[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G11C 7/00 (2006.01)
G11C 29/00 (2006.01)

[12] 发明专利说明书

专利号 ZL 02830034.3

[45] 授权公告日 2009 年 10 月 21 日

[11] 授权公告号 CN 100552805C

[22] 申请日 2002.12.16 [21] 申请号 02830034.3
[86] 国际申请 PCT/US2002/040473 2002.12.16
[87] 国际公布 WO2004/061852 英 2004.7.22
[85] 进入国家阶段日期 2005.6.13
[73] 专利权人 国际商业机器公司
地址 美国纽约
[72] 发明人 M·R·乌埃利特 J·罗兰
[56] 参考文献
US4819205A 1989.4.4
US5907514A 1999.5.25
审查员 程 琼

[74] 专利代理机构 北京市中咨律师事务所
代理人 于 静 杨晓光

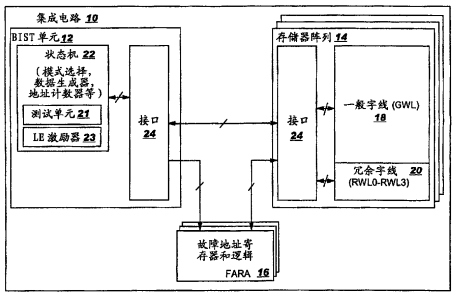
权利要求书 3 页 说明书 11 页 附图 4 页

[54] 发明名称

在测试期间启动存储器冗余的方法和电路

[57] 摘要

本发明涉及用于在存储器阵列的测试期间启动冗余存储器元件的方法和电路。 本发明实施例的集成电路包括：具有多个存储器元件的存储器阵列，存储器阵列包括：多个一般存储器元件；和至少一个用于与多个存储器元件中有故障的存储器元件进行交换的冗余存储器元件。 集成电路还包括内置存储器自测试单元，其包括：测试单元，用于确定存储器元件是否有故障，并且响应有故障产生故障信号，以及只有当冗余存储器元件已经被启动时，确定该冗余存储器元件是否有故障；冗余启动激励器，用于通过载入启动信号对该冗余存储器元件的启动进行定时；以及故障地址寄存器，用于当该故障信号被激活时，基于该载入启动信号控制相应的冗余存储器元件的启动。



1. 一种集成电路（10），包括：

具有多个存储器元件（18、20）的存储器阵列（14），所述存储器阵列包括：

多个一般存储器元件（18）；以及

至少一个用于与所述多个存储器元件中的有故障的存储器元件进行交换的冗余存储器元件（20）；

内置存储器自测试单元（12），包括：

测试单元（21），用于确定存储器元件是否有故障，并且响应有故障产生故障信号，以及只有当冗余存储器元件（20）已经被启动时，确定所述冗余存储器元件是否有故障；

冗余启动激励器（23），用于通过载入启动信号对所述冗余存储器元件的启动进行定时；以及

故障地址寄存器阵列（16），用于当所述故障信号被激活时，基于所述载入启动信号控制相应的冗余存储器元件的启动，其中故障地址寄存器阵列中的每个故障地址寄存器（26）包括：一组地址位（A0-An），用于包含将要被冗余存储器元件（20）替换的故障存储器元件的地址位置；启动位，用于控制其地址位置被包含在所述地址位中的存储器元件是否将被相应的冗余存储器元件替换；坏冗余位，用于覆盖所述启动位；暂时启动位，用于保存响应所述载入启动信号将被载入到所述启动位中的值；以及暂时坏冗余位，用于保存响应所述载入启动信号将被载入到所述坏冗余位中的值。

2. 根据权利要求 1 的集成电路，其中所述测试单元（12）通过将值写到存储器元件，并且确定在读操作期间所述存储器元件的输出是否与被写入的值相匹配，来确定所述存储器元件的故障。

3. 根据权利要求 2 的集成电路，其中所述测试单元（12）将值的一种模式写到所述存储器阵列的所述多个存储器元件。

4. 根据权利要求 1 的集成电路，其中所述冗余启动激励器（23）启动冗余，以防止所述测试单元（12）读还没有被写入值的冗余存储器元件（20）。

5. 一种用于修复存储器阵列（14）的方法，所述存储器阵列（14）具有多个存储器元件（18、20），其包括多个一般存储器单元（18）和至少一个冗余存储器元件（20），所述方法包括如下步骤：

测试所述存储器阵列中的所述一般存储器元件，以便确定哪些存储器元件有故障，并且只有当所述冗余存储器元件（20）已经被启动时，测试所述冗余存储器元件；以及

在测试期间基于寄存器（26）的载入启动信号、启动位、坏冗余位、暂时启动位以及暂时坏冗余位控制相应的冗余存储器元件的启动以便替换故障存储器元件，其中所述载入启动信号控制用于相应的冗余存储器元件（20）的寄存器（26）；启动位，用于控制其地址位置被包含在所述地址位中的存储器元件是否将被相应的冗余存储器元件替换；坏冗余位，用于覆盖所述启动位；暂时启动位，用于保存响应所述载入启动信号将被载入到所述启动位中的值；以及暂时坏冗余位，用于保存响应所述载入启动信号将被载入到所述坏冗余位中的值。

6. 根据权利要求 5 的方法，其中所述控制冗余存储器单元的启动步骤包括当测试不会造成伪故障时，启动所述冗余存储器元件（20）。

7. 根据权利要求 5 的方法，其中所述测试步骤包括通过将值写到存储器元件（18、20），并且确定在读操作期间存储器元件的输出是否与被写入的值相匹配，来确定所述存储器元件的故

障。

8. 根据权利要求 7 的方法，其中所述测试步骤将值的一种模式写到所述存储器阵列的多个存储器元件以及仅仅那些被启动的冗余存储器元件（18、20）。

9. 一种用于测试集成电路（10）的存储器（14）的缺陷的装置，所述装置包括：

测试单元（12），用于仅当冗余存储器元件（20）已经被启动以便替换故障存储器元件时，将值写入所述冗余存储器元件，并且确定在读操作期间所述冗余存储器元件的输出是否与被写入的值相匹配，来测试所述冗余存储器元件。

