(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102318007 B
(45)授权公告日 2015.02.18

(21)申请号 200880115127.0
(22)申请日 2008.08.20
(30)优先权数据
11/852,229 2007.09.07 US
(85)PCT国际申请进入国家阶段日
2010.05.07
(86)PCT国际申请的申请数据
PCT/US2008/073750 2008.08.20
(87)PCT国际申请的公布数据
WO2009/035834 EN 2009.03.19
(73)专利权人 桑迪士克科技股份有限公司
地址 美国得克萨斯州
(72)发明人 李艳 方家荣 尼马·莫克莱西
(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 黄小临
(51)Int.Cl.
G11C 7/10(2006.01)
G11C 11/56(2006.01)
G11C 16/10(2006.01)
G11C 16/34(2006.01)
(56)对比文件
US 2004160829 A1,2004.08.19,
US 2004160829 A1,2004.08.19,
US 2006126390 A1,2006.06.15,
CN 101002279 A,2007.07.18,
审查员 刘细金

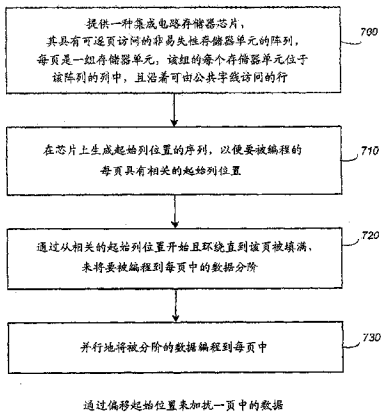
权利要求书2页 说明书27页 附图27页

(54)发明名称

用于对一页内和多页间的数据进行芯片上伪随机化的非易失性存储器和方法

(57)摘要

集成电路存储器芯片内的特征使能加扰或随机化被存储在非易失性存储器单元的阵列中的数据。在一个实施例中，每页内的随机化有助于控制在感测期间的源极载入错误和在相邻单元之间的浮置栅极与浮置栅极的耦合。逐页的随机化有助于减少由于具体数据样式的重复且长期的存储而导致的编程干扰、用户读干扰和浮置栅极与浮置栅极的耦合。在另一实施例中，在一页内和在各页之间实现随机化。在不同的实施例中，加扰或随机化可以是预定的、或是代码生成的伪随机或用户驱动的随机化。在集成电路存储器芯片的有限的资源和负担内实现这些特征。



CN 102318007 B

1. 一种集成电路存储器芯片，包括：

可逐页访问的非易失性存储器单元的阵列，每页是一群存储器单元，该群的每个存储器单元位于该阵列的列中，且沿着可由公共字线访问的行；

独立起始列位置的序列，使得要被编程的每页具有相关的起始列位置；

地址生成器，用于生成对于该相关的起始列位置的地址；

响应于地址生成器的、与每列相关的一组数据锁存器，用于将要被编程的数据安排到每页中，所安排的数据从相关的起始列位置开始且绕回直到该页被填满；以及

编程电路，用于并行地将所安排的数据编程到每页中。

2. 如权利要求 1 所述的存储器芯片，其中，各页被连续编号，且与一页相关的起始列位置是页号的函数。

3. 如权利要求 2 所述的存储器芯片，其中，该函数使得该列位置是对所述群中的单元的数量求模加上预定数。

4. 如权利要求 3 所述的存储器芯片，其中，所述预定数是零。

5. 如权利要求 1 所述的存储器芯片，还包括：

伪随机生成器；以及

由该伪随机生成器在芯片上生成起始列位置的序列。

6. 如权利要求 5 所述的存储器芯片，其中：

所述伪随机生成器还响应于由所述存储器芯片外部的事件触发的定时；以及

所述起始列位置的序列还是所述定时的函数。

7. 如权利要求 1 所述的存储器芯片，还包括：

用于要被编程的每页数据的第一和第二编码；

极性位的序列，其中每页一个极性位；以及

编码器，根据该页的极性位是处于第一还是第二状态，用第一或第二编码来编码该页数据。

8. 一种集成电路存储器芯片，包括：

可逐页访问的非易失性存储器单元的阵列，每页是一群存储器单元，该群的每个存储器单元位于该阵列的列中，且沿着可由公共字线访问的行；

用于其中要编程数据的每组的列的第一和第二编码；

伪随机极性位的序列，其中一个极性位用于一页的一组的列；

编码器，根据每组的列的伪随机极性位是处于第一还是第二状态，用第一或第二编码来编码与该组的列相关的数据位；以及

编程电路，并行地将被编码的数据编程到每页中。

9. 如权利要求 8 所述的存储器芯片，其中，所述第一编码是使得与每列相关的数据位不改变，且所述第二编码是翻转所述数据位。

10. 如权利要求 8 所述的存储器芯片，还包括：

伪随机生成器；以及其中

由所述伪随机生成器来生成伪随机极性位的序列。

11. 如权利要求 1 所述的存储器芯片，其中，其中所述阵列的存储器单元被组织成 NAND 结构。

12. 在具有可逐页访问的非易失性存储器单元的阵列的集成电路存储器芯片中，其中每页是一群存储器单元，该群的每个存储器单元位于该阵列的列中，且沿着可由公共字线访问的行，一种用于将数据编程到所述阵列中的方法，包括：

在芯片上生成独立起始列位置的序列，使得要被编程的每页具有相关的起始列位置；

通过从相关的起始列位置开始且绕回直到该页被填满，来将要被编程的数据安排到每页中；以及

并行地将所安排的数据编程到每页中。

13. 如权利要求 12 所述的方法，其中，各页被连续编号，且与一页相关的起始列位置是页号的函数。

14. 如权利要求 13 所述的方法，其中，该函数使得该列位置是对所述群中的单元的数量的求模加上预定数。

15. 如权利要求 14 所述的方法，其中，所述预定数是零。

16. 如权利要求 12 所述的方法，其中：

由伪随机生成器在芯片上生成起始列位置的序列。

17. 如权利要求 16 所述的方法，其中：

所述伪随机生成器还响应于由所述存储器芯片外部的事件触发的定时；以及

所述起始列位置的序列还是所述定时的函数。

18. 如权利要求 17 所述的方法，其中所述外部事件由存储器芯片的用户发起。

19. 如权利要求 17 所述的方法，其中，所述外部事件由外部存储器控制器发起。

20. 如权利要求 12 所述的方法，还包括：

提供用于要被编程的每页数据的第一和第二编码；

在芯片上生成独立极性位的序列，其中每页一个极性位；以及

根据该页的极性位是处于第一还是第二状态，用第一或第二编码来编码该页数据。

21. 在具有可逐页访问的非易失性存储器单元的阵列的集成电路存储器芯片中，每页是一群存储器单元，该群的每个存储器单元位于该阵列的列中，且沿着可由公共字线访问的行，一种用于将数据编程到该阵列中的方法，包括：

提供用于其中要编程数据的每组的列的第一和第二编码；

在芯片上生成伪随机极性位的序列，其中一页的每组的列一个极性位；

根据每组的列的伪随机极性位是处于第一还是第二状态，用第一或第二编码来编码与每组的列相关的数据位；以及

并行地将被编码的数据编程到每页中。

22. 如权利要求 21 所述的方法，其中，所述第一编码是使得与每组的列相关的数据位不改变，且所述第二编码是翻转所述数据位。

23. 如权利要求 21 所述的方法，其中：由伪随机生成器在芯片上生成所述伪随机极性位的序列。

用于对一页内和多页间的数据进行芯片上伪随机化的非易失性存储器和方法

技术领域

[0001] 本发明通常涉及诸如具有电荷存储元件的闪存的非易失性存储器，且更具体地涉及使得该存储器伪随机地存储数据以避免可能导致该存储器故障的可能的不期望的数据样式 (pattern)。

背景技术

[0002] 随着闪存卡和闪存盘的容量增加，在存储器阵列内的存储器单元的尺寸继续降低。在高密度阵列内，特别是 NAND 类型的阵列内，在该阵列的一个单元或部分中存储的电荷可能影响相邻单元的读取或编程操作。这是所谓的读取或编程干扰和单元耦合。

[0003] 通常为了得到关于 NAND 闪存的单元耦合、干扰和操作以及结构的更多信息，请参考美国专利申请公开 No.US-2006-0233026-A1，题为“Method forNon-Volatile Memory With Background Data Latch Caching During ProgramOperations（针对在编程操作期间具有背景数据锁存器高速缓存的非易失性存储器的方法）”；US-2006-0233023-A1，题为“Method for Non-Volatile MemoryWith Background Data Latch Caching During Erase Operations（针对在擦除操作期间具有背景数据锁存器高速缓存的非易失性存储器的方法）”；US-2006-0221696-A1，题为“Method for Non-Volatile Memory With BackgroundData Latch Caching During Read Operations（针对在读取操作期间具有背景数据锁存器高速缓存的非易失性存储器的方法）”；US 专利 No.6870768，题为“Techniques for Reducing Effects of Coupling Between Storage Elements ofAdjacent Rows of Memory Cells（用于降低存储器单元的相邻行的存储元件之间的耦合效应的技术）”；以及 US-2006-0140011-A1，题为“Reducing FloatingGate to Floating Gate Coupling Effect（降低浮置栅极与浮置栅极耦合效应）”，为了所有目的将其全部引用附于此。

[0004] 一些用户经常使用闪存来在闪存的某些块中反复一次又一次地不断存储相同的数据样式。结果是将留下一些将被擦除但从不被编程的位。另外，还将存在一些总是被编程而很少被擦除的位。这些持久的数据样式是有问题的：它们可能导致干扰和诸如浮置栅极到浮置栅极效应、NAND 串阻抗应和降低的存储器持久性和可靠性等的其他难题。

[0005] 能够进行电荷的非易失性存储的固态存储器、特别是被包装为小型卡的 EEPROM 和快闪 EEPROM 形式的固态存储器近来已经成为在各种移动和手持设备、尤其是信息电器和消费电子产品中的存储器的选择。不像也是固态存储器的 RAM（随机存取存储器）那样，闪存是非易失性的，即使在断电以后仍保持其存储的数据。尽管成本较高，但是，闪存越来越多地被使用在海量存储应用中。基于旋转诸如硬盘驱动器和软盘的磁介质的传统海量存储器不适合用于移动和手持环境。这是因为盘驱动器趋于体积大，易于发生机械故障，且具有高等待时间和高功率要求。这些不期望的属性使得基于盘的存储器在大多数移动和便携式应用中不实用。另一方面，嵌入式和可移除卡形式的闪存由于其小尺寸、低功耗、高速和高可靠性特征而理想地适用于移动和手持环境中。

[0006] EEPROM 和电可编程只读存储器 (EPROM) 是可以被擦除且使得新的数据被写入或“被编程”到它们的存储器单元中的非易失性存储器。两者都利用在源极和漏极区之间、在半导体衬底中的沟道区域上方的、场效应晶体管结构的浮置 (未连接) 导电栅极。然后，在浮置栅极上方提供控制栅极。由在浮置栅极上维持的电荷量来控制该晶体管的阈值电压特性。也就是说，对于在浮置栅极上的给定水平的电荷，存在必须在该晶体管被“导通”以允许在其源极和漏极区之间导电之前施加到控制栅极的对应的电压 (阈值)。

[0007] 浮置栅极可以保持一个范围的电荷，且因此可以被编程到阈值电压窗内的任何阈值电压电平。阈值电压窗的大小由该器件的最小和最大阈值电平来界定，该器件的最小和最大阈值电平又对应于可以被编程到浮置栅极上的电荷的范围。阈值窗通常取决于存储器器件的特性、工作条件和历史。在该窗内的每个不同的、可分辨的阈值电压电平范围原则上可以用于指定该单元的明确的 (definite) 存储器状态。

[0008] 通常通过两个机制之一将用作存储器单元的晶体管编程为“编程”状态。在“热电子注入”中，施加到漏极的高电压将穿过衬底沟道区域的电子加速。同时，施加到控制栅极的高电压将热电子通过薄栅极介电质 (thin gatedielectric) 拉到浮置栅极上。在“隧道注入 (tunneling injection)”中，相对于衬底，高电压被施加到控制栅极。以此方式，电子从衬底被拉到居间的浮置栅极。

[0009] 可以通过一些机制来擦除存储器器件。对于 EPROM，存储器可通过用紫外线辐射将电荷从浮置栅极移除而大批 (bulk) 擦除。对于 EEPROM，存储器单元可通过相对于控制栅极将高电压施加到衬底以便诱使浮置栅极中的电子以隧道效应穿过薄氧化物到衬底沟道区域 (即，Fowler-Nordheim 隧道技术) 而电擦除。典型地，EEPROM 可逐字节擦除。对于快闪 EEPROM，该存储器可一次全部或每次一个或多个块地电擦除，其中，一个块可以由存储器的 512 字节或更多字节组成。

[0010] 非易失性存储器单元的例子

[0011] 存储器器件通常包括可以被安装在卡上的一个或多个存储器芯片。每个存储器芯片包括由诸如解码器和擦除、写和读电路的外围电路支持的存储器单元的阵列。更复杂的存储器器件还带有进行智能和更高级存储器操作和接口的控制器。存在许多现今正使用的商业上成功的非易失性固态存储器器件。这些存储器器件可以使用不同类型的存储器单元，每个类型具有一个或多个电荷存储元件。

[0012] 图 1A-1E 示意性地图示非易失性存储器单元的不同例子。

[0013] 图 1A 示意性地图示具有用于存储电荷的浮置栅极的 EEPROM 单元形式的非易失性存储器。电可擦除和可编程只读存储器 (EEPROM) 具有类似于 EPROM 的结构，但另外提供用于在施加适当的电压后将电荷从其浮置栅极电载入和移除而不需要曝露在 UV 辐射下的机制。在美国专利 No.5595924 中给出了这种单元和其制造方法的例子。

[0014] 图 1B 示意性地图示了具有选择栅极和控制或操纵栅极 (steering gate) 的快闪 EEPROM。存储器单元 10 具有在源极 14 和漏极 16 扩散之间的“分裂沟道”12。有效地用串联的两个晶体管 T1 和 T2 来形成单元。T1 用作具有浮置栅极 20 和控制栅极 30 的存储器晶体管。浮置栅极能够存储可选择量的电荷。可以流过沟道的 T1 的部分的电流量取决于在控制栅极 30 上的电压和在居间 (intervening) 浮置栅极 20 上驻留的电荷量。T2 用作具有选择栅极 40 的选择晶体管。当 T2 由选择栅极 40 处的电压导通时，其允许在该沟道的

T1 的部分中的电流在源极和漏极之间流过。选择晶体管提供独立于控制栅极处的电压的沿着源极 - 漏极沟道的开关。一个优点是它可以用于截止由于在其浮置栅极处的其电荷耗尽（正）而在零控制栅极电压处仍然导电的那些单元。另一优点是，其允许更容易地实现源极侧注入编程。

[0015] 分裂沟道存储器单元的一个简单的实施例是其中选择栅极和控制栅极连接到如由图 1B 中示出的虚线示意地指示的同一字线。这是通过使得电荷存储元件（浮置栅极）位于该沟道的一部分上且使得控制栅极结构（其是字线的一部分）位于另一沟道部分上以及该电荷存储元件上来实现的。这有效形成了具有串联的两个晶体管的单元，一个（存储器晶体管）具有在电荷存储元件上的电荷量和在控制可以流过该沟道的该晶体管的部分的电流量的字线上的电压的组合，且另一个（选择晶体管）使得该字线单独用作其栅极。在美国专利 No. 5070032、5095344、5315541、5343063 和 5661053 中给出了这种单元的例子、其在存储器系统中的使用以及制造它们的方法。

[0016] 图 1B 中示出的分裂沟道单元的更细化的实施例是当选择栅极和控制栅极独立且不通过它们之间的虚线连接时。一个实施方式使得在单元阵列中的一列的各控制栅极连接到与字线垂直的控制（或操纵）线。效果是使得字线免于当读取或编程所选单元时必须同时进行两个功能。那两个功能是 (1) 用作选择晶体管的栅极，由此需要适当的电压以将选择晶体管导通和截止，(2) 通过在字线和电荷存储元件之间的电场（容性）耦合将该电荷存储元件的电压驱动到期望的电平。通常难以用单个电压以最佳方式进行这两个功能。通过控制栅极和选择栅极的分离控制，字线仅需要进行功能 (1)，而添加的控制线进行功能 (2)。该能力允许设计在调整编程电压到目标数据的情况下的更高性能编程。在例如美国专利 No. 5313421 和 6222762 中描述了在快闪 EEPROM 阵列中的独立的控制（或操纵）栅极的使用。

[0017] 图 1C 示意性地图示了具有双浮置栅极和独立的选择和控制栅极的另一快闪 EEPROM 单元。该存储器单元 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。在这类单元中，两个存储元件（即 T1- 左和 T1 右的存储元件）被包括在源极和漏极扩散区之间的其沟道之上，且选择晶体管 T1 位于其间。存储器晶体管分别具有浮置栅极 20 和 20’以及控制栅极 30 和 30’。选择晶体管 T2 被选择栅极 40 控制。在任何一个时间，仅存储器晶体管对中的一个被存取用于读或写。当存储单元 T1- 左被存取时，T2 和 T1- 右两者都被导通以允许在沟道的 T1- 左的部分中的电流在源极和漏极之间流过。类似地，当存储单元 T1- 右正被存取时，T2 和 T1- 左被导通。通过使得选择栅极多晶硅的一部分很靠近浮置栅极并向选择栅极施加相当大的正电压（例如 20V）使得存储在浮置栅极内的电子可以隧道效应到达选择栅极多晶硅，来进行擦除。

[0018] 图 1D 示意性地图示了被组织为 NAND 单元的存储器单元的串。NAND 单元 50 由被其源极和漏极菊链连接 (daisy-chain) 的一系列存储器晶体管 M1、M2.....Mn(n = 4、8、16 或更高）构成。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 54 和漏极端 56 与外部的连接。在存储器阵列中，当源极选择晶体管 S1 导通时，源极端被耦接到源极线。类似地，当漏极选择晶体管 S2 导通时，NAND 单元的漏极端被耦接到存储器阵列的位线。在该链中的每个存储器晶体管具有用于存储给定量的电荷以便呈现想要的存储器状态的电荷存储元件。每个存储器晶体管的控制栅极提供对读和写操作的控制。每个选择

晶体管 S1、S2 的控制栅极提供分别经由其源极端 54 和漏极端 56 对 NAND 单元的控制存取。

[0019] 当在编程期间读取和验证 NAND 单元内的被寻址的存储器晶体管时，该晶体管的控制栅极被供应了适当的电压。同时，在 NAND 单元 50 中的剩余的未被寻址的存储器晶体管通过施加在其控制栅极上的足够的电压而完全导通。以此方式，有效地创建了从单独的存储器晶体管的源极到 NAND 单元的源极端 54 的导电路径，并且对于单独的存储器晶体管的漏极到该单元的漏极端 56 也是同样。在美国专利 No. 5570315、5903495 和 6046935 中描述了具有这种 NAND 单元结构的存储器器件。

[0020] 图 1E 示意性地图示了具有用于存储电荷的介电层的非易失性存储器。取代先前描述的导电的浮置栅极元件，使用介电层。已经由 Eitan 等人的“NROM :A Novel Localized Trapping, 2-Bit Nonvolatile Memory Cell (NROM :新颖的局部俘获、2 位非易失性存储器单元)”，IEEE Electron Device Letters, Vol. 21, no. 11, 2000 年 11 月，543-545 页中描述了使用介电存储元件的这种存储器器件。ONO 介电层延伸穿过在源极和漏极扩散区之间的沟道。用于一个数据位的电荷被局限 (localize) 在与漏极相邻的介电层中，而用于另一数据位的电荷被局限在与源极相邻的介电层中。例如美国专利 No. 5768192 和 6011725 公开了具有夹在两个硅氧化物层之间的俘获电介质 (trappingdielectric) 的非易失性存储器单元。通过分离地读取在该电介质内的空间分离的电荷存储区域的二进制状态来实现多状态数据存储。

[0021] 存储器阵列

[0022] 存储器器件典型地由按行和列排列的并可由字线和位线寻址的存储器单元的二维阵列组成。可以根据 NOR 型或 NAND 型架构来形成该阵列。

[0023] NOR 阵列

[0024] 图 2 图示了存储器单元的 NOR 阵列的例子。已经利用图 1B 或 1C 示出的类型的单元实现了具有 NOR 型架构的存储器器件。每行存储器单元以菊链方式通过其源极和漏极连接。该设计有时称为虚拟接地设计。每个存储器单元 10 具有源极 14、漏极 16、控制栅极 30 和选择栅极 40。一行中的各单元使得其选择栅极连接到字线 42。一列中的各单元使得其源极和漏极分别连接到所选位线 34 和 36。在存储器单元使得其控制栅极和选择栅极独立地被控制的某些实施例中，操纵线 36 还连接一列中的各单元的控制栅极。

[0025] 利用其中每个存储器单元用连接到一起的其控制栅极和选择栅极形成的存储器单元来实现许多快闪 EEPROM 器件。在这种情况下，不需要操纵线，且字线仅连接沿着每行的单元的所有控制栅极和选择栅极。在美国专利 No. 5172338 和 5418752 中公开了这些设计的例子。在这些设计中，字线本质上进行两个功能 :行选择 ;和向在该行中的所有单元供应控制栅极电压用于读取或编程。

[0026] NAND 阵列

[0027] 图 3 图示了诸如图 1D 所示的存储器单元的 NAND 阵列的例子。沿着 NAND 单元的每列，位线被耦合到每个 NAND 单元的漏极端 56。沿着 NAND 单元的每行，源极线可以连接其所有源极端 54。而且，沿着一行的 NAND 单元的控制栅极被连接到一系列对应的字线。可以通过用经由所连接字线的在该对选择晶体管 (见图 1D) 的控制栅极上的适当的电压导通该对选择晶体管来对整行 NAND 单元寻址。当在 NAND 单元的链内的存储器晶体管正被读取时，在该链中剩余的存储器晶体管经由其相关字线而硬导通 (turned onhard)，使得流过该链

的电流主要取决于正被读取的单元中存储的电荷的水平。在美国专利No.5570315、5774397和6046935中找到作为存储器系统的部分的NAND架构阵列和其操作的例子。

[0028] 块擦除

[0029] 电荷存储存储器器件的编程只会导致向其电荷存储元件添加更多的电荷。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（未示出）来擦除存储器单元的一个或多个块。当整个阵列的单元或该阵列的很多组单元一起（即一瞬间）被电擦除时，诸如EEPROM的非易失性存储器被称为“快闪”EEPROM。一旦被擦除，然后就可以重新编程单元组。可以一起擦除的单元组可以组成一个或多个可寻址的擦除单位。擦除单位或块典型地存储一页或多页数据，页是编程和读取的单位，虽然也可以在单个操作中编程或读取多于一页。每页典型地存储一个或多个扇区的数据，扇区的大小由主机系统限定。一个例子是遵循关于磁盘驱动器建立的标准的512字节用户数据加上关于用户数据和/或存储该用户数据的块的管理信息的一些字节的扇区。

[0030] 读/写电路

[0031] 在通常的两状态EEPROM单元中，建立至少一个电流断点水平，以便将导电窗划分为两个区域。当通过施加预定、固定的电压来读取单元时，通过与断点水平（或参考电流IREF）比较来将其源极/漏极电流解释（resolve）为存储器状态。如果读取的电流高于该断点水平的电流，则确定该单元处于一个逻辑状态（例如，“零”状态）。另一方面，如果该电流小于该断点水平的电流，则确定该单元处于另一逻辑状态（例如，“一”状态）。因此，这种两状态单元存储一位数字信息。通常提供可以被外部编程的参考电流源作为存储器系统的一部分来生成断点水平电流。

[0032] 为了增加存储器容量，随着半导体技术状态的进步，正在制造具有越来越高的密度的快闪EEPROM器件。增加存储容量的另一方法是使得每个存储器单元存储多于两个状态。

[0033] 对于多状态或多级EEPROM存储器单元，导电窗被多于一个断点划分为多于两个区域，以便每个单元能够存储多于一位的数据。因此，给定的EEPROM阵列可以存储的信息随每个单元可以存储的状态的数量而增加。已经在美国专利No.5172338中描述了具有多状态或多级存储器单元的EEPROM或快闪EEPROM。

[0034] 实践中，通常通过当向控制栅极施加参考电压时感测穿过该单元的源极和漏极电极的导电电流来读取单元的存储器状态。因此，对于在单元的浮置栅极上的每个给定电荷，可以检测相对于固定参考控制栅极电压的对应导电电流。类似地，可编程到浮置栅极上的电荷的范围限定了对应的阈值电压窗或对应的导电电流窗。

