



(12)发明专利

(10)授权公告号 CN 106233269 B

(45)授权公告日 2019.12.10

(21)申请号 201580019963.9

J·安德鲁斯 J·塞尔 K·波

(22)申请日 2015.04.06

(74)专利代理机构 上海专利商标事务所有限公司 31100

(65)同一申请的已公布的文献号

代理人 顾嘉运

申请公布号 CN 106233269 A

(43)申请公布日 2016.12.14

(51)Int.Cl.

(30)优先权数据

G06F 13/16(2006.01)

14/252,673 2014.04.14 US

(85)PCT国际申请进入国家阶段日

(56)对比文件

2016.10.14

US 2007/0113023 A1, 2007.05.17, 说明书第[0005]-[0131]段, 图1-8.

(86)PCT国际申请的申请数据

US 2011/0154352 A1, 2011.06.23, 说明书第[0033]段, 图1-5.

PCT/US2015/024414 2015.04.06

CN 101046784 A, 2007.10.03, 全文.

(87)PCT国际申请的公布数据

US 2007/0113023 A1, 2007.05.17, 说明书第[0005]-[0131]段, 图1-8.

W02015/160541 EN 2015.10.22

US 2008/0104293 A1, 2008.05.01, 全文.

(73)专利权人 微软技术许可有限责任公司

审查员 张秋芳

地址 美国华盛顿州

(72)发明人 N·T·考齐 S·卡里

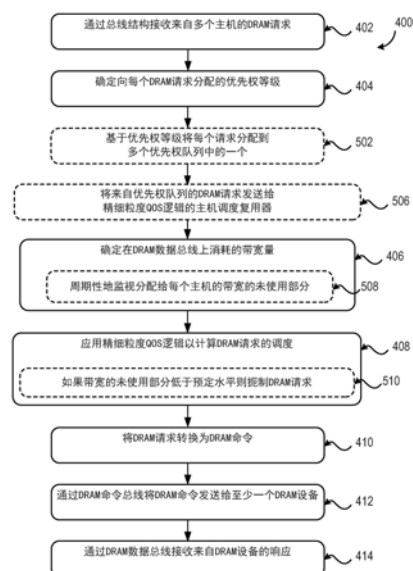
权利要求书2页 说明书12页 附图6页

(54)发明名称

在存储器控制器中的精细粒度带宽供应

(57)摘要

提供了用于应用精细粒度QoS逻辑的系统和方法。所述系统可以包括存储器控制器,所述存储器控制器被配置为通过总线结构接收来自多个主机的存储器访问请求。所述存储器控制器确定所述多个主机的每个主机的优先权等级,并进一步确定由每个主机在存储器数据总线上所消耗的存储器数据总线带宽量。基于分配给每个主机的优先权等级以及每个主机所消耗的存储器数据总线带宽量,存储器控制器应用精细粒度QoS逻辑以计算对存储器请求的调度。基于该调度,存储器控制器将存储器请求转换为存储器命令,将这些存储器命令通过存储器命令总线发送给存储器设备,并通过存储器数据总线从存储器设备接收响应。



1. 一种存储器控制器,被配置为:

通过总线结构接收来自多个主机的存储器请求;

确定分配给所述多个主机的每个主机的优先权等级;

确定由每个主机在存储器数据总线上消耗的存储器数据总线带宽量;

应用精细粒度服务质量QoS逻辑以基于分配给每个主机的优先权等级和由每个主机所消耗的存储器数据总线带宽量计算对所述存储器请求的调度,由每个主机在每次请求所消耗的存储器数据总线带宽是基于关于最后发出的请求的信息来计算的;以及

基于所述调度;

将所述存储器请求转换成存储器命令;

通过存储器命令总线将所述存储器命令发送给至少一个存储器设备;以及

通过所述存储器数据总线接收来自所述存储器设备的响应,其中所述存储器设备是动态随机存取存储器DRAM设备,所述存储器数据总线是DRAM数据总线,所述存储器请求是DRAM存储器请求,所述存储器命令是DRAM存储器命令,而所述存储器命令总线是DRAM命令总线,所述存储器控制器还被配置为:

基于所述优先权等级将每个DRAM请求分配到多个优先权队列中的一个;

将来自所述多个优先权队列的DRAM请求发送到主机调度复用器;

周期性地监视分配给每个主机的带宽的未使用部分;以及

计算所述调度以便如果所述带宽的所述未使用部分落至预定水平之下就在所述存储器控制器处扼制所述DRAM请求;

其中周期性地监视分配给每个主机的带宽的未使用部分包括应用精细粒度QoS带宽供应逻辑以将所述DRAM数据总线中的预定的带宽量分配给所述多个主机的每个主机,以及

其中周期性地监视分配给每个主机的带宽的未使用部分包括应用QoS信用累加器以递增地累加所述多个主机的每个主机的信用,累加的速率对应于由所述精细粒度QoS带宽供应逻辑分配给所述主机的所述带宽,并且每个主机的总信用不超过预定的最大QoS信用值。

2. 一种用于在存储器控制器中执行精细粒度带宽供应的方法,包括:

通过总线结构接收来自多个主机的存储器请求;

确定分配给每个所述存储器请求的优先权等级;

确定由每个主机在存储器数据总线上消耗的存储器数据总线带宽量;

应用精细粒度服务质量QoS逻辑以基于分配给每个主机的优先权等级和由每个主机所消耗的存储器数据总线带宽量计算对所述存储器请求的调度,由每个主机在每次请求所消耗的存储器数据总线带宽是基于关于最后发出的请求的信息来计算的;以及

基于所述调度;

将所述存储器请求转换成存储器命令;

通过存储器命令总线将所述存储器命令发送给至少一个存储器设备;以及

通过存储器数据总线接收来自所述存储器设备的响应,所述存储器设备是动态随机存取存储器DRAM设备,所述存储器数据总线是DRAM数据总线,所述存储器请求是DRAM存储器请求,所述存储器命令是DRAM存储器命令,而所述存储器命令总线是DRAM命令总线,所述方法还包括:

基于所述优先权等级将每个DRAM请求分配到多个优先权队列中的一个;

将来自所述多个优先权队列的DRAM请求发送到主机调度复用器；

周期性地监视分配给每个主机的带宽的未使用部分；以及

计算所述调度以便如果所述带宽的所述未使用部分落至预定水平之下就在所述存储器控制器处扼制所述DRAM请求；

其中周期性地监视分配给每个主机的带宽的未使用部分包括应用精细粒度QoS带宽供应逻辑以将所述DRAM数据总线中的预定的带宽量分配给所述多个主机的每个主机，以及

其中周期性地监视分配给每个主机的带宽的未使用部分包括应用QoS信用累加器以递增地累加所述多个主机的每个主机的信用，累加的速率对应于由所述精细粒度QoS带宽供应逻辑分配给所述主机的所述带宽，并且每个主机的总信用不超过预定的最大QoS信用值。

3. 如权利要求2所述的方法，其特征在于，周期性地监视分配给每个主机的带宽的未使用部分包括：

应用精细粒度QoS计费结算逻辑来计算由每个DRAM存储器请求所消耗的在所述DRAM数据总线上的带宽；以及

通过从对应于每个主机的所述QoS信用累加器中的信用中减去所计算的带宽来计算每个主机的信用余额。

4. 如权利要求3所述的方法，其特征在于，计算由每个DRAM存储器请求所消耗的在所述DRAM数据总线上的带宽包括基于关于最后发出的请求的信息计算请求的带宽成本。

5. 如权利要求4所述的方法，其特征在于，应用精细粒度调度逻辑并且如果所述带宽的未使用部分落至预定水平之下则在所述存储器控制器处扼制DRAM请求是通过应用精细粒度QoS调度逻辑以基于每个请求的所述优先权等级调度所述DRAM请求并计算所述调度以便如果发出所述DRAM请求的主机的所述QoS信用累加器中的所述信用余额低于预定最小QoS信用值则扼制所述DRAM请求来实现。

6. 如权利要求5所述的方法，其特征在于，发送所述DRAM请求给多个DRAM设备中的至少一个包括：

将所述DRAM请求从所述主机调度复用器发送到DRAM协议引擎；以及

将所述DRAM请求从所述DRAM协议引擎发送到所述多个DRAM设备中的至少一个。

在存储器控制器中的精细粒度带宽供应

[0001] 背景

[0002] 动态随机存取存储器 (DRAM) 设备可以在各种各样的电子设备中找到, 包括计算机、智能电话、平板、游戏控制台和电视。DRAM 设备允许在这些电子设备上运行的应用在来自所述设备的非易失存储 (例如闪存或硬盘) 的指令被所述设备的中央处理单元 (CPU) 或其它固定硬件单元执行之前临时存储所述指令和数据。

