(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 112151103 A
(43)申请公布日 2020.12.29

(21)申请号 202010982769.3

(22)申请日 2020.09.17

(71)申请人 深圳市宏旺微电子有限公司
地址 518000 广东省深圳市南山区沙河街道华侨城创意文化园开平街2号G2栋2楼

(72)发明人 魏佳辉 刘敏 戴洋洋 陈宗廷 李斌

(74)专利代理机构 深圳市诺正鑫泽知识产权代理有限公司 44689
代理人 林国友

(51)Int.Cl.
G11C 29/18(2006.01)

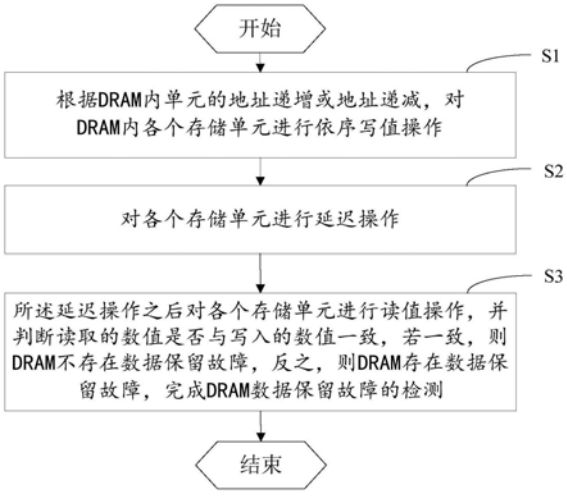
权利要求书2页 说明书6页 附图3页

(54)发明名称

基于March算法的DRAM故障检测方法和装置

(57)摘要

本申请提供了一种基于March算法的DRAM故障检测方法和装置,运用于半导体集成电路测试技术领域,通过不同数据背景对一个存储单元的反复读写,能够检测到原有March算法难以发现的单元内不同位之间的耦合故障,同时能够检测内存单元的读写稳定性,通过在写操作和读操作之间增加一个延迟操作,若芯片内部出现漏电故障BF,则高电压的cell会向低电压的cell漏电,经过一段时间的漏电之后高电压的cell则不能维持原有的数据,故会发生故障,而增加的延迟延时操作可以有效检测到存储单元的数据保留故障DRF。



CN 112151103 A

1.一种基于March算法的DRAM故障检测方法,其特征在于,包括:

S1,根据DRAM内单元的地址递增或地址递减,对DRAM内各个存储单元进行依序写值操作;

S2,对各个存储单元进行延迟操作;

S3,所述延迟操作之后对各个存储单元进行读值操作,并判断读取的数值是否与写入的数值一致,若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障,完成DRAM数据保留故障的检测。

2.根据权利要求1所述的基于March算法的DRAM故障检测方法,其特征在于,所述根据DRAM内单元的地址递增或地址递减,对DRAM内各个存储单元进行依序写值操作的步骤,包括:

根据DRAM内单元的地址递增或地址递减,向各个存储单元依序写入“0”值或者“1”值。

3.根据权利要求2所述的基于March算法的DRAM故障检测方法,其特征在于,所述延迟操作之后对各个存储单元进行读值操作,并判断读取的数值是否与写入的数值一致的步骤,包括:

从各个存储单元读取“0”值或者“1”值;

判断读取的数值是否与写入的数值一致;

若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障。

4.根据权利要求1所述的基于March算法的DRAM故障检测方法,其特征在于,所述完成DRAM数据保留故障的检测的步骤之后,还包括:

S4,根据DRAM内单元的地址递增或地址递减,向DRAM内各个存储单元依序写入16位数值,第一次向DRAM内各个存储单元依序写入所述16位数值包括“0000000000000001”或者“1111111111111110”;

S5,根据DRAM内单元的地址递增或地址递减,读取DRAM内各个存储单元中输入的16位数值,并判断读取数值与写入数值是否一致,若一致,则向DRAM内各个存储单元依序写入将“1”或“0”向左移1位后的16位数值,即第二次的16位数值包括“0000000000000010”或者“1111111111111101”;

S6,重复执行上述步骤S5,直至进行16次的读写过程后,若每次的读写过程中判断数值均无误,则DRAM无故障,若读写过程中出现数值有误,则认定DRAM出现故障。

5.根据权利要求4所述的基于March算法的DRAM故障检测方法,其特征在于,在执行步骤S1至S3的过程中,还包括:

所述写值操作时,依地址顺序写入第一次的16位数值;