[0035] 或者，取代检测在划分的电流窗之间的导电电流，可以设置在控制栅极处被测试的给定的存储器状态的阈值电压，且检测导电电流低于还是高于阈值电流。在一个实施方式中，通过检查导电电流通过位线的电容放电的速率来实现相对于阈值电流的导电电流的检测。

[0036] 图4图示了对于浮置栅极可以在任一时间选择性地存储的四个不同的电荷Q1-Q4的、在源极漏极电流I_D和控制栅极电压V_{CG}之间的关系。四个实线I_D对V_{CG}曲线表示分别对应于四个可能的存储器状态的、可以编程在存储器单元的浮置栅极上的四个可能的电荷水平。作为例子，全体单元的阈值电压窗的范围可以从0.5V到3.5V。可以通过将阈值窗以每

个 0.5V 的间隔划分为五个区域来界定六个存储器状态。例如，如果如所示地使用 2μA 的参考电流 I_{REF}，则用 Q1 编程的单元可以被认为为处于存储器状态“1”，因为其曲线在由 V_{CG} = 0.5V 和 1.0V 所界定的阈值窗的区域中与 I_{REF} 交叉。类似地，Q4 处于存储器状态“5”。

[0037] 如可以从上述描述看出的，使得存储器单元存储的状态越多，其阈值窗被划分得越精细。这将要求在编程和读取操作中更高的精确度，以便能够实现要求的分辨率。

[0038] 美国专利 No. 4357685 公开了编程 2 状态 EPROM 的方法，其中，当单元被编程到给定状态时，其经历连续的编程电压脉冲，每次向浮置栅极增加递增的电荷。在脉冲之间，该单元被回读或验证以确定其相对于断点水平的源极 - 漏极电流。当已经验证电流状态达到期望的状态时，编程停止。使用的编程脉冲列可以具有增加的周期或幅度。

[0039] 现有技术的编程电路仅施加编程脉冲以从擦除或接地状态步经阈值窗，直到达到目标状态。实际上，为了允许足够的分辨率，每个划分或界定的区域将需要至少五个编程步来横跨 (transverse)。该性能对于 2 状态存储器单元是可接受的。但是，对于多状态单元，所需要的步的数量随着划分的数量而增加，因此，必须增加编程精确度或分辨率。例如，16 状态单元可能需要平均至少 40 个编程脉冲来编程到目标状态。

[0040] 图 5 示意性地图示了具有可由读 / 写电路 170 经由行解码器 130 和列解码器 160 存取的存储器阵列 100 的典型布置的存储器器件。如结合图 2 和图 3 所述，在存储器阵列 100 中的存储器单元的存储器晶体管可经由（一个或多个）所选字线和（一个或多个）位线的一组来寻址。行解码器 130 选择一个或多个字线，且列解码器 160 选择一个或多个位线，以便向被寻址的存储器晶体管的各个栅极施加适当的电压。提供读 / 写电路 170 来读或写（编程）被寻址的存储器晶体管的存储器状态。读 / 写电路 170 包括经由位线可连接到该阵列中的存储器元件的一些读 / 写模块。

[0041] 图 6A 是单独的读 / 写模块 190 的示意方块图。实质上，在读取或验证期间，感测放大器确定流过经由所选位线连接的被寻址的存储器晶体管的漏极的电流。该电流取决于在存储器晶体管中存储的电荷和其控制栅极电压。例如，在多状态 EEPROM 单元中，其浮置栅极可以被充电到若干不同的电平之一。对于 4 级单元，其可以用于存储两位数据。由级到位 (level-to-bit) 转换逻辑将感测放大器检测的级转换为要被存储在数据锁存器中的一组数据位。

[0042] 影响读 / 写性能和准确度的因素

[0043] 为了改进读和编程性能，并行读取或编程在阵列中的多个电荷存储元件或存储器晶体管。因此，一起读取或编程存储器元件的逻辑“页”。在已有的存储器架构中，一行典型地包含若干交错 (interleaved) 的页。将一起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的页中的每页连接到对应数量的读 / 写模块。例如，在一个实施方式中，指定存储器阵列具有 532 字节的页大小 (512 字节加上管理开销的 20 字节）。如果每列包含漏极位线且每行存在两个交错的页，则其总共 8512 列，且每页与 4256 列相关。将存在可连接以并行读或写所有偶数位线或奇数位线的 4256 个感测模块。以此方式，并行地从该页存储器元件读取一页的 4256 位（即 532 字节）数据，或将其编程到该页存储器元件中。形成读 / 写电路 170 的读 / 写模块可以被布置为各种架构。

[0044] 参考图 5，读 / 写电路 170 被组织为读 / 写堆栈 (stack)180 的堆 (bank)。每个读 / 写堆栈 180 是读 / 写模块 190 的堆栈。在存储器阵列中，列间隔由占据它的一个或两个晶

体管的大小来确定。但是，如从图 6A 可以看出，读 / 写模块的电路将很可能用更多个晶体管和电路元件来实现，因此将占据超过许多列的空间。为了对在所占据的列之中的多于一列提供服务，在彼此的顶部堆叠多个模块。

[0045] 图 6B 示出了传统上由读 / 写模块 190 的堆栈实现的图 5 的读 / 写堆栈。例如，读 / 写模块可以延伸过 16 列，然后可以使用具有八个读 / 写模块的堆栈的读 / 写堆栈 180 来并行地对八个列服务。读 / 写堆栈可以经由列解码器被耦合到在该堆中的八个奇数 (1、3、5、7、9、11、13、15) 列或八个偶数 (2、4、6、8、10、12、14、16) 列。

[0046] 如前所述，传统的存储器器件通过一次以大量并行的方式在所有偶数或所有奇数位线上操作来改进读 / 写操作。由两个交错的页组成的行的该架构将有助于缓解安置读 / 写电路的块的问题。这还通过考虑控制位线与位线电容耦合来支配 (dictate)。使用块解码器来复用该组读 / 写模块到偶数页或奇数页。以此方式，无论何时一组位线正被读取或编程时，交错的组可以被接地以最小化紧密相邻耦合。

[0047] 但是，交错页架构在至少三个方面中是有缺点的。首先，其要求额外的复用电路。第二，其性能慢。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，在解决诸如当在不同时间、诸如在奇数和偶数页中分离地编程在浮置栅极电平处的两个相邻电荷存储元件时该相邻电荷存储元件之间的场耦合之类的其他干扰效应方面也不是最佳的。

[0048] 相邻场耦合的问题随着存储器晶体管之间的间隔更近而变得越明显。在存储器晶体管中，电荷存储元件被夹在沟道区域和控制栅极之间。在沟道区域中流动的电流是由在控制栅极和电荷存储元件处的场贡献的结果电场的函数。随着不断增加的密度，存储器晶体管在一起被形成得越来越近。来自相邻的电荷元件的场则变为对受影响的单元的结果场的重大贡献者。相邻的场取决于被编程到相邻者的电荷存储元件中的电荷。该扰动场由于随着相邻者的编程状态而改变，因此本质上是动态的。因此，受影响的单元可以取决于相邻者的改变的状态，在不同的时间不同地读取。

[0049] 交错页的传统架构恶化了由相邻的浮置栅极耦合造成的误差。由于彼此独立地编程和读取偶数页和奇数页，因此，取决于同时对居间的页发生了什么，可以在一组条件下编程一页，而在完全不同的一组条件下回读该页。读取的误差随着增加的密度将变得更严重，对于多状态实施方式，需要更准确的读操作和对阈值窗的更粗略的划分。将损害性能，且在多状态实施方式中的可能的容量被限制。

[0050] 美国专利公开 No.US-2004-0060031-A1 公开了具有用于并行地读和写对应块的存储器单元的一大块读 / 写电路的高性能且致密的非易失性存储器器件。具体地，该存储器器件具有将读 / 写电路块中的冗余减少到最小的架构。通过将读 / 写模块的块重新分配为并行地操作的块读 / 写模块核心部分，同时以时间复用的方式与实质更小的公共部分的组交互，来实现空间以及功率的明显节省。具体地，由共享的处理器来进行在多个感测放大器和数据锁存器之间的读 / 写电路之中的数据处理。

[0051] 因此，通常需要高性能和高容量的非易失性存储器。具体地，需要具有致密且有效的、又对于在读 / 写电路之中处理数据极通用的改进处理器的、具有增强的读和编程性能的致密非易失性存储器。

发明内容

[0052] 根据本发明的一个方面，在每个存储器页中的数据被随机化，以便当来自若干页的数据被排队时，在编程期间避免有问题的数据样式。

[0053] 在一个优选实施例中，加扰在一页上的数据的简单方式是将数据写在对于每个不同的页的独立或不同的起始地址上。对应页的数据被写到在对于每页不同的起始位置处的每页存储器单元。当数据被填充到该页的尾部时，其通过从该页的第一地址回绕来继续，直到就在起始位置之前。

[0054] 在另一优选实施例中，由伪随机生成器提供对于每页一个的起始物理列地址的序列。

[0055] 根据本发明的另一方面，页中的数据位被随机化，使得平均而言该页包含相等的具有擦除和编程状态的单元混合。以此方式，源极线偏压或载入实质上不改变，且可以允许在感测操作期间的适当调整。

[0056] 这通过随机化一页内的各个位来实现。优选地，使用每位指定某个极性的伪随机位的序列来编码该页内的位。在一个实施例中，对于页中的每个数据位存在一个极性位。在另一实施例中，对于页中的每字节数据存在一个极性位。在该实施例中，如果该极性位指定了位的翻转，则在该数据字节内的所有位将被翻转。

[0057] 根据本发明的另一方面，一页内的随机化与逐页的随机化结合。具体地，给定了芯片上电路的有限资源，优选地通过具有独立的起始位置的每页来实现一页内的随机化，且优选地通过具有独立的编码极性的每页来实现逐页随机化。

[0058] 在另一实施例中，通过具有独立起始位置的每页实现的一页内的随机化还通过具有独立编码极性的一页内的每组数据位而加强。

[0059] 各种随机化方法和实施例是存储器芯片 (EEPROM) 实现的。也就是说，它们发生在存储器芯片本身内，而不是利用与该芯片通信的存储器控制器。这不同于解决已知为损耗均衡 (wear leveling) 的问题的不同技术，该技术通常在系统级上实现，且使用控制器来改变如何将数据存储在存储器芯片内。

[0060] 本发明将减少或消除可能导致编程干扰或用户读干扰的具体数据样式，以及减少 NAND 串电阻效应，并增加存储器容忍度和可靠性。还将减少浮置栅极与浮置栅极耦合的问题。

附图说明

[0061] 图 1A-1E 示意性地图示了非易失性存储器单元的不同例子。

[0062] 图 2 图示了存储器单元的 NOR 阵列的例子。

[0063] 图 3 图示了诸如图 1D 所示的存储器单元的 NAND 阵列的例子。

[0064] 图 4 图示了对于浮置栅极可以任一时间存储的四个不同的电荷 Q1-Q4 的在源极 - 漏极电流和控制栅极电压之间的关系。

[0065] 图 5 示意性地图示了可由读 / 写电路经由行和列解码器存取的存储器阵列的典型布置。

[0066] 图 6A 是单独的读 / 写模块的示意方块图。

[0067] 图 6B 示出了传统上由读 / 写模块的堆栈实现的图 5 的读 / 写堆栈。

[0068] 图 7A 示意性地图示了具有被划分的读 / 写堆栈的堆的致密存储器器件，其中实现了本发明的改进的处理器。

[0069] 图 7B 图示了图 7A 中所示的致密存储器器件的优选布置。

[0070] 图 8 示意性地图示了在图 7A 所示的读 / 写堆栈中的基本组件的一般布置。

[0071] 图 9 图示了在图 7A 和 7B 所示的读 / 写电路之中的读 / 写堆栈的一个优选布置。

[0072] 图 10 图示了图 9 所示的公共处理器的改进的实施例。

[0073] 图 11A 图示了图 10 所示的公共处理器的输入逻辑的优选实施例。

[0074] 图 11B 图示了图 11A 的输入逻辑的真值表。

[0075] 图 12A 图示了图 10 所示的公共处理器的输出逻辑的优选实施例。

[0076] 图 12B 图示了图 12A 的输出逻辑的真值表。

[0077] 图 13 是示出与在本发明的 2 位的实施例中的讨论有关的一些具体元件的图 10 的简化版本。

[0078] 图 14 指示在读入了较低页数据的情况下用于较高页编程的与图 13 相同的元件的锁存器分配。

[0079] 图 15 图示了在单个页模式中的高速缓存编程的方面。

[0080] 图 16 示出了可以在较低页到全序列转换 (full sequence conversion) 中使用的编程波形。

[0081] 图 17 图示了用全序转换的高速缓存编程操作中的相对时序。

[0082] 图 18 描述了在高速缓存页复制操作中的锁存器的部署。

[0083] 图 19A 和 19B 图示了在高速缓存页复制操作中的相对时序。

[0084] 图 20 图示了当每个存储器单元使用 LM 码存储两位数据时的 4 状态存储器阵列的阈值电压分布。

[0085] 图 21 是 EEPROM 或存储器芯片 600 的某些组件的示意方块图。

[0086] 图 22A 是根据页地址的编码方案和各页数据的极性位的图示。

[0087] 图 22B 是图示用于变换用户数据的编码的示例 17 位码的应用的表。

[0088] 图 22C 图示向存储在示例 NAND 链 / 串的存储器单元中的较高和较低位应用极性位。

[0089] 图 23A 是作为命令时钟信号的函数的编码方案确定的图示。

[0090] 图 23B 是命令的时钟信号。

[0091] 图 23C 图示用于数据编码确定和反转的控制电路的实施例。

[0092] 图 23D 图示了其中存储了极性位的用户数据的页。

[0093] 图 23E 图示了用于针对读操作来回复 (revert) 被电势反转的数据的编码的控制电路的实施例。

[0094] 图 24A 是命令时钟信号和示出了在命令时钟信号的上升沿处的 FSM 时钟的值的有限状态机时钟信号的时间线。

[0095] 图 24B 图示了确定极性位为图 24A 所示的命令时钟信号的函数的示例电路。

[0096] 图 25 更详细地图示了在图 7A 和图 9 中示出的芯片上控制电路。

[0097] 图 26 图示了根据从每页的不同的起始位置开始写的一个优选实施例的、加扰在存储器页上的数据的方法。

[0098] 图 27 是图示具有用于写数据的不同起始位置的不同页的例子的表。

[0099] 图 28A 图示了由于在具有对地的有限电阻的源极线中流动的电流而引起的源极电压误差的问题。

[0100] 图 28B 图示了由源极线电压降导致的存储器单元的阈值电压电平中的误差。

[0101] 图 29 图示了在一页内对位随机化的方法。

[0102] 图 30 图示了根据本发明的另一方面逐页并在每页内对数据随机化的方法。

具体实施方式

[0103] 图 7A 到图 20 图示了实现本发明的优选存储器系统。

[0104] 图 7A 示意性地图示了具有被划分的读 / 写堆栈的堆的致密存储器器件，其中实现了本发明的改进的处理器。该存储器器件包括存储器单元 300、控制电路 310 和读 / 写电路 370 的二维阵列。可由字线经由行解码器 330 和由位线经由列解码器 360 来对存储器阵列 300 寻址。读 / 写电路 370 被实现为被划分的读 / 写堆栈 400 的堆，且允许一块（也称为“页”）存储器单元并行被读取或编程。在优选实施例中，一页由邻近行的存储器单元构成。在另一实施例中，在一行存储器单元被划分为多个块或页的情况下，提供块复用器 350 来将读 / 写电路 370 复用到各个块。

[0105] 控制电路 310 与读 / 写电路 370 合作以对存储器阵列 300 进行存储器操作。控制电路 310 包括状态机 312、芯片上地址解码器 314 和功率控制模块 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 提供在由主机或存储器控制器所用与由解码器 330 和 370 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被供应给字线和位线的功率和电压。

[0106] 图 7B 图示了图 7A 所示的致密存储器器件的优选布置。以对称的方式在存储器阵列 300 的相对侧上实现由各种外围电路对存储器阵列 300 的存取，以便在每侧上的存取线和电路减少一半。因此，行解码器被分割成行解码器 330A 和 330B，且列解码器被分割为列解码器 360A 和 360B。在其中一行存储器单元被划分为多个块的实施例中，块复用器 350 被分割为块复用器 350A 和 350B。类似地，读 / 写电路被分割为连接到来自阵列 300 的底部的位线的读 / 写电路 370A 和连接到来自阵列 300 的顶部的位线的读 / 写电路 370B。以此方式，读 / 写模块的密度、以及因此的被划分的读 / 写堆栈 400 的密度实质上降低了一半。

[0107] 图 8 示意性地图示了图 7A 所示的读 / 写堆栈中的基本组件的大体布置。根据本发明的大体架构，读 / 写堆栈 400 包括用于感测 k 个位线的感测放大器 212 的堆栈、用于经由 I/O 总线 231 输入或输出数据的 I/O 模块 440、用于存储输入或输出数据的数据锁存器 430 的堆栈、用于处理和存储在读 / 写堆栈 400 之间的数据的公共处理器 500、和用于在堆栈组件之间通信的堆栈总线 421。在读 / 写电路 370 之间的堆栈总线控制器经由线路 411 提供控制和定时信号，用于控制在读 / 写堆栈之间的各种组件。

[0108] 图 9 图示了图 7A 和 7B 示出的读 / 写电路之间的读 / 写堆栈 (stack) 的一个优选布置。每个读 / 写堆栈 400 并行地在一组 k 个位线上操作。如果一页具有 p = r*k 个位线，则将有 r 个读 / 写堆栈，400-1. . . . . 400-r。

[0109] 并行操作的被划分的读 / 写堆栈 400 的整个堆 (bank) 允许并行地读取或编程沿着一行的 p 个单元的块（或页）。因此，将有 p 个读 / 写模块用于整行单元。由于每个堆栈

服务于 k 个存储器单元，因此在该堆中的读 / 写堆栈的总数由 r ＝ p/k 给出。例如，如果 r 是在该堆中的堆栈的数量，则 p ＝ r*k。一个示例存储器阵列可能具有 p ＝ 512 字节 (512x8 位)，k ＝ 8，因此 r ＝ 512。在优选实施例中，该块是整行单元的一连串 (run)。在另一实施例中，该块是一行中的单元的子集。例如，单元的子集可以是整行的一半或整行的四分之一。单元的子集可以是相邻单元的一连串或每隔一个单元一个，或每隔预定数量的单元一个。

[0110] 每个读 / 写堆栈、比如 400-1 实质上包含并行地服务于一段 k 个存储器单元的感测放大器 212-1 到 212-k 的堆栈。在美国专利公开 No. 2004-0109357-A1 中公开了优选的感测放大器，其全部公开被引用附于此。

[0111] 堆栈总线控制器 410 经由线路 411 向读 / 写电路 370 提供控制和定时信号。堆栈总线控制器本身经由线路 311 依赖于存储器控制器 310。在每个读 / 写堆栈 400 之间的通信受互连的堆栈总线 431 影响，且受堆栈总线控制器 410 控制。控制线 411 从堆栈总线控制器 410 向读 / 写堆栈 400-1 的组件提供控制和时钟信号。

[0112] 在优选布置中，堆栈总线被划分为用于在公共处理器 500 和感测放大器 212 的堆栈之间通信的 SA 总线 (SABus)422 和用于在处理器和数据锁存器 430 的堆栈之间通信的 D 总线 (DBus)423。

[0113] 数据锁存器 430 的堆栈包括数据锁存器 430-1 到 430-k，与该堆栈相关的每个存储器单元一个数据锁存器。I/O 模块 440 使得这些数据锁存器能够经由 I/O 总线 231 与外部交换数据。

[0114] 公共处理器还包括输出 507，用于输出指示存储器操作的状态、比如错误状况的状态信号。该状态信号用于驱动以线或 (Wired-Or) 配置联系于 FLAGBUS(标记总线)509 的 n- 晶体管 550 的栅极。该标记总线优选地被控制器 310 预充电，且当由任一读 / 写堆栈给状态信号赋值 (assert) 时将被下拉。

[0115] 图 10 图示了图 9 所示的公共处理器的改进的实施例。公共处理器 500 包括处理器总线、用于与外部电路通信的 PBUS 505、输入逻辑 510、处理器锁存器 PLatch 520 和输出逻辑 530。

[0116] 输入逻辑 510 接收来自 PBUS 的数据，并取决于经由信号线 411 来自堆栈总线控制器 410 的控制信号，向 BSI 节点输出作为处于逻辑状态“1”、“0”或“Z”(浮置) 之一的变换数据。然后，设置 / 复位锁存器、PLatch 520 锁存 BSI，得到了一对互补的输出信号，为 MTCH 和 MTCH*。

[0117] 输出逻辑 530 接收 MTCH 和 MTCH* 信号，并取决于经由信号线 411 来自堆栈总线控制器 410 的控制信号，在 PBUS 505 上输出处于逻辑状态“1”、“0”或“Z”(浮置) 之一的变换数据。

[0118] 在任一时间，公共处理器 500 处理与给定存储器单元相关的数据。例如，图 10 图示了耦合于位线 1 的存储器单元的情况。相应的感测放大器 212-1 包括出现感测放大器数据的节点。在优选实施例中，该节点采用存储数据的 SA 锁存器 214-1 的形式。类似地，相应组的数据锁存器 430-1 存储与耦合于位线 1 的存储器单元相关的输入或输出数据。在优选实施例中，该组数据锁存器 430-1 包括用于存储 n 位数据的足够的数据锁存器 434-1. . . . . 434-n。

[0119] 当传输门(transfer gate)501被一对互补的信号SAP和SAN使能时，公共处理器500的PBUS 505具有经由SBUS 422对SA锁存器214-1的访问。类似地，当传输门502被一对互补的信号DTP和DTN使能时，PBUS 505具有经由DBUS423对该组数据锁存器430-1的访问。将信号SAP、SAN、DTP和DTN明确图示为来自堆栈总线控制器410的控制信号的部分。

[0120] 图11A图示了图10所示的公共处理器的输入逻辑的优选实施例。输入逻辑520接收在PBUS 505上的数据，且取决于控制信号，使得输出BSI为原样、或被反转、或浮置。输出BSI节点实质上受传输门522或上拉电路或下拉电路的输出的影响，该上拉电路包括串联到Vdd的p晶体管524和525，该下拉电路包括串联到地的n晶体管526和527。上拉电路具有到分别由信号PBUS和ONE控制的p晶体管524和525的栅极。下拉电路具有到分别由信号ONEB<1>和PBUS控制的n晶体管526和527的栅极。

[0121] 图11B图示了图11A的输入逻辑的真指表。该逻辑由PBUS和控制信号ONE、ONEB<0>、ONEB<1>控制，这些控制信号是来自堆栈总线控制器410的控制信号的一部分。实质上，支持三个传输模式，通过(PASSTHROUGH)、反转(INVERTED)和浮置(FLOATED)。

[0122] 在BSI与输入数据相同的PASSTHROUGH模式的情况下，信号ONE处于逻辑“1”，ONEB<0>处于“0”，且ONEB<1>处于“0”。这将禁用上拉或下拉，而使得传输门522能够将PBUS 505上的数据传递到输出523。在BSI是输入数据的反转的INVERTED模式的情况下，信号ONE处于“0”，ONEB<0>处于“1”，且ONEB<1>处于“1”。这将禁用传输门522。而且，当PBUS处于“0”时，下拉电路将被禁用，而上拉电路被使能，导致BSI处于“1”。类似地，当PBUS处于“1”时，上拉电路将被禁用，而下拉电路被使能，导致BSI处于“0”。最后，在FLOATED模式的情况下，可以通过使得信号ONE处于“1”，ONEB<0>处于“1”，且ONEB<1>处于“0”来浮置该输出BSI。为了完整起见列出FLOATED模式，尽管实际上不使用它。

[0123] 图12A图示了图10所示的公共处理器的输出逻辑的优选实施例。来自输入逻辑520的在BSI节点处的信号被锁存到处理器锁存器PLatch 520中。输出逻辑530从PLatch 520的输出接收数据MTCH和MTCH*，并取决于控制信号，在处于PASSTHROUGH、INVERTED或FLOATED模式任一时在PBUS上输出。换句话说，四个分支充当PBUS 505的驱动器，主动地将其拉到HIGH(高)、LOW(低)、或FLOATED(浮置)状态。这通过PBUS 505的四个分支电路、即两个上拉电路和两个下拉电路来实现。第一上拉电路包括串联到Vdd的p晶体管521和532，且能够在MTCH处于“0”时上拉PBUS。第二上拉电路包括串联到地的p晶体管533和534，且能够在MTCH处于“1”时上拉PBUS。类似地，第一下拉电路包括串联到Vdd的n晶体管535和536，且能够在MTCH处于“0”时下拉PBUS。第二下拉电路包括串联到地的n晶体管537和538，且能够在MTCH处于“1”时下拉PBUS。

