

# 基于故障传播模型的硬件安全性 与可靠性验证方法

张茜歌<sup>1</sup>, 朱嘉诚<sup>2</sup>, 马俊<sup>1</sup>, 沈利香<sup>2</sup>, 周佳慧<sup>1</sup>, 慕德俊<sup>2</sup>

(1.北京智芯微电子科技有限公司, 北京 100000; 2.西北工业大学深圳研究院, 广东 深圳 518057)

**摘要:**大规模集成电路正面临着诸如设计脆弱性、侧信道、硬件木马等安全漏洞的威胁。传统的功能测试验证方法无法遍历所有的输入空间,同样无法检测侧信道安全漏洞。现有的形式化验证方法关注硬件设计的等价性和功能的正确性,难以满足安全性和可靠性验证需求。研究面向安全性和可靠性验证的形式化模型,形成有效的硬件安全性与可靠性形式化验证方法。该方法能够从门级对集成电路进行建模,生成细粒度的形式化模型,实现对安全性与可靠性的形式化验证,可以捕捉硬件设计中潜在的安全隐患。实验结果表明该验证方法对硬件设计中存在的侧信道和硬件木马导致的信息泄露和篡改有很好的检测效果。

**关键词:**形式化模型;故障效应分析;漏洞检测;硬件安全

**中图分类号:**TP309 **文献标志码:**A **文章编号:**1000-2758(2024)01-0092-06

集成电路是电子设备的基石,是数据存储、电子通信等重要领域无可替代的核心。传统的研究通常假设底层硬件是安全、可信的,然而事实上硬件设计会由于设计缺陷<sup>[1-2]</sup>、故障攻击、硬件木马<sup>[3-4]</sup>、侧信道攻击<sup>[5-6]</sup>、X-传播<sup>[7]</sup>等原因引入安全漏洞。这些有意或无意的安全漏洞,会导致存储数据被非法篡改、敏感信息泄露、计算机系统非正常运转等严重的安全问题。

集成电路验证最常见的手段是逻辑功能测试<sup>[8]</sup>,通过观察芯片输出端的逻辑值,与原始设计的预期结果对比,判断逻辑功能是否一致。此方法有较大的局限性,具体包括:覆盖率依赖测试激励信号,设计规模庞大或包含复杂时序电路情况下,难以生成有效的激励信号,导致覆盖率低、验证不充分;验证仅针对电路逻辑功能,对于利用侧信道的安全漏洞无法进行有效检测。随着集成电路规模的日益

增大,逻辑功能测试难以满足硬件安全验证的需求。

形式化验证方法<sup>[9]</sup>是保证计算机系统可靠性的重要途径,基于数学理论的严格方法,使用数学符号抽象系统模型和系统期望性质,通过推理或图形搜索的方式验证系统的形式化模型是否满足规范。形式化方法能够避免非形式化方法产生的模糊性、二义性以及不一致性等,进而能够比较深入地检测到系统中可能存在的细微漏洞。形式化验证的优点在于能够覆盖完整的设计状态空间,且无需开发测试向量,有效地解决了逻辑功能测试的不足。

在芯片设计的形式化验证中,需要建立一个数学模型来描述电路的行为和规范,模型的精确度和准确度能够直接影响验证结果的正确性。在硬件安全领域,对硬件电路的建模主要包括2个层面:门级建模<sup>[10]</sup>和寄存器传输级建模<sup>[11]</sup>。门级建模指对电路的门级设计进行建模,精确度高但模型规模较大,验证耗时较长;寄存器传输级建模指对电路的RTL设计进行建模,精确度略低但规模较小,验证速度度块。

针对硬件的安全性与可靠性验证问题,本文提出了一种故障效应传播的建模方法,对电路设计的门级网表进行建模,并使用形式化验证方法对故障

收稿日期:2023-02-14

基金项目:北京智芯微电子科技有限公司实验室开放基金  
(SGSC0000SJQT2207164)资助

作者简介:张茜歌(1978—),研究员

通信作者:朱嘉诚(1996—),博士研究生

e-mail:zhu\_jc@mail.nwpu.edu.cn

效应传播模型进行验证。该方法能够检测诸如硬件木马、时间侧信道、X-传播等安全漏洞。

## 1 故障效应传播模型

故障效应传播是指故障数据在数字电路中的传播,最终对输出信号产生影响。在故障效应传播模型中,待验电路中的每一个信号节点都将分配一个故障属性标签,包括输入和输出节点。该标签用以表示对应信号的属性,如机密性、可靠性等。故障效应传播模型中,仅输入信号的标签通过手动方式进行标记,其他标签都由对应逻辑门的输入、标签及逻辑门类型决定。由此,输入信号的标签将在电路中进行传播,最终传播至待验电路的输出。这可以帮助识别故障影响分析的可能点,回溯故障源并选择部署容错机制的关键点。

