



[12] 发 明 专 利 说 明 书

[21] ZL 专利号 97194257.9

[45] 授权公告日 2003 年 10 月 29 日

[11] 授权公告号 CN 1126256C

[22] 申请日 1997.4.28 [21] 申请号 97194257.9

[30] 优先权

[32] 1996. 4. 30 [33] US [31] 08/640,367

[86] 国际申请 PCT/US97/07152 1997.4.28

[87] 国际公布 WO97/41640 英 1997.11.6

[85] 进入国家阶段日期 1998.10.30

[71] 专利权人 阿加特半导体公司

地址 美国加利福尼亚州

[72] 发明人 乔治·J·科尔什

萨卡瓦特·M·汗

审查员 段成云

[74] 专利代理机构 中原信达知识产权代理有限公司

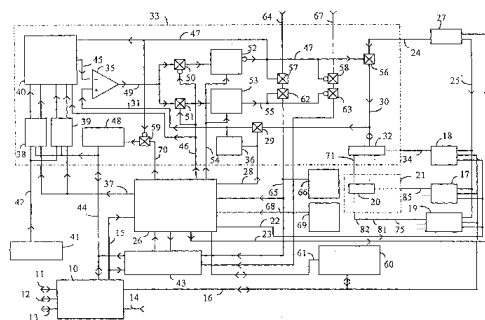
代理人 袁炳泽

权利要求书 4 页 说明书 40 页 附图 23 页

[54] 发明名称 用于多数字位的稳定电路

[57] 摘要

一种具有每一存储单元能够存储多个位的存储单元的集成电路存储系统。具有一种恢复操作，其中存储单元存储的可能会从它的初始设定条件漂移的电荷被保持在多个预定电平中之一内，多个电平对应于信息的数字位并且由一组特定的参考电压值定义。存储系统具有小编程和小擦除操作以仅把足以保持电荷进入和移出预定电平的电荷量加入或移出存储单元。存储系统还有一用于高速存储单元编程和一擦除操作，以使电荷分布变窄，增加电平间的分布和安全边界。



1. 一种集成电路存储系统包括：

5 多个存储单元，每个存储单元可以存储对应于信息位的多个离散状态中的一个，每一离散状态具有一预选的范围；和
用于读出所述存储单元超出所述一种离散状态的预选范围的漂移的装置。

10 2. 如权利要求 1 所述的存储系统，其中所述读出装置基本上同时地并与所选的多个存储单元无关地运作。

3. 如权利要求 1 所述的存储系统还包括用于基于所述存储单元超出所述预选范围的漂移恢复所述一离散状态的装置，所述恢复装置响应所述读出装置。

15 4. 如权利要求 1 所述的存储系统，其中每个存储单元包括一个在浮栅上存储对应于所述信息位的电荷量的非易失性存储单元；并且还包括：

20 用于响应所述电荷量在所述多个存储单元的所述每个浮栅中保持所述电荷量的电路装置；

因而避免了每个存储单元中的所述信息位的丢失。

25 5. 如权利要求 4 所述的存储系统，其中所述保持电路装置递增地向每个非易失性存储单元增加电荷或从每个非易失性存储单元减少电荷以调节所述电荷量，而不首先从所述非易失性存储单元实际上擦除所有电荷。

30 6. 如权利要求 4 所述的存储系统，其中在每个存储单元中的所述信息位包括至少两个位。

7. 如权利要求 1 所述的存储系统，还包括：

用于产生多于 2^N 个参考值的装置；和其中

35 每个存储单元能够存储对应于 N 个信息位的 2^N 个离散状态中的一个；和所述读出装置通过与所述参考值比较读出所述存储单元中的所述一种离散状态和通过参考所述参考值确定超出所述一种离散状态

的预选范围的漂移。

5 8. 如权利要求 7 所述存储系统, 其中所述读出装置和所述参考值产生装置协同操作依次地把所述一种离散状态与一个定序的序列中的所述参考值比较, 以确定所述一种离散状态, 和参考所述参考值确定来自所述一种离散状态的所述预选范围的所述漂移。

10 9. 如权利要求 7 所述的存储系统, 其中所述参考值包括第一和第二组, 所述读出装置响应于所述第一组参考值以读出所述一种离散状态和所述读出装置响应所述第二组参考值以确定来自所述存储单元中的所述一种离散状态的所述预选范围的漂移, 所述第二组参考值具有与所述第一组参考值的一种预定的关系。

15 10. 如权利要求 9 所述的存储系统还包括响应所述读出装置用于恢复在所述一种离散状态中的所述存储单元的装置, 使得对应于所述一种离散状态的信息保持在所述多个存储单元的每一个中。

20 11. 如权利要求 7 所述的存储系统, 其中每个存储单元存储指示所述一种离散状态的一定量的电荷; 和其中所述读出装置确定是否要向所述存储单元增加电荷或从所述存储单元移出电荷以保持指示所述一种离散状态的电荷; 并且还包括:

响应于所述读出装置的确定用于设置至少一个状态标志的装置。

25 12. 如权利要求 11 所述的存储系统, 还包括:
用于存储具有关于所述存储单元的存储位置的所述至少一种状态标志的装置。

30 13. 如权利要求 11 所述的存储系统, 其中所述读出装置把所述一种离散状态与至少两个所述参考值比较以确定是否有过多和不足的电荷已经加到所述存储单元中以恢复所述一种离散状态。

35 14. 如权利要求 1 所述的存储系统, 其中所述多个存储单元排列成阵列, 每个存储单元存储在对应于所述信息位的预选范围内的一离散量的电荷; 并且还包括:

一个耦合于所述存储阵列的编程电路, 所述电路向所述存储单元阵列提供信号以向所述存储单元移送或从所述存储单元移走电荷, 因

而使存储在所述存储单元中的电荷对应于所述信息位；

一个能够产生多个参考电压的参考电压电路；

耦合于所述存储单元和所述参考电压电路的比较器；和

5 耦合于所述存储单元阵列、所述参考电压电路和所述比较器的控制电路，所述控制电路把所述存储单元阵列中选定的多个存储单元和所述参考电压电路连接到比较器以读出电荷量，和确定所述电荷量中的漂移，所述电荷量对应于所述多个选定存储单元每一个中的信息位。

10 15. 如权利要求 14 所述的集成电路，其中所述控制电路响应于漂移确定结合所述编程电路以调节所述多个选定存储单元中的电荷，因而避免了每个存储单元中信息位的丢失。

15 16. 如权利要求 15 所述的集成电路，其中所述编程电路调节所述电荷而不首先从所述多个存储单元实际上擦除所有电荷。

20 17. 如权利要求 14 所述的集成电路，其中所述比较器响应所述选定多个存储单元每一个中的所述电荷量，把一个电压与所述多个参考电压中的一个第一组参考电压比较，以确定所述选定多个存储单元每一个中的所述数字信息；和其中

所述比较器将响应于所述多个选定存储单元每一个中的所述电荷量的所述电压与所述多个参考电压中的第二组参考电压比较，以确定在对应于所述数字信息的所述预选范围外的所述电荷量中的漂移；和其中

25 所述控制电路设置指示在对应于所述数字信息的所述预选范围外的所述电荷量中的所述漂移的所述确定的状态标志。

18. 一种操作具有多个存储单元的集成电路存储系统的方法，包括：

30 在每个存储单元中存储多个离散量中一个离散量的电荷，每个离散量对应于信息位，并具有一预选范围；和

响应来自所述一个离散量的预选范围的漂移来恢复所述存储单元中的电荷；

因而避免了所述存储单元中信息位的丢失。

35 19. 如权利要求 18 中所述的方法还包括：

响应于漂出所述一离散量的所述预选范围的漂移，恢复所述存储单元中的电荷。

20. 如权利要求 19 所述的方法，其中所述读出步骤包括：

5 将一个对应于所述电荷量的值与一个参考值的所述第一序列反复比较，在一个比较步骤中的一个参考值取决于一个以前比较步骤的结果；和还包括：

对照参考值的第二序列确定每个存储单元中所述电荷量中的漂移；和

10 如果这种漂移确定步骤是肯定的，设置至少一个指示漂移方向的状态标志。

21. 如权利要求 20 所述的方法，还包括存储具有与所述存储单元的存储位置相关的所述至少一种状态标志。

15

22. 如权利要求 20 所述的方法，其中所述漂移确定步骤包括：

把对应于所述电荷量的所述值与参考值的所述第二序列反复比较，一个比较步骤中的一个参考值取决于一个以前比较步骤的结果。

20 23. 如权利要求 22 所述的方法，其中与参考值的所述第二序列的所述反复比较步骤中的每一个步骤是在与参考值的所述第一序列的所述反复比较步骤中的一个步骤后进行的。

24. 如权利要求 20 所述的方法，还包括把对应于所述电荷量的所述值与一个大于或小于所述参考值序列中任何一个的参考值比较，取决于所述读出步骤确定了所述对应信息位全都为逻辑“1”还是全都为逻辑“0”。

25 25. 如权利要求 20 所述的方法，其中所述在每个存储单元中用于存储电荷的所述多个离散量包括 2^N 个，并且所述比较步骤反复至少 30 N 次。

用于多数字位的稳定电路

5 技术领域

本发明涉及半导体存储器，具体地讲，涉及受到漂移作用的半导体存储器的稳定性。这种存储器每存储单元可以存储一个和多个数字位。

10 背景技术

诸如 EEPROM, EPROM, FLASH 和 DRAM 这样的半导体存储器集成电路一般被用来在每个存储单元中存储一个单一的数字位，下文称之为位存储。以前对每存储单元能够存储多于一个数据位的存储器以及它们的优点有过说明。这种每单元存储多位的存储器被称为多电平存储器，因为它们需要多于通常用于一位存储技术的两个（导通与非导通）单元阈值电压 V_T 的电平。

多电平存储器中的每个电平代表存储在每个存储单元中的电荷的一个特定范围，并且在非易失性存储器的情况下，即，EEPROM, EPROM 和 FLASH 存储器的情况下，代表单元 V_T 值的一个特定范围。为了每一存储单元存储 N 位，需要把单元的 V_T 范围和存储电荷的量划分为 2^N 个电平。每个电平对应于用于所有 N 位的一个唯一的二进制数据模式。将单元擦除或编程以存储电荷，使得单元的 V_T 能够设定在这些 2^N 个电平中的一个内。读出电路确定存储单元的 V_T 在哪一个电平内，并且读出为 N 位存储的对应的二进制数据模式。由于可以在以前每单元仅存储单一位的相同的存储单元阵列区中存储 N 倍的位数，因此可以用很低的每位价格存储数字信息。

在以下对非易失性存储器的说明中，“电平”被用来表示 V_T 值的一个范围，而不是一个单一的电压值。此外，单元 V_T 一词不是固态元件物理学所定义的最严格意义中使用的，而是在读出电路如何确定存储单元的电导率状态的意义上使用的。电导率与单元的 V_T 有关。同样，DRAM 电平代表存储电荷的一个范围，而不是一个单一的电荷值。

35

读出单个电平的操作是通过把读出电压或电流表示的存储单元的电导率（或存储电荷）与多个参考电压或电流进行比较执行的。由于在读出技术领域中的技术人员可以容易地通过负载电路将电流转换成电压，所以本发明是以电压读出进行说明的。

5

存在着许多与多电平存储器相关的问题。为使每存储单元存储 N 位，一般建议用 $2^N - 1$ ，或 2^N 个参考电压值 V_{RI} ，其中 $I = 1, 2, \dots, 2^N - 1$ ，或 2^N ，使每一存储单元存储 N 位，以使 2^N 个电平相互分离， $V_{R1} < V_{R2} < \dots < V_{(2^N)}$ 。应当注意，为简化起见，有时把 V_{RI} 简单地写为 V_R 。图 1A-1C 中示出了参考电压值与单元 V_T 之间的关系，图 1A-1C 分别示出了用于每单元存储一位、两位和四位的整个存储器芯片的多电平读出参考电压和单元 V_T 的分布。

10

15

20

当一个存储单元 V_T 接近 V_R 电压之一时将产生不希望的条件。单元 V_T 的确定变模糊得。确定单元 V_T 的实际读出电路受限于电路的稳定性和速度，和数字交换噪声以及电源电压、温度、和硅工艺中的其它变化造成的电压和电流的变化。与不需要离散多个电平或多个参考电压的模拟信号存储不同，数字存储器存储技术要求无二义性地确定电平，并且需要这种离散参考电压。如果单元中的电平被不正确地读出，那么数字存储器将误操作，并且可能每单元丢失多达 N 位之多。

25

30

为了避免读出单元 V_T 值接近，或等于， V_R 电压之一的问题，提出了使一个电平中的单元 V_T 与其它相邻电平分离的容限电压范围 V_{MPI} （见图 2）。在单元被擦除和编程时执行这种分离。但是，没有定义每个 V_{MPI} 的两端。而是，这种提议的技术在硅工艺中使用了和一个冗余的编程算法结合的统计控制，以便为每单元存储一个单一位、或可能两位建立适当的 V_{MPI} 。此外，没有机构来确定是否一个单元 V_T 超出了恰当的范围。但是，该技术仅在满足了两个条件时才是可靠的。首先， V_R 值之间的分离足够大，以便为无二义性的读出提供适当的容限。其次，单元的 V_T 必须在恰当的电平内保持稳定，并在同样长的时间内保持数据有效。这个时间周期可能与存储器芯片的寿命一样长。

35

但是，所有多电平存储技术都必须克服的一个问题是在每个电平的很窄范围内对存储单元的 V_T 控制的问题。这种 V_T 的控制问题应

用于存储器的所有操作模式，包括编程，擦除和读取存储单元。由于每单元存储 N 位需要每单元 2^N 个电平，因此随每存储单元存储的位数增加， V_T 控制问题的难度成几何级地增加。随电平数量在所有有效单元 V_T 值的一个固定范围 V_F 内的增加，在一单一电平内的 V_T 值的范围 V_L ，以及在不同电平中分离单元 V_T 的容限范围 V_M （见图 3A-C）变窄。在本发明中，为简洁起见，我们有时把 V_{LI} （其中 I 是 2^N 个电平中的一个）简称为 V_L 。

V_F 经常是固定的，因为它受编程、擦除和读出操作过程中能够施加于存储单元的终端的电压范围的限制。 V_F 受电路速度、复杂性和数据存储可靠性的约束。对于许多以前提出的非易失性存储器技术， V_F 大致地等于用于读出操作的电源电压 V_{CC} 。

对于一个极简单的例子，其中定义所有电平的 V_L 的范围都是相同的，并且电平之间的容限是零，那么 $V_L = V_F / 2^N$ 。例如，对于从 5 伏 V_{CC} 操作的一个一位存储技术的 V_L 的范围是 $5/2 = 2.5V$ ，而对于从 3V 操作的每单元 4 位的多电平存储器的 V_L 的范围减少到 $3/16 = 187.5mV$ 。如果在每个电平之间加上一个容限电压范围 V_M ，那么 $V_F = V_{L1} + V_{M1} + V_{L2} + V_{M2} + \dots + V_M (2^N - 1) + V_L (2^N)$ 。继续上述简单例子，并假设所有的 V_M 范围也相等，那么现在电平范围减小到 $V_L = (V_F - (2^N - 1) \times V_M) / 2^N$ 。把 V_M 等于 0.1V 代入，那么在上述的 4-位，3-伏 V_{CC} 例子中的 V_L 现在已减小到 $(3 - (15 \times 0.1)) / 16 = 93.8mV$ 。

加入 V_T 控制的问题是擦除存储单元的程序。在存储阵列的实际实施例中，单元的擦除是以包含许多单元的块执行的，所以完全擦除单元的 V_T 分布比更有选择地编程的其它电平宽。用于定义完全擦除单元的较宽 V_L 电平 V_{LERASE} 进一步减小编程电平的 V_L 范围。图 3A 至 3C 示出了图 1A-1C 中所示的相同技术的这种问题的曲线图。

应当注意，图 1A，1B，1C，2，3A，3B 和 3C 是编程提高单元 V_T 技术的例子。上述的说明类似地应用于编程过程中降低单元 V_T 的技术。在这种场合，这些附图应当在单元 V_T 的高端显示较宽的擦除电平，而不是低端。

因此，由于随着每单元位数的增加每一电平中的 V_T 范围成几何

级数地减小，所以在多电平存储中的 V_T 控制比在一位存储系统中更为重要。

5 此外，存在着各种可以导致单元的 V_T 从它的初始编程值漂移开的机制。许多这些机制是由在单元的原始编程之后施加于该单元的电压应力造成的。这些条件称为“干扰”，并且当该单元分组在其它类似单元的阵列内以构成一有用存储系统时是不可避免的。由于在这些操作中使用了高电压，编程和擦除干扰只存在最短的累加时间，但却是 V_T 漂移的重要原因。例如，在介绍新的 FLASH 存储技术的技术文
10 献中经常报导干扰数据。

V_T 漂移的其它原因是由于在浮栅周围或浮栅与存储单元的基底之间的俘获电荷的数量或位置改变的结果。俘获电荷可能是由于缺陷或是在诸如反复的编程/擦除 (P/E) 循环过程之类的时间中施加的高电
15 场的累积效应造成的。被驱动通过栅极介质的擦除或编程电流越大，累积的俘获电荷越多。最终导致了那些操作的延迟，和该部分的误操作。

俘获电荷也可能以 P/E 循环的不可重复的形式发生。这归因于所谓的“劣等位 (rogue bit)”效应。劣等位显示出在一个循环中编程或擦除能力上的偏移，并且在另一个循环中又返回正常。由于缺陷统计变化的存在， V_T 漂移率在一段时间中可能是不恒定的，并且从单元到单元可能是不一样的。由于单元在其中相互连接的不同的行和列是以不同的次序存取的，所以一个存储单元阵列中的每个单元经历其本身唯一的干扰和电荷应力条件的组合。随机电压应力造成的漂移效应
20 附加在随机缺陷造成的漂移效应上。这些效应累积在单元的 V_T 漂移上。

非易失性单元通过反复的 P/E 循环最终放松持续性，即，它们编程或擦除对于允许这些操作进行的时间来说太慢，并最终失败。一些
30 以前的编程技术对存储单元阵列的一个特定区段已经循环的次数计数。在一个实例中，利用这种循环数据根据预定保留的最大循环次数防止存储单元阵列的过度使用的部分继续 P/E 循环。因此这种编程技术可以用来在需要之前降低存储器的功能性。

35

最后，由于随存储单元物理尺寸减小造成的单元中电场增强，使

得上述所有 V_T 干扰和漂移机制问题更为严重。已知这些问题妨碍单元的定标和阻碍了更经济的存储芯片的生产。例如，当每单元存储多位或使用较低的电源电压时 V_L 范围变窄，使得单元 V_T 漂移问题更加严重。

5

本发明解决了或实质上减轻了这些问题。根据本发明的存储器直接测量每个非易失性存储单元在其整个寿命中的稳定性，可编程性和可擦除性。

10 发明内容

本发明提供了一种集成电路存储系统，该存储系统具有多个存储单元，每个存储单元有一个在对应于数字信息的多个电平之一中的存储电荷。存储系统还包括响应于存储单元中电荷预定变化量恢复每个存储单元中的电荷以避免存储单元中数字信息丢失的电路。电荷恢复是由小编程和小擦除操作进行的，它仅把足够的电荷移入和移出存储单元，从而足以使存储电荷恢复到它的由一个第一专用参考电压值组定义的原始值。

集成电路存储系统还具有用于新编程和擦除操作的电路。在存储单元的编程过程中，存储系统把对应于数字信息的多个电平之一中的电荷写入被编程的每个存储单元。电平是由一个第二专用参考电压值组定义的，第二专用参考电压值组比第一专用参考电压值组更宽地定义电平。随后，在需要时执行小编程和小擦除操作，以使存储电荷范围变窄为第一专用参考电压值组定义的电平。存储电荷的初始宽范围加速存储单元的编程。

在存储单元擦除过程中，存储系统以专用参考电压值缩小擦除存储单元的电平中的存储电荷的范围。变窄的擦除电平增大了对应于数字信息的电平之间的差距。这增大了电平之间的安全容限，或允许增加能够存储在存储单元中的位数。

30 附图说明

图 1A 是一个每单元一位数据存储的整个存储器芯片的读出参考电压和单元 V_T 分布的一个实例。

35 图 1B 是一个每单元两位数据存储的整个存储器芯片的读出参考电压和单元 V_T 分布的一个现有技术的实例。

图 1C 是一个每单元四位数据存储的整个存储器芯片的读出参考电压和单元 V_T 分布的一个实例。

图 2 是一个显示了每单元两位数据存储的整个存储器芯片的用于读出, 编程和擦除模式的读出参考电压和所得到的单元 V_T 分布的一个特定现有技术的实例, 并示出了分离不同 V_T 电平的容限、。

图 3A 是一个每单元一位数据存储的整个存储器芯片的定义 V_F , V_{LERASE} , V_M 和 V_L 单元 V_T 分布范围的现有技术实例。

图 3B 是一个每单元两位数据存储的整个存储器芯片的定义 V_F , V_{LERASE} , V_M 和 V_L 单元 V_T 分布范围的现有技术实例。

图 3C 是一个每单元四位数据存储的整个存储器芯片的定义 V_F , V_{LERASE} , V_M 和 V_L 单元 V_T 分布范围的实例。

图 4A 是上升 V_T 的单元 V_T 漂移与时间关系曲线的一个实例。

图 4B 是下降 V_T 的单元 V_T 漂移与时间关系曲线的一个实例。

图 5A 是示出了两个新的多电平数字读取读出参考电压和一个给定电平的防护带的上升 V_T 的单元 V_T 漂移与时间关系曲线的一个实例。

图 5B 是示出了两个新的多电平数字读取读出参考电压和一个给定电平的防护带的下降 V_T 的单元 V_T 漂移与时间关系曲线的一个实例。

图 6 示出了用于读出每存储单元多电平数字数据的四个位的 BSERD 技术的优选实施例的一般操作。

图 7 是本发明的一个优选实施例的方框图。

图 8 是在该优选实施例中执行的 BSERD 技术的流程图。

图 9 示出了利用 BSERD 技术从一个存储单元读出多电平数字数据的一个实例。

图 10 是图 9 中所示实例的时序图。

图 11 示出了本发明的一个实施例的页模式操作的存储阵列和 Y-驱动器 (Y-DRIVER) 的详细结构的方框图。

图 12 是小编程技术的一个实例, 示出了在两个不同时间的存储器中多个单元的单元 V_T 分布, 并且示出了在一给定电平内两个新的小编程读出参考电压和防护带。

图 13A 是小擦除技术的一个实例, 示出了在四个不同时间的存储器中多个单元的单元 V_T 分布, 并且示出了在一给定电平内新的预擦除小编程读出参考电压和防护带以及两个新的小擦除读出参考电压和防护带。

图 13B 是图 13A 的继续, 并且是小擦除技术的一个实例, 示出了

在三个不同时间的存储器中多个单元的单元 V_T 分布，和示出了在一给定电平内新的预擦除小编程读出参考电压和防护带以及两个新的小擦除读出参考电压和防护带。

5 图 14A 是 V_T 过冲恢复技术的一个实例，示出了在四个不同时间的存储器中多个单元的单元 V_T 分布，并且示出了在一给定电平内新的预擦除微编程读出参考电压和防护带。

10 图 14B 是图 14A 的继续，并且是 V_T 过冲恢复技术的一个实例，示出了在三个不同时间的存储器中多个单元的单元 V_T 分布，和示出了在一给定电平内新的预擦除微编程读出参考电压和防护带以及两个新的微编程读出参考电压和防护带。

图 15A 是一个现有技术的实例，示出了紧接着编程之后的在一给定电平内存储器中多个单元的初始紧密单元 V_T 分布，并示出了在一长时间周期之后刚好在模糊读出故障点的相同单元的实际展宽分布。

15 图 15B 是一个实例，示出了在一给定电平内两个新的编程读出参考电压和防护带以及在三个不同时间的存储器中多个单元的单元 V_T 分布；紧接着第一部分编程之后的初始宽单元 V_T 分布，紧接着第二部分编程之后的紧密分布，和在其中应用恢复技术防止模糊读出故障的如图 15A 中指出的同样长的时间周期之后相同单元的恢复分布。

