

大规模芯片内嵌存储器的 BIST 测试方法研究

葛云侠¹ 陈龙¹ 解维坤^{1,2} 张凯虹³ 宋国栋¹ 奚留华³

(1. 中国电子科技集团公司第58研究所 无锡 214035; 2. 电子科技大学自动化学院 成都 611731;
3. 无锡中微腾芯电子有限公司 无锡 214028)

摘要:随着大规模芯片的块存储器(block random access memory, BRAM)数量不断增多,常见的存储器内建自测试(memory build-in-self test, Mbist)方法存在故障覆盖率低、灵活性差等问题。为此,提出了一种新的基于可编程有限状态机的 Mbist 方法,通过3个计数器驱动的可编程 Mbist 控制模块和算法模块集成8种测试算法,提高故障覆盖率和灵活性。采用 Verilog 语言设计了所提出的 Mbist 电路,通过 Modelsim 对 1 Kbit×36 的 BRAM 进行仿真并在自动化测试系统上进行了实际测试。实验结果表明,该方法对 BRAM 进行测试能够准确定位故障位置,故障的检测率提高了 15.625%,测试效率提高了 26.1%,灵活性差的问题也得到了很大改善。

关键词:大规模芯片;块存储器;存储器内建自测试;可编程存储器内建自测试控制器;故障覆盖率

中图分类号: TN4 **文献标识码:** A **国家标准学科分类代码:** 510.3010

Research on BIST testing method for large-scale chip embedded memory

Ge Yunxia¹ Chen Long¹ Xie Weikun^{1,2} Zhang Kaihong³ Song Guodong¹ Xi Liuhua³

(1. Technology Group Corporation No. 58 Research Institute, Wuxi 214035, China;
2. School of Automation Engineering, University of Electronic Science and Technology of China,
Chengdu 611731, China; 3. Zhongwei Tengxin Electronics Co. Ltd., Wuxi 214028, China)

Abstract: With the increasing number of block random access memory (BRAM) in large-scale chip, common memory built-in self-test (Mbist) methods have some problems such as low fault coverage, poor flexibility etc. Therefore, in this paper a new Mbist method based on programmable finite state machine is proposed. The method is to integrate eight test algorithms through three counter-driven programmable Mbist control modules and algorithm modules to improve fault coverage and flexibility. The proposed Mbist circuit is designed in Verilog language, and the 1 Kbit×36 BRAM is simulated by Modelsim, and conducted actual testing on an automated testing system. The experimental results show that the method can accurately locate the fault location when testing the BRAM, the fault detection rate is increased by 15.625%, the test efficiency has been improved by 26.1% and the problems of poor flexibility are also greatly improved.

Keywords: large-scale chip; block RAM; memory built-in self-test; programmable memory built-in self-test controller; fault coverage

0 引言

随着半导体技术的进步,集成电路的规模在不断扩大,与其相关的数字系统也变得越来越复杂^[1-2]。在目前的大规模集成电路产品中,为缩短设计周期,降低设计成本,使基于知识产权(intellectual property, IP)核复用的(system-on-chip, SOC)技术成为大规模集成电路设计技

术的主流之一,嵌入的 IP 核所占的空间变得越来越大,测试难度大幅提升,如何高效率实现对 IP 核的测试成为目前研究的热点^[3-4]。块存储器(block random access memory, BRAM)内嵌于大规模集成电路芯片中,使大规模芯片具有快速、高效和可靠地传输数据和缓存功能,BRAM 性能的好坏会影响整个大规模芯片的性能^[5]。因此对大规模芯片中 BRAM 进行高覆盖率测试,是确保芯片正常

