



(12) 发明专利申请

(10) 申请公布号 CN 102129412 A

(43) 申请公布日 2011.07.20

(21) 申请号 201110008985.9

(22) 申请日 2011.01.17

(30) 优先权数据

1050044-5 2010.01.18 SE

61/295794 2010.01.18 US

(71) 申请人 厄塞勒拉特公司

地址 瑞典斯德哥尔摩

(72) 发明人 V·苏科尼克 S·利夫内

B·维尔丁

(74) 专利代理机构 中国专利代理(香港)有限公司

72001

代理人 王岳 蒋骏

(51) Int. Cl.

G06F 13/18(2006.01)

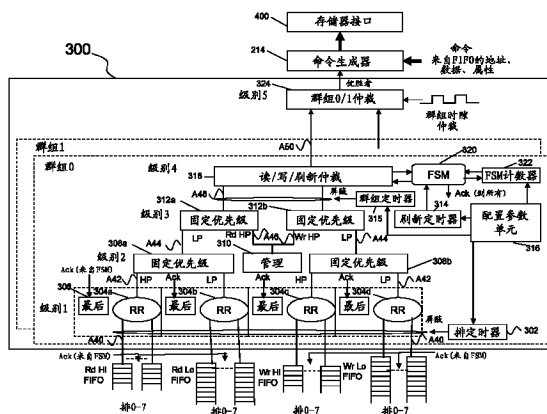
权利要求书 3 页 说明书 15 页 附图 6 页

(54) 发明名称

存取调度器

(57) 摘要

本发明涉及一种用于对一个或多个存储器器件(106)的存取进行调度的存取调度器(300),包括:第一层次级别(级别1),包括:一个或多个排定时器(302),被配置成当与存储器器件(106)的每排关联的FIFO队列被存取时设定该排的屏蔽位,所述FIFO队列存储与存取请求有关的请求信息;以及一个或多个第一级别仲裁器(304),被配置成选择来自多个FIFO队列中的一个FIFO队列的存取请求并且把关于所述一个FIFO队列的合格信息传播到下一层次级别;以及第四层次级别(级别4),包括:第四级别仲裁器(318),被配置成接收传播的合格信息并且基于传播的合格信息来选择请求。



1. 一种用于对一个或多个存储器器件(106)的存取进行调度的存取调度器(300),包括:

—第一层次级别(级别1),包括:

——一个或多个排定时器(302),被配置成当与存储器器件(106)的每排关联的FIFO队列被存取时设定该排的屏蔽位,所述FIFO队列存储与存取请求有关的请求信息;以及

——一个或多个第一级别仲裁器(304),被配置成选择来自多个FIFO队列中的一个FIFO队列的存取请求并且把关于所述一个FIFO队列的合格信息传播到下一层次级别;以及

—第四层次级别(级别4),包括:

——第四级别仲裁器(318),被配置成接收传播的合格信息并且基于传播的合格信息来选择请求。

2. 根据权利要求1的存取调度器(300),其中所述第四级别仲裁器(318)进一步被配置成基于由有限状态机(320)生成的刷新请求来选择请求。

3. 根据权利要求1或2的存取调度器(300),其中所述有限状态机(320)被配置成控制在切换到第二类型的许多存取请求例如写请求之前要执行的第一类型的存取请求例如读请求的数量。

4. 根据权利要求1-3中任一项的存取调度器(300),进一步包括:

—第五层次级别(级别5),包括:

——第五级别仲裁器(324),被配置成在对第一存储器器件(106)的存取请求和对第二存储器器件(106)的存取请求之间选择。

5. 根据权利要求1-4中任一项的存取调度器(300),进一步包括:

—第二层次级别(级别2),包括:一个或多个第二级别仲裁器(308),适于从第一级别仲裁器(302)接收传播的合格信息并且被配置成在低优先级请求和高优先级请求之间执行仲裁且把与选择的请求有关的合格信息传播到下一层次级别;以及

—第三层次级别(级别3),包括:一个或多个第三级别仲裁器(312),被配置成:从第二级别仲裁器(308)接收传播的合格信息;从管理请求单元(310)接收管理请求,以基于传播的合格信息和管理请求来选择请求;并且把与选择的存取请求有关的合格信息传播到下一层次级别。

6. 根据权利要求1-5中任一项的存取调度器(300),进一步包括:刷新定时器(314),被配置成基于存储在配置参数单元(316)中的配置参数来指令所述有限状态机(320)生成刷新请求。

7. 根据权利要求1-6中任一项的存取调度器(300),其中所述刷新请求是对存储器器件(106)的维护请求,诸如刷新存储器器件(106)的内容的请求。

8. 根据权利要求1-7中任一项的存取调度器(300),其中所述刷新请求被周期性地发送并且优先于其它请求。

9. 根据权利要求1-6中任一项的存取调度器(300),进一步包括:一个或多个群组定时器(315),被配置成根据预定时间周期来设定群组即存储器器件(106)的屏蔽位。

10. 根据权利要求9的存取调度器(300),其中所述时间周期被选择为以下中的任一个:(tFAW),其限制时间窗口内的行激活命令的数量;(tRRD),是到DRAM器件的两个连

续激活命令之间的最小时间；(tRead2write),是读命令和后续写命令之间的最小时间；(tWrite2read),是写命令和后续读命令之间的最小时间；或者(tCalibration),是两个校准命令之间的时间周期。

11. 根据权利要求 1-6 中任一项的存取调度器(300),其中所述配置参数涉及在要执行许多写请求之前要执行的读请求的数量或者在写之前要读的突发的数量。

12. 一种存储器控制器(104),包括:

—根据权利要求 1-11 中任一项的存取调度器(300),

—存取缓冲器(200),连接到所述存取调度器(300)且到一个或多个存储器器件(106)并且被配置成从连接到存储器控制器(104)的处理器(102)接收对所述一个或多个存储器器件(106)的一个或多个存取请求,以及

—重新排序器件(500),连接到所述存取缓冲器(200),被配置成接收来自一个或多个存储器器件(106)的响应数据并且对接收的响应数据进行分类,使得以与处理器(102)发送存取请求相同的顺序产生到处理器(102)的存取响应。

13. 一种借助于存取调度器(300)对一个或多个存储器器件(106)的存取进行调度的方法,包括以下步骤:

—在包括一个或多个第一级别仲裁器(304)的第一层次级别(级别 1)中:

—借助于一个或多个排定时器(302),当与存储器器件(106)的每排关联的 FIFO 队列被存取时设定该排的屏蔽位;

—一把与存取请求有关的请求信息存储在 FIFO 队列中;

—一把选择来自多个 FIFO 队列中的一个 FIFO 队列的存取请求;并且

—一把关于所述一个 FIFO 队列的合格信息传播到下一层次级别;以及

—在包括第四级别仲裁器(318)的第四层次级别(级别 4)中:

—接收传播的合格信息;并且

—基于传播的合格信息来选择请求。

14. 根据权利要求 13 的方法,进一步包括以下步骤:借助于第四级别仲裁器(318):

—基于由有限状态机(320)生成的刷新请求来选择请求。

15. 根据权利要求 13 或 14 的方法,进一步包括以下步骤:借助于有限状态机(320):

—控制在切换到第二类型的许多存取请求例如写请求之前要执行的第一类型的存取请求例如读请求的数量。

16. 根据权利要求 13-15 中任一项的方法,进一步包括以下步骤:借助于第五层次级别(级别 5)中的第五级别仲裁器(324):

—在对第一存储器器件(106)的存取请求和对第二存储器器件(106)的存取请求之间选择。

17. 根据权利要求 13-16 中任一项的方法,进一步包括以下步骤:

—在包括一个或多个第二级别仲裁器(308)的第二层次级别(级别 2)中:

—从第一级别仲裁器(302)接收传播的合格信息;

—在低优先级请求和高优先级请求之间执行仲裁;并且

—一把与选择的请求有关的合格信息传播到下一层次级别;以及

—在包括一个或多个第三级别仲裁器(312)的第三层次级别(级别 3)中:

- 从第二级别仲裁器(308)接收传播的合格信息;
- 从管理请求单元(310)接收管理请求;
- 基于传播的合格信息和管理请求来选择请求;并且
- 把与选择的存取请求有关的合格信息传播到下一层次级别。

18. 根据权利要求 13-17 中任一项的方法,进一步包括以下步骤:借助于刷新定时器(314):

—基于存储在配置参数单元(316)中的配置参数来指令所述有限状态机(320)生成刷新请求。

19. 根据权利要求 13-18 中任一项的方法,其中所述刷新请求是对存储器器件(106)的维护请求,诸如刷新存储器器件(106)的内容的请求。

20. 根据权利要求 13-19 中任一项的方法,进一步包括以下步骤:借助于群组定时器(315):

—根据预定时间周期来设定每个群组即每个存储器器件(106)的屏蔽位。

21. 根据权利要求 20 的方法,其中所述时间周期被选择为以下中的任一个:(tFAW),其限制时间窗口内的行激活命令的数量;(tRRD),是到 DRAM 器件的两个连续激活命令之间的最小时间;(tRead2write),是读命令和后续写命令之间的最小时间;(tWrite2read),是写命令和后续读命令之间的最小时间;或者(tCalibration),是两个校准命令之间的时间周期。

## 存取调度器

### 技术领域

[0001] 本发明涉及用于对用于分组缓冲的诸如一个或多个动态随机存取存储器 (DRAM) 之类的一个或多个存储器器件的存取进行调度的存取调度器。本发明也涉及包括这种存取调度器的存储器控制器并且涉及包括该存储器控制器的数据处理系统。

### 背景技术

[0002] 因为动态随机存取存储器 (DRAM) 的低成本和低功率, 所以往往使用动态随机存取存储器 (DRAM) 来实施分组缓冲器, 例如排队器件。然而, DRAM 由于其长的等待时间以及其对随机存取的约束而提出挑战。DRAM 被成排 (bank) 地组织并且 DRAM 的物理性质对排存取提出限制。例如, 存取参数行循环时间  $t_{RC}$  给出对 DRAM 排中的行的存取和对相同 DRAM 排中的另一行的连续存取之间的最小时间。另一个存取参数滚动时间帧  $t_{FAW}$  限制时间窗口内的行激活命令的数量, 在所述滚动时间帧  $t_{FAW}$  中可以同时进行对相同 DRAM 器件的最大四行激活。行是排的一部分。在可以执行对行内的地址的读或写之前, 必须激活该行。