10. 根据权利要求 9 的装置，其中所述故障存储器元件是另一个冗余存储器元件（20）。

11. 根据权利要求 9 的装置，还包括用于在所述用于测试的装置的操作期间控制冗余实现的定时的装置（23）。

12. 根据权利要求 9 的装置，其中所述测试单元（12）还测试非冗余存储器元件（18）。

在测试期间启动存储器冗余的方法和电路

技术领域

本发明一般地涉及存储器冗余的启动（enablement）。

背景技术

一般而言存储器阵列并且具体而言静态随机存取存储器（SRAM）占据了许多专用集成电路（ASIC）芯片上大部分的硅面积。这些存储器阵列往往对这些芯片的制造成品率造成不良影响，这是因为它们是芯片的如此大的一部分，并且它们往往会将技术的制造工艺推至极限。然而，它们的极其重复和可预知的设计使得它们适合于在芯片被制造好后修复存储器的方法。典型的存储器修复方法是包括额外的或“冗余的”行或列，它们将被与有缺陷的存储器元件进行“交换”。用于测试和修复包括行（即，字线）冗余的固定设计的SRAM存储器阵列的传统方法分为三个大类。

一种技术规定，在测试一般的存储器元件的同时（即，在熔丝烧断之前）测试所有冗余存储器元件，并且如果任何冗余存储器元件出现故障，则将芯片标记为是不可修复的。此技术是有利的，因为存在与修复芯片有关的巨大成本。因此，能够越快地将芯片识别为“不可修复的”，在测试过程中就可以节约越多的钱，因为在模块测试时将不会出现有故障的冗余元件。然而，这种解决方法效率低而且费用高，因为在不被需要用于修复芯片的冗余存储器元件中发现的缺陷将导致可修复的芯片被扔掉。

用于测试冗余存储器元件的第二种技术是等待直到有故障的一般存储器元件已经被以冗余存储器元件替换之后（即，在熔丝烧断之后）为止，并且测试所述冗余存储器元件，如同它们是它们所替换的一般存储器元件一样。此技术比第一种技术有利，因为当在未被使用的冗余存储器元件中发现缺陷时可修复的芯片不会被扔掉。然而，此技术直到在熔丝烧断之后

才识别有故障的冗余存储器元件，因此带来了测试时间、熔丝烧断、以及可能封装到模块中（这常常包括芯片成本的一半或更多）的增加的成本。

最后，第三种技术规定，如在第一种技术中那样，在替换一般的存储器元件之前测试所有的冗余存储器元件，但是提供一种附加机制，通过这种机制，在替换一般的存储器元件时识别并且在匹配时避开有故障的冗余存储器元件。通过减少或消除熔丝烧断之后有故障的冗余存储器元件的数量，并且在测试过程中早期识别不可修复的芯片，这种技术节约了成本。然而，引入了硅芯片面积和测试复杂性的附加成本。在高密度存储器阵列例如动态 RAM（DRAM）中，这种额外的成本是有理由的，但对于较高性能、较低密度的存储器例如 SRAM 和寄存器阵列（RA）来说是不可接受的。如同在第二种技术中，附加的成本是在已经替换了存储器元件之后，在相同条件下再次测试芯片所带来的额外的测试时间。

可编辑（或可定制）存储器表现出了实现和测试冗余存储器元件的其他障碍。例如，在可编辑存储器设计中，测试和在匹配时避开有故障的冗余存储器元件要麻烦得多。

鉴于以上所述，本领域需要有用于在测试期间测试并修复冗余存储器元件的更有效的方法和装置。

发明内容

本发明涉及用于在存储器阵列的测试期间，启动冗余存储器元件的方法和装置。本发明仅在冗余存储器元件已被启动以便替换有故障的存储器元件时提供该冗余存储器元件的测试。因此，避免了未被使用的冗余存储器元件的测试。本发明实现了内置的自测试（BIST）单元，该内置自测试单元当测试存储器时运行多种模式以便隔离并检测特定故障。本发明在测试期间的多个点上以冗余存储器元件修复任何新发现的故障，从而在不引起伪故障的情况下尽早地检测并修复故障。然后，在后续的模式测试过程中测试被激活的冗余存储器元件中的故障。结果，通过在冗余存储器元件被启动以便进行修复时测试并修复冗余存储器元件，一般存储器元件空间

中的所有故障最终将被修复。因为修复发生在测试期间，用户可以在任何时间修复存储器，例如，在启动 IC 的过程中，而不仅仅是在制造过程中。这种功能允许修复在完成制造之后很久才出现的可靠性故障，因此，防止了 IC 被返回制造者。此外，本发明包括具有控制位的寄存器，所述控制位允许使用相同的锁存器收集有故障的地址以及实现冗余以便替换它们。

根据以下对本发明的实施例的更具体的说明，本发明的上述和其他特征将很明显。

附图说明

参考以下附图，将详细地描述本发明的实施例，其中类似的标记指示类似的元件，并且其中：

图 1 示出了本发明的一个实施例的具有 BIST 的集成电路的方块图；

图 2 示出了图 1 的故障地址修复寄存器（FARA）的方块图；

图 3 示出了图 2 的故障地址寄存器（FAR）的方块图；

图 4 示出了本发明的操作的流程图。

具体实施方式

参见附图，现在将描述与 SRAM 应用有关的用于在存储器测试期间启动冗余的方法和装置。然而，应当认识到，本发明的技术可以应用于任何类型的存储器。如图 1 中所示，集成电路 10（IC）包括，尤其是，存储器内置自测试（BIST）单元 12，所述存储器内置自测试单元 12 与一个或多个存储器阵列 14 以及每个存储器的相应的故障地址寄存器和逻辑（FARA）16 相匹配。每个存储器阵列 14 可以包括多个一般字线或存储器元件 18（下文中称为“GWL”），以及一组冗余字线或存储器元件 20（下文中称为“RWL”）的一种特定设计。GWL 是那些最初旨在用于操作的存储器元件，而 RWL 是被提供用于替换有故障的存储器元件（GWL 或 RWL）的存储器元件。“替换”的意思是，RWL 被用于交换或互换所述有故障的存储器元件。或者，每个存储器阵列 14 可以采取可编辑的（或可定制的）存储器设计的形式，其中，