[0003] 出于效率的原因, DRAM 设备由电子系统中的许多不同的主机设备或可以生成对 DRAM 存储器设备的读或写通信的知识产权 (IP) 逻辑块来共享。每个主机设备通常经由总线结构和存储器控制器向 DRAM 设备发出存储器请求。所述存储器控制器负责调度这些存储器请求并确定如何将所述请求发送给 DRAM 设备。DRAM 设备的访问等待时间以及进而有效利用带宽取决于存储器控制器能够如何高效地且有效地调度所述请求。当所述调度是低效的时, 过度的带宽变化可以引起系统的硬件和软件应用中的故障, 例如在视频游戏中的帧率下降或由于丢失音频帧导致的声音爆音。常见的防止这些故障的方案是利用确定如何调度请求的服务质量 (QoS) 逻辑并执行广泛测试以确保应用在尽可能多的已知带宽变化场景下将不会发生故障。

[0004] 在现有技术中, QoS 逻辑解决方案主要依靠两种机制: 优先级和速率限制。优先级机制相对于来自己被分配了较低优先权的主机的请求偏爱来自己已经被系统分配了较高优先权的主机的请求。然而, 为了防止低优先权主机的饥饿, 速率限制机制被用于防止高优先权主机超出由系统分配给它们的带宽量。

[0005] 然而, 这些现有的 QoS 逻辑解决方案还允许在 DRAM 设备的有效带宽中的宽变化, 如主机设备所察觉到的。这样, 使用先前的 QoS 逻辑解决方案针对电子系统开发的软件应用被局限于特定类型的 DRAM, 所述特定类型的 DRAM 的带宽变化已经被广泛测试。如果一种不同类型的 DRAM 设备 (较快的或较慢的) 被用在所述系统中, QoS 逻辑解决方案将经历附加的带宽变化并且软件应用将不能正确运行。在通常具有长产品生命周期的游戏控制台的示例中, 这种缺陷阻止了控制台制造者在游戏控制台的生命周期的后期阶段出于担心先前开发的游戏将无法在经更新的硬件上正确运行而切换到较快、较廉价类型的 DRAM。

[0006] 概述

[0007] 提供了用于应用精细粒度 QoS 逻辑的系统和方法。所述系统可以包括存储器控制器, 所述存储器控制器被配置为通过总线结构接收来自多个主机的存储器访问请求。所述存储器控制器确定所述多个主机的每个主机的优先权等级, 并进一步确定由每个主机在数据总线上所消耗的数据总线带宽量。基于分配给每个主机的优先权等级以及每个主机所消耗的数据总线带宽量, 存储器控制器应用精细粒度 QoS 逻辑以计算对存储器访问请求的调度。基于该调度, 存储器控制器将存储器访问请求转换为存储器访问命令, 将这些存储器访问命令通过存储器命令总线发送给存储器访问设备, 并通过存储器数据总线从存储器设备接收响应。

[0008] 提供本概述以便以简化的形式介绍以下在详细描述中进一步描述的一些概念。本发明内容并不旨在标识所要求保护主题的关键特征或必要特征, 也不旨在用于限制所要求

保护主题的范围。而且,所要求保护的主体不限于解决该公开的任一部分中所注的任何或全部缺点的实现方式。

[0009] 附图简述

[0010] 图1是根据一个实施例被配置为执行精细粒度QoS逻辑以供应DRAM数据总线的带宽的系统的示意图。

[0011] 图2是图1的系统的精细粒度QoS逻辑的示意图,其可被配置为计算由DRAM请求在DRAM数据总线上所消耗的带宽量。

[0012] 图3是被配置为执行精细粒度QoS逻辑以供应DRAM数据总线的带宽的系统的第二实施例的示意图,在其中主机、总线结构和DRAM存储器控制器被形成为片上系统。

[0013] 图4是根据一个实施例的用于在DRAM存储器控制器中执行精细粒度带宽供应的方法的流程图。

[0014] 图5是示出可用于完成图4的方法的步骤的子步骤的流程图。

[0015] 图6是示出可用于完成图4的方法的一个步骤的子步骤的流程图。

[0016] 图7是示出系统的各附加组件的示意图。

[0017] 详细描述

[0018] 在此公开了用于控制对诸如动态随机存取存储器 (DRAM) 之类的存储器的访问的系统和方法以解决上述挑战。

[0019] 在现代电子系统中,通常被称为主机设备的CPU和其它硬件设备通过存储器控制器和各种总线向系统的DRAM发出请求是很常见的。存储器控制器可以使用各种队列和复用器来调度从主机设备发出的请求并可应用服务质量 (QoS) 逻辑以便确定执行所述请求的最高效的方式。

[0020] 图1描绘了系统10的实施例,所述系统包括存储器控制器12,该存储器控制器被配置为通过总线结构18从多个主机16接收动态随机存取存储器 (DRAM) 请求14,确定分配给所述多个主机16的每个主机的优先权等级,确定由每个主机在DRAM数据总线30上所消耗的DRAM数据总线带宽量48,将精细粒度QoS逻辑26应用到基于分配给每个主机16的优先权等级以及由每个主机16所消耗的带宽量48计算对DRAM请求14的调度,并基于所述调度将DRAM请求14转换为DRAM命令42,通过DRAM命令总线34将DRAM命令42发送给至少一个DRAM设备32,并通过DRAM数据总线30接收来自DRAM设备32的响应。主机16可以例如是视频游戏控制台的CPU和GPU,并可以正在请求在DRAM设备32上读取或写入与控制台的用户当前正在玩的视频游戏有关的数据。

[0021] 当由主机16发出请求14时,系统10被配置为通过总线结构18传送请求。总线结构在图1中被示作安排在主机16和存储器控制器12之间的单个结构。然而,总线结构18还可以包括适应来自主机的16的所有通信量模式所需的数量的总线分段。图3示出了系统10的第二实施例,在其中总线结构被分成多个总线分段,包括具有速率限制器的分段60和不具有速率限制器的分段62。转回到图1,总线结构18可以被配置为应用QoS优先权分配逻辑20以基于分配给每个主机16的优先权等级将一个优先权等级分配给每个请求14。优先权等级可以由在系统10上运行的软件应用分配给每个主机16。当应用是在系统10上运行时这样的分配可以(例如在每次启动系统10时一次)在应用边界上发生,或在每次启动系统10上的虚拟化环境时发生。主机16可以包括可以对带宽变化敏感且如果带宽变化变得过度可以引起降

级的用户体验的实时主机设备。这样的实时主机设备可以由在系统10上运行的软件应用分配较高的优先权等级。

[0022] 在将优先权等级分配给每个请求14之后,总线结构8可以被配置为将所述请求发送给存储器控制器12。存储器控制器12可以被进一步配置为基于优先权等级将每个DRAM请求14分配到多个优先权队列22中的一个队列,并调度所述多个优先权队列22中的每个队列中的DRAM请求14。

[0023] 为了理解可由优先权队列22用于调度请求14的调度参数,简要转至DRAM设备32的结构和操作的讨论是有帮助的。DRAM设备32可以被配置为“页面”阵列,每个页面包含多个存储器单元(例如在每页8千比特和16千比特之间)。所述页面可以以行和列安排,并且进一步被分组到库(bank)中,每个库包含数万个页面。取决于在DRAM设备32中的页面是否已经被所述请求14之一访问,它可以是打开或关闭的。系统10可以被配置以便DRAM设备32的所有库中的所有页面在系统10启动时是关闭的。当DRAM请求14之一被发出为请求访问在DRAM设备32的库中的一个关闭的页面时,所述请求14将最初导致“页面丢失”并且存储器控制器12将必须执行“页面打开操作”以打开目标页面。在页面打开操作正在发生的同时,存储器控制器12将不能访问该库中的任何其它页面的内容,因为存储器控制器12通常被配置为在任何一个时间在每个库中仅允许有一个打开页面。因此,在页面打开操作期间,存储器控制器12可以被配置为寻求以其它库中的页面为目标的请求14。

[0024] 当页面打开时,访问相同页面的后续请求14将最初导致在存储器控制器12中的“页面命中”并且这样可以在无需执行页面打开操作的情况下被完成。这样,存储器控制器12可以被配置为执行“相同页面收集”,其中请求访问相同页面的请求14被分组在一起以减少对页面打开操作的需求。

[0025] 当后续请求以与最后发出的请求(即在当前请求之前完成的最后的请求)相同库中的一不同的页面为目标,所谓的“页面冲突”发生了,并且存储器控制器12在通过页面打开操作打开被请求14作为目标的页面之前必须通过“页面关闭操作”关闭当前打开的页面。页面打开操作和页面关闭操作两者都消耗DRAM数据总线30上的带宽。在DRAM数据总线30上的带宽消耗可以被作为完成给定请求14所需的总线周期数来测量。总线周期数以及进而每次操作在DRAM数据总线30上消耗的带宽量48是由用于DRAM设备32的特定类型的存储器的定时参数确定的。然而,不管所使用的存储器的类型如何,在DRAM数据总线30上的带宽消耗48依据所述请求14是页面命中、页面丢失还是页面冲突而变化。

