(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 112543925 B
(45)授权公告日 2024.08.09

(21)申请号 201980050194.7
(22)申请日 2019.07.25
(65)同一申请的已公布的文献号
申请公布号 CN 112543925 A
(43)申请公布日 2021.03.23
(30)优先权数据
16/046,602 2018.07.26 US
(85)PCT国际申请进入国家阶段日
2021.01.26
(86)PCT国际申请的申请数据
PCT/US2019/043521 2019.07.25
(87)PCT国际申请的公布数据
WO2020/023797 EN 2020.01.30
(73)专利权人 赛灵思公司
地址 美国加利福尼亚州
(72)发明人 S·辛格 H·C·耐玛 S·珊坦 K·K·道 K·科比特 Y·王 C·J·凯斯
(74)专利代理机构 北京市君合律师事务所 11517
专利代理师 毛健 程烁宇
(51)Int.Cl.
G06F 13/28(2006.01)
G06F 13/40(2006.01)
(56)对比文件
WO 2018064418 A1,2018.04.05
Tal Ben-Nun等.“Groute: An Asynchronous Multi-GPU Programming Model for Irregular Computations”.《PPoPP '17: Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming》.2017,第235-248页.
Tal Ben-Nun等.“Groute: An Asynchronous Multi-GPU Programming Model for Irregular Computations”.《PPoPP '17: Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming》.2017,第235-248页.
审查员 黄彰

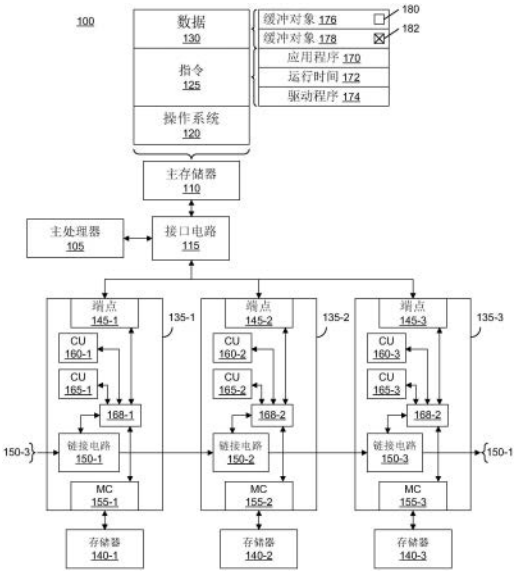
权利要求书2页 说明书18页 附图6页

(54)发明名称

用于使用专用低延迟链路的多个硬件加速器的统一地址空间

(57)摘要

系统可以包括耦接到通信总线的主机处理器(105);通过通信总线通信地链接到主机处理器(105)的第一硬件加速器(135-1);以及通过通信总线通信地链接到主处理器(105)的第二硬件加速器(135-2)。第一硬件加速器(135-1)和第二硬件加速器(135-2)通过独立于通信总线的加速器链接而直接耦接。主机处理器(105)被配置为发起在第一硬件加速器(135-1)和第二硬件加速器(135-2)之间直接通过加速器链接的数据传输。



CN 112543925 B

1.一种硬件加速系统,其特征在于,所述系统包括:

主处理器,所述主处理器耦接到通信总线;

第一硬件加速器,所述第一硬件加速器通过所述通信总线可通信地链接到所述主处理器;以及

第二硬件加速器,所述第二硬件加速器通过所述通信总线可通信地链接到所述主处理器;

其中,所述第一硬件加速器和所述第二硬件加速器通过独立于所述通信总线的加速器链接直接耦接;以及

其中,所述主处理器被配置为发起所述第一硬件加速器和所述第二硬件加速器之间直接通过所述加速器链接的数据传输,其中,所述第二硬件加速器被配置为响应于经由所述加速器链接接收到事务,将所述数据传输的目标地址递减所述第二硬件加速器的地址范围的上限,以及确定递减的目标地址是否为本地地址,如果所述递减的目标地址不是本地地址,所述事务被转发到下一个硬件加速器。

2.根据权利要求1所述的硬件加速系统,其特征在于,所述数据传输包括:所述第一硬件加速器通过所述加速器链接访问所述第二硬件加速器的存储器。

3.根据权利要求2所述的硬件加速系统,其特征在于,所述主处理器被配置为通过向所述第一硬件加速器发送包括目标地址的数据来访问所述第二硬件加速器的所述存储器,其中,所述目标地址由所述主处理器转换以对应于所述第二硬件加速器,以及所述第一硬件加速器基于所述目标地址发起通过加速器链接访问所述第二硬件加速器的存储器的事务。

4.根据权利要求1所述的硬件加速系统,其特征在于,所述主处理器被配置为基于耦接到所述通信总线的所述第二硬件加速器的直接存储器访问电路的状态,来发起在所述第一硬件加速器和所述第二硬件加速器之间的数据传输。

5.根据权利要求1所述的硬件加速系统,其特征在于,所述主处理器被配置为自动确定在环形拓扑中所述第一硬件加速器和所述第二硬件加速器的序列。

6.根据权利要求1所述的硬件加速系统,其特征在于,所述主处理器被配置为使用远程缓冲器标志来跟踪对应于所述第一硬件加速器和所述第二硬件加速器的缓冲器。

7.一种集成电路,其特征在于,所述集成电路包括:

端点,所述端点被配置为通过通信总线与主处理器进行通信;

存储器控制器,所述存储器控制器耦接到所述集成电路本地的存储器;以及

链接电路,所述链接电路耦接到所述端点和所述存储器控制器,其中所述链接电路被配置为与同样耦接到所述通信总线的目标硬件加速器建立加速器链接,其中所述加速器链接是独立于所述通信总线的、在所述集成电路和所述目标硬件加速器之间的直接连接,其中,所述目标硬件加速器被配置为将从所述集成电路接收的事务中的目标地址递减所述集成电路的地址范围的上限,以及确定递减的目标地址是否为本地地址,如果所述递减的目标地址不是本地地址,所述事务被转发到下一个硬件加速器。

8.根据权利要求7所述的集成电路,其特征在于,所述链接电路被配置为通过所述加速器链接发起与所述目标硬件加速器的数据传输,并且所述数据传输是响应于由所述集成电路通过所述通信总线从所述主处理器接收的指令而发生的。

9.一种硬件加速方法,其特征在于,所述方法包括:

在第一硬件加速器中,接收通过通信总线从主处理器发送的用于数据传输的指令和目标地址;

所述第一硬件加速器将所述目标地址与所述第一硬件加速器对应的地址范围的上限进行比较;

响应于基于比较确定所述目标地址超出了所述地址范围,所述第一硬件加速器发起与第二硬件加速器的事务,从而通过使用直接耦接所述第一硬件加速器和所述第二硬件加速器的加速器链接执行数据传输;以及

响应于在所述第二硬件加速器中接收到所述事务,所述第二硬件加速器从所述目标地址减去所述第二硬件加速器的地址范围的上限,并确定相减的结果是否在所述第二硬件加速器的地址范围内, 如果所述相减的结果不在所述第二硬件加速器的地址范围内,所述事务被转发到下一个硬件加速器。

10.根据权利要求9所述的硬件加速方法,其特征在于,所述加速器链接独立于所述通信总线。

11.根据权利要求9所述的硬件加速方法,其特征在于,所述方法还包括:

确定所述第二硬件加速器的直接存储器访问电路的状态;以及响应于所述第二硬件加速器的直接存储器访问电路的状态来发起所述数据传输。

12.根据权利要求9所述的硬件加速方法,其特征在于,所述数据传输包括:所述第一硬件加速器通过所述加速器链接访问所述第二硬件加速器的存储器。

用于使用专用低延迟链路的多个硬件加速器的统一地址空间

技术领域

[0001] 本公开涉及硬件加速,更具体地,涉及通过统一的地址空间和低等待时间的通信链路来促进多个硬件加速器的使用。

背景技术

[0002] 异构计算平台(HCP)是指一种数据处理系统,其包括通过接口电路耦接到一个或多个其他设备的主机处理器。这些设备通常在架构上不同于主机处理器。主机处理器能够将任务卸载到设备。这些设备能够执行任务并将结果提供给主机处理器。作为说明性示例,主机处理器通常被实现为中央处理单元,而设备被实现为图形处理单元(GPU)和/或数字信号处理器(DSP)。

[0003] 在其他HCP中,执行从主机处理器卸载的任务的一个或多个设备包括适用于硬件加速的设备(称为“硬件加速器”)。硬件加速器包括能够执行从主机卸载的任务的电路,与执行该任务的软件或程序代码相反。硬件加速器的电路在功能上等同于执行软件,但通常能够在更短的时间内完成任务。

[0004] 硬件加速器的示例包括可编程集成电路(IC),例如现场可编程门阵列(FPGA),部分可编程IC,专用IC(ASIC)等。适当地,HCP可以包括不同设备的组合,其中一个或多个适于执行程序代码,而一个或多个其他适于硬件加速。

发明内容

[0005] 在一个或多个实施例中,系统可包括:耦接到通信总线的主处理器;通过通信总线可通信地链接到主处理器的第一硬件加速器;以及通过通信总线可通信地链接到主处理器的第二硬件加速器。第一硬件加速器和第二硬件加速器通过独立于通信总线的加速器链接直接耦接。主处理器被配置为发起第一硬件加速器和第二硬件加速器之间直接通过所述加速器链接的数据传输。

[0006] 在一个或多个实施例中,硬件加速器可以包括:端点,被配置为通过通信总线与主处理器进行通信;存储器控制器,其耦接到所述硬件加速器本地的存储器;以及链接电路,其被耦接到所述端点和所述存储器控制器。链接电路被配置为与也耦接到通信总线的目标硬件加速器建立加速器链接。加速器链接是独立于通信总线的、在硬件加速器和目标硬件加速器之间的直接连接。

[0007] 在一个或多个实施例中,一种方法可以包括:在第一硬件加速器内接收通过通信总线从主处理器发送的用于数据传输的指令和目标地址,第一硬件加速器将目标地址与上限进行比较。第一硬件加速器对应于第一硬件加速器的地址范围,并且响应于基于比较确定目标地址超出了地址范围,第一硬件加速器发起与第二硬件加速器的事务,以通过使用直接耦接所述第一硬件加速器和所述第二硬件加速器的加速器链接执行数据传输。

[0008] 提供本发明内容部分仅是为了引入某些概念,而不是标识所要求保护的主题的任何关键或必要特征。根据附图和以下详细描述,本发明布置的其他特征将显而易见。

附图说明

[0009] 在附图中示例性地示出了本发明的布置。然而,附图不应被理解为将本发明的布置限制为仅示出的特定实施方式。在阅读以下详细描述并参考附图后,各个方面和优点将变得显而易见。

[0010] 图1示出了具有多个硬件加速器的系统的示例。

[0011] 图2示出了硬件加速器的示例性实现。

[0012] 图3示出了重发引擎(RTE)的示例。

[0013] 图4示出了具有多个硬件加速器的系统的示例性操作方法。

[0014] 图5示出了具有多个硬件加速器和一个或多个附加设备的系统的示例。

[0015] 图6示出了用于集成电路(IC)的示例架构。

具体实施方式

[0016] 虽然本公开以限定新颖特征的权利要求作为结尾,但是可以相信,通过结合附图考虑说明书,将更好地理解本公开中描述的各种特征。为了说明的目的,提供了本文描述的过程,机器,制造及其任何变型。本公开内容中描述的特定结构和功能细节不应被理解为限制性的,而仅仅是作为权利要求的基础,并且作为教导本领域的技术人员以实质上任何适当的详细结构来不同地采用所描述的特征的代表性基础。此外,在本公开内容中使用的术语和短语不意图是限制性的,而是提供所描述的特征的可理解的描述。

