



(12) 发明专利申请

(10) 申请公布号 CN 102037519 A

(43) 申请公布日 2011. 04. 27

(21) 申请号 200980117908. 8

(74) 专利代理机构 北京市柳沈律师事务所

(22) 申请日 2009. 02. 19

11105

代理人 黄小临

(30) 优先权数据

12/051, 462 2008. 03. 19 US

(51) Int. Cl.

G11C 16/10(2006. 01)

G11C 7/10(2006. 01)

G11C 11/56(2006. 01)

(85) PCT申请进入国家阶段日

2010. 11. 17

(86) PCT申请的申请数据

PCT/US2009/034573 2009. 02. 19

(87) PCT申请的公布数据

W02009/117204 EN 2009. 09. 24

(71) 申请人 桑迪士克公司

地址 美国加利福尼亚州

(72) 发明人 李艳 戈保凌

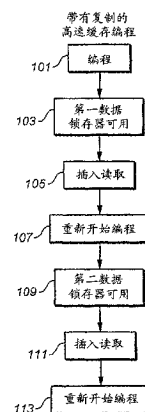
权利要求书 2 页 说明书 5 页 附图 4 页

(54) 发明名称

带有动态数据锁存器要求的高速缓存操作中的适应性算法

(57) 摘要

一种非易失存储器可以使用存储在相应组的数据锁存器中的第一数据集对指定组的一个或多个寻址的页进行第一操作(例如诸如写入),并且还接收对于也使用带有第二数据集的这些相应的数据锁存器中的一些的第二操作(例如诸如读取)的请求。在所述第一操作期间,当相应的每组的至少一个数据锁存器变得可用于所述第二操作时,所述存储器确定是否存在足够数量的相应组的数据锁存器来在所述第一操作期间进行所述第二操作;如果不存在,则延迟第二操作。当足够数量的锁存器变得可用时,所述存储器锁随后可以在所述第一操作期间进行所述第二操作;并且如果响应于确定是否存在足够数量的相应组的数据锁存器来进行所述第二操作,确定存在足够的数量,则在所述第一操作期间进行所述第二操作。



1. 一种操作具有可寻址的页的存储器单元的非易失性存储器的方法,包括:

为一个寻址的页的每个存储器单元提供具有用于锁存预定数量的位的容量的一组相应的数据锁存器;

使用存储在相应组的数据锁存器中的第一数据集对指定组的一个或多个寻址的页进行第一操作;

接收对于使用带有与对存储器阵列的一个或多个随后的存取操作相关的数据的相应组的数据锁存器的第二操作的请求;

在所述第一操作期间,确定每组数据锁存器中的至少一个锁存器可用于所述第二操作;

随后确定是否存在足够数量的相应组的数据锁存器来在所述第一操作期间进行所述第二操作;以及

响应于确定没有足够数量的相应组的数据锁存器来进行第二操作,延迟所述第二操作。

2. 如权利要求1所述的方法,还包括:

随后当足够数量的锁存器变得可用时,在所述第一操作期间进行所述第二操作。

3. 如权利要求1所述的方法,还包括:

响应于确定存在足够数量的相应组的数据锁存器来进行所述第二操作,在所述第一操作期间进行所述第二操作。

4. 如权利要求1所述的方法,其中所述第一操作是擦除操作的软编程阶段。

5. 如权利要求1所述的方法,其中所述第一操作是编程操作。

6. 如权利要求5所述的方法,其中所述第二操作是读取操作。

7. 如权利要求6所述的方法,其中所述读取操作是用在所述编程操作中的先行读取操作。

8. 如权利要求1所述的方法,其中所述第一操作是具有交替的编程和验证阶段的写操作,并且第一组数据是要被写入第一组存储器单元中的数据。

9. 如权利要求1所述的方法,其中所述存储器单元是存储N位数据的多级别存储器单元,其中N大于1,以及其中所述组的数据锁存器中的每组包括N个数据锁存器,并且所述第一数据集是N位数据。

10. 如权利要求1所述的方法,其中在所述第一操作的执行期间接收对于所述第二操作的请求。

11. 如权利要求1所述的方法,其中对于所述第二操作的请求在开始所述第一操作的执行之前。

12. 如权利要求1所述的方法,其中基于准备好/忙碌信号确定每组数据锁存器中的至少一个数据锁存器可用于所述第二操作。

13. 一种非易失性存储器,包括:

可寻址的页的存储器单元;以及

多个数据锁存器,其中一个可寻址的页的每个存储器单元具有一组相应的数据锁存器,该组相应的数据锁存器具有用于锁存预定数量的位的容量,

其中在使用存储在相应组的数据锁存器中的第一数据集对指定组的一个或多个寻址

的页的第一操作期间,存储器可以确定每组数据锁存器中的至少一个锁存器可用于使用带有与对存储器阵列的一个或多个随后的存取操作相关的数据的相应组的数据锁存器之一的所请求的第二操作,随后确定是否存在足够数量的相应组的数据锁存器来在所述第一操作期间进行所述第二操作,以及响应于确定没有足够数量的相应组的数据锁存器来进行所述第二操作,延迟所述第二操作。

