



(12)发明专利

(10)授权公告号 CN 104081462 B

(45)授权公告日 2017.06.06

(21)申请号 201280068501.2

(22)申请日 2012.11.13

(65)同一申请的已公布的文献号

申请公布号 CN 104081462 A

(43)申请公布日 2014.10.01

(30)优先权数据

13/315213 2011.12.08 US

(85)PCT国际申请进入国家阶段日

2014.07.30

(86)PCT国际申请的申请数据

PCT/US2012/064881 2012.11.13

(87)PCT国际申请的公布数据

W02013/085676 EN 2013.06.13

(73)专利权人 硅存储技术公司

地址 美国加利福尼亚州

(72)发明人 H.V.特兰 H.Q.阮 A.利 T.吴

(74)专利代理机构 中国专利代理(香港)有限公司 72001

代理人 俞华梁 姜甜

(51)Int.Cl.

G11C 11/00(2006.01)

(56)对比文件

US 5280447 A, 1994.01.18,

US 6438032 B1, 2002.08.20,

US 6292868 B1, 2001.09.18,

US 6292868 B1, 2001.09.18,

US 2005180213 A1, 2005.08.18,

US 2009108328 A1, 2009.04.30,

CN 1377042 A, 2002.10.30,

US 2011157958 A1, 2011.06.30,

CN 1577605 A, 2005.02.09,

US 2006259847 A1, 2006.11.16,

审查员 张丽娜

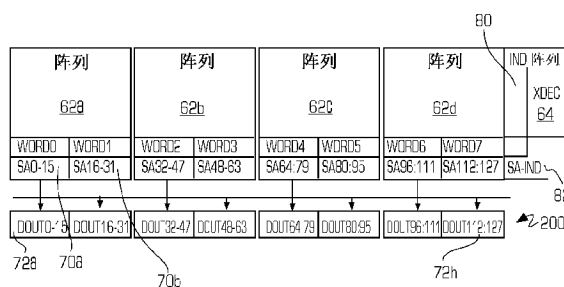
权利要求书3页 说明书8页 附图8页

(54)发明名称

非易失性存储器装置及对此装置编程的方法

(57)摘要

非易失性存储装置具有用于提供编程电流的电荷泵和非易失性存储器单元的阵列。阵列的每一存储器单元由来自电荷泵的编程电流进行编程。非易失性存储器单元的阵列被划分成多个单位,每一单位包括多个存储器单元。指示器存储器单元与每一单位的非易失性存储器单元关联。当每一单位的50%或更少的存储器单元将被编程时,编程电路使用编程电流对每一单位的存储器单元进行编程,并且当每一单位的多于50%的存储器单元将被编程时,编程电路使用编程电流对每一单位的存储器单元的逆以及与每一单位关联的指示器存储器单元进行编程。



1. 一种非易失性存储器装置,包括:

电荷泵,用于提供编程电流;

非易失性存储器单元的阵列,其中每一存储器单元以所述编程电流编程;所述非易失性存储器单元的阵列被划分为多个单位,其中每一单位包括多个存储器单元;

指示器存储器单元,其与每一单位的非易失性存储器单元关联;

计数器电路,用于对被编程的每个单位的存储器单元的数量进行计数,所述计数器电路包含包括时钟脉冲发生器的数字'0'比特检测器,所述'0'比特检测器被配置为顺序检验关于'0'比特的被编程比特;以及

编程电路,用于当每一单位的某一比例或更少的所述存储器单元将被编程时,使用所述编程电流对每一单位的所述存储器单元进行编程,并且用于当多于每一单位的所述某一比例的存储器单元将被编程时,使用所述编程电流对每一单位的所述存储器单元的逆以及与每一单位关联的所述指示器存储器单元进行编程。

2. 根据权利要求1所述的存储器装置,还包括:

多个读出放大器,一个读出放大器连接至某一单位的各个非易失性存储器单元用于感测所述非易失性存储器单元的状态;

指示器读出放大器,其连接至所述指示器存储器单元用于感测与被感测的所述单位关联的所述指示器存储器单元的状态;

逻辑电路,用于在所述指示器读出放大器感测到所述指示器存储器单元的状态处于一个状态的情况下输出所述多个读出放大器的状态;以及用于在所述指示器读出放大器感测到所述指示器存储器单元的状态处于另一状态的情况下输出所述多个读出放大器的反状态。

3. 根据权利要求1所述的存储器装置,其中所述非易失性存储器单元的阵列成多行多列布置,其中指示器存储器单元的阵列在行方向上与所述非易失性存储器单元的阵列相邻设置,其中一个指示器存储器单元与相同行上的多个存储器单元关联。

4. 根据权利要求1所述的存储器装置,其中所述非易失性存储器单元的阵列成多行多列布置,其中指示器存储器单元的阵列在列方向上与所述非易失性存储器单元的阵列相邻设置,其中一个指示器存储器单元与相同列上的多个存储器单元关联。

5. 根据权利要求1所述的存储器装置,其中所述存储器单元的每个包括:

第一导电类型的半导体衬底,其具有平坦表面;

在所述平坦表面上的第二导电类型的第一区域;

在所述平坦表面上的所述第二导电类型的第二区域,其与所述第一区域间隔开,其中沟道区域位于所述第一区域与所述第二区域之间;

浮栅,与所述沟道区域的第一部分间隔开;

字线,在所述浮栅一侧与其相邻,与所述浮栅绝缘,并且与所述沟道区域的第二部分间隔开;

擦除栅,在所述浮栅另一侧与其相邻,与所述浮栅绝缘,并且与所述第二区域间隔开;以及

耦合栅,位于所述浮栅之上,与所述浮栅间隔开,并且位于字线和所述擦除栅之间并且与其绝缘。

6. 根据权利要求1所述的存储器装置,其中所述某一比例为百分之五十。
7. 根据权利要求1所述的存储器装置,还包括数字式‘0’位检测器。
8. 根据权利要求7所述的存储器装置,其中所述数字式‘0’位检测器包括时钟脉冲发生器并且基于计时的顺序数字式‘0’位检测。
9. 根据权利要求1所述的存储器装置,还包括模拟‘0’位检测器。
10. 根据权利要求9所述的存储器装置,其中所述模拟‘0’位检测器基于电流偏置的单位进行检测。
11. 根据权利要求10所述的存储器装置,其中所述电流偏置为微安级。
12. 根据权利要求1所述的存储器装置,其中所述存储器阵列还包括多个存储器子阵列。
13. 根据权利要求12所述的存储器装置,其中没有用于编程的存储器单元的每一子阵列使编程偏置被禁能。
14. 一种对非易失性存储器装置编程的方法,所述存储器装置具有多个存储器单位,其中每一存储器单位具有多个存储器单元,每一存储器单位带有关联的指示器存储器单元,某一单位的所述多个存储器单元和所述关联的指示器存储器单元由来自电荷泵的编程电流进行编程,所述方法包括:

利用计数器电路,通过顺序检验关于‘0’比特的被编程比特来确定是否单位中某一比例或更少的所述存储器单元将由所述编程电流进行编程,所述计数器电路包含包括时钟脉冲发生器的数字‘0’比特检测器;

如果是,由所述编程电流对所述存储器单元编程;以及

如果否,由所述编程电流对所述存储器单元的逆和所述关联的指示器存储器单元进行编程。
15. 根据权利要求14所述的方法,还包括:

感测来自相同单位的所述存储器单元的每个的状态;

感测与所述相同单位关联的所述指示器存储器单元的状态;

在所述关联的指示器存储器单元处于一个状态的情况下,输出来自所述相同单位的所述存储器单元的每个的状态;以及

在所述关联的存储器单元处于另一状态的情况下,输出来自所述相同单位的所述存储器单元的每个的反状态。
16. 根据权利要求14所述的方法,其中所述某一比例为百分之五十。
17. 根据权利要求14所述的方法,其中指示器存储器单元包括多个基本存储器单元。
18. 根据权利要求14所述的方法,其中指示器存储器与行解码器相邻。
19. 一种对非易失性存储器装置编程的方法,所述存储器装置具有多个存储器单位,其中每一存储器单位具有多个存储器单元,某一单位的所述多个存储器单元由来自高压电荷泵的编程电流进行编程,所述方法包括:

利用计数器电路,从单位中第一个存储器单元到最后一个存储器单元按顺序地确定通过高压电荷泵将存储状态从擦除改为编程状态的、所述单位中存储器单元的数量是否达到预定数量,所述计数器电路包含包括时钟脉冲发生器的数字‘0’比特检测器;

