



(12) 发明专利

(10) 授权公告号 CN 1853239 B

(45) 授权公告日 2012.02.29

(21) 申请号 200480026443.2

(56) 对比文件

(22) 申请日 2004.06.30

US 6456528 B1, 2002.09.24, 全文.

(30) 优先权数据

CN 1466765 A, 2004.01.07, 全文.

10/629,068 2003.07.29 US

US 5771349 A, 1998.06.23, 全文.

(85) PCT申请进入国家阶段日

审查员 刘清泉

2006.03.14

(86) PCT申请的申请数据

PCT/US2004/021254 2004.06.30

(87) PCT申请的公布数据

W02005/015566 EN 2005.02.17

(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 陈建 杰弗里·W·卢策

(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 11/56(2006.01)

G11C 16/34(2006.01)

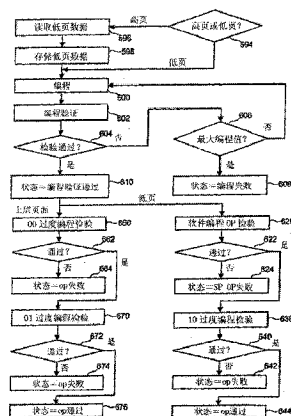
权利要求书 4 页 说明书 18 页 附图 12 页

(54) 发明名称

检测过度编程的存储器

(57) 摘要

在非易失性半导体存储器系统(或其它类型的存储器系统)中,通过改变所述存储器单元的阈电压来编程一存储器单元。由于所述系统中不同存储器单元的编程速度的变化,所以存在某些存储器单元将被过度编程的可能性。即,在一个实例中,阈电压将被移动超过预期值或预期的值范围。本发明包括判定所述存储器单元是否被过度编程。



1. 一种用于在使用多个逻辑页的存储器系统中检测过度编程的方法,其包含以下步骤:

使用要存储在一特定逻辑页中的数据来编程一第一多状态存储元件;和

使用来自不同于所述特定逻辑页的一逻辑页的数据来判定用于编程所述第一多状态存储元件的所述步骤是否过度编程所述多状态存储元件。

2. 根据权利要求 1 所述的方法,其进一步包含以下步骤:

在使用要储存在所述特定逻辑页中的数据来编程所述第一多状态存储元件之前,通过使用要储存在不同于所述特定逻辑页的所述逻辑页中的数据来编程所述第一多状态存储元件。

3. 根据权利要求 2 所述的方法,其中所述使用数据的步骤包括以下步骤:

判定所述第一多状态存储元件的一阈电压是否高于一比较值;和

如果所述第一多状态存储元件的所述阈电压高于所述比较值,且用于所述逻辑页的数据指示所述第一多状态存储元件的所述阈电压不应高于所述比较值,那么判定所述第一多状态存储元件被过度编程。

4. 根据权利要求 3 所述的方法,其中:

所述第一多状态存储元件能够存储处于包括状态 0、状态 1、状态 2 和状态 3 在内的至少四个状态中的至少两个比特的数据;

所述两个比特的数据的一第一比特用于所述特定逻辑页;

所述两个比特的数据的一第二比特用于不同于所述特定逻辑页的所述逻辑页;

所述第一比特在状态 2 和状态 3 中被编程;

所述第二比特在状态 1 和状态 2 中被编程;

所述比较值在状态 2 与状态 3 之间;且

如果编程所述第二比特使得所述第一多状态存储元件应处于状态 2 中,那么用于不同于所述特殊页的所述逻辑页的所述数据指示所述第一多状态存储元件的所述阈电压不应高于所述比较值。

5. 根据权利要求 1 所述的方法,其中所述使用数据的步骤包括以下步骤:

执行一检验操作以判定所述第一多状态存储元件的一阈电压;且

使用来自不同于所述特定逻辑页的所述逻辑页的所述数据来判定所述阈电压是否应高于一第一比较点。

6. 根据权利要求 1 所述的方法,其中:

所述第一多状态存储元件能够存储多个状态中的多个比特的数据;

所述多个比特的数据的一第一比特用于所述特定逻辑页;

所述多个比特的数据的一第二比特用于不同于所述特定逻辑页的所述逻辑页;且

所述使用数据的步骤包括判定使用要储存在所述特定逻辑页中的数据来编程所述第一多状态存储元件的所述步骤是否期望将所述第一多状态存储元件编程为一第一状态但将所述第一多状态存储元件编程超出所述第一状态。

7. 根据权利要求 6 所述的方法,其进一步包含以下步骤:

在所述使用数据的步骤之后判定所述第一多状态存储元件是否被编程超出一第二状态。

8. 根据权利要求 7 所述的方法,其进一步包含以下步骤:

在使用要储存在所述特定逻辑页中的数据来编程所述第一多状态存储元件之前,使用要储存在不同于所述特定逻辑页的所述逻辑页中的数据来编程所述第一多状态存储元件,所述特定逻辑页是一高页,不同于所述特定逻辑页的所述逻辑页是一低页;和

响应使用要储存在不同于所述特定逻辑页的所述逻辑页中的数据来编程所述第一多状态存储元件的所述步骤,判定所述第一多状态存储元件是否被编程超出一第三状态,所述第三状态与编程所述第二比特相关联,响应使用要储存在一特定逻辑页中的数据来编程一第一多状态存储元件的所述步骤执行使用并判定所述第一多状态存储元件是否被编程超出一第二状态的所述步骤。

9. 根据权利要求 1 所述的方法,其进一步包含以下步骤:

在编程所述第一多状态存储元件之后且在所述使用步骤之前编程一第二多状态存储元件,所述第一多状态存储器连接到一第一字线且所述第一多状态存储器连接到一第二字线。

10. 根据权利要求 1 所述的方法,其中:

所述第一多状态存储元件使用根据一格雷码的多状态配置来存储数据。

11. 根据权利要求 1 所述的方法,其中:

由一状态机来执行编程并使用数据的所述步骤。

12. 根据权利要求 1 所述的方法,其中:

所述第一多状态存储元件是一存储元件阵列的一部分;

所述存储元件阵列位于一集成电路芯片上;

由一状态机来执行编程并使用数据的所述步骤;且

所述状态机位于所述集成电路芯片上。

13. 根据权利要求 1 所述的方法,其中:

所述第一多状态存储元件是一存储元件阵列的一部分;

所述存储元件阵列位于一集成电路芯片上;

所述存储器系统包括一控制器;

所述控制的并不位于所述集成电路芯片上;

由一状态机来执行编程并使用数据的所述步骤;且

所述状态机位于所述集成电路芯片上。

14. 根据权利要求 1 所述的方法,其中:

所述第一多状态存储元件是一存储元件阵列的一部分;

所述存储元件阵列位于一集成电路芯片上;且

由所述集成电路芯片上的电路来执行编程并使用数据的所述步骤。

15. 根据权利要求 1 所述的方法,其中:

所述第一多状态存储元件是一存储元件阵列的一部分;

响应来自一与所述存储元件阵列进行通信的主机系统的命令来执行所述编程步骤;且

所述存储元件阵列可从所述主机系统移除。

16. 根据权利要求 1 所述的方法,其中:

所述第一多状态存储元件是一快闪存储器元件。

17. 根据权利要求 1 所述的方法, 其中:

所述第一多状态存储元件是一 NAND 快闪存储器元件。

18. 一种用于在使用多个逻辑页的存储器系统中检测过度编程的方法, 其包含以下步骤:

判定一第一多状态存储元件的一阈电压是否高于一比较值, 所述第一多状态存储元件存储至少一第一逻辑页和一第二逻辑页的信息, 在将数据写入到所述第二逻辑页之后执行判定所述第一多状态存储元件的所述阈电压是否高于所述比较值的所述步骤; 和

基于来自所述第一逻辑页的数据和判定一第一多状态存储元件的一阈电压是否高于一比较值的所述步骤来判定所述第一多状态存储元件是否被过度编程。

19. 根据权利要求 18 所述的方法, 其中:

所述第一多状态存储元件存储使用多个状态的数据;

至少所述多个状态的一第一状态和一第二状态通过使用要储存在所述第二逻辑页中的数据来相关联;

将来自所述第一逻辑页的所述数据存储于所述第一多状态存储元件中;

来自所述第一逻辑页的所述数据指示所述第一多状态存储元件应处于所述第一状态或所述第二状态的哪一个中; 且

如果所述第一多状态存储元件应处于所述第一状态中但已被编程超出所述第一状态, 那么所述第一多状态存储元件被过度编程。

20. 根据权利要求 18 所述的方法, 其中:

所述第一多状态存储元件是一存储元件阵列的一部分;

所述存储元件阵列位于一集成电路芯片上; 且

由所述集成电路芯片上的一个或一个以上电路来执行判定一多状态存储元件的一阈电压是否高于一比较值和判定所述第一多状态存储元件是否被过度编程的所述步骤。

21. 根据权利要求 18 所述的方法, 其中:

所述第一多状态存储元件是一 NAND 快闪存储器元件。

22. 一种存储器系统, 其包含:

一存储元件阵列; 和

一与所述存储元件阵列进行通信的管理电路, 所述管理电路执行包括使用要储存在一特定逻辑页中的数据来编程一第一多状态存储元件在内的编程操作, 所述管理电路还执行过度编程检测, 其包括使用来自不同于所述特定逻辑页的一逻辑页的数据来判定使用要储存在所述特定逻辑页中的数据对所述第一多状态存储元件的所述编程是否过度编程所述第一多状态存储元件。

23. 根据权利要求 22 所述的存储器系统, 其中:

所述管理电路包括一状态机。

24. 根据权利要求 23 所述的存储器系统, 其进一步包含:

一控制器, 所述状态机和所述存储元件阵列都位于一第一集成电路芯片上且所述控制器不位于所述第一集成电路芯片上。

25. 根据权利要求 24 所述的存储器系统, 其中:

所述状态机执行所述过度编程检测。

26. 根据权利要求 23 所述的存储器系统,其中:

在使用所述特定逻辑页的数据来编程所述第一多状态存储元件之前,所述管理电路进一步使用不同于所述特定逻辑页的所述逻辑页的所述数据来编程所述第一多状态存储元件。

27. 根据权利要求 26 所述的存储器系统,其中:

所述管理电路执行所述过度编程检测,此通过判定所述第一多状态存储元件的一阈电压是否高于一比较值且通过如果所述第一多状态存储元件的所述阈电压高于所述比较值且用于所述逻辑页的数据指示所述第一多状态存储元件的所述阈电压不应高于所述比较值,那么判定所述第一多状态存储元件被过度编程来达到。

28. 一种使用多个逻辑页的存储器系统,其包含:

用于使用要存储在一特定逻辑页中的数据来编程一第一多状态存储元件的构件;和
用于使用来自不同于所述特定逻辑页的一逻辑页的数据判定所述第一多状态存储元件是否被过度编程的构件。

29. 根据权利要求 28 所述的存储器系统,其进一步包含:

用于在编程所述第一多状态存储元件之后且在所述使用步骤之前编程一第二多状态存储元件的构件,所述第一多状态存储器连接到一控制线且所述第二多状态存储器连接到一控制线。

30. 根据权利要求 29 所述的存储器系统,其中:

所述第一多状态存储元件是一 NAND 快闪存储器元件。

检测过度编程的存储器

[0001] 相关申请案的交叉参考

[0002] 本申请案涉及 Jian Chen, Yan Li 和 Jeffrey W. Lutze 的题为“DETECTING OVERPROGRAMMED MEMORY AFTER FURTHER PROGRAMMING”的美国专利申请案,其与本申请案同一天提出申请且以引用的方式并入本文中。

技术领域

[0003] 本发明一般涉及用于存储器装置的技术,且更确切地说,本发明涉及检测存储器装置是否已过度编程。

背景技术

[0004] 半导体存储器装置在各种电子装置中的使用已变得越来越普遍。举例来说,非易失性半导体存储器用于蜂窝电话、数码相机、个人数字助理、移动计算装置、非移动计算装置和其它装置中。电可擦除可编程只读存储器 (EEPROM) 和快闪存储器是其中最普遍的非易失性半导体存储器。

[0005] 典型的 EEPROM 和快闪存储器利用具有一浮动栅极的存储器单元,所述浮动栅极提供于一半导体衬底中的通道区上并与所述通道区绝缘。所述浮动栅极定位于源极区与漏极区之间。一控制栅极提供于浮动栅极上并与所述浮动栅极绝缘。存储器的阈电压由保持在浮动栅极上的电荷量来控制。即,接通存储器单元以允许其源极与漏极之间的传导之前所必须施加到控制栅极的最低电压量由浮动栅极上的电荷电平来控制。

[0006] 某些 EEPROM 和快闪存储器装置具有一用于存储两个电荷范围的浮动栅极,且因此可在两个状态之间编程/擦除存储器单元。当编程 EEPROM 或快闪存储器装置时,施加一编程电压到控制栅极并将位线接地。将来自 p 阱的电子注入到浮动栅极中。当电子在浮动栅极中聚集时,浮动栅极变为带负电且存储器单元的阈电压升高。

[0007] 通常,施加到控制栅极的编程电压是以脉冲序列的方式施加的。所述脉冲的量值随每一个脉冲而增加一预定步长。在脉冲之间的周期中,进行检验操作。即,在每一个编程脉冲之间读取正并行地编程的每一个单元的编程电平,以判定其是否等于或高于正编程的单元的检验电平。一种检验所述编程的方法是测试一特定比较点处的传导性。

[0008] 传导表示装置的“接通”状态,所述状态对应于电流流过装置的通道。“关闭”状态对应于没有电流流过源极与漏极之间的通道。通常,如果施加到控制栅极的电压大于阈电压,那么快闪存储器就传导,且如果施加到控制栅极的电压小于阈电压,那么所述快闪存储器就不传导。通过将单元的阈电压设置到适当值,可使得所述单元在一给定组的施加电压下可传导或不可传导电流。因此,可通过判定单元在一给定组的施加电压下是否传导电流来判定所述单元的状态。

[0009] 可通过在装置内识别多个不同的阈电压范围来制造多位或多状态快闪存储单元。每个不同的阈电压范围对应所述组数据位的预定值。被编程到存储器单元中的数据与所述单元的阈电压电平之间的特定关系取决于所述单元所采用的数据编码方案。举例来说,

美国专利第 6,222,762 号和 2003 年 6 月 13 日提出申请的 "Tracking Cells For A Memory System" 的美国专利申请案第 10/461,244 号描述了多种数据编码方案,二者的全文以引用的方式并入本文中。适当的数据存储要求多状态存储器单元的阈电压电平的多个范围彼此相差足够余量 (margin),使得可以明确的方式来编程或擦除存储器单元的电平。

[0010] 在许多状况下,举例来说,为了生产商业上所需要的可在合理时间量内编程的存储器系统,有必要并行地编程多个存储器单元。然而,当同一时间要编程许多存储器单元时会出现问题。这是因为:由于包含存储器单元的半导体装置在结构和操作上存在微小变化,因此每个存储器单元具有不同的特征;因此,不同单元的编程速度通常会不同。这导致了某些存储器单元的编程过程可能会稍快于其它单元且某些存储器单元可能会编程为不同于预期状态的状态。多个存储器单元的较快编程可导致超出所要的阈电压电平范围,从而在正存储的数据中产生错误。