[0124] 本发明的一个特征是用PMOS晶体管来构成上拉电路且用NMOS晶体管来构成下拉电路。由于NMOS的拉动比PMOS的拉动强得多，因此在任何竞争中，下拉将总是胜于上拉。换句话说，节点或总线可以总是缺省为上拉或“1”状态，且如果希望，可以总是通过下拉翻转(flip)到“0”状态。

[0125] 图12B图示了图12A的输出逻辑的真值表。该逻辑由从输入逻辑锁存的MTCH、MTCH*和控制信号PDIR、PINV、NDIR、NINV控制，这些控制信号是来自堆栈总线控制器410的控制信号的一部分。支持四个操作模式，通过(PASSTHROUGH)、反转(INVERTED)、浮置

(FLOATED) 和预充电 (PRECHARGE)。

[0126] 在 FLOATED 模式中，禁用所有四个分支。这通过使得也是缺省值的信号 PINV ＝ 1、NINV ＝ 0、PDIR ＝ 1、NDIR ＝ 0 来实现。在 PASSTHROUGH 模式中，当 MTCH ＝ 0 时，它要求 PBUS ＝ 0。这通过仅使能具有 n- 晶体管 535 和 536 的下拉分支、而除了 NDIR ＝ 1 之外所有控制信号处于其缺省值来实现。当 MTCH ＝ 1 时，将要求 PBUS ＝ 1。这通过仅使能具有 p- 晶体管 533 和 534 的上拉分支、而除了 PINV ＝ 0 之外所有控制信号处于其缺省值来实现。在 INVERTED 模式中，当 MTCH ＝ 0 时，将要求 PBUS ＝ 1。这通过仅使能具有 p- 晶体管 531 和 532 的上拉分支、而除了 PDIR ＝ 0 之外所有控制信号处于其缺省值来实现。当 MTCH ＝ 1 时，要求 PBUS ＝ 0。这通过仅使能具有 n- 晶体管 537 和 538 的下拉分支、而除了 NINV ＝ 1 之外所有控制信号处于其缺省值来实现。在 PRECHARGE 模式中，PDIR ＝ 0 和 PINV ＝ 0 的控制信号设置将在 MTCH ＝ 1 时使能具有 p 晶体管 531 和 532 的上拉分支，或在 MTCH ＝ 0 时使能具有 p 晶体管 533 和 534 的上拉分支。

[0127] 在美国专利申请公开号 US-2006-0140007A1 中更全面地展开了公共处理器操作，其全部被引用附于此。

[0128] 高速缓存操作中的数据锁存器的使用

[0129] 本发明的一些方面利用在以上图 10 中所述的读 / 写堆栈的数据锁存器，用于将在内部存储器正进行诸如读、写或擦除的其他操作的同时、进数据 (datain) 和出数据 (data out) 的高速缓存操作。在上述架构中，数据锁存器由一些物理页共享。例如，在位线的读 / 写堆栈上时，由所有字线共享，因此当正进行一个操作时，如果这些锁存器中有任何是空闲的，则它们可以高速缓存用于在同一字线或另一字线中的进一步操作的数据，节省传输时间，因为这可以隐藏在另一操作背后。这可以通过增加不同操作或操作的不同阶段的流水线 (pipelining) 量来改进性能。在一个例子中，在高速缓存编程操作中，在编程一页数据的同时，可以载入另一页数据，节省了传输时间。再例如，在一个示例实施例中，在一个字线上的读操作被插入在另一字线上的写操作中，允许来自读操作的数据从存储器传输出，而数据写仍继续。

[0130] 注意，这允许来自同一块中的另一页、而不是不同字线上的数据被切换出 (toggle out)(以例如进行 ECC 操作)，同时对第一页数据的读或其他操作正进行。这种操作的阶段间流水使得用于数据传输的时间被隐藏在对第一页数据的操作背后。更通常地，这使得一个操作的一部分被插入在另一操作、典型是更长的操作的阶段之间。另一个例子是将感测操作插入到所谓擦除操作的阶段之间，比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。

[0131] 为了讨论一些操作所需的相对时间，用于上述系统的一组示例时间值可以取为如下：

[0132] 数据写：～ 700 μ s(较低页～ 600 μ s，较高页 800 μ s)

[0133] 二进制数据写：～ 200 μ s

[0134] 擦除：～ 2500 μ s

[0135] 读：～ 20-40 μ s

[0136] 读和切换出数据 :2KB 数据，～ 80 μ s ;4KB ～ 160 μ s ;8KB ～ 320 μ s

[0137] 这些值可以用于参考以给出以下时序图所涉及的相对时间的思想。如果存在带有

不同阶段的长的操作，则如果锁存器可用，主要方面将使用读 / 写堆栈的共享锁存器在较快的操作中穿插 (interpose)。例如，读可以被插入到编程或擦除操作中，或二进制编程可以被插入到擦除中。在例如要被切换出且更改的数据的读被插入到数据写的验证阶段中的情况下，主要示例实施例将在针对共享相同读写堆栈的另一页的编程操作期间对一页切入 (toggle in) 和 / 或切换出数据。

[0138] 可以以许多方式提高开放数据锁存器的可用性。通常，对于每个单元存储 n 位的存储器来说，对每个位线将需要 n 个这种数据锁存器；但是，并不总是需要所有这些锁存器。例如，在以较高页 / 较低页格式存储数据的每个单元二位的存储器中，在编程较低页时将需要两个数据锁存器。更通常地，对于存储多页的存储器，仅当编程最高页时才需要所有锁存器。这剩余了可用于高速缓存操作的其他锁存器。另外，即使当写最高页时，随着从写操作的验证阶段移除各种状态，锁存器将释放 (free up)。具体地，一旦仅剩下要验证最高状态，则为了验证仅需要单个锁存器，且其他锁存器可以用于高速缓存操作。

[0139] 以下讨论将基于每个单元存储两位且具有用于每个位线上的数据的两个锁存器和用于快速通过写 (quick pass write) 的一个另外的锁存器的四状态存储器，如以上并入的与本申请同时提交的题为“Use of Data Latches inMulti-Phase Programming of Non-Volatile Memories(非易失存储器的多阶段编程中数据锁存器的使用)”的美国专利申请中描述的。写较低页、或擦除、或进行后擦除软编程 (post erase soft program) 的操作基本上是二进制操作，且具有空闲的数据锁存器之一，其可以使用它来高速缓存数据。类似地，在进行较高页或全序列写的情况下，一旦除了最高级以外所有都被验证，则仅单个状态需要验证，且存储器可以释放可用于缓存数据的锁存器。可以如何使用它的例子是当编程一页时，比如在复制操作中，可以在该写的验证阶段期间塞入 (slip in) 对共享同一组数据锁存器的另一页、比如在同一组位线上的另一字线的读。然后，可以将地址切换到正被写的页，允许在停止处捡起 (pickup) 写处理而无需重新开始。当写继续时，在穿插的读期间缓存的数据可以被切换出、检验或更改，并被传输回以呈现用于一旦较早的写操作完成则将其写回去。这种高速缓存操作使得第二页数据的切换出和更改被隐藏在第一页的编程背后。

[0140] 作为第一例子，以单页 (较低页 / 较高页格式) 编程模式操作的二位存储器的高速缓存编程操作。图 13 是图 10 的简化版本，示出了与两位的实施例中的本讨论有关的一些具体元件，省去了其他元件以简化讨论。这些元件包括与数据 I/O 线 231 连接的数据锁存器 DL0434-0、通过线路 423 连接到公共处理器 500 的数据锁存器 DL1 434-1、通过线路 435 与其他数据锁存器共同连接的数据锁存器 DL2 434-2、以及通过线路 422 连接到公共处理器 500 的感测放大器数据锁存器 DLS 214。图 13 的各种元件根据其在编程较低页期间的部署而标注。锁存器 DL2 434-2 用于快速通过写模式中的较低验证 (VL)，如在与本申请同时提交的题为“Use of Data Latches in Multi-Phase Programmingof Non-Volatile Memories(非易失性存储器的多阶段编程中数据锁存器的使用)”的美国专利申请中描述的；对寄存器的包括以及当包括寄存器时对使用快速通过写的包括是可选的，但是示例实施例将包括该寄存器。

[0141] 较低页的编程可以包括以下步骤：

[0142] (1) 处理通过将数据锁存器 DL0 434-0 复位到缺省值“1”而开始。该惯例用于简

化部分页编程，因为在所选行中的不将被编程的单元将被禁止编程。

[0143] (2) 编程数据沿着 I/O 线 231 供应到 DL0 434-0。

[0144] (3) 编程数据将被传输到 DL1 434-1 和 DL2 434-2（如果包括该锁存器且实现快速通过写的话）。

[0145] (4) 一旦编程数据被传输到 DL1 434-1，数据锁存器 DL0 434-0 就可以被复位为“1”，且在编程时间期间，下一数据页可以沿着 I/O 线 231 被加载到 DL0 434-0，允许在正写第一页的同时缓存第二页。

[0146] (5) 一旦第一页被载入到 DL1 434-1 中，就可以开始编程。DL1 434-1 数据用于锁定 (lockout) 单元免于进一步编程。DL2 434-2 数据用于较低验证锁定，该较低验证锁定管理到快速通过写的第二阶段的转变，如与本申请同时提交的题为“Use of Data Latches in Multi-Phase Programming of Non-VolatileMemories（非易失性存储器的多阶段编程中的数据锁存器的使用）”的美国专利申请中所描述。

[0147] (6) 一旦开始编程，在编程脉冲之后，较低验证的结果被用于更新 DL2434-2；较高验证的结果被用于更新 DL1 434-1。（该讨论基于“传统”编码，其中较低页编程要到 A 状态。在与本申请同时提交的题为“Use of Data Latchesin Multi-Phase Programming of Non-Volatile Memories（非易失性存储器的多阶段编程中的数据锁存器的使用）”和在 2005 年 3 月 16 日提交的题为“Non-Volatile Memory and Method with Power-Saving Read andProgram-Verify Operations（非易失性存储器和功率节省读和编程验证操作的方法）”的美国专利申请中讨论了这种和其他编码。容易得到本讨论对于其他编码的扩展。）

[0148] (7) 在确定编程是否完成时，仅检查行单元（或编程的适当物理单位）的 DL1 434-1 寄存器。

[0149] 一旦较低页被写入，就可以编程较高页。图 14 示出了与图 13 相同的元件，但指示在读入较低页数据的情况下用于较高页编程的锁存器分配。（该描述再次使用传统编程，以便较高页的编程要到 B 和 C 状态。）较高页的编程可以包括以下步骤：

[0150] (1) 一旦较低页结束编程，则在保持了（未执行的）高速缓存编程命令的情况下，较高页（或下一页）写入将以来自状态机控制器的信号而开始。

[0151] (2) 该编程数据将从 DL0 434-0 传输（在其在较低页写期间在步骤 (3) 中被载入的情况下）到 DL1 434-1 和 DL2 434-2。

[0152] (3) 较低页数据将从阵列被读入且置于 DL0 434-0 中。

[0153] (4) 再次分别使用 DL1 434-1 和 DL2 434-2 用于验证高和验证低锁定数据。检查锁存器 DL0 434-0（持有较低页数据）作为编程参考数据，但不用验证结果来更新它。

[0154] (5) 作为验证 B 状态的一部分，在较低验证 VBL 处的感测之后，由此将在 DL2 434-2 中更新数据，而用高验证 VBH 结果来更新 DL1 434-1 数据。类似地，C 验证将具有相应的命令以用相应的 VCL 和 VCH 结果来更新 DL2434-2 和 DL1 434-1。

[0155] (6) 一旦 B 数据完成，则由于仅需要进行对于 C 状态的验证，因此不需要（保持在 DL0 434-0 中用于参考的）较低页数据。DL0 434-0 被复位为“1”，且另一页编程数据可以从 I/O 线 231 载入，并被缓存锁存器 DL0 434-0 中。公共处理器 500 可以设置仅要验证 C 状态的指示。

[0156] (7) 在确定较高页编程是否完成时，对于 B 状态，检查锁存器 DL1 434-1 和 DL0

434-0 两者。一旦单元被编程到 B 状态且仅在验证 C 状态，则仅需要检查锁存器 DL1 434-1 数据来看是否存在任何未被编程的位。

[0157] 注意，在该布置下，在步骤 6 中，不再需要锁存器 DL0 434-0，且其可用于缓存下一编程操作的数据。另外，在使用快速通过写的实施例中，一旦进入第二、慢编程阶段，则还可以使得锁存器 DL2 434-2 可用于缓存数据，虽然实际上通常是如下情况：以这样的方式，这仅可用于不调整 (justify) 通常用于实现该特征所需的额外开销的较短时间段。

[0158] 图 15 可以用于图示以在上几段中描述的单页模式进行高速缓存编程的许多方面。图 15 示出了在存储器内部（较低“实际忙碌 (True Busy)”线）发生了什么事件和从存储器外部（较高“高速缓存忙碌 (Cache Busy)”线）看上去的相对时间。

[0159] 在时间 t_0，要被编程到所选字线 (WLn) 上的较低页被载入该存储器中。这假定先前没有缓存第一较低页的数据，因为其将用于随后的页。在时间 t_1，较低页完成载入，且该存储器开始向其写入。由于在这点上这与二进制操作等效，因此仅需要验证状态 A(“pvfyA”) 且数据锁存器 DL0 434-0 可用于接收下一页数据，在此将该下一页数据取作要在时间 t_2 被编程到 WLn 中的较高页，其因此在编程较低页期间被缓存到锁存器 DL0 434-0 中。较高页在时间 t_3 完成载入，且较低页在 t_4 时一完成，就可以编程该较高页。在该布置下，虽然所有数据（较低和较高页）要被写入编程的物理单位中（在此，字线 WLn），但是在可以写较高页数据之前，该存储器必须从时间 t_3 等待到时间 t_4，不像以下描述的全序列实施例那样。

[0160] 较高页编程在时间 t_4 开始，其中首先仅验证 B 状态（“pvfyB”），且在 t5 添加 C 状态（“pvfyB/C”）。一旦在 t_6 不再验证 B 状态，就仅需要验证 C 状态（“pvfyC”），且锁存器 DL0 434-0 变为空闲。这允许在较高页完成编程的同时缓存下一数据组。

[0161] 如所知的，根据具有高速缓存编程的单页算法，如图 15 所示，即使较高页数据可能在时间 t_3 可用，但是该存储器在开始写该数据之前将等待直到时间 t_4。在转换到全序列编程操作时，比如在美国专利申请 11/013125 中更全面展开的，一旦较高页可用，就可以同时编程较高和较低页数据。

[0162] 用于全序列（低到全变换）写中的高速缓存编程的算法以如上的较低页编程开始。因此，步骤 (1)-(4) 与用于单页编程模式中的较低页处理一样：

[0163] (1) 该处理通过将数据锁存器 DL0 434-0 复位为缺省值“1”而开始。该惯例用于简化部分页编程，因为在所选行中的不将被编程的单元将被禁止编程。

[0164] (2) 编程数据沿着 I/O 线 231 被供应到 DL0 434-0。

[0165] (3) 编程数据将被传输到 DL1 434-1 和 DL2 434-2（如果包括该锁存器且实现快速通过写的话）。

[0166] (4) 一旦编程数据被传输到 DL 1 434-1，数据锁存器 DL0 434-0 就可以被复位为“1”，且在编程时间期间，下一数据页可以沿着 I/O 线 231 被载入 DL0 434-0，允许在正写第一页的同时缓存第二页。

[0167] 一旦载入第二页数据，如果该第二页数据对应于正被写入的较低页的较高者、且较低页还没有完成编程，则可以实现向全序列写的转换。该讨论关注这种算法中的数据锁存器的使用，其许多其他细节在共同未决的、共同转让的美国专利 No. 7120051 中更全面地展开。

[0168] (5) 在较高页数据被载入锁存器 DL0 434-0 中之后，将在地址块中进行判断，来检

查 2 个页是否在同一字线和同一块上，且一页是较低页，一页是较高页。如果是，则如果允许的话，编程状态机将触发较低页编程到全序列编程转换。在任何未决的验证完成之后，则实现该变换。

[0169] (6) 当编程序列从较低页改变为全序列时，典型地一些操作参数将改变。在示例实施例中，这些包括：

[0170] (i) 如果较低页数据还没有被锁定，则针对脉冲验证周期数的最大编程循环将从较低页算法的最大编程循环改变为全序列的最大编程循环，但是已完成的编程循环数将不会通过转换而被复位。

[0171] (ii) 如图 16 所示，编程波形以在较低页编程处理中使用的值 VPGM_L 开始。如果编程波形已经发展到超过在较高页处理中使用的开始值 VPGM_U 的情况，则在向全序列转换时，该阶梯将退回到在继续升高阶梯之前的 VPGM_U。

[0172] (iii) 确定编程脉冲的步长和最大值的参数不改变。

[0173] (7) 应该进行对存储器单元的当前状态的全序列读，来保证对于多级编码将编程正确的数据。这确保了当全序列开始时，以前可能已被锁定在较低页编程中的、但需要进一步编程来考虑其较高页数据的状态不被禁止编程。

[0174] (8) 如果激活了快速通过写，则还将更新锁存器 DL2 434-2 的数据，以反映较高页编程数据，因为其以前是基于仅针对 A 状态的较低验证的。

[0175] (9) 然后，以多级、全序列编程算法再继续编程。如果较低页处理中的编程波形已经增加到较高页起始电平以上，则在转换时将该波形退回到该电平，如图 16 所示。

[0176] 图 17 是在较低页向全序列转换写处理中涉及的相对时间的示意表示。直到时间 t_3，该处理是如上所述用于图 15 的处理。在 t_3，较高页的数据已经被载入，且进行向全序列算法的变换，验证处理被切换为包括 B 状态和 A 状态。一旦所有 A 状态都被锁定，则验证处理在时间 t_4 切换到检查 B 和 C 状态。一旦在 t_5 验证了 B 状态，则仅需要检查 C 状态，且可以释放寄存器来载入接下来要被编程的数据，比如在下一字线 (WL_{n+1}) 上的较低页，如高速缓存忙碌线 (Cache Busy line) 上指示的。在时间 t_6，该下一数据组已经被高速缓存，且用于前一组的 C 数据的编程在 t_7 终止，下一数据组开始编程。另外，在（此时）字线 WL_{n+1} 上的较低页正编程的同时，下一数据（比如相应的较高页数据）可以被载入开放锁存器 DL0 434-0 中。

[0177] 在全序列写期间，以独立地给出较低页和较高页状态的方式来实现状态报告。在编程顺序结束时，如果存在未完成的位，则可以进行物理页的扫描。第一扫描可以检查锁存器 DL0 434-0 以寻找未完成的较高页数据，第二扫描可以检查 DL1 434-1 以寻找未完成的较低页数据。由于 B 状态的验证将改变 DL0 434-0 和 DL1 434-1 数据两者，因此应该以如下方式进行 A 状态验证：如果该位的阈值高于 A 验证电平，则 DL1 434-1 数据“0”将改变为“1”。该后验证将检查任何编程下的 B 电平是否在 A 电平处通过；如果它们在 A 电平处通过，则错误仅出现在较高页而不在较低页；如果它们不在 A 电平处通过，则较低和较高页都有错误。

[0178] 如果使用高速缓存编程算法，则在编程了 A 和 B 数据之后，C 状态将被转移到锁存器 DL1 434-1 来完成编程。在该情况下，扫描锁存器对较低页来说不是必须的，因为该较低页将已经通过了编程，而没有任何失败的位。

[0179] 本发明的另一组示例实施例涉及页复制操作，其中数据组从一个位置被重新定位到另一位置。数据重新定位操作的各个方面在美国专利申请 No.US-2006-0257120-A1；US-2006-0136687-A1；以及 US-2006-0031593-A1；和美国专利号 6266273 中描述，其全部被引用附于此。当将数据从一个位置复制到另一位置时，通常切换出该数据以被检查（例如以寻找错误）、更新（比如更新头标 (header)）或这两者（比如校正所检测的错误）。这种转移还要在垃圾收集操作中整理 (consolidate) 数据。本发明的首要方面允许被读到开放寄存器的数据在写操作的验证阶段期间被插入，而被高速缓存的该数据然后随着写操作的继续而被移出存储器器件，使得用于切换出该数据的时间被隐藏在写操作背后。

[0180] 以下呈现了高速缓存页复制操作的两个示例实施例。在两种情况下，描述了使用快速通过写实施方式的实施方式。图 18 指示了随着处理的进展锁存器的示例布置的部署。

[0181] 高速缓存页复制的第一版本将向较低页写入，且可以包括以下步骤，其中读地址被标注为 M、M+1..... 且写地址被标注为 N、N+1..... ：

[0182] (1) 要复制的页（“页 M”）被读入锁存器 DL1 434-1 中。这可以是较高页或较低页的数据。

[0183] (2) 然后，页 M 被转移到 DL0 434-0 中。

[0184] (3) 然后在 DL0 434-0 中的数据被切换出，并被更改，之后其被移回到该锁存器中。

[0185] (4) 然后，编程序列可以开始。在要被写入较低页 N 中的数据被转移到 DL1 434-1 和 DL2 434-2 之后，锁存器 DL0 434-0 准备好用于高速缓存数据。该较低页将被编程。对于该实施例，编程状态机将在此停止。

[0186] (5) 要被复制的下一页然后被读入 DL0 434-0 中。然后，编程可以恢复。在步骤 (4) 结束时停止的状态机将从头重新开始该编程序列。

[0187] (6) 编程继续直到较低页完成。

[0188] 复制目的地页地址将确定写是向较低页还是向较高页。如果编程地址是较高页地址，则该编程序列将不停止，直到该编程完成，且在写完成之后执行步骤 (5) 的读。

[0189] 在第二高速缓存页复制方法中，可以暂停编程 / 验证处理来插入读操作，然后重新开始写操作，在其停下的点捡起。然后，在恢复的写操作继续的同时，在该交错的感测操作期间所读的数据可以被切换出。而且，该第二处理允许一旦仅 C 状态正被验证且在每个位线上的一个锁存器开放、页复制机制就用于在较高页或全序列写处理中。第二高速缓存页复制操作以与第一种情况相同的前三个步骤开始，但然后不同。其可以包括以下步骤：

[0190] (1) 要复制的页（“页 M”）被读入锁存器 DL1 434-1 中。这可以是较低页或较高页。

[0191] (2) 然后，来自页 M 的数据被转移到 DL0 434-0 中。（如之前，N 等将指示写地址，M 等用于读地址。）

[0192] (3) 在 DL0 434-0 中的数据然后被切换出、更改，之后其被移回到该锁存器中。

[0193] (4) 状态机编程将进入无限期等待状态，直到读命令进入，且另一页、即下一页 M+1 向锁存器 DL0 434-0 的读将开始。

[0194] (5) 一旦步骤 (4) 的读完成，地址就被切换回字线和块地址，以将在步骤 (1-3) 中的数据编程到页 N（在此，较低页）中，且恢复编程。

[0195] (6) 在页 M+1 的读完成之后，该数据可以被切换出、更改和返回。一旦该处理结束，如果两页是在同一 WL 上的相应较高和较低页，则该写就可以被转换为全序列操作，

[0196] (7) 一旦在全序列写中完成了 A 和 B 电平，则在 DL0 434-0 中的数据将被转移到 DL1 434-1 中，如在先前描述的正常高速缓存编程中一样，且可以发出对于另一页（例如，页 M+2) 的读命令。如果没有单个页向全序列转换，则较低页将完成该写，且然后较高页将开始。在完全完成了 B 电平状态之后，相同 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 的读命令的状态。

[0197] (8) 一旦读命令到达，该地址就被切换到读地址，且读出下一页（页 M+2)。

[0198] (9) 一旦该读完成，该地址就将被切换回先前的较高页地址（编程地址 N+1)，直到该写结束。

[0199] 如上所述，除了在保持可以被编程到每个存储器单元中的（在此，2 位）数据时使用的锁存器 DL0 434-0 和 DL1 434-1 之外，示例实施例还包括用于快速通过写技术的较低验证的锁存器 DL2 434-2。一旦较低验证通过，还可以释放锁存器 DL2 434-2，并将其用于高速缓存数据，虽然在示例实施例中没有这样做。

