



(12) 发明专利

(10) 授权公告号 CN 101861571 B

(45) 授权公告日 2013. 11. 06

(21) 申请号 200880116111. 1

(22) 申请日 2008. 11. 12

(30) 优先权数据

11/940, 745 2007. 11. 15 US

(85) PCT申请进入国家阶段日

2010. 05. 14

(86) PCT申请的申请数据

PCT/US2008/083251 2008. 11. 12

(87) PCT申请的公布数据

W02009/064793 EN 2009. 05. 22

(73) 专利权人 美光科技公司

地址 美国爱达荷州

(72) 发明人 戴维·R·雷斯尼克

(74) 专利代理机构 北京律盟知识产权代理有限

责任公司 11287

代理人 沈锦华

(51) Int. Cl.

G06F 12/00(2006. 01)

G06F 9/00(2006. 01)

(56) 对比文件

US 6564304 B1, 2003. 05. 13, 说明书第 4 栏
倒数第 2 段到第 7 栏第 4 段, 说明书附图 1、2.

US 6748493 B1, 2004. 06. 08, 说明书第 3 栏
第 2 段, 第 5 栏第 3 段.

CN 1071771 A, 1993. 05. 05, 全文.

US 2003/0217239 A1, 2003. 11. 20, 全文.

US 2006/0215481 A1, 2006. 09. 28, 全文.

US 6564304 B1, 2003. 05. 13, 说明书第 4 栏
倒数第 2 段到第 7 栏第 4 段, 说明书附图 1、2.

审查员 杨继爽

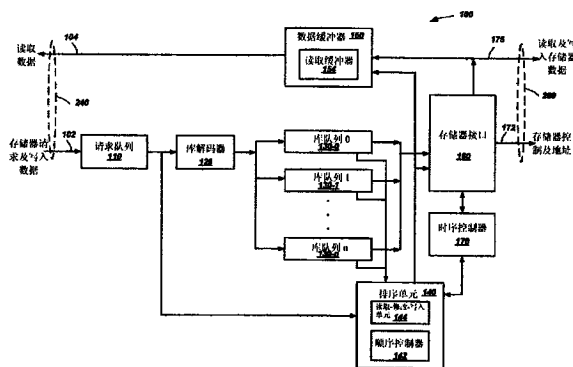
权利要求书4页 说明书10页 附图5页

(54) 发明名称

用于修改存储器存取次序的系统、设备及方法

(57) 摘要

本发明揭示用于控制存储器存取操作的系统及方法。所述系统可包含执行针对存储器装置的请求的一个或一个以上请求器。在存储器控制器内, 请求队列从请求器接收请求, 库解码器确定目的地库, 且将所述请求置于适当库队列中。排序单元确定是否可相对于所接收次序重新排序当前请求, 并基于所述重新排序确定产生新存储器循环次序。所述重新排序可基于是否存在针对相同存储器页的多个请求、多个读取或多个写入。存储器接口按所述存储器循环次序执行每一存储器请求。数据缓冲器保持写入数据直到其被写入到所述存储器且保持读取数据直到其被返回到所述请求器。所述数据缓冲器还可保持在读取-修改-写入操作中使用的存储器字。



1. 一种存储器控制器,其包括:

库解码器,其用于响应于多个存储器请求中的每一存储器请求的存储器地址而按所接收次序针对所述存储器请求中的每一者确定存储器的目的地库;

多个库队列,每一库队列包括读取队列和写入队列且所述每一库队列用于存储对与所述库队列相关联的存储器库的未决存储器请求并追踪所述所接收次序;

排序单元,其用于响应于在相同库队列中且针对相同存储器页的至少两个存储器请求的存在而确定不同于所述所接收次序的存储器循环次序;及

存储器接口,其用于按所述存储器循环次序执行每一存储器请求。

2. 根据权利要求1所述的存储器控制器,其进一步包括读取-修改-写入 RMW 单元,所述读取-修改-写入 RMW 单元用于:

检测针对所请求存储器字的一部分的 RMW 请求;

如果数据缓冲器中不存在所述所请求存储器字,那么从所述存储器读取所述所请求存储器字并将其存储于所述数据缓冲器中;

用来自所述 RMW 请求的数据修改所述数据缓冲器中的所述所请求存储器字的所述部分。

3. 根据权利要求2所述的存储器控制器,其中所述 RMW 单元进一步经配置以将所述所请求存储器字保留于所述数据缓冲器中以用于将来的 RMW 请求。

4. 根据权利要求1所述的存储器控制器,其中所述排序单元进一步经配置以将相同类型的至少两个额外存储器请求重新排序为依序发生,其中所述相同类型选自自由读取请求及写入请求组成的群组。

5. 根据权利要求1所述的存储器控制器,其进一步包括数据缓冲器,所述数据缓冲器用于保持来自所述存储器的所请求数据直到所述所请求数据被返回到所述至少一个请求器。

6. 根据权利要求5所述的存储器控制器,其进一步包括:

至少一个请求器,其用于执行所述存储器请求;且

所述存储器包括多个存储器装置,所述多个存储器装置中的每一存储器装置包括至少一个存储器库。

7. 一种用于控制存储器存取操作的方法,其包括:

按所接收顺序接收指定用于多个存储器库的多个存储器请求;

在多个库队列中存储所述多个存储器请求,每一库队列包括至少读取队列和写入队列;

针对所述多个存储器请求中的每一者确定所述多个存储器库中的目的地库;

通过将所述多个存储器请求中的一个请求与所述多个存储器请求中的至少另一请求进行比较来按所述所接收顺序在时间上接近而非顺序的未执行存储器请求中检测相同类型的至少两个未执行存储器请求,其中所述相同类型选自自由读取请求及写入请求组成的群组;

基于检测结果重新排序所述请求顺序;

依序执行所述相同类型的所述至少两个存储器请求。

8. 根据权利要求7所述的方法,其进一步包括:

检测针对存储器字的一部分的读取 - 修改 - 写入操作, 其中所述存储器字包括对应于所述多个存储器库的数据宽度的数据宽度;

确定存储器控制器中是否存在所述存储器字;

如果所述存储器控制器中不存在所述存储器字, 那么将所述存储器字从存储器读取到所述存储器控制器中;

将所述存储器字的所述部分传递到起始所述读取 - 修改 - 写入操作的请求器; 及

用来自所述读取 - 修改 - 写入操作的所述存储器字的所述部分修改所述存储器控制器中的所述存储器字。

9. 根据权利要求 8 所述的方法, 其中所述数据宽度选自自由存储器总线宽度及存储器突发循环数据宽度组成的群组。

10. 根据权利要求 7 所述的方法, 其进一步包括:

按所述所接收顺序在时间上接近的未执行请求中检测针对相同存储器页的至少两个额外存储器请求; 及

基于检测针对所述相同存储器页的至少两个额外存储器请求的结果重新排序所述请求顺序。

11. 根据权利要求 10 所述的方法, 其包括:

按经重新排序的顺序执行所述至少两个额外存储器请求;

如果按所述所接收顺序执行针对所述多个存储器库内的每一存储器页的存储器存取, 那么在所述存储器存取结束时关闭所述存储器页; 及

在经重新排序了顺序的请求之间维持所述相同存储器页打开。

12. 根据权利要求 7 所述的方法, 其进一步包括:

监视未决存储器请求的年龄;

在所述未决存储器请求等待被执行时, 增加其优先级; 及

防止将新存储器请求重新排序为在具有高优先级的所述未决存储器请求之前发生。

13. 根据权利要求 7 所述的方法, 其进一步包括:

监视所述多个所述存储器库中的每一存储器库的繁忙状态; 及

执行针对未指示繁忙状态的存储器库的下一存储器请求。

14. 根据权利要求 7 所述的方法, 其进一步包括给所述读取请求指派比所述写入请求高的优先级并在所述读取请求与任何未决写入请求之间不存在地址冲突时在所述写入请求之前执行所述读取请求。

15. 一种存储器控制器, 其包括:

请求队列, 其用于从至少一个请求器按所接收次序接收存储器请求;

库解码器, 其用于响应于所述存储器请求中的每一者的存储器地址而针对所述请求队列中的每一存储器请求确定存储器的目的地库;