工作的途径之一,目前对存储器测试最为有效的测试方法是存储器内建自测试(memory build-in-self test, Mbist)^[6]技术。Mbist技术是指在存储器周围建立自测试电路^[7]。Mbist电路结构主要包括控制器模块、地址生成模块、向量生成模块、比较分析模块^[8],而且Mbist控制器设计是Mbist电路设计的核心。目前Mbist控制器设计主要有3种技术,基于有限状态机(finite state machine, FSM)的Mbist^[9]、基于微码的Mbist^[10]以及基于微处理器的Mbist^[11]。基于FSM的Mbist控制器是以FSM形式对单一测试算法进行硬件实现,将测试算法固定到芯片中,若需实现新算法或更改现有算法就需要更改Mbist电路设计,因此该方法的灵活性较低。基于微码Mbist控制器是具有专用架构的可编程控制器,其中许多测试模式都是以预定义指令的形式编写。基于微码的控制器在实现多种算法或新算法时不需要改变现有Mbist电路,因此具有一定的灵活性。但这种灵活性是以增加面积开销为代价的,使用基于微码的Mbist不适合于大规模芯片的测试^[12]。基于微处理器的Mbist电路是根据测试算法通过控制微处理器生成测试向量的^[13],测试时间长、可控性和可观察性差。因此,需要为大规模芯片内嵌存储器设计一种能提高灵活性且故障覆盖率高的可编程Mbist。为此,本文提出一种新的基于可编程FSM的Mbist方法,该方法是由三个计数器驱动的可编程Mbist控制模块、算法模块和比较分析模块组成,能够集成八种测试算法来测试BRAM。若需要更换测试算法时,所提出的测试方法不需重新设置Mbist电路,有效地减少了人力和配置时间从而降低测试成本,故障覆盖率和灵活性都有所提高,对BRAM测试的研究有着重要的应用价值和参考意义。

1 存储器内建自测试技术

1.1 Mbist 常见电路结构

根据测试算法在BRAM周围建立自测试电路,控制BRAM的输入输出数据,把BRAM的输出数据在比较器中与期望值进行比较,如果不同则表明BRAM存在故障,

反之BRAM不存在故障^[14]。Mbist电路通常包括4个模块:测试向量生成模块、Mbist控制模块、地址生成模块以及分析比较模块,BRAM常见Mbist的电路结构如图1所示。

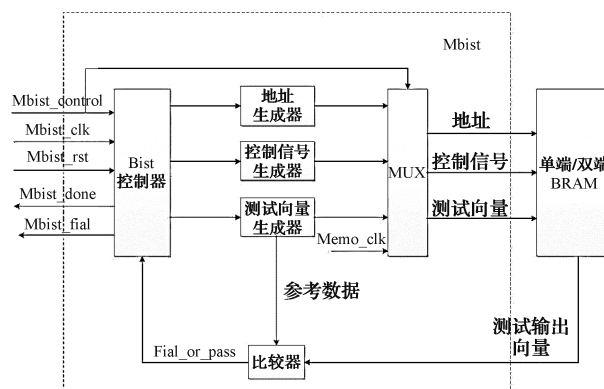


图1 常见的Mbist电路结构

Fig.1 Common Mbist circuit structures

1.2 测试算法

March算法是由一系列March元素组成,通过对BRAM的每个地址进行读取或写入操作来检测BRAM是否存在故障。March算法的复杂度与存储单元数量 N 成正比,测试元素具有对称性和规则性,具有测试时间较短和故障覆盖率较高等优点,在实际应用中March算法目前已成为存储器测试的最有效测试算法^[15]。因此,很多学者对March算法进行了研究,针对不同的故障覆盖需求提出了许多March算法的改进算法,不同的测试算法如表1所示,其中 $\downarrow$ 表示操作方向是不关心,根据对称性选择方向; $\uparrow$ 表示必须从最低地址到最高地址进行操作; $\downarrow$ 表示必须从最高地址到最低地址进行操作; $w0$ 表示写“0”操作; $w1$ 表示写“1”操作; $r0$ 表示读“0”操作; $r1$ 表示读“1”操作; $M0\sim M5$ 表示第1~5个测试元素;“:”两边表示对两个端口进行操作;“ n ”表示任意操作;“-”表示不进行操作。

