

FPGA 芯片的链结构 LUT 自测试方法研究

张双悦 李 硕 王 红 杨士元
(清华大学自动化系 北京 100084)

摘 要 基于内建自测试(BIST)思想的 FPGA 测试方法利用被测芯片中的资源来构建测试所需的 TPG 或 ORA,以减少测试对输入输出引脚和外部 ATE 的需求。传统的 FPGA 芯片 BIST 方法仅考虑自测试结构内被配置为 CUT 的资源,从而需要进行多次组测试来完成整个芯片的测试。在现有 LUT 自测试链结构的基础上,通过合理选择 TPG 的电路结构及测试配置,能够在相同测试开销下增加 TPG 部分的故障覆盖率,提高测试效率。

关键词 现场可编程门阵列(FPGA),查找表(LUT),内建自测试(BIST),故障覆盖率
中图分类号 TN791 **文献标识码** A

Study on Chain-based BIST Architecture of LUTs in FPGA

ZHANG Shuang-yue LI Shuo WANG Hong YANG Shi-yuan
(Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract BIST-based FPGA test uses the resource in the chip under test to build the TPG or ORA to reduce the usage of I/O pins and ATE. In traditional BIST methods of FPGA, only the fault detection of the CUT of BIST architectures is considered and the chip needs to be reconfigured several time to change different parts into CUT. In order to improve the efficiency of a chain-based BIST method of LUTs in FPGA, this article tried to find more suitable architecture and testing configurations of TPG which can reach a higher fault coverage of the TPG part.

Keywords Field programmable gate array(FPGA), Lookup table(LUT), Built-in-self test(BIST), Fault coverage

1 引言

FPGA 目前正逐渐取代 ASIC 或者中小规模芯片,得到了广泛的应用。有别于传统数字电路实现方式,目前应用最广泛的 SRAM 型 FPGA 芯片中采用大量可配置 SRAM 和数据选择器来实现可变逻辑功能,但在带来高度灵活性的同时也使得已有的数字电路测试方法不能很好地解决 FPGA 的测试问题。同时,在目前高集成度的 FPGA 芯片中,输入输出引脚相对于丰富的内部资源而言极为有限,因此很多 FPGA 测试问题的研究中都采用了 BIST 的思想来降低测试对芯片外部资源的需求。

在基于 BIST 思想的 FPGA 故障测试方法中,被测 FPGA 芯片中的某些资源被用作构建 BIST 电路,对其余部分的资源进行测试。一些 CLB(LUT)的测试方法将待测 CLB 串联成测试链结构,把 CUT 的故障逐级传播至末端 ORA,以减少测试对测试引脚及片外 ORA 的需求^[1-5];另一些文献则将 FPGA 芯片分为若干个区域,在每个区域内构造较为简单的计数器型 TPG 和异或比较型 ORA 来完成测试^[6-9]。

由于 FPGA 芯片的可配置型,其 BIST 电路只在测试时存在而在测试完成后被重配置为功能电路,因此不会产生额外的测试面积开销。但现有的 BIST 方法通常仅考虑测试算

法对被设置为 CUT 部分的故障覆盖率。而在很多 BIST 算法中,很高比例的芯片资源被用作构建非 CUT 的部分,例如文献[1]设计的测试链中仅有三分之一的 CLB 为 CUT,文献[4,7]所采用的测试结构中 CUT 的比例分别为 50%和 40%。一组完整的测试只针对被配置为 CUT 的芯片资源进行测试。为完成芯片内部所有资源的测试,必须通过多次配置使芯片内不同部分交替成为 CUT。

为了在不影响 CUT 部分的故障覆盖率的前提下,有效提升非 CUT 部分的故障覆盖率,以提高测试的效率,从而降低测试开销,本文针对 FPGA 芯片 BIST 测试链结构中非 CUT 部分的故障覆盖问题展开研究,通过优化测试配置并适当修改测试电路结构的方法,在相同的测试开销下尽可能增加单次测试对所有涉及的芯片资源的故障覆盖率。

