



(12) 发明专利申请

(10) 申请公布号 CN 101765887 A

(43) 申请公布日 2010. 06. 30

(21) 申请号 200880100528. 9

(51) Int. Cl.

(22) 申请日 2008. 07. 25

G11C 11/406 (2006. 01)

(30) 优先权数据

11/828, 569 2007. 07. 26 US

(85) PCT申请进入国家阶段日

2010. 01. 26

(86) PCT申请的申请数据

PCT/US2008/071153 2008. 07. 25

(87) PCT申请的公布数据

W02009/015324 EN 2009. 01. 29

(71) 申请人 高通股份有限公司

地址 美国加利福尼亚州

(72) 发明人 杰拉尔德·保罗·米夏拉克

巴里·乔·沃尔福德

(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 刘国伟

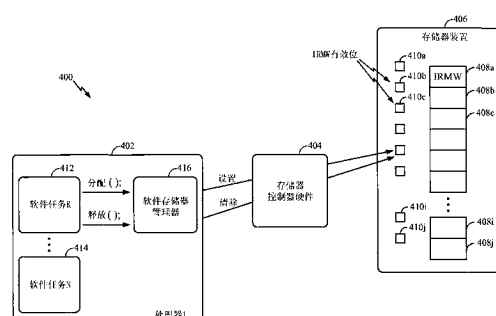
权利要求书 2 页 说明书 6 页 附图 6 页

(54) 发明名称

经由使用有效数据指示符减少动态 RAM 功率消耗的系统和方法

(57) 摘要

DRAM 或 SDRAM 组件维持指示符, 所述指示符指示 DRAM 阵列的可独立刷新的存储器单位 (例如, 行) 是否含有有效数据。当刷新操作针对相关联的存储器时, 如果所述存储器不含有有效数据, 那么抑制所述刷新操作。可通过抑制针对无效数据的刷新操作来实现显著的功率节省。



1. 一种刷新动态存储器的方法,其包括:
将指示符与每一可独立刷新的存储器单位相关联;
在将数据写入到可独立刷新的存储器单位后,设置所述相关联的指示符以反映有效数据;以及
仅刷新其相关联的指示符反映其中存储有有效数据的所述可独立刷新的存储器单位。
2. 根据权利要求1所述的方法,其中所述可独立刷新的存储器单位是行。
3. 根据权利要求1所述的方法,其中所述可独立刷新的存储器单位包括跨越两个或两个以上存储器排的行。
4. 根据权利要求1所述的方法,其中所述指示符是有效位。
5. 根据权利要求4所述的方法,其中所述有效位存储于DRAM阵列中。
6. 根据权利要求4所述的方法,其中所述有效位存储于静态存储器中。
7. 根据权利要求4所述的方法,其中所述有效位存储于寄存器中。
8. 根据权利要求1所述的方法,其中设置所述相关联的指示符以反映有效数据包括:
在将数据写入到所述相关联的可独立刷新的存储器单位后,自动设置所述指示符。
9. 根据权利要求1所述的方法,其中设置所述相关联的指示符以反映有效数据包括:
在来自存储器控制器的命令时设置所述指示符。
10. 根据权利要求1所述的方法,其中在复位时清除经设置以反映有效数据的指示符。
11. 根据权利要求1所述的方法,其中在来自存储器控制器的命令时清除经设置以反映有效数据的指示符。
12. 根据权利要求1所述的方法,其中仅刷新其相关联的指示符反映其中存储有有效数据的所述可独立刷新的存储器单位包括:仅自刷新其相关联的指示符反映其中存储有有效数据的所述可独立刷新的存储器单位。
13. 根据权利要求1所述的方法,其中仅刷新其相关联的指示符反映其中存储有有效数据的所述可独立刷新的存储器单位包括:仅自动刷新其相关联的指示符反映其中存储有有效数据的所述可独立刷新的存储器单位。
14. 根据权利要求1所述的方法,其中仅刷新其相关联的指示符反映其中存储有有效数据的所述可独立刷新的存储器单位包括:连续刷新非邻接的可独立刷新的存储器单位。
15. 根据权利要求1所述的方法,其中仅刷新其相关联的指示符反映其中存储有有效数据的所述可独立刷新的存储器单位包括:
接收刷新命令;
检查与当前刷新地址相关联的所述指示符;以及
如果所述指示符反映有效数据,那么刷新所述所寻址的可独立刷新的存储器单位。
16. 根据权利要求14所述的方法,其进一步包括:如果所述指示符反映无效数据,那么抑制刷新循环。
17. 根据权利要求14所述的方法,其进一步包括:如果所述指示符反映有效数据,那么将所述刷新地址递增到具有反映有效数据的指示符的下一可独立刷新的存储器单位。
18. 根据权利要求16所述的方法,其进一步包括:与跳过的刷新地址的数目成比例地降低刷新频率。
19. 一种DRAM组件,其包括:

DRAM 阵列,其操作以存储数据,所述 DRAM 阵列被组织为多个可独立刷新的存储器单位;

多个指示符,每一指示符与可独立刷新的存储器单位相关联,且指示是否有有效数据存储于所述可独立刷新的存储器单位中;以及

控制器,其接收控制信号且操作以检查所述指示符并仅刷新存储有效数据的所述可独立刷新的存储器单位。

20. 根据权利要求 19 所述的 DRAM 组件,其进一步包括刷新计数器,所述刷新计数器操作以产生所述 DRAM 阵列中的可独立刷新的存储器单位的地址。

21. 根据权利要求 19 所述的 DRAM 组件,其进一步包括操作以在将数据写入到所述相关联的可独立刷新的存储器单位时设置指示符的电路。

22. 根据权利要求 19 所述的 DRAM 组件,其中所述指示符在初始化期间被清除。

23. 根据权利要求 19 所述的 DRAM 组件,其中所述控制器进一步操作以响应于控制信号来设置或清除所述指示符。

24. 根据权利要求 19 所述的 DRAM 组件,其中所述指示符存储于一个或一个以上可独立刷新的存储器单位中。

25. 根据权利要求 19 所述的 DRAM 组件,其中所述指示符存储于不同于所述 DRAM 阵列的存储器中。

26. 一种至少包含存储器管理器计算机程序的计算机可读媒体,所述存储器管理器计算机程序操作以执行以下步骤:

接收来自软件任务的对将存储器分配给所述任务的请求;

从存储器装置中的一个或一个以上可独立刷新的存储器单位将存储器分配给所述任务;以及

设置与每一可独立刷新的存储器单位相关联的指示符,以引导所述存储器装置不抑制针对所述可独立刷新的存储器单位的刷新操作。

27. 根据权利要求 26 所述的计算机可读媒体,其中所述存储器管理器计算机程序进一步操作以执行以下步骤:

接收来自所述软件任务的对释放先前分配给所述任务的存储器的请求;

释放先前分配给所述任务的存储器;以及

如果已释放了可独立刷新的存储器单位中的所有所述存储器,那么清除所述相关联的指示符以引导所述存储器装置抑制针对所述可独立刷新的存储器单位的刷新操作。

28. 根据权利要求 26 所述的计算机可读媒体,其中所述存储器管理器计算机程序将存储器分配给在单个处理器上执行的多个软件任务。

29. 根据权利要求 26 所述的计算机可读媒体,其中所述存储器管理器计算机程序将存储器分配给两个或两个以上软件任务,每一软件任务在不同的处理器上执行。

30. 根据权利要求 26 所述的计算机可读媒体,其中所述存储器管理器计算机程序在最初将数据写入到所述相关联的可独立刷新的存储器单位后设置可独立刷新的存储器单位指示符。

经由使用有效数据指示符减少动态 RAM 功率消耗的系统和方法

技术领域

[0001] 本发明大体上涉及存储器领域,且特定来说,涉及用于减少动态 RAM 功率消耗的系统和方法。

背景技术

[0002] 固态动态随机存取存储器 (DRAM) 是用于许多现代计算系统 (包含便携式电子装置) 的具成本效益的大容量存储器解决方案。DRAM (包含同步 DRAM (SDRAM)) 与更快速的芯片上存储器结构 (例如,寄存器、静态 RAM (SRAM) 等) 相比提供较高的位密度和相对较低的每位成本,且提供比电、磁或光学机械大容量存储装置 (例如,硬盘、CD-ROM 等) 高得多的存取速度。

[0003] 图 1 描绘代表性 512 兆位 DRAM 阵列 100 的逻辑图。阵列 100 被组织为多个可单独寻址的排 102、104、106、108。每一排被划分为大量 (例如,4096 个) 行 110。每一行 110 被划分为多个列 (例如,512 个列),且每一列包含许多数据位,其通常被组织为字节 (例如,8 个字节)。此项技术中已知若干数据寻址方案。举例来说,在排、行、列 (BRC) 寻址中,可将存储器地址解译为

[0004]

31-26	25-24	23-12	11-3	2-0
芯片选择	排选择	行选择	列选择	字节选择

[0005] 在替代的寻址方案 (例如,行、排、列 (RBC) 寻址) 中,可将存储器地址解译为

[0006]

31-26	25-14	13-12	11-3	2-0
芯片选择	行选择	排选择	列选择	字节选择

[0007] DRAM 存储器阵列是易失性的;存储在 DRAM 阵列中的数据必须经周期性地刷新以维持其完整性。在 DRAM 刷新操作期间,大量数据存储位置被同时从阵列 100 中读出且被再充电。常规上,逐行地刷新 DRAM 阵列。也就是说,选择一行 (或者,在某些实施方案中,同时选择每个排中的同一行) 且在单个操作中刷新所述行内的所有数据。如本文中所使用,术语“可独立刷新的存储器单位”或 IRMU 是指在单个刷新操作中刷新的数据量。用于 DRAM 阵列的 IRMU 通常为行,但本发明不限于逐行刷新操作。

[0008] 针对 IRMU 的刷新操作常规上与存储器存取交替,且被定时以使得在任何数据由于电荷衰减而丢失之前刷新整个 DRAM 阵列。传统上,由存储器控制器 (例如,处理器) 供应刷新地址 (即,每一可独立刷新的存储器单位的地址),所述存储器控制器通过唯一控制信号组合来指定刷新操作。现代的 SDRAM 组件可包含两个额外的刷新模式:自刷新和自动刷

新。在两种模式中,SDRAM 组件包含内部刷新地址计数器。自刷新用于许多系统(例如,电池供电的电子装置)中,所述系统采用“休眠”模式来节省功率。在自刷新模式中,不可存取 SDRAM 组件来存储或检索数据;然而,SDRAM 在内部执行刷新操作以确保所存储数据的完整性。在自动刷新模式中,存储器控制器指定刷新操作,但不提供刷新地址。而是,SDRAM 组件使内部刷新地址计数器递增,所述计数器提供连续的可独立刷新的存储器单位(例如,行)地址。

[0009] 每一刷新操作在从 DRAM 阵列读取数据且对其进行再充电时消耗功率。然而,尤其在通电或系统复位后,DRAM 阵列中的大多数存储器存储位置不含有有效数据。

发明内容

[0010] 根据本文中所揭示和主张的一个或一个以上实施例,维持指示符,其指示可刷新的存储器片段是否含有有效数据。当刷新操作针对相关联的存储器时,如果所述存储器不含有有效数据,那么抑制所述刷新操作。可通过抑制针对无效数据的刷新操作来实现显著的功率节省。

[0011] 一个实施例涉及一种刷新动态存储器的方法。指示符与每一可独立刷新的存储器单位相关联。在将数据写入到可独立刷新的存储器单位后,设置相关联的指示符以反映有效数据。仅刷新其相关联的指示符反映其中存储有有效数据的可独立刷新的存储器单位。

[0012] 一个实施例涉及一种 DRAM 组件。所述 DRAM 组件包含 DRAM 阵列,所述 DRAM 阵列操作以存储数据且被组织为多个可独立刷新的存储器单位。所述 DRAM 组件还包含多个指示符,每一指示符与可独立刷新的存储器单位相关联且指示是否有有效数据存储于所述可独立刷新的存储器单位中。所述 DRAM 组件进一步包含控制器,所述控制器接收控制信号且操作以检查指示符并仅刷新存储有效数据的可独立刷新的存储器单位。