20 图 16A 是编程技术的一个实例，示出了在四个不同时间存储器中多个单元的单元 V_T 分布，并示出了在一给定电平内两个新的编程读出参考电压和防护带。

图 16B 是图 16A 的继续，并且是小擦除技术的一个实例，示出了在一给定电平内三个不同时间的存储器中多个单元的单元 V_T 分布。

25 图 17A 是定义每单元四位数据存储的一个擦除块的 V_F ， V_{LERASE} ， V_M 和 V_L 单元 V_T 分布范围的一个实例，其中象以前使用过的技术一样把 V_{LERASE} 用于 V_{L0} 。

30 图 17B 是定义每单元四位数据存储的一个擦除块的 V_F ， V_{LERASE} ， V_M 和 V_L 单元 V_T 分布范围的一个实例，其中没有把 V_{LERASE} 用于 V_{L0} ，并且其中 V_L 电平向外延伸跨越整个 V_F 范围，导致了比图 17A 中更宽的 V_L 和 V_M 范围。

图 17C 是定义每单元四位数据存储的一个擦除块的 V_F ， V_{LERASE} ， V_M 和 V_L 单元 V_T 分布范围的一个实例，其中没有把 V_{LERASE} 用于 V_{L0} ，并且其中 V_L 电平没有叠盖 V_{LERASE} 范围，这在一些技术中对于可靠的数据读出可能是必须的。

35 图 18A 是擦除技术的一个实例，示出了在四个不同时间跨越四个电平的擦除块中多个单元的单元 V_T 分布，并且示出了在 V_{L0} 电平内

四个新的擦除读出参考电压和新的 V_{MEFL} 防护带。

图 18B 是图 18A 的继续，并且是擦除技术的一个实例，示出了在四个不同时间跨越四个电平的擦除块中多个单元的单元 V_T 分布，并且示出了在 V_{L0} 电平内两个新的擦除读出参考电压和新的 V_{MEFL} 防护带。

具体实施方式

为避免上述问题，根据本发明的一种存储系统进行较小的单元 V_T 调节，甚至在编程很长时间之后，自动地改正漂移效应。单元 V_T 被恢复到正确 V_{LI} 范围内的最佳值。在需要时进行这种恢复操作，以便在整个存储器使用寿命中维持可靠数据存储的适当读出容限。存储系统确定什么时间应当通过加入、确定什么时候单元的 V_T 在恰当的 V_L 范围的任何一侧太靠近 V_R 值之一的读出参考电压来恢复单元的 V_T 。

本发明也不同于不使用被离散读出参考电压分离的多个电平的以前的模拟存储技术。这种模拟存储技术一旦在原始模拟信号被 V_T 漂移破坏后不能对单元漂移或干扰效应进行检测和改正。

单元的 V_T 可以在三种路径之一中移动：

V_T 上升（例如，当把电子加到浮栅时，如图 4A 中所示），

V_T 下降（例如，当把电子从浮栅移开时，如图 4B 中所示），和稳定的 V_T ，由于电荷的平衡或缺乏应力条件。

本发明的存储器响应于每个存储单元的上述三种可能性中的任何一个，并确定恢复单元 V_T 所需的操作。这种响应和确定可能在原始数据编程很长时间之后发生。存储系统在编程之前不必执行擦除。作为替代，存储器执行将 V_T 在正确方向上调节的小擦除和小编程操作。在这些操作过程中，仅有少量的进行较小 V_T 恢复所需要的存储电荷被移动。这避免了干扰其它单元上的数据。以前的 P/E 循环技术的耗损效应被降至最低。

此外，存储系统改正在一次正常 P/E 循环中大量电荷流过栅极介质时产生的劣等位。在一个恢复操作过程中，仅需要单元 V_T 中很小的改变来恢复 V_T 。

通过使单元在被称为区段或擦除块的大组中被电擦除，正常地优化 FLASH（闪速）非易失性存储单元阵列以减小芯片的硅面积。编程是以更大的选择方式完成的，使得只有很少的位在一个较小的组中，例如在一个页、字或字节中被同时编程。在编程页、字或字节内的所有单元也可以被同时读出（读取或验证）。一个擦除区段或块可以包含多个编程页。

编程以与进行擦除的方向相反的方向移动存储单元的 V_T 。应当注意，这里使用的对编程和擦除的定义与极性无关，并且也与电荷移动到一个诸如浮栅之类的电荷存储区上的方向无关，而是由存储单元阵列确定的。因此，本发明在通过增加或减小单元 V_T 编程的两种技术中都可以使用。

在对集成电路单元的编程中，在把任何新的数据位编程到擦除块内的任何页中之前，存储系统首先完全擦除擦除块中的所有单元。以前的编程算法把每个单元以一系列编程脉冲编程到希望的 V_T 。多电平编程则预想到将来的漂移，最好是单独地设定每个单元的 V_T ，以在编程过程中获得最紧密的可能的 V_L 。以前的多电平编程算法需要多次的编程/验证反复，以使单元 V_T 向一个方向偏移。极大地依赖统计过程控制以防止最快的编程单元的 V_T 超出一电平内的恰当范围，和停留在太靠近一个 V_R 的位置。为避免超出，需要许多小编程反复。

与此相反，这里所述的存储系统的编程操作在编程模式中应用小擦除操作可以使单元的 V_T 向两个方向偏移。小擦除操作允许执行大编程迭代。在编程过程中控制每个 V_L 范围的两侧，而不是依赖每个单元的编程特性的统计控制。下述编程技术的另一个优点是，减少了一个多电平存储器的编程时间，这是由于小编程和小擦除操作减小了每个电平中的初始编程的宽单元 V_T 分布，并改正了任何 V_T 超出。

利用本发明，一个存储系统在嵌入到外部系统的同时可以通过直接测量确定什么时候一个单元不再能够可靠地编程、擦除或存储数据。只有确定了立即需要这种动作之后才执行适当的动作。这延长了存储器芯片的使用寿命。

以下更详细地说明根据本发明的存储系统的各个方面。多电平数据读出、小编程、小擦除，编程和擦除的操作如下：

I.多电平数字数据读出

读出操作的一般说明

5 本发明提供了一种新的多电平数字数据读出技术。这种技术在读出操作过程中使用了附加的参考电压和防护带，以确定一个存储单元 V_T 是否已经不恰当地漂移靠近电平分离参考电压 V_{RI} 或 $V_{R(I+1)}$ 中的任何一个，电平分离参考电压 V_{RI} 和 $V_{R(I+1)}$ 定义了多个 V_{LI} 电平中的一个。

10 如图 5A 和 5B 中所示，在读出模式过程中，参考电压 V_{UGI} 和 V_{LGI} 引入单元 V_T 读取容限防护带。对于每个电平“ I ”，有两个读取容限防护带；1) 上读取容限防护带， $V_{MRUI} = V_{R(I+1)} - V_{UGI}$ ，和 2) 下读取容限防护带， $V_{MRLI} = V_{LDI} - V_{RI}$ 。防护带 V_{MRUI} 和 V_{MRLI} 相对于 V_{RI} 定位。例如， V_{MRUI} 相对于 $V_{R(I+1)}$ 定位，
15 和 V_{MRLI} 相对于 V_{RI} 定位。一个 V_R 周围的总读取容限防护带是 $V_{MRI} = V_{MRU(I-1)} + V_{MRLI}$ 。

如果新的多电平数字数据读出技术检测到一个单元 V_T 已经漂移进入一个读取容限防护带，那么建立起一个恢复循环条件，并设定一个
20 条件状态标志。新的多电平数字数据读出技术也使用 $(2^N + 1)$ 电平分离参考电压 V_{RI} ，其中 N 是存储在每个存储单元的多电平数字位数， $I=0$ 至 $(2^N + 1)$ 。两个附加读取容限防护带也被用于读出是否一个单元已经漂移进入可靠数据读出的极限。

25 二进制检索嵌入恢复检测方法

称为二进制检索嵌入恢复检测 (BSERD) 技术的新的多电平数字数据读出技术读出作为多电平数字数据存储在每个存储单元中的数据的
30 多个位，并利用带读出检测确定是否建立了一个恢复循环条件。

35 图 6 示出了用于读出每存储单元多电平数字数据的四个位 ($N=4$) 的 BSERD 技术的一个优选实施例的一般操作。对这个实施例，选择 $N=4$ 只是为了说明本发明的目的，并不限制本发明应用于选择其它的 N 。例如，本发明可以用于每单元存储一位 ($N=1$)，或任何其它多位存储应用，其中 $N>1$ 。

在每个“Set BitX (设定位 X)”循环 ($X=0$ 至 $N-1$) 过程中，

执行两个子循环操作。这些子循环是后面跟随着恢复检测子循环的数据检测子循环。在每个子循环中，设定一个适当的参考电压 V_{CRF} ，并且与读出的单元 V_T 比较。在某一时刻的 V_{CRF} 电压取决于 V_{CRF} 电压与在先前的每个时刻读出的单元 V_T 之间的比较结果。在必要数量的 Set BitX 循环结束后，执行一个边缘检测子循环和一个边缘恢复子循环，以检测单元 V_T 是否在单元 V_T 读出范围的极端可靠的界限之内，和单元 V_T 是否在极上或极下恢复容限范围内。

多电平存储器读出系统

图 7 示出了本发明的一个优选实施例的方框图。从每个存储单元读出多电平数字数据的四个位。一个数据，地址和控制系统接口块 10 经过一个 DATA（数据）总线 11，一个 ADDR（地址）总线 12，一个 CNTRL（控制）总线 13 和一个 CLKIN（时钟）线 14 实现外部系统与本优选实施例之间的数据，控制，和系统时钟信号的连接。作为选择，一个内部时钟块（图 7 中未示出）在 CLCK（时钟）信号线 15 上产生一个时钟信号，如图 7 中所示。在读出操作过程中，一个数据，地址和控制系统接口块 10 通过一个双向地址，ADDRBUS 总线 16 向一个对应的水平解码器 X-DEC 17，垂直解码器 Y-DEC 18 和一个块解码器 BLOCK-DEC 19 提供地址信息，以便在一存储阵列块 CELL ARRAY 21 中选择一个适当的存储单元 20。

解码器根据 PRG-ERS 控制总线 22，HVCTRL 控制总线 23，PRGHV 高电压线 24，和 HVOUT 高电压总线 25 上的信号在单元 20 的终端提供适当的多电平编程，擦除或读出电压。PRG-ERS 和 HVCTRL 信号是编程擦除读取恢复序列发生器块 26 产生的。HVOUT 和 PRGHV 信号是由 HVGEN 块 27 产生的。

在读出过程中，RD 线 28 趋于高电平，以接通 GATEC 开关 29，使 MLLINE 线 30 连接到 VMEM 线 31。MLLINE 线 30 通过一个 Y-MUX 块 32 和一个 BL00 位线 71 连接到单元 20。在读出过程中，连接到 MLLINE 线 30 的特定单元 20 依赖于 Y-MUX 块 32，响应 YDECO 总线 34 上的由 Y-DEC 块 18 产生的信号。VMEM 线 31 连接到一个比较器 CMP35 的非倒相输入端。当高电平时，RD 线 28 也允许负载块电路 CELL LOAD36 连接于 VMEM 线 31。

当适当的电压施加于编址单元 20 的终端并且 CELL LOAD36 被连

接时，在 VMEM 线路 31 上产生一个表示四位数字数据的电压。编程擦除读取恢复序列发生器块 26 也经过 PRGRDB 总线 37 把信号输送到参考电压选择块，VX VY GEN 块 38 和 Δ VU Δ VL GEN 块 39。VX VY GEN 块 38 和 Δ VX Δ VY GEN 块 39 的输出信号被输送到一个比较器参考电压产生块，VCRFGEN 块 40。VX XY GEN 和 Δ VX Δ VY GEN 块 38 和 39 通过一参考电压总线 RVBUS42 从一电压参考块 41 选择参考电压。

在读出操作过程中，二进制检索读取序列发生器块 43 和编程擦除读取恢复序列发生器块 26 控制和排序由来自一个数据，地址和控制系统接口块 10 的 CLCK 线 15 上的时钟信号同步化的 BSERD 技术。二进制检索读取序列发生器块 43 使双向总线 DATABUS[0-3] 44 上的适当的数据线导通。

VCRFGEN 块 40 还从来自编程擦除读取恢复序列发生器块 26 的控制线 PHSEL 线 46 和来自一 STPDIR 锁存器 52 的控制线 STPDIR 线路 47 接收控制信号。STPDIR 锁存器 52 通过一个输出线 VCMPO49 和一个传输门 GATEA50 连接到 CMP 比较器 35 的输出端。根据来自 DATABUS[0-3]总线 44 的输入信号，VX VY GEN 块 38 和 Δ VX Δ VY GEN 块 39 对 VCRFGEN 块 40 产生适当的电压。表 1 示出了在读出过程中，根据 PHSEL 和 STPDIR 线 46 和 47 上的电平，从 VCRFGEM 块 40 在输出 VCRF 线 45 上的输出电压。VCRF 线 45 连接于 CMP 比较器 35 的倒相输入端。

表 1

PHSEL	STPDIR	VCRF 上电压
低	不考虑	VX
高	低	VX + Δ VL
高	高	VY + Δ VU

图 6 示出了参考电压 V_{R0} 至 V_{R16} ；它们中任何一个都是在 PRGRDB 总线 37 和 DATABUS[0-3]总线 44 的控制信号下由 VX VY GEN 块 38 作为输出产生的，而 PRGRDB 总线 37 和 DATABUS[0-3]总线 44 又受序列发生器 26 和 43 控制。在读出过程中，从 VX VY GEN 块 38 输出的电压是 $VX = V_{RI}$ 和 $VY = V_{RI}$ ，其中 I 是一个 0 至 16 之

间的整数,由 DATABUS[0-3]总线 44 上的数据线确定。在读出过程中,也如图 6 中所示, $\Delta V_X \Delta V_Y$ GEN 块 39 产生输出电压 ΔV_L 和 ΔV_U , 其中 $\Delta V_L = V_{MRLI}$, $\Delta V_U = V_{MRU} (I-1)$, $I=0$ 至 15。

5 数据[0-3]锁存器块 48 具有四个锁存器,这四个锁存器在通过 BSERD 技术读出时从每个存储单元 20 存储四个位。CMP 比较器 35 比较 VMEM 输入线 31 和 VCRF 输入线路 45 上的电压,并在 VCMPO 输出线 49 上产生一个逻辑高或低信号。当 VMEM 线 31 上的电压高于 VCRF 线 45 上的电压时, VCMPO 线 49 输出一个逻辑高信号。相反, 10 当 VMEM 线 31 上的电压低于 VCRF 线 45 上的电压时, VCMPO 线输出一个逻辑低信号。

除了传输开关 GATEA 50 之外,输出 VCMPO 线 49 还连接于一个传输开关 GATEB 51 的输入端。GATEA 50 的输出线连接于 STPDIR 15 锁存器 52 的输入线。GATEB 51 的输出线连接于一个锁存器 RES LATCH 53 的输入端。来自编程擦除读取恢复序列发生器块的 PHSEL 线 46 控制开关 GATEA 50 和 GATEB 51 的操作。当 PHSEL 线 46 为高电平时,来自 CMP 比较器 35 的输出线 VCMPO 49 通过开关 GATEB 51 连接于 RES LATCH 53 的输入端。当 PHSEL 线 46 为低电平时, 20 VCMPO 线 49 通过 GATEA 开关 50 连接于 STPDIR 锁存器 52 的输入端。

来自编程擦除读取恢复序列发生器块 26 的 RST 线 54 连接于 STPDIR 锁存器 52 和 RES 锁存器 53。当 RST 线 54 为逻辑高时,两个 25 锁存器 52 和 52 被复位。当 RES 锁存器 53 被复位时,一个来自锁存器 53 的输出线 RES 线 55 为低电平。来自 STPDIR 锁存器 52 的输出 STPDIR 线 47 携带着一个来自锁存器 52 的倒相信号。当 STPDIR 锁存器 52 被复位时,输出 STPDIR 线 47 为高电平。STPDIR 线连接到一个高压开关 HVSW56, VCRFGEN 块 40, 和传输开关 GATED 57、GATEF 30 58 和 GATEH 59。当 STPDIR 线路 47 为高电平时, HVSW 开关 56 在连接于高电压发生器 HVGEN 块 27 的 PRGHV 线 24 上向 MLLINE 线 30 发送一个高电压脉冲。当 STPDIR 线 47 为低电平时, HVSW 开关 56 从 MLLINE 线 30 上断开 PRGHV 线 24。

35 当来自编程擦除读取恢复序列发生器块 26 的 HVCTRL 总线 23 被驱动设置一个编程或擦除操作时, HVGEN 块 27 产生高电压脉冲。在

读出过程中, 因为 HVCTRL 总线被设置用于读出, 允许高电压 PRGHV 线 24 浮动, 并且 HVGEN 块 27 不产生任何高电压脉冲。在编程、擦除、恢复和也有可能的读出操作模式过程中, HVGEN 块 27 还通过 HVOUT 总线 25 向 Y-DEC 块 18, X-DEC 块 17 和 BLOCK-DEC 块 19 提供高电压。

Y-MUX 块 32 在 Y-DEC 块 18 的控制下选择被编址单元 20 适当的位(读出)线。一个寄存器和地址缓冲器块 60 存储存储单元的地址, 当 BSERD 技术确定时该地址应当与恢复条件状态标志一同恢复。编程擦除读取恢复序列发生器 26 经过一个 BFCTLBUS 总线 61 控制连接到 ADDRBUS 总线 16 的寄存器和地址缓冲器块 60。

GATED 和 GATEE 开关 57 和 62 串联到一个设置为逻辑低信号的 GROUND (接地) 线 64。开关 57 和 62 分别由 STPDIR 线 47 和 RES 线 55 上的信号控制。同样, GATEF 和 GATEG 开关 58 和 63 串联到一个设置为逻辑高信号的 POWER (电源) 线 67。开关 58 和 63 分别由 STPDIR 线 47 和 RES 线 55 上的信号控制。

当 STPDIR 线 47 和 RES 线 55 一同为逻辑高时, 一个逻辑低信号通过 GATED 和 GATEE 开关 57 和 62 传送到一个 FLGB 线 65。在 STPDIR 线 47 和 RES 线 55 上的信号的任何其它组合情况下, 该逻辑低信号不传送到 FLGB 线 65, 这使得一个 PULL UP LOAD (上拉负载) 块 66 在 FLGB 线 65 上施加一个高逻辑信号。当 STPDIR 线 47 和 RES 线 55 一同为逻辑低时, 一个逻辑高信号通过 GATEF, GATEG 开关 58 和 63 传送到 FUGB 线 68。在 STPDIR 线 47 和 RES 线 55 上的信号的任何其它组合的情况下, 逻辑高信号不传送到 FUGB 线 68; 那么一个 PULL DOWN LOAD (下拉负载) 块 69 在 FUGB 线 68 上施加一个低电压。FLGB, FUGV 线 65 和 68 是到编程擦除读取恢复序列发生器块 26 和二进制检索读取序列发生器块 43 的输入线路。当 STPDIR 线 47 为低电平时, 一个编程擦除读取恢复序列发生器块 26 产生的并且通过 LCHDAT 线 70 传导的数据锁存脉冲经过 GATEH 开关 59 传送到数据 [0-3]锁存器块 48。

图 8 示出了执行 BSERD 技术的流程图。二进制检索读取序列发生器块 43 和编程擦除读取恢复序列发生器块 26 包含有必要的控制电路以执行 BSERD 技术。熟悉集成电路设计的技术人员应当清楚这种电

路的设计。

图 9 示出了从一个存储单元读出多电平数字数据的实例。图 9 中箭头示出了响应如图上时间轴上所示的每一个先前时刻 CMP35 的输出条件，在特定的时刻在 VCRF 输入线 45 上将参考电压输送到 CMP 比较器 35。

图 10 示出了如图 9 中所示的和以下将说明的，从一个存储单元 20 读出多电平数字数据的示例性的时序图。此前已经适当地编程的选定单元 20 的单元 V_T 在 VMEM 线 31（见图 7）被读出，是在深黑的点划水平线上，在图 9 中示为“单元 V_T ”。在时间 T_0 ，BSERD 技术开始，RES 线 54 被施加逻辑高脉冲，以使 STPDIR 和 RES 锁存器 52 和 53 复位。设置 PRGRDB 总线 37，PHSSEL 线 46 为逻辑低，HVCTRL 总线 23 被设置为读出，RD 线 28 是逻辑高，并且 ADDRBUS 总线包含有正在被读出的选定存储单元的地址。在线 54 上的一个复位脉冲之后，STPDIR 线是逻辑高，RES 线 55 是逻辑低。FLGB 线 65 是逻辑高，FUGB 线 68 是逻辑低。

现在“Set Bit3”循环（见图 6）和第一数据检测子循环开始。编程擦除读取恢复序列发生器块 43 把 DATABUS[0-3]总线 44 的最有意义的位线 Bit3 强制为高电平。由于 Bit3 被强制为高电平，DATABUS[0-3]总线上是 (0001)，VX VY GEN 块 38 在时间 T_{1D} 输出参考电压 V_{R8} ， $\Delta V_X \Delta V_Y$ GEN 块 39 输出参考电压 V_{MRL8} 和 V_{MRU7} 。由于此时 PHSSEL 线 46 是低电平，VCRFGEN 块 40 在 VCRF 线 45 上输出参考电压 V_{R8} 。刚好在时间 T_{1D} 之后，CMP 比较器 35 读出 VMEM 线 31 高于 VCRF 线 45，并在 VCMPO 线 49 上产生逻辑高。由于 PHSSEL 线 46 为低电平，GATEA 开关 50 使 VCMPO 线路 49 上的逻辑高信号传送到 STPDIR 锁存器 52 的输入线。锁存器 52 被设定，并在 STPDIR 线 47 上产生一个逻辑低信号。由于 STPDIR 线 47 为低电平，来自编程擦除读取恢复序列发生器块 26 的 LCHDAT 线 70 上的锁存脉冲通过 GATEH 开关 59，以便在数据[0-3]锁存器块 48 中的一个 DATA3 锁存器中锁存一个“1”。这标志着数据检测子循环的终点。

接下来，开始第一恢复检测子循环。PHSSEL 线 46 被设置为高电平。由于 PHSSEL 线 46 为高电平和 STPDIR 线 47 为低电平，在时间

5 T_{1R} , VCRFGEN 块 40 在 VCRF 线 45 上输出电压 ($V_{R8} + V_{MRL8}$)。CMP 比较器 35 读出 VMEM 线 31 上的信号仍然高于 VCRF 线 45 上的信号, 并在 VCMPO 线 49 上产生一个逻辑高信号。由于 PHSEL 线 46 是逻辑高, VCMPO 线 49 上的信号通过 GATEB 开关 51 传送到 RES 锁存器 53, 并且不通过 GATEA 开关 50 传送到 STPDIR 锁存器 52。由于 RES 锁存器 53 的输入端是高电平, 输出 RES 线 55 被设置为高电平。这是第一恢复检测子循环的终点, 和“Set Bit3”循环的终点。

10 BSERD 技术如上面段落中所述的那样执行事项和子循环序列, 顺序地执行 Set BIT2, Set BIT1 和 Set BIT0 循环。图 9 示出了各个不同时刻在 VCRF 线 45 上设置的参考电压。图 10 示出了不同离散时间 VCMPO 线 49, STPDIR 线 47, RES 线 55, FLGB 线 65, FUGB 线 68 和 DATABUS[0-3]总线 44 上的输出逻辑电压。如果在一个数据检测子循环的终点, 来自 CMP 比较器 35 的输出线 VCMPO49 保持逻辑低, 这表示 VMEM 线 31 上的电压低于 VCRF 线 45 上的电压, 那么 STPDIR 线 47 是逻辑高。当发生这种情况时, 没有来自 LCHDAT 线 70 的数据锁存脉冲从 GATEH 开关 59 通过。接着, 数据[0-3]锁存器块 48 中的适当数据锁存器保持复位在“0”。本例中不产生这种条件。同样, 如果 15 在任何一个恢复检测子循环的终点, STPDIR 线 47 和 RES 线 55 都成为逻辑高, 那么建立了一个恢复低循环条件, FLGB 线 65 被强制为低电平。如果在任何一个恢复检测子循环的终点, STPDIR 线 47 和 RES 线 55 都成为逻辑低电平, 那么建立了一个恢复高循环条件, FUGB 线 68 被强制为逻辑高。

