

基于 RTL 级硬件木马的检测方法

成 祥,李 磊,程 伟

(电子科技大学 电子科学技术研究院,四川 成都 611731)

摘 要: 随着集成电路产业的飞速发展和产业布局的改变,交由第三方设计制造的芯片越来越流行,如何保证芯片设计安全成为了人们日益关注的问题.对此,以目前硬件木马检测的主要方法为背景,以 AES 木马为基准电路,从 RTL 级语言结构出发,依据 perl 语言强大的文本匹配能力,提出了一种基于 Verilog IEEE 标准的 RTL 级硬件木马的分析方法,以期从源头上保证硬件木马设计的安全性.

关键词: 硬件木马;AES 基准木马;RTL 级;Verilog;perl

中图分类号: TN492

文献标识码: A

文章编号: 1000—7180(2017)03—0056—05

A Detection Method of Hardware Trojans Based on RTL

CHENG Xiang, LI Lei, CHENG Wei

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

Abstract: With the development of the integrated circuits and the transform of industrial distribution, it becomes more and more popular for some company to rely on the third party to design their chips. To ensure the security of chip design flow has become the concern of people. With the context of the main detections of Hardware Trojan, the paper proposed a method to detect the RTL level HTs, based on the verilog IEEE characteristics and the AES benchmarks and the powerful capability of text-matching of perl, thus, to ensure security from the initial design.

Key words: hardware tojan; AES benchmarks; RTL level; verilog; perl

1 引言

近些年,随着集成电路行业的飞速发展以及第三方加工生产的特性,如何保证整个芯片设计流程的可靠性和稳定性,日益成为人们关心和关注的焦点,尤其是我国芯片行业还处于刚刚起步的阶段,很多时候都是外包给芯片设计公司来完成整个设计,第三方芯片的大量使用对我国的国防和基础建设安全产生了非常大的潜在威胁.

本文针对硬件木马在 RTL 级设计的结构和特点,以 Verilog IEEE 为准,以 AES 基准电路为依托,依据 perl 强大的文本匹配能力,研究一种 RTL 级硬件木马的分析和检测方法,以期从源头上阻止硬件木马的插入.

2 硬件木马简介

2.1 硬件木马定义

硬件木马(Hardware Trojan),是指设计者在集成电路的设计过程中,在原有的目标硬件电路上插入具有恶意功能的冗余电路.一般来讲,硬件木马电路由两个部分构成:触发电路和执行电路,如图 1 所示.触发电路是木马电路的激活机制,这一部分通常是由芯片的内部信号或者外部信号完成,等待触发条件满足(电路达到某种状态,或者电路运行到特定的时间).执行电路是指木马电路对原始电路发生作用的那部分,执行电路的效果一般是改变原始电路功能、改变原始电路参数、泄露关键信息、降低系统性能甚至拒绝服务,乃至毁坏芯片.

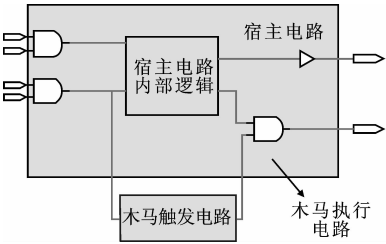


图 1 木马电路模型

2.2 目前常用的检测方法

随着硬件木马的出现,检测方法也随之变得多种多样,但基于 Verilog 的 RTL 级硬件木马检测方法还很少有人进行研究.

文献[1]是基于旁路分析的检测技术,分别从功耗和时序信号分析木马,但是当木马电路规模小到不足以使旁路信号发生较大改变时,就很难检测到木马了. 文献[2]则利用电路平均动态电流与最大工作频率来分析,但需在工艺噪声一定的前提下. 文献[3]是通过在芯片中所有未被开发利用的空间里植入 BISA 单元来阻止任何木马的插入,同时自身构成一个功能模块来侦测硬件木马插入与否,但需要确定芯片剩余了哪些空间,又要确保插入的单元不影响芯片本身的功能,文献[4]则是通过逆向工程获取芯片的门级网表,然后在门级网表的基础上检测木马,此种方法属破坏性检测方法,需人工分析网表,效率较低. 另外,文献[5]通过在 RTL 中插入计数器来设计和控制木马,文献[6]也是通过 RTL 级控制木马,说明 RTL 级木马的插入必定有一定的规则可追寻. 文献[7]则是通过 ATPG 工具处理简单和稍微复杂的 RTL 代码,然后再通过区域隔离的方式进行分析,进而确定可信的设计.