附图说明

[0013] 图 1 是 DRAM 阵列中的数据组织的功能框图。

[0014] 图 2 是 DRAM 阵列中的数据组织的功能框图,其中有效指示符或位与每一可独立刷新的存储器单位相关联。

[0015] 图 3 是 SDRAM 组件的功能框图。

[0016] 图 4 是单处理器计算系统的功能框图。

[0017] 图 5 是多处理器计算系统的功能框图。

[0018] 图 6 是刷新 DRAM 阵列的方法的流程图。

具体实施方式

[0019] 图 2 描绘根据一个实施例的 DRAM 阵列 200 组织的逻辑图。阵列 200 在逻辑上被组织为四个排 202、204、206、208,每一排包括 4096 个行。代表性行被描绘为 210。在此实施例中,行 210 是最小的可独立刷新的存储器单位。指示符 211 与阵列 200 中的行 210 相关联,所述指示符 211 反映行 210 是否含有有效数据。在所描绘的实施例中,每一指示符 211 包括单个位,其在本文中也称作有效位,与每一行相关联。图 2 描绘若干组指示符位 212、214、216、218,每一指示符位分别与排 202、204、206、208 中的一行相关联。在最小的可独立

刷新的存储器单位包括横跨所有四个排 202、204、206、208 的行的实施例中,将仅需要一组指示符位 212。

[0020] 在刷新操作期间,检查与当前寻址的可独立刷新的存储器单位(例如,IRMU 210)相关联的指示符或有效位(例如,指示符 211)。如果指示符位被设置,从而指示相关联的 IRMU 含有有效数据,那么对所述 IRMU 执行刷新操作以维持数据。如果指示符位未被设置,从而指示相关联的 IRMU 不含有有效数据,那么在一个实施例中,抑制刷新操作,从而节省了原本将在刷新 IRMU 中消耗的功率。因此,将仅刷新含有有效数据的 IRMU,且不刷新阵列中的处于未经初始化或“无关”状态的 IRMU。例如在自动刷新或自刷新期间,可由存储器控制器供应或可由内部地址计数器产生刷新地址。

[0021] 可以多种方式维持指示符位。在一个实施例中,将指示符位存储于存储器阵列 200 的固定或可编程部分中。在此情况下,阵列 200 的可使用大小减少了 0.003%。在另一实施例中,将指示符位存储在存储器中的除 DRAM 阵列 200 以外的 DRAM/SDRAM 组件上,例如存储于静态 RAM 结构中,存储于寄存器中,等等。在一个实施例中,经由类似于 SDRAM 组件的模式寄存器和扩展模式寄存器存取序列的 2 循环序列来存取 IRMU 有效指示符存储器。

[0022] 图 3 是根据一个实施例的 SDRAM 组件 300 的功能框图。SDRAM 300 包含 DRAM 阵列 301,其被组织为四个排 302、304、306、308。每一排包含行和列解码器 312、310。读出放大器 314 将来自 DRAM 阵列 301 的读取数据提供到 I/O 缓冲器 316。来自 I/O 缓冲器 316 的写入数据穿过输入缓冲器 318,且在写入到 DRAM 阵列 301 中之前存储于写入数据寄存器 320 中。

[0023] 由状态机 322 控制 SDRAM 组件 300 的操作。排和存储器地址被输入到地址缓冲器 324 且存储于地址寄存器 326 中,其中其控制列预解码器和计数器电路 328。模式寄存器 330 和扩展模式寄存器 332 存储模式选择位,例如列地址选通(CAS)延迟、突发长度等,其控制突发计数器 334 和数据输出控制电路 336 的操作。

[0024] 刷新逻辑和定时器电路 338 从内部计数器 340 接收 IRMU 地址,并从 IRMU 有效存储器 342 接收 IRMU 有效位。所述刷新逻辑将 IRMU 地址输出到行预解码器 344。请注意,虽然在图 3 中将 IRMU 有效存储器 342 描绘为单独的且与 DRAM 阵列 301 远离的功能块,但在物理上专用于存储 IRMU 有效指示符的存储器可为 DRAM 阵列 301 的一部分,或可为单独的 DRAM、SRAM、寄存器或其它存储器。

[0025] 在一个实施例中,SDRAM 组件 300 自动监视写入地址,且设置对应于每一写入操作所针对的 IRMU 的 IRMU 有效指示符。刷新逻辑 338 随后在每一刷新操作时检查 IRMU 有效存储器 342,且抑制针对不含有有效数据的任何 IRMU 的刷新循环。这将 SDRAM 组件 300 的功率消耗减到最少,但不需要存储器控制器或处理器的任何刷新抑制认知或参与。当 DRAM 阵列 301 大多数不具有有效数据时,主要的功率节省有可能发生在通电或复位之后。随着存储器控制器将数据写入到 DRAM 阵列 301 中的较多 IRMU,较多的 IRMU 有效位被设置,且较少的刷新循环被抑制。在此实施例中,IRMU 存储器 342 在加电或复位之后作为 SDRAM 组件 300 初始化的一部分被自动清除。此实施例允许系统设计者利用 SDRAM 组件 300 的较低功率消耗,同时利用不包含复杂的存储器管理功能性的现有存储器控制器和软件。

[0026] 在一个实施例中,可由来自存储器控制器的命令(例如,预定义的模式寄存器 330 或扩展模式寄存器 332 写入操作或位模式)清除 IRMU 存储器 342。此实施例允许在软

(即,软件起始的)复位之后获得减少的 SDRAM 组件 300 功率消耗,但需要存储器控制器发布 IRMU 存储器 342 清除命令。

[0027] 图 4 描绘控制并减少 DRAM 功率消耗的计算系统 400。系统 400 包含例如处理器 402、存储器控制器硬件 404(其可与处理器 402 集成)等存储器控制器以及例如 SDRAM 组件 406 等存储器装置。SDRAM 组件 406 内的 DRAM 阵列在逻辑上被划分为可独立刷新的存储器单位 408a、408b、408c、...408j。例如位 410a、410b、410c、...410j 等 IRMU 有效指示符与每一 IRMU 相关联,其指示相关联的 IRMU 是否含有有效数据。

