



(12) 发明专利

(10) 授权公告号 CN 109213691 B

(45) 授权公告日 2023. 09. 01

(21) 申请号 201710526281.8

(22) 申请日 2017.06.30

(65) 同一申请的已公布的文献号
申请公布号 CN 109213691 A

(43) 申请公布日 2019.01.15

(73) 专利权人 伊姆西IP控股有限责任公司
地址 美国马萨诸塞州

(72) 发明人 杨利锋 贾瑞勇 李雄成 高宏坡
徐鑫磊

(74) 专利代理机构 北京市金杜律师事务所
11256
专利代理师 王茂华

(51) Int. Cl.
G06F 12/0846 (2016.01)
G06F 12/0871 (2016.01)

(56) 对比文件

CN 104503710 A, 2015.04.08

CN 106575271 A, 2017.04.19

CN 102567220 A, 2012.07.11

CN 103365793 A, 2013.10.23

CN 105912479 A, 2016.08.31

US 2016179681 A1, 2016.06.23

US 2003159001 A1, 2003.08.21

US 2013046933 A1, 2013.02.21

US 2012179874 A1, 2012.07.12

US 2017094011 A1, 2017.03.30

秦秀磊等. 云计算环境下分布式缓存技术的现状与挑战.《软件学报》.2013, 50-66.

审查员 陈治帆

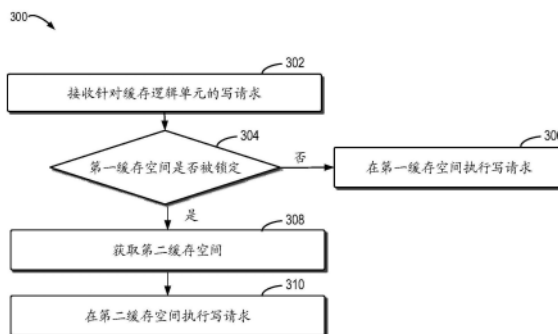
权利要求书2页 说明书7页 附图6页

(54) 发明名称

用于缓存管理的方法和设备

(57) 摘要

本公开的实施例涉及用于缓存管理的方法和设备。该方法包括：响应于接收针对缓存逻辑单元的写请求，确定与该缓存逻辑单元相关联的多个缓存空间中的第一缓存空间是否被锁定；响应于确定该第一缓存空间被锁定，获取该多个缓存空间中的第二缓存空间，该第二缓存空间不同于该第一缓存空间并且处于未锁定状态；以及在该第二缓存空间执行针对该缓存逻辑单元的该写请求。



1. 一种用于缓存管理的方法,所述方法包括:

响应于接收针对缓存逻辑单元的写请求,确定与所述缓存逻辑单元相关联的多个缓存空间中的第一缓存空间是否被锁定;

响应于确定所述第一缓存空间没有被锁定,在所述第一缓存空间中执行针对所述缓存逻辑单元的所述写请求;

响应于确定所述第一缓存空间被锁定,获取所述多个缓存空间中的第二缓存空间,所述第二缓存空间不同于所述第一缓存空间并且处于未锁定状态;并且在所述第二缓存空间中执行针对所述缓存逻辑单元的所述写请求;以及

响应于后续接收针对所述缓存逻辑单元的读请求,确定所述读请求的目标空间是否在所述第二缓存空间内,并且响应于所述目标空间在所述第二缓存空间内,将所述写请求所写入的数据返回到所述第二缓存空间中作为对所述读请求的响应,并且响应于所述目标空间的至少一部分位于所述第二缓存空间外部,合并所述第一缓存空间和所述第二缓存空间的数据并且返回经合并的所述数据作为对所述读请求的响应。

2. 根据权利要求1所述的方法,其中获取所述第二缓存空间包括:

记录所述写请求的标识以及所述第二缓存空间的地址。

3. 根据权利要求2所述的方法,其中所述写请求的所述标识指示接收所述写请求的时间戳。

4. 根据权利要求2所述的方法,还包括:

响应于接收到缓存冲刷请求,合并与所述缓存逻辑单元相关联的多个缓存空间中的数据;以及

将经合并的所述数据写入到与所述缓存逻辑单元相关联的后端存储器地址空间。

5. 根据权利要求4所述的方法,还包括:

在完成将经合并的所述数据写入到与所述缓存逻辑单元相关联的所述后端存储器地址空间后,释放所述写请求的所述标识以及所述第二缓存空间的所述地址的记录。

6. 根据权利要求1所述的方法,其中所述多个缓存空间各自包括多个缓存页面,并且所述合并包括:

针对所述多个缓存空间中的对应的缓存页面,确定最后被写入的有效数据作为合并后的数据。

7. 一种用于缓存管理的设备,包括:

至少一个处理单元;

至少一个存储器,所述至少一个存储器被耦合到所述至少一个处理单元并且存储用于由所述至少一个处理单元执行的指令,所述指令当由所述至少一个处理单元执行时,使得所述设备执行动作,所述动作包括:

响应于接收针对缓存逻辑单元的写请求,确定与所述缓存逻辑单元相关联的多个缓存空间中的第一缓存空间是否被锁定;

响应于确定所述第一缓存空间没有被锁定,在所述第一缓存空间中执行针对所述缓存逻辑单元的所述写请求;

响应于确定所述第一缓存空间被锁定,获取所述多个缓存空间中的第二缓存空间,所述第二缓存空间不同于所述第一缓存空间并且处于未锁定状态;并且在所述第二缓存空间

中执行针对所述缓存逻辑单元的所述写请求;以及

响应于后续接收针对所述缓存逻辑单元的读请求,确定所述读请求的目标空间是否在所述第二缓存空间内,并且响应于所述目标空间在所述第二缓存空间内,将所述写请求所写入的数据返回到所述第二缓存空间中作为对所述读请求的响应,并且响应于所述目标空间的至少一部分位于所述第二缓存空间外部,合并所述第一缓存空间和所述第二缓存空间的数据并且返回经合并的所述数据作为对所述读请求的响应。

8. 根据权利要求7所述的设备,其中获取所述第二缓存空间包括:

记录所述写请求的标识以及所述第二缓存空间的地址。

9. 根据权利要求8所述的设备,其中所述写请求的所述标识指示接收所述写请求的时间戳。

10. 根据权利要求8所述的设备,所述动作还包括:

