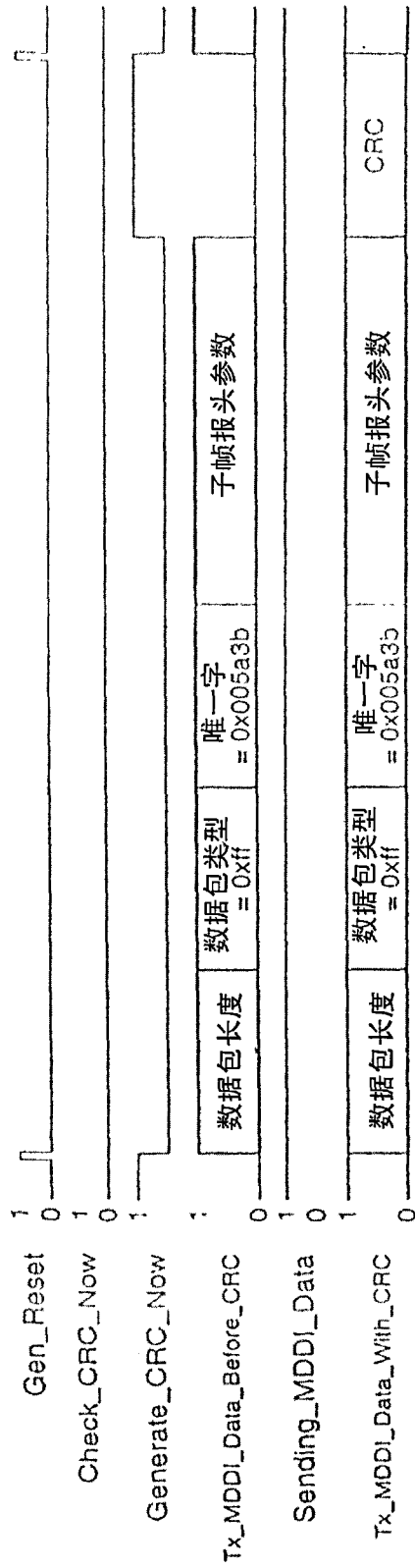


图37A

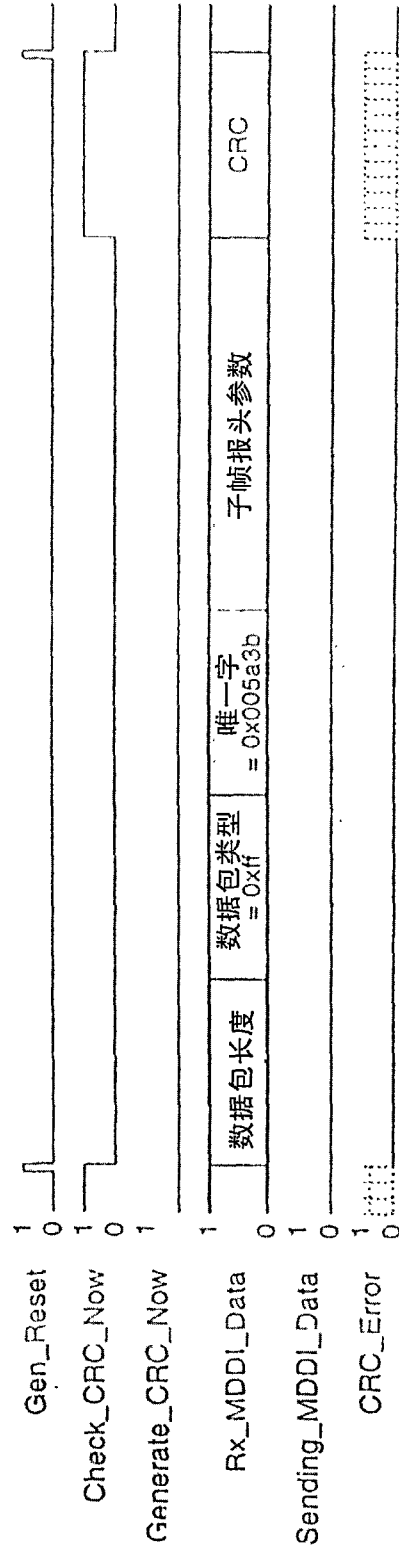


图37B

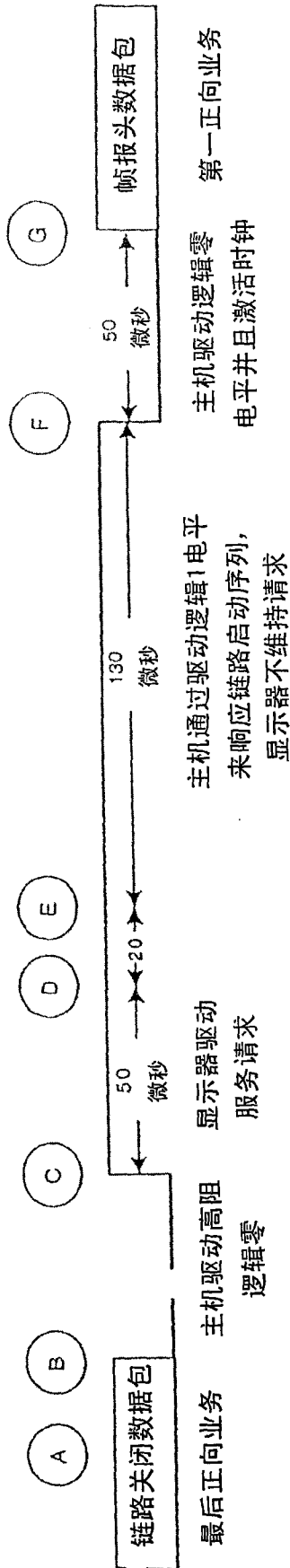


图38

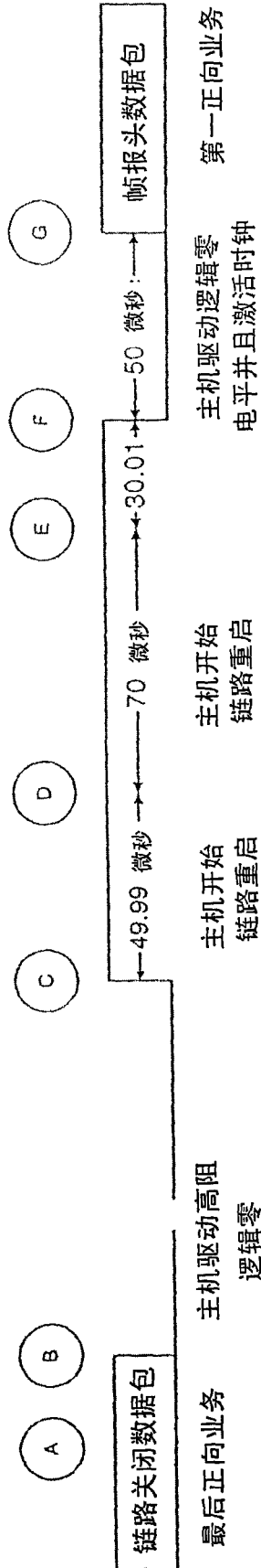


图39

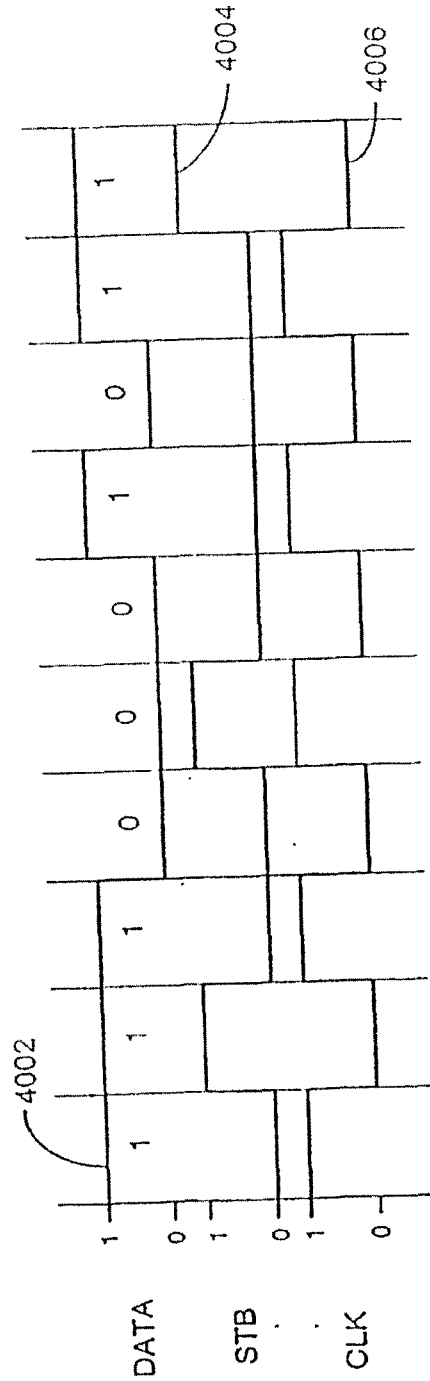


图40

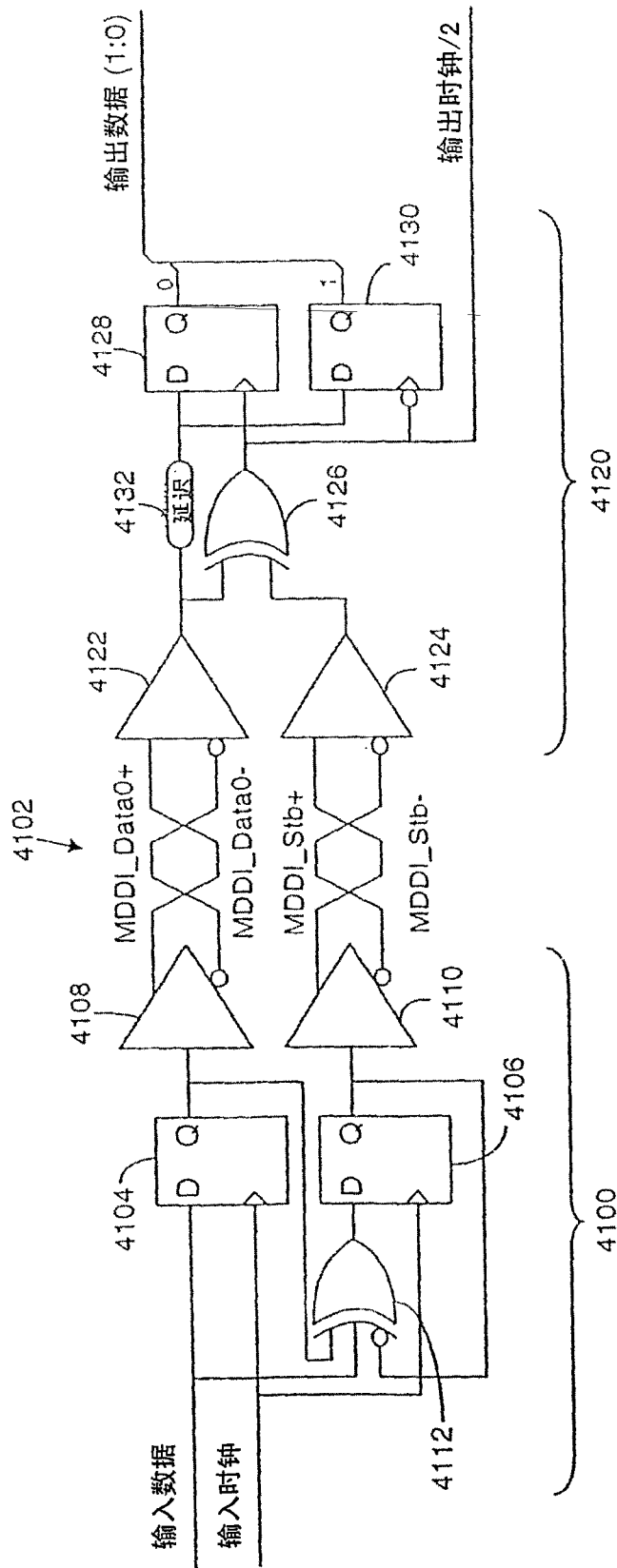


图41

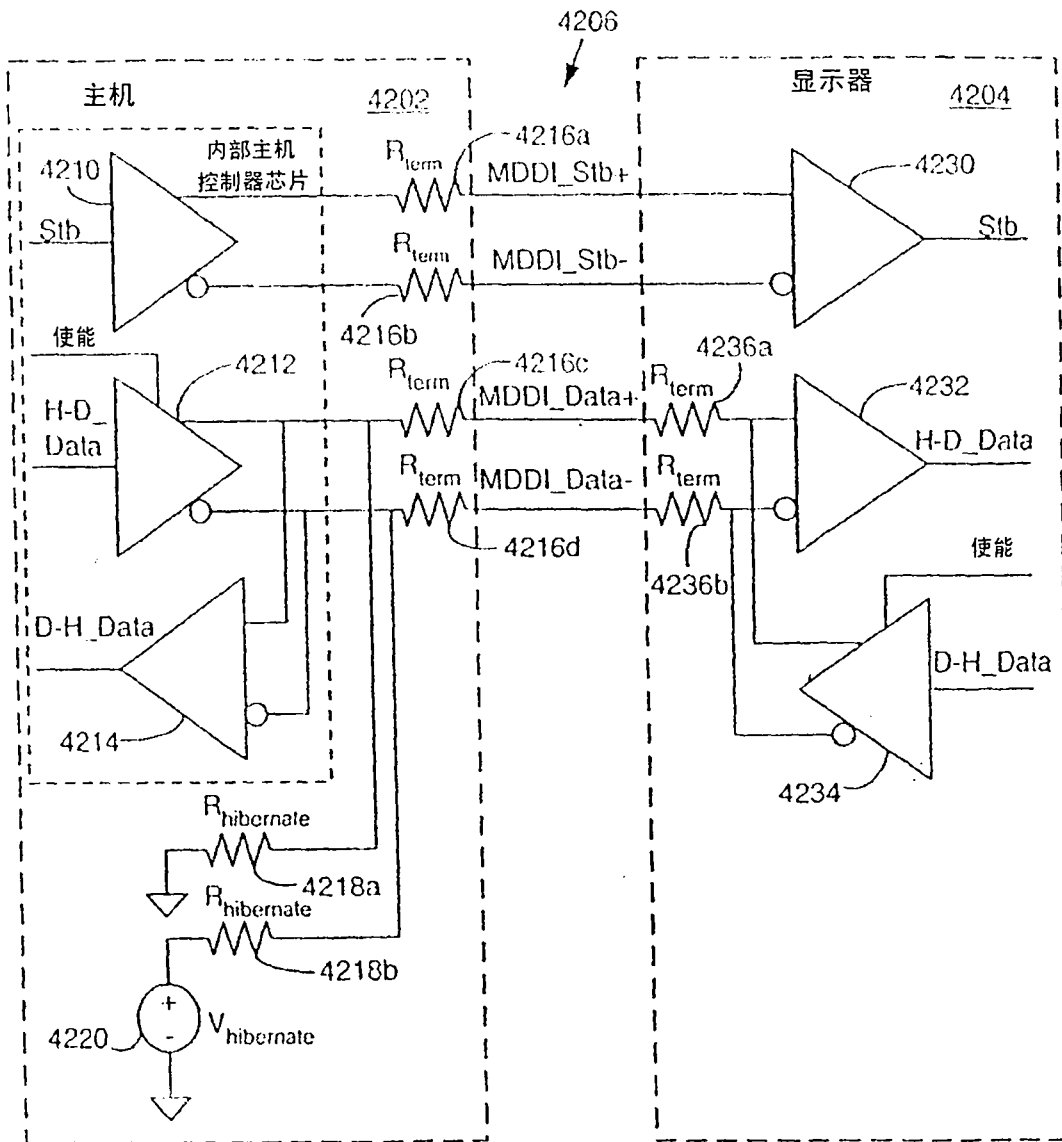