[0011] 通常,当编程数据时,对装置进行检验处理能保证存储器单元的阈电压高于最低电平。然而,装置通常不能保证阈电压的上限。某些装置查看软件编程过程(下文所述)是否将阈电压升得过高;然而,这些装置不查看规则的编程过程是否将阈电压升得过高。因此,稍不注意就有可能发生升高阈电压以致超出所要状态的范围的过度编程。过度编程可导致存储器单元存储错误数据,从而导致随后的读取操作过程中的错误。可在美国专利第 5,321,699 号、第 5,386,422 号、第 5,469,444 号、第 6,134,140 号和第 5,602,789 号中找到有关过度编程的更多信息。

[0012] 为了校正过度编程,许多存储器系统在随后的读取操作过程中使用错误校正码 ("ECC")。当从装置读取数据时,使用 ECC 来判定是否发生错误。如果所述错误很小,那么可使用 ECC 来校正所述错误。然而,当使用 ECC 来校正由于过度编程而产生的错误时,至少产生三个问题。第一,ECC 过程需要大量处理时间,且因此大大减慢存储器系统的操作。第二,ECC 需要附加的专用硬件以在合理的时间量内执行 ECC。此种专用硬件可占用存储器系统芯片的大量空间。目前的研发趋势是减小存储器系统的尺寸以在系统中能放置更多的存储器单元且使得系统尽可能地小以适用于较小的主机装置。因此,新近的存储器设计通常正在减少可用于 ECC 的不动产 (real estate) 量。第三,如果存在多个错误,那么 ECC 便不能校正所述错误。

[0013] 因此,需要一种改进的方法来检测存储器单元的过度编程。

发明内容

[0014] 概括地来说,本发明属于用于检测存储器装置是否已被过度编程的技术。一旦检测到一个或一个以上存储器装置已被过度编程,那么系统可通过多种用于校正数据的已知方法中的任一种方法来加以校正所述情况。举例来说,系统可将数据重写到相同或不同的存储器装置。这种过度编程检测确保数据不会由于过度编程而发生错误,使得可减少对 ECC 的需要。

[0015] 本发明的一个实施例包括一种用于在使用多个逻辑页的存储器系统中检测过度编程的方法。所述方法包括使用特定逻辑页的数据和来自不同逻辑页的数据编程第一多状态存储元件,以判定所述第一多状态存储元件是否过度编程。在一个实施中,所述系统判定所述第一多状态存储元件的阈电压是否大于比较值。如果所述第一多状态存储元件的阈电

压大于所述比较值且不同逻辑页的数据指示所述第一多状态存储元件的阈电压不应大于所述比较值,那么系统推断出所述第一多状态存储元件被过度编程。在两个逻辑页被编程后执行过度编程检测。

[0016] 本发明的另一实施例包括编程与第一控制线相关联的一个或一个以上多状态存储元件,且随后编程与第二控制线相关联的一个或一个以上多状态存储元件。在编程与第二控制线相关联的一个或一个以上多状态存储元件后,做出有关与第一控制线相关联的一个或一个以上多状态存储元件是否被过度编程的判定。所述控制线可为字线、位线或另一类型的控制线(视技术而定)。在一个实施例中,通过以下步骤来做出过度编程判定:在与第一控制线相关联的多状态存储元件上执行读取操作以得到一个或一个以上过度编程范围的边缘;且如果与第一控制线相关联的多状态存储元件具有在所述过度编程范围中的一个范围内的阈电压,那么判定其被过度编程。在另一实施例中,通过以下步骤来做出过度编程判定:在与第一控制线相关联的多状态存储元件上执行读取操作以得到一组一个或一个以上读取比较点,以判定与第一控制线相关联的多状态存储元件的初始状态;对与第一控制线和初始状态相关联的多状态存储元件执行错误校正码处理;且如果所述错误校正码处理对多状态存储元件失效,那么判定所述多状态存储元件被过度编程。

[0017] 在又一实施例中,本发明包括对一第一多状态存储元件进行编程且对一第二多状态存储元件进行编程。系统在对所述第二多状态存储元件进行编程后判定所述第一多状态存储元件是否被过度编程。

[0018] 在一个实施例中,实施本发明的系统包括一存储元件阵列和一管理电路。所述管理电路可包括专用硬件和/或可包括由存储于一个或一个以上存储器装置(例如,非易失性存储器(例如快闪存储器、EEPROM等)或其它存储器装置)上的软件编程的硬件。在一个实施例中,所述管理电路包括一控制器和一状态机。在另一实施例中,所述管理电路仅包括状态机而不包括控制器。所述管理电路可执行上文论述的步骤。在某些实施例中,由状态机来执行检验存储元件是否被过度编程的过程。在某些实施中,状态机与存储元件阵列处于同一集成电路芯片上。

[0019] 本发明的这些和其它目标以及优点将从以下描述中更显而易见,其中结合附图阐述本发明的优选实施例。

附图说明

[0020] 图1是与非(NAND)串的俯视图。

[0021] 图2是NAND串的等效电路图。

[0022] 图3是NAND串的横截面图。

[0023] 图4是描绘三个NAND串的电路图。

[0024] 图5是其中实施了本发明的多个方面的非易失性存储器系统的一个实施例的方框图。

[0025] 图6说明存储器阵列的构造的一个实例。

[0026] 图7描绘列控制器电路的一部分。

[0027] 图8描绘存储器单元的阈分布并说明用于编程多状态存储器单元的技术的一个实例。

- [0028] 图 9 是描述用于编程一存储器阵列的过程的一个实施例的流程图。
- [0029] 图 10-13 是描述用于执行过度编程检测的一组过程的一个实施例的流程图。
- [0030] 图 14A 和 14B 是描述用于编程和执行过度编程检测的过程的两个实施例的流程图。
- [0031] 图 15A 描绘存储器的阈分布。
- [0032] 图 15B 和 15C 是描述用于执行过度编程检测的过程的两个实施例的流程图。
- [0033] 图 16 描绘具有用于执行过度编程检测的另一组比较点的存储器的阈分布。
- [0034] 图 17-22 是执行过度编程检测的附加实施例的真值表。

具体实施方式

[0035] 快闪存储器系统的一个实例使用 NAND 结构,其包括在两个选择栅极之间串联排列多个晶体管。串联的晶体管和选择栅极称为 NAND 串。图 1 是显示一个 NAND 串的俯视图。图 2 是其等效电路。图 1 和图 2 中所描绘的 NAND 串包括四个串联的晶体管 100、102、104 和 106,且所述四个晶体管夹在第一选择栅极 120 与第二选择栅极 122 之间。选择栅极 120 将 NAND 串连接到位线 126。选择栅极 122 将 NAND 串连接到源极线 128。通过向控制栅极 120CG 施加适当电压来控制选择栅极 120。通过向控制栅极 122CG 施加适当电压来控制选择栅极 122。晶体管 100、102、104 和 106 中的每一个都具有一控制栅极和一浮动栅极。晶体管 100 具有控制栅极 100CG 和浮动栅极 100FG。晶体管 102 包括控制栅极 102CG 和浮动栅极 102FG。晶体管 104 包括控制栅极 104CG 和浮动栅极 104FG。晶体管 106 包括控制栅极 106CG 和浮动栅极 106FG。控制栅极 100CG 连接到字线 WL3,控制栅极 102CG 连接到字线 WL2,控制栅极 104CG 连接到字线 WL1,且控制栅极 106CG 连接到字线 WL0。

[0036] 图 3 提供上述 NAND 串的横截面图。如同图 3 中所描述, NAND 串的晶体管(又称作单元或存储器单元)形成于 p 阱区 140 中。每个晶体管包括由控制栅极(100CG、102CG、104CG 和 106CG)与浮动栅极(100FG、102FG、104FG 和 106FG)组成的一堆叠栅极结构。所述浮动栅极形成于氧化物薄膜顶端上的 p 阱表面上。控制栅极位于浮动栅极上方,其中氧化物层分隔控制栅极与浮动栅极。应注意的是,图 3 似乎描述了晶体管 120 和 122 的控制栅极和浮动栅极。然而,对于晶体管 120 和 122 来说,控制栅极与浮动栅极是连接在一起的。存储器单元(100、102、104、106)的控制栅极形成字线。相邻单元之间共享 N+ 扩散层 130、132、134、136 和 138,借此所述这些单元彼此串联连接以形成一 NAND 串。这些 N+ 扩散层形成每个单元的源极和漏极。举例来说,N+ 扩散层 130 充当晶体管 122 的漏极并充当晶体管 106 的源极,N+ 扩散层 132 充当晶体管 106 的漏极并充当晶体管 104 的源极,N+ 扩散区 134 充当晶体管 104 的漏极并充当晶体管 102 的源极,N+ 扩散区 136 充当晶体管 102 的漏极并充当晶体管 100 的源极,且 N+ 扩散层 138 充当晶体管 100 的漏极并充当晶体管 120 的源极。N+ 扩散层 126 连接到所述 NAND 串的位线,而 N+ 扩散层 128 连接到多个 NAND 串的共用源极线。

[0037] 应注意的是,尽管图 1-3 显示 NAND 串中的四个存储器单元,但是仅提供使用四个晶体管的作为实例。一 NAND 串的存储器单元的个数可少于四个或多于四个。举例来说,某些 NAND 串将包括 8 个存储器单元、16 个存储器单元、32 个存储器单元等。本文的讨论并不受限于 NAND 串中的任何特定个数的存储器单元。

[0038] 使用 NAND 结构的快闪存储器系统的典型构造将包括若干 NAND 串。举例来说,图 4 显示具有更多的 NAND 串的存储器阵列的三个 NAND 串 202、204 和 206。图 4 的每个 NAND 串包括两个选择晶体管和四个存储器单元。举例来说, NAND 串 202 包括选择晶体管 220、230 和存储器单元 220、224、226、228。NAND 串 204 包括选择晶体管 240、250 和存储器单元 242、244、246、248。每一串通过其选择晶体管(例如,选择晶体管 230 和选择晶体管 250)连接到源极线。选择线 SGS 用于控制源极侧的选择栅极。各个 NAND 串通过选择晶体管 220、240 等连接到各自的位线,其中所述选择晶体管由选择线 SGD 控制。在其它实施例中,未必需要共用选择线。字线 WL3 连接到存储器单元 222 和存储器单元 242 的控制栅极。字线 WL2 连接到存储器单元 224、存储器单元 244 和存储器单元 250 的控制栅极。字线 WL1 连接到存储器单元 226 和存储器单元 246 的控制栅极。字线 WL0 连接到存储器单元 228 和存储器单元 248 的控制栅极。如同可看到的,每个位线和各自的 NAND 串包含存储器单元阵列的列。字线(WL3、WL2、WL1、和 WL0)包含所述阵列的行。

[0039] 每个存储器单元可存储数据(模拟或数字数据)。当存储一个位的数字数据时,存储器单元的可能的阈电压范围划分成两个范围,其分配为逻辑数据"1"和"0"。在 NAND 型快闪存储器的一个实例中,擦除存储器单元之后的阈电压为负,且被界定为逻辑"1"。编程操作之后的阈电压为正,且被界定为逻辑"0"。当阈电压为负且试图读取时,存储器单元将会接通以指示正在存储逻辑 1。当阈电压为正且试图读取操作时,存储器单元将不会接通,其指示逻辑 0 被存储。存储器单元还可存储多个信息电平,例如,多个位的数字数据。在存储多个数据电平的情况下,可能的阈电压范围划分成数据电平的个数。举例来说,如果存储四个信息电平,那么会存在分配有数据值"11"、"10"、"01"和"00"的四个阈电压范围。在 NAND 型存储器的一个实例中,擦除操作之后的阈电压为负且被界定为"11"。正阈电压用于状态"10"、"01"和"00"。

[0040] 下述美国专利/专利申请案中提供了 NAND 型快闪存储器的相关实例及其操作(所有申请案的全文以引用的方式并入本文中):美国专利第 5,570,315 号、美国专利第 5,774,397 号、美国专利第 6,046,935 号、美国专利第 6,456,528 号和美国专利申请案第 09/893,277 号(公开案第 US2003/0002348 号)。其它类型的快闪存储器装置也可与本发明一起使用。举例来说,下述专利描述或非型快闪存储器且所述专利的全文以引用的方式并入本文中:美国专利第 5,095,344 号、第 5,172,338 号、第 5,890,192 号和第 6,151,248 号。美国专利第 6,151,248 号中可找到快闪存储器类型的另一实例,所述案的全文以引用的方式并入本文中。

[0041] 图 5 是可用于实施本发明的快闪存储器系统的一个实施例的方框图。存储器单元阵列 302 由列控制器电路 304、行控制器电路 306、c 源极控制器电路 310 和 p 阱控制器电路 308 来控制。列控制器电路 304 连接到存储器单元阵列 302 的位线,以用于读取存储于存储器单元中的数据、用于在编程操作过程中判定存储器单元的状态,且用于控制位线的电位电平以促进编程或禁止编程。行控制器电路 306 连接到字线以选择所述字线中的一个来施加读取电压、施加编程电压且施加擦除电压。C 源极控制器电路 310 控制连接到存储器单元的共用源极线(图 6 中标记为"C 源极")。P 阱控制器电路 308 控制 p 阱电压。

[0042] 由列控制器电路 304 读出存储于存储器单元中的数据并经由数据输入/输出缓冲器 312 输出到外部 I/O 线。待存储于存储器单元中的编程数据经由外部 I/O 线输入到数据

输入 / 输出缓冲器 312, 并传输到列控制器电路 304。外部 I/O 线连接到控制器 318。

[0043] 将用于控制快闪存储器装置的命令数据输入到控制器 318。命令数据通知快闪存储器所请求的是何操作。将输入命令传输到状态机 316, 其控制列控制器电路 304、行控制器电路 306、c 源极控制器电路 310、p 阱控制器电路 308 和数据输入 / 输出缓冲器 312。状态机 316 还可输出快闪存储器的状态数据, 例如 READY/BUSY 或 PASS/FAIL。

[0044] 控制器 318 连接到或可连接到主机系统, 例如个人计算机、数码相机、个人数字助理等。控制器 318 与主机通信以接收来自主机的命令、接收来自主机的数据、向主机提供数据并向主机提供状态信息。控制器 318 将来自主机的命令转换成可由命令电路 314 解译并执行的命令信号, 其中所述命令电路 314 与状态机 316 进行通信。控制器 318 通常含有缓冲存储器, 其用于写入到存储器阵列或自存储器阵列读取的用户数据。

[0045] 一个示范性存储器系统包含: 一个集成电路, 其包括控制器 318; 和一个或一个以上集成电路芯片, 其中每个集成电路芯片含有一存储器阵列和相关的控制、输入 / 输出和状态机电路。目前的研发趋势是将系统的存储器阵列和控制器电路一起集成在一个或一个以上集成电路芯片上。可将存储器系统嵌入作为主机系统的一部分, 或可将其包括于一可移除地插入到主机系统中的存储器卡 (或其封装) 中。此类可移除卡可包括整个存储器系统 (例如, 包括控制器) 或仅包括存储器阵列和相关的外围电路 (控制器嵌入主机中)。因此, 控制器可嵌入主机中或包括于可移除的存储器系统内部。

