

一种将测试集嵌入到 Test-per-Clock 位流中的方法

刘铁桥^{1,2} 邝继顺¹ 蔡 烁¹ 尤志强¹

¹(湖南大学信息科学与工程学院 长沙 410082)

²(杭州电子科技大学 杭州 310018)

(tieqiao120@163.com)

A New Method of Embedding Test Patterns into Test-per-Clock Bit Stream

Liu Tieqiao^{1,2}, Kuang Jishun¹, Cai Shuo¹, and You Zhiqiang¹

¹(College of Information Science and Engineering, Hunan University, Changsha 410082)

²(Hangzhou Dianzi University, Hangzhou 310018)

Abstract The key of IC testing lies in the test patterns generator (TPG) design. Traditional testing methods do not make full use of the test data bit stream to construct test patterns during the test generation and application process, which results in high test cost due to the enormous test data. Test-per-scan scheme exposes the drawback of long test application time with the test data volumes increasing. In order to reduce the test cost, a built-in self test (BIST) scheme based on test-per-clock testing is proposed. Based on the analysis of linear shift test structure, a corresponding forward-backward test patterns generation method is proposed, which efficiently embeds the test set into test-per-clock bit stream. In this method, test patterns are determined by the solution of input-stream with fault dropping, where the input-stream is composed by the first bits of these patterns. The solved minimum input-stream after repeatedly reduction is directly stored in the memory to control the linear shifter in the test application, so as to generate the whole required test set. The experimental results demonstrate that the proposed method, under the precondition of meeting the required fault coverage, can obviously shorten test application time and reduce storage area overhead compared with other approaches.

Key words built-in self test; test-per-clock; test bit stream; test generation; test cost

摘 要 集成电路测试方案的关键在于测试向量产生器的设计. 传统的测试方法在测试向量生成、测试应用的过程中, 没有充分利用测试数据位流来构建测试向量, 从而造成了测试时间和存储开销的增加. 为了减少测试成本, 提出了一种基于 test-per-clock 模式的内建自测试方法. 通过对线性移位测试结构的分析, 提出了一种递进式的反复测试生成方法: 顺序求解输入位流, 逆向精简, 多次求解以获得更优值, 最终将测试集以较小的代价嵌入到 test-per-clock 位流中. 在测试应用时, 只需存储求解后的最小输入流, 通过控制线性移位的首位从而生成所需的测试集. 实验结果表明, 在达到故障覆盖率要求的前提下, 能显著地减少测试应用时间和存储面积开销.

关键词 内建自测试; test-per-clock; 测试位流; 测试生成; 测试开销

中图法分类号 TP302

收稿日期: 2013-02-27; 修回日期: 2013-08-13

基金项目: 国家自然科学基金项目(60773207); 教育部新世纪优秀人才支持计划基金项目(NCET-12-0165)

通信作者: 邝继顺(jshkuang@hotmail.com)

随着微电子技术的迅速发展,集成电路(integrated circuit, IC)的复杂度不断提高,芯片面积不断增加而芯片特征尺寸却日益缩小. 半导体工艺技术的进步给广大消费者带来福音的同时也给集成电路测试带来了严峻的挑战. 首先,大规模集成电路测试需要越来越多的测试数据,测试设备需要更大的存储容量来存储这些数据;其次,测试设备需要更多的测试通道和测试应用时间;再次,某些故障,比如时延故障只有当信号在高频率下发生翻转时才能被检测^[1]. 为了保证芯片的高可靠性,全速测试(at-speed testing)变得越来越重要. 所有的这些对集成电路测试系统提出了更高的要求.

集成电路测试过程可由生成测试向量、应用测试向量、捕获和分析测试响应这 3 部分组成. 从测试数据流的角度来看,即测试数据流由测试向量产生器(test patterns generator, TPG)产生、流经被测电路(circuit under test, CUT)、最后被响应分析器(response analyzer, RA)捕获的过程. 在集成电路测试方案中,TPG 的设计是关键,因为 TPG 决定了整个测试方案能够达到的故障覆盖率以及所需的面积开销和测试时间的大小. 高效的 TPG 即为如何利用较少的硬件开销产生较小的测试数据位流,在较短的时间内通过被测电路的问题. 常见的 TPG 方案主要有以下 3 种:

1) 基于伪随机测试. 线性移位反馈寄存器(linear feedback shift register, LFSR)常用来生成伪随机测试向量,这种方法只需要很少的硬件,并且生成的测试向量序列是可以重复的. LFSR 的问题是由于随机向量难测故障(random pattern resistant faults)的存在,经常需要很长的测试向量序列和测试应用时间才能达到较高的故障覆盖率^[2]. 文献[3]采用加权伪随机向量测试,通过调节 LFSR 加载到 CUT 每个输入端上 0 或 1 的概率,来提高故障覆盖率并降低测试向量规模. 文献[4]使用受控 LFSR 跳过伪随机测试序列中对故障覆盖率没有贡献的测试向量,从而减少了测试向量长度,缩短了测试时间. 文献[5]通过利用存储的控制信号对伪随机无关块进行阻塞,从而达到减少测试存储的目的.