2 基于 Partial Chain 的 LUT 测试链

由于 FPGA 进行一次完整配置的时间长,因此目前的研究主要关注于如何减少配置文件的数量。文献[10]提出,对于 n 输入的 LUT,最少需要 $2n$ 个配置才能检测其中的所有 stuck-at 故障、WCR(wrong cell read)故障和 ACR(additional cell read)故障。其中 WCR 故障为存储单元地址错误,即读取了其他地址的值;ACR 故障为存储单元的逻辑值受到同一

到稿日期:2013-09-17 返修日期:2013-11-21

张双悦(1989—),女,硕士生,主要研究方向为 FPGA 的测试与容错技术;李 硕(1989—),男,硕士生,主要研究方向为数字电路的诊断与测试技术;王 红(1971—),女,副教授,主要研究方向为电子技术、检测技术、系统故障诊断理论和技术;杨士元(1945—),男,教授,博士生导师,主要研究方向为电子技术、检测技术、系统故障诊断理论和技术、智能家居系统。

个 LUT 内另一存储单元的影响而表现为逻辑与或逻辑非的形式。图 1 显示了激活 4 输入 LUT 所有上述故障类型的 8 个测试模式。

地址	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
P1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
P2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
P3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
P4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
P5	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
P6	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
P7	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
P8	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

图 1 4 输入 LUT 所需的 8 个测试模式

文献[4,5]提出了一种测试链结构,其能够使测试链在外加时钟下自发产生测试数据并逐级敏化故障至输出端。如图 2(a)所示,该测试链由交替的 TPG 和 CUT 构成,每一对 TPG 和 CUT 构成一个 Partial Chain 单元,前一个 Partial Chain 单元的输出被作为后一个单元的时钟输入。

对于 n 输入 LUT 的 FPGA 芯片来说,TPG 单元与 CUT 单元均由 n 个 LUT 构成。以 4 输入 LUT 为例,图 2 (b) 显示了单个 Partial Chain 单元的内部结构。CUT 中第一个 LUT 直接利用 TPG 的输出作为地址输入,而之后的 LUT 其中一位的地址输入来自于前一个 LUT 的输出。

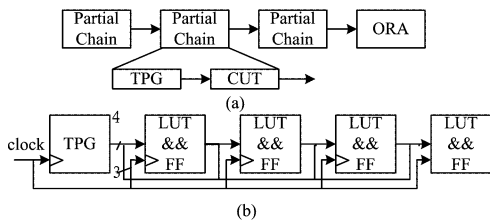


图 2 基于 Partial Chain 的测试结构

在一个测试配置下,Partial Chain 单元中所有属于 CUT 的 LUT 被配置为相同的测试模式,TPG 则产生 $0x0-0xF$ 中的所有地址,对 CUT 的所有存储单元进行读取。TPG 的配置需要保证 CUT 输出与 TPG 输出中的某一位完全相同,以使得 CUT 内所有 LUT 具有相同的地址输入。为了满足 Partial Chain 单元级联的要求,CUT 的输出即与之对应的 TPG 输出中的一位为输入时钟信号的二分频,下文中称 TPG 的该位输出为时钟输出。当电路无故障时,具有 m 个 Partial Chain 单元的测试链的输出为输入时钟信号的 $2m$ 次分频。

地址	无故障	故障
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0
5	0 1 0 1	0 1 0 1
6	0 1 1 0	0 1 1 0
7	0 1 1 1	0 1 1 1
8	1 0 0 0	1 0 0 0
9	1 0 0 1	1 0 0 1
A	1 0 1 0	1 0 1 0
B	1 0 1 1	1 0 1 1
C	1 1 0 0	1 1 0 0
D	1 1 0 1	1 1 0 1
E	1 1 1 0	1 1 1 0
F	1 1 1 1	1 1 1 1

图 3 TPG 的测试配置

在图 1 中 8 个 CUT 测试模式下,TPG 所产生的地址序列可以保持不变,只需要调整其输出线的顺序。例如,当 CUT 采用图 1 中的 P4 / P8 配置时,TPG 使用图 3 所示的配置,顺序产生地址 $0x0-0xF$,TPG 输出的最低位为时钟输出,此时 CUT 中 LUT 的输出 0、1 交替变化,并作为下一个