[0028] 多个软件任务 412、414 在处理器 402 上执行。每一软件任务可分配存储器以用于数据存储,且可释放不再需要的存储器。软件存储器管理器 416 是管理用于处理器 402 的存储器的软件模块。软件存储器管理器 416 从软件任务 412、414 接收存储器“分配”和/或“释放”请求。作为响应,软件存储器管理器 416 将存储器分配给任务 412、414 和从所述任务 412、414 分配存储器,将所分配的存储器映射到一个或一个以上可独立刷新的存储器单位 408a、408b、408c、...408j(例如,行),且设置和清除对应的 IRMU 有效指示符 410a、410b、410c、...410j 以反映当前在 IRMU 408a、408b、408c、...408j 中的数据的状态。在一个实施例中,实际的存储器控制器是独立的硬件元件 404;在另一实施例中,存储器控制器功能性被集成到处理器 402 中。SDRAM 组件 406 抑制针对含有无效数据的 IRMU 408a、408b、408c、...408j 的所有刷新操作。

[0029] 图 5 描绘控制存储器分配并将 SDRAM 功率消耗减到最少的多处理器系统 500。处理器 502、504 在系统总线 506 上彼此通信且与存储器控制器硬件 508 通信。总线 506 也可实施为交换构造、纵横开关等,如此项技术中已知的。一个或一个以上软件任务 503、516、518 在处理器 502、504 上执行。系统范围的软件存储器管理器 520 在一个处理器 504 上执行,从而将存储器分配给在系统中执行的所有软件任务 503、516、518 和从所述软件任务 503、516、518 分配存储器。在处理器 502 上执行的任何软件任务 503 可在总线 506 上将存储器分配和释放请求发送到软件存储器管理器 520。如上文所描述,软件存储器管理器 520 将存储器分配给任务 503、516、518 和从所述任务 503、516、518 分配存储器,将所分配的存储器映射到一个或一个以上可独立刷新的存储器单位 512a、512b、512c、...512j,且经由存储器控制器硬件 508 设置和清除对应的 IRMU 有效指示符 514a、514b、514c、...514j 以反映当前存储在 IRMU 512a、512b、512c、...512j 中的数据的状态。SDRAM 组件 510 抑制针对含有无效数据的 IRMU 512a、512b、512c、...512j 的刷新操作。

[0030] 在常规的刷新模式、自动刷新模式或自刷新模式中,SDRAM 组件 300、406、510 将刷新地址(由存储器控制器或内部计数器供应)与 IRMU 有效存储器 342、410、514 进行比较,且抑制针对不含有有效数据的 IRMU 408、512 的刷新操作。在一个实施例中,其中软件存储器管理器 416、520 主动管理存储器并设置/清除 IRMU 有效位 410、514,所述系统可通过在物理存储器从分配给软件任务中释放且返回到“池”中时动态地抑制去往 IRMU 的刷新命令来进一步优化存储器刷新且将功率消耗减到最少,在此情况下其数据内容不相关。

[0031] 在常规的刷新模式中,软件存储器管理器 416、520 可仅提供含有有效数据的 IRMU 408、512 的刷新地址。在自动刷新或自刷新模式中,SDRAM 组件 300、406、510 可通过在每一刷新操作之后将其刷新地址计数器递增到含有有效数据的下一 IRMU 408、512 而“跳过”无效的存储器。在任一情况下,存储器控制器 404、508 可增加刷新操作之间的延迟,使

得用最大刷新周期仅全部刷新含有有效数据的 IRMU 408、512。在此实施例中, SDRAM 组件 300、406、510 不抑制刷新命令。这通过避免不必要的存储器命令循环而进一步优化功率消耗(并减少总线拥塞),且减少强加于进行中的存储器存取的延迟刷新命令。

[0032] 图 6 描绘根据一个或一个以上实施例的刷新 DRAM 的方法 600。在初始化后,清除所有 IRMU 指示符(方框 602)。所述方法随后检查是否将执行刷新操作(方框 604)。在传统的刷新模式中,由从存储器控制器发送到 DRAM 组件的控制信号来指示刷新操作,且在地址总线上指示待刷新的 IRMU。在自动刷新模式中,由存储器控制器命令刷新操作,且内部计数器提供 IRMU 刷新地址。在自刷新模式中,刷新定时器的期满指示需要刷新操作,且内部计数器提供 IRMU 地址。

[0033] 如果指示刷新操作(方框 604),那么检查与当前 IRMU 地址(例如,行地址)相关联的 IRMU 指示符(方框 606)。如果 IRMU 指示符指示 IRMU 含有有效数据(方框 608),那么对所寻址的 IRMU 执行刷新操作(方框 610)。如果 IRMU 指示符指示所述 IRMU 不含有有效数据(方框 608),那么抑制刷新操作,从而节省原本将由刷新无效(或“无关”)数据所耗费的功率。

[0034] 在自刷新模式中, SDRAM 组件在方框 604 处等待刷新地址计数器的下一期满。在其它刷新模式中,如果不命令刷新操作(方框 604),那么 DRAM(或 SDRAM) 组件执行存储器控制器所命令的读取、写入和/或寄存器存取操作(方框 612)。在一个实施例中,其中存储器管理软件模块分配和释放存储器块,存储器或寄存器存取操作可包含针对 IRMU 存储器的操作——读取、设置以及清除 IRMU 指示符。在一个实施例中,在针对相关联的 IRMU 的写入操作后自动设置 IRMU 指示符(方框 614)。在此实施例中,仅在初始化后清除 IRMU 指示符(方框 602),但可在至少一次将有效数据写入到许多 IRMU 之前提供显著的功率节省。

[0035] 通过对 DRAM 阵列的物理刷新操作应用存储器管理的软件范例(其中存储器仅在被分配给任务时才为相关的,且在分配之前或在释放之后呈现“无关”状态),可通过消除针对未保持有效数据的存储器片段的不必要的刷新操作而实现显著的功率节省。在一个实施例中,通过设置相关联的 IRMU 位来对有效数据进行跟踪是自动的。在此实施例中,在不具有对用以选择性地抑制刷新操作的能力的软件存储器管理或认知的系统中可得到本发明的功率节省益处。在其它实施例中,对 IRMU 存储器的直接控制允许实现复杂的存储器管理和最大的功率节省。

