(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103000224 A
(43)申请公布日 2013.03.27

(21)申请号 201110274850.7

(22)申请日 2011.09.16

(71)申请人 中国科学院微电子研究所
地址 100029 北京市朝阳区北土城西路 3 号

(72)发明人 刘明 陈映平 冀永辉 谢常青

(74)专利代理机构 中科专利商标代理有限责任公司 11021
代理人 周国城

(51)Int.Cl.
G11C 16/14(2006.01)
G11C 16/34(2006.01)

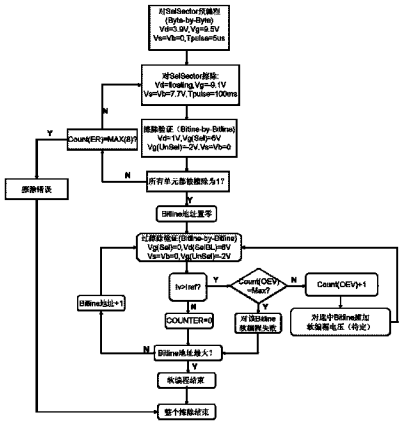
权利要求书 1 页 说明书 4 页 附图 3 页

(54)发明名称

一种对存储器芯片进行擦除的方法

(57)摘要

本发明公开了一种对存储器芯片进行擦除的方法，将存储器芯片的共源极从接地方式改为接限流装置，该方法包括：对存储单元中的块进行预编程；对所有的存储单元进行擦除；对擦除之后的存储单元进行验证；以及以位线为单位进行软编程验证。利用本发明，在有效缩短擦除时间的同时，由于限流装置的使用，使得虽然同时对多个单元进行擦除，也不会使流过存储单元的电流过大而使存储单元被破坏，这使得擦除操作的可靠性也得到了保证，所以特别适用大容量存储器芯片。



CN 103000224 A

1. 一种对存储器芯片进行擦除的方法，其特征在于，将存储器芯片的共源极从接地方式改为接限流装置，该方法包括：

对存储单元中的块进行预编程；

对所有的存储单元进行擦除；

对擦除之后的存储单元进行验证；以及

以位线为单位进行软编程验证。

2. 根据权利要求 1 所述的对存储器芯片进行擦除的方法，其特征在于，所述将存储器芯片的共源极从接地方式改为接限流装置，是将存储器芯片的存储单元的源端连接于限流装置的漏端。

3. 根据权利要求 1 所述的对存储器芯片进行擦除的方法，其特征在于，所述对存储单元中的块进行预编程包括：

在存储单元的栅端和漏端分别加正电压，使存储单元沟道中流过电流，电流中的电子被浮栅捕获，因而将存储单元编程至 0。

4. 根据权利要求 3 所述的对存储器芯片进行擦除的方法，其特征在于，所述在存储单元的栅端加正电压为 7V，在存储单元的漏端加正电压为 4V。

5. 根据权利要求 1 所述的对存储器芯片进行擦除的方法，其特征在于，所述对所有的存储单元进行擦除包括：

在存储单元的栅端加负压，在衬底加正压，将浮栅上的电子除去，因而将所有的存储单元擦除至 1。

6. 根据权利要求 5 所述的对存储器芯片进行擦除的方法，其特征在于，所述在存储单元的栅端加负压为 -9V，在衬底加正压为 7V。

7. 根据权利要求 1 所述的对存储器芯片进行擦除的方法，其特征在于，所述对擦除之后的存储单元进行验证包括：

以位线为单位，在存储单元的栅端和漏端分别加正压，通过读取流过存储单元中的电流判断是否擦除成功，如果擦除成功，则将位线地址置零，转入过擦除验证，否则对块再次进行擦除；如果对同一个块的擦除次数超过一定的次数，则认为擦除出错，产生错误信息，退出擦除。

8. 根据权利要求 7 所述的对存储器芯片进行擦除的方法，其特征在于，所述在存储单元的栅端加正电压为 5V，在存储单元的漏端加正电压为 1V。

9. 根据权利要求 1 所述的对存储器芯片进行擦除的方法，其特征在于，所述以位线为单位进行软编程验证包括：

以位线为单位，在存储单元的栅端和漏端分别加正压，通过读取流过存储单元中的电流判断是否擦除成功，如果擦除成功，则转入下一个位线地址；否则对该位线进行软编程，再转过擦除验证；如果对同一条位线的软编程次数超过设定的最大次数，则返回错误信息，进入下一个位线地址，直至选中的块中最后一条位线。