所述多个 GWL 18 和一组 RWL 20 的数目可以由用户选择。

BIST 单元 12 包括状态机 22，所述状态机 22 包括，尤其是，用于实现存储器阵列 14 的自测试的测试单元 21，在这种测试中值被写入被启动的存储器元件（即，GWL 和被启动的 RWL）中，并且然后被读回来。然后测试单元 21 通过将在读操作过程中存储器元件的输出与被写入其中的值进行比较确定是否存在故障。如果输出不匹配，则激活（生成）故障信号。BIST 单元 12 可以包括普通的功能，例如写模式选择、数据生成器、地址计数器等，以便执行自测试。此外，BIST 单元 12 包括载入启动信号（LE）激励器 23， LE 激励器 23 被配置为在测试期间确定这样的适当的（多个）时刻或（多个）点，在所述时刻或点改变冗余状态，即，可以激活启动/禁止 RWL 而不造成伪故障。将在下面描述 LE 激励器 23 的细节。BIST 单元 12 和存储器阵列 14 中的每一个都可以包括适当的接口 24。FARA 16 包括一组故障地址寄存器（FAR）26 和控制逻辑 28（图 2）。每个存储器阵列 14 可以包括一个相关联的 FARA 16，或者一组存储器阵列 14 可以共享一个 FARA 16。虽然作为一个单独的实体示出了 FARA 16，但是应当认识到，FARA 16 可以作为存储器阵列 14 的一部分被提供。

在图 2 中更详细地示出了一个示例性 FARA 16，它实现了 4 个 RWL 并且没有冗余位线(列)。FARA 16 为每个 RWL 包括一个故障地址寄存器(FAR) 26。被表示为 FAR0 至 FAR3 的 4 个 FAR 中的每一个馈入被表示为 RWL0 至 RWL3 的相应的 RWL。如将在下面描述的，控制逻辑 28 可以向每个 FAR 26 输出 LOAD（载入）信号和 DISABLE（禁止）信号。控制逻辑 28 还从存储器阵列 14 接收故障信号，并且如果存储器阵列 14 不能被修复，则输出不可修复信号。虽然示出了 4 个 RWL/FAR，但是应当认识到，存储器阵列 14 可以实现所需的任何数目的 RWL/FAR。

图 3 中示出了 FAR 26 的细节。每个 FAR 26 包括 n 个地址位 A0－An（图 3 中，n = 8）以及下面的控制位或锁存器：启动 EN、暂时启动 TE、坏冗余 BR 以及暂时坏冗余 TB。地址位 A0－An 包含将由 RWL 替换的故障存储器元件的地址位置。启动位 EN 控制其地址位置被包含在地址位 A0－An

中的存储器元件是否将被相应的冗余存储器元件所替换，即，它激活 FAR，并且启动相应的 RWL。暂时启动 TE 用于保存响应载入启动信号将被载入到启动位 EN 中的值，即，暂时启动 TE 指出当 LE 激励器 23 通过激活载入启动信号确定适当时，FAR 26 的相应的 RWL 将被激活。坏冗余位 BR，当被设置时，通过与门 32 覆盖启动位 EN，即，当被设置时，它禁止 FAR 26 以及相应的 RWL。暂时坏冗余位 TB 用于保存响应载入启动信号将被载入到坏冗余位 BR 中的值，即，当被设置时，暂时坏冗余位 TB 指出当 LE 激励器 23 通过激活载入启动信号确定适合时，FAR 以及相应的 RWL 将被禁止。

每个 FAR 26 包括相应的比较逻辑部分 30，并且接收载入启动信号 LE（当激活时）以及故障地址。此外，来自 FARA 控制逻辑 28 的 LOAD 信号馈入到暂时启动位 TE 和地址位 A0－An，并且来自逻辑 28 的 DISABLE 信号馈入到暂时坏冗余位 TB。载入启动信号 LE 馈入到启动位 EN 以及坏冗余位 BR。暂时启动位 TE 馈入到启动位 EN 并且馈入到 FAR 比较逻辑 30，并且暂时坏冗余位 TB 馈入到坏冗余位 BR。当所述载入启动信号 LE 激活时，如果没有设置启动位 EN，则暂时启动位 TE 的值被强加给启动位 EN，并且如果没有设置坏冗余位，则暂时坏冗余位 TB 的值被强加给坏冗余位 BR。如果相应的位（EN 或 BR）已经被设置，则忽略相应的暂时位（分别为 TE 或 TB）。启动位 EN 馈入到与门 32，并且坏冗余位 BR 在反转以后馈入到与门 32。因此，所得到的存储器启动信号 RE 最终被坏冗余位 BR 控制，即，当 BR 位被设置时，它覆盖启动位 EN 的值。启动位 EN 和坏冗余位 BR 的每一个都馈入 FAR 比较逻辑部分 30。地址位 A0－An 被连接到比较逻辑部分 30 并且被连接到存储器阵列 14（通过 RA 信号）。存储器启动信号 RE 也通过与门 32 的操作数连接到存储器 14。

出于说明的目的，“没有被设置”的意思是位具有“0”值，而“被设置”的意思是位具有“1”值。然而，应当认识到，逻辑可以被设计为位是低电平激活的，即“被设置”＝“0”，而“没有被设置”＝“1”，而不影响本发明的精神。此外，设置或状态“X”的意思是位或是没有值被写入其中或是该值不是即刻与本说明有关。

参见图 4，现在将描述本发明的操作。应当认识到，虽然将提供一个步骤接一个步骤的说明，但是对于所述方法而言，也许并不是所有的步骤都是必须的。此外，可以与所述一组步骤不同的其他步骤来叙述本发明的操作。