14. 如权利要求 13 所述的非易失性存储器,其中当足够数量的锁存器变得可用时,所述存储器随后在所述第一操作期间进行所述第二操作。

15. 如权利要求 13 所述的非易失性存储器,其中,响应于确定存在足够数量的相应组的数据锁存器来进行所述第二操作,所述存储器在所述第一操作期间进行所述第二操作。

16. 如权利要求 13 所述的非易失性存储器,其中所述第一操作是擦除操作的软编程阶段。

17. 如权利要求 13 所述的非易失性存储器,其中所述第一操作是编程操作。

18. 如权利要求 17 所述的非易失性存储器,其中所述第二操作是读取操作。

19. 如权利要求 18 所述的非易失性存储器,其中所述读取操作是用在所述编程操作中的先行读取操作。

20. 如权利要求 13 所述的非易失性存储器,其中所述第一操作是具有交替的便车和验证阶段的写操作,并且第一数据集是要被写入第一组存储器单元中的数据。

21. 如权利要求 13 所述的非易失性存储器,其中所述存储器单元是存储 N 位数据的多级别存储器单元,其中 N 大于 1,以及其中所述组的数据锁存器中的每组包括 N 个数据锁存器,并且所述第一数据集是 N 位数据。

22. 如权利要求 13 所述的非易失性存储器,其中在所述第一操作的执行期间接收对于所述第二操作的请求。

23. 如权利要求 13 所述的非易失性存储器,其中对于所述第二操作的请求在开始所述第一操作的执行之前。

24. 如权利要求 13 所述的非易失性存储器,其中每组数据锁存器中的至少一个数据锁存器可用于所述第二操作的确定基于准备好 / 忙碌信号。

25. 如权利要求 13 所述的非易失性存储器,还包括:

状态机,其中所述状态机进行每组数据锁存器中的至少一个锁存器可用于所请求的第二操作的所述确定、是否存在足够数量的相应组的数据锁存器来在所述第一操作期间进行所述第二操作的所述确定、以及所述第二操作的延迟。

带有动态数据锁存器要求的高速缓存操作中的适应性算法

技术领域

[0001] 本发明一般涉及非易失性半导体存储器,诸如电可擦除可编程只读存储器(EEPROM)和快闪 EEPROM,具体地涉及基于允许重叠存储器操作的共享锁存器结构的高速缓存操作。

背景技术

[0002] 在诸如闪存的非易失性存储器的设计中,存在通过增加这些存储器的存储密度、提高其性能、并降低功耗来改善其存储的连续处理。这些要求之一的改善将经常对其他要求之一带来负面影响。例如,为了改进存储密度,可以使用具有每单元多级别(level)的闪存来替代二进制芯片;但是,在多状态存储器中,比如在各状态间的容许量变得更严格时写入数据的情况下,操作的速度会更慢。从而,具有多级别单元的存储器的性能级别具有很大范围来改善。

[0003] 这些和相关问题与另外的背景信息在美国专利申请公开号 US-2006-0221704A1 和 US-2007-0109867-A1 的背景技术部分中给出。以下美国专利申请公开号也提供了另外的背景信息:US 2006-0233023-A1;US2006-0233021-A1;US 2006-0221696-A1;US 2006-0233010-A1;US2006-0239080-A1;以及 US 2007-0002626-A1。如下所述,所有这些通过参考全部并于此。

[0004] 因此,存在对于高性能和高容量的非易失性存储器的普遍需要。具体地,存在对于一种紧凑和有效的、带有增强的读取和编程性能的紧凑非易失性存储器的需要,该非易失性存储器具有改善的处理器,还高度通用于处理读/写电路中的数据。

发明内容

[0005] 本发明给出了非易失性存储器和相应的操作方法,其中存储器具有可寻址的页,并且为一个寻址的页的每个存储器单元提供了可以锁存预定数量的位的一组相应的数据锁存器。该存储器可以使用存储在相应组的数据锁存器中的第一数据集对指定组的一个或多个寻址的页进行第一操作(例如诸如写入),并且还接收对于也使用带有第二数据集的这些相应的数据锁存器中的一些的第二操作(例如诸如读取)的请求。在所述第一操作期间,当相应的每组的至少一个数据锁存器变得可用于所述第二操作时,确定是否存在足够数量的相应组的数据锁存器来在所述第一操作期间进行所述第二操作;如果不存在,则延迟第二操作。在另外的方面,当足够数量的锁存器变得可用时,所述存储器随后在所述第一操作期间进行所述第二操作;并且如果响应于确定是否存在足够数量的相应组的数据锁存器来进行所述第二操作,确定存在足够的数量,则在所述第一操作期间进行所述第二操作。