进行延迟操作;

所述读值操作时,依地址顺序读取第一次的16位数值,并判断是否与写入时一致,若一致,则循环至写值操作以写入第二次的16位数值,再延迟操作,而后读取第二次的16位数值并判断,依此循环16次。

6.一种基于March算法的DRAM故障检测装置,其特征在于,采用检测装置执行权利要求1-5任一项所述的检测方法,所述检测装置包括:

第一写值单元,用于根据DRAM内单元的地址递增或地址递减,对DRAM内各个存储单元进行依序写值操作;

延迟单元,用于对各个存储单元进行延迟操作;

第一读取单元,用于所述延迟操作之后对各个存储单元进行读值操作,并判断读取的数值是否与写入的数值一致,若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障,完成DRAM数据保留故障的检测。

7.根据权利要求6所述的基于March算法的DRAM故障检测装置,其特征在于,所述第一写值单元包括:

写值模块,用于根据DRAM内单元的地址递增或地址递减,向各个存储单元依序写入“0”值或者“1”值。

8.根据权利要求7所述的基于March算法的DRAM故障检测装置,其特征在于,所述第一读取单元包括:

读值模块,用于从各个存储单元读取“0”值或者“1”值;

判断模块,用于判断读取的数值是否与写入的数值一致,若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障。

9.根据权利要求6所述的基于March算法的DRAM故障检测装置,其特征在于,包括:

第二写值单元,用于根据DRAM内单元的地址递增或地址递减,向DRAM内各个存储单元依序写入16位数值,第一次向DRAM内各个存储单元依序写入所述16位数值包括“0000000000000001”或者“1111111111111110”;

第二读取单元,用于根据DRAM内单元的地址递增或地址递减,读取DRAM内各个存储单元中输入的16位数值,并判断读取数值与写入数值是否一致,若一致,则向DRAM内各个存储单元依序写入将“1”或“0”向左移1位后的16位数值,第二次的16位数值包括“0000000000000010”或者“1111111111111101”;

重复单元,用于重复执行上述第二读取单元所执行的操作,直至进行16次的读写过程后,若每次的读写过程中判断数值均无误,则DRAM无故障,若读写过程中出现数值有误,则认定DRAM出现故障。

基于March算法的DRAM故障检测方法和装置

技术领域

[0001] 本申请涉及半导体集成电路测试技术领域，特别涉及为一种基于March算法的DRAM故障检测方法和装置。

背景技术

[0002] 随着半导体技术的不断发展和集成电路制造水平的不断提高，集成电路芯片的密度越来越大，存储器的容量和速度也在快速增长，存储器发生故障的概率也越来越大，故障种类越来越多，存储器的故障检测也越来越难。存储器故障可分为物理故障和逻辑故障，存储器物理故障指的是存储器在生产制造过程中物理结构发生了改变，逻辑故障是简化的存储器故障模型，所有的物理缺陷都可以映射为存储器故障模型。

[0003] 目前直接检测存储器的物理故障比较困难，现有技术一般检测存储器的逻辑故障以反映物理故障。逻辑故障包括固定故障SAF、转换故障TF、耦合故障CF、寻址故障AF、数据保留故障DRF等。DRAM内存的检测一般是由软件实现的，针对上述存储器故障模型，人们设计出了许多存储器检测算法，不同的算法具有不同的实现方式、复杂度和故障覆盖率。检测算法的复杂度越低、故障覆盖率越高，则算法的检测效率越好，如果一个算法能够用最少的测试向量检测最多的故障，那么这个算法的检测效率越好。

[0004] March算法是最常用的存储器检测算法，也是目前存储器检测算法研究的重点。March算法的基本思路是对所有单元按照地址升序(降序)一次进行一组读/写操作(可以是一次或者多次读/写操作，即一个March过程)，自从March算法诞生以来，经过长期的研究和发展，针对各种存储器故障类型，March算法已经衍生出许多不同的形式，它们的March过程不同，能够检测到的故障种类也不同。原有March C-算法已经能覆盖到如固定故障、转换故障、耦合故障、寻址故障等大多数存储器故障，且该测试算法相对简单，但无法检测到一个单元内的相互耦合产生的故障和数据保留故障DRF。

[0005] 其中，数据保留故障(Data Retention Fault)，是数据在规定时间内不能保留其逻辑值而使存储器发生故障。由于内存单元发生故障导致电容漏电增加，而使内存单元逻辑值经过一些周期由于漏电而改变，即存储单元在规定的时间内不能保存逻辑值，如果不能检测出存储器这种故障，那么使用这种存储器时将会导致数据丢失。

