

生成确定性测试图形的内建自测试方法

雷绍充, 邵志标, 梁 峰

(西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 为了以低的硬件开销自动生成高效率的确定型测试图形, 提出一种新型的内建自测试(BIST)方法. 先对原型设计用自动测试图形工具生成长度短、故障覆盖率高的确定性测试图形, 然后对生成的图形排序以取得低功耗测试序列, 再选择状态机优化和综合方案, 最后自动生成 BIST 电路描述. 由于结合了确定性测试和伪随机测试的优点, 该方法具有低功耗、长度短、故障剪盖率高、测试图形自动生成等特色, 特别适于 CMOS 组合逻辑电路的测试. 基于 ISCAS85 Benchmark 的实验结果表明, 所设计的 BIST 电路在硬件开销、速度、测试功耗等方面均优于传统的伪随机测试电路, 测试时间显著减少.

关键词: 低功耗; 确定性测试图形; 内建自测试; 状态机

中图分类号: TN47; TN407 **文献标识码:** A **文章编号:** 0253-987X(2005)08-0880-05

New Built-in-Self-Test Technique for Generating Deterministic Test Pattern

Lei Shaochong, Shao Zhibiao, Liang Feng

(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A new built-in self-test (BIST) technique that automatically generates high efficient deterministic patterns at low hardware overhead is presented. The principle of the proposed technique is firstly to utilize automatic test pattern generator tools to generate deterministic test patterns with short length and high fault coverage, then the generated patterns are sorted to get low power test sequence, and finally the BIST circuit description is automatically generated by selecting finite state machine optimization and synthesis scheme. The BIST has merits of combining deterministic and pseudorandom test, and has the features of low power consumption, low length, high fault coverage, and pattern generated automatically, and is especially suitable for testing of CMOS combinational logic circuits. Experimental results on ISCAS85 Benchmark circuits show that the designed BIST circuits are prior to pseudorandom circuits in power consumption, test speed and hardware overhead, and the testing time is decreased distinctly.

Keywords: *low power consumption; deterministic test pattern; built-in self-test; finite state machine*

测试图形有两类生成方法, 一是确定性生成方法, 用专门的算法, 如 D 算法、FAN 算法生成测试图形; 二是随机(伪随机)方法产生测试图形. 前者生成的测试图形相当少, 相应的测试时间相当短, 但生成过程比较复杂, 实现比较困难, 后者则是由测试微处理器的软件算法或用专门的片上测试电路生成随机的或伪随机的测试图形, 生成容易, 通常还能达到比较高的故障覆盖率. 内建自测试(Built-in Self-Test, BIST)技术常采用基于线性反馈移位寄存器

的伪随机方法, 这种方法存在许多问题, 如大量的、不断变化的位码, 会使测试功耗增加^[1-4], 导致电路发热, 影响器件寿命, 还存在故障覆盖率低、测试图形过长等问题. 此外, 伪随机测试中常用的是固定型故障模型^[5], 难以描述 CMOS 深亚微米技术中的缺陷, 还需要延迟、桥接、恒定开路等故障类型的测试图形. 为解决这些问题, 近年来发展出一种称之为 RSIC(Random Single-Input Change)的新 BIST 方法^[6,7], 所产生的序列中相邻测试图形间只有一位

改变,可以有效地减少测试功耗和提高测试速度,改善测试效率。

本文结合确定性算法和 RSIC 法的优点,提出了一种适合嵌入式 CMOS 电路的、可综合的、低功耗的测试生成方法,先对原型设计用 ATPG 工具生成确定性测试图形,然后建立算法进行排序,使排序后的序列的位跳变次数最少。接着,对该序列进行状态机优化的设计和综合,通过开发的软件自动生成序列的状态机和组合逻辑硬件描述,借助工具分别进行位跳变密度、硬件开销、速度和测试施加时间等方面的分析,最后确立 BIST 电路设计方案。

