

FPGA 互连测试中的反馈桥接故障覆盖问题

毛志明, 张颖, 姚嘉祺, 华屹峰, 杨济中, 陈鑫

(南京航空航天大学 电子信息工程学院, 南京 211100)

摘要: 现场可编程门阵列 (FPGA) 内部资源众多, 其中互连资源出现故障的概率远远高于片内其他资源, 而在以往许多互连测试研究中, 所生成的测试配置存在无法覆盖反馈桥接故障的难题, 所以较难有测试配置实现故障列表的 100% 覆盖; 因此通过约束桥接故障只发生在单个查找表 (LUT) 内的信号线上, 并结合单项函数, 对反馈桥接故障模型进行优化改进, 从根本上解决难题; 然后对优化后的反馈桥接故障设置相应的约束条件, 再使用布尔可满足性理论 (SAT) 生成满足约束条件的测试配置; 采用优化后的故障模型对 ISCAS'89 基准电路进行了测试配置生成实验, 结果表明生成的测试向量解决了反馈桥接故障的覆盖难题, 并且在实现故障列表的 100% 覆盖下, 优化后的故障模型所需要的测试配置数最少。

关键词: 反馈桥接故障; FPGA; 互连测试; 应用相关测试; SAT

Coverage Problem of Feedback-bridging Fault in the FPGA Interconnection Test

MAO Zhiming, ZHANG Ying, Yao Jiaqi, HUA Yifeng, YANG Jizhong, CHEN Xin

(Department of Electronic and Information Engineering, Nanjing University of Aeronautics and

Astronautics, Nanjing 211100, China)

Abstract: There are many internal resources in Field Program Gate Array (FPGA), among which the failure probability of interconnect resources is much higher than that of other on-chip resources. However, in many interconnect test researches, the generated test configurations are hardly able to cover the feedback-bridging faults, so it was difficult to achieve 100% coverage of interconnection fault list. Therefore, in order to fundamentally solve the problem, the feedback bridge fault model is optimized by constraining the bridge faults within a single look-up table (LUT) and by using single-term functions. Then the corresponding constraints are set for the optimized feedback bridge faults and the test configurations meeting, the constraints are generated using the Boolean Satisfiability theory (SAT). Finally, the test configuration generation experiment is carried out by the ISCAS'89 benchmark circuit, the experimental results show that the generated test vectors not only solve the coverage problem of feedback-bridging faults, but also achieve 100% coverage of the fault list with the minimum configuration times.

Keywords: feedback-bridging fault; FPGA; interconnect testing; application-dependent testing; SAT

0 引言

由于 FPGA 的可多次编程、结构灵活和开发周期短等优点, 其应用越来越广^[1], 对器件本身的可靠性要求也不断提高, 所以 FPGA 测试的研究也越来越受重视。FPGA 互连资源占用芯片的大部分面积, 且随着 FPGA 规模和集成度的不断提高, 元器件以及布线的密度越来越大, 使得互连资源在制造和使用过程中出现故障的概率增大。为确保互连资源的可靠性, 国内外已有大量文献对 FPGA 互连资源的测试展开了研究。

文献 [2] 对现有测试方法进行了完整的综述, 该文献中将测试方法大致分为应用无关测试和应用相关测试。

国外对 FPGA 的测试研究较早, 技术也很成熟, 主要在研究如何实现最少的测试配置数和测试时间^[3-7], 而国内由于专利壁垒造成的技术垄断^[8], 研究起步较晚, 且研究主要是基于应用无关测试^[9-13], 应用无关互连测试考虑的是整片 FPGA 的互连资源, 其测试目的是保证整片 FPGA 互连资源的正确性, 而通常在设计中使用不到所有的互连资源, 因此应用无关测试的效率相对较低; 应用相关互连测试只需考虑设计中所使用的互连资源, 无需测试整片 FPGA 的互连资源, 效率更高, 且能在测试出互连故障后, 在设计中对故障进行容错, 从而使用户定义的设计不受影响。

在现有的应用相关互连测试方法中, 文献 [14] 提出

收稿日期: 2021-09-06; 修回日期: 2021-10-15。

基金项目: 国家自然科学基金项目 (61701228, 61106029); 模拟集成电路重点实验室基金项目 (61428020304); 航空科学基金 (20180852005)。

作者简介: 毛志明 (1997-), 男, 江西抚州人, 在读硕士研究生, 主要从事 FPGA 测试、诊断以及容错方向的研究。

通讯作者: 张颖 (1977-), 女, 江苏徐州人, 博士, 讲师, 主要从事集成电路设计与测试、硬件安全方向的研究。

引用格式: 毛志明, 张颖, 姚嘉祺, 等. FPGA 互连测试中的反馈桥接故障覆盖问题[J]. 计算机测量与控制, 2022, 30(4): 23-28, 49.