如果是,由所述高压电荷泵对所述预定数量的存储器单元进行编程;以及

如果否,那么继续确定将要被编程的所述单位中存储器单元的剩余数量直到所述单位中全部存储器单元被确定为将要被编程或不被编程。

20. 根据权利要求19所述的方法,其中所述确定基于位基础。

21. 根据权利要求19所述的方法,其中所述确定基于字基础。

22. 根据权利要求19所述的方法,其中每一存储器单位还包括关联的指示器存储器单元。

23. 根据权利要求19所述的方法,其中所确定的用于编程的存储器单元与数据输入位‘0’对应。

非易失性存储器装置及对此装置编程的方法

技术领域

[0001] 本发明涉及非易失性存储器装置及对这样一个非易失性存储器装置编程的方法。

背景技术

[0002] 具有用于在其存储电荷的浮栅或捕捉电荷层的非易失性存储器单元在现有技术中是众所周知的。参照图1,示出现有技术的非易失性存储器单元10(分裂栅源侧注入热电子编程单元)的横截面视图。存储器单元10包括第一导电类型(诸如P型)的单晶衬底12。在衬底12的表面或附近为第二导电类型(诸如N型)的第一区域14。同样为第二导电类型的第二区域16与第一区域14间隔开。在第一区域14和第二区域16之间为沟道区18。由多晶硅制成的字线20定位于沟道区18的第一部分之上。字线20通过诸如(二)氧化硅的绝缘层22与沟道区18间隔开。与字线20直接相邻并且间隔开的是浮栅24,浮栅24也由多晶硅制成,并且位于沟道区18的另一部分之上。浮栅24通过另一绝缘层30与沟道区18分离,另一绝缘层30典型地也由(二)氧化硅制成。也由多晶硅制成的耦合栅26位于浮栅24之上并且通过另一绝缘层32与浮栅24绝缘。也由多晶硅制成的擦除栅28在浮栅24的另一侧之上并且与浮栅24间隔开。擦除栅28位于第二区域16之上并且与第二区域16绝缘。擦除栅28与耦合栅26邻近并间隔开。擦除栅28能够具有在浮栅24上面的细小突出物。在存储器单元10的操作中,在浮栅24上存储的电荷控制第一区域14和第二区域16之间的电流的流动。在浮栅24其上带负电荷的场合,存储器单元被编程。在浮栅24其上带正电荷的场合,存储器单元被擦除。存储器单元10在美国专利7,868,375中完全公开,通过引用将其公开全部结合于此。

[0003] 存储器单元10按如下方式操作。在擦除操作期间,当电子从浮栅24移除时,高正电压(例如8-11V)被施加到擦除栅28。负电压(例如-6至-8V)或地电压能够被施加到耦合栅26和/或字线20。电子通过Fowler-Nordheim隧道效应经由浮栅24和擦除栅28之间的绝缘层从浮栅24转移到擦除栅28。特别地,浮栅24可以形成具有面向擦除栅28的锋利尖端,从而促进电子的所述隧道效应。在擦除操作期间,高正电压从电荷泵52(图2中示出)来提供。典型地,由于擦除操作仅涉及电子从浮栅24中移除,电荷泵52不需要提供大电流(典型地在毫微安范围中)。

[0004] 其后,存储器单元10能够被编程。在编程操作期间,当电子通过热电子注入而注入到浮栅24,其中在浮栅24之下的沟道18的部分反相,采用脉冲形式的第一正电压(例如1V至2V)施加到字线20,导致在字线20之下的沟道区域18的部分变得导电。第二正电压(例如8V至10V),也以脉冲的形式,施加到耦合栅26,以利用耦合栅26和浮栅24之间的高耦合比率来使耦合到浮栅24的电压最大化。第三正电压(例如3V至6V),也以脉冲的形式,施加到擦除栅28,以利用擦除栅28和浮栅24之间的耦合比率来使耦合到浮栅24的电压最大化。高电压差(例如4V至7V),也以脉冲的形式,施加在第一区域14和第二区域16之间,以提供沟道18中热电子的生成。从而,在编程操作期间,必须从电荷泵52供应的电流(典型地以微安级)在第一区域14和第二区域16之间流动。

[0005] 在读操作期间,第一正电压(例如1V至3V)施加到字线20以导通字线20之下的沟道

区域18的部分。第二正电压(例如0V至4V)施加到耦合栅26。第三电压(例如0V至3V)施加到擦除栅28。电压差(例如0.5V至2V)施加到第一区域14和第二区域16。如果浮栅24被编程,即浮栅24存储电子,那么施加到耦合栅26的第二正电压和施加到擦除栅28的第三电压不能够克服存储在浮栅24上的负电子并且在浮栅24之下的沟道区域18的部分保持不导电。从而,没有电流或可忽略的小量电流会在第一区域14和第二区域16间流动。然而,如果浮栅24没有被编程,即浮栅24保持不带电或带正电,那么施加到耦合栅26的第二正电压和施加到擦除栅28的第三电压能够导致在浮栅24之下的沟道区域18的部分变得导电。从而,电流会在第一区域14和第二区域16间流动。

[0006] 众所周知,存储器单元10典型地在半导体晶圆上以阵列来形成,阵列具有存储器单元10的多个行和列。参考图2,其中示出具有存储器单元10的阵列60的、现有技术的存储器装置50的模块级示图。图3为图2中所示的阵列60的更详细的示图。阵列60包括多个子阵列62(a-d),其中诸如存储器单元10的多个存储器单元成多行多列布置。行解码器64(又叫做xdec,也即字线解码器)与横越子阵列62a、62b、62c和62d的存储器单元10的每行关联。读出放大器70a和70b与每一子阵列(例如子阵列62a)中的存储器单元10的列关联。列(位线)解码器(ymux,未示出)用于选择(复用,解码)存储器单元的列到读出放大器。信号从读出放大器70提供给输出寄存器72。在图3所示的实施例中,每一子阵列62被复用(即由ymux解码,未示出)为两个字,每一字具有16位。在一个实施例中,在子阵列阵列62中,每一行具有2048个存储器单元,第一批1024个单元被复用为第一字(将要被编程)并且第二批1024个单元被复用为第二字(将要被编程)。

[0007] 在擦除期间,存储器单元的单位(诸如许多字节(每字节8位))的块或组立刻被擦除。擦除操作将多个位设置为“FF”(十六进制)或“11111111”的状态。在编程期间,所选的字节位通过将电子注入浮栅上而被编程为“0”的逻辑状态。然而,电荷泵52必须能够提供编程电流就好像字节的所有位将被编程。从而,电荷泵52必须被设计具有提供编程电流的能力,就好像字节的所有位将被编程为“00”(十六进制)或“00000000”的状态。而且,为增强性能,许多字节被同时编程。这为电荷泵52提供大的编程电流增加了更多负荷。由于能够提供大电流的电荷泵52需要硅小片上的大量不动结构,较大电荷泵52占据更多硅不动结构。从而,本发明的一个目的是减少电荷泵52的尺寸。

[0008] 最后,校验位在现有技术中众所周知。校验位是与多个位(诸如一个字节)关联的位,其中校验位被编程至一种表示对存储在字节中的数据进行错误检验的状态。然而,校验位典型地已与易失性存储器单元一起使用,并且已仅用于错误校正。

发明内容

[0009] 本发明为具有用于提供编程电流的电荷泵的非易失性存储器装置。存储器装置具有非易失性存储器单元的阵列,其中每一存储器单元由编程电流来编程。非易失性存储器单元的阵列被划分为多个单位,每一单位包括多个存储器单元。指示器存储器单元与每一单位的非易失性存储器单元关联。当每一单位的某一比例或更少的存储器单元将被编程时,编程电路使用编程电流对每一单位的存储器单元进行编程,并且当多于该每一单位的某一比例的存储器单元将被编程时,编程电路使用编程电流对每一单位的存储器单元的逆以及与每一单位关联的指示器存储器单元进行编程。

[0010] 本发明也为对依赖于电荷泵在编程操作期间提供编程电流的非易失性存储器装置进行编程的方法。

附图说明

[0011] 图1是具有用于在其上存储电荷的浮栅的、现有技术的非易失性存储器单元的横截面视图,本发明的编程方法对该非易失性存储器单元适用并且该非易失性存储器单元能够用于本发明的存储器装置中。

[0012] 图2是现有技术的非易失性存储器装置的模块级示图,该装置包括电荷泵,其提供编程电流以供在非易失性存储器单元的编程期间使用。

[0013] 图3是图2所示的现有技术的非易失性存储器装置的阵列部分的详细模块级示图。

[0014] 图4是本发明的非易失性存储器装置的一个实施例的阵列部分的详细模块级示图。

[0015] 图5是本发明的非易失性存储器装置的另一实施例的阵列部分的详细模块级示图。

[0016] 图6是用于在本发明的非易失性存储器装置中使用的、编程电路的模块级示图。

[0017] 图7是用于在本发明的非易失性存储器装置中使用的、读出电路的模块级示图。

[0018] 图8是用于在本发明的非易失性存储器装置中使用的、数据输入‘0’检测模拟电路。

[0019] 图9是用于在本发明的非易失性存储器装置中使用的、数据输入‘0’模拟检测的流程图。

[0020] 图10是用于在本发明的非易失性存储器装置中使用的、数据输入‘0’检测数字电路的模块图。

[0021] 图11是用于在本发明的非易失性存储器装置中使用的、编程的流程图。

[0022] 图12是用于在本发明的非易失性存储器装置中使用的、编程的备选流程图。

[0023] 图13是用于在本发明的非易失性存储器装置中使用的、编程的另一备选流程图。

具体实施方式

[0024] 参考图4,示出本发明的非易失性存储器阵列200的一个实施例的模块级示图。非易失性存储器阵列200能够代替阵列60用于图2中所示的非易失性存储器装置50中。由于本发明的存储器阵列200与存储器阵列60相似,相同的标号将用于相似部件。