多个库队列, 其用于存储对与所述多个库队列中的所述库队列相关联的存储器库的未决存储器请求并追踪所述所接收次序;

数据缓冲器, 其用于保持具有对应于所述存储器的数据宽度的数据宽度的至少一个存储器字;

读取 - 修改 - 写入 RMW 单元, 其用于检测针对所请求存储器字的一部分的 RMW 请求并

确定所述数据缓冲器中是否存在所述所请求存储器字；及

存储器接口,其用于响应于存储于所述多个库队列中的一者中的所述 RMW 请求而执行针对所述存储器的读取请求以在所述数据缓冲器中不存在所述所请求存储器字的情况下读取所述所请求存储器字并将所述所请求存储器字存储于所述数据缓冲器中；

其中所述 RMW 单元控制将所述所请求存储器字的所述部分在修改之前返回到所述至少一个请求器并用来自所述 RMW 请求的数据修改所述数据缓冲器中的所述所请求存储器字的所述部分。

16. 根据权利要求 15 所述的存储器控制器,其中所述 RMW 单元进一步经配置以将所述所请求存储器字保留于所述数据缓冲器中以用于将来的 RMW 请求并在修改所述数据缓冲器中的所述存储器字之后将所述存储器字写入到所述存储器。

17. 根据权利要求 15 所述的存储器控制器,其进一步包括排序单元,所述排序单元经配置以基于在相同库队列中且针对相同存储器页的至少两个存储器请求的存在而确定在时间上接近的未执行请求中的存储器循环次序,且

其中所述存储器接口进一步经配置以：

按所述存储器循环次序执行每一存储器请求；

如果按所接收顺序执行对每一存储器页的存储器存取,那么在所述存储器存取结束时关闭所述存储器页；且

在经重新排序了顺序的请求之间维持所述相同存储器页打开。

18. 根据权利要求 15 所述的存储器控制器,其进一步包括排序单元,所述排序单元经配置以以将相同类型的至少两个额外存储器请求重新排序为依序发生,其中所述相同类型选自由读取请求及写入请求组成的群组,且其中所述存储器接口进一步经配置以按由额外存储器循环修改的所述存储器循环次序执行每一存储器请求。

19. 一种用于控制存储器存取操作的方法,其包括：

按所接收顺序接收多个存储器请求；

针对所述多个存储器请求中的每一者确定多个存储器库中的目的地库；

通过将所述多个存储器请求中的一个请求与所述多个存储器请求中的至少另一请求进行比较来按所述所接收顺序在时间上接近的未执行请求中检测所述多个存储器请求中针对相同存储器页的至少两个存储器请求；

根据检测结果重新排序所述所接收顺序；

按经重新排序的顺序执行所述至少两个存储器请求；

如果按所述所接收顺序执行对每一存储器页的存储器存取,那么在所述存储器存取结束时关闭所述存储器页；及

在经重新排序了顺序的请求之间维持所述相同存储器页打开。

20. 根据权利要求 19 所述的方法,其进一步包括：

按所述所接收顺序在时间上接近而非顺序的未执行存储器请求中检测相同类型的至少两个额外未执行存储器请求,其中所述相同类型选自由读取请求及写入请求组成的群组；及

基于检测所述相同类型的至少两个额外未执行存储器请求的结果重新排序所述请求顺序。

21. 根据权利要求 19 所述的方法,其进一步包括:

检测针对存储器字的一部分的读取 - 修改 - 写入操作,其中所述存储器字包括对应于所述多个存储器库的数据宽度的数据宽度;

确定存储器控制器中是否存在所述存储器字;

如果所述存储器控制器中不存在所述存储器字,那么将所述存储器字从存储器读取到所述存储器控制器中;

将所述存储器字的所述部分传递到起始所述读取 - 修改 - 写入操作的请求器;及

用来自所述读取 - 修改 - 写入操作的所述存储器字的所述部分修改所述存储器控制器中的所述存储器字。

22. 根据权利要求 21 所述的方法,其中所述数据宽度选自由存储器总线宽度及存储器突发循环数据宽度组成的群组。

用于修改存储器存取次序的系统、设备及方法

[0001] 相关申请案交叉参考

[0002] 本申请案主张 2007 年 11 月 15 日提出申请的名称为“System, Apparatus, and Method for Modifying the Order of Memory Accesses (用于修改存储器存取次序的系统、设备及方法)”、序列号为 11/940,745 的美国专利申请案的申请日期的权益。

技术领域

[0003] 实施例涉及计算机系统,且更特定来说涉及存储器控制器增加存储器存取效率的操作方法。

背景技术

[0004] 在大多数计算机及数据处理系统中,主有源存储器(其通常为随机存取存储器(RAM))为动态随机存取存储器(DRAM)。DRAM 的结构通常由组织成若干库的若干存储器单元组成。每一库对应于一存储器单元阵列,其中每一单元均相应地与存储器地址处的数据数字(例如,位)相关联。特定来说,一库内的存储器地址各自由行地址及列地址指定,其中每一行地址寻址一存储器页。因此,每一存储器页含有对应于所述页内的不同列指定的数个存储器位置。

[0005] 当执行一连串存取请求时,页请求可发生于当前正打开另一页的库,通常将此称为“页冲突”,因此必须首先关闭(例如,预充电)先前所打开的页。在关闭所述先前页之后,可接着打开(例如,启动)所请求页,且接着可执行对所述所请求页的读取或写入操作。如果在无打开页的库中发现当前所请求页,那么可发生“页未命中”,因此需要执行启动程序。在当前存储器存取请求是针对已根据先前存储器存取请求而打开的页,则可称“页命中”发生。

[0006] 由于必须针对页冲突及相对于页命中请求的页未命中存储器存取执行的额外处理,因此执行前两个过程所需的时间明显多于后者所需的时间。在微处理器技术发展的早期阶段,以先进先出方式接收及完成对存取用于读取及写入操作的 DRAM 存储器页的请求。此处理可能无效,从而导致大量页未命中及冲突,且因此需要处理器及/或存储器控制器资源广泛致力于对存储器页进行预充电及启动。

[0007] 最近,已研发出基于优先级来进行存储器存取的更先进处理方法。存取请求的优先级可基于各种因素,例如发送所述请求的装置类型、所请求存取的类型、期望由所述请求存取的存储器地址等。然而,严格依照优先级来提供存储器存取的问题是,低优先级请求可在过长的时间周期内被拒绝存取。

[0008] 此外,系统中微处理器的数目、微处理器中核心的数目以及每核心的处理线程的数目近期已显著增加,且预期在接下来的几年内会继续增加。可构想出具有成百上千个线程的系统。这些系统通常经设计以使得多个处理器芯片存取一共同存储器。这些请求存取一共同存储器的多个资源对所述存储器施加了额外的压力。

[0009] 核心与线程数目增加的效应中的一者是将需要显著增加的存储器带宽,而主要负

面效应是,所述存储器系统所经历的地址请求串流将由于实际独立或看似独立的程序执行顺序的数目增加而更为随机。第一级及第二级高速缓冲存储器的大小增加(此为过去大多数系统实施方案解决总存储器带宽及等待时间问题的方式)可不够有效且由于核心数目增加以及对合理裸片大小的限制而具有较小增长可能性。此外,每一核心中正执行的线程数目的增加将可能使平均高速缓存命中率降低,而此又导致存储器业务量增加。

[0010] 在当前的 DRAM 技术中,使存储器库循环(启动所述库、读取或写入所请求数据并对所述库进行预充电)的时间比数据移动时间长得多。此长循环时间意味着,如果两个请求在时间上接近而又针对同一存储器库,那么存储器输入/输出(I/O)引脚变为闲置状态达一时间周期以等待第一库循环完成以便可开始第二库循环。由于 DRAM 通常具有可独立循环的多个库,因而此库时序冲突会浪费可用存储器带宽。

[0011] 对于此类不同存储器请求源,需要可在多个线程及多个处理器的系统环境中产生经改进存储器性能的设备与方法。

发明内容

[0012] 本发明实施例可包含用于控制存储器存取操作的系统、存储器控制器及方法。处理系统可包含执行对多个存储器装置的存储器请求的一个或一个以上请求器。所述存储器装置中的每一者包含至少一个存储器库。存储器控制器包含请求队列、库解码器及多个库队列。所述请求队列从请求器接收存储器请求,且所述库解码器基于伴随所述存储器请求的存储器地址确定目的地库。接着将所述请求置于适当库队列中。