[0026] 另外,DRAM请求14可以是读取请求或写入请求。如果请求14是读取请求,存储器控制器12被配置为执行“读取操作”、在DRAM命令总线34上发送“读取命令”并在预定数目的时钟周期之后接收在DRAM数据总线30上的“读取数据”。如果请求14是写入请求,存储器控制器12被配置为执行“写入操作”、在DRAM命令总线34上发送写入命令并在等待预定数目的时钟周期之后在数据总线30上发送写入数据。然而,当请求14是写入请求并且最后发出的请求是读取请求时,存储器控制器12被配置为执行“读取到写入总线回转”以避免数据总线30的使用中的冲突所导致的数据事故(即对数据的处理不当的风险)。例如,如果当写入操作开始时读取数据还是在DRAM数据总线30上返回,在传出写入数据和传入读取数据之间的总线冲突可能发生。为了避免在写入操作期间的这样的发生,在开始写入操作之前,存储器控制器12可以被配置为确认没有先前的读取数据正在DRAM数据总线30上被返回。当存在冲突

时,存储器控制器12可以被配置为执行读取到写入总线回转,其中延迟所述写入命令以允许在将写入数据发送到DRAM设备32之前读取数据被在数据总线30上返回。类似地,当请求14是读取请求且最后发出的请求是写入请求时,存储器控制器12被配置为执行“写入到读取总线回转”,延迟读取命令以允许在通过DRAM数据总线30从DRAM设备32将读取数据返回到存储器控制器12之前将写入数据写入到DRAM设备32的存储器中。读取到写入总线回转和写入到读取总线回转操作这两者都消耗多个总线周期,以及进而在DRAM数据总线30上的带宽量。这样,完成请求14所需的在DRAM数据总线30上所消耗的带宽量48可以取决于存储器控制器12正发出给DRAM设备32的当前请求14的请求类型(即读取或写入)以及最后发出的请求14的请求类型。为了最小化该带宽消耗,存储器控制器12可以被配置为执行“读取/写入分组”,在其中基于请求类型分组请求14以减少对读取到写入总线回转和写入到读取总线回转操作的需求。

[0027] 而且,如果请求14是读取请求,发出请求14的主机16可能请求还未在DRAM设备32上被写入的数据。为了避免在数据已经被写入之前发出请求14,必须将请求14的页面地址与所有的优先权队列22中的所有写入请求14的页面地址进行比较。这个过程被称为地址冲突解决,并且可以被用于确保数据一致性。如果存在匹配,正被讨论的读取请求14必须一直等待直到所有早先的写入请求14都被完成。

[0028] 下面示出的表1描述了基于所请求的页面位置、请求类型以及在做出请求14时刻的DRAM设备32的状态(即先前完成的请求类型)的给定请求14的某些常见情况的带宽成本的示例。

[0029]

先前 DRAM 请求	当前 DRAM 请求					
	相同页面		不同页面			
	读取	写入	相同库		不同库	
			读取	写入	读取	写入
读取	BL/2	BL/2 + tR2W	tPRE + tACT + BL/2	tR2W + tPRE + tACT + BL/2	<ul style="list-style-type: none"> • 如果与目标页面不同的页面是打开的: tPRE + tACT + BL/2 • 如果目标页面是关闭的: tACT + BL/2 • 如果目标页面是打开的: BL/2 	<ul style="list-style-type: none"> • 如果与目标页面不同的页面是打开的: tR2W + tPRE + tACT + BL/2 • 如果目标页面是关闭的: tACT + BL/2 • 如果目标页面是打开的: BL/2
写入	BL/2 + tW2R	BL/2	tW2R + tPRE + tACT + BL/2	tPRE + tACT + BL/2	<ul style="list-style-type: none"> • 如果与目标页面不同的页面是打开的: tW2R + tPRE + tACT + BL/2. • 如果目标页面是关闭的: tACT + BL/2 • 如果目标页面是打开的: BL/2 	<ul style="list-style-type: none"> • 如果与目标页面不同的页面是打开的: tPRE + tACT + BL/2 • 如果目标页面是关闭的: tACT + BL/2 • 如果目标页面是打开的: BL/2

[0030] Table 1: 依据总线周期的每个请求的带宽成本

[0031] 在上述表1中,变量BL指示DRAM设备32的脉冲长度;BL/2是DRAM数据总线30将忙于承载从DRAM设备32读取或写入到DRAM设备32的用于任何双数据率存储器设备(诸如DDRn、低功耗DDRn(LPDDRn)或图形DDRn(GDDRn))的数据的持续时间;变量tACT是打开页面的带宽

成本,它是其所花费的如上所述激活页面的时间; t_{PRE} 是关闭页面的带宽成本,它是其所花费的如上所述关闭页面的时间; t_{W2R} 是以将从DRAM数据总线30在写入操作之后通过执行如上所述的写入到读取总线回转而转用于读取操作所需的DRAM数据总线30的周期数目表示的成本;而 t_{R2W} 是以将从DRAM数据总线30在读取之后通过执行如上所述的读取到写入总线操作而转用于写入操作所需的DRAM数据总线30的周期数目表示的成本。重要的是,注意,尽管不同的DRAM存储器设备(例如DDR3、LPDDR3或GDDR3)可以为这些参数使用不同的名字,在本公开中所述的概念应用到如上所述的任何DRAM设备。注意,在下述段落中将被详细描述刷新操作不包括在表1中,因为它们被认为是DRAM设备32的开销成本的,而非主机16的开销成本。

[0032] 表1表示了系统10的一种可能的实施例。系统10不限于表1中所列出的参数,并且取决于QoS方案的最终目标,可以根据在此所述的原理构建其它表以实现期望结果。

[0033] 存储器控制器12可以被配置为根据上述方法调度优先权队列22中的DRAM请求14以便最小化在DRAM数据总线30上的带宽消耗。每个优先权队列22可以由多个队列46构成。用于每个优先权等级的优先权队列22的数目和大小以及在每个优先权队列22内的队列46的数目可以取决于实现而改变。在图1中,描绘了N个主机设备16和M个优先权队列22。对于给定优先权等级的每个优先权队列22内的队列46可以包括至少一个读取队列和一个写入队列。存储器控制器12可以被配置为调度所述多个优先权队列22的每个队列内的请求14。所述调度可以是主机依赖的且基于诸如地址冲突解决、相同页面收集、读取/写入分组以及分配给每个请求14的优先权等级之类的参数。

[0034] 存储器控制器12还可以被配置为将DRAM请求14发送给主机调度复用器24,周期性地监视分配给每个主机16的带宽的未使用部分,并且如果带宽的未使用部分跌至预定水平之下则扼制在存储器控制器12处的DRAM请求14。存储器控制器12可以被配置为在从优先权队列22中选择了最佳候选请求14之后,基于上述的调度参数将DRAM请求14发送至主机调度复用器24。

[0035] 主机调度复用器24可以被配置为应用精细粒度QoS逻辑26,包括被配置为将DRAM数据总线30中的预定量的带宽分配给所述多个主机16的每个主机的精细粒度QoS带宽供应逻辑28。精细粒度QoS逻辑26可以被配置为将DRAM设备32的带宽划分成两个类别:有用的QoS带宽和开销带宽。开销带宽可以由存储器控制器12消耗的用于确保DRAM设备32的正确操作的带宽,并且可以包括刷新操作所消耗的带宽。在刷新操作期间,存储器控制器12被配置为将对DRAM设备32的所有库的所有的读取操作和写入操作挂起,关闭所有打开的页面并发出刷新命令以刷新在DRAM设备32中的所有页面的内容。在刷新操作完成之后,存储器控制器12被配置为恢复读取和写入操作。有用的QoS带宽可以包括由主机16所消耗的以执行诸如在DRAM设备32上的读取和写入操作之类的操作的带宽48。存储器控制器12可以被配置为通过精细粒度QoS带宽供应逻辑28将预定量的有用QoS带宽分配给主机16。分配给每个主机的带宽量可以由在系统10上运行的软件应用来确定。当应用是在系统10上运行时这样的分配可以(例如在每次启动系统10时)在应用边界上发生或在每次启动系统10上的虚拟化环境时发生。

[0036] 存储器控制器12可以被配置为通过计算每个主机16在DRAM数据总线30上为每个请求14所消耗的带宽48并从由精细粒度QoS带宽供应逻辑28分配给每个主机的有用QoS带