本文采用大写字母  $A, B, O$  等表示信号的逻辑值,  $A_e, B_e$  和  $O_e$  分别对应上述信号的故障标签。其中  $A, B, O \in \{0, 1\}$ ,  $A_e, B_e, O_e \in \{\text{low}, \text{high}\}$ , “low”表示电路信号不受故障影响, “high”表示电路信号受到故障影响。

### 1.1 基本逻辑门的故障效应传播模型

基本逻辑门故障有随机故障、固定-0故障、固定-1故障、位翻转故障等。随机故障指信号节点随机输出高或低信号,与输入无关;固定-0故障指信号节点固定输出低信号;固定-1信号指信号节点固定输出高信号;位翻转故障指信号节点输出始终取反。根据不同的故障类型可以分别建立不同的故障效应传播模型,从而追溯可能存在的故障类型。随机故障导致信号节点可能正确也可能错误,需做出保守考虑,本文以随机故障举例说明基本逻辑门的故障效应传播模型建模方法。

首先以二输入与门在随机故障中的传播为例详细介绍故障标签传播策略。二输入与门中2个输入为  $A$  和  $B$ ,输出的布尔函数为  $O = A \& B$ 。表1描述了二输入与门故障传播的真值表。

表1 二输入与门在随机故障下故障状态的真值表

$(A_e, A)$	$(B_e, B)$			
	(low, 0)	(low, 1)	(high, 0)	(high, 1)
(low, 0)	(low, 0)	(low, 0)	(low, 0)	(low, 0)
(low, 1)	(low, 0)	(low, 1)	(high, X)	(high, X)
(high, 0)	(low, 0)	(high, X)	(high, X)	(high, X)
(high, 1)	(low, 0)	(high, X)	(high, X)	(high, X)

由表1可得,当输入  $A, B$  都不包含故障时,电路运行正常,输出同样不会包含故障,  $O_e$  表现为“low”;当输入  $A, B$  都包含故障时,本文出于保守考虑,假定输出  $O$  受故障影响,表现为“high”。因此,只需考虑输入故障情况下,输出的故障状态。

以信号  $A$  受随机故障影响为例,由于不能确定故障类型,所以将变成“X”态。若信号  $B$  为逻辑“0”,此时随机故障被与门过滤,输出  $O$  不受  $A$  影响,  $O_e$  将表现为“low”;若信号  $B$  为逻辑“1”,输出  $O$  将不可预知地表现为逻辑“0”或逻辑“1”,在仿真时呈现“X”态,此时  $O_e$  表现为“high”。考虑二输入与门的真值表,对应的故障效应传播模型可表示为

$$O_e = AB_e + A_e B + A_e B_e \quad (1)$$

然后考虑非门的故障传播模型。非门的布尔函数为  $O = \bar{A}$ ,其中  $A$  为非门的输入,  $O$  为非门的输出。非门对故障十分敏感,线路中的任何故障都会反映在输出上,因此其对应的故障传播模型可以表示为

$$O_e = A_e \quad (2)$$

再考虑两输入或门  $O = A + B$ ,根据德摩根定律可以得到  $\bar{O} = \bar{A} \cdot \bar{B}$ 。又根据非门的故障传播模型可知,非门输入和输出的标签相等  $O_e = \bar{O}_e$ ,可以将或门转换为两输入与门。此时  $O_e = (\bar{A} \cdot \bar{B})_e$ ,接着将公式(1)展开可得二输入或门的故障传播模型,表示为

$$O_e = \bar{A}B_e + A_e\bar{B} + A_e B_e \quad (3)$$

对于异或门  $O = A \oplus B$ ,其输出对于输入的变化很敏感,随机故障下,无法确定有多少输入会发生翻转,所以做保守的假设,只要有一个输入发生故障,输出就会受到故障影响。故二输入异或门的故障模型表示为