25 当数据检测和恢复检测子循环完成时, BSERD 技术执行边缘检测和边缘恢复子循环。在前面的例子中最后“Set BIT0”的终点, DATABUS[0-3]总线 44 上的数据是 (1111)。在边缘检测子循环的开始, 编程擦除读取恢复序列发生器块 26 读出 DATABUS[0-3]总线 44 上的 (1111), 并设置 PRGRDB 总线 37 以在 VX VYGEN 块 38 的输出中选择 V_{R16} 参考电压, 和在 $\Delta V_X \Delta V_Y$ GEN 块 39 的输出中选择 V_{MRU15} , V_{MRL15} 参考电压。此时 PHSEL 线 46 是逻辑低, 在时间 T_{5F} VCRFGEN 块 40 在 VCRF 线 45 上输出 V_{R16} 参考电压。CMP 比较器 35 读出 VMEM 线 31 低于 VCRF 线 45, 并在 VCMPO 线 49 上 30 输出一个逻辑低。由于 PHSEL 线 46 是逻辑低, GATEA 开关 50 把 VCMPO 线 49 上的逻辑低信号输送到 STPDIR 锁存器 52, 并且锁存器 35

52 在 STPDIR 线 47 上输出一个逻辑高。这是边缘检测子循环的终点。在边缘检测子循环过程中，在 LCHDAT 线 70 上不产生锁存脉冲。

5 接下来，在边缘恢复子循环的开始，PHSSEL 线 46 被设置为高电平。由于 STPDIR 线 47 是高电平和 PHSSEL 线 46 是高电平，根据表 1，VCRFGEN 块 40 在时间 T_{5R} 在 VCRF 线 45 上输出电压 $V_{R16} - V_{MRU15}$ 。CMP 比较器 35 读出 VMEM 线 31 上的电压高于 VCRF 线 45 上的电压，并在 VCMPO 线 49 上产生一个逻辑高。由于 PHSSEL 线 46 为高电平，VCMPO 线 49 上的逻辑高通过 GATEB 开关 51 传送到 RES 锁存器 53。逻辑高信号不通过 GATEA 开关 50 传送到 STPDIR 锁存器 52。由于在 RES 锁存器 53 的输入端为高电压，RES 线 55 保持设置为高电平。利用条件：STPDIR 线 47=高，和 RES 线 55=高，建立了一个恢复低循环条件，并且 FLGB 线 65 被强制为低电平。这结束了边缘恢复子循环，并且也结束了 BSERD 技术。如果在边缘检测子循环的开始，BSERD 技术已经检测到 DATABUS[0-3]总线 44 上既不是 (1111) 也不是 (0000)，那么 BSERD 读出技术应当已经结束。

20 在 BSERD 技术完结时，唯一地对应于存储在选定单元 20 中的 V_T 电平的 N 位的二进制数据模式被锁存在数据[0-3]锁存器块 48 中，并且随时可以经过 DATABUS[0-3]总线 44，一个数据，地址和控制系统接口块 10 和 DATA 总线 11 读取。

25 在本优选实施例中，有两个可以在恢复检测或边缘恢复子循环的终点造成恢复循环条件的条件，恢复低和恢复高。STPDIR 线 47=高和 RES 线 55=高是强制使 FLGB 线 65 为低电平的恢复低条件。STPDIR 线 47=低和 RES 线 55=低是强制使 FUGB 线 68 为高电平的恢复高条件（见图 7 和图 8）。如果在 BSERD 技术操作过程中的任何时间建立了一个恢复循环条件，那么编程擦除读取恢复序列发生器块 26 检测该条件，并且把 FLGB 线 65 和 FUGB 线 68 的状态内部地存储在恢复循环状态锁存器中。恢复循环状态锁存器设置在编程擦除读取恢复序列发生器块 26 的内部。如果在 BSERD 技术的操作过程中已经设定了一个恢复循环条件，那么一旦它完成，编程擦除读取恢复序列发生器块 26 启动一操作，以把要求该恢复循环的存储单元的地址以及需要的恢复循环的类型存储在寄存器和地址缓冲器块 60 中。

35 要求恢复循环的地址以及需要的恢复类型的存储允许具有在晚些

时候执行该恢复循环的灵活性。如果立即执行当前存储单元上的恢复循环操作，那么不把需要的恢复循环的地址和类型存储在寄存器和地址缓冲器块 60 中。寄存器和地址缓冲器块 60 可以存储多个要求恢复循环操作的地址的恢复循环信息。寄存器和地址缓冲器块 60 可以设置在同一集成电路中或分离的系统中，并且可以由易失性或非易失性存储器构成。

图 6 和图 7 示出了从一个存储单元读出数据。这个优选实施例也进行从单元的页读出数据的工作。图 7 中的 Y-驱动器块 33 重复与一页中存在的单元号数 J 一样多的次数。图 7 中跨越 Y-驱动器块 33 边界的线，除了通向单元阵列块 21 的线之外，是所有 Y-驱动器块 33 共用的。FLGB 线 65 和 FUGB 线 68 在所有 Y-驱动器块 33 之间以线“或”方式连接在一起。

图 11 示出了一个实施例中的用于页模式操作的存储阵列和 Y-驱动器块 33 的更详细的结构。同一结构在编程和恢复循环过程中被用于在单元的一页上操作，并且在擦除循环中被用于在单元的一个擦除块上操作。有“J”个所示的 Y-驱动器块 33。来自图 7 中 Y-DEC 块 18 的 YDECO 总线 34 通过每个 Y-驱动器块 33 内的 Y-MUX32 控制活动页的选择。

图 11 中所示的标为单元阵列 21 的轮廓方框内的电路和存储单元阵列不是本发明的一部分，但是在这里加以说明以解释本发明的实施例是如何连接到一个任意存储单元阵列的。

有“S”个图 11 中所示的页。还有“K”个所示的块 83-84。每个块可以由多行单元组成。块选择和行选择是分别由 BLOCK-DEC 和 X-DEC 块 19 和 17 一同执行的，如图 7 中所示。每个块 83-84 内的 BSU 子块 77 和 BSL 子块 78 执行阵列中每个块 83-84 的位（读出）线（BL00 线 71 至 BLSJ 线 72）或共用线（CL0 线 75 至 CLK 线 76）的选择或释放。SLUK 线 79 和 SLLK 线 80 控制 BSU 块 77 和 BSL 块 78。如图 11 中所示，用于 BLOCK0 块 83 的 SLL0 线 81，SLU0 线 82 和 CL0 线 75 以及用于 BLOCKK 块 84 的 SLLK 线 80，SLUK 线 79 和 CLK 线 76 是图 7 中的 BLOCK-DEC 块 19 产生的。

字线（WL00 线 85 至 WLKT 线 86）选择所有块 83-84 内的适当

的行。图 11 中所示的每个块有“T”条字线。字线由图 7 中所示的 X-DEC 块 17 控制。使用了多个 X-DEC 块 17, BLOCK-DEC 块 19 和 Y-DEC 块 18。为了简洁, 仅分别将它们示出了一个。在页操作中, 选择了一个块 83-84, 选择这个块内的一行, 并且该行内的一页。图 11 中所示的用粗轮廓线标出的单元 20 形成了块 0 中行 1 中的页 1。这个页 1 是通过位(读出)BL10 线 73 至 BL1J 线 74, 字 WL01 线 87 和块选择 SLU0 线 82, SLL0 线 81 以及 CL0 线 75 选择的。一个擦除操作直接作用于一个选择的块 83 内的所有单元。一个编程操作仅作用一个选择的页中的单元。本发明的这个实施例可以连接到任何非易失性存储单元阵列。

II. 小编程

小编程操作的一般说明

在这一节中作为本发明的一部分说明一种称为小编程的新的多电平编程技术。与仅仅到处在完全擦除条件的单元中开始编程的现有技术的编程不同, 小编程技术操作不需要完全擦除并且在仍然包含有效数据的单元上进行编程。在现有技术中, 擦除操作把选定擦除块或区段中的所有单元的 V_T 一直偏移到的可用单元 V_T 范围 V_F 的一端, 如图 3A-3C 中所示, 并标为 V_{LERASE} 。在擦除操作之后, 现有的编程技术利用移动大量的电荷穿过存储单元浮栅周围的氧化物把选定单元的 V_T 偏移出 V_{LERASE} 范围并且进入一个其它电平。

与现有技术相反, 本发明的小编程技术仅移动少量的电荷, 而不需要完全擦除。每个选定的单元的 V_T 仅被少量的偏移, 并且仅在那个单元的原始 V_L 电平界限内。由于新的小编程技术仅移动少量的电荷, 因此避免了现有编程技术操作所造成的对单元 P/E 循环耐久性的有害影响。另外, 仅有改正以前确定的 V_T 漂移所需数量的电荷被移动。小编程技术比现有技术所作的那样从擦除条件开始编程用的时间少。此外, 小编程仅轻微地影响选定的单元, 所以产生的编程干扰对其它未选定单元的影响可以忽略。小编程操作可以在一 V_L 电平内以与擦除相反的方向少量地偏移 V_T , 并因此执行了一种在一恢复操作过程中恰当的 V_T 恢复以使可靠的多电平数据存储成为可能所需要的功能。

如同前面多电平数字读出章节中所述的那样, 小编程技术利用本发明的带读出能力, 读取和验证每个单元的 V_T 存储在 V_L 范围的恰

当带中。这保证了适当的容限防护带，以防止在该 V_L 范围两端的模糊读出。这是在小编程验证操作过程中利用新的多个参考电压完成的。

5 本发明的小编程技术具有在 $V_{RI}+V_{MPLI}$ 和 $V_{R(I+1)} - V_{MPUI}$ 的两个附加新参考电压。这些参考电压，与 V_{RI} 和 $V_{R(I+1)}$ 结合，在每个 V_{LI} 电平内定义了两个新的下和上小编程容限防护带 V_{MPLI} 和 V_{MPUI} ，如图 12 中所示。 V_{MPLI} 和 V_{MPUI} 可以等于或不等于，并且可以大于或不大于前面说明的 V_{MPLI} 和 V_{MPUI} 读取容限防护带。适当地选择每个 V_{LI} 电平内的 V_{MPLI} 和 V_{MPUI} ，使得小编程后所得的 V_T 分布优化到一个比 V_{LI} 范围窄的恰当的带内。优化可能要求 V_{MPLI} 和 V_{MPUI} 带互不相等，并且可能要求每一个 V_{LI} 电平有不同的带值，甚至是由存储单元阵列处理技术决定的任意带值。作为选择，每个 V_{LI} 电平中的 V_{MPLI} 和 V_{MPUI} 可以是等值的。本发
10
15 明可以适合于提供对各种各样的存储器处理技术的优化。

 本发明的验证技术与以前的技术不一样，以前的技术仅相对于一个单一的值验证单元 V_T ，和依赖统计过程控制来防止单元 V_T 超出恰当的 V_T 带。由于前面提到的劣等单元效应， V_T 超出在任何编程操作过程中都有可能发生，并可能发生。这些效应是由俘获在存储单元的
20 浮栅周围的氧化物中的电荷或其它加工缺陷造成的。

 小编程技术通过仅移动少量电荷通过栅氧化物使电荷俘获与产生劣等单元的可能性降至最低。小编程验证操作检测是否 V_T 已经偏移太多进入了一个防护带，因而使用了一个小擦除操作（将在以后说明）以便在该电平内重新建立恰当的 V_T 。这种不首先完成一完全擦除和再编程操作的对 V_T 超出范围的恢复可能性在现有技术中是不存在的。利用本发明的恢复方法，在 V_T 超出范围的恢复过程中避免了完全擦除和再编程，并保护了芯片的 P/E 持续寿命。
25
30

小编程方法

 设想多个单元连接在同一编程页中是对小编程技术的最好说明。一个实际多电平存储系统使用了页模式构造，使得多个单元被同时编程，以减少编程时间。如果不了解整个芯片内的，那么必须充分了解至少是这样一页中的所有单元的性能，以便观察编程的任何形式的所有可能的分枝。
35

图 12 示出了存储器中同一电平 V_{LI} 内的多个单元在两个不同时间 T_{RESR} 和 T_{RESP} 的 V_T 分布。在这个例子中，一个 V_T 分布显示了已经利用本发明的上述 BSERD 技术在时间 T_{RESR} 读取单元后的条件。BSERD 技术已经检测到两个具有在 V_{MRLI} 防护带内 A 和 B 的 V_T 值的单元。已经设置了恢复标志以指示需要一恢复操作。BSERD 技术还确定了恢复操作可以不需小擦除而完成，因为在 V_{MRUI} 防护带中没有检测到单元，并且恢复高条件标志保留本页地址的设定。

在本发明的一个实施例中，芯片在晚些时候方便地执行一小编程操作以恢复前面已经设置了一个不带小擦除的恢复操作标志的单元页的 V_T 。首先，通过一仅利用数据检测子循环而不用恢复检测子循环的简化 BSERD 技术读取单元，以锁存存储在每个被读取单元中的对应于 V_T 的 N 个二进制位。

在本发明的一个替代实施例中，如果在一个常规的完整 BSERD 读取之后立即执行一小编程操作，那么删除第一个简化的 BSERD 步骤，因为完整 BSERD 技术已经锁存了 N 个二进制位。接下来，将单元的 V_T 与一个新的下小编程容限参考电压 $V_{RI} + V_{MPLI}$ 比较（验证），其中 I 唯一地对应于由锁存的 N 个二进制位代表的电平。这个验证读出与 BSERD 过程中的读取读出不同，因为验证不改变以前锁存的对应于存储在单元中的 V_T 的 N 个二进制位的状态。因此，现在被验证带有低于 $V_{RI} + V_{MPLI}$ 的 V_T 的任何单元是在由值 V_{RI} 和 $V_{RI} + V_{MPLI}$ 定义的下小编程容限防护带 V_{MPLI} 内。

接下来，把一个适当电压和持续时间的小编程脉冲有选择地仅施加于前面验证步骤事先确定了具有在下 V_{MPLI} 防护带内的 V_T 的单元。具有高于 V_{MPLI} 防护带的 V_T 的单元被从小编程脉冲中排除。在第一小编程脉冲之后，执行另一个验证操作。如果任何单元被验证在下 V_{MPLI} 防护带内，另一个小编程脉冲再次有选择地仅施加于那些在 V_{MPLI} 带内的单元。反复进行验证/小编程脉冲序列，直到如图 12 中所示的，所有页内的单元被验证它们具有 V_{MPLI} 防护带以上的 V_T 的时间 T_{RESP} ，或直到已经施加了预定最大数量的小编程脉冲并设置了一系统误差标志的时候。得到的单元 V_T 分布现在呈现如图 12 中所示的在 T_{RESP} 的第二分布所示的状态。

在本例中，恢复操作是由事先被发现处于下读取容限防护带 V_{MPLI} 内的单元，例如图 12 中的单元 A 和 B 最初启动的。具有 V_{MPLI} 内 V_T 的单元也已经被小编程，导致了对整个页的更紧密的 V_T 分布，此外还改正了最初做了标志的 V_{MPLI} 内的单元。通过消除此页上对将来恢复操作的需要，得到的更紧密的页宽 V_T 分布改善了系统的性能。

为检测 V_T 超出范围的条件，在上述的验证/小编程脉冲序列之后执行一个附加的验证步骤。作为选择，该附加步骤可以在每个小编程脉冲之后进行，但将用较长的时间完成。这个附加步骤相对于第二新的小编程上容限参考电压 $V_R(I+1) - V_{MPUI}$ 验证选定小编程页中所有单元的 V_T ，其中 I 也是唯一地对应于由锁存的 N 个二进制位代表的同一电平。此时被验证为它们的 V_T 在电压 $V_R(I+1) - V_{MPUI}$ 以上的任何单元是在由值 $V_R(I+1)$ 和 $V_R(I+1) - V_{MPUI}$ 定义的小编程上容限防护带 V_{MPUI} 内。如果任何单元 V_T 被检测在 V_{MPUI} 防护带内，那么设置一个 V_T 超出范围系统条件标志。此后，一个恢复操作检查是否设置了 V_T 超出范围系统条件标志，并确定是否需要一个 V_T 超出范围恢复操作。 V_T 超出范围恢复操作是包括将在以下本发明的小擦除节中更详细地说明的小擦除操作的一系列操作。

多电平存储器小编程系统

图 7 示出了本发明的一个优选实施例的方框图，并且已经与 BSERD 读出技术一同进行了说明。小编程验证读出操作与读取读出操作的不同在于，编程擦除读取恢复序列发生器块 26 经过 PRGRDB 总线 37 指令 $\Delta VU \Delta VY$ GEN 块 39 从参考电压总线 RVBUS42 产生 V_{MPUI} 和 V_{MPLI} 电压，而不是 $V_{MPU}(I-1)$ 和 V_{MPLI} 电压。还指令 VX ， VY GEN 块 38 产生 V_{RI} 和 $V_{R(I+1)}$ 电压，其中 I 对应于在以前 BSERD 步骤中事先锁存在每个 Y-驱动器 33 内数据[0-3]锁存器块 48 中的 N 个二进制位代表的电平。

在验证操作序列之后，如果验证单元 V_T 在 V_{MPLI} 防护带内，设置每个 Y-驱动器 33 内的 STPDIR 锁存器 52 和 RES 锁存器 53，使 STPDIR 线 47 为高电平。STPDIR 线 47 上的逻辑高信号导通高电压 HVSW 开关 56，高电压 HVSW 开关 56 将高电压编程脉冲从 PRGHV 线 24 输送到 MLLINE 线 30。每个 Y-驱动器 33 内的 Y-MUX 多路复用器 32 将高电压脉冲输送到被 Y-DEC 解码器 18 选定时要编程的单元阵

列 21 中的选定页的适当单元 V_T 在 V_{MPLI} 防护带之外, STPDIR 线 47 是逻辑低。STPDIR 线 47 上的逻辑低信号关断高电压 HVSW 开关 56。这使高电压编程脉冲与 MLLINE 线 30 分离。

5 如果验证任何单元 V_T 在上 V_{MPUI} 防护带内, 在编程擦除读取恢复序列发生器块 26 内设置 V_T 超出条件标志。把对应于 V_T 超出条件标志的地址单元存储在寄存器和地址缓冲器块 60 中。每进行一次验证/小编程脉冲序列, 在编程擦除读取恢复序列发生器块 26 增加一次计数。如果计数超过了一个预定的最大数, 停止对页的小编程, 并在
10 编程擦除读取恢复序列发生器块 26 中设定系统误差标志, 同时在寄存器和地址缓冲器块 60 中设置地址。

III. 小擦除

小擦除操作的一般说明

15 在本节中说明作为本发明一部分的称为小擦除的多电平擦除技术。与破坏存储在单元中的数据的数据的以前的擦除技术不同, 小擦除技术完整地保留最初存储在单元中的数据。在现有技术中, 擦除操作把选定擦除块中的所有单元的 V_T 一直偏移到可用单元 V_T 范围 V_F 的一端, 如图 3A-3C 中所示, 并标为 V_{LERASE} 。这种擦除利用把大量的
20 电荷移动通过存储单元的浮栅周围的氧化物将选定的单元的 V_T 移出一个其它 V_L 电平, 并移入 V_{LERASE} 范围。

 与此相反, 小擦除仅移动少量电荷。如同小编程情况一样, 仅少量地偏移单元的 V_T , 并且是仅在那个单元的原始 V_L 电平界限内移动。
25 小擦除避免了现有擦除对单元 P/E 循环耐久性的有害影响。另外, 仅有改正以前确定的 V_T 漂移所需数量的电荷被移动。小擦除操作比现有技术所作的那样擦除到完全擦除条件所需的时间少。此外, 小擦除仅轻微地影响选定的单元的 V_T , 所以产生的擦除干扰对其它未选定单元的影响可以忽略。小擦除操作可以在一 V_L 电平内以与编程相反的方向少量地偏移 V_T , 并因此执行了一种进行希望的 V_T 恢复以使可靠的多电平数据存储成为可能所需要的功能。
30

 小擦除技术和上述的小编程技术共同工作, 以克服与大尺寸擦除块相关的问题。可以连接在同一擦除块中的大量单元, 导致在一个擦除脉冲后的单元 V_T 的分布比编程脉冲施加于一编程页内较少数量的
35 单元后获得的分布更宽。小擦除技术运用了小编程和小擦除操作的组

合，以减小在一擦除块中所有单元的每一个 V_{LI} 电平内单元 V_T 分布。

与在多电平数字读出一节中所述的相同，小擦除技术利用了本发
5 明的带读出性能，以验证每个单元的 V_T 被偏移到保证适当的容限防
护带的 V_L 范围的恰当带中，以防止在该 V_L 范围的两端模糊读出。
这是通过在小擦除验证过程中的多个参考电压完成的。如图 13A 和 13B
中所示，在每个 V_{LI} 电平中，在 $V_{RI}+V_{MPELI}$ ， $V_{RI}+V_{MELI}$ ，和 V_R
10 $(I+1)-V_{MEUI}$ 的三个附加参考电压，与 V_{RI} 和 $V_R(I+1)$ 结合，
定义了三个小擦除防护带 V_{MPELI} ， V_{MELI} 和 V_{MEUI} 。 V_{MPELI} ，
 V_{MELI} 和 V_{MEUI} 可以等于或不等于，并且可以大于或不大于前面所
述的 V_{MRLI} 和 V_{MRUI} 读取容限防护带，或 V_{MPLI} 和 V_{MPUI} 小
编程容限防护带。适当地选择每个 V_{LI} 电平内的 V_{MPELI} ， V_{MELI} 和
15 V_{MEUI} 容限防护带，使得所得的 V_T 分布在小擦除之后在一个比 V_{LI}
范围窄的恰当的带中被优化。

优化可以要求 V_{MPELI} ， V_{MELI} 和 V_{MEUI} 带互不相等，并且
可以要求每个 V_{LI} 电平的不同带值，或存储单元阵列处理技术确定的
更任意的带值。作为选择，每个电平中的容限防护带可以是等值的。
20 小擦除技术有足够的灵活性，以为各种存储器处理技术提供优化。

本发明的小擦除验证技术与仅相对于一个单一的电压参考值验证
单元 V_T ，并且依赖统计过程控制以防止单元 V_T 低于恰当的 V_T 带的
现有技术不同。由于来自存储单元的浮栅周围的氧化物中的俘获电
25 荷或其它加工缺陷的前述劣等单元效应， V_T 下冲（编程过程中 V_T 过
冲的擦除模拟）在任何擦除操作过程中都是可能的，并可以发生。此
外，小擦除通过仅移动少量电荷通过栅氧化物，使电荷俘获以及产生
劣等单元的可能性减小到最低程度。

30 小擦除验证操作检测是否擦除块中任何单元的 V_T 已经偏移到一个
 V_{MPELI} 防护带之一，并停止进一步小擦除，因而使得小编程技术
可以在晚些时候在那个电平内重新建立恰当的 V_T 。这种不用首先完
成完全擦除和再编程操作，来恢复 V_T 下冲的能力在现有技术中不存
在。利用本发明的恢复操作，在 V_T 下冲恢复过程中避免了完全擦除
35 和再编程，并提高了芯片的 P/E 持续寿命。

擦除过程中单元 V_T 偏移的方向对于一种特定硅技术是一定的。小擦除提供了任何一个方向的擦除 V_T 偏移，并可以在各种存储器技术中使用。为了简洁，结合实例和附图说明小擦除操作，以解释擦除操作致使单元 V_T 减小的情况。熟悉本领域的技术人员可以用类似的方式处理用擦除操作增大 V_T 的相反情况。因此，特定实例和方法不应限于有关擦除极性的一种特定存储单元技术。

小擦除方法

考虑连接在同一擦除块中的多个单元是对小擦除技术的最好说明。一个实际的多位存储系统应当使用减小单元面积的擦除块存储单元结构，因而可以同时擦除许多单元。如果不了解整个芯片，那么必须充分了解至少是这样一个擦除块中的所有单元的行为，以便观察擦除的任何形式的所有可能的分枝。