表1 March 算法
Table 1 March algorithm

序号	算法名称	测试元素						复杂度
		M0	M1	M2	M3	M4	M5	
0	March X	$\downarrow(w0)$	$\uparrow(r0, w1)$	$\uparrow(r1, w0, r0)$	$\downarrow(r0, w1, r1)$	$\downarrow(r1, w0)$	$\downarrow(r0)$	$6N$
1	March C+	$\downarrow(w0)$	$\uparrow(r0, w1, r1)$	$\uparrow(r1, w0)$	$\downarrow(r0, w1)$	$\downarrow(r1, w0, r0)$	$\downarrow(r0)$	$14N$
2	March C-	$\downarrow(w0)$	$\uparrow(r0, w1)$			$\downarrow(r1, w0)$	$\downarrow(r0)$	$10N$
3	March Y	$\downarrow(w0)$	$\uparrow(r0, w1, r1)$	$\uparrow(r1, w0, w0, r0)$	$\downarrow(r0, w1, w1, r1)$	$\downarrow(r1, w0, r0)$	$\downarrow(r0)$	$8N$
4	March CG	$\downarrow(w0)$	$\uparrow(r0, w1, w1, r1)$	$\uparrow(r1, r1, w1, r1, w0)$	$\downarrow(r0, r0, w0, r0, w1)$	$\downarrow(r1, w0, w0, r0)$	$\downarrow(r0)$	$18N$
5	March SS	$\downarrow(w0)$	$\uparrow(r0, r0, w0, r0, w1)$	$\uparrow(r1, w0, r0, w1)$	$\downarrow(r1, w0)$	$\downarrow(r1, r1, w1, r1, w0)$	$\downarrow(r0)$	$22N$
6	March LR	$\downarrow(w0)$	$\uparrow(r0, w1)$	$\uparrow(R1: R1, R1: -, \downarrow(R0: R0, R0: -, \downarrow(r0, w1, r1, w0)$			$\downarrow(r0)$	$14N$
7	March s2pf	$\uparrow(W0: n)$	$\uparrow(R0: R0, R0: -, W0: R1, W1: R0,)$		$W1: R0,)$	$\downarrow(R1: R1, R1: -, \downarrow(R0: W0: R1)$	$\downarrow(R0: -)$	$14N$

不同的 March 算法具有不同的测试时间和故障覆盖率。例如, March X 只能检测固定故障(stuck at fault, SAF)、地址译码故障(address decoder fault, ADF)和部分转换故障(transition fault, TF)、耦合故障(coupling fault, CF),而 March SS 能够检测 SAF、ADF、TF、CF、读取破坏性故障和写入破坏性故障。不同 March 算法之间的故障覆盖率的比较如图 2 所示。

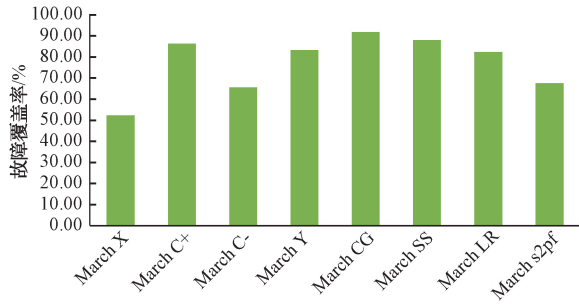


图 2 各 March 算法的故障覆盖率

Fig. 2 Fault coverage of each March algorithm

2 可编程 Mbist 电路结构设计

常见 Mbist 电路是根据所选择的单一测试算法进行设计的,如果需要更换其他测试算法对 BRAM 进行测试,则需要重新设计符合所选择测试算法的 Mbist 电路结构,所以灵活性较差。因此,本文提出一种新的基于可编程 FSM 的 Mbist 测试方法,以提高故障覆盖率和灵活性。所提出的可编程 Mbist 电路是由 3 个计数器驱动的可编程 Mbist 控制模块、算法模块和比较分析模块组成。可编程 Mbist 控制器模块采用 FSM 实现,如图 3 所示。可编程 Mbist 电路以 Mbist_clk、Mbist_rst、Mbist_control 和 select_cant 信号作为输入,以地址、读写控制信号、测试向

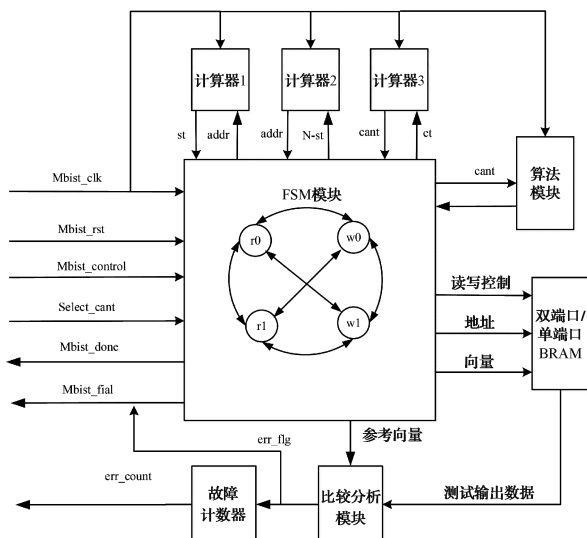


图 3 提出的可编程 Mbist 电路架构

Fig. 3 Proposed programmable Mbist circuit architecture

量、err_count、Mbist_done 以及 Mbist_fial 信号作为输出。

FSM 中有 r0、r1、w0 和 w1 状态,这些状态之间的转换取决于 4 个因素。

1) cant 的值,表示表 1 中测试算法的序号;