在第一步骤 S1 中，FARA 16 被初始化为两个状态中的一个。在第一初始化状态中：对于所有的 FAR 26，控制位 TE、TB、EN 以及 BR 没有被设置，即为 0。地址位 A0－An 被认为处于“X”状态。然而，实际上，地址寄存器 A0－An 通常被设置为一个已知的值，例如，全部是“0”。另一方面，在第二初始化状态中：将前面的冗余解决方案载入某些或所有的 FAR 26。例如，A0－An 包含地址数据，启动位 EN 被设置，坏冗余位 BR 被设置（或没有被设置），并且暂时启动和暂时坏冗余位 TE 和 TB 被设置为“X”。剩余的（即，未使用的）FAR 26 按照以上所述第一状态被初始化为“没有被设置”。

在第二步骤 S2 中，由测试单元 21 执行一个操作。在此步骤中，如以上所述，测试单元 21 将值写入被启动的存储器元件（即，GWL 和被启动的 RWL），并且然后将值读回来。然后通过将存储器元件在读操作期间的输出与被写入的值进行比较，测试单元 21 确定是否存在故障。如果输出不匹配，则激活（产生）故障信号。在一个实施例中，测试单元 21 被配置为通过顺序地将不同的（即，多个）模式写入各存储器元件并且然后将它们读回来测试每个激活的存储器元件，以便确定数据是否被正确地存储。示例性的模式可包括覆盖层（blanket）0、覆盖层 1、方格盘（checkerboard）、反转方格盘（reverse checkerboard）、字线条带、位线条带等。如下面将变得很明显的，所述顺序测试允许修复有故障的存储器元件，并且然后测试被用于进行修复的 RWL。

在步骤 S3，确定被定时控制的载入启动（LE）信号是否已经被 LE 激励器 23 激活。当载入启动（LE）信号激活，即，被设置时，它指出存在改变 FAR 26 的冗余状态的适当的时刻／点。在测试期间，当检测到 GWL 中的故障时，测试单元 21 可以在若干机器周期中不识别该故障。如果在故障检

测之后立即启动 RWL，则随后的一机器周期可能在还没有向该 RWL 中写入任何东西时读该 RWL。因此，测试单元 21 将会指出该 RWL 的故障，即使它没有任何问题。通过确定何时适合于改变 RWL 的状态，例如从被禁止改变为被启动或者从被启动改变为被禁止，载入启动（LE）信号防止了这种伪故障的发生。适当的时刻可以是，例如，恰好在测试单元 21 将要开始向所有地址写之前，以及在测试单元 21 已经评估了所有以前的读取操作的故障状态之后。

在步骤 S4，如果载入启动信号已被激活，则所有被设置的暂时启动位 TE 都被载入到尚未被设置的相应的启动位 EN 中，并且所有被设置的暂时坏冗余位 TB 被载入到尚未被设置的相应的坏冗余位 BR 中。即，以信号通知 FAR 26 改变冗余状态。如以上所述，如果各个位（EN 或 BR）已经被设置，则忽略相应的暂时位（分别为 TE 或 TB）。下面将更详细地描述这个步骤的含义。

在步骤 S5，确定存储器阵列 14 是否正被访问（读）。如果没有，逻辑返回到步骤 S2 并且进行进一步测试。如果存储器是工作着的，则逻辑进入到步骤 S6，在步骤 S6，确定是否已经检测到存储器元件的故障。如以上就测试单元 21 所描述的，当一个地址没有读回被写入到该地址中的数据时，则检测到故障。如果没有检测到故障，操作继续步骤 S2 上的测试。

如果检测到故障，则操作进入步骤 S7，在步骤 S7 中，将故障信号转发给 FAR 比较逻辑部分 30，并且确定该故障存储器元件地址是否与任何以前被检测到的故障存储器元件地址相匹配。即，由每个 FAR 的比较逻辑部分 30 将该故障存储器元件地址与被存储在其中的相应地址位 A0－An 进行比较，以便确定所述故障是新检测到的故障还是以前已经被检测到的故障。如果该故障存储器元件地址与该多个 FAR 26 中的一个中所存储的地址相匹配，则将匹配信号转发给 FARA 控制逻辑 28，指出该故障不是新检测到的故障，并且操作进入下面讨论的步骤 S10。然而，如果该故障存储器元件地址与该多个 FAR 26 中的任何一个中所存储的任何地址都不匹配，则表示所述故障以前没有被检测到，即，它是新的故障。在这种情况下，在步骤

S8 用于下一个可用的 FAR/RWL 的暂时启动位 TE 被设置，并且在步骤 S9 故障存储器元件地址被载入到用于该 RWL 的相应的 FAR 中。通过来自 FARA 控制逻辑 28 的 LOAD 信号进行暂时启动位的设置和地址载入。通过为下一个 RWL 设置暂时启动位 TE， 下次当 LE 激励器 23 确定实现冗余状态改变的适当时间存在时，用于该 RWL 的暂时启动位 TE 被载入到启动位 EN 中，以便启动该 RWL。因此，可以注意到需要替换故障存储器元件，并且所述启动被阻止直到一个适当的时刻为止，以便防止由在测试期间在还没有向被启动的 RWL 中写入任何东西时读该被启动的 RWL 而引起的伪故障。

转到步骤 S10，确定是否设置了用于任何匹配的 FAR 26 的启动位 EN。提供这个步骤以便处理由于 FARA 16 的初始化状态而出现的情况。特别地，4 个控制位（EN、TE、BR、TB）中的每一个都被初始化为“没有被设置”，并且地址位 A0－An 被初始化为某个值。当测试在与一个或多个所述初始化地址相匹配的地址中检测到故障时，所述情况就会发生。例如，一个或多个初始化地址可以都是 0，并且测试可能在地址为 0000 0000 的 GWL 中测试到故障，从而产生一个或多个匹配地址。在这种情况下，系统必须仍然能够将故障地址载入到 FARA 16 中。步骤 S10 的是否设置了用于任何匹配地址（可以是多于一个）的启动位 EN 的判定指出该故障是否实际上已经被载入了。如果没有设置用于任何匹配地址的启动位 EN，则表示该故障还没有被持久地载入，并且操作进入步骤 S11。