2) 基于线性解压的方法. 测试压缩技术与伪随机测试不同的是,它是针对给定测试向量的压缩. 最常用的是利用 LFSR 对测试向量进行压缩,其核心思想是先将确定的测试立方编码成 LFSR 种子,而在测试应用时,将种子载入作为 LFSR 的初始状态,

自动展开解压成目标向量^[6]. LFSR 重播种技术在压缩无关位较多的测试集中表现出很好的性能. 基于 LFSR 重播种方法有许多,如多多项式重播种(MP-LFSR)^[7]、并行 LFSR 重播种^[8]等. 文献[9]提出了一种应用于时延测试的基于种子选择和排序的 LFSR 重播种方法. 文献[10]提出了一种结合 LFSR 重播种和加权伪随机测试产生方法. 文献[11]中提出一种基于 Viterbi 算法的线性解压方法,该方法结合了多扫描链技术,通过将 n 位的输入解压成 m 位输出($m \gg n$),从而达到很好的测试压缩目的.

3) 基于数据编码方法. 该类测试压缩技术采用编码的方法,将预先计算好的测试向量集编码成一系列码字,并存储在测试仪中. 测试时测试仪中的压缩数据通过片上的解压电路被还原成测试向量,并加载到被测试电路,从而解决了测试应用存储开销过大的问题. 但测试编码技术在解码成原始测试集时,所需的解码器无疑额外增加了硬件成本. 常见编码压缩方法有 Huffman 编码^[12]、Golomb 编码^[13]、FDR 编码^[14]等.

传统测试方案在测试时间或存储开销上不同程度地减少了测试代价,但其总的效果却不甚满意. 究其根本原因,传统基于数据编码的方法由于在测试向量施加时,向量与向量之间彼此独立,即当前向量生成后,下一向量又得以同样或类似的方式重新生成;而对基于 LFSR 的测试方法而言,下一向量由前向量移位反馈得到,向量之间彼此的牵制性又太大. 这些都是由于在测试向量生成、应用的过程中,没有很好地利用测试数据位流构建测试向量,从而造成测试时间和存储开销的增加. 另一方面,在传统 test-per-scan 测试模式下,每一扫描时间周期产生一个测试向量,测试向量和测试响应分别从扫描链中被串行地移入和移出. 因此, test-per-scan 的测试应用时间为电路中扫描链的长度和测试向量数目的乘积. 随着电路规模的增大, test-per-scan 所需的过长测试时间将不可忍受. 如考虑测试向量 $V_1 = \{1001\}$, $V_2 = \{0100\}$, 传统 test-per-scan 测试施加 V_1 和 V_2 时都需要 5 个测试时钟周期(4 个移位周期加 1 个响应捕获周期,假定单链模式),而从重复利用测试位流考虑,在施加 V_1 后,只需一个测试时钟右移入数据 0,即可得到 V_2 . 因此,本文拟从优化 TPG 的测试数据流出发,提出了一种 test-per-clock 测试模式下基于串行移位的递进式反复测试生成方法. 在每次测试生成中,通过利用输入位流控制移位

寄存器的移位来动态匹配测试向量,直到所有故障都被检测.当输入位流被求解后,逆向模拟测试集,进一步精简输入位流.此过程可反复 N 次以获得更少的输入位流(在下次测试产生时,将得到的输入位流删除前面 l 位,继续测试生成).在本文方案中,输入位流即为扫描链中的测试数据流,通过充分利用输入位流构建测试向量,从而达到有效减少测试时间和存储开销的目的.

1 相关工作

在 test-per-clock 测试结构模式下,每一测试时钟周期产生一个测试向量,同时其响应被响应分析器捕获.因而,与 test-per-scan 模式相比, test-per-clock 模式在测试应用时间上具有较强的优势.但是其测试结构的硬件代价却比常规 test-per-scan 结构要高,尤其是全扫描时序电路中响应分析器的实现. BILBO(built-in logic block observer)^[15] 将触发器进行了改造,使其可以在正常模式、移位模式、伪随机模式和 MISR(multiple input shift register)响应压缩模式间切换. CBILBO^[16] 将触发器改造成 double flip-flop 模式,可以同时进行伪随机向量生成和响应压缩.专利文献[17]将扫描链进行修改,将测试响应的压缩处理结果用作下一个测试产生,使其也同时具备向量产生和响应压缩的功能.文献[18]提出了一种退化的 MISR 来替换响应压缩,简化了文献[17]的结构.然而,上述文献都采用了(类)伪随机向量生成器作为测试产生器,由于抗伪随机故障的存在,故障检测在短时间很难达到满意的覆盖率,使得 test-per-clock 模式在测试应用时间上的优势不能得到很好的体现.文献[19]提出了一种确定型的 test-per-clock 测试结构.文献[20]结合细胞自动机(cellular automaton, CA),提出了一种伪随机和确定型的混合 test-per-clock 测试方法.在文献[19-20]中,当电路处于测试时,触发器不再接收电路响应,而是由额外设计的空间压缩器和 MISR 结构并行接收.空间压缩器虽然减少了 MISR 的输入数,但由于其由异或网构成,多层的异或门会恶化响应捕获的时延,不能适应高频率的测试.另外,空间压缩器使得该测试结构的适应性变得单一,电路的任何结构变化(如后期工程更改)和测试程序的变化(如更换测试集)都会导致故障覆盖率的降低.