[0036] 如本文中所使用,术语“可独立刷新的存储器单位”或 IRMU 是指在单个刷新操作中刷新的数据量。用于 DRAM 阵列的 IRMU 通常为行,但本发明不限于此。如本文中所使用,术语“设置”是指将数据写入到 IRMU 指示符以指示有效数据存储于相关联的 IRMU 中,而不管所述数据的值(例如,0 或 1,或多位模式)。“清除”是指将数据写入到 IRMU 指示符以指示有效数据不存储于相关联的 IRMU 中,而不管所述数据的值(例如,0 或 1,或多位模式)。如本文中所使用,“DRAM 阵列”是指动态随机存取存储器阵列,其将数据存储于 DRAM 和 SDRAM 集成电路组件两者中。如本文中所使用,术语单独的“DRAM”或“DRAM 组件”的范围包含异步 DRAM 存储器组件和 SDRAM 组件两者。如本文中所使用,术语“分配”是指将一定范围的存储器地址指派给软件任务,且术语“释放”是指将先前分配的存储器地址返回到未经分配的存储器的池。

[0037] 虽然本文中已关于本发明的特定特征、方面和实施例描述了本发明,但将明白,能

够在本发明的广泛范围内做出许多变化、修改和其它实施例,且因此,所有变化、修改和实施例应被视为处于本发明的范围内。因此,本发明的实施例在所有方面应被解释为说明性的而不是限制性的,且希望在其中涵盖属于所附权利要求书的含义和等效范围内的所有改变。