响应于接收到缓存冲刷请求,合并与所述缓存逻辑单元相关联的多个缓存空间中的数据;以及

将经合并的所述数据写入到与所述缓存逻辑单元相关联的后端存储器地址空间。

11. 根据权利要求10所述的设备,所述动作还包括:

在完成将经合并的所述数据写入到与所述缓存逻辑单元相关联的所述后端存储器地址空间后,释放所述写请求的所述标识以及所述第二缓存空间的所述地址的记录。

12. 根据权利要求7所述的设备,其中所述多个缓存空间各自包括多个缓存页面,并且所述合并包括:

针对所述多个缓存空间中的对应的缓存页面,确定最后被写入的有效数据作为合并后的数据。

13. 一种非瞬态计算机存储介质,所述非瞬态计算机存储介质上存储计算机程序,所述计算机程序包括机器可执行指令,所述机器可执行指令在设备中运行时使所述设备执行根据权利要求1-6中的任一项所述的方法。

用于缓存管理的方法和设备

技术领域

[0001] 本公开的实施例涉及数据存储领域,并且更具体地,涉及缓存管理的方法、设备及计算机程序产品。

背景技术

[0002] 随着数据存储技术的发展,各种数据存储设备已经能够向用户提供越来越高的数据存储能力,并且数据访问速度也有了很大程度的提高。在提高数据存储能力的同时,用户对于数据可靠性和存储系统的响应时间也提出了越来越高的需求。

[0003] 目前,已经开发出了基于具有不同访问速度的多级存储介质来建立存储系统的技术方案。存储系统可能接收多个并行的I/O请求,当存在多个并行的I/O请求指向同一物理存储地址时,缓存I/O冲突就可能会发生。也即,某个I/O请求在访问缓存时,会锁定其访问的缓存页面,使得其他I/O请求需要等待前一I/O请求的执行完成,从而影响缓存的效率。因而,如何有效地提高针对缓存的I/O请求之间的并发度成为一个关注焦点。

发明内容

[0004] 本公开的实施例提供一种用于缓存管理的方案。

[0005] 根据本公开的第一方面,提出了一种用于缓存管理的方法。该方法包括:响应于接收针对缓存逻辑单元的写请求,确定与该缓存逻辑单元相关联的多个缓存空间中的第一缓存空间是否被锁定;响应于确定该第一缓存空间被锁定,获取该多个缓存空间中的第二缓存空间,该第二缓存空间不同于该第一缓存空间并且处于未锁定状态;以及在该第二缓存空间执行针对该缓存逻辑单元的该写请求。

[0006] 根据本公开的第二方面,提出了一种用于缓存管理的设备。该设备包括:至少一个处理单元;至少一个存储器,该至少一个存储器被耦合到该至少一个处理单元并且存储用于由该至少一个处理单元执行的指令,该指令当由该至少一个处理单元执行时,使得该设备执行动作,该动作包括:响应于接收针对缓存逻辑单元的写请求,确定与该缓存逻辑单元相关联的多个缓存空间中的第一缓存空间是否被锁定;响应于确定该第一缓存空间被锁定,获取该多个缓存空间中的第二缓存空间,该第二缓存空间不同于该第一缓存空间并且处于未锁定状态;以及在该第二缓存空间执行针对该缓存逻辑单元的该写请求。

[0007] 在本公开的第三方面,提供了一种计算机程序产品。该计算机程序产品被存储在非瞬态计算机存储介质中并且包括机器可执行指令,该机器可执行指令在设备中运行时使该设备执行根据本公开的第一方面所描述的方法的任意步骤。

附图说明

[0008] 通过结合附图对本公开示例性实施例进行更详细的描述,本公开的上述以及其它目的、特征和优势将变得更加明显,其中,在本公开示例性实施例中,相同的参考标号通常代表相同部件。

- [0009] 图1示出了缓存管理系统架构的示意图；
- [0010] 图2示出了传统缓存管理的示意图；
- [0011] 图3示出了根据本公开的实施例的用于缓存管理的方法的流程图；
- [0012] 图4示出了根据本公开的实施例的缓存管理的示意图；
- [0013] 图5示出了根据本公开的实施例的处理读请求的方法的流程图；
- [0014] 图6示出了根据本公开的实施例的合并多个缓存空间数据的示意图；
- [0015] 图7示出了根据本公开的实施例的处理冲刷请求的方法的流程图；以及
- [0016] 图8示出了可以用来实施本公开内容的实施例的示例设备的示意性框图。

具体实施方式

[0017] 下面将参照附图更详细地描述本公开的示例实施例。虽然附图中显示了本公开的优选实施例，然而应该理解，可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反，提供这些实施例是为了使本公开更加透彻和完整，并且能够将本公开的范围完整地传达给本领域的技术人员。

[0018] 在本文中使用的术语“包括”及其变形表示开放性包括，即“包括但不限于”。除非特别申明，术语“或”表示“和/或”。术语“基于”表示“至少部分地基于”。术语“一个示例实施例”和“一个实施例”表示“至少一个示例实施例”。术语“另一实施例”表示“至少一个另外的实施例”。术语“第一”、“第二”等等可以指代不同的或相同的对象。下文还可能包括其他明确的和隐含的定义。

[0019] 如上文所述，传统的缓存管理提供较低的I/O请求并发度。图1示出了缓存管理系统架构100的示意图。如图1所示，在该架构中，通过缓存逻辑单元管理器130来控制缓存响应所接收的各种I/O请求（诸如读I/O 112、写I/O 114、填零I/O 116、预取I/O、移动数据I/O等），并通过缓存逻辑单元管理器130经由各种后端I/O 140-148实现缓存110与后端存储器150之间交互。

[0020] 缓存逻辑单元管理器130可以包括缓存逻辑单元132、缓存页面134和映射表136。缓存逻辑单元132是缓存模块中的最小管理单元，每个缓存逻辑单元132可以对应后端存储器150中的具体物理地址空间。每个缓存逻辑单元132可以包括多个缓存页面134。例如，在一些实施例中，单个缓存逻辑单元可以为64Kb，其可以包括8个缓存页面（每个为8Kb）。此外，每个缓存逻辑单元132与其对应的多个缓存页面134之间的映射关系被存储在映射表136中。存储管理器130可以通过查询映射表136获得缓存逻辑单元132所对应的多个缓存页面134。