2 测试结构和原理分析

2.1 test-per-clock 内建自测试结构

基于线性移位的内建自测试 test-per-clock 结构如图 1 所示.图 1(a)给出了测试结构总体框架图. SRL 单元由上下两部分构成,上半部分由电路的原始触发器组成,下半部分由额外添加的触发器和对应的异或门以响应分析器(如 MISR)的形式组成.原始输入端(primary input, PI)的辅助触发器和 SRL 单元上半部分串联在一起,构成了移位寄存器.在测试时钟控制下,存储器每移入一位数据到移位寄存器,移位寄存器中便产生了一个新的测试向量.测试向量通过 PI 和伪输入端(pseudo-primary input, PPI)施加到电路,电路响应则由原始输出端(primary output, PO)和伪原始输出端(pseudo-primary output, PPO)传出. PO 端的响应由传统 MISR 捕获, PPO 端的响应则被 SRL 的下半部分捕获. SRL 单元的具体设计如图 1(c)所示,它在原始 SRL 单元(如图 1(b)所示)上作了一些修改.

与文献[17]最大不同的是,文献[17]中 SRL 单元中锁存器 L_1 的值由异或处理后的前一测试响应提供,而修改后的 SRL 单元中触发器 L_1 的值直接由存储器中的位流提供.因此,修改后的 SRL 单元将原始伪随机测试转变成确定型测试.其工作模式如下:当 test_mode 和 res_mode 信号为“on”时, SRL 单元工作在测试模式下, scan-in 测试数据由前一 L_1 单元移入,同时来自 sys-data 的电路响应被 L_2 和它的异或网络捕获.当 test_mode 信号为“on”而 res_mode 为“down”时, SRL 单元工作在移位模式下.当 shift_clk2 时钟信号未被激活而 test_mode 信号为“down”时, SRL 单元工作为正常模式.由于时钟信号 shift_clk1 和 shift_clk2 可由系统时钟提供,因此该测试结构可适用于全速测试.

2.2 分析

在讨论测试生成方法之前,我们先来看一个范例.假设一被测电路有 6 个故障 $f_0 \sim f_5$, 对每个故障,用测试向量自动生成工具(automatic test pattern generator, ATPG)分别产生一个对应的测试向量,称为目标向量集 $\{V_0 \sim V_5\}$.我们新建一个空的测试集,将 V_0 作为初始向量 V'_0 加入到新测试集中,假设 V'_0 可以检测故障 $\{f_0, f_4\}$, 则将故障 $\{f_0, f_4\}$ 从故障列表中删除,并删除故障对应的向量 $\{V_0, V_4\}$.将 V'_0 移若干位后从剩余目标向量集 $\{V_1, V_2, V_3,$

