



(12) 发明专利

(10) 授权公告号 CN 112331253 B

(45) 授权公告日 2023. 12. 08

(21) 申请号 202011188415.8

(22) 申请日 2020.10.30

(65) 同一申请的已公布的文献号
申请公布号 CN 112331253 A

(43) 申请公布日 2021.02.05

(73) 专利权人 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 冯修圣 陈霖 刘敏 戴洋洋
陈宗廷 李斌

(74) 专利代理机构 深圳中一联合知识产权代理
有限公司 44414
专利代理师 唐佳芝

(51) Int. Cl.
G11C 29/10 (2006.01)
G11C 29/12 (2006.01)
G11C 29/44 (2006.01)

(56) 对比文件

- CN 101567221 A, 2009.10.28
- CN 102263818 A, 2011.11.30
- CN 104281545 A, 2015.01.14
- CN 107845406 A, 2018.03.27
- CN 110648715 A, 2020.01.03
- CN 1679118 A, 2005.10.05
- CN 203311409 U, 2013.11.27
- US 5818772 A, 1998.10.06
- US 7797594 B1, 2010.09.14
- CN 111078459 A, 2020.04.28
- CN 102394111 A, 2012.03.28
- CN 105573881 A, 2016.05.11
- CN 108899061 A, 2018.11.27
- CN 111554344 A, 2020.08.18
- S.A. McKee等. Dynamic access ordering for streamed computations.《IEEE》.2000, 1255-1271.
- 高剑 等. FLASH存储器的测试方法.《电子测量技术》.2008, (第7期), 117-120.

审查员 郑艳梅

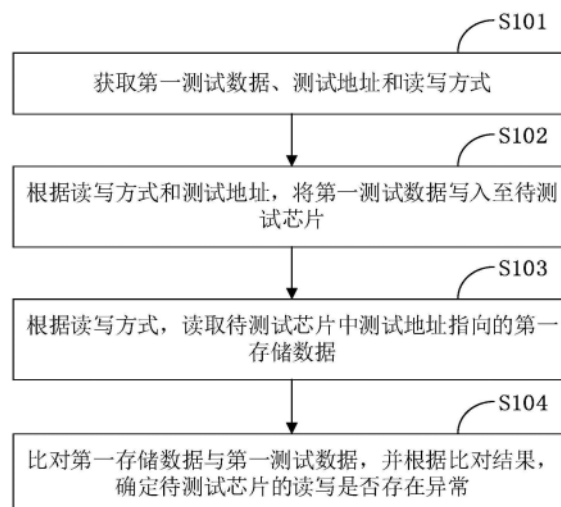
权利要求书2页 说明书13页 附图5页

(54) 发明名称

一种芯片的测试方法、终端和存储介质

(57) 摘要

本申请适用于计算机领域,提供了一种芯片的检测方法、终端和存储介质。其中,该芯片的检测方法包括:获取第一测试数据、测试地址和读写方式,所述测试地址为待测试芯片的数据存储地址;根据所述读写方式和所述测试地址,将所述第一测试数据写入至所述待测试芯片;根据所述读写方式,读取所述待测试芯片中所述测试地址指向的第一存储数据;比对所述第一存储数据与所述第一测试数据,并根据比对结果,确定所述待测试芯片的读写是否存在异常。本申请的实施例能够提高芯片测试的可靠性。



1. 一种芯片的测试方法,其特征在于,所述方法包括:
获取第一测试数据、测试地址和读写方式,所述测试地址为待测试芯片的数据存储地址;
根据所述读写方式和所述测试地址,将所述第一测试数据写入至所述待测试芯片;
根据所述读写方式,读取所述待测试芯片中所述测试地址指向的第一存储数据;
比对所述第一存储数据与所述第一测试数据,并根据比对结果,确定所述待测试芯片的读写是否存在异常;
所述读写方式的数量大于1,所述待测试芯片的数据存储地址包含:存储块地址、行地址和列地址;每种所述读写方式中,均包含一种对存储块地址、行地址和列地址的读写顺序,且每种读写方式包含的读写顺序及对应的命令时序均不相同;
所述根据所述读写方式和所述测试地址,将所述第一测试数据写入至所述待测试芯片,包括:根据各个所述读写方式和所述测试地址,将所述第一测试数据多次写入至所述待测试芯片,其中,每次写入所述待测试芯片所采用的所述读写方式均不相同。
2. 如权利要求1所述的芯片的测试方法,其特征在于,所述获取第一测试数据,包括:
获取所述待测试芯片的存储总容量;
根据所述存储总容量,确定多个测试数据体积;
获取多个第一测试数据,其中所述第一测试数据的体积,与所述测试数据体积一一对应。
3. 如权利要求2所述的芯片的测试方法,其特征在于,所述根据所述读写方式和所述测试地址,将所述第一测试数据写入至所述待测试芯片,包括:
从多个所述第一测试数据中筛选出一个第二测试数据;
获取测试首地址;
根据所述第二测试数据的体积和所述测试首地址,确定测试末地址;
根据所述读写方式、所述测试首地址和所述测试末地址,将所述第二测试数据写入至所述待测试芯片;
相应的,所述芯片的测试方法还包括:
若所述芯片的读写不存在异常,则返回执行所述从多个所述第一测试数据中筛选出一个第二测试数据的操作,直至所有所述第一测试数据,均被写入至所述待测试芯片。
4. 如权利要求3所述的芯片的测试方法,其特征在于,所述根据所述第二测试数据的数据大小和所述测试首地址,确定测试末地址,包括:
获取所述待测试芯片中每个存储单元的存储容量,以及所述待测试芯片的存储块数量、行数量和列数量;
根据所述第二测试数据的体积、所述测试首地址和所述存储容量,确定测试末地址。
5. 如权利要求1所述的芯片的测试方法,其特征在于,根据单个所述读写方式和所述测试地址,将所述第一测试数据多次写入至所述待测试芯片,包括:
根据该读写方式包含的读写顺序,向所述待测试芯片中的所述测试地址写入所述第一测试数据。
6. 如权利要求1所述的芯片的测试方法,其特征在于,所述获取测试地址,包括:
获取预设的第一随机算法,并根据所述第一随机算法生成所述测试地址。

7. 如权利要求1所述的芯片的测试方法,其特征在于,所述获取第一测试数据,包括:
获取所述待测试芯片中每个存储单元的存储容量;
获取预设的第二随机算法;
根据所述存储容量和所述第二随机算法生成多个随机数据,每个所述随机数据的大小等于所述存储容量;
根据所述多个随机数据,确定所述第一测试数据。
8. 如权利要求1所述的芯片的测试方法,其特征在于,所述测试地址包含读测试地址和写测试地址;
所述根据所述读写方式和所述测试地址,将所述第一测试数据写入至所述待测试芯片,包括:
根据所述读写方式和所述写测试地址,将所述第一测试数据写入至所述待测试芯片;
在将所述第一测试数据写入至所述待测试芯片的过程中,根据所述读写方式,读取所述待测试芯片中所述读测试地址指向的第二存储数据;
所述根据所述读写方式,读取所述待测试芯片中所述测试地址指向的第一存储数据,包括:
根据所述读写方式,读取所述待测试芯片中所述写测试地址指向的第一存储数据。
9. 一种终端,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1至8任一项所述方法的步骤。
10. 一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至8任一项所述方法的步骤。

一种芯片的测试方法、终端和存储介质

技术领域

[0001] 本申请属于计算机领域,尤其涉及一种芯片的测试方法、终端和存储介质。

背景技术

[0002] 存储器是用来存储程序和各种数据信息的记忆部件。而对存储芯片的测试算法是存储器测试的核心,现有的主要测试方式是March算法,该算法能够检测多种故障类型,如固定故障(Stuck-at faults,SAF)、转换故障(Transition faults,TF)、耦合故障(Coupling faults,CF)和寻址故障(Address decoder faults,AF)。

