



(12)发明专利

(10)授权公告号 CN 109144906 B

(45)授权公告日 2019.11.26

(21)申请号 201710451238.X

(22)申请日 2017.06.15

(65)同一申请的已公布的文献号
申请公布号 CN 109144906 A

(43)申请公布日 2019.01.04

(73)专利权人 北京忆芯科技有限公司
地址 100085 北京市海淀区创业中路32号
楼32-1-1-167室

(72)发明人 黄好城 王祎磊 伍德斌 兰彤

(74)专利代理机构 北京卓特专利代理事务所
(普通合伙) 11572

代理人 段宇

(51)Int.Cl.

G06F 13/16(2006.01)

G06F 13/28(2006.01)

(56)对比文件

- CN 102521179 A, 2012.06.27,
- CN 101673253 A, 2010.03.17,
- CN 1679295 A, 2005.10.05,
- CN 103473188 A, 2013.12.25,
- CN 102567256 A, 2012.07.11,
- CN 101290605 A, 2008.10.22,
- US 6058438 A, 2000.05.02,

审查员 刘朝兵

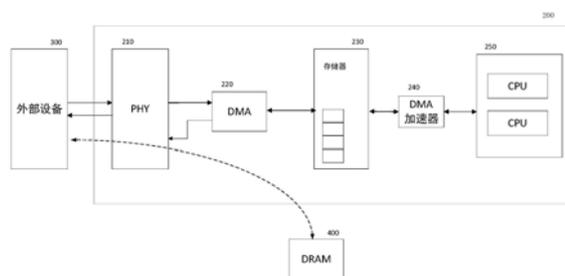
权利要求书2页 说明书17页 附图8页

(54)发明名称

电子设备及其DMA命令处理方法

(57)摘要

本申请公开了电子设备及其DMA命令处理方法。公开的电子设备包括直接存储器访问加速器,直接存储器访问加速器将中央处理器提供的数据包转换为直接存储器访问命令并写入存储器;以及获取存储器中的直接存储器访问命令的执行结果;直接存储器访问模块依据从存储器获取的直接存储器访问命令发起直接存储器访问传输,并将直接存储器访问命令的执行结果写入存储器;电子设备通过物理层模块同电子设备的外部设备通信。本申请的DMA加速器替代CPU与存储器进行交互,大大减轻了CPU的负载,加快了DMA命令的处理速度。



1. 一种电子设备,其特征在于,包括物理层模块、直接存储器访问模块、存储器、直接存储器访问加速器以及中央处理器,所述物理层模块与所述直接存储器访问模块耦合,所述存储器耦合到所述直接存储器访问模块和所述直接存储器访问加速器,所述直接存储器访问加速器与所述中央处理器耦合;

其中,所述直接存储器访问加速器将所述中央处理器提供的数据包转换为直接存储器访问命令并写入所述存储器;以及获取所述存储器中的直接存储器访问命令的执行结果;

所述直接存储器访问模块依据从所述存储器获取的直接存储器访问命令发起直接存储器访问传输,并将直接存储器访问命令的执行结果写入所述存储器;

所述电子设备通过所述物理层模块同所述电子设备的外部设备通信;

所述直接存储器访问加速器包括直接存储器访问命令接收单元和第一处理单元,以及直接存储器访问命令完成单元和第二处理单元;其中:

所述第二处理单元与所述存储器耦合,所述第二处理单元从所述存储器获取直接存储器访问命令执行结果;以及

所述直接存储器访问命令完成单元耦合到所述第二处理单元和所述中央处理器,所述直接存储器访问命令完成单元从所述第二处理单元获取直接存储器访问命令的执行结果,并向所述中央处理器提供直接存储器访问命令的执行结果;

所述直接存储器访问加速器还包括指针管理器,所述指针管理器与所述第一处理单元和所述第二处理单元耦合,所述第一处理单元从所述指针管理器获取写指针,所述第二处理单元从所述指针管理器获取读指针和向所述指针管理器更新读完成指针。

2. 根据权利要求1所述的电子设备,其特征在于,所述直接存储器访问加速器设有用于所述中央处理器访问的流式写入接口和流式读出接口。

3. 根据权利要求1-2中任一项所述的电子设备,其特征在于,所述直接存储器访问命令接收单元耦合到所述中央处理器和所述第一处理单元,所述直接存储器访问命令接收单元接收所述中央处理器提供的数据包;以及

所述第一处理单元与所述存储器耦合,所述第一处理单元从所述直接存储器访问命令接收单元获取所述中央处理器提供的数据包,将所述数据包转换为直接存储器访问命令,并写入所述存储器。

4. 根据权利要求1-2中任一项所述的电子设备,其特征在于,所述存储器存储的多个直接存储器访问命令被组织为队列,所述第一处理单元依据所述队列的写指针将所述直接存储器访问命令写入所述存储器。

5. 根据权利要求4所述的电子设备,其特征在于,所述第二处理单元依据所述队列的读指针从所述存储器获取所述直接存储器访问命令的执行结果。

6. 一种直接存储器访问命令处理方法,其特征在于,包括如下步骤:

接收数据包;

将所述数据包转换为直接存储器访问命令,并写入存储器;

响应于所述存储器中的直接存储器访问命令的执行结果被更新,获取被更新的直接存储器访问命令的执行结果;

依据写指针将直接存储器访问命令写入所述存储器;

更新所述写指针,并将更新后的写指针写入所述存储器;

获取读完成指针；

其中，所述写指针指向所述存储器中存储直接存储器访问命令的队列的队尾，所述读完成指针指向所述存储器中存储直接存储器访问命令的队列的队头；

其中，依据所述读完成指针落后于所述写指针，确定所述存储器中存储直接存储器访问命令的队列未满载；仅在所述存储器中的队列未满载时，将直接存储器访问命令写入所述存储器。

7. 根据权利要求6所述的直接存储器访问命令处理方法，其特征在于，包括：

提供流式接口的状态标记；

若所述流式接口的状态标记为可用状态标记，从所述流式接口接收数据包或向所述流式接口写入直接存储器访问命令的执行结果。

电子设备及其DMA命令处理方法

技术领域

[0001] 本申请涉及电子设备技术领域,尤其涉及一种电子设备中的直接存储器访问技术。

背景技术

[0002] 在电子设备与外部设备之间可以进行DMA (Direct Memory Access,直接存储器访问) 传输。如图1所示,电子设备100与外部设备300之间进行DMA传输。电子设备100包括物理层 (PHY) 模块110、DMA模块120、存储器130以及中央处理器 (CPU) 140。

[0003] CPU 140通过生成DMA命令(或称DMA描述符,用于指示在电子设备100与外部设备300之间的一次或多次DMA传输)并写入存储器130来指示DMA模块120进行DMA传输。并且,响应于DMA传输的完成,DMA模块120将DMA命令的执行结果写入存储器130,从而使CPU 140知晓DMA命令处理完成,并获得DMA命令的执行结果。

[0004] 外部设备300通过PHY模块110耦合到电子设备100。PHY模块110可以是用于处理PCIe底层协议的PCIe PHY模块、用于处理FC底层协议的FC PHY模块或用于处理以太网底层协议的以太网PHY模块。DMA模块120通过PHY模块110在外部设备300与电子设备100之间维护DMA传输。例如,DMA模块120从存储器130获取指示DMA传输的DMA命令,然后依照DMA命令发起DMA传输。

[0005] 电子设备100还耦合到存储器400。在DMA传输中,在DMA模块120的控制下,将存储器400的数据传输给外部设备300,或者将外部设备300提供的数据存储到存储器400。作为举例,存储器400是DRAM(动态随机存取存储器),同存储器130相比,存储器400具有更大的存储容量。

[0006] CPU在存储器130中建立描述符表或描述符链接表,描述符表或链接表中包括一组描述符,每个描述符都描述了数据块移动方向、源地址、目的地地址以及可选的传输的字节数。因此,在处理过程中,CPU不需要为要从某一源移动到某一目的地的每个数据块对DMA控制器进行编程操作。

[0007] 但是,其中,CPU与DMA模块通过存储器130交换DMA命令时,CPU需要按DMA模块要求的格式和数据结构提交DMA命令,这增加了CPU的负担。

发明内容

[0008] 本申请的目的在于提供电子设备及其直接存储器访问命令处理方法,用于减轻CPU的负担并加快DMA命令的处理速度。

[0009] 为达到上述目的,根据本申请的第一方面,提供了根据本申请第一方面的第一电子设备,其中,包括物理层模块、直接存储器访问模块、存储器、直接存储器访问加速器以及中央处理器,物理层模块与直接存储器访问模块耦合,存储器耦合到直接存储器访问模块和直接存储器访问加速器,直接存储器访问加速器与中央处理器耦合;

[0010] 其中,直接存储器访问加速器将中央处理器提供的数据包转换为直接存储器访问

命令并写入存储器;以及获取存储器中的直接存储器访问命令的执行结果;

[0011] 直接存储器访问模块依据从存储器获取的直接存储器访问命令发起直接存储器访问传输,并将直接存储器访问命令的执行结果写入存储器;

[0012] 电子设备通过物理层模块同电子设备的外部设备通信。

[0013] 根据本申请的第一方面的第一电子设备,提供了根据本申请第一方面的第二电子设备,其中,直接存储器访问加速器设有用于中央处理器访问的流式接口或先入先出接口。

[0014] 根据本申请的第一方面的第一电子设备,提供了根据本申请第一方面的第三电子设备,直接存储器访问加速器设有用于中央处理器访问的流式写入接口和流式读出接口。

[0015] 根据本申请的第一方面的第一至第三电子设备之一,提供了根据本申请第一方面的第四电子设备,其中,直接存储器访问加速器包括直接存储器访问命令接收单元和第一处理单元;其中:

[0016] 直接存储器访问命令接收单元耦合到中央处理器和第一处理单元,直接存储器访问命令接收单元接收中央处理器提供的数据包;以及

[0017] 第一处理单元与存储器耦合,第一处理单元从直接存储器访问命令接收单元获取中央处理器提供的数据包,将数据包转换为直接存储器访问命令,并写入存储器。

[0018] 根据本申请的第一方面的第一至第四电子设备之一,提供了根据本申请第一方面的第五电子设备,其中,直接存储器访问加速器包括直接存储器访问命令完成单元和第二处理单元;其中:

[0019] 第二处理单元与存储器耦合,第二处理单元从存储器获取直接存储器访问命令执行结果;以及

[0020] 直接存储器访问命令完成单元耦合到第二处理单元和中央处理器,直接存储器访问命令完成单元从第二处理单元获取直接存储器访问命令的执行结果,并向中央处理器提供直接存储器访问命令的执行结果。

[0021] 根据本申请的第一方面的第四至第五电子设备,提供了根据本申请第一方面的第六电子设备,其中,直接存储器访问命令接收单元和/或直接存储器访问命令完成单元设有缓存区。

