



## (12)发明专利

(10)授权公告号 CN 104778013 B

(45)授权公告日 2019.06.25

(21)申请号 201510021504.6

(22)申请日 2015.01.15

(65)同一申请的已公布的文献号  
申请公布号 CN 104778013 A

(43)申请公布日 2015.07.15

(30)优先权数据  
61/927,636 2014.01.15 US  
14/560,674 2014.12.04 US

(73)专利权人 三星电子株式会社  
地址 韩国京畿道水原市

(72)发明人 林江 马修·加勒特

(74)专利代理机构 北京铭硕知识产权代理有限公司 11286  
代理人 李云霞 王艳娇

(51)Int.Cl.

G06F 3/06(2006.01)

(56)对比文件

CN 1509436 A, 2004.06.30,  
CN 101923499 A, 2010.12.22,  
CN 101123113 A, 2008.02.13,  
CN 1858720 A, 2006.11.08,  
US 2010250857 A1, 2010.09.30,  
CN 101082882 A, 2007.12.05,

审查员 赵鹏翔

权利要求书3页 说明书7页 附图4页

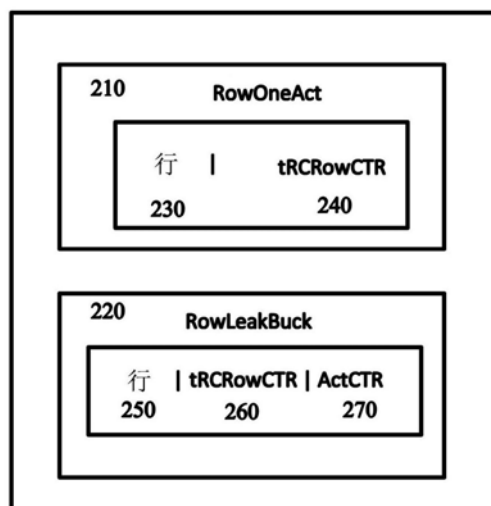
(54)发明名称

追踪对存储器的行的激活的方法和设备

(57)摘要

一种追踪对存储器的行的激活的方法和设备,使用更少数量的行激活计数器来有效追踪对于存储器的行的激活,所述行激活计数器指示存储器行是否在激活时间段期间被激活,并且行激活计数器指示在最大激活窗口内存储器行的被允许激活的次数。

**200**



1. 一种用于控制存储器的存储器控制器,所述存储器控制器包括:  
页表,被配置为存储以下项:  
第一条目表,包括第一条目,第一条目包括:  
第一标识符,标识存储器的在第一时间段期间被激活的第一存储器行;  
超时计数器,指示在第一时间段期间剩余的第一时间;  
第二条目表,包括第二条目,第二条目包括:  
第二标识符,标识存储器的在至少一个第二时间段期间被激活的第二存储器行;  
激活计数器,对第二存储器行的激活次数进行计数;  
第二超时计数器,指示在所述至少一个第二时间段期间剩余的第二时间。
2. 如权利要求1所述的存储器控制器,其中,第二存储器行的激活次数基于在最大激活窗口期间第二存储器行的被允许激活的最多次数被重置,其中,所述最大激活窗口是这样的一段时间:在该段时间期间第二存储器行的被允许激活的最多次数被允许,而不会使存储器的位于与第二存储器行相邻的相邻存储器行中存储的数据恶化。
3. 如权利要求1所述的存储器控制器,其中,第二存储器行的激活次数基于在最大激活窗口期间第二存储器行的被允许激活的最多次数被重置,其中,所述最大激活窗口是这样的一段时间:在该段时间期间第二存储器行的被允许激活的最多次数被允许,而不用刷新存储器的位于与第二存储器行相邻的相邻存储器行。
4. 如权利要求3所述的存储器控制器,其中,所述至少一个第二时间段中的每个第二时间段是在所述最大激活窗口内第二存储器行的最多次数的被允许激活中的每次激活之间的平均时间量。
5. 如权利要求4所述的存储器控制器,其中,所述存储器控制器被配置为:接收对于激活存储器的存储器行的请求,确定所述存储器行是否与第二条目的第二标识符所标识的第二存储器行匹配,确定激活计数器是否指示被允许激活的最多次数,并且响应于确定激活计数器指示被允许激活的最多次数,执行对存储器的刷新操作和对所述请求的限制操作中的一个操作。
6. 如权利要求5所述的存储器控制器,其中,所述存储器控制器还被配置为:响应于确定激活计数器没有指示被允许激活的最多次数而使激活计数器的计数值增加,并激活第二存储器行。
7. 如权利要求4所述的存储器控制器,其中,所述存储器控制器被配置为:接收对于激活存储器的存储器行的请求,确定所述存储器行与第一条目的第一标识符所标识的第一存储器行匹配,取消对于第一条目的分配,在第二条目表中分配所述存储器行的相应条目,并激活所述存储器行。
8. 如权利要求4所述的存储器控制器,其中,所述存储器控制器被配置为:接收对于激活存储器的存储器行的请求,确定所述存储器行与第一条目的第一标识符所标识的第一存储器行不匹配,并在第一条目表中分配所述存储器行的相应条目。
9. 如权利要求4所述的存储器控制器,其中,所述存储器控制器被配置为:接收对于激活存储器的存储器行的请求,确定所述存储器行与第一条目的第一标识符所标识的第一存储器行匹配,确定第二条目表不能分配与第一存储器行相应的附加条目,并响应于确定第二条目表不能分配与第一存储器行相应的附加条目而执行对存储器的刷新操作和对所述