宽量中减去所消耗的带宽48来确定每个主机16的带宽的未使用部分。存储器控制器12可以被配置为依据完成请求14所需的DRAM数据总线30的总线周期数测量由请求14所消耗的带宽48。完成请求14所需的总线周期数可以依据请求14的各种特性而变化,诸如请求类型(即读取或写入)、在DRAM设备32中作为请求14的目标的页面当前是打开还是关闭的以及作为请求14的目标的页面是否位于与作为最后发出的请求的目标的页面相同的库中。

[0037] 存储器控制器12的精细粒度QoS逻辑26还可以被配置为包括QoS信用累加器36。QoS信用累加器36可以被配置为递增地累加所述多个主机16的每个主机的信用。累加率可以对应于由精细粒度QoS带宽供应逻辑28分配给主机的带宽,并且每个主机16的总信用不可以超过预定的最大QoS信用值。例如,如果给定主机16被分配了每秒1千兆字节的DRAM数据总线30中的带宽,并且DRAM数据总线30以200MHz(每秒200,000,000周期)的时钟频率运行,则QoS信用累加器36将在每个周期为正被讨论的主机16指定的信用累加5个字节。如果主机16发出了请求14,则将从对应于主机16的QoS信用累加器36中的信用中减去由请求14所消耗的带宽48。如果主机16没有发出请求14,QoS信用累加器36将继续以每周期5个字节的速率累加信用,直到主机16的总信用达到了最大QoS信用值。系统10可以被配置为允许在系统10上运行的软件应用确定最大QoS信用值。

[0038] 精细粒度QoS逻辑26可以被配置为进一步包括精细粒度QoS计费结算逻辑38,它可以被配置为计算由每个DRAM存储器请求在DRAM数据总线上消耗的带宽48,并通过从对应于每个主机的QoS信用累加器中的信用减去所计算的带宽来计算每个主机的信用余额。图2示出可以被包括在精细粒度QoS计费结算逻辑38的操作中的过程的示意图。如所示,精细粒度QoS计费结算逻辑38可以被配置为从优先权队列22中接收关于当前排队的请求的当前请求信息50。当前请求信息50可以包括请求14的请求类型(即读取或写入)以及当前正被从优先权队列22发送到主机调度复用器24的请求14所针对的页面的页面地址和库。另外,精细粒度QoS计费结算逻辑38可以进一步被配置为从DRAM协议引擎44接收最后发出的请求信息52。最后发出的请求信息54可以包括请求14的请求类型(即读取或写入)以及最后被发送到DRAM设备32的请求14所针对的页面的页面地址和库。最后发出的请求信息54可以进一步包括最后被发送到DRAM设备32的请求14是否具有页面命中、页面丢失或页面冲突。精细粒度QoS计费结算逻辑38可以进一步被配置为使用当前请求信息50和最后发出的请求信息54中的一者或两者来计算由请求14在DRAM数据总线上消耗的带宽48并从对应于发出请求14的主机16的QoS信用累加器36中的信用中减去该带宽量。精细粒度QoS计费结算逻辑38还可以被配置为使用从DRAM协议引擎44接收的全局页面状态信息58。全局页面状态信息58可以包括在DRAM设备32中的所有页面的当前状态。精细粒度QoS计费结算逻辑38可以使用全局页面状态信息58来确定如果被发到DRAM设备32则哪些请求14将具有页面命中、页面丢失或页面冲突。精细粒度QoS计费结算逻辑38还被进一步配置成在从QoS信用累加器36中的总信用中减去由请求14所消耗的带宽48之前考虑刷新所消耗的全局开销带宽56以及刷新的每个时钟周期的信用增益54。因为系统10可以被配置为仅将有用的QoS带宽分配给主机16,QoS计费结算逻辑38可以被配置为提供每时钟周期的信用增益以补偿由刷新操作消耗的带宽或其它形式的开销带宽消耗。在QoS计费结算逻辑38已经减去由最后发出的请求14所消耗的带宽48之后,精细粒度QoS信用累加器36可以进一步被配置为将每个主机16的QoS信用累加器36中的当前余额发送给QoS调度逻辑40,如下所讨论的。

[0039] 转回到图1,精细粒度的QoS逻辑可以被进一步配置为包括精细粒度QoS调度逻辑40,被配置为基于每个请求的优先权等级调度DRAM请求14,并且如果在发出DRAM请求14的主机16的QoS信用累加器36中的信用余额低于预定的最小QoS信用值则扼制DRAM请求14。精细粒度QoS调度逻辑40可以被配置为从精细粒度QoS计费结算逻辑38接收对应于每个排队的请求14的每个主机16的当前信用余额。精细粒度QoS调度逻辑40还可以被进一步配置为选择具有最高优先权等级和在预定最小QoS信用值之上的信用余额的请求14。如果请求14具有低于最小QoS信用值的信用余额,精细粒度QoS调度逻辑40可以被配置为无论请求14的优先权等级如何都扼制该请求14。如果多个请求14具有相同的优先权等级和高于最小QoS信用值的信用余额,可以以循环的方式或根据另一个合适的默认调度算法来执行调度。如果所有可用的请求14都具有低于最小QoS信用值的信用余额,可以选择具有最高优先权等级的请求14。系统10可以被配置为允许在系统10上运行的软件应用确定最小QoS信用值。

[0040] 存储器控制器12可以进一步被配置为包括DRAM协议引擎44,它被配置为接收来自主机调度复用器24的DRAM请求14、通过DRAM命令总线34将DRAM请求14发送给DRAM设备32。DRAM协议引擎44可以被配置为接收已经由主机调度复用器24基于由精细粒度QoS逻辑26所创建的调度所选择的DRAM请求14。存储器控制器12可以被进一步配置为在将DRAM请求14发送给DRAM协议引擎44之前将DRAM请求14转换为DRAM命令42。DRAM协议引擎44可以被进一步配置为接收来自DRAM设备32的响应并将关于最后发出的请求14的信息发送给精细粒度QoS计费结算逻辑38。可选地,协议引擎44可以被配置为接收来自DRAM设备32的响应,所述响应可以是告知存储器控制器12DRAM请求14已被成功完成的中断的形式。协议引擎44可以进一步被配置为将最后发出的请求信息54,如图2所示讨论的,发送给精细粒度QoS计费结算逻辑38。该信息可以包括请求14的请求类型(即读取或写入)、请求14所针对的页面的页面地址和库,以及该请求14是否具有页面命中、页面丢失或页面冲突。

[0041] 图3示出被配置为执行精细粒度QoS逻辑以供应DRAM数据总线的带宽的系统100的第二实施例。系统100包括片上系统(SoC)101,在其上形成了主机16、总线结构18和DRAM存储器控制器12,其进而与DRAM设备32通信。DRAM设备32可以在SoC 101上形成或作为独立设备形成。在片上系统上形成的各种组件起如上所述相类似的作用,并且出于简明的目的将不再重复描述。如上所述,在该实施例中,总线结构被分成多个总线分段,包括具有速率限制器的分段60和不具有速率限制器的分段62。将理解,片上系统101利用速率限制器62来确保在总线结构上的最小固定的带宽量被保留给每个主机,同时通过限制每个主机所消耗的最大带宽确保每个主机不超过该固定量。这避免了调度冲突并促进了从速率限制的主机对DRAM设备的有效访问。如果仅仅通过总线结构的速率不受限的分段(例如总线分段M)的新的主机被加入到这样的系统,它可以消耗总线结构上太多的带宽和/或使被设计为期望具有固定的保留带宽量的其它主机的性能降级。这样,将理解,尽管速率限制器强制实施某种调度控制,由于在总线分段M上没有配置速率限制器,尤其是通过速率不受限总线分段M的DRAM请求可以得益于优先权等级和调度复用器24所应用的精细粒度QoS调度,这进而将防止由通过总线分段M相连的主机对保留带宽的非有意的消耗。这可以改善所有主机的性能。

[0042] 现在转向图4,描绘了用于在存储器控制器中执行精细粒度带宽供应的方法400的流程图。下文描述的方法可在以上参考图1和2描述的系统10的硬件上或由其它合适的硬件来实现。将理解,在其上可以实现在此所述的方法的合适的硬件包括膝上和桌面个人计算

机、智能电话、视频游戏控制台、平板计算设备、智能电视等。在该方法的一些实现中，在图4中的虚线中的步骤可以被忽略，而在其它实现中，虚线中的步骤被包括。

[0043] 所述方法400可以包括在402通过总线结构接收来自多个主机的动态随机存取存储器 (DRAM) 请求。在404，所述方法400可以包括确定为每个DRAM请求分配的优先权等级。