[0025] 阵列200包括多个子阵列62(a-d),其中诸如存储器单元10的多个存储器单元成多行及多列布置。行解码器64与横越子阵列62a、62b、62c和62d的存储器单元10的每行关联。读出放大器70a和70b与每一子阵列(例如子阵列62a)中的存储器单元10的每列关联。信号从读出放大器70提供给输出寄存器72。在图4所示的实施例中,每一子阵列62具有多行,其中每行具有2048个存储器单元10,第一批1024个单元被复用为第一字(每一字具有16位将要被编程)并且第二批1024个单元被复用为第二字(将要被编程)。阵列60中存储器单元10的整行从而包括多个单位的128个存储器单元10,其中每一单位被复用为8个字word0-word7并且耦合到读出放大器SA0-SA127 70中。

[0026] 阵列200还包括存储器单元的阵列80(被称为指示器存储器单元),其能够为与子

阵列62(a-d)中所使用的(即存储器单元10)相同类型的存储器单元,阵列80的每行具有多个指示器存储器单元。存储器单元的阵列80位于子阵列62(d)和行解码器64之间,指示器存储器单元10的一行与子阵列62(a-d)中的存储器单元10的每行关联。指示器单元阵列80的位置紧邻于行解码器64以具有从行解码起的最快的访问时间来减轻由于与指示器位关联的逻辑而引起的总体速度的影响。在一个实施例中,一个指示器单元与每一子阵列62中两个所选字(32个所选存储器单元10)关联。在另一实施例中,一个指示器存储器单元10与每一所选单位的8个字(128个存储器单元10)关联。当然,在下文中将会看到,本发明并不如此受限,并且每一指示器存储器单元10可与一个单位的存储器单元关联,其中该单位的存储器单元具有任何数量。读出放大器82与阵列80中指示器存储器单元10关联。将会看到,读出放大器82的输出选通(gate)来自读出放大器70(a-h)的输出进入寄存器72(a-h)。在一个实施例中,每一指示器位(单元)由一个(基本)存储器单元10组成。在另一实施例中,每一指示器位由2个或更多(基本)存储器单元10组成,在这种情况下指示器单元的感测由于数据输出的检查和反转将更快而不影响总体速度。

[0027] 本发明的操作通过以下示例可能会最好理解。让我们假定指示器存储器单元10关联的一个单位的存储器单元10为一个字节或8位。在该字节或8位被擦除后,其处于“FF”(十六进制)状态或“11111111”的位格式。如果将要被编程形成字节的新数据为“00000000”,那么作为提供充足的编程电流来改变8个存储器单元10的状态的替代,本发明的存储器装置200代替地仅将关联的指示器存储器单元10从“1”编程为“0”并且保持关联字节的位的状态为“11111111”。以这种方式,仅1位被编程,并且仅对1个存储器单元10编程足够的编程电流需要由电荷泵52提供。

[0028] 本发明的其他示例如下。再次让我们假定指示器存储器单元10关联的一个单位的存储器单元10为一个字节或8位。在该字节或8位被擦除后,其处于“FF”(十六进制)状态或“11111111”的位格式。如果将要被编程形成关联字节的新数据为“01010101”,其为关联单位中位的50%,那么编程电路导致“01010101”的位格式被编程为关联字节的位。在这种情况下,仅4位被编程,并且所需的总编程电流是用于4位的编程。然而,如果将要被编程为关联字节的新数据为“01010100”,其多于关联字节中位的50%,那么编程电路导致反转位格式或“10101011”被编程为关联字节的位并且导致关联的指示器位被编程为“0”的状态。在这示例中,再次仅4个位需要被编程并且总编程电流再次为仅4位。从这些示例中可看出,通过本发明的方法,与现有技术的阵列60相比,电荷泵52需要提供以对阵列200编程的、编程电流的最大值仅为以其他方式对于现有技术的存储器阵列60所需的编程电流的二分之一或50%。在这个描述的方法中,将被存储的数据格式主要为‘1’数据(因如果多于50%的‘0’数据将被反转成‘1’数据)。备选实施例不需正好50%但可以为大约50%或任何比例数量。在少于50%的情况下,芯片操作是次优的。

[0029] 参考图5,示出本发明的非易失性存储器阵列300的另一实施例的模块级示图。非易失性存储器阵列300与图4中示出的非易失性存储器阵列200类似并且能够代替阵列60用于图2中所示的非易失性存储器装置50中。由于本发明的存储器阵列300与存储器阵列200类似,相同的标号将用于相似部件。

[0030] 阵列300包括多个子阵列62(a-d),其中诸如存储器单元10的多个存储器单元成多行多列布置。行解码器64与横越子阵列62a、62b、62c和62d的存储器单元10的每行关联。读

出放大器70a和70b与每一子阵列(例如子阵列62a)中的存储器单元10的每列关联。信号从读出放大器70提供给输出寄存器72。在图5所示的实施例中,每一子阵列62具有多行,每一行具有2048个存储器单元10,第一批1024个单元被复用为第一字(每一字具有16位将要被编程)并且第二批1024个单元被复用为第二字(将要被编程)。阵列60中存储器单元10的整行从而包括多个单位的128个存储器单元10,其中每一单位被复用为8个字word0-word7并且耦合至读出放大器SA0-SA12770。

[0031] 阵列300还包括多个存储器单元(也称为指示器存储器单元)的阵列80(a-d),其能够为与子阵列62(a-d)中使用的相同类型的存储器单元,即存储器单元10。指示器存储器单元的每一阵列80与存储器单元的子阵列62关联并且位于关联子阵列62和关联的读出放大器70之间,其中一个指示器存储器单元10与子阵列62中存储器单元10的关联(associative)列关联。与有关每一指示器存储器单元与一行存储器单元关联的先前讨论相似,每一指示器存储器单元10不需与一行存储器单元关联并且可与一个单位的存储器单元关联,其中该单位的存储器单元具有任何数量。读出放大器82与指示器存储器单元的每一阵列80关联,读出放大器的输出选通来自读出放大器70的输出进入寄存器72。

[0032] 参考图6,示出用于测试将要被编程的数据以确定是该数据还是其反转将依照本发明的方法被编程的电路210。电路210包括数字计数器电路246,反转电路204和复用器MUX 202。数字计数器电路246用于对数据输入流中逻辑位‘0’的数量进行计数。如果逻辑位‘0’的数量多于某一预定的数字比例(例如50%),则数据经由选择MUX202和反转电路204被反转。备选地,可使用数字比例,例如33%、66%、51%。MUX电路202的输出为将存至存储器单元10的数据。

[0033] 参考图7,示出用于依照本发明的方法选通来自读出放大器70的输出进入寄存器72的数据输出复用器电路260。数据输出复用器电路260包括指示器读出放大器82、读出放大器70、反转电路264和复用电路MUX 262。来自阵列80的指示器存储器单元输入至指示器读出放大器82。来自阵列62的存储器单元10输入至读出放大器70。取决于指示器读出放大器82的输出,读出放大器70的输出经由MUX262和反转电路264馈送或由MUX 262和反转电路264所反转。MUX262的输出典型地通向耦合到IOBUF(输入/输出缓冲器)电路的DOUT(数据输出)电路72。

[0034] 参考图8,示出模拟数字‘0’计数器(检测)电路310。电路310包括多个使能偏置电路320、PMOS晶体管316、PMOS晶体管330、电流偏置电路332。使能偏置电路320包括电流偏置电路IBIAS 324和NMOS晶体管322。反转的数据输入IN/输入至NMOS 322的栅极,因此如果数据输入IN=‘0’,反转的数据输入IN/=‘1’将使能NMOS晶体管322。偏置电路324实现为单位偏置电流IBIAS,例如IBIAS=2 μ A。例如存在16个数据输入IN<0-15>,因此存在16个电路320。例如如果存在9个数据输入=‘0’(其他7个数据输入=‘1’),导致9个单位的偏置电流IBIAS 324使能至PMOS 316中。这电流被镜像至PMOS晶体管330中。例如电路332具有M $\times$ IBIAS,其中M=8并且IBIAS=2 μ A。PMOS 330中的电流为9 $\times$ IBIAS而电路332中的电流为8 $\times$ IBIAS,因此输出334将会被拉高,这指示数字‘0’的数量比16个数据输入位中的8个多。电路310同时也在所有数据输入位上进行操作。

[0035] 参考图9,示出数字式数字‘0’检测的算法流程图。在开始步骤410,初始化序列进行初始化Dn=D0(例如64个数据输入中的第一数据输入)并且时钟脉冲CKn=CK0(第一时