通过修改逻辑块配置并保留互连配置来测试互连资源，并实现互连资源的固定 0/1 故障的覆盖测试。随后文献 [15] 中考虑到互连中的无反馈桥接故障，使用单项函数，并结合“沃尔什码”实现固定故障和无反馈桥接故障的测试。测试 FPGA 所花费的时间大多由测试配置的数量决定，因此应用相关互连测试的主要目标是通过最小化测试配置的总数来减少测试时间。文献 [16] 中提出只使用一种测试配置，在该配置下施加多次测试向量。文献 [17] 中提出使用一个激活输入连接多个网络，实现测试配置以及激活输入的减少，但该方案需要设计电路具有较好的连通性，即存在较大扇出数。随后文献 [18] 中指出现有的应用相关测试技术的限制性问题，并对互连故障类型进行了进一步的补充，提出“主导与”无反馈桥接故障及“主导或”无反馈桥接故障，文中还首次对反馈桥接故障进行了定义。文献 [19] 中使用 SAT 对故障进行测试配置生成，其中针对反馈桥接故障，借鉴了文献 [18] 中的定义，但文中未给出使用该定义约束生成的测试向量实例。

针对上述互连测试文章中存在的反馈桥接故障的覆盖难题，本文只考虑单个 LUT 内的桥接故障，并结合单项函数的使用，对反馈桥接故障模型进行优化，再对相应故障设计约束条件，最后使用 SAT 求解器生成符合约束条件的测试配置。在 ISCAS'89 基准电路上进行的测试配置生成实验表明生成的测试向量不仅成功覆盖反馈桥接故障，而且所生成的测试配置使用了最少的配置次数实现了故障列表的 100% 覆盖。

1 应用相关互连测试准备工作

FPGA 互连资源主要分为 CLB 内的互连资源和 CLB 间的互连资源。CLB 内的互连资源位于每个 CLB 内部，其测试可以同 CLB 内的逻辑资源测试同时进行，不在这里讨论。CLB 间的互连资源实现了 CLB 与 CLB 之间的路由，多为 LUT 与 LUT 的连接，本文讨论的 CLB 间的互连资源测试，主要考虑 LUT 与 LUT 的连接测试，若原始设计中使用了进位链、多路复用器等资源，将绕过这些部分。对于 CLB 间的互连资源测试，可通过修改逻辑资源配置，保持路由资源的配置不变进行测试，具体测试过程将在下文给出。

1.1 单项函数

互连测试在研究测试方法之前首先要考虑故障在测试过程中的传播。单项函数是一个只含有最大项或最小项的逻辑函数式，在其真值表中，只有一组向量的输出值与其他向量值不同，该组向量被称之为激活输入。图 1 中展示了一个 4 输入 LUT 单项函数和其对应的激活输入，该单项函数只有一个最大项，在施加了对应的激活输入 (1001) 之后，相应信号线间的敏感故障会被覆盖激活，若此时信号线 A 存在固定 0 故障，输入会由 1001 变为 0001，输出则会由 0 变为 1，于是在 LUT 输出处就会发现输出响应不匹配，并将错误的输出向后传播，最终到达观测点。使用单项函数可以保证电路所有敏感故障的检测，并解决了故障向后传播的问题。

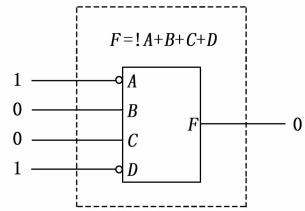


图 1 应用激活输入的单项函数

1.2 故障列表

最早的互连测试只考虑了固定故障，而后对桥接故障进行了分析建模，由“线与”、“线或”桥接发展到“主导”、“主导与”以及“主导或”桥接，最后又增加了对反馈桥接故障的分析，因此目前给出的互连资源故障列表包括固定故障，“主导”、“主导与”、“主导或”桥接故障以及反馈桥接故障。