[0200] 图 19A 和 19B 图示了第二高速缓存页复制方法的相对时序，其中图 19B 图示了具有全序列写转换的算法，且图 19A 图示了没有其的算法。（图 19A 和 19B 都由两部分组成，第一较高部分在虚线垂直线 A 处开始，对应于 t_0，且以虚线垂直线 B 结束，对应于 t_5；第二较低部分是较高部分的继续，且以虚线垂直线 B 开始，对应于 t_5。在两种情况下，在时间 t5 的线 B 在较低部分中与在较高部分中相同，恰好是在两个部分中的接缝处，允许其在两个线上显示）。

[0201] 图 19A 示出了以在该例子中被取为较低页的第一页（页 M) 的读开始、假设先前没有高速缓存数据、且以单页模式操作的处理，其中在开始写较高页之前等待直到较低页完成写为止。该处理在时间 t_0 以页 M 的读（感测页 M(L)）开始，该页 M 在此是较低的，其在该编码中通过在 A 和 C 电平处的读而被感测。在时间 t_1，读完成，且页 M 可以被切换出、检验或更改。在时间 t_2 开始，通过在 B 电平处读来感测下一页（在此是页 M+1，对应于与较低页 M 相同物理 (physical) 的较高页），该处理在时间 t_3 结束。此时，第一页（来源于页 M)（较低）准备好被编程回到在页 N 处的存储器中，且从页 M+1 读取的数据正被保持在锁存器中，且可以移出以被更改 / 检查。这两个处理可以同时开始，在此在 t_3 开始。使用上述的典型时间值，来自页 M+1 的数据到时间 t_4 已经被移出且更改；但是，对于不实施全序列转换的实施例，该存储器将等待直到页 N 在时间 t_5 结束以开始将（来源于页 M+1 的）第二读出页的数据写入页 N+1 中。

[0202] 由于页 N+1 是较高页，因此其写首先以在 B 电平处的验证开始，在 t_6 添加 C 电平。一旦具有目标状态 B 的存储元件在时间 t7 都被锁定（或到达了最大计数），就放弃 (drop) B 状态验证。如上所述，根据本发明的几个主要方面，这允许释放数据锁存器，暂停正进行的写操作，穿插 (interpose) 读操作（在与暂停的编程 / 验证操作不同的地址处），然后在写停下的地方恢复，且在穿插的写操作中感测的数据可以在恢复的写操作运行的同时被切换出。

[0203] 在时间 t7，对于在此的较低页 M+2 进行穿插的写操作。在时间 t8 结束了该感测，且拾起页 N+1 的写，且来自页 M+2 的数据同时被切换出并更改。在该例子中，在页 M+2 在时

间 t10 结束之前，页 N+1 在时间 t9 结束编程。在时间 t10，来源于页 M+2 的数据的写可以开始；但是，在该实施例中，而是，首先执行页 M+3 的读，允许要切换出和更改的该页数据被隐藏在开始于时间 t11 时的来源于页 M+2 的数据向页 N+2 中写入的背后。然后，该处理如在该图的较前部分中一样继续，但页号转变，时间 t11 对应于时间 t3，时间 t12 对应于时间 t4，等等，直到复制处理停止。

[0204] 图 19B 再次示出了以读较低页、即被取为较低页的页 M 开始、且假设先前没有高速缓存数据的处理。图 19B 与图 19A 不同在于在时间 t4 实现向全序列写的转换。这大致将处理加快了图 19A 的时间 (t5-t4)。在时间 t4(＝图 19A 中的 t5)，如先前所述那样实施与全序列转换有关的各种改变。在此之外，该处理类似于图 19A 的处理，包括在时间 t7 和 t12 之间得到的本发明的那些方面。

[0205] 在涉及写数据的在此描述的页复制处理和其他技术中，可以根据美国专利公开号 US-2004-0109362-A1 中所描述的线索智能地选择在给定时间验证哪些状态，该美国专利公开被引用附于此。例如，在全序列写中，该写处理可以开始仅验证 A 电平。在之前的 A 验证之后，检查以看任何位是否都通过了。如果是，可以向验证阶段添加 B 电平。在具有 A 电平验证作为其目标值验证的所有存储单元后（或除了基于可设置的参数的最大计数以外），将移除 A 电平验证。类似地，在 B 电平处的验证之后，可以添加 C 电平的验证，且在具有 B 电平验证作为其目标值验证的所有存储单元后（或除了基于可设置的参数的最大计数以外），移除 B 电平验证。

[0206] 关于优选的多状态编码来描述带有用于其他操作的背景数据高速缓存的编程操作。

[0207] 用于 4 状态存储器的示例优选“LM”编码

[0208] 图 20 图示了用 2 位逻辑码（“LM”码）来编码的 4 状态存储器的编程和读。该码提供了错误容限，且缓解了由于 Yupin 效应的相邻单元耦合。图 20 图示了当每个存储器单元使用 LM 码来存储两位数据时 4- 状态存储器阵列的阈值电压分布。LM 编码不同于传统格雷码之处在于，保留较高和较低位用于状态“A”和“C”。“LM”码已经在美国专利 No.6657891 中公开，且有益于通过避免需要电荷的很大改变的编程操作来减少在相邻浮置栅极之间的场效应耦合。

[0209] 设计该编码使得可以分离地编程和读取 2 位，即较低和较高位。当编程较低位时，该单元的阈值电平仍然处于未编程区域中，或被移动到阈值窗的“中下 (lower middle)”区域。当编程较高位时，在这两个区域的任一中的阈值电平进一步提高到不大于阈值窗的四分之一的稍高的电平。

[0210] 数据样式的伪 (psuedo) 和用户驱动随机化

[0211] 存储器 EEPROM 或芯片和随机化方法的各种实施例追求最小化由于重复数据存储样式引起的问题，比如增加的 NAND 串电阻、降低的容忍度和可靠性以及不希望的耦合。本发明的伪随机技术是实用的，且在数据处理容量方面，它们实现起来不贵。

[0212] 本发明包括实现对在闪存芯片上存储的数据的伪随机化和基于真实用户的随机化的不同实施例和方法。所有实施例有如下优点：仅需要在快闪 EEPROM 中实现简单且小的电路更改。这是值得注目的，因为随机化技术和电路在计算上强度不大，且即使有性能损失也只损失很少就实现了。本发明的解决方案也很灵活，在于可以在任何时间容易地使能或

禁用随机化。另外，在某些实施例中使用的伪随机化的样式可以以许多方式变化，且在时间上容易改变。

[0213] 图 21 图示了与随机化处理有关的 EEPROM 或存储器芯片 600 的主要组件。芯片 600 包括存储器阵列 602、在外围电路中的（一个或多个）寄存器 610、以及复用器 614。芯片 600 的其他组件将在另外的附图中图示并参考这些附图描述。寄存器 610 能够保持多位，且可以包括多个寄存器。在一些实施例中，其起到移位寄存器的作用。存储器阵列 602 包括隐藏区 604 和用户数据区 606。隐藏区可以用于存储固件和其他开销数据，比如存储器操作控制码。在 NAND 架构中，如先前描述的，数据被组织在块中，每块可以包括多页数据。在某些实施例中，将既不出现寄存器 610 也不出现复用器 614。

[0214] 本发明的各个实施例将减少或消除可能导致编程干扰或用户读干扰的、对具体数据样式的长期且重复的存储。这通过用伪随机机制或用户触发的随机化来变化数据的编码而实现。由于用户活动性的定时完全不可预测，因此使用该活动性作为触发器得到了编码方案的真正随机序列。每个实施例还将减少 NAND 串电阻效应，增加存储器容忍度和可靠性，且减少浮置栅极与浮置栅极耦合的问题。

[0215] 每个实施例仅需要对快闪 EEPROM 的电路进行最小的更改，但同时将显著地增加数据存储的随机性，因此增加 EEPROM 的性能。可以容易地在该阵列中使能或禁用数据的随机化。另外，负责该伪随机化的序列可以持续改变，提供了系统内的灵活性。

[0216] 在一个实施例中，可以是零或一的位的码或序列被存储在阵列 602 的隐藏区 604 中。隐藏区 604 中存储了该码的部分可以被称为“ROM 块”。该码可以包括 2 个或更多位，但优选地包括 17 或更多位。位越多，随机化将越大。在芯片 600 通电后，值被载入寄存器 610 中。在该寄存器中的每位都被分配到具体的页地址。将每位与页的页地址比较，且基于该比较，该页的数据的编码将被反转，或对该页将仍然相同（通过 (passed)）。例如，该位的 0 值可以用于指示该数据的编码方案将仍然相同，而该寄存器中的 1 值可以指示在页内的数据的编码将被反转。如果该码包括少于一块内的页的数量的位，则该码可以应用于不止一个一页或多页的组。换句话说，该码可以连续重复使用，直到所有页都被比较。该码还可以在各周期之间改变。或者，可以通过复用器 614 来复用该码，使得一个码的一位将确定在用户数据区 602 中存储的多页数据的编码。该码的每位可以称为极性位 (polarity bit)，因为其用于改变对用户数据的某部分采用的编码的极性。这在图 22A 中绘出。在这种情况下，编码是基于页地址的，以便知道页 0、N 具有极性 1，而页 1、n+1 具有极性 0，页 2、n+2 具有极性 1，等等。因此，在编码是基于页地址的实施例中，不需要存储页的极性位，虽然为了冗余可以这样做。

[0217] 以下看到的并如图 22B 再现的表 1 图示了在寄存器 610 中的码的极性位应用于用户数据的部分。虽然可以将用户数据的任意部分与具体极性位比较且相关联，但所描述的优选实施例图示了一页作为基本单位。

[0218]

寄存器位置	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
码（极性位）	1	0	1	0	0	1	1	1	1	0	0	0	1	1	0	1	0
UD 原始编码	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
UD 随后编码	1	1	1	1	0	0	1	0	1	1	0	1	1	0	0	0	0
状态	ER		ER		B		C		ER		A		C		B		

[0219] 表 1

[0220] 如表中可见，码的每个（极性）位将确定位的原始编码（数据）是将仍然相同还是将改变。例如，见寄存器位置 1，在该位置中的极性码具有值 1。因此，在 1 指示数据将被反转的实施例中，被存储为 0 的用户数据的原始位将被反转为值 1。该表图示了多状态单元，其中 2 位用于定义一个状态。图 20 中示出了各状态，且从图 20 中可见，较高和较低位定义了这些状态。在图 20 所示的这类 2 位或 4 状态存储器单元中，(1:1) 定义了擦除（“ER”）或未编程（“U”）状态；(0:1) 定义了状态 A，(0:0) 定义了状态 B，且 (1:0) 定义了状态 C。较高和较低位可以物理上位于单个存储器单元中。相同或不同码可以被应用于另一组数据，以便对应于位 17 的数据组将与对应于所应用的下一码的位 1 的数据结合使用来确定状态。每个极性寄存器将控制在相应的页上的所有数据的极性。优选地，较低和较高位位于相同物理字线上。在表 1 中给出的例子图示了用于将单个样式转换为经过许多字线的（一个或多个）随机样式的极性位的功能。对位于相同 NAND 链结构的数据实现随机化，在图 22C 中提供其例子以图示该概念。

[0221] 在图 22C 中，在图示的 NAND 串或链的每个单元处图示了给定单元的较低和较高位。所示的 NAND 串仅是例子，且当然可以在一串中存在更多或更少的单元，且本发明可以使用不同于所示的结构。例如，还可以使用存储 3、4 或更多位的单元。而且，应该记得，极性位优选地应用于一页或更多数据，虽然示出了在位电平上的应用以图示位反转的概念，尤其是在多状态存储器中。在图 22C 中，向用户数据的每位应用极性位，且由极性位的反转或通过得到的用户数据被标注为保存的数据 (saved data)。该保存的数据是随后将被写到存储器阵列并被存储为随机化操作的结果的数据。如可以看到的，由该单元的较高和较低位来定义在图的右侧指示的状态。在图 22C 中使用的“保存的数据”术语对应于在表 1 和图 22B 中所称的“用户数据 (UD) 随后编码”。

[0222] 图 23A 图示了另一实施例，其中图 21 所示的寄存器是有反馈的移位寄存器。在这种实施例中，该寄存器 610 被配置为伪随机生成器。其内容被周期地反馈以生成伪随机数的序列。以此方式，与一次使用码的所有位的实施例相反，将一次使用一位。在到来的用户页上使用的极性位将来自上一寄存器输出。优选地在命令的上升沿这样做。触发命令可以是编程命令、高速缓存编程命令、读命令、擦除命令、或其他用户发出的命令。示例的编程命令信号在图 23B 中示出。示出了与该命令相关的时钟信号，且由用户请求来触发该命令的例化 (instantiation)，该请求的时序和类型是不可预测且实质上是随机的。图 23A 图示了作为确定极性位的输入之一的与用户命令相关的时钟信号。图中的另一输入是用户数据。极性位的应用将该数据的编码反转或维持原样，如先前描述的。

[0223] 图 23C 图示了用于数据转换的控制电路的例子。具有单个反相器的数据路径将

导致编码的反转，而具有串联的两个反相器的路径将导致数据编码方案不改变。在这种情况下，与一组数据相关的极性位将随该组数据被存储。例如，如图 23D 所示，对于一页数据 630，用于用户区 636 中的数据的极性位 632 将被编程到页 630 的隐藏区 634 中。当读该页 630 时，极性位 632 将被移出且被锁存以控制输出数据，且如果编码机制被反转，则将回复 (revert) 编码机制，如由图 23E 所示的示例电路完成的。以此方式，该页的极性将被回复到其原始编码。

[0224] 由移位寄存器使用的码的样式可以改变，且可以更改用于不同应用。如果所有位被设置为 0（在 0 指示不改变的情况下），则将禁用随机化。尽管在寄存器中的位的样式是伪随机的，但用户行为是不可预测的，且在任何给定的时间得到的极性因此也是不可预测且随机的。用户行为的两个例子如下 :1) 用户编程一些页，且跳到不同的地址来读或编程一些页，或擦除一些块，然后返回到上次编程发生的块，且继续编程更多页 ;以及 2) 用户顺序地编程所有页，而不跳到另一地址。在情况 1 中，对于每个用户命令可以触发新的极性位，而在情况 2 中，顺序编程将利用且基于一个极性位。因此，即使用户想要存储的原始数据可能对两个情况来说相同，但是在存储器中最后编程的数据很可能在这 2 个情况下对于各个不同的页和各组页中的至少一些而不同。注意，EEPROM 典型地由控制器芯片控制，且“用户”的一些动作可以是控制器芯片的动作。

[0225] 在另一实施例中，由于用户命令，比如先前描述的高速缓存编程操作，还随机地生成极性位。该实施例使用不同步的两个输入。第一是用户命令的时序，如先前所述其是不可预测的。第二是有限状态机时钟。在某些存储器系统中，有限状态机时钟仅在某些时间（例如，在高速缓存操作期间）是有效的，而在其他系统中，其可以总是有效。无论何时存储器系统的有限状态机时钟有效时，该实施例的此技术都可用。

[0226] 在用户命令时钟信号的上升沿，参考有限状态机（“FSM”）时钟的电平或状态。该状态可以是高或低，如在图 24A 中看到的。低状态可以对应于零的极性位（虽然相反的对应也是可能的）。在时间 t = 0 时，FSM 是低，因此极性位将是零，指示不改变数据编码，如前所述。在时间 t = 1 时，FSM 是高，且极性位将是一，而在时间 t = 3 时，FSM 再次位于低状态。在一些实施例中，只要发出执行命令且感测到它，就将极性位 632 载入隐藏区 634。在其他实施例中，其可以被临时存储在该系统的另一存储器中。图 24B 图示了用于确定如上所述的极性位的示例电路。将再次优选在上升沿触发反相器。

[0227] 图 25 更详细地图示了图 7A 和图 9 中示出的芯片上控制电路。除了状态机 312 和地址解码器或生成器 314 以外，该芯片上控制电路还包含了数据加扰器 318。在优选实施例中，它包含了图 21 和 23A 所示的寄存器 610 和复用器 614。在另一优选实施例中，其还包含图 23C 所示的数据反转电路和图 23E 所示的数据回复电路。

[0228] 如图 21 到图 25 和相关文本所公开的，并行地编程或读可由公共字线访问的每页数据。通过伪随机地选择某些页来使得它们的位的极性翻转，来实现逐页的随机化。

[0229] 在 Yan Li 等人的于 2006 年 9 月 8 日提交的题为“Methods in a PseudoRandom and Command Driven Bit Compensation for the Cycling Effects in FlashMemory（对于闪存中的循环效应的伪随机和命令驱动位补偿中的方法）”的美国申请 NO.11/530392 中公开了数据的芯片上、逐页随机化，该申请全部被引用附于此。

[0230] 一页内的数据的芯片上伪随机化

[0231] 还期望加扰每页内的数据。这有益于避免在编程期间可能引起问题的某些高度规则的数据样式，还有益于当并行感测一页数据时控制源极载入错误。

[0232] 如果重复的数据样式被用户或者控制器存储到某些页中，则数据可以将在某NAND链的编程期间对升压模式(boosting mode)有害的某些样式排队(line up)。当大量NAND链(见图1D和图3)在编程期间共享所选的字线时，不将被编程的链通过使得它们的沟道区域升压以便减少施加到所选字线的有效编程电压，而被禁止编程。例如，NAND型存储器典型地使得每个NAND链从源极侧编程到漏极侧。如果在源极侧上的大量存储器单元处于擦除状态，则由于来自漏极侧的升压沟道的电荷可能经由由擦除的单元建立的高度导电的沟道而向源极泄漏，因此在编程禁止期间该升压沟道将不是非常有效。不太有效的沟道升压和由此的编程禁止可能导致编程干扰和错误的结果。在美国申请公开No.US-2006-0198195-A1和Farookh Moogat等人在2006年12月29日提交的题为“Method of NAND Flash Memory Cell Array withAdaptive Memory State Partitioning(具有适应的存储器状态划分的NAND闪存单元的方法)”的美国申请No.US 11/618482中讨论了与升压效率有关的问题，其全部公开被引用附于此。

[0233] 根据本发明的一个方面，页中的数据被随机化以便当来自几页的数据被排队时，在编程期间避免有问题的数据样式。

[0234] 通过偏移每页的起始位置来加扰

[0235] 在一个优选的实施例中，加扰在一页上的数据的简单方式是将数据写到每个不同的页的独立或不同的起始地址。相应页的数据被写入位于每页的不同起始位置处的每页存储器单元。当数据被填入到页的结尾时，通过从该页的第一地址绕回(wrap around)而继续，直到就在起始位置之前。

[0236] 图26图示了根据从每页的不同起始位置开始写的一个优选实施例加扰存储器页上的数据的方法。

[0237] 步骤700：提供具有可逐页访问的非易失性存储器单元的阵列的集成电路存储器芯片，每页是一群存储器单元，一群的每个存储器单元位于该阵列的列中，且沿着可由公共字线访问的行。

[0238] 步骤710：在芯片上生成起始列位置的序列，以便要被编程的每页具有相关的起始列位置。

[0239] 步骤720：通过从相关起始列位置开始且绕回直到该页被填满，将要被编程的数据安排(stage)到每页中。

[0240] 步骤730：并行将安排的数据编程到每页中。

[0241] 图27是图示具有用于写数据的不同起始位置的不同页的例子的表。例如，在页0上，字节0将从列0开始被载入。在页1上，字节0将从列1开始被载入。该数据将继续通过列n-1载入，并绕回列0。在该例子中，每页具有指定的偏移来帮助错开(misalign)页与页之间的数据中的任何重复样式。通常，给出起始列地址作为页号的函数。当到达物理列的结尾时，数据绕回到物理列的开始处。例如，Starting_Column_Address(Page_Number)(开始列地址(页号))＝Page_Number(MOD(n-1))+k，其中k是预定数，且(n-1)是正被并行编程的存储器单元的总数。例如，当k＝0时，每页从前一页偏移了一列。

[0242] 在优选实施例中，通过控制图9所示的I/O电路440来实现对于给定页的起始列

位置的偏移。典型地，地址解码器 314 在数据载入操作中，将物理页的起始地址发给 I/O 电路。根据起始地址，数据被逐列按时钟输入 (clockinto) I/O 电路。在绕回的情况下，当到达物理列的结尾时发出第二起始地址。

[0243] 图 23D 图示了每页的起始列地址可以被存储在该页中预留用于系统使用的部分中。例如，页 630 的起始列地址被存储在存储器阵列的部分 634 中。

[0244] 在另一优选实施例中，通过诸如图 23A 所示的伪随机生成器来提供每页一个的起始物理列地址的序列。

[0245] 偏移每页的起始位置可以避免不期望的数据样式在 NAND 链内排队，且有助于缓解在编程期间的升压问题。

[0246] 除了缓解在编程期间的沟道升压问题以外，加扰在一页内的数据还有助于控制在感测期间的源极载入错误。

[0247] 由在源极线和芯片的地垫之间的有限电阻引入了源极载入错误。感测存储器单元的一个潜在的问题是由跨越有限电阻的源极载入而引起的源极线偏压。当并行地感测大量存储器单元时，其组合的电流可以导致在具有有限电阻的地回路中的显著电压降。这导致源极线偏压，其引起在使用阈值电压感测的读操作中的错误。

[0248] 图 28A 图示由于在具有对地的有限电阻的源极线中的电流流动引起的源极电压错误的问题。读 / 写电路 370A 和 370B 同时在一页存储器单元上操作。在读 / 写电路中的每个感测模块 480 经由位线 36 耦合到相应的单元。根据图 8 所示的读 / 写堆栈 400，每个感测模块 480 包括连接到位线之一的感测放大器 212、一组数据锁存器 430 和共享公共处理器 500 和 I/O 电路 440。对于并行操作，将存在用于一页中的每个存储器单元的感测模块。

[0249] 例如，感测模块 480 感测存储器单元 10 的导电电流 i_1（源极 - 漏极电流）。该导电电流从感测模块经过位线 36 流入存储器单元 10 的漏极，且在经过源极线 34 流到地之前从源极 14 流出。在集成电路芯片中，在存储器阵列中的单元的源极被联系在一起，作为连接到存储器芯片的某外部地基垫 (groundpad，例如 Vss 基垫）的源极线 34 的多个分支。即使当使用金属带来降低源极线的电阻时，有限电阻 R 仍然在存储器单元的源极电极和地基垫之间。典型地，地回路电阻 R 是大约 10 欧姆。

[0250] 对于正被并行感测的存储器的整个页来说，流过源极线 34 的总电流是所有导电电流的总和，即 $i_{TOT} = i_1+i_2+...+i_p$。通常，每个存储器单元具有依赖于被编程到其电荷存储元件中的电荷量的导电电流。对于存储器单元的给定控制栅极电压，小的电荷将产生相对较高的导电电流（见图 4）。当在存储器单元的源极电极和地基垫之间存在有限电阻时，跨越该电阻的电压降由 $V_{drop} = i_{TOT}R$ 给出。

[0251] 例如，如果 64000 个位线同时放电，每个具有 1μA 的电流，则源极线电压降将等于 64000 线 ×1μA/ 线 ×10 欧姆～ 0.64 伏特。假设体效应 (bodyeffect) 使得在源极电压中 0.64V 的升压导致阈值电压中 0.96V 的升压，则当感测存储器单元的阈值电压时，该源极线偏压将促成 0.96 伏特的感测误差。

[0252] 图 28B 图示了由源极线电压降引起的存储器单元的阈值电压电平中的误差。被供应给存储器单元 10 的控制栅极 30 的阈值电压 V_T 与 GND 有关。但是，由存储器单元看到的有效的 V_T 是在其控制栅极 30 和源极 14 之间的电压差。在供应的 V_T 和有效的 V_T 之间存在大约 $1.5\times V_{drop}$ 的差（忽略从源极 14 到源极线的电压降的较小影响）。当感测存储器单元

的阈值电压时，该 V_{drop} 或源极线偏压将促成例如 0.96 伏特的感测误差。

[0253] 该偏压不能轻易地被移除，因为它是独立于数据的，即独立于该页的存储器单元的存储器状态。该偏压在一种极端情况下最高：即当该页的所有存储器单元都处于擦除状态时。在该情况下，每个单元都是高度导电的，促成大的 V_{drop} 及由此的高源极线偏压。另一方面，在另一极端下，当在该页中的所有存储器单元都在最大编程状态 (most programmed state) 时，则每个单元是不导电的，导致最小或无源极线偏压。

[0254] 根据本发明的另一方面，页中的数据位被随机化，以便平均而言该页包含具有擦除和编程状态的相等的单元混合。以此方式，源极线偏压或载入实质上不改变，且可以允许在感测操作期间的适当调整。

