doi:10.3969/j.issn.1008-2956.2015.06.009

硬件木马防护研究综述

李雄伟，王晓晗，张阳，陈开颜，徐璐
（军械工程学院信息工程系，河北 石家庄 050003）

摘要：硬件木马具有体积小、易实现、难防护等特点，是目前集成电路面临的主要安全威胁之一，如何保护集成电路不受硬件木马侵袭已日益成为各应用领域迫切需要解决的问题.分析了集成电路面临的 3 种安全威胁，介绍了硬件木马的概念，讨论了几种比较常见的硬件木马分类方法，在此基础上，从集成电路测试、设计、运行 3 个方面着重阐述了硬件木马的防护方法，并分析了各自的特点，重点对未来硬件木马检测技术研究进行了展望.

关键词：集成电路；硬件木马；旁路分析；安全性设计
中图分类号：TN918 **文献标识码**：A **文章编号**：1008-2956（2015）06-0040-11

Survey on the Hardware Trojan Protection

LI Xiong-wei, WANG Xiao-han, ZHANG Yang, CHEN Kai-yan, XU Lu
(Information Engineering Department, Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract: Hardware Trojan' s characteristics are small in size, easy to implement, and difficult to protect against, and they are currently one of the major security threats to Integrated Circuits. How to protect Integrated Circuits from the invasion of Hardware Trojan has increasingly become a problem to be solved urgently in various applications. In this paper, we analyze three security threats of Integrated Circuits, introduce the concept of Hardware Trojan, and discuss some of the more common classification of Hardware Trojans. On this basis, we focus on the Hardware Trojan protection methods, and analyze their characteristics from three aspects of Integrated Circuits, namely Test-time, Design-time, and Run-time. Finally, we propose the prospect of the future research about Hardware Trojan Protection.

Key words: integrated circuits; hardware Trojan; side channel analysis; security design

随着微电子与计算机技术的快速发展，集成电路(integrated circuits，IC)已广泛应用于军事、金融等各个领域，大到航天飞机、武器系统，小到 IC 智能卡，IC 应用无处不在.IC 能保存和处理大量敏感信息，一旦被不法分子或敌方利用甚至破坏，将会严重威胁相关系统安全，损害国家和军事利益.IC 制造大致可以分为 3 个阶段：1)IC 电路设计；2)光刻和晶圆制造；3)封装和成品测试.许多环节都存在着安全隐患，图 1 为 IC 生命周期中每个环节的可信程度[1].目前还无法独自打造一条完全可信的 IC 生产供应链，很多生产商为了减少成本、缩短生产周期而选择外包，并引用第三方 EDA 工具和 IP 核，导致 IC 设计和制造分离，进一步增大了 IC 的安全隐患.

目前，IC 主要面临 3 方面安全威胁[2]：

1)赝品 IC，主要是指外观上与正规 IC 相同但不符合标准的 IC，包括翻新的老旧 IC、过量制造的 IC、有缺陷的 IC、不符合标准的 IC 和仿制 IC 等；

2)逆向工程，主要是指逆向 IC 制造过程，通过

收稿日期：2015-09-12；修回日期：2015-10-08
基金项目：国家自然科学基金项目（61271152；51377170）；河北省自然科学基金项目（F2012506008）
作者简介：李雄伟（1975—），男，博士，副教授. 主要研究方向：信息安全与对抗.

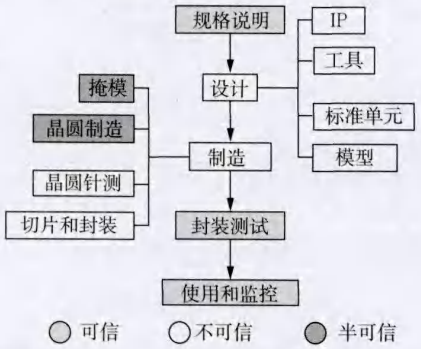


图1　IC生命周期中面临的安全威胁

物理电路来获取 IP 设计或者存储器中的敏感信息;

3)硬件木马(hardware Trojan),通常是指在原始电路中植入的具有恶意功能的冗余电路[3]10,它可以篡改数据、修改或者破坏电路功能、修改电路参数和拒绝服务等[4].

这 3 种威胁都具有低成本、易实现、难防护等特点,严重威胁着知识产权和信息的安全.其中,赝品 IC 虽然在性能、质量、使用周期等方面不及正规 IC,但仍可以"放心"使用;逆向工程主要涉及知识产权保护问题;而硬件木马更具威胁,且很难检测出长期潜伏在 IC 中的木马,主要体现在:

1)IC 内部结构复杂、集成度高,传统的功能测试和逻辑测试方法不能满足检测需求;

2)硬件木马规模相对很小,很难有效区分工艺噪声扰动和硬件木马影响;

3)硬件木马种类繁多、功能各异,没有通用的防护方法和手段.

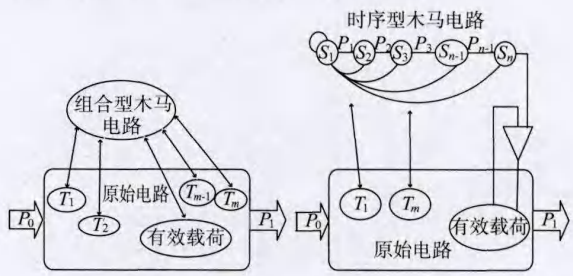
随着 IC 集成度的不断提高,结构越来越复杂,如何快速、准确、高效地检测出 IC 中潜藏的硬件木马,以保证 IC 应用安全显得尤为迫切.

1　硬件木马分类

硬件木马具有与软件木马相同的特性,如破坏性、寄生性、隐蔽性、可变异性、潜伏性等[5],它能够在稀有条件下引发故障,对 FPGA 比特流、专用集成电路(application specific integrated circuit,ASIC)、数字信号处理器(digital signal processing,DSP)、微处理器、微控制器或者网络处理器等进行修改[3]10,如可在军用级 FPGA 芯片中设置后门,通过该后门可轻易从芯片中提取配置信息,并访问/修改敏感信息[6].硬件木马比软件木马危害更大,在 IC 制成后,即使检测到硬件木马的存在也不能像软件木马一样将其查杀.硬件木马种类多样,根据植入方式、工作环境、实现功能的不同以及抗检测性的需求,能够繁衍出很多变种,给硬件木马防护带来困难.因此,对硬件木马进行分类,将有助于认识和防护硬件木马.