[0022] 根据本申请的第一方面的第一至第六电子设备之一,提供了根据本申请第一方面的第七电子设备,其中,存储器存储的多个直接存储器访问命令被组织为队列,第一处理单元依据队列的写指针将直接存储器访问命令写入存储器。

[0023] 根据本申请的第一方面的第七电子设备,提供了根据本申请第一方面的第八电子设备,其中,第二处理单元依据队列的读指针从存储器获取直接存储器访问命令的执行结果。

[0024] 根据本申请的第一方面的第八电子设备,提供了根据本申请第一方面的第九电子设备,其中,第二处理单元依据直接存储器访问命令的执行结果被中央处理器接收而更新读完成指针。

[0025] 根据本申请的第一方面的第九电子设备,提供了根据本申请第一方面的第十电子设备,其中,直接存储器访问加速器还包括指针管理器,指针管理器与第一处理单元和第二处理单元耦合,第一处理单元从指针管理器获取写指针,第二处理单元从指针管理器获取读指针和向指针管理器更新读完成指针。

[0026] 根据本申请的第一方面的第九电子设备,提供了根据本申请第一方面的第十一电子设备,其中,第一处理单元与第二处理单元耦合,第一处理单元从第二处理单元获取读完成指针。

[0027] 根据本申请的第一方面的第九电子设备,提供了根据本申请第一方面的第十二电子设备,其中,第二处理单元向存储器写入读完成指针,并且第一处理单元从存储器获取读完成指针。

[0028] 根据本申请的第一方面的第一至第六电子设备之一,提供了根据本申请第一方面的第十三电子设备,其中,存储器中存储的多个直接存储器访问命令被组织为链表、线性表或数组。

[0029] 根据本申请的第一方面的第一至第十三电子设备之一,提供了根据本申请第一方面的第十四电子设备,其中,直接存储器访问命令指示直接存储器访问模块在多个数据帧中传输数据包指示的待传输数据。

[0030] 根据本申请的第一方面的第十四电子设备,提供了根据本申请第一方面的第十五电子设备,其中,数据帧的大小是512字节。

[0031] 根据本申请的第一方面的第十四电子设备,提供了根据本申请第一方面的第十六电子设备,其中,数据帧的大小是用高级加密标准加密的数据块大小。

[0032] 根据本申请的第一方面的第十四电子设备,提供了根据本申请第一方面的第十七电子设备,其中,数据帧的大小是用循环冗余校验码校验的数据块的大小。

[0033] 根据本申请的第一方面的第一至十七电子设备之一,提供了根据本申请第一方面的第十八电子设备,其中,直接存储器访问加速器与中央处理器之间通过多个相互独立的流进行数据传输。

[0034] 根据本申请的第一方面的第一至十七电子设备之一,提供了根据本申请第一方面的第十九电子设备,其中,数据包指示标识符,直接存储器访问加速器依据标识符确定直接存储器访问命令在存储器中的存储地址。

[0035] 根据本申请的第一方面的第二至十七电子设备之一,提供了根据本申请第一方面的第二十电子设备,其中,直接存储器访问加速器包括一个或多个流式接口或先入先出接口,直接存储器访问加速器依据接收数据包的流式接口或先入先出接口,确定直接存储器访问命令在存储器中的存储地址。

[0036] 根据本申请的第二方面,提供了根据本申请第二方面的第一直接存储器访问命令处理方法,包括如下步骤:

[0037] 接收数据包;

[0038] 将数据包转换为直接存储器访问命令,并写入存储器;

[0039] 响应于存储器中的直接存储器访问命令的执行结果被更新,获取被更新的直接存储器访问命令的执行结果。

[0040] 根据本申请的第二方面的第一直接存储器访问命令处理方法,提供了根据本申请第二方面的第二直接存储器访问命令处理方法,其中,包括:通过流式接口进行数据传输。

[0041] 根据本申请的第二方面的第二直接存储器访问命令处理方法,提供了根据本申请第二方面的第三直接存储器访问命令处理方法其中,包括:

[0042] 提供流式接口的状态标记;

[0043] 若流式接口的状态标记为可用状态标记,从流式接口接收数据包或向流式接口写入直接存储器访问命令的执行结果。

[0044] 根据本申请的第二方面的第一直接存储器访问命令处理方法,提供了根据本申请第二方面的第四直接存储器访问命令处理方法其中,包括:通过先入先出接口进行数据传输。

[0045] 根据本申请的第二方面的第四直接存储器访问命令处理方法,提供了根据本申请第二方面的第五直接存储器访问命令处理方法其中,包括:

[0046] 提供先入先出接口的状态;

[0047] 若先入先出接口内的先入先出队列未滿,则从先入先出接口接收数据包。

[0048] 根据本申请的第二方面的第四直接存储器访问命令处理方法,提供了根据本申请第二方面的第六直接存储器访问命令处理方法其中,包括:

[0049] 提供先入先出接口的状态;

[0050] 若先入先出接口内的先入先出队列为非空,则向先入先出接口写入直接存储器访问命令的执行结果。

[0051] 根据本申请的第二方面的第一至第六直接存储器访问命令处理方法之一,提供了根据本申请第二方面的第七直接存储器访问命令处理方法其中,包括:

[0052] 依据写指针将直接存储器访问命令写入存储器;

[0053] 更新写指针,并将更新后的写指针写入存储器;

[0054] 获取读完成指针;

[0055] 其中,写指针指向存储器中存储直接存储器访问命令的队列的队尾,读完成指针指向存储器中存储直接存储器访问命令的队列的队头;

[0056] 其中,依据读完成指针落后于写指针,确定存储器中存储直接存储器访问命令的队列未滿;仅在存储器中的队列未滿时,将直接存储器访问命令写入存储器。

[0057] 根据本申请的第二方面的第七直接存储器访问命令处理方法,提供了根据本申请第二方面的第八直接存储器访问命令处理方法其中,直接存储器访问模块从存储器中的队列的队头获取直接存储器访问命令;

[0058] 响应于直接存储器访问命令被完成,直接存储器访问模块将直接存储器访问命令的执行结果写入存储器,更新存储器中的读指针。

[0059] 根据本申请的第二方面的第七直接存储器访问命令处理方法,提供了根据本申请第二方面的第九直接存储器访问命令处理方法其中,响应于读指针超前于读完成指针,依据读完成指针从存储器获取直接存储器访问命令的执行结果;

[0060] 更新读完成指针;

[0061] 其中,响应于直接存储器访问命令的执行结果被写入存储器而更新读指针。

[0062] 根据本申请的第二方面的第七直接存储器访问命令处理方法,提供了根据本申请第二方面的第十直接存储器访问命令处理方法其中,包括:

[0063] 响应于读指针与读完成指针不同,从存储器获取直接存储器访问命令的执行结果。

[0064] 根据本申请的第二方面的第七至第十直接存储器访问命令处理方法之一,提供了根据本申请第二方面的第十一直接存储器访问命令处理方法其中,还包括:将读完成指针

写入存储器。

[0065] 根据本申请的第二方面的第一至第十一直接存储器访问命令处理方法之一,提供了根据本申请第二方面的第十二直接存储器访问命令处理方法其中,通过多个相互独立的流进行数据传输,并且每个流与存储器中的队列一一对应。

[0066] 根据本申请的第二方面的第十二直接存储器访问命令处理方法,提供了根据本申请第二方面的第十三直接存储器访问命令处理方法其中,还包括:

[0067] 提供每个流的状态标记;

[0068] 若流的状态标记为可用状态标记,通过流接收数据包;

[0069] 依据数据包所指示的流将直接存储器访问命令写入存储器中与数据包所指示的流相应的队列;

[0070] 通过被完成的直接存储器访问命令所在的队列对应的流,将直接存储器访问命令的执行结果提供给中央处理器。

[0071] 根据本申请的第二方面的第一至十一直接存储器访问命令处理方法之一,提供了根据本申请第二方面的第十四直接存储器访问命令处理方法其中,通过多个相互独立的流和多个流式接口进行数据传输;流与流式接口一一对应;

[0072] 直接存储器访问命令处理方法包括:

[0073] 依据接收数据包的流式接口识别数据包所属的流;

[0074] 向与被完成的直接存储器访问命令所在的流对应的流式接口写入直接存储器访问命令的执行结果。

[0075] 根据本申请的第二方面的第一至十四直接存储器访问命令处理方法,提供了根据本申请第二方面的第十五直接存储器访问命令处理方法其中,直接存储器访问命令指示直接存储器访问模块在多个数据帧中传输数据包指示的待传输数据。

[0076] 根据本申请的第二方面的第十五直接存储器访问命令处理方法,提供了根据本申请第二方面的第十六直接存储器访问命令处理方法其中,数据帧的大小是用高级加密标准加密的数据块大小。

[0077] 根据本申请的第二方面的第十五直接存储器访问命令处理方法,提供了根据本申请第二方面的第十七直接存储器访问命令处理方法其中,数据帧的大小是用循环冗余校验码校验的数据块的大小。

[0078] 根据本申请的第三方面,提供了根据本申请第三方面的通过队列交换信息的第一方法,包括:

[0079] 第一生产者将第一消息写入队列;

[0080] 第一消费者从队列获取第一消息;

[0081] 第一消费者将第一消息的处理结果写入队列内的第一消息;其中,第一消息的处理结果形成第二消息;

[0082] 第二消费者从队列获取第二消息。

[0083] 根据本申请的第三方面的通过队列交换信息的第一方法,提供了根据本申请第三方面的通过队列交换信息的第二方法,其中,第一生产者依据队列的写指针将第一消息写入队列;

[0084] 第一消费者依据队列的读指针从队列获取第一消息;

[0085] 第一消费者依据读指针将第二消息写入队列。

[0086] 根据本申请的第三方面的通过队列交换信息的第一方法,提供了根据本申请第三方面的通过队列交换信息的第三方法,其中,第一生产者依据队列的写指针将第一消息写入队列;

[0087] 第一消费者依据队列的读指针从队列获取第一消息;

[0088] 第一消费者记录第一消息在队列中的位置,并依据所记录的位置将第二消息写入队列。

[0089] 根据本申请的第三方面的通过队列交换信息的第一至第三方法之一,提供了根据本申请第三方面的通过队列交换信息的第四方法,其中,第二消费者依据队列的读完成指针从队列获取第二消息。

[0090] 根据本申请的第三方面的通过队列交换信息的第二至第四方法之一,提供了根据本申请第三方面的通过队列交换信息的第四方法,其中,包括:

[0091] 响应于第一消息被写入队列,更新写指针;

[0092] 响应于第二消息被写入队列,更新读指针。

[0093] 根据本申请的第三方面的通过队列交换信息的第四方法,提供了根据本申请第三方面的通过队列交换信息的第六方法,其中,包括:响应于第二消费者从队列获取第二消息,更新读完成指针。