请求的限制操作中的一个操作。

10. 如权利要求4所述的存储器控制器,其中,所述存储器控制器还被配置为:从第二超时计数器确定所述至少一个第二时间段已经过去,响应于从第二超时计数器确定所述至少一个第二时间段已经过去而使激活计数器的计数值减少。

11. 一种追踪对存储器的行的激活的方法,所述方法包括:

在存储器中存储页表,所述页表被配置为存储以下项:

第一条目表,包括第一条目,第一条目包括:

第一标识符,标识存储器的在第一时间段期间被激活的第一存储器行;

超时计数器,指示在第一时间段期间剩余的第一时间;

第二条目表,包括第二条目,第二条目包括:

第二标识符,标识存储器的在至少一个第二时间段期间被激活的第二存储器行;

激活计数器,对第二存储器行的激活次数进行计数;

第二超时计数器,指示在所述至少一个第二时间段期间剩余的第二时间;

使用所述页表来追踪对存储器的行的激活。

12. 如权利要求11所述的方法,其中,第二存储器行的激活次数基于在最大激活窗口期间第二存储器行的被允许激活的最多次数被重置,其中,所述最大激活窗口是这样的一段时间:在该段时间期间第二存储器行的被允许激活的最多次数被允许,而不会使存储器的位于与第二存储器行相邻的相邻存储器行中存储的数据恶化。

13. 如权利要求11所述的方法,其中,第二存储器行的激活次数基于在最大激活窗口期间第二存储器行的被允许激活的最多次数被重置,其中,所述最大激活窗口是这样的一段时间:在该段时间期间第二存储器行的被允许激活的最多次数被允许,而不用刷新存储器的位于与第二存储器行相邻的相邻存储器行。

14. 如权利要求13所述的方法,其中,所述至少一个第二时间段中的每个第二时间段是在所述最大激活窗口内第二存储器行的最多次数的被允许激活中的每次激活之间的平均时间量。

15. 如权利要求14所述的方法,其中,追踪步骤包括:

接收对于激活存储器的存储器行的请求;

确定所述存储器行与第二条目的第二标识符所标识的第二存储器行匹配;

确定激活计数器是否指示被允许激活的最多次数;

响应于确定激活计数器指示被允许激活的最多次数,执行对存储器的刷新操作和对所述请求的限制操作中的一个操作。

16. 如权利要求15所述的方法,其中,追踪步骤还包括:

响应于确定激活计数器没有指示被允许激活的最多次数而使激活计数器的计数值增加,并激活第二存储器行。

17. 如权利要求14所述的方法,其中,追踪步骤包括:

接收对于激活存储器的存储器行的请求;

确定所述存储器行与第一条目的第一标识符所标识的第一存储器行匹配;

取消对于第一条目的分配;

在第二条目表中分配所述存储器行的相应条目;

激活所述存储器行。

18. 如权利要求14所述的方法,其中,追踪步骤包括:

接收对于激活存储器的存储器行的请求;

确定所述存储器行与第一条目的第一标识符所标识的第一存储器行不匹配;

在第一条目表中分配所述存储器行的相应条目。

19. 如权利要求14所述的方法,其中,追踪步骤包括:

接收对于激活存储器的存储器行的请求;

确定所述存储器行与第一条目的第一标识符所标识的第一存储器行匹配;

确定第二条目表不能分配与第一存储器行相应的附加条目;

响应于确定第二条目表不能分配与第一存储器行相应的附加条目而执行对存储器的刷新操作和对所述请求的限制操作中的一个操作。

20. 如权利要求14所述的方法,其中,追踪步骤包括:

从第二超时计数器确定所述至少一个第二时间段已经过去;

响应于从第二超时计数器确定所述至少一个第二时间段已经过去而使激活计数器的计数值减少。

## 追踪对存储器的行的激活的方法和设备

[0001] 本申请要求于2014年1月15日在美国专利商标局提交的第61/927,636号美国临时申请和2014年12月4日提交的第14/560,674号美国专利申请的权益,该申请的公开全部合并于此以资参考。

### 技术领域

[0002] 与示例性实施例一致的方法和设备涉及追踪对于存储器的行的激活,更具体地,涉及一种使用更少数量的行激活计数器来有效地追踪对于存储器的行的激活的方法和设备。

### 背景技术

[0003] 当动态随机存取存储器 (DRAM) 的目标行在一段时间内被激活太多次时,存储在物理上与目标行相邻的相邻行的数据会被干扰或丢失。具体地,由于DRAM设计的增加的密度,存储在相邻行的数据会由于目标行的频繁激活而产生的噪声而损坏。

[0004] 因此,对于某些DRAM装置,制造商可基于例如DRAM的架构和目标行的激活的速率(以该速率激活目标行可能使DRAM的相邻行中存储的数据损坏)来确定在一段时间内对于目标行的最多激活次数。一旦达到在一段时间的激活的次数,相邻行应被刷新以避免在相邻行中存储的数据的损坏。

[0005] 通常在一段时间内对目标行的最多激活次数可被称为最大激活计数 (MAC)。MAC是在相邻行应被刷新以避免数据损坏之前,在一段时间内对于目标行的最多激活次数,即,最大激活窗口 (tMAW)。

