



(12) 发明专利

(10) 授权公告号 CN 101438353 B

(45) 授权公告日 2012. 10. 03

(21) 申请号 200780016198. 0

(22) 申请日 2007. 05. 02

(30) 优先权数据

11/381, 994 2006. 05. 05 US

11/381, 997 2006. 05. 05 US

(85) PCT申请进入国家阶段日

2008. 11. 04

(86) PCT申请的申请数据

PCT/US2007/068069 2007. 05. 02

(87) PCT申请的公布数据

W02007/131062 EN 2007. 11. 15

(73) 专利权人 桑迪士克股份有限公司

地址 美国加利福尼亚州

(72) 发明人 李彦

(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

(51) Int. Cl.

G11C 16/26(2006. 01)

G11C 7/22(2006. 01)

G11C 11/56(2006. 01)

(56) 对比文件

CN 1480950 A, 2004. 03. 10, 全文.

US 2002/0126531 A1, 2002. 09. 12, 说明书第 142-172, 270 段、附图 1-3, 20.

US 2005/0162913 A1, 2005. 07. 28, 说明书第 34-60 段、附图 1-7.

CN 1466150 A, 2004. 01. 07, 全文.

US 2002/0126531 A1, 2002. 09. 12, 说明书第 142-172, 270 段、附图 1-3, 20.

审查员 李元

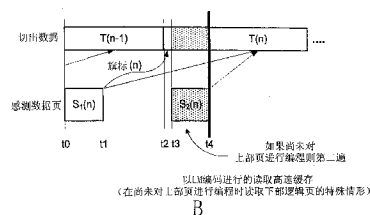
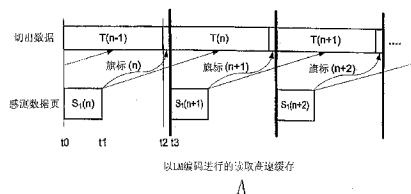
权利要求书 4 页 说明书 39 页 附图 42 页

(54) 发明名称

在读取操作期间具有后台数据锁存器高速缓存的非易失性存储器及其方法

(57) 摘要

来自存储器读取或写入操作的等待时间的一部分是用于将数据经由 I/O 总线输入到存储器的数据锁存器或从存储器的数据锁存器输出数据。存在通过允许存储器在存储器核心忙于读取操作时于后台中执行所述数据高速缓存及转移操作中的一些而改进非易失性存储器装置的性能的方法及电路。实施用于高速缓存读取数据的方案, 以使得即使对邻近字线上的数据的预先必要的读取必须先于对当前字线上的当前页的读取时, 在用于读取先前页的循环中抢先完成所述预先必要的读取连同任何 I/O 存取, 以使得可在所述先前读取的页忙于所述 I/O 存取的同时执行当前读取。



1. 一种非易失性存储器装置,其具有在相关联字线上的可寻址的存储器单元页,所述非易失性存储器装置包含:

为经寻址页的每一存储器单元提供的一组数据锁存器,所述组数据锁存器具有用于锁存预定数目的位的能力;

控制对指定群组的页进行的读取操作的状态机;

所述状态机在一系列读取循环中的每一者中感测并锁存数据页,其中在当前读取循环中的所述感测及锁存针对于当前字线上的当前页的数据,且响应于来自邻近字线的预先必要的数据而执行以校正从其产生的任何扰动效应;

所述状态机在所述当前读取循环之前抢先地感测并锁存所述当前页的所述预先必要的数据;且

所述状态机在所述当前读取循环而非第一读取循环中执行对所述当前页的所述感测及锁存,同时输出在刚刚过去的读取循环中感测并锁存的先前页。

2. 根据权利要求1所述的存储器装置,其中所述存储器单元每一者存储一个数据位。

3. 根据权利要求1所述的存储器装置,其中所述存储器单元每一者存储一个以上数据位,且字线上的每一存储器单元页与一个以上二进制数据页相关联,所述二进制数据页对应于每一存储器单元的个别位。

4. 根据权利要求3所述的存储器装置,其中所述状态机对所述当前页的所述预先必要的数据执行的抢先感测及锁存进一步包含:

输出作为来自所述抢先感测的所述预先必要的数据的一部分而获得的第一旗标;

响应于所述第一旗标而调整所述预先必要的数据;及

锁存经调整的所述预先必要的数据以指示是否需要所述当前页进行校正。

5. 根据权利要求4所述的存储器装置,其中所述第一旗标指示是否已就所有所述位对所述邻近字线上的预先必要的存储器单元页进行编程。

6. 根据权利要求5所述的存储器装置,其中:

当所述第一旗标指示所述邻近字线尚未就所有所述位而进行编程时,所述预先必要的数据经调整以指示不需要校正。

7. 根据权利要求5所述的存储器装置,其中:

当所述第一旗标指示所述邻近字线已就所有所述位而进行编程时,所述预先必要的数据经调整以指示需要校正。

8. 根据权利要求3所述的存储器装置,其中对当前页的所述状态机感测及锁存进一步包含:

通过来自经调整的所述预先必要的数据的校正而执行对所述当前页的所述感测;

输出作为所述当前页的一部分而获得的第二旗标以用于所述状态机;

响应于所述第二旗标,所述状态机通过将所述页保持为不改变或将所述页调整到预定值或者以通过在另一组感测条件下重复对所述当前页的所述感测而获得的新页进行替换来刷新所述当前页;及

锁存经刷新的所述当前页以用于在下一读取循环中输出。

9. 根据权利要求8所述的存储器装置,其中对那些存储器单元的如经调整的所述预先必要的数据所指示需要校正的所述当前页的所述状态机感测是以具有预定升压的字线电

压执行的。

10. 根据权利要求 8 所述的存储器装置,其中所述第二旗标指示是否已就所有所述位对所述当前字线上的存储器单元的所述当前页进行编程。

11. 根据权利要求 10 所述的存储器装置,其中:

当所述第二旗标指示所述邻近字线已就所有所述位而进行编程时,所述状态机通过将所述当前页保持为不改变而在空值操作中刷新所述当前页。

12. 根据权利要求 10 所述的存储器装置,其中:

所述存储器单元每一者存储两个数据位;

所述存储器单元页由下部位页及上部位页构成;且

当所述第二旗标指示所述邻近字线尚未就所有所述位而进行编程且正被感测的所述当前页为上部位页时,所述刷新所述当前页是通过将所述当前页全部重设为指示所述上部位页尚未进行编程的预定位而进行。

13. 根据权利要求 10 所述的存储器装置,其中:

所述存储器单元每一者存储两个数据位;

所述存储器单元页由下部位页及上部位页构成;且

当所述第二旗标指示所述邻近字线尚未就所有所述位而进行编程且正被感测的所述当前页为下部位页时,所述刷新所述当前页是通过相对于另一预定参考来感测所述当前页而进行。

14. 一种非易失性存储器装置,其具有在相关联的字线上的可寻址的存储器单元页,所述非易失性存储器装置包含:

为经寻址页的每一存储器单元提供的一组数据锁存器,所述组数据锁存器具有用于锁存预定数目的位的能力;

用于控制对指定群组的页进行的读取操作的构件;

用于在一系列读取循环中的每一者中感测并锁存数据页的构件,其中用于当前读取循环中的所述感测及锁存的所述构件针对于当前字线上的当前页的数据,且响应于来自邻近字线的预先必要的的数据以校正从其产生的任何扰动效应;

用于在所述当前读取循环之前抢先地感测并锁存所述当前页的所述预先必要的的数据的构件;且

所述用于在所述当前读取循环而非第一读取循环中感测并锁存的构件感测并锁存所述当前页,同时输出在刚刚过去的读取循环中感测并锁存的先前页。

15. 一种读取具有在相关联的字线上的可寻址的存储器单元页的非易失性存储器的方法,其包含:

在一系列读取循环中的每一者中感测并锁存数据页,其中在当前读取循环中的所述感测及锁存针对于当前字线上的当前页的数据,且响应于来自邻近字线的预先必要的的数据而执行以校正从其产生的任何扰动效应;

在所述当前读取循环之前抢先地感测并锁存所述当前页的所述预先必要的的数据;及

在所述当前读取循环而非第一读取循环中执行对所述当前页的所述感测及锁存,同时输出在刚刚过去的读取循环中感测并锁存的先前页。

16. 根据权利要求 15 所述的方法,其中所述存储器单元每一者存储一个数据位。

17. 根据权利要求 15 所述的方法,其中所述存储器单元每一者存储一个以上数据位,且字线上的每一存储器单元页与一个以上二进制数据页相关联,所述二进制数据页对应于每一存储器单元的个别位。

18. 根据权利要求 17 所述的方法,其中对所述当前页的所述预先必要的数据的所述抢先地感测及锁存进一步包含:

输出作为来自所述抢先地感测的所述预先必要的数据的一部分而获得的第一旗标;

响应于所述第一旗标而调整所述预先必要的数据;及

锁存经调整的所述预先必要的数据以指示是否需要所述当前页进行校正。

19. 根据权利要求 18 所述的方法,其中所述第一旗标指示是否已就所有所述位对所述邻近字线上的预先必要的存储器单元页进行编程。

20. 根据权利要求 19 所述的方法,其中:

当所述第一旗标指示所述邻近字线尚未就所有所述位而进行编程时,调整所述预先必要的数据以指示不需要校正。

21. 根据权利要求 19 所述的方法,其中:

当所述第一旗标指示所述邻近字线已就所有所述位而进行编程时,调整所述预先必要的数据以指示需要校正。

22. 根据权利要求 17 所述的方法,其中对当前页的所述感测及锁存进一步包含:

通过来自经调整的所述预先必要的数据的校正而执行对所述当前页的所述感测;

输出作为所述当前页的一部分而获得的第二旗标;

响应于所述第二旗标,通过将所述页保持为不改变或将所述页调整到预定值或者以通过在另一组感测条件下重复对所述当前页的所述感测而获得的新页进行替换来刷新所述当前页;及

锁存经刷新的所述当前页以用于在下一读取循环中输出。

23. 根据权利要求 22 所述的方法,其中以具有预定升压的字线电压执行对那些存储器单元的如经调整的所述预先必要的数据所指示需要校正的所述当前页的所述感测。

24. 根据权利要求 22 所述的方法,其中所述第二旗标指示是否已就所有所述位对所述当前字线上的存储器单元的所述当前页进行编程。

25. 根据权利要求 24 所述的方法,其中:

当所述第二旗标指示所述邻近字线已就所有所述位而进行编程时,所述刷新所述当前页是通过将所述当前页保持为不改变的空值操作。

26. 根据权利要求 24 所述的方法,其中:

所述存储器单元每一者存储两个数据位;

所述存储器单元页由下部位页及上部位页构成;且

当所述第二旗标指示所述邻近字线尚未就所有所述位而进行编程且正被感测的所述当前页为上部位页时,所述刷新所述当前页是通过将所述当前页全部重设为指示所述上部位页尚未经编程的预定位而进行。

27. 根据权利要求 24 所述的方法,其中:

所述存储器单元每一者存储两个数据位;

所述存储器单元页由下部位页及上部位页构成;且

当所述第二旗标指示所述邻近字线尚未就所有所述位而进行编程且正被感测的所述当前页为下部位页时,所述刷新所述当前页是通过相对于另一预定参考来感测所述当前页而进行。

在读取操作期间具有后台数据锁存器高速缓存的非易失性 存储器及其方法

技术领域

[0001] 本发明大体来说涉及例如电可擦除可编程只读存储器 (EEPROM) 及快闪 EEPROM 的非易失性半导体存储器, 且特定来说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] 能够进行电荷的非易失性存储 (特定来说采取封装为较小形状因数的卡的 EEPROM 及快闪 EEPROM 的形式) 的固态存储器近来已成为多种移动及手持装置 (尤其为信息设备及消费型电子产品) 中所选的存储装置。不同于也为固态存储器的 RAM (随机存取存储器), 快闪存储器为非易失性的, 即使在断开功率之后仍保持其存储的数据。尽管成本较高, 但快闪存储器越来越多地用于大容量存储应用中。基于例如硬驱动器及软磁盘的旋转磁性媒体的常规大容量存储装置不适于移动及手持环境。这是由于硬驱动器往往体积较大, 易于产生机械故障且具有较高等待时间及较高功率要求。所述不合需要的属性使得基于磁盘的存储装置在大多数移动及便携应用中不实用。另一方面, 嵌入式及采取可拆卸式卡的形式的快闪存储器由于其较小大小、较低功率消耗、较高速度及较高可靠性的特征而理想地适于移动及手持环境中。

[0003] EEPROM 及电可编程只读存储器 (EPROM) 为可经擦除且有新的数据写入或“编程”到存储器单元的非易失性存储器。两者均利用场效应晶体管结构中定位于半导体衬底中源极与漏极区域之间的沟道区域上的浮动 (未连接) 传导栅极。接着在浮动栅极上提供控制栅极。通过保持于浮动栅极上的电荷量来控制晶体管的阈值电压特征。即, 对于浮动栅极上的给定的电荷电平, 存在必须于“接通”晶体管以允许其源极与漏极区域之间的传导之前施加到控制栅极的对应电压 (阈值)。

[0004] 浮动栅极可保持一定范围的电荷且因此可经编程为阈值电压窗口内的任何阈值电压电平。通过装置的最小及最大阈值电平而对阈值电压窗口的大小定界, 所述阈值电平又对应于可编程到浮动栅极上的电荷的范围。阈值窗口一般视存储器装置的特征、操作条件及历史而定。窗口内的每一独特、可解析阈值电压电平范围可 (原则上) 用以表示单元的明确存储器状态。

[0005] 通常通过两个机制中的一者将用作存储器单元的晶体管编程为“经编程”状态。在“热电子注入”中, 施加到漏极的高电压加速电子跨越衬底沟道区域。同时, 施加到控制栅极的高电压经由薄栅极电介质将热电子拉到浮动栅极上。在“隧穿注入”中, 相对于衬底向控制栅极施加高电压。以此方式, 将电子从衬底拉到介入的浮动栅极。

[0006] 可通过许多机制而擦除存储器装置。对于 EPROM 来说, 通过以紫外线辐射将电荷从浮动栅极移除而可整体擦除存储器。对于 EEPROM 来说, 通过相对于控制栅极施加高电压到衬底以便诱发浮动栅极中的电子经由薄氧化物而隧穿到衬底沟道区域 (即, 福勒 - 诺德海姆隧穿) 而可电擦除存储器单元。通常, 可逐字节地擦除 EEPROM。对于快闪 EEPROM 来

说,可一次全部或每次一个或一个以上块地电擦除存储器,其中一块可由存储器的 512 或 512 以上的字节组成。

[0007] 非易失性存储器单元的实例

[0008] 存储器装置通常包含可安装于卡上的一个或一个以上存储器芯片。每一存储器芯片包含由例如解码器及擦除、写入及读取电路的外围电路支持的存储器单元的阵列。较为尖端的存储器装置还与执行智能型及较高层级的存储器操作及介接的控制器一同提供。现今正使用许多商业成功的非易失性固态存储器装置。所述存储器装置可使用不同类型的存储器单元,每一类型具有一个或一个以上电荷存储元件。

[0009] 图 1A 到图 1E 示意性地说明非易失性存储器单元的不同实例。

[0010] 图 1A 示意性地说明采取具有用于存储电荷的浮动栅极的 EEPROM 单元的形式非易失性存储器。电可擦除及可编程只读存储器 (EEPROM) 具有与 EPROM 类似的结构,但额外提供用于在施加适当电压时电装载电荷及从其浮动栅极电移除电荷 (无需暴露于 UV 辐射) 的机制。美国专利第 5,595,924 号中给出所述单元的实例及制造其的方法。

[0011] 图 1B 示意性地说明具有选择栅极及控制或操纵栅极的快闪 EEPROM 单元。存储器单元 10 具有源极 14 与漏极 16 扩散之间的“分开沟道 (split-channel)”12。用串联的两个晶体管 T1 及 T2 有效地形成单元。T1 用作具有浮动栅极 20 及控制栅极 30 的存储器晶体管。浮动栅极能够存储可选量的电荷。可流过沟道的 T1 部分的电荷的量视控制栅极 30 上的电压及驻留于介入的浮动栅极 20 上的电荷量而定。T2 用作具有选择栅极 40 的选择晶体管。当通过选择栅极 40 处的电压而接通 T2 时,其允许沟道的 T1 部分中的电流穿过源极与漏极之间。选择晶体管提供沿源极-漏极沟道独立于控制栅极处的电压的开关。一优点在于其可用以断开于零控制栅极电压下仍导通的所述单元,零控制栅极电压归因于所述单元在其浮动栅极处的电荷耗尽 (正)。另一优点在于其使得能够较易于实施源极侧注入编程。

[0012] 分开沟道存储器单元的一简单实施例为选择栅极与控制栅极连接到如通过图 1B 所示的虚线而示意性地指示的同一字线的存储器单元。此通过将电荷存储元件 (浮动栅极) 定位于沟道的一部分上且将控制栅极结构 (其为字线的部分) 定位于另一沟道部分以及电荷存储元件上而完成。此用串联的两个晶体管有效地形成单元,一晶体管 (存储器晶体管) 具有电荷存储元件上的电荷量与字线上的电压的组合,所述组合控制可流过其沟道的部分的电流,且另一晶体管 (选择晶体管) 具有单独用作其栅极的字线。美国专利第 5,070,032、5,095,344、5,315,541、5,343,063 及 5,661,053 号中给出所述单元的实例、其在存储器系统中的使用及制造其的方法。

[0013] 图 1B 所示的分开沟道单元的一较为精细的实施例为选择栅极与控制栅极独立且不由其间的虚线相连接时的实施例。一实施方案具有单元阵列中的一列控制栅极,其连接到垂直于字线的控制 (或操纵) 线。作用为使得字线无需在读取或编程所选单元时同时执行两个功能。所述两个功能为 (1) 用作选择晶体管的栅极,因此要求适当电压以接通及断开选择晶体管,及 (2) 经由耦合于字线与电荷存储元件之间的电场 (电容性的) 而将电荷存储元件的电压驱动到所要电平。通常难以通过单一电压而以最佳方式执行所述功能的两者。通过对控制栅极与选择栅极的单独控制,字线仅需执行功能 (1),而所添加的控制线执行功能 (2)。此能力允许较高性能编程的设计,其中使编程电压适应目标数据。独立控

制（或操纵）栅极在快闪 EEPROM 阵列中的使用描述于（例如）美国专利第 5,313,421 及 6,222,762 号中。

[0014] 图 1C 示意性地说明具有双浮动栅极及独立的选择及控制栅极的另一快闪 EEPROM 单元。存储器单元 10 类似于图 1B 的存储器单元，除了其有效地具有串联的三个晶体管。在此类型的单元中，在其于源极与漏极扩散之间的沟道上包括两个存储元件（即，T1- 左及 T1- 右的存储元件）连同所述两个存储元件之间的选择晶体管 T1。存储器晶体管分别具有浮动栅极 20 及 20'，以及控制栅极 30 及 30'。由选择栅极 40 来控制选择晶体管 T2。任何时候仅存取所述对存储器晶体管中的一者用于读取或写入。当存取存储单位 T1- 左时，接通 T2 及 T1- 右以允许沟道的 T1- 左的部分中的电流通过源极与漏极之间。类似地，当存取存储单位 T1- 右时，接通 T2 及 T1- 左。通过使选择栅极多晶硅的一部分接近于浮动栅极及向选择栅极施加大量正电压（例如，20V）以使得存储于浮动栅极内的电子可隧穿到选择栅极多晶硅而实现擦除。

[0015] 图 1D 示意性地说明经组织为 NAND 单元的一串存储器单元。NAND 单元 50 由一连串存储器晶体管 M1、M2、...、Mn ($n = 4, 8, 16$ 或 16 以上) 组成，所述存储器晶体管通过其源极及漏极而经菊式链接。一对选择晶体管 S1、S2 控制存储器晶体管链经由 NAND 单元的源极端子 54 及漏极端子 56 与外部的连接。在存储器阵列中，当接通源极选择晶体管 S1 时，源极端子耦合到源极线。类似地，当接通漏极选择晶体管 S2 时，NAND 单元的漏极端子耦合到存储器阵列的位线。链中的每一存储器晶体管具有电荷存储元件以存储给定量的电荷从而表示预期的存储器状态。每一存储器晶体管的控制栅极提供对读取及写入操作的控制。选择晶体管 S1、S2 中的每一者的控制栅极分别经由 NAND 单元的源极端子 54 及漏极端子 56 而提供对 NAND 单元的控制存取。

[0016] 当在编程期间读取并验证 NAND 单元内的经寻址的存储器晶体管时，向其控制栅极供应适当电压。同时，NAND 单元 50 中未经寻址的存储器晶体管的剩余部分通过在其控制栅极上施加充足电压而完全接通。以此方式，从个别存储器晶体管的源极到 NAND 单元的源极端子 54 及同样地针对个别存储器晶体管的漏极到单元的漏极端子 56 而形成的传导路径为有效的。美国专利第 5,570,315、5,903,495、6,046,935 号中描述具有所述 NAND 单元结构的存储器装置。

[0017] 图 1E 示意性地说明具有用于存储电荷的介电层的非易失性存储器。替代早先描述的传导浮动栅极元件而使用介电层。利用介电存储元件的所述存储器装置已由埃特恩 (Eitan) 等人于“NROM: A Novel Localized Trapping, 2-Bit Nonvolatile Memory Cell (NROM: 新颖的定位捕集、2 位非易失性存储器单元),” IEEE Electron Device Letters (IEEE 电子器件快报), 第 21 卷, 第 11 号, 2000 年 11 月, 第 543-545 页中描述。ONO 介电层延伸跨越源极与漏极扩散之间的沟道。将一数据位的电荷定位于介电层中邻近于漏极处，且将另一数据位的电荷定位于介电层中邻近于源极处。举例来说，美国专利第 5,768,192 及 6,011,725 号揭示具有夹于两个二氧化硅层之间的捕集电介质的非易失性存储器单元。通过单独读取电介质内的空间分离电荷存储区域的二元状态来实施多状态数据存储。

[0018] 存储器阵列

[0019] 存储器装置通常包含布置成行及列且可通过字线及位线寻址的存储器单元的二

维阵列。可根据 NOR 型或 NAND 型架构而形成阵列。

[0020] NOR 阵列

[0021] 图 2 说明存储器单元的 NOR 阵列的一实例。已通过图 1B 或图 1C 中所说明的类型的单元而实施具有 NOR 型架构的存储器装置。每一行存储器单元通过其源极及漏极以菊式链接的方式而连接。有时将此设计称作虚接地设计。每一存储器单元 10 具有源极 14、漏极 16、控制栅极 30 及选择栅极 40。一行中的单元使其选择栅极连接到字线 42。一列中的单元使其源极及漏极分别连接到所选位线 34 及 36。在独立控制存储器单元的控制栅极及选择栅极的一些实施例中,操纵线 36 还连接一列中的单元的控制栅极。

[0022] 许多快闪 EEPROM 装置通过存储器单元而实施,其中每一存储器单元由其连接到一起的控制栅极及选择栅极而形成。在此情形中,不需要操纵线且字线简单地连接沿每一行的单元的所有控制栅极及选择栅极。美国专利第 5,172,338 及 5,418,752 号中揭示所述设计的实例。在所述设计中,字线本质上执行两个功能:行选择及向行中的所有单元供应控制栅极电压以进行读取或编程。

[0023] NAND 阵列

[0024] 图 3 说明例如图 1D 所示的存储器单元的 NAND 阵列的一实例。沿 NAND 单元的每一列,位线耦合到每一 NAND 单元的漏极端子 56。沿 NAND 单元的每一行,源极线可连接所有其源极端子 54。沿一行的 NAND 单元的控制栅极还连接到一连串对应字线。可通过经由所连接的字线以其控制栅极上的适当电压接通所述对选择晶体管(见图 1D)而寻址一整行 NAND 单元。在读取 NAND 单元的链内的一存储器晶体管时,链中的剩余存储器晶体管经由其相关联的字线而经硬接通以使得流过所述链的电流本质上视存储于所读取的单元中的电荷的电平而定。在美国专利第 5,570,315、5,774,397 及 6,046,935 号中可找到 NAND 架构阵列的一实例及其作为存储器系统的部分的操作。

[0025] 块擦除

[0026] 电荷存储存储器装置的编程可仅导致向其电荷存储元件添加更多电荷。因此,在编程操作之前,必须移除(或擦除)电荷存储元件中的现有电荷。提供擦除电路(未图示)以擦除存储器单元的一个或一个以上块。当一同(即,瞬间)电擦除单元的整个阵列或阵列的单元的重要群组时,将例如 EEPROM 的非易失性存储器称作“快闪”EEPROM。一旦经擦除,则单元的群组可经再编程。可一同擦除的单元的群组可由一个或一个以上可寻址擦除单位组成。擦除单位或块通常存储一个或一个以上数据页,页为编程及读取的单位,但可在单一操作中编程或读取一个以上页。每一页通常存储一个或一个以上扇区的数据,扇区的大小由主机系统来界定。一实例为具有 512 字节的用户数据(其遵循关于磁盘驱动器而建立的标准)加上某一数目的字节的关于用户数据及/或其存储于的块的开销信息的扇区。

[0027] 读取/写入电路

[0028] 在常见双态 EEPROM 单元中,建立至少一电流断点电平以便将导通窗口分割为两个区域。当通过施加预定、固定电压而读取单元时,其源极/漏极电流通过与断点电平(或参考电流 IREF)比较而转变为存储器状态。如果电流读数高于断点电平的读数,则确定单元处于一逻辑状态(例如,“零”状态)。另一方面,如果电流小于断点电平的电流,则确定单元处于另一逻辑状态(例如,“一”状态)。因此,所述双态单元存储一位的数字信息。通常提供可在外部可编程的参考电流源作为存储器系统的部分以产生断点电平电流。

[0029] 为了增大存储器容量,随着半导体技术的状态进步而制造具有越来越高的密度的快闪 EEPROM 装置。增大存储容量的另一方法为使得每一存储器单元存储两个以上状态。

[0030] 对于多状态或多电平 EEPROM 存储器单元,通过一个以上断点将执行窗口分割为两个以上区域以使得每一单元能够存储一个以上数据位。给定 EEPROM 阵列可存储的信息因此随每一单元可存储的状态的数目而增加。美国专利第 5,172,338 号中已描述具有多状态或多电平存储器单元的 EEPROM 或快闪 EEPROM。

[0031] 实际上,通常通过在向控制栅极施加参考电压时感测跨越单元的源极与漏极电极的传导电流而读取单元的存储器状态。因此,对于单元的浮动栅极上的每一给定电荷,可检测到相对于固定参考控制栅极电压的对应传导电流。类似地,可编程到浮动栅极上的电荷的范围界定对应阈值电压窗口或对应传导电流窗口。

[0032] 或者,替代检测经分割的电流窗口中的传导电流,在控制栅极处的测试中可能对于给定存储器状态设定阈值电压且检测传导电流是否低于或高于阈值电流。在一实施方案中,通过检查传导电流经由位线的电容而放电的速率来完成对传导电流相对于阈值电流的检测。

[0033] 图 4 说明对于浮动栅极于任一时间可选择性地存储的四个不同电荷 Q1-Q4 的源极-漏极电流 I_D 与控制栅极电压 V_{CG} 之间的关系。四个实线 I_D 与 V_{CG} 的关系曲线表示可在存储器单元的浮动栅极上编程的四个可能电荷电平,其分别对应于四个可能存储器状态。作为一实例,一定数目的单元的阈值电压窗口可在 0.5V 到 3.5V 的范围中。可通过以各 0.5V 的间隔将阈值窗口分割为五个区域而划分六个存储器状态。举例来说,如果如图示而使用 $2\mu A$ 的参考电流 I_{REF} ,则可将以 Q1 而编程的单元视作处于存储器状态“1”中,因为其曲线与 I_{REF} 相交于阈值窗口的通过 $V_{CG} = 0.5V$ 与 $1.0V$ 所划分的区域中。类似地, Q4 处于存储器状态“5”中。

[0034] 如从上文的描述可见,使存储器单元存储越多状态,其阈值窗口受到越细致的划分。此将要求编程及读取操作中的较高精确度以便能够实现所需分辨率。

[0035] 美国专利第 4,357,685 号揭示编程 2- 状态 EPROM 的方法,其中当将单元编程为给定状态时,其经受连续编程电压脉冲,每一次向浮动栅极添加递增的电荷。在脉冲之间,回读或验证单元以确定其相对于断点电平的源极-漏极电流。在已验证当前状态达到所要状态时停止编程。所使用的编程脉冲串可具有递增的周期或振幅。

[0036] 现有技术的编程电路简单地施加编程脉冲以从擦除状态或基态逐步调试阈值窗口直到达到目标状态。实际上,为了允许足够的分辨率,每一经分割或划分的区域将需要至少约五个编程步骤以横穿。性能对于 2- 状态存储器单元来说为可接受的。然而,对于多状态单元,所需步骤的数目随分割的数目而增加,且因此,编程精确度或分辨率必然增加。举例来说,16- 状态单元可能需要平均至少 40 个编程脉冲以编程到一目标状态。

[0037] 图 5 示意性地说明具有存储器阵列 100 的通过读取/写入电路 170 经由行解码器 130 及列解码器 160 可存取的典型布置的存储器装置。如结合图 2 及图 3 所描述,存储器阵列 100 中的存储器单元的存储器晶体管可经由一组所选字线及位线而寻址。行解码器 130 选择一个或一个以上字线且列解码器 160 选择一个或一个以上位线以便向经寻址的存储器晶体管的相应栅极施加适当电压。提供读取/写入电路 170 以读取或写入(编程)经寻址的存储器晶体管的存储器状态。读取/写入电路 170 包含可经由位线连接到阵列中的

存储器元件的若干读取 / 写入模块。

[0038] 图 6A 为个别读取 / 写入模块 190 的示意性框图。本质上,在读取或验证期间,感测放大器确定流过经由所选位线而连接的经寻址的存储器晶体管的漏极的电流。所述电流视存储于存储器晶体管中的电荷及其控制栅极电压而定。举例来说,在多状态 EEPROM 单元中,可对其浮动栅极充电到若干不同电平中的一者。对于 4- 电平单元,其可用以存储两个数据位。通过电平 - 位转换逻辑而将感测放大器所检测的电平转换为待存储于数据锁存器中的一组数据位。

[0039] 影响读取 / 写入性能及准确性的因素

[0040] 为了改进读取及编程性能,并行地读取或编程阵列中的多个电荷存储元件或存储器晶体管。因此,一同读取或编程存储器元件的逻辑“页”。在现有存储器架构中,一行通常含有若干交错页。将一同读取或编程一页的所有存储器元件。列解码器将选择性地将交错页中的每一者连接到对应数目的读取 / 写入模块。举例来说,在一实施方案中,将存储器阵列设计为具有 532 字节 (512 字节加上 20 字节的开销) 的页大小。如果每一列含有一漏极位线且每一行存在两个交错页,则此量达到 8512 列,其中每一页与 4256 列相关联。将存在可连接以并行读取或写入所有偶数位线或奇数位线的 4256 个感测模块。以此方式,从存储器元件的页读取数据的并行的 4256 个位 (即,532 字节) 的页或将其编程到存储器元件的页。可将形成读取 / 写入电路 170 的读取 / 写入模块布置为各种架构。

[0041] 参看图 5,将读取 / 写入电路 170 组织为成组的读取 / 写入堆栈 180。每一读取 / 写入堆栈 180 为读取 / 写入模块 190 的堆栈。在存储器阵列中,列间距由占据其的一或两个晶体管的大小而确定。然而,如从图 6A 可见,读取 / 写入模块的电路将可能用多得多的晶体管及电路元件而实施且因此将占据越过许多列的空间。为了服务于所占据的列中的一个以上的列,将多个模块堆叠于彼此顶部上。

[0042] 图 6B 展示由读取 / 写入模块 190 的堆栈按照惯例实施的图 5 的读取 / 写入堆栈。举例来说,读取 / 写入模块可延伸越过十六列,接着可使用具有八个读取 / 写入模块的堆栈的读取 / 写入堆栈 180 来服务并联的八列。读取 / 写入堆栈可经由列解码器而耦合到组中的八个奇数 (1、3、5、7、9、11、13、15) 列或八个偶数 (2、4、6、8、10、12、14、16) 列。

[0043] 如前文所提及,常规存储器装置通过以整体并行方式一次对所有偶数或所有奇数位线进行操作而改进读取 / 写入操作。由两个交错页组成的行的此架构将有助于减轻装配读取 / 写入电路的块的问题。其还由对控制位线到位线的电容耦合的考虑而规定。使用块解码器以将所述组读取 / 写入模块多路复用到偶数页或奇数页。以此方式,当读取或编程一组位线时,可使交错组接地以最小化紧邻的耦合。

[0044] 然而,交错页架构在至少三个方面存在缺点。第一,其需要额外多路复用电路。第二,其在性能上较慢。为了完成由字线连接的或一行中的存储器单元的读取或编程,需要两个读取或两个编程操作。第三,其在处理例如浮动栅极电平处的相邻的电荷存储元件之间的场耦合 (当例如分别处于奇数与偶数页中的两个相邻元件于不同时间受到编程时) 的其它干扰效应中也不是最佳的。

[0045] 相邻场耦合的问题随着存储器晶体管之间的日益紧密的间距而变得较为显著。在存储器晶体管中,电荷存储元件夹于沟道区域与控制栅极之间。在沟道区域中流动的电流依据由控制栅极与电荷存储元件处的场起作用的合成电场而变。随着日益增加的密度,存

储器晶体管越来越紧密地形成在一起。接着来自相邻电荷元件的场变为对受影响的单元的合成场的重要贡献者。相邻场视经编程到相邻者的电荷存储元件中的电荷而定。此扰动场本质上为动态的,因为其随相邻者的编程状态而改变。因此,可视相邻者的改变状态而在不同时间对受影响的单元进行不同的读取。

[0046] 交错页的常规架构加剧由相邻浮动栅极耦合而造成的误差。由于偶数页与奇数页独立于彼此而经编程及读取,因此可视当时发生于交错页上的情况而在一组条件下编程一页但在一个完全不同组的条件下回读所述页。读取误差随着增加的密度将变得较为严重,从而要求较为准确的读取操作及对阈值窗口的较粗分割以用于多状态实施方案。性能将受到损害且多状态实施方案中的潜在能力受到限制。

[0047] 美国专利公开案第 US-2004-0060031-A1 号揭示高性能而又紧密的非易失性存储器装置,其具有读取/写入电路的一较大块以并行地读取及写入存储器单元的对应块。具体来说,存储器装置具有将读取/写入电路的块中的冗余度降低到最小的架构。通过将读取/写入模块的块再分配到块读取/写入模块核心部分中而完成空间以及功率的显著节省,所述核心部分在与显著较小组的共同部分以时间多路复用的方式相互作用时并行操作。具体来说,通过共享处理器来执行多个感测放大器与数据锁存器之间的读取/写入电路中的数据处理。

[0048] 因此,存在对于高性能及高容量非易失性存储器的普遍需要。具体来说,存在对于具有增强的读取及编程性能的紧密非易失性存储器的需要,所述存储器具有紧密且有效而对于处理读取/写入电路中的数据更是高通用的改进处理器。

发明内容

[0049] 根据本发明的一方面,提出允许在内部存储器处于例如读取、编程或擦除的另一操作中时将数据转移入或转移出存储器的高速缓存操作。具体来说,描述允许所述高速缓存操作的数据锁存器的布置及其使用方法。

[0050] 描述许多物理页共享数据锁存器的架构。举例来说,读取/写入堆栈与存储器的由多个字线共享的位线相关联。当一操作在存储器中进行时,如果所述锁存器中的任一者空闲,则其可高速缓存数据用于同一或另一字线中的将来的操作,从而节省转移时间,因此可藏于另一操作之后。此可通过增加对不同操作或操作的不同阶段的管线式操作的量而改进性能。在一实例中,在高速缓存编程操作中,当编程一页数据时,可加载另一页数据以节省转移时间。对于另一实例,在一示范性实施例中,将对一字线的读取操作插入到对另一字线的写入操作中,从而允许来自所述读取的数据在数据写入继续的同时转移出存储器。