图42

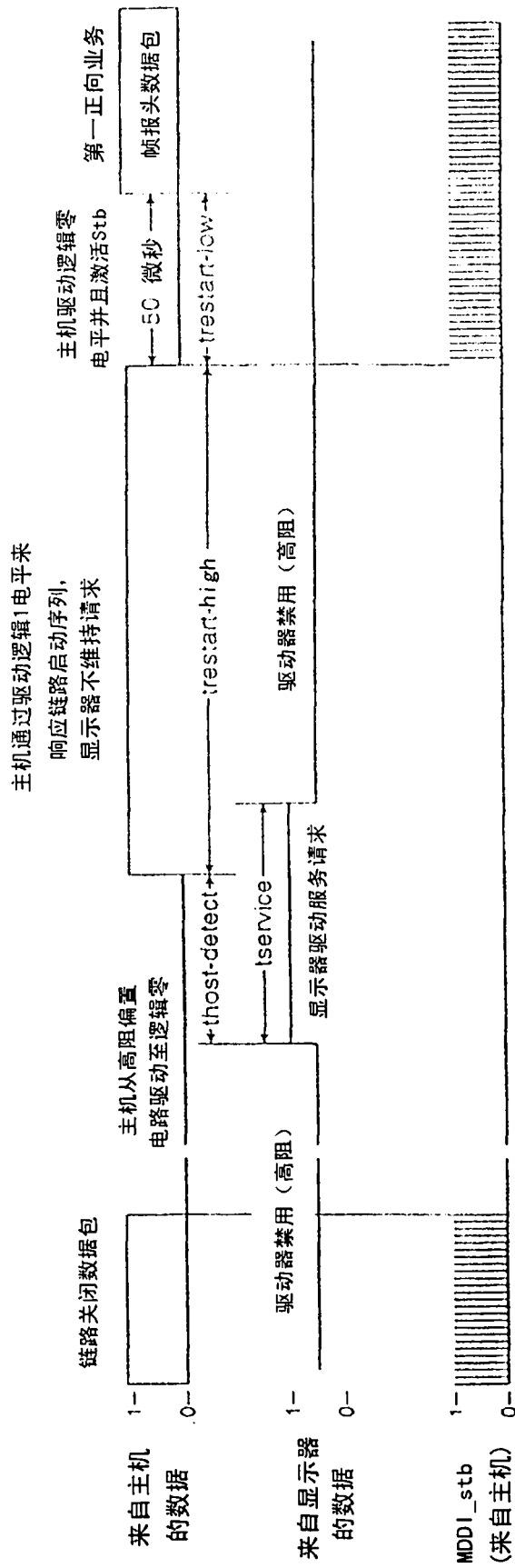


图 43

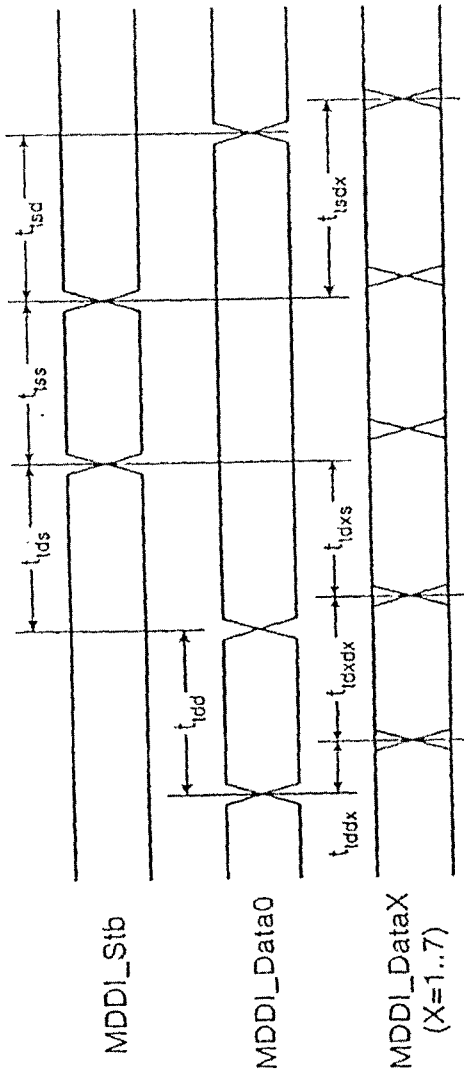


图44

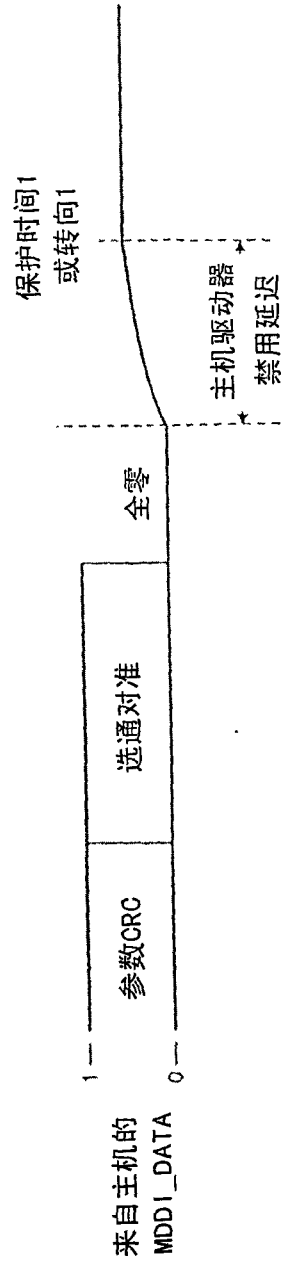


图45

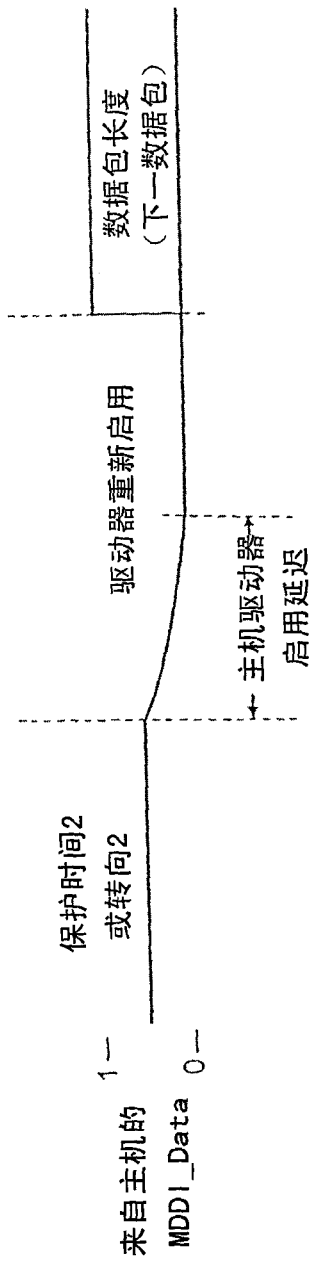


图46

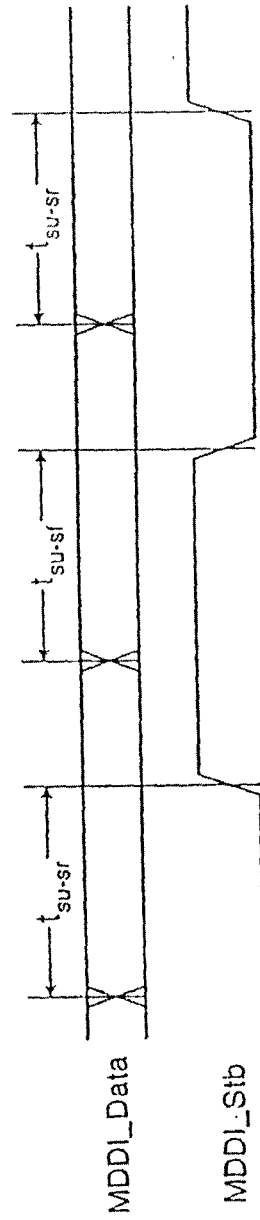


图47

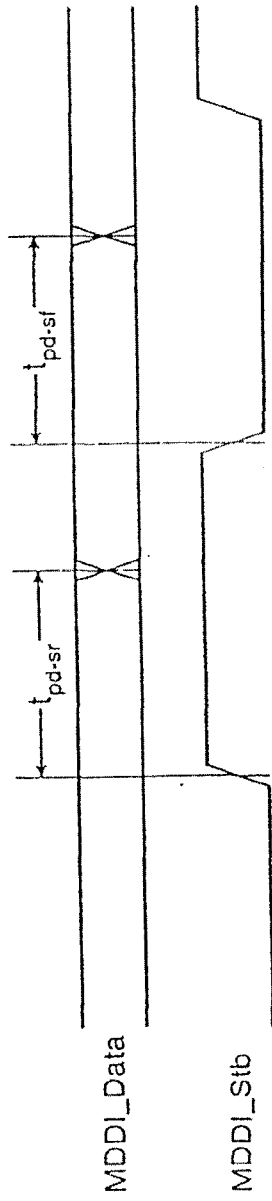
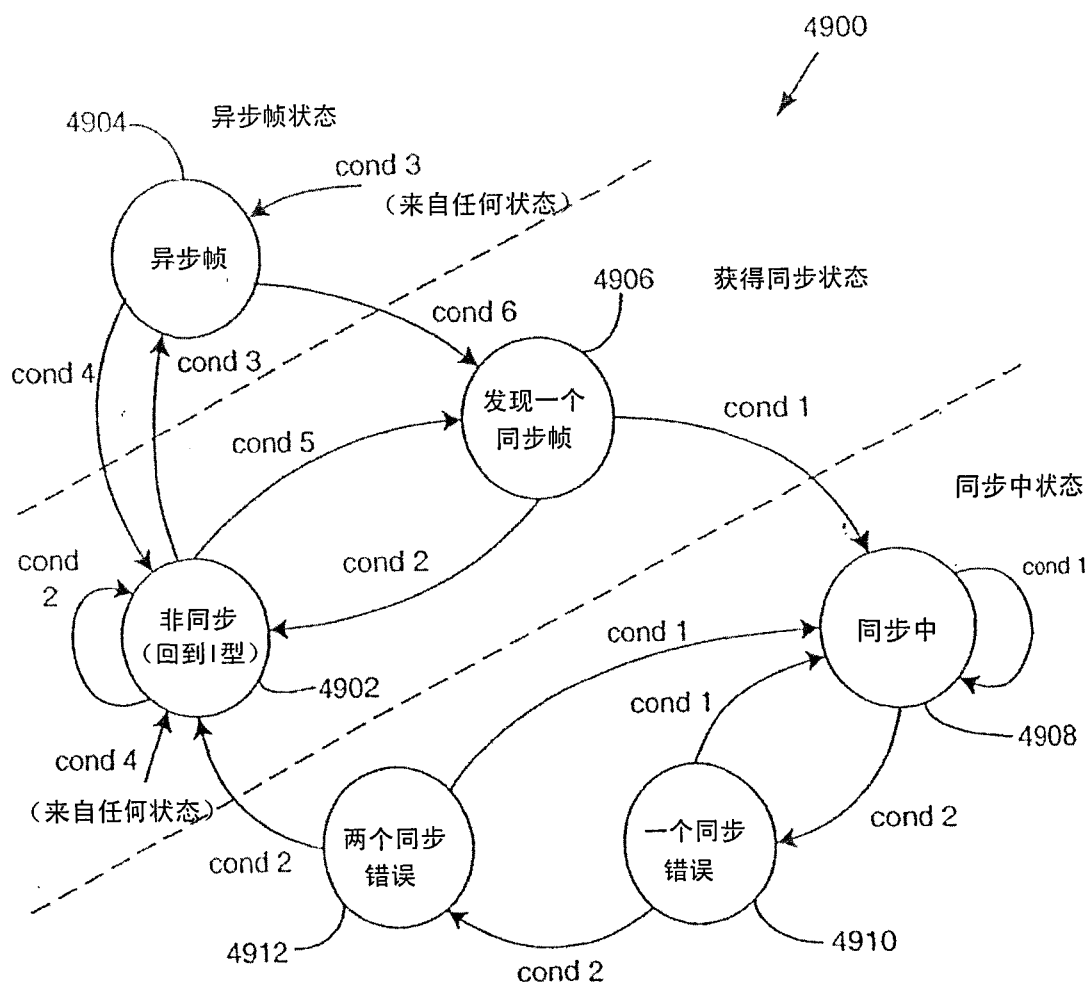


