DOI:10.11918/j.issn.0367-6234.201611138

加速硬件木马检测方法研究

徐 力[1], 吴新春[1], 周 彬[2], 叶文霞[1]

(1.西南交通大学 信息科学与技术学院,成都 611756;2.哈尔滨工业大学 空间基础科学研究中心,哈尔滨 150001)

摘 要: 为有效检测出芯片在设计和外包制造过程中是否被插入硬件木马电路, 提出一种在芯片设计阶段插入二选一数据选择器(MUX)来提高电路节点转移概率的方法. 即在电路中转移概率低于转移概率阈值的候选节点的主要输入端插入 MUX 来提高相关节点的转移概率, 从而实现加速电路中硬件木马的检测. 通过对扇出锥和电路逻辑拓扑结构的分析,选择对整个电路转移概率影响最大的节点作为候选节点,实现对 MUX 插入算法的优化,从而减少 MUX 的插入数量. 同时增加关键路径延时限制,避免电路关键路径延迟超过预先设定的阈值. 将预先设计的硬件木马电路的输入端插入在电路中转移概率较小的节点,并向电路输入端输入激励信号,分析计算在 MUX 插入前后电路转移概率变化以及硬件木马电路的激活概率. ISCAS'89 基准电路的实验结果表明:在插入 MUX 之后,电路整体转移概率显著提高,电路中转移概率小于转移概率阈值的节点数明显降低;被插入在电路中的硬件木马被激活的概率显著提高;电路关键路径延时增加百分比控制在预先设定的比例因子之内.

关键词: 二选一数据选择器;硬件木马;转移概率;路径延时

中图分类号: TN407 **文献标志码**: A **文章编号**: 0367-6234(2017)11-0137-06

Study on acceleration of hardware Trojan detection

XU Li[1], WU Xinchun[1], ZHOU Bin[2], YE Wenxia[1]

(1.The School of Information Science and Technology, Southwest Jiao Tong University, Chengdu 611756, China;
2.Research Center of Basic Space Science, Harbin Institute of Technology, Harbin 150001, China)

Abstract: In order to effectively detect whether the chip is inserted into the hardware Trojan circuit during the design and manufacturing process, a method is proposed to increase the transition probability of the circuit nodes by inserting 2-to-1 MUXs in the chip design stage. The main input of the candidate node whose transition probability is lower than the transition probability threshold is inserted into the MUX to improve the transition probability of the relevant nodes, so as to realize acceleration of hardware Trojan detection in the circuit. The optimization of the insertion algorithm is realized by analyzing the fan-out cone and logic topology, and the node with the greatest influence on the transition probability of the whole circuit is selected as the candidate node, thus the number of MUXs insertion is reduced. Meanwhile, the critical path delay limit is increased to avoid the critical path delay of the circuit exceeding the preset threshold. The input terminals of the pre-designed hardware Trojan circuit are inserted into the nodes with small transition probability in the circuit, and the excitation signal is inputted to the input terminals of the circuit to analyze the change of the circuit's transition probability and the activation probability of the hardware Trojan circuit before and after the MUX insertion. The experimental results of the ISCAS'89 reference circuit show that the number of nodes whose transition probability is less than the transition probability threshold in the circuit is significantly lower; the probability of the inserted hardware Trojan being activated is significantly improved; the increased percentage of circuit critical path delay is controlled within a preset scale factor.

Keywords: 2-to-1 MUX; hardware Trojan; transition probability; path delay

硬件木马也被称之为恶意电路,是在第三方 IP 或者制造过程中插入到电路中的微小电路模块. 一般情况下在电路系统中并不发挥作用,但在特定情况下会被激活. 一旦激活后可能改变电路功能、损坏电路甚至泄露电路信息,从而达到插入者的目的,危害可想而知[1]. 硬件木马模块相对于整个电路结构而言十分微小,激活前并不改变电路功能,使得硬件木马的检测变得十分困难.