[0051] 根据各种方面,可在写入或其它操作对于第一页数据进行的同时将来自同一块中但不同字线上的另一页的数据切出(toggle out)(以(例如)进行 ECC 操作)。对操作的此阶段间管线式操作允许数据转移所需的时间藏于对第一页数据的操作之后。更一般地,此允许将一操作的一部分插入于另一操作(通常较长)的阶段之间。另一实例会将感测操作插入于(如)擦除操作的阶段之间,例如在擦除脉冲之前或在用作擦除的稍后部分的软编程阶段之前。

[0052] 如果正执行一具有不同阶段的相对较长的操作,则主要方面将通过使用读取/写入堆栈的共享锁存器(如果锁存器可用)而插入较快速的操作。举例来说,可将读取插入

于编程或擦除操作中,或者可将二进制编程插入于擦除中。主要示范性实施例将在编程操作期间针对一页切入及 / 或切出数据以用于另一页,所述页共享相同的读取写入堆栈,其中(例如),将对待切出并修改的数据的读取插入于数据写入的验证阶段中。

[0053] 开放的数据锁存器的可用性可以许多方式而发生。一般来说,对于每单元存储 n 个位的存储器来说,对于每一位线将需要 n 个所述数据锁存器;然而,并非总是需要所述锁存器的全部。举例来说,在以上部页 / 下部页的格式存储数据的每单元两位的存储器中,在编程下部页时将需要一个数据锁存器(如果实施快速通过写入则使用另一锁存器)。在编程上部页时将需要两个数据锁存器(如果实施快速通过写入则使用第三锁存器)。更一般来说,对于存储多个页的存储器来说,仅在编程最高页时将需要全部锁存器。此使得其它锁存器可用于高速缓存操作。此外,即使在写入最高页时,由于从写入操作的验证阶段移除各种状态,因此锁存器将为自由的。特定来说,一旦仅剩最高状态待验证,则仅需单一锁存器用于验证的目的且其它锁存器可用于高速缓存操作。

[0054] 一示范性实施例基于每单元存储两个位且具有针对每一位线上的数据的两个锁存器及用于快速通过写入的一额外锁存器的四状态存储器。写入下部页或擦除或进行后擦除软编程的操作基本上为二进制操作且其中数据锁存器中的一者为空闲的,可使用其来高速缓存数据。类似地,在进行上部页或全序列写入时,一旦除最高层级之外的所有层级已经验证,则仅单一状态需验证且存储器可释放一锁存器,可使用所述锁存器来高速缓存数据。如何可使用此的一实例为在(例如于复制操作中)编程一页时,对共享同一组的数据锁存器的另一页(例如同一组位线上的另一字线)的读取可在编程脉冲与写入的验证之间插入。接着可将地址切换到正写入的页,允许写入处理在其停止处处拾起而无需重新开始。在写入继续的同时,在插入的读取期间高速缓存的数据可经切出、检查或修改且转移返回以存在用于在一旦早先写入操作完成时即写回。此种类的高速缓存操作允许将对第二页数据的切出及修改藏于对第一页的编程之后。

[0055] 读取期间的高速缓存操作——后台读取及写入操作

[0056] 根据本发明的一方面,提供一读取高速缓存方案以用于具有最小化存储器单元之间的扰动(Yupin 效应)的需要的多状态存储器单元的情形。用于高速缓存读取数据的方案如下:即使对于校正视来自相邻物理页或字线的数据而定的读取操作,数据锁存器及 I/O 总线也有效地用以在当前页正从存储器核心而被感测的同时切出先前读取页。

[0057] 在优选实施方案中,对于以“LM”编码而编码且以先行(“LA”)校正而读取的存储器使用有效读取高速缓存方案。“LM”编码及“LA”校正两者均需要除仅仅切换读取数据以外的额外锁存器及总线活动。在必须以对邻近字线上的数据的预先必要的读取而先行于对当前字线上的当前页的读取时,所述预先必要的读取连同任何 I/O 存取在读取先前页的循环中经抢先完成以使得可在先前读取页忙于 I/O 存取的同时执行当前读取。

[0058] 根据本发明的另一方面,对于一个以上的位经一同感测的多位存储器单元页实施读取高速缓存方案。在优选实施方案中,一同感测所有多个位。所感测的多位存储器单元页对应于多个二进制页,对于多个位中的每一者存在一个二进制页。在每一循环中输出二进制页,同时在多个循环中执行多位感测。所述多个与多个位中的位的数目相同。

[0059] 将从以下对于其优选实施例的描述而了解本发明的额外特征及优点,应结合附图而进行所述描述。

附图说明

- [0060] 图 1A 到图 1E 示意性地说明非易失性存储器单元的不同实例。
- [0061] 图 2 说明存储器单元的 NOR 阵列的一实例。
- [0062] 图 3 说明例如图 1D 所示的存储器单元的 NAND 阵列的一实例。
- [0063] 图 4 说明对于浮动栅极于任一时间可存储的四个不同电荷 Q1-Q4 的源极 - 漏极电流与控制栅极电压之间的关系。
- [0064] 图 5 示意性地说明通过读取 / 写入电路经由行及列解码器可存取的存储器阵列的典型布置。
- [0065] 图 6A 为个别读取 / 写入模块的示意性框图。
- [0066] 图 6B 展示由读取 / 写入模块的堆栈按照惯例实施的图 5 的读取 / 写入堆栈。
- [0067] 图 7A 示意性地说明具有一组经分割的读取 / 写入堆栈的的紧密存储器装置, 其中实施本发明的改进处理器。
- [0068] 图 7B 说明图 7A 所示的紧密存储器装置的优选布置。
- [0069] 图 8 示意性地说明图 7A 所示的读取 / 写入堆栈中的基本组件的一般布置。
- [0070] 图 9 说明图 7A 及图 7B 所示的读取 / 写入电路中的读取 / 写入堆栈的一优选布置。
- [0071] 图 10 说明图 9 所示的共用处理器的改进实施例。
- [0072] 图 11A 说明图 10 所示的共用处理器的输入逻辑的优选实施例。
- [0073] 图 11B 说明图 11A 的输入逻辑的真值表。
- [0074] 图 12A 说明图 10 所示的共用处理器的输出逻辑的优选实施例。
- [0075] 图 12B 说明图 12A 的输出逻辑的真值表。
- [0076] 图 13 为图 10 的简化版本, 其展示在本发明的二位实施例中与当前论述相关的一些特定元件。
- [0077] 图 14 指示对于其中读入下部页数据的上部页编程的对与图 13 的相同元件的锁存器指派。
- [0078] 图 15 说明以单页模式进行的高速缓存编程的方面。
- [0079] 图 16 展示可用于下部页到全序列转换中的编程波形。
- [0080] 图 17 说明在具有全序列转换的高速缓存编程操作中的相对时序。
- [0081] 图 18 描述锁存器在高速缓存页复制操作中的部署。
- [0082] 图 19A 及图 19B 说明高速缓存页复制操作中的相对时序。
- [0083] 图 20A 说明在每一存储器单元使用 LM 代码存储两个位的数据时 4 状态存储器阵列的阈值电压分布。
- [0084] 图 20B 说明使用 LM 代码在现有 2 循环编程方案中进行的下部页编程。
- [0085] 图 20C 说明使用 LM 代码在现有 2 循环编程方案中进行的上部页编程。
- [0086] 图 20D 说明辨别以 LM 代码编码的 4 状态存储器的下部位所需的读取操作。
- [0087] 图 20E 说明辨别以 LM 代码编码的 4 状态存储器的上部位所需的读取操作。
- [0088] 图 21 为下部页编程的示意性时序图, 其说明将下一页编程数据加载到未使用的数据锁存器中的后台操作。
- [0089] 图 22 为展示在使用 QWP 的 4 状态上部页或全序列编程的各种阶段期间需跟踪的

状态的数目的表。

[0090] 图 23 为上部页或全序列编程的示意性时序图,其说明将下一页编程数据加载到未使用的数据锁存器中的后台操作。

[0091] 图 24 为说明根据本发明的一般实施例的与当前多阶段存储器操作同时发生的锁存器操作的流程图。

[0092] 图 25 为下部页编程的示意性时序图,其说明使用可用锁存器而进行的读取中断操作。

[0093] 图 26 为上部页编程的示意性时序图,其说明使用可用锁存器而进行的读取中断操作。

[0094] 图 27 说明与典型存储器操作相关联的信息的封装。

[0095] 图 28 说明支持简单高速缓存操作的常规存储器系统。

[0096] 图 29 为说明多个存储器操作的排队及可能合并的流程图。

[0097] 图 30 说明并入有存储器操作队列及存储器操作队列管理器的优选芯片上控制电路的示意性框图。

[0098] 图 31 为说明擦除操作期间在后台中的高速缓存操作的示意性流程图。

[0099] 图 32 为对存储器阵列进行的擦除操作的示意性时序图,其说明擦除操作的第一擦除阶段期间的编程数据加载操作。

[0100] 图 33 为对存储器阵列进行的擦除操作的示意性时序图,其说明擦除操作的软编程 / 验证阶段期间的编程数据加载操作。

[0101] 图 34 为对存储器阵列进行的擦除操作的示意性时序图,其说明插入的读取操作及使用可用锁存器而进行的所得数据输出操作。

[0102] 图 35 为说明图 31 的步骤 780 中在擦除操作期间在后台中用于读取擦洗应用的特定高速缓存操作的示意性流程图。

[0103] 图 36 说明擦除期间的抢先后台读取。

[0104] 图 37 示意性地说明典型读取高速缓存方案。

[0105] 图 38A 为针对高速缓存读取以 LM 代码编码的逻辑页的示意性时序图。

[0106] 图 38B 为针对以 LM 代码进行的高速缓存读取在尚未对上部位逻辑页进行编程时读取下部位逻辑页的特殊情形中的示意性时序图。

[0107] 图 39 说明对于 2 位存储器以所有位感测而进行的高速缓存读取的示意性时序图。

[0108] 图 40 说明一存储器的实例,其具有 2 位存储器单元且其中其页以最佳序列编程从而最小化邻近字线上的存储器单元之间的 Yupin 效应。

[0109] 图 41 说明根据图 37 所示的常规方案对于 LM 代码连同 LA 校正的读取高速缓存的实施方案。

[0110] 图 42 说明以 LM 代码及 LA 校正进行的改进读取高速缓存方案。

[0111] 图 43 为说明改进读取高速缓存的示意性流程图。

[0112] 图 44 为以进一步的清晰度说明图 43 的步骤 850 的示意性流程图。

[0113] 图 45 为以进一步的清晰度说明图 43 的步骤 830 的示意性流程图。

具体实施方式

[0114] 图 7A 示意性地说明具有一组经分割的读取 / 写入堆栈的紧密存储器装置, 其中实施本发明的改进处理器。存储器装置包括存储器单元的二维阵列 300、控制电路 310 及读取 / 写入电路 370。存储器阵列 300 通过字线经由行解码器 330 及通过位线经由列解码器 360 而可寻址。读取 / 写入电路 370 经实施为一组经分割的读取 / 写入堆栈 400 且允许存储器单元的块 (还称作“页”) 经并行地读取或编程。在优选实施例中, 一页由一行邻接存储器单元构成。在将一行存储器单元分割为多个块或页的另一实施例中, 提供块多路复用器 350 以将读取 / 写入电路 370 多路复用到个别块。

[0115] 控制电路 310 与读取 / 写入电路 370 协作以对存储器阵列 300 执行存储器操作。控制电路 310 包括状态机 312、芯片上地址解码器 314 及功率控制模块 316。状态机 312 提供存储器操作的芯片级控制。芯片上地址解码器 314 提供由主机或存储器控制器所使用的地址与解码器 330 及 370 所使用的硬件地址之间的地址接口。功率控制模块 316 控制在存储器操作期间供应到字线及位线的功率及电压。

[0116] 图 7B 说明图 7A 所示的紧密存储器装置的优选布置。以在阵列的相对侧上对称的方式实施各种外围电路对存储器阵列 300 的存取, 以使得每侧上的存取线及电路减半。因此, 将行解码器拆分为行解码器 330A 及 330B 且将列解码器拆分为列解码器 360A 及 360B。在将一行存储器单元分割为多个块的实施例中, 将块多路复用器 350 拆分为块多路复用器 350A 及 350B。类似地, 将读取 / 写入电路拆分为从阵列 300 的底部连接到位线的读取 / 写入电路 370A 及从阵列 300 的顶部连接到位线的读取 / 写入电路 370B。以此方式, 读取 / 写入模块的密度及因此经分割的读取 / 写入堆栈 400 的密度本质上减半。

[0117] 图 8 示意性地说明图 7A 所示的读取 / 写入堆栈中的基本组件的一般布置。根据本发明的一般架构, 读取 / 写入堆栈 400 包含用于感测 k 个位线的感测放大器堆栈 212、用于数据经由 I/O 总线 231 的输入或输出的 I/O 模块 440、用于存储输入或输出数据的数据锁存器堆栈 430、用以处理及存储读取 / 写入堆栈 400 中的数据的数据的共用处理器 500 及用于堆栈组件中的通信的堆栈总线 421。读取 / 写入电路 370 中的堆栈总线控制器经由线 411 提供控制及定时信号以控制读取 / 写入堆栈中的各种组件。

[0118] 图 9 说明图 7A 及图 7B 所示的读取 / 写入电路中的读取 / 写入堆栈的一优选布置。每一读取 / 写入堆栈 400 对 k 个位线的群组并行地操作。如果一页具有 $p = r * k$ 个位线, 则将存在 r 个读取 / 写入堆栈 400-1, ..., 400- r 。

[0119] 并行操作的经分割的读取 / 写入堆栈 400 的整个组允许并行地读取或编程沿一行的 p 个单元的块 (或页)。因此, 将存在 p 个读取 / 写入模块用于整行单元。由于每一堆栈服务于 k 个存储器单元, 因此通过 $r = p/k$ 而给出组中的读取 / 写入堆栈的总数。举例来说, 如果 r 为组中的堆栈的数目, 则 $p = r * k$ 。一实例存储器阵列可具有 $p = 512$ 字节 (512×8 位), $k = 8$ 且因此 $r = 512$ 。在优选实施例中, 块为整行单元的游程。在另一实施例中, 块为行中的单元的子组。举例来说, 单元的子组可为整行的一半或整行的四分之一。单元的子组可为一游程的邻接单元或每隔一个的单元或每隔预定数目个的单元。

[0120] 例如 400-1 的每一读取 / 写入堆栈本质上含有并行地服务于 k 个存储器单元的段的感测放大器 212-1 到 212- k 的堆栈。美国专利公开案第 2004-0109357-A1 号中揭示优选感测放大器, 所述公开案的全部揭示内容以引用的方式并入本文中。

[0121] 堆栈总线控制器 410 经由线 411 向读取 / 写入电路 370 提供控制及定时信号。堆

栈总线控制器自身经由线 311 视存储器控制器 310 而定。通过互连堆栈总线 431 而实现每一读取 / 写入堆栈 400 中的通信且通过堆栈总线控制器 410 而控制所述通信。控制线 411 从堆栈总线控制器 410 向读取 / 写入堆栈 400-1 的组件提供控制及时钟信号。

[0122] 在优选布置中,将堆栈总线分割为用于共用处理器 500 与感测放大器的堆栈 212 之间的通信的 SA 总线 (SABus) 422 及用于处理器与数据锁存器的堆栈 430 之间的通信的 D 总线 (DBus) 423。

[0123] 数据锁存器的堆栈 430 包含数据锁存器 430-1 到 430-k,对于与堆栈相关联的每一存储器单元存在一者。I/O 模块 440 使得数据锁存器能够经由 I/O 总线 231 与外部交换数据。

[0124] 共用处理器还包括用于输出指示存储器操作的状态 (例如误差状况) 的状态信号的输出 507。状态信号用以驱动线或 (Wired-Or) 配置中连结到旗标总线 (FLAG BUS) 509 的 n 晶体管 550 的栅极。旗标总线优选地通过控制器 310 而预充电且将在读取 / 写入堆栈中的任一者断言状态信号时被下拉。

[0125] 图 10 说明图 9 所示的共用处理器的改进实施例。共用处理器 500 包含用于与外部电路通信的处理器总线 P 总线 (PBUS) 505、输入逻辑 510、处理器锁存器 P 锁存器 (PLatch) 520 及输出逻辑 530。

[0126] 输入逻辑 510 从 P 总线接收数据且作为视经由信号线 411 来自堆栈总线控制器 410 的控制信号而处于逻辑状态“1”、“0”或“Z”(浮动)中的一者的经变换数据输出到 BSI 节点。设定 / 重设锁存器 P 锁存器 520 接着锁存 BSI,导致如 MTCH 及 MTCH* 的一对补充输出信号。

[0127] 输出逻辑 530 接收 MTCH 及 MTCH* 信号且在 P 总线 505 上输出视经由信号线 411 来自堆栈总线控制器 410 的控制信号而处于逻辑状态“1”、“0”或“Z”(浮动)中的一者的经变换数据。

[0128] 在任一时刻,共用处理器 500 处理与给定存储器单元相关的数据。举例来说,图 10 说明存储器单元耦合到位线 1 的情形。对应感测放大器 212-1 包含感测放大器数据出现的节点。在优选实施例中,节点采取存储数据的 SA 锁存器 214-1 的形式。类似地,对应组的数据锁存器 430-1 存储与耦合到位线 1 的存储器单元相关联的输入或输出数据。在优选实施例中,所述组的数据锁存器 430-1 包含充足数据锁存器 434-1,……,434-n 以存储 n 位的数据。

[0129] 当通过一对补充信号 SAP 及 SAN 而启用转移栅极 501 时,共用处理器 500 的 P 总线 505 经由 S 总线 422 可接近 SA 锁存器 214-1。类似地,当通过一对补充信号 DTP 及 DTN 而启用转移栅极 502 时,P 总线 505 经由 D 总线 423 可接近所述组数据锁存器 430-1。明确地将信号 SAP、SAN、DTP 及 DTN 说明为来自堆栈总线控制器 410 的控制信号的部分。

[0130] 图 11A 说明图 10 所示的共用处理器的输入逻辑的优选实施例。输入逻辑 520 在 P 总线 505 上接收数据且视控制信号而使得输出 BSI 为相同、反相或浮动的。输出 BSI 节点本质上受转移栅极 522 或包含串联到 Vdd 的 p 晶体管 524 及 525 的上拉电路或者包含串联接地的 n 晶体管 526 及 527 的下拉电路的输出的影响。上拉电路具有到 p 晶体管 524 及 525 的栅极,其分别由信号 P 总线及 ONE 控制。下拉电路具有到 n 晶体管 526 及 527 的栅极,其分别由信号 ONEB<1> 及 P 总线控制。

[0131] 图 11B 说明图 11A 的输入逻辑的真值表。由 P 总线及作为来自堆栈总线控制器 410 的控制信号的的控制信号 ONE、ONEB<0>、ONEB<1> 而控制逻辑。本质上,支持三个转移模式:通过 (PASSTHROUGH)、反相 (INVERTED) 及浮动 (FLOATED)。

[0132] 在 BSI 与输入数据相同的通过模式的情形下,信号 ONE 处于逻辑“1”,ONEB<0> 处于“0”且 ONEB<1> 处于“0”。此将停用上拉或下拉但将启用转移栅极 522 以在 P 总线 505 上将数据传递到输出 523。在 BSI 为输入数据的反相的反相模式的情形下,信号 ONE 处于“0”,ONEB<0> 处于“1”且 ONEB<1> 处于“1”。此将停用转移栅极 522。同样,当 P 总线处于“0”时,将停用下拉电路而启用上拉电路,此导致 BSI 处于“1”。类似地,当 P 总线处于“1”时,将停用上拉电路而启用下拉电路,此导致 BSI 处于“0”。最后,在浮动模式的情形下,可通过使得信号 ONE 处于“1”,ONEB<0> 处于“1”且 ONEB<1> 处于“0”而使输出 BSI 浮动。为了完整性而列出浮动模式,但在实际上不使用所述模式。

[0133] 图 12A 说明图 10 所示的共用处理器的输出逻辑的优选实施例。在处理器锁存器 P 锁存器 520 中锁存 BSI 节点处来自输入逻辑 520 的信号。输出逻辑 530 从 P 锁存器 520 的输出接收数据 MTCH 及 MTCH* 且视控制信号而在 P 总线上以通过、反相或浮动模式输出。换句话说,四个分支充当 P 总线 505 的驱动器,主动将其拉到高、低或浮动状态。此通过 P 总线 505 的四个分支电路(即两个上拉电路及两个下拉电路)而完成。第一上拉电路包含串联到 Vdd 的 p 晶体管 531 及 532,且能够在 MTCH 处于“0”时上拉 P 总线。第二上拉电路包含串联到接地的 p 晶体管 533 及 534,且能够在 MTCH 处于“1”时上拉 P 总线。类似地,第一下拉电路包含串联到 Vdd 的 n 晶体管 535 及 536,且能够在 MTCH 处于“0”时下拉 P 总线。第二上拉电路包含串联到接地的 n 晶体管 537 及 538,且能够在 MTCH 处于“1”时上拉 P 总线。

[0134] 本发明的一特征为以 PMOS 晶体管构成上拉电路且以 NMOS 晶体管构成下拉电路。由于通过 NMOS 的拉动远强于 PMOS 的拉动,因此在任何竞争中下拉将总是胜过上拉。换句话说,节点或总线可总是默认为上拉或“1”状态,且必要时可总通过下拉而倒转为“0”状态。

[0135] 图 12B 说明图 12A 的输出逻辑的真值表。通过从输入逻辑锁存的 MTCH、MTCH* 及作为来自堆栈总线控制器 410 的控制信号的的控制信号 PDIR、PINV、NDIR、NINV 而控制逻辑。支持四个操作模式:通过、反相、浮动及预充电。

[0136] 在浮动模式中,停用所有四个分支。此通过使信号 PINV = 1、NINV = 0、PDIR = 1、NDIR = 0(此也为默认值)而完成。在通过模式中,当 MTCH = 0 时,其将要求 P 总线 = 0。此通过仅启用具有 n 晶体管 535 及 536 的下拉分支(其中所有控制信号处于其默认值,除了 NDIR = 1)而完成。当 MTCH = 1 时,其将要求 P 总线 = 1。此通过仅启用具有 p 晶体管 533 及 534 的上拉分支(其中所有控制信号处于其默认值,除了 PINV = 0)而完成。在反相模式中,当 MTCH = 0 时,其将要求 P 总线 = 1。此通过仅启用具有 p 晶体管 531 及 532 的上拉分支(其中所有控制信号处于其默认值,除了 PDIR = 0)而完成。当 MTCH = 1 时,其将要求 P 总线 = 0。此通过仅启用具有 n 晶体管 537 及 538 的下拉分支(其中所有控制信号处于其默认值,除了 NINV = 1)而完成。在预充电模式中,PDIR = 0 及 PINV = 0 的控制信号设定将在 MTCH = 1 时启用具有 p 晶体管 531 及 531 的上拉分支或在 MTCH = 0 时启用具有 p 晶体管 533 及 534 的上拉分支。

[0137] 共用处理器操作在 2004 年 12 月 29 日的美国专利申请案号 11/026,536 中展开地

更为充分,所述申请案的全文以引用的方式并入本文中。

[0138] 数据锁存器在高速缓存操作中的使用

[0139] 本发明的若干方面利用上文于图 10 中描述的读取 / 写入堆栈的数据锁存器以用于高速缓存操作,所述执行将在内部存储器进行例如读取、写入或擦除的其它操作的同时输入及输出数据。在上文所述的架构中,若干物理页共享数据锁存器。举例来说,由于处于由所有字线所共享的位线的读取 / 写入堆栈上,因此当一操作进行时,如果所述锁存器中的任一者空闲,则其可高速缓存数据用于同一或另一字线中的将来的操作,从而节省转移时间(因为此可隐藏于另一操作之后)。此可通过增加对不同操作或操作的不同阶段的管线式操作的量而改进性能。在一实例中,在高速缓存编程操作中,当编程一页数据时,可加载另一页数据以节省转移时间。对于另一实例,在一示范性实施例中,将对一字线的读取操作插入到对另一字线的写入操作中,从而允许来自所述读取的数据在数据写入继续的同时转移出存储器。

[0140] 注意,此允许在写入或其它操作针对第一页数据进行的同时将来自同一块中但不同字线上的另一页的数据切出(以(例如)进行 ECC 操作)。对操作的此阶段间管线式操作允许数据转移所需的时间藏于对第一页数据的操作之后。更一般地,此允许将一个操作的一部分插入于另一(通常较长)操作的阶段之间。另一实例会将感测操作插入于(如)擦除操作的阶段之间,例如在擦除脉冲之前或在用作擦除的稍后部分的软编程阶段之前。

[0141] 为了论述操作中的一些所需的相对时间,可将用于上文所述的系统的一组示范性时间值取为:

[0142] 数据写入: $\sim 700 \mu s$ (下部页 $\sim 600 \mu s$, 上部页 $800 \mu s$)

[0143] 二进制数据写入: $\sim 200 \mu s$

[0144] 擦除: $\sim 2,500 \mu s$

[0145] 读取: $\sim 20-40 \mu s$

[0146] 读取及切出数据: 2KB 数据, $\sim 80 \mu s$; 4KB $\sim 160 \mu s$; 8KB $\sim 320 \mu s$

[0147] 所述值可用于参考以给出对下文的时序图所涉及的相对时间的概念。如果具有一具有不同阶段的较长操作,则主要方面将通过使用读取 / 写入堆栈的共享锁存器(如果锁存器可用)而插入于较快速的操作中。举例来说,可将读取插入于编程或擦除操作中,或者可将二进制编程插入于擦除中。主要示范性实施例将在编程操作期间针对一页切入及 / 或切出数据以用于另一页,所述页共享相同的读取写入堆栈,其中(例如)将对待切出并经修改的数据的读取插入于数据写入的验证阶段中。

[0148] 开放的数据锁存器的可用性可以若干方式而发生。一般来说,对于每单元存储 n 个位的存储器来说,对于每一位线将需要 n 个所述数据锁存器;然而,并非总是需要所有所述锁存器。举例来说,在以上部页 / 下部页的格式存储数据的每单元两位的存储器中,在编程下部页时将需要两个数据锁存器。更一般地,对于存储多个页的存储器来说,仅在编程最高页时将需要所有锁存器。此使得其它锁存器可用于高速缓存操作。此外,即使在写入最高页时,由于从写入操作的验证阶段移除各种状态,因此锁存器将释放。特定来说,一旦仅剩最高状态待验证,则仅需单一锁存器用于验证的目的且其它锁存器可用于高速缓存操作。

[0149] 以下论述将基于如并入于上文中的题为“数据锁存器在非易失性存储器的多阶段编程中的使用 (Use of Data Latches in Multi-Phase Programming of Non-Volatile

Memories)”的美国专利申请案中所描述的每单元存储两个位且具有针对每一位线上的数据的两个锁存器及用于快速通过写入的一额外锁存器的四状态存储器,所述申请案与本申请案同时申请。写入下部页或擦除或进行后擦除软编程的操作基本上为二进制操作且其中数据锁存器中的一者为空闲的,可使用其来高速缓存数据。类似地,在进行上部页或全序列写入时,一旦除最高层级的所有层级已经验证,则仅单一状态需验证且存储器可释放一锁存器,可使用所述锁存器来高速缓存数据。如何可使用此的一实例为在(例如于复制操作中)编程一页时,对共享同一组数据锁存器的另一页(例如同一组位线上的另一字线)的读取可在写入的验证阶段期间塞入。接着可将地址切换到正写入的页,从而允许写入处理在其停止之处拾起而无需重新开始。在写入继续的同时,在经内插的读取期间高速缓存的数据可经切出、检查或修改且转移返回以存在用于在一旦早先写入操作完成时即写回。此种类的高速缓存操作允许将对第二页数据的切出及修改藏于对第一页的编程之后。

[0150] 作为第一实例,用于二位存储器的高速缓存编程操作以单页(下部页/上部页的格式)编程模式而操作。图13为图10的简化版本,其展示在二位实施例中与当前论述相关的一些特定元件,去除其它元件以简化论述。所述包括连接数据I/O线231的数据锁存器DL0 434-0、通过线423而连接到共用处理器500的数据锁存器DL1 434-1、通过线435而与其它数据锁存器共同地连接的数据锁存器DL2 434-2以及通过线422而连接到共用处理器500的感测放大器数据锁存器DLS214。图13的各种元件根据其在对下部页的编程期间的部署而被标记。如题为“数据锁存器在非易失性存储器的多阶段编程中的使用(Use of Data Latches in Multi-Phase Programming of Non-Volatile Memories)”的与本申请案同时申请的美国专利申请案中所描述,锁存器DL2434-2用于快速通过写入模式中的下部验证(VL);对寄存器的包括以及在包括寄存器时对使用快速通过写入的包括为可选的,但示范性实施例将包括此寄存器。

[0151] 对下部页的编程可包括以下步骤:

[0152] (1) 处理以将数据锁存器DL0 434-0重设为默认值“1”而开始。此惯例用以简化部分页的编程,因为将禁止对所选中不待编程的单元进行编程。

[0153] (2) 沿I/O线231将编程数据供应到DL0 434-0。

[0154] (3) 编程数据将被转移到DL1 434-1及DL2 434-2(如果包括此锁存器且实施快速通过写入)。

[0155] (4) 一旦将编程数据转移到DL1 434-1,即可将数据锁存器DL0 434-0重设为“1”且在编程时间期间,可沿I/O线231将下一数据页加载到DL0 434-0,从允许在写入第一页的同时对第二页的高速缓存。

[0156] (5) 一旦将第一页加载到DL1 434-1中,编程即可开始。使用DL1 434-1数据以将单元从进一步编程锁定。如题为“数据锁存器在非易失性存储器的多阶段编程中的使用(Use of Data Latches in Multi-Phase Programming of Non-Volatile Memories)”的与本申请案同时申请的美国专利申请案中所描述,DL2 434-2数据用于管理向快速通过写入的第二阶段的转变的下部验证锁定。

[0157] (6) 一旦编程开始,在一编程脉冲之后,下部验证的结果即用以更新DL2434-2;较高验证的结果用以更新DL1 434-1。(此论述基于“常规”编码,其中下部页编程将达到A状态。题为“数据锁存器在非易失性存储器的多阶段编程中的使用(Use of Data Latches in

Multi-Phase Programming of Non-Volatile Memories)”的与本申请案同时申请的美国专利申请案及于 2005 年 3 月 16 日申请的题为“具有功率节省读取及编程验证操作的非易失性存储器及方法 (Non-Volatile Memory and Method with Power-Saving Read and Program-Verify Operations)”的美国专利申请案进一步论述了此及其它编码。当前论述向其它编码的扩展易于随后产生)。

[0158] (7) 在确定编程是否完成的过程中,仅检查行的单元的 DL1434-1 寄存器(或编程的适当物理单位)。

[0159] 一旦写入下部页,则可对上部页进行编程。图 14 展示与图 13 相同的元件,但指示对于其中读入下部页数据的上部页编程的锁存器指派。(所述描述再次使用常规编码,使得上部页的编程将达到 B 及 C 状态)。对上部页的编程可包括以下步骤:

[0160] (1) 一旦下部页结束编程,即以来自状态机控制器的信号而开始上部页(或下一页)写入,其中(未执行的)高速缓存编程命令得以保存。

[0161] (2) 编程数据将被从 DL0 434-0(在步骤(3)中下部页写入期间将数据加载到 DL0434-0)转移到 DL1 434-1 及 DL2 434-2。

[0162] (3) 将从阵列读入下部页数据且将其放置于 DL0 434-0 中。

[0163] (4) DL1 434-1 及 DL2 434-2 再次分别用于验证高及验证低锁定数据。锁存器 DL0434-0(保持下部页数据)作为编程参考数据而经检查,但并不以验证结果对其加以更新。

[0164] (5) 作为验证 B 状态的部分,在于下部验证 VBL 处感测之后,将在 DL2 434-2 中相应地更新数据,同时通过高验证 VBH 结果而更新 DL1 434-1 数据。类似地,C 验证将具有对应命令以通过相应 VCL 及 VCH 结果来更新锁存器 DL2 434-2 及 DL1 434-1。

[0165] (6) 一旦 B 数据完成,则不需要下部页数据(保持于 DL0 434-0 中用于参考),因为仅需执行对 C 状态的验证。将 DL0 434-0 重设为“1”且可从 I/O 线 231 加载另一页的编程数据且在锁存器 DL0 434-0 中对其进行高速缓存。共用处理器 500 可设定仅 C 状态待验证的指示。

[0166] (7) 在确定上部页编程是否完成的过程中,对于 B 状态检查锁存器 DL1 434-1 及 DL0434-0 两者。一旦将单元编程为 B 状态且仅验证 C 状态,则仅需检查锁存器 DL1 434-1 数据以观察是否存在未经编程的任何位。

[0167] 注意,在此布置下,在步骤 6 中,不再需要锁存器 DL0 434-0 且其可用以高速缓存数据以进行下一编程操作。另外,在使用快速通过写入的实施例中,一旦进入第二缓慢编程阶段,即也可使得锁存器 DL2 434-2 可用于高速缓存数据,但在实际上,实际情形常为此仅可以此方式可用相当短而无法证明经常需要以实施此特征的额外开销是合理的时间周期。

[0168] 图 15 可用以说明上几幅图中已描述的以单页模式进行的高速缓存编程的许多方面。图 15 展示存储器内部事件发生(下部“真实忙碌”线)与从存储器外部观察(上部“高速缓存忙碌”线)的相对时序。

[0169] 在时间 t_0 处,将待编程到所选字线(WLn)上的下部页加载到存储器中。此假定先前未曾高速缓存第一下部页的数据,因为其将用于后续页。在时间 t_1 处,完成下部页加载且存储器开始写入下部页。由于这在此点上等效于二进制操作,因此仅需验证状态 A(“pvfyA”)且数据锁存器 DL0 434-0 可用于接收下一页数据,此处将下一页数据取作待

在时间 t_2 处经编程到 WLn 中的上部页,其因此在对下部页的编程期间于锁存器 $DL0\ 434-0$ 中经高速缓存。上部页在时间 t_3 处完成加载且可在下部页于 t_4 处一结束时即得以编程。在此布置下,虽然数据的全部(下部及上部页)待写入编程的物理单位(此处为字线 WLn)中,但是存储器必须从时间 t_3 等待到时间 t_4 才可写入上部页数据,此不同于下文描述的全序列实施例。

[0170] 对上部页的编程开始于时间 t_4 ,其中最初仅验证 B 状态(“pvfyB”),在 t_5 处添加 C 状态(“pvfyB/C”)。一旦于 t_6 处不再验证 B 状态,则仅 C 状态需经验证(“pvfyC”)且锁存器 $DL0\ 434-0$ 被释放。此允许在上部页完成编程的同时高速缓存下一数据组。

[0171] 如所提到的,根据如图 15 所示的关于高速缓存编程的单页算法,即使上部页数据可在时间 t_3 处可用,存储器仍将在开始写入此数据之前等待直到时间 t_4 。在向全序列编程操作的转换(例如在美国专利申请案 11/013,125 中更为充分展开的转换)中,一旦上部页可用,上部及下部页数据即可同时经编程。

[0172] 用于全序列(低到全的转换)写入中的高速缓存编程的算法如同上文而以下部页编程开始。因此,步骤(1)到(4)关于以单页编程模式进行的下部页处理:

[0173] (1) 处理以将数据锁存器 $DL0\ 434-0$ 重设为默认值“1”而开始。此惯例用以简化部分页的编程,因为将禁止对所选中不待编程的单元进行编程。