图 48



cond 1 = 子帧报头数据包&处于帧边界的良好CRC, 帧长度>0

cond 2 = 没有同步模式 (pattern) 或在帧边界为坏CRC

cond 3 = 发现同步模式, 帧长度=0

cond 4 = 接收到链路关闭数据包

cond 5 = 发现同步模式, 帧长度>0

cond 6 = 帧报头数据包&良好CRC, 帧长度>0

图49

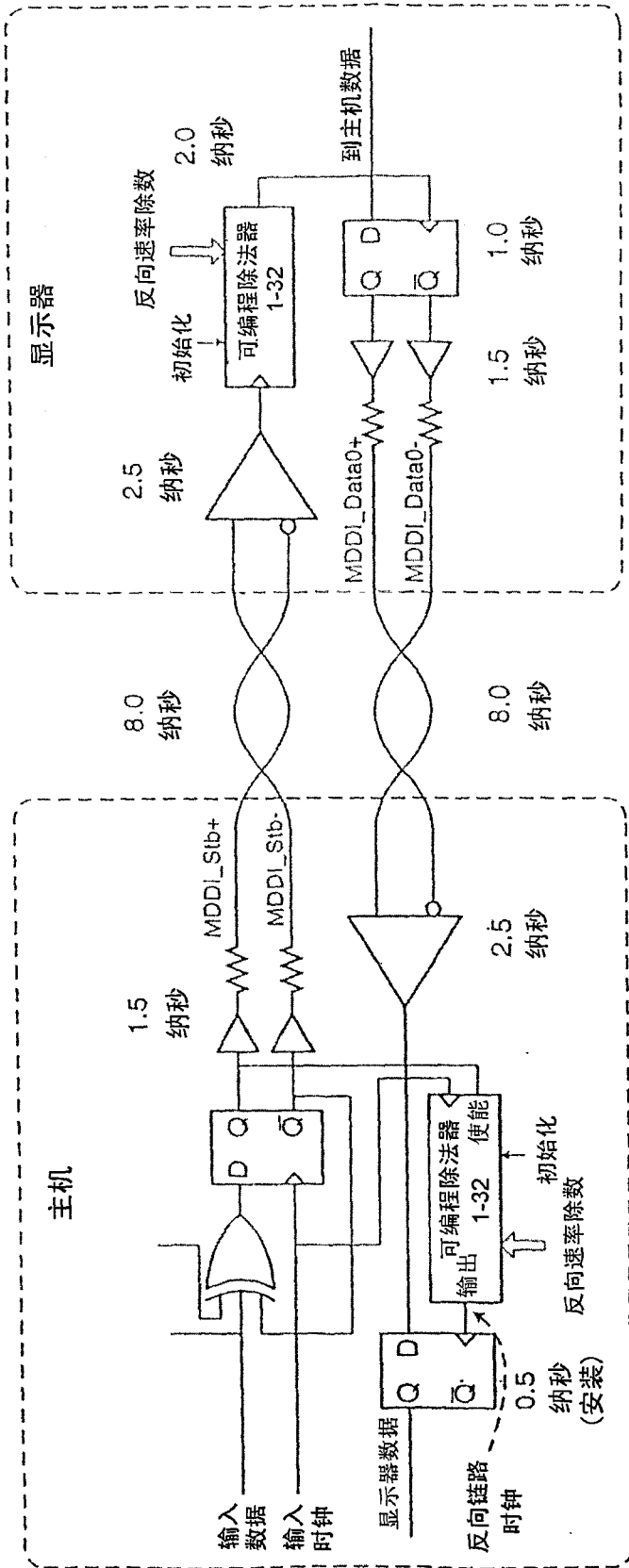


图50

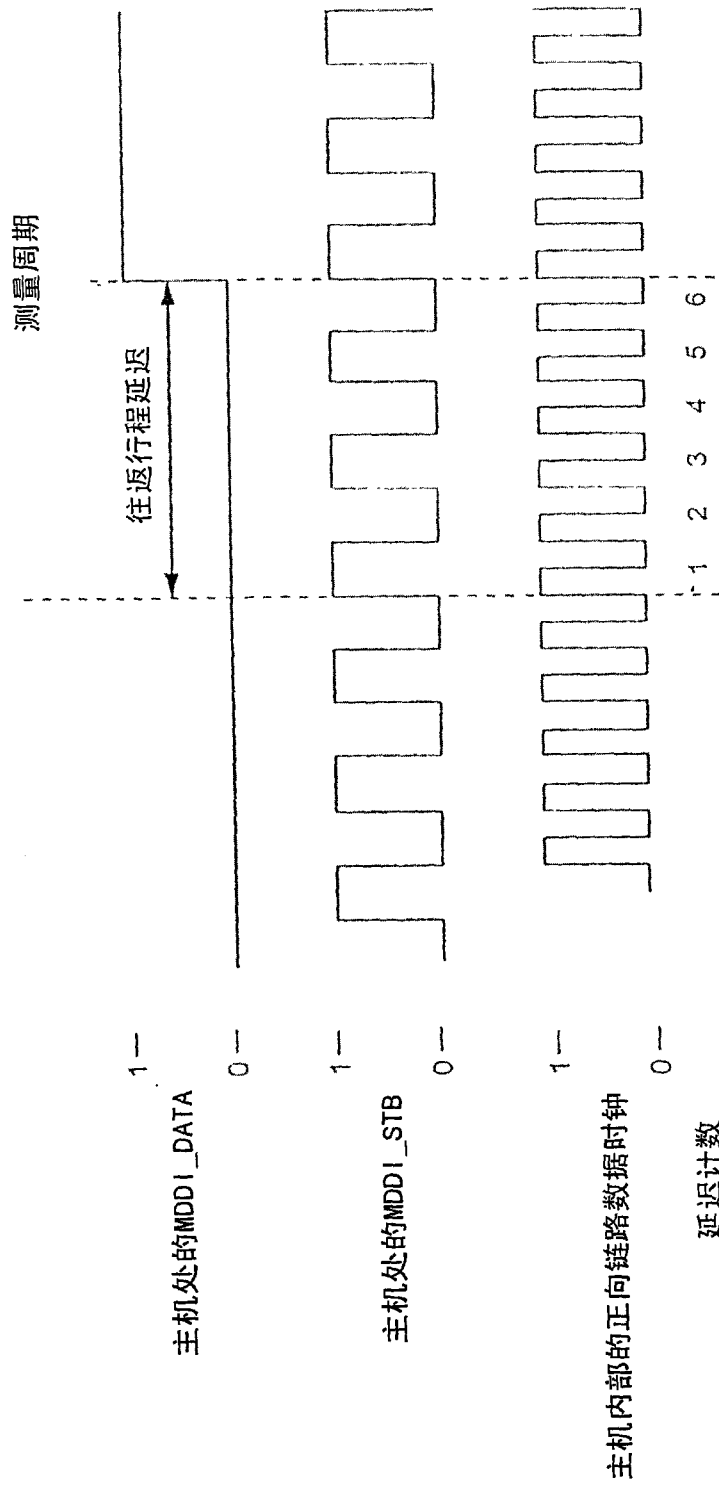


图51

