



(12) 发明专利申请

(10) 申请公布号 CN 103077749 A

(43) 申请公布日 2013. 05. 01

(21) 申请号 201210566395. 2

(22) 申请日 2012. 12. 24

(71) 申请人 西安华芯半导体有限公司
地址 710055 陕西省西安市高新6路38号腾
飞创新中心A座4层

(72) 发明人 熊保玉 拜福君

(74) 专利代理机构 西安西交通盛知识产权代理
有限责任公司 61217

代理人 田洲

(51) Int. Cl.
G11C 29/44 (2006. 01)

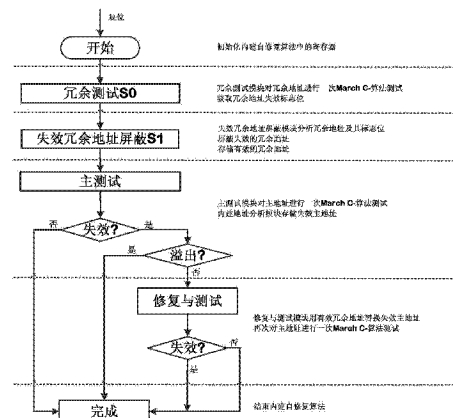
权利要求书1页 说明书5页 附图5页

(54) 发明名称

一种适用于静态随机存储器的冗余容错内建自修复方法

(57) 摘要

本发明提供一种适用于静态随机存储器的冗余容错内建自修复方法,该方法对传统的内建自修复算法提出了改进,加入了冗余测试模块和失效冗余地址屏蔽模块,解决了传统内建自修复算法在冗余地址失效时,无法正确修复的缺陷。冗余测试模块先对冗余地址测试,如果被测试的冗余地址失效,则将其失效标志位置为1,默认为0。待冗余测试结束时,每一个冗余地址都得到它的失效标志位。此时失效冗余地址屏蔽模块工作,对冗余地址进行分析,如果该冗余地址失效标志位为0,则将该冗余地址存入有效冗余地址寄存器内,否则,将其屏蔽掉,不存入。经过对失效的冗余地址屏蔽后,就可以保证自修复地址替换时,每一个被替换的冗余地址都是有效的。



1. 一种适用于静态随机存储器的冗余容错内建自修复方法,其特征在于,包括以下步骤:

- 1)、首先,进行复位操作,对 SRAM 内建自修复算法中所有的寄存器初始化;
- 2)、进入冗余测试,对 SRAM 冗余地址执行 March C- 算法;获取冗余地址失效标志位;
- 3)、分析冗余地址的失效标志位,屏蔽失效的冗余地址,存储有效的冗余地址。

2. 如权利要求 1 所述的一种适用于静态随机存储器的冗余容错内建自修复方法,其特征在于,所述方法还包括以下步骤:

4)、进行主测试,对 SRAM 主地址执行 March C- 算法;在主测试中,存储检测到的失效主地址,并计数;没有失效,则结束测试;有失效,并且溢出,则结束测试;有失效,但没有溢出,则进入修复和测试;

5)、在修复和测试中,用有效冗余地址替换失效主地址,再次对主地址执行一次 March C- 算法;如果有错误,则提示有错误,测试结束;如果没有错误,则提示没有错误,测试结束。

3. 如权利要求 1 所述的一种适用于静态随机存储器的冗余容错内建自修复方法,其特征在于,步骤 2) 具体包括以下步骤:对 SRAM 冗余地址执行 MarchC- 算法,对每个冗余地址进行测试,根据测试结果判断其是否失效,如果失效,则将该地址的失效标志位置为 1,否则置为 0;然后,判断 March C- 算法是否结束,如果结束,则冗余测试结束,否则,跳转到下一个地址,对下一个地址进行测试,直到 March C- 算法结束。

4. 如权利要求 1 所述的一种适用于静态随机存储器的冗余容错内建自修复方法,其特征在于,步骤 3) 具体包括以下步骤:判断当前冗余地址是否小于冗余地址深度,如果不小于,则将失效冗余地址屏蔽完成标志位置 1,失效冗余地址屏蔽结束;否则,将失效冗余地址屏蔽完成标志位置 0,并检测该冗余地址失效标志位是否为 0,如果是 0,则将该冗余地址存入有效冗余地址寄存器中,有效冗余地址索引号 i 加 1,冗余地址加 1;如果该冗余地址失效标志位为 1,则冗余地址直接加 1;无论当前冗余地址是否失效,都进入对下一个冗余地址的分析,直到对所有的冗余地址分析完成后,结束。

5. 如权利要求 1 所述的一种适用于静态随机存储器的冗余容错内建自修复方法,其特征在于,步骤 3) 中对冗余地址进行分析,如果该冗余地址失效标志位为 0,则将该冗余地址存入有效冗余地址寄存器内,否则,将其屏蔽掉,不存入。