[0044] 在502，所述方法400可以包括基于优先权等级将每个DRAM请求分配到多个优先权队列中的一个。

[0045] 在506，方法400可以包括将DRAM请求发送给主机调度复用器。

[0046] 在406，方法400可以包括确定由每个主机在DRAM数据总线上所消耗的DRAM数据总线带宽量。在508，示出了这可以至少部分通过周期性监视分配给每个主机的带宽的未使用部分来完成。

[0047] 在408，方法400可以包括应用精细粒度服务质量 (QoS) 逻辑以基于分配给每个主机的优先权等级和每个主机所消耗的DRAM数据总线带宽量计算对DRAM请求的调度。在510，示出了可以计算调度以便如果带宽的未使用部分落至预定水平之下就在存储器控制器处扼制DRAM请求。

[0048] 在410，基于在步骤408处创建的调度，方法400可以包括将DRAM请求转换成DRAM命令。在412，方法400可包括通过DRAM命令总线将DRAM命令发送给至少一个DRAM设备。在414，方法400可包括通过DRAM数据总线接收来自DRAM设备的响应。

[0049] 图5示出图解可以完成图4的方法400的步骤508、510和412的示例子步骤的流程图。方法400的步骤508可以通过执行子步骤602-608来完成。在602，方法包括应用精细粒度QoS带宽供应逻辑以将DRAM数据总线中的预定的带宽量分配给所述多个主机的每个主机。在604，方法可以包括应用QoS信用累加器以递增地累加所述多个主机的每个主机的信用，累加的速率对应于由精细粒度QoS带宽供应逻辑分配给主机的带宽，并且每个主机的总信用不超过预定的最大QoS信用值。在606，方法可以包括应用精细粒度QoS计费结算逻辑来计算由每个DRAM存储器请求所消耗的在DRAM数据总线上的带宽。在608，所述方法可以包括通过从对应于每个主机的QoS信用累加器中的信用中减去所计算的带宽48来计算每个主机的信用余额。

[0050] 方法400的步骤510可以通过子步骤610来完成。如在子步骤610处所示，应用精细粒度调度逻辑并且如果带宽的未使用部分落至预定水平之下则在存储器控制器处扼制DRAM请求是通过应用精细粒度QoS调度逻辑以基于每个请求的优先权等级调度DRAM请求并计算调度以便如果发出DRAM请求的主机的QoS信用累加器中的信用余额低于预定最小QoS信用值则扼制所述DRAM请求来实现。

[0051] 方法400的步骤412可以通过子步骤612-614来完成。在612，方法可以包括将DRAM请求从主机调度复用器发送给DRAM协议引擎。在614，方法可以包括将DRAM请求从DRAM协议引擎发送到所述多个DRAM设备中的至少一个。

[0052] 图6示出图解可以完成图4的方法400的步骤406的示例子步骤650、652的流程图。在650，方法可以包括从DRAM协议引擎接收关于最后发出的请求的信息。在652，方法601可以包括基于关于最后发出的请求的信息计算由每个DRAM存储器请求所消耗的在DRAM数据总线上的带宽。

[0053] 暂时返回到图1，系统10还可以被用于对一个或多个软件程序的性能建模，所述软

件程序最初被设计为在具有比系统10更少的总存储器带宽的不同的系统上运行。为此,来自主机16的请求14的一部分可以被标记为源自这些程序的执行。存储器控制器12可以计算如果程序在它们最初设计针对的系统上运行时主机16将接收到的带宽和QoS。使用该计算,存储器控制器12随后可以将带宽和QoS分配给系统10上的请求14。这样,计算设备可以被进一步配置为将来自所述多个主机的存储器请求的一部分标记为源自最初被设计成在具有比该计算设备更少的总存储器带宽的系统上运行的一个或多个软件程序的执行,并且计算如果所述一个或多个软件程序在具有更少的总存储器带宽的系统上运行时所述多个主机将接收到的带宽和服务质量(QoS)。而且,计算设备可以被配置为将带宽和QoS分配给来自所述多个主机的存储器请求;并且,因此,对在具有更少的总存储器带宽的系统上的一个或多个软件程序的性能建模。

[0054] 上述的系统和方法具有各种有用的工业应用。例如,第一应用是允许较快存储器类型产生与较慢存储器类型相同的带宽变化。例如,所述系统可以是使用例如以2133Mbps时钟频率运行的双数据率3(DDR3)存储器设计的视频游戏控制台。在视频游戏控制台的生命周期期间,将存储器改变成例如以2400Mbps时钟频率运行的双数据率4(DDR4)存储器是有利的(例如出于性能和成本原因)。如果这样的改变在视频游戏控制台的生命周期期间做出,存在被设计为与DDR3一起使用的软件将不能与DDR4一起正确运行的风险。然而,如上所述的精细粒度QoS逻辑可以允许在DRAM数据总线上的带宽量被供应给每个主机设备,使得利用DDR4的供应与利用DDR3的供应是一样的。这样,精细粒度QoS逻辑可以限制存储器设备的将来一代(例如DDR4),这样,其带宽变化与存储器设备的前代(例如DDR3)基本上类似,并且这样做允许针对存储器设备的前代设计的软件在无需经历附加带宽变化的情况下在使用存储器设备的更晚一代的系统上起作用。

[0055] 用于如上所述的系统和方法的用例场景的另一个示例是当附加主机设备被添加到系统时减少主机设备所经历的带宽变化量。如果所述系统是例如视频游戏控制台,即使在控制台已经被启动且软件已经针对它被设计之后也可添加附加主机。附加主机设备可能不希望地将附加带宽扰动引入到系统中。然而,精细粒度QoS逻辑可以被用于基于如上所讨论的优先权等级和精细粒度QoS参数以逻辑方式隔离主机设备并调度从其生成的请求,由此,保护系统免受任何添加新的主机设备原本会引起的增加的带宽扰动。为了实现这点,主机设备可以被分配固定量的DRAM数据总线上的带宽,并且,只要存在足够的带宽来支持所有的主机设备,分配给每个主机的带宽将保持未改变的,即使在新主机被加入时。这导致确保该主机设备以与附加主机被加入之前相同方式运转所需的测试的量的显著减少。在这样的应用中,精细粒度QoS逻辑被用于为系统中的每个主机设备保留固定量的带宽,并且使用优先级等级和精细粒度QoS技术来限制由新主机所消耗的带宽免受现有主机的固定保留的带宽的干扰。

[0056] 第三示例是使用如上所述的系统和方法来将物理DRAM设备分隔成许多个逻辑上独立的DRAM设备。主机设备可以被分配给这些逻辑上独立的DRAM设备,并且因而保护了它们免受由分配给其它分区的主机设备所引起的带宽扰动。一旦完成这样的分区,在不干扰其它分区上的主机设备的情况下可以加入并移除主机设备。

[0057] 将理解,在DRAM存储器设备的上下文中如上所述的系统和方法也可以被应用到其它类型的存储器设备,包括静态随机存取存储器(SRAM)、闪存、磁盘存储器、磁带存储器和

其它类型的易失的和非易失的存储器设备。这样,如上所述的DRAM数据总线可以是存储器数据总线,如上所述的DRAM存储器请求可以是存储器请求,如上所述的DRAM命令可以是存储器命令,并且如上所述的DRAM命令总线可以是存储器命令总线,这些各种各样存储器组件被配置为与特定类型的存储器设备一起工作,诸如在本段落中的上述列出的那些。

[0058] 图7示出了其中上述系统和方法 and 实现的计算环境。如图7中所示,计算系统700可不仅包括具有处理器720、非易失存储器722和易失存储器724的主存系统718,还可包括显示子系统706、通信系统708、以及输入子系统710,全部由通信总线子系统702链接。

[0059] 系统10可采取以下形式:一个或多个个人计算机、服务器计算机、平板计算机、家庭娱乐计算机(诸如有线机顶盒、数字视频记录器、视频服务器等)、网络计算设备、诸如游戏控制台之类的游戏设备、移动计算设备、移动通信设备(例如,智能电话)和/或其它计算设备。

[0060] 主存系统718的处理器720是配置用于使用非易失存储器722和易失存储器724的各部分执行指令的物理设备。例如,处理器可以被配置成执行指令,所述指令是一个或多个应用、服务、程序、例程、库、对象、部件、数据结构或其他逻辑构造的一部分。这种指令可被实现以执行任务、实现数据类型、转换一个或多个部件的状态、实现技术效果、或以其它方式得到期望结果。

[0061] 非易失存储器722包括被配置成保存可由处理器720执行的指令的一个或多个物理设备。非易失存储器722可以包括诸如FLASH, EPROM, EEPROM等的非易失存储器,它们在处理期间将指令保存有限时间段,并且即使当对计算系统断电它们也继续保存所述指令。易失存储器724可以包括DRAM,诸如如上所讨论的DRAM设备32,或在处理期间将指令保存有限时间段但在计算机断电后通常不继续存储指令的RAM。