[0006] 本发明的各个方面、优点、特征和实施例被包括在其示例例子的以下描述中,该描述应与附图来结合。在此所参考的所有专利、专利申请、文章、其他公开物、文档和事物为了所有目的通过以其全部内容的此参考而被并于此。对于在任何并入的公开物、文档或事物与本申请之间的术语的定义和使用中的任何不一致或矛盾之处,应以本申请的定义和使用

为准。

附图说明

- [0007] 图 1 图示了在高速缓存编程操作内插入读。
- [0008] 图 2 图示了在高速缓存擦除操作内插入读。
- [0009] 图 3 示出了对于页以及一些相应的先行读取的具体编程顺序。
- [0010] 图 4 图示了当没有足够的锁存器可用时在高速缓存编程操作内插入读。
- [0011] 图 5 图示了当没有足够的锁存器可用时在高速缓存擦除操作内插入读。
- [0012] 图 6 是当没有足够的锁存器可用时在高速缓存编程操作内插入读的另一例子。
- [0013] 图 7 示意性示出了各种高速缓存点如何出现。
- [0014] 图 8 是对于适应性算法的一个基本实施例的流程图。

具体实施方式

[0015] 具有每单元多级别的非易失性存储器,诸如 NAND 闪存,被用于替换二进制芯片,存在对于改进性能的普遍赞同的需要。一组性能增强的方法是基于利用复杂的高速缓存算法来同时进行并行计算。一些这样的增强的例子在美国专利申请公开号 US-2006-0221704-A1 和 US-2007-0109867-A1 中给出。还参考以下美国专利申请公开号,其提供了关于许多方面的另外的细节:US2006-0233023-A1;US 2006-0233021-A1;US 2006-0221696-A1;US2006-0233010-A1;US 2006-0239080-A1;以及 US 2007-0002626-A1。所有这些参考与其中所列的参考一起通过引用全部并于此,如上所述。以下描述技术中的一些来提供随后开发的背景(context),但是为了进一步的细节请参考这些在先申请。在这点上,在某种程度上可以认为以下是那儿所描述的内容的进一步研究。

[0016] 在前述专利文件中,具体参考描述数据锁存器的使用 and 如何将其用于在诸如程序的第二较长操作中插入诸如读的一个操作的部分。尽管该先前讨论的大部分主要针对每单元 2 位的实施例给出,但是在此的大部分讨论与对每单元 3 位或更多位的情况最适当的方面有关。

[0017] 先行读取是使用取决于下一个字(word)的数据的校正读取的算法。带有 LA(“先行”)校正的读取主要检查被编程到临近字线上的单元中的存储器状态,并校正其对当前字线上的正被读取的存储器单元具有的任何扰动影响。如果已经根据在所列参考中描述的优选编程方案对页进行了编程,则邻近字线将来自紧位于当前字线以上的字线。LA 校正方案会需要将在当前页之前被读取的邻近字线上的数据。

[0018] 进行 LA 读取所需的数据锁存器的数量将取决于需要的校正。在某些情况下,这将是 1 位校正,其他将使用 2 位或 3 位校正。每页所需的校正将取决于该页和邻近页经历过的程序次序。在某些情况下,一页将仅需要 1 位校正,而另一页将可能使用 2 位校正。这些不同的校正级别将使用不同数量的数据锁存器来应对 LA 读取。当进行混合的高速缓存操作时,比如在用于复制功能的高速缓存程序中插入读或者在高速缓存擦除操作中插入读时,优选在高速缓存算法中说明该读的数据锁存器需要的变化。在用户(例如控制器或主机)发出地址以前也不知道数据锁存器需要。为了更好地应对这些复杂性,引入了在以下称为适应性算法的新高速缓存算法。

[0019] 为了提供背景,以下将在使用针对数据多状态的“先行”读取 (“LA”) 和“较低中间” (“LM”) 编码的实施例中描述这些技术。这样的安排在以上所列的美国专利申请中更详细地给出,比如 US 2006-0239080-A1,在标题为“Cache Read Algorithm for LM code with LA Correction” (“针对带有 LA 校正的 LM 码的高速缓存读算法”) 的部分中开始于 [0295] 段。简要地说,如上所述,实现用于高速缓存读取数据的方案,使得即使对于其校正取决于来自相邻物理页或字线的数据的读操作,在正从存储器核心感测当前页时,数据锁存器和 I/O 总线也有效率地用于轮出 (toggle out) 先前读取的页。一个优选的读操作是“先行” (“LA”) 读取,并且对于存储器状态的优选编码是“较低中间” (“LM”) 编码。当邻近字线上的数据的先决 (prerequisite) 读取必需领先于对于当前字线上的当前页的读取时,该先决读取与 I/O 存取一起在用于读取先前页的周期中抢先进行,使得可以在先前读取的页忙于 I/O 存取的同时进行当前读取。在 2005 年 4 月 5 日提交的题为“Read Operations for Non-Volatile Storage that Includes Compensation for Coupling (对于包括对耦合的补偿的非易失性存储器的读取操作)”的美国专利申请 No. 11/099,049 中已经公开了 LA 读取方案,其全部内容通过参考并于此。带有 LA (“先行”) 校正的读取主要检查被编程到邻近字线上的存储器单元中的存储器状态,并校正其对当前字线上正被读取的存储器单元具有的任何扰动影响。如果已经根据上述优选编程方案对页进行了编程,则邻近字线将来自就在当前字线以上的字线。LA 校正方案将需要在当前页之前读取邻近字线上的数据。