[0017] 本公开涉及硬件加速,并且更具体地,涉及通过统一的地址空间和低等待时间的通信链路来促进多个硬件加速器的使用。将硬件加速器与数据处理系统一起使用已成为一种有效的技术,可从主处理器上卸载任务,从而减少主处理器上的工作量。硬件加速器通常通过总线连接到主处理器。例如,硬件加速器可以连接到电路板,该电路板插入到主系统的可用总线插槽(slot)中。通常,每个硬件加速器都连接到相应的电路板上。向系统添加额外的硬件加速器通常需要将带有硬件加速器的额外电路板插入可用的总线插槽中。

[0018] 在常规系统中,必须更新和/或重写由主处理器执行的应用程序,以专门访问任何新添加的硬件加速器(例如,按硬件地址)。此外,为了将数据从一个硬件加速器传输到另一硬件加速器,数据将从源硬件加速器移动到主处理器,然后从主处理器向下移动到目标硬件加速器。数据通过总线通过主处理器与每个硬件加速器之间来回移动。这样,添加到系统的每个附加硬件加速器都会增加总线上的设备数量,从而导致争用总线上的带宽。随着由硬件加速器(或其他设备)执行的任务的复杂性,数量和/或大小的增加,总线上的可用带宽进一步受到限制。

[0019] 根据本公开内容中描述的发明性布置,提供了用于设备的统一地址空间。此外,提供了硬件加速器之间的直接通信链接,在此称为“加速器链接”,其能够独立于总线进行操作。主执行的运行时间库和驱动程序能够利用统一的地址空间,以便主处理器执行的应用程序可以在不直接引用(例如寻址)系统中特定硬件加速器的情况下运行。运行时间库能够确定用于实现硬件加速器之间的数据传输的正确地址。这样,无需修改应用程序即可访问可以添加到系统中的其他硬件加速器。此外,可以在加速器链接上执行数据传输,从而允许将数据直接从一个硬件加速器传输到另一个硬件加速器,而无需通过主处理器,从而有效地绕过了总线。这样,可以显著减少总线上的硬件加速器使用的带宽,从而提高整体系统性

能。

[0020] 如前所述,可以使用现有的地址空间将附加的硬件加速器添加到系统中,而无需对由主处理器执行的程序代码(例如,应用程序)进行相应的更改或修改。至少在某些情况下,这一点是通过为硬件加速器板实施自动发现过程并将此类板添加到系统中,使用远程或本地缓冲区标志,在至少某些情况下自动切换到加速器链接以进行数据传输,以及用于远程缓冲区的自动地址转换,而得以支持的。

[0021] 下面参考附图更详细地描述本发明装置的其他方面。为了说明的简单和清楚起见,附图中所示的元件未必按比例绘制。例如,为了清楚起见,一些元件的尺寸可能相对于其他元件被放大。此外,在认为适当的情况下,在附图中重复参考数字以表示相应、类似、或相似的特征。

[0022] 图1示出了具有多个硬件加速器的系统100的示例。系统100是可用于实现计算机、服务器、或其他数据处理系统的计算机硬件的示例。系统100也是异构计算系统的示例。如图所示,系统100包括至少一个通过接口电路115耦接到主存储器110的主处理器105。

[0023] 系统100还包括多个硬件加速器135。如图1所示,系统100包括三个硬件加速器135-1、135-2和135-3。虽然图1的示例显示了三个硬件加速器,但是应当理解,系统100可以包括少于三个的硬件加速器或不止三个的硬件加速器。此外,系统100可以包括一个或多个其他设备,例如图形处理单元(GPU)或数字信号处理器(DSP)。

[0024] 系统100能够在主存储器110内存储计算机可读指令(也称为“程序代码”)。主存储器110是计算机可读存储介质的示例。主处理器105能够执行经由接口电路115从主存储器110存取的程序代码。在一个或多个实施例中,主处理器105通过存储器控制器(未示出)与主存储器110通信。

[0025] 主存储器110可以包括一个或多个物理存储器设备,例如本地存储器和大容量(bulk)存储设备。本地存储器是指在程序代码的实际执行期间通常使用的非永久性存储设备。本地存储器的示例包括随机存取存储器(RAM)和/或适合于在执行程序代码(例如DRAM,SRAM,DDR SDRAM等)期间由处理器使用的各种类型的RAM中的任何一种。大容量存储设备是指永久性数据存储设备。大容量存储设备的示例包括但不限于硬盘驱动程序(HDD),固态驱动程序(SSD),闪存,只读存储器(ROM),可擦可编程只读存储器(EPROM),电可擦可编程只读存储器(EEPROM)或其他合适的存储器。系统100还可包括一个或多个高速缓冲存储器(未示出),其提供至少一些程序代码的临时存储以减少在执行期间必须从大容量存储设备中检索程序代码的次数。

[0026] 主存储器110能够存储程序代码和/或数据。例如,主存储器110可以存储操作系统120,指令125和数据130。在图1的示例中,指令125可以包括一个或多个应用程序170,运行时间库(runtime library)(在本文中称为“运行时间”)172,以及能够与硬件加速器135通信的驱动程序174。运行时间172能够处理完成事件,管理命令队列,并向应用程序170提供通知。数据130除其他类型的数据项外,还可以包括诸如缓冲对象176和178之类的缓冲对象,它们便于进行在硬件加速器135之间的直接数据传输。缓冲对象176包括远程标志180,而缓冲对象178包括远程标志182。为了说明的目的,未设置远程标志180,而设置了远程标志182。系统100,例如,主处理器105,能够执行操作系统120和指令125,以执行本公开中描述的操作。

[0027] 接口电路115的示例包括,但不限于,系统总线和输入/输出(I/O)总线。接口电路115可以通过使用多种总线架构中的任何一种来实现。总线架构的示例可以包括但不限于增强型工业标准架构(EISA)总线,加速图形端口(AGP),视频电子标准协会(VESA)本地总线,通用串行总线(USB)和外围组件互连高速(PCIe)总线。主处理器105可以通过与用于耦接到硬件加速器135不同的接口电路被耦接到主存储器110。出于说明的目的,未示出接口电路115的端点,主处理器105通过该端点与其他设备通信。

[0028] 系统100还可以包括被耦接到接口电路115的一个或多个其他I/O设备(未示出)。I/O设备可以直接或通过中间I/O控制器耦接到系统100,例如接口电路115。I/O设备的示例包括,但不限于,键盘,显示设备,定点设备,一个或多个通信端口以及网络适配器。网络适配器是指使得系统100能够通过中间专用或公用网络而与其他系统,计算机系统,远程打印机和/或远程存储设备进行耦接的电路。调制解调器,电缆调制解调器,以太网卡和无线收发器是可以与系统100一起使用的不同类型的网络适配器的示例。

[0029] 在图1的示例中,硬件加速器135-1、135-2和135-3中的每一个分别耦接到存储器140-1、140-2和140-3。存储器140-1、140-2和140-3被实现为RAM,正如通常结合主存储器110所描述的。在一个或多个实施例中,每个硬件加速器135被实现为IC。该IC可以是可编程IC。可编程IC的一个示例是现场可编程门阵列(FPGA)。

[0030] 在图1的示例中,每个硬件加速器135包括端点145,链接电路150,存储器控制器(在图1中缩写为“MC”)155和互连电路168。每个硬件加速器135还包括一个或多个计算单元(缩写为图1中的“CU”)。计算单元是能够执行从主处理器105卸载的任务的电路。为了说明的目的,每个硬件加速器135被示出为包括计算单元160和计算单元165。应当理解,硬件加速器135可以包括比所示更少或更多的计算单元。

[0031] 在一个示例中,每个端点145被实现为PCIe端点。应当理解,端点145可以被实现为适合于在系统100所使用的特定类型的接口电路115或其实施例上进行通信的任何类型的端点。每个存储器控制器155耦接到相应的存储器140以便由硬件加速器135对该存储器140进行存取(例如,读出和写入)。

[0032] 在一个或多个实施例中,硬件加速器135-1和存储器140-1被附接到第一电路板(未示出),硬件加速器135-2和存储器140-2被附接到第二电路板(未示出),硬件加速器135-3和存储器140-3被附接到第三电路板(未示出)。这些电路板中的每一个可以包括用于耦接到总线端口或插槽的合适的连接器。例如,每个电路板可以具有被配置为用于插入系统100的可用PCIe插槽的连接器(或其他总线/接口连接器)。

[0033] 每个链接电路150能够与至少另一个链接电路(例如,相邻的链接电路150)一起建立加速器链接。如本文所使用的,“加速器链接”是指直接连接两个硬件加速器的通信链路。例如,具有硬件加速器135的每个电路板可以通过连接到链接电路150的电线耦接。链接电路150可以在电线上建立加速器链接。

[0034] 在特定实施例中,链接电路150通过环形拓扑进行通信链接。经由由链接电路150建立的加速器链接发送的数据是从左到右逐个进行,如方向箭头所示。例如,参考图1的示例,左侧的链接电路(例如,链接电路150-1)可以作为主设备,而右侧的相邻链接电路(例如,链接电路150-2)可以作为从设备。类似地,链接电路150-2可以用作为相对于链接电路150-3的主设备。链接电路150-3可以用作为相对于链接电路150-1的主设备。

[0035] 在一个或多个实施例中,每个链接电路150包括一个表或寄存器,该表或寄存器为每个硬件加速器(例如,在每个板上)指定了存储器140的数量(或大小)。使用该表,每个链接电路150能够修改事务中指定的地址,以使用加速器链接交换信息。在特定实施例中,表或寄存器是静态的。在一个或多个其他实施例中,驱动程序能够动态地,例如在运行时,读取和/或更新被存储在表或寄存器中的信息。

[0036] 出于说明的目的,描述了硬件加速器135-2的运行。应当理解,每个相应的硬件加速器中相同编号的组件能够以相同或相似的方式运行。因此,参考硬件加速器135-2,链接电路150-2能够从各种不同源或发起者中的任何一个接收事务,并将该事务路由到各种目标中的任何一个。例如,链接电路150-2能够接收来自端点145-2(例如,源自主处理器105)、计算单元160-2、计算单元165-2、经由链接电路150-1的硬件加速器135-1、或经由链接电路150-3的硬件加速器135-3的事务,流到链接电路150-1然后接着到链接电路150-2。链接电路150-2能够将事务路由到任何目标,例如端点145-2(例如,到主处理器105),计算单元160-2,计算单元165-2,存储器控制器155-2,经由链接电路150-3的硬件加速器135-1然后接着到链接电路150-1,或经由链接电路150-3的硬件加速器135-3,其中目标不同于源或发起者。

[0037] 例如,作为统一地址空间的一部分,主处理器105能够访问存储器140-1,存储器140-2和/或存储器140-3中的任何位置,然而,在访问这样的存储器时,主处理器105可以通过访问所选的硬件加速器(例如,硬件加速器135-2),然后通过使用加速器链接的选定硬件加速器,到达诸如存储器140-1,存储器140-2或存储器140-3的任何目标而做到这一点。

[0038] 作为说明性和非限制性示例,主处理器105可以发起涉及硬件加速器135-2和135-3的数据传输。硬件加速器135-2可以是发起者。在该示例中,主处理器105(例如,运行时间172和/或驱动程序174)创建对应于硬件加速器135-2的缓冲器对象176和对应与硬件加速器135-3的缓冲器对象178。主处理器105设置远程标志182,该远程标志182指示用于数据传输的目标地址(位于硬件加速器135-3中)相对于发起硬件加速器(硬件加速器135-2)是远程的。

[0039] 端点145-2能够经由接口电路115接收从主处理器105卸载的任务。在一个或多个实施例中,主处理器105通过执行运行时间172和驱动程序174,能够将硬件加速器135看作为统一地址空间。端点145-2可以将任务(例如,数据)提供给计算单元160-2。所述任务可以指定存储器140-3内的目标地址,计算单元160-2将从该目标地址中检索用于执行卸载任务的数据。硬件加速器135-2通过使用链接电路150-2,能够通过在链接电路150-2和链接电路150-3之间建立的加速器链接直接发起和执行与硬件加速器135-3的数据传输。