图 13A 和 13B 示出了在六个不同时间： T_{RESR} ， T_{RESPE1} ， T_{RESE1} ， T_{RESPE2} ， T_{RESE2} ，和 T_{RESP} ，同一电平 V_{LI} 内多个电平的 V_T 分布。在本例中，在时间 T_{RESR} 的 V_T 分布示出了利用前述 BSERD 技术读取单元后各单元的条件。BSERD 技术检测具有 V_{MRLI} 防护带内 A 和 B 处 V_T 值的两个单元，和具有 V_{MRUI} 防护带内 C 和 D 处 V_T 值的另外两个单元。设置恢复条件标志以指示需要恢复操作。BSERD 技术还确定了恢复操作可以用一个小擦除完成，因为检测到这些单元在 V_{MRUI} 防护带内并且恢复低条件保持对这个块地址的设置。

在小擦除技术的一个实施例中，集成电路执行一系列包括小编程和小擦除的操作。前面已经对小编程操作进行了说明。小擦除技术使用小编程操作的两个版本。一个版本与前面说明的小编程相同。前述的 V_{MPLI} 和 V_{MPUI} 帮助进行在小擦除序列终点的最后 V_T 调节。

小编程操作的第二个版本与第一个版本的区别在于使用了一个不同的（预擦除）编程防护带容限 V_{MPELI} 。利用 V_{MPELI} 的小编程操作是在每个小擦除操作之前执行的。可能要进行多次小擦除操作，直到一个预定的小擦除操作的最大次数。以下说明小擦除技术的一个实施例中的操作序列。

在一个实施例中，集成电路在晚些时候利用小擦除方便地执行恢

复操作，以恢复事先已经为用小擦除恢复作出标志的单元的擦除块的 V_T 。首先对要小擦除的擦除块内的所有单元进行小编程，使得擦除块内所有单元的 V_T 都在 I 个电平中每一个的 V_{MPELI} 容限防护带之上。依次用具有设置为 $V_{RI} + V_{MPELI}$ ，而不是 $V_{RI} + V_{MPLI}$ 的下防护带验证值的小编程技术对块中的每一页进行这个预擦除小编程。在时间 T_{RESPE1} 首先完成了对块中所有页的小编程。图 13A 中示出了在每个 V_{LI} 电平内的和擦除块内所有单元的所得的单元 V_T 分布的一个例子。通过这个首先完成的小编程操作，太靠近一个 V_R 值的 V_{MPELI} 带内的单元，例如在时间 T_{RESR} 的单元 A 和 B，在小擦除操作中可能被偏移到相邻的 V_L 范围中。这将导致数据丢失。预擦除小编程防止数据丢失，如果擦除块中的一个单元 V_T 在 V_{MPELI} 带内，数据丢失可能会发生。

接下来，对擦除块中的所有单元施加一个小擦除脉冲。小擦除脉冲把擦除块中所有单元的 V_T 向擦除方向（与编程方向相反的方向）偏移。接着对擦除块（由多个页组成的）中的每个单元进行一个小擦除验证操作，因而为一个擦除验证操作选定了擦除块中的第一页。接下来通过仅利用数据检测子循环的简化 BSERD 技术读取各单元，以锁存对应于存储在单元中的 V_T 的 N 个二进制位。然后将擦除块的被选定页中的单元的 V_T 与一个新的下小擦除容限参考电压 $V_{RI} + V_{MELI}$ 比较（验证），其中 I 唯一地对应于锁存的 N 个二进制位代表的电平。

由于验证不改变对应于存储在单元中的 V_T 的以前锁存的 N 个二进制位的状态，因此这个验证读出不同于上面的读取读出。现在被验证的具有低于 $V_{RI} + V_{MELI}$ 的 V_T 的单元位于由值 V_{RI} 和 $V_{RI} + V_{MELI}$ 定义的下小擦除防护带 V_{MELI} 内。如果验证没有单元在第一页的 V_{MELI} 防护带内，那么选择下一页，并进行同样的读取和验证。如果验证在整个擦除块没有单元在 V_{MELI} 防护带内，那么向擦除块施加另一个小擦除脉冲，并重复上面的读取和验证操作。

当在时间 T_{RESE1} 的擦除块验证过程中发现有选定擦除块的任何页中的任何单元在 V_{MELI} 防护带内时，或达到一预定小擦除脉冲的最大次数时，终止小擦除脉冲/擦除块验证步骤的序列，并设置一所得误差标志。图 13A 中示出了在时间 T_{RESE1} 的单元 V_T 分布的一个例子。这个反复的小擦除脉冲/擦除块验证操作序列被用来同时地以小

的增量调节擦除块中所有单元的 V_T ，并且以这样的方式把所有单元的 V_T 维持在一个比原始 V_{LI} 范围更窄的带内。因此防止了单元 V_T 下冲和数据丢失。

5 一个更快的替代实施例使用了没有 V_{MELI} 防护带验证操作的单个小擦除脉冲。依赖擦除块中所有单元的擦除特性更紧密的过程控制来防止数据丢失。要注意图 13A 显示了单元 V_T 分布的终端，在擦除方向侧有一个长的尾部。这是由于在擦除块中的大量单元中有一些单元的小擦除比其它的快。此时在擦除方向上尽可能远地偏移了擦除块的 V_T 分布，没有 V_T 下冲也没有任何数据丢失。最初启动小擦除的
10 单元（具有图 13A 中的 C 和 D 的 V_T ）现在已经在恰当的方向上偏移了一个在图 13A 中示为 V_{DELE1} 的量。

 下一个验证操作对在前面的预擦除小编程操作过程中表现出 V_T
15 下冲行为的任何单元，例如图 14A 中的在时间 T_{RESPE1} 的单元 E，进行检查，并且检查是否需要更多的小擦除。再次对擦除块中的每个单元进行这种小擦除验证操作，因而为小擦除验证操作选定了擦除块中的第一页。接着用仅有数据位检测子循环的，即没有恢复检测子循环的简化 BSERD 技术读取各单元，以锁存对应于存储在单元中的 V_T
20 的 N 个二进制位。然后把各单元的 V_T 与一个新的上小擦除容限参考电压 $V_{R(I+1)} - V_{MEUI}$ 比较，其中 I 唯一地对应于由锁存的 N 个二进制位代表的电平。

 由于验证操作不改变对应于存储在单元中的 V_T 的以前锁存的 N
25 个二进制位的状态，这种验证读出操作也与前面所述的读取读出不同。因此，此时验证具有在 $V_{R(I+1)} - V_{MEUI}$ 之上的 V_T 的任何单元处于由值 $V_{R(I+1)}$ 和 $V_{R(I+1)} - V_{MEUI}$ 定义的上小擦除防护带 V_{MEUI} 之内。如果验证没有单元在第一页的 V_{MEUI} 防护带内，那么选择下一页，并进行类似的读取和验证。如果验证在整个擦除块
30 中有任何单元在 V_{MEUI} 防护带内，那么再次对该擦除块进行上述的预擦除小编程/小擦除/验证操作序列并设置所得的误差标志，除非是达到了这种序列的预定最大次数。

 如果在验证过程中检测到有任何单元在 V_{MEUI} 防护带内，那么
35 将这些单元的地址位置与最初启动小擦除的单元的以前存储的地址位置，例如图 13A 中的 C 和 D，比较。如果检测到新地址，例如图 14A

中的 E, 那么设置 V_T 下冲条件标志, 并存储页地址。图 13A 和 13B 中所示的例子需要两个上述序列, 这是由于在 C 和 D 的两个单元在时间 T_{RESE1} 仍然处于 V_{MEUI} 防护带内。在这个例子中, 第二预擦除小编程在时间 T_{RESPE2} 完成, 紧接着第二小擦除脉冲系列, 导致单元 C 的 V_T 在时间 T_{RESE2} 偏移了量 V_{DELE2} 。图 13B 中示出了得到的单元 V_T 分布。

在上述擦除块验证过程中, 在时间 T_{RESE2} 发现没有单元在 V_{MEUI} 防护带内之后, 开始最后的小编程操作, 如图 13B 中的例子所示。除了依次对擦除块内的每页进行编程之外, 用小编程一节中所述的同样方式进行最后的小编程操作。在最后的小编程操作过程中, 如果检测到有任何单元 V_T 在上 V_{MPUI} 防护带内, 那么设置一 V_T 过冲系统条件标志, 并存储该页地址。在时间 T_{RESP} , 最后的小编程操作存储 V_{MPLI} 和 V_{MPUI} 防护带之间每个 V_L 电平的整个擦除块的单元 V_T 分布, 如图 13B 中的例子所示。

在任何页上开始任何小编程操作之前检查 V_T 超出系统条件标志。每当在要进行小编程的一页遇到这种标志设置时, 改变小编程脉冲序列, 以用每个脉冲产生一个小于正常增大的单元 V_T 偏移, 称为微编程。验证/微编程脉冲序列完成之后的选定页中的每个电平内的单元 V_T 分布比验证/小编程脉冲序列之后的更紧密。微编程一个标志页要比小编程花费更长的时间, 但是由于仅有必要的标志页被微编程, 整个系统的性能得到优化。

在一个替代的微编程操作中, 用 V_{CPELI} , $V_{CP LI}$ 和 V_{CPUI} 防护带替换小编程过程中使用的 V_{MPELI} , V_{MPLI} 和 V_{MPUI} 防护带, 因而优化了每个 V_L 范围内得到的单元 V_T 分布 (见图 14A 和 14B)。如同小编程技术中一样, 如果验证/微编程脉冲序列的数量超过一预定的数, 停止微编程, 并设置一系统误差标志。在标志页成功地微编程后, V_T 超出系统条件标志被复位。在没有小擦除的恢复过程中设置 V_T 超出系统条件标志立即简单地开始对那个擦除块的小擦除操作, 或作为选择, 在稍后方便的时间进行。与微编程结合使用的小擦除技术用作 V_T 超出恢复的方法。

35 多电平存储器的小擦除系统

图 7 的编程擦除读取恢复序列发生器块 26 与二进制检索读取序列

发生器 43 一同执行所有必要的控制和事项的排序，以进行小擦除操作。图 11 示出了一个到单元阵列 21 的特殊接口。

例如，为一个小擦除操作选择了 BLOCK0 块 83，因为为这个擦除块中的一个地址先期设定了一个恢复低条件。在图 13 中时间 T_{RESR} ，编程擦除读取恢复序列发生器 26 根据恢复条件标志开始小擦除操作。如图 11 中所示，每一页有 J 个单元，每一行有 S 页，每一块有 T 行。因此，每一块有 $(S \times T)$ 页，和每一块有 $(S \times T \times J)$ 单元。利用图 13A 实例，首先在 BLOCK0 块 83 的所有页上进行预擦除小编程操作。对于 BLOCK0 块 83 的每页，首先进行一没有恢复检测子循环的简化 BSERD 技术，以同时从选定页中的所有 J 个单元的每个单元 20 读取 4 位数据。在读取操作完成后，每个 Y-驱动器 33 内的数据[0-3]锁存器块 48 包含选定页的相应 J 个单元的每单元数据的独立的 4 位。

15

接下来，编程擦除读取恢复序列发生器块 26 设定 PRGRDB 总线 37。根据来自每个 Y-驱动器 33 的数据[0-3]锁存器块 48 的输出，从 VX VY GEN 块 38 产生适当的 V_{RI} 电压作为 VX 输出，和产生 $V_R(I+1)$ 电压作为 VY 输出。 $\Delta V_X \Delta V_Y$ GEN 块 39 产生 V_{MPELI} 作为 ΔV_L 输出。在所有的预擦除小编程操作过程中，VCRFGEN 块 40 在 VCRF 线 45 上产生电压 $(V_{RI} + V_{MPELI})$ 。现在用一个反复的小编程脉冲发生和验证操作的序列进行一预擦除小编程操作，直到选定页内所有 J 个单元都具有高于 $(V_{RI} + V_{MPELI})$ 电压的 V_T 为止。然后，编程擦除读取恢复序列发生器块 26 和二进制检索读取序列发生器块 43 继续进行第一 WL00 线 85 的所有 S 个页的预擦除小编程。这样继续到全部具有 S 页的 T 行都被预擦除小编程，使所有单元 V_T 都被编程到高于 $(V_{RI} + V_{MPELI})$ 电压。这样完成了选定的 BLOCK0 块 83 的预擦除小编程。图 13A 中在时间 T_{RESPE1} 示出了 BLOCK0 的单元 V_T 的典型分布。

30

强制 PRG-ERS 总线 22 设置 BLOCK-DEC 块 19，X-DEC 块 17 和 Y-DEC 块 18，以允许在选定的存储块或区段上进行擦除操作。接着，对选定 BLOCK0 块 83 中的所有单元 20 施加小擦除脉冲。小擦除脉冲是由 HVGEN 块 27 产生的，并经 HVOUT 总线 25 传送。

35

在施加小擦除脉冲之后，在选定的 BLOCK0 块 83 中的所有单元

20 上进行验证操作。不带恢复检测子循环的 BSERD 技术把来自一页中每个单元 20 的数据的 4-位锁存在每个 Y-驱动器 33 内的数据[0-3]锁存器块 48 中。根据来自每个 Y-驱动器 33 内的数据[0-3]锁存器块 48 的输出, VX VY GEN 块 38 产生 V_{RI} 电压作为 VX 输出, 和 $V_{R(I+1)}$ 电压作为 VY 输出。 $\Delta V_X \Delta V_Y$ GEN 块 39 产生 V_{MELI} 电压作为 ΔV_L 输出。

在所有的擦除验证操作过程中, VCRFGEN 块 40 在 VCRF 线 45 上输出电压 ($V_{RI} + V_{MELI}$) , 该电压是由来自每个 Y-驱动器 33 的数据[0-3]锁存器块 48 唯一定义的。然后, 编程擦除读取恢复序列发生器块 26 和二进制检索读取序列发生器块 43 在选定的 BLOCK0 块 83 的所有 T 行的所有 S 页上进行小擦除验证操作。小擦除验证操作针对每个 Y-驱动器 33 将 VMEM 线 31 上的电压与 VCRF 线上的电压 ($V_{RI} + V_{MELI}$) 比较。如果在小擦除验证操作过程中, VMEM 线 31 低于 VCRF 线 45, 那么在编程擦除读取恢复序列发生器块 26 中设置一个条件标志。如果没有设置条件标志, 在选定的 BLOCK0 块 83 上进行另一个小擦除脉冲和小擦除验证操作。每当设置了一个条件标志, 或到达小擦除脉冲操作的最大计数时, 编程擦除读取恢复序列发生器块 26 就停止小擦除/验证操作。图 13A 示出了在时间 T_{RESE1} , 设置了条件标志后的单元 V_T 的分布。

接下来, 在选定的 BLOCK0 块 83 的所有单元上进行过冲单元验证操作。编程擦除读取恢复序列发生器块 26 和二进制检索读取序列发生器块 43 在选定的 BLOCK0 的所有 T 行的所有 S 页上进行过冲单元验证操作。在整个过冲单元验证过程中, VCRF 线 45 上的电压被设定至 $V_{R(I+1)} - V_{MEUI}$, 这个电压唯一地取决于每个 Y-驱动器 33 内的数据[0-3]锁存器块 48 的输出。对于每一页, 和对于每个 Y-驱动器 33, 如果 VMEM 线 31 上的电压低于 VCRF 线 45 的电压, 那么过冲验证继续到下一页。如果任何时间, 对于一选定页, 对于任何 Y-驱动器 33, VMEM 线 31 上的电压高于 VCRF 线 45 的电压, 那么把那些单元 20 的地址存储在寄存器和地址缓冲器块 60 中。

过冲条件标志的单元的地址与最初启动小擦除条件标志的单元的地址匹配。如果发现一个地址不匹配, 那么为那一页, 在寄存器和地址缓冲存储器块 60 中与该页的地址一起设置过冲条件标志。如果地址

匹配, 那么进行另一个预擦除小编程操作。图 13A 和 13B 示出了在时间 T_{RESPE2} , 第二预擦除小编程操作之后的单元 V_T 的分布。

然后, 施加另一个小擦除脉冲, 和进行小擦除验证。图 13B 在时间 T_{RESE2} 示出了在第二小擦除/验证操作之后的单元 V_T 的分布。如果在 V_{MEUI} 防护带验证操作之后没有设置过冲条件标志, 那么进行选定 BLOCK0 块 83 中所有页的最后小编程操作。在对每页的最后小编程操作的开始, 编程擦除读取恢复序列发生器块 26 为寄存器和地址缓冲器块 60 中的页进行一个 V_T 过冲条件标志检查。如果没有发现标志, 那么进行一个常规的小编程操作。图 13B 在时间 T_{RESP} 示出了在进行了最后常规小编程操作之后单元 V_T 的分布。每当发现了针对一个页的一个条件标志时, 就在该页上进行一个微编程操作。微编程除了以下几项以外, 最好与小编程相同: 1) 编程擦除读取恢复序列发生器块 26 经过 PRGRDB 总线 37 指令 $\Delta V_U \Delta V_Y$ GEN 块 39 产生 V_{CPELI} , V_{CPUI} 和 V_{CPLI} 电压, 而不是从模拟电压总线 RVBUS42 产生 V_{MPELI} , V_{MPUI} 和 V_{MPLI} 电压, 和 2) 编程擦除读取恢复序列发生器块 26 经过 HVCTRL23 总线指令 HVGEN 块 27 将小编程脉冲改变为一个微编程脉冲, 因而使得单元 20 的 V_T 偏移比小编程操作过程中小。

IV. 编程

编程操作的一般说明

说明过的集成电路也通过一种新的多电平编程技术操作。这种编程技术不同于前面说明的小编程技术。这种编程把新数据存储到以前擦除的页中, 而小编程恢复现存的数据而不改变以前存储的数据。与小编程相比, 编程技术移动更大量的电荷通过浮栅周围的氧化物, 并可以把单元 V_T 偏移更大的量。与以前仅在一个(编程)方向上偏移单元 V_T 的编程技术不同, 本发明的编程技术在编程和擦除两个方向上偏移单元 V_T 。

这种以前的多电平存储器编程要求一个十分紧密的单元 V_T 分布。编程后每个电平内很宽的容限允许将来 V_T 漂移, 这种漂移将在存储器芯片的整个寿命中发生。图 15A 在时间 T_P 示出了紧接着在 PV_I 的单一容限值的现有技术编程之后的 V_L 电平内单元 V_T 的初始紧密分布的一个实例。同一图中, 在时间 T_F 还示出了在一长时间周期之后的分布, 在这个时间周期中 V_T 漂移效应实际上已经把初始紧密的

分布变宽到模糊读出故障点。

5 与此相反，本发明允许使用每个电平内从初始编程步骤得到的很宽的单元 V_T 分布，可靠地将数据存储一段短时间（见图 15B 时间 T_P ），直到存储器可以在最近的将来执行一恢复操作（如图 15 中在时间 T_{RESP} 所示）。此外，由本发明的单元 V_T 恢复技术保持和说明的容限（见图 15B 时间 T_P ）不如未使用本发明的编程过程中要求的容限宽。现有技术中要求的紧密编程的 V_T 分布大大地延长了编程时间，因为在证明成功地到达恰当的参考电压之前需要许多验证/编程脉冲序列以小递增的方式偏移 V_T 。在大多数系统应用中，从外部系统的观点看，较长的编程时间是极不可取的。另一方面，通过使外部系统提供的编程算法使用每一电平内的宽单元 V_T 分布，本发明缩短了多电平存储的编程时间。

15 本发明的一个实施例与以前技术的不同在于，需要为所有 V_{LI} 电平设定单元 V_T 的最后操作是一个编程操作，而不是现有技术中使用的为电平之一在 V_F 范围的任意一端的擦除操作。

20 本发明把编程循环分为两个部分。第一部分初始地把数据很快地存储在单元中，这些单元对于有关每个电平的 V_R 参考值仅有小的容限。小的容限适合于在长期 V_T 漂移效应造成数据丢失之前暂时可靠地存储数据。从外部系统观点看第一部分以尽可能短的时间优化编程时间。对于外部系统来说，整个编程时间仅仅是在第一部分中进行数据存储所需的时间。第一部分进行大部分电荷移过栅氧化物和大部分单元的 V_T 偏移操作。

25 第二部分与上述用小擦除操作恢复相同。由于数据在第一部分中已经存储在单元中，第二部分可以由外部系统自动地完成。第二部分可以在外部系统觉察编程周期结束之后，否则存储器芯片空闲周期中进行。编程循环的第二部分建立有关 V_R 值的较宽容限，这是优化系统操作和长期可靠性所需要的。第二部分编程循环用的时间比第一部分长。

30 在本发明的一个实施例中，第二部分是紧接着第一部分之后发生的。在另一个实施例中，较慢的第二部分延迟到第一部分之后晚一些时间，但是使用了外部系统看到的编程时间大大缩短的这样一种方

式。

第一部分编程操作利用了本发明的带读出的能力，与多电平数字读出一节中所述的相同。验证每一个单元的 V_T 已经偏移到 V_L 范围的恰当带中，以保证适当的容限防护带，防止在那个 V_L 范围的两端模糊读出。在编程验证操作中使用了多个参考电压。如图 15B 所示，在每个 V_{LI} 电平内，第一部分编程技术使用了在 $V_{RI} + V_{R(I+1)}$ 和 $V_{R(I+1)} - V_{PUI}$ 的参考电压，它们与 V_{RI} 和 $V_{R(I+1)}$ 结合定义了两个新的下和上编程容限防护带 V_{PLI} 和 V_{PUI} 。 V_{PLI} 和 V_{PUI} 可以等于或不等于，并且可以大于或不大于前面所述的 V_{MRLI} 和 V_{MRUI} 读取，或 V_{MPLI} 和 V_{MPUI} 小编程，或 V_{MPELI} ， V_{MELI} 和 V_{MUI} 小擦除，或 V_{CPELI} ， V_{CPLI} 和 V_{CPUI} 微编程防护带。每个 V_{LI} 电平内的 V_{PLI} 和 V_{PUI} 防护带是这样选择的，使第一部分编程之后得到的 V_T 分布被优化在一个比 V_L 范围窄的恰当的带内。优化可以要求 V_{PLI} 和 V_{PUI} 带彼此不相等，和可以要求对每一个 I 电平有不同的带值，或由存储单元处理技术规定的更为任意的带值。作为选择，在每个电平中的带可以是等值的。编程技术有足够的灵活性为各种存储器处理技术提供优化。

本发明的验证技术与以前仅相对于每个电平一个单一的参考电压验证单元 V_T 的编程技术不同。在现有技术中，可以这样为单元编程，使得 V_T 如此地接近一个 $V_{R(I+1)}$ 参考值，以致产生了上述模糊读出问题。通过在利用每个电平两个附加验证参考电压的优化带内验证编程的数据，本发明防止了利用以前的技术可能产生的模糊数据读出的问题。

编程方法

首先，对于每个单元，把新数据锁存在要编程的页中，并转换为 2^N 个 V_L 电平中的一个。在本发明的一个实施例中，集成电路执行第一部分的编程序列。将单元的 V_T 与一个新的下编程容限参考电压 $V_{RI} + V_{PLI}$ 比较（验证），其中 I 唯一地对应于由锁存的 N 个二进制位代表的电平。该验证读出技术与前述的读取和验证读出不同，并且与现有技术的不同在于：由值 V_{RI} 和 V_{PLI} 定义的 V_{PLI} 防护带的值大大小于任何一个前者所能允许的值。本发明允许这样小的 V_{PLI} 编程容限防护带，是由于：1) 为了长期的 V_T 稳定性第二部分编程序列增大了容限，和 2) 在紧接着第一部分完成之后，明显的 V_T 漂移能

够发生之前，执行第二部分。由于验证不改变对应于存储在单元中的 V_T 的以前锁存的 N 个二进制位的状态，这个验证读出不同于读取读出操作（见前述的 BSERD 技术）。

5 接着，仅对选定页中选定的单元施加适当电压和持续时间的一个编程脉冲。仅对由前面的验证步骤事先确定了具有低于 $V_{RI} + V_{PLI}$ 电压的 V_T 的那些单元有选择地施加编程脉冲。把具有高于 $V_{RI} + V_{PLI}$ 电压的 V_T 的单元从编程中排除。在第一编程脉冲之后，执行另一个验证操作。如果验证有任何单元低于 $V_{RI} + V_{PLI}$ 电压，再次仅对具有
10 低于 $V_{RI} + V_{PLI}$ 电压的 V_T 的单元有选择地施加另一个编程脉冲。

