**安全芯片EDA设计工具**

**（使用说明书）**

**目 录**

[安全芯片EDA文档 1](#_Toc20369)

[1. EDA介绍 1](#_Toc13334)

[1.1. EDA历史发展 1](#_Toc2843)

[1.1.1. EDA世界历史发展 1](#_Toc24513)

[1.1.2. EDA中国历史发展 4](#_Toc24157)

[1.2. EDA相关公司介绍 6](#_Toc4053)

[1.3. EDA相关技术介绍 8](#_Toc19128)

[2. 故障注入 12](#_Toc26267)

[2.1. 故障注入介绍 12](#_Toc15868)

[2.1.1. 软件界面与功能介绍 13](#_Toc17578)

[2.1.2. 故障注入技术介绍 2](#_Toc8288)4

[2.2. 故障注入技术 26](#_Toc15213)

[2.2.1. 电源故障注入 26](#_Toc10090)

[2.2.2. 电磁故障注入 2](#_Toc10090)8

[2.2.3. 电压故障注入 2](#_Toc10090)9

[2.2.4. 时钟故障注入 2](#_Toc10090)9

[2.2.5. 光故障注入 3](#_Toc10090)0

[2.2.6. 温度故障注入 3](#_Toc10090)0

[2.2.7. 重离子束故障注入 3](#_Toc10090)1

[2.2.8. 管脚故障注入 3](#_Toc10090)1

[2.2.9. 开短路故障注入 3](#_Toc10090)2

[2.2.10. 信号线串行阻抗和信号线并行阻抗故障注入 3](#_Toc10090)2

[2.2.11. 外部信号替换叠加及噪音耦合故障注入 3](#_Toc5364)3

[2.2.12. 固定型故障注入 3](#_Toc15622)4

[2.2.13. 杂散电流故障注入 3](#_Toc15622)4

[2.2.14. 电涌故障注入 3](#_Toc15622)5

[2.2.15. 存储数据损坏故障注入 3](#_Toc15622)5

[2.2.16. 软件缺陷故障注入 3](#_Toc15622)5

[2.2.17. 激光故障注入 3](#_Toc15622)5

[2.3. 故障注入应用 36](#_Toc9004)

[2.3.1. 在装备测试性验证中的应用 36](#_Toc1112)

[2.3.2. 在复杂航电系统中的应用 38](#_Toc23023)

[2.4. 故障注入技术的发展前景 38](#_Toc4616)

[3. 侧信道泄露分析 40](#_Toc4755)

[3.1. 侧信道泄露分析介绍 40](#_Toc23586)

[3.1.1. 软件界面与功能介绍 43](#_Toc3288)

[3.1.2. 侧信道泄露的危害 51](#_Toc17852)

[3.1.3. 侧信道泄露分析的作用 52](#_Toc8297)

[3.2. 侧信道分析方法 53](#_Toc3329)

[3.2.1. DPA差分功耗分析 54](#_Toc27473)

[3.2.2. TA模板分析 55](#_Toc22092)

[3.2.3. CPA相关性功耗分析 55](#_Toc27910)

[3.2.4. SPA简单功耗分析 56](#_Toc9602)

[3.3. 泄露分析工具 5](#_Toc15110)6

[3.3.1. 芯片功耗/电磁泄露模型 5](#_Toc24261)6

[3.3.2. 测信道泄露仿真引擎 6](#_Toc8980)5

[4. 算法门限安全检测 68](#_Toc12906)

[4.1. 算法门限安全介绍 68](#_Toc30099)

[4.1.1. 软件界面和功能介绍 68](#_Toc5035)

[4.1.2. 算法门限安全的定义 74](#_Toc15771)

[5. 版图安全检测 77](#_Toc16974)

[5.1. 版图安全检测介绍 77](#_Toc16159)

[5.1.1. 软件界面和功能介绍 77](#_Toc12444)

[5.1.2. 版图是什么 81](#_Toc19012)

[6. 硬件木马分析 83](#_Toc6245)

[6.1. 硬件木马介绍 83](#_Toc7621)

[6.1.1. 软件界面和功能介绍 84](#_Toc14667)

[6.1.2. 什么是硬件木马 89](#_Toc27701)

[6.1.3. 历史上硬件木马造成的数据与经济损失 93](#_Toc27065)

[6.2. 硬件木马分类 95](#_Toc31316)

[6.2.1. 硬件木马的工作原理 103](#_Toc2360)

[6.3. 硬件木马检测技术 105](#_Toc12361)

[6.3.1. 物理检测技术 1](#_Toc2360)06

[6.3.2. 逻辑检测技术 1](#_Toc2360)07

[6.3.3. 测信道分析技术 1](#_Toc2360)08

[6.4. 硬件木马对抗的未来发展与展望 112](#_Toc31218)

# 安全芯片EDA文档

## EDA介绍

#### EDA历史发展

随着电子技术的快速发展，信息化产品的更新与迭代速度也逐步加快，车联网、芯片等各行业对新型电子产品的设计提出了更轻量化、集成化需求。然而，电子技术的发展离不开 EDA（Electronic Design Automation，电子设计自动化）软件的帮助，这类软件是一整套设计电子系统的辅助工具，为电子产品设计系统集成开发提供了从初始设计、电子调试到迭代更新等工具支持，从而推动电子产品小型化、快速化发展，设计系统也由组合芯片转向单片系统。

#### EDA世界历史发展

随着智能化应用在制造业领域的深入发展，EDA软件的诞生为电子工程设计行业带来了新的视角，明确了电子工程设计的新方向，并且在电子信息、通信、航空航天等领域的发展中扮演着关键角色。EDA 软件的发展主要包括四个阶段。第一个时期是20世纪70年代CAD时代，设计人员主要通过手动操作完成电路图的设计和布局布线；第二阶段到80年代，EDA逐步走向商业化，诞生了电路设计的基础超高速集成电路硬件描述语言（Very-High-Speed Integrated Circuit Hardware Description Language，VHDL）等硬件描述语；到第三阶段90年代，EDA技术正式成为了一种高级语言描述、系统级仿真和综合技术；从21世纪到现在，EDA的发展速度很快，硬件描述语言趋于更加高效。随着EDA技术的发展和芯片需求量的逐步递增，EDA软件层出不穷，提高了EDA行业市场的规模。目前从全世界市场占有率来看，EDA软件的供应商主要是Synopsys、Cadence和Mentor三家企业，其总的市场占有率高达60%。市面上具有广泛影响的EDA软件有Altium Designer、Pads、Cadence、Mento、EWB、Zuken等。这些软件的功能性很强大，例如大部分都可以进行电路设计与仿真，同时可以进行PCB（Printed Circuit Board，印制电路板）自动布局布线，并且可以输出多种网表文件与第三方软件对接。每个EDA软件虽然都有自身强大的功能特点，但同时也存在一些缺陷，具体分析如下表所示。

|  |  |  |
| --- | --- | --- |
| 表1.1 EDA软件优缺点分析 | | |
| EDA 软件 | 优势 | 缺陷 |
| Altium Designer | 1. 市场的普及率较高。很多高校将此工具作为基础课程学习，操作简单方便；   2、原理图设计、电路仿真等技术融合性较好；  3、电路设计的质量和效率较高；  4、支持应用程序接口（Application Program Interface，API）接口开发，利于二次开发。 | 占用系统资源较多，对电脑配置要求较高。 |
| Pads | 1、市场的普及率较高；  2、布局布线功能强大。 | 功能覆盖面较薄弱，做高速板时需要使用仿真软件。 |
| Cadence | 1、支持跨集成电路、封装和PCB协同设计高性能互连；  2、Cadence自动布线工具布通率较高；  3、PCB Layout工具能完成信号完整性仿真和电源完整性仿真等；  4、支持API接口开放，利于二次开发。 | 入门比较难，操作不太灵活。 |
| Mentor | 1、在整个电子产品设计过程中具备多个基本功能模块，包括图纸创建、版图设计仿真以及产品制造的诸多流程。  2、元器件封装以及库管理较完整 | 国内其支持商相对较少，普及率较低，相关参考资料较少。 |
| EWB | 1、占用资源较少；  2、混合仿真功能强大。 | 自动排线效果较差，仿真过程及结果缺乏真实性。 |
| Zuken | 1、解决方案较全面；  2、灵活性较强。 | 在国内市场占有率和相关参考资料较少。 |

#### EDA中国历史发展

国务院总理李克强在2015年全国两会上作《政府工作报告》时首次提出“中国制造2025”的宏大计划。此次计划作为我国制造强国战略的第一个十年行动纲领其重点要大力发展“新一代信息技术产业”、“高档数控机床和机器人”和“电力装备”三大电子技术领域。电子技术作为我国未来发展的关键驱动力，迫切需要高校、产业界投入资源、技术，提升我国整体电子工程设计水平，加强电子技术的研究和应用，使电子技术能够更好、更快地适应我国科技发展水平与人们日益增长的物质文化需求。电子设计自动化（Electronic Design Automation，EDA）作为当前电子技术领域的核心设计技术，对芯片、电路等电子产品与系统的发展与演进有着至关重要的支撑作用，是支撑我国科技进步与弯道超车的关键基础设施。随着EDA技术的发展衍生出了电子设计软件，即EDA软件。通过EDA软件实现集成电路芯片的功能设计、模拟验证、图纸设计等重要流程，进而增强我国电子芯片等高精尖产品的自主设计能力、加快设计自动化程度，从而更有效的完成电子电路设计工作。

电子产业是信息技术产业的基石，是国家的基础性、战略性和先锋性产业。当今世界，电子行业正在快速发展，国际上领先的电子企业不断加快先进技术和工艺的研发，然而，由于我国的电子行业起步较迟，在集成电路设计领域企业的自主创新能力比较薄弱，与世界先进水平还有很大的距离。根据智研咨询发布的《2020-2026年中国电子设计自动化（EDA）产业运营现状及发展前景分析报告》结果分析：我国集成电路产业竞争优势不明显，尤其在上游设计领域缺乏竞争力，导致国内EDA软件市场发展缓慢。因此EDA软件仍被视为是我国集成电路产业发展过程当中的一个“卡脖子”环节。

在2020年5月15日，美国商务部对华为公司实施禁令，全面限制华为使用美国EDA软件和半导体材料。美国对中国实施的经济制裁，使我们清楚地看到了国内的EDA软件与外国的差别，同时也证明了EDA软件处于芯片产业链的上游，几乎涉及IC设计的各个方面。一旦国外限制中国使用EDA软件，中国的芯片发展将遭受巨大冲击。

由于我国EDA起步较晚，很难突破多年的技术积累瓶颈和市场占有率，并且研究一款与国外EDA软件功能相似的软件希望渺小，不仅面临EDA软件产业链长，复杂度高，物理尺度小的挑战，而且需要芯片设计公司、晶圆生产厂商等企业共同合作开发，以至于达到一个满足用户设计需求的产品水平。在严峻的EDA软件发展环境中，为了提高电子产品的设计效能和质量，给予研发国产EDA软件的企业启发和思考，使EDA软件尽早实现国产化。

### EDA相关公司介绍

随着电子产业规模化，以及国家对芯片设计的支撑技术，尤其是以EDA软件为代表的芯片设计基础平台的重视，涌现了一批国产化EDA软件的公司，如下表所示。我国已拥有华大九天、芯愿景、广立微电子、芯禾科技、概伦电子、珂晶达、创联智软等企业，通过这些企业在克服“技术壁垒”、“人才缺失”等难题后，在全国范围内逐步实现电子设计企业由“点”到“面”的全面突破。

表1.2 中国EDA公司概括

|  |  |  |
| --- | --- | --- |
| 公司名称 | 主要产品 | 布局领域 |
| 华大九天 | Standard Cell/IP 设计、仿真、验证； | IC 设计、IP 产品、平板显示电路设计。 |
| 芯愿景 | 显微图像采集和处理系统；  集成电路分析再设计系统；  集成电路分析验证系统；  集成电路设计优化系统。 | 集成电路分析、集成电路设计及 EDA 软件授权。 |
| 广立微电子 | 参数化单元创建工具；  测试芯片设计平台；  可寻址测试芯片设计平台；  WAT（Wafer Acceptance Test，晶圆接受测试）和测试芯片数据的分析工具。 | 高效测试芯片自动设计、高速电学测试和智能数据分析的全流程平台。 |
| 芯禾科技 | 高速仿真解决方案SnpExpert；  Xpeedic标准IPD元件库。 | 设计仿真工具，集成无源器件。 |

在上表中华大九天是目前国内规模最大、技术实力最强的 EDA 企业。华大九天具有全定制的 IC 设计平台和物理验证工具，IC 设计平台涵盖设计数据库管理、原理图 编辑器和版图编辑器，物理验证工具包括DRC（Design Rule Check，层次设计规则检测）和LVS（Layout Versus Schematic，版图与原理图一致性比较），这两者可以帮助版图设计工程师检查版图和电路原理图的结构是否一致，从而快速定位版图设计错误，加速验证并缩短产品设计周期。另外，概论电子正在向实现设计工艺协同优化的目标发展，努力完成从数据到仿真的创新型 EDA 软件。在 EDA 软件发展的储备力量中不仅仅只有企业，在高校也有进行 EDA 软件研发的团队，像是清华大学计算机系的 EDA 研究室和复旦大学集成电路国家重点实验室等。

### EDA相关技术介绍

EDA 技术与集成电路相辅相成，先进的 EDA 技术可以设计大规模的集成电路，而大规模的集成电路则又支持更先进的 EDA 技术开发，两者在数字化和智能化的现代电子系统开发中起着支撑作用。

对于较大规模的设计，一般采用层次化和结构化的设计方法。这种设计方法可以很好地维持设计边界和层次，符合人的思维方式并利于复用代码。但是 EDA 逻辑综合工具在把 RTL 描述转换为门级网表的时候，会以每一个模块的边界作为默认的逻辑优化边界，模块划分过小或者层次过多的时候，不能获得最优的电路面积结果。若在综合命令中加入“-flat”选项，综合工具会将所选择的层次结构打平后再进行优化，在获得较优的面积结果的同时，输出网表的可读性较差，会给调试工作带来不便。

硬件描述语言（HDL）是一种用形式化方法来描述数字电路和设计数字逻辑系统的语言。由于绝大多数的数字集成电路设计以及FPGA设计都以VerilogHDL的代码为载体，Verilog HDL 语法，在层次化结构的展平处理中，主要完成以下内容：用模块内的设计代码替换实例化语句，主要是模块内代码信号的唯一化和输入输出的替换操作；对于带有 parameter 的参数化设计，将实例化时的参数变成没有参数的设计；对于带有 generate 的可配置化设计，实例化时从可配置的代码中选择代码进行提取；对于带有 define 和ifdef 的宏定义参数，判断决定代码保留。经层次结构打平后，代码使用扁平的链式结构。

“模块”是 Verilog 的基本设计单元，在一个模块中引用另一个模块，对其端口进行关联，叫作模块的例化。在一个顶层设计中实例化子模块电路，类似于编程中在主函数调用子函数。模块的例化建立了描述的层次，一个复杂的电路往往通过模块嵌套形成了复杂的层次结构。而去层次化则类似于编程中将用调用函数实现的代码转换为不用函数，而是直接在主程序中实现各个函数的内部代码。经过去层次化后，设计代码变为打平的、没有层次的、不带参数、不带可选择配置的代码。在 Python 中，re 模块就是一个用于处理正则表达式的模块。鉴于 Python 强大的文本处理能力，本文采用Python和正则表达式对Verilog HDL代码的文本进行处理。

正则表达式被提出时，主要用于描述正则集代数。它是提供计算机操作和检验所要抽取的字符串数据的一种强大的工具，是一串由特定意义的字符组成的字符串，提供了一种从字符集合中搜寻特定字符串的规则。正则表达式可以运用于多种操作系统中，目前很多计算机语言都支持正则表达式匹配，如 C++、PHP、Java 等。

正则表达式最基本的3种功能包括匹配、替换和提取。匹配功能用于判断文本里面是否有符合要求的字符，将不同的字符按照匹配内容的不同进行分组，如在此项目实践中将宏定义的标识符和内容根据“`”符号区分。替换功能用于在文档中使用匹配模式来标识特定文字，然后进行删除或进行替换，如删除注释、端口名和参数声明名称的替换、generate语句中带有$符号的标识符和begin/end语句的替换。提取功能用于从字符串中提取子字符串，如获取需要的变量名称、模块名称、判断语句、模块内容等。Verilog 继承了 C 语言的多种操作符和结构，其本身就是一系列字符串组成的文本，在信息抽取的过程中完全适应正则表达式的特点。

正则表达式由普通字符和特殊字符组成，在正则表达式中，可以用方括号括起若干个字符来表示一个元字符（特殊字符）。同时正则表达式支持限定符的概念。限定符指定输入中必须存在字符、组或字符类的多少实例才能找到匹配项，这些限定符可以指定正则表达式的一个给定组间需要出现多少词才能满足匹配，因而可以适应一些不确定情。正则表达式常见的元字符和限定符如下表所示。

|  |  |  |
| --- | --- | --- |
| 表格1.3 正则表达式常见的元字符和限定符 | | |
| 名称 | **字符** | **描述** |
| 元 字 符 | ^ | 匹配输入字符串的开始位置 |
| $ | 匹配输入字符串的结束位置 |
| \* | 匹配前面的子表达式零次或多次 |
| {n} | n 是一个非负整数。匹配确定的 n 次 |
| (pattern) | 匹配 pattern 并获取这一匹配 |
| [xyz] | 字符集合。匹配所包含的任意一个字符 |
| [a-z] | 字符范围。匹配指定范围内的任意字符 |
| \num | 匹配 num，其中 num 是一个正整数 |
| \un | 匹配 n，其中 n 是用四个十六进制数字表示的 Unicode |
| 修 饰 符 | i | 将匹配设置为不区分大小写 |
| g | 查找所有的匹配项 |
| m | 使边界字符 ^ 和 $ 匹配每一行的开头和结尾 |
| s | 使字符 . 匹配换行符 \n |

## 故障注入

### 故障注入介绍

故障注入起源于上个世纪70年代，是一种[可靠性](https://baike.baidu.com/item/%E5%8F%AF%E9%9D%A0%E6%80%A7/512935?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E6%95%85%E9%9A%9C%E6%B3%A8%E5%85%A5/_blank)验证技术，通过受控实验向系统中刻意引入故障，并观察系统中存在[故障](https://baike.baidu.com/item/%E6%95%85%E9%9A%9C/9850042?fromModule=lemma_inlink)验证技术，通过受控实验向系统中刻意引入故障，并观察系统中存在时的行为。目前国外复杂电子系统的设计及测试已经广泛地使用故障注入。

故障模型是集成电路测试领域中的一个重要概念，是一类具有相同属性的故障的集合。在集成电路测试领域中存在两个基本概念：缺陷和故障。缺陷表示集成电路物理层上由于材料、工艺、封装以及设备等原因出现与原先设计意愿相违背的问题。故障表示集成电路由于缺陷存在引起电路逻辑结构或功能变化。集成电路制造过程中，芯片需要经历一系列氧化、扩散、光刻、刻蚀、积淀、煺火等工艺环节。每道工艺都可能在芯片内部引入金属互连线断路、寄生晶体管、氧化层崩溃等缺陷。这些缺陷可能导致电路逻辑结构或逻辑值变化使得电路中出现故障。当故障表现出来时，即对电路产生作用，可能引起电路特定功能无法正常工作，例如加密电路无法正常执行加密功能。由于缺陷种类多、原因复杂、具有隐蔽性等特点，无法直接模拟缺陷对电路进行可靠性测试或查找电路失效原因。提取缺陷形成的故障，建立故障模型，可以有效减小集成测试模拟空间。