钟),并且计数 $C_n=0$ 。接下来在步骤412,检查是否数据输入 $D_n=0$ (数字‘0’),如果Y(是)那么转到步骤414来使计数 C_n 增加1($C_n=C_n+1$),如果N(否)那么转到步骤416来保持计数 C_n 与当前计数相同。接下来转到步骤418将时钟脉冲增加至下一时钟脉冲 $CK_n=CK_n+1$ 并且输入下一数据输入 $D_n=D_n+1$ 。接下来转至步骤420并且检测是否 $n=K$,例如 $K=64$,如果否那么转回步骤412循环并且重复对数字‘0’的检查。如果 n 确实等于 K ,那么转到步骤422。这时检查是否计数 $C_n>M$,例如 $M=33$,如果为真那么检测‘0’ $DET0=true=1$,如果否那么 $DET0=false=0$ 。接下来是最终步骤424——完成序列。

[0036] 参考图10,示出数字式数字‘0’检测电路500。电路500包括a-bit计数器510和检测逻辑Det-Logic电路520。a-bit计数器提供时钟脉冲 CK_n 并且检测逻辑520提供对 D_n 的数字‘0’检查,用于 C_n 、 D_n 、 CK_n 的增量功能,对计数 C_n 、初始化、启动以及完成功能的检查。例如对于64个数据输入位 $D<0:63>$,如果存在33个数据输入 $=0$,则输出 $DECT0\ 522=1$ 指示‘0’数字的数量多于64个中的32个。电路500每次按顺序地逐位进行操作,例如对于64个数据输入位,电路500在64个时钟脉冲上进行操作。

[0037] 图11、12和13中的编程算法能够与上述发明相结合地用于仅在所需数据位进行操作中的进一步的操作优化。

[0038] 参考图11,示出用于‘0’数据输入格式的有效编程的编程算法600的流程图。在这个实施例中,存在64个数据输入位(在图4和5中存在128个数据输入位)并且在编程中能够仅每次编程16个‘0’数据输入位(意味着高压电荷泵设计成每次为16位供电(以使电荷泵面积最小化),例如对32位编程将需要按顺序进行操作的两个编程脉冲,其中每一编程脉冲每次对16位编程)。算法将从第一数据输入和‘0’数据输入数字的计数开始,一旦达到16个‘0’数字,就执行编程(写)操作以将这些16个位编程写入合适的存储器单元10,意味着将这些单元的存储器单元存储状态从擦除状态‘1’改为编程状态‘0’。在这编程完成后,接下来是计数并且识别将要被编程的下一批16个‘0’数据输入位。序列继续进行直到对于编程操作全部64个数据输入被考虑。

[0039] 编程算法序列按如下方式工作。在擦除全部所选存储器单元的擦除操作之后,算法600用于将数据输入编程写入存储器单元10中。在步骤开始602,初始化步骤进行初始化 $D_{in}=D_{in0}$,接下来是步骤604使能计数 $C=0$,接下来是步骤606检查如果 $D_{in}=0$,那么使 D_{in} 能够随后被编程(写)并且使计数 C 增加1($C=C+1$),如果 $D_{in}=1$,那么使 D_{in} 不能用于随后的编程操作并且保持计数 C 为当前值。接下来是步骤608转到下一数据输入 $D_{in}=D_{in}+1$,接下来是步骤610检查是否已到达最终数据输入 $D_{in}=D_{in63}$ 。如果没有,那么转到步骤612检查计数 C 是否已到达16,如果是,就转到步骤614来对这些16个被使能的数据输入 D_{in} 执行编程操作。如果 C 不是16,那么转回步骤606来检查以使能下一个‘0’数据输入。如果在步骤610中 $D_{in}=D_{in63}$,那么转到步骤616,如果现在计数 $C<1$ (没有存储器单元需要被编程),那么转到最终步骤666完成,如果 $C\geq 1$,那么转到步骤618对已经被使能用于编程的数据输入位执行编程操作并且随后转到最终步骤666完成。

[0040] 参考图12,示出用于‘0’数据输入格式的有效编程的另一编程算法600的流程图。在这个实施例中,‘0’数字检测在字(16位)基础上进行操作。能够被编程(写)的、所选字的零的数量在这个实施例中为每次16个。示出的示例在这个实施例中针对4个字。这算法仅采用‘0’数字对必要的字进行编程,意味着将存储器单元存储状态从擦除状态(‘1’)改为编程

状态(‘0’)。这种方法与图11中的方法(其在位而不是字上进行操作)相比不大有效,但需要较简单的实现。

[0041] 该算法按如下方式工作。首先是开始步骤702,其对任何必要的参数进行初始化。接下来是步骤704检查4个字W0-3的‘0’数字的数量是否 <1 (没有存储器单元需要被编程),如果是,那么转到最终步骤777完成(意味着没有编程操作需要进行)。如果否,那么转到步骤706检查是否4个字W0-3的‘0’数字的数量 <16 ,如果是,那么转到步骤708对所有字编程(由于将要被编程的位的数量小于16位,16位为能够同时被编程的位的数量)并且随后转到最终步骤777完成。如果在步骤706中,如果4个字W0-3的‘0’数字的数量 <16 为否,那么转到步骤710检查2个字W0,W1的‘0’数字的数量是否 <1 ,如果否,那么转到步骤712检查2个字W0,W1的‘0’数字的数量是否 <16 ,如果是,那么同时对两个字W0,W1编程,随后转到步骤720,如果否,那么单独地对W0编程以及对W1编程(即,按顺序地每次一个字),随后转到步骤720。如果在步骤710,回答为是,那么转到步骤720。在步骤720中,这时检查字W2,W3的‘0’数字的数量是否 <1 ,如果是,那么转到最终步骤777完成,如果否,那么转到步骤722检查W2,W3的‘0’数字的数量是否 <16 。如果是,那么转到步骤724同时对字W2,W3两者编程,如果否,那么转到步骤726单独地(按顺序地)对字W2,W3编程,随后转到最终步骤777完成。

[0042] 参考图13,示出用于‘0’数据输入格式的有效编程的另一编程算法600的流程图。在这个实施例中,‘0’数字检测在字(16位)基础上进行操作。能够被编程(写)的、所选字的零的数量在这个实施例中为每次16个。示出的示例在这个实施例中针对4个字。这算法仅采用‘0’数字对必要的字编程。这种方法与图12中的方法相比更有效,但需要用于更详尽的字组合检查的更复杂的实现。

[0043] 该算法按如下方式工作。首先是步骤802开始对任何必要的参数进行初始化。接下来是步骤804,其检查4个字W0-3的数据输入中‘0’数字的数量是否 <16 。如果是,那么转到步骤806来同时对所有字编程并且随后转到最终步骤888完成。如果否,那么转到步骤810检查对于字W0,W1的‘0’数字的数量是否 <16 。如果是,那么转到步骤812同时对字W0,W1两者编程。在步骤812之后,转到步骤814检查字W2,W3的‘0’数字的数量是否 <16 。如果是,转到步骤816同时对字W2,W3两者编程并且随后转到最终步骤888完成。如果在步骤814,回答为否,那么转到步骤818单独地(按顺序地)对字W2编程以及对W3编程,随后转到最终步骤888完成。类似地,用于4个字中剩余字的详尽组合的步骤820-828,步骤830-838,步骤840-848,步骤850-858,步骤860-868类似于步骤810-818。步骤870单独地对每一字W0,W1,W2,W3编程并且随后转到最终步骤888完成。

[0044] 用于图12和图13中的上面4个字的数据输入(64个数据输入位)的阵列结构是分别与图4和图5中阵列200或阵列300类似的一个阵列结构,仅由子阵列62a和62b组成(没有子阵列62c或62d)。

[0045] 对于由多个子阵列(62a-d)组成的存储器阵列,在所选的(使能的)字或多个字将要被编程的情况下,与这些字关联的子阵列被使能(编程偏置、编程高压以及电路被使能进入阵列存储器单元10中)并且与未选择的(禁能的)字关联的子阵列被禁能(编程偏置、编程高压以及电路被禁止进入阵列存储器单元10中)以使任何高压和偏置最小化以免扰乱未选的字阵列。

[0046] 在上述发明中,指示器位和操作算法与编程数据格式关联。并且数据输入‘0’与被

编程状态关联。数据‘1’能够通过使数据输入反转以及使数据输出反转而与编程状态关联。备选实施例具有与用于如下技术的擦除数据格式关联的指示器位和操作算法,该技术能够针对擦除操作进行优化。

[0047] 另一实施例具有与用于如下技术的数据格式关联的指示器位和操作算法,该技术能够针对持续时间(对于工作寿命的擦除/编程周期的数量)和数据保持(对于例如10年的工作寿命永久保持数据的能力)进行优化,例如持续时间和数据保持更倾向于‘1’状态以用于最佳持续时间和数据保持性能,那么存在存储器中的数据格式在由操作算法处理后将会有更多的数据处于‘1’状态。另一实施例具有与用于如下技术的数据格式关联的指示器位和操作算法,该技术能够针对扰乱进行优化(意味着在擦除或编程操作中未选的单元的数据状态较少变化),因为某种数据格式不易扰乱。