综上所述的各种检测方法,为本文提出 RTL 级木马的检测方法提供了背景,尤其是文献[5-6]在 RTL 级设计木马更是为本文提供了逆向分析的思路.

3 基于 RTL 的硬件木马分析方法研究

3.1 硬件木马检测流程

RTL 级硬件木马的检测流程如图 2 所示,整个流程的每一步到最后将检测结果生成报告都由相应特定的 perl 脚本处理完成,之所以选择 perl 来处理源文件主要原因有两点:一是 perl 编程语言依托于其强大的正则表达式匹配能力对文本有很强的处理能力;二是逻辑简单清晰,可以将 RTL 源文件原本的映射到检测结果而不增加额外的逻辑,保证了处

理前后功能上的一致性. 另外,从整个流程可以看出,本文将只对 RTL 文件进行处理,提取所需要的特征点,对其实现的功能和内部逻辑结构并不进行深入分析,从而避免了对大型复杂设计分析时的功能映射,也降低了从源文件映射到检测流程出错的概率.

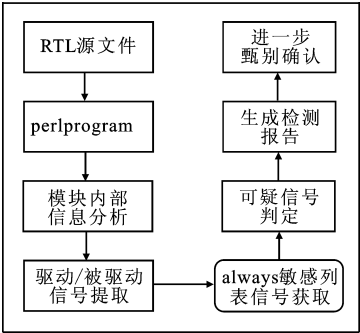


图 2 硬件木马检测流程图

3.2 RTL 源文件分析

任何一个 RTL 文件都离不开一些 Verilog 的构成元素,它们可以在各种抽象层次上描述硬件的行为. 依据 Verilog-2005 IEEE 标准可知,Verilog 是一个并行的编程语言,其构成元素的执行是由块(或进程)的并行执行定义的,而进程可以有计算,可以有状态,能够响应输入变化,能够改变输出信号. 进程包括原语、模块、initial 块、always 块、连续赋值、异步任务和过程赋值语句等. Verilog 的基本设计单元称为模块,由描述接口和描述逻辑功能两部分组成,通常一个模块中包括模块名、I/O 端口说明、内部信号(wire/reg 类型)定义、功能定义(always 块/assign 语句等)等几部分组成. 由于大部分模块内部参数和整数基本都是用来表示信号位数或者方便模块被复用是更改的,所以 perl 脚本处理时将自动忽略掉这些参数信息. 具体处理 verilog 的部分如图 3 所示,perl 将会遍历模块内部每一行信息进行正则匹配,包括 I/O 口、内部信号、功能模块中的驱动和被驱动信号,并将结果以文本形式保存,便于后面的处理调用. 其中,驱动信号和被驱动信号分别是阻塞赋值或非阻塞赋值符号的左侧和右侧的表达式中所包含的信号信息.

3.3 模块功能分析

在任何一个模块中最重要的就是逻辑功能定义部分,在实现整个逻辑功能的语句中,除了常用的 assign 语句和 always 块之外,还要用到一些条件语句(if-else)、case 语句和其他循环语句(while、for 等),但在对其进行功能分析的过程中,重点分析语

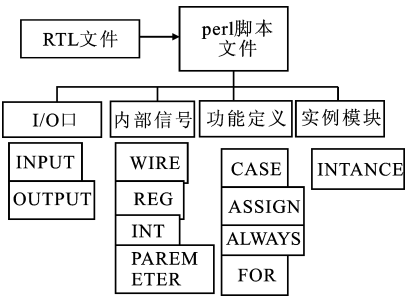


图 3 VERILOG 基本要素

句将是条件语句、always 实现组合逻辑时敏感信号的变化、驱动信号和被驱动信号的提取这三部分. 这一步将引入 perl 编程语言中的一种很重要的数据结构哈希, 如图 4 所示, 哈希结构中的键(keys)和值(values)是一一对应的关系, 分别是胖箭头(=>)左边和右边的字符串, keys 函数和 values 函数分别用于存储提取的被驱动信号和驱动信号, 方便了驱动信号和被驱动信号的查找和定位. 之所以要引入哈希结构在于它可以容纳任意多的值并能按需取用, 可以应对大型模块设计中信号较多的情况.

```
my %hash =(  
    'fred' => 'flintstone',  
    'dino'  => 'undef',  
    'barney'=> 'rubble',  
    'betty' => 'knife',  
);
```