[0062] 处理器720、永久存储器722和易失存储器724的各方面可以被一起集成到一个或多个硬件逻辑组件中。这种硬件逻辑组件可以包括例如场可编程门阵列(FPGA)、程序和应用专用集成电路(PASIC/ASIC)、程序和应用专用标准产品(PSSP/ASSP)、片上系统(SoC)以及复杂可编程逻辑器件(CPLD)。

[0063] 显示子系统706可用于呈现由永久存储器722或易失存储器724保存的数据的视觉表示。该视觉表示可采用图形用户界面(GUI)的形式。由于此处所描述的方法和过程改变了由非易失存储器722或易失存储器724保持的数据,并由此变换了非易失存储器722或易失存储器724的状态,因此同样可以转变显示子系统706的状态以视觉地表示底层数据的改变。显示子系统706可以包括使用实质上任何类型的技术的一个或多个显示设备。可将这样的显示设备与处理器720和/或非易失存储器722或易失存储器724组合在共享封装中,或这样的显示设备可以是外围显示设备。

[0064] 输入子系统710可包括诸如键盘、鼠标、触摸屏或游戏控制器等一个或多个用户输入设备或者与这些用户输入设备对接。在一些实施例中,输入子系统可以包括或相接于所选择的自然用户输入(NUI)部件。这样的部件可以是集成式的或者是外设,并且输入动作的转换和/或处理可以在板上或板下处理。示例NUI部件可包括用于语言和/或语音识别的话筒;用于机器视觉和/或姿势识别的红外、色彩、立体显示和/或深度相机;用于运动检测和/或意图识别的头部跟踪器、眼睛跟踪器、加速计和/或陀螺仪;以及用于评估脑部活动的电场感测部件。

[0065] 通信子系统708可以被配置成将系统10与一个或多个其他计算设备通信地耦合。通信子系统708可以包括与一个或多个不同通信协议兼容的有线和/或无线通信设备。作为非限制性示例,通信子系统708可被配置成用于经由无线网络或者有线或无线局域网或广域网来进行通信。在一些实施例中,通信子系统可允许计算系统708经由诸如因特网这样的网络将消息发送至其他设备以及/或者从其它设备接收消息。

[0066] 将会理解,此处描述的配置和/或方法本质是示例性的,这些具体实施例或示例不应被视为限制性的,因为许多变体是可能的。此处描述的具体例程或方法可以表示任何数量的处理策略中的一个或多个。如此,所示和/或所述的各种动作可以以所示和/或所述顺序、以其它顺序、并行地执行,或者被省略。同样,上述过程的次序可以改变。