2) st,表示表 1 中的 M0—M5,用于执行测试算法中指定的 March 元素;

3) N-st,表示执行每个 March 元素中所有操作完成信号;

4) et,表示一个测试算法的所有 March 元素在存储器的所有地址执行完成信号。

3 个计数器用于生成合适的 st、cant 和 addr。由于 Mbist 为满足故障覆盖率需求,集成了 8 个测试算法,这 8 个测试算法中最多有 6 个测试元素,所以计数器 1 采用一个三位计数器生成有效的 st。使用计数器 2 生成一个十位计数器的地址(考虑一个 1 Kbit×36 的存储单元)。在 8 个测试算法中选取一个 March 算法,因此计数器 3 采用一个四位计数器来生成 cant。当每个 March 元素必须在一个地址上完成所有操作时,N-st 由 0 变成 1,计数器 2 递增输出 addr,然后 N-st 复位。当一个 March 元素的所有操作都在所有地址完成且 N-st 由 0 变成 1 时,计数器 1 递增输出 st,并且计数器 2 必须复位。当 March 算法的所有操作都在所有地址完成时,信号 et 由 0 变成 1,计数器 3 递增输出 cant,然后 et 复位。算法模块根据 cant 的值决定执行的测试算法,生成 st 的最大值,以及每个测试元素需要操作的次数和地址执行方向。可编程 Mbist 电路实现 8 个测试算法的完整流程如图 4(a)所示,单个测试算法的具体流程如图 4(b)所示。

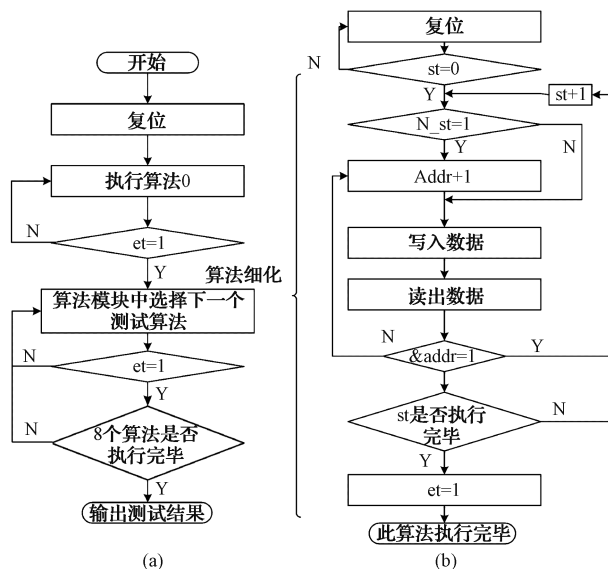


图 4 可编程 Mbist 控制器流程

Fig. 4 Programmable Mbist controller flowchart

2.1 Mbist 时序分析

March 系列算法要求对存储器地址执行包括读、写以

Timing diagram for the write-back sequence. The diagram shows a Clock signal and a sequence of operations: Write, Read, Expectation Comparison, Write, and Read. The operations are performed in pairs (Write then Read) for each of the two data items. The 'Expectation Comparison' operation occurs after the first Read and before the second Write.

Fig. 5 Traditional timing design

Timing diagram for a 64Kbit DRAM array. The diagram shows the relationship between Clock, Control Signal, Input Data, Output Data, Expected Data, and Comparison Result over six clock cycles. Vertical dashed lines mark the start of each cycle. The Control Signal alternates between Write and Read operations. Input Data is provided during write cycles and Read Data is provided during read cycles. The Comparison Result shows mismatches (X) at the end of the second, fourth, and sixth cycles.

Signal	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6
Clock	High	Low	High	Low	High	Low
Control Signal	写操作	读操作	写操作	读操作	写操作	读操作
Input Data	DDDD	ZZZZ	CCCC	ZZZZ	BBBB	ZZZZ
Output Data		DDDD	XXXX	CCCC	XXXX	BBBB
Expected Data		DDDD	XXXX	CCCC	XXXX	
比较结果			0	X	0	X

Fig. 6 Parallel timing design

根据表 1 设定每个测试算法对应的测试元素序号以及设计每个测试元素需要执行几次操作并将这些信号传输给控制器。算法模块端口描述如图 7 所示。算法模块根据所选择的测试算法输出指定的 March 元素,并设定每个测试算法对应的测试元素内的步骤数,如 March Y 对应的测试元素为 M0、M1、M4、M5 并设计该算法下 M1、M4 测试元素内的步骤数 state₂=010。