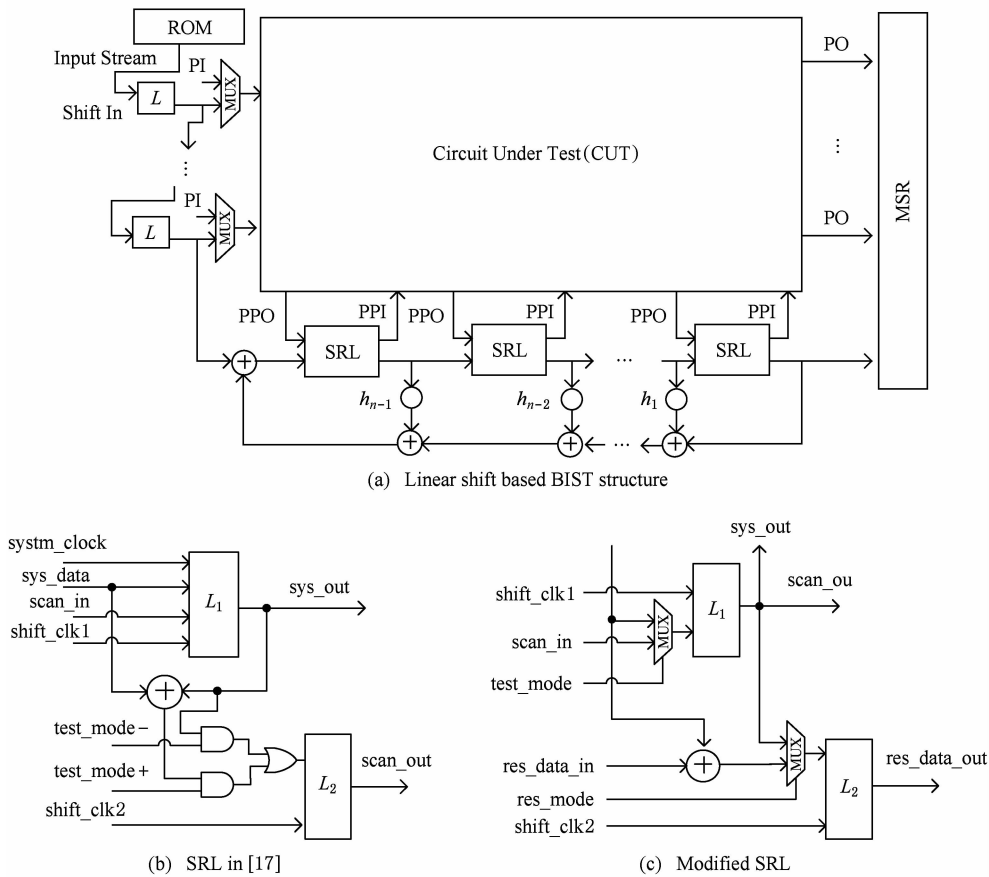


Fig. 1 Test-per-Clock BIST scheme.
图 1 Test-per-Clock BIST 结构图

$V_5\}$ 寻找到最佳匹配向量 V_3 , 假设 V'_0 移位到 V'_2 后找到匹配向量 V_3 还需经过向量 V'_1 , 且 V'_1 能新检测故障 f_1 , V'_2 能新检测到故障 f_3, f_5 , 则将 f_1, f_3, f_5 和 V_1, V_3, V_5 从相应列表删除. 同理, 将 V'_2 移若干位寻求最佳匹配向量, 并动态删除故障和对应的剩余向量, 直到目标向量集或剩余故障为空. 其过程如表 1 所示:

Table 1 Example for Test Generation
表 1 测试生成示例

New Test Pattern	Fault New Detected	Fault Remained	Original Test Set Remained
Φ	Φ	$f_0, f_1, f_2, f_3, f_4, f_5$	$V_0, V_1, V_2, V_3, V_4, V_5$
$V'_0=V_0$	f_0, f_4	f_1, f_2, f_3, f_5	V_1, V_2, V_3, V_5
V'_1	f_1	f_2, f_3, f_5	V_2, V_3, V_5
$V'_2=V_3$	f_3, f_5	f_2	V_2
$V'_3=V_4$	f_2	Φ	Φ

最后得到的新测试集为 $\{V'_0, V'_1, V'_2, V'_3\}$, 我们可以看出, 当该移位方法运用到上文提出的测试结

构时, 在移位寄存器初始化 V'_0 后, 只需移入 3 位数据 (V'_1, V'_2, V'_3 的首位), 便可实现电路的完整测试生成. 下面分析该移位方法的可行性.

定义 1. 若测试向量 V_1 和 V_2 相容, 即向量 V_1 和 V_2 的海明距离为 0, 则 V_1 和 V_2 互为匹配向量.

显然, 对于任意 n 阶的移位寄存器, 只需移入 n' ($n' \leq n$) 位数据位, 一定能够将移位寄存器中现有的测试向量转变成任何一个指定的目标向量. 因为当移入 n 个输入位时, 移位寄存器中的向量变为 $\{I_n, I_{n-1}, I_{n-2}, \dots, I_1\}$, 此时可以匹配任何指定的目标向量 $B = \{b_{n-1}, b_{n-2}, \dots, b_1, b_0\}$, 只需 $I_i = b_{i-1}$. 而在第 n' ($n' < n$) 个移位时, 移位寄存器中的向量就有可能匹配了向量 B .

我们从另一个角度来分析这个问题: 假定移位寄存器中当前的向量 $A^0 = \{a_{n-1}, a_{n-2}, \dots, a_1, a_0\}$ 为完全确定型向量 (每一位的值为 0 或 1), 目标向量 B 为部分确定型向量, 含无关位 X 的比率为 λ ($0 \leq \lambda \leq 1$), 则向量 A 和 B 相互匹配的概率为

$$p_0 = \left(\lambda \times 1 + (1 - \lambda) \times \frac{1}{2} \right)^n = \left(\frac{1 + \lambda}{2} \right)^n. \quad (1)$$

每个测试时钟周期移位寄存器移入的数据位可被临时看成为无关位 X , 当第 i 个测试时钟到来时, 移位寄存器中的向量变成为 $A^i = \{X_i, \cdots, X_1, a_{n-1}, a_{n-2}, \cdots, a_i\}$. 此时向量 A^i 匹配目标向量 B 的概率为

$$P_i = \left(\frac{1+\lambda}{2}\right)^{n-i}, \tag{2}$$

其中, n 表示向量的位数, i 表示移位时钟数, 而 λ 表示目标向量中不确定位的比率. 从式(2)我们可以看出, 匹配概率 P_i 随 λ 和 i 值的增大而增大, 当 P_i 一定时 λ 和 i 反向增长. 因此, 我们要想在较少的移位时间内获得成功的匹配, 需要目标向量具有较高的 λ 值.