[0040] 虽然数据传输可以由主处理器105发起,但是数据传输是使用链接电路150执行的,并且不涉及主处理器105,主存储器110或接口电路115参与而发生。数据传输直接在硬件加速器之间发生。在常规系统中,数据传输将通过主处理器105经由接口电路115从硬件加速器135-3检索数据,然后通过接口电路115向硬件加速器135-2提供数据来进行。

[0041] 硬件加速器135在彼此之间读取和写入数据而不使数据通过主处理器105传输的能力显著减少了通过接口电路115(例如,PCIe总线)传递的数据量。这节省了接口电路115的相当大的带宽,用于在主处理器105和其他硬件加速器135之间传输数据。此外,由于减少了硬件加速器135共享数据所需的时间,因此可以提高系统100的运行速度。

[0042] 系统100可以包括比所示的更少的组件或图1中未示出的其他组件,取决于所实现的设备和/或系统的特定类型。另外,所包括的特定操作系统,应用程序和/或I/O设备可能会根据系统类型而有所不同。此外,一个或多个说明性组件可以合并到另一组件中,或者以其他方式形成另一组件的一部分。例如,处理器可以包括至少一些存储器。系统100可以用于实现单个计算机或多个联网或互连的计算机,每个计算机使用图1的架构或类似的架构来实现。

[0043] 图2示出了图1的硬件加速器135-2的示例性实现方案。在图2中,提供了链接电路150-2的示例性实现方案。应当理解的是,图2中针对链接电路150-2示出的架构可用于实现图1所示的任何链接电路150。

[0044] 在一个或多个实施例中,链接电路150-2能够将要发送到其他硬件加速器的事务转换为基于数据流的分组,并通过在链接电路150之间建立的加速器链接来路由分组。在特定的实施例中,链接电路150-2能够将符合AMBA可扩展接口(AXI)的内存映射事务转换为AXI数据流以进行传输。在本公开内,AXI被用作示例通信协议。应当理解,可以使用其他通信协议。在这方面,AXI的使用仅出于说明目的,而非限制。链接电路150-2还能够处理来自其他硬件加速器(例如,硬件加速器135-1和135-3)的输入分组,将分组转换成存储器映射的事务,以及在硬件加速器135-2内本地路由数据。此外,链接电路150-2能够将接收到的分组转换为存储器映射的事务,修改事务,将存储器映射的事务转换为分组,以及将分组传递给下一硬件加速器。经由加速器链接接收的数据可以作为存储器映射的事务在硬件加速器135-2内部进行路由。

[0045] 在图2的示例中,链接电路150-2包括收发器202和204,重发引擎(RTE)206和208,以及存储器映射到流(MM-流)映射器210和212。MM-流映射器210和212耦接到互连电路214。

[0046] 如图所示,收发器202可以耦接到硬件加速器135-1中的对应收发器,而收发器204可以耦接到硬件加速器135-3中的对应收发器。收发器202和204实现与其他硬件加速器建立的加速器链接的物理层。收发器202和204中的每一个能够为多千兆位通信链路实现轻量级的串行通信协议。在一个或多个实施例中,收发器202和204中的每一个能够实现到相邻IC中的收发器的双向接口。收发器202和204能够自动初始化与其他硬件加速器的加速器链接。通常,收发器202和204能够进行双向通信以实现与流控制有关的低级信令和低PHY级协议。然而,数据流是使用环形拓扑来实现的,并且,如前所述,是从主到从机流动(例如,围绕环的单个方向)。

[0047] 例如,收发器202能够与硬件加速器135-1的链接电路150-1内的对应收发器双向通信。收发器204能够与硬件加速器135-3的链接电路150-3内的对应收发器双向通信。收发器202和204中的每一个能够使用例如AXI数据流的数据流与相邻的收发器通信。

[0048] 在具体的实施例中,收发器202和204能够使用8B/10B编码规则向相邻的硬件加速器发送和接收数据。收发器202和204中的每一个能够使用8B/10B编码规则来检测单比特和大多数多比特错误。

[0049] 在一个或多个实施例中,收发器202和204中的每一个被实现为Aurora 8B/10B IP核,其可从加利福尼亚州圣何塞的Xilinx公司获得。然而,应当看到,所指出的特定核心是出于说明的目的而提供的,并不旨在作为限制。可以使用能够按照本文所述进行操作的其他收发器。

[0050] 收发器202耦接到RTE 206。收发器202和RTE 206能够通过在每个方向上运行的多个数据流进行通信,从而支持双向通信。收发器204耦接到RTE208。收发器204和RTE208能够通过在每个方向上运行的多个数据流进行通信,以支持双向通信。

[0051] RTE 206和208能够管理事务(transaction)。在一个或多个实施例中,RTE 206和RTE 208每个在分别由收发器202和204实现的那些之上实现通信协议的附加层。例如,RTE 206和RTE 208各自实现事务层(TL)/链路层(LL)和用户层。这些附加层为数据完整性提供了额外的保证。初始化后,应用程序可以跨加速器链接将数据作为数据流传递。附加的数据完整性措施特别有益,因为在将内存映射的事务转换为流数据时,控制信号会与数据合并。数据完整性问题可能会导致控制信号损坏。片上互连和/或总线是相对于控制信号不容许数据丢失的。

[0052] TL/LL实现了基于令牌的流控制,以确保无损数据通信。在一个或多个实施例中,相邻收发器之间以及收发器与RTE之间的通信信道的宽度为128位。在发送数据时,每个RTE能够在由收发器实现的、将事务实际发送到物理层之前,检查目标硬件加速器中的接收链接电路是否具有足够的缓冲资源(例如令牌token)来接收要发送的整个事务。例如,在将数据提供给收发器202(在链接电路150-2内)以进行发送之前,RTE 206可以检查硬件加速器135-1中的接收链接电路150-1具有足够的缓冲器资源来接收数据。

[0053] RTE 206和208能够检测数据损坏。例如,RTE 206和208中的每一个能够针对接收到的每个分组验证分组长度信息,分组序列信息和/或循环冗余校验(CRC)校验和。当RTE从设备(例如,接收RTE)检测到数据包错误时,RTE可能会进入错误中止模式。在错误中止模式下,RTE会将带有错误的数据包作为失败数据包丢弃。RTE进一步丢弃该事务的所有后续数据包。在特定实施例中,错误中止模式的启动使RTE启动链路重试序列。一旦链路重试序列成功,则链路主(例如,发送RTE)能够通过从故障点开始重新恢复传输。

[0054] RTE 206耦接到MM流映射器210。RTE 206能够经由在每个方向上运行的多个数据流与MM流映射器210通信,以支持双向通信。RTE 208耦接到MM流映射器212。RTE208能够经由在每个方向上运行的多个数据流与MM流映射器212通信,以支持双向通信。

[0055] MM流映射器210和MM流映射器212中的每一个都耦接到互连电路214。互连电路214能够在MM流映射器210和212以及与其耦接的硬件加速器135-2的其他主电路和/或从属电路之间路由数据。互连电路214可以被实现为一个或多个片上互连。片上互连的一个示例是AXI总线。AXI总线是嵌入式微控制器总线接口,用于在电路块和/或系统之间建立片上连接。互连电路的其他示例实施方式可以包括,但不限于,其他总线、交叉开关、芯片上网络(NoC)等。

[0056] MM流映射器210和212能够分别将从RTE 206和208接收的数据流转换成可以提供给互连电路块214的存储器映射的事务。就这一点而言,数据流可以被多路分解为支持内存映射的事务。MM流映射器210和212还能够将从互连电路块214接收的存储器映射的事务转换为可以分别提供给RTE 206和208的流数据。MM流映射器210和212能够将支持存储器映射的事务(例如,包括所讨论的控制信号)的多个信道复用为单个数据流,以分别发送到RTE 206和208。

[0057] 在一个或多个实施例中,MM流映射器210和212中的每一个能够调整在事务中接收的目标地址。MM流映射器210,例如,在经由加速器链接从硬件加速器135-1接收事务时,可

以从事务的目标地址中减去硬件加速器135-2的地址范围的上限(例如,存储器140-2的地址范围)。通过在事务处理通过链接电路150时调整目标地址,事务可以经由加速器链接从一个硬件加速器被引导到另一硬件加速器。与使用加速器链接中的地址操作有关的更多细节将结合图4更详细地描述。

[0058] 出于说明的目的,硬件加速器135-2的其他部分将相对于链接电路150-2进行描述。在图2的示例中,互连电路214耦接到直接存储器访问(DMA)主电路216。DMA主电路216例如包括用于与互连电路块214通信的存储器映射接口。DMA主电路216耦接到PCIe端点218。PCIe端点218是图1的端点145-2的示例实现,它通信地链接到主处理器105。

[0059] 在图2的的示例中,互连电路214也耦接到一个或多个计算单元主(compute unit master)220-1至220-N。每个计算单元主220在硬件加速器135-2内实现的计算单元与互连电路块214之间提供双向接口。每个计算单元主220还包括用于与互连电路块214进行通信的存储器映射接口。每个计算单元160-2和计算单元165-2可以通过从属接口(未示出)连接到互连电路214。

[0060] 在图2的的示例中,互连电路214也耦接到一个或多个存储器控制器从属电路225-1至225-N。每个存储器控制器从属电路225促进存储器140-2的读取和写入操作。存储器140-2可以被实现为可由硬件加速器135-2访问的一个或多个片外存储器。存储器控制器225-1至225-N中的每一个还包括用于与互连电路块214进行通信的存储器映射接口。

[0061] 图3示出了RTE 206的示例性实施方式。结合图3描述的示例性架构使用流控制单元(FLIT)实现基于信用的流控制/重传控制方案。RTE 206能够在内部使用的基于FLIT的协议和/或接口之间转换为可被应用程序使用的协议和/或接口。

[0062] RTE 206包括发送信道330。发送信道330能够将数据(例如,AXI)流解封装为基于FLIT的事务。在图3的示例中,发送信道330包括发送(TX)分组循环冗余校验(CRC)生成器302,重试指针返回命令(PRET)分组/初始重试命令(IRTRY)分组生成器和返回重试指针(RRP)嵌入器304,令牌返回(TRET)数据包生成器和序列(SEQ)编号/前向重试指针(FRP)/返回令牌计数(RTC)嵌入器306,流控制电路308和输出缓冲器310。TRET生成器和SEQ/FRP/RTC嵌入器306还耦接到重试缓冲器312。

[0063] RTE 206包括接收信道340。接收信道340能够封装基于FLIT的接口并将该接口转换成数据(例如,AXI)流。在图3的示例中,接收信道340包括分组边界检测器316,接收(RX)分组CRC电路318,RX分组处理器320和输入缓冲器322。Rx分组处理器320耦接到错误处理器324和重试序列电路314。

[0064] 提供RTE 206是为了说明而不是限制。应当理解,可以使用适合于实现基于信用的流量控制/重传控制方案的其他架构。结合图3描述的架构还可以被使用于实现图2的RTE208,在数据流方面具有翻转或反转的方向。

[0065] 图4示出了用于具有多个硬件加速器的系统的示例性操作方法400。方法400示出了在硬件加速器之间直接进行数据传输的示例。方法400可以由与结合图1描述的系统100相同或相似的系统执行。方法400示出了如何减轻耦接主处理器和硬件加速器的总线上的不足带宽。否则会在总线上发生的数据传输可能会被转移到加速器链接,从而释放总线上的带宽以进行其他操作。

[0066] 在方框405,系统能够自动发现硬件加速器序列。在一个或多个实施例中,硬件加

速器,例如硬件加速器的板,被布置在系统内的环形拓扑中。主处理器知道现有的PCIe拓扑,因此知道存在于耦接到PCIe总线的系统中的硬件加速器的数量。此外,主处理器,例如通过运行时间,知道加载到每个硬件加速器中的特定电路(例如,图像或配置比特流)。这样,主处理器知道硬件加速器支持本文所述的加速器链接。主处理器仍然必须确定硬件加速器的顺序。该驱动程序例如能够执行所描述的硬件加速器序列的自动发现。这种自动发现能力支持在系统中添加新的和/或附加的硬件加速器,而不必修改主处理器执行的应用程序。