[0003] DRAM 的存储器控制器接收针对 DRAM 的不同排的读和写请求。由于就每时间单位的存取而言 DRAM 带宽往往是瓶颈, 所以 DRAM 的存储器控制器可以重新布置读和写请求的顺序, 使得最大化存储器接口的利用。

[0004] 一种优化是以固定的顺序循环地存取这些排, 因而确保对任一 DRAM 排的两个连续存取之间的时间大于或等于行循环时间  $t_{RC}$ 。

[0005] 另一种优化是重新布置读请求和写请求, 使得多个读请求后面是多个写请求: 例如把序列  $S1 = (R1, W2, R3, W4)$  重新布置成  $S2 = (R1, R3, W2, W4)$ , 其中 R 代表读, W 代表写并且数字指示其中存储器控制器接收请求的顺序。对于在对 DRAM 的读和写存取之间的转向而言经常存在带宽惩罚, 因此  $S2$  比  $S1$  在更短的时间内完成。

[0006] Bains 的已公开美国申请 US 2004/0236921 A1 揭示了一种改进高速缓存数据总线上的带宽使得可以更高效地使用诸如 DRAM 之类的高速缓存存储器方法。在一个实施例中, 重新排序读或写存取以高效地利用数据总线上的带宽。

[0007] Van Hook 等人的美国专利 US 6, 564, 304 B1 揭示了一种用于对图形处理系统中的存储器进行存取的存储器处理系统, 其中存储器控制器仲裁来自多个存储器请求器的存储器存取请求。读被分组在一起并且写被分组在一起以避免模式切换。

[0008] 然而, 例如, 如果 DRAM 排中的地址在变成写之前被读, 则诸如读和写存取之类的存取的重新排序可能造成逻辑错误。例如, 在上面提及的序列  $S1 = (R1, W2, R3, W4)$  和  $S2 = (R1, R3, W2, W4)$  中,  $W2$  和  $R3$  可能存取相同的排地址。如果  $W2$  写数据结构 (例如链接表) 的元素并且  $R3$  存取该数据结构的相同元素, 则如果如以  $S2$  那样重新排序  $W2$  和  $R3$ , 将发生逻辑错误, 原因在于这将使程序在地址被写完之前读该地址。即, 重新排序将使解析该链接表的程序使用陈旧的指针, 造成程序故障。

[0009] Laskshmanamurthy 等人的已公开美国申请 US 2007/0156946 A1 揭示了一种利用排分类 (sort) 和调度的存储器控制器。存储器控制器包括 FIFO 缓冲器、仲裁器、排 FIFO 集

和排调度器。来自 FIFO 缓冲器的输出被馈送到把存储器请求分类到适当的排 FIFO 中的仲裁器中。仲裁器可以使用循环仲裁方案来分类和优先化输入请求流。排调度器接收来自排 FIFO 集的输出并且循环地处理这些请求。在每个循环中,排调度器可以选择对读 / 写效率进行优化的业务,例如排调度器可以对读和 / 或写进行分组以最小化读写周转。

[0010] 在 US 2007/0156946 A1 中,如上面所描述的逻辑错误的问题由“无序”机构解决,该“无序”机构确保从不违反支配对相同地址的读和写的业务排序规则,即地址在其被写完之前不能被读。

[0011] US 2007/0156946 A1 的存储器控制器的缺点 / 问题在于其未提供存储器带宽的加权公平共享。通过把读和写存储器请求存储在相同的排 FIFO 中,读和写之间的共享由请求到达过程确定而不由存储器控制器调整。

[0012] 进一步,US 2007/0156946 A1 的存储器控制器具有确保例如在写请求之前不发出读请求(如果这些请求是针对相同地址的话)的仲裁器。因而,尽管存在要在存储器控制器的内部存储器中读的数据,还发出对 DRAM 的读请求。这意味着 DRAM 未被最优地利用并且读等待时间未被最小化。

[0013] 而且,US 2007/0156946 A1 的存储器控制器没有装置来优先化需要低等待时间的请求,例如与控制信息有关的请求,使得它们在容忍较长等待时间的请求(例如与分组数据有关的请求)之前被服务。

## 发明内容

[0014] 本发明的目标是克服现有技术的问题和 / 或缺点。

[0015] 按照本发明,提供 DRAM 带宽的共享和优化。进一步,被配置成设定每一 DRAM 排或每一 DRAM 器件(群组)的位屏蔽(mask)的发明存取调度器使存取调度器与存储器接口同步地执行 DRAM 存取的开环控制,即当存取调度器发出命令序列时,保证它们以与它们由存取调度器发出相同的速率由存储器接口执行。

[0016] 用于对一个或多个存储器器件的存取进行调度的发明存取调度器的实施例包括:

—第一层次级别,包括:

——一个或多个排定时器,被配置成当与存储器器件的每排关联的 FIFO 队列被存取时设定该排的屏蔽位,所述 FIFO 队列存储与存取请求有关的请求信息;以及

——一个或多个第一级别仲裁器,被配置成选择来自多个 FIFO 队列中的一个 FIFO 队列的存取请求并且把关于所述一个 FIFO 队列的合格(eligibility)信息传播到下一层次级别;以及

—第四层次级别,包括:

——第四级别仲裁器,被配置成接收传播的合格信息并且基于传播的合格信息来选择请求。

[0017] 在实施例中,第四级别仲裁器进一步被配置成基于由有限状态机生成的刷新请求来选择请求。

[0018] 在实施例中,有限状态机被配置成控制在切换到第二类型的许多存取请求例如写请求之前要执行的第一类型的存取请求例如读请求的数量。

[0019] 实施例进一步包括：第五层次级别，包括：

—第五级别仲裁器，被配置成在对第一存储器器件的存取请求和对第二存储器器件的存取请求之间选择。

[0020] 实施例进一步包括：

—第二层次级别，包括：一个或多个第二级别仲裁器，被布置成从第一级别仲裁器接收传播的合格信息并且被配置成在低优先级请求和高优先级请求之间执行仲裁且把与选择的请求有关的合格信息传播到下一层次级别；以及

—第三层次级别，包括：一个或多个第三级别仲裁器，被配置成：从第二级别仲裁器接收传播的合格信息；从管理请求单元接收管理请求，以基于传播的合格信息和管理请求来选择请求；并且把与选择的存取请求有关的合格信息传播到下一层次级别。

[0021] 实施例进一步包括：刷新定时器，被配置成基于存储在配置参数单元中的配置参数来指令所述有限状态机生成刷新请求。

[0022] 在实施例中，刷新请求是对存储器器件的维护请求，诸如刷新存储器器件的内容的请求。

[0023] 在实施例中，刷新请求被周期性地发送并且优先于其它请求。

[0024] 在实施例中，配置参数涉及在要执行许多写请求之前要执行的读请求的数量或者在写之前要读的突发(burst)的数量。

[0025] 本发明也涉及一种包括发明存取调度器的存储器控制器。存储器控制器的实施例进一步包括：

—存取缓冲器，连接到所述存取调度器且到一个或多个存储器器件并且被配置成从连接到存储器控制器的处理器接收对所述一个或多个存储器器件的一个或多个存取请求，以及

—重新排序器件，连接到所述存取缓冲器，被配置成接收来自一个或多个存储器器件的响应数据，以对接收的响应数据进行分类，使得以与处理器发送存取请求相同的顺序产生到处理器的存取响应。

[0026] 本发明也涉及一种对一个或多个存储器器件的存取进行调度的方法，包括提供具有如下的存取调度器的步骤：

—第一层次级别，包括：

——一个或多个排定时器，被配置成当与存储器器件的每排关联的FIFO队列被存取时设定该排的屏蔽位，所述FIFO队列存储与存取请求有关的请求信息；以及

——一个或多个第一级别仲裁器，被配置成选择来自多个FIFO队列中的一个FIFO队列的存取请求并且把关于所述一个FIFO队列的合格信息传播到下一层次级别；以及

—第四层次级别，包括：

——第四级别仲裁器，被配置成接收传播的合格信息并且基于传播的合格信息、由有限状态机生成的可能刷新请求以及由群组定时器设定的可能屏蔽来选择请求。

[0027] 方法的实施例进一步包括以下步骤：借助于有限状态机，控制在切换到第二类型的许多存取请求例如写请求之前要执行的第一类型的存取请求例如读请求的数量。

[0028] 方法的实施例进一步包括以下步骤：提供包括第五级别仲裁器的第五层次级别，所述第五级别仲裁器被配置成在对第一存储器器件的存取请求和对第二存储器器件的存

取请求之间选择。

[0029] 方法的实施例进一步包括提供如下的步骤：

—第二层次级别,包括:一个或多个第二级别仲裁器,布置成从第一级别仲裁器接收传播的合格信息并且被配置成在低优先级请求和高优先级请求之间执行仲裁且把与选择的请求有关的合格信息传播到下一层次级别;以及

—第三层次级别,包括:一个或多个第三级别仲裁器,被配置成:从第二级别仲裁器接收传播的合格信息;从管理请求单元接收管理请求,以基于传播的合格信息和管理请求来选择请求;并且把与选择的存取请求有关的合格信息传播到下一层次级别。

[0030] 方法的实施例进一步包括提供刷新定时器的步骤,所述刷新定时器被配置成基于存储在配置参数单元中的配置参数来指令所述有限状态机生成刷新请求。

[0031] 被配置成调度对分组缓冲器(诸如 DRAM)的存取的发明存取调度器的实施例具有以下特征和优点:

—使用定时器控制的屏蔽来防止排存取模式(pattern)违反诸如排随机存取时间之类的 DRAM 约束。

[0032] —产生也防止对存取的 DRAM 约束造成存储器总线拖延(stall)的排存取模式。

[0033] —在 DRAM 群组之间交替以进一步获得存储器总线带宽,其中存储器器件共享横跨群组的地址总线但是具有每一群组的专用数据总线。

[0034] —产生不同长度的 DRAM 存取;即包括每一存取不同的数据突发量而不是迫使处理器产生每一请求每一排和行包括特定固定数量的突发的存取。

[0035] —给读和写存取灵活地分配带宽共享。

[0036] —支持例如分别针对控制和高优先级和数据的低优先级存取。

[0037] —管理存取的超高优先级。

[0038] 包括发明存取调度器的发明存储器控制器适合于用于缓冲器系统,即以诸如队列之类的结构暂时存储在输入上接收的数据以等待传输到输出上的系统;例如用于通信量(traffic)管理器。通信量管理器是一种例如通过组合用于队列管理、调度、通信量整形和阻塞控制但不限于此类使用的机构而给联网系统中的流提供带宽管理的器件。