LUT 地址的最低位;而采用 P3 / P7 配置时,只需要将 TPG 输出线的低两位进行交换,使 TPG 输出的次低位成为时钟输出,即可使 CUT 的输出 0、1 交替变化。

当 CUT 内发生故障时,会造成输出信号在至少一个配置下出现时钟翻转的缺失并影响到下一级的输出,ORA 通过最终输出信号的时钟宽度可以进行故障的检测并将其定位到确定的某一个或某几个 Partial Chain 单元。

但当 TPG 中发生故障时,故障很可能没有被激活或者激活后无法通过测试链进行传播。实验表明,TPG 中的故障只有很小一部分能够在 ORA 中得到检测。

3 Partial Chain 中的 TPG 配置研究

本节对上节所述测试链结构中 TPG 的配置进行了研究,在不影响 BIST 电路总体结构及 CUT 中的故障覆盖的前提下,在相同的配置下对 TPG 中尽可能多的故障进行覆盖。

3.1 TPG 故障的激活与传播

事实上,激活一个 LUT 中的 stuck-at、WCR 和 ACR 故障的条件为同一 LUT 中的任意两个存储单元的逻辑值的 00、01、10 和 11 这 4 种组合需要在至少一个测试模式下出现。满足这一要求的一个测试模式生成方法(即图 1 所示测试模式的生成)是将每一个存储单元地址,即顺序排列的数字 $0 \sim 2^n - 1$ 的每一位作为一个配置得到总共 n 个测试模式,再将这些模式取反得到另 n 个测试模式。交换其中任意两个存储单元的所有测试模式不会对故障的激活产生影响,也就是说,任给一组 $0 \sim 2^n - 1$ 数字的排列作为种子,顺序对应 LUT 每一个存储单元,按上述方法进行测试模式的生成,均可激活所有故障。

在 CUT 中,只需要对所有存储单元进行 0、1 交替读取即可使被激活的故障传播到输出端。但 TPG 的故障传播则较为复杂。

TPG 实际是一个周期为 2^n 的有限状态机电路,当其存储单元中的单故障被激活时会改变状态机的某一状态跳转条件,从而改变其状态循环及周期。例如图 3 右侧 TPG 中一个存储单元发生了 stuck-at 0 故障并被激活,TPG 的输出按 $3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 3$ 周期性变化,而该地址序列经过 CUT 后会产生与无故障电路相同的输出,因此该故障无法得到传播,继而无法被 ORA 检测。

简单分析可知,由于 TPG 产生的地址序列会交替读取 CUT 中的 0 和 1,因此若发生单故障的 TPG 的周期为偶数时,CUT 的输出仍为正常的时钟信号。所以故障能够得到传播的充要条件是发生故障的 TPG 周期为奇数。

3.2 单 TPG 结构的故障覆盖率提升

由于 CUT 的输出实质上仅取决于 TPG 的时钟输出,因此仅有与输出时钟对应的 LUT 内被激活的故障才会使得 TPG 的周期变为奇数,从而传播到输出端。

如图 4 所示,通过改变 TPG 内部连接线的关系,可以在不影响 TPG 对外输出的地址序列的情况下,循环配置其中各个 LUT,每次测试配置中对应时钟输出的 LUT 均不相同,从而提高其中故障被检测的概率。

对于 4 输入 LUT 而言,由于 CUT 的测试至少需要 8 次配置,而满足测试结构要求的地址序列并不唯一,因此可以采

用两个不同的地址序列得到 8 个测试模式,进一步提高 TPG 的故障覆盖率。此时 TPG 内 4 个 LUT 在 8 个测试配置下的测试模式分配如表 1 所列。

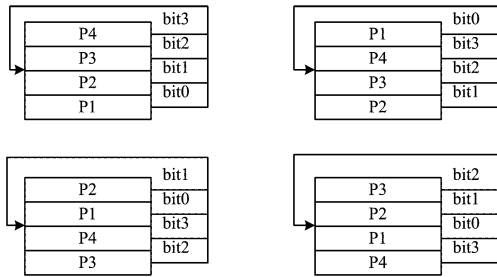


图 4 TPG 内的 LUT 循环配置方法

表 1 改进后的单 TPG 结构中的测试模式分配

	C1	C2	C3	C4	C5	C6	C7	C8
LUT0	P1	P2	P3	P4	P5	P6	P7	P8
LUT1	P2	P3	P4	P1	P6	P7	P8	P5
LUT2	P3	P4	P1	P2	P7	P8	P5	P6
LUT3	P4	P1	P2	P3	P8	P5	P6	P7