“主导”桥接只要两信号线逻辑值不同就会激活故障；“主导与”桥接在两信号线逻辑值不同且主导信号线的逻辑值为“0”时激活；“主导或”桥接在两信号线逻辑值不同且主导信号线的逻辑值为“1”时激活，相关故障的真值表在表 1 中给出。表中第 1 行展示的是无故障时的真值情况，而其他行都存在相应的故障，表中加粗带下划线的逻辑值为故障所导致的变化。

表 1 互连故障真值表

AB	00	01	10	11
AB(A 固定 0)	00	01	<u>00</u>	<u>01</u>
AB(A 固定 1)	<u>10</u>	<u>11</u>	10	11
A 主导 B	00	<u>00</u>	<u>11</u>	11
B 主导 A	00	<u>11</u>	<u>00</u>	11
A 主导与 B	00	<u>00</u>	10	11
B 主导与 A	00	01	<u>00</u>	11
A 主导或 B	00	01	<u>11</u>	11
B 主导或 A	00	<u>11</u>	10	11

2 故障模型改进优化

文献 [20] 中将桥接故障分为无反馈和反馈两种类型，在图 2 (a) 中虚线展示了无反馈桥接故障，发生在输入与输入之间，不存在通路；图 2 (b) 中虚线展示了反馈桥接故障，发生在输入与输出之间，存在反馈通路。而在文献 [18-19] 对反馈桥接故障的研究中，都只考虑到输入通过意外桥接通路影响输出的情况，而未考虑到输出通过意外桥接通路影响输入。

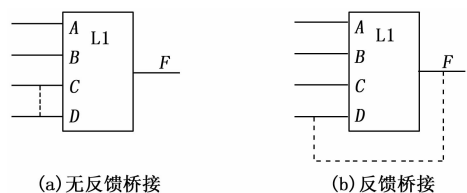


图 2 桥接故障

现有文献研究反馈桥接故障时考虑了任意信号线之间的可能性, 而在 FPGA 设计中, 考虑所有信号线之间的桥接故障是不合适的, 因为这样不仅使故障的数量变得十分庞大, 使整个电路的状态分析变得十分困难, 增加了测试的难度, 并且基于 FPGA 的布局布线信息, 一些物理距离相对较远的信号线之间不可能发生桥接故障。

为了优化对反馈桥接故障的测试, 本文将反馈桥接故障划分为影响输出型桥接和反馈输入型桥接, 并且只考虑单个 LUT 内的反馈桥接故障, 其中电路中存在的桥接故障数与 LUT 的规格、数量有关。下面在结合单项函数的使用下对这两种类型的反馈桥接故障进行故障测试现象分析。

影响输出型桥接表现为输入信号线 D 通过意外通路导致输出信号线 F 发生变化; 反馈输入型桥接表现为输出信号线 F 通过意外通路导致输入信号 D 发生变化。结合单项函数的使用, 在图 3 中针对反馈桥接的两类故障的故障现象进行分析。图 3 (a) 将影响输出型桥接故障分为“主导”桥接、“主导与”以及“主导或”桥接。当存在“主导”桥接故障时, 若 D 和 F 逻辑值相同, 则无法观察到故障现象, 也就无法检测出 D 主导 F 桥接, 若 D 和 F 的逻辑值不相同, 如图 3 (a) 中描述的例①, 当 D 为 0 (1), F 为 1 (0) 时, 若存在“主导”桥接故障, 输入信号线 D 通过意外桥接环路使得输出线 F 的逻辑值发生变化, 即由 1 (0) 变为 0 (1), 而这种变化可以通过单项函数传播到最终输出, 从而可以检测到影响输出型“主导”反馈桥接故障的存在。“主导或”桥接的逻辑值变化由主导信号线的逻辑值 1 主导, 图 3 (a) 中的例②描述了在 D 为 1, F 为 0 时, 若存在 D “主导或” F 桥接故障, D 会导致 F 由 0 变为 1, 结合单项函数的故障传播机制, 使得在后续的观测点能观测到故障现象, 同样能够检测到故障的存在。“主导与”桥接的逻辑值变化由主导信号线的逻辑值 0 主导, 图 3 (a) 中的例③描述了在 D 为 0, F 为 1 时, 若存在 D “主导与” F 桥接故障, D 会导致 F 由 1 变为 0, 同理也能检测到故障的存在。