[0039] 进一步,这样的发明存储器控制器产生:

存储器接口的高利用,即存储器请求序列被重新排序,使得对于任何序列的存储器请求的速率(rate)接近最优重新排序序列的速率;

到高带宽的可伸缩性,即发明存储器控制器容易适于与任意数量的 DRAM 器件一起使用;

有限的等待时间,即从发送要求来自存储器控制器的响应的请求(例如读请求)起的时间具有已知的上限;

存取类型之间的存储器带宽的受控共享,即 DRAM 带宽的已知(例如最小)共享可以被分配给存取类型,例如读存取或写存取;以及

数据相对于请求顺序的不重新排序,即如果处理器向存储器控制器产生后面是第二请求的第一请求,其中这两个请求触发响应(例如读请求),则存储器控制器在对第二请求的响应之前返回对第一请求的响应。



## 附图说明

[0040] 通过连同附图一起阅读时的本发明实施例的以下详细描述,将更容易理解本发明的目标、优点和效果以及特征,在附图中:

图 1 示意性地示出根据本发明的数据处理系统的实施例的概观;

图 2 示意性地示出根据本发明的数据处理系统的另一个实施例的概观;

图 3 示意性地示出根据本发明的存取调度器的实施例;

图 4 示意性地示出根据本发明的存取调度器的广义实施例;

图 5 示意性地示出存取缓冲器的实施例;

图 6 示意性地示出存取缓冲器的广义实施例;

图 7 示意性地示出根据本发明的存取调度器中的排定时器设定的屏蔽位的序列图;

图 8 示意性地示出根据本发明的存取调度器中的群组定时器设定的屏蔽位的序列图;

以及

图 9 示意性地示出根据本发明的存取调度器中的选择和命令生成的序列图。

## 具体实施方式

[0041] 虽然本发明覆盖各种修改以及可选方法、装置和系统,但是本发明的实施例在附图中被示出并且将在下文中被更详细地描述。然而,要理解,特定的描述和附图不旨在把发明限制为所揭示的特定形式。相反,所要求保护的发明的范围旨在包括落入如在所附权利要求到其等效物的整个范围中表示的发明的精神和范围内的其所有修改和可选构造。在附图中,相同的参考数字用于相同或类似的特征。

[0042] 图 1 示出根据本发明的数据处理系统 100 的实施例的概观。数据处理系统 100 包括连接到存储器控制器 104 的处理器 102。处理器 102 被配置成把存取请求发送到存储器控制器 104 并且接收来自存储器控制器 104 的存取响应,如箭头所示的。

[0043] 处理器 102 可以是执行计算机程序的中央处理器单元(CPU)、多处理器、网络处理器、可编程管线、IO 处理器或者诸如有限状态机之类的逻辑电路。

[0044] 存储器控制器 104 经由存储器接口 400 连接到一个或多个存储器器件 106 并且被配置成控制对一个或多个存储器器件 106 的存取。有时也被称为分组缓冲器的存储器器件 106 可以是具有随机存取约束的长等待时间存储器,诸如动态随机存取存储器(DRAM)。

[0045] 在实施例中,存储器控制器 104 包括存取缓冲器 200、存取调度器 300、存储器接口 400 和重新排序器件 500,参见图 1。如先前所提及的,存储器接口 400 被配置成把存储器控制器 104 连接到一个或多个存储器器件 106 (例如 DRAM) 并且实施针对使用的器件类型的通信协议(例如 DRAM 协议)。

[0046] 然而,如图 2 示意性地所示的,应当理解,在实施例中存储器接口 400 可以被布置在一个或多个存储器控制器 104 的外部。在这样的实施例中,存储器接口 400 可以借助于交换机 450 而连接到一个或多个存储器控制器 104,参见图 2。该交换机可以例如是交叉开关(crossbar)、网状网络、环或总线。

[0047] 优选地,存储器接口 400 被微编码并且被配置成响应于接收的命令而执行微编码的序列。该命令可以例如是从存取缓冲器 200 接收的读请求或写请求。

[0048] 当用于本描述中时,措词“微代码”/“微编码”指代实施功能基元以供其它单元使

用的硬件级指令。微代码驻留在特殊的存储器(未示出)中并且把输入转换成内部电路或输入/输出(I/O)管脚上的电路级控制信号序列。

[0049] 存取缓冲器 200 被配置成接收来自连接到存储器控制器 104 的一个或多个处理器 102 的一个或多个存取请求。在实施例中,存取缓冲器 200 被配置成处理对一个或多个存储器器件 106 例如 DRAM 的存取。

[0050] 存取请求可以具有不同类型,例如具有高或低优先级的读或写请求以及管理请求。然而,应当理解,其它实施例可以包括其它请求类型,诸如 DRAM 刷新请求。

[0051] 在实施例中,管理请求具有最高优先级,并且与写和读控制信息数据有关的请求一般具有比与写和读数据例如有效载荷数据有关的请求更高的优先级。

[0052] 管理请求可以例如是对存储器接口中的微代码存储器或配置参数的读或写请求、或者对存储器接口的状态位的读请求。

[0053] 数据请求可以例如是对在 DRAM 中缓冲的分组数据例如有效载荷数据的读或写请求。

[0054] 控制信息数据请求可以例如是对与对数据进行存储的数据结构有关的信息(诸如链接表指针或分组长度)的读或写请求。

[0055] 在实施例中,存取缓冲器 200 基于先进先出(FIFO),即存取缓冲器 200 包括许多 FIFO 缓冲器,所述 FIFO 缓冲器在本说明书文本中也被称为 FIFO 队列。

[0056] 如果例如存取缓冲器 200 被配置成处理两个存储器器件 106 (例如 DRAM),每个具有 8 排,则与 FIFO 队列关联的每一请求类型存在 16 个 FIFO 队列。管理请求经常不与 FIFO 队列关联,因为它们具有最高的优先级并因此被优先化而不用缓冲。具有高优先级的读请求、具有低优先级的读请求、具有高优先级的写请求和具有低优先级的写请求每个与每排存储器器件的 FIFO 队列关联。

[0057] 根据从处理器接收的请求类型且根据其旨在用于哪个存储器器件和该存储器器件的哪排,该请求被存储在请求类型和所讨论的存储器器件排对应的 FIFO 队列中。

[0058] 存取缓冲器 200 包括或连接到存取调度器 300。存取调度器被配置成从存取缓冲器的 FIFO 队列接收或检索请求信息和/或从存取缓冲器接收或检索管理请求信息。请求信息可以例如是存取缓冲器中的 FIFO 队列包括存取请求的指示,其也可以包括 FIFO 队列为空的信息以及关于请求长度的信息。优选地,由于管理请求具有最高优先级,所以存取调度器可以直接接收或检索管理请求信息而不用管理请求信息在存取缓冲器的任何缓冲。

[0059] 存取调度器进一步被配置成借助于包括一个或多个调度算法的一个或多个调度器或仲裁器来调度请求。也被称为选择的请求的所调度请求然后作为请求命令发送到所讨论的存储器器件,优选地到存储器器件的特定排。请求命令优选地在存取缓冲器中被生成并且从存取缓冲器直接或经由交换机发送到存储器接口。

[0060] 存储器控制器 104 进一步包括重新排序器件 500,该重新排序器件 500 被连接成并且被配置成从存取缓冲器 200 和/或从存储器器件 106 接收响应数据。在实施例中,重新排序器件 500 被连接成并且被配置成经由存储器接口 400 以及也可能经由交换机 450 从存储器器件 106 接收响应数据。重新排序器件 500 被配置成对接收的响应数据进行分类,使得与处理器 102 向存储器控制器 104 发出对应的存取请求相同的顺序产生到处理器 102 的存取响应。

[0061] 图 3 和 4 示出发明存取调度器 300 的实施例。该发明存取调度器被配置为层次调度器,其中也被称为合格信息的状态信息从叶子朝树根传播,如下面将描述的。在根中做出调度决策以便选择服务哪个请求,然后调度决策从根传播到叶子,如下面将描述的。在叶子处,请求信息被检索并发送到基于所检索的请求信息来生成 DRAM 命令的命令生成器。

[0062] 在实施例中,发明存取调度器 300 包括五个层次级别,如图 3 示意性地所示的。然而,如图 4 示意性地所示的,存取调度器 300 的实施例可以包括树层次级别。在图 4 中,在存取调度器 300 中包括三个级别:级别 1,级别 4 和级别 5。

[0063] 现在将参考图 3 来描述合格信息从叶子到根的传播。

[0064] FIFO 队列,诸如“Rd Hi FIFO”,“Rd Lo FIFO”,“Wr Hi FIFO”或者“Wr Lo FIFO”中的一个,在其不是空的且其排定时器屏蔽允许调度的情况下有资格由在第一级别(级别 1)处的调度器 304 调度。

[0065] 进一步,如果由第一级别调度器 304 服务的任一 FIFO 是合格的,则在级别 1 处的第一级别调度器 304 有资格由在级别 2 处的第二级别调度器 308 调度。例如,为了第一级别调度器 304a 有资格由第二级别调度器 308a 调度,读高优先级 FIFO 队列“Rd Hi FIFO”之一必须是合格的。

[0066] 如果由在级别 3 处的第三级别调度器 312 服务的在级别 2 处的任一第二级别调度器 308 是合格的,则在级别 2 处的第二级别调度器 308 有资格由在级别 3 处的第三级别调度器 312 调度。

[0067] 进一步,如果由在级别 4 处的第四级别调度器 318 服务的在级别 3 处的任一第三级别调度器 312 是合格的,则在级别 3 处的第三级别调度器 312 有资格由在级别 4 处的第四级别调度器 318 调度。为了使第三级别调度器 312 是合格的,群组定时器 315 必须允许调度。在级别 5 处,在级别 4 处的群组以循环的方式交替。

[0068] 现在将参考图 3 和 9 来描述调度决策从根到叶子的传播。

[0069] 对于该群组,即在级别 5 处选择的 DRAM 器件,如果也被称为刷新定时器 314 的用于刷新的计数器达到了在配置参数单元 316 中可配置的阈值,则该群组的有限状态机(FSM)320 选择刷新操作。刷新操作可以被延迟,直到达到其中 FSM 否则将从读切换到写或空闲或者从写切换到读或空闲的状态为止。如果未选择刷新,并且读和 / 或写是合格的,则读 / 写仲裁器 318 使用例如亏损加权循环(DWRR)调度算法来选择读或写。否则调度器 318 可以是空闲的。如果选择读或写,则被选择的在级别 3 处的第三级别调度器 312 在合格的第二级别调度器 308 之间选择具有最高优先级的第二级别调度器 308。