硬件木马的结构一般分为触发和负载 2 部分[7],通常作为木马分类的一种依据,即以主电路的某些内部电路节点作为输入(触发),连接到负载部分,并再次连接到其他的内部电路节点.当满足触发条件时负载开始工作,对电路节点信号进行修改或者改变电路特征参数.

根据硬件木马的触发特性一般将木马分成组合型木马和时序型木马[8-9],如图 2 所示.组合型木马是指电路中的某些内部信号同时运行到某一特定值时才被激活的木马,该木马不包含任何状态信息(如寄存器和锁存),一般逻辑规模较小,通过选取一些稀有的内部信号来提高木马的抗检测能力,但仍存在被触发的可能.而时序型木马一般是经过一个连续的状态转换,如有限状态机(finite state machine,FSM)才被激活的木马.由于有限状态机中存在一些冗余状态、隐藏状态和孤立状态,该木马的隐蔽性非常强,不易被触发,但是该木马一般会用寄存器来存储逻辑值,使整个电路的功耗增加.



a) 组合型木马　　b) 时序型木马

图2　根据硬件木马触发特性划分的木马示意图

Wang 等[3]11,[10]2根据硬件木马的基本特征提出一个比较详尽的木马分类方法,按照木马的物理特征、激活特征和行为特征,分别从硬件木马的不同物理表现形式、激活方式或条件以及木马所导致的恶意功能这 3 个角度出发,将木马分为 3 类,如图 3 所示.

Zhang 等[11]根据木马对电路正常功能的影响,将木马分为故障型木马和寄生型木马.前者主要是改变电路并影响其一些正常功能,而后者一直隐藏在原始电路中,并不对任何正常功能产生影响,直至被激活.目前,大多数木马都是寄生型木马,由于该木马的恶意行为并没包含在规格说明中,更难在功能测试时被激活或发现.

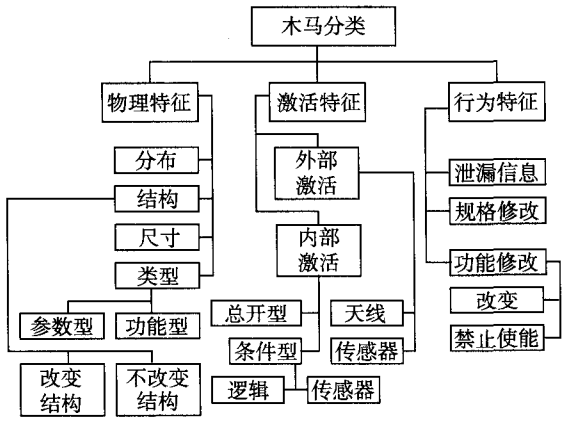


图3 根据硬件木马基本特征划分的木马分类示意图

Bhunia 等[12]在上述分类方法的基础上进行了整合和扩展，从木马的基本结构入手，在触发和负载的基础上将硬件木马进一步分为数字型木马（与电路逻辑值相关）和模拟型木马（与电路中的特征参数有关），如图 4 所示.该方法基于木马的触发机制和功能，深层次地描述了木马的工作特性，更有助于对木马防护方法的研究.

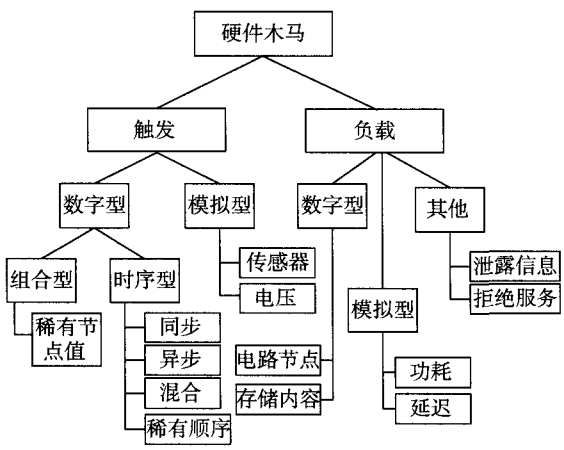


图4 Bhunia的硬件木马分类方法

硬件木马通常在稀有条件下触发，并且能够在 IC 生命周期的任何阶段植入，木马防护可以从 2 个方面着手：

1）对已植入的木马进行检测；

2）防止木马植入.

本文借鉴文献[13]的思路，围绕 3 个方面展开分析：

1）IC 测试过程中的木马检测方法；

2）IC 设计过程中的安全性设计方法；

3）IC 运行过程中的木马行为检测方法.如图 5 所示.

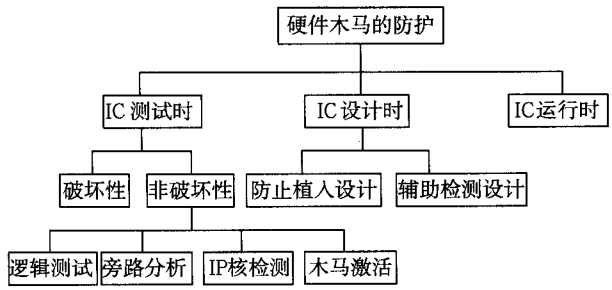


图5 硬件木马防护方法

2 IC 芯片测试过程中的木马检测方法

由于传统功能测试和逻辑测试主要针对 IC 参数中的缺陷检测或者不可接受误差测试，这些方法不能有效识别由木马激活而引起的额外功能，近年来，在 IC 测试过程中，相关研究人员根据不同的木马种类，提出了一些硬件木马的检测方法.

2.1 破坏性检测

破坏性检测方法是基于失效分析技术的检测手段.对集成电路进行封装后，无法观察内部的组件信息，通过逆向工程方法将封装的集成电路打开，逐层扫描电路，然后重建电路结构图，对比需求规格，找出电路中是否存在硬件木马，以保证集成电路安全[10]1.该检测方法能进行彻底的检测，发现电路中的任何恶意修改，但是耗时长，根据集成电路复杂程度的不同，需花费几周到几个月的时间.同时，经过破坏性检测后的 IC 不能再使用，不适合对所有 IC 进行逐个检测，只能通过抽样的方式，检测结论的可靠性大大降低.但通过该方法可获取“金片”（不含木马的 IC），作为其他检测方法的参考模板.