在步骤 S11，确定是否设置了用于该单个匹配 FAR 的暂时启动位 TE，即，匹配 FAR 的任何重复都被忽略。如果设置了暂时启动位 TE，则表示此故障的修复正在进行，并且在等待通过载入启动信号 LE 启动，即，等待适当的时刻以便改变冗余状态。在这种情况下，操作通过继续步骤 S2 上的检测进入下一个机器周期。然而，如果没有设置暂时启动位 TE，则表示此故障尚未被载入。在这种情况下，在步骤 S8，用于下一个可用的 FAR 26 的暂时启动位 TE 被设置，并且在步骤 S9，将故障存储器元件地址载入到相应的 FAR 26 地址位中，从而当载入启动信号 LE 激活时，可以通过激活相应的 RWL 进行修复。

转到步骤 S10，如果任何匹配 FAR 的启动位 EN 被设置了，则表示这个故障已经被载入，并且操作进入步骤 S12。在步骤 S12，确定用于任何匹配 FAR（可以是多于一个）的坏冗余位 BR 是否没有被设置。如果用于所有匹配 FAR 的坏冗余位 BR 都被设置了，则表示这个故障是由于以前的一个被启动的 RWL 引起的，该 RWL 也出现了故障，即是被禁止的或死亡的（dead）RWL。将以另一个 RWL 来完成这个 RWL 故障的修复，或者可以等待通过载入启动信号 LE 的激活来修复。在这种情况下，通过继续步骤 S2 上的测试，操作进行到下一个机器周期。如果用于任何匹配 FAR 的坏冗余位 BR 没有被设置，则表示此故障是在这个机器周期上新检测到的被启动的 RWL 的故障，或者正等待通过载入启动信号 LE 的激活被禁止，并且操作进入步骤 S13。

在步骤 S13，通过确定是否设置了用于该单个匹配 FAR 的暂时坏冗余位 TB，来确定出现了以上情况中的哪一种。如果设置了暂时坏冗余位 TB，则表示该 RWL 故障仅仅是等待载入启动信号 LE 以便将 RWL 的状态改为是坏的。在这种情况下，操作进入步骤 S2 上的测试。然而，如果没有设置暂时坏冗余位 TB，则表示这个 RWL 故障是在这个机器周期上新检测到的，并且在步骤 S14，该匹配 FAR 的暂时坏冗余位 TB 被设置，即，被来自 FARA 控制逻辑 28 的 DISABLE 信号设置。随后，在步骤 S8，设置下一个可用的 FAR 26 的暂时启动位 TE，并且将地址载入到相应的 FAR 26 的地址位中，以便当载入启动信号 LE 被激活时，可以通过相应的 RWL 的激活进行修复。

本发明的全部操作可以被描述如下：如果观察到故障，则将故障信号（图 2）从存储器阵列 14 传送到 FARA 控制逻辑 28。将故障地址与 FARA 16 中每个 FAR 26 的内容进行比较，并且按照以下情况，逻辑 28 向下一个可用的空的寄存器发出 LOAD 信号：

情况 1：如果故障地址与任何 FAR 26 的内容都不匹配，那么将该故障地址被载入到下一个可用的 FAR 中，并且设置所述下一个可用 FAR 的相应的暂时启动位 TE。

情况 2：如果故障地址与至少一个 FAR 地址相匹配，并且没有设置用于每个匹配的 FAR 26 的暂时启动位 TE 和启动位 EN，则将该故障地址载入

到下一个可用的 FAR，并且设置所述下一个可用的 FAR 的相应的暂时启动位 TE。

情况 3：如果故障地址与一个 FAR 相匹配，并且用于这个 FAR 26 的启动位 EN 被设置了，并且用于每个匹配的 FAR 的暂时坏冗余位 TB 和坏冗余位 BR 都没有被设置，则将该故障地址载入到下一个可用的 FAR，设置所述下一个可用的FAR的相应的暂时启动位TE，并且设置用于这个匹配的FAR的暂时坏冗余位 TB。

在所有其他情况中，没有 FAR 的内容被更新。

在测试期间的某个（某些）点上，LE 激励器 23 激活载入启动信号 LE，所述载入启动信号 LE 迫使暂时启动位 TE 的值被载入到启动位 EN 中，并且复位暂时启动位 TE。当启动位 EN 被设置为“1”并且坏冗余位 BR 没有被设置为“0”时，与门 32 启动（通过冗余启动（RE）信号）存储器阵列 14 中的 RWL 以便替换此地址，即冗余地址（RA）。类似地，当 LE 激励器 23 激活载入启动信号 LE 时，它迫使暂时坏冗余位 TB 的值被载入到坏冗余位 BR 中，并且复位暂时坏冗余位 TB。在这种情况下，当启动位 EN 被设置为“1”并且坏冗余位 BR 被设置为“1”时，与门 30 禁止（通过冗余启动（RE）信号）存储器阵列 14 中的 RWL，即 BR 覆盖 EN。

上述的发明提供了若干优点。第一，通过在多模式测试期间控制启动的（多个）时刻，从而防止基于过早启动冗余的伪故障，本发明允许动态的冗余实现。结果，可以在单个、多个模式的自测试期间修复存储器元件故障。第二，所述方法和装置允许用户在启动 IC 期间通过 BIST 修复存储器，而不是仅仅在制造过程中修复。这种功能允许修复在完成制造之后很久发生的可靠性故障，因此，防止了 IC 被返回给制造商。第三，对一般存储器元件（GWL）和仅仅已经被启动以便替换故障存储器元件的冗余存储器元件的测试使得测试更加高效。最后，暂时启动 TE 位和暂时坏 TB 位的包含和使用允许使用相同的锁存器来收集故障地址并且实现冗余以便替换它们。结果，对于熔丝，许多选择现在都成为可能：1）它们可以被删除，并且仅在 BIST 期间完成修复，2）它们可以被保持，并且被用作 BIST 的开始

