



(12) 发明专利

(10) 授权公告号 CN 109074341 B

(45) 授权公告日 2023. 04. 18

(21) 申请号 201780025415.6

(22) 申请日 2017.04.14

(65) 同一申请的已公布的文献号
申请公布号 CN 109074341 A

(43) 申请公布日 2018.12.21

(30) 优先权数据
62/340,750 2016.05.24 US
15/283,310 2016.10.01 US

(85) PCT国际申请进入国家阶段日
2018.10.24

(86) PCT国际申请的申请数据
PCT/US2017/027720 2017.04.14

(87) PCT国际申请的公布数据
W02017/204922 EN 2017.11.30

(73) 专利权人 英特尔公司
地址 美国加利福尼亚

(72) 发明人 M·珍 D·弗勒利克
D·D·夏尔马 B·坦南特
Q·德温 S·W·利姆

(74) 专利代理机构 永新专利商标代理有限公司
72002
专利代理师 刘瑜 王英

(51) Int.Cl.
G06F 13/40 (2006.01)

(56) 对比文件
CN 101189577 A, 2008.05.28
US 6718417 B1, 2004.04.06

审查员 罗煜川

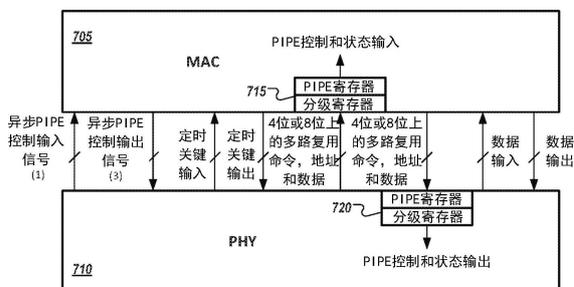
权利要求书3页 说明书26页 附图9页

(54) 发明名称

减少引脚计数接口

(57) 摘要

提供了一种装置,包括一组寄存器和计算块的接口。计算块包括物理层块或介质访问控制层块中的一个。该接口包括用于发送异步信号的一个或多个引脚、用于接收异步信号的一个或多个引脚,以及一组引脚,用于传送特定信号以访问该组寄存器,其中定义接口的一组控制和状态信号被映射到该组寄存器的相应位。



700

1. 一种用于数据通信的装置,所述装置包括:
介质访问层MAC电路装置;以及
接口,所述接口用于耦合到包括物理层PHY电路装置的设备,其中,所述接口包括用于基于快速外围组件互联PIPE架构的接口的PHY接口,并且所述接口包括:
用于向所述PHY电路装置发送数据的一组发送引脚;
用于从所述PHY电路装置接收数据的一组接收引脚;以及
用于实现消息总线接口的一组引脚,
其中,所述MAC电路装置用于经由所述消息总线接口访问与所述PHY电路装置相关联的一组消息总线寄存器,所述一组消息总线寄存器中的相应位表示多个控制和状态信号,所述一组消息总线寄存器中的一特定位表示所述多个控制和状态信号中的一特定信号,并且所述MAC电路装置用于将值写入所述一特定位以发起与所述一特定信号相关联的操作。
2. 如权利要求1所述的装置,其中,所述接口缺乏用于所述多个控制和状态信号的专用引脚。
3. 如权利要求1所述的装置,其中,所述多个控制和状态信号包括基于PIPE的信号。
4. 如权利要求1所述的装置,其中,所述装置还包括与所述MAC电路装置相关联的一组消息总线寄存器,并且所述设备用于使用所述消息总线接口来访问与所述MAC电路装置相关联的所述一组消息总线寄存器。
5. 如权利要求4所述的装置,其中,所述MAC电路装置用于从所述设备接收用于将值写入第二组消息总线寄存器的一特定位的写入命令,以向所述MAC电路装置指示由所述第二组消息总线寄存器的所述一特定位表示的对应信号。
6. 如权利要求1所述的装置,其中,多个基于PIPE的信号包括针对所述接口定义的信号的子集,并且所述信号的子集包括没有关键定时要求的信号。
7. 如权利要求1所述的装置,其中,所述MAC电路装置使用已提交写入来将所述值写入所述一特定位。
8. 如权利要求7所述的装置,其中,所述MAC电路装置用于响应于所述已提交写入,通过所述消息总线接口接收来自所述PHY电路装置的确认。
9. 如权利要求1所述的装置,其中,所述MAC电路装置用于与实现协议栈中更高层的逻辑进行通信。
10. 如权利要求1所述的装置,其中,所述一组消息总线寄存器包括12位存储器空间。
11. 一种用于数据通信的装置,所述装置包括:
物理层PHY电路装置;以及
基于快速外围组件互联PIPE架构的接口,所述基于快速外围组件互联PIPE架构的接口用于耦合到包括介质访问层MAC电路装置的设备,其中,所述基于快速外围组件互联PIPE架构的接口包括消息总线接口,所述消息总线接口包括:
用于发送命令以访问与所述MAC电路装置相关联的第一消息总线寄存器的一组发送引脚,其中,所述第一消息总线寄存器中的位表示第一组信号;以及
用于接收命令以允许所述MAC电路装置访问与所述PHY电路装置相关联的第二消息总线寄存器的一组接收引脚,其中,所述第二消息总线寄存器中的位表示第二组信号,
其中,所述PHY电路装置用于将值写入所述第一消息总线寄存器以访问与所述第一组

信号相关联的功能,所述PHY电路装置用于响应于由所述MAC电路装置将值写入所述第二消息总线寄存器而发起操作,并且所述操作对应于所述第二组信号。

12.如权利要求11所述的装置,其中,所述消息总线寄存器实现与所述第一组信号和所述第二组信号相关联的功能,而不是实现所述第一组信号和所述第二组信号作为所述基于快速外围组件互联PIPE架构的接口中的专用信号。

13.如权利要求11所述的装置,其中,所述第一组信号或所述第二组信号中的一个包括状态和控制信号。

14.如权利要求11所述的装置,其中,所述一组接收引脚在所述消息总线接口的接收方向上实现8位,并且所述一组发送引脚在所述消息总线接口的发送方向上实现8位。

15.如权利要求11所述的装置,其中,一组4位命令被定义用于使用所述消息总线接口来访问所述第一消息总线寄存器和所述第二消息总线寄存器。

16.如权利要求11所述的装置,其中,所述装置包括具有所述PHY电路装置的知识产权块。

17.一种用于数据通信的系统,所述系统包括:

包括物理层PHY电路装置和第一组消息总线寄存器的第一设备,其中,所述第一组消息总线寄存器包括用于表示第一组信号的第一位;

包括介质访问层MAC电路装置和第二组消息总线寄存器的第二设备,其中,所述第二组消息总线寄存器包括用于表示第二组信号的第二位;以及

用于将所述第一设备耦合到所述第二设备的接口,其中,所述接口包括:

用于将第一数据从所述第二设备发送到所述第一设备的第一组通路;

用于将第二数据从所述第一设备发送到所述第二设备的第二组通路;以及

消息总线接口,所述消息总线接口包括:用于实现对所述第二组消息总线寄存器的访问的第三组通路;以及用于实现对所述第一组消息总线寄存器的访问的第四组通路,

其中,所述第一设备用于在所述第三组通路上向所述第二设备发送用于写入所述第二组寄存器中的所述位的命令,以发起与所述第二组信号相关联的操作,并且所述第二设备用于在所述第四组通路上向所述第一设备发送用于写入所述第一组寄存器中的所述位的命令,以发起与所述第一组信号相关联的操作。

18.如权利要求17所述的系统,其中,所述PHY电路装置用于建立链路,所述第一数据要在所述链路上被发送,并且所述第二数据要在所述链路上被接收。

19.如权利要求18所述的系统,其中,所述链路包括符合以下各项中的一项的链路:基于快速外围组件互联PCIe的协议、基于串行ATA SATA的协议、或者基于通用串行总线USB的协议。

20.如权利要求17所述的系统,其中,所述第一设备包括第一IP块,并且所述第二设备包括第二IP块。

21.如权利要求17所述的系统,其中,所述第一组信号和所述第二组信号各自包括根据PIPE规范定义的相应的信号。

22.如权利要求21所述的系统,其中,所述第一组信号和所述第二组信号中的每一个是使用所述第一组寄存器和所述第二组寄存器来实现的,而不是通过所述接口中的专用引脚来实现的。

23. 如权利要求17所述的系统,还包括:基于SATA的存储器设备,所述基于SATA的存储器设备包括固态存储器驱动器和基于SATA的端口,以经由链路连接到主机,其中,所述链路符合基于SATA的协议。

24. 如权利要求23所述的系统,还包括:主机,其中,所述主机包括处理器核心。

25. 如权利要求17所述的系统,其中,所述第三组通路和所述第四组通路各自相应地包括八条通路。

减少引脚计数接口

[0001] 相关申请的交叉引用

[0002] 本申请要求于2016年10月1日提交的美国专利申请序号15/283、310的优先权,该申请要求于2016年5月24日提交的美国临时专利申请序号62/340、750的优先权。在先申请的公开内容被认为是本申请的公开内容的一部分,并且通过引用方式将其全部内容并入本文。

技术领域

[0003] 本公开涉及计算系统,并且特别地(但非排他地)涉及计算机接口。

背景技术

[0004] 半导体工艺和逻辑设计的进步允许增加集成电路器件上可能存在的逻辑量。作为必然结果,计算机系统配置已经从系统中的单个或多个集成电路发展到存在于各个集成电路上的多个核、多个硬件线程和多个逻辑处理器,以及集成在这些处理器内的其他接口。处理器或集成电路通常包括单个物理处理器管芯,其中处理器管芯可包括任何数量的核、硬件线程、逻辑处理器、接口、存储器、控制器集线器等。随着处理能力随着计算系统中设备的数量而增长,插槽和其他设备之间的通信变得更加关键。因此,互连已经从主要处理电通信的更传统的多点总线发展到促进快速通信的全面互连架构。不幸的是,随着对未来处理器以更高速率消耗相应需求的需求被放在现有互连架构的能力上。互连架构可以基于多种技术,包括快速外围组件互连(PCIe)、通用串行总线等。

附图说明

[0005] 图1示出了包括互连架构的计算系统的实施例。

[0006] 图2示出了包括分层堆栈的互连架构的实施例。

[0007] 图3示出了要在互连架构内生成或接收的请求或分组的实施例。

[0008] 图4示出了用于互连架构的发射器和接收器对的实施例。

[0009] 图5示出了PHY/MAC接口的表示。

[0010] 图6示出了PIPE PHY/MAC接口的第一版本的表示。

[0011] 图7示出了PIPE PHY/MAC接口的第二基于寄存器的版本的表示。

[0012] 图8示出了示例PHY/MAC接口的示例状态和控制寄存器的一部分的表示。

[0013] 图9示出了说明涉及示例PHY/MAC接口的寄存器的示例事务的信令图。

[0014] 图10是示出了示例PHY/MAC接口内的示例技术的流程图。

[0015] 图11示出了包括多核处理器的计算系统的框图的实施例。

[0016] 图12示出了计算系统的框图的另一实施例。

具体实施方式

[0017] 在以下描述中,阐述了许多具体细节,诸如特定类型的处理器和系统配置、特定硬

件结构、特定架构和微架构细节、特定寄存器配置、特定指令类型、特定系统组件、特定测量/高度、特定的处理器流水线阶段和操作等的示例,以便提供对本发明的透彻理解。然而,对于本领域技术人员显而易见的是,不需要采用这些具体细节来实施本发明。在其他实例中,没有详细描述众所周知的组件或方法,例如特定和替代处理器架构、用于所描述的算法的特定逻辑电路/代码、特定固件代码、特定互连操作、特定逻辑配置、特定制造技术和材料、特定编译器实现、算法在代码中的特定表达、特定的断电和门控技术/逻辑以及计算机系统的其他特定操作细节,以免不必要地模糊本发明。

[0018] 尽管可以参考特定集成电路中(例如在计算平台或微处理器中)的节能和能量效率来描述以下实施例,但是其他实施例也适用于其他类型的集成电路和逻辑设备。本文描述的实施例的类似技术和教导可以应用于其他类型的电路或半导体器件,其也可以受益于更好的能量效率和节能。例如,所公开的实施例不限于台式计算机系统或Ultrabooks™。并且还可以用于其他设备,例如手持设备、平板电脑、其他薄型笔记本电脑、片上系统(SOC)设备和嵌入式应用。手持设备的一些示例包括蜂窝电话、互联网协议设备、数码相机、个人数字助理(PDA)和手持PC。嵌入式应用通常包括微控制器、数字信号处理器(DSP)、片上系统、网络计算机(NetPC)、机顶盒、网络集线器、广域网(WAN)交换机或可执行下面讲述的功能和操作的任何其他系统。此外,这里描述的装置、方法和系统不限于物理计算设备,还可以涉及用于节能和效率的软件优化。

[0019] 随着计算系统的发展,其中的组件变得更加复杂。结果,在组件之间耦合和通信的互连架构的复杂性也在增加,以确保针对最佳组件操作满足带宽要求。此外,不同的细分市场需要互连架构的不同方面来满足市场需求。例如,服务器需要更高的性能,而移动生态系统有时会牺牲整体性能以节省电力。然而,大多数结构的独特目的是提供尽可能最高的性能和最大功率节省。下面,讨论了许多互连,这些互连将潜在地受益于本文所述的本发明的各方面。

[0020] 一种互连结构架构包括外围组件互连(PCI)快速(PCIe)架构。PCIe的主要目标是使来自不同供应商的组件和设备能够在开放式架构中互操作,跨越多个细分市场;客户端(台式机和移动)、服务器(标准和企业)以及嵌入式和通信设备。快速PCI是一种高性能、通用I/O互连,针对各种未来的计算和通信平台而进行定义。一些PCI属性(例如,其使用模型、加载-存储架构和软件接口)已通过其修订版进行维护,而先前的并行总线实现方式已被高度可扩展的完全串行接口所取代。最近的快速PCI版本利用了点对点互连、基于交换机的技术和分组协议的优势以提供更高水平的性能和特征。快速PCI支持的一些高级特征包括电源管理、服务质量(QoS)、热插拔/热交换支持、数据完整性和错误处理。

[0021] 参见图1,示出了由互连一组组件的点对点链路组成的结构的实施例。系统100包括耦合到控制器集线器115的处理器105和系统存储器110。处理器105包括任何处理元件,例如微处理器、主处理器、嵌入式处理器、协处理器或其他处理器。处理器105通过前端总线(FSB)106耦合到控制器集线器115。在一个实施例中,FSB 106是如下所述的串行点对点互连。在另一实施例中,链路106包括符合不同互连标准的串行差分互连架构。

[0022] 系统存储器110包括任何存储器设备,例如随机存取存储器(RAM)、非易失性(NV)存储器或系统100中的设备可访问的其他存储器。系统存储器110通过存储器接口116耦合到控制器集线器115。存储器接口的示例包括双倍数据速率(DDR)存储器接口、双通道DDR存

存储器接口和动态RAM (DRAM) 存储器接口。

[0023] 在一个实施例中,控制器集线器115是快速外围组件互连 (PCIe或PCIE) 互连层级中的根集线器、根联合体或根控制器。控制器集线器115的示例包括芯片组、存储器控制器集线器 (MCH)、北桥、互连控制器集线器 (ICH)、南桥和根控制器/集线器。通常,术语芯片组指的是两个物理上分离的控制器集线器,即耦合到互连控制器集线器 (ICH) 的存储器控制器集线器 (MCH)。注意,当前系统通常包括与处理器105集成的MCH,而控制器115以与下面描述的类似方式与I/O设备进行通信。在一些实施例中,可选地通过根联合体115支持对等路由。

[0024] 这里,控制器集线器115通过串行链路119耦合到交换机/桥接器120。输入/输出模块117和121 (也可以称为接口/端口117和121) 包括/实现分层协议栈以提供控制器集线器115和交换机120之间的通信。在一个实施例中,多个设备能够耦合到交换机120。