2.2 逻辑测试

由于木马具有潜伏性，在非常稀有的条件下才能被激活，针对这一特性，Wolff 等[7]3 提出使用逻辑测试方法来进行检测.该方法源于基于 VLSI 故障测试的 ATPG（automatic test pattern generation）测试技术，主要通过生成测试激励，并在输出端口观察硬件木马对电路值造成的影响，从而达到检测目的.由于硬件木马空间能够根据电路节点数而无限放大，无法通过枚举所有可能的木马激活条件来生成关键性的测试向量或者计算测试覆盖.因此，在最大木马激活概率下优化测试向量生成方法是逻辑测试的首要目标.

Chakraborty 等[14]3 提出了一种基于统计的方法 MERO（multiple excitation of rare occurrence）来生成测试向量.MERO 类似于 ATPG 测试技术中

的 N-detection 测试[15],该方法首先检测出电路节点中的稀有事件,根据每个稀有事件的激活条件生成一个最佳测试向量集.对于每个稀有事件,该向量集都能将其激活 N 次(N 的值由检测者定义).由此得到的测试向量集具有很好的测试质量和木马检测覆盖,即使是随机模式生成的向量也能通过此方法增强木马激活的概率.实验证明,同等木马检测覆盖下,该方法生成的测试向量比随机测试向量减少了 85%[14]4.Waksman 等[16]针对隐蔽性强的木马提出使用工具 FANCI 进行检测,该工具主要是对电路逻辑值进行布尔函数分析,从而检测出潜伏的硬件木马.

2.3　旁路分析

由于 IC 芯片在工作时会产生一些热信号、电磁辐射信号以及功耗信号等旁路信号[17],植入的硬件木马势必会对其造成影响,例如降低性能和改变功耗等特征.虽然硬件木马多数时间处于未激活状态,但是硬件木马时刻检测触发条件的行为也会对旁路信号产生一些影响,使之成为检测的突破口.旁路分析方法就是基于这一现象,比对待测芯片与"金片"之间旁路信号的差异,如果不同则表明待测芯片中可能含有硬件木马.

2.3.1　基于功耗的旁路分析

基于功耗方法是比较待测芯片和"金片"的电路功耗,一般是在同等条件下测量芯片电路 V_{DD} 引脚上的电流.其中,每条电流包含若干元素,主要包括:1)主电路电流;2)测量噪声,可以通过多次测量求平均来消除;3)工艺噪声,随机产生不能被抵消;4)可能存在的木马信号,通过比对电流差异(木马信号)来判断是否含有木马.

Agrawal 等[18]3较早使用功耗旁路信息来检测电路中的硬件木马,通过应用随机测试向量来测量功耗,比较待测芯片与"金片"之间的功耗差异,并检测不同工艺噪声下不同尺寸的硬件木马.Aarestad,Alkabani 和 Potkonjak 等[19-22]通过比较电路中的静态电流差异来检测木马,如果硬件木马改变了原始电路中门电路的个数,即使木马未激活,理论上都能判断待测电路中是否含有木马.但是由于产生的静态电流很小,相对于整个电路,少数几个门电路引起的静态电流差异不足以进行充分的判断.针对此问题,Aarestad 等[19]提出测量多个引脚上的电流来进行检测.Alkabani 和 Koushanfar[20]利用门级特征的统计收敛特性和信号完整性对木马进行检测.Potkonjak 等[21]使用线性规划和奇异值分解处理电路的门级特征公式来检测木马,并使用电路校准技术削弱噪声的影响.Potkonjak 和 Wei 等[22]提出基于分割的方法检测木马,通过测试向量将电路分为若干子电路,以子电路中门电路的泄露电流为模板进行检测,并对高泄露电流的子电路进行重点检查.

Narasimhan[23]提出通过瞬态电流分析来检测木马电路中的翻转活动,该方法需要减少噪声的影响,因为在测量过程中的电压、温度和测量噪声信号的瞬间抖动都可能造成误判,目前大多数研究都主要采用归一化和求均值来减弱噪声的影响.Narasimhan[24]提出通过使用自相关矩阵比较不同区域中的瞬态电流来削弱工艺噪声.此外,有人提出可以通过增加木马电路的翻转活动来提高检测效果,例如可以通过基于区域分割[25]和直接生成测试向量[26]来提高检测效果.Alkabani 等[18]6使用信号处理方法,如 Karhunen-Loève 变换,将功耗轨迹映射到噪声的特征空间凸显木马,并在±7.5%的工艺噪声下检测到占整个电路 0.01% 的木马.Wang 等[27]对一段时间内的电流求积分,以此来检测电流中的累计差异,并很容易检测到 0.01%大小的木马.

2.3.2　基于延迟的旁路分析

如果硬件木马的植入导致电路中门电路个数的变化,就会改变电路的延迟特征[28-30],如信号翻转延迟以及信号传输延迟.基于延迟的旁路分析就是通过比对测量电路的延迟信息差异来检测木马.

Jin 和 Rai 等[28-29]提出基于路径延迟来进行木马检测.该方法测量几个芯片上几条指定路径的延迟并将工艺噪声考虑在内,在±7.5%工艺噪声范围内检测出占总电路大小 0.36%的木马.Li 等[30]提出另外一种基于延迟的方法,使用基于影子寄存器的方法来测量路径延迟,将 2 个寄存器之间的路径延迟定义为影子寄存器,影子寄存器具有与真实寄存器相同的时钟频率但有一个偏移,通过比对设计与测试时影子寄存器的频率,如果存在差异则表明存在木马.

2.3.3　基于多参数的旁路分析

理论上通过功耗和延迟信息可以检测出电路中存在的木马,但是由于工艺噪声的存在,检测效果往往不是很理想,尤其是基于延迟信息的旁路检测,因此 Narasimhan 等[31]提出可通过测量多个旁路参数来提高检测效果,将电路最大工作频率与静态电流和动态电流 2 个参数进行运算,减小工艺噪声的影响,提高木马检测效果.

旁路分析和逻辑测试 2 种方法对于硬件木马检

测各有特点，表1对两者的优缺点进行了比较，两者相辅相成、优势互补，从理论角度看，2种方法的结合可以检测大多数木马.