上面已经提到,在 TPG 的 8 个测试配置内只有两个配置所激活的故障可以被传输到输出端,因此可以得到相同故障覆盖率的若干 TPG 测试模式。在仿真实验中选取一组测试模式,即 TPG 所存储的两个地址序列:

{1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,0}

{2,3,1,4,6,7,5,8,A,B,9,C,E,F,D,0}

图 5 显示了根据这组地址序列得到的 8 个测试模式,其中 P4 和 P7 对应于时钟输出。

地址	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
P1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
P2	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	0
P3	0	1	1	0	0	1	0	0	1	0	0	0	0	1	1	0
P4	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
P5	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
P6	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0
P7	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
P8	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

图 5 采用改进的单 TPG 结构时的一组测试模式

3.3 双 TPG 结构的故障覆盖率提升

由于 TPG 的输出须含有一位时钟输出,因此在单 TPG 的结构中可以采用的配置模式受到限制,无法激活 TPG 中的所有故障。另一方面,在单 TPG 结构中,仅有与输出时钟对应的 LUT 内被激活的故障才能通过 CUT 传播到输出端,即实际只有两个测试模式中被激活的故障能够被传播到输出端,使得故障覆盖率大大受限。

因此,本文在上述单 TPG 测试配置改进的基础上设计了一种双 TPG 结构(如图 6 所示),该 TPG 由 8 个 LUT 组成,分为两组串联成环状结构,并将其中一组的输出作为 TPG 的输出。

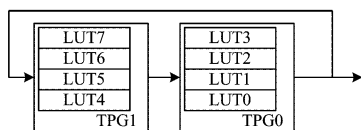


图 6 双 TPG 内部结构

在双 TPG 结构中,可以在满足时钟输出的要求的情况下,通过 8 个测试模式激活 TPG 中 LUT 的所有故障。表 2 给出了采用该结构后不同测试配置下的 TPG 中 LUT 的测试模式分配。

表 2 双 TPG 结构中的测试模式分配

	C1	C2	C3	C4	C5	C6	C7	C8
LUT0	P1	P2	P3	P4	P5	P6	P7	P8
LUT1	P2	P3	P4	P1	P6	P7	P8	P5
LUT2	P3	P4	P1	P2	P7	P8	P5	P6
LUT3	P4	P1	P2	P3	P8	P5	P6	P7
LUT4	P5	P6	P7	P8	P1	P2	P3	P4
LUT5	P6	P7	P8	P5	P2	P3	P4	P1
LUT6	P7	P8	P5	P6	P3	P4	P1	P2
LUT7	P8	P5	P6	P7	P4	P1	P2	P3

此时的故障传播变得较为复杂,通过仿真实验可以得到,该结构中也存在若干相同故障覆盖率的 TPG 测试模式组。其中一组选取其中一组测试模式,其中两个 TPG 所存储的两个地址序列为:

TPG0 : {1,0,E,F,8,9,7,6,4,3,D,A,B,C,2,5}

TPG1 : {7,F,8,0,B,3,4,C,D,6,1,A,2,9,E,5}

图 7 显示了根据这组地址序列得到的 8 个测试模式。

地址	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
P1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
P2	0	0	1	1	0	0	1	1	1	0	1	0	0	1	0	1
P3	0	0	1	1	0	0	1	1	0	1	0	1	1	0	1	0
P4	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
P5	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1
P6	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1
P7	1	1	0	0	1	1	0	0	0	1	0	1	1	0	1	0
P8	1	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1