3 基于线性移位的递进式反复测试生成方法

正如前面所述, 存储在存储器中的输入位流不仅决定了每个时钟移入移位寄存器的测试向量, 同时也决定了测试的存储开销和测试应用时间. 因此, 测试生成算法的关键就在于如何确定这些输入位流的值, 最终以较少的位流实现高效测试.

基于线性移位的测试生成算法如图 2 所示, 整

个算法流程由 4 部分组成:

1) 初始化阶段. 根据式(2)的分析, 为了提高移位匹配概率, 我们试图将目标向量中无关位的比率(λ 值)最大化: 即对故障列表 FL 中的每一个故障, 都生成一个对应的测试向量(含 X). 这些向量构成了原始目标向量集 Obj_T . 初始向量我们选择随机模式产生, 对其故障模拟后, 分别从故障列表和原始目标集中删除已测故障以及这些故障对应的测试向量.

2) 一个单次的测试生成过程. 它主要包括移位和向量匹配. 对当前测试集 C_T 末尾向量进行移位时, 移入的位暂时定为无关位 X , 称为临时向量. 临时向量在目标集 Obj_T 中寻找匹配向量, 当没有可匹配的向量时继续移位; 当存在多个可匹配向量时, 匹配的策略为 λ 值最小的目标向量优先, 然后是“尾端密集度”高的向量. 由式(1)和式(2)可知, 一个测试向量中确定值的位数越多, 移位生成该向量则越难. 因此, 我们优先选择匹配确定位多的目标向量, 剩下无关位多的目标向量, 从而提高后续向量的匹配概率. “尾端密集度”指的是向量尾部确定位的个数. 虽不像 λ 值那样直观地影响测试向量生成, 但尾端密集度高的向量, 说明其确定位相对集中在移位

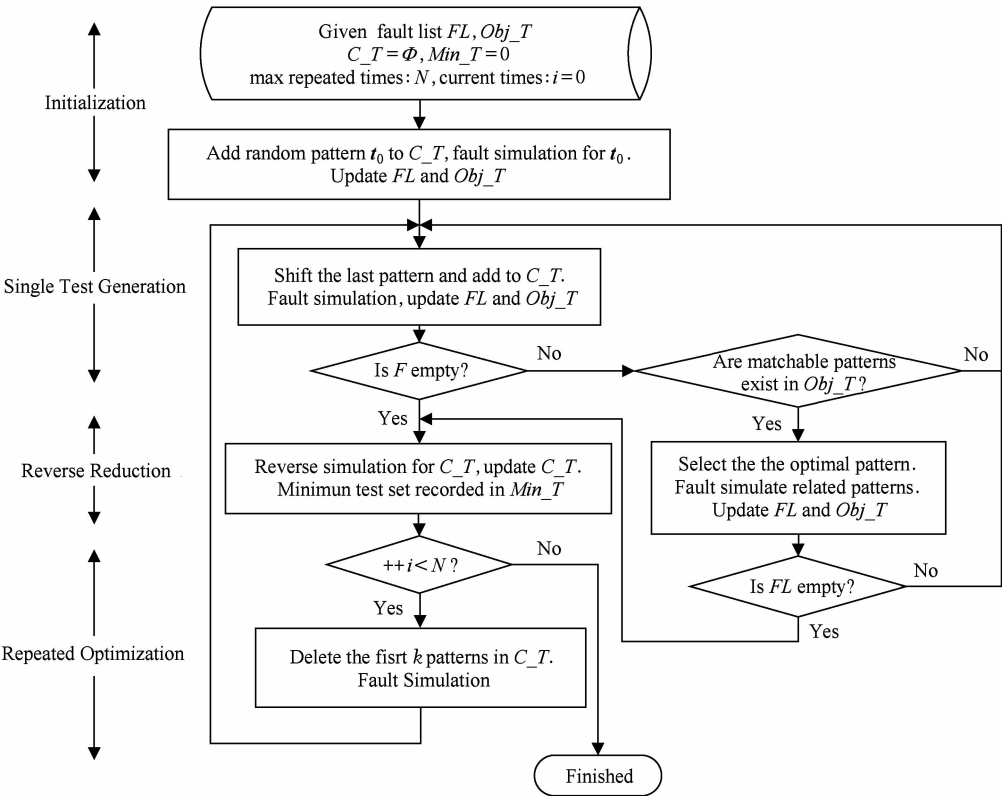


Fig. 2 Flow of test generation algorithm.