发明内容

[0006] 本申请提供一种基于March算法的DRAM故障检测方法和装置，在原有March算法的基础上，提出的一个新型改进型方法，该方法能大大提高故障覆盖率从而提高存储器测试效率，根据存储器测试程序对待测试的DRAM进行写读操作，写读操作覆盖到所有可操作的存储器地址，针对存储器常见的单元内的耦合故障CF和数据保留故障DRF，用该方法可有效检测出这些故障。

[0007] 本申请为解决技术问题采用如下技术手段：

[0008] 本申请提出一种基于March算法的DRAM故障检测方法，包括：

[0009] S1,根据DRAM内单元的地址递增或地址递减,对DRAM内各个存储单元进行依序写值操作;

[0010] S2,对各个存储单元进行延迟操作;

[0011] S3,所述延迟操作之后对各个存储单元进行读值操作,并判断读取的数值是否与写入的数值一致,若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障,完成DRAM数据保留故障的检测。

[0012] 进一步地,所述根据DRAM内单元的地址递增或地址递减,对DRAM内各个存储单元进行依序写值操作的步骤,包括:

[0013] 根据DRAM内单元的地址递增或地址递减,向各个存储单元依序写入“0”值或者“1”值。

[0014] 进一步地,所述延迟操作之后对各个存储单元进行读值操作,并判断读取的数值是否与写入的数值一致的步骤,包括:

[0015] 从各个存储单元读取“0”值或者“1”值;

[0016] 判断读取的数值是否与写入的数值一致;

[0017] 若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障。

[0018] 进一步地,所述完成DRAM数据保留故障的检测的步骤之后,还包括:

[0019] S4,根据DRAM内单元的地址递增或地址递减,向DRAM内各个存储单元依序写入16位数值,第一次向DRAM内各个存储单元依序写入所述16位数值包括“0000000000000001”或者“1111111111111110”;

[0020] S5,根据DRAM内单元的地址递增或地址递减,读取DRAM内各个存储单元中输入的16位数值,并判断读取数值与写入数值是否一致,若一致,则向DRAM内各个存储单元依序写入将“1”或“0”向左移1位后的16位数值,第二次的16位数值包括“0000000000000010”或者“1111111111111101”;

[0021] S6,重复执行上述步骤S5,直至进行16次的读写过程后,若每次的读写过程中判断数值均无误,则DRAM无故障,若读写过程中出现数值有误,则认定DRAM出现故障。

[0022] 进一步地,在执行步骤S1至S3的过程中,还包括:

[0023] 所述写值操作时,依地址顺序写入第一次的16位数值;

[0024] 进行延迟操作;

[0025] 所述读值操作时,依地址顺序读取第一次的16位数值,并判断是否与写入时一致,若一致,则循环至写值操作以写入第二次的16位数值,再延迟操作,而后读取第二次的16位数值并判断,依此循环16次。

[0026] 本申请提出一种基于March算法的DRAM故障检测装置,采用检测装置执行上述的检测方法,所述检测装置包括:

[0027] 第一写值单元,用于根据DRAM内单元的地址递增或地址递减,对DRAM内各个存储单元进行依序写值操作;

[0028] 延迟单元,用于对各个存储单元进行延迟操作;

[0029] 第一读取单元,用于所述延迟操作之后对各个存储单元进行读值操作,并判断读取的数值是否与写入的数值一致,若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障,完成DRAM数据保留故障的检测。

[0030] 进一步地,所述第一写值单元包括:

[0031] 写值模块,用于根据DRAM内单元的地址递增或地址递减,向各个存储单元依序写入“0”值或者“1”值。

[0032] 进一步地,所述第一读取单元包括:

[0033] 读值模块,用于从各个存储单元读取“0”值或者“1”值;

[0034] 判断模块,用于判断读取的数值是否与写入的数值一致,若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障。

[0035] 进一步地,基于March算法的DRAM故障检测装置包括:

[0036] 第二写值单元,用于根据DRAM内单元的地址递增或地址递减,向DRAM内各个存储单元依序写入16位数值,第一次向DRAM内各个存储单元依序写入所述16位数值包括“0000000000000001”或者“1111111111111110”;