[0025] 交换机/桥120将分组/消息从设备125向上游 (即,朝向根联合体顺着层次结构向上) 路由到控制器集线器115并且向下游 (即从远离根控制器顺着层次结构向下) 从处理器105或系统存储器110路由到设备125。在一个实施例中,交换机120被称为多个虚拟PCI到PCI桥设备的逻辑组件。设备125包括要耦合到电子系统的任何内部或外部设备或组件,例如I/O设备、网络接口控制器 (NIC)、附加卡、音频处理器、网络处理器、硬盘驱动器、存储设备、CD/DVD ROM、监视器、打印机、鼠标、键盘、路由器、便携式存储设备、火线设备、通用串行总线 (USB) 设备、扫描仪和其他输入/输出设备。通常在PCIe术语中,例如设备,被称为端点。尽管没有具体示出,但是设备125可以包括PCIe到PCI/PCI-X桥,以支持传统或其他版本的PCI设备。PCIe中的端点设备通常被分类为传统、PCIe或根联合体集成端点。

[0026] 图形加速器130还通过串行链路132耦合到控制器集线器115。在一个实施例中,图形加速器130耦合到MCH,而MCH耦合到ICH。然后将交换机120和I/O设备125耦合到ICH。I/O模块131和118还用于实现分层协议栈以在图形加速器130和控制器集线器115之间进行通信。类似于上面的MCH讨论,图形控制器或图形加速器130本身可以集成在处理器105中。应当理解,图1中所示的组件 (例如,105、110、115、120、125、130) 中的一个或多个可以得以增强以执行、存储和/或体现用于实现本文描述的特征中的一个或多个的逻辑。

[0027] 转到图2,图2示出了分层协议栈的实施例。分层协议栈200包括任何形式的分层通信栈,例如快速路径互连 (QPI) 栈、PCIe栈、下一代高性能计算互连栈或其他分层栈。尽管下面参考图1-图4进行的讨论涉及PCIe栈,但是相同的概念可以应用于其他互连栈。在一个实施例中,协议栈200是PCIe协议栈,包括事务层205、链路层210和物理层220。接口,例如图1中的接口117、118、121、122、126和131,可以表示为通信协议栈200。作为通信协议栈的表示也可以称为实现/包括协议栈的模块或接口。

[0028] 快速PCI使用分组来在组件之间传递信息。在事务层205和数据链路层210中形成分组以将信息从发送组件携带到接收组件。当发送的分组流经其他层时,它们会扩展为处理这些层的分组所需的附加信息。在接收侧,发生反向过程,并且分组从其物理层220表示变换为数据链路层210表示,并且最终 (对于事务层分组) 变换为可以由接收设备的事务层205处理的形式。

[0029] 事务层

[0030] 在一个实施例中,事务层205用于提供设备的处理核心和互连架构 (例如,数据链

路层210和物理层220)之间的接口。在这方面,事务层205的主要职责是分组的汇编和反汇编(即事务层分组或TLP)。事务层205通常管理针对TLP的基于信用的流控制。PCIe实现分离事务,即具有由时间分隔的请求和响应的事务,允许链接在目标设备收集针对响应的数据时携带其他业务。

[0031] 此外,PCIe利用基于信用的流控制。在该方案中,设备在事务层205中针对每个接收缓冲器通告初始信用量。在链路的相对端处的外部设备,例如图1中的控制器集线器115,对由每个TLP消费的信用数进行计数。如果事务未超过信用额度,则可以发送事务。在接收到响应后,将恢复一定数量的信用。信用方案的一个优点是,如果没有遇到信用额度,信用返还的延迟不会影响性能。

[0032] 在一个实施例中,四个事务地址空间包括配置地址空间、存储器地址空间、输入/输出地址空间和消息地址空间。存储器空间事务包括将数据传送到存储器映射位置或从存储器映射位置传送数据的读请求和写请求中的一个或多个。在一个实施例中,存储器空间事务能够使用两种不同的地址格式,例如短地址格式,例如32位地址,或长地址格式,例如64位地址。配置空间事务用于访问PCIe设备的配置空间。用于配置空间的事务包括读请求和写请求。定义消息事务以支持PCIe代理之间的带内通信。

[0033] 因此,在一个实施例中,事务层205组装分组报头/有效载荷156。当前分组报头/有效载荷的格式可以在PCIe规范网站的PCIe规范中找到。

[0034] 快速参考图3,示出了PCIe事务描述符的实施例。在一个实施例中,事务描述符300是用于携带事务信息的机制。在这方面,事务描述符300支持系统中的事务的识别。其他潜在用途包括跟踪默认事务排序的修改以及事务与通道的关联。

[0035] 事务描述符300包括全局标识符字段302、属性字段304和通道标识符字段306。在所示示例中,全局标识符字段302被描绘为包括本地事务标识符字段308和源标识符字段310。在一个实施例中,全局事务标识符302对于所有未完成的请求是唯一的。

[0036] 根据一种实现方式,本地事务标识符字段308是由请求代理生成的字段,并且对于需要完成该请求代理的所有未完成请求它是唯一的。此外,在该示例中,源标识符310在PCIe层次结构内唯一地标识请求者代理。因此,与源ID 310一起,本地事务标识符308字段提供层次结构域内的事务的全局标识。

[0037] 属性字段304指定事务的特征和关系。在这方面,属性字段304可能用于提供允许修改事务的默认处理的附加信息。在一个实施例中,属性字段304包括优先级字段312、保留字段314、排序字段316和非窥探字段318。这里,优先级子字段312可以由发起者修改以向事务分配优先级。保留属性字段314保留用于将来或供应商定义的用途。可以使用保留属性字段来实现使用优先级或安全性属性的可能使用模型。

[0038] 在该示例中,排序属性字段316用于提供传达可以修改默认排序规则的排序类型的可选信息。根据一个示例实现方式,排序属性“0”表示要应用默认排序规则,其中排序属性“1”表示放宽排序,其中写入可以在相同方向上传递写入,并且读取完成可以在相同方向上传递写入。窥探属性字段318用于确定是否窥探了事务。如图所示,通道ID字段306标识与事务相关联的通道。

[0039] 链路层

[0040] 链路层210(也称为数据链路层210)充当事务层205和物理层220之间的中间级。在

一个实施例中,数据链路层210的职责是提供用于在链路的两个组件之间交换事务层分组(TLP)的可靠机制。数据链路层210的一侧接受由事务层205组装的TLP,应用分组序列标识符211,即标识号或分组号,计算并应用错误检测码,即CRC 212,并将修改的TLP提交给物理层220用于跨物理到外部设备的传输。

[0041] 物理层

[0042] 在一个实施例中,物理层220包括逻辑子块221和电子块222,以物理地将分组发送到外部设备。这里,逻辑子块221负责物理层221的“数字”功能。在这方面,逻辑子块包括用于准备用于由物理子块222发送的传出信息的发送部分,以及用于在将所接收的信息传递到链路层210之前识别和准备所接收的信息的接收部分。

[0043] 物理块222包括发射器和接收器。发射器由逻辑子块221提供符号,发射器将符号串行化并发送到外部设备。接收器被提供有来自外部设备的串行符号,并将接收的信号转换为位流。位流被反序列化并提供给逻辑子块221。在一个实施例中,采用8b/10b传输码,其中发送/接收10位符号。这里,特殊符号用于利用帧223对分组进行成帧。此外,在一个示例中,接收器还提供从传入串行流中恢复的符号时钟。

[0044] 如上所述,尽管参考PCIe协议栈的特定实施例讨论了事务层205、链路层210和物理层220,但是分层协议栈不限于此。实际上,可以包括/实现任何分层协议。作为示例,表示为分层协议的端口/接口包括:(1)用于组装分组的第一层,即事务层;用于对分组进行排序的第二层,即链路层;以及用于发送分组的第三层,即物理层。作为具体示例,使用公共标准接口(CSI)分层协议。

[0045] 接下来参考图4,示出了PCIe串行点对点结构的实施例。尽管示出了PCIe串行点对点链路的实施例,但是串行点对点链路不限于此,因为它包括用于发送串行数据的任何传输路径。在所示的实施例中,基本PCIe链路包括两个低压差分驱动信号对:发送对406/412和接收对411/407。因此,设备405包括用于将数据发送到设备410的传输逻辑406以及用于从设备410接收数据的接收逻辑407。换句话说,在PCIe链接中包含两个发送路径,即路径416和417,以及两个接收路径,即路径418和419。

[0046] 传输路径指的是用于发送数据的任何路径,例如传输线、铜线、光学线、无线通信通道(channel)、红外通信链路或其他通信路径。两个设备(例如设备405和设备410)之间的连接被称为链路,例如链路415。链路可以支持一个通路(lane)-每个通路代表一组差分信号对(一对用于传输,一对用于接收)。为了对带宽进行缩放,链路可以聚合由xN表示的多个通路,其中N是任何支持的链路宽度,例如1、2、4、8、12、16、32、64或更宽。在一些实现方式中,每个对称通路包含一个发送差分对和一个接收差分对。非对称通路可包含不等比率的发送和接收对。一些技术可以利用对称通路(例如,PCIe),而其他技术(例如,显示端口)可以不包括并且甚至可以仅包括仅发送或仅包括接收对,以及其他示例。

[0047] 差分对指的是两个发送路径,例如线路416和417,以用于发送差分信号。作为示例,当线416从低电压电平切换到高电压电平,即上升沿时,线路417从高逻辑电平驱动到低逻辑电平,即下降沿。差分信号潜在地表现出更好的电特性,例如更好的信号完整性,即交叉耦合、电压过冲/下冲、振铃等。这允许更好的定时窗口,这实现更快的传输频率。

[0048] 在一些实现方式中,数据链路层或逻辑物理层可以包括控制器或体现介质访问控制(MAC)层。在一些实现方式中,物理(PHY)层(例如,其逻辑和/或物理结构)可以作为单独

的知识产权 (IP) 或计算块提供,其可以与提供硬件逻辑的其他部分的其他计算块耦合以实现互连堆栈。为了实现这样的实现方式,可以提供接口以连接计算块,同时仍然在所得到的互连上支持特定的互连协议(或者可能是多个不同的互连协议)(例如,由互连的计算块提供)。作为示例,已经开发了用于快速PCI架构(PIPE)的PHY接口以定义这样的接口。实际上,PIPE已经被扩展以在现在多种不同互连技术中实现控制器与PHY之间的接口,所述多种不同互连技术不仅包括PCIe,还包括SATA和USB架构。PIPE旨在开发功能相当的快速PCI、SATA和USB PHY。因此,PHY可以作为分立的集成芯片封装(IC)或宏单元来传送,以包含在ASIC设计或其他系统中。该规范定义了一组PHY功能,这些功能必须包含在符合PIPE的PHY中。PIPE被定义为在这种PHY与介质访问层(MAC)和/或链路层ASIC之间提供标准接口。标准化PHY接口(如PIPE)可以提供ASIC和端点设备供应商可以开发的接口。

[0049] 图5是示出了PHY与MAC层(例如,实现为两个或更多个不同的计算块(或“IP块”))之间的定义的接口505的简化框图500。在该示例中,通过物理通道510发送或接收的数据由PHY层逻辑515处理。在一个示例中,PHY层逻辑515可以包括物理介质附接(PMA)层和物理编码子层(PCS)。PMA可以包括模拟缓冲器、串行器/解串器(SERDES)和接口(到通道510)(例如,10位或130位接口),以及其他示例逻辑和元件。PCS可以包括编码/解码逻辑(例如,8b/10b编码/解码或128b/130b编码/解码),弹性缓冲器和接收器检测逻辑,以及其他示例逻辑和元件。在该示例中,PHY层逻辑515的全部或一部分可以体现在与体现链路层逻辑520的全部或一部分的另一计算块分开的计算块上。链路层逻辑520可以包括控制器逻辑,例如介质访问控制(MAC)层和与协议栈中的更高链路和事务层的接口。在一个示例中,MAC层可以包括用于链路训练、流控制和状态的状态机、加扰和解扰逻辑以及其他示例逻辑和元件。而图5的示例示出了MAC层作为链路层的一部分,在其他实现方式中,MAC层(或其他控制器层)可以是逻辑PHY层的一部分,并且基于寄存器的接口(采用本文描述的特征中的至少一些)可以将逻辑PHY连接到模拟或物理PHY,以及其他示例。

[0050] 控制器与PHY之间定义的接口可以尝试在控制器和PHY之间定义标准化接口,包括定义用于与对接口的管理和链路上的一个或多个互连协议的支持相关联地在实现控制器的计算模块与PHY之间传输的控制和状态信号。例如,PIPE使用用于涉及MAC和PHY之间的通信的每个操作的专用控制和状态信号线来定义MAC与PHY之间的接口。随着PIPE支持的每个协议规范(例如PCIe,SATA,USB)发展(并且随着附加协议被添加以通过PIPE支持),PIPE信号的数量随着时间的推移而增长。实际上,最近关于PIPE的工作已经尝试增加对USB Type-C的支持,其中包括对显示端口、Thunderbolt和可配置Rx/Tx通路对的支持,以及其他示例。然而,这种改进将导致使用用于每个操作的专用信号的当前方案显著增加PIPE接口的信号线(和引脚)数量。实际上,传统的PIPE接口正在达到这样一个程度,即不断增加的引脚数会威胁到其未来的可扩展性和可用性。

[0051] 本说明书呈现了解决上述问题的至少一些的特征和示例实现方式。例如,在一个示例中,通过利用基于寄存器的状态和控制接口来提供定义的接口(例如,PIPE接口)的低引脚数版本,可以解决定义的MAC-PHY的不断增加的信号和引脚数量。在一些示例接口中,可以定义一组数据路径信号以及控制和状态信号。例如,当前PIPE接口定义每个Rx/Tx通路对的数据路径信号和控制和状态信号(并且其他接口可以另外支持可配置对,其中对被配置为{Rx,Tx}、{Rx,Rx}、{Tx,Tx}或{Tx,Rx}等)。随着新信号被添加到接口定义中,这可能导

致提供额外的引脚以适应新信号。在一个实施例中,可以实现传统PIPE接口的低引脚数版本,例如,通过提供如下接口,其维持用于数据路径信号、异步控制和状态信号以及延迟敏感控制和状态信号的专用线路,但是将针对该接口定义的剩余控制和状态信号映射到寄存器(例如,8位、16位或32位寄存器),其可以通过少量额外的引脚/线路访问,所述额外的引脚/线路例如有助于每个方向的4位、8位等的数据传输的线路。为了使用寄存器支持这些控制和状态信号的消息传送,可以提供地址空间(例如,12个地址位),其中,定义的寄存器被映射到该地址空间中。在一些实现方式中,该地址空间可以被设计为故意大,以适应将使用这些定义的寄存器的操作集、控制和状态信号的扩展。这为将来的扩展提供了充足的空间,并提供了用于容纳特定于供应商的寄存器的空间,PHY设计可使用特定于供应商的寄存器向控制器提供有用的状态信息或提供额外的可配置性。

[0052] 继续上述示例,为了便于使用寄存器来进行这些控制和状态信号的消息传递,可以新定义读取、写入、完成和其他命令以访问寄存器。包括了一种用于将多个写入分组在一起以便它们在同一个循环中生效的机制。还包括了一种用于区分1周期断言类型信号和被保持为恒定值的信号的机制。涉及这些寄存器的事务可以包括命令、地址和数据或这三个元素的任何子集,其可以以时间复用的方式(例如,在多个单位间隔或时钟周期上)通过小的线路集传输。还可以结合接口定义成帧方案,通过该接口,相对应的计算块可以识别可能多个顺序(或同时)的寄存器事务的边界(例如,开始和结束),每个事务用于传送一个或多个控制或状态信号代替通过专用线驱动这些相同的信号,例如如在传统PIPE接口中以及其他示例特征中所做的那样。