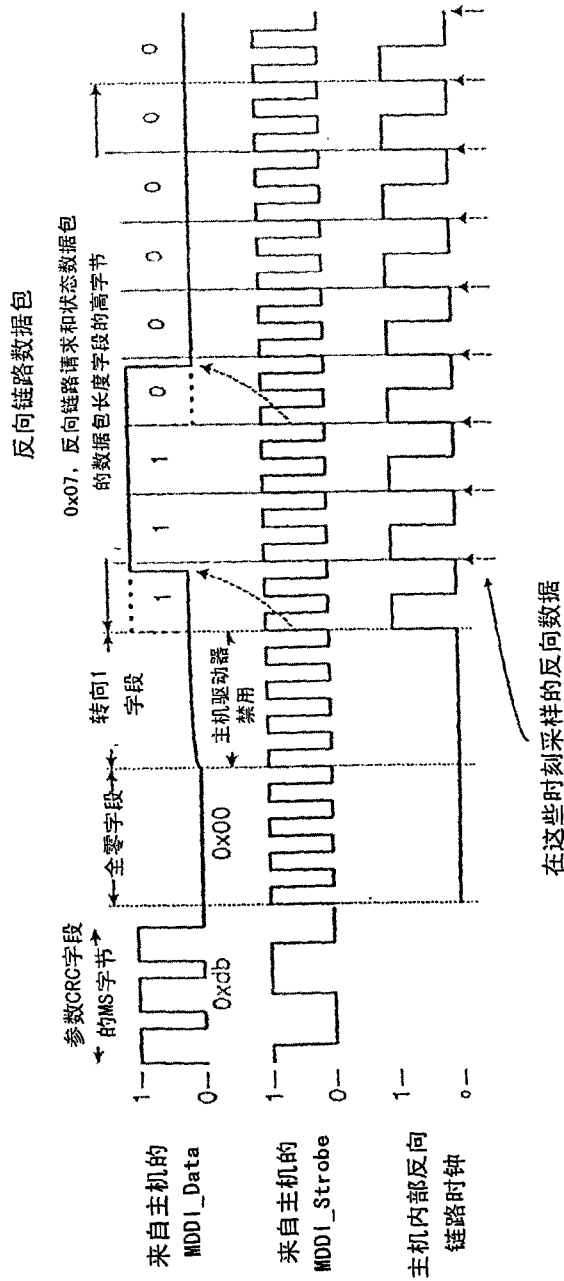


图52

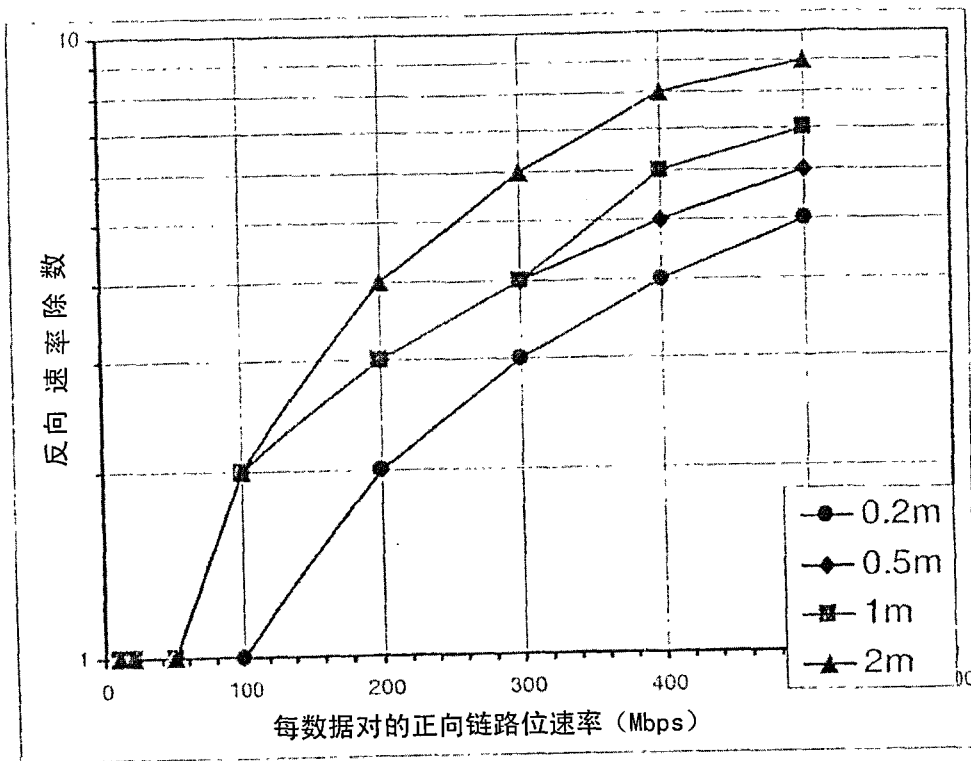


图53

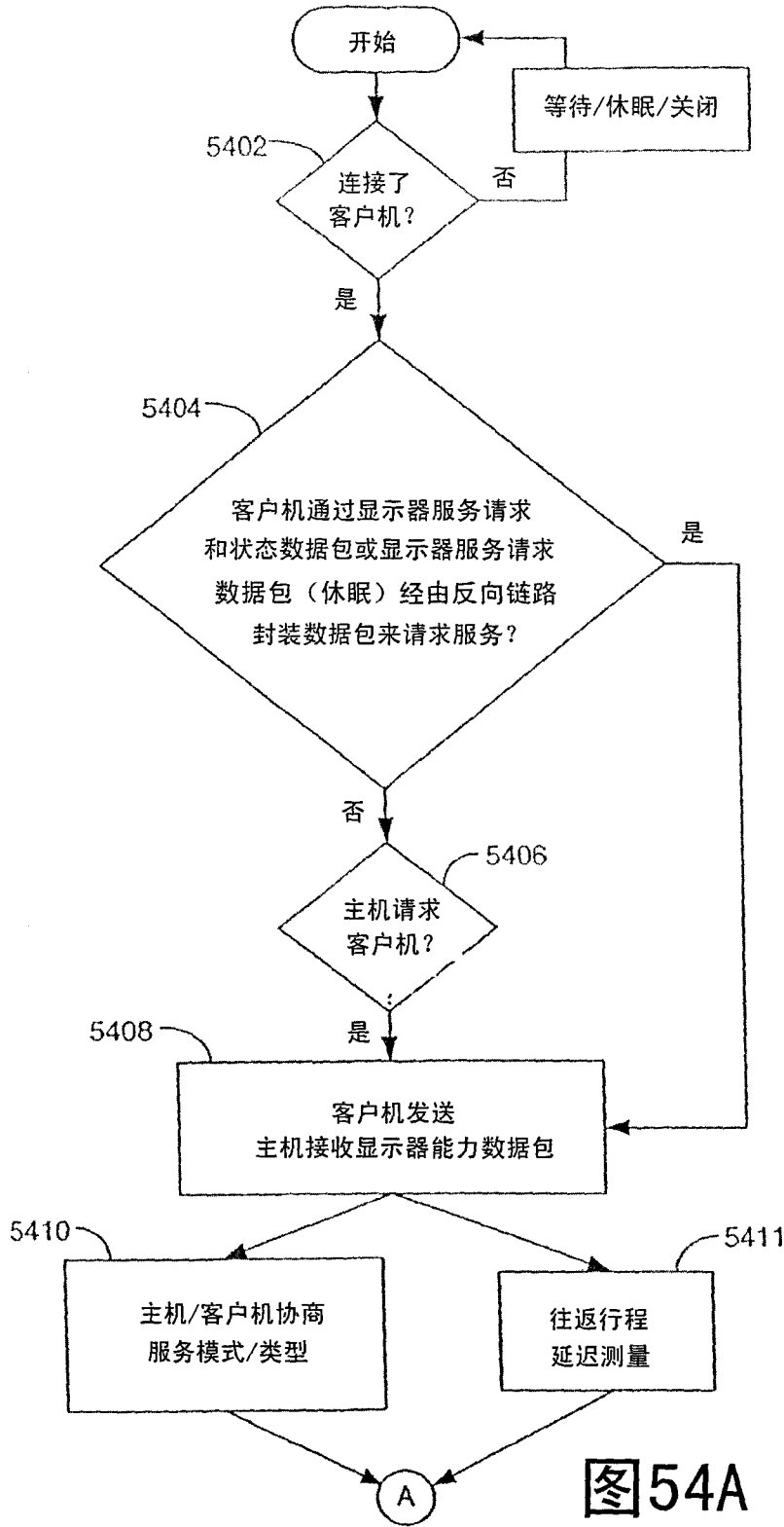


图54A

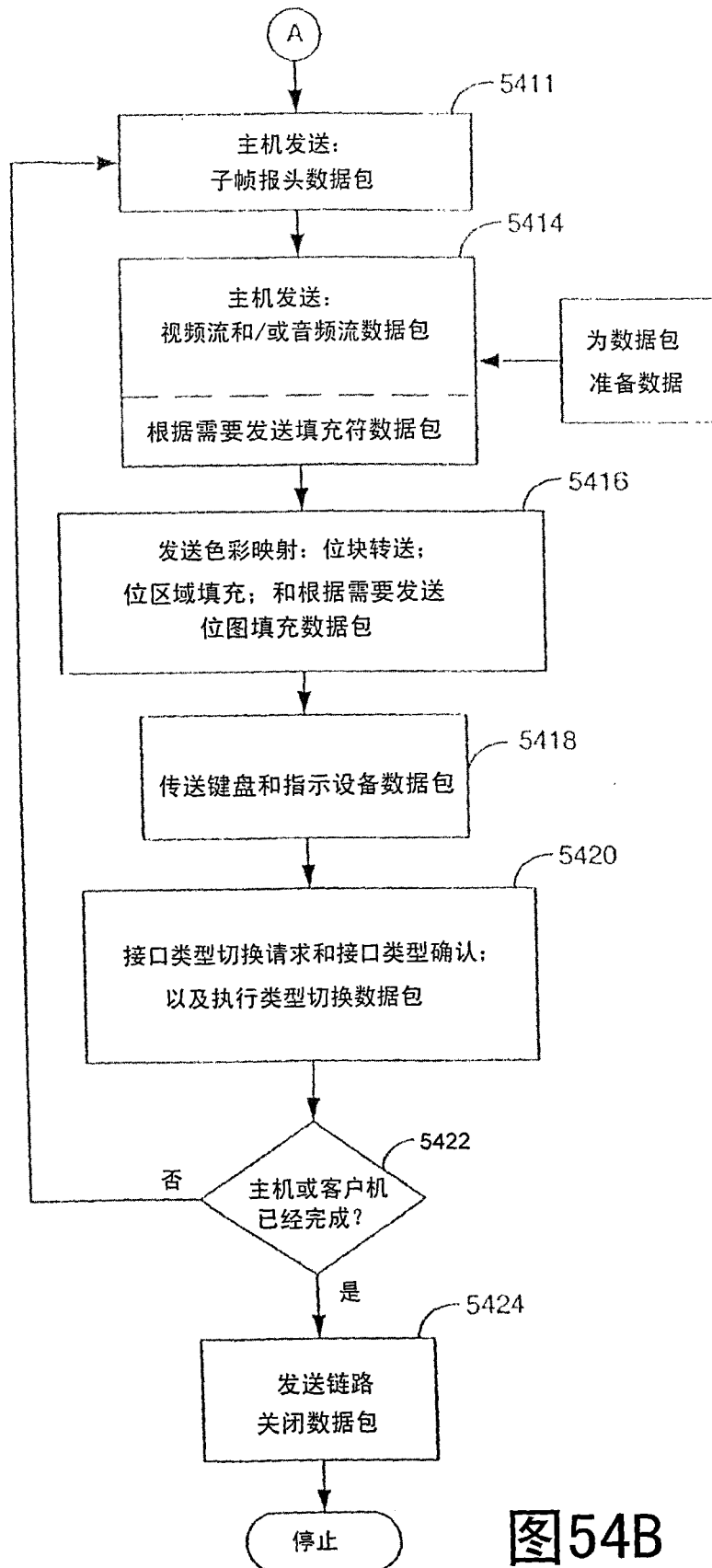


图54B

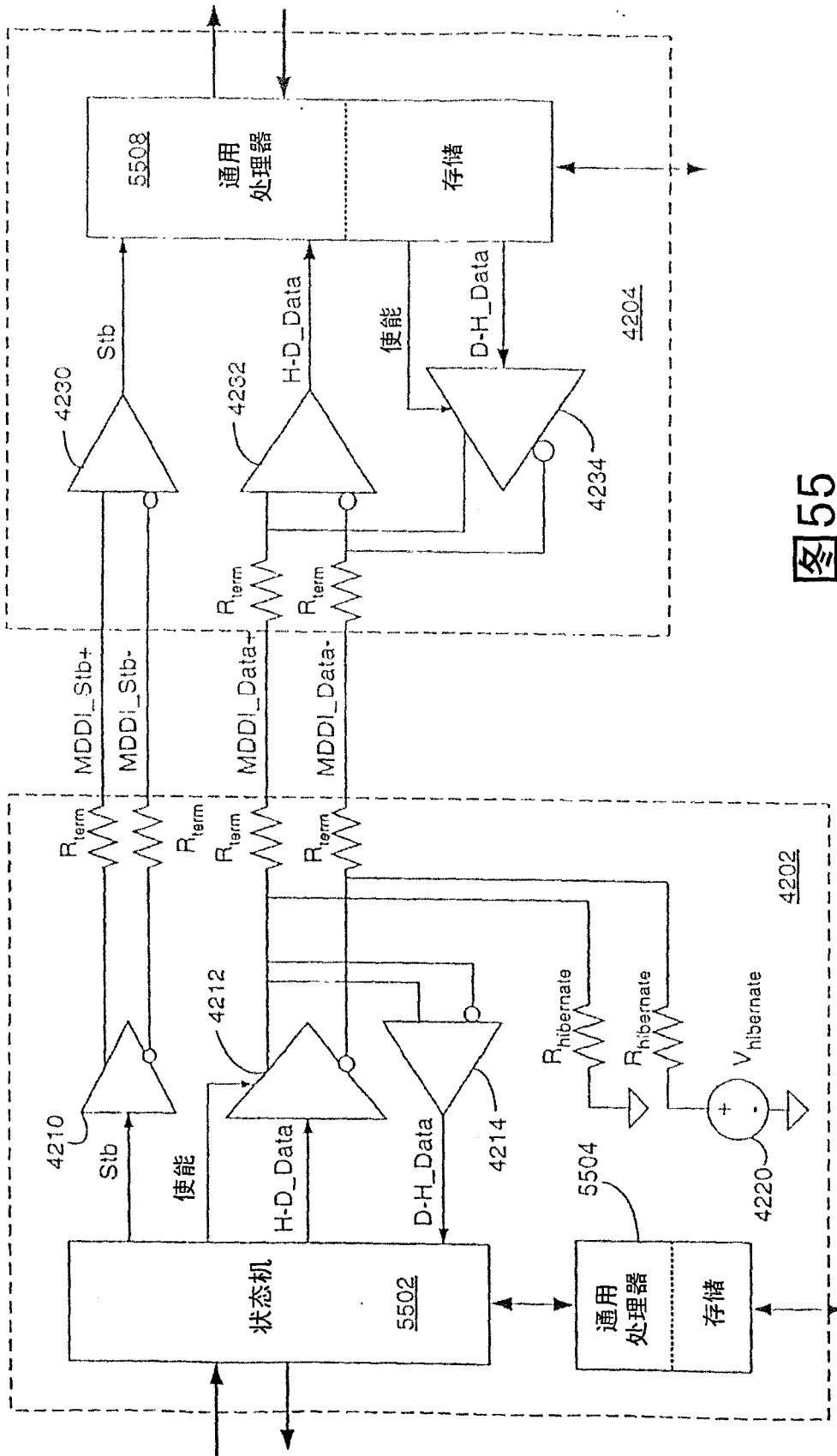


图55

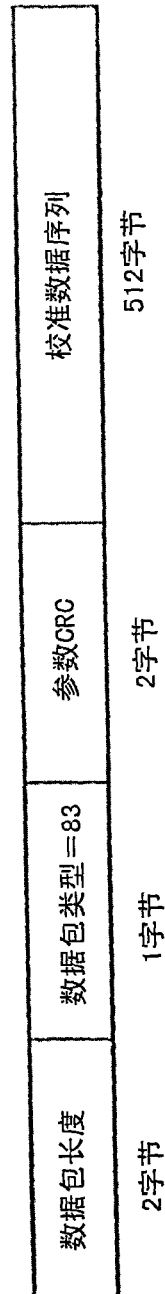


图56

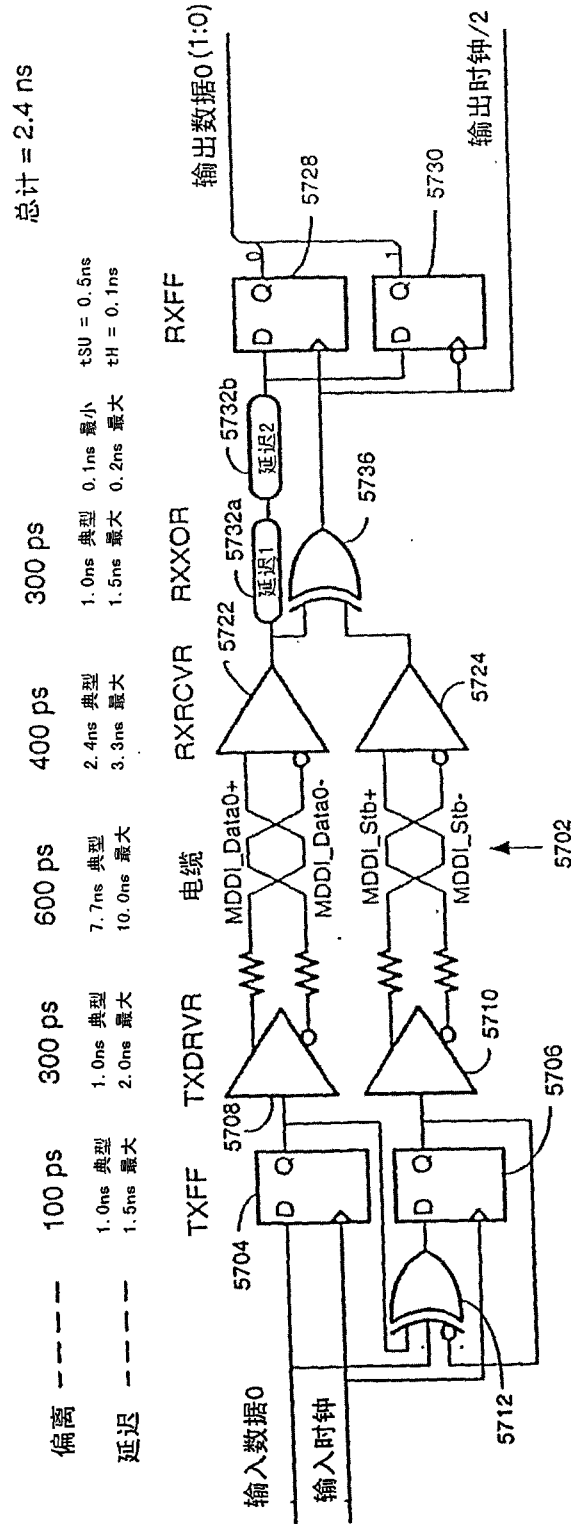