[0021] 图2示出了传统方案中的缓存管理的示意图200。如图2所示，缓存210中存在第一缓存逻辑单元212以及第二缓存逻辑单元216，第一缓存逻辑单元212与第二缓存逻辑单元216分别与第一缓存页集合214和第二缓存页集合218相关联。第一写请求220执行对缓存逻辑单元212的写入时，添加对缓存逻辑单元212的第一锁定222。而当接收到后续的针对缓存逻辑单元212的第二写请求224时，由于缓存逻辑单元212处于第一锁定222而无法被第二写请求224所写入。因此第二写请求224将被加入队列，并添加对缓存逻辑单元212的第二锁定226。在第一写请求220完成后，第一锁定222将被解除，此时第二写请求224将被执行。以这种方式，针对同一缓存逻辑单元212的多个写请求只能够顺序地执行，从而影响缓存的

效率。

[0022] 为了至少部分地解决上述问题以及其他潜在问题,本公开的实施例提供了一种缓存管理的方案。根据本公开的各种示例实施例,单个缓存逻辑单元将与多个不同的缓存空间相关联,在接收针对该缓存逻辑单元的多个写请求时,每个写请求可以被并行地写入到与同一个缓存逻辑单元相关联的不同的缓存空间,从而提高缓存写请求之间的并发度。并且,在接收针对该缓存逻辑单元的读请求时,将判断读请求与最后一次写请求目标空间之间的关系确定是否执行对与缓存逻辑单元相关联的多个缓存空间之间的数据合并。此外,系统还可以响应于冲刷请求,执行与缓存逻辑单元相关联的多个缓存空间数据之间的合并,并将经合并的数据写入到后端存储器中。

[0023] 基于这样的缓存管理方法,缓存系统可以支持多个写请求之间的并行执行,提供对读请求及时且准确的响应,并确保写入到后端存储器数据的准确性。从而解决了现有技术中针对同一缓存逻辑单元的多个I/O请求无法并发执行的问题,能够极大地提高针对缓存的多个I/O请求之间的并发程度,改善存储系统的效率。

[0024] 下面将参考图3-图4详细描述根据本公开的缓存管理的方法。图3示出了根据本公开的实施例的用于缓存管理的方法300的流程图,图4示出了根据本公开的实施例的缓存管理的示意图400。

[0025] 在框302中,接收针对缓存逻辑单元的写请求。在框304,判断与缓存逻辑单元相关联的第一缓存空间是否被锁定。若第一缓存空间未被锁定,则方法进行到框306,即在第一缓存空间执行写请求。如图4所示,系统接收到针对缓存逻辑单元410的第一写请求422和第二写请求432。对于第一写请求422,与缓存逻辑单元412相关联的第一缓存空间420在第一写请求422被执行前未被锁定,因此可以在第一缓存空间420中执行第一写请求422。若第一缓存空间被锁定,则方法300进行到框308。

[0026] 在框308,获取与缓存逻辑单元相关联的第二缓存空间。在一些实施例中,单个缓存逻辑单元可以预先与多个缓存空间相关联。例如,单个缓存逻辑单元可以与128个缓存空间相关联,进而可以支持最多128个针对该缓存逻辑单元的写请求的并发执行。

[0027] 在一些实施例中,可以通过缓存空间列表管理与缓存逻辑单元关联的多个缓存空间。每当获取一个新的缓存空间用于写请求时,可以在缓存空间列表中添加一个新的节点以代表该缓存空间已被使用,节点中可以记录该写请求的标识以及该缓存空间的地址。在一些实施例中,可以通过链表的方式实现该缓存空间列表,并在获取一个新的缓存空间时,添加一个新的节点到该缓存空间链表中。

[0028] 具体地,如图4所示,对于第二写请求432,由于第一缓存空间420被锁定,系统可以获取与缓存逻辑单元412关联的第二缓存空间430,并可以在缓存空间列表中添加第二节点,该第二节点可以记录第二写请求的标识以及第二缓存空间420的地址。在一些实施例中,写请求的标识还可以指示接收到该写请求的时间戳。诸如,第一写请求422的标识可以为2056,而第二写请求432的标识可以为3106,其分别指示了接收第一写请求422和第二写请求432的时间戳,并体现了第二写请求432晚于第一写请求422所接收。

[0029] 在框310,在第二缓存空间中执行写请求。具体地,如图4所示,在第二缓存空间430执行对第二写请求432的写入。通过这样的方式,如图4所示,可以支持针对单个缓存逻辑单元412的多个写请求(第一写请求422、第二写请求432等)的并发执行,进而提高了存储系统

的响应效率。

[0030] 上文已经参考图3和图4描述了缓存并行写机制的若干实施例。根据本公开的实施例,系统在处理多个并发写请求时不需要再锁定多个写请求所针对的缓存逻辑单元,系统还可以并发地执行针对该缓存逻辑单元的读请求。下面将参考图5-图6详细描述根据本公开的并发执行读请求的过程。图5示出了根据本公开的实施例的处理读请求的方法500的流程图,图6示出了根据本公开的实施例的合并多个缓存空间数据的示意图600。

[0031] 在框502,接收针对缓存逻辑单元的读请求。具体地,如图6所示,系统分别接收针对缓存逻辑单元的第一读请求660以及第二读请求680,其中缓存逻辑单元包括128个缓存空间,第一缓存空间620以及第二缓存空间640中被写入有数据。在框504中,判断读请求的目标空间是否在最近一次写请求的目标空间内。若是,则方法进行到框506,即可以直接返回最近一次写请求所写入的数据。具体地,如图6所示,对于第一读请求660,其目标空间为缓存逻辑单元中的后三个缓存页面662-666,第一读请求660的目标空间位于最近一次写请求(即缓存空间640所对应的写请求)的目标空间(即缓存逻辑单元的后6个缓存页面646-656)内。因此可以返回第二缓存空间640中的缓存页面652-656的数据作为第一读请求660的响应。

[0032] 另一方面,如果在框504中确定读请求的目标目标空间的部分位于最近一次写请求的目标空间外,则方法500进行到框508。在框508,合并与缓存逻辑单元关联的多个缓存空间中的数据。在一些实施例中,可以针对缓存空间中的对应的缓存页面,确定最后被写入的有效数据作为合并后的数据。这样,系统可以快速地返回读请求所请求的数据。