[0053] 如上所述,改进的MAC-PHY接口的一些实现方式可以使得定义的接口能够被扩展以支持复杂和大的信号集(例如,当扩展对USB Type-C的PIPE规范支持时),同时减少接口的信号和引脚数。为了说明,当前的PIPE接口信号计数大致需要加倍以支持Type-C可配置对,这将使PIPE兼容的Type-C设计从物理设计的角度而言实现起来极具挑战性。改进的接口设计(以前使用专用线路将一些信号卸载到专用寄存器)也可以提供用于在未来启用更多接口操作的能力,因为接口支持的协议(例如,PIPE)发展为添加新功能,同时保护接口以免接口信号数量的进一步增加。

[0054] 转到图6,示出了将MAC计算块605与PHY计算块610耦合的传统PIPE接口的简化框图600。该接口可以包括控制和状态接口(用于控制和状态信令),每个通路有大约67个控制和状态输入以及80个控制和状态输出(即Rx/Tx对)。因此,实现多通路设计的接口将复制每个通路的这些信号中的每一个,这可能导致专用于支持该控制和状态接口的总引脚数的指数增长。

[0055] 转到图7,示出了利用基于寄存器的低引脚数PIPE控制和状态接口的改进的PIPE接口的简化框图700。该接口适于将MAC计算块705与PHY计算块710耦合。在一个示例中,针对接口定义的控制和状态信号的子集可以被分类为异步信号、定时关键信号以及常规控制和状态信号。可以在改进的接口上为异步和定时关键控制和状态信号分配专用线路,例如如图7中所示。然而,常规控制和状态信号可以映射到寄存器的位(例如,715、720)并由其替换,这些寄存器位通过一小组线(例如,四位或八位)访问,如本示例中所示。注册命令(例如读取和写入)、寄存器地址和寄存器数据可以在这个小型串行接口上以时间复用的方式发送。此外,接口的数据路径相关信号可以与控制和状态信号分离,并且实际上可以与传统接

口实现方式(例如,传统PIPE接口)中提供的信号相同或非常相似。

[0056] 转到图8,示出了表800,其示出了如何将一些PIPE控制和状态信号映射到8位PIPE寄存器的示例。虽然图8的示例示出了可以映射到计算块中的寄存器位的众多控制和状态信号的小子集,但是应当意识到的是,在实践中,可能是所定义的链路层-物理层接口的所有控制和状态信号(例如,PIPE)可以被映射到寄存器位(例如,异步和定时关键控制信号例外,其仍可以通过专用线实现)。此外,尽管图8的示例示出了8位寄存器,但是其他可能的寄存器宽度可以很容易地使用,包括16位或32位寄存器等。在一个示例实现方式中,MAC→PHY控制和状态信号可以映射到与PHY计算块相对应的寄存器的第一地址空间,而PHY→MAC控制和状态信号可以映射到与MAC计算块相对应的寄存器的第二地址空间。在一些情况下,第一和第二地址空间可以利用独立的地址空间,使得可以在PHY和MAC的寄存器中的每一个中使用相同的地址。在其他示例中,可以利用公共或共享的地址空间,使得第一和第二地址空间不重叠,PHY和MAC中的每个寄存器具有唯一的地址。在一个示例中,MAC→PHY控制和状态信号可以被映射到从地址0开始的地址空间,而PHY-MAC控制和状态信号可以被映射到从地址0开始的另一个地址空间。在图8的示例中,示出了12位的地址空间,其可以被认为足够大以容纳当前定义的PIPE信号,具有用于未来信号增长的充足余量,然而,在其他示例中可以选择其他地址空间大小。可以结合寄存器使用大地址空间,以便为供应商特定寄存器的专用地址范围留出空间,该供应商特定寄存器可用于暴露有用的PHY状态信息和/或提供附加的可配置性。在其他示例中,可以提供不同大小的地址空间,其可以经由不同的命令来访问,这取决于在串行接口上发送完整命令加地址位的延迟要求,以及其他示例实现方式。

[0057] 可以将示例PHY/MAC接口的一组状态/控制寄存器内的位映射到PHY/MAC接口中定义或扩展的一组信号中的定义信号(例如,PIPE规范中定义的信号)。在一个实现方式中,当将“1”写入映射到特定信号的位时,该值被解释为与在将专用线提供给每个信号的接口的实现方式中接收特定信号相同。作为示例,第一计算块可以确定应该进入TxDetect状态并且可以通过准备写入将其以消息形式发送到另一个计算块(通过被指定为PHY/MAC接口的状态和控制接口的接口的引脚的子集发送),其导致“1”被写入相对应的位(例如,12'h000[6])以指示信号“TxDetectRx/Loopback”(例如,如图8的示例中那样)。接收的、第二计算块可以检测到“1”已经被写入地址12'h000处的寄存器的位6,并且将该值解释为PIPE TxDetectRx/Loopback信号的接收,以及其他可能的示例。

编码	命令	描述
4'b0000	NOP	在空闲时段期间使用
4'b0001	write_committed	指示应提交当前写入以及任何先前未提交的写入，例如它们的值应在 PIPE 寄存器中更新。包含地址和数据。
[0058] 4'b0010	write_uncommitted	指示当前写入应保存为关闭，并且在将来收到 write_committed 时，其关联值将在 PIPE 寄存器中更新。包含地址和数据。
4'b0011	read	包含地址。
4'b0100	read completion	这是对读取的数据响应。仅包含数据。
其它	Reserved	预留

[0059] 表1:寄存器命令的示例

[0060] 表1提供了一些寄存器命令的示例,这些寄存器命令用于访问与针对MAC-PHY接口(例如PIPE)定义的控制和状态信号相关联地维护的寄存器。例如,可以利用无操作(或“NOP”)命令来指示不存在被请求的操作(例如,在空闲状态期间使用)。写操作可用于替换针对接口定义的一组控制和状态信号中的一个或多个的传输。例如,写入可以将值写入映射到该组控制和状态信号中的特定一个的特定寄存器的特定位。特定位的值可以被解释为特定信号的接收(即使实际上没有发送特定信号(例如,因为在改进的接口设计中已经省略了专用线))。

[0061] 在一些实例中,接口可以提供要在同一时间发送的一组控制和状态信号中的信号的组合。例如,某些PIPE信号可能需要对齐,以使它们的值在同一周期内生效。在接口的传统版本中,这种信号组合可以在它们各自的线上同时发送。在基于寄存器的改进实现方式中,同时写入与信号组合相对应的每个寄存器位可能是不可行的(例如,这些位可以分散在具有多个不同地址的多个寄存器中)。在一个示例中,写命令可以包括已提交写入和未提交写入。例如,未提交命令可用于临时地写入或将写入入队到与该命令相对应的识别的寄存器地址。在接收到下一个已提交写入之前,可以保持未提交写入,此时,与到已提交写入中所请求的寄存器中的写入一起,在插入的未提交写入中请求的值(例如,自上次提交的写入以来)被写入其各自的寄存器位。例如,可以将未提交写入写入到缓冲区(在已提交写入时刷新)或写入影子寄存器以存储写入,直到接收到下一个已提交写入并更新状态和控制寄存器,而已提交写入直接写入状态和控制寄存器。以这种方式,可以请求一个或多个未提交写入,随后是已提交写入,以同时将值写入多个不同的寄存器和位,以便实现映射到这些位的信号的对齐。

[0062] 作为示例,在具有8位寄存器的实现方式中,可以在三个或更多个寄存器(例如寄存器A、B和C)上映射24个不同的信号(来自定义的接口)。在一个示例中,映射到寄存器A中的三个相应位的三个信号可能需要与映射到寄存器B中的相应位的另一信号以及映射到寄存器C中的两个相应位的两个信号对齐。在该特定说明性示例中,模拟这些信号的对齐,值

可以在第一个write_uncommitted命令中写入寄存器A中的三个位,然后是第二个write_uncommitted命令将值写入寄存器B中的位。此后,write_committed命令不仅可以用于将值写入到寄存器C中的两个位,而且也“提交”并导致对寄存器A和B的未提交写入与写入寄存器C同时执行,从而导致与到寄存器A、B和C的写入相关联的所有值在同一周期中生效。

[0063] 可以与改进的接口的状态和控制寄存器相关联地提供附加操作。例如,可以提供读取和读取完成命令以访问写入到特定状态寄存器的值。还可以定义确认(ACK)命令,例如,以指示如下确认(即,向请求计算块):已经在特定寄存器处成功执行了已提交写入或未提交写入。

[0064] 一些实现可以省略对write_uncommitted命令的支持。例如,在一个实现方式中,特定计算块的寄存器可以以具有宽度和信号分配的这样的方式定义,被理解为需要对齐的信号被映射到相同寄存器或相邻寄存器中的位,从而使得可能在单个已提交写入中写入到寄存器中的每个相对应的位。其他可能有用的命令可以包括(但不限于)跨越多个相邻寄存器的写入以及其他示例。

[0065] 转到图9,示出了信号图900,示出了MAC-PHY接口的8位状态和控制接口930上的示例信令。在每个时钟(PCLK)935周期或单位间隔(UI)期间可以发送8位数据。在启动时或者在空闲状态之后,可以发送零,因为在MAC块和PHY块之间没有发送控制或状态信号。当空闲后发送非零数据时,数据可以被解释为接口上状态/控制事务的开始。例如,在图9的示例中,计算块中的第一个可以确定将定义的状态和控制信号中的特定一个发送到由接口定义的另一个计算块。在基于寄存器的实现方式中,省略了专用信令引脚,而代替地第一计算块则通过状态和控制接口发送数据。例如,事务可以用四位寄存器命令(例如,“4'd1”)开始(在910),然后在第一UI中发送可以应用于命令的寄存器地址的前四位。在下一个UI中,寄存器地址的剩余8位被发送(在915),随后是数据的四个UI(32位),包含要写入32位寄存器的值(从920开始)。

[0066] 在一些实现方式中,所有状态和控制寄存器事务可以包含命令。对于写和读命令,事务还可以包括相关联的寄存器地址。对于写入和读取完成,事务还可以包含数据(标识寄存器的内容)。因此,可以从命令类型推导出跨接口传输事务所需的周期数。例如,图9中所示的示例事务涉及在6个周期(或UI)中完成的在8位串行接口上传送的写命令905,假定4位命令、32位寄存器和12位地址空间。此配置中的其他事务预计将花费相应数量的UI来完成。例如,读取可以采用两个UI(例如,对于4位命令和12位地址),并且读取完成可以采用五个UI(例如,对于4位命令和32位读取数据),以及其他例子。给定这些各种事务的长度的可预测性,可以基于事务类型来检测事务的结束。因此,例如,当非零数据紧跟在检测到的表示先前事务的结束的UI或者位之后时,同样可以检测到另一个事务的开始。在一些实现方式中,这可以允许省略事务标识符。此外,当在空闲或空信号(例如,925)之后接收到有效命令时,同样可以检测到事务的开始,以及其他示例。

[0067] 在一些定义的接口(例如,PIPE)中,一些现有的状态和控制信号不仅基于它们在其上被传输的指定线路而且基于信号保持在相对应线路上的持续时间来定义。因此,在用寄存器映射(例如上面描述的)替换这些专用信令线中的至少一些的实现方式中,可能希望能够区分需要1个周期断言的信号和需要在多个UI(例如,静态值)上保持的信号。例如,可以配置特定的寄存器位或寄存器,使得写入该位的值保持在该值,但随后自动返回到默认

值或未断言值(即,无需显式写入事务将值返回到默认值(例如,从“1”返回到“0”)。例如,特定定位可以映射到具有1个周期断言的特定信号,使得当“1”被写入特定定位时,“1”被解释为特定信号的实例。但是,不是将特定定位的值保持为“1”,而是在相对应的单个UI或周期到期后,该值可以自动返回到“0”。同样地,要在多于一个UI保存某值的信号可以映射到配置为保持在该值直到定义的周期数到期或直到该位被覆盖为止的寄存器位,以及其他示例。在一些实例中,具有相似配置的位可以分组在同一个寄存器或连续寻址的寄存器中。例如,给定寄存器内的位都可以映射到相应的单周期信号断言,使得寄存器被处理以总是将值返回到寄存器中的任何位的默认值。其他寄存器可以用于将映射到具有类似相等的信号断言长度的其他信号的其他位分组,以及其他示例。在另一个实现方式中,可以简单地通过将两个不同的信号类型分组到位于不同地址空间或相同地址空间的不同地址范围中的不同寄存器并且基于其各自的地址来解释它们的值从而区分1周期断言类型信号和静态值类型信号。在又一实现方式中,可以将不同的信号类型映射到不同的寄存器并使用不同的写命令类型(例如,静态类型写入和单周期写入等)来访问,以及其他示例。

[0068] 图10是示出了与使用状态和控制寄存器(例如,代替专用信令线)实现的定义接口的控制和状态信令相对应的示例过程的流程图。例如,可以在物理层计算块和链路层计算块之间的接口的引脚子集上接收1005数据,该引脚子集被指定用于访问用于替换定义接口的信号的专用信令线的一组寄存器。数据可以包括命令、该组寄存器中特定一个寄存器的地址,以及要写入特定寄存器的值。特定寄存器可以利用该值进行修改1010或覆盖值使得特定寄存器的一个或多个位可以通过写入改变。寄存器的每个位可以映射到定义的接口的多个不同状态和控制信号中的一个(在定义的接口的其他实施例中提供专用信令线),以及将特定值(例如,“1”)写入到任何一个位都可以代替专用线上相对应信号的接收。因此,当接收到所定义接口的相对应信号时,可以检测和解释1015一个或多个修改的寄存器位值。然后,响应于接收到在寄存器中识别的特定状态或控制信号,计算块(例如,托管寄存器和“接收”特定信号)可以执行1020相对应的动作。例如,取决于特定状态和控制信号的性质,接收计算块可以执行1020动作,诸如改变操作频率,改变功率状态,改变发射器系数设置,以及其它示例。还可以通过基于寄存器的减少引脚的接口与状态和控制事务相关联地执行附加任务。例如,计算块可以检测特定寄存器位或相关联的信号是1周期或静态值信号类型,并确定是否自动将写入位返回到默认值或使用特定类型的写命令以便于准确表示寄存器中的信号。计算块还可以通过基于在第一事务中使用的命令识别第一事务的预期长度来识别第一事务与第二事务之间的边界,以及其他示例任务。

[0069] 注意,上述装置、方法和系统可以在如上所述的任何电子设备或系统中实现。作为具体说明,下图提供了用于利用如本文所述的本发明的示例性系统。由于以下更详细地描述了该系统,因此从上面的讨论中公开、描述和重新考虑了许多不同的互连。并且显而易见的是,上述进步可以应用于那些互连、结构、或架构中的任何一种。

[0070] 参考图11,描绘了包括多核处理器的计算系统的框图的实施例。处理器1100包括任何处理器或处理设备,例如微处理器、嵌入式处理器、数字信号处理器(DSP)、网络处理器、手持处理器、应用处理器、协处理器、片上系统(SOC)或执行代码的其他设备。在一个实施例中,处理器1100包括至少两个核-核心1101和1102,其可包括不对称核或对称核(所示实施例)。然而,处理器1100可以包括可以是对称或不对称的任何数量的处理元件。

[0071] 在一个实施例中,处理元件是指用于支持软件线程的硬件或逻辑。硬件处理元件的示例包括:线程单元、线程槽、线程、进程单元、上下文、上下文单元、逻辑处理器、硬件线程、核心,和/或能够保持处理器的状态(诸如执行状态或架构状态)的任何其它元件。换言之,在一个实施例中,处理元件是指能够独立地与代码(诸如软件线程、操作系统、应用程序或其它代码)相关联的任何硬件。物理处理器(或处理器插槽)通常是指集成电路,可能包括诸如核心或硬件线程的任意数量的其它处理元件。

[0072] 核心通常是指位于能够维护独立架构状态的集成电路上的逻辑,其中每个独立维护的架构状态与至少一些专用执行资源相关联。与核不同,硬件线程通常是指位于能够维护独立架构状态的集成电路上的任何逻辑,其中独立维护的架构状态共享对执行资源的访问。能够看出,当某些资源被共享而其它专用于架构状态时,硬件线程和核心的命名法之间的线重叠。更通常地,核心和硬件线程被操作系统视为单独的逻辑处理器,其中操作系统能够单独地调度每个逻辑处理器上的操作。