[0037] 第二读取单元,用于根据DRAM内单元的地址递增或地址递减,读取DRAM内各个存储单元中输入的16位数值,并判断读取数值与写入数值是否一致,若一致,则向DRAM内各个存储单元依序写入将“1”或“0”向左移1位后的16位数值,第二次的16位数值包括“0000000000000010”或者“1111111111111101”;

[0038] 重复单元,用于重复执行上述第二读取单元所执行的操作,直至进行16次的读写过程后,若每次的读写过程中判断数值均无误,则DRAM无故障,若读写过程中出现数值有误,则认定DRAM出现故障。

[0039] 本申请提供了基于March算法的DRAM故障检测方法和装置,具有以下有益效果:

[0040] 通过不同数据背景对一个存储单元的反复读写,能够检测到原有March算法难以发现的单元内不同位之间的耦合故障,同时能够检测内存单元的读写稳定性,通过在写操作和读操作之间增加一个延迟操作,若芯片内部出现漏电故障BF,则高电压的cell会向低电压的cell漏电,经过一段时间的漏电之后高电压的cell则不能维持原有的数据,故会发生故障,而增加的延迟延时操作可以有效检测到存储单元的数据保留故障DRF。在顺序读写的过程中,可以覆盖到地址解码故障ADF,而在读写的数据"0"和"1"之间的相互转换中,可以检测到存储单元之间存在的固定故障SAF与转换故障TF。综上,该方法是内存测试领域一个行之有效的测试方法,可覆盖到绝大多数的存储器故障,经过实际测试,该方法可有效适用于正常的量产程序。

附图说明

[0041] 图1为本申请基于March算法的DRAM故障检测方法一个实施例的流程示意图;

[0042] 图2为本申请基于March算法的DRAM故障检测方法另一个实施例的流程示意图;

[0043] 图3为本申请基于March算法的DRAM故障检测方法一个实施例中进行数据保留故障测试的原理示意图;

[0044] 图4为本申请基于March算法的DRAM故障检测方法一个实施例中进行耦合数据测试等的原理示意图。

[0045] 本申请为目的的实现、功能特点及优点将结合实施例,参照附图做进一步说明。

具体实施方式

[0046] 应当理解,此处所描述的具体实施例仅仅用以解释本申请,并不用于限定本申请。

[0047] 下面将结合本申请的实施例中的附图,对本申请的实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请的一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0048] 需要说明的是,本申请的说明书和权利要求书及上述附图中的术语“包括”、“包含”和“具有”以及它们任何变形,意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、系统、产品或设备没有限定于已列出的步骤或单元,而是可选地还包括没有列出的步骤或单元,或可选地还包括对于这些过程、方法、产品或设备固有的其他步骤或单元。在本申请的权利要求书、说明书以及说明书附图中的术语,诸如“第一”和“第二”等之类的关系术语仅仅用来将一个实体/操作/对象与另一个实体/操作/对象区分开来,而不一定要求或者暗示这些实体/操作/对象之间存在任何这种实际的关系或者顺序。

[0049] 在本文中提及“实施例”意味着,结合实施例描述的特定特征、结构或特性可以包含在本申请的至少一个实施例中。在说明书中的各个位置出现该短语并不一定均是指相同的实施例,也不是与其他实施例互斥的独立的或备选的实施例。本领域技术人员显式地和隐式地理解的是,本文所描述的实施例可以与其他实施例相结合。

[0050] 参考附图1,为本申请一实施例中的基于March算法的DRAM故障检测方法的流程示意图;

[0051] 本申请提出的基于March算法的DRAM故障检测方法,包括:

[0052] S1,根据DRAM内单元的地址递增或地址递减,对DRAM内各个存储单元进行依序写值操作;

[0053] S2,对各个存储单元进行延迟操作;

[0054] S3,所述延迟操作之后对各个存储单元进行读值操作,并判断读取的数值是否与写入的数值一致,若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障,完成DRAM数据保留故障的检测。

[0055] 具体的,

[0056] 所述根据DRAM内单元的地址递增或地址递减,对DRAM内各个存储单元进行依序写值操作的步骤,包括:

[0057] 根据DRAM内单元的地址递增或地址递减,向各个存储单元依序写入“0”值或者“1”值。

[0058] 所述延迟操作之后对各个存储单元进行读值操作,并判断读取的数值是否与写入的数值一致的步骤,包括:

[0059] 从各个存储单元读取“0”值或者“1”值;

[0060] 判断读取的数值是否与写入的数值一致;