在图 3 (b) 中分析了反馈输入型桥接故障的故障现象检测。同样将反馈输入型桥接分为“主导”桥接、“主导或”和“主导与”桥接。F “主导” D 桥接故障的现象只有在 D 和 F 的逻辑值不等时才能被激活, 图 3 (b) 中例①描述了 D 和 F 的逻辑值不相等的两种情况, 当 D 为 0, F 为 1 时, 若存在 F “主导” D 桥接故障, 输出信号线 F 通过意外桥接环路使得输入 D 的逻辑值发生变化, 即 D 会由 0 变为 1, 但输入 D 发生改变后, 在使用单项函数的情况下, 此时的输入向量与单项函数的激活输入不同, 因此输出 F 会发生改变, 即由 1 变为 0。这时 D 为 1, F 为 0, D 和 F 的值又不相同, “主导”桥接故障激活, F 又会使 D 发生由 1 变为 0 的变化, 这时输入恢复为原来的激活输入, 导致 F 又会发生改变, 重复以上变化, 将形成振荡。当 D 为 1, F 为 0 时, 同理可得, 故障现象也表现为逻辑值振荡, 虽然反馈输入型“主导”桥接会导致电路发生振荡, 但是由于单项函数的存在, 故障的振荡现象会向后传播, 在最后的观测

点可以观测到, 所以能检测反馈输入型“主导”桥接故障的存在。图 3 (b) 的例②和例③分别分析了 F “主导或” D 桥接故障以及 F “主导与” D 桥接故障, 其中例②故障发生的逻辑变化由 F 的逻辑 1 主导, 例③故障发生的逻辑变化由 F 的逻辑值 0 主导, 对应故障现象的分析步骤与上述相同, 且都能在输出端检测到故障的存在, 但反馈输入型“主导与”桥接、“主导或”桥接并不会发生反馈输入型“主导”桥接的反复振荡现象。

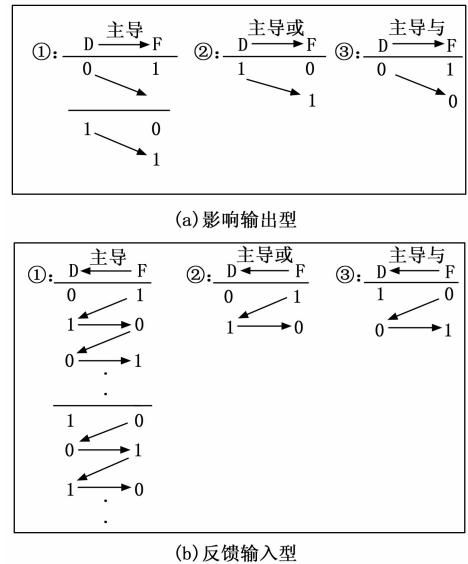


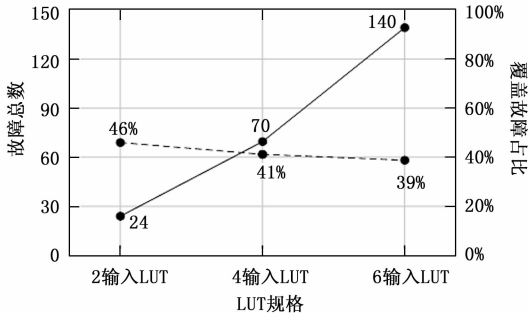
图 3 反馈桥接故障测试现象分析

从上述对优化的故障模型进行分析中, 可以看出使用单项函数并施加特定的测试向量能够很好地检测到反馈桥接故障的存在, 相比于以往文献, 该优化的故障模型既解决了数量庞大的故障列表问题, 又优化了反馈桥接故障存在的测试难题。表 2 中给出了单个 4 输入 LUT 存在的所有互连故障。

表 2 单个 4 输入 LUT 的故障列表

		固定故障		桥接故障			
A	固定 0	固定 1	A 主导 B	A 主导 C	A 主导 D	A 主导 F	
			A 主导与 B	A 主导与 C	A 主导与 D	A 主导与 F	
			A 主导或 B	A 主导或 C	A 主导或 D	A 主导或 F	
B	固定 0	固定 1	B 主导 A	B 主导 C	B 主导 D	B 主导 F	
			B 主导与 A	B 主导与 C	B 主导与 D	B 主导与 F	
			B 主导或 A	B 主导或 C	B 主导或 D	B 主导或 F	
C	固定 0	固定 1	C 主导 A	C 主导 B	C 主导 D	C 主导 F	
			C 主导与 A	C 主导与 B	C 主导与 D	C 主导与 F	
			C 主导或 A	C 主导或 B	C 主导或 D	C 主导或 F	
D	固定 0	固定 1	D 主导 A	D 主导 B	D 主导 C	D 主导 F	
			D 主导与 A	D 主导与 B	D 主导与 C	D 主导与 F	
			D 主导或 A	D 主导或 B	D 主导或 C	D 主导或 F	
F	固定 0	固定 1	F 主导 A	F 主导 B	F 主导 C	F 主导 D	
			F 主导与 A	F 主导与 B	F 主导与 C	F 主导与 D	
			F 主导或 A	F 主导或 B	F 主导或 C	F 主导或 D	