[0013] 排序单元确定是否可相对于所接收次序重新排序当前存储器请求,并基于所述重新请求确定产生新存储器循环次序。所述重新排序可基于任一给定库队列内是否存在针对相同存储器页的多个请求。所述重新排序还可基于是否存在相同类型的多个请求,例如读取请求及写入请求。存储器接口以所述排序单元所定义的存储器循环次序来执行每一存储器请求。所述存储器循环次序可基于相同类型的请求、针对相同页的请求或其组合。所述排序单元控制将读取数据按原始接收次序返回到适当请求器。

[0014] 数据缓冲器保持从存储器读取循环返回的读取数据直到将所述所请求数据返回到请求器。所述数据缓冲器还可保持可在来自请求器的读取-修改-写入请求中使用的存储器字。

[0015] 所述排序单元还可检测并控制对所请求存储器字的一部分的读取-修改-写入操作。如果数据缓冲器中不存在所请求存储器字,那么所述重新排序单元引导存储器接口执行所请求存储器字的读取并将其存储于数据缓冲器中。所述重新排序单元还控制将所请求存储器字的所述部分在修改之前返回到至少一个请求器,并用来自 RMW 请求的数据来修改数据缓冲器中的所请求存储器字的所述部分。

[0016] 虽然已描述特定实施例,但这些所描述实施例并非限定性。而是,本发明的范围由所附权利要求书及其法律等效内容涵盖。

附图说明

[0017] 在图解说明本发明实施例的图式中:

[0018] 图 1 是处理系统的简化系统框图;

- [0019] 图 2 是存储器控制器的简化框图；
- [0020] 图 3 是图解说明用于重新排序存储器请求的过程的简化流程图；
- [0021] 图 4 是图解说明用于执行经重新排序存储器请求的过程的简化流程图；且
- [0022] 图 5 是图解说明用于执行读取 - 修改 - 写入操作的存储器存取请求的过程的简化流程图。

具体实施方式

[0023] 本文所揭示实施例包含用于在多个线程及多个处理器的系统环境中产生经改进存储器性能的设备及方法。

[0024] 在以下详细描述中,将参照形成本文一部分的附图,且附图中以图解说明方式显示其中可实践本发明的具体实施例。充分详细地描述这些实施例以使所属领域的技术人员能够实践本发明,但应了解,也可利用其它实施例并可在本发明的范围内做出结构、逻辑及电改变。

[0025] 在此描述中,为不在不必要细节上模糊本发明,可以框图形式显示电路及功能。此外,所显示及描述的具体电路实施方案仅为实例,且除非本文中另有规定,否则不应被视为实施本发明的唯一方式。块定义以及对各块之间的逻辑的划分表示具体实施方案。所属领域的技术人员将容易明了,可以众多其它划分解决方案来实践本发明。一般来说,当关于时序考虑因素等细节对获得对本发明的完整理解并非必需且处于所属领域的技术人员的能力范围内时,可省略这些细节。

[0026] 为易于描述,下文所论述的实施例与动态随机存取存储器 (DRAM) 装置联合使用。然而,应理解,本发明不限于涉及 DRAM 的应用。而是,应强调,本发明实施例可与其它类型的随机存取存储器联合使用,例如静态 RAM (SRAM) 以及 DRAM 的许多不同子种类,以非限定性实例方式包含:快速页模式 DRAM (FPM DRAM)、扩充数据输出 DRAM (EDO DRAM)、突发式 EDO DRAM、同步 DRAM (SDRAM)、双倍数据速率 DRAM (DDR2 DRAM 及 DDR3 DRAM)、内存总线式 (Rambus) DRAM (RDRAM) 等。

[0027] 出于清晰呈现及描述的目的,一些图式可将若干信号图解说明为单个信号。所属领域的技术人员应理解,所述信号可表示信号总线,其中所述总线可具有各种位宽度,且可对任一数目个数据信号(包含单个数据信号)实施本发明。

[0028] 图 1 是根据本发明实施例包含存储器控制器 100 的处理系统 200 的简化系统框图。处理系统 200 可包含耦合到请求总线 240 的一个或一个以上处理器 210、一个或一个以上图形处理器 220 及一个或一个以上专用请求器 235。以非限定性实例方式,这些专用请求器可包含数字信号处理器、浮点处理器、媒体处理器、网络管理器等等。处理器 210、图形处理器 220 以及专用请求器 235 在本文中可统称为请求器 230。此外,在一些实施例中,请求器 230 中的每一者可包含单独请求总线 240 (未显示)。请求总线 240 可包含例如控制信号、地址信号、数据信号、每一请求器 230 的唯一识别符以及请求器 230 内的多个处理器或多个处理线程的唯一识别符等元素。

[0029] 桥接器单元 250 耦合到请求总线 240 (或多个请求总线)。桥接器单元 250 包含存储器控制器 100 且还可包含总线桥接器 260。尽管未显示,但所属领域的技术人员将认识到,处理系统 200 可包含多个存储器控制器 100 及多个总线桥接器 260。另外,桥接器单

元 250 可与作为单独装置的存储器控制器 100 及总线桥接器 260 配置在一起,或桥接器单元 250 可将存储器控制器 100 及总线桥接器 260 集成到单个装置中。另外,桥接器单元 250 或存储器控制器 100 及总线桥接器 260 的单独组件可与请求器单元 230 集成到相同封装或集成电路中。

[0030] 存储器控制器 100 通过一个或一个以上存储器总线 280 耦合到存储器子系统 300。每一存储器总线 280 接纳包含至少一个存储器装置 292 的存储器组件 290 (本文中也称为存储器卡)。存储器组件 290 可形成为存储器卡或存储器模块。可用于处理系统 200 中的存储器模块的非限定性实例包含单联机存储器模块 (SIMM)、双联机存储器模块 (DIMM) 及内存总线式联机存储器模块 (RIMM)。包含存储器组件 290 及存储器装置 292 的各种配置的存储器子系统 300 在本文中可简称为存储器。此外,无需将所述存储器组件封装为卡或模块。作为非限定性实例,可将所述存储器组件以三维封装配置安装于处理器 210 或其它请求器 230 的顶部上。

[0031] 总线桥接器 260 耦合到至少一个外围总线 265。各种装置 270 可耦合到外围总线 265。作为非限定性实例,这些装置可包含存储器控制器、辅助总线桥接器、多媒体处理器、遗留装置接口及杂项输入 / 输出 (I/O) 装置,例如键盘、鼠标及打印机。总线桥接器 260 或存储器控制器 100 也可耦合到一个或一个以上专用高速总线。作为非限定性实例,在个人计算机中,所述专用总线可以是用于将高性能视频卡或其它高带宽外围设备耦合到处理系统 200 的加速图形端口 (AGP) 总线或外围组件互连扩充 (PCI-X) 总线。

[0032] 所属领域的技术人员将认识到,图 1 中所图解说明的处理系统 200 仅为可与存储器控制器 100 的实施例一起使用的处理系统的一个非限定性实例。尽管图 1 图解说明尤其适合于通用计算机 (例如,个人计算机或工作站) 的处理架构,但应认识到,可做出已众所周知的修改以将处理系统 200 配置为更适合在各种应用中使用。举例来说,可使用较简单架构来实施需要处理的许多电子装置,所述架构依赖于处理器 210 及耦合到存储器组件 290、直接耦合到存储器装置 292 或其组合的存储器控制器 100。

[0033] 这些电子装置可包含但不限于:音频 / 视频处理器及记录器、游戏控制台、数字电视机、有线或无线电话、导航装置 (包含基于全球定位系统 (GPS) 及 / 或惯性导航的系统) 及数码相机及 / 或记录器。所述修改可包含 (举例来说) 消除不必要的组件、添加专用装置或电路及 / 或集成多个装置。

[0034] 图 2 是根据本发明实施例的存储器控制器 100 的简化框图。存储器控制器 100 包含:请求队列 110、库解码器 120、库队列 (130-0、130-1 至 130-n)、排序单元 140、数据缓冲器 150、存储器接口 160 及时序控制器 170。可存在与存储器子系统 300 中每一存储器库相关联的库队列 130。然而,库队列 130 也可经配置以处置多个库。