[0070] 然后,被选择的第二级别调度器 308 在合格的第一级别调度器 304 之间选择具有最高优先级的第一级别调度器 304。第一级别调度器 304 以循环方式选择下一合格的 FIFO 队列。

[0071] 应当理解,合格信息从叶子到根的传播以及调度决策从根到叶子的传播在具有除 5 之外的另一个层次级别数量的存取调度器中以相同的方式运行。

[0072] 根据实施例,第一级别(级别 1)包括一个或多个排定时器 302,排定时器的数量对应于排的数量。排定时器 302 被配置成当在存取缓冲器中包括的且与该排关联的 FIFO 队列被存取时设定每排存储器器件 106 的屏蔽位,如图 7 示意性地所示的。在图 3 中,屏蔽位的设定由环绕连接 FIFO 与第一级别调度器 304 的线 A40 的环所示。通过当相关 FIFO 队列被

存取时设定该排的屏蔽位,该排不能被选择用于另一个存取直到该屏蔽位被清除为止。进一步,当 DRAM 特定的随机存取时间  $t_{RC}$  过期时,借助于排定时器 302 来清除该屏蔽位。从而,可以防止存取错误,诸如在对排的先前存取之后过早地存取相同排。

[0073] 进一步,调度器 300 包括一个或多个第一级别仲裁器 304,例如与存取缓冲器中的 FIFO 队列关联的每一存取请求类型一个第一级别仲裁器 304。FIFO 队列把合格信息传播到第一级别仲裁器 304,如图 9 中的箭头 A40 所示的。第一级别仲裁器 304 被配置成借助于调度算法诸如如图 3 中由 RR 所示的循环调度算法,以公平的方式且基于所接收的请求信息来选择与排 0-7 关联的 FIFO 队列之一中包括的存取请求之一。

[0074] 因而,向排的工作守恒循环服务提供每一存取请求类型一个循环;即每一读高优先级请求、读低优先级请求、写高优先级请求和写低优先级请求一个循环。

[0075] 一旦选择了存取请求,第一级别仲裁器 304 将把关于存储与选择的存取请求有关的请求信息的 FIFO 的合格信息传播到在下一级别(级别 2)中包括的第二级别仲裁器 308,如图 9 中的箭头 A42 所指示的。

[0076] 如图 3 示意性地所示的,用于调度读高优先级请求的第一级别仲裁器 304a 被配置成基于与排 0-7 关联的 8 个 FIFO 队列“Read Hi FIFO”之一中包括的请求信息来选择读高优先级请求之一。以相同的方式,第一级别仲裁器 304b、304c、304d 被配置成分别从 FIFO 队列“Read Lo FIFO”、“Write Hi FIFO”和“Write Lo FIFO”选择读低优先级请求、写高优先级请求和写低优先级请求。

[0077] 在最后存储器 306 中,存储来自调度器 304 的指针。该指针被配置成告诉哪个排 0-7 是要服务的最后一排。在调度决策之后最后指针被增加 1 模与调度器关联的排的数量,以指示调度器 304 接着要服务哪个排。

[0078] 根据实施例,第二级别(级别 2)包括一个或多个第二级别仲裁器 308。优选地,存取调度器 300 包括每一请求类型(不管高低优先级)一个第二级别仲裁器 308,即每一写请求一个和每一读请求一个。因而,第一第二级别仲裁器 308a 被配置成在读请求之间仲裁而第二第二级别仲裁器 308b 被配置成在写请求之间仲裁。

[0079] 第二级别仲裁器 308 被配置成借助于固定优先级调度算法在低优先级请求和高优先级请求之间执行仲裁,即第二级别仲裁器 308 被配置成选择具有最高优先级的请求并且把合格信息传播到在下一级别(级别 3)中包括的第三级别仲裁器 312,如图 9 中的箭头 A44 所指示的。

[0080] 因而,只要读高优先级请求存在,在选择读低优先级请求之前,第一第二级别仲裁器 308a 将选择读高优先级请求。

[0081] 按照相同的推理,第二第二级别仲裁器 308b 被配置成在选择写低优先级请求之前选择写高优先级请求。

[0082] 在第二级别(级别 2)中,存取调度器 300 也包括管理请求单元 310,其被配置成把具有高优先级的管理读请求从存取缓冲器传播到第三级别调度器 312a 并且把合格信息传播到第三级别调度器 312b,如图 9 中的箭头 A46 所指示的。

[0083] 根据实施例,在第三级别(级别 3)中,存取调度器 300 包括一个或多个第三级别仲裁器 312。第三级别仲裁器 312 被配置成在正常存取请求即读请求和写请求以及管理存取请求之间调度。如先前所提及的,管理存取请求是高优先级存取请求,其可以源自除处理器

之外的另一个部件；例如源自连接到存取调度器 / 存取缓冲器的控制平面 CPU。

[0084] 如图 3 示意性地所示的，第一第三级别仲裁器 312a 被配置成在读请求和读管理请求之间选择，而第二第三级别仲裁器 312b 被配置成在写请求和写管理请求之间选择。由于管理请求比读请求和写请求具有更高的优先级，所以只要管理请求可用于选择，第三级别仲裁器 312 将选择管理请求。

[0085] 如图 9 中的箭头 A48 所指示的，合格信息被传播到在下一级别(级别 4)中包括的第四级别仲裁器 318。

[0086] 在实施例中，存取调度器 300 也包括刷新定时器 314，其被配置成基于存储在配置参数单元 316 中的配置参数来指令有限状态机(FSM) 320 以生成刷新请求。该刷新请求可以例如是对 DRAM 的维护请求，即请求维护 DRAM 以例如通过内部电机构来刷新存储器的内容的请求。若不刷新，存储器内容变得走样(corrupted)。刷新请求可以被周期性地发送并且它们可以优先于其它存取请求和管理请求。

[0087] 在实施例中，存取调度器包括一个或多个群组定时器 315。群组定时器 315 被配置成根据例如 tFAW、tRRD、tRead2write、tWrite2read 和 tCalibration 来设定群组即 DRAM 器件的屏蔽位，如图 8 示意性地所示的。tFAW 限制时间窗口内的行激活命令的数量，并且 tRRD 是到 DRAM 器件的两个连续激活命令之间的最小时间。tRead2write 是读命令和后续写命令之间的最小时间，而 tWrite2read 是写命令和后续读命令之间的最小时间。tCalibration 是两个校准命令之间的时间周期。可以例如执行校准命令以补偿操作条件的变化，诸如温度变化。

[0088] 在第四级别中，存取调度器可以包括配置参数单元 316。配置参数单元 316 被配置成把诸如在要执行许多写请求之前要执行的读请求的数量或者在写之前要读的突发的数量之类的配置参数传递到在第四级别(级别 4)中包括的有限状态机(FSM) 320。借助于配置参数，FSM 320 被配置成控制何时进行读、写、刷新或空闲。

[0089] FSM 被配置成把也被称为 Ack 信号的确认信号传输到一个或多个 FIFO 队列和传输到与一个或多个第一级别仲裁器 304 关联的一个或多个最后寄存器 306，如图 3、4 和 9 示意性地所示的。Ack 信号触发从选择的 FIFO 检索请求。另外，其触发在与选择的 FIFO 关联的 RR 调度器 304 的最后寄存器 306 中更新 RR 指针。

[0090] 根据实施例，存取调度器在第四级别(级别 4)中包括第四级别仲裁器 318、有限状态机(FSM) 320、FSM 定时器计数器 322。

[0091] 第四级别仲裁器 318 被配置成在不考虑优先级的情况下在请求之间即在写请求、读请求、管理请求和刷新请求之间调度，即在第四级别处请求是具有低优先级还是高优先级无关紧要。第四级别仲裁器 318 也可以被配置成调度在要执行许多写请求之前要执行的许多读请求以便最小化存储器存取的延迟，原因在于不同存取请求(例如读、写、管理、刷新和维护)之间的切换花费时间并且也可能要求插入空闲循环。

[0092] 在实施例中，刷新定时器 314 被配置成对刷新请求之间的循环数量进行计数并且一个或多个 FSM 计算器 322 被配置成对写和读请求的数量以及对在诸如读、写和刷新之类的命令类型之间切换时需要插入的循环的数量进行计数。

[0093] 读和写带宽是可配置的并且以公平、加权、工作守恒的方式被动态地共享。在配置参数单元 316 中包括的两个可配置参数是“MaxRead”“MaxWrite”。参数“MaxWrite”指示

在转向服务 DRAM 读之前被写(如果合格的话)的 DRAM 突发的数量。参数“MaxRead”指示在转向服务 DRAM 写之前被读(如果合格的话)的 DRAM 突发的数量。在实施例中,“MaxRead”和“MaxWrite”是亏损加权循环(DWRR)调度器的权重。

[0094] 进一步,第四级别仲裁器 318 被配置成把合格信息传播到在存取调度器的下一级别(级别 5)中包括的第五级别仲裁器 324,如图 9 中的箭头 A50 所指示的。

[0095] 根据实施例,存取调度器在第五级别(级别 5)中包括第五级别仲裁器 324,其被配置成在到也被称为第一群组 0 的第一存储器器件 106 的存取请求和到也被称为第二群组 1 的第二存储器器件 106 的存取请求之间选择。在实施例中,这些群组共享地址总线但是具有单独的数据总线。在实施例中,第五级别仲裁器 324 在选择对第一存储器器件和对第二存储器器件的存取请求之间交替。

[0096] 为了最大化存储器接口的利用,DRAM 器件因而可以被组织成群组,例如组织成群组 0 和群组 1。通过在群组之间交替,如由 tFAW 参数表示的对时间窗口内的行激活命令数量的 DRAM 约束不会减损存储器接口带宽。在群组之间交替也防止由于 tRRD 即两个连续行激活命令之间的最小时间和 tCCD 即两个列命令之间的最小时间引起的带宽减损。

[0097] 命令生成器 214 被布置成与存取调度器 300 通信并且被配置成基于关于来自存取调度器的优胜者的请求信息(如图 9 中的箭头 Rq Info 所指示)且基于与请求关联的且从 FIFO 队列检索的地址、数据和属性(如图 9 中的箭头“地址、数据和属性”所指示)来生成命令。该命令然后经由存储器接口 400 发送到 DRAM,如图 9 中的箭头 Cmd 所指示的。