[0067] 每个硬件加速器可能具有已知且相同的地址范围。例如,可以假定每个硬件加速器具有对应于存储器140的16GB的16GB的地址范围。在一个或多个实施例中,主处理器能够以16GB的间隔将独特值写入存储器地址。然后,主处理器可以回读这些值,以基于写入和读取的值来确定环形拓扑内的硬件加速器的顺序。

[0068] 在方框410中,主处理器能够在启动时在每个硬件加速器上创建缓冲器。例如,由主处理器执行的运行时间能够与每个硬件加速器通信以在每个相应的硬件加速器的存储器内创建缓冲器。参考图1,硬件加速器135-1在存储器140-1内创建缓冲器。硬件加速器135-2在存储器140-2内创建缓冲器。硬件加速器135-3在存储器140-3内创建缓冲器。

[0069] 在方框415中,主处理器发起硬件加速器之间的数据传输。例如,数据传输可能是要从主处理器卸载到硬件加速器的任务的一部分。作为说明性和非限制性示例,主处理器105可以将针对应用的任务卸载到硬件加速器135-1的计算单元160-1。任务可以包括指令和目标地址,计算单元160-1将从目标地址获得任务的数据。在该示例中,目标地址位于硬件加速器135-2中(例如,在存储器140-2中)。因此,为了执行从主处理器卸载的任务,计算单元160-1必须从存储器140-2中的目标地址检索数据。

[0070] 在方框420中,运行时间可以请求硬件加速器135-1和135-2之间的数据传输。例如,运行时间可以请求由硬件加速器135-1,或从硬件加速器135-1,读取硬件加速器135-2。

[0071] 在方框425中,驱动程序能够在与硬件加速器135-2相对应的主存储器中创建缓冲对象,并且在与硬件加速器135-1相对应的主存储器中创建缓冲对象。缓冲对象是在主存储器中实现的影子数据结构。每个缓冲对象可以对应于或代表系统中的设备。缓冲对象可以包括支持由主处理器执行的运行时间所执行的管理功能的数据。

[0072] 在一个或多个实施例中,在主存储器中创建的缓冲对象可以包括远程标志。从发起事务的硬件加速器的角度来看,可以设置远程标志,以指示缓冲对象是远程的。在这个示例中,硬件加速器135-1正在从硬件加速器135-2读取数据。这样,硬件加速器135-1正在发起事务。驱动程序在创建时将远程标志设置在与硬件加速器135-2相对应的缓冲对象中。

[0073] 在方框430中,运行时间库通过发起硬件加速器来发起对缓冲对象(例如,远程缓冲对象)的访问。运行时间库从硬件加速器135-1启动对与硬件加速器135-2相对应的缓冲对象的访问。例如,运行时间确定在硬件加速器135-2的缓冲对象内设置了远程标志。响应于确定设置了远程标志,运行时间库使用链接电路建立的加速器链接来调度传输。在使用硬件加速器之间的加速器链接来调度传输时,运行时间确定要由硬件加速器135-1用于访问来自硬件加速器135-2的数据的地址。

[0074] 为了便于说明,考虑一个示例,其中每个硬件加速器135的地址范围为1-1000。在这样的示例中,运行时间可以确定要由硬件加速器135-1从硬件加速器135-2检索的数据位

于与硬件加速器135-2相对应的地址500处的缓冲器中(例如,在对应于存储器140-2的地址500处)。在此示例中,运行时间将1000添加到目标地址,从而得到地址1500,该地址提供给硬件加速器135-1作为目标地址,用于读取对其进行卸载任务的数据。

[0075] 作为另一示例,如果数据被存储在存储器140-3内的地址500处,则运行时间将增加2000,假设每个硬件加速器135具有1-1000的地址范围,以便使事务到达硬件加速器135-3。通常,众所周知,可以通过所使用的片上总线互连(例如,AXI互连)来跟踪返回路径数据。例如,当从主设备发出读取请求时,在读取请求遍历每个硬件加速器时,通过一系列地址解码和/或地址移位(由MM流映射器执行),将读取请求通过互连而路由到从属设备每个硬件加速器。每个单独的互连都能够跟踪对每个从属机还有未完成的事务的那些主。在返回读取数据后,读取的数据可以通过正确的接口被发送回去。在某些情况下,标识符(ID)比特可用于将特定的读取数据与特定的主关联起来,以便返回读取的数据。

[0076] 在方框435中,进行发起的硬件加速器(例如,第一硬件加速器)从主处理器接收任务。端点145-1例如可以接收任务并将任务提供给计算单元160-1。该任务指定要由计算单元160-1操作的数据放置在目标地址,在此示例中该目标地址为1500。计算单元160-1例如可以具有控制端口,目标地址可以存储到该控制端口。在尝试访问位于地址1500处的数据时,计算单元160-1识别出该地址不在硬件加速器135-1的范围内。例如,计算单元160-1能够将地址与地址范围1000的上限进行比较,并确定该地址超过上限。在此示例中,计算单元160-1能够发起从地址1500读取事务。例如,计算单元160-1可以发起读取事务作为通过互连214发送的内存映射事务。

[0077] 在方框440中,进行发起的硬件加速器通过加速器链接访问目标硬件加速器(例如,第二硬件加速器)。例如,链接电路150-1能够将由计算单元160-1发起的存储器映射的事务转换成基于流的分组(例如,使用MM流映射器)。链接电路150-1还能够使用支持数据完整性检查,重传,初始化和错误报告(例如,使用RPE)的附加数据来对分组进行编码。环形拓扑可以从左到右进行。这样,分组可以由链接电路150-1的收发器输出到链接电路150-2。

[0078] 链接电路150-2在收发器202中接收数据流,并在RTE 206中处理事务。MM流映射器210响应于接收到基于流数据的分组,能够执行各种操作。MM流映射器210例如能够将基于流的分组转换为存储器映射的事务。此外,MM流映射器210能够递减目标地址1500到硬件加速器135-2的地址范围的上限。如上所述,上限可以存储在链接电路150-2内的表或寄存器中,例如在MM流映射器210中。在该示例中,MM流映射器210将目标地址1500减1000,从而导致目标地址为500。由于目标地址对于硬件加速器135-2是本地的,因此硬件加速器135-2可以对接收到的事务进行操作。在该示例中,MM流映射器210将存储器映射的事务提供给互连214。可以将存储器映射的事务提供给存储器控制器155-2(例如,通过存储器控制器从属)以执行读取事务。以这种方式,硬件加速器135-1能够从硬件加速器135-2读取数据(或向其写入数据)。可以使用用于发送读取请求的相同路径将请求的数据从存储器140-2提供回请求者。例如,从存储器140-2读取的数据从硬件加速器135-2发送到硬件加速器135-1,而不必通过环形拓扑向前遍历到硬件加速器135-3然后到硬件加速器135-1。

[0079] 例如,如果目标地址是2500,则减量的结果将是1500。在那种情况下,MM流映射器210确定目标地址不在硬件加速器135-2中,因为目标地址大于用于硬件加速器135-2的地址范围(例如1000)的上限。在那种情况下,MM流映射器210可以通过互连电路将事务发送到

MM流映射器212,以转发到下一个硬件加速器。

[0080] 在方框445中,硬件加速器135-1中的计算单元160-1能够向主处理器生成中断,以通知主处理器硬件加速器之间的数据传输已完成。在方框450中,运行时间能够向应用提供数据传输完成的任何通知。例如,运行时间能够处理完成事件、命令队列、和对应用程序的通知。

[0081] 在一个或多个实施例中,PCIe端点和DMA主控能够写入位于不同硬件加速器中的目标地址。作为说明性且非限制性的示例,主处理器可以使用位于硬件加速器135-2中的目标地址将数据发送到硬件加速器135-1。在那种情况下,DMA主设备能够识别目标地址位于不同的硬件加速器中,并调度通过加速器链接进行的数据传输。例如,DMA主设备可以将目标地址与硬件加速器135-1的地址范围的上限进行比较。响应于确定目标地址超过上限,DMA主设备能够通过互连电路向链接电路150-1中的MM流映射器212发起存储器映射的事务,以经由加速器链接发送到硬件加速器135-2。

[0082] 在一个或多个实施例中,主处理器能够使用加速器链接来实现负载平衡。例如,主处理器能够使用运行时间来确定要向其提供数据或要卸载任务的所选硬件加速器中的DMA通道(例如,DMA主设备)的状态。响应于确定DMA主设备正忙或正以大于活动的阈值量运行,主处理器可以通过总线将数据发送到其他硬件加速器。数据可以指定所选硬件加速器内的目标地址。接收硬件加速器中的DMA主设备从主处理器接收到数据后,便能够通过加速器链接将数据转发到选定的硬件加速器。在特定实施例中,主处理器能够基于对其中的DMA主控器不忙或正在低于活动阈值量的确定来选择接收硬件加速器。

[0083] 出于说明的目的,从硬件加速器135-1到硬件加速器135-3的写事务的示例通常被描述为由主处理器发起。主处理器通过运行时间和驱动程序设置目标硬件加速器的远程标志,并确定地址2500(使用先前示例,其中所需地址位于硬件加速器135-3中的地址500处)。主处理器向硬件加速器135-1提供指令以写入地址2500。在硬件加速器135-1中,地址为2500的事务被呈现给互连214。由于该地址超出了硬件加速器135-1的上限,然后,互连214将事务发送到链接电路150-1。链接电路150-1将事务发送到链接电路150-2。硬件加速器135-2中的MM流映射器将地址递减1000,导致新地址1500。由于1500超过硬件加速器135-2的上限地址,因此新地址仍处于远程状态。这样,事务被转发到硬件加速器135-3。

[0084] 硬件加速器135-3中的MM流映射器使地址递减,得到新的地址500。然后,通过硬件加速器135-3中的互连214将事务提供给存储器控制器,并将数据写入存储器140-3。在描述的示例中,每个硬件加速器使用该地址来确定事务是否可以由硬件加速器服务,如果可以,则确定在内部将事务路由到何处(例如,到存储控制器或其他电路块),或者应该转发给下一个硬件加速器。在特定实施例中,该地址不同于将数据写入存储器中的实际地址。如上所述,将写确认从硬件加速器135-3通过硬件加速器135-2发送到硬件加速器135-1。

[0085] 为了说明的目的,通常将由硬件加速器135-1发起的、到硬件加速器135-3读取事务的另一示例描述为由主处理器发起。主处理器通过运行时间和驱动程序设置目标硬件加速器的远程标志,并确定地址2500(使用先前示例,其中所需地址是位于硬件加速器135-3中的地址500)。主处理器向硬件加速器135-1提供指令以从地址2500读取。在硬件加速器135-1中,地址为2500的事务被呈现给互连214。由于该地址超出了硬件加速器135-1的上限,然后,互连214将事务发送到链接电路150-1。链接电路150-1将事务发送到链接电路

150-2。硬件加速器135-2中的MM流映射器将地址递减1000,导致新地址1500。由于1500超过硬件加速器135-2的上限地址,因此新地址仍处于远程状态。这样,事务被转发到硬件加速器135-3。

[0086] 硬件加速器135-3中的MM流映射器使地址递减,得到新的地址500。然后,通过硬件加速器135-3中的互连214将事务提供给存储器控制器,并从存储器140-3中读取数据。在描述的示例中,该地址被每个硬件加速器使用来确定事务是否可以由硬件加速器提供服务,如果可以,则确定在何处内部地路由事务,或者应该转发给下一个硬件加速器。在特定实施例中,该地址不同于从存储器中读取数据的实际地址。如上所述,读取的数据被从硬件加速器135-3通过硬件加速器135-2发送到硬件加速器135-1。