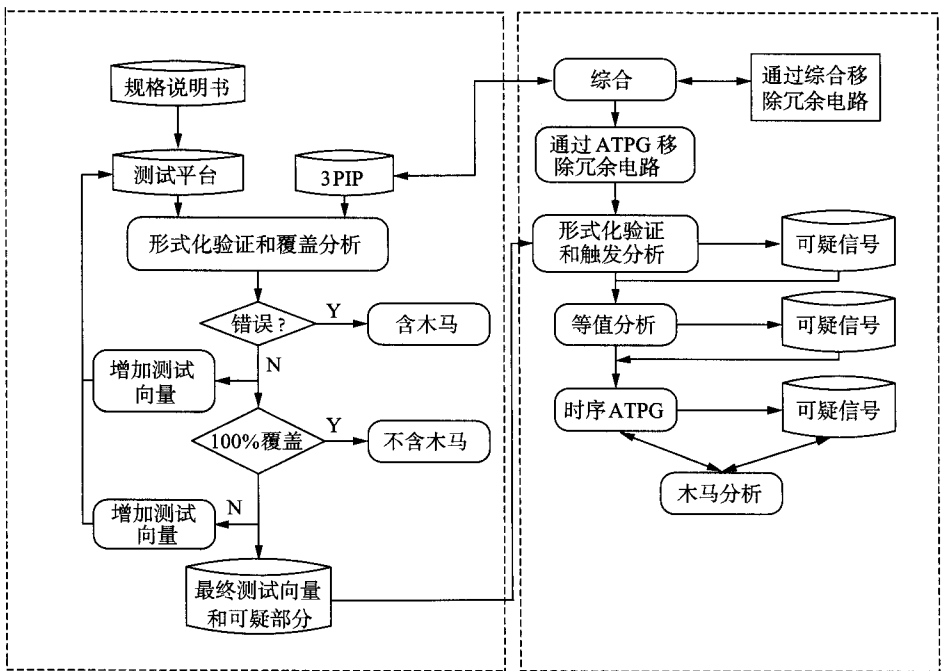
表1 逻辑测试方法和旁路分析方法的比较

	逻辑测试方法	旁路分析方法
优点	对小型木马有效，不受工艺噪声影响	对大型木马有效，测试向量简单
缺点	测试向量复杂，不适合检测大型木马	受工艺噪声影响，不适合检测小型木马

2.4 IP核中的木马检测

随着IC集成度不断提高，设计的复杂度也不断提高，使用IP核可以避免很多重复性工作，提高工作效率，缩短集成电路的生产周期，但是由于其来源广泛，安全性很难保证，如果IP核中含有木马，那么由该设计生产出来的所有电路中都含有木马，所以保证IP核的安全可信至关重要.但是IP核中的木马检测不同于IC中的木马检测，因为没有一个已知不含木马的“黄金”参考模型，即使是厂商提供的规格说明书和源代码也很有可能含有木马，而且IP核一般用VHDL或者Verilog语言描述，工业级的IP核中能够包含成千上万条代码，而代表木马的代码可能仅仅只有几行，如何识别出代码中的木马部分是检测的重点，亦是难点.

现阶段检测IP核中木马的主要思路是使IP核充分运行，找出可疑信号(不经常翻转)，再判定是否含有木马[32]6，具体流程如图6所示.第1阶段是生成测试平台和识别可疑信号.生成测试平台类似于前面提到的逻辑测试，主要根据规格说明书中的性能和基本功能来生成测试向量，使用该测试向量来检验代码是否100%运行，但是检测时间会随着测试向量数的增加而增加，很难达到100%代码覆盖，需要优化测试平台以达到最大代码覆盖率和最少的仿真时间.检测完成时，如果代码覆盖率为100%，则IP核是可信的，否则未覆盖代码部分中可能含有木马.第2阶段是可疑信号分析.冗余电路同木马电路一样，不包含在规格说明中，测试向量不能使其工作，因此，在该阶段需要先将可疑信号中的冗余电路移除，然后采用等值分析方法进一步减少可疑信号的数量，最后再用时序ATPG激活可疑信号并判断其是否为木马信号.



a) 生成测试平台与识别可疑信号 b) 可疑信号分析

图6 IP核中木马的检测流程

Zhang等[32]5对19个植入木马的RS232基准进行仿真测试，移除冗余电路之后72%的可疑信号与木马相关，完成等值分析时百分比增加到85.2%，应用时序ATPG后93.6%的可疑信号是木马信号.但是这样的检测效果仍不尽人意，需进一步提高木马检测率，尤其需对大型IP核进行检测.

2.5 木马激活

硬件木马在其大多数生命周期中都处于“休眠”

状态,影响了基于旁路信号检测的效果,如果能结合逻辑测试方法来使木马电路完全激活或者部分激活,使木马电路表现出更多的恶意行为或者泄露更多的旁路信号,将有助于木马的检测.当前木马激活策略主要分为全区激活和区域感知 2 类.

2.5.1 全区激活

该类方法不考虑木马的位置信息,而是依靠木马的偶然或者有意激活.Jha 等[33] 使用基于随机概率的方法来检测木马,该方法根据电路的测试向量构建一个特殊的概率签名,并以相同的概率将测试向量应用到待测电路中,将测试结果与原始电路的输出结果相比较,如果有差异就表明存在木马,并评估其置信水平.Wolff 等[10]6 分析设计中的稀有节点组合,针对以稀有事件作为触发的木马,生成一个向量集来激活这些稀有节点,并结合该向量集和传统的 ATPG 测试向量来激活木马,如在逻辑测试中用到的 MERO 方法.

2.5.2 区域感知

该类方法的主要目标是局部放大待测 IC 和原始设计功耗波形之间的差异.Banga 和 Hsiao[8]41 提出了基于区域的测试向量生成方法,该方法根据 IC 结构的连通性,将待测电路中的触发器分到不同的区域,并找出可能存在木马的区域作为被测试区域,对被测试区域生成新的测试向量来放大原始电路和可能含木马电路之间的功耗差异.此外,Banga 和 Hsiao[9]3 通过持续测试向量技术来放大木马的活动,其思路源于电路活动主要来自设计的状态元件(如寄存器),若保持输入引脚的状态几个时钟周期不变,可以减少整体的翻转活动,并限制设计中特定部分的翻转活动,有助于定位木马.