不同规格的 LUT 由于输入信号线数量的不同，因此所存在的互连故障数量会不相同，如图 4 所示，图中实线描绘了单个不同规格的 LUT 所存在的互连故障总数，虚线描绘了单个配置下能覆盖到的最多的互连故障数占单个 LUT 总故障数的百分比情况。从图 4 中可看出含输入信号线数量越多的 LUT，所存在的互连故障越多，而随着 LUT 输入信号线的增多，单个配置能覆盖到的最多的故障数占相应故障总数的比重逐渐减小。针对反馈桥接故障所优化的故障模型适用于任意规格的 LUT，且对于具体的电路映射在不同型号的 FPGA 上，测试该电路所需的互连测试配置数与该 FPGA 的 LUT 规格呈反比趋势。



注：实线为单个 LUT 的故障总数；
虚线为单个配置所覆盖到的最多故障占比。

图 4 不同规格 LUT 的故障情况

3 测试向量生成

在对反馈桥接故障模型进行改进优化之后，再对整个互连故障列表进行测试向量约束，即给出覆盖到上述所有故障应该满足的条件，并用布尔等式或不等式对这些条件进行描述的过程。而通过使用 SAT，可知晓是否存在这样的测试配置满足上述的布尔等式约束，若存在，则使用 SAT 求解器进行求解，从而生成测试配置，最终目的是使用最少数量的测试配置并覆盖到所有的故障。所使用的 SAT 是一类用来发现是否有任何值可以满足公式表示的约束的问题。来自不同领域的许多问题的表述都可以使用 SAT 来完成。用布尔公式表示的问题可用布尔可满足性来解决，而需要非布尔值的问题可以使用可满足性模理论 (SMT) 来表述^[21]。因此，测试向量生成问题可以被表述为一组约束条件，并使用 SAT 求解器来求解满足约束条件的测试向量。为了构建测试电路，将 SAT 求解器算出的满足约束条件的测试向量，即分配给每个 LUT 信号线的值，充当对应 LUT 的激活输入，这些值也决定了在每个 LUT 上实现的单项函数功能。

1) 固定故障约束：为了检测互连网络中的特定故障，测试向量首先应该激活故障，因此检测固定故障，就需要施加激活固定故障的测试向量。检测固定 0 故障，需要施加逻辑值“1”；检测固定 1 故障，需要施加逻辑值“0”。在一个应用设计中存在许多互连信号线，这些信号线无法在一次测试配置中完全覆盖所有信号线的固定故障，因此需

要多次测试配置，所以固定故障的约束条件为至少存在一种配置，使得某一条被测互连线被施加逻辑值“0”，存在另一种配置使得同一条被测互连线被施加逻辑值“1”，即：

$$\exists (x, y), \text{使得 } V_n^x \neq V_n^y \text{ 且 } x \neq y$$

其中： V_n 代表一个 LUT 的第 n_i 条输入信号线的值， x 和 y 代表配置号。

2) 无反馈桥接故障约束：无反馈桥接故障主要发生在 LUT 输入端的成对信号线中，只有在同一个测试配置中对相应的成对信号线施加相反的逻辑值时，才能检测到无反馈桥接故障的存在。如图 2 中 C 与 D 存在的无反馈桥接故障，而桥接故障又分为“主导”型、“主导与”型、“主导或”型，要激活对应的故障，只需在同一个测试配置中对 C 和 D 施加不同的逻辑值。因此，对于某一个 LUT 的无反馈桥接约束如下：

主导型： $\exists (i, j), \text{使得 } V_n^x \neq V_n^y \text{ 且 } i \neq j。$

主导与型： $\exists (i, j), \text{使得 } V_n^x \neq V_n^y, i \neq j, \text{ 且 } V_n^x = 0, V_n^y = 1。$

主导或型： $\exists (i, j), \text{使得 } V_n^x \neq V_n^y, i \neq j, \text{ 且 } V_n^x = 0, V_n^y = 0。$

