

一种低触发率的硬件木马设计

李 振, 李 磊, 冯稳锋, 唐 棠, 房 磊

(电子科技大学 电子科学技术研究院, 四川 成都 611731)

摘 要: 在当今的集成电路设计、制造过程中, 用户使用的芯片可能被植入硬件木马电路, 这给用户带来不可预估的危害, 故硬件木马也成为近年来研究的热点问题. 首先介绍了硬件木马的概念, 进而引出了低触发率木马的特点, 然后以乘法器为参考电路, 分析了触发概率对原始电路面积、功耗的影响, 最后验证了木马的有效性.

关键词: 硬件木马; 触发概率; 乘法器

中图分类号: TN403

文献标识码: A

文章编号: 1000-7180(2014)05-0175-04

The Design of Hardware Trojan with a Low Activation Probability

LI Zhen, LI Lei, FENG Wen-feng, TANG Tang, FANG Lei

(Research Institute of Electronic Science and Technology, UESTC, Chengdu 611731, China)

Abstract: In today's Integrated Circuits design and fabrication process, the users' chips are likely to be implanted Hardware Trojans, which brings unpredictable hazards. The Hardware Trojan has become a research focus in recently years. Firstly, the concept of Hardware Trojan and the feature of low activation probability are given. And then insert Trojan in the multiplier circuit, analysis the influence of activation probability to original trigger circuit in power and consumption. Lastly, verified the validity of Trojan.

Key words: Hardware Trojan; activation probability; multiplier

1 引言

集成电路的全球化致使设计者对制造过程的控制减少, 为了降低制造成本, 缩短上市周期, 不可避免地要使用第三方的 EDA 工具, 第三方提供的 IP 核. 在芯片的整个生命周期, 从芯片的设计到生产, 包括 RTL 代码级、门级网表、版图级等等都可能被攻击者恶意加入木马^[1], 这给芯片安全带来严重危害. 文献[2]中介绍了木马设计的 8 个实例, 成功地应用在实际系统中, 并定性分析了每种木马被检测出的难易程度. 随着芯片的功能越来越多, 设计越来越复杂, 要想检测出硬件木马的存在确实很困难. 本文在乘法器中插入木马, 并在适当的时候激活, 改变乘法器的输出结果. 通过实验, 计算出每个木马被激活的概率, 以及对面积和功耗的影响.

2 硬件木马简介

2.1 硬件木马概念与激活方式

硬件木马是指插入到原始电路的恶意电路, 通常相对于原始电路来说规模较小, 在电路运行到特定条件时被激活, 使芯片发生功能失效, 泄露芯片信息, 甚至使芯片遭到不可恢复的破坏.

木马的分类方式很多, 在文献[3]中有详细的木马攻击模型的介绍. 本文中植入的木马是功能型木马, 分为触发部分和有效载荷^[1], 触发部分就是激活木马的方式, 不工作时处于休眠状态, 在一定的特殊条件、事件下, 木马被激活, 有效载荷就是木马被触发后发挥功能的电路^[3], 即向电路注入错误. 木马的激活又分为部分激活和全激活. 部分激活是指在电路的内部节点的状态被激活, 但在芯片的输出管脚不一定能检测到错误, 全激活是指在电路输出管脚

检测到错误.图 1 所示为一个由多个基本门所构成的木马电路^[1]. Trojan In1、In2、In3、In4 可以是原电路中或刻意添加的逻辑,当满足 4'b0010 时输入至有效载荷与门的值为 0,若 Data Input 此时为 1,则 Payload Output 的值变为 0,木马被激活改变了原有的逻辑值.

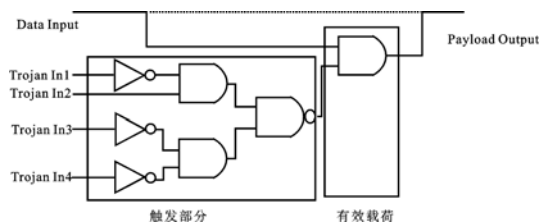


图 1 由基本门构成的木马电路

2.2 常用检测方法

近几年来,硬件木马成为集成电路中新的研究热点,检测方法发展也很迅速,但木马具有很高的隐蔽性,在芯片正常工作时一般处于休眠状态,加之木马电路相对于原始设计相对较小,很多攻击者选取电路中极少出现的状态节点,使木马检测相当困难.