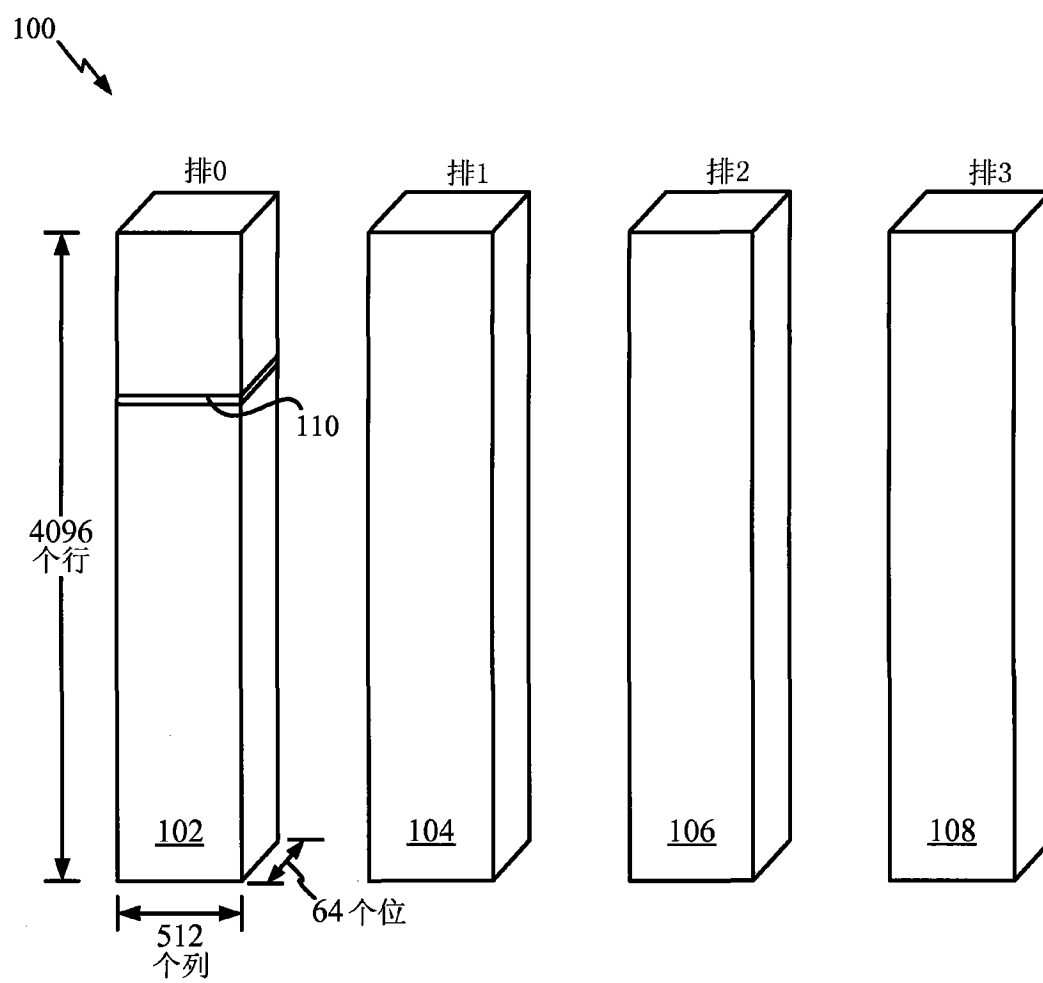


图 1 现有技术

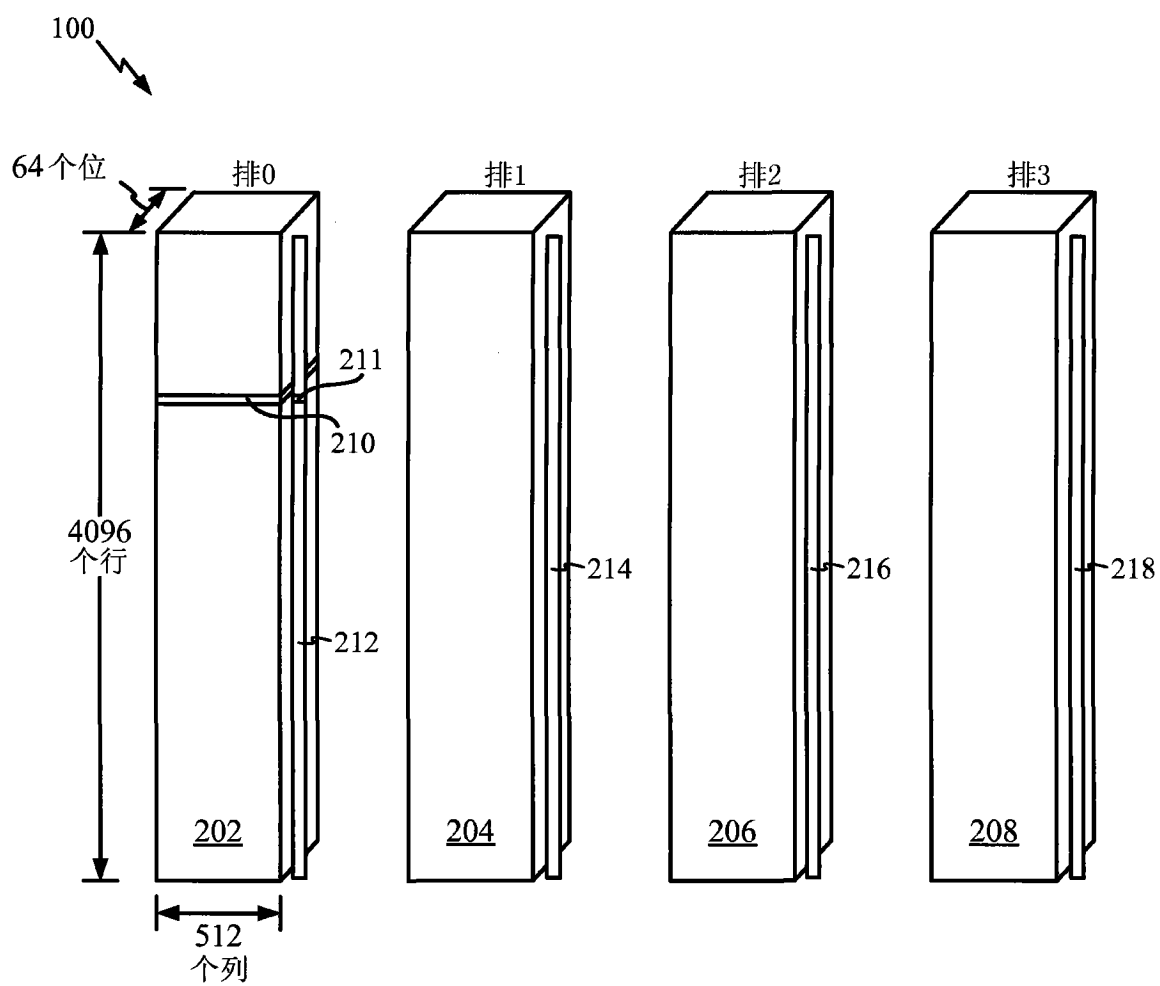