其中： n_i 和 n_j 代表该 LUT 的两个不同输入信号线， n_i 为主导方， n_j 为被主导方， V_n^x 和 V_n^y 代表相应的逻辑值， x 为配置号。

3) 反馈桥接故障约束：文献 [18] 中对反馈桥接故障的测试进行了一系列的等式约束，但该约束条件只考虑了输入影响输出的情况；在文献 [19] 中对反馈桥接故障进行测试约束时，借鉴了文献 [18] 中的约束条件，但在该文中并未举例说明反馈桥接故障的测试配置结果。而本文只考虑单个 LUT 内的反馈桥接故障，并将反馈桥接分为影响输出型和反馈输入型，对应的约束条件相比于以往研究中所使用的一系列等式约束更加简捷方便，使用简单的测试向量就能够覆盖反馈桥接故障的各种类型。其原理与无反馈桥接故障约束类似，若要检测图 2 中的 D 与 F 之间存在的反馈桥接故障，只需要在同一配置中对 D 和 F 施加不同的逻辑值，以激活不同类型的反馈桥接故障。因此，对于某一个 LUT 的影响输出型桥接的约束如下：

主导型： $\exists (i, f), \text{使得 } V_n^x \neq V_n^y, \text{ 且 } i \neq f。$

主导与型： $\exists (i, f), \text{使得 } V_n^x \neq V_n^y, i \neq f, \text{ 且 } V_n^x = 0, V_n^y = 1。$

主导或型： $\exists (i, f), \text{使得 } V_n^x \neq V_n^y, i \neq f, \text{ 且 } V_n^x = 1, V_n^y = 0。$

其中： n_i 代表该 LUT 的第 n_i 条输入信号线， n_f 代表该 LUT 的输出线， V_n^x 和 V_n^y 代表其相应的逻辑值， n_i 为主导方， n_f 为被主导方， x 为配置号。反馈输入型桥接与影响输出型桥接的区别在于主导和被主导的对象相反，其约束与影响输出型桥接约束同理，不再赘述。

4) 赋值冲突约束：对于特定的设计，其中的 LUT 之间的错综复杂连接，会使得同一条信号线分别连接到多个 LUT，而在上述对不同故障类型的激活约束过程中，可能

会使得同一条信号线被赋值为多个逻辑值, 即某条信号线在同一测试配置中被同时赋值“0”和“1”两种逻辑值, 所以对于此类的赋值冲突需要进行避免。赋值冲突约束如下:

若 $\exists n_i^{lut1} = n_i^{lut2}$, 则必须使得 $V_{n_i}^{lut1} = V_{n_i}^{lut2}$ 成立。

其中: n_i^{lut1} 和 n_i^{lut2} 表示相同的信号线连接到两个不同的 LUT, $V_{n_i}^{lut1}$ 和 $V_{n_i}^{lut2}$ 代表相应的信号逻辑值。

为了覆盖电路中所有故障的测试, 生成的测试向量必须满足上述所有的约束条件, 使用 SAT 求解器算出满足约束条件的测试配置。针对图 5 所示的电路, 使用上述的约束条件进行测试向量生成, 在表 3 中展示了所提出的方法生成的测试配置和对应的单项函数, 考虑到文章篇幅, 未在表 3 中列出对应测试配置所覆盖的故障, 但结合文章上述对互连故障的介绍, 可以分析出表 3 中所生成的测试配置不仅覆盖到所有固定故障和无反馈桥接故障, 而且还覆盖到所有的反馈桥接故障, 成功解决了以往研究中的反馈桥接覆盖难题。上述生成的测试配置所覆盖的故障数占总故障数的百分比折线如图 6 所示。其中使用 SAT 求解器生成的第一次测试配置覆盖了 41% 的故障, 在第二次覆盖 69% 的故障, 第三和第四次分别覆盖了 89% 和 100% 的故障。