 考虑有多个连接在同一页中的单元，最好地说明编程技术。图 16A 和 16B 示出了在六个不同时间 T_P ， T_{RESPE1} ， T_{RESE1} ， T_{RESPE2} ， T_{RESE2} 和 T_{RESP} 的，在相同电平 V_{LI} 内的存储器中
15 多个单元的 V_T 分布。重复验证/编程脉冲序列直到验证了该页内的所有单元的 V_T 都高于 $V_{RI} + V_{PLI}$ 电压的时间 T_P （图 16A 所示），或直到已经施加了预定最大数量的脉冲，和设置了系统误差标志。

 一旦第一页成功地编程，选择要编程的下一个擦除的页，并重复
20 这个过程直到最后的页。要编程的页必须在以前被擦除的块或区段中。存储其上进行了第一部分编程操作的每个擦除块的地址，因而可以在以后进行第二部分的编程操作。这个第一部分编程过程的结束确定了从外部系统看的编程时间。利用本发明明显地减少了多电平编程时间，因为在编程的这一部分使用的编程脉冲不需要每个 V_L 电平内的一个单元 V_T 分布象现有技术要求的那样紧密。
25

 本发明在上述的验证/编程脉冲序列之后增加了一个额外的验证步骤。作为选择，新的验证步骤可以在每个编程脉冲之后进行。当然，这种变化将占用更多的时间。额外步骤对照一个第二新的编程容限参考电压 $V_{R(I+1)} - V_{PUI}$ 验证刚刚编程的选定页中所有单元的 V_T ，
30 其中 I 仍然是唯一对应由锁存的 N 个二进制位代表的同样的电平（见图 16A）。现在已验证它们的 V_T 高于电压 $V_{R(I+1)} - V_{PUI}$ 的任何单元被过编程，并设置一个系统误差标志。验证操作保证了通过验证的被编程单元的 V_T 设置在一个带内，这个带比 V_L 范围窄，并建立了上和
35 下编程容限 V_{PLI} 和 V_{PUI} 。避免了当一个位编程太接近 $V_{R(I+1)}$ 电压参考值时的模糊读出的可能性。与现有技术相比，现有

技术仅建立一个下容限。

在上述第一部分编程序列之后的方便时间，集成电路利用小擦除操作相继地在所有刚刚编程的擦除块上进行恢复。图 13A 在时间 T_{RESR} 示出了在用小擦除恢复之前的单元 V_T 分布，在图 16A 中的时间 T_P 是这些操作的结果的图示。

因此，第二部分编程序列完全是通过前面所述的利用小擦除技术的恢复完成的。图 16A 和 16B 示出了这些步骤，并且对应于图 13A 和 13B 所示的在时间 T_{RESPE1} ， T_{RESE1} ， T_{RESPE2} ， T_{RESE2} 和 T_{RESP} 之后用小擦除操作恢复的同样的步骤。在第一部分编程中用最小化防护带存储的单元 V_T 有时间漂移之前已恢复了完整的防护带 V_{MPLI} 和 V_{MPUI} 。现在可以看到得到的单元 V_T 分布，如图 16B 在时间 T_{RESP} 所示的最后分布所示。

作为选择，集成电路可以方便地在迟一些的时间执行第二部分编程操作（用小擦除恢复），以便恢复以前为第二部分编程操作标志过的单元的擦除块的 V_T 。

在另一种变化中，在以前第一部分编程的擦除块上的第二部分编程紧随第一部分编程步骤之后进行。在任一种情况下，第二部分编程操作自动地进行，因为所有的数据已经存储在存储单元中。因此，自动执行的本发明的第二部分编程序列不会增加能由外部系统感觉到的编程时间。

多电平存储器的编程系统

用于编程的优选实施例已经与 BSERD 读出技术、小编程技术和小擦除技术一同进行了说明。首先，在图 7 中，来自一外部源的数据通过数据、地址和控制系统接口块 10 装载到每个 Y-驱动器 33 内的数据[0-3]锁存器块 48 中。也提供用于存储数据的地址位置。将数据编程到已经被擦除的块中。第一部分编程与前述小编程技术非常相似。此时，编程擦除读取恢复序列发生器块 26 经过 PRGRDB 总线 37 指令 ΔVU ΔVLG EN 块 39 产生 V_{PUI} 和 V_{PLI} 参考电压，而不是从参考电压总线 RVBUS42 产生 V_{MPUI} 和 V_{MPLI} 电压。指令 $VX VY$ GEN 块 38 产生 V_{RI} 和 $V_{R(I+1)}$ 参考电压，其中 I 对应于由以前装载的并锁存在每个 Y-驱动器 33 中的数据[0-3]锁存器块 48 中的 N 个二进制位代表的电

平。编程/验证和超出验证操作与以前说明过的小编程技术中适用的那些一样。编程技术的第二部分与前面说明的小擦除技术完全一样。

V.擦除

5 擦除操作过程一般说明

集成电路也进行一种与上述小擦除技术不同的多电平擦除技术操作。擦除破坏以前在擦除块或区段中的数据，而小擦除恢复现存数据却不改变以前存储的数据。擦除也移动大量的电荷通过浮栅周围的氧化物，并能以比小擦除过程中出现的更大量地偏移单元 V_T

10

与以前仅在一个（擦除）方向上偏移单元 V_T 的擦除技术不同，本发明的擦除技术可以在编程和擦除两个方向上偏移单元 V_T 。擦除多电平存储器存储的现有技术在擦除块中产生所有单元非常宽的单元 V_T 分布。在以前的技术中已经把宽擦除的单元 V_T 分布 V_{LERASE} 用作多电平存储的 2^N 个 V_L 范围中的一个。但是，本发明的擦除技术不使用宽擦除的单元 V_T 分布作为 2^N 个 V_L 范围之一。优化所有的 V_L 范围，使得 V_{L0} 范围在宽度上与其它任何一个 V_L 范围一样，甚至同时迭盖 V_{LERASE} 范围。

15

20

图 17A-17C 示出了用于每单元四位多电平存储的一个擦除块内的所有单元 V_T 的长期数据存储分布的实例。另外，图 17A-17C 都享有同样的存储阵列处理技术，因此具有完全相同的：1) 用于编程和擦除的有用单元 V_T 的范围，表示为 V_F ，和 2) V_{LERASE} 范围。

25

但是，这些图在擦除技术的类型和所得的 V_R 值， V_L 范围和 V_M 范围彼此不同。图 17A 示出了用 V_{LERASE} 范围作为 V_{L0} 电平（ V_{R0} 和 V_{F1} 参考值之间）的实例，与前面使用的技术类似。图 17A 中示出了十六个电平。每个电平由 V_M 分离；十五个电平为 V_L 宽；一个电平为 V_{LERASE} 宽； $V_{LERASE} \gg V_L$ ；和 $V_F = V_{LERASE} + 15V_M + 15V_L$ 。图 17B 示出了本发明的一个实施例，它优化整个 V_F 范围的利用是通过：1) 不用 V_{LERASE} 范围作为 V_{L0} 电平，和代之以，2) 在整个 V_F 范围上分布实际上相等的 V_L 。图 17B 中示出的十六个 V_L 电平由 V_M 分离，并且 $V_F = 16V_L + 15V_M$ 。由于 V_F 相等，用于图 17B 中的实例的 $V_L + V_M$ 大于（因而更容易制造和控制）图 17A 中的。因此，图 17B 中所示的实施例优于图 17A 中所示的技术，并且使高密度多电平数据存储成为可能。

30

35

图 17C 中示出了另一种变化。与图 17B 中的分布整个 V_F 范围上的 V_L 电平不同，图 17C 中的电平不迭盖至 V_{LERASE} 范围中。这并没有带来宽 $V_L + V_M$ 的利益，但是在一些技术中更可靠的数据读出可能是需要的。此外，如前面小擦除一节中所述，集成电路的擦除技术提供了擦除 V_T 偏移的任一方向。本发明可用于各种存储阵列处理和多电平存储的读出电路。

与多电平数据读出一节中所述的一样，擦除操作利用带读出的性能，验证每个单元的 V_T 被偏移到 V_L 范围的恰当带中。这保证了适当的容限以防止在该 V_L 的两端模糊读出。

在擦除验证过程中，使用了多个参考电压。在 $V_{R0} + V_{MEFL}$ 和 $V_{R1} - V_{MEFU}$ 的参考电压与 V_{R0} 和 V_{R1} 结合，在 V_{L0} 内定义了两个新的下和上擦除容限防护带 V_{MEFL} 和 V_{MEFU} ，如图 18A 和 18B 中所示。使用了在 V_{EA} ， V_{EB} 的擦除参考电压和在 V_{PEB} ， V_{PEF} 的编程参考电压。如下所述，还可以使用其它电压。 V_{MEFL} 和 V_{MEFU} 可以等于或不等于，或可以大于或不大于前面所述的 V_{MRLI} 和 V_{MRUI} 读取、或 V_{MPLI} 和 V_{MPUI} 小编程、或 V_{MPELI} ， V_{MELI} 和 V_{MUI} 小擦除、或 V_{CPELI} ， $V_{CP LI}$ 和 V_{CPUI} 微编程、或 V_{PLI} ， V_{PUI} 编程防护带。

选择 V_{L0} 内的 V_{MEFL} 和 V_{MEFU} 容限防护带，以使擦除技术后所得的 V_T 分布在一个比 V_{L0} 范围窄的恰当的带内被优化。优化可能需要 V_{MEFL} 和 V_{MEFU} 带互不相等。作为选择，带可以是相等值。

优化这种技术以提供各种存储器处理技术。验证技术与现有的仅相对于一个单一擦除参考电压验证单元 V_T 的现有技术不同。在以前的技术中，可以这样擦除单元，使得 V_T 如此地接近 V_{R0} 或 V_{R1} ，致使发生了前述的模糊读出问题。通过利用附加验证参考电平验证擦除数据在优化带内，本发明的擦除技术防止了模糊数据读出问题。

擦除方法

考虑连接在同一个擦除块中的多个单元，最好地说明擦除技术。如上所述，实际的多电平存储系统使用了一种用于同时擦除许多单元的擦除块存储单元结构以减小单元面积。如果不了解整个芯片，那么

必须充分了解至少是这样一个擦除块中的所有单元的运行情况，以便观察任何形式擦除的所有可能的分枝。

图 18A 和 18B 示出在七个不同时间 T_{PE} , T_{E1} , T_{EPE2} , T_{E2} , T_{EPEF} , T_{EF} , 和 T_P , 四个电平 V_{L0} 至 V_{L31} 内的存储器中多个单元的 V_T 分布。图 18A 和 18B 遵循图 17B 中的实例。在这个实例中， V_{LERASE} 迭盖四个 V_L 范围。但是，不应当认为该技术限于这个特定的实例。

在时间 T_{PE} 的 V_T 分布示出了刚好在擦除前的单元的条件。该擦除方法在许多方面与小擦除方法相同，并且由于使用了许多相同的原理，不再特别详细地说明。首先，对选定的区段施加一个擦除脉冲。然后根据存储阵列处理的特性，用两种方法中的任意一种验证区段中的所有单元。第一种方法验证所有单元低于 V_{EA} 参考电压，而第二种方法验证所有单元高于 $V_{R0} + V_{MEFL}$ 容限参考电压。进行随后的擦除脉冲/验证序列，直到所有单元验证低于 V_{EA} 参考电压，或直到验证第一个单元在 V_{MEFL} 容限防护带内，或是直到达到了一个序列的最大数(误差标志)。图 18A 中在时间 T_{E1} 示出了所得的单元 V_T 分布。

接着，进行交替地使擦除单元分布靠紧和把擦除单元分布向 $V_{R0} + V_{MEFL}$ 容限参考电压偏移的一系列预擦除小编程/小擦除步骤。这种预擦除小编程/小擦除序列的数量同样取决于构造存储器的工艺，并且通过适当地选择参考电压的值使之减到最小。

图 18A 和 18B 中的实例分别示出了在时间 T_{EPE2}/T_{E2} 和 T_{EPEF}/T_{EF} 的两个这种预擦除小编程/小擦除序列。在图 18A 中所示的实例中，第一预擦除小编程参考电压设定在 V_{PEB} 。进行在 V_{PEB} 的小编程脉冲/验证操作直到验证所有单元高于 V_{PEB} 参考电压。在时间 T_{EPE2} 示出了从这个第一预擦除小编程步骤得到的单元 V_T 分布。接下来，进行小擦除脉冲/验证操作直到验证所有单元都低于 V_{EB} 参考电压，或直到验证第一单元在 V_{MEFL} 容限防护带内，或是直到到达序列的一个最大数量(误差标志)。在以前的 V_{EA} 和 $V_{R0} + V_{MEFL}$ 参考电压之间择优地选择 V_{EB} 参考电压。图 18A-18B 中的时间 T_{E2} 示出了得到的单元 V_T 分布。

再重复预擦除小编程/小擦除步骤。接着，第二预擦除小编程参考电压设定在 V_{PEF} 。在以前的 V_{PEB} 和 $V_{R0} + V_{MEFL}$ 参考电压之间择优地选择 V_{PEF} 参考电压。进行在 V_{PEF} 的小编程脉冲/验证操作直到验证所有的单元高于 V_{PEF} 参考电压。在时间 T_{EPEF} 示出了从这个第二预擦除小编程步骤得到的单元 V_T 分布。接着，进行小擦除脉冲/验证操作，直到验证所有单元低于 $V_{R1} - V_{MEFU}$ 防护带参考电压，直到验证第一单元在 V_{MEFL} 容限防护带内，或是直到到达一最大数量（误差标志）。

此时，不是进行另一个小编程步骤，已经预定是使用 $V_{R1} - V_{MEFU}$ 防护带参考电压的进行验证时间。如果验证没有单元在 V_{MEFU} 防护带内，或如果已经进行了最大次数的擦除步骤（误差标志），那么擦除操作完成。作为选择，只有在前面的小擦除验证步骤结束验证单元高于 $V_{R1} - V_{MEFU}$ 防护带参考电压时，利用下一个 V_{PE} 参考电压执行附加预擦除小编程/小擦除序列。下一个 V_{PE} 参考电压是在前面的 V_{PE} 和 $V_{R0} + V_{MEFL}$ 参考电压之间择优选择的。图 18B 中的时间 T_{EF} 示出了第二小擦除此后得到的单元 V_T 分布，并示出了验证低于 $V_{R1} - V_{MEFU}$ 防护带参考电压的所有单元。擦除操作完成。在图 18B 中的时间 T_{EF} 示出了得到的单元 V_T 分布，它的宽度与用前述的第一部分编程技术获得的相似。现在擦除块已经可以用来编程没有保留在 V_{L0} 电平中的单元。图 18B 中时间 T_P 示出了前述第一部分编程技术后的单元 V_T 分布。

多电平存储器擦除系统

擦除操作与前面所述的小擦除技术的操作十分类似。不同之处是，编程擦除读取恢复序列发生器块 26 经过 PRGRDB 总线 37 指令 ΔVU ΔVL GEN 块 39 产生 V_{MEFU} 和 V_{MEFL} 参考电压，而不是来自参考电压总线 RVBUS42 的 V_{MEUI} 和 V_{MELI} 电压。指令 $VX VY$ GEN38 产生 V_{R0} 和 V_{R1} 参考电压。以类似的方式产生 V_{EA} ， V_{PEB} ， V_{EB} 和 V_{PEF} 参考电压，并且根据上述的优化定位到最靠近的 V_{RI} 。此外，如前面小擦除一节中所述，强制 PRG-ERS 总线 22 设置 BLOCK-DEC 块 19，X-DEC 块 17 和 Y-DEC 块 18，以允许在选定的存储块或区段上产生一擦除操作。

尽管以上是本发明的优选实施例的完整的说明，但可以使用各种替代，修改和等同物。应当理解，通过对上述实施例适当的修改本发

5 明可以同样地应用。许多存储技术可以应用于本发明，并且不仅限于电荷的存储。其它本质的基本特性可以存储在集成电路存储单元中。例如，覆盖半导体沟道的薄膜中的铁电极化调制沟道的导电性，并形成了铁电存储器的基础。通过施加于复盖铁电膜的导体控制栅的电压可以控制极化量，这样就可以获得不同程度的导电性。可以用参考电压设置控制栅电压。因此，本发明可以应用于类似于电荷存储器的铁电存储器。