[0255] 这通过随机化一页内的各个位来实现。优选地，使用每位指定特定极性的伪随机位的序列来编码该页内的位。在一个实施例中，对于页中的每个数据位存在一个极性位。在另一实施例中，对于页中的每字节数据存在一个极性位。在此实施例中，如果极性位指定了位的翻转，则在该数据字节内的所有位将被翻转。

[0256] 图 29 图示了随机化一页内的位的方法。

[0257] 步骤 750：提供具有可逐页访问的非易失性存储器单元的阵列的集成电路存储器芯片，每页是一群存储器单元，该群的每个存储器单元位于该阵列的列中，且沿着可由公共字线访问的行。

[0258] 步骤 760：提供用于要被编程的页的每组数据位的第一和第二编码。

[0259] 步骤 762：在芯片上生成极性位的序列，每组数据位一个极性位。

[0260] 步骤 764：根据每组数据位的极性位是处于第一还是第二状态，用第一或第二编码来编码每组数据位。

[0261] 步骤 770：并行地将所有编码的数据位的集合编程到页中。

[0262] 优选地通过诸如图 23A 所示的伪随机生成器来提供每组数据位一个的极性位的序列。每组数据位包含预定数量的位。例如，在一个实施例中，位的预定数量是 1。在另一实施例中，位的预定数量是 8 位。

[0263] 一页内和各页之间的数据的芯片上伪随机化

[0264] 对于一些极其规则的数据样式、比如所有都具有擦除状态的页，在一页内加扰的方案是不够的。

[0265] 根据本发明的另一方面，一页内的随机化与各页间的随机化相结合。具体地，给定了芯片上电路的有限资源，优选地通过具有独立的起始位置的每页来完成一页内的随机化，且优选地通过具有独立的编码极性的每页来完成各页间随机化。

[0266] 图 30 图示了根据本发明的另一方面的逐页和在每页内随机化数据的方法。

[0267] 步骤 800：提供具有可逐页访问的非易失性存储器单元的阵列的集成电路存储器芯片，每页是一群存储器单元，该群的每个存储器单元位于该阵列的列中，且沿着可由公共字线访问的行。

[0268] 步骤 810：提供用于要被编程的每页数据的第一和第二编码。

[0269] 步骤 812：在芯片上生成极性位的序列，每页一个极性位。

[0270] 步骤 814：根据该页的极性位是处于第一还是第二状态，用第一或第二编码来编码该页数据。

[0271] 步骤 820 :在芯片上生成起始列位置的序列,以便要被编程的每页具有相关的起始列位置。

[0272] 步骤 822 :通过从相关起始列位置开始且绕回直到该页被填满,将要被编程的数据安排到每页中。

[0273] 步骤 830 :并行将安排的数据编程到每页中。

[0274] 在另一实施例中,还通过在步骤 810 到步骤 822 之间插入图 29 所示的步骤 760、步骤 762 和步骤 764,来完成一页内的数据位随机化。在此实施例中,在字线方向上的位和在列方向上的位都被随机化。

[0275] 在此引用的所有专利、专利申请、文章、书籍、规范、其他公开、文档和事物为了一切目的而通过此全部被引用附于此。对于在任何并入的公开物、文档、事物与本发明的文本之间的术语的定义或使用中的任何矛盾或冲突的程度,在本文档中的术语的定义和使用应该是优先的。

[0276] 虽然已经描述了本发明的实施例,应该理解,本发明不局限于这些例示的实施例,而是由所附权利要求来定义。

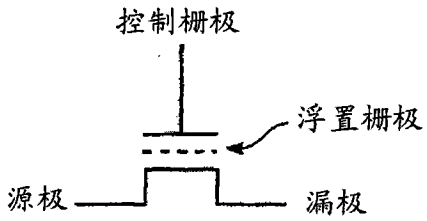


图 1A

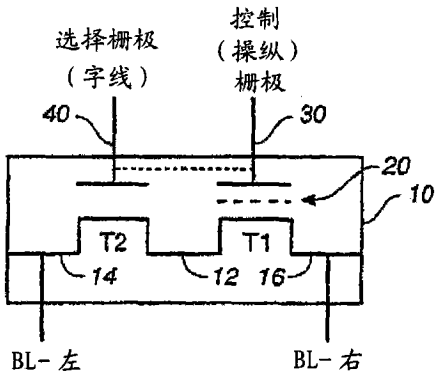


图 1B

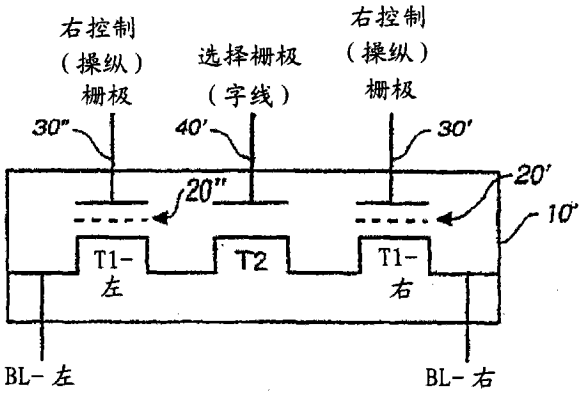


图 1C

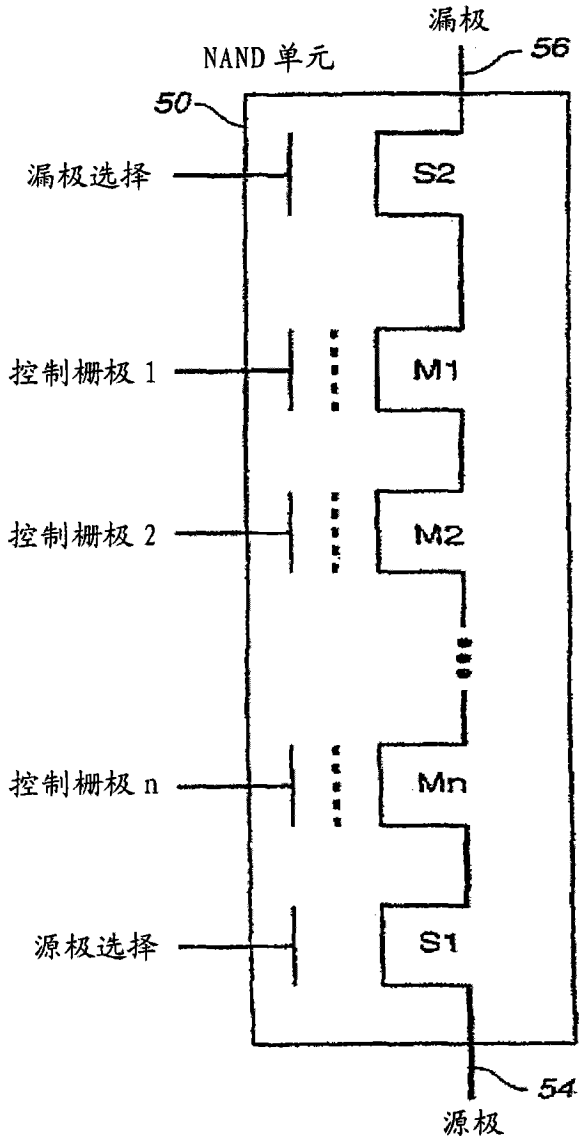


图 1D

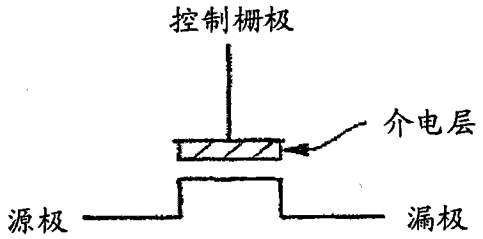


图 1E

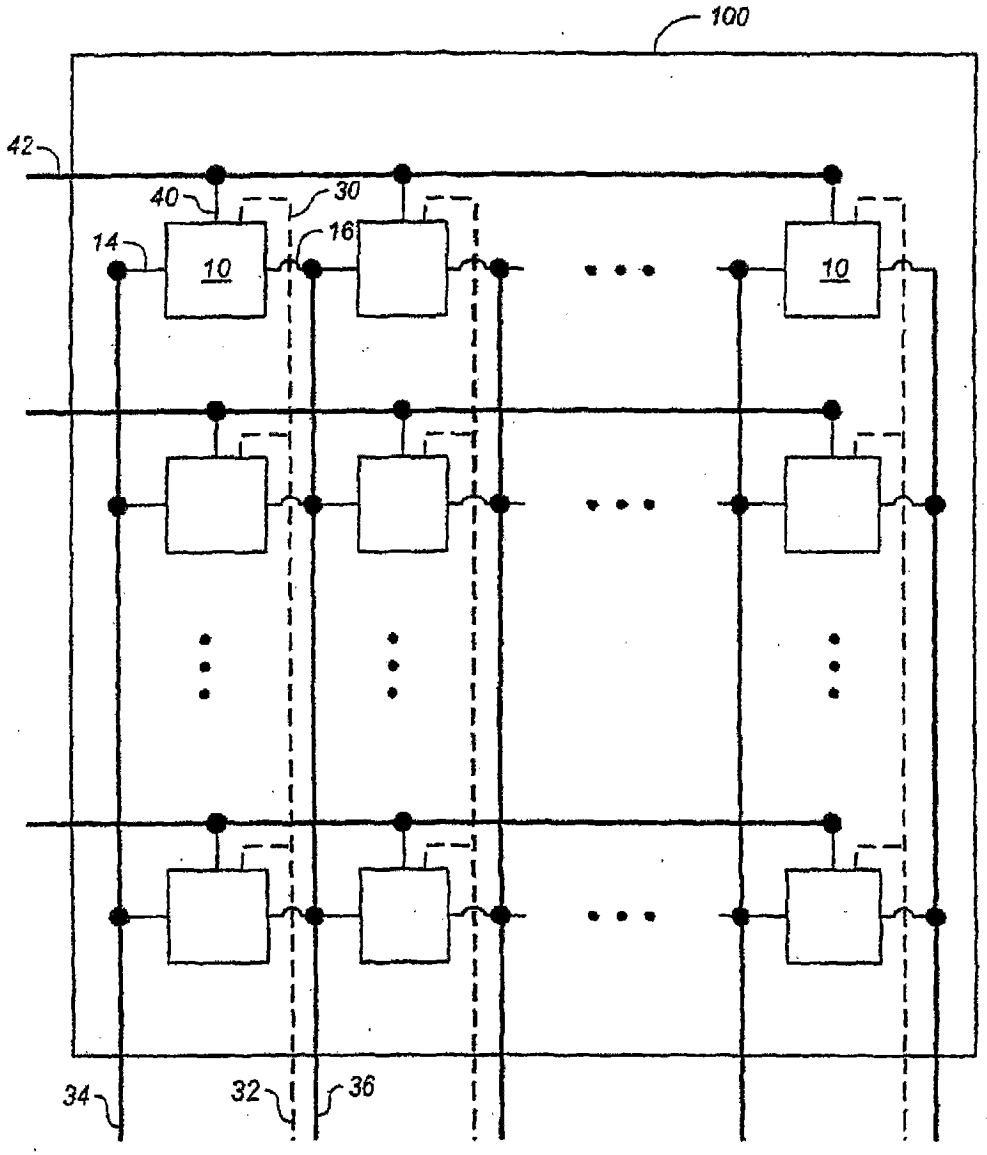


图 2

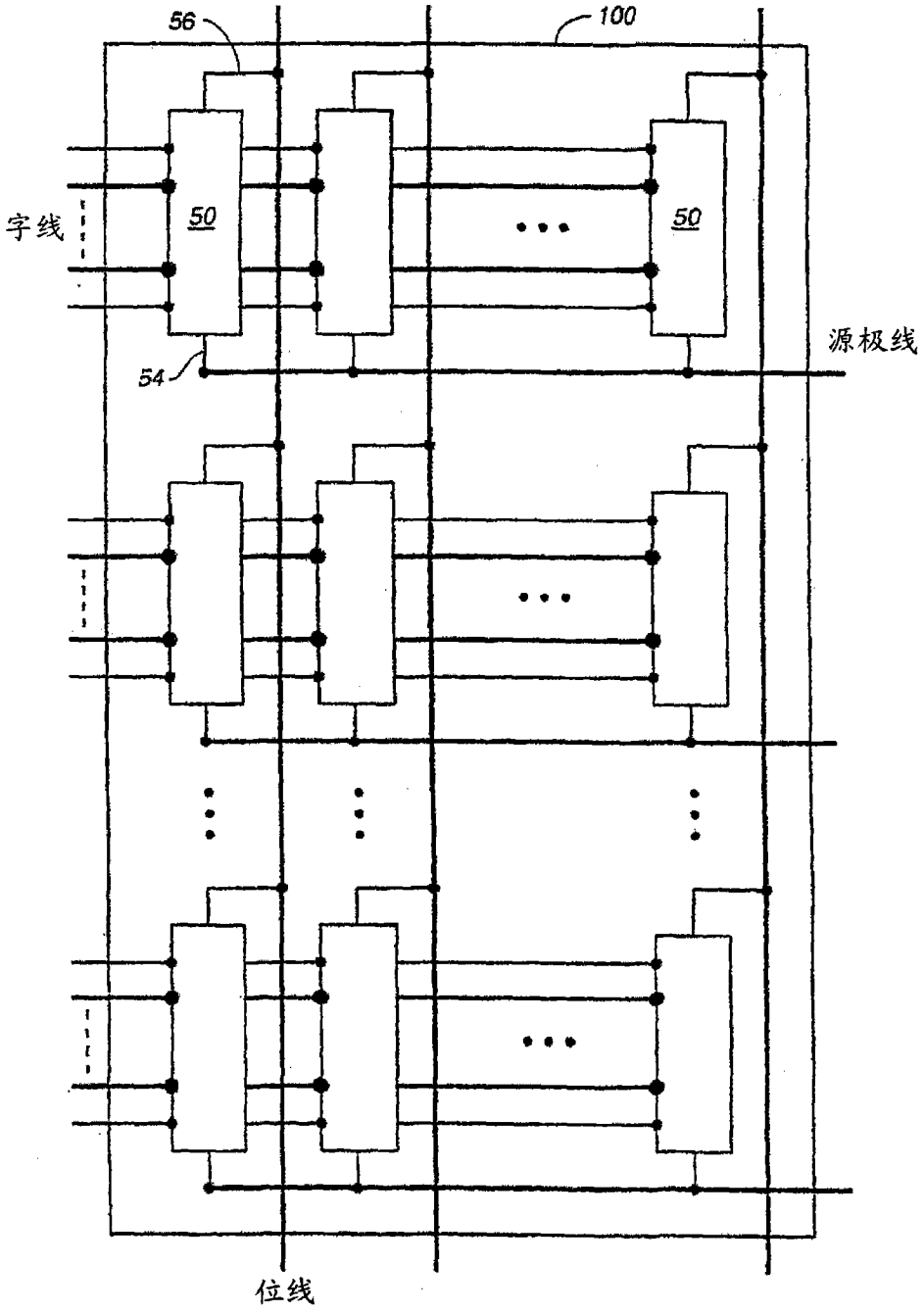


图 3

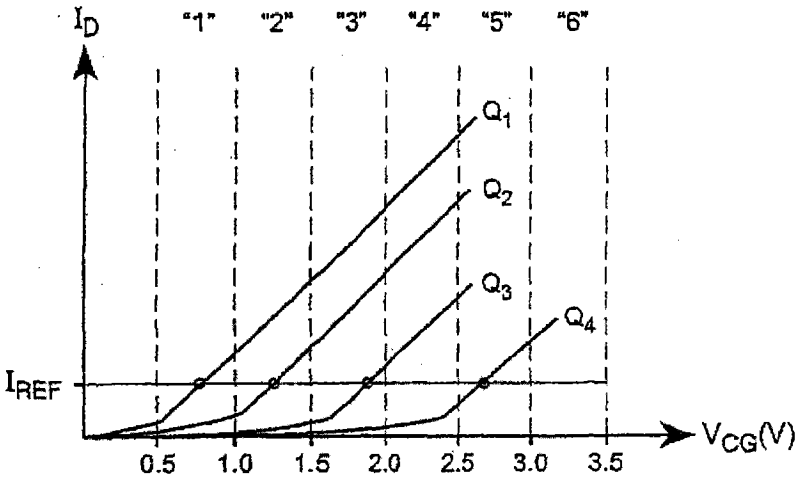

I_D
"1"
"2"
"3"
"4"
"5"
"6"
Q_1
Q_2
Q_3
Q_4
I_{REF}
$V_{CG}(V)$
0.5
1.0
1.5
2.0
2.5
3.0
3.5

图 4

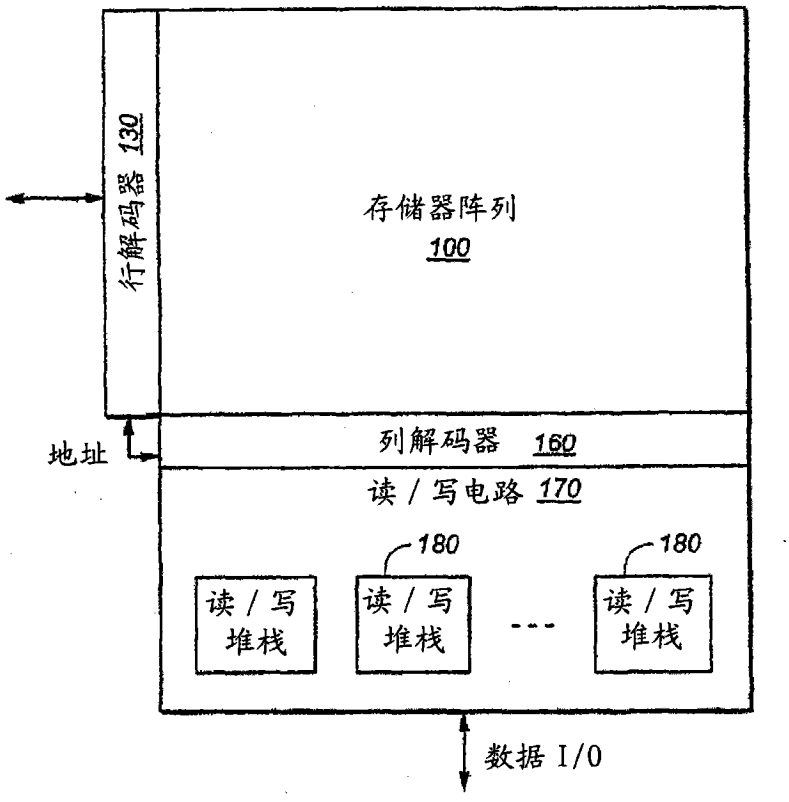
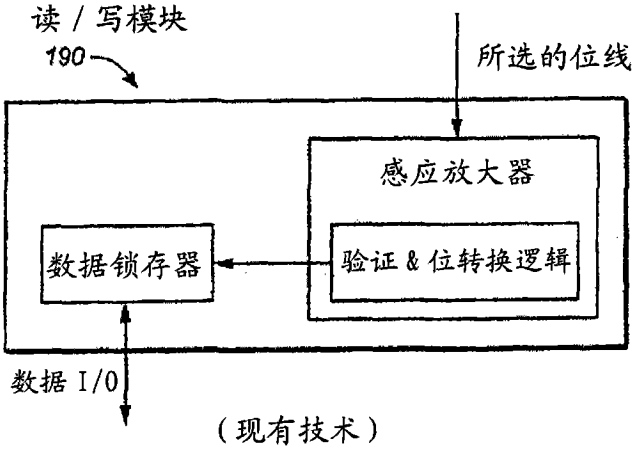
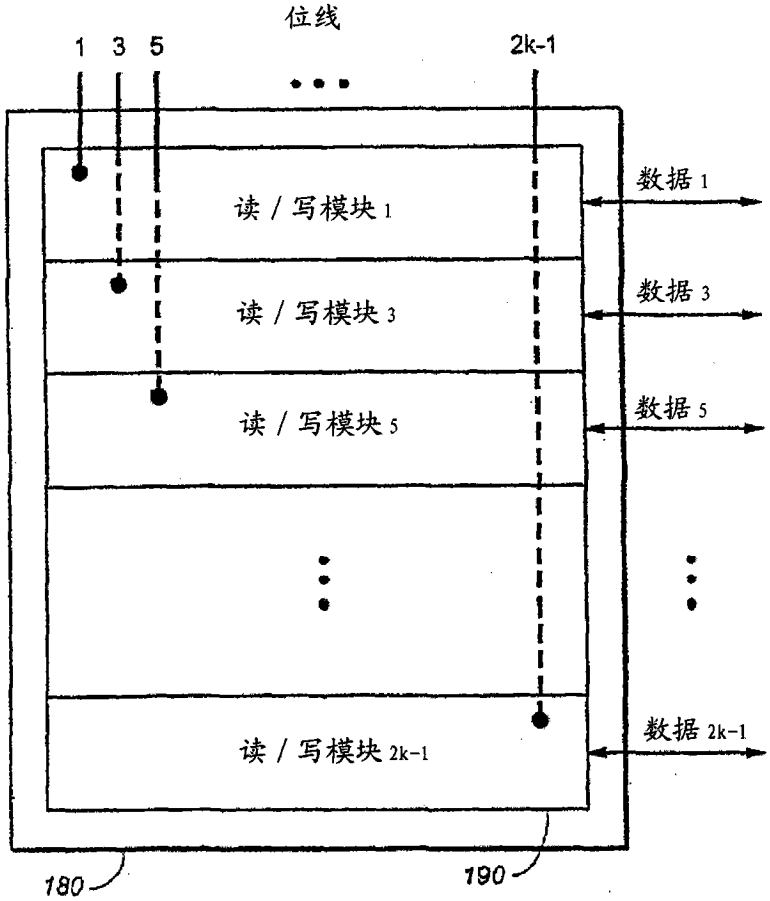

行解码器 130
存储器阵列
100
地址
列解码器 160
读 / 写电路 170
180
180
读 / 写
堆栈
读 / 写
堆栈
读 / 写
堆栈
数据 I/O

图 5



（现有技术）

图 6A



（现有技术）

图 6B

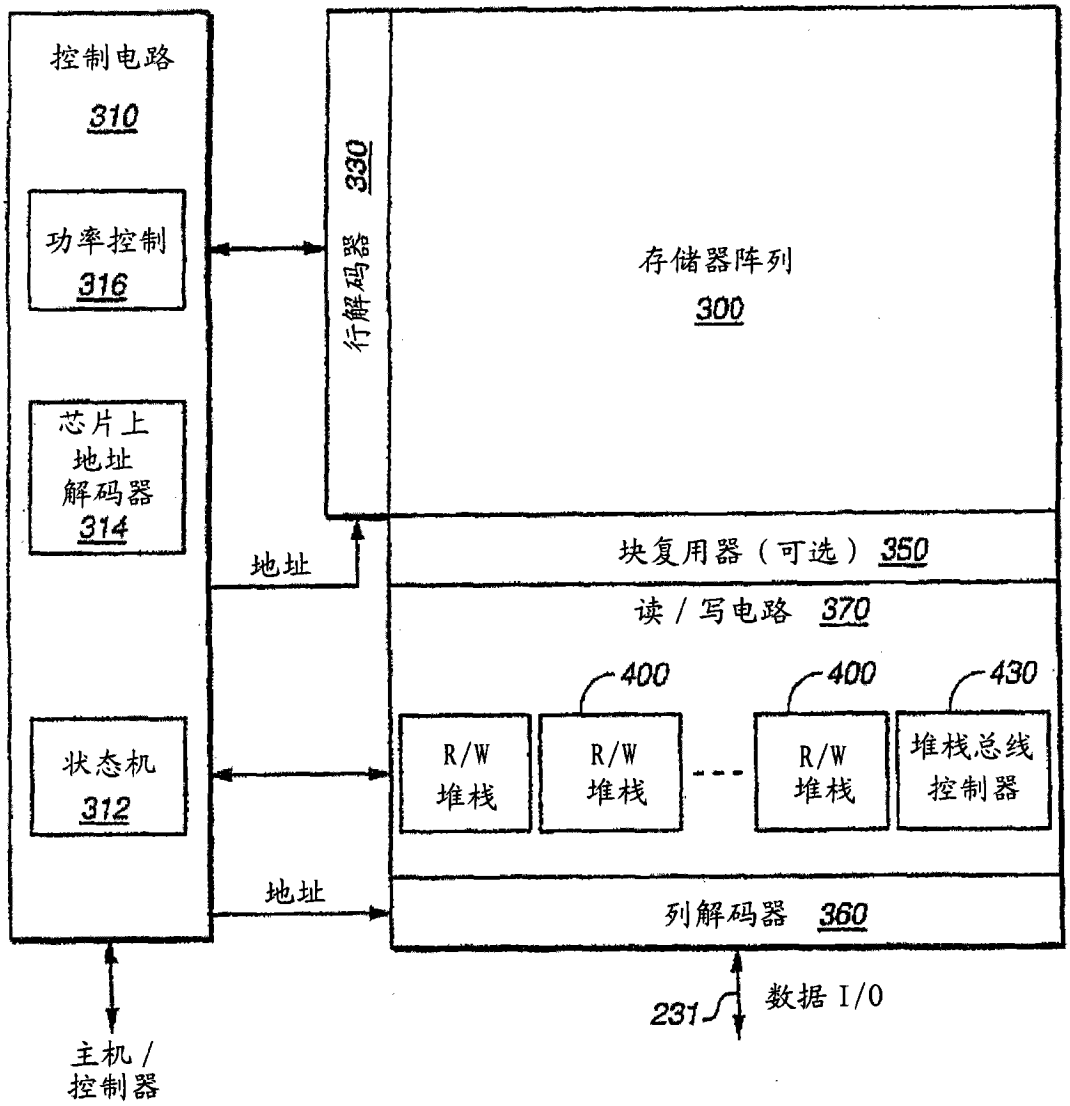

控制电路
310
功率控制
316
芯片上
地址
解码器
314
状态机
312
主机 /
控制器
行解码器 330
存储器阵列
300
地址
块复用器（可选） 350
读 / 写电路 370
400
400
430
R/W
堆栈
R/W
堆栈
R/W
堆栈
堆栈总线
控制器
地址
列解码器 360
231
数据 I/O