[0006] 当一个或两个“干扰 (aggressor)”行的目标是一个“受害 (victim)”行时,会发生与由于目标行的频繁激活而导致存储在相邻行中的数据损坏相关联的情况。例如,相邻受害行被布置于其间的多个目标干扰行可被频繁激活。因此,针对给定受害行的两个干扰行的激活的总数不应超过MAC,这是因为对于两个干扰行的激活会造成存储在相邻受害行的数据的损坏。

[0007] 如果MAC限制达到最大激活窗口,在向目标行发送另一激活之前,存储器控制器可刷新DRAM中的所有行,执行目标行刷新 (TRR) 以仅对与过度活跃的目标行相邻的行进行刷新,或者限制对于目标行的激活使得不会达到MAC限制,这避免了必须对所有行执行完全刷新或TRR,但是这会使用行激活的实现延迟。

[0008] 在以上情景中,由于控制对于DRAM中的行的激活的存储器控制器不知道行的物理定向,因此管理针对受害行的MAC很复杂。这样,存储器控制器不能追踪针对关于给定的受害行的两个干扰行的激活。为了防止超过针对受害行的MAC,可在最大激活窗口中将有效MAC值设置为MAC/2激活次数。

[0009] 管理MAC限制通常需要在最大激活窗口内追踪已经被发送到DRAM的每一行的激活次数。为了真实地追踪对于每一行的激活次数,需要用于每个存储体 (bank) 的每一行以及用于DRAM中的每个组 (rank) 的计数器。在具有总计4个组、每个组有16个存储体、每个存储

体有128K行的DRAM中,计数器的数量将会是 $4 \times 16 \times 128K = 8192K$ 个计数器,每个计数器需要N位。这样的数量的计数器和对于它们的管理所需的相关联的空间从存储器控制器设计的功率/面积方面来说很快就变得无法管理。

## 发明内容

[0010] 示例性实施例可克服以上缺点。然而,示例性实施例不必克服以上缺点。

[0011] 根据示例性实施例的一方面,提供了一种用于控制存储器的存储器控制器,所述存储器控制器包括:页表,被配置为存储以下项:第一条目表,包括第一条目,所述第一条目包括:第一标识符,标识存储器在第一时间段期间被激活的第一存储器行;超时计数器,指示在第一时间段期间的第一剩余时间;第二条目表,包括第二条目,第二条目包括:第二标识符,标识存储器在至少一个第二时间段期间被激活的第二存储器行;激活计数器,对第二存储器行的激活次数进行计数;第二超时计数器,指示所述至少一个第二时间段期间剩余的第二时间。

[0012] 根据示例性实施例的一方面,提供了一种追踪对存储器的行的激活的方法,所述方法包括:在存储器中存储页表,所述页表被配置为存储以下项:第一条目表,包括第一条目,所述第一条目包括:第一标识符,标识存储器在第一时间段期间被激活的第一存储器行;超时计数器,指示在第一时间段期间剩余的第一时间;第二条目表,包括第二条目,第二条目包括:第二标识符,标识存储器在至少一个第二时间段期间被激活的第二存储器行;激活计数器,对第二存储器行的激活次数进行计数;第二超时计数器,指示在所述至少一个第二时间段期间剩余的第二时间;使用所述页表来追踪对于存储器的行的激活。

## 附图说明

[0013] 通过参照附图详细描述示例性实施例,以上和/或其它方面将会变得更清楚,其中:

[0014] 图1示出了根据示例性实施例的DRAM的时序参数。

[0015] 图2示出根据示例性实施例的页表。

[0016] 图3示出根据示例性实施例的追踪对于存储器的行的激活的方法。

[0017] 图4示出根据示例性实施例的存储器控制器。

[0018] 图5示出根据示例性实施例的第四代双倍数据速率(DDR4)DRAM的装置参数。

[0019] 图6示出根据示例性实施例的采用追踪对于存储器的行的激活的方法和图5的装置参数的性能数据测试结果。

## 具体实施方式

[0020] 以下,参照附图来更完整地描述示例性实施例,其中,相同的标号始终表示相同的元件。当诸如“…中的至少一个”的表达出现在列出的元件之后时,修饰列出的全部元件而不是修饰列出的单个元件。

[0021] 图1示出了根据示例性实施例的DRAM的时序参数。

[0022] DRAM时序参数限制在最大激活窗口(tMAW)中可被声明用于DRAM中的目标行的激活次数,通过理解各种DRAM时序参数,可减少为了避免违背MAC限制或显著影响整个DRAM系

统的性能而需要被同时追踪的DRAM的行的数量。

[0023] 如图1所示,DRAM的时序参数可包括DRAM最大激活窗口 (tMAW) 和在tMAW内对于行的最多激活次数 (tMAC)。如上面讨论的,可由DRAM的制造商确定这样的参数。