[0048] 另一实施例具有与用于如下技术和/或设计的数据格式关联的指示器位和操作算法,该技术和/或设计能够针对总体最佳的误比特率(BER)而进行优化(意味着在擦除或编程或读操作中存储器单元的数据状态较少变化)。这将涉及随工作寿命和用户应用的操作模式的技术和产品设计的特性描述,以找到最匹配将存储在存储器单元中的数据格式(例如大部分‘1’或大部分‘0’数据格式)的算法。

[0049] 另一实施例具有与用于如下技术的数据格式关联的指示器位和操作算法,该技术针对读功率消耗而进行优化(诸如将要存在存储器中的数据格式在由用于存储器单元10的算法处理后将会为更多的数据‘0’状态,由于处于‘0’状态没有存储器单元电流,意味着没有来自存储器单元10的功率消耗)。另一实施例具有与用于如下技术和设计的数据格式关联的指示器位和操作算法,该技术和设计根据取决于用户用法的功率曲线针对所有操作模式(例如读、编程、擦除)的总体功率消耗而进行优化。

[0050] 从前述内容可看出,本发明能够应用到需要电荷泵提供编程电流的任何非易失性存储器装置上。从而,本发明能够应用到具有用于存储电荷的浮栅的非易失性存储器装置,其中电荷泵在编程期间在浮栅上存储电荷,并且本发明并不限于图1中所示的存储器单元10的类型。而且,本发明也能够应用到具有用于存储电荷的捕捉电荷层的存储器装置,其中电荷泵在编程期间在捕捉电荷层中存储电荷。

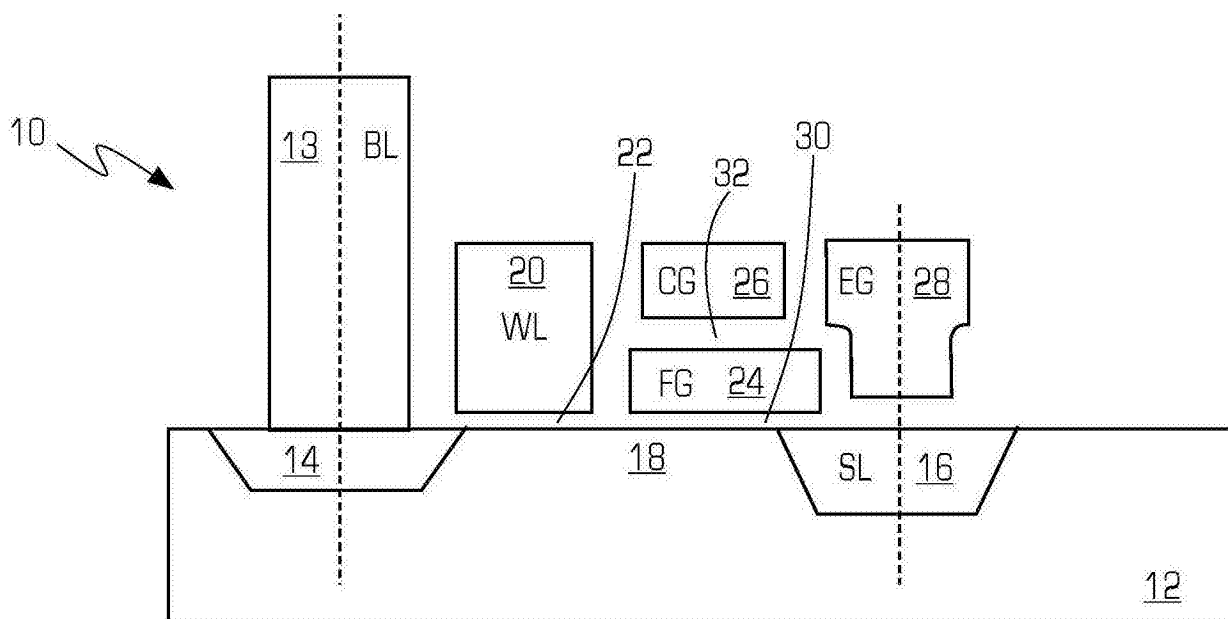


图1 (现有技术)

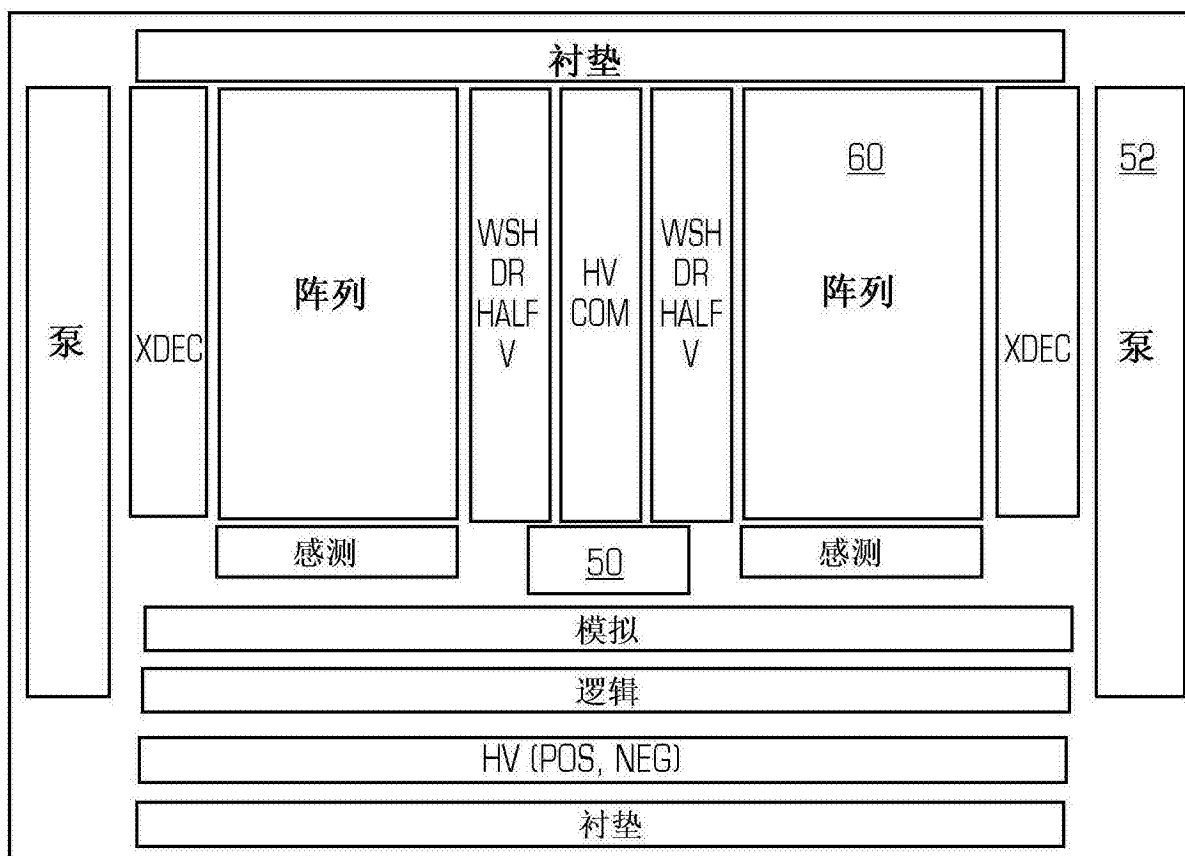


图2(现有技术)

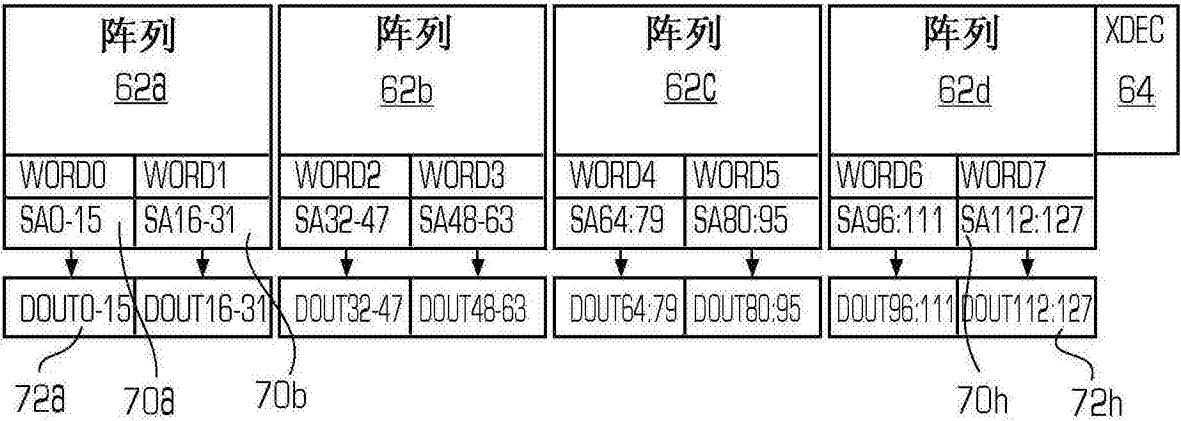


图3 (现有技术)