[0035] 存储器请求在存储器请求总线 240 上进入存储器控制器 100 并以先进先出方式在请求队列 110 中被接收。如较早所陈述,请求总线 240 可包含控制信号、地址信号、数据信号、每一请求器 230 (图 1) 的唯一识别符及请求器 230 内多个处理器或多个处理线程的唯一识别符。可将许多或所有这些多个信号存储于库队列 130 中,因此存储器控制器 100 可保存并追踪在存储器控制器 100 中未决的关于每一存储器请求的必要信息。

[0036] 库解码器 120 按所接收请求进入请求队列 110 的顺序来处理所述请求,并基于 (举例来说) 库及行地址信息将所述所接收请求置于库队列 130 中的一者中。排序单元 140

追踪库队列 130 中的信息,并向存储器接口 160 及时序控制器 170 指示应从库队列 130 提取存储器请求的次序以及哪一库队列 130 应供应下一存储器存取请求。根据以下论述将明了,排序单元 140 可包含用于控制库队列内各种存储器请求的顺序(例如,次序)的顺序控制器 142。排序单元 140 还可包含用于控制读取-修改-写入(RMW)循环的 RMW 单元 144。RMW 单元 144 也可与排序单元 140 分离。在此情况下,库解码器 120 可将 RMW 单元视为内部库,同样可将排序单元 140 视为内部库。

[0037] 存储器接口 160 及时序控制器 170 基于存储器总线 280 上的存储器类型借助适当的时序及控制在存储器总线 280 上产生并控制存储器存取请求。作为非限定性实例,在来自排序单元 140 的引导下,存储器接口 160 及时序控制器 170 可产生用于在存储器总线 280 上对 DRAM 的总线循环进行预充电、启动、读取、写入及刷新的适当命令及总线循环。

[0038] 数据缓冲器 150 可配置为单个缓冲器,其能够保持去往存储器子系统 300 的写入数据、从存储器子系统 300 返回的读取数据、读取-修改-写入数据或其组合。库队列 130 可分离成读取队列及写入队列(未显示)以使得能够相对于读取更容易地追踪写入。此分离可有助于检测可重新排序哪些存储器请求,如下文更全面解释。另外,可将写入数据连同相应存储器写入请求置于库队列中,且可将所述数据从数据缓冲器 150 移除。

[0039] 对于写入操作,写入数据伴随存储器请求。在读取或写入命令之后将写入存储器请求从适当库队列 130 移除(即,解除分配)且已在存储器控制总线 172 及存储器数据总线 175 上启动相关联写入数据。

[0040] 对于读取操作,请求总线 240 上的存储器请求 102 不包含数据。因此,无需在将存储器请求置于库队列 130 中时分配总线缓冲器 150 中用于读取请求的空间。而是,可在已于存储器控制总线 172 上启动读取操作且读取数据在存储器数据总线 175 上返回时分配数据缓冲器 150 空间。另一选择为,控制逻辑可追踪未完成读取请求的数目并在读取缓冲器面临溢出危险时防止进一步的读取请求。

[0041] 所述读取存储器请求可根据存储器控制器 100 如何配置而在库队列 130 中保持未决状态达各种不同时间。作为一个实例,库队列 130 可包含将数据返回到其适当请求器 230 可需要的所有请求信息,例如存储器循环类型、始发请求器 230、始发处理线程及类似信息。在此实例中,数据缓冲器 150 可仅包含读取数据及指示哪一物理曾做出相应请求使得数据可在读取数据总线 104 上正确返回的标签。排序单元 140 使用来自库队列 130 的请求信息及来自数据缓冲器 150 的相关联数据以在请求总线 240 上将数据返回到适当请求器 230。在返回所述数据之后,可将存储器请求及相关联读取数据分别从适当库队列 130 及数据缓冲器 150 移除(即,解除分配)。

[0042] 作为另一实例,当在存储器总线 280 上处理读取请求时,可将请求信息传送到数据缓冲器 150 使得所述请求信息处于数据缓冲器 150 中。在此情形下,由于欲将读取数据返回到适当请求器 230 的所有信息均处于数据缓冲器 150 中,因此当在存储器总线 280 上处理存储器请求时可将所述请求从库队列移除。排序单元 140 使用来自数据缓冲器 150 的请求信息及相关联数据以在请求总线 240 上将数据返回到适当请求器 230。在返回所述数据之后,可将存储器请求及相关联读取数据从数据缓冲器 150 移除(即,解除分配)。

[0043] 有效存储器系统设法使其数据引脚在尽可能接近 100%时间内保持充满有用数据。为此目的且除非在下文论述中有所陈述,将存储器操作视为“已关闭库”操作。换句话

说,在存储器参考结束时对所有存储器参考进行预充电。尽管可存在与必须执行更多行地址选择操作相关联的额外开销,但由于多处理器及多线程系统环境中请求串流的随机性,已关闭库操作仍有用。即使线程正做出经排序请求串流(如存储器控制器 100 及存储器系统所经历),但存在太多与其它线程交错的其它请求以至于几乎没有机会在其它请求已使所述线程曾参考的打开页无效之前经历来自原始线程的第二请求。

[0044] 此已关闭库策略不同于通常用于单个处理器系统中的常规存储器控制器。在那些常规存储器控制器中使用打开库策略,这是因为极有可能来自请求器 230 的时间接近的存取将针对相同库及那些库内的相同页。

[0045] 然而,在一些情况下,可有利地辨识对相同 DRAM 页的存取并绕过预充电操作,使得针对所述相同页的背对背操作为打开库且可使其更为有效。为增强辨识针对相同页的存储器请求并使所述库打开以供后续存取的此能力,可有用地相对于请求队列 110 中的接收次序重新排序存储器请求。

[0046] 在单个处理器系统中,由于来自存储器的数据恰好是按照所述单个处理器对其期望的次序,因此存储器控制器可经设计以使请求保持其被做出的次序。然而,维持此次序对于存储器系统来说通常并不有效(即,存储器带宽可损失)。然而,假定有多个核心,且甚至同一核心中存在多个线程时,将会出现可由使请求保持按次序而引起的存储器系统性能问题。

[0047] 一般来说,只要参考相同存储器地址保持按次序,就几乎不存在由不按次序方式返回请求而引起的程序正确性问题。一般来说,将同步及强制排序指令添加到处理器的指令集以视需要启用数据及程序排序,以在尚未存在所述指令的情况下协调核心与线程之间的活动。因此,编译器写入器及类似低级程序最担心排序问题。的确有时将程序同步功能添加到系统的存储器架构中,如下文对存储器系统中原子及字节启用能力的论述中所说明。

[0048] 即使必须将每一核心或每一线程的请求保持按严格程序请求次序,但如果存储器控制器 100 可断定请求是来自此线程或核心抑或彼线程或核心(基于请求总线 240 上的信息),那么其可将每一程序串流的请求保持按次序同时允许无冲突请求(来自不同核心及线程的请求)相对于彼此打乱次序。另外,所述控制器使用此源信息将读取数据路由回到请求源或使用此源信息来指示错误(如果有错误发生)。

[0049] 在此重新排序策略中,如果库队列 130 中的“较老”请求是针对繁忙库,那么可先于所述较老请求来选择对空闲库的请求(只要遵循其它存储器时序及使用规则即可)。此重新排序使得原本可能不会使用的总线时间得以有效地使用。

[0050] 另外,可将来自请求队列 110 的传入存储器请求与适当库队列中的其它存储器请求进行比较以确定新存储器请求是否与其它未决请求中的一者针对相同页。如果是,那么可将所述新存储器请求置于针对所述相同页的所述未决请求之后。在此情况下,可超驰所述关闭库策略以允许对所述相同页的背对背存取。

[0051] 当然,在重新排序的情况下,一些请求可被推迟或不止一次地移动。因此,排序单元 140 可包含时序单元或类似机构以指示所述队列中请求的年龄并修改“老”请求以使其具有较高优先级并及时被执行。

[0052] 在读取循环与写入循环之间切换数据方向时可花费多个存储器时钟来使数据总线转向。如果所述转向时间为两个时钟,其中每一数据传送花费四个时钟(假设双向 DDR 传