[0087] 图5示出了包括硬件加速器和一个或多个附加设备的系统的示例。在图5的示例中,示出了硬件加速器135-1和135-2,并且使用每个相应的硬件加速器中的链接电路通过加速器链路来耦接。为了说明的目的,未示出硬件加速器135-3。该系统还包括耦接到存储器520的GPU 515和I/O设备525。

[0088] 在图5的示例中,GPU 515可将数据写入硬件加速器135-2或从硬件加速器135-2读取数据。在该示例中,主处理器(未示出)提供句柄505-N给GPU 515。在特定实施例中,句柄可以被实现为文件描述符。句柄505-N可以指向对应于硬件加速器135-2的缓冲器对象510-N。通过GPU 515将句柄505-N用于读或写操作,主处理器对对应于句柄505-N的缓冲对象(例如,缓冲对象510-N)发起动作。主处理器确定缓冲对象510-N是本地的还是远程的。由于未设置缓冲对象510-N中的远程标志,因此主处理器可以通过PCIe从存储器140-2检索数据并且通过PCIe将数据提供给GPU 515。

[0089] 在一个或多个其他实施例中,主处理器可以通过访问不同的硬件加速器来发起从存储器140-2检索数据。例如,主处理器可以经由PCIe启动与硬件加速器135-1的通信,以从存储器140-2检索数据。在那种情况下,硬件加速器135-1可以使用链接电路直接与硬件加速器135-2通信以从存储器140-2检索数据。硬件加速器135-1然后可以将数据提供回主处理器,该主处理器继而通过PCIe将数据提供给GPU 515。

[0090] 在另一个例子中,I/O设备525,例如照相机,可以将数据写入硬件加速器135-1。在那种情况下,主处理器能够将句柄505-1提供给I/O设备525。句柄505-1可以指向对应于硬件加速器135-1的缓冲器对象510-1。通过I/O设备525将句柄505-1用于写操作,主处理器对对应于句柄505-1的缓冲对象(例如,缓冲对象510-1)发起动作。主处理器确定缓冲器对象510-1是本地的还是远程的。主处理器可以从I/O设备525接收数据,并通过PCIe将此类数据提供给硬件加速器135-1,以便写入存储器140-1和/或进一步处理,因为缓冲区对象510-1中的远程标志还未设置。

[0091] 在一个或多个实施例中,仅在能够使用加速器链接的硬件加速器之间进行数据传输的情况下,驱动程序才能够在缓冲对象内设置远程标志。图5显示,虽然其他类型的设备可结合硬件加速器一起使用,但是在这样的其他设备与硬件加速器之间的数据传输是通过总线进行的,并且涉及到主处理器。

[0092] 图6示出了用于IC的示例架构600。一方面,架构600可以在可编程IC内实现。例如,架构600可以用于实现现场可编程门阵列(FPGA)。架构600也可以代表IC的片上系统(SOC)类型。SOC是包括执行程序代码的处理器和一个或多个其他电路的IC。其他电路可以被实现

为硬连线电路,可编程电路和/或其组合。电路可以彼此协作和/或与处理器协作。

[0093] 架构600包括几种不同类型的可编程电路,例如逻辑块。例如,架构600可以包括大量不同的可编程块,包括多千兆位收发器(MGT)601,可配置逻辑块(CLB)602,随机存取存储器块(BRAM)603,输入/输出块(IOB)604,配置和时钟逻辑(CONFIG/CLOCKS)605,数字信号处理模块(DSP)606,专用I/O模块607(例如,配置端口和时钟端口)以及其他可编程逻辑608,诸如数字时钟管理器,模拟到-数字转换器,系统监视逻辑等。

[0094] 在一些IC中,每个可编程区块包括可编程互连元件(INT)611,该可编程互连元件(INT)611具有与每个相邻区块中的对应INT 611之间的标准连接。因此,INT611一起实现了所示IC的可编程互连结构。每个INT 611还包括与在同一贴片内的可编程逻辑元件之间的连接,如在图6顶部所包括的例子所示。

[0095] 例如,CLB 602可以包括可配置逻辑元件(CLE)612,其可以被编程为实现用户逻辑加上单个INT 611。BRAM 603除了一个或多个INT 611之外还可以包括BRAM逻辑元件(BRL)613。通常,贴片中包含的INT 611的数量取决于贴片的高度。如图所示,BRAM贴片具有与五个CLB相同的高度,但是也可以使用其他数目(例如四个)。除适当数量的INT 611外,DSP贴片606还可以包括DSP逻辑元素(DSPL)614。IOB 604除了包括INT 611的一个实例之外,还可以包括I/O逻辑元素(IOL)615的两个实例。连接到IOL 615的实际I/O焊盘可以不限于IOL 615的区域。

[0096] 在图6的示例中,例如由区域605、607和608形成的管芯中心附近的柱状区域可以用于配置,时钟和其他控制逻辑。从该列延伸的水平区域609可以用于在可编程IC的宽度上分布时钟和配置信号。

[0097] 一些利用图6所示架构的IC包括附加的逻辑块,这些逻辑块破坏了构成IC大部分的规则柱状结构。附加逻辑块可以是可编程块和/或专用电路。例如,被描绘为PROC610的处理器块跨越CLB和BRAM的几列。

[0098] 一方面,PROC 610可以被实现为专用电路,例如被实现为硬连线处理器,它们被制造为实现IC的可编程电路的管芯的一部分。PROC 610可以代表复杂程度不同的各种不同处理器类型和/或系统中的任何一种,其复杂范围从单个处理器(例如,能够执行程序代码的单个内核)到具有一个或多个内核,模块,协同工作的处理器,接口等的整个处理器系统。

[0099] 另一方面,PROC 610可以从体系结构600中被省略,并且用所描述的可编程块的一个或多个其他种类的替代。此外,此类块可用于形成“软处理器”,因为可编程电路的各个块可用于形成可像PROC 610那样执行程序代码的处理器。

[0100] 短语“可编程电路”是指IC内的可编程电路元件,例如本文所述的各种可编程或可配置电路块或贴片,以及根据加载到IC的配置数据,选择地耦接各种电路块、贴片、和/或元件的互连电路。例如,图6中所示的、在PROC 610外部的电路块诸如CLB 602和BRAM 603被认为是IC的可编程电路。

[0101] 通常,只有在将配置数据加载到IC之后,才能确定可编程电路的功能。一组配置位可用于对IC(例如FPGA)的可编程电路进行编程。一个或多个配置比特通常被称为“配置比特流”。通常,在不首先将配置比特流加载到IC中的情况下,可编程电路将无法工作或起作用。配置比特流有效地实现了可编程电路系统中的特定电路设计。电路设计规定了例如可编程电路块的功能方面以及各种可编程电路块之间的物理连接性。

[0102] “硬连线的”或“硬化的”(即不可编程)电路是IC的一部分。与可编程电路不同,硬连线电路或电路块不是在制造IC之后通过加载配置比特流才被实施的。硬连线电路通常被认为具有例如专用电路块和互连,这些电路块和互连在不首先将配置比特流加载到IC(例如PROC 610)的情况下即可工作。

[0103] 在某些情况下,硬连线电路可能具有一种或多种操作模式,可以根据寄存器设置或被存储在IC中一个或多个存储元件中的值来设置或选择一种或多种操作模式。例如,可以通过将配置比特流加载到IC中来设置操作模式。尽管具有这种能力,但是硬连线电路不被认为是可编程电路,因为当硬连线电路被制造为IC的一部分时,该硬连线电路就是可操作的并且具有特定功能。

[0104] 在SOC的情况下,配置比特流可以指定将在可编程电路系统内实现的电路系统以及将由PROC 610或软处理器执行的程序代码。在某些情况下,架构600包括专用配置处理器,其将配置比特流加载到适当的配置存储器和/或处理器存储器。专用配置处理器不执行用户指定的程序代码。在其他情况下,架构600可以利用PROC 610来接收配置比特流;将配置比特流加载到适当的配置存储器中;和/或提取程序代码用于执行。

[0105] 图6旨在示出可用于实现包括可编程电路例如可编程结构的IC的示例架构。例如,在图6的顶部所包括的一列中逻辑块的数量,列的相对宽度,列的数量和顺序,列中包含的逻辑块的类型,逻辑块的相对大小以及互连/逻辑实现方案,纯粹是说明性的。例如,在实际的IC中,无论CLB出现在何处,通常都包括一个以上的相邻列的CLB,以促进用户电路设计的有效实施。但是,相邻CLB列的数量可能会随IC的整体尺寸而变化。此外,IC内诸如PROC 610之类的块的尺寸和/或位置仅出于说明的目的,而无意作为限制。

[0106] 架构600可用于实现本文所述的硬件加速器。在特定实施例中,端点,链接电路和存储器控制器中的一个或多个或每一个可以被实现为硬连线电路块。在特定实施例中,可以使用可编程电路来实现端点,链接电路和存储器控制器中的一个或多个或每个。在其他实施例中,一个或多个所述电路块可以被实现为硬连线电路块,而其他电路块可以通过利用可编程电路来实现。

[0107] 本公开内容中描述的实施例可被使用于多种应用中的任何一种,例如,数据库加速,处理多个视频流,实时网络流量监视,机器学习或可能涉及多个硬件加速器的任何其他应用。

[0108] 出于说明的目的,阐述了特定的术语以提供对本文公开的各种发明概念的透彻理解。然而,本文所使用的术语仅出于描述本发明布置的特定方面的目的,而并非旨在进行限制。

[0109] 如本文所定义,单数形式的“一”,“一个”和“该”也意图包括复数形式,除非上下文另有明确说明。

[0110] 如本文所定义,术语“大约”是指几乎正确或精确,值或量接近但不精确。例如,术语“大约”可以表示所列举的特性、参数或值是在准确的特性、参数或值的预定量之内。

[0111] 如本文所定义,术语“至少一个”,“一个或多个”和“和/或”是开放式表达,即它们在操作上是结合的和分离的,除非另有明确说明。例如,每个表达语句“A,B和C中的至少一个”,“A,B或C中的至少一个”,“A,B和C中的一个或多个”,“A,B或C中的一个或多个”,以及“A,B和/或C”表示单独A,单独B,单独C,A和B一起,A和C一起,B和C一起,或A,B和C一起。

[0112] 如本文所定义,术语“自动”是指没有用户干预。如本文所定义,术语“用户”是指人类。

[0113] 如本文所定义,术语“计算机可读存储介质”是指包含或存储供指令执行系统,装置或设备使用或与其结合使用的程序代码的存储介质。如本文所定义,“计算机可读存储介质”本身不是瞬变的传播信号。计算机可读存储介质可以是但不限于电子存储设备,磁存储设备,光存储设备,电磁存储设备,半导体存储设备或前述的任何合适的组合。如本文所述,各种形式的存储器是计算机可读存储介质的示例。计算机可读存储介质的更具体示例的非详尽列表可以包括:便携式计算机磁盘,硬盘,RAM,只读存储器(ROM),可擦可编程只读存储器(EPROM或闪存)存储器),可电子擦除的可编程只读存储器(EEPROM),静态随机存取存储器(SRAM),便携式光盘只读存储器(CD-ROM),数字多功能磁盘(DVD),软盘等。

[0114] 如本文所定义,术语“如果”是指“在…时”或“在……上”或“响应于”或“响应于”,取决于上下文。因此,短语“如果确定”或“如果检测到[所陈述的条件或事件]”可以被解释为表示“在确定时”或“响应于确定”或“在检测到[所陈述的条件或事件]时”,或“响应于检测到[陈述的条件或事件]”,或“响应于检测到[陈述的条件或事件]”,取决于上下文。

[0115] 如本文所定义,术语“响应于”和如上所述的类似语言,例如“如果”,“在……时”或“在……之上”,是指容易对动作或事件做出响应或做出反应。响应或反应是自动执行的。因此,如果“响应”第一动作而执行第二动作,则在第一动作的发生与第二动作的发生之间存在因果关系。术语“响应于”表示因果关系。

