



(12) 发明专利申请

(10) 申请公布号 CN 103885907 A

(43) 申请公布日 2014. 06. 25

(21) 申请号 201310714653. 1

(22) 申请日 2013. 12. 20

(30) 优先权数据

13/722, 240 2012. 12. 20 US

(71) 申请人 辉达公司

地址 美国加利福尼亚州

(72) 发明人 迈克尔·霍普古德 黄伟杰

维沙·梅塔 希滕德拉·杜塔

(74) 专利代理机构 北京市磐华律师事务所

11336

代理人 董巍 谢梅

(51) Int. Cl.

G06F 13/20 (2006. 01)

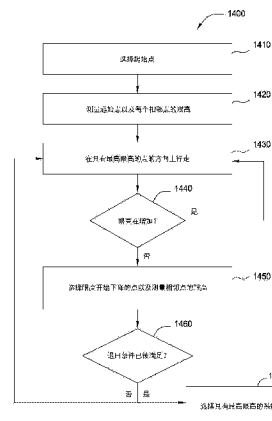
权利要求书2页 说明书14页 附图12页

(54) 发明名称

均衡系数搜索算法

(57) 摘要

公开了均衡系数搜索算法。一种方法,其包括选择均衡系数的映射图上的起始点以及测量使用与起始点相关联的均衡系数集所传送的信号的眼高以及与相对于起始点的映射图上的每个相邻点相关联的眼高。与相邻点相关联的眼高是基于使用与相邻点相关联的均衡系数集所传送的信号的。方法还包括在从起始点到与最大眼高相关联的相邻点的第一方向上在映射图上行走,其中与相邻点相关联的眼高大于或等于与起始点相关联的眼高。



1. 一种用于分析用于高速总线的均衡系数的计算机实现方法,所述方法包括:  
选择均衡系数的映射图上的起始点,其中所述映射图上的每个点包括不同的均衡系数集;  
测量使用与所述起始点相关联的均衡系数集所传送的信号的眼高以及与所述映射图上的相对于所述起始点的每个相邻点相关联的眼高,其中所述与相邻点相关联的眼高是基于使用与所述相邻点相关联的均衡系数集所传送的信号的;以及  
在从所述起始点到与最大眼高相关联的相邻点的第一方向上在所述映射图上行走,其中与所述相邻点相关联的眼高大于或等于与所述起始点相关联的眼高。
2. 根据权利要求1所述的方法,进一步包括:  
将与最大眼高相关联的相邻点指定为当前横断点;以及  
在从所述当前横断点到所述映射图上的相对于所述当前横断点的相邻点的第一方向上在所述映射图上行走,其中与所述相邻点相关联的眼高大于或等于与所述当前横断点相关联的眼高。
3. 根据权利要求2所述的方法,其中重复所述指定以及在第一方向上在所述映射图上行走的步骤,直到到达所述映射图上的当前横断点,此处与所述映射图上的相对于所述当前横断点的相邻点相关联的眼高小于与所述当前横断点相关联的眼高。
4. 根据权利要求3所述的方法,进一步包括:  
测量与所述映射图上的相对于所述当前横断点的每个相邻点相关联的眼高,其中所述与相邻点相关联的眼高是基于使用与所述相邻点相关联的均衡系数集所传送的信号的;以及  
在从所述当前横断点到与最大眼高相关联的相邻点的第二方向上在所述映射图上行走,其中与所述相邻点相关联的眼高大于或等于与所述当前横断点相关联的眼高。
5. 根据权利要求2所述的方法,其中重复所述指定以及在第一方向上在所述映射图上行走的步骤直到退出条件被满足。
6. 根据权利要求5所述的方法,其中所述退出条件包括:  
如果所测量的眼高高于预定的阈值,则退出;  
如果已经探访过预定数目的点,则退出;  
如果已经超过预定的时限,则退出;或者  
如果与所述当前横断点相邻的每个点具有比与所述当前横断点相关联的眼高更低的眼高,则退出。
7. 根据权利要求1所述的方法,其中选择所述起始点包括:  
将所述映射图划分成多个区域;  
测量每个区域中的一个点的眼高;以及  
选择具有最高眼高的点作为所述起始点。
8. 根据权利要求1所述的方法,其中选择所述起始点包括:  
沿所述均衡系数的映射图迭代地行走至少一个新点;以及  
测试由所述至少一个新点所表示的均衡系数集。
9. 根据权利要求8所述的方法,其中当退出条件已被满足时选择所述起始点。
10. 一种计算设备,包括:

处理器 ;以及

存储器,其耦连到所述处理器并且包括软件应用,所述软件应用当由所述处理器执行时使所述处理器 :

选择均衡系数的映射图上的起始点,其中所述映射图上的每个点包括不同的均衡系数集 ;

测量使用与所述起始点相关联的均衡系数集所传送的信号的眼高以及与所述映射图上的相对于所述起始点的每个相邻点相关联的眼高,其中与所述相邻点相关联的眼高是基于使用与所述相邻点相关联的均衡系数集所传送的信号的 ;以及

在从所述起始点到与最大眼高相关联的相邻点的第一方向上在所述映射图上行走,其中与所述相邻点相关联的眼高大于或等于与所述起始点相关联的眼高。

## 均衡系数搜索算法

### 技术领域

[0001] 本发明总地涉及数据传送,并且更具体地,涉及用于使用均衡系数搜索算法实施信道均衡调训的方法。

### 背景技术

[0002] 诸如外围部件接口(PCI)或 PCI express (PCIe) 的典型的数据连接器允许计算机系统内的不同处理单元彼此交换数据。例如,常规计算机系统可包括跨 PCIe 总线与图形处理单元(GPU)交换数据的中央处理单元(CPU)。

[0003] 当信号跨数据连接器在传送信道上传送时,一些频率分量可能比其它分量衰减更多,这可能使信号在接收端处难以辨认。随着传送速度变得更快,传送可能由于噪声效应更严重而变得更易于出错。在高速传送信道中,信号质量是至关重要的。减轻该倾向的一个技术是对信道进行“均衡”使得输入端处信号的频率域属性如实地再现在输出端处,产生较少错误。高速串行通信协议比如 PCIe 使用均衡器来准备数据信号用于传送。

[0004] 可以对信道的传送端和接收端二者实施均衡。对于传送均衡而言,可以在发送信号以尝试克服将由信道所引入的失真之前在传送端处重塑信号。在接收端处,可对信号进行修复以改进信号质量。

[0005] 对于 PCIe 中的传送均衡而言,公知为均衡系数的两个参数可以用来调谐传送器。典型的系统可以具有数百个均衡系数的组合,并且这些组合中的一些组合将产生比其它组合更好的均衡结果。信号质量在高速传送信道中是至关重要的,因此系数的最优集合对于确保准确传送是极其重要的。在均衡过程期间,必须选择满足系统的性能要求的系数的一个组合。此外,必须在固定的时限内选择该组合,使得系统可以启动或开始其它过程。测试系数的每一个组合以找到最佳的一个是不可行的,因为该方法通常将花费太多时间。此外,用来测试系数的组合的子集的当前方法经常将导致选择次优的组合。

[0006] 因此,本领域需要的是以更高效的方式测试和选择用于高速总线的均衡系数的技术。

### 发明内容

[0007] 本发明的一个实施例阐述了用于分析用于高速数据总线的均衡系数的方法。方法包括在均衡系数的映射图(map)上选择起始点以及测量使用与起始点相关联的均衡系数集所传送的信号的眼高以及与相对于起始点的映射图上的每个相邻点相关联的眼高。与相邻点相关联的眼高是基于使用与相邻点相关联的均衡系数集所传送的信号的。方法还包括在从起始点到与最大眼高相关联的相邻点的第一方向在映射图上行走,其中与相邻点相关联的眼高大于或等于与起始点相关联的眼高。

[0008] 有利地,使用上述技术选择均衡系数及对满足系统要求的质量标准的系数的较快选择。

## 附图说明

[0009] 因此,可以详细地理解本发明的上述特征,并且可以参考实施例得到对如上面所简要概括的本发明更具体的描述,其中一些实施例在附图中示出。然而,应当注意的是,附图仅示出了本发明的典型实施例,因此不应被认为是对其范围的限制,本发明可以具有其他等效的实施例。

[0010] 图 1 是示出了配置为实现本发明的一个或多个方面的计算机系统的框图;

[0011] 图 2 是根据本发明的一个实施例的、用于图 1 的计算机系统的并行处理子系统的框图;

[0012] 图 3 是所发射信号和所接收信号的常规例示;

[0013] 图 4 是产生错误的所发射数据和所接收数据的常规例示;

[0014] 图 5 是根据本发明的一个实施例的、在均衡之前和之后接收端处的信号的例示;

[0015] 图 6 是来自示波器的眼图的常规例示;

[0016] 图 7 是根据本发明的一个实施例的、均衡系数的映射图的例示;

[0017] 图 8A 和 8B 示出了根据本发明的一个实施例的、用于对均衡系数的映射图实施粗粒度搜索的一个技术;

[0018] 图 9 是根据本发明的一个实施例的、对均衡系数的映射图实施粗粒度搜索的另一个技术的例示;

[0019] 图 10 是根据本发明的一个实施例的、用于对均衡系数的映射图实施细粒度搜索的技术的例示;

[0020] 图 11 是根据本发明的一个实施例的、示出用于信道均衡调训的示范性多道方法的流程图;

[0021] 图 12 是根据本发明的一个实施例的、示出用于实施粗粒度搜索的示范性技术的流程图;

[0022] 图 13 是根据本发明的一个实施例的、示出用于实施粗粒度搜索的另一个示范性技术的流程图;以及

[0023] 图 14 是根据本发明的一个实施例的、示出用于实施细粒度搜索的示范性技术的流程图。

## 具体实施方式

[0024] 在下面的描述中,将阐述大量的具体细节以提供对本发明更透彻的理解。然而,本领域的技术人员应该清楚,本发明可以在没有或多个这些具体细节的情况下得以实践。在其他实例中,未描述公知特征以避免对本发明造成混淆。

### [0025] 系统概述

[0026] 图 1 是示出了配置为实现本发明的一个或多个方面的计算机系统 100 的框图。计算机系统 100 包括中央处理单元(CPU) 102 和包括设备驱动程序 103 的系统存储器 104。CPU102 和系统存储器 104 经由可以包括存储器桥 105 的互连路径进行通信。存储器桥 105 可以是例如北桥芯片,经由总线或其他通信路径 106 (例如超传输(HyperTransport)链路)连接到输入/输出(I/O)桥 107。I/O 桥 107,其可以是例如南桥芯片,从一个或多个用户输入设备 108 (例如键盘、鼠标)接收用户输入并且经由路径 106 和存储器桥 105 将该输入转

发到 CPU102。并行处理子系统 112 经由总线或其他通信路径 113(例如外围部件互连(PCI Express、加速图形端口(AGP)或超传输链路)耦连到存储器桥 105;在一个实施例中,并行处理子系统 112 是将像素传递到显示设备 110(例如常规的基于阴极射线管(CRT)或液晶显示器(LCD)的监视器)的图形子系统。系统盘 114 也连接到 I/O 桥 107。交换器 116 提供 I/O 桥 107 与诸如网络适配器 118 以及各种插卡 120 和 121 的其他部件之间的连接。其他部件(未明确示出),包括通用串行总线(USB)或其他端口连接、压缩光盘(CD)驱动器、数字视频光盘(DVD)驱动器、胶片录制设备及类似部件,也可以连接到 I/O 桥 107。将图 1 中的各部件互连的通信路径可以使用任何适合的协议实现,诸如 PCI、PCI-Express (PCIe)、AGP、超传输或者任何其他总线或点到点通信协议,并且如本领域已知的,不同设备间的连接可使用不同协议。