[0098] 图 4 示意性地示出发明存取调度器 300 的更广义实施例。

[0099] 如所示的,存取调度器 300 的实施例在第一级别(级别 1)中包括一个或多个排定时器 302,排定时器的数量对应于排的数量。排定时器 302 被配置成当在存取缓冲器中包括的且与该排关联的 FIFO 队列被存取时设定每排存储器器件 106 的屏蔽位,如先前所描述的。

[0100] 进一步,调度器 300 包括一个或多个第一级别仲裁器 304,例如与存取缓冲器中的 FIFO 队列关联的每一存取请求类型一个第一级别仲裁器 304。在所示的实施例中,存取请求类型的数量是二:读请求和写请求。

[0101] 一旦选择了存取请求,第一级别仲裁器 304 将把关于存储与选择的存取请求有关的请求信息的 FIFO 的合格信息传播到在下一级别(级别 4)中包括的第四级别仲裁器 318,如线 A44 所指示的。

[0102] 在实施例中,存取调度器 300 也包括刷新定时器 314、群组定时器 315、配置参数单元 316、有限状态机(FSM) 320 和 FSM 计数器 322,如先前所描述的。

[0103] 第四级别仲裁器 318 被配置成在请求之间即在写请求、读请求、管理请求和刷新请求之间调度。第四级别仲裁器 318 也可以被配置成调度在要执行许多写请求之前要执行的许多读请求以便最小化存储器存取的延迟,原因在于不同存取请求(例如读、写、管理、刷新和维护)之间的切换花费时间并且也可能要求插入空闲循环。

[0104] 进一步,第四级别仲裁器 318 被配置成把合格信息传播到在存取调度器的下一级别(级别 5)中包括的第五级别仲裁器 324,如线 A50 所指示的。

[0105] 第五级别仲裁器 324 被配置成在到也被称为第一群组 0 的第一存储器器件 106 的存取请求和到也被称为第二群组 1 的第二存储器器件 106 的存取请求之间选择,如先前所

描述的。关于优胜者的请求信息被传输到命令生成器 214, 该命令生成器 214 基于请求信息且基于与请求关联的且在 FIFO 队列中存储的地址、数据和属性来生成命令。该命令然后经由存储器接口 400 发送到 DRAM 器件。

[0106] 图 5 示意性地示出存取缓冲器 200 的实施例。存取缓冲器 200 优选地基于先进先出 (FIFO), 即存取缓冲器 200 包括许多 FIFO 缓冲器, 所述 FIFO 缓冲器在本描述文本中也被称为 FIFO 队列。

[0107] 图 5 所示的存取缓冲器 200 的实施例被配置成处理两个存储器器件 106 (例如 DRAM), 每个具有 8 排, 意味着每一请求类型存在 16 个 FIFO 队列。由于与 FIFO 队列关联的请求类型的数量为 4 (管理请求不与 FIFO 队列关联), 所以存取缓冲器 200 的所示实施例包括总共 64 个 FIFO 队列。为了清楚起见, 在图 5 中未示出所有 64 个 FIFO 队列。

[0108] 然而, 应当理解, FIFO 队列的数量可以根据要处理的存储器器件的数量、在每个存储器器件中包括的排的数量和 / 或与一个或多个 FIFO 队列关联的请求类型的数量而不同。

[0109] 在实施例中, 存取缓冲器 200 包括: 许多写 / 读低优先级 / 高优先级 FIFO 队列 204: “Wr Lo Prio FIFO”, “Wr Hi Prio FIFO”, “Rd Hi Prio FIFO” 和 “Rd Lo Prio FIFO”; 许多写 / 读低优先级 / 高优先级存储器 206: “Wr LoPrio Storage”, “Rd LoPrio storage”, “Wr HiPrio Storage” 和 “Rd HiPrio Storage”; 搜索单元 208, 仲裁器 210, 复用器 212, 存取调度器 300, 以及命令生成器 214。命令生成器 214 被配置成以所需的格式构造存储器控制器命令并且经由存储器接口 400 把所生成的命令发送到存储器器件 106。

[0110] 在实施例中, 可搜索地址存储器 206 是内容可寻址相关存储器 (CAM)。

[0111] 进一步, 应当理解, 在其它实施例中, 存取调度器 300 可以被布置在存取缓冲器 200 的外部并且与存取缓冲器 200 通信。

[0112] 如图 5 的左手侧的向右箭头 A1-A5 示意性地所示的, 不同类型的请求, 例如读 / 写、高优先级 / 低优先级、和管理, 可以从处理器发送到存取缓冲器 200。然而, 应当理解, 其它实施例可以包括其它请求类型, 诸如 DRAM 刷新请求。

[0113] 在图 5 中, 示出以下不同请求类型: 管理请求 “Mgmt”; 具有低优先级的写请求 “WriteLoPrio”, 例如写数据请求; 具有高优先级的写请求 “WriteHiPrio”, 例如写控制信息数据请求; 具有高优先级的读请求 “ReadHiPrio”, 例如读控制信息数据请求; 以及具有低优先级的读请求 “ReadLoPrio”, 例如读数据请求。

[0114] 在实施例中, 例如参见图 5, 写请求包括要写到存储器器件的数据、与要写数据的存储器器件的地址对应的地址、以及定义该请求相对于其它请求的顺序的属性; 而读请求包括与要读数据的存储器器件的地址对应的地址、以及定义该请求相对于其它请求的顺序的属性。

[0115] 在其它实施例中, 例如参见图 6, 写请求不包括属性。在这样的实施例中, 写请求包括要写到存储器器件的数据、以及与要写数据的存储器器件的地址对应的地址; 而读请求包括与要读数据的存储器器件的地址对应的地址、以及定义该请求相对于其它请求的顺序的属性。

[0116] 在实施例中, 管理请求具有最高优先级, 并且与写和读控制信息数据有关的请求一般具有比与写和读数据例如有效载荷数据有关的请求更高的优先级。

[0117] 管理请求可以例如是对存储器接口中的微代码存储器或配置参数的读或写请求。

[0118] 数据请求可以例如是对在 DRAM 中缓冲的分组数据例如有效载荷数据的读或写请求。

[0119] 控制信息数据请求可以例如是对与对数据进行存储的数据结构有关的信息(诸如链接表指针或分组长度)的读或写请求。

[0120] 如图 5 示意性地所示的,当存取缓冲器 200 如箭头 A1 所指示的那样从处理器 102 接收把例如低优先级数据写到存储器器件 106 的请求“WriteLoPrio”时,低优先级数据的地址将如箭头 A1a 所指示的那样被写到在存取缓冲器 200 中包括的 FIFO 队列“Wr LoPrio FIFO”之一。进一步,写低优先级请求的数据和属性(Attr)将被存储在存取缓冲器 200 中包括的可搜索地址存储器“Wr LoPrio Storage”206 中,如箭头 A1b 所指示的。

[0121] 进一步,如图 5 示意性地所示的,当存取缓冲器 200 如箭头 A2 所指示的那样从处理器接收把例如控制信息数据写到存储器器件 106 的请求“WriteHiPrio”时,请求“WriteHiPrio”的地址将如箭头 A2a 所指示的那样被写到在存取缓冲器 200 中包括的 FIFO 队列“Wr HiPrio FIFO”之一。进一步,请求“WriteHiPrio”的数据和属性将被存储在存取缓冲器 200 中包括的可搜索地址存储器“Wr HiPrio Storage”206 中,如箭头 A2b 所指示的。

[0122] 而且,如图 5 示意性地所示的,当存取缓冲器 200 如箭头 A3 所指示的那样从处理器接收例如从存储器器件 106 读控制信息数据的请求“ReadHiPrio”时,请求“ReadHiPrio”的地址将如箭头 A3a 所指示的那样被写到在存取缓冲器 200 中包括的 FIFO 队列“Rd HiPrio FIFO”之一。进一步,请求“ReadHiPrio”的属性将被存储在存取缓冲器 200 中包括的可搜索地址存储器“Rd HiPrio Storage”206 中,如箭头 A3b 所指示的。

[0123] 如图 5 示意性地所示的,当存取缓冲器 200 如箭头 A4 所指示的那样从处理器 102 接收例如从存储器器件 106 读数据信息的请求“ReadLoPrio”时,存取缓冲器 200 被配置成借助于搜索单元 208 来(如虚线箭头所示的那样)搜索 FIFO 队列“Wr LoPrio FIFO”以确定请求“ReadLoPrio”的地址是否包括在 FIFO 队列“Wr LoPrio FIFO”之一中。

[0124] 如果要读的数据的地址是在 FIFO 队列“Wr LoPrio FIFO”之一中,则搜索单元 208 检测到“命中(hit)”。搜索单元 208 从 FIFO 队列“Wr LoPrio FIFO”去除该地址并且要读的数据和属性从数据存储器“Wr LoPrio Storage”进行读。在从“Wr LoPrio Storage”读之前,“Hit”请求可以如箭头 A4c 所指示的那样被写到 FIFO 队列“Hit Synch FIFO”,并且如箭头 A6 所指示的那样借助于仲裁器 210 被调度,以便如箭头 A7 所指示的那样以有序的且经调度的方式从存储器“Wr LoPrio Storage”读数据和属性。当从存储器“Wr LoPrio Storage”读了数据和属性时,数据和属性分别如箭头 A8b 和 A8c 所指示的那样被发送到重新排序器件 500。此后,重新排序器件 500 被配置成以正确的顺序把存取响应传输到处理器 102,该响应是基于所读的数据和属性,如图 5 中的重新排序器件 500 中的符号“属性+数据(命中)”所指示的。

[0125] 仲裁器 210 被配置成借助于严格的优先级调度算法来执行调度。在实施例中,由存取调度器 300 选择的存取请求应当优先于“Hit”请求,原因在于存储器接口的带宽应当被优化。在这种情况下,“Hit”请求可以在 Hit Synch FIFO 中等待可用的时隙(例如存取调度器读命令)。

[0126] 如果要读的地址不在 FIFO 队列之一中,则搜索单元 208 检测到“未中(miss)”。要



读的数据的地址然后如箭头 A4d 所指示的那样被写到在存取缓冲器 200 中包括的 FIFO 队列“Rd LoPrio FIFO”之一。进一步,该请求的属性(Attr)将被存储在存取缓冲器 200 中包括的存储器“Rd LoPrio Storage”中,如箭头 A4b 所指示的。

