

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

G11B 20/18

G11B 19/04



[12] 发明专利申请公开说明书

[21] 申请号 200410059048.6

[43] 公开日 2005 年 2 月 9 日

[11] 公开号 CN 1577587A

[22] 申请日 2004. 7. 29

[21] 申请号 200410059048.6

[30] 优先权

[32] 2003. 7. 29 [33] US [31] 10/629,675

[71] 申请人 日立环球储存科技荷兰有限公司

地址 荷兰阿姆斯特丹

[72] 发明人 理查德·M.·H·纽

斯潘塞·W.·影

[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所

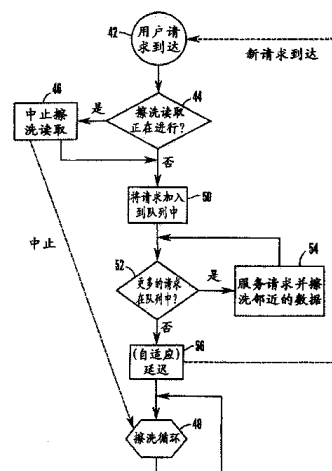
代理人 李德山

权利要求书 5 页 说明书 10 页 附图 8 页

[54] 发明名称 硬盘驱动器的自主数据擦洗系统及方法

[57] 摘要

具有多个硬盘驱动器的 RAID 系统使其每个控制器自主地对其磁盘执行数据擦洗，而不是需要 RAID 控制器来执行数据擦洗。可以响应来自 RAID 控制器的 I/O 请求而中断该数据擦洗操作，或者如果没有达到有保证擦洗速率，可以继续擦洗。数据被存储于同心数据磁道的大段中，可以基于对该段的读取频率或写入该段的新近程度，优先选择对该段进行擦洗。



ISSN 1008-4274

1. 一种硬盘驱动器，包括：
至少一个存储盘；
至少一个驱动器控制器，用于针对该盘读出数据或写入数据，该驱动器控制器执行包括下述的逻辑：
执行擦洗周期，包含：
读取至少一个数据单元；
确定该数据单元中是否存在错误，如果存在错误，采取以下步骤的至少一个：记录错误，和报告错误；
对后继数据单元执行读取和确定逻辑；
如果接收到用于以下操作中的至少一个的用户请求：数据读取和数据写入，则中断该擦洗周期以完成该请求，然后恢复该擦洗周期。
2. 如权利要求 1 中的磁盘驱动器，其与 RAID 系统相结合，其中由驱动器控制器从 RAID 控制器接收用户请求。
3. 如权利要求 2 中的磁盘驱动器，包括在重新恢复擦洗周期之前，在完成用户请求队列中的所有用户请求之后等待一个延迟周期。
4. 如权利要求 3 中的磁盘驱动器，其中延迟周期是自适应的延迟周期，它取决于从 RAID 控制器接收的用户请求的频率和数量中的至少一个。
5. 如权利要求 1 中的磁盘驱动器，其中磁盘驱动器被划分成数据段，驱动器控制器维护至少指示对这些段的最近访问的段表。
6. 如权利要求 1 中的磁盘驱动器，包括对磁盘上与用户请求所涉及的数据直接相邻的数据执行擦洗周期，以作为完成该请求的一部分。
7. 如权利要求 1 中的磁盘驱动器，包括对比第二数据更新近写入的第一数据执行擦洗周期，然后擦洗第二数据。
8. 如权利要求 1 中的磁盘驱动器，包括对比其它较少频繁使用的磁盘区域更频繁访问的磁盘区域执行擦洗周期，然后再擦洗该较少

频繁使用的区域。

9. 一种硬盘驱动器，包括：

至少一个存储盘；

至少一个驱动器控制器，用于对该盘读取和写入数据，该驱动器控制器执行包括以下的逻辑：

执行擦洗周期，包含：

读取至少一个数据单元；

确定数据单元中是否存在错误，如果存在错误，采取以下步骤的至少一个：记录错误，和报告错误；

对后继数据单元执行读取和确定逻辑；

如果接收到用于以下操作中的至少一个的用户请求：数据读取和数据写入，并且如果当前擦洗速率超过门限速率，则中断该擦洗周期以完成该请求，然后重新恢复该擦洗周期，否则不中断擦洗周期的至少已有擦洗读取。

10. 如权利要求 9 中的硬盘驱动器，其与 RAID 系统相结合，其中由驱动器控制器从 RAID 控制器接收用户请求。

11. 如权利要求 10 中的硬盘驱动器，包括在重新恢复擦洗周期之前，在完成用户请求队列中的所有用户请求之后等待一个延迟周期。

12. 如权利要求 11 中的硬盘驱动器，其中延迟周期是自适应的延迟周期，它取决于从 RAID 控制器接收的用户请求的频率和数量中的至少一个。

13. 如权利要求 9 中的硬盘驱动器，其中磁盘驱动器被划分成数据段，驱动器控制器维护至少指示对这些段的最近访问的段表。

14. 如权利要求 9 中的硬盘驱动器，包括对磁盘上与用户请求所涉及的数据直接相邻的数据执行擦洗周期，以作为完成该请求的一部分。

15. 如权利要求 9 中的硬盘驱动器，包括对比第二数据更新近写入的第一数据执行擦洗周期，然后擦洗第二数据。

16. 如权利要求 9 中的硬盘驱动器，包括对比其它较少频繁使用

的磁盘区域更频繁访问的磁盘区域执行擦洗周期，然后再擦洗该较少频繁使用的区域。

17. 一种硬盘驱动器，包括：

至少一个存储盘；

至少一个驱动器控制器，用于对该盘读取和写入数据，该驱动器控制器执行包括以下的逻辑：

执行擦洗周期，包含：

读取至少一个数据单元；

确定数据单元中是否存在错误，如果存在错误，采取以下步骤的至少一个：记录错误，和报告错误；

对后继数据单元执行读取和确定逻辑，其中对比第二数据更新近写入的第一数据执行擦洗周期，然后再对第二数据执行擦洗周期。

18. 如权利要求 17 中的硬盘驱动器，其中如果接收到用于以下操作中的至少一个的用户请求：数据读取和数据写入，则中断该擦洗周期以完成该请求，然后重新恢复该擦洗周期。