[0116] 如本文所定义,术语“一个实施例”,“一个实施例”,“一个或多个实施例”,“特定实施例”或类似语言是指结合该实施例描述的特定特征,结构或特性被包括在本公开内描述的至少一个实施例中。因此,在整个本公开内容中,短语“在一个实施例中”,“在一个实施例中”,“在一个或多个实施例中”,“在特定实施例中”和类似语言的出现可以但不一定全部指同一个实施例。在本公开内容中,术语“实施例”和“布置”可互换使用。

[0117] 如本文所定义,术语“处理器”是指至少一个硬件电路。硬件电路可以被配置为执行程序代码中包含的指令。硬件电路可以是集成电路。处理器的示例包括但不限于中央处理器(CPU),阵列处理器,矢量处理器,数字信号处理器(DSP),FPGA,可编程逻辑阵列(PLA),ASIC,可编程逻辑电路和控制器。

[0118] 如本文所定义,术语“输出”是指存储到物理存储单元,例如设备;写入显示器或其他外围输出设备;发送到另一系统;导出等。

[0119] 如本文中所定义,术语“实时”是指用户或系统对于要进行的特定过程或确定而言感测为足够即时的处理响应度,或者使处理器能够跟上某些外部过程。

[0120] 如本文所定义,术语“基本上”是指不需要精确地实现所列举的特性,参数或值,而是有偏差或变化,包括例如公差,测量误差,测量精度限制和那些已知的其他因素。本领域技术人员可以不排除该特性旨在提供的效果的量发生。

[0121] 此处可以使用术语第一,第二等来描述各种要素。这些要素不应受这些术语的限制,因为除非另有说明或上下文另有明确说明,否则这些术语仅用于将一个要素与另一个要素区分开。

[0122] 计算机程序产品可以包括其上具有计算机可读程序指令的计算机可读存储介质,该计算机可读程序指令用于使处理器执行本文所述的发明布置的各方面。在本公开内容

中,术语“程序代码”与术语“计算机可读程序指令”可互换使用。此处描述的计算机可读程序指令可以从计算机可读存储介质下载到相应的计算/处理设备,或者通过网络(例如,互联网,LAN,WAN和/或无线网)下载到外部计算机或外部存储设备。该网络可以包括铜传输电缆,光传输纤维,无线传输,路由器,防火墙,交换机,网关计算机和/或包括边缘服务器的边缘设备。每个计算/处理设备中的网络适配器卡或网络接口从网络接收计算机可读程序指令,并转发计算机可读程序指令以存储在相应的计算/处理设备内的计算机可读存储介质中。

[0123] 用于执行本文所述的发明布置的操作的计算机可读程序指令可以是汇编程序指令,指令组结构(ISA)指令,机器指令,机器相关指令,微代码,固件指令或写入的源代码或目标代码的一种或多种编程语言的任何组合,包括面向对象的编程语言和/或过程编程语言。计算机可读程序指令可以包括状态设置数据。计算机可读程序指令可以完全在用户计算机上,部分在用户计算机上,作为独立软件包执行,部分在用户计算机上并且部分在远程计算机上或完全在远程计算机或服务器上执行。在后一种情况下,远程计算机可以通过任何类型的网络(包括LAN或WAN)连接到用户的计算机,或者可以与外部计算机建立连接(例如,使用互联网服务提供商通过互联网进行连接)。在某些情况下,包括例如可编程逻辑电路,FPGA或PLA的电子电路可以通过利用计算机可读程序指令的状态信息来个性化电子电路,执行计算机可读程序指令,从而执行以下描述的发明安排方面的工作。

[0124] 在此参考方法,装置(系统)和计算机程序产品的流程图和/或框图描述了本发明装置的某些方面。将会看到,流程图图示和/或框图的每个框以及流程图图示和/或框图中的框的组合可以由计算机可读程序指令,例如程序代码,来实现。

[0125] 可以将这些计算机可读程序指令提供给通用计算机,专用计算机或其他可编程数据处理装置的处理器,以产生机器,从而使得该指令经由计算机的处理器或其他可编程数据处理装置在该装置中执行,创建用于实现流程图和/或框图方框中指定的功能/动作的装置。这些计算机可读程序指令还可以被存储在计算机可读存储介质中,该计算机可读存储介质可以指导计算机,可编程数据处理装置和/或其他设备以特定方式起作用,从而使得其中存储有指令的计算机可读存储介质包括制造的制品,该制品包括实现在流程图和/或方框图或多个方框中指定的操作的各方面的指令。

[0126] 计算机可读程序指令也可以被加载到计算机,其他可编程数据处理设备,或其他设备上,以使在计算机,其他可编程设备或其他设备上执行一系列操作以产生计算机实现的过程,从而在计算机,其他可编程装置,或其他设备上执行的指令实现了流程图和/或框图方框指定的功能/动作。

[0127] 附图中的流程图和框图示出了根据本发明布置的各个方面的系统,方法和计算机程序产品的可能实现的体系结构,功能和操作。就这一点而言,流程图或框图中的每个方框可以表示指令的模块,片段或部分,其包括用于实施指定操作的一个或多个可执行指令。

[0128] 在一些替代实施方式中,方框中指出的操作可以不按图中指出的顺序发生。例如,取决于所涉及的功能,连续示出的两个框可以基本同时执行,或者有时可以以相反的顺序执行。在其他示例中,通常可以按增加的数字顺序来执行块,而在其他示例中,可以按变化的顺序来执行一个或多个块,结果被存储并利用在随后的或不立即跟随的其他块中。还应注意,框图和/或流程图说明的每个方框以及框图和/或流程图说明中的方框的组合可以由

执行指定功能或动作,或实现特殊用途的硬件和计算机指令的组合的专用的基于硬件的系统来实现。

[0129] 在下面的权利要求中可以找到的所有装置或步骤加上功能元件的相应结构,材料,动作和等同物,旨在包括用于执行图1中的功能的任何结构,材料或动作。

[0130] 在一个或多个实施例中,系统可以包括:耦接到通信总线的主处理器;通过通信总线可通信地链接到主处理器的第一硬件加速器;以及通过通信总线可通信地链接到主处理器的第二硬件加速器。第一硬件加速器和第二硬件加速器通过独立于通信总线的加速器链路直接耦接。主处理器被配置为直接通过加速器链路在第一硬件加速器和第二硬件加速器之间发起数据传输。

[0131] 一方面,主处理器被配置为通过通信总线与第一硬件加速器和第二硬件加速器通信。

[0132] 另一方面,数据传输包括第一硬件加速器通过加速器链接访问第二硬件加速器的存储器。

[0133] 另一方面,主处理器被配置为通过向第一硬件加速器发送包括目标地址的数据来访问第二硬件加速器的存储器,其中目标地址被主处理器转换为与第二硬件加速器相对应,以及其中第一硬件加速器基于目标地址发起事务,以通过加速器链路访问第二硬件加速器的存储器。

[0134] 另一方面,第二硬件加速器被配置为响应于经由加速器链路接收到事务,将用于数据传输的目标地址递减第二硬件加速器的地址范围的上限,并确定所减少的目标地址是否为本地。

[0135] 另一方面,主处理器被配置为基于耦接到通信总线的第二硬件加速器的直接存储器访问电路的状态来发起第一硬件加速器和第二硬件加速器之间的数据传输。

[0136] 另一方面,主处理器被配置为自动确定环形拓扑中的第一硬件加速器和第二硬件加速器的顺序。

[0137] 另一方面,主处理器被配置为使用远程缓冲器标志来跟踪与第一硬件加速器和第二硬件加速器相对应的缓冲器。

[0138] 在一个或多个实施例中,硬件加速器可以包括:端点,被配置为通过通信总线与主处理器进行通信;存储器控制器,其耦接到硬件加速器本地的存储器;以及链接电路,其被耦接到端点和存储器控制器。链接电路被配置为与也耦接到通信总线的目标硬件加速器建立加速器链接。加速器链接是硬件加速器和目标硬件加速器之间的直接连接,独立于通信总线。

[0139] 一方面,链接电路被配置为通过加速器链路发起与目标硬件加速器的数据传输,并且响应于硬件加速器通过通信总线接收到的来自主处理器的指令,发生数据传输。

[0140] 另一方面,链接电路包括第一存储器映射到流映射器电路和第二存储器映射到流映射器电路,每个被配置为将数据流转换为存储器映射的事务,并且将存储器映射的事务转换为数据流。

[0141] 另一方面,每个映射到流映射器的存储器被配置为将接收到的事务中的目标地址递减硬件加速器的地址范围的上限。

[0142] 另一方面,链接电路包括被配置为发送和接收流数据的第一收发器,以及被耦接

到第一收发器和第一存储器映射到流映射器电路的第一重发引擎。

[0143] 一方面,链接电路还包括被配置为发送和接收流数据的第二收发器,以及被耦接到第二收发器以及第二存储器映射到流映射器电路的第二重发引擎。

[0144] 在一个或多个实施例中,一种方法可以包括:在第一硬件加速器内接收通过通信总线从主处理器发送的用于数据传输的指令和目标地址,第一硬件加速器将目标地址与对应于第一硬件加速器的地址范围的上限进行比较,并且响应于基于比较确定目标地址超出了地址范围,第一硬件加速器发起与第二硬件加速器的事务,以通过使用直接耦接第一硬件加速器和第二硬件加速器的加速器链路,以执行数据传输。

[0145] 一方面,加速器链路独立于通信总线。

[0146] 另一方面,发起事务包括发起存储器映射的事务并将存储器映射的事务转换为要通过加速器链路发送的数据流。

[0147] 另一方面,该方法包括:响应于在第二硬件加速器中接收到事务,第二硬件加速器从目标地址减去第二硬件加速器的地址范围的上限,并确定相减的结果是否为在第二个硬件加速器的地址范围内。

[0148] 另一方面,第二硬件加速器将事务作为数据流接收,并将数据流转换为存储器映射的事务。

[0149] 另一方面,该方法包括确定第二硬件加速器的直接存储器访问电路的状态,并响应于第二硬件加速器的直接存储器访问电路的状态来发起数据传输。

[0150] 本文提供的发明性布置的描述是出于说明的目的,而不是穷举性的或限于所公开的形式和实例。这里使用的术语被选择为用来解释发明的装置的原理,对市场上发现的技术的实际应用或对技术的改进,和/或使本领域的其他普通技术人员能够理解这里公开的发明的装置。在不脱离所描述的发明布置的范围和精神的情况下,修改和变化对本领域普通技术人员而言是显而易见的。因此,应指出以下权利要求,而不是前面的公开内容,以指示这些特征和实施方式的范围。