1 确定性测试图形排序方法研究

对于组合逻辑电路,测试图形施加顺序不固定,但随机地施加测试图形时,如果相邻向量变化的位比较多,测试中势必会增加功耗,导致电路更多的热效应,还会带来时间延迟,降低测试速度。研究表明,CMOS 电路的动态功耗除与电路供电电压 V_{dd} 、电路时钟频率 f 有关外,还与平均输入信号概率 P_{in} 、平均输入跳变密度 D_{in} 、平均输入相关系数 S_{in} 和平均输出无延迟跳变密度 D_{out} 有关系^[5]。降低动态测试功耗最有效的方法就是降低跳变密度 D_{in} 。为此,将一组合逻辑电路的测试图形向量集 $T = \{T_0, T_1, \dots, T_n\}$ 的最小跳变密度矩阵定义为

$$M =$$

$$\begin{cases} M(i, j) = 1, i \neq j, T(i) \text{ 与 } T(j) \text{ 间位跳变次数最少} \\ M(i, i) = 0, \text{其他} \end{cases}$$

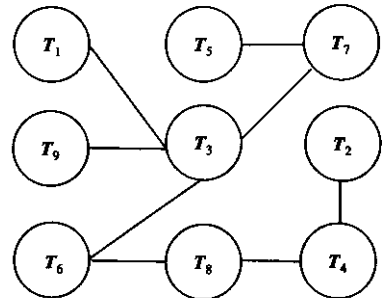
例如,由 ISCAS85 Benchmark 电路 C17 的测试图形向量集 $T = \{00000, 11111, 00001, 10111, 01100, 00011, 01001, 10011, 00101\}$ 得到的最小跳变密度矩阵如图 1a 所示。进一步的研究表明,最小跳变密度矩阵是一种稀疏矩阵,对应的贯通图是无向的 BDD(binary decision diagraph)图,如图 1b 所示。这样,可把寻找最低跳变密度方法的问题转化为最小跳变密度矩阵的排序问题。

按照传统的稀疏矩阵排序算法^[6-8]很难对类似图 1b 所示的 BDD 图找到包含所有节点的贯通图,该贯通图要历经所有的节点,并且每个节点只历经一次,而且总的跳变次数最小。为此,本文提出一种新的算法,其核心是修改最小跳变密度矩阵,使其不但包含与最小跳变相关节点的信息,而且包含与次最小跳变密度相关节点的信息,再计算每个贯通图对应的目标函数,若此值不可再减小,则算法中止;否则,修改最小跳变密度矩阵再次搜索,直至目标函

数值达到要求,流程图如图 2 所示。所谓的目标函数是指每个贯通图中所对应的总的跳变次数。例如,ISCAS85 Benchmark C17 的测试图形向量集 $T = \{00000, 11111, 00001, 10111, 01100, 00011, 01001, 10011, 00101\}$,对应此顺序的序列的跳变密度为 0.75,按照图 2 排序方法形成的序列为 $\{11111, 10111, 10011, 00011, 00001, 00101, 00000, 01001, 1100\}$,跳变密度则降为 0.25。当然,这种排序方法并非惟一,但降低跳变密度十分有效。把本例中排序