19. 如权利要求 18 中的硬盘驱动器，其与 RAID 系统相结合，其中由驱动器控制器从 RAID 控制器接收用户请求。

20. 如权利要求 17 中的硬盘驱动器，其中如果接收到用于以下操作中的至少一个的用户请求：数据读取和数据写入，并且如果当前擦洗速率超过门限速率，则中断该擦洗周期以完成该请求，然后重新恢复该擦洗周期，否则不中断擦洗周期的至少已有擦洗读取。

21. 一种硬盘驱动器，包括：

至少一个存储盘；

至少一个驱动器控制器，用于对该盘读取和写入数据，该驱动器控制器执行包括以下的逻辑：

执行擦洗周期，包含：

读取至少一个数据单元，其中如果该数据单元在磁盘上与用户请求所涉及的数据直接相邻，则优先选择该数据单元进行擦洗；

确定数据单元中是否存在错误，如果存在错误，采取以下步骤的

至少一个：记录错误，和报告错误；

对后继数据单元执行读取和确定逻辑。

22. 如权利要求 21 中的硬盘驱动器，其中如果接收到用于以下操作中的至少一个的用户请求：数据读取和数据写入，则中断该擦洗周期以完成该请求，然后重新恢复该擦洗周期。

23. 如权利要求 22 中的硬盘驱动器，其与 RAID 系统相结合，其中由驱动器控制器从 RAID 控制器接收用户请求。

24. 如权利要求 21 中的硬盘驱动器，其中如果接收到用于以下操作中的至少一个的用户请求：数据读取和数据写入，并且如果当前擦洗速率超过门限速率，则中断该擦洗周期以完成该请求，然后重新恢复该擦洗周期，否则不中断擦洗周期的至少已有擦洗读取。

25. 一种硬盘驱动器，包括：

至少一个存储盘；

至少一个驱动器控制器，用于对该盘读取和写入数据，该驱动器控制器执行包括以下的逻辑：

执行擦洗周期，包含：

读取至少一个数据单元；

确定数据单元中是否存在错误，如果存在错误，采取以下步骤的至少一个：记录错误，和报告错误；

对后继数据单元执行读取和确定逻辑，其中对比其它较少频繁使用的磁盘区域更频繁访问的磁盘区域执行擦洗周期，然后再对该较少频繁使用的区域执行擦洗周期。

26. 如权利要求 25 中的硬盘驱动器，其中如果接收到用于以下操作中的至少一个的用户请求：数据读取和数据写入，则中断该擦洗周期以完成该请求，然后重新恢复该擦洗周期。

27. 如权利要求 26 中的硬盘驱动器，其与 RAID 系统相结合，其中由驱动器控制器从 RAID 控制器接收用户请求。

28. 如权利要求 25 中的硬盘驱动器，其中如果接收到用于以下操作中的至少一个的用户请求：数据读取和数据写入，并且如果当前

擦洗速率超过门限速率，则中断该擦洗周期以完成该请求，然后重新恢复该擦洗周期，否则不中断擦洗周期的至少已有擦洗读取。

29. 一种独立磁盘冗余阵列 (RAID) 系统，包括 RAID 控制器和多个硬盘驱动器，每个硬盘驱动器包括至少一个存储盘，和至少一个用于对盘读取数据和写入数据的驱动器控制器，其中每个磁盘驱动器的驱动器控制器与 RAID 控制器相连，每个驱动器的驱动器控制器自主地对其磁盘执行擦洗周期，并且如果接收到用于以下操作中的至少一个的用户请求：数据读取和数据写入，则中断该擦洗周期以完成该请求，然后重新恢复该擦洗周期。

硬盘驱动器的自主数据擦洗系统及方法

技术领域

本发明一般性地涉及硬盘驱动器, 尤其涉及但不限于独立磁盘冗余阵列(RAID)系统的硬盘驱动器。

背景技术

数据完整性是许多数据存储子系统最为关心的问题。这些子系统使用各种 RAID 技术来提供数据的冗余, 使得当数据无法从一个磁盘驱动器获取时, 该数据能够从 RAID 子系统的其它驱动器中重新获得。多数子系统仅提供单容错, 因为多容错的成本很高。不幸的是, 不能恢复的读错误(也被称之为硬错误)在磁盘驱动器中出现。如果即使在进行完所有的驱动器错误恢复过程(ERP)步骤之后, 扇区都不能读出, 就引起这种读取失败的情形。这样的步骤包含调用所有级别的已经实现的错误纠正控制(ECC), 并使用不同的磁头偏移量进行重试。当遇到不可恢复的读取错误, 而在 RAID 中所有其它的驱动器仍然可以访问时, 该扇区能够由那些驱动器重构。不过, 如果当 RAID 中的一个驱动器已经失败时遇到硬错误, 则读取失败的驱动器中的硬错误和其对应扇区是不能恢复的。这造成了数据的丢失情况。

为了减少在驱动器已经失败后遇到硬错误的概率, 某些存储子系统实现某种数据擦洗(scrubbing)例程, 以在实际需要该数据之前刷新掉这样的错误。在这种方案中, RAID 控制器本身周期性地向每个驱动器控制器发出验证读(即, 不从高速缓存中读取, 而是从磁盘介质中读取)命令, 并且循环遍历每个驱动器的每个地址。当一个驱动控制器遇到不可恢复的错误并报告该错误时, RAID 控制器将利用 RAID 的冗余性从其它驱动器重构该数据。某些子系统在磁盘驱动器中启用自动重新分配, 使得具有硬错误的扇区自动地被重新分配到空余的位置上。而其它则关掉自动重新分配, 使得 RAID 控制器能够决定是否

首先尝试重写该数据回到最初的位置，以检查问题是否得到清除，从而只有当问题持续存在（即，介质上的划痕）时，才进行重新分配。在所有这些情形中，RAID 控制器记录不可恢复错误的统计数据。当这样的错误超过预定的门限时，调度该驱动器准备替换。

上述的子系统控制器管理的数据擦洗具有两个缺点。第一，RAID 系统控制器的资源需要管理和执行该擦洗过程。第二，即使控制器实际不需要数据，数据也从驱动器返回。这在总线上产生不必要的业务量并且可能潜在地退化系统的性能。由于这种限制，数据擦洗周期可以比希望的要长。