[0003] 但是,March算法的可靠性较低。

发明内容

[0004] 本申请实施例提供一种芯片的测试方法、终端和存储介质,可以解决目前芯片测试方法的可靠性较低的问题。

[0005] 本申请实施例第一方面提供一种芯片的检测方法,包括:

[0006] 获取第一测试数据、测试地址和读写方式,所述测试地址为待测试芯片的数据存储地址;

[0007] 根据所述读写方式和所述测试地址,将所述第一测试数据写入至所述待测试芯片;

[0008] 根据所述读写方式,读取所述待测试芯片中所述测试地址指向的第一存储数据;

[0009] 比对所述第一存储数据与所述第一测试数据,并根据比对结果,确定所述待测试芯片的读写是否存在异常。

[0010] 本申请实施例第二方面提供的一种芯片的检测装置,包括:

[0011] 获取单元,用于获取第一测试数据、测试地址和读写方式,所述测试地址为待测试芯片的数据存储地址;

[0012] 写入单元,用于根据所述读写方式和所述测试地址,将所述第一测试数据写入至所述待测试芯片;

[0013] 读取单元,用于根据所述读写方式,读取所述待测试芯片中所述测试地址指向的第一存储数据;

[0014] 测试单元,用于比对所述第一存储数据与所述第一测试数据,并根据比对结果,确定所述待测试芯片的读写是否存在异常。

[0015] 本申请实施例第三方面提供一种终端,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述方法的步骤。

[0016] 本申请实施例第四方面提供一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现上述方法的步骤。

[0017] 本申请实施例第五方面提供了一种计算机程序产品,当计算机程序产品在终端上

运行时,使得终端执行时实现方法的步骤。

[0018] 本申请的实施例,首先,获取待测试芯片和第一测试数据、测试地址和读写方式。然后,根据读写方式,将第一测试数据写入到测试地址中。接着,根据读写方式,读取测试地址中存储的第一存储数据。比对第一存储数据与第一测试数据,并根据比对结果,确定芯片的读写是否存在异常。本申请的实施例考虑了测试数据、测试地址和读写方式三个不同的维度对芯片进行读写能力的测试。相较于March算法,能够测试出实际应用过程中进行读写存在的故障。因此,本申请的实施例能够检测出March算法无法检测出的故障,提高了芯片测试的可靠性。

附图说明

[0019] 为了更清楚地说明本申请实施例中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0020] 图1是本申请实施例提供的一种芯片的检测方法的实现流程示意图;

[0021] 图2是本申请实施例提供的一种获取第一测试数据的第一实现流程示意图;

[0022] 图3是本申请实施例提供的一种使用不同容量的数据进行测试的实现流程示意图;

[0023] 图4是本申请实施例提供的一种获取第一测试数据的第二实现流程示意图;

[0024] 图5是本申请实施例提供的一种并行读写的实现流程示意图;

[0025] 图6是本申请实施例提供的一种芯片测试的流程示意图;

[0026] 图7是本申请实施例提供的一种芯片的检测装置的结构示意图;

[0027] 图8是本申请实施例提供的终端的结构示意图。

具体实施方式

[0028] 为了使本申请的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本申请进行进一步详细说明。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。基于本申请的实施例,本领域技术人员在没有做出创造性劳动的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。

[0029] 存储器是用来存储程序和各种数据信息的记忆部件。而对存储芯片的测试算法是存储器测试的核心,现有的主要测试方式是March算法。该算法的基本原理是利用有限状态机,对所有地址逐个进行读写操作,算法指令较为简单只有,读写0、读写1和变化的指令。一般地,该算法是先对每个存储单元写“0”,然后对每个内存单元读“0”并写“1”,最后读“1”,判断存储单元中存储的是否均为“1”。接着可以对每个内存单元读“1”并写“0”,最后读“0”,判断存储单元中存储的是否均为“0”。

[0030] March算法能够检测多种故障类型,如固定故障(Stuck-at faults,SAF)、转换故障(Transition faults,TF)、耦合故障(Coupling faults,CF)和寻址故障(Address decoder faults,AF)。

[0031] 其中,固定故障是指一个存储单元的值固定在0或固定在1,不会被改变。转换故障

是指在存储器阵列中一个储存单元不能进行0转换为1或1转换为0的转换。耦合故障是指存储单元与存储单元之间的短路和耦合,造成对一个存储单元进行改变必然会引起另一个存储单元的状态改变。寻址故障是指不能正确的找到对应的地址。这些故障均是最常见的故障类型。

[0032] 但是, March算法的可靠性较低。因此, 本申请提供了一种芯片的测试方法, 可以检测March算法难以检测到的故障类型, 提高芯片检测的可靠性。

[0033] 为了说明本申请的技术方案, 下面通过具体实施例来进行说明。

[0034] 图1示出了本申请实施例提供的一种芯片的测试方法的实现流程示意图, 该方法可以应用于终端, 可适用于需提高芯片检测的可靠性的情形。

[0035] 具体的, 上述芯片的检测方法可以包括以下步骤S101至步骤S104。

[0036] 步骤S101, 获取第一测试数据、测试地址和读写方式。

[0037] 其中, 上述测试地址为待测试芯片的数据存储地址。其中, 待测试芯片是指需要进行测试的存储芯片。

[0038] 一般地, 在存储器出厂前, 都会将存储器提供给测试人员进行测试, 以保证流向市场的存储器不易出现故障, 保障消费者的权益。在另一些场景中, 存储器的使用者为了了解或确认存储器的芯片工作是否正常, 也可以将存储器提供给测试人员进行测试。因此, 测试人员对存储器进行芯片测试时, 所使用的存储器内的存储芯片, 即为本申请的待测试芯片。

[0039] 在本申请的实施方式中, 在进行测试之前, 上述终端可以和待测试芯片建立连接, 以使上述终端可以对待测试芯片进行读写操作, 完成对待测试芯片的测试。其中, 建立连接的方式可以由测试人员根据实际情况进行选择。例如, 可以由管理员将携带有待测试芯片的存储器插入到上述终端的印制电路板(Printed Circuit Board, PCB)上。

[0040] 上述第一测试数据是指在测试过程中, 写入到待测试芯片的存储单元中的数据。该数据可以直接由具有丰富测试经验的测试人员提供, 也可以是通过一定的随机算法计算出的测试数据。

[0041] 上述测试地址是指在测试过程中, 需写入第一测试数据的存储单元对应的地址。同样的, 该地址可以直接由具有丰富测试经验的测试人员提供, 也可以是通过一定的随机算法计算出的测试数据。

[0042] 上述读写方式是指将第一测试数据写入到测试地址对应的存储单元所使用的写入方式, 或者从测试地址对应的存储单元中读取数据时所使用的读取方式。

[0043] 在本申请的实施方式中, 上述终端会考虑第一测试数据、测试地址和读写方式三个方面, 对待测试芯片进行测试, 以检测测试数据、测试地址或读写方式的不同带来的故障类型。

[0044] 步骤S102, 根据读写方式和测试地址, 将第一测试数据写入至待测试芯片。

[0045] 在本申请的实施方式中, 在获取到第一测试数据、测试地址和读写方式之后, 上述终端可以生成多条用于指示终端进行写入操作的命令, 然后执行生成的命令, 以根据读写方式, 将第一测试数据写入到测试地址指向的存储单元中。

[0046] 步骤S103, 根据读写方式, 读取待测试芯片中测试地址指向的第一存储数据。

[0047] 其中, 第一存储数据是指从测试地址指向的存储单元中读出的数据。

[0048] 在本申请的实施方式中, 正常情况下, 在将第一测试数据写入到测试地址之后, 测

试地址对应的存储单元中存储的第一存储数据会和第一测试数据相同。因此,可以根据读写方式,读取测试地址中存储的第一存储数据,以判断芯片的读写是否存在异常。

[0049] 同样的,上述终端可以根据测试地址和读写方式,生成多条用于指示终端进行读取数据操作的命令,然后执行生成的命令,以根据读写方式,读取测试地址指向的存储单元中存储的第一存储数据。