[0073] 如图11所示,物理处理器1100包括两个核心-核心1101和1102。这里,核心1101和1102被认为是对称核心,即具有相同配置、功能单元和/或逻辑的核心。在另一实施例中,核心1101包括乱序处理器核,而核心1102包括有序处理器核。然而,核心1101和1102可以从任何类型的核中单独选择,所述任何类型的核例如本机核、软件管理核、适于执行本机指令集架构(ISA)的核、适于执行转换指令集架构(ISA)的核、共同设计的核心或其他已知核心。在异构核心环境(即,非对称核心)中,可以利用某种形式的转换(例如二进制转换)来在一个或两个核上调度或执行代码。然而,为了进一步讨论,核心1101中示出的功能单元在下面进一步详细描述,因为核心1102中的单元在所描绘的实施例中以类似的方式操作。

[0074] 如图所示,核心1101包括两个硬件线程1101a和1101b,它们也可以称为硬件线程槽1101a和1101b。因此,在一个实施例中,诸如操作系统的软件实体可能将处理器1100视为四个单独的处理器,即,能够同时执行四个软件线程的四个逻辑处理器或处理元件。如上所述,第一线程与架构状态寄存器1101a相关联,第二线程与架构状态寄存器1101b相关联,第三线程可以与架构状态寄存器1102a相关联,并且第四线程可以与架构状态寄存器1102b相关联。这里,架构状态寄存器(1101a、1101b、1102a和1102b)中的每个可以被称为处理元件、线程槽或线程单元,如上所述。如图所示,架构状态寄存器1101a在架构状态寄存器1101b中被复制,因此能够为逻辑处理器1101a和逻辑处理器1101b存储单个架构状态/上下文。在核心1101中,还可以为线程1101a和1101b复制其他较小的资源,例如分配器和重命名器块1130中的指令指针和重命名逻辑。可以通过分区来共享一些资源,诸如重新排序/引退单元1135中的重新排序缓冲区、ILTB 1120、加载/存储缓冲区和队列。其他资源(例如,通用内部寄存器、页表基址寄存器、低级数据高速缓存和数据TLB 1115、执行单元1140以及乱序单元1135的部分)是可能完全共享的。

[0075] 处理器1100通常包括其他资源,其可以是完全共享的、通过分区共享的,或由处理元件/对处理元件专用的。在图11中,示出了具有处理器的说明性逻辑单元/资源的纯示例性处理器的实施例。注意,处理器可以包括或省略这些功能单元中的任何一个,以及包括未描绘的任何其他已知功能单元、逻辑或固件。如图所示,核心1101包括简化的、代表性的乱序(000)处理器核心。但是可以在不同的实施例中使用有序处理器。000核心包括用于预测要执行/采用的分支的分支目标缓冲器1120和用于存储用于指令的地址转换条目的指令转

换缓冲器 (I-TLB) 1120。

[0076] 核心1101还包括解码模块1125,其耦合到取指单元1120以解码所获取的元素。在一个实施例中,取指逻辑包括分别与线程槽1101a、1101b相关联的各个定序器。通常,核心1101与第一ISA相关联,第一ISA定义/指定在处理器1100上能执行的指令。通常作为第一ISA的一部分的机器代码指令包括指令的一部分(称为操作码),其引用/指定要执行的指令或操作。解码逻辑1125包括从其操作码识别这些指令并在管线中传递解码的指令以便进行由第一ISA定义的处理的电路。例如,如下面更详细讨论的,在一个实施例中,解码器1125包括被设计或适于识别特定指令的逻辑,例如事务指令。作为解码器1125识别的结果,架构或核心1101采取特定的预定义动作来执行与适当指令相关联的任务。重要的是要注意,可以响应于单个或多个指令来执行本文描述的任何任务、块、操作和方法;其中一些可能是新的或旧的指令。注意,在一个实施例中,解码器1126识别相同的ISA(或其子集)。可替代地,在异构核心环境中,解码器1126识别第二ISA(第一ISA的子集或不同的ISA)。

[0077] 在一个示例中,分配器和重命名器块1130包括用于预留资源的分配器,诸如用于存储指令处理结果的寄存器文件。然而,线程1101a和1101b可能能够乱序执行,其中分配器和重命名器块1130还保留其他资源,例如用于跟踪指令结果的重新排序缓冲器。单元1130还可以包括寄存器重命名器,以将程序/指令参考寄存器重命名为处理器1100内部的其他寄存器。重新排序/引退单元1135包括组件,例如上面提到的重排序缓冲器、加载缓冲器和存储缓冲器,以支持乱序执行和稍后乱序执行的指令的有序引退。

[0078] 在一个实施例中,调度器和执行单元块1140包括调度器单元,用于调度执行单元上的指令/操作。例如,浮点指令被调度在具有可用浮点执行单元的执行单元的端口上。还包括与执行单元相关联的寄存器文件以存储信息指令处理结果。示例性执行单元包括浮点执行单元、整数执行单元、跳转执行单元、加载执行单元、存储执行单元和其他已知的执行单元。

[0079] 较低级数据高速缓存和数据转换缓冲器(D-TLB) 1150耦合到执行单元1140。数据高速缓存用于存储最近使用/操作的元素,例如数据操作数,其可能保存在存储器一致性状态中。D-TLB用于存储最近的虚拟/线性到物理地址转换。作为特定示例,处理器可以包括页表结构以将物理存储器分成多个虚拟页面。

[0080] 这里,核心1101和1102共享对更高级或更远级的高速缓存(例如,与片上接口1110相关联的第二级高速缓存)的访问。注意,更高级或更远级是指高速缓存级增加或从执行单位进一步远离。在一个实施例中,较高级高速缓存是最后级数据高速缓存——处理器1100上的存储器层次结构中的最后高速缓存,诸如第二或第三级数据高速缓存。然而,较高级高速缓存不限于此,因为它可以与指令高速缓存相关联或包括指令高速缓存。追踪高速缓存是一种类型的指令高速缓存,可以耦合在解码器1125之后以存储最近解码的踪迹。这里,指令可能指宏指令(即,由解码器识别的一般指令),其可以解码为多个微指令(微操作)。

[0081] 在所描绘的配置中,处理器1100还包括片上接口模块1110。历史上,下面更详细描述存储器控制器已被包括在处理器1100外部的计算系统中。在这种情景下,片上接口1110用于与处理器1100外部的设备进行通信,所述处理器1100外部的设备例如系统存储器1175、芯片组(通常包括连接到存储器1175的存储器控制器集线器和用于连接外围设备的I/O控制器集线器)、存储器控制器集线器、北桥或其他集成电路。并且在这种情况下,总线

1105可以包括任何已知的互连,例如多点总线、点对点互连、串行互连、并行总线,相干(例如高速缓存相干)总线、分层协议架构、差分总线和GTL总线。

[0082] 存储器1175可以专用于处理器1100或者与系统中的其他设备共享。存储器1175类型的常见示例包括DRAM、SRAM、非易失性存储器(NV存储器)和其他已知的存储设备。注意,设备1180可以包括图形加速器、耦合到存储器控制器集线器的处理器或卡、耦合到I/O控制器集线器的数据存储装置、无线收发器、闪存设备、音频控制器、网络控制器或其他已知的设备。

[0083] 然而,最近,随着更多逻辑和设备被集成在诸如SOC的单个管芯上,这些设备中的每一个可以合并处理器1100上。例如,在一个实施例中,存储器控制器集线器与处理器1100在同一封装和/或管芯上。这里,核心的一部分(核心上部分)1110包括一个或多个控制器,用于与诸如存储器1175或图形设备1180之类的其他设备对接。包括互连和用于与这些设备对接的控制器的该配置通常被称为核心上(或非核心配置)。作为示例,片上接口1110包括用于片上通信的环形互连和用于片外通信的高速串行点对点链路1105。然而,在SOC环境中,甚至更多的设备,例如网络接口、协处理器、存储器1175、图形处理器1180和任何其他已知的计算机设备/接口可以集成在单个管芯或集成电路上以提供具有高功能和低功耗的小形状因子。

[0084] 在一个实施例中,处理器1100能够执行编译器、优化和/或转换器代码1177以编译、转换和/或优化应用代码1176以支持本文描述的装置和方法或与其对接。编译器通常包括用于将源文本/代码转换成目标文本/代码的程序或程序集。通常,使用编译器编译程序/应用代码是在多个阶段和多遍中完成的,以将高级编程语言代码转换为低级机器或汇编语言代码。然而,单遍编译器仍可用于简单编译。编译器可以利用任何已知的编译技术并执行任何已知的编译器操作,例如词法分析、预处理、解析、语义分析、代码生成、代码转换和代码优化。

[0085] 较大的编译器通常包括多个阶段,但是大多数情况下这些阶段包括在两个一般阶段中:(1)前端,即通常可以进行句法处理、语义处理和一些变换/优化的地方,以及(2)后端,即通常进行分析、转换、优化和代码生成的地方。一些编译器引用了一个中间部分,它示出了编译器的前端和后端之间描绘的模糊。结果,对编译器的插入、关联、生成或其他操作的引用可以在任何上述阶段或遍以及编译器的任何其他已知阶段或遍中发生。作为说明性示例,编译器可能在编译的一个或多个阶段插入操作、调用、函数等,例如在编译的前端阶段插入调用/操作,并且然后在变换阶段将调用/操作变换为更低级别的代码。请注意,在动态编译期间,编译器代码或动态优化代码可以插入此类操作/调用,并且对代码进行优化以便在运行时执行。作为特定说明性示例,可以在运行时期间动态地优化二进制代码(已编译的代码)。这里,程序代码可以包括动态优化代码、二进制代码或其组合。

[0086] 与编译器类似,转换器(例如二进制转换器)静态地或动态地转换代码以优化和/或转换代码。因此,对代码、应用程序代码、程序代码或其他软件环境的执行的引用可以指代:(1)动态地或静态地执行编译器程序、优化代码优化器或转换器以编译程序代码,维护软件结构,执行其他操作,优化代码,或转换代码;(2)执行包括操作/调用的主程序代码,例如已优化/编译的应用代码;(3)执行与主程序代码相关联的其他程序代码,例如库,以维护软件结构,执行其他软件相关操作,或优化代码;或(4)其组合。

[0087] 现在参考图12,示出的是根据本发明的实施例的第二系统1200的框图。如图12所示,多处理器系统1200是点对点互连系统,并且包括经由点对点互连1250耦合的第一处理器1270和第二处理器1280。处理器1270和1280中的每一个可以是处理器的某个版本。在一个实施例中,1252和1254是串行点对点相干互连结构的一部分,例如Intel的快速路径互连(QPI)架构。结果,本发明可以在QPI架构内实现。

[0088] 虽然仅用两个处理器1270、1280示出,但应理解,本发明的范围不限于此。在其他实施例中,一个或多个附加处理器可以存在于给定处理器中。

[0089] 示出的处理器1270和1280分别包括集成存储器控制器单元1272和1282。处理器1270还包括作为其总线控制器单元的一部分的点对点(P-P)接口1276和1278;类似地,第二处理器1280包括P-P接口1286和1288。处理器1270、1280可以使用P-P接口电路1278、1288经由点对点(P-P)接口1250交换信息。如图12所示,IMC 1272和1282将处理器耦合到相应的存储器,即存储器1232和存储器1234,其可以是本地附接到相应处理器的主存储器的一部分。

[0090] 处理器1270、1280各自使用点对点接口电路1276、1294、1286、1298经由各个PP接口1252、1254与芯片组1290交换信息。芯片组1290还经由接口电路1292沿着高性能图形互连1239与高性能图形电路1238交换信息。

[0091] 共享高速缓存(未示出)可以包括在处理器中或两个处理器外部;然后经由P-P互连与处理器连接,使得如果处理器被置于低功率模式,则处理器的本地高速缓存信息中的任一个或两者可以存储在共享高速缓存中。

[0092] 芯片组1290可以经由接口1296耦合到第一总线1216。在一个实施例中,第一总线1216可以是外围组件互连(PCI)总线,或者诸如快速PCI总线或另一第三代I/O互连总线的总线,但本发明的范围不限于此。

[0093] 如图12所示,各种I/O设备1214沿着将第一总线1216耦合到第二总线1220的总线桥1218耦合到第一总线1216。在一个实施例中,第二总线1220包括低引脚数(LPC)总线。在一个实施例中,各种设备耦合到第二总线1220,所述各种设备包括例如键盘和/或鼠标1222、通信设备1227和存储单元1228,诸如磁盘驱动器或其他大容量存储设备,其通常包括指令/代码和数据1230。此外,音频I/O 1224被示出耦合到第二总线1220。注意,其他架构是可能的,其中所包括的组件和互连架构是变化的。例如,代替图12的点对点架构,系统可以实现多分支总线或其他这样的架构。

[0094] 计算系统可包括组件的各种组合。这些组件可以实现为IC、其部分、分立电子器件或者在计算机系统中适用的其他模块、逻辑、硬件、软件、固件或其组合,或者实现为以其他方式结合在计算机系统的机箱内的组件。然而,应该理解,可以省略所示的一些组件,可以存在另外的组件,并且在其他实现方式中可以出现所示组件的不同布置。结果,上面描述的本发明可以在下面示出或描述的互连的一个或多个的任何部分中实现。

[0095] 在一个实施例中,处理器包括微处理器、多核处理器、多线程处理器、超低电压处理器、嵌入式处理器或其他已知处理元件。在所示的实现方式中,处理器充当主处理单元和中央集线器,用于与系统的许多不同组件通信。作为一个示例,处理器被实现为片上系统(SoC)。作为具体的说明性示例,处理器包括基于Intel®Architecture Core™的处理器,诸如i3、i5、i7或可从加利福尼亚州圣克拉拉市的Intel公司获得的另一种这样的处理器。然而,要了解的是,可替代地存在其他低功率处理器,例如可从加利福尼亚州

Sunnyvale的Advanced Micro Devices公司(AMD)获得的,来自加利福尼亚州Sunnyvale的MIPS科技公司的基于MIPS的设计,ARM控股公司或其客户或其许可证持有者或采用者许可的基于ARM的设计,例如Apple A5/A6处理器、Qualcomm Snapdragon处理器或TI OMAP处理器。请注意,此类处理器的许多客户版本都经过修改和更改;但是,它们可以支持或识别执行由处理器许可方规定的定义算法的特定指令集。这里,微架构实现方式可能会有所不同,但处理器的架构功能通常是一致的。下面将进一步讨论关于一个实现方式中的处理器的架构和操作的某些细节以提供说明性示例。

[0096] 在一个实施例中,处理器与系统存储器进行通信。作为说明性示例,在一个实施例中,其可以经由多个存储器设备来实现以提供给定量的系统存储器。例如,存储器可以根据基于联合电子器件工程委员会(JEDEC)的低功率双倍数据速率(LPDDR)的设计,例如根据JEDEC JESD 209-2E(2009年4月发布)的当前LPDDR2标准,或者下一代LPDDR标准,称为LPDDR3或LPDDR4,它将为LPDDR2提供扩展以增加带宽。在各种实现方式中,各个存储器设备可以是不同的封装类型,诸如单管芯封装(SDP)、双管芯封装(DDP)或四管芯封装(QDP)。在一些实施例中,这些设备直接焊接到母板上以提供较低轮廓的解决方案,而在其他实施例中,这些设备被配置为一个或多个存储器模块,其继而通过给定连接器耦合到母板。当然,其他存储器实现方式也是可能的,例如其他类型的存储器模块,例如,不同种类的双列直插式存储器模块(DIMM),包括但不限于microDIMM、MiniDIMM。在特定说明性实施例中,存储器的大小在2GB到16GB之间,并且可被配置为经由球栅阵列(BGA)焊接到母板上的DDR3LM封装或LPDDR2或LPDDR3存储器。