图 4 哈希结构图

Perl 脚本在处理模块功能时将采用正则表达式匹配所需信息的方式进行循环遍历 RTL 文件中的信息, 直到所有的信息遍历完为止, 并将对应提取的信息以文本的方式进行保存. 如图 5 所示, 分别是处理文件脚本所用到的部分正则表达式, 依次查找模块名、I/O 端口、驱动信号、被驱动信号等, 结果会先以列表形式存储, 待遍历完整个文件后生成文本.

```
/\s * module\s+(\w+)/  
/input\s+(\[. * \])? \s*(. *)/  
/output\s+(reg)? \s*(\[. * \])? \s*(. *)/  
/assign\s+(\. *)\s*\s*=\s*(. *)/  
/(. *)\s*<=\s*(. *)/  
.....
```

图 5 部分正则表达式处理程序

通常, 木马设计者会尽可能地将电路设计成看起来是有用的部分, 在设计表达式时尽可能去显示

其功能性, 比如表 1 是文献[8]中的木马设计的 RTL 级的描述, 针对这种情况, 为了方便后面的处理, 我们采用信号互联性特性分析方法, 首先将哈希结构的中键和值以数列的形式提取到 @keys 和 @values 中, 再遍历数列 @keys 中与数列 @values 同时存在的信号元素, 然后查找 %hash 结构中的键和对应的值, 将值带入到前面遍历出的相同的信号, 循环遍历函数, 层层迭代, 最后把最终输出的信号表示成只有输入信号表示的形式.

表 1 信号互联特性处理

处理前	<pre>assign PATTEN1 = sig1 &. sig2 &. sig3 &. ~sig4; assign PATTEN2 = sig5 &. sig6 &. sig7 &. ~sig8; assign Ten = PATTEN1 &. PATTEN2; assignFunction = Ten ? FTrojan:Fnormal;</pre>
处理后	<pre>Function = (sig1 &. sig2 &. sig3 &. ~sig4&.sig5 &. sig6 &. sig7 &. ~sig8)? FTrojan:Fnormal;</pre>

3.4 可疑信号的判断标准

由于硬件木马隐蔽性强, 为了躲避多种检测机制通常需要宿主电路运行一段时间或者输入特定序列等才能触发, 反映到 RTL 设计阶段则通常会采用计数器达到特定值, 特定序列赋值给木马启动信号, 启动信号由多个信号的某一特定值共同满足时驱动等多种形式, 从当前角度出发, 本文在处理 RTL 级木马时从三个角度去判定: DOS(Denial-of-Service)型木马、触发性(Trigger)木马、混合型木马. DOS 木马主要攻击电路的时钟信号, 触发性木马主要攻击电路的条件判断语句, 主要判定方式有以下几点. A: 子模块时钟信号直接被数值驱动, 造成子模块不正常工作; B: 被驱动信号出现了子模块时钟信号, 而驱动它的信号却不是时钟类型的信号; C: always 块敏感列表中的信号在模块内部出现的概率特别小或者一直满足条件; D: if 条件语句中的条件直接出现特定序列时视为可疑条件; E: 驱动信号列表中出现了四个信号及以上且表达式中夹杂着很多降低被驱动信号出现概率的符号, 如与、或、异或等符号的, 视为降低启动信号概率的可疑条件; F: 同一种信号通过不断增加逻辑增加电路的功耗; G: 计数器会出现一个一般很难达到的满状态, 比如 16'h0000 到 16'hffff; 其中, 前两种为 DOS 型木马, 后几种为触发型木马, 如果文件中出现了上述的两种以上, 就是为混合型木马.