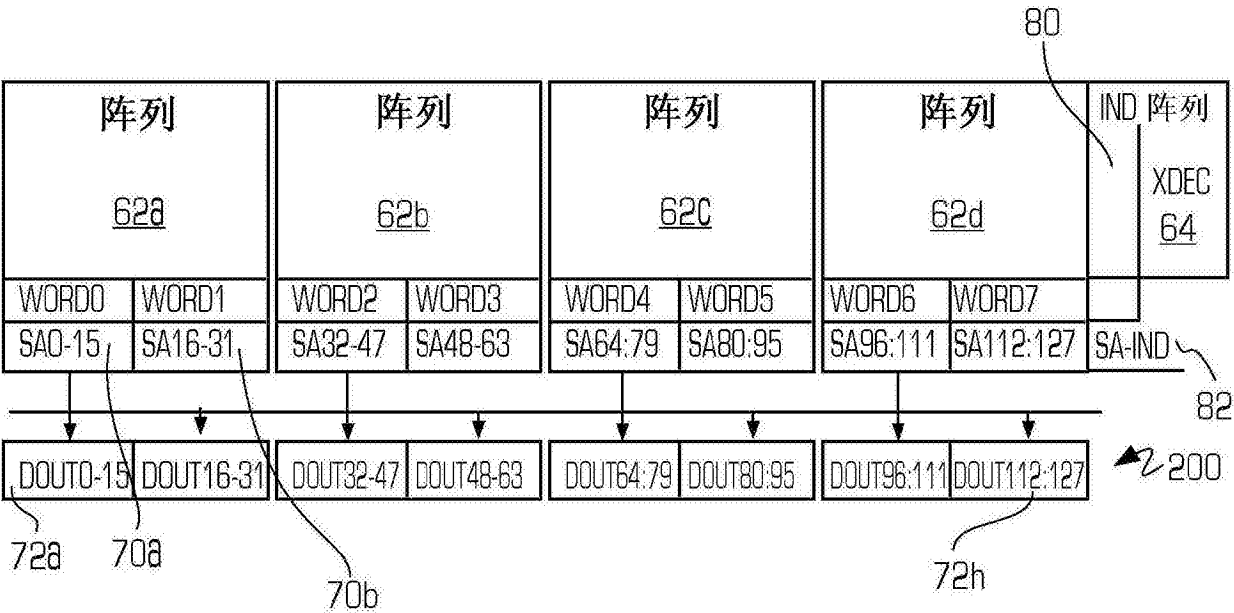


图4

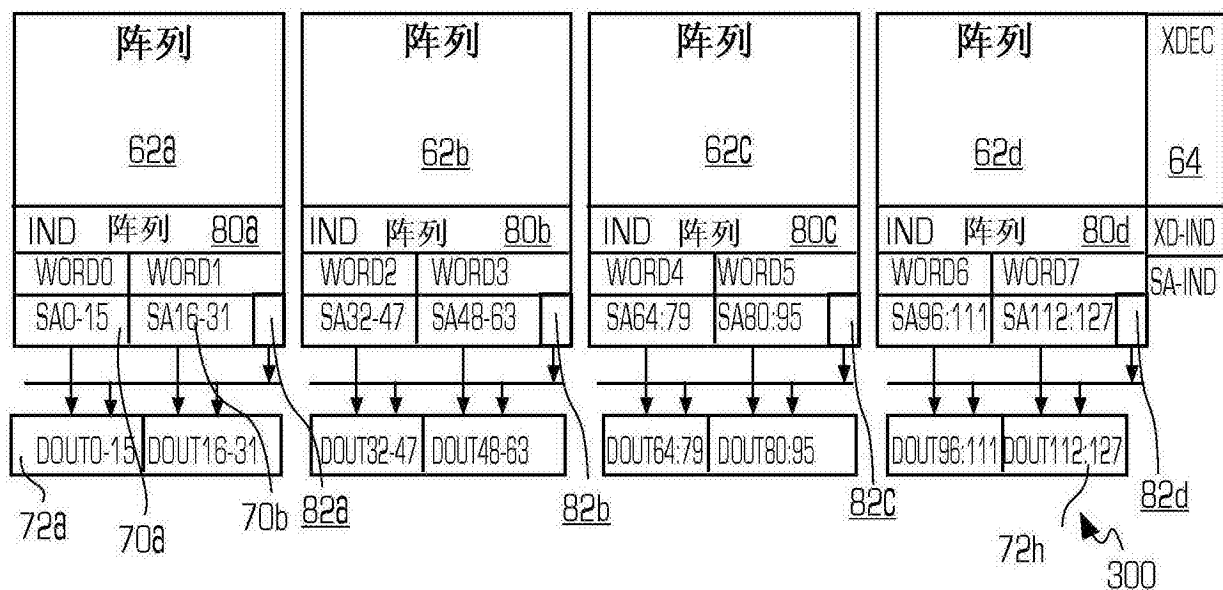


图5

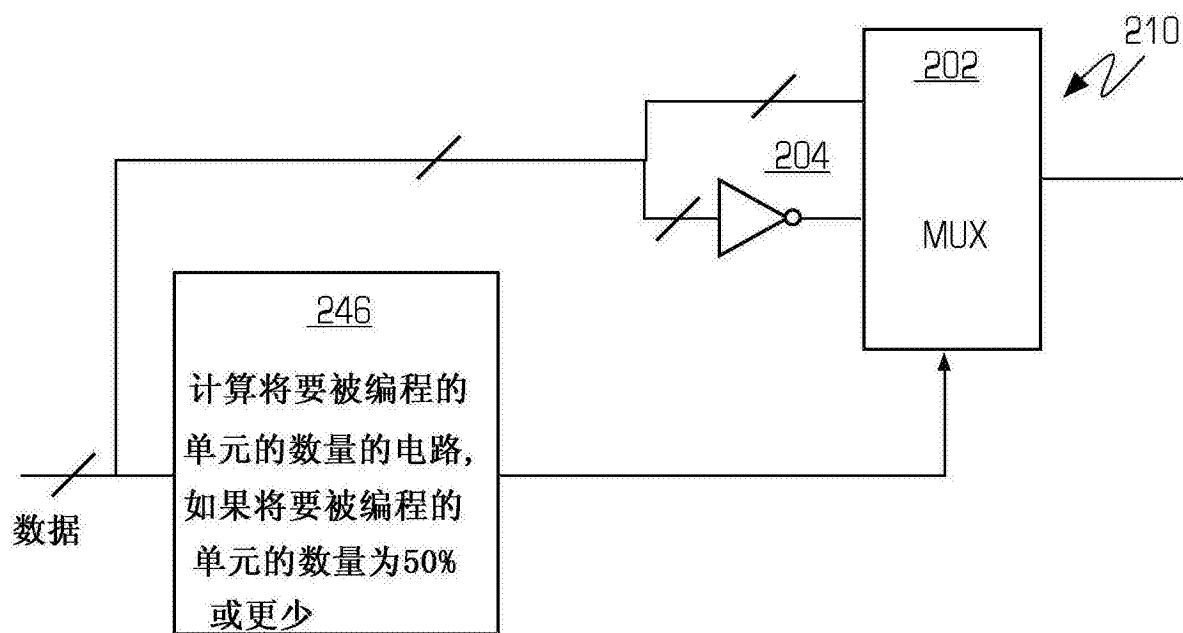


图6