除此之外,硬件木马可连接到电路网表的任何节点上,尤其是那些转移概率相对较低的节点. 传统自动测试矢量生成算法(ATPG)并不能有效的激活和检测硬件木马. 尤其当硬件木马的输入端连接到转移概率相对较小的节点后,逻辑测试将变得十分困难. 并且硬件木马电路对整个电路的功耗和延

收稿日期: 2016-11-29
基金项目: 国家自然科学基金(61100031)
作者简介: 徐 力(1992—),男,硕士研究生
通信作者: 周 彬,zbhit@hit.edu.cn

时的影响较小,通过时序和功耗检测的方法也收效甚微. 近年来,硬件木马的检测技术得到了明显的发展. 硬件木马的检测方法主要分为以下三种:基于失效性分析、逻辑测试和旁路信号分析[1].

基于失效性分析是最早用于硬件木马的检测方法,它主要依赖于高精度设备,诸如光学显微镜、电子显微镜等进行扫描分析[2]. 通过高精度设备扫描和重构原始电路,将反向设计和原始电路设计进行比对来判断电路中是否被插入硬件木马[2-4]. 这种测试方法对于小规模集成电路有一定的实用性. 但是随着大规模集成电路的发展,芯片的集成度越来越高,晶体管尺寸已经达到了纳米级,这种检测方法已经不能满足检测要求.

基于旁路信号的检测方法是目前一种有效的测量方法[5-12],通过测量和分析原始电路的信息,诸如延时[7-9]和功耗[5-6,10,12]等;得到原始电路的功耗或延时的特性曲线,也被称之为"IC 指纹"[1]. 再通过同样的方法得到待测芯片的特性曲线,然后与之前的特性曲线相比较,去判断待测电路中是否存在硬件木马[2]. 此种测量方法易受到外界环境和工艺差别的影响,当硬件木马的影响较小而环境和工艺噪声较大时,硬件木马很难被检测出来[1].

基于逻辑测试的检测方法通过向电路输入端输入激励信号,尽可能的激活电路中的硬件木马,通过比对电路的响应和正确的输出结果来判断电路中是否存在硬件木马[13-17]. 逻辑测试不受工艺噪声和环境的影响,能有效检测一些结构较小的木马[1]. 但是如果硬件木马的输入端连接到电路中转移概率很小的节点,这样以来硬件木马的活性大大降低,穷举测试就会变得十分耗时.

针对逻辑测试方法中硬件木马难以激活的问题,可通过插入二选一数据选择器(MUX)来提高电路整体节点的转移概率,从而提高硬件木马的激活概率的方法[18]. 本文在此基础上提出新的插入选择算法,同时通过设置最大延时比例来保证电路的最大延时在一定范围之内,避免因插入点的增加而引起关键路径延时过大的问题.

1 提出方法

当 MUX 的选择信号为'0'时,电路工作在正常模式下;当选择信号为'1'时,电路工作在测试模式,该模式下可直接在原电路内部节点输入测试信号,提高节点的转移概率,从而提高硬件木马的激活与检测概率. 随着插入节点的增多,电路的功耗、面积、延时等参数会有所增加. 通过优化算法,使得 MUX 的插入数量减小. 同时设置最大延时比例来控制电路关键路径的延时.

1.1 分析插入 MUX 对转移概率的提高

节点的转移概率是指节点的跳变概率. 转移概率越高的节点在测试中跳变的次数就越高,所以提高整个电路节点的转移概率能有效的提高硬件木马激活和被检测的概率. 基础逻辑门的转移概率计算方法见表 1.

表 1 逻辑门转移概率计算规则

Tab.1 Calculation rule of transition probability of logic gate

逻辑门类型	计算规则
与门/与非门	$t_{\mathrm{p}}^{\mathrm{out}} = \prod_{i=1}^{n} s_1^{\mathrm{in}^i} \cdot \left(1 - \prod_{i=1}^{n} s_1^{\mathrm{in}^i}\right)$
或门/或非门	$t_{\mathrm{p}}^{\mathrm{out}} = \prod_{i=1}^{n} s_0^{\mathrm{in}^i} \cdot \left(1 - \prod_{i=1}^{n} s_0^{\mathrm{in}^i}\right)$
非门	$t^{\mathrm{out}} = s_0^{\mathrm{in}} \cdot s_1^{\mathrm{in}}$
二选一数据选择器	$s_1^{\mathrm{out}} = s_1^{\mathrm{in}^1} \cdot s_0^{\mathrm{in}^3} + s_1^{\mathrm{in}^2} \cdot s_1^{\mathrm{in}^3}$ $s_0^{\mathrm{out}} = 1 - s_1^{\mathrm{out}}$ $t_{\mathrm{p}}^{\mathrm{out}} = s_0^{\mathrm{out}} \cdot s_1^{\mathrm{out}}$