[0050] 步骤S104,比对第一存储数据与第一测试数据,并根据比对结果,确定待测试芯片的读写是否存在异常。

[0051] 在本申请的实施方式中,在读取第一存储数据之后,可以将第一存储数据和第一测试数据进行比对。若第一存储数据和第一测试数据不相同,则说明在存储芯片在读写过程中存在一定的异常,并且异常可能是由于读写方式、第一测试数据、测试地址之中的至少一项造成的。

[0052] 若第一存储数据和第一测试数据相同,则说明存储芯片在读写过程中不存在异常。因此,可以至少确认使用该读写方式在对测试地址指向的存储单元进行第一测试数据的读写时,存储芯片的表现是正常的。

[0053] 在本申请的一些实施方式中,会获取多个第一测试数据、多个测试地址和多种读写方式,并根据不同的读写方式,以不同的第一测试数据对不同的测试地址进行读写测试,进而保证芯片测试的可靠性。

[0054] 本申请的实施例,首先,获取待测试芯片和第一测试数据、测试地址和读写方式。然后,根据读写方式,将第一测试数据写入到测试地址中。接着,根据读写方式,读取测试地址中存储的第一存储数据。比对第一存储数据与第一测试数据,并根据比对结果,确定芯片的读写是否存在异常。本申请的实施例考虑了测试数据、测试地址和读写方式三个不同的维度对芯片进行读写能力的测试。相较于March算法,能够测试出实际应用过程中进行读写存在的故障。因此,本申请的实施例能够检测出March算法无法检测出的故障,例如读写方式上的故障、特殊测试数据的故障等,能够提高芯片测试的可靠性。

[0055] 考虑到实际应用中,March算法是对所有存储单元进行读写,但芯片在实际使用过程中可能只用到部分存储单元,因此March算法检测的只是针对所有存储单元在使用时的状态,并不能完全模拟芯片在实际使用过程中的状态。

[0056] 基于此,在本申请的一些实施方式中,可以依据芯片容量确定多个不同大小的第一测试数据,并依次进行测试。

[0057] 具体的,在本申请的一些实施方式中,如图2所示,上述获取第一测试数据,可以包括:步骤S201至步骤S203。

[0058] 步骤S201,获取待测试芯片的存储总容量。

[0059] 其中,上述存储总容量是指待测试芯片的总存储容量,即待测试芯片能够存储的最大数据量。

[0060] 在本申请的一些实施方式中,上述终端可以根据待测试芯片的规格确定存储总容量。

[0061] 具体的,在本申请的一些实施方式中,上述终端在与待测试芯片建立连接之后,可以读取芯片的规格。实际应用中,芯片在完成生产后都会张贴有携带芯片信息的标识码或者说明条,因此,在本申请的另一一些实施方式中,上述终端也可以通过读取标识码或者采集

说明条的图像,获取芯片的规格。

[0062] 在获取到芯片的规格之后,上述终端可以根据固态技术协会(JEDEC)提供的标准确定存储总容量,也可以根据供应待测试芯片的厂商提供的规格书确定存储总容量。

[0063] 其中,固态技术协会是微电子产业的领导标准机构。一般地,在固态技术协会标准或厂商规格书中,一个规格会对应有一个存储总容量。因此,上述终端可以根据芯片的规格,确定出存储总容量。

[0064] 步骤S202,根据存储总容量,确定多个测试数据体积。

[0065] 其中,上述测试数据体积是指第一测试数据的数据量大小。

[0066] 在本申请的一些实施方式中,可以将存储总容量乘上多个预设的比例,得到多个测试数据体积。例如将存储总容量的1/4、1/2、3/4分别确定为一个测试数据体积。

[0067] 在本申请的另一一些实施方式中,可以预先确定出多个预设容量,然后根据存储总容量,从多个预设容量中筛选出小于或等于存储总容量的预设容量,并将每个筛选出的预设容量确定为一个测试数据体积。

[0068] 步骤S203,获取多个第一测试数据,其中第一测试数据的体积,与测试数据体积一一对应。

[0069] 在本申请的实施例中,可以根据测试数据体积,确定出多个第一测试数据,并且,每个第一测试数据的体积(即数据量大小)分别和一个测试数据体积相同。该第一测试数据的具体数据内容可以直接由具有丰富测试经验的测试人员根据测试数据体积提供,也可以是通过一定的随机算法计算出的测试数据。还可以是对一大容量的测试数据依据测试数据体积进行截取,得到的第一测试数据。

[0070] 例如,一个待测试芯片的存储总容量为8Gb,则根据存储总容量,可以将存储总容量的1/4、1/2、3/4分别确定为一个测试数据体积,因此,可以分别获取一个2Gb、一个4Gb和一个6Gb的第一测试数据。在此基础上,还可以获取预先确定的预设容量,该预设容量可以为较小的容量如4MB的容量。因此,最后可以得到一个4MB的第一测试数据、一个2Gb的第一测试数据、一个4Gb的第一测试数据和一个6Gb的第一测试数据。此时可以利用这些第一测试数据对芯片进行测试。

[0071] 具体的,如图3所示,上述对芯片进行测试,可以包括以下步骤S301至步骤S307。

[0072] 步骤S301,从多个第一测试数据中筛选出一个第二测试数据。

[0073] 其中,第二测试数据为当前需要写入芯片的测试数据。

[0074] 在本申请的一些实施方式中,不对从第一测试数据中筛选出一个第二测试数据的具体方式进行限制。例如可以从第一测试数据中筛选出任一数据作为第二测试数据,也可以将上述多个第一测试数据依据容量大小由小到大的顺序依次确认为第二测试数据。

[0075] 步骤S302,获取测试首地址。

[0076] 其中,上述测试首地址是指写入第二测试数据的起始位置。该测试首地址可以根据实际需要进行选择。例如可以将整个芯片的首地址作为测试首地址,也可以获取管理员预设的一个测试首地址。

[0077] 步骤S303,根据第二测试数据的体积和测试首地址,确定测试末地址。

[0078] 其中,测试末地址是指完成写入第二测试数据的结束位置。在获取到第二测试数据的数据大小和测试首地址之后,根据测试首地址,可以测算出在进行对第二测试数据的

写操作后,完成写入的地址。

[0079] 具体的,在本申请的一些实施方式中,可以获取待测试芯片中每个存储单元的存储容量,以及待测试芯片的存储块(Bank)数量、行数量和列数量,并根据第二测试数据的体积、测试首地址和存储容量,确定测试末地址。

[0080] 一般地,待测试芯片可以划分出多个存储块,每个存储块包含若干数量的行和若干数量的列。在每一个存储块中,每一行的一列对应一个存储单元,每个存储单元的存储容量一般是相同的。

[0081] 其中,上述存储单元的存储容量、存储块数量、行数量和列数量一般也都是在出厂时已经确定好的,对应的获取方式可以参考步骤S201中芯片规格的获取方式。

[0082] 例如,当前需要对一4MB的第二测试数据进行测试。若待测试芯片的行地址线为15,列地址线为10,存储块地址线为3,即行数量为 2^{15} 条,列数量为 2^{10} 条,存储块数量为 2^3 个。每个存储单元的容量为32bit。如果以列作为限制进行计算,4MB的第二测试数据需要 $4\text{MB}=4096\text{K}=4194304\text{Byte}=33554432\text{bit}$, $33554432\text{bit}\div 32\text{bit}\div 1024\text{列}=1024\text{行}$ 。也就是说,4MB的第二测试数据需要1024行x1024列个存储单元,若首地址为存储块0、0行、0列,则末地址为存储块0、1023行、1023列。