一种适用于静态随机存储器的冗余容错内建自修复方法

【技术领域】

[0001] 本发明涉及存储器的内建自修复算法领域,特别涉及一种适用于静态随机存储器的冗余容错内建自修复方法。

【背景技术】

[0002] 嵌入式静态随机存储器(SRAM)由于其高速性能而被广泛用于SOC中。根据国际半导体技术蓝图(ITRS)预测,SOC系统中嵌入式SRAM所占面积比例将不断变大,到2014年将提升到94%。

[0003] 由于工艺缺陷等原因,导致SRAM的良率降低。SRAM内建自修复(BISR)是一种有效的修复方法。BISR的自修复功能,是通过将SRAM的一部分主地址作为冗余地址使用,当主地址失效时,BISR使用冗余地址替换失效地址来实现的。传统的BISR直接对主地址进行测试,如果主地址发生失效,即通过冗余地址替换来达到修复的目的。然而,当冗余本身出现失效时,通过将常规的发生错误的地址用冗余地址替换,并不能修复失效,因为被替换的冗余地址,仍然是失效的。因此,传统的内建自修复算法存在缺陷,对SRAM的良率提升是有限的。

[0004] 因此,设计一种对冗余具有容错功能的内建自修复算法,对提高嵌入式SRAM的良率,是十分有意义的。

【发明内容】

[0005] 本发明的目的在于提出一种适用于静态随机存储器的冗余容错内建自修复方法,以解决上述技术问题。本发明方法对传统的内建自修复算法提出了改进,加入了冗余测试模块和失效冗余地址屏蔽模块,解决了传统内建自修复算法在冗余地址失效时,无法正确修复的缺陷。

[0006] 为了实现上述目的,本发明采用如下技术方案:

[0007] 一种适用于静态随机存储器的冗余容错内建自修复方法,包括以下步骤:

[0008] 1)、首先,进行复位操作,对SRAM内建自修复算法中所有的寄存器初始化;

[0009] 2)、进入冗余测试,对SRAM冗余地址执行March C-算法;获取冗余地址失效标志位。

[0010] 3)、分析冗余地址的失效标志位,屏蔽失效的冗余地址,存储有效的冗余地址。

[0011] 本发明进一步的改进在于:

[0012] 4)、进行主测试,对SRAM主地址执行March C-算法;在主测试中,存储检测到的失效主地址,并计数;没有失效,则结束测试;有失效,并且溢出,则结束测试;有失效,但没有溢出,则进入修复和测试;

[0013] 5)、在修复和测试中,用有效冗余地址替换失效主地址,再次对主地址执行一次March C-算法;如果有错误,则提示有错误,测试结束;如果没有错误,则提示没有错误,测试结束。

[0014] 本发明进一步的改进在于：步骤 2) 具体包括以下步骤：对 SRAM 冗余地址执行 March C- 算法，对每个冗余地址进行测试，根据测试结果判断其是否失效，如果失效，则将该地址的失效标志位置为 1，否则置为 0；然后，判断 March C- 算法是否结束，如果结束，则冗余测试结束，否则，跳转到下一个地址，对下一个地址进行测试，直到 March C- 算法结束。

[0015] 本发明进一步的改进在于：步骤 3) 具体包括以下步骤：判断当前冗余地址是否小于冗余地址深度，如果不小于，则将失效冗余地址屏蔽完成标志位置 1，失效冗余地址屏蔽结束；否则，将失效冗余地址屏蔽完成标志位置 0，并检测该冗余地址失效标志位是否为 0，如果是 0，则将该冗余地址存入有效冗余地址寄存器中，有效冗余地址索引号 i 加 1，冗余地址加 1；如果该冗余地址失效标志位为 1，则冗余地址直接加 1；无论当前冗余地址是否失效，都进入对下一个冗余地址的分析，直到对所有的冗余地址分析完成后，结束。

[0016] 本发明进一步的改进在于：步骤 3) 中对冗余地址进行分析，如果该冗余地址失效标志位为 0，则将该冗余地址存入有效冗余地址寄存器内，否则，将其屏蔽掉，不存入。

[0017] 相对于现有技术，本发明具有以下优点：本发明提供一种适用于静态随机存储器的冗余容错内建自修复方法，该方法对传统的内建自修复算法提出了改进，加入了冗余测试模块和失效冗余地址屏蔽模块，解决了传统内建自修复算法在冗余地址失效时，无法正确修复的缺陷。冗余测试模块先对冗余地址测试，如果被测试的冗余地址失效，则将其失效标志位置为 1，默认为 0。待冗余测试结束时，每一个冗余地址都得到它的失效标志位。此时失效冗余地址屏蔽模块工作，对冗余地址进行分析，如果该冗余地址失效标志位为 0，则将该冗余地址存入有效冗余地址寄存器内，否则，将其屏蔽掉，不存入。经过对失效的冗余地址屏蔽后，就可以保证自修复地址替换时，每一个被替换的冗余地址都是有效的。