10. 根据权利要求 9 所述的对存储器芯片进行擦除的方法，其特征在于，所述在存储单元的栅端加正电压为 5V，在存储单元的漏端加正电压为 1V。

一种对存储器芯片进行擦除的方法

技术领域

[0001] 本发明涉及集成电路和非易失性存储器芯片技术领域，特别涉及一种对存储器芯片进行擦除的方法。

背景技术

[0002] 非易失性存储器是一个发展很快的领域，现在用于非易失性存储器芯片中的存储器件多采用浮栅结构存储器件，对该浮栅结构存储器件进行擦除多采用电可擦除方式。在擦除时，是在浮栅上加上负偏压，同时在衬底上加上正偏压，进而将浮栅上的电子移除，实现对该浮栅结构存储器件的擦除。

[0003] 对浮栅存储器件进行擦除存在的一个问题是过擦除现象。当浮栅上的电子被移除过多时，就会出现过擦除，此时浮栅上就会存在一定量的正电荷，使得存储器件处于微开启的状态，在器件沟道中会有漏电流存在。这样会引起误读现象，而且会对下一次编程造成阻碍，因为如果要编程完全，需要往浮栅上加入比正常情况下更多的电子以中和浮栅上存在的正电荷。

[0004] 对于存在过擦除现象的存储单元，需要进行软编程，即将浮栅上多余的正电荷除去，使存储单元恢复到正常的情况。软编程时在栅极端加上比编程时低的正电压，在漏极端加上一定的正电压，这样沟道中流过一定的电流，其中的一些电子会被捕获到浮栅上，进而中和过擦除时在浮栅上留下的正电荷。

[0005] 现有常用的对存储器芯片进行擦除的方法如图 1 所示，首先对需要擦除的存储单元进行预编程，然后进行擦除操作，之后针对位元 (Byte) 进行擦除验证，软编程和软编程验证也是 Byte-by-Byte。该方法具体描述如下 ：

[0006] (1) 对所选中的块 (Sector) 进行预编程，即在存储单元的栅端和漏端分别加适当的正电压 (如分别为 7V 和 4V)，使存储单元沟道中流过电流，电流中的电子被浮栅捕获，因而将选中的所有存储单元编程至 0 ；

[0007] (2) 对所有的存储单元进行擦除，即在存储单元的栅端加适当的负压 (如 -9V)，衬底加适当的正亚 (如 7V)，将浮栅上的电子除去，因而将所有存储单元擦除至 1 ；

[0008] (3) 对擦除之后的存储单元进行验证，即以 Byte 为单位，在存储单元的栅端和漏端分别加适当的正压 (如分别加 5V 和 1V)，通过读取流过存储单元中的电流判断是否擦除成功。如果擦除成功，则将 Byte 地址置零，转入过擦除验证，否则对 Sector 再次进行擦除。如果对同一个 Sector 的擦除次数超过一定的次数如 MAX(8) 次，为了防止对一些正常单元造成损害，一般认为擦除出错，则产生错误信息，退出擦除 ；

[0009] (4) 以 Byte 为单位进行过擦除验证 (即软编程验证)，即以 Byte 为单位，在存储单元的栅端和漏端分别加适当的正压 (如分别加 5V 和 1V)，通过读取流过存储单元中的电流判断是否擦除成功。如果通过，则转入下一个 Byte 地址 ；否则对该 Byte 进行软编程，再转过擦除验证 ；如果对同一 Byte 的软编程次数超过 Maxu 次，则返回错误信息，进入下一个 Byte 地址，直至选中的 Sector 中最后那个 Byte。

[0010] 对于存储容量小的存储芯片来说，这种方法可靠性高，而且时间也在合理的范围内，是一种不错的选择。但是当存储容量变大时，这种方法操作起来将消耗较长的时间，这对于存储器芯片的使用将是严重的缺点。

发明内容

[0011] （一）要解决的技术问题

[0012] 针对现有擦除方法时间过长的问题，本发明的主要目的在于提供一种对存储器芯片进行擦除的方法，以缩短擦除时间，并保证芯片的可靠性。

[0013] （二）技术方案

[0014] 为了达到上述目的，本发明提出了一种对存储器芯片进行擦除的方法，将存储器芯片的共源极从接地方式改为接限流装置，该方法包括：对存储单元中的块进行预编程；对所有的存储单元进行擦除；对擦除之后的存储单元进行验证；以及以位线为单位进行软编程验证。