[0083] 步骤S304,根据读写方式、测试首地址和测试末地址,将第二测试数据写入至待测试芯片。

[0084] 在本申请的一些实施方式中,在获取到读写方式、测试首地址和测试末地址之后,上述终端可以根据读写方式、测试首地址和测试末地址生成对应的命令行,并由终端执行命令行,以将第二测试数据完整地由测试首地址指向的存储单元依据读写方式写入到测试末地址指向的存储单元。

[0085] 步骤S305,根据读写方式,读取待测试芯片中测试地址指向的第一存储数据。

[0086] 步骤S306,比对第一存储数据与第二测试数据,并根据比对结果,确定芯片的读写是否存在异常。

[0087] 其中,上述步骤S305和步骤S306可以参看前述步骤S103和步骤S104的描述,本申请对此不进行赘述。

[0088] 步骤S307,若芯片的读写不存在异常,则返回执行从多个第一测试数据中筛选出一个第二测试数据的操作,直至所有第一测试数据,均被写入至待测试芯片。

[0089] 其中,筛选出的第二测试数据应为多个第一测试数据中,未进行测试的数据。

[0090] 需要说明的是,在筛选出新的第二测试数据后,测试首地址同样可以根据实际情况调整。也就是说,对每个第二测试数据进行测试时,对应的测试首地址可以不相同。例如,可以为每个第二测试数据都关联一个预设的测试首地址。也可以是每次测试都从整个存储芯片的首地址开始。或者,也可以是将每次测试的首地址确认为上一次测试的测试末地址的下一个地址。

[0091] 本申请的实施例,通过多次循环测试,每次测试所使用的数据对应的数据量大小等于存储总容量的部分或全部,实际模拟了芯片在使用过程中不断由部分启用至全部启用的一个过程,更加贴合芯片在实际使用过程中的实际情况。相比March算法直接对全部存储单元进行写入与读取,本申请提供的测试方法可以测试出在进行部分读写的过程中可能出现的故障。

[0092] 在本申请的一些实施方式中,上述待测试芯片的存储地址包含:存储块地址、行地址和列地址。一般的测试方式,如March算法都是只适用一种读写方式进行写入与读取的,即常规的写入方式是选定一个存储块,再选定一行,把这一行所有的列全部写完,再进行下一列的写入。

[0093] 但是实际应用过程中,由于芯片的读写一般都是通过执行命令完成的,而不同的读写方式,对应的命令时序并不相同。例如在选定一个存储块之前,需要执行一个Active命令激活存储块。在切换行之前,会有一个precharge的命令,用于关闭当前行,打开新的行。因此不同的写入方式会使命令的时序也发生变化。而实际应用中,各种读写方式都可以发生,因此针对这种情况,本申请的实施例中会增加多种写入方式进行测试。

[0094] 具体的,在本申请的一些实施方式中,上述读写方式的数量大于1,上述根据读写方式和测试地址,将第一测试数据写入至待测试芯片,可以包括:根据各个读写方式和测试地址,将第一测试数据多次写入至待测试芯片。

[0095] 其中,每次写入待测试芯片所采用的读写方式均不相同。也就是说,在本申请的一些实施方式中,可以用多种读写方式,多次进行测试,以测试芯片在不同读写方式上的表现。

[0096] 在本申请的一些实施方式中,上述待测试芯片的数据存储地址包含:存储块地址、行地址和列地址,并且,每种读写方式中,均包含一种对存储块地址、行地址和列地址的读写顺序,且每种读写方式包含的读写顺序均不相同。因此,在本申请的一些实施方式中,可以根据单个读写方式包含的读写顺序,向待测试芯片中的测试地址写入第一测试数据。

[0097] 具体的,上述多个读写方式可以包含以下几种读写方式的至少两种。

[0098] 第一种读写方式是存储块-行-列。其具体操作是选定一个存储块,选定一行,对这一行所有的列进行写入,然后,在将这一行所有的列写入完成后,换行。直到这个存储块的所有行都被写完,最后再换存储块,直到所有的地址写入完成。读取时与写入同理。

[0099] 第二种读写方式是存储块-列-行。其具体操作是先选定一个存储块,再选定一列,对这一列所有的行进行写入,然后换列,最后换存储块,直到所有的地址写入完成。读取时与写入同理。

[0100] 第三种读写方式是行-存储块-列,先选定一行(所有存储块的这一行),再选定一个存储块,对这个存储块所有的列进行写入,列写完后换存储块,直到存储块均写完后换行。一直循环直到所有的地址写入完成。读取时与写入同理。

[0101] 第四种读写方式是列-存储块-行。先选定一列(所有存储块的这一列),再选定一个存储块,对这个存储块所有的行进行读写,行写完后换存储块,直到存储块均写完后换列。一直循环直到所有的地址写入完成。读取时与写入同理。

[0102] 第五种读写方式是行-列-存储块。先选定一行(所有存储块的行),再选定这一行所有的列(所有存储块的列),最后选定一个存储块进行写入。一直循环直到所有的地址写入完成。读取时与写入同理。

[0103] 第六种读写方式是列-行-存储块,先选定一列(所有存储块的列),再选定这一行所有的行(所有存储块的行),最后选定一个存储块进行写入。一直循环直到所有的地址写入完成。读取时与写入同理。

[0104] 为了保证测试的全面性,在本申请的一些实施方式中,会同时获取六个读写方式,

并将六个读取方式分别进行测试。

[0105] 在本申请的另一些实施方式中,也可以是在获取多种读写方式之后,根据多种读写方式进行多种写入测试,每次写入测试的读写方式、第一测试数据和测试地址均不相同。

[0106] 利用多个读写方式,在每次写入之后均可以进行一次读操作,读取对应的测试地址指向的存储单元中存储的存储数据。然后,比对存储数据与第一测试数据,若存储数据与第一测试数据不相同,则可以确定待测试芯片的读写存在异常,并且说明待测试芯片不能够应用当前的读写方式。

[0107] 本申请的实施方式中,通过对不同读写方式的测试,每次测试所使用的读写方式不相同,模拟了芯片在使用过程中可能出现的不同写入方式,更加贴合芯片在实际使用过程中的实际情况。相比March算法或其他测试算法只适用一种读写方式,本申请提供的测试方法可以测试出在利用不同读写方式进行读写的过程中,由于不同读写方式对应的不同命令时序导致的故障。

[0108] 为了更全面的模拟芯片在实际使用过程中的运行状况,在本申请的另一些实施方式中,还可以获取预设的第一随机算法,并根据第一随机算法生成测试地址。然后,根据测试地址进行读写测试。

[0109] 由于March算法等的检测方式一般是顺序式的读写,并不能检测出模拟在随机地址读写时的故障。但是实际应用中,芯片可能是从某一特殊位置开始读写。因此,本申请的实施例,可以通过生成多个随机的地址,对多个随机地址就行读写,检测芯片在随机地址的应用中是否会出现故障,弥补常规测试中不能检测到的随机地址故障。

[0110] 其中,第一随机算法用于生成随机地址,可以由管理员根据实际情况进行选择,例如第一随机算法可以是生成三个随机数,每个随机数对应存储块、行和列。

[0111] 对随机地址的测试可以测试出芯片在随机地址的应用中是否会出现故障,而对随机数据的测试,则可以测试出芯片在读写随机数据时是否会出现故障。因此,在本申请的一些实施方式中,可以进行随机数据的测试。具体的,如图4所示,可以包括:步骤S401至步骤S404。