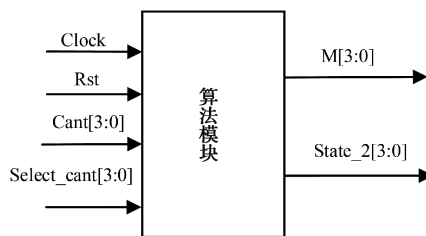


Fig. 7 Algorithm module port description

11) 输出测试结束信号。

Fig. 8 State transition diagram of state machine

“1”的数据,同时比较读出数据的对错。当第一个算法执行结束后,状态机进入 alog1_start 状态,依此类推,完成对 BRAM 的测试任务。

2.4 比较分析器

比较分析电路主要将从 BRAM 输出的读取数据与测试向量生成器生成的参考数据进行比较,以分析 BRAM 是否存在故障。比较分析电路的结构如图 9 所示,主要包括比较电路和故障计数器两个部分。Mbist 电路开始工作后,BRAM 的输出值与参考值进入比较分析器进行比

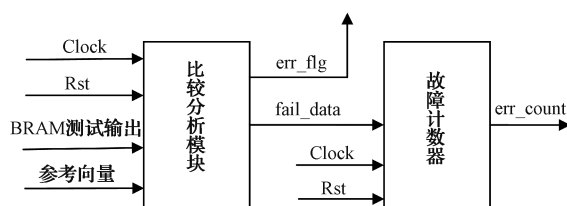


图 9 比较分析结构

Fig. 9 Comparative analysis structure

较,通过内部模块 err_flg 信号判断 BRAM 是否存在故障。如果 err_flg 信号从低电平跳变到高电平,表明存在故障。同时,比较分析电路会将故障信息传递给故障计数器,方便统计故障数。

3 实验结果

3.1 仿真结果

根据所提出的可编程 Mbist 测试方法,采用 Verilog HDL 语言编程实现所提出的电路设计,通过 modelsim 仿真工具对 1Kbit×BRAM 的 BRAM 进行仿真测试。为保证测试方法的正确性,需要对所提出的可编程 Mbist 测试方法以及 8 个测试算法的功能进行仿真,仿真结果如图 10 所示。其中每个算法执行完成时 fnsh_flg 信号会从低电平跳变到高电平,cant 的值加 1,直到执行到数字 7 并且 fnsh_flg_8 和 Mbist_done 信号从低电平跳变到高电平,表示 8 个测试算法执行完毕,Mbist_fial 一直处于低电平,说明待测 BRAM 不存在故障。

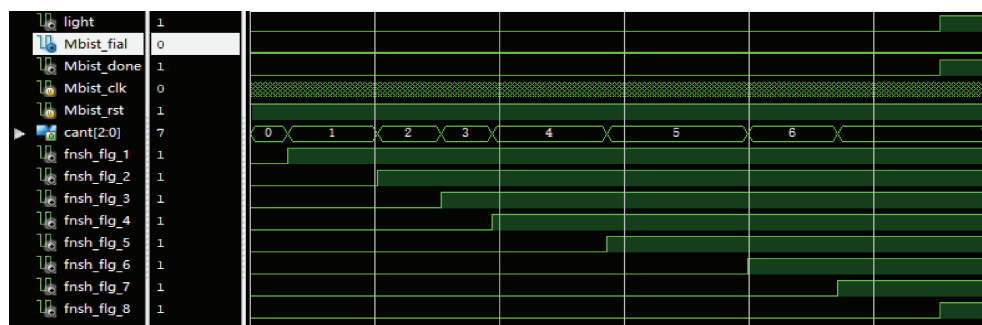


图 10 功能仿真

Fig. 10 Functional simulation diagram

为验证所提出的可编程 Mbist 方法的故障检测能力,在 BRAM 的“0101101001”地址注入一个固定“0”故障,对所有地址执行写 1 操作,采用 March X 测试算法的仿真结果如图 11 所示。图 11 中在“0101101001”地址处,Mbist_fial 从低电平跳变成高电平,表明检测并定位到该故障。

进一步进行一组对比仿真实验,比较提出的可编程

Mbist 方法和 2.1 节所提到的常见 Mbist 方法的故障检测率。仿真工具采用 modelsim,待测块存储器为 1 Kbit×36 的 BRAM,方法是对 BRAM 进行故障注入后分别采用提出的可编程 Mbist 方法和常见的 Mbist 方法进行测试,比较故障检测率。