10 因此，不应当把上述说明作为对本发明范围的限制，本发明的范围是由所附的权利要求分界定义的。

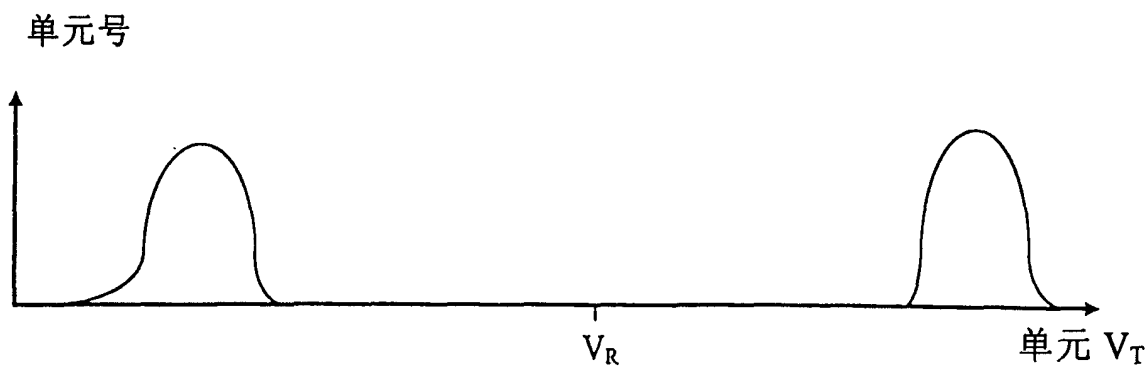


图 1A
现有技术

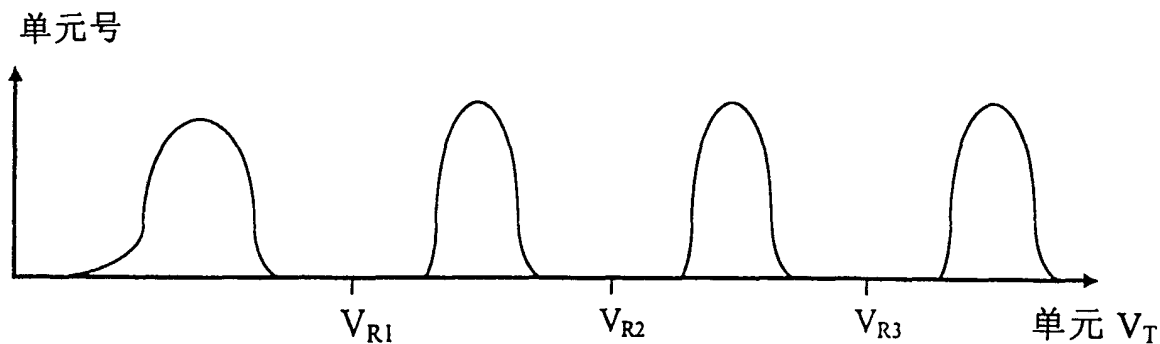


图 1B
现有技术

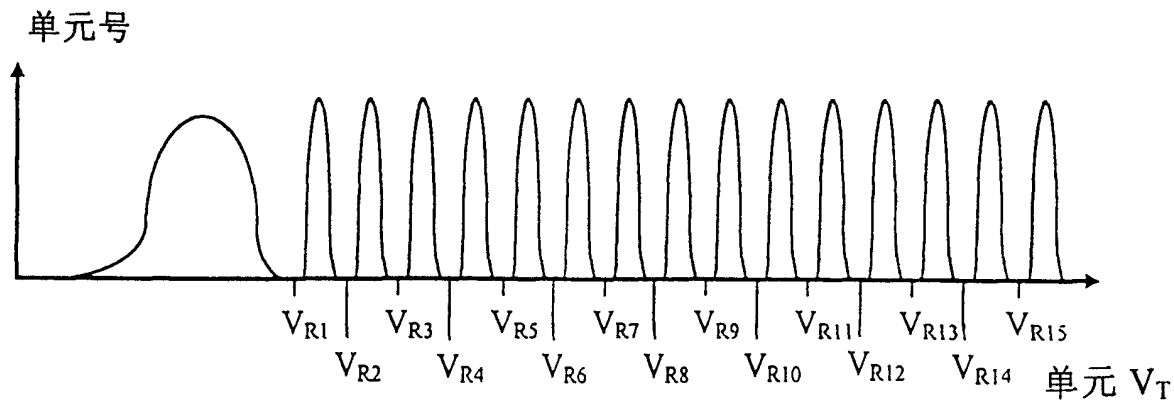


图 1C

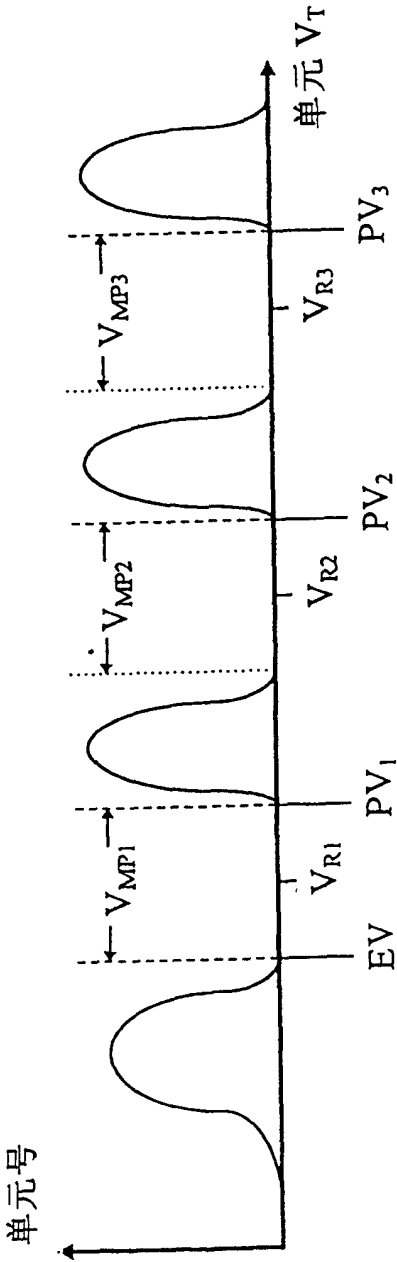


图 2
现有技术

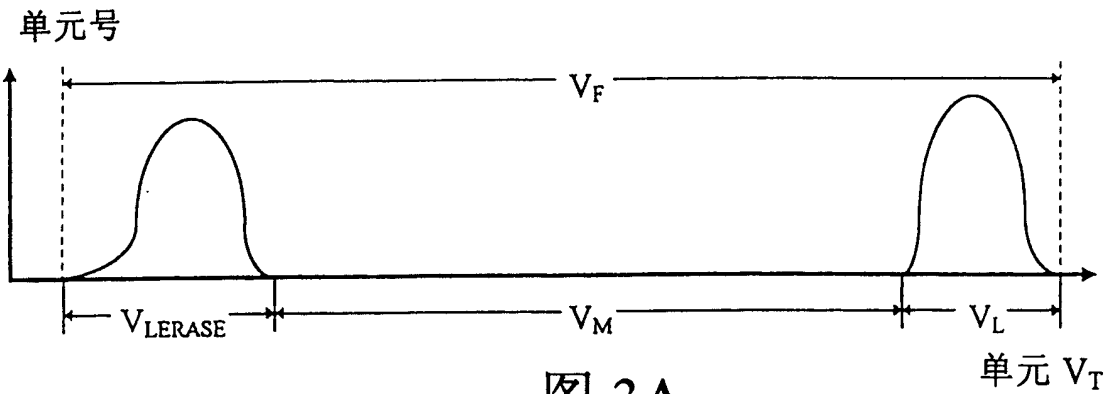


图 3A
现有技术

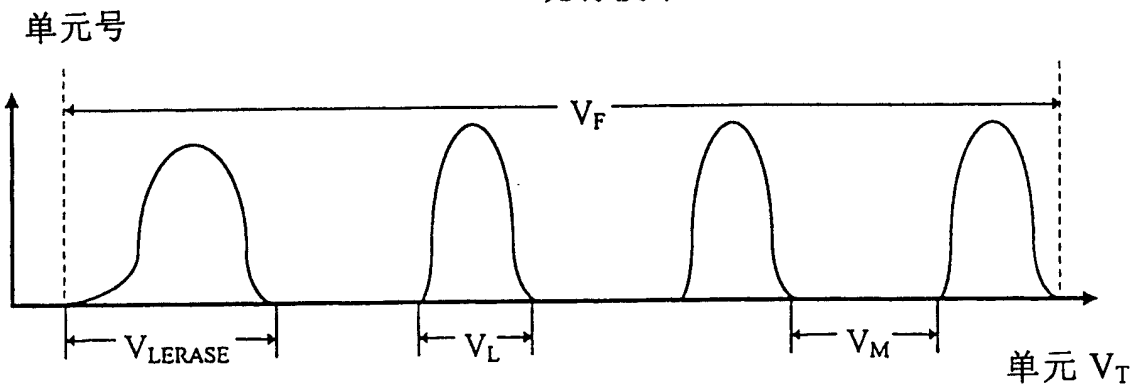


图 3B
现有技术

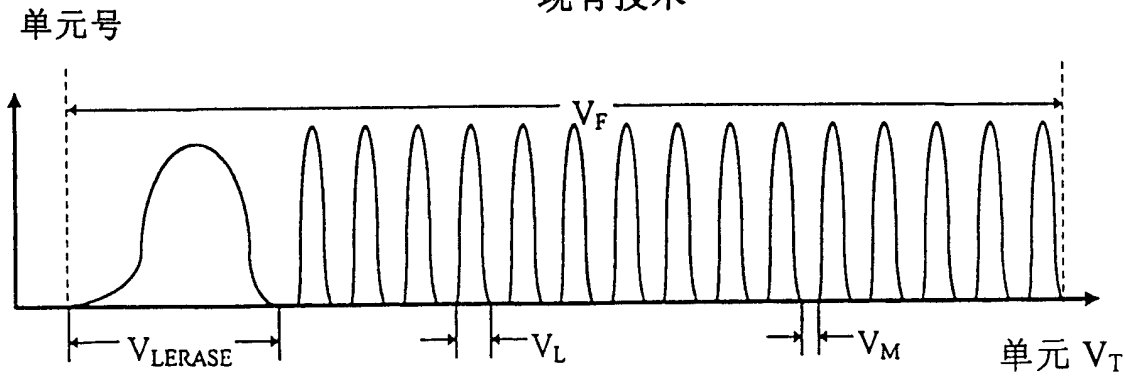


图 3C

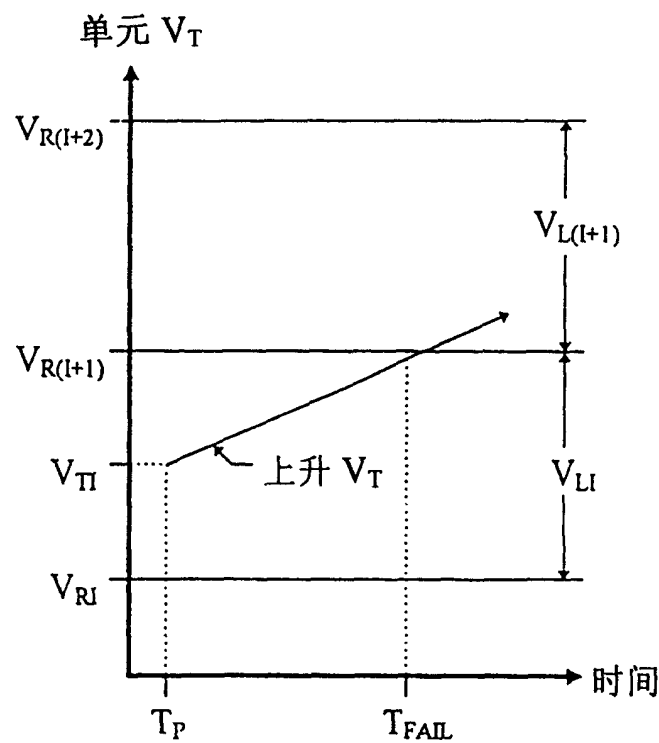


图 4A

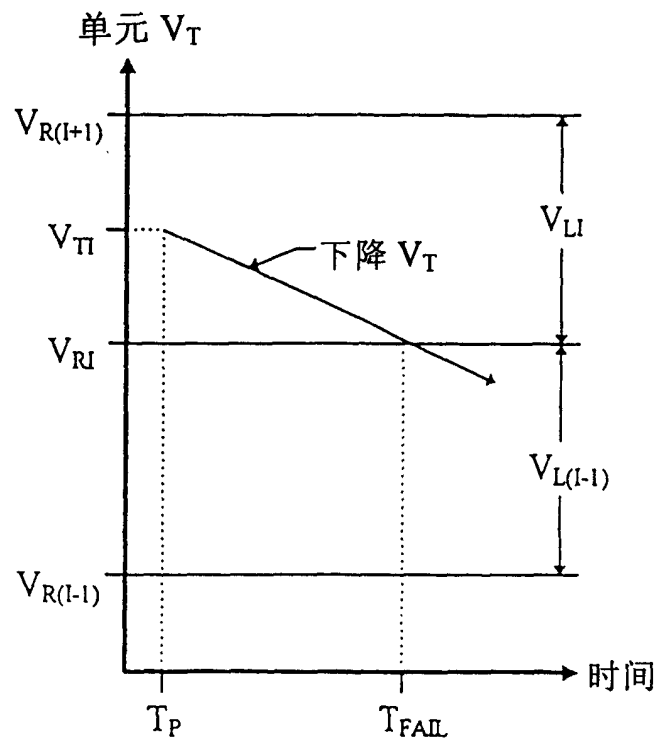


图 4B

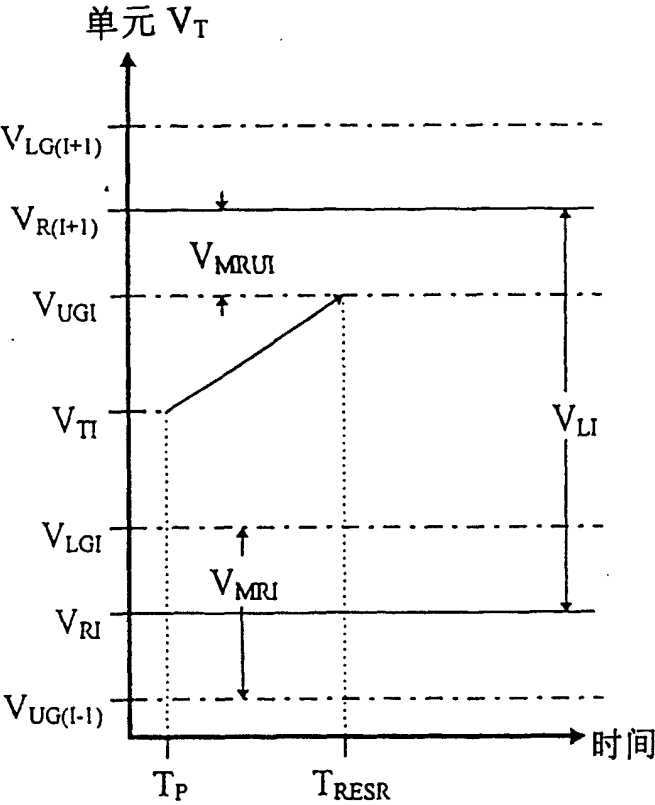


图 5A

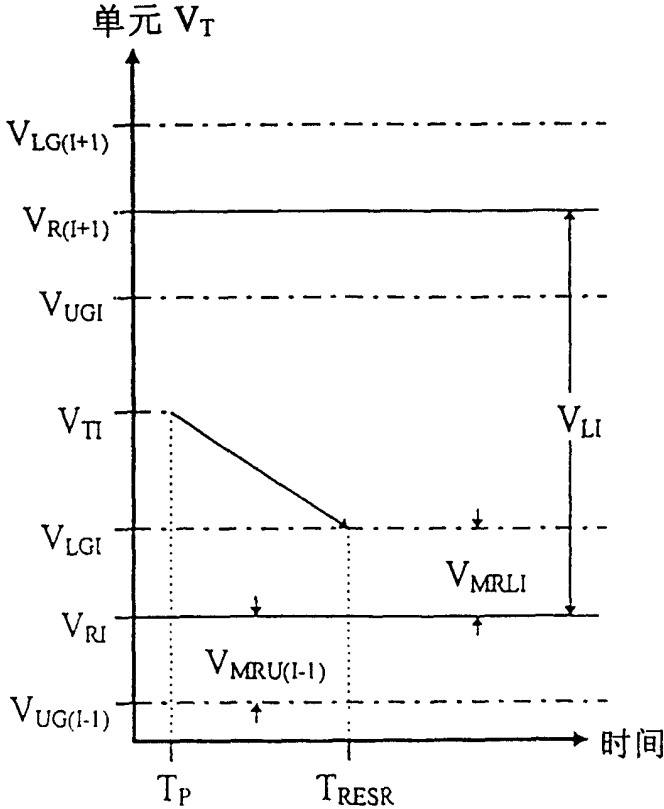


图 5B

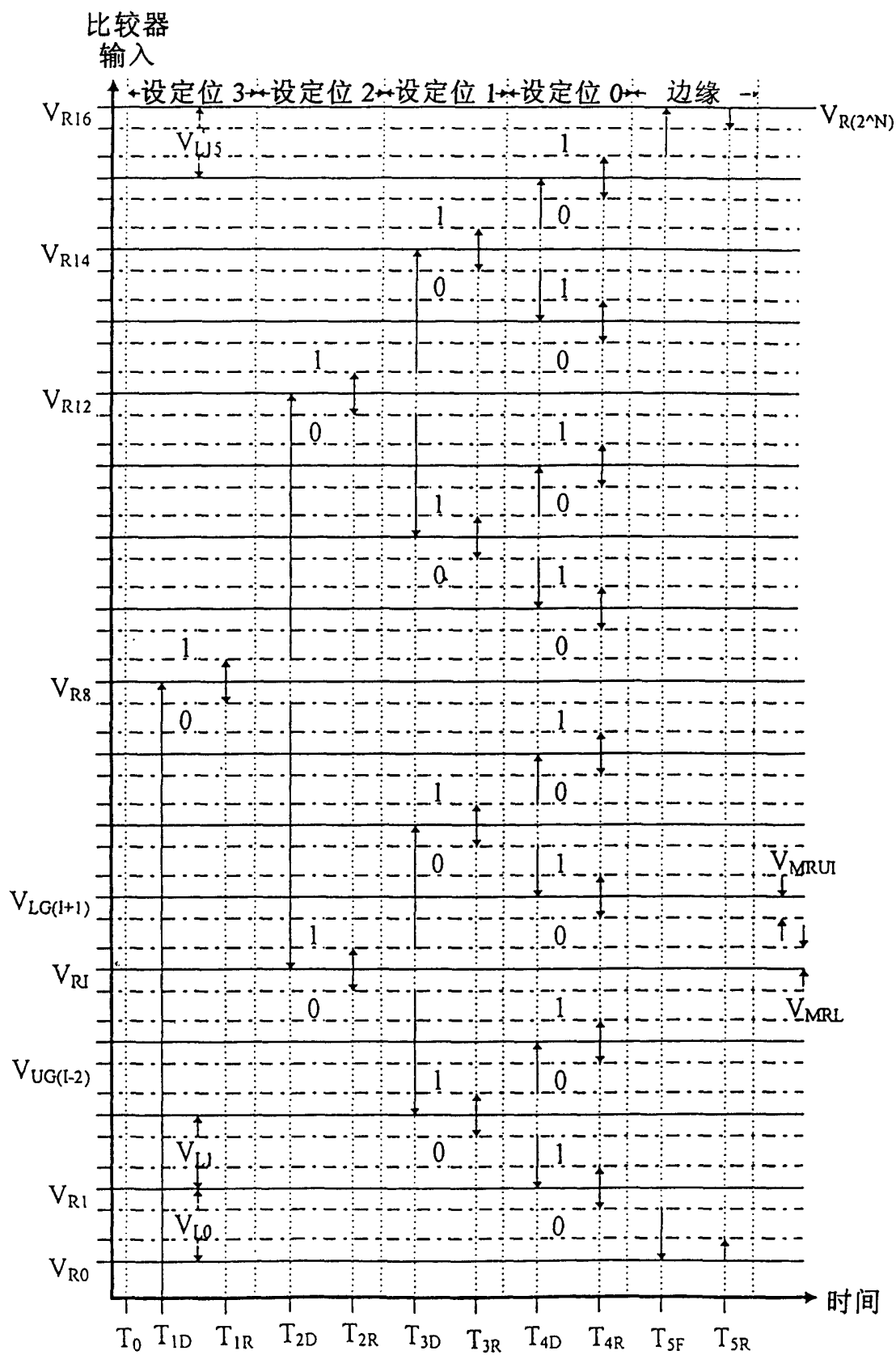


图 6

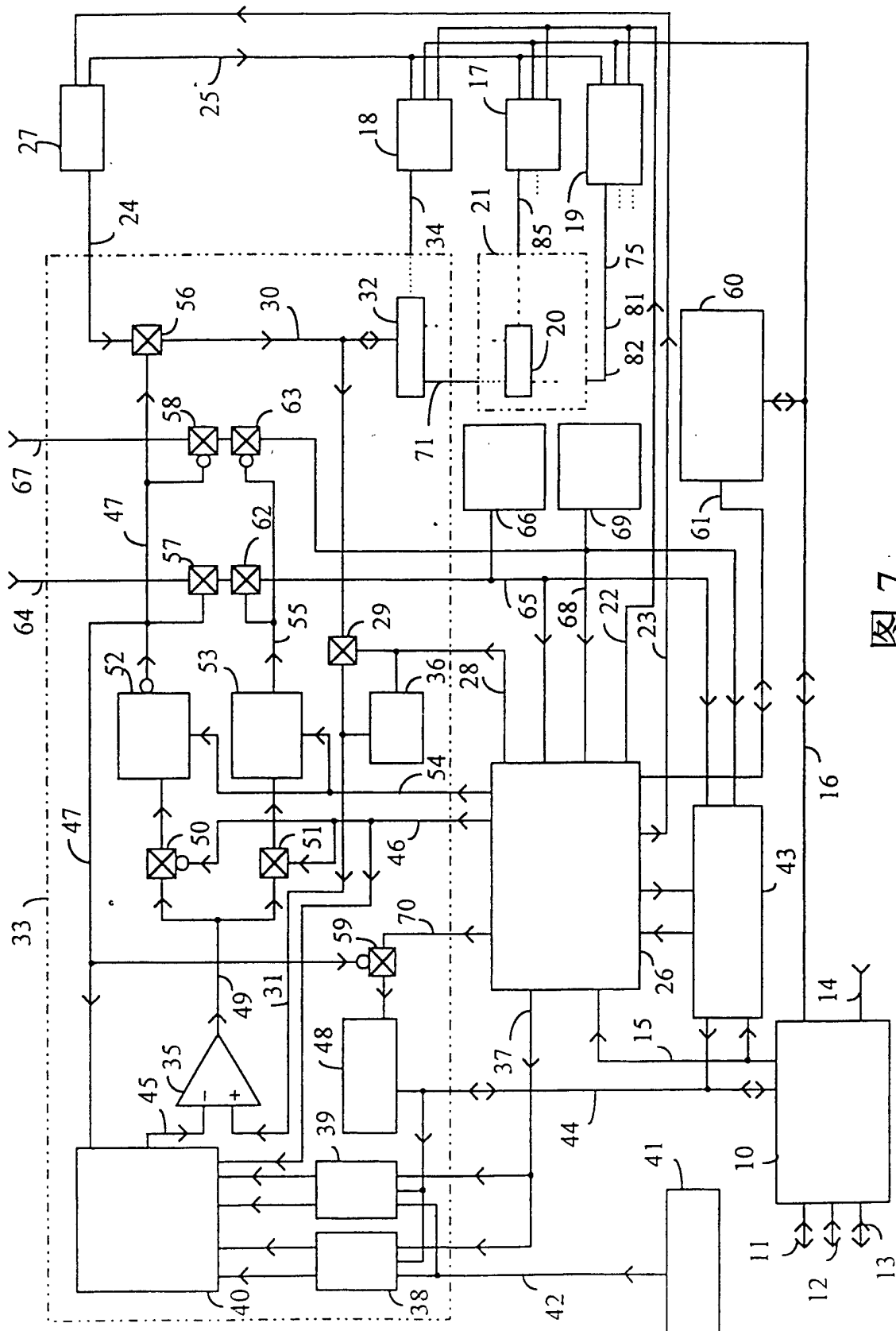


图 7

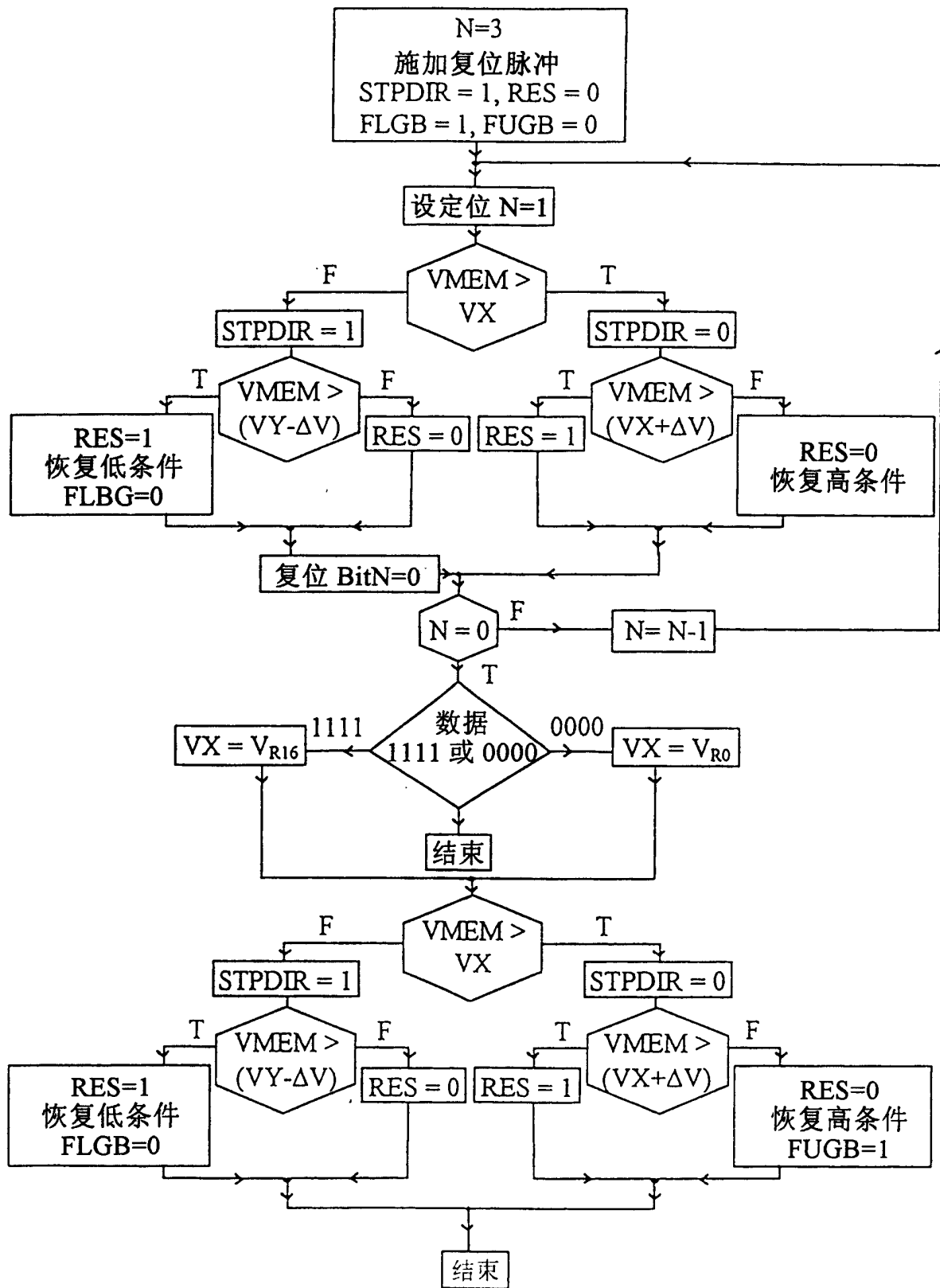


图 8

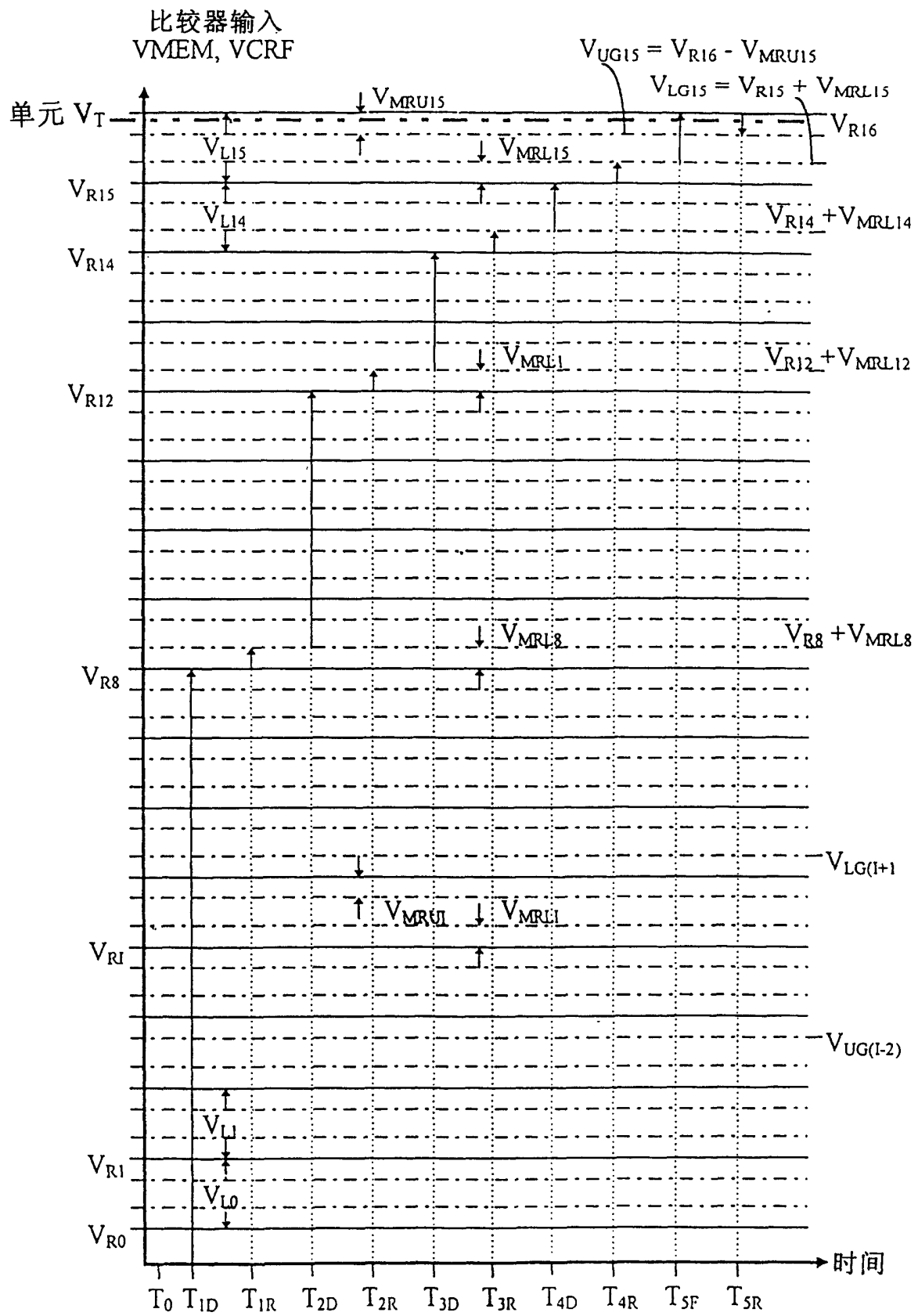


图 9

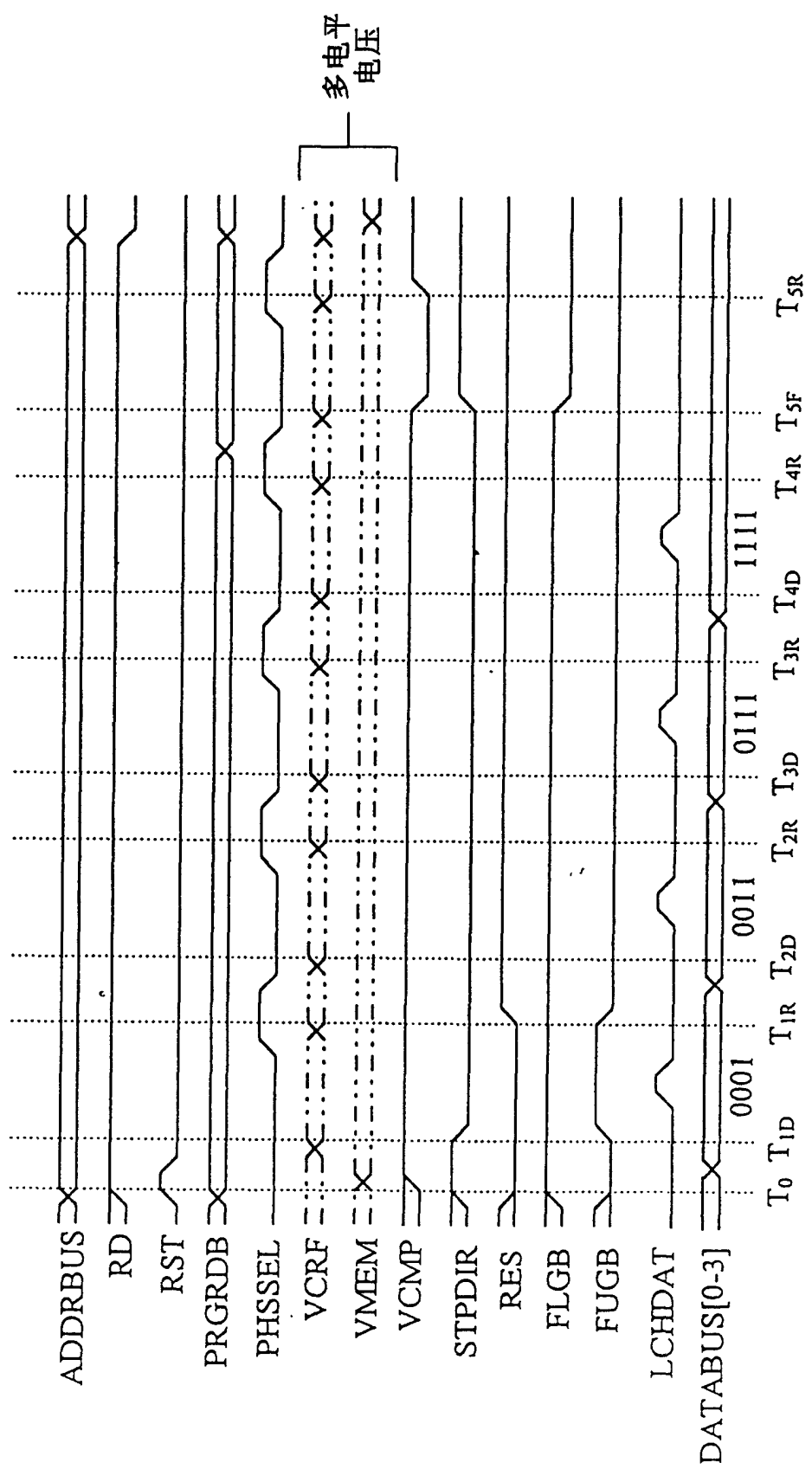
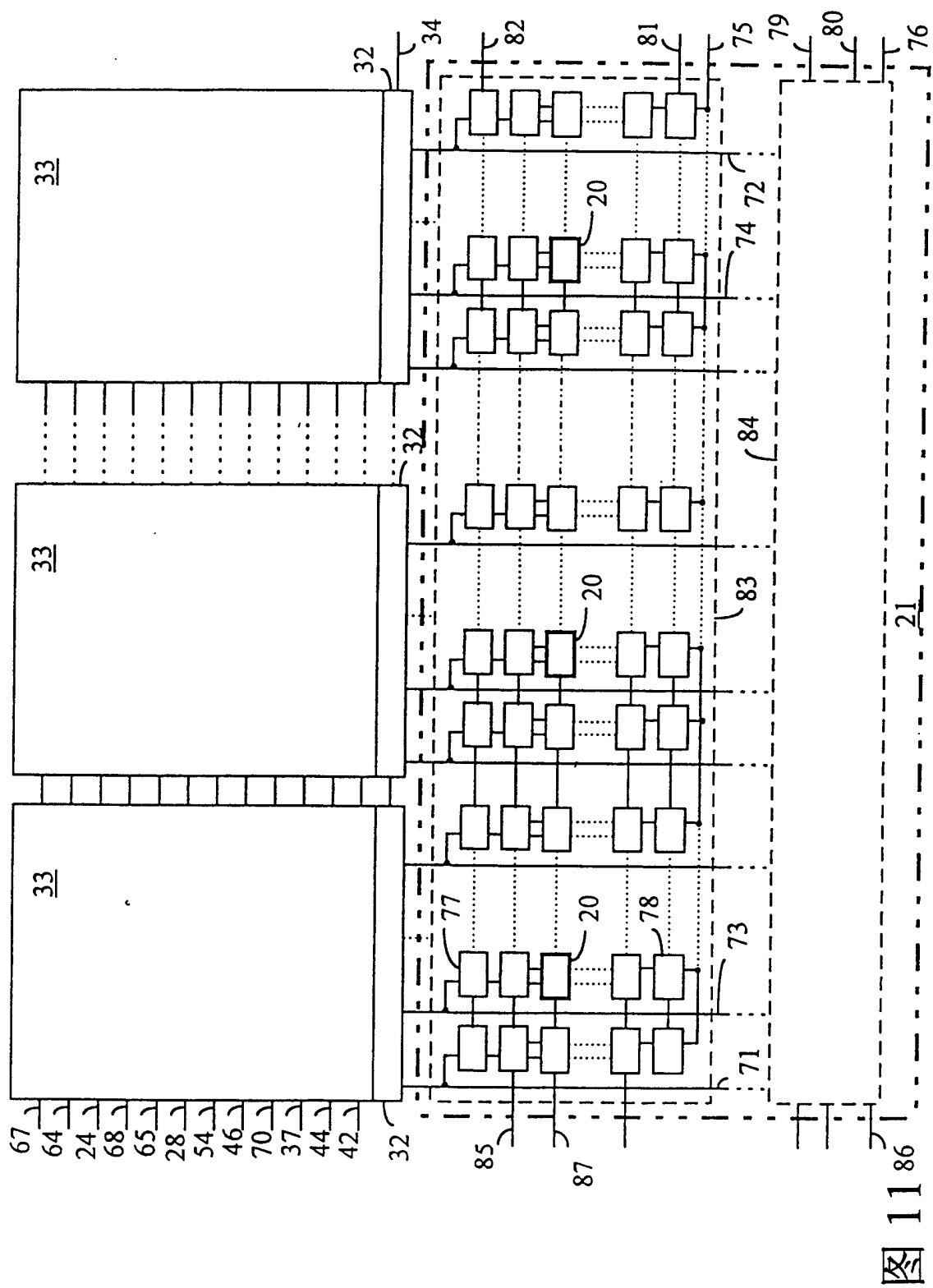