[0094] 根据本申请的第三方面的通过队列交换信息的第四方法,提供了根据本申请第三方面的通过队列交换信息的第七方法,其中,包括:将读完成指针写入存储器,第一生产者从存储器获取读完成指针。

[0095] 根据本申请的第三方面的通过队列交换信息的第四方法,提供了根据本申请第三方面的通过队列交换信息的第八方法,其中,包括:第二消费者将读完成指针提供给第一生产者。

[0096] 根据本申请的第三方面的通过队列交换信息的第二或第三方法,提供了根据本申请第三方面的通过队列交换信息的第九方法,其中,包括:

[0097] 第一生产者将写指针提供给第一消费者;

[0098] 第一消费者将读指针提供给第二消费者。

[0099] 根据本申请的第三方面的通过队列交换信息的第一至第九方法之一,提供了根据本申请第三方面的通过队列交换信息的第十方法,其中,写指针包括回绕标记或发生回绕的次数信息。

[0100] 根据本申请的第三方面的通过队列交换信息的第四至第九方法之一,提供了根据本申请第三方面的通过队列交换信息的第十一方法,其中,响应于读完成指针和读指针指向相同的地址,第二消费者暂停从队列中获取第二消息。

[0101] 根据本申请的第三方面的通过队列交换信息的第四至第十一方法之一,提供了根据本申请第三方面的通过队列交换信息的第十二方法,其中,响应于读完成指针超前于写指针,第一生产者暂停向队列写入第一消息。

[0102] 根据本申请的第三方面的通过队列交换信息的第四至第十二方法之一,提供了根据本申请第三方面的通过队列交换信息的第十三方法,其中,读完成指针指向队列的队头;写指针指向队列的队尾。

[0103] 根据本申请的第三方面的通过队列交换信息的第一至第十三方法之一,提供了根据本申请第三方面的通过队列交换信息的第十四方法,其中,在存储器中提供队列。

[0104] 根据本申请的第三方面的通过队列交换信息的第一至第十四方法之一,提供了根据本申请第三方面的通过队列交换信息的第十五方法,其中,第一生产者与第二消费者是直接存储器访问加速器;

[0105] 第一消费者是直接存储器访问处理模块;

[0106] 第一消息是直接存储器访问命令,第二消息是直接存储器访问命令的处理结果。

[0107] 根据本申请的第三方面的通过队列交换信息的第十四方法,提供了根据本申请第三方面的通过队列交换信息的第十六方法,其中,存储器包括多个相互独立的队列,第一生产者、第一消费者和第二消费者通过多个队列交换信息。

[0108] 根据本申请的第四方面,提供了根据本申请第四方面的第一处理队列的系统,包括:

[0109] 第一生产者,将第一消息写入队列;

[0110] 第一消费者,从队列获取第一消息;并且将第一消息的处理结果写入队列内的第一消息;其中,第一消息的处理结果形成第二消息;

[0111] 第二消费者,从队列获取第二消息。

[0112] 根据本申请的第四方面的第一处理队列的系统,提供了根据本申请第四方面的第二处理队列的系统,其中,队列包括:

[0113] 写指针,指示向队列写入消息的地址;

[0114] 读指针,指示第一消费者从队列读出消息的地址。

[0115] 根据本申请的第四方面的第二处理队列的系统,提供了根据本申请第四方面的第三处理队列的系统,其中,第一生产者依据队列的写指针将第一消息写入队列;

[0116] 第一消费者依据队列的读指针从队列获取第一消息;

[0117] 第一消费者依据读指针将第二消息写入队列。

[0118] 根据本申请的第四方面的第二处理队列的系统,提供了根据本申请第四方面的第四处理队列的系统,其中,第一生产者依据队列的写指针将第一消息写入队列;

[0119] 第一消费者依据队列的读指针从队列获取第一消息;

[0120] 第一消费者记录第一消息在队列中的位置,并依据所记录的位置将第二消息写入队列。

[0121] 根据本申请的第四方面的第一至第四处理队列的系统之一,提供了根据本申请第四方面的第五处理队列的系统,其中,队列还包括读完成指针,指示第二消费者从队列读出消息的地址。

[0122] 根据本申请的第四方面的第五处理队列的系统,提供了根据本申请第四方面的第六处理队列的系统,其中,第二消费者依据队列的读完成指针从队列获取第二消息。

[0123] 根据本申请的第四方面的第二至第六处理队列的系统之一,提供了根据本申请第四方面的第七处理队列的系统,其中,响应于第一消息被写入队列,更新写指针;

[0124] 响应于第二消息被写入队列,更新读指针。

[0125] 根据本申请的第四方面的第五至第七处理队列的系统之一,提供了根据本申请第四方面的第八处理队列的系统,其中,响应于第二消费者从队列获取第二消息,更新读完成

指针。

[0126] 根据本申请的第四方面的第五处理队列的系统,提供了根据本申请第四方面的第九处理队列的系统,其中,还包括存储器,读完成指针被写入存储器,第一生产者从存储器获取读完成指针。

[0127] 根据本申请的第四方面的第五处理队列的系统,提供了根据本申请第四方面的第十处理队列的系统,其中,第二消费者将读完成指针提供给第一生产者。

[0128] 根据本申请的第四方面的第三或第四处理队列的系统,提供了根据本申请第四方面的第十一处理队列的系统,其中,第一生产者将写指针提供给第一消费者;

[0129] 第一消费者将读指针提供给第二消费者。

[0130] 根据本申请的第四方面的第一至第十一处理队列的系统,提供了根据本申请第四方面的第十二处理队列的系统,其中,写指针包括回绕标记或发生回绕的次数信息。

[0131] 根据本申请的第四方面的第五至第十二处理队列的系统,提供了根据本申请第四方面的第十三处理队列的系统,其中,响应于读完成指针和读指针指向相同的地址,第二消费者暂停从队列中获取第二消息。

[0132] 根据本申请的第四方面的第五至第十三处理队列的系统,提供了根据本申请第四方面的第十四处理队列的系统,其中,响应于读完成指针超前于写指针,第一生产者暂停向队列写入第一消息。

[0133] 根据本申请的第四方面的第五至第十四处理队列的系统,提供了根据本申请第四方面的第十五处理队列的系统,其中,读完成指针指向队列的队头;写指针指向队列的队尾。

[0134] 根据本申请的第四方面的第一至第十五处理队列的系统,提供了根据本申请第四方面的第十六处理队列的系统,其中,在存储器中提供队列。

[0135] 根据本申请的第四方面的第一至第十六处理队列的系统,提供了根据本申请第四方面的第十七处理队列的系统,其中,第一生产者与第二消费者是直接存储器访问加速器;

[0136] 第一消费者是直接存储器访问处理模块;

[0137] 第一消息是直接存储器访问命令,第二消息是直接存储器访问命令的处理结果。

[0138] 根据本申请的第四方面的第一至第十七处理队列的系统,提供了根据本申请第四方面的第十八处理队列的系统,其中,存储器包括多个相互独立的队列,第一生产者、第一消费者和第二消费者通过多个队列交换信息。

[0139] 本申请的技术方案获得的有益效果如下:

[0140] (1) 本申请的DMA加速器替代CPU与存储器进行交互,替代CPU生成符合DMA模块所需的格式、传输长度等要求的DMA命令,并替代CPU维护存储器中DMA命令的数据结构,将DMA命令写入存储器,并监视存储器中DMA命令的状态更新,以及将DMA命令的执行结果提供给CPU,大大减轻了CPU的负载,加快了DMA命令的处理速度。

[0141] (2) 本申请通过使用流式接口,CPU将数据写入到流,无须关心数据的存储地址和数据结构,从而减轻了CPU的负载;在读出数据时,CPU通过使用流式接口从流中获取数据,无须关心数据的存储地址和数据结构;尽管流式接口提供了可访问地址,但是该可访问地址为单一地址或指定地址,CPU无须处理地址的更新,也无须进行内存管理。

[0142] (3) 本申请通过对存储器中队列的维护,使得DMA加速器的第一处理单元和DMA模

块形成队列的一对生产者与消费者，DMA模块和DMA加速器的第二处理单元形成队列的另一对生产者与消费者，单一队列既用于DMA加速器向DMA模块提交DMA命令，也用于DMA模块向DMA加速器提交DMA命令的执行结果，降低了对存储器的存储空间的需求。

附图说明

[0143] 为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0144] 图1为现有技术的DMA传输的系统结构图；

[0145] 图2为根据本申请实施例的DMA传输的系统结构图；

[0146] 图3为根据本申请实施例的直接存储器访问加速器提供的流式写入接口的示意图；

[0147] 图4为根据本申请实施例的直接存储器访问加速器提供的流式读出接口的示意图；

[0148] 图5为本申请实施例一提供的直接存储器访问加速器的结构图；

[0149] 图6到图10为根据本申请实施例一提供的单一队列的指针示意图；

[0150] 图11为本申请实施例二提供的直接存储器访问加速器的结构图；

[0151] 图12为本申请实施例三提供的直接存储器访问命令处理方法的流程图；以及

[0152] 图13为本申请实施例的CPU通过直接存储器访问加速器执行直接存储器访问操作的流程图。

具体实施方式

[0153] 下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

[0154] 图2为根据本申请实施例的DMA传输的系统结构图。如图2所示，电子设备200包括物理层模块(PHY模块)210、直接存储器访问模块(DMA模块)220、存储器230、直接存储器访问加速器(DMA加速器)240以及中央处理器(CPU)250。PHY模块210与DMA模块220耦合，存储器230耦合到DMA模块220和DMA加速器240，DMA加速器240与CPU250耦合。

[0155] DMA加速器240将CPU250提供的数据包转换为DMA命令并写入存储器230，并获取存储器230中的DMA命令的执行结果。其中，DMA命令指示DMA模块220在多个数据帧中传输由数据包指示的待传输数据。

[0156] 具体地，DMA加速器240替代CPU 250生成符合DMA模块220所需的格式、传输长度等要求的DMA命令。作为一个实施例，CPU 250接口的数据总线宽度是32比特，要向外部设备300传输4KB的数据，并且数据帧的大小是用高级加密标准(AES)加密的数据块大小。DMA模块220接受的DMA命令为16或32字节，AES加密支持的数据块长度为512字节。因此，DMA加速器240依据接收的来自CPU 250的多个32比特数据生成DMA命令，DMA命令中指示8笔512字节

的DMA传输,以符合AES加密的要求。作为另一个实施例,数据帧的大小也可以是用循环冗余校验码校验的数据块的大小。

[0157] DMA加速器240替代CPU 250维护存储器230中存储的DMA命令的数据结构,将DMA命令写入存储器230,并监视存储器230中DMA命令的状态更新,并将DMA命令的执行结果提供给CPU 250。