本发明认识到上述问题能够通过通过在磁盘驱动器控制器级别上实施数据擦洗而得到缓解，而不需要 RAID 控制器的任何动作，除非遇到了不可恢复的错误。

发明内容

硬盘驱动器包含至少一个存储磁盘和从磁盘中读出数据和写入数据的驱动器控制器。驱动器控制器执行这样的逻辑，包含执行需要读出至少一个数据单元和确定该数据单元中是否存在错误的擦洗周期 (scrub cycle)。如果存在，则错误被记录和/或报告。针对随后的数据单元执行错误检测逻辑。如果收到用户数据读出或数据写入请求，则擦洗周期被中断以完成该请求，然后重新恢复擦洗。

在优选实施例中，磁盘驱动器与 RAID 系统进行组合，并且由驱动控制器从 RAID 控制器接收用户请求。

在一个非限制性的实施例中，驱动器控制器逻辑包含在重新恢复擦洗周期之前，在满足用户请求队列中的所有用户请求之后等待一个延迟周期。延迟周期可以是自适应的延迟周期，它取决于从 RAID 控制器接收的用户请求的频率和/或数量。

磁盘驱动器可以被划分成数据段(data band)。驱动器控制器最好维护例如指示最近对段的访问的表。

可以优先对磁盘上与用户请求所涉及的数据直接相邻的数据进行擦洗，同时完成该请求。同样可以优先对比其它数据更新近写入的

数据, 和/或比其它磁盘区域更频繁访问的磁盘区域执行擦洗。

另一方面, 硬盘驱动器包含至少一个存储盘和针对该盘读出数据和写入数据的驱动器控制器。驱动器控制器执行这样的逻辑, 包含执行需要读出至少一个数据单元并确定该数据单元中是否存在错误的擦洗周期。如果存在错误, 该错误被记录和报告。对随后的数据单元执行错误检测逻辑。如果接收了数据读或写的用户请求, 并且如果当前擦洗速率超过了门限速率, 则中断该擦洗周期以完成该请求, 然后恢复擦洗周期。否则, 不中止擦洗周期中现有擦洗读取。

在另一个方面, 硬盘驱动器包含至少一个存储盘和针对该盘读出数据和写入数据的驱动器控制器。驱动器控制器执行这样的逻辑, 包含执行需要读出至少一个数据单元并确定该数据单元中是否存在错误的擦洗周期。如果存在错误, 那么该错误被记录和/或报告。对随后的数据单元进行错误检测逻辑。可以对比第二数据更新近写入的第一数据执行擦洗周期, 然后再对第二数据执行。或者, 可以优先对频繁访问的磁盘区域执行擦洗周期。

附图说明

本发明关于它的结构和操作的详细说明, 最好能够参照附图加以理解, 其中同样的参考数字表示同样的部分, 其中:

图 1 是与客户机处理器配合的 RAID 系统的方框图;

图 2 是图 1 中所示的系统中一个硬盘驱动器 (HDD) 的方框图;

图 3 是数据擦洗过程的流程图, 示出的是支持有保证擦洗速率 (guaranteed rate of scrub) 的可选逻辑;

图 4 是中断图 3 所示的擦洗过程的处理的流程图, 示出了将擦洗的恢复延迟一个固定或自适应的周期的可选逻辑, 和用于寻机 (opportunistic) 进行数据擦洗的可选逻辑;

图 5 示出的是利用图 3 中的擦洗过程建立有保证擦洗速率的过程的流程图;

图 6 是图 3 的擦洗过程期间用于选择要擦洗的下一个数据单元的优选过程的流程图;

图 7 是图 4 的擦洗中断逻辑期间用于执行用户请求的读或写的优选处理过程的流程图;

图 8 示出的是优选数据段表;

图 9 是维护图 8 的表的优选处理过程的流程图;

图 10 是选择要擦洗的单元的优选处理过程的流程图。

具体实施方式

从参照图 1 开始, 示出的是一般性地标明为 10 的 RAID 系统, 它包含为了存储和检索多个硬盘驱动器 (HDD) 16 中的数据而与一个或多个客户机处理器 14 通信的 RAID 控制器 12。图 2 示出的是本发明的硬盘驱动器 16, 它包含响应 RAID 控制器 12 的请求而针对一个或多个数据存储盘 20 读写数据的驱动器控制器 18。如现有技术已知的 HDD 原理所示, 每个 HDD 16 也可以包含固态数据高速缓存 22。根据本发明的原理, RAID 系统 10 的每个 HDD 控制器 18 自主执行下述的擦洗逻辑, 即每个驱动器控制器 18 均负责其自己的盘上的数据擦洗, 并且不为 RAID 控制器 12 所控制地自主进行。

图 3 示出的是本发明的擦洗循环的逻辑, 其最好由驱动器控制器 18 无限地重复进行。在开始状态 24 处开始, 当如参照图 5 所详细描述的那样实现有保证擦洗速率时, 逻辑移到框 26 以记录擦洗周期的开始时间。

当实现无保证擦洗速率时, 如参照下面的图 6 所讨论的或使用其它下面所讨论的准则, 从框 26 或从开始状态 24, 该逻辑继续进行到框 28 以选择下一个数据单元进行擦洗。在非限制性的实施例中, 数据以“段”的形式存储在磁盘 20 上, 其中段是通过对磁盘的可用空间进行逻辑划分而建立的。例如, 100GB 驱动器可以被划分成 100 个均为 1GB 的段, 或 1000 个均为 100MB 的段, 等等。单个数据单元可以是针对相对大数目的扇区, 例如 512 个扇区, 或针对整个磁道。对于全磁道读取的情形, 当进行擦洗读取操作时, 能够容易地使用零等待读取技术 (即一旦磁头在磁道上便开始读取), 使得旋转等待开销得以消除。正如下面进一步所提出的, 可以使用与擦洗数据单元对应的位映射来

跟踪已经擦洗过的单元，但是也可以使用任何其它的方法。在每个擦洗周期的开始处，位映射被重新设置。

一旦待被擦洗的数据单元被识别出来，在框 30 处读取该数据单元。在菱形决策框 32 中，驱动器控制器 18 确定在数据单元中是否存在有任何不可纠正的错误。如果检测到错误，该错误的位置在框 34 中被记录下来。可选地，可以在这个步骤中立即向 RAID 控制器 12 通报该错误。然后，在框 36 处，驱动器控制器 18 更新其有关已经擦洗过的数据单元的列表（即位映射）。当使用了有保证擦洗速率时，该逻辑也进行到框 38，其中擦洗速率通过用框 26 处所记录的擦洗过的数据量，除以自框 36 所记录的擦洗周期开始起已经过的总时间来确定。最后，在菱形决策框 40 中，确定整个磁盘是否已经得到了擦洗，如果不是，该逻辑循环回到框 28。否则，该逻辑重新在起始状态 24 开始另一个擦洗周期。