[0024] 为了实现对于MAC的行激活追踪,根据示例性实施例的存储器控制器包括作为对于相同目标行的激活之间的平均时间量的附加时序参数tRCRow, tRCRow被计算为如下:

$$[0025] \quad tRCRow = \frac{tMAW}{\left(\frac{tMAC}{2}\right)}$$

[0026] 如上面记载的,值 $\left(\frac{tMAC}{2}\right)$ 可以是有效MAC值。

[0027] tRCRow的值可如上所示的被初始化,并且tRCRow的值可根据在tRCRow窗口期间已被激活的行的激活次数 (ActCTR) 在执行期间被重置,如下所示:

$$[0028] \quad tRCRow = \frac{tMAW}{\left(\frac{tMAC - ActCTR}{2}\right)}$$

[0029] 包括在tRCRow计算中的ActCTR值基于可被页表结构计数的最多激活次数,所述页表结构存储用于指示在一个或更多个tRCRow窗口期间已被激活的特定行的激活次数的条目,将在下面进行讨论。这由ActCTR实现方式的宽度确定。因此,tRCRow的值可基于对于在tRCRow窗口期间已被激活的行的激活次数 (ActCTR) 的最大值被重置。

[0030] 附加时序参数可包括刷新闻隔 (tREFI)、完成刷新命令所需的时间 (tRFC)、功率限制传递参数 (tRRD\_L和tFAW)、更新时序参数 (tMOD) 和读延迟时序参数 (tRCD)、存储体时序参数 (tRAS、tRP和tRC)。再一次,可基于架构选择由DRAM的制造商来确定这样的参数。

[0031] 图2示出根据示例性实施例的页表。

[0032] 页表200可包括每存储体两个页表结构,以使用更少数量的行激活计数器来有效地追踪对于存储器的行的激活。

[0033] 页表结构 (RowOneAct) 210可存储指示在tRCRow窗口期间已被激活的行的条目。例如,RowOneAct 210页表结构可以是存储体的16条目结构,用于在tRCRow窗口期间追踪行的激活。

[0034] RowOneAct 210页表结构可将在tRCRow窗口期间已被激活的行230与超时计数器 (tRCRowCTR) 240关联,以针对行230追踪tRCRow窗口的过期。

[0035] 当存储器行的激活发生时,存储器控制器可将条目实例化到针对行230的RowOneAct 210页表结构,并将超时计数器240初始化为与tRCRow相等的值。一旦基于超时计数器240确定tRCRow失效,存储器控制器还可从RowOneAct 210页表结构清除条目。因此,存储器控制器可查询RowOneAct 210页表结构以识别在tRCRow窗口期间已被激活的行。

[0036] 第二条目 (RowLeakBuck) 220可存储用于指示在一个或更多个tRCRow窗口期间已被激活的特定行的激活次数的条目。

[0037] RowLeakBuck 220页表结构可将在tRCRow窗口期间已被激活的行250、用于针对行250追踪一个或更多个tRCRow窗口的过期的超时计数器 (tRCRowCTR) 260和对在tRCRow窗口期间已被激活的特定行250的激活次数 (ActCTR) 270进行计数的激活计数器关联。当条目被

实例化到RowLeakBuck 220页表结构中时,超时计数器260可被初始化为与tRCRow相等的值,或者可选择地,如果如以下将要讨论的,条目从RowOneAct 210页表结构被移动到RowLeakBuck 220页表结构,则超时计数器260可被初始化为与超时计数器240的值相等的值。另外,当条目被实例化到RowLeakBuck 220页表结构中时,激活计数器270可被初始化为1。

[0038] 例如,RowLeakBuck 220页表结构可以是存储体的8条目结构,存储体的8条目结构针对RowLeakBuck 220页表结构中的每个行条目250采用8位激活计数器270。

[0039] 每次与RowLeakBuck 220页表结构中的条目250匹配的特定行被激活时,激活计数器270可被增加例如1。如以上提到的,一旦与条目250匹配的特定行被激活,超时计数器260可被初始化。相反,随着超时计数器260失效,在每个tRCRow窗口,激活计数器270可被减少例如1。

[0040] 在超时计数器260失效时,如果激活计数器270仍然指示非零的激活次数,则超时计数器260可被重置为与tRCRow相等的值。如以上讨论的,根据非零的激活次数,超时计数器260的值可基于tRCRow窗口的值被重置。

[0041] 通过说明如激活计数器270所指示的激活的数量,tRCRow窗口可被控制以保证最多激活次数tMAC不会在DRAM最大激活窗口tMAW内发生。

[0042] 在超时计数器260过期并重置超时计数器260时使激活计数器270减少的操作可被迭代地重复,直到激活计数器270的值达到零为止。当激活计数器270达到零时,可从RowLeakBuck 220页表结构移除相应行250的条目。

[0043] 因此,存储器控制器可查询RowLeakBuck 220页表结构以识别高度活跃的行的激活次数,并识别潜在的过度活跃的行,过度活跃的行需要刷新或限制激活以防止相邻行的数据损坏。

[0044] 图3示出根据示例性实施例的追踪对于存储器的行的激活的方法。

[0045] 如以上讨论的,存储器控制器可执行追踪对于存储器的行的激活的方法以防止由于对于行的频繁访问而导致的数据损坏。

[0046] 在步骤S305,存储器控制器接收对于行的存储器访问的请求。

[0047] 在步骤S310,存储器控制器查询第二页表结构220以确定被请求的行是否与第二页表结构220中的现有条目的行250匹配。

[0048] 如果确定被请求的行与第二页表结构220中的现有条目的行250匹配(步骤S310-是),则在步骤S315,存储器控制器通过将激活计数器270与阈值比较来确定被请求的行是否是过度活跃的行。如以上讨论的,阈值可以是指示对于被请求的行的激活次数达到最大值的激活计数器270的值。