点，或者 3）由 BIST 完成的修复的结果可以被硬连线在熔丝中，在这个步骤之后可能有进一步的自测试。

在前面的讨论中，应当理解所讨论的方法步骤是由被包含在 IC 10 内的硬件，以及由一组寄存器和控制硬件实现的。然而可以理解，此处所描述的各种设备、模块、机制和系统可以用硬件或软件或硬件和软件的组合来实现，并且可以被不同于所示出的那样划分。它们可以由适合于执行此处所述的方法的任何类型的计算机系统或其他装置来实现。典型的硬件和软件的组合可以是具有计算机程序的通用计算机系统，当载入并执行所述程序时，所述程序控制所述计算机系统，以便执行此处所述的方法。或者，可以使用包含专用硬件的专用计算机来实现本发明的一个和多个功能任务。本发明还可以被嵌入到计算机程序产品中，所述计算机程序产品包括使得能够实现此处所述的方法和功能的所有特征，以及当它被载入到计算机系统中时，能够执行这些方法和功能。在本文中，计算机程序、软件程序、程序、程序产品或者软件是指一组指令的采用任何语言、代码或符号的任何表达方式，所述指令旨在使得具有信息处理能力的系统或是直接地或者在以下操作之后实现特定功能：（a）转换到另一种语言、代码或符号；和/或（b）以不同的材料形式再现。

虽然已经结合若干优选实施例描述了本发明，但是本领域技术人员将认识到，可以在以下权利要求的精神和范围内以不同的方案实施本发明。

工业实用性

本发明可用于测试并实现任何存储器中，并且尤其是在 SRAM 中的冗余存储器元件。

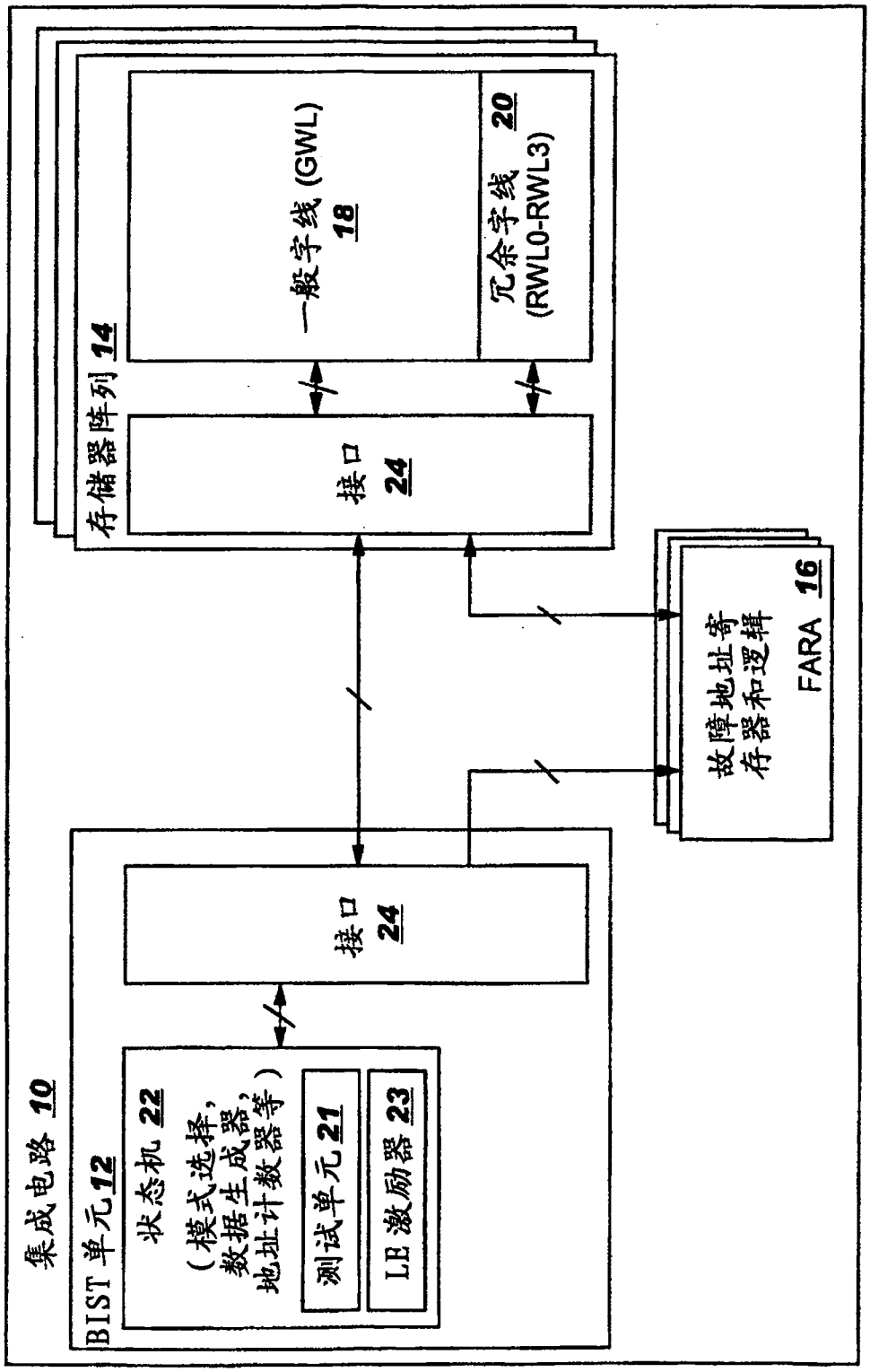

集成电路 10
BIST 单元12
状态机 22
（模式选择，
数据生成器，
地址计数器等）
测试单元 21
LR 激励器 23
接口
24
存储器阵列 14
一般字线 (GWL)
18
冗余字线 20
(RWL0-RWL3)
接口
24
故障地址寄
存器和逻辑
FARA 16