[0112] 步骤S401,获取待测试芯片中每个存储单元的存储容量。

[0113] 其中,上述存储单元的存储容量一般是在出厂时已经确定好的,对应的获取方式可以参考步骤S201中芯片规格的获取方式。

[0114] 步骤S402,获取预设的第二随机算法。

[0115] 步骤S403,根据存储容量和第二随机算法生成多个随机数据,每个随机数据的大小等于存储容量。

[0116] 其中,第二随机算法用于生成随机数据,可以由管理员实际情况进行选择。

[0117] 在本申请的一些实施方式中,上述终端可以根据单个存储单元的存储容量,利用上述第二随机算法是直接生成与存储容量相对应个数的数字,并对数字进行拼接得到随机数据。例如,单个存储单元的存储容量为32Bit,则上述第二随机算法可以生成32个数字,每个数字为0或1的其中一个,对32个数字进行拼接则可以得到一个随机数据。

[0118] 在本申请的另一些实施方式中,上述终端可以利用第二随机算法可以是直接生成一个数据,该数据的容量大于存储容量。然后,上述终端可以根据存储容量对该数据进行截取,得到大小等于存储容量一个随机数据。

[0119] 需要说明的是,本申请可以生成多个随机数据,并对每个随机数据进行测试,以保证测试的全面性。

[0120] 步骤S404,根据多个随机数据,确定第一测试数据。

[0121] 基于上述说明,在得到多个随机数据后,可以将每个随机数据确定为一个第一测试数据。然后对每个第一测试数据分别进行读写测试,可以确定芯片在读写时,是否会出现对某一随机数据读写存在异常。

[0122] 由于March算法等的检测方式一般是直接读写0或1,即如果一个存储单元能存储32bit的数据,则读写的是“00000000000000000000000000000000”(32个0)或者“11111111111111111111111111111111”(32个1)。但是实际应用中,如果32bit的数据中0和1都有,就要考虑0和1占比例的情况是否会影响到芯片的工作性能。因此本申请的实施方式中,通过随机数据的读写检测,随机组成与存储容量大小相同的数据,因此,第一测试数据是可以一个包含多个0和多个1的数据。通过0和1的多种排列,增加了更多的数据呈现,是正常定义数据所没有的,可以检测出March算法等的检测方式不能检测出的故障。

[0123] 在芯片的正常工作运行中,芯片会存在一边存储数据,一边读取数据的情况。因此,在本申请的一些实施方式中,还会继续并行读写的测试。具体的,在本申请的一些实施方式中,上述测试地址包含读测试地址和写测试地址。

[0124] 其中,上述写测试地址是指在测试过程中,需将第一测试数据写入的位置;上述读测试地址则是指在测试过程中,需并行读取数据的位置。

[0125] 上述读测试地址和写测试地址均可以由管理员进行选择。需要说明的是,读测试地址和写测试地址一般不同,但可以出现交叉。

[0126] 如果写测试地址对应的存储块、行、列为(0,0,0)至(2,2,2),则读测试地址不能为(0,0,0)至(2,2,2),但可以为(1,1,1)至(3,3,3)。

[0127] 如图5所示,上述根据读写方式,将第一测试数据写入到测试地址,还包括:步骤S501至步骤S502。

[0128] 步骤S501,根据读写方式和写测试地址,将第一测试数据写入至待测试芯片。

[0129] 步骤S502,在将第一测试数据写入至待测试芯片的过程中,根据读写方式,读取待测试芯片中读测试地址指向的第二存储数据。

[0130] 其中,第二存储数据是指读测试地址指向的存储单元中存储的数据。

[0131] 相应的,上述根据读写方式,读取待测试芯片中测试地址指向的第一存储数据,可以包括:根据读写方式,读取待测试芯片中写测试地址指向的第一存储数据。

[0132] 在将第一测试数据写入到写测试地址之后,可以根据读写方式,读取测试地址中存储的第一存储数据,并比对第一存储数据与第一测试数据。如果第一存储数据与第一测试数据相同,则说明芯片能够进行并行读写,即在写入第一测试数据的过程中,读取第二存储数据,并不会影响写入操作的结果。

[0133] 例如,在对A部分测试地址对应的存储单元进行写操作的同时,对B部分测试地址对应的存储单元进行读操作。如果完成写操作后,读取出的A部分存储单元中的第一存储数据,若与第一测试数据是相同的,则说明芯片可以同时进行读写操作。

[0134] 在本申请的实施方式中,通过在将第一测试数据写入到写测试地址的过程中,根据读写方式,读取读测试地址的数据,实现了对并行读写的测试。

[0135] 需要说明的是,在本申请的一些实施方式中,可以对多种测试方式进行组合。

[0136] 为了保证芯片测试的全面性和可靠性,如图6所示,在本申请的一些实施方式中,可以在使用March算法进行测试之后,分别进行不同容量数据的测试、不同读写方式的测试、随机地址测试、随机数据测试以及并行读写测试。其中任一项测试失败,则可以确认芯片的读写存在异常。若所有测试均成功,则可以确认芯片没有异常。

[0137] 需要说明的是,上述测试的顺序可以交换,即可以先进行并行读写测试再进行March算法的测试等等。

[0138] 在本申请的实施例,使用多种测试方式进行组合测试,可以更加全面的获知芯片的读写能力,方便测试人员对芯片进行标记、回收以及返工。

[0139] 需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本申请并不受所描述的动作顺序的限制,因为根据本申请,某些步骤可以采用其它顺序进行。

[0140] 如图7所示为本申请实施例提供的一种芯片的检测装置700的结构示意图,所述芯片的检测装置700配置于终端上。所述芯片的检测装置700可以包括:

[0141] 获取单元701,用于获取第一测试数据、测试地址和读写方式,所述测试地址为待测试芯片的数据存储地址;

[0142] 写入单元702,用于根据所述读写方式和所述测试地址,将所述第一测试数据写入至所述待测试芯片;

[0143] 读取单元703,用于根据所述读写方式,读取所述待测试芯片中所述测试地址指向的第一存储数据;

[0144] 测试单元704,用于比对所述第一存储数据与所述第一测试数据,并根据比对结果,确定所述待测试芯片的读写是否存在异常。

[0145] 在本申请的一些实施方式中,上述获取单元701,还具体用于:获取所述待测试芯片的存储总容量;根据所述存储总容量,确定多个测试数据体积;获取多个第一测试数据,其中所述第一测试数据的体积,与所述测试数据体积一一对应。

[0146] 在本申请的一些实施方式中,上述写入单元702,还具体用于:从多个所述第一测试数据中筛选出一个第二测试数据;获取测试首地址;根据所述第二测试数据的体积和所述测试首地址,确定测试末地址;根据所述读写方式、所述测试首地址和所述测试末地址,将所述第二测试数据写入至所述待测试芯片。相应的,上述检芯片的检测装置700还包括循环单元,用于:若所述芯片的读写不存在异常,则返回执行所述从多个所述第一测试数据中筛选出一个第二测试数据的操作,直至所有所述第一测试数据,均被写入至所述待测试芯片。

[0147] 在本申请的一些实施方式中,上述写入单元702,还具体用于:获取所述待测试芯片中每个存储单元的存储容量,以及所述待测试芯片的存储块数量、行数量和列数量;根据所述第二测试数据的体积、所述测试首地址和所述存储容量,确定测试末地址。

[0148] 在本申请的一些实施方式中,上述读写方式的数量大于1,上述写入单元702,还包括:根据各个所述读写方式和所述测试地址,将所述第一测试数据多次写入至所述待测试芯片,其中,每次写入所述待测试芯片所采用的所述读写方式均不相同。

[0149] 在本申请的一些实施方式中,上述待测试芯片的数据存储地址包含:存储块地址、

行地址和列地址；每种所述读写方式中，均包含一种对存储块地址、行地址和列地址的读写顺序，且每种读写方式包含的读写顺序均不相同；相应的，上述写入单元702，还具体用于：根据该读写方式包含的读写顺序，向所述待测试芯片中的所述测试地址写入所述第一测试数据。