[0046] 参看图 6, 其描述了存储器单元阵列 302 的实例结构。作为一个实例, 其描述了分割为 1, 024 个块的 NAND 快闪 EEPROM。可同时擦除存储于每个块中的数据。在一个实施例中, 块是可同时擦除的单元的最小单位。在此实例中, 每个块中存在 8, 512 个列, 其划分为偶数列和奇数列。位线同样划分成偶数字线 (BL_e) 和奇数字线 (BL_o)。图 6 显示串联连接以形成 NAND 串四个存储器单元。尽管显示每个 NAND 串中包括四个单元, 但是可使用的单元个数可多于四个或少于四个。NAND 串的一个终端经由第一选择晶体管 SGD 连接到对应位线, 且另一终端经由第二选择晶体管 SGS 连接到 c 源极。

[0047] 在读取操作和编程操作过程中, 同时选择 4, 256 个存储器单元。所述选定的存储器单元具有相同字线和相同种类的位线 (例如, 偶数字线)。因此, 可同时读取或编程 532 个字节的数据。这些同时读取或编程的 532 个字节的数据形成了一逻辑页。因此, 一个块可存储至少 8 个逻辑页 (4 个字线, 其中每个字线具有奇数页和偶数页)。当每个存储器单元存储两个比特的数据 (例如, 多电平单元) 时, 一个块可存储 16 页。具有其它尺寸的块和页也可与本发明一起使用。

[0048] 通过将 p 阱升高到擦除电压 (例如, 20 伏特) 并将选定块的字线接地来擦除存储器单元。源极和位线是浮动的。可对整个存储器阵列、独立块或单元的另一单位执行擦除过程。从浮动栅极传输电子到 p 阱区且阈电压变为负。

[0049] 在读取操作和检验操作中, 将选择栅极 (SGD 和 SGS) 和未选择的字线 (例如, WL₀、WL₁ 和 WL₃) 升高到读通电压 (read pass voltage) (例如, 4.5 伏特) 以使所述晶体管可作为通过栅极来操作。选定字线 (例如, WL₂) 连接到一电压, 为每个读取操作和检验操作指定所述电压的电平以判定相关存储器单元的阈电压是否已达到此电平。举例来说, 在对两个电平的存储器单元进行读取操作的过程中, 可将选定字线 WL₂ 接地, 使得可检测阈电压是否高于 0V。在检验操作中, 举例来说, 将选定字线 WL₂ 连接到 2.4V 使得可检验阈电压是

否已达到至少 2.4V。源极和 p 阱处于零伏特。将选定位线 (BLe) 预先充电到 (例如) 0.7V 的电平。如果阈电压高于读取电平或检验电平, 那么相关位线 (BLe) 的电位电平由于非传导存储器单元而保持高水平。另一方面, 如果阈电压低于读取电平或检验电平, 那么相关位线 (BLe) 的电位电平由于传导存储器单元 (M) 而降低到低电平 (例如, 小于 0.5V)。通过连接到位线的读出放大器来检测存储器单元的状态。是擦除存储器单元还是编程存储器单元之间的差别取决于负电荷是否存储于浮动栅极中。举例来说, 如果负电荷存储于浮动栅极中, 那么阈电压会变为更高且晶体管可处于增强模式中。

[0050] 根据所属领域中的已知技术来执行上述擦除、读取和检验操作。因此, 所属领域的技术人员可改变许多所解释的细节。

[0051] 图 7 描绘图 5 的列控制器电路 304 的一部分。每一对位线 (例如, BLe 和 BLo) 耦接到读出放大器。读出放大器连接到三个数据锁存器: 第一数据锁存器 402、第二数据锁存器 404 和第三数据锁存器 406。三个数据锁存器中的每一个都能存储一个比特的数据。读出放大器在读取操作或检验操作过程中感测选定位线的电位电平、以二进制方式存储感测的数据并在编程操作过程中控制位线电压。通过选择“偶数位线”和“奇数位线”信号中的一个来选择性地连接读出放大器到选定位线。数据锁存器 402、404 和 406 耦接到 I/O 线 408 以输出读取数据并存储编程数据。I/O 线 408 连接到图 5 的数据输入/输出缓冲器 312。数据锁存器 402、404 和 406 同样耦接到状态线 410 以接收并发送状态信息。在一个实施例中, 每一对位线 (偶数位线和奇数位线) 都具有读出放大器、第一数据锁存器 402、第二数据锁存器 404 和第三数据锁存器 406。

[0052] 图 8 说明存储两个比特的数据 (例如, 四个数据状态) 的存储器单元的阈电压分布。分布 460 表示处于擦除状态 (存储“11”) 的单元的阈电压分布, 其具有负阈电压电平。分布 462 表示存储“10”的单元的阈电压分布。分布 464 表示存储“00”的单元的阈电压分布。分布 466 表示存储“01”的单元的阈电压分布。在此实例中, 存储于单个存储器单元中的两个比特中的每个比特是不同逻辑页的部分。正方形中显示的比特对应于低页。圆形中显示的比特对应于高页。在一个实施例中, 使用格雷 (gray) 码定向 (11、10、00、01) 来将这些逻辑状态分配给物理状态, 使得如果浮动栅极的阈电压错误地变动, 那么仅会影响一个比特。尽管图 8 显示四个状态, 但是本发明所使用的状态个数可少于四个或多于四个。

[0053] 为了提供改进的可靠性, 较好的措施是使个别分布变得紧密 (分布窄化), 这是因为较紧密的分布会产生较宽的读取余量 (状态之间的距离)。根据 1995 年关于 VLSI (超大规模集成电路) 技术的研讨会的摘要第 129-130 页的文章 “Fast and Accurate Programming Method for Multi-level NAND EEPROMs” (所述文章以引用的方式并入本文中), 在原理上, 将分布限制为 0.2V 的宽度需要通常的重复编程脉冲在步长之间增加 0.2V。为了使 0.05V 宽度内的分布变得紧密, 需要 0.05V 的步长。采用编程电压的此种小步长增量来编程单元会导致编程时间的增加。

[0054] 图 8 说明编程 4- 状态 NAND 存储器单元的双行程 (two pass) 技术的实例。在第一编程行程中, 根据待编程为低逻辑页的比特来设置单元的阈电压电平。如果所述比特是逻辑“1”, 那么阈电压不变, 因为其由于早先已被擦除而处于适当的状态下。然而, 如果待编程的比特是逻辑“0”, 那么, 如同箭头 470 所显示的, 所述单元的阈电平增大到处于阈

电压分布 462 内。那结束第一编程行程。

[0055] 在第二编程行程中,根据正被编程为高逻辑页的比特来设置所述单元的阈电压电平。如果高逻辑页比特要存储逻辑“1”,那么不会发生编程过程,因为所述单元取决于低页比特的编程而处于对应于阈电压分布 460 或 462 的状态中的一个,所述阈电压分布 460 和 462 的都承载有高页比特“1”。如果高页比特将为逻辑“0”,那么阈电压被改变。如果第一行程导致所述单元保持对应于极限分布 460 的擦除状态,那么,在第二相位中编程所述单元,使得阈电压增大到处于阈电压分布 466 内,如箭头 474 所描绘。如果所述单元由于第一编程行程而已编程为对应于极限分布 462 的状态,那么,在第二行程中进一步编程所述存储器单元,使得阈电压增大到处于阈电压分布 464 内,如箭头 472 所描绘。第二行程的结果是将所述单元编程为被指定为高页存储逻辑“0”而不改变用于低页的数据。

[0056] 应注意的是,尽管已将特定的比特模式分配给所述分布中的每一个,但是可分配不同的比特模式,在所述状况下,其间发生编程的状态可不同于图 8 中所描绘的状态。

[0057] 通常,被并行地编程的单元是沿一字线的交替单元。举例来说,图 4 说明沿字线 WL2 的数目大得多的单元中的三个存储器单元 224、244 和 250。一组交替单元(包括单元 224 和 250)存储来自逻辑页 0 和 2(“偶数页”)的比特,而另一组交替单元(包括单元 244)存储来自逻辑页 1 和 3(“奇数页”)的比特。在上述实例中,逻辑页 0 和 1 是低页,而逻辑页 2 和 3 是高页。

[0058] 应注意的是,图 8 还显示读取比较点和检验比较点。举例来说,读取比较点包括 Vr10、Vr00 和 Vr01。在读取操作过程中,将对应于 Vr10、Vr00 和 Vr01 的电压施加到存储器单元的控制栅极来判定所述存储器单元在哪一点处传导且在何点处不传导。基于所述三个读取操作,存储器系统可判定所述存储器单元处于哪个阈分布(例如,哪个状态)中。举例来说,如果存储器单元对所述三个读取操作中的任何一个都不传导,那么存储器单元处于状态 01 中。如果存储器单元仅在将 Vr01 施加到控制栅极时才传导,那么存储器单元处于状态 00 中。如果施加 Vr00 时存储器传导而施加 Vr10 时存储器不传导,那么存储器单元处于状态 10 中。如果施加 Vr10、Vr00 和 Vr01 时存储器单元传导,那么存储器单元处于状态 11 中。在一个实施例中,读取比较点 Vr10、Vr00 和 Vr01 中的每一个都是阈分布的相邻状态之间的中点。在编程过程中使用检验比较点 Vr10、Vr00 和 Vr01 来判定单元是否已充分编程。

[0059] 图 8 还显示过度编程检验比较点 VopSP、Vop10、Vop00 和 Vop01。这些过度编程检验比较点用于判定一特定存储器单元是否已被过度编程。举例来说,将 VopSP 施加到期望处于状态 11 下的存储器单元的控制栅极后,如果所述单元不传导,那么可判定所述单元被过度编程(过度软件编程)。如果期望所述单元处于状态 10 下,那么将 Vop10 施加到所述单元的控制栅极时,如果所述单元不传导(例如,接通),可推断所述单元被过度编程。当假设所述单元处于状态 00 下时,将 Vop00 施加到控制栅极时,如果所述存储器单元不传导,那么所述单元已被过度编程。当所述单元编程为状态 01 时,如果将 Vop01 施加到控制栅极时,如果所述单元不传导,那么可推断所述单元被过度编程。所述过度编程检验比较点中的每一个都稍低于相邻的读取比较点。举例来说,在一个实施例中,所述过度编程检验比较点低了 50mV。在其它实施例中,检验比较点与相邻的读取比较点之间的差值可多于或少于 50mV。在一个实施例中,由状态机 316 执行过度编程检验过程。

[0060] 在编程过程的一个实施例中,在低页编程之前存储器单元首先被擦除(在块或其它单位中)。当编程过程开始时,由控制器从主机接收数据负载或写入命令。控制器向数据输入/输出缓冲器 312 和状态机 316 发布适当命令。控制器 318 接收一地址,其中写入数据到所述地址。将此地址映射到一特定逻辑页(或多个)和存储器单元阵列 302 中的实际地址。将地址数据从控制器 318 发送到数据输入/输出缓冲器 312 和(在某些实施例中)状态机 316。然后,基于编程操作的地址将数据发送到适当的第一数据锁存器 402。将来自第一数据锁存器 402 的数据载入对应的第二数据锁存器 404 中。响应来自控制器的编程命令,状态机 316 接着开始图 9 的流程图所描述的编程过程。

[0061] 在步骤 594 中,状态机 316 判定数据被写入到高页还是低页。如果数据被写入到低页,那么过程在步骤 600 处继续。如果数据被写入到高页,那么状态机执行低页读取操作来判定步骤 596 中的低页数据。返回参看图 8,高页编程包括将存储器单元编程为状态 00 或状态 01。如果低页数据是逻辑“1”,那么高页编程过程包括将存储器单元编程为状态“01”(见图 8 的箭头 474)。如果低页数据是逻辑“0”,那么高页编程过程包括将存储器单元编程为状态 00(参考图 8 的箭头 472)。因此,系统需要知道低页数据以判定在高页编程过程中必须将存储器单元编程为何状态。

[0062] 在步骤 596 的一个实施例中,状态机在 Vr10 处执行读取操作。如果存储器单元传导(接通),那么低页的读取数据是“1”。如果存储器单元不传导(不接通),那么低页的读取数据是“0”。接着将低页数据存储于第三数据锁存器 406 中。应注意的是,可使用其它读取过程来读取低页数据且可将低页数据存储于除第三数据锁存器 406 之外的位置中。在将低页数据存储于第三数据锁存器 406 中之后,过程在步骤 600 处继续。

[0063] 在步骤 600 中,状态机将编程脉冲施加到适当字线。在驱动第一脉冲之前,将编程电压初始化为起始脉冲(例如,12 伏特),且将状态机所维持的编程计数器初始化为 0。在步骤 600 的第一次迭代中,将第一编程脉冲施加到选定字线。如果逻辑“0”存储于特定的第一数据锁存器中,那么将所述存储器单元的对应位线接地。相反,如果逻辑“1”存储于特定的存储器单元的第一数据锁存器中,那么将对应位线连接到 Vdd 以抑制编程。2003 年 3 月 5 日申请的题为“Improved Self Boosting Technique”的美国专利申请案第 10/379,608 号中可找到有关抑制编程的更多信息,所述案的全文以引用的方式并入本文中。

[0064] 在步骤 602 中,检验选定的存储器单元。如果检测到选定单元的目标阈电压已达到其目标电平(例如,通过将读出放大器与第二数据锁存器的输出进行比较),那么将第二数据锁存器 404 中所存储的数据变为逻辑“1”。如果检测到所述阈电压没有达到适当电平,那么不改变第二数据锁存器 404 中所存储的数据。以此方式,不需要编程对应的第二数据锁存器 404 中存储有逻辑“1”的存储器单元。当所有第二数据锁存器都存储逻辑“1”时,状态机(经由线 410)可了解到所有选定单元已被编程。因此,步骤 604 包括检验是否所有的第二数据锁存器 404 都存储逻辑“1”。如果是这样,那么完成编程的第一阶段且在步骤 610 中报告适当的状态消息。

[0065] 如果在步骤 604 中判定并不是所有的第二数据锁存器 404 都存储逻辑“1”,那么继续编程过程。在步骤 606 中,检查编程计数器的编程限制值。编程限制值的一个实例是 20。如果编程计数器并不小于 20(例如,已执行的编程尝试的最大数目),那么编程过程失败(步骤 608)并报告适当的状态消息。如果编程计数器小于最大值,那么编程电压电平增

加了下一步长、编程计数器增加且编程过程在步骤 600 处继续以施加下一脉冲。

[0066] 如果检验步骤在步骤 604 处成功,那么经编程的存储器单元的每一个应具有图 8 中所描绘的所述分布中的一个分布中的阈电压。然而,前述方法仅检验了正被编程的单元的阈电压最少与适当的检验比较点一样高。步骤 602 中所使用的检验比较点是目标阈分布中最低的可接受电压电平。举例来说,参看图 8,步骤 604 的用于检验单元被编程为状态 10 的检验比较点是 V_{v10} ,用于编程为状态 00 的检验比较点是 V_{v00} ,且用于编程为状态 01 的检验比较点是 V_{v01} 。因此,上述过程检验所述编程过程并不是没有达到所要的状态。直到此点,还没有检查所述编程过程是否超出了所要的状态。举例来说,如果我们期望将存储器单元编程为状态 10 且将所述存储器单元的阈电压被编程为高于 V_{op10} 的任何值,那么所述存储器单元被过度编程。步骤 620-676 用于判定所述编程过程是否过度编程了所述存储器单元。

[0067] 在步骤 610 之后,图 9 的过程分成了两个并行路径。如果所述编程过程编程低页,那么所述过程执行步骤 620-644。如果编程过程编程高页,那么所述过程执行步骤 660-676。可能某些单元将使一低页编程且相同的编程过程中的其它单元将使高页编程。