[0097] 为了提供诸如数据、应用、一个或多个操作系统等信息的持久存储,大容量存储装置也可以耦合到处理器。在各种实施例中,为了实现更薄更轻的系统设计以及提高系统响应性,可以经由SSD实现该大容量存储装置。然而,在其他实施例中,大容量存储装置可以主要使用具有少量SSD存储的硬盘驱动器(HDD)来实现,以充当SSD高速缓存以在断电事件期间实现上下文状态和其他此类信息的非易失性存储,使得可以在重新启动系统活动时快速加电。闪存设备可以例如经由串行外围接口(SPI)耦合到处理器。该闪存设备可以提供系统软件的非易失性存储,所述系统软件包括基本输入/输出软件(BIOS)以及系统的其他固件。

[0098] 在各种实施例中,系统的大容量存储由SSD单独实现或者作为具有SSD高速缓存的磁盘、光盘或其他驱动器实现。在一些实施例中,大容量存储装置被实现为SSD或HDD以及恢复(ST)高速缓存模块。在各种实现方式中,HDD提供320GB-4TB(TB)之间和向上的存储,而RST高速缓存用容量为24GB-256GB的SSD实现。注意,这种SSD高速缓存可以被配置为单级高速缓存(SLC)或多级高速缓存(MLC)选项,以提供适当级别的响应。在仅SSD的选项中,模块可以容纳在各种位置,例如mSATA或NGFF插槽。例如,SSD的容量范围为120GB-1TB。

[0099] 各种输入/输出(I/O)设备可以存在于系统内,例如显示器,其可以是配置在机箱的盖子部分内的高清LCD或LED面板。该显示面板还可以提供触摸屏,例如,在显示面板外部适配,使得经由用户与该触摸屏的交互,可以向系统提供用户输入以实现期望的操作,例如关于信息的显示、信息的访问等。在一个实施例中,显示器可以经由显示互连耦合到处理器,该显示互连可以实现为高性能图形互连。触摸屏可以经由另一互连耦合到处理器,在一个实施例中,另一互连可以是I²C互连。除了触摸屏之外,通过触摸方式的用户输入也可以经

由触摸板发生,触摸板可以配置在机箱内并且还可以耦合到与触摸屏相同的I²C互连。

[0100] 显示面板可以以多种模式操作。在第一模式中,显示面板可以布置在透明状态,其中显示面板对可见光透明。在各种实施例中,除了围绕外围的边框之外,显示面板的大部分可以是显示器。当系统以笔记本模式操作并且显示面板以透明状态操作时,用户可以查看在显示面板上呈现的信息,同时还能够查看显示器后面的对象。另外,显示在显示面板上的信息可以由位于显示器后面的用户观看。或者,显示面板的操作状态可以是不透明状态,其中可见光不通过显示面板发射。

[0101] 在平板模式中,系统被折叠闭合,使得当基板的底表面搁置在表面上或由用户持有时,显示面板的显示器后显示表面停留在使其向外面向用户的位置。在平板电脑操作模式中,显示器后表面起到显示器和用户界面的作用,因为该表面可以具有触摸屏功能并且可以执行传统触摸屏设备(例如,平板设备)的其他已知功能。为此,显示面板可以包括设置在触摸屏层与前显示表面之间的透明度调节层。在一些实施例中,透明度调节层可以是电致变色层(EC)、LCD层或EC和LCD层的组合。

[0102] 在各种实施例中,显示器可以具有不同的尺寸,例如11.6”或13.3”的屏幕,并且可以具有16:9的纵横比,并且具有至少300尼特的亮度。此外,显示器可以是全高清(HD)分辨率(至少1920x 1080p),与嵌入式显示器端口(eDP)兼容,并且是具有面板自刷新功能的低功率面板。

[0103] 关于触摸屏功能,系统可以提供多触摸电容并且至少具有5个手指能力的显示器多触摸面板。并且在一些实施例中,显示器可以是10个手指能力的。在一个实施例中,触摸屏容纳在损坏和防刮擦玻璃和涂层(例如,Gorilla Glass™或Gorilla Glass 2™)内,以获得低摩擦力以减少“手指烧伤”并避免“手指跳过”。为了提供增强的触摸体验和响应性,在一些实现方式中,触摸面板具有多点触摸功能,例如在压缩变焦期间每静态视图小于2帧(30Hz),以及每帧(30Hz) 200ms(手指滞后于指针)小于1cm的单点触摸功能。在一些实现方式中,显示器支持边缘到边缘的玻璃,其具有也与面板表面齐平的最小的屏幕边框,并且在使用多点触摸时受到IO干扰限制。

[0104] 对于感知计算和其他目的,各种传感器可以存在于系统内并且可以以不同的方式耦合到处理器。某些惯性和环境传感器可以通过传感器集线器耦合到处理器,例如,经由I²C互连。这些传感器可以包括加速度计、环境光传感器(ALS)、罗盘和陀螺仪。其他环境传感器可以包括一个或多个热传感器,在一些实施例中,这些热传感器经由系统管理总线(SMBus)总线耦合到处理器。

[0105] 使用平台中存在的各种惯性和环境传感器,可以实现许多不同的用例。这些用例支持高级计算操作,包括感知计算,并且还允许在电源管理/电池寿命、安全性和系统响应性方面进行增强。

[0106] 例如,关于电源管理/电池寿命问题,至少部分地基于来自环境光传感器的信息,确定平台位置中的环境光条件并相应地控制显示器的强度。因此,在某些光条件下减少了操作显示器所消耗的功率。

[0107] 关于安全操作,基于诸如位置信息的从传感器获得的上下文信息,可以确定是否允许用户访问某些安全文档。例如,可以准许用户在工作场所或家庭位置访问这些文档。然而,当平台存在于公共场所时,阻止用户访问这些文档。在一个实施例中,该确定基于位置

信息,例如,经由GPS传感器或地标的相机识别来确定位置信息。其他安全操作可以包括提供彼此近距离内的设备的配对,例如,如本文所述的便携式平台和用户的台式计算机、移动电话等。在一些实现方式中,当这些设备如此配对时,经由近场通信实现某些共享。但是,当设备超过一定范围时,可以禁用这种共享。此外,当将如本文所述的平台与智能电话配对时,警报可被配置为当设备在公共场所移动超过彼此预定距离时被触发。相反,当这些配对设备处于安全位置时,例如工作场所或家庭位置时,设备可能超过该预定限制而不会触发这种警报。

[0108] 还可以使用传感器信息来增强响应性。例如,即使当平台处于低功率状态时,仍然可以使传感器能够以相对低的频率运行。因此,确定平台位置的任何变化,例如由惯性传感器、GPS传感器等确定的。如果没有注册这样的改变,则发生到先前无线集线器(例如Wi-Fi™接入点或类似的无线使能器)的更快连接,因为在这种情况下不需要扫描可用的无线网络资源。因此,实现了从低功率状态唤醒时的更高水平的响应性。

[0109] 应当理解,可以使用经由如本文所述的平台内的集成传感器获得的传感器信息来启用许多其他用例,并且以上示例仅用于说明的目的。使用如本文所述的系统,感知计算系统可以允许添加替代输入模态,包括手势识别,并使系统能够感测用户操作和意图。

[0110] 在一些实施例中,可以存在一个或多个红外或其他热感测元件,或用于感测用户的存在或移动的任何其他元件。这种感测元件可包括一起工作、顺序工作或两者的多个不同元件。例如,感测元件包括提供初始感测的元件,例如光或声音投影,然后通过例如超声飞行时间相机或图案化光相机感测手势检测。

[0111] 同样在一些实施例中,该系统包括光发生器以产生照明线。在一些实施例中,该线提供关于虚拟边界的视觉提示,即空间中的虚构或虚拟位置,其中用户通过或突破虚拟边界或平面的动作被解释为使用计算机系统的意图。在一些实施例中,当计算系统相对于用户转变到不同状态时,照明线可以改变颜色。照明线可以用于为空间中的虚拟边界的用户提供视觉提示,并且系统可以使用该照明线来确定计算机关于用户的状态的转变,包括确定用户何时希望使用计算机。

[0112] 在一些实施例中,计算机感测用户位置并操作以将用户的手移动通过虚拟边界解释为指示用户使用计算机的意图的手势。在一些实施例中,在用户通过虚拟线或平面时,由光发生器产生的光可以改变,从而向用户提供用户已经进入用于提供手势以向计算机提供输入的区域视觉反馈。

[0113] 显示屏可以提供关于用户的计算系统的状态转换的视觉指示。在一些实施例中,以第一状态提供第一屏幕,其中系统感测用户的存在,例如通过使用感测元件中的一个或多个。

[0114] 在一些实现方式中,系统用于感测用户身份,例如通过面部识别。这里,可以在第二状态中提供到第二屏幕的转换,其中计算系统已经识别出用户身份,其中第二屏幕向用户提供用户已经转换到新状态的视觉反馈。转换到第三屏幕可以在用户已经确认对用户的识别的第三状态中发生。

[0115] 在一些实施例中,计算系统可以使用转换机制来确定用户的虚拟边界的位置,其中虚拟边界的位置可以随用户和上下文而变化。计算系统可以生成光,例如照明线,以指示用于与系统接合的虚拟边界。在一些实施例中,计算系统可以处于等待状态,并且可以以第

一颜色产生光。计算系统可以检测用户是否已经到达经过虚拟边界,例如通过使用感测元件感测用户的存在和移动。

[0116] 在一些实施例中,如果已经检测到用户已经越过虚拟边界(诸如用户的手比虚拟边界线更靠近计算系统),则计算系统可以转换到用于从用户接收手势输入的状态,其中用于指示转变的机制可以包括指示虚拟边界改变为第二颜色的光。

[0117] 在一些实施例中,计算系统然后可以确定是否检测到手势移动。如果检测到手势移动,则计算系统可以继续进行手势识别过程,该手势识别过程可以包括使用来自手势数据库的数据,该手势数据库可以驻留在计算设备中的存储器中或者可以由计算设备以其他方式访问。

[0118] 如果识别出用户的手势,则计算系统可以响应于输入而执行功能,并且如果用户在虚拟边界内,则返回以接收附加手势。在一些实施例中,如果未识别手势,则计算系统可以转换到错误状态,其中用于指示错误状态的机制可以包括指示虚拟边界改变为第三颜色的灯,如果用户在虚拟边界内以便与计算系统接合,则系统返回以接收另外的手势。

[0119] 如上所述,在其他实施例中,系统可以被配置为可转换平板电脑系统,其可以用于至少两种不同模式,平板电脑模式和笔记本电脑模式。可转换系统可以具有两个面板,即显示面板和基板,使得在平板模式下,两个面板堆叠地设置在彼此之上。在平板模式中,显示面板面向外并且可以提供传统平板电脑中常见的触摸屏功能。在笔记本模式中,两个面板可以以打开的翻盖配置来布置。

[0120] 在各种实施例中,加速计可以是具有至少50Hz的数据速率的3轴加速计。还可以包括陀螺仪,其可以是3轴陀螺仪。此外,可以存在电子罗盘/磁力计。而且,可以提供一个或多个接近传感器(例如,用于打开盖子以感测人何时接近(或不接近)系统并调节功率/性能以延长电池寿命)。对于包括加速度计、陀螺仪和罗盘的某些OS的传感器融合功能,可以提供增强特征。另外,经由具有实时时钟(TC)的传感器集线器,可以实现来自传感器机构的唤醒,以在系统的其余部分处于低功率状态时接收传感器输入。

[0121] 在一些实施例中,内盖/显示器打开开关或传感器用于指示盖子何时关闭/打开,并且可用于将系统置于连接待机或自动从连接待机状态唤醒。其他系统传感器可以包括ACPI传感器,以用于内部处理器、存储器和皮肤温度监测,以基于所感测的参数实现处理器和系统操作状态的改变。

[0122] 在实施例中,OS可以是实现连接待机的微软®Windows® OS(在此也称为Win8 CS)。Windows 8连接待机或具有类似状态的另一OS可以经由如本文所述的平台提供非常低的超空闲功率,以使应用能够以非常低的功耗保持连接(例如,连接到基于云的位置)。该平台可支持3种电源状态,即屏幕开启(正常);已连接待机(作为默认“关闭”状态);以及关机(零瓦特的功耗)。因此,在连接待机状态下,即使屏幕关闭,平台也在逻辑上开启(以最小功率水平)。在这样的平台中,可以使电源管理对应用透明并保持恒定连接,部分原因是卸载技术使最低功率组件能够执行操作。

[0123] 各种外围设备可以经由低引脚数(LPC)互连耦合到处理器。在所示的实施例中,各种组件可以通过嵌入式控制器耦合。这些组件可包括键盘(例如,经由PS2接口耦合)、风扇和热传感器。在一些实施例中,触摸板还可以经由PS2接口耦合到EC。另外,根据2003年10月2日的可信计算组(TCG)TPM规范版本1.2的安全处理器(诸如可信平台模块(TPM))也可以经

由该LPC互连耦合到处理器。然而,应理解,本发明的范围不限于此方面,并且安全信息的安全处理和存储可以在另一个受保护的位置,例如安全协处理器中的静态随机存取存储器(SRAM),或者作为仅在受安全包围区(SE)处理器模式保护时才被解密的加密数据块。

[0124] 在特定实现方式中,外围端口可以包括高清媒体接口(HDMI)连接器(其可以具有不同的形状因数,例如全尺寸、迷你或微型);一个或多个USB端口,例如符合通用串行总线修订版3.0规范(2008年11月)的全尺寸外部端口,当系统处于连接待机状态并插入交流墙电源时,至少有一个被供电以用于USB设备(如智能手机)的充电。此外,还可以提供一个或多个Thunderbolt™端口。其他端口可以包括可从外部访问的读卡器,例如全尺寸SD-XC读卡器和/或SIM读卡器(例如,8引脚读卡器)以用于WWAN。对于音频,可以存在具有立体声和麦克风功能(例如,组合功能)的3.5mm插孔,支持插孔检测(例如,仅支持使用盖子中的麦克风的耳机或使用电缆中的麦克风的耳机)。在一些实施例中,该插孔可以在立体声耳机和立体声麦克风输入之间重新安排任务。而且,可以提供电源插孔以耦合到AC砖。

[0125] 系统可以以各种方式与外部设备通信,包括无线方式。在一些实例中,存在各种无线模块,每个无线模块可以与被配置用于特定无线通信协议的无线电相对应。用于诸如近场的短距离无线通信的一种方式可以经由近场通信(NFC)单元,其可以在一个实施例中经由SMBus与处理器进行通信。请注意,经由此NFC单元,彼此靠近的设备可以进行通信。例如,用户可以经由将两个设备紧密地配合在一起并且能够传送诸如标识信息、支付信息的信息,诸如图像数据之类的数据等来使系统能够与诸如用户的智能电话之类的另一个(例如)便携式设备进行通信。还可以使用NFC系统来执行无线功率传输。

[0126] 使用这里描述的NFC单元,用户可以通过利用在这种设备的一个或多个的线圈之间的耦合来并排地对设备进行并排放置并将设备并排放置用于近场耦合功能(例如近场通信和无线功率传输(WPT))。更具体地,实施例提供具有策略性成形和放置的铁氧体材料的器件,以提供线圈的更好耦合。每个线圈都有与其相关联的电感,可以与系统的电阻、电容和其他特性一起选择,以便为系统提供共同的谐振频率。

[0127] 此外,附加无线单元可以包括其他短程无线引擎,包括WLAN单元和蓝牙单元。使用WLAN单元,可以实现根据给定的电气和电子工程师协会(IEEE)802.11标准的Wi-Fi™通信,而经由蓝牙单元,可以发生经由蓝牙协议的短距离通信。这些单元可以经由例如USB链路或通用异步接收器发射器(UART)链路与处理器进行通信。或者这些单元可以经由根据快速外围组件互连™(PCIe™)协议的互连耦合到处理器,例如,根据快速PCI™规范基础规范版本3.0(2007年1月17日公布)或另一个这样的协议,例如串行数据输入/输出(SDIO)标准。当然,可以在一个或多个附加卡上配置的这些外围设备之间的实际物理连接可以通过适用于主板的NGFF连接器的方式。