[0020] 返回在此在该示例实施例中给出的进一步研究,当数据锁存器需要与 LM 标志相关时,则可以执行用户命令,并内部确定没有足够的数据锁存器来完成该命令的执行。适应性算法记下用户命令,等待足够的数据锁存器可用,然后随着在操作的过程期间数据锁存器变得可用而执行命令。

[0021] 图 1 和图 2 分别给出了将读取插入高速缓存编程和高速缓存擦除操作中的例子。在图 1 的带有复制操作的高速缓存编程中,处理以开始于始于 101 处的编程操作。这继续直到在 103 处在相应堆栈的数据锁存器中第一锁存器被释放 (free up),在 US-2006-0221704-A1 和 US-2007-0109867-A1 中更详细描述了处理。在此点时,可以在 105 将读取插入编程操作中,在这之后在 107 写操作继续。在编程的过程中,在 109,第二锁存器再次变为可用。该第二锁存器可以是与 103 处相同的锁存器,或者是同一堆栈中的不同锁存器。再次,因为该处理一般实现在页级别,则通常的实施例会对于页中的每个单元需要相应锁存器。在任何情况下,然后在 111 再次插入读取,在这之后在 113,编程操作继续。

[0022] 图 2 是对于带有读的高速缓存擦除的相应安排。在此包括软编程操作的擦除处理开始于 210。在 203,数据锁存器可用于插入的操作。因为软编程操作可被认为是对于 N 状态存储器单元的一类二进制编程操作,因此这通常将导致存在 (N-1) 个可用的锁存器。然后可以在 205 插入读操作,在这之后软编程阶段可以继续。

[0023] 当插入读操作 (图 1 中的 105 或 111,图 2 中的 205) 是先行读操作时,数据锁存器需要将取决于使用的校正量。在一个算法中,为了对字线 n (W/In) 进行先行读取,1 位校正将使用 2 个数据锁存器,一个数据锁存器用于 $WIn+1$ 数据,一个用于 WIn 数据的一页。类似地,关于 2 位校正,使用 3 个数据锁存器 (两个用于 $WIn+1$,一个用于 WIn 数据的一页),并且关于 3 位校正,使用 4 个数据锁存器 (三个用于 $WIn+1$ 数据,一个用于 WIn 数据的一页)。在美国专利申请号 11/618,569 和 11/618,578 中描述对于所有的 LA 校正将仅需要 2 个

锁存器的替换实施例。

[0024] 接下来,将考虑当与 LA 读取结合时包括较低中间 (“LM”) 页顺序和相应的锁存器需要,这是在 2008 年 3 月 19 日提交的 YanLi 的题为“Different Combinations of Wordline Order and Look-Ahead read to improve Non-Volatile Memory performance (用于改进非易失性存储器性能的仔细按顺序和先行读取的组合)”的美国专利是其中更详细地开发的安排。取每单元 3 位的情况,页可以被安排为使得较低的和中间的连续并且一起被编程,但是其中,将按较高页编程将趋向于消除中间页到中间页 WL-WL 耦合效应的方式跳过较高页。在下一字线的中间页编程后,对较高页编程。

[0025] 该处理示出在图 3 中,其中页 0 和页 1 作为较低和中间页一起在第一字线 (WL0) 上被编程,随后页 2 和页 3 一起作为较低和中间页在下一字线 (WL1) 上被编程。接下来,处理退后一页 (到 WL0) 并编程较高页 (页 4),然后向前跳过两个字线 (到 WL2) 并编程较低和中间页 (页 5 和页 6)。对于较高页的该退后以及对于两个较低页的向前跳对于剩余的数据集继续,其中 a、b 和 c 是十六进制符号的页 10、11、12。以此方式,将通过较高页编程有效校正 WL-WL 以及 BL-BL 耦合效应的大部分。在较高页编程期间,将利用 LA 读取读入较低和中间页,以从存储器单元校正地读入数据。在此页安排中,较高页读取将仅需 1 位校正,因为较高页仅与下一 WL 较高页编程耦合。另一方面,中间页读取将使用 2 位用于 LA 读取校正,因为较低和中间页可以耦合先前字线中间页电压阈值。