图 2 测试生成算法流程图

寄存器尾端,相比尾端密集度低的向量,能尽快地移出确定位,增加后续向量的匹配概率.当匹配向量确定后,比较匹配向量和临时向量之间的关系,确定临时向量中 X 的值,并对测试集 C_T 中与该 X 位有关的向量进行 X 还原.在向量移位中,由于部分 X 到移出移位寄存器时仍未被赋值,为了提高测试集的故障检测能力,我们对此时的 X 进行随机赋值并还原.在测试生成的过程中,每当测试集中的向量发生变动时,为了尽快减小向量匹配的搜索空间,我们都需要对这些向量进行故障模拟,删除检测的故障和对应的目标向量,并判断故障是否都被检测.若故障未被完全检测,则继续移位匹配,否则进入下一步.

3) 测试集精简的过程.当测试集产生后,测试集中前面向量检测到的故障可能再次被后面的向量检测到.因此,为了进一步精简测试集,我们从尾部向量开始对测试集进行逆向模拟.注意到本文方法下的向量精简与常规精简方法不同.在常规向量精简方法中,向量是彼此独立的,任何向量检测到的故障同时被其他向量检测到时,该向量都可以被删除.而在本文方法中,前后向量构成了连续的数据流,因此,中间向量即使对故障检测无贡献也不能删除.

4) 反复求解、进一步优化的过程.本文采取一种递进式的反复测试生成方法,在反复测试生成的过程中,企图获得更优值.具体实现为:将当前测试集首部的 k 个向量删除,重新模拟剩余测试集,将未检测的故障和其对应的向量重新加入到故障链表 FL 和目标向量集 Obj_T 中,继续测试产生,直到重复次数 i 超出限制值 N .在反复测试生成的过程中,最小测试集被 Min_T 记录.

反复测试生成算法的时间复杂度为 $O(N \times n^2 \times NoF^2)$,其中 N 表示重复测试生成的次数, n 表示电路输入数, NoF 表示故障数目.测试生成完成后, Min_T 即为最终求解的测试集.测试应用时,当初始化 Min_T 首向量后,我们只需移入其最小输入流(Min_T 中每个向量的首位),即可完成整个移位测试生成.

4 实验结果

表 2 给出了本文测试产生方法在基准电路 ISCAS'85 和 ISCAS'89 上的实验结果,并和 test-per-clock 和 test-per-scan 方法进行了对比.原始目

标向量集的产生采用了 ATALANTA^[21] 工具,故障模拟采用了 HOPE^[22] 模拟工具.程序运行的硬件平台为奔腾 IV 3.0 GHz 处理器,2 GB 内存.

表 2 第 1 列为电路的名称,第 2~4 列为本文方法在重复次数 $N=1\,000$ 时的实验结果(10 次实验结果的平均值).第 2 列给出了第 1 次测试生成时输入位流的个数,每次删掉的首部向量数 k 约为该值的 1/10.第 3 列给出了重复 N 次的最小位流数(在本文方法中,位流数即等于需要存储的位数及测试应用时钟周期数).第 4 列给出了第 1 次测试生成的时间(单位 s).第 5 列给出了相对于传统 test-per-scan 测试结构,本文方法在 RA 上需增加的硬件开销占 CUT 硬件面积的百分比.硬件代价估算采用了 2 输入门等价方法^[23],如 2 输入与非门代价为 1.第 6~11 列给出了其他测试方法的结果以及本文最小位流数对比这些结果的减少百分比.文献[19]为 test-per-clock 测试生成方法,文献[11,24]为 test-per-scan 测试方法.其中文献[24]为 MDPRC 编码压缩方法,文献[11]为基于线性解压的方法.第 6 列给出了文献[19]的存储位数和测试应用时间,第 8 列和第 10 列给出了文献[11,24]的存储位数.最后一行给出了平均值.

从表 2 可以看出,在减少测试存储和测试应用时间上,本文方法与其他方法相比具有明显的优势.同为 test-per-clock 测试,文献[19]简单地将测试向量按故障检测能力排序,而本文方法能更有效地利用测试位流,与之相比,本文方法第 1 次测试产生就压缩了平均 31.51% 的测试数据,反复优化后平均减少率达到 50.19%,且这种优势随着电路尺寸的增大更明显.与文献[11,24]相比,本文的存储位数平均减少率达到了 49.92% 和 79.51%.由于本文的测试应用时间即为存储位的个数,而文献[11,24]是基于 test-per-scan 的测试方法,输入的数据位解压/解码后还需要经过若干个时钟周期(扫描链长度)才能施加到被测电路,因此在测试应用时间上本文方法的优势更大.