这 2 类木马激活策略都不考虑木马的类型和尺寸,只针对木马的分布特性.如果木马电路的输入来自电路的一部分,区域感知方法比较有效;如果木马的输入来自电路的各个部分,则全区激活方法能增加检测的概率.

2.6 检测方法总结

根据硬件木马的各式特点,涌现了多种木马检测技术,由于目前没有一个统一的硬件木马模型和检测效果衡量标准,各种检测方法的优劣尚不足以盖棺定论,仍需要对各种检测技术进一步展开研究,从而寻找更为有效的通用木马检测方法.

随着硬件木马规模的不断减小,IC 芯片内部结构更加复杂,当下最具发展前景的木马检测方法是基于旁路分析的硬件木马检测.由于该类方法检测效率高,且对 IC 芯片没有损害,具有广阔的研究价值.但是该方法尚处于起步阶段,仍需进一步进行探索,需要在以下几个方面加强基于旁路分析的硬件木马检测技术研究工作:

1)旁路信号的采集与处理.IC 芯片工作时产生的旁路信号是一种模拟高频信号,这种信号十分微弱,而且信号在从元器件到示波器的传递过程中会受到噪声等多种因素的影响,很难准确地对其进行测量.不仅如此,对旁路信号进行去噪处理也会削弱原始信号,对后期的分析检测造成一定影响.因此,如何获取高质量的有效旁路信号是木马检测的重要前提.

2)特征选择与提取.由于旁路信号具有成分复杂、高维等特点,直接对原始信号进行处理很容易造成"维数"灾难,检测效果往往不是很理想.因此,有必要对原始信号进行特征选择和提取,探究旁路信号中最能体现硬件木马特点的辨别特征,去除不具辨别能力的特征,并使用变换方法对选定的特征进行信息压缩处理,减少或去除信息冗余,简化计算复杂性.

3)模板最优匹配.以"金片"旁路信号作为参考模板,对待测芯片的旁路信号进行识别,本质上属于分类决策问题.由于工艺噪声等因素影响,对于不同芯片或者同一芯片的不同批次,采集到的旁路信号不尽相同,这给硬件木马的潜伏提供了便利条件,同时也给模板的匹配带来困难,不利于进一步的分类决策.因此,有必要对旁路信号进行物理建模,描述其统计分布特性,找出由木马造成的旁路信号差异,建立最优匹配测度,达到与参考模板的最佳匹配.

4)寻求无"金片"的检测方法.目前大多数木马检测方法都需要"金片"作为参考模板,该检测形式虽然有效但仍存在一定的局限性.一方面,"金片"的获取需要采用破坏性检测方法,十分耗时;另一方面,对于某些类型的木马,如通过改变电路板上的电路厚度来植入的木马,就很难得到真正的"金片".因此,需要拓展硬件木马检测的思路.

3 IC 设计过程中的安全性设计方法

由于木马具有潜伏性和抗检测性等特点,直接用旁路分析和逻辑测试对木马进行检测效果不是很理想,或者成本远远超出检测者的可承受范围.如图 7 所示的木马电路,很难采用基于电路延迟的旁路分析技术进行检测,因为木马带来的延迟影响很容易降至最低.因此,有必要在 IC 设计过程中采取一

些有针对性的安全性设计方法，尽可能降低木马对IC的威胁.目前，安全性设计方法主要分为2类：1)防止木马的植入；2)辅助木马检测.

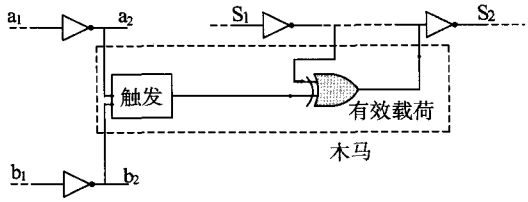


图7 木马电路（虚框部分）

防止木马的植入需要考虑多方面的因素，因为敌方可以在IC生命周期的任何阶段植入木马，如果制造过程分离，则更难防止木马的植入.解决这一现象的主要思路是保密，Imeson等[34]提出不向外透露设计的内部连接信息来避免木马攻击.Xiao等[35]提出了另外一种防止木马植入的手段——内建自验证技术(built-in-self-authentication，BISA)，该技术主要是用填充单元将IC中的剩余空间填满，可以防止增加木马电路的攻击，但是不能防止对电路的恶意修改或者重新设计原电路来植入木马.阻止木马植入的设计虽然有效但不能从根本上防止木马的植入，目前安全性设计多以辅助木马检测为主.

辅助木马检测主要是通过修改设计，使设计中的木马更容易被检测.最常见的一种安全性设计是内建自检测技术(built-in-self-test，BIST)，这种设计方法是设计者在设计电路的时候额外设计一个测试模块，并以此来分析集成电路的安全性[36].该测试模块需要检测者输入一个特定的密钥来启动，在检测者输入测试向量之后，它会检测电路中的稀有节点的逻辑值并生成一个自检结果，通过比较待测IC与“金片”生成的自检结果来检测芯片中是否含有木马.该设计方法虽然能够辅助木马检测，降低检测成本和时间，但是增加了IC内部电路冗余，使内部电路结构更加复杂，在一定程度上影响IC的性能，不太适合需要高运算性能的芯片.当前，随着相关研究的深入，针对木马检测的一些安全设计策略也不断涌现.

3.1 植入环形振荡器

该方法类似于内建自检测技术，在IC中植入环形振荡器来辅助木马检测[37]，这种特殊的硬件结构是在一个环路中加入一些零散的反相器(通过多路转换器与原始电路相连)，通过检测环形振荡器中每个逻辑单元的延迟、多路转换器的延迟以及路径延迟(称为环形振荡器的频率)，判断电路中是否含有木马.在电路中加入环形振荡器后，任何对环路的修改都能被检测，因为一旦环路中的其他门电路加载反相器，振荡器的频率将会改变.如图8所示，在C17电路中植入一个环形振荡器，当g8处置高电平时，在右侧的g9处将会得到环形振荡器的频率；如果在这个环路中植入额外的门电路或者改变现有门电路的功能，都会引起振荡器频率的改变.与识别电路中逻辑值的变化不同，该方法主要检测电路中延迟信息的变化，对木马更敏感，适用范围更广泛.但是该设计受温度和工艺的影响较大，容易引起环形振荡器频率的改变，检测条件比较苛刻.