[0068] 因此,可对不同的存储器单元执行不同的步骤组。在另一实施例中,不管正在被编程的是低页还是高页,对正在编程的每一个单元都可执行步骤 620-644 和步骤 660-676。

[0069] 如果正在编程低页,那么执行 620。步骤 620 在编程检验过程中包括软件编程过程。即,步骤 620 包括执行如下过程:判定在擦除过程之后或作为擦除过程的一部分所执行的软件编程过程是否将所擦除的存储器单元的阈电压升高到可接受值之上。当存储器单元被擦除时,所述所擦除的存储器单元的阈电压有可能太低。为了补救太低的阈电压,执行软件编程过程以将所擦除的存储器单元的阈电压升高到可接受的电平。所述软件编程过程可能将阈电压升得太高使得所述存储器单元变为过度编程(例如,阈电压变为高于 V_{opSP})。步骤 620 检测所述过度编程单元。在步骤 622 中,判定多个单元是否通过了 620 的软件编程过度编程检验过程。如果所有单元都没有通过,那么在步骤 624 中将返回一指示所述软件编程过度编程检验失败的状态。如果通过了步骤 620 的过程(因为在所述软件编程过程中没有单元被过度编程),那么在步骤 638 中执行状态 10 过度编程检验过程。

[0070] 步骤 638 包括执行状态 10 过度编程检验过程。状态机判定期望处于状态 10 中的单元是否具有高于 V_{op10} 的阈电压。如果是这样,那么所述单元没有通过测试(步骤 640)且返回一指示存在过度编程失败的状态(步骤 642)。在一个实施例中,所述状态可包括指示其为状态 10 过度编程失败。在另一实施例中,还可返回高于可接受的电平的阈电压的量值。如果步骤 638 的过程未识别出被过度编程的存储器单元,那么在步骤 644 中返回状态“通过”。在步骤 644 的替代实施例中,所述过程可返回存在状态 10 过度编程检验通过。

[0071] 如果高页被编程,那么所述过程在步骤 660 中执行状态 00 过度编程检验过程。系统进行检查以判定期望处于状态 00 的存储器单元的阈电压是否高于 V_{op00} 。如果所述单元没有通过(因为期望处于状态 00 的存储器单元具有高于 V_{op00} 的阈电压),那么在步骤 664 中将提供一指示过度编程失败的状态。在替代实施例中,所述状态可指示其为状态 00 过度编程失败。如果判定所述单元没有被过度编程,那么所述过程循环到步骤 670 并执行状态 01 过度编程检验过程。

[0072] 在步骤 670 中,系统判定期望处于状态 01 的单元是否具有高于 V_{op01} 的阈电压。

如果期望所述单元处于状态 01 且阈电压高于检验比较点 V_{op01} , 那么所述单元没有通过所述测试且在步骤 674 中报告失败状态。如果所有单元都通过测试 (以为没有被过度编程或不处于状态 01), 那么将在步骤 676 中报告通过状态。在一个实施例中, 可省略步骤 670-676。

[0073] 图 10 是流程图, 其描述用于执行软件编程过度编程检验 (图 9 的步骤 620) 的过程的一个实施例。在步骤 700 中, 先前被编程到存储器单元中的数据仍然存在于第一数据锁存器 402 中。对于所述存储器单元, 将所述数据从第一数据锁存器 402 载入第二数据锁存器 404 中。在步骤 702 中, 使用 V_{opSP} 来执行检验过程。也就是说, 系统可执行读取操作, 其中所述控制栅极接收电压 V_{opSP} 。如果所述存储器单元传导, 那么读出放大器返回逻辑 1。如果所述单元不传导, 那么读出放大器返回逻辑 0。如果检验数据与第二数据锁存器 404 中的数据相匹配 (步骤 704), 那么将第二数据锁存器 404 中的数据设置为 1 (步骤 706)。对每一个单元来并行地执行步骤 700-706。如果所述存储器单元的所有第二数据锁存器都设置为 1 (步骤 708), 那么所述软件编程过度编程检验过程返回结果“通过” (步骤 710)。如果并不是所有的第二数据锁存器 404 都设置为 1, 那么所述过程失败 (步骤 712)。应注意的是, 如果所述单元处于状态 10, 那么, 在步骤 700 中被载入第二数据锁存器中的数据将为逻辑 0、检验步骤 702 将施加 V_{opSP} 电压, 晶体管将不传导且读取数据将指示为 0。因此, 读取数据将与第二数据锁存器 404 中的数据相匹配, 第二数据锁存器 404 将被设置为逻辑 1 且所述过程仍通过。

[0074] 图 11 是流程图, 其描述执行状态 10 过度编程检验 (图 9 的步骤 638) 的过程。在图 11 的步骤 764 中, 系统使用 V_{op10} 来执行检验过程。也就是说, 通过施加 V_{op10} 到多个存储器单元的控制栅极来执行读取操作。如果单元处于状态 11 或 10 且没有被过度编程, 那么所述单元应传导; 因此应接收到逻辑 1。在步骤 766 中, 第二数据锁存器 404 载入来自检验步骤的结果。如果所有单元都没有被过度编程, 那么其均应使逻辑 1 存储于其各自的第二数据锁存器中。如果所有第二数据锁存器都处于逻辑 1 (步骤 768), 那么所述过程通过 (步骤 770), 如果并不是所有的第二数据锁存器都处于逻辑 1, 那么所述过程失败 (步骤 772)。

[0075] 图 12 是流程图, 其描述用于执行状态 00 过度编程检验过程 (图 9 的步骤 660) 的过程的一个实施例。在图 12 的步骤 802 中, 使用 V_{op00} 来执行检验过程。也就是说, 通过施加 V_{op00} 到存储器单元的控制栅极来执行读取操作。在一个实施例中, 如果存储器单元的阈电压高于 V_{op00} , 那么所述存储器单元将不接通且将返回逻辑“0”。如果存储器单元的阈电压低于 V_{op00} , 那么所述存储器单元将接通且将返回逻辑“1”。从步骤 802 的检验返回的结果被存储于适当的第二数据锁存器 404 中。在步骤 804 中, 系统将检验步骤 802 的结果与存储于第三数据锁存器 406 中的低页数据相比较。在高页编程之后执行图 12 的过程。如同上文所描述的, 通过读取低页数据并将低页数据存储于第三数据锁存器 406 中 (见 596 和 598) 来开始高页编程。将存储于第三数据锁存器中的低页数据与检验步骤 802 的结果相比较。

[0076] 在高页编程之后, 状态机需要知道存储器单元应处于状态 00 还是状态 01。低页数据提供所述信息。如果低页数据为 0, 那么意味着所述存储器单元在高页编程之前处于状态 10。如果所述存储器单元处于状态 10 中, 那么执行高页编程会将所述存储器单元移动到状

态 00 (见图 8 的箭头 472)。如果低页数据为 1, 那么意味着所述存储器单元在高页编程之前处于状态 11。如果所述存储器单元处于状态 11, 那么执行高页编程会将所述存储器单元移动到状态 01 (见图 8 的箭头 474)。因此, 状态机知道: 如果低页数据为逻辑“0”, 那么其需要查看所述存储器单元是否被过度编程。如果低页数据为逻辑“1”, 那么所述存储器单元不可能处于状态 00, 且因此状态 00 过度编程检验不可能失败。

[0077] 在一个实施例中, 如果第三数据锁存器 406 中的低页数据为逻辑“0”且存储于第二数据锁存器 404 中的从步骤 802 的检验返回的结果为逻辑“0”, 那么在步骤 806 中判定存储器单元已被过度编程。此情况指示所述存储器单元应处于状态 00; 然而, 阈电压高于 V_{op00} 。如果低页数据为逻辑“1”或存储于第二数据锁存器 404 中的从步骤 802 的检验过程返回的结果为逻辑“1”, 那么证实所述存储器单元相对于状态 00 没有被过度编程。在步骤 806 中, 对于相对于状态 00 没有被过度编程的每一个存储器单元而言, 将相关的第二数据锁存器 404 设置为逻辑“1”。应注意的是, 如果期望高页编程过程将高页保持为逻辑 1 (例如, 状态 11 或状态 10), 那么各自的第二数据锁存器将已处于逻辑“1”。如果将所有的第二数据锁存器 404 设置为逻辑“1” (步骤 808), 那么图 12 的过程通过 (步骤 810)。如果并没有将所有的第二数据锁存器 404 都设置为逻辑“1”, 那么, 因为存在相对于状态 100 而被编程的一个或一个以上存储器单元, 所以图 12 的过程失败 (步骤 812)。

[0078] 图 13 是流程图, 其描述用于对状态 01 执行高页过度编程检验过程 (图 9 的步骤 670) 的过程的一个实例。在步骤 862 中, 使用比较点 V_{op01} 来执行检验过程。如果所述单元传导 (返回的检验结果为逻辑 1), 那么将第二数据锁存器 404 设置为逻辑“1”; 否则, 将第二数据锁存器 404 设置为逻辑“0”。如果将所有的第二数据锁存器设置为 1 (步骤 864), 那么图 13 的过程通过 (步骤 866)。如果并没有将所有第二数据锁存器设置为逻辑“1”, 那么图 13 的过程未通过 (步骤 868)。

[0079] 应注意的是, 上述流程图假设通过向控制栅极施加特定电压并判定所述存储器单元是接通还是关闭来执行检验过程。然而, 存在检验 (并读取) 存储器单元以判定状态的其它手段, 且可在本发明的精神内使用所述手段。举例来说, 可使用电流感测系统, 其中将处于某种读取条件下的存储器单元的电流与一组参考电流相比较。在另一实施例中, 可使用许多不同参数来判定存储器单元的状态。举例来说, 可通过电流感测来执行单元中所存储的电荷电平的判定, 其中使用固定的偏压条件来感测其传导的量值。或者, 可通过感测阈电压来判定, 其中使用多种操纵栅极偏压条件来感测此类传导的开始。或者, 通过使单元的电平判定驱动器强度来控制动态地固定的感测节点的放电速率 (例如通过预充电电容器), 可动态地执行所述判定过程。通过感测达到给定的放电电平的时间来判定所存储的电荷电平。在此种情况下, 能指示所述单元条件的参数是时间。在美国专利第 6, 222, 762 号中描述了此方法, 所述专利的全文以引用的方式并入本文中。另一替代技术是将频率用作参数来判定存储器单元的状态的技术, 如美国专利第 6, 044, 019 号中所描述, 所述专利的全文以引用的方式并入本文中。美国专利第 5, 172, 338 号中更全面地研究了电流感测方法, 所述专利的全文也以引用的方式并入本文中。

[0080] 因为相邻浮动栅极上的电荷可引起电场, 所以存储于浮动栅极上的表观电荷会发生移动。问题是在正被读取的当前单元被编程之后可同时编程或擦除相邻存储器单元。举例来说, 我们认为一组单元被编程以将一电荷电平添加到其浮动栅极以对应于一组数据。

在用第二组数据来编程第二组单元（与第一组单元相邻）之后，由于受到第二组浮动栅极的电场的影响，从第一组单元的浮动栅极读取的电荷电平有时候似乎与被编程的不同。也就是说，观察到的错误可归因于正被读取的单元的浮动的表观电压电平受到相邻浮动栅极的电场的影响。此通常称为 Yupin 效应，且在美国专利第 5,867,429 号和第 5,930,167 号中较充分地描述了所述 Yupin 效应，所述专利的全文以引用的方式并入本文中。这些专利描述使两组浮动栅极彼此间物理上绝缘或考虑读取第一组浮动栅极时第二组浮动栅极上的电荷所产生的影响。

[0081] 有可能正确地编程第一组存储器单元，但当接着编程相邻单元时，Yupin 效应便会导致第一组存储器单元似乎被过度编程。举例来说，对期望编程为状态 10（见图 8）的第一单元进行检查以检验所述单元的阈电压高于 V_{v10} （编程检验）并低于 V_{op10} （过度编程检验）。在编程相邻存储器单元之后，从所述第一单元读取的阈电压可能似乎高于原先编程的阈电压（例如，高出 2 伏特）。如果将第一存储器单元编程为正好低于 V_{op10} 的阈电压，那么在编程所述相邻单元之后第一单元具有高于 V_{op10} 的表观阈电压。因此，过度编程了第一存储器单元。所述表观阈电压可能高于读取比较点（例如， V_{r00} ），借此导致数据错误。应注意的是，邻近单元可能在相邻字线或相邻位线上。举例来说，在图 4 中，单元 244 与单元 224、242、250 和 246 相邻。在某些实施中，不与单元 244 相邻的存储器单元可能具有能影响从单元 244 读取的表观电荷的电场。

[0082] 为了解决 Yupin 效应，在使用第一控制线（例如，字线或位线）进行编程之后，本发明的一组实施例将在先前已被编程且与第一控制线相邻的控制线相关的页上执行过度编程检验检查。图 14A-15C 描述这些实施例。

[0083] 图 14A 描述在与先前选定字线相关的逻辑页上执行过度编程检验的过程的实施例。在图 14A 的步骤 902 中，使用第一字线来编程一组存储器单元。举例来说，参看图 4，可使用字线 WL1 来编程逻辑页。步骤 902 的过程包括执行图 9 的过程，如同上文关于图 10-13 的所描述的，所述过程包括检查过度编程的存储器单元。在步骤 904 中，使用第二字线（例如，图 4 的字线 WL2）对存储器单元执行第二组编程操作。步骤 904 的过程包括执行图 9 的过程，其包括检查过度编程的存储器单元。在步骤 906 中，系统检查由与第一字线（例如，字线 WL1）相关的逻辑页的存储器单元的上述 Yupin 效应所导致的过度编程。在步骤 908 中，系统将使用第三字线（例如，字线 WL3）来编程第三组存储器单元。步骤 908 的过程包括执行图 9 的过程，其包括检查过度编程的存储器单元。在步骤 910 中，系统检查由与第二字线（例如，WL2）相关的逻辑页的上述 Yupin 效应所导致的过度编程。然后，所述过程可继续（或不继续）编程与附加附加字线相关的存储器单元并为先前编程的字线检查过度编程的存储器单元。

[0084] 应注意的是，在某些实施中，沿 NAND 串顺序地编程字线；因此，在第一字线用于编程之后，下一编程操作使用与第一字线相邻的字线。通常（但并不总是）以从源极侧到漏极侧的顺序来编程 NAND 串，举例来说，从存储器单元 228 到存储器单元 222（见图 4）。因此，在使用 WL0 进行编程之后，系统接着将使用 WL1（与 WL0 相邻）。因此，在图 14A 的实例中，对最接近的先前字线执行过度编程检验。然而，在其它实施例中，可检查其它先前编程的存储器单元，而不是检查最接近的先前编程的存储器单元。

[0085] 图 14B 提供流程图，描述通过检查先前选定的位线的过度编程的存储器单元来解

决 Yupin 效应的一个实施例的过程。在一个实施例中,将位线分组为奇数字线和偶数字线。编程奇数字线,接着编程偶数字线,接着编程奇数字线,接着编程偶数字线等。在其它实施例中,不同组的位线可用于不同的编程操作。图 14B 期望:通过一组位线进行编程之后,相对于先前编程的其它组位线执行对过度编程检查。在步骤 950 中,使用偶数字线来执行编程操作。步骤 950 的过程包括执行图 9 的过程,如同上文关于图 10-13 所描述的,其包括检查过度编程的存储器单元。在步骤 952 中,使用奇数字线来执行编程操作。步骤 952 的过程包括执行图 9 的过程,其包括检查过度编程的存储器单元。在步骤 954 中,系统检查由与先前编程的偶数字线相关的逻辑页的上述 Yupin 效应所导致的过度编程。在步骤 956 中,使用偶数字线来执行编程。步骤 956 的过程包括执行图 9 的过程,其包括检查过度编程的存储器单元。在步骤 958 中,系统检查由与先前编程的奇数字线相关的逻辑页的上述 Yupin 效应所导致的过度编程。此过程可通过使编程与检查先前位线上的过度编程交替进行而得以继续。