图 2

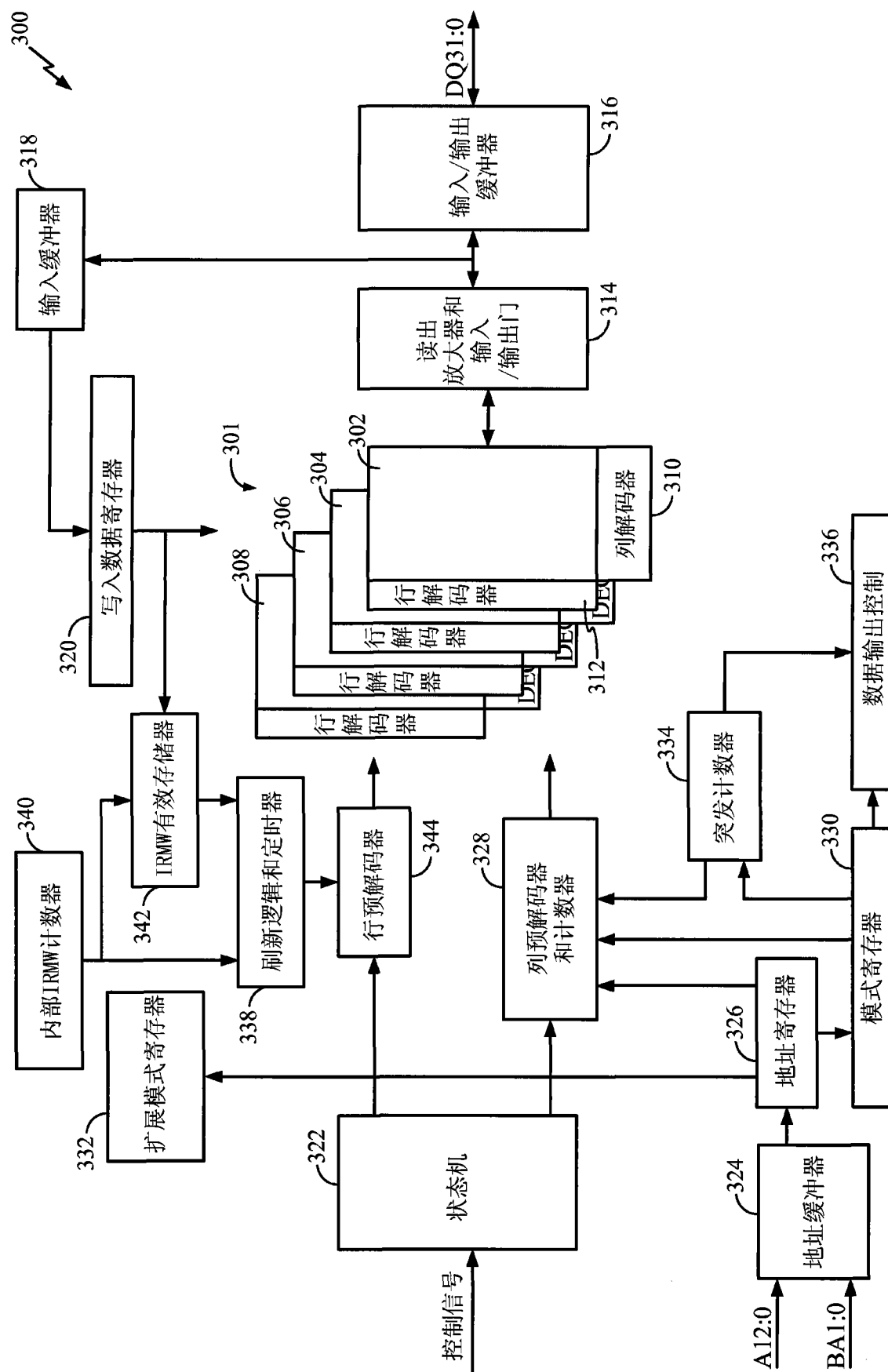


图 3

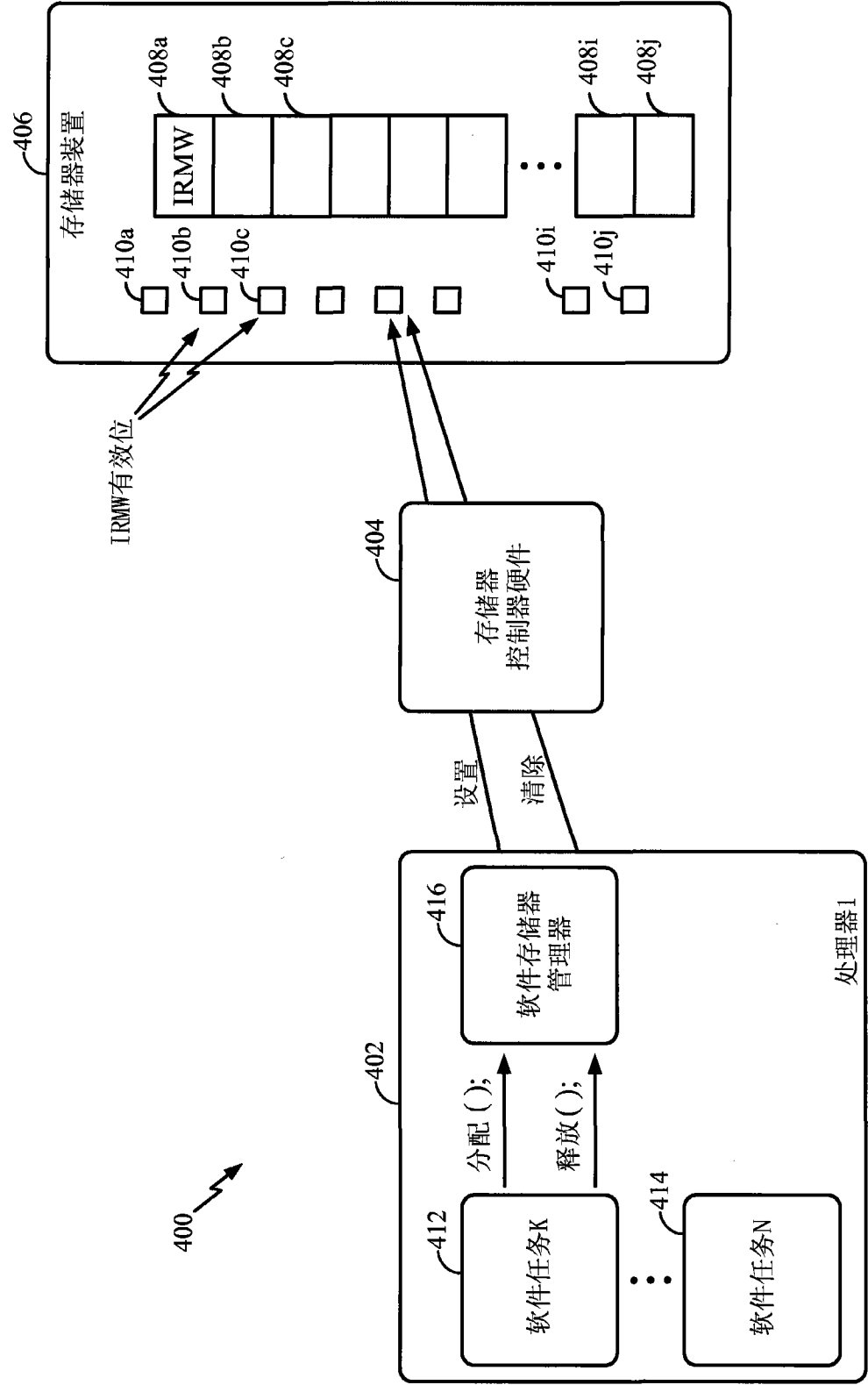