[0033] 如图6所示,第二读请求的目标空间(在此例中,是缓存逻辑单元的缓存页面682-694)中的缓存页面682位于最近一次写请求(即,缓存空间640所对应的写请求)的目标空间(在此例中,是缓存逻辑单元的缓存页面646-656)外,因此需要对缓存逻辑单元的多个缓存空间中的数据进行合并操作以保证返回数据的准确性。

[0034] 在图6中,缓存逻辑单元包括已被写入有数据的第一缓存空间620及第二缓存空间640。在第一缓存空间620中,缓存页622-628被写入了有效数据,而在第二缓存空间640中,缓存页646-656被写入了有效数据。因此针对缓存页面622-624及缓存页面642-644,第一缓存空间620中缓存页面622-624中的有效数据将作为经合并的数据。对于缓存页面626-628及缓存页面646-648,第一缓存空间620和第二缓存空间640都被写入了有效数据,而由于第二缓存空间640所对应的写请求新于第一缓存空间620所对应的写请求。因此第二缓存空间640中的缓存页646-648中的数据将作为经合并的数据。

[0035] 对于缓存页面630-636以及缓存页面650-656,第一缓存空间620的缓存页面630-636中不包括有效数据,因此第二缓存空间640中的缓存页面650-656中的数据将作为经合并的数据。由此,该缓存逻辑单元最终合并后的数据为第一缓存空间620中的缓存页面622-624中的数据以及第二缓存空间640中的缓存页面646-656中的数据。如上文所述,在缓存空间列表的节点中记录了各缓存空间所对应的写请求的标识,该标识指示了接收到该请求的时间戳,因此,在一些实施例中,可以基于写请求的标识判断各缓存空间所对应的写请求的先后顺序以执行合并。

[0036] 在框508,返回经合并的数据作为读请求的响应。如图6所示,针对第二读请求680,将返回第一缓存空间620中的缓存页面624以及第二缓存空间640中的缓存页面646-656中

的数据作为第二读请求680的响应。基于以上所描述的方式，系统可以在并发地执行多个写请求的同时执行读请求，并确保所返回数据的准确性。

[0037] 在缓存响应前端I/O请求的同时，缓存还需要将前端I/O请求所写入到缓存的数据最终写入到后端存储器中，这一过程被称之为“冲刷”(flush)。基于常规缓存架构的冲刷操作通常需要锁定相应的缓存逻辑单元，从而同样影响缓存对于前端I/O请求的响应。而基于根据本公开的缓存架构，可以实现异步的冲刷过程。以下将结合图7具体描述基于本公开的缓存架构的冲刷机制。图7示出了根据本公开的实施例的处理冲刷请求的方法700的流程图。

[0038] 在框702，接收缓存冲刷请求以将缓存逻辑单元中的数据冲刷至后端存储器。在一些实施例中，系统可以定时地发出冲刷请求，在响应冲刷请求的同时系统仍可以继续执行前端I/O请求，即将缓存数据冲刷到后端存储器的过程异步于所接收的I/O请求的执行。

[0039] 在框704中，响应于接收冲刷缓存冲刷请求以合并缓存逻辑单元所关联的多个缓存空间中的数据。该合并过程与上文所述的处理读请求方法中的合并过程相同，固在此不再详细描述。在该过程中，由于缓存逻辑单元可能仍具有可用的缓存空间，因此其可以在执行合并的同时并发地响应前端写请求，进而使得前端I/O请求与后端冲刷请求不再冲突，提高了存储系统的效率。

[0040] 在框706，将经合并的数据写入到后端存储器中与缓存逻辑单元相关联的物理地址空间中。

[0041] 在框708，当缓存逻辑单元中的数据被写入到后端存储器后，缓存逻辑单元对应的多个缓存空间可以释放以被再次使用。应当理解的是，该释放过程只是使得缓存空间可以被重复写入，而并非被释放到存储空间，缓存逻辑单元仍保持与该多个缓存空间之间的关联。在一些实施例中，可以通过释放与多个缓存空间相关联的缓存空间列表中的节点来实现缓存空间的释放，也即释放节点中所记录的写请求的标识以及缓存空间的地址。

[0042] 基于这样的异步冲刷机制，前端I/O请求和后端冲刷请求可以被异步地执行，进一步地提高了响应效率，同时基于上文所述的合并机制还保证了被写入到后端存储器数据的准确性。

[0043] 图8示出了可以用来实施本公开内容的实施例的示例设备800的示意性框图。如图所示，设备800包括中央处理单元(CPU)801，其可以根据存储在只读存储器(ROM)802中的计算机程序指令或者从存储单元808加载到随机访问存储器(RAM)803中的计算机程序指令，来执行各种适当的动作和处理。在RAM 803中，还可存储设备800和/或设备800操作所需的各种程序和数据。CPU 801、ROM 802以及RAM 803通过总线804彼此相连。输入/输出(I/O)接口805也连接至总线804。

[0044] 设备800中的多个部件连接至I/O接口805，包括：输入单元806，例如键盘、鼠标等；输出单元807，例如各种类型的显示器、扬声器等；存储单元808，例如磁盘、光盘等；以及通信单元809，例如网卡、调制解调器、无线通信收发机等。通信单元809允许设备800通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0045] 上文所描述的各个过程和处理，例如方法300，可由处理单元801执行。例如，在一些实施例中，方法300可被实现为计算机软件程序，其被有形地包含于机器可读介质，例如存储单元808。在一些实施例中，计算机程序的部分或者全部可以经由ROM 802和/或通信单

元808而被载入和/或安装到设备800上。当计算机程序被加载到RAM 803并由CPU 801执行时,可以执行上文描述的方法300的一个或多个动作。

[0046] 本公开可以是方法、装置、系统和/或计算机程序产品。计算机程序产品可以包括计算机可读存储介质,其上载有用于执行本公开的各个方面的计算机可读程序指令。

[0047] 计算机可读存储介质是可以保持和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质例如可以是一—但不限于——电存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或者上述的任意合适的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式压缩盘只读存储器(CD-ROM)、数字多功能盘(DVD)、记忆棒、软盘、机械编码设备、例如其上存储有指令的打孔卡或凹槽内凸起结构、以及上述的任意合适的组合。这里所使用的计算机可读存储介质不被解释为瞬时信号本身,诸如无线电波或者其他自由传播的电磁波、通过波导或其他传输媒介传播的电磁波(例如,通过光纤电缆的光脉冲)、或者通过电线传输的电信号。