[0128] 另外,例如根据蜂窝或其他无线广域协议的无线广域通信可以经由WWAN单元发生,该WWAN单元又可以耦合到订户身份模块(SIM)。另外,为了能够接收和使用位置信息,还可以存在GPS模块。WWAN单元和诸如相机模块的集成捕获设备可以经由给定的USB协议(例如USB 2.0或3.0链路,或UART或I²C协议)进行通信。同样,这些单元的实际物理连接可以经由将NGFF附加卡适配到主板上配置的NGFF连接器。

[0129] 在特定实施例中,可以模块化地提供无线功能,例如,利用WiFi™802.11ac解决方案(例如,与IEEE 802.11ab向后兼容的附加卡)以及对Windows 8CS的支持。该卡可以配置

在内部插槽中(例如,经由NGFF适配器)。附加模块可以提供蓝牙能力(例如,具有向后兼容性的蓝牙4.0)以及Intel®无线显示功能。此外,NFC支持可以经由单独的设备或多功能设备提供,并且可以作为示例定位在机箱的右前部分中以便于访问。另外的模块可以是WWAN设备,其可以提供对3G/4G/LTE和GPS的支持。该模块可以在内部(例如,NGFF)插槽中实现。可以为WiFiTM、蓝牙、WWAN、NFC和GPS提供集成天线支持,实现从WiFiTM到WWAN无线电的无缝过渡,根据无线千兆规范(2010年7月)无线千兆位(WiGig),反之亦然。

[0130] 如上所述,集成相机可以结合在盖子中。作为一个示例,该相机可以是高分辨率相机,例如,具有至少200万像素(MP)的分辨率并且延伸到6.0MP或更高。

[0131] 为了提供音频输入和输出,可以经由数字信号处理器(DSP)实现音频处理器,DSP可以经由高清晰度音频(HDA)链路耦合到处理器。类似地,DSP可以与集成的编码器/解码器(CODEC)和放大器进行通信,该集成的编码器/解码器和放大器继而可以耦合到可以在机箱内实现的输出扬声器。类似地,放大器和CODEC可以耦合以从麦克风接收音频输入,在实施例中,麦克风可以经由双阵列麦克风(例如数字麦克风阵列)实现,以提供高质量的音频输入,以实现系统内的各种操作的语音激活控制。另请注意,音频输出可以从放大器/CODEC提供给耳机插孔。

[0132] 在特定实施例中,数字音频编解码器和放大器能够驱动立体声耳机插孔、立体声麦克风插孔、内部麦克风阵列和立体声扬声器。在不同的实现方式中,编解码器可以集成到音频DSP中,或者经由HD音频路径耦合到外围控制器集线器(PCH)。在一些实现方式中,除了集成立体声扬声器之外,还可以提供一个或多个低音扬声器,并且扬声器解决方案可以支持DTS音频。

[0133] 在一些实施例中,处理器可以由外部电压调节器(VR)和集成在处理器管芯内部的多个内部电压调节器(称为完全集成的电压调节器(FIVR))供电。在处理器中使用多个FIVR使得能够将组件分组到单独的电源平面中,使得电力由FIVR调节并仅供给组中的那些组件。在电源管理期间,当处理器处于某个低功率状态时,一个FIVR的给定电源平面可以掉电或断电,而另一个FIVR的另一个电源平面保持活动或完全供电。

[0134] 在一个实施例中,可以在一些深度睡眠状态期间使用维持电源平面来为用于若干I/O信号的I/O引脚供电,例如处理器和PCH之间的接口,与外部VR的接口和与EC的接口。该维持电源平面还为片上电压调节器供电,该电压调节器支持板载SRAM或在睡眠状态期间存储处理器上下文的其他高速缓冲存储器。维持电源平面还用于为处理器的唤醒逻辑供电,该唤醒逻辑监视和处理各种唤醒源信号。

[0135] 在功率管理期间,当处理器进入某些深度睡眠状态时,而其他电源平面掉电或断电时,维持电源平面保持通电以支持上述组件。但是,当不需要这些组件时,这会导致不必要的功耗或耗散。为此,实施例可以提供连接待机睡眠状态以使用专用电源平面来维持处理器上下文。在一个实施例中,连接的待机睡眠状态有助于使用PCH的资源进行处理器唤醒,PCH本身可以与处理器一起存在于封装中。在一个实施例中,连接待机睡眠状态有助于维持PCH中的处理器架构功能直到处理器唤醒,这使得能够关闭先前在深度睡眠状态期间保持通电的所有不必要的处理器组件,包括关闭所有时钟。在一个实施例中,PCH包含时间戳计数器(TSC)和连接待机逻辑,用于在连接待机状态期间控制系统。用于维持电源平面的集成电压调节器也可以驻留在PCH上。

[0136] 在一个实施例中,在连接待机状态期间,集成电压调节器可以用作专用电源平面,其保持通电以支持存储处理器上下文的专用高速缓冲存储器,例如当处理器进入深度睡眠状态和连接待机状态时的临界状态变量。该临界状态可以包括与架构、微架构、调试状态相关联的状态变量和/或与处理器相关联的类似状态变量。

[0137] 在连接待机状态期间,来自EC的唤醒源信号可以被发送到PCH而不是处理器,使得PCH可以管理唤醒处理而不是处理器。此外,TSC保留在PCH中以便于维持处理器架构功能。

[0138] 处理器中的功率控制可以导致增强的功率节省。例如,可以在核之间动态分配功率,各个核可以改变频率/电压,并且可以提供多个深度低功率状态以实现非常低的功耗。另外,核心或独立核心部分的动态控制可以通过在不使用组件时将组件断电来提供降低的功耗。

[0139] 一些实现方式可以提供特定的电源管理IC (PMIC) 来控制平台电源。使用该解决方案,当处于给定待机状态时,例如当处于Win8连接待机状态时,系统可以在延长的持续时间(例如,16小时)内看到非常低(例如,小于5%)的电池劣化。在Win8空闲状态下,可以实现超过例如9小时的电池寿命(例如,150尼特)。关于视频回放,可以实现长的电池寿命,例如,全高清视频回放可以发生最少6小时。在一个实现方式中,平台可以具有使用SSD的用于Win8 CS的例如35瓦特小时(Whr)的能量容量和使用具有RST高速缓存配置的HDD的用于Win8 CS的(例如)40-44Whr的能量容量。

[0140] 特定实现方式可以提供对15W标称CPU热设计功率(TDP)的支持,具有高达大约25W TDP设计点的可配置CPU TDP。由于上述热特征,平台可包括最小的通风口。此外,该平台是枕头友好的(因为没有热空气吹向用户)。取决于机箱材料可以实现不同的最高温度点。在塑料机箱的一种实现方式中(至少必须具有塑料的盖子或基部),最大操作温度可以是52摄氏度(°C)。并且对于金属机箱的实现方式,最高工作温度可以是46°C。

[0141] 在不同的实现方式中,诸如TPM的安全模块可以集成到处理器中,或者可以是诸如TPM 2.0设备的分立设备。通过集成安全模块(也称为平台信任技术(PTT)),可以启用BIOS/固件以公开某些安全功能的某些硬件功能,包括安全指令、安全启动、Intel®防盗技术、Intel®身份保护技术、Intel®可信执行技术(TXT)和Intel®可管理性引擎技术以及安全的用户界面,如安全的键盘和显示器。

[0142] 尽管已经关于有限数量的实施例描述了本发明,但是本领域技术人员将意识到由此产生的许多修改和变化。所附权利要求旨在覆盖落入本发明的真实精神和范围内的所有这些修改和变化。

[0143] 设计可以经历从创建到模拟到制造的各个阶段。表示设计的数据可以以多种方式表示设计。首先,如在模拟中有用的,可以使用硬件描述语言或另一种功能描述语言来表示硬件。另外,可以在设计过程的某些阶段产生具有逻辑和/或晶体管栅极的电路级模型。此外,大多数设计在某个阶段达到表示硬件模型中各种设备的物理位置的数据级别。在使用传统半导体制造技术的情况下,表示硬件模型的数据可以是指定用于制造集成电路的掩模在不同掩模层上存在或不存在各种特征的数据。在设计在任何表示中,数据可以存储在任何形式的机器可读介质中。存储器或诸如盘的磁或光存储装置可以是机器可读介质,用于存储经由调制或以其他方式生成以发送这样的信息的光波或电波发送的信息。当发送指示或携带代码或设计的电载波时,在执行电信号的复制、缓冲或重传的程度上,制作新的副

本。因此,通信提供商或网络提供商可以将体现本公开的实施例的技术的诸如编码到载波中的信息的制品至少临时存储在有形的机器可读介质上。

[0144] 本文所使用的模块是指硬件、软件和/或固件的任意组合。作为示例,模块包括硬件,诸如微控制器,与用于存储适于由微控制器执行的代码的非暂时介质相关联。因此,在一个实施例中对模块的引用是指特别配置成识别和/或执行要保持在非暂时介质上的代码的硬件。此外,在另一实施例中,模块的使用是指包括特别适于由微控制器执行以实施预定操作的代码的非暂时介质。并且,如可推断的,在又一实施例中,术语模块(在该示例中)可以是指微控制器和非暂时介质的组合。通常,图示为单独的模块界限通常变化并且可能重叠。例如,第一模块和第二模块可以共享硬件、软件、固件或其组合,同时可能保留一些独立的硬件、软件或固件。在一个实施例中,术语逻辑的使用包括硬件,诸如晶体管、寄存器或其它硬件,如可编程逻辑器件。

[0145] 在一个实施例中,短语“用于”或“被配置为”的使用是指布置、放在一起、制造、提供以售卖、进口和/或设计用于执行指定或确定的任务的装置、硬件、逻辑或元件。在该示例中,没有进行操作的装置或其元件仍‘被配置为’执行指定任务,前提是其被设计、耦合和/或互连来执行所述指定任务。纯粹作为说明性的示例,逻辑门可以在操作期间提供0或1。但是‘被配置为’向时钟提供使能信号的逻辑门不包括可提供1或0的每一个可能的逻辑门。相反,该逻辑门是以在操作期间1或0输出用于使能时钟的某种方式耦合的逻辑门。再次注意到,术语‘被配置为’的使用不要求操作,而是侧重于装置、硬件和/或元件的潜在状态,其中在潜在状态下,装置、硬件和/或元件设计成当装置、硬件和/或元件正在操作时执行特定任务。

[0146] 此外,在一个实施例中,短语“能够/能”和/或“可操作用于”的使用是指以这样的方式设计来使能按指定方式使用装置、逻辑、硬件和/或元件的某种装置、逻辑、硬件和/或元件。如上文所提到的,在一个实施例中,用于、能够用于或可操作用于的使用是指装置、逻辑、硬件和/或元件的潜在状态,其中装置、逻辑、硬件和/或元件不是正在操作,但是被设计成使得能够以指定方式使用装置。

[0147] 如本文所使用的,值包括数字、状态、逻辑状态或二进制逻辑状态的任何已知表示。通常,逻辑电平、逻辑值或逻辑上的值还称为1和0,简单地表示二进制逻辑状态。例如,1表示高逻辑电平,而0表示低逻辑电平。在一个实施例中,诸如晶体管或闪存单元的存储单元能够保持单个逻辑值或多个逻辑值。但是,已经使用了计算机系统其他值表示。例如,十进制数十也可以表示为二进制值1010和十六进制字母A。因此,值包括能够保存在计算机系统信息中的信息的任何表示。

[0148] 此外,状态可由值或值的部分表示。作为示例,诸如逻辑1的第一值可以表示默认或初始状态,而诸如逻辑0的第二值可以表示非默认值。另外,在一个实施例中,术语复位和置位分别是指默认值和更新值或状态。例如,默认值可能包括高逻辑值,即复位,而更新的值可能包括低逻辑值,即置位。注意,值的任何组合可用于表示任意数量的状态。

[0149] 以下实施例涉及根据本说明书的实施例。示例1是一种包括一组寄存器和计算块的接口的装置。计算块包括物理层块或介质访问控制层块中的一个,并且接口包括用于发送异步信号的一个或多个引脚,用于接收异步信号的一个或多个引脚,以及用于传送特定信号以访问该组寄存器的一组引脚,其中定义的接口的一组控制和状态信号被映射到该组

寄存器的相应位。

[0150] 示例2可以包括示例1的主题,其中所述接口还包括:用于发送定时关键信号的一个或多个引脚;以及用于接收定时关键信号的一个或多个引脚。

[0151] 示例3可以包括示例1-2中任一个的主题,其中所述接口还包括用于传送数据信号的一组引脚。

[0152] 示例4可以包括示例1-3中任一个的主题,其中该组寄存器的位的值用于代替所述控制和状态信号。

[0153] 示例5可以包括示例4的主题,其中所述定义的接口包括用于快速外围组件互连(PCIe)架构(PIPE)接口的PHY接口。

[0154] 示例6可以包括示例5的主题,其中所述接口包括比在特定PIPE规范中使用的引脚少的引脚。

[0155] 示例7可以包括示例6的主题,其中所述接口包括小于所述特定PIPE规范的引脚数量的一半的引脚。

[0156] 示例8可以包括示例4-7中任一项的主题,其中所述特定信号包括命令,并识别该组寄存器中的特定一个的地址。

[0157] 示例9可以包括示例8的主题,其中所述命令包括写入,并使得值被写入与所述一组控制和状态信号中的特定一个相对应的特定寄存器。

[0158] 示例10可以包括示例1-9中任一项的主题,其中所述一组控制和状态信号包括用于多个定义的互连协议的信号。

[0159] 示例11可以包括示例10的主题,其中所述多个定义的互连协议包括PCIe、通用串行总线(USB)和串行高级技术附件(SATA)。

[0160] 示例12是其上存储有代码的存储介质,其中代码在机器上执行时使机器用于:在接口的特定部分上接收信号,其中该接口将第一计算块耦合到第二计算块并且与控制层与物理层之间的定义的接口相对应,该信号从所述第二计算块接收并且包括一个命令,其中该命令识别该命令要应用于的所述第一计算块的一组寄存器中的特定一个的特定地址,并包括要写入该特定寄存器的一个或多个位的值,并且这些值至少指示所定义的接口的一组控制和状态信号中的特定一个;基于命令来修改所述特定寄存器的一个或多个位的值;以及将特定寄存器的修改的一个或多个值解释为接收所述控制和状态信号中特定一个。

[0161] 示例13可以包括示例12的主题,其中该命令包括写入。

[0162] 示例14可以包括示例13的主题,其中所述命令包括未提交写入,并且所述一个或多个位的值并不进行修改直到后续事务中包括已提交写入的后续实例。

[0163] 示例15可以包括示例12-13中任一项的主题,其中,所述代码在被执行时还生成要通过所述接口的特定部分发送的数据,以指示对所述特定寄存器的所述一个或多个位的值已被修改的确认。

[0164] 示例16可以包括示例12-15中任一项的主题,其中所述命令是多个支持的命令中的一个,所述多个支持的命令包括已提交写命令、未提交写命令、读命令、读完成和确认命令。

[0165] 示例17可以包括示例12-15中任一项的主题,其中,所述代码在被执行时还根据所述命令确定与所述数据相关联的事务的长度。

[0166] 示例18可以包括示例17的主题,其中,所述信号包括第一信号,所述事务包括第一事务,并且所述代码在被执行时还用于:在所述第一信号之后接收第二信号,并且基于所确定的事务长度,确定所述第二信号与紧接在所述第一事务之后的另一事务的开始相对应。

[0167] 示例19可以包括示例12-18中任一示例的主题,其中,所述代码在被执行时还确定所述一组控制和状态信号中的特定一个的类型,并使得一个或多个位的值在与该组控制和状态信号中的特定一个的类型相对应的预定数量的单位间隔期满之后,自动返回到默认值。

[0168] 示例20是一种方法,包括:在接口上对接收数据进行接收,所述接口用于将第一计算块耦合到第二计算块,其中该接口是根据控制层与物理层之间的定义的接口的,该信号从第二计算块接收,该数据包括命令,识别命令要应用于的第一计算块的一组寄存器中的特定一个的特定地址,并且包括要写入特定寄存器的一个或多个位的值,并且该数据与事务相对应以至少指示所定义的接口的一组控制和状态信号中的一个特定一个;基于所述命令修改所述特定寄存器的一个或多个位的值;以及将所述特定寄存器的修改的一个或多个值解释为接收所述控制和状态信号中特定一个。