表 1 中 $t_{\mathrm{p}}^{\mathrm{out}}$ 为该逻辑门输出端节点的转移概率;$s_x^{\mathrm{in}^i}$ 为逻辑门第 i 个输入端信号为 x 的概率,x 取'0'或'1';s_x^{out} 为逻辑门输出端节点信号为 x 的概率,x 取'0'或'1'.

通过 n 输入与门分析插入 MUX 对逻辑门转移概率的提高作用,其他类型的逻辑门可通过类似过程分析. 如图 1 所示,NET i 为候选节点,它的转移概率小于转移概率阈值 T_{th}. NET j 为逻辑门的第 j 个输入端. 由数学知识可知,当且仅当一个节点信号为'0'和为'1'的概率相等,即 $s_1 = s_0 = 0.5$ 时该节点的转移概率取最大值 0.25. NET i 的转移概率小于 T_{th} 的原因是由于 s_1 和 s_0 的差值过大引起的. 假设该与门的第 k 个输入端信号为'0'和'1'的概率为 s_0^k 和 s_1^k,则该逻辑门输出端节点在插入 MUX 前的信号为'1'的概率为

$$s_1^i = \prod_{k=1}^{n} s_1^k.$$

假定该逻辑门第 j 个输入端为'1'的信号概率最小,则在该输入端插入二选一数据选择器. 插入之后该逻辑门输出端节点信号为'1'的概率为

$$s_1^{i'} = s_1^{j'} \prod_{k \neq j}^{n} s_1^k.$$

对于一个逻辑门而言,转移概率小于 T_{th} 可分为以下两种情况:

1) $s_1^i \gg s_0^i$. 这表明与门的所有输入端信号为'1'的概率全部大于 0.5,即 $s_1^k > 0.5$,也表明 $s_1^j > s_1^{j'}$. 这样,$s_1^{i'}$ 的值比原来减小,$s_0^{i'}$ 的值比原来大,候选节点的转移概率得以增加.

2）$s_1^i \ll s_0^i$. 这表明存在某个输入节点，它得信号为‘1’的概率小于 0.5. 即 $s_1^j < 0.5$. 则 $s_1^j < s_1^{j'}$. 这样，$s_1^{i'}$ 的值比原来增大，$s_0^{i'}$ 的值比原减小，候选节点的转移概率得以增加.

分析可知，在输入端插入 MUX 可提高与门的转移概率. 其他类型的逻辑门也可通过类似的方法进行分析讨论.