[0049] 如果存储器控制器确定对于被请求的行的激活次数小于阈值(步骤S315-否),则存储器控制器在步骤S320使针对第二页表结构220的条目中的存储器行250的激活计数器270增加,并在步骤S355执行行激活。

[0050] 另一方面,如果存储器控制器确定对于被请求的行的激活次数达到阈值(步骤S315-是),则存储器控制器可在步骤S325执行存储器激活的刷新或限制。这里,如果存储器控制器执行刷新,则激活计数器270可被重置为零,或者行250的条目可被从第二页表结构220取消分配。可选择地,如果存储器控制器执行限制,则如基于超时计数器260指示的窗口

的连续失效确定的,存储器控制器可限制行的激活,直到激活计数器270达到可接受的较低的阈值为止。

[0051] 回到存储器控制器接收到对于存储器访问的请求(步骤S305),如果存储器控制器确定被请求的行与第二页表结构220中的现有条目不匹配(步骤S310-否),则在步骤S330,存储器控制器确定被请求的行是否与第一页表结构210中的现有条目匹配。

[0052] 如果确定被请求的行与第一页表结构210中的现有条目的行230匹配(步骤S330-是),从而指示被请求的行已经在当前窗口期间被激活,则在步骤S350存储器控制器确定是否可在第二页表结构220中分配附加条目以追踪激活次数。

[0053] 如果存储器控制器确定第二页表结构220的活跃条目已满,因此不可在第二页表结构220中分配附加的条目(步骤S350-是),则存储器控制器可在步骤S325执行存储器激活的刷新或限制。这里,如果存储器控制器执行刷新,则可从第一页表结构210取消对于行230的条目的分配。可选择地,如果存储器控制器执行限制,则如基于超时计数器240确定的,存储器控制器可限制行的激活直到窗口过期为止,并且可从第一页表结构210取消对于行230的条目的分配。

[0054] 另一方面,如果存储器控制器确定第二页表结构220的活跃条目未满(步骤S350-否),则在步骤S345,存储器控制器可从第一页表结构210取消对于行230的条目的分配,并在第二页表结构220中针对被请求的行分配相应的新条目,并在步骤S355执行行激活。

[0055] 返回到步骤S330存储器控制器确定被请求的行是否与第一页表结构210中的现有条目匹配,如果被请求的行与第一页表结构中的现有条目不匹配(步骤S330-否),则在步骤S335,存储器控制器确定是否可在第一页表结构210中分配附加条目来追踪在窗口期间行是否已被激活。

[0056] 如果存储器控制器确定第一页表的活跃条目已满,因此在第一页表结构210中不可分配附加条目(步骤S335-是),则在步骤S350,存储器控制器确定是否可在第二页表结构220中分配附加条目来追踪激活次数,并执行如上讨论的后续确定操作。

[0057] 或者,如果存储器控制器确定第一页表结构210的活跃条目未满(步骤S335-否),则在步骤S355,存储器控制器可在第一页表结构210中分配针对被请求的行的新条目,并执行行激活。

[0058] 图4示出根据示例性实施例的存储器控制器。

[0059] 如图4所示,存储器控制器400包括系统接口410、地址转译单元420、主请求队列430、裁决器440、DDR物理层器件(DFI)接口450、选择器460和页表470。

[0060] 图4中示出的页表470与图2中示出的页表200相似,因此省略对其的详细描述。

[0061] 系统接口410是存储器控制器400所属的系统的附加组件的接口,消息和数据通过系统接口410被发送和接收。

[0062] 地址转译单元420对通过系统接口410接收的存储器请求执行地址转译。

[0063] 主请求队列430存储通过系统接口410接收并被地址转译单元420转译的存储器请求。

[0064] 裁决器440将请求信号(Pick Req)输出到选择器460,以从存储在主请求队列430中的、经过DFI接口450处理的多个请求中进行选择。

[0065] 可结合存储在主请求队列中的请求来查询页表470,以确定是否激活与存储器请

求相应的存储器行。已经针对图3在上面讨论了激活确定的详细描述,因此省略其详细描述。

[0066] 页表470还可从DFI接口450接收关于存储器行的激活的反馈。

[0067] 图5示出根据示例性实施例的第四代双倍数据速率(DDR 4) DRAM的装置参数。图6示出根据示例性实施例的采用追踪存储器的行的激活的方法和图5的装置参数的性能数据测试结果。

[0068] 如图5所示,示出了DDR4DRAM装置参数。本领域的技术人员将理解,可由制造商确定DDR4DRAM装置参数。

[0069] 图6示出根据示例性实施例的采用追踪对于存储器的行的激活的方法和图5的装置参数的性能数据测试结果。

[0070] 如在此所示出的,使用七种通信量样式示出了针对三种方案(方案1、方案2和方案3)的性能数据:每次激活一个CAS、对于N行的循环访问,其中, $N=128K、16、8、6、4、2、1$ 。性能度量示出在tMAW窗口中可针对存储体发出多少次激活的比较。根据示出的示例性实施例,在一个存储器通道中可存在多于一个存储体,假设所有通信量仅进入一个存储体。