图 4 示出的是当启用无保证擦洗速率时，驱动器控制器 18 可以实现的逻辑。假定图 3 所示的擦洗周期正在不停地执行，在状态 42 处接收到对磁盘 20 进行读取或写入的用户请求（来自例如 RAID 控制器 12）。在菱形决策框 44 中，驱动器控制器 18 确定擦洗读取操作是否正进行，如果是，该擦洗读取操作在框 46 处被中止或暂停。图 4 中的虚线表示图 3 中（用多边形 48 来表示）的擦洗周期子程序得到通告或者暂停。

如果没有擦洗读取操作正在进行，从框 46 或从菱形决策框 44，该逻辑继续执行到框 50，以将 I/O 请求加入到请求队列中。通过移到菱形决策框 52 处，确定在队列中是否有任何未完成的请求。如果有，则在框 54 处对该请求进行服务。在选择队列中哪一个请求将在下一次得到服务时，可以使用任何调度算法，包含先到先服务，最短时间先服务等等。

在一个改进(augmentation)中，作为框 54 的逻辑的一部分，当完成用户请求的同时，可以寻机对磁盘上直接靠近用户请求所涉及的数据的数据执行擦洗循环。这能够通过扩展该请求以包含一个或多个邻

近的数据擦洗单元来完成。例如，读取请求可以是读取 128 个扇区，比如说从地址 5020 到地址 5147。当调用寻机擦洗时，驱动器控制器 18 可以读取含有所请求扇区的整个擦洗单元（比如说，从地址 5000 到 5999），从而在此同时擦洗整个单元。

如果需要，一旦已经完成了所有的用户 I/O 请求，该逻辑能够移到框 56 处，以在图 3 中的擦洗循环重新恢复之前等待一个延迟周期。该延迟周期可以是固定的，也可以是自适应的，并且取决于例如所接收的用户请求的频率（对于较多的请求，例如会导致较长的延迟周期）和/或取决于从 RAID 控制器接收的用户请求的数量，其中对于较多的请求，例如会导致较长的延迟周期。延迟周期增加了新请求不可能在近期到达的概率，即 HDD 正进入到空闲时间周期。如果新请求在该延迟周期内到达，该逻辑回到起始状态 42；否则，它重新进入状态 48 处的擦洗循环。

在确定自适应延迟周期时，驱动器控制器 18 能够计算自从状态 42 处所记录的最后用户请求的到达所经过的时间量。在这个例子中，利用这个经过时间计算待执行的延迟。这种计算的一个例子是延迟 = $k/\text{经过时间}$ ，其中 k 是常数。于是，如果该经过时间较短，则延迟较长，因为另一请求很快到达的可能性很高；如果该经过时间较长，则延迟较短，因为在一段延长的时间周期内没有任何关于 HDD 的活动，因此更频繁地进行擦洗不可能影响性能。如上所述，可以使用其它自适应的方法或方式。

图 5 示出了当希望至少按照可由用户定义的某个门限速率进行擦洗时能够遵循的逻辑。假定图 3 所示的擦洗循环不断地执行，在状态 58 处（从例如 RAID 控制器 12）所接收的对磁盘 20 进行读取或写入的用户请求。在菱形决策框 60 中，确定擦洗读取是否正在进行，如果是，则该逻辑继续到菱形决策框 63，以确定当前的擦洗速率（如图 3 中的框 38 所确定的）是否低于门限速率。如果不是，则该逻辑移到框 64，其中擦洗读取被中止或暂停。虚线表示图 3 的擦洗循环子程序（用多边形 66 来表示）被通知中止或暂停擦洗。

相反, 当在菱形决策框 62 中确定擦洗速率不足(即太慢)时, 在服务任何用户请求之前, 该逻辑移到框 68 以完成当前的擦洗读取。从菱形决策框 60(当没有擦洗读取正在进行时), 或从框 64 和 68(当读取正在进行并取决于擦洗速率时), 该逻辑移到框 70, 以将用户请求加入到请求队列中。

移到菱形决策框 72, 确定是否在队列中有任何未完成的请求。如果有, 在框 74 处服务请求。如果需要, 就象图 4 中逻辑的情形那样, 可以寻机对磁盘上与用户请求所涉及的数据直接相邻的数据执行擦洗循环, 同时作为框 74 中的逻辑部分, 完成该请求。

从框 74 处, 驱动器控制器 18 的逻辑移到框 76 处以计算新的擦洗速率, 然后继续到菱形决策框 78 处确定该擦洗速率是否太慢。如果是, 该逻辑重新进入图 3 的擦洗循环(如多边形 66 所示), 并且停留在状态 66 和 78 之间的循环中, 直到擦洗速率等于用户定义的门限。当发生此情况时, 该逻辑移回到菱形决策框 72。如图 5 所示, 当如菱形决策框 72 所确定的那样已经完成了队列中所有的用户请求时, 该逻辑可以在重新进入状态 66 的擦洗循环之前, 可选地移到框 80 以启用上述延迟周期。

如上所述, 本发明的优选实施例使用物理磁道或其一部分作为数据擦洗单元。当调度要擦洗的数据单元时, 不是顺序地从开始执行到结束, 而是可以调度下一个擦洗读取操作在最接近磁盘驱动器 16 的磁头的位置的附近进行。因此, 在一个非限制性实施例中, 图 3 的框 28 处执行的该逻辑可以通过图 6 所示的逻辑来实现。

在框 82 处开始, 磁头当前所在位置的物理柱面号“C”被记录下来。在菱形决策框 84 中, 驱动器控制器 18 确定柱面 C 上的所有磁道是否已经擦洗。如果至少有一个磁道还没有被擦洗, 则该逻辑移到框 86 以从该柱面选择未擦洗的磁道, 然后该逻辑移到图 3 的框 30。

当在决策框 84 处确定柱面“C”上的所有磁道已经擦洗时, 该逻辑移到框 88 处, 其中索引变量“i”被初始化成 1。继续进行到菱形决策框 90, 确定柱面 C+i 上所有磁道是否已经擦洗。如果不是这样, 该逻辑