送及对 DRAM 的含 8 个数据请求的突发),那么当请求在读取与写入之间交替时可损失 33% 的峰值存储器带宽。因此,如果将存储器读取与其它存储器读取分组在一起,且将存储器写入与其它存储器写入分组在一起,那么可减少因总线转向时间所致的无效。

[0053] 一般来说,可重新排序读取及写入两种类型存储器存取请求。然而,存储器控制器 100 可实施一策略:仅在必要时重新布置写入请求,以达到经重新排序的顺序不与较早接收的请求产生任何寻址冲突,或另外不会因队列 130 中的其它存储器存取请求而干扰已存储或将存储于相关存储器位置中的数据的数据的程度。此外,必须使对相同程序地址的参考保持按程序请求次序。如果请求是在不同程序实体之间,那么此请求次序通常由软件强制执行,而如果请求是通过单个请求源(例如核心或线程)完成,那么认为由存储器控制器 100 来遵循此请求次序。

[0054] 作为另一读取对写入策略,赋予读取分组优于写入分组的优先级可为有用的。换句话说,可给任何读取指派优先级,直到某一原因(例如,队列已满或地址冲突)迫使切换到写入。从请求器 230 的角度来说,此读取优先级策略可减少读取等待时间,其代价是包含检验未将读取请求重新排序在针对相同地址的写入请求之前以确保所述读取请求从处于队列之前的未决写入获得数据的逻辑。然而,在一些实施例,如果针对具有未决写入的地址执行读取,那么存储器控制器 100 可从适当队列直接返回写入数据,因此并未实际上读取存储器。

[0055] 处理器或其它请求器 230 有时有必要写入个别字节并执行读取-修改-写入操作。尽管可能看起来个别小数据量不会被很频繁地写入,但随着核心及线程在共用程序上工作时数目的增加,这些操作也将增加。一种多个线程进行通信的最佳方式是使用作为存储器系统的一部分而存储及管理的“原子”存储器操作,例如旗标及信标。这些旗标及信标通常需要存储器字节级或字级操作良好地工作。

[0056] 读取-修改-写入(RMW)是从存储器地址读取值、修改曾读取的值且接着将所述经修改值写回到所述存储器地址的原子顺序。多线程编程的常规途径是使用 RMW 锁定来使对共享资源的存取同步。例如信标等同步基元可由多线程程序用来确保如果同时执行代码的某些区段将破坏代码的所述区段之间可共享的存储器结构,那么不会这样做。如果一个线程试图获取已由另一线程保持的锁定,那么所述线程将阻塞直到所述锁定解除。

[0057] 无阻塞算法可使用例如比较与交换(CAS)的原子读取-修改-写入操作。所述比较与交换 CPU 指令(或 x86 架构中的 CMPXCHG 指令)是以原子方式将存储器位置的内容与给定值进行比较且在所述二者相同的情况下将所述存储器位置的内容修改为给定新值的特殊指令。CAS 可用于有效地实施多处理器系统中的信标。

[0058] 传统上,已用处理器中的执行逻辑或借助处理器的数据高速缓冲存储器中的逻辑完成对字节写入及信标的支持。然而,在一些系统中,其可用于支持将这些小操作数作为个别数据项写入到最终存储器目的地(即,存储器子系统 300)且不作为整个高速缓冲存储器线的一部分。

[0059] 另外,许多处理器使用 32 位及 64 位数据项且可极为频繁地写入那些类型的项。但是,作为非限定性实例,典型 DRAM 协议及接口可移动 32 字节突发。如果仅需要写入四个字节,那么可需要个别字节-写入-选择信号。然而,在一些存储器系统中,字节启用可能不可用。在这些系统中,可必要地读入(举例来说)含有 4 字节数据项的 32 字节突发。接

着,将所述四个数据字节插入到相当于 32 个字节的数据,且接着将经修改的 32 字节量重写回到存储器。在此实例中,当四个字节本应在接口中存在字节启用能力的情况下被发送时,已既读取又写入 64 个数据字节。持续对不需要的数据的此移动,但其浪费存储器性能。一般来说,在本文中存储器存取的宽度称为“数据宽度”。作为非限定性实例,此数据宽度可以是存储器总线 280 的物理位宽度或存储器突发循环的宽度。另外,除非本文另有规定,否则 RMW 操作将一般指代具体 RMW 操作以及对数据宽度的若干部分的写入。

[0060] 因此,对于 RMW 操作来说,控制器 100 可经配置以管理这些部分写入操作。另外, RMW 单元 144 中的数据缓冲器可经配置以保持这些 RMW 操作的经保留区域,使得存储器控制器 100 中存在将读取的数据,此可消除在存储器总线 280 上执行读取操作的需要。

[0061] 图 3、4 及 5 是图解说明根据本发明实施例的用于管理经重新排序存储器存取请求及 RMW 操作的过程的简化流程图。当描述图 3、4 及 5 时,也将参照图 1 及 2 中的各种元件。

[0062] 图 3 是图解说明用于重新排序存储器存取请求的过程 300 的简化流程图。在操作块 302 处,接收新请求。基于库地址,在操作块 304 处将所述请求移动到适当库队列。决策块 306 确定刚接收的请求是否在时间上接近相同库队列中未决的任何先前请求且是否针对与相同库队列中未决的任何先前请求相同的库及页。本文所使用的在时间上接近定义在请求队列 110 处接收请求的时间间隔。所述请求可以时间戳来标记或附加有队列属性形式的年龄。以此方式,可不重新排序时间上相距太远的存储器请求,或可给“老”存储器请求指派高优先级,使得不将新请求置于“老”存储器请求前面。可将队列中的时间间隔长度或年龄设定为存储器控制器 100 的配置参数。

[0063] 如果所述请求与另一未决请求针对相同库及页,那么操作块 308 为所述库队列重新定义存储器循环次序以将所述新存储器循环直接置于针对相同页的另一未决请求之后。当然,如较早所陈述,不应将新读取循环置于针对相同地址的写入循环之前,且不应将写入循环置于针对相同地址的读取或写入循环之前。另外,库队列中可能存在基于年龄或对读取循环的偏好而具有足够高优先级的未决请求,因此不应将新请求置于高优先级请求之前。

[0064] 操作块 310 指示,将新请求直接置于其后的未决请求应被标记为打开页请求。换句话说,对于所标记的请求来说,已超驰关闭库策略。因此,在已在存储器总线 280 上执行标记为打开页的请求之后,已知后续存储器请求是针对相同页且可背对背地执行所述两个请求而无需预充电及执行其之间的行地址选通。

[0065] 重新排序过程 300 可包含读取对写入的重新排序。如果所述重新排序过程不包含重新排序读取对写入,则所述过程经由路径 311 返回到操作块 302 以等待新存储器请求。如果所述重新排序过程的确包含重新排序读取对写入,那么所述过程经由路径 312 继续到决策块 313。决策块 313 确定所述新请求是否在时间上接近于相同类型的未决请求,其中所述类型可以是读取请求或写入请求。如果所述请求并非相同类型,那么所述过程返回到操作块 302 以等待新存储器请求。

[0066] 如果所述请求与时间上接近的存储器请求为相同类型,那么操作块 314 为所述库队列重新定义存储器循环次序以将新存储器循环直接置于相同类型的未决存储器循环之后。以此方式,可使用重新排序过程 300 将读取与写入循环分组在一起以减少可需要的数据总线转向循环的数目。另外,可赋予读取高于写入的优先级以减少读取等待时间。同样,

不应将一个类型的循环置于针对相同地址的不同类型循环之前。另外，库队列中可能存在基于年龄或对读取循环的偏好而具有足够高优先级的未决请求，因此不应将新请求置于高优先级请求之前。在重新定义请求次序之后，所述过程返回到操作块 302 以等待新存储器请求。还可能针对每一库将读取与写入请求分离成单独队列。

[0067] 图 4 是图解说用于在存储器总线 280 上执行经重新排序存储器请求的存储器存取过程 400 的简化流程图。在操作块 401 处，从适当库队列中检索下一存储器请求。可通过使用基于库队列的循环方案或优先化选择来确定所述适当库队列。另一选择为，库队列选择可使用不从对应于繁忙存储器库的库队列选择请求的策略，如较早所论述。