[0027] PPU112 配置为执行诸如例如设备驱动程序 103 的软件应用,其允许 PPU112 生成任意可以跨通信路径 113 传送的包字节。这些包字节由通信路径 113 使用的通信协议指定。在新包字节被引入通信协议(例如由于通信协议的增强)的情况下,PPU112 可配置为基于新包字节生成包以及使用新包字节跨通信路径 113 与 CPU102(或其他处理单元)交换数据。

[0028] 在一个实施例中,并行处理子系统 112 包含经优化用于图形和视频处理的电路,包括例如视频输出电路,并且构成图形处理单元(GPU)。在另一个实施例中,并行处理子系统 112 包含经优化用于通用处理的电路,同时保留底层(underlying)的计算架构,本文将更详细地进行描述。在又一个实施例中,可以将并行处理子系统 112 与一个或多个其他系统元件集成,诸如存储器桥 105、CPU102 以及 I/O 桥 107,以形成片上系统(SoC)。

[0029] 应该理解,本文所示系统是示例性的,并且变化和修改都是可能的。连接拓扑,包括桥的数目和布置、CPU102 的数目以及并行处理子系统 112 的数目,可根据需要修改。例如,在一些实施例中,系统存储器 104 直接连接到 CPU102 而不是通过桥,并且其他设备经由存储器桥 105 和 CPU102 与系统存储器 104 通信。在其他替代性拓扑中,并行处理子系统 112 连接到 I/O 桥 107 或直接连接到 CPU102,而不是连接到存储器桥 105。而在其他实施例中,I/O 桥 107 和存储器桥 105 可能被集成到单个芯片上。大型实施例可以包括两个或更多个 CPU102 以及两个或更多个并行处理系统 112。本文所示的特定部件是可选的;例如,任何数目的插卡或外围设备都可能得到支持。在一些实施例中,交换器 116 被去掉,网络适配器 118 和插卡 120、121 直接连接到 I/O 桥 107。

[0030] 图 2 示出了根据本发明的一个实施例的并行处理子系统 112。如所示的,并行处理子系统 112 包括一个或多个并行处理单元(PPU)202,每个并行处理单元 202 都耦连到本地并行处理(PP)存储器 204。通常,并行处理子系统包括 U 个 PPU,其中  $U \geq 1$ 。(本文中,类似对象的多个实例需要时以标识对象的参考数字和标识实例的括号中的数字来表示。)PPU202 和并行处理存储器 204 可使用一个或多个集成电路设备来实现,诸如可编程处理器、专用集成电路(ASIC)或存储器设备,或者以任何其他技术可行的方式来实现。

[0031] 再参考图 1,在一些实施例中,并行处理子系统 112 中的一些或所有 PPU202 是具有渲染管线的图形处理器,其可以配置为实施与下述相关的各种任务:经由存储器桥 105 和总线 113 从 CPU102 和/或系统存储器 104 所供应的图形数据生成像素数据,与本地并行处理存储器 204(可被用作图形存储器,包括例如常规帧缓冲区(buffer))交互以存储和更新像素数据,传递像素数据到显示设备 110 等等。在一些实施例中,并行处理子系统 112

可包括一个或多个作为图形处理器而操作的 PPU202 以及一个或多个用于通用计算的其他 PPU202。这些 PPU 可以是同样的或不同的,并且每个 PPU 可具有它自己的专用并行处理存储器设备或不具有专用并行处理存储器设备。一个或多个 PPU202 可输出数据到显示设备 110,或者每个 PPU202 可输出数据到一个或多个显示设备 110。

[0032] 在操作中,CPU102 是计算机系统 100 的主处理器,控制和协调其他系统部件的操作。具体地,CPU102 发出控制 PPU202 的操作的命令。在一些实施例中,CPU102 写入用于每个 PPU202 的命令流到入栈缓冲区(pushbuffer)中(在图 1 或图 2 中未明确示出),该数据结构可位于系统存储器 104、并行处理存储器 204、或 CPU102 和 PPU202 都可访问的其他存储位置中。PPU202 从入栈缓冲区读取命令流,然后相对于 CPU102 的操作异步地执行命令。

[0033] 现在返回参考图 2,每个 PPU202 包括经由连接到存储器桥 105 (或者,在一个替代性实施例中,直接连接到 CPU102)的通信路径 113 与计算机系统 100 的其余部分通信的 I/O 单元 205。PPU202 到计算机系统 100 的其余部分的连接也可以变化。在一些实施例中,并行处理子系统 112 可实现为可插入到计算机系统 100 的扩展槽中的插卡。在其他实施例中,PPU202 可以和诸如存储器桥 105 或 I/O 桥 107 的总线桥集成在单个芯片上。而在其他实施例中,PPU202 的一些或所有元件可以和 CPU102 集成在单个芯片上。

[0034] 在一个实施例中,通信路径 113 是 PCIe 链路,如本领域所知的,其中专用通道被分配到每个 PPU202。也可以使用其他通信路径。如上所述,逆流(contrafLOW)互连也可以用来实现通信路径 113 以及计算机系统 100、CPU102 或 PPU202 内的任何其他通信路径。I/O 单元 205 生成用于在通信路径 113 上传送的包(或其他信号),并且还从通信路径 113 接收所有传入的包(或其他信号),将传入的包引导到 PPU202 的适当部件。例如,可将与处理任务相关的命令引导到主机接口 206,而将与存储器操作相关的命令(例如,对并行处理存储器 204 的读取或写入)引导到存储器交叉开关单元 210。主机接口 206 读取每个入栈缓冲区,并且将由入栈缓冲区所指定的工作输出到前端 212。

[0035] 有利地,每个 PPU202 都实现高度并行处理架构。如详细示出的,PPU202 (0) 包括处理集群阵列 230,该阵列 230 包括 C 个通用处理集群(GPC)208,其中  $C \geq 1$ 。每个 GPC208 能够并发执行大量的(例如,几百或几千)线程,其中每个线程是程序的实例(instance)。在各种应用中,可分配不同的 GPC208 用于处理不同类型的程序或用于实施不同类型的计算。例如,在图形应用中,GPC208 的第一集合可经分配以实施曲面细分(tessellation)操作以及产生用于补丁(patch)的基元拓扑,并且 GPC208 的第二集合可经分配以实施曲面细分着色来评估用于基元拓扑的补丁参数和其他每顶点属性。GPC208 的分配可以取决于因每种类型的程序或计算所产生的工作量而变化。

[0036] GPC208 经由工作分布单元 200 接收所要执行的处理任务,其接收定义来自前端单元 212 的处理任务的命令。处理任务包括所要处理的数据的索引,所述数据例如表面(补丁)数据、基元数据、顶点数据和 / 或像素数据,以及定义数据将被如何处理(例如,什么程序将被执行)的状态参数和命令。工作分布单元 200 可配置为获取与任务相对应的索引,或者工作分布单元 200 可以接收来自前端 212 的索引。前端 212 确保在由入栈缓冲区所指定的处理发起前,将 GPC208 配置为有效状态。

[0037] 当 PPU202 用于图形处理时,例如,用于每个补丁的处理工作量被分成接近相等大小的任务以使曲面细分处理能够分布到多个 GPC208。工作分布单元 200 可配置为以能够

将任务提供到多个 GPC208 用于处理的频率来产生任务。相比之下,在常规系统中,处理典型地由单个处理引擎来实施,而其他处理引擎保持空闲,在开始他们的处理任务之前等待单个处理引擎完成它的任务。在本发明的一些实施例中, GPC208 的各部分配置为实施不同类型的处理。例如,第一部分可配置为实施顶点着色和拓扑生成,第二部分可配置为实施曲面细分和几何着色,以及第三部分可配置为在屏幕空间中实施像素着色以产生经渲染的图像。由 GPC208 所产生的中间数据可以存储在缓冲区中以允许中间数据在 GPC208 之间进行传送用于进一步处理。

[0038] 存储器接口 214 包括 D 个分区单元 215,每个分区单元 215 直接耦连到并行处理存储器 204 的一部分,其中  $D \geq 1$ 。如所示的,分区单元 215 的数目一般等于 DRAM220 的数目。在其他实施例中,分区单元 215 的数目也可以不等于存储器设备的数目。本领域的技术人员应该理解动态随机存取存储器(DRAM) 220 可以用其他合适的存储设备来替代并且可以是一般常规的设计。因此省略了详细描述。诸如帧缓冲区或纹理映射图的渲染目标可以跨 DRAM220 加以存储,这允许分区单元 215 并行写入每个渲染目标的各部分以有效地使用并行处理存储器 204 的可用带宽。

[0039] 任何一个 GPC208 都可以处理要被写到并行处理存储器 204 内的任何 DRAM220 的数据。交叉开关单元 210 配置为路由每个 GPC208 的输出到任何分区单元 215 的输入或到另一个 GPC208 用于进一步处理。GPC208 通过交叉开关单元 210 与存储器接口 214 通信,以对各种外部存储器设备进行读取或写入。在一个实施例中,交叉开关单元 210 具有到存储器接口 214 的连接以和 I/O 单元 205 通信,以及到本地并行处理存储器 204 的连接,从而使得在不同 GPC208 内的处理内核能够与系统存储器 104 或对于 PPU202 而言非本地的其他存储器通信。在图 2 所示的实施例中,交叉开关单元 210 直接与 I/O 单元 205 连接。交叉开关单元 210 可使用虚拟信道来分开 GPC208 与分区单元 215 之间的业务流。

[0040] 另外, GPC208 可被编程以执行与种类繁多的应用相关的处理任务,包括但不限于,线性和非线性数据变换、视频和 / 或音频数据过滤、建模操作(例如,应用物理定律以确定对象的位置、速率和其他属性)、图像渲染操作(例如,曲面细分着色器、顶点着色器、几何着色器、和 / 或像素着色器程序)等等。PPU202 可将数据从系统存储器 104 和 / 或本地并行处理存储器 204 转移到内部(片上)存储器中,处理该数据,并且将结果数据写回到系统存储器 104 和 / 或本地并行处理存储器 204,其中这样的数据可以由其他系统部件访问,所述其他系统部件包括 CPU102 或另一个并行处理子系统 112。

[0041] PPU202 可配备有任何容量(amount)的本地并行处理存储器 204,包括没有本地存储器,并且可以以任何组合方式使用本地存储器和系统存储器。例如,在统一存储器架构(UMA)实施例中, PPU202 可以是图形处理器。在这样的实施例中,将不提供或几乎不提供专用的图形(并行处理)存储器,并且 PPU202 会以排他或几乎排他的方式使用系统存储器。在 UMA 实施例中, PPU202 可集成到桥式芯片中或处理器芯片中,或作为具有高速链路(例如, PCIe)的分立芯片提供,所述高速链路经由桥式芯片或其他通信手段将 PPU202 连接到系统存储器。

[0042] 如上所示,在并行处理子系统 112 中可以包括任何数目的 PPU202。例如,可在单个插卡上提供多个 PPU202、或可将多个插卡连接到通信路径 113、或可将一个或多个 PPU202 集成到桥式芯片中。在多 PPU 系统中的 PPU202 可以彼此同样或不同。例如,不同的 PPU202



可能具有不同数目的处理内核、不同容量的本地并行处理存储器等等。在存在多个 PPU202 的情况下,可并行操作那些 PPU 从而以高于单个 PPU202 所可能达到的吞吐量来处理数据。包含一个或多个 PPU202 的系统可以以各种配置和形式因素来实现,包括台式电脑、笔记本电脑或手持式个人计算机、服务器、工作站、游戏控制台、嵌入式系统等等。