【附图说明】

[0018] 图 1 为根据本发明实施的一个冗余容错内建自修复方法的实例流程图。

[0019] 图 2 为冗余测试的流程图。

[0020] 图 3 为失效冗余地址屏蔽的流程图。

[0021] 图 4 为所示实例的硬件实现的框图。

[0022] 图 5 为所示实例的硬件实现的有限自由状态机。

【具体实施方式】

[0023] 下面结合附图对本发明的实施方式做进一步描述。

[0024] 请参阅图 1 所示，图 1 为根据本发明实施的一个冗余容错内建自修复方法实例。本发明方法包括以下步骤：

[0025] 1)、首先，进行复位操作，对 SRAM 内建自修复算法中所有的寄存器初始化；

[0026] 2)、然后，进入冗余测试，冗余测试模块对 SRAM 冗余地址执行 March C- 算法。在冗余测试中，获取冗余地址失效标志位。

[0027] 3)、然后，进入失效冗余地址屏蔽，失效冗余地址屏蔽模块通过分析冗余地址的失效标志位，屏蔽失效的冗余地址，存储有效的冗余地址。

[0028] 4)、然后，进入主测试，主测试模块对 SRAM 主地址执行 March C- 算法。在主测试中，存储检测到的失效主地址，并计数。没有失效，则结束测试。有失效，并且溢出，则结束

测试。有失效,但没有溢出,则进入修复和测试。

[0029] 5)、在修复和测试中,修复与测试模块用有效冗余地址替换失效主地址,再次对主地址执行一次 March C- 算法。如果有错误,则提示有错误,测试结束。如果没有错误,则提示没有错误,测试结束。

[0030] 请参阅图 2 所示,图 2 为冗余测试的流程图。具体步骤如下:首先,进行复位操作,对所有的寄存器初始化,算法开始。然后,对 SRAM 冗余地址执行 MarchC- 算法,对每个冗余地址进行测试,根据测试结果判断其是否失效,如果失效,则将该地址的失效标志位置为 1,否则置为 0。然后,判断 March C- 算法是否结束,如果结束,则冗余测试结束,否则,跳转到下一个地址,对下一个地址进行测试,直到 March C- 算法结束。当冗余测试结束时,每一位冗余地址都得到一位的标志位,为下一步失效冗余地址屏蔽做好准备。

[0031] 请参阅图 3 所示,图 3 为失效冗余地址屏蔽的流程图。具体步骤如下,首先,进行复位操作,对所有的寄存器初始化,算法开始。然后,判断当前冗余地址是否小于冗余地址深度,如果不小于,则将失效冗余地址屏蔽完成标志位置 1,失效冗余地址屏蔽结束;否则,将失效冗余地址屏蔽完成标志位置 0,并检测该冗余地址失效标志位是否为 0,如果是 0,则将该冗余地址存入有效冗余地址寄存器中,有效冗余地址索引号 i (复位时为 0)加 1,冗余地址(复位时为 0)加 1;如果该冗余地址失效标志位为 1,则冗余地址直接加 1。无论当前冗余地址是否失效,都进入对下一个冗余地址的分析,直到对所有的冗余地址分析完成后,结束。

[0032] 请参阅图 4 所示,图 4 为所示实例的硬件实现的框图。图中 BISR 表示冗余容错内建自修复方法的硬件,SRAM 表示待测静态随机存储器。箭头方向表示数据流向。BISR 由冗余测试、失效冗余地址屏蔽、主测试、有限自由状态机,内建地址分析和 3 选 1 选择器组成。BISR 的输入信号包括测试信号 $test_h$ 、复位信号 rst_1 、修复信号 $bisr_h$ 、时钟信号 clk 、SRAM 原始片选信号 cen_0 、原始写使能信号 wen_0 、原始地址信号 a_0 、原始写数据 d_0 、读数据 q ;输出信号包括测试完成信号 $test_done$ 、溢出信号 $over_h$ 、失效信号 $fail_h$ 、SRAM 片选信号 cen 、写使能信号 wen ,地址信号 a 和写数据 d 。冗余测试输入包括 SRAM 读数据 q 、 $test_phase$ 、 $test_h$ 、 rst_1 、 clk ;它为失效冗余地址屏蔽提供冗余地址失效标志位 $redun_a_flag$,为有限自由状态机提供冗余测试结束标志位 $test_redun_done$,为 3 选 1 选择器提供冗余片选信号 cen_redun 、冗余写使能信号 wen_redun 、冗余地址信号 a_redun 、冗余数据信号 d_redun 。失效冗余地址屏蔽输入包括 SRAM 读数据 q 、 $test_phase$ 、 $test_h$ 、 rst_1 、 clk ;它为有限自由状态机提供有效冗余地址数量 $ava_redun_a_num$ 、失效冗余地址屏蔽完成标志位 $def_redun_a_mask_done$,为内建地址分析提供有效冗余地址。主测试输入包括 $test_phase$ 、 $test_h$ 、 rst_1 、 clk ;它为有限自由状态机提供主测试完成标志位 $test_main_done$ 、主测试失效标志位 $fail_h_main$ 、修复测试完成标志位 $test_repair_done$ 、修复测试失效标志位 $fail_h_repair$ 和失效主地址数量 $def_main_a_num$;为内建地址分析提供失效主地址 def_main_a ;为 3 选 1 选择器提供主片选择信号 cen_main 、主写使能信号 wen_main ,主冗余地址 a_main 和主写数据 d_main 。有限自由状态机输入包括 $test_h$ 、 rst_1 、 clk 、 $bisr_h$ 、 $test_redun_done$ 、 $ava_redun_a_num$ 、 $def_redun_a_mask_done$ 、 $test_main_done$ 、 $fail_h_main$ 、 $test_repair_done$ 、 $fail_h_repair$ 、 $def_main_a_num$;为其他五个模块提供 $test_phase$,为整个 BISR 输出 $over_h$ 、 $fail_h$ 和 $test_done$ 。3 选 1 选择器的输入包括 $bisr_h$ 、