[0026] 用于应对动态数据锁存器需要的适应性算法

[0027] 如从刚刚给出的例子可见,对于这种高速缓存操作的数据锁存器需要根据情况而变化。在此给出的适应性算法将其纳入考虑。

[0028] 回到针对复制情况的带有插入的读的高速缓存编程的例子,其示出在图 4 中,其中编程操作开始于 401。随着操作继续,在某点,用于页中的每个存储器单元的一对锁存器在 403 被释放。在此点时,可以如在 405 所示插入读。这可以是状态机已经保持的、等待锁存器打开的读,或者是在锁存器打开后到来的读请求,在该情况下,编程将继续直到读到来。在任一情况下,一旦插入了读,就可以确定 2 个锁存器不足够完成此读取。读命令可以在 2 个数据锁存器可用时被输入,但是可能仅通过假设上部页读取仅使用 1 位 LA 读取而执行;但是,如果页未使其上部页被编程,则将需要 2 位用于 LA 读取。通常,锁存器将已经被填充了编程数据,但是一旦确定需要进一步的锁存器,读数据就将被当作无效的。在此情况下,不能完成用户 (例如:控制器) 发出的读命令,直到更多的数据锁存器可用为止。在此例子中,不能执行读取,直到 3 个数据锁存器可用为止。一旦存储器确定不存在足够的锁存器并且不能完成读命令,命令就被保持,直到所需的锁存器可用,如箭头所示。

[0029] 同时,写处理在 407 继续,直到在 409,另一锁存器打开。然后在 411 再次插入读并完成,在此之后在 413 写继续。应该注意,这仅仅是可以具有多个变化的处理的一个一般例子;例如,该情况假设在步骤 413 之前未完成写。

[0030] 类似的情况可以发生在插入读的高速缓存擦除中,如图 5 所示。带有软编程的擦除开始于 501。对于每单元 $N = 3$ 位的数据,在软编程中存在 2 个数据锁存器可用 (在 503),其能够应对 1 位的 LA 读取;但是如果需要 2 位 LA 读取,则需要在整个软编程完成后执行该读取。从而,在 505 插入读取,并且如果没有所需数量的锁存器来成功完成该读取,则其仅是被保持,软编程在 507 恢复并在 509 完成,在其之后在 511 重新插入读取。

[0031] 图6图示了另一情况,其中适应性高速缓存操作对于管理可以是复杂的。图6示出了其中在601-607执行第一读取而不需要比可用锁存器更多的锁存器,非常像以上参考的专利公开。为了ECC将检查该读入页,然后该读入页准备好以被编程到另一位置。在较高页编程期间(607的部分),可以在611插入第二读取。如果由于足够的数据锁存器不可用而不能执行第二读取,则第二读取不能立即执行,并将等待直到较高页完成其编程(613)。

[0032] 在较高页编程的结束时,仍然处于数据锁存器中的第一读取数据(还未被编程)将被传送到合适的(right)位置。在完成较高页编程后,可以执行第二读取命令。一旦需要再次开始编程(615),就可以再次执行未完成的读取(617),非常像关于图4所述。

[0033] 针对适应性算法的一般管理

[0034] 一般的适应性高速缓存操作算法可以由图7和图8图示。在高速缓存操作中,存在可以插入另一操作的多个高速缓存点。图7概念性地示出此点,其中将时间取为行进到右侧,并且该图在正进行的写命令的操作中在某个或多或少任意的点进行截取(pick up)。各种可用的高速缓存点如在701、703、.....713所示,其中当涂7开始时在处理中的写操作结束于731,在其之后发生下一写入。

[0035] 图8是一个示例实施例的流程图。将发出对于操作的命令,并且在高速缓存点(例如由准备好/繁忙信号指示)输入该命令。在用户命令发出后,在803状态机将检查是否存在足够的数据锁存器来执行此命令。如果存在足够的数据锁存器(来自805的是),则在807可以立即执行用户命令,然后返回到801。

[0036] 如果没有足够的数据锁存器可用(来自805的否),或者先前排队的高速缓存仍在进行中,则将重新开始旧的操作(809),同时追踪高速缓存指针以得到下一可用高速缓存点。在该下一高速缓存点,将基于2个因素再次评估先前的用户命令的执行(811):1)进行中的高速缓存队列;2)数据锁存器可用性。饮弹该命令处于队列的头部并且存在足够的锁存器,则可以在807执行该命令。在所有的高速缓存操作中,需要在FIFO管线中保存地址和命令。