图 7A

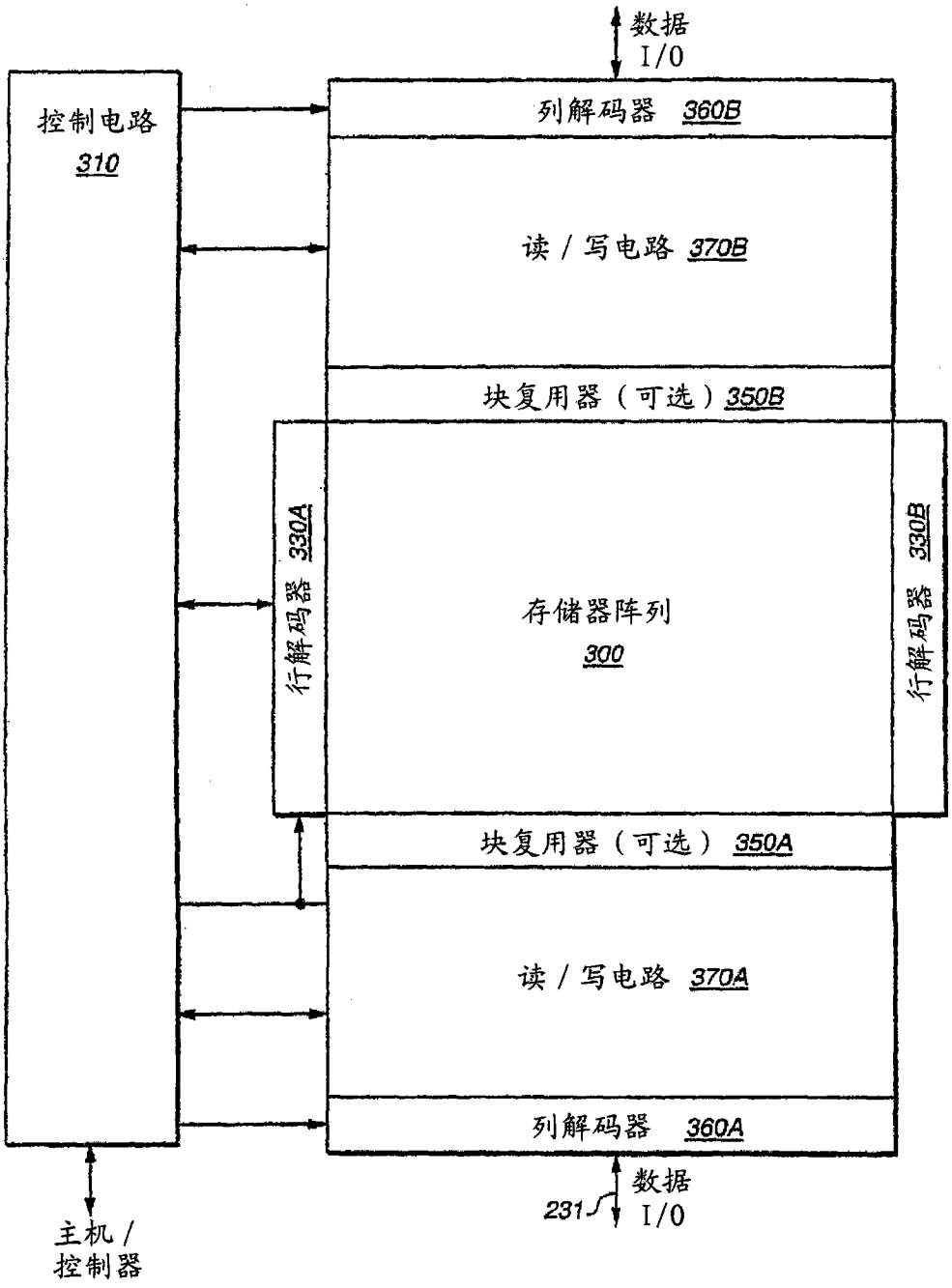

数据
I/O
控制电路
310
列解码器 360B
读 / 写电路 370B
块复用器（可选）350B
行解码器 330A
存储器阵列
300
行解码器 330B
块复用器（可选） 350A
读 / 写电路 370A
列解码器 360A
数据
I/O
231
主机 /
控制器

图 7B

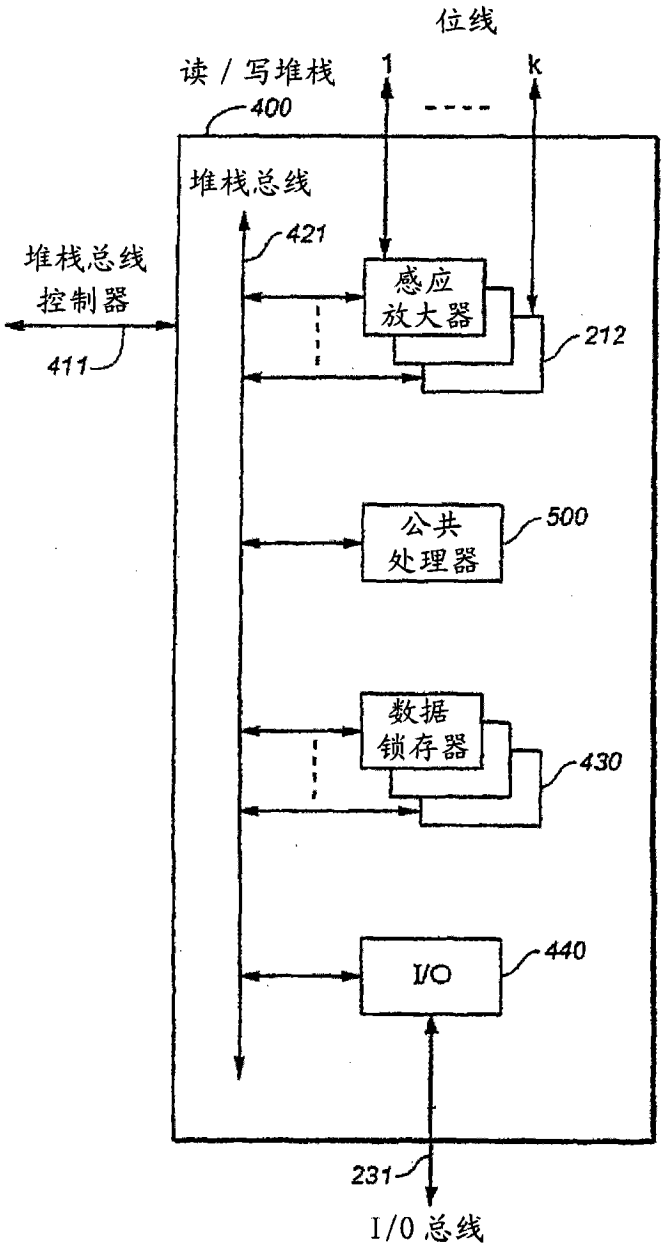

位线
读 / 写堆栈
1
k
400
堆栈总线
421
堆栈总线
控制器
411
感应
放大器
212
公共
处理器
500
数据
锁存器
430
I/O
440
231
I/O 总线

图 8

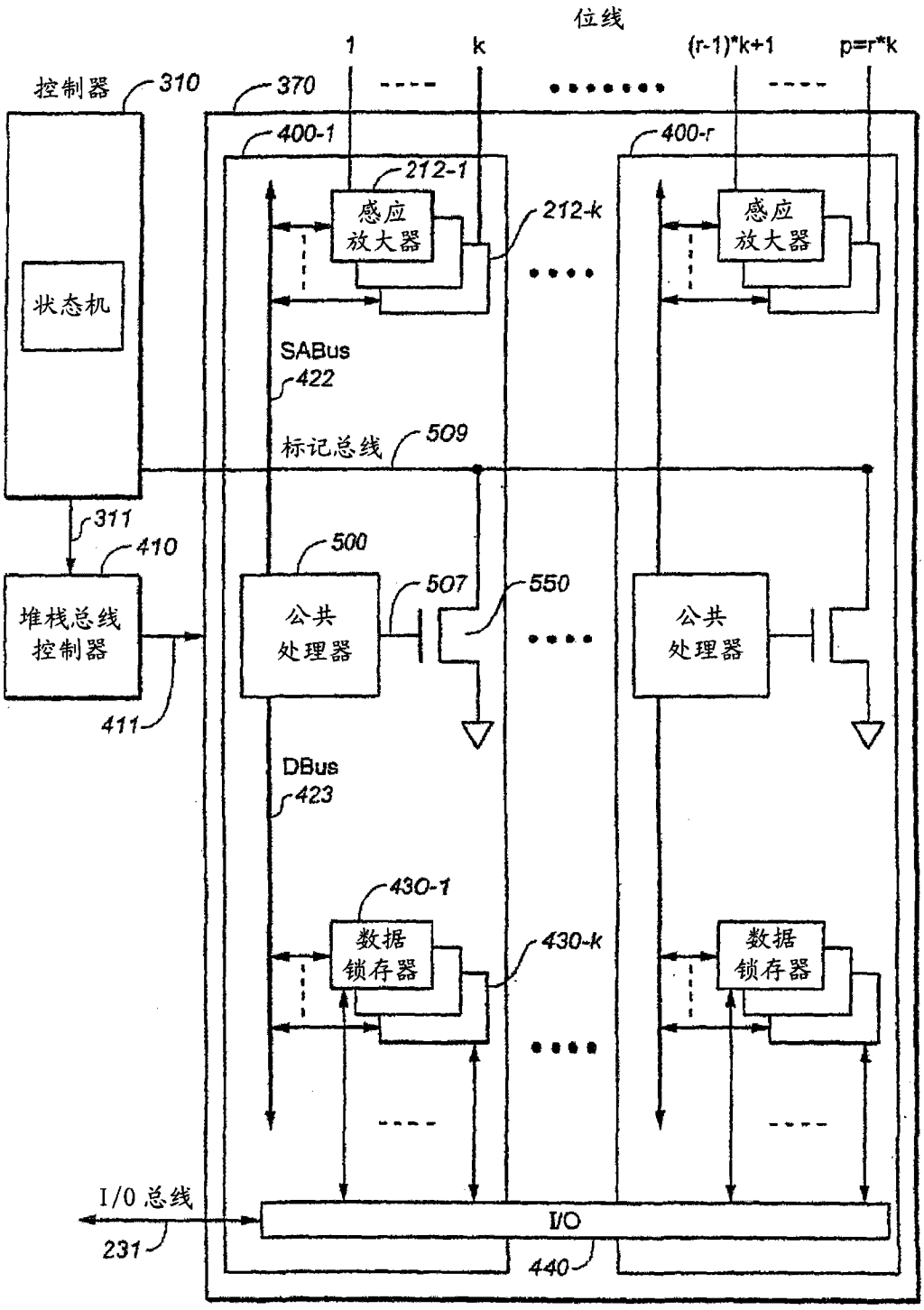

位线
1
k
(r-1)*k+1
p=r*k
控制器
310
370
400-1
400-r
212-1
212-k
感应
放大器
感应
放大器
状态机
SABus
422
标记总线
509
311
410
500
507
550
堆栈总线
控制器
公共
处理器
公共
处理器
411
DBus
423
430-1
430-k
数据
锁存器
数据
锁存器
I/O 总线
231
I/O
440

图 9

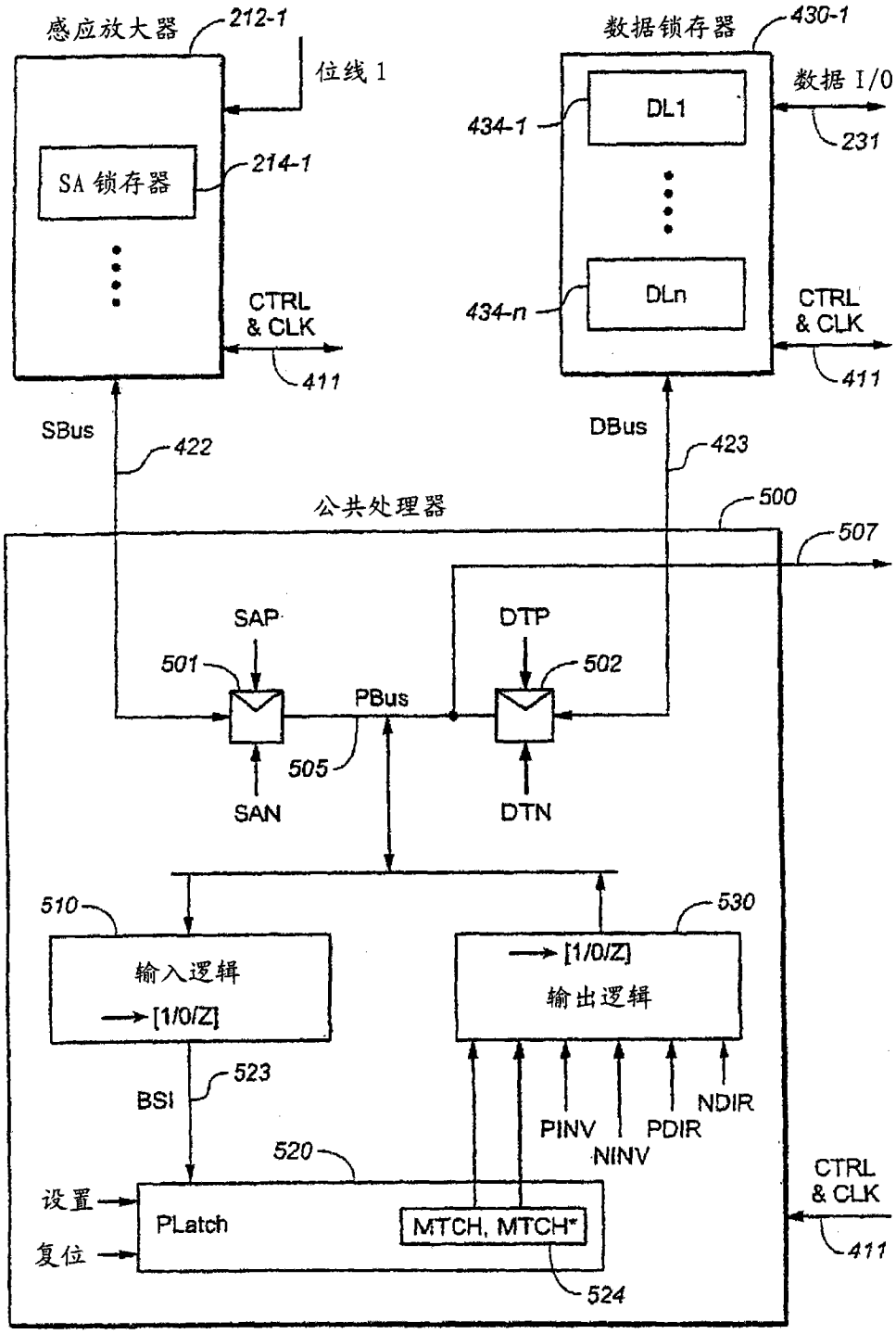

感应放大器
212-1
位线 1
SA 锁存器
214-1
CTRL
& CLK
411
SBus
422
数据锁存器
430-1
DL1
434-1
数据 I/O
231
DLn
434-n
CTRL
& CLK
411
DBus
423
公共处理器
500
507
SAP
501
PBus
505
SAN
DTP
502
DTN
510
输入逻辑
[1/0/Z]
BSI
523
530
[1/0/Z]
输出逻辑
PINV
NINV
PDIR
NDIR
520
设置
复位
PLatch
MTCH, MTCH*
524
CTRL
& CLK
411

图 10

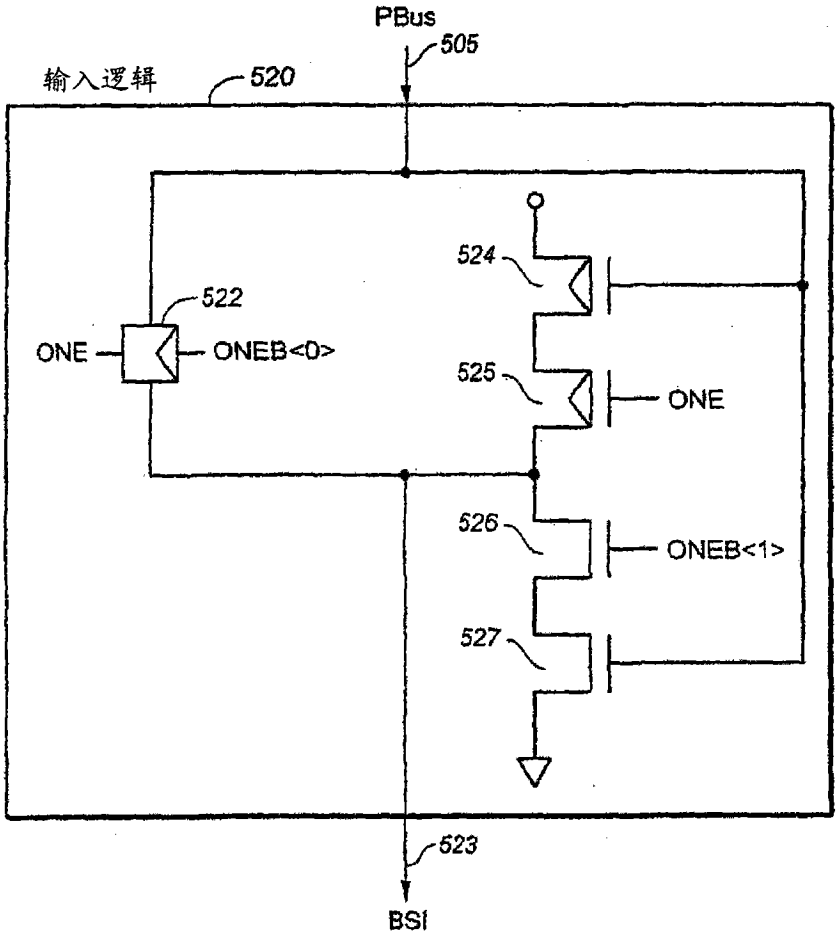


图 11A

输入逻辑真值表

转移模式	ONE	ONEB<0>	ONEB<1>	PBus（输入）	BSI（输出）
通过	1	0	0	PBus	PBus
反转	0	1	1	PBus	PBus*
浮置	1	1	0	PBus	浮置

图 11B

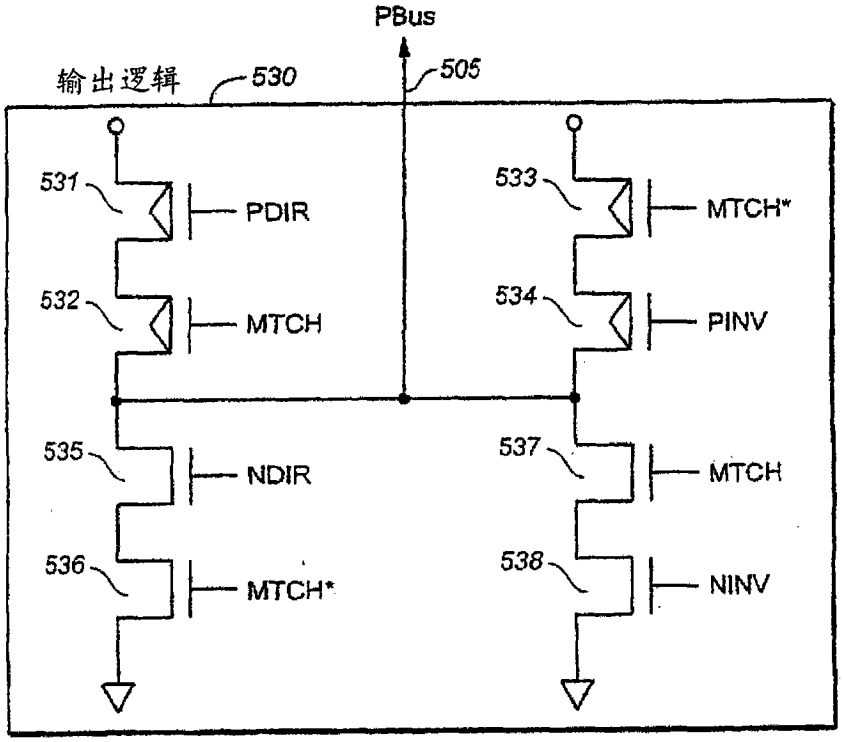


图 12A

输出逻辑真值表

转移模式	PINV	NINV	PDIR	NDIR	MTCH	PBus（输出）
通过	D	D	D	1	0	0
	0	D	D	D	1	1
反转	D	D	0	D	0	1
	D	1	D	D	1	0
浮置	D	D	D	D	X	Z
预充电	0	D	0	D	X	1

（缺省值：PINV=1、NINV=0，PDIR=1，NDIR=0）

图 12B

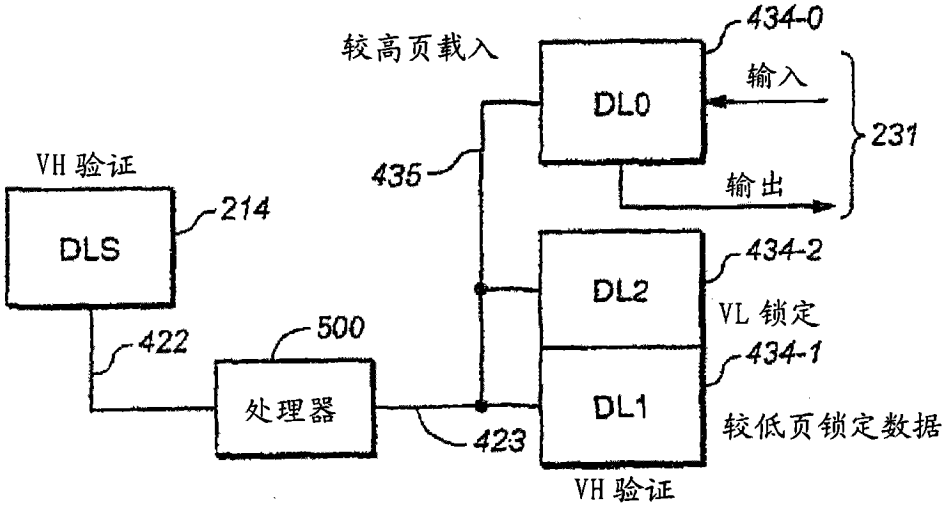

较高页载入
434-0
输入
DL0
231
输出
435
VH 验证
DLS
214
434-2
DL2
VL 锁定
422
500
434-1
处理器
DL1
较低页锁定数据
423
VH 验证