$$M = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a)最小跳变密度矩阵



(b)BDD 图

图 1 ISCAS85 Benchmark 电路 C17 的测试图形向量集

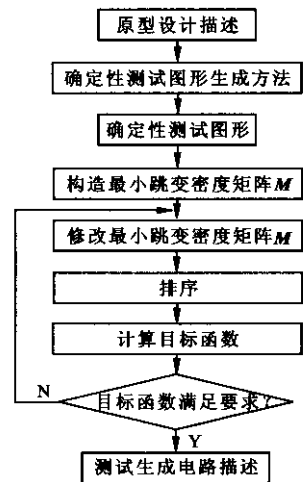
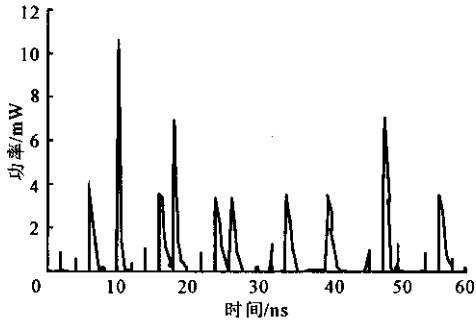
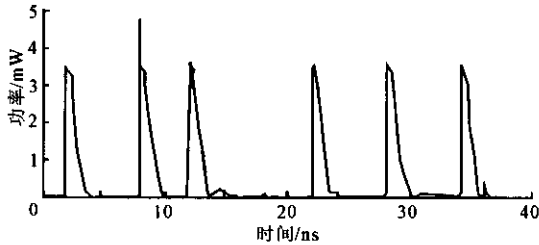


图 2 排序算法流程图

前后的序列施加给 ISCAS85 Benchmark C17 后,按照 TSMC. 35 工艺模型用 SPICE 模拟的动态功耗曲线如图 3 所示. 进一步分析计算表明,排序后功耗大约降低了 28%. 如果采用 TSMC. 18 工艺模型,排序后动态功耗约降低了 32%.



(a) 施加随机测试图形



(b) 低跳变密度排列测试图形

图 3 ISCAS85 Benchmark C17 电路的 SPICE 功耗曲线

2 状态机综合方法研究

对于嵌入式组合逻辑模块,传统的内建自测试方法是基于线性反馈移位寄存器的伪随机生成和压缩电路,存在的问题是测试图形过长、跳变密度大等,而近期得以发展的 RSIC 法则存在种子矢量的植入比较困难,硬件开销过大,且不利于优化和综合等问题. 为此,本文提出用状态机和组合逻辑生成确

定性测试图形的方法. 先比较用不同的状态机所设计的 BIST 电路的性能,对于用 RSIC 法、Moore 法和 Encoding 法设计的输出位数分别为 5、8、16、32 和 41 位的 BIST 电路,分别用 Synopsys 综合工具分析,得到的性能参数如表 1 所示,其中硬件开销是把组合电路部分和时序电路都转换为等效的 2 输入基本逻辑门的个数,最高时钟是指电路可工作的频率,单元参数采用 TSMC. 5 模型.

由表 1 可见:对于输出位数相同的 BIST 电路,用 RSIC 法设计的电路速度最快;用 Moore 状态机设计的电路在硬件开销上有优势;用 Encoding (编码)形式设计的电路在速度和硬件开销方面都比较差. 因此,下面的研究中只讨论 RSIC 法和 Moore 法.

考虑到对于具体的被测电路,RSIC 测试生成电路需要多次植入种子矢量,而基于 Moore 状态机综合的电路不存在这个问题,因此还需要比较测试施加时间.

3 实验与分析

3.1 实验方法

本文以 ISCAS85 Benchmark 电路为对象,从测试功耗、测试电路硬件开销、测试电路可运行最高速度和测试施加时钟数等方面来验证本文所提出的方法的有效性,其中涉及到电路的各种描述语言、设计和分析工具.

实验分析方法是先用 ATPG 工具对 ISCAS85 Benchmark 电路生成确定性测试图形,再用 C 语言和 MATLAB 编写排序算法程序,寻找和分析最大跳变密度和最小跳变密度序列,进而分别生成 BIST 电路的 SPICE 描述和 HDL 描述,最后在 SPICE 工具和 HDL 综合工具下分别编制分析程序,对设计的电路进行功耗、硬件开销、速度和测试时间方面的分析. 整个实验分析流程如图 4 所示.

表 1 状态机设计方法分析

	硬件开销					最高时钟/MHz				
	5 位	8 位	16 位	32 位	41 位	5 位	8 位	16 位	32 位	41 位
RSIC	208	304	960	3 168	3 808	632.0	320.6	818.2	818.2	818.2
Moore	288	528	1 024	2 176	3 056	270.5	470.8	252.5	178.5	242.7
Encoding	160	752	2 122	3 648	—	632.0	224.0	123.0	68.7	—

注:硬件开销为等效的 2 个输入基本逻辑门的个数;最高时钟为电路可工作的频率.