[0061] 若一致,则DRAM不存在数据保留故障,反之,则DRAM存在数据保留故障。

[0062] 按照DRAM内各单元的地址升序(或降序),从最低地址到最高地址(或从最高地址到最低地址)对每一个地址所对应的内存单元依次进行写"0"操作,直到所有的存储单元的每一位都被写入数据"0"。然后进行“delay(延迟)”操作,增加“delay”延迟操作是为了探测

和诊断DRF故障。之后再次按照地址升序(或降序),从最低地址到最高地址(或从最高地址到最低地址)对每一个地址所对应的内存单元依次进行读"0"操作,直到所有的存储单元都进行了读"0"操作。如果发现有某个单元读出的值与写入的值不同,则认为发生故障。

[0063] 参考附图2,为本申请另一实施例中的基于March算法的DRAM故障检测方法的流程示意图,所述完成DRAM数据保留故障的检测的步骤之后,还包括:

[0064] S4,根据DRAM内单元的地址递增或地址递减,向DRAM内各个存储单元依序写入16位数值,第一次向DRAM内各个存储单元依序写入所述16位数值包括“0000000000000001”或者“1111111111111110”;

[0065] S5,根据DRAM内单元的地址递增或地址递减,读取DRAM内各个存储单元中输入的16位数值,并判断读取数值与写入数值是否一致,若一致,则向DRAM内各个存储单元依序写入将“1”或“0”向左移1位后的16位数值,第二次的16位数值包括“0000000000000010”或者“1111111111111101”;

[0066] S6,重复执行上述步骤S5,直至进行16次的读写过程后,若每次的读写过程中判断数值均无误,则DRAM无故障,若读写过程中出现数值有误,则认定DRAM出现故障。

[0067] 具体的,

[0068] DRAM内各个单元按照地址升序(或降序),从最低地址到最高地址(或从最高地址到最低地址)对每一个地址所对应的内存单元依次进行操作,该操作为,以16bit为一个读写单元,第一次写入数据0000000000000001,接着再读取该数据,第二次将数据中的"1"左移一位,即写入数据0000000000000010,接着再读取该数据,依此类推,第十六次写入数据1000000000000000,接着再读取该数据,该算法每次将"1"左移一位,覆盖到所有的数据背景需要写读16次,该测试对检测到一个存储单元内不同位之间的相互影响非常有效。依此来检测存储器所有单元内每一位的写"1"的能力,克服了原有March算法不能检测到一个存储单元内不同位之间相互影响的困难。如果发现某次读操作读出的值与写入的值不同,则认为发生故障。

[0069] 或者,对一个存储单元写数据1111111111111110,接着再读取该数据,第二次将数据中的"0"左移一位,即写入数据1111111111111101,接着再读取该数据,依此类推,第十六次写入数据0111111111111111,接着再读取该数据。如果发现有某个单元读出的值与写入的值不同,则认为发生故障。

[0070] 在一个实施例中,在执行上述步骤S1至S3的过程中,还包括:

[0071] 所述写值操作时,依地址顺序写入第一次的16位数值;

[0072] 进行延迟操作;

[0073] 所述读值操作时,依地址顺序读取第一次的16位数值,并判断是否与写入时一致,若一致,则循环至写值操作以写入第二次的16位数值,再延迟操作,而后读取第二次的16位数值并判断,依此循环16次。

[0074] 通过上述手段使数据保留故障和耦合故障等其它故障进行同时测试,从而将上述的步骤S1～S6压缩成三个步骤即可。

[0075] 本申请还提出一种基于March算法的DRAM故障检测装置,采用检测装置执行上述的检测方法,所述检测装置包括:

[0076] 第一写值单元,用于根据DRAM内单元的地址递增或地址递减,对DRAM内各个存储

单元进行依序写值操作；

[0077] 延迟单元，用于对各个存储单元进行延迟操作；

[0078] 第一读取单元，用于所述延迟操作之后对各个存储单元进行读值操作，并判断读取的数值是否与写入的数值一致，若一致，则DRAM不存在数据保留故障，反之，则DRAM存在数据保留故障，完成DRAM数据保留故障的检测。