test_phase、cen_0、wen_0、a_0、d_0、cen_redun、wen_redun、a_redun、d_redun、cen_main、wen_main、a_main、d_main；它为内建地址分析提供内建自修复地址 bisr_a；为 SRAM 提供 cen、wen、d。内建地址分析输入包括 test_h、rst_1、clk、bistr_h、test_phase、ava_redun_a、def_main_a、bistr_a；为 SRAM 提供地址 a。BISR 工作原理将结合图 5 详细说明。

[0033] 请参阅图 5 所示，图 5 为所示实例的硬件实现的有限自由状态机。结合图 4 详细描述冗余容错的内建自修复方法硬件实现的工作原理如下。当复位信号 rst_1=0 时，所有状态跳都将转到“测试开始 test_start”状态，“测试开始”状态为该状态机的复位状态。在“测试开始”状态时，如果测试信号 test_h=1，则 test_phase 的下一个状态为“冗余测试”test_redun 状态。

[0034] 在“冗余测试”状态中，对 SRAM 冗余地址进行 March C- 算法。其原理如下，冗余测试产生 cen_redun、wen_redun、a_redun、d_redun；3 选 1 选择器检测到当前 test_phase=test_redun，将 cen_redun、wen_redun、d_redun 选通给 SRAM 输入 cen、wen、d，将 a_redun 输出给内建地址分析输入 bisr_a，内建地址分析直接将 bisr_a 送给 SRAM 输入 a。SRAM 将读数据 q 反馈给冗余测试，如果 q 与预期值相同，则该冗余地址有效，否则失效，将该冗余地址对应的失效标志位 redun_a_flag 标记为 1（复位时为 0）。执行完对冗余地址的 March C- 算法后，将冗余测试完成标志位 test_redun_done 置为 1（复位时为 0）。在“冗余测试”状态中，如果 test_h=1 且 test_redun_done=1，则 test_phase 的下一状态为“失效冗余地址屏蔽”def_redun_a_mask 状态。

[0035] 在“失效冗余地址屏蔽”状态中，对每一个冗余地址进行分析，如果其失效标志位为 0，则将其存入有效冗余地址寄存器 ava_redun_a 中，且将有效冗余地址数量 ava_redun_a_num 加 1；如果其失效标志位为 1，则将该冗余地址丢弃掉，不存入 ava_redun_a 中。在完成对所有冗余地址的分析之后，将失效冗余地址屏蔽完成标志位 def_redun_a_mask_done 置为 1（复位时为 0）。在“失效冗余地址屏蔽”状态中，如果 test_h=1 且 def_redun_a_mask_done=1，则 test_phase 的下一状态为“主测试”test_main 状态。

[0036] 在“主测试”状态中，对 SRAM 主地址执行 March C- 算法。其原理如下，主测试产生 cen_main、wen_main、a_main、d_main；3 选 1 选择器检测到当前 test_phase=test_main，将 cen_main、wen_main、d_main 选通给 SRAM 输入 cen、wen、d，将 a_main 输出给内建地址分析输入 bisr_a，内建地址分析直接将 bisr_a 送给 SRAM 输入 a。SRAM 将读数据 q 反馈给主测试，如果 q 与预期值不相同，则主测试发生失效，将主测试错误标志位 fail_h_main 置为 1（复位时为 0），并把失效主地址存入失效主地址寄存器 def_main_a 中，并且将失效主地址数量 def_main_a_num 加 1。执行完对主地址的 March C- 算法后，将主测试完成标志位 test_main_done 置为 1（复位时为 0）。如果发生错误的主地址数量 def_main_a_num 大于有效冗余地址数量 ava_redun_a_num，则将溢出标志位 over_h 置为 1（复位时为 0）。

[0037] 在“主测试”状态中，如果 test_h=1 且 test_main_done=1，并且 fail_h_main=0，则 fail_h=0，test_done=0，表示主测试完成，没有错误，则 test_phase 的下一状态为“测试完成”状态。在“主测试”状态中，如果 test_h=1 且 test_main_done=1，且 fail_h_main=1，且 over_h=1，表示主测试完成，失效的主地址数量大于有效冗余地址的数量，溢出，不可修复，则 test_phase 的下一状态为“测试完成”状态。在“主测试”状态中，如果 test_h=1 且 test_main_done=1，且 fail_h_main=1 且 over_h=0，且 bisr_h=1，表示主测试完成，有错误，