[0127] 如图 5 所示的,存取调度器 300 被配置成基于存取缓冲器 200 的 FIFO 队列 204 即“Wr LoPrio FIFO”、“Wr HiPrio FIFO”、“Rd HiPrio FIFO”和“Rd LoPrio FIFO”中的信息且基于如箭头 A5a 所指示的那样由存取调度器 300 接收的管理请求的可能地址来选择存取请求之一。存取调度器 300 被配置成借助于一个或多个仲裁器,基于调度算法例如如图 5 中用“RR”指示的循环调度算法、或先来先服务、或者另一种合适的调度算法来选择存取请求之一。存取调度器可以进一步包括调度器和有限状态机(FSM) 216,其被配置成执行写/读/刷新请求调度并且确保不违反 DRAM 协议。

[0128] 然而,如箭头 A5b 所指示的,管理请求的数据和属性可以直接发送到被配置生成命令的命令生成器 214,所述命令借助于存储器接口 400 发送到存储器器件 106。

[0129] 进一步,如箭头 A13 所指示的,存取调度器 300 被配置成把选择的存取请求的地址传输到命令生成器 214。当存取请求被存取调度器 300 选择时,存取缓冲器 200 被配置成从 FIFO 队列清除该存取请求的地址并且从存储器 206 之一读所选择请求的数据和/或属性。

[0130] 如果例如选择的存取请求是“WriteLoPrio”,即写低优先级请求,则存取调度器 300 将如箭头 A12 所指示的那样把在“Wr LoPrio FIFO”中包括的所选择存取请求的地址传达到“Wr LoPrio Storage”并且如箭头 A13 所指示的那样传达到命令生成器 214。进一步,如图 3 所示,地址可以通过仲裁器 210,以便在从“Wr LoPrio Storage”读和去除数据和属性之前,基于搜索单元 208 所检测的命中或者基于从存取调度器 300 选择的写低优先级请求来优先化“Wr LoPrio Storage”中的读。当从“Wr LoPrio Storage”读了数据和属性时,数据和属性被传输到命令生成器 214,如箭头 A8a 和 A14 所指示的。进一步,当从“Wr LoPrio Storage”读了数据和属性时,数据和属性被传输到重新排序器件 500,如箭头 A8b 和 A8c 所指示的。

[0131] 在实施例中,数据和属性经由复用器 212 被传输到命令生成器 214 以便在如箭头 A8a、A9a、A10 和 A11 所指示的那样从不同数据存储器传输的不同数据和属性之间进行选择。

[0132] 当从“Wr LoPrio Storage”读的数据和属性在命令生成器 214 中被接收时,其将与从“Wr LoPrio FIFO”读的地址组合成被传输到存储器器件的写低优先级命令。

[0133] 进一步,如果例如所选择的存取请求是“WriteHiPrio”,即写高优先级请求,则存取调度器 300 将如箭头 A12 所指示的那样把在“Wr HiPrio FIFO”中包括的写高优先级请求的地址传达到“Wr HiPrio Storage”并且如箭头 A13 所指示的那样传达到命令生成器 214。当从“Wr HiPrio Storage”读和去除了数据和属性时,数据和属性被传输到命令生成器 214,如箭头 A10 和 A14 所指示的。

[0134] 可能地,数据和属性经由复用器 212 被传输到命令生成器 214 以便在如箭头 A8a、A9a、A10 和 A11 所指示的那样从不同数据存储器传输的不同数据和属性之间进行选择。

[0135] 当来自“Wr HiPrio Storage”的数据和属性在命令生成器 214 中被接收时,其将与从“Wr HiPrio FIFO”读的地址组合成被传输到存储器器件的写高优先级命令。

[0136] 而且,如果例如所选择的存取请求是“ReadHiPrio”,即读高优先级请求,则存取调

度器 300 将如箭头 A12 所指示的那样把与所选择的存取请求对应的“Rd HiPrio FIFO”中存储的读高优先级请求的地址传达到“Rd HiPrio Storage”并且如箭头 A13 所指示的那样传达到命令生成器 214。当从“Rd HiPrio Storage”读和去除了属性时,属性被传输到命令生成器 214,如箭头 A11 和 A14 所指示的。可能地,属性经由复用器 212 被传输到命令生成器 214 以便在如箭头 A8a、A9a、A10 和 A11 所指示的那样从不同数据存储器传输的不同数据和属性之间进行选择。当从“Rd HiPrio Storage”读的属性在命令生成器 214 中被接收时,其将与从“Rd HiPrio FIFO”读的地址组合成被传输到存储器器件的读高优先级命令。

[0137] 如果例如所选择的存取请求是“ReadLoPrio”,即读低优先级请求,则存取调度器 300 将如箭头 A12 所指示的那样把与所选择的存取请求对应的“Rd LoPrio FIFO”的地址传达到“Rd LoPrio Storage”并且如箭头 A13 所指示的那样传达到命令生成器 214。当从“Rd LoPrio Storage”读和去除了属性时,属性被传输到命令生成器 214,如箭头 A9a 和 A14 所指示的。进一步,属性也被传输到重新排序器件 500,如箭头 A9 所指示的。

[0138] 可能地,属性经由复用器 212 被传输到命令生成器 214 以便在如箭头 A8a、A9a、A10 和 A11 所指示的那样从不同数据存储器传输的不同数据和属性之间进行选择。

[0139] 当从“Rd LoPrio Storage”读的属性在命令生成器 214 中被接收时,其将与从“Rd LoPrio FIFO”读的地址组合成被传输到存储器器件的读低优先级命令。

[0140] 当命令从命令生成器 214 发送到存储器器件时,关于发送的请求类型例如写 / 读低 / 高优先级请求“Type”和关于请求的序列标识“SerID”的信息被发送到重新排序器件 500,如箭头 A15 所指示的。

[0141] 在实施例中, SerID 是由处理器设定的属性。其用来识别回到处理器的响应。另外,其被重新排序器件 500 用来恢复可能被存取调度器打断的原始顺序。例如,处理器可以针对每个请求使 SerID 递增一直达到诸如 65 535 之类的最大值为止,此后 SerID 被复位到 0 并且继续按一递增。

[0142] 重新排序器件 500 被配置成使用所接收的属性来为与该属性关联的存取请求的顺序分配重新排序器件中的地址。因而,当在重新排序器件 500 中接收存取响应时,该属性可以用来知道何时把存取响应发送到处理器。因而借助于请求的属性,知道该请求相对于从处理器发送的其它请求的顺序。如果例如该请求的顺序在请求序列中是第二,则来自存储器器件的对这个请求的响应应当在发送了对第一请求的响应之后且在发送对第三请求的响应之前发送到处理器。

[0143] 图 6 示出被配置成处理分别如箭头 A1' 和 A3' 示意性地所示的所接收的写请求和读请求的存取缓冲器 200 的广义实施例。存取缓冲器 200 包括许多 FIFO 队列例如写、读和命中 FIFO 队列 204、搜索单元 208,存取调度器 300 以及命令生成器 214。进一步,存取缓冲器连接到被配置成如先前所描述的那样运行的重新排序器件 500。

[0144] 当如箭头 A1' 所示的那样接收包括地址和数据的写请求时,写请求的地址和数据被存储在写 FIFO 队列“Wr FIFO”之一中。

[0145] 当如箭头 A3' 所示的那样接收包括地址和属性的读请求时,搜索单元 208 被配置成在“Wr FIFO”中搜索与读请求的地址匹配的地址。

[0146] 如果找到这样的地址,则搜索单元 208 检测到命中并且可能经由命中 FIFO 队列“Hit FIFO”把读请求的属性和地址传输到重新排序器件 500。

[0147] 如果未找到这样的地址,则搜索单元 208 检测到未中并且把地址和属性存储在读 FIFO 队列“RD FIFO”之一中。

[0148] 如先前所描述的,存取调度器 300 被配置成选择请求之一。

[0149] 如果所选择的请求是写请求,则地址和数据从写 FIFO 进行检索并且发送到命令生成器 214 以生成写命令。

[0150] 如果所选择的请求是读请求,则属性被发送到重新排序器件 500 并且地址被发送到命令生成器 214 以生成读命令。

[0151] 命令生成器 214 被配置成把生成的命令发送到存储器器件并且把请求类型例如写或读“Type”以及 SerID 发送到重新排序器件 500。

[0152] 进一步,存取缓冲器的实施例可以被配置成具有拥有背压(backpressure)阈值的一个或多个 FIFO 队列。如果 FIFO 队列的填充级别超过其背压阈值,则存取缓冲器被配置成把这个背压传达到处理器。从而,处理器被配置成阻止对存取缓冲器的相同类型(例如读或写,或者具有优先级的读或写)的进一步存取请求,直到存取缓冲器向处理器传达 FIFO 队列的填充级别已返回到在阈值之下的级别。背压级别应当被设定为使最大不可阻止请求流适应 FIFO 之一。如果例如 FIFO 可以包括 16 个请求并且不可阻止量是 5 个请求,则背压阈值应当被设定为 11 个请求。

[0153] 而且,存取缓冲器的实施例可以被配置成具有拥有应急(panic)阈值的一个或多个 FIFO 队列。如果写 FIFO 队列或读 FIFO 队列的最大填充级别增长到大于应急阈值,则用于读和写的权重在存取调度器中被自动地改变,使得写或读得到更高的相对带宽共享。当填充级别返回到在阈值之下的级别时,离开应急模式并且把权重复位到正常值。