[0086] 可由状态机、控制器、用于实施所述过程的特定逻辑或上述各项的组合来执行图 14A 和 14B 的过程。

[0087] 存在许多不同的适当方法,其中系统可使用所述方法来检查由与先前编程的控制线相关的逻辑页的上述 Yupin 效应所导致的过度编程(作为图 14A 和 B 的过程的一部分)。图 15A-C 描述了两种这些方法;然而,还可使用许多其它的适当方法。

[0088] 与图 8 类似,图 15A 说明存储两个比特的数据的存储器单元的阈电压分布 460、462、464 和 466。除读取比较点 (V_{r10} 、 V_{r00} 和 V_{r01})、检验比较点 (V_{v10} 、 V_{v00} 和 V_{v01}) 和过度编程比较点 (V_{opSP} 、 V_{op10} 、 V_{op000} 和 V_{op01}) 之外,图 15A 还描绘了过度编程测试点 V_{opA} 、 V_{opB} 和 V_{opC} 。在编程第一单元且接着编程相邻的第二单元之后,由第二单元引起的 Yupin 效应将使第一单元的阈电压升高约 0.2 伏特(可多于或少于 0.2 伏特,此取决于技术、尺寸、间距等)。所述数据并没有被完全破坏。相反,可预知地破坏所述数据。一种情况是:如果原始编程导致第一单元的阈电压正好低于过度编程检验电压(例如,正好低于 V_{op10})且所述单元接着经受最大的 Yupin 效应(例如,约 0.2 伏特,或上文所述的另一值)。因此,如果所述单元由于 Yupin 效应而变为过度编程,那么形成过度编程区域来表示何处为阈电压的预测。图 15A 显示三个过度编程区域。第一过度编程区域包括 V_{opSP} 与 V_{opA} 之间的区域。第二过度编程区域包括 V_{op10} 与 V_{opB} 之间的区域。第三过度编程区域包括 V_{op00} 与 V_{opC} 之间的区域。应注意的是,过度编程比较点与过度编程测试点的值可不同于图 15A 中所描绘的值。

[0089] 用于判定存储器单元是否由于 Yupin 效应而被过度编程的一个过程是测试所述单元的阈电压是否在所述过度编程区域中的一个内。如果单元的阈电压在过度编程区域内,那么认为所述单元会由于 Yupin 效应而被过度编程。举例来说,如果判定单元的阈电压在第二过度编程区域内,那么认为所述单元应已处于分布 462 内,但是由于 Yupin 效应而变得过度编程。应注意的是,尽管图 15A 描绘四个阈分布(存储两个比特的数据),但是当实施本发明时还可使用更多或更少的分布(和比特的数据)。

[0090] 图 15B 提供流程图,其描述测试如上文所述的由于 Yupin 效应而导致过度编程的过程的一个实施例。图 15B 的过程可作为图 14A 和 14B 的步骤 906、910、954 和 958 中的检查过度编程的一部分而执行。在步骤 1002 中,在每一个过度编程区域中的边缘处执行读取

操作。举例来说,在 VopSP、VopA、Vop10、VopB、Vop00 和 VopC 处来执行读取操作。基于所述读取操作,判定所述单元的阈电压是否在任意一个过度编程区域之内。举例来说,如果当将 Vop10 施加到字线时所述存储器单元接通而当将 VopB 施加到字线时所述存储器单元不接通,那么可认为所述单元在第二过度编程区域中。如果所述存储器单元的阈电压在所述过度编程区域中的一个内(步骤 1006),那么推断所述单元由于相邻单元的耦接(Yupin 效应)而被过度编程(步骤 1008)。在步骤 1010 中,将过度编程的单元的数据固定。可部分地擦除所述单元以将阈电压降回到适当分布中,可完全擦除并重写所述单元,可将所述数据写入到另一位置,可存储错误信息,其中所述错误信息指示所述单元被过度编程使得随后的读取将考虑过度编程,或可执行另一适合措施。

[0091] 如果在步骤 1006 中判定所述阈电压不在所述过度编程区域中的任意一个中,那么对过度编程比较点 Vop01 执行读取操作。如果所述存储器单元的阈电压高于 Vop01(步骤 1022),那么认为所述存储器单元将被过度编程且所述过程将在步骤 1008 处继续。如果所述存储器单元的阈电压低于 Vop01(步骤 1022),那么认为所述存储器单元没有被过度编程(步骤 1024)。

[0092] 图 15C 提供流程图,其描述测试由于 Yupin 效应而导致过度编程的过程的另一实施例。图 15C 的过程可作为图 14A 和 14B 的步骤 906、910、954 和 958 中的检查过度编程的一部分而执行。在步骤 1060 中,在每一个读取比较点 Vr10、Vr00 和 Vr01 处来执行读取操作。基于所述读取操作,做出关于所述存储器单元的状态的初始判定。在步骤 1062 中,根据所属领域中的已知方法来执行 ECC 过程。如果不存在 ECC 错误(步骤 1064),那么认为所述存储器单元并没有由于 Yupin 效应而被过度编程(步骤 1066)。如果检测到 ECC 错误(步骤 1064),那么认为所述错误是由于 Yupin 效应而被过度编程的存储器单元所引起的(步骤 1070)。在步骤 1072 中,将数据固定(与图 15B 的步骤 1010 类似)。举例来说,假设在步骤 1060 中,读取操作导致判定所述阈电压高于 Vr00 且低于 Vr01。假设所述存储器单元在存储有逻辑数据 00 的阈分布 464 中。如果 ECC 失败,那么认为所述存储器单元应在存储逻辑数据 10 的阈分布 462 中;然而,所述存储器单元被过度编程。因此,在步骤 1072 的一个实施例中,可将所述存储器单元的阈电压校正到阈分布 462 内。

[0093] 图 17-22 提供真值表,其结合图 16 描述执行过度编程检验的替代实施例。与图 8 类似,图 16 说明存储两个比特的数据的存储器单元的阈电压分布 460、462、464 和 466。然而,图 16 显示不同的过度编程检验比较点 VopSP'、Vop10'、Vop00' 和 Vop01'。比较点 VopSP' 是稍高于阈电压分布 460 中的最高电压的值。Vop10' 是稍高于阈电压分布 462 中的最高电压的值。Vop00' 是稍高于阈电压分布 464 中的最大电压的值。Vop01' 是稍高于阈电压分布 466 中的最大电压的值。应注意的是,图 16 的 VopSP' 低于图 8 的 VopSP,图 16 的 Vop10' 低于图 8 的 Vop10,图 16 的 Vop00' 低于图 8 的 Vop00,且图 16 的 Vop01' 低于图 8 的 Vop01。在其它实施例中,还可使用高于各自的阈电压分布的检验比较点 VopSP'、Vop10'、Vop00' 和 Vop01' 的其它值。

[0094] 图 17 解释在低页编程之后来执行软件编程过度编程检验的一个实施例。图 17 的真值表具有两个列,其对应于在之前的低页编程过程中是编程数据"0"还是编程数据"1"。所述真值表的第二行(在 VopSP' 处读取)对应于从正被检验过度编程的存储器单元读取的数据。第三行指示存储于第二数据锁存器 404 中的数据。第四行指示存储于

第一数据锁存器 402 中的数据。第五行指示状态 11 的过度编程检验的结果。

[0095] 如果编程“1”（图 17 的左侧列），那么所述存储器单元保持于状态 11 内。如果编程“0”，那么期望所述单元被编程为状态 10。所述真值表的第二行指示读取操作（包括将 $V_{opSP'}$ 施加到存储器单元的控制栅极）的结果。如果正被读取的存储器单元的阈电压高于 $V_{opSP'}$ ，那么所述存储器单元将不接通且将读取逻辑 0。如果所述存储器单元的阈电压低于 $V_{opSP'}$ ，那么所述存储器单元将接通且将读取逻辑 1。在对应于正被编程到所述存储器单元中的逻辑 1 的列中，读取步骤将读取逻辑“1”（适度编程）或逻辑“0”（过度编程）。在所述单元被编程为状态 10 的情况下，因为在编程过程中检验步骤检验到所述电压至少高于 V_{v10} ，所以读取步骤将读取逻辑“0”。将读取的数据存储于第二数据锁存器 404 中。第一数据锁存器 402 存储先前被编程到低页中的数据。状态机将检查第一数据锁存器 402 和第二数据锁存器 404 的内容。如果第一数据锁存器 402 正存储逻辑“1”且第二数据锁存器 404 正存储逻辑“0”，那么过度编程测试失败。也就是说，状态机推断出所述存储器单元被过度编程。

[0096] 图 18 是真值表，其描述在已执行低页与高页编程之后来执行软件编程过度编程检验的实施例。为适应此任务，需要考虑低页数据和高页数据。因为已执行高页和低页编程操作，所以所述真值表包括所有四个状态（11、10、00 和 01）的列。所述真值表的第二行指示在将 $V_{opSP'}$ 施加到控制栅极的同时从所述单元读取的数据。如果期望所述单元处于状态 11，那么读取操作将返回 1（如果所述单元没有被过度编程）和返回 0（如果所述单元被过度编程）。如果所述单元处于其它三个状态中的任意一个状态，那么应返回 0，因为在编程过程中先前的检验步骤已确保所述单元的阈电压充分地高于适当检验点。所述真值表的第三行指示第二锁存器 404 的内容，其中所述第二锁存器 404 存储前述读取操作的结果。第四行指示第三数据锁存器 402 的内容。如同上文所描述的，第三数据锁存器存储有低页数据。第五行指示状态 11 的过度编程检验的结果。状态机检查以判定第三数据锁存器 406 是否存储逻辑 1 且第二数据锁存器 404 是否存储逻辑 0。如果是这样，那么，因为存储器单元被过度编程，所以过度编程测试失败。如果不是这样，那么测试通过。

[0097] 在另一实施例中，系统接着在 V_{r10} 处执行另一读取，而不是使用先前存储于第三数据锁存器中的数据。此读取操作能判定所述单元的阈电压是否高于 $V_{opSP'}$ 且低于 V_{r10} 。如果是这样，那么状态机推断出所述存储器单元被过度编程；否则，状态机认为所述存储器单元没有被过度编程。

[0098] 图 19 是真值表，其描述检验将单元编程为状态 10 的编程过程是否过度编程所述单元的过程。在低页编程过程之后且在高页编程过程之前来执行与图 19 相关联的检验过程。所述真值表具有两个列，一个列对应于当逻辑“1”被编程到低页时的情况而另一个列对应于逻辑“0”被编程到低页时的情况。第二行指示来自将 $V_{op10'}$ 施加到控制栅极的读取操作的数据。如果期望存储器单元保持于状态 11 中或所述存储器单元被正确地编程到状态 10，那么 $V_{op10'}$ 处的读取操作将导致所述存储器接通并报告逻辑“1”。将读取操作的结果存储于第二数据锁存器 404 中（见真值表的第三行）。如果期望所述单元被编程到状态 10 并被过度编程，那么，因为所述单元将不接通，所以 $V_{op10'}$ 处的读取操作将导致逻辑“0”。第一数据锁存器 402 存储被编程的数据（见第四行）。如果逻辑“1”被编程（例如，没有进行任何编程），那么第二数据锁存器 404 与第一数据锁存器 402 存储逻辑

1 且测试将通过（见第五行）。如果适度地编程逻辑“0”，那么第一数据锁存器 404 将存储逻辑 0、第二数据锁存器 402 将存储逻辑 1 且测试将通过。如果第二数据锁存器 404 处于逻辑 0 且第一数据锁存器 402 处于逻辑 0，那么，因为所述单元被过度编程，所以测试失败。

[0099] 图 20 是真值表，其描述在编程低页与较高页之后执行状态 10 过度编程检验的过程。因为两种页均被编程，所以存在四个列，一个列用于可将单元编程为的每一个状态—11、10、00 和 01。第二行指示来自将 V_{op10}' 施加到控制栅极的读取操作的数据。将所述读取操作的结果存储于第二数据锁存器 404 中（第三行）。如果存储器单元保持于状态 11 中或被正确地编程到状态 10，那么所述存储器单元将接通且将读取逻辑“1”。如果期望所述单元被编程到状态 10，但被过度编程了，那么所述单元将不接通且返回逻辑“0”。如果所述单元在高页编程过程中被编程到状态 00 或状态 01，那么所述单元将不接通且读取逻辑“0”。第一数据锁存器 402 将存储来自最近执行的高页编程操作的数据（第四行）。举例来说，如果所述单元处于状态 11 或 10，那么第一数据锁存器将存储逻辑“1”。如果所述单元处于状态 00 或 01，那么第一数据锁存器将存储逻辑“0”。接着系统将进行测试以判定第一数据锁存器 402 是否存储逻辑 1 且第二数据锁存器 404 是否存储逻辑 0。如果是这样，那么，因为所述存储器单元被过度编程，所以检验过程失败。否则，检验过程通过（第五行）。

[0100] 图 21 是真值表，其描述执行状态 00 过度编程检验的过程。在高页和低页编程之后执行检验状态 00 的过程；因此，图 21 的真值表显示四个列（一个列用于可将单元编程为的每一个状态）。首先，执行读取操作，所述读取操作包括将 V_{op00}' 施加到控制栅极（见所述真值表的第二行）。如果所述单元被正确地编程为状态 11、10 或 00，那么所述单元将接通且返回逻辑“1”。如果所述单元编程为状态 01，那么所述单元将不接通且返回逻辑“0”。如果期望所述单元编程为状态 00 但被过度编程了，那么所述单元将不接通且将返回逻辑“0”。将从读取操作返回的数据存储于第二数据锁存器 404 中（见真值表的第三行）。第四行指示第三数据锁存器 402 的内容。如同上文所描述的，第三数据锁存器存储低页数据。第五行指示状态 00 的过度编程检验的结果。状态机进行检查以判定第三数据锁存器 406 是否存储逻辑“0”且第二数据锁存器 404 是否存储逻辑 0。如果是这样，那么，因为所述存储器单元被过度编程，所以过度编程测试失败。如果不是这样，那么测试通过。

[0101] 在另一实施例中，系统接着在 V_{r01} 处执行另一读取，而不是使用先前存储于第三数据锁存器中的数据。此读取操作判定所述单元的阈电压是否高于 V_{op00}' 且低于 V_{r01} 。如果是这样，那么状态机推断出所述存储器单元被过度编程；否则，状态机认为所述存储器单元没有被过度编程。

[0102] 图 22 是真值表，其描述执行状态 01 过度编程检验的过程。因为在高页读取之后来执行所述过程，所以存在四个列，一个列用于可将单元编程为的每一个状态。图 22 的第二行指示来自包括将 V_{op01}' 施加到控制栅极的读取操作的结果。将读取操作所产生的数据存储于第二数据锁存器 404 中（第三行）。不管单元处于何种状态，所述读取操作所产生的数据应总是为逻辑 1。因此，如果返回逻辑“0”，那么检验步骤失败；否则，检验步骤通过（第四行）。