#### 软件界面与功能介绍

故障注入是一种通过电磁、激光等设备或手段在电路中注入故障以干扰电路的正常运行迫使电路泄露关键信息的技术。故障注入界面启动界面如下图所示：

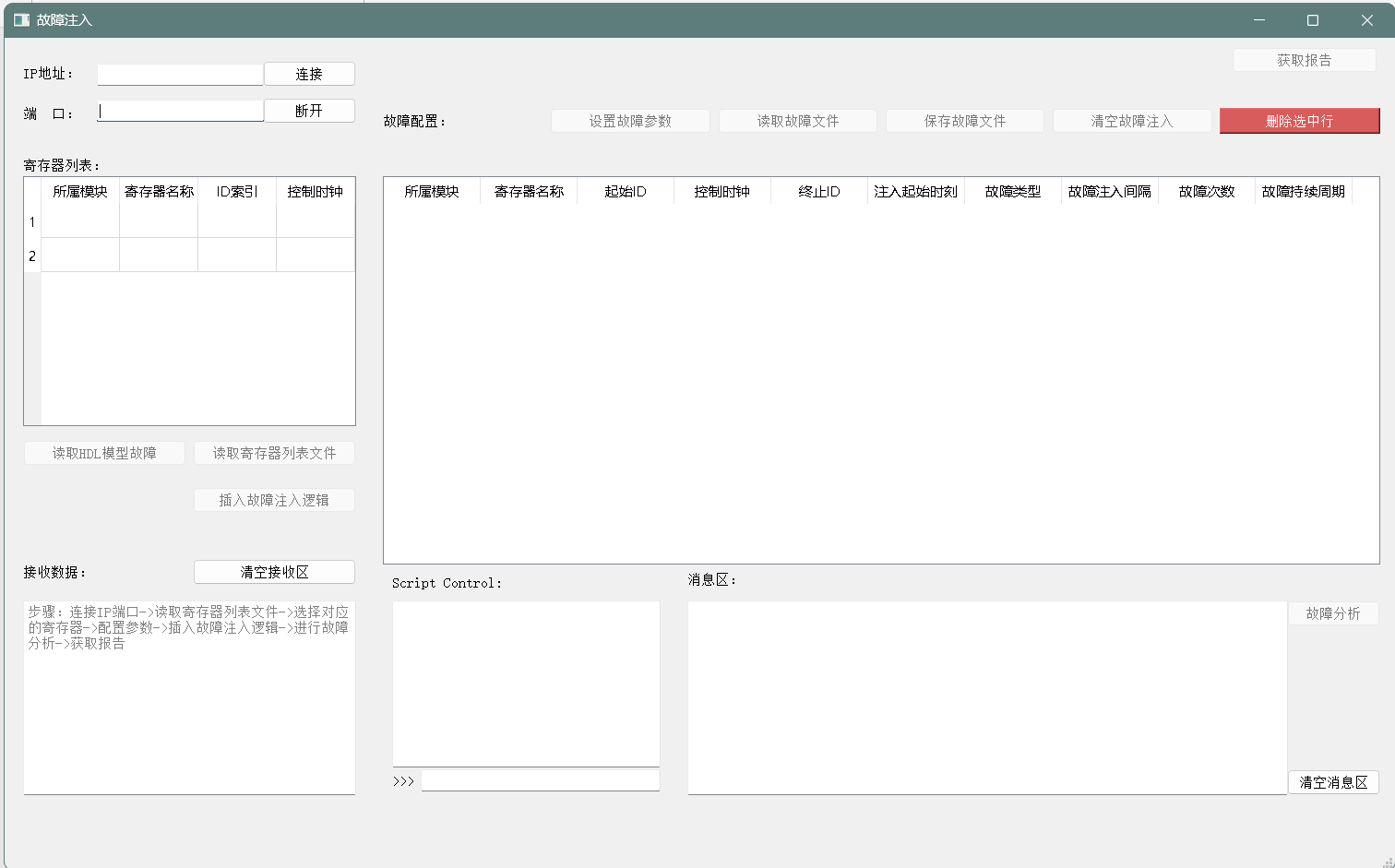


图2.1 故障注入启动界面

故障注入界面分为：登录模块、寄存器列表、数据接收区、故障配置模块、故障数据列表、脚本控制模块、故障注入结果消息区模块。

登录模块：用于芯片的连接，通过ip地址和端口的方式，实现了芯片的无线有线双连接，且允许在多个芯片中切换，使得多芯片的故障注入实现变得更加方便。对于每个芯片，都可以给其分配特定的ip地址和端口号，用户需要输入芯片对应的ip地址和端口号即可连接到相关芯片，如下图所示：

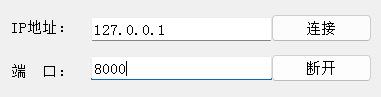


图2.2 登录模块

当完成ip地址和端口号的输入后，点击“连接”按钮，即可进行芯片连接，接收区会给出是否连接成功的提示，如下图：

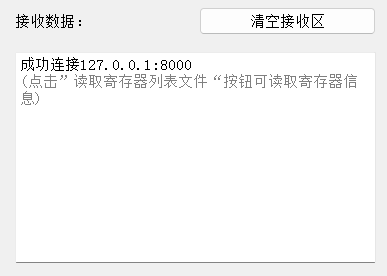


图2.3 登录成功

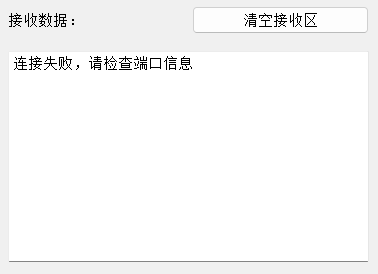


图2.4 登录失败

同时，系统会自动判断ip地址与端口号的格式是否正确，当发现ip地址格式错误时，或者端口号超出合理范围时，会给出弹窗提示，以提醒用户，检查输入的ip地址和端口号以保证输入的合法性，弹窗如下图：



图2.5 ip地址、端口格式错误弹窗

在成功连接后，部分按钮功能将会解锁，用户可以点击“读取HDL模型故障”、“读取寄存器列表文件”、“插入故障注入”三个按钮，以进行下一步操作。

脚本控制模块，可以通过直接输入命令，代替手动点击按钮，方便快速操作故障注入，以节约点击按钮的时间，如下图：

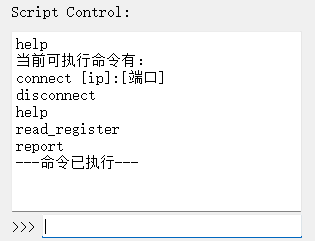


图2.6 脚本控制

对于“读取HDL模型故障”会显示对应的HDL模型故障，如下图所示：



图2.7 HDL模型故障

对于“读取寄存器列表文件”按钮，该按钮用于读取芯片的寄存器列表，然后在寄存器列表中展示出来。同时之前在成功连接芯片时，接收区也会对点击“读取寄存器列表文件”按钮发出文字提示，以指导用户进行故障注入操作，文字提示如下：

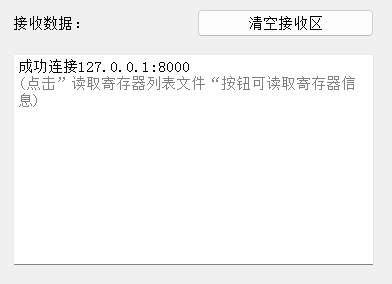


图2.8 接收区提示信息

寄存器列表：当点击“读取寄存器列表文件”按钮后，故障配置部分的按钮将会解锁，寄存器列表将会读取并显示该芯片的寄存器信息，同时接收区会提示下一步骤，如下图：



图2.9 寄存器列表信息

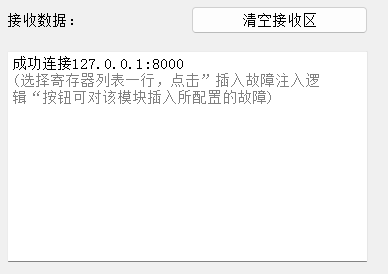


图2.10 接收区提示信息

根据接收区提示信息，选择寄存器列表需要注入故障的寄存器，之后点击“插入故障注入逻辑”按钮即可对该寄存器进行预设故障注入，如下图：



图2.11 寄存器选中一行

进行预设故障注入的寄存器，会连带着故障信息一起显示在故障数据列表中，故障数据列表显示的信息包括：所属模块、寄存器名称、起始ID、控制时钟、终止ID、注入起始时刻、故障类型、故障注入间隔、故障次数、故障持续周期。如下图所示：



图2.12 故障数据列表

故障配置按钮区，用于设置故障注入信息与保存之前插入的一系列故障，以方便之后一键引用，同时也能对错误的故障信息进行删除。

当点击设置故障参数时，会弹出一个故障配置页面，用于设置预设故障注入，当配置完后，再点击“插入故障注入逻辑”就会生成新的故障数据行，故障配置页面如下图：



图2.13 故障配置参数

可以设置的故障配置参数包括：ID范围、注入起始时刻、故障模型、故障注入间隔、故障次数、故障持续时间，配置好了参数后，点击保存即可将信息存储在系统中，等待下次点击“插入故障注入逻辑”就会被用在下一个寄存器中。如下图，新的一行与上一行的故障参数不一样：



图2.14 故障数据列表

对于“读取故障文件”按钮，点击后即可将之前保存的故障数据一起读入故障数据列表，同时消息区也会显示读取的故障条数，如下图：



图2.15 读取故障文件的故障数据列表

同样“保存故障文件”将当前故障数据列表保存入系统，以方便下次一键使用，同时消息区会显示保存信息，如下图：

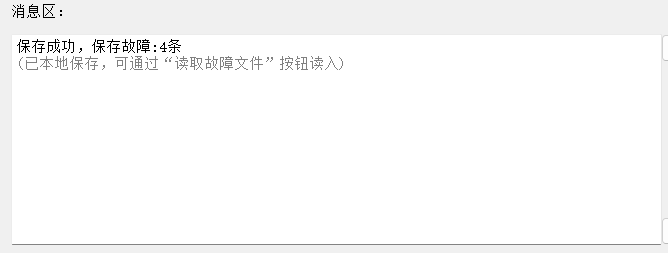


图2.16 保存故障文件后消息区

对于“清空故障注入”按钮，会将故障数据列表清空，“删除选中行”按钮会删除鼠标选中的一行故障数据。

当完成了所需的故障插入，接下来就可以点击“故障分析按钮”，对故障数据列表每行的故障指令进行执行，执行结果会显示在消息区，如下图：



图2.17 故障分析按钮后消息区

最后一步就是生成报告，在故障分析完成后，点击页面右上角的“获取报告”按钮即可弹出报告预览页面，如下图：



图2.18 故障注入报告预览

报告预览界面中，左上角“故障注入报告”表示该报告属于哪个模块，右上角有生成的唯一编码，以方便识别报告，除此之外还有报告类型与报告对应的芯片ip地址与端口信息，报告完成时间等基本信息，同时报告也会显示故障数据列表与故障分析结果，以方便用户对比查看，在右下角点击“生成word”即可导出word文件，如下图：

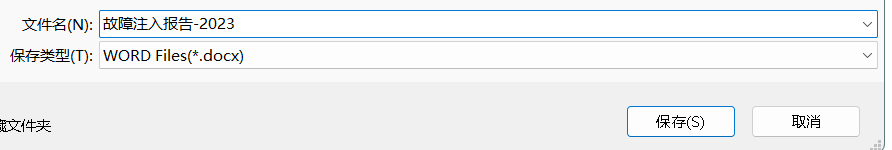


图2.19 保存word文件

当成功生成word报告后，会弹窗告诉用户生成成功，如下图：

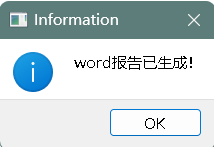


图2.20 word报告生成成功提示

#### 故障注入技术介绍

故障注入攻击通过电磁故障注入、激光故障注入等故障注入技术在芯片的薄弱区域中诱导电路故障，强迫芯片产生异常输出，然后借助于差分故障分析等故障分析技术对异常输出进行推理和数学分析，从而获取芯片中处理的密钥等关键信息。与由于材料、工艺、封装以及设备等原因在芯片中产生的缺陷类似，故障注入技术种类繁多，这些技术在引入故障的物理机理上存在着差异。通过提取各种故障注入技术造成的芯片故障，可以建立用于故障注入攻击的故障模型。利用简单的逻辑模型代替复杂的故障注入技术，实现故障注入攻击模拟。

用于故障注入攻击的故障模型提取了故障注入攻击造成的芯片故障。按故障类型分，有固定故障、翻转故障、随机故障等；按故障持续时间分，有瞬态故障、永久故障、周期性故障等；按故障强度分，有单比特故障、多比特故障等。固定故障表征了故障注入技术造成的芯片逻辑门层次上的线网逻辑值被固定为逻辑0或逻辑1。固定故障被广泛用于电路可靠性测试和故障注入攻击模拟。翻转故障表征寄存器或随机存储器等存储单元中存储的数值或线网上的逻辑值等，在受到故障注入攻击时变为相反的逻辑值。通常翻转故障多为瞬态翻转，不会造成芯片损坏，难以检测。这也导致翻转故障成为可靠性分析的重点研究对象。桥接故障则提取了集成电路逻辑门层次上线网短接的情形。由于特征尺寸持续缩小，芯片内部金属互连线距离极小，故障注入技术可能造成金属互连线之间短接使得两个原本输入相互独立的逻辑门的输入相连接，即形成了桥接故障。使用这些故障模型在模拟平台上模拟故障注入攻击，可以在芯片设计期间对芯片进行安全评估，查找设计漏洞，指导设计者提高芯片安全性。

故障注入技术一般分为：基于硬件的故障注入、基于软件的故障注入以及基于仿真的故障注入。

|  |  |
| --- | --- |
| 表2.1 故障注入技术 | |
| 分类 | **介绍** |
| 基于硬件的故障注入技术 | 基于硬件的故障注入是在物理级完成的，通过改变环境参数(重离子辐射，电磁干扰，电源干扰等)干扰硬件或者通过改变[集成电路芯片](https://baike.baidu.com/item/%E9%9B%86%E6%88%90%E7%94%B5%E8%B7%AF%E8%8A%AF%E7%89%87/0?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E6%95%85%E9%9A%9C%E6%B3%A8%E5%85%A5/_blank)管脚输入来达到故障注入的效果。 |
| 基于软件的故障注入技术 | 基于软件的故障注入，是通过在软件级生成错误，从而造成硬件级的故障。有很多注入方式，如修改内存数据，通过[应用软件](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E8%BD%AF%E4%BB%B6/0?fromModule=lemma_inlink)生成故障或者通过底层软件如操作系统生成故障。 |
| 基于模拟的故障注入技术 | 这种技术使用在模拟测试的计算机原型上，通过在模拟过程中，改变逻辑值来达到故障注入的效果。 |
| 基于仿真的故障注入技术 | 该技术主要是针对基于模拟故障注入技术中的时间开销问题。 |
| 混合故障注入技术 | 综合采用多种故障注入技术，取长补短。 |

下面对几种常见的故障注入技术进行简单介绍。

### 故障注入技术

#### 电源故障注入

如在嵌入式系统中广泛使用的80C51微控制器以及在手机等需要加密技术的领域广泛使用的智能卡等芯片，通常都需要外接电源供电。电源故障注入成为了一种常见且经济的故障注入技术。电源故障注入技术通过修改电源电压或在电源引脚上叠加电源毛刺等方式为芯片提供异常电源，使得芯片内部的触发器建立时间和保持时间时序违约，从而在芯片内部注入故障。寄存器保持时间（tsu）和建立时间（thold）示意图如下图所示：寄存器建立时间是指在时钟翻转（对于正沿触发寄存器为0到1的翻转）之前数据（d）必须有效的时间；寄存器保持时间是指在时钟沿到达之后数据输入必须仍然有效的时间。

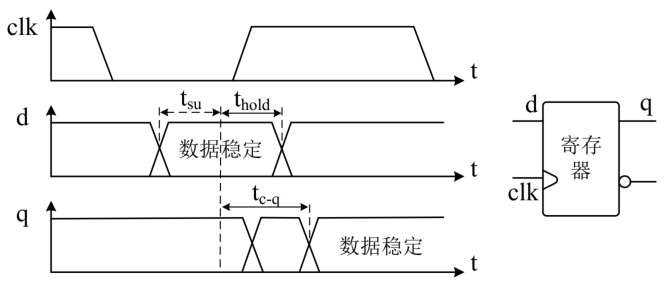


图2.21 寄存器的保持时间、建立时间和传播延时的定义

下图给出了一个数据通路的例子，假设已知时序电路的时序参数：寄存器的最小延时和最大传播延时，寄存器建立时间tsu和保持时间thold，组合逻辑的最小延时和最大延时，寄存器R1和R2的时钟相对于全局参考时钟的位置tclk1和tclk2。理想情况下tclk1=tclk2，则寄存器建立时间约束和保持时间约束分别由如下式子给出：

当电源电压不足时，由于组合逻辑延时增加及寄存器速度变慢等原因，寄存器建立时间和保持时间不满足上述公式，寄存器就会存入错误的运算结果从而在电路中引入故障。

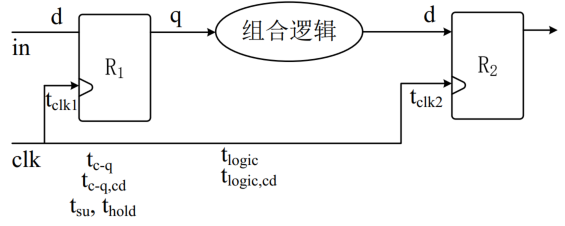


图2.22 数据通路的电路和时序参数

#### 电磁故障注入

电磁故障注入很早就被认为是电子设备的安全威胁，但直到2007年，低成本的电磁故障注入设备才出现，应用于Bellcore攻击，故障注入期间无需解除芯片封装。攻击设备通常由电脑，高压开关电路和电磁探针构成。高压开关电路将存储的能量释放到电磁探针上。探针线圈上循环的电流形成电磁场，将电磁冲击传递到芯片。在芯片内部，由于电磁感应，瞬态变化的磁场在电路中产生脉冲电流，引发电路逻辑错误。

奥斯特实验指出通电导体周围存在磁场。在线圈或微型针上施加快速变化的电流或电压可以产生较强的电磁场。攻击者可以利用这一电磁场对芯片进行攻击。强电磁场会对芯片造成一些复杂影响：

1. 激发硅半导体内电子空穴对，改变流过半导体器件的电流，导致半导体器件失效；
2. 破坏芯片电磁兼容性，造成如芯片内部互连线耦合等问题；
3. 耦合电源等金属线，改变电源电压或器件工作参数造成电路中关键路径时序违约。

电磁故障注入设备简单，攻击代价低，是目前研究最为广泛的故障注入方式之一。

#### 电压故障注入

电压故障注入是指通过改变芯片供电电源的电压来使芯片发生故障。例如，瞬时降低电压值，干扰芯片的正常运行。攻击者可以通过调整电压毛刺脉冲宽度，脉冲延时，脉冲幅度等参数来细化对芯片的影响。例如，降低电源电压会增加组合逻辑的延时，进而引发寄存器采样错误。

这种攻击的主要缺点是攻击者无法精确的将故障注入目标芯片的特定部分。并且，现在大多数智能卡都具有故障检测器和直流滤波器来抵御此类攻击。

#### 时钟故障注入

时钟故障注入与电压故障注入同样属于非入侵式攻击。攻击者可以通过强制时钟信号的过早切换来缩短单个周期的长度，故障会导致一个或多个时序单元的存储错误。例如，经过修改的智能卡读卡器可以通过强制时钟上升沿发生得较早或延迟时钟下降沿来缩短指定时钟周期的长度。这需要攻击设备在目标设备的工作范围内提供常规时钟，同时还要具有更改单个时钟沿的能力；这意味着攻击设备必须以比目标设备更高的时钟频率工作。除了改变时钟周期，时钟毛刺也会引发数据信号的时序违例，进而导致触发器错误采样，引发逻辑错误。

#### 光故障注入

芯片中的主要材料是掺有少量杂质元素的半导体硅。在共价键中处于束缚状态的电子和空穴在获得足够能量后会脱离束缚成为自由载流子。自由载流子在电场作用下形成电流，可使处于关断状态的场效应晶体管等半导体器件导通造成半导体器件失效。提供足够能量的方法可以是光、热或电磁等技术。光激发自由电子和空穴的现象称为光电效应。光故障注入利用光电效应在芯片中诱导故障。光故障注入攻击有着极高的位置精度，能够实现精准的故障注入。

#### 温度故障注入

芯片通常有一个工作温度范围。该温度范围限定了芯片工作环境中温度的上限和下限。当芯片工作在该温度范围以外时，芯片受环境温度影响变得不稳定，可能造成芯片内存储器内容随机变化或芯片功能受限。与光故障注入原理类似，温度变化改变了半导体中自由载流子的浓度，使半导体器件功能异常引发芯片内部故障。借助于这一原理，攻击者可以将芯片暴露在异常温度下，诱导芯片内部故障。

#### 重离子束故障注入

当高能重离子轰击硅衬底时，由于库伦相互作用，可以激发电子和空穴。这些激发出来的电子和空穴只有少部分复合，大部分由于电场作用获得动能成为载流子，增大了半导体器件中的电流。增加的电流可能导致半导体器件工作异常，造成芯片内部故障。重离子束故障注入是一种非常昂贵的故障注入方式。

#### 管脚故障注入

管脚级故障注入试验中采用管脚级故障模型，用以模拟芯片内部及芯片间连线上可能发生的硬件故障．试验中所采用的管脚级故障模型相对于真实故障发生情况的精确程度，直接决定了故障注入试验结果的正确性和准确程度。

#### 开短路故障注入

这两种故障类型属于动态故障注入，是模拟通信总线突然出现的信号线开路或短路情况。故障执行模块检测到来自于指令板的故障触发线为高时，便按照故障注入的配置信息将相应的模拟开关在设定的时间范围内进行打开或闭合操作。

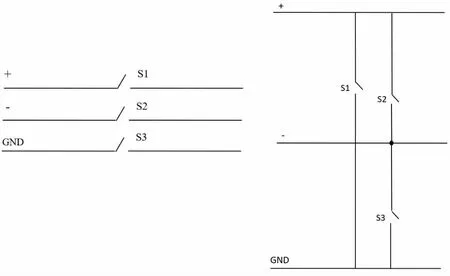


图2.23 信号线开路（左）、信号线短路（右）

#### 信号线串行阻抗和信号线并行阻抗故障注入

串行阻抗和并行阻抗故障注入用于模拟通信总线的物理线路上不同阻抗大小对信号产生的影响，其属于静态故障注入，即无故障触发机制，电阻值在故障注入配置信息下发后就设置完成，通信总线在整个通信过程中均处于该故障模式下。