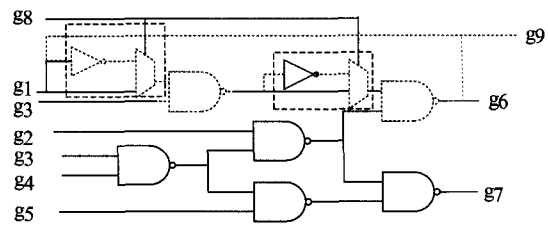


图8 环形振荡器(虚线部分)

3.2 移除稀有事件

木马一般都在稀有条件下激活，如用稀有的电路状态、某一温度或噪声来激活木马，并且木马的位置、类型、大小都是未知的，无法生成决定性的测试向量来检测木马.通过改变电路设计，减少电路中的稀有事件，使随机测试向量都能高效地激活木马.如对于一个有 $q(q>1)$ 个触发输入的木马，$P_i(i=1,2,\cdots,q)$ 为第 i 个输入被触发的概率，则生成一个特定触发向量的概率为 $P_{\text{trigger}}=\prod P_i(i=1,2,\cdots,q)$.为了使木马的触发概率最小，就应当使 P_{trigger} 最小，因此，通常选取 q 个输入的稀有组合来激活木马，或者令每个输入有非常低的0－1翻转概率($P_i(0)\gg P_i(1)$或者$P_i(1)\gg P_i(0)$).由于木马的规模限制，q 的选取不宜过大，木马设计者往往选取第2种方式，即将电路中的稀有事件作为木马的触发输入.在这种情况下，Salmani等[38]提出在电路中植入哑扫描触发器(dummy scan flip-flop，dSFF)来增加电路节点的0－1翻转概率(0－1翻转概率低于一个阈值(Pth)的电路节点)，从而移除电路中的稀有事件，如图9所示，用dSFF-AND增加 $P_i(1)\gg P_i(0)$ 的电路节点的0－1翻转概率，用dSFF-OR来增加 $P_i(0)\gg P_i(1)$ 的电路节点的0－1翻转概率.该设计方法在一定程度上可以提高木马被激活的概率，但不能完全消除电路中所有的稀有事件(如电路中需要稀有事件来控制或者实现某些功能)，而且对总开型木马几乎无效.

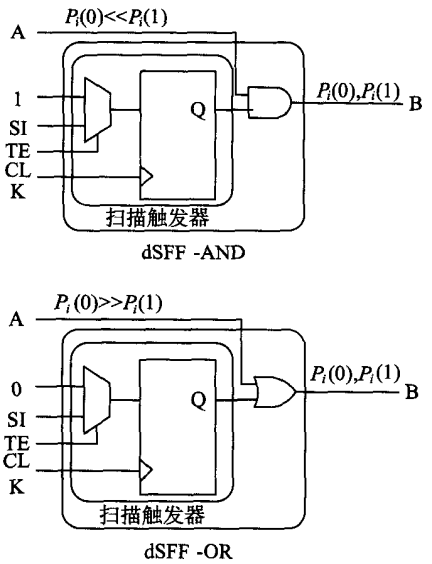


图9　哑扫描触发器

3.3　重排扫描单元

由于电路的总功耗与电路中的翻转总数呈正相关,所以压低正常电路的翻转活动,保持木马电路的正常翻转将会增加木马的相对电路活动(Trojan-to-circuit activity,TCA),这将大大增强检测出对电路功耗几乎没有影响的小型木马的概率,根据这一原理对原始设计进行适当的修改可以提高检测效果.如在电路中植入若干扫描单元,则电路的总功耗与扫描单元的翻转总数呈正比,按照不同的标准将扫描单元划分成若干条扫描链(一般贯穿整个电路),根据扫描单元在电路布局中的物理位置定位扫描单元并重构扫描链,保持一个区域的翻转活动同时限制其他区域的翻转活动,从而提高 TCA 和检测率[39].这种布局感知扫描单元重新排列对于基于功耗的旁路检测技术的检测效果提升较大,不论木马大还是小、分散还是集中都能取得比较好的检测效果,因为正常电路翻转的变化要远远多于木马电路.

3.4　安全性设计方法总结

由于敌方可在 IC 制造过程中的任一环节植入硬件木马,无法有效防止木马的植入,目前的安全性设计方法多以辅助木马检测为主,使木马检测更容易进行.植入环形振荡器和重排扫描单元均是为旁路分析提供后门,比较适合检测小规模电路中的木马,但容易受工艺噪声的影响;移除稀有事件主要是针对逻辑测试方法,使测试向量生成更容易,但是需要设计人员详尽了解 IC 芯片的内部结构,不适合引入 IP 核较多的 IC 芯片.

这类方法虽然为硬件木马的防护研究提供了一条行之有效的思路,但均是在 IC 芯片中植入额外的电路来实现其功能需求,其规模随 IC 芯片规模的增加而增加,在一定程度上增加了电路冗余,其电路规模甚至远远超出硬件木马,不适合进行高精度运算的 IC 芯片,仍需探寻更为有效的安全性设计策略,使硬件木马无处可藏.

4　IC 运行过程中的木马行为检测方法

测试时该检测方法虽然能有效地检测出木马,但是不能完全覆盖所有类型和尺寸的木马,尤其是 IP 核中的木马和一段时间后才被激活的木马,即使是采用安全性设计,仍不能完全保证交付使用的集成电路芯片中没有木马.因此,在使用 IC 过程中有必要采用运行行为检测对电路进行可信验证,进一步减少可能存在的木马攻击.