[0158] DMA模块220通过PHY模块210在外部设备300与电子设备200之间维护DMA传输。例如,DMA模块220从存储器230获取指示DMA传输的DMA命令,然后依照DMA命令与外部设备300发起DMA传输,并将DMA命令的执行结果写入存储器230。

[0159] 电子设备200通过PHY模块210同电子设备的外部设备(如外部设备300)进行DMA传输。PHY模块210可以是用于处理PCIe底层协议的PCIe PHY模块、用于处理FC底层协议的FC PHY模块或用于处理以太网底层协议的以太网PHY模块。

[0160] 电子设备200还耦合到存储器400。在DMA传输中,在DMA模块220的控制下,将存储器400的数据传输给外部设备300,或者将外部设备300提供的数据存储到存储器400。作为一个实施例,存储器400是DRAM(动态随机存取存储器),同存储器230相比,存储器400具有更大的存储容量。

[0161] 作为一个实施例,DMA加速器设有用于CPU访问的流式接口。图3为根据本申请实施例的DMA加速器的流式写入接口的示意图;图4为根据本申请实施例的DMA加速器的流式读出接口的示意图。

[0162] 如图3所示,DMA加速器提供了具有可访问地址的流式写入接口和指示流式接口是否可用的状态标记。CPU通过可用状态标记识别出流式写入接口可用时,将与DMA命令相关的数据按指定宽度(例如32比特)写入流式写入接口。

[0163] 作为一个示例,CPU通过如下的代码段使用DMA加速器提供的流式写入接口提供数据。变量stream指示从流式写入接口的可用状态标记获取的流式写入接口的状态,在其指示流式写入接口可用时,通过DMA_Write (CMD) 过程将DMA命令数据(由变量CMD指示)写入流式接口的可访问地址。若DMA命令数据CMD较大,CPU通过执行程序将其拆分为多个32比特的数据提供给流式写入接口。

[0164] If (stream!=full)

[0165] DMA_Write (CMD)

[0166] 如图4所示,读出接口是流式读出接口。CPU访问流式读出接口提供的可用状态标记,当流式读出接口可用时,指示存在可读出的数据。相应地,CPU从流式读出接口提供的可访问地址读出数据(指示DMA命令的执行结果)。

[0167] 作为一个示例,CPU通过如下的代码段使用DMA加速器提供的流式读出接口获取数据。变量stream指示从流式读出接口的可用状态标记获取的流式读出接口的状态,在其指示流式读出接口非空时,通过DMA_Read (CMD) 过程从流式读出接口的可访问地址获取DMA命令的执行结果(由变量CMD_Status指示)。若DMA命令的执行结果尺寸较大,CPU通过执行程序按照指定宽度(例如32比特)通过读多次数据从流式读出接口获取DMA命令的执行结果。

[0168] If (stream!=empty)

[0169] DMA_Read (CMD_Status)

[0170] 可选地,DMA加速器向CPU提供中断。作为对中断的响应,CPU知晓流式读出接口存

在可读出数据,则CPU从流式读出接口读出数据。

[0171] 可选地,流式写入接口和流式读出接口也可以不使用可用状态标记。CPU直接向流式写入接口提供的可访问地址写入数据和直接从流式读出接口提供的可访问地址读出数据。

[0172] 通过使用流式接口,CPU将数据写入到流,无须关心数据的存储地址和数据结构,从而减轻了CPU的负载。在读出数据时,CPU通过使用流式接口从流中获取数据,无须关心数据的存储地址和数据结构。尽管流式接口提供了可访问地址,但是该可访问地址为单一地址或指定地址,即使访问多份数据,CPU也无须处理地址的更新,也无须进行内存管理。

[0173] 可以理解地,该可访问地址也可以是所访问的流的标识符,从而在软件看到,通过向流添加数据的操作,将数据提供给DMA加速器或DMA模块。

[0174] 作为另一个实施例,DMA加速器设有用于CPU访问的先入先出接口(FIFO接口)。DMA加速器向CPU提供FIFO写入接口。在DMA加速器提供的FIFO队列未满时,CPU可向FIFO队列的队尾写入数据。DMA加速器向CPU提供FIFO读出接口。在FIFO队列非空时,CPU可从FIFO队列的队头读出数据。

[0175] 实施例一

[0176] 图5为根据本申请实施例一提供的DMA加速器240(也参看图2)的结构图。如图5所示,DMA加速器240包括DMA命令接收单元501、处理单元502、DMA命令完成单元504和处理单元503。

[0177] 如图5所示,DMA命令接收单元501分别耦合到CPU 250和处理单元502,DMA命令接收单元501接收CPU 250提供的数据包。

[0178] 处理单元502与存储器230耦合,处理单元502从DMA命令接收单元501获取CPU 250提供的数据包,将数据包转换为符合DMA模块220可接受的格式,形成DMA命令并写入存储器230,并维护DMA模块220可接受的数据结构(例如,队列)。

[0179] 处理单元503与存储器230耦合,处理单元503从存储器230获取DMA命令执行结果。DMA命令完成单元504分别耦合到处理单元503和CPU 250,DMA命令完成单元504从处理单元503获取DMA命令的执行结果,并向CPU 250提供DMA命令的执行结果。

[0180] 作为一个实施例,DMA命令接收单元501和/或DMA命令完成单元504设有缓存区,以缓存来自CPU 250的数据包和/或来自处理单元503的DMA命令的执行结果。

[0181] 作为一个实施例,如图5所示,DMA命令接收单元501提供流式写入接口,以接收CPU 250提供的数据包。并且,DMA命令完成单元504缓存处理单元503提供的DMA命令的状态,并通过其流式读出接口向CPU 250指示流式读出接口可用或者流式读出接口中存在待读取的数据,CPU 250通过DMA命令完成单元504的流式读出接口从DMA命令完成单元504读取DMA命令的执行结果。

[0182] 通过向CPU 250提供流式接口来访问DMA加速器240,CPU 250无须维护存储器230中存储DMA命令的数据结构,无须关心DMA模块220接受的DMA命令的格式,无须对不同类型DMA命令的特定格式进行适配,简化了CPU 250访问DMA命令的接口,降低了CPU 250处理DMA命令时的负载。

[0183] 存储器230中存储的多个DMA命令可以被组织为队列、链表、线性表或数组等。

[0184] 作为一个实施例,存储器230中存储的多个DMA命令被组织为队列。在该实施例中,

CPU250与DMA加速器240之间通过单一流进行数据传输,存储器230中设有一个队列(单一队列)。处理单元502依据队列的写指针将DMA命令写入存储器230,处理单元503依据队列的读指针从存储器230获取DMA命令的执行结果,并依据DMA命令的执行结果被CPU 250接收而更新读完成指针。

[0185] 如图5所示,处理单元502维护读指针和写指针,处理单元502在由标记①②③指示的过程中,依据写指针指示的地址将DMA命令写入存储器230。接下来,处理单元502更新自身维护的写指针,使写指针指向更新后的队列的队尾。在由标记④⑤⑥指示的过程中,将更新后的写指针写入存储器230,以在存储器230中记录队尾的位置。处理单元502还通过由标记⑦⑧⑨指示的过程监视存储器230中的读完成指针(指示队列的队头位置),并在处理单元502内部记录读完成指针的最新值。通过维护读完成指针和写指针,处理单元502知晓存储器230中的队列是否未滿。在队列未滿时,处理单元502可向队尾添加DMA命令。

[0186] 继续参看图5,DMA模块220从存储器230获取DMA命令。DMA模块220依据存储器230中记录的写指针和读指针确定队列中是否存在被添加的DMA命令,并从队头(由读指针指示)获取DMA命令。DMA模块220依据取出的DMA命令在外部设备与电子设备之间发起DMA传输,并在DMA命令对应的DMA传输完成并将DMA命令的执行结果写入存储器230后,更新存储器230中的读指针,以指示DMA命令被DMA模块220处理完成。

[0187] 需要理解的,处理单元502操作的队列由写指针指示队尾,由读完成指针指示队头。以及由处理单元503操作的队列,由读指针指示队尾,由读完成指针指示队头。

[0188] 处理单元503监视存储器230中的读指针。响应于存储器230的读指针发生变化,或者存储器230的读指针与处理单元503记录的读完成指针不同,处理单元503知晓有新的DMA命令被DMA模块处理完成。在读完成指针与存储器230的读指针不同时,处理单元503依据自身记录的读完成指针从存储器230获取被处理完成的DMA命令的执行结果,并提供给DMA命令完成单元504。处理单元503还响应于将DMA命令的执行结果提供给DMA命令完成单元504,更新自身维护的读完成指针。

[0189] 作为一个实施例,处理单元503可以将读完成指针写入存储器230。处理单元502监视存储器230中的读完成指针,并将读完成指针作为队列队头的位置。处理单元502从存储器230获取读完成指针。

[0190] 作为另一个实施例,处理单元502与处理单元503耦合,处理单元502直接从处理单元503获取读完成指针,将读完成指针作为队列队头的位置,而无须从存储器230获取读完成指针。

[0191] 作为再一个实施例,DMA加速器240还包括指针管理器(请参见图11中的指针管理器1105),指针管理器分别与处理单元502和处理单元503耦合,处理单元502从指针管理器获取读完成指针以及向指针管理器更新写指针,处理单元503从指针管理器获取读指针和向指针管理器更新读完成指针。

[0192] 作为又一个实施例,处理单元502将写指针直接提供给DMA模块220,DMA模块220将读指针直接提供给处理单元503。

[0193] 图6到图10为根据本申请实施例一提供的单一队列的指针示意图。与该单一队列相关的指针包括读指针、写指针与读完成指针。存储器230中的队列以及与队列相关的指针可被DMA加速器240或DMA模块220访问。

[0194] 可选地,DMA加速器240与DMA模块220可维护与队列相关的指针的副本。

[0195] 图6展示了初始状态的队列与指针。队列包括16个条目(分别编号为0-15),可容纳16个DMA命令。可选地,DMA命令可具有相同或不同的大小。

[0196] 在初始状态,电子设备200上电或复位后,队列中未被写入任何DMA命令(队列为空),读指针、写指针与读完成指针均为0,指向队列的条目0。

[0197] DMA加速器240的处理单元502依据写指针向队列(图6中写指针为0)添加DMA命令,并在将DMA命令写入队列后,更新存储器230中的写指针。参看图7,队列的编号为0的条目被写入DMA命令,写指针被更新为1(指向编号为1的队列条目)。而读指针和读完成指针依然为0。

[0198] DMA模块220依据读指针落后于写指针,识别出队列被写入了DMA命令。DMA模块220依据读指针(图7中指向编号为0的队列条目)从队列中获取DMA命令并处理。