#### [0043] 均衡系数搜索算法

[0044] 图 3 是所发射信号和所接收信号的常规例示。在计算机系统中,信号可以以 1 或 0 的形式沿线(wire)传送到其目的地。信号可能由于其通过信道行进而退化,使得目的地难以确定所接收的位是 1 还是 0。输入信号 10 表示信道的传送端处的数据 1。输入信号 10 包括接近  $T_b$  的脉冲宽度。输入信号 12 表示信道的传送端处的数据 0。输入信号 12 也包括接近  $T_b$  的脉冲宽度。输入信号 10 和 12 显现锐利的、相似的阶跃函数。

[0045] 图 3 的右侧示出了信道的接收端处的两个输出信号。输出信号 20 包括所接收数据 1,并且输出信号 22 包括所接收数据 0。这些输出信号由于码间干扰(ISI)而失真并且因此输出信号不精确地匹配输入信号 10 和 12 的锐利的样子。如果失真在接收端变得太大,那么 1 可能误解为 0 (或反之亦然)并且在传送中将引入错误。

[0046] 图 4 是产生错误的所接收数据和所发射数据的常规例示。波形 30 表示信道输入处的发射数据。波形 30 包括传送在信道上的一系列 1 和 0。波形 30 显现锐利,具有 1 与 0 之间的清晰的跃迁。虚线表示限幅电平,其标志数据 1 和数据 0 之间的界限。波形 40 是在接收端处的信号的表示。由于信道中的干扰,所接收信号可能失真并且不显现像波形 30 所表示的所发射数据那样锐利。波形 50 是波形 40 中的数据的表示。换句话说,当波形 40 转换成 1 和 0 时结果是波形 50。如图 4 所示,波形 50 (所接收数据)不精确地匹配波形 30 (所发射数据)。传送中引入了错误。示出了出错的零 42,其意指传送了 0 而接收器接收了 1。还示出了出错的一 44,其中传送了 1 而接收器接收了 0。当数据迅速在 1 和 0 之间切换时,如波形 30 所示,一些跃迁可能由于所接收信号(波形 40)中的干扰而丢失,这导致所接收数据(波形 50)中的错误(诸如错误 42 和 44)。随着数据连接和数据传送变得更快,这些类型的错误变得更普遍,因为所接收信号没有足够的时间完全从 1 跃迁到 0 或反之亦然。可以实施均衡来帮助防止这些错误。

[0047] 在 PCI Express Gen3 中,均衡包括实现补偿 ISI 以使所接收信号看起来像原始所发射信号的设定。可以实施传送和接收均衡二者。在传送均衡中,传送端处的信号在发送信号之前以是对将由信道所引入的失真的补充的方式被“重塑”。换句话说,重塑可以抵消失真。重塑可以允许接收端更容易地在 1 和 0 之间进行区分。在接收均衡中,信号在接收端处经修复以抵制由信道所引入的失真并且进一步改进信号质量。在 PCI Express Gen3 中,可以实施传送和接收均衡二者。

[0048] 图 5 示出了根据本发明的一个实施例的、在均衡之前和之后传送的接收端处的信号的一个示例。波形 60 示出的信号尚未对其实施均衡。波形 62 示出的信号示出了均衡之后的信号。由于均衡,在波形 62 中可以比在波形 60 中更清晰地看到 1 和 0 之间的跃迁。

[0049] 图 6 是来自在传送信道的接收端处所探测的示波器的眼图 70 的常规例示。测量结果 #5 是眼高。眼高提供信道中的噪声或干扰的测量。随着干扰增加,“眼”似乎闭合并且眼高测量变得较小。随着干扰降低,眼高测量变得较大。因此,眼高可经测量以检查信道上的传送的质量。在均衡调训过程期间,参数的多个集合每个可以使用眼图来测试以找到参

数的最优集合。如很好理解的,眼图 70 中示出的其他测量结果(测量 1-4 和 6-9)可以用来导出信道的其他性能测量。

[0050] 下文是对若干示范性均衡算法的讨论。这些算法可以用来在给定时限内实施均衡。可对这些算法进行修改和变化而不脱离本发明的较宽精神和范围。

[0051] PCIe 中的均衡可以通过找到可用来调谐传送器的最优传送均衡系数来完成。可以指定三个系数(被称为前标记(precursor)、后标记(postcursor)以及主标记(maincursor)),并且系数受下面的等式限制:

[0052] (1) 前标记 + 主标记 + 后标记 = FS

[0053] (2)  $FS - 2 * (\text{前标记} + \text{后标记}) \geq LF$

[0054] (3) 前标记  $\leq FS/4$

[0055] 其中 LF= 低频率以及 FS= 全摆幅,这二者都是用于特定端口的常量。满足这些等式的系数被叫做“合法的”,不满足这些等式的系数被叫做“非法的”。术语“最优系数”意指产生  $10^{-12}$  或更小的误比特率(BER)的系数。典型的计算机系统可能具有几百个均衡系数的合法组合。本公开中所描述的算法和技术可以用来高效地找到系数的合法组合之中的最优系数的集合。

[0056] 图 7 是根据本发明的一个实施例的均衡系数的映射图的例示。诸如该示例的映射图可以用来将系数可视化。映射图上的每个点表示前标记和后标记的组合。因为主标记可使用上文的等式(1)从其他两个系数导出,所以搜索算法仅跟踪前标记和后标记。图 7 示出了 x 轴上的后标记( $C_{+1}$ )和 y 轴上的前标记( $C_{-1}$ )。在该示例中,FS 值是 24,并且系数表达为 FS 的比例。映射图上的阴影的方格表示预设定,虚线表示将合法系数与非法系数分开的界限。

[0057] 映射图上的每个点具有与它相关联的眼高,其表示由该点处的系数所产生的链路质量。该链路质量的评估可以通过状态机来实施,诸如决策反馈均衡器或 DFE。任何合适的均衡器可被使用并且在本公开的某些示范性实施例中可被称作 EQ Train。最后结果是在本公开的某些示范性实施例中被称作“qeye”的得分。qeye 是眼高的测量,并且较高值表示较好链路质量。搜索算法的目标是找到映射图上的最高值,或者可替代地找到满足或超过预定的阈值的值。可以使用除眼高的质测量量标准,诸如眼宽的测量或眼面积的测量。

[0058] 如上所示,常常没有足够的时间来测试映射图上的每个点。在使用 PCIe 的一个示范性实现方案中,每个点可能花费多于 200 微秒来进行评估。评估 250 个点因此将要求接近 50 毫秒,但是 PCIe 规范例如仅允许最多 24 毫秒来找到最优系数。此外,需要尽可能快地完成,使得可以实施其他操作。这里所描述的搜索算法将对短信道(在该处映射图上的许多点是可接受的)以及长信道(在该处仅映射图上的少量点是可接受的)起作用。这里所描述的搜索算法也可以利用映射图的局部区域中的单调递增的 qeye 值来进行更高效的搜索。

[0059] 在本发明的一个实施例中,在大型系数映射图上进行粗粒度搜索以找到很可能具有最高 qeye 的映射图的区域。一旦该分析完成,那么可以对映射图的较小部分实施细粒度搜索来找到最高 qeye。实施粗粒度搜索在短时间内提供对映射图的概览,然后更准确的细粒度搜索可以用来寻求最优系数的集合。

[0060] 可以以各种方式进行粗粒度搜索。图 8A 和 8B 示出了粗粒度搜索的一个示范性实施例。如所示,粗粒度搜索包括四个序列—其中的每一个使用步长 2 迭代地

(iteratively) “行走”映射图。图 8A 示出的第一序列起始于映射图上的坐标(0, 0), 而第二、第三以及第四序列分别起始于(1, 1)、(0, 1) 以及(1, 0)。第一序列以下面的次序覆盖映射图上的阴影的点:(0, 0)、(0, 2)、(0, 4)、(0, 6)、(2, 6)、(2, 4)、(2, 2)、(2, 0)、(4, 0)、(4, 2)、(4, 4)、(6, 2)、(6, 0)、(8, 0)。图 8B 示出了第二序列, 其从(1, 1) 开始并且以步长 2 行走映射图, 如所示。还进行序列三和四(未示出)。通过将粗粒度搜索分解成四个序列, 算法能够在搜索的早期覆盖大面积的映射图, 然后如果时间许可则覆盖整个映射图。如果粗粒度搜索在所有四个序列完成之前用完时间, 那么直到该点所找到的最佳点及时被存储并且传递到细粒度搜索。

[0061] 可以针对粗粒度搜索建立退出条件。在一个示范性实施例中, 如果下面的事件中的任何一个发生, 那么粗粒度搜索可以退出:

[0062] (1) 已经找到超过最大可接受 qeye 阈值的点;

[0063] (2) 迭代的数目已超过预定的值;

[0064] (3) 粗粒度搜索的总时间已超过预定的值; 或者

[0065] (4) 已经测试所有所选择的粗粒度序列中所指定的所有点。

[0066] 一旦退出条件发生, 则映射图上的最佳点用作细粒度搜索的起始点。

[0067] 图 9 是根据本发明的一个实施例的、用于对均衡系数的映射图实施粗粒度搜索的另一个技术的例示。在该示范性实施例中, 粗粒度搜索将每个轴划分成 2、4 或 8 个部分以将系数映射图分成接近相等的区域。然后选择每个区域的中心点并且检查与这些中心点相关联的系数的合法性。确定每个合法点的 qeye, 具有最高 qeye 的点被选择为“最佳粗点”并且用作细粒度搜索的起始点。

[0068] 在一些实施例中, 除粗粒度搜索的技术可以用来选择用于细粒度搜索的起始点。这些技术的使用可能比实施粗粒度搜索更快。例如, 软件程序或进程可以用来指定系数的一个集合以进行评估。系数的所指定集合可能是先前用于系统中的均衡的系数的集合。在一些封闭系统中, 诸如膝上型计算机, 最优均衡系数可能在每个启动循环(boot cycle) 期间不改变, 因此这些系数可以用作细粒度搜索的起始点或者如果与系数相关联的眼高满足可接受阈值则甚至可以在没有细粒度搜索的情况下被重新使用。

[0069] 图 10 是根据本发明的一个实施例的、用于对均衡系数的映射图实施细粒度搜索的技术的例示。上文所描述的粗粒度搜索用来找到要用作细粒度搜索的起始点的“最佳粗点”。现在将描述用于细粒度搜索的一个示范性算法。所评估的系数的每个集合被叫做算法的“调谐尝试”或“迭代”。细粒度搜索算法通过评估起始点周围邻近的系数开始。然后, 算法一次一个点地向具有最高眼高的邻近的点的方向“行走”。与最大眼高相关联的相邻、或邻近的点可被称作当前横断点(currently traversed point)。向该方向(第一方向)行走以及继续沿途评估每个点直到眼高开始下降或直到到达界限。在眼高开始下降(或到达界限)的映射图上的点处, 算法评估映射图上的八个周围的点(就界限而言更少)。算法然后又向具有最高眼高的点的方向(第二方向)行走, 然后重复这些行走以及探索步骤直到满足退出条件。四个潜在退出条件是:

[0070] (1) 所有八个周围的点较低—算法已到达局部最大值;

[0071] (2) 时间用完;