另外, 很多 RTL 级木马设计者为了避免现已公开的 RTL 级木马检测 UCI 技术, 常常会把电路伪装成有用的电路, 因为 UCI 技术是将处于休眠状

态,在验证阶段没有被激活的电路视为可疑电路,而本文上述的判断标准正好弥补 UCI 技术的不足,例如文献[6]在 RTL 级设计的木马虽然有效避开了 UCI 技术,但是却忽略了降低概率所采取的措施,

最后的触发信号 $f \leq (t_en_1 \& t_en_2 \& \dots \& t_en_k) \& fm \mid (\sim t_en_1 \& \sim t_en_2 \& \dots \& \sim t_en_k) \& (c1 \& f1 \mid c2 \& f2)$ 正好违反了上述判断准则的 E 标准,将很容易被检测出.

表 2 AES 基准木马电路分析

Benchmark	Effect	Module	Same benchmark	Type	Malicious statement
AES-T100	leak info	lfsr_counter	T-200,	D,E	assign d0 = lfsr_stream[15] ^ lfsr_stream[11] ^ lfsr_stream[7] ^ lfsr_stream[0]; lfsr_stream <= "10011001100110011001";
AES-T300	leak info	TSC	T1300, T1400, T1500	E	enable1<= (state[0]&.rk1[0]) ^ (state[1]&.rk1[1]) ^ (state[2]&.rk1[2]) ^ (state[3]&.rk1[3]) ^ (state[4]&.rk1[4]) ^ (state[5]&.rk1[5]) ^ (state[6]&.rk1[6]) ^ (state[7]&.rk1[7]);
AES-T400	leak info	Trojan Trigger	T1800	D	if(state== 128hFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF)
AES-T500	Dos	TSC	T-800, T2000, T2100	D,E	if(rst==1)DynamicPower<= 128'haaaaaaaa_aaaaaaaaa_aaaa_aaaa_aaaaaaaaa; if ((state == 128'h00112233_44556677_8899aabb_ccddeeff) &. (State0 == 1)); Tj_Trig <= State0 &. State1 &. State2 &. State3;
AES-T600	leak info	TSC	T2000, T2100	F	assign INV1_out = ~(LEAKBit); assign INV2_out = ~(INV1_out);.....; assign INV10_out = ~(INV1_out); assign INV11_out = ~(INV1_out);
AES-T700	leak info	Tro _ trigger,lfsr	T1000, T1100, T1200	D	if (state == 128h00112233_44556677_8899aabb_ccddeeff) Tj_Trig <= 1;
AES-T900	leak info	TSC	T1900, T1500, T1700,	D,G	if (Counter == 128hffff_ffff_ffff_ffff_ffff_ffff_ffff) begin Tj_Trig <= 1;end
AES-T1600	leak info	Trojan Trigger		D,E	if((state == 128'h00112233_44556677_8899aabb_ccddeeff) &. (State0 == 1)) begin State1 <= 1;end -> Detected <= State0 &. State1 &. State2 &. State3

4 实例验证

根据前面的分析,本文将从 Trust-hub 网站上获取的 AES 基准木马电路去分析 RTL 级硬件木马的检测方法,如表 2 所示,采用 perl 的正则表达式可以对上述 A-G 的各个特点分别编辑相应的脚本去提取特征信息,提取到的结果正如表中最后一列显示的信息一样,AES-T100 使用假的随机数据产生器产生一个序列,并通过预定义的输入序列进行初始化操作;AES-T300 提取到的是可疑木马在完成一轮信息加密后,如果使能信号有效,将泄露相应的一比特信号;AES-T400 当输入到预定义的序列事就会激活模块内一个未用的引脚,进行信息发送,类似的传输机制还有 T700 和 T900 两种类型;另外,T500 和 T1600 是在原来的机制上又增加了一种减小概率的操作;T600 则是在特定序列到达时,

通过一种有特定寄存器构成的电容组来存储要发送的信息.

此外,如表 3 所示,文献[5]采用计数器的方式控制木马,当 8 比特计数器达到预定义序列为 0111_1111 时,整个电路的输出将会翻转,文献[6]则是通过与前面的计数器和状态转移寄存器结合,最后通过如表 3 所示的赋值表达式来控制触发信号. 综上可以表明,无论是基于 AES 的木马设计还是大型电路中插入指定功能的木马,都离不开上述判断标准中一种或多种类型.