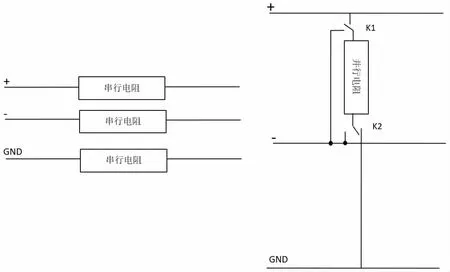


图2.24 信号线串行阻抗（左）、信号线并行阻抗（右）

#### 外部信号替换叠加及噪音耦合故障注入

该故障注入模块为动态故障注入方式，可模拟通信总线在外部信号干扰下的通信。其作为一个整体只能供一路总线使用，下图(a)为外部信号叠加或替换的示意图；下图(b)为噪音串联或并联耦合的示意图，在实际电路中，这4种故障注入模式集成在一起，只能选择一种模式。

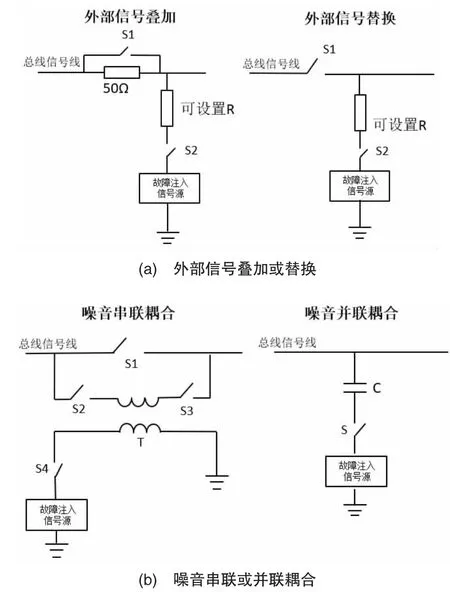


图2.25 外部信号替换叠加及噪音耦合故障注入

#### 固定型故障注入

电路中的某个信号不可控，永远固定在同一值。

#### 杂散电流故障注入

指在设计电路外，因泄漏导致流动的电流。

#### 电涌故障注入

指电压瞬间超过了正常工作电压。

#### 存储数据损坏故障注入

如寄存器、内存、磁盘引发的存储数据损坏故障。

#### 软件缺陷故障注入

机器级或更高级的软件缺陷。

#### 激光故障注入

S. Skorobogatov和R. Anderson在2002年首次使用激光束将故障注入到电路中。激光与硅相互作用产生光电效应，激光器发出高于硅带隙能级能量的激光束，激发电子空穴对，在反向pn结所处的强电场区域，载流子漂移形成电流脉冲。随着电荷耗尽，该电流脉冲消失。电流脉冲会引起瞬态电压脉冲，该瞬态电压脉冲可能会在电路中传播引起逻辑变化，即引发故障。

激光故障注入的特点是能够精确地控制激光注入的能量，波长，位置和注入时间，较为灵活。

### 故障注入应用

#### 在装备测试性验证中的应用

故障注入技术是装备测试性验证中的关键技术。随着装备现代化程度的发展，其测试性水平需要进一步提高，用于验证测试性的故障注入技术需要进一步发展。

测试性是指产品能够及时、准确地确定其状态（可工作、不可工作或性能下降），并隔离其内部故障的一种设计特性。随着总线技术等高新技术的发展，装备复杂度、集成度越来越高，测试性水平的高低直接影响着装备的维修保障能力和作战效能的发挥。

美国等发达国家在武器装备的测试性问题上研究较深，如今其已经广泛应用于航天、航海、导弹、地面装甲车辆复杂系统中。我国的测试性相关技术起步较晚，直到20世纪90年代后期才在《GJB2547A-1995》中明确了装备测试性工作通用要求，而且由于我国测试性验证评价体系构建的不完善，缺少相关软硬件技术作为支撑，只能借鉴国外手段和试验方法，花费大量资金购买测试性设计与验证工具，这对于我国测试性技术研究十分不利。

测试性验证一般在研制阶段和定型阶段进行。在产品的研制过程中，为了确认研制产品是否满足了测试性设计的要求，识别设计缺陷，需要多次进行测试性验证试验；针对于已经定型并且配发部队的装备，由于无法衡量测试性设计水平，为了获取测试性相关数据，发现测试性设计缺陷，提出改进措施，使其改进型装备更加满足测试性设计要求，实现测试性的增长，同样需要对定型装备进行测试性验证实验。

故障注入技术是测试性验证试验中的一项关键技术，针对于不同级别的、不同类型的被单元，其适用的故障注入方法是不同的。现阶段的故障注入方法可以按照不同的标准划分为不同的类型。例如按照故障类型分类可以分为软件故障注入和硬件故障注入；按系故障系统的抽象级别可以分为晶体管开关级、逻辑门级、芯片级、系统级、应用系统级等；按照装备生命周期又可以将不同阶段的故障注入进行区分，如设计、定型、试验﹑接收使用阶段等。按照故障注入实现方法又可以将其分为基于模拟实现的故障注入方法和基于物理实现的故障注入方法两类。基于模拟实现的故障注入方法主要是通过运用相关软件例如MADe、TEAMS或者其他配套软件在计算机上建立被测单元的硬件仿真模型，并在优化选取的测试点进行故障注入或者在模型内部插入故障单元实现故障注入。这种方法成本相对较低，测试点选取灵活，但是其故障模型建立工作量大，无法完全模拟实际被测单元，以至于在工程领域可信度不高，在测试性验证试验中常作为辅助验证手段。固故障注入还需要进一步发展。

#### 在复杂航电系统中的应用

综合航电系统是高度模块化、综合化的系统，随着电子技术的不断进步，航电系统已经成为机上最为关键的组成部分。由于综合航电系统组成的设备众多，信号交联复杂，数据总线和非总线包含的信号数量众多，为满足综合航电系统集成验证过程中在安全性、可靠性、容错性以及测试性等方面的需求，需要对各个系统进行集成测试和故障注入测试。根据航电系统结构，结合功能故障树以及安全性分析结果，在被测航电系统中注入特定故障，进而评估被测航电系统的可靠性。

针对航电系统的故障注入技术目前主要有模拟故障注入和物理故障注入两类，两类故障注入技术对于故障注入的对象均具有一定的针对性和局限性刁而对于复杂的航电系统来说，需要建立针对多种类型信号在不同层次上的故障注入资源，满足航电系统集成验证环节中对于A664、A825、A429、模拟量﹑离散量等不同类型信号在协议层、物理层、电气层不同层次进行故障注入测试的需求。

### 故障注入技术的发展前景

随着装备集成性、封装性的提高,装备发展将会继续沿着信息化、智能化发展。在这样的大趋势背景下,测试性研究进程需要先于装备当前发展,故障注入技术研究也应该更趋向于智能化。故障注入工具应将广泛的理论研究应用于实践,摸索软硬件结合方法,研究便携式.智能化故障注入终端,其软件发展应在吸取国外先进软件,如TEAMS、MADe等的基础上,逐渐开发出国内通用的辅助软件。故障注入技术应结合故障样本选取、故障样本分配的新技术来形成一套故障体系以应用于测试性研究中。

## 侧信道泄露分析

### 侧信道泄露分析介绍

侧信道分析（Side Channel Analysis，SCA）又称侧信道攻击，是一种安全性测试方法，即通过对密码芯片泄漏信息的测试，找到存在的安全漏洞（或破解秘密）。该分析方法需要相应的侧信道仿真工具的支持，即通过仿真技术在设计阶段对密码芯片运行时的侧信道泄漏信息进行模拟。

因为密码学的发展使得单纯在数学分析方面无法进行密码破解工作，所以侧信道攻击的方式被提出且高速发展。促使该技术发展的主要因素有以下几方面：攻击对象的变更、攻击者与理想攻击者的不同、测量工具的精度提升以及密码算法芯片的广泛应用。

攻击对象的变更主要体现在密码学的发展使得密码破解在单纯的数学分析领域的碰壁，促使避开正面破解，采用利用硬件平台的设计局限来进行侧面的攻击方式。采集硬件平台无意识的电磁泄漏分析其中所蕴含的信息来达到破解密码的目的。在密码学中理想攻击者应该在资源受限的条件下进行密码破解过程，但是现实中存在着恶意用户在加密过程中进行信息采集。并且随着科技发展，测量精度越来越高从而促使侧信道攻击的提出，而加密芯片的广泛应用使得该技术迅速发展。

侧信道攻击的原理是在密码算法运行过程中，状态的转变会引起其余物理形式的改变，例如电压，电磁，功耗等。随着这些变化的泄漏，使得内部信息被观测成为可能。另一方面，受到计算平台运算能力的限制，通常密码加密算法会将

密码以及密钥分为较小的子区域，然后按照确定的顺序进行运算。攻击者在不同时间点获取到的泄漏信息通过相应手段取得对应子区域密钥，随后将不同子区域密钥结合密钥扩展设计恢复完整密钥。侧信道攻击原理示意图如图3.1所示：

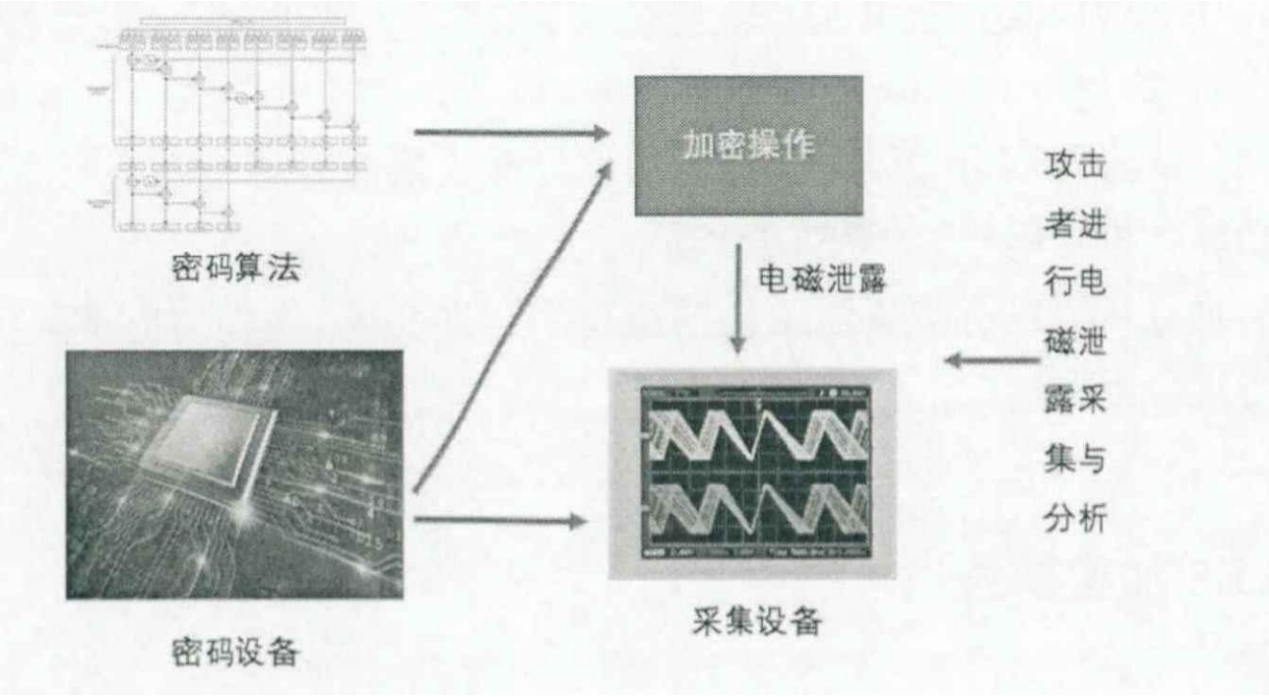


图3.1 侧信道攻击原理示意图

由图3.1可知，密码设备在运行加密算法期间会产生无意识电磁泄漏，攻击者使用采集设备采集电磁泄漏信号，进行相应分析可获取与加密过程相关的信息。

侧信道攻击分为采集阶段和分析阶段。

采集阶段即是攻击者进行电磁泄漏的采集从而收集进行攻击时所需要的信息。采集阶段有被动泄漏和主动诱导两种，而采集的信息的有效性依赖于测量设备以及采集设备的精度。

分析阶段是在采集阶段的基础上针对已获得的信息进行数学统计分析的阶段，利用采集信息不断地进行子区域密钥恢复，最终利用密钥扩展算法恢复完整密钥。

两个阶段缺一不可，相辅相成。采集阶段的精度有利于分析阶段的结果准确度，而分析阶段的算法性能也影响着最终成果。

如今侧信道攻击常常按照信息采集模式、信息采集手段、信息采集内容和信息分析方法来进行分类。

信息采集模式分类即按照采集信息的方式不同来分类，可分为被动和主动两种。被动攻击就是攻击者仅仅利用专业设备针对硬件设备的无意识泄漏信息进行采集已获得密码加密时的相关信息，以期获取密钥信。主动攻击则是攻击者利用专业设备主动干扰正在运行的密码加密设备，从而读取加密期间中间值或者篡改设备固有属性。

信息采集手段分类可分为非入侵式、半入侵式以及入侵式。非入侵式仅仅是对硬件设备进行电磁泄漏采集，并不会对硬件设备进行更改或破坏。半入侵式则需要攻击者对硬件设备进行一定的破坏，但是破坏比较轻微，不会损坏芯片或者硬件设备的主体，例如化学腐蚀表层封装等方式。入侵式则会在半入侵式的基础上进一步进行设备破坏，甚至破坏芯片本体。相应的，半侵入式与侵入式能够获得更加详细的信息，但是攻击更明显，代价更大。而非侵入式更加隐蔽，但是信息不够精确。

采集信息内容分类则是按照采集信息时注重的信息种类不同而分类，例如时耗、功耗、电磁等。

信息分析方法分类有简单侧信道攻击、查分侧信道攻击、相关侧信道攻击、模板攻击等。

#### 软件界面与功能介绍

侧信道泄露分析界面由：连接与输入参数模块、日志模块、曲线仿真模块、脚本控制模块、泄露分析与抗击模块、侧信道结果信息模块组成，如下图：



图3.2 侧信道泄露分析界面

侧信道泄露分析界面主要帮助用户对导入的芯片仿真文件或者通过ip地址与端口进行连接的芯片进行侧信道泄露检测与分析，并且尝试抗物理攻击逻辑测试，然后输出对应的分析报告。

对于登录与输入参数模块，用户可以导入芯片仿真文件，然后系统会模拟分配一个ip地址和端口，以供用户连接，或者用户可以直接输入真实的ip地址和端口来连接芯片读取实时的芯片侧信道数据，同时用户可以设置输入参数以在芯片启动时观察输入参数对芯片侧信道数据的影响。

点击“文件夹”按钮，选择芯片仿真文件，然后导入系统，如下图：



图3.3 打开芯片仿真文件

然后点击“连接”按钮就可以连接芯片了，同时对输入参数进行设置保存，成功后会在日志体现，如下图：

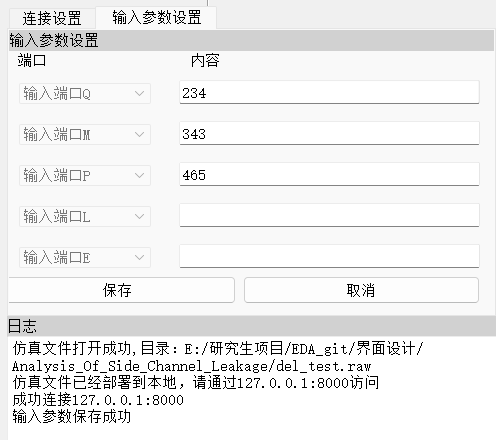


图3.4 连接与保存输入参数

连接成功后，“启动芯片曲线”按钮将会解锁为可点击状态，点击该按钮，即可将读取的芯片功耗与电磁辐射以曲线的形式动态加载出来，如下图：

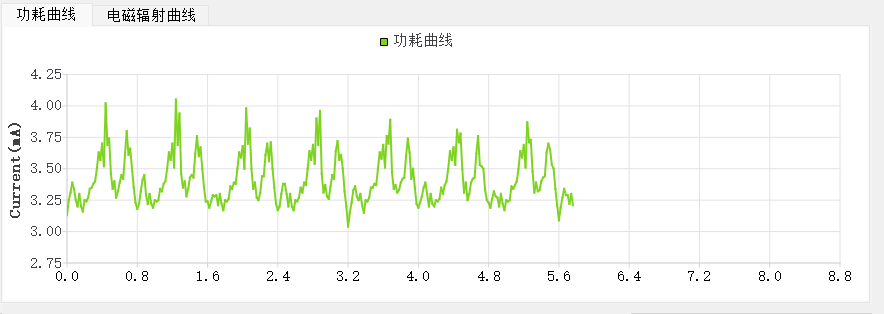


图3.5 功耗曲线

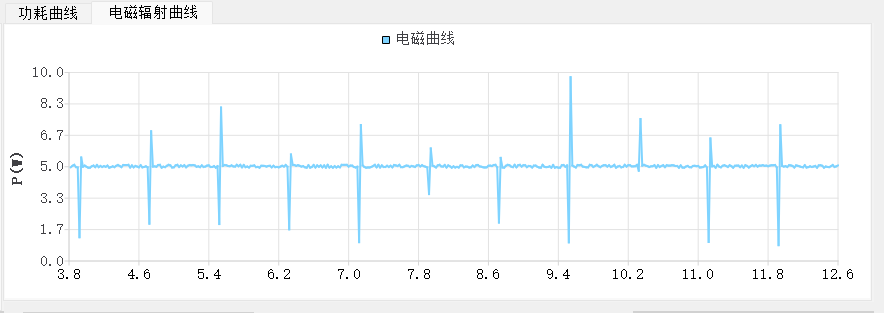


图3.6 电磁曲线

芯片的实时功耗曲线和电磁辐射曲线将会动态加载出来，此时用户遇到需要分析的地方可以点击“暂停芯片曲线”以冻结曲线窗口，使得特殊部分得以暂停住，分析完毕后再次点击“启动芯片曲线”即可继续曲线的运动。如果曲线分析完毕想要重置曲线状态，可以点击“停止连接芯片”，此时所有曲线将会被清空，同时系统停止继续读入功耗与电磁辐射数据，如下图：