在该仿真实验中,人为注入 24 个静态随机存储器故

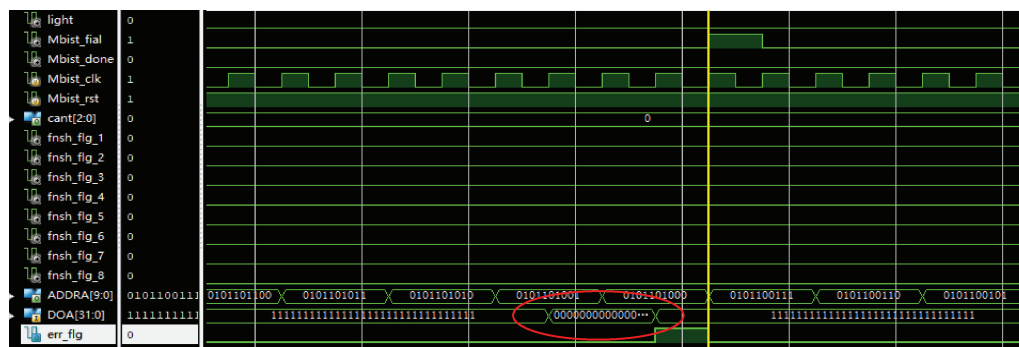


图 11 March X 的故障定位

Fig. 11 Fault location map of March X

障,其中包括4个SAF、3个TF、3个翻转耦合故障、3个等幂耦合故障、3个状态耦合故障、4个读破坏性故障、4个写破坏性故障、4个双端口读写故障、2个先进先出功能故障、2个级联功能故障,共注入32个故障。

常见 Mbist 方法的故障检测结果如图 12 所示。其中, Mbist_done 信号从低电平未变到高电平之前, err_

count 信号的值得从 0 逐渐变大,表明检测到的故障数也越来越多,只需要观察当 Mbist_done 信号从低电平跳变到高电平时, err_count 信号的值就是常见 Mbist 测试方法检测到的故障数,由此可知常见的 Mbist 测试方法检测到的故障数是 27 个,故障检测率为 84.375%。

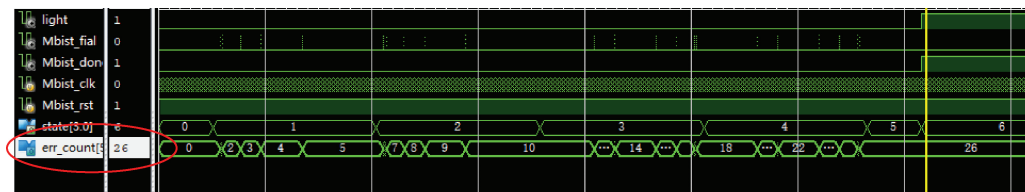


图 12 常见 Mbist 测试方法的注入故障仿真

Fig. 12 Injection fault simulation of common mbist testing methods

可编程 Mbist 方法的故障检测如图 13 所示, cant 的值表示测试算法的序号, Mbist_done 信号从低电平未变到高电平之前, cant 的值逐渐增大,直到 8 个测试算法执

行完毕, Mbist_done 信号才跳变到高电平,此时可以看到 err_count 信号的值是 32,表明所提出的可编程 Mbist 测试方法检测到的故障数是 32 个,故障检测率为 100%。