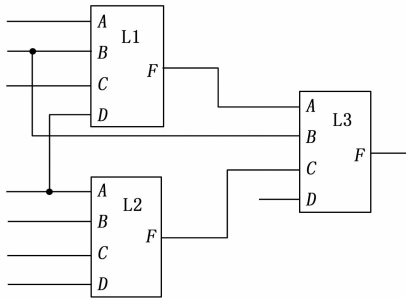


图 5 示例电路

表 3 针对图 6 电路生成的测试配置及对应的单项函数

LUT	LUT 输入	测试向量	生成的测试配置以及对应的 LUT 单项函数			
			配置 1	配置 2	配置 3	配置 4
L1	A	0011	00101	01110	10010	11001
	B	0101	F = A' · B' · C · D'	F = A + B' + C' + D'	F = A' + B + C + D'	F = A · B · C' · D'
	C	1100				
	D	0110				
	F	1001				
L2	A	0110	01110	10100	10011	01001
	B	1001	F = A + B' + C' + D'	F = A' + B + C' + D	F = A · B' · C' · D	F = A' · B · C' · D'
	C	1100				
	D	1010				
	F	0011				
L3	A	1001	10010	01001	00111	11100
	B	0101	F = A' + B + C + D'	F = A' · B · C' · D'	F = A' · B' · C · D	F = A' + B' + C' + D
	C	0011				
	D	1010				
	F	0110				

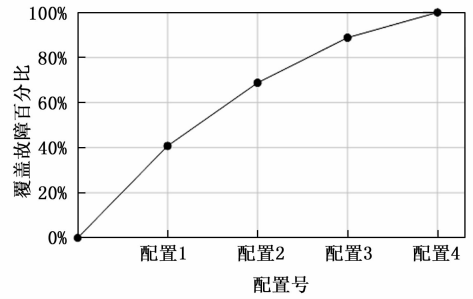


图 6 测试配置所覆盖的故障数占总故障数的百分比

4 仿真与分析

针对上述优化后的反馈桥接故障模型, 在 ISCAS'89 基准时序电路上进行测试配置生成实验, 本文所有实验都是在一个 8 GB 内存的 Window 系统上进行的。为了与先前研究结果进行对比, 选择了先前文献 [19] 中相同的 20 个基准电路, 并将这些电路映射到相同型号的 Xilinx Virtex4 (XC4VLX100) FPGA 上, 表 4 展示了这些基准电路所使用的资源数量, 以及相应互连资源所存在的故障数量。

表 4 ISCAS'89 基准电路实现在 Virtex-4 上的结果

基准电路	LUT 总数	Net 总数	总故障个数
S298	28	48	1 272
S344	37	73	2 847
S349	37	73	2 847
S382	44	75	3 000
S400	44	76	3 078
S420	37	76	3 078
S444	45	78	3 237
S510	78	121	7 623
S526	43	76	3 078
S641	67	120	7 500
S713	71	126	8 253
S820	107	145	10 875
S1196	200	257	33 667
S1238	209	277	39 057
S1423	195	316	50 718
S1488	236	296	44 548
S1494	235	292	43 362
S5378	383	664	222 108
S9234	291	495	123 750
S15850	928	1 502	1 131 757

在整个实验过程中, 首先选择对应型号的 FPGA 建立工程, 并对基准电路进行综合, 使用 Python 脚本对综合后生成的网表文件提取电路的连接关系; 再根据电路的连接关系对电路互连线进行故障约束以及赋值冲突约束, 并使用 Python 自带的库 SAT 求解器: Z3^[22], 生成满足上述约束的测试配置, 将生成的测试向量作为 LUT 的激活输入, 计算出对应的单项函数。生成测试配置后, 需要将原始电路修改为测试电路, 即将原始电路中 LUT 的逻辑功能修改为测试配置中的单项函数。为了修改 LUT 的内容, 使用

“xdl-ncd2xdl * *.ncd”命令将基准电路经过映射后所生成的本地电路描述(NCD)文件转成赛灵思设计语言(XDL)文件,其中XDL文件是可读文本文件。同样使用脚本语言对XDL文件中LUT的逻辑功能进行批量修改,再使用“xdl-xdl2ncd * *.xdl”命令将修改好的XDL文件转回NCD文件,得到了对应的测试电路,最后施加测试向量进行测试分析,表5中列出了对应的基准电路所需要的测试配置数。

表5 基准电路所需要的测试配置结果

基准电路	配置生成时间/s	配置次数	配置次数 ^[19]
S298	0.172	5	5
S344	0.235	5	5
S349	0.237	5	5
S382	0.283	5	5
S400	0.331	5	5
S420	0.249	5	5
S444	0.284	5	6
S510	0.481	5	6
S526	0.335	5	6
S641	0.451	5	6
S713	0.470	5	6
S820	0.793	5	7
S1196	1.346	5	7
S1238	1.615	5	9
S1423	1.384	5	9
S1488	1.858	5	9
S1494	1.891	5	11
S5378	3.575	5	11
S9234	2.128	5	12
S15850	9.942	5	12