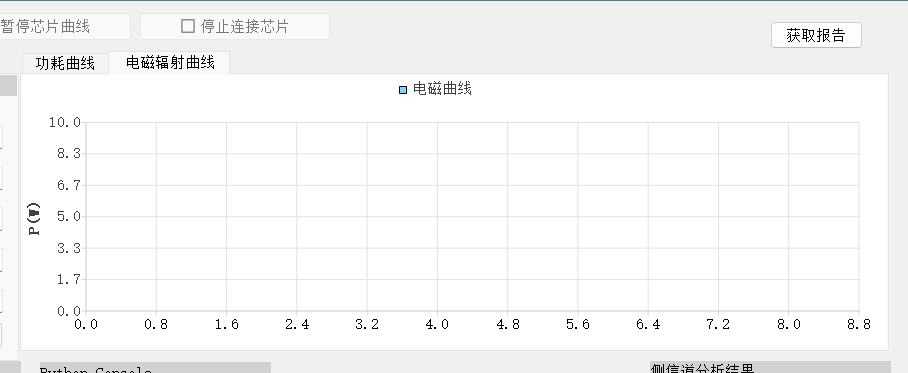


图3.7 停止连接芯片

对于脚本控制模块，用户可以使用命令来替代按钮，以更快的进行侧信道泄露分析任务，如下图：

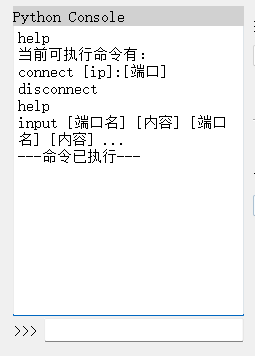


图3.8 脚本控制模块

泄露分析与抗击部分，是针对当前芯片读取的曲线，进行抗物理攻击逻辑注入或者进行侧信道泄露分析的操作区域，下面图片将分别展示各个逻辑和方法的效果：

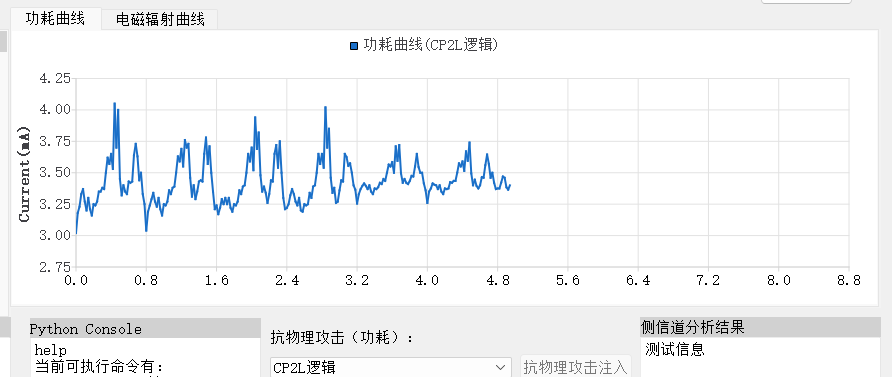


图3.9 CP2L逻辑

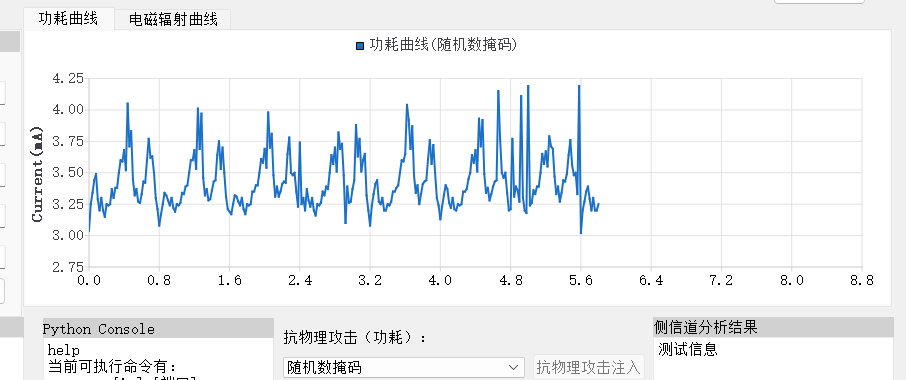


图3.10 随机数掩码

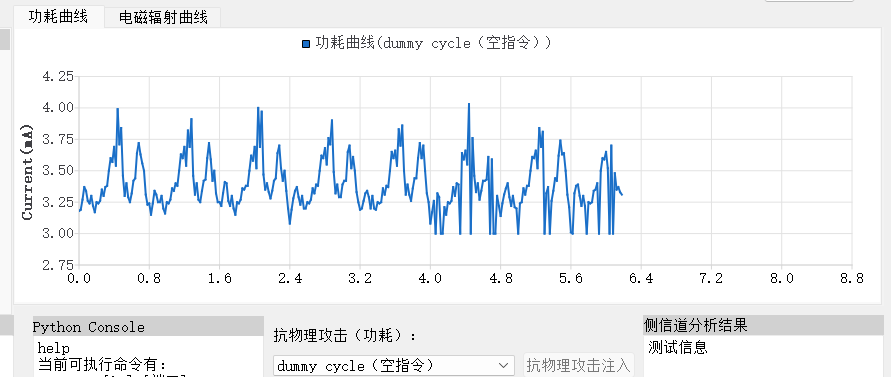


图3.11 空指令

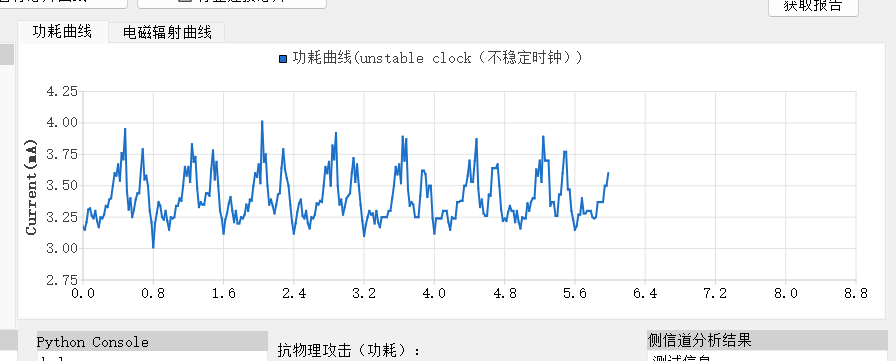


图3.12 不稳定时钟



图3.13 差分功耗分析

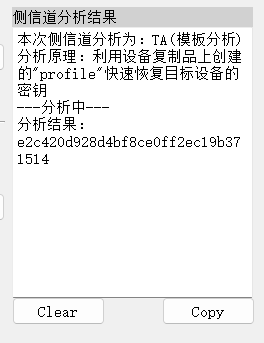


图3.14 模板分析

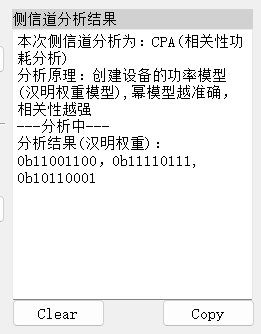


图3.15 相关性功耗分析

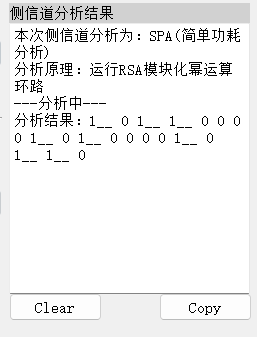


图3.16 简单功耗分析

当完成了全部的侧信道泄露分析后，可以点击界面右上角的“获取报告”，以预览当前芯片的侧信道泄露分析报告，如下图：



图3.17 报告预览

在报告预览页面，点击页面右下角的“生成word”按钮，即可生成对应报告的word版本。

#### 侧信道泄露的危害

1984年IBM Selectric II和Selectric III两款打字机中有16个机电漏洞被发现。苏联，在这些漏洞别世人知晓之前利用它们开发了世界上第一个键盘侧信道攻击硬件Selectric Bug。

Selectric Bug包含6个磁力计，每次磁力计都能感应到IBM Selectric打字机中6个独立杠杆的运动。驱动杆的独特组合使得Selectric Bug能够在忽略了Shift键，空格键和其他一些非字母键下确定使用者按下了哪个键。它最多记录八次键盘敲击，当它储存满时，存储器中的内容通过无线电突发传输到附近的监听站。苏联人开发了至少5个版本的Selectric Bug，使他们能够记录美国驻莫斯科大使馆和美国驻列宁格勒领事馆人员的击键超过8年。

Selectric Bug是世界上发现的首批键盘记录器之一，代表了监控技术的里程碑。它可能是公众所知的最成功执行的键盘输入攻击之一。

2016年CT-RSA上，以色列特拉维夫大学Tromer团队展示了如何利用电磁攻击获得隔壁房间计算机中的信息。2017年，NXP公司的Wagner等人破解了竞争对手生产的一款通过了国际CC EAL5+安全认证的智能卡中的3DES算法。2018年1月，Intel处理器被爆出MeltDown和Spectre漏洞，该漏洞使云服务器可能遭受来自千里之外的侧信道缓存分析。

#### 侧信道泄露分析的作用

随着集成电路技术的发展，密码算法以硬件电路的实现方式出现在密码设备中。在实际应用中，这些硬件设备会泄漏出能量、电磁等多种类型的物理信息，称为侧信息。侧信道分析技术利用这些信息，能够直接或间接获取密码算法运算过程中的中间值信息，进而可以分段恢复较长的密钥。

### 侧信道分析方法

侧信道分析方法主要分为两大类：建模类侧信道分析和非建模类侧信道分析。两者的攻击流程不同，实施的假设条件也不相同。

在非建模类分析中，假设攻击者能力较弱，只能从被攻击设备被中获得能量迹的测量值，通过选择算法中间值和合适的能量消耗模型并结合统计分析技术来推测正确密钥。这一类中代表性的方法为差分能量分析和相关能量分析。

在建模类分析中，假设有一个强大的攻击者能够操控与攻击设备相同的设备，通过控制其输入输出采集大量的数据来构建不同中间值与设备泄漏间的模型，最终对目标设备进行攻击，因此该分析需要经过两个阶段:建模阶段和分析阶段。模板攻击是最经典的一种建模类分析方法，在建模阶段为通过假设多元高斯分布为泄漏模型的每一类（如汉明重量有9类，对应每个汉明重量值）建立模板。在分析阶段，利用每个模板计算能量迹属于此类的概率，通过极大似然估计确定实际的密钥。因为建模类分析一般比非建模类分析更强大，恢复正确密钥所需要的更少的能量迹。

侧信道分析涵盖以下工作：

1. 选择分集函数

加密算法不同，对应的区分函数实现也不同，因此使用区分函数库对不同的区分函数进行保存。曲线分集子模块根据系统配置模块的设置，从库中选择指定的区分函数，并将侧信道泄漏曲线数据分配到两个集合中。为了实现分析策略和仿真机制的分析。

对于不同的分析策略，如简单侧信道分析、差分侧信道分析、高阶差分侧信道分析、模板分析等，用构件技术对选择函数进行封装。

例如CAesSboxSelFunctions、CAesXorSelFunctions分别代表针对AES算法Sbox和异或运算的区分函数类。通过策略和机制相分离，使得不同攻击方法和区分函数能灵活地加入到仿真引擎中，从而让仿真环境更具可扩展性。

1. 对泄露曲线进行分析

在（1）的基础上，分别对不同的集合求取其平均信号，然后再求取两个结合的平均差分信号。

1. 绘制分析结果曲线图

利用步骤（2）的输出结果生成对应的分析曲线，提供较为直观的分析结果。

#### DPA差分功耗分析

与传统密码分析不同，差分功耗分析(Differential Power Analysis, DPA)作为一种侧信道分析形式，更加关注密码算法的具体实现，利 用功耗与密码电路所处理的数据之间的相关性实施攻击，给密码芯片带来严重 的安全威胁并引起国内外相关学者的极大关注。在此形势下，分析密码芯片的 潜在威胁并探索有效的防护措施，对保障信息安全具有重要意义。

#### TA模板分析

模板攻击是 2002 年 Chari等人提出的一种新型侧信道攻击方法，主要分为模板构建阶段和密钥恢复阶段，其中模板构建阶段需要利用算法加密过程中泄露的物理信息，建立能反映泄露信息特征的模板。

#### CPA相关性功耗分析

在 CPA 攻击中，针对首轮的明文攻击和针对最后一轮的密文攻击是两种主要的攻击方式，两种攻击方式的基本原理和攻击方法相似，但相形之下，由于首轮运算中包含了轮置换函数，所以明文攻击要比密文攻击复杂度高。针对明文的 CPA攻击主要分为以下4个步骤:

1）利用 HDL 语言完成 Piccolo 算法的硬件设计

2）采集不同明文加密时的功耗信息，建立矩阵Pact，同时记录对应的明文

3）基于汉明距离建立假设功耗模型，建立假设功耗矩阵 Phyp，利用明文和密钥猜测值推算出加密过程的某一中间值，将每一条明文的该过程映射为功耗信息，形成假设功耗矩阵 Phyp，这一步是能否成功实施CPA攻击的关键

4）对Pact和Phyp进行数学统计分析，完成对攻击密钥的攻击，获得攻击密钥的最可能值

#### SPA简单功耗分析

SPA是一种直接解释功率消耗测定值的技术，能够给出关于一个设备的运行信息以及密钥信息。在攻击中，攻击者直接观察一个系统的功率消耗，利用加密操作实现细节与功耗之间的关系，直接从一次测量的功耗轨迹获取密钥信息。

### 泄露分析工具

#### 芯片功耗/电磁泄露模型

当前的电子集成电路大都采用MOSFET实现，常用的电路逻辑有静态互补CMOS逻辑、传输门逻辑和动态逻辑，其中以静态互补CMOS逻辑使用最为广泛，目前密码芯片电路的制造核心基本都是CMOS工艺技术，密码芯片只要在运行就会产生功耗泄露，因为在工作状态（加密和解密）下电路的元件就会利用电平翻转完成工作逻辑，不同的工作逻辑所调用的元件是不同的，并且每个元件调用的次数也是不同的，而所有元件消耗的能量就会表现为整个加密芯片的电流变化，也就产生了密码设备功耗。以反相器说明静态CMOS的功耗特征，如下图所示，该反相器由PMOS和NMOS组成，CL为其输出端负载电容。

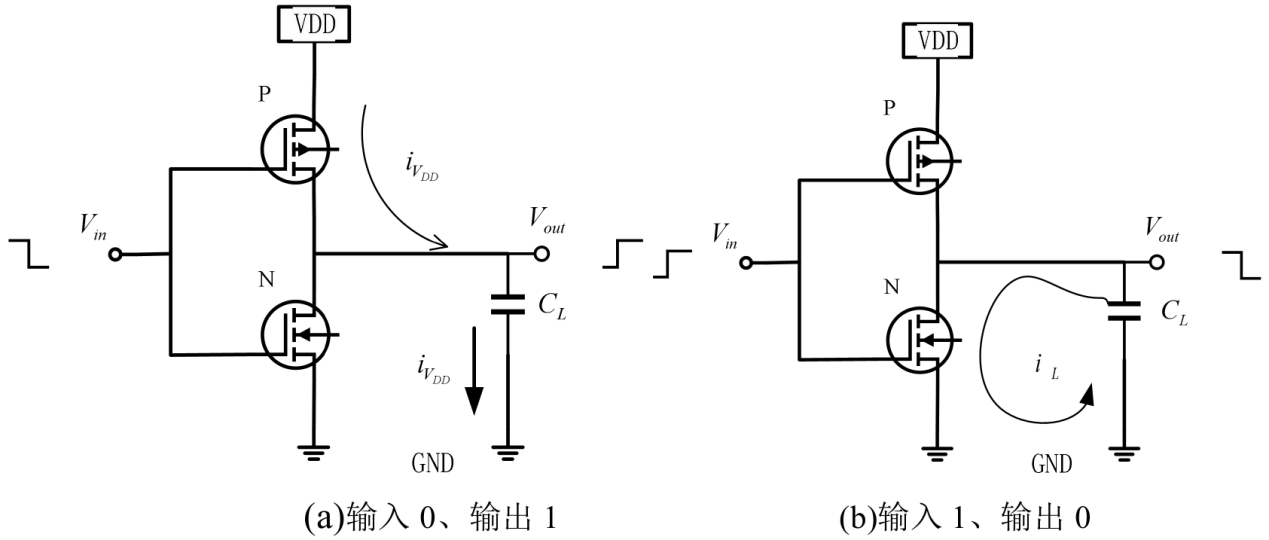


图3.18 CMOS反相器工作电路图

上图(a)中是输入为0、输出为1的过程，此时PMOS导通、NMOS截止，LC充电，消耗外部能量；上图(b)是输入为1，输出为0，PMOS截止，NMOS导通，LC放电，释放能量。

CMOS门电路总功耗为：

其平均功耗为；令为充电功耗，短路功耗则是，这部分是电路中的动态功耗，约占全部功耗的99%；是CMOS的漏电流产生的功耗，为静态功耗，占剩余的所有功耗。

当CMOS门电路状态变化的时候，由两个主要原因产生和；

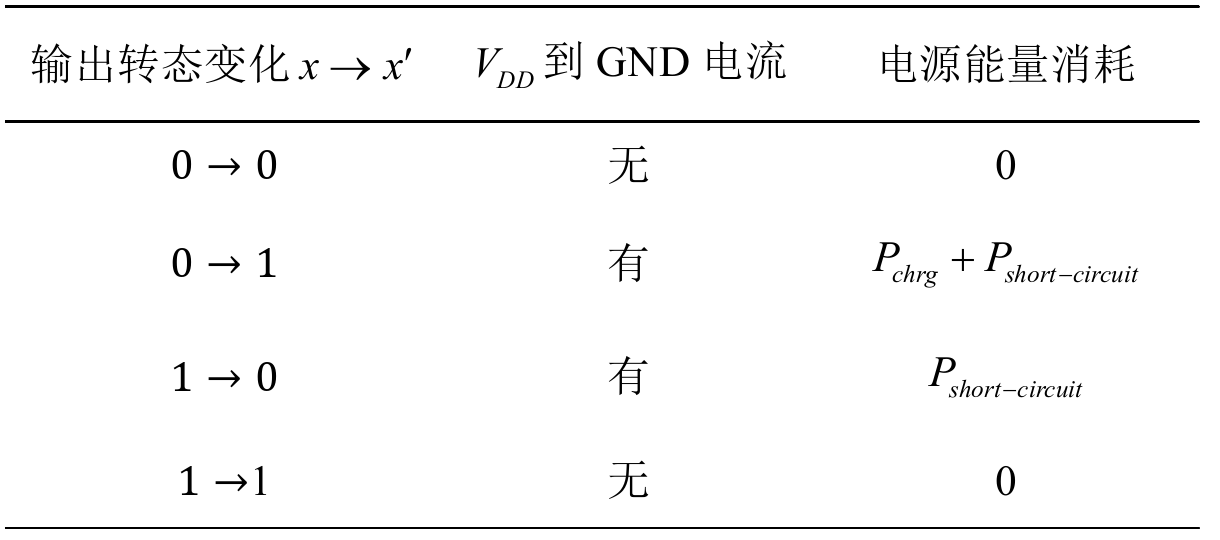
①负载电容。导线与CMOS晶体管的门电极会形成CL，其大小由导线长度和晶体管数量决定，该负载电容在每次1→0时通过PMOS充电，反之则放电，充电功耗约占CMOS门电路电流的15%，由下方公式计算

其中代表一个单元内的瞬时消耗；f表示时钟频率；则是稳压电源；是一个周期里0→1的变换概率。

②短路电流。虽然相应的通道已经打开，但是整个电路的状态无法瞬时变化，不论是从0→1或者从1→0，都存在PMOS和NMOS同时导通的可能，虽然这种短路时间非常短（短路时间记作），但是其短路电流却非常明显（电流峰值记作），大概是整个CMOS门电流的85%，由短路电流产生的功耗如下，其中α表示逻辑变化的概率之和。

以电路中的反相器为例，设x和x’分别代表静态CMOS反相器变换前后的状态，其数据变化与侧信道信号量值间相关性如下表所示，已忽略1%的漏电流。

表3.1 静态CMOS反相器功耗与内部运算的相关性



由上述可知，加密芯片的能量消耗与数据状态转换有关，数据状态的转换是数据计 算的表现，计算越复杂产生的能量消耗也就越多。芯片的加密过程可以视为函数映射，输入明文、输出密文，芯片的总功耗P和输入data的关系:

但是加密芯片的构成太过复杂，任何一块加密芯片中都有百万甚至千万个逻辑门、 而且制作芯片的材料不同其内部连接也各有不同，在现有的水平下，无法在某一时刻对一块加密芯片工作的逻辑门进行统计，因此不能直接根据上述公式计算出电路中的能量消耗。目前只能利用统计学工具，计算某一时刻数据与功耗的相关性并进行累计，而噪声 则被强制平均分布在所有能量迹中，因此当某一数据与功耗的累计值超过其他数据时， 就能够利用功耗计算出加密数据。

在基于深度学习的能量分析中，通常将密码设备中的操作数映射为能量消耗值，利 用能量消耗搭建攻击模型，其中汉明重量模型和数字模型是现代侧信道分析中常用的两种功耗模型。

①汉明重量模型。在AES-128中，数据经加密过后的输出总是在0-255之间，将 每一个输出用二进制表示，汉明重量则是每个二进制表示中“1”的个数，下表是输出数据与汉明重量的映射关系。

表3.2 汉明重量模型映射关系表

文本

中度可信度描述已自动生成

②数字模型。在汉明重量模型中，以二进制中“1”的个数作为标签，但是这种标签与深度学习搭配的效率并不高，因为在同一种标签中有多个情况需要另外处理，因此数字模型直接将输出的数字作为标签，在数字模型中共有256个标签，每一个标签就是加 密芯片输出的十进制数字，其映射关系如下表所示，这时每一种情况都指向唯一的输出，在模型进行数据分类时可以最快速度获得目标类别，有利于提高侧信道攻击的效率。

表3.3 数字模型映射关系表

形状

描述已自动生成

AES中的字节替换是加密流程中的唯一的非线性操作，也是AES加密算法的核心步骤。字节替换对状态中的每个值进行可逆替换，将字节矩阵中的每个值独立的按照一定的规律在S盒中进行查找，映射到具体值之后将映射值替换原来的字节的值。这种替换操作给密码算法带来的非线性能够有力地抑制传统密码分析技术。在硬件加密过程中，由于S盒的输出值是根据明文和密钥异或之后查表确定的，因此，如果能够确定S盒的输出值与明文，那么就可以逆推出这次加密过程中使用的密钥。如下图所示，获得S盒输出值z之后，通过S盒列表查询S盒输入值y，再结合输入明文p就可以利用异或操作计算出密钥k。

卡通人物

中度可信度描述已自动生成

图3.19 功耗攻击流程

1. SPA

在简单功耗分析中，攻击者采集加密芯片的功耗后，能够根据自身经验从功耗数据 中观察出加密芯片的具体操作。

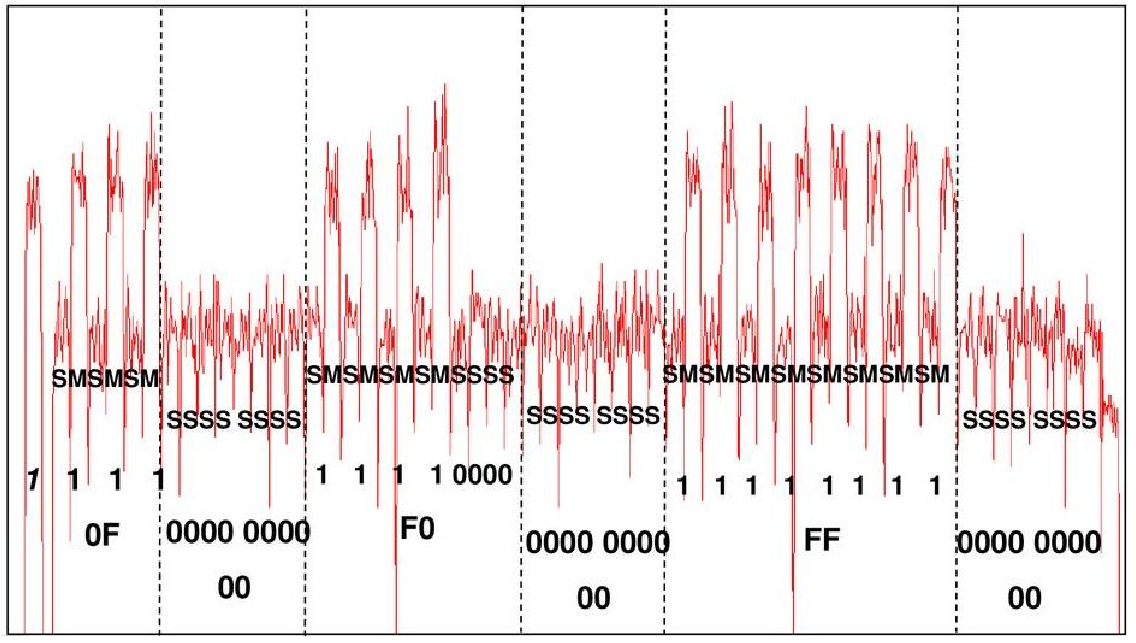


图3.20 RSA的简单功耗分析

加密设备消耗的能量与具体的操作有关，通过这种逻辑，有经验的攻 击者在观察功耗踪迹时能够发现所用加密算法的执行细节，利用这些细节，攻击者就可以推测出加密算法的种类以及加密过程中的某些具体操作。如上图所示，利用RSA 加密算法中乘法和平方运算之间的差异恢复RSA加密密钥，其中S代表平方运算 （Square）、M代表乘法运算（Multiplication），根据S、M运算转换规律，可推测出加密秘钥为000F00F000FF。