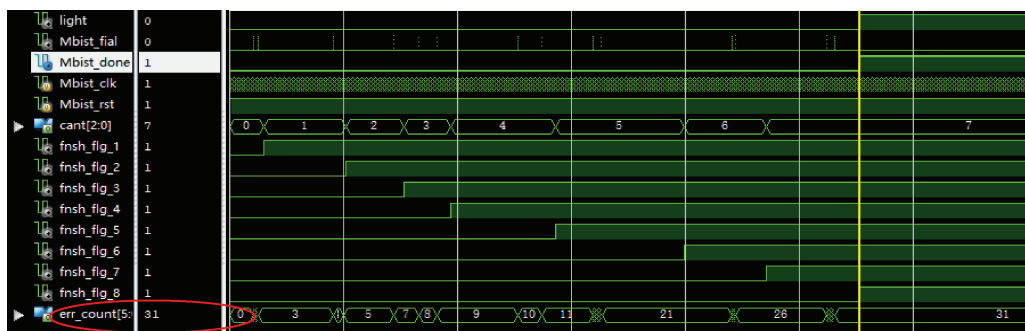


图 13 可编程 Mbist 测试方法的注入故障仿真

Fig. 13 Injection fault simulation of programmable mbist testing method

3.2 实际测试

本文选用 Xilinx ZYNQ-7000 系列芯片为待测芯片,进行 BRAM 的功能测试。所有 BRAM 测试项目均在自动化测试设备的 SmarTest 软件环境中完成,验证本文提出的测试方法有效性,自动化测试设备硬件系统如图 14 所示。采用常见的 Mbist 方法和可编程 Mbist 方法实现 BRAM 的资源全覆盖,均设计了 16 段测试码,比较两种测试方法的测试时间。如表 2 所示,由于每个功能配置项需要 3 个通用测试项辅助完成且通用测试项的测试时间相同,表 2 未做统计,可以看到每个功能配置项所用时间。

可编程 Mbist 方法集成 8 种测试算法可以提高故障的覆盖率,同时采用重配置技术,在配置一次全码后只需要再配置 15 次待测 BRAM 部分,同样可以实现资源全覆盖。图 15 所示为采用常见的 Mbist 方法实现 BRAM 资源全配置所用的时间,图 16 所示为可编程 Mbist 测试方法实现 BRAM 资源全配置所用的时间,可以看出可编程 Mbist 方法的测试时间与常见的 Mbist 方法测试时间相



图 14 自动化测试设备硬件系统

Fig. 14 Hardware system of automated testing equipment

比缩短了 76.894 s,测试效率提高了 26.1%,从而降低成本。

表 2 配置时间
Table 2 Configuration schedule

测试项	常见 Mbist	可编程
	方法全配置 时间/s	Mbist 方法 配置时间/s
Config_ram1_no_change_wa_ra	17.267	148.446
Config_ram1_no_change_wb_rb	17.344	3.441
Config_ram1_read_first_wa_rb	17.150	3.452
Config_ram1_write_first_wb_ra	17.163	3.422
Config_ram2_read_first_wa_rb	17.133	3.426
Config_ram2_write_first_wb_ra	17.226	3.432
Config_ram4_read_first_wa_ra	17.225	3.437
Config_ram4_write_first_wb_rb	17.062	3.378
Config_ram9_read_first_wb_ra	17.375	3.588
Config_ram9_write_first_wa_rb	17.428	3.596
Config_ram18_read_first_wb_rb	17.713	3.637
Config_ram18_write_first_wa_ra	17.811	3.642
Config_ram36_no_change_wa_ra	17.911	3.893
Config_ram36_no_change_wb_rb	17.954	3.965
Config_ram36_read_first_wa_rb	18.123	4.203
Config_ram36_write_first_wb_ra	18.194	4.225
总计	280.079	203.185

```
Site 1 has PASSED. Binned to bin N/A(-1)
Testflow ended after 234.341 seconds on 01/03/2023 at 07:08:46 PM
Level ended after 294.566 seconds on 01/03/2023 at 07:08:46 PM
Test program terminated after 294.566 seconds on 01/03/2023 at 07:08:46 PM
##### Ended Testprogram Testflow #####
```

图 15 常用测试方法实现 BRAM 资源全配置时间
Fig. 15 Common testing methods for achieving full configuration time of BRAM resources

```
Site 1 has PASSED. Binned to bin N/A(-1)
Testflow ended after 217.654 seconds on 01/03/2023 at 07:22:49 PM
Level ended after 217.675 seconds on 01/03/2023 at 07:22:49 PM
Test program terminated after 217.675 seconds on 01/03/2023 at 07:22:49 PM
##### Ended Testprogram Testflow #####
```

图 16 可编程 Mbist 方法实现 BRAM 资源全配置时间
Fig. 16 Programmable Mbist method for achieving full configuration time of BRAM resources

4 结 论

本文在分析多种测试算法以及 Mbist 一般结构的基础上,提出了一种新的基于可编程 FSM 的 Mbist 测试方法,该方法是通过 3 个计数器驱动的可编程 Mbist 控制模块和算法模块集成 8 种测试算法测试 BRAM,提高故障覆盖率和灵活性。所提出的方法能够精准统计和定位故障,比常见 Mbist 测试方法故障检测率提高了 15.625%,测试效率提高了 26.1%;所提出的测试方法更换测试算

法时不需重新设置 Mbist 电路,大大提高了灵活性和减少人力从而降低测试成本。但设计的可编程 Mbist 电路主要是针对 BRAM 进行设计的,可应用于其他型号的大规模芯片内 BRAM 的测试,还未对其他类型的存储器进行验证,不具备广泛性,对其他类型存储器可以考虑进一步探究。

参 考 文 献

[1] 张鲁萍. 超大规模集成电路适应性测试方法综述[J]. 数字技术与应用, 2023, 41(9): 134-136.
ZHANG L P. Overview of adaptive testing methods for very large scale integrated circuits [J]. Digital Technology & Application, 2023, 41(9): 134-136.