在文献[15]中使用“沃尔什码”生成测试配置的方法测试 n 条信号线需要 $\log_2(n+2)$ 次配置;而在文献[18]中指出“沃尔什码”生成的配置存在无法完备覆盖“主导与”、“主导或”故障的问题,并提出在 $2 \leq n < 7$ 时,需要额外一个配置实现故障列表的100%覆盖。而本文因为只考虑了单个LUT内的桥接故障,需要对LUT与LUT之间的连接进行赋值冲突约束,对于较为复杂的电路连接关系就需要额外的1个配置处理赋值冲突,所以针对本文所描述的优化故障模型,需要最多 $\log_2(n+2)$ 次配置才能实现故障列表的100%覆盖,其中 n 为单个LUT的信号线数。

而本文是在Virtex4上进行测试配置生成实验,其中Virtex4的LUT规格为4输入,按上述分析得出实现故障列表的100%覆盖所需要的最大配置数为5,与表5中所展示的生成实验的结果相吻合,表明了本文分析的最大配置数的正确性。本文的主要贡献是解决了反馈桥接故障的覆盖难题,为了更加公平地进行结果比对,在表5中只列出了文献[19]中的测试配置次数,而未列出现有研究中未覆盖反馈桥接故障的实验结果。文献[19]针对反馈桥接故障进行的测试配置生成实验,其所使用的测试生成

方法与本文相同,但该文针对反馈桥接故障的约束条件借鉴了文献[18]中的对反馈桥接故障的定义。因此,将本文实验数据与文献[19]的数据进行比较最能表明本文优化后的反馈桥接故障模型的可行性。从表5中可看出,本文与文献[19]所生成的测试配置次数在前几个基准电路上持平,而在后续的基准电路上,本文实现了测试配置数量的减少,而测试FPGA所花费的时间主要取决于测试配置的数量。上述结果表明本文所做的反馈桥接故障模型优化和改进不仅成功地解决了反馈桥接故障的覆盖问题,还实现了测试配置数的减少,对提升应用相关的FPGA互连资源测试的效率具有重要的意义。

5 结束语

本文针对以往研究中所生成的测试配置存在的反馈桥接故障覆盖难题进行了研究,分析了以往研究中的不足,并对反馈桥接故障模型进行优化改进,而后对固定故障、无反馈桥接故障以及优化后的反馈桥接故障进行覆盖约束,并用SAT求解器生成满足约束条件的互连测试配置,最后在ISCAS'89基准电路上进行测试配置生成实验,生成的测试向量成功解决反馈桥接故障的覆盖难题,实现了故障列表的100%覆盖,且与先前研究相比,所生成的测试配置次数实现了最小化。为提高了在FPGA上实现的任意设计的可靠性,不仅需要生成测试配置进行故障测试,后续还需要相应的诊断配置实现进行故障定位以及故障容错工作。本文所做的应用相关互连测试配置生成研究旨在解决FPGA互连资源的测试配置的故障列表覆盖问题,为完成对设计的测试以及后续互连故障定位和互连故障容错等研究奠定了基础。

参考文献:

- [1] 智研咨询. 2019年中国FPGA市场现状分析、FPGA芯片的应用及未来发展前景分析 [EB/OL]. Available in <http://www.chyxx.com/industry/201910/793680.html>, 2019.
- [2] 张颖, 毛志明, 陈鑫. 基于静态随机存取存储器型FPGA的测试技术发展 [J]. 电子与封装, 2021, 21(1): 37-47.
- [3] STROUD C, KONALA S, CHEN P, et al. Built-in self-test of logic blocks in FPGAs (Finally, a free lunch: BIST without overhead!) [C] // Proceedings of 14th VLSI Test Symposium, USA: IEEE, 1996: 387-392.
- [4] RENOVELL M, FIGUERAS J, ZORIAN Y. Test of RAM-based FPGA: methodology and application to the interconnect [C] // Proceedings of 15th IEEE VLSI Test Symposium, USA: IEEE, 1997: 230-237.
- [5] RENOVELL M, PORTAL J M, FIGUERAS J, et al. Testing the interconnect of RAM-based FPGAs [J]. IEEE Design & Test of Computers, 1998, 15(1): 45-50.
- [6] YU Y L, XU J, HUANG W K, et al. A diagnosis method for interconnects in SRAM based FPGAs [C] // Proceedings Seventh Asian Test Symposium, Singapore: IEEE, 1998: 278-282.

(下转第49页)