$$O_e = A_e + B_e \quad (4)$$

根据非门的故障效应传播模型可知,与非门、或非门、异或非门的故障效应传播模型分别和与门、或门、异或门一样,可以用公式(1),(3),(4)表示。

### 1.2 数字电路的故障效应传播模型

图1提出的算法以建立数字电路的故障效应传播为模型。该算法首先枚举门级网表所有基本逻辑门的类型,根据类型分别生成对应的模型,由此构建包含所有基本逻辑门的基本模型库;然后使用逻辑综合工具(如 Design Compiler)将包含复杂逻辑函数的电路综合为门级网表。图1中将 MUX2 综合成3个基本逻辑门的组合,再将其中的基本逻辑门映射至基本模型库,生成 MUX2 的故障传播模型,并

集成至复杂故障效应传播模型库中,以构建一个可扩展、可复用的大电路故障效应传播模型库。

对于完整的数字电路而言,使用相同的步骤,通过逻辑综合工具生成门级网表,将基本逻辑门映射至大电路故障效应传播模型库,以生成完整数字电路的故障效应传播模型。该模型同样可以扩展至大电路故障效应传播模型库中进行复用,以增加映射效率。

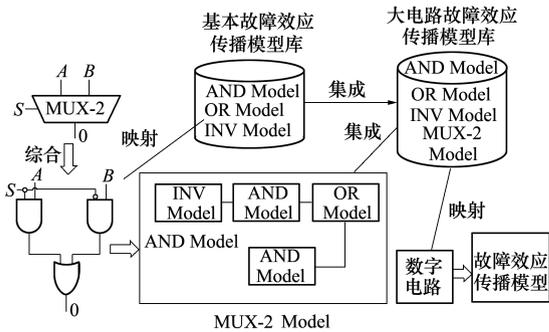


图 1 数字电路故障效应传播模型

## 2 硬件安全性与可靠性验证方法

### 2.1 安全属性

实现数字电路安全性与可靠性验证的首要步骤是分析电路需要满足的安全属性。安全属性的描述通常情况下借鉴 Property Specification Language 等功能属性描述语言的语义,主要包含安全级别的设置、故障效应传播模型精度的选择以及属性断言等。一般使用关键字 set 对特定信号进行安全级别的设置,同时使用关键字 assert 对需要验证的信号进行断言。

本文所描述的安全属性包括安全性与可靠性两方面。安全性属性包括机密性、完整性、隔离特性等,高(机密或不受信任)信息不应流向低(非机密或受信任)安全域。例如在处理器核电路中进行完整性验证,将不受信任的硬件信号 USB 设置为高安全级别,同时断言其不会覆盖程序计数器信号 PC,描述如下:

```
set USB=high
assert PC=low
```

可靠性属性主要描述在 X 传播问题中的确定性属性,指出不确定性信息不应流向确定性域。例如未初始化的信号 G 可能是不确定的 X 状态,将其设置为高安全级别,同时断言其不应流入内存 MEM

中,描述为:

```
set G=high
assert MEM=low
```

### 2.2 安全性与可靠性验证

Design Compiler 是一款电路综合的核心工具,用于将 HDL 描述的 RTL 级电路转换成门级网表。Questa Formal 是一款基于断言的形式化验证工具,内置 SAT (boolean satisfiability problem) 求解器,用于验证是否存在违背安全约束的电路设计。

图 2 描述了硬件安全性与可靠性验证方案。首先使用 Design Compile 综合工具将数字电路的 RTL 级描述生成门级网表,然后根据 1.2 节所描述的方法,将基本逻辑单元映射至基本故障效应传播模型库,最终生成数字电路的故障效应传播模型。