图57

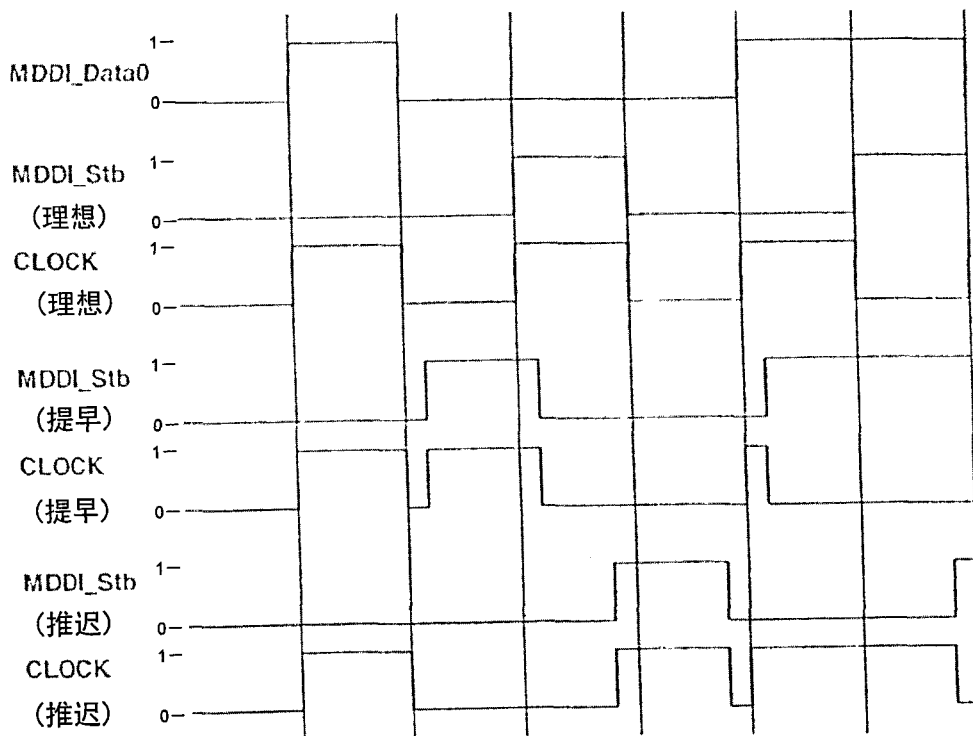


图58

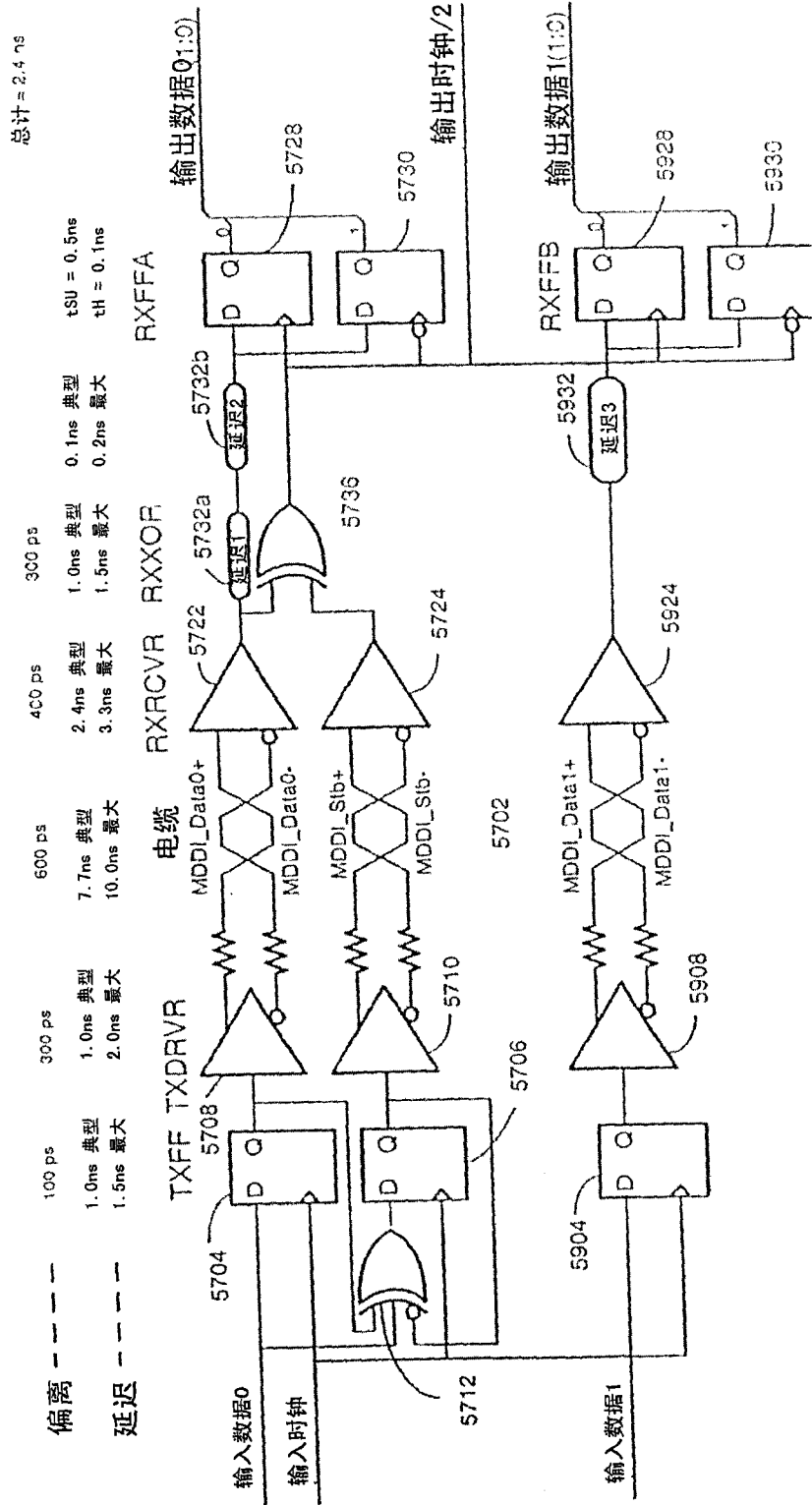


图59

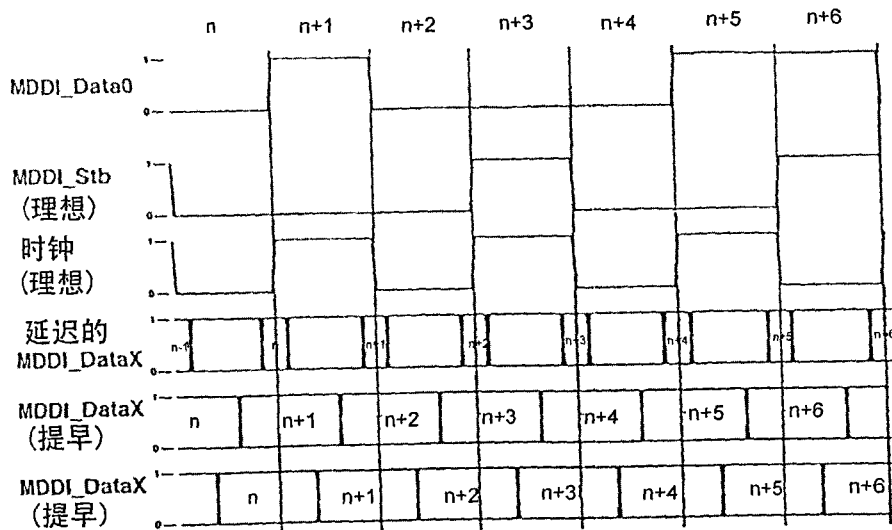


图 60A

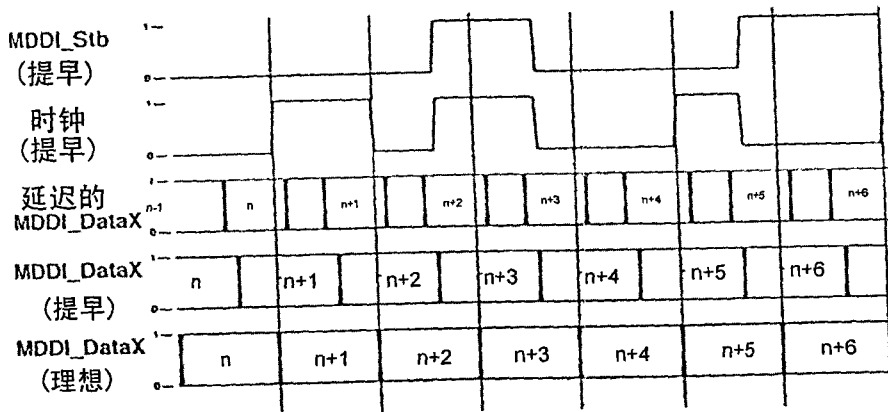


图 60B

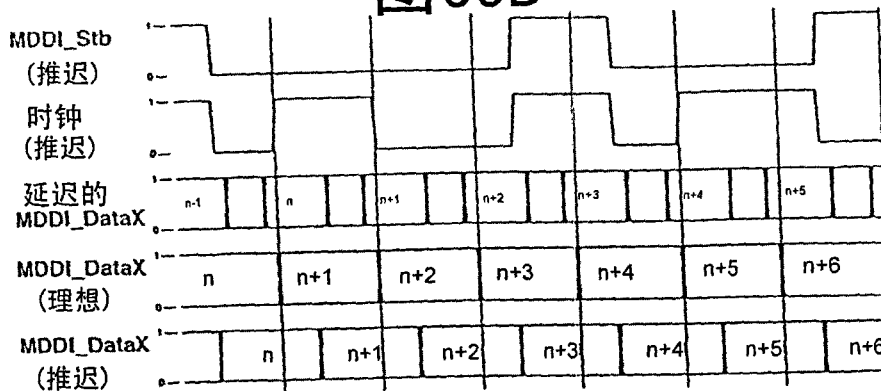


图 60C

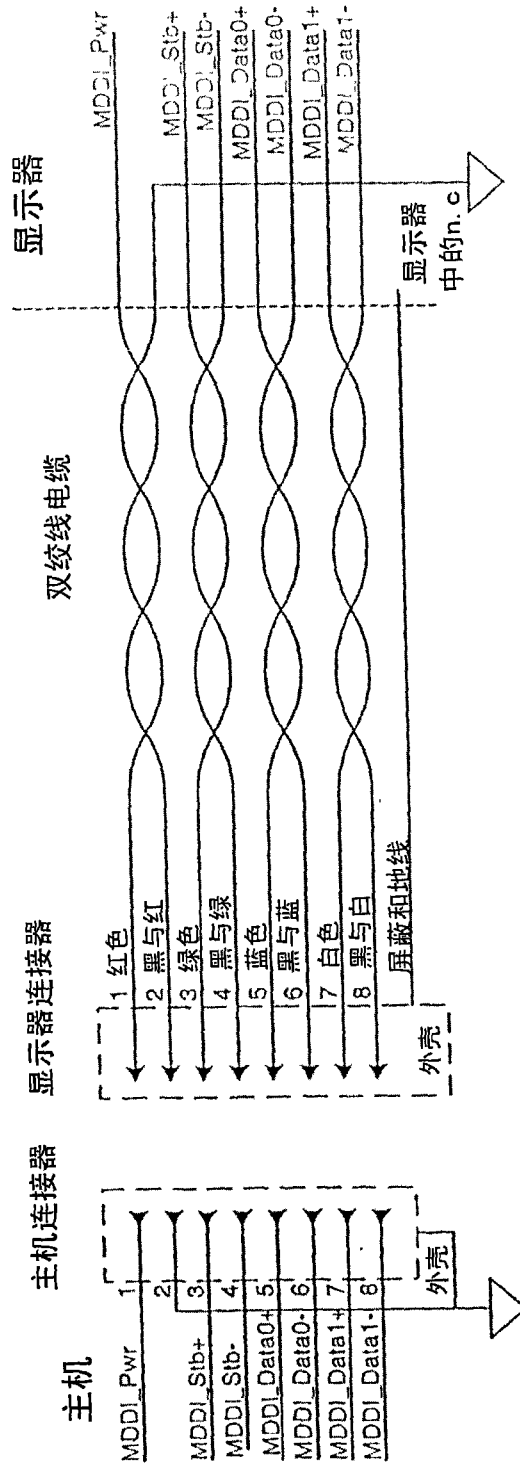