[0048] 这里所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各个计算/处理设备,或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光纤传输、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或者网络接口从网络接收计算机可读程序指令,并转发该计算机可读程序指令,以供存储在各个计算/处理设备中的计算机可读存储介质中。

[0049] 用于执行本公开操作的计算机程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、或者以一种或多种编程语言的任意组合编写的源代码或目标代码,所述编程语言包括面向对象的编程语言—诸如Smalltalk、C++等,以及常规的过程式编程语言—诸如“C”语言或类似的编程语言。计算机可读程序指令可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络—包括局域网(LAN)或广域网(WAN)—连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。在一些实施例中,通过利用计算机可读程序指令的状态信息来个性化定制电子电路,例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA),该电子电路可以执行计算机可读程序指令,从而实现本公开的各个方面。

[0050] 这里参照根据本公开实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述了本公开的各个方面。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机可读程序指令实现。

[0051] 这些计算机可读程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理单元,从而生产出一种机器,使得这些指令在通过计算机或其它可编程数据处理装置的处理单元执行时,产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中,这

些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作,从而,存储有指令的计算机可读介质则包括一个制品,其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。

[0052] 也可以把计算机可读程序指令加载到计算机、其它可编程数据处理装置、或其它设备上,使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤,以产生计算机实现的过程,从而使得在计算机、其它可编程数据处理装置、或其它设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

[0053] 附图中的流程图和框图显示了根据本公开的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分,所述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或动作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0054] 以上已经描述了本公开的各实施方式,上述说明是示例性的,并非穷尽性的,并且也不限于所披露的各实施方式。在不偏离所说明的各实施方式的范围和精神的情况下,对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择,旨在最好地解释各实施方式的原理、实际应用或对市场中的技术的改进,或者使本技术领域的其他普通技术人员能理解本文披露的各实施方式。