[0072] (3) 迭代的数目已到达预定的阈值; 或者

[0073] (4) 眼高到达可接受阈值。

[0074] 此外,图 10 提供细粒度搜索的详细的示例。假定粗粒度搜索选择了点(3,5)作为起始点。图 10 中的映射图上的数字仅是示例性的并且用来表示与映射图上的每个点相关联的均衡系数的眼高。在起始点(3,5)处的数字是 5。算法评估围绕该起始点的邻近的系数,并且找到具有所有邻近的点的最高眼高一眼高 6 的点(4,5)。以该方式评估邻近的点可被称作“探索模式”。算法然后一次一步地向点(4,5)的方向行走,直到眼高开始下降。该方向可被认为是第一方向,点(4,5)可被称作当前横断点。沿映射图行走可被称作“行走模式”。在点(8,5)处眼高是 10,在点(9,5)处眼高是 9。算法因此将重新集中(re-center)在眼高开始下降的点即点(8,5)周围。算法然后评估围绕映射图上的点(8,5)的八个点,并且找到具有眼高 12 的点(8,6)是具有最高眼高的点。所以算法从点(8,5)向点(8,6)的方向(即第二方向)行走,并且继续直到眼高开始下降。在点(8,10)处眼高是 19,并且在点(8,11)处下降到 14。再一次,算法这一次重新集中在点(8,10)周围并且评估围绕点(8,10)的八个点。在该评估之后,算法找到所有八个周围的点的眼高比点(8,10)处的眼高更低。因此,算法已满足退出条件并且点(8,10)处的均衡系数被选择用于系统中使用。该示例中的细粒度搜索结束。

[0075] 在细粒度搜索期间,可以首先检查算法遭遇的映射图上的系数的每个集合的合法性。如果在探索模式中时遭遇非法点,那么算法可以跳过该点继续前进。如果在行走模式中时遭遇非法点,那么算法可以在该点处停止行走并且切换到探索模式。

[0076] 上文所描述的粗粒度搜索和细粒度搜索可以以各种方式和各种系统实现。例如高速串行通信协议如 PCIe 可以使用这些技术来准备数据信号用于传送。PCIe 3.0 基础规范不解决均衡搜索算法。

[0077] 在涉及 PCIe 3.0 的一个示范性实现方案中,恢复.均衡(Recovery.Equalization)是链路调训和状况状态机(Link Training and Status State Machine)的恢复状态的子状态。该子状态用来为以 8.0GT/s 速度的适当操作找到最佳的传送系数。均衡系数可以通过使用握手协议由硬件自动地进行确定。因此,恢复.均衡调训算法用来在最短可能时间中找到最佳的系数。

[0078] 恢复.均衡子状态进一步分成五个子子状态(sub-substate):

[0079] (1) 恢复.均衡阶段(phase) 0

[0080] (2) 恢复.均衡阶段 1

[0081] (3) 恢复.均衡阶段 2Req Coeff

[0082] (4) 恢复.均衡阶段 2EQ Train

[0083] (5) 恢复.均衡阶段 3

[0084] 恢复.均衡阶段 0 仅为上行端口(即端点)而保留。当下行端口(即根端口)请求进入恢复.均衡时,端点(endpoint)首先进入该状态。在进入该状态之前,端点将根端口(rootport)建议的均衡系数应用到其传送器(根端口建议的预设定在第一速度改变到 gen3 之前由根端口以 gen1/gen2 速度来传达)。用于根端口建议的预设定的原因是,因为大部分线路(trace)在主板上,所以根端口制造商将处于较好位置来预测最优预设定将是什么,因此制造商将指定作为开始的预设定。该制造商规范可能允许搜索算法较早定留(settle)在最优点处。

[0085] 恢复. 均衡阶段 1 用于上行和下行端口二者。这两个端口将根端口建议的预设定应用到其传送器。端口还经由调训设定(TS1)彼此传达其 FS 和 LF 值,使得每个端口可将这些值插入上文所指定的约束等式中。该步骤使每个端口能够在其各自的“主”阶段(即用于上行端口的阶段 2、用于下行端口的阶段 3)中仅搜索合法系数。

[0086] 在恢复. 均衡阶段 2Req Coeff 中,上行端口(端点)对下行端口(根端口)发送请求以将其 Tx (传送)设定设定到上行端口认为将是最优的值,以及等待请求被接受或拒绝。上行端口可注意仅请求合法系数,但是也可能具有处置合法系数被另一侧拒绝的情况的机制。上行端口可以通过指定单独系数(前标记、主标记、后标记)来请求所要应用的均衡设定。

[0087] 在恢复. 均衡阶段 2EQ Train 中,上行端口评估由恢复. 均衡阶段 2Req Coeff 中所请求的系数所产生的链路质量。该评估可由均衡器来实施,最后结果是 qeye。端口然后记录该值,并且将它与较早的请求中所看到的 qeye 值相比较—如果这是到目前为止的最高一个,那么端口将所请求的系数存储在临时变量中,所述临时变量诸如(best\_precursor、best\_maincursor、best\_postcursor)。

[0088] 如果需要试用更多系数,那么下一个状态是恢复. 均衡阶段 2Req Coeff,或者如果最大 qeye 比可接受阈值更大(或者如果算法不能找到任何其他值得试用的点),那么下一个状态是恢复. 均衡阶段 3。在该实施例中,恢复. 均衡阶段 2Req Coeff 与恢复. 均衡阶段 2EQ Train 之间的迭代可以花费的最大时间是 24ms。在 24ms 用完之后,端口跃迁到恢复. 均衡阶段 3。

[0089] 在恢复. 均衡阶段 3 中,下行端口请求上行端口改变上行端口的 Tx 设定,下行端口试图找到上行端口到下行端口方向的链路的最佳 qeye。该阶段具有 32ms 的超时(timeout)。上行端口仅仅不得不接收请求,反映回他们以及当他们合法时指示“已接受”(或当他们非法时指示“拒绝”),以及将合法的请求应用到它自己的传送器设定。

[0090] 恢复. 均衡调训算法仅适用于子子状态恢复. 均衡阶段 2Req Coeff 以及恢复. 均衡阶段 2EQ Train,这是因为仅这些状态被分配到端点用于实施系数搜索。所试用的系数的每个集合被叫做算法的“调谐尝试”或“迭代”。当满足某些条件时,可能发生来自恢复. 均衡阶段 2Request Coeff 的跃迁。当 EQ Train 信号变高时,可能出现从恢复. 均衡阶段 2EQ Train 到恢复. 均衡阶段 2Req Coeff 的跃迁。

[0091] 下文描述了包含粗和细方法以及其他方法的搜索算法的一个示范性实施例。该示范性实施例可宽广地分类为五个子算法:

[0092] (1)请求先前最佳设定—如果“最佳系数”变量具有有效值,那么再次请求这些值。如果链路再次进入恢复. 均衡状态并且用户想要测试先前均衡搜索所找到的点,那么该子算法是有用的。

[0093] (2)SBIOS 预设定请求—在进入 gen3 之前请求根端口所推荐的预设定。

[0094] (3)软件指定的系数请求—软件经由寄存器应用可能最优的系数的集合。

[0095] (4)粗粒度算法—以某一步幅(或 step\_size)搜索系数空间以找到高 qeye 值的区域;或者实施任何其他类型的粗粒度算法,如上文所描述的。

[0096] (5)细粒度算法—在先前步骤中揭露的最高点附近进行搜索,但将 step\_size 限定到 1。总是在具有 qeye 中的增加的最高梯度的方向上前进,如上文所描述的。

[0097] (6) 最后的步骤是重新请求先前 5 个步骤中所找到的最佳点。该步骤是一迭代 (one-iteration) 通过步骤。

[0098] 图 11 是根据本发明的一个实施例的、用于实施多道信道均衡调训的方法步骤的流程图。尽管结合图 1、2 以及 7-10 描述了方法步骤,但是本领域的技术人员将理解的是,配置为以任何次序实施方法步骤的任何系统均落在本发明的范围内。处理单元 102 配置为当执行存储在存储器诸如系统存储器 104 中的软件应用时实施方法 1100 的各步骤。在一些实施例中,并行处理子系统 112 可以实现方法 1100 的一些步骤。

[0099] 如所示,方法 110 开始于步骤 1110,在该处处理单元 102 对均衡系数集实施第一遍测试。第一遍测试产生一个或多个经过滤的均衡系数集,其每个满足某一标准。第一遍测试可以包括如上文所描述的粗粒度搜索算法中的一个。第一遍测试可以例如确定与均衡系数的每个集合相关联的眼高以及至少部分地基于该眼高过滤均衡系数集。

[0100] 在步骤 1120,处理单元 102 对一个或多个经过滤的均衡系数集实施第二遍测试以确定最后的均衡系数集。第二遍测试产生比第一遍测试更准确的结果。第二遍测试可能花费比第一遍测试更长的时间来实施。此外,第二遍测试可以包括如上文所描述的细粒度搜索算法。第二遍测试可以例如确定与每个均衡系数集相关联的眼高以及至少部分地基于眼高来选择或拒绝均衡系数集。

[0101] 图 12 是根据本发明的一个实施例的、用于实施粗粒度搜索的方法步骤的流程图。尽管结合图 1、2 以及 7-10 描述了方法步骤,但是本领域的技术人员将理解的是,配置为以任何次序实施方法步骤的任何系统均落在本发明的范围内。处理单元 102 配置为当执行存储在存储器诸如系统存储器 104 中的软件应用时实施方法 1200 的各步骤。在一些实施例中,并行处理子系统 112 可以实施方法 1200 的一些步骤。

[0102] 过程开始于步骤 1210。处理器 102 执行软件应用来选择均衡系数的映射图上的起始点。在图 8A 示出的示例性实施例中,选择的起始点是 (0, 0)。在其他实施例中可以选择其他点。

[0103] 在步骤 1220,处理器 102 执行软件应用以使用大于一步的步长沿映射图迭代地行走。在图 8A 示出的示例性实施例中,步长是 2。在其他实施例中可以使用其他步长。在示例性实施例中,技术从点 (0, 0) 行走到 (0, 2), 然后到点 (0, 4)、(0, 6)、(2, 6) 等,如上文关于图 8A 所描述的。该方法通过使用大于一步的步长可以在短时间内覆盖大面积的映射图。

[0104] 在步骤 1230,处理器 102 执行软件应用来测量每个点的眼高。具有最高眼高的点存储在存储器中,并且其他点也可以存储在存储器中。可以使用用于测量眼高的任何合适的过程。具有最高眼高的点可以用作最佳系数或者可以用作另一个搜索算法诸如细粒度搜索的起始点。

[0105] 在步骤 1240,处理器 102 执行软件应用以如果退出条件发生则退出方法。可以使用若干潜在退出条件。如果已经找到超过最大可接受 qeye 阈值的点,那么第一退出条件发生。如果该退出条件发生,那么已经找到最佳系数的集合,并且这些系数可经选择以及利用用于传送。如果迭代的数目已超过预定的值,那么第二退出条件发生。可以利用预定的迭代的最大数目来实施搜索已不超过用于粗粒度搜索的时限。如果到达迭代的数目,那么搜索过程可以退出。如果粗粒度搜索中的总时间已超过预定的值,那么另一个退出条件可能发生。如果所有所选择的粗粒度序列中所指定的所有点已被测试,那么第四退出条件可能

发生。当退出条件发生时,那么处理器 102 可以执行软件应用以选择所找到的最佳的均衡系数(如由眼高所测量的),以及或者使用这些系数用于均衡或者将这些系数作用于细粒度搜索的起始点。

[0106] 在步骤 1250,如果没有退出条件发生,那么处理器 102 执行软件应用以通过在步骤 1210 处起始以及使用不同起始点来重复方法。作为示例性实施例,过程可以使用起始点(0,0)用于第一序列,然后使用起始点(0,1)用于第二序列。第二序列随首次通过使用大于一步的步长沿映射图行走、测量眼高以及如果退出条件发生则退出而精确地进行。如果退出条件尚未得到满足,那么任何数目的附加的序列可能发生。

[0107] 图 13 是根据本发明的一个实施例的、用于实施粗粒度搜索的方法步骤的流程图。尽管结合图 1、2 以及 7-10 描述了方法步骤,但是本领域的技术人员将理解的是,配置为以任何次序实施方法步骤的任何系统均落在本发明的范围内。处理单元 102 配置为当执行存储在存储器诸如系统存储器 104 中的软件应用时实施方法 1300 的各步骤。在一些实施例中,并行处理子系统 112 可以实施方法 1300 的一些步骤。