[0015] 上述方案中，所述将存储器芯片的共源极从接地方式改为接限流装置，是将存储器芯片的存储单元的源端连接于限流装置的漏端。

[0016] 上述方案中，所述对存储单元中的块进行预编程包括：在存储单元的栅端和漏端分别加正电压，使存储单元沟道中流过电流，电流中的电子被浮栅捕获，因而将存储单元编程至 0。所述在存储单元的栅端加正电压为 7V，在存储单元的漏端加正电压为 4V。

[0017] 上述方案中，所述对所有的存储单元进行擦除包括：在存储单元的栅端加负压，在衬底加正压，将浮栅上的电子除去，因而将所有的存储单元擦除至 1。所述在存储单元的栅端加负压为 -9V，在衬底加正压为 7V。

[0018] 上述方案中，所述对擦除之后的存储单元进行验证包括：以位线为单位，在存储单元的栅端和漏端分别加正压，通过读取流过存储单元中的电流判断是否擦除成功，如果擦除成功，则将位线地址置零，转入过擦除验证，否则对块再次进行擦除；如果对同一个块的擦除次数超过一定的次数，则认为擦除出错，产生错误信息，退出擦除。所述在存储单元的栅端加正电压为 5V，在存储单元的漏端加正电压为 1V。

[0019] 上述方案中，所述以位线为单位进行软编程验证包括：以位线为单位，在存储单元的栅端和漏端分别加正压，通过读取流过存储单元中的电流判断是否擦除成功，如果擦除成功，则转入下一个位线地址；否则对该位线进行软编程，再转过擦除验证；如果对同一条位线的软编程次数超过设定的最大次数，则返回错误信息，进入下一个位线地址，直至选中的块中最后一条位线。所述在存储单元的栅端加正电压为 5V，在存储单元的漏端加正电压为 1V。

[0020] （三）有益效果

[0021] 从上述技术方案可以看出，本发明具有以下有益效果：

[0022] 1、本发明相对现有常用的对存储器芯片进行擦除的方法，不需要对电路做大的改动，只需要将原有存储器阵列的共源极从接地方式改为接限流装置，限流装置如图 3 所示，存储单元 320 的源端与限流装置 322 的漏端相连，这易于实现。

[0023] 2、在有效缩短擦除时间的同时，由于限流装置的使用，使得虽然同时对多个单元进行擦除，也不会使流过存储单元的电流（镜像作用使该电流与限流装置中的电流成比

例）过大而使存储单元被破坏，这使得擦除操作的可靠性也得到了保证，所以特别适用大容量存储器芯片。

附图说明

[0024] 图 1 是现有常用的对存储器芯片进行擦除的方法流程图；

[0025] 图 2 是依照本发明实施例的对存储器芯片进行擦除的方法流程图；

[0026] 图 3 是依照本发明实施例的限流装置的示意图。

具体实施方式

[0027] 为使本发明的目的、技术方案和优点更加清楚明白，以下结合具体实施例，并参照附图，对本发明进一步详细说明。

[0028] 本发明提出的对存储器芯片进行擦除的方法，其基本思想是在擦除验证、软编程和软编程验证时都针对一条位线 (Bit line)，并将存储器芯片的共源极从接地方式改为接限流装置，具体包括：对存储单元中的块进行预编程；对所有的存储单元进行擦除；对擦除之后的存储单元进行验证；以及以位线为单位进行软编程验证。

[0029] 为使擦除验证和软编程验证时流过存储单元的电流过大而使单元和电路受损，在存储单元的源端加入限流装置。为了保证软编程的可靠性，在存储单元的源端添加限流装置，使软编程电流不超过某一数值，这样在恒定电流的情况下，存在过擦除现象的单元相对于没有存在过擦除现象的单元，将流过更多的电流，使软编程主要集中于存在过擦除现象的单元上。在字线方向加入适当的控制电路，使得在擦除验证、软编程验证和软编程时选中所有的字线。

[0030] 图 2 是依照本发明实施例的对存储器芯片进行擦除的方法流程图，该方法具体描述如下：

[0031] (1) 对所选中的块 (Sector) 进行预编程，即在存储单元的栅端和漏端分别加适当的正电压（如分别为 7V 和 4V），使存储单元沟道中流过电流，电流中的电子被浮栅捕获，因而将选中的所有存储单元编程至 0；

[0032] (2) 对所有的存储单元进行擦除，即在存储单元的栅端加适当的负压（如 -9V），衬底加适当的正亚（如 7V），将浮栅上的电子除去，因而将所有存储单元擦除至 1；