图 4

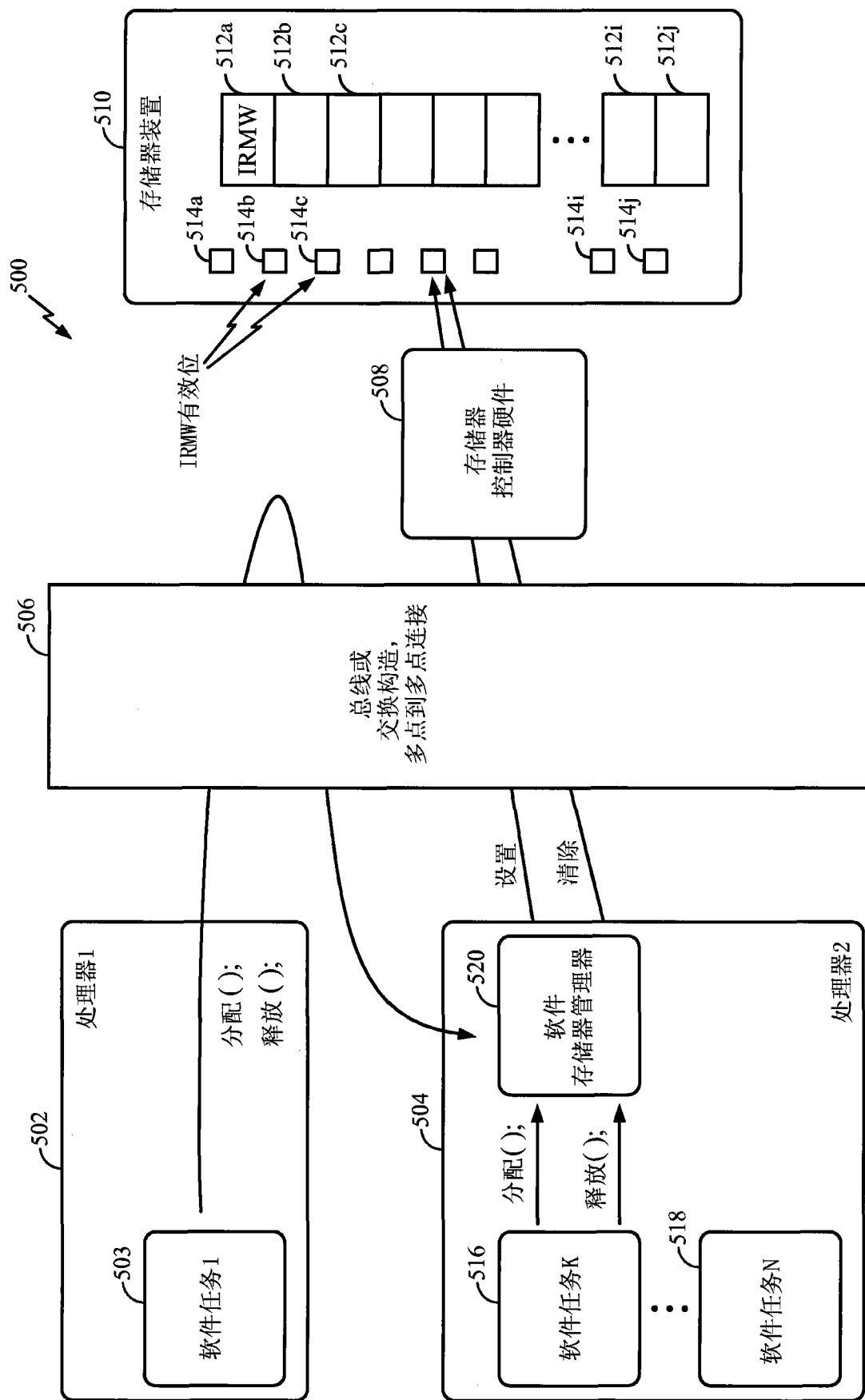


图 5

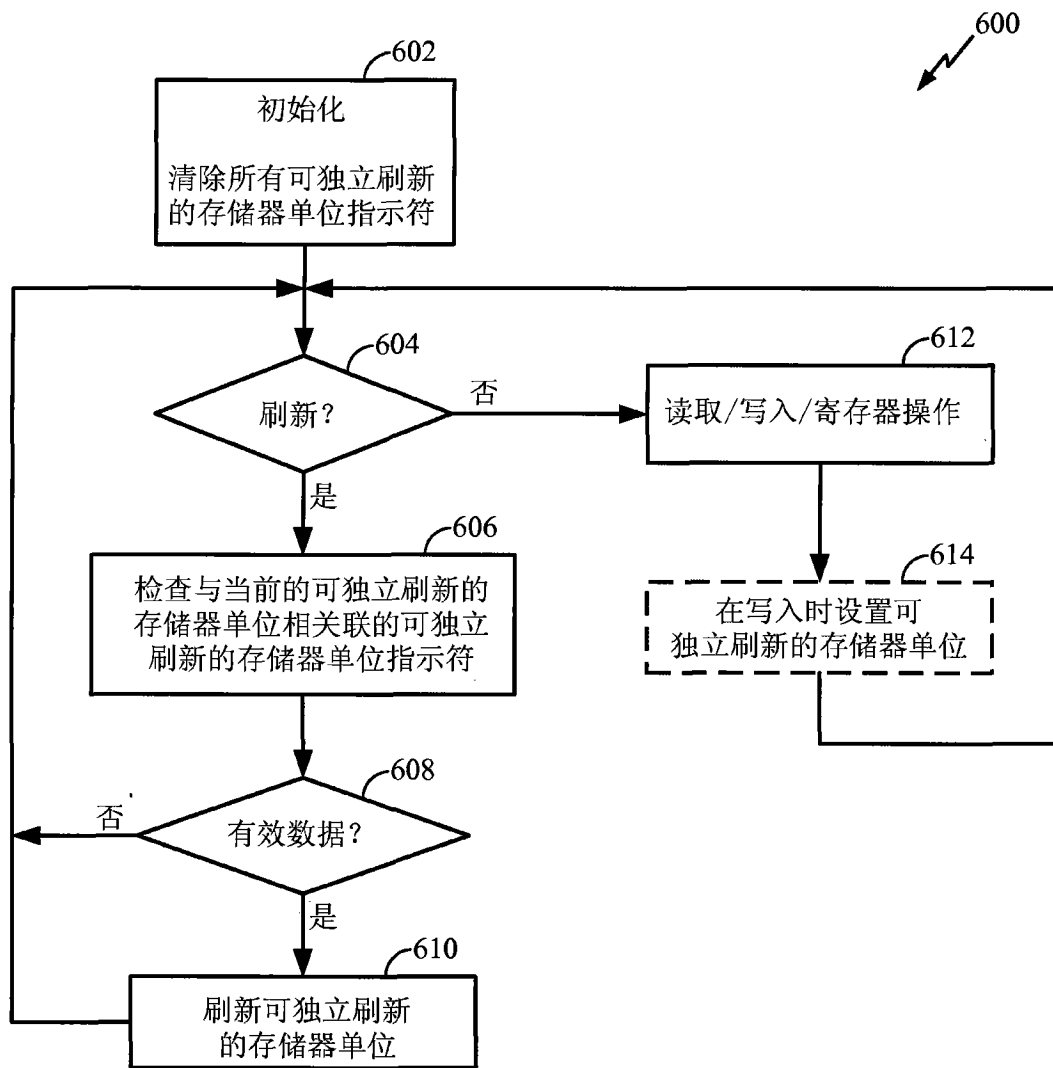


图 6