移到框 86。否则，该逻辑继续进行到菱形决策框 92，其中确定柱面 C-i 上的所有磁道是否已经擦洗。如果在柱面 C-i 上还至少有一个磁道未得到擦洗，该逻辑移到框 86，但当菱形决策框 92 处的测试结果为肯定时，该逻辑移到框 94，以将索引变量“i”加 1。

继续进行到菱形决策框 96，确定索引变量“i”是否超过门限。如果不是，该逻辑回到菱形决策框 90。然而，如果超过了该门限，则进入框 98，其中可以使用不同的算法选择下一个擦洗单元。这就将搜索限制到合理的时间量，并且在“i”变得太大以致不具有任何性能优势之后，也切换到不同的选择方法。

作为可以用来选择要擦洗的数据单元的不同选择方法的非限制性例子，可以选择比其它数据更新近写入的数据在擦洗其它数据之前进行擦洗。或者，可以在擦洗其它不经常使用的区域之前擦洗频繁访问的磁盘区域。

更具体地，本发明认为到某些不可恢复的错误可以由坏写(bad write)引起。为了更快地发现这样的错误，驱动器控制器 18 能够调度擦洗最近写入的磁道和相邻磁道，如图 7 所示。根据分别结合图 4、图 5 中的框 54 和框 74 所进行的描述，对请求的服务可以包含在框 100 中执行所请求的读或写命令，然后，在菱形决策框 102 处，确定刚完成的命令是否写入命令。如果它是写入命令，在框 104 处，驱动器控制器 18 将受写入命令影响的擦洗单元标记为尚未擦洗过，不管它们在这个周期中是否已经擦洗。在使用位映射跟踪已擦洗单元的优选实施例中，那些单元的对应位被重新设置。以这种方式，那些单元在它们被调度进行擦洗之前，不会等待下一个擦洗周期，而是可以立刻得到擦洗。而且，当使用图 6 的调度策略时，这些数据单元有可能会被迅速地调度。在擦洗单元是物理磁道的优选实施例中，受影响的擦洗单元会是包含那些刚写入数据的磁道和同一表面上与其相邻的磁道。

如前所述，磁盘 20 可以被逻辑地分成不必大小相等的数据段。与段有关的是图 8 所示的表 106，每个表项对应一段。每个表项包含如第一列所示的计数域。为了减少计数，这些表项最好建立双链表，

因此每一表项包含这个链表的前向指针和后向指针，如第二列和第三列所示。除段表之外，HDD 维护一个指向这个链表的顶部（该表项具有最高的计数）的指针。

在这个非限制性实施方案中，并且现在参照图 9，每当在图 9 中状态 108 处 HDD 16 收到新的用户 I/O 请求时，驱动器控制器 18 在框 110 处确定该请求所处的段。在框 112 处，递增该段的计数。移到菱形决策框 114，确定得到的计数是否超过计数域能够保存的最大值。例如，如果计数域使用了 16 个位，则该计数域的最大计数是 65535。如果大于该数值，则逻辑移到框 116 以将表中的每项的计数域除以 2。

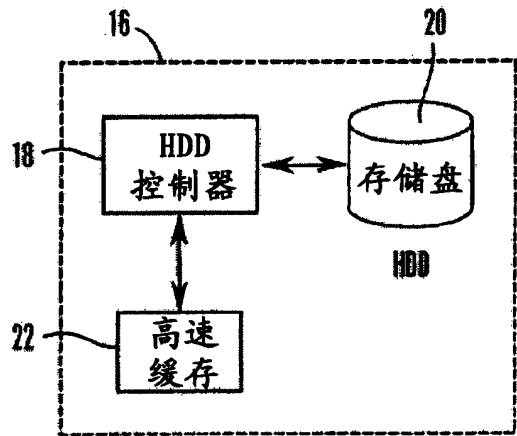
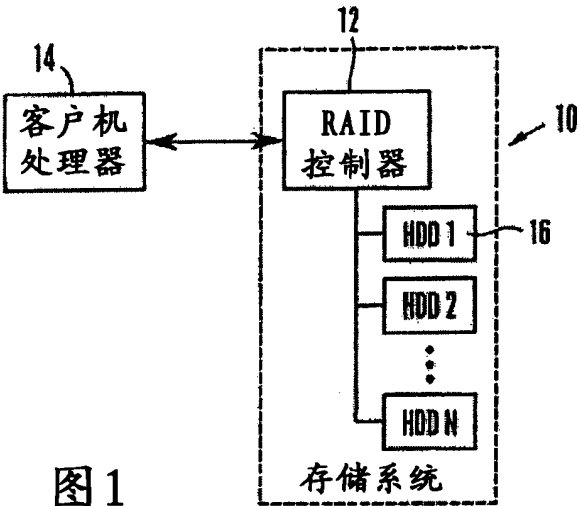
从框 116 继续到菱形决策框 116，或在测试结果为否定的情况下从菱形决策框 114 继续到菱形决策框 116，驱动器控制器 18 确定新计数现在是否大于由与当前段有关的后向指针所指向的表项，即有序链表中次最高表项的计数。如果现在的计数较大，该逻辑移到框 120，在这里将这两项在链表中的位置交换。然后循环返回到菱形决策框 118，并重复执行，直至链表被适当地排序。

最后，图 10 示出的是针对图 8 和 9 的上述表实现，在状态 122 处进入的选择算法。在框 124 中，选择链表中顶部的表项（该项具有最高的计数）。移到菱形决策框 126，确定这段中所有的擦洗单元是否已经擦洗。如果所有的单元已经擦洗，该逻辑移到框 128，其中选择链表的下一段，然后回到菱形框 126。如果在菱形决策框 126 中发现一个段中并不是所有单元已经得到擦洗，则进入框 130 以从该段中选择擦洗单元。

对于使用错误恢复过程（ERP）可恢复的错误，驱动器控制器 18 能够尝试将数据写回到原来的位置，然后读出该数据以确定它是否再次包含错误。如果还包含错误，则表明该物理位置被损坏。能够使用新模式页参数来控制这个选项。如果用户已经在它的模式页中指定了自动重新分配，则控制器 18 就将恢复的数据写到新的位置。当没有指定自动重新分配，但也没有指定在原位置进行重写时，恢复的数据能够存储在保护区域中，可以在随后出现不可恢复的错误时检查该保护