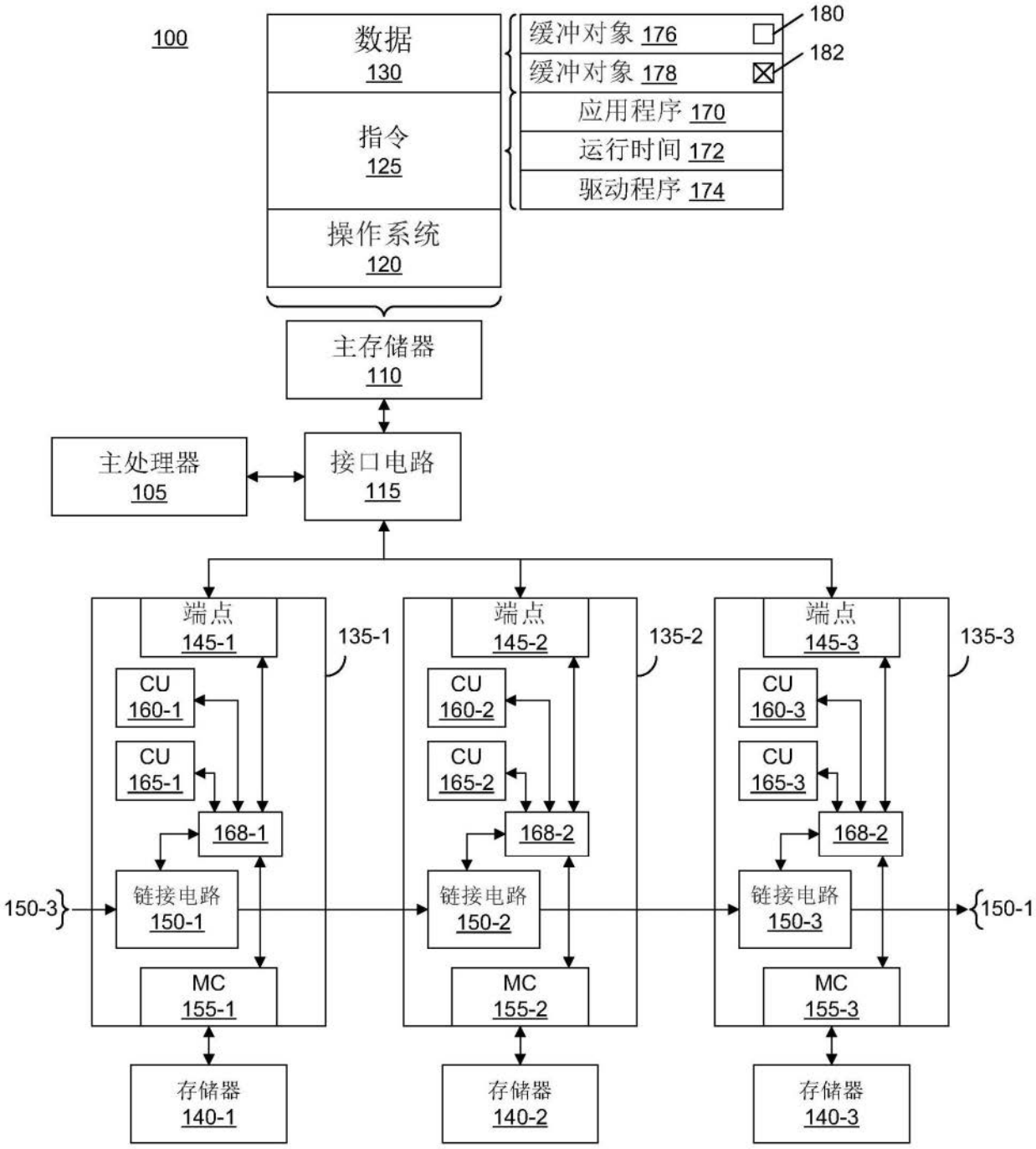

100
数据
130
指令
125
操作系统
120
缓冲对象 176
缓冲对象 178
应用程序 170
运行时间 172
驱动程序 174
180
182
主存储器
110
主处理器
105
接口电路
115
端点
145-1
端点
145-2
端点
145-3
135-1
135-2
135-3
CU
160-1
CU
165-1
CU
160-2
CU
165-2
CU
160-3
CU
165-3
168-1
168-2
168-2
150-3
链接电路
150-1
链接电路
150-2
链接电路
150-3
150-1
MC
155-1
MC
155-2
MC
155-3
存储器
140-1
存储器
140-2
存储器
140-3

图1

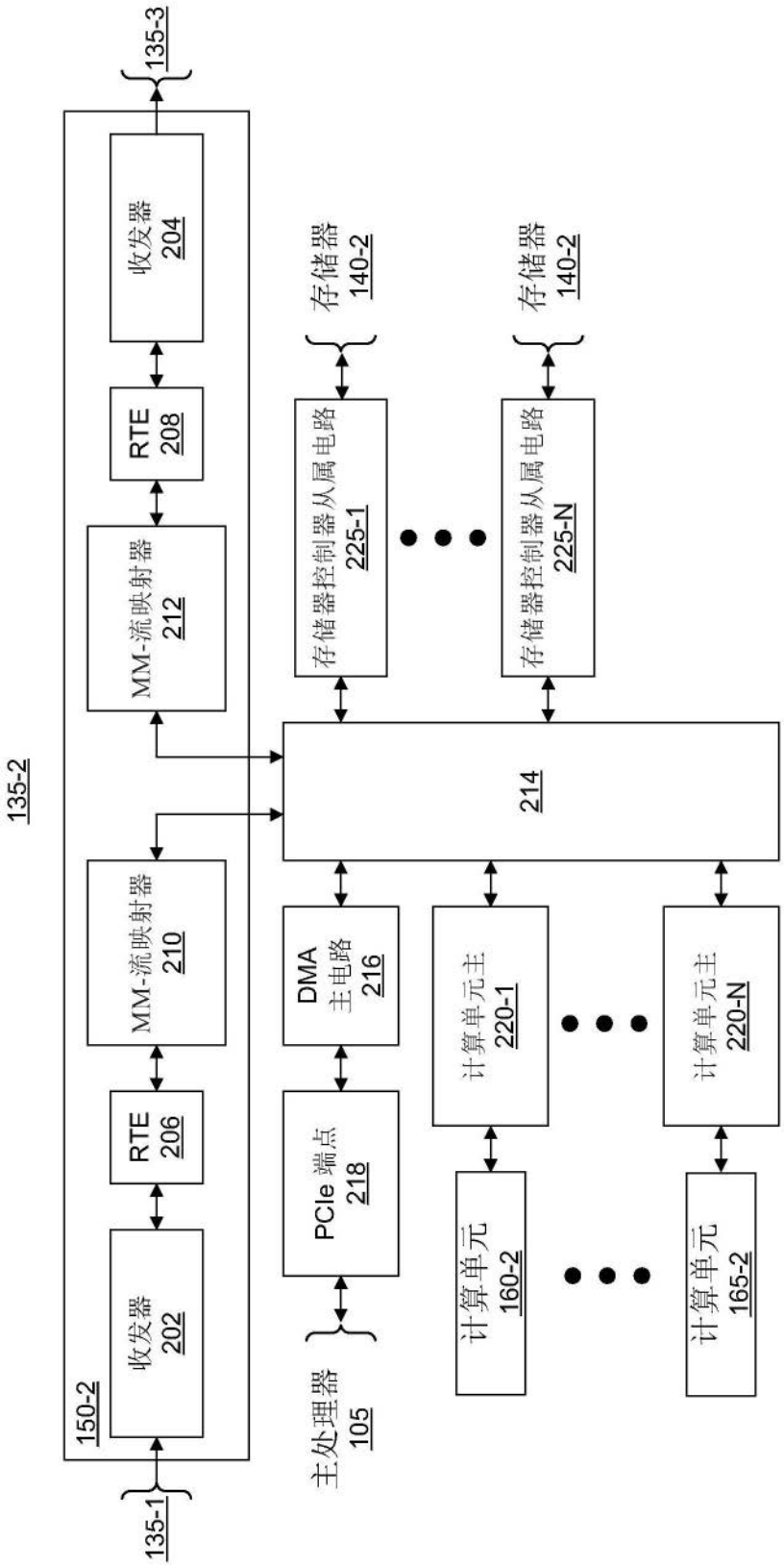

135-2
150-2
收发器 202
RTE 206
MM-流映射器 210
MM-流映射器 212
RTE 208
收发器 204
135-1
135-3
主处理器 105
PCIe 端点 218
DMA 主电路 216
214
存储器控制器从属电路 225-1
存储器控制器从属电路 225-N
存储器 140-2
存储器 140-2
计算单元 160-2
计算单元 165-2
计算单元主 220-1
计算单元主 220-N

图2

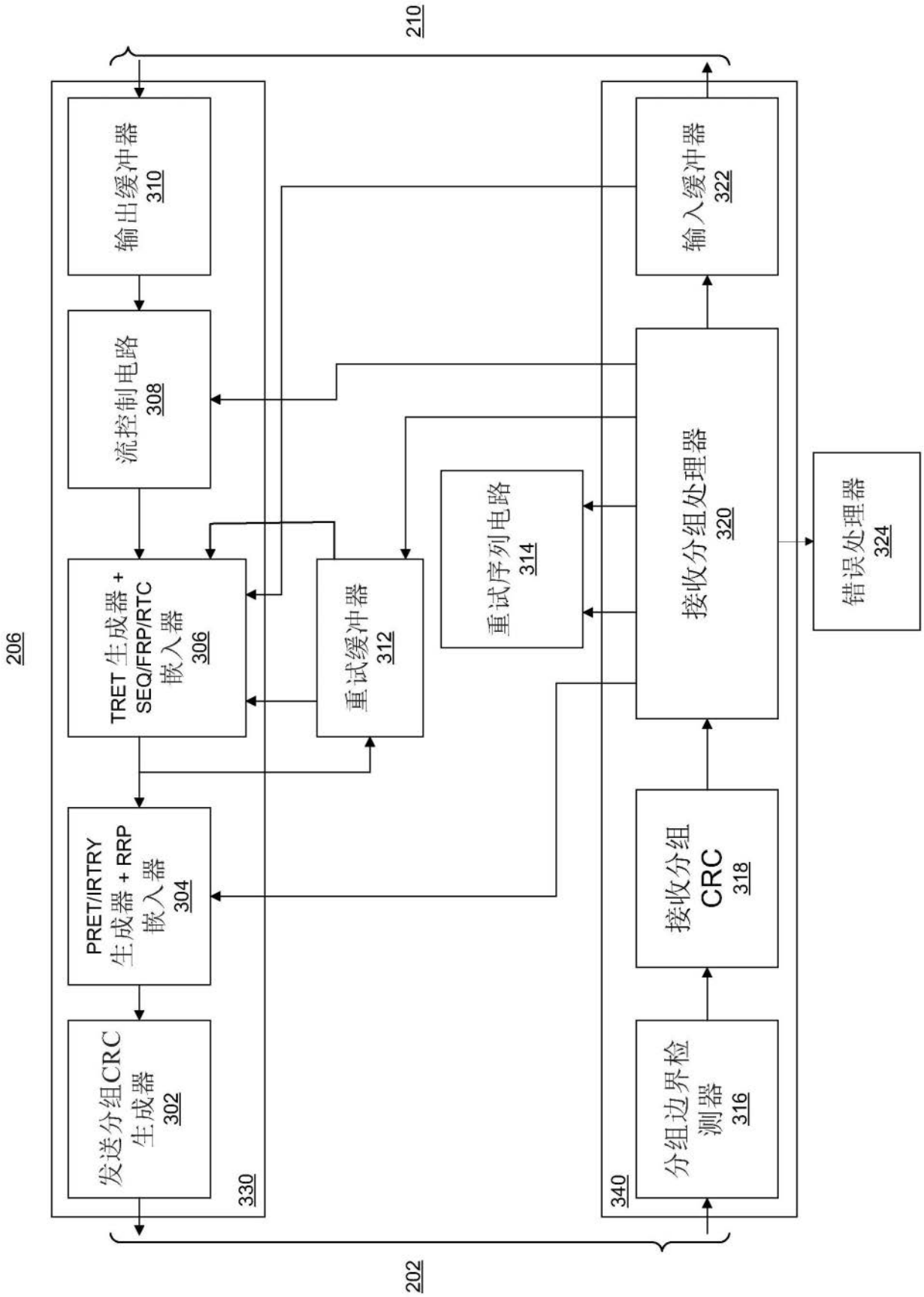

210
206
输出缓冲器 310
流控制电路 308
TRET 生成器 + SEQ/FRP/RTC 嵌入器 306
重试缓冲器 312
PRET/IRTRY 生成器 + RRP 嵌入器 304
发送分组CRC 生成器 302
330
重试序列电路 314
输入缓冲器 322
接收分组处理器 320
错误处理器 324
接收分组 CRC 318
分组边界检测器 316
340
202