但失效的主地址数量小于有效冗余地址的数量,可修复,并且进行修复,则 test_phase 的下一状态为“修复和测试” test_repair 状态。

[0038] 在“修复和测试”状态中,对 SRAM 主地址进行修复(如果有需要)并执行 March C- 算法。其原理如下,主测试产生 cen_main、wen_main、a_main、d_main;3 选 1 选择器检测到当前 test_phase=test_repair,将 cen_main、wen_main、d_main 选通给 SRAM 输入 cen、wen、d,将 a_main 输出给内建地址分析输入 bisr_a。内建地址分析测到当前 test_phase=test_repair,将 bisr_a 与 def_main_a 寄存器中的每一个地址进行比较,如果命中,则用 ava_redun_a 中有效冗余地址替换 bisr_a 后送给 SRAM 输入 a;如果不命中,直接将 bisr_a 送给 SRAM 输入 a。SRAM 将读数据 q 反馈给主测试,如果 q 与预期值不相同,则修复和测试发生失效,将修复和测试错误标志位 fail_h_repair 置为 1 (复位时为 0)。执行完对主地址的修复和测试后,将修复和测试完成状态标志位 test_repair_done 置为 1 (复位时为 0)。

[0039] 在“修复和测试”状态中,如果 test_h=1 且 test_repair_done=1 且 fail_h_repair=0,则 fail_h=0, test_done=0,表示修复成功,提示修复成功,修复完成,则 test_phase 的下一状态为“测试完成”状态;如果 test_h=1 且 test_repair_done=1 且 fail_h_repair=1,则 fail_h=1, test_done=0 表示修复失败,提示修复失败,修复完成,则 test_phase 的下一状态为“测试完成”状态。