[0150] 在本申请的一些实施方式中，上述获取单元701，还具体用于：获取预设的第一随机算法，并根据所述第一随机算法生成所述测试地址。

[0151] 在本申请的一些实施方式中，上述获取单元701，还具体用于：获取所述待测试芯片中每个存储单元的存储容量；获取预设的第二随机算法；根据所述存储容量和所述第二随机算法生成多个随机数据，每个所述随机数据的大小等于所述存储容量；根据所述多个随机数据，确定所述第一测试数据。

[0152] 在本申请的一些实施方式中，上述测试地址包含读测试地址和写测试地址；上述写入单元702还具体用于：根据所述读写方式和所述写测试地址，将所述第一测试数据写入至所述待测试芯片；在将所述第一测试数据写入至所述待测试芯片的过程中，根据所述读写方式，读取所述待测试芯片中所述读测试地址指向的第二存储数据。相应的，上述读取单元703，还有：根据所述读写方式，读取所述待测试芯片中所述写测试地址指向的第一存储数据。

[0153] 需要说明的是，为描述的方便和简洁，上述芯片的检测装置700的具体工作过程，可以参考图1至图6所述方法的对应过程，在此不再赘述。

[0154] 如图8所示，为本申请实施例提供的一种终端的示意图。该终端8可以包括：处理器80、存储器81以及存储在所述存储器81中并可在所述处理器80上运行的计算机程序82，例如芯片的检测程序。所述处理器80执行所述计算机程序82时实现上述各个芯片的检测方法实施例中的步骤，例如图1所示的步骤S101至S104。或者，所述处理器80执行所述计算机程序82时实现上述各装置实施例中各模块/单元的功能，例如图7所示单元701至704的功能。

[0155] 所述计算机程序可以被分割成一个或多个模块/单元，所述一个或者多个模块/单元被存储在所述存储器81中，并由所述处理器80执行，以完成本申请。所述一个或多个模块/单元可以是能够完成特定功能的一系列计算机程序指令段，该指令段用于描述所述计算机程序在所述终端中的执行过程。

[0156] 例如，所述计算机程序可以被分割成获取单元、写入单元、读取单元和测试单元。各单元具体功能如下：

[0157] 获取单元，用于获取第一测试数据、测试地址和读写方式，所述测试地址为待测试芯片的数据存储地址；

[0158] 写入单元，用于根据所述读写方式和所述测试地址，将所述第一测试数据写入至所述待测试芯片；

[0159] 读取单元，用于根据所述读写方式，读取所述待测试芯片中所述测试地址指向的第一存储数据；

[0160] 测试单元，用于比对所述第一存储数据与所述第一测试数据，并根据比对结果，确定所述待测试芯片的读写是否存在异常。

[0161] 所述终端可包括，但不仅限于，处理器80、存储器81。本领域技术人员可以理解，图8仅仅是终端的示例，并不构成对终端的限定，可以包括比图示更多或更少的部件，或者组

合某些部件,或者不同的部件,例如所述终端还可以包括输入输出设备、网络接入设备、总线等。

[0162] 所称处理器80可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0163] 所述存储器81可以是所述终端的内部存储单元,例如终端的硬盘或内存。所述存储器81也可以是所述终端的外部存储设备,例如所述终端上配备的插接式硬盘,智能存储卡(Smart Media Card,SMC),安全数字(Secure Digital,SD)卡,闪存卡(Flash Card)等。进一步地,所述存储器81还可以既包括所述终端的内部存储单元也包括外部存储设备。所述存储器81用于存储所述计算机程序以及所述终端所需的其他程序和数据。所述存储器81还可以用于暂时地存储已经输出或者将要输出的数据。

[0164] 所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,仅以上述各功能单元、模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能单元、模块完成,即将所述装置的内部结构划分成不同的功能单元或模块,以完成以上描述的全部或者部分功能。实施例中的各功能单元、模块可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中,上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。另外,各功能单元、模块的具体名称也只是为了便于相互区分,并不用于限制本申请的保护范围。上述系统中单元、模块的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0165] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中未详述或记载的部分,可以参见其它实施例的相关描述。

[0166] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0167] 在本申请所提供的实施例中,应该理解到,所揭露的装置/终端和方法,可以通过其它的方式实现。例如,以上所描述的装置/终端实施例仅仅是示意性的,例如,所述模块或单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通讯连接可以是通过一些接口,装置或单元的间接耦合或通讯连接,可以是电性,机械或其它的形式。

[0168] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0169] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0170] 所述集成的模块/单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请实现上述实施例方法中的全部或部分流程,也可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一计算机可读存储介质中,该计算机程序在被处理器执行时,可实现上述各个方法实施例的步骤。其中,所述计算机程序包括计算机程序代码,所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括:能够携带所述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory,RAM)、电载波信号、电信信号以及软件分发介质等。需要说明的是,所述计算机可读介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减,例如在某些司法管辖区,根据立法和专利实践,计算机可读介质不包括电载波信号和电信信号。

[0171] 以上所述实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围,均应包含在本申请的保护范围之内。