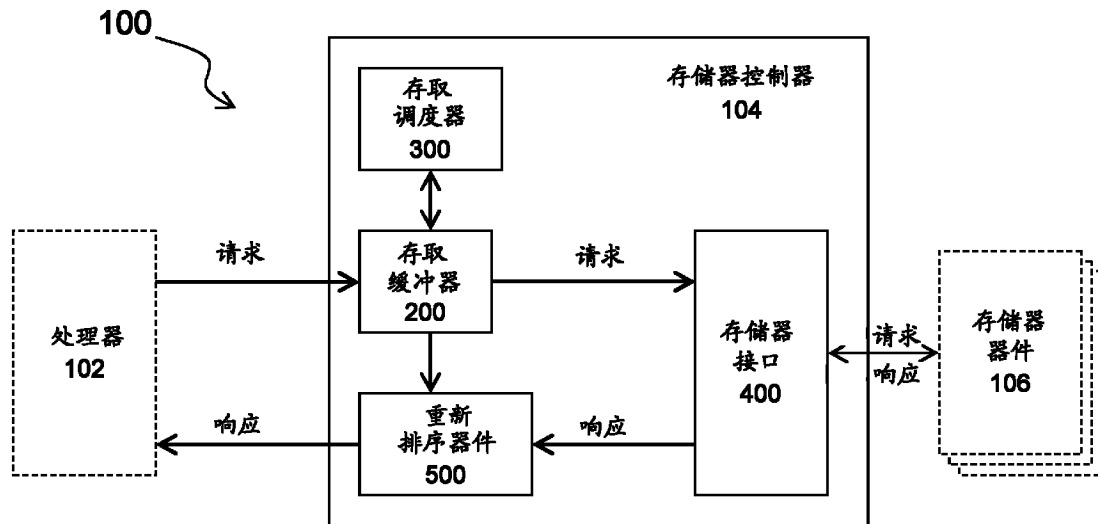


图 1

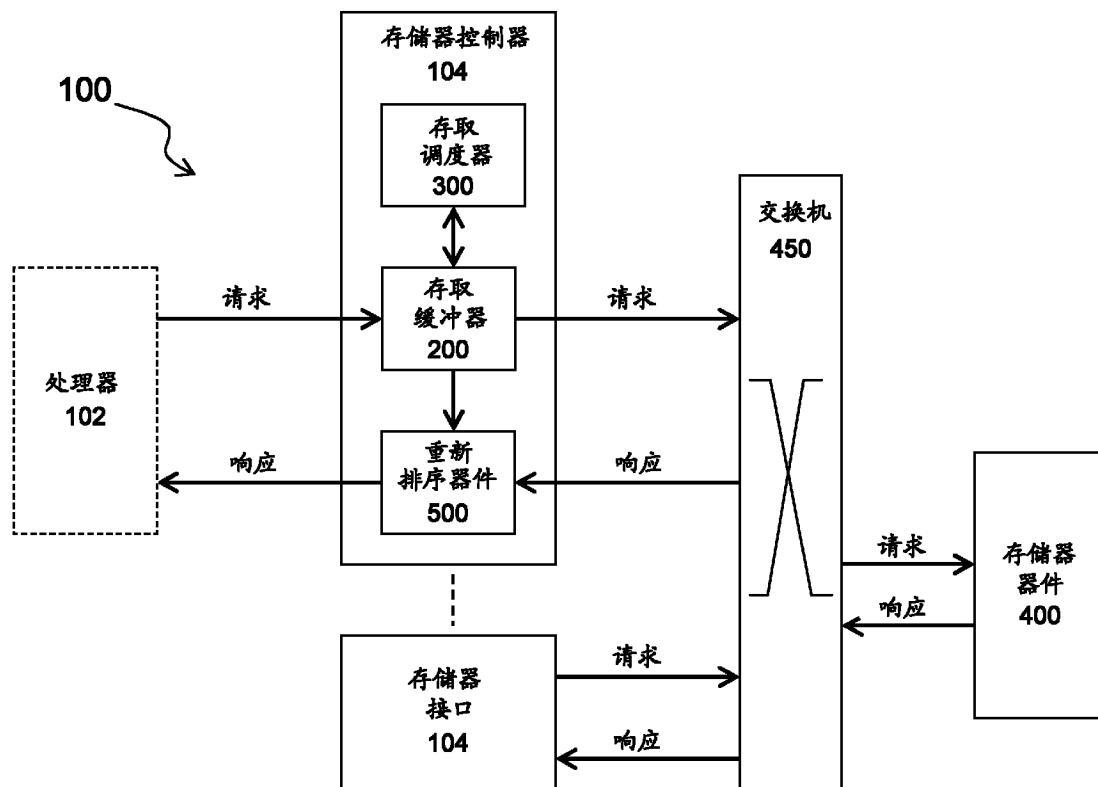


图 2

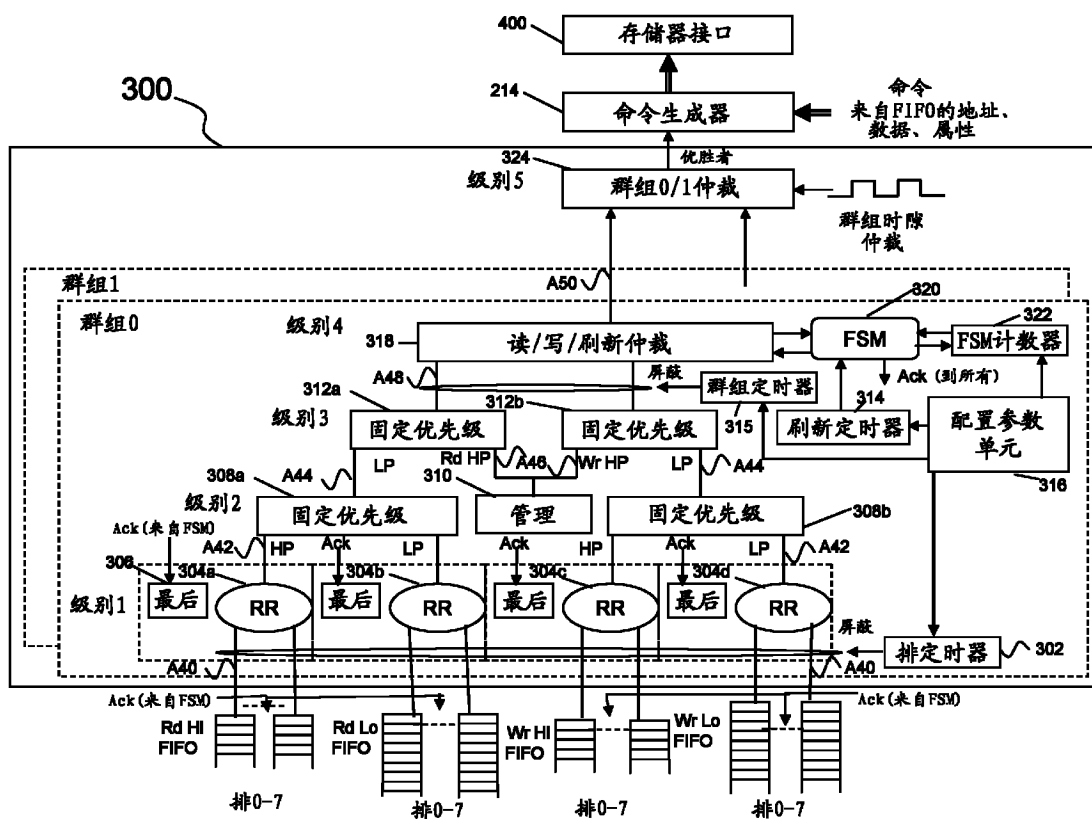


图 3

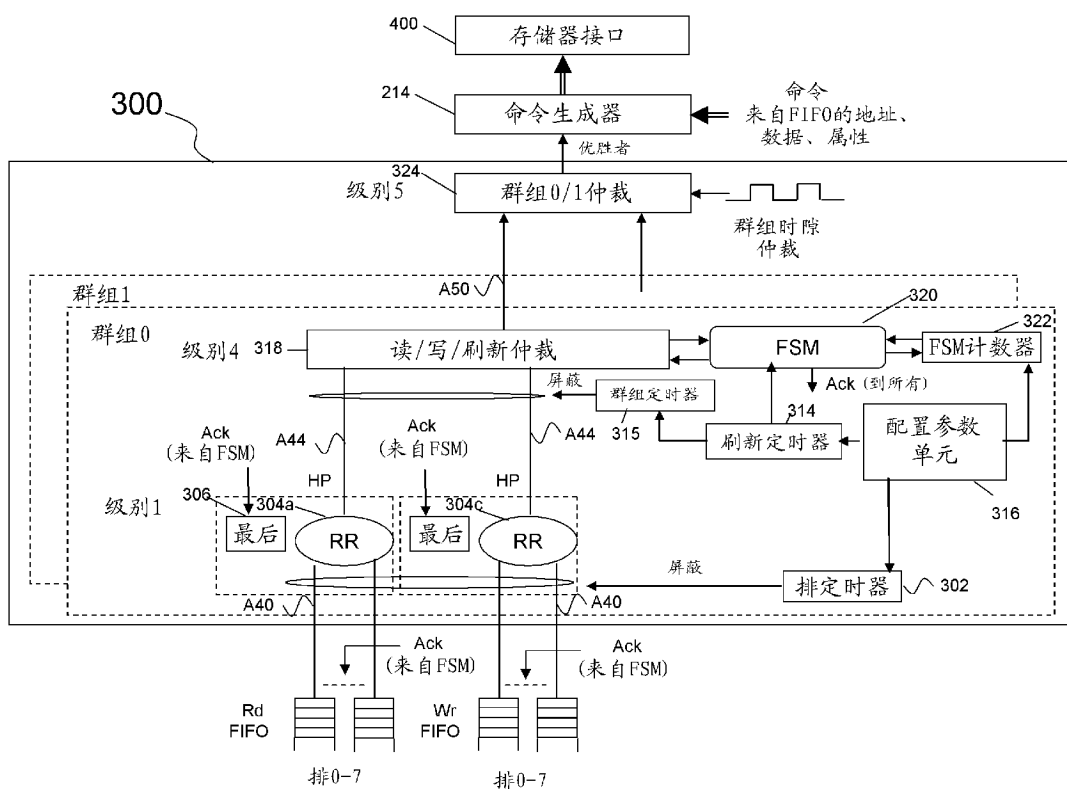


图 4

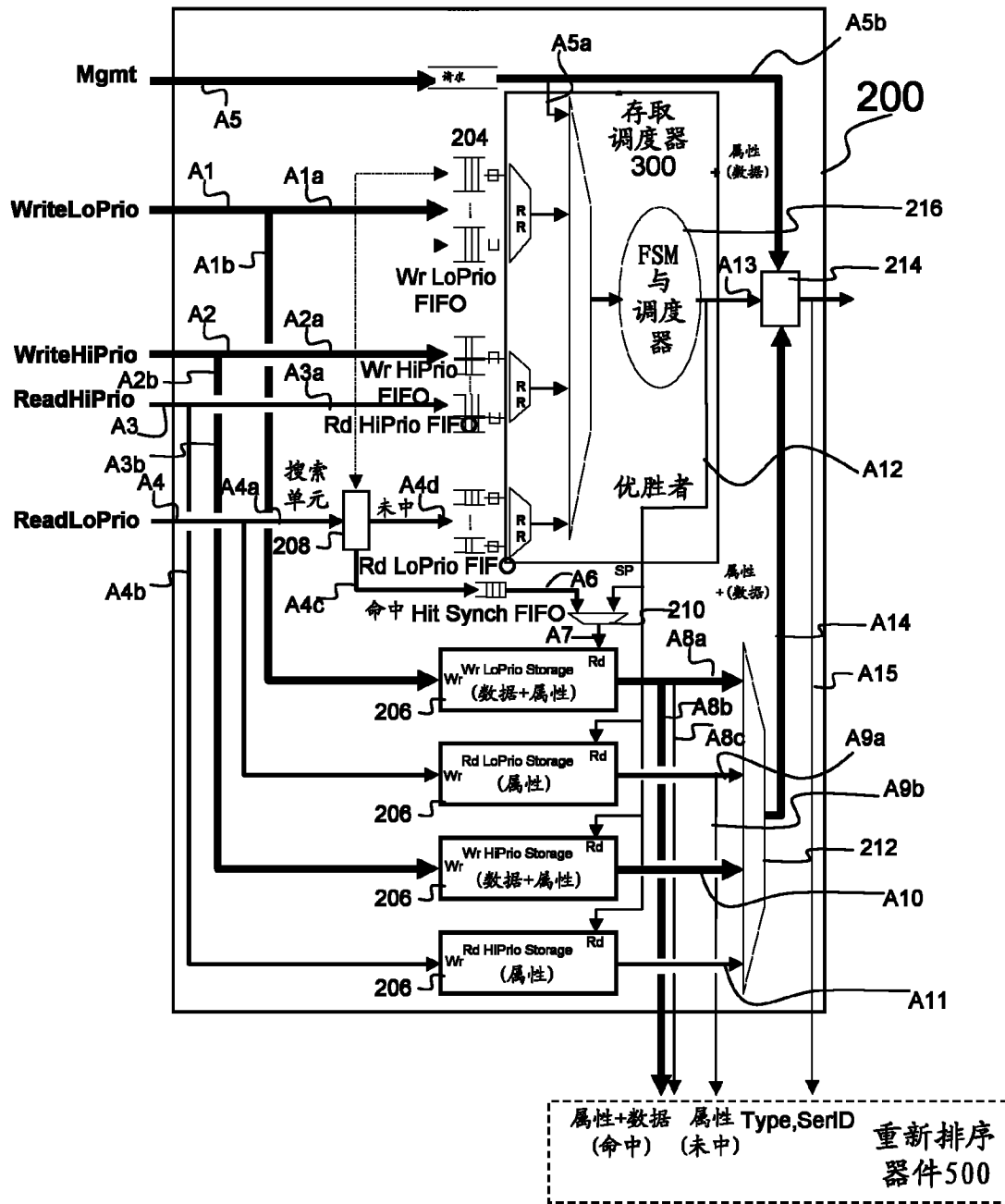