[0033] (3) 对擦除之后的存储单元进行验证，即以位线 (Bit line) 为单位，在存储单元的栅端和漏端分别加适当的正压（如分别加 5V 和 1V），通过读取流过存储单元中的电流判断是否擦除成功。如果擦除成功，则将 Bitline 地址置零，转入过擦除验证，否则对 Sector 再次进行擦除。如果对同一个 Sector 的擦除次数超过一定的次数如 MAX(8) 次，一般认为擦除出错，则产生错误信息，退出擦除；

[0034] (4) 以 Bit line 为单位进行过擦除验证（即软编程验证），即以 Bit line 为单位，在存储单元的栅端和漏端分别加适当的正压（如分别加 5V 和 1V），通过读取流过存储单元中的电流判断是否擦除成功。如果通过，则转入下一个 Bit line 地址；否则对该 Bit line 进行软编程，再转过擦除验证；如果对同一条 Bitline 的软编程次数超过设定的最大次数 (Maxu)，则返回错误信息，进入下一个 Bit line 地址，直至选中的 Sector 中最后一条 Bit line。

[0035] 为了使上述方案中的擦除验证、过擦除验证和软编程可靠性提高，本发明提出了在存储单元的源端加限流装置的解决方案。图 3 是依照本发明实施例的限流装置的示意图，其中器件第一晶体管 310、过第二晶体管 312 和限流装置 322 起到镜像作用，存储单元 320 的源端与限流装置 322 的漏端连接。这是一个电流镜结构，通过 Vload 控制第一晶体管 310 的电流，通过第二晶体管 312 和限流装置 322 组成的电流镜将第一晶体管 310 的电流镜像到限流装置 322 中；限流装置 322 的漏端接存储单元 320 的源端。根据镜像电路的原理可知流过存储单元 320 的电流与流过第一晶体管 310 的电流成比例，因而采用这样一种结构可以通过控制第一晶体管 310 的电流而有效地控制流过存储单元 320 的电流，可以保护电路和存储单元，同时由于电流是恒定的，所以在软编程的时候，电流更趋向流过存在过擦除现象的单元，使得软编程效率和可靠性都得到提高。

[0036] 以上所述的具体实施例，对本发明的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本发明的具体实施例而已，并不用于限制本发明，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