1. DPA

差分能量分析主要工具是特征分析方法，利用元件的功耗特征和大量加解密时的泄露功耗恢复密钥。DPA与SPA不同，攻击者不需要对加密设备有很深的掌握程度，只需要采集大量的泄露功耗就可以进行特征分析。在DPA攻击中，首先需要一台加密硬件和采集功耗泄露的设备，由于DPA攻击利用的是包括噪声的功耗分析模型，所以采集的功耗踪迹包含的噪声越小越好，因为噪声越小带来的干扰也就越少。

而加密芯片的噪声有外部噪声（外部环境和数据连接线不稳定造成）、内部噪声（元件之间的相互干扰造成）、量化噪声（由调制解调算法导致）和算法噪声（防御侧信道 而添加的干扰操作造成），与外部噪声和算法噪声相比，逻辑元件之间的内部噪声和量 化噪声所占噪声比列小很多，因此可以使用精度高的仪器减少外部噪声，而算法噪声则主要通过DPA算法进行平均，因此大量的能量迹能够减小噪声对DPA攻击的影响。DPA攻击可以化分为六个步骤：

① 选择AES计算过程中合适的中间状态。state=f(d, k)，其中d为明文或密文，k为假设密钥；

② 记录能量迹。设每条能量迹有T个采样点，共记录D条能量迹，则功耗矩阵为D×T的数组；

③ 计算假设中间状态的十进制具体数值。，D×K的矩阵V是所有假设密钥与明文计算的结果；

④ 将中间状态值转化为能量消耗。利用汉明重量模型，将所有中间值的数值换算为汉明重量矩阵H；

⑤ 计算能量迹与汉明重量矩阵的关系。将功耗迹线T与H在每个位置上都进行计算，获得的K×T矩阵R；

⑥ 查找R中的最大值，该值的位置即为DPA攻击的结果。

1. 相关能量分析（CPA）

CPA是另一种常用的攻击方法，也是当前基于深度学习的侧信道攻击的重要工具。和DPA攻击一样，在进行CPA攻击之前，首先采集加密设备连续功耗，然后构造汉明 重量功率模型。CPA的目标是区分出真实秘钥的中间状态和功耗的突出关系，其工具一般为pearson相关系数法，计算过程下方公式。其中是猜测密钥i与采样点j处的相关系数，是第d条能量迹在采样点j处的功耗测量值，是能量迹的假设密钥i的预计功耗，其输入一共D条功耗踪迹，就是根据汉明重量计算得来的，越大表明相应位置的猜测密钥更接近真实密钥。

#### 测信道泄露仿真引擎

侧信道泄露仿真引擎在虚拟的特定硬件平台上运行算法程序，根据预先建立的侧信道信息泄露模型分析从算法库中选择的加密算法指令，并产生相应的侧信道泄露曲线。

（1）建立侧信道泄露特征库

侧信道泄漏特征库是整个仿真软件的核心，存储着不同硬件平台指令系统的侧信道泄漏模型。建立侧信道泄漏信息库有两条途径：一是利用侧信道泄漏特征的数学模型，对其泄漏特征进行建模，例如执行时间、功耗等；二是通过大量的实验，测试不同硬件平台执行各条指令时真实的侧信道泄漏信息，并将指令与泄漏特征对应起来形成特征库。对于模板攻击方法（Template Attack）而言，主要采用第二种方法建立泄漏特征库。

（2）建立密码算法指令和泄露特征的映射关系表

侧信道泄漏的特征信息与机器指令存在着密切的关系，不同的机器指令执行时泄漏的信息是不相同的。另外，即使是同种指令，操作数据不同时泄漏信息也是有差异的。为了建立密码算法与侧信道泄漏的联系，首先利用编译工具对密码算法进行编译，生成优化的机器指令代码；接着利用机器指令作为索引，在侧信道泄漏特征库中查询对应的泄漏特征的入口。如下表所示，通过机器指令的分析，可以建立指令和泄漏之间的映射关系。

表3.4 指令与泄露特征映射表

|  |  |  |  |
| --- | --- | --- | --- |
| 芯片类型 | 机器指令类型 | 泄露类型 | 泄露特征库入口地址 |
| AT89C51 | JZ跳转指令 | 功耗泄露 | 地址1 |
| AT89C51 | Move传送指令 | 电磁泄露 | 地址2 |
| AT89C20 | Add加法指令 | 执行时间 | 地址3 |

然而，对于被分析的密码算法而言，哪些指令属于需要重点分析的运算指令，由用户选择的侧信道攻击方式所确定。以 AT89C51 为例，如果用户选择数据 DPA 攻击，那么主要关心的是跟寄存器和存储器操作相关的指令，则会重点选择比如 move 指令、ADD指令；如果是简单能量分析SPA攻击，那么条件跳转语句JZ、JNZ指令等则属于主要分析的运算指令。

（3）绘制密码算法运行全过程的侧信道泄露曲线

对（1）、（2）的结果进行综合，生成对应的特征曲线。

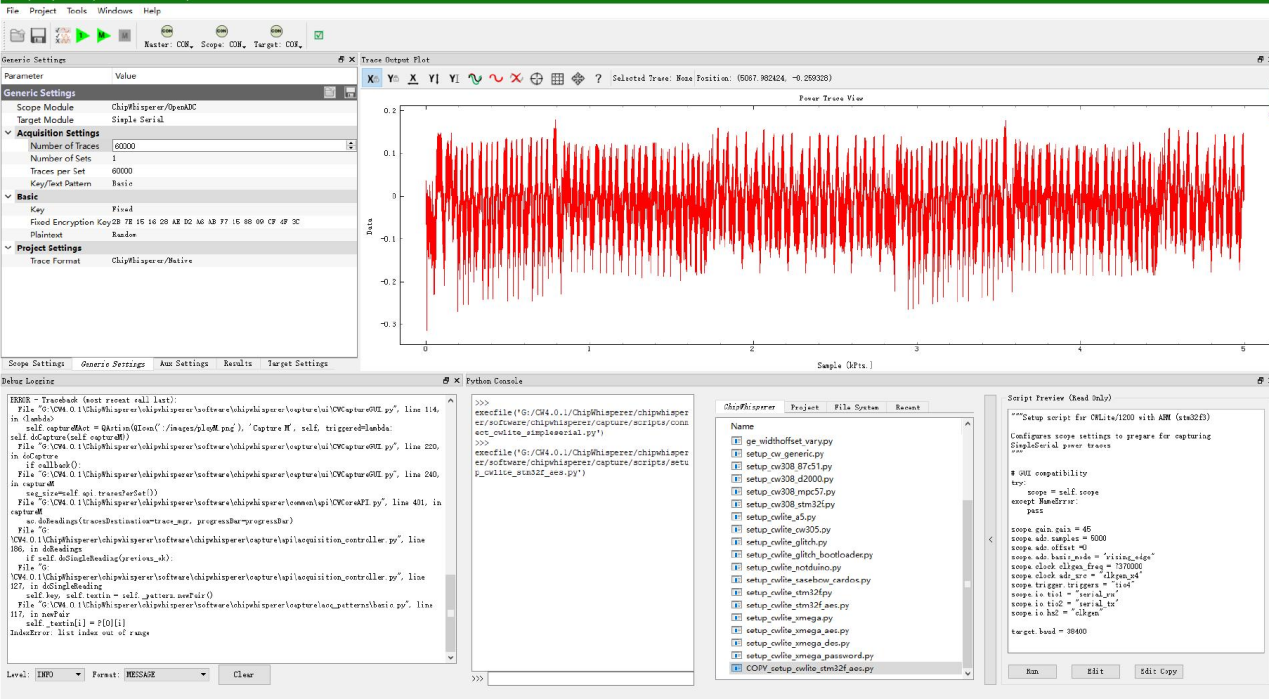


图3.21 Chipwhisperer功耗分析平台

## 算法门限安全检测

### 算法门限安全介绍

随着计算机网络以及信息技术的飞速发展，为了解决网络系统中的单点故障问题，增强系统抗攻击能力，分布式环境下的安全性越来越受到人们的重视。门限密码体制是解决这类问题的一种最有效和最具潜力的技术。同时，由于分布式环境的复杂性，使得研究工作具有一定的难度和挑战。近几年来，国内外许多学者对于门限密码系统进行了深入研究。然而，就我们所知，对不同公钥密码环境下的门限密码系统的形式化安全性研究尚不够完善。而是否具有形式化的可证安全性，是衡量密码体制是否安全的最重要的标准。因此，开展门限密码体制的形式化研究，不仅对网络安全及信息化建设具有重要意义，而且具有重要的学术价值。

#### 软件界面和功能介绍

算法门限安全检测界面主要有算法文件读取、门限设置、功能区、消息框几个部分组成，主要作用是对算法文件进行门限检测，以衡量算法门限的安全性，算法门限安全检测界面如下图：



图4.1 算法门限检测界面

在该界面，可以点击算法文件目录这一行右边的三个点按钮，该按钮点击后可以选择算法文件，进行算法门限安全检测，如下图：



图4.2 算法文件导入

导入算法文件后，可以在下方进行算法门限设置，点击“算法门限设置”会弹出门限设置窗口，即可配置相关参数，如下图：



图4.3 门限设置

门限设置窗口有如下参数可以配置：密钥长度、安全参数、密码算法轮数、迭代次数、算法模式、时间限制、空间限制、攻击模型，设置好参数后，点击保存即可将相关设置存入系统，然后退出弹窗点击“密钥生成”按钮，如下图：



图4.4 密钥生成

点击“密钥生成”后，会在密钥栏依托算法文件生成密钥，并在消息栏记录密钥生成时间是否通过门限检测。同时会开启“随机性测试”、“密钥空间分析”、“自动化检测”三个按钮。

随机性测试会从系统中的样例中随机抽取部分样例对该算法门限进行检测，然后输出通过样例与通过率，以检测算法的健壮性，如下图：

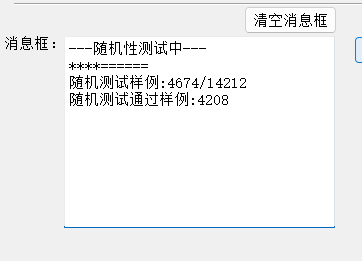


图4.5 随机性检测中

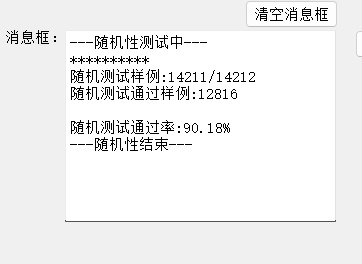


图4.6 随机性检结束

密钥空间分析，会根据密钥空间算出密钥长度，与密钥可显示范围，自动化检测即使系统结合算法门限检测自动安排密钥生成、随机性测试、密钥空间分析等操作，如下图：

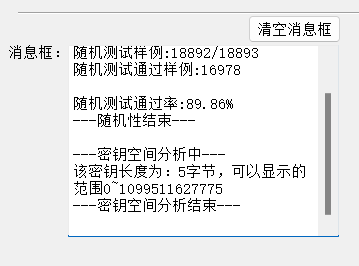


图4.7 自动化检测

通过算法门限安全检测，可以对算法门限的安全阈值进行探测，以保证在一定范围内，算法可以安全在芯片中运行，以防止芯片出现未知错误，同时增强芯片的稳定性。

#### 算法门限安全的定义

传统的数字签名，例如RSA以及DSA仅允许一个签名者对消息进行签名，任何人可以在任何时候验证该签名。但是很多情况下，为了将单个权威的权利进行分散时，可能要求对某一个消息进行多次签名。为了解决这个问题，研究人员提出了多重签名和门限签名。但门限签名和多重签名有着很大的区别：其一，在多次签名方案中，对参与有效签名的签名者个数没有严格的限制，但是在门限签名中，为了使至少t个参与者能代表群组合作完成签名而少于t-1个参与者无法完成，必须事先确定门限值t的大小；其二，门限签名代表的是某一个群组对某个消息的签名，即该签名代表了这个群组的行为，而多重签名则是代表每个参与者对消息进行了签名，签名代表的是个体行为。

Desmedt于1987年首次提出了用于群组与群组之间的安全通信的面向群组的密码系统。面向群组的密码系统能够被用于门限签名。Desmedt和Frankel首次提出了基于RSA和秘密共享的面向群组的(t,n)门限数字签名方案，后来，Li等指出t个或者更多的恶意参与者能形成合谋攻击，从而得到系统的密钥进而伪造签名，但验证者对此一无所知。因此，很多学者利用各种密码技术提出了门限签名方案：如ElGamal型、RSA型、椭圆曲线型等，这些方案具备不可否认性、可跟踪/不可跟踪性等特性等多种特性。

Harn在1994年结合Shamir的秘密共享方案和改进的EIGamal签名实现了一个(t,n)门限签名方案。由于Lagrange多项式的特点，群组的密钥被分散成n个不同的影子密钥并分发到每一个参与者手中。任何t个或者更多的参与者能用他们的影子密钥产生他他们的个人签名。可以随机的指定参与者中的任何-一个参与者负责个人签名的验证并且利用Lagrange插值方案计算出群组的签名。参与者的个人签名和群组的签名的安全性都是基于离散对数难题(DLP)。后来，Horster等，Chang 等，Michels 641分别指出 Harn 的方案无法抵抗假冒攻击。另一方面，Park 等针对ElGamal签名的一个变体提出了一个门限签名方案，Miyazaki等提出了门限 Nyberg-Rueppel型签名和签名共享。签名共享是另一种面向群组的技术，在这种技术中，只有一个签名者负责签名，他将该签名分发给n个签名持有人。Wu等在2003年利用自认证的公钥提出了一个门限签名方案。与以前的方案不同，这种方案属于基于证书的公钥系统。基于证书的公钥系统不会有额外的通信代价用于传输公钥证书，验证公钥的计算代价及用于存储证书的存储代价(在基于证书的密码系统中需要存储证书)。Shao在2005年基于Wu-Hsu的方案提出了一个改进方案，该方案在签名计算和验证方面更有效。Liu等在2003年提出了一个基于RSA 的门限GQ签名方案，Miyazaki 等在2001年提出了一个基于椭圆曲线ElGamal 的门限签名方案。由于椭圆曲线密码的优点,他们的方案很容易在智能卡上实现。Wu等在2003年指出Miyazaki-Takaragi方案无法抵抗假冒攻击并提出了一个改进方案。

不可否认的门限签名是一种特殊的签名，签名的有效性必须有签名者的合作才能得到验证。由于不可否认签名方案中签名的有效性受限于指定的验证者，所以可用于电子商务，公平交易协议等。如果签名是某个签名者所签，他是不可否认的。

## 版图安全检测

### 版图安全检测介绍

#### 软件界面和功能介绍

版图安全检测界面，分为文件目录选择、型号完整性审查、DRC审查、脚本规则审查、反馈结果三个部分，如下图：

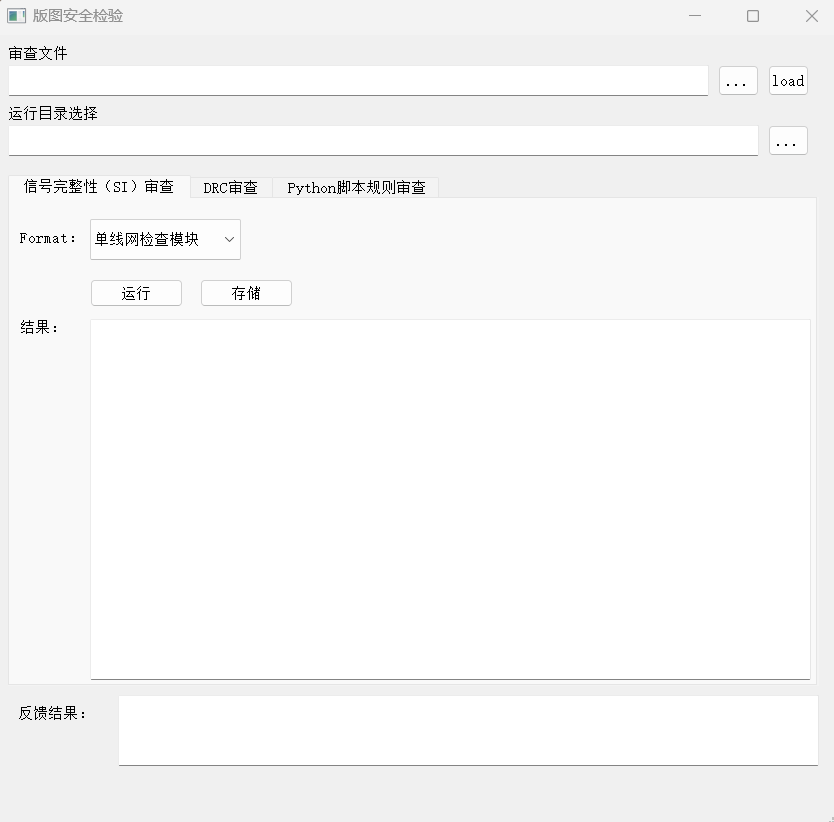


图5.1 版图安全检验

点击审查文件一行的三个点按钮，即可选择需要审查的版图文件，如下图：

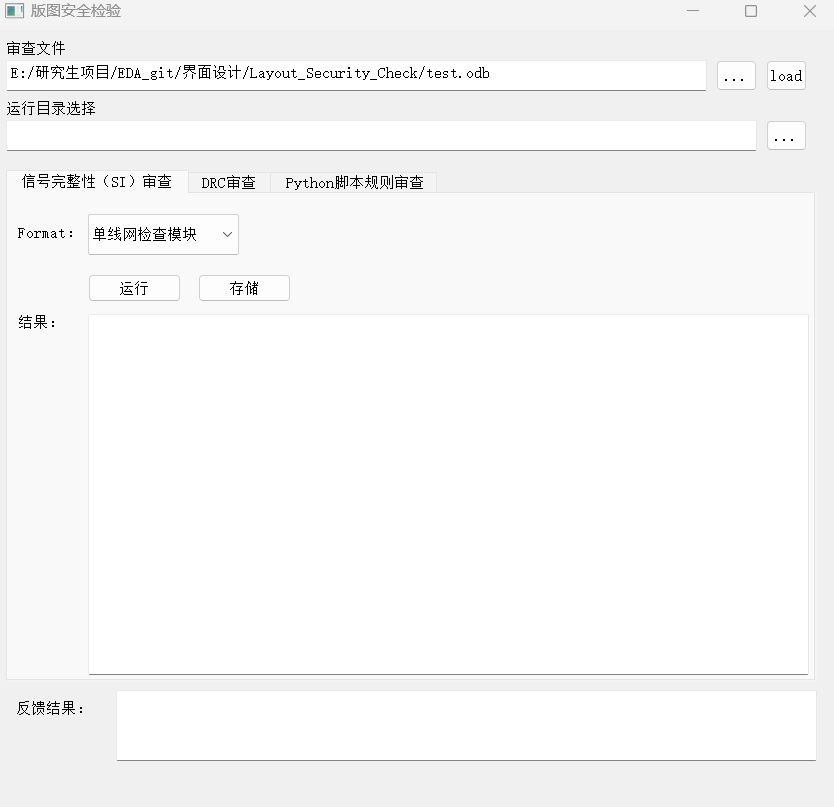


图5.2 导入版图文件

选择好审查文件后点击“load”按钮，载入审查文件，在反馈结果处会给出文字提示，如下图：

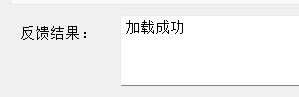


图5.3 导入版图文件

然后就是选择运行目录，点击运行目录后面的“…”按钮，以导入运行目录。共有三种审查模式可供选择。按钮运行则可在结果中展示运行后的结果，点击存储则储存内容。反馈界面则是反映软件运行情况，如下图：