[0174] (2) 沿 I/O 线 231 将编程数据供应到 $DL0\ 434-0$ 。

[0175] (3) 编程数据将被转移到 $DL1\ 434-1$ 及 $DL2\ 434-2$ (如果包括此锁存器且实施快速通过写入)。

[0176] (4) 一旦将编程数据转移到 $DL1\ 434-1$,即可将数据锁存器 $DL0\ 434-0$ 重设为“1”且在编程时间期间,可沿 I/O 线 231 将下一数据页加载到 $DL0\ 434-0$,从而允许在写入第一页的同时对第二页的高速缓存。

[0177] 一旦加载第二页数据,则如果对应于正写入的下部页的上部且下部页尚未结束编程,则可实施向全序列写入的转换。此论述集中于数据锁存器在所述算法中的使用,其中许多其它细节较充分地展开于共同待决、共同转让的美国专利申请案 11/013,125 中。

[0178] (5) 在将上部页数据加载到锁存器 $DL0\ 434-0$ 中之后,将在地址块中进行判断以检查 2 页是否在同一字线及同一块上,其中一页为下部页且一页为上部页。如果是,则编程状态机将触发下部页编程向全序列编程的转换(如果此为允许的)。在所有未决验证完成之后,接着实现转变。

[0179] (6) 在编程序列从下部页改变为全序列时通常将改变一些操作参数。在示范性实施例中,所述参数包括:

[0180] (i) 如果下部页数据尚未经锁定,则对于脉冲验证循环的数目的最大编程循环将从下部页算法的最大编程循环改变为全序列的最大编程循环,但已完成的编程循环的数目将不由转换重设。

[0181] (ii) 如图 16 所示,编程波形以用于下部页编程处理中的值 $VPGM_L$ 开始。如果编程波形已前进到其超过用于上部页处理中的开始值 $VPGM_U$ 之处,则在向全序列转换时,在使阶梯继续上升之前,阶梯将降回到 $VPGM_U$ 。

[0182] (iii) 确定编程脉冲的步长大小及最大值的参数不改变。

[0183] (7) 应执行对存储器单元的当前状态的全序列读取以保证将编程正确数据用于多

级编码。此确保可能之前在下部页编程中已经锁定但需要进一步编程以计及上部页数据的状态在全序列开始时不被禁止编程。

[0184] (8) 如果启动快速通过写入,则将同样更新锁存器 DL2 434-2 的数据以反映上部页编程数据,因为锁存器 DL2 434-2 的数据之前基于仅关于 A 状态的下部验证。

[0185] (9) 编程接着以多级、全序列编程算法而恢复。如图 16 所示,如果下部页处理中的编程波形已增加超过上部页开始电平,则在转换时波形逐步后退到此电平。

[0186] 图 17 为下部页向全序列转换写入处理中所涉及的相对时间的示意性表示。直到时间 t_3 ,处理与上文关于图 15 中的处理所描述的相同。在已加载上部页的数据且进行向全序列算法的转变的 t_3 处,切换验证处理以包括 B 状态连同 A 状态。一旦锁定 A 状态的全部,验证处理即在时间 t_4 处切换为检查 B 及 C 状态。一旦于 t_5 处已验证 B 状态,则仅 C 状态需检查且寄存器可被释放以加载待编程的下一数据,例如如在高速缓存忙碌线上所指示的下一字线 (WL_{n+1}) 上的下部页。在时间 t_6 处,已高速缓存此下一数据组且一旦对先前组的 C 数据的编程于 t_7 处结束,此下一数据组即开始编程。另外,在(此处)字线 WL_{n+1} 上的下部页正编程的同时,可将下一数据(例如对应的上部页数据)加载到开放的锁存器 DL0 434-0 中。

[0187] 在全序列写入期间,以独立给出下部页与上部页状态的方式实施状态报告。在编程序列的结尾,如果存在未完成的位,则可执行对物理页的扫描。第一扫描可检查锁存器 DL0 434-0 以获得未完成的上部页数据,第二扫描可检查 DL1 434-1 以获得未完成的下部页数据。由于对 B 状态的验证将改变 DL0 434-0 及 DL1 434-1 数据,因此应以如果位的阈值高于 A 验证电平则 DL1434-1 数据“0”将改变为“1”的方式而执行 A 状态验证。此后验证将检查是否存在任何编程不足的 B 电平在 A 电平通过;如果其在 A 电平通过,则误差仅存在于上部页而不存在于下部页上;如果其在 A 电平未通过,则下部页及上部页均具有误差。

[0188] 如果使用高速缓存编程算法,则在编程 A 及 B 数据之后,C 状态将经转移到锁存器 DL1 434-1 以完成编程。在此情形下,对锁存器的扫描对于下部页不必要,因为下部页将已通过编程而无任何不合格位。

[0189] 本发明的另一组示范性实施例关于页复制操作,其中将数据组从一个位置再定位到另一位置。2004 年 5 月 13 日申请的美国专利申请案第 US10/846,289 号;2004 年 12 月 21 日申请的第 11/022,462 号;及 2004 年 8 月 9 日申请的第 US10/915,039 号;以及美国专利第 6,266,273 号中描述了数据再定位操作的各种方面,所述专利的全部以引用的方式并入本文中,所述专利的全部以引用的方式并入本文中。当将数据从一位置复制到另一者时,常将数据切出以对其进行检查(以(例如)寻找误差)、更新(例如更新标头)或两者(例如校正所检测到的误差)。所述转移还为了在垃圾收集操作中使日期固定。本发明的主要方面允许在写入操作的验证阶段期间内插对开放寄存器的数据读取,其中接着随着写入操作继续而将此经高速缓存的数据转移出存储器装置,从而允许用于切出数据的时间藏于写入操作之后。

[0190] 下文存在高速缓存页复制操作的两个示范性实施例。在两种情形下,均描述使用快速通过写入实施方案的实施方案。图 18 指示随着处理进行的锁存器的示范性布置的部署。

[0191] 高速缓存页复制的第一版本将写入到下部页且可包括以下步骤,其中将读取地址

标记为 M、M+1..... 且将写入地址标记为 N、N+1..... :

[0192] (1) 将待复制的页 (“页 M”) 读取到锁存器 DL1434-1 中。此可为上部页或下部页的数据。

[0193] (2) 接着将页 M 转移到 DL0434-0 中。

[0194] (3) 接着切出 DL0 434-0 中的数据且对其进行修改, 在此之后将其转移回锁存器中。

[0195] (4) 编程序列接着可开始。在将待写入下部页 N 中的数据转移到 DL1 434-1 及 DL2434-2 之后, 锁存器 DL0 434-0 准备好高速缓存数据。将对此下部页进行编程。对于此实施例, 编程状态机将停于此处。

[0196] (5) 接着将待复制的下一页读取到 DL0 434-0 中。接着编程可恢复。在步骤 (4) 的结尾停止的状态机将从开始重新开始编程序列。

[0197] (6) 编程继续直到下部页结束。

[0198] 复制目的页地址将确定写入是到下部页还是上部页。如果编程地址为上部页地址, 则编程序列将不停止直到编程结束且将在写入完成之后执行步骤 (5) 的读取。

[0199] 在第二高速缓存页复制方法中, 可暂停编程 / 验证处理以插入读取操作且接着重新开始写入操作 (在其停止之处拾起)。接着可切出在此交错的感测操作期间读取的数据, 同时经恢复的写入操作继续。同样, 此第二处理允许一旦仅 C 状态正被验证且每一位线上的一个锁存器开放, 即在上部页或全序列写入处理中使用页复制机制。第二高速缓存页复制操作以与第一情形中相同的前三个步骤开始, 但接着不同。其可包括以下步骤:

[0200] (1) 将待复制的页 (“页 M”) 读取到锁存器 DL1 434-1 中。此可为下部或上部页。

[0201] (2) 接着将来自页 M 的数据转移到 DL0 434-0 中。(如同之前一样, N 等等将表示写入地址, M 等等用于读取地址)。

[0202] (3) 接着切出 DL0 434-0 中的数据且对其进行修改, 且接着将其转移回锁存器。

[0203] (4) 状态机编程将进入无限等待状态直到输入命令 (读取命令) 且接着到锁存器 DL0434-0 的对另一页 (如下一页 M+1) 的读取将开始。

[0204] (5) 一旦步骤 (4) 的读取完成, 即将地址切换回字线及块地址以将步骤 (1 到 3) 中的数据编程到页 N (此处为下部页) 且编程得以恢复。

[0205] (6) 在对页 M+1 的读取结束之后, 可切出数据, 对其进行修改且将其返回。如果两页为同一 WL 上的对应的上部页及下部页, 则一旦处理完成, 即可将写入转换为全序列操作。

[0206] (7) 如在早先所述的正常高速缓存编程中一样, 一旦在全序列写入中完成 A 及 B 电平, 即将 DL0 434-0 中的数据转移到 DL1 434-1, 且可发出对于另一页 (例如, 页 M+2) 的读取命令。如果不存在单页到全序列的转换, 则下部页将完成写入且接着上部页将开始。在完全完成 B 电平状态之后, 相同的 DL0 434-0 到 DL1 434-1 数据转移将发生, 且状态机将进入等待对于页 M+2 的读取命令的状态。

[0207] (8) 一旦读取命令到达, 即将地址切换到读取地址且读出下一页 (页 M+2)。

[0208] (9) 一旦读取完成, 即将地址切换回先前的上部页地址 (编程地址 N+1) 直到写入完成。

[0209] 如上文所提到的, 示范性实施例除了包括用于保持可经编程到存储器单元中的每

一者中的（此处，2 位）数据的锁存器 DL0 434-0 及 DL1 434-1 之外还包括用于快速通过写入技术的下部验证的锁存器 DL2 434-2。一旦通过下部验证，即锁存器 DL2 434-2 还可被释放且用以高速缓存数据，但此在示范性实施例中未进行。

[0210] 图 19A 及图 19B 说明第二高速缓存页复制方法的相对时序，其中图 19B 说明具有全序列写入转换的算法且图 19A 说明不具有全序列写入转换的算法。（图 19A 及图 19B 均由两个部分构成：开始于对应于 t_0 的断续竖直线 A 处且以对应于 t_5 的断续竖直线 B 结束的第一上部部分；为上部部分的延续且以对应于 t_5 的断续竖直线 B 开始的第二下部部分。在两种情形中，时间 t_5 处的线 B 在上部部分中与在下部部分中相同，两部分中仅存在一接缝以允许将其显示于两条线上）。

[0211] 图 19A 展示一过程，其以读取在此实例中取作下部页的第一页（页 M）而开始，假定先前尚未高速缓存数据，且以单页模式操作，在开始写入上部页之前等待直到下部页结束写入。过程以时间 t_0 处对页 M 的读取（感测页 M(L)）而开始，页 M 在此处为由此编码中的 A 及 C 电平处的读取而感测的下部页。在时间 t_1 处读取完成且可将页 M 切出且对其进行检查或修改。开始于时间 t_2 ，通过于 B 电平的读取而感测下一页（此处为页 M+1，对应于与下部页 M 相同的物理的上部页），其为结束于时间 t_3 的过程。在此点上，第一页（来源于页 M）（下部）准备好被编程返回到存储器中页 N 处，且从页 M+1 读取的数据保持于锁存器中且可被转移出以受到修改 / 检查。所述过程中的两者均可开始于同一时间，在此处为 t_3 。通过使用上文所述的典型时间值，到时间 t_4 为止已切出来自页 M+1 的数据且已对其进行修改；然而，对于未实施全序列转换的实施例来说，存储器将等待直到页 N 于时间 t_5 处结束以开始将第二读取页的数据（来源于页 M+1）写入到页 N+1 中。

[0212] 由于页 N+1 为上部页，因此其写入最初以 B 电平处的验证而开始，在时间 t_6 处添加 C 电平。一旦存储元件于时间 t_7 处使目标状态 B 全部锁定（或者达到最大计数），即撤销 B 状态验证。如上文所述，根据本发明的若干主要方面，此允许数据锁存器被释放，中止正在进行的写入操作，插入读取操作（在与经中止的编程 / 验证操作不同的地址处），写入接着在其停止之处恢复，且可在经恢复的写入操作继续的同时将经插入的写入操作期间所感测的数据切出。

[0213] 在时间 t_7 处针对（此处）下部页 M+2 而执行经插入的写入操作。此感测结束于时间 t_8 ，且页 N+1 的写入重新拾起，且来自页 M+2 的数据同时经切出及修改。在此实例中，页 N+1 在页 M+2 结束于时间 t_{10} 之前在时间 t_9 结束编程。在时间 t_{10} 处，源自页 M+2 的数据的写入可开始；然而，在此实施例中，替代地，首先执行页 M+3 的读取，此允许将此页的数据切出及修改藏于开始于时间 t_{11} 处的将源自页 M+2 的数据写入页 N+2 中之后。过程接着如图式的早先部分中而继续，但页码移位，其中时间 t_{11} 对应于时间 t_3 ，时间 t_{12} 对应于时间 t_4 等等，直到复制过程停止。

[0214] 图 19B 再次展示以读取下部页（取作下部页的页 M）而开始且假定先前未高速缓存数据的过程。图 19B 不同于图 19A 在于其于时间 t_4 实施向全序列写入的转换。此一般说来将过程加速了图 19A 的时间（ $t_5 - t_4$ ）。在时间 t_4 （= 图 19A 中的 t_5 ）处，如先前所述而实施与全序列转换相关的各种改变。其它方面，过程类似于图 19A 的过程，包括在时间 t_7 与 t_{12} 之间的本发明的所述方面。

[0215] 在页复制过程及此处描述的涉及写入数据的其它技术中，可遵循以引用的方式并

入本文中的美国专利公开案号 US-2004-0109362-A1 中描述的方法而明智地选择在给定时间验证的状态。举例来说,在全序列写入中,写入过程可开始仅验证 A 电平。在 A 验证之后,对其进行检查以观察是否存在已通过的任何位。如果是,则向验证阶段添加 B 电平。将在所有存储单位以 A 电平验证作为其目标值验证(或除了基于可设定参数的最大计数)之后将 A 电平验证移除。类似地,在 B 电平处的验证之后可添加 C 电平的验证,其中将在所有存储单位以 B 电平验证作为其目标值验证(或除了基于可设定参数的最大计数)之后将 B 电平验证移除。

[0216] 编程操作期间数据锁存器中的高速缓存操作

[0217] 关于优选多状态编码而描述具有用于其它操作的后台数据高速缓存的编程操作。

[0218] 对于 4 状态存储器的示范性优选“LM”编码

[0219] 图 20A 到图 20E 说明对于以 2 位逻辑代码(“LM”代码)编码的 4 状态存储器的编程及读取。此代码提供容错性且减轻归因于 Yupin 效应的相邻单元耦合。图 20A 说明在每一存储器单元使用 LM 代码存储两个的数据位时 4 状态存储器阵列的阈值电压分布。LM 编码不同于常规格雷码(Gray code)在于上部及下部位对于状态“A”及“C”反转。“LM”代码已揭示于美国专利第 6,657,891 号中且优点在于通过避免需要电荷的较大改变的编程操作而减少邻近浮动栅极之间的场效应耦合。如将于图 20B 及图 20C 中所见,每一编程操作导致电荷存储单位中的电荷的适度改变(如从阈值电压 V_T 的适度改变所显而易见)。

[0220] 对编码进行设计以使得 2 个位(下部及上部)可单独经编程及读取。当编程下部位时,单元的阈值电平保持于未经编程的区域中或移动到阈值窗口的“中下”区域。当编程上部位时,在所述两个区域中的任一者中的阈值电平进一步前进到稍高(不多于阈值窗口的四分之一)的电平。

[0221] 图 20B 说明使用 LM 代码在现有 2 循环编程方案中进行的下部页编程。容错 LM 代码本质上避免任何上部页编程转变越过任何中间状态。因此,第一循环下部页编程使得逻辑状态 (1,1) 转变为某一中间状态 (x,0),如由将“未经编程”的存储器状态“U”编程为以 (x,0) 表示的具有在大于 D_A 但小于 D_C 的宽广分布中的编程阈值电压的“中间”状态所表示。在编程期间,相对于界线 DV_A 而验证中间状态。

[0222] 图 20C 说明使用 LM 代码在现有 2 循环编程方案中进行的上部页编程。在将上部页位编程为“0”的第二循环中,如果下部页位处于“1”,则逻辑状态 (1,1) 转变为 (0,1),如由将“未经编程”的存储器状态“U”编程为“A”所表示。在编程为“A”期间,验证是关于 DV_A 。如果下部页位处于“0”,则通过从“中间”状态编程为“B”而获得逻辑状态 (0,0)。编程验证是关于界线 DV_B 。类似地,如果上部页将保持于“1”,而下部页已经编程为“0”,则其将需要从“中间”状态向 (1,0) 的转变,如由将“中间”状态编程为“C”所表示。编程验证是关于界线 DV_C 。由于上部页编程仅涉及向下一邻近存储器状态的编程,因此从一循环到另一循环无大量电荷改变。设计从“U”到大致“中间”状态的下部页编程以节省时间。

[0223] 在优选实施例,实施在较早章节中所提到的“快速通过写入”编程技术。举例来说,在图 20C 中,最初编程验证 (“pvfyA_L”) 关于经设定于低于 DV_A 的边缘处的 DV_{AL} 。一旦对单元进行于 DV_{AL} 处的编程验证,则后续编程将以较精细的步长而进行且编程验证 (pvfyA) 将关于 DV_A 。因此在编程操作期间必须锁存额外转变态 A_{LOW} 以指示已对单元进行关于 D_{AL} 的编程验证。类似地,如果实施 QPW 以编程为“B”状态,则将存在额外转变态 B_{LOW} 待锁存。对于

B_{LOW} 的编程验证将关于界线 DV_{BL} 且对于“B”的编程验证将关于界线 DV_B 。在处于 A_{LOW} 或 B_{LOW} 状态中时,对所讨论存储器单元的编程将通过位线电压的适当偏置或通过修改编程脉冲而被切换到较缓慢(即,较精细)的模式。以此方式,最初可使用较大编程步长以用于在无超出目标状态的危险的情况下快速收敛。2005 年 12 月 29 日申请且题为“用于非易失性存储器中的改进编程验证操作的方法(Methods for Improved Program-Verify Operations in Non-Volatile Memories)”的美国专利申请案序号 11/323,596 中已揭示“QPW”编程算法,该专利全部揭示内容以引用的方式并入本文中。

[0224] 图 20D 说明辨别以 LM 代码编码的 4 状态存储器的下部位所需的读取操作。解码将视是否已对上部页进行编程而定。如果已对上部页进行编程,则读取下部页将需要关于划界阈值电压 D_B 的读取 B 的一遍读取。另一方面,如果尚未对上部页进行编程,则将下部页编程为“中间”状态(图 20B),且读取 B 将引起误差。确切地,读取下部页将需要关于划界阈值电压 D_A 的读取 A 的一遍读取。为了分辨两种情形,在上部页被编程时在上部页中(通常在开销或系统区中)写入旗标(“LM”旗标)。在读取期间,将首先假定已对上部页进行编程且因此将执行读取 B 操作。如果 LM 旗标被读取,则假定正确且完成读取操作。另一方面,如果第一读取未产生旗标,则其将指示尚未对上部页进行编程且因此需通过读取 A 操作而读取下部页。

[0225] 图 20E 说明辨别以 LM 代码编码的 4 状态存储器的上部位所需的读取操作。如从图式清楚可见,上部页读取将需要读取 A 及读取 C 的 2 遍读取,其分别关于划界阈值电压 D_A 及 D_C 。类似地,如果尚未对上部页进行编程,则还可通过“中间”状态干扰上部页的解码。再一次,LM 旗标将指示是否已对上部页进行编程。如果尚未对上部页进行编程,则读取数据将被重设为“1”而指示未对上部页数据进行编程。

[0226] 以 LM 代码及 QPW 进行的编程操作期间的锁存器利用

[0227] 如图 10 所示,每一位线允许读取/写入模块沿存储器阵列的所选行而存取给定存储器单元。存在于一行的存储器单元的一页上并行操作的 P 个读取/写入模块的页。每一读取/写入模块包含耦合到共用处理器 500 的感测放大器 212-1 及数据锁存器 430-1。感测放大器 212-1 经由位线感测存储器单元的传导电流。数据由共用处理器 500 处理且存储于数据锁存器 430-1 中。通过耦合到数据锁存器的 I/O 总线 231(见图 13 及图 14)而实现存储器阵列外部的数据交换。在优选架构中,由沿一行的一游程的 p 个邻接存储器单元形成页,所述存储器单元共享相同字线且可由存储器阵列的 p 个邻接位线存取。在替代架构中,通过沿一行的偶数或奇数存储器单元而形成页。以足以执行各种所需存储器操作的最少 n 个锁存器 DL1 到 DLn 而实施数据锁存器 430-1。图 13 及图 14 说明 4 状态存储器的优选配置,其中存在三个锁存器 DL0 到 DL2。

[0228] 当前页编程期间的下一页编程数据加载

[0229] 图 21 为下部页编程的示意性时序图,其说明将下一页编程数据加载未使用的数据锁存器中的后台操作。同时展示主机、I/O 总线、数据锁存器及存储器核心的活动。图 20B 中说明以 LM 代码进行的下部页编程,其中将擦除或未经编程的状态 (1,1) 编程为“中下”或中间状态 (X,0)。在此情形下,一位(即,下部位)将足以在未经编程的“1”状态与中间“0”状态之间进行区分。举例来说,DL2(见图 13 及图 14)可用以存储下部位。

[0230] 在第 N 页数据待写入时,主机最初向存储器发布写入命令以将所述页数据写入到

指定地址。此后为将待经编程的所述页数据发送到存储器。经由 I/O 总线将编程数据切入且将其锁存到每一读取 / 写入模块的 DL2 中。因此, I/O 总线在此切入周期(例如可具有 300 μ s 的持续时间)期间暂时忙碌。

[0231] 下部页编程为二进制的且仅需在如通过 DV_A 阈值电平划界的“U”状态与“中间状态”(见图 20B)之间区分。施加到字线的每一编程脉冲由读回或编程验证跟随以确定单元是否已达到表示编程数据的目标状态。在此情形下,编程验证为关于 DV_A 的 (“pvfyA”)。因此仅需要来自每一读取 / 写入模块的一锁存器以存储每一单元的一位。

[0232] 关于数据锁存器,含有编程数据的 DL2 有效地用于发生于存储器阵列或存储器核心中的当前下部位编程操作。因此,正由核心使用的锁存器的数目为一,而另两个锁存器(即 DL0 及 DL1)仍为闲置的。

[0233] 在核心处的编程继续的同时,两个闲置的锁存器及空闲的 I/O 总线可用于设立下一页编程数据。主机可发布另一命令以写入第 N+1 页数据且经由 I/O 总线切换数据以锁存于两个空闲的锁存器中的一者(如 DL0)中。以此方式,一旦核心完成编程第 N 页,其即可开始对第 N+1 页进行编程而无需等待另一 300 μ s 而使数据切入。

[0234] 在此点处,已使用两个锁存器(例如,DL2 及 DL0),一者用于正在进行的对第 N 页(下部页)的编程且一者用于高速缓存第 N+1 页的编程数据。因此,多出一锁存器为空闲的,但对其的利用将视已经高速缓存的第 N+1 页为上部页还是下部页而定。

[0235] 如果第 N+1 页为通常属于相同页单元或字线的上部页,则在优选实施例中,必须保留最后空闲的锁存器以最佳化上部页的后续编程。这是由于“快速通过写入”(“QPW”)编程算法(在早先章节中提及)的实施要求额外锁存器以存储旗标来指示是否已将单元编程为接近于目标状态。

[0236] 如果第 N+1 页为属于单元或字线的另一页的另一下部页,则可视情况使用最后空闲的锁存器以在主机提出的情况下对另一第 N+2(下部或上部)页数据进行高速缓存。

[0237] 图 22 为展示在使用 QWP 的 4 状态上部页或全序列编程的各种阶段期间需跟踪的状态的数目的表。图 20C 中说明以 LM 代码进行的上部页或全序列编程,其中分别将下部页状态“U”或 (1,1) 中的一些及“中间”状态 (X,0) 进一步编程为状态“A”或 (0,1)、“B”或 (0,0) 及“C”或 (1,0)。具体来说,状态“A”是由“U”编程而来且状态“B”及“C”是由“中间”编程而来。在实施 QWP 技术用于状态“A”及“B”但非“C”的情况下,编程最初需要在总计共五个状态的基本状态“A”、“B”及“C”加上“ALOW”及“B_{LOW}”之间进行区分。在三个位处于三个锁存器的情况中,存在 2^3 或九个可能代码,其对于在所述六个状态之间进行区分来说是绰绰有余。

[0238] 编程期间的若干阶段可随编程前进而出现

[0239] “A”完成——在已关于 D_A 界线而编程验证目标为“A”状态的页中的所有单元之后。此将需要首先完成关于 D_{AL} 界线的编程验证。存在四个状态“L”(编程锁定)、“B_L”、“B”及“C”需留意。此将需要以两位代码表 2CT (“A”)提供的预定编码而存储两个位的两个锁存器。

[0240] “B”完成——在已关于 D_B 界线而编程验证目标为“B”状态的页中的所有单元之后。此将需要首先完成关于 D_{BL} 界线的编程验证。存在四个状态“L”、“A_L”、“A”及“C”需留意。此将需要以两位代码表 2CT (“B”)提供的预定编码而存储两个位的两个锁存器。

[0241] “C”完成——在已关于 D_C 界线而编程验证目标为“C”状态的页中的所有单元之后。存在五个状态“L”、“ A_L ”、“A”、“ B_L ”及“B”需留意。此将需要以三位代码表 3CT (“C”) 提供的预定编码而存储三个位的三个锁存器。

[0242] “A”+ “B”完成——在已分别关于 D_A 界线及 D_B 界线而编程验证目标为“A”状态及“B”状态的页中的所有单元之后。存在两个状态“L”及“C”需留意。此将需要以一位代码表 1CT (“A”+ “B”) 提供的预定编码而存储一位的一锁存器。

[0243] “A”+ “C”完成——在已分别关于 D_A 界线及 D_C 界线而编程验证目标为“A”状态及“C”状态的页中的所有单元之后。存在三个状态“L”、“ B_L ”及“B”需留意。此将需要以两位代码表 2CT (“A”+ “C”) 提供的预定编码而存储两个位的两个锁存器。

[0244] “B”+ “C”完成——在已分别关于 D_B 界线及 D_C 界线而编程验证目标为“B”状态及“C”状态的页中的所有单元之后。存在三个状态“L”、“ A_L ”及“A”需留意。此将需要以两位代码表 2CT (“B”+ “C”) 提供的预定编码而存储两个位的两个锁存器。

[0245] “A”+ “B”+ “C”完成——在已分别关于 D_A 界线、 D_B 界线及 D_C 界线而编程验证目标为“A”状态、“B”状态及“C”状态的页中的所有单元之后。已编程验证页的所有目标状态且完成对所述页的编程。将不需要锁存器。

[0246] 图 23 为上部页或全序列编程的示意性时序图,其说明将下一页编程数据加载到未使用的数据锁存器中的后台操作。同时展示主机、I/O 总线、数据锁存器及存储器核心的活动。

[0247] 当上部页数据的第 N 页待写入时,必须参考先前编程的下部页数据。先前编程的下部页已锁存于每一读取 / 写入模块的 DL2 中。关于上部页数据的第 N 页,主机最初向存储器发布写入命令以将所述页数据写入到指定地址。此后为将待经编程的所述页数据发送到存储器。经由 I/O 总线将编程数据切入且将其锁存到每一读取 / 写入模块的 DL0 中。因此, I/O 总线在此切入周期 (例如可具有 $300 \mu s$ 的持续时间) 期间暂时忙碌。

[0248] 上部页或全序列编程为多状态的,其中状态“A”、“B”及“C”分别由 D_A 、 D_B 及 D_C 划界 (见图 20C)。施加到字线的每一编程脉冲由读回或编程验证跟随以确定单元是否已达到表示编程数据的目标状态。

[0249] 如图 22 中所示,在编程期间需要的锁存器的数目关于编程已进行到何阶段而变化。举例来说,最初使用所有三个锁存器。当已编程验证所有“A”状态 (“A”完成) 时,在后续编程期间存储器核心仅需要两个锁存器 (例如,DL2 及 DL1) 以存储四个可能状态。此使得一锁存器 (例如,DL0) 空闲以用于高速缓存操作。

[0250] 在核心处的编程继续的同时,空闲的锁存器及空闲的 I/O 总线可用于设立下一页编程数据。主机可发布另一命令以写入第 N+1 页数据 (下部页数据) 且经由 I/O 总线切换数据以锁存于空闲的锁存器 DL0 中。以此方式,一旦核心完成编程第 N 页,其即可开始对第 N+1 页进行编程而无需等待另一 $300 \mu s$ 而使数据切入。将相同考虑应用于如图 22 所示存在至少一空闲的锁存器的其它编程阶段中。

[0251] 另一可能性为当编程进入仅需一锁存器以进行操作且因此具有两个空闲的锁存器用于高速缓存操作的阶段时。举例来说,如图 22 所示,此发生于已编程验证“A”及“B”状态两者时。在此点处,两个锁存器可用。如果为了加载 (N+1) 下部页数据而用尽一锁存器,则剩余一者可用以加载 (N+2) 上部或下部页数据。

[0252] 如果第 N+1 页为通常属于相同页单元或字线的上部页,则在优选实施例中,必须保留最后空闲的锁存器以最佳化上部页的后续编程。这是由于“快速通过写入”(“QPW”)编程算法(在早先章节中提及)的实施要求额外锁存器以存储一或两个旗标来指示是否已将单元编程为接近于目标状态。

[0253] 如果第 N+1 页为属于单元或字线的另一页的另一下部页,则可视情况使用最后空闲的锁存器以在主机提出的情况下对另一第 N+2(下部或上部)页数据进行高速缓存。

[0254] 根据本发明的一方面,当写入操作的多个阶段关于待跟踪的状态的数目而变化时,阶段相关的编码启用对可用数据锁存器的有效利用,从而允许大量的剩余锁存器用于后台高速缓存操作。

[0255] 图 24 为说明根据本发明的一般实施例的与当前多阶段存储器操作同时发生的锁存器操作的流程图。

[0256] 步骤 600 :开始操作具有一具有可寻址的存储器单元页的存储器阵列的存储器。

[0257] 步骤 610 :向经寻址的页的每一存储器单元提供一组具有锁存预定数目的位的能力的数据锁存器。

[0258] 存储器阵列中的当前多阶段存储器操作

[0259] 步骤 620 :对存储器阵列执行当前存储器操作,所述存储器操作具有一个或一个以上阶段,每一阶段与一组预定操作状态相关联。

[0260] 通过有效的阶段相关的编码而释放锁存器

[0261] 步骤 622 :为每一阶段提供一阶段相关的编码,以使得对于阶段中的至少一些来说,其一组操作状态以大致最小量的位编码从而有效地利用所述组数据锁存器且释放一子组空闲数据锁存器。

[0262] 同时发生的锁存器操作

[0263] 步骤 624 :与当前存储器操作同时地以与对存储器阵列进行的一个或一个以上后续存储器操作相关的数据对所述子组空闲数据锁存器执行操作。

[0264] 当前编程期间的读取中断

[0265] 图 25 为下部页编程的示意性时序图,其说明使用可用锁存器而进行的读取中断操作。同时展示主机、I/O 总线、数据锁存器及存储器核心的活动。

[0266] 在第 N 页数据待写入时,主机最初向存储器发布写入命令以将所述页数据写入到指定地址。此后为将待编程的所述页数据发送到存储器。经由 I/O 总线将编程数据切入且将其锁存到每一读取/写入模块的 DL2 中(见图 13 及图 14)。因此, I/O 总线在此切入周期(例如可具有 300 μ s 的持续时间)期间暂时忙碌。

[0267] 下部页编程为二进制的且仅需在如通过 D_A 阈值电平划界的“U”状态与“中间状态”(见图 20A)之间区分。施加到字线的每一编程脉冲由读回或编程验证跟随以确定单元是否已达到表示编程数据的目标状态。在此情形下,编程验证为关于 D_A 的(“pvfyA”)。因此,仅需要来自每一读取/写入模块的一锁存器以存储每一单元的一位。

[0268] 关于数据锁存器,含有编程数据的 DL2 有效地用于发生于存储器阵列或存储器核心中的当前下部位编程操作。因此,正由核心使用的锁存器的数目为一,而另两个锁存器(即 DL0 及 DL1)仍为闲置的。

[0269] 在核心处的编程继续的同时,两个闲置的锁存器及空闲的 I/O 总线可用于读取操

作。读取操作需要已由当前编程操作先占的存储器核心（即，存储器阵列）自身中的感测。然而，读取操作的实际感测阶段通常远远短于编程操作（通常为编程时间的十分之一），从而可中断后者而插入感测操作而不引起性能的较大损失。在感测之后，将读取数据锁存于空闲数据锁存器中的一者或一者以上中。用户接着可将读取数据切出到 I/O 总线。此处可节省时间，因为其与存储器阵列中的编程操作同时发生。

[0270] 因此，在对下部页进行编程的同时，主机可发布读取命令以中断编程时应暂停的要求将编程状态保存于数据锁存器中。感测另一页数据且将其锁存于两个空闲锁存器中的一者（如 DL0）中。接着编程可以所存储的编程状态而恢复。在存储器阵列仍由恢复的编程所占据的同时可将数据锁存器中的读取数据切出到 I/O 总线。

[0271] 如早先所描述，在四状态（2 位）存储器的实例中，对于所述页的每一存储器单元来说优选锁存器数目为三。仅需要用以存储下部页编程数据的一锁存器用于下部页编程。此留下两个空闲锁存器。在通常的读取操作中仅需一个空闲锁存器来锁存经感测的数据位。在优选先行（“LA”）读取操作中，需要两个空闲锁存器。将在稍后章节中对此进行更详细描述。

[0272] 图 26 为上部页编程的示意性时序图，其说明使用可用锁存器而进行的读取中断操作。同时展示主机、I/O 总线、数据锁存器及存储器核心的活动。已结合图 23 描述了多阶段编程，其导致在不同阶段期间不同数目的空闲的数据锁存器可用。举例来说，在已对状态“A”进行编程验证之后，一个数据锁存器空闲，且在已对状态“A”及状态“B”进行编程验证之后，两个数据锁存器空闲。

[0273] 因此，在已对状态“A”进行编程验证之后，单一空闲锁存器可用以锁存从常规读取感测的数据。另一方面，如果已对状态“A”及状态“B”进行编程验证，则两个可用锁存器将能够支持上文所解释的 LA 读取。

[0274] 对多个经高速缓存命令的管理

[0275] 需管理同时发生的存储器操作以支持高速缓存操作，其中在存储器的核心中正执行一个存储器操作，同时于数据锁存器处高速缓存或经由 I/O 总线转移用于额外未决存储器操作的数据。常规存储器装置通常不具有足够数目的空闲数据锁存器来执行高速缓存操作。即使其具有足够数目的空闲数据锁存器，仍仅在已完成当前存储器操作之后执行未决存储器操作（其数据正被高速缓存）。

[0276] 图 27 说明与典型存储器操作相关联的信息的封装。当请求存储器执行存储器操作时，其接收表示指定存储器操作开始的前命令。此后为存储器阵列中操作将要发生的地址。在擦除操作的情形下，地址为待擦除的存储器单元的块。在编程或读取操作的情形下，地址为待操作的存储器单元的页。如果所指定的操作为编程操作，则将供应编程数据以加载到数据锁存器中。当编程数据处于适当位置时，将发布执行命令以关于可用编程数据而执行编程操作。如果所指定的操作为读取操作，则将不向存储器发送数据。将发布执行命令以执行读取操作。将感测经寻址的存储器单元的页且将把经感测的数据锁存于数据锁存器中以最后经由 I/O 总线切出。

[0277] 图 28 说明支持简单高速缓存操作的常规存储器系统。存储器系统包括存储器控制器 8，其经由存储器控制器 8 而控制存储器芯片 301。存储器芯片具有由芯片上主机接口 / 控制电路 310 控制的存储器阵列 100。控制电路包括管理存储器阵列的基本存储器操作