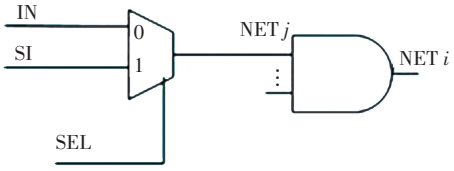


图 1　被插入 MUX 的逻辑门

Fig.1　A logic gate inserted MUX

1.2　插入点选择算法

具体插入算法见图 2. 流程所需要的输入为电路网表、人为设定的 T_{th}、最大延时系数 R 以及提供电路物理特性的库文件. 计算每个节点的逻辑深度 L_d 和扇出锥的节点数量 N_{cone}，通过给所有输入端赋逻辑值为“1”的概率为 0.5 的输入向量，可得到各个节点信号概率 s，通过计算得到每个节点的转移概率 t_p. 插入之前计算原始电路关键路径延时记为 C_{delay}. 所有 $t_p < T_{th}$ 并且可被插入的节点记为 N_1；N_1 中节点数记为 N，如果 $N = 0$，说明电路中没有 $t_p < T_{th}$ 或可被插入的节点，结束操作. 每个 N_1 的输入节点中拥有一个 s 最小的节点，选择方法见表 2，记为 n_0，所有的 N_1 中节点的 n_0 构成 N_2. 将 N_2 中的所有节点按照逻辑深度（到输入端）从小到大排序，相同逻辑深度按照扇出锥的节点数量排序，记为 N_3. 依次将 N_3 中的节点作为插入点进行插入操作，每次插入之后计算电路关键路径延时记为 T_{delay}. 判断 T_{delay} 是否满足要求？若满足，此次成功插入并更新电路重新计算电路的 s、L_d、N_{cone}、t_p，并再次进行插入点的选择直到结束；若不满足，删除最近一次插入的 MUX 后更新电路，继续执行插入操作. 若此次 N_3 中的所有节点均不满足时序要求，结束操作.

表 2　候选节点最小概率的输入节点选择方法

Tab. 2　Input node selection method for minimum signal probability of candidate node

逻辑门	选择方法
与门/与非门	为‘1’概率最小的输入节点
或门/或非门	为‘0’概率最小的输入节点

2　实验结果分析

本实验以 ISCAS’89 基准电路作为实验对象，使用 STM65 纳米标准单元库计算电路信息. 采用 C 语言搭建实验平台进行电路仿真. 在仿真测试过程中，主要输入端都赋予信号为‘1’的概率为 0.5 的信号，来计算电路的整体信息.

2.1　转移概率提高

在此实验当中，以 S9234 和 S5378 电路作为基准电路进行测试. 比较在插入前后电路所有节点的转移概率的变化. 通过图 3、4、5、6 可看出，在插入 MUX 之前电路中有大量转移概率小于转移概率阈值 0.05 的节点，其中还有不少 $t_p < 10^{-4}$ 的节点. 通过本文提出的方法，将转移概率阈值设置为 0.05. 在插入 MUX 之后，整体的转移概率提高，S5378 电路中 $t_p < 0.05$ 的节点数为 9 个，S9234 电路中这一数字为 84 个，并且都没有 $t_p < 0.01$ 的节点. 整体看来通过插入 MUX 之后，转移概率的提高十分明显.