[0199] 可以理解地,DMA加速器240的处理单元502向队列添加DMA命令的过程,与DMA模块从队列获取DMA命令的过程可以并发,并且互相不会影响对方。

[0200] 参看图8,DMA模块220处理DMA命令的速度慢于DMA加速器240向队列中添加命令的速度,DMA加速器240的处理单元502持续向队列添加DMA命令,写指针已被更新为10(编号为0到9的队列条目均被写入DMA命令)。DMA模块220处理了队列中的编号为0到3的队列条目的DMA命令,读指针被更新为4。DMA模块220在完成对DMA命令的处理后,更新队列中的DMA命令,在DMA命令中记录DMA命令的执行结果。DMA模块220向队列的条目中写入DMA命令的执行结果。在图8中,队列中编号为0到3的队列条目中的DMA命令均被DMA模块220处理完成。

[0201] 可选地,在此过程中,读指针指示了DMA模块220向队列写入DMA命令的执行结果时的队列条目。DMA模块220完成向队列条目写入DMA命令的执行结果后,更新存储器230中的读指针。

[0202] 可选地,DMA模块220记录了DMA命令在队列中的位置,并在处理完成该DMA命令后依据所记录的位置将DMA命令的执行结果写入队列。

[0203] DMA加速器240的处理单元503从队列的条目中获取DMA命令的执行结果。DMA加速器240的处理单元503识别读指针超前于读完成指针,知晓队列中被DMA模块220写入了DMA命令的状态,而依据读完成指针所指示的条目获取DMA命令的执行结果。处理单元503从队列条目获取DMA命令的执行结果后,还更新读完成指针,使其指向队列的下一个条目。参看图8,DMA加速器240的处理单元503依据读完成指针获取了编号为0和1的队列条目的DMA命令的执行结果,并将读完成指针更新为指向编号为2的队列条目。

[0204] DMA加速器240的处理单元502继续向队列添加DMA命令。当编号为15的条目被写入DMA命令后,由于队列总共有16个条目,写指针发生回绕,并指向编号为0的条目(参看图9)。同时,DMA模块220继续处理队列中的DMA命令,并将DMA命令执行状态写入队列,并更新读指针(图9中指向编号为11的条目)。DMA加速器240的处理单元503继续从队列获取DMA命令的执行结果,并更新读完成指针(图9中指向编号为7的条目)。

[0205] 优选地,在写指针0中还记录其发生回绕,基于回绕标记或回绕发生的次数来识别写指针超前于读指针。

[0206] DMA加速器240的处理单元502依据写指针和读完成指针识别队列是否未滿。若读完成指针超前于写指针,意味着队列中充满了被添加但尚未被取走的DMA命令(可能已被

DMA模块220处理完成),从而DMA加速器240的处理单元502暂停向队列添加DMA命令,而等待DMA加速器240的处理单元503从队列取走DMA命令的执行结果。

[0207] DMA加速器240的处理单元503从队列获取DMA命令的执行结果的过程与DMA模块220向队列添加DMA命令的执行状态的过程可并行执行,而不会互相影响。

[0208] 请参见图10,DMA加速器240的处理单元503从队列获取DMA命令的执行结果的速度快于DMA模块220向队列添加DMA命令的执行结果的速度,从而读完成指针逐渐追上读指针。图10中,相对于图9,读完成指针和读指针均发生回绕,并且读完成指针和读指针指向相同的位置(编号为2的队列条目)。作为响应,DMA加速器240的处理单元503识别到DMA模块220尚未产生更新的DMA命令执行结果,因此DMA加速器240的处理单元503暂停从队列中获取DMA命令执行结果。

[0209] 在该队列的使用过程中,DMA加速器的处理单元502和DMA模块220是使用队列的一对生产者与消费者,DMA模块220和DMA加速器的处理单元504是使用队列的另一对生产者与消费者。单一队列既用于DMA加速器向DMA模块提交DMA命令,也用于DMA模块向DMA加速器提交DMA命令的执行结果,降低了对存储器的存储空间的需求。

[0210] 实施例二

[0211] 图11为本申请实施例二提供的直接存储器访问加速器的结构图。如图11所示,DMA加速器240包括DMA命令接收单元1101、处理单元1102、DMA命令完成单元1104和处理单元1103。

[0212] 如图11所示,DMA命令接收单元1101分别耦合到CPU 250和处理单元1102,DMA命令接收单元1101接收CPU 250提供的数据包。

[0213] 处理单元1102与存储器230耦合,处理单元1102从DMA命令接收单元1101获取CPU 250提供的数据包,将数据包转换为符合DMA模块220可接受的格式,形成DMA命令并写入存储器230,并维护DMA模块220可接受的数据结构(例如,队列)。

[0214] 处理单元1103与存储器230耦合,处理单元1103从存储器230获取DMA命令执行结果。DMA命令完成单元1104分别耦合到处理单元1103和CPU 250,DMA命令完成单元1104从处理单元1103获取DMA命令的执行结果,并向CPU 250提供DMA命令的执行结果。

[0215] 实施例二与实施例一的区别在于:DMA加速器240与CPU 250之间通过多个相互独立的流进行数据传输,存储器230中设有与各个流对应的队列。DMA模块独立处理CPU 250在各个流中提供的DMA命令。

[0216] 作为一个例子,DMA命令接收单元1101设有端口(单一端口,如流式写入接口)。DMA命令接收单元1101的流式写入接口为每个流提供可用状态标记,CPU 250可独立或联合地获取每个流的可用状态标记。该端口依据流的可用状态而接收CPU 250的数据包,数据包中标记要访问的流。DMA命令接收单元1101依据数据包中的标记识别DMA命令所属的流。处理单元1102依据DMA命令所属的流,获取与该流对应的队列的指针(即DMA命令在存储器230中的存储地址),并将DMA命令写入与该流对应的队列。

[0217] 处理单元1103监视各个流对应的队列的指针。在队列中出现处理完成的DMA命令的执行状态后,处理单元1103获取DMA命令的执行结果,并通过DMA命令完成单元1104提供给CPU 250。具体地,DMA命令完成单元1104的流式读出接口为每个流提供可用状态标记,CPU 250可独立或联合地获取每个流的可用状态标记,并通过流式读出接口从各个流获取

DMA命令的执行结果。

[0218] DMA加速器240确保CPU 250提交的DMA命令所属的流与CPU 250获取的DMA命令的执行结果所属的流是相同的流。即使CPU 250同时向多个流提交DMA命令,但是CPU 250从各个流获取的DMA命令的执行结果出现于与CPU 250提交的DMA命令相同的流中。例如,DMA加速器240提供4个流(S0、S1、S2与S3),CPU 250向流S1提交了DMA命令C1和C2,向流S2提交了DMA命令C3和C4,而CPU 250相应地从DMA命令完成单元1104提供的流S1中获取到DMA命令C1和C2的执行结果,而从DMA命令完成单元1104提供的流S2中获取到DMA命令C3和C4的执行结果。

[0219] 可选地,在同一流中,DMA命令的执行结果按DMA命令被提交到流的顺序提供给CPU 250。同样可选地,在同一流中,DMA命令的执行结果提供给CPU 250的顺序可以是乱序的。

[0220] 作为另一个具体实施例,DMA命令接收单元1101设有多个端口(如流式接口或先入先出接口),端口的数量与流的数量相同,端口与流一一对应。DMA命令接收单元1101通过与流一一对应的端口接收CPU 250的数据包。DMA命令接收单元1101依据接收数据包端口识别DMA命令所属的流。处理单元1102依据DMA命令所属的流,获取与该流对应的队列的指针(即DMA命令在存储器230中的存储地址),并将DMA命令写入与该流对应的队列。DMA命令完成单元1104依据DMA命令所属的流从与该流对应的端口将DMA命令执行结果提供给CPU 250。

[0221] 作为又一个实施例,数据包指示标识符(sID,也称为流标识符),流标识符指示了数据包所属的流。DMA加速器240依据标识符确定DMA命令在存储器230中的存储地址。

[0222] 作为一个实施例,DMA加速器240还包括指针管理器1105。指针管理器1105分别与处理单元1102和处理单元1103耦合,用于记录存储器230中各个队列的读指针、写指针和读完成指针(例如,用于流标识符为S0的流的多个指针、用于流标识符为S1的流的多个指针)。处理单元1102向指针管理器1105更新与每个流对应的队列的写指针,处理单元1103从指针管理器1105获取与每个流对应的队列的读指针和向指针管理器1105更新与每个流对应的队列的读完成指针。指针管理器用于对存储器中的多个队列进行管理,并且用于处理单元1102和处理单元1103交换指针。处理单元1102与处理单元1103依据流标识符从指针管理器1105获取或向指针管理器1105写入同流对应的指针。

[0223] 可选地,存储器230为每个流对应的队列和与队列对应的指针提供存储空间。图11中展示了存储器230容纳两个队列,以及两个队列的每个的读指针与写指针。流与存储器230中的队列一一对应。

[0224] 实施例三

[0225] 图12为本申请实施例三提供的DMA命令处理方法的流程图。如图12所示,DMA加速器对DMA命令的处理方法包括如下步骤:

[0226] 步骤1201:接收来自CPU的数据包。

[0227] 步骤1202:将数据包转换为DMA命令,并写入存储器。

[0228] 步骤1203:响应于存储器中的DMA命令的执行结果被更新,获取被更新的DMA命令的执行结果。

[0229] 步骤1204:将DMA命令的执行结果提供给CPU。

[0230] 作为一个实施例,CPU与DMA加速器之间通过流式接口进行数据传输,DMA加速器为

每个流提供状态标记。

[0231] 步骤1201中,DMA加速器提供流式接口的状态标记,该状态标记指示流式接口是否可以接收数据。若CPU识别出流式接口的状态标记为可用状态标记,则CPU向流式接口发送数据包。

[0232] 步骤1204中,DMA加速器提供流式接口的状态标记,该状态标记指示流式接口是否存在需要输出的数据。若CPU识别出流式接口的状态标记为可用状态标记,则CPU从流式接口读取数据包。

[0233] 作为另一个实施例,CPU与DMA加速器之间通过先入先出接口(FIFO接口)进行数据传输。

[0234] 步骤1201中,DMA加速器提供先入先出接口的状态。若CPU识别出先入先出队列未滿,则CPU向先入先出接口发送数据包。

[0235] 步骤1204中,DMA加速器提供先入先出接口的状态。若CPU识别出先入先出队列为非空,则CPU从先入先出接口读取数据包。

[0236] 可选地,存储器中存储的多个DMA命令被组织为队列。通过写指针、读指针和读完成指针来维护队列。写指针指向存储器中存储DMA命令的队列的队尾,读完成指针指向存储器中存储DMA命令的队列的队头。

[0237] DMA加速器依据读完成指针落后于写指针,确定存储器中存储DMA命令的队列未滿。仅在存储器中的队列未滿时,DMA加速器依据写指针将DMA命令写入存储器,然后更新写指针。

[0238] DMA模块依据读指针从存储器中的队列获取DMA命令并进行处理。响应于DMA命令被完成,DMA模块将DMA命令的执行结果写入存储器,并更新存储器中的读指针。