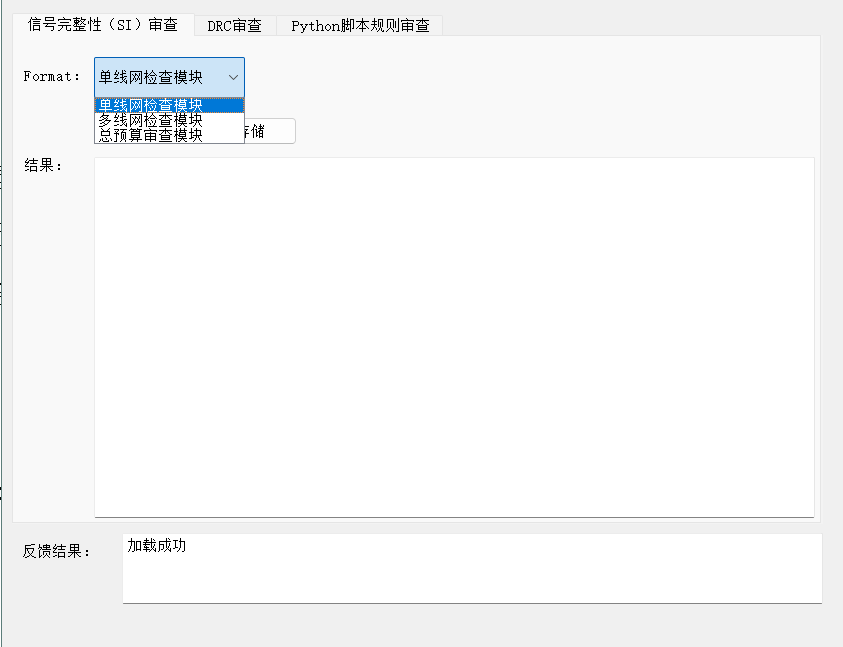


图5.4 三种审查模式

对于信号完整性审查，我们实现了分为单线网检查模块，多线网检查模块和总预算审查模块。对于DRC审查模块，该板块需要导入预设设计规则，运行及存储按钮作用与信号完整性检查模块相同，如下图：

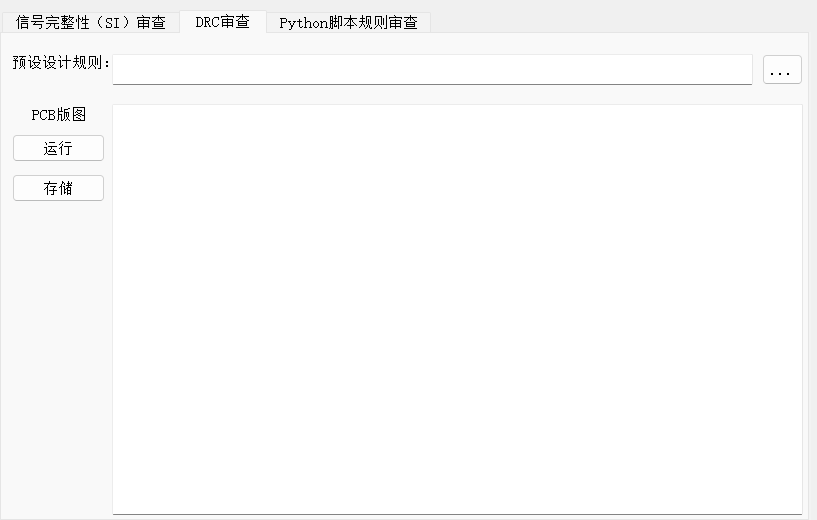


图5.5 DRC审查

对于python脚本规则文件，先选择对应的python脚本规则文件，再点击“开始审查”按钮，即可得到对应结果，如下图：

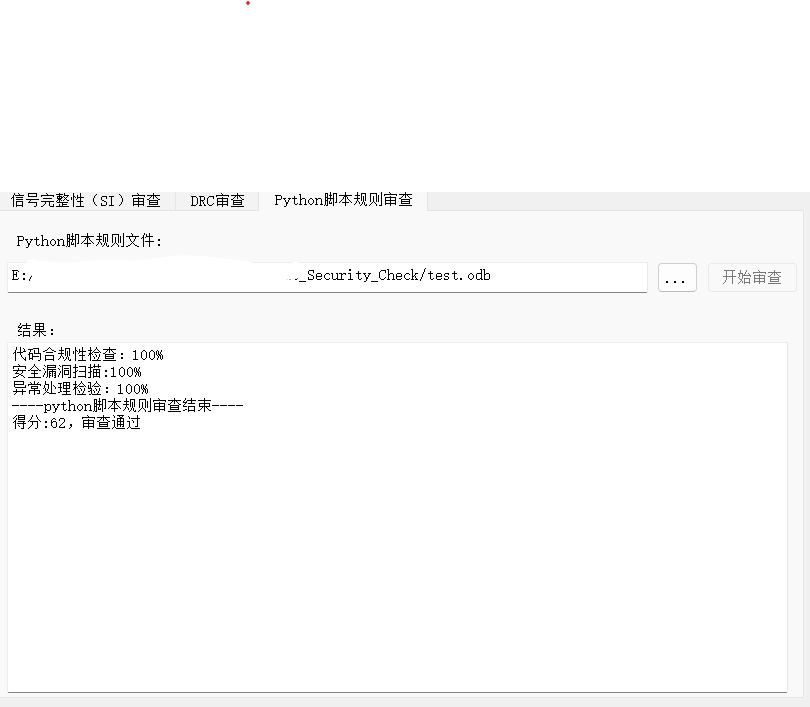


图5.6 脚本规则审查

#### 版图是什么

PCB设计也就是所谓的印刷电路板设计。在进行PCB设计之前，首先需要设计出符合我们要求的、具有一定功能的电路图。然后再以设计好的电路图为基础，设计出 PCB板上各个器件的位置布局，并且用合适的走线将这些器件的焊盘连接起来。最后，要对PCB板进行优化从而完成PCB版图的设计。这个设计完成的PCB板就能够实现电路设计人员所要实现的功能。在设计版图的过程中，不仅要考虑PCB板上各个电子元件的位置布局，各个电子元件焊盘之间走线的方式，以及 PCB板上过孔所产生的影响，还要考虑到PCB板的散热和电磁兼容等其他因素。由于PCB板具有很多优点，例如PCB上面可以集成很多电子元件，具有很高的密度；PCB板工作的可靠性较高并且设计起来比较容易；电路设计人员可以很方便的对PCB板进行测试和维护，因此目前的各种电子产品中都广泛地运用PCB板。

## 硬件木马分析

### 硬件木马介绍

硬件木马（hardware Trojan, HT）又称为电路“后门”，指的是在芯片或者电子系统中故意植入的特殊模块或者设计者无意留下的缺陷模块，在特殊条件触发下，该模块能够被攻击者利用而实现具有破坏性的功能。硬件木马的威胁与木马程序的威胁类似，它主要包括破坏芯片性能、更改芯片功能、窃取机密信息、实现信息泄漏、设备功能篡改、数据丢失、拒绝服务和系统瘫痪攻击。

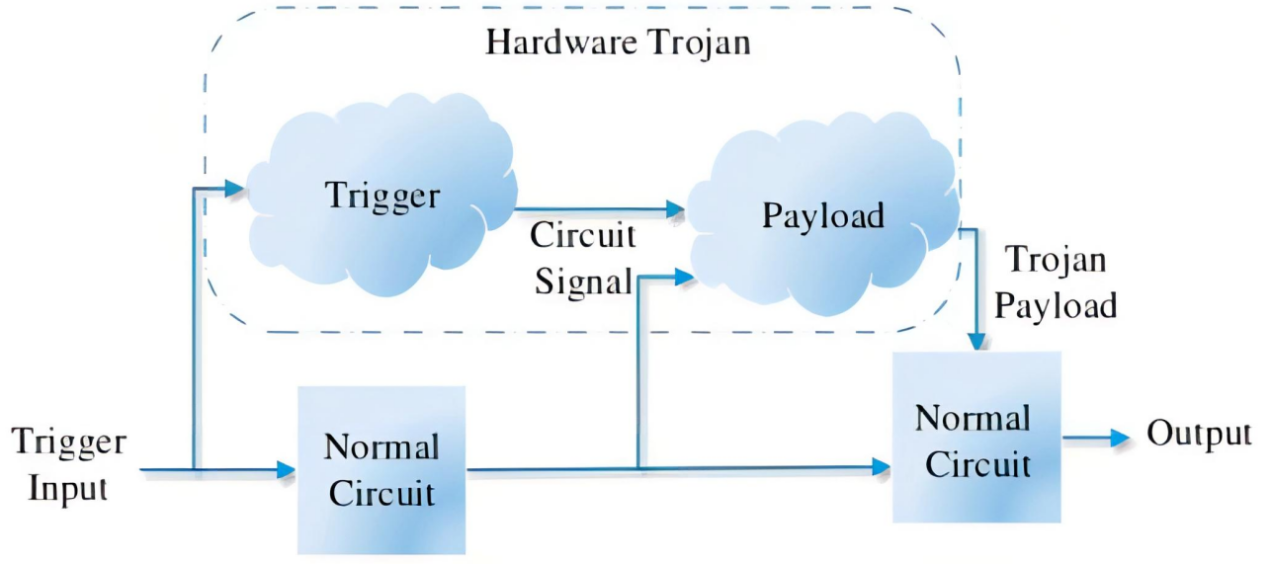


图6.1 硬件木马结构

硬件木马相比于软件木马更棘手的地方在于，软件木马可以通过删除程序来保证系统的纯净，但是硬件木马如果在流片之后发现就无法删除，无法挽回损失。从这一点上来看硬件木马的危害性更大。

#### 软件界面和功能介绍

硬件木马分析界面分为以下几个部分：连接、木马检测、逻辑测试、侧信道信息曲线，如下图：

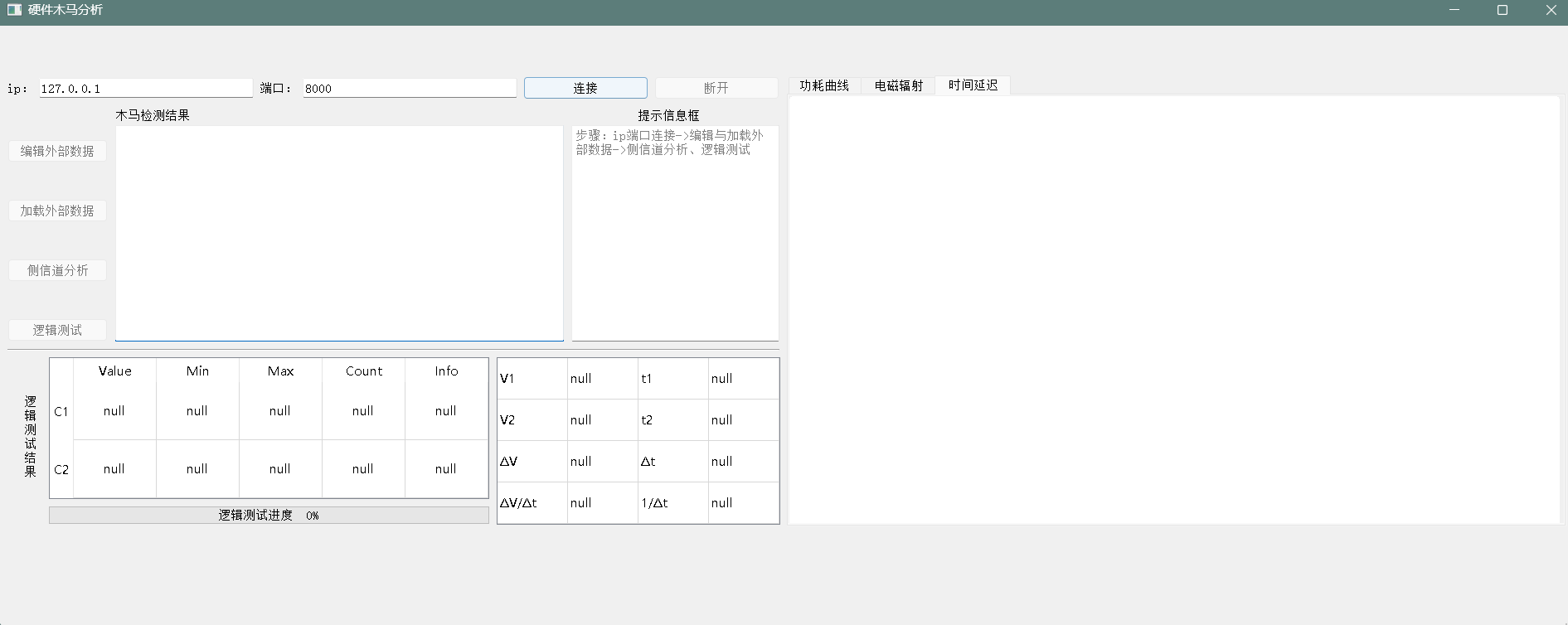


图6.2 硬件木马界面

当打开硬件木马分析界面的时候，首先需要通过ip地址和端口连接到对应芯片，在左上角一行输入芯片对应的ip地址和端口，在点击连接，系统即可连接到芯片，连接到芯片后提示信息框会给出文字提醒，同时木马检测部分的按钮会解锁一部分，如下图：

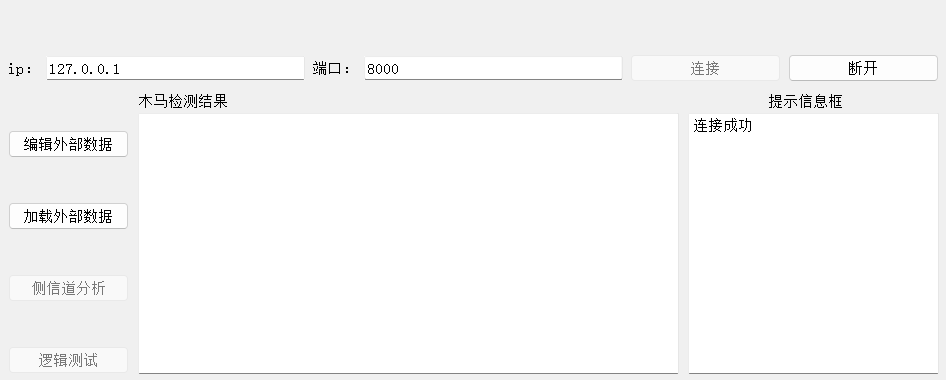


图6.3 连接成功

对于ip地址和端口，系统会自动检测格式合理性，如果检测到格式有问题，会弹出窗口提醒用户纠正错误，如下图：

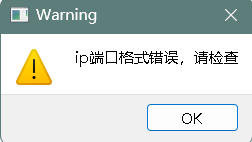


图6.4 格式错误

完成芯片的连接后，点击“编辑外部数据”按钮，即可编辑芯片的输入参数，如下图：

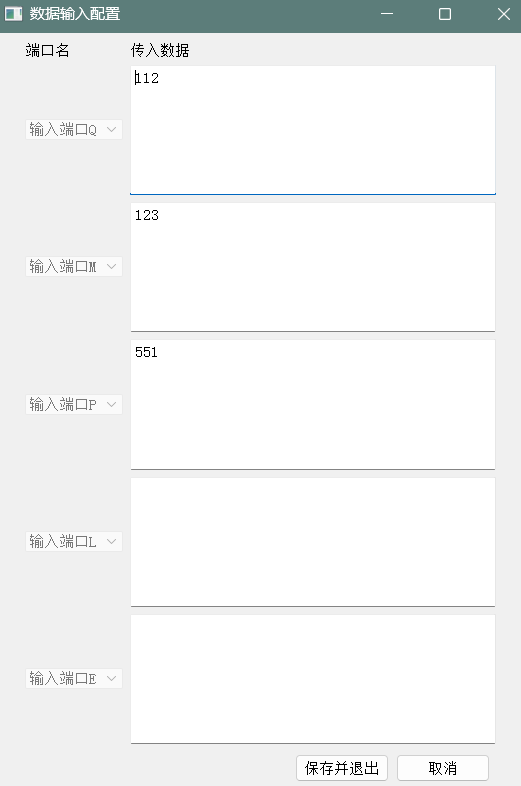


图6.5 输入参数配置

完成输入参数的设置后，点击保存并退出，系统会自动记录输入参数信息，然后在硬件木马分析界面点击“加载外部数据”按钮，即可将输入参数变成预加载状态，该状态会检测与芯片的数据传输线路是否通畅，并把几个端口的数据预加载在线路中，以方便及时读取侧信道信息，如下图：

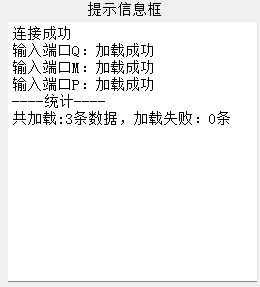


图6.6 加载外部数据

完成数据的预加载后，“侧信道分析”按钮解锁，可以点击“侧信道分析”按钮，输入参数就会传递给芯片，同时系统会记录芯片接收数据后的侧信道信息，并在一定的延迟后，显示在界面右边的功耗曲线、电磁辐射、时间延迟处，同时根据曲线特征分析硬件木马情况，如下图：

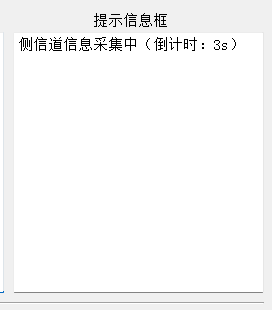


图6.7 侧信道信息采集

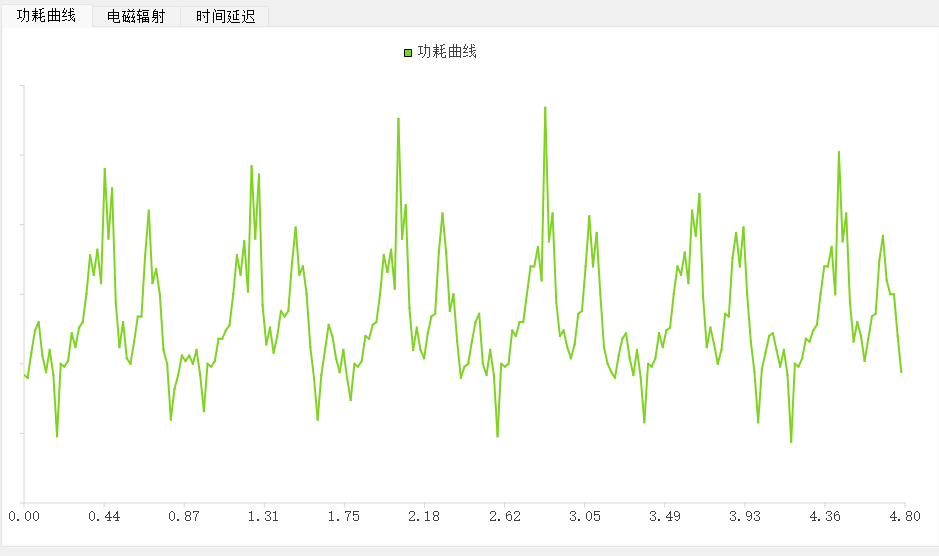


图6.8 侧信道信息

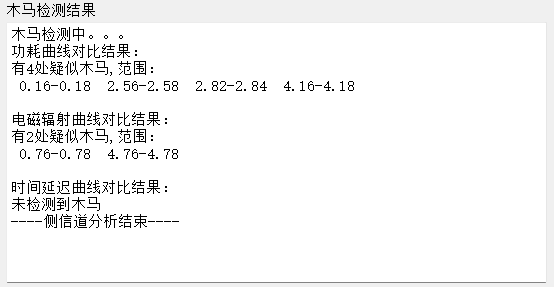


图6.9 木马检测结果

通过侧信道分析，可以通过对芯片的功耗曲线和电磁辐射对芯片进行硬件木马检测，以此推测处硬件木马可能出现的时间段。

在完成侧信道分析后，可以点击“逻辑测试”按钮，对芯片进行逻辑测试，以探出芯片的边界信息，如下图：

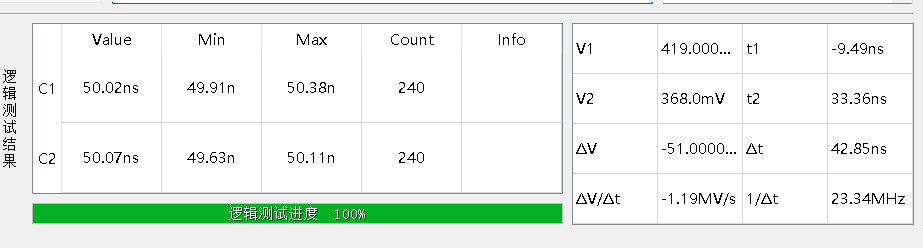


图6.10 逻辑测试结果

#### 什么是硬件木马

硬件木马是指在集成电路设计或制造过程中被恶意篡改或植入的微小冗余电路，它能够在某些特定条件下激活并且对原始电路产生影响，有些影响甚至是破坏性的，包括泄露电路中的关键信息、改变电路的功能使之失效、严重的甚至

能够摧毁电路。

硬件木马具有以下四个基本特点：

1. 微小性，木马单元的面积在整个电路面积中所占的比重微乎其微，木马的结构一般比较简单，只占电路中的一个或几个逻辑门，有些甚至复用原始电路中的逻辑。
2. 潜伏性，通常情况下，木马在电路中处于静默状态，即木马电路的存在不会影响电路的正常功能。只有满足某种预先设定的激活条件或达到一定的激活时间木马才会激活并发挥作用。
3. 隐蔽性，木马的输入信号一般来源于原始电路的内部，可以在电路的任意位置进行植入或改动。随着集成电路设计复杂程度的提高，对芯片进行反向解剖的难度越来越大，代价越来越高，因此木马在潜伏状态下是很难被检测出来的。
4. 破坏性，木马一旦激活将会对电路造成严重影响，有些木马会改变电路的功能，修改电路的输出；有些能够窃取电路中的信息，以某种方式反馈给攻击者；有些木马能够干扰电路的功能甚至是摧毁整个电路。