[0071] 根据图6中示出的示例性实施例,第一方案和第二方案(方案1和方案2)是第三方案(方案3)的对比示例,其中,第三方案实现本公开的特征。

[0072] 在第一方案(方案1)中,存储体比正常刷新率更频繁地被刷新以避免达到最大激活计数限制,因此不需要进入刷新(即,TRR)模式。

[0073] 存储体可在 $47ns(t_{RC}) \times 300K = 14.1ms$ 中被激活300K( $t_{MAC}$ )次。如果在该窗口中每一行都被刷新,则保证将不多于300K次激活应用于任意行。

[0074] 刷新的性能开销是 $t_{RFC}/t_{REFI} \times t_{MAW}/14.1 = 20.4\%$ 。正常刷新率的性能开销是 $t_{RFC}/t_{REFI} = 4.5\%$ 。因此,第一方案的性能开销是 $1 - (1 - 20.4\%) / (1 - 4.5\%) = 16.6\%$ 。根据示例性实施例,对于所有列出的访问模式,性能开销是相同的。

[0075] 在第二方案(方案2)中,计数器被用于追踪最近的激活的次数:在每个tRCRow使计数器减少并在激活被发送到行时使计数器增加。每当计数器溢出时启动刷新(即,TRR模式),或者每当计数器达到最大计数器值时限制激活。

[0076] 当计数器宽度为19位时,每个存储体的总存储开销为2432K位( $19\text{位}/\text{行} \times 128K\text{行} = 2432K\text{位}$ )。如果当发生计数器溢出时使用TRR模式,则在tMAW中溢出的次数根据 $(64ms/47ns/300k < 5)$ 被限制,因此性能开销可忽略。为了减少存储开销,可按照每个计数器减少位的数量。然而,当针对相同的通信量样式使用更小的计数器时溢出的次数更多,因此造成更高的性能开销。例如,如果计数器宽度为8,则存储开销是1024K位/存储体,在 $N=1$ 的情况下(即,仅访问一行),性能降级增加到5.5%。

[0077] 如果没有使用TRR模式,则存储器控制器在计数器达到最大值时限制激活。例如,如果 $N=1$ (仅访问一行),一旦该行的计数器达到最大值(300k),则存储器控制器可每个tRCRow(213.33ns)仅发出一次激活,因此性能开销是 $1 - 47ns/213.ns = 77.9\%$ 。

[0078] 在第三方案(方案3)中,实现图3中示出的完整方法。

[0079] 性能开销与第二方案(方案2)的性能开销相等。然而,存储开销大幅降低。使用此方案,仅需要168( $16 + (8 \times 19)$ )个计数器位。因为不是按每行存储计数器,因此采用每条目的行索引(17个行地址位),这增加了另外的408位( $24 \times 17$ )。

[0080] 该方案的总存储量(168+408)仍然明显少于其它的方案。性能开销与第二对比方案(方案2)相同,因为8个计数器(如图2的第二页表结构220中实现的)足够在 $N \leq 4$ 时捕捉到具有高激活次数的行。当 $N > 4$ 时,对于上面讨论的配置来说,存储体不会遇到行串扰(row hammer)问题。此外,示出的方案(方案3)比方案2具有更低的存储量和性能开销,示出的方案(方案3)具有远比第一对比方案(方案1)更低的性能和功率开销。

[0081] 尤其是对于其它配置来说,可调整计数器的数量(如图2的第二页表结构220中实现)。

[0082] 示例性实施例的功能可实施为计算机可读记录介质中的计算机可读代码。计算机可读记录介质包括存储有计算机可读数据的所有类型的记录介质。计算机可读记录介质的示例包括ROM、RAM、CD-ROM、磁带、软盘和光学数据存储器。此外,记录介质可按照诸如在互联网传输中使用的载波的形式被实现。另外,计算机可读记录介质可分布于网络上的计算机系统,在该计算机系统中计算机可读代码可按照分布的方式被存储和执行。

[0083] 如本领域的技术人员将理解的,示例性实施例可通过执行特定任务的软件和/或硬件组件(诸如现场可编程门阵列(FPGA)或专用集成电路(ASIC))的任意组合来实现。单元或模块可有利地被配置为驻留在可寻址存储介质上并可被配置为在一个或多个处理器或微处理器上运行。因此,单元或模块可包括例如组件(诸如软件组件、面向对象的软件组件、类组件和任务组件)、处理、功能、属性、程序、子程序、程序代码段、驱动、固件、微代码、电路、数据、数据库、数据结构、表、阵列和变量。针对组件和单元设置的功能可被组合为更少的组件和单元或模块,或者进一步被分成附加的组件和单元或模块。

[0084] 以上已经描述了多个示例性实施例。然而,应理解,可进行各种修改。例如,如果描述的技术按照不同的顺序并且/或者如果描述的系统、架构、装置或电路中的组件按照不同的方式被组合和/或被其它组件或它们的等同物所取代或补充,则可实现合适的结果。因此,其它的实现方式在权利要求的范围内。

DRAM时序参数		
tMAW	64000000	ns
tMAC	300000	次
tRCRow	213.3333	ns
tREFI	7800	ns
tRFC	350	ns
tRRD_L	5	ns
tFAW	13	ns
tMOD	15	ns
tRAS	32	ns
tRP	15	ns
tRCD	15	ns
tRC	47	ns