[0079] 在一个实施例中，所述第一写值单元包括：

[0080] 写值模块，用于根据DRAM内单元的地址递增或地址递减，向各个存储单元依序写入“0”值或者“1”值。

[0081] 在一个实施例中，所述第一读取单元包括：

[0082] 读值模块，用于从各个存储单元读取“0”值或者“1”值；

[0083] 判断模块，用于判断读取的数值是否与写入的数值一致，若一致，则DRAM不存在数据保留故障，反之，则DRAM存在数据保留故障。

[0084] 在一个实施例中，基于March算法的DRAM故障检测装置，包括：

[0085] 第二写值单元，用于根据DRAM内单元的地址递增或地址递减，向DRAM内各个存储单元依序写入16位数值，第一次向DRAM内各个存储单元依序写入所述16位数值包括“0000000000000001”或者“1111111111111110”；

[0086] 第二读取单元，用于根据DRAM内单元的地址递增或地址递减，读取DRAM内各个存储单元中输入的16位数值，并判断读取数值与写入数值是否一致，若一致，则向DRAM内各个存储单元依序写入将“1”或“0”向左移1位后的16位数值，第二次的16位数值包括“0000000000000010”或者“1111111111111101”；

[0087] 重复单元，用于重复执行上述第二读取单元所执行的操作，直至进行16次的读写过程后，若每次的读写过程中判断数值均无误，则DRAM无故障，若读写过程中出现数值有误，则认定DRAM出现故障。

[0088] 尽管已经示出和描述了本申请的实施例，对于本领域的普通技术人员而言，可以理解在不脱离本申请的原理和精神的情况下可以对这些实施例进行多种变化、修改、替换和变型，本申请的范围由所附权利要求及其等同物限定。

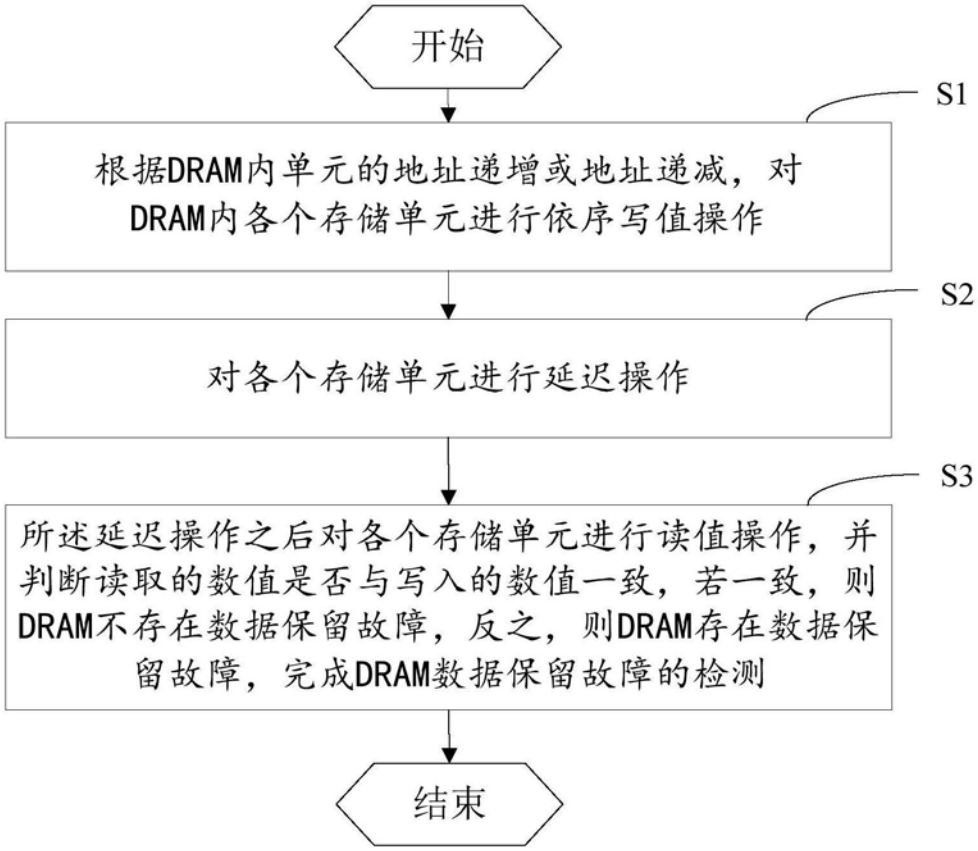

开始
S1
根据DRAM内单元的地址递增或地址递减，对DRAM内各个存储单元进行依序写值操作
S2
对各个存储单元进行延迟操作
S3
所述延迟操作之后对各个存储单元进行读值操作，并判断读取的数值是否与写入的数值一致，若一致，则DRAM不存在数据保留故障，反之，则DRAM存在数据保留故障，完成DRAM数据保留故障的检测
结束