图 5

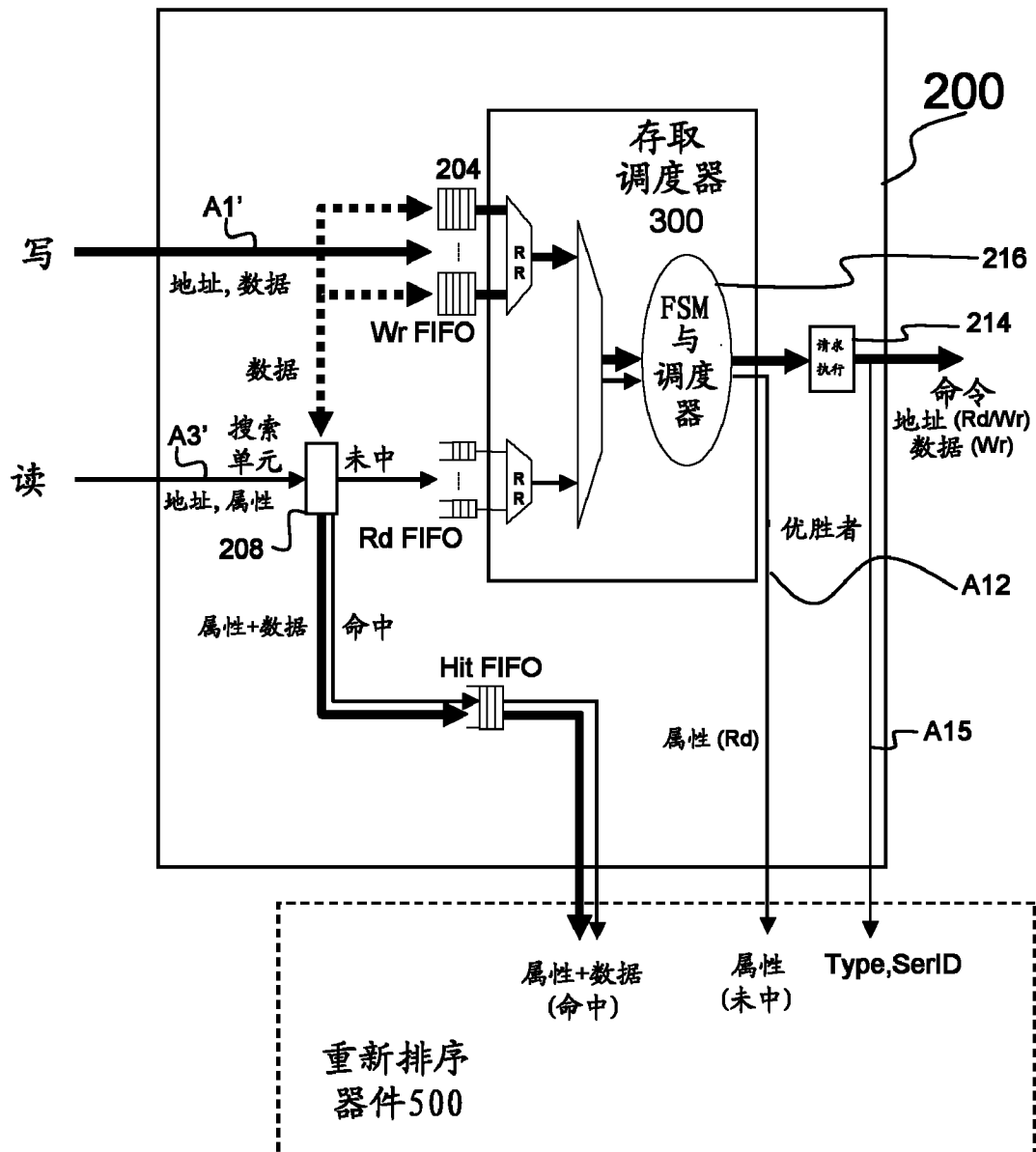


图 6



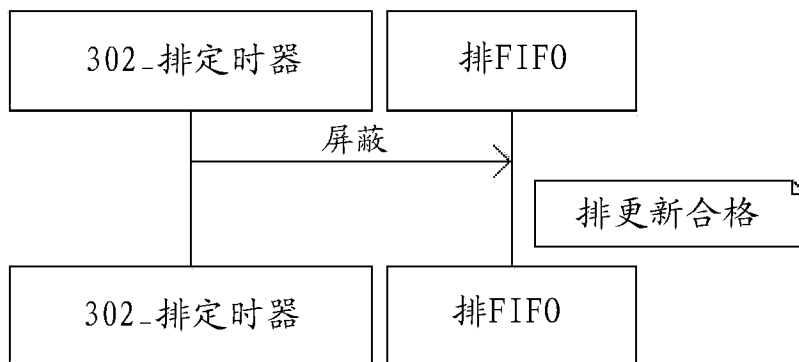


图 7

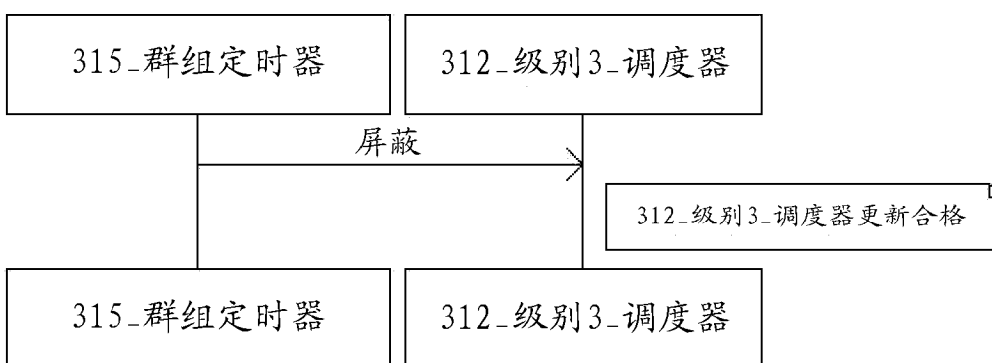


图 8

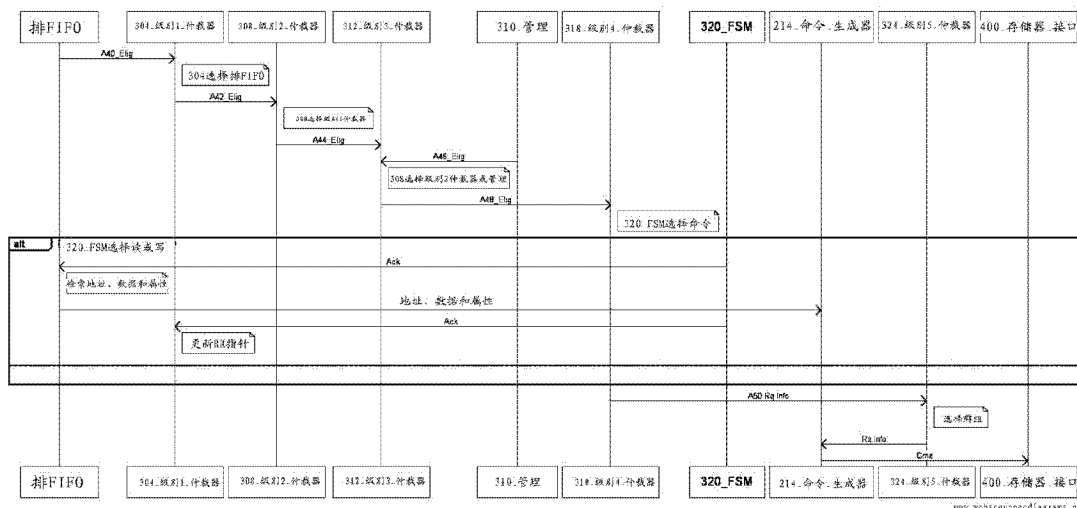


图 9