区域以确定丢失的数据是否驻留在保护区域中。

虽然这里所示和详细描述硬盘驱动器的自主数据擦洗的系统和方法完全能够达到本发明的上述目的，但是应当理解，它只是本发明当前提出的优选实施例，因此它只是本发明所广泛考虑的主题的代表，本发明的范围完全涵盖本领域技术人员可以想到的其它实施例，因此，本发明的范围仅受附加的权利要求的限制，其中对单数要件的引用不意味着“一个且唯一一个”，除非专门指出，而是表示“一个或多个”。用一种设备或方法解决本发明试图解决的每个问题是不必要的，因为它被本发明的权利要求所涵盖。而且，在本发明的公布中没有任何要件、部件或方法步骤是仅限于所公开内容的，无论元件、部件或方法步骤是否明确在权利要求中进行了叙述。这里没有任何权利要求要件将按照 35 U.S.C.'112，第六款的规定加以解释，除非该要件明确使用短语“用于...的装置”加以表述，或者在方法权利要求中，该要件被表述为“步骤”而不是“动作”。这里没有明确的定义，权利要求中的术语具有所有普通的和习惯的含义，其与本说明书和文件的历史是一致的。



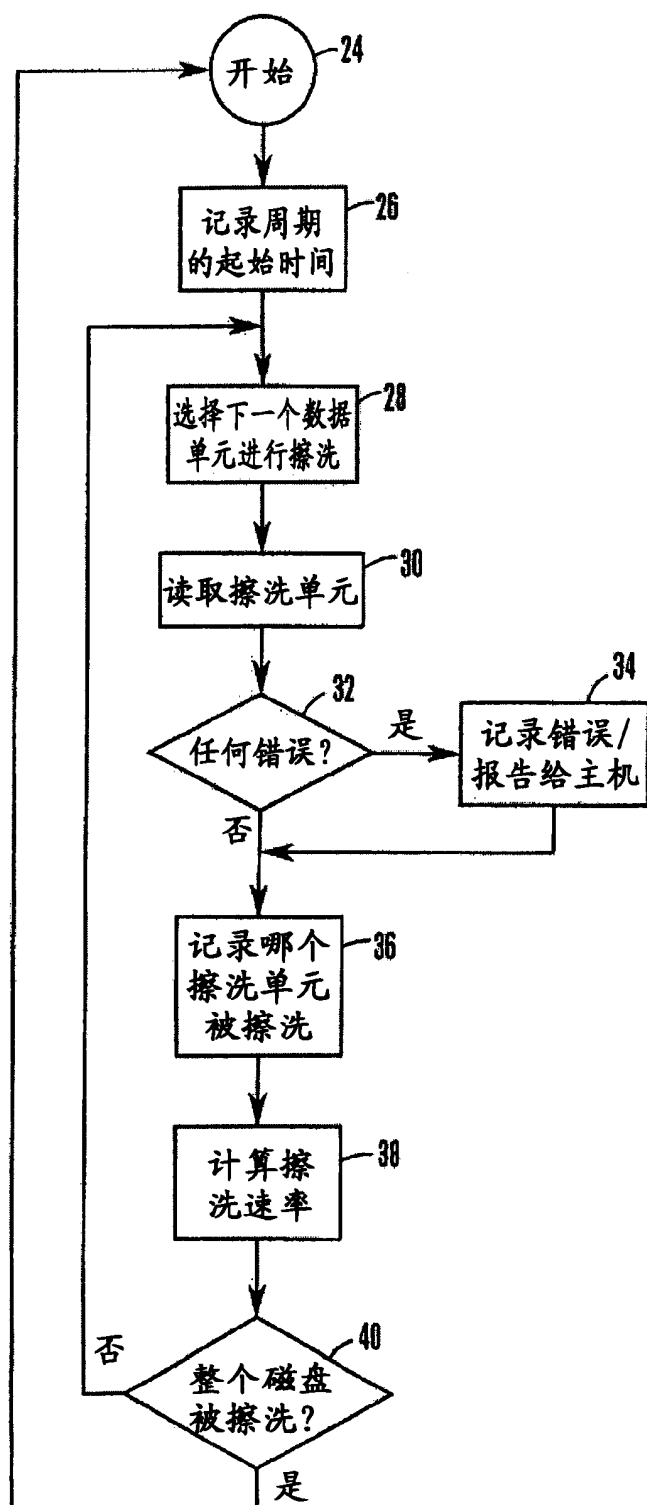


图3

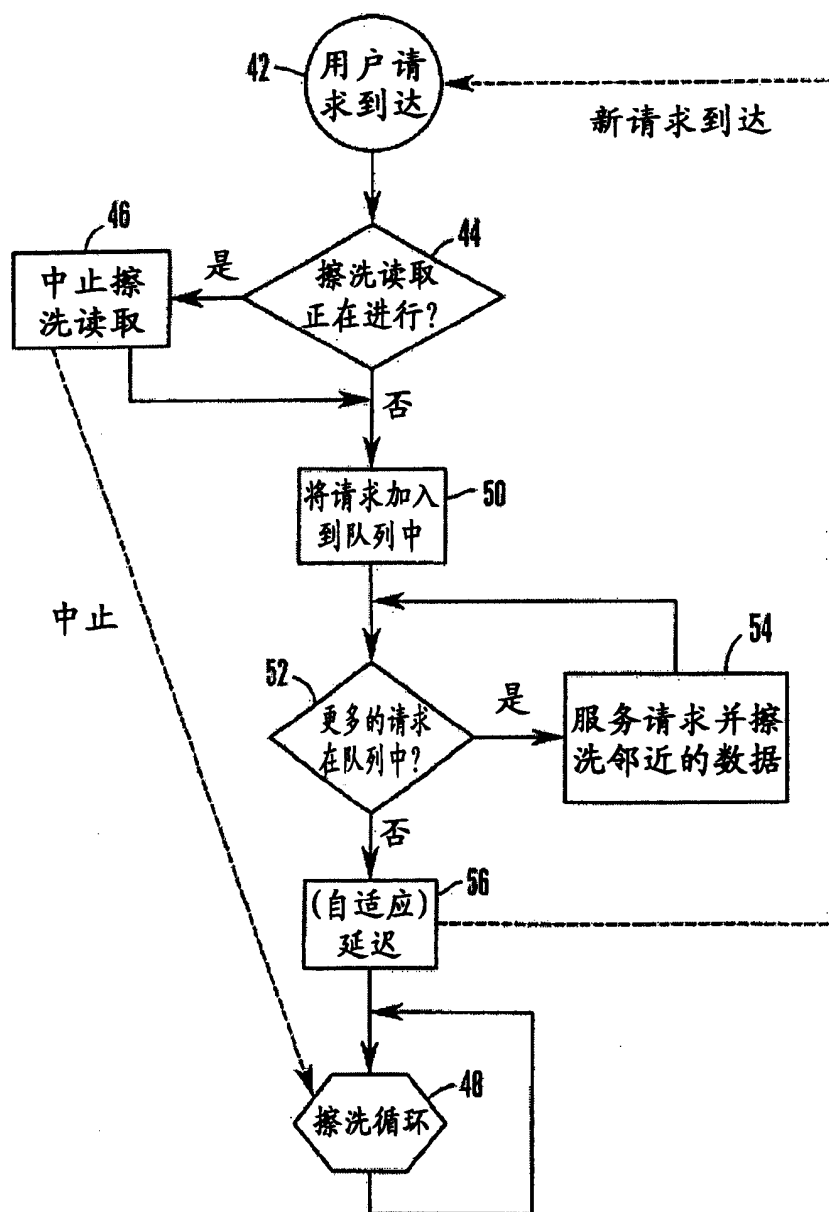


图4