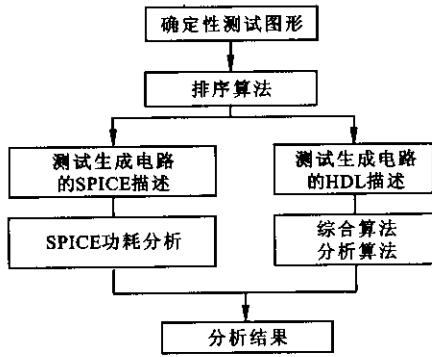


图 4 基于确定性算法的可综合的、较低功耗的测试生成方法的实验分析流程图

3.2 跳变密度算法分析

本文提出的排序方法与采用文献[6,7]的排序方法得出的实验结果列于表 2。由表 2 可见,本文方法找出的序列的最大跳变密度要比文献[6,7]方法大 10%~20%,找出的序列的最小跳变密度要比文献[6,7]的小 10%~20%,由此可认为本文的排序算法可更准确地找出最小跳变密度序列。

由表 3 可知,用本文方法设计的 BIST 电路,电路 C17 和 C1355 的硬件开销比 RSIC 电路的大,但在可接受的增加范围,其余电路的硬件开销都比 RSIC 电路的要减少很多。在速度方面,除 C1355 和 C432 的外,都比相应的 RSIC 电路的要快。在测试时间方面,用本文方法设计的所有 BIST 电路的则比相应的 RSIC 电路的要少得多,一般后者为前者的 10 倍以上,而且从表 2 可以分析出,用本文方法设计的所有 BIST 电路的最小跳变密度也得以减小,因此测试功耗要比 RSIC 小得多。在固定型故障

覆盖率方面,本文方法的要比 RSIC 法的略高一些。

表 2 排序算法实验结果

电路	最大跳变密度		最小跳变密度	
	本文方法	文献[6,7]方法	本文方法	文献[6,7]方法
C17	0.775 0	0.700 0	0.250 0	0.275 0
C1355	0.204 2	0.168 7	0.105 5	0.112 5
C432	0.280 5	0.241 0	0.166 7	0.154 1
C880	0.461 5	0.365 6	0.337 5	0.354 2
C6288	0.502 8	0.420 5	0.247 2	0.264 2
74182	0.456 8	0.444 4	0.333 3	0.358 0
74181	0.668 8	0.655 8	0.370 1	0.396 1

研究 C1355 和 C432 电路发现,相对其他电路,它们的 I/O 和确定性测试图形比较多,采用本文方法设计的 BIST 电路中组合电路的比重比较大,这是本文方法有待改进的地方。

3.3 综合分析

对于使用排序方法后的最小跳变密度序列,先用 Moore 状态机设计,用 HDL 语言描述,再用 Synopsys 综合工具得到硬件开销、速度、测试时间和固定型故障覆盖率方面的参数,如表 3 所示。为了对比,对 RSIC 方法也进行 HDL 语言描述,用 Synopsys 综合工具得到的参数也如表 3 所列。

综合硬件开销、电路可工作最高速度、测试功耗和故障覆盖率诸因素的分析,本文所提出的方法要优于传统的伪随机法和近期发展的 RSIC 法,有望成为一种新的内建自测试方法。

表 3 综合分析实验结果

电路	硬件开销		最高时钟/MHz		测试施加数		固定型故障覆盖率/%	
	RSIC 方法	本文方法	RSIC 方法	本文方法	RSIC 方法	本文方法	RSIC 方法	本文方法
C17	169	224	632.0	1 550.0	10+5	10	100.00	100.00
C1355	3 808	4 064	818.2	242.7	64(2m+1)	129	93.39	99.47
C432	3 488	1 120	818.2	500.4	36(2m+1)	54	93.75	99.12
C880	2 048	1 328	168.5	445.4	60(2m+1)	78	99.80	100.00
C6288	2 112	400	123.0	716.3	32(2m+1)	38	99.34	99.56
74182	320	160	632.0	777.3	9(2m+1)	20	100.00	100.00
74181	560	400	448.8	701.2	14(2m+1)	22	100.00	100.00