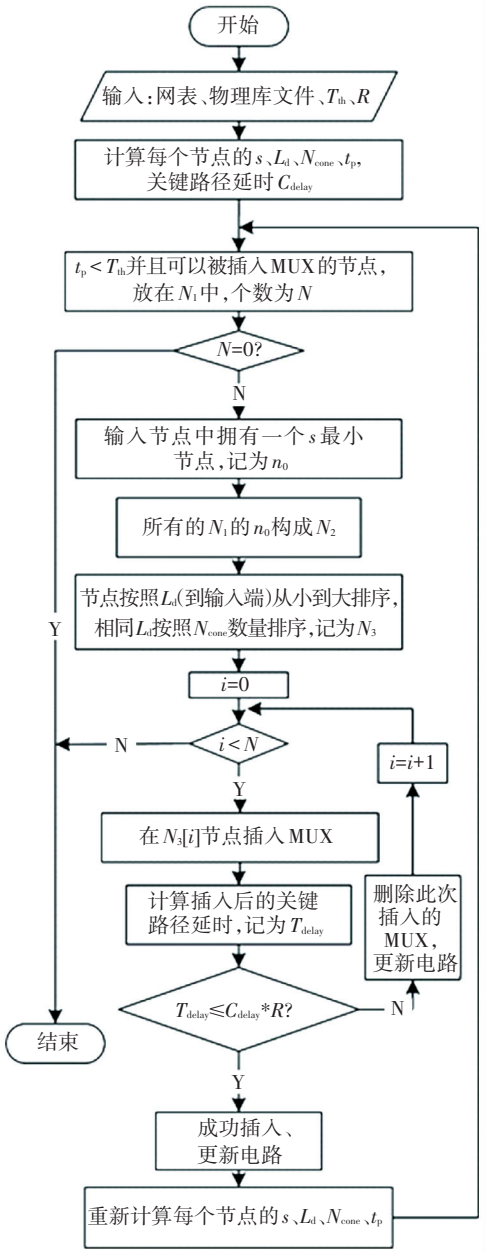


图 2　选择算法

Fig.2　Proposed selection algorithm

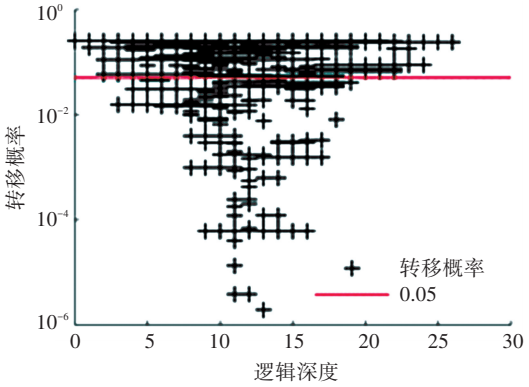


图 3 S5378 电路插入 MUX 前电路转移概率

Fig.3 Transition probability of S5378 circuit before inserting MUX

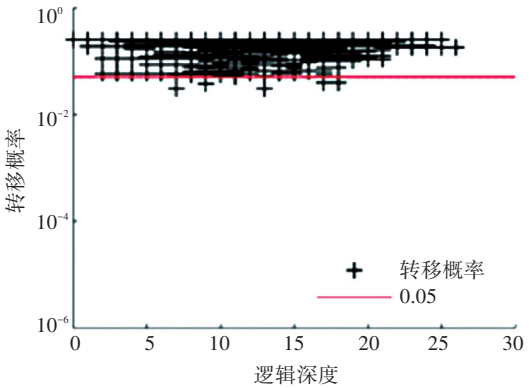


图 4 S5378 电路插入 MUX 后电路转移概率

Fig.4 Transition probability of S9234 circuit after inserting MUX

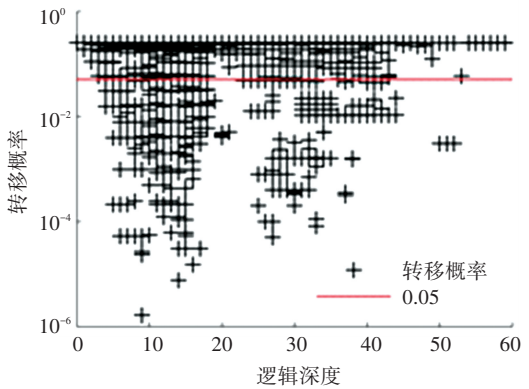


图 5 S9234 电路插入 MUX 前电路转移概率

Fig.5 Transition probability of S5378 circuit before inserting MUX

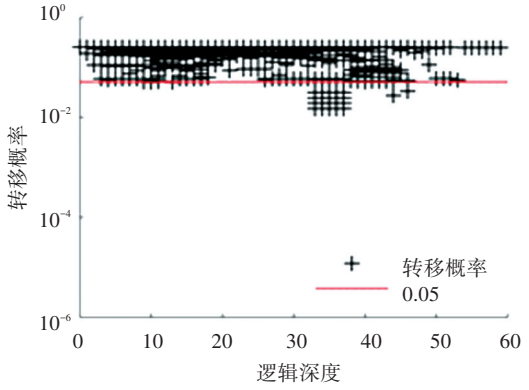


图 6 S9234 电路插入 MUX 后电路转移概率

Fig.6 Transition probability of S5378 circuit after inserting MUX

2.2 面积、功耗和延时的增加

通过插入 MUX 来提高整体电路的转移概率，随着插入数量的不断增加，势必会引起功耗、面积和延时增加. 特别是关键路延时的增加会给整个电路带来严重的影响. 通过设置最大延时比例，可将关键路径延时控制在可接受范围之内.

表 3 中为 S5378 电路的仿真结果，当 T_{th} 分别为 0.05、0.1，最大延时比例为 1.03、1.1 时，最大延时分别增加 1.66%、9.16%. 面积分别增加 7.83%、15.44%. 功耗分别增加 12.47%、19.63%.

表 4 中为 S9234 电路的仿真结果，当 T_{th} 分别为 0.05、0.1，最大延时比例为 1.03、1.1 时，最大延时分别增加 2.1%、13.3%. 面积分别增加 5%、9.29%. 功耗分别增加 17.94%、24.69%.