[0067] 本公开的主题包括各种过程、系统和配置以及此处公开的其它特征、功能、动作和/或属性、以及它们的任一和全部等价物的所有新颖且非显而易见的组合和子组合。

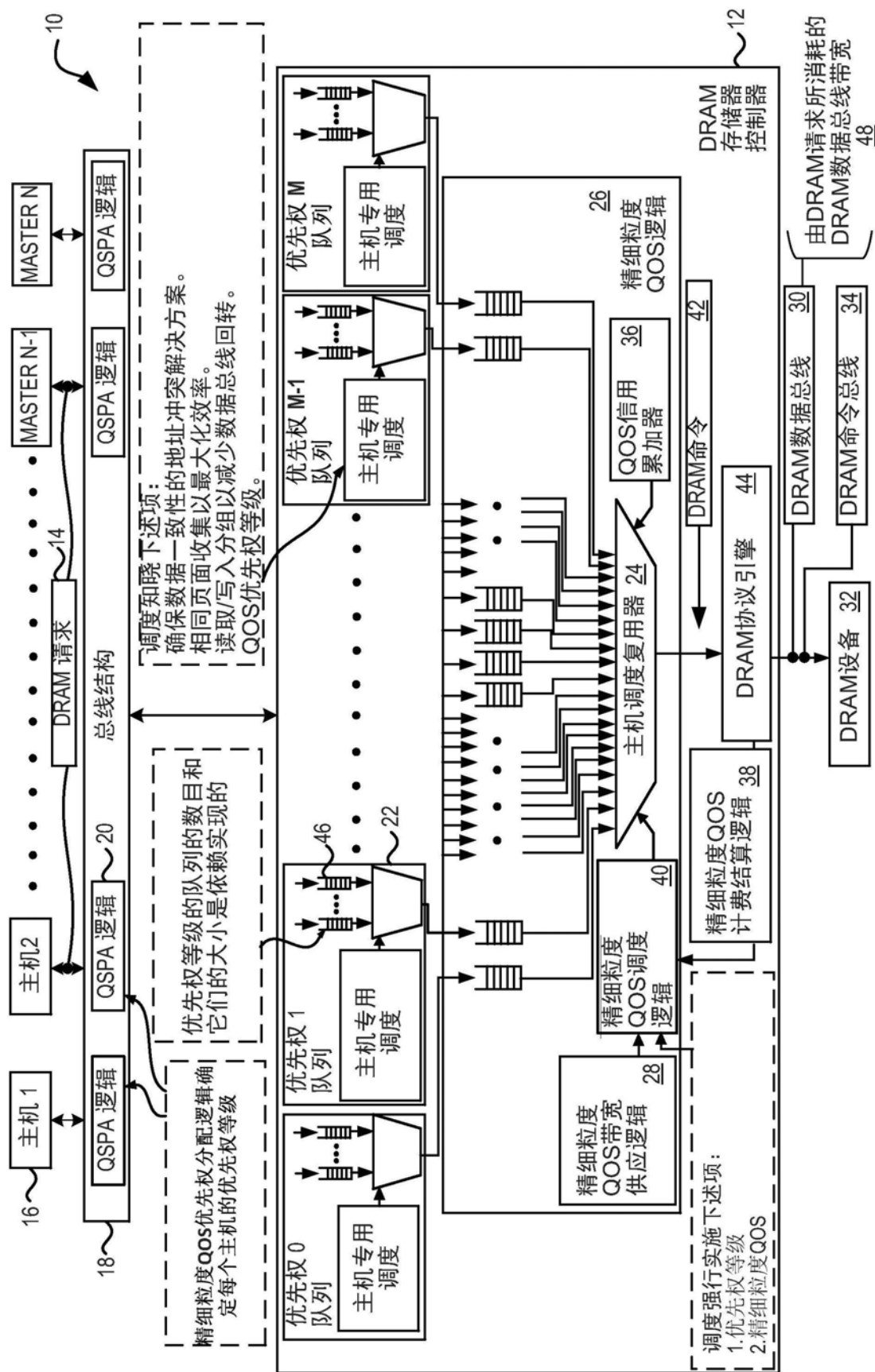


图1

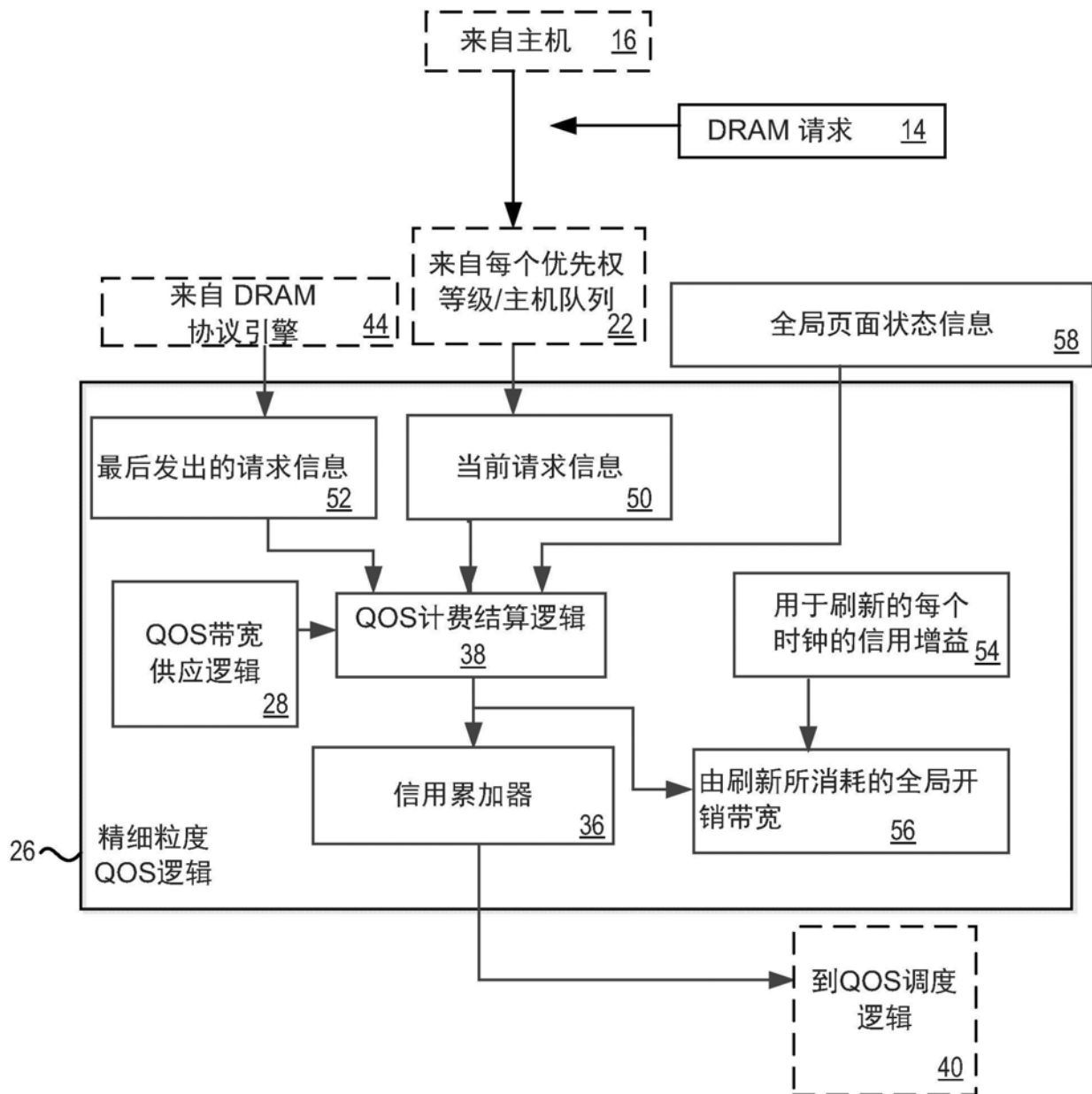


图2

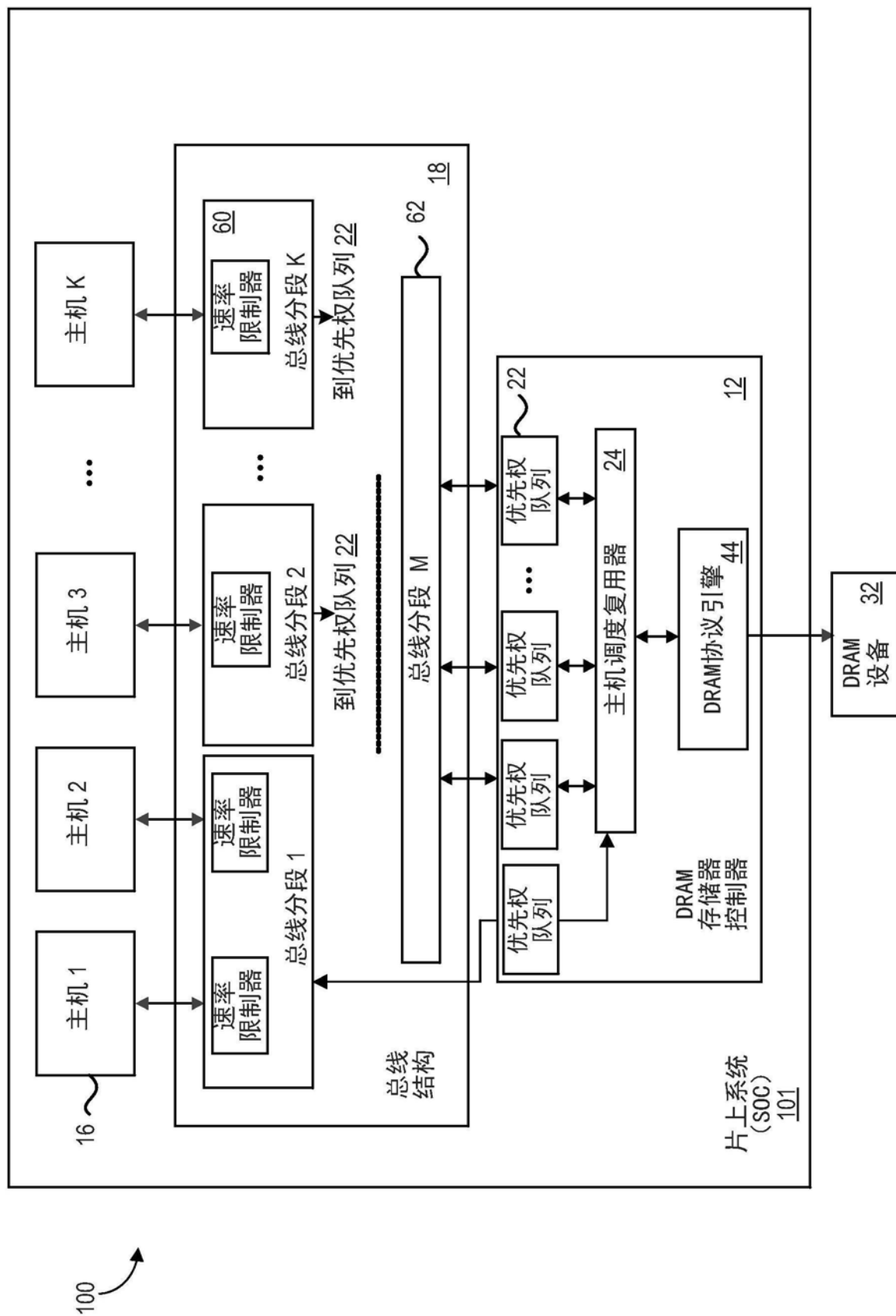