图61

I型数据实例

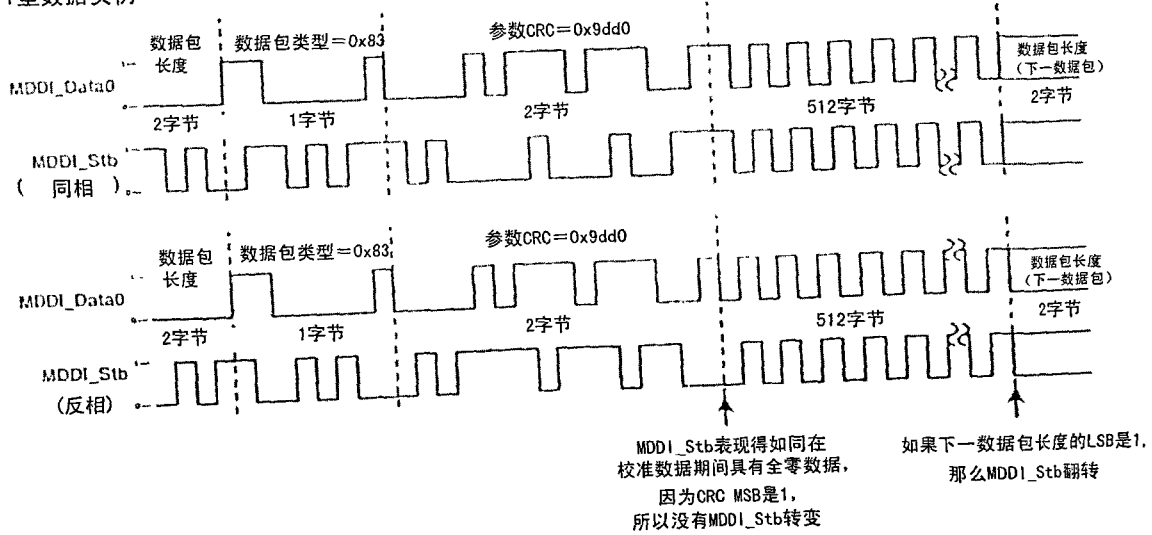


图62A

II型数据实例

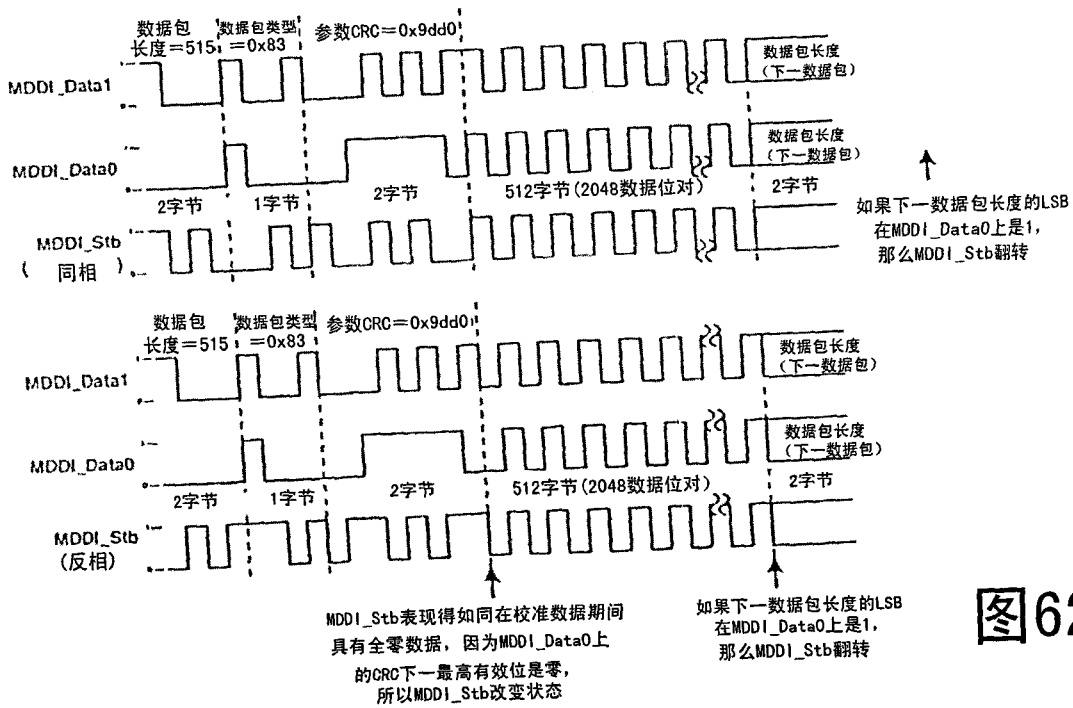
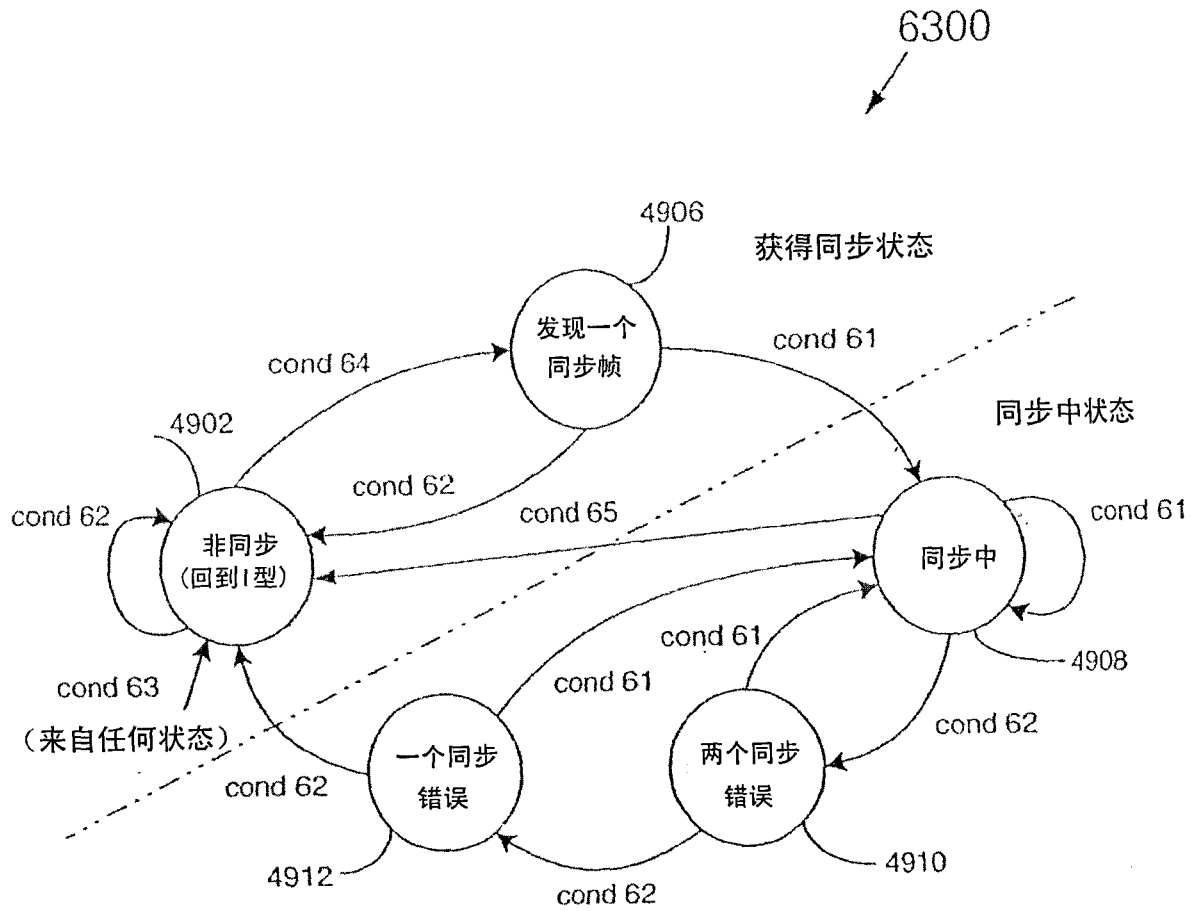


图62B



- cond 61 = 子帧报头数据包&处于帧边界的良好CRC, 帧长度>0
- cond 62 = 没有同步模式或帧边界处为坏的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