Bhunia 等[40]提出在电路中植入可重配置的安全监视器(security monitor,SM)来检查由木马引起的非法行为(如非法访问地址空间或者非法进入调试模式等非法操作),其结构如图 10 所示,每个

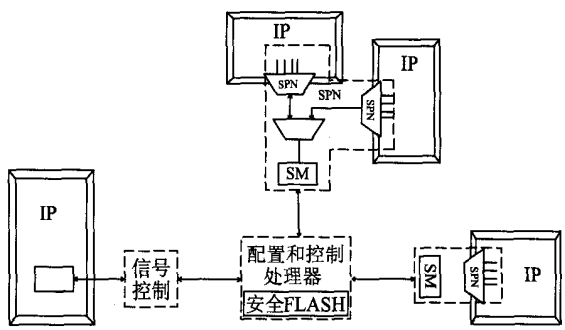


图10　运行行为检测

SM 都由有限状态机(FSM)控制,利用信号探测网(signal probe network,SPN)来抓取感兴趣的信号.配置和控制处理器(configuration and control processor,CCPRO)主要是对 SPN 和 SM 进行配置来选择被监视的信号群和分析选择的信号,在对 SM 配置过程中不会干扰芯片的正常操作,也不会影响其他 SM,配置信息储存在 CCPRO 中的安全存储器中(如 FLASH).该方法可以适用于不同设计规格的电路,并且在检测过程中不需要黄金参考模型,只对少量的系统行为进行检查就能达到非常高的错误覆盖[41].但是由于每种安全检查只能对应一种非法行为,需要重复配置来动态地执行不同的安全检查,仍然需要考虑如何利用有限的硬件资源来高效地进行大量的安全检查.

McIntyre 等[42]提出使用硬件多核系统来检查木马,在不同的处理部件(processing element,PE)

上执行相同功能不同形式的程序(不同算法或者不同编译语言),通过比较它们的运行结果来确定每个PE中是否含有木马,即使是2个PE中含有同一种木马,也不会同时运行.但是本质上多核系统是冗余的,需要采用分布式软件来调度,这也增加了额外的运算,而且其检测效率也依靠程序的选取(不同形式).Bloom, Narahari 和 Simha[43] 针对拒绝服务(denial-of-service,DOS)和权限提升攻击类型的木马,提出使用硬件监视电路和操作系统来进行定时检查.该方法类似于软件查杀,在检查到木马后将其隔离,对于IP核或者处理器控制代码中的木马也同样有效.

5 结束语

硬件木马防护技术作为信息安全领域中的一个新兴方向,其各项防护或检测技术仍不十分成熟,存在各种各样的局限性,由其得到的结论并不能完全使人信服,即便是主流的旁路分析方法和逻辑测试方法目前也还停留在仿真和实验阶段,并没有真正应用于实际工程中.同时应该看到,在现实中攻击总是要比防御容易得多,敌方还可以针对现有的防护或检测方法设计有一定抗防护/检测能力的木马,不仅如此,硬件木马的攻击也正在随着IC的复杂性和IC生命周期中不受信任的第三方工具、IP核或设备的侵扰而不断升级,并不时涌现出新的攻击模式,如何预防硬件木马还有很长的路要走.未来除了需要加强相关防护技术的研究外,还需要对芯片安全评测标准和相关法律制定等方面加以重视.

参考文献:

[1] DARPA. Trust in integrated-circuits proposer information pamphlet[EB/OL]. Cleveland: IEEE, 2007[2015-07-25]. http://www. Darpa. Mil/MTO/solicitations/baa07-24/index.html.

[2] GOERTZEL K, HAMILTON B. Integrated circuit security threats and hardware assurance countermeasures [C]// Proceedings of Cyber Security and Information. New York: ACM, 2013: 33-38.

[3] TEHRANIPOOR M, KOUSHANFAR F. A survey of hardware Trojan taxonomy and detection [J]. IEEE Design & Test of Computers, 2010, 27(1): 10-25.

[4] KARRI R, RAJENDRAN J, ROSENFELD K. Trojan taxonomy [C] //Proceedings of Introduction to Hardware Security and Trust, 2012. New York: Springer, 2012: 325-338.

[5] 牛小鹏,李清宝,王炜,等.硬件木马技术研究综述[J].信息工程大学学报,2012,13(6):740-748.

[6] SKOROBOGATOV S, WOODS C. Breakthrough Silicon scanning discovers backdoor in military chip [C] // Proceedings of Cryptographic Hardware and Embedded Systems (CHES12). Berlin: Springer, 2012: 23-40.

[7] WOLFF F, PAPACHRISTOU C, BHUNIA S, et al. Towards Trojan free trusted ICs: problem analysis and detection scheme [C] //Proceedings of Design, Automation and Test in Europe (DATE08). Munich: IEEE, 2008: 1362-1365.

[8] BANGA M, HSIAO M. A region based approach for the identification of hardware Trojans [C] //Proceedings of Hardware-oriented Security and Trust (HOST08). Anaheim: IEEE, 2008: 40-47.

[9] BANGA M, HSIAO M. A novel sustained vector technique for the detection of hardware Trojans [C] //Proceedings of the 22nd International Conference on VLSI Design. New Delhi: IEEE, 2009: 327-332.

[10] WANG X, TEHRANIPOOR M, PLUSQUELLIC J. Detecting malicious inclusions in secure hardware: challenges and solutions [C] //Proceedings of Hardware-oriented Security and Trust (HOST08). Anaheim: IEEE, 2008: 15-19.

[11] ZHANG J, YUAN F, WEI L, et al. VeriTrust: verification for hardware trust [C] //Proceedings of Computer-aided Design of Integrated Circuits and Systems. Anaheim: IEEE, 2013: 1148-1161.

[12] BHUNIA S, HSIAO M, BANGA M. Hardware Trojan attacks: threat analysis and countermeasures [J]. IEEE Journal Impact Factor & Information, 2014, 102(8): 1229-1247.

[13] BHUNIA S, ABRAMOVICI M, AGRAWAL D, et al. Protection against hardware Trojan attacks: towards a comprehensive solution [J]. IEEE Design & Test, 2013, 30(3): 6-17.

[14] CHAKRABORTY R, WOLFF F, PAUL S, et al. MERO: a statistical approach for hardware Trojan detection [C] //Proceedings of Cryptographic Hardware and Embedded Systems (CHES09). Berlin: IEEE, 2009: 396-410.

[15] POMERANZ I, REDDY S. A measure of quality for n-detection test sets [J]. IEEE Transactions on Computers, 2004, 53(11): 1497-1503.

[16] WAKSMAN A, SUOZZO M, SETHUMADHAVAN S. FANCI: identification of stealthy malicious logic using boolean functional analysis [C] //Proceedings of the 2013 ACM SIGSAC Conference on Computer &

Communications Security (CCS13). Berlin: IEEE, 2013:697-708.