文献[4]介绍了一种物理验证方法,这种方法不仅昂贵而且费时,对大规模的电路无能为力,并且对样本的检测并不能保证所有的芯片都是可靠的.在文献[5],作者介绍了指纹验证的概念,每个芯片都有其自己的特征,用 IC 指纹法检测是通过测试 IC 的功耗得到一个“指纹”,即原始电路的功率曲线,然后用这个 IC 指纹认证其他需要测试的芯片.在文献[6]中,作者提出了高速延迟技术,可以在运行、测试阶段检测电路,路径延迟是在电路中插入锁存器,每个周期锁存一次数据与原始的电路数据做比较.在文献[7]中作者提出了基于功耗延时的技术.在文献[8]中,提出了通过插入寄存器来提高木马激活概率的方法,可以用来辅助旁路检测技术,也可以用在逻辑测试时观察由木马引起的逻辑错误.

2.3 低检测率硬件木马的可行性

下面的原因可以解释为何标准的测试方法对检查硬件木马无效:

(1)预料之外的行为并不在错误列表之内,结构测试并不覆盖木马测试向量;

(2)如果对插入的木马电路没有任何的先验知识,是很难发现错误的;

(3)当芯片变得很复杂时,无穷尽的输入测试是很难做到的.

低检测率硬件木马从验证时间的角度来看要满足:一是要让木马的触发概率低,二是木马在电路运

行足够长时,木马能被激活.第一点保证需要长时间的检测时间才能发现木马的存在,第二点保证木马的有效性.

木马的低触发特性由两点可以保证:(1)利用电路的自身属性,如 32 位的数据总线,当检测到某一特殊输入时激活木马,在输入 32 位值的概率都相等的情况下,触发概率为 $\frac{1}{2^{32}}$,在仿真中,平均 4 294 967 296 个时钟周期才能发现一个错误.(2)类似于计数器这种, k 位的计数器,可以使木马的触发概率为 $\frac{1}{2^k}$, k 的位数越大,木马的触发概率越低.在芯片的功能验证中,不会去验证总线上的每个数值,若总线上输入几个特定的数据才满足激活木马的条件,检测会更加困难,甚至不可能检测到木马.本文基于这两点设计硬件木马.

3 乘法器木马设计

3.1 乘法器设计

乘法器是计算机运算不可缺少的部分,在数字信号处理,如卷积运算、数字滤波、FFT 等都有重要的应用.本论文利用移位加乘法器,共设计 8 位和 32 位两种乘法器,并向上构成多个 8 位或 32 位乘法器.

乘法器为移位加乘法器,以 8 位乘法器为例.其原理为:先把被乘数高位补 0,扩展成 16 位,乘数依次右移,判断其最低位是否为 1,若为 1 则将乘数与存放临时结果的变量相加,然后将被乘数左移 1 位,若为 0,则直接将被乘数左移.每个时钟迭代 1 次,8 个时钟才能完成计算.

本设计模块输入包括时钟 clk,复位信号 rst_n,开始信号 start_sig,被乘数 a,乘数 b,均为 8 位,输出 16 位乘法结果 product,运算完成标志 done_sig.其 32 位乘法器设计与其类似.并由这两个乘法器向上构成多个乘法器.其消耗资源分别如表 1 所示,时钟约束为 20 ns,采用 smc0.13 μm 标准库.

表 1 各乘法器面积、功耗

乘法器位数	乘法器个数	综合后面积/ μm^2	功耗/ μW
8 位乘法器	1	11 150.41	64.09
	2	22 348.35	128.86
32 位乘法器	1	51 714.87	270.71
	4	207 081.29	1 094.55
	16	828 388.51	4 363.31
	32	1 656 734.78	8 678.21

3.2 木马插入与概率计算

若判断乘数、被乘数的某些位符合预判值时改变乘法的运算结果. 为简单起见, 只改变运算结果的最低位, 进行取反操作. 共设计两类木马.

(1) 第一类木马

若被乘数的某些位符合木马激活条件, 则激活木马, 设计如表 2 所示的木马.

表 2 第一类木马设计

乘法器位数	乘法器个数	木马激活条件	激活概率	综合后面积/ μm^2	功耗/ μW
8 位乘法器	1	$a[0] = 1'b1$	1/2	11 179.45	64.40
	1	$a[0] = 1'b1$ & $b[0] = 1'b1$	1/4	11 197.94	64.38
	1	$a[3:0] = 4'b1010$	1/16	11 237.54	64.21
	1	$a[3:0] = 4'b1010$ & $b[3:0] = 4'b0101$	1/256	11 311.47	64.20
	1	$a = 8'b10101010$ & $b = 8'b01010101$	$\frac{1}{2^{16}}$	11 459.34	64.26
	2	两个乘法器 $a = 8'b10101010$ & $b = 8'b01010101$	$\frac{1}{2^{32}}$	22 966.20	128.15
32 位乘法器	1	$a = 32'd123$ & $b = 32'd321$	$\frac{1}{2^{64}}$	52 844.96	272.04
	4	每个乘法器都符合 $a = 32'd123$ & $b = 32'd321$	$\frac{1}{2^{128}}$	211 955.46	1 099.63
	16	每个乘法器都符合 $a = 32'd123$ & $b = 32'd321$	$\frac{1}{2^{512}}$	848 136.06	4 384.68
	32	每个乘法器都符合 $a = 32'd123$ & $b = 32'd321$	$\frac{1}{2^{1024}}$	1 696 335.50	8 718.74