表 3 S5378 电路插入 MUX 后带来的影响

Tab.3 The impact of inserting MUX into S5378 circuit

转移概率阈值 T_{th}	最大延时比例	面积增加百分比/%	功耗增加百分比/%	延时增加百分比/%	插入 MUX 数量/个
0.05	1.03	7.83	12.47	1.66	104
0.10	1.1	15.44	19.63	9.16	205

表 4 S9234 插入 MUX 后带来的影响

Tab.4 The impact of inserting MUX into S9234 circuit

转移概率阈值 T_{th}	最大延时比例	面积增加百分比/%	功耗增加百分比/%	延时增加百分比/%	插入 MUX 数量/个
0.05	1.03	5.00	17.94	2.10	134
0.10	1.15	9.29	24.69	13.30	249

2.3 木马电路的激活

如图 7 所示，该硬件木马由触发器和负载两部分构成. 与门和非门构成触发器部分，异或门构成负载部分. TJ1、TJ2、TJ3、TJ4 和 TJ5 作为触发器的 5 个输入节点可被插入到目标电路的任意节点，当目标电路节点达到一定的逻辑值，木马电路的触发部分将被触发，电路的功能将被改变. 而当木马电路没有被触发时，电路的功能将不被改变.

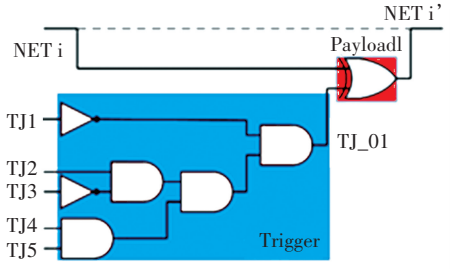


图 7 一种 5 输入的硬件木马实例

Fig.7 An example of hardware Trojan with 5 inputs

通过表 5 可知，如果硬件木马的输入端连接在 S5378 电路中转移概率相对较小的节点，那么硬件木马的激活将变得十分困难. 如图 7 所示的一个 5

输入组合逻辑的硬件木马，其输入端分别连接到 S5378 电路的 n219gat、n89gat、n110gat、n22gat 和 n200gat 节点上，在插入 MUX 之前，硬件木马的激活概率为 2.247×10^{-13}，要用逻辑测试的方法将其检测出来是一件十分困难的事. 转移概率阈值 T_{th} 设置为 0.05 后由图 4 可知，绝大多数的节点的转移概率都大于了 0.05. 通过表 5 可看出当 $T_{th} = 0.05$ 时，插入 MUX 之后木马的激活概率增加到 2.216×10^{-03}，在这样的激活概率下可有效的检测出硬件木马. 适当地将转移概率阈值 T_{th} 提高为 0.1，硬件木马的激活概率增加到 8.574×10^{-03}.

表 5　S5378 电路硬件木马激活信息

Tab.5　Information of hardware Trojan activation in S5378 circuit

木马输入端口	连接电路节点	插入前各节点 s_1	木马激活概率	$T_{th} = 0.05$ 时插入 MUX		$T_{th} = 0.1$ 时插入 MUX	
				s_1	木马激活概率	s_1	木马激活概率
TJ1	n219gat	0.997 71		0.906 25		0.625 00	
TJ2	n89gat	0.002 93		0.375 00		0.375 00	
TJ3	n110gat	0.996 95	2.247×10^{-13}	0.551 76	2.216×10^{-03}	0.566 40	8.574×10^{-03}
TJ4	n22gat	0.002 93		0.375 00		0.375 00	
TJ5	n200gat	0.002 93		0.375 00		0.375 00	

表 6　S9234 电路硬件木马激活信息

Tab.6　Information of hardware Trojan activation in S9234 circuit

木马输入端口	连接电路节点	插入前各节点 s_1	木马激活概率	$T_{th} = 0.05$ 时插入 MUX		$T_{th} = 0.1$ 时插入 MUX	
				s_1	木马激活概率	s_1	木马激活概率
TJ1	g6714	0.943 20		0.574 90		0.487 40	
TJ2	g4588	0.053 20		0.053 20		0.212 89	
TJ3	g6540	0.940 10	4.490×10^{-09}	0.500 00	7.069×10^{-04}	0.500 00	3.410×10^{-03}
TJ4	g6091	0.000 05		0.125 00		0.125 00	
TJ5	g1740	0.500 00		0.500 00		0.500 00	