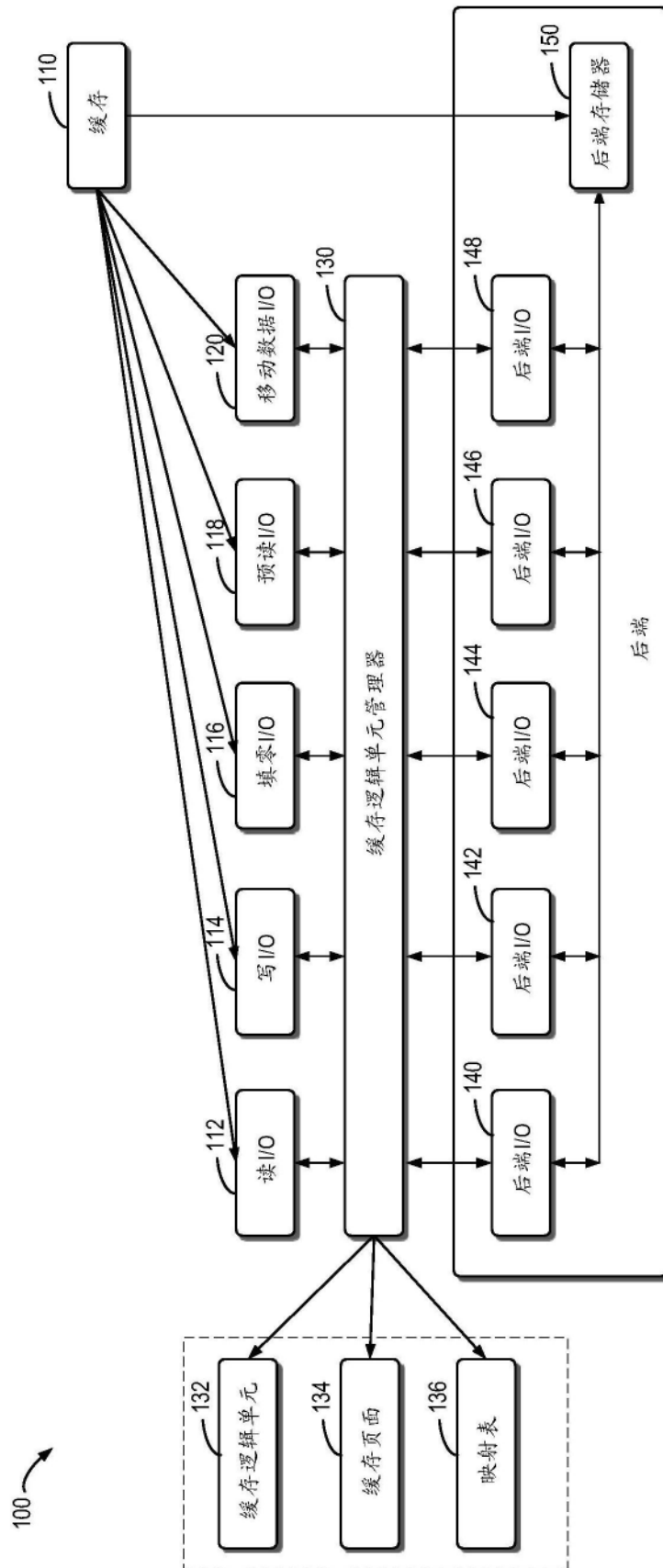


图1

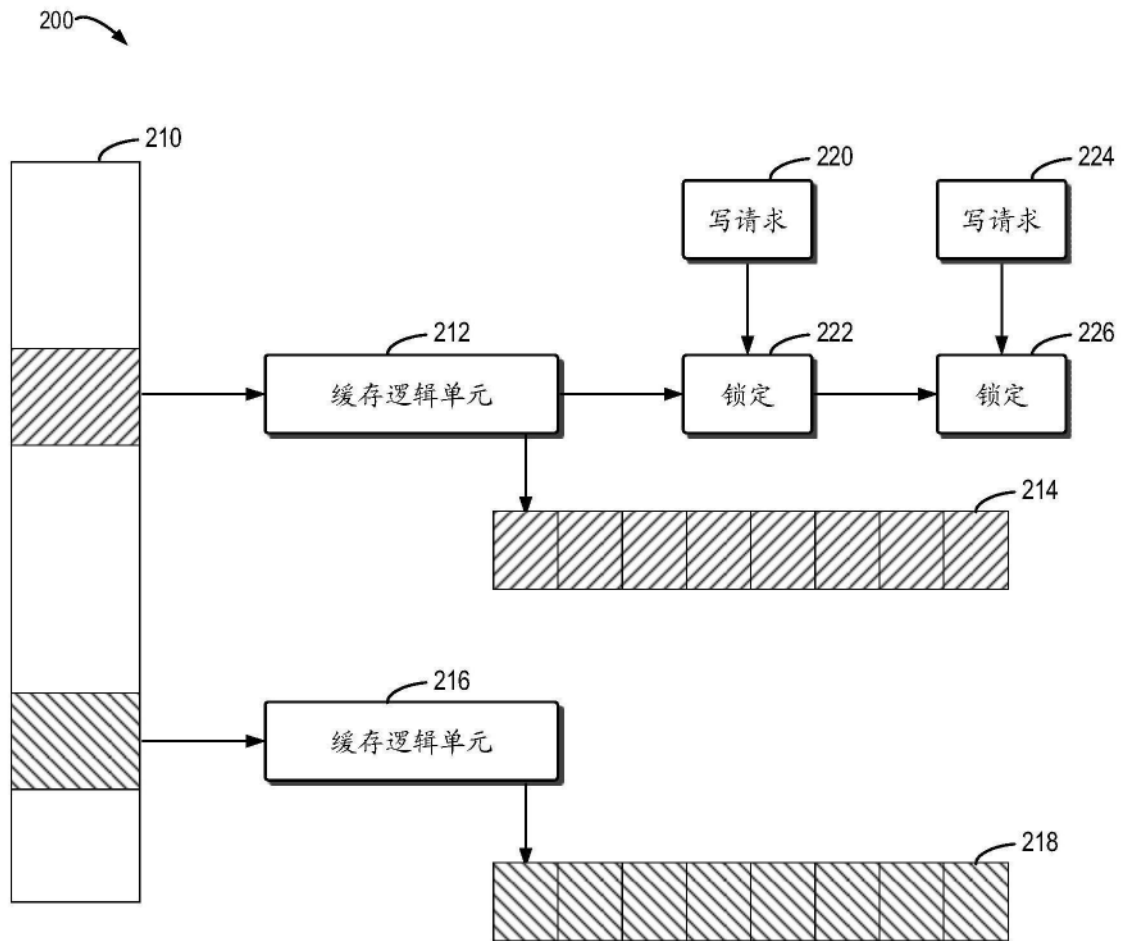


图2

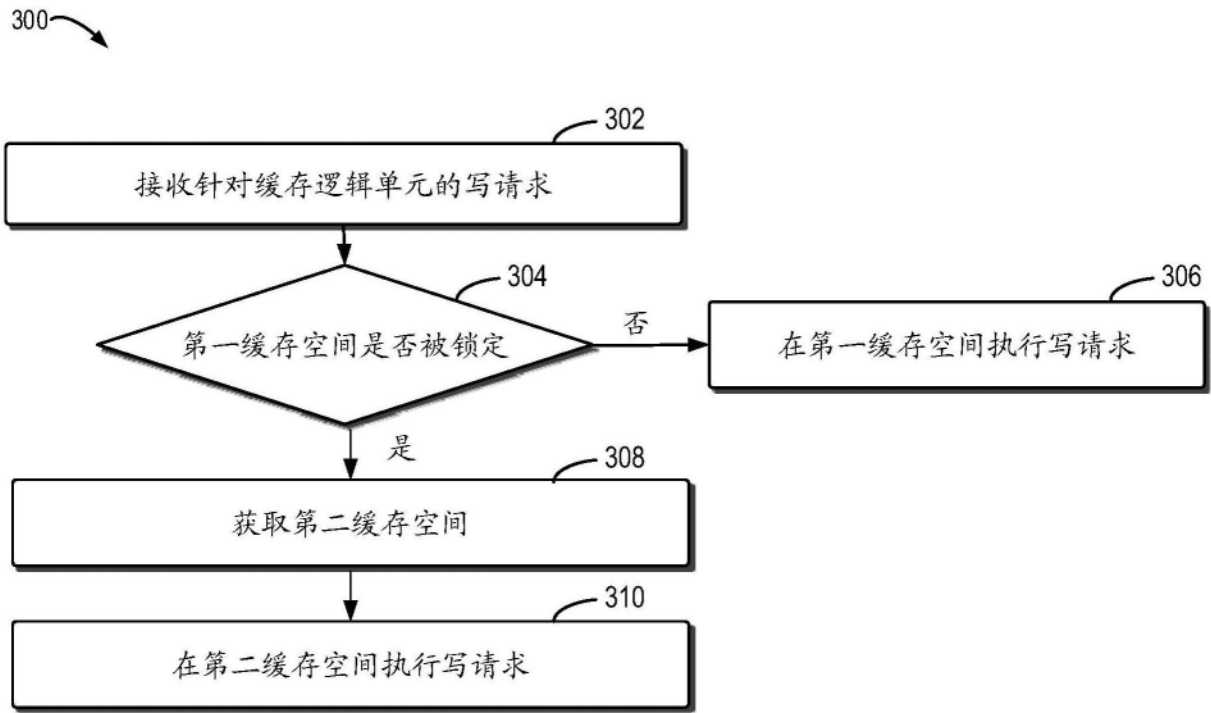