[0068] 决策块 402 测试所述下一存储器请求是读取还是写入。如果所述过程当前处于写入顺序中，那么操作在决策块 454 处继续，如下文解释。如果所述过程当前处于读取顺序中，那么操作在决策块 404 处继续，其确定所述请求是否为读取以及所请求库是否不繁忙（例如，被启动或预充电）。如果不是，那么操作块 406 更新当前读取请求及循环的库编号以着眼于下一库 / 队列。

[0069] 决策块 408 测试以确认当前读取请求的库是否打开。如果所述库未打开，那么操作块 410 启动所述库。决策块 412 测试以确定是否设定控制逻辑使得所述库在当前读取请求结束时应保持打开。如果将使所述库保持打开，那么操作块 414 在存储器总线 280（图 2）上执行读取循环。如果将关闭所述库，那么操作块 416 在存储器总线 280（图 2）上执行读取循环，同时在相同命令中指示在读取操作之后需要预充电。

[0070] 在读取循环完成的情况下，一些实施例可包含等待决策块 418，其一直等到读取缓冲器 154 中有存储所返回的读取数据的空间。决策块 420 测试以确认所述过程是否完成执行读取循环。存储器控制器 100 可经配置以执行某一数目个背对背读取。此可由若干因素来确定。作为非限定性实例，所述因素可包含在队列中等待的读取的数目、相对于等待的写入数目的等待被执行的读取的数目、等待被执行的写入的优先级、等待被执行的读取的优先级及其组合。

[0071] 如果将继续读取循环，那么操作块 406 在需要时更新库编号以进行下一读取循环。如果不继续读取循环，那么操作块 480 将当前模式从执行读取操作切换到执行写入操作，且操作在操作块 401 处继续。

[0072] 从决策块 402 返回到退出，如果所述过程当前处于写入顺序中，那么操作在决策块 454 处继续，其确定所述请求是否为写入且所请求库是否不繁忙。如果不是，那么操作块 456 更新当前写入请求及循环的库编号以着眼于下一库 / 队列。

[0073] 决策块 458 测试以确认当前写入请求的库是否打开。如果所述库未打开，那么操作块 460 启动所述库。决策块 462 测试以确定是否设定控制逻辑使得所述库在当前写入请求结束时应保持打开。如果所述库将保持打开，那么操作块 464 在存储器总线 280（图 2）上执行写入循环。如果所述库将关闭，那么操作块 466 在存储器总线 280（图 2）上执行写入循环，同时在相同命令中指示在写入操作之后需要预充电。

[0074] 决策块 470 测试以确认所述过程是否完成执行写入循环。存储器控制器 100 可经配置以执行某一数目个背对背写入。此可由若干因素来确定。作为非限定性实例，所述因素可包含在队列中等待的读取的数目、相对于等待的写入数目的等待被执行的读取的数目、等待被执行的写入的优先级、等待被执行的读取的优先级及其组合。

[0075] 如果写入循环将继续,那么操作块 456 在需要时更新库编号以进行下一写入循环。如果写入循环不继续,那么操作块 480 将当前模式从执行写入操作切换到执行读取操作,且操作在操作块 401 处继续。

[0076] 如果存储器循环未被标记为打开页,那么遵循关闭库策略且操作块 410 对刚刚存储的页进行预充电且所述过程返回到操作块 402 以从适当库队列中检索下一存储器循环。

[0077] 图 5 是图解说明用于执行读取-修改-写入操作及部分数据宽度写入的存储器存取请求的过程 500 的简化流程图,所述过程可与图 3 的重新排序过程 300 在某种程度上平行进行。在操作块 502 处,从请求队列 110(图 2)接收新请求。基于库地址,在操作块 504 处将所述请求移动到适当库队列。决策块 506 确定刚刚接收的请求是否为如来自请求总线 240 的具体信号或命令所识别的 RMW 请求。如果所述请求并非 RMW 请求,那么控制返回到操作块 502 以进行下一请求。实际上, RMW 过程 500 与重新排序过程 300 在某种程度上平行进行,因此针对非 RMW 请求,图 3 的重新排序过程 300 可在决策块 306 处继续。

[0078] 如果所述请求是 RMW 请求,那么决策块 508 确定所寻址 RMW 位置的整个数据字是否已位于 RMW 单元 144(图 2)中的数据缓冲器中。如果所述整个数据字不在所述数据缓冲器中,那么操作块 510 在存储器总线 280 上执行读取操作并将数据字存储于 RMW 单元 144 中的数据缓冲器中。

[0079] 对于常规 RMW 操作而非部分数据写入操作来说,操作块 512 在需要时将数据字的 RMW 部分返回到请求器 230。一般来说,所返回的 RMW 部分是已发生写入之前的读取部分。然而,存储器控制器 100 可经配置以在已发生写入之后返回所述 RMW 部分。另外,可将额外信息与所述 RMW 部分一起返回。作为非限定性实例,可返回比较与交换操作的比较结果。

[0080] 操作块 514 将数据字的 RMW 部分写入到 RMW 单元 144 中的 RMW 数据缓冲器中。如果需要,那么操作块 516 经由存储器总线 280 将包含所写入 RMW 部分的整个数据字写回到存储器。

[0081] 决策块 518 测试以确认是否将 RMW 数据字标记为保留于数据缓冲器 150 中。如较早所陈述, RMW 数据缓冲器可经配置以维持多个 RMW 数据字。以此方式,可能避免存储器总线 280 上针对 RMW 请求的存储器循环。如果将所述数据字标记为保留,那么所述过程返回到操作块 502 以等待下一请求。如果未将所述数据字标记为保留,那么操作块 520 将数据字从数据缓冲器 150 清除以允许有用于额外读取或写入数据的空间。

[0082] RMW 单元 144 还可经配置以始终保留 RMW 值直到需要用于到达存储器控制器的新 RMW 请求的空间,此可避免与每一新请求一起写入到存储器。如果已完成此保留,那么重新排序逻辑必须在需要时创建对存储器的写入操作。因此,所述保留可允许在每次更新 RMW 值时推迟写入,其通常针对用于在过程之间(例如,在不同核心、线程及其组合之间)通信及更新的 RMW 值而发生。