图1

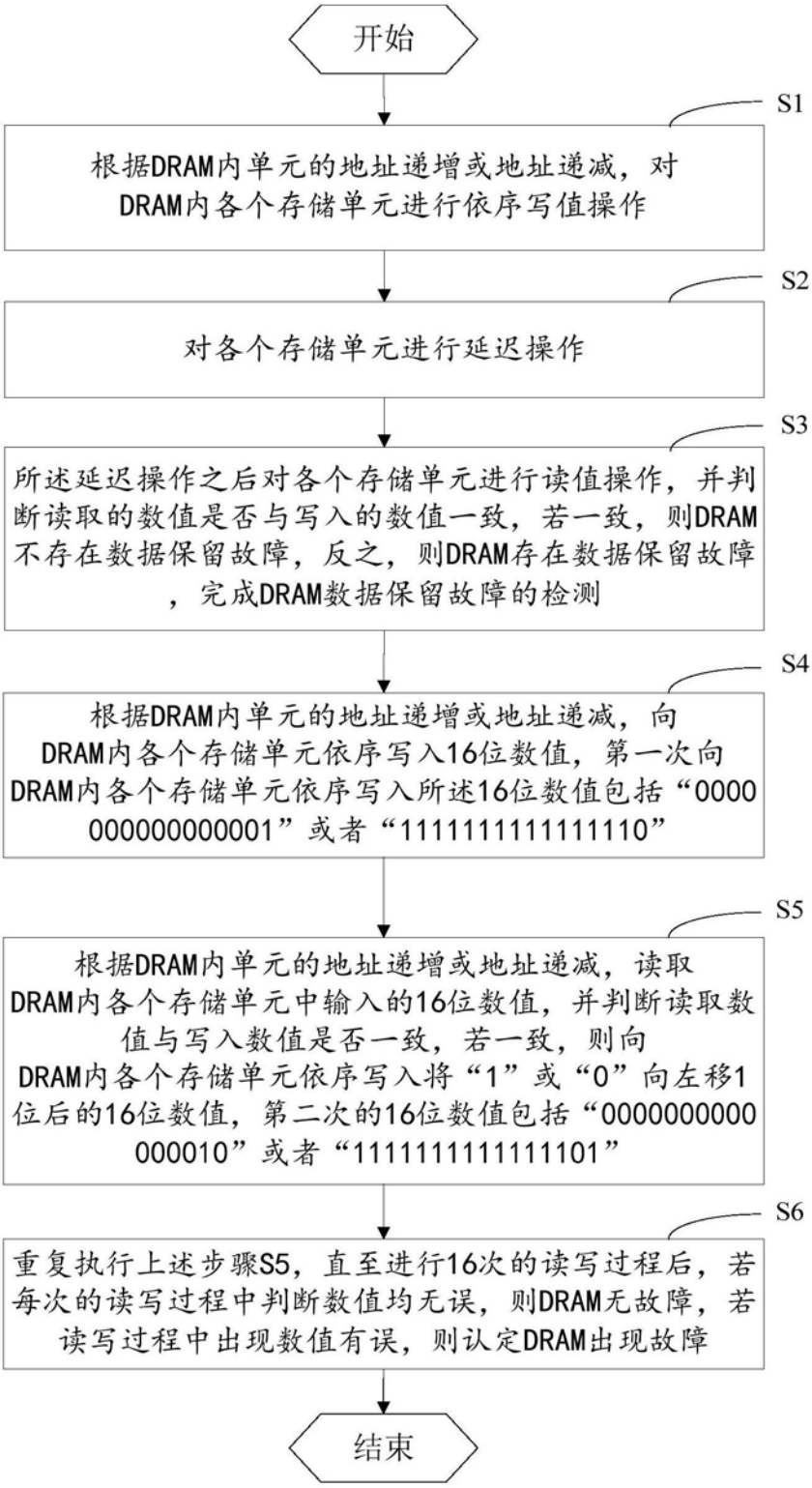

开始
S1
根据DRAM内单元的地址递增或地址递减，对DRAM内各个存储单元进行依序写值操作
S2
对各个存储单元进行延迟操作
S3
所述延迟操作之后对各个存储单元进行读值操作，并判断读取的数值是否与写入的数值一致，若一致，则DRAM不存在数据保留故障，反之，则DRAM存在数据保留故障，完成DRAM数据保留故障的检测
S4
根据DRAM内单元的地址递增或地址递减，向DRAM内各个存储单元依序写入16位数值，第一次向DRAM内各个存储单元依序写入所述16位数值包括“0000000000000001”或者“1111111111111110”
S5
根据DRAM内单元的地址递增或地址递减，读取DRAM内各个存储单元中输入的16位数值，并判断读取数值与写入数值是否一致，若一致，则向DRAM内各个存储单元依序写入将“1”或“0”向左移1位后的16位数值，第二次的16位数值包括“0000000000000010”或者“1111111111111101”
S6