如图6.11所示，木马的基本结构分为两个部分：触发部分( Trigger )和有效载荷部分( Payload )。

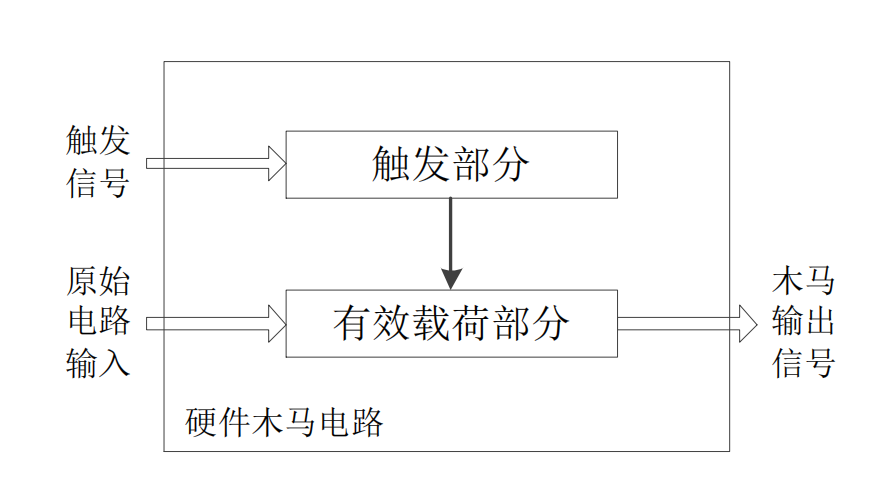


图6.11 硬件木马的基本结构

触发部分就是使木马激活的控制机制。大多数的硬件木马在电路中长期处于潜伏状态，当满足某种预设的条件时，木马才能被激活。触发部分通常是电路中一些内部节点的信号，也有些木马自身带有传感器，当感知到外界信号如温度、电磁辐射等发生变化时就会激活。

图6.12列举了两种简单的木马激活机制，第一种激活机制将电路中两个翻转概率极低的节点a、b经过一个两输入与门后输出，a和b两个节点通常情况下都为低电平，只在极少情况下才能出现两者都为高电平的情况，这时可以将木马的激活条件设定为与门的输出值c为高电平；第二种激活机制通过一个n位计数器实现触发，当计数器的数值达到预设目标即n位触发器的输出都为1时，与门的输出y的值从0变为1进而触发有效载荷部分,使原始电路中的信号输出改变。

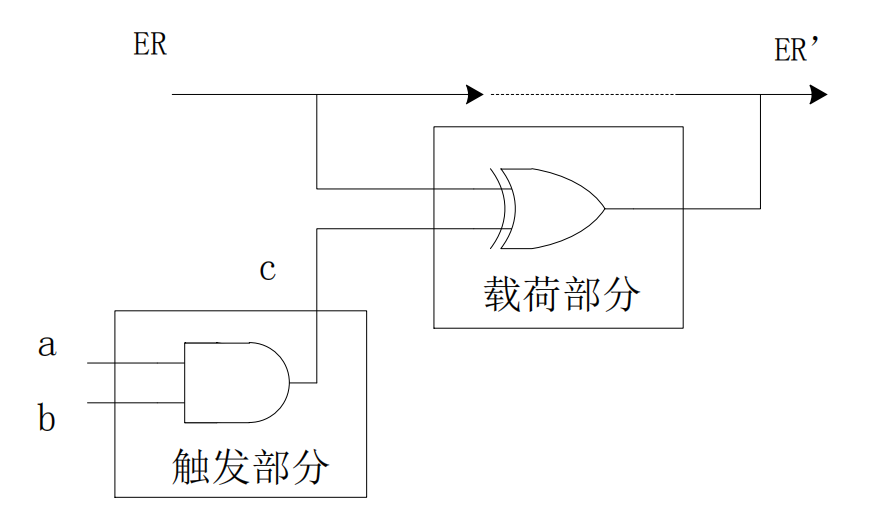


图6.12（a） 基于逻辑门的硬件木马激活机制

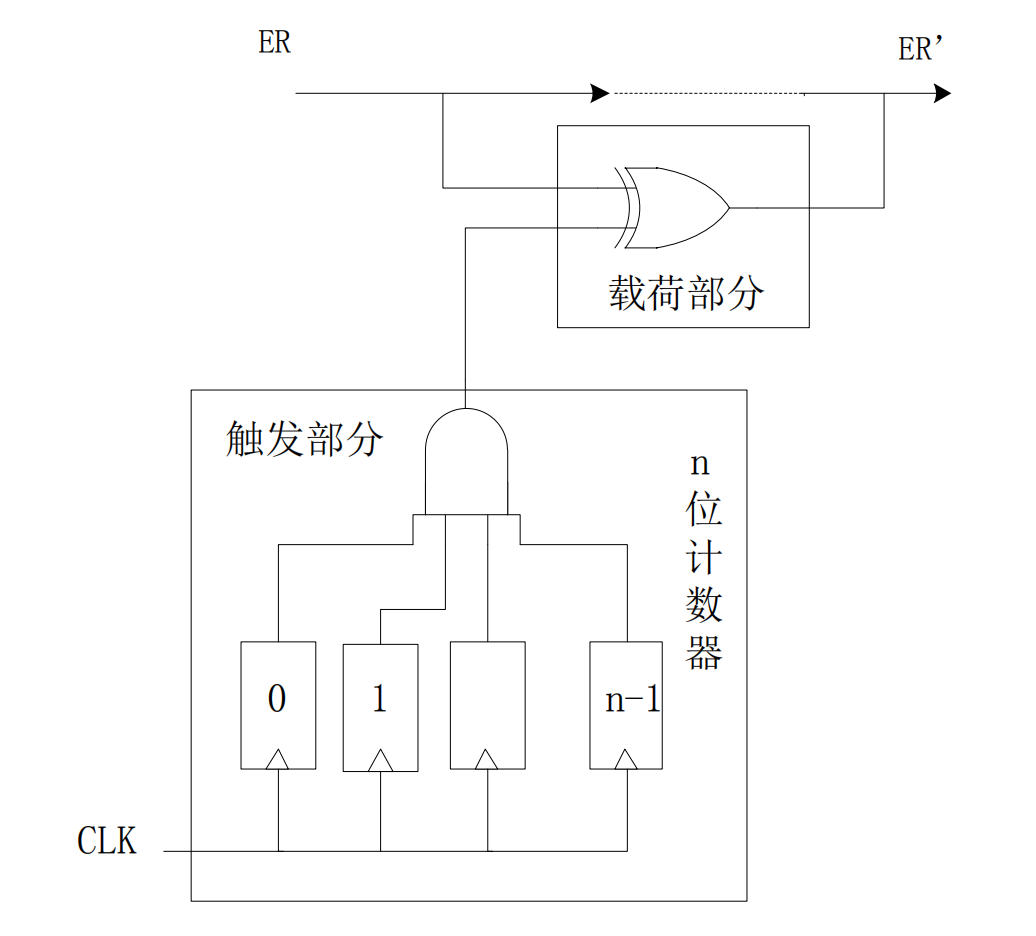


图6.12（b） 基于计数器的硬件木马激活机制

有效载荷部分是木马激活后发挥作用的部分。这部分的功能有些与电路相关，有些与电路无关。例如将电路中的某个信号接入有效载荷部分，木马激活后会修改关键信号的值再输出给电路，干扰芯片功能，使其不能正常工作；或者只收集电路运行中的一些关键数据，存储起来然后通过某种方式提取。

如图6.12所示，原始电路中的信号ER被连接到木马的有效载荷部分，当木马激活，即触发信号c为1时，信号ER'的值与原始值ER相反。

硬件木马在电子系统中无处不在，例如一些计算机或先进打印设备，若在其内部存储器中植入硬件木马电路，在用户不知情的情况下，木马会将存储器中的数据转移到某个隐蔽位置，攻击者利用仪器维修的契机或者通过网络将窃取到的信息读出来。有些功能更强大的木马在芯片中植入射频发射模块，实时将信息通过天线发送给攻击者。还有一些木马植入芯片的控制电路中，当木马在满足触发条件激活后能够发挥破坏性作用，使电路的功能改变或失效。

总体而言，木马主要会产生以下几方面的危害:

1. 潜伏在电路中并实时窃取内部的信息，通过隐秘渠道存储或向攻击者发送重要数据。
2. 修改电路的功能或者更改电路中的存储数据。
3. 通过破坏系统防火墙，降低系统的安全防护级别，配合软件木马攻击计算机系统。
4. 对芯片产生破坏性影响，使芯片工作异常，使芯片老化甚至功能失效。

#### 历史上硬件木马造成的数据与经济损失

1991年海湾战争中，美军通过激活设置在打印机芯片中的“木马”，侵入了伊拉克防空指挥系统，导致后者战争伊始即陷入瘫痪，关键性的制空权“拱手相让”。2007年的“果园行动”中，以色列空军第69战斗机中队18架非隐身战机，悄无声息地突破叙利亚在叙以边境部署的先进俄制防空系统，对纵深地带的一处疑似核设施实施精确轰炸，并从原路安然返回。外媒报道称，根据美国国防部供应商匿名提供的情况，一个“欧洲芯片制造商”在叙军所用雷达的微处理器中加入了可远程访问的“毁灭开关”，空袭时被适时“唤醒”，致使叙防空雷达临时关闭。2015年7月，部署在土耳其边境的德国“爱国者”防空导弹系统遭受不明网络攻击，短暂失控。据推测，攻击者可能是通过入侵指挥控制系统的计算机芯片实施网络攻击的。

早在2005年，美国国防部就已注意到“集成电路供应链问题”，提出了硬件安全这一概念，并警告称“非认证的硬件电路会严重威胁国防安全”。这在一定程度上反映了美国军方在硬件安全认知上的超前性。但即便如此，十余年后的今天，其仍然深感“硬件木马”的威胁急剧上升，足见防范硬件安全风险是何其艰难。一方面，集成电路供应链不可控因素多，防范“硬件木马”植入困难重重。缩短开发周期、整合优势技术、降低制造成本成为集成电路制造全球化的动力。一个集成电路产品通常需要在多个国家、多家单位流转，经过设计、综合、布局、布线、加工、测试、封装和组装等环节。而每一个不受控的环节，都存在着“易受攻击之窗”，存在着恶意修改原始电路、插入额外电路、器件等风险。美国国防部称，F-35战机可能使用了数个“危险的芯片”。但更可怕的是，制造一架完整的飞机，需要成千上百个零件供应商，仅查明供应商的来源就是件非常不容易的事。2017年3月，美国国防科学委员会发布报告称，美军武器系统可能已经被注入“硬件后门”，一旦在这种状态下实战，“敌方完全能够令美方武器陷入瘫痪”，但也坦承“国防部自行建立并维护‘代工设施’以制造微芯片恐怕不具备可行性”。

而在中国，2007年，一种可以穿透各种还原软件与硬件还原卡的木马病毒肆虐。由于该木马采用电子狗的照片作为，因此被很多受害者形象地称为“机器狗”。严格意义上来说，“机器狗”是一个木马下载器，它的主要任务是把各种游戏盗号木马输送到受害者电脑里。它的最大特点是能够穿透网吧常用的还原体系，并快速感染整个局域网，长期驻留在系统底层，对当时习惯在网吧打游戏的玩家来说无疑是一场浩劫。有不完全统计显示，机器狗对网吧行业造成的经济损失高达数十亿人民币。

### 硬件木马分类

随着硬件木马概念的提出，涌现出非常多不同场景下的、不同攻击形式硬件木马，越来越多的硬件木马设计和检测、防御机制研究需要对硬件木马进行分类，以研究它们的共性和特性。从不同的角度出发，硬件木马可以有多种分类方式。6.1.2节中提到根据木马的结构和功能可以将其分为触发部分和有效载荷部分。

硬件安全专家Swarup Bhunia和Mark M. Tehranipoor在他们的著作《The Hardware Trojan War》将硬件木马分成5类，分别按照植入阶段，抽象级别，激活机理，触发效果和植入位置进行了细致的划分； Samer 等人在此基础上明确了木马属性之间的关系，并且阐述了芯片生命周期与硬件木马植入位置和方式，在此基础上将硬件木马如图6.17分成8类：

1. 按照不同的植入阶段将硬件木马分类，与硬件产品制造流程相关。芯片的客户与设计公司通过合作协商交出一份芯片定义说明，包含了设计规范，封装和测试要求等。在芯片规范制定阶段，设计者可以确定诸如大小，结构，类型，预期功能，功率，时序和延迟之类的参数。恶意设计者可以通过修改协议和功能来植入木马，规范的改变将直接导致芯片的改变。在设计阶段，设计人员按照规范的功能，时序，物理约束等要求进行芯片设计，对于芯片中的一些常见功能，设计人员可能会使用第三方IP，通用设计工具以及标准单元库等等。如果这些组件来自不可信的第三方，则它们可能包含恶意组件。测试阶段是检测木马最容易的阶段，但是，在不可信第三方工厂进行测试可能会导致测试向量泄露和检测缺陷。攻击者可以利用测试权限来忽略或掩盖用于触发硬件木马的测试向量。也可以通过设计让木马的触发向量避开测试向量。在组装阶段，组件之间的每个接口都是可能的木马植入点。在接口处，攻击者可以利用端接不当、对电磁耦合屏蔽不当等疏忽来注入故障或收集关键信息。
2. 按照硬件木马逻辑类型分类。组合型木马使用一个或多个电路位置上的特定逻辑值作为触发信号。时序型木马被触发后，还需经过一定的的运行时间，之后再产生恶意行为。

组合逻辑型木马的触发电路依赖于原始电路内部节点的稀有逻辑值。在图6.13中，显示了一例组合逻辑型木马，当A=1和B=1在各自的节点上发生时，触发条件满足，导致有效载荷节点S的值被修改为S'。通常，攻击者会试图找到这种罕见的组合，而这种组合在常规的电路测试中不会被检测到。

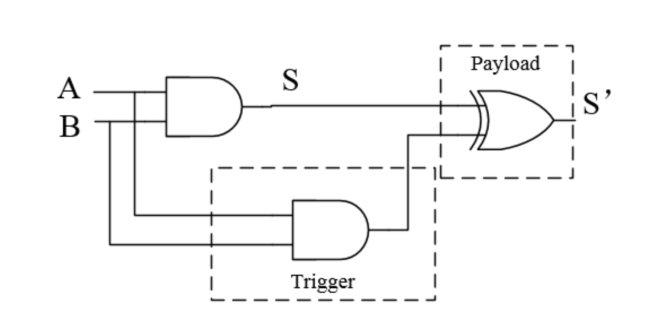


图6.13 组合逻辑型硬件木马

时序逻辑型木马可以看作是“定时炸弹”，在长时间的电路工作中发生一系列稀有事件后，实现其恶意效应。最简单的时序逻辑型木马可以由简单的基于时的计数器构成，该计数器在达到某个计数后，激活有效载荷。在图6.14(a)中，当计数值达到2k-1时，同步k位计数器被触发，从而使电路中的原始信号ER被修改为ER\*。图6.14(b)显示了异步触发的硬件木马，该木马计数器中的计数值不再由时钟沿进行控制，而是由与门逻辑的输出翻转决定，当达到攻击者设置的阈值后，触发器将被激活，从而导致原始信号ER被修改为ER\*。

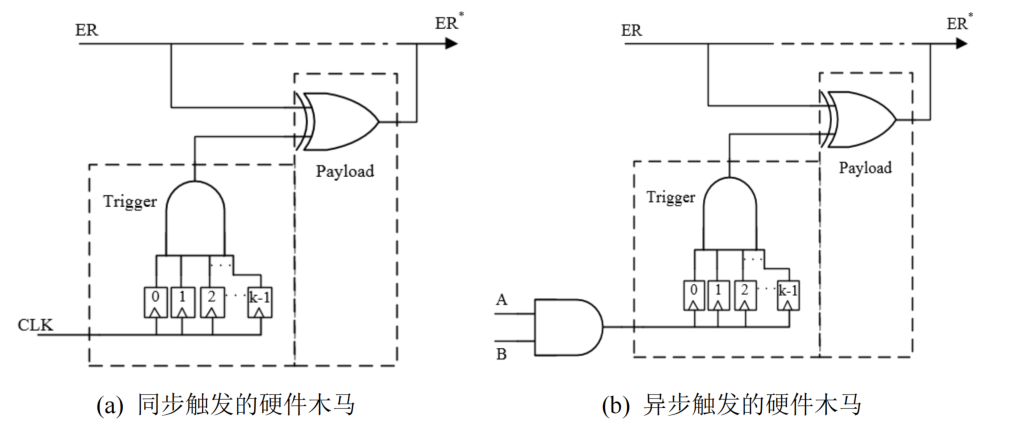


图6.14 时序逻辑型硬件木马

1. 按照硬件木马触发方式划分。常开型硬件木马相比于触发型硬件木马，没有特定的触发条件，始终处于活动状态。触发型木马的触发条件可以是外部或内部的。外部触发的木马被天线或传感器接收的外部信号激活。内部触发的木马经过内部系统中特定的的一个或多个事件的序列激活。如图6.15所示。

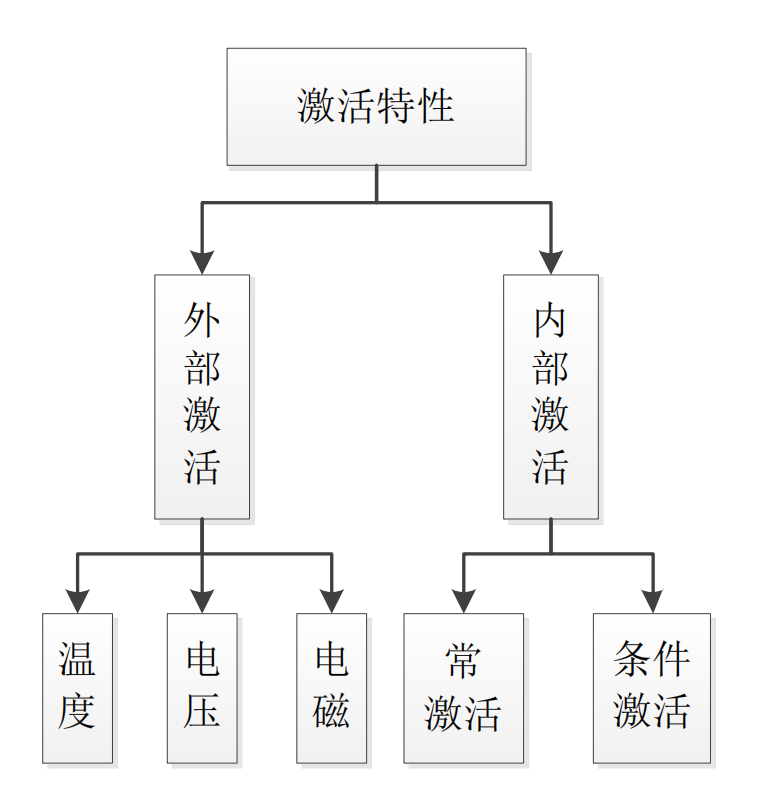


图6.15 基于激活特性的分类

外部激活是指木马的触发源自芯片外部，例如自身设计中包含传感器的木马，当外部环境如温度、电压、电磁辐射等发生改变或攻击者故意设定某种触发条件时，可以使木马激活。

内部激活又包括常激活和条件激活两种，常激活木马是一种持续处于激活状态的木马，例如植入加密芯片中窃取密钥信息的木马会随时监听芯片的工作信息，将密钥通过某种方式存储下来或发送给攻击者；条件激活木马可以通过电路中某个节点的逻辑输出或计数器的输出控制木马的激活。

1. 按照硬件木马在芯片布局中的物理特征分类。有些硬件木马体积小，因此通过侧信道分析实际上无法检测到。面积较大的木马在设备中可以利用未使用的电路来避免检测。木马的分布基于木马在芯片上的布局，群集木马的拓扑结构中，各个组件彼此靠近；分布式木马具有分散的拓扑结构，可以零星分布在整个芯片中。一些木马使用更改后的布局结构，其中对现有布局进行了修改。如果添加了布局，则称为增强电路。如图6.16所示，物理特征是指硬件木马呈现的外在表现，主要包括木马的类型、分布、尺寸以及结构。

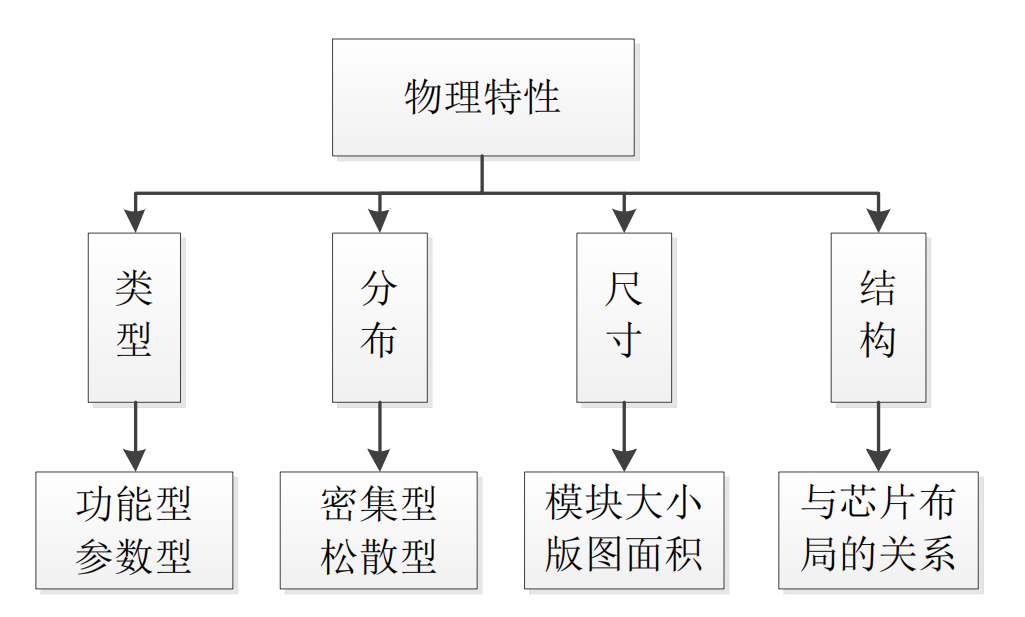


图6.16 基于物理特性的分类

木马的类型包括功能型和参数型，功能型木马是指在芯片设计当中植入冗余的电路单元，木马激活后由冗余单元发挥作用；参数型木马是指木马电路复用原始电路中的逻辑单元，通过修改电路单元某些逻辑的连接关系，进而达到激活后改变电路功能的作用。