图 1

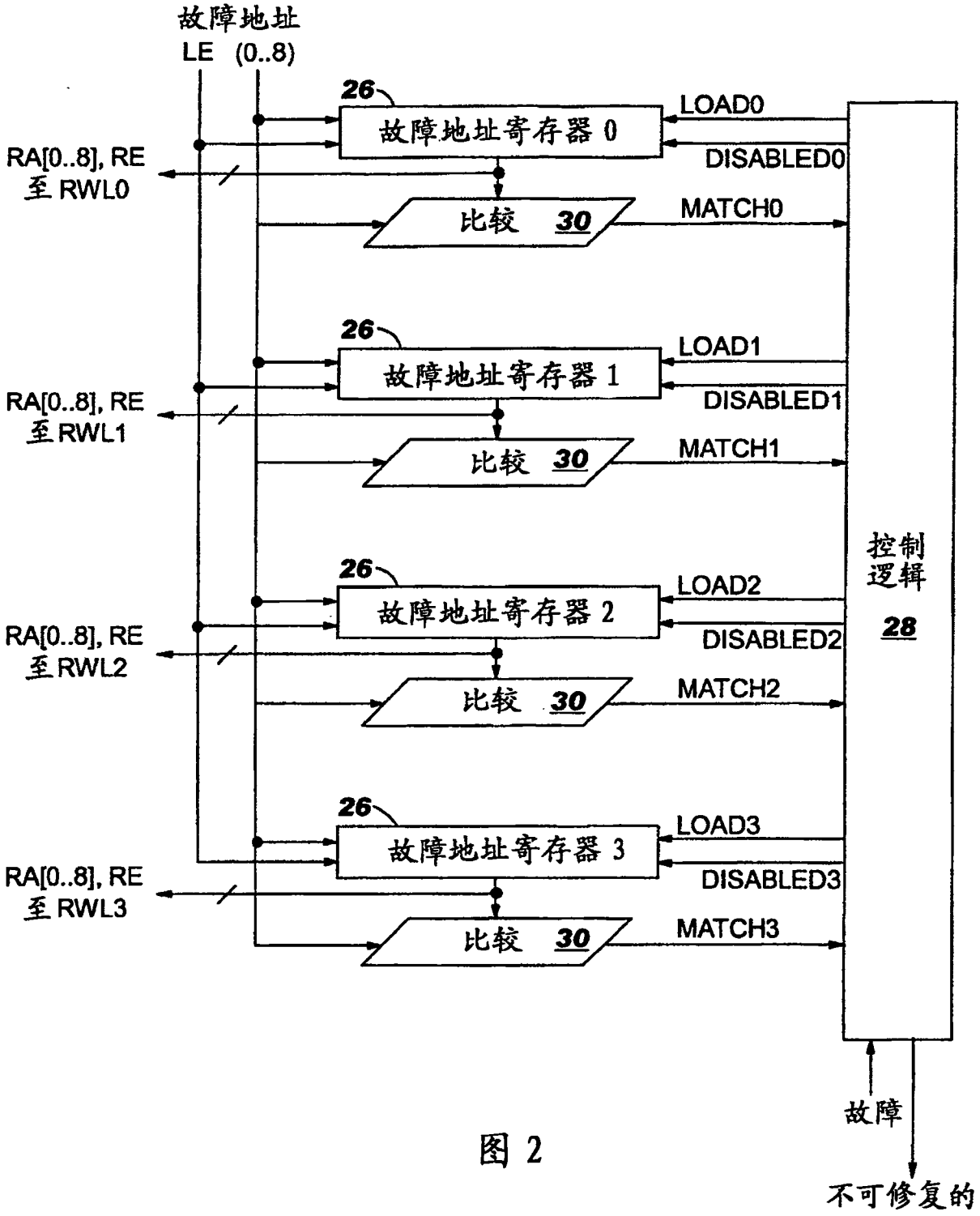

故障地址
LE (0..8)
26
故障地址寄存器 0
LOAD0
DISABLED0
RA[0..8], RE
至 RWL0
比较 30
MATCH0
26
故障地址寄存器 1
LOAD1
DISABLED1
RA[0..8], RE
至 RWL1
比较 30
MATCH1
控制
逻辑
28
26
故障地址寄存器 2
LOAD2
DISABLED2
RA[0..8], RE
至 RWL2
比较 30
MATCH2
26
故障地址寄存器 3
LOAD3
DISABLED3
RA[0..8], RE
至 RWL3
比较 30
MATCH3
故障
不可修复的