[0108] 过程开始于步骤 1310。在该步骤中,处理器 102 执行软件应用以将均衡系数的映射图划分成多个区域。在图 9 中可以看到以该方式划分映射图的示例。映射图可以划分成接近相等大小的区域或者划分成不等大小的区域。映射图可以划分成任何适当数目的区域。

[0109] 在步骤 1320,处理器 102 执行软件应用以测量每个区域中的一个点的眼高。可以以各种方式选择每个区域中被测量的点。在一些实施例中,可以选择区域的中心点。在其他实施例中,可以选择随机或伪随机点。使用任何合适的技术来测量眼高。

[0110] 在步骤 1330,处理器 102 执行软件应用以选择具有最高眼高的点作为用于细粒度搜索算法或另一个搜索算法的起始点。理想地,选择具有最高眼高的点作为起始点导致比以另一个点起始更迅速地找到最佳的均衡系数集。

[0111] 图 14 是根据本发明的一个实施例的、用于实施细粒度搜索的方法步骤的流程图。尽管结合图 1、2 以及 7-10 描述了方法步骤,但是本领域的技术人员将理解的是,配置为以任何次序实施方法步骤的任何系统均落在本发明的范围内。处理单元 102 配置为当执行存储在存储器诸如系统存储器 104 中的软件应用时实施方法 1400 的各步骤。在一些实施例中,并行处理子系统 112 可以实施方法 1400 的一些步骤。

[0112] 技术开始于步骤 1410。在步骤 1410,处理单元 102 选择均衡系数的映射图上的起始点。在一个示范性实施例中,均衡系数的映射图可以与图 7 示出的映射图类似。起始点可以通过任何合适的技术选择,包括上文所描述的粗粒度搜索算法中的一个或子算法中的一个。

[0113] 在步骤 1420,处理单元 102 执行软件应用以测量起始点以及毗邻起始点的每个点的眼高。这些眼高然后可以用来搜索映射图上的最佳的均衡系数集。任何适合的方法可以用来测量眼高。

[0114] 在步骤 1430,处理器 102 执行软件应用以在毗邻具有最高眼高的起始点的点的方向沿均衡系数的映射图行走。上文关于图 10 讨论该过程。

[0115] 在步骤 1440,处理器 102 执行软件应用以确定眼高是否仍在增加。如果眼高仍在增加,那么方法返回到步骤 1430 并且继续在具有最高眼高的点的方向上行走。只要眼高正

在增加(或直到某些退出条件得到满足,如下文所讨论),就重复步骤 1430 和 1440。如果眼高仍未增加,那么方法进行到步骤 1450。

[0116] 在步骤 1450,处理器 102 执行软件应用以选择眼高从其开始下降的点以及测量每个相邻点的眼高。上文关于图 7 也讨论了该过程。这些眼高用来确定行走的新方向以及确定退出条件是否得到满足。过程然后继续到步骤 1460。

[0117] 在步骤 1460,处理器 102 执行软件应用以确定退出条件是否已被满足。可以使用若干潜在退出条件,如上文关于图 10 所讨论的。在某些实施例中,如果每个相邻点具有比眼高开始下降的点更低的眼高,那么方法已找到局部最大值并且这可以结束细粒度搜索算法。另一个退出条件可以是预定的实际阈值是否已满足。第三个退出条件可以是眼高是否高于预定的阈值。如果是,该眼高满足系统的要求并且这些系数可由处理器 102 选择用于使用。当算法的迭代的数目已到达某一阈值时,第四个退出条件可被满足。如果退出条件尚未得到满足,那么方法返回到步骤 1430 以继续在具有最高眼高的点的方向上行走。如果退出条件已得到满足,那么方法继续到步骤 1470。

[0118] 在步骤 1470,处理器 102 执行软件应用以在已经分析的系数的所有集合中选择具有最高眼高的系数。上文在步骤 1460 中所描述的退出条件的至少一个已被满足,并且系统可以以其他启动过程继续。理想地,被选择的均衡系数将降低所传送的信号中的错误率。

[0119] 总而言之,两遍式方法可以用来找到以及选择最佳的均衡系数集用于高速数据传送。第一遍包括映射图之上的粗粒度搜索,所述映射图包括在每个点处包括均衡系数集的点。粗粒度搜索的目标是选择可用于高速数据传送或可用作用于第二遍的起始点的均衡系数集。可以以各种方式实施粗粒度搜索,包括将系数映射图划分成若干区域并且测试每个区域中的一个点,或者围绕映射图迭代地行走以及测试映射图上的点的子集。被实施的测试可以包括测量与均衡系数的每个集合相关联的眼高。第二遍包括映射图之上的细粒度搜索。细粒度搜索涉及测量起始点的眼高以及毗邻起始点的每个点的眼高。搜索算法然后在具有最高眼高的点方向沿映射图行走,并且只要眼高保持增加就继续行走。如果眼高将下降,那么算法重新集中在具有最高眼高的点上并且评估所有相邻的点。算法然后在具有最高眼高的点的方向上行走并且继续细粒度搜索直到退出条件得到满足。一旦退出条件得到满足,则算法选择具有最高眼高的均衡系数以及使用这些系数用于沿高速数据连接的传送的均衡。

[0120] 有利地,使用上述技术选择均衡系数虑及对满足由系统所要求的质量标准的系数的较快选择。在一些系统中,诸如服务器,信道长并且可能极其嘈杂。因此,可接受系数的子集非常小。上文所描述的搜索算法在通过首先标识最佳系数可能位于的均衡系数的映射图的区域以及然后通过实施对该区域实施彻底搜索以找到最佳的系数方面是有效的。在短信道系统(诸如台式计算机)中,大量点可以满足操作条件。本文所描述的算法可以迅速地进行粗粒度搜索并且然后使用细粒度搜索对映射图上的最佳点进行定位。细粒度搜索可以在八个可能方向中的任何一个方向上行走并且这有助于迅速地找到最佳点。此外,一些系统诸如膝上型计算机具有不可改变的嵌入式链路,这意味着均衡系数也不期望进行改变。本公开提供了将已知质量均衡系数直接指派为用于搜索的起始点的软件。在质量点上或附近开始的搜索常常可以非常迅速地完成。

[0121] 其他优势包括通过更改步长、退出条件或其他变量来细调谐上文所讨论的搜索算



法的能力。上文所讨论的算法可以处置合法点和非法点二者。也可以调节为粗粒度搜索和 / 或细粒度搜索所分配的时间。

[0122] 本发明的一个实施例可被实施为与计算机系统一起使用的程序产品。该程序产品的程序定义实施例的各功能(包括本文中描述的方法)并且可以被包含在各种计算机可读存储介质上。示例性计算机可读存储介质包括但不限于:(i)不可写的存储介质(例如,计算机内的只读存储器设备,诸如可由CD-ROM驱动器读取的CD-ROM盘、闪存、ROM芯片或任何类型的固态非易失性半导体存储器),在其上存储永久性信息;和(ii)可写的存储介质(例如,磁盘驱动器或硬盘驱动器内的软盘或者任何类型的固态随机存取半导体存储器),在其上存储可更改的信息。

[0123] 以上已参照特定实施例对本发明进行了描述。然而,本领域技术人员将理解的是,可对此做出各种修改和变化而不脱离如随附权利要求书中所阐述的本发明的较宽精神和范围。因此,前面的描述以及附图应被视为是例示性而非限制性的意义。