的状态机。主机 6 经由执行例如映射及维护的较高级存储器功能的存储器控制器 8 而啮合存储器系统。

[0278] 状态信号就绪 / 忙碌 * 允许主机或存储器控制器在存储器芯片不忙碌时请求存储器操作。将所请求的存储器操作保持于缓冲器 322 中且释放到状态机 312 以在状态机不执行另一存储器操作时执行。举例来说, 在存储器阵列中由状态机控制而执行存储器操作 MEM OP0。如果存在可用的空闲数据锁存器, 则将向控制器发信号以允许将未决存储器操作 MEM OP1 发送到存储器芯片且在缓冲器 322 中经缓冲。同时, 将与 MEM OP1 相关联的数据切入存储器芯片且锁存到数据锁存器中。MEM OP0 一完成执行, 状态机即释放缓冲器中的 MEM OP1 以开始其执行。因此, 在常规存储器系统中, 在完成当前存储器操作之后执行未决存储器操作。

[0279] 在图 28 所示的实例中, 每一命令在其可开始执行之前必须等待直到最后一个完成, 但其数据在最后一个的执行期间经高速缓存。因此, 在 MEM OP0 执行于存储器核心中的同时, 与 MEM OP1 相关联的数据 1 正被锁存。MEM OP1 将在完成 MEM OP0 之后作用于经高速缓存的数据 1。类似地, 在 MEM OP1 执行于存储器核心中的同时, 与 MEMOP2 相关联的数据 2 正被锁存。此方案阻碍加载同一字线的下部及上部逻辑页及有效地在同一编程操作中编程多个位的可能性。

[0280] 存在影响编程操作 (尤其对于连续编程) 的性能的两个因素。第一个关于加载编程数据的时间。随着快闪存储器容量变得较大, 其页大小也随每一新生代而增加。待被编程的较大页数据因此占用较长时间来加载到数据锁存器中。为了增大编程性能, 需要将数据加载时间藏于别处。此通过在存储器核心于前台中忙于编程操作但使其数据锁存器及 I/O 总线闲置的同时在后台中高速缓存尽可能多的编程数据而完成。

[0281] 本发明的一特征为通过在编程期间于后台中将较多页加载到数据锁存器以使得数据锁存器一可用即被用于高速缓存未决编程数据而处理第一因素。此包括允许在同一前台操作期间于后台中高速缓存与一个以上命令相关联的数据。

[0282] 针对编程性能的第二因素关于编程一页 (尤其对于编程具有同一字线的多位单元的页) 的时间。如之前所述, 可将多位单元的页作为个别单位页的集合而处理。举例来说, 可将 2 位页作为两个稍微独立的单位页 (即下部位页及上部位页) 而对其进行编程及读取。具体来说, 下部位页的编程数据一可用即可对下部位页进行编程。在第二遍中将上部位页编程到存储器单元的同一页且所述编程视已编程于单元中的下部页的值而定。以此方式, 可在两个不同时间于两个单独的遍中对两个位进行编程。然而, 较为有效且较为准确的方式 (具有较少编程干扰) 为在称作“所有位”或“全序列”的编程中在单遍中编程两个位。此仅在所有数据位在编程期间可用的情况下为可能的。因此, 在实际上, 如果所有位可用, 则优选地执行所有位编程。另一方面, 如果仅下部页数据可用, 则将首先对下部页进行编程。稍后如果属于同一字线的上部页数据变得可用, 则将在第二遍中对所述页的单元进行编程。或者, 如果上部页数据在下部页编程完成之前变得可用, 则将需要停止下部页编程且替代地转为执行所有位编程。

[0283] 图 28 所示的方案将不支持在后台中将一个以上命令排队且因此不支持高速缓存一个以上页的数据。此外, 其无法处理以下情形: 下部页编程过早终止且在所有位变得可用时替代地转为执行不同的“所有位”编程。

[0284] 本发明的另一特征为通过允许高速缓存对于所有位编程为必要的所有位以使得所有位编程可发生而处理第二因素。此外,命令队列管理器管理多个未决命令且允许特定命令(视其相关联的数据的状态而定)在完成之前终止以有利于下一未决命令。

[0285] 本发明的两个特征合作以通过高速缓存较多编程数据及允许使用较为有效的编程算法而增强编程性能。

[0286] 根据本发明的一方面,可在将其它多个未决存储器操作排队的同时执行当前存储器操作。此外,当满足特定条件时,所述命令中用于个别操作的一些命令可合并到组合操作中。在一情形中,当满足条件以将队列中的多个未决存储器操作中的一者或一者以上与在执行中的当前存储器操作合并时,当前存储器操作终止且由对经合并操作的操作而替代。在另一情形中,当满足条件以合并队列中的多个未决存储器操作中的两者或两者以上时,对经合并操作的操作将在处于执行中的当前操作完成之后开始。

[0287] 一个实例为在编程共享一共用字线的若干存储器单元的多位页中。可将多个位中的的每一者视作形成二进制逻辑页的位。以此方式,2位存储器单元的页将具有下部逻辑页及上部逻辑页。3位存储器单元的页将具有另外一中部逻辑页。可单独对每一二进制逻辑页进行编程。因此,对于2位存储器单元来说,可在第一遍中对下部逻辑页进行编程且在第二遍中对上部逻辑页进行编程。或者且更为有效地,如果2个位的编程数据可用,则优选地在单遍中对多位页进行编程。

[0288] 视有多少编程数据位可用而定,对于多个二进制编程或经合并且单遍多位编程来说若干情况为可能的。理想地,如果所有位在编程之前可用,则在单遍中对存储器单元的多位页进行编程。如早先所描述,如果仅下部逻辑页编程数据可用,则对下部逻辑页的单位编程可开始。随后,当上部逻辑页编程数据可用时,可在第二遍中对存储器单元的同一页进行编程。另一可能性为上部页数据在下部页编程完成之前变得可用。在所述情形下,为了利用较为有效的单遍多位或“全序列”编程,终止下部页编程且用多位编程替代。其好像合并或组合对于下部逻辑页与上部页的编程。

[0289] 对于具有多位单位的存储器,由主机发送的逻辑编程数据的页可为下部、上部或一些其它中间逻辑页的混合物。因此,一般来说,需要高速缓存数据锁存器将允许的尽可能多的编程数据的页。此将增大逻辑页属于存储器单元的同一页以执行多位编程的合并可能性。

[0290] 图29为说明多个存储器操作的排队及可能合并的流程图。向具有核心阵列及用于锁存与阵列的经寻址的页相关联的数据的数据锁存器的存储器应用用于管理多个存储器操作的算法。

[0291] 步骤710:提供一先进先出队列以对待执行于核心阵列中的传入的存储器操作进行排序。

[0292] 步骤720:无论何时数据锁存器可用于高速缓存传入的存储器操作的数据时接受传入的存储器操作进入队列。

[0293] 步骤730:确定正执行于核心阵列中的存储器操作是否可潜在地与队列中的存储器操作中的任一者合并。如果其潜在地可合并,则前进到步骤740,否则前进到步骤750。

[0294] (就“潜在可合并”来说,其意味着可在单遍中对与存储器单元的同一页相关联的至少两个逻辑页一同进行编程。举例来说,在具有2位存储器单元的存储器中,分别用以编

程下部逻辑页与编程上部逻辑页的两个操作潜在地可合并。类似地,在具有 3 位存储器单元的存储器中,用以编程下部逻辑页与中间页的操作潜在地可合并。同样,用于下部、中间及上部逻辑页的编程操作潜在地可合并。返回到 2 位单元的实例,如果下部逻辑页正在核心阵列中处于执行中,则其在下一编程是编程属于存储器单元的同一页的上部逻辑页的情况下与来自队列未决的下一编程操作潜在地可合并。另一方面,如果上部页正在核心阵列中处于执行中,则其并非潜在可合并的,因为待编程的下一未决页将必须来自于存储器单元的不同页。类似考虑应用于存储器操作为读取操作的情况中)。

[0295] 步骤 740 :无论何时来自队列的下一个或一个以上存储器操作与核心阵列中的存储器操作可合并时,

[0296] 终止核心中对存储器操作的执行且开始替代地执行经合并的存储器操作;

[0297] 否则

[0298] 在执行来自队列的下一存储器操作之前等待直到核心中存储器操作完成。前进到步骤 720。

[0299] (就“可合并”来说,其意味着满足可合并性的条件。在此情形下,下部及上部逻辑页的编程数据在其经锁存于数据锁存器中之后可用。类似地,“合并的存储器操作”将对应于一同编程或感测下部及上部逻辑页)。

[0300] 步骤 750 :等待直到核心中的存储器操作完成;及

[0301] 无论何时来自队列的下两个或两个以上存储器操作可合并时,在核心阵列中执行经合并的存储器操作;

[0302] 否则

[0303] 在核心阵列中执行来自队列的下一存储器操作。前进到步骤 720。

[0304] 通过提供由存储器操作队列管理器控制的存储器操作队列而完成对多个命令的管理。优选地将存储器操作队列管理器实施为状态机中控制存储器阵列中的存储器操作的执行的模块。

[0305] 图 30 说明并入有存储器操作队列及存储器操作队列管理器的优选芯片上控制电路的示意性框图。芯片上控制电路 310' 包括用来控制存储器阵列 100 (也见图 28) 的基本操作的有限状态机 312'。通过先进先出堆栈存储器而实施存储器操作队列 330 以保持任何传入的存储器操作请求。通常,从主机或存储器控制器 (见图 28) 发布存储器操作请求。

[0306] 将存储器操作队列管理器 332 实施为状态机 312' 中的一模块以管理多个未决及执行的存储器操作。队列管理器 332 基本上调度队列 330 中待释放到状态机 312' 中以执行的未决存储器操作。

[0307] 当将例如 MEM OP0 的存储器操作从队列释放到状态机的程序寄存器 324 中时,将在存储器阵列上由状态机控制而执行 MEM OP0。在任何时候,状态机均知晓可用的空闲数据锁存器的数目且此状态经由信号就绪 / 忙碌 * 而传达到主机 / 存储器控制器。如果一个或一个以上空闲的数据锁存器可用,则主机将能够请求例如编程或读取的额外存储器操作。因此容许由主机发送的 MEM OP1、MEM OP2 等等进入队列 330。将通过可用的空闲数据锁存器的数目而确定队列中存储器操作的最大数目。

[0308] 当存储器操作在队列 330 中处于未决状态时,队列管理器 332 将控制未决存储器操作从队列 330 向状态机中的程序寄存器 324 的释放。此外,其确定是否存储器操作中的

任一者可合并到如结合图 29 而描述的组合操作中。在队列中的两个或两个以上的操作可合并的情形下,队列管理器 332 将从队列 330 释放所述可合并操作且将在状态机中的当前操作完成执行之后由状态机 312' 执行组合的操作。在队列中的一个或一个以上操作可与正由状态机执行的操作合并的情形下,队列管理器将使得状态机终止当前执行的操作且替代地执行组合的操作。因此,存储器操作管理器 332 与状态机 312' 的剩余部分协作以调度且(可能地)合并多个存储器操作。

[0309] 已将本发明描述为使用具有 2 位存储器的实例。只要在当前存储器操作期间释放数据锁存器,即可使用其以高速缓存更多数据用于任何未决存储器操作。此将允许将更多数据位加载到可用数据锁存器中以及增加合并存储器操作的可能性。所属领域的技术人员将易于能够对具有可各自存储两个以上数据位的单元的存储器(例如,3 位或 4 位存储器)应用相同原理。举例来说,在 3 位存储器中,可将存储器的页视作具有三个个别位页,即下部、中部及上部位页。可在存储器单元的同页上于不同时间个别地对所述页进行编程。或者,所有三个位在可用时可以所有位编程模式而一同经编程。此要求将高速缓存编程命令排队用于许多页。在 2 位存储器中,可在全序列转换为可能时一同执行两个编程命令。类似地,在 3 位存储器中,三个连续编程命令可在转换为所有位或全序列模式时一同经执行。又,命令队列管理器将跟踪哪一命令已完成或终止且哪一者为下一待执行的。以此方式,在编程期间到达特定存储器状态里程碑时,一些数据锁存器得以释放且可有效地用于高速缓存未决编程数据。

[0310] 擦除期间的高速缓存操作——后台读取及写入操作

[0311] 擦除操作的等待时间为快闪存储系统的整体性能开销的主要组成部分中的一者。举例来说,擦除操作的周期可能比编程操作的周期长四或五倍且比读取操作的周期长十倍。为了改进快闪存储器的性能,例如高速缓存操作的后台操作变得非常重要以利用等待擦除操作结束的时间。本发明将在存储器由存储器核心中的擦除操作占用而忙碌时利用数据锁存器及 I/O 总线。举例来说,可与擦除操作同时执行用于下一编程操作的数据或从读取操作输出的数据。以此方式,当下一编程或读取操作确实发生时,所述操作的数据输入或输出部分已完成,从而减少编程或读取等待时间且增加性能。

[0312] 可以许多方式而实施擦除操作。美国专利第 5,172,338 号中揭示的一方法通过交替擦除脉冲继之以验证而擦除。一旦对单元进行了擦除验证,即禁止其不受进一步擦除脉冲的影响。另一擦除操作(优选地用于 NAND 存储器)包括两个阶段。在第一阶段中,存在通过将电荷从存储器单元的电荷元件移除到预定“擦除”或“接地”状态以下的某一阈值电平而进行的擦除。在第二阶段中,通过一系列关于预定“擦除”阈值的软编程/验证而将经擦除的单元的阈值值收紧为处于适当界定的阈值分布内。

[0313] 根据本发明的一般方面,在擦除操作发生的同时,任何空闲的数据锁存器均用以高速缓存与另一未决存储器操作相关的数据。

[0314] 图 31 为说明擦除操作期间在后台中的高速缓存操作的示意性流程图。

[0315] 步骤 760:向经寻址的页的每一存储器单元提供一组具有锁存预定数目的位的能力的数据锁存器。

[0316] 步骤 770:对指定组的页执行擦除操作。

[0317] 步骤 780:与擦除操作同时地以与对于存储器阵列进行的一个或一个以上后续存

存储器操作相关的数据对所述组数据锁存器执行操作。

[0318] 根据本发明的一方面,在擦除操作发生的同时,经由 I/O 总线而将用于未决编程操作的编程数据加载到数据锁存器中。具体来说,在擦除操作的第一阶段期间移除电荷时,所有数据锁存器均可用于高速缓存编程数据。在擦除操作的第二阶段期间软编程发生时,除一数据锁存器之外的所有数据锁存器可用于高速缓存编程数据,因为需要数据锁存器中的一者来存储成功验证软编程之后所述位置处的编程锁定状况。如果存储器架构支持每单元 2 个位,则存在至少 2 个数据锁存器,每一位一个。在优选实施例中,使用额外数据锁存器以存储在操作期间出现的特定状况。因此,视存储器架构而定,对于 2 位单元存在向每一单元提供的至少两个且优选地三个数据锁存器。所有所述数据锁存器可在擦除的第一阶段期间用于高速缓存用途,且除一者之外的所有所述数据锁存器可在擦除操作的第二阶段期间用于高速缓存用途。因此可视擦除阶段及存储器架构而将一个或一个以上页的编程数据加载到可用数据锁存器中。

[0319] 图 32 为对存储器阵列进行的擦除操作的示意性时序图,其说明擦除操作的第一擦除阶段期间的编程数据加载操作。同时展示主机、I/O 总线、数据锁存器及存储器核心的活动。如图中所示,存储器核心处的擦除操作包括第一擦除阶段,随后为第二软编程 / 验证阶段。

[0320] 在擦除操作的第一阶段期间,存储器阵列或核心被先占,但数据锁存器及 I/O 总线为空闲以用于后台操作。在此时间期间,可经由 I/O 总线而将编程数据加载到数据锁存器中。举例来说,在对于每一单元存在三个数据锁存器的优选实施例中,所有所述锁存器在第一擦除阶段期间均可用于高速缓存操作。

[0321] 举例来说,在第 N 页数据待写入时,主机最初向存储器发布写入命令以将所述页数据写入到指定地址。此后为将待经编程的所述页数据发送到存储器。经由 I/O 总线将编程数据切入且将其锁存到每一读取 / 写入模块的 DL2 中(见图 13 及图 14)。因此,I/O 总线在此切入周期(例如可具有 300 μ s 的持续时间)期间暂时忙碌。在三个数据锁存器可用的情况下,原则上可高速缓存高达三页的编程数据。举例来说,在擦除操作进行的同时可加载第 N 页的下部页部分,或者可顺序地加载第 N 页的下部及上部页部分。

[0322] 图 33 为对存储器阵列进行的擦除操作的示意性时序图,其说明擦除操作的软编程 / 验证阶段期间的编程数据加载操作。同时展示主机、I/O 总线、数据锁存器及存储器核心的活动。

[0323] 在擦除操作的第二软编程 / 验证阶段期间,存储器阵列或核心也被先占。然而,如上文所述,除一数据锁存器以外的所有数据锁存器及 I/O 总线为空闲的。可将编程数据加载到未由擦除操作使用的数据锁存器中。举例来说,在对于每一单元存在三个数据锁存器的优选实施例中,软编程 / 验证操作仅使用锁存器中的一者。因此仍存在两个空闲的锁存器可用于高速缓存操作。

[0324] 举例来说,在第 N 页数据待写入时,主机最初向存储器发布写入命令以将所述页数据写入到指定地址。此后为将待经编程的所述页数据发送到存储器。经由 I/O 总线将编程数据切入且将其锁存到每一读取 / 写入模块的 DL2 中(见图 13 及图 14)。因此,I/O 总线在此切入周期(例如可具有 300 μ s 的持续时间)期间暂时忙碌。在两个数据锁存器可用的情况下,原则上可高速缓存高达两页的编程数据。举例来说,在擦除操作进行的同时可

加载第 N 页的下部页部分,或者可顺序地加载第 N 页的下部及上部页部分。

[0325] 一般来说,可加载到数据锁存器中的页的最大数目依据存储器架构以及并行编程多少平面 / 组及多少芯片 / 电路小片及数据传送率的速度而变。

[0326] 根据本发明的另一方面,在擦除操作发生时,可插入读取操作且可在擦除操作期间输出数据锁存器中的所得读取数据。优选地,在不中断软编程脉冲自身的情形下将读取操作插入于软编程 / 验证操作之间。一旦将数据感测且锁存到未使用的数据锁存器中,即可在擦除于阵列内部进行时经由 I/O 总线而将数据输出到主机系统。此特征对于隐藏系统开销以 (例如) 执行读取擦除操作及其它系统维护来说为理想的。

[0327] 在现有技术的系统中,当擦除操作被中断时,其将必须从循环开始处重新开始。此可为非常耗时的 (尤其在 NAND 存储器中)。

[0328] 可将读取操作插入于软编程与擦除验证脉冲之间。可将与软编程脉冲的数目一样多的读取插入擦除操作中。感测时间为额外时间,但与整体软编程 / 验证操作相比具有较短持续时间。益处在于与正在进行中的编程 / 验证操作并行发生时切出读取数据中获得。读取操作还可用于在管理内部控制及数据管理时执行后台操作。

[0329] 读取在快闪存储系统中于擦除期间的一有用应用在于实施读取擦除操作以将所存储的数据保持于良好状况。周期性地读取存储器的存储数据的部分以检查单元中的编程电荷是否随时间而改变或在其环境中改变。如果是,则通过以适当容限再编程单元而对其进行校正。美国专利第 7,012,835 号中已揭示读取擦除的各种方案,所述专利的全部揭示内容以引用的方式并入本文中。由于读取擦除为主机的操作的外部的系统操作,因此将读取擦除藏于一些其它操作之后为最佳的,其中存储器无论如何均将为忙碌的。在此情形下,在擦除操作期间,可插入读取擦除操作以使得可隐藏读取等待时间。

[0330] 图 34 为对存储器阵列进行的擦除操作的示意性时序图,其说明插入的读取操作及使用可用锁存器而进行的所得数据输出操作。同时展示主机、I/O 总线、数据锁存器及存储器核心的活动。如图中所示,在擦除操作的第二阶段中,操作为软编程 / 验证。优选地在不中断任何软编程脉冲的完成的情形下插入一个或一个以上读取操作。

[0331] 在芯片处于擦除操作的第二阶段中时,用于软编程 / 验证的算法将执行。例如忙碌 / 就绪 * (未图示) 的状态信号将以信号表明存储器核心忙于内部擦除操作。同时,如高速缓存忙碌 / 高速缓存就绪 * (未图示) 的另一状态信号将从忙碌变为就绪以接受读取命令输入。读取命令一进入,高速缓存忙碌 / 高速缓存就绪 * 即转为忙碌以防止另一命令进入。读取命令接着将等待直到当前软编程脉冲在内部完成才可对同一芯片中的另一经寻址的块执行。在读取完成后,将地址变回先前执行的擦除块。软编程 / 验证操作可对于擦除块而恢复。

[0332] 同时,可将数据锁存器中的读取数据切出。切出时间通常远长于读取时间。举例来说,读取时间为大约 $25\ \mu\text{s}$,而切出时间为大约 $200\ \mu\text{s}$ 。因此将读取插入于擦除操作中的益处为从原本在等待擦除结束时浪费的时间抢救约 $200\ \mu\text{s}$ 。

[0333] 可在擦除期间在擦除时间允许的情况下将此高速缓存读取插入尽可能多次。然而,过多读取可延长总擦除时间且读取可能招致的擦除操作的时间损失与从读取抢救的切换时间之间的平衡将受到冲击。如果擦除期间在一个或一个以上插入的读取之后仍存在剩余空闲时间,则可如早先章节中所述而使用可用数据锁存器以高速缓存任何编程数据。如

果加载编程数据,则编程操作仅可在整个擦除操作完成之后开始。必须保留足够的空闲锁存器用于对编程操作的适当执行,因此在多数情形下在加载编程数据之后其它高速缓存操作将为不可能的。

[0334] 图 35 为说明图 31 的步骤 780 中在擦除操作期间在后台中用于读取擦洗应用的特定高速缓存操作的示意性流程图。

[0335] 将图 31 所示的步骤 780 进一步清楚表示为如下:

[0336] 步骤 782:暂停擦除操作以感测一指定页。

[0337] 步骤 784:在将用于指定页的数据锁存于数据锁存器中之后恢复擦除操作。

[0338] 步骤 786:在擦除操作期间输出用于指定页的数据。

[0339] 步骤 788:调度指定页以在输出数据含有误差的情况下进行再编程。

[0340] 到此为止对高速缓存读取的描述大部分关于优选擦除操作的第二阶段而进行。优选擦除操作如下:第一阶段为擦除所有单元到预定阈值以下的某一阈值电平且第二阶段为将单元软编程到预定阈值。如上文所述,此擦除方案优选地用于具有 NAND 结构的快闪存储器,因为其需要相当准确的基态且通过对 N 阱偏置而擦除存储器,此耗费时间。因此,优选地在软编程之前一同执行所有擦除。在使用擦除脉冲发出/验证/禁止的方案的另一存储器架构中,还预期高速缓存操作。举例来说,可在循环的验证部分期间插入读取操作。

[0341] 图 36 说明擦除期间的抢先后台读取。当读取恰于擦除操作之前发生以使得无需中断擦除操作时,此为更优选的高速缓存读取。此在于擦除操作开始之前已知读取操作的情况下为可能的。举例来说,主机可能具有一未决的读取请求或者如果存储器系统具有经调度的某一读取操作。或者,一智能算法可能预见下一读取可能在何处且调度所述读取。即使稍后弄清楚其为未命中,也将不会招致严重损失。如果其为一命中,则其可利用擦除时间以切出读取数据。

[0342] 可组合擦除操作期间高速缓存读取数据及高速缓存编程数据的两个方面以提供进一步的灵活性来最小化整体系统或存储器开销。即使在多平面及多芯片数据输入操作的情况下,数据输入时间也可能未充分利用擦除操作所招致的忙碌时间。在所述情形下,还可添加读取操作及或编程操作以充分利用擦除时间。

[0343] 读取期间的高速缓存操作——后台读取及写入操作

[0344] 在顺序地读出许多页时通常实施高速缓存读取以节省时间。可在切出先前感测的页的时间期间隐藏对一页的感测以使得用于感测的时间不招致用户的额外等待时间。一常见方案将在切出当前页时感测下一页。

[0345] 图 37 示意性地说明典型读取高速缓存方案。在先前循环中感测第 (n-1) 页且将其锁存于数据锁存器中。在时间 t_0 处,如由 $T(n-1)$ 所指示而经由 I/O 总线从数据锁存器切出第 (n-1) 页。在切换发生的同时,可如 $S(n)$ 所指示而感测且锁存第 n 页。在 t_2 处,完成对第 (n-1) 页的切换且因此其可继之以如由 $T(n)$ 所指示的从数据锁存器切换第 n 页的数据。类似地,在切出第 n 页数据时,可如 $S(n+1)$ 所指示而感测且锁存第 (n+1) 页的数据。可紧于第 n 页完成切换之后切换此第 (n+1) 页。理想地,数据锁存器及 I/O 总线在整个读取高速缓存期间完全处于使用中以使得任何闲置时间得以最小化。

[0346] 根据本发明的一方面,提供读取高速缓存方案用于具有最小化存储器单元之间的扰动(Yupin 效应)的需要的多状态存储器单元的情形。在优选实施方案中,使用有效读取

高速缓存方案用于以“LM”编码而编码且以先行（“LA”）校正而读取的存储器。“LM”编码及“LA”校正均需要除仅仅切换读取数据以外的额外锁存器及总线活动。结合图 37 而描述的常规方案的直接应用将不产生最佳读取高速缓存。

[0347] 随着半导体存储器中的日益提高的集成度，存储器单元之间归因于所存储的电荷的电场的扰动（Yupin 效应）在单元间间距正在收缩时变得越来越明显。优选地使用 LM 编码来对存储器的多状态存储器单元进行编码，以最佳次序编程存储器中的页，且使用 LA 校正而读取经编程的页。改进的读取操作将实施最佳高速缓存操作。

[0348] 对于 LM 代码的高速缓存读取算法

[0349] 当待读取的页为多状态时，读取高速缓存的实施方案需满足所使用的多状态编码的要求。如之前结合图 20A 到图 20E 而描述，用于多状态存储器的 LM 编码本质上使存储器单元中经编程的电荷在不同遍的编程之间的改变最小化。所示的实例是针对 2 位存储器，其用于编码每一单元中如由三个不同划界阈值值（例如， D_A 、 D_B 、 D_C ）而划界的四个可能存储器状态（例如，“U”、“A”、“B”、“C”）。举例来说，在 2 位存储器单元中，对下部逻辑页的编程至多将阈值电平推进为略低于单元的阈值窗口的中部。后续上部逻辑页编程将现有阈值电平进一步推进约距离的另一四分之一。因此，从第一遍下部到第二遍最终上部编程，净改变至多为阈值窗口的大约四分之一，且此将为单元从其沿一字线的相邻者处可能经历的扰动的最大量。

[0350] LM 编码的一特征在于可单独地考虑两个位（下部及上部位）中的每一者。然而，对下部位页的解码将视是否已对上部页进行编程而定。如果已对上部页进行编程，则读取下部页将需要关于划界阈值电压 D_B 的读取 B 的一读取通过。如果尚未对上部页进行编程，则读取下部页将需要关于划界阈值电压 D_A 的读取 A 的一读取通过。为了区分两种情形，在对上部页进行编程时在上部页中（通常在开销或系统区中）写入旗标（“LM”旗标）。在对下部位页的读取期间，将首先假定已对上部页进行编程且因此将执行读取 B 操作。如果 LM 旗标经读取，则假定正确且完成读取操作。另一方面，如果第一读取未产生旗标，则其将指示尚未对上部页进行编程且因此必须通过读取 A 操作而再读取下部页。

[0351] 对上部位页读取的解码将需要操作读取 A 及读取 C，其分别关于划界阈值电压 D_A 及 D_C 。类似地，如果尚未对上部页进行编程，则上部页的解码也可被干扰。再一次，LM 旗标将指示是否已对上部页进行编程。如果尚未对上部页进行编程，则读取数据将被重设为“1”而指示未对上部页数据进行编程。

[0352] 在使用 LM 编码而实施对存储器的高速缓存读取时，存在需要检查与数据存储于同一区上的 LM 旗标的额外考虑。为了使状态机检查 LM 旗标，其将必须经由 I/O 总线而从数据锁存器输出。此将需要对 I/O 总线进行分配以在具有高速缓存的读取操作期间除了切换所感测的数据以外用于输出 LM 旗标。

[0353] 图 38A 为关于以 LM 代码编码的逻辑页的高速缓存读取的示意性时序图。在感测当前页的同时切换上一页数据的一般方案类似于图 37 所示的常规读取的方案。然而，以 LM 代码进行的感测由于潜在地必须进行两遍感测（LM 旗标的检查在其间）而为复杂的。

[0354] 在时间 t_0 处，如由 $T(n-1)$ 所指示而将上一循环中所感测的第 $(n-1)$ 逻辑页从数据锁存器切出到 I/O 总线。同时， $S_1(n)$ 感测下一逻辑页 (n) 。在 LM 编码的情况下，需区分两种情形：对下部位逻辑页的读取；及对上部逻辑页的读取。

[0355] 对于读取下部位逻辑页的情形, 优选感测将以对于已对上部位逻辑页进行编程的假定而开始, 因此第一感测 $S_1(n)$ 将处于关于划界阈值电压 D_B 的读取 B 处。在 t_1 处完成 $S_1(n)$ 且将产生 LM 旗标。然而, 其仅可在 I/O 总线完成切换第 $(n-1)$ 页之后的 t_2 处输出。在将 LM 旗标传达到状态机之后, 对其进行检查以确定上部页是否存在。如果 LM 旗标经设定, 则假定正确且下部位页经正确读取。已锁存的页 (n) 的数据准备好在下一循环中被切出。

[0356] 对于读取上部位逻辑页的情形, $S_1(n)$ 将逐步通过分别关于划界阈值电压 D_A 及 D_C 的读取 A 及读取 C。上部位页的所感测的数据将存储于 DL2 中且 DL0 数据锁存器用于切出数据 (见图 13 及图 14)。在 t_2 处, 将 DL2 的感测的数据转移到 DL0。又, 在于第 $(n-1)$ 页的切换的结尾处输出 LM 旗标之后对其进行检查。如果上部页经编程, 则一切情况良好且锁存器中的所感测的数据 (页 (n)) 准备好在下一循环中被切出。

[0357] 在读取上部位逻辑页时, 如果发现 LM 旗标未经设定, 则其将指示上部页未经编程。从 $S_1(n)$ 感测的数据将被重设为“1”以与 LM 编码适当地一致。感测的数据接着准备好输出。接着将预取出第一字节且随后为下一循环开始时的整页切出。

[0358] 图 38B 为关于以 LM 代码进行的高速缓存读取在尚未对上部位逻辑页进行编程时读取下部位逻辑页的特殊情形中的示意性时序图。又, 在 t_0 处开始第一感测 $S_1(n)$ 且在 t_1 处读取 LM 旗标。输出 LM 旗标用于 t_2 处的检查。如果发现 LM 旗标未经设定, 则 $S_1(n)$ 在读取 B 处不正确地读取了下部位页。第二感测 $S_2(n)$ 将开始于 t_3 以在读取 A 处执行。然而, 此额外感测 (结束于 t_4) 无法隐藏于第 $(n-1)$ 页的切换 (例如, $T(n-1)$) 的时间后, 因为在第二感测之前检查来自 $S_1(n)$ 的旗标将需要存取 I/O 总线且将必须等待直到 $T(n-1)$ 切换完成。

[0359] 以所有位感测而进行的高速缓存读取算法

[0360] 在替代方案中, 当在一字线上待读取的页为具有同一物理页上的多个逻辑页的多个位时, 可在一感测操作中一同感测所有多个位以节省功率。

[0361] 图 39 说明对于 2 位存储器以所有位感测而进行的高速缓存读取的示意性时序图。在 2 位的情形下, 在同一操作中感测表示四个存储器状态的两个位。此将需要在读取 A、读取 B 及读取 C 处感测以区分四个状态。在此情形下, 感测将在每隔一个的循环中发生。举例来说, 感测仅在奇数循环上发生且在偶数循环上将被跳过。将在每一循环顺序地切出在一感测中获得的两个逻辑页。

[0362] 在存在八个状态 (例如“U”、“A”、“B”、“C”、“D”、“E”、“F”及“G”) 的 3 位情形下, 所有位感测将涉及在读取 A、读取 B、读取 C、读取 D、读取 E、读取 F 及读取 G 处的感测以区分八个状态。

[0363] 一般来说, 少于所有位的任何多位感测将用来减少读取页的所有位所需感测的次数且将有助于节省功率。结合图 30 而描述的存储器操作队列及队列管理器可用以通过合并两个或两个以上的二进制页感测而管理所有位感测操作。所有位感测方案可应用于具有 LM 代码的存储器且还可应用于具有 LA 校正的存储器 (其将在下一章节中得以描述)。

[0364] 关于 LM 代码连同 LA 校正的高速缓存读取算法

[0365] 关于邻近字线上的存储器单元之间的扰动, 其可通过使用优选编程方案而在编程期间得以减轻。此将有效地将扰动减半。还可通过使用优选 LA 读取方案而在读取期间校正剩余的一半。

[0366] 优选编程方案将以最佳序列而编程与字线相关联的页。举例来说,在每一物理页保持一页二进制数据的二进制存储器的情形下,优选地沿始终如一的方向(例如从底部到顶部)而顺序地对页进行编程。以此方式,当编程特定页时,其下侧的页已经编程。无论其对于当前页有何扰动效应,在鉴于所述扰动而对当前页进行编程验证时均对其加以计及。本质上,编程页的序列应允许正进行编程的当前页在其经编程之后经历围绕其环境的最小改变。因此,每一经编程的页仅受其上侧的页的扰动且字线到字线的 Yupin 效应通过此编程序列而有效地减半。

[0367] 在存储器单元的每一物理页为多状态的存储器的情形下,序列较不直接。举例来说,在 2 位存储器中,可将与一字线相关联的每一物理页视作具有 2 位数据的单一页或两个单独的逻辑页(各自具有 1 位数据的下部及上部位)。因此可在一遍中关于两个位对物理页进行编程,或在两个单独遍中,首先关于下部位页且接着稍后关于上部位页而对物理页进行编程。当将在两个单独遍中对每一物理页进行编程时,经修改的最佳序列为可能的。

[0368] 图 40 说明一存储器的实例,其具有 2 位存储器单元且其中其页以最佳序列编程从而最小化邻近字线上的存储器单元之间的 Yupin 效应。为了方便,表示法为如下:物理页 P0、P1、P2、..... 分别驻留于字线 W0、W1、W2、..... 上。对于 2 位存储器来说,每一物理页具有与其相关联的两个逻辑页,即各自具有二进制数据的下部位及上部位逻辑页。一般来说,通过 LP(字线. 逻辑_页)而给出特定逻辑页。举例来说,将 W0 上的 P0 的下部位及上部位页分别标为 LP(0.0) 及 LP(0.1),且 W2 上的对应者将为 LP(2.0) 及 LP(2.1)。

[0369] 本质上,逻辑页的编程将遵循序列 n 以使得正进行编程的当前页在其经编程之后将经历围绕其环境的最小改变。在此情形下,再次在从底部到顶部的一始终如一的方向上递增地移动将有助于消除来自一侧的扰动。此外,因为每一物理页可能具有两遍编程,所以在编程对于物理页上移时,当前上部位页在已对其邻近的下部位页进行编程之后经编程以使得将在对当前上部位页进行编程时计及所述下部位页的扰动效应将为优选的。因此,如果编程从 LP(0.0) 开始,则序列将如以将产生 LP(0.0)、LP(1.0)、LP(0.1)、LP(2.0)、LP(1.1)、LP(3.0)、LP(2.1)、..... 的页编程次序 0、1、2、... n、... 而做记号。