本文方法由于在 RA 中采用了类 double flip-flop 结构,时序电路中 RA 的代价随电路触发器数量的增加而增加,而组合逻辑电路中由于不含触发器,可沿用传统测试的 RA 结构.值得注意的是,当本文测试生成方法应用到 BILBO 测试结构或分块测试(partitioning)技术中时,RA 可以利用电路内部资源进行构建.

Table 2 Results for Comparison with Other Approaches

表 2 本文方法实验结果及和其他文献比较

Circuit	Proposed Method				Ref [19]		Ref [24]		Ref [11]	
	Org_T	Min_T	CPU/s	Area/%	# S,T	Reduction/%	# Store	Reduction/%	# Store	Reduction/%
C432	183	89	0.02	0	200	55.50	N/A	—	N/A	—
C499	240	128	0.03	0	410	68.78	N/A	—	N/A	—
C880	372	202	0.09	0	440	54.09	N/A	—	N/A	—
C1908	1 002	730	0.38	0	1 310	44.27	N/A	—	N/A	—
C3540	705	629	0.65	0	820	23.29	N/A	—	N/A	—
C7552	7 963	6 985	14.02	0	9 350	25.29	N/A	—	N/A	—
S5378	2 106	1 743	3.13	34.1	N/A	—	10 777	83.83	4 162	58.12
S9234	8 352	7 603	20.23	22.7	9 350	18.68	18 380	58.63	9 149	16.90
S13207	4 458	3 768	60.24	41.5	12 400	69.61	23 111	83.70	8 941	57.86
S15850	4 820	4 225	71.65	32.4	54 100	92.19	23 104	81.71	9 532	55.68
S38417	17 936	16 204	1 401.35	37.4	N/A	—	73 505	77.96	29 457	44.99
S38584	7 005	5 628	890.25	30.9	N/A	—	64 251	91.24	16 529	65.95
Average						50.19		79.51		49.92

N/A: Not available; —: Not applicable

5 结 论

测试数据量和测试应用时间是测试成本的两个主要因素. 尽管基于扫描的可测性设计方法提高了集成电路的可测性、减少了测试生成成本, 但 test-per-scan 类方案由于扫描路径的固有串行性和测试向量的独立产生导致过长的测试应用时间和较大的测试数据存储. test-per-scan 类测试方案在测试应用时间上的较大优势越来越引起关注. 为了降低测试成本, 本文提出了一种 test-per-clock 模式下基于线性移位的内建自测试生成方法, 通过控制线性移位的首位生成测试序列, 将测试集以较小的代价嵌入到输入位流中, 从而达到压缩测试数据和测试应用时间的目的. 为了进一步精简输入位流, 本文采用一种递进式的反复测试生成方法. 实验结果表明, 本文与同类 test-per-clock 方法相比, 平均减少了 50.19% 的测试存储和测试应用时间. 与基于 test-per-scan 模式下的最近方法^[11,24]相比, 存储减少量分别达到了 49.92% 和 79.51%. 本文的测试结构和测试生成方法可扩展到其他故障测试, 如针对时延故障的全速 LOS(latch-on-shift) 模式测试. 由于本文方法仅需要单个测试通道, 而且不同的被测电路所需的测试流可以重迭, 因此本文方法可有效缓解传统 SoC(system on chip) 测试中面临的测试带宽

不足、测试应用时间过长等问题. 同时, 在本文实验中, 输入位流采取的是直接存储, 不需要传统压缩技术所需的额外解压电路开销, 从而进一步降低了 TPG 的面积开销. 综上所述, 本文方法在减少测试数据存储、测试应用时间等低成本测试应用上具有广泛的应用前景.

参 考 文 献

[1] Nadeau B, Cote J F, Hulvershorn H, et al. An embedded technique for at-speed interconnect testing [C] //Proc of Int Test Conf 1999 (ITC'99). Piscataway, NJ: IEEE, 1999: 431-438

[2] Touba N A, McCluskey E J. Test point insertion based on path tracing [C] //Proc of the 14th Very Large Scale Integration Test Symp. Piscataway, NJ: IEEE, 1996: 2-8

[3] Savir J. On chip weighted random patterns [C] //Proc of the 6th Asian Test Symp (ATS). Piscataway, NJ: IEEE, 1997: 344-352

[4] Hu Chen, Xu Gefu, Zhang Zhe, et al. A BIST structure and test pattern generation method based on controlled LFSR [J]. Journal of Circuits and Systems, 2002(3): 13-16 (in Chinese)
(胡晨, 许舸夫, 张哲, 等. 一种基于受控 LFSR 的内建自测试结构及其测试矢量生成[J]. 电路与系统学报, 2002(3): 13-16)