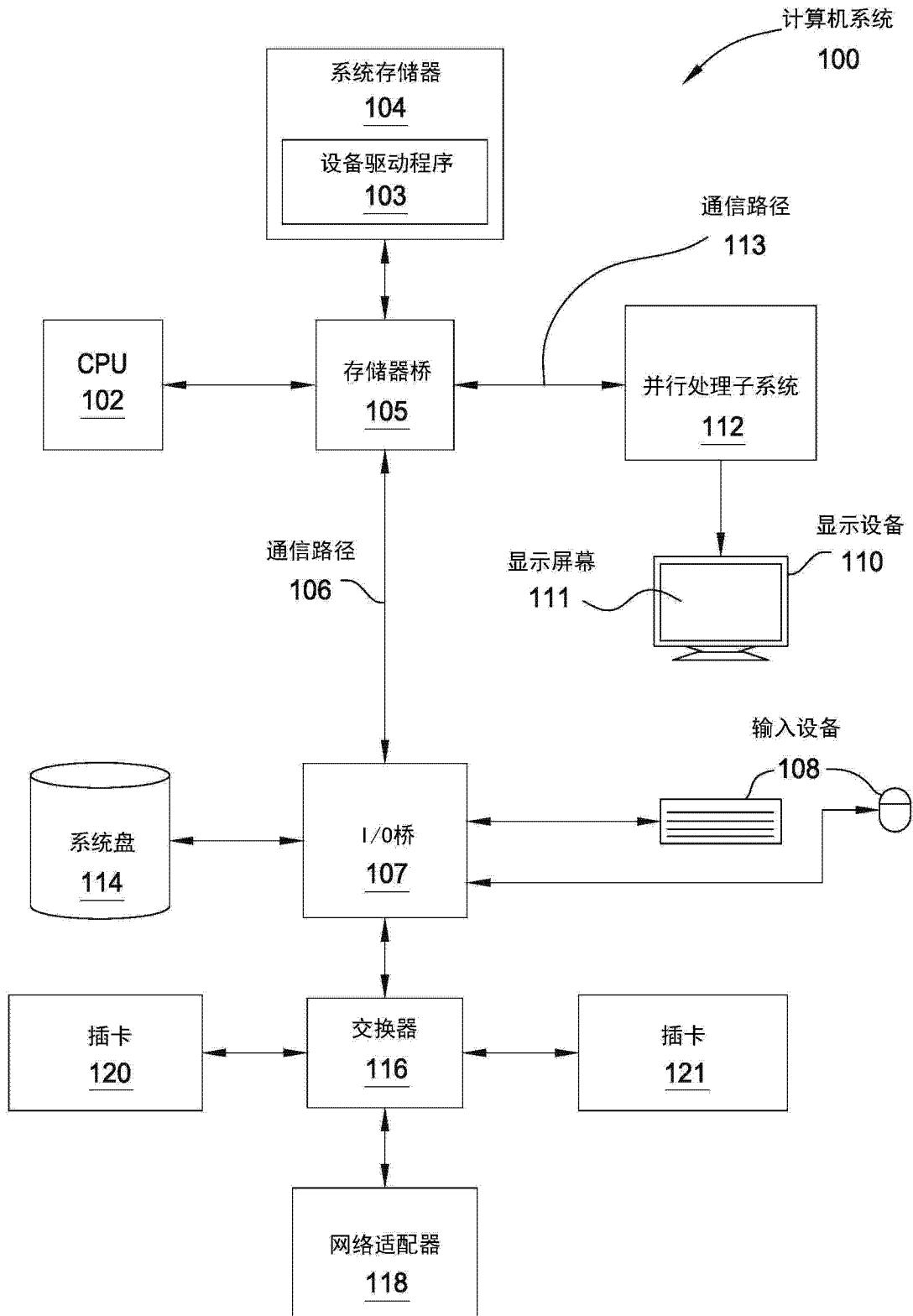


图 1

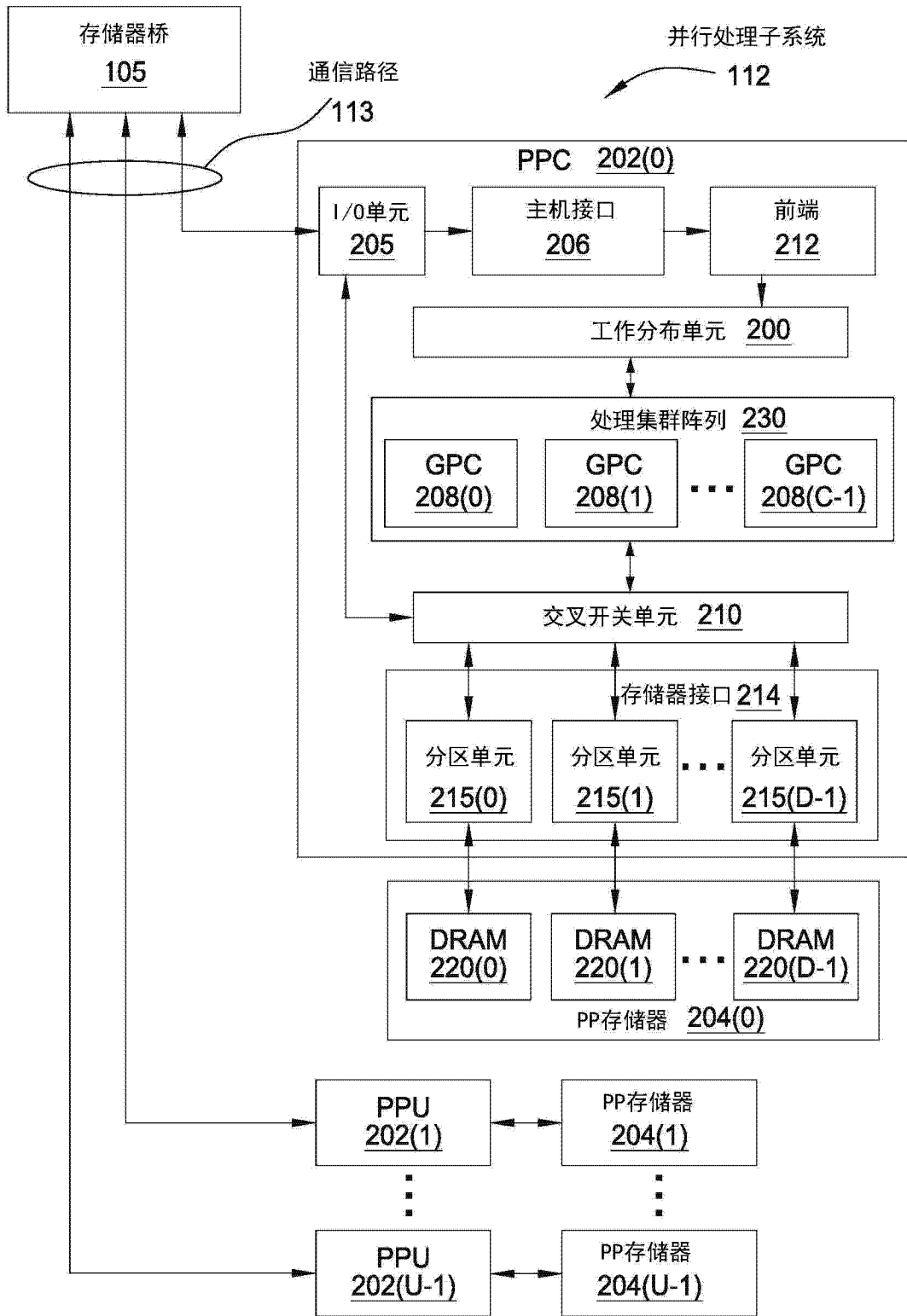


图 2

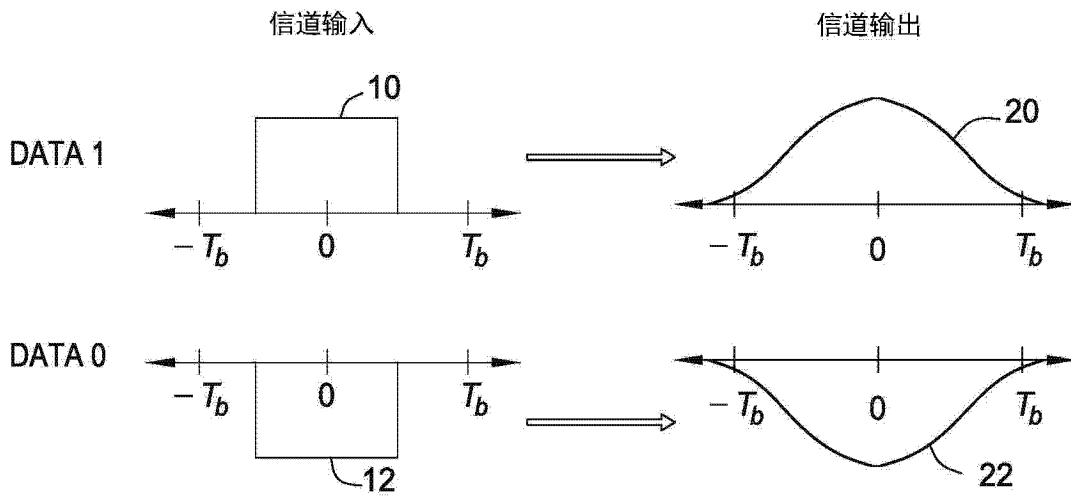


图 3

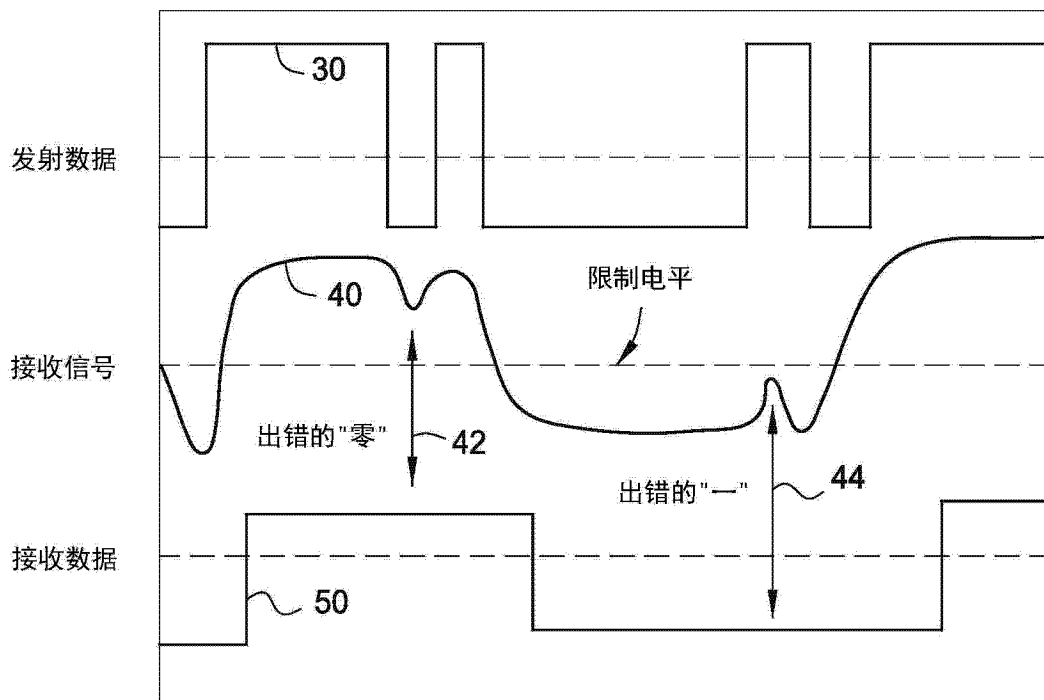


图 4

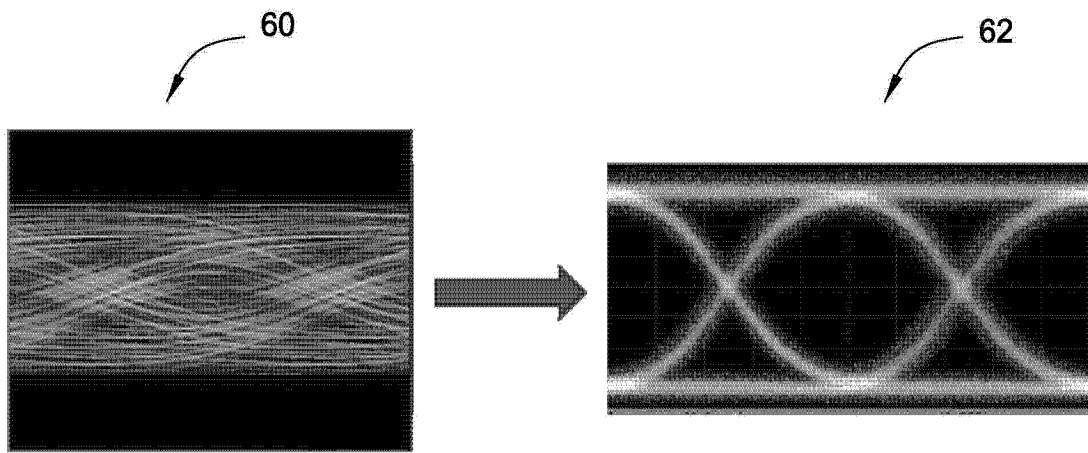


图 5

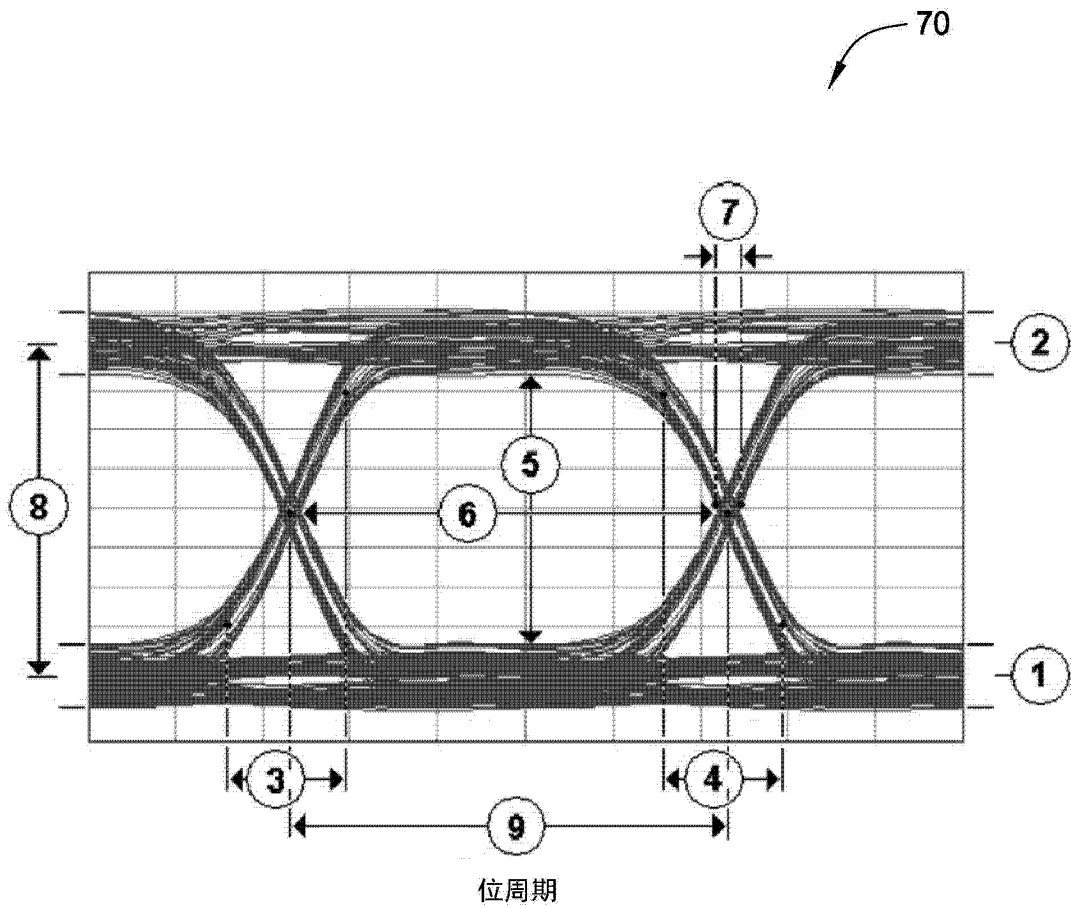


图 6

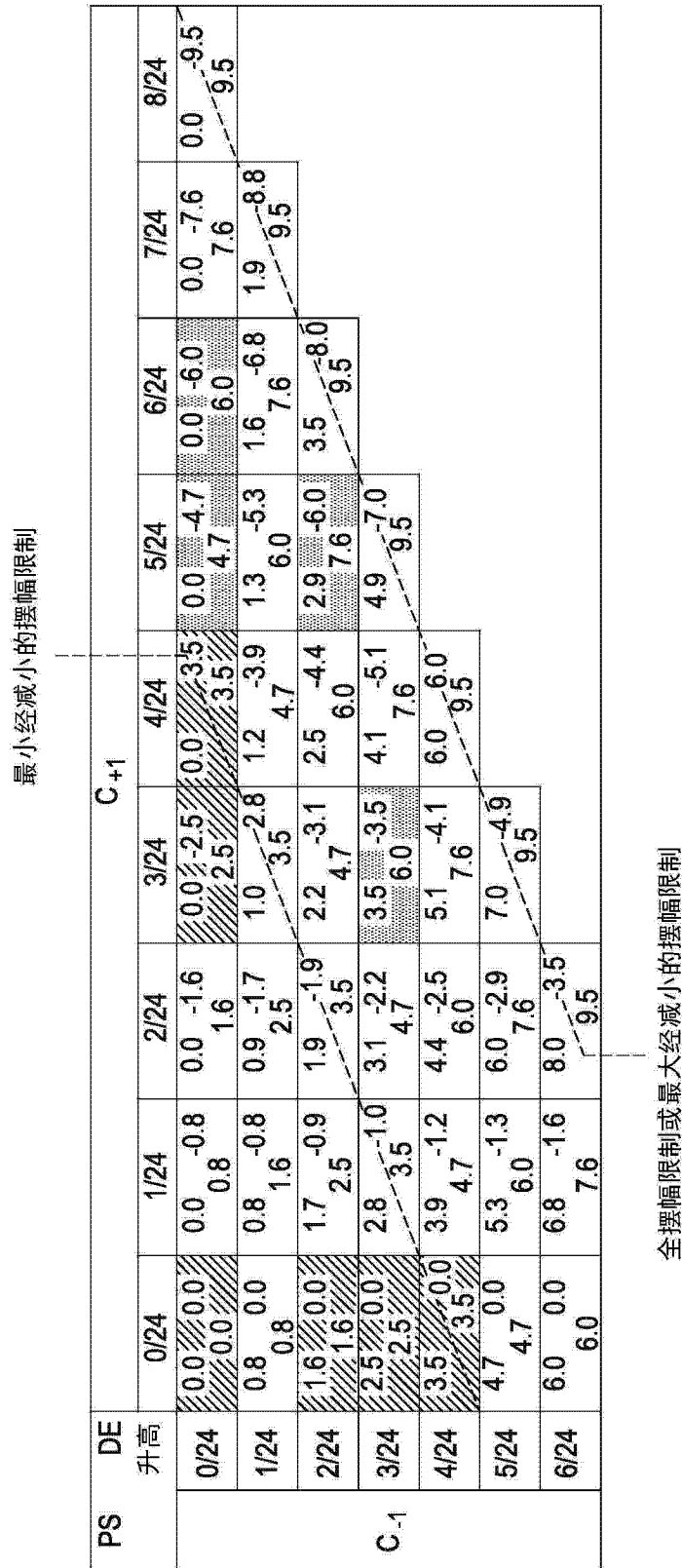


图 7

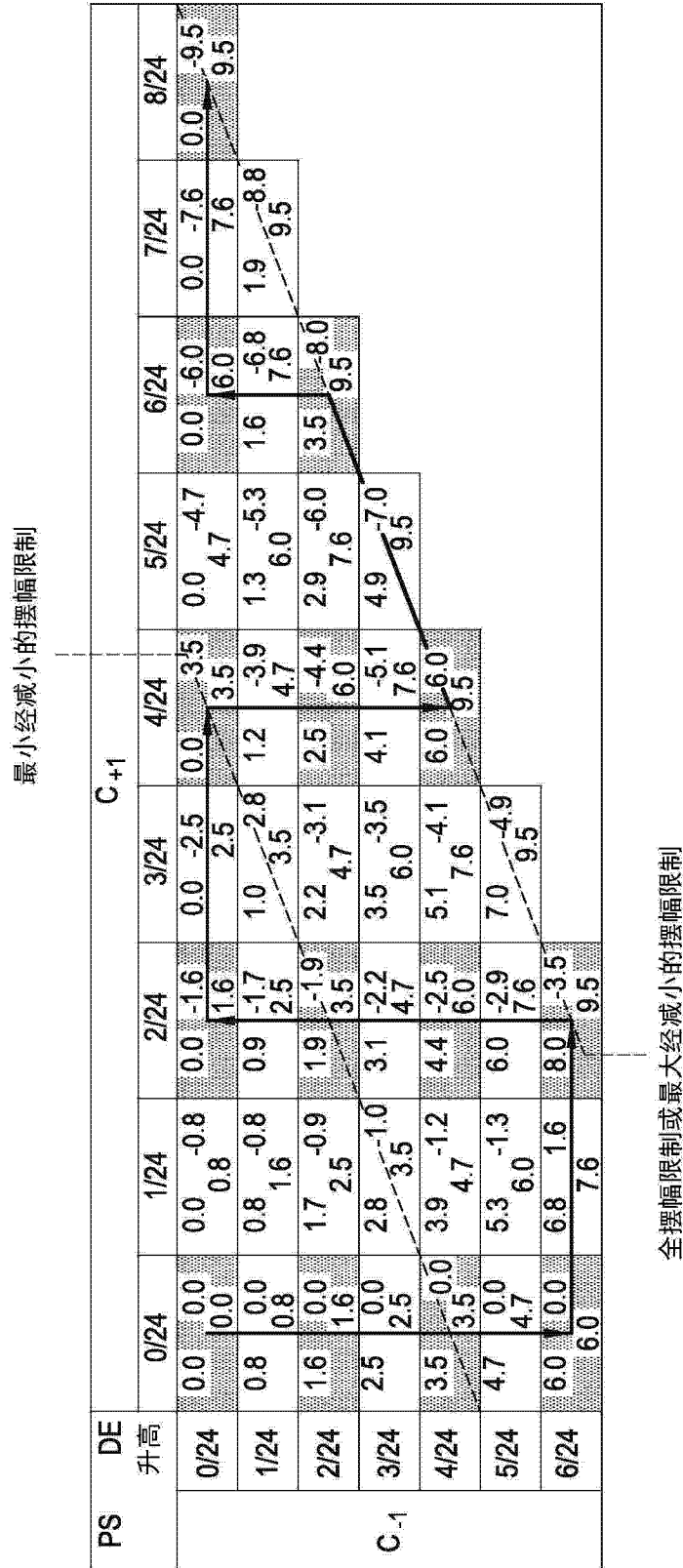


图 8A

最小经减小的摆幅限制

PS	DE 升高	C+1																	
		0/24	1/24	2/24	3/24	4/24	5/24	6/24	7/24	8/24	9/24	10/24	11/24	12/24	13/24	14/24	15/24		
C-1	0/24	0.0	0.0	0.0	-0.8	0.0	-1.6	0.0	-2.5	0.0	3.5	0.0	-4.7	0.0	-6.0	0.0	-7.6	0.0	-9.5
	1/24	0.8	0.0	0.8	-0.8	0.9	-1.7	1.0	-2.8	1.2	-3.9	1.3	-5.3	1.6	-6.8	1.9	-8.8	1.9	-9.5
	2/24	1.6	0.0	1.6	-0.9	1.9	-1.9	2.2	-3.1	2.5	-4.4	2.9	-6.0	3.5	-8.0	3.5	-9.5	3.5	-9.5
	3/24	2.5	0.0	2.5	-1.0	3.1	-2.2	3.5	-3.5	4.1	-5.1	4.9	-7.0	4.9	-7.0	4.9	-7.0	4.9	-7.0