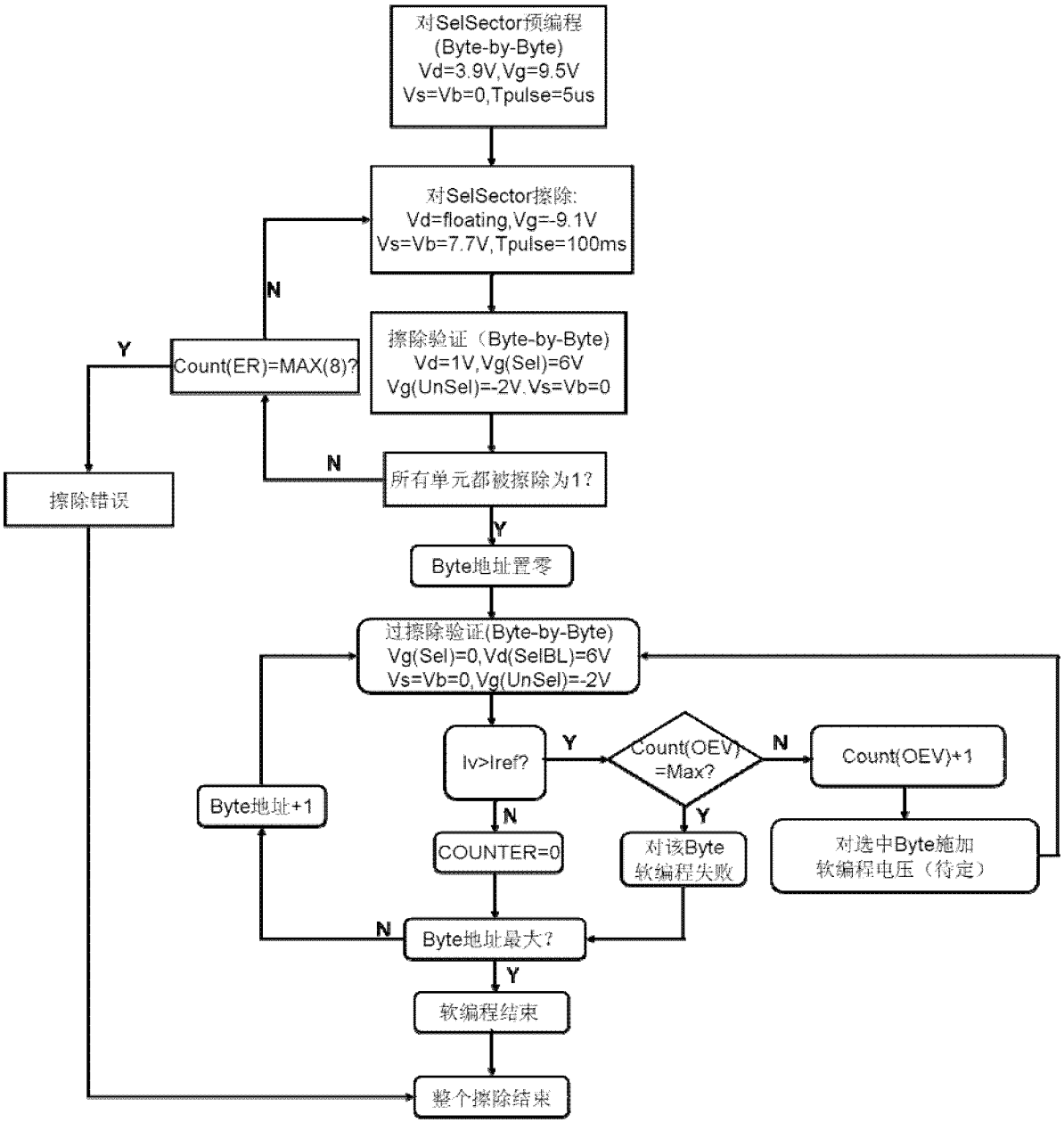

对SelSector预编程
(Byte-by-Byte)
Vd=3.9V,Vg=9.5V
Vs=Vb=0,Tpulse=5us
对SelSector擦除:
Vd=floating,Vg=-9.1V
Vs=Vb=7.7V,Tpulse=100ms
擦除验证（Byte-by-Byte)
Vd=1V,Vg(Sel)=6V
Vg(UnSel)=-2V.Vs=Vb=0
所有单元都被擦除为1?
N
Y
Count(ER)=MAX(8)?
Y
N
擦除错误
Byte地址置零
过擦除验证(Byte-by-Byte)
Vg(Sel)=0,Vd(SelBL)=6V
Vs=Vb=0,Vg(UnSel)=-2V
Iv>Iref?
Y
Count(OEV)
=Max?
N
Count(OEV)+1
对选中Byte施加
软编程电压（待定）
N
Y
Byte地址+1
COUNTER=0
对该Byte
软编程失败
N
Byte地址最大?
Y
软编程结束
整个擦除结束