图 10



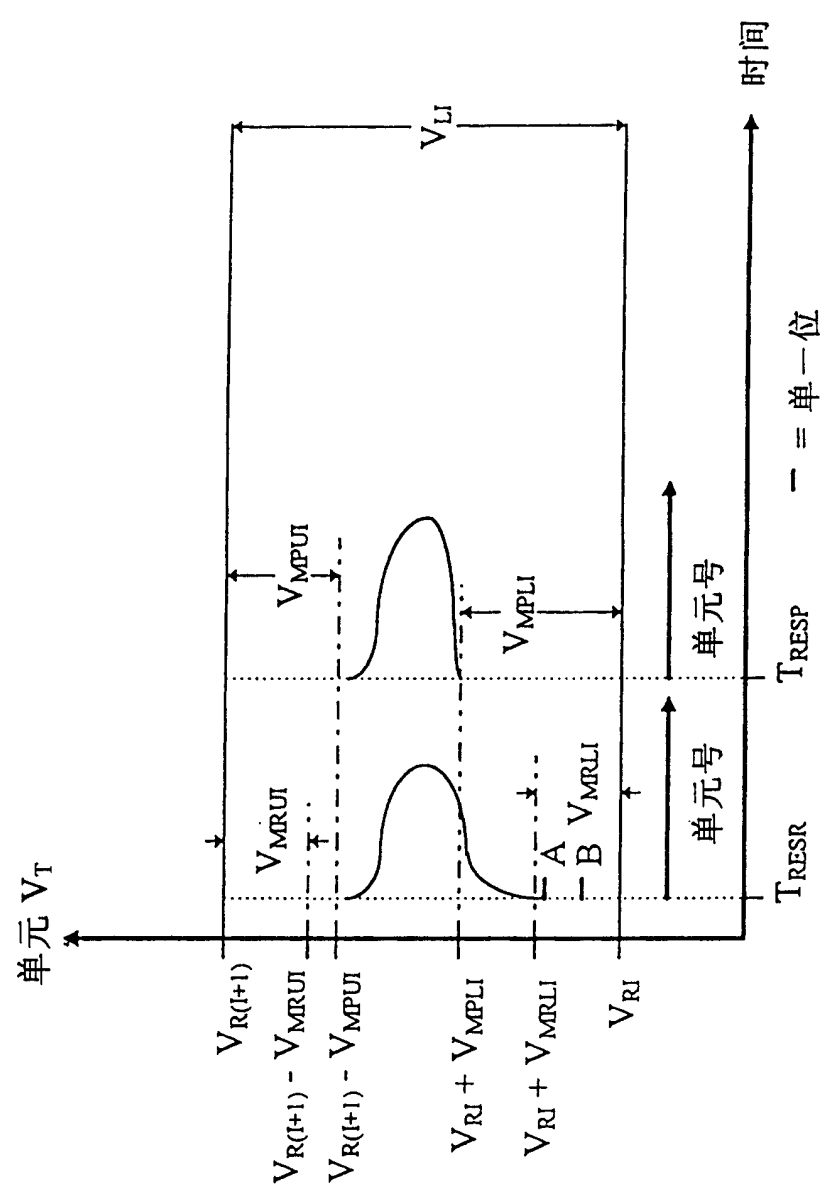


图 12

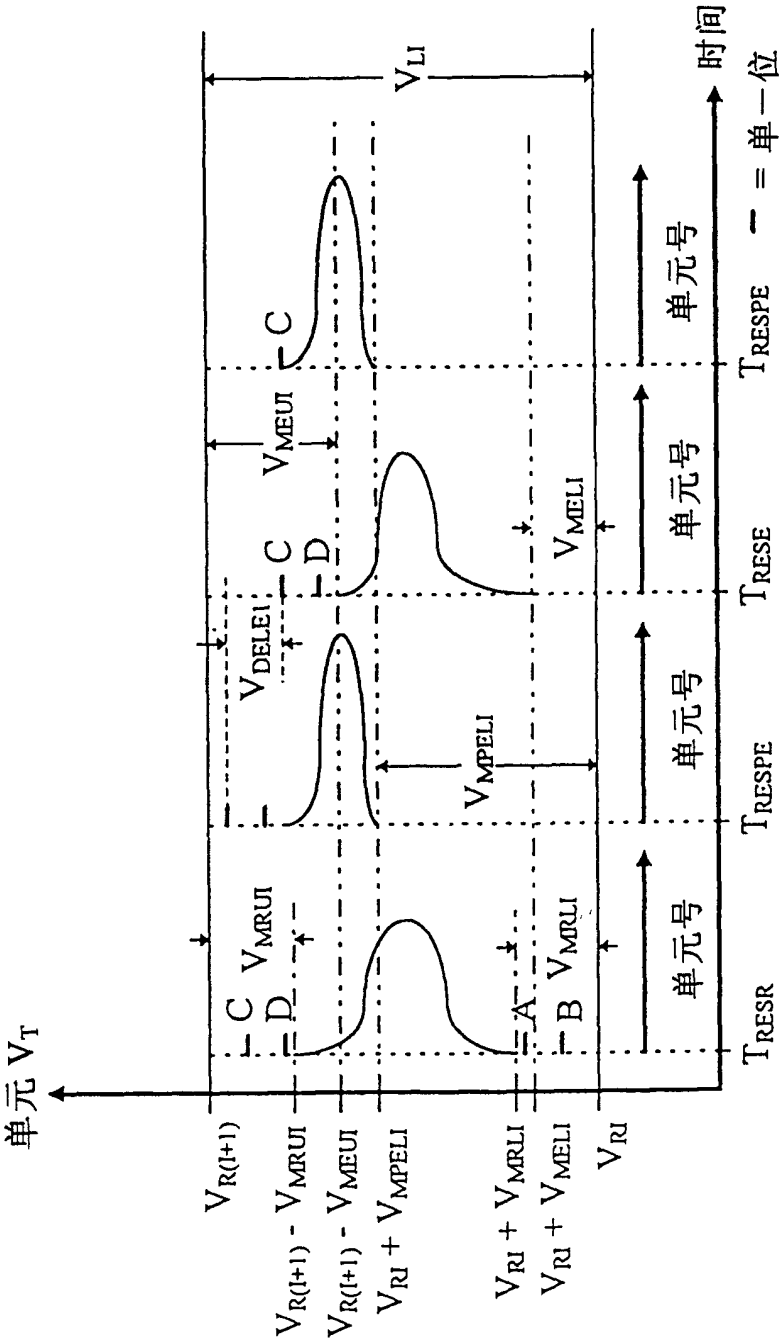


图 13A

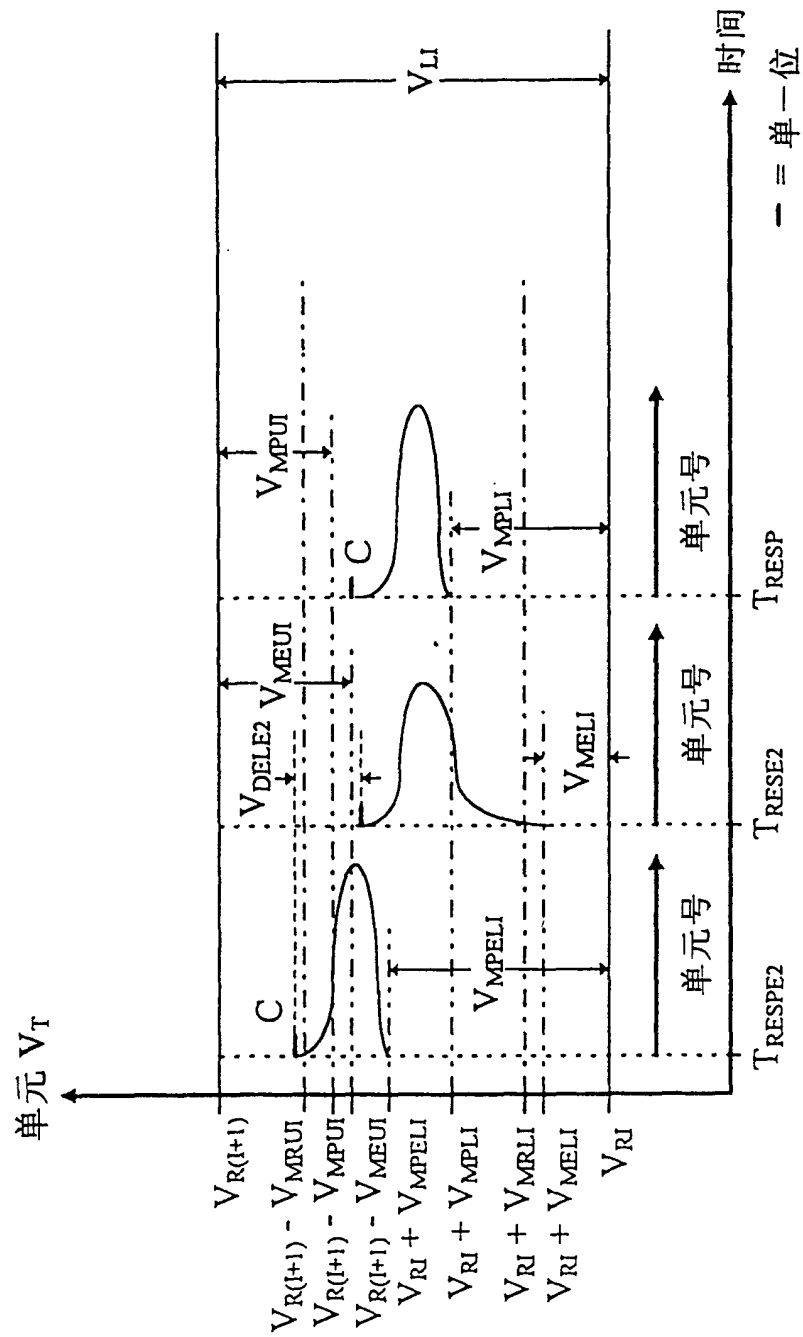


图 13B

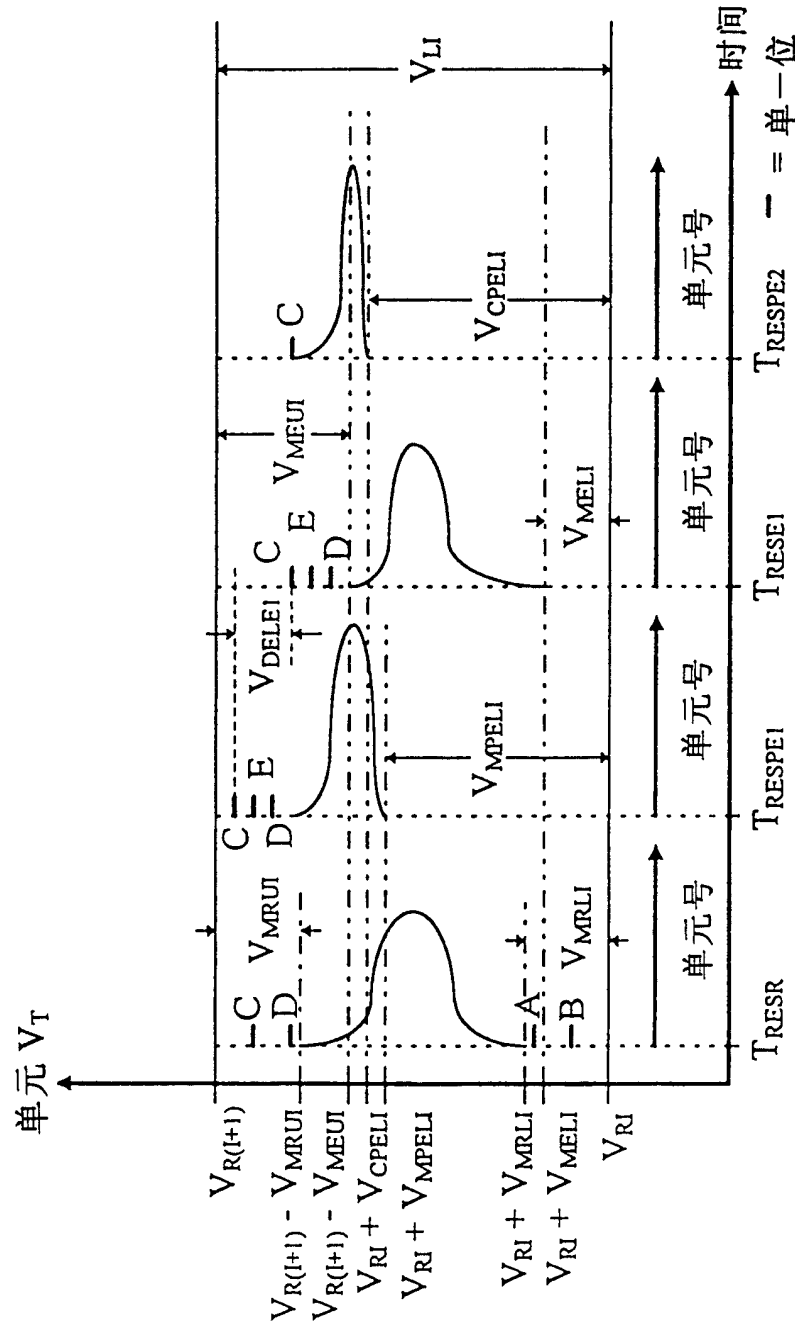


图 14A

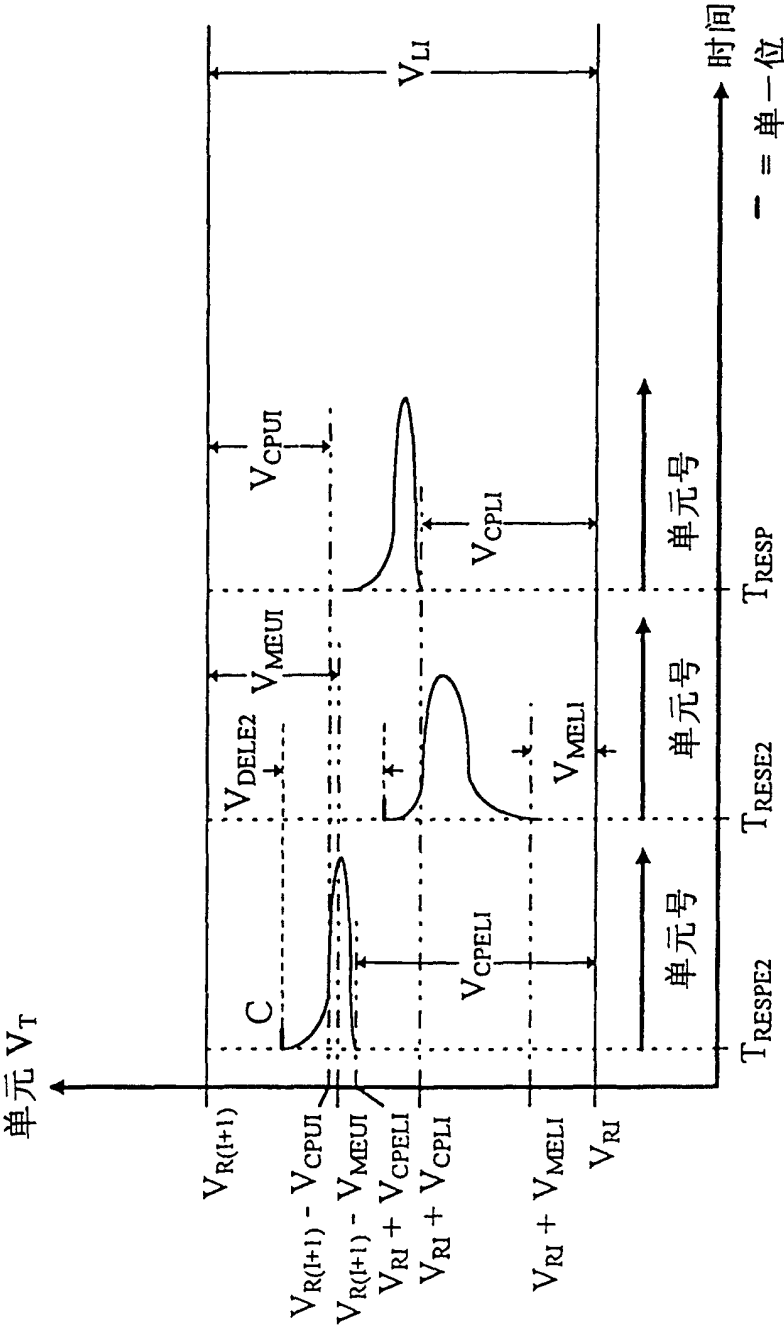


图 14B

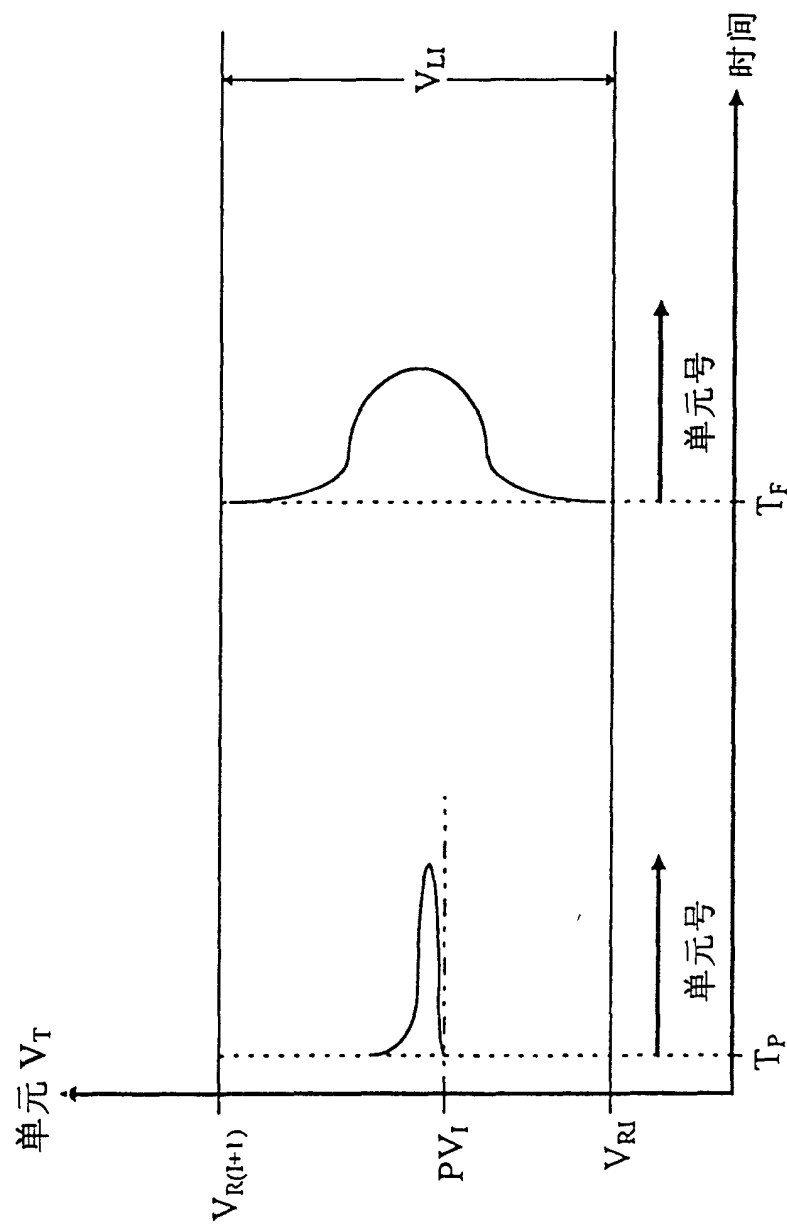


图 15A
现有技术

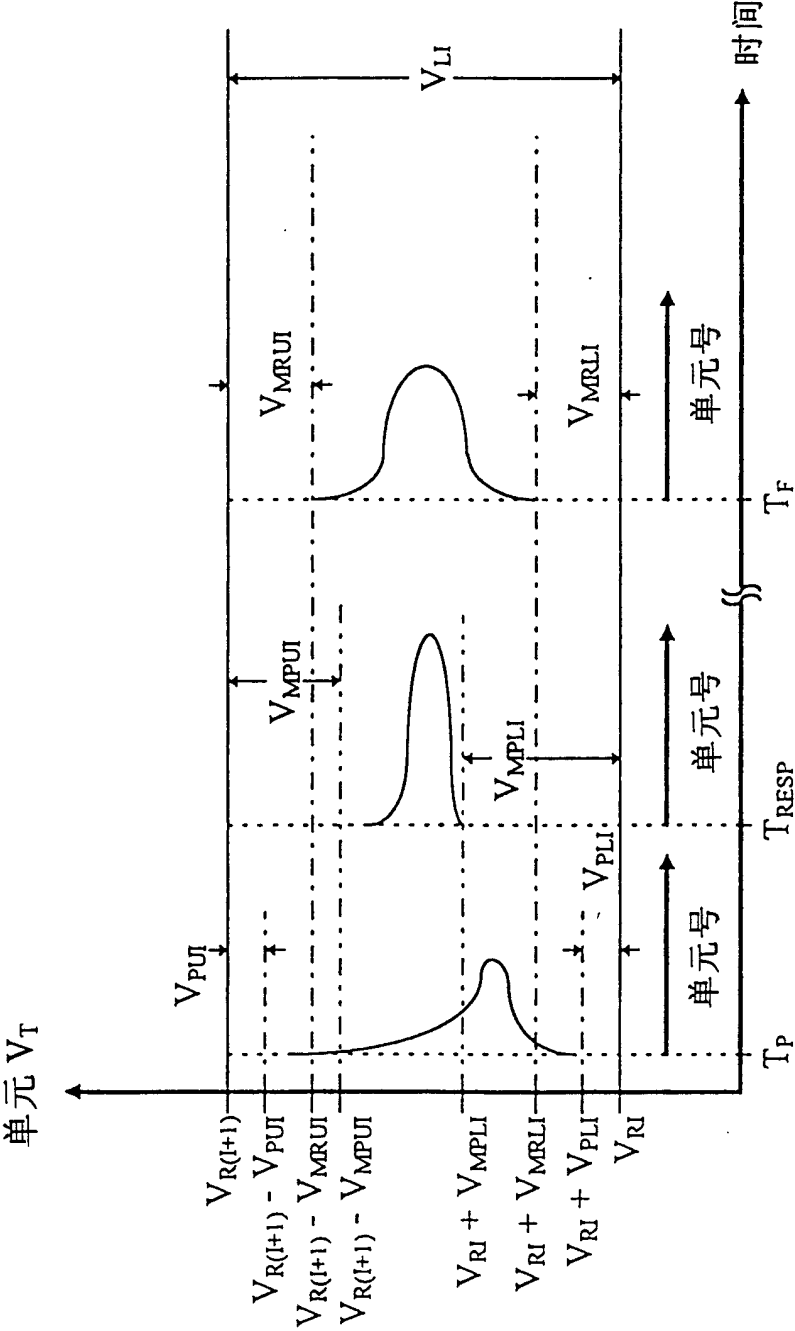