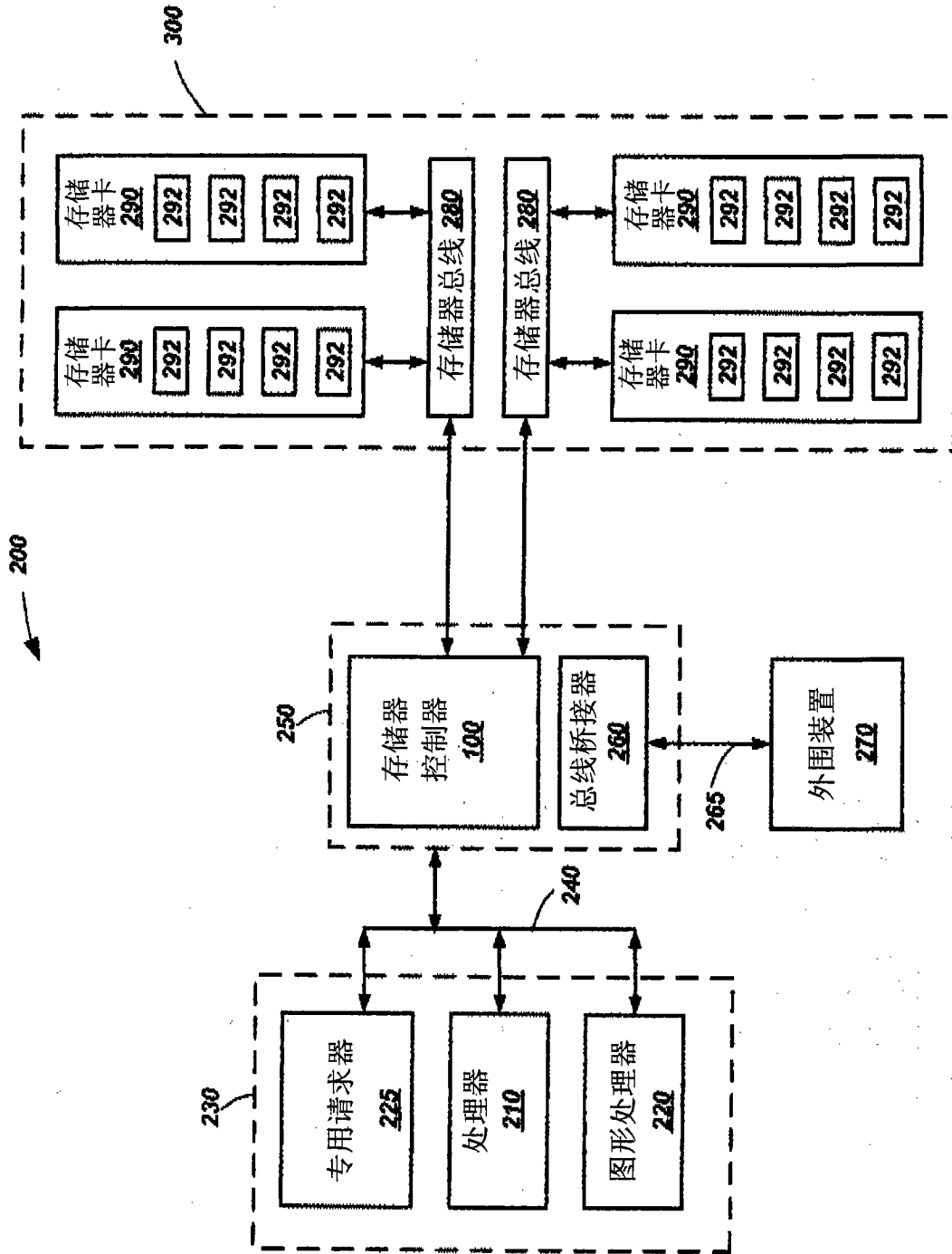


图 1

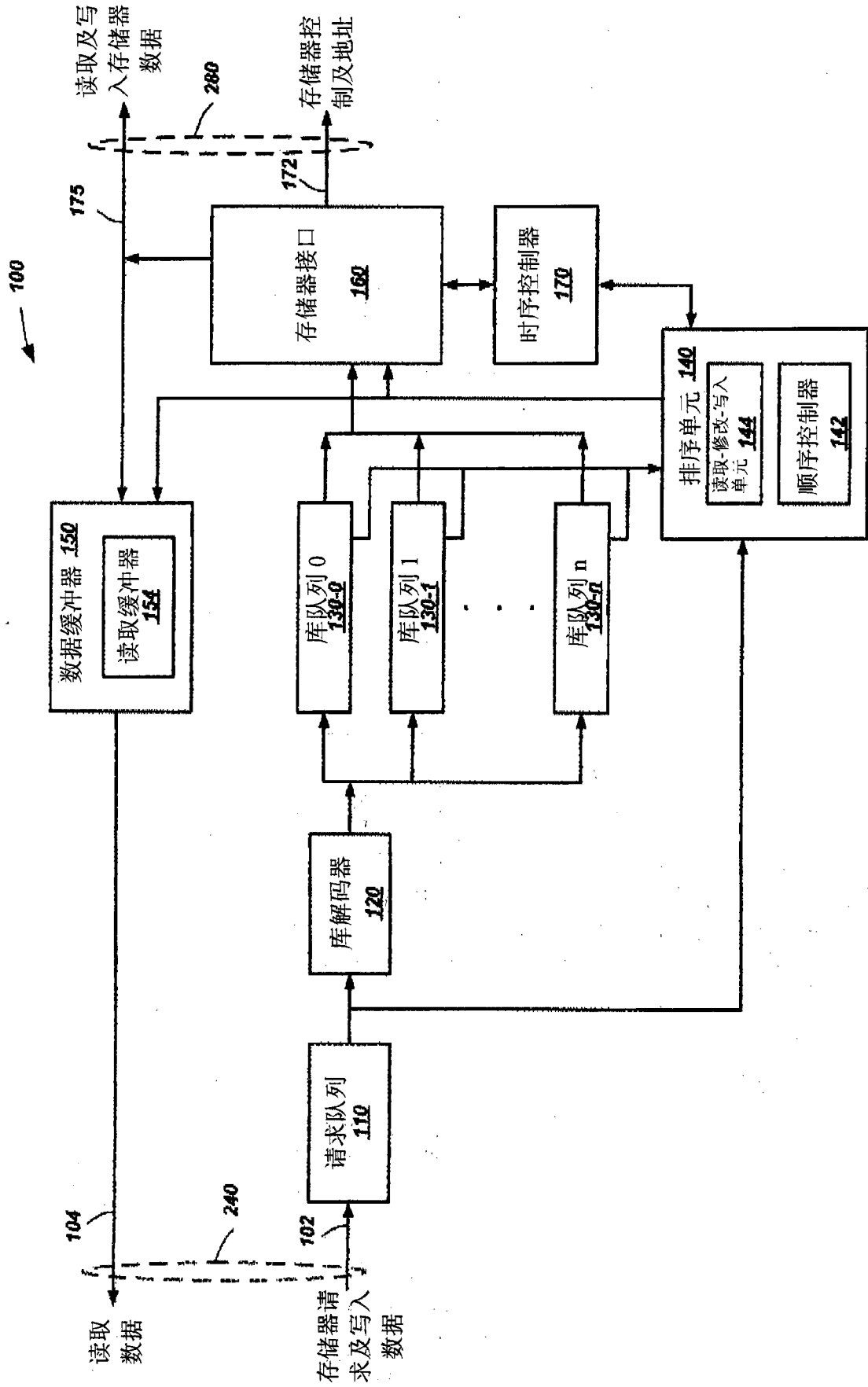


图 2

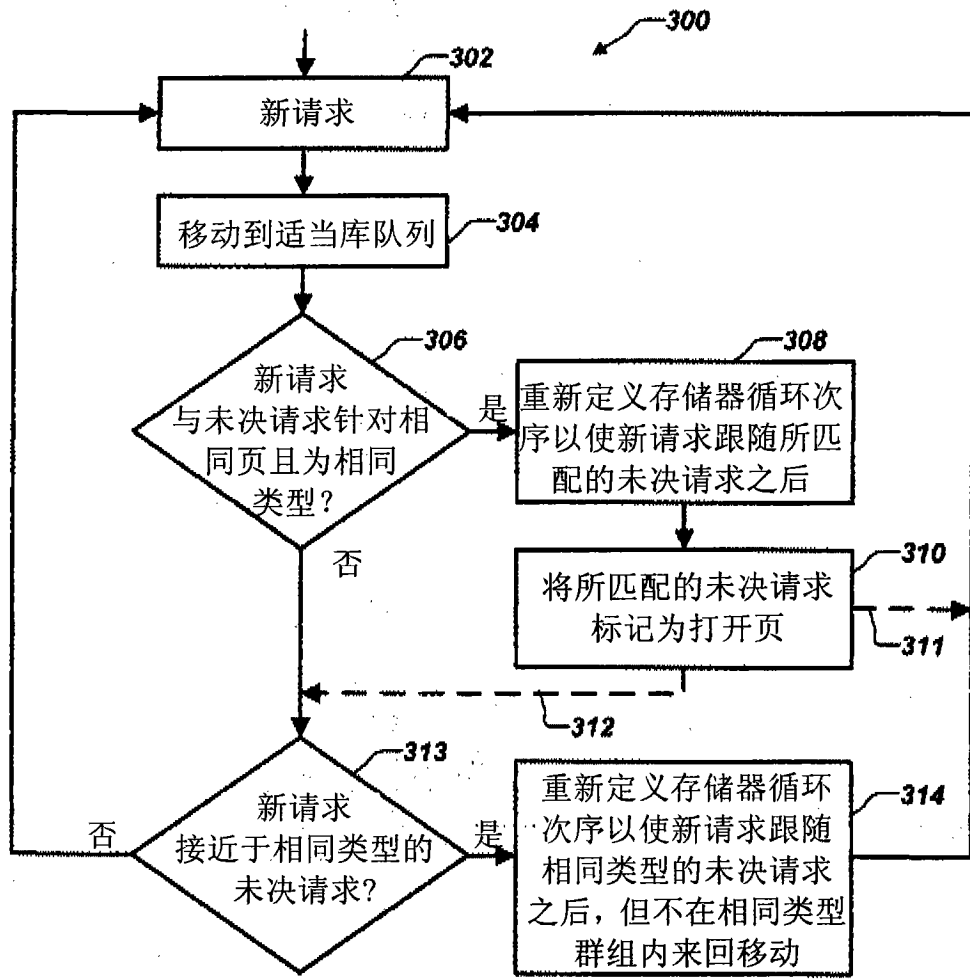