图3

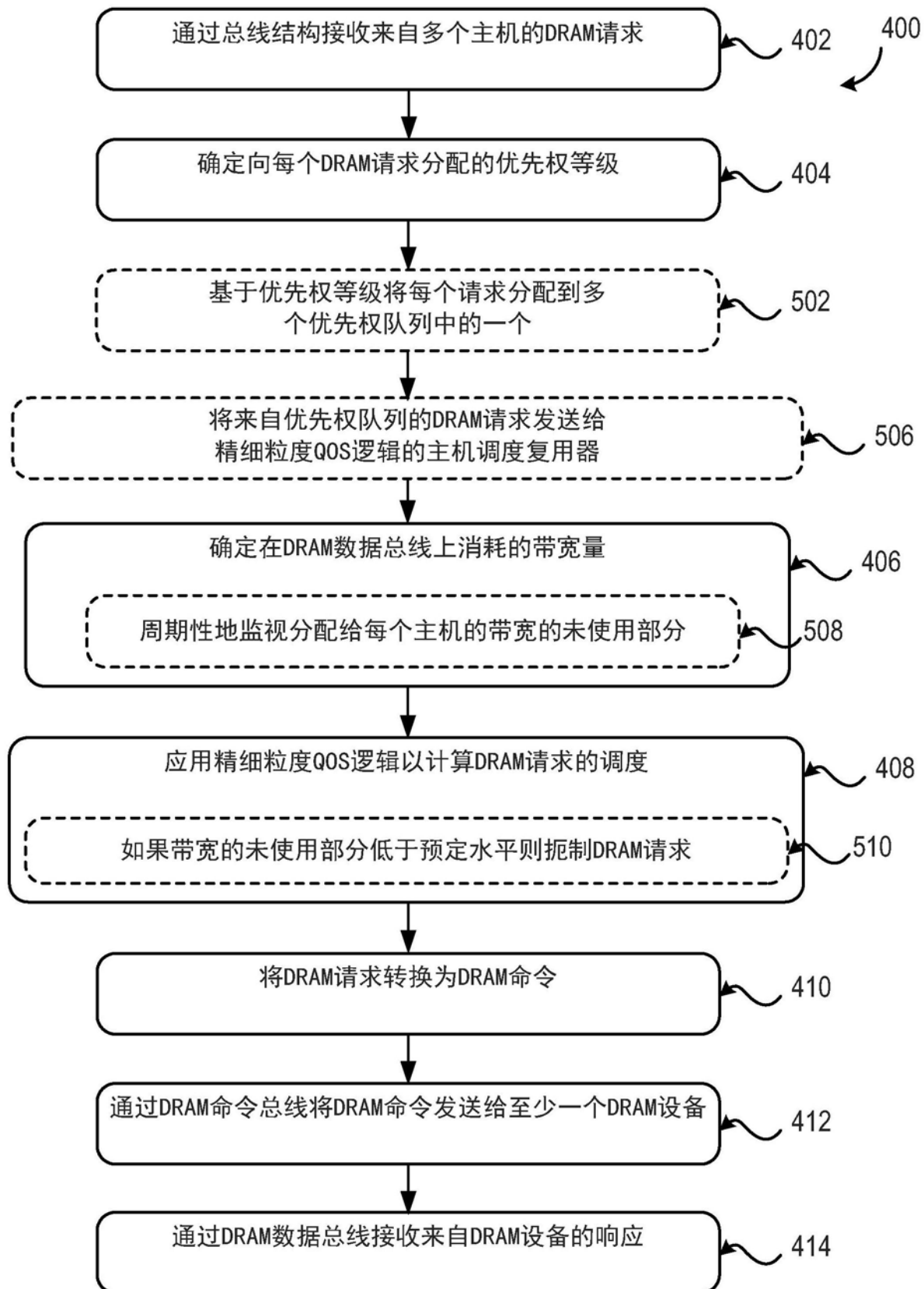


图4

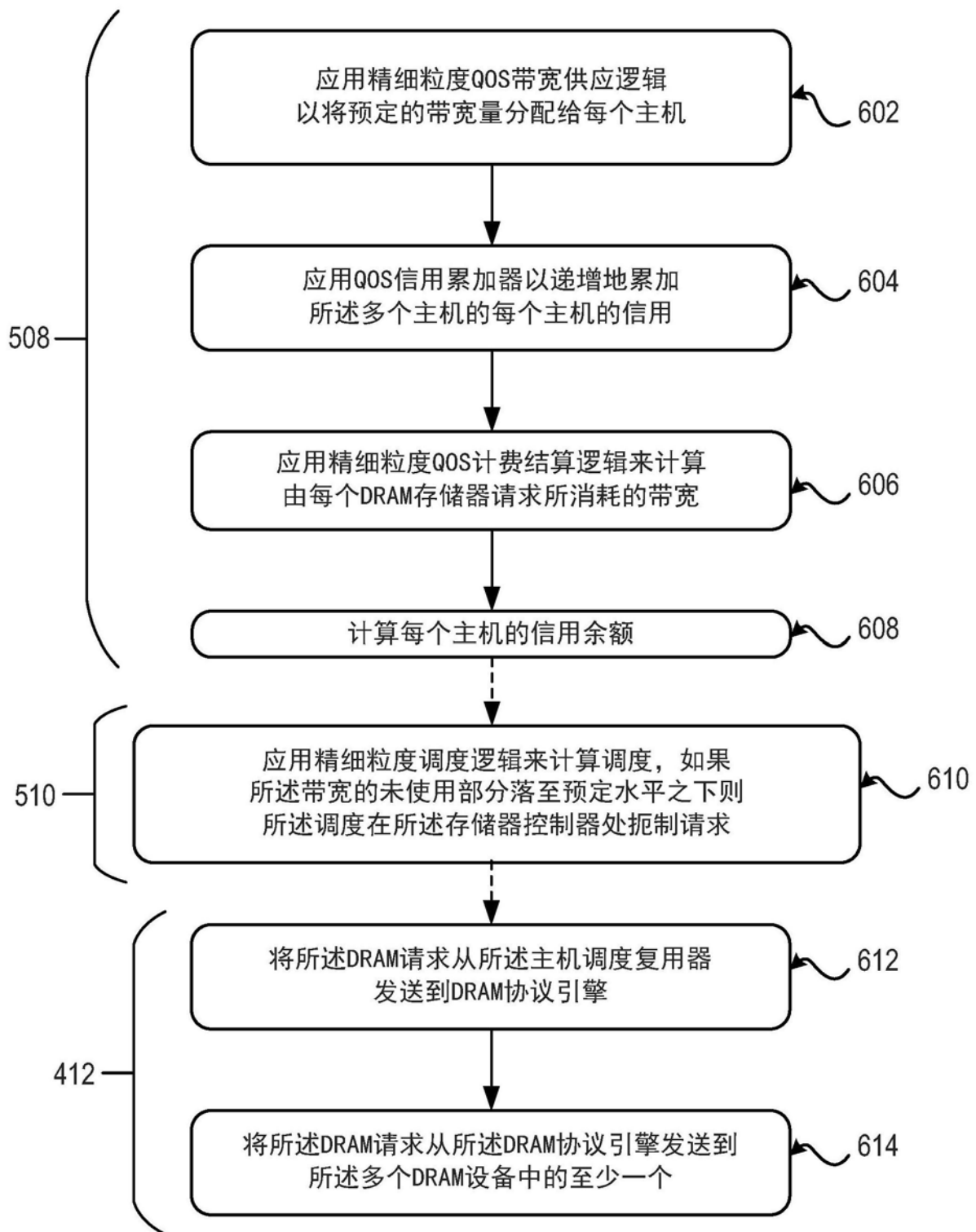


图5

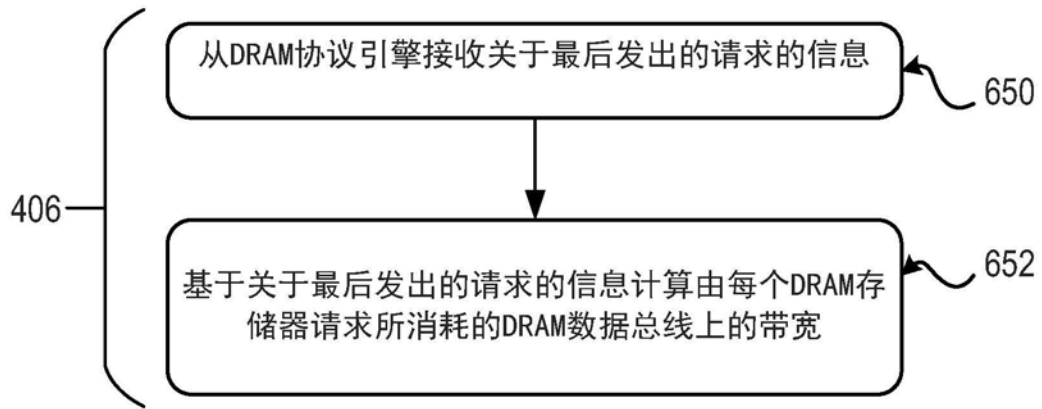


图6

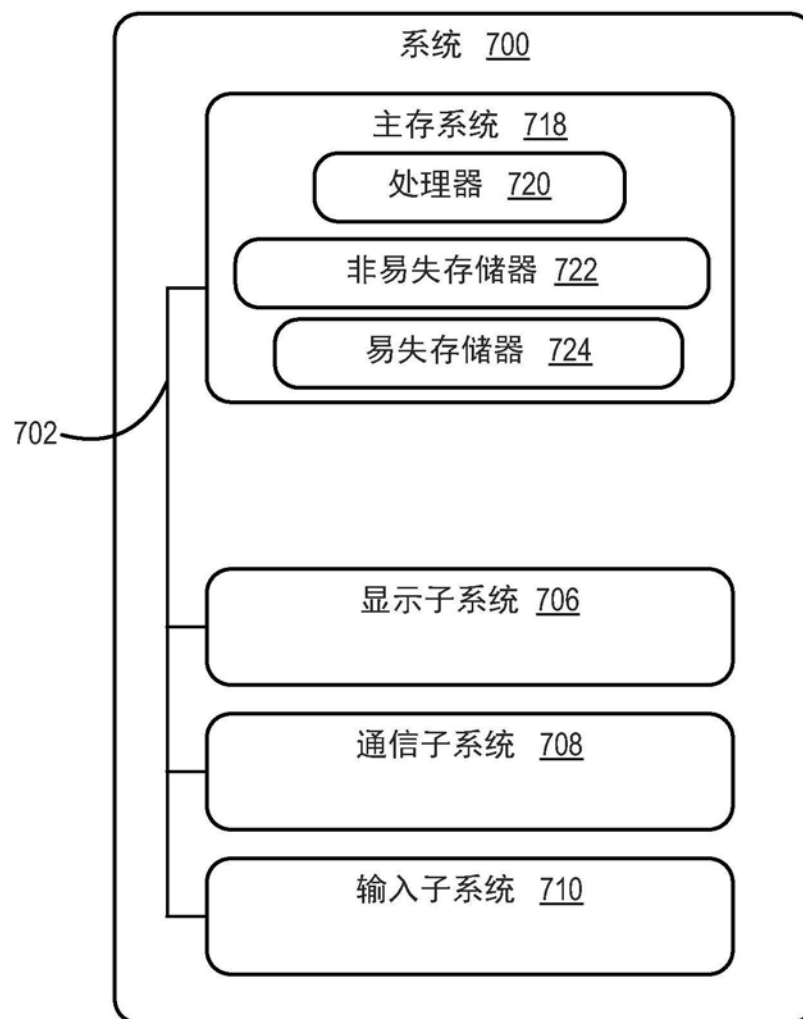


图7