图 15B

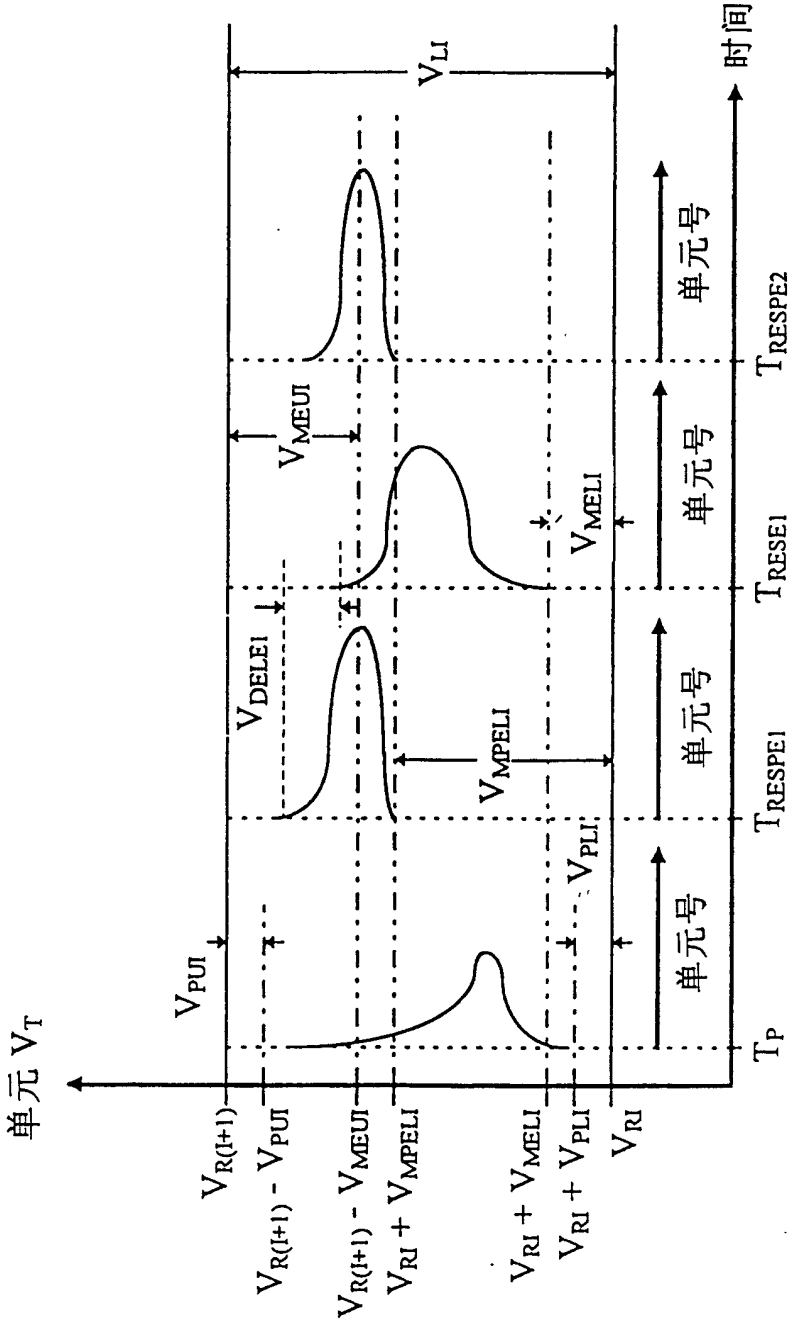


图 16A

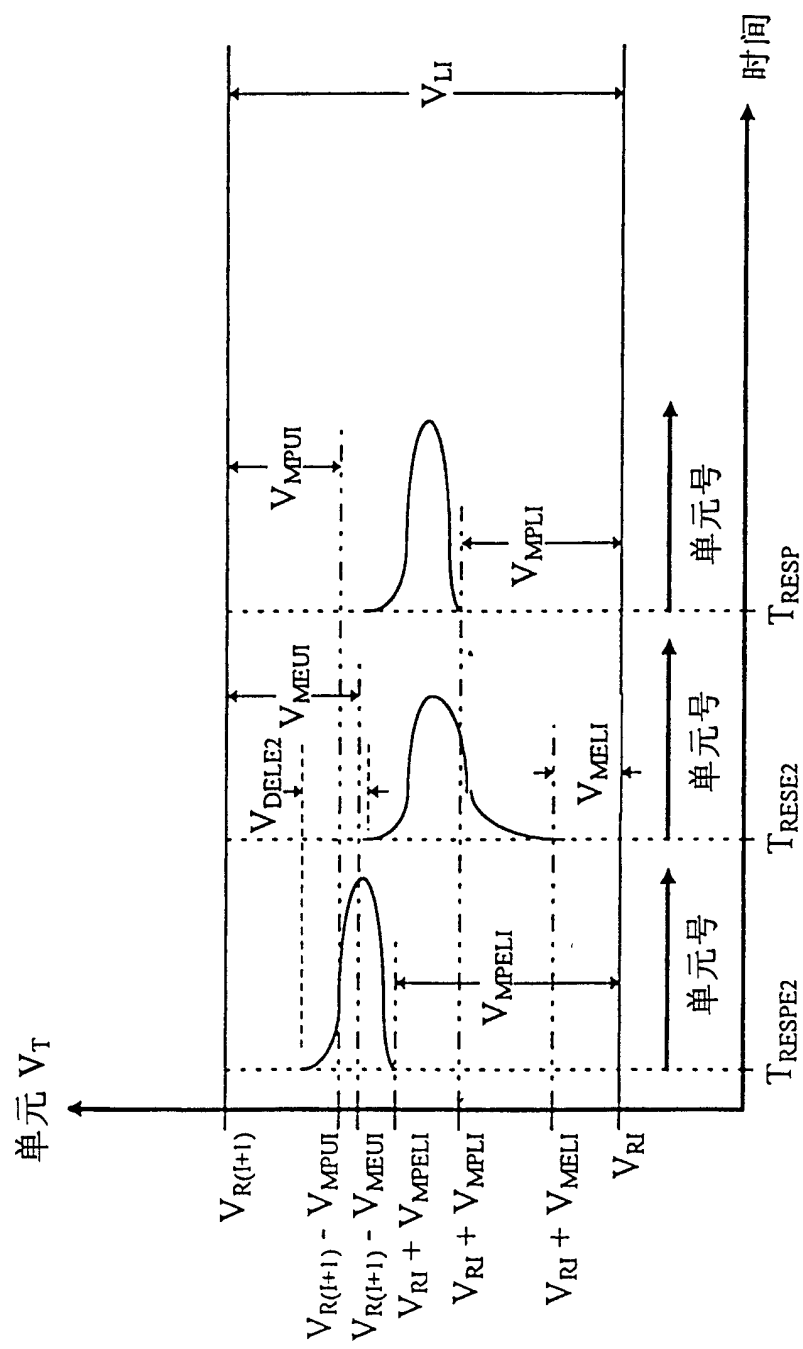


图 16B

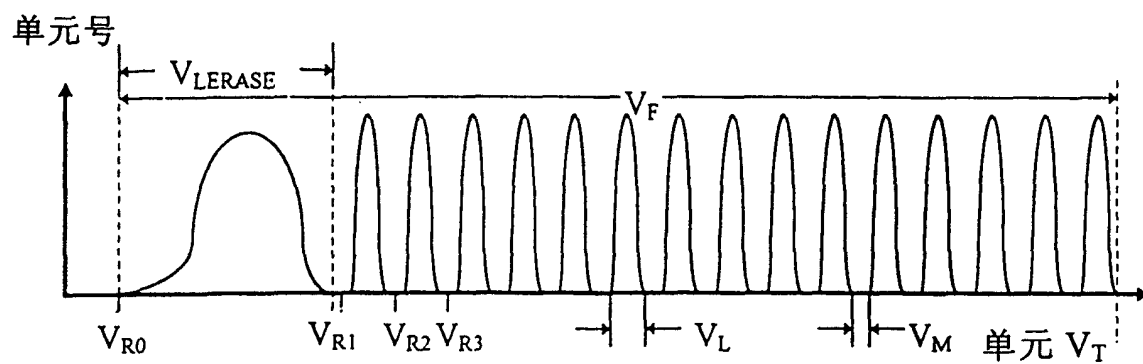


图 17A

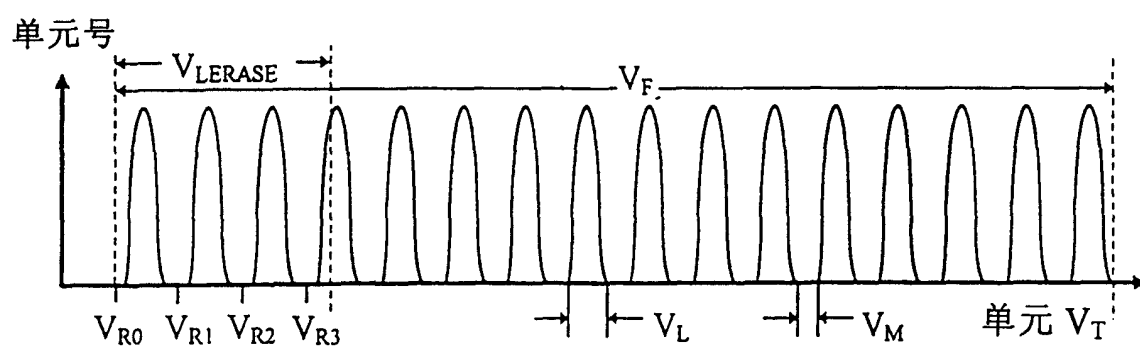


图 17B

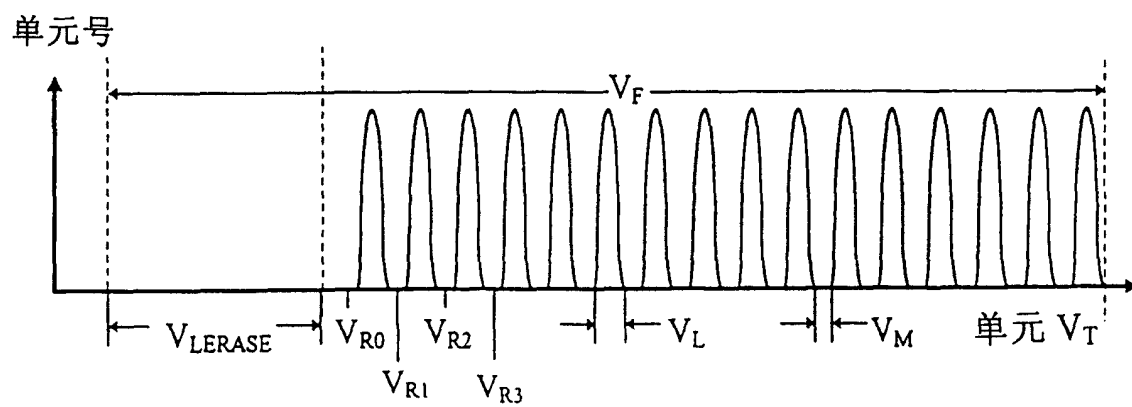


图 17C

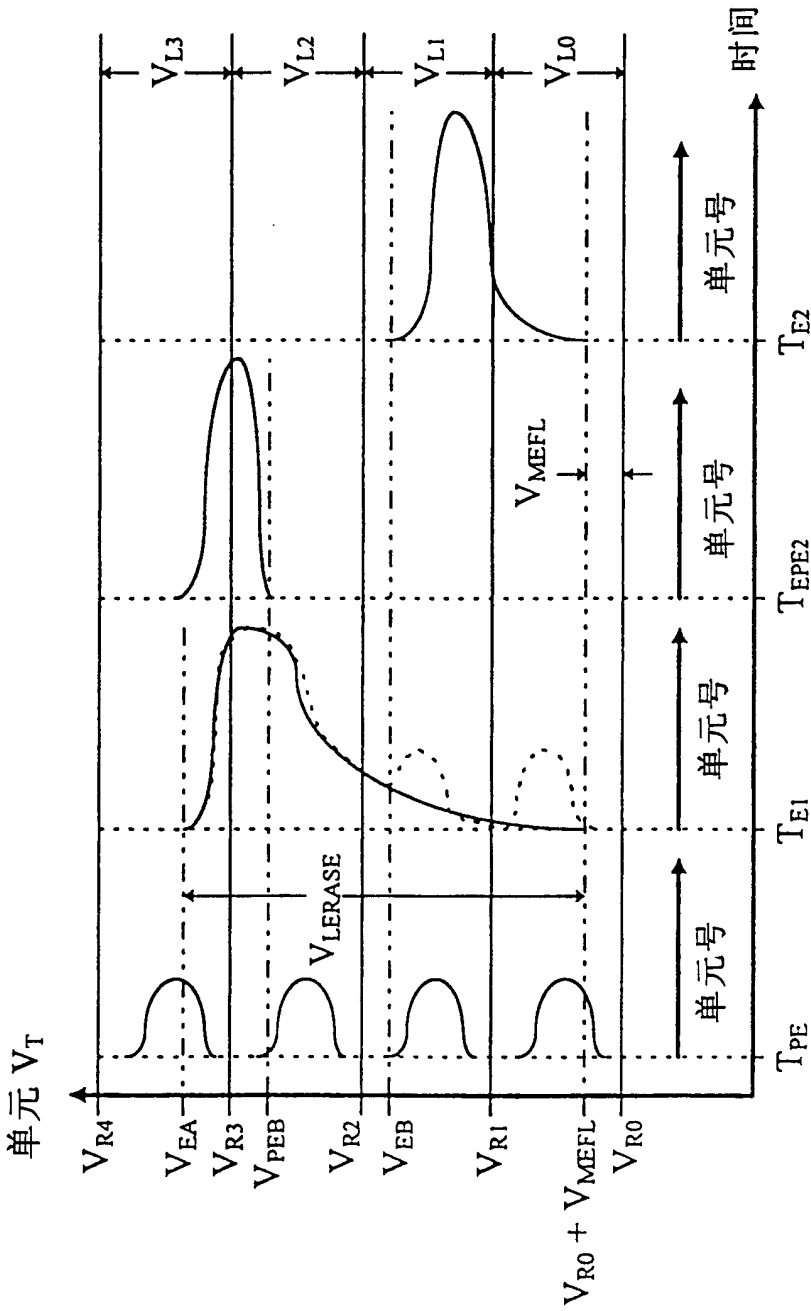


图 18A

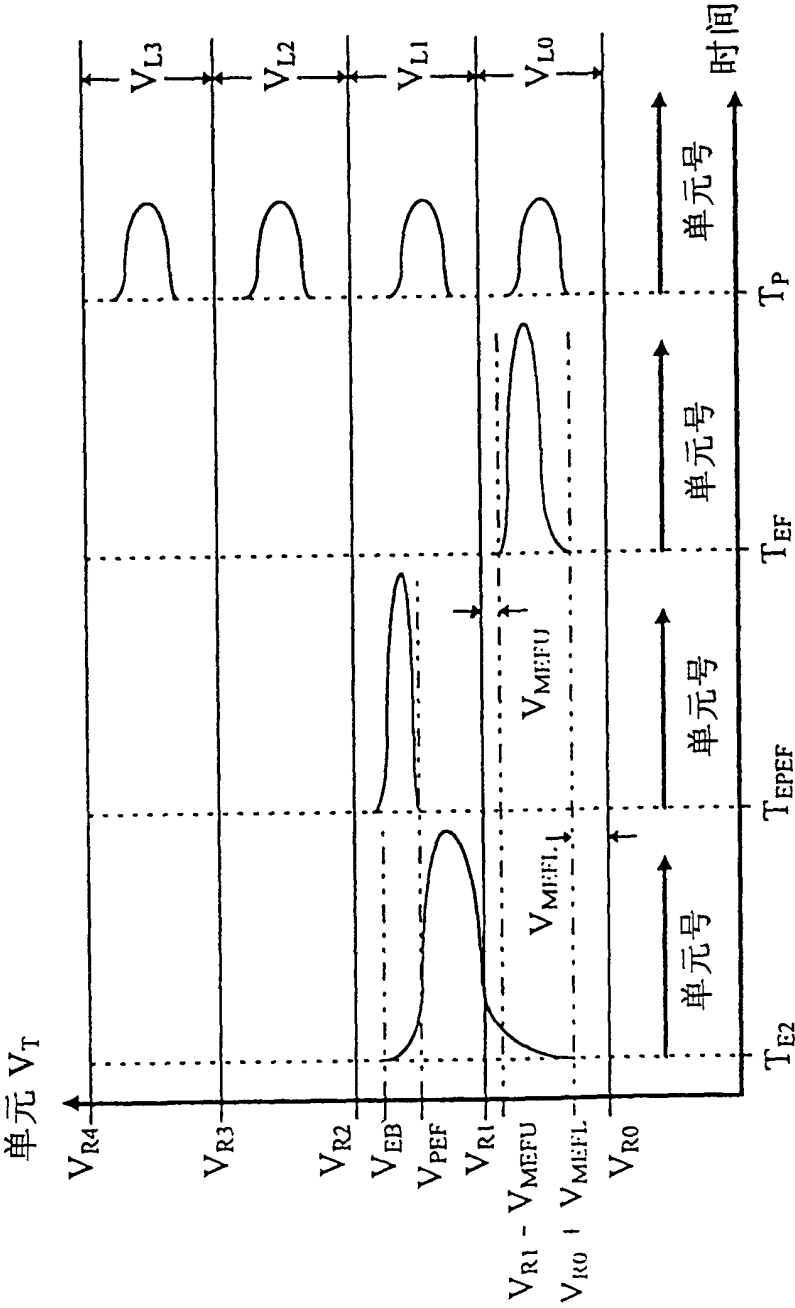


图 18B