图 13

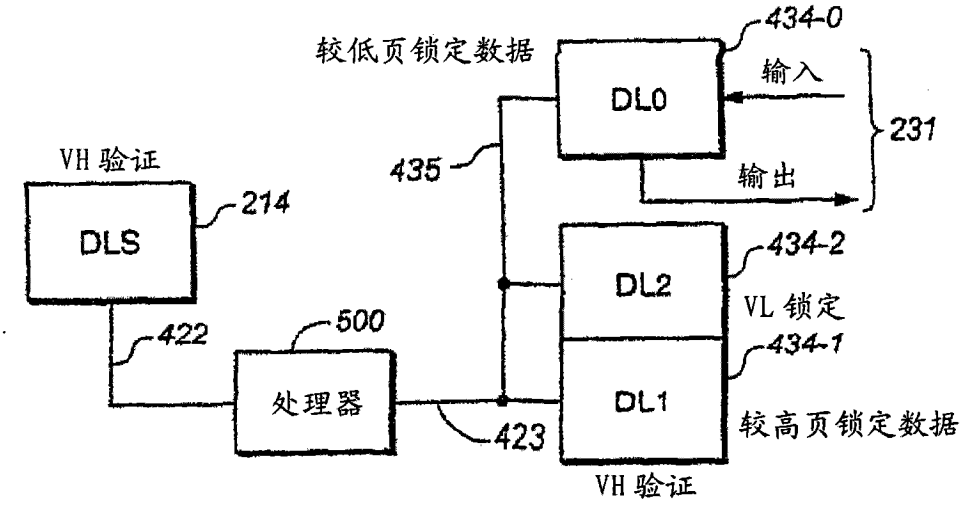

434-0
较低页锁定数据
输入
DL0
231
435
输出
VH 验证
DLS
214
434-2
DL2
VL 锁定
422
500
434-1
处理器
DL1
较高页锁定数据
423
VH 验证

图 14

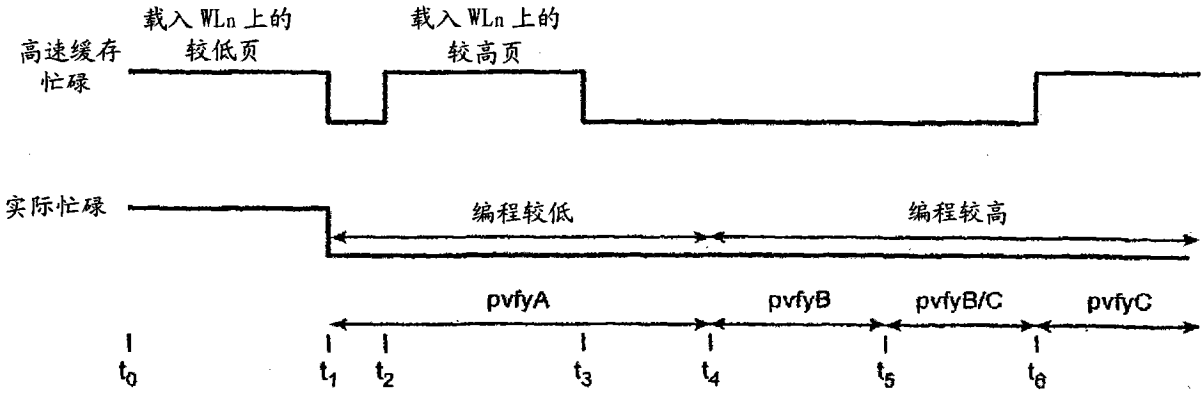

载入 WLn 上的
较低页
载入 WLn 上的
较高页
高速缓存
忙碌
实际忙碌
编程较低
编程较高
pvfyA
pvfyB
pvfyB/C
pvfyC
t0
t1
t2
t3
t4
t5
t6

图 15

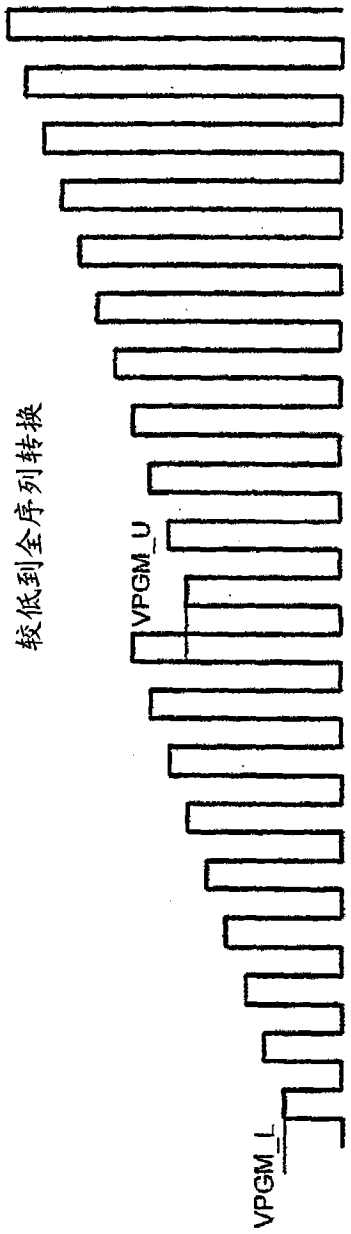

较低到全序列转换
VPGM_U
VPGM_L

图 16

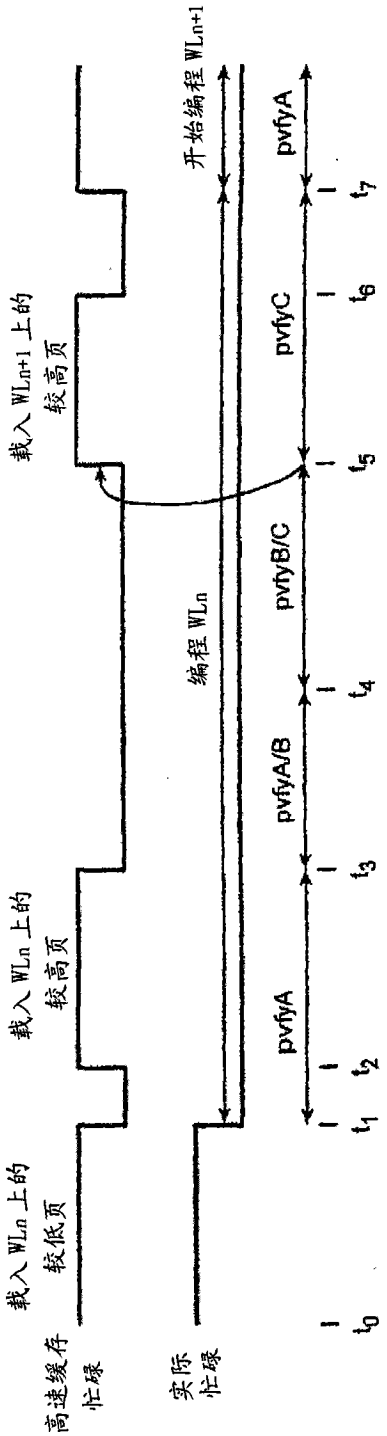

载入 WLn 上的 较低页
载入 WLn 上的 较高页
载入 WLn+1 上的 较高页
开始编程 WLn+1
编程 WLn
pvfyA
pvfyA/B
pvfyB/C
pvfyC
pvfyA
t0
t1
t2
t3
t4
t5
t6
t7
高速缓存忙碌
实际忙碌

图 17

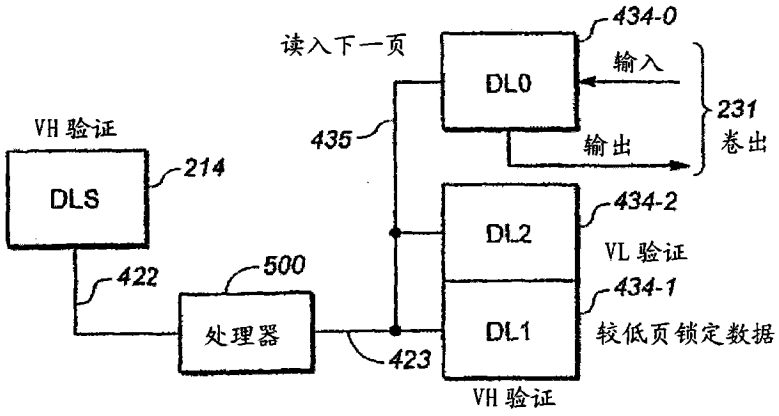

读入下一页
434-0
DL0
输入
231
卷出
输出
435
VH 验证
214
DLS
434-2
DL2
VL 验证
500
422
处理器
434-1
DL1
较低页锁定数据
423
VH 验证

图 18

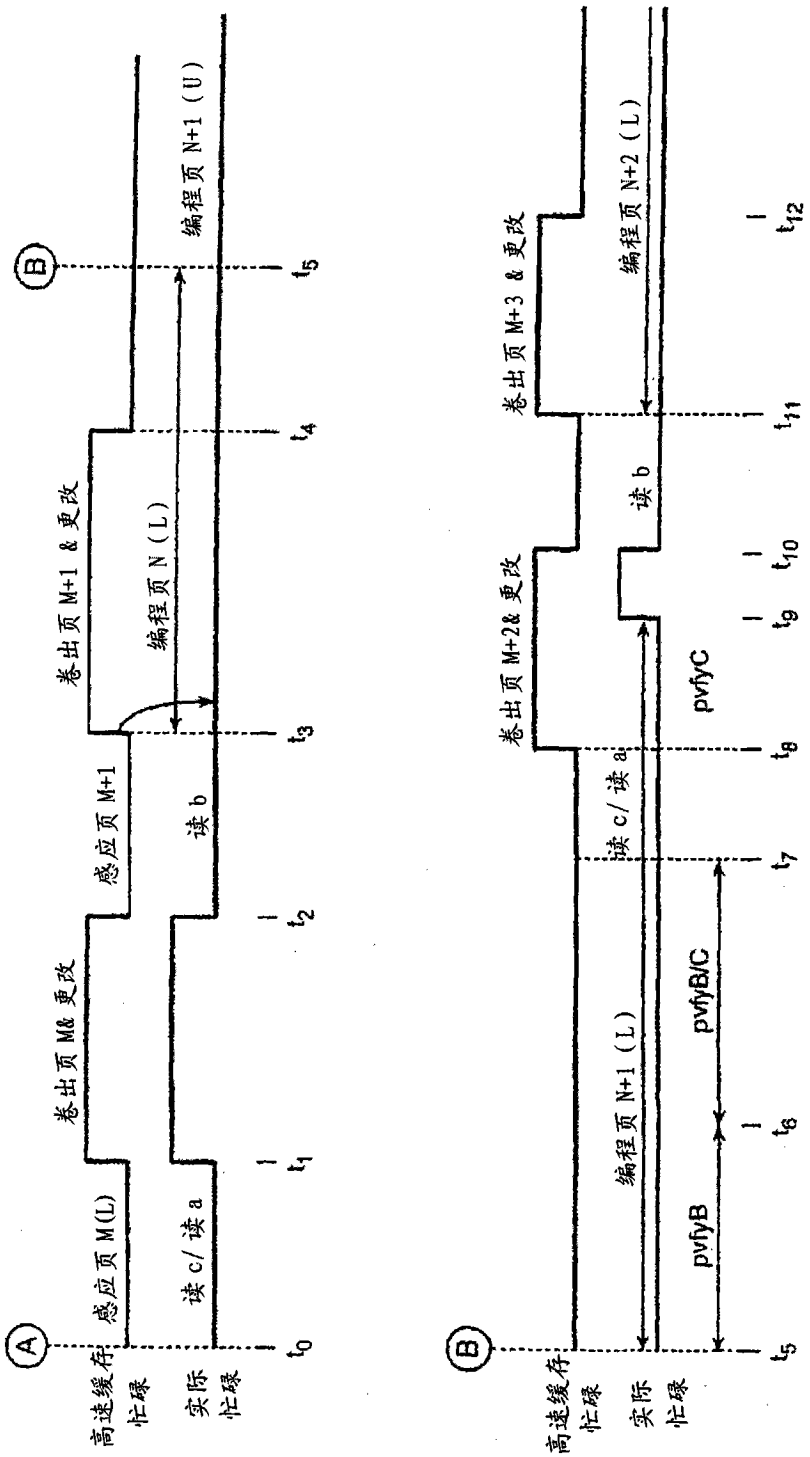

A
高速缓存忙碌
实际忙碌
感应页 M(L)
读 c/ 读 a
卷出页 M& 更改
感应页 M+1
读 b
编程页 N (L)
卷出页 M+1 & 更改
编程页 N+1 (U)
B
t0
t1
t2
t3
t4
t5
B
高速缓存忙碌
实际忙碌
编程页 N+1 (L)
pvfyB
pvfyB/C
读 c/ 读 a
卷出页 M+2& 更改
pvfyC
读 b
卷出页 M+3 & 更改
编程页 N+2 (L)
t5
t6
t7
t8
t9
t10
t11
t12

图 19A

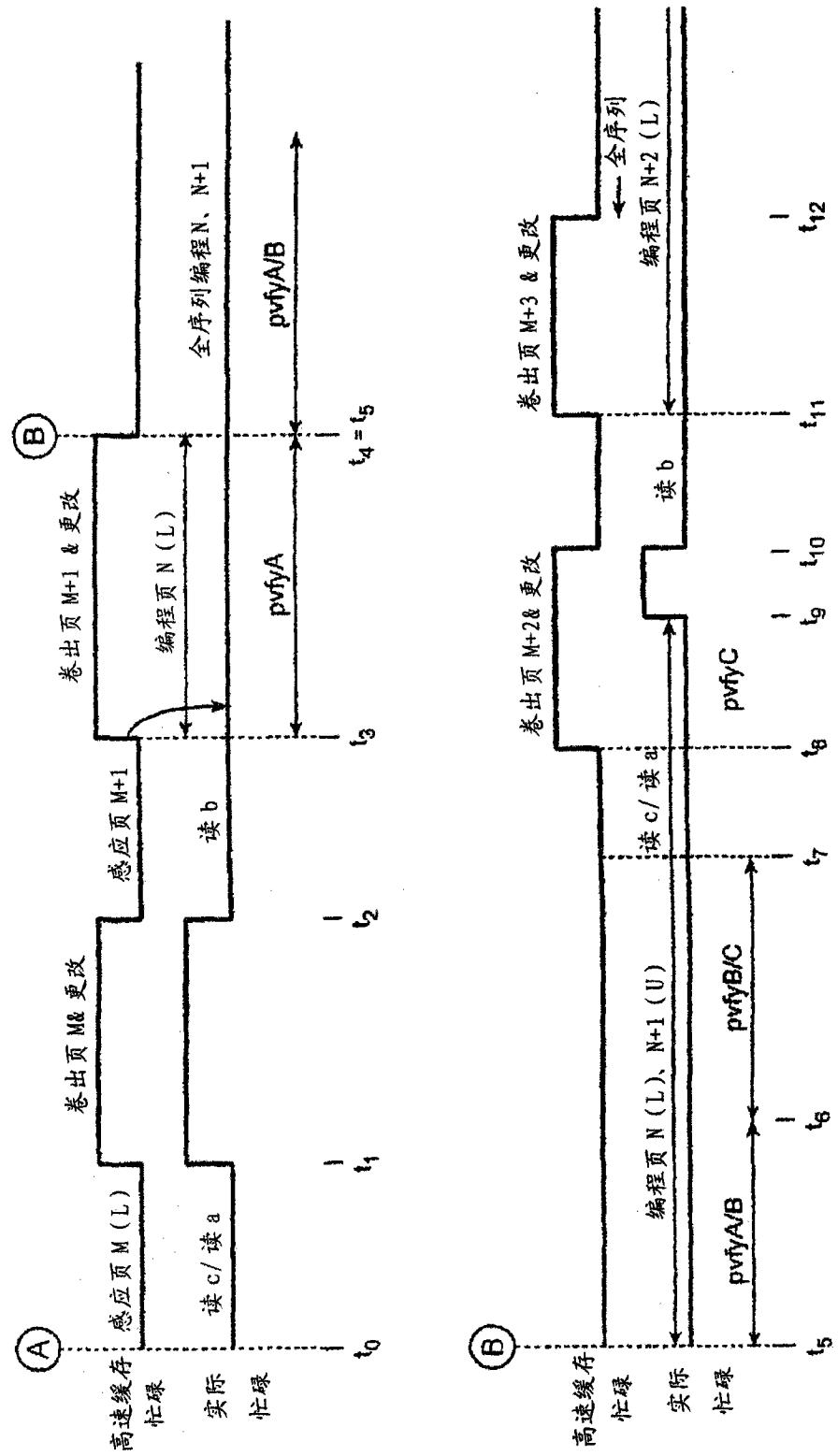
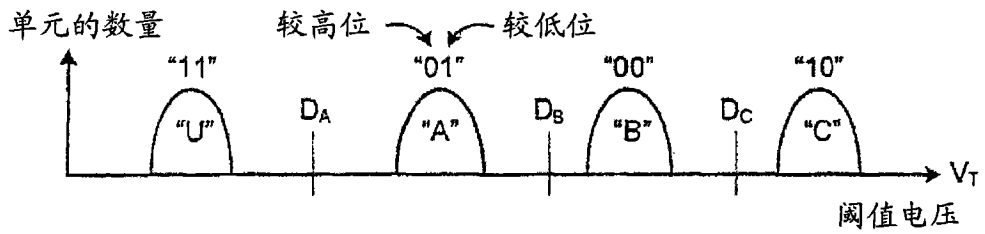

高速缓存忙碌
实际忙碌
A
感应页 M (L)
卷出页 M& 更改
读 c/ 读 a
感应页 M+1
读 b
卷出页 M+1 & 更改
编程页 N (L)
全序列编程 N、N+1
pvfyA
pvfyA/B
t0
t1
t2
t3
t4 = t5
B
高速缓存忙碌
实际忙碌
B
卷出页 M+2& 更改
编程页 N (L)、N+1 (U)
读 c/ 读 a
卷出页 M+3 & 更改
读 b
全序列
编程页 N+2 (L)
pvfyA/B
pvfyB/C
pvfyC
t5
t6
t7
t8
t9
t10
t11
t12

图 19B



多状态存储器（LM格雷码）

图 20

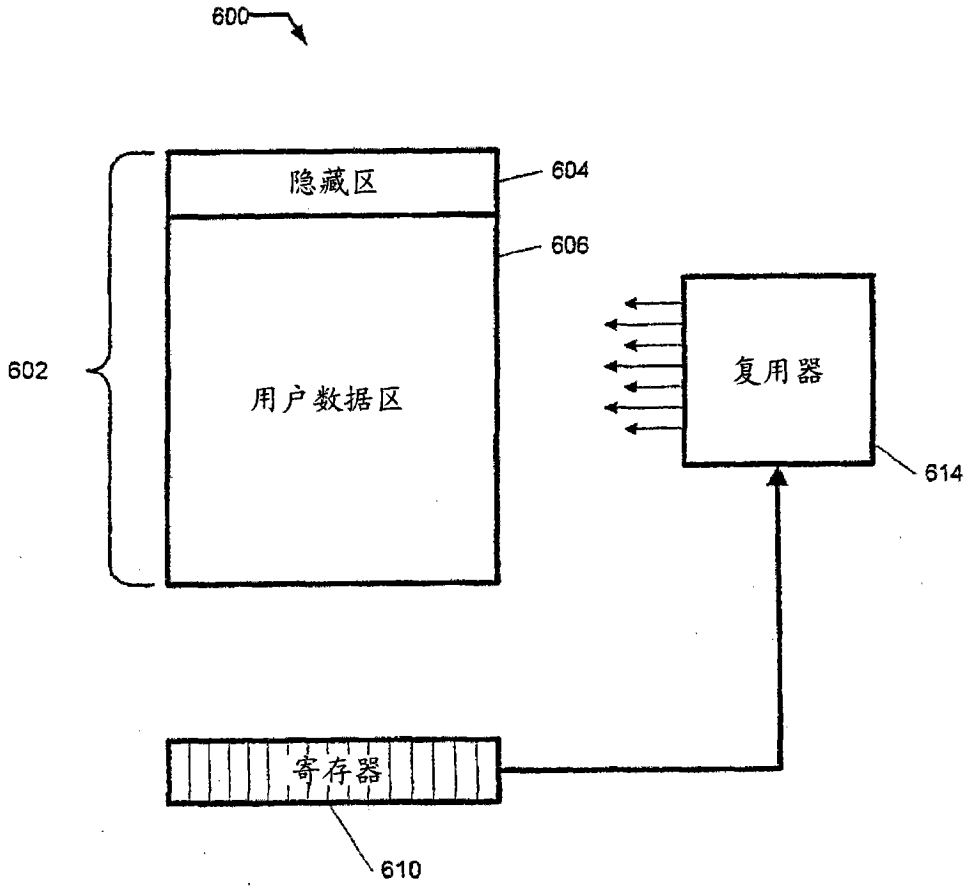


图 21

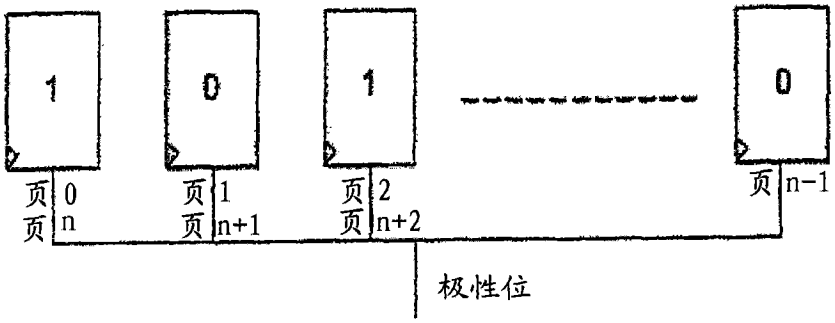


图 22A

寄存器位置	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
码（极性位）	1	0	1	0	0	1	1	1	1	0	0	0	1	1	0	1	0
UD 原始编码	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
UD 随后编码	1	1	1	1	0	0	1	0	1	1	0	1	1	0	0	0	0
状态	ER		ER		B		C		ER		A		C		B		

图 22B

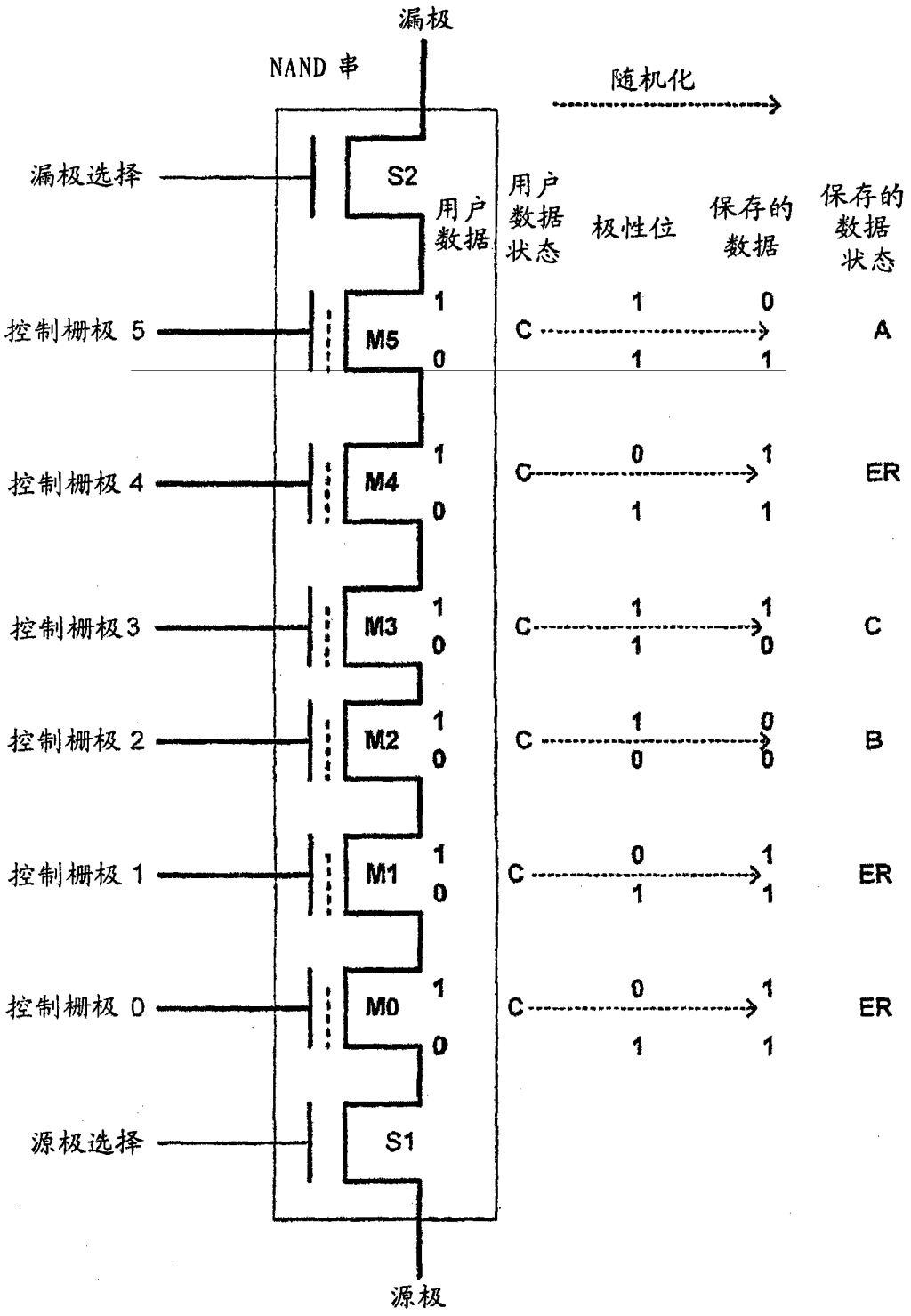

漏极
NAND 串
随机化
漏极选择
S2
用户数据
用户数据状态
极性位
保存的数据
保存的数据状态
控制栅极 5
M5
C
A
控制栅极 4
M4
ER
控制栅极 3
M3
C
控制栅极 2
M2
B
控制栅极 1
M1
ER
控制栅极 0
M0
ER
源极选择
S1
源极