[0040] 在“测试完成”状态,如果 rst_l=0,则 test_phase 的下一状态为“测试开始 test_start”状态。

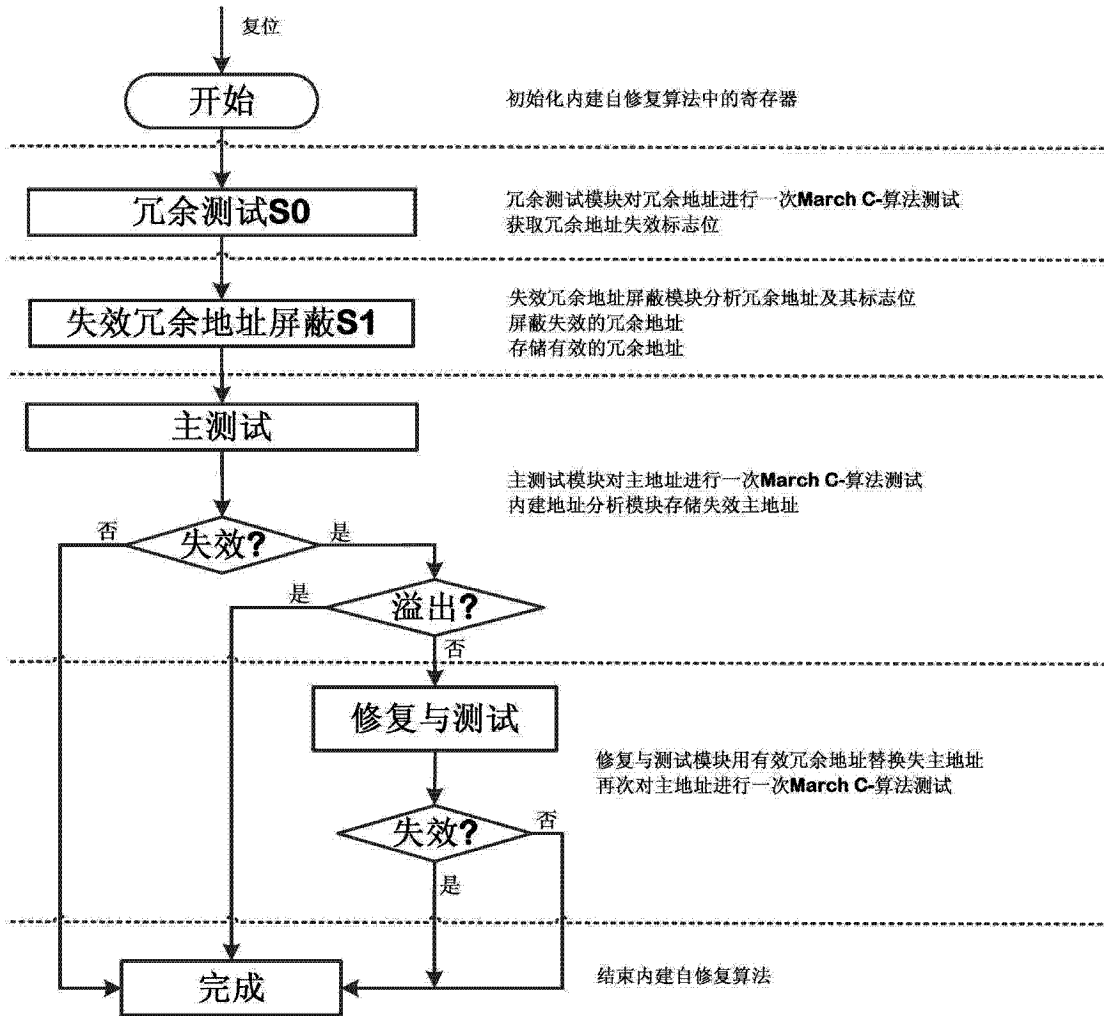