从表 2 中可以看到, 利用乘法器的特殊输入条件就可以激活木马, 而且面积、功耗增加都很少, 在文献[5]中, 介绍在 $\pm 7.5\%$ 的工艺变量的情况下, 指纹检测能检测出芯片电路中不同类型的硬件木马. 但现在纳米级 IC 的工艺变量往往大于 $\pm 7.5\%$ 这个范围, 并且 1 个 32 位乘法器仅凭乘数、被乘数激活木马的概率就可以降低到, 面积增加不大. 在实际的应用中, 如 GPU 运算会应用大量的乘法器, 插入木马后相对于芯片电路面积变化更小, 并且实际应用中, 乘数、被乘数的输入概率并不是相等的, 如果设计者对此了解, 可以设计出更低的木马激活概率.

(2) 第二类木马

利用计数器、状态机和乘数、被乘数结合设计木马. 以一个 8 位乘法器为例, 当检测到 $a = 8'd1, 8'd2, 8'd3$ 时激活木马, 当要求连续时用状态机实现, 激活概率 $\frac{1}{2^{24}}$; 当为不连续时既可以用状态机, 又可以简单的使用计数器, 本文都统一使用状态机, 激活概率为 $\frac{1}{3 \times 2^8}$, 使第三次乘法计算的结果出错. 表 3

列出了根据此方法设计出的木马.

表 3 第二类木马设计

木马激活条件	激活概率	面积/ μm^2	功耗/ μW
$a = 8'd1, 8'd2$ (不连续)	$\frac{1}{2 \times 2^8}$	12 161.68	67.82
$a = 8'd1, 8'd2$ (连续)	$\frac{1}{2^{16}}$	11 997.98	67.46
$a = 8'd1, 8'd2$ & $b = 8'd1, d'2$ (不连续)	$\frac{1}{2 \times 2^{16}}$	12 277.86	68.07
$a = 8'd1, 8'd2$ & $b = 8'd1, d'2$ (连续)	$\frac{1}{2^{32}}$	12 219.77	67.89
$a = 8'd1, 8'd2, 8'd3$ (连续)	$\frac{1}{2^{24}}$	12 180.16	69.67
$a = 8'd1, 8'd2, 8'd3, 8'd4$ (连续)	$\frac{1}{2^{32}}$	12 436.28	69.38

3.3 木马对比

从上面表 2、表 3 可以得到如下两个结论:

(1) 用相同的木马设计方法, 如第一类木马的设计, 随着木马触发概率的降低, 所付出的代价(面积、功耗)也随之上升, 如图 2(a)、(b)所示. 其中横坐标用处理后的触发概率值 $\log_2(-\log_2 P)$ 表示(为了坐标显示方便, P 表示原有的触发概率), 图 2(a)中的纵坐标表示面积变化的百分比, 图 2(b)中的纵坐标表示功耗的变化百分比.

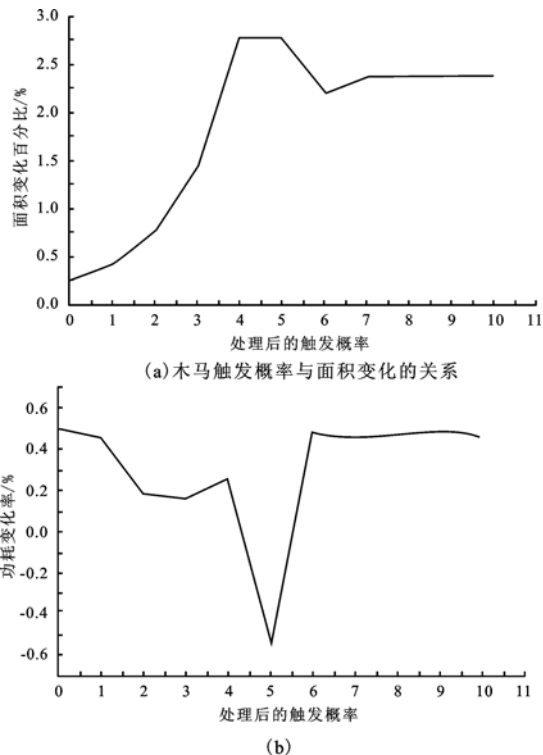


图 2 木马电路对面积、功耗的影响

(2) 付出相同的代价, 用不同的木马设计方法, 可以得到不同的触发概率, 如第二类木马, 好的设计方法不仅可以得到较低的触发概率, 所付出的代价也较小.