[0103] 应注意的是，图 9 的流程图期望如果执行低页编程操作，那么执行软件编程检验

和状态 10 过度编程检验,且如果执行高页编程操作,那么执行状态 00 过度编程检验且执行状态 01 过度编程检验。还可与上文关于图 17、19、21 和 22 所讨论的过程一起使用此方法。在一个替代实施例中,可在已写入低页和高页之后使用上文关于图 18、20、21 和 22 的真值表所讨论的过程来执行所有的过度编程检验。在另一替代实施例中,在执行低页编程过程之后,可执行上面关于图 17 所讨论的检验过程。当编程高页时,系统将使用图 20、21 和 22 的真值表来对状态 10、00 和 01 执行过度编程检验。

[0104] 上文提供了关于 NAND 型快闪存储器的上述实例。然而,可将本发明的原则应用于其它类型的快闪存储器和其它类型的非易失性存储器,包括所述现存和所述正在开发的期待着使用的新颖技术的快闪存储器和存储器。

[0105] 本发明所提供的前述详细描述是用于说明和描述的目的。并不期望所述描述是穷尽的或将本发明限制为所揭示的形式完全一致。可根据以上教示做出许多修改和变化。经过选择的所描述的实施例是为了最好地解释本发明的原理和其实际应用,借此使其它所属领域技术人员能够在各种实施例中最好地利用本发明并作出各种修改以使其适合于所期待的特殊用途。我们期望本发明的保护范围由所附的权利要求书限定。