表 3 参考文献木马设计

参考文献	木马类型	可疑信号
[5]	D	if(counter=0111_1111) $f \leq (t_en_1 \& t_en_2 \& \dots \& t_en_k) \& fm \mid (\sim t_en_1 \& \sim t_en_2 \& \dots \& \sim t_en_k) \& (c1 \& f1 \mid c2 \& f2)$
[6]	E	

5 结束语

本文从从硬件木马结构特性的角度分析了 RTL 级木马常用的设计类型,并提出了用特定的 PERL 脚本分步去处理带有可疑木马文件的方法,并结合 AES 基准木马电路和参考文献中的木马证明了该方法的可行性,虽然在面对大型设计时可能出现可疑信号较多需要人工甄别的情况,但其在大幅度降低处理文件的时间以及完善其他硬件木马检测方法不足的方面有着独特的优势,可以在芯片设计的下一个流程前进行第一遍检查筛选,很有现实意义.

参考文献:

- [1] 唐 棠. 基于旁路分析的硬件木马检测技术研究[D]. 成都:电子科技大学,2014.
- [2] 李雄伟,王晓晗,张 阳,等. 基于多旁路综合分析的硬件木马检测方法[J], 计算机仿真, 2015, 32(3): 216-219.
- [3] Kan Xiao, Domenic Forte, Mohammed Tehranipoor, et al. A novel built-In self-authentication technique to prevent inserting hardware Trojans[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2014, 33(12):1778-1791.

- [4] 房磊. 基于门级网表的硬件木马检测技术研究[D]. 成都:电子科技大学,2014.
- [5] Liu Huafeng, Luo Hongwei, Wang Liwei, et al. Design of hardware Trojan horse based on counter[C]// IEEE International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering. Xi'an: IEEE, 2011:1007-1009.
- [6] Jie Zhang, Qiang Xu. On hardware Trojan design and implementation at register-transfer level[C]// IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). Austin: IEEE, 2013:107-112.
- [7] Mainak Banga, Michael S Hsiao. Trusted RTL: Trojan detection methodology in pre-silicon designs[C]// IEEE International Symposium on Hardware-Oriented Security and Trust. Anaheim: IEEE, 2010:56-59.
- [8] 李振. 数字类型的硬件木马设计实现与分析[D]. 成都:电子科技大学,2011.

作者简介:

成 祥 男, (1989-), 硕士研究生. 研究方向为 ASIC 可信设计流程的方法. E-mail: 85278407@qq. com.

李 磊 男, (1982-), 研究员. 研究方向为专用集成电路设计.

程 伟 男, (1987-), 硕士研究生. 研究方向为微电子与固体电子.

(上接第 55 页)

- [5] Bruce N, Tsotsos J. Attention based on information maximization [J]. Journal of Vision, 2010, 7(9):950-950.
- [6] 徐凯. 基于显著性分析的视觉注意模型研究[D]. 上海:复旦大学, 2013.
- [7] Hou X, Zhang L. Saliency Detection: A Spectral Residual Approach[C]// IEEE Conference on Computer Vision and Pattern Recognition. Minneapolis: IEEE Computer Society, 2007:1-8.
- [8] Chen L. Topological structure in visual perception. [J]. Science, 1982, 218(4573):699-700.
- [9] Eckhorn R, Reitboeck H J, Arndt M, et al. Feather linking via synchronization among distributed assemblies: Simulation of results from cat cortex [J]. Neu-

ral Computation, 1990, 2(3): 293-307.

- [10] Gu X, Yu D, Zhang L. Image shadow removal using pulse coupled neural network [J]. IEEE Transactions on Neural Networks, 2005, 16(3):692 - 698.
- [11] Ell T A, Sangwine S J. Hypercomplex fourier transforms of color images [J]. IEEE Transactions on Image Processing, 2007, 16(1):22-35.

作者简介:

王健丞 男, (1989-), 硕士研究生. 研究方向为图像处理. E-mail: 14210720044@fudan. edu. cn.

顾晓东 男, (1970-), 博士, 博士后, 教授. 研究方向为人工神经网络、图像处理、模式识别、生物建模等.