[0037] 尽管已经关于某些实施例描述了本发明的各个方面,但是将理解,本发明享有在所附权利要求的全部范围内的保护。

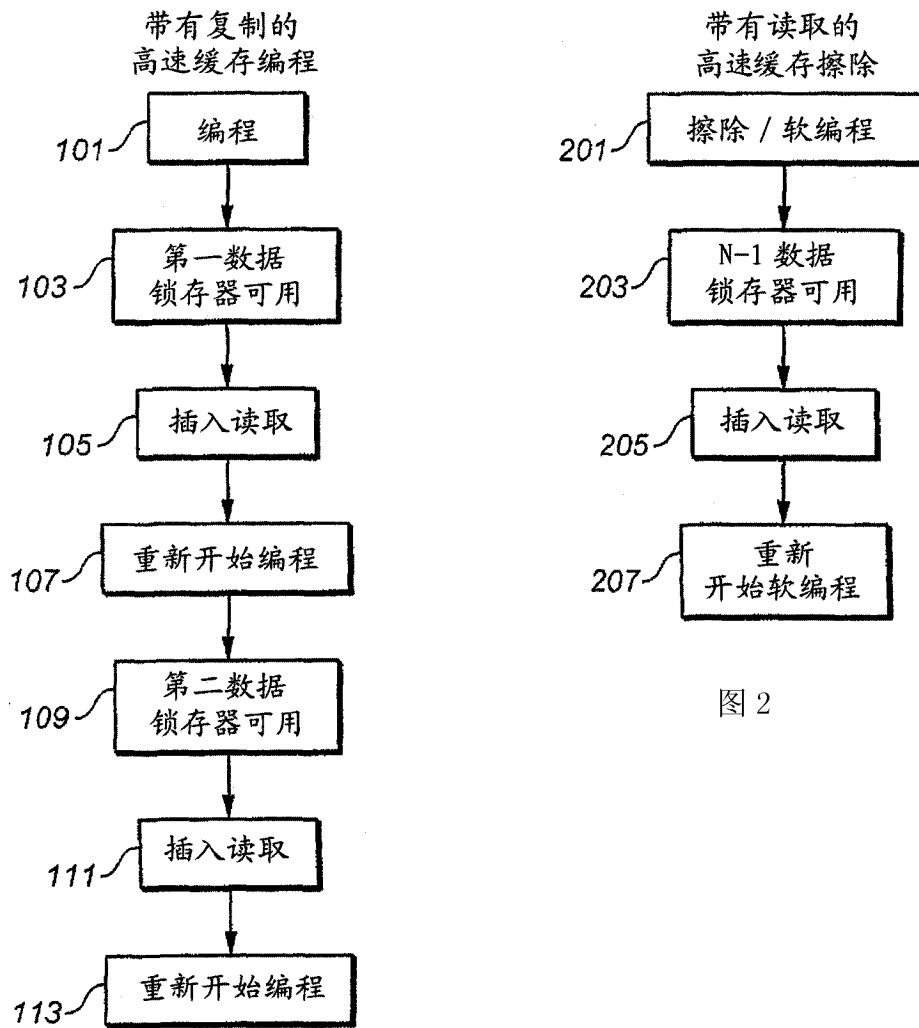


图 2

图 1