图3

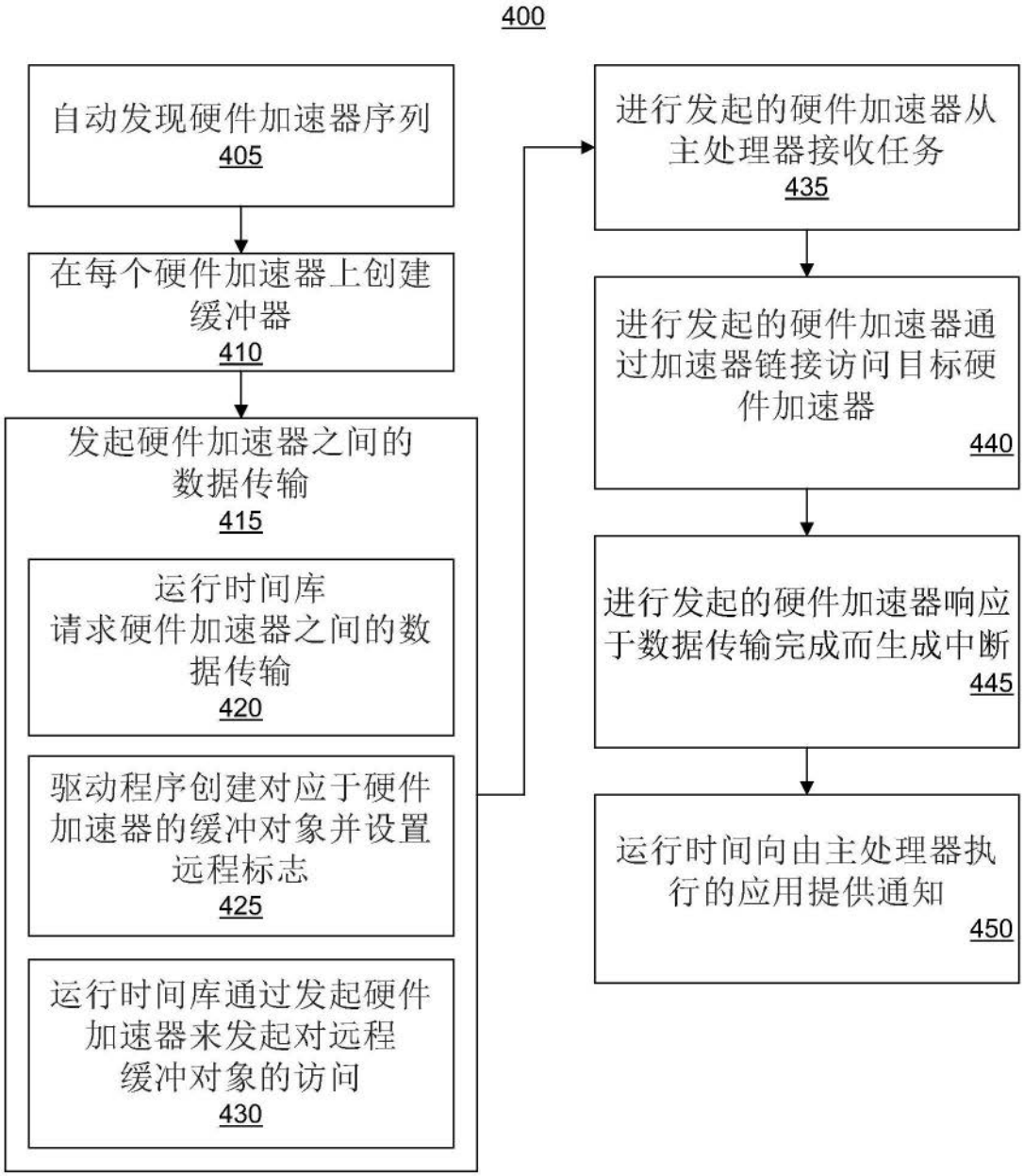

400
自动发现硬件加速器序列
405
在每个硬件加速器上创建缓冲器
410
发起硬件加速器之间的数据传输
415
运行时间库请求硬件加速器之间的数据传输
420
驱动程序创建对应于硬件加速器的缓冲对象并设置远程标志
425
运行时间库通过发起硬件加速器来发起对远程缓冲对象的访问
430
进行发起的硬件加速器从主处理器接收任务
435
进行发起的硬件加速器通过加速器链接访问目标硬件加速器
440
进行发起的硬件加速器响应于数据传输完成而生成中断
445
运行时间向由主处理器执行的应用提供通知
450

图4

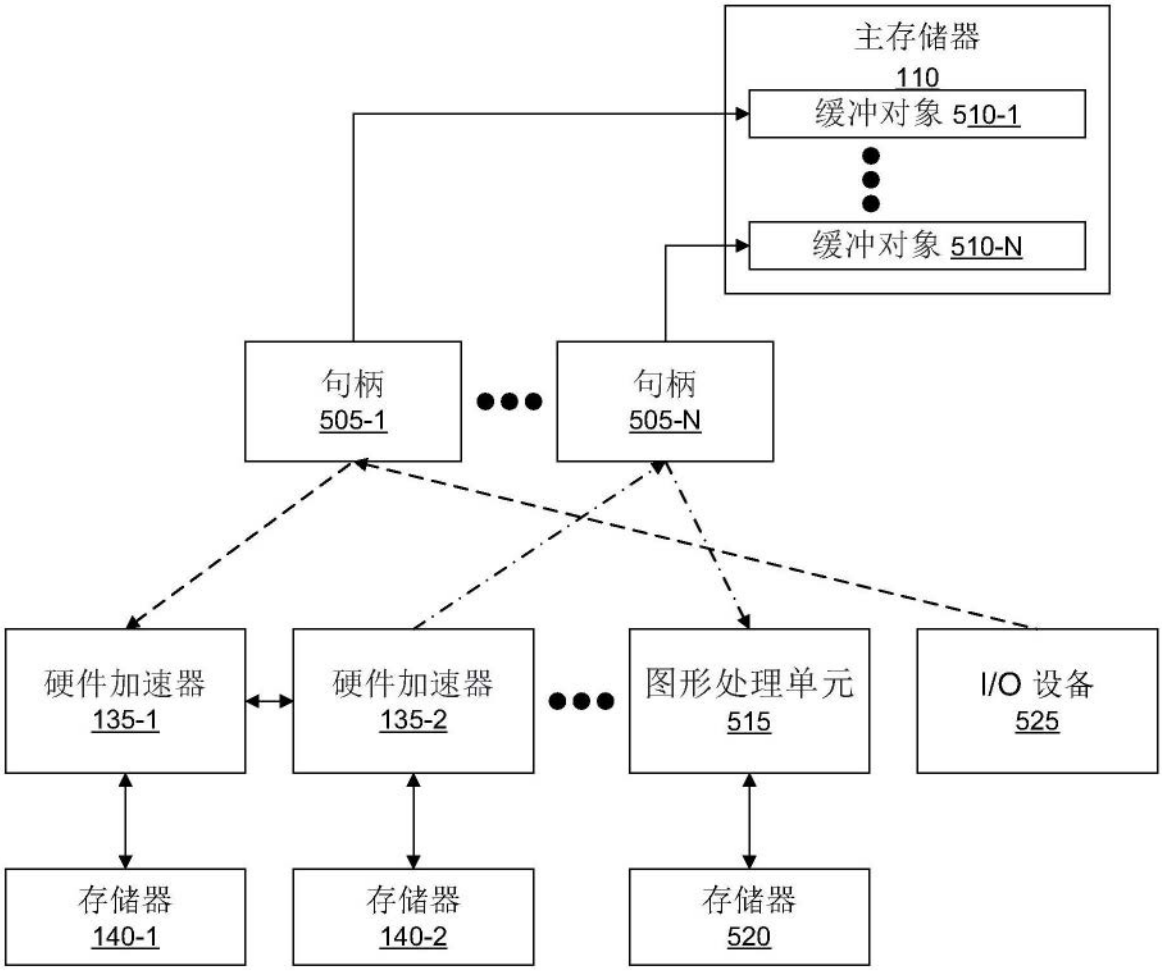

主存储器
110
缓冲对象 510-1
缓冲对象 510-N
句柄
505-1
句柄
505-N
硬件加速器
135-1
硬件加速器
135-2
图形处理单元
515
I/O 设备
525
存储器
140-1
存储器
140-2
存储器
520

图5

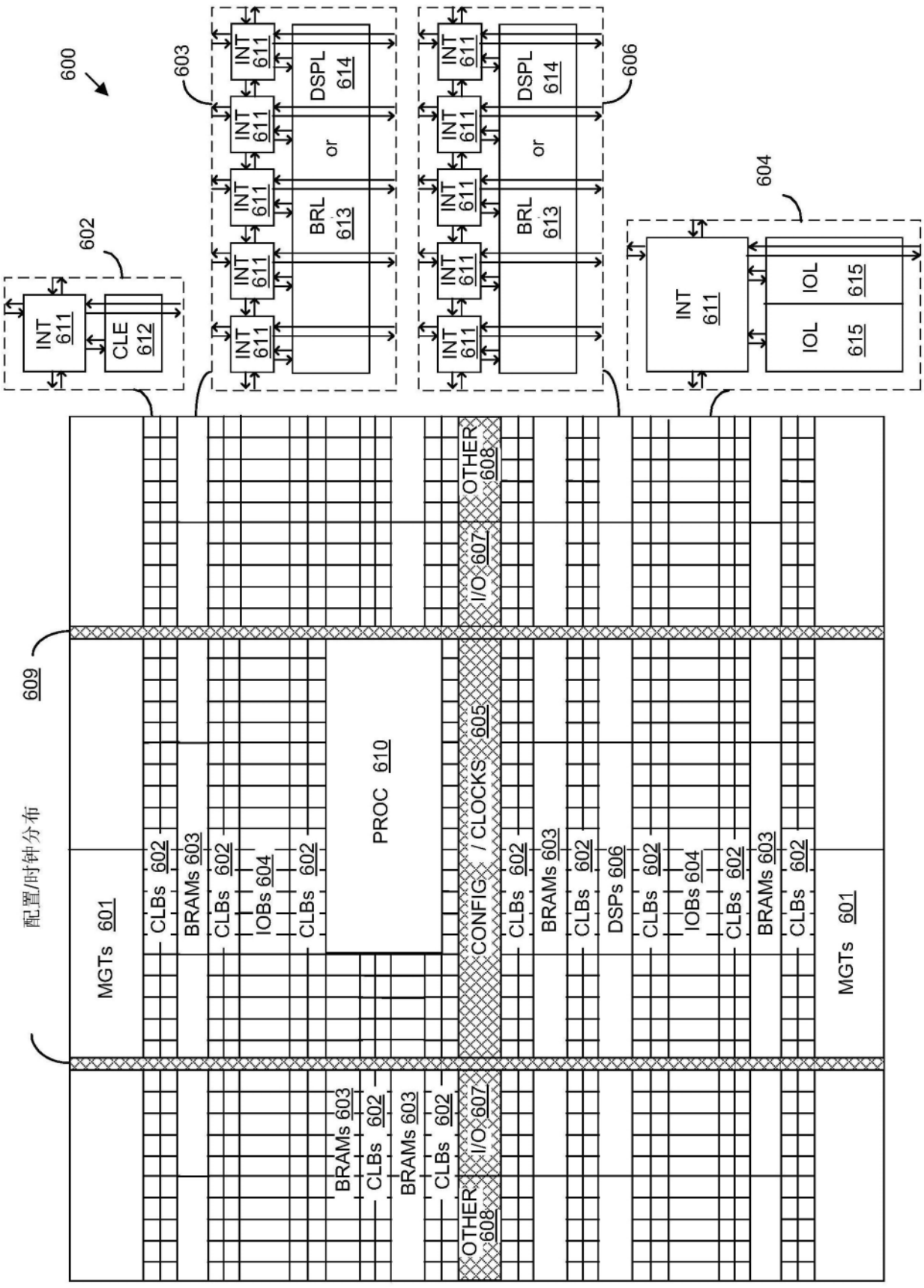

600
602
603
604
606
609
INT 611
CLE 612
DSPL 614
or
BRL 613
IOL 615
MGTs 601
CLBs 602
BRAMs 603
IOBs 604
PROC 610
CONFIG / CLOCKS 605
I/O 607
OTHER 608
DSPs 606
配置/时钟分布

图6