图 2

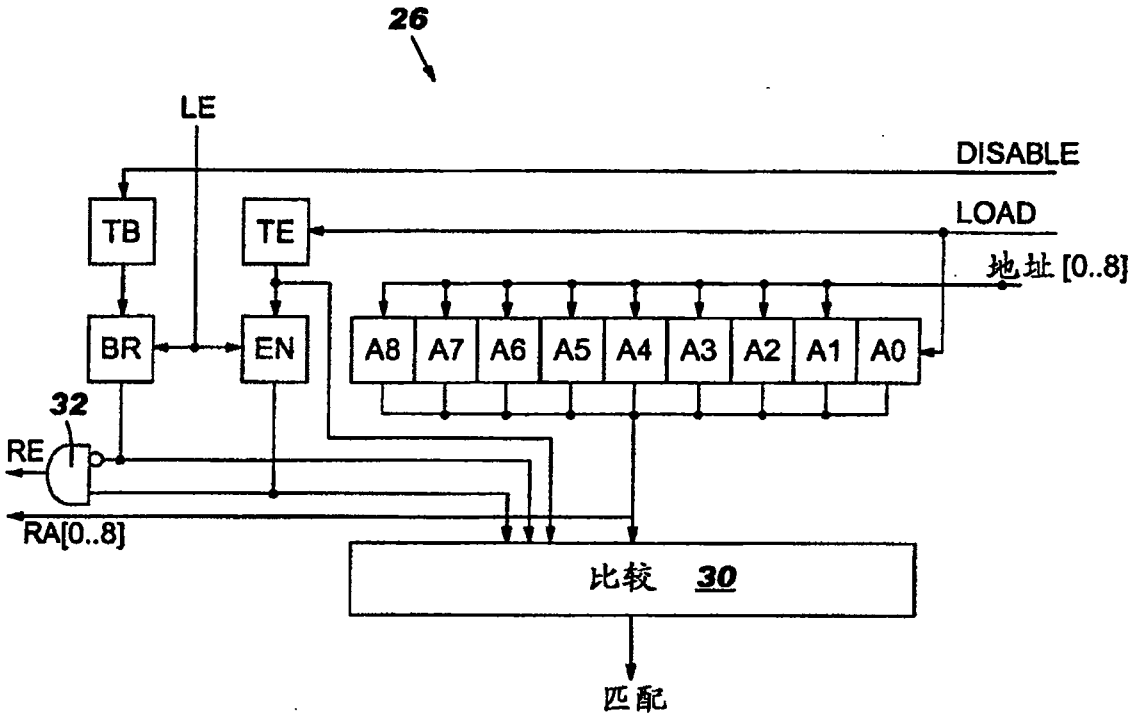

26
LE
DISABLE
LOAD
TB
TE
地址 [0..8]
BR
EN
A8
A7
A6
A5
A4
A3
A2
A1
A0
32
RE
RA[0..8]
比较 30
匹配

图 3

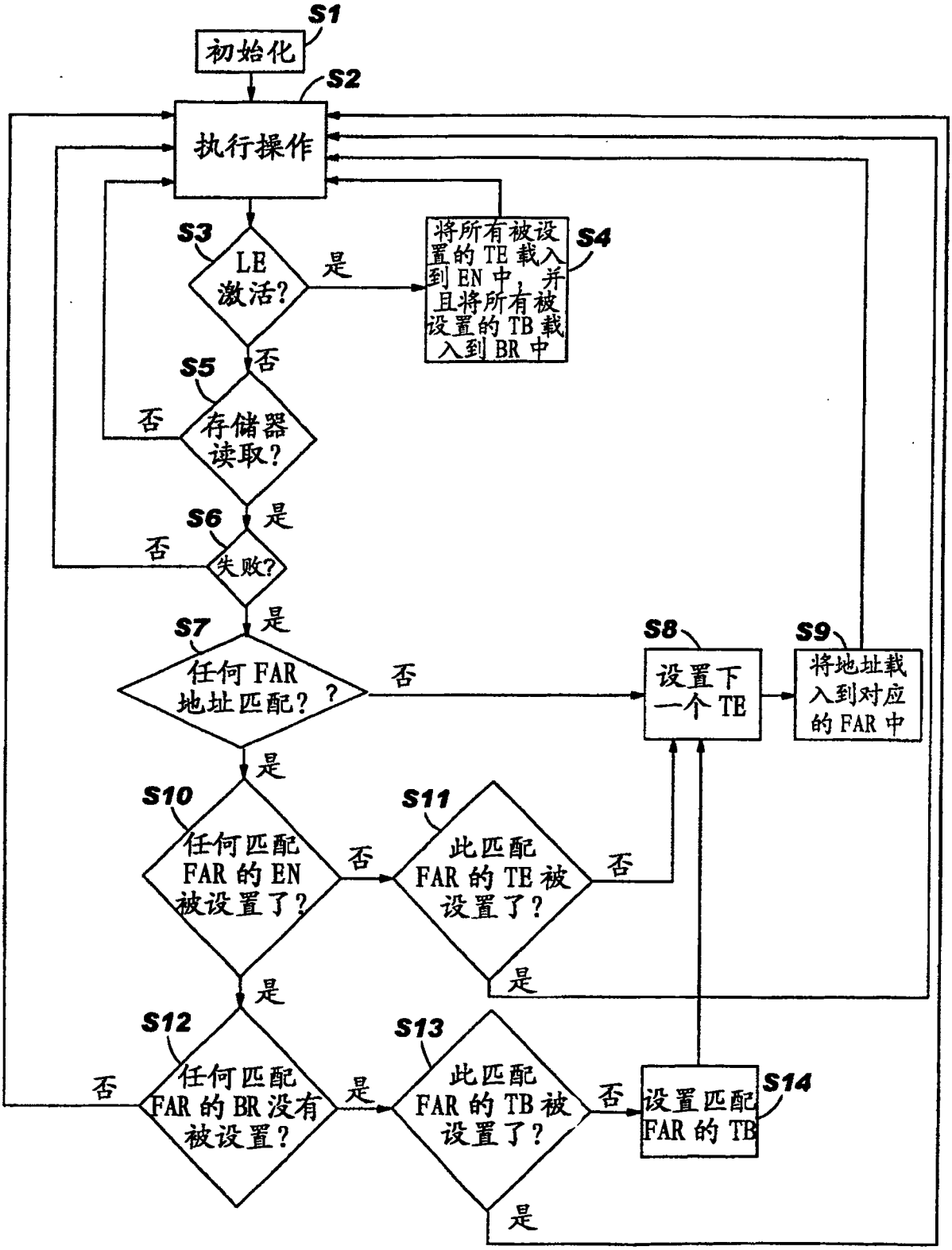

S1
初始化
S2
执行操作
S3
LE 激活?
是
S4
将所有被设置的 TE 载入到 EN 中，并且将所有被设置的 TB 载入到 BR 中
否
S5
存储器读取?
否
是
S6
失败?
否
是
S7
任何 FAR 地址匹配？?
否
S8
设置下一个 TE
S9
将地址载入到对应的 FAR 中
是
S10
任何匹配 FAR 的 EN 被设置了?
否
S11
此匹配 FAR 的 TE 被设置了?
否
是
是
S12
任何匹配 FAR 的 BR 没有被设置?
否
是
S13
此匹配 FAR 的 TB 被设置了?
否
S14
设置匹配 FAR 的 TB
是

图 4