在 S9234 中插入图 7 所示的硬件木马，在插入 MUX 之前，硬件木马的激活概率为 4.490×10^{-09}，当转移概率阈值分别设置为 0.05 和 0.1 之后，硬件木马的激活概率分别增加到了 7.069×10^{-04} 和 3.410×10^{-03}. 硬件木马激活概率的提高十分明显.

2.4　最大延时比例系数对结果的影响

在表 7 中以 S5378 电路为例，如果转移概率阈值设置较大，如 $T_{th} = 0.15$，电路中 $t_p < T_{th}$ 的节点数量将会增加，此时随着 MUX 的不断插入，电路的关键路径延时会不断增加. 如表 7 所示，当 $T_{th} = 0.15$ 而文献[18]中不对关键路径延时加以限制的话，关键路径延时将会增加 41.68%，这将会严重影响电路性能. 可根据电路性能要求设置一定的最大延时比例系数，当最大延时比例系数设置为 1.1 时延时，关键路径延时增加为 9.87%，由于插入 MUX 数量的减少，插入完成后 $t_p < T_{th}$ 的节点有所增加. 当最大延时比例为 1.2 和 1.3 时，关键路径延时增量分别为 19.00% 和 29.29%，插入完成之后 $t_p < T_{th}$ 的节点分别比不设置最大延时比例时减小了 38 个和 1 个.

表 7　最大延时比例系数对插入结果的影响

Tab.7　Influence of maximum delay proportional coefficient on insertion

最大延时比例	延时增加百分比/%	面积增加百分比/%	功耗增加百分比/%	插入 MUX 数量	$t_p < T_{th}$ 的节点数/个
1.1	9.87	27.03	25.43	359	319
1.2	19.00	28.50	28.39	379	292
1.3	29.29	29.74	29.74	395	255
无	41.68	30.05	28.47	399	254

3　结　论

在本文中，提出一种通过在转移概率较低节点的输入端插入二选一数据选择器的方法实现对电路节点转移概率的提高，从而实现加速硬件木马检测. 通过对扇出锥和电路逻辑拓扑结构分析选择对整个电路影响最大的候选节点，通过分析候选节点的逻辑门类型和输入信号概率选择最佳的插入点，从而

减少 MUX 的插入数量,实现对插入算法的优化. 同时引入最大延时比例系数用以控制电路关键路径延时,使电路关键路径延时控制在可接受的范围内. 通过实验分析,电路的转移概率得到整体提升,可有效防止硬件木马的插入. 被插入在电路中硬件木马的激活概率得到明显提高,可有效实现对硬件木马的检测. 同时,关键路径延时也得到有效控制.

参考文献

[1] 刘华锋, 罗宏伟, 王力纬. 硬件木马综述[J]. 微电子学, 2011, 41(5):709-713.
LIU Huafeng, LUO Hongwei, WANG Liwei. Survey on Hardware Trojan Horse[J]. Microelectronics, 2011, 41(5):709-713.

[2] AGRAWAL D, BAKTIR S, KARAKOYUNLU D, et al. Trojan detection using IC fingerprinting[C]// Proceedings of the IEEE Symposium on Security and Privacy. Berkeley: IEEE Computer Society, 2007:296-310. DOI: 10.1109/SP.2007.36.

[3] WANG Xiaoxiao, TEHRANIPOOR M, PLUSQUELLIC J. Detecting malicious inclusions in secure hardware: challenges and solutions [C]// Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust. Anaheim: IEEE, 2008:15-19. DOI: 10.1109/HST.2008.4559039.

[4] JIN Y, KUPP N, MAKRIS Y. Experiences in hardware Trojan design and implementation[C]// Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust. Francisco: IEEE Computer Society, 2009:50-57. DOI: 10.1109/HST.2009.5224971.