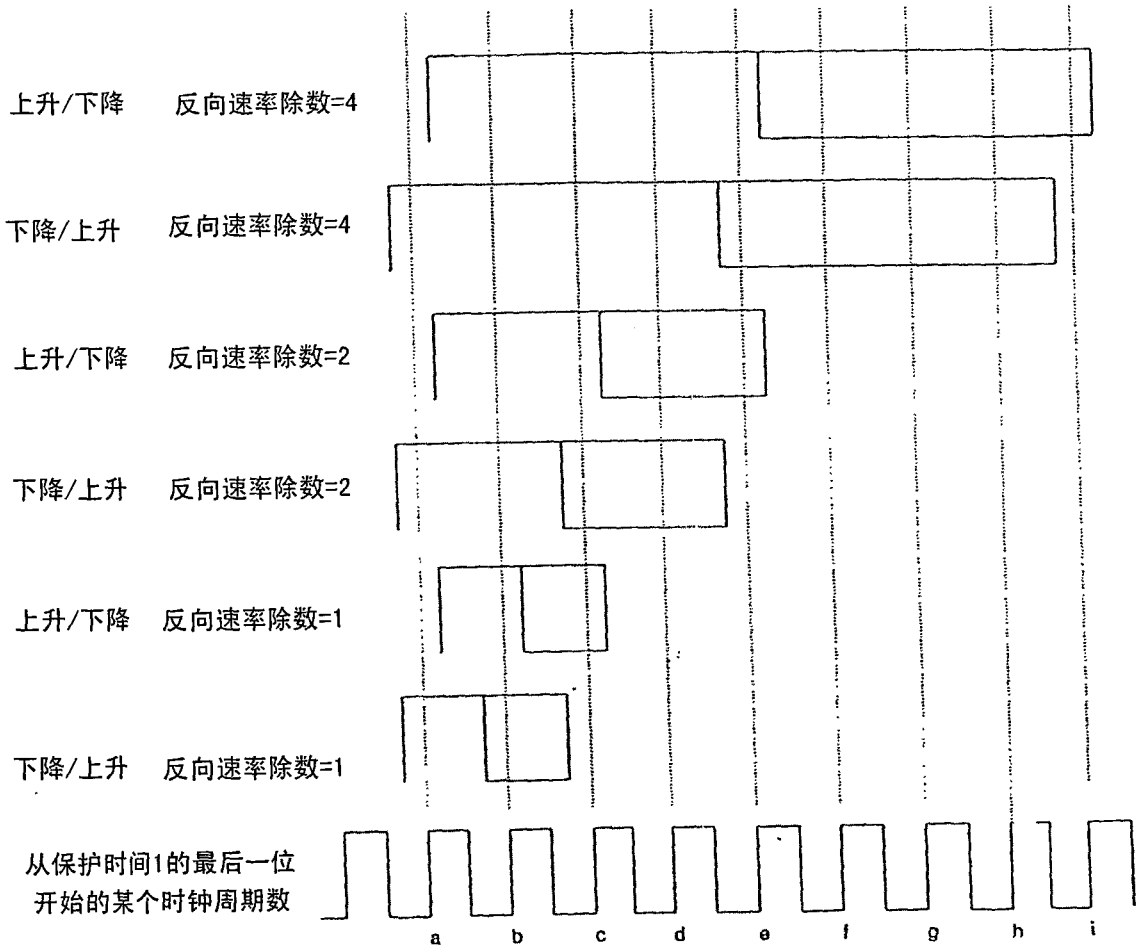


图64

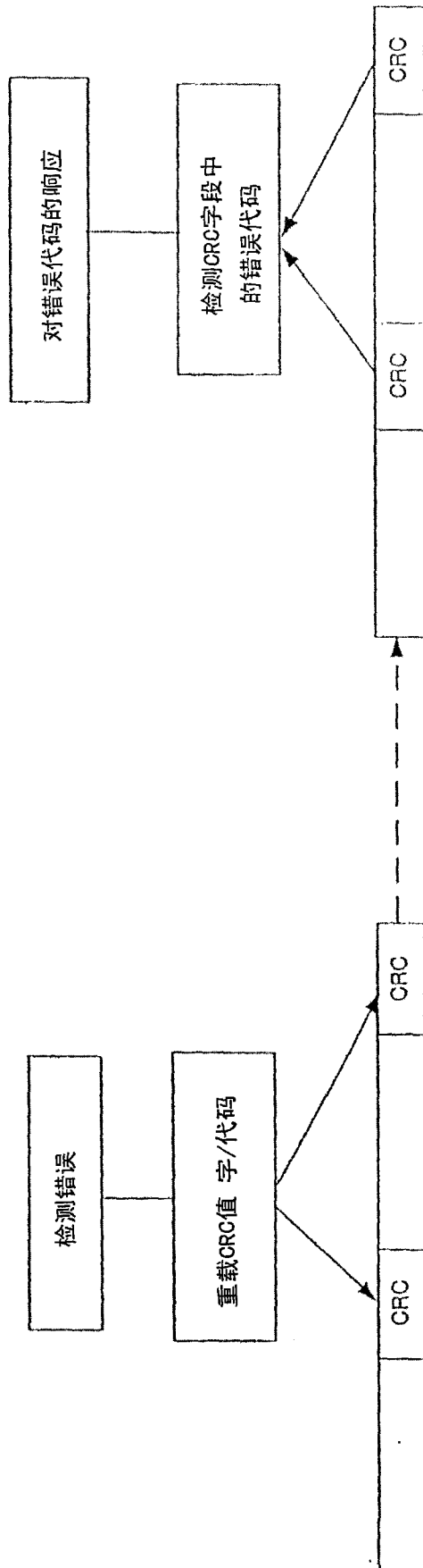


图65

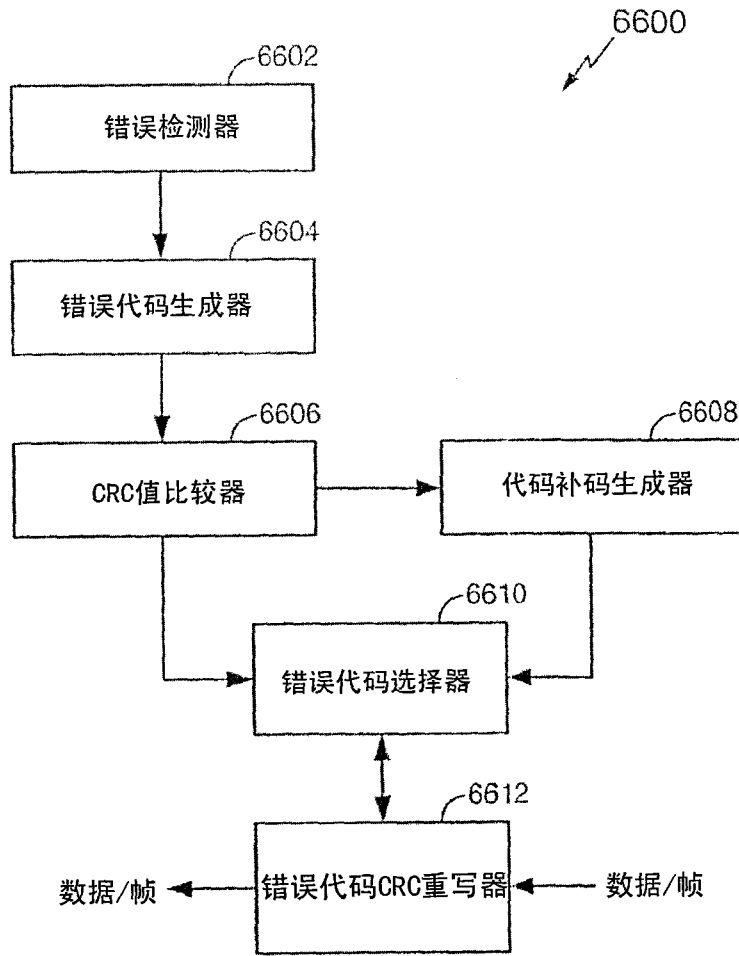


图66

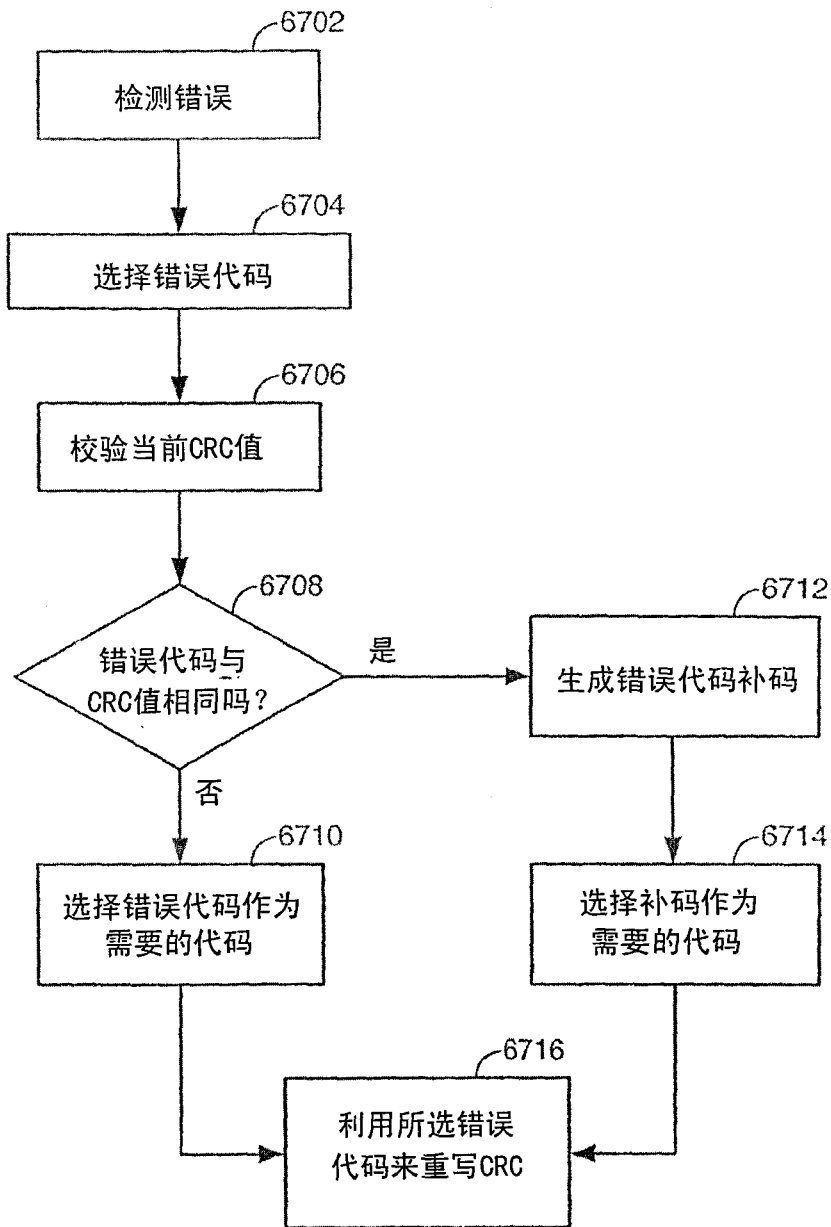


图67A

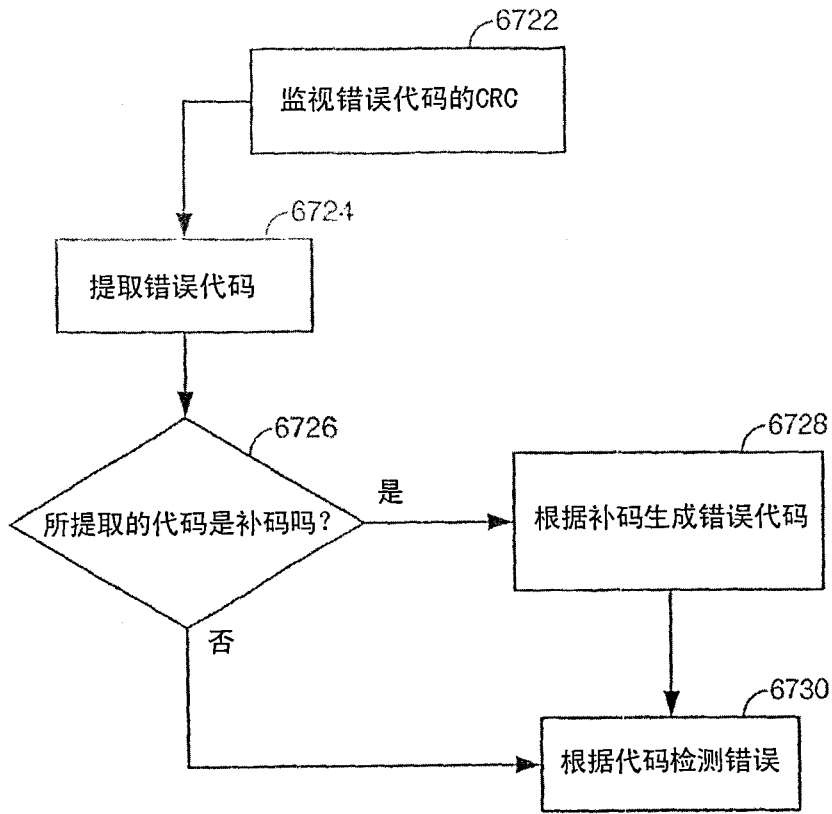


图67B

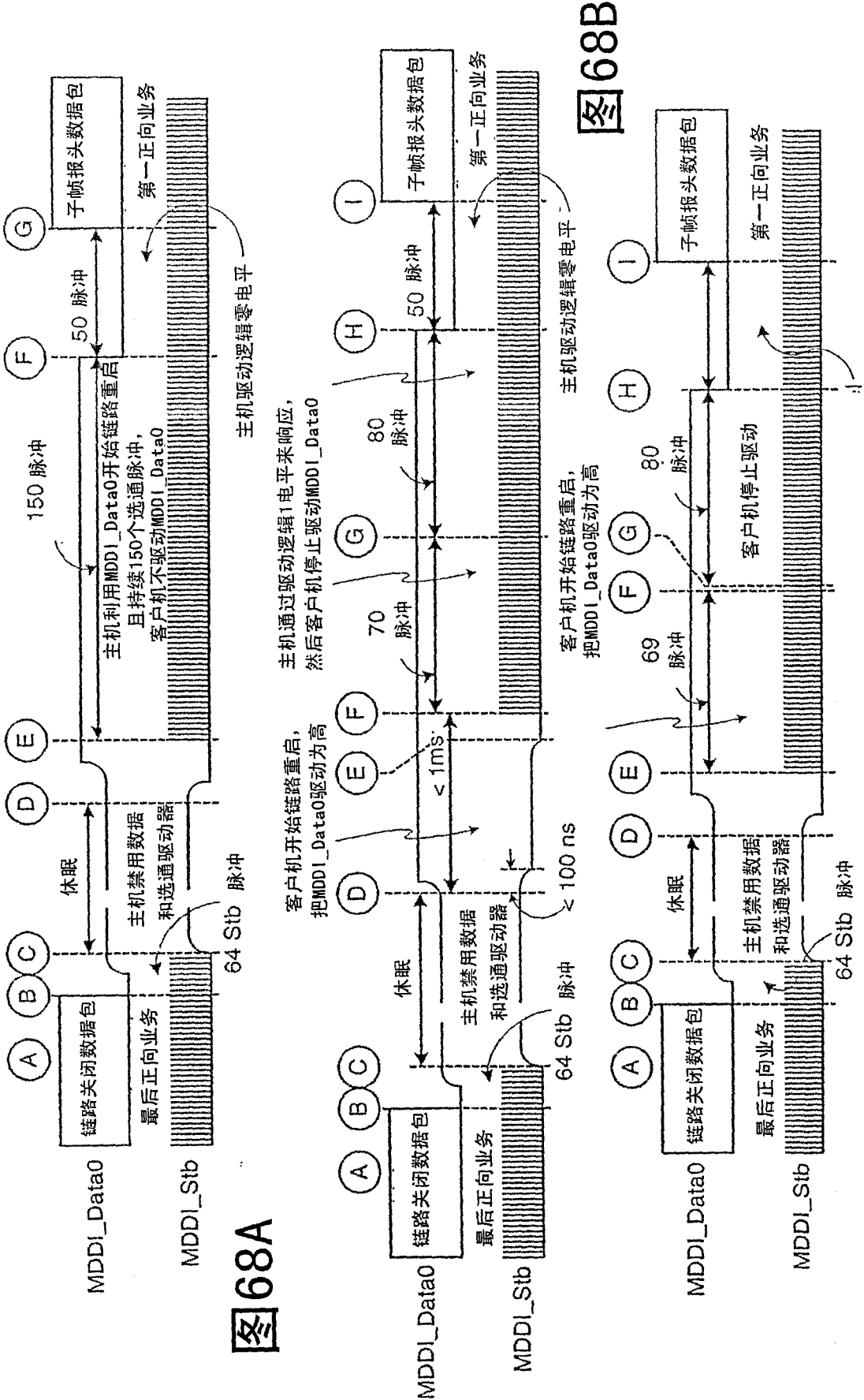


图 68C

请求VCP特征数据包

数据包长度	数据包类型 =128	hClient ID	MCCS VCP代码	CRC
2字节	2字节	2字节	2字节	2字节

图69

VCP特征应答数据包

数据包长度	数据包类型 =129	cClient ID	MCCS版本	应答序列号	列表中特征 数目	VCP特征应答列	CRC
2字节	2字节	2字节	2字节	2字节	2字节	(数据包长度-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特征数据包

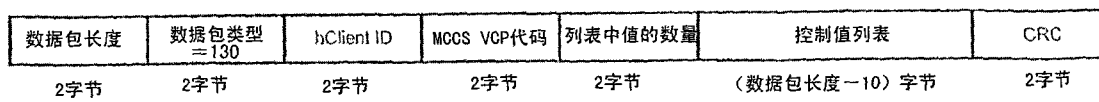


图72

请求有效参数数据包

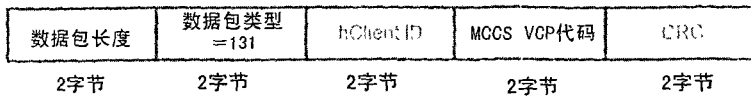


图73

有效参数应答数据包

数据包长度	数据包类型 =132	cClientID	MCGS 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 (数据包长度-14) 字节	CRC
2字节	2字节	2字节	2字节	2字节	2字节	2字节		2字节

图74

阿尔法光标图像能力数据包

数据包长度	数据包类型 = 133	cClient ID	阿尔法光标 标识符	阿尔法光标 位图宽度	阿尔法光标 位图高度
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	透明度映射 分辨率	能力位	CRC
2字节	1字节	1字节	2字节	1字节	1字节	2字节