[0169] 示例21是包括用于执行示例20的方法的单元的系统。

[0170] 示例22可以包括示例21的主题,其中该单元包括硬件电路。

[0171] 示例23是一种系统,包括:物理层计算块,包括第一组寄存器,以及控制器计算块,其中所述控制器计算块包括第二组寄存器和与物理层计算块耦合的接口。所述接口包括:用于将异步信号发送到所述物理层计算块的一个或多个引脚;用于从所述物理层计算模块接收异步信号的一个或多个引脚;以及一组引脚,包括用于将特定信号传送到物理层计算块以访问第一组寄存器和引脚以从物理层计算块接收信号以准许访问所述第二组寄存器的引脚,其中定义的接口的一组控制和状态信号被映射到第一和第二组寄存器的相应位。

[0172] 示例24可以包括示例23的主题,其中控制器计算块包括介质访问控制层计算块。

[0173] 示例25可以包括示例24的主题,其中所述物理层计算块包括快速外围组件互连(PCIe)物理层。

[0174] 示例26可以包括示例24的主题,其中所述物理层计算块包括通用串行总线(USB)物理层。

[0175] 示例27可以包括示例24的主题,其中所述物理层计算块包括串行高级技术附件(SATA)物理层。

[0176] 上文所阐述的方法、硬件、软件、固件或代码集的实施例可以经由存储在机器可访问、机器可读、计算机可访问或计算机可读介质上的、能够由处理元件执行的指令或代码来实现。非暂时性机器可访问/可读介质包括提供(即,存储和/或发送)诸如计算机或电子系统的机器能够读取的形式的信息的任何机制。例如,非暂时性机器可访问介质包括随机存取存储器(RAM),如静态RAM(SRAM)或动态RAM(DRAM);ROM;磁或光存储介质;闪存设备;电存储设备;光存储设备;声存储设备;用于保存从暂时(传播)信号(例如,载波、红外信号、数字信号)接收到的信息的其它形式的存储设备;等等,其区别于可从其接收信息的非暂时性介质。

[0177] 用于对逻辑进行编程以执行本公开的实施例的指令可以存储在系统中的存储器内,如DRAM、高速缓存、闪存存储器或其它存储装置。此外,指令可以经由网络或通过其它计

计算机可读介质来分发。因此,机器可读介质可以包括用于以机器(例如,计算机)可读形式存储或发送信息的任何机制,但不限于软盘、光盘、压缩盘、只读存储器(CD-ROM)和磁光盘、只读存储器(ROM)、随机存取存储器(RAM)、可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、磁卡或光卡、闪速存储器或用于经由电、光、声或其它形式的传播信号(例如,载波、红外信号、数字信号等)通过因特网传输信息的有形的机器可读存储装置。因此,计算机可读介质包括适合于存储或发送机器(例如,计算机)可读形式的电子指令或信息的任何类型的有形机器可读介质。

[0178] 在本说明书通篇提到“一个实施例”或“实施例”是指结合该实施例描述的特定的特征、结构或特性包括在本公开的至少一个实施例中。因此,在本说明书通篇各处出现的短语“在一个实施例中”或“在实施例中”不一定都指同一实施例。此外,特定的特征、结构或特性可以任何适合的方式结合在一个或多个实施例中。

[0179] 在前面的说明书中,已经参考具体的示例性实施例进行了详细说明。然而,将显而易见的是,可以对其进行各种修改和改变,而不偏离如随附权利要求中阐述的本公开的更宽泛的精神和范围。说明书和附图因此应在说明性的意义而非限制性的意义上考虑。此外,前面对实施例和其它示范性语言的使用不一定是指同一实施例或同一示例,但是可以指不同和独特的实施例以及可能同一实施例。