重复执行上述步骤S5，直至进行16次的读写过程后，若每次的读写过程中判断数值均无误，则DRAM无故障，若读写过程中出现数值有误，则认定DRAM出现故障
结束

图2

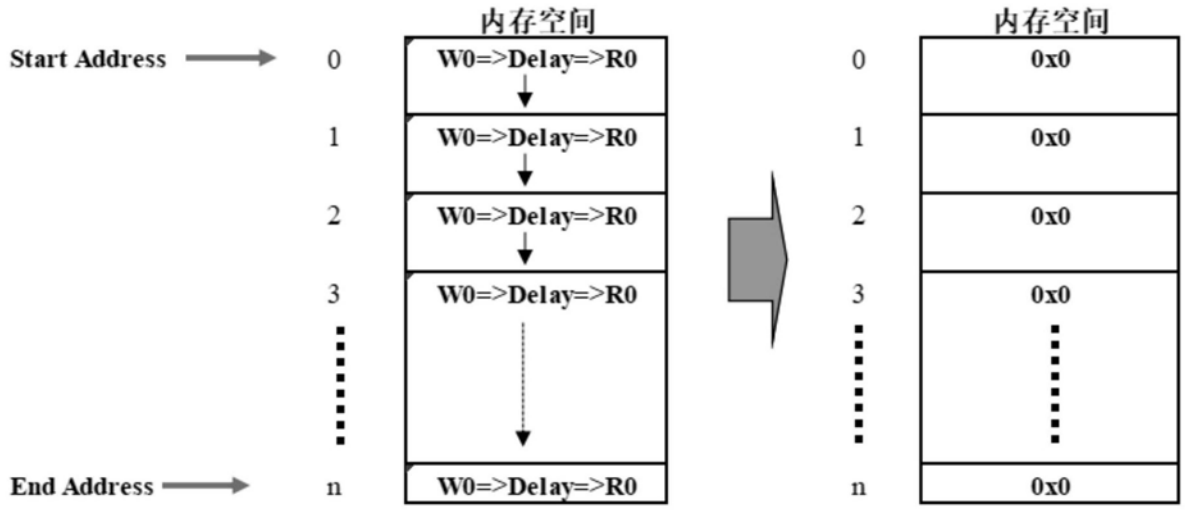

内存空间
Start Address
0
W0=>Delay=>R0
1
W0=>Delay=>R0
2
W0=>Delay=>R0
3
W0=>Delay=>R0
End Address
n
W0=>Delay=>R0
内存空间
0
0x0
1
0x0
2
0x0
3
0x0
n
0x0

图3

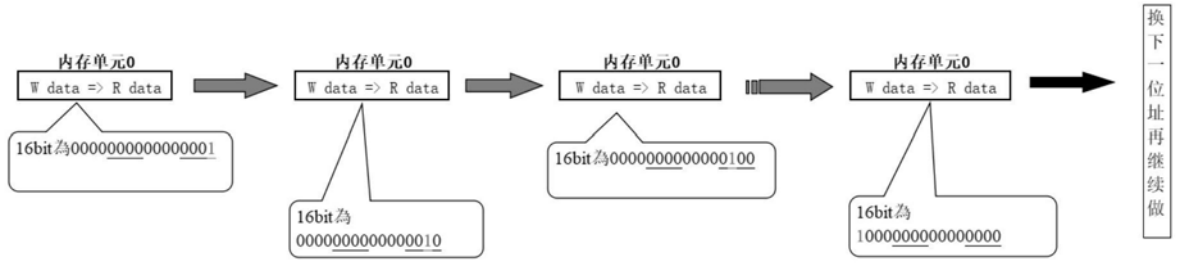

内存单元0
W data => R data
16bit為0000000000000001
内存单元0
W data => R data
16bit為
0000000000000010
内存单元0
W data => R data
16bit為0000000000000100
内存单元0
W data => R data
16bit為
1000000000000000
换下一位址再继续做

图4