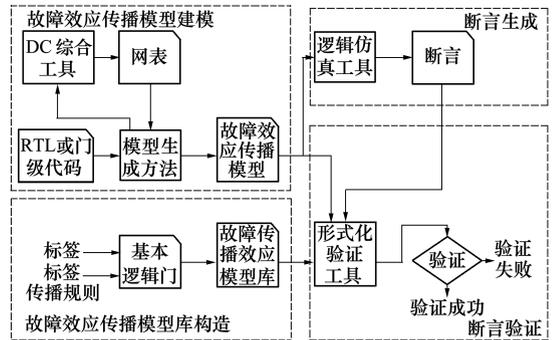


图 2 硬件安全性与可靠性验证方案

数字电路故障效应传播模型是安全性和可靠性验证的基础。不同的数字电路具有不同的安全性和可靠性需求,例如密码算法电路注重机密性属性,而摘要签名电路注重完整性属性。通过对数字电路功能的分析,建立不同的安全属性库,然后对数字电路中的信号进行安全等级划分,其次对相关信号的故障属性标签进行设置,指定其具有的安全属性,同时采用断言语句规定安全属性是否允许传播。例如在验证未初始化寄存器对系统可靠性影响时,需要将非确定性信号的标签标记为高安全级别 high,并禁止其传播至低安全级别 low 的输出中。假设寄存器 G 未初始化,将 G 的安全属性级别设置为 high,其他信号的安全属性级别设置为 low,系统可靠性要求断言输出信号的标签始终为 low,具体描述为:

```
set G_t high
set default_value_t low
assert output_t low
```

调用 Questa Formal 进行安全验证,分析数字电

路故障传播模型中安全属性,观察故障属性标签值。根据 Questa Formal 形式化验证的结果判断是否存在电路故障或违背设计安全要求的电路设计,若验证成功则符合电路设计规范,若验证失败则需要根据形式化验证给出的反例,对故障传播模型进行功能测试,跟踪故障传播路径,回溯故障源。

### 3 实验与分析

#### 3.1 实验环境与方案

实验使用英特尔 i9 处理器平台,16 GB 内存,在虚拟机中运行 64 位 Ubuntu 20.04 操作系统。实验流程如图 3 所示,数字电路经过综合、模型库映射和安全属性添加生成信息流模型,使用 SAT 求解器对属性断言进行计算,再通过仿真对求解结果进行验证、回溯故障源头。实验验证的对象是 Trust-Hub 和 OpenCores 上的标准测试基准。

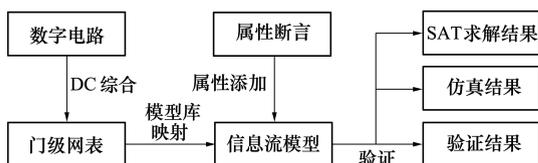


图 3 实验流程

#### 3.2 时间侧信道检测

本实验采用 OpenCores 中的一个 RSA 密码算法核,该密码核被验证具有时间侧信道漏洞,可以通过对加密时间的统计分析来恢复私钥。

密码算法核的硬件结构如图 4 所示。

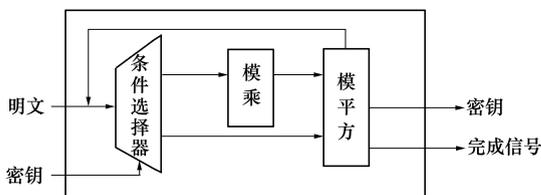


图 4 RSA 密码算法核硬件结构

首先按照 3.1 节中描述的流程生成故障传播模型。

对生成的故障传播模型进行时间侧信道检测,需要检查的一个安全属性为:信息未从密钥流向完成信号。如果该属性无法保持,则存在计时通道。设置密钥标签全为 1,以表明其具有机密性,其他信号标签全为 0;设置断言完成信号标签为 0,表明密

钥信息不应传播至完成信号。具体描述为:

```
set inExp_t=32'hFFFFFFF
set default_label=0
assert ready_t=0
```

使用形式化验证工具 Questa Formal 对断言进行验证,显示该断言无法通过验证,任何情况下完成信号的标签都将在加密完成时置 1。

图 5 是硬件安全仿真波形图,其结果表明:当密钥位标签标记为全高时,加密完成后,ready 信号的标签总是由 0 变成 1,密钥的故障标签传播到了完成信号(ready)。这是由于密钥位的不同导致模乘运算消耗的时间不同,ready 信号由 0 置 1 的时间也发生了变化,说明该数字电路存在时间侧信道。

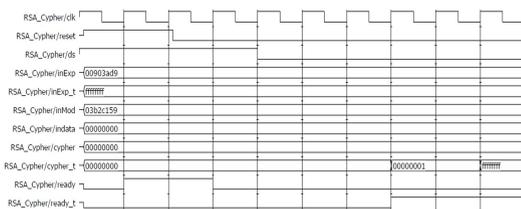


图 5 时间侧信道检测的反例波形

#### 3.3 X-传播验证

本实验采用 Trust-Hub 上的标准测试基准 BasicRSA-T100 进行 X-传播验证,在验证的过程中,将所有主要输入的故障标签全部标记为 0,即确定性,并检查输入是否可以传播至非确定性状态。按照 3.1 节中的流程使用 Synopsys Design Compiler 工具对 BasicRSA-T100 基准进行逻辑综合,生成门级网表并映射至传播模型库中。然后通过实例化门级网表中的标准单元生成故障传播模型。