图 1

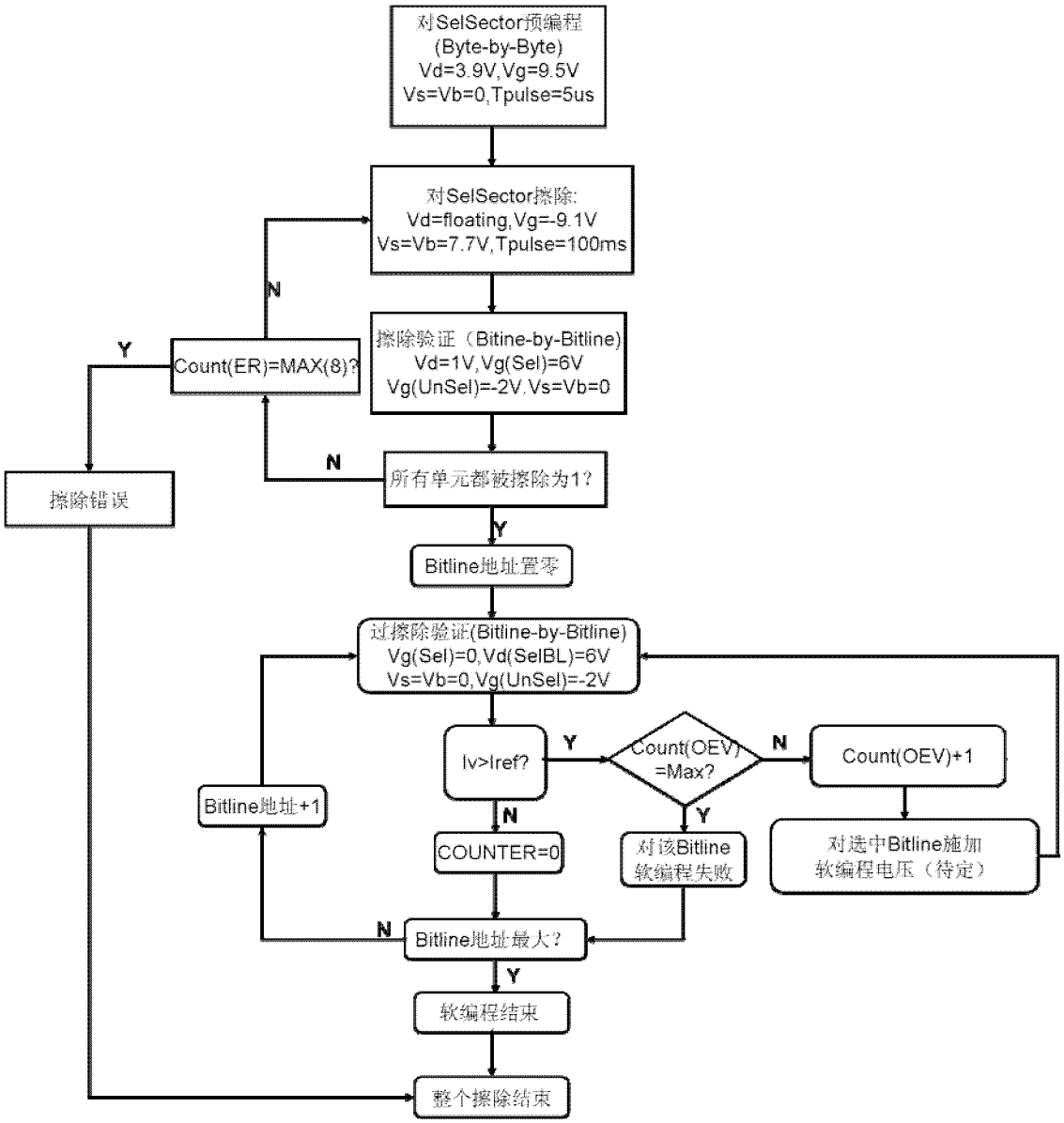

对SelSector预编程
(Byte-by-Byte)
Vd=3.9V,Vg=9.5V
Vs=Vb=0,Tpulse=5us
对SelSector擦除:
Vd=floating,Vg=-9.1V
Vs=Vb=7.7V,Tpulse=100ms
擦除验证（Bitine-by-Bitline)
Vd=1V,Vg(Sel)=6V
Vg(UnSel)=-2V.Vs=Vb=0
所有单元都被擦除为1?
N
Count(ER)=MAX(8)?
Y
擦除错误
Y
Bitline地址置零
过擦除验证(Bitline-by-Bitline)
Vg(Sel)=0,Vd(SelBL)=6V
Vs=Vb=0,Vg(UnSel)=-2V
Iv>Iref?
Y
Count(OEV)
=Max?
N
Count(OEV)+1
对选中Bitline施加
软编程电压（待定）
N
COUNTER=0
Y
对该Bitline
软编程失败
Bitline地址+1
N
Bitline地址最大?
Y
软编程结束
整个擦除结束

图 2

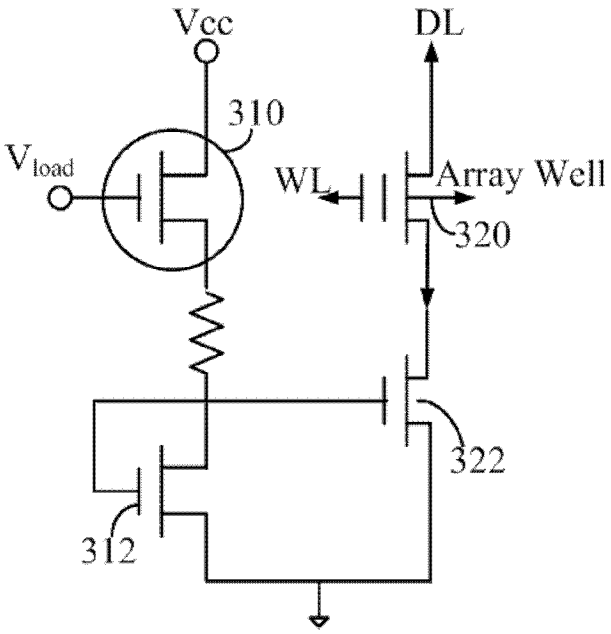


图 3