注:m 为种子数。

4 结 论

本文提出一种新型的自测试方法,该方法结合了确定性测试和伪随机测试的优点,生成的测试图形具有低功耗、长度短、故障覆盖率高、测试图形自动生成等特点.基于 ISCAS85 Benchmark 电路的实验结果表明,用本文方法设计的 BIST 电路在硬件开销、速度、测试功耗方面优于传统的伪随机测试电路,尤其是测试所需时间显著减少.

参考文献:

- [1] Chatterjee M, Pradhan D K. A BIST pattern generator design for near-perfect fault coverage [J]. IEEE Design and Test of Computers, 2003, 52(12): 1 543-1 557.
- [2] Virazel A, Wunderlich H J. High defect coverage with low-power test sequences in a BIST Environment [J]. IEEE Design and Test of Computers, 2002, 18(6): 44-52.
- [3] Girard P. Survey of low-power testing of VLSI circuits [J]. IEEE Design and Test of Computers, 2002, 19

(3): 82-92.

- [4] David R, Girard P, Landraut C, et al. On Hardware generation of random single-input change test sequences [A]. Proc 6th IEEE European Test Workshop [C]. Los Alamitos, USA: IEEE CS Press, 2001. 117-123.
- [5] Najm F. Transition density: a new measure of activity in digital Circuits [J]. IEEE Trans on CAD, 1993, 12(4): 310-323.
- [6] Amestoy P R, Davis T A, Duff I S. An approximate minimum degree ordering algorithm [J]. SIAM on Matrix Analysis and Applications, 1996, 17(4): 886-905.
- [7] Larimore S I. An approximate minimum degree column ordering algorithm [D]. Gainesville, USA: Dept of Computer and Information Science and Engineering, University of Florida, 1998.
- [8] Karypis G, Kumar V A. Fast and high quality multi-level scheme for partitioning irregular graphs [J]. SIAM on Sci Comput, 1998, 20(1): 359-392.

(编辑 刘 杨)

(上接第 811 页)

为模型,再根据不同的测试准则,可生成满足要求的测试用例.

4 结束语

本文定义了软件体系结构的组成单元(组件)、连接器、端口的结构、信息传递关系.在软件体系结构的动态模型基础上,分析了体系结构单元之间的动态交互关系,建立了动态行为图,并提出了基于结构的测试准则和测试需求.根据不同的体系结构抽象层次,在不同的测试阶段运用不同的覆盖级别,并基于覆盖的路径可生成测试用例.

本文还研究了软件体系结构描述影响软件系统测试的重要意义,这项研究不仅在节省软件测试成本方面有意义,而且可以进一步完善软件的体系结构.

参考文献:

- [1] Medvidovic N, Taylor R N. A classification and com-

parison framework for software architecture description languages [J]. IEEE Trans Software Engineering, 2000, 26(1): 483-491.

- [2] 赵会群. 软件体系结构抽象模型 [J]. 计算机学报, 2002, 25(7): 730-736.
- [3] Bertolino A, Inverardi P, Muccini H, et al. An approach to integration testing based on architectural descriptions [A]. The Third IEEE International Conference on Engineering of Complex Computer Systems, Como, Italy, 1997.
- [4] Richardson D J, Wolf A L. Software testing at the architecture level [A]. The Second International Software Architecture Workshop, San Francisco, USA, 1996.
- [5] 赵会群. 软件体系结构一致性仿真测试 [J]. 东北大学学报, 2001, 22(6): 615-618.
- [6] Jin Zhenyi, Offutt J. Deriving tests from software architectures [A]. 12th International Symposium on Software Reliability Engineering, Hong Kong, 2001.

(编辑 苗 凌)