对生成的故障传播模型进行 X-传播验证,为相应的 X 源和观测点创建合适的约束和断言,然后调用形式化验证工具查看是否存在从 X 源至观测点的传播路径,具体描述为:

```
assume $ all_input_label=0
assume $ default_init_value=1
assert cypher_t==32'h0
assert @rose(ready)cypher_t==32'h0
```

使用形式化验证工具 Questa Formal 对断言进行验证。验证通过了第二个属性,表明加密完成时,密文处于确定性状态,但是第一个属性验证失败了,并返回一个反例。使用 Mentor Graphics QuestaSim 对返回的反例进行波形回放,图 6 显示了 X-传播验

证反例的回放波形,在进行复位后, cypher\_t 为 32'hFFFFFFF, cypher 为 32'hFFFFFFF,而加密完成后, cypher\_t 变为 32'h0, cypher 同样变为 32'h0,说明在复位时 X 源能够传播至 cypher 信号端口,这是由于复位引脚在 BasicRSA-T100 测试基准中是浮动的。

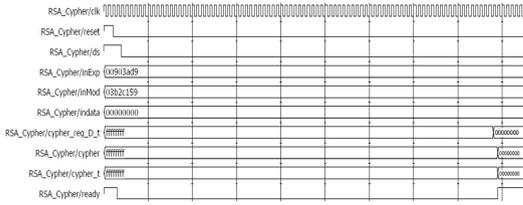


图 6 X-传播验证的反例波形

### 3.4 硬件木马检测

本实验同样选用 Trust-Hub 上的 BasicRSA-T100 硬件木马测试基准作为实验对象,该测试基准存在硬件木马,特定的明文将激活木马电路,将密钥输出至密文端口。

BasicRSA-T100 硬件结构如图 7 所示。

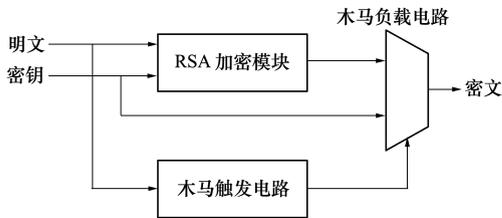


图 7 BasicRSA-T100 硬件结构图

首先按照 3.1 节中的流程生成故障传播模型。

为验证该密码算法核中存在硬件木马导致信息泄露,需要检查的属性是没有密钥未直接流向输出端口。由于密钥存在扩散特性,每一位密钥都将流向多个密文位,如果将密钥标签全设置为 1,密文标签必然也全为 1,无法判断是正确加密引起的传播还是存在木马电路引起的传播。故设置第二位密文位标签(inExp\_t[1])为 0,其他密文位标签为 1。如

若存在硬件木马泄露密钥值,则第二密文标签(cypher\_t[1])不会被密码核扩散传播,将与第二密文位标签一致,保持为 0。因此,向故障传播模型中加入断言:当加密完成后,第二密文位标签应该保持为 0。具体描述为:

```
assume inExp_t=32'hFFFFFFFD
assume $ default_label=0
assert $ rose(ready) cypher_t[1] == 1
```

使用形式化验证工具 Questa Formal 进行验证,返回了一个反例,此时 indata = 32'h44444444。使用 Mentor Graphics QuestaSim 对返回的反例进行波形回放,图 8 显示了反例的回放波形。验证结果表明该密码核存在硬件木马电路泄露密钥信息,与基准电路设计一致。