图 7 采用双 TPG 结构时的一组测试模式

4 实验结果

对原始 Partial Chain 结构及上节中提出的两种方法进行仿真实验,得到不同的方法故障覆盖率,如表 3—表 5 所列。

表 3 原始 TPG 测试配置的故障覆盖率

故障类型	Stuck-at	WCR	ACR
故障总数	128	960	1920
故障检测数	16	128	128
故障覆盖率	12.5%	13.3%	6.7%

表 4 采用改进的单 TPG 结构的故障覆盖率

故障类型	Stuck-at	WCR	ACR
故障总数	128	960	1920
故障检测数	96	768	896
故障覆盖率	75%	80%	46.7%

表 5 采用双 TPG 结构的故障覆盖率

故障类型	Stuck-at	WCR	ACR
故障总数	256	1920	3840
故障检测数	256	1920	3232
故障覆盖率	100%	100%	84.2%

从实验结果可知,本文所提出的两种改进方法在与原方法相同的测试配置数量及测试时间下,能够有效地提高测试链中 TPG 部分的故障覆盖率。其中,采用双 TPG 结构的测试链能够覆盖 TPG 中的全部 Stuck-at 故障和 WCR 故障,即该测试链方法能对所有涉及的芯片资源中的这两种故障进行检测。

结束语 FPGA 的自测试方法存在着很多优点,可以采用较少的芯片引脚和片外 ORA 对较多的芯片资源进行测试,从而提高 FPGA 芯片的并行测试效率,但目前此类方法只对部分被配置为 CUT 的芯片资源进行测试,需要通过循环配置的方式测试全部芯片资源。对于 4 输入 LUT 的 FPGA 芯片,基于 Partial Chain 的 FPGA 自测试链采用 8 个测试配置能够覆盖 CUT 中的全部故障和 TPG 中的少量故障。本文对该测试结构中 TPG 内故障的激活和传播条件进行了

一定的分析,并提出两种不同的 TPG 结构和测试配置不同的改进方法,其在相同的测试开销下显著提高了 TPG 内故障的覆盖率。

由于 TPG 内的故障覆盖率显著提高,测试链中 TPG 与 CUT 的划分被模糊,使得 TPG 在某种程度上也可以被视作 CUT 的一部分,提高了测试方法对所涉及资源的覆盖率。

后续工作将在此基础上,进一步对考虑全部涉及资源的链式 FPGA 自测试方法进行研究,以期得到测试效率更高的自测试方法。

参 考 文 献

- [1] Wei K H, Meyer F J, Xiao-Tao C, et al. Testing configurable LUT-based FPGA's[J]. IEEE Trans. on Very Large Scale Integration (VLSI) Systems, 1998, 6(2): 276-283
- [2] Wei K H, Meyer F J, Lombardi F. An approach for detecting multiple faulty FPGA logic blocks[J]. IEEE Trans. on Computers, 2000, 49(1): 48-54
- [3] Abramovici M, Stroud C E. BIST-based test and diagnosis of FPGA logic blocks[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2001, 9(1): 159-172
- [4] Atoofian E, Navabi Z. A BIST architecture for FPGA look up

table testing reduces reconfigurations[C] // 12th Asian Test Symp(ATS). 2003: 84-89

- [5] Alaghi A, SadoughiYarandi M, Navabi Z. An Optimum ORA BIST for Multiple Fault FPGA Look-Up Table Testing[C] // Test Symposium, 2006(ATS' 06). 15th Asian, 2006
- [6] Nandha Kumar T, Chong C W. An automated approach for locating multiple faulty LUTs in an FPGA[J]. Microelectronics Reliability, 2008, 48(11): 1900-1906
- [7] Chun-Lung H, Ting-Hsuan C. Built-in Self-Test Design for Fault Detection and Fault Diagnosis in SRAM-Based FPGA[J]. IEEE Trans. on Instrumentation and Measurement, 2009, 58(7): 2300-2315
- [8] Abramovici M, Stroud C. BIST-based detection and diagnosis of multiple faults in FPGAs[C] // Proceedings of the IEEE International Test Conference. 2000
- [9] Harris I G, Tessier R. Testing and diagnosis of interconnect faults in cluster-based FPGA architectures[J]. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 2002, 21(11): 1337-1343
- [10] Huang W, et al. Testing memory modules in SRAM-based configurable FPGAs[C] // Proceedings, International Workshop on Memory Technology, Design and Testing, 1997. 1997

(上接第 26 页)