[0239] 响应于读指针与读完成指针不同,DMA加速器从存储器获取DMA命令的执行结果。响应于读指针超前于读完成指针,DMA加速器依据读完成指针从存储器获取DMA命令的执行结果,并更新读完成指针。

[0240] 可选地,DMA加速器可以将读完成指针写入存储器,并且监视存储器中的读完成指针,并将读完成指针作为队列队头的位置。

[0241] 作为一个实施例,DMA加速器与CPU之间通过多个相互独立的流进行数据传输,存储器中设有与各个流一一对应的队列。DMA模块独立处理CPU在各个流中提供的DMA命令。DMA加速器为每个流提供状态标记。

[0242] 图13为本申请实施例的CPU通过直接存储器访问加速器执行直接存储器访问操作的流程图。如图13所示,CPU通过DMA加速器执行DMA操作包括如下步骤:

[0243] 步骤1310:CPU从DMA加速器中获取可用的流。例如,CPU访问DMA加速器提供的流式接口是否可用的状态标记,以获取可用的流。该状态标记指示对应的流是否可以写入数据或读出数据。

[0244] 若CPU要向DMA加速器提供DMA命令,则执行步骤1320:向可用的用于写入DMA命令的流发送数据包。

[0245] 若CPU要从DMA加速器获取DMA命令的执行结果,则执行步骤1330:从可用的用于读出数据的流获取DMA命令的执行结果。

[0246] 通过使用流式接口,CPU将数据写入到流,无须关心数据的存储地址和数据结构,

从而减轻了CPU的负载。在读出数据时,CPU通过使用流式接口从流中获取数据,无须关心数据的存储地址和数据结构。尽管流式接口提供了可访问地址,但是该可访问地址为单一地址或指定地址,即使访问多份数据,CPU也无须处理地址的更新,也无须进行内存管理。

[0247] 本申请的技术方案获得的有益效果如下:

[0248] (1) 本申请的DMA加速器替代CPU与存储器进行交互,替代CPU生成符合DMA模块所需的格式、传输长度等要求的DMA命令,并替代CPU维护存储器中的DMA命令的数据结构,将DMA命令写入存储器,并监视存储器中DMA命令的状态更新,以及最后将DMA命令的执行结果提供给CPU,大大减轻了CPU的负载,加快了DMA命令的处理速度。

[0249] (2) 本申请通过使用流式接口,CPU将数据写入到流,无须关心数据的存储地址和数据结构,DMA加速器将CPU提供的数据组织为DMA命令,从而减轻了CPU的负载;在读出数据时,DMA加速器将DMA命令的执行结构组织为数据流,CPU通过使用流式接口从流中获取数据,无须关心数据的存储地址和数据结构;尽管流式接口提供了可访问地址,但是该可访问地址为单一地址或指定地址,CPU无须处理地址的更新,也无须进行内存管理。

[0250] (3) 本申请通过对存储器中队列的维护,使得DMA加速器的第一处理单元和DMA模块形成队列的一对生产者与消费者,DMA模块和DMA加速器的第二处理单元形成队列的另一对生产者与消费者,单一队列既用于DMA加速器向DMA模块提交DMA命令,也用于DMA模块向DMA加速器提交DMA命令的执行结果,降低了对存储器的存储空间的需求。