[2] NGUAN KONG T S, EZAILA ALIAS N, HAMZAH A, et al. An efficient march (5n) FSM-based memory built-in self test (MBIST) architecture[C]. 2021 IEEE Regional Symposium on Micro and Nanoelectronics (RSM), 2021: 76-79.

[3] 谈恩民, 朱峰, 尚玉玲. 基于 SPEA-II 算法的 SoC 测试多目标优化研究[J]. 国外电子测量技术, 2015, 34(8): 29-33.
TAN EN M, ZHU F, SHANG Y L. Optimization of SoC test multiple objects based on SPEA-II algorithm[J]. Foreign Electronic Measurement Technology, 2015, 34(8): 29-33.

[4] 陈冬明, 成建兵, 蔡志匡. 基于 40nm 超大规模 SoC 芯片存储器测试电路设计与实现[J]. 电子器件, 2017, 40(4): 813-818.
CHEN D M, CHENG J B, CAI ZH K. MBIST design for the 40 nm SoC chip[J]. Chinese Journal of Electron Devices, 2017, 40(4): 813-818.

[5] 刘宇翔, 张战刚, 杨凯歌, 等. 28 nm 超大规模 FPGA 的 BRAM 单粒子效应测试方法研究[J]. 电子产品可靠性与环境试验, 2017, 35(5): 5.
LIU Y X, ZHANG ZH G, YANG K G, et al. Research on the single event effect test method for BRAM of 28 nm ultra large scale FPGA[J]. 2017, 35(5): 5.

[6] SHARMA M, DHANOA J. Smart logic built in self-test in SOC [C]. 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 2020: 1-4.

[7] 陈佳楠, 马永涛, 李松, 等. 多目标优化的多存储器内建自测试[J]. 电子测量与仪器学报, 2020, 34(1): 193-199.
CHEN J M, MA Y T, LI S, et al. Multi-memory built-in self-test based on multi-objective optimization[J]. Journal of Electronic Measurement and Instrumentation, 2020, 34(1): 193-199.

- [8] 葛云侠, 武乾文, 赵益波, 等. 一种优化 FPGA 内嵌 BRAM 自检测 March C+ 算法[J]. 国外电子测量技术, 2022, 41(4): 1-7.
GE Y X, WU Q W, ZHAO Y B, et al. An optimized March C+ algorithm for embedded BRAM self-test in FPGA [J]. Foreign Electronic Measurement Technology, 2022, 41(4): 1-7.
- [9] MANESHINDE N, HEGADE P, MITTAL R, et al. Programmable FSM based built-in-self-test for memory[C]. 2016 IEEE International Conference on Recent Trends in Electronics Information Communication Technology, 2016: 194-199.
- [10] PAUL A, RONY A P. Optimized microcode BIST architecture for multiple memory cores in SoCs[C]. 2018 3rd IEEE International Conference on Recent Trends in Electronics Information Communication Technology, 2018: 910-914.
- [11] 吕凯. 低电压存储器 BIST 测试技术研究[实现][D]. 南京: 南京邮电大学, 2020.
LYU K. Research and implementation of low-voltage memory BIST technology [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2020.
- [12] WOJCIECHOWSKI A A, MARCINEK K, PLESKACZ W A. Configurable MBIST processor for embedded memories testing[C]. 2019 MIXDES-26th International Conference on Mixed Design of Integrated Circuits and Systems, 2019: 341-344.
- [13] AGRAWAL M, MUKESH A, CHAKRABARTY K, et al. A distributed, reconfigurable, and reusable BIST infrastructure for test and diagnosis of 3-D-stacked ICs [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems, 2016, 35(2): 309-322.
- [14] ZHANG L, WANG Z, LI Y, et al. A precise design for testing high-speed embedded memory using a BIST circuit [J]. Iete Journal of Research, 2017, 63(4): 1-9.
- [15] PARK Y, KIM H S, CHOI I, et al. A flexible programmable memory BIST for embedded single-port memory and dual-port memory [J]. ETRI Journal, 2013, 35(5): 808-818.

作者简介

葛云侠(通信作者), 硕士, 测试工程师, 主要研究方向为集成电路测试。
E-mail: gyxia2021@163.com