图 22C

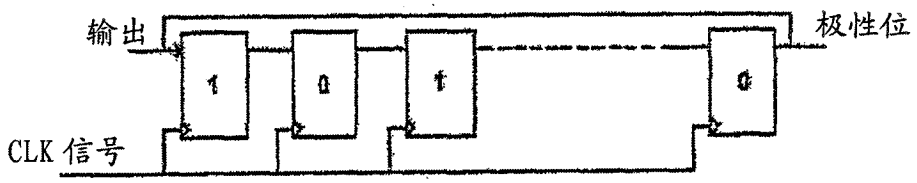


图 23A

图 23B

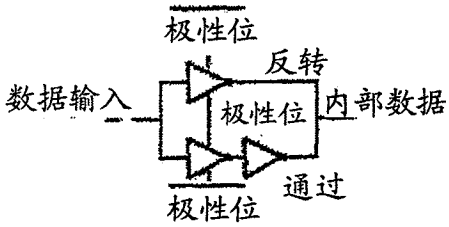


图 23C

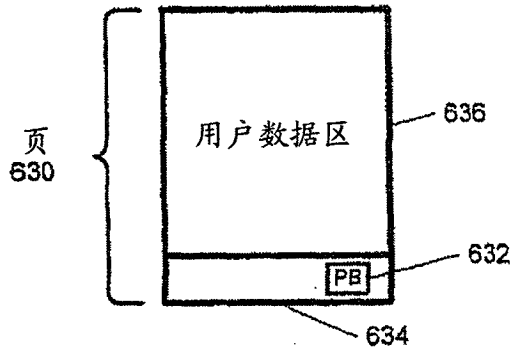


图 23D

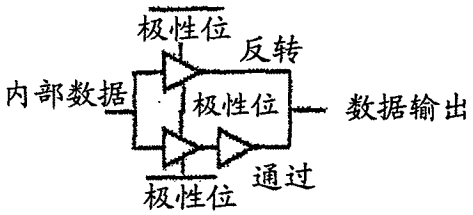


图 23E

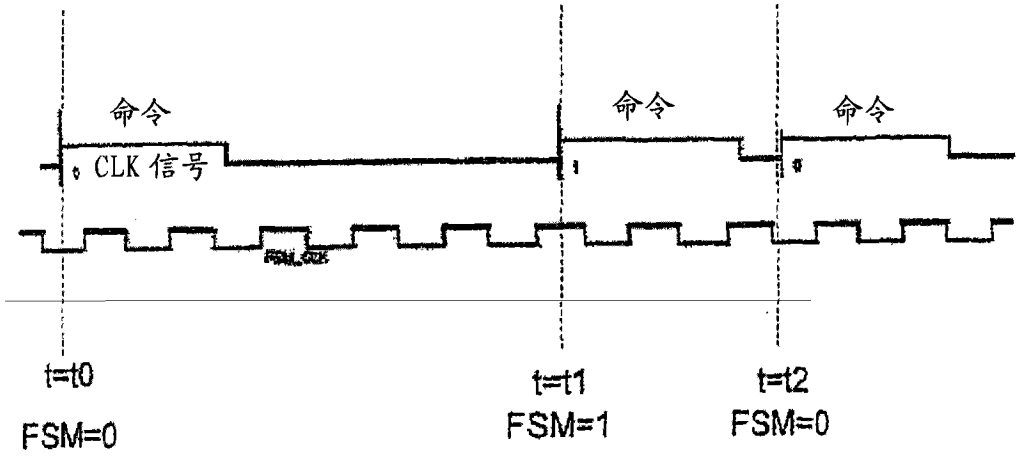


图 24A

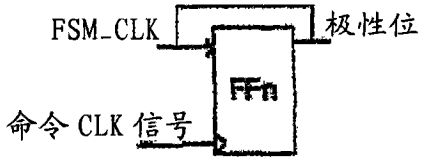


图 24B

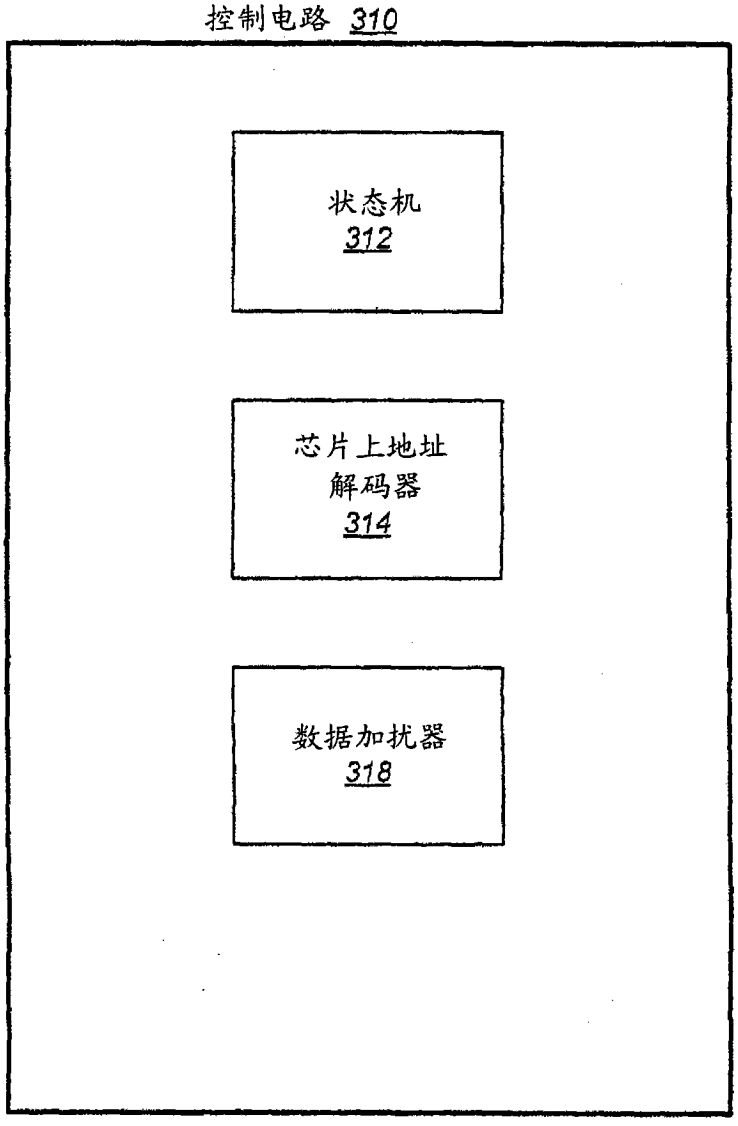
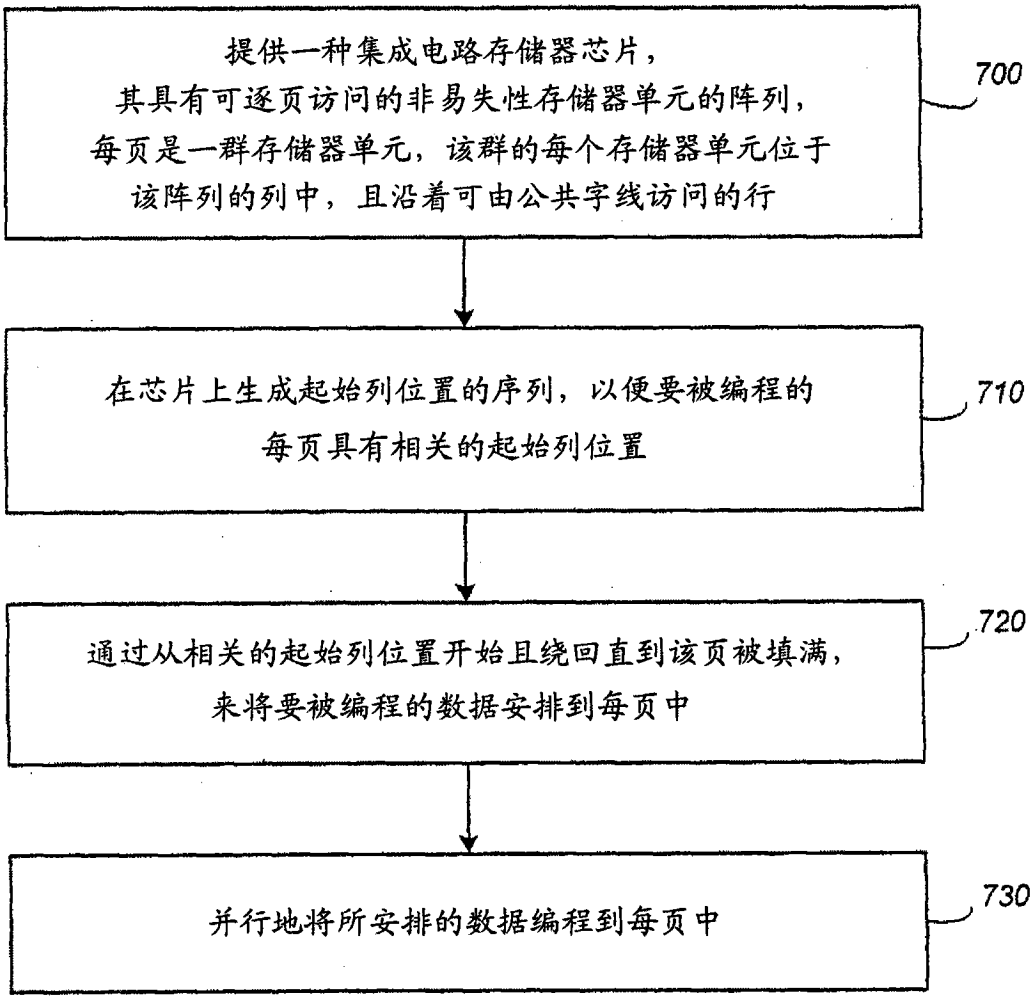


图 25



通过偏移起始位置来加扰一页中的数据

图 26

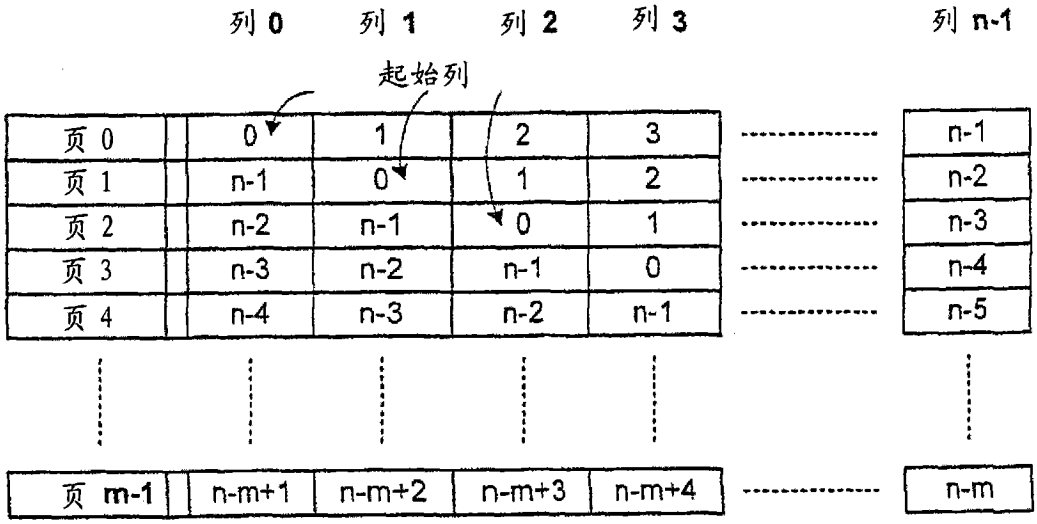

列 0
列 1
列 2
列 3
列 n-1
起始列
页 0
0
1
2
3
n-1
页 1
n-1
0
1
2
n-2
页 2
n-2
n-1
0
1
n-3
页 3
n-3
n-2
n-1
0
n-4
页 4
n-4
n-3
n-2
n-1
n-5
页 m-1
n-m+1
n-m+2
n-m+3
n-m+4
n-m

图 27

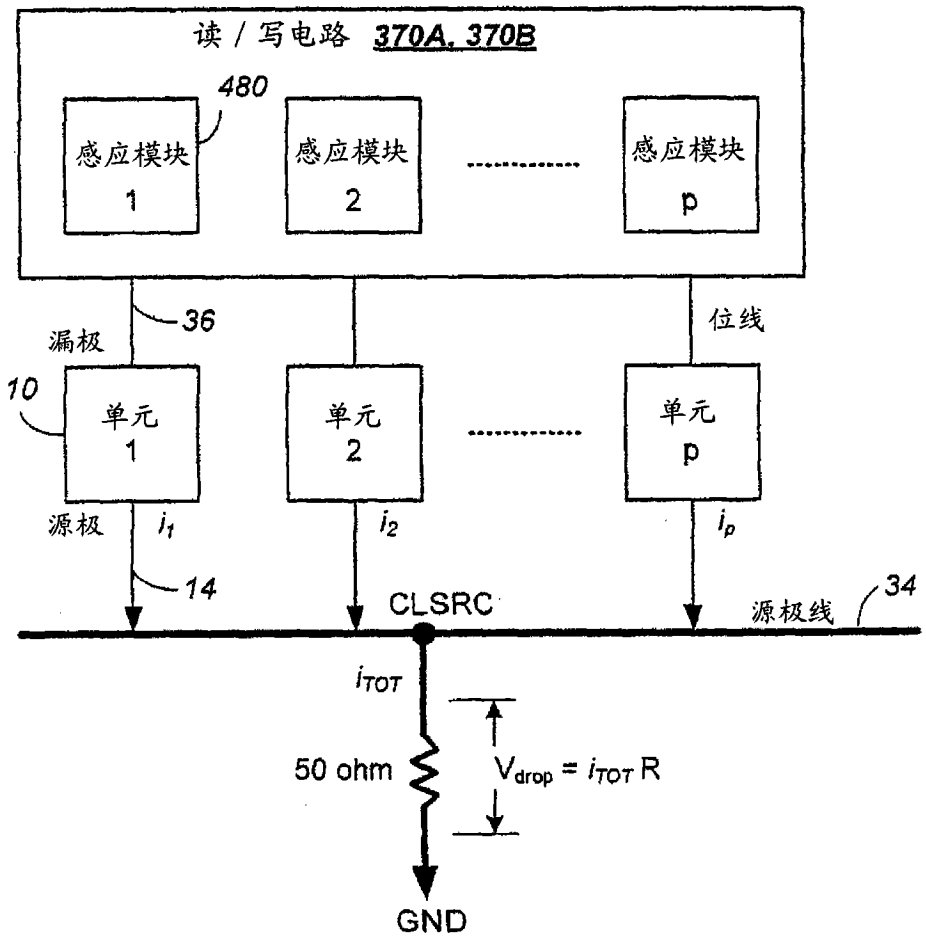

读 / 写电路 370A, 370B
480
感应模块 1
感应模块 2
感应模块 p
36
漏极
位线
10
单元 1
单元 2
单元 p
源极
i_1
i_2
i_p
14
CLSRC
源极线
34
i_{TOT}
50 ohm
$V_{drop} = i_{TOT} R$
GND

图 28A

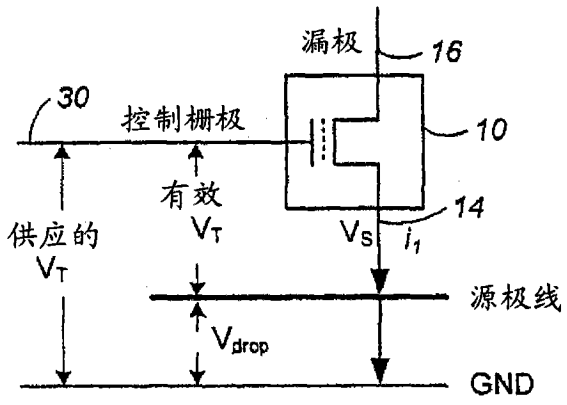
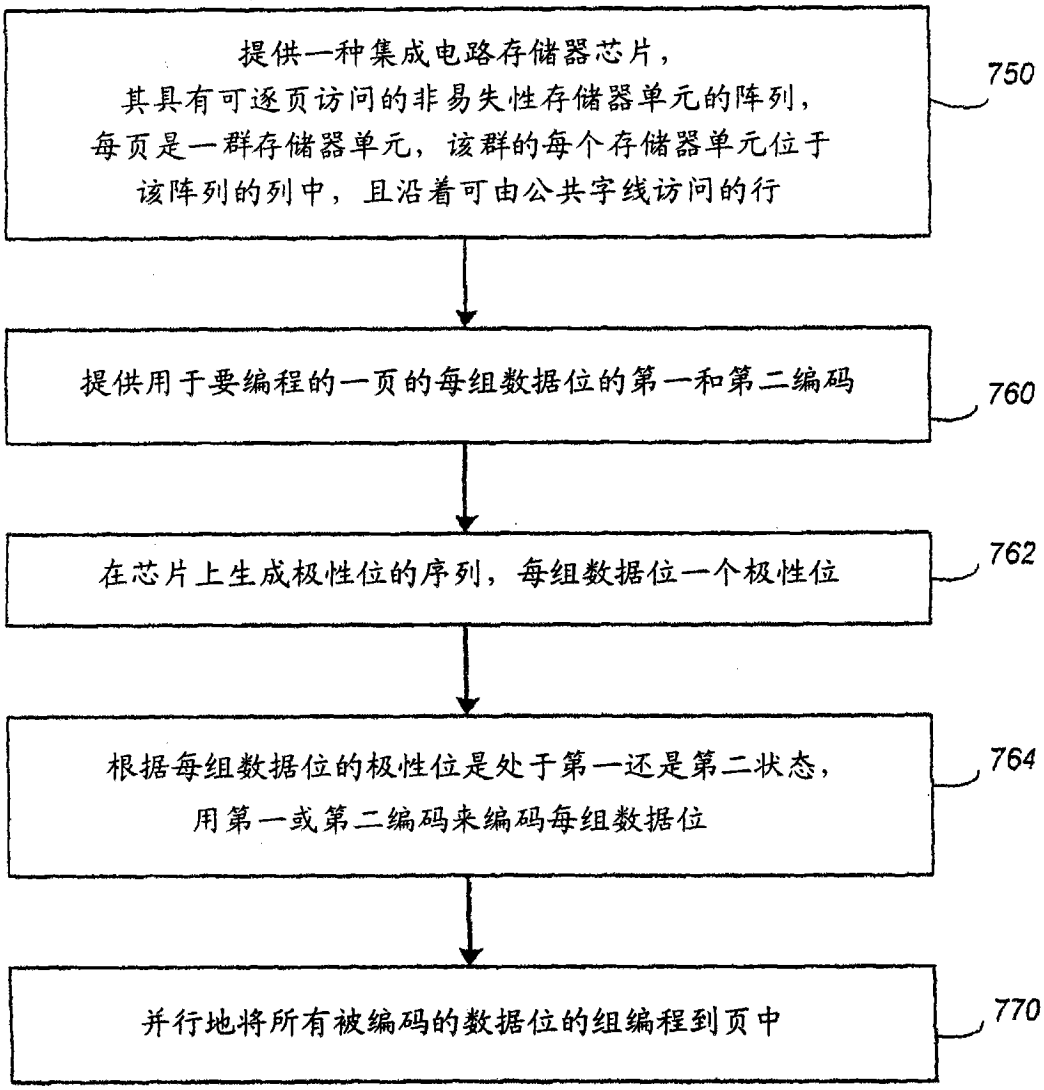
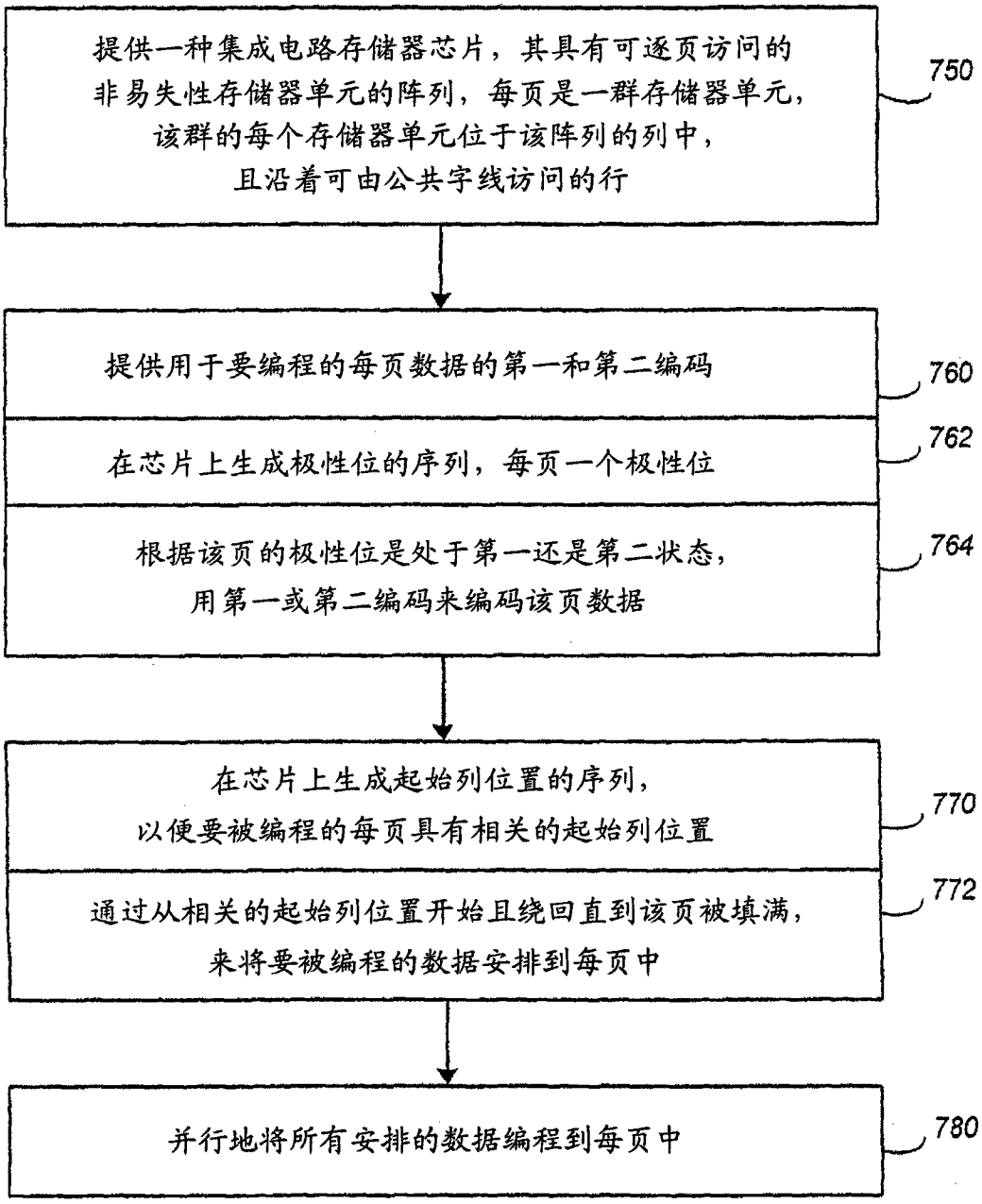


图 28B



随机化每页内的位

图 29



逐页并在每页内随机化

图 30