据本文提出的路径约减算法步骤 2 可以约减掉 5 条肯定不会受老化影响的路径,由步骤 3 可以约减掉 6 条肯定不会受老化影响的路径,所以共约减了 s27 电路中 11 条肯定不会受老化影响电路性能的路径,即只需对余下的 13 条路径进行老化防护。

表 1 的实验结果显示,该约减算法对于绝大部分电路的路径约减是十分有效的。当 $q\% = 5\%$ 时,对于 5 年的老化时间,平均可以约减 51% 的路径,对于 10 年的老化时间,平均可以约减 40% 的路径。另外,从表中数据可看出,对于路径数较多的电路,该算法的效率相对更高,因此具有较好的实用性。

结束语 在已有研究中电路老化预测量的约减方法对电路划分不够精确,预测方法的计算量比较大,老化预测效率不够好。本文针对 NBTI 效应对电路老化的影响,对现有的老化模型框架公式进行了改进,并提出了一种基于逻辑门种类对电路老化影响的路径划分和约减方案,即利用老化对于不同门的不同影响程度,对路径中门种类和数目进行时序分析,找出不会由于老化影响导致电路失效的路径。通过本文方案,可以有效地约减电路中不受老化影响电路性能的路径,而实验结果显示通过路径划分及约减,在 5 年老化时间后,平均有至少 50% 的路径可以不用进行老化预测,大大减小了老化预测的工作量,有效节省了老化防护的开销。

参 考 文 献

- [1] 李忠贺,刘红侠,郝跃. 超深亚微米 PMOS 器件的 NBTI 退化机理[J]. 物理学报, 2006, 55(2): 820-824
- [2] Saneyoshi E, Nose K, Mizuno M. A Precise-tracking NBTI-degradation Monitor Independent of NBTI Recovery Effect[C] // Proceedings of the 57th IEEE International Solid-State Circuits Conference Digest of Technical Papers. San Francisco, CA, USA, 2010: 192-193
- [3] Askari S, Nourani M, Rawat M. An on-chip NBTI monitor for estimating analog circuit degradation[C] // VLSI Test Sympo-

sium (VTS), 2012 IEEE 30th, 2012: 68-73

- [4] Borkar S. Electronics beyond nano-scale CMOS [C] // Proc. ACM/IEEE Design Autom. Conf. 2006: 807-808
- [5] Wang Y, Chen X M, Wang W P, et al. On the Efficacy of Input Vector Control to Mitigate NBTI Effects and Leakage Power [C] // Proceedings of the 10th International Symposium on Quality Electronic Design. San Jose, CA, USA, 2009: 19-26
- [6] Vattikonda R, Wang W, Cao Y. Modeling and Minimization of PMOS NBTI Effect for Robust Nanometer Design[C] // Proceedings of the 43th ACM/IEEE Design Automation Conference. San Francisco, CA, USA, 2006: 1047-1052
- [7] Wang Wen-ping, Wei Zi-le, Yang Sheng-qi, et al. Efficient Method to Identify Critical Gates under Circuit Aging[C] // Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference, 2007: 735-740
- [8] 靳松, 韩银和, 李华伟, 等. 一种考虑工作负载的电路老化预测方法[J]. 计算机辅助设计与图形学报, 2010, 22(12): 2242-2249
- [9] Wang Wen-ping, Yang Sheng-qi, Cao Yu. Node Criticality Computation for Circuit Timing Analysis and Optimization under NBTI Effect [C] // Quality Electronic Design, 2008. ISQED 2008. 9th International Symposium, 2008: 763-768
- [10] Lee Y, Kim T. A fine-grained technique of NBTI-aware voltage scaling and body biasing for standard cell based designs[C] // Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific, 2011: 603-608
- [11] Mintarno E, Skaf J, Rui Zheng, et al. Optimized self-tuning for circuit aging Design[C] // Automation & Test in Europe Conference & Exhibition (DATE), 2010: 586-591
- [12] Wang W P, Yang S Q, Bhardwaj S, et al. The Impact of NBTI Effect on Combinational Circuit: Modeling, Simulation, and Analysis [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2010, 18(2): 173-183
- [13] Schroder D K, Babcock J A. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing[J]. Applied Physics, 2003, 94(1): 1-18