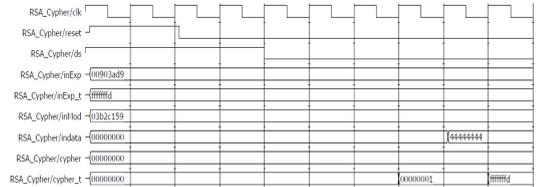


图 8 硬件木马验证的反例波形

## 4 结论

本文提出了利用故障传播模型的方法对数字电路进行安全与可靠性验证。通过对数字电路进行建模,为每个信号添加故障标签生成故障传播模型,利用形式化验证工具 Questa Formal 对模型进行安全与可靠性验证,从而检测是否存在设计缺陷、侧信道或硬件木马。使用仿真工具 Mentor Graphics QuestaSim 对模型进行仿真,更加直观地跟踪故障传播路径,回溯故障源。对于更大型的电路设计,需要解决安全属性自动化书写的问题。另一方面,对于大型电路设计,故障传播模型将扩大其电路规模,形式化验证方法在工程实践中存在状态空间爆炸、运行效率低等问题。本课题将进一步研究缩小模型规模以及细粒度建模的方法。

### 参考文献:

[1] HORN J, HAAS W, PRESCHER T, et al. Meltdown: reading kernel memory from user space[C]//27th Security Symposium Security, 2018

[2] WEISSE O, NEAL I, LOUGHLIN K, et al. NDA: preventing speculative execution attacks at their source[C]//Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, 2019

- [3] BHUNIA S, HSIAO M S, BANGA M, et al. Hardware trojan attacks; threat analysis and countermeasures[J]. Proceedings of the IEEE, 2014, 102(8): 1229-1247
- [4] SHAKYA B, HE T, SALMANI H, et al. Benchmarking of hardware trojans and maliciously affected circuits[J]. Journal of Hardware and Systems Security, 2017, 1: 85-102
- [5] MAO B, HU W, ALTHOFF A, et al. Quantitative analysis of timing channel security in cryptographic hardware design[J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(9): 1719-1732
- [6] KAUSHIK P, MAJUMDAR R. Timing attack analysis on AES on modern processors[C]//6th International Conference on Reliability, Infocom Technologies and Optimization, 2017
- [7] DINA G M, HU W, MIRJANA S. X-attack: remote activation of satisfiability don't-care hardware Trojans on shared FPGAs[C]//30th International Conference on Field-Programmable Logic and Applications, 2020
- [8] GAO Y, LIU L, DU H, et al. Software and hardware co-verification technology based on virtual prototyping of RF SOC[C]//IEEE International Conference on Computer and Communication Engineering Technology, 2018
- [9] FERN N, SAN I, CHENG K. Detecting hardware trojans in unspecified functionality through solving satisfiability problems[C]//2nd Asia and South Pacific Design Automation Conference, 2017
- [10] TIWARI M, WASSEL H, MAZLOOM B, et al. Complete information flow tracking from the gates up[C]//Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems, 2009
- [11] ARDESHIRICHAM A, WEI H, MARXEN J, et al. Register transfer level information flow tracking for provably secure hardware design[C]//Design, Automation & Test in Europe Conference & Exhibition, 2017

## Hardware security and reliability verification based on fault propagation model

ZHANG Xige<sup>1</sup>, ZHU Jiacheng<sup>2</sup>, MA Jun<sup>1</sup>, SHEN Lixiang<sup>2</sup>, ZHOU Jiahui<sup>1</sup>, MU Dejun<sup>2</sup>

(1.Beijing Smart-Chip Microelectronics Technology Co.,Ltd., Beijing 100000, China;  
2.Shenzhen Research Institute of Northwestern Polytechnical University, Shenzhen 518057, China)

**Abstract:** Large scale integrate circuits is facing serious threat such as design vulnerabilities, side channels, and hardware Trojans. Traditional functional verification method is difficult to ensure high test coverage, and it is also difficult to detect security vulnerabilities such as side channels and stealthy hardware Trojans. Formal verification methods focus on the equivalence and functional correctness of design, and are difficult to meet security and reliability verification needs. The present work proposes a hardware security and reliability verification method from formal model. The present method can develop formal models for describing the security and reliability behaviour of hardware designs. It can detect potential security vulnerabilities in hardware designs. Experimental results show that the verification method is effective in detecting sensitive information leakage and modification caused by side channels and hardware Trojans.

**Keywords:** formal model; fault effect analysis; vulnerability detection; hardware security

**引用格式:**张茜歌,朱嘉诚,马俊,等.基于故障传播模型的硬件安全性与可靠性验证方法[J].西北工业大学学报,2024,42(1):92-97

ZHANG Xige, ZHU Jiacheng, MA Jun, et al. Hardware security and reliability verification based on fault propagation model[J]. Journal of Northwestern Polytechnical University, 2024, 42(1): 92-97 (in Chinese)