图1

**200**

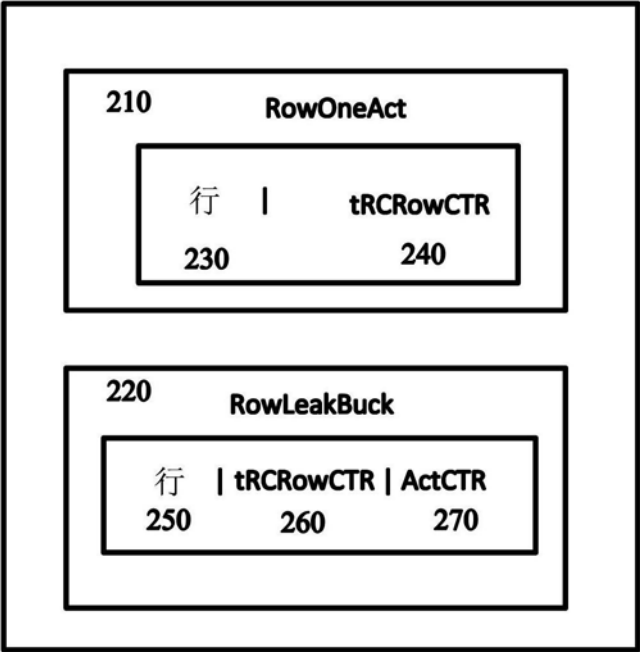


图2

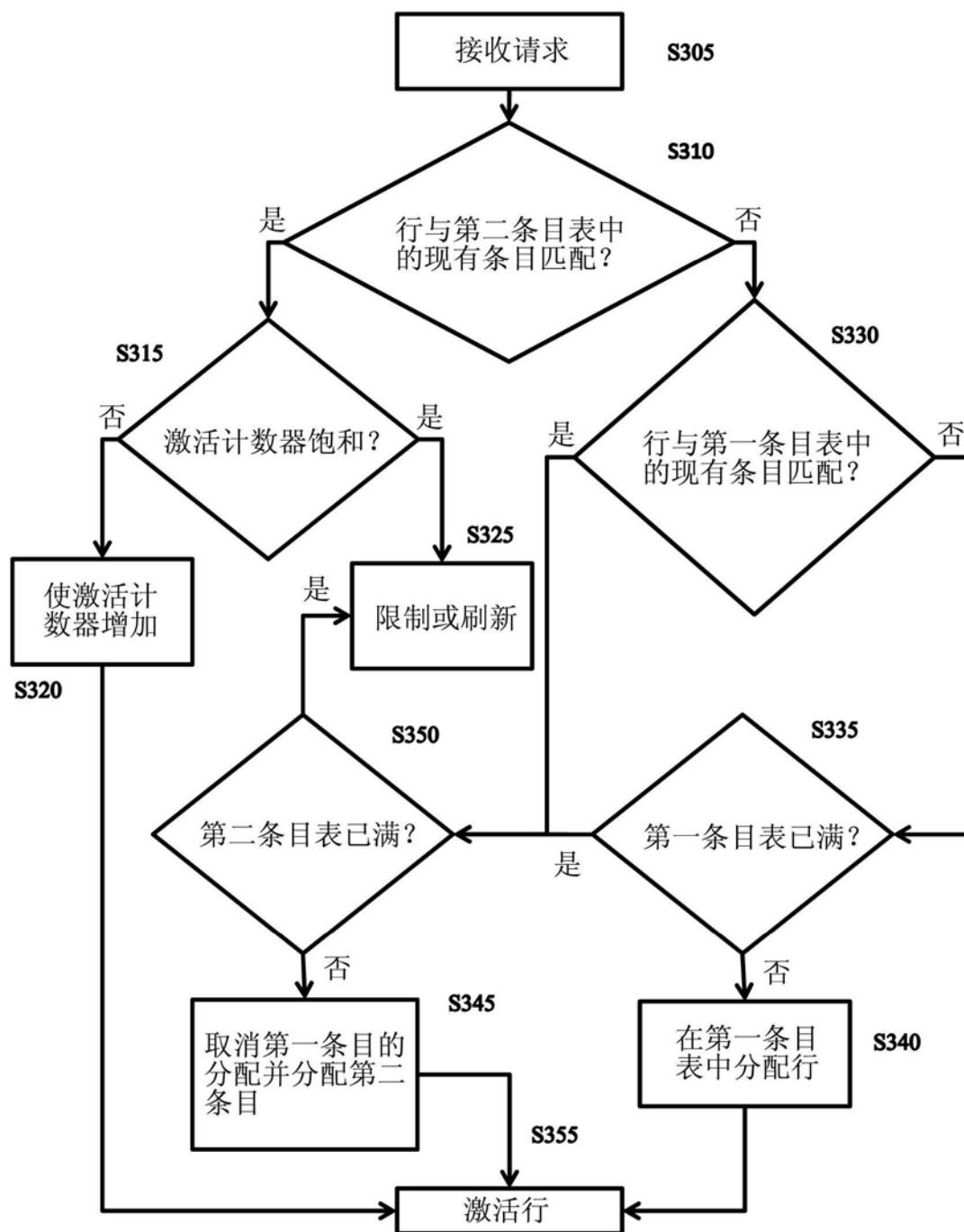


图3

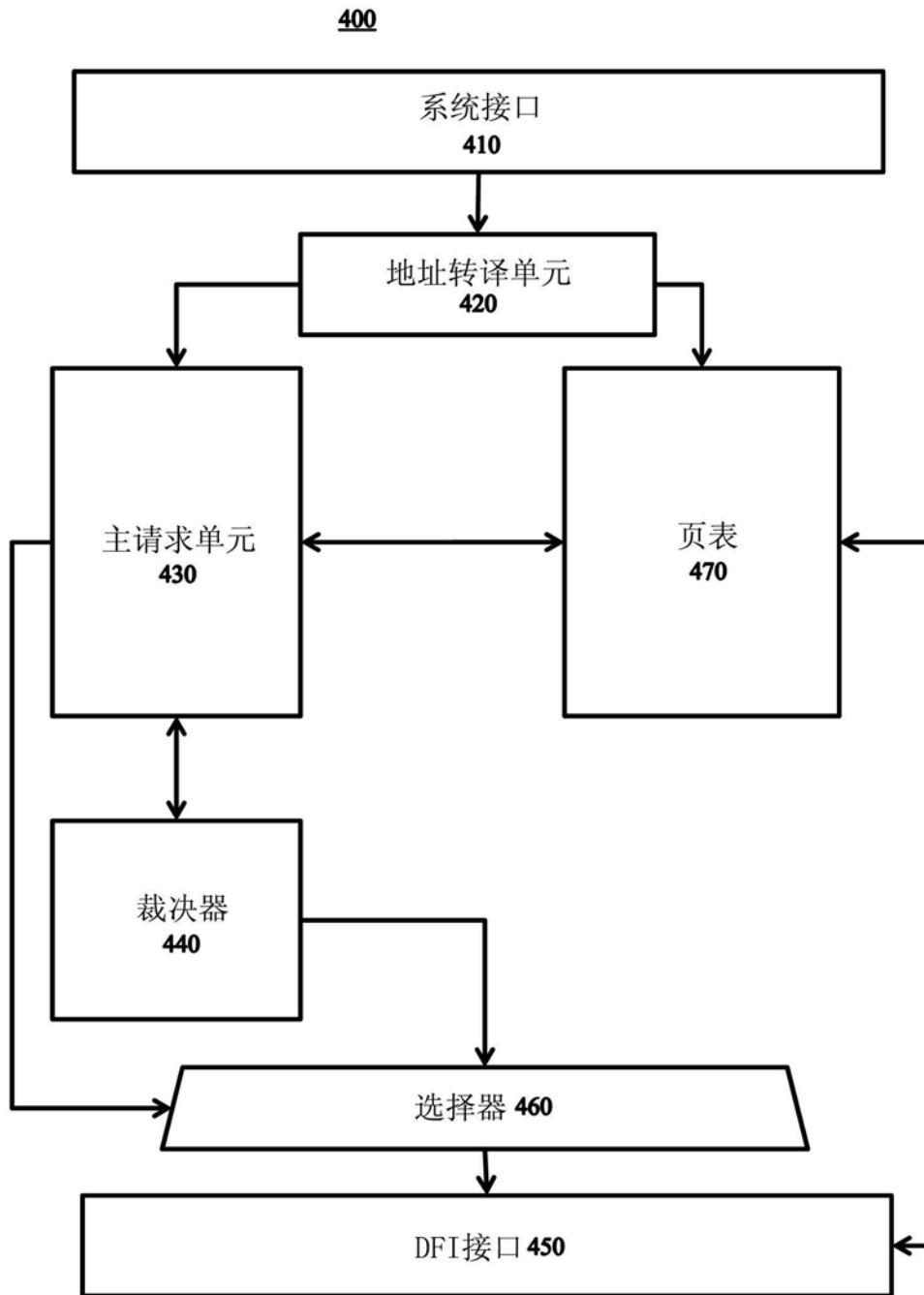


图4

DDR4器件参数	
器件密度	8GBx4
器件/组	16
X64通道容量	16GB
存储体/组	16
行/存储体	128K
页面尺寸/组	8KB
页面尺寸/器件	512B
数据速率	2400MT/s
总线频率	1200MHz

图5

方案			性能降级(越小越好)							
			行数/组: 128k 地址模式: 循环, 达到N行, 64B读/激活							
			进入TRR模式				限制激活			
名称	计数器的宽度(位)	每个存储体存储量(K位)	N>=6	N=4	N=2	N=1	N>=6	N=4	N=2	N=1
方案1: 更频繁刷新	0	0	16.6%	16.6%	16.6%	16.6%	NA			
方案2: 每行一个漏桶计数器	19	2432	0.0%	0.0%	0.0%	0.0%	0.0%	11.7%	55.9%	77.9%
	16	2048	0.0%	1.4%	1.4%	1.4%	0.0%	11.7%	55.9%	77.9%
	8	1024	0.0%	5.3%	5.4%	5.5%	0.0%	11.7%	55.9%	77.9%
	4	512	0.0%	16.9%	18.6%	19.6%	0.0%	11.7%	55.9%	77.9%
方案3: 每个存储体16个一位计数器和8个漏桶计数器	19	0.168	0.0%	0.0%	0.0%	0.0%	0.0%	11.7%	55.9%	77.9%
	16	0.144	0.0%	1.4%	1.4%	1.4%	0.0%	11.7%	55.9%	77.9%
	8	0.080	0.0%	5.3%	5.4%	5.5%	0.0%	11.7%	55.9%	77.9%
	4	0.048	0.0%	16.9%	18.6%	19.6%	0.0%	11.7%	55.9%	77.9%

图6