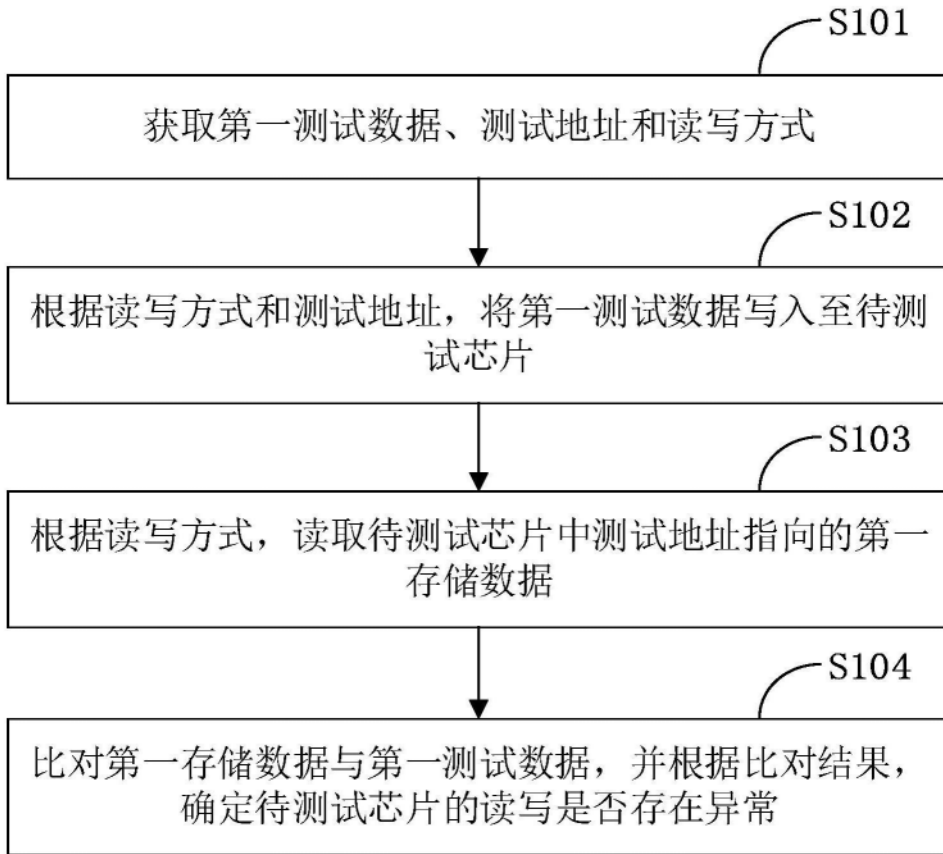


图1

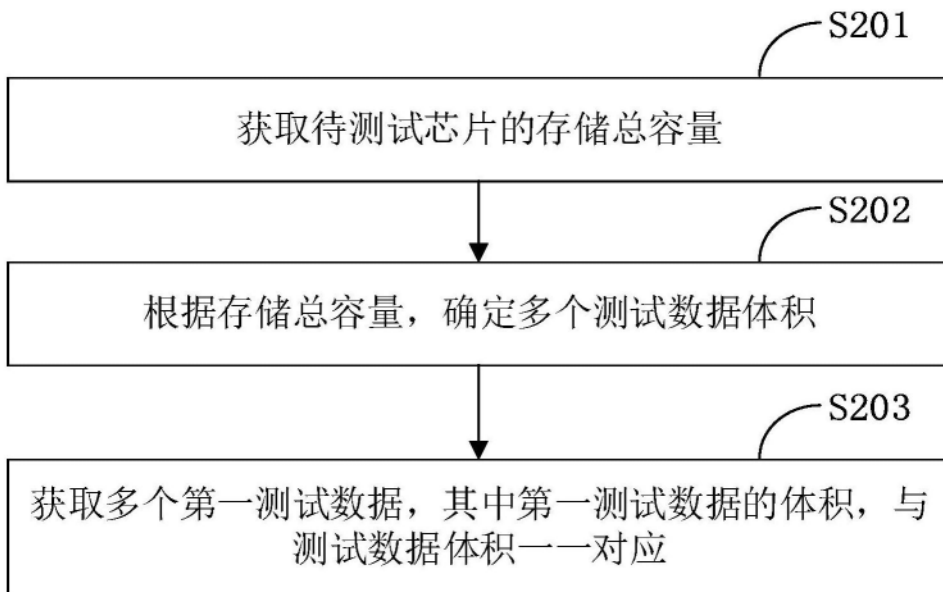


图2

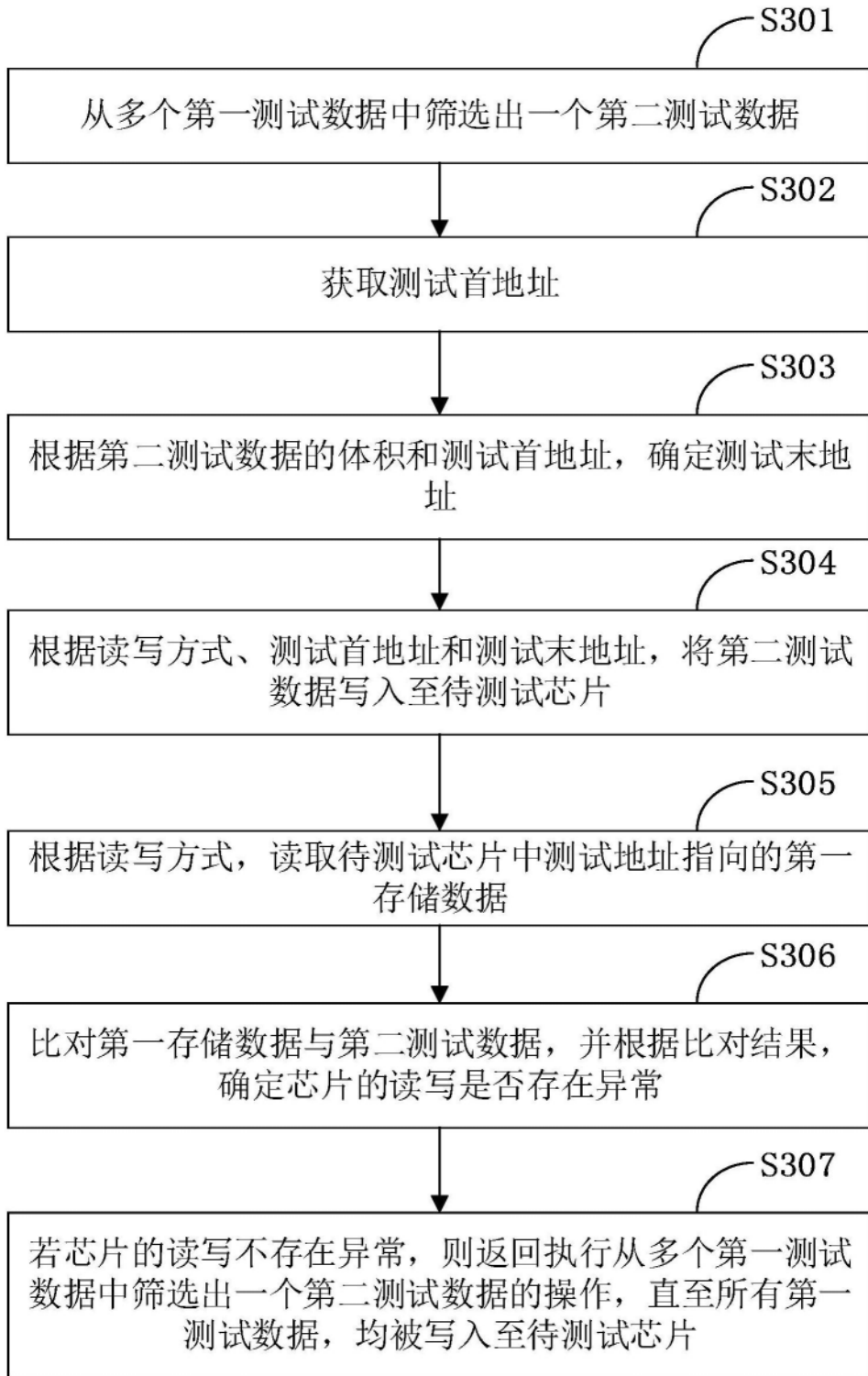


图3

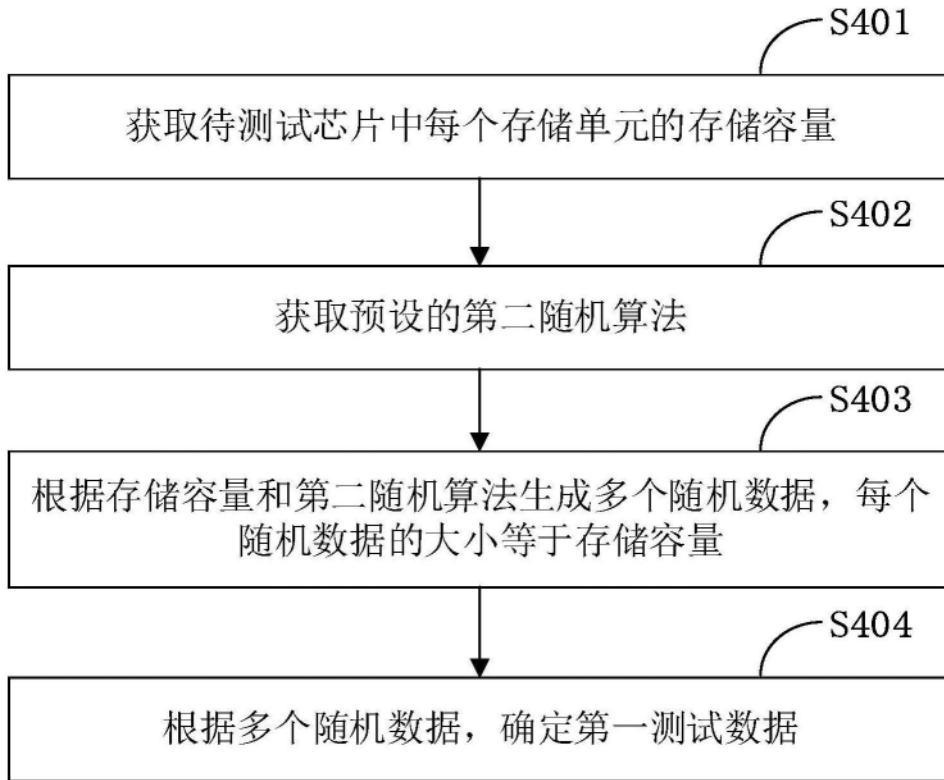


图4