图 1

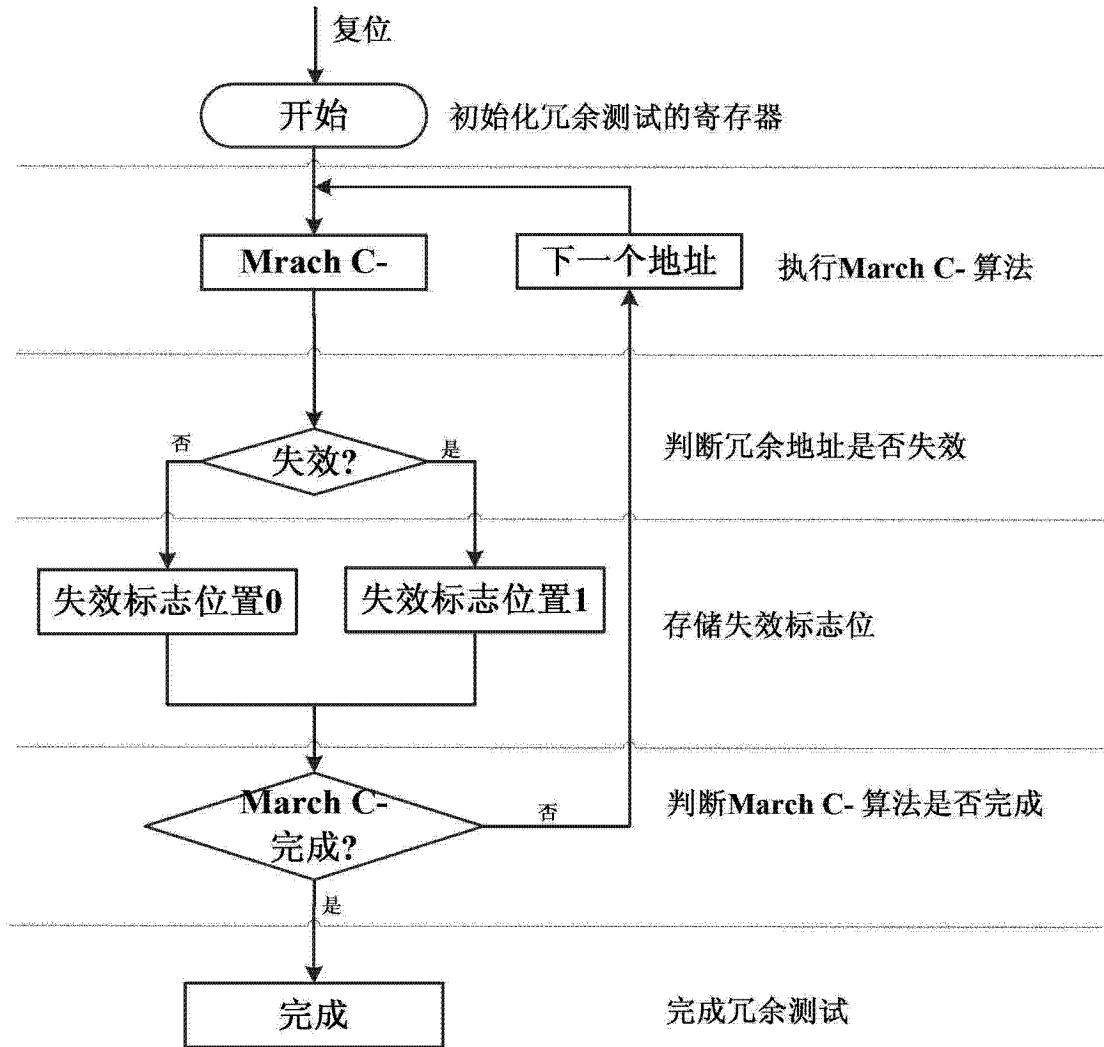


图 2

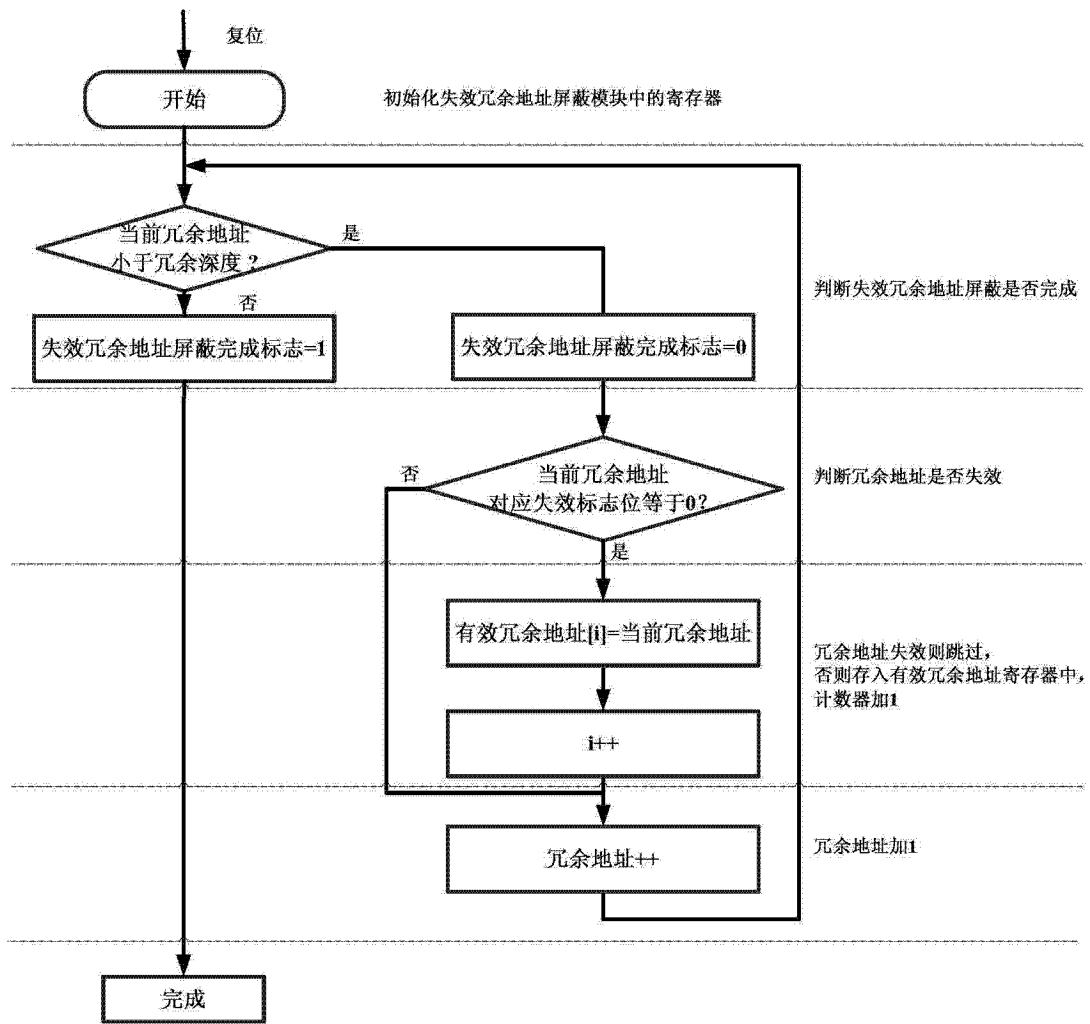