木马的分布包括密集型和松散型两种，对于芯片中有剩余资源的电路，木马电路可以紧密排布。当芯片的资源利用率较高，木马的功能多样或者为了达到更好的隐蔽效果，攻击者通常会将木马的结构打散后植入到电路的整个版图中。

木马的尺寸是指木马电路在芯片中所占的版图面积或等效门数，显然木马的面积越小，等效门数越少，木马的检测难度越高，越难被发现和移除。

木马的结构是指木马设计与芯片布局之间的关系。通常攻击者植入木马都会在版图层次完成，这种方式只对芯片局部进行修改，不会大面积影响芯片的结构。

如果在版图生成之前的步骤进行木马的强行植入，需要重新进行后续的版图生成工作，虽然能够增强木马的隐蔽性，但会影响芯片的结构布局甚至是芯片尺寸。

1. 按照硬件木马功能特征分类。功能型木马的攻击目标是改变或增添设备的功能。参数型木马的目标是使设备电路的功耗，热量和延时分布等参数发生异常，改变参数可以通过弱化晶体管特性、修改导线的长度和/或厚度或更改物理几何形状来实现。
2. 按照硬件木马所在的抽象级别分类。在系统级，可以通过更改互连，硬件模块或通信协议的方式植入木马。在RTL级别，寄存器型变量，线网型变量和布尔函数在模块中进行了描述，攻击者可以通过修改RTL代码植入木马。在网表级别，硬件木马植入难度增加，由于攻击者接触到网表可能被打平，因此对于电路内部逻辑功能的掌握并不细致，但是恶意的更改依然是有可能发生的。在晶体管级别，攻击者可以通过控制电路参数植入木马。例如，可以通过调整现有晶体管大小或插入新晶体管以修改电路功能和特性。在物理级别，攻击者可以通过修改晶体管或布局，来改变影响电路可靠性或电路功能的具体参数。
3. 按照硬件木马触发后对目标电路的影响分类。这些影响包括：功能改变，信息泄漏，可靠性降低和拒绝服务（DoS）。功能改变是由引入的冗余逻辑或绕过现有逻辑以产生意外结果的木马引起的。硬件木马会导致电路关键信息通过秘密或现有通道泄漏，比如电路板上的多余引脚，JTAG或RS232接口。硬件木马还可以通过更改接口，功能或电路特性，例如路径延迟和功耗，来降低可靠性。功耗增加可能会导致电路的环境温度升高到正常工作水平以上或导致耗电加快。DoS木马可以通过修改设备参数以耗尽片上资源。例如电源和内存，或引入计算延迟以降低性能或产生故障。
4. 按照硬件木马在系统中的不同位置分类。硬件木马可能位于处理器、内存、输入/输出、电源、时钟等模块或网络中。植入处理器逻辑单元中的木马可以更改指令或执行周期。植入内存单元或接口中的木马可以创建错误的地址，修改内存内容或启用/禁用读/写指令。输入/输出外围设备通过通信和数据总线，例如串行端口，与外部设备接口。植入此处的木马可以修改数据或更改外部设备与芯片进行通信的方式。植入电源控制模块中的木马可以改变芯片的电流和/或电压供应状态以引起故障或异常行为。时钟网络中的木马会导致时钟频率变化，跳过或停止提供给芯片模块的时钟信号。

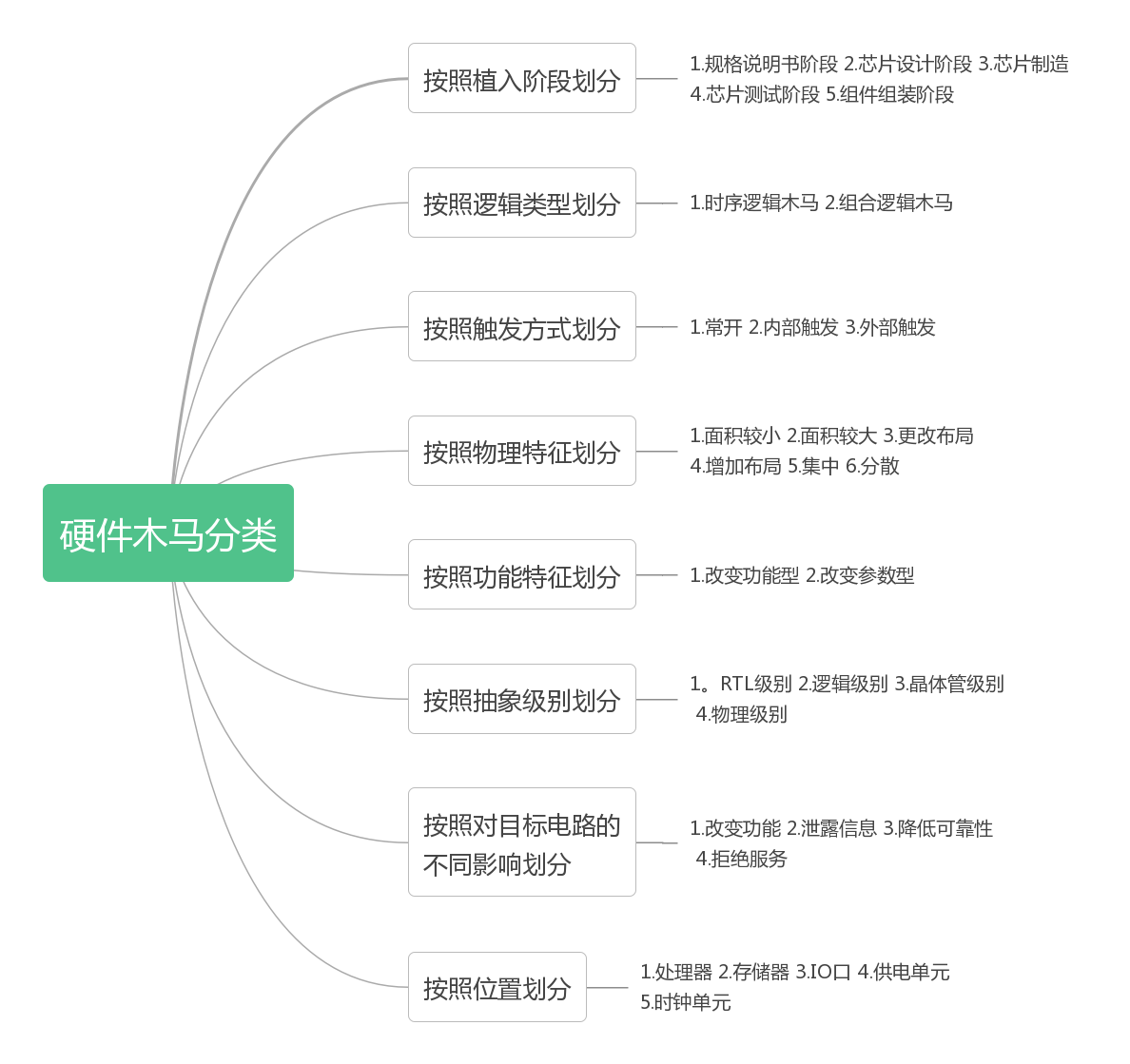


图6.17 硬件木马分类

#### 硬件木马的工作原理

硬件木马电路由两部分组成：触发电路和负载电路。原始的硬件电路被称为木马的载体电路。触发电路是用于激活木马恶意行为的开关。为了避免被轻易地检测出来，硬件木马仅在罕见条件下激活自己的恶意行为。触发电路时刻监测电路的中间信号，一旦满足木马的触发条件，便会激活硬件木马的负载电路。负载电路是实现木马恶意功能的电路。硬件木马的恶意行为多种多样，例如，毁坏系统、修改电路输出、泄露敏感信息等。硬件木马的触发电路决定了它的隐蔽性，负载电路决定了它的恶意功能以及杀伤力。如下图，展示了一种结构最简单的硬件木马电路。由图可以看出，该木马的触发电路为一个与门，负载电路为一个异或门。与门的输入端口为电路的中间节点A和B，若 A和 B的状态为00，01或10时，中间节点 S 值不变被传输至S\*；若A和B同时为 1，则木马电路被激活，木马激活后，负载部分的异或门使得S被取反，输出至S\*，从而改变电路的输出。

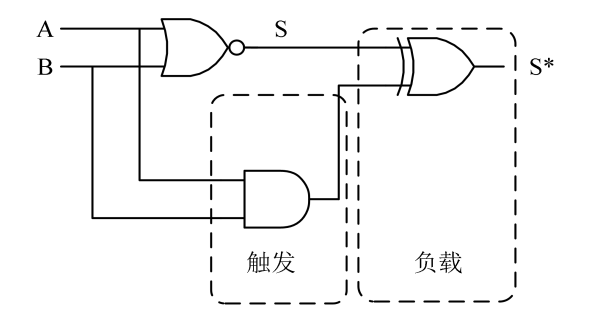


图6.18 一种简单的硬件木马

硬件木马具有恶意性、硬件相关性、功能固定性和隐蔽性等特点。恶意攻击者通常为了实现自己的恶意目的，才会耗费大量的时间和精力去植入硬件木马，因此硬件木马都具有恶意性。硬件相关性是指硬件木马是依赖于硬件电路而存在的。因此，硬件木马设计者必须充分了解目标载体电路的功能和电路结构之后，才能设计出实现特定功能的硬件木马。硬件木马一般是在芯片的设计和制造阶段就植入载体电路中的，因此在芯片制造完成后木马的电路结构和功能将无法改变。为了使得硬件木马能够躲避过各类检测手段，实现特定的功能，这就要求硬件木马的隐蔽性强，即触发概率低，在未激活状态下隐藏在电路中难以被发现，在激活状态时实现该木马的特定功能。

### 硬件木马检测技术

IC产业链的全球化，导致木马可以植入到设计制造流程中的各个阶段。为了确保IC的可信性，可以根据IC设计中每个阶段的信任级别，采用不同的方法来检测硬件木马。但由于硬件木马的样式繁多且功能各异，目前还没有一种统一的适用于所有木马的检测方法。一种木马检测方法的有效性、可靠性可以根据以下几方面评估：

1. 适用于不同类别的硬件木马；
2. 非破坏性，不能对芯片造成物理上的损伤或破坏；
3. 能够处理小型木马；
4. 检测成本尽量小；
5. 检测时间要短；
6. 不需要完全激活木马；
7. 能够适应实际制造过程中的各种工艺变化和测量噪声。

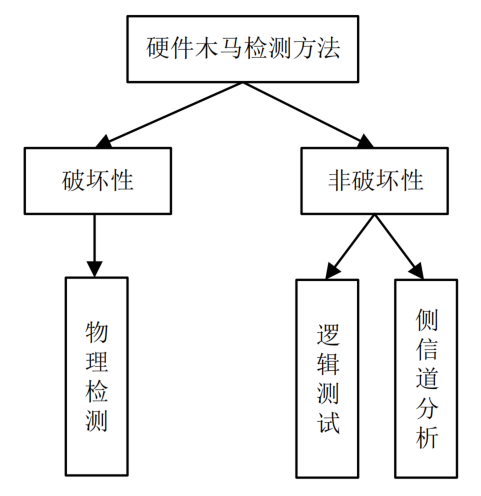


图6.19 木马检测方法分类

基于现有的硬件木马检测方法，可将其分为破坏性和非破坏性两个检测类别，如图6.19所示。破坏性检测需要对设计进行逆行工程，以便对木马的植入进行物理检查，该方法适用性小，受限度大，在实际的检测中很难应用。非破坏性检测研究方向较广，主要包含逻辑测试和侧信道分析等检测技术，这是目前硬件木马检测领域的两个主要研究方面。下面针对破坏性和非破坏性检测这两个广泛研究的方面进行阐述。

#### 物理检测技术

物理检测技术，即以逆向工程为主的失效分析技术。在硬件木马出现的早期，还没有行之有效的硬件木马检测技术，检测人员只能采用这种破坏性的物理检测方法。首先使用打磨、腐蚀等方法打开待测芯片封装，然后对解除封装后的裸片逐层解剖，之后利用扫描电子显微镜等高精度仪器逐层提取待测芯片的版图图像信息，利用提取后的图像还原待测芯片的电路网表。最后将待测芯片的电路网表与原始电路网表进行对比分析，判断待测芯片是否被恶意攻击。

尽管该技术简单直接，但仍存在一些缺陷。首先，这种检测方法非常费时且随着集成电路特征尺寸的缩小变得愈发困难且昂贵；其次待测芯片被逆向分析后，电路结构遭到破坏，将无法继续使用；此外，这种破坏性检测方法不能分辨剩余的芯片有无木马，特别是当木马被有选择的插入到一小部分芯片时。因此，这种破坏性检测方法获得的信息仅能证明单个样品的可信性。尽管该技术存在一些问题，但它能够用来收集获取一组无木马芯片的参考数据，用于构成侧信道分析所需的黄金芯片模型。

#### 逻辑检测技术

逻辑测试技术是一种以自动测试向量生成技术(Automatic Test Pattern Generation, ATPG)和可测性设计(Design For Test, DFT)技术为基础的硬件木马检测方法。通过ATPG产生大量的测试向量，以最大程度激活芯片中未知的硬件木马并将其结果输出，对比端口实际输出与预期输出结果的差异，来判断待测芯片中是否存在硬件木马。其中DFT技术主要用于生成扫描链，增强内部节点的可测性，使得木马产生的影响能够在输出端口显现。该方法的优势在于，对于工艺偏差和环境噪声具有很好的鲁棒性和稳定性。

当然，采用逻辑测试技术检测木马也会遇到困难。首先，由于硬件木马的隐蔽特性，木马在大多数情况下处于休眠状态且不会产生任何异常行为，只有在电路运行到满足某些特定条件时，木马才会激活。为了在检测过程中激活硬件木马，检测人员需要采用穷举的方式生成测试向量。随着电路规模的增大，测试向量将会呈指数增长，这也导致整个测试流程需要耗费大量的时间与精力；其次，当硬件木马的激活与输入端口无关时，生成的测试向量无法激活木马，则整个测试过程无法观察到木马的影响；最后，该检测方法只能用于检测对输出端口产生影响的硬件木马，即功能型木马，如果木马产生的影响无法传播到输出端口，则检测方法无效。

#### 测信道分析技术

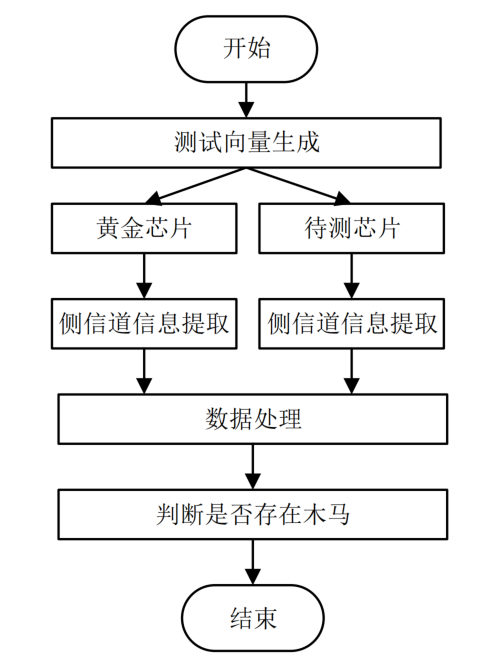


图6.20 侧信道检测流程

侧信道信息指的是电路正常运行过程中产生的各种物理信息，如功耗信息、延迟特性、电磁信息和温度信息等，而侧信道分析(SCA)主要通过检查这些物理信息来还原芯片的细节。应用侧信道分析检测硬件木马基于以下事实：即使硬件木马处于未激活状态，也会导致各种侧信道参数发生不规律的改变。通过将待测芯片的参数信息与黄金模型进行对比，根据两者是否存在差异，即可完成硬件木马存在的判断。图6.20给出了侧信道分析检测方法的流程。

根据所选的侧信道参数的不同，可以延伸出不同的侧信道检测方法。常见的侧信道检测方法主要包括功耗和延迟参数分析。

1. 基于侧信道功耗的硬件木马检测方法

侧信道功耗分析是一种通过测量功耗和功耗的分布来确定芯片各种特性的方法，该方法能够有效的区分电路行为的不规则性。有两种功耗可以作为侧信道功耗分析的指标：静态功耗和动态功耗。

静态功耗指的是当电路处于等待或不工作状态下时静态电流所造成的功耗，主要是由晶体管的泄露功耗组成。当硬件木马植入到电路中时，即使硬件木马未激活，插入到原始电路中的额外木马门也会产生额外的泄露功耗。该功耗的大小与木马门的数目成正相关，因此随着木马规模的增大，木马引入的泄露功耗越显著；动态功耗主要归因于晶体管的状态切换，反映了电路内部的开关活动。当木马内部存在任何开关活动时，电路所产生的动态功耗就会增加。此外，由于电路中的电压相对稳定，因此侧信道技术也可以用静态电流和瞬时动态电流来表征功耗。

在基于侧信道功耗检测方法中，将待测芯片的功耗数据与黄金芯片的功耗模型进行对比，以检测硬件木马产生的异常功耗增加。该检测方法的效率与木马的激活程度有关，因此测试向量的生成在侧信道功耗的分析中发挥着重要作用。一个好的测试向量生成技术能够有效的增加木马电路的开关活动，增大木马对电路功耗的影响，提高木马检测效率。但是，该检测方法存在着很大的挑战，随着电路规模的不断扩大，小型木马产生的功耗相比于整个电路功耗很小甚至可以忽略不计，很容易逃脱检测。此外，小型木马产生的微弱功耗很容易被工艺偏差和测量噪声产生的变化所掩盖。

1. 基于侧信道延迟的硬件木马检测方法.

随着电路规模的不断扩大，芯片中存在着大量路径，每一条路径的延迟都代表着芯片整体特性的一部分。植入到原始电路中的硬件木马以多种方式影响电路延迟，无论是有效载荷插入到路径中产生的直接门延迟，还是木马引入的互连产生的额外电容负载，都会影响原始电路的时序特性。也就是说，不管硬件木马激活与否，一旦硬件木马植入到电路中，它至少会增加电路中一条时序路径的延迟。这意味着无木马植入的电路的信号传输时间比有木马电路的少。基于侧信道延迟的硬件木马检测步骤为：

1. 收集芯片的侧信道路径延迟特征信息：对一定数量的样本芯片施加测试向量获得芯片的路径延迟信息。
2. 将所得到的路径延迟信息构建成检测所需的指纹。
3. 对待测芯片施加和样本芯片相同的测试向量，获得相应的路径延迟信息。
4. 硬件木马检测：将待测样本的路径延迟信息与构建的路径延迟指纹进行比较以检测待测芯片中是否含有硬件木马。

此外，DFT技术也可用于增强基于路径延迟的硬件木马检测技术，对于没有全扫描的时序电路，只能够测量源自主输入并终止与主输出的路径延迟，无法有效的覆盖硬件木马。采用DFT技术插入扫描链有助于测量寄存器与寄存器之间的组合路径延迟，能够更有效地获得受硬件木马影响的时序路径信息。

相比于其他侧信道检测技术和逻辑测试技术，基于延迟的硬件木马检测方法完全不需要激活木马，且在输出端无法观察到木马效应的情况下仍能获得检测结果。遗憾的是，现有的基于延迟的硬件木马检测方法仍面临以下挑战：

1. 确保最大的检测覆盖范围，可以检测放置和分布在非关键路径上的硬件木马。攻击者通常巧妙地将硬件木马植入到原始电路中，不会对关键路径产生影响，从而逃避基于延迟的故障检测。
2. 降低工艺偏差的影响。对于延迟较大的路径，木马产生的额外延迟很容易被工艺偏差所掩盖。

### 硬件木马对抗的未来发展与展望

随着半导体产业不断地发展，集成电路芯片对国家和民生安全发挥着重要的作用。集成电路设计与生产相互分离、各模块设计相互独立的产业供应链模式，使得芯片在设计、制造以及使用过程中不可避免地存在硬件安全隐患。硬件木马是指在芯片设计或制造阶段对硬件电路进行有意的修改或植入的结构，导致电路出现恶意的行为。硬件木马的出现给 国家的军事和民生安全带来了巨大隐患。因此，解决硬件木马的检测问题变得刻不容缓。现有的功耗侧信道检测方法需要“黄金电路”作为参考样本，使得硬件木马检测受限，可以进一步探究无参考样本的木马检测方法。