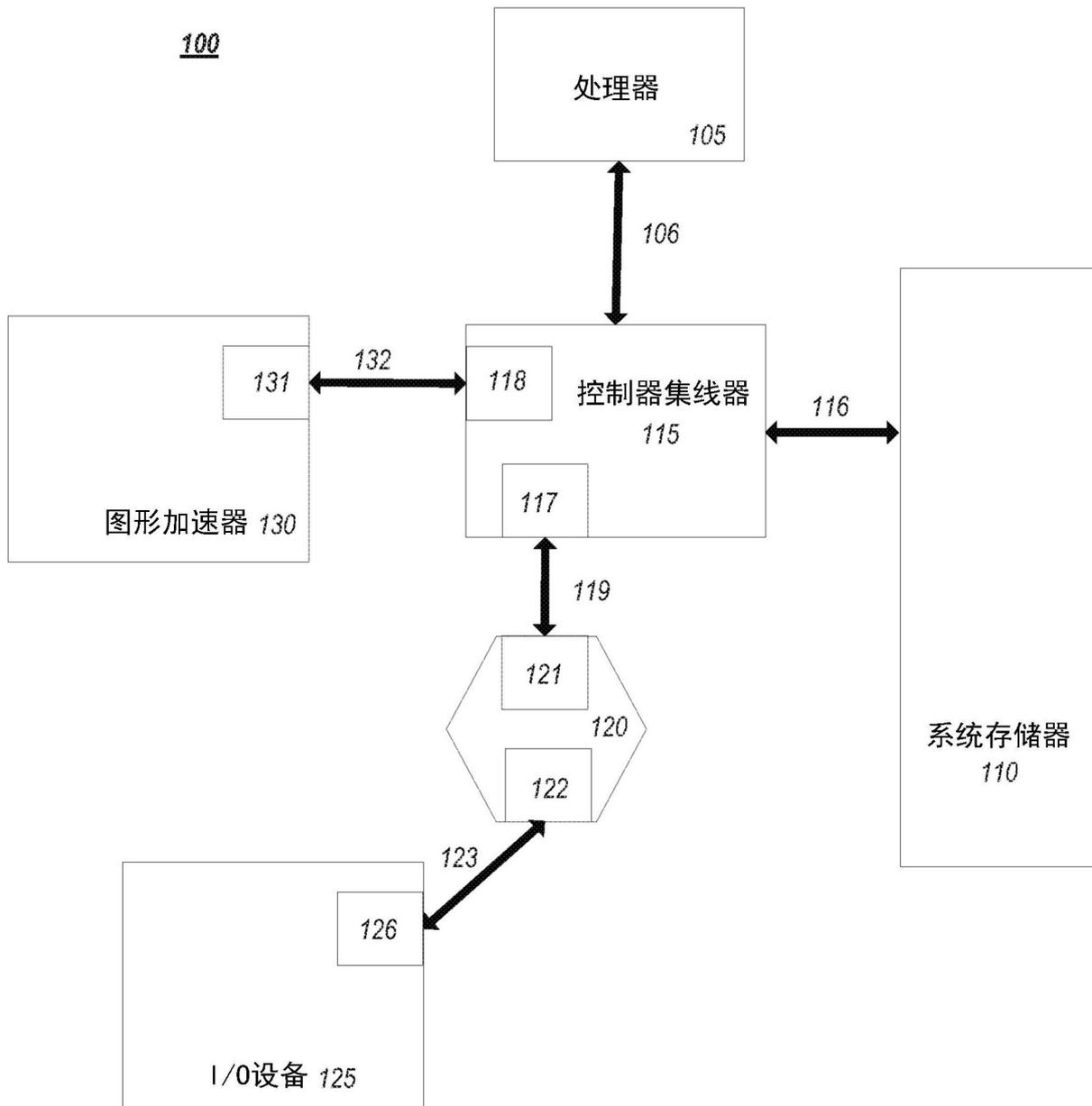


图1

分层协议栈 200

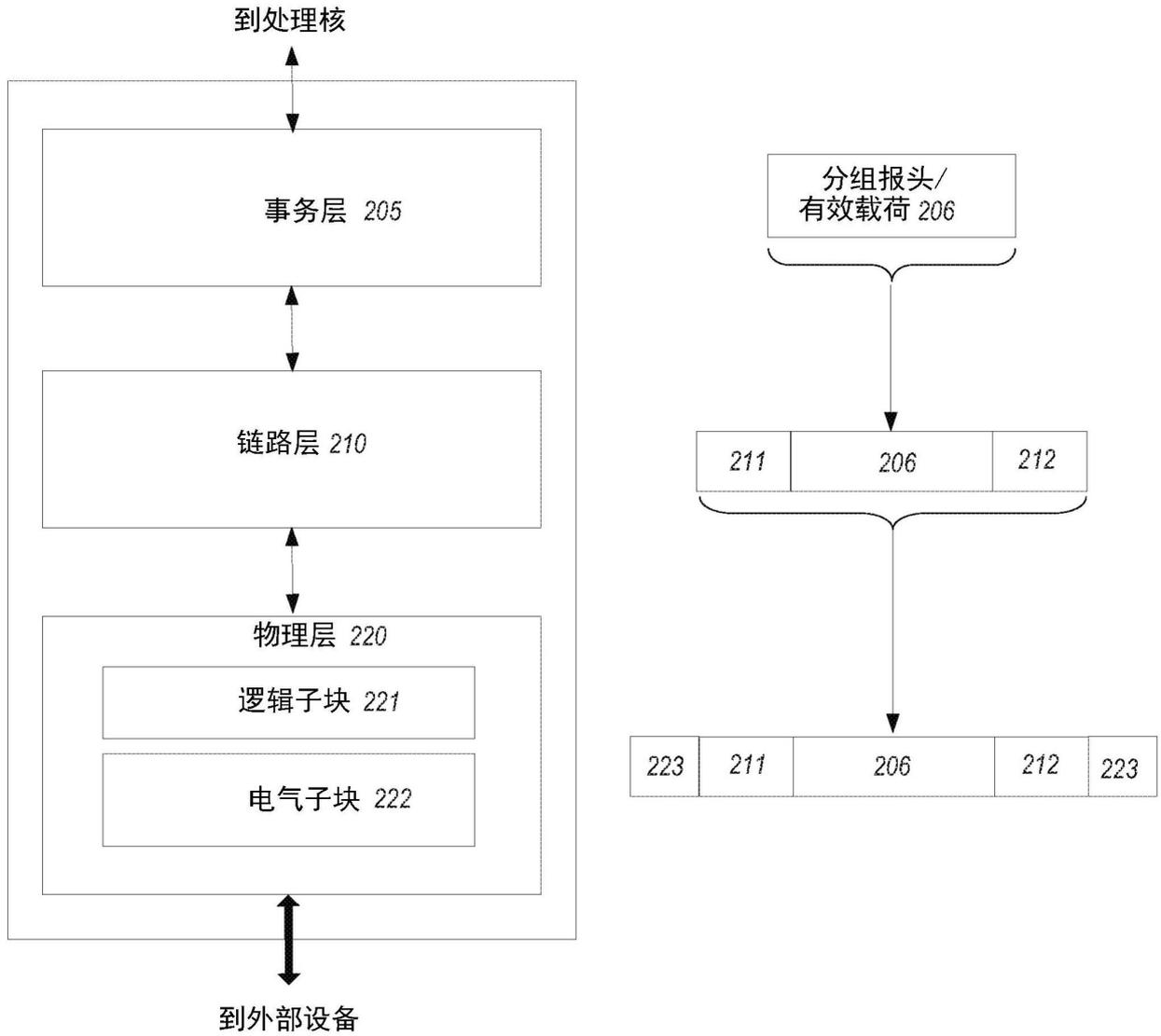


图2



图3

400

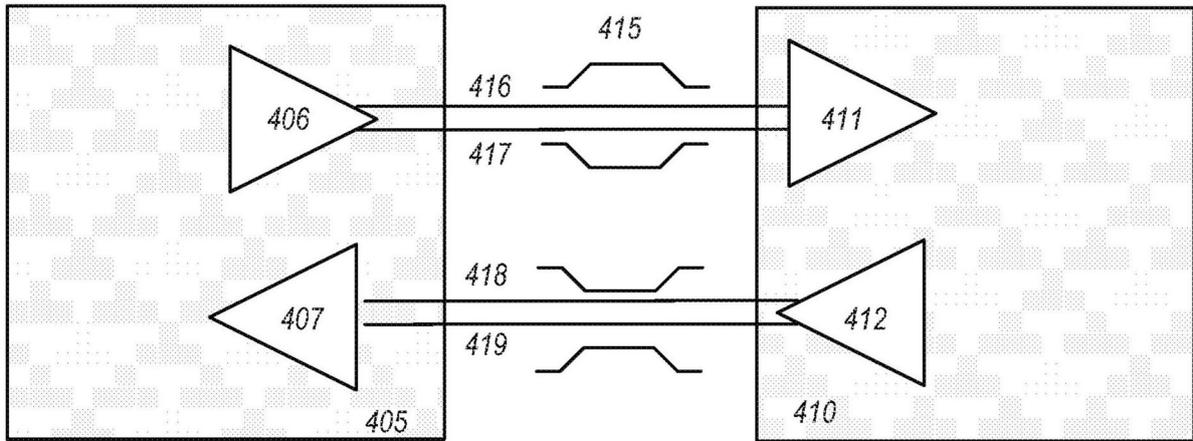


图4

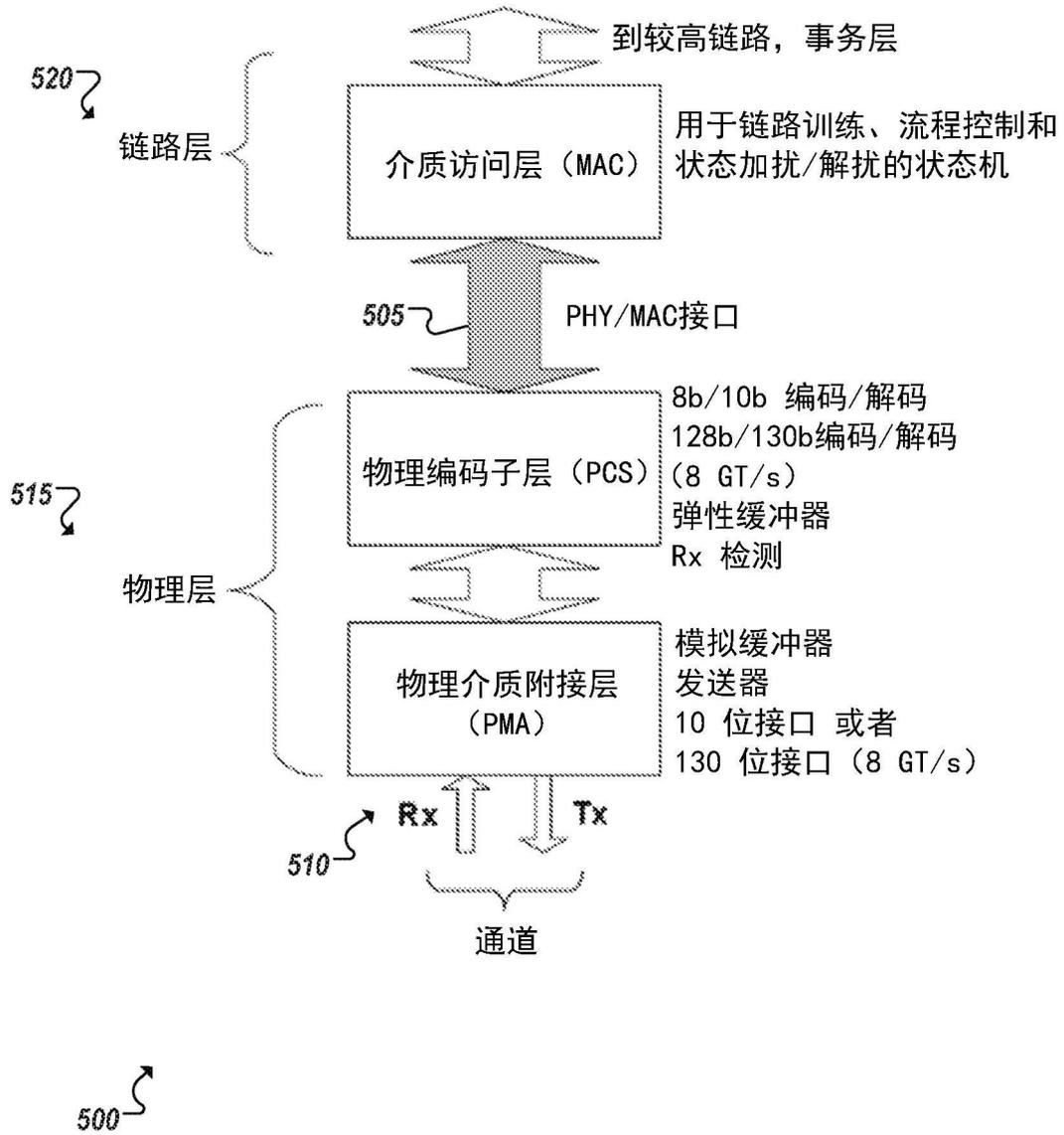


图5

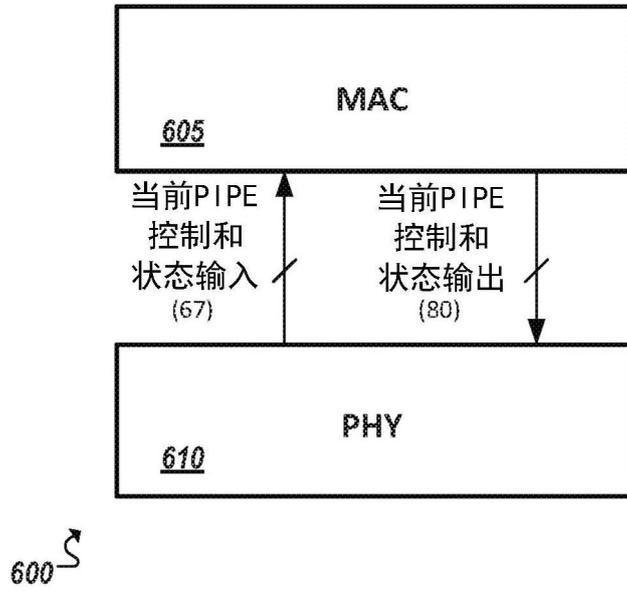


图6

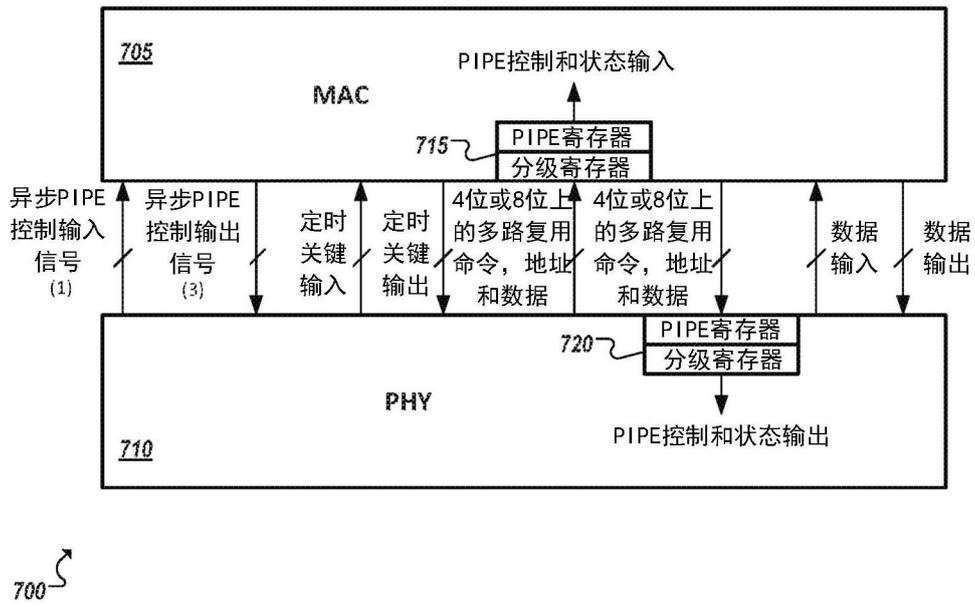


图7

800 ↷

地址	位	信号名称
12'h000	[3:0]	PHY 模式 [3:0]
	[4]	弹性缓冲器模块
	[5]	Tx检测Rx/循环
	[6]	TxElecdle
	[7]	<预留>
12'h001	[1:0]	TX模式 [1:0]
	[2]	TxCompliance
	[3]	TxOnesZeros
	[4]	RxPolarity
	[5]	RxEqTraining
	[7:6]	<预留>

图8

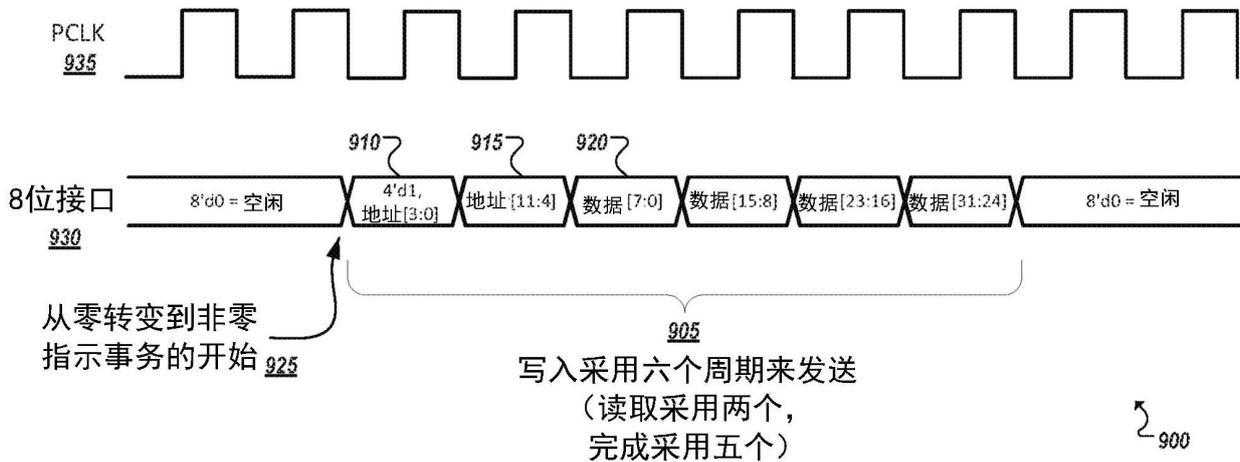


图9

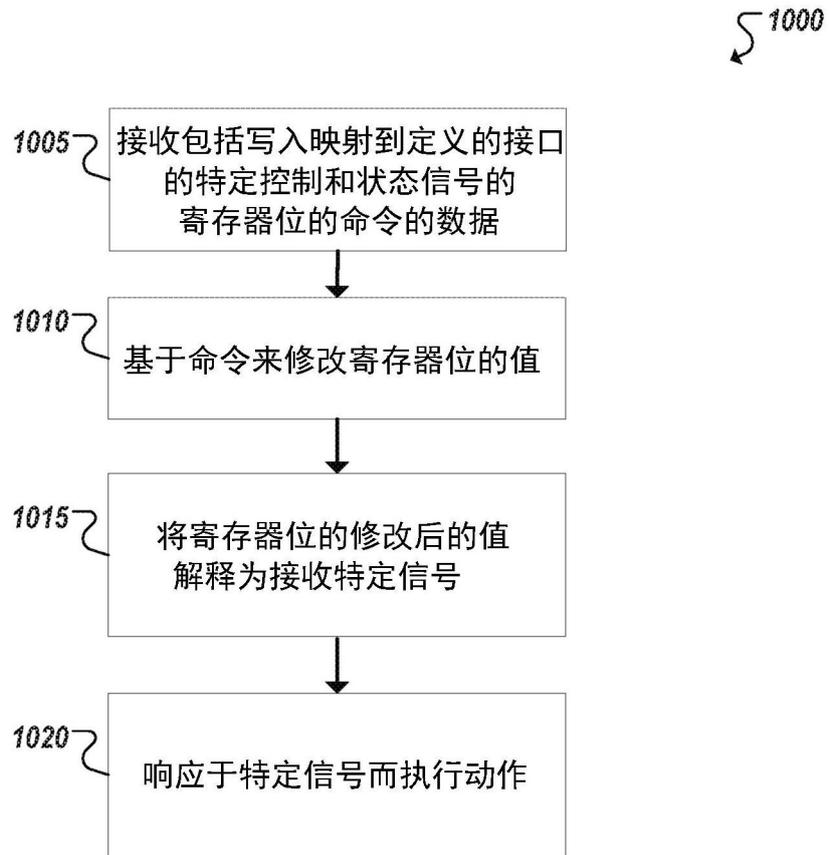


图10

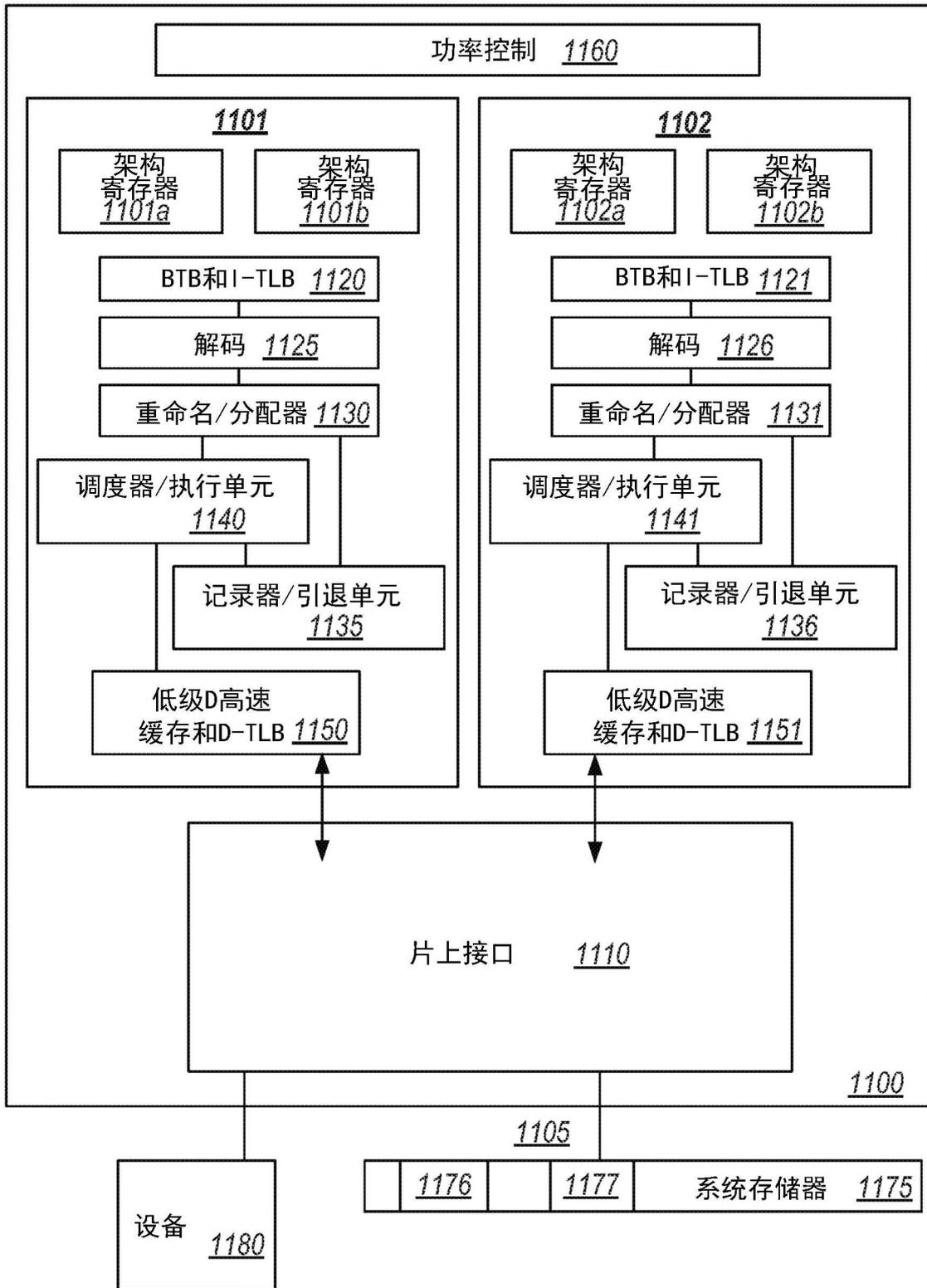


图11

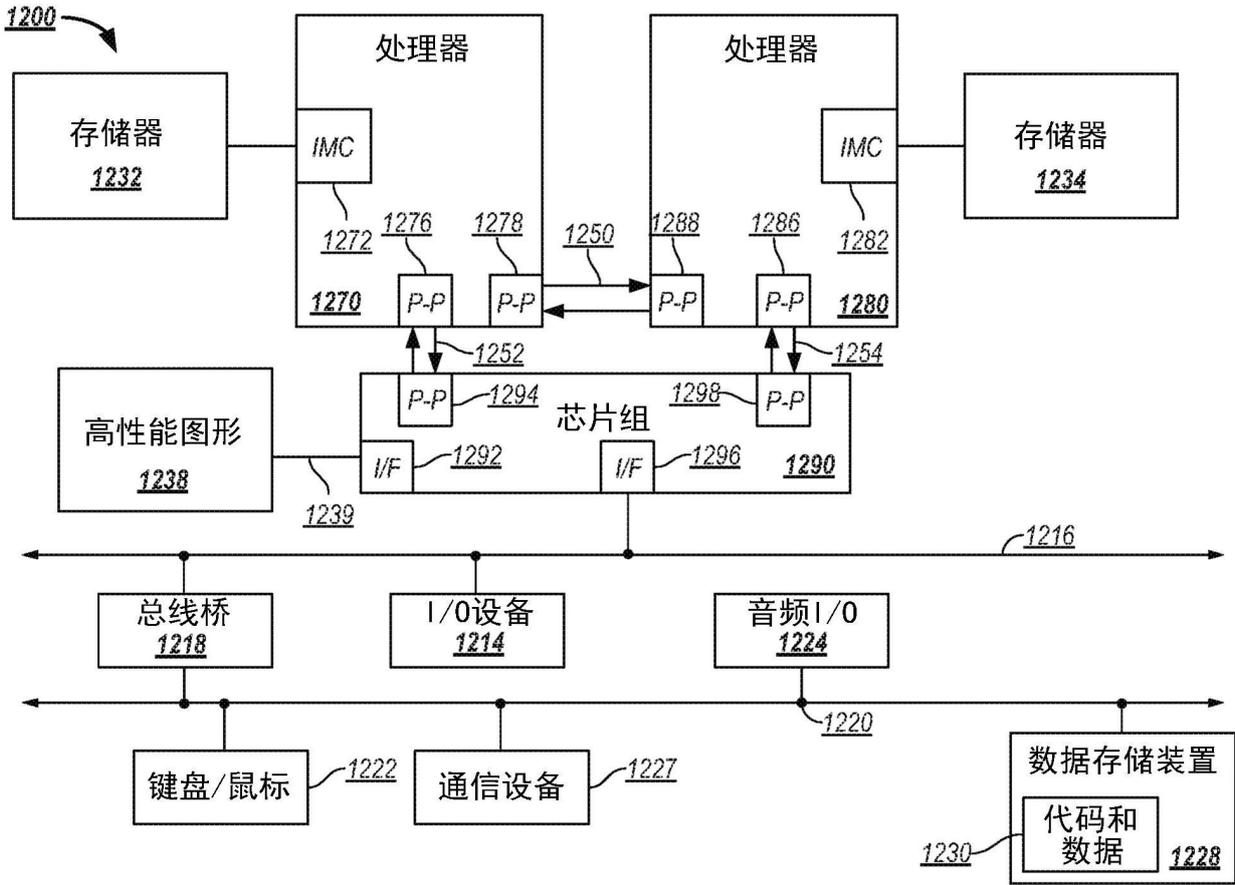


图12