图 3

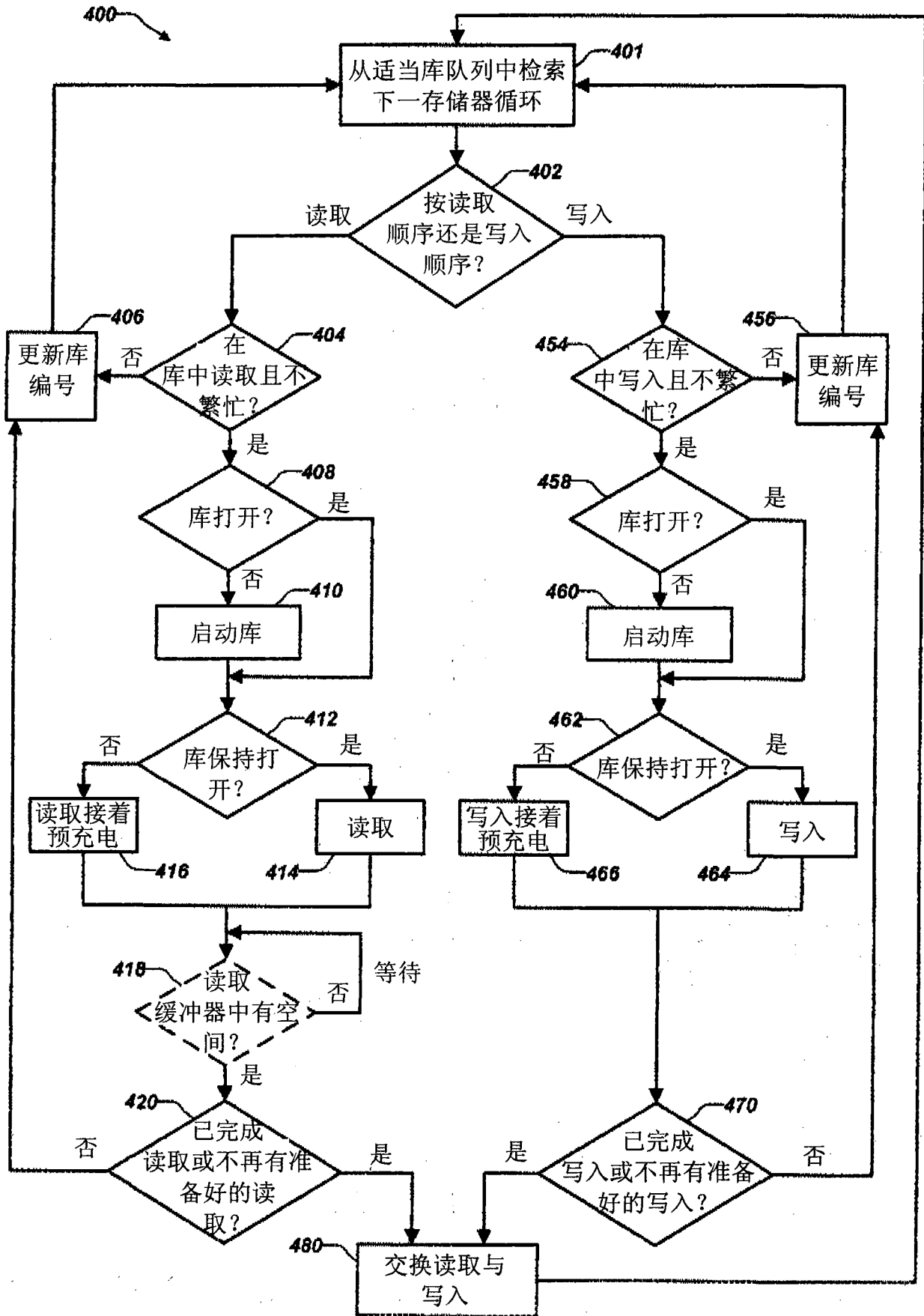


图 4

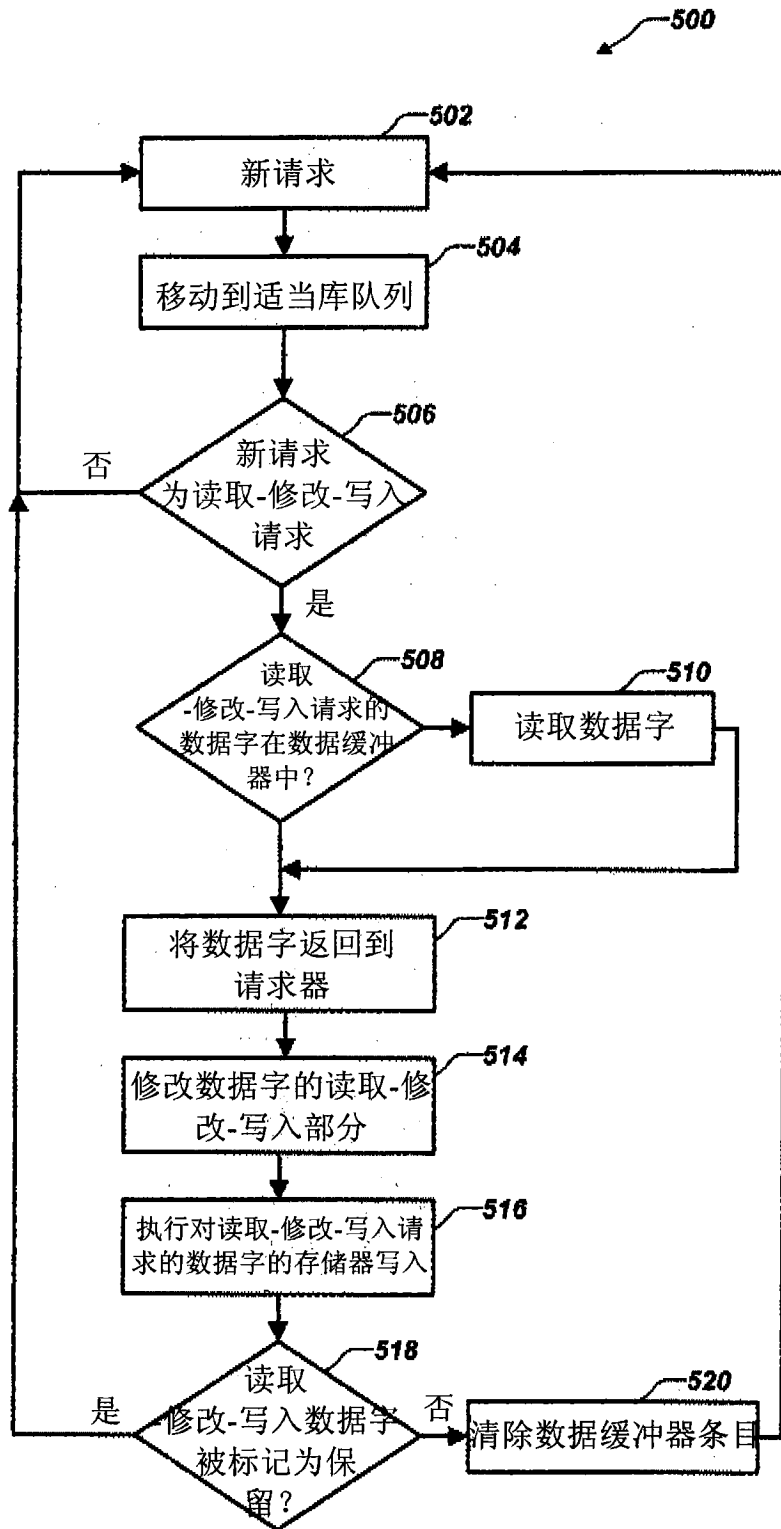


图 5