4 木马检测

4.1 木马仿真

仍然以一个 8 位乘法器为例,当 $a[3:0] = 4'b1010$ 时激活木马,激活概率为 $\frac{1}{16}$. 为了便于观察木马是否被激活,用文件记录测试结果,第一列记录被乘数 a ,第二列记录乘数 b ,第三列记录通过乘法器记录的结果,第四列为正确的计算结果,第五列标识运算结果是否正确,pass 表示运算正确,fail 表示运算错误. 文件部分截图如图 3 所示.

197	170	33490	33180	--pass
229	119	27251	27251	--pass
18	143	2574	2574	--pass
242	206	49852	49852	--pass
232	197	45704	45704	--pass
92	189	17388	17388	--pass
58	101	5859	5858	--fail!!!
99	10	990	990	--pass
128	32	4096	4096	--pass

图 3 木马验证文件结果

4.2 FPGA 测试

为了方便验证触发概率很低的木马,比如验证 $a = 32'd123$,并且 $b = 32'd321$,激活一个 32 位乘法器中木马电路. 使用 ml501 开发板,由于激活概率太低,让乘数 b 一直等于 $32'd321$,被乘数存储在 RAM 中,每次取出后加 1 写入 RAM 中,这样才能遍历被乘数,最终通过 LED 灯观测木马是否被激活,测试框图如图 4 所示.

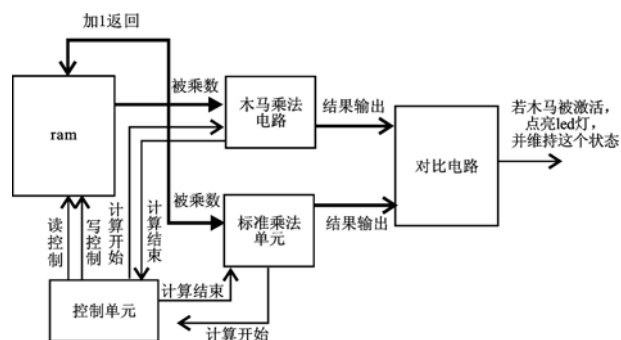


图 4 FPGA 测试木马电路

5 结束语

硬件木马的研究近年来成为新的研究热点,本文利用数字逻辑电路在乘法器中插入木马电路,验证可以在面积、功耗改变很小的情况下,插入激活概率很低的木马,证明了硬件木马在实际应用中的可行性,为以后的木马设计、检测工作做铺垫.

参考文献:

- [1] Chakraborty RS, Narasimhan S, Bhunia S. Hardware trojan: threats and emerging solutions. Proceedings of the international [C]// High Level Design Validation and Test Workshop. San Francisco:IEEE,2009:166-171.
- [2] Yier Jin, Kupp N, Makris Y. Experiences in Hardware trojan design and implementation[C]//IEEE International Workshop on. Hardware-Oriented Security and Trust. San Francisco:IEEE,2009:50-57.
- [3] 郑朝霞,韩玲,李阳,等. 一种木马电路的实现与特征分析. 微电子学与计算机,2012,29(10):78-80.
- [4] Sanno B. Detecting hardware trojans[EB/OL]. [2013-04-20]. http://www.crypto.rub.de/imperia/md/content/seminare/its09/benjamin_sanno_semembsec_termpaper_20090723_final.pdf, 2009.
- [5] Agrawal D, Baktir S, Karakoyunlu D, et al. Trojan detection using ic fingerprinting[C]// Proceedings of the Berkeley, CA:IEEE, Security and Privacy, 2007.
- [6] Li J, Lach J. At-speed delay characterization for IC authentication and trojan horse detection[C]//IEEE International Workshop on. Hardware-Oriented Security and Trust Anaheim, CA, USA:IEEE,2008:8-14.
- [7] Rad R, Plusquellic J, Tehranipoor M. A sensitivity analysis of power signal methods for detecting hardware trojans under real process and environmental conditions [J]. IEEE Transactions on. Very Large Scale Integration (VLSI) Systems, 2010,18(12):1735-1744.
- [8] 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 (VLSI) Systems, 2012, 20(1): 112-115.

作者简介:

李 振 男,(1988-),硕士. 研究方向为低检测率硬件木马设计.

冯稳锋 男,(1989-),硕士. 研究方向为硬件木马植入技术.

唐 棠 男,(1989-),硕士. 研究方向为硬件木马旁路检测技术.

房 磊 男,(1989-),硕士. 研究方向为低触发率硬件木马检测技术.

李 磊 男,(1982-),研究员. 研究方向为专用集成电路(ASIC)研究.