图3

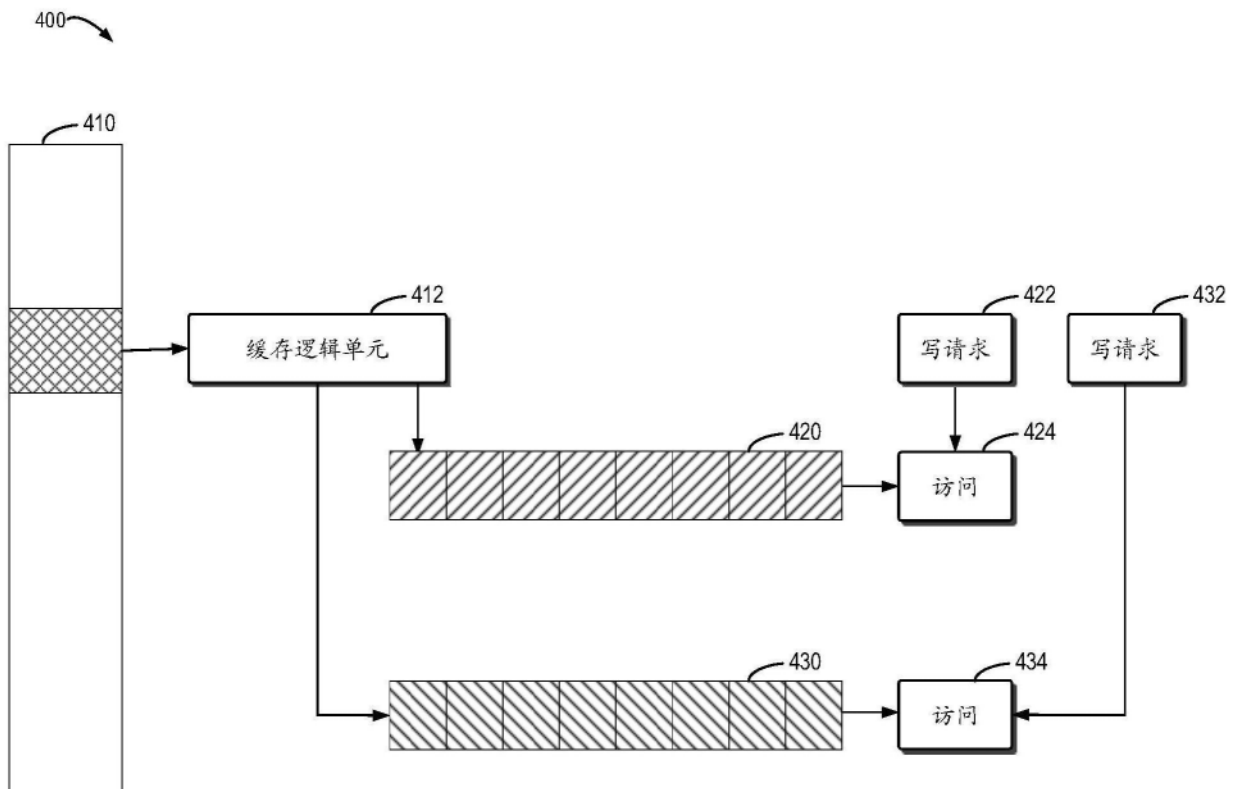


图4

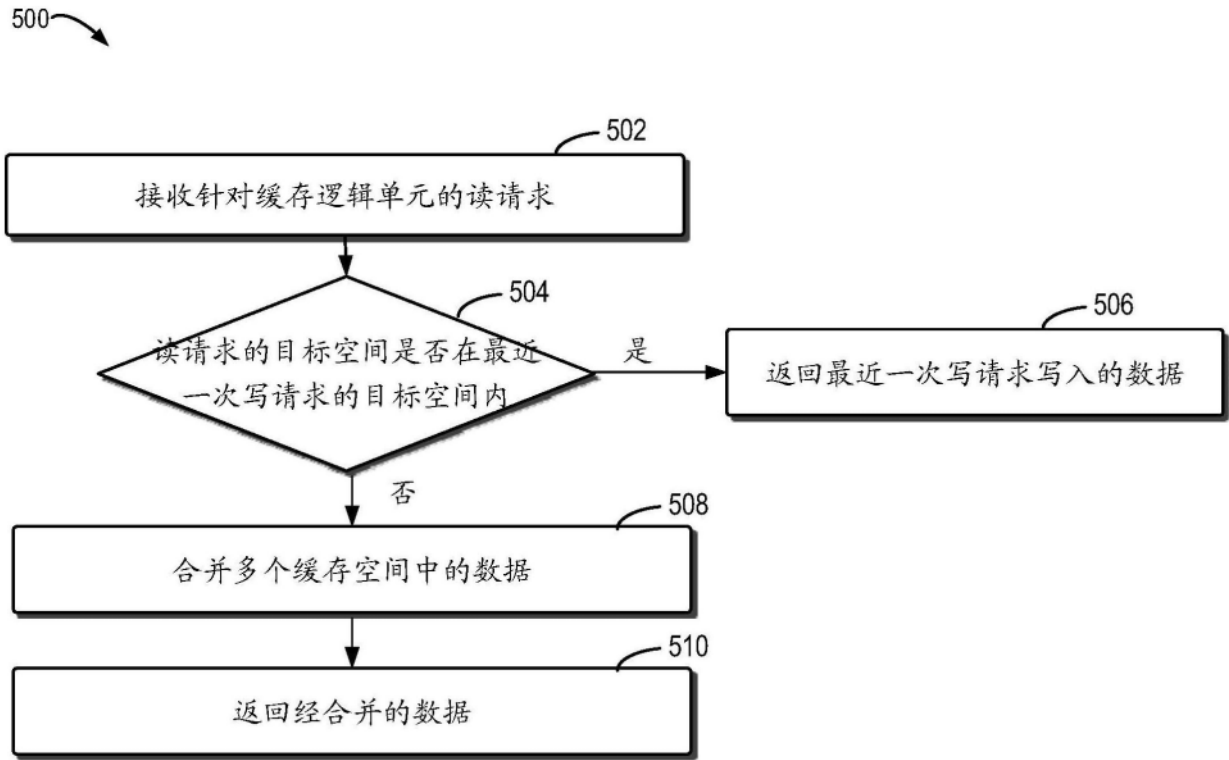


图5