图75

阿尔法光标透明度映射数据包

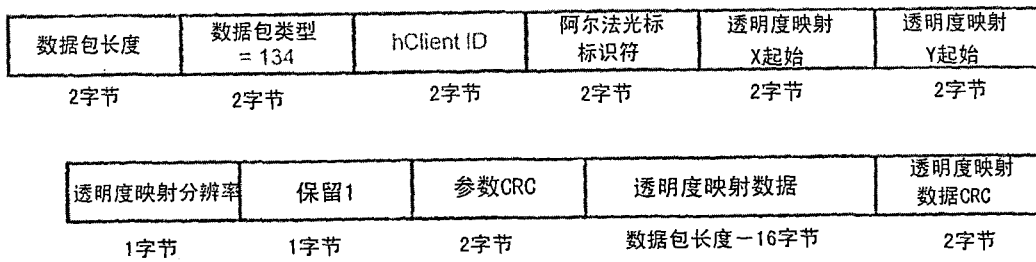


图76

阿尔法光标图像偏移数据包

数据包长度	数据包类型 = 135	hClientID	阿尔法光标 X偏移	阿尔法光标 Y偏移	CRC
2字节	2字节	2字节	2字节	2字节	2字节

图77

阿尔法光标视频流数据包

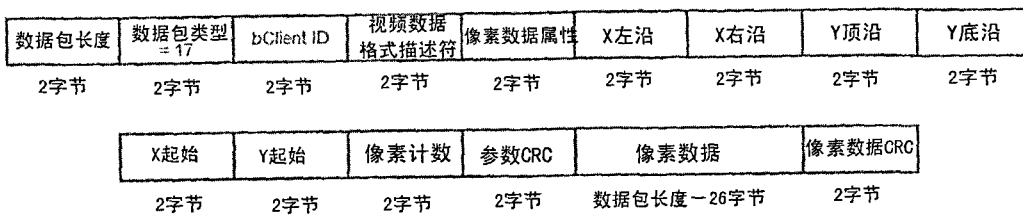


图78

可缩放视频流能力数据包

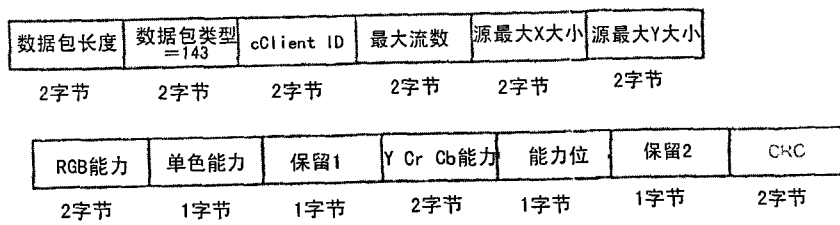


图79

可缩放视频流设置数据包

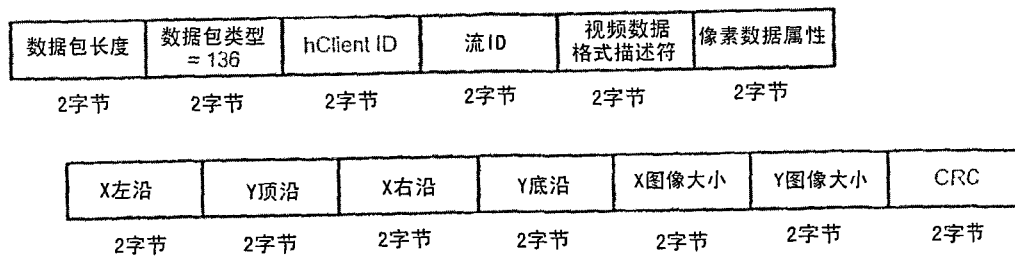


图80

可缩放视频流确认数据包

数据包长度	数据包类型 = 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81

可缩放视频流数据包



图82

请求专用状态数据包

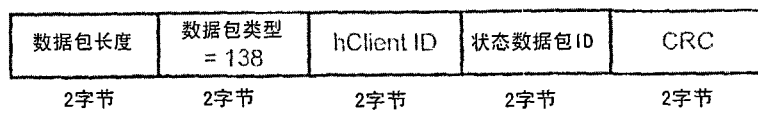


图83

有效状态应答列表数据包

数据包长度	数据包类型 = 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	数据包长度-8字节	2字节

图84

数据包处理延迟参数数据包

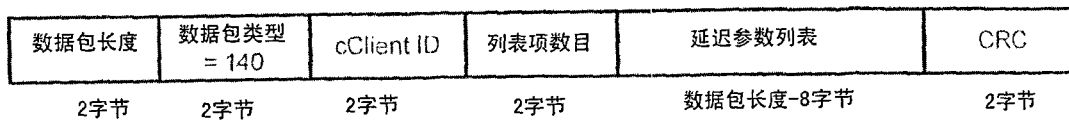


图85

个人显示器能力数据包

数据包长度	数据包类型 = 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 覆盖	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	1字节

图86

显示器错误报告数据包

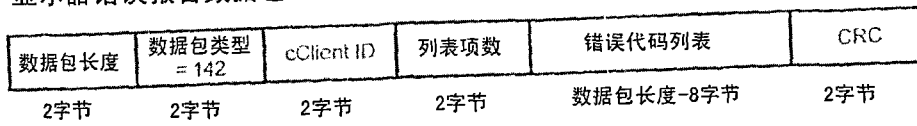


图87

显示器标识数据包

数据包长度	数据包类型 = 144	cClient ID	制造星期	制造商年份	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	2字节	2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 88

可选显示器能力数据包

数据包长度	数据包类型 =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89

寄存器访问数据包

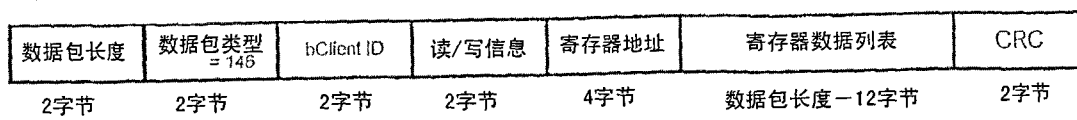
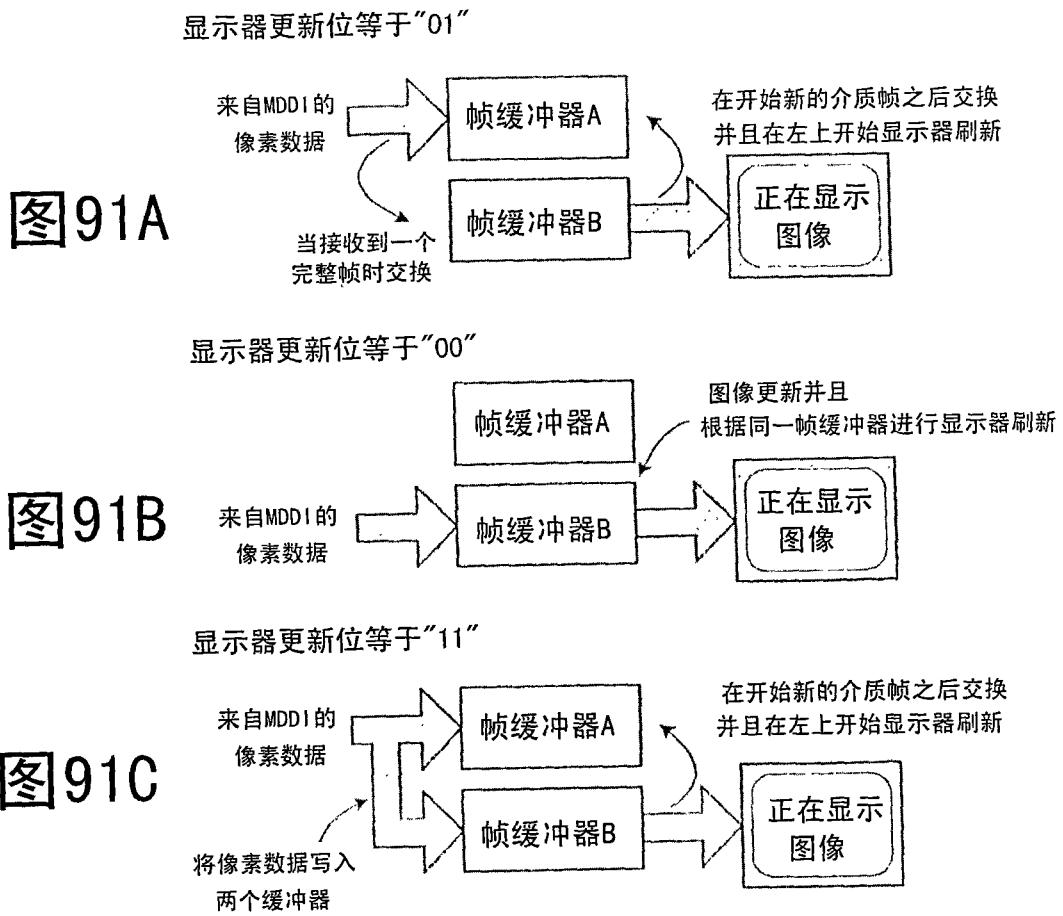


图90



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

图92

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

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

图93

两个缓冲器，显示器刷新比图像传送快，小视频窗

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=A 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
------------------------	------------------------	------------------------	------------------------	------------------------	------------------------

图94

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

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

图95

三个缓冲器，显示器刷新慢于图像传送

图像传送

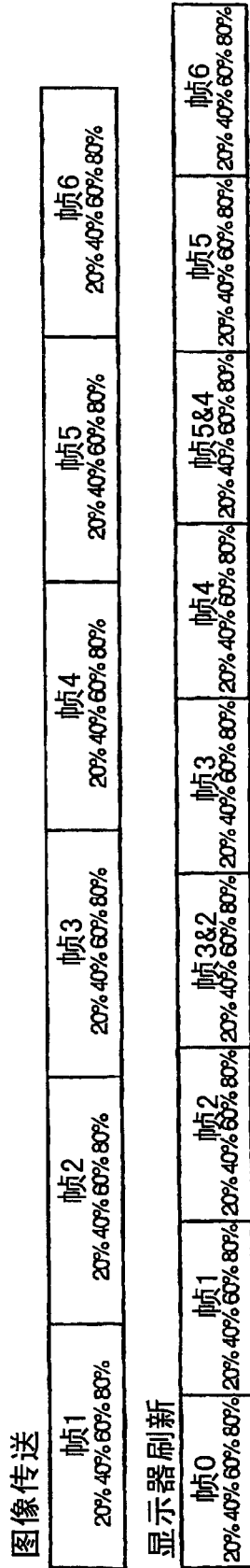
帧1, 缓冲器A 20% 40% 60% 80%	帧2, 缓冲器B 20% 40% 60% 80%	帧3, 缓冲器C 20% 40% 60% 80%	帧4, 缓冲器A 20% 40% 60% 80%	帧5, 缓冲器B 20% 40% 60% 80%	帧5, 缓冲器C 20% 40% 60% 80%	帧7 20%
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------

显示器刷新

20%	帧0, 缓冲器C 40% 60% 80%	20%	帧1&4, 缓冲器A 40% 60% 80%	20%	帧3, 缓冲器C 40% 60% 80%	20%	帧4&7, 缓冲器A 40% 60% 80%
-----	-------------------------	-----	---------------------------	-----	-------------------------	-----	---------------------------

图96

一个缓冲器，显示器刷新快于图像传送



一个缓冲器，显示器刷新比图像传送快得多

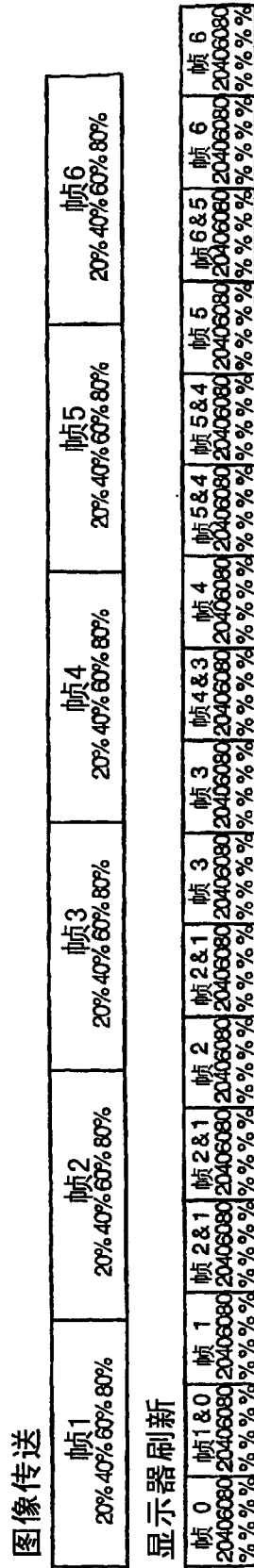
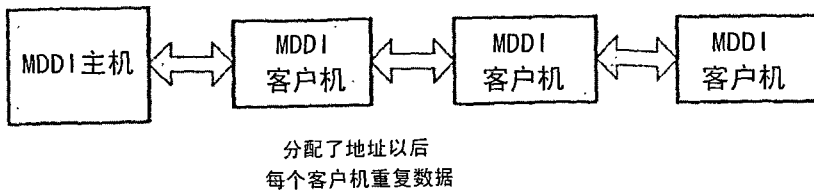


图97

菊花链配置



通过集线器连接的客户机

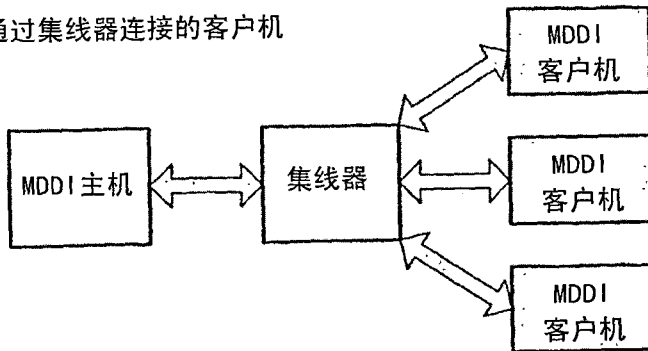


图98

集线器和菊花链的组合

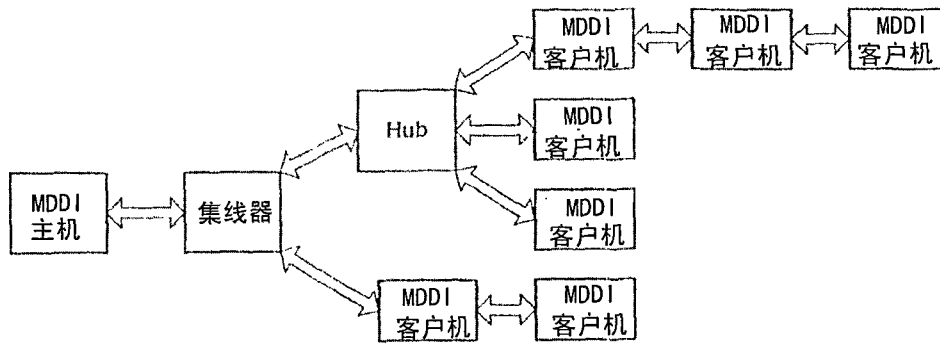


图99