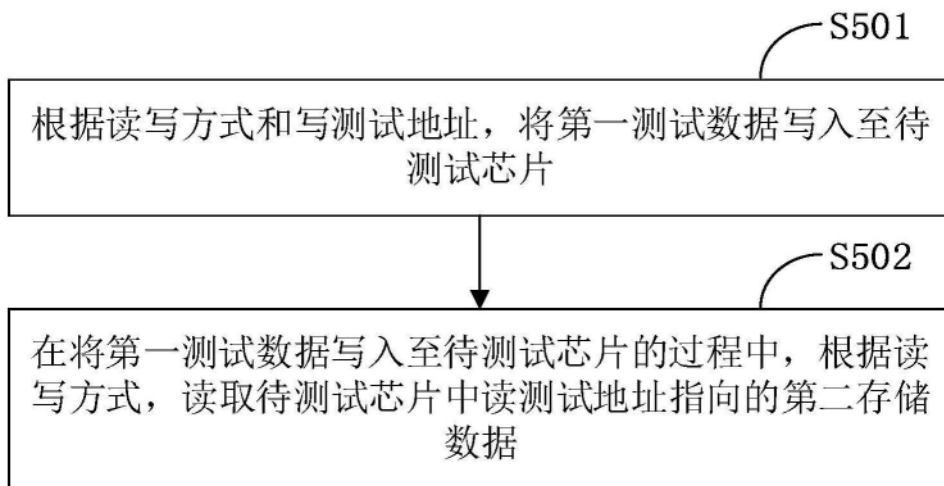


图5

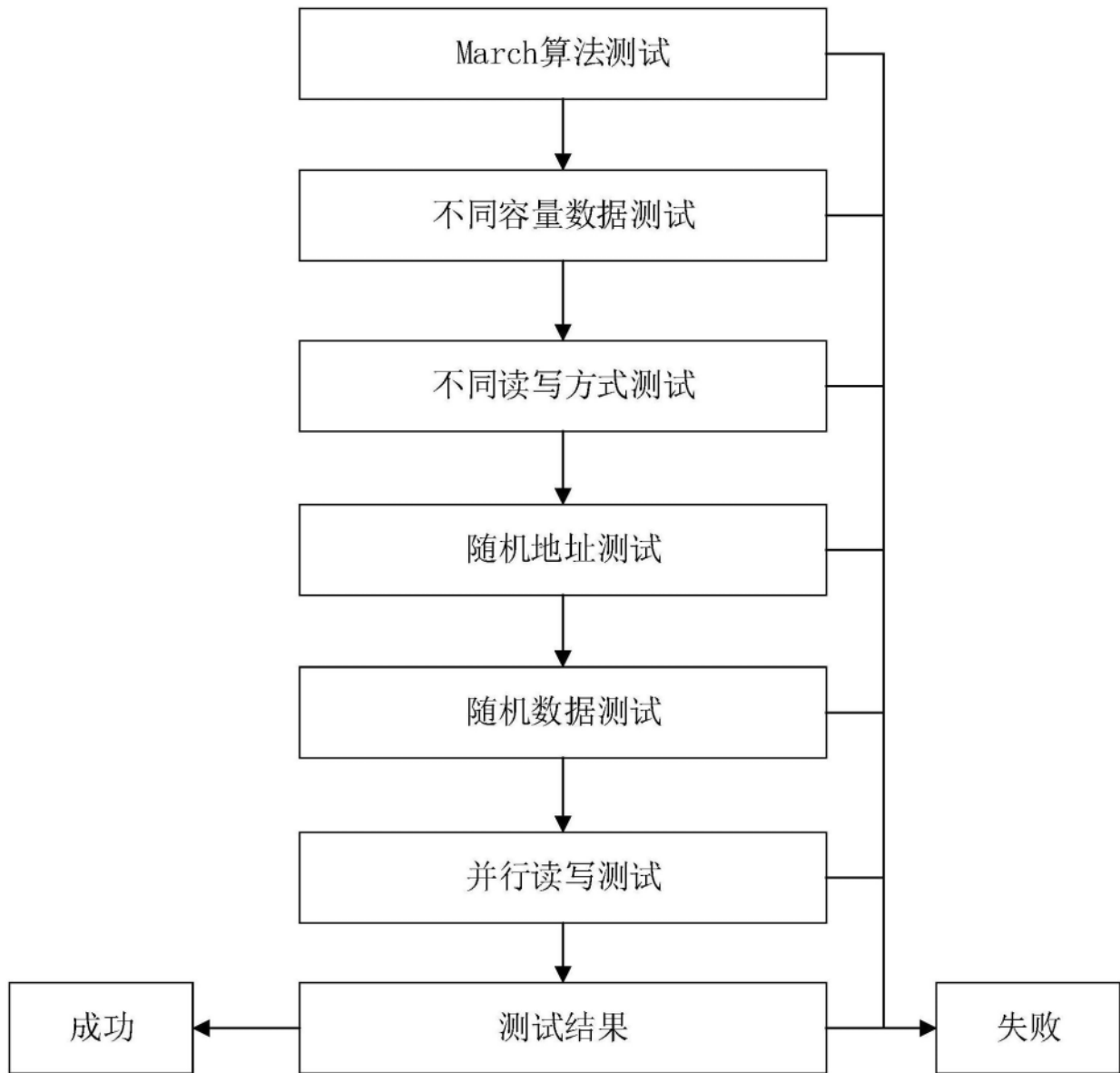


图6

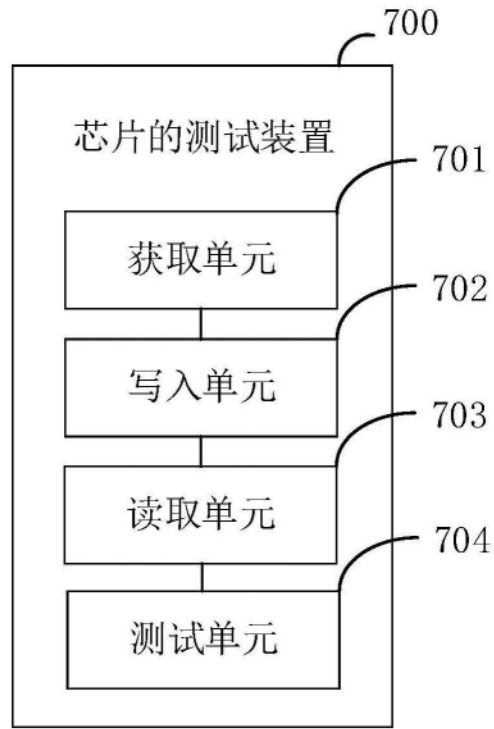


图7

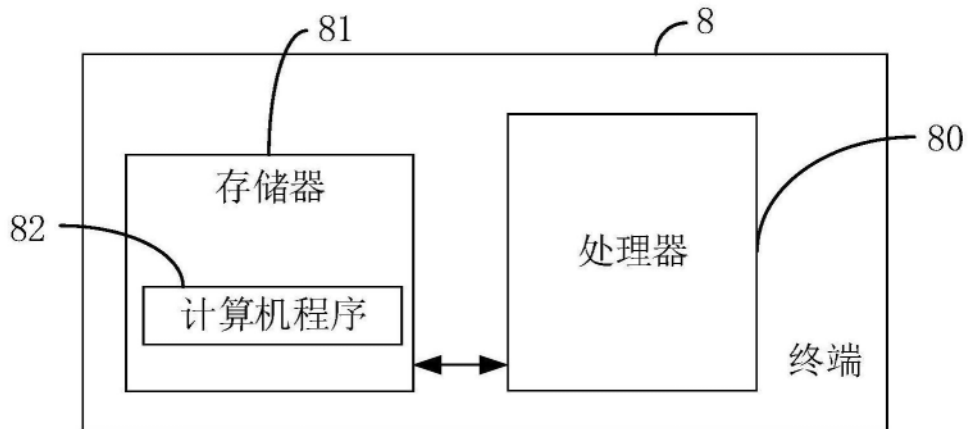


图8