	4/24	3.5	0.0	3.5	-1.2	4.4	-2.5	5.1	-4.1	6.0	-6.0	6.0	-6.0	6.0	-6.0	6.0	-6.0	6.0	-6.0
	5/24	4.7	0.0	4.7	-1.3	6.0	-2.9	7.0	-4.9	7.6	-7.6	7.6	-7.6	7.6	-7.6	7.6	-7.6	7.6	-7.6
6/24	6.0	0.0	6.0	1.6	8.0	-3.5	9.5	-9.5	9.5	-9.5	9.5	-9.5	9.5	-9.5	9.5	-9.5	9.5	-9.5	

图 8B

全摆幅限制或最大经减小的摆幅限制

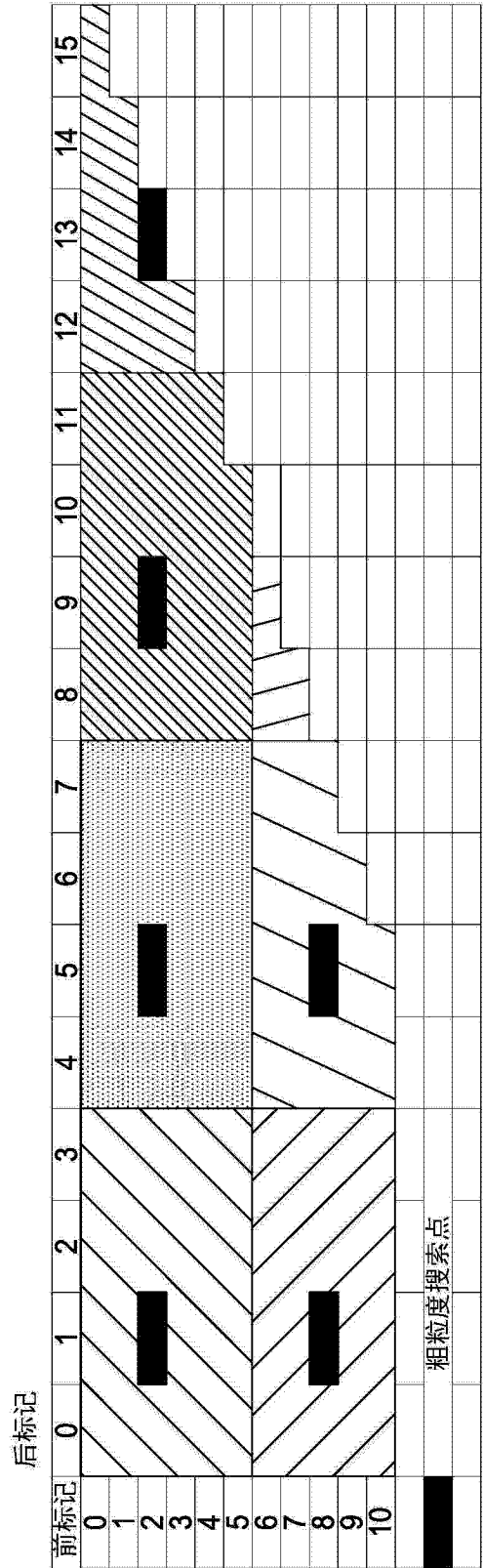


图 9



后标记	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
前标记																	
	1																
	2																
	3																
	4		3	3	4			8	6	7							
	5		4	5	6	7	8	9	10	9	8						
	6		2	1	4			11	12	11							
	7								14								
	8								17								
	9																
	10							11	18	16							
	11							12	19	3							
	12							13	14	16							
	13																
	14																
	15																
	16																
	17																