[5] SALMANI H, TEHRANIPOOR M. Layout-aware switching activity localization to enhance hardware Trojan detection [J]. IEEE Transactions on Information Forensics & Security, 2012, 7(1):76-87. DOI: 10.1109/TIFS.2011.2164908.

[6] RAD R, PLUSQUELLIC J, TEHRANIPOOR M. Sensitivity analysis to hardware Trojans using power supply transient signals[C]// Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust. Anaheim: IEEE, 2008:3-7. DOI: 10.1109/HST.2008.4559037.

[7] CHA B, GUPTA S K. Trojan detection via delay measurements: a new approach to select paths and vectors to maximize effectiveness and minimize cost[C]// Proceedings of the Design, Automation & Test in Europe Conference & Exhibition. Grenoble: IEEE, 2013: 1265-1270. DOI: 10.7873/DATE.2013.262.

[8] JIN Y, MAKRIS Y. Hardware Trojan detection using path delay fingerprint[C]// Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust. Anaheim: IEEE, 2008:51-57. DOI: 10.1109/HST.2008.4559049.

[9] XIAO Kan, ZHANG Xuehui, TEHRANIPOOR M. A clock sweeping technique for detecting hardware trojans impacting circuits delay[J]. IEEE Design & Test, 2013, 30(2):26-34. DOI: 10.1109/MDAT.2013.2249555.

[10] BANGA M, HSIAO M S. A novel sustained vector technique for the detection of hardware Trojans[C]// Proceedings of the International Conference on Vlsi Design. New Delhi: IEEE Computer Society, 2009:327-332. DOI: 10.1109/VLSI.Design.2009.22.

[11] 赵崇征, 邓高明,赵强. 基于旁路分析的集成电路芯片硬件木马检测[J]. 微电子学与计算机, 2011, 28(11):5-9.
ZHAO Chongzheng, DENG Gaoming, ZHAO Qiang. Detecting hardware Trojans in IC chips with side channel analysis[J]. Microelectronics & Computer, 2011, 28(11):5-9.

[12] 王力纬, 罗宏伟, 姚若河. 基于旁路分析的硬件木马检测方法[J]. 华南理工大学学报:自然科学版, 2012, 40(6):6-10.
WANG Liwei, LUO Hongwei, YAO Ruohe. Hardware Trojan detection method based on side channel analysis[J]. Journal of South China University of Technology (Natural Science Edition), 2012, 40(6):6-10.

[13] SALMANI H, TEHRANIPOOR M, PLUSQUELLIC J. A novel technique for improving hardware trojan detection and reducing trojan activation time[J]. IEEE Transactions on Very Large Scale Integration Systems, 2012, 20(1):112-125. DOI: 10.1109/TVLSI.2010.2093547.

[14] CHAKRABORTY R S, PAUL S, BHUNIA S. On-demand transparency for improving hardware Trojan detectability[C]// Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust. Anaheim: IEEE, 2008:48-50. DOI: 10.1109/HST.2008.4559048.

[15] WOLFF F, PAPACHRISTOU C, BHUNIA S, et al. Towards Trojan-free trusted ICs: problem analysis and detection scheme[C]// Proceedings of the Design, Automation & Test in Europe. Munich: IEEE, 2008:1362-1365. DOI: 10.1109/DATE.2008.4484928.

[16] CHAKRABORTY R S, WOLFF F, PAUL S, et al. MERO: a statistical approach for hardware Trojan detection[M]. Berlin: Springer-Verlag, 2009:51-57.

[17] XUE Mingfu, HU Aiqun, HUANG Yi, et al. Monte Carlo based test pattern generation for hardware trojan detection[C]// Proceedings of the IEEE International Conference on Dependable, Autonomic and Secure Computing. Chengdu: IEEE Computer Society, 2013: 131-136. DOI: 10.1109/DASC.2013.50.

[18] ZHOU Bin, ZHANG Wei, SRIKANTHAN T, et al. Cost-efficient acceleration of hardware Trojan detection through fan-out cone analysis and weighted random pattern technique[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015, 35(5):1-1. DOI: 10.1109/TCAD.2015.2460551.

(编辑 苗秀芝)