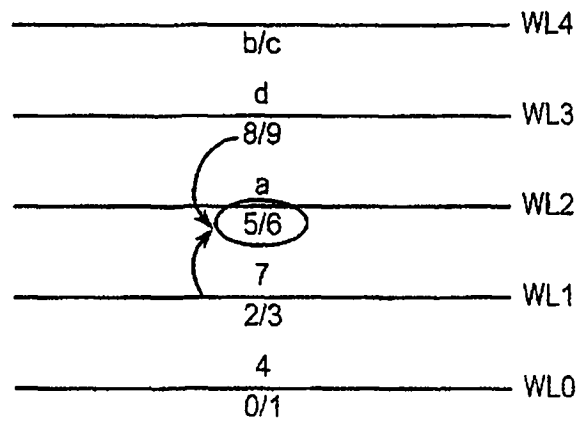


图 3

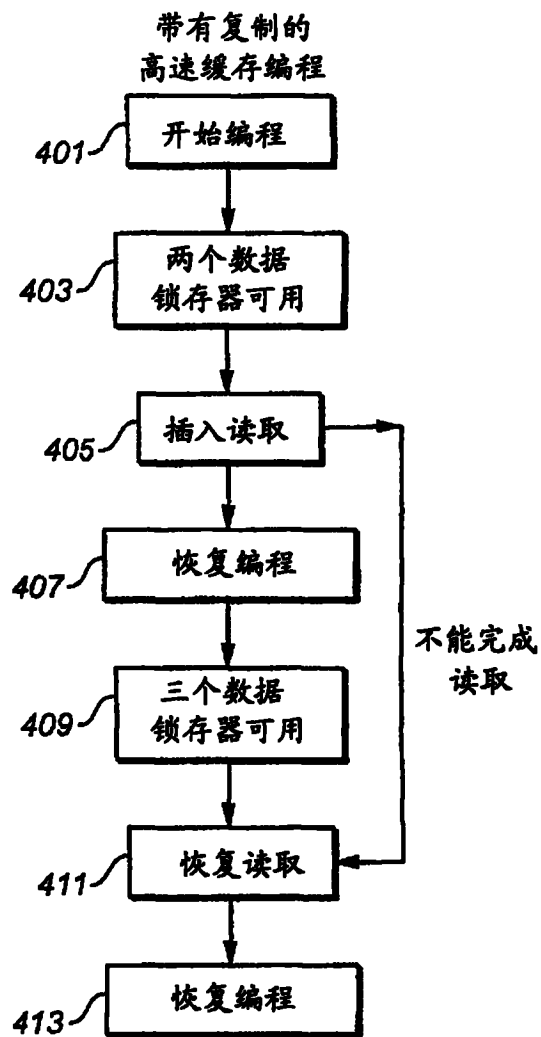


图 4

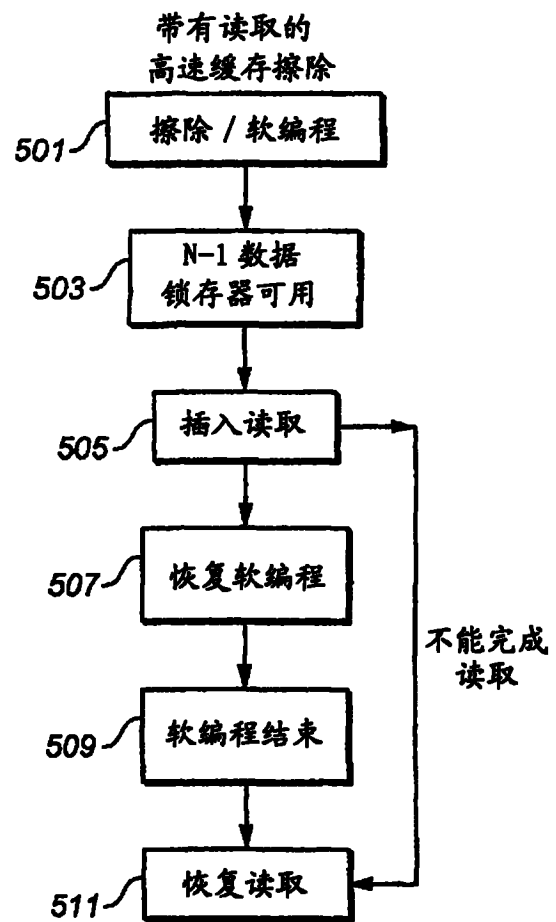


图 5