[0370] 关于 LM 代码连同 LA 校正的高速缓存读取算法

[0371] 根据本发明的一方面,实施用于高速缓存读取数据的方案以使得即使对于校正视来自相邻物理页或字线的数据而定的读取操作,数据锁存器及 I/O 总线也有效地用以在当前页正从存储器核心而被感测的同时切出先前读取页。具体来说,优选读取操作为“先行”(“LA”)读取且对于存储器状态的优选编码为“中下”(“LM”)代码。在必须以对邻近字线上的数据的预先必要的读取而居先于对当前字线上的当前页的读取时,所述预先必要的读取连同任何 I/O 存取在读取先前页的循环中经抢先完成以使得可在先前读取的页忙于 I/O 存取的同时执行当前读取。

[0372] 在 2005 年 4 月 5 日申请的题为“用于包括补偿耦合的非易失性存储装置的读取操作 (Read Operations for Non-Volatile Storage that Includes Compensation for Coupling)”的美国专利申请案第 11/099,049 号中已揭示 LA 读取方案,所述专利全部揭示内容以引用的方式并入本文中。伴随 LA(“先行”)校正的读取基本上检查编程到邻近字线上的单元中的存储器状态且校正其对当前字线上正被读取的存储器单元所造成的任何扰动效应。如果页已根据上文描述的优选编程方案而编程,则邻近字线将来自紧在当前字线

上方的字线。LA 校正方案将需要邻近字线上的数据先于当前页而经读取。

[0373] 举例来说,参看图 40,如果待读取的当前页 (n) 处于 WL_m (例如, WL_1) 上,则如将由 $S_{LA}(n)$ 所表示的 LA 读取将首先读取下一字线 WL_{m+1} (例如, WL_2) 且将数据结果保存于一数据锁存器中。接着,将接着鉴于 $S_{LA}(n)$ 结果而感测当前页且此将由 $S_1'(n)$ 表示。

[0374] 如早先结合图 40 所描述,在具有优选编程序列的 LM 代码中,下部页 (例如, $LP(1,0)$) 将经编程到 D_b 或接近于 D_b (中间状态)。将仅在对 WL_{m+1} 下部页 (例如, $LP(2,0)$) 编程之后编程上部页 (例如, $LP(1,1)$)。接着将完全消除下部页的 WL-WL 的 Yupin 效应。因此,将仅对“A”及“C”状态而不对“U”或“B”状态执行数据相关的校正。

[0375] 在 LA 读取的优选实施中,使用锁存器以指示 LA 读取是否发现“A”或“C”状态或者“U”或“B”状态。在前一情形下需要校正且在后一情形下不需要校正。将通过对感测参数的合适调整 (例如提升感测期间的字线电压) 而对应地校正当前读取 $S_1(n)$ 中的对应单元。此通过在调整的情况中感测一次且在未调整的情况下感测另一次而对整个当前页进行。接着将根据锁存器是否指示校正而从所述两次感测选择页的每一单元的数据。

[0376] 以 LM 代码进行的读取将需要在最终定下读取结果之前检查 LM 旗标 (通过第二遍读取或通过重设读取数据)。LA 校正需在读取当前字线之前首先进行下一字线读取。因此,需通过状态机而检查来自下一字线读取的 LM 旗标及来自当前字线的 LM 旗标。需在 I/O 总线不忙于切换读取数据时经由 I/O 总线将所述两个 LM 旗标输出到状态机。

[0377] 图 41 说明根据图 37 所示的常规方案的对于 LM 代码连同 LA 校正的读取高速缓存的实施方式。基本上,常规方案关于将对当前页的感测藏于所感测的先前页的数据切出时间内。然而,在此情形下,必须以 WL_{m+1} 上的额外先行读取 $S_{LA}(n)$ 居先于 WL_m 上的当前页感测 $S_1'(n)$ 。必须在确定所感测的数据之前经由 I/O 总线输出所述感测中的每一者的 LM 旗标。鉴于来自 $S_{LA}(n)$ 的数据而执行当前页感测 $S_1'(n)$ 以产生当前页的经校正的数据。应了解如图 38B 所示,如果 n 为下部位页且上部位页尚未经编程,则 $S_1'(n)$ 之后可存在额外 $S_2'(n)$ 。

[0378] 在开始于 t_0 的下一循环中,接着如 $T(n)$ 所指示而切出页 n 的经校正的感测的数据。同时,当前感测现已以必须由 $S_{LA}(n+1)$ 居先的 $S_1'(n+1)$ 而移动到下一页。然而,来自所述感测的 LM 旗标的输出必须等待直到对页 n 的切换 $T(n)$ 完成。此外,仅可在 $S_{LA}(n+1)$ 的结果确定之后执行 $S_1(n+1)$ 。因此, $S_1'(n+1)$ 仅可在数据切换周期之外执行且因此无法藏于其之后。此在未充分利用锁存器及 I/O 总线时添加额外感测时间,且浪费的时间对于每一后续循环均重复。此实施方式在使用 LA 校正时使用户的读取性能降级。

[0379] 以 LM 代码连同 LA 校正进行的高速缓存读取的优选实施为以所有感测将藏于数据切换内的方式而对下一字线感测及当前字线感测进行管线式操作。下一字线感测总在当前字线感测之前执行。在每一群组数据切换内,将执行当前字线感测且随后为下下一字线感测。当已结束切出所述群组数据且 I/O 总线可用时,将首先取出下下一字线 LM 旗标且对其进行检查。如果 LM 旗标处于指示上部页未经编程的状态中,则将下下一字线的感测的数据重设为“1”(由于无校正)。随后将检查当前字线 LM 旗标。视当前字线 LM 旗标而定,保持所感测的数据或需执行另一感测 (在下部页读取的情形下) 或者将数据重设为均为“1” (在上部页读取的情形下)。对于具有 2 位存储器单元的存储器,可通过 3 个数据锁存器而管理所有所述感测及数据切出。

[0380] 图 42 说明以 LM 代码及 LA 校正进行的改进读取高速缓存方案。从 $-t_5$ 到 t_0 的第一循环为读取 WL_m 上的当前页 (n) 的时间且不同于循环的剩余部分。如前所述, LA 校正需要在先的读取 $S_{LA}(n)$, 其中读取 A、读取 B 及读取 C 将感测 WL_{m+1} 上的单元状态。来自此读取的 LM 旗标 $F_{LA}(n)$ 将于 $-t_4$ 输出且经检查。如果旗标指示上部页在 WL_{m+1} 上未受到编程, 则感测的数据将被重设为均为“1”以指示将不存在校正。如果旗标指示上部页已经编程, 则指示校正与否的经锁存的数据将保持为原状。在 $-t_3$ 处, 将根据早先描述的 LM 代码及 LA 校正方案以 $S_1'(n)$ 及 (可能地) $S_2'(n)$ 感测 WL_m 上的当前页。与图 41 所说明的方案形成对比, 还对于下一页 (n+1) 执行抢先先行读取。因此, 在时间 $-t_2$ 处执行 $S_{LA}(n+1)$ 且在 $-t_1$ 处输出并检查其 LM 旗标。

[0381] 在第一循环之后, 在 t_0 处的下一循环的开始时, 将如由 $T(n)$ 所指示而切出先前从 $S_1'(n)$ 感测的数据 (现经 LA 校正)。页地址将首先递增到驻留于由图 38 所指示的次序给出的字线上的 (n+1)。因此, 在时间 t_0 处, 伴随着 $T(n)$ 的开始, 对第 (n+1) 页的感测 $S_1'(n+1)$ 可立刻开始, 因为其预先必要的先行 $S_{LA}(n+1)$ 已在先前循环中完成。在 t_1 处的 $S_1'(n+1)$ 的结尾处, 将取出并检查 LM 旗标 $F(n+1)$ 且任何额外动作将视 LM 旗标而跟随。经校正的页 (n+1) 的数据接着将准备好在下一循环中切换。同时, 虽然仍在切出页 (n), 但可预先且在 $T(n)$ 的切换周期内执行对下一页的先行感测 $S_{LA}(n+2)$ 。

[0382] $T(n)$ (对页 (n) 的切换) 一完成, 下一循环即开始且 $T(n+1)$ 以对经 LA 校正的页 (n+1) 的数据的切出而跟随。对于页 (n+1) 的循环以与对于页 (n) 的循环相似的方式而继续。重要特征在于在早先循环中抢先执行对于给定页的先行读取。

[0383] 图 43 为说明改进读取高速缓存的示意性流程图:

[0384] 步骤 810: 在每一读取循环 (其中将从存储器感测来自一系列页中的一页) 中, 在当前循环中输出在上一循环中感测的先前页。

[0385] 步骤 830: 在所述输出先前页期间感测当前页, 所述感测当前页执行于当前字线上且需要在邻近字线处的预先必要的感测以校正来自于邻近字线上的数据的任何扰动效应。

[0386] 步骤 850: 在早于当前循环的循环中抢先地执行与当前页相关的邻近字线的所述预先必要的感测。

[0387] 图 44 为以进一步的清晰度说明图 41 的步骤 850 的示意性流程图:

[0388] 步骤 852: 输出作为来自所述预先必要的感测的数据的部分而获得的第一旗标。

[0389] 步骤 854: 根据输出的第一旗标而调整来自所述预先必要的感测的数据。

[0390] 步骤 856: 锁存数据以指示是否需要对于跟随的对当前页的所述感测而进行校正。

[0391] 图 45 为以进一步的清晰度说明图 41 的步骤 830 的示意性流程图:

[0392] 步骤 832: 以或不以来自预先必要的感测的校正而执行对当前页的所述感测。

[0393] 步骤 834: 输出作为来自所述当前感测的数据的部分而获得的第二旗标。

[0394] 步骤 836: 响应于第二旗标, 通过将数据保持为不改变或将数据调整一预定值或者在另一组感测条件下重复对当前页的所述感测而获得新数据来修订来自所述当前感测的数据。

[0395] 步骤 838: 锁存根据来自预先必要的感测的数据是指示校正还是无校正而经校正

或未经校正的修订数据。

[0396] 已使用 2 位 LM 代码而描述以上的算法。算法为对于 3 个或 3 个以上的位同样地可应用的 LM 代码。

[0397] 虽然已关于特定实施例而描述本发明的各种方面,但应了解,本发明有权保护所附权利要求书的全部范围。

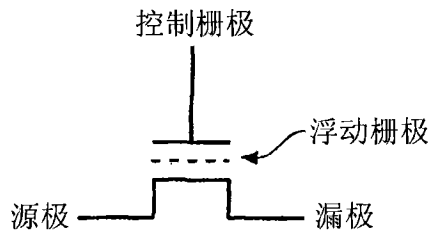


图 1A

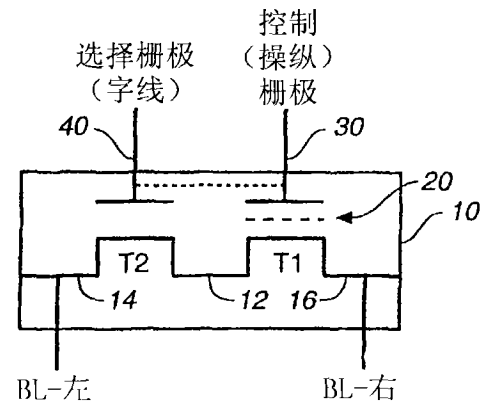


图 1B

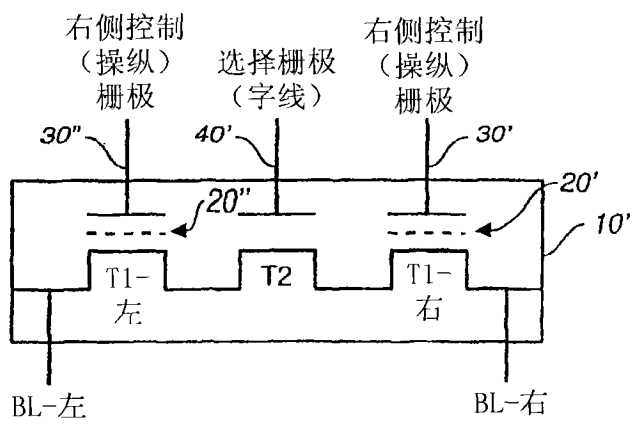


图 1C

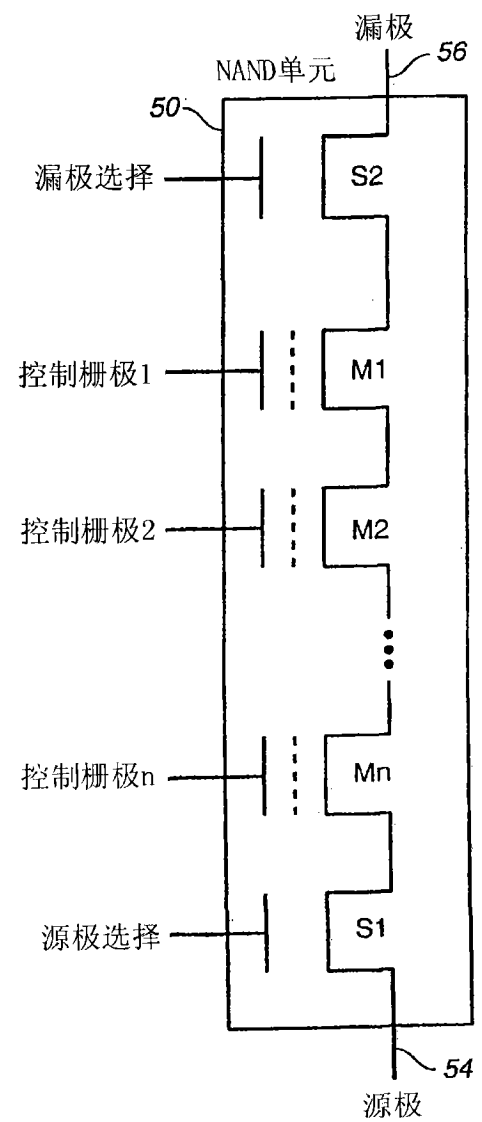


图 1D

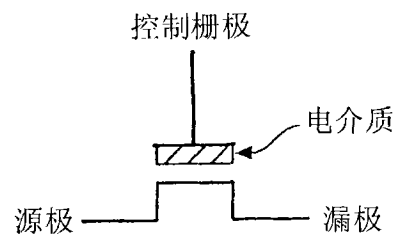


图 1E

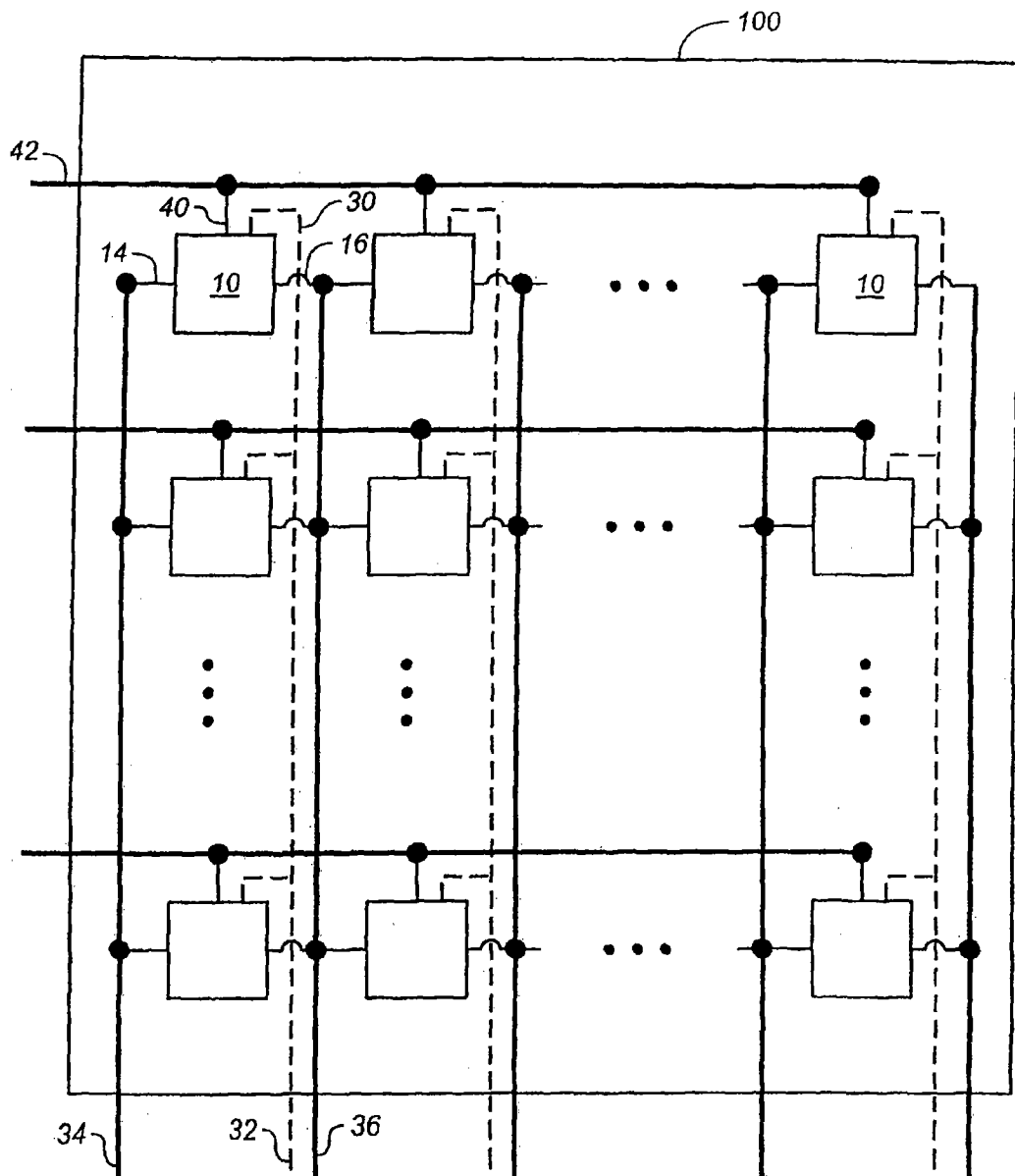


图 2

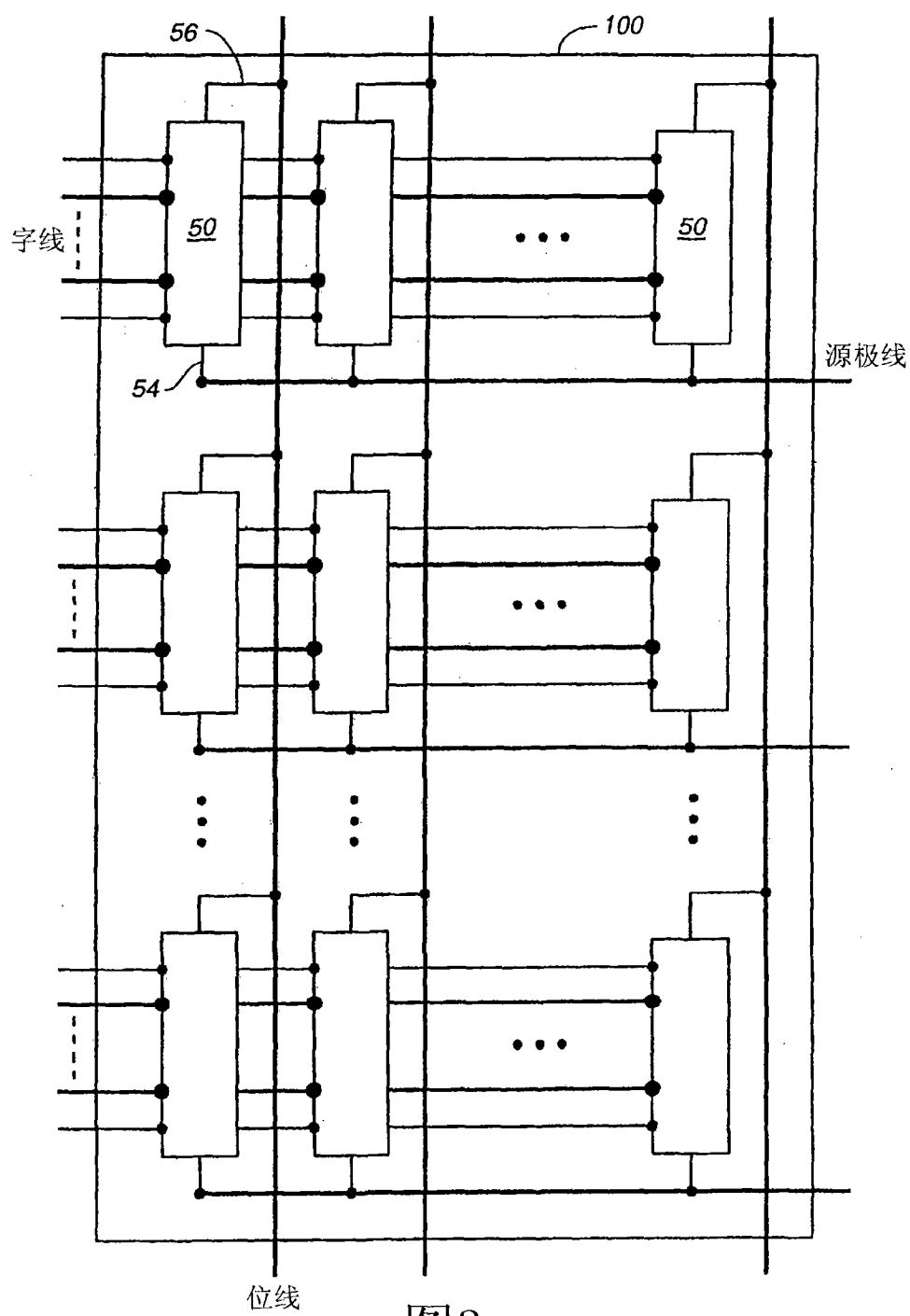


图3

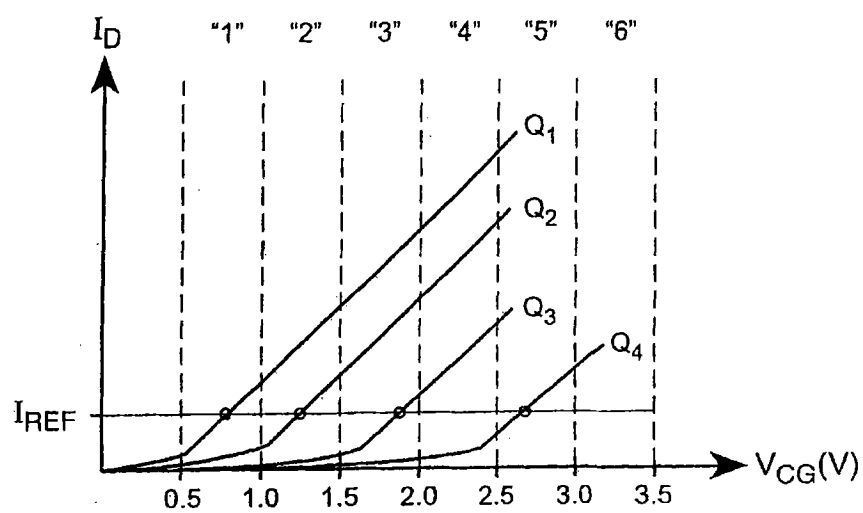


图 4

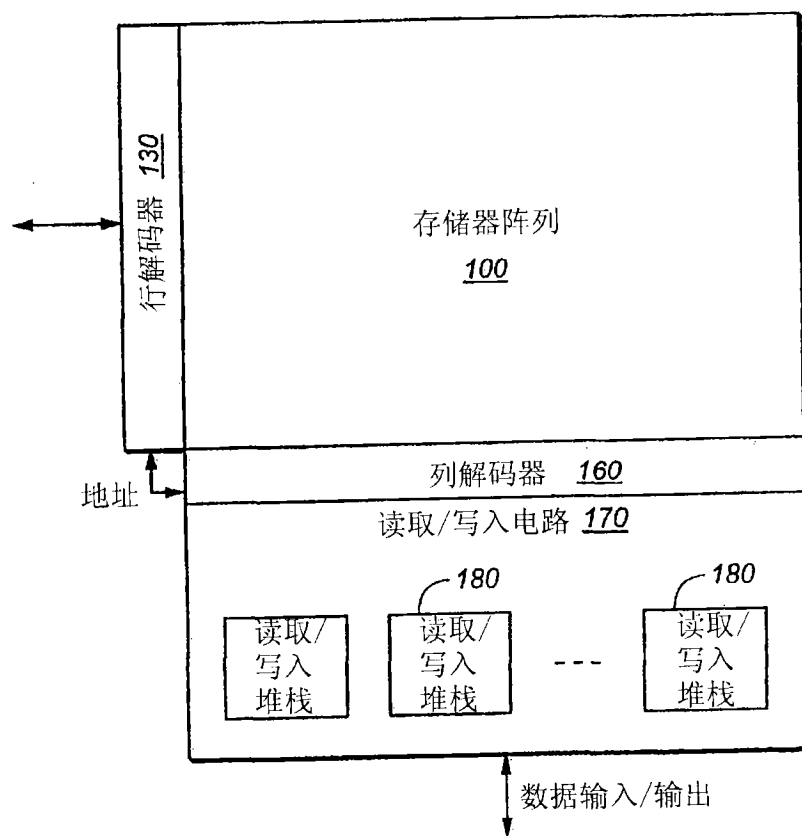


图 5

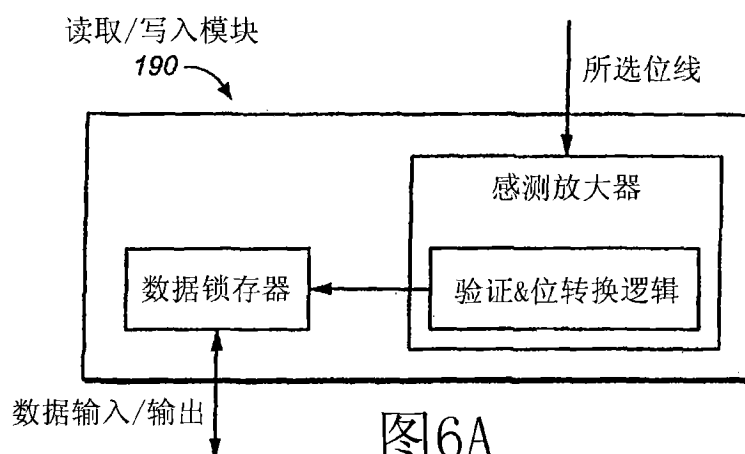


图6A
(现有技术)

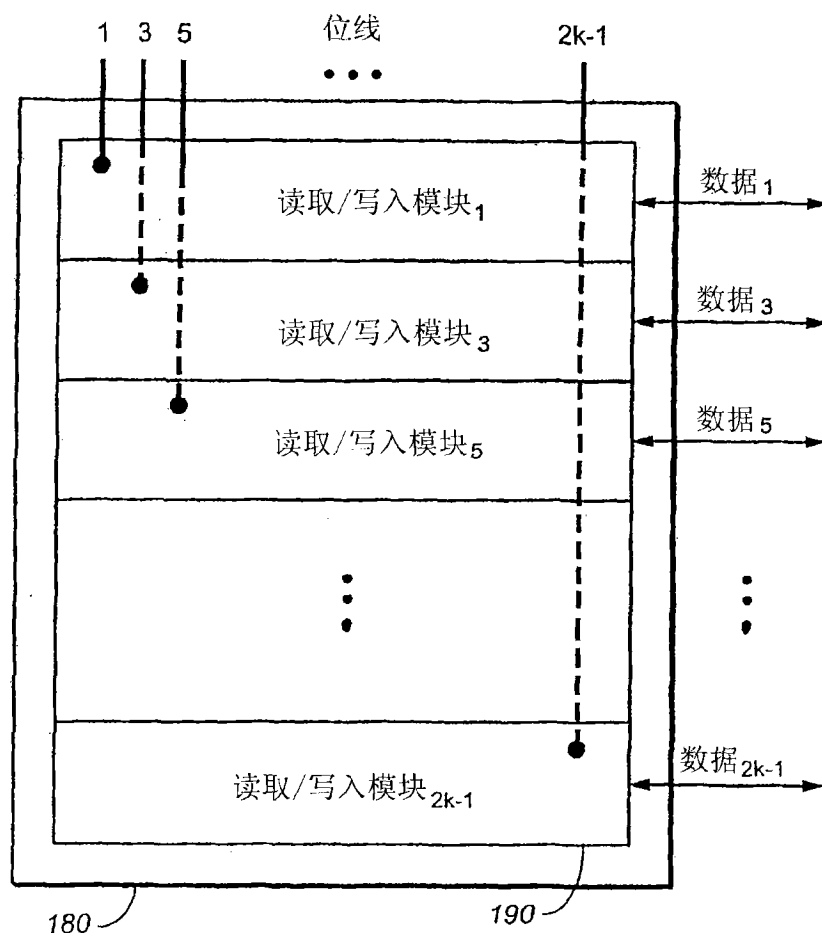


图 6B

(现有技术)