[0251] 以上所述,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以所述权利要求的保护范围为准。

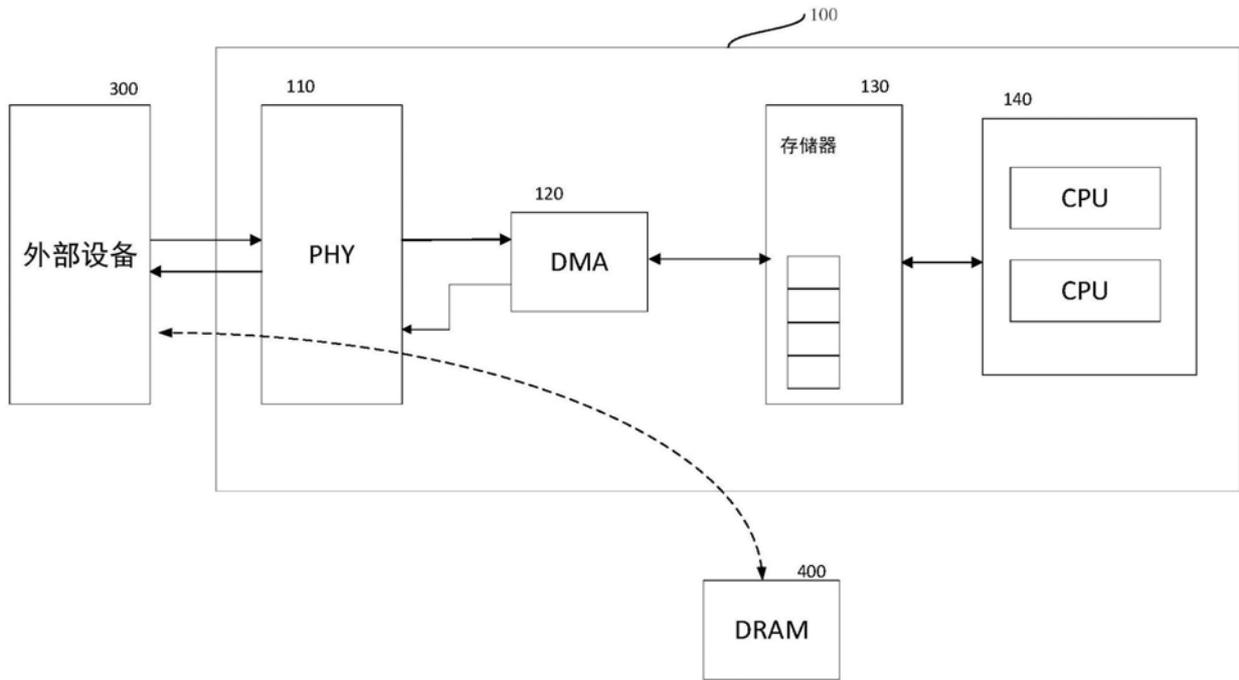


图1

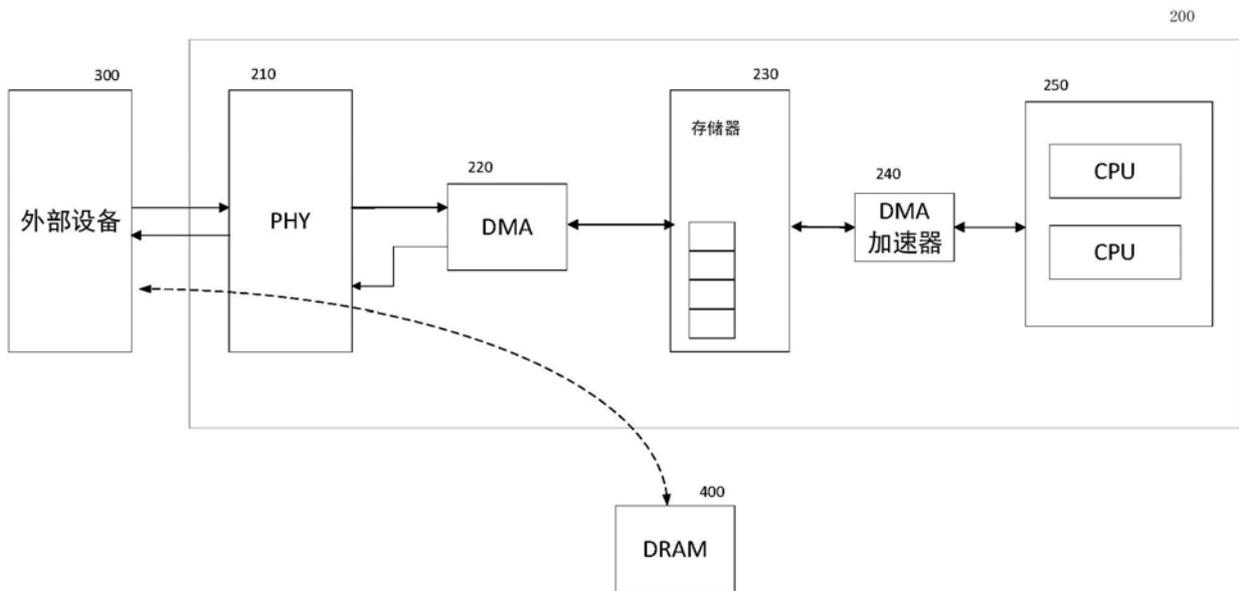


图2

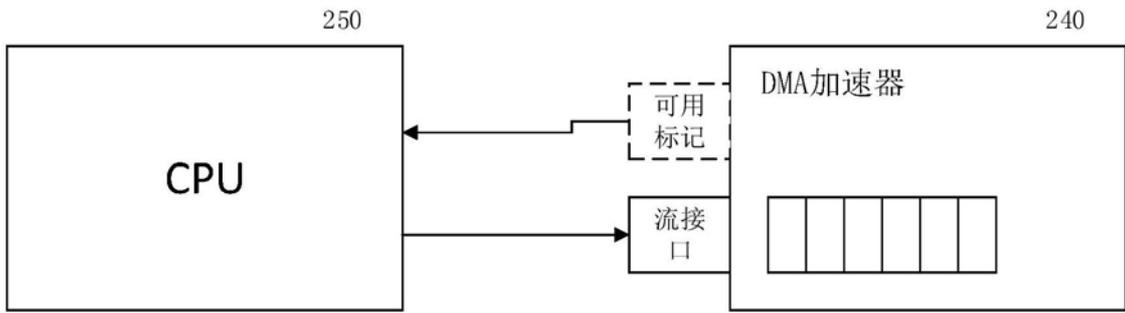


图3

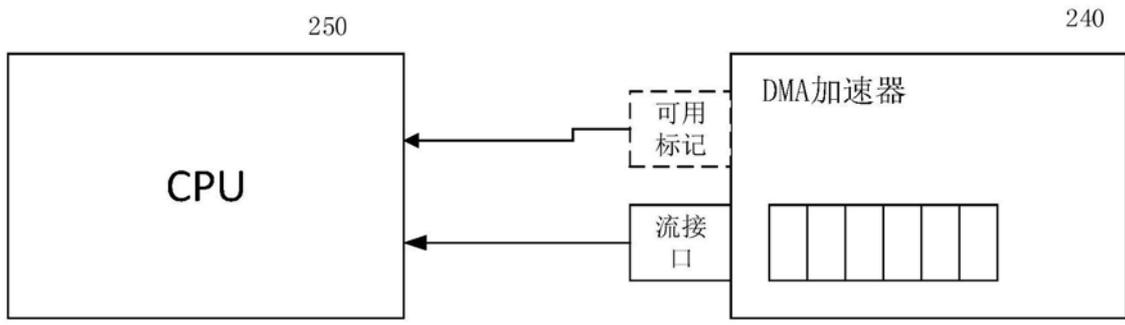


图4

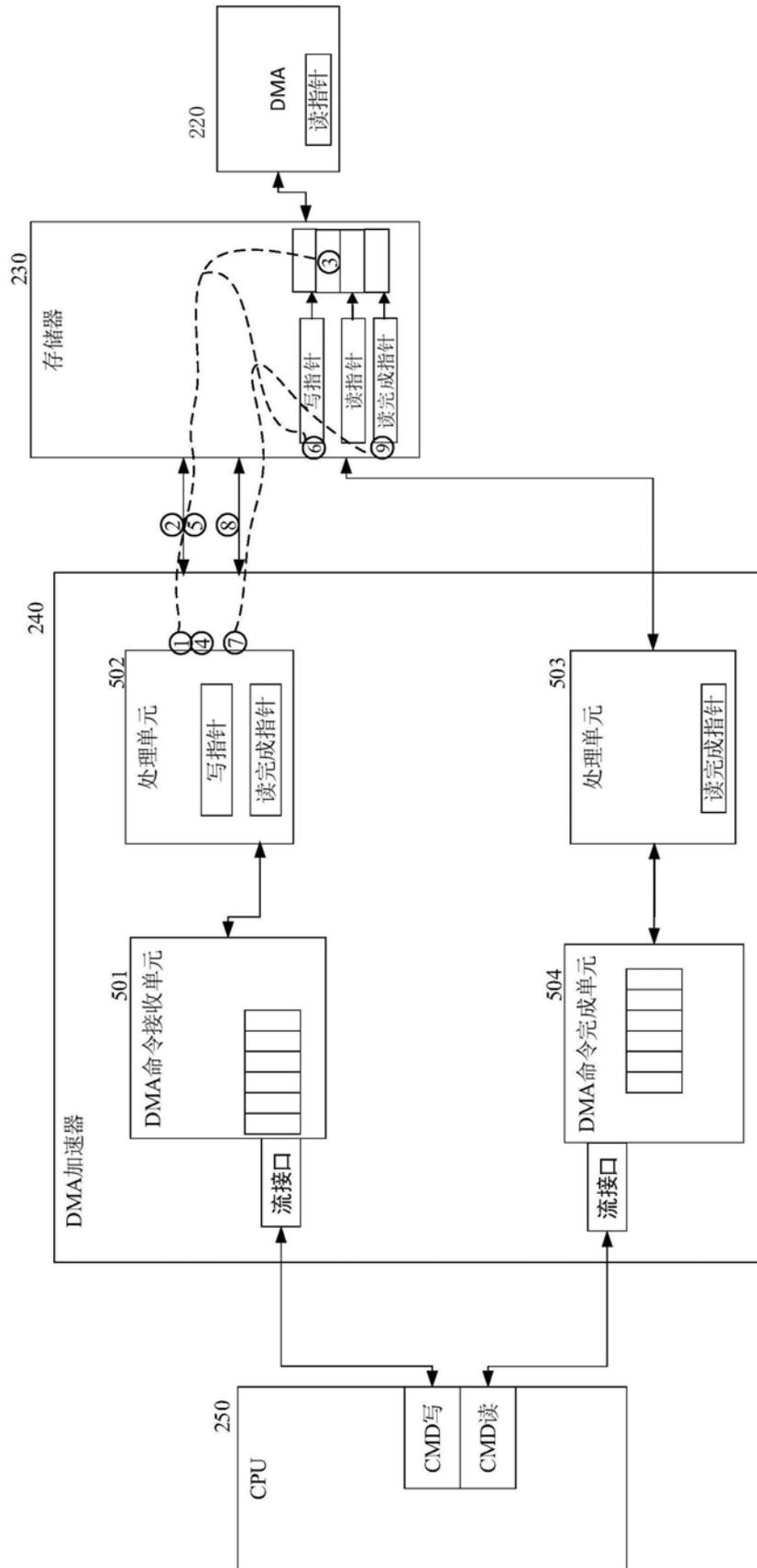


图5

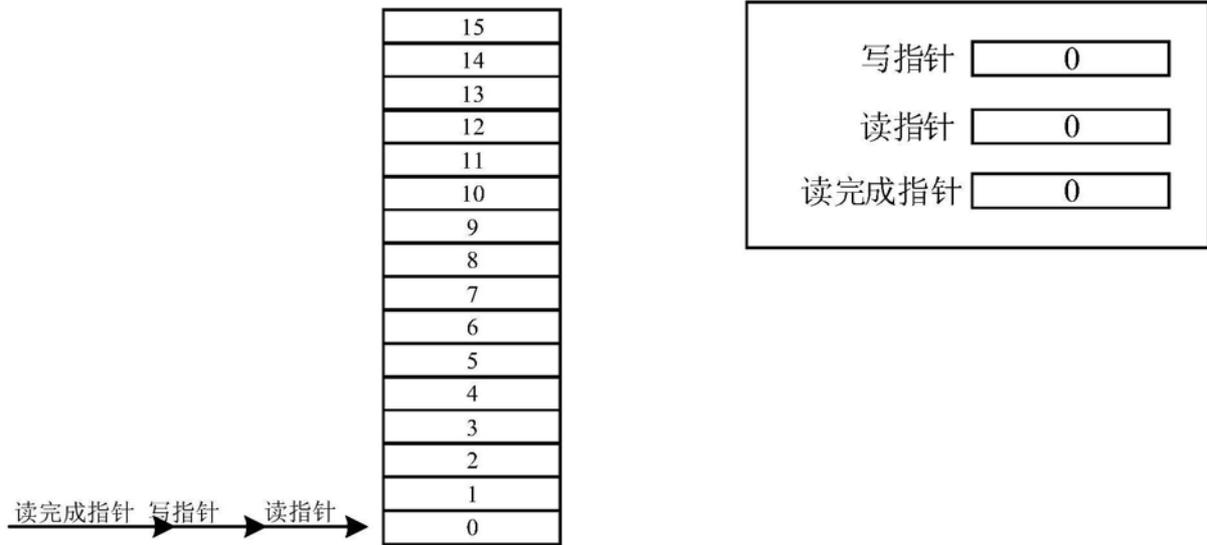