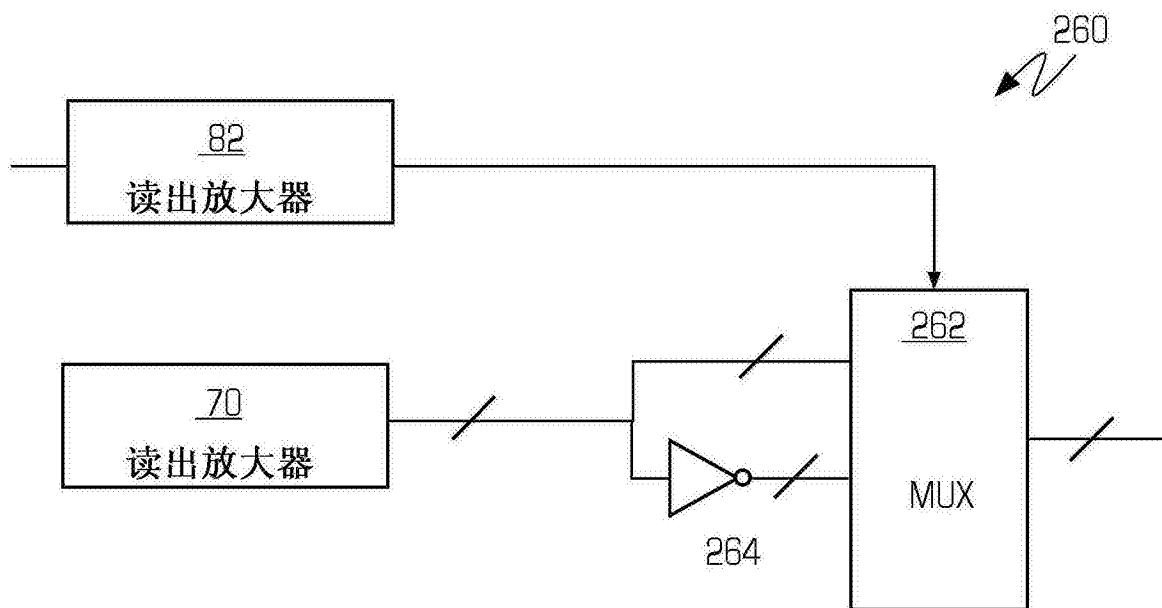


图7

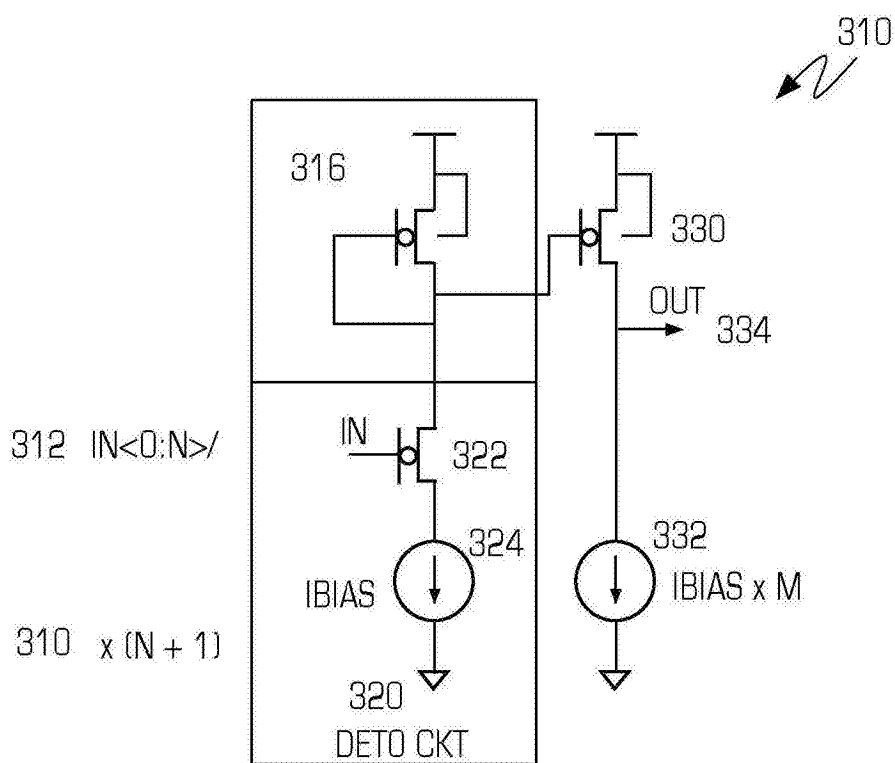


图8

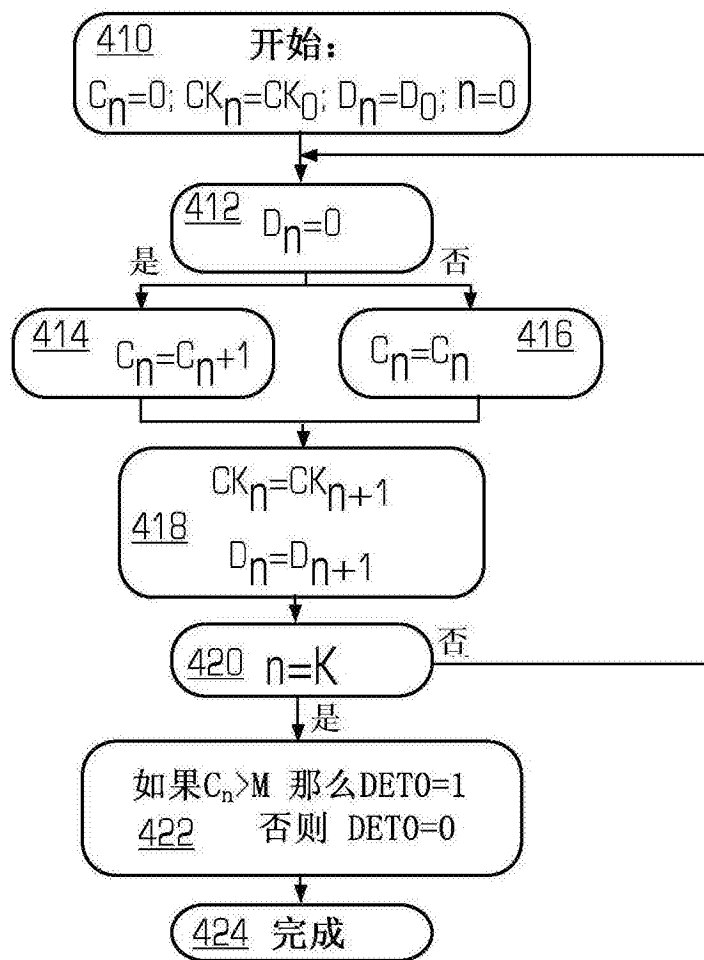


图9

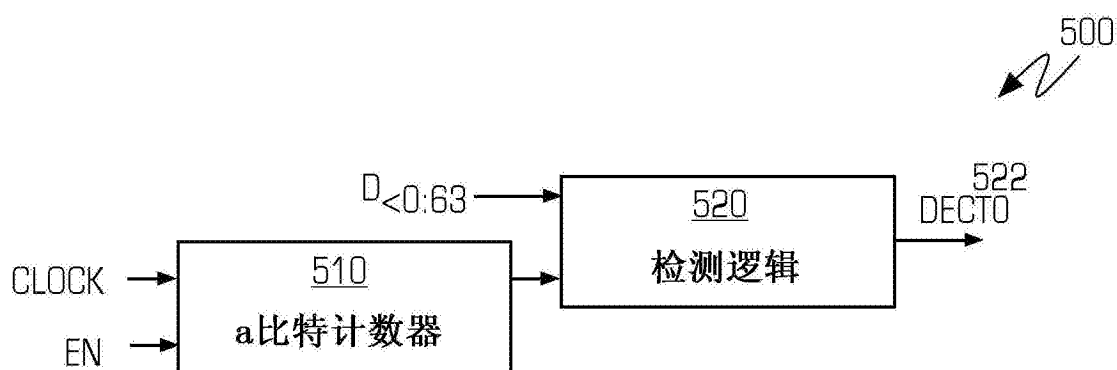