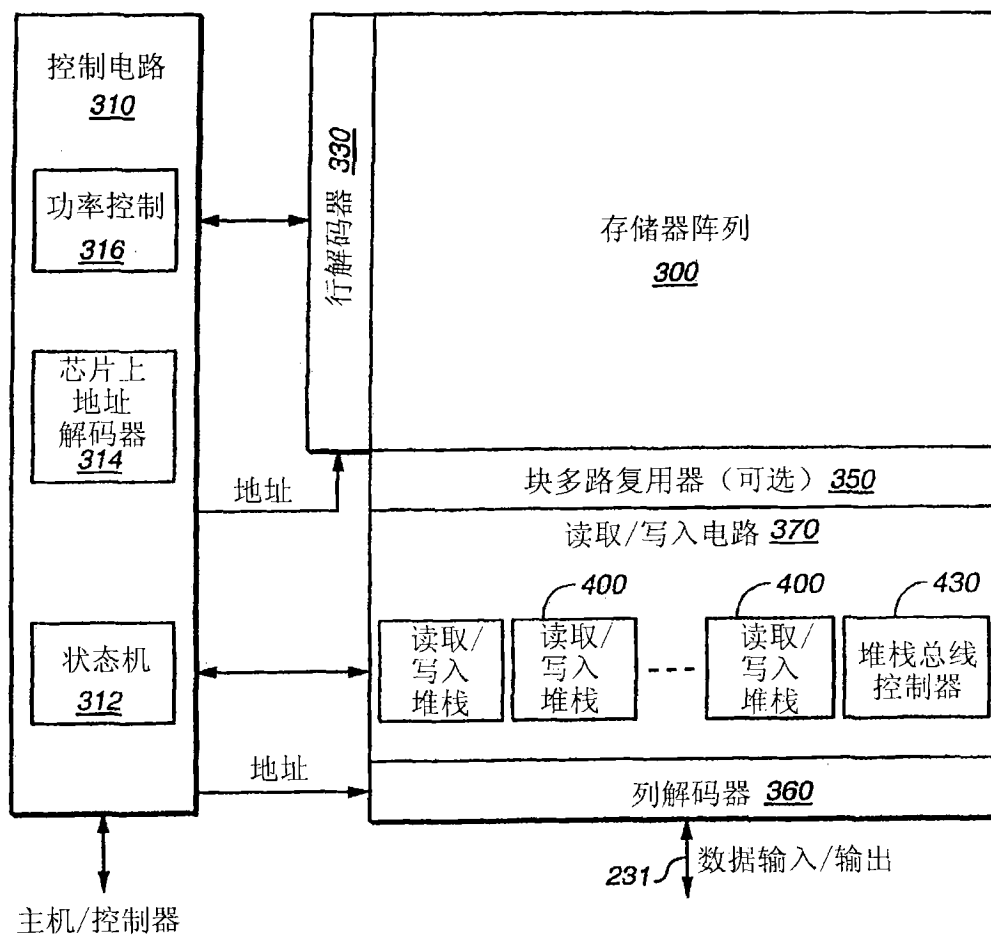


图 7A

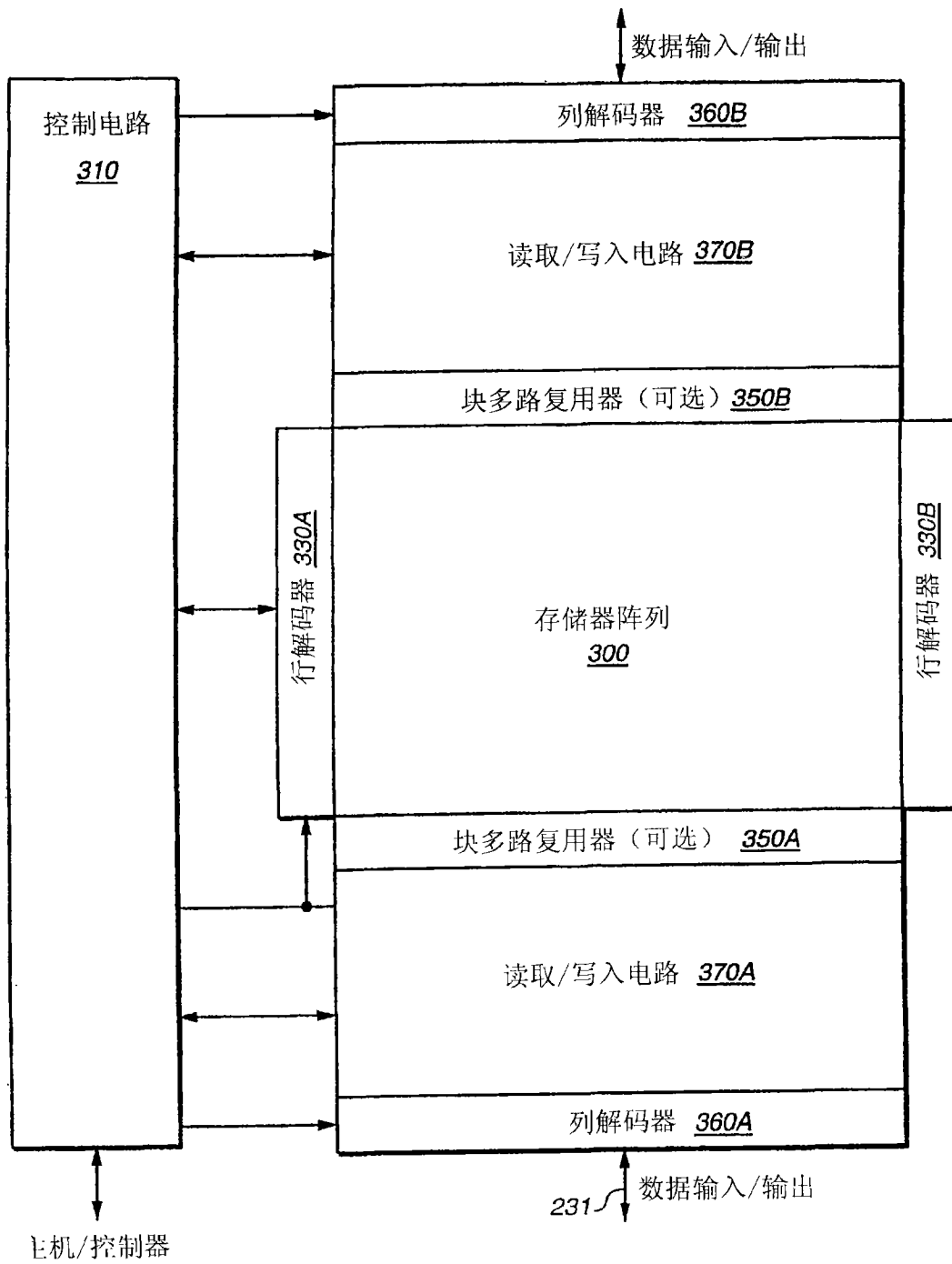


图 7B

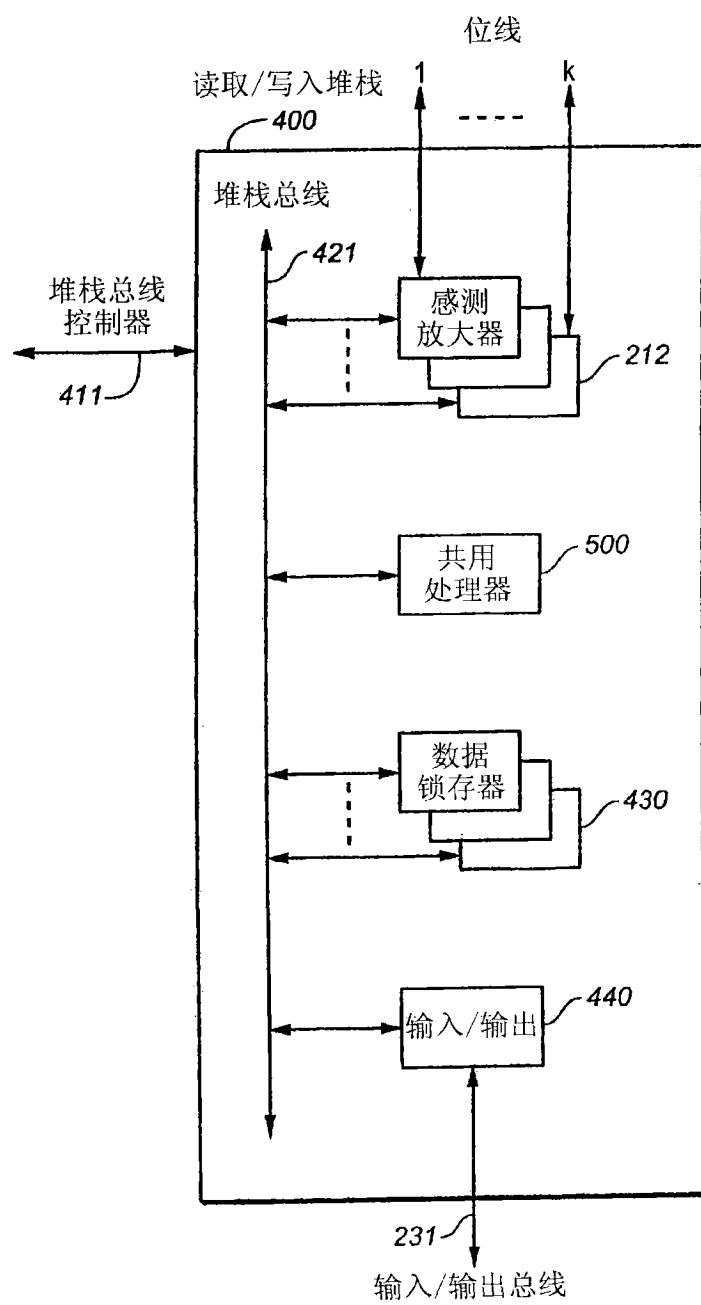


图 8

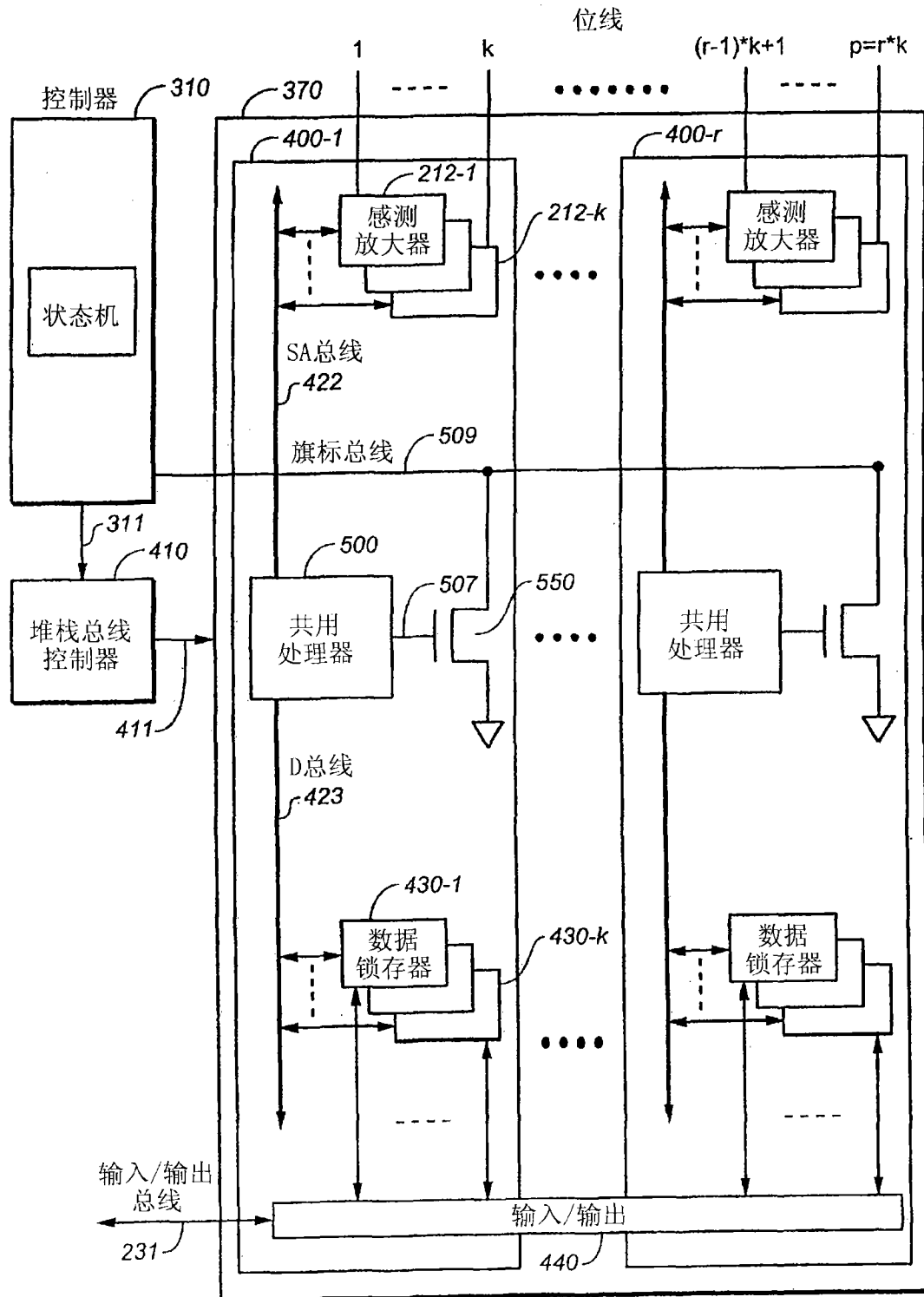


图 9

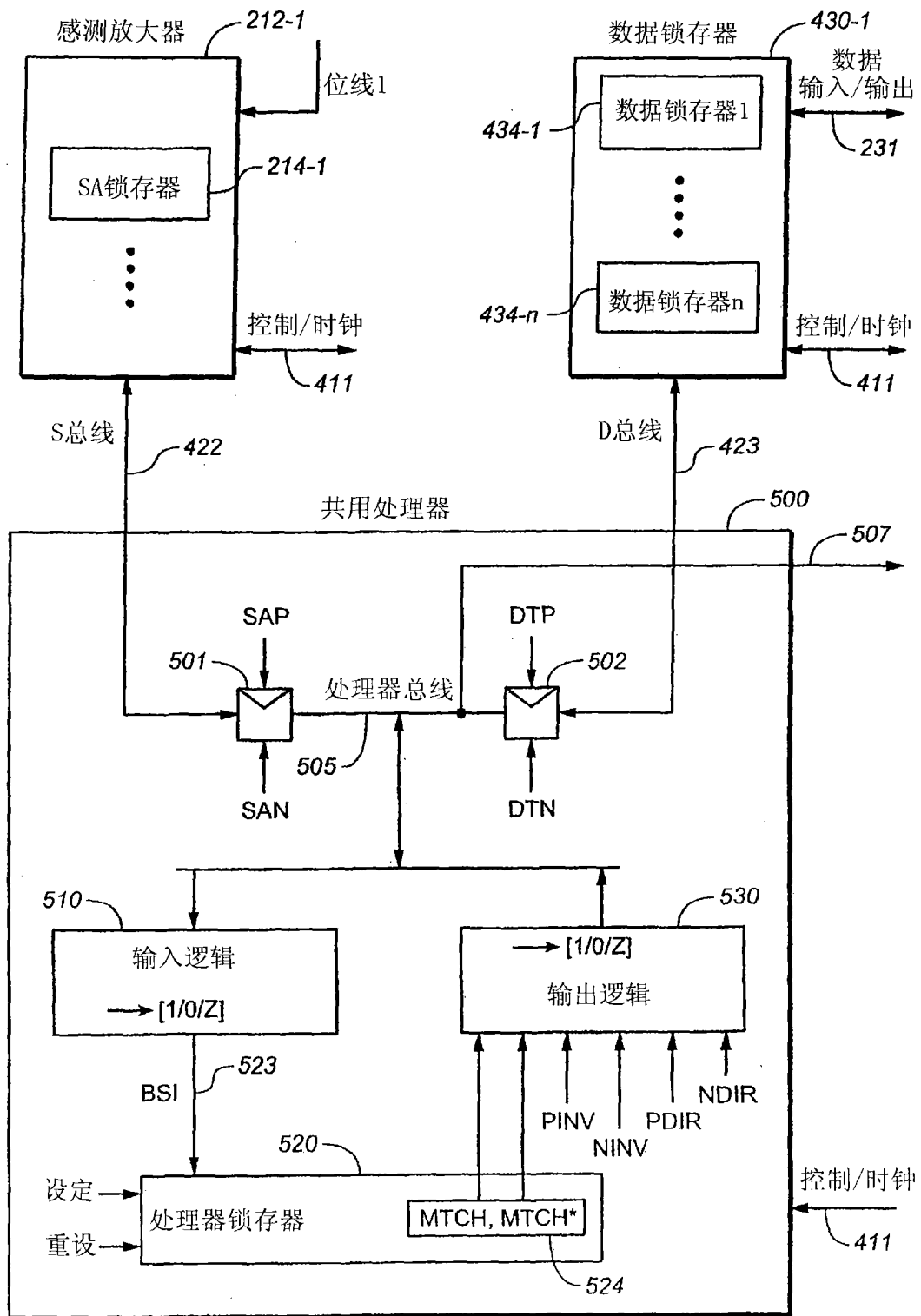


图 10

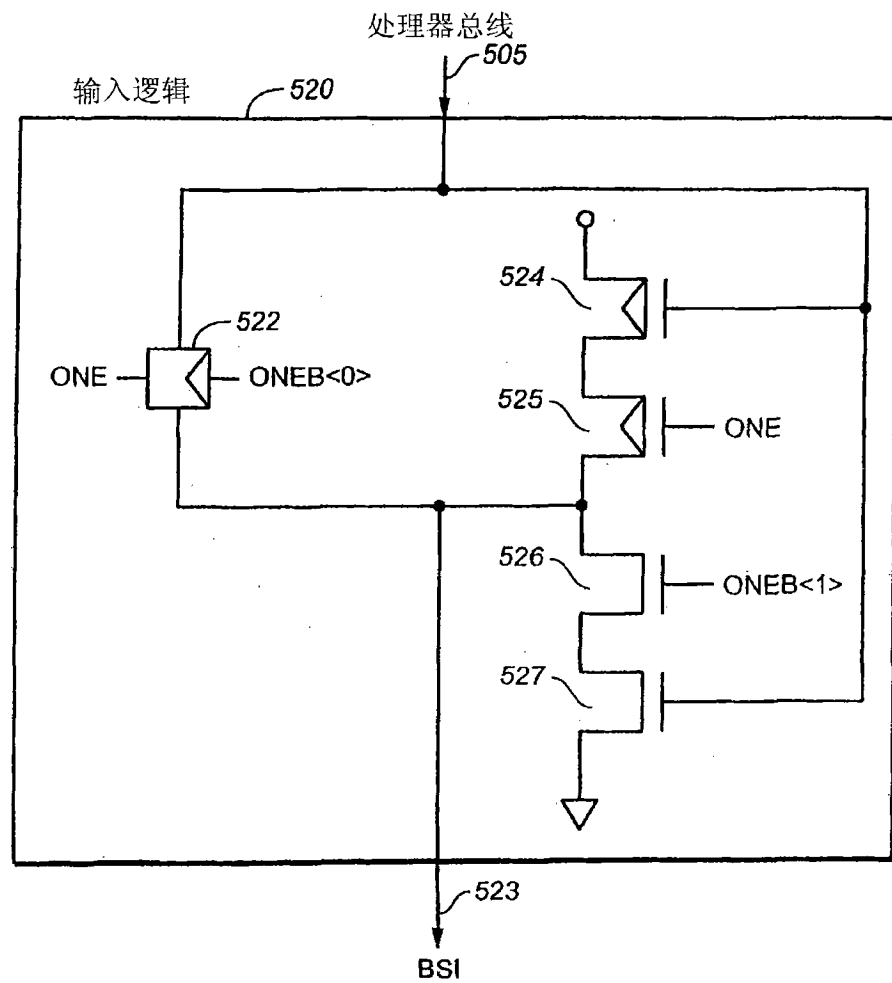


图 11A

输入逻辑真值表

转移模式	ONE	ONEB<0>	ONEB<1>	处理器总线 (输入)	BSI (输出)
通过	1	0	0	处理器 总线	处理器 总线
反相	0	1	1	处理器 总线	处理器 总线*
浮动	1	1	0	处理器 总线	浮动

图 11B

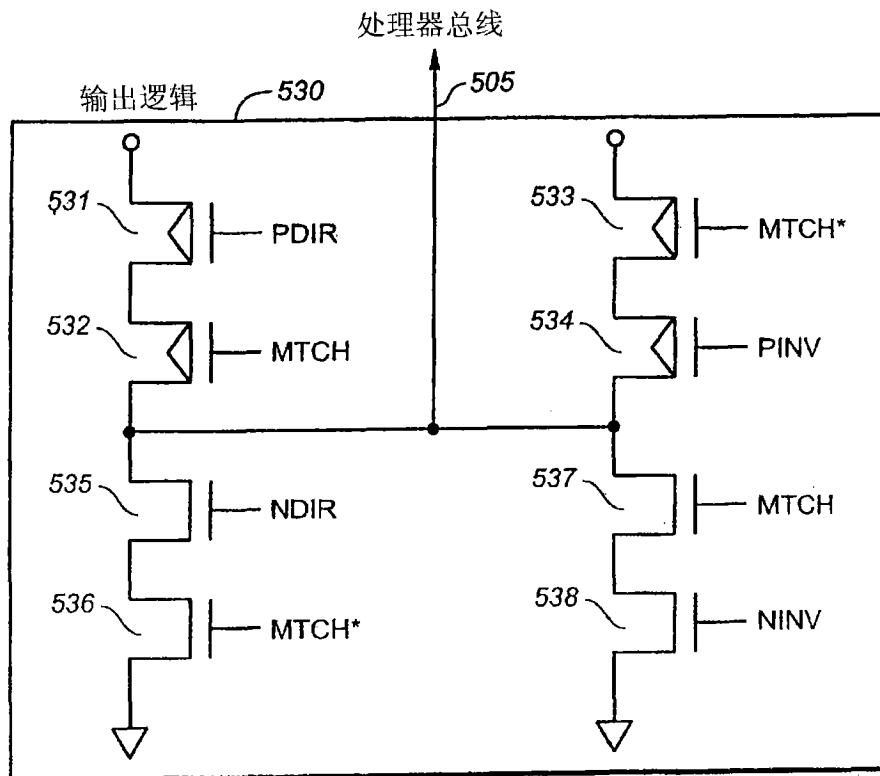


图 12A

输出逻辑真值表

转移模式	PINV	NINV	PDIR	NDIR	MTCH	处理器总线 (输出)
通过	D	D	D	1	0	0
	0	D	D	D	1	1
反相	D	D	0	D	0	1
	D	1	D	D	1	0
浮动	D	D	D	D	X	Z
预充电	0	D	0	D	X	1

(默认值: PINV=1, NINV=0, PDIR=1, NDIR=0)

图 12B

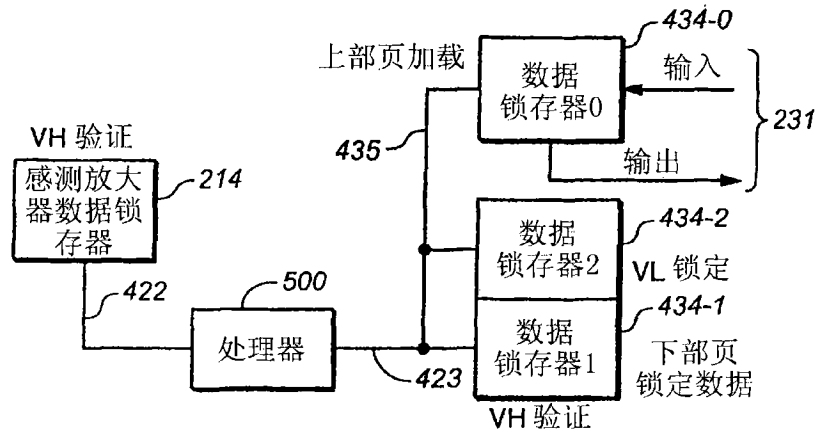


图 13

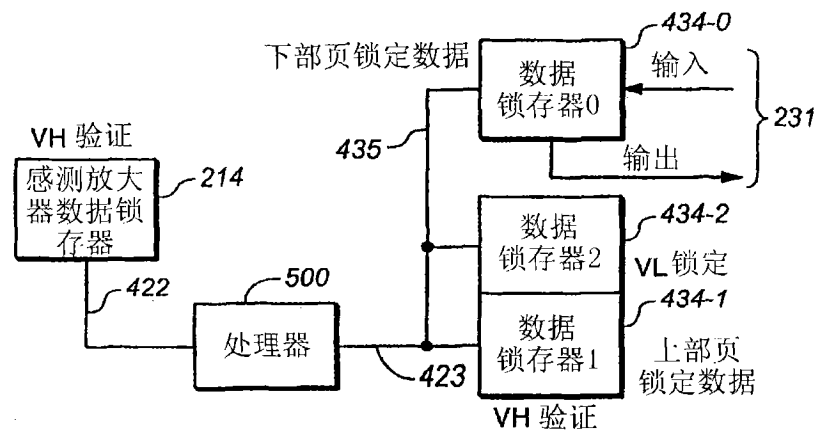


图 14

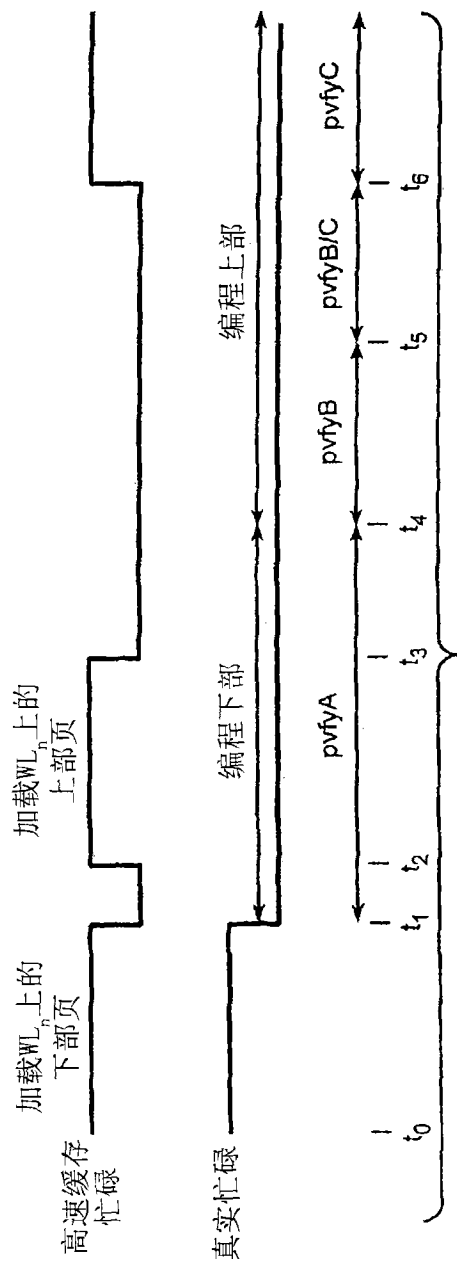


图15

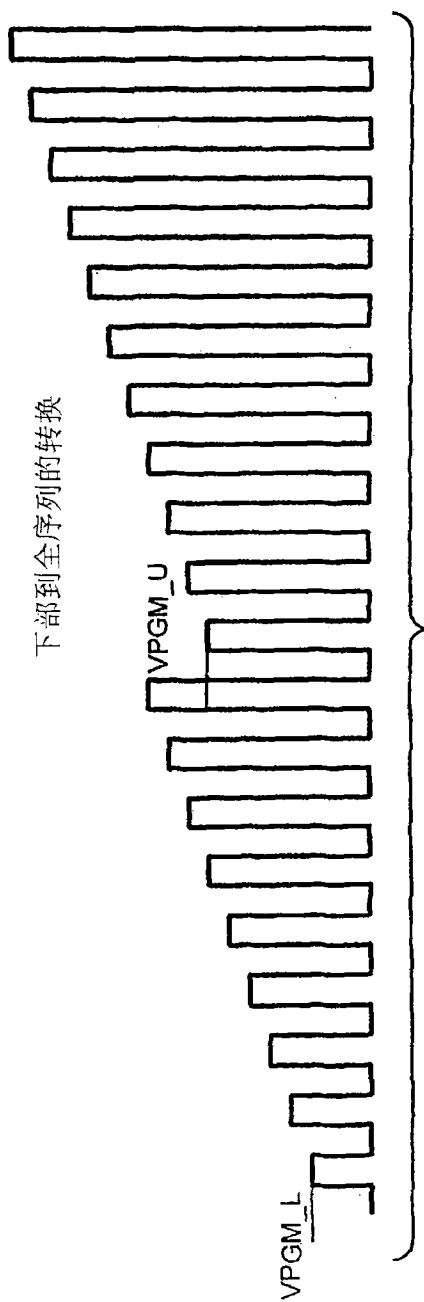
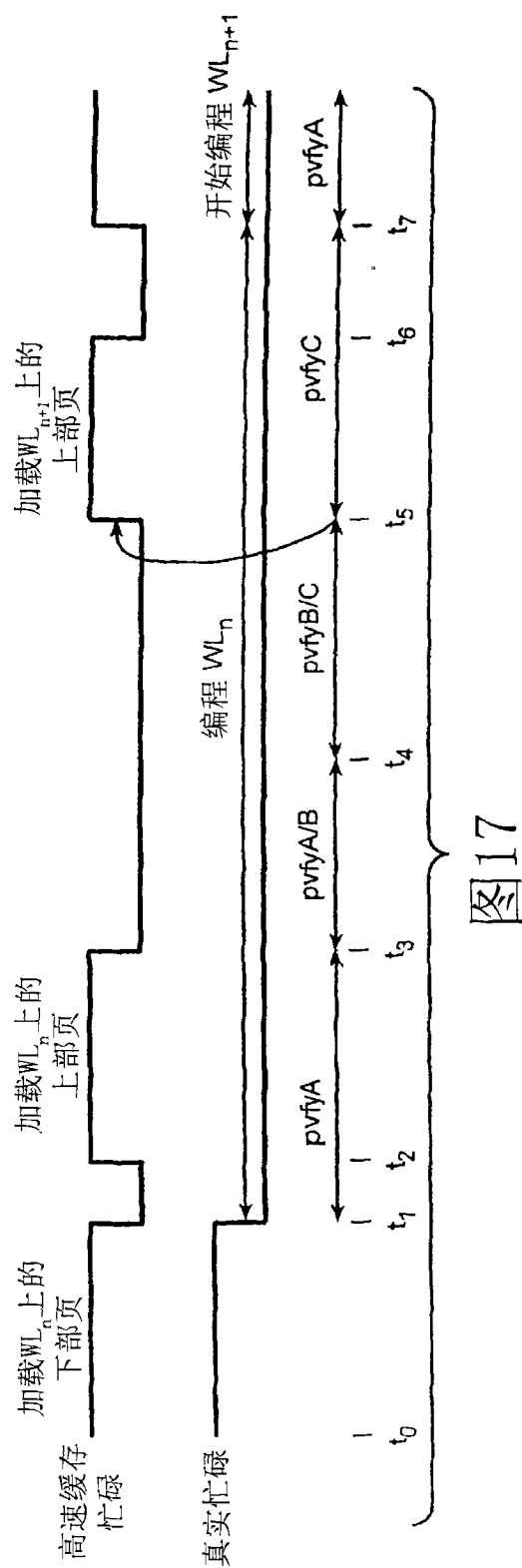


图16



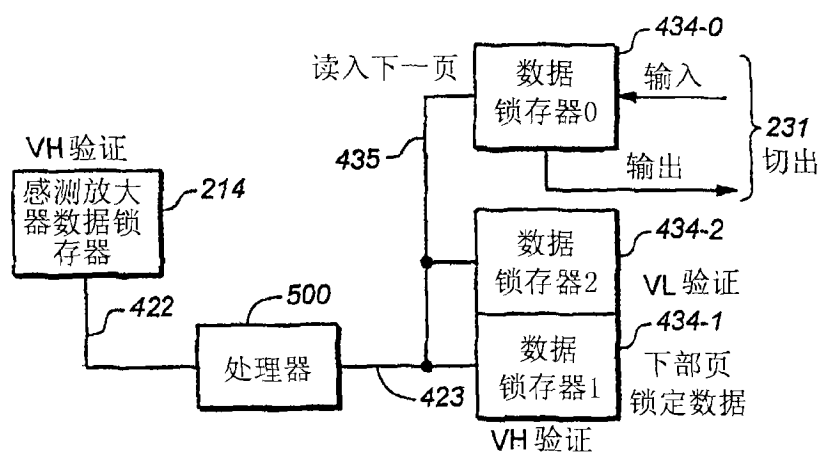
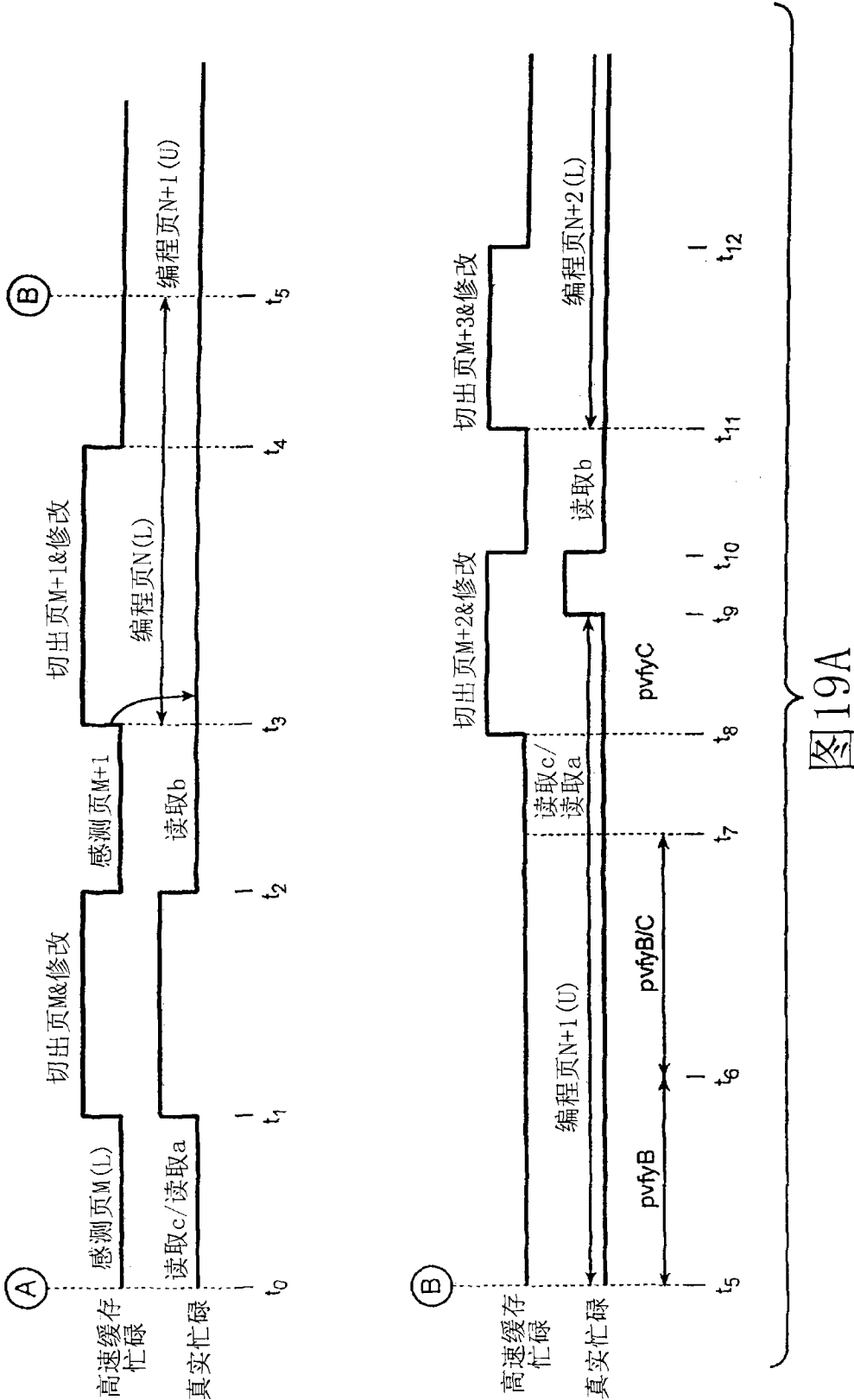


图18



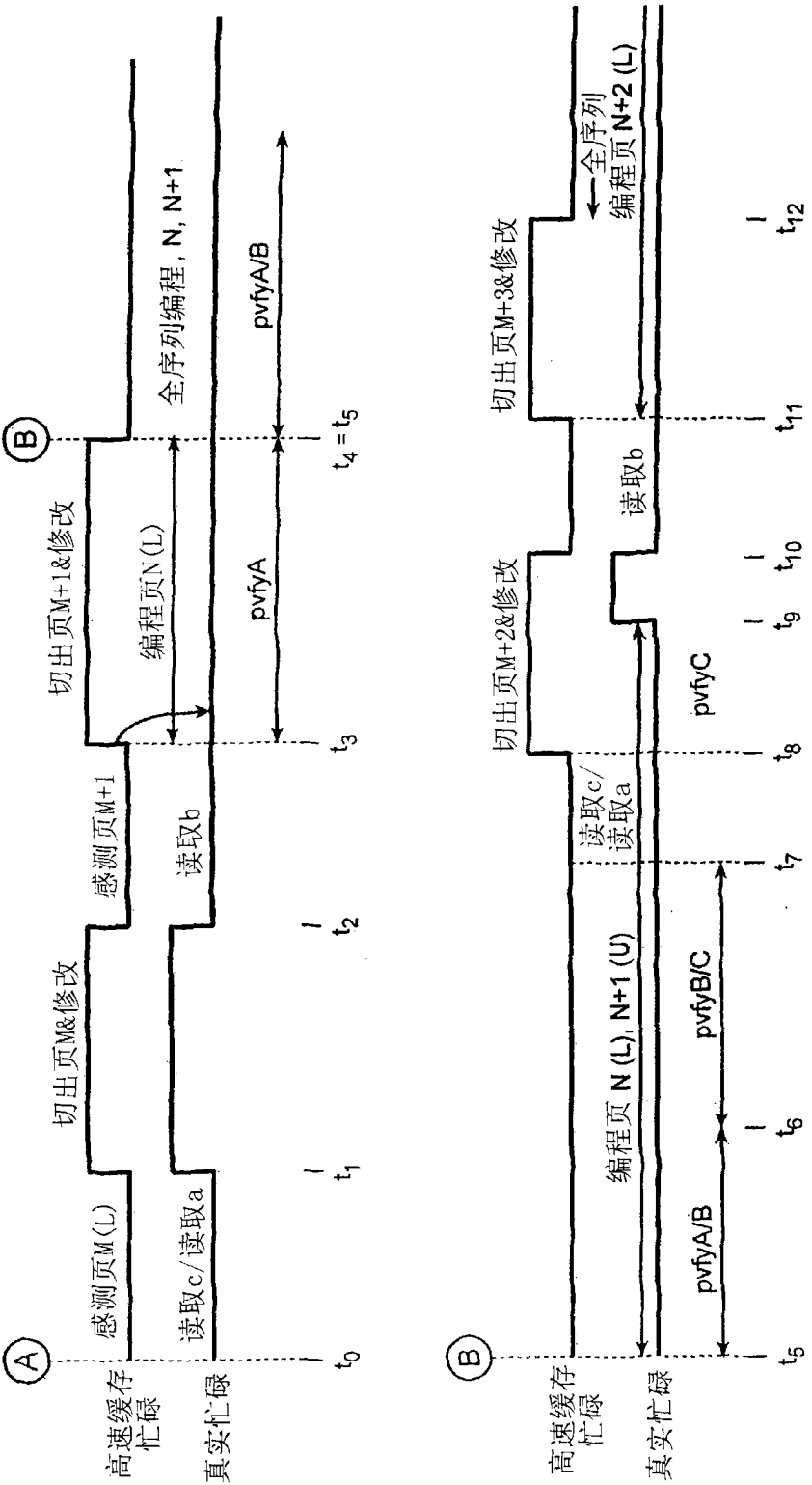


图19B

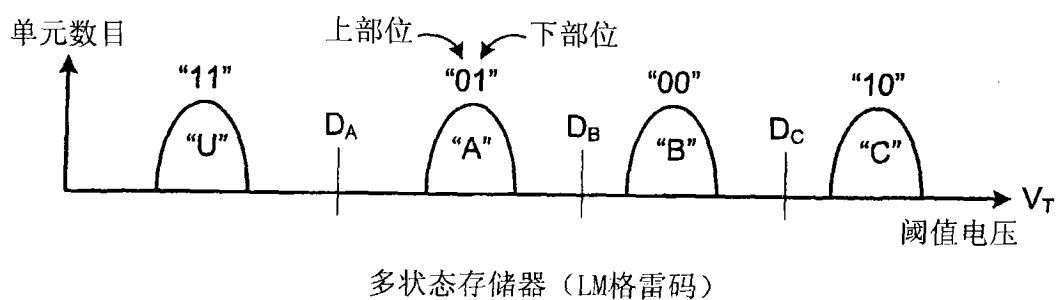


图 20A

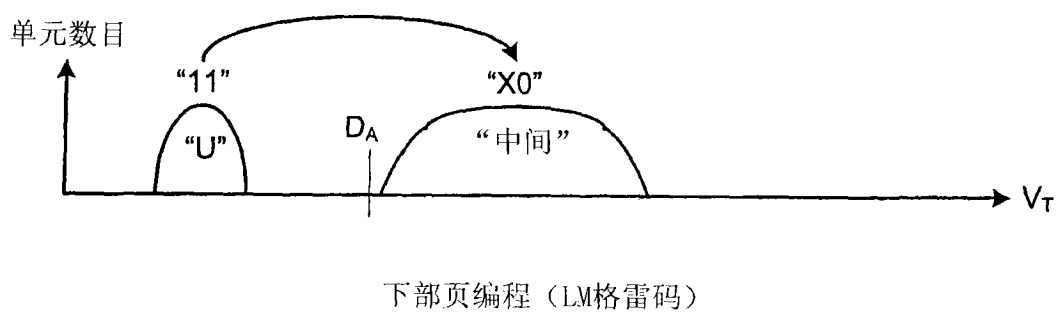


图 20B

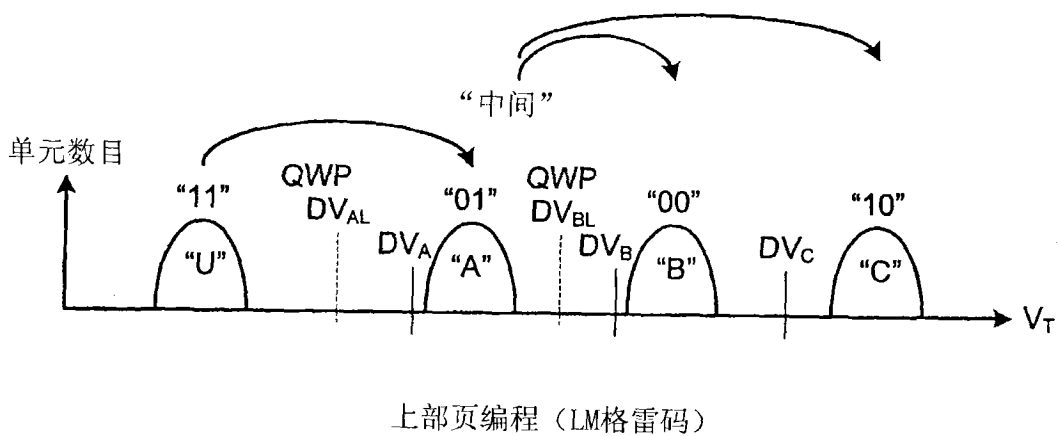
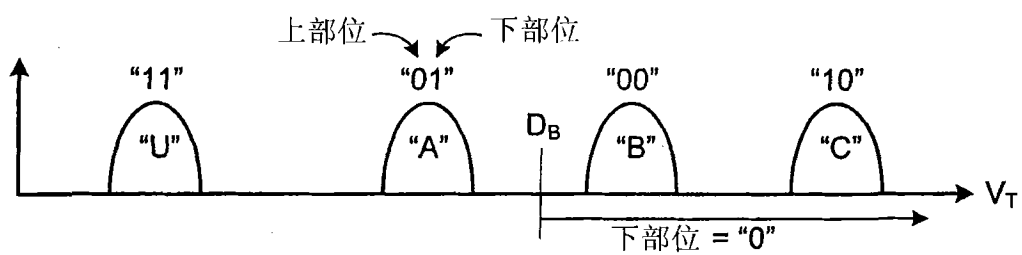
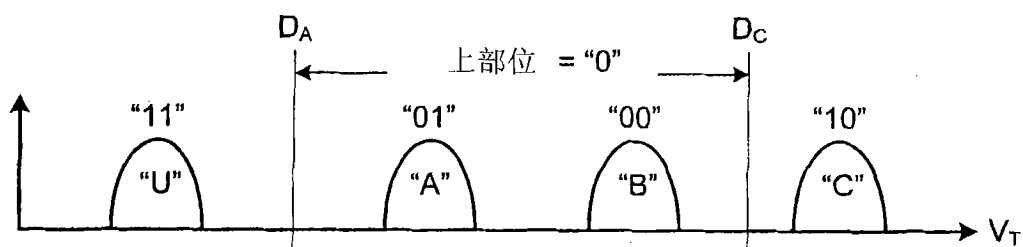