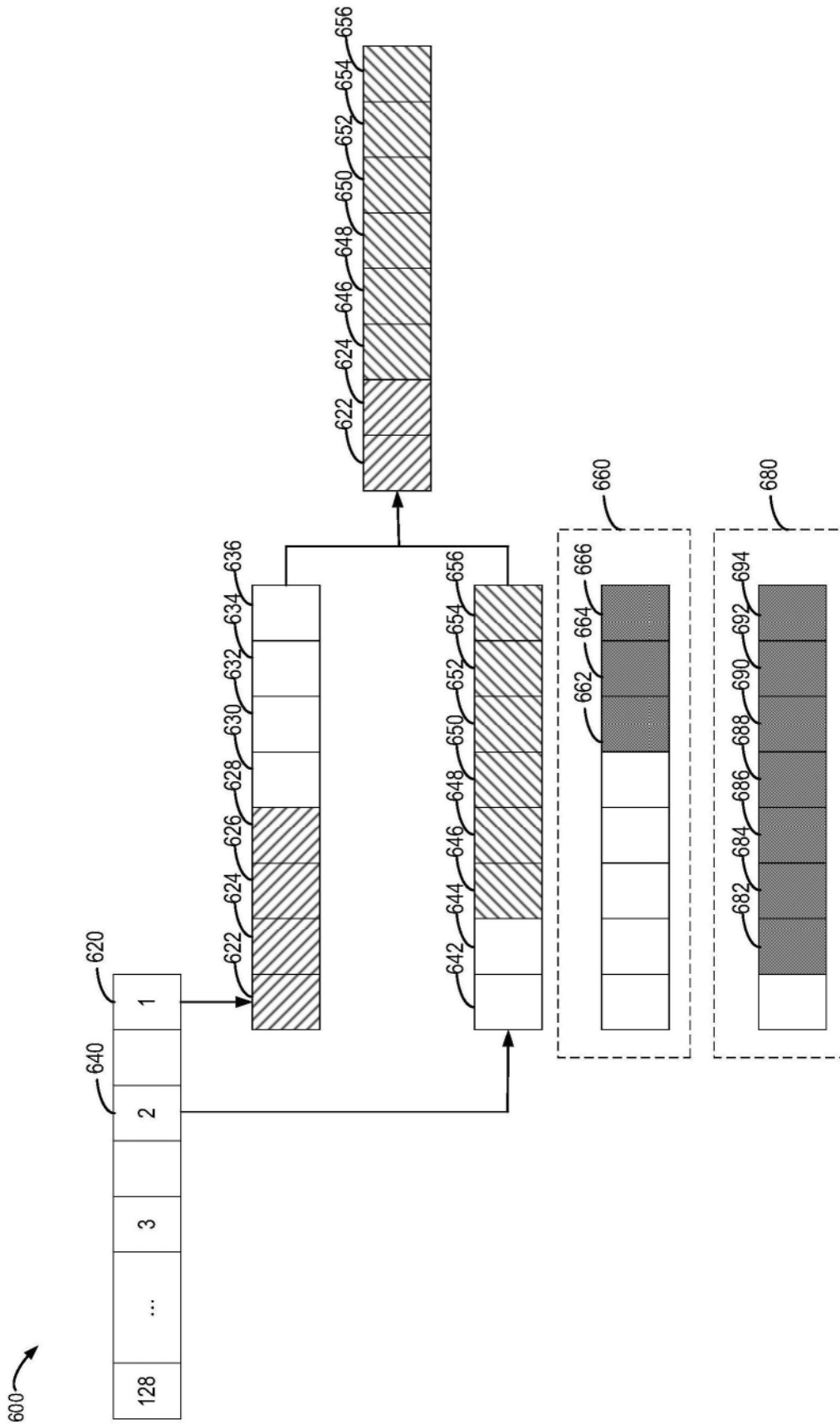


图6

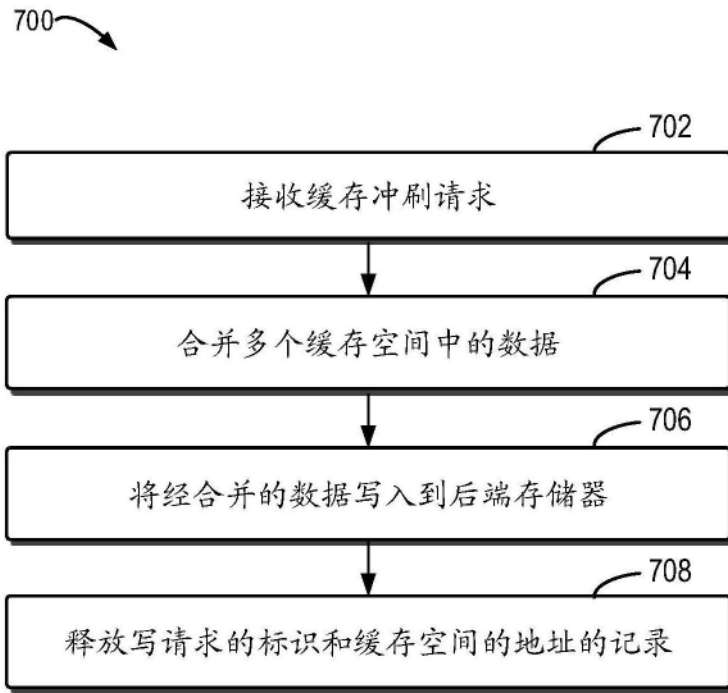


图7

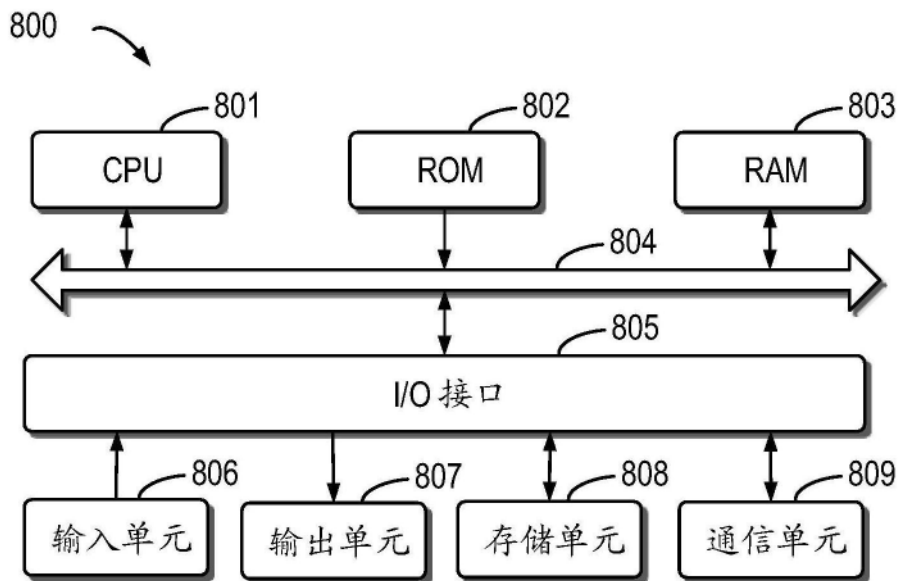


图8