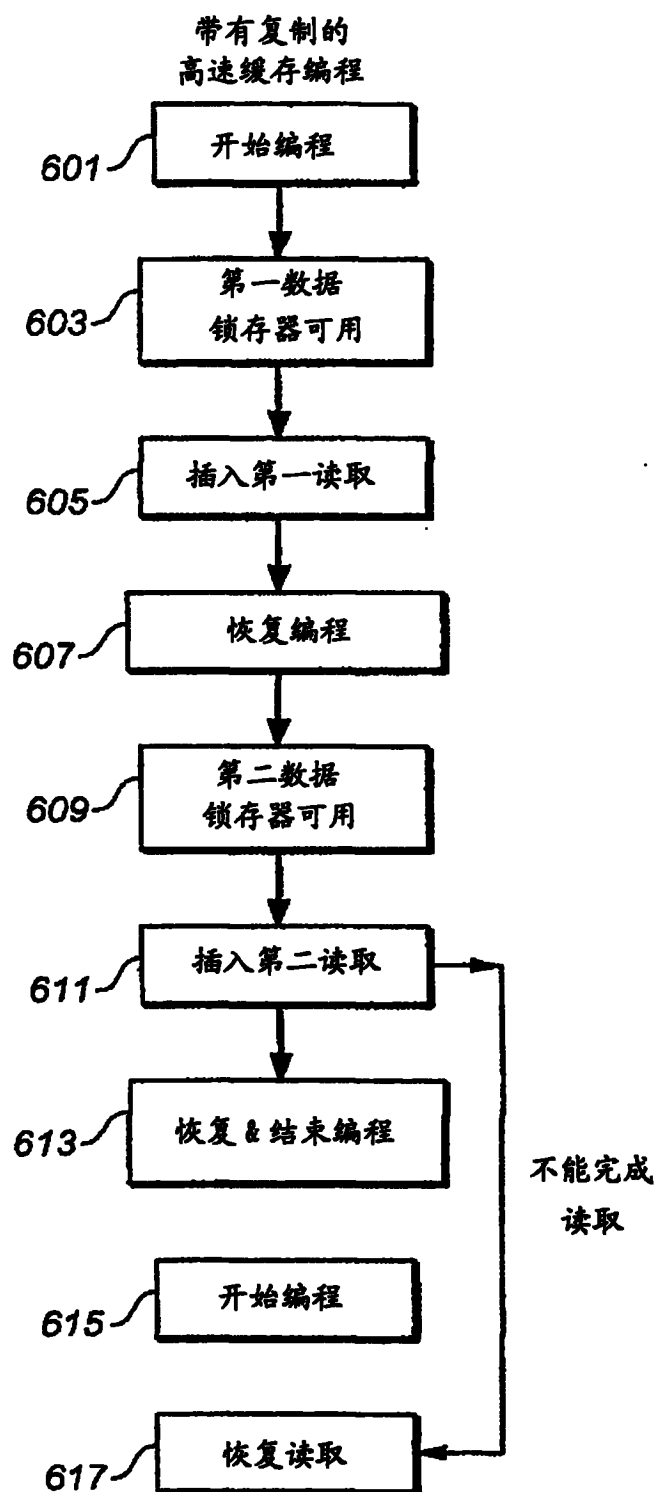


图 6

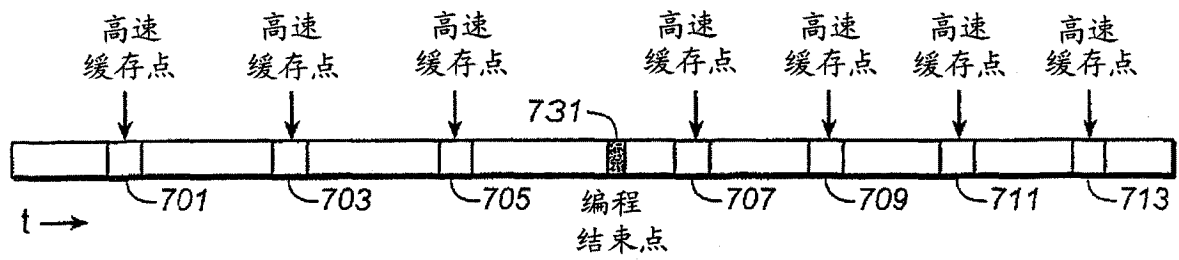


图 7

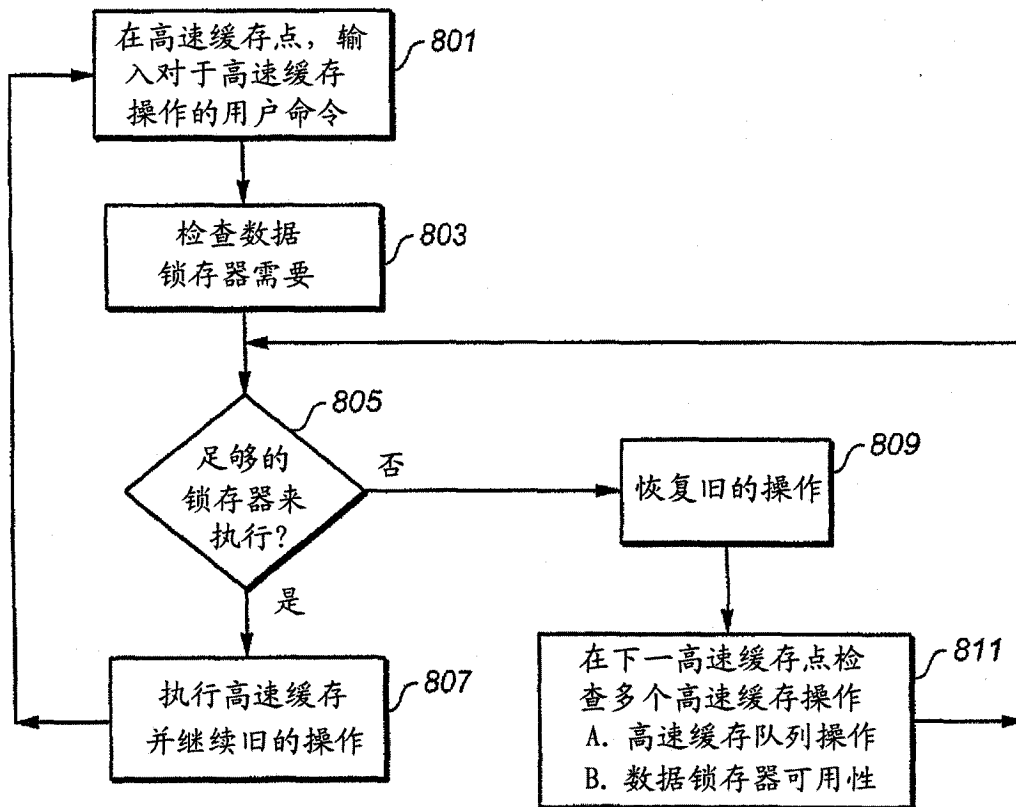


图 8