图 20C



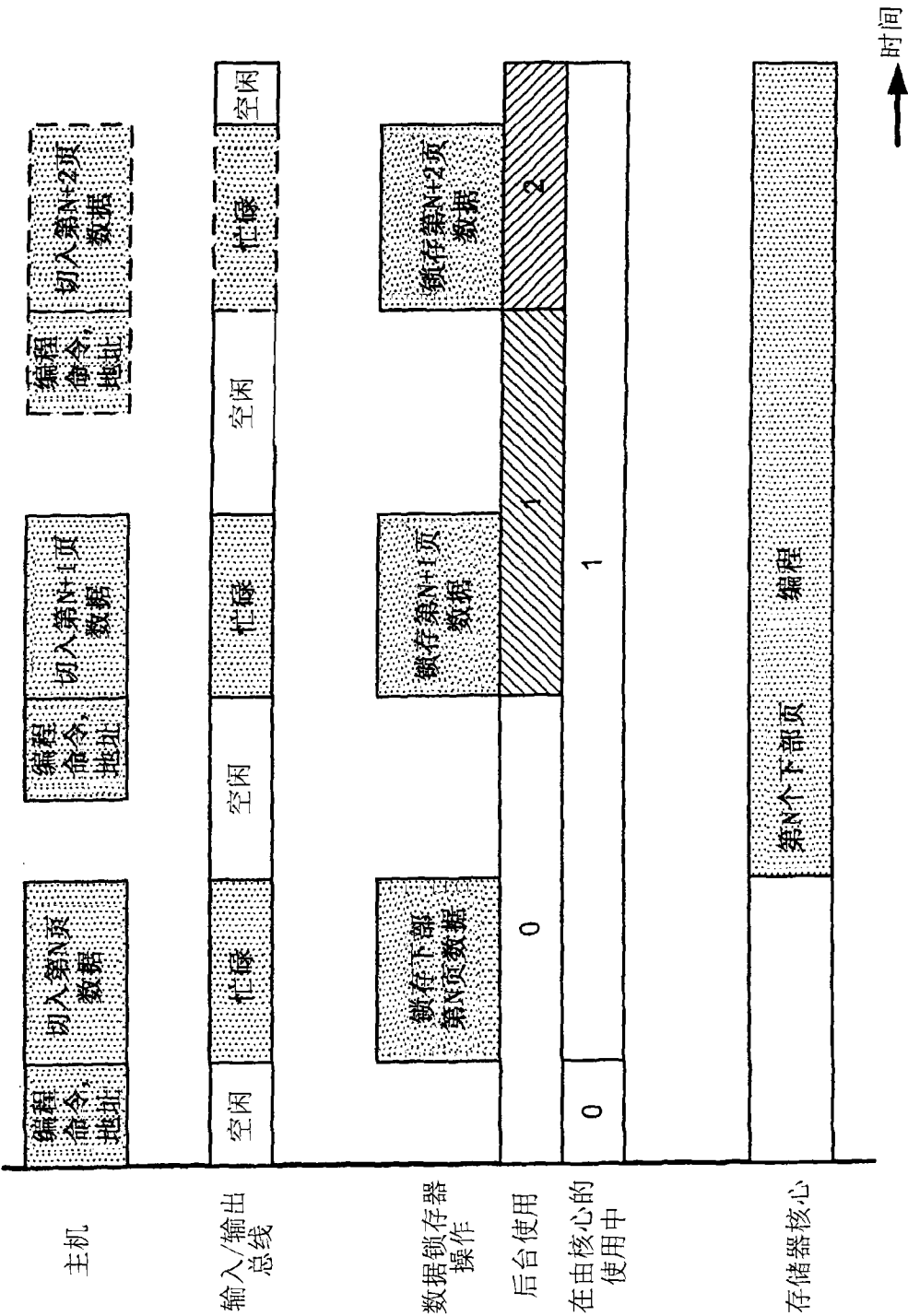
下部页读取 (LM格雷码)

图 20D



上部页读取 (LM格雷码)

图 20E



下部页编程期间的后台下一页数据加载

图21

编程阶段	所验证的目标 状态	待跟踪以进行编程的状态	所需锁存器 的数目	代码表
"I" (初始)	无	"L", "A _L ", "A", "B _L ", "B", "C"	3	3CT()
"A" 完成	"A"	"L", "B _L ", "B", C	2	2CT("A")
"B" 完成	"B"	"L", "A _L ", "A", C	2	2CT("B")
"C" 完成	"C"	"L", "A _L ", "A", "B _L ", "B"	3	3CT()
"A"+"B" 完成	"A", "B"	"L", "C"	1	1CT("A"+"B")
"A"+"C" 完成	"A", "C"	"L", "B _L ", "B"	2	2CT("A"+"C")
"B"+"C" 完成	"B", "C"	"L", "A _L ", "A"	2	2CT("B"+"C")
"A"+"B"+"C" 完成	"A", "B", "C"	无	0	无

"L" = 编程锁定

各种编程阶段期间对状态的上部页或全序列编程跟踪

图22

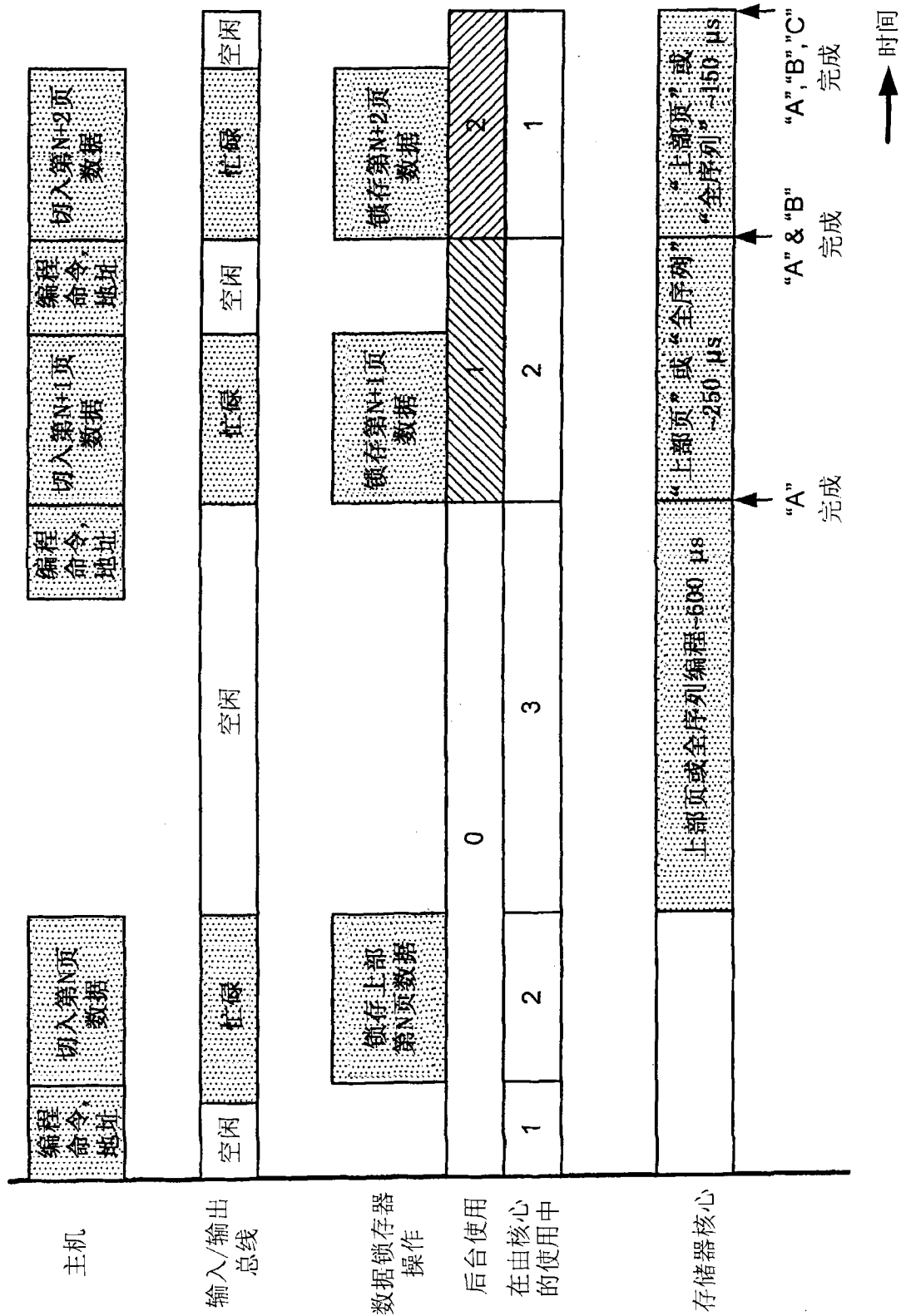


图23

上部页或全序列编程期间的后台下一页数据加载

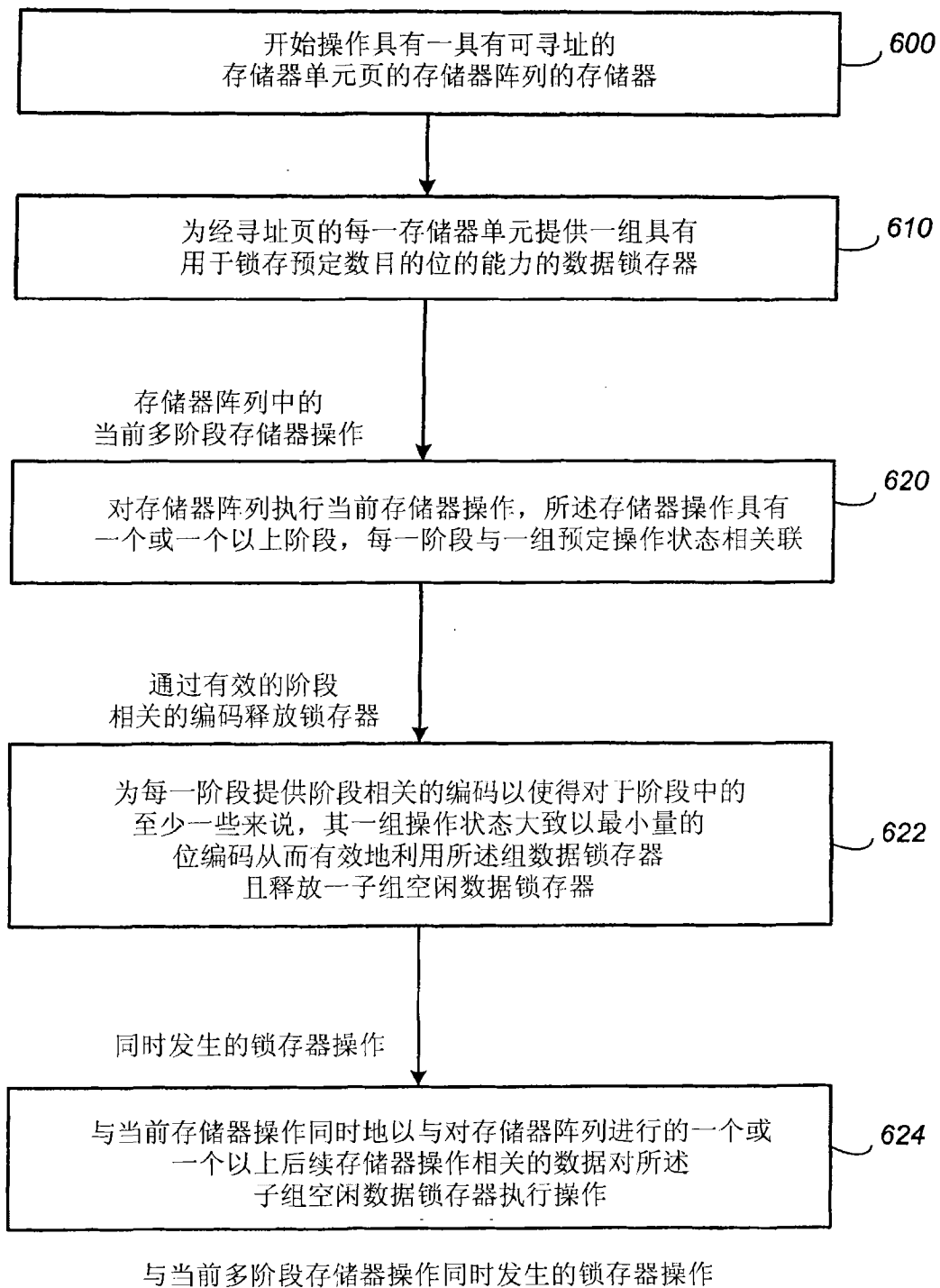
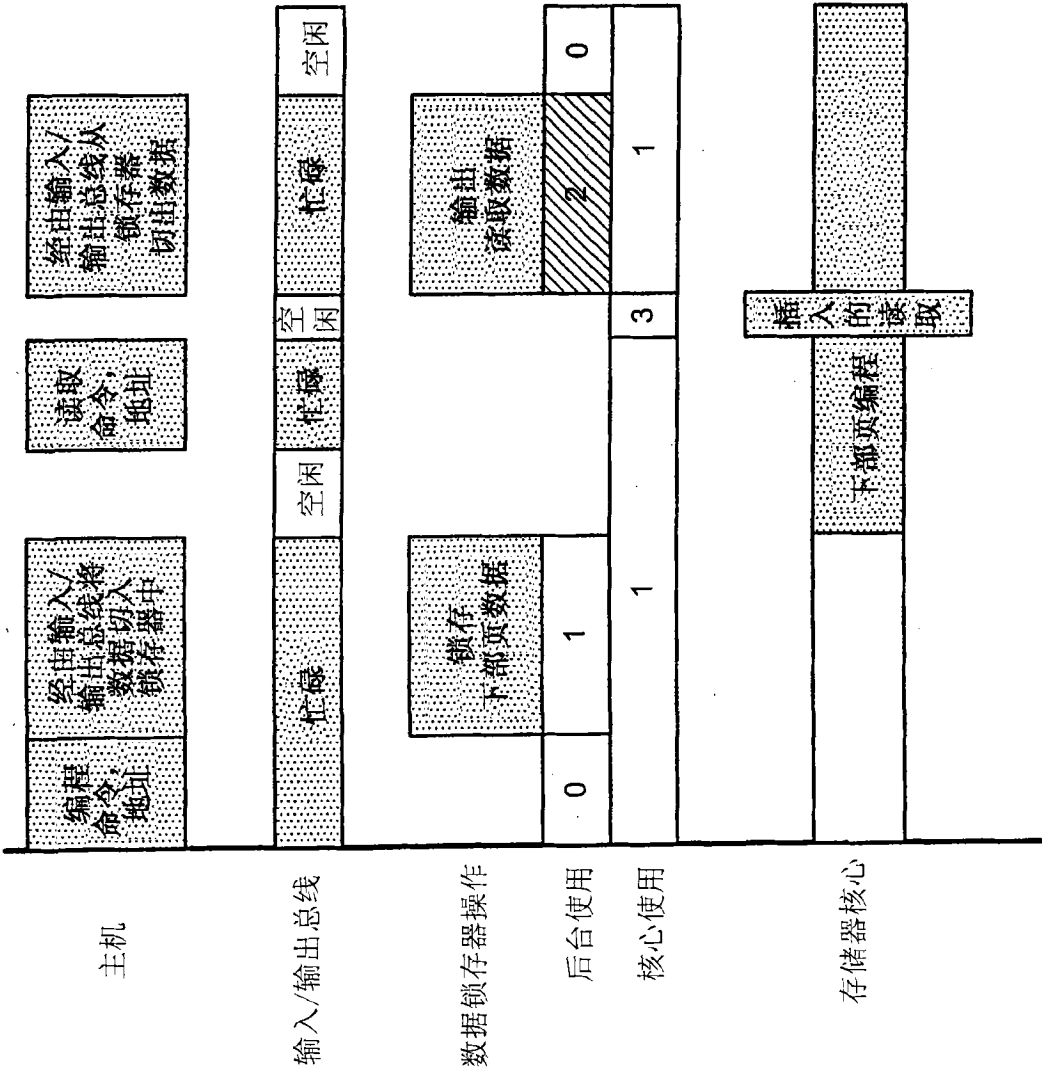
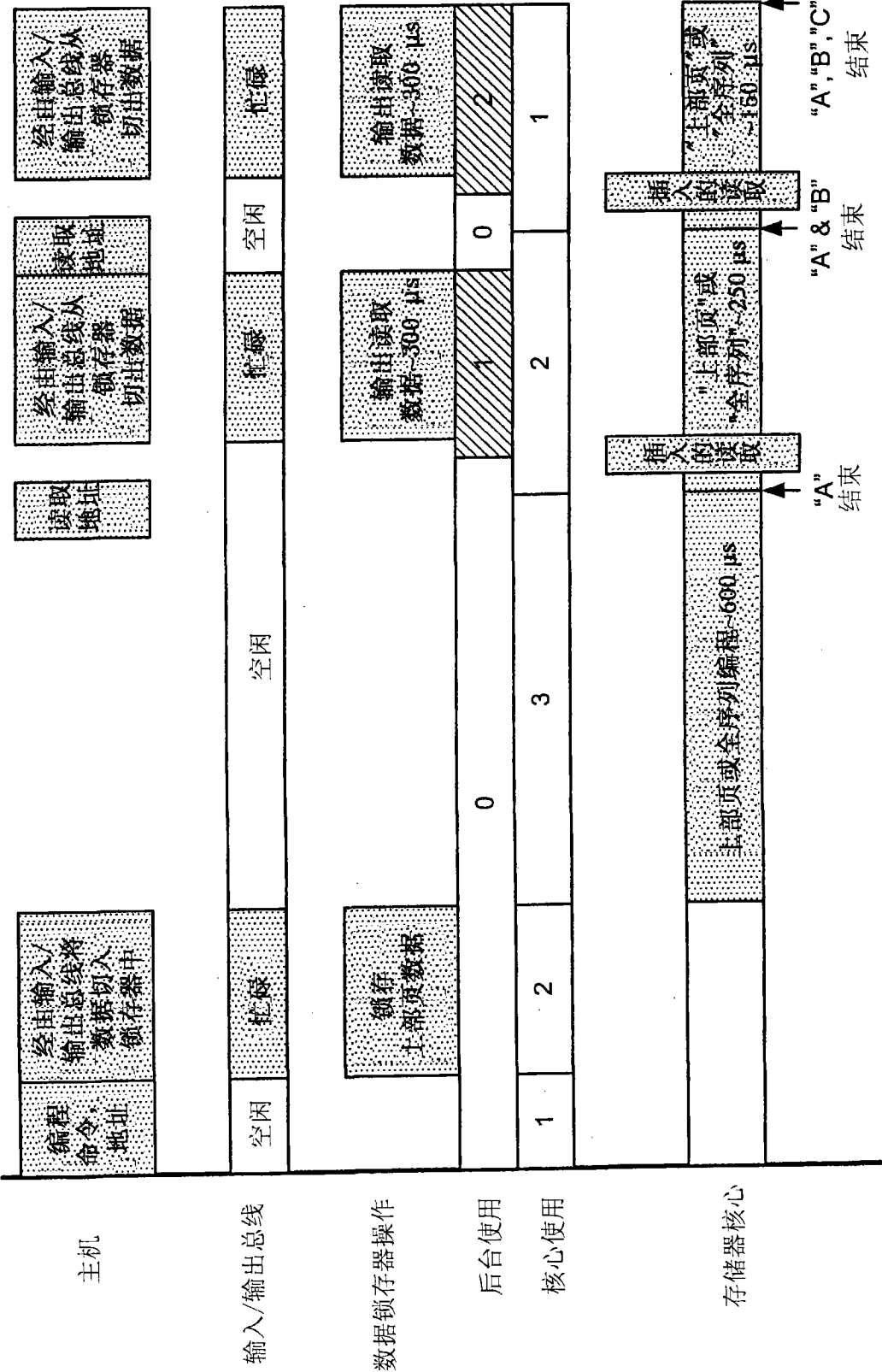


图 24



下部页编程期间的后台读取

图25



上部页或全序列编程期间的后台读取

图26

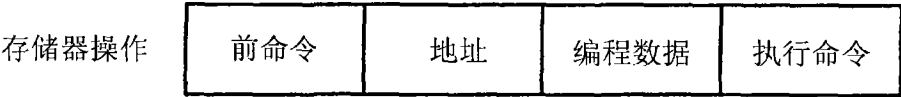
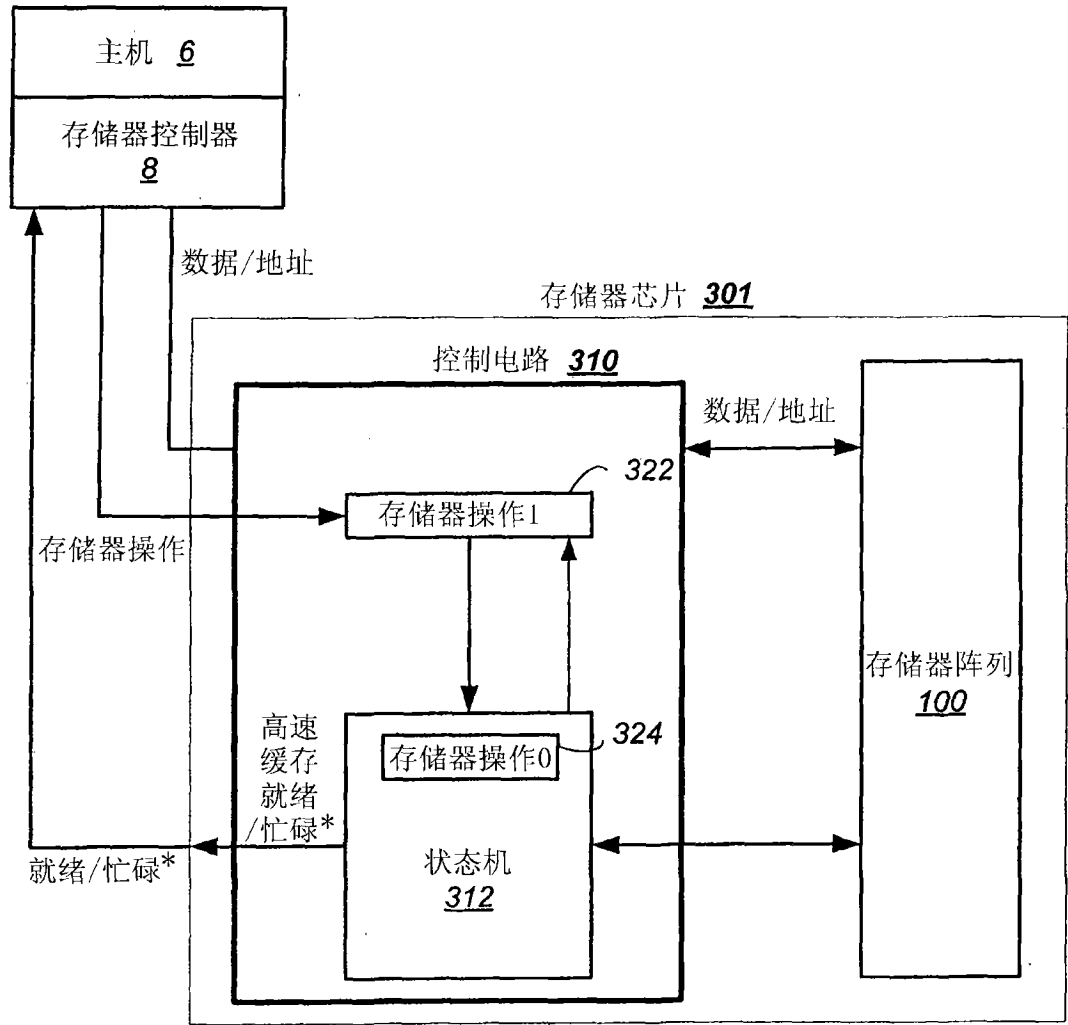
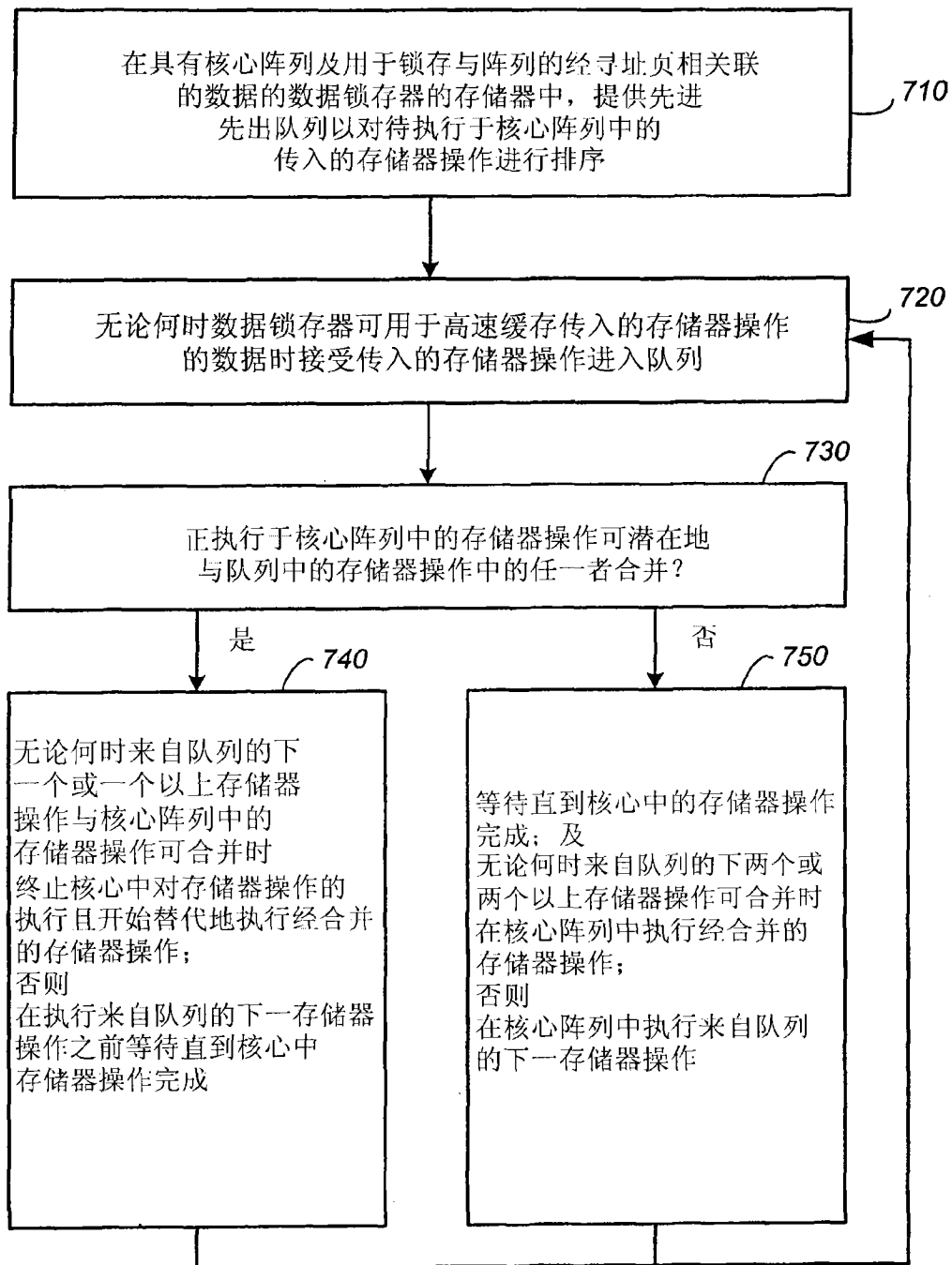