图10

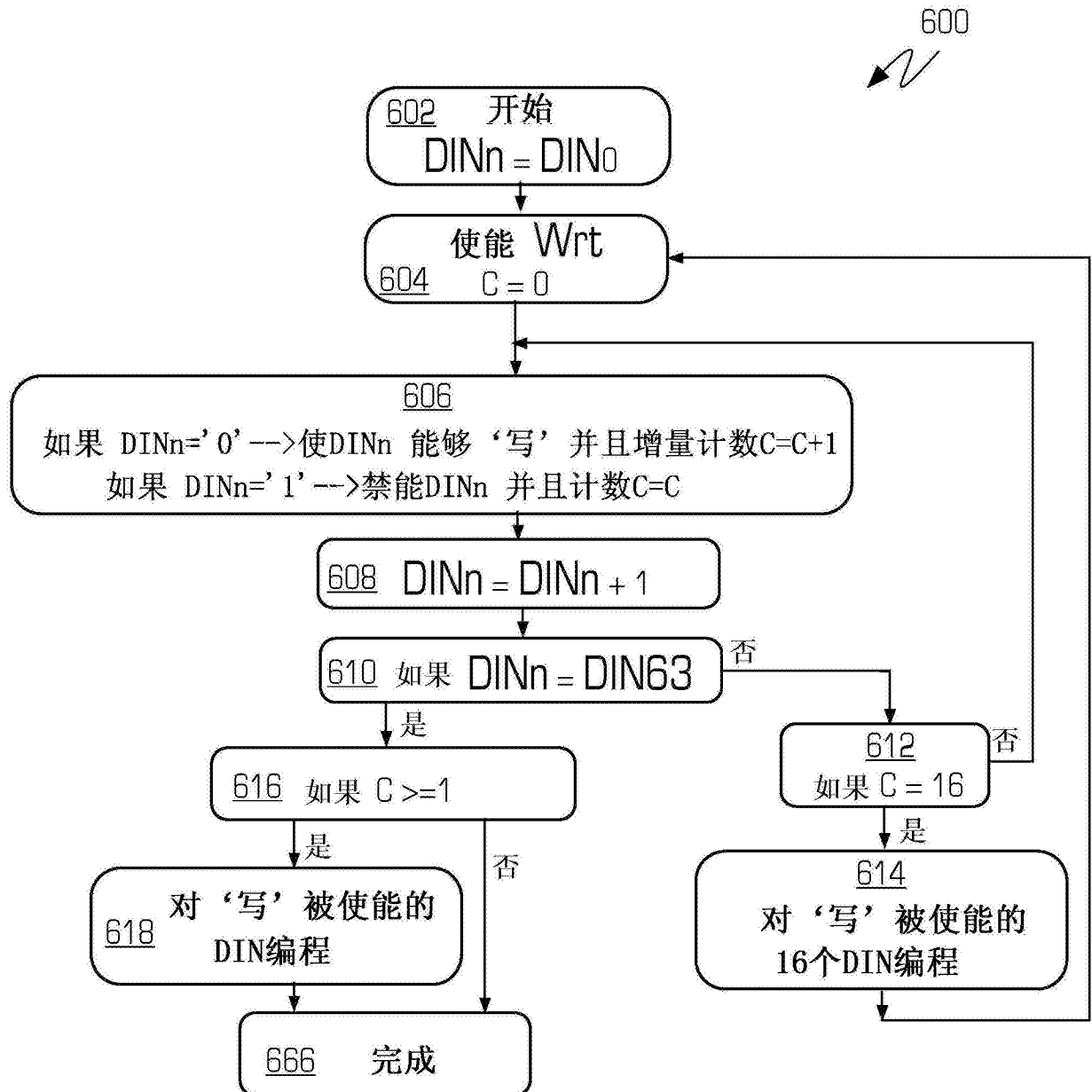


图11

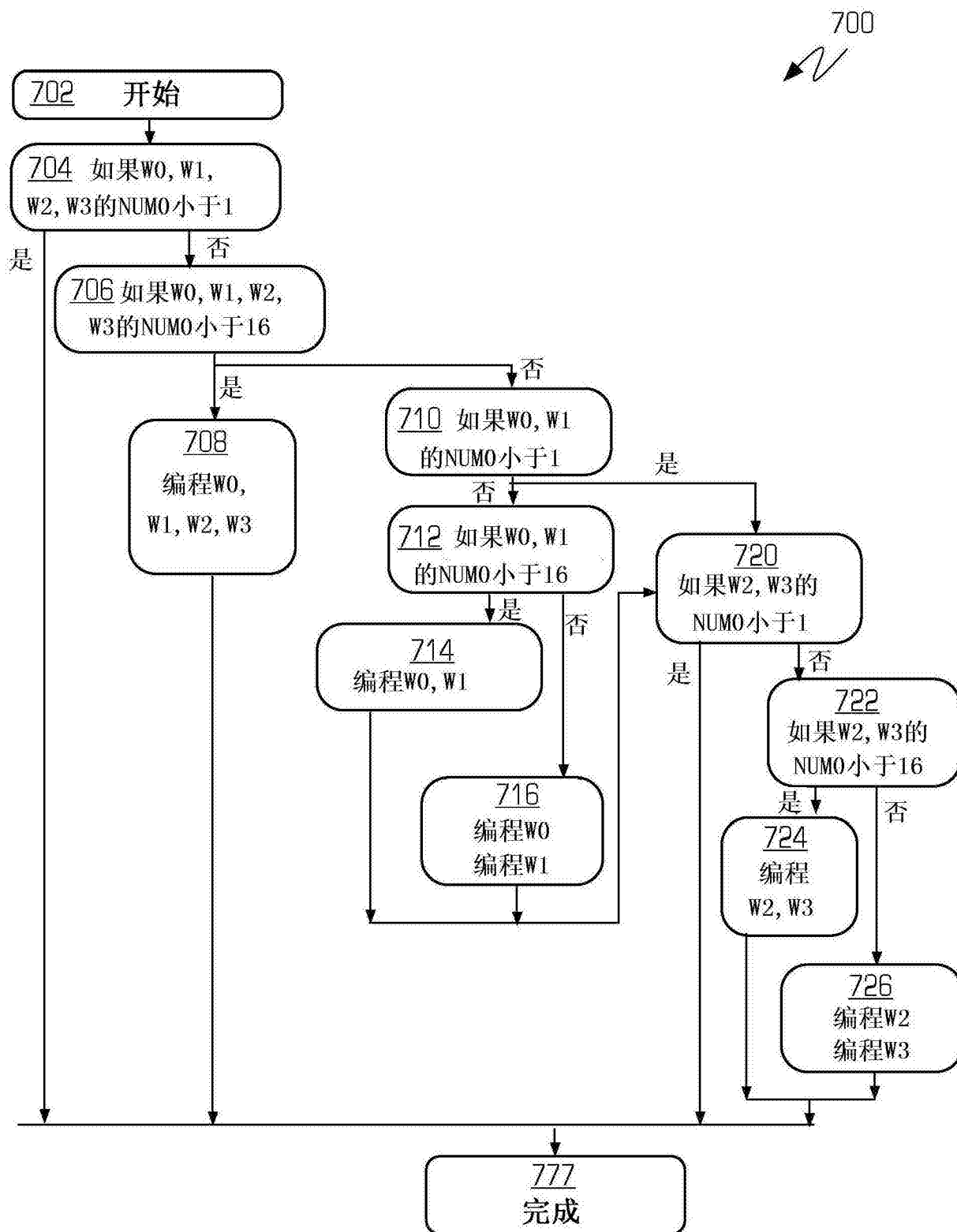


图12

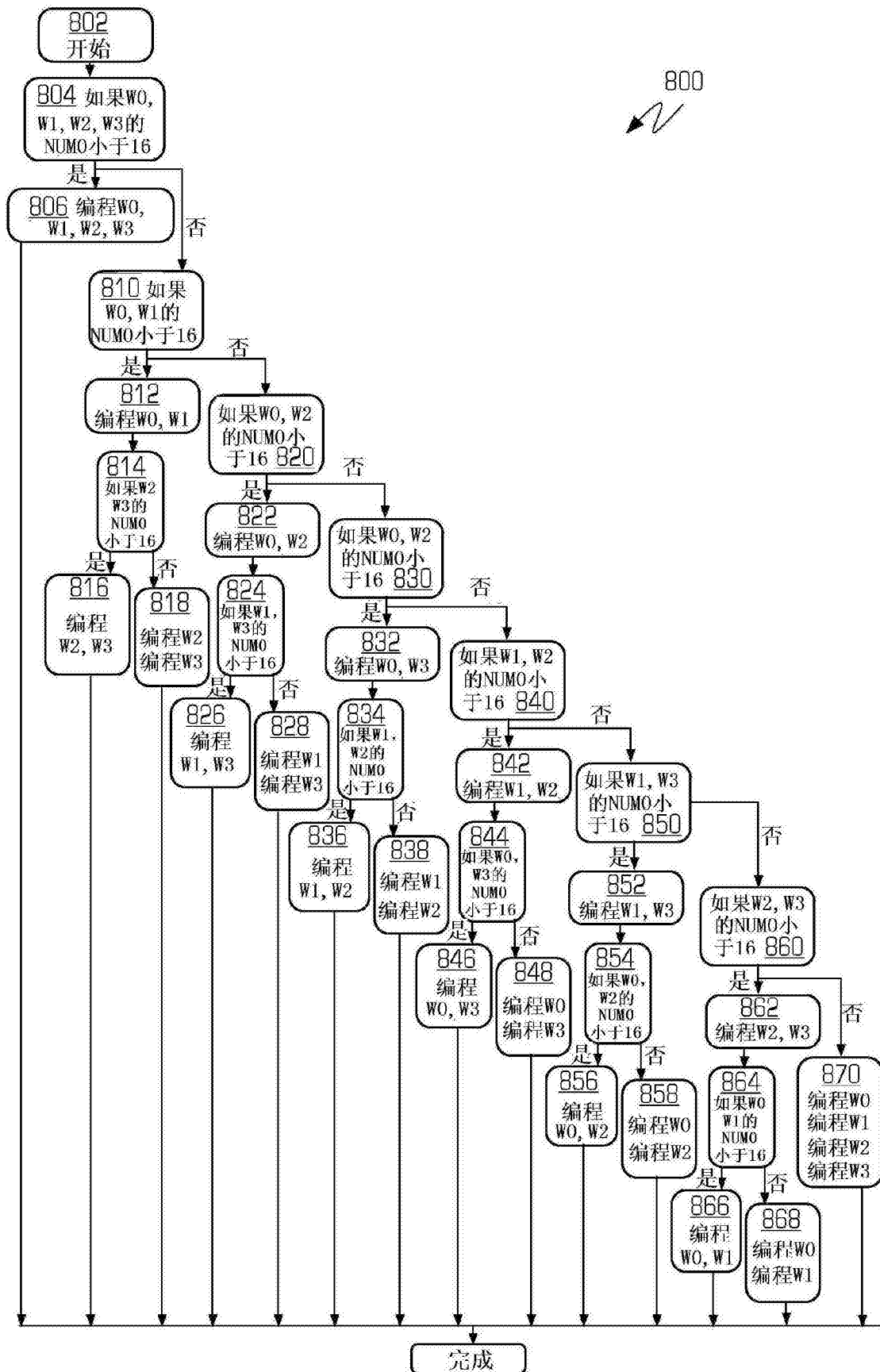


图13