[17] CHAKRABORTY R, NARASIMHAN S, BHUNIA S. Hardware Trojan: threats and emerging solutions [C] //Proceedings of High Level Design Validation and Test Workshop. San Francisco: IEEE, 2009: 166-171.

[18] AGRAWAL D, BAKTIR S, KARAKOYUNLU D, et al. Trojan detection using IC fingerprinting [C] // Proceedings of Security and Privacy (SP07). Berkeley: IEEE, 2007:296-310.

[19] AARESTAD J, ACHARYYA D, RAD R, et al. Detecting Trojans though leakage current analysis using multiple supply pad IDDQs [J]. IEEE Transactions on Information Forensics Security, 2010,5(4):893-904.

[20] Alkabani Y, Koushanfar F. Consistency-based characterization for IC Trojan detection [C] //Proceedings of Computer-aided Design-digest of Technical Papers (ICCAD09). San Jose: IEEE, 2009:123-127.

[21] POTKONJAK M, NAHAPETIAN A, NELSON M, et al. Hardware Trojan horse detection using gate-level characterization [C] //Proceedings of Design Automation Conference (DAC09). San Francisco: IEEE, 2009:688-693.

[22] WEI S, POTKONJAK M. Scalable hardware Trojan diagnosis [J]. IEEE Transactions on Very Large Scale Integration Systems, 2012,20(6):1049-1057.

[23] NARASIMHAN S, DONGDONG D, CHAKRABORTY R, et al. Hardware Trojan detection by multiple-parameter side-channel analysis [J]. IEEE Transactions on Computers, 2013,62(11):2183-2195.

[24] DU D, NARASIMHAN S, CHAKRABORTY R, et al. Self-referencing: a scalable side-channel approach for hardware Trojan detection [C] // Proceedings of Cryptographic Hardware and Embedded Systems (CHES10). Berlin: Springer, 2010: 173-187.

[25] WILLIAMS M. Anti-Trojan and Trojan detection with in-kernel digital signature testing of executables [R]. New Delhi: Security Software Engineering NetXSecure NZ Limited, 2002.

[26] BANGA M, CHANDRASEKAR M, FANG L, et al. Guided test generation for isolation and detection of embedded Trojans in ICs [C] // Proceedings of the 18th ACM Great Lakes Symposium on VLSI. New York: ACM, 2008:363-366.

[27] WANG X, SALMANI H, TEHRANIPOOR M, et al. Hardware trojan detection and isolation using current integration and localized current analysis [C] // Proceedings of Fault and Defect Tolerance of VLSI Systems (DFT08). Boston: IEEE, 2008:87-95.

[28] JIN Y, MAKRIS Y. Hardware Trojan detection using path delay fingerprint [C] //Proceedings of Hardware-oriented Security and Trust. Anaheim: IEEE, 2008:51-57.

[29] RAI D, LACH J. Performance of delay-based Trojan detection techniques under parameter variations [C] //Proceedings of Hardware-oriented Security and Trust (HOST09). Francisco: IEEE, 2009:58-65.

[30] LI J, LACH J. At-speed delay characterization for IC authentication and Trojan horse detection [C] // Proceedings of Hardware-oriented Security and Trust (HOST08). Anaheim: IEEE, 2008:8-14.

[31] NARASIMHAN S, DU D, CHAKRABORTY S, et al. Multiple-parameter side-channel analysis: a non-invasive hardware Trojan detection approach [C] // Proceedings of Hardware-oriented Security and Trust (HOST10). Anaheim: IEEE, 2010:13-18.

[32] ZHANG X, TEHRANIPOOR M. Case study: detecting hardware Trojans in third-party digital IP cores [C] // Proceedings of Hardware-oriented Security and Trust (HOST11). San Diego: IEEE, 2011:67-70.

[33] JHA S, JHA S K. Randomization based probabilistic approach to detect Trojan circuits [C] //Proceedings of High Assurance Systems Engineering Symposium (HASE08). Nanjing: IEEE, 2008:117-124.

[34] IMESON F, EMTENAN A, GARG S, et al. Securing computer hardware using 3D integrated circuit (IC) technology and split manufacturing for obfuscation [C] //Proceedings of the 22nd USENIX Security Symposium. Washington D. C.: USENIX, 2013: 495-510.

[35] XIAO K, TEHRANIPOOR M. BISA: built-in-self-authentication for preventing hardware Trojan insertion [C] //Proceedings of Hardware-oriented Security and Trust. Austin: IEEE, 2013:45-50.

[36] CHAKRABORTY R, PAUL S, BHUNIA S. On-demand transparency for improving hardware Trojan detectability [C] //Proceedings of Hardware-oriented Security and Trust (HOST08). Anaheim: IEEE, 2008:48-50.

[37] TEHRANIPOOR M, SALMANI H, ZHANG X, et al. Trustworthy hardware: Trojan detection and design-for-trust challenges [J]. IEEE Computer Society, 2011,44(7):66-74.

[38] SALMANI H, TEHRANIPOOR M, PLUSQUELLIC J.

A novel technique for improving hardware Trojan detection and reducing Trojan activation time [J]. IEEE Transactions on VLSI, 2011,20(1):112-125.

[39] SALMANI H, TEHRANIPOOR M. A layout-aware approach for improving localized switching to detect hardware Trojans in integrated circuits [C] // Proceedings of Information Forensics and Security (WIFS10). Seattle: IEEE, 2010:1-6.

[40] BHUNIA S, ABRAMOVICI M, AGRAWAL D, et al. Protection against hardware Trojan attacks: towards a comprehensive solution [J]. IEEE Design &Test of Computers, 2013,30(3):6-17.

[41] REDDY V, AL-ZAWAWI A S, ROTENBERG E. Assertion-based microarchitecture design for improved fault tolerance [C] //Proceedings of International Conference on Computer Design (ICCD06). San Jose: IEEE, 2006:362-369.

[42] MCINTYRE D, WOLFF F, PAPACHRISTOU C, et al. Dynamic evaluation of hardware trust [C] // Proceedings of Hardware-oriented Security and Trust (HOST09). Francisco: IEEE, 2009:108-111.

[43] BLOOM G, NARAHARI B, SIMHA R. OS support for detecting Trojan circuit attacks [C] //Proceedings of Hardware-oriented Security Trust (HOST09). Francisco: IEEE, 2009:100-103.

(责任编辑:赵薇)