[5] You Z Q, Wang W Z, Dou Z P, et al. A scan disabling based BAST scheme for test cost reduction [J]. IEICE Electronics Express, 2011, 8(16): 1367-1373

- [6] Wang S, Wei W L, Chakradhar S T. A high compression and short test sequence test compression technique to enhance compressions of LFSR reseeding [C] //Proc of the 16th Asian Test Symp. Piscataway, NJ: IEEE, 2007: 79-86
- [7] Venkataraman S, Rajski J, Hellebrand S, et al. An efficient BIST scheme based on reseeding of multiple polynomial linear feedback shift registers [C] //Proc of Int Conf on Computer-Aided Design. Piscataway, NJ: IEEE, 1993: 572-577
- [8] Kongtim P, Reungpeerakul T. Parallel LFSR reseeding with selection register for mixed-mode BIST [C] //Proc of the 19th IEEE Asian Test Symp (ATS). Piscataway, NJ: IEEE, 2010: 153-158
- [9] Yoneda T, Inoue M, Taketani A, et al. Seed ordering and selection for high quality delay test [C] //Proc of the 19th IEEE Asian Test Symp (ATS). Piscataway, NJ: IEEE, 2010: 313-318
- [10] Wang S, Wei W L, Wang Z L. A low overhead high test compression technique using pattern clustering with n -detection test support [J]. IEEE Trans on Very Large Scale Integration (VLSI) Systems, 2010, 18(12): 1672-1685
- [11] Lee D, Roy K. Viterbi-based efficient test data compression [J]. IEEE Trans on Computer Aided Design of Integrated Circuits and Systems, 2012, 31(4): 610-619
- [12] Gonciari P T, Al-Hashimi B M, Nicolici N. Variable length input Huffman coding for system on a chip test [J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2003, 22(6): 783-796
- [13] Chandra A, Chakrabarty K. Test data compression and decompression based on internal scan chains and Golomb coding [J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2002, 21 (6): 715-722
- [14] Chandra A, Chakrabarty K. Frequency-directed run length (FDR) codes with application to system-on-a-chip test data compression [C] //Proc of the 19th IEEE Very Large Scale Integration Test Symp. Piscataway, NJ: IEEE, 2001: 42-47
- [15] Konemann B, Mucha J, Zwiehoff G. Built-in logic block observation techniques [C] //Proc of Int Test Conf 1979 (ITC'79). Piscataway, NJ: IEEE, 1979: 37-41
- [16] Wang L T, Mccluskey E J. Concurrent built-in logic block observer (CBILBO) [C] //Proc of 1986 IEEE Int Symp on Circuits and Systems. Piscataway, NJ: IEEE, 1986: 1054-1057
- [17] Bardell P H, Mcanney W H. Simultaneous self-testing system; USA, US4513418 [P]. 1985-04-23
- [18] Son Y, Chong J, Russell G. E-BIST: Enhanced test-per-clock BIST architecture [J]. Computers and Digital Techniques, 2002, 149(1): 9-15
- [19] Novak O, Nosek J. Test-per-clock testing of the circuits with scan [C] //Proc of the 7th IEEE Int On-Line Testing Workshop. Piscataway, NJ: IEEE, 2001: 90-92
- [20] Novák O, Pliva Z, Nosek J, et al. Test-per-clock logic BIST with semi-deterministic test patterns and zero-aliasing compactor [J]. Journal of Electronic Testing, 2004, 20(1): 109-122
- [21] Lee H K, Ha D S. Atalanta: An efficient ATPG for combinational circuits, 93-12, [R]. Blacksburg, Virginia: Department of Electrical Engineering, Virginia Polytechnic Institute and State University, 1993
- [22] Lee H K, Ha D S. HOPE: An efficient parallel fault simulator for synchronous sequential circuits [J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 1996, 15(9): 1048-1058
- [23] Micheli G. De. Synthesis and Optimization of Digital Circuits [M]. New York: McGraw-Hill, 1994
- [24] Tseng W D, Lee L J. Test data compression using multi-dimensional pattern run-length codes [J]. Journal of Electronic Testing, 2010, 26(3): 393-400



computing, design for testability and low cost test.



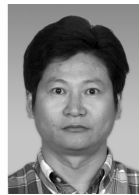
and embedded system.



Liu Tieqiao, born in 1983. Received his MEn degree in 2010. PhD candidate at the School of Information Science and engineering, Hunan University. His main research interests include fault-tolerant

Kuang Jishun, born in 1959. Professor and PhD supervisor of the School of Information Science and engineering, Hunan University. His main research interests include fault-tolerant computing

Cai Shuo, born in 1982. PhD candidate at the School of Information Science and engineering, Hunan University. His main research interests include reliability evaluation, fault-tolerant computing.



testability.

You Zhiqiang, born in 1972. Associate professor of the School of Information Science and engineering, Hunan University. His main research interests include low power testing, low cost test and design for