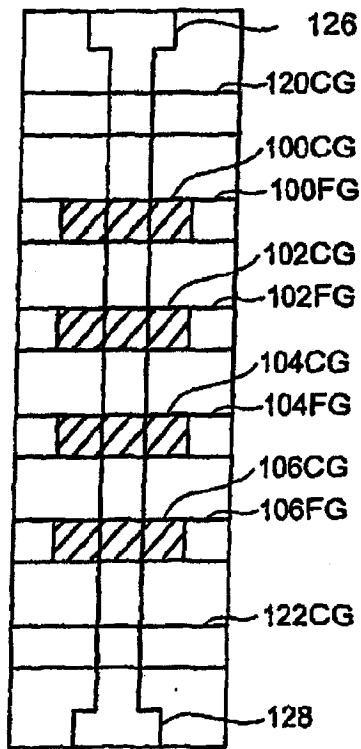


图1

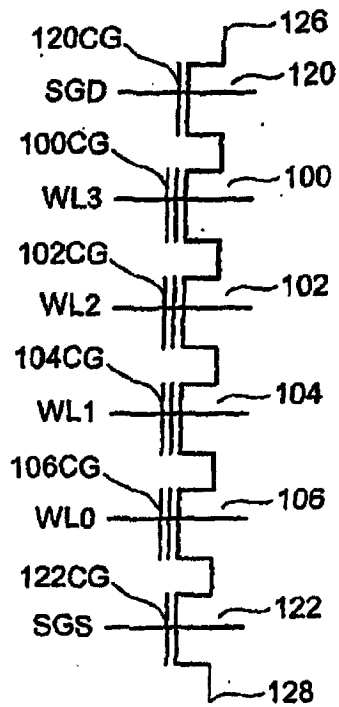


图2

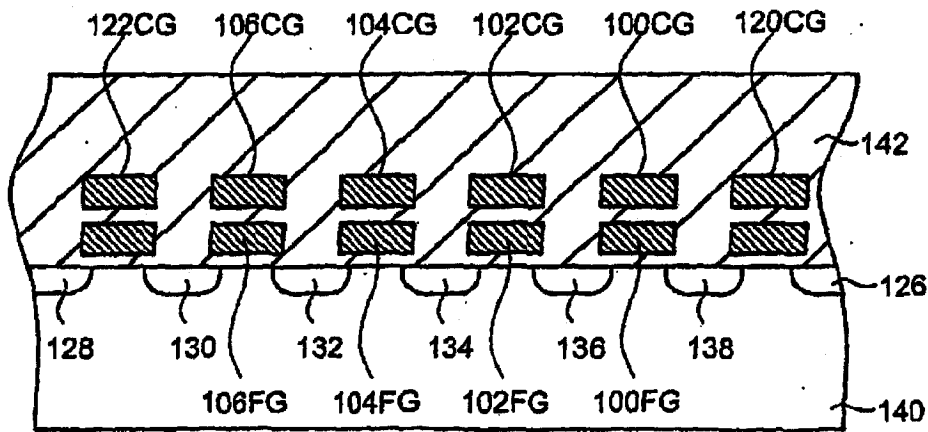


图3

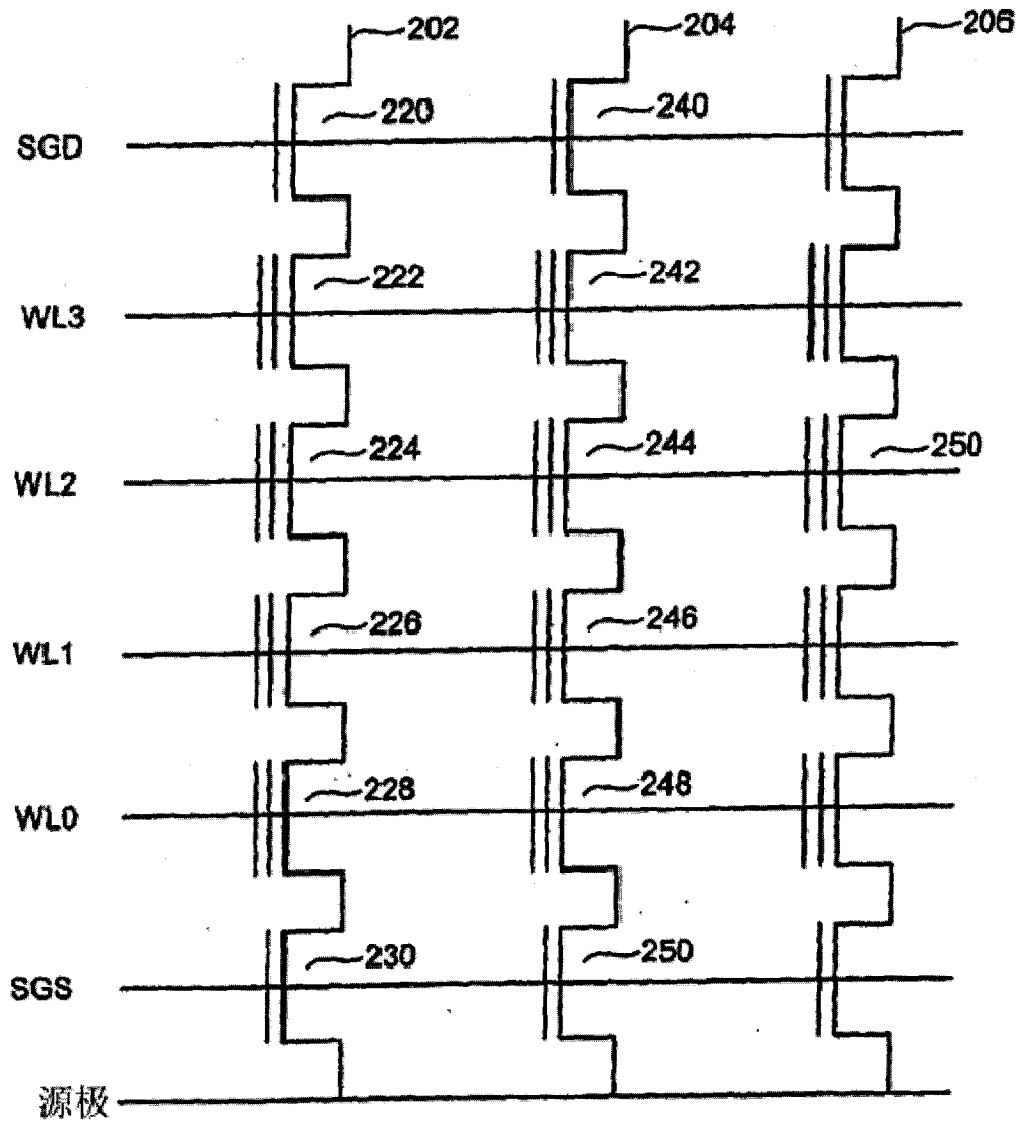


图4

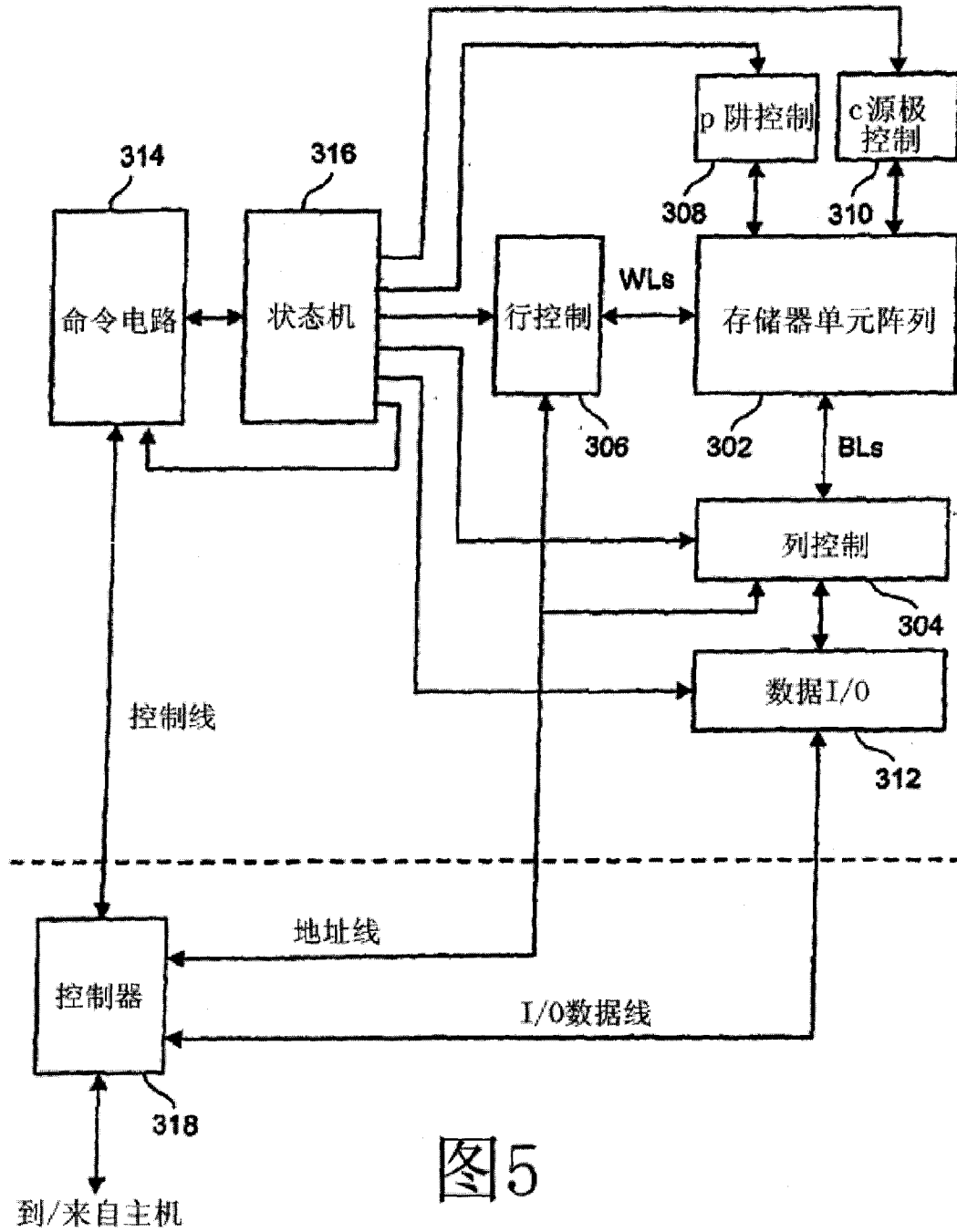


图5

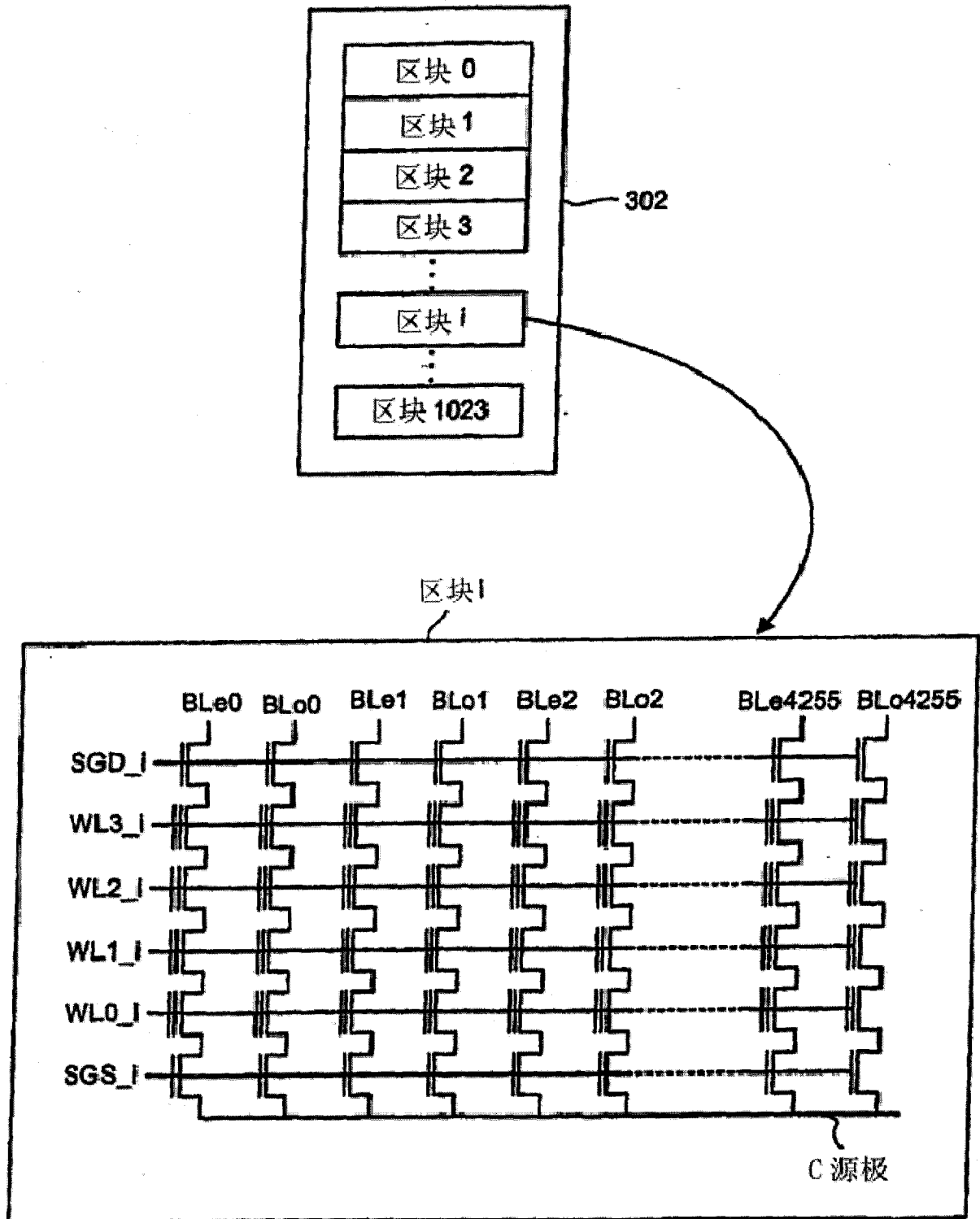


图6

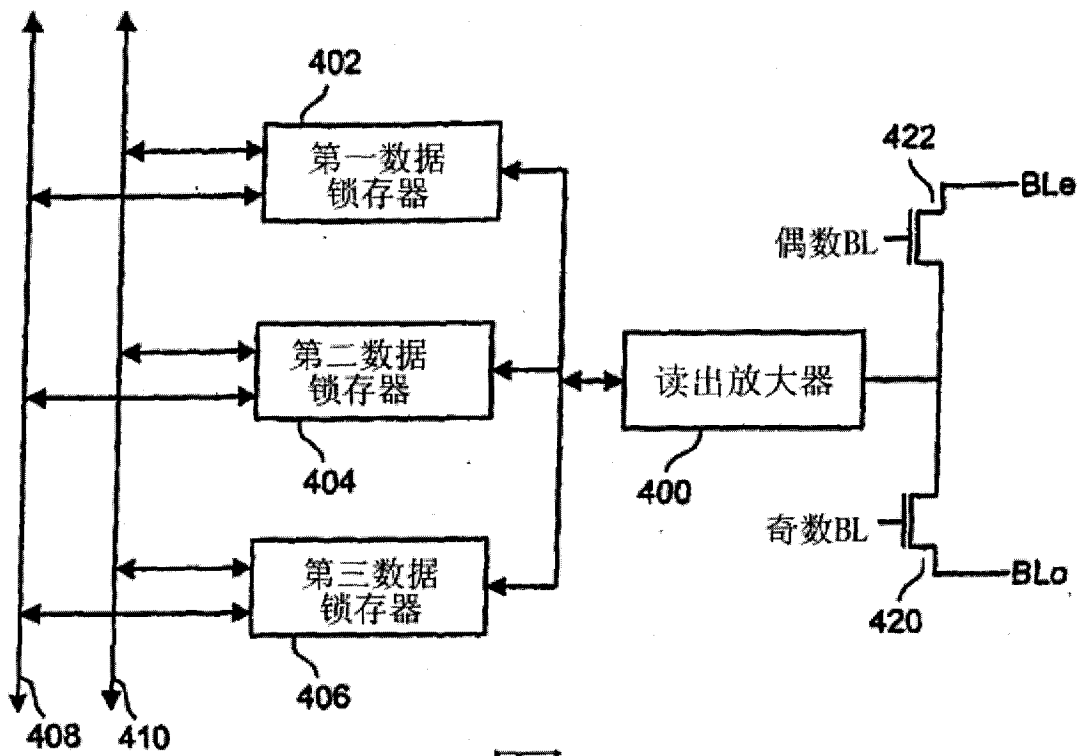


图7

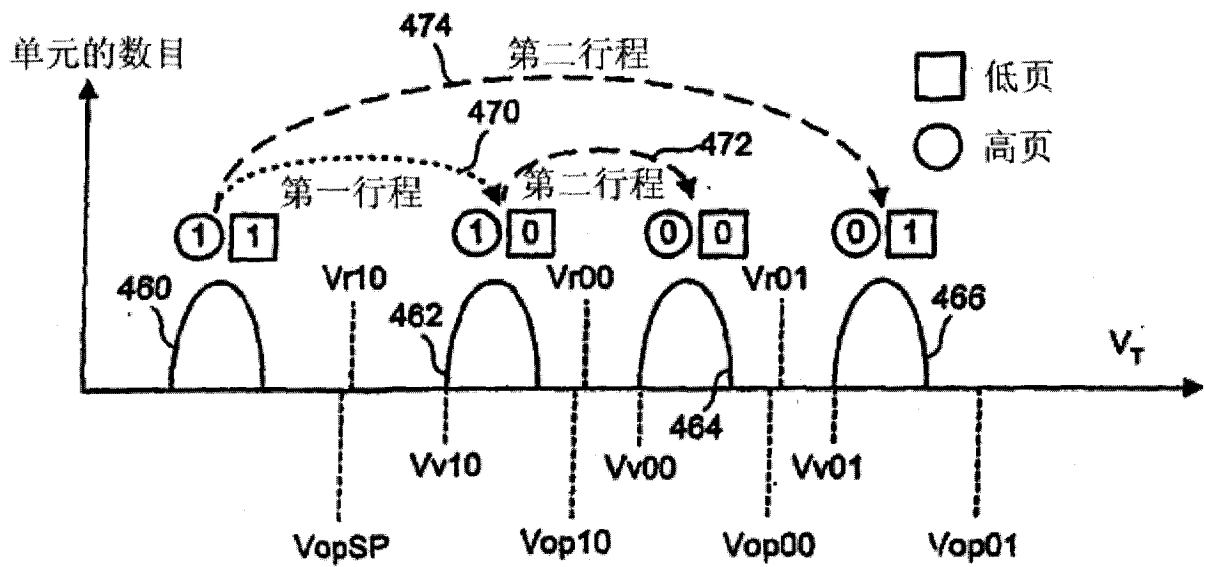


图8

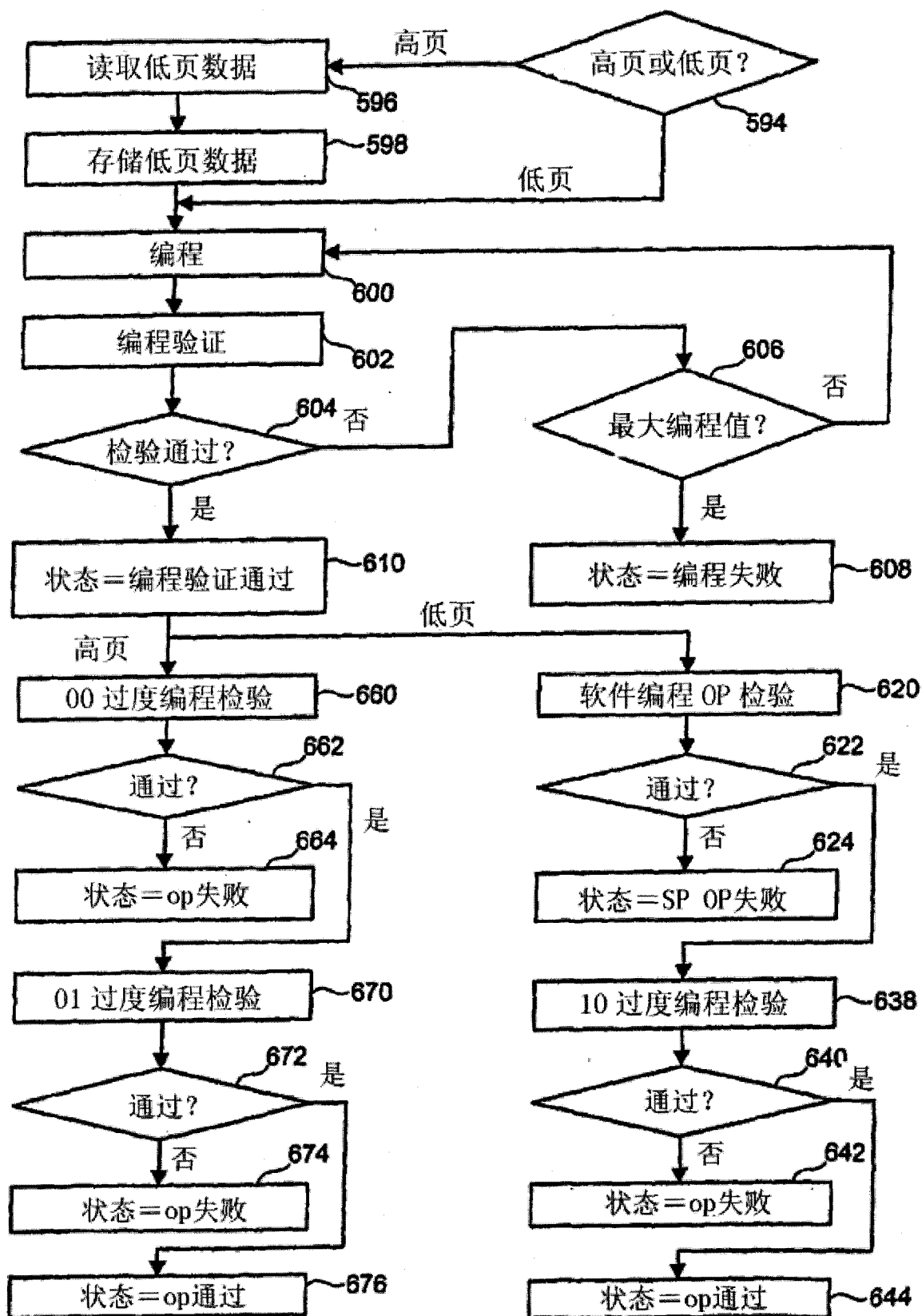


图9

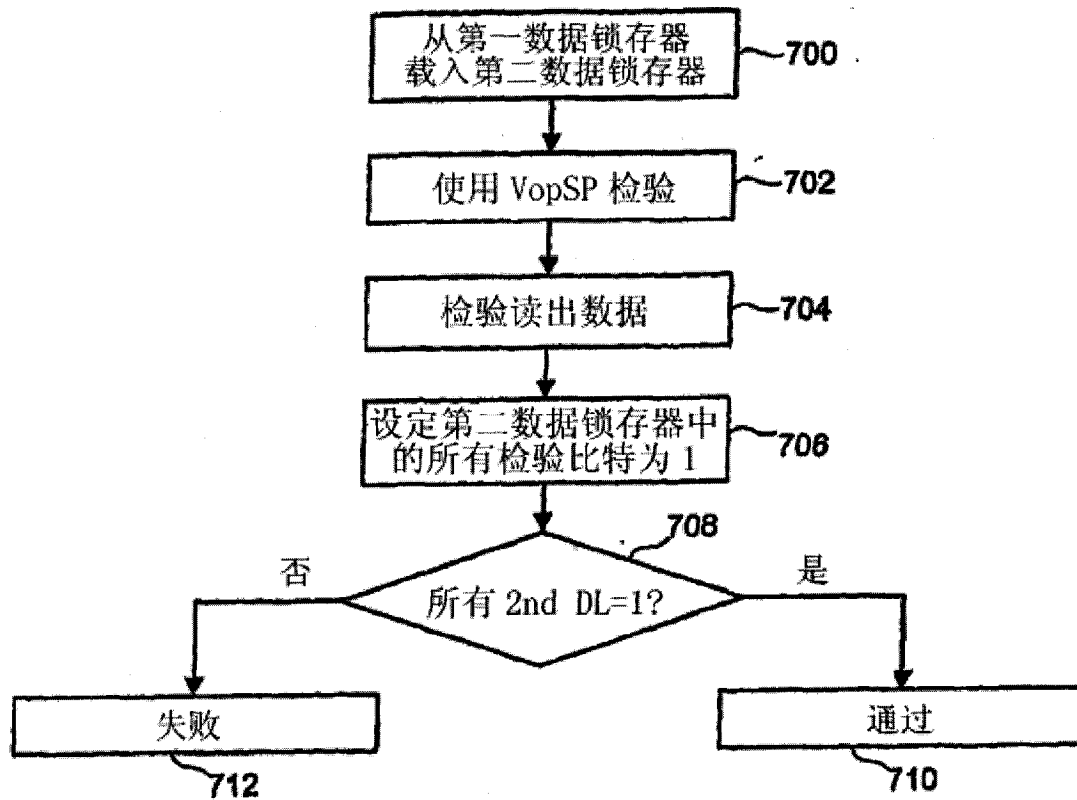


图 10

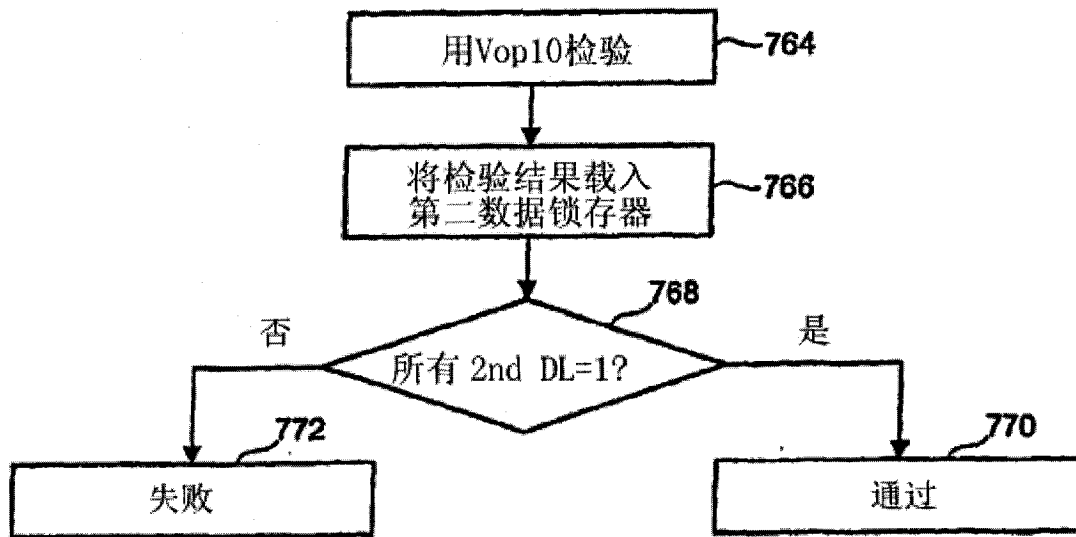


图 11

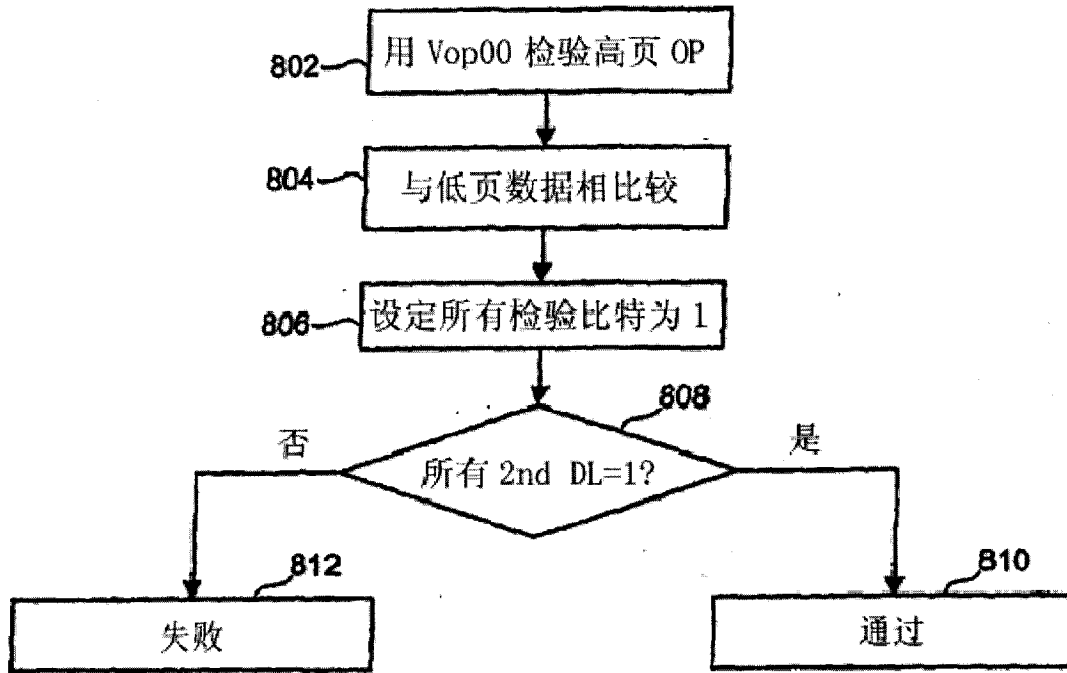


图12

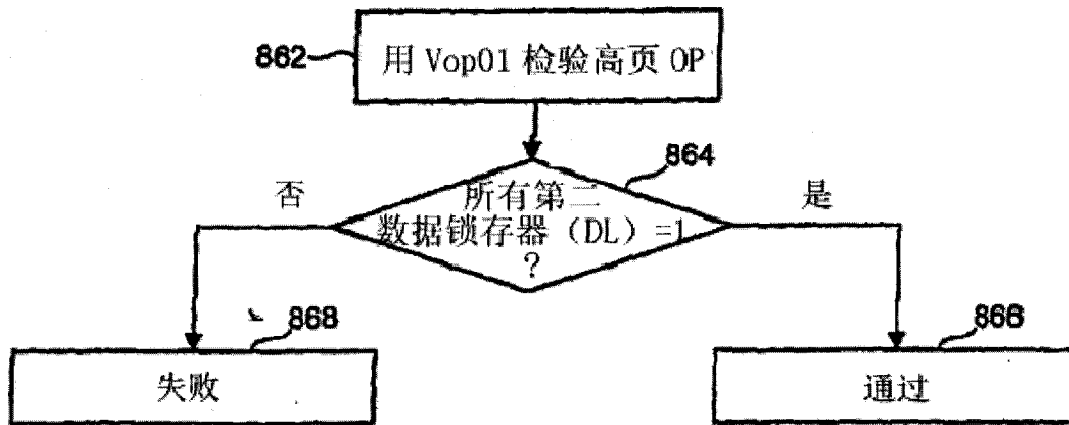


图13

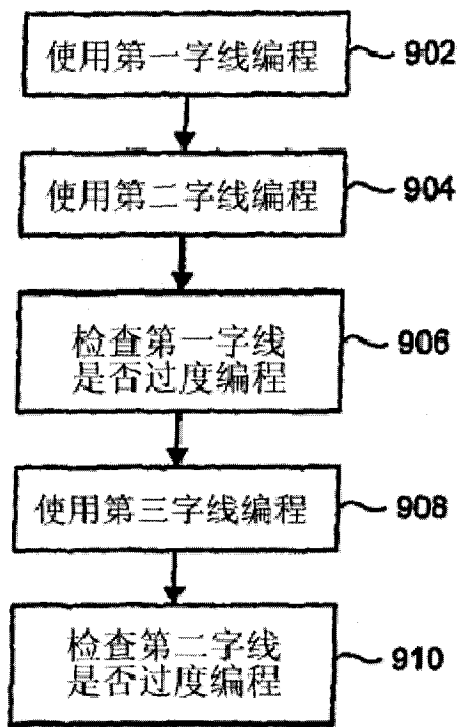


图14A

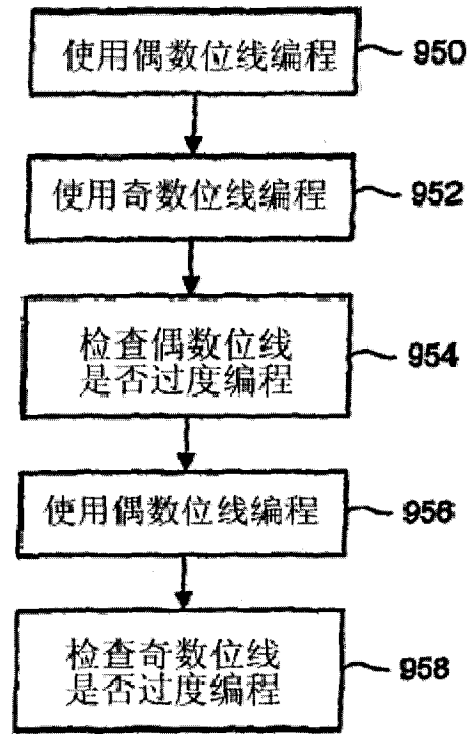