图 10

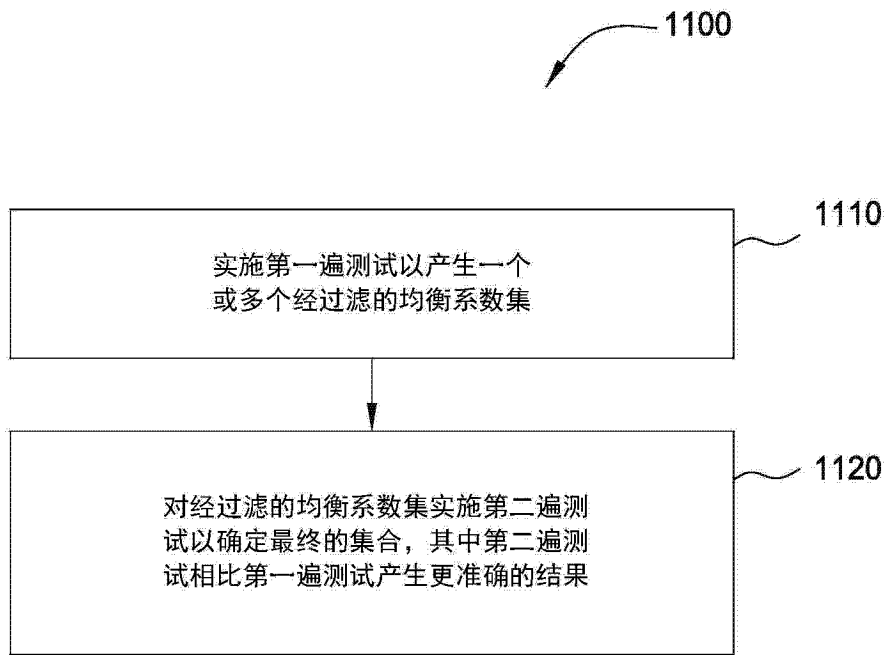


图 11

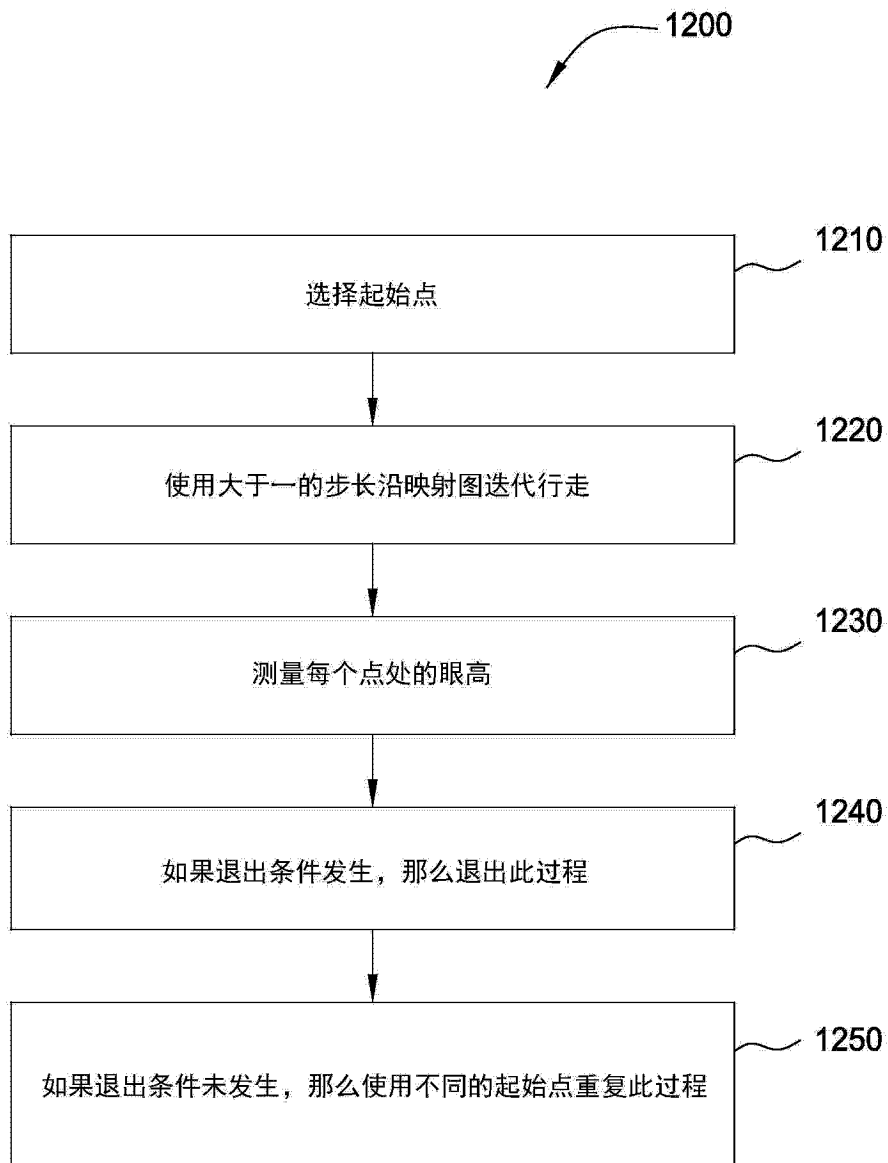


图 12

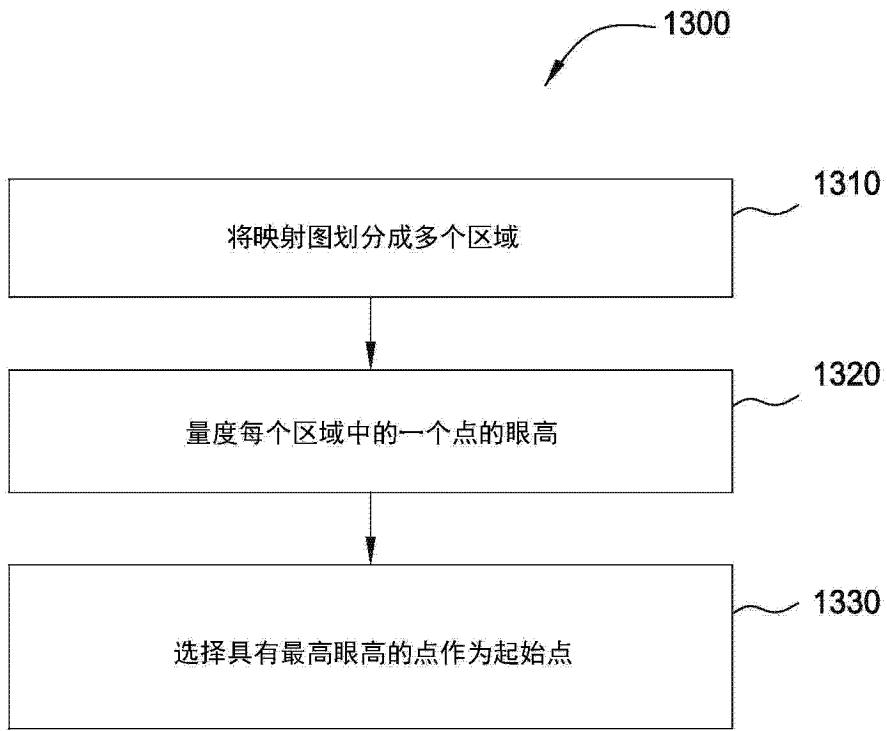


图 13

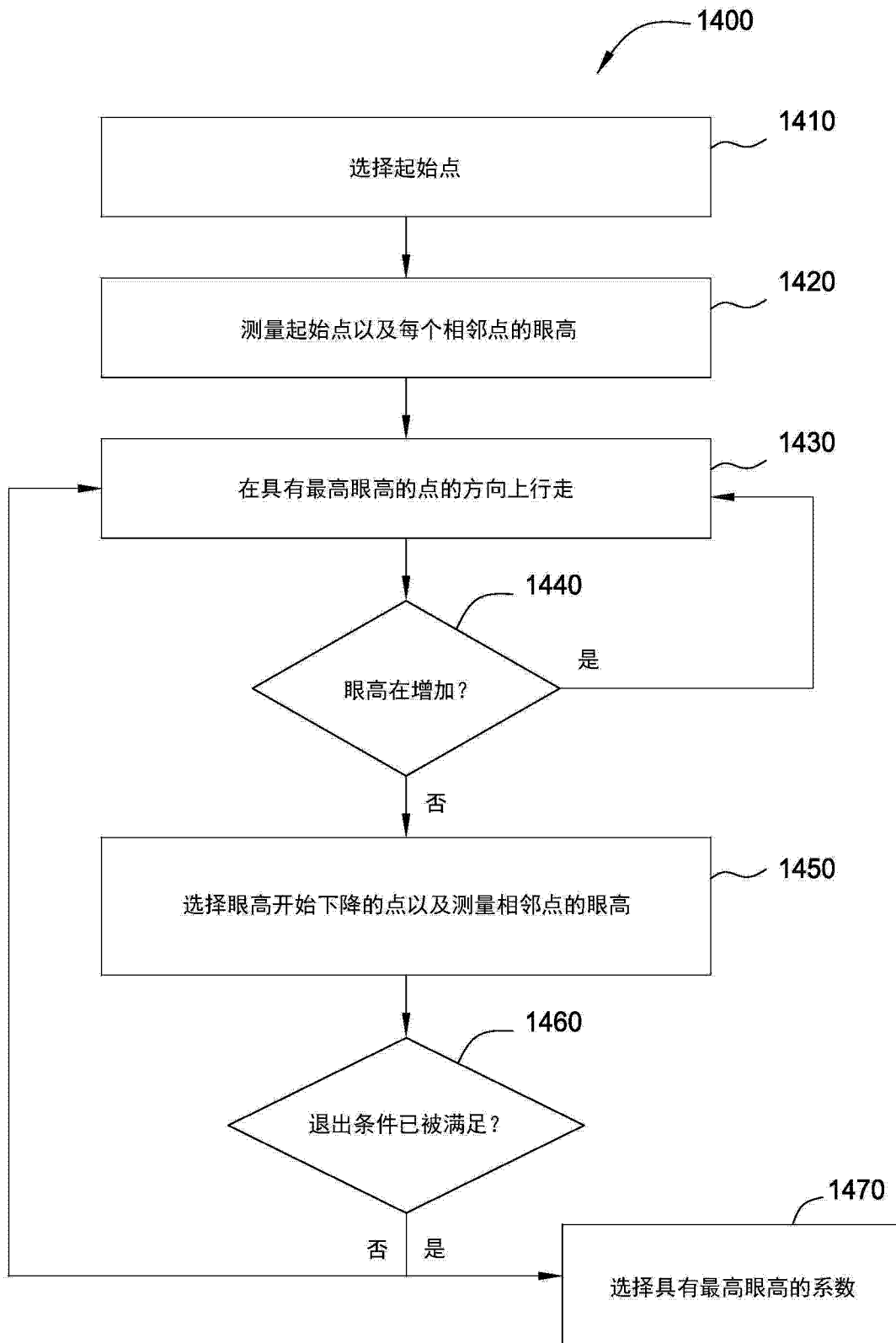


图 14