图6

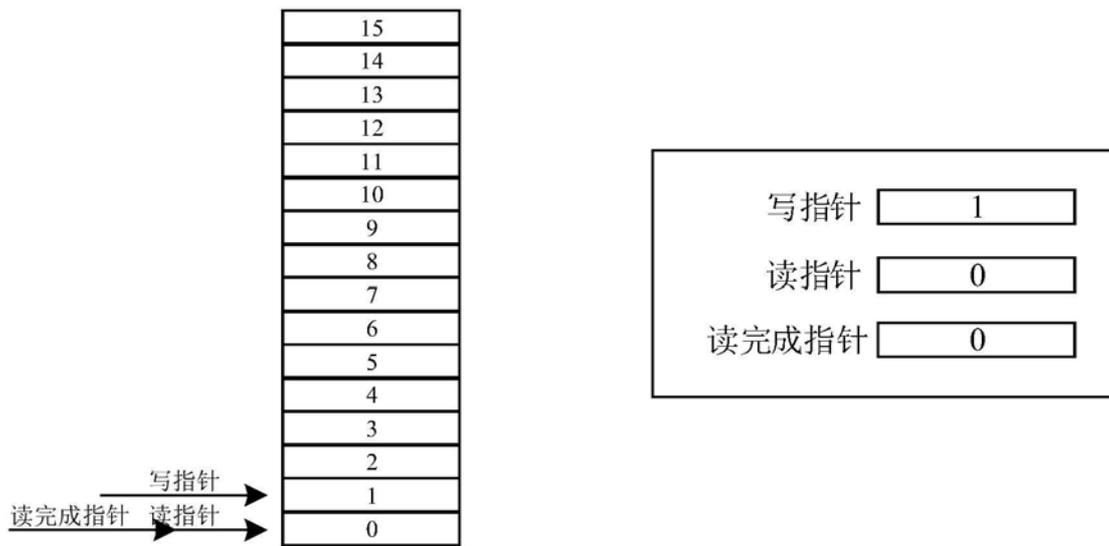


图7

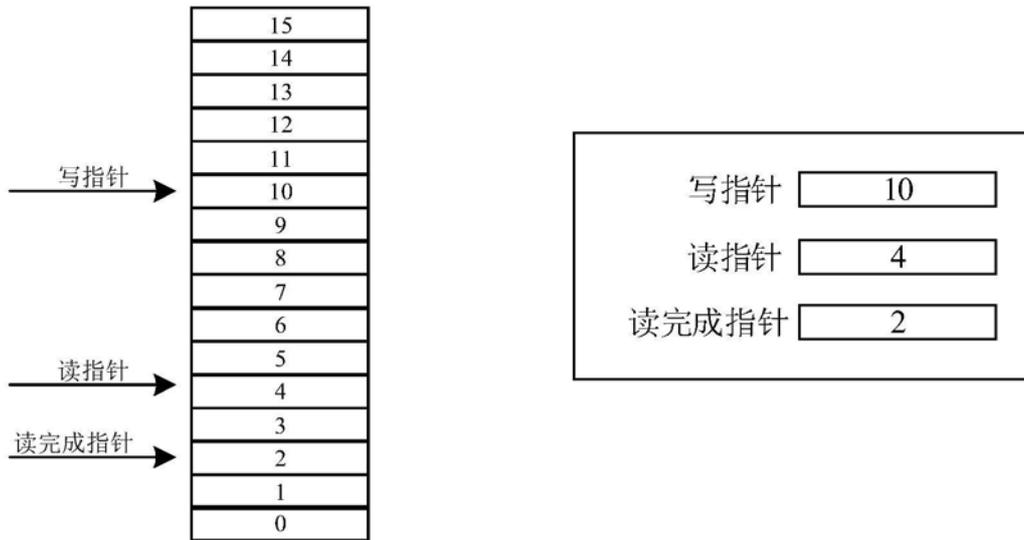


图8

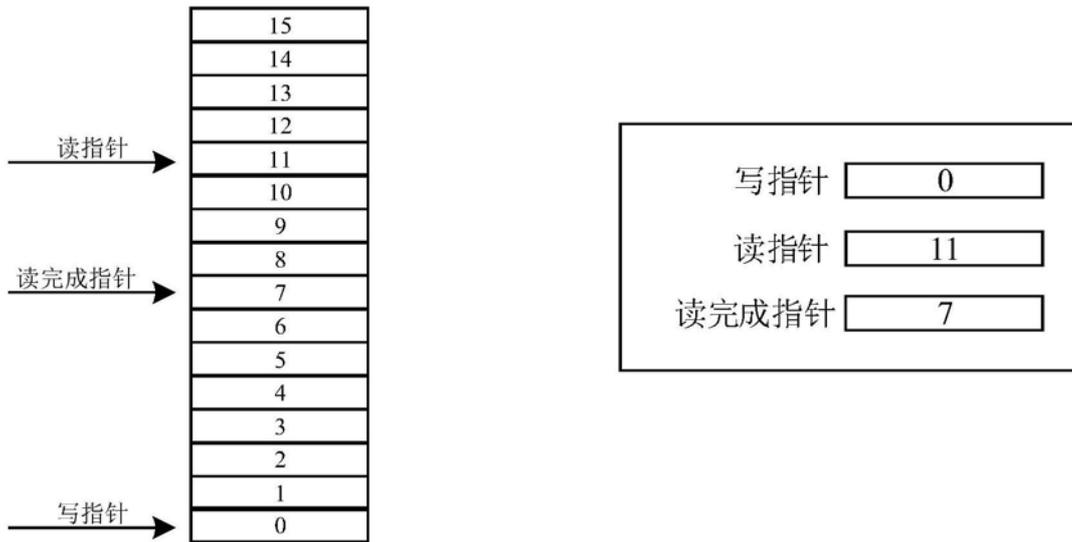


图9

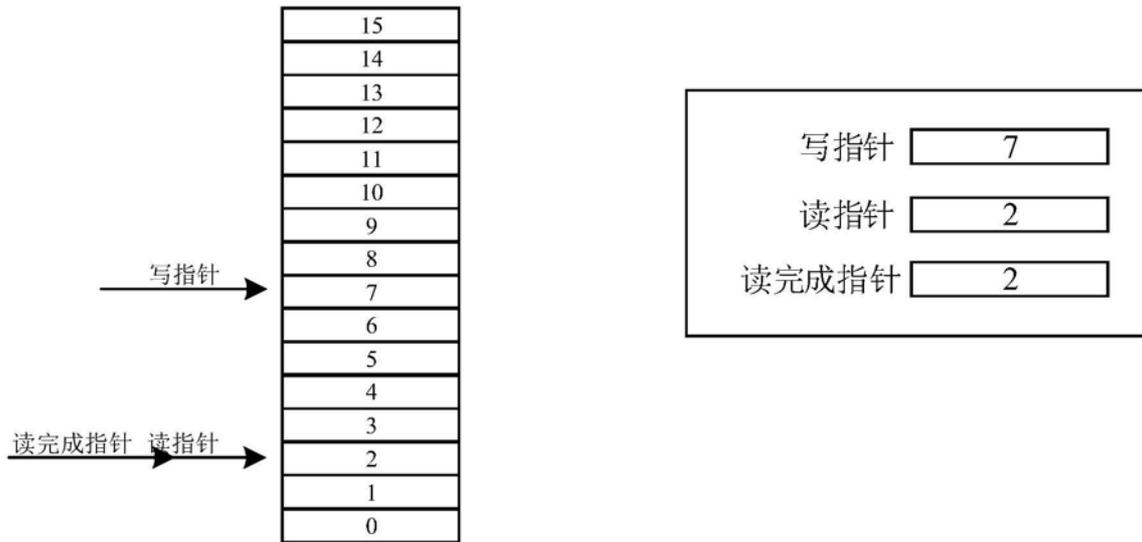


图10

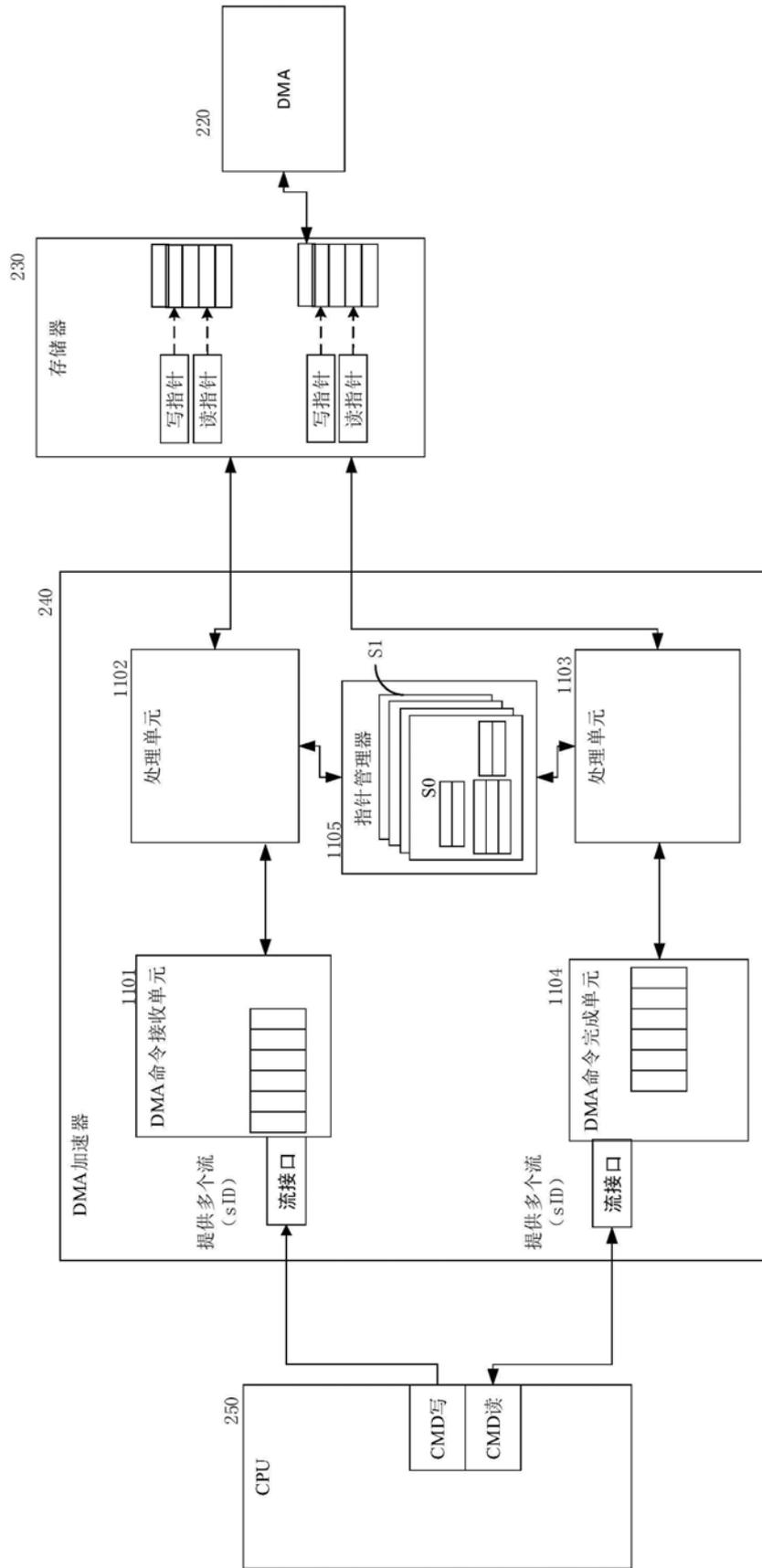


图11

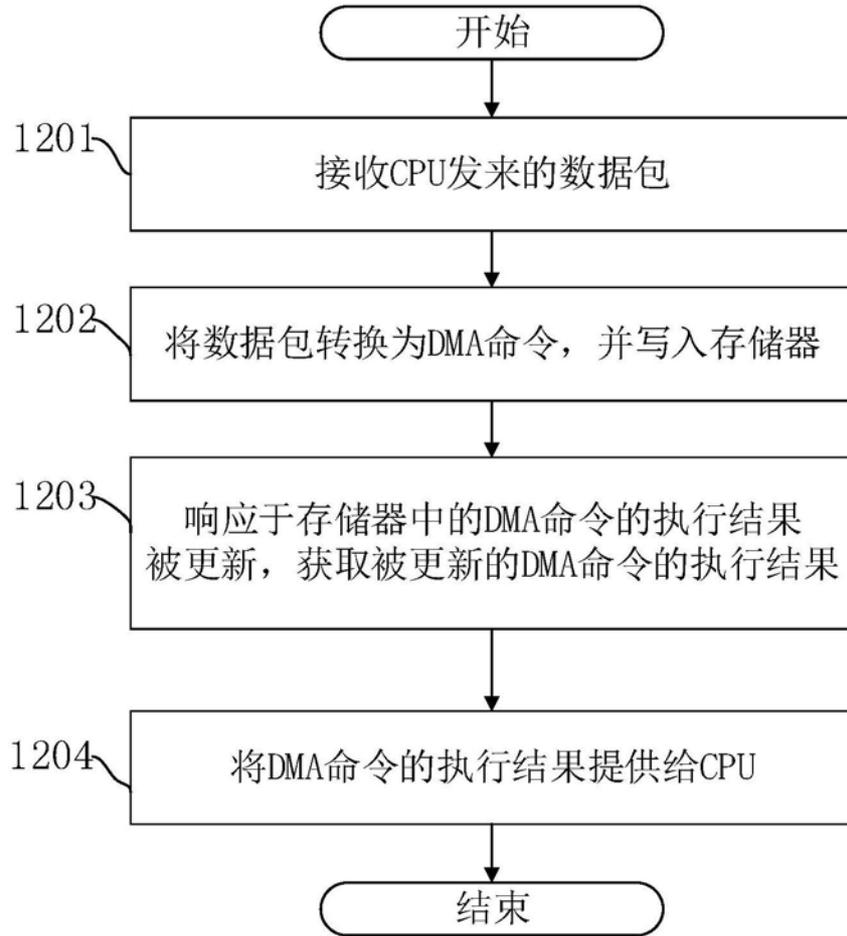


图12

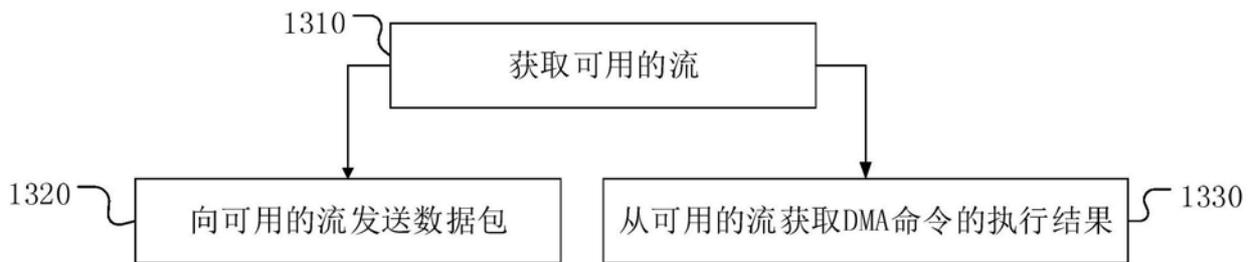


图13