图 3

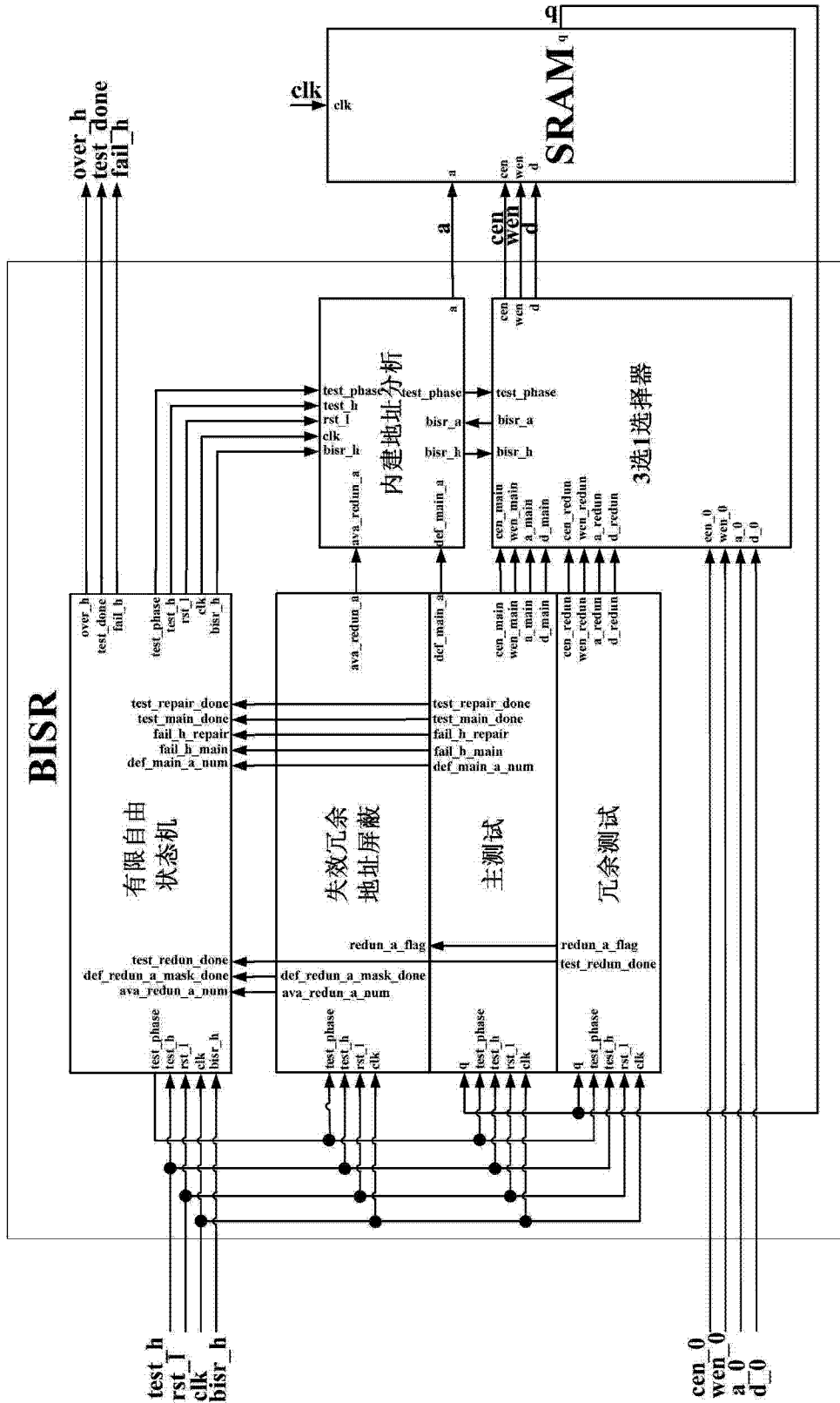


图 4

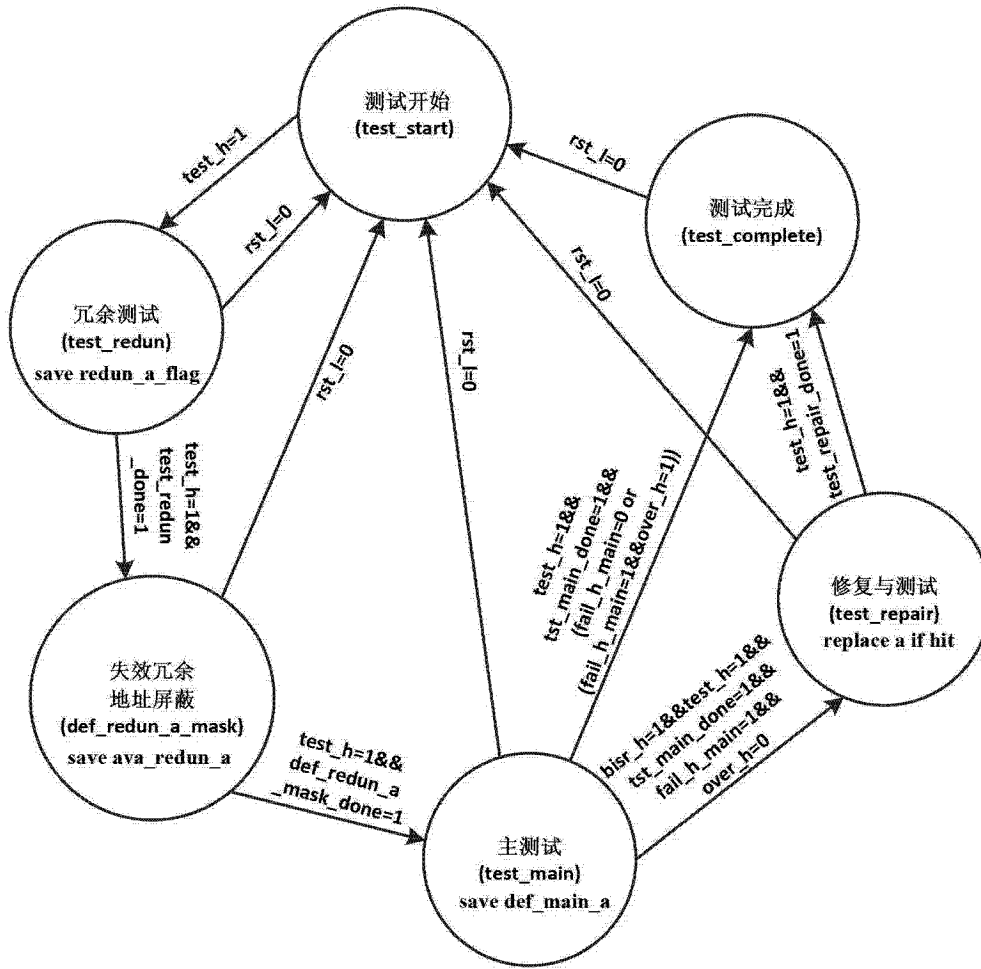


图 5