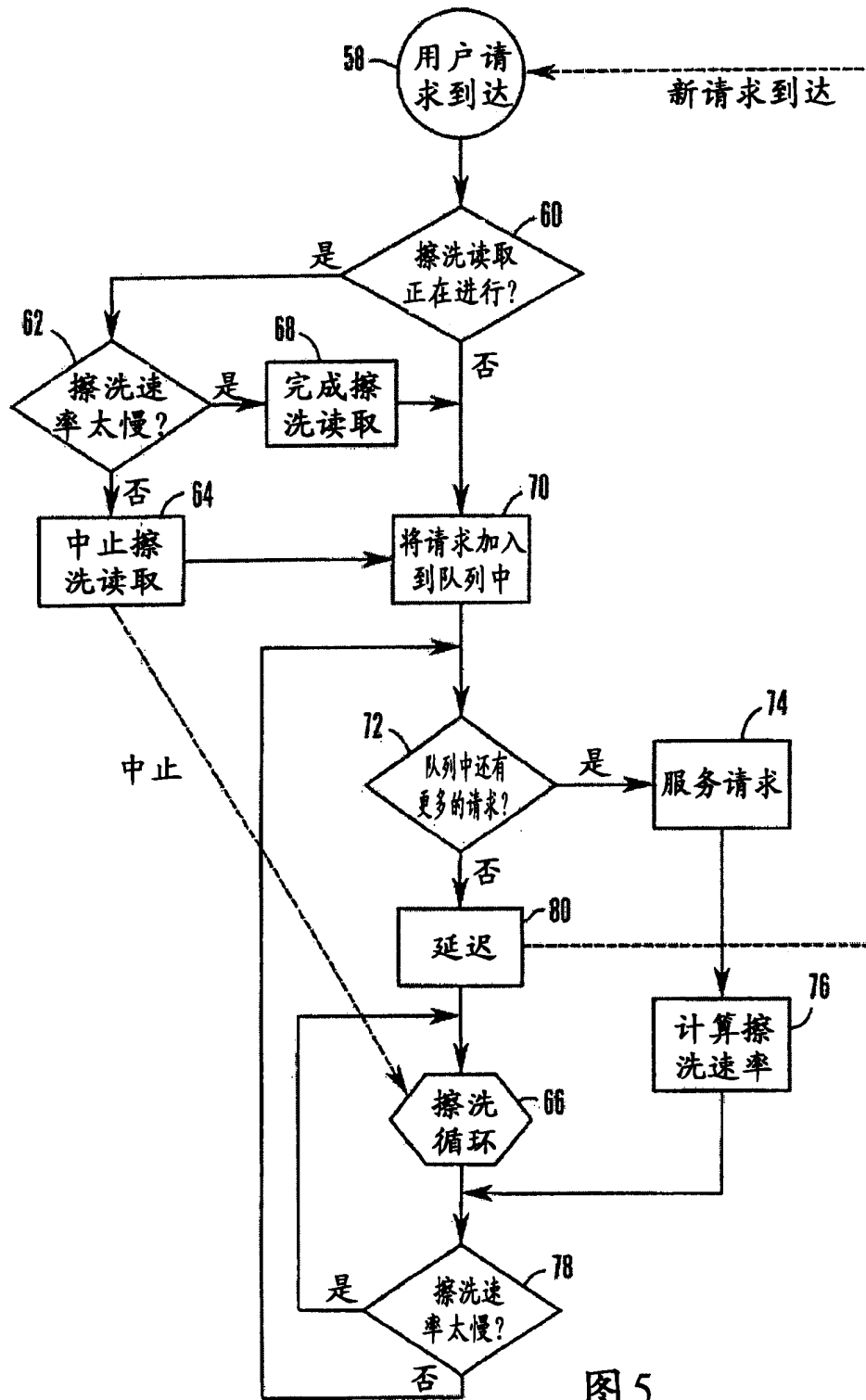


图5

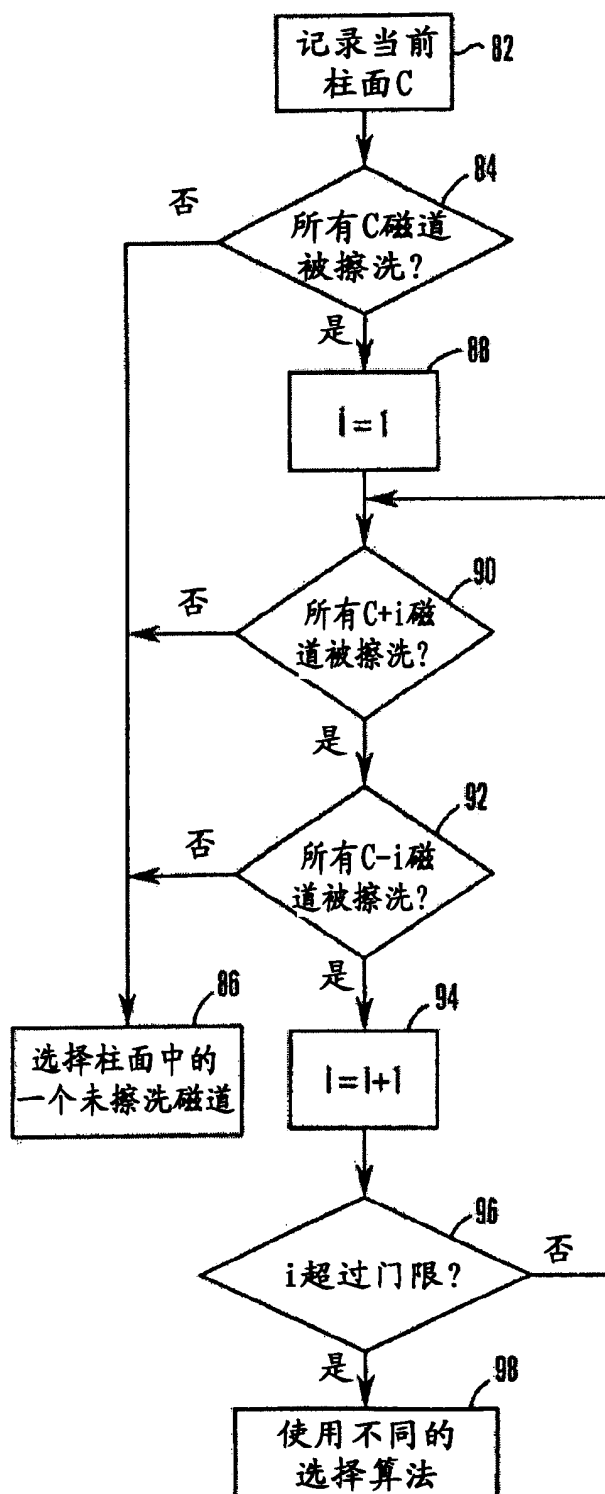


图6

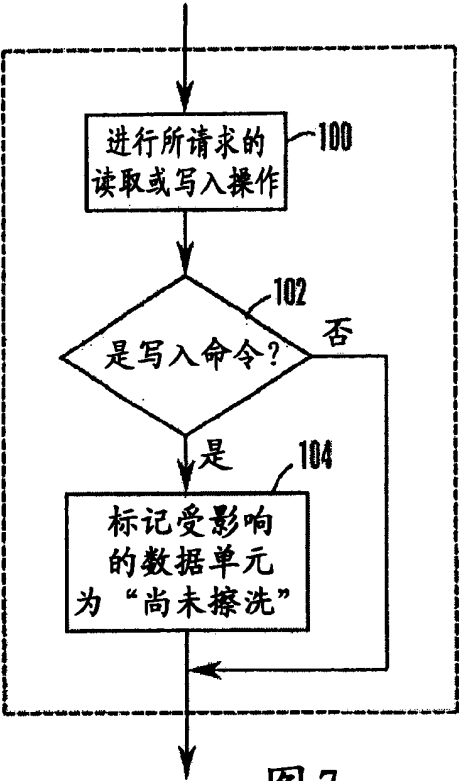


图 7

	计数	前向指针	后向指针
第1段			
第2段			
第3段			
⋮			
第N段			

106 ↗

图 8

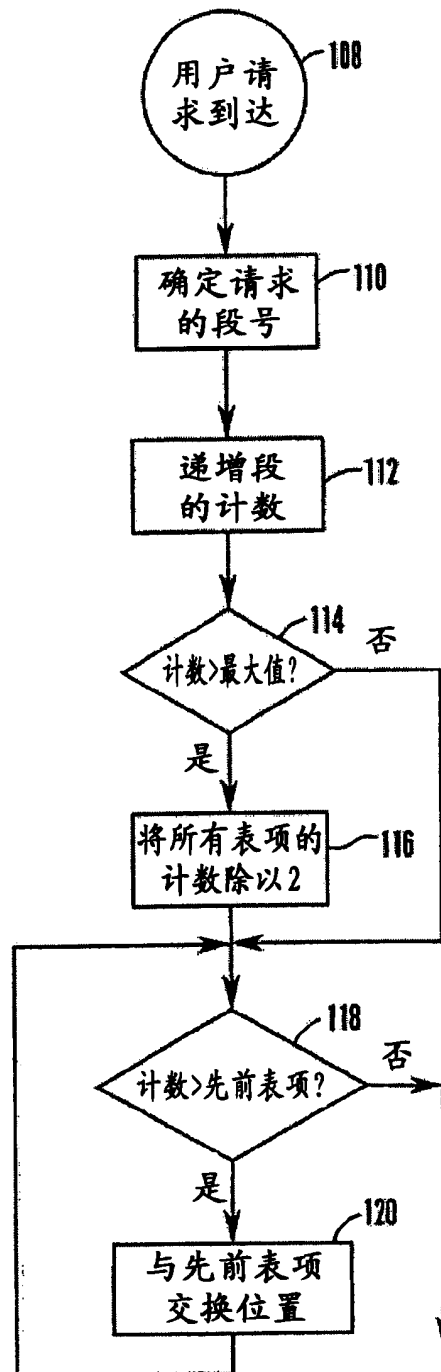


图 9

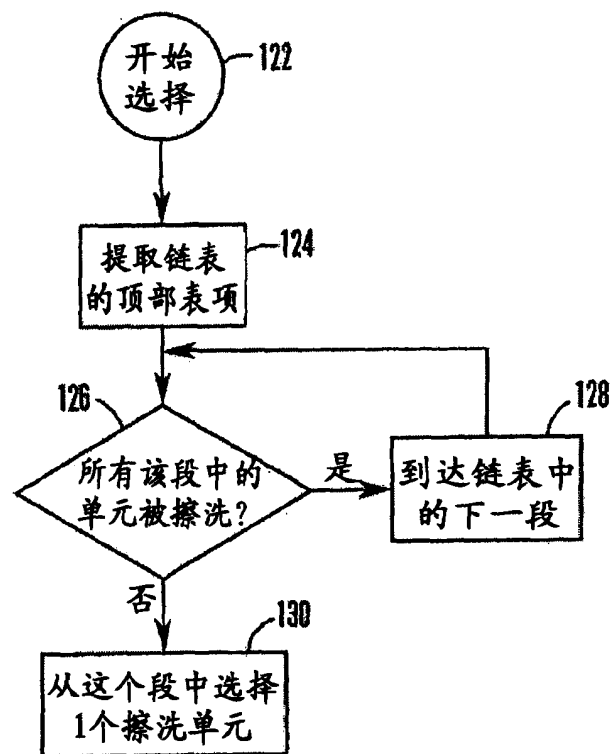


图10