图 27



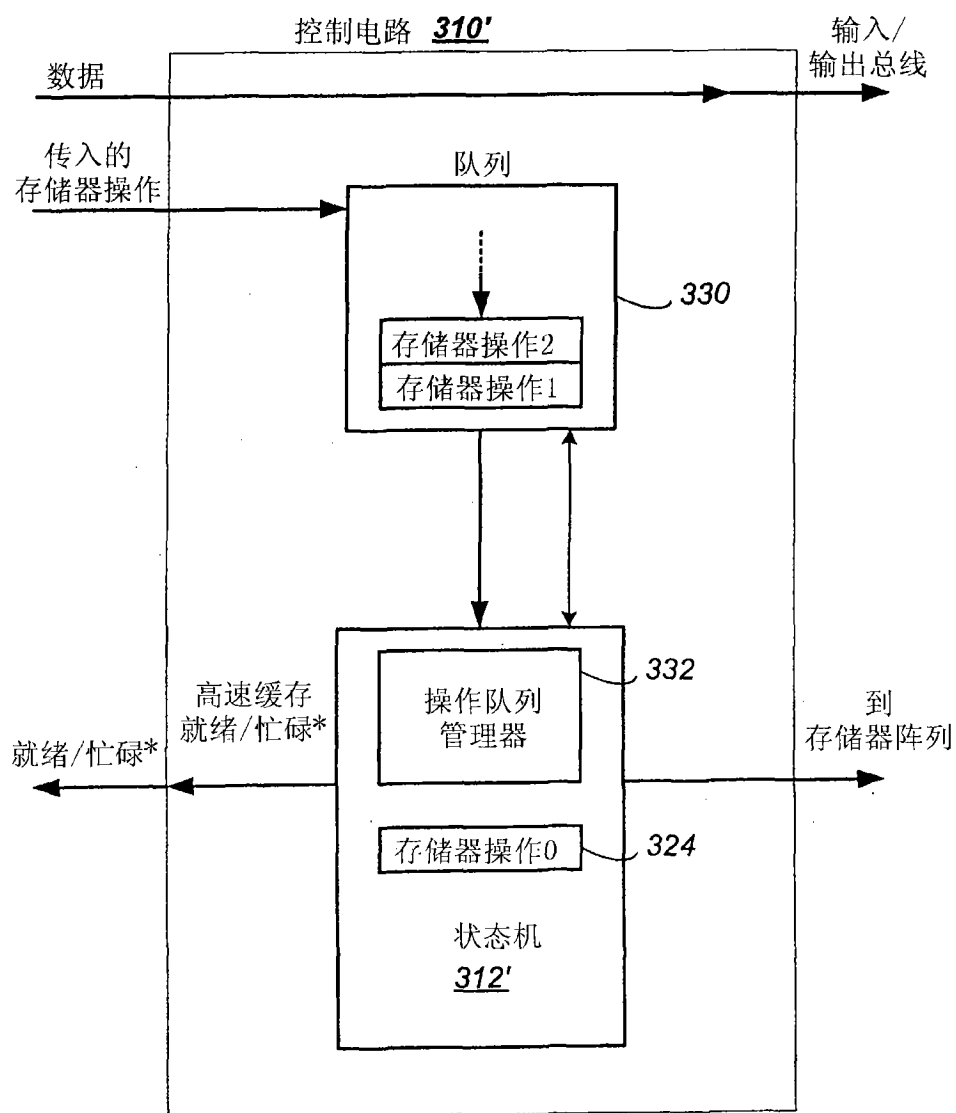
常规高速缓存存储器操作

图 28



多个存储器操作管理

图 29



对多个存储器操作的管理

图 30

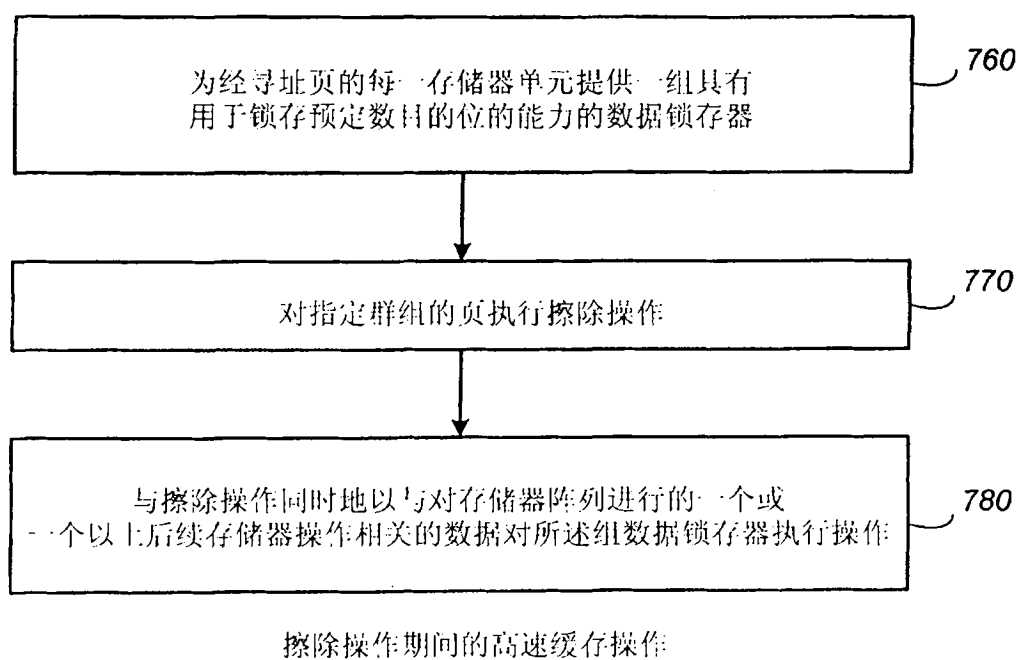
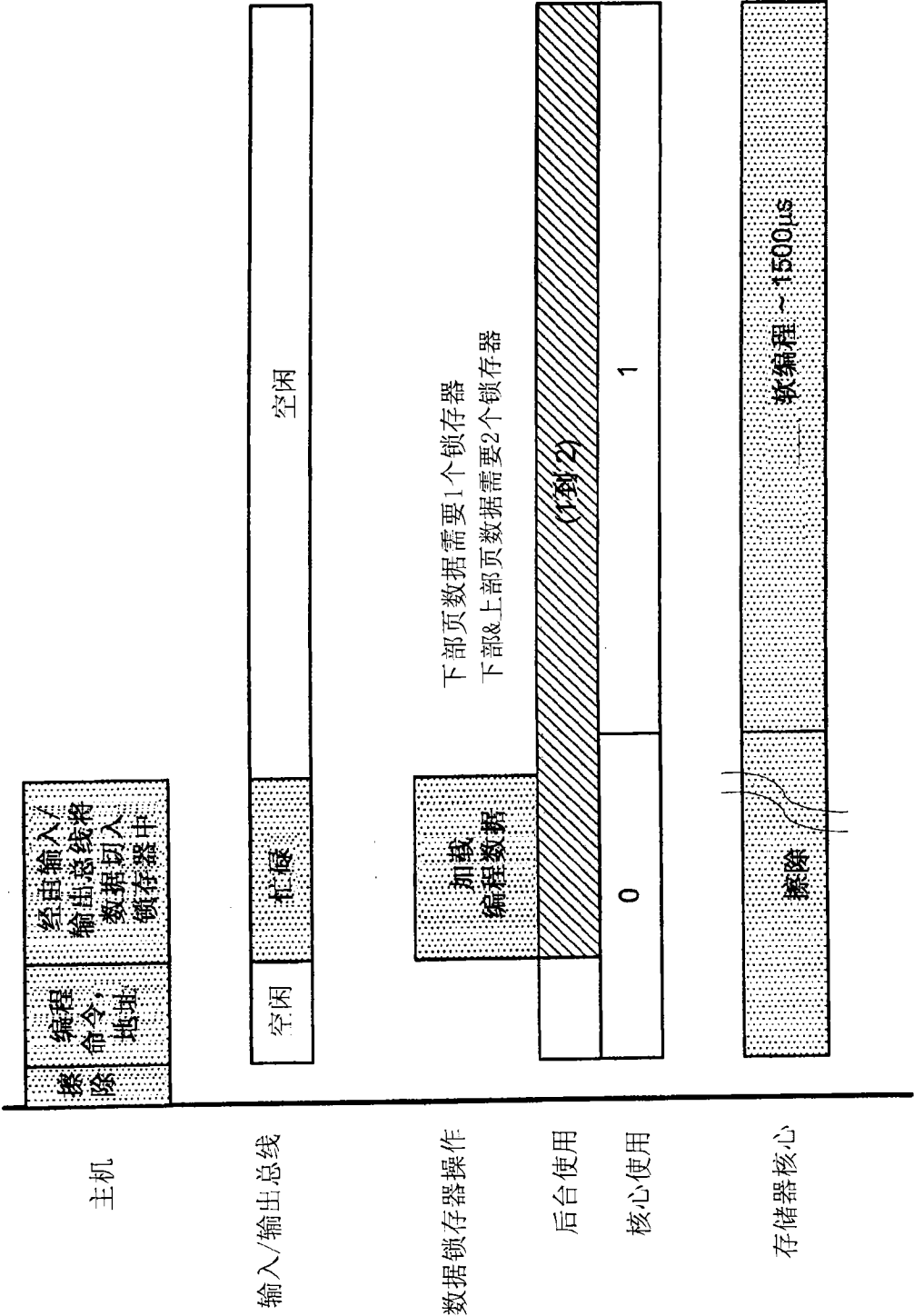
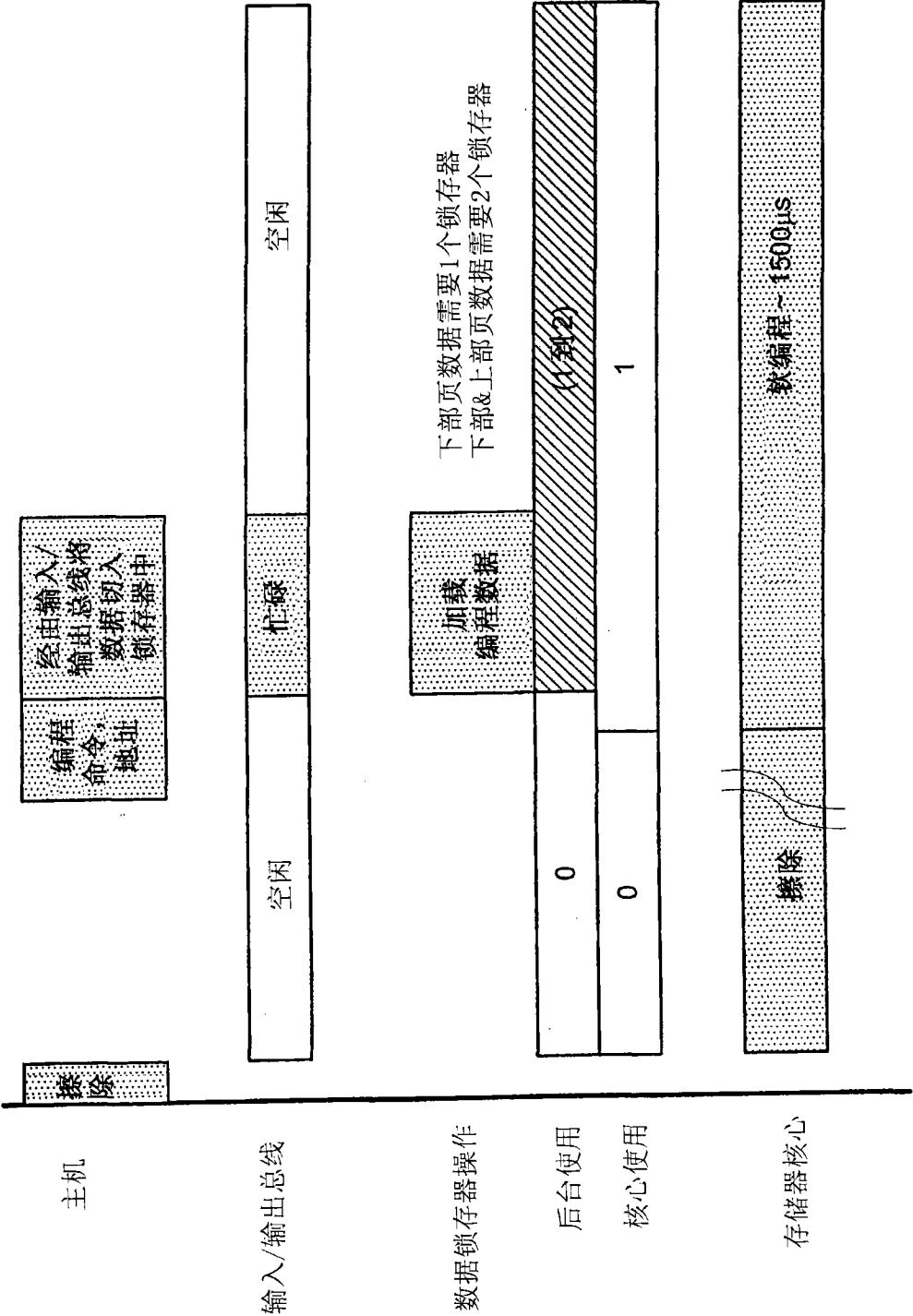


图 31



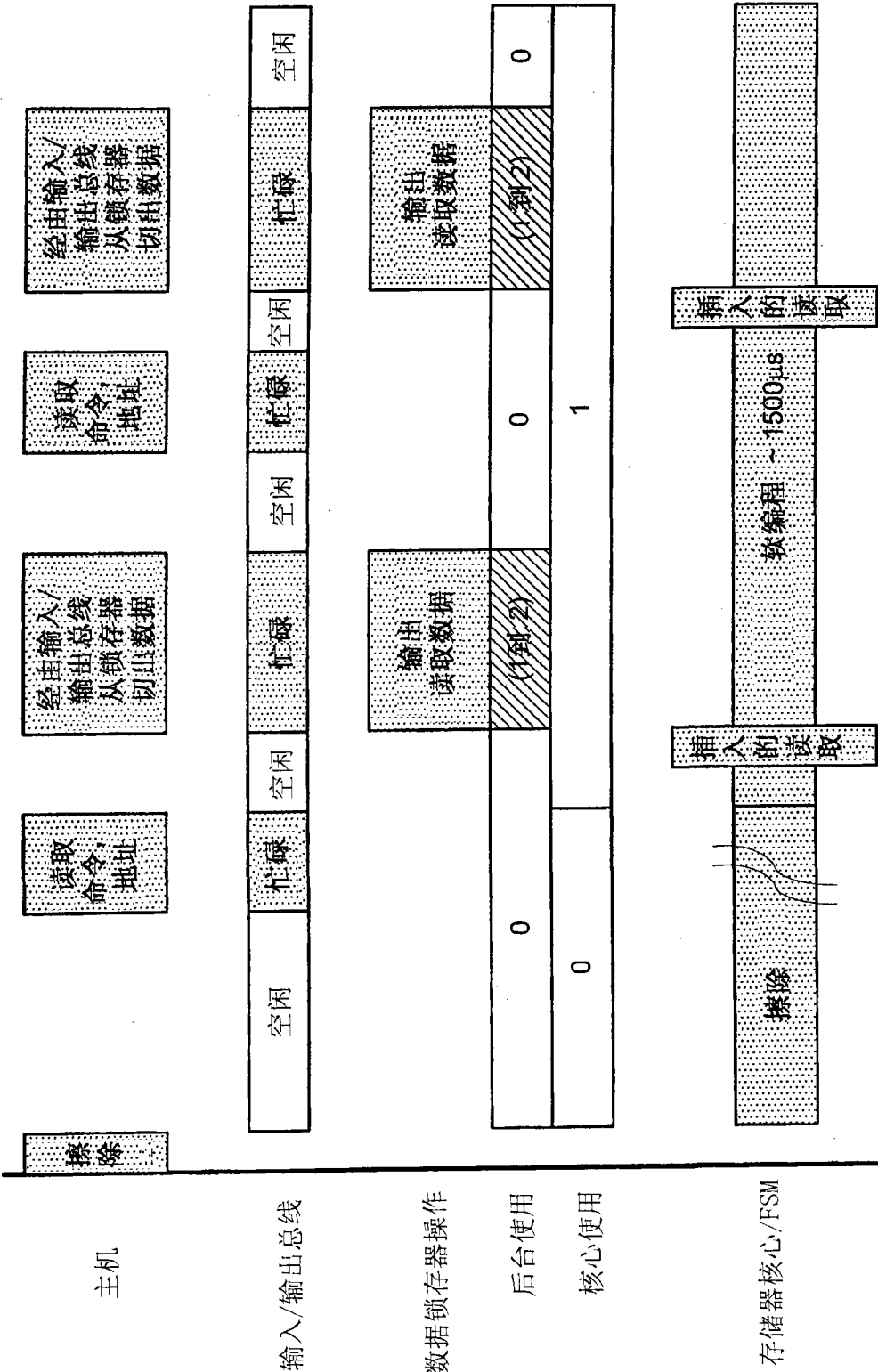
擦除阶段期间的后台编程数据加载

图32



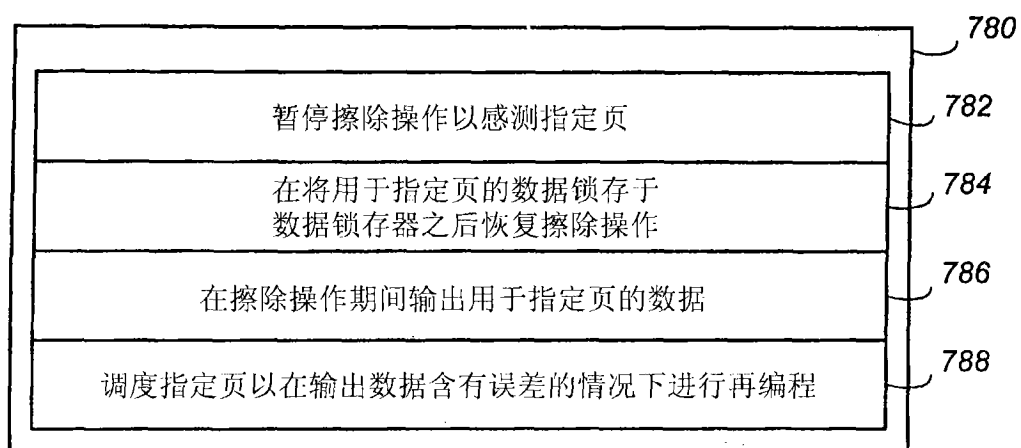
软编程阶段期间的后台编程数据加载

图33



擦除期间的后台读取

图34



擦除操作期间的高速缓存读取擦洗操作

图 35

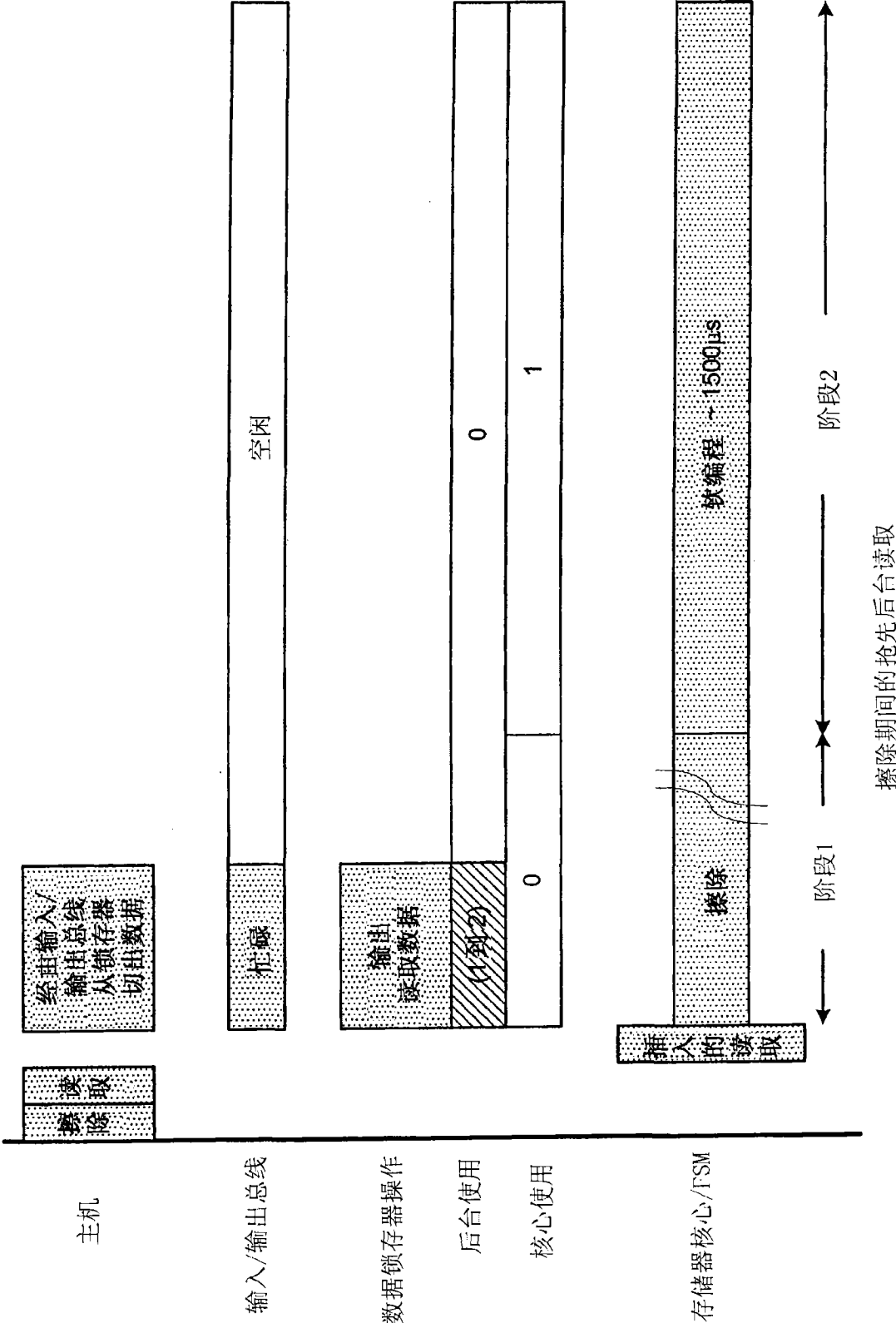


图36

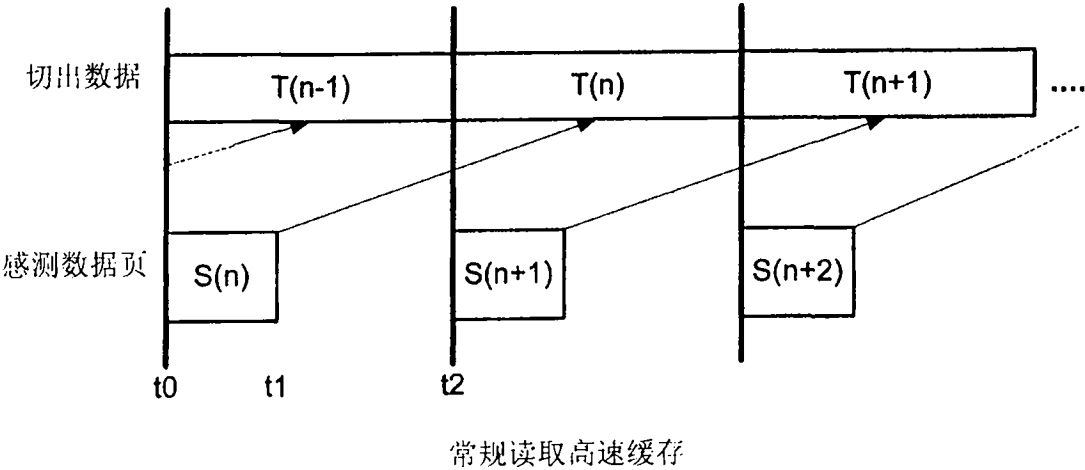


图 37

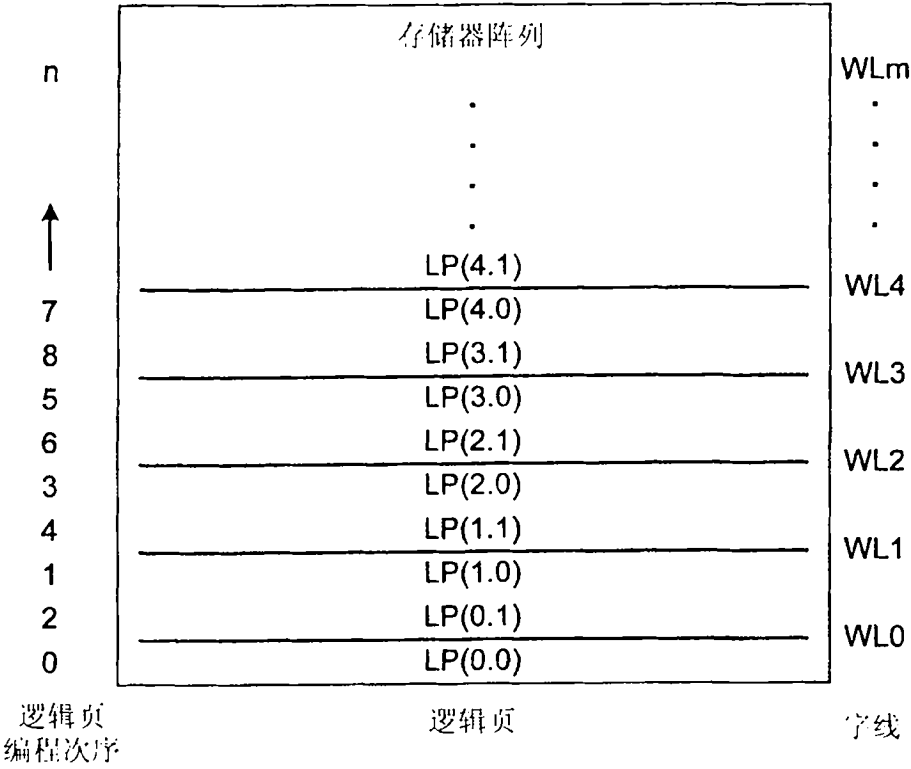
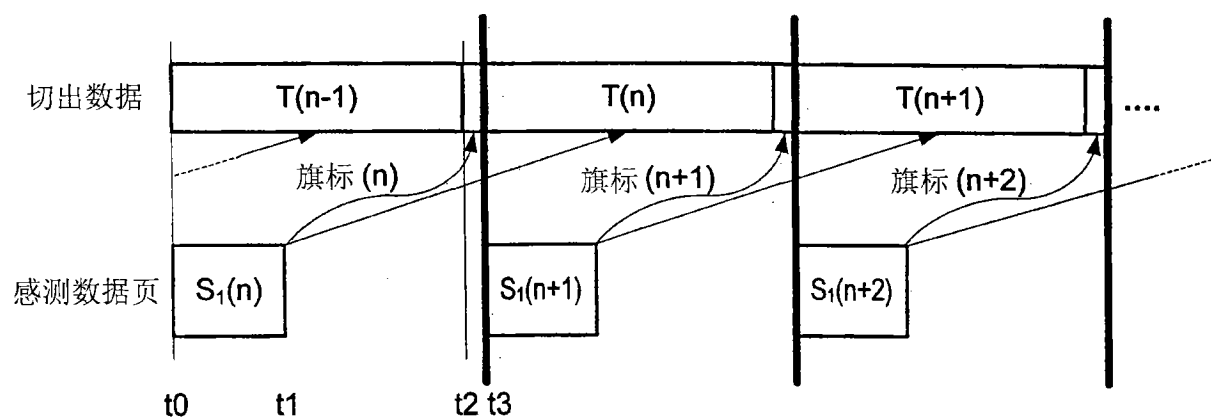
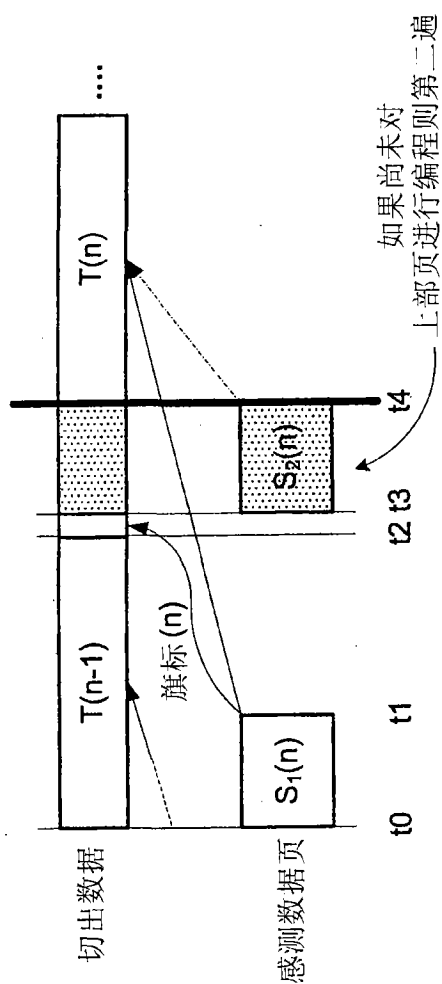


图 40



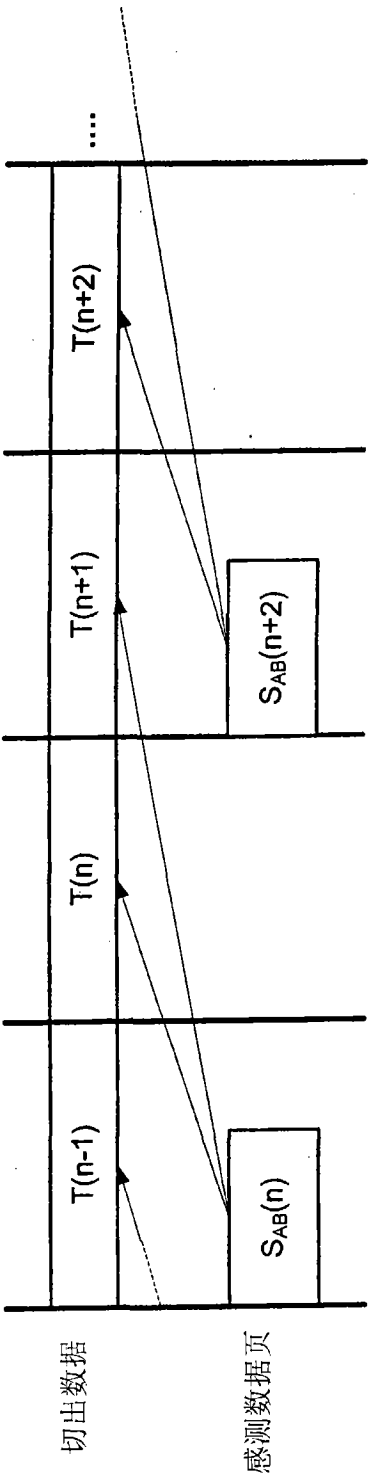
以LM编码进行的读取高速缓存

图38A



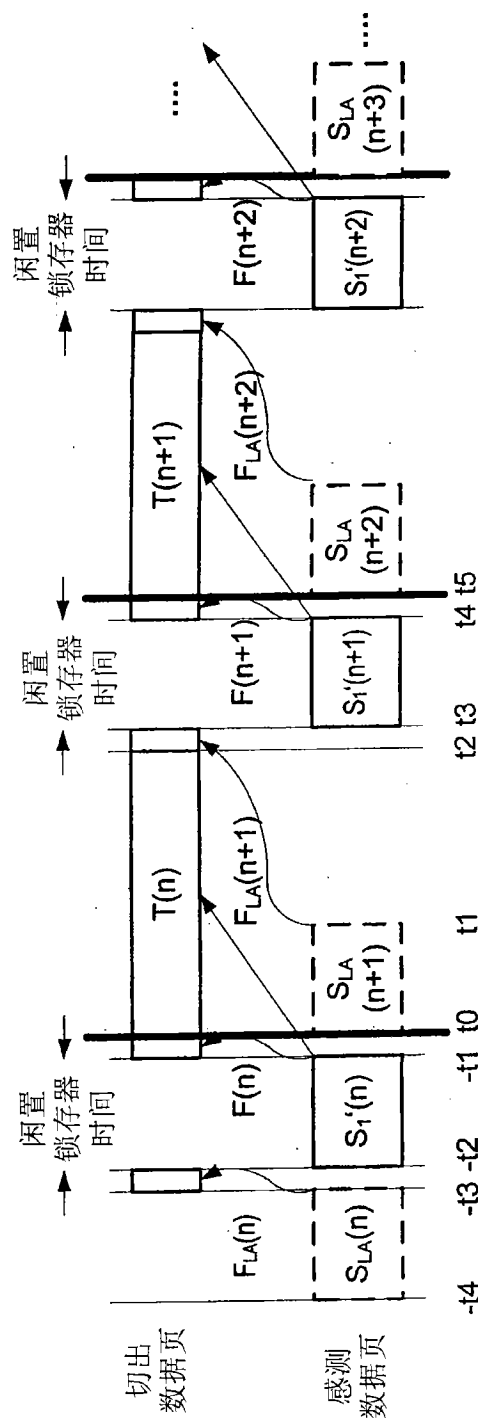
以LM编码进行的读取高速缓存
(在尚未对上部页进行编程时读取下部逻辑页的特殊情形)

图38B



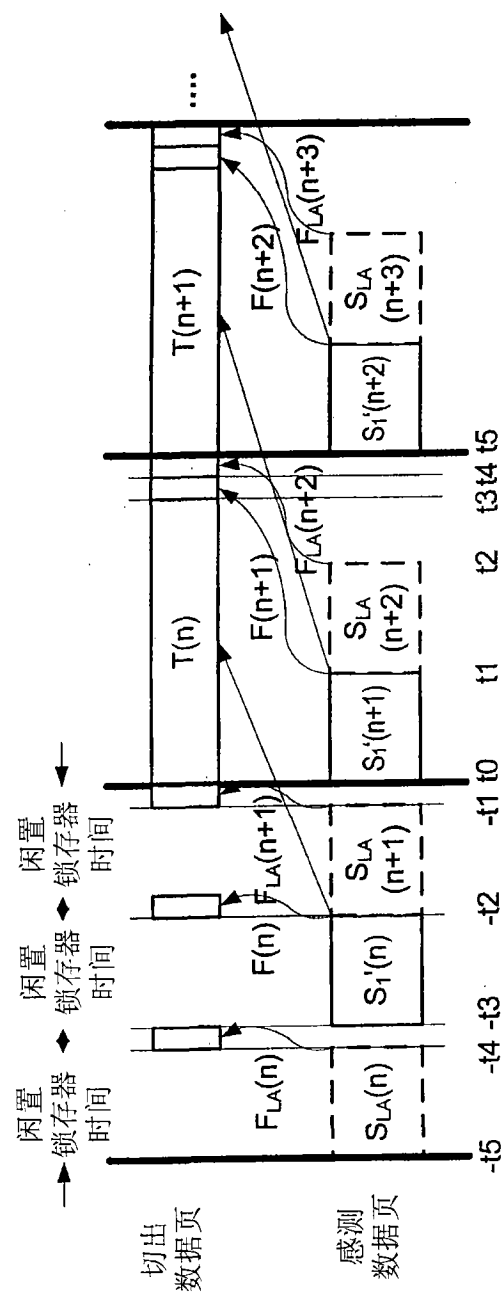
以所有位感测而进行的高速缓存读取

图39



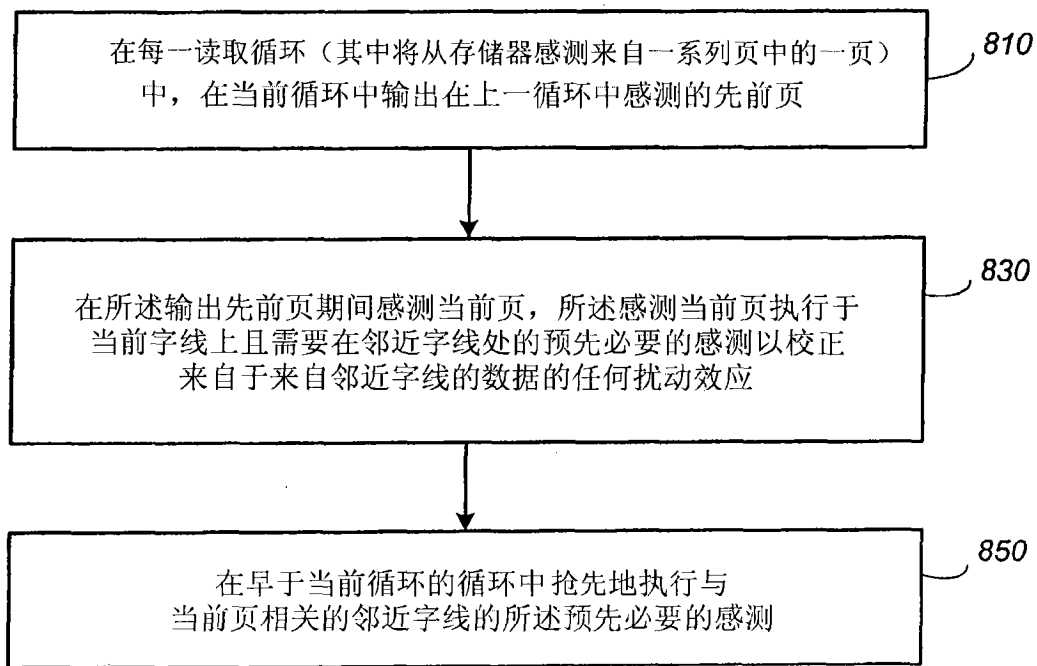
以LLM编码&LA校正进行的读取高速缓存
(根据常规方案)

图41



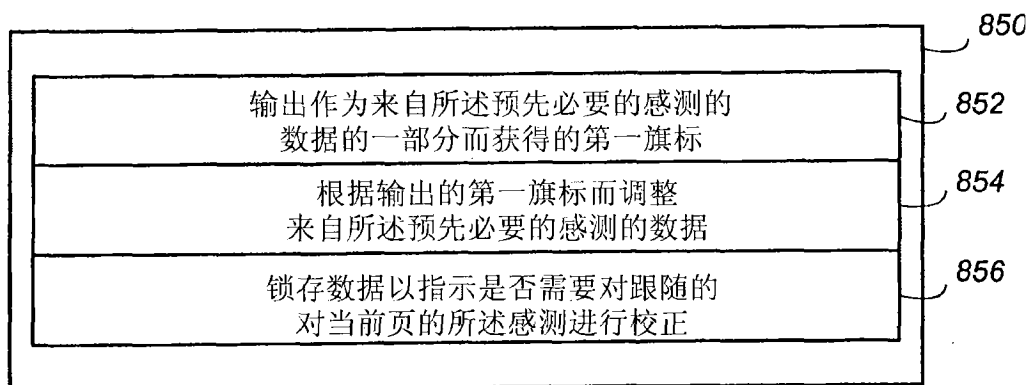
以LM编码&LA校正进行的改进读取高速缓存

图42



具有先行校正的改进读取高速缓存

图 43



预先必要的先行感测的操作

图 44

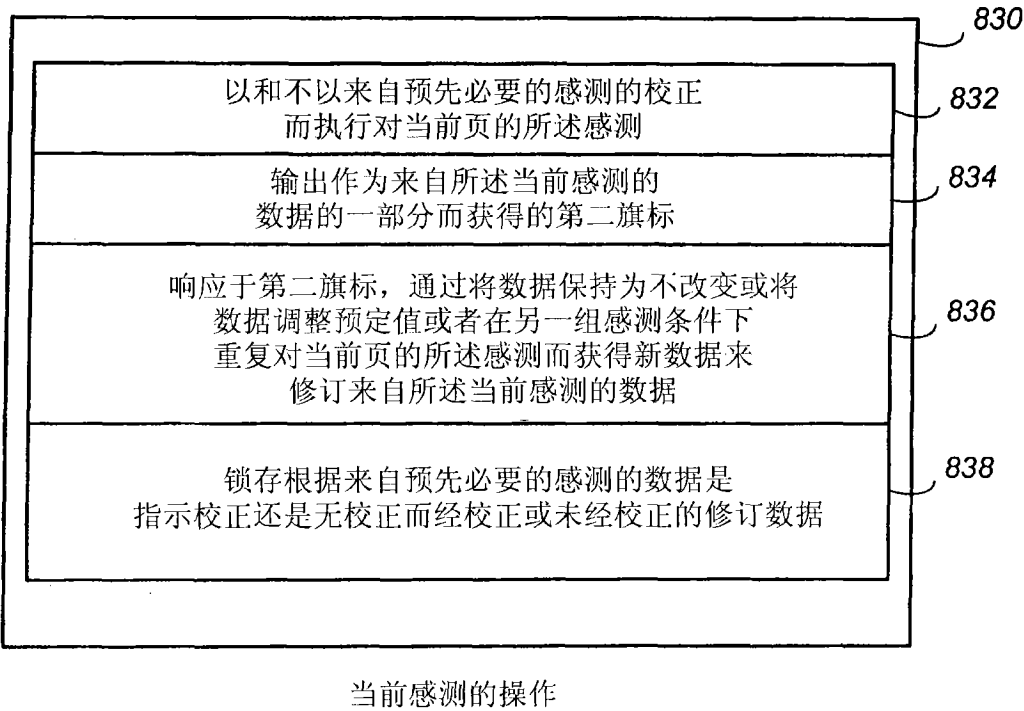


图 45