图14B

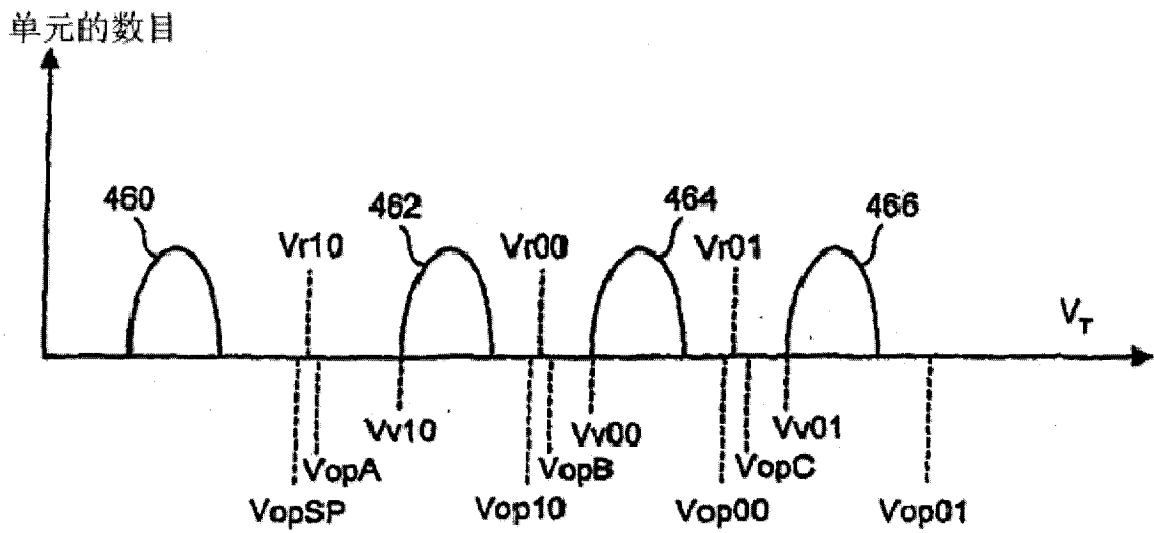


图 15A

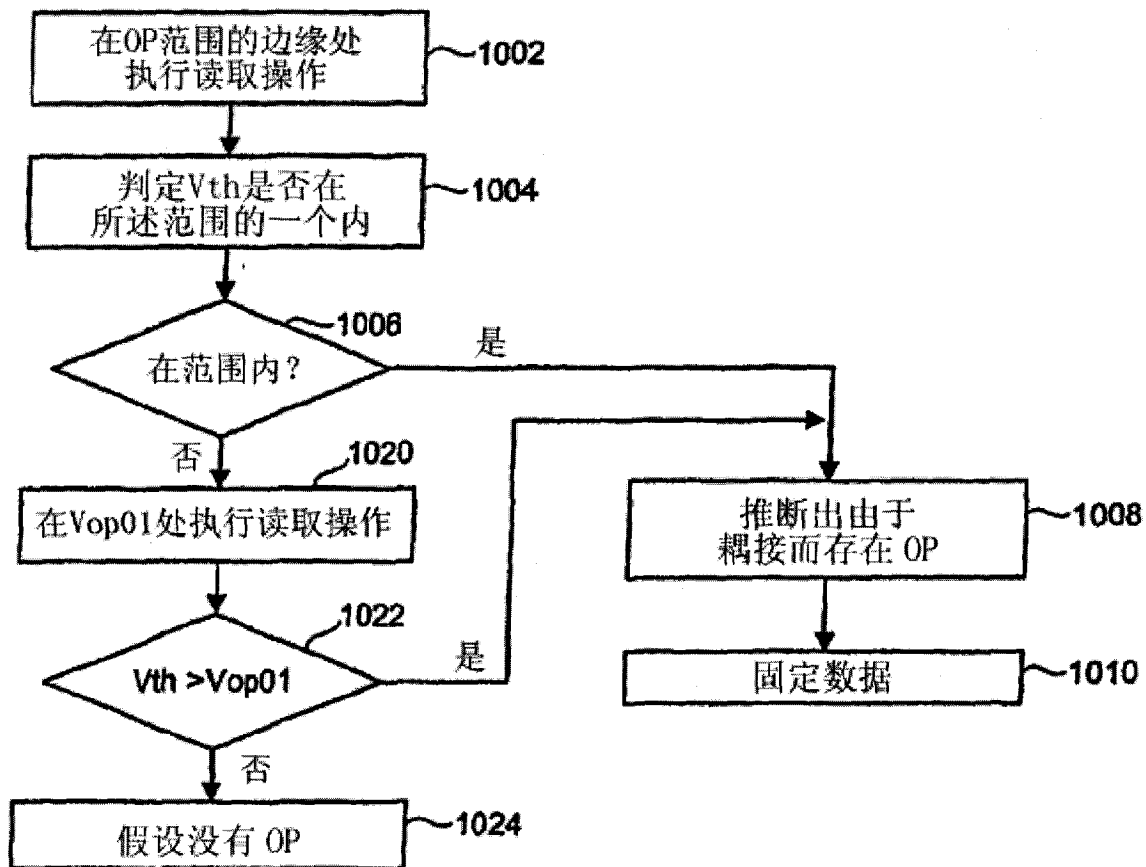


图15B

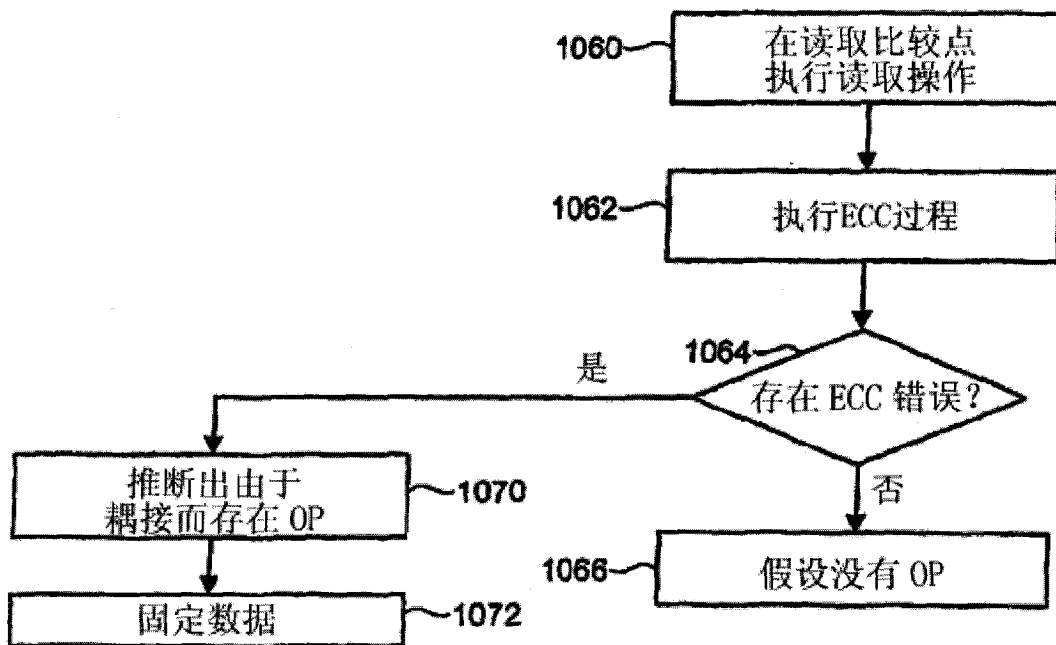


图15C

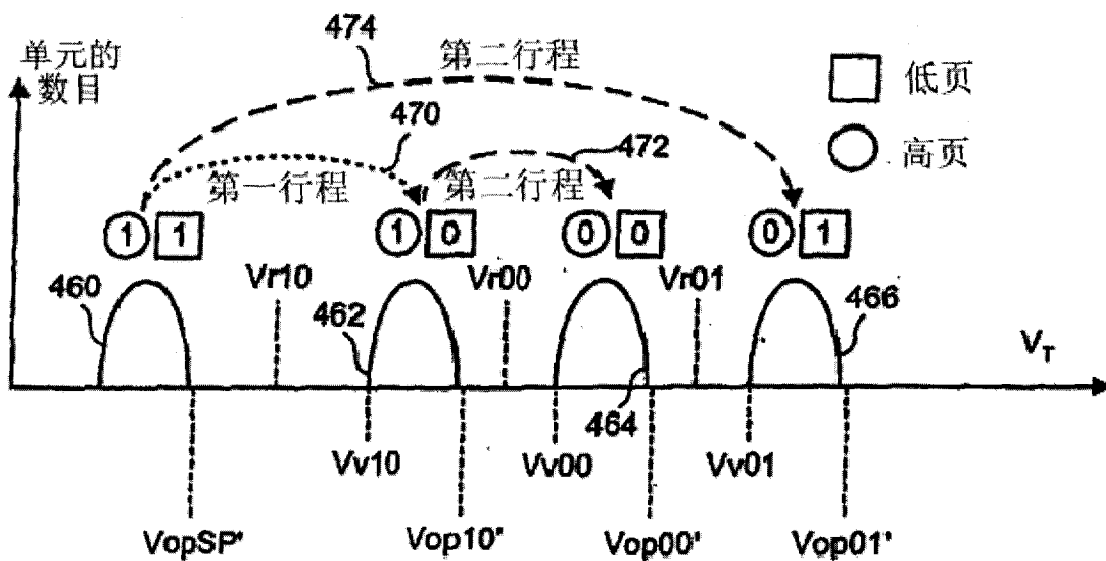


图16

	1	0
在VopSP 处读取	1/0	0
第二数据锁存器	1/0	0
第一数据锁存器	1	0
测试	通过/失败	通过

图17

	11	10	00	01
在VopSP 处读取	1/0	0	0	0
第二数据锁存器	1/0	0	0	0
第三数据锁存器	1	0	0	0
测试	通过/失败	通过	通过	通过

图18

	1	0
在Vop10 处读取	1	1/0
第二数据锁存器	1	1/0
第一数据锁存器	1	0
测试	通过	通过/失败

图19

	11	10	00	01
在Vop10 处读取	1	1/0	0	0
第二数据锁存器	1	1/0	0	0
第一数据锁存器	1	1	0	0
测试	通过	通过/失败	通过	通过

图20

	11	10	00	01
在Vop00 处读取	1	1	1/0	0
第二数据锁存器	1	1	1/0	0
第三数据锁存器	1	0	0	1
测试	通过	通过	通过/失败	通过

图21

	11	10	00	01
在Vop01 处读取	1	1	1	1/0
第二数据锁存器	1	1	1	1/0
测试	通过	通过	通过	通过/失败

图22